

# Contents

1	Introdu	uction	8
1.1	Wha	at is Intent-based Automation?	8
1.2	Inte	nt-Based Automation Use Cases and Flows	10
1	.2.1	Intent-Based Collaborative Troubleshooting	10
1	.2.2	Continuous Network Assessment	12
1	.2.3	Change Management and Assessment	14
1.3	Hov	v to Use This Cookbook?	15
1	.3.1	Sign in to Netbrain Lab for the First Time	16
2	Autom	ate Frequently Used Commands	21
2.1	Sea	rch and Add Devices to the Map	22
2	.1.1	Draw Devices from a Device Group	22
2	.1.2	Draw Devices from the Search Bar	23
2	.1.3	Save the Device Map	23
2.2	Qui	ck Intent Creation	24
2.3	Defi	ne Variables with a Visual Parser	26
2	.3.1	Define parser for ping <i><destination ip=""></destination></i>	26
2	.3.2	Define parser for Command Show ip route	
2.4	Defi	ne the Diagnosis	
2.5	Defi	ne Intent Output	
2.6	Dup	licate the Intent to Other Devices	
2.7	Diag	gnosis Tree	40
2.8	Viev	v the Result on Map	41
2.9	Crea	ate Intent and Summary Dashboard	42
2	.9.1	Intent Dashboard	42
2	.9.2	Summary Dashboard	44
2	.9.3	Add an Intent Dashboard to an Existing Summary Dashboard	47

3 Auton	nate Basic Device Health Checking	50
3.1 Sel	ect a Device	51
3.2 Ret	rieve Data and Parse Variables for Health Checks	53
3.2.1	Parse variables from show version	53
3.2.2	Parse variables from Show Process CPU	56
3.2.3	Parse variables from Show Interface	57
3.3 Def	ine Diagnosis for Health Checks	59
3.3.1	Define Diagnosis for Show version	59
3.3.2	Define Diagnosis for Show Process CPU	61
3.3.3	Define Diagnosis for Show interface	63
3.4 Mo	nitor F5 Virtual Server Status	67
3.4.1	Parse variables from show Itm virtual detail recursive	67
3.4.2	Define Diagnosis for Monitor F5 virtual server	70
3.5 Exe	cute Intents and Create Dashboards	78
3.5.1	Execute Intents	78
3.5.2	Intent Dashboard	79
3.5.3	Summary Dashboard	81
4 Enable	e Others to Use Your Automation	83
4.1 Aut	o Intent Wizard: Replicate Intent to Multiple Devices	84
4.1.1	Intent Template	85
4.1.2	Pre Decode	89
4.1.3	Auto Intent	92
4.1.4	Run the Auto intent:	93
4.2 Wra	apper Intent as End User Interface	96
4.2.1	Create Wrapper Intent	96
4.2.2	Execute Wrapper Intent	103
4.3 Rur	n intent with Chatbot	106
4.3.1	Create the Netbrain Chatbot	106
4.3.2	Use Chatbot	111
4.3.3	Add Auto Intent to Chatbot	114
5 Netwo	ork Assessment and Document: Essential	118

\_

5.1 Do	cument Your Network Inventory and Performance	119
5.1.1	Prerequisites	120
5.1.2	Create Network Intent	121
5.1.3	Use Intent Replication Wizard	123
5.1.4	Run the Intent	127
5.1.5	Export ADT to CSV file	129
5.1.6	Create Intent Dashboard for ADT Automation Column	130
5.2 AD	T Drilldown	133
5.2.1	Build Base ADT	134
5.2.2	Build ADT from a CSV File	141
5.2.3	Build ADT from Intent Template	144
5.3 Ass	sess and Document Your Network Operational Status	146
5.3.1	Create Network Intent	147
5.3.2	Add Signature Variables to ADT Base Table	154
5.3.3	Export ADT to CSV file	162
6 Netwo	ork Assessment Case Study: Security Assessment	
6.1 Che	eck NIST Compliance and Vulnerability	
6.1.1	Build ADT Base Table	
6.1.2	Create Network Intent: Check Vty Telnet Access	
6.1.3	Create NI: AAA Config Check	
6.1.4	Create NI: Device Unused Ports Config Check	
6.1.5	Create NI: Device Password Policy Config Check	193
6.1.6	Create NI: Device Line Session Timeout	195
6.1.7	Create Intent and Summary Dashboard	197
6.2 CVI	E Security Advisory	202
6.2.1	Prerequisites	203
6.2.2	Build Base ADT	205
6.2.3	Create an Intent to Check if a CVE is Affecting any Device	207
6.2.4	Create Intent to Check Device Against all CVEs	215
6.2.5	Create Intent and Summary Dashboard	216
7 Netwo	ork Assessment Case Study: Configuration Drift	217

7.1	Check Configuration Drift Against the Golden Template	218
7.1	7.1.1 Prerequisites	218
7.1	7.1.2 Create Intent to Check ACL Against Golden Template	
7.1	7.1.3 Modify Intent to Create CSV File	232
7.1	7.1.4 Use Intent Replication Wizard	238
7.1	7.1.5 Run Intent Once and Rebuild Table	241
7.1	7.1.6 View Summary Report	242
7.′	7.1.7 Create Intent and Summary Dashboard	243
7.2	Create Golden Template	244
7.2	7.2.1 Create Intent to Export ACL Configuration to a CSV File	244
7.2	7.2.2 Create ACL Report for the Whole Network Devices	248
7.2	7.2.3 Analyze the Report to Find the Golden Config	250
7.3	Check NTP Server Against Golden Template	256
7.3	7.3.1 NTP Server Golden Template	256
7.3	7.3.2 Create Intent to Check NTP Server Against Golden Template	256
		250
/.:	7.3.3 Create the Summary Report and Dashboard	259
7.4	Create the Summary Report and Dashboard Check Configuration Settings in Public Cloud	
7.4		261
7.4 7.4	Check Configuration Settings in Public Cloud	261
7.4 7.4 7.4	Check Configuration Settings in Public Cloud 7.4.1 Prerequisites	261 262 264
7.4 7.4 7.4 7.4	Check Configuration Settings in Public Cloud 7.4.1 Prerequisites 7.4.2 Create Intent to Check AWS EC2 Configuration	
7.4 7.4 7.4 7.4	Check Configuration Settings in Public Cloud7.4.1Prerequisites7.4.2Create Intent to Check AWS EC2 Configuration7.4.3Replicate Intent to All AWS EC2	
7.4 7.4 7.4 8 N 8.1	Check Configuration Settings in Public Cloud 7.4.1 Prerequisites 7.4.2 Create Intent to Check AWS EC2 Configuration 7.4.3 Replicate Intent to All AWS EC2 Network Assessment Case Study: Failover	
7.4 7.4 7.4 8 N 8.1 8.1	Check Configuration Settings in Public Cloud 7.4.1 Prerequisites 7.4.2 Create Intent to Check AWS EC2 Configuration 7.4.3 Replicate Intent to All AWS EC2 Network Assessment Case Study: Failover Check Failover Status Change	
7.4 7.4 7.4 8 N 8.1 8.1 8.7	Check Configuration Settings in Public Cloud 7.4.1 Prerequisites 7.4.2 Create Intent to Check AWS EC2 Configuration 7.4.3 Replicate Intent to All AWS EC2 Network Assessment Case Study: Failover Check Failover Status Change 8.1.1 Create HSRP Device Group	
7.4 7.4 7.4 8 N 8.1 8.1 8.7	Check Configuration Settings in Public Cloud 7.4.1 Prerequisites 7.4.2 Create Intent to Check AWS EC2 Configuration 7.4.3 Replicate Intent to All AWS EC2 7.4.3 Replicate Intent to All AWS EC2 Network Assessment Case Study: Failover Check Failover Status Change 8.1.1 Create HSRP Device Group 8.1.2 Create an intent to check the failover status change	
7.4 7.4 7.4 8 N 8.1 8.1 8.7	Check Configuration Settings in Public Cloud7.4.1Prerequisites7.4.2Create Intent to Check AWS EC2 Configuration7.4.3Replicate Intent to All AWS EC27.4.3Replicate Intent to All AWS EC2Network Assessment Case Study: FailoverCheck Failover Status Change8.1.1Create HSRP Device Group8.1.2Create an intent to check the failover status change8.1.3Replicate the intents for all HSRP devices using ADT	
7.4 7.2 7.2 8 N 8.1 8.1 8.2	Check Configuration Settings in Public Cloud         7.4.1       Prerequisites         7.4.2       Create Intent to Check AWS EC2 Configuration         7.4.3       Replicate Intent to All AWS EC2         7.4.3       Replicate Intent to All AWS EC2         Network Assessment Case Study: Failover         Check Failover Status Change         8.1.1       Create HSRP Device Group         8.1.2       Create an intent to check the failover status change         8.1.3       Replicate the intents for all HSRP devices using ADT         8.1.4       Create the Dashboard	
7.4 7.2 7.2 8 N 8.1 8.1 8.2	Check Configuration Settings in Public Cloud	
7.4 7.4 7.4 8 N 8.1 8.1 8.1 8.2 8.2 8.2 8.3	Check Configuration Settings in Public Cloud	

8.3.3	Replicate the intents for all HSRP devices using ADT	
8.3.4	Replicate the intent to all HSRP devices	
8.3.5	View and Run the intents in ADT	
8.3.6	Create the Dashboard	
9 Collabo	prative Troubleshooting	
9.1 Doc	ument BGP Devices and Configurations	
9.1.1	Create BGP Device Group	
9.1.2	Create an intent to parse BGP AS numbers and configuration	
9.1.3	Define the replicate settings and decode	
9.1.4	Build Base ADT	
9.2 Crea	ate a wrapper intent for BGP Troubleshooting	
9.2.1	Create an intent to Check BGP Configuration	
9.2.2	Create a Wrapper Intent	
9.2.3	Execute the Auto Intent BGP Wrapper	
9.3 Crea	ate frequently used BGP Troubleshooting Intents	
9.3.1	Create NI to Check BGP Neighbor Status Check	
9.3.2	Replicate the Intent to ADT	
9.3.3	Create NI to Check BGP Route Reflector Client	
9.3.4	Replicate the Intent to ADT	
9.4 Crea	ate a chatbot for the BGP TS wrapper intent	
9.4.1	Create the Netbrain Chatbot	
9.4.2	Use Chatbot	
10 Applica	ation Path Assessment and Troubleshooting	
10.1 De	ocument the application path when the network is healthy	
10.2 Bu	uild and Manage Automations for Path	
10.2.1	Create the ADT base table from the path browser	
10.2.2	Replicate the Intents of Device Health Check	
10.2.3	Create Path Intent for Path Next Hop Stability check	
10.2.4	Create Wrapper Intent and Enable Auto Intent	
10.3 Tr	oubleshoot the path issue	405
11 Intent-	based Change Verification and Assessment	406

11.1 B	uild Intents for Change Assessment	407
11.1.1	What has Changed	407
11.1.2	Auto Test	410
11.2 In	tent-Driven Change Management and Assessment	412
11.2.1	Define Change	412
11.2.2	Create Intent to Analyze Route Changes	413
11.2.3	Organize and Run Automations Before Change	420
11.2.4	Execute Change	
11.2.5	Verify Change	425
12 Map a	nd Document Your Network	427
12.1 C	reate a Map	427
12.1.1	Create Map with Intent	428
12.1.2	Intent Data View	433
12.2 Ci	reate the Inventory Report	438
12.2.1	Built-in Inventory Reports	438
12.2.2	Use ADT to Build Inventory Report	440
12.3 Ci	reate the Report from CLI Command Results	
12.3.1	Create CSV Report from CLI Command Results	
12.3.2	Create Report by Pre-Replicated Intent Template	

### **1** Introduction

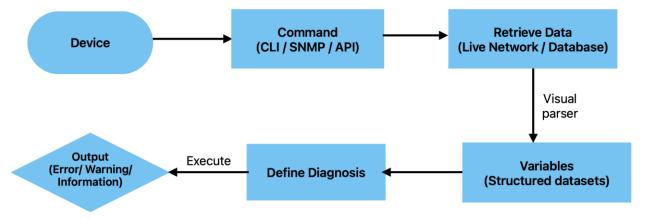
### 1.1 What is Intent-based Automation?

**Network Intent (NI)** is an automation construct that allows users to define expected design and operational state. It establishes a baseline configuration and operational state to validate the network design. Further, users can run NIs to simply detect design and operational deviations during the troubleshooting or set up the scheduled task to run a set of NIs to assess the whole network.

A standard NI has the following key components:

- **Device(s)**: An intent is specific to a device or a set of devices. The intent is designed this way so that it can include the current, last, and baseline data.
- **Raw data**: The data can be retrieved from devices by the CLI, SNMP, and API.
- **Parser**: The Visual Parser transforms the raw data into structured datasets. These datasets serve as the input for the diagnosis. Visual Parser is the UI-based for users without any programming language to create one.
- **Diagnosis**: The diagnosis defines rules or conditions to evaluate different aspects of the network and the status code (the output) for different conditions. Users can then specify the appropriate actions or recommendations to be executed based on the status codes.

So, an NI takes a device and a command (CLI/SNMP/API) as the input, retrieves data from the live network or database, parses the data into the structured datasets (variables), runs the diagnosis on these variables, and outputs the network status (error/warning/info).

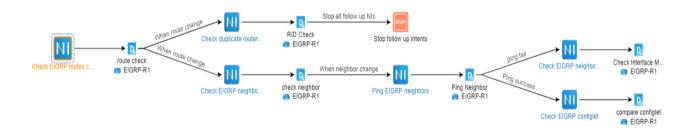


The following are some examples of NI:

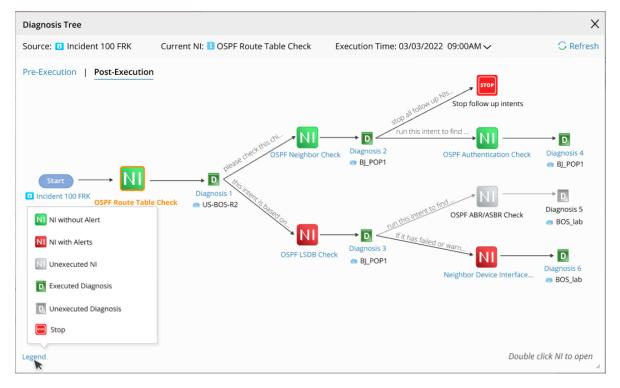
• Enforce configuration compliance and prevent configuration drift. For example, you can create an NI to ensure that an access list is configured in all Cisco firewalls.

- Continuously monitor the operational status. For example, you can create NIs to monitor the general health of a network device, such as the reachability, uptime and reboot reason, performance data, and temperature. You can also create NIs to monitor the specific vendor functions, such as the virtual server and pool member of an F5 load balancer.
- Check the failover. Create an intent to check whether the failover status changes and monitor the performance if a failover occurs.

Multiple NIs can be grouped to form a diagnosis flow. Users can define **follow-up NIs** in the Diagnosis so that these NIs can be executed under certain conditions. For example, users can add Check duplicate router ID and Check EIGRP neighbors as the follow-up NI if the EIGRP routes change. An Intent can also call itself as a follow-up intent (**follow-up self**), so the system will jump into other devices to execute the same defined logic as the original intent. For example, users can create an intent to check the BGP neighbor status and call itself for its BGP neighbors. This follow-up intent can be recursively called till no neighbor is found or it reaches a user-defined depth.



The execution results of a parent NI and its follow-up NIs are displayed in a diagnosis tree:



An intent (seed NI) can be cloned for all qualified devices through the whole network or as a set of network devices via an **Intent Replication Wizard**. An **Intent Template** (NIT) will be defined for this purpose, including the definition of the target devices, rules to replace the Macro Variables, etc. The cloned intents can be grouped into a column of an **Automation Data Table (ADT)**, which is the other key component (besides intent itself) for intent-based automation.

An ADT contains a **base table**, which can be the critical assets and be built from the device group, sites, application table, intent template, and CSV file, and many **Column groups**, which are associated with the intent and results and can be replicated primarily from the intent template.

The following is an example of an ADT table showing all BGP devices and intents associated with the BGP devices:

🐻 BGP Document and	d TS	Table Builder Lost Up	dated at: 07/08/2024 02:18 PM 🛛 🔍	Rebuild Table				Adr	d Data Manually 🗸 🚊 🧃
Description: Type des	cription here								
Items: 15 Rows 11 C	olumns							Search Q 🛛 Adva	nced Filter: Undefined
Cevice	BGP ASN	S Router ID	S BGP Configurations	💶 Check BGP Config Change 🛛 🛇	SIntent Status Code	S Device Status Code	TS BGP (Wrapper)	🖪 BGP neighbor Check 🛛 🛇	Check Reflector Cli-
UK-LHR-CR01-02	65200	192.168.10.253	router bgp 65200	Check BGP configure change UK 💿	BGP does not change	UK-LHR-CR01-02	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check UK 👁	BGP Route Reflector
UK-LHR-CR01-01	65200	192.168.10.254	router bgp 65200	Check BGP configure change UK 👁	BGP does not change	UK-LHR-CR01-01	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check UK 📀	BGP Route Reflector
US-NYJ-CR01-01	65101	192.168.1.254	router bgp 65101	Check BGP configure change US 📀	BGP does not change	US-NYJ-CR01-01	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check US 🐵	BGP Route Reflector
ISP-PE03	10000		router bgp 10000	Check BGP configure change IS	mp Group	ISP-PE03	Vrapper intent for BGP Tre	nn Group 2 <sup>ets</sup> .	BGP Route Reflector
ISP-P02	100 Base I	able	router bgp 10000	Check BGP configure change ISP	BGP does not change	U ISP-P02	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check IS 📀	BGP Route Reflector
US-BOS-CR01-01	65100	192.168.0.254	router bgp 65100	Check BGP configure change US 📀	BGP does not change	US-BOS-CR01-01	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check US 🐵	BGP Route Reflector
JP-TYO-CR01-01	65300	192.168.20.254	router bgp 65300	Check BGP configure change JP-T 📀	BGP does not change	JP-TYO-CR01-01	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check JP	BGP Route Reflector
US-BOS-R2	65001	10.11.11.11	router bgp 65001	Check BGP configure change US 👁	BGP does not change	US-BOS-R2	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check US 🐵	BGP Route Reflector
SG-SIN-CR01-01	65301	192.168.21.254	router bgp 65301	Check BGP configure change SG 👁	BGP does not change	SG-SIN-CR01-01	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check SG 🐵	BGP Route Reflector
ISP-PE01	10000		router bgp 10^^^	Charle DCD configures changes ICD	not change	ISP-PE01	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check IS 🐵	BGP Route Reflector
ISP-PE02	10000		router bgp 10	Intents/GD/TS and Document BGP/Check BGP	s not change	ISP-PE02	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check IS	BGP Route Reflector
DE-MUC-CR01-01	65201	192.168.11.254	router bgp 65	fange/Check BGP configure change JP+TYO+CR0	s not change	DE-MUC-CR01-01	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check DE	BGP Route Reflector
JP-TYO-CR01-02	65300		router bgp 65300	Check BGP configure change JP-T ⊙ ≡	BGP does not change	JP-TYO-CR01-02	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check JP 📀	BGP Route Reflector
US-BOS-R1	65001	10.10.10.10	router bgp 65001				Vrapper intent for BGP Troubles 👁		
US-BOS-CR01-02	65100		router bgp 65100	Check BGP configure change US 👁	BGP does not change	US-BOS-CR01-02	Vrapper intent for BGP Troubles 👁	BGP Neighbor Stability Check US 👁	BGP Route Reflector

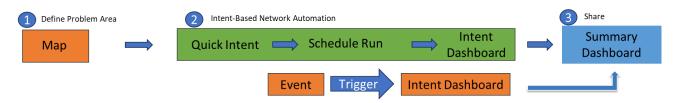
### **1.2 Intent-Based Automation Use Cases and Flows**

Intent-based automation can be applied for many use cases. In this book, we will cover three categories of use cases which can be automated:

- Collaborative troubleshooting.
- Continuous network assessment.
- Network change and assessment.
- Map and document the network.

### 1.2.1 Intent-Based Collaborative Troubleshooting

The collaborative troubleshooting workflow starts with mapping the problem area. Different levels of users can create intents via the **Quick Intent** tab within the map and run intents to troubleshoot the same problem. The active diagnosis results and the relevant triggered event results can be shared in a **Summary Dashboard**, which summarizes the results of groups of intents.



After mapping out the problem area, a user can leverage the Quick Intent tab inside the map to quickly define an intent for a map device by following three steps: collect the data, define the logic, and run the intent. Users can repeat these three steps till they are satisfied with the results. An intent can be scheduled to run (e.g., run every minute for 20 times) to troubleshoot a transient problem.

The final quick intent can then be replicated for all devices on the map. The cloned intent can be saved as **a map intent**, and the result can be displayed on the map or in the **Intent Dashboard**.

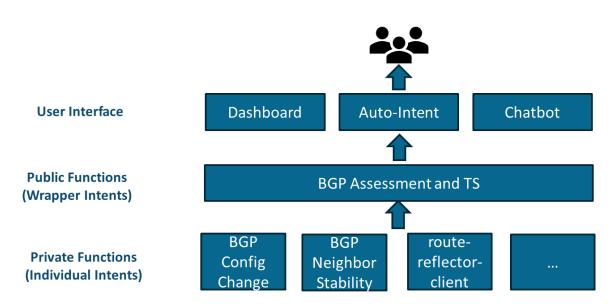
Troubleshooting Slow Application Description						μ.	Acto-refresh C	tati d <sup>a</sup> n et
Detection Device Summary	STINZULL, 1342/36 AM View Report	Intent Result Hist Time Targe: All V		• Sum of Intent Alert State	u Code Count 🔶 Sum of	Intent Success Status Code Count	8/19/2022 10:42:56.4	M View Report
6 Detected Alert Count		10	A/14/2015A, 10:15 A2 AM		6062003, 012133	59	* */16/2020, 10/2010 AM	
7 Detected Device Count		Intert Name	Execution Time	Intert Alert Statut Code Co	Top Fire Intent Alerts Intern Success Status Code .	. International Code Summary Internation	Intern Klern D	Recto
<		path path	8/16/3323, 10/95/7 AV 8/19/2323, 10/24/85 AV 8/19/2323, 10/24/85 AV	3	3	The staffic usage of liters	1	>
Device Information  © Exce/Aftrewal  © Gase 55 Series  © Exce Rater	SifferZock, 12-43/09 AM View Report	Detection Times 1	ummery				8/16/2023 10/4636-4	M Water Reports
	7 Device Name			3	Detected Aut			

Different users at different levels can create many intents to troubleshoot the same problem. The Intent Dashboards from these intents can be added to a Summary Dashboard. The final output of the collaborative workflow is the Summary Dashboard, with each row as a diagnosis activity and each column as a device.

Dashboard and Intent Group	Device Results				<u> </u>					
bashboard and ment droup	Total Device Results	Internet	Global-Core-WAN	USA-WAN	EMEA-WAN	Distribution-WAN	NY Data Center	London Data C	Core Network	
ServiceNow Trigger										
BGP Neighbor Stability Ch 🔗	55 8	2 2	7 2	3 1	4 1	23 1	6 1	3	7	
BGP Routes Stability Check	67 0	4	10	4	6	25	7	4	7	
IGP Injected to BGP Redistribu	40 0	1	5	3	3	14	5	3	6	
Level-1 NOC										
PIM Neighbor Stability Check	21 0	1	6	1	2	6	2	2	1	
Multicast Route Stability Check	15 0	1	4	1	2	3	1	2	1	
Level-2 Escalation										
Multicast RP Stability Check	21 0	1	6	1	2	6	2	2	1	
Multicast RPF Neighbor Stabili	15 0	1	4	1	2	3	1	2	1	

The individual intents can be grouped to troubleshoot a problem. One way to do this is to create a wrapper intent, which includes all relevant intents as the follow-up intents as a public interface for

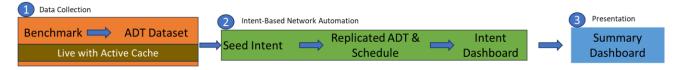
the end users so that any change of underlined intent will not change the end-user interfaces. For example, you can wrap all intents to troubleshoot the BGP-related issues, such as checking BGP config change, BGP neighbor stability, and route-reflector-client, into a wrapper intent, BGP Assessment and TS, which will be exposed to the end user interface, such as Dashboard, Autointent, and Chatbot.



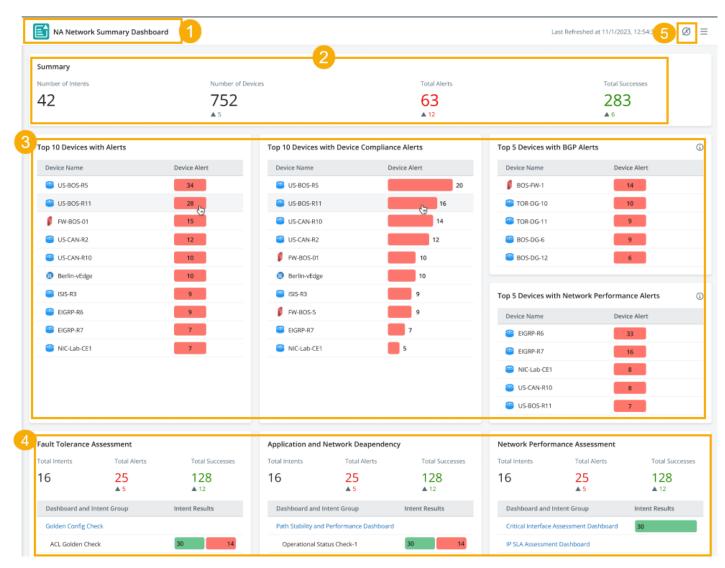
#### **1.2.2 Continuous Network Assessment**

The **Automation Data Table** (**ADT**) is a key component of the continuous network assessment across your whole network. You can build an **ADT Base Table** from the device group (such as all BGP devices) or from the intent template, which forms the scope for the network assessment. Then, replicate the intents to the ADT table as the **Column Group** to assess the network configuraitons and operational status.

In the continuous assessment workflow, the data is collected through the benchmark and saved in the ADT as the **Dataset**, which can be used to decode, replicate and run intents.



The results of the continuous assessment are presented in the Summary Dashboard. The dashboard displays the key metrics, the top devices with the most alerts, and the detail charts, with each row as an assessment point and each column as the alert counts for a device group or site.



In this book, we will study three important cases of network assessments:

- Security: check the configurations against the NIST compliance and find out the CVE vulnerability for network devices.
- Configuration Drift: check the configuration drift from the Golden template and the industry standard.
- Failover: check the configuration consistency between the failover pair, failover status, and the performance degradation after the failover.

#### 1.2.3 Change Management and Assessment

NetBrain **Change Management** module is a comprehensive solution for controlling a network change process. With intent-based automation, you can upgrade the change process with more accurate verifications of the change, for example,

	Net3rain <sup>Next-</sup>	Sen	SF	× 📀	,				Ø	Incident Searc	ch Incident Q	🗬 🌲	\rm e La	b Domain	$\bigcirc$
	ក្តែភ្វ Map63 (Master) ។	<b>/*</b> ≻ Pa	ige 1 🗸 🧧 Summary				📀 Overall Health Viev	v 🚦 Network	📑 Stencils	👬 Map 📔	e	+	100%	0 8 <	×
$\oplus$	😞 🚺 Inten	t	Runbook	D Data View										+ New Incide	ent
	Add a network subne	t (for Cool	kbook) 🗸	📋 Document	Auto Test Intent (Bef	ore Change)		t No	ote (0) 🗙				. <b>.</b>		
Recents	Select Action			=	Please input description	for this action.									
Setwork	2		Define Change		Items:4 + Add Inte	nt									
					Intent Name	Target Dev	Status Code	CSV Repor	Actions						
Files			Benchmark Before		Ping Check	1						•			
Site		Ŧ			Assess OSPF neighbo	rst 1									
			¥	=	Check CPU	1						ÓS-R2			
op Path	3		Auto Test Intent (Before		Cookbook Server Farr	n F 1	G					Router			
Dashboard	4	ø	Execute												
Intents					4				► F						
Ę	5		What is Changed Intent		Items:0 Filter Intent b	by Tags: type to	o select tags.	Filter			A.C.				
Chatbot			<b>\</b>		Intent Name Targe	et Dev Status (	Code CSV R	epor Diagnos	sis Tr		11.204/20 172.16.8	1.0/22 Et2/3	1		
) Data	6		Auto Test Intent (After C							EBRITRIT	•/		172.16.11.2	06/22	•
			Benchmark After							BOS-SW3				10.8	DS-SW2 8.1.242
			Ļ						1	0.8.1.243				CISCO IC	OS Switch
			Compare						50	o IOS Switch					
Мар			ě												
Desktop	Incident: 105XRX							Ru	in 🐵						

- 1. Run the auto-test before and after the change. You can add as many as intents to verify that the change does not violate the designs and that important operation status is normal. For example, you may want to run the batch ping for the key applications to ensure that they are still accessible.
- 2. Run the intent, **what is changed**, to verify that the change is successfully executed and there are no unexpected impacts. For example, for routing change, verify that no critical routes are removed.

The intent to discover **what is changed** can be useful not just for Change Management but also for troubleshooting and network assessments. You can create a dashboard to view what is changed across the whole network:

Last Refreshed at 4/14/2024, 2:01:18 PM 🛛 🧭 📃



Summary		Number of Devices		N	umber of Alerts			Number of Successes		
330		59			50			530		
What's Changed(Last Result)										
Number of Intents		Number of Devices		Ne	umber of Alerts			Number of Successes		
330	1	59		6	50			530		
▲ 209	4	11		*	60			▲ 530		Q
Dashboard and Intent Group	Intent Results	Device Results By Site								< >
Dashboard and Intent Group	intent Results	Total Device Results	JP-TYO	SG-SIN	DE-MUC	UK-LHR	US-NYJ	US-BOS	ISP	
AAA Config Change	42	42 0	6	5	5	10	6	10		
ACL Config Change	84	84 0	12	10	10	20	12	20		
BGP Config Change	10	10 0	1	1	1	2	1	1	3	
Device Uptime Change		0 0								
EIGRP Config Change	4	4 4 4	2 2	2	2					
HSRP Config Change	8	8 0	2	1		2	2	2		
OSPF Config Change	9	2 9 2			2 1	3 1		4		
QoS Config Change	24	0 24	4	4	4	4	4	4		
Redistribute Config Change	7	7 0	1	1	1	1	1	1	1	
SNMP Config Change	44	44 0	5	5	5	10	6	10	3	
Software Version Change	15	15 0	2	2	2	2	2	2	3	
Static Route Config Change	42	42 0	6	5	5	10	6	10		

### 1.3 How to Use This Cookbook?

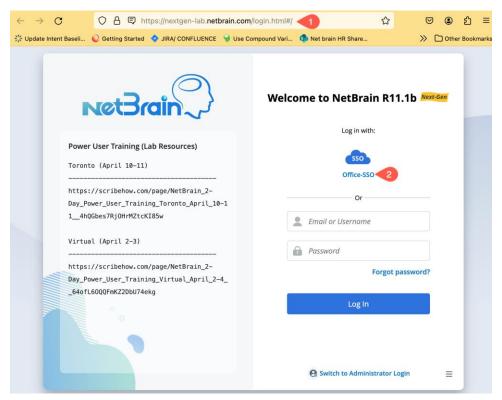
In this book, you will learn the key concepts of NetBrain **Intent-Based Automation** with real-world use cases. The book is organized into two parts:

- Chapters 2 to 5 walk you through the basics of intent-based automation, and we
  recommend you read all of these materials and follow the examples. In these chapters, you
  will learn how to automate the most frequently used commands, such as commands to
  probe the device and interface health. It covers the basic workflow of intent-based
  automation: create an intent and ADT, replicate it to auto intent and ADT, and create the
  dashboard and chatbot for end users.
- In chapters 6 to 12, you will apply the intent-based automation to different use cases. You can skip any of these chapters and jump to the use cases you are interested.
  - Chapters 6 to 8 study three cases of network assessments: security, configuration drift, and failover.
  - Chapter 9 covers collaboration troubleshooting, and chapter 10 teaches how to troubleshoot the path-related issues.
  - Chapter 11 studies how intent-based automation can be used to assess the impact of network change.
  - Finally, chapter 12 teaches how the intent can be used to create the map and documentation.

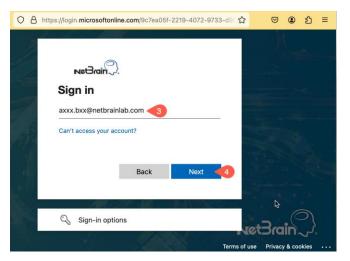
The best way for you to learn is to follow the examples step by step. You can practice using your own NetBrain system. Or, You can use the free NetBrian lab. Start by signing up for an account as detailed in the Section <u>Sign-in Netbrain Lab</u>.

#### 1.3.1 Sign in to Netbrain Lab for the First Time

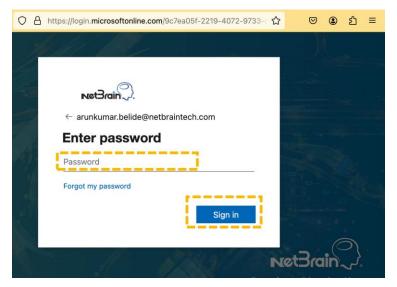
- 1. Navigate to <u>nextgen-lab.netbrain.com/desktop.html</u> in your browser.
- 2. Click Office-SSO to log in with your @netbrainlab.com account.



- 3. Enter the username. The username provided by NetBrain ends with **@netbrainlab.com**. You can reach out to NetBrain support if you do not have one.
- 4. Click Next.



5. Enter the provided **Password** and then click the **Sign in** button.



6. To set up the MFA (multi-factor authentication), click the Next button.

Net Brain	
john.doe@netbrainlal	b.com
More inform	ation required
Your organization nee	eds more information to keep
Use a different accou	unt
Learn more	Next

7. Click I want to set up a different method to set up SMS-based MFA.

Microso	oft Authenticator
6	Start by getting the app
	On your phone, install the Microsoft Authenticator app. Download now
	After you install the Microsoft Authenticator app on your device, choose "Next".
	I want to use a different authenticator app
	Next
<u>l want to set up</u>	a different method

8. If you want to use an Authenticator app, click **Choose a method >> Phone** >> **Confirm.** 

6	Start by getting the app								
	On your phone, install the Microsoft Authenticator app. Download now								
	After you install the Microsoft Authenticator app on your device, choose "Next".								
	l want to use a different authenticator app								
<u>l want to set up</u>	a different method × Which method would you like to use?								

9. Enter your mobile phone number and then click **Next**.

Phone
You can prove who you are by texting a code to your phone. What phone number would you like to use?
United States (+1)
Text me a code     Message and data rates may apply. Choosing Next means that you agree to the Terms of service and Privacy
and cookies statement.
I want to set up a different method

10. Click the **Enter code** field and enter the code sent to your mobile number >> **Next.** 

Keep your account secure	ving who you are
Your organization requires you to set up the following methods of prov	ving who you are.
Phone	
We just sent a 6 digit code to +1 . Enter the code below.	
454253	$\frown$
Resend code	
	Back Next
want to set up a different method	
<u></u>	

11. Click **Done**.

		p your accour		g who you are.
		up your security info. Choos	se "Done" to continu	e signing in.
S	Phone +1 7058225146			Done

Next, you will need to change your password. Enter the provided password and a new password, and click **Sign In**.

NetBrain	
john.doe@netbrainla	ab.com
Update you	r password
	your password because this is e signing in, or because your ed.
Confirm password	
	Sign in

12. On the first login, select your Tenant **demo\_tenant >> Lab domain** >> **Apply.** 

nanti demo_t	enant 🗸			Search	Q	😋 Refresh
Tenant Name	Domain Name	Foundation Nodes	Description		Cre	ator
demo_tenant	demo_domain	50000 (49980 available)			adm	min 🖂
demo_tenant	ab domain	1000 (980 available)			zun	iyu.liu 🔽

You have now successfully setup MFA and logged into the NetBrain lab environment.

PART 1 - Intent Based Automation Essential

## 2 Automate Frequently Used Commands

In this chapter, you will create your first intent. You will find that it is easy to automate the frequently used CLI commands such as *Ping <destIP>* and *show ip route <destIP>*. No programming knowledge is required. We will use two examples:

CLI Command	Intent Description
Ping < destination ip>	Check whether the success rate is equal to 100% and the average round trip time is reasonable. If not, create an alert.
show ip route <destination ip=""></destination>	Compare the next hop to the last known value (baseline). If it changes, create an alert and set the baseline value.

The key steps to define an intent are:

- 1. Select a device, enter the CLI command, and retrieve the data.
- 2. <u>Define variables</u>: from the CLI command results, find the data you are interested in and define it as the variable.
- 3. <u>Define the Diagnosis:</u> compare the variable with the normal status or design and create an alert.

This chapter also covers a common intent-based automation creation flow, which has the following steps:

- 4. Create a new map or open an existing map.
- 5. Under the **Quick Intent** tag, create an intent. You can edit and run the quick intent recursively till you are satisfied with the intent results.
- 6. Save the intent as a map intent.
- 7. Create the dashboard from the intent to better view the results.

### 2.1 Search and Add Devices to the Map

The intent is always associated with a device or a set of devices, which are called the **seed device(s)**. You can draw these devices either by searching devices from the search bar and adding them to the map or by drawing the predefined group of devices to the map (recommended best practice).

#### 2.1.1 Draw Devices from a Device Group

You can create a device group and draw the predefined group of devices to the map as follows:

- 1. Go to **Device Group** from the **start** menu <sup>III</sup> from the Netbrain desktop.
- 2. In the device group pane, select the predefined group **Cisco Routers** and click the drop-down menu.
- 3. Click **Draw Devices on Map** to add the devices to a new map.
- 4. Close the Device group pane.

‱t3rain								
Search App	Q					🍓 Domain Ma	anagement	
Network Map-Based Aut	omation Intent-B	ased Automat	tion Incider	nt & Change	Misc			
letwork Analysis	Discovery			Data Mo	odel			
🔅 Network	Ø Discover	O Discover		🔯 Dat	🔝 Data Accuracy Discovery			
🖳 One-IP Table	🎲 Network	🍪 Network Settings			Oata Accuracy Resolution			
🚡 Device Group	🔜 API Serv	🔜 API Server Manager		Open Driver Manager				
🚴 Site	🔰 🛛 💿 Do-Not-	S Do-Not-Scan			E Health Report			
歳 Site Manager	Net3rain Next-Gen	Search Anyt	thing and Create Map	٩	<b>~</b>			
Inventory Report	Device Group > My Device	Groups > Cisco F	Routers			😮 Help 🗧	G # X	
Change Analysis Report	Search	٩	Search				Q	
Recents	My Device Groups		Hostname	Vendor	Model	Mgmt IP		
-	Cisco Routers (4)		2 CA-TOR-R1	Cisco	CGS-MGS-AGS	10.8.3.1		
Network	SI Move		😋 CA-TOR-R2	Cisco	CGS-MGS-AGS			
Files	Delete		US-BOS-R1	Cisco	CGS-MGS-AGS			
	Open Group Ma		😁 US-BOS-R2	Cisco	CGS-MGS-AGS	10.8.1.240		
Site	Draw Devices of							
<u>@</u> 2	Draw Group on	Мар						
بر Path	WAN Cloud (7)							

#### 2.1.2 Draw Devices from the Search Bar

You can search and map devices from the Netbrain desktop:

Search the device **US-BOS-R1** in the search bar.

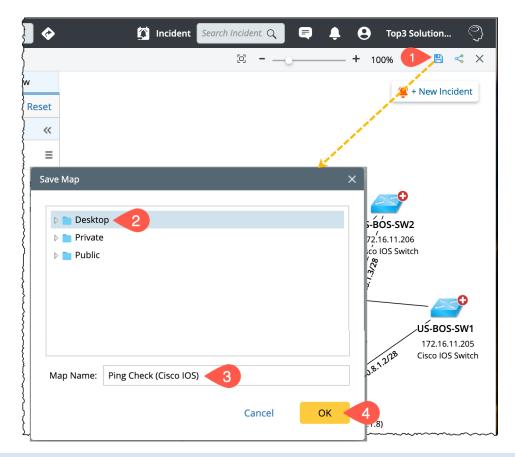
- 1. In the results device section, click **Map** to draw the device to the map.
- 2. Close the search results window and open the **Intent** pane.
- 3. Repeat the Step 1 thru Step 3 and add other cisco devices US-BOS-R2 and CA-TOR-R1.

	NetBrain Next-Gen us-bos-sw1 1 × 📀	
÷	Search Results(103)	≡ ₹ ×
	✓ □ Device (1/1)	
Recents	US-BOS-SW1 Site: test-site-jun	Map 2
Network	18 Interfaces 9 L3 Neighbors 7 L2 Neighbors Cisco 3560E Software Version: 15.2(HL_201702	
	Configuration 1 - <mark>US-BOS-SW1</mark> #show run	
Files	14 - hostname <mark>US-BOS-SW1</mark>	
Site	<ul> <li>Site (1/1)</li> </ul>	
<u></u>	Device Group (20/61)	

#### 2.1.3 Save the Device Map

In the Upper-Right corner of the screen, click the floppy disk icon to save the current map.

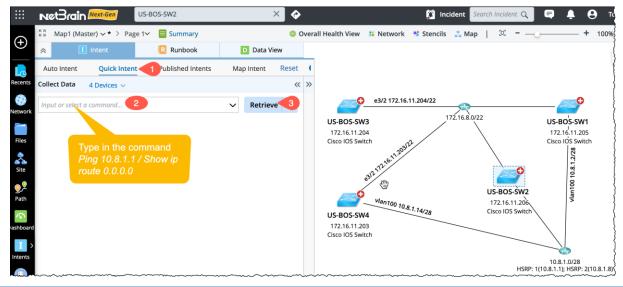
- 1. In the **Save Map** window, choose either a Desktop or another location to save the map.
- 2. Add a name [Ping Check (Cisco IOS) / Route Check (Cisco IOS)] to the map.
- 3. Click **OK** to save and close the window.



### 2.2 Quick Intent Creation

From the map you just created,

- 1. Go to the **Quick Intent** tab in the Intent section.
- 2. Enter the command *Ping 10.8.1.1* in *Input or Select a command...* field to collect data. For another example, you enter the command *show ip route 0.0.0.0*.
- 3. Click **Retrieve** to parse the data.



4. The retrieved data sample will be:

CLI	Retrieve data Sample							
	요금 Map32 (Master) ~ * > Page 1~ 📃 Summary							
	Runbook D Data View							
	Auto Intent Quick Intent Published Intents Map Intent							
	Collect Data 3 Devices V							
5.	Ping 10.8.1.1 X V Retrieve							
Ping <destination< td=""><td>12:15:14 PM: Successfully retrieved!</td></destination<>	12:15:14 PM: Successfully retrieved!							
ip>	Select parser ∨ New ∨							
	Add to Intent >>							
	06/21/2024 12:15:09 PM (Live) US-B0S-R1#Ping 10.8.1.1 Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 10.8.1.1, timeout is 2 seconds: !!!!! Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms							
	Ka Map32 (Master) マ★ > Page 1マ							
	Runbook D Data View							
	Auto Intent Quick Intent Published Intents Map Intent							
	Collect Data 3 Devices V Collect Data 3 Device							
	show ip route 0.0.0.0 × V Retrieve							
show ip route	12:16:01 PM: Successfully retrieved!							
<destination ip&gt;</destination 	Image: Select parser       ✓       New       ✓							
	Add to Intent >> Pleas							
	06/21/2024 12:15:58 PM (Live) 🎪							
	<pre>US-BOS-R1#show ip route 0.0.0.0 Routing entry for 0.0.0.0/0, supernet Known via "static", distance 1, metric 0, candidate default path Routing Descriptor Blocks:</pre>							

### 2.3 Define Variables with a Visual Parser

Defining variables is simple with **Quick Intent** using the auto-parser feature. Entry-level users can quickly learn and utilize the parser. The system will automatically select the appropriate parser mode (**single** or **multiple**) based on input words.

Once the auto-parser completes the parser definition task, you can add additional variables or make adjustments to the **line pattern** to achieve the desired result.

#### 2.3.1 Define parser for ping <destination ip>

You can define a variable with the following steps:

1. Create and add a parser using the option **New** located next to the **Parser** field.

K 3	Map32 (Ma	aster) 🗸 * 🗦	Page 1∨	Summary				
~	I	Intent		Runbook		Data View	/	
Auto Intent Quick Intent Published Intents Map Intent								}
Coll	ect Data	3 Devices $ \smallsetminus $					«	Define Logic
Pin	g 10.8.1.1				$\times \checkmark$	Retrieve	0	
12:1	12:15:14 PM: Successfully retrieved!							
63	US-BOS-R1	~	Parser:	Select parser	1	New	~	l
					Ado	d to Intent >	<b>&gt;&gt;</b>	Diagon
06/	06/21/2024 12:15:09 PM (Live) 🐥							
Typ Ser	pe escape nding 5, 1 !!!	-	o abort. MP Echos	to 10.8.1.1, t				

2. Select the text **100** in the sentence **rate is 100 percent** and click **Parse Variable** in the tip window. You can also double-click the text to get the same result.

Select	Double-click a variable to parse. Select multiple lines to parse a table. Critical Variable (0) No Pattern + New Pa Current Device  Search Search US-BOS-R1#ping 10.8.1.1 Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 10.8.1.1, tim HILLII Success rate is 100 percent (5/5), round-trip m Double click a variable fi		
	Format1 V +		Qualific
i Do	uble-click a variable to parse. Select multiple lines to parse a table.	Critical Variable (0)	No Pattern $\checkmark$ + New Patter
Cur	rrent Device V Search	Q 🔺 🔻	
2 3	Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 10.8.1.1, tim		
5 6			Double click a variable fiel

3. Click Apply.

ping 10.8.1.1	X 🗸 on 🥔 U	IS-BOS-SW1 Retrieve vith Live Data
		Qualification: Defined Test on Devices: 0
} ct multiple lines to parse a table. {	Critical Variable (0)	Name: Pattern1 Type: O Single O Multiple Can 3 Apply =
Search	Q 🔺 🔻	VarLine 1 rate is \$int:percent percent
p abort. MP Echos to 10.8.1.1, tim		5 Success rate is 100 percent (5/5), round-trip m > 1 Line
rcent (5/5), round-trip m	Var Line 1	

- 4. Click **+Field** to another variable for round trip time.
- 5. Similar to Step 2, select the round-trip text value 1/1/2.

o parse a table.	Critical Variable (0)	Image: Pattern1   ✓   ✓   Type: Single ?   + New Pattern ✓	
Search seconds :	Q • •	Var Line 1 rate is \$int:percent percent 5 Success rate is 100 percent (5/5), round-trip m > 1 Line	=
x = 1/1/1 ms Parse \	Var Line 1 /ariable 5	+ Field 4	

6. Modify the Variable line pattern to min/avg/max = \$int:min/\$int:avg/\$int:max ms.

NOTE: The variable name is defined as \$<type>:<variable\_name>. The default variable type is string, and the type mstring is a string with spaces.

7. Click Apply.

Cancel Apply 7
ent percent = percent (5/5), round-trip m > 1 Line
<pre>t:min/\$int:avg/\$int:max ms 6 percent (5/5), round-trip mi &gt; 1 Line</pre>
ł

8. Select **OK** to save and close the window.

Output	+ Parse Lines	<u>*</u>	-
\$percent (in	t) = 100		
\$min (int)	= 1		
\$avg (int)	= 1		
\$max (int)	= 1	Variables	
		parsed from the sample data	
			Can 8 OK
			-

9. Click **Add to Intent**, and the parsed variables are added to the diagnosis.

Map2 (Master) V* > Page 1V	Summary				📀 Overall Health Viev	v 🚦 Network	📑 Stencils
< 🚺 Intent	Runbook	D Data View	/				
Auto Intent Quick Intent	Published Intents	Map Intent				Reset	🕑 Help
Collect Data 4 Devices V			~	Define Logic			~~
ping 10.8.1.1		×∽ Retrieve	<b>#</b>				Create
11:28:39 PM: Successfully retrieved!				🥏 US-BOS-SW1	@ ping 10.8.1.1	1 Diagnosi	s 🗸
✓ US-BOS-SW1 ∨ Parser:	Parser1	1 ~ Edit	~				
	9	Add to Intent >	»>				
07/19/2024 11:28:10 PM (Live) 🧛	Success	fully parsed all 4 varia	ables.				
US-BOS-SW1#ping 10.8.1.1				Diagnosis: Diagnos	sis 1 🗸	+ New	Diagnosis
Type escape sequence to abort Sending 5, 100-byte ICMP Echo !!!!! Success rate is 100 percent (	s to 10.8.1.1, ti				Anchor: S	elect Variable	~

#### 2.3.2 Define parser for Command Show ip route

You can define a variable with the following steps:

1. Create and add a parser using the option **New** located next to the **Parser** field.

K 2	Route Che	ck (Cisc (Maste	er) 🗸 >	Page 1∨	📒 Sumr	mary			
~	1	Intent		<u> R</u> unbo	ook		D Data View		
Au	ito Intent	Quick Inten	t	Published	Intents	Ма	p Intent		
Coll	ect Data	2 Devices 🗸						~	Define
sho	ow ip route	0.0.0.0				×v	Retrieve	69	
03:0	4:19 PM: Su	ccessfully retriev	ved!						
	US-BOS-SW	/1 ∨ F	Parser:	Select pa	rser	1	New	~	
						Ado	d to Intent >	·>	
07/	20/2024 03	:03:18 PM (Live)	dia.						
US-	-BOS-SW1>	show ip route	0.0.0	).0		~ ~~~ ~		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	

- 2. Select the text **Routing Entry For 0.0.0/0**, and click **Parse Variable** in the tip window. You can also double-click the text to get the same result.
- 3. Modify the Variable line pattern to Routing entry for \$subnet,.
- 4. Similar to Step 2, copy the next hop line **10.8.1.18** from the sample data and parse the variable.

w Parser			
Select Data: CLI Command v show ip route 0.0.0.0	Qualification: Defined       Test on Devices: 0       =         ultiple lines to parse a table.       Critical Variable (0)       Name: Pattern1       Type: Image: Single Image: Multiple Cancel Image: Apply Image: A		
Format1 V +		Qualification: Defined Test on Devices: (	0 ≡
① Double-click a variable to parse. Select multiple lines to parse a table.	Critical Variable (0)	Name: Pattern1 Type: • Single O Multiple Cancel Apply	≡
US-B05-SW2>show ip route 0.0.0.0 Routing entry for 0.0.0.0/0, supernet Known via "ospf 1", distance 110, metric 1, Tag 1, type extern 2, forward metric 21 Last update from 10.8.1.18 on Vlan101, 4w1d Routing Descriptor Blocks: 10.8.1.18, from 10.10.10, 4w1d ago, via		VarLine 1 Routing entry for \$var1, 3	

5. Modify the Variable line pattern to **^\$mstring:nexthop**, **from**.

NOTE: The variable name is defined as \$<type>:<variable\_name>. The default variable type is string, and the type mstring is a string with spaces. The created variables will appear under the Parsed Result section.

6. Click Apply.

7. Select **OK** to save and close the window.

New Parser		×
Select Data: CLI Command v show ip route 0.0.0.0	× ✓ on	etrieve v with Live Data
Format1 V +		Qualification: Defined Test on Devices: 0
Double-click a variable to parse. Select multiple lines to parse a table.	itical Variable (0) Name: Pattern	1 Type: O Single O Multiple Cancel Apply 6
Current Device     Search       1     US-B0S-SW2>show ip route 0.0.0.0       2     Routing entry for 0.0.0.0(0.1) supernet		Routing entry for <pre>\$string:subnet,</pre>
3 Known via "ospf 1", distance 110, metric 1, 4 Tag 1, type extern 2, forward metric 21 5 Last update from 10.8.1.18 on Vlan101, 4w1d		2 Routing entry for 0.0.0.0/0, supernet > 1 Line
7 10.8.1.18, from 10.10.10.10, 4w1d ago, via 8 Route metric is 1, traffic share count i 9 Route tag 1	ar Line 2 Var Line 2	<ul> <li>^ \$mstring:nexthop, from 5</li> <li>7 10.8.1.18, from 10.10.10.10, 4w1d ago, via V &gt; 1 Line</li> </ul>
11 Route metric is 1, traffic share count i 12 Route tag 1 13		
		+ Parse Lines —
	anexerep (insen	
Select Data: CLI Command v show ip route 0.0.0 v on use US-BOS-SW2 Retrieve v with Live Data Format1 v + Qualification: Defined Test on Devices: 0 9 Double-click a variable to parse. Select multiple lines to parse a table. Critical Variable (0) Current Device v Search v Search v I Line 1 1 US-BOS-SW2-Show ip route 0.0.0.0 Routing entry for 0.0.0.00 supernet v I Line 1 3 Known via "050f1", distance 110, metric 1, 4 Tog 1, type extern 2, forward metric 21 5 Lost update from 10.8.1.18 on Vlan101, 4w1d 6 Routing Descriptor Blocks: 7 JU.8.1.18, from 10.10.10, 10, 10, 10, 4w1d ago, via 8 Route metric is 1, traffic share count i 9 Route tag 1 10 * 10.8.1.2, from 10.10.10, 4w1d ago, via 11 Route tag 1 12 Route tag 1 13 Route tag 1		
		Сал 7 ОК

8. Click **Add to Intent**, and the parsed variables are added to the diagnosis.

🖏 Route Check (Cisc (Master) 🗸 > Page 1 🗸 🧧 Summary	😂 Overall Health View 💠 Network 📑 Stencils 🔮 Map
Runbook D Data View	
Auto Intent Quick Intent Published Intents Map Intent	Reset 🛛 Help
Collect Data 2 Devices ~ 《	Define Logic «
show ip route 0.0.0.0 X 🗸 Retrieve	Create
03:04:19 PM: Successfully retrieved!	US-BOS-SW2 Is show ip route 0.0 P1 IDiagnosis
US-BOS-SW2 v     Parser:     Parser1     P1     Edit     v       Add to Intent >>	
07/20/2024 03:03:18 PM (Live) 🔅 Successfully parsed all 2 variables.	
JS-BOS-SW2>show ip route 0.0.0.0 Routing entry for 0.0.0.0/0, supernet Known via "ospf 1", distance 110, metric 1, candidate default path Tag 1, type extern 2, forward metric 21 Last update from 10.8.1.18 on Vlan101, 4w1d ago Routing Descriptor Blocks: 10.8.1.18, from 10.10.10.10, 4w1d ago, via Vlan101 Route metric is 1, traffic share count is 1 Route tag 1 * 10.8.1.2, from 10.10.10, 4w1d ago, via Vlan100 Route metric is 1, traffic share count is 1 Route tag 1	Diagnosis:       Diagnosis 1        + New Diagnosis         Name:       Diagnosis 1       Anchor:       Select Variable          Type description of the diagnosis

### 2.4 Define the Diagnosis

Go to the **Define Logic** section to define the diagnosis logic as follows:

- 1. Enter the diagnosis name, e.g., **Ping Cisco** or **Subnet next hop check**.
- 2. Define the condition as detailed in the following table and images:

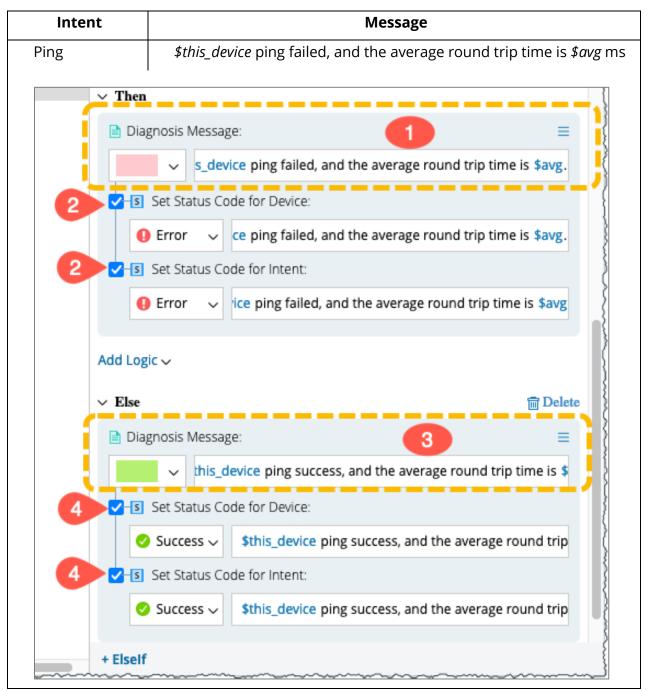
Intent		Steps
	a.	Variable percent does not equal 100.
a. Variable percent does not equal 100.         Ping < destination ip>         b. Variable avg is Greater than 60.         c. Boolean Expression: A or B.         Ping Cisco (Master) >* > Page 1~ Summary         Intent       Intent         Runbook       Data View         Intent       Quick Intent         Published Intents       Map Intent         Reset       Image: Cisco (Master) >* Parser:         Ping 10.8.1.1       X ~ Retrieve (a)         05/BOS-R1 ~       Parser:         Ping [Cisc       @ J         Add to Intent>       Osersfully parsed all 4 variables.		
	c.	Boolean Expression: <b>A or B.</b>
	View	
	eve 👳	
		-
US-BOS-R1#Ping 10.8.1.1 Type escape sequence to abort. Sending 5, 100-byte ICMP Echos to 10.8.1.1, timeout is 2 se !!!!!	conds :	Diagnosis: D Ping Cisco V + New Diagnosis
Success rate is 100 percent (5/5), round-trip min/avg/max =	1/1/2 m	S Name: Ping Cisco 2 Anchor: \$percent ~
		Type description of the diagnosis
	-	Loop Table Rows
	Í	
		A ⊜ US-B Current ∨
		a percent v Does not v 100 v
		B 😁 US-B Current 🗸
		b avg v Greater th v 60 v 🗑
		C Select Variable 🗸
		Boolean Expression: A or B
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	

Intent		Steps				
		a. The	/ariable <b>s</b>	<b>ubnet</b> Is not e	empty.	
how ip route <desti< td=""><td colspan="6">b. Variable <i>nexthop</i> (Current) Does not eq <i>nexthop</i> (Baseline).</td></desti<>	b. Variable <i>nexthop</i> (Current) Does not eq <i>nexthop</i> (Baseline).					
	c. Bool	ean Expr	ession: <b>A and</b>	В.		
Add Note D Add	Diagnosis		Can als	so click a variable on	the left to add au	itomati
Name: Subnet nexthop	check 🕗			Anchor:		~
Type description of	of the diagnosis					
Loop Table Rows						
A 👩 CA-TOR-R1	Current 🗸					
a subnet	~	ls not empty	~			Ē
B 👩 CA-TOR-R1	Current 🗸				Baseline 🗸	
b nexthop	~	Does not equal	~	nexthop	~	Ē
C Select Variable	~					

### 2.5 Define Intent Output

Enter a message under the **Then** and **Else** output areas to appear as the result of the diagnosis.

- 1. **Then**: Define a color, message and status in case **If** condition is true, as shown in the figure.
- 2. Click the checkbox of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.



Intent	Message				
Show ip route	On <i>\$this_device</i> subnet \$subnet nexthop <i>\$nextHop</i> is changed ( <i>\$nextHop(Baseline)</i> ).				

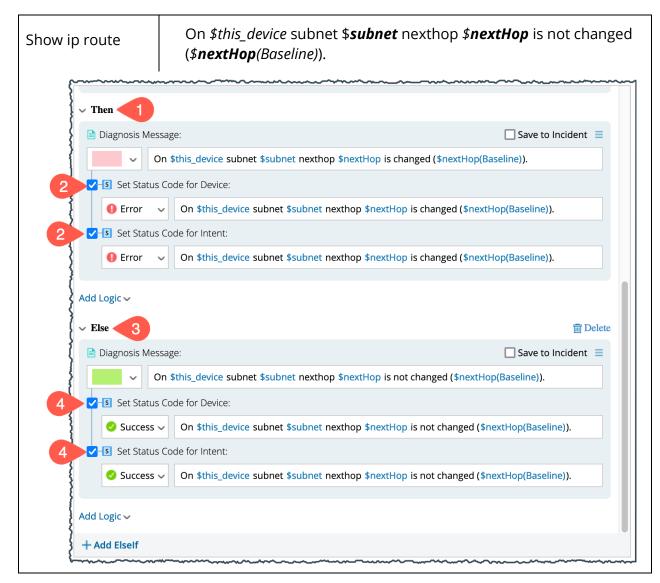
#### NOTE:

- By typing \$, you can get the variable selection pop-up.
- For a variable, you can get the current value and baseline value.

Define Link to Variable	×					
▲ → Paragraph1						
<b>i</b> subnet (US-BOS-R1.Paragraph1)						
Tenexthop (US-BOS-R1.Paragraph1)						
🕅 this_device						
Source: Current V						
t Current						
Baseline 2 Variable						
Last selected	rom					
baseline						
Link Text: \$nexthop(Baseline)						
Ca	ncel OK 3					

- 3. **Else**: Define a color, message and status in case **If** condition is not true, as shown in the figure.
- 4. Click the checkbox of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.

Intent	Message				
Ping	\$ <b>this_device</b> ping succeeded, and the average round trip time is \$ <b>avg</b>				



5. Click on **Create** to save and create the intent.

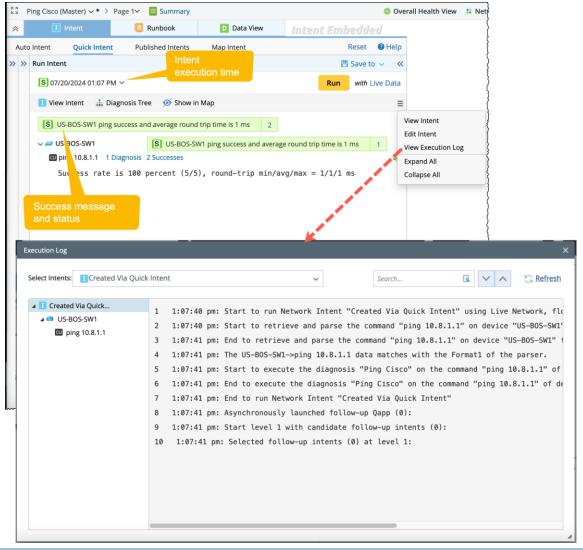
< 🚺 Intent	Runbook	D Data View		/		
Auto Intent Quick Intent	Published Intents	Мар	o Intent			Reset 🛛 🕄 Hel
Collect Data US-BOS-R1 V				~	Define Logic	
ping 10.8.1.1		$\times$	Retrieve			5 Create
11:55:11 AM: Successfully retrieved!					OS-BOS-R1 III show ip route 0.0	P2 D 1 Diagnosis
OS-BOS-R1 ∨ Parse	er: 🔜 Show ip ro 🦻	2 ~	Edit	~		
		Add	to Intent >	<b>&gt;&gt;</b>		
08/06/2024 11:15:26 AM (Live) 🎄	Succes	sfully pa	rsed all 1 vai	riable.		
<pre>JS-BOS-R1#show ip route 0.0.0.0 Routing entry for 0.0.0.0/0, supernet Known via "static", distance 1, metric 0, candidate default path</pre>			Diagnosis: Diagnosis 1 ~	+ New Diagnosi		

6. Go to the **Run Intent** pane and click **Run** to execute the diagnosis.

D Data View						
ap Intent					Reset 🛛 🕄	Help
{	~	Run Intent		9 	Save to 🗸	~ «
{	Recreate	Not Executed		6 Run	with Live	Data
1 Diagnosis	V	I View Intent	嚞 Diagnosis Tree	<ul> <li>Show in Map</li> </ul>		≡
<pre>{     /////////////////////////////////</pre>		✓ ┛ US-BOS-SV আ ping 10.8. Success		-cent (5/5), rou	und-trip	∰ mi…
}						

Upon completing the diagnosis by the system, the following result will appear:

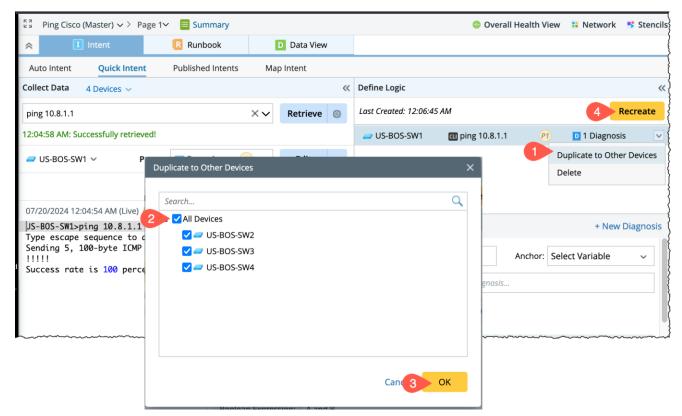
- Intent execution date and time.
- Success/Failure message and status as defined in **Then** and **Else** conditions.
- The execution log can be accessed using the **View Execution Log** located under **E**:



# 2.6 Duplicate the Intent to Other Devices

An intent can be duplicated to other devices. The system will automatically copy the parser and logic to other devices.

- 1. You can duplicate the created intent to other devices as follows:
- 2. Go to the **Define Logic** pane, and from the drop-down menu, select the option **Duplicate to Other Devices**.
- 3. Select all the devices to which you want to copy the intent.
- 4. Click **OK** to save.



- 5. All the selected devices will be added to the list and click **Recreate**.
- 6. In the **Run Intent** pane, all the devices will be listed with intent configured.
- 7. Click **Run** to execute the intent diagnosis on all the devices. And the results are displayed.

		Reset 🛛 Help
Define Logic	~	Run Intent 💾 Save to 🗸 🔍
Last Created: 12:14:51 AM	4 Recreate	Not Executed 6 Run with Live Data
🚽 US-BOS-SW1 🛛 🔃 ping 10.8.1.1 🦻	D 1 Diagnosis	Ⅰ View Intent 🔒 Diagnosis Tree ④ Show in Map 🛛 ≡
🚽 📨 US-BOS-SW2 🛛 💷 ping 10.8.1.1 🛛 😰	D 1 Diagnosis	√ 💋 US-BOS-SW1
	1 Diagnosis	💷 ping 10.8.1.1 1 Diagnosis
🚽 US-BOS-SW4 🛛 🔃 🔃 🔐 ping 10.8.1.1	1 Diagnosis	Success rate is 100 percent (5/5), roun
Diagnosis: D Ping Cisco 🗸	+ New Diagnosis	✓
Name: Ping Cisco Anchor: Se	elect Variable 🗸 🗸	Success rate is 100 percent (5/5), roun
Type description of the diagnosis		V 🚅 US-BOS-SW3
Loop Table Rows		💷 ping 10.8.1.1 1 Diagnosis 🎄
- If		Success rate is 100 percent (5/5), roun…
A 🥔 US-B Current 🗸		✓ 🚄 US-BOS-SW4
percent v Does not v 10	00 v 🗊	💷 ping 10.8.1.1 1 Diagnosis 🗛
B		Success rate is 100 percent (5/5), roun…

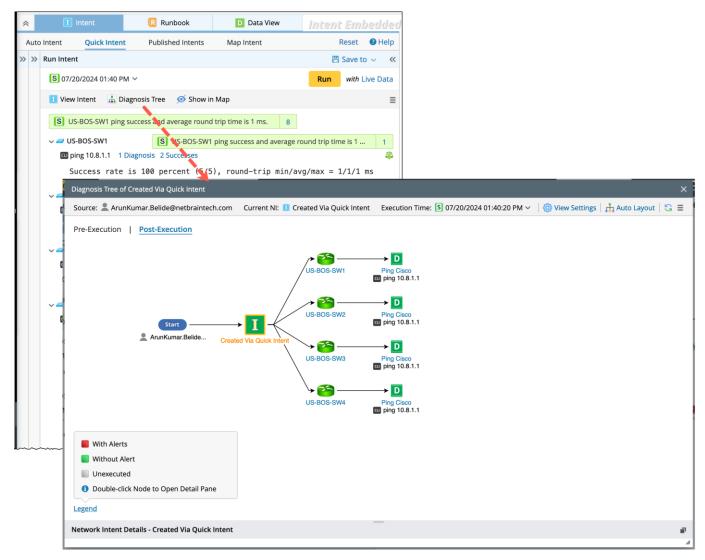
8. In the **Run Intent** pane, click **Save to > Save to Map Intent** and then **OK.** 

		Reset 🛛 Help
«	Run Intent	💾 Save to 🗸 🔍
Recreate	S 07/20/2024 12:17 AM	Save to Map Intent ata
		Save to Path Intent
1 Diagnosis	I View Intent 🚠 Diagnosis	Save to Common Intent
1 Diagnosis	S US-BOS-SW1 ping success	and average round tri 8
1 Diagnosis	V 🚄 US-BOS-SW1 S US-BOS	S-SW1 ping success an 1
1 Diagnosis		is 2 Successes
+ New Diagnosis	Success rate is 10	0 percent (5/5), roun…

# 2.7 Diagnosis Tree

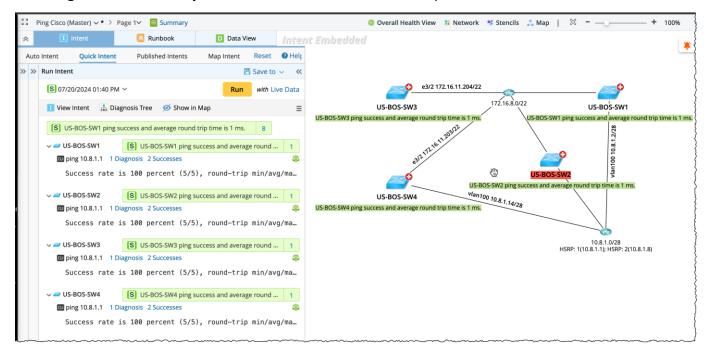
Click 📠 to view the diagnosis tree. It includes the following details as shown in the figure:

- Current NI medium,
- Execution date and time,
- The diagnosis tree in Pre-Execution and Post-Execution mode,
- The status of execution in each device with color legends.



# 2.8 View the Result on Map

The result of the intent execution can be visualized in a map as shown in the following map by selecting **Show in Map** button located in the **Run Intent** pane:



# 2.9 Create Intent and Summary Dashboard

You can view the intent results through two different dashboards:

- Intent Dashboard to view the individual Intent results and
- **Summary Dashboard** to view the consolidated Intent results in a single view.

#### 2.9.1 Intent Dashboard

The Intent Dashboard observes specific network issues with details and displays the results. You can save the frequently used dashboards as templates. Intent Dashboard can be created directly from the **Map Intent** tab as follows:

- 1. Go to the **Map Intent** tab and click on **New Dashboard.**
- 2. In the **Create Intent Dashboard** window, define the following:
  - a. Enter the Dashboard Name, Ping Latency Check
  - b. Select the Location to save the Intent Dashboard.
  - c. **Data Source:** By default, **Specified Intent** is selected from the dropdown.
  - d. Intent: Keep the default intent to create the dashboard for the same intent.
  - e. Click **Create**.

Ping Cisco (Master) V * > Page 1V Summary	🙁 Overall Hea
Runbook D Data View	Intent Embedded
Auto Intent Quick Intent Published Intents Map Intent	
S 07/20/2024 01:46 PM ~ Run ~ wit	Create Intent Dashboard
🚺 View Intent 🚠 Diagnosis Tree 🚿 Show in 🚺 🖼 New Dashboard	Create Intent Dashboard for 'Ping Cisco'
S US-BOS-SW1 ping success and average round trip time is 1	Name: Ping Latency Check
US-BOS-SW1 S US-BOS-SW1 ping success and average round trifts m ping 10.8.1.1 1 Diagnosis 2 Successes	Location: My Dashboards
Success rate is 100 percent (5/5), round-trip min/avg/	max Use Template
S US-BOS-SW2 ping success and average round trip t.	
ping 10.8.1.1 1 Diagnosis 2 Successes	Data Source Specified Intent
Success rate is 100 percent (5/5), round-trip min/avg/	
✓	Intent: Ping Cisco
Image ping 10.8.1.1 1 Diagnosis 2 Successes	✓ Include Triggered Follow-up Intent Results
Success rate is 100 percent (5/5), round-trip min/avg/	max
S US-BOS-SW4 ping success and average round trip t.	•
ping 10.8.1.1 1 Diagnosis 2 Successes	
Success rate is 100 percent (5/5), round-trip min/avg/	max
	Time Range Last 7 Days ~
	Cancel Create e

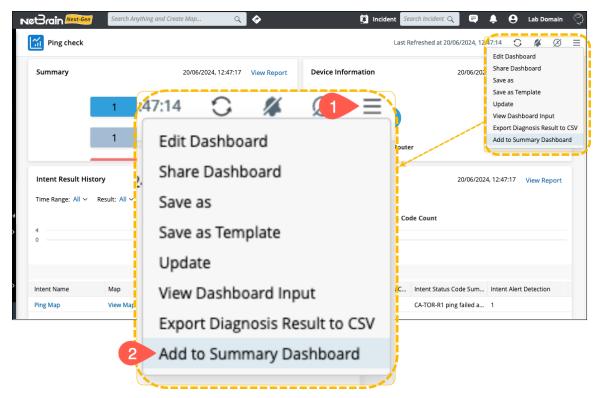
3. **Create Intent Dashboard** dialog pops up. Click **Open Intent Dashboard**, and the dashboard will be created.

ate Intent Dashboard		×					
	board has been created	d. rd, or choose to add it to a					
summary da	ashboard.						
Ado	d to Summary Dashb	Open Intent Dashboard					
Ping Latency C	heck			L	ast Refreshed at 20/7/2024, 2:04	:28 pm 🖸	<b>%</b> Ø
Summary		20/7/2024, 2:04:32 pm	View Report Devi	ice Information	20/7/20	24, 2:04:32 pm	View Report
	1	ntents	- I.				
	1 T	imes Executed				4	Devices
	0 1	ntent-level Alerts		Cisco IOS Swit	ch		
Intent Result Hist	ory				20/7/20	24, 2:04:32 pm	View Report
Time Range: All ~	Result: All 🗸	🗢 Sum of Intent Alert Status	s Code Count 🛛 🔶 Sum	of Intent Success Status Code Co	unt		
0			•				
			20/7/2024, 1:46:2	7 pm			
			Top Five Intent Aler	**			
			Top Five Intent Aler	LS			
Intent Name	Мар	Execution Time		Cou Intent Success Status Code	Intent Status Code Summary	Intent Alert De	tection

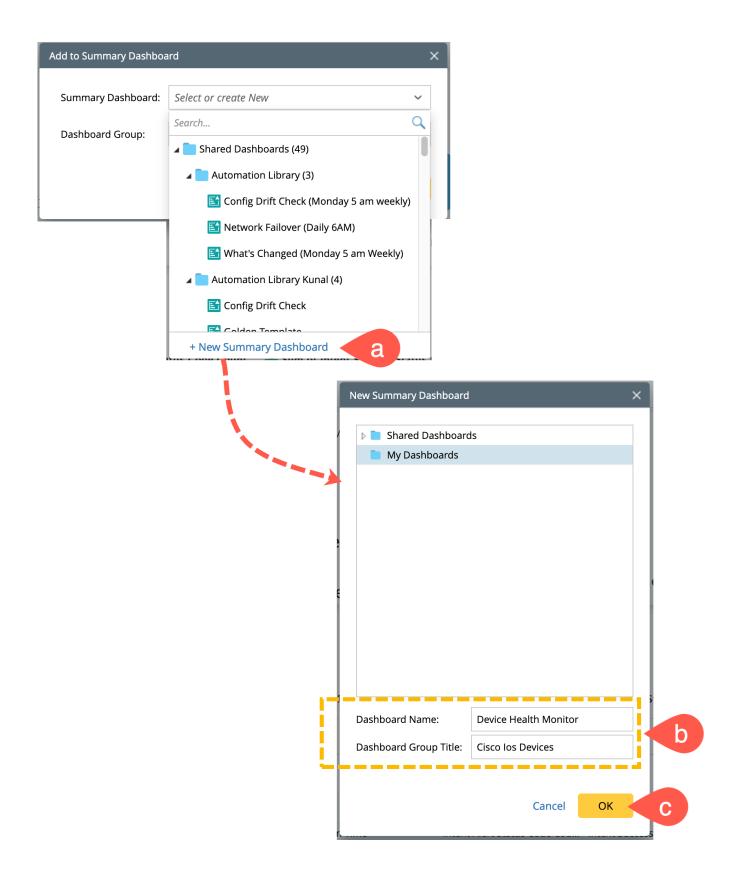
## 2.9.2 Summary Dashboard

The summary dashboard provides an overview displaying results from multiple Intent dashboards of the entire network or a set of network devices. With Summary Dashboard, you can group Intent Dashboards into widgets based on diagnosis purpose and display results by device, site or device groups. You can use the summary dashboard to monitor critical information across thousands of devices and discover the root cause for issues in one view.

- 1. Let us create a Summary Dashboard using the step-by-step instructions as follows:
- 2. Click  $\equiv$  to open the menu located at the top-right corner of the dashboard window.
- 3. Select **Add to Summary Dashboard** to open the corresponding window for creating a summary dashboard.

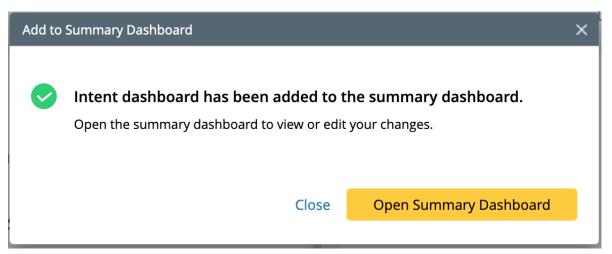


- 4. In the **Add to Summary Dashboard** window, let us create the new summary dashboard and group as follows:
  - a. **Summary Dashboard**: open the dropdown menu and select **+New Summary Dashboard** to pop up its dialogue.
  - b. Enter the dashboard basic details like **name**, **group title** and **location** of the summary dashboard to save.
  - c. Click **OK** to save and create the summary dashboard.



You can add more intent dashboards to the same summary dashboard by choosing this summary dashboard located under My dashboards locationAdd to Summary Dashboard dropdown menu.

5. In the **Add to Summary Dashboard** dialog, a success message prompt appears along with the option to **Open Summary Dashboard**. Click to view the dashboard.



6. Review the resulting Summary Dashboard and explore the dashboard interface.

Device Health Monitor				La	st Refreshed at 20/7/202	24, 2:28:37 pm	Ø
ummary							
lumber of Intents	Number of Devices	Number o	of Alerts	Nu	mber of Successes		
1	4	0		8			
isco los Devices							
lumber of Intents	Number of Devices	Number o	of Alerts	Nu	mber of Successes		
1	4	0		8			
Dashbaard and Intent Group	Intent Results	Device Results				<	
Dashboard and Intent Group	intent Results	Total Device Results	US-BOS-SW2	US-BOS-SW4	US-BOS-SW1	US-BOS-SW3	
Ping Latency Check	4	4 0	1	1	1	1	
Total Alert Count $\downarrow$			0	0	0		

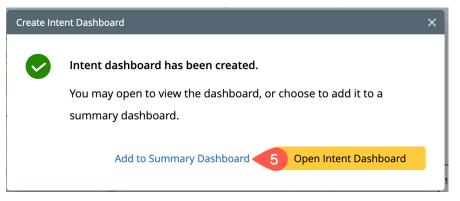
## 2.9.3 Add an Intent Dashboard to an Existing Summary Dashboard

Now, let us add the route check intent dashboard to the **Device Health Monitor** Summary Dashboard created in Section 2.9.2 as follows:

- 1. Go to the Map Intent tab and click New Dashboard to launch the Create Intent Dashboard window.
- 2. Change the default dashboard name to **Route check**.
- 3. Keep the default settings for Location (My Dashboards) and other fields
- 4. Click **Create**.

នឹង Route Check (Cisc (Master) 🗸 * > Page 1 🗸 🧧 Summary	© Ov
Runbook	Data View Intent Embedded
Auto Intent Quick Intent Published Intents Map Inte	ent
S 07/20/2024 03:53 PM 🗸	Create Intent Dashboard X
💶 View Intent 🔚 Diagnosis Tree 💋 Show i 🚺 🕍 New Dashboar	rd
S On US-BOS-SW2 subnet 0.0.0.0/0 nexthop 10.8.1.18 is not chang	Create Intern Dashboard for 'Route Check (Cisco IOS)'
✓	0.0.000 net Name: Route Check 2
Routing entry for 0.0.0.0/0, supernet 10.8.1.18, from 10.10.10.10, 4w1d ago, via	Location: My Dashboards
	Use Template
✓ ZUS-BOS-SW1 S On US-BOS-SW1 subnet 0.1 I biagnosis 2 Successes	0.0.0/0 ne:
Routing entry for 0.0.0/0, supernet	Data Source Specified Intent ~
* 10.8.1.33, from 10.10.10.10, 4w1d ago, via	a Ethern Intent: Route Check (Cisco IOS)
V = US-BOS-SW3 Subnet 0.	0.0.0/0 ne;
show ip route 0.0.0.0 1 Diagnosis 2 Successes	
Routing entry for 0.0.0.0/0, supernet	
✓	0.0.0/0 nex
d show ip route 0.0.0.0 1 Diagnosis 2 Successes	
Routing entry for 0.0.0.0/0, supernet * 10.8.1.2, from 10.10.10.10, 4w1d ago, via	Vlan100
	Time Range Last 7 Days ~
	Cancel Create 4

5. In the **Create Intent Dashboard** dialog, a success message prompt appears along with Click **Add to Summary Dashboard to open its corresponding dialog**.



6. In the **Add to Summary Dashboard** window, click the **Summary Dashboard** dropdown menu to open and select the **Device Health Monitor** that you have created in Section 2.9.2.

Add to Summary Dashboa	ard	×
Summary Dashboard:	Select or create New	~
Dashboard Group:	Search	Q
Dashboard Group.	Shared Dashboards (49)	
	🖌 📄 My Dashboards (2)	
	🛃 Device Health Checks	2
এত	🗈 Device Health Monitor	6
n/avg/max = 1	+ New Summary Dashboard	Device Health Monito

7. Choose the dashboard group that you have created in the previous section or create a new group as per the need.

Add to Summary Dashboard			
Summary Dashboard:	E Device Health Monitor	~	
Dashboard Group:	Cisco los Devices	~	
	Search	Q	
	All Dashboard Groups (1)	2	
	Cisco los Devices		
n/avq/max = 1	+ New Dashboard Group	is 1	

- 8. Click **OK** to save and create the summary dashboard.
- 9. Click **Open Summary Dashboard** to view the dashboard.

Add to Summary Dashboa	ırd	×
Summary Dashboard:	E Device Health Monitor ~	]
Dashboard Group:	Cisco los Devices ~	]
	Can 8 OK	
Add to Summary	Dashboard	×
	dashboard has been added to the summary dashboard. e summary dashboard to view or edit your changes.	
	Cl 9 Open Summary Dashboard	d

10. Review the resulting Summary Dashboard and explore the dashboard interface.

Device Health Monitor					Last Refreshed at 20	/7/2024, 4:27:31 pm	Ø	5
Summary								
Number of Intents	Number of Devices	Nun	nber of Alerts		Number of Successes			
1	4	0			16			
Cisco los Devices								
Number of Intents	Number of Devices	Nun	nber of Alerts		Number of Successes			
1	4	0			16			
Route	Check						C	С
added	to the	Device Results					<	>
dashbo		Total Device Results	US-BOS-SW2	US-BOS-SW4	US-BOS-SW1	US-BOS-SW3		
Ping Latency Check		4 0	1	1	1	1		
Route Check	4	4 0	1	1	1	1		
Total Alert Count $\downarrow$			0	0	0	0		

# **3** Automate Basic Device Health Checking

While troubleshooting a network-related problem, the first step is always to check the overall health of the corresponding network devices and their interfaces/ports. Instead of manually checking, you can automate the frequently used device and interface health checks. The following table lists some basic common health checks:

Intent Name	Description
Uptime check (Cisco IOS)	Check whether the device reboots within the last 24 hours (the uptime is shorter than a day). If so, then provide the reload reason. Use the CLI command <i>show</i> <i>version</i> .
CPU usage check (Cisco IOS)	Check whether the device CPU utilization is normal or higher using the CLI command show process CPU.
Interface status check (Cisco IOS)	Check the interface status and also whether errors (Input, Output, CRC errors) are increasing using the CLI command <i>show interface</i> .

NOTE: It is recommended to name the intent as **<purpose of the intent><Device type>**.

Besides the overall health check, you may also automate some important operational status. For example, for an F5 load balancer, you may create an intent to check the status and state of the virtual servers and their pool member using the CLI command *show ltm virtual detail recursive*.

Due to the limitations in Quick Intent, we shall use the standard intent editor and create the Intents for Cisco devices. There are three main steps to define an intent:

- 1. Select a device, define the basic information of intent and save the intent to Intent Manager.
- 2. Parse the variables from retrieved command data.
- 3. Define the diagnosis logic and output.

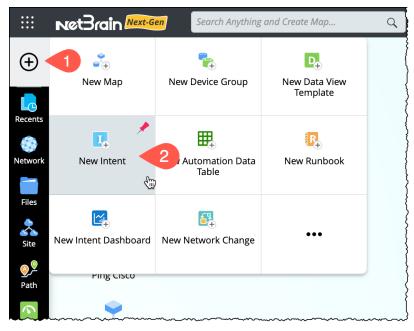
After executing the intent, you can create a dashboard to show its results:

- 4. Execute the intent and create a dashboard.
- 5. Create an intent dashboard.
- 6. Add Intent dashboards to a common Summary Dashboard.

# 3.1 Select a Device

Define the basic information of the intent, such as name, description, and device and save it to the intent manager as follows:

- 1. Click 🕑 icon from your desktop.
- 2. Select **New Intent.** A window **Network Intent** in edit mode will appear.



- 3. Enter the title and a brief description (optional) of the intent.
- 4. Add the seed devices by selecting the option **+Device**.

Network intent (Edit Mode)	×
1 Type intent title here 2 3 agnosis Tree	Run with Live Data Save 🛛 Help 🚍
Type description here	💑 Intent Map: Select 🗸
Seed Logic     Greek Control Cont	
■ + Device 4	Intent Variables: Manager   Tag: + Add 🔳
	4

- 5. Click **Save** to save the intent to the Intent Manager.
- 6. Click **+Add CLI Diagnosis** to open the corresponding window and proceed to the next section to parse the variables and define the diagnosis.

Network Intent (Edit Mode	)	×
I Check uptime (Cisco I	Diagnosis Tree	Run with Live Data 5 Save @ Help =
Type description here		🍣 Intent Map: Select 🗸 🗸
I Seed Logic	🔟 Replication Logic 🛕	
🐸 + Device		Intent Variables: Manager Tag: + Add 🗐
v 🥏 US-BOS-SW1	Type Description here	🗟 + Add Config Diagno 6 💷 + Add CLI Diagnosis 🚍
No content has been added	d.	
otwork (24)		Å

# **3.2 Retrieve Data and Parse Variables for Health Checks**

After choosing the device as described in <u>Section 3.1</u>, parse the variables from the retrieved CLI command data in the following manner:

- 1. Parse variables from Show Version
- 2. Parse variables from Show Process CPU
- 3. <u>Parse variables from Show Interface</u>
- 3.2.1 Parse variables from show version
- 1. In the command field, enter the command *show version*.
- 2. Click Retrieve.
- 3. In the sample data, select the uptime value *3 weeks, 4 days, 11 hours, 15 minutes* and click **Parse Variable** in the tip window.

NOTE: You can also double-click the text to get the same result.

CLI Comm	and Diagnosis
<i>շ</i> US-	BOS-S 1 show version 2 Retrieve vith Live Data
1. Defin	e Variable 2. Define Diagno
	Format1 V +
i Doul	ble-click a variable to parse. Select multiple lines to parse a table. Critical Variable (0) No Pattern
Curr	ent Device V 07/16/2024 12:05:03 PM Search Q • •
1 2 3 4 5 6 7 8 9 10 11 12 13	US-BOS-SW1>show version Cisco IOS Software, Linux Software (I86BI_LINUXL2-ADV Technical Support: http://www.cisco.com/techsupport Copyright (c) 1986-2017 by Cisco Systems, Inc. Compiled Thu 02-Feb-17 03:38 by mmen ROM: Bootstrap program is Linux US-BOS-SW1 upt 3 30 weeks, 3 hours, 15 minutes System returned to ROM by reload at 0 System image file is "unix:/opt/unetlab/addons/iol Last reload reason: Unknown reason

4. In the right pane **Var Line 1** field, update the variable name to *\$mstring:uptime*.

× e Data 2. Define Diagnosis Test on Devices: 0 ≡ Name: Pattern1 Type: O Single O Multiple Cancel ariable (0) Apply  $\equiv$  $\otimes$ Var Line 1 uptime is \$mstring:var1 US-BOS-SW1 uptime is 31 weeks, 2 days, 11 hours, 47 minu... > 1 Lin 9 Var Line 1 uptime is \$mstring:uptime 4 9 US-BOS-SW1 uptime is 31 weeks, 2 days, 11 hours, 47 minu... > 1 Line . + Parse Lines Output \$var1 (mstring) = 31 weeks, 2 days, 11 hours, 47 minutes

NOTE: The variable name is defined as *\$<type>:<variable\_name>*.

5. Similarly, parse the reload reason and update the variable to *\$mstring:reload\_reason*.

I. Define Variable	2. Define Diagnosis
Format1 V +	Test on Devices: 0
Double-click a variable to parse. Select multiple lines to parse a table.	le (0) 🖸 Pattern 1 🗸 Z Type: Single 👔 + New Pattern 🗸 🚍
Current Device     07/14/2024 10:25:31 PM     Search     Q       1     JJS-BOS-SW1>show version     2     Cisco IOS Software, Linux Software (I86BI_LINUXL2-ADV)       3     Technical Support: http://www.cisco.com/techsupport       4     Copyright (c) 1986-2017 by Cisco Systems, Inc.       5     Compiled Thu 02-Feb-27 03:38 by nmerk	Var Line 1 uptime is \$mstring:uptime 9 US-BOS-SW1 uptime is 29 weeks, 5 days, 13 hours, 35 minu > 1 Line
ROM: Bootstrap program is Linux US-BOS-SW1 uptime is 29 weeks, 5 days, 13 hours, 35 m System image inclusion of the constant o	VarLine 2 reason: \$mstring:reload_reason
Last reload reason: Unknown reason	Output         + Parse Lines         -         -           Suptime (mstring)         -         29 weeks. 5 days. 13 hours. 35 minutes         -

6. Parsed variables will be listed under the **Output** section in the right pane. Validate the variables.

7.	Click Apply	to save and	go to <b>Defin</b>	e Diagnosis	Section ?	331.
<i>'</i> •	Check mpphj				<u>Section s</u>	<u></u> .

2	2. Define Diagnosis Test on Devices	
6	☑ Pattern1 ✓ Ž Type: Single ② + New Pattern ✓	5. U E
	Var Line 1 uptime is \$mstring:uptime 9 US-BOS-SW1 uptime is 29 weeks, 5 days, 13 hours, 35 minu > 1 Line	=
	Var Line 2 reason: \$mstring:reload_reason 12 Last reload reason: Unknown reason > 1 Line	=
	+ Field	
	Output + Parse Lines	-
	<pre>\$uptime (mstring) = 29 weeks, 5 days, 13 hours, 35 minutes \$reload_reason (mstring) = Unknown reason</pre>	

### 3.2.2 Parse variables from *Show Process CPU*

- 1. In the command field, enter the command *Show Process CPU*.
- 2. Click **Retrieve** and the sample data is displayed under the **Define Variable** section.
- 3. In the sample data, from the CPU utilization line (you can search the word *CPU utilization* to find the line), select the *five minutes* value *0*% and click **Parse Variable** in the tip window.

NOTE: You can also double-click the text to get the same result.

4. In the right pane **Var Line 1** field, update the variable name to *\$int:cpu*.

NOTE: The created variables will appear under the **Output** section. The variable name is defined as *\$<type>:<variable\_name>*.

5. Click **Apply** to save and go to **Define Diagnosis** <u>Section 3.3.2.</u>

					2. Define Diagnosis	
Format1	<b>∨</b> +					Test on Device
Double-click a variab	le to parse. Select	multiple lines to pars	se a table.	Critical Variable (0)	Name: Pattern1	Type: O Single O Multiple Cancel Apply
3 uSecs 4 0 5 36 6 0	one minute: 55ec 1Mir 0.00% 0.00% 0.00% 0.00% 0.00% 0.00% 2. Define Dia Name: Pa	6 0.00% 0 Chu 6 0.00% 0 Loo 6 0.00% 0 Spo gnosis ttern1 e1 five minut	ts: 0k ccess Parse Varia ink M ind Meter inTree Flush Type: ⊙ Sing es: \$int:cpu		Test on Devices: 0 =	es: \$minutes Jutilization for five seconds: 0%/0%; one minute: 0%; fiv[ > 1 Line
			<b>A</b>		}	

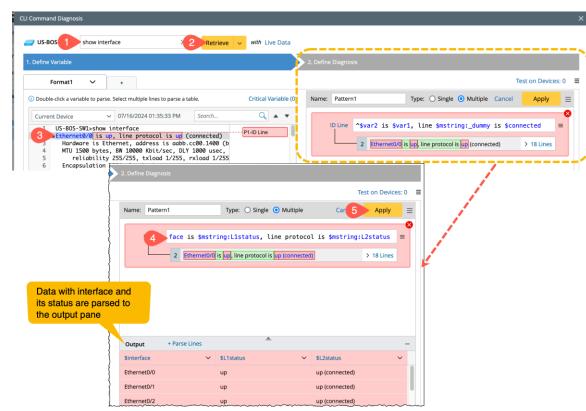
## 3.2.3 Parse variables from Show Interface

Unlike the CLI command outputs in earlier sections, the output of the **show interface** includes multiple interfaces, and each of them has an identical format. For this type of text format, you will use **Paragraph Parser** to convert multiple text sections of an identical pattern into a table variable:

- 1. In the command field, enter the command *show interface*.
- 2. Click Retrieve.
- 3. In the sample data, select the line *Ethernet0/0 is up, line protocol is up* and click **Parse Variable** from the tip window. This line will be defined as the ID line, which identifies the starting line of a paragraph. The ID line is a normal line pattern where you can define the variables and keywords.
- 4. In the right pane **Var Line 1** field, modify the line to be: *\$string:interface* is *\$mstring:L1status*, line protocol is *\$mstring:L2status*.

NOTE: The variable names are defined as \$<type>:<variable\_name>.

NOTE: All the lines in source data with interface and its status will be highlighted automatically and reflected in the **Output** pane.



5. Click Apply.

6. Similarly, parse other variables (input errors, CRC and output errors) by referring to the following table:

Variables		Parsing Procedure
Input_errors	a.	Select the input error value <b>0</b> and click Parse Variable in the tip window.
	b.	Modify the Variable to <i>\$int:input_errors.</i>
	с.	Select the CRC value 0 and click Parse Variable in the tip window.
CRC errors	d.	Modify the Variable line to <i>\$int:CRC</i> .
Output_errors	e.	Select the output error value 0 and click Parse Variable in the tip window.
	f.	Modify the Variable to <i>\$int:output_errors</i> .

7. Click **Apply** to save and go to **Define Diagnosis** <u>Section 3.3.3</u>.

1. Define Variable 2. Define Diagnosis									
Format1 V +	Test on Devices: 0								
Double-click a variable to parse. Select multiple lines to parse a table.     Critical Variable	e (0) 👍 Pattern 1 🗸 🖌 Type: Paragraph 👔 🕴 + New Pattern 🗸 🚍								
Current Device07/14/2024 10:38:22 PMSearchQ1810077236 packets input, 867625071 bytes, 0 no b19Received 9311774 broadcasts (0 multicasts)200 input errors, 0 CRC, 0 f220 input packets with dribb2379422477 packets output, 6240 output errors, 0 collisi	<pre>ID LineA \$interface is \$mstring:L1status, line protocol is \$mstr = 2 Ethernet0/0 is up, line protocol is up (connected) &gt; 18 Lines Var Line 1 ^ \$int:input_errors input errors, \$int:CRC CRC, = 21 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ign &gt; 18 Lines</pre>								
25 <b>0 unknown protocol drops</b> 31 MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec, 32 reliability 255/255, txload 1/255, rxload 1/255 33 Encapsulation ARPA, loopback not set 34 Keepalive set (10 sec) 35 Full-duplex, Auto-speed, media type is RJ45 36 input flow-control is off, output flow-control is 37 ARP type: ARPA, ARP Timeout 04:00:00 38 Last input never, output 00:00:00, output hang nev 45 Last clearing of "show interface" counters never	Var Line 2  \$int:output_errors output errors 24 output errors, 0 collisions, 0 interface resets > 18 Lines Output +Parse Lines								
<ul> <li>Input queue: 0/2000/0/0 (size/max/drops/flushes);</li> <li>Queueing strategy: fifo</li> <li>Output queue: 0/0 (size/max)</li> <li>5 minute input rate 0 bits/sec, 0 packets/sec</li> <li>5 minute output rate 3000 bits/sec, 3 packets/sec</li> <li>0 packets input, 0 bytes, 0 no buffer</li> </ul>	Sinterface IP ×       \$L1status ×       \$L2Status ×       \$input_errors ×       \$CRC         Ethernet0/0       up       up (connected)       0       0       Image: the status input_errors input_error input_erro								
Help   All Intent Variables	7 Apply								

# **3.3 Define Diagnosis for Health Checks**

Let's define the logical condition and diagnosis output using the parsed variables for all the intents as follows:

#### 3.3.1 Define Diagnosis for Show version

- 1. Go to the **Define Diagnosis** ribbon and click on **Add Diagnosis**.
- 2. Enter the diagnosis name as **Check Uptime**.
- 3. **Anchor**: Select the variable *\$uptime* from the drop-down menu.

NOTE: Selecting the anchor will draw a line from the variable to the message in the diagnosis result.

- 4. Define the **If** condition as follows:
  - a) A: Variable *uptime* | Contains | day.
  - b) B: Variable *uptime* | Contains | week.
  - c) Boolean Expression: A or B.

	2. Define I	Diagnosis							
Q ~ ^	📄 Add Note	Add I	Diagnosis	Ca	an also clici	k a varial	ole on the left to a	dd au	utomation.
hours,	Name: Cl	heck Uptime	2		A	nchor:	\$uptime		~
	Ty	ype description o	f the diagnos	is					
	Loop Ta	ible Rows							- \
	A 🥏	US-BOS-S	Current 🗸						
	a ur	ptime	~	Contains	~	day		~	Î
	В 🥏	US-BOS-S	Current 🗸						
	b up	ptime	~	Contains	~	week		~	۲.
	C Se	elect Variable	~						
	c Boolea	n Expression:	A or B						
	✓ Then								

5. **Then**: In case **If** logic is true, define the color (**green**), status (**Success**), and message as *\$this\_device* uptime is *\$uptime*.

NOTE: By typing \$, you can get the variable selection pop up.

- 6. Check the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.
- 7. **Else**: In case **If** logic is not true, define the color (**red**), status (**Error**), and message as *\$this\_device* uptime is *\$uptime* because of *\$reload\_reason*.

NOTE: By typing \$, you can get the variable selection pop up.

- 8. Check the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.
- 9. Click **Apply** to save the settings and then close the window.

Diagnosis Mess	age:	$\Box$ Save to Incident $\equiv$
5 - st	his_device uptime is \$uptime	
Set Status	Code for Device:	
Success 🗸	\$this_device uptime is \$uptime	
Set Status	Code for Intent:	
Success 🗸	<pre>\$this_device uptime is \$uptime</pre>	
Add Logic ~		
} ~ Else		<u> Delete</u>
Diagnosis Mess	age:	Save to Incident 📃
7 v \$t	his_device uptime is \$uptime because of \$reload_re	ason
Set Status	Code for Device:	
Error	<pre>\$this_device uptime is \$uptime because of \$relo</pre>	oad_reason
🗧 🔽 💿 Set Status	Code for Intent:	
Error	<pre>\$this_device uptime is \$uptime because of \$reld</pre>	oad_reason
+ Add Elself		
}		

10. Back in the **Network Intent** window, click **Save** to save the intent to the Intent Manager.

### 3.3.2 Define Diagnosis for Show Process CPU

- 1. Go to the **Define Diagnosis** ribbon and click on **Add Diagnosis**.
- 2. Enter the diagnosis name as **Check CPU Usage** and select the variable **\$CPU** for the anchor.

**NOTE**: Anchor helps when viewing the resulting diagnosis. It will draw a line from the diagnosis message to the variable you select as the anchor.

3. Define the **If** condition as:

#### A: Variable *cpu* | Greater than | 50.

NOTE: Set the condition as *cpu is Greater than* 80 (or any number you think implies the high CPU in your network).

4. **Then**: In case **If** logic is true, define the color (**red**), status (**Error**) and message *\$this\_device* CPU utilization is high *\$cpu*.

NOTE: By typing \$, you can get the variable selection pop up.

5. Check the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.

	2. Define Diag	nosis							
Q ~ ^	📄 Add Note 🧲	Add Diagnosis	Can als	o click a varia	ble on the left to add a	automation.			
Check CPU Usage	Name: Check	CPU Usage		Anchor:	\$cpu	~			
	Туре с	description of the diagnos	is						
	<ul> <li>Loop Table Rows</li> <li>If</li> </ul>								
	A 🥏 US-BOS-S Current 🗸								
	3 cpu v Greater than v 50 v m B Select Variable v								
	~ Then								
	📄 Diagnosis N	Message:			Save to Incider	nt 🔳			
{	4 sthis_device CPU utilization is high \$cpu								
{	Set Status Code for Device:								
	Error	\$this_device CP	U utilization is high	\$cpu					
	🔽 💿 Set Sta	tus Code for Intent:							
	Error		U utilization is high						

6. **Else**: In case **If** logic is not true, define the color (**green**), status (**Success**) and message as *\$this\_device* CPU utilization is normal.

NOTE: By typing \$, you can get the variable selection pop up.

- 7. Check the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.
- 8. Click **Apply** to save the settings and then close the window.

v Else		The Delete
🗎 Diagnosis N	Message:	$\Box$ Save to Incident $\equiv$
	<pre>\$this_device CPU utilization is normal</pre>	
🔽 💽 Set Sta	tus Code for Device:	
Succe	ss 🗸 \$this_device CPU utilization is normal	
🔽 💽 Set Sta	tus Code for Intent:	
Succe	ss <b>v</b> \$this_device CPU utilization is normal	
AddLegic		
+ Add Elself		
		Can 8 Apply

9. Back in the **Network Intent** window, click **Save** to save the intent to the Intent Manager.

Network Intent (Edit Mode)		×
I Cisco Device Level CPU Check	osis Tree	Run with Live Data 8 Save 9 Help =
Type description here		👗 Intent Map: Select 🗸
I Seed Logic	<b>A</b>	
Herice		Intent Variables: Manager   Tag: + Add 🗐
v 🚄 US-BOS-SW1	Type Description here	Image: The second s
🔺 💷 show process cpu 💈	Type Description here	Edit Diagnosis 🛛 🚍
1 JS-BOS-SW1>show process cpu		Check CPU Usage
2 CPU utilization for five secor 3 PID Runtime(ms) Invoked	ds: 0%/0%; one minute: 0%; five minutes: 0% uSecs 5Sec 1Min 5Min TTY Process	
4 1 0 9	0 0.00% 0.00% 0.00% 0 Chunk Mana	📄 Message 🚺 Status Code
5 2 130752 3604525	36 0.00% 0.00% 0.00% 0 Load Meter	5
6 3	0_0.00% 0.00% 0.00% 0.5panTree_F	lush

## 3.3.3 Define Diagnosis for Show interface

In this section, we need to create two diagnoses separately to <u>Check Status</u> and <u>errors</u> as follows:

3.3.3.1 Create a Diagnosis to check Interface Status

- 1. Go to the **Define Diagnosis** ribbon and click on **Add Diagnosis**.
- 2. Enter the diagnosis name as **Interface Status** and select the variable *\$interface* for the anchor.
- 3. Check in the Loop Table Rows and select the table Pattern1.
- 4. Select the variable **\$interface** for the Table Key.
- 5. Define the **If** condition as follows:
  - a) A: Variable *L1status* | Contains | down.
  - b) B: Variable *L2status* | Contains | down.
  - c) Boolean Expression: A or B.

Name:	Interface Status	2		And	hor:	Pattern1.\$interface	~
	Type description	of the diagnos	is				
🔽 Loop 🗸 If	o Table Rows 🔎	Pattern1 🗸	Table Key: j	nterface	~	4	ĝ
A	🥔 US-BOS-S	Current 🗸					
	L1status	~	Contains	~	down	~	Î
В	🥏 US-BOS-S	Current 🗸					
	L2Status	~	Contains	~	down	~	Î
С	Select Variable	~					
	lean Expression:	A or B					

- 6. **Then**: In case **If** logic is true, define the color (**red**), status (**Error**), and message On *\$this\_device*, Interface *\$interface* is down, Current L1 status is *\$L1status* and L2 status is *\$L2Status*.
- 7. Check the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.
- 8. **Else**: In case **If** logic is not true, define the color (**green**), status (**Success**) and message as On *\$this\_device*, Interface *\$interface* is UP.
- 9. Check the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.
- 10. Click **Apply** to save and create the intent.

📄 Add Note	Add Diagnosis	Can also click a variable on the left to add auton
∨ Then		
📄 Diagnosis I	Message:	Save to Incident
~		nterface is down, Current L1 status is \$L1status and L2
🔽 💿 Set Sta	atus Code for Device:	
Error	✓ on \$this_device, Interf	ace <a>sinterface</a> is down, Current L1 status is <a>sheet</a> status
🔽 💿 Set Sta	atus Code for Intent:	
Error	on this douise Interf	in a distante in deux. Comment I 1 status is di 1 status
Add Logic 🗸		ace \$interface is down, Current L1 status is \$L1status
Add Logic ~ ~ Else		m Dek
Add Logic 🗸	Message:	፼ Dek
Add Logic ~ ~ Else Diagnosis I	Message: On \$this_device, Interface \$	፼ Dek
Add Logic ~ ~ Else Diagnosis I	Message:	፼ Dek
Add Logic ~ ~ Else Diagnosis I	Message: On \$this_device, Interface \$ atus Code for Device:	☐ Dek     ☐ Save to Incident      = interface is UP
Add Logic ~ Flse Diagnosis I Government	Message: On \$this_device, Interface \$ atus Code for Device:	☐ Dele     Save to Incident      interface is UP
Add Logic ~ Flse Diagnosis I Government	Message: On \$this_device, Interface \$ atus Code for Device: ess V On \$this_device, Inter atus Code for Intent:	Dele Save to Incident face is UP

#### 3.3.3.2 Create a Diagnosis to check interface errors

- 1. Go to the **Define Diagnosis** ribbon and click on **Add Diagnosis**.
- 2. Enter the diagnosis name as **Check Errors.**
- 3. Select the variable *\$input\_error* for the anchor.
- 4. Check in the Loop Table Rows and select the table Pattern1.
- 5. Select the variable **\$interface** for the Table Key.
- 6. Define the **If** condition as follows:
  - a) A: Variable *input\_errors* (Current)|Does not equal | *input\_errors* (Last)
  - b) B: Variable CRC (Current) | Does not equal | CRC (Last)
  - c) C: Variable *output\_errors* (Current)|Does not equal | *output\_errors* (Last)
  - d) Boolean Expression: A or B or C

📄 Add N	ote 🗾 Add	Diagnosis	1 Can a	lso clic	k a variable on the l	left to add au	tom
Name:	Check Errors	2		A	nchor: Pattern1	.\$input_err.	. ~
	Type description	of the diagnos	is				
🔽 Loop	o Table Rows 🏹	Pattern1 🗸	Table Key: in	terfac	e 🗸 5		6
А	🬌 US-BOS-S	Current 🗸				Last 🗸	
	input_errors	~	Does not equal	~	input_errors	~	ī
В	🥏 US-BOS-S	Current 🗸				Last 🗸	
	CRC	~	Does not equal	~	CRC	~	Ē
С	🥏 US-BOS-S	Current 🗸				Last 🗸	
i i	output_errors	~	Does not equal	~	output_errors	~	Ē
D	Select Variable	~					
	lean Expression:	A or B or C					

7. Then: In case If logic is true, define the color (red), status (Error) and message as: *\$this\_device* Interface *\$interface* has:
Current input errors *\$input\_errors* and last input errors *\$input\_errors(Last)*Current CRC *\$CRC* and last CRC *\$CRC(Last)*Current output errors *\$output\_errors* and last *\$output\_errors(Last)*

NOTE: By typing \$, you can get the variable selection pop up.

- 8. Check the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.
- 9. **Else**: In case **If** logic is not true, define the color (**green**), status (**Success**) and message as Errors on *\$this\_device has not increased*.
- 10. Check the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.
- 11. Click **Apply** to save and create the intent.

	lessage	
~	\$this_	_device, Interface \$interface has: Current input errors \$input_errors and las
Set Statu	us Cod	le for Device:
Error	~	<pre>\$this_device, Interface \$interface has: Current input errors \$input_errors a</pre>
Set State	us Cod	de for Intent:
Error	~	<pre>\$this_device, Interface \$interface has: Current input errors \$input_errors a</pre>
dd Logic 🗸		
Else		m Dele
Else	essage	
	-	
	Error	e: Save to Incident =
Diagnosis M	Errors	e: Save to Incident =
Diagnosis M	Errors	e: Save to Incident = s on this device \$this_device have not increased de for Device: Errors on this device \$this_device have not increased
Set State	Errors	es on this device <b>\$this_device</b> have not increased de for Device: Errors on this device <b>\$this_device</b> have not increased
Diagnosis M Set Statu Success Set Statu	Errors	e: Save to Incident = s on this device \$this_device have not increased de for Device: Errors on this device \$this_device have not increased de for Intent:

12. Back in the **Network Intent** window, click **Save** to save the intent to the Intent Manager.

# 3.4 Monitor F5 Virtual Server Status

Besides the overall health check, you may also automate some important operational status. For example, for an F5 load balancer, you may create an intent to check the status and state of the virtual servers and their pool member using the CLI command *show ltm virtual detail recursive*.

## 3.4.1 Parse variables from *show ltm virtual detail recursive*

The output of the command *show ltm virtual detail recursive* may include the multiple virtual servers, and so you use the Paragraph Parser to parse the data like what you have done for the command *show interface*. A paragraph of a virtual server may have multiple pool members, so you need to create a Paragraph Parser for the Paragraph. In other words, you create a Paragraph Parser within a Paragraph Parser (the Parent Parser).

The result will be two tables: the virtual server and the pool member. And you can create two separate diagnoses from the same sample data.

- 1. In the command field, enter the command *show ltm virtual detail recursive*.
- 2. Click Retrieve.
- Use Paragraph Parser to parse the virtual server since there can be multiple virtual servers. Define the ID line as Virtual Server: \$Virtual\_Server. Parse the variables and modify them as listed in the following table:

Variable	ID/Var Line
Virtual Server	Virtual Server: <i>\$Virtual_Server</i>
Status	^ Availability: <i>\$mstring:Status</i>
State	^ State: <i>\$string:state</i>
CPU Usage	Last 5 Minutes <i>\$int:CPU_Usage_Minutes</i>

4. Change the name of the paragraph parser to **Virtual Server**.

CLI Command Diagnosis	×
1. Define Variable	2. Define Diagnosis
Format1 V +	Test on Devices: 0 🗧
Obuble-click a variable to parse. Select multiple lines to parse a table.          Current Device       07/15/2024 01:02:35 PM       Search       Q       I         1       poot@(12)(cfg-sync Sync Failed)(Active)(/)(tmos)#sho       P1-ID Line A         2       4       +Ltm::Virtual Server: Common/F5-lab       P1-ID Line A         5       5       Var Line 1         7       Availability       : available         8       State       : enabled         9       Reason       : The virtual serve         13       Traffic       ClientSide         14       Traffic       278         15       Bits In       176.0K         16       Bits Out       326         19       Current Connections       0         20       Maximu Connections       24         21       Total Connections Killed       0         23       Slow Connections Killed       0         24       Wire Connections Killed       0	<pre>     Virtual_Server 4</pre>
	Var Line 3 Last 5 Minutes \$int:CPU_Usage_Minutes

- 5. Create a second pattern to extract the pool member variables and add it as a child to the pattern **Virtual\_server** (the parent parser).
- 6. Go to the **+New Pattern** drop down menu and choose **Paragraph**.

				Test on Devices:	: <b>0</b> =
÷	Virtual_Server	🗸 🞽 Type: Paragra	aph 👔 🛛 + New Pattern 🗸		Ξ
			Auto Pattern		
	ID Line A V	irtual Server: \$Virtua	Single Variable	≡	
		4 Ltm::Virtual Server: Comn	Table	> 1 Line	
			Paragraph 6		
	Var Line 1	^ Availability	Advanced	≡	
		7 Availability : availa	ble	> 1 Line	

7. In the new paragraph, Go to the menu  $\equiv$  and select **Add Parent**.

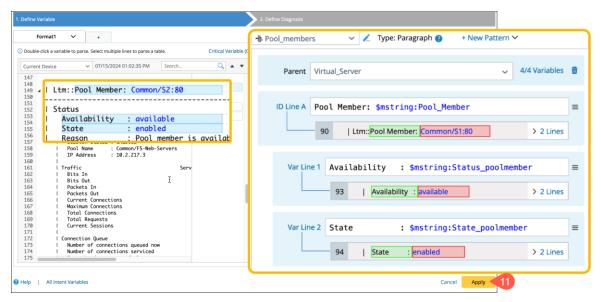
8. Rename the pattern name to **Pool\_members**.

	Test on Dev	vices: 0
🖶 Paragraph1 🛛 🚯 🍈 Type: Par	agraph 👔 🛛 + New Pattern 🏏	:
ID Line A	Add Parent Add Start Line	7
+ Field ~	Add End Line Define Replace	ment
	Comparison Se End of Paragrag	-
	Rename Delete	
	Insert Pattern	>

- 9. In the Parent field, select **Virtual\_Server** from the drop down menu.
- 10. Parse the pool member, state and availability as listed in the following table:

Variable	ID/Var Line
Pool Member	Pool Member: <i>\$mstring:poolmember</i>
Status	Availability: <i>\$mstring:Status_poolmember</i>
State	State: <i>\$string:state_poolmember</i>

11. Click **Apply** to save and go to **Define Diagnosis.** 



# 3.4.2 Define Diagnosis for Monitor F5 virtual server

In this use case, we shall create three different diagnoses each for:

- 1. <u>Virtual server status and state</u>
- 2. <u>Checking CPU usage</u>
- 3. Pool member status and state

#### 3.4.2.1 Create a Diagnosis to check the virtual server Status and State

- 1. Go to the **Define Diagnosis** section and click on **Add Diagnosis**.
- 2. Enter the diagnosis name as Virtual server status and state.
- 3. **Anchor**: Select the variable *\$Virtual\_server* from the drop-down menu.
- 4. Check in the **Loop Table Rows** and select the table **Virtual\_server.**
- 5. Select the variable **\$Virtual\_server** for the Table Key.
- 6. Define the **If** condition as follows:
  - a) A: Variable *status (Current)* | Does not equal | *Status* (Baseline).
  - b) B: Variable *State (Current)* | Does not equal | *State* (Baseline).
  - c) Boolean Expression: **A or B.**

📄 Add N	ote D Ac	ld Diagnosis ┥	1 Can als	so click	k a variab	ole on the lef	t to add au	itoma
Name:	Virtual Server	Status and Stat	e 2	А	nchor:	Virtual_Ser	ver.\$Virt	
	Type description	n of the diagnos	is					
🗸 Loop	Table Rows	Virtual_Serv	rer 🗸 🛛 Table Key	y: Virt	ual_Serv	ver 🔨 5		Ø
✓ If								
				_		_	_	
A	<b>a</b> 12	Current ~				В	aseline 🗸	
	🛃 l2 Status	Current ~ ~	Does not equal	~	Status	В	aseline 🗸	Î
			Does not equal	~	Status			Î
А	Status	~	Does not equal Does not equal	~	Status		~	
А	Status	✓ Current ✓					✓ aseline ✓	

- 7. Then: In case If logic is true, define the color (red), status (Error) and message as: On *\$this\_device* Virtual Server *\$Virtual\_Server* status or state has changed: Current status is *\$Status* and Baseline status was *\$Status(Baseline)* Current state is *\$state* and Baseline state was *\$state(Baseline)*
- 8. Check-in both the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.
- 9. **Else**: In case **If** logic is not true, define the color (**green**), status (**Success**) and message as *\$this\_device* Virtual Server *\$Virtual\_Server* status *\$Status* or state\$ *state* is not changed.
- Check-in both the selection boxes of the Set Status Code for Device and Set Status Code for Intent to duplicate the message to the Device Status Code.
- 11. Click **Apply** to save the settings and then close the window.

	Note	D Add Diagnosis	Can also click a variable on the left to add auton
📄 Di	iagnosis Me	ssage:	Save to Incident
	~ (	On \$this_device Virtual Server \$Virt	ual_Server status or state has changed: Currer
	Set Statu	s Code for Device:	
	🚺 Error	<ul> <li>On \$this_device Virtual Server</li> </ul>	r <b>\$Virtual_Server</b> status or state has changed:
	Set Statu	s Code for Intent:	
	🕕 Error	On \$this_device Virtual Server	r \$Virtual_Server status or state has changed:
<ul> <li>Else</li> <li>Diagnosis Message:</li> </ul>			Dele
ם 📄	iagnosis Me		
	vi	ce Virtual Server \$Virtual_Server sta	Save to Incident =
	vi Set Statu	s Code for Device:	atus:\$Status and state:\$state is not changed
	Set Statu	s Code for Device:	atus:\$Status and state:\$state is not changed
	Set Statu Success Set Statu	<pre>ice Virtual Server \$Virtual_Server sta s Code for Device: \$this_device Virtual Server \$V s Code for Intent:</pre>	atus:\$Status and state:\$state is not changed /irtual_Server status:\$Status and state:\$state is
	Set Statu	<pre>ice Virtual Server \$Virtual_Server sta s Code for Device: \$this_device Virtual Server \$V s Code for Intent:</pre>	Save to Incident atus:\$Status and state:\$state is not changed /irtual_Server status:\$Status and state:\$state is /irtual_Server status:\$Status and state:\$state is
	Set Statu Set Statu Success Set Statu Success	<pre>ice Virtual Server \$Virtual_Server sta s Code for Device: \$this_device Virtual Server \$V s Code for Intent:</pre>	atus:\$Status and state:\$state is not changed /irtual_Server status:\$Status and state:\$state is

#### 3.4.2.2 Create a Diagnosis to check virtual server CPU usage

- 1. Go to the **Define Diagnosis** section and click on **Add Diagnosis**.
- 2. Enter the diagnosis name as **Check CPU Usage**.
- 3. **Anchor**: Select the variable *\$Virtual\_server* from the drop-down menu.
- 4. Check in the Loop Table Rows and select the table Virtual\_server.
- 5. Select the variable **\$Virtual\_server** for the Table Key.
- 6. Define the **If** condition as:

A: Variable CPU\_Usage\_Minutes | Greater than | 50

Add N	De Add Diagnosis	Can also click	a variable (	on the left to add au	utom
Name:	Check CPU usage	A	nchor: Vir	tual_Server.\$Virt	. 🗸
	Type description of the diagnosis				
🔽 Loop	Table Rows 📜 Virtual_Server 🗸	Table Key: Virt	ual_Server	~ 5	Ę
✓ If	Table Rows 📜 Virtual_Server 🗸	Table Key: Virt	ual_Server	~ 5	Ę
✓ If		Table Key: Virt	ual_Server	~ 5	6

7. **Then**: In case **If** logic is true, define the color (**red**), status (**Error**) and message as "*\$this\_device* Virtual server *\$Virtual\_Server*, CPU usage *\$CPU\_Usage\_Minutes* is high".

NOTE: By typing \$, you can get the variable selection pop up.

8. Check-in both the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.

- 9. **Else**: In case **If** logic is not true, define the color (**green**), status (**Success**) and message as *\$this\_device* Virtual Server *\$Virtual\_Server*, CPU usage *\$CPU\_Usage\_Minutes* is normal.
- 10. Check-in both the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.
- 11. Click **Apply** to save the settings and then close the window.

Add Note D	Add Diagnosis	Can also click a variable on the left to add autom
√ Then		
📄 Diagnosis Messa	ge:	$\Box$ Save to Incident $\equiv$
▶ <b>~</b> \$th	is_device Virtual server	Virtual_Server, CPU usage \$CPU_Usage_Minutes is hig
🔽 💿 Set Status Co	ode for Device:	
❶ Error 🗸 🗸	\$this_device Virtual se	erver \$Virtual_Server, CPU usage \$CPU_Usage_Minutes
Set Status Co	ode for Intent:	
🌖 Error 🔍 🗸	\$this_device Virtual se	erver \$Virtual_Server, CPU usage \$CPU_Usage_Minutes
✓ Else		
		面 Delet
Diagnosis Messa		Save to Incident
Diagnosis Messa	is_device Virtual server s	_
<ul> <li>Diagnosis Messa</li> <li>\$th</li> </ul>	is_device Virtual server s	Save to Incident Virtual_Server, CPU usage \$CPU_Usage_Minutes is not
Diagnosis Messa th Set Status Co	is_device Virtual server solution of the server solution of the server solution of the server ser server serve server ser	Save to Incident Virtual_Server, CPU usage \$CPU_Usage_Minutes is no
Diagnosis Messa th th Set Status Co Success v	is_device Virtual server solutions ode for Device: \$this_device Virtual so ode for Intent:	Save to Incident Virtual_Server, CPU usage \$CPU_Usage_Minutes is no
Diagnosis Messa th th Set Status Co Success - Set Status Co Set Status Co	<pre>is_device Virtual server 3 ode for Device:     \$this_device Virtual server 3 ode for Intent:     \$this_device Virtual server 3 </pre>	<pre>\$Virtual_Server, CPU usage \$CPU_Usage_Minutes is no erver \$Virtual_Server, CPU usage \$CPU_Usage_Minutes</pre>

12. Back in the **Network Intent** window, click **Save** to save the intent to the Intent Manager.

Network	Intent (Edit Mode	2)		×
🚺 Monit	tor F5 Virtual Sei	rver 🚠 Diagnosis Tree	Run with Live Data 12 Save	🛿 Help 📃
Type desc	ription here		💦 Intent	Map: Select 🗸
	Seed Logic	🗍 Replication Logic 🛕		
3	+ Device		Intent Variables: Manager   Ta	ag: + Add 🔳
~ 🥶 I2	2	Type Description here…	🔁 + Add Config Diagnosis 💷 + Add CLI Dia	gnosis 📃
⊿ Œ s	how ltm virtual de	etail recursive	Edit Di	iagnosis ≡
1	root@(12)(cf	fg-sync Sync Failed)(Active)(/)(tmos)#show ltm virtual detail re	ecursiv	0
3			Check CPU usage	Ē
45	Ltm::Virtual	l Server: Common/F5-lab	📄 Message 🛛 Status Code	
6	Status		Pool Member State and Sta	tu 🗑
7	Availabili State	ity : available : enabled	Message S Status Code	
9	Reason	: The virtual server is available	Message Status Code	
10	CMP	: enabled	Virtual Server Status and St	ate 💼
11	CMP Mode	: all-cpus	Message     Status Code	
12 13	Destinatio	on : 10.1.217.50:80	invessage is status code	
13	Traffic	ClientSide Ephemeral General		
15	Bits In	176.0K 0 -		
16	Dito Out	<i>λ</i> 77 <i>γν</i> Ω		4
1	Design 1	120 (1		

#### 3.4.2.3 Create a Diagnosis to check pool member Status and State

- 1. Go to the **Define Diagnosis** section and click on **Add Diagnosis**.
- 2. Enter the diagnosis name as Pool Member State and Status check.
- 3. **Anchor**: Select the variable *\$Pool\_member* from the drop-down menu.
- 4. Check in the Loop Table Rows and select the table Pool\_members.
- 5. Select the variable **\$Pool\_member** for the Table Key.
- 6. Define the **If** condition as:
  - a) A: Variable *Status\_poolmember*(Current) | Does not equal | *Status\_poolmember*(Baseline)
  - b) B: Variable *State\_poolmember*(Current) | Does not equal | *State\_poolmember*(Baseline)
  - c) Boolean Expression: **A or B.**

	2. Defin	e Diagnosis Dte D Add	Diagnosis	1 Can a	also clici	k a varia	ble on the left to	add au	Itomation
	Name:	Pool Member St	ate and Statu	is check 2	A	nchor:	Pool_member	s.\$Po	. ~ 3
		Type description	of the diagnos	sis					
4	🗸 Loop	Table Rows 🏾 🕫	Pool_memb	oers 🗸 🛛 Table	Key: Po	ol_Mer	nber 🗸 5		Ø
$\left\{ \right.$	√ If	6							
}	A	I2	Current 🗸				Base	line 🗸	- 1
}	a	Status_poolmem	ıber 🗸	Does not equal	~	Status	s_poolmember	~	<b>m</b>
}	В	<b>a</b> 12	Current 🗸				Base	line 🗸	
}	b	State_poolmemb	oer 🗸	Does not equal	~	State_	poolmember	~	Ŵ
	С	Select Variable	~						
}	Bool	ean Expression:	A or B	c					

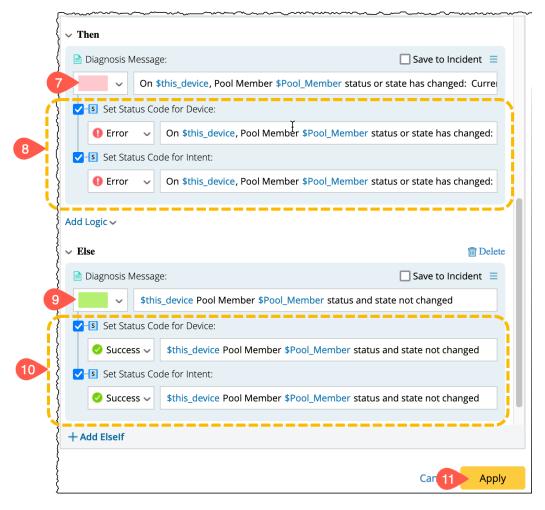
7. Then: In case If logic is true, define the color (red), status (Error) and message as "On *\$this\_device*, Pool Member *\$Pool\_Member* status or state has changed: Current status is *\$Status\_poolmember* and Baseline status was *\$Status\_poolmember(Baseline)* Current state is *\$State\_poolmember* and Baseline state was *\$State\_poolmember(Baseline)*"

NOTE: By typing \$, you can get the variable selection pop up.

- 8. Check-in both the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.
- Else: In case If logic is not true, define the color (green), status (Success) and message as \$this\_device Pool Member \$Pool\_Member status or state not changed.

NOTE: By typing \$, you can get the variable selection pop up.

- Check-in both the selection boxes of the Set Status Code for Device and Set Status Code for Intent to duplicate the message to the Device Status Code.
- 11. Click **Apply** to save the settings and then close the window.



## 3.5 Execute Intents and Create Dashboards

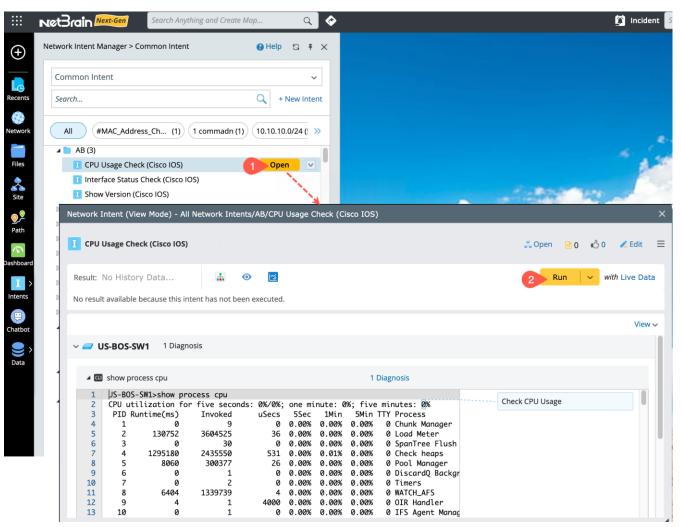
Now, let us create intent dashboards for each intent created in Chapter 3 and previous Chapter 2. And then a summary dashboard which provides an overview displaying results from multiple Intent dashboards.

- 1. Intent Dashboard to view the individual Intent results and
- 2. Summary Dashboard to view the consolidated Intent results in a single view.

### **3.5.1 Execute Intents**

Let us execute all the intents created in this chapter and add them to a dashboard as follows:

- 1. Open the intent CPU Usage Check (Cisco IOS) from the intent manager.
- 2. Click Run.



3. After successful execution, the results will appear as defined in the diagnosis section. To create the intent dashboard from this window, go to next Section 3.4.2.

letwork Intent (View Mode)	- All Network Intents/AB/CPU Usage Check (Cisco IOS)	×
CPU Usage Check (Cisco I	os)	🗸 Open 📄 0 🔥 0 🖍 Edit 🚍
Result: 07/26/2024 12:13 AM This intent execution is finishe	d at 07/26/2024 12:13 AM with 1 err Click to create dashboard	Run vith Live Data
V Z US-BOS-SW1	SUS-BOS-SW1 CPU utilization is normal	
2 Execution log: 3 2024-07-25 1 4 2024-07-25 1 5 2024-07-25 1 6 2024-07-25 1 7 2024-07-25 1 8 2024-07-25 1 9 2024-07-25 1	t to the device US-BOS-SW1. 4:43:34 Sending task(contain 4:43:34 Front Server netbrainfs(192.168.30.51) received the task( 4:43:34 Dispatch the task to live thread 4:43:34 Start running the task in live thread 4:43:34 Telnet to device 10.8.1.241 4:43:35 Failed to access the device, Front Server will not access 4:43:35 Can not Telnet/SSH to the device.	ī 6

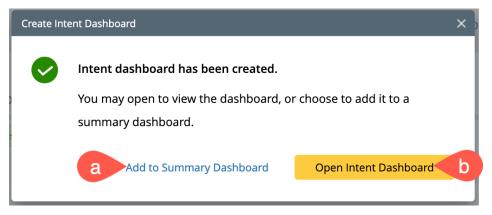
## 3.5.2 Intent Dashboard

Create an Intent Dashboard directly from the Network Intent results window as follows:

- 1. Click the **Create Intent Dashboard** icon in the intent results window to open the corresponding window.
- 2. In the Create Intent Dashboard window:
  - a) Enter the Dashboard Name as **CPU utilization check**.
  - b) Select the **Location** to save the Intent Dashboard.
  - c) **Data Source:** By default, **Specified Intent** is selected from the dropdown.
  - d) **Intent**: Created intent is selected by default. Keep it the same.
  - e) Click **Create**.

I CPU Usage Check (Cisco IOS)		
	Create Intent Dashboard	×
Result: 07/26/2024 12:13 AM V	Create Intent Dashboard for 'CPU Usage Check (Cisco IOS	5)'
S US-BOS-SW1 CPU utilization is normal 2	Name: CPU utilization check	
✓	Location: My Dashboards	
▲ 💷 show process cpu	Use Template	
<ol> <li>Failed to connect to the device US-BOS-SW1.</li> <li>Execution log:</li> <li>2024-07-25 14:43:34 Sending task(contait</li> <li>2024-07-25 14:43:34 Front Server netbrox</li> </ol>	i Data Source Specified Intent	~
5 2024-07-25 14:43:34 Dispatch the task t 6 2024-07-25 14:43:34 Start running the t	t Intent: CPU Usage Check (Cisco IOS)	
7 2024-07-25 14:43:34 Telnet to device 10 8 2024-07-25 14:43:35 Failed to connect, 9 2024-07-25 14:43:35 Failed to access th 10 2024-07-25 14:43:35 Can not Telnet/SH	<ul> <li>Include Triggered Follow-up Intent Results</li> </ul>	
k		
erver literation		
sit Gateway Config Compliance		
Library (186)	Time Range Last 7 Days	~
vork Dependency Assessment (5)		
sment (24)	Cancel	Create
Assessment (48)		

3. A prompt window appears with the success message Intent Dashboard has been created with two options: Open Intent Dashboard or Add to Summary Dashboard.



- a) **Add to Summary Dashboard:** The Summary dashboard provides an overview displaying results from multiple Intent dashboards.
- b) **Open Intent Dashboard:** Displays the intent executed results of the created intent. Refer to Section 2.9.2 for more information.
- 4. Click **Add to Summary Dashboard**. Refer to next <u>Section 3.4.3</u> to view all the Cisco device network intent results together in a single dashboard.

### 3.5.3 Summary Dashboard

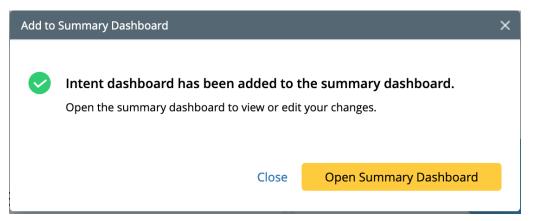
The summary dashboard provides an overview displaying results from multiple Intent dashboards of the entire network or a set of network devices. With Summary Dashboard, you can group Intent Dashboards into widgets based on diagnosis purpose and display results by device, site or device groups. You can use the summary dashboard to monitor critical information across thousands of devices and discover the root cause for issues in one view.

Let us create a Summary Dashboard using the step-by-step instructions as follows:

- 1. After selecting **Add to Summary Dashboard**, its corresponding window will appear.
- 2. In the **Add to Summary Dashboard** window, let us create a new summary dashboard and new dashboard group as follows:
  - a) **Summary Dashboard**: open the dropdown menu.
  - b) Select +New Summary Dashboard to pop up its dialogue.
  - c) Enter the dashboard basic details like **name**, **group title** and **location** of the summary dashboard to save.
  - d) Click **OK** to save and create the summary dashboard.

Add to Summary Dashboa	ırd		×		
Summary Dashboard:	Select or create New		a ×	Select or create New	
Dashboard Group:	Select or create New		~	Search	Q
		Cancel	ОК	<ul> <li>Shared Dashboards (2)</li> <li>Murali_Kirthi (1)</li> <li>adf</li> </ul>	
		_		+ New Summary Dashboard	
New Summary Dash	hboards irthi	×		 ****	
Dashboard Name:	Device Health Monitor		С		
Dashboard Group	Title: Cisco IOS Devices				
	Cancel	ОК	d		

3. A prompt window appears with the success message "Intent dashboard has been added to the summary dashboard".



4. Close the window and add two other intents, **Uptime check** and **Interface status check**, to dashboards created for Cisco IOS switch devices in Chapter 3. The final summary dashboard will be:

Device Health Monitor			Last Refreshed at 26/7/2024,	12:57:21 am 🖉 🗏
Summary				
Number of Intents	Number of Devices	Number of Alerts	Number of Successes	
3	1	0	76	
Cisco IOS Devices				
Number of Intents	Number of Devices	Number of Alerts	Number of Successes	
3	1	0	76	
	-			Q
Dashboard and Intent Group	Intent Results	Device Results		< >
Dashboard and Intent Group	Intent Results	Total Device Results	US-BOS-SW1	
CPU utilization check	1	1 0	1	
Interface Check	36	36 0	36	
Uptime Check	1	1 0	1	
Total Alert Count $\downarrow$				0

# **4** Enable Others to Use Your Automation

In the last two chapters, you learned how to create intents to automate a task usually done via the CLI commands. The intent results can be viewed in the map, the diagnosis tree, and in the Dashboard. An intent is always associated with one or multiple devices. The intent diagnosis can be duplicated to other devices, which are included as the seed devices. However, this duplication method is not scalable. In this chapter, you will learn a scalable way to replicate an intent to other devices via an easy-to-follow wizard, **Auto intent wizard**. While a user opens a map, The intents (**auto intents**) that are qualified for the devices in the map will be listed in the **Auto Intent** pane.

In this chapter, we will introduce a key feature, **follow up intent**, and teach how you can use the follow up intent to encapsulate a set of intents into a parent intent. This parent intent, sometimes called **wrapper intent**, hides the detailed intent implementation from the end user and can function as a public interface to the end user, which will be accessed as an auto intent or as a chatbot.

This chapter will reuse the intents created in previous <u>Section 2</u> and <u>Section 3</u> and cover the following flows:

- 1. <u>Use Auto Intent Wizard for others to use the automation.</u>
- 2. <u>Use Follow up intent to create a wrapper intent.</u>
- 3. Chatbot to run intents.

## 4.1 Auto Intent Wizard: Replicate Intent to Multiple Devices

Replicating the intents to multiple devices with **Auto Intent Wizard** is straightforward. The steps involved are detailed as follows:

- 1. Click the **Intents** menu located on the desktop sidebar and select **Auto Intent Wizard** to open its corresponding wizard.
- 2. Input the data for replication in the following three ribbons of the Auto Intent Wizard:
  - Intent Template
  - <u>Pre Decode</u>
  - <u>Auto Intent</u>

			Auto Intent Wizard				×
			Inter	nt Template	Pre-Decode	Auto Intent	
Image: state sta	Files One-IP Table	en Search Any	Seed Intent:		Select	(2) Full Settings for Templat	te
Path	Ping Cisco		Select a seed intent to r	replicate and define the key para	neters to convert a seed intent to temp	late. Ner	xt
Intents	Intent Manager	Intent Replice	Auto Intent Wizard				
) Data	Automation Data Table (ADT)	Intent Based Automation Center	Schedule Intent				

### 4.1.1 Intent Template

In this ribbon, you will provide the basic details like seed intent, target devices and template settings such as **macro variables**.

- 1. Seed Intent: Click Select and choose the intent Ping Cisco from the Select Intent dialog.
- 2. Intent Template for: Choose the Device-based Replication.
- 3. Intent Qualification: Click the **Select** drop-down menu beside **Define via Device Group/Sites**.
- 4. Click **Select Device Groups** to open its corresponding window.
  - a) Check the selection box of the device group you want to replicate.
  - b) Click OK.

Auto Intent Wizard	d - Ping Cisco						×		
	Intent Template		Pre-Decode		A	uto Intent			
					Ę	္ဌိ} Full Settings for Tem	plate		
Seed Intent:				25:21 AM 07/20/2024					
	late for: <ul> <li>Device-based Replaced</li> <li>Device Group</li> <li>Device Group</li> </ul>		ased Replication O via Dynamic Search: Under	fined					
		Select Device							
Define Ma ltems: 4	acro Variables and Rules for T	Select Sites New Device G	roup	Add Device Group		×			
Seed De	evice	Seed Command		All Device G					
🥔 US-E	BOS-SW1	💷 ping 10.8.1.1	× *		IOS devices (4)				
🚅 US-E	BOS-SW2	💷 ping 10.8.1.1			IOS switches (4)		_		
	BOS-SW3	🖽 ping 10.8.1.1		⊿ □ Shared D			_		
	BOS-SW4	ping 10.8.1.1		<ul> <li>▷ □ Anne</li> <li>▷ □ Anne</li> <li>▷ □ Anne</li> <li>▷ □ Auton</li> </ul>	g				
	Please click to select an entry fr								
Select a seed i	intent to replicate and define t	he key parameters to conve	ert a seed intent to template.				Next		

Now, set the option to pass an interface using **Full Settings for Template** located at the top right corner.

5. Click Full Settings for Template.

o Intent Wizard	- Ping Cisco					:
	Intent Template			Pre-Decode		Auto Intent
Seed Intent:	Ping Cisco		Select	Last Modified at: 02:54:31 PM 12/19/	2023	
	ate for: <ul> <li>Device-based Repland</li> <li>Ilification: <ul> <li>via Device Group</li> </ul> </li> </ul>	-	ath-based Repli	rnamic Search: Defined		
intent qua		porsities. Select o	O via by	Denneu		
Define Mae ltem: 1	cro Variables and Rules for Th	neir Substitution:				Preview
Seed Dev	vice	Seed Command		Macro Va	riables	

6. A prompt will appear to save the intent modifications before proceeding further, Click **Save**.

Unsaved Changes							
1	This intent has been mod proceed?	This intent has been modified. Do you want to save it before you proceed?					
	Cancel	Continue without Saving	Save				

- 7. Toggle the Intent Template on by clicking on the **Serve as Template for**.
- 8. Go to the **Macro Variable** tab and click **+ Command Variable** to select the command to customize.

A **Macro Variable** is a special intent variable that can be used as an input of an intent. For example, for the intent using the command *ping 10.8.1.1*, you can replace the value *10.8.1.1* as a macro variable *\$server\_ip* (the command becomes *ping \$server\_ip*) so that a user can enter any server IP address while executing the intent. Besides functioning as the user input for the CLI command, Macro Variables can also be used to pass the values from a parent intent to follow up intents, which will be covered later.

9. Select the *ping 10.8.1.1* command and click **OK** to open the variable pattern modifier window.

		Device-based Replica		/ Fath-Daseu	Replication 🗌 En	able Neighbor Pai	Replication		
Int	ent Qualification	Macro Variable	Critica	Variable	Advance Settings				
0 ltems	+ Device Variable	+ Command Variab	le 8	Add Com	mand Variable		×'		
4	Seed Device	Macro Variable	Source					for Device	
	🥏 US-BOS-SW1			<b>EU</b> p	ing 10.8.1.1				
	🥏 US-BOS-SW2								
	🥏 US-BOS-SW3								
	🥏 US-BOS-SW4								
					C C	Cancel OK	9		

- 10. In the Variable Pattern dialog, modify **Ping 10.8.1.1** with **Ping \$server\_ip**.
- 11. Add the **Description** and set the **Default Value** to **10.8.1.1**.
- 12. Click **OK** to save and close your settings.

efine Command Va	ariable		×	
Command:	ping 10.8.1.1			
Variable Pattern:	ping 10.8.1.1  (2) Tip: You can define a macro variab			
	Example: show ip ospf 1 -> sh			``````````````````````````````````````
		Command:	ping 10.8.1.1	I
		Variable Pattern:	Ping \$server	
			<b>~</b>	define a macro variable by replacing the string with \$var_name. e: show ip ospf 1 -> show ip ospf \$process_id
		\$server_ip:		
		Description:	Enter the de	estination server ip
		Туре:	string	-
		* Default Value:	10.8.1.1	11
		Prompt for Input:	server_ip	
		Hint:	Set value	
				Cancel OK

- 13. You will directed back to template settings. Set the **Used Default Value** option in the **Look up Data for Device** column from the dropdown menu.
- 14. Click **OK** to save and close the window.

Int	ent Qualification	Macro Variable	Critical Variable	Advar	nce Settings			
ltem	+ Device Variable	+ Command Variable						
4	Seed Device	Macro Variable	Source	Туре	Default Value	Look up Data for Device		
4	🥏 US-BOS-SW1							
		server_ip	Ping \$server_ip	string	10.8.1.1	Used Default Value	~	
	av US-BOS-SW2							
	🥔 US-BOS-SW3							
	av US-BOS-SW4							

- 15. Click the **Preview** button to validate the list of target devices configured with Macro Variables.
- 16. Click **Next** in the Intent Replication Wizard to move to the **Pre-Decode** step.

	Intent Template		Pre-Decode		Auto Intent
					Full Settings for Template
Seed Intent:	Ping Cisco	Sel	ect Last Modified at: 08:54:59	PM 07/20/2024	
	-			111 07/20/2024	
Intent Templa	te for: 🧿 Device-based Rep	blication O Path-base	d Replication		
Intent Qua	lification: 🧿 via Device Gro	ups/Sites: 1 Device Groups/	Folders 🗊 🛛 🔿 via Dynam	ic Search: Undefined	
	ro Variables and Rules for T	heir Substitution:			15 Preview
Items: 4					Preview
Seed Dev		Seed Command		Macro Variables	
🥏 US-BC		💷 ping 10.8.1.1		\$server_ip	
🛹 US-B	Preview Data				×
ar US-Bi	8 items of 8 devices			Search	٩
	Hostname		\$server_ip(US-BOS-SV	V1)	
	av US-BOS-SW1		10.8.1.1		
	US-BOS-SW2		10.8.1.1		
	ar US-BOS-SW3		10.8.1.1		
	US-BOS-SW4		10.8.1.1		
	av US-NYJ-CW01-01		10.8.1.1		
	US-NYJ-CW01-02		10.8.1.1		
	US-NYJ-CW02-01		10.8.1.1		
Select a seed in	US-NYJ-CW02-02		10.8.1.1		16 Nex

### 4.1.2 Pre Decode

With the settings configured in this ribbon **Pre Decode**, the seed intent will be decoded to the target devices. If the Netbrain system fails to retrieve and parse the critical variable of the target devices, then the device will not be qualified for intent replication.

- 1. **Install Intent Template to**: Select a group from the dropdown menu to add the intent in the IBA center.
- 2. Click the **Install & Decode** button to install the intent to the target devices.

NOTE: The status of the intent installation is displayed next to the **Install & Decode** button.

Auto Intent Wizard - Ping Cisco				×
Intent Tem	plate	Pre-Decode	Auto Intent	
			Intent Based Auto	mation Center
Install Intent Template to:	Automation Cookbook 🚽	<ul> <li>රූ Decor</li> </ul>	ding Settings	
2 Install & Decode	Status: Uninstalled			
2 Decode Now	Status: Not Started			ß
Output: 0 devices deco	oded for this intent			
1 Decode the intent template on	n domain devices.		Previous	Next

3. Click **Decode Now** to start decoding the intent to the target devices.

NOTE: Click the **Refresh** icon after a minute to see the updated decode status. The updated decoded devices can be seen at the bottom of the **Decode Now** button.

4. Check in the **Update Intent Baseline** to update the seed intent baseline information before replicating it to target devices and click **OK**.

NOTE: At this step, wait for the decode to complete by refreshing the status until you see a number of devices have been decoded.

Auto Intent Wizard - Ping Cisco					×
Intent Templ	late	Pre-Decode	Auto	Intent	
			Inte	ent Based Automat	ion Center
Install Intent Template to:	Automation Cookbook	✓ @ D	ecoding Settings		
1 Install & Decode	Status: Installed	Decode Now		×	
2 Decode Now	3 us: Not Started		ill start soon, it may take a number of qualified devic	÷	S
Output: 0 devices decoo	ded for this intent	4 Update Intent Ba	seline		
			Cance	ОК	
Occode the intent template on of the intent template on of the intent template.	domain devices.			Previous	Next

#### View decoded results from Intent Based Automation Center:

- 5. Click Intent Based Automation Center at the top right corner to open it in a new window.
  - a) In the Intent Based Automation Center, go to the intent template **Ping Cisco**.

NOTE: Its corresponding results can be seen on the left side of the screen.

b) Click **4 Devices decoded** to view the detailed report and the log of the decoded devices.

Installed Intent Template       Published Intents       Auto Intent Profile       NetBrain Download         terms: 163       + Add Intent Template       Filter:       All       Search       C       Refresh       Intent Template Name: Ping Cisco         a       Intent Template       Location       Intent Decoding       Decoded Device       Auto Intent       Cloned Intents         a       Ajeet       Intent Decoding       Decoded at 21       16       Intent Decoding         StaticRoutes, R.       All Network Inten       Last Decoded at 46       92       Recurring Decode         BGP Golden In       Vew Decode Result       Vew Decode Result       Search       Recurring Decode         Automation Cookoo       Automation Cookoo       Natched Device: 8       Filter:       NoAll Trigger Sources,Auto In       Search       Q       Search       Q       Search       Q       Search       Q       Search       Ping Cisco         Automation Library       Vew Decode Task Run at 07/20/2024 09:09:52 PM       View Decode Task Creation Log       Search       Q       Search       Ping Cisco       Search       Q       Search       Ping Cisco       Natched Device: 8       Filter: NoAll Trigger Sourc       O7/20/2024 09:09:52 PM       O7/20/2024 09:09:5	
<ul> <li>Intent Template Location Intent Decoding Decoded Device Auto Intent Cloned Intents</li> <li>Ajeet</li> <li>STP Lab3 All Network Inten Last Decoded at 21 16</li> <li>Amy</li> <li>StaticRoutes R All Network Inten Last Decoded at 46 92</li> <li>BGP Golden In</li> <li>Automation Cookboo</li> <li>Automation Cookboo</li> <li>Automation Library</li> <li>Matched Device: 8</li> <li>Filter: No,All Trigger Sources,Auto In v</li> <li>Sterrh Cooked At Baseline Data Updated At US-805-SW2 UI-880-SW2 Ping 10.8.1.1 All Trigger Source. 07/20/2024 09:09:52 PM 07/20/2024 09:09:54 PM US-805-SW2 Ping 10.8.1.1 All Trigger Source. 07/20/2024 09:09:55 PM 07/20/2024 09:09:54 PM US-805-SW3 Ping 10.8.1.1 All Trigger Source. 07/20/2024 09:09:55 PM 07/20/2024 09:09:54 PM US-805-SW3 Ping 10.8.1.1 All Trigger Source. 07/20/2024 09:09:52 PM 07/20/2024 09:09:54 PM US-805-SW3 Ping 10.8.1.1 All Trigger Source. 07/20/2024 09:09:55 PM 07/20/2024 09:09:54 PM US-805-SW3 Ping 10.8.1.1 All Trigger Source. 07/20/2024 09:09:55 PM 07/20/2024 09:09:54 PM US-805-SW3 Ping 10.8.1.1 All Trigger Source. 07/20/2024 09:09:55 PM 07/20/2024 09:09:54 PM US-805-SW3 Ping 10.8.1.1 All Trigger Source. 07/20/2024 09:09:55 PM 07/20/2024 09:09:54 PM US-805-SW3 UI-805-SW3 Ping 10.8.1.1 All Trigger Source. 07/20/2024 09:09:55 PM 07/20/2024 09:09:54 PM US-805-SW3 UI-805-SW3 Ping 10.8.1.1 All Trigger Source. 07/20/2024 09:09:55 PM 07/20/2024 09:09:54 PM US-805-SW3 UI-805-SW3 Ping 10.8.1.1 All Trigger Source. 07/20/2024 09:09:55 PM 07/20/2024 09:09:55 PM 07/20/2024 09:09:54 PM US-805-SW3 Ping 10.8.1.1 All Trigger Source. 07/20/2024 09:09:55 PM 07/20/2024</li></ul>	
A jeet STP Lab3 All Network Inten Last Decoded at 21 Amy StaticRoutes R All Network Inten Last Decoded at 46 92 Wew Decode Result Concernation C	
STP Lab3       All Network Inten.,, Last Decoded at ,,, 21       16       Intent Decoding         Army       StaticRoutes R.,       All Network Inten.,, Last Decoded at ,,, 46       92         BGP Golden In.,       View Decode Result       View Decode Result       Intent Decoding         Automation Cookbor       Isat Decoded Devices: 8       Filter: NoAll Trigger Sources,Auto In.,, v       Search.,       Iserch.,       Iserch.,         Automation Library       Matched Device       Matched Seed Device       Matched Command       Create Intent       Last Decoded At       Baseline Data Updated At         USBOS-SW2       USBOS-SW2       USBOS-SW2       ping 10.8.1.1       All Trigger Source., 07/20/2024 09:09:52 PM       07/20/2024 09:09:54 PM       07/20/2024 09:09:54 PM         USBOS-SW1       USBOS-SW2       ping 10.8.1.1       All Trigger Source., 07/20/2024 09:09:54 PM       07/20/2024 09:09:54 PM       07/20/2024 09:09:54 PM         USBOS-SW1       USBOS-SW2       ping 10.8.1.1       All Trigger Source., 07/20/2024 09:09:54 PM       07/20/2024 09:09:54 PM       07/20/2024 09:09:54 PM         USBOS-SW1       USBOS-SW3       ping 10.8.1.1       All Trigger Source., 07/20/2024 09:09:54 PM       07/20/2024 09:09:54 PM       07/20/2024 09:09:54 PM         USBOS-SW3       USBOS-SW3       ping 10.8.1.1       All Trigger Sourcc., 07/20/2024 09:09:54 PM       0	
Amy StaticRoutes R., All Network Inten., Last Decoded at 46 92 Wew Decode Result Last Decoded Result Control Cookbox Automation Cookbox Automation Library Cocep. Device Matched Device Matched Seed Device Matched Command Create Intent Last Decoded At Baseline Data Updated At US-805-SW2 US-805-SW3 ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM US-805-SW4 US-805-SW4 US-805-SW4 Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result Seed Sw1 US-805-SW4 US-805-SW4 Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result US-805-SW4 US-805-SW4 Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result US-805-SW4 US-805-SW4 Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result US-805-SW4 US-805-SW3 Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result US-805-SW4 US-805-SW3 Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result US-805-SW4 US-805-SW3 Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result US-805-SW3 Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result US-805-SW3 Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result US-805-SW3 Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result US-805-SW3 Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Result Ping 10.8.11 All Trigger Source. 07/20/2024 09:09:52 PM View Decode Riew Ping View Decode Riew Ping View Decode Riew Ping	
StaticRoutes P       All Network Inten., Last Decoded at	Decode Now
BGP Golden In       View Decode Result       X         A Automation Cookbor       Last Decode Task Run at 07/20/2024 09:09:52 PM       View Decode Task Creation Log         Pring Cisco       Total Decode Devices: 8       Filter:       No.All Trigger Sources,Auto In V       Search       C Caste Intent       Last Decoded At       Baseline Data Updated At         US-R05/SW2       US-BOS/SW2       US-BOS/SW2       Dis-BOS/SW2       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:09:52 PM       07/20/2024 09:09:54 PM         US-BOS/SW1       US-BOS/SW2       Dis-BOS/SW2       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:09:54 PM       07/20/2024 09:09:54 PM         US-BOS/SW1       US-BOS/SW2       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:09:54 PM       07/20/2024 09:09:54 PM         US-BOS/SW1       US-BOS/SW2       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:09:54 PM       07/20/2024 09:09:54 PM         US-BOS/SW1       US-BOS/SW3       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:09:54 PM       07/20/2024 09:09:54 PM         US-BOS/SW2       US-BOS/SW3       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:09:54 PM       07/20/2024 09:09:54 PM         US-BOS/SW2       US-BOS/SW3       ping 10.8.1.1       All Trigger Sourc	
BGP Golden In         Last Decode Task Run at 07/20/2024 09:09:52 PM         View Decode Task Creation Log           Pring Cisco         Total Decoded Devices: 8         Filter:         No.All Trigger Sources.Auto In v         Search v         Search v         Refresh           Automation Library         Matched Device         Matched Seed Device         Matched Command         Create Intent         Last Decoded At         Baseline Data Updated At           US-BDS-SW2         US-BDS-SW3         ping 10.8.1.1         All Trigger Sourc 07/20/2024 09:09:51 PM         07/20/2024 09:09:54 PM         07/20/2024 09:09:54 PM           US-BDS-SW1         US-BDS-SW2         ping 10.8.1.1         All Trigger Sourc 07/20/2024 09:09:52 PM         07/20/2024 09:09:54 PM         07/20/2024 09:09:54 PM           US-BDS-SW1         US-BDS-SW2         ping 10.8.1.1         All Trigger Sourc 07/20/2024 09:09:52 PM         07/20/2024 09:09:54 PM         07/20/2024 09:09:54 PM           US-BDS-SW1         US-BDS-SW2         ping 10.8.1.1         All Trigger Sourc 07/20/2024 09:09:52 PM         07/20/2024 09:09:54 PM         07/20/2024 09:09:54 PM           US-BDS-SW4         US-BDS-SW3         ping 10.8.1.1         All Trigger Sourc 07/20/2024 09:09:52 PM         07/20/2024 09:09:54 PM           US-BDS-SW3         US-BDS-SW3         ping 10.8.1.1         All Trigger Sourc 07/20/2024 09:09:52 PM         07	
Ping Cisco       Total Decoded Devices: 8       Filter:       No.All Trigger Sources.Auto In       Search       C Refresh         Automation Library       Matched Device       Matched Seed Device       Matched Command       Create Intent       Last Decoded At       Baseline Data Updated At         US-B05-SW2       US-B05-SW3       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:51 PM       07/20/2024 09:08:54 PM         US-B05-SW1       US-B05-SW2       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW1       US-B05-SW2       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW1       US-B05-SW2       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW1       US-B05-SW3       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW3       US-B05-SW3       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW2       US-B05-SW3       ping 10.8.1.1       All Trigger Source       07/20/2024 09:08:54 PM         US-B05-SW2       US-B05-SW2       ping 10.8.1.1       All Trigger Source <td>~</td>	~
Ping Clsco       Total Decoded Devices: 8       Filter:       No.All Trigger Sources,Auto In       Search       C       Refresh         Automation Library       Matched Device       Matched Seed Device       Matched Command       Create Intent       Last Decoded At       Baseline Data Updated At         US-B05-SW2       US-B05-SW3       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:51 PM       07/20/2024 09:08:54 PM         US-B05-SW1       US-B05-SW2       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW1       US-B05-SW2       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW1       US-B05-SW2       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW1       US-B05-SW3       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW1       US-B05-SW3       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW2       US-B05-SW3       ping 10.8.1.1       All Trigger Source       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW2       US-B05-SW2       p	
Automation Library         Matched Device         Matched Seed Device         Matched Command         Create Intent         Last Decoded At         Baseline Data Updated At           US-B05-SW2         US-B05-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:51 PM         07/20/2024 09:09:54 PM           US-B05-SW1         US-B05-SW2         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-B05-SW1         US-B05-SW2         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-B05-SW1         US-B05-SW2         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-B05-SW4         US-B05-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-B05-SW4         US-B05-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-B05-SW3         US-B05-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-B05-SW3         US-B05-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:08:54 PM           US-B05-S	evices decoded
Matched Device         Matched Seed Device         Matched Seed Device         Matched Seed Device         Matched Seed Device         Baseline Data Updated At           US-BDS-SW1         US-BDS-SW1         Pring 10.8.1.1         Autor Intent Curry         Or720/2024 09:09:51 PM         07/20/2024 09:09:54 PM           US-BDS-SW1         US-BDS-SW2         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:09:54 PM           US-BDS-SW4         US-BDS-SW2         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:09:54 PM           US-BDS-SW1         US-BDS-SW4         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:09:54 PM           US-BDS-SW4         US-BDS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:54 PM           US-BDS-SW4         US-BDS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:54 PM           US-BDS-SW3         US-BDS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:54 PM           US-BDS-SW2         US-BDS-SW2         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:54 PM           US-BDS-SW2         US-BDS-SW2         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:54 PM <td>Apply</td>	Apply
US-B05-SW2       US-B05-SW3       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:09:51 PM       07/20/2024 09:08:54 PM         US-B05-SW1       US-B05-SW2       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW4       US-B05-SW2       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW1       US-B05-SW4       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW4       US-B05-SW3       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW3       US-B05-SW3       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:09:52 PM       07/20/2024 09:08:54 PM         US-B05-SW3       US-B05-SW3       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:08:54 PM         US-B05-SW2       US-B05-SW2       ping 10.8.1.1       All Trigger Sourc       07/20/2024 09:08:54 PM	
US-BOS-SW4         US-BOS-SW2         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW1         US-BOS-SW4         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW4         US-BOS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW3         US-BOS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW3         US-BOS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW3         US-BOS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW2         US-BOS-SW2         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
US-BOS-SW1         US-BOS-SW4         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW4         US-BOS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW3         US-BOS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW3         US-BOS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW2         US-BOS-SW2         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM	
US-BOS-SW4         US-BOS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW3         US-BOS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW3         US-BOS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW2         US-BOS-SW2         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM	
US-BOS-SW3         US-BOS-SW3         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM           US-BOS-SW2         US-BOS-SW2         ping 10.8.1.1         All Trigger Sourc         07/20/2024 09:09:52 PM         07/20/2024 09:08:54 PM	
US-BOS-SW2 US-BOS-SW2 ping 10.8.1.1 All Trigger Sourc 07/20/2024 09:09:52 PM 07/20/2024 09:08:54 PM	
US-B05-SW3 US-B05-SW4 ping 10.8.1.1 All Trigger Sourc 07/20/2024 09:09:52 PM 07/20/2024 09:08:54 PM	
US-B05-SW4 US-B05-SW4 ping 10.8.1.1 All Trigger Sourc 07/20/2024 09:09:52 PM 07/20/2024 09:08:54 PM	
US-B05-SW1 US-B05-SW3 ping 10.8.1.1 All Trigger Sourc 07/20/2024 09:09:52 PM 07/20/2024 09:08:54 PM	
US-BQS-SW3 US-BQS-SW2 plng 10.8.1.1 All Trigger Sourc 07/20/2024 09:09:52 PM 07/20/2024 09:08:54 PM	

6. Go back to Auto Intent Wizard and click **Next** to move on to the Auto Intent section.

Auto Intent Wizard - Ping Cisco				×
Intent Template	Pre-Decod	•	Auto Intent	
			Intent Based Automation C	Center
Install Intent Template to: Automation	Cookbook ~	② Decoding Settings		
1 Install & Decode Status: In	nstalled			
2 Decode Now Status: Last D	Decoded at 09:09 PM 07/20/2024			G
Output: 8 devices decoded for this in	tent			
Oecode the intent template on domain device	25.		Previous 6 N	lext

### 4.1.3 Auto Intent

The **Auto Intent** section is the final step:

- 1. **Enable for Auto Intent:** Enable the option by clicking the radio button.
- 2. Select Folder: Choose a folder location for the Auto Intent by clicking on Select.
- 3. Auto Intent Description: Add a description, e.g., Latency Check.
- 4. Choose a profile from Select Profiles, or create a new profile using +Add Profile and click OK.
- 5. Click Finish to save and close the Auto Intent Wizard for NIT Ping Cisco.

Auto Intent Wizard - Ping Cisco		×
Intent Template	Pre-Decode	Auto Intent
		Auto Intent Manager
Enable for Auto Intent		
Select Folder: AB		
Auto Intent Description:		
Ping Latency Check	Select Profi	le >
Member this Intent to Auto Intent Profiles:		
1 Profile	4 Select Profiles Search	۹
My Profiles/Ping	D Na	ared Profiles
		Ping
	× 1	
	+Add Pro	ofile
	<b>L</b>	
		Cancel
	-	
1 Enable Auto Intent for this intent template.		Previous 5 Finish

### 4.1.4 Run the Auto intent:

Execute the intent by drawing the targeted device group to map as follows:

- 1. On the desktop, go to the <sup>III</sup> menu located on the top right corner and select **Device Group**.
- 2. In the device group pane, select the predefined group, **Cisco IOS devices**, and click the dropdown menu to open.
- 3. Select **Draw Devices on Map** to add the devices to a new map.
- 4. Close the Device group pane.

Search App Q			🧠 Domain Manage		
Network Map-Based Automation	Intent-Based Automation	Incident & Change	Misc		
Network Analysis	Discovery	Data Mo	odel		
🎲 Network	Oiscover	🔯 Dat	🔀 Data Accuracy Discovery		
😤 One-IP Table	🍪 Network Settings	🔕 Dat	\delta Data Accuracy Resolution		
Sevice Group	🔜 API Server Manager	😫 Ope	😫 Open Driver Manager		
🚴 Site	🔕 Do-Not-Scan	🖭 Hea	alth Report		
🞝 Site Manager	🍓 Network Definition	🍪 Plat	form Validation Manager		
Inventory Report ::: Not Brain	Next-Gen Search Anything and Create	мар Q 📀			
Change Analysis Re	My Device Groups > Cisco IOS devices		😢 Help 😋 푸 🗙 🥑		
Search	Q Search		٩		
Recents		Vendor Mo	del Mgmt IP		
Recents	te Groups Hostname IOS devices (4)				
Recents Cisco I SI Network P	IOS devices (4)	SW1 Cisco 356	0E 172.16.11.205		
Recents	IOS devices (4)	5W1 Cisco 356 5W2 Cisco 356	i0E         172.16.11.205           i0E         172.16.11.206		
Recents Cisco I Cisco I Network P Files D M	IOS devices (4)	5W1 Cisco 356 5W2 Cisco 356 5W3 Cisco 356	ioe         172.16.11.205           ioe         172.16.11.206           ioe         172.16.11.204		
Recents Recents Cisco I Cisco	IOS devices (4)	5W1 Cisco 356 5W2 Cisco 356 5W3 Cisco 356	ioe         172.16.11.205           ioe         172.16.11.206           ioe         172.16.11.204		

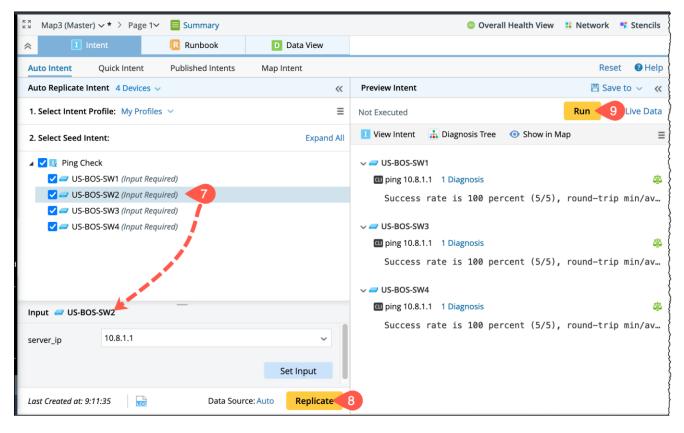
- 5. Open the **Auto Intent** pane under the **Intent** tab and **Select Profile** (**My Profiles**) that you have created in the previous section.
- 6. Check the selection boxes of Intent **Ping Check** and the devices listed underneath.

мар3 (I	Master) 🗸 * 👌 Page	e 1 🗸 📒 Summary				
~	D Data View					
Auto Inten	t Quick Intent	Published Intents	Map Intent	Rese	et	
Auto Repli	cate Intent 4 Device	es 🗸		~	»	
1. Select Intent Profile: My Profiles 5						
2. Select S	eed Intent:		Expa	nd All		
🔰 🗌 🕵 Pi	ing Check					
🗌 📨 US-BOS-SW1 (Input Required)						
🗌 🛹 US-BOS-SW2 (Input Required)						
	US-BOS-SW3 (Input	Required)				
	US-BOS-SW4 (Input	Required)				

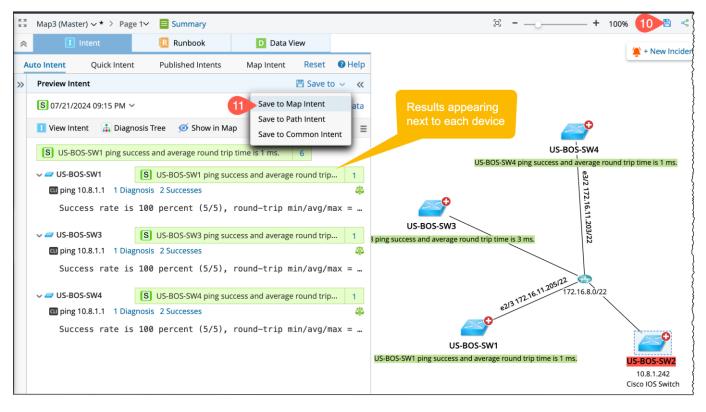
7. Select each device and the **Input** field appears at the bottom of the intent pane.

NOTE: The default value for the destination **server\_ip** will appear for every device. It can be modified as required by clicking **Set Input.** 

- 8. Click **Replicate**, and all the devices will appear under the new **Preview Intent** section.
- 9. Click **Run** to execute the diagnosis.



- 10. The result will appear next to each corresponding device in **Preview Intent** and on **Map**.
- 11. **Save** the Map with all the devices and results for future reference.
- 12. Save to Map Intent to view and run the intent (apart from schedule) as and when needed.



## 4.2 Wrapper Intent as End User Interface

Often, you want to hide an intent implementation from an end user so that any improvement of your intent will not affect how others run the intent. You can use the **follow up intent** to encapsulate a set of intents into a parent intent. This parent intent, sometimes called **wrapper intent**, hides the detailed intent implementation from the end user and can function as a public interface to the end user, which will be accessed as an auto intent or as a chatbot.

This section covers two topics:

- 1. Create a Wrapper Intent
- 2. Execute the wrapper intent

#### **4.2.1 Create Wrapper Intent**

- 1. From the NetBrain Desktop, open the + menu and click **New Intent.**
- 2. In the Network Intent (Edit Mode) window, rename the Intent by clicking on the blue pen icon beside the "Type intent title here..." field.

	Net3rain Next-G	en Search Anyth	ing and Create Map	२ 📀				
Ð	New Map	Rew Device Group	New Data View Template					
Recents	1 New Intent	New Auto	R.					
	New Intent		Intent (Edit Mode)					×
Files Site	New Intent Dashboard	Type desc	ription here	lagnosis Tree	Run	with Live Data	Save 🛛 Ə Help	
Path Dashboard			+ Device			Intent Variables: N	Nanager   Tag: + Add	
I > Intents	Route Check (C							
Chatbot								
						}		h

- 3. Update the intent name to **Device Health Monitoring checks.**
- 4. Click **+Device** to add a device to the Intent,e.g., *US-BOS-SW1*.
- 5. Click 🗉 located beside the add diagnosis buttons, then click **Add Device Diagnosis Block**.

Network Intent (Edit Mode)			×
I Devcie Health Monitori	ing checks 3 Diagnosis Tree	Run with Live Data	Save 🕜 Help 😑
Type description here		یں ایک	ntent Map: Select 🗸
Seed Logic	🗊 Replication Logic 🛕		
😂 + Device 😽 4		Intent Variables: Manager	Tag: + Add 📰
v 🥔 US-BOS-SW1	Type Description here	+ Add Config Diagnosis	CLI Diagnosis 📃
No content has been added.		Add SNMP Diagno Add API Diagnosis Add Ping Diagnosis Add Trace Diagnos Add Device Diagno Add Action Duplicate Section Delete	s

6. Rename Diagnosis 1 to **Device monitoring.** 

Add Device Diagnosis Block	×
Diagnosis1 +	
Name: Device monitoring 6	
Description:	
Loop Table Rows	
	,⊥

- 7. In the **If** condition, select **True** from the Select Variable dropdown.
  - NOTE: Using **True** as the variable selection tells the NetBrain to always call the follow-up action without performing any diagnosis. As we always want to call the follow-up action, the True option is correct.
- 8. Remove the **Diagnosis Message** by clicking on the menu in the top right of it's section, then click **Delete.**

Add Device Diagnosis Block		×
Diagnosis1 +		
Name: Device monitoring 6 Description:		
<ul> <li>Loop Table Rows</li> <li>If</li> </ul>		
A US-BOS-SW1 True 7 ~ B Select Variable ~	圓	
✓ Then		
<ul> <li>Diagnosis Message:</li> <li>\$intf is down</li> <li>Set Status Code for Device:</li> <li>Set Status Code for Intent:</li> </ul>	□ Save to Incident Pop up Delete 8	
Add Logic ~ + Add Elself + Add Else		
	Cancel OK	4

- 9. Add a follow-up action by selecting **Follow-up Intent** from the **Add logic** dropdown.
- 10. After the Follow-up Intent option is added, click the **Network Intent** link to select the Network Intent in a new window, **Follow-up Intents**.

Add Device Diagno	osis Block X	
Diagnosis1	1 +	
Name: Description:	Device monitoring	
Create Diag	nosis Message	
Draw Map Send Email		
Follow-up In Advanced Add Logic	9   B   Select Variable	
	∨ Then	
+ Add Elself	+ Add Else	

#### 11. Click Select Intent (Standalone).

- a) In the Select Intents dialog, navigate to the intents you have created in Chapter 3 and checkin all.
- b) Click OK.

Follow-up Intents	×
0 Follow-up Intents: + Follow-up v Select Intent (St Select Intent Te Select Intents fr Select Intent Te Select Intent Te	tandalone) 11 emplate rom ADT emplates via ADT
	Select Intents ×
Prune other follow-up intents	Search         All amy (7) cmd (2) L1 - Config Change (2) L >         All amy (7) cmd (2) L1 - Config Change (2) L >         Ajay         Ajaet         Arun Belide         CPU Usage Check (Cisco IOS)         CPU Usage Check (Cisco IOS)         Interface Status Check (Cisco IOS)         Show Version (Cisco IOS)         Show Intents Serving as Template Only         Show Published Intents Only
	Can b OK

- c) Ensure the correct intents are added here in Follow-up Intents.
- d) Click Save.

12. Click **OK** in the Add Device Diagnosis Block to save all the data and close the window.

Add Device Diagnosis Block	×
Diagnosis1 +	
Name: Device monitoring	
Description:	
□ Loop Table Rows ~ If	
A 🥔 US-BOS-SW1	
B Select Variable ~	Î
∽ Then	
Follow-up Intent:      Network Intent     Current Intent (Self)     Stop      Intent: Check uptime (Cisco IOS)	=
Intent: CPU Usage Check (Cisco IOS) Intent: Interface Status Check (Cisco IOS)	
Intent: Show Version (Cisco IOS)	
Add Logic 🗸	
+ Add Elself + Add Else	
Canc <mark>12</mark>	ок 🥻

13. Back in the Network Intent (Edit Mode) dialog, click **Save**.

Network Intent (Edit Mode)			×
<b>I</b> Devcie Health Monitoring checks	h Diagnosis Tree	Run with Live Data	13 Save ③ Help Ξ
Type description here			💦 Intent Map: Select 🗸
I Seed Logic	ogic 🛕		
😂 + Device		Intent Variables: 🕅	Nanager 🕴 Tag: + Add 🔳
∼ 🚄 US-BOS-SW1	Type Description here	+ Add Config Diagnosis	the second
Solution Contraction Contractica Contracti	Type Description here		Edit Diagnosis 🛛 🗏
			h.

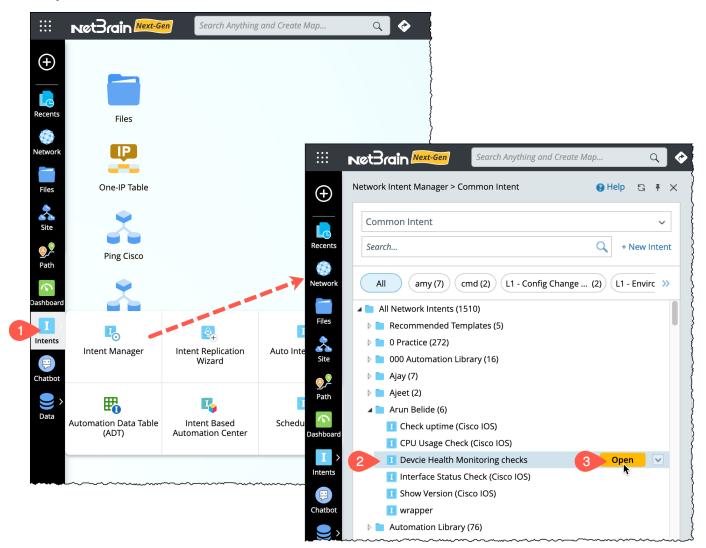
- 14. Browse to the **All Network Intents > [Your folder location].**
- 15. Click **Save** to save and close the window.

Save Netw	vork Intent	×
Search.		Q
🔺 🖿 A	ll Network Intents	
	Recommended Templates	
▷ 🖿	0 Practice	
▷ 🖿	000 Automation Library	
	Ajay	
	Ajeet	
14	Arun Belide	
	Automation Library	
▷ 🗖	Automation Library Kunal	
▷ 🗖	Automation Library PKG11	
D 🗖	Automation Library Ty	
	СК	
	Cloud Assessment Reference Library	
	Debug	
	DrDave	
	Edgar Library	
Name:	Devcie Health Monitoring checks	
	Cance 1	5 Save

### 4.2.2 Execute Wrapper Intent

Let us validate and execute the wrapper intent created in the previous section.

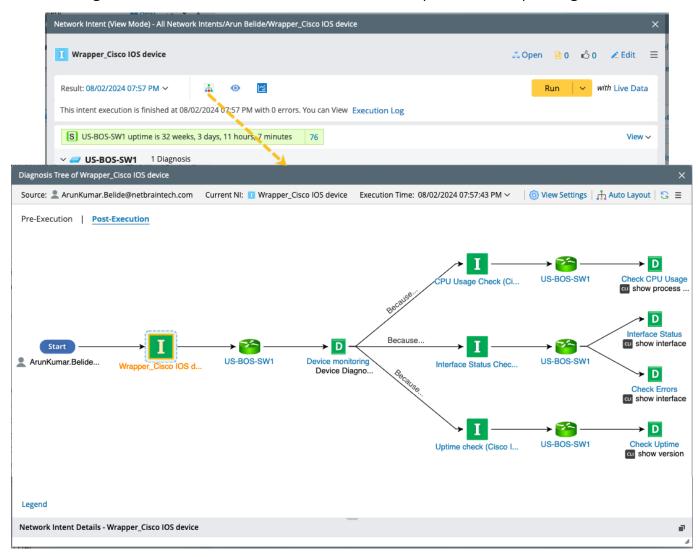
- 1. In the sidebar of the End User Desktop, click **Intents** > **Intent Manager**.
- 2. In the Intent Manager, navigate to the wrapper intent **Device Health Monitoring checks** created in the previous section.
- 3. Click **Open** to view the intent in view mode.



- 4. Click **Run** to execute all the configured follow up intents.
- 5. Results will be displayed with the status, message and number of executed follow up intents.
- 6. Click the number in the diagnosis message to view the messages by intent.

twork Intent (View Mode) - All Network Inte	nts/Arun Belide/Devci	e Health Monitoring checks		
Devcie Health Monitoring checks		Click to view detailed	💑 Open 📄 🛛 🔥 0 🛛 差 Edit	2
Result: 07/21/2024 07:22 PM 🗸	•	messages intent wise	Run 🗸 with Live D	)ata
his intent execution is finished at 07/21/2024 07	:22 PM with 0 errors. Yo	ou can View xecution Log		
S US-BOS-SW1 uptime is 30 weeks, 5 days, 1	0 hours, 32 minutes	76	Vie	ew
V 🥔 US-BOS-SW1 1 Diagnosis	1		Number of follow up intents	
🍣 Device Diagnosis Block		1 Diagnosis 4 Follow-up NI	executed	
	i			
	Diagnosis Messag	ge (07/21/2024 07:22 PM)	×	
		ria Haalth Manitaring chacks		
	With 4 Follow-up	cie Health Monitoring checks 🗸 🗸	✓ 76 Successes ✓ 38 Messages	
	View by Inter	View by Device		
	✓ Ⅰ Devcie	Health Monitoring checks		
	🗸 I Check u	uptime (Cisco IOS)		
		age Check (Cisco IOS)	2 Successes 1 Message	
		tent Level Status Code: 1 Success		
		S US-BOS-SW1 CPU utilization is norr	nal	
		nies Level Chetur Cader - 4 Success - 4 M		
		evice Level Status Code: 1 Success 1 Me	essage	
	✓ Ⅰ Interface	ce Status Check (Cisco IOS)	72 Successes 36 Messages	
	√s In	tent Level Status Code: 36 Successes		
		S On US-BOS-SW1, Interface Etherne	t0/0 is UP	
		S On US-BOS-SW1, Interface Etherne	t0/1 is UP	
		S On US-BOS-SW1, Interface Etherne	t0/2 is UP	
		S On US-BOS-SW1, Interface Etherne	t0/3 is UP	
		S On US-BOS-SW1, Interface Etherne		

7. View the results in the diagnosis tree to check whether all the follow up intents are executed.Click the diagnosis tree button in the results window to open the corresponding view:



## 4.3 Run intent with Chatbot

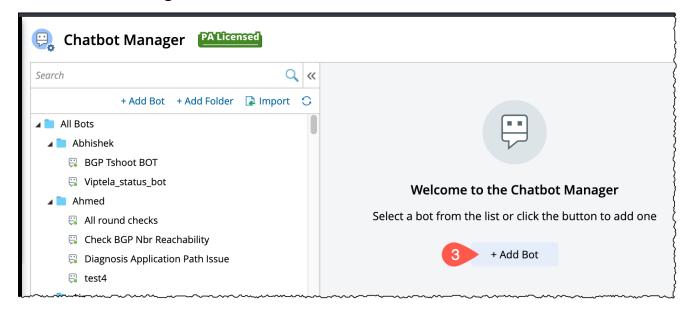
Chatbot allows power users to build interactive chatbots, so end users can execute intent-based automation to solve real-world challenges without accessing NetBrain system UI. Building a chatbot flow is straightforward.

### 4.3.1 Create the Netbrain Chatbot

- 1. In the Upper-Left corner of the desktop, click  $\stackrel{\rm III}{=}$
- 2. Select Chatbot Manager under the Intent-Based Automation tab.

::: <b>1</b> etBrain	ł
Search App	🧠 Domain Management
Network Map-Based Automation	Intent-Based Automation Incident 8
Intent	Execute Intent
🎝 Intent Manager	🛂 Intent Based Automation Center
🔩 Intent Cluster Manager	🖏 Triggered Automation Manager
🖽 Automation Data Table (ADT)	💩 Preventive Automation Manager
🖼 Task Variable Manager	Schedule Automation
	🔎 🥄 Chatbot Manager 🛛 📌
	Preventive Automation Console

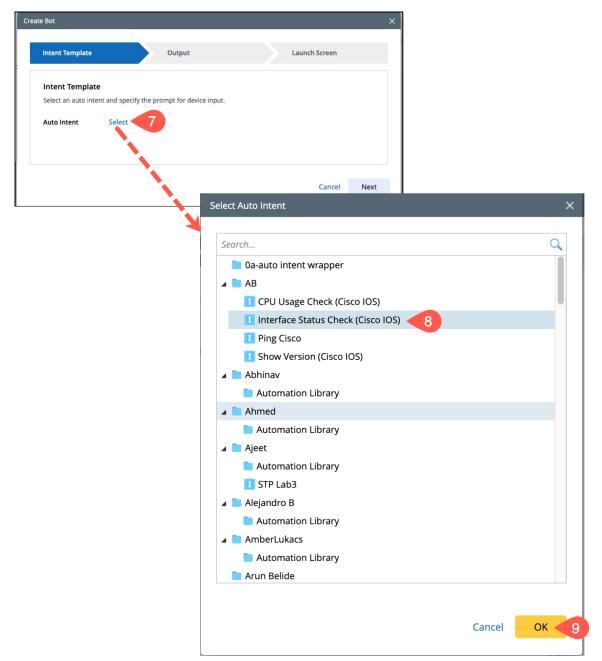
3. In the Chatbot Manager, click + Add Bot.



- 4. Name the bot **Device Health Monitor.**
- 5. Location: To choose a location for the bot, click **All Bot**.
  - a) In the dialog, **Select Folder for Chatbot**, select All Bots or create a folder under it using the option **New Folder** from the dropdown menu.
  - b) Click **OK.**
- 6. Click **Start** to begin the **Chatbot creation wizard**.

e Bot	×	Select Folder for Chatbot
t Name	Device Health Monitor 4	All Bots
cation 5	All Bots	<ul> <li>Abhishek</li> <li>Ahmed</li> <li>Ajay</li> </ul>
lect Method	Intent Template 🗸 🗸	▲ ▲ Ajeet
eator	ArunKumar.Belide@netbraintech.com 🖉	<ul> <li>New Folder</li> <li>Arun</li> <li>DoctorDave</li> <li>EMEA - DT</li> <li>Giorgio</li> <li>Gopalakrishna</li> <li>Joey</li> </ul>
	A A A A A A A A A A A A A A A A A A A	Cancel
reate Bot		Cancel
reate Bot Intent To	emplate	
Intent Te	emplate Output Template	×
Intent Te		X Launch Screen
Intent Te	Template n auto intent and specify the prompt for device input	X Launch Screen
Intent To Intent Select ar	Template n auto intent and specify the prompt for device input	X Launch Screen

- 7. In the first ribbon of the **Create Bot Wizard (Intent Template)**, click **Select** located next to **Auto Intent.**
- 8. From the Select Auto intent dialog, choose the intent Interface Status Check.
- 9. Click **OK** to select and close the window.



- 10. Modify the **Input Prompt** as required or retain the default prompt.
- 11. Check the checkbox, **Allow multiple selection**, to choose multiple device selection.

NOTE: This ensures to run intent multiple times (once per unique device).

12. Click **Next** to go to the next ribbon Output.

eate Bot		>
Intent Template	Output	Launch Screen
Intent Templat	e	
Select an auto int	ent and specify the prompt for device input.	
Auto Intent	/AB/Interface Status Check (Cisco IOS)	
Input Prompt	Please select a device: 10	Allow multiple selection
		Cancella Next

13. In the **Output** ribbon, review the potential output (or click the Pencil icon to edit it), then click **Next.** 

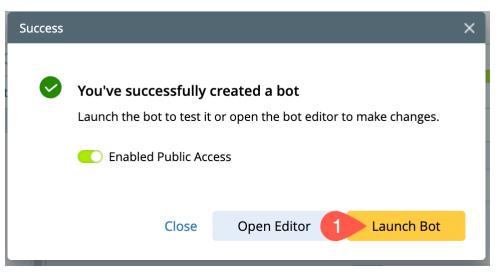
Create Bot				×
Intent Te	mplate	Output	Launch Screen	1
	" <u>\$intent_name</u> " exe	output_map	output options for drill-down.	
	+ Add		Cancel Previous	13 Next

- 14. In the Launch Screen, review the default details (make any desired edits).
- 15. Click **OK** to complete the configuration of the chatbot.
- 16. A success prompt appears on the screen upon bot creation.

Intent Template	Output	Launch Screen
Bot Launch Scree	n	
Personalize the logo	and description on the launch screen that is sh	hown to users after they log in.
Logo	Supported file types a Max file size is 5M. Upload Reset	are .jpg, .jpeg, .gif and .png.
Description	Large <sub>↓</sub> <u>A</u> B <i>I</i> <u>U</u> ⊨ ≔	€ =
	Welcome to NetBra	in Chatbot!
	This bot will execute automation -	Device Health Monitor
	Preview Launch Screen	
Greeting Message	Hello! What do you want to do?	
		Cancel Previous O
Success		×
	've successfully created a bot	
📀 You		make changes.
_	ich the bot to test it or open the bot editor to n	

## 4.3.2 Use Chatbot

1. Click Launch Bot from the Succes prompt window of the chatbot creation.



#### (or)

In the chatbot Manager, click the **Launch** button located on the top right corner of the window.

🖳 Chatbot Manager 🕰 🕮			
Search Q « + Add Bot + Add Folder 🕞 Import 🗘	Last Modified on: 23/7/2024, 9:33:39 pm      Enabled      Launch	Edit	
All Bots	Usage: Used 0 times by 0 users	+ 100%	
<ul><li>Iun liang</li><li>Sophia Wang</li></ul>	► Start		
▲ ▲ AB □ Device Health Monitor			
<ul><li>Abhinav</li><li>Alejandro B</li></ul>	NIT Interface Status Check (Cis		
⊿ 💼 cary 🐺 test			
Chris P Cookbook			
BGP_TS			
Fanny      What's Change and Failover Ch			

2. In the newly opened browser tab, click **START** to launch the Chatbot.

Welcome to NetBrain Chatboર્ષે!	
This bot will execute automation - Device Health Monitor	
Created by ArunKumar.Belide@netbraintech.com	
START	
🗌 Don't show me again	

- 3. Click the Intent Interface Status Check (Cisco IOS).
- 4. Under **Please Select a Device**, click *Select or type to enter input* field and select the number of devices from the drop-down as required. E.g., select two devices, *US-BOS-SW1* and *US-BOS-SW2*.
- 5. Click **Submit** to execute the intent.

	Device Health Monitor Session 2 ∨		Ę	(j
Hello! What do you want to do?  Hello! What do you want to do?  Interface Status Check (Cisco IOS)  04:13 PM				
		Interface Status Check (Cisco IOS) 04:15 PM	AB	
Please select a device:	Submit 5			
04:				
Type or press Tab key to see options			Send D	

6. The chatbot will execute the intent and provide the diagnosis output.

NOTE: Intent Diagnosis Message indicates the ethernet is UP.

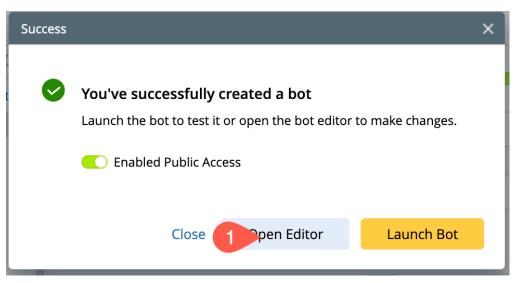
NOTE: Map and Diagnosis tree links to view the error details if any.

Device Health Monito	
Session 3 🗸	
Interface Status Check (Cisco IOS)	
04:19 PM	
	Interface Status Check (Cisco IOS)
	04:19 PM
Please select a device:	
Select V Submit	
04:19 PM	
	US-BOS-SW1
	04:19 PM
"Interface Status Check (Cisco IOS) US-BOS-SW1 1" executed on dev	vice US-BOS-SW1.
<u>0</u> errors found:	
On US-BOS-SW1, Interface Ethernet0/0 is UP	Ethernet status is
On US-BOS-SW1, Interface Ethernet0/1 is UP	UP
On US-BOS-SW1, Interface Ethernet0/2 is UP	
On US-BOS-SW1, Interface Ethernet0/3 is UP On US-BOS-SW1, Interface Ethernet1/0 is UP	
On US-BOS-SW1, Interface Ethernet1/0 is UP	
On US-BOS-SW1, Interface Ethernet1/2 is UP	
On US-BOS-SW1, Interface Ethernet1/3 is UP	
On US-BOS-SW1, Interface Ethernet2/0 is UP	
On US-BOS-SW1, Interface Ethernet2/1 is UP	
On US-BOS-SW1, Interface Ethernet2/2 is UP	
On US-BOS-SW1, Interface Ethernet2/3 is UP	
On US-BOS-SW1, Interface Loopback0 is UP	
On US-BOS-SW1, Interface Loopback1 is UP	
On US-BOS-SW1, Interface Loopback2 is UP	
On US-BOS-SW1, Interface Port-channel35 is UP On US-BOS-SW1, Interface Vlan100 is UP	
On US-BOS-SW1, Interface Vlan100 is UP	
Errors on this device US-BOS-SW1 have not increased	
View Map: Intent Output Map	Diagnosis tree
View Diagnosis Tree: Diagnosis Tree links	
This is the last step. You can type "restart" to start over.	
04:19 PM	
for shortcut commands	Send
TOT SHOTCH CONTINUITUS	Sena

7. Close all browser tabs except for the **NetBrain desktop.** 

## 4.3.3 Add Auto Intent to Chatbot

1. Click Launch Bot from the Succes prompt window of the chatbot creation.



#### (or)

Click **Edit** located on the top right corner of the chatbot manager window.

🖳 Chatbot Manager 🛛 🕅	red
Search Q «	🚯 Last Modified on: 23/7/2024, 9:33:39 pm 🌔 Enabled 🗠 Launch 🚺 Edit ?
+ Add Bot 🛛 + Add Folder 🛛 🔒 Import 🙄	
All Bots	Usage: Used 0 times by 0 users [1] + 100%
a 📄 OPrictice	
Þ 📄 Jun liang	
🖻 📄 Sophia Wang	► Start
A 🖿 AB	
😳 Device Health Monitor	•
Abhinav	
Alejandro B	Interface Status Check (Cis
🖌 📄 cary	
😳 test	
📔 Chris P	
Cookbook	
😳 BGP_TS	
DukeZhang	
🖌 🦲 Fanny	
🖳 What's Change and Failover Ch	

- 2. In the **Chatbot Manager (Visual Editor)**, click and drag the element **Intent Template** into the editor.
- 3. In the right pane, click **Select** in the Intent Template editor to open the **Select Intent Template** dialog.
- 4. In the **Select Intent Template**, navigate to your required intent, e.g., **Show Version**.
- 5. Click **OK**.

🖳 Chatt	oot Manager (PA Licensed)	
Device He	ealth Monitor ∠	🔓 🦲 Enabled  ස් 😳 Close 🛛 Launch Save 🕜
		<ul> <li>أحم المراجع</li> <li>أحم الممراجع</li> <li>أحم المراجع</li> <li>أحم ال</li></ul>
I Intent	▶ Start	אזא Intent Template Node Input Node Output
Intent Template	Interface Status Check (Cis	Intent Template
Automation Data Table		Select 3
000		Chat Preview 😗
Prompt		Select Intent Template X Select Intent Template from:  All Intents  Installed Intents
Condition		Type:     Common Intent     ~     Filter by:     Device-based     ~     Search
·		All         amy (9)         cmd (2)         L1 - Config Change (2)         L1 - Environment C (1)         L1         >>
		Arun Belide     CPU Usage Check (Cisco IOS)     Interface Status Check (Cisco IOS)     Show Version (Cisco IOS)
		Wrapper_Cisco IOS device
		Automation Library
		Cancel OK 5

6. Under **Replicated Device Candidates**, click "--- **Select ---**" and:

Node Input Node Output	
ntent Template	
All Network Intents/Arun Belide/Show Version (Cisco IOS)	Select
Replicated Device Candidates  📵	
Select	~
User Input Mode  🟮	
Multiple selection	~
Allow user custom input	
Input Prompt	
Please provide input:	
Candidate Value Source 🚯	
Select C	~

- a) User Input Mode: Choose **Multiple Selection** from the drop down menu.
- b) **Input Prompt**: Keep the default (**Please provide input).** Or modify it to give a hint for the user to provide input.
- c) **Candidate Value Source**: These values will appear as a drop-down menu for input selection, choose the option **Manual Input** and add the list of devices.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		
Candidate Value Source 🔋		
Manual Input 🗸 🗸	US-BOS-CW01-01, US-BOS-CW01-02,	
List of devices added	Cancel OK	

- d) Click **OK**.
- 7. Click the **Start** element.

8. Hover the mouse on the element, click the blue circle and click-drag the resulting arrow connecting it to **Show Version (Cisco IOS).** 

E Chatbot Manager PA Licensed			
Device He	alth Monitor 🚄		
Intent Intent Intent Template	NIT Interface Status Check (Cis		
Automation Data Table	NIT Show Version (Cisco IOS)		

- 9. Click Save, then Launch.
- 10. In the newly opened browser tab, click **START** to launch the Chatbot. The final chatbot with two intents will be as follows:

	Device Health Monitor				í	
	Session 8 🗸					
	Hello! What do you want to do?					
	Interface Status Check (Cisco IOS)	Show Version (Cisco IOS)				
	10:45 PM					
Type or	press Tab key to see options			Send 🗦	>	

# 5 Network Assessment and Document: Essential

In the last three chapters, you learn how to create an intent and replicate an intent to auto intents, which can be used by the end users for troubleshooting. In this chapter, you will learn how to use the intents for the network assessment across the whole network.

Network Assessments have always been a 'necessary evil' to satisfy the need to understand the network and its vulnerabilities and bottlenecks. However, this meant costly projects that spanned months, only to get a rudimentary understanding of your infrastructure as it existed at a certain point in time. The problem is that they quickly become out of date with today's constantly changing networks.

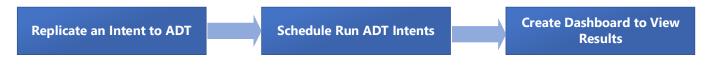
NetBrain intent-based no-code automation technology provides an ideal platform for Network Assessment, transferring it from a static document to a dynamic continuous health check. In this chapter, you will learn how to apply an intent you have created in the last three chapters to the network assessment across the whole network. You will learn an important concept, **Automation Data Table** (ADT), a global table to manage your network assets, such as core routers and critical applications, and automations associated with these objects. ADT and NI are essential for the network assessment. Some important use cases of the network assessments will be covered in the next three chapters:

- Security assessment
- Configuration drift
- Failover assessment

In this chapter, we will use the following examples to illustrate the essential concepts and workflows of the network assessment:

- Create an ADT for your network devices and the performance data, such as CPU.
- Create a report for all down interfaces within your network. This report can help your company assess the capacity of your network.

The workflow of the network assessment will be as follows:



# 5.1 Document Your Network Inventory and Performance

You will start your first network assessment with a simple example: monitoring the CPU usage of the network devices, which can offer real-time insights into the performance of your network devices. By proactively managing CPU utilization, you can detect potential issues early and implement preventive measures. High CPU usage is often a symptom of performance bottlenecks.

In this section, you will learn how to diagnose high CPU usage within your network ecosystem, enabling you to address issues before they affect overall network performance.

You will create an ADT as follows, including the device's built-in properties, the replicated intents, and the intent status code:

🐻 СР	U Performance Check	Table Builde	Last Updated at: 06/14/2	2024 06:59 PM 🛛 🍳 Rebuild Tab	le		Add Data Manually 🗸 📃 🧃
Descri	ption: Type description here						
Items:	24 Rows 7 Columns					Search Q	Advanced Filter: Undefined
No.	Device	S Hostname	s Vendor	s Model	CPU Performance	S Intent Status Code	( Last Execution Time
	US-BOS-SW5	US-BOS-SW5	Cisco	3560E	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:3
2	US-BOS-SW4	US-BOS-SW4	Cisco	3560E	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:3
3	US-BOS-SW3	US-BOS-SW3	Cisco	3560E	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:3
L.	US-BOS-SW2	US-BOS-SW2	Cisco	3560E	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:3
	US-BOS-SW1	US-BOS-SW1	Cisco	3560E	CPU Performance Check US-BOS 👁	Low CPU: 0	06/14/2024 06:44:3
5	US-BOS-R2	US-BOS-R2	Cisco	CGS-MGS-AGS	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:3
,	US-BOS-R1	US-BOS-R1	Cisco	CGS-MGS-AGS	CPU Performance Check US-BOS 👁	Low CPU: 0	06/14/2024 06:44:3
3	BST_POP2	BST_POP2	Cisco	2621	CPU Performance Check BST_PO •	High 5 second average	06/14/2024 06:44:4
9	BSTX	BSTX	Cisco	2811	CPU Performance Check BSTX O	Low CPU: 1	06/14/2024 06:44:3
0	BST, POP1	BST, POP1	Cisco	2514	CPU Performance Check BST, PO 💿	Low CPU: 4	06/14/2024 06:44:3
1	BOS-cEdge-01	BOS-cEdge-01	Cisco	CSR1000V	CPU Performance Check BOS-cE 💿	Low CPU: 1	06/14/2024 06:44:3
2	BJ_core_3550	BJ_core_3550	Cisco	catalyst355024	CPU Performance Check BJ_core 💿	Low CPU: 3	06/14/2024 06:44:3
3	BJ_L2_Core_6	BJ_L2_Core_6	Cisco	catalyst356048TS	CPU Performance Check BJ_L2_C •	Low CPU: 4	06/14/2024 06:44:4
4	BJ_L2_Core_5	BJ_L2_Core_5	Cisco	catalyst356048TS	CPU Performance Check BJ_L2_C •	Low CPU: 5	06/14/2024 06:44:3
5	BJ_L2_Core_4	BJ_L2_Core_4	Cisco	catalyst37xxStack	CPU Performance Check BJ_L2_C •	High 5 second average	06/14/2024 06:44:4
6	BJ_L2_Core_3	BJ_L2_Core_3	Cisco	catalyst37xxStack	CPU Performance Check BJ_L2_C •	High 5 second average	06/14/2024 06:44:3

This section includes the following main steps:

- <u>Prerequisites</u>
- <u>Create Network Intent</u>
- Use Intent Replication Wizard
- <u>Run the Intent</u>
- Export ADT to CSV file
- <u>Create Intent Dashboard for ADT Automation Column</u>

## **5.1.1 Prerequisites**

The first step is to create a device group to be assessed. This device group will be used during the **Intent Replication Wizard** and for creating the **ADT** for the base table using the **Devices of Device Group** method. If the device group already exists, you can skip these steps.

Here, you will create a device group for all the Cisco devices. You can create this device group by **Dynamic Search**: set the **Device Type** to be Cisco IOS switch or router.

niet3rain	Next-Gen Search Anything	and Create Map							
New Map	New Device Group	New Data View Template							
Device Gr	oup Properties						×		
< Na	me: Cisco Devices (for CPU per	formance) 2							
	ion: Input description here		Select Devices Select Devices by:	Device Type 5	Device Group	<ul> <li>Site</li> </ul>			27 Devices Selected
Nev			All Device Types		✓ Search		۹		BJPOP
Locat	ion: Sachin_TW 3	~	Hostname		Mgmt IP	Vendor	Model		BJ_Acc_SW1
Devices	and Interfaces		!@#\$%^&*()=+ ##################################		172.25.52.9 172.16.101.75	Cisco Amazon	3560E  EC2 Instance		BJ_Acc_SW2-bbb-eee-ii-JJJJ-II1-NNN BJ_Acc_SW6
+ Stati	c ~ + Dynamic Search ~	+ Exclude ~	(eni-0013dbbc77		172.10.101.75	Amazon	AWS Unattached Netwo		BJ_Acc_Sw4
Static De		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	(eni-01fece0b34	6dbd9fa)		Amazon	AW5 Unattached Netwo	>	BJ_Acc_Sw4-bbb-eee-ii-JJJJ-II1-NNN.
Static In	lentace		😊 (eni-0253f49d77			Amazon	AWS Unattached Netwo	»	BJ_Dis_SW2
			eni-050a740c89			Amazon	AWS Unattached Netwo		BJ_L2_Core_3
			(eni-051b68b343)			Amazon	AWS Unattached Netwo	<	BJ_L2_Core_4
			😊 (eni-0520365cb3			Amazon	AWS Unattached Netwo	~~	BJ_L2_Core_5
			🤤 (eni-07536c7e24			Amazon	AWS Unattached Netwo		BJ_L2_Core_6
			💿 (eni-077a811131			Amazon	AWS Unattached Netwo		BJ_L2_test_1
			🤤 (eni-07827c73a2	77a882c)		Amazon	AWS Unattached Netwo		
			👳 (eni-08877fc6db	76a5a4d)		Amazon	AWS Unattached Netwo		BJ_core_3550
			🤤 (eni-09c29b7599	291837f)		Amazon	AWS Unattached Netwo		BOS-N9K-L3OUT
			短 (eni-0a29d24078	3b271628)		Amazon	AWS Unattached Netwo		BOS-cEdge-01
			👳 (eni-0ad0e788e4	I1ccb3a0)		Amazon	AWS Unattached Netwo 🗣		A RST
									Cancel
vice Group >	Shared Device Group	ips > Sachin_T	W > Cisco Devices	(for CPU per	formance)		😮 Help 😋 🖣	×	
earch		Q Searc	h					Q	
📄 My Devi	ce Groups	Hos	tname	Vendor	Model		Mgmt IP		
-	etwork Devices (449) EC2 Devices (118)	<i>2</i> E	BJ_Acc_SW1	Cisco	catalyst	295	172.24.101.21	-	
	Device Groups	<i>—</i> E	BJ_Acc_SW2-bb	Cisco	WS-C29	50-24	172.24.101.22		
🖌 📄 Sach	in_TW	<b>e</b>	J_Acc_Sw4	Cisco	catalyst	295	172.24.101.24		
S Ci	sco Devices (for CPU	p 🥥 E	J_Acc_Sw4-bbb	Cisco	WS-C29	50-24	172.24.101.24		
≬ 📄 AG_I(	DS_Nexus		J_Acc_SW6	Cisco	WS-C20	50-24	172.24.101.26		

WS-C2950-24 172.24.101.11

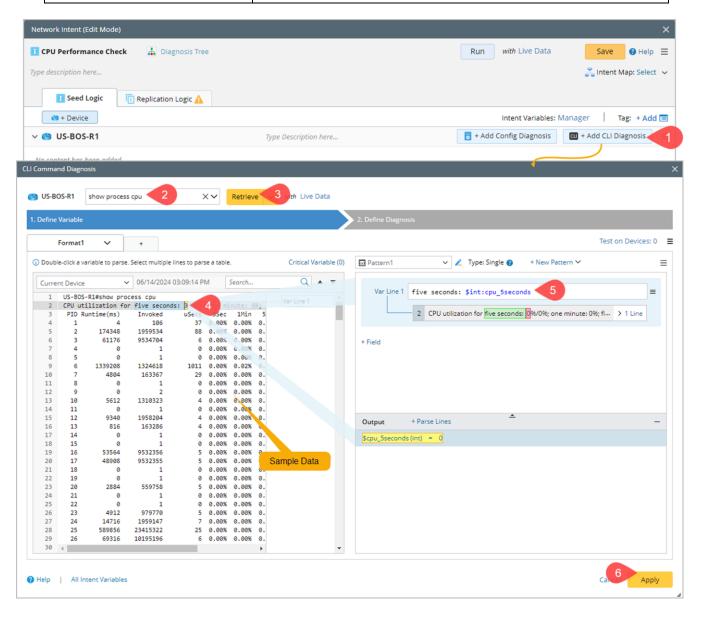
WS-C3550-24 172.24.36.1

### 5.1.2 Create Network Intent

You will reuse the network intent you created in Chapter 3. Let us recap the key steps to create an intent:

• Select a seed device and add a CLI Diagnosis:

Command	show process cpu
Line Pattern	Auto Pattern - Single
Var Line 1	Five seconds: <i>\$int:cpu_5seconds</i>



### • Define the diagnosis

2. Define Diagnosis	
Add Note Add Diagnosis	also click a variable on the left to add automation.
Name: CheckCPU usage	Anchor: \$cpu_5seconds
Type description of the diagnosis	
Loop Table Rows     VIf	, I
A 👩 US-BOS-R1 🥑 Current 🗸	
cpu_5seconds v Greater than v 5	<ul> <li>↓</li> </ul>
B Select Variable V	
∨ Then	
Diagnosis Message:  High 5 second average CPU: \$cpu_5seconds%	Save to Incident 📃
☑ Ingr 5 second sec	
Error V High 5 second average CPU: \$cpu_5seconds%	
Set Status Code for Intent:     High 5 second average CPU: \$cpu_5seconds%	
	i
Add Logic~	
Diagnosis Message:	Delete
✓ Low CPU: \$cpu_5seconds	
Set Status Code for Device:	
Success  Low CPU: \$cpu_5seconds	
Success      Low CPU: \$cpu_5seconds	
Add Logic -	i -
+ Add Elself	
	Cancel Apply

## 5.1.3 Use Intent Replication Wizard

In this section, you will use the **Intent Replication Wizard** to replicate the seed intent to a set of devices (of a device group). The replicated intents will be added to the Automation Column in the ADT table. The New ADT will be created.

- 1. Edit the intent. From the  $\equiv$  menu, open the **Intent Replication Wizard**.
- 2. In the **Seed Intent** tab, check for your NI (*CPU Performance Check*) and then click **Next** to go to the **Define ADT** tab.
- 3. Define ADT.
  - a) In the Define ADT tab, click **Create a New ADT** option to create a new ADT using Device Group.

Intent Replication Wizard - CPL	J Performance Che	ck				×
Seed Intent	3	Define ADT		Replication Settings	Replicate Intent	
a Create a Ne	ew ADT			Use an Existing ADT		
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	ᠬᠬᠬᡔ᠆᠆᠇᠊ᠵᠵ	^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	anner - mar an		

- i. Specify the ADT table name as **CPU Performance Check**. The description is optional.
- ii. Select the **Location** where you want to save the table.
- iii. From the **Device Groups** dropdown, click **Select Device Group** and then select a device group from your specific folder.
- iv. Click the **Additional Column** dropdown and select the column for ADT (e.g., Hostname, Vendor, and Model).
- v. Click Save.

New ADT by D	evice Group		×
Name:	CPU Performance Check	ocation:	Sachin i 🗸
Description:			Select Device Groups X
	New Device Group		Search
	Column Data	Column	Rajendar Reddy Vadhira(DG)
1	Device	Device	▶ □ ■ Raju
	5 Device Group	Device	D 📄 Rakesh
	S Hostname	Hostna	⊳ 🗋 rmj
	s Mgmt IP	Mgmt I	Sachin
	s Mgmt Interface	Mgmt I	Sachin_TW Sachin_TW Sandeep - Cisco,Palo,CP,Arista,Ju
	5 Device Type	Device	Suneet
	S Vendor	Vendor	Sunil
	s Model	Model	Support Automation Demo
	s Software Version	Softwar	▷ 🗋 Suresh_VI 🗸
	s Serial Number	Serial N	
	s Site	Site	Cancel OK
	s Location	Location	
	s ESXI Host	ESXI Hos	
		LOXITIOS	•
•			Cancel Save

- b) In the **Replicate Intent** section, enter the Intent group as an Intent column group name.
- c) Click **Next** to go to the **Replicate Intent** tab.

- 4. Replication Settings.
  - a) In the **Intent Qualification** section, click the **Select** link to add Device Group.
  - b) Add the device group.
  - c) Click **Next** to go to the **Replicate Intent** tab.

		Replication Settings
		Add Device Group
tent Qualification:   via C  efine Macro Variables and F  em: 1  Seed Device  US-BOS-R1	Device Groups/Sites: Select a via Dynamic Select Device Groups Select Sites New Device Group Seed Command show process cpu	

- 5. Replicate Intent:
  - a) In the **ADT Columns** section, you can see the **Column Name** as **Replicated Intent**. You can change this name to **CPU Performance**.
  - b) Add more columns by ticking the checkboxes in the **Additional Columns** dropdown. Add columns, **Intent Status Code**, and **Last Execution Time**.
  - c) Click **Save and Replicate** to save all the settings.
  - d) You can see the **Open Output ADT** option after the successful submission of the replication request. Click **Open Output ADT** to check the replicated Intent column in the ADT Manager.

ent Replication Wizard - CPU Performance Check			>
Seed Intent	Define ADT	Replication Settings	5 Replicate Intent
ADT Columns:			Additional Columns ~
Column Data	Column Name	Tag	Replicated Intent
Replicated Intent	a CPU Performance	0 tags	🗌 Intent Message
s Intent Status Code	Intent Status Code		Intent Status Code
Last Execution Time	Last Execution Time		Device Status Code
			Intent Devices
			Intent Map Intent CLI Comma
			∠ Intent CLI Comma
		Replication Request submitted a	C Save and Replicate t: 08/10/2024 01:49 P C Open Output ADT
Selection Mode: Device-based Replication, ADT:	CPU Performance Check, 0 Macro Variables	5.	Previous Finish

The table will now be populated with devices and the replicated Intents (**CPU Performance**).

- 6. Review the new Intent columns.
  - NOTE: All the columns related to the Intent results are empty because the Intents are not executed yet.

👪 CF	U Performance Check	Table Builder	Last Updated at: 08/10/2024	01:49 PM 🛛 🍳 Rebuild Table			Add Data Manually 🗸 🗧
Descr	iption: Type description here						
Items	: 24 Rows 9 Columns					Search Q	Y Advanced Filter: Undefined
No.	Device	5 Hostname	S Vendor	5 Model	CPU Performance	S Intent Status Code	CLast Execution Time
	US-BOS-SW5	US-BOS-SW5	Cisco	3560E	CPU Performance Check US-BOS 👁		
2	US-BOS-SW4	US-BOS-SW4	Cisco	3560E	CPU Performance Check US-BOS 👁		
	US-BOS-SW3	US-BOS-SW3	Cisco	3560E	CPU Performance Check US-BOS 👁		
	US-BOS-SW2	US-BOS-SW2	Cisco	3560E	CPU Performance Check US-BOS 👁		
	US-BOS-SW1	US-BOS-SW1	Cisco	3560E	CPU Performance Check US-BOS 💿		
	US-BOS-R2	US-BOS-R2	Cisco	CGS-MGS-AGS	CPU Performance Check US-BOS 👁		
	US-BOS-R1	US-BOS-R1	Cisco	CGS-MGS-AGS	CPU Performance Check US-BOS 💿		
	BST_POP2	BST_POP2	Cisco	2621	CPU Performance Check BST_PO 💿		
	BSTX	BSTX	Cisco	2811	CPU Performance Check BSTX 📀		
0	BST,POP1	BST,POP1	Cisco	2514	CPU Performance Check BST,PO 💿		
1	BOS-cEdge-01	BOS-cEdge-01	Cisco	CSR1000V	CPU Performance Check BOS-cE 💿		
2	BJ_core_3550	BJ_core_3550	Cisco	catalyst355024	CPU Performance Check BJ_core 💿		
3	BJ_L2_Core_6	BJ_L2_Core_6	Cisco	catalyst356048TS	CPU Performance Check BJ_L2_C •		
4	BJ_L2_Core_5	BJ_L2_Core_5	Cisco	catalyst356048TS	CPU Performance Check BJ_L2_C •		
5	BJ_L2_Core_4	BJ_L2_Core_4	Cisco	catalyst37xxStack	CPU Performance Check BJ_L2_C 💿		
6	BJ_L2_Core_3	BJ_L2_Core_3	Cisco	catalyst37xxStack	CPU Performance Check BJ_L2_C ④		

### 5.1.4 Run the Intent

In this section, you will run intents once to populate all the results in the Intent columns (the Run Intent Once is useful for testing purposes. For continuous network assessments, you can schedule running intents by **Run Intent via Timer**). Follow the step-by-step instructions to **Run Intent Once** and **Rebuild Table**:

- 1. Hover over the Intent column, click  $\equiv$  menu and then click **Run Intents Once** to execute all the Intents.
- 2. In the **Run Intents Once** dialog, various options are available for specifying the data source and filtering the Intents to be executed.
  - a) **Select Data Source**: Any one of the three Data Sources can be selected: **Live Data**, **Current Baseline**, and **ADT Dataset**. In this example, we will use **Live Data** as a Data Source.
  - b) Click **OK** to run the Intent once.
- 3. In the **Notification** window, click **OK**.

	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~_~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	S Model	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	CPU Performance	1		
			3560E		Run Intents Once			
			3560E	$\left( \right)$	Run Intents via Timer		<b>🖴</b> {	
			3560E	Ļ	Open Seed Intent		)	
	Run Intents Once - CPU	Performance				×	roup	
	Data Source: Live Data		✓ belongs to	Select 🗸				
Notificat	tion			×			2	
The	The intents in the current ADT column are executed once now.							
View I	Details		3	ОК			ار بار مارکان مارکان	

4. Wait for the system to finish executing the intents. Then, click **Rebuild Table** to open the **Rebuild Table** dialog.

**Note**: To ensure the intents are executed successfully, randomly click any intent row and check the diagnosis message for confirmation.

- 5. Select the **Column Group** from the dropdown and tick the **Production Mode** as per your preference.
- 6. Click **Build** to update the Intent column.

Please wait a few minutes for the Intents to execute, then check the Intent status code and execution time.

} { <b>₿</b> СР	U Performance Check 🙎		Table Builder	Last Updated at	: 06/14/2024	4 06: 4	🌯 Rebuild Table	in the
} Descr	iption: Type description here							100
} Items	Rebuild Table						×	
No.	Build the column groups:	All 5				~		IC
1		All				· ·		CPU
}	Log:	Production	Mode Only	show major execution	process log	5		() () () () () () () () () () () () () (
		○ Debug Mod	e Show	all the detailed log				
	~				Cancel	Build	6	

You can see the updated ADT with the Intent Status Code and the Last Execution columns. The ADT table supports the standard operations of a table. For example, you can search the keyword **High** to filter the devices with the alert status message, **high CPU**. The ADT table can be exported to a CSV file.

🖷 CF	PU Performance Check	Table Builder	Last Updated at: 08/10/2024	01:49 PM 🔍 Rebuild Table			Add Data Manually $\sim~\equiv~$
Descr	ription: Type description	n here					
ltems	a: 24 Rows 7 Columns					Search	Q T Advanced Filter: Undefined
No.	Device	SHostname	5 Vendor	S Model	CPU Performance	SIntent Status Code	CLast Execution Time
1	US-BOS-SW5	US-BOS-SW5	Cisco	3560E	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:32 PM
2	US-BOS-SW4	US-BOS-SW4	Cisco	3560E	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:32 PM
3	US-BOS-SW3	US-BOS-SW3	Cisco	3560E	CPU Performance Check US-BOS 👁	Low CPU: 0	06/14/2024 06:44:32 PM
1	US-BOS-SW2	US-BOS-SW2	Cisco	3560E	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:32 PM
5	US-BOS-SW1	US-BOS-SW1	Cisco	3560E	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:38 PM
6	US-BOS-R2	US-BOS-R2	Cisco	CGS-MGS-AGS	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:32 PM
7	US-BOS-R1	US-BOS-R1	Cisco	CGS-MGS-AGS	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:38 PM
8	BST_POP2	BST_POP2	Cisco	2621	CPU Performance Check BST_POP2 •	High 5 second average CPU: 18%	06/14/2024 06:44:42 PM
9	BSTX	BSTX	Cisco	2811	CPU Performance Check BSTX 📀	Low CPU: 1	06/14/2024 06:44:34 PM
10	BST,POP1	BST,POP1	Cisco	2514	CPU Performance Check BST, POP1 💿	Low CPU: 4	06/14/2024 06:44:35 PM
11	BOS-cEdge-01	BOS-cEdge-01	Cisco	CSR1000V	CPU Performance Check BOS-cEd 💿	Low CPU: 1	06/14/2024 06:44:32 PM
12	BJ_core_3550	BJ_core_3550	Cisco	catalyst355024	CPU Performance Check BJ_core 💿	Low CPU: 3	06/14/2024 06:44:38 PM
13	BJ_L2_Core_6	BJ_L2_Core_6	Cisco	catalyst356048TS	CPU Performance Check BJ_L2_C •	Low CPU: 4	06/14/2024 06:44:42 PM
14	BJ_L2_Core_5	BJ_L2_Core_5	Cisco	catalyst356048TS	CPU Performance Check BJ_L2_C •	Low CPU: 5	06/14/2024 06:44:35 PM
15	BJ_L2_Core_4	BJ_L2_Core_4	Cisco	catalyst37xxStack	CPU Performance Check BJ_L2_C •	High 5 second average CPU: 11%	06/14/2024 06:44:40 PM
16	BJ_L2_Core_3	BJ_L2_Core_3	Cisco	catalyst37xxStack	CPU Performance Check BJ_L2_C •	High 5 second average CPU: 6%	06/14/2024 06:44:39 PM

## 5.1.5 Export ADT to CSV file

ADT can be exported as a CSV file for documentation. To export the file, follow these steps:

1. In the ADT, click  $\equiv$  menu from the upper-right corner and select the option **Export to CSV Only**.

				😮 Help
Table Builder	Last Updated at: 08/10/2024 01:49 PM 🛛 🍭 R	ebuild Table	Q	Add Data Manually V = 1
SHostname	CPU Performance	SIntent Status Code	OL EX	Export Export to CSV Only
US-BOS-SW5	CPU Performance Check US-BOS 💿	Low CPU: 0	06/1-, 202	Export Datasets to File
US-BOS-SW4	CPU Performance Check US-BOS 📀	Low CPU: 0	06/14/202	Dataset Tag Settings
JS-BOS-SW3	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/202	Table Settings
US-BOS-SW2	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/202	4 06:44:32 PM
US-BOS-SW1	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/202	4 06:44:38 PM
US-BOS-R2	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/202	4 06:44:32 PM
US-BOS-R1	CPU Performance Check US-BOS ④	Low CPU: 0	06/14/202	4 06:44:38 PM
BST_POP2	CPU Performance Check BST_POP2	High 5 second average CPU: 18%	06/14/202	4 06:44:42 PM

The exported CSV file will be saved automatically to your computer's default download location, typically the "Downloads" folder (e.g., **C:\Users<your username>\Downloads**).

A	В	с	D	E	F	G
1 Device	Hostname	Vendor	Model	CPU Performance	Intent Status Code	Last Execution Time
2 US-BOS-SW5	US-BOS-SW5	Cisco	3560E	CPU Performance Check US-BOS-SW5	Low CPU: 0	2024-06-14T13:14:32.236Z
3 US-BOS-SW4	US-BOS-SW4	Cisco	3560E	CPU Performance Check US-BOS-SW4	Low CPU: 0	2024-06-14T13:14:32.236Z
4 US-BOS-SW3	US-BOS-SW3	Cisco	3560E	CPU Performance Check US-BOS-SW3	Low CPU: 0	2024-06-14T13:14:32.236Z
5 US-BOS-SW2	US-BOS-SW2	Cisco	3560E	CPU Performance Check US-BOS-SW2	Low CPU: 0	2024-06-14T13:14:32.236Z
6 US-BOS-SW1	US-BOS-SW1	Cisco	3560E	CPU Performance Check US-BOS-SW1	Low CPU: 0	2024-06-14T13:14:38.533Z
7 US-BOS-R2	US-BOS-R2	Cisco	CGS-MGS-AGS	CPU Performance Check US-BOS-R2	Low CPU: 0	2024-06-14T13:14:32.236Z
8 US-BOS-R1	US-BOS-R1	Cisco	CGS-MGS-AGS	CPU Performance Check US-BOS-R1	Low CPU: 0	2024-06-14T13:14:38.548Z
9 BST_POP2	BST_POP2	Cisco	2621	CPU Performance Check BST_POP2	High 5 second average CPU: 18%	2024-06-14T13:14:42.064Z
10 BSTX	BSTX	Cisco	2811	CPU Performance Check BSTX	Low CPU: 1	2024-06-14T13:14:34.392Z
11 BST, POP1	BST,POP1	Cisco	2514	CPU Performance Check BST,POP1	Low CPU: 4	2024-06-14T13:14:35.752Z
12 BOS-cEdge-01	BOS-cEdge-01	Cisco	CSR1000V	CPU Performance Check BOS-cEdge-01	Low CPU: 1	2024-06-14T13:14:32.236Z
13 BJ_core_3550	BJ_core_3550	Cisco	catalyst355024	CPU Performance Check BJ_core_3550	Low CPU: 3	2024-06-14T13:14:38.267Z
14 BJ_L2_Core_6	BJ_L2_Core_6	Cisco	catalyst356048TS	CPU Performance Check BJ_L2_Core_6	Low CPU: 4	2024-06-14T13:14:42.486Z
15 BJ_L2_Core_5	BJ_L2_Core_5	Cisco	catalyst356048TS	CPU Performance Check BJ_L2_Core_5	Low CPU: 5	2024-06-14T13:14:35.892Z
16 BJ_L2_Core_4	BJ_L2_Core_4	Cisco	catalyst37xxStack	CPU Performance Check BJ_L2_Core_4	High 5 second average CPU: 11%	2024-06-14T13:14:40.923Z
17 BJ_L2_Core_3	BJ_L2_Core_3	Cisco	catalyst37xxStack	CPU Performance Check BJ_L2_Core_3	High 5 second average CPU: 6%	2024-06-14T13:14:39.642Z
18 BJ_Dis_SW2	BJ_Dis_SW2	Cisco	catalyst295024			
19 BJ_Dis_SW1	BJ_Dis_SW1	Cisco	catalyst295024	CPU Performance Check BJ_Dis_SW1	High 5 second average CPU: 10%	2024-06-14T13:14:37.22Z
20 BJ_Acc_Sw4-bbb	BJ_Acc_Sw4-bbb-ee	Cisco	catalyst295024	CPU Performance Check BJ_Acc_Sw4-b	High 5 second average CPU: 8%	2024-06-14T13:14:44.267Z
21 BJ_Acc_Sw4	BJ_Acc_Sw4	Cisco	catalyst295024	CPU Performance Check BJ_Acc_Sw4	Low CPU: 3	2024-06-14T13:14:37.439Z
22 BJ_Acc_SW6	BJ_Acc_SW6	Cisco	catalyst295024	CPU Performance Check BJ_Acc_SW6	High 5 second average CPU: 6%	2024-06-14T13:14:38.689Z
23 BJ_Acc_SW2-bbb	BJ_Acc_SW2-bbb-ee	Cisco	catalyst295024	CPU Performance Check BJ_Acc_SW2-I	Low CPU: 1	2024-06-14T13:14:34.251Z
24 BJ_Acc_SW1	BJ_Acc_SW1	Cisco	catalyst295024			
25 BJPOP	BJPOP	Cisco	2811	CPU Performance Check BJPOP	Low CPU: 0	2024-06-14T13:14:38.252Z

## 5.1.6 Create Intent Dashboard for ADT Automation Column

The dashboard provides an easy-to-understand and visual way to look into the ADT data, including the intent results. Especially the user can have an overview of the data with a glance. Creating the Dashboard from the ADT is easy as follows:

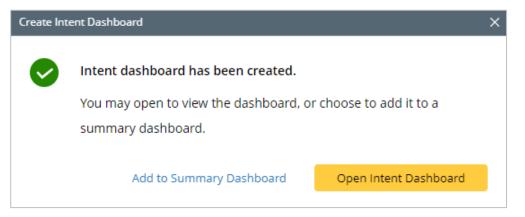
- 1. Hover over the Intent column, click  $\equiv$  menu and select the **New Intent Dashboard**.
- 2. In the **Create Intent Dashboard** window, define the following:
  - a) Enter the **Dashboard Name**, such as **CPU performance check**.
  - b) Select the **Location** where you wish to save the Intent Dashboard.

The **Data Source** section is already set to be predefined ADT and Intent Columns since you are creating the dashboard using the automation column.

c) Click Create.

Create Inten	t Dashboard for ADT 'CPU Performance Check'	
Name:	CPU Performance check	CPU Perfo 🕨 Run 🖹 Det
Location:	My Dashboards	Run Intents Once Run Intents via Timer
🗌 Use Templat	te	Open Seed Intent Rebuild Intent-related Column Gr
Data Source	Automation Data Table ~	Remove Empty Wrapper Intent Enable Auto Intent
Automation Da	ta Table: CPU Performance Check	Export Diagnosis Result to CSV
🖌 Include Trigg	gered Follow-up Intent Results	View Summary Report Export Intent Output Map
Intent Column:	CPU Performance 🗸	Debug Empty Cells
🗌 Filter Intent	by Devices	Tag Current Column
		Edit Delete Set as Table Key
Time Range	Last 7 Days 🗸	Submit Related Commands to Bencl
		New Intent Dashboard

3. In the Create Intent Dashboard dialog, click Open Intent Dashboard.



NOTE: The Intent Dashboard observes specific network issues with details, while the summary dashboard provides an overall view by displaying Intent results from multiple Intent dashboards. With Summary Dashboard, you can group Intent Dashboards into widgets based on diagnosis purposes and display Intent results by device, site or device groups. You can use the summary dashboard to monitor critical information across thousands of devices and discover the root cause for issues in one view.

CPU Performance check					La	st Refreshed at 17/06/2024, 11:41:02 🖸 🕻 🦉 🚍
Summary		17/06/2024, 11:4	:10 View Report	Device Information		17/06/2024, 11:41:10 View Report
	22 22 6	Intents Times Executed Intent-level Alerts		• Cisco 105 S	Switch	22 Devices
Intent Result History						17/06/2024, 11:41:10 View Report
Time Range: All V Result: All V		🔶 Sum of Intent Ale	t Status Code Count -	← Sum of Intent Success Status Code Co	ount	
5 0 14/06/2024, 14/06/2024, 18:44:32 18:44:34	14/06/2024, 1 18:44:34	4/06/2024, 14/06/2024, 14/06/2024, 14/06/2024 18:44:35 18:44:37 18:44:37		6/2024, 14/06/2024, 14/06/2024, 44:38 18:44:38 18:44:38		109/2024, 14/05/2024, 14/05/2024, 14/05/2024, 18:44:40 18:44:42 18:44:42 18:44:44
			Top Five Int	ent Alerts		
Intent Name	Мар	Execution Time	Intent Alert Status Coo	de Count Intent Success Status	Code Count Intent Status Code S	ummary Intent Alert Detection
CPU Performance Check BJ_L2_Core_4		14/06/2024, 18:44:40	1	0	High 5 second avera	
CPU Performance Check BJ_L2_Core_3		14/06/2024, 18:44:39	1	0	High 5 second avera	
CPU Performance Check BJ_Acc_SW6	View Map	14/06/2024, 18:44:38	1	0	High 5 second avera	ge CPU: 6% 1

#### 5.1.6.1 Edit Intent Dashboard

You can modify the Intent Dashboard in edit mode by selecting **Edit Dashboard** from  $\equiv$  menu located at the top-right corner of the dashboard window:

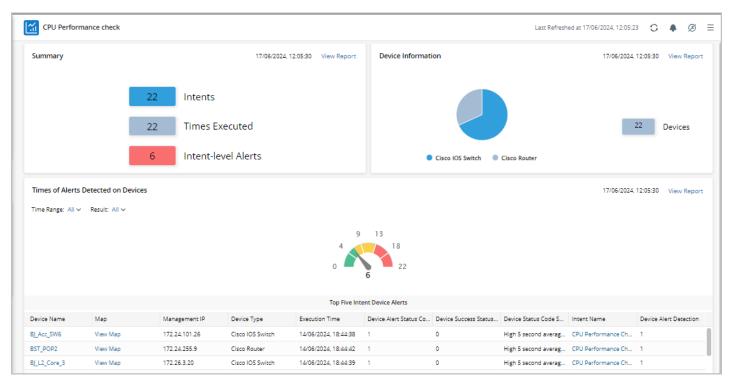
- 1. Click the **Intent Result History** section to view the dashboard data to be edited.
- 2. Change the Historical Result to the Last Result.
- 3. Click Edit Data and select Intent Device Result
- 4. Change Measure to Times of Alerts Detected on Devices.

From the **Dashboard**, you can observe that 6 devices out of 22 devices have *High 5 second average CPU*.

- 5. Use the **Alert Notifications** feature to enable Alert Notification to send alerts to the recipient's email address. You can adjust the Notification Frequency.
- 6. Click **Save** to save the Intent Dashboard and close the Dashboard.

		Last Refreshed at 10/08/2024, 14:15:05 C 🔶 🖉 🦉						
		Information 17/06/: Alert Notification 5 Close Save 6	Share Dashboard Save as Save as Template Update					
Alert Notifica	ation ×	Chart Data	View Dashboard Input Export Diagnosis Result to CSV					
	ert Notification  C Is automatically when alerts are detected in selected intent groups.	<ul> <li>→ Historical Result</li> <li>● Last Result</li> <li>2</li> <li>✓ Edit Data</li> </ul>	Add to Summary Dashboard					
Monitor lı	ntent Groups: All Intent Groups (1)	Intent Result     Intent Device Result						
	on Frequency: Maximum once every 10 Days ~	Times of Alerts Detected on Devices						
То:	abc@hotmain.com	Alert Count						
Subjects:	Alerts Detected for Dashboard "CPU Performance check"	Success Count Times of Alerts Detected on Devices						
Body:	8 Details Included in Email ~							
	Notification message(optional)							
Send Em	nail with Historical Alerts Cancel OK							

The final Intent Dashbaord will be like:



# 5.2 ADT Drilldown

In the last section, you learn how to create a new ADT via the **Intent Replication Wizard**. The key steps are:

- Create a new ADT from devices of a device group and add the built-in device properties to the ADT column. The devices and their properties form the **Base Table** of ADT, corresponding to the critical asset of your network.
- The seed intent is replicated, and cloned intents are added to the ADT group, which is called the **Column Group**, corresponding to the intents associated with the critical assets.

In this section, we will drill down deep into the ADT functions and operations:

• Create an ADT from scratch.

- Build ADT base from two common methods: from **Devices of a Device Group**, and from **a CSV file.** Another method, from the **Pre-replicated Intent Template**, is useful for the network assessment and document and will be covered in Section 5.3.
- Build ADT group from the intent template.
- Manually adjust the ADT.

You will use the same **Device Group** and a **Seed Intent** from Section 5.1 (CPU Performance Check) so that you can focus on the concepts and functions of the ADT. The goal is to obtain the same ADT as in Section 5.1, providing you with another way to understand ADT and its features.

# 5.2.1 Build Base ADT

To create an ADT and build the base table with the data in devices, follow the steps below:

- 1. Click the plus icon 🕑 and click **New Automation Data Table**.
- 2. In the New Automation Data Table popup:
  - a) Enter ADT Name.
  - b) Select the Location you wich to store the ADT.
  - c) Click **Save** and wait a moment for ADT to open.

	Net3rain Next-G	en Search Anything	and Create Map	
Ð	Provide the second seco	Rew Device Group	New Automation Data Table	×
Recents		New Device Group	Name: CPU Performance_Tes	st 🛃
Network	I. New Intent	New Automation Data Table	Location: Sachin b	~
Files		<b>4</b>		Cancel Save c
Site	New Intent Dashboard	New Network Change	•••	

- 3. Click **Table Builder** to open **Automation Data Table Builder** to define the Base Group.
- 4. Under the **Base** tab of Automation Data Table Builder, define the following settings:
  - a) Input **Description** for the base table to describe its use and function.
  - b) Select **Method**, the **Devices of Device Group** from the dropdown to build the base table.

- c) Click the **Select** link to select the created device group for building the base table with the devices in the device group, and then click **OK**.
- d) Within the Built-in Fields section, choose each column and move it into the **Column Group** (**Base**) pane.

You are going to build an ADT table with the following columns: **Device**, **Hostname**, **Vendor**, and **Model**.

e) Click **Save and Build**. The **Build** Table dialog appears, define the settings as per your preferences and then click **Build** to save all the settings.

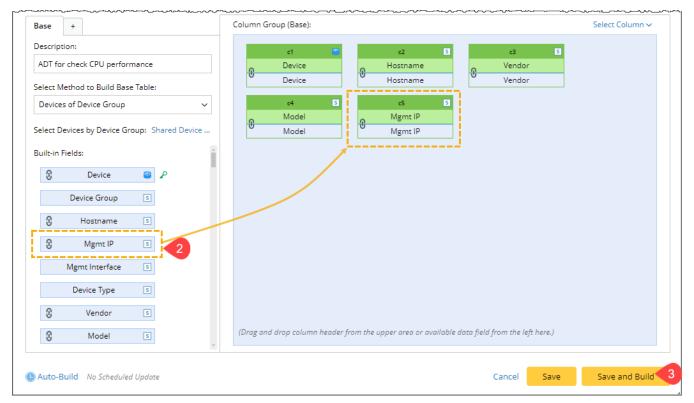
omation Data Table Builde	r							
Column Header:								Reset
	C c2	s 🕻	G	s ()	c4	S		nese:
Device	Hostname	_	Vendor		Model			
Base +		Colu	ımn Group (B	Base):				Select Column 🗸
Description:		1.1	c			c2	5	
ADT to check CPU perform	mance			vice	۰ ۵	Hostname		
Select Method to Build Bas	se Table:		De	vice	U	Hostname		
Devices of Device Group	Ь	~	C <sup>4</sup>			з	5	
Select Devices by Device G	roup Shared Devic	e	ß	odel	0	Vendor Vendor		
Built-in Fields:           ①         Device           Device Group         Device Group	م <b>ڪ</b> 5		Select Device	e Groups		_	×	
🕄 Hostname	s d		Search			Q		
Mgmt IP	S		ÞL	] Sachin				
Mgmt Interface	S		⊿ 🕻	Z Sachin_TV		_		
Device Type	S		⊳ ⊏	Cisco D 🚭 🔽 Top 10 Su		PU perfor		
🕄 Vendor	S			] <mark>=</mark> vamsi	, 200			
C Model	S		Г	NS All BGP D	evices (59)	•		
					Canc	el OK		
Software Version	S				Carlo		he left here.)	
Serial Number	S	~					ne rejt nere.j	
Auto-Build No Schedule	ad Update						Cancel Save	e Save and Build

j Au	itomation Data Tab	le Manager			(2) Help
🖽 CP	'U Performance_Test	Table Builder	Last Updated at: 06/18/2024 01:46	PM 🔌 Rebuild Table	Add Data Manually 🗸 😑
Descri	iption: Type description here.				
ltems:	: 27 Rows 4 Columns			Search	Q T Advanced Filter: Undefined
No.	Device	SHostname	s Vendor	5 Model	
I	US-BOS-SW5	US-BOS-SW5	Cisco	3560E	
2	US-BOS-SW4	US-BOS-SW4	Cisco	3560E	
3	US-BOS-SW3	US-BOS-SW3	Cisco	3560E	
Ļ	US-BOS-SW2	US-BOS-SW2	Cisco	3560E	
5	US-BOS-SW1	US-BOS-SW1	Cisco	3560E	
5	US-BOS-R2	US-BOS-R2	Cisco	CGS-MGS-AGS	
7	US-BOS-R1	US-BOS-R1	Cisco	CGS-MGS-AGS	
3	BST_POP2	BST_POP2	Cisco	2621	
9	BSTX	BSTX	Cisco	2811	
10	BST,POP1	BST,POP1	Cisco	2514	
1	BST	BST	Cisco	2503	
2	BOS-cEdge-01	BOS-cEdge-01	Cisco	CSR1000V	
3	BOS-N9K-L3OUT	BOS-N9K-L3OUT	Cisco	NexusC9372TXE	
4	BJ_core_3550	BJ_core_3550	Cisco	WS-C3550-24	
5	BJ_L2_test_1	BJ_L2_test_1	Cisco	WS-C3750-24TS	
16	BJ_L2_Core_6	BJ_L2_Core_6	Cisco	catalyst356048TS	

#### 5.2.1.1 Edit ADT Table to Add Additional Fields

You will add additional fields Mgmt IP using table builder:

- 1. Click Table Builder.
- 2. From the **Built-in Fields** section, drag and drop **Mgmt IP** to the **Column Group** (**Base**) section.
- 3. Click **Save and Build**. The **Build Table** dialog appears, define the settings as per your preferences and then click **Build** to save all the settings.



#### Here is the ADT after adding the additional column:

to CPI	J Performance_Test	Table Builder	Last Updated at: 06/17/2024 05:0 🍳 Rebuild Table Add Data Manually 🗸 😑 🔐										
Descri	Description: Type description here												
Items:	27 Rows 5 Columns			Search	Q	🍸 Adva	nced Filter: Undefined	0					
No.	管 Device	5 Hostname	s Vendor		s Model		s Mgmt IP	≡					
1	US-BOS-SW5	US-BOS-SW5	Cisco		3560E		10.8.1.52	1					
2	US-BOS-SW4	US-BOS-SW4	Cisco		3560E		10.8.1.244						
3	US-BOS-SW3	US-BOS-SW3	Cisco		3560E		10.8.1.243						
4	US-BOS-SW2	US-BOS-SW2	Cisco		3560E		10.8.1.242						
5	US-BOS-SW1	US-BOS-SW1	Cisco		3560E		10.8.1.241						
6	US-BOS-R2	US-BOS-R2	Cisco		CGS-MGS-AGS		10.8.1.240						
7	US-BOS-R1	US-BOS-R1	Cisco		CGS-MGS-AGS		10.8.1.51						

### 5.2.1.2 Remove a Column

Follow the steps to delete a column:

- 1. Click Table Builder.
- 2. From the **Column Header** section, hover over the **Mgmt IP** field, click and then click Delete from dropdown.
- 3. Click Save.

olumn	Header:																	Reset A
0	c1	-	C	c2	S	0	c3	S	C	c4	E	0	c5		5 - 2			
	Device			Hostname			Vendor			Model			Mgmt	IP.	Edit			
Base Descript	+ tion:					Colur	mn Group (B	ase):	*		c2		5			S	Selec	Column 🗸
ADT fo	r check CPU j	perfor	mance				Dev	ice		0	Hostname				Vendor			
Select Method to Build Base Table:					Ċ	5 Dev	ice		0	Hostna	ame		Ü	Vendor				
Davica	s of Device G	roup			~		c4		S		c5		S					

### 5.2.1.3 Adjust ADT Column Order

After an ADT is created, you are allowed to change the order of its columns. Further, the columns can be set to be invisible to display only essential column data. Follow the below steps to adjust the ADT columns:

- 1. Click  $\equiv$  menu in the top right corner of the column headers.
- 2. In the **Column Setting** window, you can perform the following operations:
  - a) Untick the checkbox of the column you wish to make invisible from the ADT.
  - b) Hover over the column you wish to adjust and click items listed in the drop-down menu of a column to change their order.
  - c) Click **OK** to save the settings.

Colum	n Setting	<b>2</b>	×	
				ble Add Data Manually 🗸 📃 🚮
	No.	Column		
	1	Device		
	2	Hostname		
	3	Vendor		T Advanced Filter: Undefined
	4	Model		
		Move to Top		
		Move Up		
		Move Down		
		Move to Bottom		
		Cancel	ок с	

### 5.2.1.4 Use of Advanced Filter in ADT

In the ADT table, **Advanced Filter** is used to accumulate all the empty rows based on the selected column. Follow the below steps to define Advanced Filter:

- 1. In the ADT table header, click **Undefined** of the **Advanced Filter** field.
- 2. In the Advanced Filter popup, tick the **Filter empty row with** the checkbox.
- 3. From the dropdown, select the column you wish to filter. In this example, the Model column is selected.
- 4. Click **OK** to apply the filter.

🍳 Re	build Table	Add Data Manually 🗸 📃 🚮
	Search	Q T Advanced Filter: Undefined
Advanced Filter		×
2 ✓ Filter empty row	vs with Model	
	Device	
	Hostname	Ok 4
	Vendor	
	Model 3	

5. When the filter is applied, you will see the two rows are filtered based on the column selection.

έ ξ ε	J Performance_Test	Table Builder	Last Updated at: 06/17/2024 05:0	🔦 Rebuild Table Add Data Manually 🗸 😑 🔐
l Descri	ption: Type description here			
ltems:	27 Rows 4 Columns		Search	Q 🛛 Advanced Filter: Defined 💼 😋
No.	Device	s Hostname	sVendor	S Model
{ { 1	US-BOS-R1	US-BOS-R1	Cisco	2
2	US-BOS-R2	US-BOS-R2	Cisco	

## 5.2.2 Build ADT from a CSV File

In the last section, you learned how to build base ADT using the **Devices of Device Groups** method. This section teaches how to create the ADT base table by **Importing a CSV File**.

**NOTE**: Make sure you have a CSV file available on your local computer. Once imported, the contents of the CSV file will be loaded into the ADT.

Sample of CSV file content:

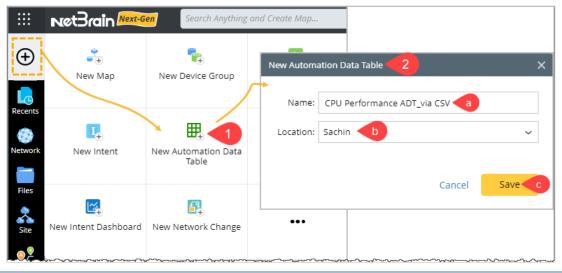
Device	Hostname	Vendor	Model
US-BOS-SW5	US-BOS-SW5	Cisco	3560E
US-BOS-SW4	US-BOS-SW4	Cisco	3560E
US-BOS-SW3	US-BOS-SW3	Cisco	3560E
US-BOS-SW2	US-BOS-SW2	Cisco	3560E
US-BOS-SW1	US-BOS-SW1	Cisco	3560E
US-BOS-R2	US-BOS-R2	Cisco	CGS-MGS-AGS
US-BOS-R1	US-BOS-R1	Cisco	CGS-MGS-AGS
BST_POP2	BST_POP2	Cisco	2621
BSTX	BSTX	Cisco	2811
BST	BST	Cisco	2514

To build the base table with the data in devices, follow the steps below:

1. Click the plus icon 🕑 and click **New Intent-Based Automation**.

#### 2. In the **New Automation Data Table** popup:

- a) Provide an ADT Name.
- b) Select the **Location** you wich to store the ADT.
- c) Click **Save** and wait a moment for ADT to open.



- 3. Click **Table Builder** to open **Automation Data Table Builder** to define the Base Group.
- 4. Under the **Base** tab of **Automation Data Table Builder**, define the following settings:
  - a) Input descriptions for the base table to describe its use and function.
  - b) Select Method **Imported CSV** from the dropdown to Build Base Table.
  - c) Click the **Select** link to import the CSV file.
  - d) Click **Browser** and select the CSV file from your local computer.
  - e) Click **OK**.

Column Header:	CSV: Import from L	ocal ~ CPU Performance.	csv	d Browse
Base +	CSV Column	Display Name	Data Type	
Description:	Device	Device	String	~
ADT using csv file	Hostname	Hostname	String	~
elect Method to Build Base Table:	Vendor	Vendor	String	~
Imported CSV b	Model	Model	String	~
SV: Select				
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~			

Under Built-in Fields, you can see Device, Hostname, Vendor, and Model fields.

- f) Within the Built-in Fields section, choose all the fields and move them into the Column Group (Base) pane.
- g) Under the **Define Table Key** dropdown, tick the **Device** checkbox.
- h) Click **Save and Build**. The Build Table dialog appears, define the settings as per your preferences and then click **Build** to save all the settings.

macioni	Data Table Build	ler											
iolumn H	leader:												Reset
C	c1	s ()	c2	s D	<b>c3</b> 5	0 0	4 s						
0	Device		Hostname		Vendor	Mo							
ase	+			Co	lumn Group (Base):							Selec	t Column 🗸
escriptio	on:				d	5	c2	S		в	5		
ADT usir	ng csv file				Device		Hostname		0	Vendor			
elect Me	thod to Build B	ase Table	s.		Device	÷	Hostname		8	Vendor			
Importe			~		c4	5							
					Model								
SV: CPU	J Performance.	SV			Model								
uilt-in Fi	elds:		ע רי	1									
C	Device	S	P										
£	Hostname	S											
				a	Drag and drop column	header from t	ne upper area or	available dat	ta field fro	om the lefi	here.)		
Ĵ	Vendor	S											
0	Model	5		Defi	ne Table Key								
				D	evice	~	g						
					s Device		9						
					_								
					S Hostname								
					S Vendor								
					S Model								

The ADT Base table is successfully built from the imported CSV file, which contains the columns.

<b>₩</b> C	PU Performance ADT_via CSV	Table Builder	Last Updated at: 06/18/2024 04:46 PM 🛛 🍳 🛛	Rebuild Table Add Data Manually 🗸 🗄	= d
Desc	ription: Type description here				
ltem	s: 10 Rows 4 Columns		Search	Q 🛛 🖓 Advanced Filter: Undefine	d C
No.	s Device	s Hostname	s Vendor	S Model	
1	US-BOS-SW5	US-BOS-SW5	Cisco	3560E	
2	US-BOS-SW4	US-BOS-SW4	Cisco	3560E	
3	US-BOS-SW3	US-BOS-SW3	Cisco	3560E	
4	US-BOS-SW2	US-BOS-SW2	Cisco	3560E	
5	US-BOS-SW1	US-BOS-SW1	Cisco	3560E	
6	US-BOS-R2	US-BOS-R2	Cisco	CGS-MGS-AGS	
7	US-BOS-R1	US-BOS-R1	Cisco	CGS-MGS-AGS	
8	BST_POP2	BST_POP2	Cisco	2621	
9	BSTX	BSTX	Cisco	2811	
10	BST, POP1	BST, POP1	Cisco	2514	

## 5.2.3 Build ADT from Intent Template

After building a base table, you can add the automation to the **Column Group**. The Column Group can be built with the same types of methods to build the base table. The most frequently used one is the **Intent Template**.

Let us add an Intent group in the ADT (choose any of the Base ADTs created in the previous sections) using the Seed Intent that you have created in Section 5.1.3. Follow the steps to add Intent columns in the ADT using the **Intent Template** method:

Automation Data Table Builder				>
Column Header:	Create Group X	c4 3 3 c5 Model Replicated Inte	1 🕄 c6 nt Intent Status Co	Reset All de
Last Execution Time Base Intent Gro + Description: Select Method to Build Group Tai Intent Template Intent Template: CPU Performat Built-in Fields: Replicated Intent Intent Output:	cs Replicated Intent Replicated Intent	from the upper area or available data fiel	ti O ast Execution Ti ast Execution Time	Select Column >
Intent Message S Intent Status Code Last Execution Time	Select Device Column to Replica Replication Settings   Auto set		- 6	
C Auto-Build N Select Intent T Select Intent	emplate Template from:	×	Cancel Save	Save and Build 7
	(10.10.10.1 (3) (192.168.29.62 (4)) (3Com (4)) (5 R work Intents MST Compliance .heck AWS EC2 Configuration Against Baseline	v     sachin     X       leport (2)     AAA (5)     Acces		
		Cancel OK		

1. In the ADT, click **Table Builder** to open **Automation Data Table Builder**.

- 2. Click the icon to open the **Create Group** popup, type the group name as *Intent Group*, and then click **OK**.
- 3. Type **Description** (optional), and check for the method selected as **Intent Template**.
- 4. Select previously created Intent, i.e., *CPU Performance Check*.
- 5. Drag and drop required fields from the Built-in Fields and Intent Output sections into the Column Group (Intent Group) section. Add Replicated Intent, Intent Status Code (which usually includes the intent execution results), and Last Execution Time (knowing when the intent was executed last time can be useful for the end users).
- 6. Select **Device** from the dropdown to replicate Intents to the device column.
- 7. Click **Save and Build**. The **Build** Table dialog appears, define the settings as per your preferences and then click **Build** to save all the settings.

h Au	itomation Data Ta	able Manager					(2) Hel	٩ŀ
-	'U Performance_Test	Table B	uilder Last Updated at	t: 06/18/2024 01:46 PM 🛛 🍕 Rebuil	d Table		Add Data Manually 🗸 🗏	đ
	27 Rows 7 Columns	re				Search Q	Advanced Filter: Undefined	C
No.	Device	5 Hostname	5 Vendor	5 Model	Replicated Intent	SIntent Status Code	CLast Execution Time	
1	US-BOS-SW5	US-BOS-SW5	Cisco	3560E	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:3	
2	US-BOS-SW4	US-BOS-SW4	Cisco	3560E	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:3	
3	US-BOS-SW3	US-BOS-SW3	Cisco	3560E	CPU Performance Check US-BOS 👁	Low CPU: 0	06/14/2024 06:44:3	
4	US-BOS-SW2	US-BOS-SW2	Cisco	3560E	CPU Performance Check US-BOS 😐	Low CPU: 0	06/14/2024 06:44:3	
5	US-BOS-SW1	US-BOS-SW1	Cisco	3560E	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:3	
6	US-BOS-R2	US-BOS-R2	Cisco	CGS-MGS-AGS	CPU Performance Check US-BOS 👁	Low CPU: 0	06/14/2024 06:44:3	
7	US-BOS-R1	US-BOS-R1	Cisco	CGS-MGS-AGS	CPU Performance Check US-BOS 💿	Low CPU: 0	06/14/2024 06:44:3	
8	BST_POP2	BST_POP2	Cisco	2621	CPU Performance Check BST_PO 💿	High 5 second average CPU:	06/14/2024 06:44:4	
9	BSTX	BSTX	Cisco	2811	CPU Performance Check BSTX 📀	Low CPU: 1	06/14/2024 06:44:3	
10	BST, POP1	BST, POP1	Cisco	2514	CPU Performance Check BST,PO 💿	Low CPU: 4	06/14/2024 06:44:3	
11	BST	BST	Cisco	2503				
12	BOS-cEdge-01	BOS-cEdge-01	Cisco	CSR1000V	CPU Performance Check BOS-cE 💿	Low CPU: 1	06/14/2024 06:44:3	
13	BOS-N9K-L3OUT	BOS-N9K-L3OUT	Cisco	NexusC9372TXE				
14	BJ_core_3550	BJ_core_3550	Cisco	WS-C3550-24	CPU Performance Check BJ_core 💿	Low CPU: 3	06/14/2024 06:44:3	
15	BJ_L2_test_1	BJ_L2_test_1	Cisco	WS-C3750-24TS				
16	BJ_L2_Core_6	BI L2 Core 6	Cisco	catalyst356048TS	CPU Performance Check BJ_L2_C •	Low CPU: 4	06/14/2024 06:44:4	

Wait a moment for ADT to populate the Intent columns.

You can create an Intent Dashboard similar to the Section 5.1.6.

# 5.3 Assess and Document Your Network Operational Status

In the last two sections, you learn how to create the base table of an ADT from the built-in system data, such as network devices and their properties. We recommend that you use this easy method to populate your base table if possible. However, the built-in data may not satisfy your need, and you have to query the network operational status from the live network via the CLI command. For this case, you will parse the data from the CLI command results and export the data (variables) to the ADT table. The method to build this use case is called the **Pre-replicated Intent Template**.

In this section, you will asked to document all down interfaces, including **Administratively Down** or **Down**, and the interface uptime. This report can help you figure out how many ports are not used in your network and future network expansion planning.

This section includes the following main steps:

- <u>Create Network Intent</u>
- Add Signature Variables to ADT Base Table
- Export ADT to CSV file

👪 Do	🖏 Down Interfaces Table Builder Lost Updated at: 06/21/2024 12:49 PM 🍕 Rebuild Table Add Data Manually 🗸 🚍 🗗												
	ption: Type description here			ignature /ariables		Device Gro	up		7				
Items:	316 Rows 9 Columns							Search Q	Advanced Filter: Undefined				
No.	Device	S Interface_Name	Status	Sinput	S Hostname	S Mgmt IP	S Vendor	S Model	S Software Version				
1	US-BOS-SW2	Ethernet1/0	administratively down	never	US-BOS-SW2	10.8.1.242	Cisco	3560E	15.2(HI_20170202)F				
2	BSTX	Serial0/2/0	down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
з	BSTX	Serial0/3/0	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
4	BSTX	Serial0/3/1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
5	BSTX	BRI1/0	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
6	BSTX	BRI1/0:1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
7	BSTX	BRI1/0:2	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
8	BSTX	BRI1/1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
9	BSTX	BRI1/1:1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
10	BSTX	BRI1/1:2	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
11	BSTX	BRI1/2	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
12	BSTX	BRI1/2:1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
13	BSTX	BRI1/2:2	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
14	BSTX	BRI1/3	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
15	BSTX	BRI1/3:1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
16	BSTX	BRI1/3:2	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
17	BSTX	BRI1/4	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
18	BSTX	BRI1/4:1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				
19	BSTX	BRI1/4:2	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4				

### The Final ADT will be like:

## 5.3.1 Create Network Intent

In this section, you will create a Network Intent to specify the CLI command for the Parser and subsequently define the **Signature Variable** for down Interfaces. The Signature Variables can be imported to the ADT.

- 1. From the Intent Manager, create a new intent, **Down Interface Report**.
- 2. Add the target device(s), e.g., device US-BOS-R1).
- 3. Add a **CLI Diagnosis**: enter the command *show interface* and retrieve the data from the **Live Data**.
- 4. In the **+ New Pattern** dropdown, select **Paragraph** parser.

Network Intent (Edit Mode)		
Down Interface Report	Diagnosis Tree	Run with Live Data Save 3 Help
Type description here		👗 Intent Map: Select 🥆
I Seed Logic	🔟 Replication Logic 🛕	
🚳 + Device		Intent Variables: Manager Tag: + Add
v 🤭 US-BOS-R1	Type Description here	• Add Config Diagnosis     • Add CLI Diagnosis
CLI Command Diagnosis		1
US-BOS-R1 show interfa	ce X V Retrieve	with Live Data
1. Define Variable		2. Define Diagnosis
Format1 V	+	
(i) Double-click a variable to parse	. Select multiple lines to parse a table.	Critical Variable (0) No Pattern V + New Pattern V
Current Device	08/13/2024 12:15:57 PM Search	Auto Pattern
JS-BOS-R1#show int	erface	Single Variable
	line protocol is up	Table
	2, address is aabb.cc00.0c00 (bia aabb.	Table
4 Description: to		Paragraph 4
	BW 10000 Kbit/sec, DLY 1000 usec,	Advanced >
6 reliability 2		la fa
7 Enconculation AB	55/255, txload 1/255, rxload 1/255	Id fr
	PA, loopback not set	
8 Keepalive set (1	PA, loopback not set	Sample Data

### 5.3.1.1 Define the Paragraph Parser

In the sample data, select the text and create the following Variables:

- 1. Define ID Line A.
  - a) Double-click the interface *Loopback99* in line 154. The Variable *\$var1* is created.

Change the default name **\$var1** to **\$Interface\_name** 

b) In the same line of the sample data, double-click the phrase **administratively down**. The Variable **\$var2 down**, is created in the same ID Line.

Change the Variable **\$var2 down** with **\$mstring:status(down|administratively down)**, line

The final **ID Line A** will be: ^\$interface\_name is \$mstring:status(down|administratively down), line

- 2. Define **Var Line 1** for interface uptime.
  - a) Double-click the word *never* in line 160. The Variable *Input \$input, output* is created in the Var Line 1.
- 3. Verify the **Ouput** and click **Apply**.

			> 2. Define Diagn					
						Test on Devi	isasi 0 -	
Format1	1 🗸 +					Test on Devi	ices. 0 🚍	
Double-click a v	variable to parse. Select multiple lines to parse a table.	Critical Variable (0)	- Interfaces	× 🖌 TV	roe: Paragraph 🙆 🕴 + New Patr		=	
Current Device	e v 06/20/2024 12:53:06 PM							
148 0	o runts, o giants, o throttles ID Lin	eA ^\$inter	face na	ne is \$ms <sup>1</sup>	tring:status(d	down administra	atively a	down), 1
	0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored	pancer						
150 3	37282 packets output, 2511677 bytes, 0 underruns							
151 0	0 output errors, 0 collisions, 0 interface resets							
	0 unknown protocol drops		_					
	0 output buffer failures, 0 output buffers swapped out	P1-ID 0						
	ack99 is administratively down, line protocol is down	P1-ID 2 Var	r Line 1	input \$i	nput, output		=	
155	ware is Loopi		1		the state of the s			
156	1514 bytes, D 00000 Kbit/sec, DLY 5000 usec,			160 Last in	put never, output never, output h	ang never > 1 Line		
	reliability 255/255, txload 1/255, rxload 1/255			100 0000	for the start of t	ang nerer v r ente		
	apsulation LOOPBACK, loopback not set							
	palive set (10 sec)	Var Line 1	+ Field ~				-	
	t input never, output never, output hang never		- Tield •		<b></b>			
161 Last 162 Inpu	t cleated of "show interface" counters never ut que //75/0/0 (size/max/drops/flushes); Total output drop		Output	+ Parse Lines			-	
	ueing strategy: fifo							
	put queue: 0/0 (size/max)		\$interface_nar	ne 🗸	\$status	✓ \$input	~	
	inute input rate 0 bits/sec, 0 packets/sec		Loopback99		administratively down			
	inute output rate 0 bits/sec, 0 packets/sec		Loopbackaa		automistratively down	never		
	0 packets input, 0 bytes, 0 no buffer							
168 R	Received 0 broadcasts (0 IP multicasts)							
	0 runts, 0 giants, 0 throttles							
	0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort							
	0 packets output, 0 bytes, 0 underruns							
	0 output errors, 0 collisions, 0 interface resets							
	0 unknown protocol drops							
174 0	a output buffer failures A output buffers swapped out							

- 4. In the **Confirmation** popup, click **Apply and Continue** to save the parser.
- 5. Click the pen icon, rename the parser name from **Paragraph1** to **Interfaces**, and then click **OK**.

## 5.3.1.2 Define Intent Variable for Seed Logic

Since you only want to report the down interfaces (not all interfaces), you can create a compound table (sub-table) to filter out these interfaces:

Intent Variables for Se	ed Logic	×
Intent Variable	Use Automation Data Table Task Variable	lp ⊟
+ Add Compound Var	iable I + Add Compound Table V	
Intent	Merged Table	
2 JUS-BOS-R1	Sub Table (Filtered by Value) Full Settings for Template	
↓ , Interfaces	Appended Table	
US-BOS-R1	B Built-in Data	
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Add Diagnosis via Addo Intent	
Add Sub Table (Filte	ring Row by Condition)	
Table Manage		
Table Name:	down_interfaces 4	
I Base Table:	Interfaces 5	
Filtering Logic:	Only Keep v table rows if values match the below condition	
6	A status V Contains V down V	
	B Please Select ✓	
	Boolean Expression: A	
	Calculate (양)	
Base Table (Inte	rfaces) New Table (down_interfaces)	
base fable (inte	US-BOS-R1	
	🖽 show interface 📜 Interfaces	
interface_name (		
Loopback99	administratively down never	
🕜 Help	Cancel OK 7	

- 1. In the **Network Intent (Edit Mode)** window, click  $\equiv$  Menu and open **Intent Variables** to define compound Variables.
- 2. In the Intent Variables for Seed Logic window, select US-BOS-R1.
- 3. From the **+ Add Compound Table** dropdown, select the **Sub Table (Filtered by Condition)** option.
- 4. In the Add Sub Table (Filtering Row by Condition) window, enter the table name like down\_interfaces.
- 5. Select variable **Interfaces** as a Base Table from the dropdown.
- 6. Define condition **A** for Variable **status** contains **down**.
- 7. Click **OK** to save the configuration. You will see the Compound Table is added in the **Intent Variables** tab.
- 8. Click **Close** to save and close the window.

Intent Variables for Seed Log	gic			×
Intent Variable	Use Automation Data Table	e Task Variable		
+ Add Formula Column				
Intent				
🖌 🤭 US-BOS-R1		🖽 show interface		
Interfaces				
⊿ <u>Ⅲ</u> Down_interfaces				<b>v</b>
		US-BOS-R1		
	cu	show interface , 📃 Interface	s	
interface_name (stri	ng) status (	string)	input (string)	
Loopback99	adminis	tratively down	never	
L				

### 5.3.1.3 Create Signature Variables

For any intent variable to be added to an ADT, you have to define it as a **Signature Variable** in the **Full Settings for Intent Template**. To create signature Variables, start by defining the **Intent Qualification** using devices and critical Variables.

- 1. In the **Network Intent (Edit Mode)** window, go to  $\equiv$  menu, and click **Full Settings for Template** to define device qualification and critical Variables.
- 2. Enable the toggle button, **Serve as Template for,** and ensure that **Device-based Replication** is selected.
- 3. Under the **Intent Qualification** tab, click **Undefined** to define the devices via **Dynamic Search**.
  - a) Set the **Device Type** to be **Cisco IOS Switch** and **Cisco Router**.
  - b) Click **OK** to save the configuration and close the window.

Intent Settings	Qu	alified Device				
Intent Variables	D	ynamic Method: Se	lect criteria to filter dev	ices in All Devices	~	
Lock Settings	A	Device Type	<b>a</b> ~	Matches any	✓ Cisco IOS Sv	vitch,Cisco Ro 🗸
Add Intent Diagnosis Block Add Config Di Add Diagnosis via Auto Intent	в	Select Criteria	~			
tttings for Intent Template Serve as Template for:  O Device-based Replication	Path-bas	oolean Expression:	A			
Intent Qualification Macro Variable	Critical Variable					Searce 432 Qualified De
Device Qualification:	·	Static Method: In	clude/Exclude target de	vices		
O via Device Groups/Sites: Select		+ Include Device	+ Exclude Device			
via Dynamic Search: Undefined		Hostname		Vendor	Model	Management IP
Command Qualification: 0/1						

- 4. Define Critical Variable.
  - a) Under the **Critical Variable** tab, click the **Manually Select** option.
  - b) Under the **Critical Variable** column, select the Variable *status*.

Note: Select only the Variable status; otherwise, the ADT will include empty lines.

Full S	ettings for Intent Templa	ate			4						>
	) Serve as Template for:	🖲 Devi	ce-based Replication	O Path-based R	Replication	🗌 Enab	le Neighbor Pair Replication				
	Intent Qualification	N	lacro Variable	Critical Variable	Advance S	Settings					
Sele	ect variables as Critical V	ariables t	o qualify devices which	h can be matched with s	eed device pars	ers: 🚯					
	Automatically Select All D	Diagnosis	Variables								
	Down Interface Repo	ort									
	🖌 👩 US-BOS-R1										
	Boolean Algebra	=	CLI Command					Critical	Variables	Replicate	
	<b>O</b> A	=	💷 show interface					1 Varia	ble Selected		
	Boolean Expression:	A							show interface 🖬 show interfaces		
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~						~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		status		0-0

- 5. Create Signature Variables in Advance Settings.
  - a) Under the **Signature Variables** section of the **Advanced Settings** tab, click + **Add via Table Columns**.
  - b) From the Select Table Columns window, check all the Interfaces Variables checkboxes.
  - c) Click OK to open the Add Signature Variable window.
  - d) In the Add Signature Variables window, provide the name Down\_interface.
  - e) Click **OK** to save the table columns.

Full Settings for Inter	nt Template	•								×
Serve as Temp	plate for: (	Device-based Replication	on O Path-based	Replication	nable Neig	hbor Pair Re	plication			
		,	0.111				le Columns		×	
Intent Qualific	ation	Macro Variable	Critical Variable	Advance Settings	5					
Match Target Device	e with Seed	Devices:				Search			۹	
O Match 1 See	d Device O	nly					5-BOS-R1 show interface			
Try to Match	All Seed D	evices					🛛 🌐 Interfaces			
Name Rule: 🚯 🖇	Snit name	<pre>\$device_name</pre>					🗹 📗 interface_name 🗸 ∏ status	Þ		22
	-						🗹 📔 input			
Automation Tag for	Cloned Int	ents: + Add 📄				C	Down_interfaces			
-							🗌 📔 interface_name 🗌 🚺 status			
Signature Variables	: /ia Table C	olumos + Add via Si	ingle Variables				🗌 📔 input			
Signature Var			Type							
Signature var	Idoles Nali		туре					Cal C OF		
								Cal		
	Add Sigr	nature Variables							×	
						_				
	Name:	Down_Interfaces	d							
	<b></b>									ОК
	Selec	cted Seed Devices and Ta	bles	Display Name of Colu				1.		4
	Seed	d Device: Table	2:	interface_name	status		input	+		
	0	US-BOS-R1	Interfaces	interface_name	status	· · ·	r input v	Ī		
	+	Add Table Columns								
	6 Hov	v it works.						Cancel Of	е	
								Cancer Or		

- 6. Click **OK** to save the Intent Template settings.
- 7. Click **Save** to save the NI and then close the window.

# 5.3.2 Add Signature Variables to ADT Base Table

In this section, you will create a new base table in ADT using a **Pre-replicated Network Intent Template** method for the inclusion of Signature Variables.

**Note**: The Signature Variables can only be used in the base table of ADT and when the method to build the base table is the **Pre-replicated Intent Template**.

A Signature Variable is a specialized type of Variable intended to reveal key Variable values used in parsers for NIT replication. It can hold two types of values: single value and table. This Variable type enables data output to an ADT as row data.

### 5.3.2.1 Decode Network Intent

Before an intent template can be used to create a base table of an ADT, you must install and decode it in the **IBA Center**:

‱ Net3rain				
Search App			🏶 Domain Management	
Network Map-Based Automation	Intent-Based Automation	Incident & Change N	Misc	
Intent	Execute Intent	Dashboard		
🍕 Intent Manager	🖏 Intent Based Automation Cente	r 🚺 📑 Summary Da	ashboard	
🔩 Intent Cluster Manager	🕏 Triggered Automation Manager	🔛 Intent Dashb	board	
🔀 Automation Data Table (ADT)	🔥 Preventive Automation Manage	r 💿 Universal Da	ashboard	
🐼 Task Variable Manager	Schedule Automation	👶 Report Mana	ager	
📭 Intent Based Automation Center	r	5		
Installed Intent Templates Published Intents	s Auto Intent Auto Intent Profile	NetBrain Download		
Items: 2 + Add Intent Template				
Intent Template Name     Location	Intent Decodir	g Deco		
0-BGP Demo				
Add Intent Template	×	Select Intent Template		×
Intent Template:	3 Browse	Select Intent Template from: 💿 A	All Intents 🔘 Installed Intents	
Group: Default	ed and a second se	-		
Group: Default		Type: Common Intent 🗸	Filter by: no filter V	sachin X
Occoding Settings	Cancel OK	All (10.10.1 (6) (19)	92.168.29.62 (4) 3Com (5) 5 Report (2	2) (AAA (5) (Acces >>
		All Network Intents		
		Sachin NIST Compliance		
		I Check AWS EC2 Config	guration Against Baseline	
		I CP1		
		I CPU Performance Che	eck	
		I Demo-OSPF Router		
		4 Down Interface Report	rt	
				Cancel OK

- 1. Click the start immenu and select **Intent Based Automation Center** from the **Intent-Based Automation** tab.
- 2. Click + Add Intent Template to add your intent.
- 3. From the **Add Intent Template** dialog, click **Browse** to select Intent Template from All Intent.
- 4. Search for your Intent Template, select it, and then click **OK**. Your Intent Template is installed in the IBA center.
- 5. Decode Intent Template.

nstalled Intent Templates	Published Intents	Auto Intent	Auto Inter	nt Profile	NetBra	in Download		
ms: 1451 + Add Intent Temp		Filter:						
ms: 1451 + Add Intent Temp	nate	Filter:	All		~	down	×	G Refresh ≡
Intent Template Name		Location		[	Intent [	Decoding		
Default				-				<b>(</b> ) He
No of Days Since Intf is do	nwc	All Network Intents/Sach	hin					
Down Interface Report		All Network Intents/Sach	hin	Intent Ter	nnlato N	ame: Down Ir	nterface Report	
DoctorDave					npiate N	ame. Down in	пенасе кероп	
Check for assigned and d	own	All Network Intents/Eng	~	Dasic	Clor	ed Intents		
Decode Now			× 'Doct					
depending on the	e number of qualit	ned devices.	(Hen	OH	Recurring	Decode		
Decoding Settings	seine	Cancel OK		0	Dne-Time	Intent Baseline Decode g Settings	e Periodically	214 Devices decoded
	seine	Cancel OK		0	Update Dne-Time Decodinț	Decode	e Periodically	
Decoding Settings	Use live data.	Cancel OK		0	Update Dne-Time Decodinț	Decode	e Periodically	
Decoding Settings Data Source:				0	Update Dne-Time Decodinț	Decode	e Periodically	
Decoding Settings Data Source:	Use live data. rom Current Baseline		je.	0	Update Dne-Time Decodinț	Decode	e Periodically	
Decoding Settings Data Source: () Live Data () Use config files f	Use live data. irom Current Baseline Use the recent d		р. 1е.	0	Update Dne-Time Decodinț	Decode	e Periodically	

- a) Select your Intent Template.
- b) In the right pane, click **Decode Now**. From the **Decode Now** dialog, tick the **Update Intent Baseline** checkbox and then click **OK**.

The message "Intent decoding request has been sent successfully." Is displayed.

- c) Click **Decoding Settings**, tick the **Use config files from Current Baseline** checkbox, and click **OK**.
- d) Click **Apply** to apply the settings.

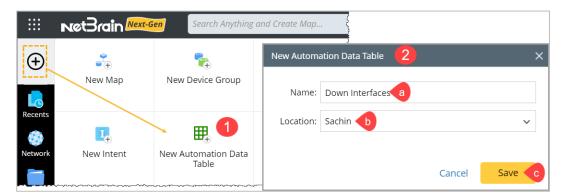
It may take some time to decode the devices if there are many devices in the device group. The final results show that 215 devices are decoded.

<b>I</b>	Intent Based Autor	nation C	enter						
In	nstalled Intent Templates	Published	d Intents	Auto Intent	Auto Intent P	rofile Ne	etBrain D	ownloa	d
lten	ns: 1388 + Add Intent Temp	olate Filter:	All	~	down interfa	ace	×	G R	tefresh ≡
	Intent Template Name	Loc	ation		Intent Dec	Decoded	Auto I	nten	Cloned Int.
-	Default								2
	Down Interface Report	A II A	Network Int	ents/Sachin	Last Deco	215			0
-	DoctorDave								
	Check for down interface	s All i	Network Int	ents/Engineer/Doc	Last Deco	152			135
									2

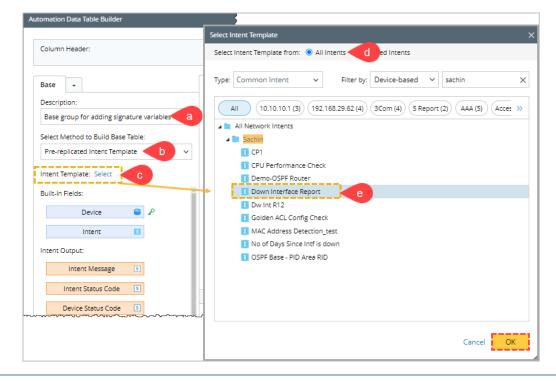
### 5.3.2.2 Build Base Table

Follow the step-by-step instructions to build the base table with the **Signature Variables**:

- 1. Click the plus icon 🕑 and click the **New Automation Data Table** option.
- 2. In the New Automation Data Table popup:
  - a) Enter ADT name, **Down Interfaces**.
  - b) Select the **Location** you wish to store the ADT.
  - c) Click **Save** and wait a moment for ADT to open.

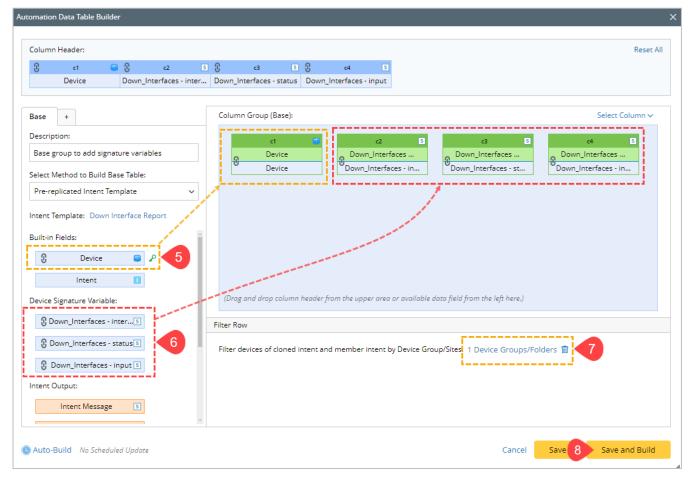


- Click Table Builder to open Automation Data Table Builder to define the Base Group.
   Base Group tab settings:
- 4. Under the **Base** tab of Automation Data Table Builder, define the following settings:



- a) Input descriptions for the base table to describe its use and function.
- b) Make sure the method, Pre-replicated Intent Template, is selected.
- c) Click the **Select** link to select the created NI (*Down Interface Report*) for building the base table with signature Variables.
- d) Click **All Intents** to access your Intent.
- e) Select your Intent and click **OK**.
- 5. From the **Built-in Fields** section, drag the **Device** and **Intent** column into the **Column Group** (**Base**) pane.
- 6. From the **Device Signature Variable** section, drag all the Signature Variables one by one into the **Column Group (Base)** pane.
- 7. In the **Filter Row** pane, click **Select** to add a device group.
- 8. Click Save and Build to save all the base table settings.

In the **Build Table** dialog, select the column groups to be built, select a **Log** mode, and then click **Build** to save the changes.



Hanager Automation Data Table Manager >> B Down Interfaces Table Builder 08:02 PM 🔍 Rebuild Table Add Data Manually 🗸 📃 🚮 Last Description: Type description here... Items: 316 Rows 4 Columns Q TAdvanced Filter: Undefined Search.. No. Device S Down\_Interfaces - interface\_na... S Down\_Interfaces - status S Down\_Interfaces - input US-BOS-SW2 Ethernet1/0 administratively down never

down

administratively down

never

You will see that the columns are populated with the data.

Serial0/2/0

Serial0/3/0

Serial0/3/1

BRI1/0

BRI1/0:1

BRI1/0:2

BRI1/1

BRI1/1:1

BRI1/1:2

BRI1/2

BRI1/2:1

BRI1/2:2

1

2

3

4

5

6

7

8

9

10

11

12

13

BSTX

Edit the Table Column to display the specific name of the Signature Variables. Repeat this step 9. to rename two more Signature Variables columns.

	Į	Table Build	ler	Last Updated at: 06/21/20	24 11:40 AM	🍳 Rebuild Table
Edit Table Column	1		×			
Display Name:	Interface_Name			S Down_Interfaces - interface	e_name	
Column Name:	Down_Interfacesinter	face_name		Ethernet1/0	Edit	t 🦯
Data Type:	E Garage		1	Serial0/2/0	Del	د د
Data Type:	s String	~		Serial0/3/0	Set	as Table Key
	Cance	OK		Serial0/3/1	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	

Help

C

≡

## 5.3.2.3 Add Device Property Columns in ADT

Use the function call method to add **Device Property** in ADT.

- 1. Click **Table Builder** to open the **Automation Data Table Builder**.
- 2. Click + to add a new tab to add the **Device group**.
- 3. From the **Select Method** to **Build Group Table** dropdown, select the **Function Call** method.
- 4. From the **Source Column** dropdown, select **Device**.
- 5. From the **Built-in Fields** section, drag and drop **Hostname**, **Mgmt** IP, **Vendor**, **Model**, and **Software Version** into the **Column Group** (**Device Type**) pane.
- 6. Click **Save and Build** to add device property columns in ADT.

tomatio	on Data Table B	uilde	r													
	in Header:	-	0		_	0				-	-	_	~		-	Reset A
C	c1	8	0	c2		0	c3 Status	5 🕄		s ()	c5	S	U	c6 Marcat ID	S	
C	Device	s	0	Interface_Name		£	c9	5	Input		Hostname			Mgmt IP		
0	Vendor		0	Model		-	oftware Versi									
Base	Device Gro	) ┥		2		Col	umn Group (D	Device Group)	:						Select	t Column 🗸
Descri	otion:					Ē							-			
	ew device grou	ps			ן ר		c5 Hosti	name	c6 Mgn	nt IP	5		7 ndor	5		
					-		8	name	Mgn		8		ndor			
	Method to Bui		oup Ta		۱ ۲		c8	S	c9		S					
Fund	tion Call	3		·			Mo	odel 3	Software	Versio						
Colum	in Type:						0	del	Software							
Sel	ect Property fro	om Sp	ecifie	ed Column		÷.								i.		
⊖ Cor	nvert Column V	alue	bv Fu	nction			1									
Select	One Column o	FADI	as th	e Source Column	1:											
Sou	urce Column:	Devi	e 🖣	4 ~	1											
O Sea	arch Existing Ma	ap Se	lect													
	n Fields:	1														
					Î											
C	Hostnar	ne	L	5												
Û	Mgmt I	Ρ	[	5												
	Mgmt Interf	ace	[	s J												
	Device Typ	e	Г	s												
0																
C	Vendo	r	L	5												
Û	Mode		[	s												
0	Software Ve	ersion		5		(5	)rag and dran	column heade	r from the upper ar	o or o	vailahle data field	l from t	ho loft	horo )		
				-	÷	10		coranni neolue			sonable doco jielo	10111	e.rejt			
													_			
Auto	-Build No Sch	edule	d Up	date								Cano	el	Sav 6	Save	e and Build

You will see the device group columns are populated with the data.

-	own Interfaces ription: Type description here	Table Builder	Last Updated at: 06/21/2	024 08:02 PM 🛛 🌯 Rebuild Tab	Group				Add Data Manually 🗸 📃
Items	:: 316 Rows 9 Columns				Columns		Search		Q T Advanced Filter: Undefined
No.	Device	SInterface_Name	Status	s Input	5 Hostname	SMgmt IP		S Vendor	S Model
1	US-BOS-SW2	Ethernet1/0	administratively down	never	US-BOS-SW2	10.8.1.242		Cisco	3560E
2	BSTX	Serial0/2/0	down	never	BSTX	172.24.32.209		Cisco	2811
3	BSTX	Serial0/3/0	administratively down	never	BSTX	172.24.32.209		Cisco	2811
4	BSTX	Serial0/3/1	administratively down	never	BSTX	172.24.32.209		Cisco	2811
5	BSTX	BRI1/0	administratively down	never	BSTX	172.24.32.209		Cisco	2811
6	BSTX	BRI1/0:1	administratively down	never	BSTX	172.24.32.209		Cisco	2811
7	BSTX	BRI1/0:2	administratively down	never	BSTX	172.24.32.209		Cisco	2811
8	BSTX	BRI1/1	administratively down	never	BSTX	172.24.32.209		Cisco	2811
9	BSTX	BRI1/1:1	administratively down	never	BSTX	172.24.32.209		Cisco	2811
10	BSTX	BRI1/1:2	administratively down	never	BSTX	172.24.32.209		Cisco	2811
11	BSTX	BRI1/2	administratively down	never	BSTX	172.24.32.209		Cisco	2811
12	BSTX	BRI1/2:1	administratively down	never	BSTX	172.24.32.209		Cisco	2811
13	BSTX	BRI1/2:2	administratively down	never	BSTX	172.24.32.209		Cisco	2811

# 5.3.3 Export ADT to CSV file

ADT can be exported as a CSV file for documentation.

				Hel
Table Builder	Last Updated at: 06/21/2	024 08:02 PM 🛛 🌯 Rebuild Tab	ble	Add Data Manually V
			Search	Lock Settings       Q       Execution log
ne	s Status	s Input	s Hostname	S Mgn Export Export to CSV Only
	administratively down	never	US-BOS-SW2	10.8.1.: Export Datasets to File
	down	never	BSTX	172.24. Dataset Tag Settings
	administratively down	never	BSTX	172.24. Table Settings
	administratively down	never	BSTX	172.24.32.209 Cisco
	administratively down	never	BSTX	172.24.32.209 Cisco

The exported CSV file will be saved automatically to your computer's default download location, typically the "Downloads" folder (e.g., **C:\Users<your username>\Downloads**).

1 Device	Interface_Name	Status	Input	Hostname	Mgmt IP	Vendor	Model	Software Version
2 US-BOS-S	Ethernet1/0	administratively down	never	US-BOS-SW2	10.8.1.242	Cisco	3560E	15.2(HI_20170202)FLO_DSGS7
3 BSTX	Serial0/2/0	down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
4 BSTX	Serial0/3/0	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
5 BSTX	Serial0/3/1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
6 BSTX	BRI1/0	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
7 BSTX	BRI1/0:1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
8 BSTX	BRI1/0:2	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
9 BSTX	BRI1/1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
10 BSTX	BRI1/1:1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
11 BSTX	BRI1/1:2	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
12 BSTX	BRI1/2	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
13 BSTX	BRI1/2:1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
14 BSTX	BRI1/2:2	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
15 BSTX	BRI1/3	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
16 BSTX	BRI1/3:1	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
17 BSTX	BRI1/3:2	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4
18 BSTX	BRI1/4	administratively down	never	BSTX	172.24.32.209	Cisco	2811	12.4(15)T4

PART 2 - Intent Based Automation Usecase Study

# 6 Network Assessment Case Study: Security Assessment

Network security is the protection of the underlying networking infrastructure from unauthorized access. The National Institute of Standards and Technology (**NIST**) publishes a set of configuration standards for security, and CVE shares vendor-reported vulnerability and exposure. The intent-based automation provides an ideal platform to enforce network security standards and best practices.

In this section, you will learn how to check NIST compliance against your network devices. You will create Intents to check the configurations such as VTY, AAA, and password policy against the NIST standard. The Base ADT will be created via the method **Devices of Device group** you learned in <u>Section 5 Prerequisites</u>. A Summary Dashbaord will be created to show the NIST compliance for your network.

In the second part of this section, you will learn how to create Intent for CVE issues to check device potential security risk. You will create two different Intents to check if a new CVE is affecting certain devices and the other to show the CVEs that may affect a device.

The workflow of the security assessment will be as follows:



# 6.1 Check NIST Compliance and Vulnerability

You will create a dashboard to check the NIST Compliance for the following configurations:

- 1. Devices to allow Telnet (check Line VTY)
- 2. AAA Configuration
- 3. Unused ports but no shutdown
- 4. Device Password Policy (not encrypted, etc.)
- 5. Line Session Timeout

### The final ADT will be:

	ption: Type description here				🔍 Rebuild Table					Add Data Manually 🗸 📃
		Device Grou	p Columns			Au	tomation Columns -	NIST		
No.	62 Rows 7 Columns				_			Search	С	Advanced Filter: Undefined
	Device	S Device Type	Telnet vty check		AAA Config Check		Device Unused Ports Config C	Device Password Policy		Device Line Session Timeout
-	Berlin-R1	Cisco Router	Check Vty Telnet Access Berlin-R1	۲	AAA Config Check Berlin-R1	۲	Device unused Ports config chec 💿	Device Password Policy Config C.	. 0	Device Line Session Timeout
	Berlin-vEdge	Cisco IOS Switch	Check Vty Telnet Access Berlin-v	0	AAA Config Check Berlin-vEdge	0	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout
	DE-MUC-CR01-01	Cisco Router	Check Vty Telnet Access DE-MUC	•	AAA Config Check DE-MUC-CR01	•	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout
	DE-MUC-CR01-02	Cisco Router	Check Vty Telnet Access DE-MUC	•	AAA Config Check DE-MUC-CR01	0	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout
	DE-MUC-CW01-01	Cisco IOS Switch	Check Vty Telnet Access DE-MUC	0	AAA Config Check DE-MUC-CW0	0	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout
	DE-MUC-CW02-01	Cisco IOS Switch	Check Vty Telnet Access DE-MUC	•	AAA Config Check DE-MUC-CW0	0	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout
	DE-MUC-CW03-01	Cisco IOS Switch	Check Vty Telnet Access DE-MUC	•	AAA Config Check DE-MUC-CW0	•	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout
	ISP-P02	Cisco Router	Check Vty Telnet Access ISP-P02	0	AAA Config Check ISP-P02	0	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout
	ISP-PE01	Cisco Router	Check Vty Telnet Access ISP-PE01	0	AAA Config Check ISP-PE01	0	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout
0	ISP-PE02	Cisco Router	Check Vty Telnet Access ISP-PE02	•	AAA Config Check ISP-PE02	0	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout
1	ISP-PE03	Cisco Router	Check Vty Telnet Access ISP-PE03	•	AAA Config Check ISP-PE03	•	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout
2	ITE_EXTEND	Cisco IOS Switch	Check Vty Telnet Access ITE_EXTE	•	AAA Config Check ITE_EXTEND	•	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout
3	JP-TYO-CR01-01	Cisco Router	Check Vty Telnet Access JP-TYO-C	•	AAA Config Check JP-TYO-CR01-01	•	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout
1	JP-TYO-CR01-02	Cisco Router	Check Vty Telnet Access JP-TYO-C	•	AAA Config Check JP-TYO-CR01-02	•	Device unused Ports config chec	Device Password Policy Config C.	. •	Device Line Session Timeout
5	JP-TYO-CW01-01	Cisco IOS Switch	Check Vty Telnet Access JP-TYO-C	•	AAA Config Check JP-TYO-CW01-01	0	Device unused Ports config chec 💿	Device Password Policy Config C.	. •	Device Line Session Timeout

This section includes the following main steps:

- Build ADT Base Table
- <u>Create Network Intent: Check Vty Telnet Access</u>
- <u>Create NI: AAA Config Check</u>
- <u>Create NI: Device Unused Ports Config Check</u>
- <u>Create NI: Device Password Policy Config Check</u>
- <u>Create NI: Device Line Session Timeout</u>
- Create Intent and Summary Dashboard

### 6.1.1 Build ADT Base Table

In the first step, you will create an ADT base table using the method **Devices of evice Group**. Make sure the Device Group is created based on your requirements.

#### 6.1.1.1 Build Base ADT

To build the base table with the data in devices, follow the steps below:

- 1. Click the plus icon 🕑 and click **New Automation Data Table**.
- 2. In the New Automation Data Table popup:
  - a) Enter the ADT Name.

- b) Select the **Location** you wich to store the ADT.
- c) Click **Save** and wait a moment for ADT to open.

	vetBrain <sup>Next-</sup>	Search Anything	and Create Map	
Ð	New Map	New Device Group	New Automation Data Table	×
Recents	New Map	New Device Group	Name: NIST Compliance	
	I.	È 🗒 🚺		~
Network	New Intent	New Automation Data Table	N Cancel Save	С

- 3. Click **Table Builder** to open **Automation Data Table Builder** to define the Base Group.
- 4. Base Group tab settings.
- 5. Under the **Base** tab of Automation Data Table Builder, define the following settings:

Column Header:		Reset All
n c1 😂 n c2	2	
Device Device Ty		
Base +	Column Group (Base):	Select Column 🗸
Description:	c1 📫 c2 5	
The base ADT for NIST compliance	Device Device Type	
Select Method to Build Base Table:	Device Device Type	
Devices of Device Group	J	
Select Devices by Device Groups My Device	Gro Select Device Groups	×
Built-in Fields:	Search	Q
🕄 Device 🖀 🖉		
Device Group 5	All Device Groups	A
	▲ 🖃 My Device Groups	
Hostname S		
	Cisco Devices (96)	
Mgmt IP S	Firewall (13)	
Mgmt IP S Mgmt Interface S	Generation Firewall (13)	
Mgmt Interface S		
Mgmt Interface S	▲ □ ► Shared Device Groups	
Mgmt Interface S	Shared Device Groups     Ahmed     Anurag     Sessment Reference I	Library
Mgmt Interface S	Shared Device Groups     Ahmed     Anurag     Assessment Reference I     Automation Library	
Mgmt Interface S Device Type S Vendor S	Shared Device Groups     Ahmed     Anurag     Assessment Reference I     Automation Library     Automation Library Kun	hal
Mgmt Interface 5 Device Type 5 Vendor 5 Model 5	Shared Device Groups     Ahmed     Anurag     Assessment Reference I     Automation Library	hal

- a) Input **Description** for the base table to describe its use and function.
- b) Select **Method**, the **Devices of Device Group** from the dropdown to build the base table.
- c) Click the **Select** link to select the created device group for building the base table with the devices in the device group, and then click **OK**.
- d) Within the **Built-in Fields** section, choose each column and move it into the **Column Group** (**Dase**) pane.

Build an ADT table with the following columns: **Device** and **Device Type**.

e) Click **Save and Build**. The **Build** Table dialog appears, define the settings as per your preferences and then click **Build** to save all the settings.

<b>F</b> /	Automation Data Table Mana	ger		
•	NIST Compliance	Table Builder	Last Updated at: 07/04/2024 12:39 PM	🌯 Rebuild Table
Des	scription: Type description here			
lter	ms: 62 Rows 2 Columns			
No.	. Operation 2015		s Device Type	
1	Berlin-R1		Cisco Router	
2	Berlin-vEdge		Cisco IOS Switch	
3	DE-MUC-CR01-01		Cisco Router	
4	DE-MUC-CR01-02		Cisco Router	
5	DE-MUC-CW01-01		Cisco IOS Switch	
6	DE-MUC-CW02-01		Cisco IOS Switch	
7	DE-MUC-CW03-01		Cisco IOS Switch	
8	ISP-P02		Cisco Router	
9	ISP-PE01		Cisco Router	
10	ISP-PE02		Cisco Router	
11	ISP-PE03		Cisco Router	
12	ITE_EXTEND		Cisco IOS Switch	
13	JP-TYO-CR01-01		Cisco Router	

The Base ADT will be like:

# 6.1.2 Create Network Intent: Check Vty Telnet Access

In this section, you will create a Network Intent to check the VTY configurations (**Check Vty Telnet Access**) from the **Intent Manager** and use the NI in the **Intent Replication Wizard** to replicate the Intent to all the devices in the ADT table. The intent will check whether the telnet access is disabled and the SSH access is enabled in the VTY configurations.

Follow the step-by-step instructions to create an Intent.

- 1. Create a new intent from the **Intent Manager** and set the intent name (title) as **Check Vty Telnet Access**.
- 2. Add a new target device, such as device US-BOS-R1.

Network Intent (Edit Mode)			×
I Check Vty Telnet Access	Diagnosis Tree	Run with Live Data	Save 🛛 Help 🚍
Type description here			💑 Intent Map: Select 🗸 🗸
I Seed Logic	tion Logic		
🚳 + Device 2		Intent Variables	: Manager 🕴 Tag: + Add 🗐
Select Devices			×
Select Devices by:   Device Type  Device Type Device Type  Device Type  Device Type  Device Type	evice Group 🔿 Site	•	1 Devices Selected
All Device Types	v us-bos-cw	>	C US-BOS-CW01-01
Hostname	Mgmt IP Vendo	or Model	
US-BOS-CW01-01	192.168.0.19 Cisco	Catalyst 29xxStack	
US-BOS-CW01-02	10.10.0.9 Cisco	3560E	
2 US-BOS-CW02-01	10.10.0.4 Cisco	3560E	
2 US-BOS-CW02-02	10.10.0.5 Cisco	3560E	
US-BOS-CW03-01	10.10.0.10 Cisco	3560E	»
2 US-BOS-CW03-02	10.10.0.11 Cisco	3560E	
2 US-BOS-CW04-01	10.10.0.12 Cisco	3560E	<
a US-BOS-CW04-02	10.10.0.13 Cisco	3560E	*
			· · · · · · · · · · · · · · · · · · ·
			Cancel OK

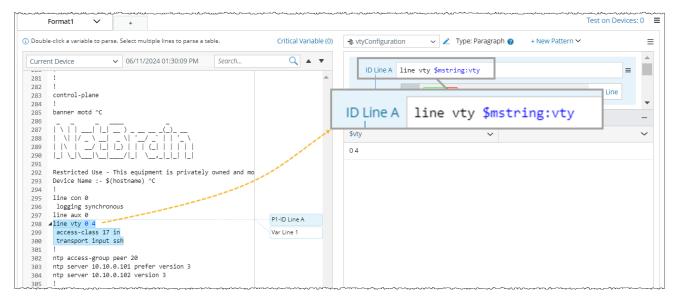
## 6.1.2.1 Add Config Diagnosis and Parse the VTY Configurations

You will parse the VTY configurations with a Config Diagnosis:

- 1. Click + Add Config Diagnosis to open the Configuration Diagnosis window.
- 2. Click **Retrieve** to retrieve the data from the **Live Data**.
- 3. The source data is displayed in the **Define Variable** pane. You can edit this data depending on the use case.
- 4. In the **+ New Pattern** dropdown, select the **Paragraph** parser to parse VTY configurations. The Paragraph parser is used since there can be multiple VTY lines.
- 5. Define **ID Line A** for the VTY configurations:
- 6. In the configurations, find the configuations of VTY. Double-click the vty number,

for example, **0 4** of the sample data. The Variable in the ID Line A **vty \$mstring:vty** will be created.

Check the Output.



- 7. Define the configurations of a vty line as a variable. Here, we will use the function *LineByVariables*, which can parse multiple lines into one variable:
  - a) Click + Parse Lines.
  - b) Enter Variable name, *vty\_config*.
  - c) From **The line between**, parse the line between **\$vty** to the **End.**
  - d) Check the **Pattern** and click **Apply**.

ID Line A vty \$mstring:vty		
Output a + Parse Lines	Parse Lines         Name:       vty_config         O The line of variable:       Select         Image: The line between:       Svty         The line between:       Svty         O The line contains keyword:       Enter keyword	×
	Output:         Svty       Svty_config P         04       line vty 04 access-class 17 in transport inpu         Pattern:       LinesByVariable[\$vty_config]:\$vty-         Card       Apply	

- e) To parse the whole vty configuration line, modify the pattern to LinesByVariable[\$vty\_config]:\$vty-
- f) Add the **End of Paragraph** using an exclamation mark (!) to parse the whole vty configuration line.

Type: Paragraph	Cancel	Apply	
ID Line A vty \$mstring:vty 298 line vty 04	Add Er	arent art Line nd Line	Ţ
Var Line 1 LinesByVariable[\$vty_config]:\$vty- 298 line vty 0 4		> 1 Line	8
End of Paragraph !			Û
Output + Parse Lines			_
\$vty	s-class 17 in t	ransport inpu	~ .t s

- 8. Verify the **Ouput** and click **Apply**.
- 9. In the **Confirmation** popup, click **Apply and Continue** to save the parser.
- Click the pen icon, rename the parser name from Paragraph1 to line\_config, and then click OK.

1	-@ P	aragraph1	V K Type: Paragraph	
1		Rename	4	×
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		Name:	line_config	
			Cancel	К

The final Variables window will be:

line_config	V 🗶 Type: Pa	aragraph 🕜	+ New Pattern 🗸		-
ID Line A vt	y \$mstring:vty 98 linevty04			∃ > 1 Line	=
Var Line 1	LinesByVariable[\$v - 298 line vty 0 4	/ty_config]	:\$vty-	∃ > 1 Line	=
End of Paragra	ph !			ť	ÿ
Output + Pa	arse Lines	_			-
\$vty		<ul> <li>✓ \$vty_</li> </ul>	config		~
0.4 line vty 0.4 access-class 17 in transport inpu					ut s

### 6.1.2.2 Define Intent Variable for Seed Logic

In the diagnosis, you will loop through each VTY line (each row of the table **\$line\_config**) and check whether the telnet access is allowed for each line. However, you do not want to create multiple alerts if the multiple VTY lines allow the telnet access. To create only one alert per device,

- you can define an intent variable, **\$telnet\_access**, with the default value 0.
- Set this variable as **1** if the telnet access if the VTY line allows telnet access while you loop through the table.
- Then, outside the loop, check this intent variable and create an alert.

To add an intent variable:

- 1. Click **All Intent Variables** in the bottom right corner.
- 2. Click the **Intent** section and then click **+ Add Intent Variable**.
- 3. In the **Add Intent Variable** popup fields, enter the following values:

Fileds	Value
Variable Name	telnet_access
Туре	number
Initial Value	0

4. Click **OK** to save the Intent Variable.

		Intent Variables for Seed Lo	ogic				
		Intent Variable	Use Auto	omation Data Table	Task Variable		
ຊໍ້ 299	access	+ Add Intent Variable	2 Add Intent	t Table			
300 301 302 303	trans ! ntp ace	▶ 🚺 Intent		Add Intent Variable			×
	ntp ser ntp ser			Variable Name:	telnet_access		
304 305 306 307 308 309	! ! end	▷ , ↓ line_config ▷ / US-BOS-CW01-01		Туре:	number		~
308			~~~~~~	Initial Value:	0		
	4	1		i			
Help		ent Variables	~~~~			Cancel	ОК 4

5. Click **Close** to exit from the **Intent Variable for Seed Logic** window. The Intent Variable you just added is displayed.

Intent Variables for Seed Lo	ogic			×
Intent Variable	Use Automation Data Table	Task Variable		
+ Add Intent Variable	+ Add Intent Table			
🔺 🚺 Intent				V
🐹 telnet_access	Type: number		Initial Value: 0	
🖌 🥏 US-BOS-CW01-01	ō	Configuration		
▷ 📜 line_config				
▶ 🚄 US-BOS-CW01-01	В	Built-in Data		
				Close

### 6.1.2.3 Define Diagnosis

As explained earlier, we will define two diagnostics: **Check VTY telnet access** to loop through VTY lines and set the intent variable *\$telnet\_access*, and **Create Alert** to set the status code according to the value of *\$telnet\_access*.

### **Diagnosis 1: Check VTY telnet access**

- 1. Go to the **Define Diagnosis** section to define the diagnosis logic.
- 2. Click **Add Diagnosis** to define conditions and Intent output message.
- 3. Enter the diagnosis name, e.g., **Check VTY telnet access** and select an Anchor **\$vty\_config** from the dropdown.
- 4. Tick the **Loop Table Rows** checkbox and select the Table Variable (*line\_config*) and the Table Key (*vty\_config*) from the dropdown.

- 5. Define **If** condition as detailed in the image:
  - a) Variable *vty\_config* Contains All.
  - b) Variable vty\_config Contains telnet
  - c) Variable **vty\_config** Does not contain **ssh**
  - d) Boolean Expression: **A or B or C.**

Add Note Add Diagnosis Can also click a variable on the left to add at Name: Check VTY telnet access Anchor: line_config.\$vty_config Type description of the diagnosis Loop Table Rows I line_config V Table Key: vty_config V 4 If A US-BOS-C Current V 5 A US-BOS-C Current V 10 B US-BOS-C Current V 10 Vty_config V Contains V All V Loop Table Rows Current V 10 Contains V 10 C	2. Define Diagnosis						
Type description of the diagnosis         Loop Table Rows       Ine_config         If         US-BOS-C       Current         vty_config       Contains         Wistor       All         Vy_config       Contains         Vty_config       Contains         Vty_config       Contains         Vty_config       Contains         Vty_config       Contains	Add Note Add Diagnosis 2 Can also click a variable on the left to add automatic						
<ul> <li>Loop Table Rows I line_config v Table Key: vty_config v 4</li> <li>If</li> <li>US-BOS-C Current v</li> <li>vty_config v Contains v All v</li> <li>US-BOS-C Current v</li> <li>US-BOS-C Current v</li> <li>telnet v</li> </ul>	~						
<ul> <li>✓ If</li> <li>A  US-BOS-C Current ~ 5</li> <li>a  vty_config  ~ Contains ~ All ~</li> <li>B  US-BOS-C Current ~</li> <li>b  vty_config  ~ Contains ~ telnet ~</li> </ul>							
A Image: US-BOS-C Current ∨   a vty_config ∨   Contains ∨   All ∨   B US-BOS-C Current ∨   b vty_config ∨   Contains ∨   telnet ∨	¢						
B ₂ US-BOS-C Current ∨ b vty_config ∨ Contains ∨ telnet ∨							
b vty_config ~ Contains ~ telnet ~	Ŵ						
C _ US-BOS-C Current ✓	Ē						
vty_configvDoes not containssh	Ē						
D Select Variable ~							
d Boolean Expression: A or B or C							

- 6. Set Intent Variable.
  - a) In the **Add Logic** dropdown, select Advanced > Set Intent Variable.
  - b) In the Set Intent or Task Variable dropdown, select *telnet\_access* ad Intent Variable and enter the value 1.

	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~ 6 ~ ~ ~ ~ ~ ~ ~	~~~~~~~~
Intent Data V	liew	Export CSV Report	0	برید. در در د
Draw Map	>	Operate on Table		Incident =
Send Email		•	0.54%2.60	
Follow-up Int	ent	Set Intent Variable		
Set Intent Ba	seline	Call Qapp		
		Call Webhook API		
Advanced	>	Break Current Loop		
Add Logic A		Hyperlink		
		Add Command to Benchmark	rk	
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		······································	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
	∨ Then		÷.	
	📄 Diagnosis N	lessage:		🗌 Save to Incident 📃
	<b>~</b>	\$intf is down		Pop up
	Set Stat	us Code for Device:		Move Down
	Set Stat	us Code for Intent:		Delete C
	i			
	🚬 Set Intent o	r Task Variable:		≡
	telnet_acces	ss 🗸 = 1 🚺		
	L			

c) Delete the **Diagnosis Message** section.

### **Diagnosis 2: Create Alert**

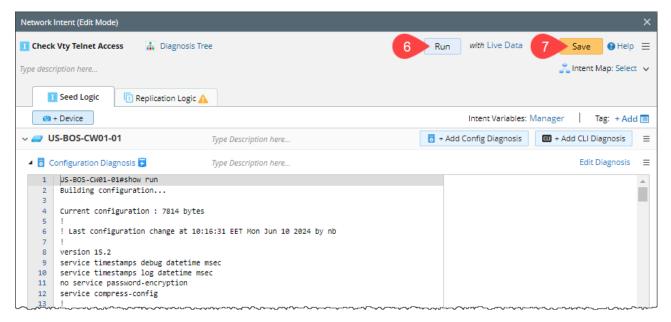
- 1. Click **Add Diagnosis** again to define conditions and Intent output message.
- 2. Enter the diagnosis name, e.g., **Create alert.**
- 3. Define the **If** condition as detailed in the image: Intent Variable *telnet\_access* Equals **1**.

	2. Defir	e Diagnosis							
۹ 🗸 ۸	📄 Add N	ote 🗾 Add D	iagnosis 🧲	1	Can also click	a variable	on the left to ad	d auto	matior
Create alert	Name:	Create alert				Anchor:	Select Variable	•	~
Check VTY telnet ac		Type description of	the diagnosis	·					
	🗌 Loop	Table Rows							
	∽ If								
	А		Current		3				
		telnet_access	~	Equals	~	Example	: 1	~	Î
	В	Select Variable	~						
	Create alert	Create alert Check VTY telnet ac Add Name: Check VTY telnet ac	Create alert Check VTY telnet ac Name: Create alert Type description of Loop Table Rows If telnet_access	Create alert Check VTY telnet ac Add Note Add Diagnosis Name: Create alert Check VTY telnet ac Add Note Add Diagnosis Type description of the diagnosis Loop Table Rows If Current telnet_access	Create alert Check VTY telnet ac Add Note Add Diagnosis  Add Diagnosis Add Diagnosis Add Diagnosis  Add Diagnosis Add Diagnosis Add Diagnosis Add Diagnosis Add Diagnosis Add Diagnosis Add Diagnosis Add Diagnosis Add Diagnos Add	Create alert Check VTY telnet ac Add Note Add Diagnosis Create alert Check VTY telnet ac	Create alert Check VTY telnet ac Add Note Add Diagnosis Create alert Check VTY telnet ac Anchor: Type description of the diagnosis Check VTY telnet ac Loop Table Rows If Loop Table Rows Lif Lelnet_access Lequals Lequals Lexample	Create alert Check VTY telnet ac Add Note Add Diagnosis Create alert Check VTY telnet ac Anchor: Select Variable Type description of the diagnosis Loop Table Rows If Loop Table Rows Lif Lelnet_access Equals Example: 1	Create alert Check VTY telnet ac Add Note Add Diagnosis Create alert Check VTY telnet ac Anchor: Select Variable Type description of the diagnosis Loop Table Rows If Loop Table Rows Equals Equals Example: 1

4. Define Intent output: Enter a message under the **Then** and **Else** output areas to appear as the result of the diagnosis.

~ Then 🛛 a	4	
📄 Diagnosis M	essage:	□ Save to Incident ≡
~	<pre>\$this_device Telnet is enabled</pre>	
Set Stat	us Code for Device:	
Error	\$this_device Telnet is enabled	
Set Stat	us Code for Intent:	
\rm () Error	\$this_device Telnet is enabled	
✓ Else C Diagnosis M	essage:	☐ Delete
	\$this_device Telnet is disabled	
🛛 🔽 🛐 Set Stat	us Code for Device:	
Succes	s 🗸 \$this_device Telnet is disabed	
Set Stat	us Code for Intent:	
Succes	ss 🗸 \$this_device Telnet is disabed	
Add Logic 🗸		
+ Add ElseIf		
		Cancel 5 Apply

- 5. Click **Apply** to save all the diagnosis settings and then close the **Configuration Diagnosis** window.
- 6. In the **Network Intent (Edit Mode) window**, click **Run** to execute the Intent and check whether it is working as per your configuration settings.
- 7. Click **Save** to save the Network Intent successfully and then close the window.



## 6.1.2.4 Use Intent Replication Wizard

In this section, you will use the **Intent Replication Wizard** to add the Automation Column to the ADT that you have created, **NIST Compliance**.

- 1. In the Network Intent (View Mode) window, go to the  $\equiv$  menu and open the Intent Replication Wizard.
- 2. In the **Seed Intent** tab, check for your NI (*Check Vty Telnet Access*) and then click **Next** to go to the **Define ADT** tab.

I Check Vty Telnet Access	👶 Open 📄 0 🔥 0 🗶 Edit
Result: No History Data 🕢 💽 🗹 No result available because this intent has not been executed.	Named Tag View Abstract Run Settings Data Clean Settings
seed Intent 2 Define ADT Replication Settings	Edit Save as Delete Refresh
	Export Share to Incident Publish Intent Intent Replication Wizard
Seed Intent: Check Vty Telnet Access Select	Auto Intent Wizard

- 3. Define ADT:
  - a) In the **Define ADT** tab, click **Use an Existing ADT** option and select the ADT from your folder.
  - b) In the **Replicate Intent to** section, enter the Intent group as an Intent column group name.
  - c) Click **Next** to go to the **Replicate Intent** tab.

	Intent Replication Wizard - Check Vty Telnet Access				
	Seed Intent	Define	ADT	Replication Settings	
Select Aut	tomation Data Table	Create a New AD	דנ	Use an Existing ADT a	
	Empty Automation Library Empty Staging Automation Library NetBrain Essential Automation Library	Automation Data Table:		VTY check b	
	Plugin Test Representative Methodology Sachin_TW 職 NIST Compliance	Slicate on Device Column:	Device		
	Staging Automation Library TEST Yan Wu Automation Library Yan Wu Staging Automation Library Z - Automation Library Test - Do Not Touch		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	

- 4. Replication Settings:
  - a) In the **Intent Qualification** section, click the **Select** link to add Device Group. Use the device group that you have used during ADT creation.
  - b) Add the device group and then click **OK**.
  - c) Click **Next** to go to the **Replicate Intent** tab.

Intent Replication V	Vizard - Check Vty Telnet Acces	s		 [
Se	eed Intent	Define ADT	Replication Settings	1
Intent Qua	ilification: 💿 via Device Grou	ps/Sites: Select V O via Dy	namic Search: Undefined	
Define Ma Item: 1	cro Variables and Rules for Th	eir Subst New Device Group	Add Device Group	×
Seed Dev	vice OS-CW01-01	Seed Command Configuration	▲	
hou-oouloonoonoonoonoonoonoonoonoonoonoonoonoon		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	→ → → Shared Device Groups	
			Anurag     Automation Library	~~~~~~~

- 5. Replicate Intent:
  - a) In the **ADT Columns** section, you can see the **Column Name** as **Replicated Intent**. You can change this name to **Telnet vty check**.
  - b) Click **Save and Replicate** to save all the settings.

You can see the **Open Output ADT** option after the successful submission of the replication request.

c) Click **Open Output ADT** to check the replicated Intent column in the ADT Manager.

Int	ent Replication Wizard - Check Vty Telnet	Access					×
	Seed Intent		Define ADT	Replication	Settings	Replicate	Intent 5
	ADT Columns:					Additio	nal Columns 🗸
	Column Data		Column Name		Tag		
	Replicated Intent		Telnet vty check		0 tags		
0							
						Save and	d Replicate b
				Replication Red	quest submitted at: 0	)7/02/2024 01:31 C Op	oen Output ADT
(	Selection Mode: Device-based Replicati	on, ADT: NIST Complia	ance, 0 Macro Variables.			Previous	Finish

The table will populate with devices and the replicated Intents (**Telnet vty check**). Review the new Intent column.

-	T Compliance	Table Builder         Last U	Jpdated at: 07/02/2024 01:3	🌯 Rebuild Table Add Data Manually $\sim~\equiv$	ď
	62 Rows 3 Columns		Search	Q T Advanced Filter: Undefined	0
No.	Device	S Device Type		Telnet vty check	Ξ
1	Berlin-R1	Cisco Router		Check Vty Telnet Access Berlin-R1 🛛 💿	ľ
2	Berlin-vEdge	Cisco IOS Switch		Check Vty Telnet Access Berlin-vEd 💿	
3	DE-MUC-CR01-01	Cisco Router		Check Vty Telnet Access DE-MUC-C 💿	
4	DE-MUC-CR01-02	Cisco Router		Check Vty Telnet Access DE-MUC-C 💿	
5	DE-MUC-CW01-01	Cisco IOS Switch		Check Vty Telnet Access DE-MUC-C 💿	
6	DE-MUC-CW02-01	Cisco IOS Switch		Check Vty Telnet Access DE-MUC-C 💿	
7	DE-MUC-CW03-01	Cisco IOS Switch		Check Vty Telnet Access DE-MUC-C 💿	
8	ISP-P02	Cisco Router		Check Vty Telnet Access ISP-P02 📀	
9	ISP-PE01	Cisco Router		Check Vty Telnet Access ISP-PE01 O	

### 6.1.2.5 Run Intent Once and Rebuild Table

Select the Intent column you added in the last section and **Run** once Intents of this column. Select the **Live Data** as a Data Source.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~		,~~~ <u>~</u> ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	·····›		
	Table Builder	Last Upd	ated at: 07/03/2024	4 10:44 PM		
S Device Type		Telnet vty	Run 1	tails $\equiv$		
{ Cisco Router		Check Vty Telne	t Access Berlin-R1	•		
Cisco IOS Switc	h	Check Vty Telne	t Access Berlin-v	• {		
Cisco Router	Run Inter	nts Once - Telnet	vty check			×
1 Cisco Router	Data So	urce: Live Data				
Cisco IOS Switc	h					
Cisco IOS Switc	h Only	Run Intents if	Intent Device		✓ belongs to Select ✓	
Notification					×	
The intents in the c	urrent ADT co	lumn are exect	uted once now.		A	Cancel OK 2
View Details				OK <	3	

Wait for the system to finish executing all intents and **Rebuild Table** to populate the intent results.

{ } 👪 NIS	ST Comp	bliance	Table Builder	Last Updated at: 07/03/2024	10:44 PM	💐 Rebu	uild Table	{
} Descri	ption:	Type description here				- Careland		j
} Items:	62 Rov	vs 7 Columns			1			10 M 10-10-10
No.	😂 De\	vice	S Device Type	Telnet vty check	- [-	AAA Cor	nfig Check	
1	Berlin	-R1	Cisco Router	Check Vty Telnet Access Berlin-R1	0	AAA Config	Check Berlin-R1	•
2	Berl	Rebuild Table			+	×	Check Berlin-vEdge	•
33	DE-I	·				·····	Check DE-MUC-CR01	0
4	DE-I	Build the column gro	ups: All 5			~	Check DE-MUC-CR01	
5	DE-I		Log: 🔘 Production Mode	Only show major execution proces	is log		Check DE-MUC-CW0	•
6	DE-I		○ Debug Mode	Show all the detailed log			Check DE-MUC-CW0	•
				Cance	В	uild 6		

## 6.1.3 Create NI: AAA Config Check

You can follow the same flow to create, replicate and run the intents for other types of configurations against NIST standards. In this and the next few sections, we will skip the detailed steps of instructions but highlight differences and new concepts. You can refer to the screenshots for each step, accompanied by brief explanations.

In this section, you will create AAA configuration check Intent for your Network devices.

### 6.1.3.1 Define Variables with Visual Parser

- 1. Use the **+ Add CLI Diagnosis** option to retrieve sample data. Alternatively, you can use the **+ Add Configuration Diagnosis** option, which does not require the use of CLI commands.
- 2. If you use **+Add CLI diagnosis**, then use the command, **show run | section aaa** to retrieve the sample data.
- 3. Define the Variable with **Paragraph** parser for **Server IP** and its **Key** value.

The ID line pattern is: ^ server-private \$Tacacs\_Server key \$int:key \$\_dummy

ACI-QA-SW1 show run   section aaa X Retrieve v with Live Data Define Variable 2. Define Diagnosis Format1 V + Test on Devices: 0 = Double-click a variable to parse. Select multiple lines to parse a table. Critical Variable (0) AAA_Servers V X Type: Paragraph • + New Pattern V = Current Device • 06/26/2024 03:25:17 PM Search Q • V ID Line A * server-private \$Tacacs_Server key \$int:key \$_dumy = 1 ACI-QA-SW1 show run   section aaa 2 aaa new-model 3 aaa group server tacass+ ACS 4 server-private \$Tacacs_Server key \$int:key \$_dumy = 1 D Line A * server-private \$Tacacs_Server key \$int:key \$_dumy = 1 D Line A * server-private \$Tacacs_Server key \$int:key \$_dumy = 1 D Line A * server-private \$Tacacs_Server key \$int:key \$_dumy = 2 interver variable to parse variable	ommand Diagnosis	×	
Formatl       +       Test on Devices: 0         D Double-click a variable to parse. Select multiple lines to parse a table.       Critical Variable (0)	ACI-QA-SW1 show run   section aaa X V	ith Live Data	
Double-dick a variable to parse. Select multiple lines to parse a table. Current Device	Define Variable	2. Define Diagnosis	
Current Device    Ob/26/2024 03:25:17 PM Search   I ACI-QA-SMIEShow run   section aaa aaa group server tacacs+ ACS A - server-private 192:168:28:100 key 7 097A1D1858113701592CS ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dummy ID Line A ^ server-private \$Tacacs_Server key \$_dummy ID Line A ^ server-private \$_dummy ID Line A ^ server-p	Format1 🗸 +	Test on Devices: 0	
1 ACI-QA-Skilshow run   section aaa aaa new-model 3 aaa rew-model 3 aaa group server tacass + ACS 4 server-private 192.168.28.100 key 7 097AID1858113701592C5 5 ip tacas source-interface Vlan11 ID Line A ^ server-private \$Tacacs_Server key \$int:key \$_dum	Double-click a variable to parse. Select multiple lines to parse a table.	ritical Variable (0) 🔹 AAA_Servers 🗸 Type: Paragraph 💿 + New Pattern 🗸 🚍	
7 aaa authentication login console group ACS local	ACI-QA-SWI#Show run   section aaa     aaa new-model     aaa group server tacacs+ ACS     asa group server tacacs+ ACS     ip tacacs source-interface Vlan11     aaa authentication login default group ACS local     aaa uthentication login console group ACS local	ID Line A       ^ server-private \$Tacacs_Server key \$int:key \$_dummy         ID Line A       ^ server-private \$Tacacs_Server key \$int:key \$_dummy	\$_dumr
8 aaa authorization exec default group ACS 9 aaa authorization commands 15 default group tacacs+ none 10 aaa accounting exec default start-stop group ACS	9 aaa authorization commands 15 default group tacacs+ none	Output + Parse Lines —	
11     aaa accounting commands 1 default start-stop group ACS     \$Tacacs_Server * * * * * * * * * * * * * * * * * * *			

- Define another Variable using Variable Operator LinesByKeyword to extract all the data related to the Variable. The function LinesByKeyword is identical to LinesByVariable but uses keywords instead of variables to combine lines.
  - a) Click + Parse Lines in the Output pane, and the Parser Lines window appears.
  - b) Set Variable name as **AAA\_New\_Model**.
  - c) Select The line contains keyword option, and enter the keyword, *aaa new-model*.
  - d) Click **Apply** to close the **Parser Lines** window.

Parse Lines	×
Name: AAA_New_Model	
O The line of variable: Select	$\sim$
O The line between: Select v to Select	~
O The line of variable: Select	(×)
\$AAA_New_Model=	
Pattern:	
LinesByKeyWord[\$AAA_New_Model]:aaa new-model	
Ca d App	ply

## 6.1.3.2 Define Compound Variable

In the diagnosis, we will check whether whether AAA new model is configured. If the variable **AAA\_New\_Model** is equal to "**no new-model**" the intent will raise an alert. Instead of repeating the string "**no new-model**" in many places of intent definition (if condition and status codes), you can create a compound variable so that it can be referenced through the intent:

- 1. Click **All the Variables** and select the device.
- 2. Click +Add Compound Variable to open edit Compound Variable.
- 3. In the Edit Compound Variable popup, define the Variable Name and its Definition.

Intent Variables for Seed Logi	c .		
Intent Variable	Use Automation Data	Table Task Variable	
+ Add Compound Variable	2 + Add Compound Ta	ble 🗸	}
Intent	Edit Compound Var	iable	×
🖌 🥏 ACI-QA-SW1	Variable Name:	disable_aaa	
▷ → AAA_Servers ▷ ∞ Variables1	Type:	string	~
ACI-QA-SW1	Definition:	no new-model + 교 Variable + 다 Function	
h			Cancel OK

## 6.1.3.3 Define Diagnosis

\_

Define Diagnosis to set the **If** condition for the Variables:

Name:	AAA_Config 2				Anchor: \$AAA_New_Mode			el ~	
	Type description of								
] Loop	Table Rows								
TC	2								
If	3								
	3 a ACI-QA-S	Current 🗸					Current 🗸	,	
			Equals	~	disab	le_aaa	Current 🗸	Î	
A	a ACI-QA-S		Equals	~	disab	le_aaa			
A	ACI-QA-S AAA_New_Model	~	Equals	~	disab	le_aaa			

Enter a message under the **Then** and **Else** output areas to appear as the result of the diagnosis:

📄 Diagnosis N	lessage	:	🗌 Save to Incident ا
~	\$this	device AAA is disabled	
🗸 🛐 Set Stat	us Coc	e for Device:	
Error	~	<pre>\$this_device AAA is disabled</pre>	
✓-IS Set Stat	us Coo	e for Intent:	
Error	~	<pre>\$this_device AAA is disabled</pre>	
➡ Export to C CSV Name: A4 dd Logic ✔		ort: Define 2	=
CSV Name: 🗚			≡
CSV Name: 🗚			≡ ⊡ Dek
CSV Name: A4 dd Logic ✓	A conf	ig check	
CSV Name: A4 dd Logic ↓ Else 3	A conf	ig check	
CSV Name: A4 dd Logic ↓ Else 3	A conf lessage \$this	ig check :: :device AAA is enabled	_
CSV Name: A4 dd Logic ↓ Else 3 Diagnosis №	A conf lessage \$this us Coc	ig check :: :device AAA is enabled	_
CSV Name: A4 dd Logic ~ Else 3 Diagnosis N	A conf lessage \$this us Coc ss ~	g check :: :: :: :: :: :: :: :: :: :	■ Dete □ Save to Incident =

## 6.1.3.4 Replicate and Run Intent

Now, use the Intent Replication Wizard to add the Automation Column (**AAA Config Check**) to the existing ADT table, **NIST Compliance**.

👪 NIS	T Compliance Table Bu	uilder Last Updated at: 07/02/2024 06:51 PM 🏻 🍕 F	Rebuild Table		Add Data Mar	iually $ \!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!$	đ
Descri	ption: Type description here						
items:	62 Rows 4 Columns			Search.	Q 🛛 Advanced Filter	: Undefined	0
No.	Device	S Device Type	Telnet vty check		AAA Config Check		=
1	Berlin-R1	Cisco Router	Check Vty Telnet Access Berlin-R1	0	AAA Config Check Berlin-R1	•	1
2	Berlin-vEdge	Cisco IOS Switch	Check Vty Telnet Access Berlin-vEdge	0	AAA Config Check Berlin-vEdge	۲	
3	DE-MUC-CR01-01	Cisco Router	Check Vty Telnet Access DE-MUC-CR01-01	•	AAA Config Check DE-MUC-CR01-01	۲	
4	DE-MUC-CR01-02	Cisco Router	Check Vty Telnet Access DE-MUC-CR01-02	•	AAA Config Check DE-MUC-CR01-02	۲	
5	DE-MUC-CW01-01	Cisco IOS Switch	Check Vty Teinet Access DE-MUC-CW01-01	0	AAA Config Check DE-MUC-CW01-01	۲	
6	DE-MUC-CW02-01	Cisco IOS Switch	Check Vty Teinet Access DE-MUC-CW02-01	0	AAA Config Check DE-MUC-CW02-01	۲	
7	DE-MUC-CW03-01	Cisco IOS Switch	Check Vty Telnet Access DE-MUC-CW03-01	0	AAA Config Check DE-MUC-CW03-01	۲	
8	ISP-P02	Cisco Router	Check Vty Telnet Access ISP-P02	0	AAA Config Check ISP-P02	۲	
9	ISP-PE01	Cisco Router	Check Vty Telnet Access ISP-PE01	0	AAA Config Check ISP-PE01	۲	
10	ISP-PE02	Cisco Router	Check Vty Telnet Access ISP-PE02	•	AAA Config Check ISP-PE02	۲	
11	ISP-PE03	Cisco Router	Check Vty Telnet Access ISP-PE03	•	AAA Config Check ISP-PE03	۲	
12	ITE_EXTEND	Cisco IOS Switch	Check Vty Telnet Access ITE_EXTEND	0	AAA Config Check ITE_EXTEND	۲	
13	JP-TYO-CR01-01	Cisco Router	Check Vty Telnet Access JP-TYO-CR01-01	0	AAA Config Check JP-TYO-CR01-01	•	
14	JP-TYO-CR01-02	Cisco Router	Check Vty Telnet Access JP-TYO-CR01-02	0	AAA Config Check JP-TYO-CR01-02	۲	
15	JP-TYO-CW01-01	Cisco IOS Switch	Check Vty Telnet Access JP-TYO-CW01-01	0	AAA Config Check JP-TYO-CW01-01	۲	

Run the intent columns once and rebuild the table.

Telnet vty check	AAA Con	tails
Check Vty Telnet Access Berlin-R1 💿	AAA Config Check Berlin-R1	•
Check Vty Telnet Access Berlin-v 💿	AAA Config Check Berlin-vEdge	
Check Vty Telnet Access DE-MUC •	AAA Config Check DE-MUC-CR01	•
Run Intents Once - Telnet vty check		×
Data Source: Live Data Only Run Intents if Intent Device	✓ belongs to Select ✓	
Notification		X
The intents in the current ADT column are	e executed once now.	Cancel OK 2
View Details	OK	3

) 👪 NIS	T Compl	iance	Table Builder	Last Updated at: 07/03/2024 10:4	44 PM 🍳 Reb	uild Table	
Descri	ption: 7	ype description here			and the second sec		
ltems:	62 Row:	s 7 Columns			1		
No.	😂 Devi	ce	S Device Type	I Telnet vty check		nfig Check	
1	Berlin-F	۲1	Cisco Router	Check Vty Telnet Access Berlin-R1	AAA Confi	g Check Berlin-R1	0
2	Berl R	tebuild Table			×	Check Berlin-vEdge	0
3	DE-I					Check DE-MUC-CR01	0
4	DE-I	Build the column gro	ups: All 5		~	Check DE-MUC-CR01	0
5	DE-I		Log: 🔘 Production Mode	Only show major execution process lo	g	Check DE-MUC-CW0	0
6	DE-I		O Debug Mode	Show all the detailed log	~	Check DE-MUC-CW0	•
				Cancel	Build 6		

# 6.1.4 Create NI: Device Unused Ports Config Check

In this section, you will repeat the same flow to create an Intent **Device Unused Ports Config Check** to check the unused ports.

### 6.1.4.1 Define Variables with Visual Parser

Add a CLI Diagnosis and enter the command **show run | sec interface** to retrieve sample data. Use the **Paragraph** Parser pattern.

Line	Pattern
ID Line A	^interface \$intf
Var Line 1	LinesByVariable[\$intf_config]:\$intf-

1. Define Variable     2. Define Diagnosis       Format1         Format1	LI Command Diagnosis	
Formatl       ·       Test on Devices 10         © Double-click a variable to parse. Select multiple lines to parse a table.       Critical Variable (0)       • Int_config       Type: Paragraph (3)       • New Pattern ×       =         Current Device       > 06/27/2024 12:14:28 PM       Search       • Int_config       Citical Variable (0)       • Int_config       Type: Paragraph (3)       • New Pattern ×       =         1       10       Interface       00/27/2024 12:14:28 PM       Search       • Int_config       Citical Variable (0)       Interface       Sintf       • Interface       Interface       Sintf       • Interface       Interface       Sintf       • Interface       Interface       Sintf       • Interface       • Interface	OS-BOS-R1     show run   sec interface     X <     Retrieve     2 with Live Data	
O Double-click a variable to parse. Select multiple lines to parse a table. Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 60/277/2024 12:14:28 PM Search Critical Variable (0) Current Device   O 10 Line A  O	1. Define Variable 2. Define Diagnosis	
Current Device       06/27/2024 12:14:28 PM       Search         1       US-BOS-REshow run   sec interface         amatch interface input       collect interface output         3       collect interface loopbacki         9       description         1       is opficated toopbacki         9       description         1       is opficated         9       description         1       is opficated         9       description         10       interface Loopbacki         9       description         11       is opficated         12       anterface Loopbacki         9       description         11       is opficated         12       anterface Loopbacki         9       description         13       no ip address         14       interface Loopbacki         15       anterface Loopbacki         16       description         17       ip opficates         18       ip odfress         19       anterface Loopbacki         10       ip odfress         11       ip odfress         11       ip odfress	Format1 V +	5: 0
Current Device       04/27/2024 12:14:28 PM       Search       Q       I         1       10-005-Fit3show run   sec interface       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I       I <td< th=""><th></th><th>Ξ</th></td<>		Ξ
<pre>2 match interface input 3 collect interface output 4 interface Loopback3 5 descript 1 This is my Boston Sites 6 ip address 10.10.10.255.255.255.255.255 7 ip ospf 1 area 0 4 interface Loopback1 9 description This is my Boston Site 10 ip address 8.8.8.25.255.255.255.255 11 jo ospf 1 area 0 12 4interface Loopback9 13 no ip redirect3 14 shutdown 15 4interface Loopback9 14 description TEST-Kunal - Remove Description later 15 bandwidth 10006 16 ip address 10.99.1.1 255.255.255.0 19 ip redirect3 19 ip redirect3 19 ip redirect4 19 ip redirect4 19 ip redirect4 19 ip redirect4 19 ip redirect5 19 ip off area 0 19 ip redirect4 19 ip redirect4 19 ip redirect5 19 ip off area 0 19 ip redirect4 19 ip off area 0 19 ip redirect5 19 virtual-reasembly in 10 interface Loopback0 description TEST-Kunal - R 10 nunel0 interface Loopback99 on ip address shuddown 10 nunel0 interface Loopback99 on ip address shuddown 10 nunel0 interface Loopback99 on ip address shuddown 10 nunel0 interface Loopback99 on interface Loopback99 on ip address shuddown 10 nunel0 interface Tunnel0 description TEST-Kunal - R 10 nunel10 interface Tunnel0 description to 192 MG</pre>	Current Device V 06/27/2024 12:14:28 PM Search Q	1
	1       JS-BOS-R18show run   sec interface         2       match interface input         3       collect interface output         4       interface loopback0         5       description This is wy Boston Sites         1       jp ospf 1 area 0         4       interface loopback1         9       description This is wy Boston Site         10       jp address 10.10.10.255.255.255.255         7       ip ospf 1 area 0         4       interface loopback1         9       description This is wy Boston Site         11       jp ospf 1 area 0         2       interface Loopback2         9       address 8.0.8.0.255.255.255.255.255.255.255.255.255.2	· · · · · · · · · · · · · · · · · · ·

### 6.1.4.2 Define Intent Variable for Seed Logic

Define an intent variable, **\$Alert**, to indicate whether to raise an alert and set the default value to *YES*.

Intent Variables for Seed Lo	gic				
Intent Variable	Use Automation	Data Table	Task Variable	5	
+ Add Intent Variable 2	+ Add Intent Table			مار الاست	
D 🚺 Intent 🚺	Edit Intent Variable				×
🖌 😋 US-BOS-R1	Variable Name:	Alert 3			
▷ 📜 Int_config					
US-BOS-R1	Type:	string			~
	Initial Value:	YES 4			
				Cancel	OK 5

### 6.1.4.3 Define Diagnosis

Define a diagnosis, *Check\_inft\_conf,* to determine if the unused ports are not shutdown. If so, create an alert for this interface and set the intent variable to *YES*.

🗎 Add N	ote D Add Diagnosis	Can also click a variable on the left to add autom
Name:	check_intf_conf	Anchor: Int_config.\$intf_config~
	Type description of the diagnosis	

- 1. Define the **If** condition for Variable:
  - a) Variable *intf\_config* does not contain *shutdown*.
  - b) Variable *intf\_config* does not contain *switchport*.
  - c) Variable *intf\_config* does not contain *address*.
  - d) Variable *intf\_config* contains no ip address.
  - e) Variable *intf\_config* contains no switchport.
  - f) Variable *intf\_config* does not contain *spanning-tree*.
  - g) Variable *intf\_config* does not contain *vrf forwarding*.
  - h) Boolean Expression: A and B and F and G and (C or D or E).

If							
A	😁 US-BOS-R1	Current 🗸	U				
	intf_config	~	Does not contain	~	shutdown	~	Ē
в	😁 US-BOS-R1	Current 🗸					
	intf_config	~	Does not contain	~	switchport	~	Ŵ
с	😋 US-BOS-R1	Current 🗸					
	intf_config	~	Does not contain	~	address	~	Ē
D	😋 US-BOS-R1	Current 🗸					
	intf_config	~	Contains	~	no ip address	~	Ē
E	😁 US-BOS-R1	Current 🗸					
	intf_config	~	Contains	~	no switchport	~	Ē
F	😋 US-BOS-R1	Current 🗸					
	intf_config	~	Does not contain	~	spanning-tree	~	<b>m</b>
G	😁 US-BOS-R1	Current 🗸					
	intf_config	~	Does not contain	~	vrf forwarding	~	Ē
н	Select Variable	~					
Bool	lean Expression:	A and B and F and	G and (C or D or E)				

2. Define Intent Output message. To Set the Intent Variable, follow the steps outlined in section 6.1.2.3.

/ Then	2	
📄 Diagnosis Messag	e:	🗌 Save to Incident 🔳
↓ \$this	_device \$intf unused but no shutdown	
Set Status Co	de for Device:	
🌖 Error 🗸 🗸	<pre>\$this_device \$intf unused but no shutdown</pre>	
Set Status Co	de for Intent:	
❶ Error ∨	<pre>\$this_device \$intf unused but no shutdown</pre>	
Set Intent or Task	Variable:	=
	✓ = "YES"	

Now, add another diagnosis to check whether this device has unused showdown interfaces and create the status code on the device level.

2. Defir	e Diagnosis									
Add No	ote D Add Diagnos	sis 1				Ca	an also click a varia	ble on the left to	o add ar	utoma
Name:	Check Device Alert	2					Anchor:			~
	Type description of the did	agnosis								
🗌 Loop 🗸 If	Table Rows									
A		Current								
	Alert	~	Equals	3		~	YES		~	Ē
В	Select Variable	~								
	°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~ <u>~</u>	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		

📄 Diagnosis Messag	ze:	🗌 Save to Incident =
↓ \$thi	s_device has unused port but no shutdown	
Set Status Co	de for Device:	
🌖 Error 🗸	\$this_device has unused port but no shutdown	
Set Status Co	de for Intent:	
🌖 Error 🗸 🗸	\$this_device has unused port but no shutdown	
P Else 5	ze:	☐ Delete
	ze:	
✓ \$thi	s_device doesn't have unused port but no shutdown	
🔽 🔄 Set Status Co	de for Device:	
🤣 Success 🗸	<pre>\$this_device doesn't have unused port but no shutdown</pre>	
✓-S Set Status Co	de for Intent:	
🕗 Success 🗸	<pre>\$this_device doesn't have unused port but no shutdown</pre>	
dd Logic 🗸		

# 6.1.4.4 Replicate and Run Intent

Replicate this intent and add the automation column (**Device Unused Ports**) to the ADT table, **NIST Compliance**. Then, run the replicated intents once and rebuild the table.

👪 NIS	ST Compliance	Table Builder	Last Updated at: 07/02/2024 08:28 PM	🌯 Rebuild Table	Add Data Manually 🗸 📃 🔐
Descri	ption: Type description here				
Items:	62 Rows 5 Columns			Search	Q T Advanced Filter: Undefined
No.	Device	S Device Type	Telnet vty check	AAA Config Check	Device Unused Ports Config C
1	Berlin-R1	Cisco Router	Check Vty Telnet Access Berlin-R1 💿	AAA Config Check Berlin-R1	Device unused Ports config 💿
2	Berlin-vEdge	Cisco IOS Switch	Check Vty Telnet Access Berlin-v 💿	AAA Config Check Berlin-vEdge 🛛 💿	Device unused Ports config 💿
3	DE-MUC-CR01-01	Cisco Router	Check Vty Telnet Access DE-MUC 💿	AAA Config Check DE-MUC-CR01 💿	Device unused Ports config 💿
4	DE-MUC-CR01-02	Cisco Router	Check Vty Telnet Access DE-MUC 💿	AAA Config Check DE-MUC-CR01 💿	Device unused Ports config 👁
5	DE-MUC-CW01-01	Cisco IOS Switch	Check Vty Telnet Access DE-MUC 💿	AAA Config Check DE-MUC-CW0 💿	Device unused Ports config 💿
6	DE-MUC-CW02-01	Cisco IOS Switch	Check Vty Telnet Access DE-MUC 💿	AAA Config Check DE-MUC-CW0 💿	Device unused Ports config 💿
7	DE-MUC-CW03-01	Cisco IOS Switch	Check Vty Telnet Access DE-MUC 💿	AAA Config Check DE-MUC-CW0 💿	Device unused Ports config 💿
8	ISP-P02	Cisco Router	Check Vty Telnet Access ISP-P02	AAA Config Check ISP-P02	Device unused Ports config 💿
9	ISP-PE01	Cisco Router	Check Vty Telnet Access ISP-PE01	AAA Config Check ISP-PE01	Device unused Ports config 💿
10	ISP-PE02	Cisco Router	Check Vty Telnet Access ISP-PE02	AAA Config Check ISP-PE02	Device unused Ports config 💿
11	ISP-PE03	_ Cisco Router	Check Vty Telnet Access ISP-PE03 🛛 👁	AAA Config Check ISP-PE03	Device unused Ports config 💿

## 6.1.5 Create NI: Device Password Policy Config Check

In this section, you will practice creating an intent to check the **Device Password Policy Config.** You will use the command **show run | section username** to retrieve the sample data and define a **Paragraph** Parser to find the username and password level.

Line	Pattern
ID Line A	^username
Var Line 1	^username \$string:username
Var Line 2	password \$int:pwd_level

#### In the diagnosis, you will check whether the *password level* is less than 7.

2. Defir	e Diagnosis						
📄 Add No	ote 🛛 🖸 Add Diag	nosis 1		Can a	also click a varia	able on the left t	o add automatio
Name:	pwd_level_check	2			Anchor:	pwd_conf.\$us	sername 🗸 3
	Type description of the	e diagnosis					
	Table Rows , 📃 pwo	d_conf ✓ Ta	able Key: userna	ame 🗸 🥑			Ø
∨ If	CA-TOR-SW2	Current 🗸	<b>5</b>				
	pwd_level	~	Less than	~	7		~ 🛍
в	Select Variable	~					

Define the Diagnosis message:

Diagnosis Message:	Save to Incident
\$this_device username \$username password level is less than 7	
Set Status Code for Device:	
Set Intent or Task Variable:	
Alert v = "YES" 2	
✓ Else	🗑 D
Add Logic V V Else Diagnosis Message:	🗑 D
v Else	

## 6.1.6 Create NI: Device Line Session Timeout

In this section, you will practice creating an intent to check the **Line Session Timeout**.

Use the *Config Diagnosis* and define the **Paragraph** Parser pattern as follows:

Line	Pattern
Start Line	^line
End Line	!
ID Line A	^line \$linetype \$int:port
Var Line 1	LinesByVariable[\$configlet]:\$linetype

Configuration Diagnosis			×
CA-TOR-SW2 Retrieve 1 with Live Data			
1. Define Variable		2. Define Diagnosis	
Format1 V +			Test on Devices: 0
<ol> <li>Double-click a variable to parse. Select multiple lines to parse a table.</li> </ol>	Critical Variable (0)	👍 line_config 🗸 🗸 Type: Paragraph 🔁 + New Pattern 💊	·
Current Device 06/27/2024 12:35:13 PM Search	۹ • •	Start Line: Aline	✓ ⊚ 💼
<pre>171 no ip http secure-server 172 ! 173 ! 174 ip sla 1 175 icmp-echo 228.1.1.1 source-interface Vlan300 176 request data-size 800 177 threshold 55 178 frequency 20 179 ip sla schedule 1 life forever start-time now 180 ! 181 ! 182 Isnmp-server community ******** 183 somp-server location Toronto 184 ! 185 control-plane 186 !</pre>		End Line: 1 4 ID Line A ^line \$linetype \$int:port 5 188 Ine con 8 Var Line 1 LinesByVariable[\$configlet]:\$linetype 6 188 Ine con 8	<ul> <li>©</li> <li>T</li> <li>T</li></ul>
187 ! 188 line con 8 189 logging 5 pronous	Start Line P1-ID Line A		
<pre>189 logging 5 pronous 190 line aux 0 191 line ty 0 4 192 login local 193 transport input all 194 [ 195 ! 196 end 197 198</pre>	Var Line A Var Line 1 P2-ID Line A Var Line 1 P3-ID Line A Var Line 1 End Line 1	Output     + Parse Lines       Slinetype     >       Slinetype     >       Source     0       aux     0       vty     0       line vty 0	-
Help   All Intent Variables			Canc 7 Apply

In the diagnosis, you check whether the device configuration contains *exec-timeout*.

Add N	ote D Add	Diagnosis	2 Can als	so clic	k a varial	ble on the left	to add a	uton
Name:	Check_line_Conf	ig 3		A	nchor:	line_config.	\$line	4
	Type description	of the diagnos	is					
🗹 Loop	Table Rows , 📃	line_config	∽ Table Key:  i	netyp	oe 🗸	5		(
v If								
× II	6							
	CA-TOR-S	Current 🗸						
		Current 🗸	Does not contain	~	no exe	20	~	Î
A a	a ca-tor-s		Does not contain	~	no exe	20	~	Î
A a	CA-TOR-S	~	Does not contain Does not contain	~		ec imeout	~	
A a B	CA-TOR-S configlet	✓ Current ✓					~	_

### Define the Diagnosis message:

📄 Diagnosis Me	issage:	Save to Incident 📃
~	<pre>\$this_device \$linetype \$port does not config exec-timeout</pre>	
🗸 💿 Set Statu	s Code for Device:	
Error	\$this_device \$linetype \$port does not config exec-timeout	
🖌 💽 Set Statu	s Code for Intent:	
Error	\$this_device \$linetype \$port does not config exec-timeout	
dd Logic∨ • Else 3		面 Dele
📄 Diagnosis Me	issage:	Save to Incident
~	<pre>\$this_device \$linetype \$port exec-timeout has been configured</pre>	
🛛 🖪 Set Statu	s Code for Device:	
Success	\$ > \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$	
🔽 🛐 Set Statu	s Code for Intent:	
Success	\$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$	
dd Logic∨		
Add Elself		

## 6.1.7 Create Intent and Summary Dashboard

You can view the Intent results through two different dashboards:

- Intent Dashboard to view the individual Intent results and
- **Summary Dashboard** to view the consolidated Intent results in a single view.

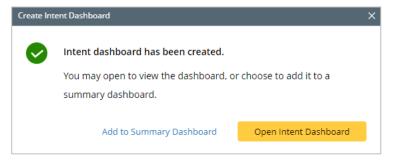
#### 6.1.7.1 Intent Dashboard

The Intent Dashboard monitors specific network issues, providing detailed information and displaying the results. You can save frequently used dashboards as templates.

To create an Intent Dashboard directly from the ADT automation column menu, follow these steps:

		{ 5 [	Device Type	~~ <u>~</u>	Telnet vty Run Details
Create Intent Dashbo	pard 2	, 		×	Run Intents Once
Create Inten Name: Location: Use Templa Data Source			~		Open Seed Intent Rebuild Intent-related Column Group Remove Empty Wrapper Intent Enable Auto Intent Export Diagnosis Result to CSV View Summary Report Export Intent Output Map Debug Empty Cells
	ata Table: NIST Compliance gered Follow-up Intent Results				Tag Current Column
Intent Column:	b Select All ✓ Telnet vty check ✓ AAA Config Check			~~~	Set as Table Key Submit Related Commands to Benchmark New Intent Dashboard 1
Time Range	<ul> <li>Device Unused Por</li> <li>Device Password P</li> <li>Device Line Sessio</li> </ul>	Cancel	✓ Create e		

- 1. Hover over the ADT automation column, go to the menu and then click **New Intent Dashboard.**
- 2. In the Create Intent Dashboard window, define the following:
  - a) Enter the Dashboard Name, NIST Compliance Monitoring.
  - b) Select the **Location** to save the Intent Dashboard.
  - c) Data Source: By default, Automation Data Table is selected.
  - d) **Intent Column**: Click the **Intent Column** dropdown and check the **Select All** checkbox to select all Intents, allowing the Intent dashboard to be created for all Intents at once
  - e) Click Create.
- 3. In the Create Intent Dashboard dialog popup, click Open Intent Dashboard.



4. The Intent Dashboard will be like:

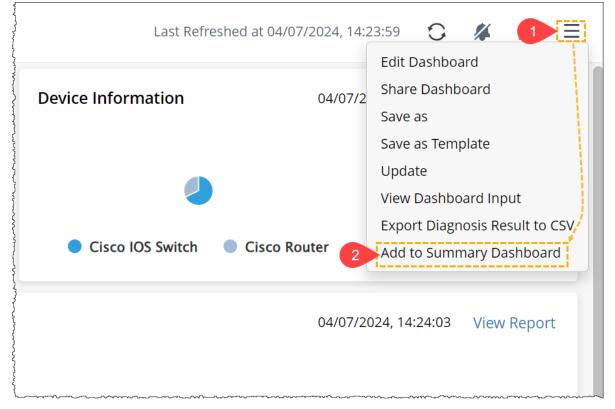
NIST Compliance Moni	itoring					Last Refreshed at 04/07/202	24, 14:23:59	0	/	∅ ≡
Summary		04/07/2024, 14:24:04	View Report	Device Inf	ormation	04	4/07/2024, 14:24	1:03	View R	eport
	280	Intents						_		
	337	Times Executed					56		)evice	es
	396	Intent-level Alerts			Cisco IOS Switch	isco Router				
Intent Result History						04	4/07/2024, 14:24	4:03	View R	eport
Time Range: All ∽ Result: A	di 🗸									
16		- Sum of Intent Alert Stat	us Code Count	→ Sum of Inte	nt Success Status Code Count					
0		A	$\sim$		L.			~		
03/01/01/01/01/01/01/01/01/01/01/01/01/01/	01 01 01 01 01 01 01 01 01		01 01 01 01 01 01 04 04 04 04 04	07 07 07 07 07 07 04 04 04 04 04	04 04 04 04 04 04 04 04 04 04 04	01 01 04 04 04 04 04 04 04 04 04 04	01 01 01 01 01 01	01.01	A107.07	04/07.
			Top Five In	tent Alerts						
Intent Name	Мар	Execution Time	Intent Alert Status	Code Count	Intent Success Status Code Co	Intent Status Code Summary	Intent Alert D	etectio	n	
Device Line Session Timeout	View Map	04/07/2024, 12:40:18	4		0	US-NYJ-CR01-01 con 0 does not	. 1			
Device Line Session Timeout		04/07/2024, 12:40:18	4		0	DE-MUC-CR01-01 con 0 does n				
Device Line Session Timeout S	View Map	04/07/2024, 12:40:16	4		0	SG-SIN-CR01-01 con 0 does not	. 1			

### 6.1.7.2 Summary Dashboard

The summary dashboard provides an overview displaying results from multiple Intent dashboards of the entire network or a set of network devices. With Summary Dashboard, you can group Intent Dashboards into widgets based on diagnosis purpose and display results by device, site or device groups. You can use the summary dashboard to monitor critical information across thousands of devices and discover the root cause for issues in one view.

Follow the step-by-step instructions to create a Summary Dashboard:

- 1. Click  $\equiv$  to open the menu located at the top-right corner of the dashboard window.
- 2. Select **Add to Summary Dashboard** to open the corresponding window for creating a summary dashboard.



- 3. In the **Add to Summary Dashboard** window, let us create the new summary dashboard and group as follows:
  - a) **Summary Dashboard**: open the dropdown menu and select **+New Summary Dashboard** to pop up its dialogue.
  - b) Enter the dashboard basic details like **name**, **group title** and **location** of the summary dashboard to save.
  - c) Click **OK** to save and create the summary dashboard.

New Summary Dashboard	;	×	Add to Summary Dashboa	rd X
Shared Dashboard	ls		Summary Dashboard:	Select or create New
My Dashboards			·	Search
			Dashboard Group:	Shared Dashboards (38)
				Automation Library (7)
				Config Drift Check
				Config Drift Check (Monday 5 am weekly)
		+		🖺 Golden Template
				🗈 Network Failover
			$\sim$	Network Failover (Daily 6AM)
				+ New Summary Dashboard
Dashboard Name:	NIST Compliance Dashboard			
Dashboard Group Title:	NIST			
	Cancel OK	C		

4. In the Add to Summary Dashboard dialog popup, click Open Summary Dashboard.

Add to	Summary Dashboard	×
0	<b>Intent dashboard has been added to the summary dashboard.</b> Open the summary dashboard to view or edit your changes.	
	Close Open Summary Dashboard	

## The Summary Dashboard will be like:

Summary						
Number of Intents	Number of Devices	Number of A	lerts	Numb	er of Successes	
280	56	792	792		2	
NIST						
Number of Intents	Number of Devices	Number of Ale	erts	Number	of Successes	
280	56	792		112	2	
						Q
Dashka and Jasan Carrie	Intent Results	Device Results				< >
Dashboard and Intent Group	Intent Results	Total Device Results	US-BOS-CR01-01	Berlin-R1	JP-TYO-CR01-01	ITE_EXTEND
NIST Compliance Monitoring	56 396	6 56 396	1 8	1 8	1 8	1
Total Alert Count $\downarrow$			8	8	8	

You can access the created Summary Dashboard, as shown in the image.

	net3rain <sup>Next</sup>	-Gen Search Anything and	Create Map Q 📀	
$(\pm)$	Summmay Dashboa	ard > Shared Dashboards		🛿 Help 😋 🖡 🗙
	Dashboard	Template	Search Dashboard and Double-click	to Open Q
Recents	Summary I	ntent and Universal	Name	Modified Date
63		utomation Dashboard (1)	Automation Library	11/04/2024, 07:03:54
Network		nboards (39) 2 on Library (7)	📒 NetBrain Essential Automati	17/06/2024, 21:22:24
		on Library PKG11 (1)	Automation Library Ty	18/06/2024, 21:24:36
Files	📒 Automati	on Library Ty (1)	📒 Xinyu Zhang	25/06/2024, 22:09:57
Site	Minghui (	Qiu (1) Essential Automation Libra	📒 Yihong Liu	25/06/2024, 22:29:09
	Sophia W		📄 Yan Wu	25/06/2024, 23:21:00
Path	📒 Xinyu Zha	ang (2)	🖿 Minghui Qiu	25/06/2024, 23:31:36
	📄 Yan Wu (2		🖿 Zelin Deng	26/06/2024, 00:23:54
Dashboard	Yan Wu A	utomation Library (4)	늘 Yueqin (Amy) Li	26/06/2024, 00:56:28
I >	<ul> <li>Yueqin (A</li> </ul>		📒 Sophia Wang	26/06/2024, 02:27:30
Intents	📄 Zelin Den	g (3)	늘 Yan Wu Automation Library	27/06/2024, 00:10:17
<b>(</b>	📄 My Dashboa	ards (0)	Automation Library PKG11	27/06/2024, 04:25:19
Chatbot			🛃 Golden Template Check	02/05/2024, 14:21:12
Data >			💕 What's Changed	21/06/2024, 00:54:59
			💕 Summary	01/06/2024, 00:14:35
			Ketwork_Failover	30/04/2024, 18:42:11
			🖺 checkSSH	25/06/2024, 23:32:21
			🖺 Michelle	27/06/2024, 23:50:57
			🔂 SA Triage Dashboard	27/06/2024, 22:19:26
			🔀 NIST Compliance Dashboard	04/07/2024, 15:37:21

## 6.2 CVE Security Advisory

CVE (Common Vulnerabilities and Exposures) classifies the vulnerabilities according to the different threat levels. It is critical to understand the severe CVEs that may affect your network devices and create a plan to address them, usually upgrading the system to a certain version that has fixed the CVE.

Each major network vendor publishes their CVEs. For example, Cisco has security advisories at <u>https://sec.cloudapps.cisco.com/security/center/publicationListing.x</u>.

In this section, we will create intents for the vendor-specific CVEs:

- 1. Given a CVE (for example, the newfound one), find how many devices are affected.
- 2. Given a device, find how many CVEs affect it.

This section includes the following main steps:

- Create CVE table
- Build Base ADT
- <u>Create an Intent to Check if a CVE is Affecting any Device</u>
- <u>Create Intent to Check Device Against all CVEs</u>
- <u>Create Intent and Summary Dashboard</u>

#### The Final ADT will be:

<b>F</b>	Automation Data Table Mar	nager	Help
	Cisco CVE Security Advisory	Table Builder         Last U	pdated at: 07/09/2024 03:17 PM 🔍 Rebuild Table Add Data Manually 🗸 🚍 💼
De	scription: Type description here		
Iter	ms: 95 Rows 4 Columns		Search Q 🛛 Advanced Filter: Undefined
No	. 🔗 Device	S Device Type	CVE-2023-20198 Device Against all CVEs
1	Berlin-R1	Cisco Router	Check if CVE is Affecting any Device Be   Check Device Against all CVEs Berl
2	Berlin-vEdge	Cisco IOS Switch	Check if CVE is Affecting any Device Be • Check Device Against all CVEs Berl •
3	DE-MUC-CR01-01	Cisco Router	Check if CVE is Affecting any Device DE • Check Device Against all CVEs DE •
4	DE-MUC-CR01-02	Cisco Router	Check if CVE is Affecting any Device DE O Check Device Against all CVEs DE
5	DE-MUC-CW01-01	Cisco IOS Switch	Check if CVE is Affecting any Device DE • Check Device Against all CVEs DE •
6	DE-MUC-CW02-01	Cisco IOS Switch	Check if CVE is Affecting any Device DE O Check Device Against all CVEs DE
7	DE-MUC-CW03-01	Cisco IOS Switch	Check if CVE is Affecting any Device DE O Check Device Against all CVEs DE
8	ISP-P02	Cisco Router	Check if CVE is Affecting any Device ISP • Check Device Against all CVEs ISP •
9	ISP-PE01	Cisco Router	Check if CVE is Affecting any Device ISP • Check Device Against all CVEs ISP •
10	ISP-PE02	Cisco Router	Check if CVE is Affecting any Device ISP • Check Device Against all CVEs ISP •
	ISP-PE03	Cisco Router	Check if CVE is Affecting any Device ISP  Check Device Against all CVEs ISP

## 6.2.1 Prerequisites

Please ensure that the following prerequisites are met:

- 1. Create the Cisco Device Group according to your requirements. Refer to the section 5.
- 2. Set up a CVE table that includes at least the CVE\_ID and affected versions.

#### 6.2.1.1 Create CVE table

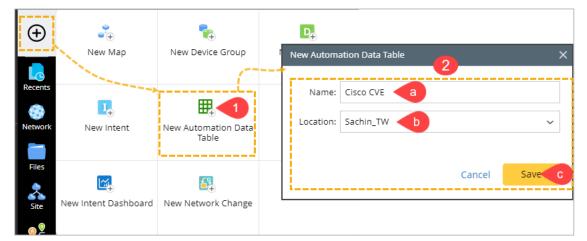
The CVE table is necessary for configuring the Intent Template Settings to define diagnoses for the available CVEs.

You can manually create an ADT table to add columns for CVE\_ID, CVE\_URL, and Affected\_Versions. Alternatively, you can use the CSV in the ADT table builder via the Imported CSV method.

Note: this exercise is for your practice only. NetBrain will maintain and update the CVE tables and you can download the table from your system.

Follow the step-by-step instructions to manually create the ADT table.

- 1. Click the plus icon 🕑 and click **New Automation Data Table**.
- 2. In the **New Automation Data Table** popup:
  - a) Enter the ADT Name, e.g., Cisco CVE.
  - b) Select the **Location** you wich to store the ADT.
  - c) Click **Save** and wait a moment for ADT to open.



- 3. In the **Automation Data Table Manager**, Click the **Add Data Manually** dropdown, and then click **Add Table Column**.
- 4. In the Add Table Column popup, define the following values and click **OK**.:
  - a) Enter Display Name, CVE\_ID

- b) Enter Column Name, CVE\_ID.
- c) Select Data Type as String
- 5. Click the **Add Data Manually** dropdown and click **Add Table Row**.
- In the Add Table Row popup, enter the Column Value as your CVE ID number and then click OK.

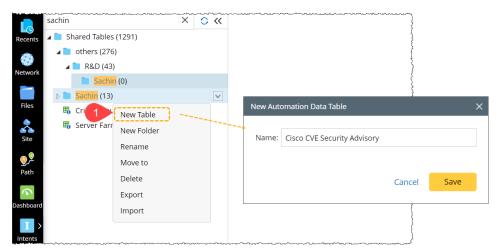
		Add Table Column X	
		Display Name: column1	
		Column Name: column1	
		Data Type: S String	С
	Table Builde	Cancel ОК e Add Data Manually 🗸 😑 е	ß
Add Table Row		Add Table Column	-
Column		Value	0
s column1		CVE-2023-20198 6	
		Cancel	

Similarly, create additional columns and rows for **CVE\_URL** and **Affected Version**. The **Cisco CVE** ADT will be:

irch	Q 0 4	< 🛒	Cisc	o CVE Ta	ole Builder	Last Updated at: 07/	💐 Rebuild Table	Add Data Manually $\sim~\equiv$	
	Build-Auto Template by Plugin666 (2) Cookbook (14)	D	escrip	otion: Type description here					
<b>E</b>	Empty Automation Library (10)	lt	ems:	4 Rows 3 Columns		Search	Q 🕎 A	dvanced Filter: Undefined	1
	Empty Staging Automation Library (24) NetBrain Essential Automation Library (18	3) N	10.	SCVE_ID	S CVE URL			s Affected Version	
F	Plugin Test (19)	1		CVE-2023-20048	https://sec.c	loudapps.cisco.com/securi	ity/center/co	9.8(3)26	
	Representative Methodology (3)	2		CVE-2023-20198	https://sec.c	loudapps.cisco.com/securi	ity/center/co	17.6.5	
	Sachin_TW (4)	3		CVE-2017-12240	https://sec.c	loudapps.cisco.com/securi	ity/center/co	15.1(04)M09	
Ħ	Cisco CVE	4		CVE-2018-0171	https://sec.c	loudapps.cisco.com/securi	ity/center/co	15.2(05)E	

## 6.2.2 Build Base ADT

1. Create a new ADT, Cisco CVE Security Advisory.



2. Build the base group with the method **Devices of Device Group**. Add the following columns: **Device** and **Device Type**.

mation Data Table Builder			
olumn Header:			Reset Al
Ĵ c1 🔐 Ĵ	c2 5		
Device	Device Type		
Base +	Column Group (B	ase):	Select Column 🗸
Description:			
The base ADT for NIST comp			
Select Method to Build Base T	U Dev		
Devices of Device Group	b v		
Devices of Device Group			
Select Devices by Device Grou	pt My Device Gro	Select Device Groups	×
Built-in Fields:		Search	
🕄 Device		Search	
Device Group	s	All Device Groups	
Device Group		▲ 🖃 🖿 My Device Groups	
Hostname	s d 🔪	BGP Devices (67)	
Mgmt IP	5	Cisco Devices (96)	
Mgmt Interface	S	Firewall (13)  Shared Device Groups	
		Ahmed	
Oevice Type	5	▷ □ Anurag	
Vendor	S	Assessment Reference Library	
Model	S	Automation Library	
Software Version	S	Automation Library Kunal	
	(Drag and drop o	Automation Library PKG	
Serial Number	S	A Character Hilton Tr	
		Cancel OK	

#### The Base ADT will be:

arch	Q	<mark>0</mark> «	E Cie	sco CVE Security Advisory	Table Builder	Last Updated at: 07/08/2024 11:27 AM	🔍 Rebuild Tabl
) 🛢 Automation Library Pr			•		Tuble Dullaci	2032 0000000 00. 0770072024 11.27 788	
: 🛢 Bulti-Auto Tempiate I			Descr	iption: Type description here			
<ul> <li>Build Auto Template 1</li> <li>Build Auto Template 1</li> </ul>			Items	: 95 Rows 2 Columns			
) 🖿 Cashbook (14)		_	No.	Device		s Device	еТуре
<ul> <li>Empty Automation (2)</li> <li>Empty Traging Autom</li> </ul>			1	Berlin-R1		Cisco Ro	uter
Matthew Issertial ha		-	2	Berlin-vEdge		Cisco IOS	Switch
Page Test (18)			3	DE-MUC-CR01-01		Cisco Ro	uter
Representative Metho	ang ta		4	DE-MUC-CR01-02		Cisco Ro	uter
a 📄 Sachin_TW (5)			5	DE-MUC-CW01-01		Cisco IOS	Switch
GA test     Gisco CVE			6	DE-MUC-CW02-01		Cisco IOS	Switch
Cisco CVE Security	Advisory		7	DE-MUC-CW03-01		Cisco IOS	Switch
🚯 NIST Compliance			8	ISP-P02		Cisco Ro	uter
🌇 Test Data			9	ISP-PE01		Cisco Ro	uter
<ul> <li>Staging Automation ()</li> <li>Next co.</li> </ul>	heary 1621		10	ISP-PE02		Cisco Ro	uter
1 Tan Wu Automation 1	heavy (47)		11	ISP-PE03		Cisco Ro	uter
1 🖥 Yan Hu Daging Autor	atter they	y CR	12	ITE_EXTEND		Cisco IOS	Switch
2 - Automation Library		of Texa	13	JP-TYO-CR01-01		Cisco Ro	uter
<ul> <li>BOP Config Change D</li> <li>Chack COPP Singleton</li> </ul>	-		14	JP-TYO-CR01-02		Cisco Ro	uter
Coller, corfs, check	(and the second		15	JP-TYO-CW01-01		Cisco IOS	Switch
R. Prog Target			16	JP-TYO-CW01-02		Cisco IOS	Switch

In the next sections, you will create two separate Intents to **Check if CVE is Affecting a Device** and another to **Check Device Against all CVEs**.

## 6.2.3 Create an Intent to Check if a CVE is Affecting any Device

From Intent Manager, create a new intent, *Check if CVE is Affecting a Device*.

	· · · · · · · · · · · · · · · · · · ·	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	
<b></b>		Network Intent Manager > Common Intent	❸Help & Ŧ X
Dashboard		Common Intent	~ }
	ę	sachin	× + New Intent
Intent Manager	Intent Re Wiz	All (cmd (2) (L1 - Config Change (2) (L1	- Environment C >>>
Chatbot		🔺 🚞 All Network Intents (1305)	<u> </u>
Chatbot		Sachin (0)	2 🥑
	L	3 New N	letwork Intent
Data		New F	older
Automation Data Table (ADT)	Intent Automati	Impor	t j
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Export	t j
		Renan	ne Š
		Move	То 3
		Delete	:

Select a Cisco device as the seed device.

work Intent (Edit Mo	ode)							
heck if CVE is Affe	ecting any Device Diagnosis Tree	2			Ru	n witi	h Live Data	Save 🕐 Help
description here								Intent Map: Select
I Seed Logic	Seplication Logic A							
📸 + Device 🛛 a	Select Devices							
	Select Devices by: 💿 Device Type	O Device Group	🔘 Site				1 Devices Selected	
	All Device Types	✓ Search			Q		US-BOS-CR01-01	A
	Hostname	Mgmt IP	Vendor	Model				
	UK-LHR-CW03-02	10.20.0.11	Cisco	3560E	^			
	UK-LHR-CW04-01	10.20.0.12	Cisco	3560E				
	UK-LHR-CW04-02	10.20.0.13	Cisco	3560E				
	US-BOS-CF01-01/act	10.10.0.6	Cisco	ASAv		>		
						2		
	US-BOS-CF01-01/stby	10.10.0.7	Cisco	ASAv				
	US-BOS-CF01-01/stby	10.10.0.7	Cisco	ASAv CGS-MGS-AGS		»		

### 6.2.3.1 Define Variables with Visual Parser

Add a **CLI Diagnosis**. Enter the command, *show version,* and then click **Retrieve** to retrieve the data from the **Live Data**.

Define a single parser for the software version: **Version \$mstring:ios\_version**,

CLI Command Diagnosis	×
US-BOS-CR01-01 show version X V Retrieve 1 Liv	re Data
1. Define Variable	2. Define Diagnosis
Format1 V +	Test on Devices: 0
Double-click a variable to parse. Select multiple lines to parse a table.     Critical Varia	ble (0) 🖸 Pattern1 → 🖍 Type: Single 👔 + New Pattern → 🚍
Current Device         O7/08/2024 12:28:15 PM         Search         Q           1         2         vare (I86BI_LINUX-ADVENTERPRISEK9-M), Version 15.4(2)T4, 2         Var Line 1           3         :isco.com/techsupport         4         :o Systems, Inc.         2         vare(IB6BI_LINUX-ADVENTERPRISEK9-M), Version 15.4(2)T4, 2         Var Line 1           4         :o Systems, Inc.         5         / prod_rel_team         6         7            6         7          8          9         :ks, 3 days, 15 hours, 11 minutes         10         id at 0         11         I/unetlab/addons/iol/bin/L3_ADVENTERPRISEK9_M_15.4_2T.bin"         13           13         14	Var Line 1 Version \$mstring:ios_version, 3 Var Line 1 Version \$mstring:ios_version, Output + Parse Lines - \$ios_version (mstring) = 15.4(2)T4

### 6.2.3.2 Define Intent Variable for Seed Logic

In this section, you will use the ADT **Cisco CVE** you created as a prerequisite. The columns in this ADT will be used as the Variables to define the diagnosis.

- 1. Click **All Intent Variables** in the bottom right corner.
- 2. Go to the **Use Automation Data Table** tab, click **+ Automation Data Table**, and select the **Cisco CVE**.
- 3. From the Select Automation Data Table window, select the Cisco CVE ADT and then click OK.
- 4. Click Close to exit from the Intent Variable for Seed Logic window.

	Intent Variables for Seed Logic X
Select Automation Data Tables	Intent Variable Use Automation Data Table Task Variable
Select Automation Data Tables  Search  S	Automation Data Tables         2           Image: Cisco_CVE         CVE_UD(string)         Affected_Version(string)           CVE-2023-20048         https://sec.cloudapps.cisco.com/sec         9.8(3)26 7.3.1.1 7.3.1 7.3.0 7.2.3.1 7           CVE-2023-20198         https://sec.cloudapps.cisco.com/sec         17.6.5 17.3.3 16.9.6 16.9.4 15.4(3)52           CVE-2017-12240         https://sec.cloudapps.cisco.com/sec         15.1(04)M09 12.2SRC 12.2(44)SE 12           CVE-2018-0171         https://sec.cloudapps.cisco.com/sec         15.2(05)E           appears after adding ADT         Appears after adding ADT

### 6.2.3.3 Define Diagnosis

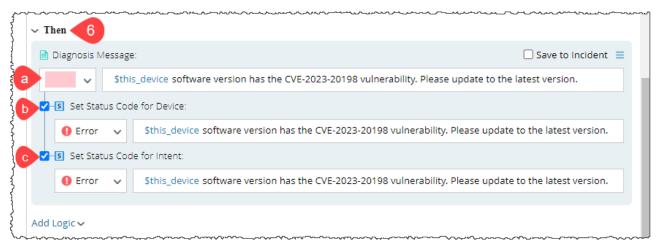
In this section, you will define diagnostics: to **Check if any CVE is affecting a device**. This diagnosis is usually useful if a new CVE is published and you want to check whether it affects your network device.

The diagnosis logic is simple: we will loop through the CVE table to find the matched CVE and check whether the affected software version includes the version of this device:

- 1. Go to the **Define Diagnosis** section.
- 2. Click **Add Diagnosis** to define conditions and Intent output message.
- 3. Enter the diagnosis name, e.g., **CVE-2023-20198** and select an Anchor **\$asa\_version** from the dropdown.
- 4. Tick the **Loop Table Rows** checkbox and select the Table Variable (*Cisco\_CVE*) and the Table Key (**CVE\_ID** and **Affected\_version**) from the dropdown.
- 5. Define **If** condition as detailed in the image:
  - a) Variable CVE\_ID Equals CVE-2023-20198.
  - b) Variable *Affected\_Version* Equals *asa\_version*.
  - c) The Boolean Expression will be **A and B**.

Add N	ote D Add	Diagnosis <	<b>2</b> c	an also clic	k a variable on the	left to add au	utom
Name:	CVE-2023-20198	3		A	nchor: \$ios_ver	sion	$\sim$
	Check this CVE is	affecting a d	evice or device	es in device	group.		
🖌 Loop 🗸 If	) Table Rows 🔛	Cisco_CVE 🔪	🗸 🛛 Table K	ey: CVE_ID	, Affected_Versio	n ~ <b>4</b>	¢ĝ
A	5	Current					
а	CVE_ID	~	Equals	~	CVE-2023-20198	~	Ī
в		Current			🥔 US-BOS-C	Current 🗸	
b	Affected_Version	~	Equals	~	asa_version	~	Ē
	Select Variable	~					
С							

6. Define Intent Output message for **If** condition.



- 7. Define **Elself** condition, i.e., Variable *CVE\_ID* Equals *CVE-2023-20198*.
- 8. Define Intent Output message for **Elself** condition.

ElseIf	í <b>(</b>						<u> Delete</u>
А			Curre	nt			
	CVE_I	D		Equals	~	CVE-2023-20198	~ 🗊
В	Select	Variable		·			
/ Then	8						
📄 Diaş	gnosis N	/lessage:					Save to Incident 🔳
	~	\$this_dev	ice software vers	ion has resolved	the CVE-2023-20	198 vulnerability	
<mark>&lt; -</mark> 5	Set Sta	tus Code fo	r Device:				
	Succe	ss ∨ \$t	his_device softwa	re version has re	solved the CVE-2	023-20198 vulnerability	
<mark>~</mark> _S	Set Sta	tus Code fo	r Intent:				
•	) Succe	ss 🗸 💲 \$t	his_device softwa	re version has re	solved the CVE-2	023-20198 vulnerability	
	~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~_~_~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~_~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	، ب <sup>7</sup> محرک محکم محکم محکم محکم محکم محکم محکم	

9. Click **Apply** to save all the diagnosis settings and run the Intent. Check whether it is working as per your configuration settings.

### 6.2.3.4 Use Intent Replication Wizard

In this section, you will use the **Intent Replication Wizard** to add the Automation Column in the ADT table. You will use the existing ADT that you have created, **Cisco CVE Security Advisory**.

- 1. In the **Network Intent** (View Mode) window, go to the  $\equiv$  menu and open the **Intent Replication Wizard**.
- 2. In the **Seed Intent** tab, check for your NI (*Cisco CVE Security Advisory*) and then click **Next** to go to the **Define ADT** tab.

Network Intent (View Mode) - All Network Intents/Sachin/CVEs/Check if CVE is Affecting any Device	x
Check if CVE is Affecting any Dev	Copen 📄 0 🔥 0 ∠Edit
Result: 07/09/2024 02:44 PM ✓       Image: Constraint of the security	View Abstract Run Settings Data Clean Settings
Intent Replication Wizard - Check if CVE is Affecting any Device	Edit Save as
Seed Intent 2 Define ADT Replication Settings	Delete Refresh Export Share to Incident Publish Intent Intent Replication Wizard
Seed Intent:       Check if CVE is Affecting any Device       Select         Intent Template for <ul> <li>Device-based Replication</li> <li>Path-based Replication</li> </ul>	Auto Intent Wizard

3. Define ADT:

**Use an Existing ADT** and select **Cisco CVE Security Advisory**. Enter a meaningful Intent group name here.

Intent Replication Wizard - Check if CVE is Aff	ecting any Device X	
Seed Intent 3 Def	ine ADT Replication Settings Replicate Intent	
Create a New ADT	a Use an Existing ADT	
Automation Data Table: Cis	sco CVE Security Advisory	
b Replicate Intent to: Ex	isting Column Group 🗸 CVEs 🗸	
Replicate on Device Column: De	Select Automation Data Table	×
	Search	Q
	Representative Methodology	
	Reverse Engineering MVS     Sachin_TW	
	ACL Configuration Drift	
	ACL Name List	
	🖽 CA test	
	👪 Cisco CVE	
	🖽 Cisco CVE Security Advisory	
	🔀 Golden Template for ACLs	
	🐻 Golden Template Table	

- 4. Replication Settings:
  - a) In the **Intent Qualification** section, click the **Select** link to add Device Group. Use the device group that you have used during ADT creation.
  - b) Add the device group and then click **OK**.
  - c) Click **Next** to go to the **Replicate Intent** tab.

Seed Intent	Defin	ne ADT	$\geq$	Replication Settings 4	Replicate	e Intent
					🙆 Full Settin	gs for Templa
			Ac	ld Device Group	×	
Intent Qualification: 🔘 via Dev	/ice Groups/Sites: Select ∽	🔿 via Dynamic Se	earch:			
	a Select Dev	vice Groups		🖌 🖃 🖿 All Device Groups		
Define Macro Variables and Rul	Select Site	25		🖌 🖃 🖿 My Device Groups	_	
Item: 1	New Devic	re Group		🗌 💊 BGP Devices (44)		
	incir beine			b 🗹 🗟 Cisco Devices (95)	_	
Seed Device	Seed Command			🗌 🚭 Firewall (15)		
() US-BOS-CR01-01	show version			🔺 🔲 📒 Shared Device Groups		
				🕨 🗌 🖿 Ahmed		
				🗌 🖿 Ajay		

- 5. Replicate Intent:
  - a) In the **ADT Columns** section, you can see the **Column Name** as **Replicated Intent**. You can change this name to *CVE-2023-20198*.
  - b) Click **Save and Replicate** to save all the settings. You can see the **Open Output ADT** option after the successful submission of the replication request.
  - c) Click **Open Output ADT** to check the replicated Intent column in the ADT Manager.

Seed Intent	Define ADT	Replication Settings	<b>5</b> Replicate Intent
DT Columns:			Additional Columns
Column Data	Column Name	Tag	
Replicated Intent	a cvE-2023-20198	0 tags	
			b Save and Replicate
		Replication Request submitted at: C	

6. Review the populated Intent column.

👪 Cis	sco CVE Security Advisory	Table Builder         Last Updated at: 07/09/2024 0.	3:17 PM 🔍 Rebuild Table	Add Data Manually $  imes                   $
Descr	ription: Type description here			
Items	s: 95 Rows 3 Columns		Search	Q Y Advanced Filter: Undefined
No.	😂 Device	S Device Type	<b>I</b> CVE-2023-	20198
1	Berlin-R1	Cisco Router	Check if CVE i	is Affecting any Device Berlin-R1 🛛 💿
2	Berlin-vEdge	Cisco IOS Switch	Check if CVE	is Affecting any Device Berlin-vE 👁
3	DE-MUC-CR01-01	Cisco Router	Check if CVE	is Affecting any Device DE-MUC 💿
4	DE-MUC-CR01-02	Cisco Router	Check if CVE i	is Affecting any Device DE-MUC 👁
5	DE-MUC-CW01-01	Cisco IOS Switch	Check if CVE i	is Affecting any Device DE-MUC 💿
6	DE-MUC-CW02-01	Cisco IOS Switch	Check if CVE i	s Affecting any Device DE-MUC 👁
7	DE-MUC-CW03-01	Cisco IOS Switch	Check if CVE i	s Affecting any Device DE-MUC 💿
8	ISP-P02	Cisco Router	Check if CVE i	s Affecting any Device ISP-P02 🛛 💿
9	ISP-PE01	Cisco Router	Check if CVE i	s Affecting any Device ISP-PE01 O

## 6.2.3.5 Run Intent Once and Rebuild Table

Run the replicated intents once and rebuild the table. Check the intent status code.

	Table Builder Last U	Ipdated at: 07/09/2024 03:00 P	M 🔍 Rebuild Tabl	e				Add Da	ta Manually 🗸 📃 🗗
						Search		Q 🛛 Advance	d Filter: Undefined
1		S Device Type				1 CVE-2023-2019	8		Run 🚺 ails 😑 🚍
		Cisco Router			_	Check if CVE is Aff	ecting any Device	Berlin-R1	•
	Run Intents Once - CVE-2	2023-20198				×	ecting any Device	Berlin-vEdge	•
YL/YL/Y	Data Source: Live Data	3					ecting any Device	DE-MUC-CR01-01	•
2							ecting any Device	DE-MUC-CR01-02	•
~	Only Run Intents if	Intent Device	✓ belongs to Sele	ct 🗸			ecting any Device	DE-MUC-CW01-01	•
	Notification			×					
	The intents in the cu	irrent ADT column are exe	ecuted once now.		Cancel	ок 2			
	View Details			ок 3					

₽	🖥 Au	to	mation Data Table N	lanager							
>>	👪 Ciso	co	CVE Security Advisory		Table Build	der	Last Updated at:	08/11/2024 0	9:36 PM	<b>&amp;</b> R	ebuild Table
	Descri	pti	on: Type description here								>
	ltems:	9	Rebuild Table							×	
	No.										CVE-2023-20198
	1	E	Build the column groups:	All						~	eck if CVE is Affecting
	2	E	Log:	Productio	n Mode	Only	show major executio	n process log			eck if CVE is Affecting
	~~~v~~	L.e		○ Debug Mo	ode	Show	w all the detailed log				~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
								Cancel	Build		

# 6.2.4 Create Intent to Check Device Against all CVEs

In this section, you will learn how to create an Intent to **Check Device Against all CVEs**. You can follow the same steps as 6.2.3 with the only following difference while defining diagnosis:

In the diagnosis, you will loop through all CVEs and check whether the affected software versions of this CVE contain the software version of this device:

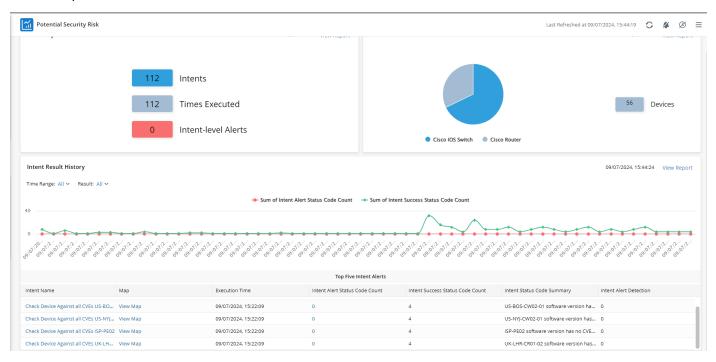
Add N	ote 🛛 🗗 Add Di	agnosis 🦯	Can a	also click a	a variable on the	e left to add a	utom
Name:	Check device for all	I CVEs 2		And	hor: \$ios_ve	r	~
	Type description of t	he diagnosis					
	Table Rows 🌐 C	isco_CVE 🗸	Table Key: ,	Affected_	Version 🗸	3	¢
✓ Loop ✓ If	Table Rows 🌐 C	isco_CVE ↓ Current	Table Key: ,		Version	i	
∽ If		Current	Table Key: , quals			i	

If so, raise an alert; if not, create a green info:

v Then 5		
📄 Diagnosis Messag	ge:	🗌 Save to Incident 🔳
↓ \$this	s_device software version has \$CVE_ID. Please update.	
Set Status Co	de for Device:	
❶ Error 🗸	\$this_device software version has \$CVE_ID. Please upda	te.
🗹 🖪 Set Status Co	de for Intent:	
🚺 Error 🗸	\$this_device software version has \$CVE_ID. Please upda	te.
Add Logic 🗸		
Else 6		🗑 Delete
Else 6	ze:	☐ Delete
Diagnosis Messag	ge: s_device software version has no \$CVE_ID.	
Diagnosis Messag	s_device software version has no \$CVE_ID.	
Diagnosis Messag	s_device software version has no \$CVE_ID.	
Diagnosis Messag	s_device software version has no \$CVE_ID. de for Device: \$this_device software version has no \$CVE_ID.	

## 6.2.5 Create Intent and Summary Dashboard

Now, you can create the Intent Dashboard and Summary Dashboard to view the results. Follow the same steps as 6.1.7 to create both dashboards and the same results as follows:



ummary										
lumber of Intents		r of Devices		Number of Alerts			Number of Successes			
112	56			0			560			
VE										
lumber of Intents	Numbe	r of Devices		Number of Alerts			Number of Successes			
112	56			0			560			
Dashboard and Intent Group	Intent Results	Device Results								<
		Total Device Results	US-NYC-SW1	SG-SIN-CW02-01	US-BOS-R2	UK-LHR-CW01-01	UK-LHR-CW03-02	UK-LHR-CW01-02	Berlin-R1	
Potential Security Risk	280	280 0	5	5	5	5	5	5	5	
Total Alert Count $\downarrow$			0	0	0	0	0	0		(

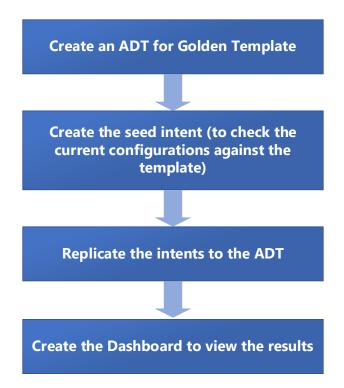
# 7 Network Assessment Case Study: Configuration Drift

Every network has certain configurations that should not be changed, sometimes called the **Golden Config** or **Golden Template**. The drift from these golden configs can lead to the network outage or performance downgrade. However, the prevention of configuration drift is a complex task since every network has its uniqueness.

In this chapter, you will apply what you have learned in previous chapters to create the automations to find out the configuration drift against the golden template across your whole network. These automations can be scheduled to run for continuous network assessment or run while troubleshooting or making the network change to ensure that the configuration drift does not cause the incident or the network change does not lead to breaking the golden config.

We will use two common examples: the standard ACL and NTP configurations. In the last section, we will demonstrate that the same principle can also applied to the public cloud configurations.

The flow to check the configuration drift is as follows:



You can also create a CSV report for the configuration drift in the seed intent. After replicating the intent to the ADT, you can create a summary CVS report from the ADT.

# 7.1 Check Configuration Drift Against the Golden Template

In this session, we will walk you through the flow to check the Golden Template using the standard **ACL** as an example.

#### The final ADT will be:

👪 А	utomation Data Tab	le Manager						🚱 Help
	CL Configuration Drift	Table Builde	Last Updated at: 07	/16/2024 06:17 PM 🛛 🍳 Rebuild Table				Add Data Manually 🗸 📃 🖬
Item	s: 95 Rows 8 Columns						Search Q	T Advanced Filter: Undefined
No.	Device	S Mgmt IP	SVendor	S Model	S Software Version	ACL Configuration Drift	SIntent Status Code	Last Execution Time
1	Berlin-R1	172.16.8.60	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 📀	Berlin-R1: acl_1 golden comp	07/16/2024 06:14:31
2	Berlin-vEdge	192.168.0.1	Cisco	WS-C4500X-32	03.04.04.SG	Check ACL against Golden Templ 📀	Berlin-vEdge: acl_1 golden cc	07/16/2024 06:14:27
з	DE-MUC-CR01-01	10.20.1.2	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 📀	DE-MUC-CR01-01: acl_1 gold	07/16/2024 06:14:27
4	DE-MUC-CR01-02	10.20.1.3	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 📀	DE-MUC-CR01-02: acl_1 gold	07/16/2024 06:14:31
5	DE-MUC-CW01-01	10.20.1.4	Cisco	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 📀	DE-MUC-CW01-01: acl_1 gold	07/16/2024 06:14:31
6	DE-MUC-CW02-01	10.20.1.5	Cisco	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 💿	DE-MUC-CW02-01: acl_1 gold	07/16/2024 06:14:27
7	DE-MUC-CW03-01	10.20.1.6	Cisco	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 💿	DE-MUC-CW03-01: acl_1 gold	07/16/2024 06:14:27
8	ISP-P02	4.0.0.2	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 💿	ISP-P02: acl_1 golden compli	07/16/2024 06:14:31
9	ISP-PE01	1.0.0.2	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 💿	ISP-PE01: acl_1 golden comp	07/16/2024 06:14:27
10	ISP-PE02	2.0.0.2	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 📀	ISP-PE02: acl_1 golden comp	07/16/2024 06:14:27
11	ISP-PE03	3.0.0.2	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 📀	ISP-PE03: acl_1 golden comp	07/16/2024 06:14:31
12	ITE_EXTEND	192.168.30.207	Cisco	WS-C3560X-48P	15.2(4)E7	Check ACL against Golden Templ 💿	ITE_EXTEND: acl_1 golden co	07/16/2024 06:14:31
13	JP-TYO-CR01-01	10.30.0.2	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 💿	JP-TYO-CR01-01: acl_1 golder	07/16/2024 06:14:31
14	JP-TYO-CR01-02	10.30.0.3	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 💿	JP-TYO-CR01-02: acl_1 golder	07/16/2024 06:14:26
15	JP-TYO-CW01-01	10.30.0.4	Cisco	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 📀	JP-TYO-CW01-01: acl_1 g	07/16/2024 06:14:26 PM
16	JP-TYO-CW01-02	10.30.0.5	Cisco	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 📀	JP-TYO-CW01-02: acl_1 g	07/16/2024 06:14:27 PM
17	JP-TYO-CW02-01	10.30.0.6	Cisco	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 📀	JP-TYO-CW02-01: acl_1 g	07/16/2024 06:14:27 PM
18	JP-TYO-CW03-01	10.30.0.7	Cisco	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 👁	JP-TYO-CW03-01: acl_1 g	07/16/2024 06:14:25 PM

The summary CSV report will be:

A	В	С	D	E	F	G	н	I. I.
1 Device	Golden Config	Current Config	Matched	<b>Missing Lines</b>	Extra Lines	Vendor	Model	Software Version
2 US-NYC-SW1			TRUE			Cisco	3560E	15.2(CML_NIGHTLY_20180510)FLO_DSGS7
3 UK-LHR-CW02-01		access-list 1	FALSE	access-list 1		Cisco	3560E	15.2(20170809:194209)
4 US-BOS-SW3			TRUE			Cisco	3560E	15.2(HI_20170202)FLO_DSGS7
5 US-NYJ-CR01-01		access-list 1	FALSE	access-list 1		Cisco	CGS-MGS-AGS	15.4(2)T4
6 SG-SIN-CW01-01		access-list 1	FALSE	access-list 1		Cisco	3560E	15.2(20170809:194209)
7 JP-TYO-CW01-02		access-list 1	FALSE	access-list 1		Cisco	3560E	15.2(20170809:194209)
8 SG-SIN-CW03-01		access-list 1	FALSE	access-list 1		Cisco	3560E	15.2(20170809:194209)
9 US-BOS-CW01-02	access-list 1	access-list 1	FALSE	access-list 1		Cisco	3560E	15.2(20170809:194209)
10 SG-SIN-CW02-01		access-list 1	FALSE	access-list 1		Cisco	3560E	15.2(20170809:194209)
11 US-BOS-CW02-02	access-list 1	access-list 1	FALSE	access-list 1		Cisco	3560E	15.2(20170809:194209)
12 JP-TYO-CW02-01		access-list 1	FALSE	access-list 1		Cisco	3560E	15.2(20170809:194209)
13 UK-LHR-CW01-01			TRUE			Cisco	Catalyst 38xx Stack	15.2(20170809:194209)
14 US-BOS-SW2			TRUE			Cisco	3560E	15.2(HI_20170202)FLO_DSGS7
15 US-NYJ-CW02-01		access-list 1	FALSE	access-list 1		Cisco	3560E	15.2(20170809:194209)
16 ISP-PE01			TRUE			Cisco	CGS-MGS-AGS	15.4(2)T4
17 DE-MUC-CW01-01		access-list 1	FALSE	access-list 1		Cisco	3560E	15.2(20170809:194209)
18 US-BOS-CW04-01	access-list 1	access-list 1	FALSE	access-list 1		Cisco	3560E	15.2(20170809:194209)
19 US-BOS-SW4			TRUE			Cisco	3560E	15.2(HI_20170202)FLO_DSGS7
20 US-BOS-CW04-02	access-list 1	access-list 1	FALSE	access-list 1		Cisco	3560E	15.2(20170809:194209)
21 Berlin-vEdge			TRUE			Cisco	WS-C4500X-32	03.04.04.SG
22 US-BOS-CR01-02		access-list 1	FALSE	access-list 1		Cisco	CGS-MGS-AGS	15.4(2)T4
23 DE-MUC-CR01-02		access-list 1	FALSE	access-list 1		Cisco	CGS-MGS-AGS	15.4(2)T4

## 7.1.1 Prerequisites

You can manually create an ADT to store your Golden Template for the standard ACLs. If you are not sure what configurations you can use as the standard ACLs, please refer to Section 7.2 on how the NetBrain system can help you identify these configurations.

To create a Golden Template table manually,

- 1. Click the plus icon 🕀 and click **New Automation Data Table**.
- 2. In the New Automation Data Table popup:
  - a) Enter the ADT Name, e.g., **Golden Template Table**.
  - b) Select the **Location** you wich to store the ADT.
  - c) Click **Save** and wait a moment for ADT to open.

Ð	New Map	Rew Device Group	New Dat Temp	
Recents	·	<b>`</b>	Temp	New Automation Data Table
Network	L. New Intent	New Automation Data Table	R New Ru	Name: Golden Template Table a
Files	New Intent Dashboard	New Network Change		Cancel Save c

- 3. In the Automation Data Table Manager, click the Add Data Manually dropdown, and then select the Add Table Column.
- 4. In the **Add Table Column** popup, define the **Display Name** as *Feature Name*, keep the date type as the default value (**String**) and click **OK**:

Table Builder	Last Updated at: N/A 🛛 🌯 Rebuild Table	Add Data Manually 🛁 🖬
	Add Table Column	Add Table Column 3
	Display Name: Feature Name	
	Column Name: Feature_Name	
}	Data Type: 🔄 String 🗸 🗸	
	Cancel OK	

Similarly, create three more columns: **Golden Configuration**, **Site**, and **Rep Device**. The golden configuration of an access list can be different for the different sites.

- 5. Click the Add Data Manually dropdown and click Add Table Row.
- 6. In the **Add Table Row** popup, enter the column values for all columns, then click **OK**.
  - **Note**: This data is for illustration purposes only; you should choose the best golden configuration for your network.
  - a) Feature Name: *acl\_1*
  - b) Golden Configuration:

access-list 1 permit 192.168.0.0 0.0.255.255

access-list 1 permit 10.0.0.0 0.255.255.255

- c) Site: My Network\NA\US-BOS (you can add more sites).
- d) Rep Device: **US-BOS-CW** (its configuration of ACL 1 is used as a golden configuration.)

Add Table Row Add Table Row 5	Table Builder         Lag	st Updated at: N/A 🛛 🍳 Rebuild Table	Add Data Manually 🗸 🚍
S Feature Name     acl_1     acl_1     access-list 1 permit 192.168.0.0 0.0.255.255     b     access-list 1 permit 10.0.0.0 0.255.255     b     S Site     My Network\NA\US-BOS     C     Image: Contract of the second se	dd Table Row	6	Add Table Row 5
S Feature Name acl_1     S Golden Configuration     access-list 1 permit 192.168.0.0 0.0.255.255     b     access-list 1 permit 10.0.0.0 0.255.255     b     S Site     My Network\NA\US-BOS     C	Column	Value	
access-list 1 permit 10.0.0 0.255.255.255	s Feature Name	acl_1	
	s Golden Configuration		
S Rep Device US-BOS-CW d	S Site	My Network\NA\US-BOS	
	s Rep Device	US-BOS-CW d	
		Cancel OK	

7. Similarly, add another row for the **acl\_10** feature.

Add Table Row	<b>7</b> ×
Column	Value
s Feature Name	acl_10 a
S Golden Configuration	access-list 10 permit 8.8.8.8 access-list 10 permit 8.8.4.4
s Site	My Network\NA\US-BOS
s Rep Device	US-BOS-CW d
	Cancel OK

## The final Golden Template Table will be:

🗓 Automation Data Table Man	ager				<b>1</b>	lelp
earch Q 🔉 «	B G	olden Template Table	Table Builder	nst Updated at: N/A 🛛 🔍 Rebuild Ta	able Add Data Manually 🗸 🗏	: d
<ul> <li>Plugin Test (19)</li> <li>Representative Methodology (4)</li> </ul>	Desci	ription: Type description here				
Reverse Engineering MVS (10)	Items	: 2 Rows 4 Columns		Search	Q T Advanced Filter: Undefined	C
🔺 📄 Sachin_TW (6)						÷.
🚯 CA test	No.	s Feature Name	S Golden Configuration	s Site	s Rep Device	-
🚯 Cisco CVE	1	acl_1	access-list 1 permit 192.168.0.0 (	My Network\NA\US-BOS	US-BOS-CW	
B Cisco CVE Security Advisory	2	acl_10	access-list 10 permit 8.8.8.8	My Network\NA\US-BOS	US-BOS-CW	
🖽 Golden Template Table	ļ					_
🐻 NIST Compliance						
🖽 Test Data						

## 7.1.2 Create Intent to Check ACL Against Golden Template

In this section, you will create a seed Intent (**Check ACL Against Golden Template**) to check the configuration drift of ACLs against those you just defined in 7.1.1. Later, you will use the **Intent Replication Wizard** to replicate to all devices you want to check the configuration drift.

From the **Intent Manager**, create a new Intent, **Check ACL against Golden Template**, and choose **US-BOS-CW-03-01** as the seed device.

#### 7.1.2.1 Define Variables with Visual Parser

In this section, you will learn how to parse the Variable for ACI list configuration using Variable Operation, **LinesByKeyword**.

- 1. Click + Add Config Diagnosis to open the Configuration Diagnosis window.
- 2. Click **Retrieve** to retrieve the data from the **Live Data**.

The source data is displayed in the **Define Variable** pane. You can edit this data depending on the use case.

3. In the + New Pattern dropdown, select Single Variable parser.

Network Intent (Edit Mode)		×
Check ACL against Golden Template     Diagnosis Tree		Run with Live Data Save 3 Help =
Type description here		💦 Intent Map: Select 🗸
I Seed Logic		
🚳 + Device		Intent Variables: Manager Tag: + Add 🔲
✓	Type Description here	1 5 + Add Config Diagnosis 🔤 + Add CLI Diagnosis
No content has been added.		
Configuration Diagnosis		× ;
US-BOS-CW03-01 Retrieve 2 th Live Data 1. Define Variable	2. Define Diagnosis	
Format1 V +		Test on Devices: 0 🚍
() Double-click a variable to parse. Select multiple lines to parse a table.	Critical Variable (0) No Pattern	✓ + New Pattern ✓
Current Device         07/15/2024 01:03:48 PM         Search           142         permit inp 172.16.0.0 0.0.255.255 any         143         permit icp mp 172.16.0.0 0.0.255.255 any           144         permit up ony any         144         permit up ony any           145         permit up 10.0.0.0 0.255.255.255 any           146         permit up ony any           146         permit up 0.0.0.255.255.255           148         access-list 1 permit 102.160.0.0 0.0.255.255.255           159         access-list 5 permit 1.1.1           151         access-list 5 permit 1.0.1.1           152         access-list 10 permit 0.80.4.8           153         access-list 10 permit 8.8.4.4           154         access-list 10 permit 8.8.4.4           155         access-list 10 permit 8.8.4.6           155         access-list 1.16.0.8	Q ▲ ▼ Sample data	Auto Pattern Single Variable Table Paragraph Advanced > Id from input text to start

- 4. Define **Var Line 1** to retrieve the configurations of **access-list 1**, which may include multiple lines starting with the keyword **access-list 1**. For this type of text, you can use the function **LinesByKeyword** to extract all data:
  - a) Click + Parse Lines in the **Output** pane, and the **Parser Lines** window appears.
  - b) Set the Variable name as *acl\_1\_config*.
  - c) Select **The line contains keyword** option, and enter the keyword, **access-list 1**.
  - d) Click **Apply** to close the **Parser Lines** window. The result includes the configurations of other access lists, such as access-list 10, since these lines also include the keyword **access-list 1**.

) 2. Define D	)iagnosis		
{		Test on Devices: 0	
😡 Variabl	es1 🗸 🗸 Type:	Single ? + New Pattern ∽ 4 =	
+ Field		Parse Lines	×
		Name: acl_1_config	
		O The line of variable: Select	~
)		O The line between: Select to Select	~
		The line contains keyword: access-list 1 C	(×)
2		Output:	
		\$acl_1_config=	<b>^</b>
Output	+ Parse Lines	access-list 1 permit 192.168.0.0 0.0.255.255	
{		access-list 1 permit 10.0.0.0 0.255.255.255 access-list 10 permit 8.8.8.8	
{		access-list 10 permit 8.8.4.4	
Ş		access-list 10 remark Google Public DNS	
l	ᠵᢦᢑ᠆᠆᠆᠆᠆᠂᠂ᢦ᠆ᠬᢑ᠆ᠬ᠆᠂᠆᠆᠆᠆᠆᠆	access-list 17 permit 172.16.0.8	
		access-list 17 permit 172.16.0.9	-
		Pattern: LinesByKeyword[\$acl_1_config]:access-list 1	
		Cancel	Apply d

e) To get the exact match for the access-list 1, you can modify the Var Line 1 by adding a space after 1 (which excludes the numbers 10-19) and add the *\$\_dummy* (to match any text after the keyword) at the end of the pattern.

The final Var Line 1 will be: *LinesByKeyword[\$acl\_1\_config]:access-list 1 \$\_dummy* 

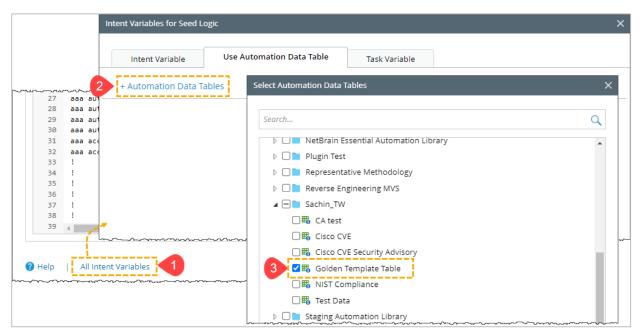
2. Define Diagnosis			
			Test on Devices: 0
🗷 Variables1	V 之 Type: Single 💡	+ New Pattern 🗸	Ξ
Var Line 1	inesByKeyword[\$acl_1_config	]:access-list 1 \$	=
	No text line matched		\$_dummy
			\$float \$int
+ Field			\$mstring
			\$string

- f) Similarly, create Var Line 2 for the access list 10: LinesByKeyword[\$acl\_10\_config]:accesslist 10 \$\_dummy.
- g) Click **Apply** to save the parser settings.

#### 7.1.2.2 Add ADT Table as Intent Variables

You have parsed the configurations of the access lists. In the diagnosis, you will compare them against the golden configurations defined in the ADT. In order for an intent to refer to the ADT elements, you need to add the ADT table as **Intent Variables**:

- 1. Click **All Intent Variables** in the bottom-right corner.
- 2. Go to the Use Automation Data Table tab and click + Automation Data Table to select ADT.
- 3. From the **Select Automation Data Table** window, select the **Golden Template Table** ADT and then click **OK**.



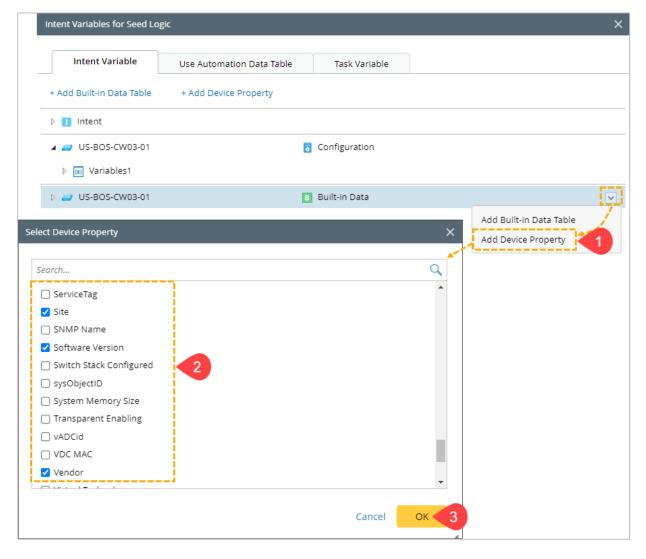
The data of the ADT table can be used in the intent diagnosis.

Intent	Variables for Seed Logic			×
	Intent Variable Use A	utomation Data Table	Task Variable	
+	Automation Data Tables			
	Golden_Template_Table			
	Feature_Name(s	Golden_Configura	Site(string)	Rep_Device(string)
	acl_1	access-list 1 permit 192.1	My Network\NA\US-BOS	US-BOS-CW
	acl_10	access-list 10 permit 8.8.8	My Network\NA\US-BOS	US-BOS-CW

#### 7.1.2.3 Add Built-in Device Properties

Similarly, you can add the **Built-in Device Properties** to the Intent Variable so that the intent can use these variables. For example, adding the model and software version to a report can be useful.

- 1. Click do f the Device **US-BOS-CW03-01**, and click **Add Device Property** from dropdown.
- 2. From the **Select Device Property** window, select the properties you wish to add. For example, add the **Site**, **Software Version**, **Vendor**, and **Model**.
- 3. Click **OK** to add device properties.



The Device properties list will be:

Device Property     Model     Display Name: Model     Site     Display Name: Site		Type: string	Default value: 3560E
x Site Display Name: Site		Type: string	Default value: 3560E
	۵		
Diselas Norras Cal	-	Type: string	Default value: My Network\NA\US
🗴 ver Display Name: Sof	ftware Version	Type: string	Default value: 15.2(20170809:194
💌 vendor Display Name: Ver	ndor	Type: string	Default value: Cisco

#### 7.1.2.4 Define Diagnosis

In the diagnosis, you will compare the configurations you defined in 7.1 against the golden configurations defined in the ADT. You will use two useful functions:

- **Match Pattern (MP)**, which compares two strings line by line and reports the differences, including the changed, added and removed lines.
- **Get\_Table\_Cell**, which will retrieve a cell value from an ADT table.

You will create a diagnosis per access list. We will use access-list 1 as an example and leave access-list 10 for your homework:

Add N	ote D Add Diagnosis	Can also	o click a variable	on the left to add au	utomati
Name:	Check acl_1 against the Golden con	fig 3	Anchor:	\$acl_1_config	~
	Type description of the diagnosis				

#### Define **If** Condition as follows:

		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~	~~~~~
🗋 Loop	Table Rows				
/ If					
Α	🥏 US-BOS-CW03	Current 🗸			
	acl_1_config	✓ MP(Rule1)	~	Get_Table_Cell(Golden_T ∨	Î
В					
	Get_Table_Cell(Go	lden_T 🗸 Is not emp	oty v		Î
С	Select Variable	~			
Boo	lean Expression: A an	d B			

#### Define **If** condition **A**:

- a) Select Variable *acl\_1\_config*.
- b) Select **Match Pattern** from the dropdown, uncheck the **Ignore Order of Lines** checkbox (the order of configurations matters for the access list), and then click **OK**. You can leave the rule name as the default or change the name if you have multiple match pattern rules in one diagnosis.

		Loop Table Rows			
			Current 🗸	·i	
		a acl_1_config B Select Variable	Equals     Does not equal	×	~ 🗇
Match Pattern	ſ		Contains X Does not contain		
Rule Name:	Rule1		Is empty Is not empty Subnet contains		Save to Incident 📃
Compare Paragraph by:	Exact Match	tue identical matched lines	In subnet     Match pattern		
	Ignore Order of lines	ave identical matched intes	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	www.
Compare Each Line By:	Equal O Contain     Î The line is idential to an expa	anded pattern line			
Learn more about Match	Pattern rule	Cancel			

c) To retrieve the golden configuration from the ADT table, select **Expression** from the dropdown.

	{ _ Loo { ~ If	p Table Rows	~~^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~	an a
	A	US-BOS-C acl_1_config	Current V V MP(Rule1)	~	
Expression				×	US-BOS-CW03-01
Expression:		at Table Cell			acl_1_config     acl_10_config     acl_10_tonfig     t ≡
+ 🐼 Variable 🛛 + 🔝 Functio	n 🕴 🎛 Ge	et_Table_Cell			Result
😮 Help			Cancel OK		Expression C Define Variable

d) In the **Expression** popup, click the **+ Get\_Table\_Cell** function.

- e) In the **Get Table Cell** window, select the ADT table, column, and condition to retrieve a table cell:
  - i. Select the table Golden\_Template\_Table.
  - ii. Select the Column Golden\_Configuration.
  - iii. Define Row Matching Condition:
    - A: Golden\_Configuration Equals "acl\_1".
    - B: Site Contains site
    - Boolean Expression: **A** and **B**

The Final Expression will be:

Get\_Table\_Cell (Golden\_Template\_Table, Golden\_Configuration, <Condition: A and B>)

Get Table Cell		e				×
Retrieve Cell Value	from:					
Select Table: Gold	en_Template_Table					~
From Column:	Golden_Configurati	on ii				~
From the Row Ma	atching Condition: ┥					
A Golden_Cont	figuration 🗸	Equals	~	"acl_1"	~	•
B Site	~	Contains	~	site	~	1
C Select Variab	ole ~					
Boolean Expressi	on: A and B					
i						i
Expression: Get Tal	ble Cell (Golden Ter	nplate_Table, Golden_0	Configuration. <c< td=""><td>ondition: A and B&gt;)</td><td></td><td></td></c<>	ondition: A and B>)		
F	(		·o-····, ·	····· <b>·</b> ···· <b>·</b> ·······················		
				с	ancel O	К

4. Define **If** Condition **B**.

Here you want to test whether the golden-config is defined or not. If it is not defined, then we should not create any message. Follow the same instructions of step 1 to use the function *Get\_Table\_Cell* to retrieve the golden template for the *access-list* 1 and check that it is not empty.

The final expression will be:

# Get\_Table\_Cell (Golden\_Template\_Table, Golden\_Configuration, <Condition: A and B>) is not empty

5. Define Intent Output message.

Enter a message under the **Then** and **Else** output areas to appear as the result of the diagnosis.

Then a	5	
📄 Diagnosis Mess		🗌 Save to Incident 🔳
✓ \$1	this_device: acl_1 Golden Compliance Passed.	
🔽 🖪 Set Status	Code for Device:	
Success 🗸	\$this_device: acl_1 Golden Compliance Passed.	
🖌 🔄 Set Status	Code for Intent:	
⊘ Success 、 dd Logic ∨	\$this_device: acl_1 Golden Compliance Passed.	
dd Logic √	\$this_device: acl_1 Golden Compliance Passed.	Delete
dd Logic ↓ Else b		☐ Delete
dd Logic ↓ y Else b Diagnosis Mess		☐ Save to Incident
dd Logic ↓ y Else b Diagnosis Mess	sage: this_device: acl_1 golden compliance failed. Rule \$Rule1.Result, Missing Lines	☐ Save to Incident
dd Logic V Y Else b Diagnosis Mess Set Status	sage: this_device: acl_1 golden compliance failed. Rule \$Rule1.Result, Missing Lines	☐ Save to Incident GRule1.Unmatched_lines
dd Logic ↓ Y Else b Diagnosis Mess Set Status	sage: this_device: acl_1 golden compliance failed. Rule \$Rule1.Result, Missing Lines : Code for Device: \$this_device: acl_1 golden compliance failed. Rule \$Rule1.Result, Missing	☐ Save to Incident GRule1.Unmatched_lines

You need to set the unmatched lines rule to check for missing IP addresses by comparing them with the Golden configuration table.

The diagnosis message will be: **\$this\_device: acl\_1 golden compliance failed. Rule \$Rule1.Result, Missing Lines \$Rule1.Unmatched\_lines**.

# 7.1.3 Modify Intent to Create CSV File

Often, you want to export the results into a CSV file. In this section, you will learn how to do this. Since we want to export the golden config to the CSV file, we will create an intent variable to hold the value first.

#### 7.1.3.1 Add Intent Variable

To add an Intent variable:

- 1. In the **Network Intent (Edit Mode)** window, go to the  $\equiv$  menu and click **Intent Variables**.
- 2. In the Intent Variables for Seed Logic window, click the **Intent** section and then click **+ Add Intent Variable**.
- 3. In the **Add Intent Variable** popup fields, enter the following values:

Fileds	Value
Variable Name	golden_config
Туре	string
Initial Value	no value

- 4. Click **OK** to save the Intent Variable.
- 5. Click **Close** to exit from the Intent Variable for Seed Logic window.

Network Intent (Edit Mode)	×
Check ACL against Golden Template  Diagnosis Tree  Run with	Live Data Save 🛛 Help 😑
Type description here	Intent Settings
Intent Variables for Seed Logic X	Full Settings for Template
Intent Variable Use Automation Data Table Task Variable	Add Intent Diagnosis Block
+ Add Intent Variable	Di Add Diagnosis via Auto Intent Switch Devices
Add Intent Variable	Define Abstract Named Tag
2 🖉 US-BOS-CW03-01	Export Save as
Variables1         Variable Name:         golden_config           golden_config	View Original Text for Diagnoses
Type: string	View Summary Text for Diagnoses
10 service timestamps log datetin 11 no service password-encryptio Initial Value:	Intent Replication Wizard
12     service compress-config       13     !       14     hostname US-BOS-CW03-01   Cancel OK 4	Auto Intent Wizard
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

#### 7.1.3.2 Set Intent Variable in Diagnosis

To set the value of an Intent variable,

- 1. Click **Edit Diagnosis** to set the intent variable under the **Define Diagnosis** tab.
- 2. Click **Define Diagnosis** to access your created diagnosis.
- 3. Open the **Check acl\_1 against the Golden config** diagnosis to set the Intent Variable.
- 4. In the **Add Logic** dropdown, select **Advanced > Set Intent Variable**.
- 5. In the Set Intent or Task Variable dropdown, select Golden\_config.

Network Intent (Edit Mode)	×
I Check ACL Against Golden Template	Runwith Live DataSaveImage: Help
Type description have 2. Define Diagnosis	💦 Intent Map: Select 🗸
Add Note D Add Dia	gnosis Can also click a variable on the left to add automation.
Check acl_1 against	□ Save to Incident
Check acl_10 agains Intent Data View	Edit Diagnosis
Send Email	Operate on Table sed.
Follow-up Intent	Set Intent Variable
Set Intent Baseline	Call Qapp ssed.
{ Advanced >	Call Webhook API
Add Logic V	Hyperlink
↓ Else	Add Command to Benchmark
Diagnosis Message:	Update ADT Dataset
E Set Intent or Task Variable	:
5 golden_config v =	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

 To add the variable value, you will use the Get\_Table\_Cell expression. Since you already have the function definition earlier, you can copy and paste it here (using the shortkeys Ctrl + C and Ctrl + V) instead of going through the UI: Get\_Table\_Cell (Golden\_Template\_Table, Golden\_Configuration, <Condition: A and B>).

Get_Table_Cell(Golden_T	Γε マ Is not empty γ	Ē	
C Select Variable	~		
Boolean Expression: A and B		Expression	Edit   Delete
Then		Expression: Get_Table_Cell(Gold	den_Template_Table,Golden_Configuration, <condition< td=""></condition<>
Else		+ 🕅 Variable 🛛 + 🔂 Fun	iction + 🎟 Get_Table_Cell
🗎 Diagnosis Message:			Select the entire expression
\$this_device: acl_1 go	Iden compliance failed. Rule \$Rule1.Result	t, N	and press "Ctrl + C" to copy in
Set Status Code for Device:		(7) Help	Cancel UK
● Error	cl_1 golden compliance failed. Rule \$Rule1	- 1	
Set Status Code for Intent:			
● Error	cl_1 golden compliance failed. Rule \$Rule1	Result, Missing Lines \$Rule1.Unmat	
Set Intent or Task Variable:		■	Use the shortcut "Ctrl + V" to paste the expression.

## 7.1.3.3 Add Logic to Export CSV File

In this section, you will learn how to export the following columns to a CSV file: **Device, Golden Config, Current Config, Matched, Missing Lines, Extra Lines, Vendor, Model, and Software Version**:

- 1. Open the **Check acl\_1 against the Golden config** diagnosis.
- 2. In the **Add Logic** dropdown, select **Advanced > Export CSV Report**.
- 3. Click **Define** to open **Export Data to CSV for <device\_name>**.
- 4. Click **Manage All CSV Reports** to open the Intent Settings window.

$\sim \sim$	Add Note D Add Diagnosis	Set Intent or Task Variable:
Check acl_1 against the G	1 v Then	<pre>golden_config v = Get_Table_Cell(Golden_Template_Table,Golden_Configuration,<condition:a and="" b<="" pre=""></condition:a></pre>
Check acl_10 against the G	Diagnosis Message:	Export to CSV Report Define
	Sthis_device: acl_1 Golden Compliance Pa	Add Logic 🗸
	Set Status Code for Device:	
	Vice: acl_1 Golden Complian	nce Passed.
	Draw Map > Export CSV Report	Export Data to CSV for US-BOS-CW03-01
	Send Email Operate on Table	ssed Export the row data of the single value to CSV Report.
	Follow-up Intent Set Intent Variable	No CSV Report
	Set Intent Baseline Call Qapp	
	Advanced > Call Webhook API	Ta
2	Add Logic 🗸 🔰 Hyperlink	
	✓ Else Add Command to Bench	mark
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Update ADT Dataset	

- 5. Add a CSV file.
  - a) Enter the CSV Name, e.g., ACL Configuration Drift.
  - b) Enter Columns name separated by commas, e.g., Device, Vendor, Model, Software Version, Golden Config, Device Config, Matched, Missing Lines, Extra Lines.
  - c) Select the Save CSV Reports to Files checkbox.
  - d) Click the **Browse** button to select the **Location** to save the CSV files.
  - e) Click **OK** to save the settings.

		Embedded Incident	CSV Report Files	
ntent Map and Data View	v Execution Settings	Embedded Incident	CSV Report Files	Follow-up Intent
Define the CSV report fil	es			
+ Add CSV				
				 m
* CSV Name: ACL	Configuration Drift_acl1			
	a			
* Columns: Devi	ce, Golden Config, Current Con	fig, Matched, Missing Lines, I	Extra Lines, Vendor, Mode	l, Sorta
	_			
🗹 Save CSV Repor	t to Files			
* Loca	tion: Private		Browse	
	Filvate		DIOWSE	
* CSV	File: ACL Configuration Drift_	acl1		
	The CSV file name will I	be reset as file name + inten	t name when this intent	
	is used as intent templ	ate/intent cluster.		

6. Select CSV Report name ACL Configuration Drift from the dropdown.

n mapping from th	he single value to t	he selected CSV R	eport.7				
Golden Config	Current Config	Matched	Missing Lines	Extra Lines	Vendor	Model	Software Version
golden_config 👻	acl_1_config 👻	Result 🗸	Unmatched_lir 🗸	Unused_patter 🗸	vendor 🗸	model 👻	ver 🔻
golden_comig +	aci_i_coning •	Nesure +	onnached_iii •	onused_patter +	Vendor •	·	vei
	Golden Config	Golden Config Current Config	Golden Config Current Config Matched		Golden Config Current Config Matched Missing Lines Extra Lines	Golden Config Current Config Matched Missing Lines Extra Lines Vendor	Golden Config Current Config Matched Missing Lines Extra Lines Vendor Model

7. Define the column value from the dropdown to the selected CSV report.

Column	Value
Device	\$this_device
Golden Config	golden_config
Current Config	acl_1_config
Matched	Result
Missing Lines	Unmatched_lines
Extra Lines	Unused_pattern_lines
Vendor	vendor
Model	model
Software Version	ver

8. Click **OK** to save the setting.

The **Export CSV Report** will be added to the Diagnosis.

2	>	2. Define Diagnosis	
Q ~ ^		Add Note D Add Diagnosis	Can also click a variable on the left to add automation.
Check acl_1 against the Gold		Diagnosis Message:	Save to Incident =
Check acl_10 against the Go		Sthis_device: acl_10 Golden Compliance Passed.	
		Image: Set Status Code for Device:         Image: Success v       Sthis_device: acl_10 Golden Compliance Passed.         Image: Set Status Code for Intent:         Image: Success v       Sthis_device: acl_10 Golden Compliance Passed.         Image: Success v       Sthis_device: acl_10 Golden Compliance Passed.	
		E. Set Intent or Task Variable:	=
		<pre>golden_config</pre>	Configuration, <condition:a and="" b="">)</condition:a>
		Export to CSV Report: Define CSV Name: ACL Configuration Drift_acl10	=

# 7.1.3.4 Run the Intent and Check CSV Report

Run the Intent. Click **CSV Report** from the dropdown menu and switch the tab to see the results.

	Network Intent (View Mode) - All Ne	etwork Intents/Sachin/Check ACL	. against Golden Template			×		
	I Check ACL against Golden Te	716/2024 04:20 PM ∨       Image: Compliance failed. Rule Faise, Missing Lines access-list to the security of the secu			🕺 Open 📄 0 🔥 0 🖌	Edit =		
	Result: 07/16/2024 04:20 PM ~		Ver en View Ersteine		Run 🗸 with L	ive Data		
				5	4	Viewv		
	✓				4 Diagnosis Tree Open Intent Map	view		
CSV Report	2 Building configurati 3 Current configuratio 5   6   Last configuration 7   8 version 15.2 9 service timestamps 1 10 no service particestamps 1 11 no service compress-con	run Jon n : 6240 bytes n change at 10:16:11 EET Mon Hebug datetime msec encryption			View on Current Map CSV Report Draw All Devices on Map Expand All Diagnoses Collapse All Diagnoses View Original Text for Diagnos View Summary Text for Diagno View Execution Log			x
ACL Configuration Drift Items: 1 row 9 columns	ACL Configuration							
Device	Golden Config	Current Config	Matched	Missing Lines	Extra Lines	Vendor	Model	Software Version
US-BOS-CW03-01	access-list 10 permit 8.8.8.8	access-list 10 permit 8.8.8.8	False	access-list 10 permit 8.8.8.8		Cisco	3560E	15.2(20170809:194209)

## 7.1.4 Use Intent Replication Wizard

In this section, you will use the **Intent Replication Wizard** to replicate the intent we just created and the replicated intents to a new ADT table.

- 1. Select Intent Replication Wizard from the menu.
- 2. The seed intent is automatically set.
- 3. Select a device group and create a new ADT. Add the built-in device properties as the additional columns.

	Seed Intent	Define ADT	3 Replication Settings
	[[	Create a New ADT	Use an Existing ADT
ew ADT by	y Device Group		×
Nam	e: ACL Configuration Drift	Location: Sachin	ı_TW (i) ~
Descriptio	on:	Se	elect Device Groups
	m Selected Device Groups: 0 Device Group Select Device	v	(ii) ch Q
• Addition	Column Data	Troup	All Device Groups
	Device	Device	□S BGP Devices (44)
	S Device Group	Device Grou	Cisco Devices (95)
	s Hostname	Hostname	☐S Firewall (15) ▲
	s Mgmt IP	Mgmt IP	▷ □ ■ Ahmed
	S Mgmt Interface	Mgmt Interf	▷ 📄 Anurag
	s Device Type	Device Type	Automation Library
	s Vendor	Vendor	C     Automation Library Kunal     D     Automation Library PKG
_	s Model	Model	▷ Automation Library Ty
			5 m 6 5 5 1 1 666
	s Software Version	Software Ver	
_	s Software Version s Serial Number	Software Ver Serial Numb	Cancel
			Cancel OK
	s Serial Number	Serial Numb	Cancel OK
	s Serial Number	Serial Numb	Cancel OK

4. In the **Replication Settings**, you select the same device group as the intent qualification.

In this step, you should also change the **Critical Variable** setting. By default, the system automatically selects all variables in the **If** conditions as critical variables. In this case, the variable **\$acl\_1** is set as the critical variable. However, we want to create an alert if the **access\_list 1** is configured, and so we want to remove **\$acl\_1** from the critical variables:

- a) Click the Full Intent Template Setting link.
- b) Click the Manually Select option to see the Critical Variable section.
- c) In the Critical Variable section, uncheck *acl\_1\_config* and *acl\_10\_config* Variables except Configuration.

	5					
		Replication Settings	Re	eplicate Intent		
			🛞 Full	Settings for Template	a	
ttings for Intent Templa	ate					
Serve as Template for:	Device-based Replication	_	Replication 🗌 Ena	ble Neighbor Pair Replicatio	n	
Intent Qualification	Macro Variable	Critical Variable	Advance Settings			
t variables as Critical Va	ariables to qualify devices wh	nich can be matched with	eed device parsers: 🔒			
utomatically Select All D	Diagnosis Variables					
anually Select						
anually Select						
I Check ACL against Go					Critical Variables	Replicate
Check ACL against Go US-BOS-CW03-01	olden Template	1			Critical Variables	Replicate
Check ACL against Ge US-BOS-CW03-01 Boolean Algebra A	e CLI Command	'n			1 Variable Selected	<ul> <li>✓</li> <li>✓ onfiguration</li> </ul>
Check ACL against G US-BOS-CW03-01 Boolean Algebra	e CLI Command	1			1 Variable Selected	

d) Click **OK** to save Intent Template Settings.

5. In the Replicate Intent, add additional columns such as **Intent Status Code** and **Last Execution Time**.

Seed Intent	Define ADT	Replication Settings	Replicate Intent 5
DT Columns:			Additional Columns 🗸
Column Data	Column Name	Tag	Replicated Intent
Replicated Intent	ACL Configuration Drift	0 tags	Intent Message
s Intent Status Code	Intent Status Code		Intent Status Cod
Last Execution Time	Last Execution Time		Device Status Coo
			Intent Devices
			🗌 Intent Map
			Intent CLI Comma
			🗹 Last Execution Tir
			Save and Replicate

The table will now be populated with devices and the replicated Intents (**ACL Configuration Drift**).

🦥 Αι	itomation Data Tab	ie manage.					🕑 Help
👪 AC	L Configuration Drift	Table B	Builder Last Updated at: 0	08/07/2024 03:46 PM 🛛 🔍 Rebuild T	able	Replicated Intent	Add Data Manually 🗸 📃 🖬
Descr	iption: Type description here					Replicated intent	
Items	: 95 Rows 8 Columns					Search	Q Y Advanced Filter: Undefined O
No.	Device	S Mgmt IP	5 Vendor	S Model	Software Version	ACL Configuration Drift	S Intent Status Code ③ Last Execution Tim
1	Berlin-R1	172.16.8.60	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 💿	·
2	Berlin-vEdge	192.168.0.1	Cisco	WS-C4500X-32	03.04.04.SG	Check ACL against Golden Templ 👁	
3	DE-MUC-CR01-02	10.20.1.3	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 👁	
4	DE-MUC-CW01-01	10.20.1.4	Cisco	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 👁	
5	DE-MUC-CW02-01	10.20.1.5	Cisco	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 👁	
6	DE-MUC-CW03-01	10.20.1.6	Cisco	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 👁	
7	ISP-P02	4.0.0.2	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 👁	
8	ISP-PE01	1.0.0.2	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 👁	
9	ISP-PE02	2.0.0.2	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 👁	
10	ISP-PE03	3.0.0.2	Cisco	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 👁	
11	ITE_EXTEND	192.168.30.207	Cisco	WS-C3560X-48P	15.2(4)E7		
12	JP-TYO-CR01-01	192.168.20.1	Cisco	Catalyst 4500 Virtual Sw	15.4(2)T4	Check ACL against Golden Templ 👁	
13	JP-TYO-CW01-01	10.30.0.4	Cisco	3560E	15.2(20170809:194209)		
14	JP-TYO-CW01-02	10.30.0.5	Cisco	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 💿	
15	JP-TYO-CW02-01	10.30.0.6	Cisco	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 👁	

## 7.1.5 Run Intent Once and Rebuild Table

Run all intents in the ADT once and rebuild the bale. Review the intent results (the column **Intent Status Codes**).

₽	Automation Data Ta	ble Manager			😢 Help
»	🚯 ACL Configuration Drift	Table Bu	ilder Last Updated at: 08/07/2024 03	:46 PM 🔍 Rebuild Table	Add Data Manually $\sim$
1	Description: Type description here	е			
1	tems: 95 Rows 8 Columns		Si	earch Q	TAdvanced Filter: Undefin
	s Model	s Software Version	ACL Configuration Drift	s Intent Status Code	Cast Execution Time
	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 👁	Berlin-R1: acl_1 golden cor	08/07/2024 03:47:0
	WS-C4500X-32	03.04.04.SG	Check ACL against Golden Templ 💿	Berlin-vEdge: acl_1 golden	07/16/2024 06:14:2
	CGS-MGS-AGS	15.4(2)T4	Check ACL against Golden Templ 💿	DE-MUC-CR01-02: acl_1 gc	07/16/2024 06:50:3
	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 💿	DE-MUC-CW01-01: acl_1 g	07/16/2024 06:50:2
	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 💿	DE-MUC-CW02-01: acl_1 g	07/16/2024 06:50:3
	3560E	15.2(20170809:194209)	Check ACL against Golden Templ 👁	DE-MUC-CW03-01: acl_1 g	08/07/2024 03:47:0

## 7.1.6 View Summary Report

You can create a summary report from the ADT, which combines all CSV reports from all replicated intents in the column:

- 1. In the Intent column, click  $\equiv$  menu and open **View Summary Report**.
- 2. Select the time frame as per your preference and click **Create** to generate the report.
- 3. Click the **Export** to export the results into a CSV file.

w Summary Report - Replicated I	ntent 1 (56 intents)			×	
reate summary report of all the C	SV reports generated by intents	in this column:			ACL Con Run Details
only merge CSV reports generated	lin 1 Hours	Create 2	6 intent results filtered	-	Run Intents Once Run Intents via Timer
ACL Configuration	ACL Configuration Dri				Open Seed Intent
items: 56 rows 9 columns			Search	٩	Rebuild Intent-related Column Group
Device	Golden Config	Current Config	Matched	Missing Lines	Remove Empty Wrapper Intent
US-BOS-CW02-02	access-list 1 permit 192.168	access-list 1 permit 192.168	False	access-list 1 *	Enable Auto Intent
JP-TYO-CW02-01		access-list 1 permit 192.168	False	access-list 1	Export Diagnosis Result to CSV
UK-LHR-CW01-01		access-list 1 permit 192.168	False	access-list 1	View Summary Report
US-BOS-SW2			True		Export Intent Output Map Debug Empty Cells
US-NYJ-CW02-01		access-list 1 permit 192.168	False	access-list 1	
ISP-PE01			True		Tag Current Column
DE-MUC-CW01-01		access-list 1 permit 192.168	False	access-list 1	Edit
US-BOS-CW04-01	access-list 1 permit 192.168	access-list 1 permit 192.168	False	access-list 1	Delete
US-BOS-SW4			True		Set as Table Key
US-BOS-CW04-02	access-list 1 permit 192.168	access-list 1 permit 192.168	False	access-list 1	Submit Related Commands to Benchma
21		_			New Intent Dashboard

# 7.1.7 Create Intent and Summary Dashboard

Create an intent dashboard to better view the result.

ACL Config Drift Information	1					Last Refreshed at 17/07/2	024, 11:32:57	0 4	×ø	=
Summary		17/07/2024, 11:33:	00 View Report	Device Information			17/07/2024, 11:3	3:00 V	iew Repor	
		Executed evel Alerts		Cisco 105	Switch Cisco Router		56	Devie	ces	
Intent Result History							17/07/2024, 11:5	3:00 V	iew Repor	
Time Range: All $\sim$ Result: All $\sim$										
20		- Sum of Intent Alert	Status Code Count	<ul> <li>Sum of Intent Success Status Code C</li> </ul>	Count					
			<u> </u>		Mattatiatiatiatiatiatiati		N. W. W.		2.2.2	=.
161, 610, 610, 610, 610, 600, 600, 600,	0, 00, 00, 00, 00, 00, 00, 00, 00, 00,	2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	, 60, 60, 60, 60, 60, 60, 7	90,90,90,90,90,90,90,90,90,90,90,90,	10,10,10,10,10,10,10,10,10,10,10,10,10,1	601,601,601,601,601,601,601,601	(elo, elo, elo, elo	,610,610,	elo, elo,	
			Top Five In	tent Alerts						
Intent Name	Мар	Execution Time	Intent Alert Status Co	de Count Intent Success Status	s Code Count Intent Status C	ode Summary Int	tent Alert Detect	ion		
Check ACL against Golden Template J		16/07/2024, 18:50:35	2	0		1: acl_1 golden compli 1				
Check ACL against Golden Template J		16/07/2024, 18:50:35	2	0		1: acl_1 golden compli 1				
Check ACL against Golden Template	View Map	16/07/2024, 20:25:55	2	0	US-NYC-SW1: a	cl_1 golden complianc 1				

You can also create a new summary dashboard for this intent dashboard:

Configuration Drift								Last Refreshed at 17/07/	2024, 12:08:56	Ø =	
Summary	Number of De			Number of Alerts			Number of Successes				
Sumber of Intents	56				1,072						
ACL Number of Intents Number of Devices				Number of Alerts			Number of Successes				
56	56	nues		1,072			0				
Dashboard and Intent Group	Intent Results	Device Results						۹ ۲			
ACL Config Drift Information	536	Total Device Results 0 536	US-NYC-SW1	SG-SIN-CR01-01	JP-TYO-CW01-01	US-NYJ-CR01-01	SG-SIN-CW02-01	UK-LHR-CR01-02	UK-LHR-CW01	-01	
Total Alert Count $\downarrow$			12	10	10	10	10	10		10	

# 7.2 Create Golden Template

In this section, you will learn how to use intent-based automation to analyze ACL configurations of your network to find the Golden Template. First, you will create a summary report for all ACL configurations, from which you will create a pivot table to identify the common Golden configuration (if an ACLI has the same configurations for all devices in one or multiple sites, it is a good candidate for the Golden Config). This template will be used to compare and verify the ACL configurations of other network devices.

If you know the Golden Config of your network, you can skip this section.

## 7.2.1 Create Intent to Export ACL Configuration to a CSV File

First, you create an intent to export all ACL configurations into a CSV file. This intent will then be replicated to all devices with the **Intent Replication Wizard**. From the replicated Intent, you can create a summary report for all ACL configurations across your network.

Create a Network Intent, **ACL Name List**, from the **Intent Manager**, and select a target device with the ACLI configurations (e.g., *US-BOS-CW-04-02*).

Add a **Config Diagnosis**, **Retrieve** data, and define the Variable using the **Paragraph** pattern:

• ID Line: *^access-list \$int:acl\_id* 

	Format1	$\sim$		+								Test on Devices	s:
Doubl	le-click a varia	ible to p	arse. S	elect mul	tiple lines to parse a t	able.		Critical Variable ((	0)	Type: Paragraph	(	Cancel Apply	
urre	nt Device		~	07/18/2	024 01:13:01 PM	Search		Q 🔺 🔻					
										ID Line A	<pre>^access-list \$int:acl_id</pre>		Ξ
146	logging h	lost 10	.10.0	.105				P1-ID Line A	•				
					8.0.0 0.0.255.255			P2-ID Line A			147 access-list 1 permit 192.168.0.0 0.0.255.255	> 51 Lines	
148	⊿access-li	st 1 p	permit	10.0.0	.0 0.255.255.255								
	⊿access-li							P3-ID Line A					
150	⊿access-1	st 5 r	remark	Cloudf	lare Public DNS			P4-ID Line A			<u>^</u>		
	⊿access-li						-08			Output	+ Parse Lines		
	⊿access-li						- N	P5-ID Line A					
					e Public DNS		- N	P6-ID Line A		\$acl_id		`	~
	⊿access-1			t 172.1			-83	P7-ID Line A					
	<pre>⊿access-li ⊿access-li </pre>						-\\\	P7-ID Line A		1			
							-88	P8-ID Line A		1			
	<pre>⊿access-li ⊿access-li </pre>			t 172.1 t 172.1			-37	P9-ID Line A		1			
	Aaccess-li			t 172.1			-33	F3-ID LINE A		5			
	access-1			t 172.1			-86	P10-ID Line A					
	Aaccess-li						-84	P11-ID Line A		5			
					6.0.2 6.2.3.		-633	VILLEN EINE A		-			

- Define the variable **\$acl\_config** using the Variable Operator **LinesByKeyword** to extract all configurations related to an ACL.
  - a) Click + Parse Lines in the Output pane, and the Parser Lines window appears.
  - b) Set the Variable name as *acl\_configs*.
  - c) Select **The line contains keyword** option, and enter the keyword, *^access-list*.

d) Check the pattern and click **Apply** to close the **Parser Lines** window. The Var Line 1 pattern will be *LinesByKeyword[\$acl\_configs]:^access-list.* 

	2. Define Diagnosis			
		Test on Devices: 0 🛛 🗮		
	Type: Paragraph	Cancel Apply =		
	ID Line A ^access-list \$int:acl_id permit 147 access-list   permit 192.168.0.0 0.0. Output + Parse Lines a Sacl_id B × Sacl_idm a	Parse Lines Name: acl_configs b O The line of variable: Select O The line between: Select	✓ to Select	× ~ ~
Var Line 1 LinesByKeyword[\$acl_c	configs]:^access-list	The line contains keyword: ^access-list Output:		(x)
147 access-list 1 permi	it 192.168.0.0 0.0.255.255 > 51 Lines	\$acl_id ∰	\$acl_configs  access-list 1 permit 192.168.0.0 0.0.255.255	<b>^</b>
		1	access-list 1 permit 10.0.0.0 0.255.255.255	
Var Line 1 LinesByKeyword[\$acl	_configs]:^access-list <%\$acl_id%> \$_dummy	5	access-list 5 permit 1.1.1.1	
147 access-list 1 peri	mit 192.168.0.0 0.0.255.255 > 51 Lines	5	access-list 5 remark Cloudflare Public DNS	
human marine and the second se		10	access-list 10 permit 8.8.8.8	
		10 Pattern: LinesByKeyword[\$acl_configs]:^	access-list 10 permit 8.8.4.4 vaccess-list	•
			Cancel Apply 4	d

e) Modify the pattern to *LinesByKeyword[\$acl\_configs]:^access-list <%\$acl\_id%> \$\_dummy*.

### 7.2.1.1 Add Formula Column in the Parser Table

The table has two columns: acl\_id and acl\_configs. For others to better understand this table, you may want to add a new column, **acl\_name**, with the value to be **"ACL " + \$acl\_id**, such as **ACL 1**.

You can add a **Formula Column** to table variables, which converts the original variables into different formats or values:

- 1. Click **All Intent Variables** in the bottom-right corner.
- 2. Expand the **US-BOS-CW04-01 Configuration** section and then expand **acl\_table**.
- 3. Click and open **Add Formula Column** from the dropdown.
- 4. In **the Add Formula Column** window, define the following fields:
  - Name: acl\_name
  - Type: string
  - Definition: "ACL " + \$acl\_id (click \$ to open pop up to acl\_id variable)

	Intent Variables for Seed Lo	ogic		×
	Intent Variable	Use Automation Data Table	Task Variable	
	+ Add Formula Column			
	Intent			
	🖌 σ US-BOS-CW04-02	2 5	Configuration	
159				
160 161	acl_id (int)		acl_configs (strir	
162 163 164	1		access-list 1 per	mit 192.168.0.0 Refresh
165	5	Add Formula	Column	<b>4 ~</b> ×
167 168	5	Name	e: acl_name	
169 170 171	10	Тур	e: string	~
172	10	Initial Value	2:	
Help	All Intent Variables	Definition	n: "ACL " + \$acl_:	id
		Demición	+ 🖾 Variable + 🗗	
		😗 Help		Cancel OK

#### The Formula column will be:

US-BOS-CW04-02	onfiguration			
acl_table				
acl_id (int)	acl_configs (string)		acl_name (string) 差 💼	
1	access-list 1 permit 192.168.0.0 0.0.255.25	5	ACL 1	
1	access-list 1 permit 10.0.0.0 0.255.255.255		ACL 1	
5	access-list 5 permit 1.1.1.1		ACL 5	
5	access-list 5 remark Cloudflare Public DNS		ACL 5	
10	access-list 10 permit 8.8.8.8		ACL 10	
10	access-list 10 permit 8.8.4.4		ACL 10	
10	access-list 10 remark Google Public DNS		ACL 10	

## 7.2.1.2 Define Diagnosis to Export ACL Configurations

Create a new diagnosis to export the **acl\_table to a CSV file**.

Follow the same step in Section 7.1.3.3. Besides this table, you may also add the built-in device properties and export them.

	SV for US-BOS-CW0							
	data of the "acl_tab	ie" to CSV Report.						
Network Fea	ture		🝷 📑 Manag	ge All CSV Reports	5			
Define the colu	ımn mapping from t	he "acl_table" to th	e selected CSV Re	port.	1			
Device	Device Type	Feature Tag	Feature Name	Configuration	Model	Version	Site	
\$this_device	✓ subTypeName ▼	ACL -	acl_name 👻	acl_configs 🔹	model 🗸	ver 🗸	site	•
4								
						Car	ncel OK	

## 7.2.2 Create ACL Report for the Whole Network Devices

Replicate the intents to all Cisco IOS devices via the **Intent Replication Wizard**. You will create a new ADT like this:

🖷 AC	'L Name List	Table Builder	Last Updated at: 07/22/2024 12:12 PM	🌯 Rebuild Table			Add Data Manually 🗸 📃 🖬
Descr	iption: Type description here						
ltems	: 95 Rows 7 Columns					Search C	Advanced Filter: Undefined
No.	Device	5 Vendor	5 Model	S Software Version	s Site	Export ACL	Clast Execution Time
1	Berlin-R1	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List Berlin-R1 1 📀	
2	Berlin-vEdge	Cisco	W5-C4500X-32	03.04.04.5G	test-site-xinyu	ACL Name List Berlin-vEdge 1 📀	
3	DE-MUC-CR01-01	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List DE-MUC-CR01-01 1 💿	
4	DE-MUC-CR01-02	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List DE-MUC-CR01-02 1 💿	
5	DE-MUC-CW01-01	Cisco	3560E	15.2(20170809:194209)	DE-MUC	ACL Name List DE-MUC-CW01-0 💿	
6	DE-MUC-CW02-01	Cisco	3560E	15.2(20170809:194209)	DE-MUC	ACL Name List DE-MUC-CW02-0 💿	
7	DE-MUC-CW03-01	Cisco	3560E	15.2(20170809:194209)	DE-MUC	ACL Name List DE-MUC-CW03-0 💿	
8	ISP-P02	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List ISP-P02 1 📀	
9	ISP-PE01	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List ISP-PE01 1	
10	ISP-PE02	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List ISP-PE02 1	
11	ISP-PE03	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List ISP-PE03 1 📀	
12	ITE_EXTEND	Cisco	W5-C3560X-48P	15.2(4)E7	test-site-xinyu	ACL Name List ITE_EXTEND 1	
13	JP-TYO-CR01-01	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List JP-TYO-CR01-01 1 🛛 🔨	
14	JP-TYO-CR01-02	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List JP-TYO-CR01-02 📀	
15	JP-TYO-CW01-01	Cisco	3560E	15.2(20170809:194209)	JP-TYO	ACL Name List JP-TYO-CW01-01 📀	
16	JP-TYO-CW01-02	Cisco	3560E	15.2(20170809:194209)	JP-TYO	ACL Name List JP-TYO-CW01-02 1 💿	
17	JP-TYO-CW02-01	Cisco	3560E	15.2(20170809:194209)	JP-TYO	ACL Name List JP-TYO-CW02-01 1 💿	
18		- Ascence	2560E	.15.2(20170809-194209)		ACL Name List JP-TYQ-CW03-01 Q	

#### Run the replicated intents once and rebuild the table:

🖷 AC	L Name List	Table Builder	Last Updated at: 07/22/2024 12:27 PM	🌯 Rebuild Table			Add Data Manually 🗸 📃 💼
Descri	ption: Type description here						
Items:	95 Rows 7 Columns					Search	Q T Advanced Filter: Undefined C
No.	Device	S Vendor	s Model	Software Version	s Site	Export ACL	③ Last Execution Time
1	Berlin-R1	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List Berlin-R1 1 🛛 💿	07/22/2024 12:24:38 PM
2	Berlin-vEdge	Cisco	WS-C4500X-32	03.04.04.5G	test-site-xinyu	ACL Name List Berlin-vEdge 1 🛛 💿	07/22/2024 12:24:38 PM
3	DE-MUC-CR01-01	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List DE-MUC-CR01-01 1 🔍	07/22/2024 12:24:43 PM
4	DE-MUC-CR01-02	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List DE-MUC-CR01-02 1 🔍	07/22/2024 12:24:38 PM
5	DE-MUC-CW01-01	Cisco	3560E	15.2(20170809:194209)	DE-MUC	ACL Name List DE-MUC-CW01-0 💿	07/22/2024 12:24:40 PM
6	DE-MUC-CW02-01	Cisco	3560E	15.2(20170809:194209)	DE-MUC	ACL Name List DE-MUC-CW02-0 💿	07/22/2024 12:24:42 PM
7	DE-MUC-CW03-01	Cisco	3560E	15.2(20170809:194209)	DE-MUC	ACL Name List DE-MUC-CW03-0 💿	07/22/2024 12:24:40 PM
8	ISP-P02	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List ISP-P02 1	07/22/2024 12:24:40 PM
9	ISP-PE01	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List ISP-PE01 1	07/22/2024 12:24:40 PM
10	ISP-PE02	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List ISP-PE02 1	07/22/2024 12:24:42 PM
11	ISP-PE03	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List ISP-PE03 1	07/22/2024 12:24:40 PM
12	ITE_EXTEND	Cisco	WS-C3560X-48P	15.2(4)E7	test-site-xinyu	ACL Name List ITE_EXTEND 1	07/22/2024 12:24:42 PM
13	JP-TYO-CR01-01	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List JP-TYO-CR01-01 1 🛛 🔨	07/22/2024 12:24:40 PM
14	JP-TYO-CR01-02	Cisco	CGS-MGS-AGS	15.4(2)T4	test-site-yihong	ACL Name List JP-TYO-CR01-02	07/22/2024 12:24:42 PM
15	JP-TYO-CW01-01	Cisco	3560E	15.2(20170809:194209)	JP-TYO	ACL Name List JP-TYO-CW01-01 🛛 💿	07/22/2024 12:24:38 PM
16	JP-TYO-CW01-02	Cisco	3560E	15.2(20170809:194209)	JP-TYO	ACL Name List JP-TYO-CW01-02 1	07/22/2024 12:24:39 PM
17	JP-TYO-CW02-01	Cisco	3560E	15.2(20170809:194209)	JP-TYO	ACL Name List JP-TYO-CW02-01 1	07/22/2024 12:24:40 PM
18	IP:TYQ:CWQ3:Q1	- Cisclon			PTYQ	ACL Name List IP-TYO-CW03-01 🚬 🔍	07/22/2024 12:24:40 PM

View the summary report and export the results into a CSV file. The CSV file, *Netawork Feature\_ACL Name List* will be downloaded in your computer's download folder.

	nerated in: 1	Hours 🗸	Create	2 56 intent results filtere	ed						
Network Feature											
ms: 2234 rows 8 columr	ıs			Search	Q				Add	d Data Manually 🗸 🔳	<b>.</b>
Device	Device Type	Fea	ture Tag	Feature Name							
DE-MUC-CR01-01	Cisco Router	ACL		ACL 1			Search	Q	🛛 🍸 Adva	anced Filter: Undefined	0
DE-MUC-CR01-01	Cisco Router	ACL		ACL 1		Export A	CL 🛛 Run 🖹	Details 📃	🕔 Last Ex	kecution Time	Ξ
DE-MUC-CR01-01	Cisco Router	ACL		ACL 2		Run Intents (	Once	2	07/22/202	24 12:24:38 PM	-
DE-MUC-CR01-01	Cisco Router	ACL		ACL 5		Run Intents	via Timer	<b>PA</b>	07/22/202	24 12:24:38 PM	
DE-MUC-CR01-01	Cisco Router	ACL		ACL 5		Open Seed In	ntent		07/22/202	24 12:24:43 PM	
DE-MUC-CR01-01	Cisco Router	ACL		ACL 10		Rebuild Inte	nt-related Colum	n Group	07/22/202	24 12:24:38 PM	
					4-1		ty Wrapper Intent		07/22/202	24 12:24:40 PM	
Export 3					Close	Enable Auto	Intent	$-\mu$	07/22/202	24 12:24:42 PM	
			,		A		osis Result to CSV		07/22/202	24 12:24:40 PM	
Export Rep	ports				×	View Summa			07/22/202	24 12:24:40 PM	
· · · • •											
						Export Inten			07/22/202	24 12:24:40 PM	
Select CS	V Reports:					Debug Empt	y Cells			24 12:24:40 PM 24 12:24:42 PM	
	V Reports: CSV Name		Seed Intent Template	Name			y Cells		07/22/202		
			Seed Intent Template	Name		Debug Empt Tag Current Edit	y Cells		07/22/202	24 12:24:42 PM	
	CSV Name			Name		Debug Empt Tag Current ( Edit Delete	y Cells		07/22/202 07/22/202 07/22/202	24 12:24:42 PM 24 12:24:40 PM	
	CSV Name			Name		Debug Empt Tag Current Edit	y Cells		07/22/202 07/22/202 07/22/202 07/22/202	24 12:24:42 PM 24 12:24:40 PM 24 12:24:42 PM	
	CSV Name Network Feature			Name		Debug Empt Tag Current of Edit Delete Set as Table Submit Relat	y Cells Column Key ed Commands to B	enchmark	07/22/202 07/22/202 07/22/202 07/22/202 07/22/202	24 12:24:42 PM 24 12:24:40 PM 24 12:24:42 PM 24 12:24:40 PM	
	CSV Name					Debug Empt Tag Current ( Edit Delete Set as Table	y Cells Column Key ed Commands to B	enchmark	07/22/202 07/22/202 07/22/202 07/22/202 07/22/202 07/22/202	24 12:24:42 PM 24 12:24:40 PM 24 12:24:42 PM 24 12:24:40 PM 24 12:24:42 PM	
	CSV Name Network Feature		ACL Name List			Debug Empt Tag Current of Edit Delete Set as Table Submit Relat	y Cells Column Key ed Commands to B	enchmark	07/22/202 07/22/202 07/22/202 07/22/202 07/22/202 07/22/202	24 12:24:42 PM 24 12:24:40 PM 24 12:24:42 PM 24 12:24:42 PM 24 12:24:40 PM 24 12:24:42 PM 24 12:24:38 PM	
	SV Name Network Feature	vnloads →	ACL Name List	el Export •		Debug Empt Tag Current of Edit Delete Set as Table Submit Relat	y Cells Column Key ed Commands to B	enchmark	07/22/202 07/22/202 07/22/202 07/22/202 07/22/202 07/22/202	24 12:24:42 PM 24 12:24:40 PM 24 12:24:42 PM 24 12:24:42 PM 24 12:24:40 PM 24 12:24:42 PM 24 12:24:38 PM	
	SV Name Network Feature	vnloads >	ACL Name List	el Export •		Debug Empt Tag Current of Edit Delete Set as Table Submit Relat	y Cells Column Key ed Commands to B	enchmark	07/22/202 07/22/202 07/22/202 07/22/202 07/22/202 07/22/202	24 12:24:42 PM 24 12:24:40 PM 24 12:24:42 PM 24 12:24:42 PM 24 12:24:40 PM 24 12:24:42 PM 24 12:24:38 PM	
	SV Name Network Feature	vnloads → È <sup>©</sup>	ACL Name List Canc csvReport (	.el Export + (1)	· · · [Ĉ	Debug Empt Tag Current of Edit Delete Set as Table Submit Relat	y Cells Column Key ed Commands to B Jashboard	enchmark	07/22/202 07/22/202 07/22/202 07/22/202 07/22/202 07/22/202	24 12:24:42 PM 24 12:24:40 PM 24 12:24:42 PM 24 12:24:42 PM 24 12:24:40 PM 24 12:24:42 PM 24 12:24:38 PM	

The following is a sample CSV file:

	А	В	С	D	E	F	G	Н
1	Device	Device Type	Featu	Feature	Configuration	Model	Version	Site
2	DE-MUC-CR01-01	Cisco Router	ACL	ACL 1	access-list 1 permit 192.168.0.0	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
3	DE-MUC-CR01-01	Cisco Router	ACL	ACL 1	access-list 1 permit 10.0.0.0 0.2	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
4	DE-MUC-CR01-01	Cisco Router	ACL	ACL 2	access-list 2 permit 192.168.201	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
5	DE-MUC-CR01-01	Cisco Router	ACL	ACL 5	access-list 5 permit 1.1.1.1	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
6	DE-MUC-CR01-01	Cisco Router	ACL	ACL 5	access-list 5 remark Cloudflare	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
7	DE-MUC-CR01-01	Cisco Router	ACL	ACL 10	access-list 10 permit 8.8.8.8	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
8	DE-MUC-CR01-01	Cisco Router	ACL	ACL 10	access-list 10 permit 8.8.4.4	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
9	DE-MUC-CR01-01	Cisco Router	ACL	ACL 10	access-list 10 remark Google Pu	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
10	DE-MUC-CR01-01	Cisco Router	ACL	ACL 17	access-list 17 permit 172.16.0.8	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
11	DE-MUC-CR01-01	Cisco Router	ACL	ACL 17	access-list 17 permit 172.16.0.9	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
12	DE-MUC-CR01-01	Cisco Router	ACL	ACL 17	access-list 17 permit 172.16.0.4	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
13	DE-MUC-CR01-01	Cisco Router	ACL	ACL 17	access-list 17 permit 172.16.0.5	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
14	DE-MUC-CR01-01	Cisco Router	ACL	ACL 17	access-list 17 permit 172.16.0.6	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
15	DE-MUC-CR01-01	Cisco Router	ACL	ACL 17	access-list 17 permit 172.16.0.7	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
16	DE-MUC-CR01-01	Cisco Router	ACL	ACL 17	access-list 17 permit 172.16.0.1	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
17	DE-MUC-CR01-01	Cisco Router	ACL	ACL 17	access-list 17 permit 172.16.0.2	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
18	DE-MUC-CR01-01	Cisco Router	ACL	ACL 17	access-list 17 permit 172.16.0.3	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
19	DE-MUC-CR01-01	Cisco Router	ACL	ACL 20	access-list 20 permit 129.6.15.2	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
20	DE-MUC-CR01-01	Cisco Router	ACL	ACL 20	access-list 20 permit 129.6.15.28	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong
21	DE-MUC-CR01-01	Cisco Router	ACL	ACL 20	access-list 20 permit 129.6.15.29	CGS-MGS-AGS	15.4(2)T4	My Network\test-container-site-yihong\test-site-yihong

## 7.2.3 Analyze the Report to Find the Golden Config

In this section, you will analyze the Access Control List (ACL) report to identify the common ACLs that are configured by many devices on one or multiple sites. This analysis will help in determining which ACLs can be standardized into a **Golden Template Table**.

**Note**: the following is just one way to analyze the report. If you are familiar with Excel, you may find a better way to do this.

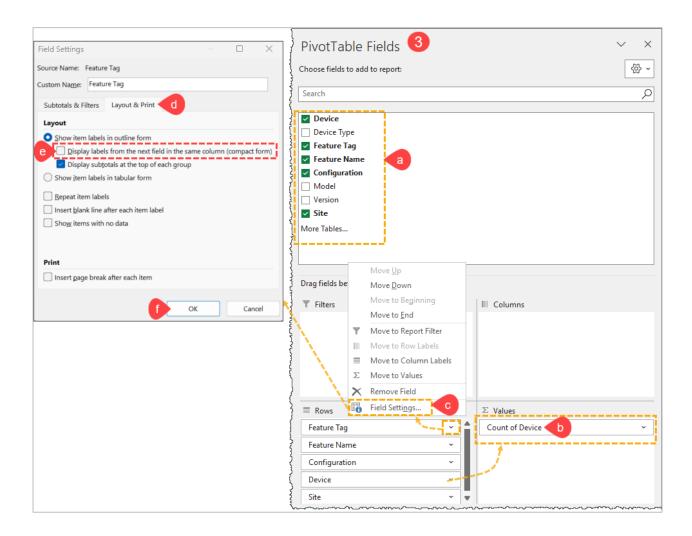
Follow the step-by-step instructions to analyze the report using the **Pivot** table.

- 1. Open the *Network Feature\_ACL Name List* CSV file from your download folder.
- 2. Create a Pivot Table.
  - a) Select the whole table, go to **Insert**, and click the **PivotTable** option.
  - b) Click **OK** in the **PivotTable from the table or range** popup.

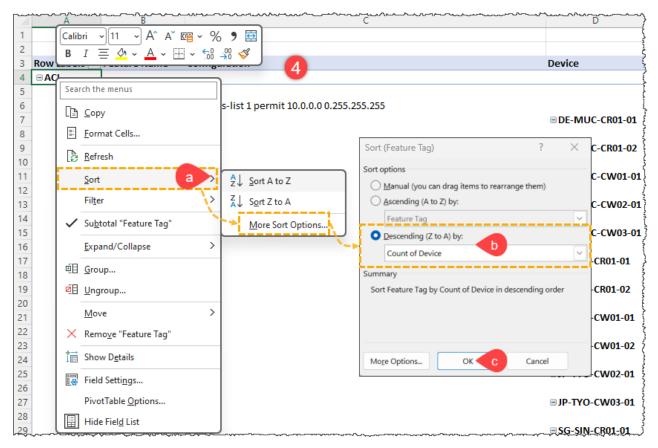
ζų	💶 AutoSave 🔵 Of	1 🗄 🔁 🤇	$\stackrel{{}_{\scriptstyle \!$	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-	File Home Ins	ert 🕴 Page Layou	t Formulas Data Review View Automate Help	10 M
	PivotTable Reader		xures Shapes Icons 3D ™ SmartArt Models → Models → Mode	×~ الله ٢
{	Tables		PivotTable from table or range ? ×	Char
{ A		< style="background-color: gray;"/> fx Devi	Select a table or range	,,,,,,,, .
<b>{</b> 1	A Device	Device Type	Iable/Range: 'Network Feature ACL Name List'!\$A:\$H	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2	DE-MUC-CR01-01	Cisco Router	Choose where you want the PivotTable to be placed	255.255
<b>ξ</b> 3	DE-MUC-CR01-01	Cisco Router	New Worksheet	55.255
<u></u>	DE-MUC-CR01-01	Cisco Router	Existing Worksheet	. <b>0.0.255</b> ]
35	DE-MUC-CR01-01	Cisco Router	Location:	}
\$6	DE-MUC-CR01-01	Cisco Router	Location	lic DNS 🥤
1 27	DE-MUC-CR01-01	Cisco Router	OK D Cancel	
j8	DE-MUC-CR01-01	Cisco Router		1

- 3. Define a PivotTable Fields.
  - a) Choose the fields to add to the report: **Feature Tag**, **Feature Name**, **Configuration**, **Device**, and **Site**.
  - b) Drag the **Device** field into the **Values** column to cerate another filed **Count of Device**.
  - c) Click menu at the right side of the **Feature Tag** field and select the **Field Settings**.
  - d) In the Field Settings window, go to the Layout & Print tab.
  - e) Uncheck the option **Display labels** from the next field in the same column (compact from).
  - f) Click **OK** to apply the field settings.

NOTE: Apply the Field Settings to all the fields.



- 4. Sort the Fields.
  - a) Right-click the field name and select **Sort** > **More Sort Options**.
  - b) In the Sort popup, select Descending (Z to A) by Count of Device.
  - c) Click **OK** to apply settings.



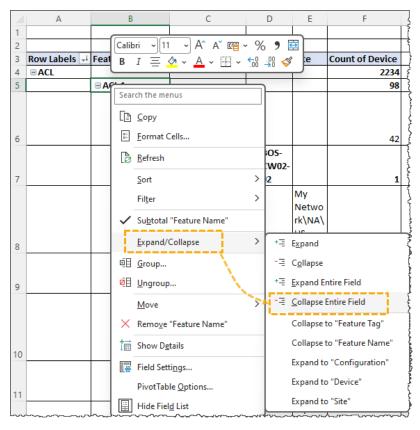
5. Wrap the Wrap Text in the entire sheet to view the Configuration easily.

×	Auto	oSave 🤇	Off	<b>日</b> り~	Q Net	work Feature_A	CL Name List	- Read-Only	~		₽ Sea	arch	
F	ile H	Home	Insert	Page Lay	out Formulas	Data Revi	ew View	Automate	Help				
	Paste	从 Cut È Copy ダ Form	↓ at Painter	Calibri B 2	- 「 <u>∪</u> -   ⊞ -	→ A^ A` <u>◇</u> → <u>A</u> →	= = =		(2) Wrap Text	5	General	• • 0 .00	.00 →0
	C	lipboard		r <u>s</u>	Font	Гя		Alignn	ient	Гъ	Numbe	er	Б
A	1	~	: × ~	fx									
	4	4		В			С				D		
1													
2													
3	Row La	bels 斗	Feature	Name	Configuration						Device		Site
4													
5			BACL 1										
6					■ access-list 1 pe	ermit 10.0.0.0	0.255.255.25	5					
7											US-BOS-CW		
8													My N
9											UK-LHR-CW		
10													N.A N

6. To add Borders to the Pivot Table, select the whole table, go to **Home**, and click **Borders** to select **All Borders** from the dropdown menu.

🗴 AutoSave 💽 Off 📙 🏷 🖓 🤜	Network Feature_ACL Name List - Re	ead-Only ∽
File Home Insert Page Layout For	rmulas Data Review View A	utomate Help
Calibri Paste Copy → → S Format Painter	$\begin{array}{c c} \bullet & 11 & \bullet & A^{*} & A^{*} & \Xi & \Xi \\ \hline \hline \hline \hline \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \Xi & \Xi \\ \hline \hline \hline \hline \hline \bullet & \bullet$	
Clipboard 🛛 F	Borders	Alignment
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	Bottom Border	
A B	Top Border	-
1	Left Border	
3 Row Labels 4 Feature Name Configur	Right Border	
4 BACL 5 BACL 1	No Border	
6 BACEI	H All Borders	
7 8	Out <u>s</u> ide Borders	
9	Thick Outside Borders	
12 month market	Bottom Doyble Border	

To find ACL Golden Configuration, right-click the Feature Name and click Expand/Collapse > Collapse Entire Field.



3	Row Labels 斗	Feature Name	Configuration	Device	Site	Count of Device
4	■ACL					2234
5		ACL 1				98
6		<b>E ACL 10</b>				126
7		<b>E ACL 100</b>				20
8		<b>E ACL 101</b>				192
9		<b>ACL 111</b>				3
10		<b>E ACL 120</b>				1
11		<b>E ACL 130</b>				1
12		<b>E ACL 17</b>				378
13		<b>E ACL 190</b>				169
14		<b>E ACL 191</b>				1
15		<b>E ACL 192</b>				254
16		<b>E ACL 193</b>				128
17		🗄 ACL 194				128
18		<b>E ACL 195</b>				134
19		<b>E ACL 196</b>				24
20		<b>E ACL 198</b>				3
21		<b>E ACL 199</b>				9
22		ACL 2				4
23		<b>H</b> ACL 20				252
24		<b>ACL 25</b>				212
25		<b>E ACL 44</b>				3
26		ACL 5				85
27		• ACL 50				3
28		• ACL 51				3
29		ACL 52				3
31	Grand Total					2234

The following is the final Pivot Table with all ACL configurations:

8. Analyze each ACL Configuration.

The ACLs having the same configurations in many devices across multiple sites are a good candidate to be added to the Golden Template Table. For example, **ACL192** has the same configuration for devices in 8 sites. Therefore, the configuration of ACL 192 shown below can be used as the Golden Template for devices within these 8 sites.

Row Labels 斗	Feature Name	Configuration	Site	Device	Count of Device
BACL					2234
	H ACL 1				98
	<b>ACL 10</b>				126
	<b>HACL 100</b>				20
	<b>ACL 101</b>				192
	<b>HACL 111</b>				3
	<b>HACL 120</b>				1
	<b>ACL 130</b>				1
	<b>HACL 17</b>				378
	<b>ACL 190</b>				169
	🗄 ACL 191				1
	<b>ACL 192</b>				254
		access-list 192 permit tcp any			44
			B My Network\test-container-site-yihong\test-site-		12
					8
			My Network\EMEA\UK-LHR		8
			My Network\APAC\JP-TYO		4
			My Network\NA\US-NYJ		4
			My Network\APAC\SG-SIN		3
			My Network\EMEA\DE-MUC		3
			My Network\test-site-jun		2
		Line 100 - consisting and			

9. The ACL has many different types of configurations and each configuration is only configured in one or few devices is not a good candidate. For example, **ACL 25** has various types of configurations over different sites.

B ACL 25			212
	access-list 25 permit 134.47.169.0 0.0.0.255		1
		My Network\test-container-site-yihong\test-site-yihong	1
	access-list 25 permit 151.94.180.0 0.0.0.255		1
		Hy Network\test-container-site-yihong\test-site-yihong	1
	access-list 25 permit 150.57.176.0 0.0.0.255		1
		My Network\NA\US-BOS	1
	■access-list 25 deny 16.6.158.27		1
		My Network\NA\US-NYJ	1
	access-list 25 permit 134.27.168.0 0.0.0.255		1
		My Network\EMEA\UK-LHR	1
	■access-list 25 deny 167.10.192.47		1
		My Network\APAC\JP-TYO	1
	access-list 25 permit 144.38.170.0 0.0.0.255		1
		Hy Network\test-container-site-yihong\test-site-yihong	1
	■access-list 25 deny 167.11.191.48		1
		My Network\APAC\JP-TYO	1
	■ access-list 25 permit 151.104.180.0 0.0.0.255		1
	Raccess-list 25 denv 167.12.192.49		1

You can add the good candidates to the Golden Template Table manually and repeat the steps of 7.1 to check the configuration drif for these ACLs.

# 7.3 Check NTP Server Against Golden Template

In this session, you will check the **NTP Server** configurations against the golden template. You will go through the same step as Section 7.1 and resuse the ADTs of that section.

## 7.3.1 NTP Server Golden Template

Open the **Golden Template Table** you created in Section 7.1.1 and manually add a row for the NTP Golden template. The golden configuration can be:

### ntp server {\$\_dummy=10.10.0.101 | 10.10.0.102} prefer version 3

Here, we use a special variable, **\$\_dummy**, which has the value **10.10.0.101** | **10.10.0.102**, which means if a device configuration, *ntp server* **10.10.0.101** *prefer version* **3**, or *ntp server* **10.10.0.102** *prefer version* **3**, will match this golden configuration.

🐻 Gol	den Template Table	Table Builder         Las	t Updated at: N/A 🛛 🔍 Rebuild	d Table Add Data Manually 🗸 😑 🔒
Descri	ption: Type description here			
Items:	3 Rows 4 Columns		Search	Q Y Advanced Filter: Undefined C
No.	s Feature Name	s Golden Configuration	s Site	S Rep Device
1	acl_1	access-list 1 permit 192.168.0.0 (	My Network\NA\US-BOS	US-BOS-CW
2	acl_10	access-list 10 permit 8.8.8.8	My Network\NA\US-BOS	US-BOS-CW
3	NTP Server	ntp server {\$_dummy=10.10.0.1(	My Network\NA\US-BOS	US-BOS-CR01-01

## 7.3.2 Create Intent to Check NTP Server Against Golden Template

Create a new Network Intent, **Check NTP Server Against Golden Template** from the **Intent Manager.** Select the seed device, e.g., *US-BOS-CR01-01*. Add a **Config Diagnosis**, and use the function **LinesByKeyword** to parse the NTP configurations. Follow the same steps in Section 7.2.1.

### LinesByKeyword[\$ntp\_config]:ntp server

Parse Lines X	Image: Variables1       ✓       Z Type: Single ②       + New Pattern ✓       Image: Variables1         Var Line 1       LinesByKeyword[\$ntp_config]:ntp server       Image: Variables2       Image: Variables2         Var Line 1       LinesByKeyword[\$ntp_config]:ntp server       Image: Variables2       Image: Variables2         Variables1       Variables2       Image: Variables2       Image: Variables2       Image: Variables2         Variables2       Variables2       Image: Variables2       Image: Variables2       Image: Variables2         Variables2       Variables2       Image: Variables2       Image: Variables2       Image: Variables2         Variables2       Variables2       Variables2       Image: Variables2       Image: Variables2         Variables2       Varia
O The line of variable: Select O The line between: Select The line contains keyword: ntp server b Output: Output: Sntp_config=	+ Field
and_comge       ntp server 10.10.0.101 prefer version 3       ntp server 10.10.0.102 version 3   Pattern: LinesByKeyword[\$ntp_config]:ntp server Cancel Apply C	Output + Parse Lines 4 -

Repeat the steps of 7.1.2.4 to create the diagnosis, certainly with a different variable and feature name.

	2. Defir	ne Diagnosis						
۹ 🗸 م	📄 Add N	ote D Ado	l Diagnosis	Can	also click a varia	ble on the left to add	automatio	on.
Check NTP Server	Check NTP Server Anchor: \$ntp_config						~	•
	Type description of the diagnosis							
		Loop Table Rows						
	✓ If							
	Α	🥏 US-BOS-C	Current 🗸					
		ntp_config	~	MP(MP_ntp)	✓ Get_	Fable_Cell(Gol( ∨	Î	
	В	Select Variable	~		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		

📄 Diagnosis Me	issage:	🗌 Save to Incident 🔳
<b>~</b>	\$this_device: NTP Server config complies the golden config.	
Set Statu	s Code for Device:	
Success	\$this_device: NTP Server config complies the golden config.	
Set Statu	s Code for Intent:	
Success	\$this_device: NTP Server config complies the golden config.	
Else		🗑 Dele
<b>Else</b> Diagnosis Me	issage:	
📄 Diagnosis Me	essage: \$this_device: NTP Server config does not comply the golden config, Rule \$MP_ntp.Result, Missing Lines \$MP_ntp.Un	Save to Incident =
Diagnosis Me		Save to Incident =
~ (	* this_device: NTP Server config does not comply the golden config. Rule \$MP_ntp.Result, Missing Lines \$MP_ntp.Un s Code for Device:	Save to Incident =
Diagnosis Me  V-S Set Statu  Error	- s Code for Device:	Save to Incident =
Diagnosis Me  T Set Statu  Error	*         \$this_device: NTP Server config does not comply the golden config. Rule \$MP_ntp.Result, Missing Lines \$MP_ntp.Un         s Code for Device:         \$this_device: NTP Server config does not comply the golden config. Rule \$MP_ntp.Result, Missing Lines \$MP_         \$this_device: NTP Server config does not comply the golden config. Rule \$MP_ntp.Result, Missing Lines \$MP_	Save to Incident matched_lines
Diagnosis Me  V-S Set Statu  C-S Set Statu  C-S Set Statu	\$this_device: NTP Server config does not comply the golden config. Rule \$MP_ntp.Result, Missing Lines \$MP_ntp.Un s Code for Device: \$this_device: NTP Server config does not comply the golden config. Rule \$MP_ntp.Result, Missing Lines \$MP_ s Code for Intent:	Save to Incident matched_lines

ξ

# 7.3.3 Create the Summary Report and Dashboard

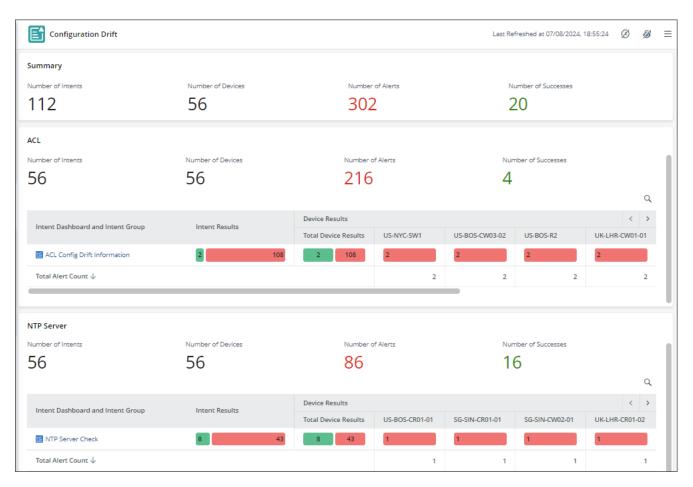
You can repeat the steps from Section 7.1.7 to create a summary report.

				Software	Version	s Site		NTP Server Config Check
				{ 15.4(2)T4		test-site-yihor	ng	Run Intents Once
				03.04.04.SG		test-site-xinyu	ı	Run Intents via Timer
w Summary Report - NTP Serv reate summary report of all the nly merge CSV reports generat	e CSV reports ge		ts in this	column: Create	53 of 53 intent	results filtered	×	Open Seed Intent Rebuild Intent-related Column Group Remove Empty Wrapper Intent Enable Auto Intent
NTP Server Config	]				,	· · · · · · · · · · · · · · · · · ·		Export Diagnosis Result to CSV View Summary Report
tems: 45 rows 8 columns					Sean	h	Q	Export Intent Output Map
Device	Site		Softwa	are Version	Golde	n Config		Debug Empty Cells
US-BOS-CF01-01/stby	My Network\	NA\US-BOS	9.5(2)	204	ntp se	rver {\$_dummy='	10.10.	Tag Current Column
US-BOS-CW03-02	My Network\	NA\US-BOS	15.2(2	0170809:19420	9) ntp se	rver {\$_dummy='	10.10.	Edit
US-BOS-CW04-02	My Network\	NA\US-BOS	15.2(2	0170809:19420	9) ntp se	rver {\$_dummy='	10.10.	Delete
US-BOS-SW1	My Network\	test-site-jun	15.2(H	II_20170202)FLC	_DSGS7			Set as Table Key
US-BOS-CW03-01	My Network\	NA\US-BOS	15.2(2	0170809:19420	9) ntp se	rver {\$_dummy='	10.10.	Submit Related Commands to Benchm
US-NYJ-CW02-02	My Network\	NA\US-NYJ	15.2(2	0170809:19420	9)			New Intent Dashboard
US-BOS-CW04-01	My Network\	NA\US-BOS	15.2(2	0170809:19420	9) ntp se	rver {\$_dummy='	10.10.	ᡔᢛᡟᠿ᠆ᡐᡣ᠋᠘ᢦ᠆᠆᠂ᡁᡐᠮ᠁ᡔ᠆᠆᠂ᢑ᠆ᡧ᠆᠂ᡔᠼᡘ᠆ᡐᠬᡐᡐ
US-NYJ-CW02-01	My Network\	NA\US-NYJ	15.2(2	0170809:19420	9)			
UK-LHR-CR01-01	My Network\	test-container-s.	15.4(2	)T4				
US-BOS-CW02-01	My Network\	NA\US-BOS	15.2(2	0170809:19420	9) ntp se	rver {\$_dummy='	10.10.	
			15 0/0				· · · · ·	

Create an Intent Dashboard from the ADT:

MTP Server Check					Last Refreshed	at 23/07/2024, 18:43:49	C #	Ø
Summary		23/07/20	24, 18:43:55 View Report	Device Information		23/07/2024, 18:4	3:55 View	w Repo
		nts es Executed nt-level Alerts		Cisco IOS Switch	Cisco Router	56	Devices	
Intent Result History						23/07/2024, 18:4	3:55 Viev	w Repi
Time Range: All 🛩 Result: All 🛩								
0		Sum of In	ent Alert Status Code Count	<ul> <li>Sum of Intent Success Status Code Count</li> </ul>				
						10 10 10 10 10 10 10 10 10	20120120 20120120	23/07/2
			Top Five In	tent Alerts				
Intent Name	Map	Execution Time	Intent Alert Status Code	Count Intent Success Status Code Count	Intent Status Code Summary	Intent Alert Detection		
Check NTP Server Against Golden Templa	View Map	23/07/2024, 18:23:19	1	0	SG-SIN-CW01-01: NTP Server config does	. 1		
Check NTP Server Against Golden Templa	View Map	23/07/2024, 18:23:19	1	0	DE-MUC-CW01-01: NTP Server config does.	1		
Check NTP Server Against Golden Templa	View Map	23/07/2024, 18:23:19	1	0	US-NYJ-CR01-02: NTP Server config does n.	. 1		
Check NTP Server Against Golden Templa	View Map	23/07/2024, 18:23:19	1	0	UK-LHR-CW03-01: NTP Server config does .	. 1		

Add the dashboard you just created to the existing Summary Dashboard, **Configuration Drift**, created in Section 7.1.7:



# 7.4 Check Configuration Settings in Public Cloud

Intent-based automation can also be used to check the configuration drift against the standard configurations or best practices in the public cloud. For this use case, you will retrieve the data using **API** and compare this configuration data with the recommended one.

You can follow the same flow of Section 7.1. The difference is that you retrieve the data via API instead of the CLI. The data returned by API is usually structured data that can be easily saved as a **Json table**.

We will use the AWS EC2 configuration as an example.

An example of the final ADT is like:

쁍 AV	VS EC2 Configuration Settir Tabl	e Builder Last Updated at: 07/25/2024 05:22 PM 🔍 Rebuild Table	Add Data Manually 🗸 📃 🖬
Descr	iption: Type description here		
Items	: 118 Rows 3 Columns		Search Q TAdvanced Filter: Undefined
No.	Device	Check EC2 Config	s Intent Status Code
1	&2*`\n/&(i-094cb7dd86455412c)	Check AWS EC2 Configuration Against Baseline &2_`_n_&(i-094cb7 •	On this VM &2*`\n/&(i-094cb7dd86455412c) instance type, availa
2	(i-00aefbb9ecd7b69de)	Check AWS EC2 Configuration Against Baseline (i-00aefbb9ecd7b69 💿	On this VM (i-00aefbb9ecd7b69de) instance type, availability zone
3	(i-03a18e36322f36e25)	Check AWS EC2 Configuration Against Baseline (i-03a18e36322f36e 🧿	On this VM (i-03a18e36322f36e25) instance type, availability zone,
4	(i-04b6d5d8acdd36c9d)	Check AWS EC2 Configuration Against Baseline (i-04b6d5d8acdd36 🧿	On this VM (i-04b6d5d8acdd36c9d) instance type, availability zon
5	(i-05363037671558dd8)	Check AWS EC2 Configuration Against Baseline (i-05363037671558 🧿	On this VM (i-05363037671558dd8) instance type, availability zon
6	(i-0bdd3caa55517ba3c)	Check AWS EC2 Configuration Against Baseline (i-0bdd3caa55517b 🧿	On this VM (i-0bdd3caa55517ba3c) instance type, availability zone
7	(i-0bf20d0e607b910e9)	Check AWS EC2 Configuration Against Baseline (i-0bf20d0e607b91 🤨	On this VM (i-0bf20d0e607b910e9) instance type, availability zone
8	5325-to-7925-sharing-Subnet-ec2	Check AWS EC2 Configuration Against Baseline 5325-to-7925-shari 💿	On this VM 5325-to-7925-sharing-Subnet-ec2-1(i-02ce07443072f33
9	5325-to-7925-sharing-TGW-ec2-1(i	Check AWS EC2 Configuration Against Baseline 5325-to-7925-shari 🧿	On this VM 5325-to-7925-sharing-TGW-ec2-1(i-06c23d9c39f1b129a
10	5325To7925-VPC2-instance(i-0c985	Check AWS EC2 Configuration Against Baseline 5325To7925-VPC2-i 🧿	On this VM 5325To7925-VPC2-instance(i-0c985b246b1801c1e) ins
11	7925-vpc1-ec2(i-06bbadaec14656c4	Check AWS EC2 Configuration Against Baseline 7925-vpc1-ec2(i-06 🧿	On this VM 7925-vpc1-ec2(i-06bbadaec14656c48) instance type, a
12	ASAV10(i-0fa494c7d572e5478)	Check AWS EC2 Configuration Against Baseline ASAV10(i-0fa494c7d 💿	On this VM ASAV10(i-0fa494c7d572e5478) instance type, availabili
13	ASAv10-NACL-appliance-ACL-Test-n	Check AWS EC2 Configuration Against Baseline ASAv10-NACL-appli 🧿	On this VM ASAv10-NACL-appliance-ACL-Test-notcommonlyused(i
14	ASAv10-duplicateip-ec2-103.6(i-0b2	Check AWS EC2 Configuration Against Baseline ASAv10-duplicateip 💿	On this VM ASAv10-duplicateip-ec2-103.6(i-0b264b6506edaae5b) i
15	AWS-Firewall-Lab-Instance-01(i-02b	Check AWS EC2 Configuration Against Baseline AWS-Firewall-Lab-In 🧿	On this VM AWS-Firewall-Lab-Instance-01(i-02bd82556acef3e05) i

## 7.4.1 Prerequisites

To prepare for this exercise, create a Device Group for all the **AWS EC2 Instances** by the dynamic criteria: **Device Type** matches any **AWS EC2 Instance**.

	ncet3rain 🗖	ext-Gen Search Anything a	and Create Map	Dynamic Search Device			×
Ð	vew Map	Rew Device Group	New Data View Template	Search Scope: All Devices Device Criteria:	~		
Recents	Device Group P	roperties	<b>}</b>	A Device Type	✓ Matches any	→ AWS EC2 Instance →	5 ~ 🛍
000 Network	Name:	AWS EC2 Devices		B Select Criteria	~		
Files	Description:	Input description here		Boolean Expression: A			
Site	N Location:	My Device Groups	~				Search 6
	Devices and In	nterfaces		Search Result:			
	+ Static ~	+ Dynamic Search $\vee$	+ Exclude ~	Hostname &2*`\n/&(i-094cb7dd8645541	Vendor Amazon	Model EC2 Instance	Management IP
	Hostna	Dynamic Search Device	Model	(i-00aefbb9ecd7b69de)	Amazon	EC2 Instance	172.31.11.208
		Dynamic Search Interface		(i-03a18e36322f36e25)	Amazon	EC2 Instance	172.26.5.5
				(i-04b6d5d8acdd36c9d)	Amazon	EC2 Instance	172.26.4.11
	L	w	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	(i-05363037671558dd8)	Amazon	EC2 Instance	10.30.1.192
							Cancel OK 7

Also, you will need a script to retrieve the data from the AWS. NetBrain Automation Library provides these scripts. You can ask the Netbrain support team if your request is not covered in the library. For now, you can use the following script:

	Begin Declare Input Parameters
	[
	]
	End Declare
	For sample
	[
	{"name": "\$param!"},
	{"name": "\$param2"}
	]
	III
	import json
	import datetime
26	2   NetBrain R11.1b

#### import ast

def BuildParameters(context, device\_name, params):

node\_props = GetDeviceProperties(context, device\_name, {'techName': 'Amazon AWS', 'paramType': 'SDN', 'params' : ['nbPathValue', 'RegionName'] })

```
dn = node_props['params']['nbPathValue'].rpartition('/ec2/')[2]
```

dev\_id = node\_props['params']['nbPathValue'].split('/')[4]

```
apiServerld = node_props('params')['apiServerld']
```

```
RegionName = node_props['params']['RegionName']
```

rtn\_params = [{ 'devName' : device\_name, 'dn': dn, 'apiServerld': apiServerld, 'RegionName': RegionName ,'dev\_id}]

return rtn\_params

def RetrieveData(rtn\_params):

if isinstance(rtn\_params, str):

rtn\_params = json.loads(rtn\_params)

```
param = rtn_params
```

dn = rtn\_params('dn') if 'dn' in rtn\_params else "

```
rtn_res = []
```

param('region\_name') = rtn\_params('RegionName')

```
param['resource_type'] = 'ec2'
```

param('func\_name') = 'describe\_instances'

```
param['field_name'] = 'Reservations'
```

```
param('func_param') = {'InstanceIds': [param['dev_id']]}
```

#raise NameError(param)

res = get\_aws\_resource\_data(param)

data = json.loads(res)

return json.dumps(data[0]['Instances'], indent=4)

# 7.4.2 Create Intent to Check AWS EC2 Configuration

Create a Network Intent, **Check AWS EC2 Configuration** from the **Intent Manager**. Select an AWS EC2 instance as the seed device, e.g., **7925-vpc1-ec2(i-06bbadaec14656c48)**.

Network Intent (Edit Mode)							×		
Check AWS EC2 Configuration Against Bainst Bains	aseline 📶	Diagnosis Tree	Run with Live D	ata		Save	😮 Help 📃		
Type description here						💦 Intent M	ap: Select 🗸		
Seed Logic     Replication Lo	gic 🔔								
Pevice 2			Intent Varia	bles:	Manage	r   Ta	g: + Add 🥅		
Select Devices									×
Select Devices by:  O  Device Type  I	Device Group	<ul> <li>Site</li> </ul>				1 Devices S	Selected		
All Device Types	✓ 7925			$\times$		Ø 7925	-vpc1-ec2(i-06b	badaec14656c.	•
Hostname	Mgmt IP	Vendor	Model						
5325-to-7925-sharing-Subnet-ec2-1(	10.55.2.100	Amazon	EC2 Instance	*					
5325-to-7925-sharing-TGW-ec2-1(i-0	172.31.3.38	Amazon	EC2 Instance						
5325To7925-VPC2-instance(i-0c985b	172.31.3.87	Amazon	EC2 Instance						
<b>7925</b> -vpc1-ec2(i-06bbadaec14656c48)	172.31.36.178	Amazon	EC2 Instance		>				
X 7925CA-to-5325NV(vgw-05f0992976		Amazon	AWS Virtual Private Gat		>>				
7925VPC1(vpc-9d91ebe7)		Amazon	VPC Router						
7925VPC2(vpc-0b4a89e5304d086fa)		Amazon	VPC Router		<				
EC2CreatedInSharedSubnetBy0701	172.29.0.158	Amazon	EC2 Instance		~~				
diff-region-vpc-peer-7925-do-not-de	172.32.1.6	Amazon	EC2 Instance						
unattached-eni- <mark>7925</mark> -to-5325(eni-0f		Amazon	AWS Unattached Netwo	•					
vpc-peer-diff-region-7925-main(vpc		Amazon	VPC Router						
									-
				*		4			►
							Ca	ancel OK	

## 7.4.2.1 Add API Diagnosis

In this section, you will retrieve the EC2 configuration settings via the API and parse the data:

- 1. Click 🗏 menu and open **Add API Diagnosis** from dropdown.
- 2. In the **API Diagnosis** window, click **Function** to define how to retrieve data. Copy and paste the script in the prerequisites section.
- 3. Click **OK** to save the Function.
- 4. Click **Retrieve** to retrieve API sample data with **Live Data** source.

Network Intent (Edit Mode)			×
I Check AWS EC2 Configuration 🔒 Diagnosis Tree	Run	with Live Data	Save 🕜 Help 😑
Type description here			💑 Intent Map: Select 🗸 🗸
I Seed Logic			
		Intent Variables: Ma	nager   Tag: + Add 🔲
	👩 + Add (		
<pre></pre>	//th Live Da	Add SNM Add API D Add Ping Add Trace Add Devia Add Devia	P Diagnosis Jagnosis Diagnosis Diagnosis te Diagnosis te Diagnosis Block n >
<pre>apiServerId = node_props['params']['apiServerId'] RegionName = node_props['params']['RegionName'] rtn_params = [{ 'deWidem' : device_name, 'dn': dn, 'apiServerId': apiServerId, 'RegionName'; RegionName , 'dev_id':dev_id return rtn_params ' fi fisststance(rtn_params); ' rtn_params = [sion.loads(rtn_params) param['region_name'] = rtn_params[RegionName'] param['resurce_type'] = 'k2: ' param['field_name'] = 'Reservations' param['field_name'] = 'Inservations' d dta = json.loads(res) return json.dumps(data[0]['Instances'], indent=4) </pre>	•	3	

5. In the the **+ New Pattern** dropdown, click **Advanced** > **Json Table**. The data returned from the AWS API is the Json formatted.

7925-vpc1-ec2(i-	06bbadaec14656c48) Section Name: API	Section 者 🛛 API Ada	pter: Amazon AWS	~	Function Retrieve ~	with Live Data
Define Variable				2. Define (		
Format1	× +					
Double-click a va	riable to parse. Select multiple lines to parse a ta	able.	Critical Variable (0)		n v + New	Pattern 🛩
Current Device	✓ 07/25/2024 12:14:16 PM	Search	Q 🔺 🗸		Auto Pattern	
1					Single Variable	
2 {					Table	
3	"AmiLaunchIndex": 0,				lable	
4	"ImageId": "ami-0947d2ba12ee1ff75",				Paragraph	
5	"InstanceId": "i-06bbadaec14656c48",				Advanced	> Collector
6	"InstanceType": "t2.micro",				Advanced	
7	"KeyName": "7925keypair", "LaunchTime": "2020-11-03 20:50:28+0					Json Table 🗲 5
8	"Launchlime": "2020-11-03 20:50:28+0 "Monitoring": {	0:00 ,				
10	"State": "disabled"					
11	}	API Sample D	ata			
12	"Placement": {					
13	"AvailabilityZone": "us-east-1a"					
14	"GroupName": "",					
15	"Tenancy": "default"					

6. Define Data Scope.

In the By Path dropdown, select [] (Square Brackets) and then {} (Curly Brackets).

The By Path value will be: []/{}

JsonTable	1 V Z Type: JsonTable 👔 + New Pattern V	
Data Scop	e 🔊	
By Path	0/0 6	~
Define Va	filter	٩
JSON Key	<b>4</b> []	<b>^</b>
$(\pm)$	<b>▲</b> ⊕	
	Amilauhchindex	
	ImageId	
	InstanceId	
Output	InstanceType	

NOTE: You can define some important keys as shown below. In this example, only a few keys are selected, but you can choose the keys according to your specific use case.

Define the Variables by Selecting **JSON Key**.

7. To add the JSON Key, click 🖽 to add the *InstanceType* JSON Key from the dropdown.

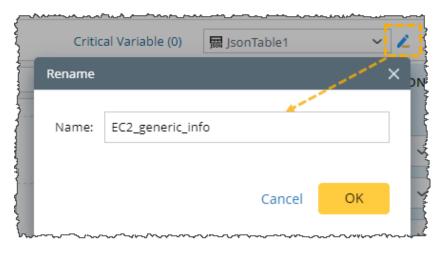
Click 🖽 and add AvailabilityZone, PrivateDnsName, PrivateIpAddress, PublicDnsName, PublicIpAddress, SecurityGroups[], and PlatformDetails.

a JsonTable1	~ ∠ ту	pe: Js	onTable ? 🛛 + N	lew Pattern ∨					
Define Variable by	Selecting JSON K	ey 🕐	0 7						
JSON Key			Variable		Ту	vpe			
InstanceType	~	⇔	\$instancetype		5	tring	~	Ē	
AvailabilityZone	~	⇔	\$availabilityzone		\$	string	~	Ŵ	
PrivateDnsName	~	⇔	\$privatednsname	2	\$	string	~	Ē	
PrivatelpAddress	~	⇔	\$privateipaddres	s	2	tring	~	Ŵ	
PublicDnsName	~	⇔	\$publicdnsname		2	string	~	Ŵ	
PublicIpAddress	~	⇔	\$publicipaddress		5	string	~	Ŵ	
SecurityGroups[]	~	⇔	\$securitygroups		\$	string	~	Ē	
PlatformDetails	~	⇔	\$platformdetails		5	tring	~	Ŵ	
$( \cdot )$									
Output			<u> </u>						_
\$instancetype 🗸	\$availabilityz 🕚	~	\$privatednsn 🗸	\$privateipad	~	\$publicdn	sna	~	\$
t2.micro	us-east-1a		ip-172-31-36-17	172.31.36.178		ec2-54-22	4-186		5

8. In the Output section, set the **\$instancetyps** as the table key.

2	Define Variable by	y Selecting JSON Key	0	$\sim$	~~~~~	~~~~	
Į.	JSON Key	٨	/ariable	Туре			
}	InstanceType	~ 🗢	\$instancetype	string	~	Û	
<u>ک</u>	AvailabilityZone	~ ₽	\$availabilityzone	string	~	Û	
	Output	<u>ح</u> (	<b>^</b>			•	-
	\$instancetype 🍟	\$availabilityz 🗸	\$privatednsn 🗸	\$privateipad 🗸	\$pub	licdns	na
ş <u></u>	Interface Key Table Key	us-east-1a	ip-172-31-36-17	172.31.36.178	ec2-5	4-224	-186

9. Click the **pen** icon to rename **JsonTable1** to **EC2\_Generic\_Info**.



### The whole **API Diagnosis** window will be:

ine Variable		2. Define Diagnosis	
Format1 V +		Test on	Devices:
	Critical Variable	🗮 EC2_generic_info 🗸 Ype: JsonTable 👔 + New Pattern 🗸	
	arch 🔍 🔺 🔻	Data Scope 🕖	
1  r 2  { 3 "AmilaunchIndex": 0.	DS1-Start	By Path D/O	~
3 "AmiLaunchIndex": 0, 4 "ImageId": "ami-09472bal2eelff75", 5 "InstanceId": "i-06bbadaec14656c48", 6 "InstanceType": "t2.micro",	Key 1	Define Variable by Selecting JSON Key 💿	
<pre>7 "KeyName": "7925keypair", 8 "LaunchTime": "2020-11-03 20:50:28+00:00",</pre>		JSON Key Variable Type	
9 "Monitoring": { 0 "State": "disabled"		InstanceType v 🕫 Sinstancetype string	~ 🕯
1 }, 2 "Placement": { 3 "AvailabilityZone": "us-east-la",		AvailabilityZone V SavailabilityZone string	~ 🕯
4 "GroupName": "", 5 "Tenancy": "default"		PrivateDnsName v 🗘 Sprivatednsname string	~ 🕯
6 }, 7 "PrivateDnsName": "ip-172-31-36-178.ec2.internal",	Кеу 3	PrivatelpAddress v 🗢 Sprivateipaddress string	~ 1
8 "PrivateIpAddress": "172.31.36.178", 9 "ProductCodes": [], 0 "PublicDnsName": "ec2-54-224-186-124.compute-1.amazonaws.	Key 4	PublicDnsName v 🗘 Spublicdnsname string	~ 🕯
1 "PublicIpAddress": "54.224.186.124", 2 "State": {	Key 6	PubliclpAddress v 🗢 Spublicipaddress string	~ 🕯
3 "Code": 16, 4 "Name": "running" 5 }.		SecurityGroups[] V Securitygroups string	~ 💼
<pre>5 }, 6 "StateTransitionReason": "", 7 "SubnetId": "subnet-30416b6c",</pre>		PlatformDetails V Splatformdetails string	~ 🕯
8 "VpcId": "vpc-9d91ebe7", 9 "Architecture": "x86_64",		$ \bigcirc $	
<pre>0 "BlockDeviceMappings": [ 1 { 2 "DeviceName": "/dev/xvda",</pre>		Output	
2 "Devicename : "/dev/xvda", 3 "Ebs": { 4 "AttachTime": "2020-11-03 20:50:29+00:00",		Sinstance 🖗 🗸 Savailabilityz 🖍 Sprivatednsn 👻 Sprivateipad 🗡 Spublicdnsna 🗸	\$publicij
5 "DeleteOnTermination": true, 6 "Status": "attached",		t2.micro us-east-1a ip-172-31-36-17 172.31.36.178 ec2-54-224-186	54.224.1
7 "VolumeTd" • "vol-06cf250029498ed98" 8 4	•	4	

### 7.4.2.2 Define Diagnosis

Defining the diagnosis from the API data is the same as the CLI or Config Diagnosis:

- 1. Go to the **Define Diagnosis** section to define the diagnosis logic.
- 2. Click **Add Diagnosis** to define conditions and Intent output message.
- 3. Enter the diagnosis name, e.g., **Recourse Change Attributes.**
- 4. Tick the **Loop Table Rows** checkbox and select the Table Variable (*EC2\_generic\_info*) and Table Key (*instancetype*).

Add N	ote D Add Diagnosis	Can also click a variable on the left to add automa
Name:	Resource Change Attributes	Anchor: 🗸 🗸
	Type description of the diagnosis	

5. Define the **If** condition for each **JSON Key Variable** as detailed in the image. For all the Variables, compare the **Current** configuration with the **Baseline** configuration.

The Boolean Expression is **A or B or C or D or E or F or G or H.** 

Add Note D Add Diagn				also click a variable on		
Loop Table Rows 🔜 EC2_	generic_info ∨	Table Key: Plea	ise Selec	t 🗸 🤨		0
A 👩 7925-vpc1-ec2(i	Current 🗸				Baseline 🗸	
a instancetype	~	Does not equal	~	instancetype	~	面
B 👩 7925-vpc1-ec2(i	Current 🗸				Baseline 🗸	
b availabilityzone	~	Does not equal	~	availabilityzone	~	Ē
C 7925-vpc1-ec2(i	Current 🗸				Baseline 🗸	
c privatednsname	~	Does not equal	~	privatednsname	~	Î
D 07925-vpc1-ec2(i	Current 🗸				Baseline 🗸	
d privateipaddress	~	Does not equal	~	privateipaddress	~	Ē
E 0 7925-vpc1-ec2(i	Current 🗸				Baseline 🗸	
e publicdnsname	~	Does not equal	~	publicdnsname	~	Ŵ
F 0 7925-vpc1-ec2(i	Current 🗸				Baseline 🗸	
f publicipaddress	~	Does not equal	~	publicipaddress	~	Ē
G 👩 7925-vpc1-ec2(i	Current 🗸				Baseline 🗸	
g securitygroups	~	Does not equal	~	securitygroups	~	Ē
H 👩 7925-vpc1-ec2(i	Current 🗸				Baseline 🗸	
h platformdetails	~	Does not equal	~	platformdetails	~	Î
I Select Variable	~					
Regione Supervision	d R and C and					3
Boolean Expression: A an	u b and C and	D and E and F and G a				i
Add Elself + Add Else						

6. Define Intent Output message.

Enter a message under the **Then** and **Else** output areas to appear as the result of the diagnosis.

	lessage	e: Save to Incident =
~	On th	his VM <code>\$this_device</code> instance type, availability zone, private dns name, private in address, public dns name, public in address, secu
🔽 💿 Set Sta	tus Coo	de for Device:
Error	~	On this VM \$this_device instance type, availability zone, private dns name, private ip address, public dns name, public ip address
Set Sta	tus Coc	de for Intent:
Error	~	On this VM \$this_device instance type, availability zone, private dns name, private in address, public dns name, public in addres
Diagnosis N	/lessage	
Else		Delet
	_	
~	one, pr	rivate dns name, private ip address, public dns name, public ip address, security groups, and platform details have not changed.
🔽 💿 Set Sta	tus Coo	de for Device:
Succe	55 🗸	On this VM \$this_device instance type, availability zone, private dns name, private ip address, public dns name, public ip addre
🔽 💿 Set Sta	tus Coo	de for Intent:
Succe	SS 🗸	On this VM \$this_device instance type, availability zone, private dns name, private ip address, public dns name, public ip addre
\dd Logic∨		

**Then**: define the Red color, message and status in case **If** condition is satisfied as follows:

On this VM **\$this\_device** instance type, availability zone, private dns name, private ip address, public dns name, public ip address, security groups, and platform details.

Current instance type is \$instancetype, baseline instance type was \$instancetype(Baseline). Current availability zone is \$availabilityzone, baseline availability zone was \$availabilityzone(Baseline). Current private DNS is \$privatednsname, baseline DNS was \$privatednsname(Baseline). Current private IP is \$privateipaddress, baseline private IP was \$privateipaddress(Baseline). Current public DNS is \$publicdnsname, baseline public DNS was \$publicdnsname(Baseline). Current public IP is \$publicipaddress, baseline public IP was \$publicipaddress(Baseline). Current public IP is \$publicipaddress, baseline public IP was \$publicipaddress(Baseline). Current security groups are \$securitygroups, baseline security group were \$securitygroups(Baseline). Current platform details are \$platformdetails, baseline platform details were \$platformdetails(Baseline)

v Then			~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Then     Diagnosis Message:			Save to Incident
Diagnosis Message of Intent Diagnosis		×	Pop up Delete
Diagnosis Message:	+ 🐹 Variabl	le + 💽 Expression	
On this VM <b>\$this_device</b> instance type, availability zone Current instance type is <b>\$instancetype</b> , baseline instan Current availability zone is <b>\$availabilityzone</b> baseline a	ce type was \$instancetype(Ba wailability zone was \$availabil	seline).	
Define Link to Variable	× ×	_	
∡ m EC2_generic_info	Ca	orcel OK	
I instancetype (7925-vpc1-ec2(i-06bbadaec14656c4	8).EC2_generic_info)		
📗 availabilityzone (7925-vpc1-ec2(i-06bbadaec14656	c48).EC2_generic_info)		
📗 privatednsname (7925-vpc1-ec2(i-06bbadaec1465	6c48).EC2_generic_info		
📔 privateipaddress (7925-vpc1-ec2(i-06bbadaec1465	6c48).EC2_generic_infc		
📗 publicdnsname (7925-vpc1-ec2(i-06bbadaec14656	c48).EC2_generic_info)		
📔 publicipaddress (7925-vpc1-ec2(i-06bbadaec14656	oc48).EC2_generic_info)		
📔 securitygroups (7925-vpc1-ec2(i-06bbadaec146566	:48).EC2_generic_info)		
📔 platformdetails (7925-vpc1-ec2(i-06bbadaec14656	c48).EC2_generic_info)		
x this_device			
4	•		
Source: Baseline ∨			
Link Text: \$instancetype(Baseline)			
	Cancel OK		

### **Else**: enter a message indicating no change:

On this VM \$this\_device instance type, availability zone, private dns name, private ip address, public dns name, public ip address, security groups, and platform details have not changed.

## 7.4.3 Replicate Intent to All AWS EC2

You can follow the same flow of Section 7.1 to replicate the intent to all AWS EC2 instances. Select the device group you created in the Section 7.4.1.

After running once all replicate intent, rebuild the table to see the intent status codes in the ADT table:

	🐻 AWS EC2 Configuration Settings Lost Updated at: 07/25/2024 05:22 PM 🔍 Rebuild Table Add Data Manually 🗸 🚍 💼									
Descri	Description: Type description here									
items:	: 118 Rows 3 Columns				Search Q Y Advanced Filter: Undefined C					
No.	Device	≡	Check EC2 Config	5 Intent	t Status Code 📃					
1	&2*`\n/&(i-094cb7dd86455412c)	≡	Check AWS EC2 Configuration Against Baseline &2_`_n_&(i-094cb7 •	On this V	/M &2*`\n/&(i-094cb7dd86455412c) instance type, availa					
2	(i-00aefbb9ecd7b69de)		Check AWS EC2 Configuration Against Baseline (i-00aefbb9ecd7b69 💿	On this V	/M (i-00aefbb9ecd7b69de) instance type, availability zone					
3	(i-03a18e36322f36e25)		Check AWS EC2 Configuration Against Baseline (i-03a18e36322f36e 💿	On this V	/M (i-03a18e36322f36e25) instance type, availability zone,					
4	(i-04b6d5d8acdd36c9d)		Check AWS EC2 Configuration Against Baseline (i-04b6d5d8acdd36 💿	On this V	/M (i-04b6d5d8acdd36c9d) instance type, availability zon					
5	(i-05363037671558dd8)		Check AWS EC2 Configuration Against Baseline (i-05363037671558 🧿	On this V	/M (i-05363037671558dd8) instance type, availability zon					
6	(i-0bdd3caa55517ba3c)		Check AWS EC2 Configuration Against Baseline (i-0bdd3caa55517b 💿	On this V	/M (i-0bdd3caa55517ba3c) instance type, availability zone					
7	(i-0bf20d0e607b910e9)		Check AWS EC2 Configuration Against Baseline (i-0bf20d0e607b91 •	On this V	/M (i-0bf20d0e607b910e9) instance type, availability zone					
8	5325-to-7925-sharing-Subnet-ec2-1(i-02ce07443072f3320)		Check AWS EC2 Configuration Against Baseline 5325-to-7925-shari 💿	On this V	/M 5325-to-7925-sharing-Subnet-ec2-1(i-02ce07443072f33					
9	5325-to-7925-sharing-TGW-ec2-1(i-06c23d9c39f1b129a)		Check AWS EC2 Configuration Against Baseline 5325-to-7925-shari 💿	On this V	/M 5325-to-7925-sharing-TGW-ec2-1(i-06c23d9c39f1b129a					
10	5325To7925-VPC2-instance(i-0c985b246b1801c1e)		Check AWS EC2 Configuration Against Baseline 5325To7925-VPC2-i •	On this V	/M 5325To7925-VPC2-instance(i-0c985b246b1801c1e) ins					
11	7925-vpc1-ec2(i-06bbadaec14656c48)		Check AWS EC2 Configuration Against Baseline 7925-vpc1-ec2(i-06 •	On this V	/M 7925-vpc1-ec2(i-06bbadaec14656c48) instance type, a					
12	ASAV10(i-0fa494c7d572e5478)		Check AWS EC2 Configuration Against Baseline ASAV10(i-0fa494c7d 💿	On this V	/M ASAV10(i-0fa494c7d572e5478) instance type, availabili					
13	ASAv10-NACL-appliance-ACL-Test-notcommonlyused(i-0eaa22b7bf3de		Check AWS EC2 Configuration Against Baseline ASAv10-NACL-appli 💿	On this V	/M ASAv10-NACL-appliance-ACL-Test-notcommonlyused(i					
14	ASAv10-duplicateip-ec2-103.6(i-0b264b6506edaae5b)		Check AWS EC2 Configuration Against Baseline ASAv10-duplicateip 👁	On this V	/M ASAv10-duplicateip-ec2-103.6(i-0b264b6506edaae5b) i					
15	AWS-Firewall-Lab-Instance-01(i-02bd82556acef3e05)		Check AWS EC2 Configuration Against Baseline AWS-Firewall-Lab-In 💿	On this V	/M AWS-Firewall-Lab-Instance-01(i-02bd82556acef3e05) I					
16	AWS-Firewall-Lab-Instance-02(i-009aeb5c93aface06)		Check AWS EC2 Configuration Against Baseline AWS-Firewall-Lab-In 💿	On this V	/M AWS-Firewall-Lab-Instance-02(i-009aeb5c93aface06) in					
17	AWS-Firewall-NAT-Lab-Instance(i-04527181641150e4b)		Check AWS EC2 Configuration Against Baseline AWS-Firewall-NAT-L 💿	On this V	/M AWS-Firewall-NAT-Lab-Instance(i-04527181641150e4b)					
18	Benchmark-Added-EC2(I-0167fe60a98a5a95a)		Check AWS EC2 Configuration Against Baseline Benchmark-Added 💿	On this V	(M.Benchmark-Added-EC2(I-0167fe60a98a5a95a) Instanc					

### You can also create a Dashboard to view the results:

AWS EC2 Config Drift Analysis					Last Refr	reshed at 26/07/2024, 13:43:20 🕻 🖇
Summary		26/07	/2024, 13:43:24 View Report	Device Information		26/07/2024, 13:43:24 View R
	368 Tir	ents nes Executed ent-level Alerts		• AWS EC	C2 Instance	112 Devices
ntent Result History ime Range: All 🗸 Result: All 🗸						26/07/2024, 13:43:24 View R
		- Sum of	Intent Alert Status Code Count	← Sum of Intent Success Status Code Count		
. <u></u>				<u> </u>		
01/10/10/10/10/10/10/10/10/10/10/10/10/1	10,10,10,10,10,10,10,10,10,10,10,10,10,1		1, 10, 10, 10, 10, 10, 10, 10, 10, 10, 1			
			Top Five	ntent Alerts		
ntent Name	Map	Execution Time	Intent Alert Status Co	de Count Intent Success Status Code Cou	unt Intent Status Code Summary	Intent Alert Detection
heck AWS EC2 Configuration Against Ba	View Map	25/07/2024, 17:25:46	0	0		0
heck AWS EC2 Configuration Against Ba	View Map	25/07/2024, 17:25:46	0	0		0
heck AWS EC2 Configuration Against Ba	View Map	25/07/2024, 17:25:46	0	0		0
	View Map		0	0		0

# 8 Network Assessment Case Study: Failover

Failover and redundancy are one of the important aspects of your network. Network outages and performance degrading often happen when the failover occurs. In this chapter, you will create the intent to check the Failover status (whether the failover occurred ) and assess the performance degradation, which usually occurs after the failover. Then, we are going to create automation to check the configuration consistency between the failover pair.

- 1. <u>Check Failover Status Change</u>
- 2. Follow-up to Check the Performance Degrade After Failover
- 3. Check Failover Consistency of Pair Devices

## 8.1 Check Failover Status Change

In this section, let us begin with creating a group of HSRP devices and then proceed with intent creation and dashboard as follows:

- 1. <u>Create a group for HSRP devices</u>.
- 2. <u>Create an intent to check the failover status change</u>.
- 3. <u>Replicate the intent to the HSRP device group using ADT</u>.
- 4. Create dashboard.

## 8.1.1 Create HSRP Device Group

First, you need to create a device group, **HSRP Devices**, to include all HRSP devices with the dynamic criteria, **Config File contains standby**, and **Vendor** contains **Cisco**.

The device group includes all Cisco HSRP devices:

Dyna	mic Search Device				×
	rch Scope: All Devices	~			
А	Config File	∽ Contains	standby 2		圓
В	Vendor 3	✓ Contains	✓ cisco ◀		圓
с	Select Criteria	~			
Вос	blean Expression: A and B	5			
				Searc	h 6
				Search	
Sea	rch Result:				
н	lostname	Vendor	Model	Management IP	
B	J-L2-Core-A	Cisco	catalyst356024TS	172.26.3.10	
B	J-L2-coreB	Cisco	WS-C3560-24TS	172.24.101.3	
B	J_L2_Core_3	Cisco	WS-C3750-24TS	172.26.3.20	
B	J_L2_Core_4	Cisco	WS-C3750-24TS	172.24.101.5	
B	J_core_3550	Cisco	WS-C3550-24	172.24.36.1	
				Cancel	ок 7

## 8.1.2 Create an intent to check the failover status change

- 1. <u>Select a device</u>
- 2. <u>Create a Paragraph Parser</u>
- 3. Define the diagnosis to check the status

### 8.1.2.1 Select a device

- 1. From the End User Desktop, click into the **Search Bar**, type **US-BOS-SW1**, and press **Enter**.
- 2. In the search results, click **Map** beside **US-BOS- SW1**.
- 3. Close the search results.



- 4. In the upper-left corner of the map, click **Intent** to expand the Intent pane. Then click the **Quick Intent** tab header.
- 5. In the *Input Command....* field, enter **show standby.**
- 6. Click Retrieve.
- 7. Click **New** located next to the **Select Parser** dropdown. A Visual parser window will appear on the screen to configure the variables.

アン	Map12 (Master) ✓ ★ > Page 1✓							
~	Runbook D Data View							
Auto Intent Quick Intent 4 ublished Intents Map Intent								
Colle	ect Data US	S-BOS-SW1 🗸	,			~~ {		
sho	w standby	5			×v	Retrieve 6		
05:1	05:11:19 PM: Successfully retrieved!							
2	US-BOS-SW1 🔻	́Р	arser:	Select parser	~	New 7		
					Ado	l to Intent >>		
07/2	23/2024 05:11:	14 PM (Live)	aja.			L. L		
Vla S	JS-BOS-SW1>show standby Vlan100 - Group 1 (version 2) State is Standby 1 state change, last state change 31w0d Virtual IP address is 10.8.1.1 Active virtual MAC address is 0000.0c9f.f001 (MAC Not In Use)							

### 8.1.2.2 Create a Paragraph Parser

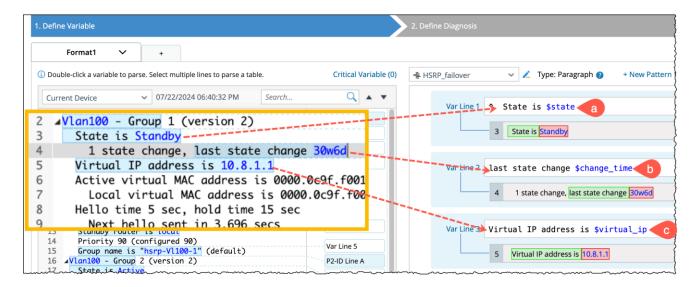
1. On **Line 2**, double-click **Vlan100** to parse the *interface variable* from the command output.

CLI Command Diagnosis									
US-BOS-SW	<b>/1</b> show stand	lby	×v	Retrieve	with Live Data				
1. Define Variat	ble					2. Define Diagnosis	1		
Forma	at1 🗸	+					1		
i Double-click	a variable to parse.	Select multiple lin	nes to parse a table.		Critical Variable (0)	Name: Pattern1	Type: 🔿 Singl		
2 ∡Vlan 3 St 4 5 Vi 6 Ac 7 8 He 9 10 Pr	OS-SW1>chow st 100 1 p 1 ate is Standby 1 state change rtual IP addre tive virtual M Local virtual llo time 5 sec Next hello sen eemption enabl	andby (version 2) e, last state ess is 10.8.1 MAC address i MAC address c, hold time it in 3.696 s	e change 30w6d 1 s 0000.0c9f.f0 is 0000.0c9f.f 15 sec	001 (v2 d	-P1-ID Line	ID Line ^\$var1 - Gr	Group 1 (version 2)		

- 2. In the right Variable Definition pane, update:
  - a) Variable name **\$var1** to **\$interface**.
  - b) Parser name Pattern1 to HSRP\_failover

CLI Command Diagnosis								
🧾 US-BOS-SW1	show standby	× v Retrieve	with Live Data					
1. Define Variable				2. Define Diagnosis				
Format1	✓ +							
(i) Double-click a vari	iable to parse. Select multiple lines to	parse a table.	Critical Variable (0)	Name: HSRP_failover b Type: O Single O Multiple				
Current Device	✓ 07/22/2024 06:40:	32 PM Search	Q • •	ID Line ^\$interface				
3 State 4 1 st 5 Virtua 6 Active	- Group 1 (version 2) is Standby ate change, last state cha l IP address is 10.8.1.1 virtual MAC address is 00 l virtual MAC address is 0	00.0c9f.f001 (MAC N	P1-ID Line	2 Vian100 - Group 1 (version 2)				

- 3. Similar to the step 1, parse other following variables from the sample data:
  - a) **\$State**
  - b) **\$change\_time**
  - c) **\$virtual\_ip**



### d) **\$standby\_router**

#### e) **\$group\_name**

1. Define	Variable						2. Defi	ne Diagnosis			
	Format1 🗸	+									
(i) Doub	le-click a variable to p	arse. Select mu	ltiple lines to parse a ta	ble.	Critical	ariable (0)	<b>H</b> ≞ HS	RP_failover		✓ ∠ Type: Paragraph ? +	New Patterr
Curre	ent Device	✓ 07/22/2	2024 06:40:32 PM	Search	Q	<b>. .</b>			5	Virtual IP address is 10.8.1.1	
1 2 3 4 5 6 7 8 9 11	Virtual IP a Active virtu Local virt Hello time 5 Next bello	p 1 (versio ndby ange, last ddress is 1 al MAC addr ual MAC add sec, hold sect in 3	state change 30w6 0.8.1.1 ess is 0000.0c9f. ress is 0000.0c9f time 15 seç	f001 (MAC N 5.f001 (∨2 d	P1-ID Line Var Line 1 Var Line 2 Var Line 3			Var Line 4	13	ndby router is \$Standby_ro Standby router is ocal up name is "\$group_name"	uter d
12	MAC o	address	is aabb.cc8		c) 100 (	exp ir e			15	Group name is " <mark>hsrp-Vl100-1</mark> " (c	lefault)
13 14	Priorit	ty 90 (c	is <mark>local</mark> onfigured 9				+ F	ield ~			
15	Group r	name is	"hsrp-Vl100	-1" (def	ault)						

- 4. Click **Apply** to exit Visual Parser Auto-Pattern mode
- 5. Exit the Visual Parser window.

						5 ×
e Data						
	2. Define Diagnosis					
}				Test	on Devices: (	⊡ 0
ariable (0)	HSRP_failover	🗸 🞽 Туре:	Paragraph ? +	New Pattern 🗸		≡
• •		5 Virtual IP add	lress is <mark>10.8.1.1</mark>	>	3 Lines	
the pa		uter >	≡ 3 Lines			
variab	ar Line 5	Group name is	"\$group_name"		=	
	Output + P	arse Lines	<b>^</b>			_
	\$interface 🗸	\$state 🗸	\$change_time 🗸	\$virtual_ip 彈 🗸	\$Standby_ro	o
	Vlan100	Standby	30w6d	10.8.1.1	local	
	Vlan100	Active	30w6d	10.8.1.8	10.8.1.3,	
	Vlan101	Standby	30w6d	10.8.1.17	local	
j						
}				Can	4 App	ly "

### 8.1.2.3 Define Diagnosis

- 1. Click **Add to Intent >>** back in the quick intent tab.
- 2. In the **Define Logic** pane, name the diagnosis **HSRP Diagnosis**.
- 3. Anchor: Select the variable *interface* from the drop-down.
- 4. Click in the *Type description of the diagnosis...* field and enter *Check HSRP failover status of device*.

[일 Map15 (Master) ~ * > Page 1~	Summary		Overal			
😞 🚺 Intent	Runbook	D Data View				
Auto Intent Quick Intent	Published Intents	Map Intent	Reset 🛛 🛛 Help			
Collect Data US-BOS-SW1 V		«	Define Logic			
show standby		× ✓ Retrieve ⊕	Create			
04:35:55 PM: Successfully retrieved!			US-BOS-SW1 🛛 show standby 🥐 📮 Diagnosis 🖂			
✓ US-BOS-SW1 ∨ Parser	: 🔜 HSRP_failo 🥐	Edit v				
	1	Add to Intent >>				
07/24/2024 04:35:52 PM (Live) 🎄	Succes	ssfully parsed all 1 variable.				
JS-BOS-SW1>show standby Vlan100 - Group 1 (version 2) State is Standby 1 state change, last stat Virtual IP address is 10.8.	e change <mark>31w1d</mark> 1.1		Diagnosis: D HSRP Diagnosis ~ + New Diagnosis			
Active virtual MAC address Local virtual MAC address	is 0000.0c9f.f00		Name: HSRP Diagnosis 2 Anchor: HSRP_failover.\$inter			
Hello time 5 sec, hold time Next hello sent in 0.608 Preemption enabled			Check HSRP failover status of device 4			
Active router is 10.8.1.3, priority 105 (expires in 13.392 sec)						

5. Check-in **Loop Table Rows** and select the table **HSRP\_failover** and *Virtual\_ip* as table key from the corresponding dropdown menu.

Define	Logic		~				
				c	reate		
🥏 US	-BOS-SW1	🖽 show standby	PI	1 Diagnosis	$\checkmark$		
Diagno	osis: 🖸 HSRP D	iagnosis 🗸	+ New Diagnosis				
Name:	HSRP Diagnosi	s	Anchor: HSRP_failover.\$inter~				
	Check HSRP fa	ilover status of device	e				
🔽 Loc	p Table Rows	∃HSRP_failover ∨	Table Key:	virtual_ip 🗸	0		
∨ If					1		
A	US-BO Cur	rrent 🗸		Baseline 🗸	,		

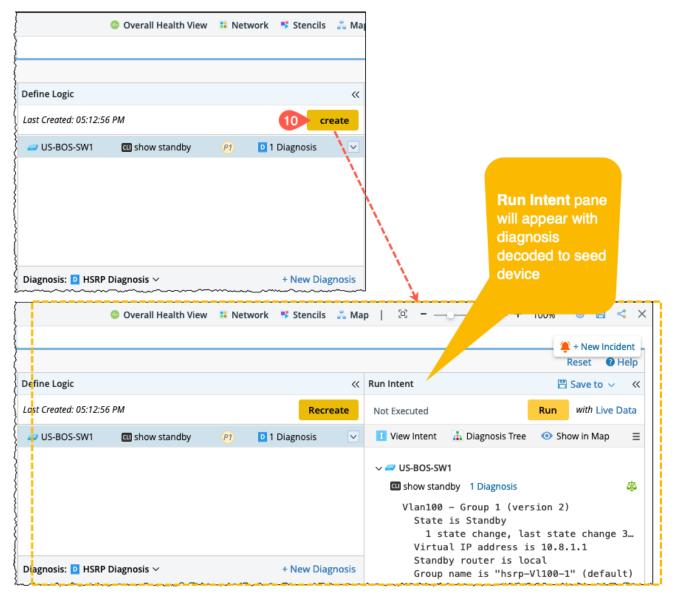
- 6. **If**: Define the condition as follows:
  - a) A: Change\_time (Current) | Does not equal | Change\_time (baseline)
  - b) B: Change\_time (Current) | Does not contain | d
  - c) **C:** *Change\_time* (Current) | Does not contain | w
- 7. Boolean Expression: A and (B or C)

Diag	Diagnosis: D HSRP Diagnosis ~ + New Diagnos									
Nan	ne:	HSRP Diagnosis			Anchor	HSRP_failove	r.\$inter	~		
		Check HSRP failover status of device								
	✓ Loop Table Rows ↓ HSRP_failover ∨ Table Key: virtual_ip ∨ ③ ✓ If 6									
А	4	US-BO Curr	ent 🗸			Bas	eline 🗸			
a	(	change_time	~	Does not e	. ~	change_time	~	面		
В	4	US-BO Curr	ent 🗸							
	(	change_time	~	Does not c	~	d	~	面		
С	4	US-BO Curr	ent 🗸							
	(	change_time	~	Does not c	~ ~	w	~	面		
D		Select Variable	~							
Вс	oole	an Expression:	A and	(B or C)	7					

- 8. **Then**: With the diagnosis logic established, let's display an **Error** message if the result of the logic is TRUE.
  - a) Click the drop-down under **Diagnosis Message** and click the **RED** square.
  - b) Click into the Diagnosis Message text box (*\$intf is down...*), then type: *\$this\_device HSRP \$group\_name* has failover, last failover is at *\$change\_time*.
  - c) Check the boxes next to Set Status Code for Device and Set Status Code for Intent.
- 9. **Else**: Click "+ **Else**" to display a success message if the diagnosis logic is successful.
  - a) Click the drop-down under **Diagnosis Message** and click the **GREEN** square.
  - b) Click into the Diagnosis Message text box (*\$intf is down...*), then type: *\$this\_device* HSRP
     *\$group\_name* is stable.
  - c) Check the selection boxes: Set Status Code for Device and Set Status Code for Intent.

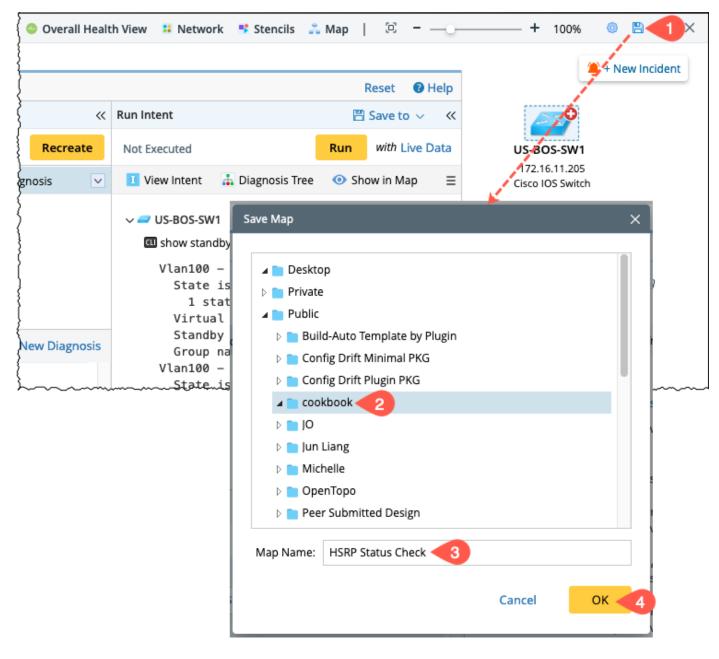


### 10. Click **Create**.

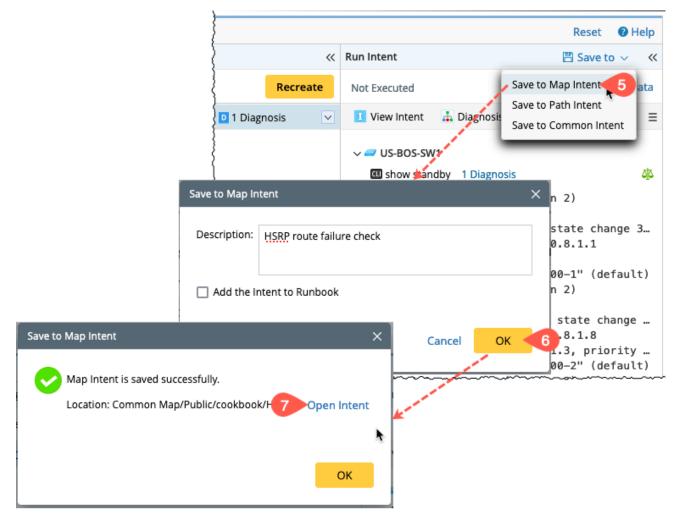


### 8.1.2.4 Save and Execute Map Intent

- 1. Save the map by clicking on the **Save** button located in the top right corner.
- 2. In the **Save Map** dialog, navigate to **Public > cookbook**, then click on the directory.
- 3. Name the map **HSRP Status Check**.
- 4. Click OK.



- 5. In the Run Intent pane, click **Save to > Save to Map Intent.**
- 6. Click **OK** in the dialogue.
- 7. Click Open Intent.



## 8.1.3 Replicate the intents for all HSRP devices using ADT

In this section, you will use the **Intent Replication Wizard** to replicate the intent to all devices in the HSRP device group. As a first step, open the intent from the intent manager and the Intent replication wizard as shown:

Network Intent Manager > Common Map Intent	
Common Map Intent	
Search Q	
All	
🖌 💼 Desktop	
I Ping Cisco	
I Route Check (Cisco IOS)	
a 📄 Public	
a 📄 cookbook	
I HSRP Status Check Open 2	
🔺 📄 Jun Liang	
Network Intent (View Mode) - Common Map/Public/cookbook/HSRP Status Check	×
Network Intent (View Mode) - Common Map/Public/Cookbook/HSKP Status Check	×
I HSRP Status Check HSRP route failure check	<u>"Open ⊜o</u> ⊵0 <u>2</u> 3 ≡
	Named Tag
Result: No History Data 🔒 💿 🔛	View Abstract
No result available because this intent has not been executed.	Run Settings
No result available because this intent has not been executed.	Data Clean Settings
	Edit
	Save as
V I US-BOS-SW1 1 Diagnosis	Delete
	Refresh
Image: show standby     1 Diagnosis	Export
1 US-BOS-SW1>show standby	Share to Incident
2 <u>Vlan100</u> - Group 1 (version 2) 3 State is Standby	Publish Intent
4 1 state change, last state change 31w1d 4	Intent Replication Wizard
5 Virtual IP address is 10.8.1.1 6 Active virtual MAC address is 0000.0c9f.f001 (MAC Not In	Auto Intent Wizard
7 Local virtual MAC address is 0000.0c9f.f001 (v2 default	
8 Hello time 5 sec, hold time 15 sec 9 Next hello sent in 0.608 secs	
10 Preemption enabled	
11 Active router is 10.8.1.3, priority 105 (expires in 13.39	
12 MAC address is aabb.cc80.1500 13 Standby router is local	

- 1. In the first step (**Seed Intent**), Validate the default seed intent and go to the next section.
- 2. In the second step (**Define ADT**), provide the basic input to create a new ADT, such as name, ADT location and target devices.
- 3. In the third step, define the **intent qualification** with device group **HSRP Devices** to include devices for the seed intent to replicate on.
- 4. In the fourth step, **Replicate Intent**, modify the name of the **Replicated Intent** In the **ADT Columns** section and add more columns that you want in the final ADT.

Seed Intent	Define ADT	Replication Settings	Repl	icate Intent
ADT Columns:				Additional Columns ~
Column Data	Column Name		Tag	Replicated Intent
Replicated Intent	HSRP failover		0 tags	🔲 Intent Message
Modify the na intent column			С	Intent Status Code Cevice Status Code Intent Devices Intent Devices
		the boxes of all the is that you want see al ADT	in	Intent Map Intent CLI Comma List Execution Time

5. Click **Save and Replicate** to save all the settings and create ADT. An option **Open Output ADT** will appear. Click it to open the table in the **ADT Manager**.

Seed Intent	Define ADT Replication Settings	Replicate Intent
DT Columns:		Additional Columns >
Column Data	Column Name	Тад
Replicated Intent	HSRP failover	0 tags
s Intent Message	Intent Message	
🚭 Intent Devices	Intent Devices	
Last Execution Time	Last Execution Time	
		5 Save and Replicate

Intent Replication Wizard - HSRP Status Check

T Columns:		Additional Columns ~
Column Data	Column Name	Tag
Replicated Intent	HSRP failover	0 tags
s Intent Message	Intent Message	
Intent Devices	Intent Devices	
Last Execution Time	Last Execution Time	
	Click to open the table in ADT manager.	Save and Replicate

Review the new Intent columns that are added to the table and results.

Search 🔍 😋 <						
Shared Tables (1201)	B HSR	P Route Failure	Table Builder	Last Update	ed at: 07/30/2024 12:57 PM 🛛 🍕 Rebuild Table	Add Data Manually 🗸 📃
My Tables (5)	Descrip	otion: Type description here				
<ul> <li>Ping (1)</li> <li>BGP Config Change Diagnosis</li> </ul>		19 Rows 4 Columns			Search	Q T Advanced Filter: Undefined
👪 Day 1 Lab	No.	Device	HSRP failover		S Intent Status Code	() Last Execution Time
HSRP Route Failure	1	BJ-L2-coreB	HSRP Status Check BJ-L2-coreB	0	BJ-L2-coreB HSRP hsrp-Vl10-100 is stable	07/29/2024 03:55:07 PM
🚯 NTP	2	BJ_L2_Core_3	HSRP Status Check BJ_L2_Core_3	0	BJ_L2_Core_3 HSRP hsrp-Fa1/0/9-10 is stable	07/29/2024 03:55:07 PM
	3	BJ_L2_Core_4	HSRP Status Check BJ_L2_Core_4	0	BJ_L2_Core_4 HSRP hsrp-Fa2/0/3-0 is stable	07/29/2024 03:55:05 PM
	4	BJ_core_3550	HSRP Status Check BJ_core_3550	0	BJ_core_3550 HSRP None is stable	07/29/2024 03:55:07 PM
	5	Bur-isp-gw1	HSRP Status Check Bur-isp-gw1	0	Bur-isp-gw1 HSRP hsrp-Gi0/0/0-200 is stable	07/29/2024 03:55:07 PM
	6	IPv6Lab-SW8	HSRP Status Check IPv6Lab-SW8	0	IPv6Lab-SW8 HSRP hsrp-Et0/1-1 is stable	07/29/2024 03:55:05 PM
	7	IPv6Lab-SW9	HSRP Status Check IPv6Lab-SW9	0	IPv6Lab-SW9 HSRP hsrp-Et0/1-1 is stable	07/29/2024 03:55:07 PM
	8	PE-3600X-01	HSRP Status Check PE-3600X-01	0	PE-3600X-01 HSRP hsrp-Gi0/24-2 is stable	07/29/2024 03:55:07 PM
	9	PE-3600X-02	HSRP Status Check PE-3600X-02	۲	PE-3600X-02 HSRP hsrp-Gi0/13-2 is stable	07/29/2024 03:55:07 PM
	10	PE-ASR1K-01	HSRP Status Check PE-ASR1K-01	0	PE-ASR1K-01 HSRP hsrp-Te0/0/0-15 is stable	07/29/2024 03:55:05 PM
	11	PE-ASR1K-02	HSRP Status Check PE-ASR1K-02	0	PE-ASR1K-02 HSRP hsrp-Gi0/0/5-10 is stable	07/29/2024 03:55:05 PM
	12	Sjc-Dist-3750-02	HSRP Status Check Sjc-Dist-3750-02	0	Sjc-Dist-3750-02 HSRP hsrp-Vl30-0 is stable	07/29/2024 03:55:07 PM
	13	US-BOS-SW1	HSRP Status Check US-BOS-SW1	0	US-BOS-SW1 HSRP hsrp-Vl100-1 is stable	07/29/2024 03:55:07 PM
	14	US-BOS-SW2	HSRP Status Check US-BOS-SW2	0	US-BOS-SW2 HSRP hsrp-Vl100-1 is stable	07/29/2024 03:55:05 PM
	15	bjta002237-SW2	HSRP Status Check bjta002237-SW2	۲	bjta002237-SW2 HSRP hsrp-Vl481-1 is stable	07/29/2024 03:55:07 PM
	16	bjta002238-SW3	HSRP Status Check bjta002238-SW3	0	bjta002238-SW3 HSRP hsrp-Vl481-1 is stable	07/29/2024 03:55:05 PM
	17	bjta002444-SW13	HSRP Status Check bjta002444-SW13	٥	bjta002444-SW13 HSRP hsrp-Et1/1-1 is stable	07/29/2024 03:55:05 PM
	18	bur-isp-gw2	HSRP Status Check bur-isp-gw2	0	bur-isp-gw2 HSRP hsrp-Gi0/0/0-200 is stable	07/29/2024 03:55:07 PM
	19	gapp-c3560-2	HSRP Status Check gapp-c3560-2	0	gapp-c3560-2 HSRP hsrp-Gi0/15-0 is stable	07/29/2024 03:55:07 PM

## 8.1.4 Create the Dashboard

Let us create the intent dashboard from the ADT. Open the **New Intent Dashboard** from the intent HSRP Failover column  $\equiv$  menu.

		>>	-	RP Route Failure	ere	Table	Builder	Last Updated at	: 07,
			ltems	: 43 Rows 4 Columns					
			No.	管 Device		IH	SRP Failover		6
			25	RT02-VNDR-TTEC-AUS-T	X-US.citfg.com		Run Intents O		
			26	RT02-VNDR-TTEC-CENT	CO-US.citfg.com		Run Intents vi		
			27	SW-WAL-01-FH12-MED-			Open Seed In Rebuild Inter	tent nt-related Column	Gro
e Intent Dashbo	ard				× <sup>II-US.citfg.c</sup>			ty Wrapper Intent	0.0
					ll-US.citfg.c		Enable Auto I	ntent	
Create Inten	t Dashboard for ADT 'HSRF HSRP र्हेoute check	P Rou	ute Fa	llure'	5	-	Export Diagno /iew Summar Export Intent		
Location:	🖿 My Dashboards 🗸	4					Debug Empty	Cells	
🗌 Use Templat	e					н	Tag Current C	olumn	
						н	Edit		
Data Source	Automation Data Table			~		н	Delete Set as Table K	ίον.	
Automation Da	ta Table: HSRP Route Failure					H		-	
🗸 Include Trigg	gered Follow-up Intent Results						New Intent Da	ed Commands to Be ashboard	nchr
						н		ck US-BOS-SW1	
Intent Column:	HSRP Failover 🗸							k US-BOS-SW2	
Filter Intent	by Devices					HSR	P Status Cheo	k US-NYJ-CW01-02	
						- and the sold of		an de Martin de Castalet	
Time Range	Last 7 Days			~					

In the **Create Intent Dashboard** dialog, **Open Intent Dashboard** to view the results summary and result history.

HSRP Route check					Last Refreshed at 30/7/202	24, 1:04:57 pm	0	<b>//</b>	Ø
Summary		30/7/2024, 1:05:00 pm	View Report	Device Information	1	30/7/2024,	1:05:00 pr	m Vi	ew Rep
61	intents								
19	Times I	Executed					19		Devic
0	Intent-l	evel Alerts	- 1	Cisco IOS	5 Switch 🛛 🔵 Cisco Router				
						30/7/2024	1:05:00 pr	n vi	OW ROM
Intent Result History Time Range: All ~ Result: All ~	•	Sum of Intent Alert Stat	us Code Count	→ Sum of Intent Succes	ss Status Code Count	30/7/2024,	1:05:00 pr	n Vi	ew Re
Time Range: All $\checkmark$ Result: All $\checkmark$	29/7/2024, 3:55:07			•	•	30/7/2024,			
Time Range: All $\checkmark$ Result: All $\checkmark$				24, 3:55:07 pm 29/7/2	•	•			ew Rep :55:07
Time Range: All $\checkmark$ Result: All $\checkmark$			07 pm 29/7/202 Top Five In	24, 3:55:07 pm 29/7/2 tent Alerts	2024, 3:55:07 pm 29/7/202	•	29/7/;		
Time Range: All ~ Result: All ~ 20 0 29/7/2024, 3:55:05 pm	<b>2</b> 9/7/2024, 3:55:07	pm 29/7/2024, 3:55:0	07 pm 29/7/202 Top Five In	24, 3:55:07 pm 29/7/2 tent Alerts	2024, 3:55:07 pm 29/7/202	24, 3:55:07 pm	29/7/;		

# 8.2 Follow-up to Check the Performance Degrade After Failover

In this section, we shall add two intents as follow-up for the HSRP device if it has failover in the last 24 hours. We shall use the intents created in <u>Section 3</u> (*CPU usage Check* and *Interface Status Check*) as a follow-up:

**NOTE**: Before proceeding further, go to the intents (*CPU usage Check* and *Interface Status Check*) **Edit Mode>Replication Logic** and define the **Intent Qualification** as HSRP device group.

Network Intent (Edit Mode)		
CPU Usage Check (Cisco IOS)     Diagnosis Tree  Type description here	Add HSRP device grou	Run with Live
I Seed Logic 🗍 Replication Logic 🤣		
Serve as Template for: O Device-based Replication	O Path ased Replication	Enable Neighbor Pair Replicati
Intent Qualification:  • via Device Groups/Sites: 1 Devi	ce Groups/Folders  🖻	O via Dyriamic Search: Defined
2 Define Rules to Replace Macro Variable in Seed Command	d 0/0	<b>3</b> Define Criti

- 1. Open the seed intent **HSRP status check** in edit mode.
- 2. Click **Edit Diagnosis** to open the parser.

Network Intent (View Mode) - All Ne	twork Intents/Cookbook/HSRP S	Status Check		×
I HSRP Status Check	HSRP route failure check		🗸 Open 📄 🛛	1 Edit =
Result: No History Data	🚠 💿 🔛		Run	with Live Data
No result available because this inter	t has not been executed.		and the second s	
				View ~
letwork Intent (Edit Mode)				×
HSRP Status Check	Tree	Run with Live Data	Save Save	=
SRP route failure check			💦 Intent Map: Select	~
Seed Logic				
	Logic 🧭		1	
Herice		Intent Variables:		
🗸 🚅 US-BOS-SW1	Type Description here	+ Add Config Diagnosis	🖽 + Add CLI Diagnosis 🗮	
🔺 💷 show standby 💈	Type Description here		2 Edit Diagnosis ≡	
1JS-BOS-SW1>show standby2Vlan100- Group 1 (version	2)	HSRP D	agnosis 💼	
<ul> <li>3 State is Standby</li> <li>4 1 state change, last s</li> </ul>	tate change 31w1d	📄 Messag	e 🚺 Status Code	
5 Virtual IP address is 10	.8.1.1			
	ss is 0000.0c9f.f001 (MAC No ess is 0000.0c9f.f001 (v2 de			
8 Hello time 5 sec, hold t	ime 15 sec			
9 Next hello sent in 0.6 10 Preemption enabled	08 secs			
	3, priority 105 (expires in	13.		
12 MAC address is aabb.cc				
13 Standby router is local 14 Priority 90 (configured)	202			
15 Group name is "hsrp-Vl10				
16 Vlan100 - Group 2 (version				
17 State is Active				
18 2 state changes, last 19 Virtual IP address is 10				
20 Active virtual MAC addre	ss is 0000.0c9f.f002 (MAC Ir	n Use)		

- 3. In the diagnosis window, go to the **Define Diagnosis** ribbon.
- 4. **Then:** Check the **CPU usage** and **Interface Status** by running the corresponding intents as the follow up intent. Remove the **Diagnosis Message** and add **Follow-up Intent** from the **Add logic** dropdown.

2. Defi	ìne Diagnosis 🧲	3						
Add N	Note D Add I	Diagnosis	Can also	o click a v	variable on the left to add a	utomat	tion.	
HSRP Diagnosis B	🚄 US-BOS-S	Current 🗸						
	change_time	~	Does not contain	~ d	~	Î		
С	🥏 US-BOS-S	Current 🗸						
	change_time	~	Does not contain	~ w	· ~			
D	Select Variable	~						
Boo	olean Expression:	A and (B or	C)					
↓ √ Ther	4							
{	agnosis Message:				Save to Inciden	t ≡		
}	nt Data View	RP \$g	roup_name has failo	ver, last f	failover is at \$change_time			
Draw	и Мар	> :e:						
Send	l Email	vice HS	RP \$group_name ha	s failover	er, last failover is at \$change	_tim	-	
Follow	w-up Intent	it:						
Set Ir	ntent Baseline	vice HS	RP \$group_name ha	s failover	er, last failover is at \$change	tim		
Adva	nced	>						
Add Log	gic 🗸							
+ Add	Elself	1						
		ί.			Cancel	Apply		
						,		
	}~ Then	1	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~		~~~
	📔 Dia	agnosis Mess	age:			🗌 Sa	ve to Incident 🔳	
	}	✓ \$t	his_device HSRP \$gro	oup_nam	ne has failover, last failover	is at \$c	change_time	
	2 🔽 🛛	Set Status (	Code for Device:					
	{	🕽 Error 🗸 🗸	\$this_device HSR	P \$group	p_name has failover, last fa	ilover i	s at \$change_tim	
	{ 🔽 s	Set Status (	Code for intent:					
	{	🕽 Error 🗸 🗸	\$this_device HSR	P \$group	p_name has failover, last fa	ilover i	s at \$change_tim	
	}		¥		•			
	Fol	llow-up Inten	t: 💿 Network Intent	5	rrent Intent (Self) O S	top	=	

- 5. Click the **Network Intent** link to choose the follow-up intent in a new dialog **Follow-up Intents**.
  - a) In the Follow-up Intents window, choose **Select Intent Template**.
  - b) In the Select Intents dialog, navigate to the follow-up intents (**CPU usage** and **Interface Status**) and check the selection box.
  - c) Click **OK** to save the selection and close the window.

Follow-up Intents		×
0 Follow-up Intents:	+ Follow-up V 🚯	Select Intent Templates
	a Select Intent Template	Select Intent Template from:  All Intents Installed Intents
	Select Intents from ADT Select Intent Templates via ADT	Type:     Common Intent     ~     Filter by:     no filter     ~     cookbook     ×
	Select Intent Cluster	All         (10.10.10.1 (6)         (192.168.29.62 (4))         (3Com (5))         (5 Report (2))         (AAA (5))         (Acces >>)
		All Network Intents     Cookbook
		I       CPU Usage Check (Cisco IOS)         I       HSRP Status Check
		Image: Description of the second s
		▷ <b>□</b> GD
		Саг С ОК

- d) Define Replicate Current Intent to: **Device by Variable** | **this\_device.**
- e) Click **Save** to save and close the follow-up intents.

Follow-	up Intents						×
2 Follo	ow-up Intents: +	Follow-up	<ul> <li>✓ (1)</li> </ul>				
~	CPU Usage Check (	(Cisco IOS)	(Intent Template)				1
De	escription: When						
	Replication Setting	gs:					
<b>d</b>	Replicate Current In	itent to:	Device by Variable	~	this_device	~	
	Merge multiple r	eplicated i	ntents into one				
	Follow-up Execution	n: Setting	5				
	Draw Arrow from	n This Devi	ce to Next Settings				
Pri	ine other follow-up in	tents				Cance Sav	
						Cance Sav	/e

- 6. Click **Apply** to save and close the diagnosis window.
- 7. In the Network Intent (Edit Mode) dialog, click **Save**.
- 8. Click **Run** to execute the intent.

Network Intent (Edit	Mode)			×
I HSRP Status Ch	eck 🚠 Diagnosis Tree		8 Run with Live Data 7	Save 🛛 Help 😑
HSRP route failure c	heck			💦 Intent Map: Select 🗸 🗸
I Seed Log	gic 🔲 🔃 Replication Logic 🥑			
😁 + Device	J		Intent Variables: Manager	Tag: + Add 🔳
🗄 🗸 🚄 US-BOS-SV	N1 Type De	escription here	• Add Config Diagnosis	d CLI Diagnosis 📃
▲ 💷 show stand	by 🛃 Type De	escription here		Edit Diagnosis 🛛 🚍
Network Intent (View Mode) - All Netwo	rk Intents/Cookbook/HSRP Status Che	eck		× 💼
I HSRP Status Check	HSRP route failure check		💑 Open 📄 0 🔥 0 🗶 Edit	=
Result: 08/02/2024 04:57 PM 🗸	🚠 💿 🔛		Run 🛛 🗸 with Live Da	ta
This intent execution is finished at 08,	/02/2024 04:57 PM with 0 errors. You	can View Execution Log		
S US-BOS-SW1 HSRP hsrp-VI100-	1 is stable 6		Viev	v~
v 🥔 US-BOS-SW1	S US-BOS-SW1 HSRP hsrp	isstable 3		
<ul> <li>▲ I show standby</li> <li>1 JS-BOS-SW1&gt;show standb</li> </ul>		Result is displayed. In this example: Stats was		
2 Vlan100 - Group 1 (ver 3 State is Standby 4 1 state change, la 5 Virtual IP address i 6 Active virtual MAC a	sion 2) st state change 32w3d s 10.8.1.1 ddress is 0000.0c9f.f001 (MAC		US-BOS-SW1 HSRP hsrp-VI100-1	
<ul> <li>8 Hello time 5 sec, ho</li> <li>9 Next hello sent in</li> <li>10 Preemption enabled</li> <li>11 Active router is 10.</li> <li>12 MAC address is aab</li> </ul>	4.848 secs 8.1.3, priority 105 (expires b.cc80.1500			
13Standby router is lo14Priority 90 (configu15Group name is "hsrp-16Vlan100 - Group 2 (ver	red 90) Vl100-1" (default)		US-BOS-SW1 HSRP hsrp-VI100-2	

9. Close all the network intent dialogues and go to **ADT Manager** to update the replicated intents in ADT.

## 8.2.1 Update replicated intents in ADT

- 1. Go to Automation Data Table Manager > My Tables> HSRP Route Failure.
- 2. Open the  $\equiv$  menu from the **Intent** Column.
- 3. Select Rebuilt Intent-related Column Group.

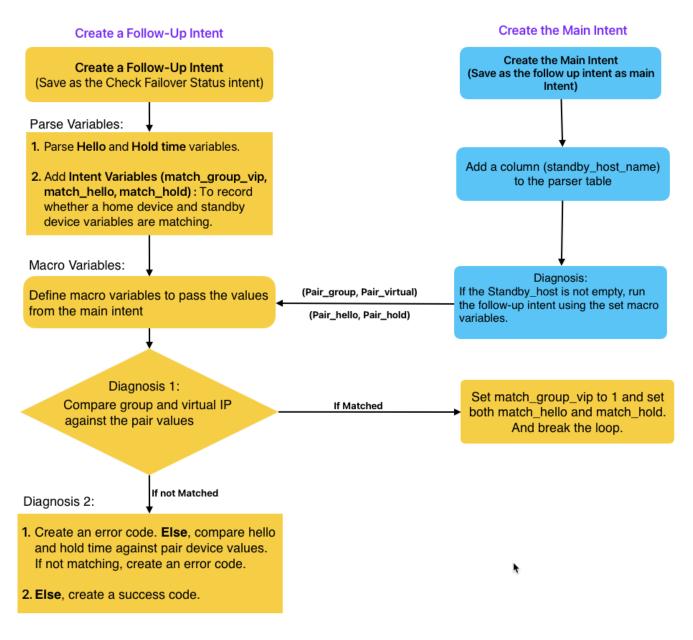
Search Q	> << 🚆	HSRP Route Failure	Table Builder	l ast l Ind	lated at: 08/02/2024 05:09 PM 🛛 🍳 Rebuild Table		
<ul> <li>Shared Tables (1208)</li> <li>My Tables (6)</li> </ul>		Image: Big HSRP Route Failure     Table Builder     Last Updated at: 08/02/2024 05:09 PM     Rebuilder       Description:     Type description here					
Ping (1) BGA Config Change Diagnosis	lte	ems: 19 Rows 4 Columns			Search		
🖽 Day 1 Lab	No	o. 🔮 Device	I HSRP failover	=	11 nt Status Code		
👪 HSRP Pair failover	1	BJ-L2-coreB	Run Intents Once	PA	BJ-L2-coreB HSRP hsrp-VI10-100 is stable		
HSRP Route Failure	2	BJ_L2_Core_3	Run Intents via Timer	PA	BJ_L2_Core_3 HSRP hsrp-Fa1/0/9-10 is stable		
ntp 📆	3	BJ_L2_Core_4	Open Seed Intent		BJ_L2_Core_4 HSRP hsrp-Fa2/0/3-0 is stable		
	4	BJ_core_3550	Rebuild Intent-related Column	Group	12 re_3550 HSRP None is stable		
	5	5 Bur-isp-gw1 Remove Empty Wrapper Intent	Bur-isp-gw1 HSRP hsrp-Gi0/0/0-200 is stable				
	6	IPv6Lab-SW8	Enable Auto Intent		IPv6Lab-SW8 HSRP hsrp-Et0/1-1 is stable		
	7	IPv6Lab-SW9	Export Diagnosis Result to CSV		IPv6Lab-SW9 HSRP hsrp-Et0/1-1 is stable		
	8	PE-3600X-01	View Summary Report		PE-3600X-01 HSRP hsrp-Gi0/24-2 is stable		
	9	PE-3600X-02	Export Intent Output Map Debug Empty Cells		PE-3600X-02 HSRP hsrp-Gi0/13-2 is stable		
	10	PE-ASR1K-01	1		PE-ASR1K-01 HSRP hsrp-Te0/0/0-15 is stable		

- 4. In the Intents column, hover the mouse on the column header and click **Run**.
- 5. Click **Rebuild** to refresh the results in the **Intent Status code**.

₿	Au	tomation Data Table	Manager				
»>	👪 HS	RP Route Failure	oute Failure Table Builder Last Updated at: 08/02/2024 05:09 PM 🔍 Rebu				
	Descri	ption: Type description here					
	ltems:	19 Rows 4 Columns					
	No.	Device	I HSRP failover	Run 1	3 ils ≡	S Intent Status Code	
1		BJ-L2-coreB	HSRP Status Check	BJ-L2-coreB	0	BJ-L2-coreB HSRP hsrp-Vl10-100 is stable	
2	2	BJ_L2_Core_3	HSRP Status Check	BJ_L2_Core_3	0	BJ_L2_Core_3 HSRP hsrp-Fa1/0/9-10 is stable	
3	3	BJ_L2_Core_4	HSRP Status Check	BJ_L2_Core_4	0	BJ_L2_Core_4 HSRP hsrp-Fa2/0/3-0 is stable	
4	ļ	BJ_core_3550	HSRP Status Check	BJ_core_3550	0	BJ_core_3550 HSRP None is stable	
5	5	Bur-isp-gw1	HSRP Status Check I	Bur-isp-gw1	0	Bur-isp-gw1 HSRP hsrp-Gi0/0/0-200 is stable	
~				~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		······	

# 8.3 Check Failover Consistency of HSRP Pair Devices

Let us create an intent to check whether the hello and hold timers are equal for the failover pair devices. You are going to create a parent intent to parse all HRSP pair devices and call a follow-up intent for each of these pair devices, similar to Section 8.1. However, since you are going to compare the hello/hold timers between pair devices, you need to pass the values of these timers from the parent to the follow-up intents, which can be done by defining the variables as Macro Variables in the follow-up intents and setting the Macro Variables in the follow-up intent replicated settings. The following diagram shows the data flow from the parent and follow-up intents:



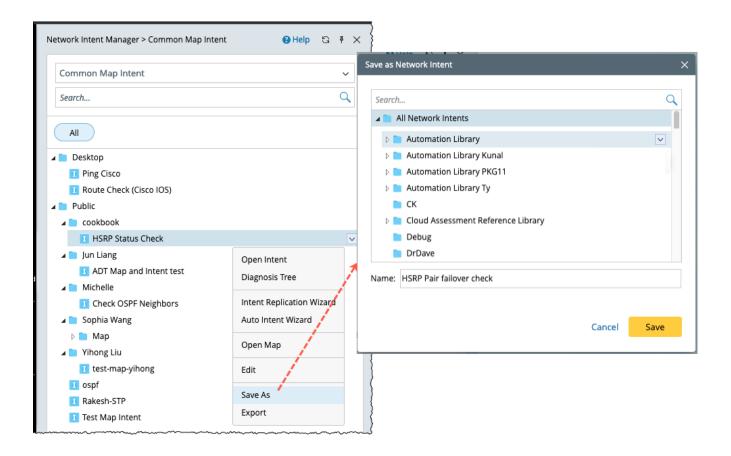
#### Check Failover Consistency of HSRP Pair Devices

- 1. <u>Create a follow-up intent</u>
- 2. <u>Create a parent intent</u>
- 3. Run the intent and view the diagnosis tree
- 4. <u>Replicate the intent to all HSRP devices using ADT</u>

#### 8.3.1 Create a follow-up intent

Create a follow-up intent using the intent created in Section 8.1 since we use the same CLI command and parser:

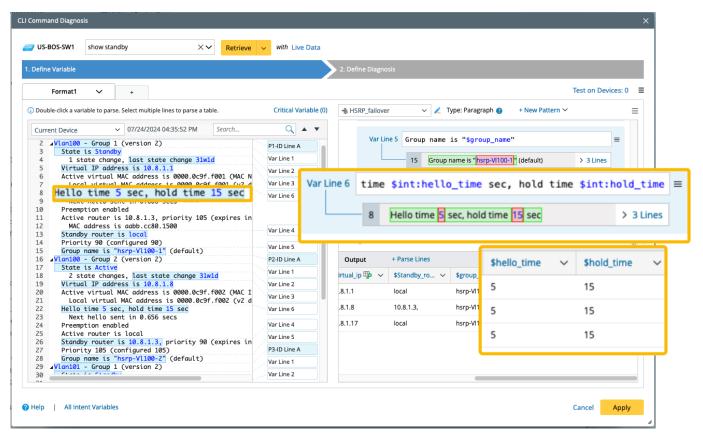
1. Save it as a new intent with the name **HSRP pair failover check\_followup**.



2. Open the new intent **HSRP pair failover check\_ followup** in edit mode and click **Edit Diagnosis** to open the parser.

Network Intent (Edit Mode)		×
I HSRP Pair failover check	Diagnosis Tree	Run with Live Data Save @ Help =
HSRP route failure check		Intent Map: HSRP Status Check 🗸 🗸
I Seed Logic	ication Logic 🤣	
🚳 + Device		Intent Variables: Manager   Tag: + Add 🔳
V IS-BOS-SW1	Type Description here	+ Add Config Diagnosis     Image: Config Diagnosis
🔺 💷 show standby 🛃	Type Description here	Edit Diagnosis 🛛 🚍
5 Virtual IP address 6 Active virtual MAC	ersion 2) Last state change 31w1d	Untitled Diagnosis 1

3. Parse the variables *hello* and *hold* time from retrieved CLI command data.



- 4. Add three intent variables:
  - a) *match\_group\_vip* (int, default value: 0),
  - b) *match\_hello*, (int, -1)
  - c) *match\_hold* (int, -1)

These variables will be used to record whether a group and virtual IP match the pair's group and virtual IP while looping through all groups, along with their corresponding **hello** and **hold** values.

US-E	BOS-SW1	show sta	ndby	X <b>Retrieve</b>	✓ with Live	Data	}	
1. Define	e Variable		Intent Variables for Seed L	ogic				
	Format1	~						
-			Intent Variable	Use Automation	Data Table	Task Variabl	e	
<li>Doub</li>	ole-click a varia	ble to pa						
			+ Add Intent Variable	+ Add Intent Table				
Curre	ent Device							
2	⊿Vlan100 -	Group	🔺 🚺 Intent					N
3	State i	s Star						
4		ate cha	🐹 match_group_vi	p Type: number	Initial	Value: 0	Add Intent Variable	
5	Virtual						Add Intent Table	
6	Active		🐹 match_hello	Type: number	Initial	Value: -1		
7		virtu	- match hold	Tunar number	Initial	Value: -1		
8	Hello t		match_hold	Type: number	Initial	value: - I		
9 10	Preempt	hello	🖌 🥏 US-BOS-SW1			a alla s		
10	Active		A = 02-802-2001		show star	haby		
12		ddress	▷ , HSRP_failover					
13	Standby							
14	Priorit		US-BOS-SW1		🖪 Built-in D	ata		
15	Group n		03-003-3441		B Built-In D	ald		
16	Vlan100 -	Group						
17	State i	s Acti					Close	
18		ate cha					Close	
19	Virtual		-					
20			MAC address is 0000.0		Var Line 3	14	101	
21			1 MAC address is 0000	.0c9f.f002 (v2 d		Viar		
22			ec, hold time 15 sec		Var Line 6		{	
23 24		hello s	ent in 0.656 secs				{	
24		· · · · ·					2	
							{	

5. Close the parser to go back to the **Network Intent** (**Edit Mode**) window.

- 6. From the menu  $\equiv$ , select **Full Settings for Template.**
- 7. Go to the **Macro Variable** tab, select the seed device and click **+Command Variable** to add macro variables.
- 8. Set macro variables: *pair\_group*, *pair\_virtual\_ip*, *pair\_hello*, *pair\_hold*, and *pair\_device*. We shall pass these variables from the parent intent to the follow-up intents for comparison.
- 9. Click **OK** and close the dialog.

US-BOS-SW1     US-BOS-SW1     US-BOS-SW1>show     Vlan100 - Group     State is Stand		Type Descrij Type Descrij	iption here		Run c + Add	6 Full : Intent V Config Di Add	nt Settings nt Variables Settings for Settings Intent Diagr	Template	Help
Seed Logic  S + Device  US-BOS-SW1  US-BOS-SW1>show  J US-BOS-SW1>show  Vlan100 - Group  State is Stand	standby				+ Add	6 Full : Intent V Config Di Add	nt Variables Settings for Settings Intent Diagr	iosis Block	
US-BOS-SW1     US-BOS-SW1     US-BOS-SW1>show     Vlan100 - Group     State is Stand	standby				C + Add	6 Full : Lock Intent Config Di Add	Settings for Settings Intent Diagr	iosis Block	
US-BOS-SW1     US-BOS-SW1     US-BOS-SW1>show     Vlan100 - Group     State is Stand	standby				☐ + Add	Intent V Lock Add Config Di Add	Settings Intent Diagr	iosis Block	
<ul> <li>US-BOS-SW1</li> <li>Ilps-BOS-SW1&gt;show</li> <li>ILps-BOS-SW1&gt;show</li> <li>Vlan100 - Group</li> <li>State is Stand</li> </ul>					5 + Add	Config Di Add	-		
▲ I show standby ↓ US-BOS-SW1>show 2 Vlan100 - Group 3 State is Stand					O + ACO	Auu	Diagnosis vi	a Auto Intent	
1JS-BOS-SW1>show2Vlan100 - Group3State is Stand		Type Descrij	ption here			Currit	ch Devices		
2 Vlan100 - Group 3 State is Stand			,				ne Abstract		
					and the second sec	Unt Nam	ned Tag		
	by					P_ Int∉ Expo			
Settings for Intent Template							×	for Diagnose	s (
🔵 Serve as Template for: 🧿	Device-based Replicatio	n 🔿 Path-based R	Replication	🗌 Enab	le Neighbor Pair Replication			t for Diagnose	
Intent Qualification 7	Macro Variable	Critical Variable	Advan	ce Settings				Wizard	
tems + Device Variable	+ Command Variable	-8							
Seed Device	Macro Variable	Source	Туре	Default Value	Look up Data for Device				
JUS-BOS-SW1		<u>\</u>							
í	😡 pair_group	Device	string	0	Used Default Value	~	·		
	😡 pair_virtual_ip	Device	string	0	Used Default Value	~	·		
	🛃 pair_hello	Device	number	0	Used Default Value	~	·		
	😡 pair_hold	Device	number	0	Used Default Value	~	·		
	😡 pair_device	Device	string	0	Used Default Value	~	· .		

- 10. Back in the Network Intent (Edit Mode) window, click Edit Diagnosis to open.
- 11. Go to **Define Diagnosis** riboon and delete all the old diagnoses.

- 12. Add a diagnosis to loop through all groups:
  - a) **If** condition: Compare *group\_name* and *virtual\_ip* against the *pair\_group* and *pair\_virtual\_ip*.
  - b) **Then**: If the condition is true, go to *Add Logic>Advanced>Set Intent Variable* and set the variables *match\_group\_vip* to 1 and *match\_hello* and *match\_hold as* in the image.
  - c) Add the logic **Break Current Loop** from *Add Logic>Advanced> Break Current Loop*. This break is added here since we already found the paired device and so do not need to continue.
  - d) Click Apply.

	Note D Add Dia	gnosis		can also clic	k a varial	ble on the left i	to add ai	uton	
Name:	Untitled Diagnosis 1			ŀ	nchor:	HSRP_failove	er.\$virtu	~	
	Type description of th	ne diagnosis.							
✓ Loo ∨ If	p Table Rows 📜 HS	SRP_failover	∼ Ta	ble Key: vir	ual_ip	~			
A	🥔 US-BOS-S C	urrent 🗸							
	group_name	~	Equals	~	pair_g	roup	~	Î	
В	🥔 US-BOS-S C	urrent 🗸							
Į	virtual_ip		Equals	~	pair_vi	rtual_ip	~	ī	
С	Select Variable	~						-	
	t Intent or Task Variable ch_group $\checkmark = 1$							=	
	t Intent or Task Variable		mo					=	
<pre>match_hello ~ = \$hello_time</pre>									
	E Set Intent or Task Variable:								
	م الما الم	and the second							
mat	ch_hold  v = \$	\$hold_tim	ie		_				
mat		\$hold_tim	1e					×	
mat	eak Current Loop	\$hold_tim						×	

- 13. Add another diagnosis:
  - a) **If** condition: Variable *match\_group\_vip* |Equals |0.
  - b) **Then**: If the condition is true, then choose **RED** and error message: *\$pair\_device* and *\$this\_device* are HSRP Pairs. Standby device Group and Virtual\_IP are not the same.
  - c) Check the selection boxes: Set Status Code for Device and Set Status Code for Intent.

	2. Define Di	iagnosis				
Q v ^	📄 Add Note	Add Diagnosis	Can al	so click a variab	le on the left to add	d automatic
Untitled Diagnosis 2	Name: Un	titled Diagnosis 2		Anchor:		~
Untitled Diagnosis 1	Тур	pe description of the diagnos	is			
	□ Loop Tab ∽ If	ole Rows				
	А	Current				
	a ma	atch_group_vip 🗸 🗸	Equals	~ 0		~ 🖬
	B Sel	ect Variable 🗸 🗸				
	∽ Then					
	Diagnosi	is Message:			Save to Incid	lent 🔳
	b	\$pair_device and \$this	s_device are HSRP I	Pairs and the st	andby device Group	and V
	Set :	Status Code for Device:				
	🕕 Err	or v \$pair_device an	d \$this_device are l	HSRP Pairs and	the standby device	Group
С	Set :	Status Code for Intent:				
	e Erro	ror v \$pair_device an	d \$this_device are I	HSRP Pairs and	the standby device	Group
	Add Logic 🗸					
	+ Add Elsel	If				

d) Add an **Elself** condition:

A: *match\_hello* | Does not equal | *pair\_hello* 

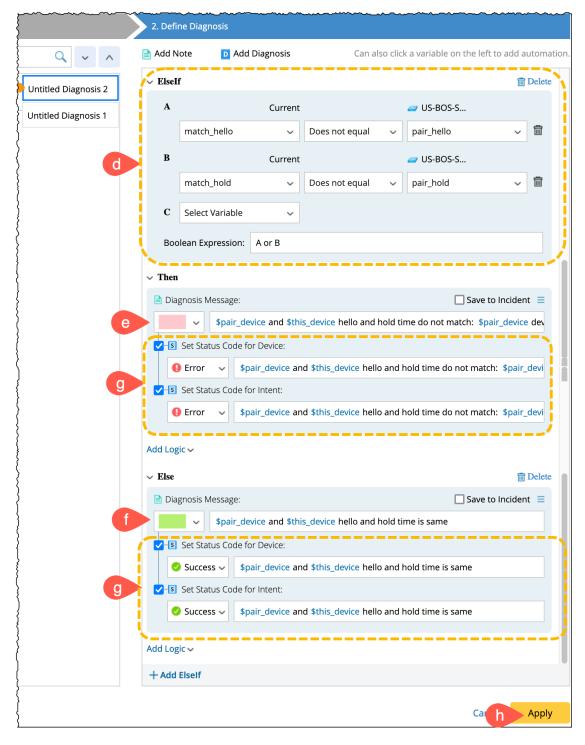
B: *match\_hold* |Does not equal| *pair\_hold* 

Boolean Expression: A or B

e) **Then**: If Elself condition is true, then choose **RED** and error message:

\$pair\_device and \$this\_device hello and hold time do not match:
\$pair\_device device Hello time \$pair\_hello and hold time \$pair\_hold
\$this\_device device Hello time \$match\_hello and hold time \$match\_hold

- f) Else: If Elself condition is not true, then choose Green and success message:
   *\$pair\_device* and *\$this\_device* hello and hold time is the same.
- g) Check the boxes next to Set Status Code for Device and Set Status Code for Intent.
- h) Click **Apply** to save and close the diagnosis window.



14. Back in the Network Intent (Edit Mode), Click **Save** and **Run** to execute the intent.

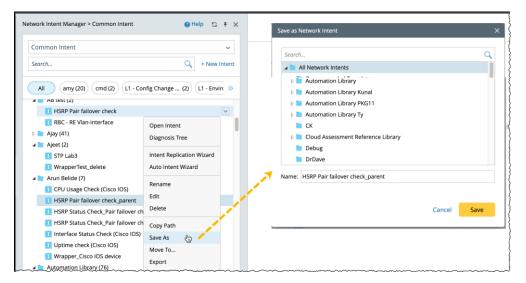
NOTE: You can adjust the default value of macro variables to cover all conditions.

Network Intent (Edit Mode)	······································		×
HSRP Pair failover check_followug	Diagnosis Tree	Run with Live Data	Save 🕜 Help 😑
Seed Logic	ition Logic 🤣		
Network Intent (View Mode) - All Network I	ntents/Arun Belide/HSRP Pair failover check_followup		× Tag: + Add 🔳
HSRP Pair failover check_follow	HSRP route failure check	💑 Open 📄 0 🔥 0 🛛 🔏 Edit	≡ <sup>DLI Diagnosis</sup> ≡
Result: 08/02/2024 01:49 PM マ This intent execution is finished at 08/02/	2024 01:49 PM with 0 errors. You can View Execution Log	Run 🗸 🗸 with Live Data	edit Diagnosis =
S Standby device and US-BOS-SW1 a	re HSRP Pairs and the standby device Group and Virtual_IP do not match Standby device and US-BOS-SW1 are HSRP Pai 1	2 View	~ 🔟
▲ 💷 show standby	2 Diagnoses	Error message appeared after	
<ul> <li>7 Local virtual MAC add</li> <li>8 Hello time 5 sec, hold</li> <li>9 Next hello sent in 4.</li> <li>10 Preemption enabled</li> </ul>	state change 32w3d 10.8.1.1 ress is 0000.0c9f.f001 (MAC Not In Use) dress is 0000.0c9f.f001 (v2 default) time 15 sec 000 secs 1.3, priority 105 (expires in 12.640 sec) cc80.1500	successful execution	

#### 8.3.2 Create a parent intent

Let us create a parent intent based on the intent created in Section 8.3.1 as a parent intent.

1. Save As the follow up intent as HSRP Pair Failover Check\_parent.



- 2. Open the new intent **HSRP pair failover check** in the edit mode.
- 3. Remove the macro variables that are set while configuring follow-up intent :
  - a) Open **Full settings for Template** from  $\equiv$  menu.

Network Intent (Edit Mode)							×
🚺 HSRP Pair failover check_parent 🛛 🛔 Diagnosis Tra	e	Run	with Liv	e Data	Save	🕑 Help	
HSRP route failure check				Intent Settings Intent Variables	;		
I Seed Logic			a	Full Settings for	Template	6	
😂 + Device			Intent \	Lock Settings	n a sia Dia ak		
✓ 🚄 US-BOS-SW1	Type Description here	🗟 + Add	Config Di	Add Intent Diag Add Diagnosis v			
▲ 💷 show standby 🕃	Type Description here			Switch Devices Define Abstract			
1 US-BOS-SW1>show standby 2 Vlan100 - Group 1 (version 2)			D Sta	Named Tag			
3 State is Standby			😡 Foll	Export			
4 1 state change, last state change 3 5 Virtual IP address is 10.8.1.1	1w1d			Save as			
<ul> <li>Active virtual MAC address is 0000.0c</li> <li>Local virtual MAC address is 0000.0</li> </ul>			-	View Original Text for Diagnoses			0
8 Hello time 5 sec, hold time 15 sec	C31.1001 (V2 delddic)			View Summary Text for Diagnoses			
<ul> <li>9 Next hello sent in 0.608 secs</li> <li>10 Preemption enabled</li> </ul>				Publish Intent			
11 Active router is 10.8.1.3, priority 1	05 (expires in 13.392 sec)			Intent Replication	on Wizard		
12 MAC address is aabb.cc80.1500				Auto Intent Wiz	ard		
13 Standby router is local 14 Priority 90 (configured 90)							-
15 Group name is "hsrp-Vl100-1" (default	)						
16 Vlan100 - Group 2 (version 2)							
State_is_Active				and when a second		~~~~	a l

- b) Go to the Macro Variable tab section.
- c) Delete the variables (*pair\_group*, *pair\_virtual\_ip*, *pair\_hello*, and *pair\_hold*).
- d) Click **OK** to save the modifications and close the window.

Intent Qualification	b Macro Variable	Critical	/ariable	Advance Sett	ings		
Items + Device Variable	e + Command Variable	•					
▲ Seed Device	Macro Variable	Source	Туре	Default Value	Look up Data for Device		
▲ 🖉 US-BOS-SW1	<i>(</i>						>
	😼 pair_group	Device	string	0	Used Default Value	~	🔺 📬 🔨
	😡 pair_virtual_ip	Device	string	0	Used Default Value	~	
	😡 pair_hello	Device	number	0	Used Default Value	~	
	😡 pair_hold	Device	number	0	Used Default Value	~	
Delete all t	he						
macro variable							

4. Click Edit Diagnosis to open the parser.

Network Intent (Edit Mode)			×				
I HSRP Pair failover check_parent	2	Run with Live Data	Save 🛛 Help 😑				
HSRP route failure check			💦 Intent Map: Select 🗸				
I Seed Logic		<b>\</b>					
🗃 + Device		Intent Variables: N	Manager   Tag: + Add 🗐				
V 🛹 US-BOS-SW1	Type Description here	+ Add Config Diagnosis	Image: Add CLI Diagnosis ≡				
▲ 💷 show standby 🕃	Type Description here		4 Edit Diagnosis ≡				
1 US-BOS-SW1>show standby		El Standhu davia					
2 Vlan100 - Group 1 (version 2)		Standby device availability 💼					
<ul> <li>3 State is Standby</li> <li>4 1 state change, last state change 31</li> </ul>	wld	Q Follow-ups					
5 Virtual IP address is 10.8.1.1	wid						
6 Active virtual MAC address is 0000.0c9	f.f001 (MAC Not In Use)						
7 Local virtual MAC address is 0000.0c							
8 Hello time 5 sec, hold time 15 sec							
9 Next hello sent in 0.608 secs							
10 Preemption enabled 11 Active router is 10.8.1.3, priority 10	E (avainag in 12 202 cas)						
11 Active router is 10.8.1.3, priority 10 12 MAC address is aabb.cc80.1500	5 (expires in 15.592 sec)						
13 Standby router is local							
14 Priority 90 (configured 90)							
15 Group name is "hsrp-Vl100-1" (default)	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~				

- 5. In the parser, add a column, **backup\_ip device name**, to the parser output table by converting the **backup\_ip** to the device name using the function call:
  - a) Go to **All Intent Variables** > **Intent Variable** >**HSRP\_failover table**, click on the drop down menu, and choose **Add Formula Column**.

CLI Command Diagnosi	s									
🧾 US-BOS-SW1	show standby	Intent	Variables fo	or Seed Logic			,			;
1. Define Variable			Intent Vari	iable Us	e Automation Data Tabl	e Task V	ariable			
Format1	<ul><li>✓</li><li>+</li></ul>	+ Add	l Formula C	olumn						
Ouble-click a variant	able to parse. Select	ÞI	Intent							
Current Device	✓ 07/2	4 🥏	US-BOS-S	W1		cu show stand	dby			
3 State	- Group 1 (vers is Standby	1	HSRP_f	failover						×
5 Virtua	ate change, las l IP address is		interface	(strin state (stri	ng) change_time (	. virtual_ip ( 🞚	Standby_route	group	ormula Co	lumn
	virtual MAC ad l virtual MAC d		Vlan100	Standby	31w1d	10.8.1.1		hsrp- Refree	sh	
8 Hello	time <mark>5</mark> sec, hol		Vlan100	Active	31w1d	10.8.1.8	10.8.1.3	hsrp-VI100-2	5	15
	hello sent in tion enabled		Vlan101	Add Formula Co	lumn			× -1	5	15
	router is 10.8 address is aabb									
	y router is loc			Name:						
	ty 90 (configur									
	name is " <mark>hsrp-</mark> - Group 2 (vers			Type:	string			× )		
	is Active									
	ate changes, lo		US-BOS-S	Initial Value:						
	l IP address is virtual MAC ad									
	l virtual MAC d	1		Definition:	e.g. \$bgp_conf	ig + \$ospf_co	onfig			
	time 5 sec, 🛌				+ 🖾 Variable 🛛 + 🖪 Fu	Inction				
	hello sent in									
25	- /		1	Help			Cancel	ок		
	-			- Help			Cancer			
🕜 Help   All Inte	nt Variables								C	lose

- b) In the Add Formula Column dialog, add the name Standby\_host\_name and Type is string.
- c) Definition: Choose the function **IpToHostname** and variable (**\$Standby\_router**).

Add F	ormula Co	lumn	×
	Name:	Standby_host_name	
b	Type:	string	~ ,
Init	ial Value:		
D	efinition:	IPToHostname(\$Standby_router)	
		+ 🖾 Variable + 🚺 Function	
<b>?</b> +	lelp	Cancel OK	d

NOTE: As you type IP in the **Definition** field, the list of functions with IP will appear as a list. Choose the function **IpToHostname.** 

Add Formula Co	lumn	×	
Name:	Standby_host_name		Task
Туре:	string	~	
Initial Value:		an sho	w sta
Definition:	<u>מו</u>		
	+ 🖾 🕞 IPNumberToString		
	IPStringToNumber	te gro	pup_r
🕜 Help	IPToHostname	IPToHostname	
	IPToInterfaceName	string IPToHostname (string ip)	
Next hello Preemption e Active route	nabl	Convert an IP address to the hostname	

- d) Click **OK** to save and close the dialog.
- e) Close the Intent Variables window.
- 6. Refine the parser variable **standby\_router** further to remove "," from the ip to use the functionality **IpToHostname**:

	2. Define Diagnosis				
{				Test on Device	s: 0 🔳
Variable (0)	- ♣ HSRP_failover ∨	者 Type: Paragra	ph 👔 🛛 + New Patter	n ~	Ē
• •	4 1	state change, last s	state change 31w1d	Add Parent	<u>i</u>
e A				Add Start Line	
		Add End Line	-		
<pre>}</pre>	Var Line 3 Virtua	l IP address i	s \$virtual_ir	Define Replacemer	nt
<b>\$</b>	5 Vi	rtual IP address is 1		Comparison Settin	gs
2				End of Paragraph	
		routor is to	tandhu soutos	Rename	
Remo	ve the "," from the		<u> </u>	Delete	
parsei	r variable using	\$virtual_ip 🕸 🗸	\$Standby_ro… ∨	Insert Pattern	>
Define	e Repalcement	- 1	local	hsrp-Vl100-1	5
2	31w1d	10.8.1.8	10.8.1.3,	hsrp-Vl100-2	5
8	31w1d	10.8.1.17	local	hsrp-VI101-1	5
ē.			~		

- a) Select **Define Replacement** from  $\equiv$  menu to the corresponding dialog.
- b) Click +Add Row.
- c) In the field **Find What**, enter the text "(\d+.\d+.\d+),". We are constructing the variable for "ip 10.8.1.3," inside the braces.
- d) Check the selection box under **Regular Expression**.
- e) In the field **Replace With**, enter the text "**\$1**".
  - NOTE: **Replace With** is to replace the expression defined in the field **Find what**. Here, **\$1** stands for variable 1, and it will replace the expression defined inside the braces of the **Find What** field.
- f) Click **OK** to save the settings and close the window.

	+ Add Row					😭 Import	₿ E
ind Wh	nat	Match Whole Word	Match Case	Regular Expres	sion Replace With	Replace Mat	hes
\d+.\d+	+.\d+.\d+), <b>C</b>				d \$1 e	All	~
	Original Text US-B0S-SW1>show standby				placement Text 5-BOS-SW1>show stand		reviev
	Vlan100 - Group 1 (version 2)			2 V1	an100 - Group 1 (ve		
3	State is Standby				State is Standby		
	1 state change, last state of Virtual IP address is 10.8.1.1			4 5	Virtual IP address	ast state change 31w1d is 10 8 1 1	
4 5		0000.0c9f.f001 (M/	AC Not In Us			address is 0000.0c9f.f001 (MAC Not I	ı Use
+ 5 6	ACLIVE VIPLUAL MAC ADDRESS IS			7		address is 0000.0c9f.f001 (v2 defau	lt)
5	Local virtual MAC address is			8	Hello time 5 sec, h		
5 6 7 8	Local virtual MAC address is Hello time 5 sec, hold time 15	sec					
5 6 7 8 9	Local virtual MAC address is Hello time 5 sec, hold time 15 Next hello sent in 0.608 sec	sec		9	Next hello sent i	n 0.008 secs	
5 6 7 8 9 10	Local virtual MAC address is Hello time 5 sec, hold time 15 Next hello sent in 0.608 sec Preemption enabled	sec s	in 13 392	10	Preemption enabled		97 sa
5 6 7 8	Local virtual MAC address is Hello time 5 sec, hold time 15 Next hello sent in 0.608 sec	5 sec cs .ority 105 (expires	s in 13.392	10	Preemption enabled	.8.1.3 priority 105 (expires in 13.3	92 se
5 6 7 8 9 10 11	Local virtual MAC address is Hello time 5 sec, hold time 15 Next hello sent in 0.608 sec Preemption enabled Active router is 10.8.1.3, pri	5 sec cs .ority 105 (expires	s in 13.392	10 se( 11 12	Preemption enabled Active router is 10	.8.1.3 priority 105 (expires in 13.3 bb.cc80.1500	92 se

g) The final output of the variable *standby\_router* will be:

schange_time 💙	\$virtual_ip 🐺 🗸	\$Standby_ro… ∨	\$group_name ∨	\$hello_time 💙
1w1d	10.8.1.1	local	hsrp-Vl100-1	5
31w1d	10.8.1.8	10.8.1.3	hsrp-Vl100-2	5
1w1d	10.8.1.17	local	hsrp-Vl101-1	5
		·>		

- 7. Back in the diagnosis window, go to the **Define Diagnosis** ribbon, delete the diagnosis created for follow-up intent, and create a new diagnosis as follows:
  - a) Name: Standby device check.
  - b) Anchor: Standby\_router.
  - c) Check the selection box of **Loop Table Rows** and select the table **HSRP\_failover**.
  - d) Table Key: Select the variable *virtual\_ip*.
  - e) **If** condition: *Standby\_router* | Is not empty.

{	ne Diagnosis			
{ 📄 Add N	ote D Add Diagnosis	Can also	click a varia	ble on the left to add automatior
} Name:	Standby device check		Anchor:	HSRP_failover.\$Stan
}	Type description of the diagnosis			
↓ ↓ Loop ↓ ↓ If	Table Rows 📜 HSRP_failover 🛇	<ul> <li>Table Key: vir</li> </ul>	tual_ip	<b>d</b>
A	🥣 US-BOS-SW1 🛛 Current 🗸			
<b>e</b>	Standby_host_name 🗸 I	s not empty	~ <mark>,</mark>	Î
В	Select Variable 🗸 🗸			
Lunn				

f) Then: Remove the Diagnosis Message and add a follow-up action from the Add logic dropdown > Follow-up Intent.

Name:	Standby device availa	bility	Anchor:	HSRP_failover.\$Stan	~
	Type description of the	diagnosis			
	e Diagnosis Message Data View	_failover 🗸 Table Ke	y: virtual_ip	~	0
Draw		ent 🗸			
Send I	Email	✓ Is not empty	~	ធ	1
Follow	-up Intent 🛛 💮				
Set Int	tent Baselire	~			
Advan	iced >				
Add Log	ic ~				
		s ,≣ HSRP_failover ∨		virtual_ip ~	

ĭ

Current Intent (Self)

Stop

B

∨ Then

Select Variable

Q Follow-up Intent: ○ Network Intent f

≡

- g) Click the **Network Intent** link to choose the follow-up intent in a new dialog **Follow-up Intents**.
  - i. In the Follow-up Intents window, choose **Select Intent Template**.
  - ii. In the Select Intents dialog, navigate to the follow-up intent (*HSRP pair failover check\_followup*) and check the selection box.
  - iii. Click **OK** to save the selection and close the window.

+ Follow-up 🗸 🚯	
Select Intent (Standalone)	
Select Intent Template	Select Intent Templates
Select Intents from ADT Select Intent Templates via ADT Select Intent Cluster	Select Intent Template from:      All Intents     Installed Intents
Select Intent Cluster	Type:     Common Intent     ~     Filter by:     no filter     ~     hsrp     ×
	All       10.10.10.1 (6)       192.168.29.62 (4)       3Com (5)       5 Report (2)       AAA (5)       Acces       >         Image: Algest in the second se
	Arun Belide     Arun Belide     II HSRP Pair failover check_followup     II HSRP Pair failover check_parent
	I HSRP Status Check  Assessment Reference Library  Figure Fault Tolerance Assessment
	Image: Second Seco
	O4 - Failover      Biel
	If HSRP Track and Preemption Configuration Check-Colector
	Cookbook

iv. In the Replication Settings:

Set the **Replicate Current Intent to**: **Device by Variable** and choose the variable *Standby\_host\_name*.

- v. In the **Set Macro Variables of Seed Intent Template**: set the variables for macro variables as follows:
- vi. Click **Save** to save and close the follow-up intents.

Follo	w-up Intents								×
1 Fo	ollow-up Inter	nt: + Follow-up	× <b>(</b> )						
~	I HSRP Pai	ir failover check_fol	lowup (Intent Template)					Ī	
	Description:	When							
	Replicati	on Settings:							
iv	Replicate	Current Intent to:	Device by Variable	~	Standby_h	ost_name (HSRP_failover)	~		
	Set Macro	o Variables of Seed I	ntent Template:						
	_ Se	eed Device	Macro Variable	Туре		Set Variable			
	4	US-BOS-SW1							
			pair_group	string		group_name (HSRP_failover)	~		
		V	pair_virtual_ip	string		virtual_ip (HSRP_failover)	~		
			pair_hello	numbe	r	hello_time (HSRP_failover)	~		
			pair_hold	numbe	r	hold_time (HSRP_failover)	~		
	🗌 Merge	multiple replicated	intents into one			1/1 Device Key Set 1	or Selected Table		
	Follow-up	Execution: Setting	5						
Ē	Prune other fo	ollow-up intents					Car vi S	ave	li

h) Back in the **Define Diagnosis** ribbon, add an **Elself** condition for the unknown standby devices:

A: *Standby\_router* | Contains | Unknown

- i) **Then**: Define the diagnosis message: *\$this device* does not have standby HSRP.
- j) Check the boxes next to Set Status Code for Device and Set Status Code for Intent.
- k) Click **Apply** to save and close the diagnosis window.

∨ ElseI	f						<b>a</b>	Delete
A	🥏 US-E	BOS-S Cu	rrent 🗸					
	Stand	by_router	~	Contains	~	unknown	~	Ē
в	Select	Variable	~					
√ Then	1							
📄 Dia	gnosis N	lessage:				🗆 S	ave to Inciden	nt ≡
	~	\$this_device	do not h	ave standby HS	RP 🕕			
2-5	Set Stat	tus Code for De	/ice:					
	Error	∽ \$this_d	evice do	not have stand	by HSRP			
<mark>-</mark> 5	Set Stat	tus Code for Inte	ent:					
e	Error	∽ \$this_d	evice do	not have stand	by HSRP			
+ Add	Elself	+ Add Else						
							Cancek	Apply

- 8. In the Network Intent (Edit Mode) dialog, click **Save**.
- 9. Click **Run** to execute the intent.

Network Intent (Edit Mode)					×	
I HSRP Pair failover check_parent	Diagnosis Tree	9 Run	with Live Data	8 Save 8	Help ☰	
HSRP route failure check				💦 Intent Map: S	Select 🧹	
I Seed Logic	gic 🤣					
Intent Variables: Manager     Tag: + Add Imager						
✓			g Diagnosis	+ Add CLI Diagnosis	≡	
🔺 🖽 show standby ਓ	Type Description here			Edit Diagnosi	s ≡	
1 JS-BOS-SW1>show standby			D Standby d	evice availability 🛛 🖬		
2 Vlan100 - Group 1 (version 2	2)					
3 State is Standby 4 1 state change, last sta	to change 21w1d		😡 Follow-up	5		
4 1 state change, last sta 5 Virtual IP address is 10.8						
6 Active virtual MAC address		AC Not In Use)				
7 Local virtual MAC addres	-					
8 Hello time 5 sec, hold tim	e 15 sec		~~~~			

10. Open the diagnosis tree to see the results.

Network Intent (View Mode) - All Network Intents/0_Power User 2 Sandbox/HSRP Pair failover check_parent	×
I HSRP Pair failover check_parent HSRP route failure check	∠Edit ≡
Result: 08/02/2024 01:59 PM 10 at 08/02/2024 01:59 PM with 0 errors. You can View Execution Log	with Live Data
<ul> <li>US-BOS-SW1 and US-BOS-SW2 hello and hold time is same</li> <li>US-BOS-SW1 1 Diagnosis</li> </ul>	View ~
Diagnosis Tree of HSRP Pair failover check_parent	×
Source: 💄 ArunKumar.Belide@netbraintech Current NI: 💶 HSRP Pair failover check_pa Execution Time: 08/02/2024 01:59:30 PM 🗸   🙆 View Se	ttings   🚠 Auto Layout   😋 🔳
Pre-Execution Post-Execution	Untitled Diagnosis 1 C show standby Untitled Diagnosis 2 C show standby
Network Intent Details - HSRP Pair failover check_parent	17 /

## 8.3.3 Replicate the intents for all HSRP devices using ADT

In this section, you will use the **Intent Replication Wizard** to replicate the intent to all devices in the HSRP device group. As a first step, open the intent from the intent manager and the Intent replication wizard as shown:

Network Intent Manager > Common Map Intent	
Common Map Intent	
Search	
All	
🔺 📄 Desktop	
I Ping Cisco	
I Route Check (Cisco IOS)	
🔺 📄 Public	
🔺 🛅 cookbook	
I HSRP Status Check Open	
🔺 📄 Jun Liang	
Network Intent (View Mode) - Common Map/Public/cookbook/HSRP Status Check	×
I HSRP Status Check HSRP route failure check	"Copen <u>⊫o</u> "∆o <u>∧</u> 3 =
	Named Tag
🖌 Result: No History Data 👔 💿 🔛	View Abstract
	Run Settings
No result available because this intent has not been executed.	Data Clean Settings
	Edit
	Save as
V 🛹 US-BOS-SW1 1 Diagnosis	Delete
	Refresh
▲ 💷 show standby 1 Diagnosis	Export
1 US-BOS-SW1>show standby	Share to Incident
2 Vlan100 - Group 1 (version 2)	Publish Intent
3 State is Standby	
4 1 state change, last state change 31w1d 5 Virtual IP address is 10.8.1.1	Intent Replication Wizard
6 Active virtual MAC address is 0000.0c9f.f001 (MAC Not In	Auto Intent Wizard
7 Local virtual MAC address is 0000.0c9f.f001 (v2 default 8 Hello time 5 sec, hold time 15 sec	
9 Next hello sent in 0.608 secs	
10 Preemption enabled	
Active router is 10.8.1.3, priority 105 (expires in 13.39 MAC address is aabb.cc80.1500	
13 Standby router is local	

- 1. In the first step (**Seed Intent**), Validate the default seed intent and go to the next section.
- 2. In the second step (**Define ADT**), provide the basic input to create a new ADT, such as name, ADT location and target devices.

- 3. In the third step, define the **intent qualification** with device group **HSRP Devices** to include devices for the seed intent to replicate on.
- 4. In the fourth step, **Replicate Intent**, modify the name of the **Replicated Intent** In the **ADT Columns** section and add more columns that you want in the final ADT.

Seed Intent	Define ADT	Replication Settings	Replicate Intent	
ADT Columns:			Additional Columns ~	
Column Data	Column Name	Тад	Replicated Intent	
Replicated Intent	HSRP failover	0 tag	s 📄 Intent Message	
			🔽 In <mark>tent Status Code</mark>	
Modify the na	me		Cevice Status Code	
intent column			C Intent Devices	
			🗌 In <mark>t</mark> ent Map	
	Check th	e boxes of all the	🔲 l <mark>i</mark> tent CLI Comma	
	columns that you want see in			

5. Click **Save and Replicate** to save all the settings and create ADT. An option **Open Output ADT** will appear. Click it to open the table in the **ADT Manager**.

	Define ADT Replication Settings	Replicate Intent
T Columns:		Additional Columns ~
Column Data	Column Name	Тад
Replicated Intent	HSRP failover	0 tags
s Intent Message	Intent Message	
🔒 Intent Devices	Intent Devices	
S Last Execution Time	Last Execution Time	
		5 Save and Replicate

	Define ADT Replication Settings	Replicate Intent
DT Columns:		Additional Columns N
Column Data	Column Name	Tag
Replicated Intent	HSRP failover	0 tags
s Intent Message	Intent Message	
육 Intent Devices	Intent Devices	
Clast Execution Time	Last Execution Time	
	Click to open the table in ADT manager.	Save and Replicate

#### Review the new Intent columns that are added to the table and results.

Search 🔍 😋 <						
Shared Tables (1201)	🖏 HS	RP Route Failure	Table Builder	Last Updat	ed at: 07/30/2024 12:57 PM 🛛 🔍 Rebuild Tab	ole Add Data Manually $\lor \equiv $
My Tables (5)	Descri	ption: Type description here				
Ping (1)	Items:	19 Rows 4 Columns			Search.	
BGP Config Change Diagnosis						
<ul> <li>Day 1 Lab</li> <li>HSRP Route Failure</li> </ul>	No.	Device	HSRP failover		s Intent Status Code	() Last Execution Time
I NTP	1	BJ-L2-coreB	HSRP Status Check BJ-L2-coreB	0	BJ-L2-coreB HSRP hsrp-VI10-100 is stable	07/29/2024 03:55:07 PM
	2	BJ_L2_Core_3	HSRP Status Check BJ_L2_Core_3	0	BJ_L2_Core_3 HSRP hsrp-Fa1/0/9-10 is stab	07/29/2024 03:55:07 PM
	3	BJ_L2_Core_4	HSRP Status Check BJ_L2_Core_4	0	BJ_L2_Core_4 HSRP hsrp-Fa2/0/3-0 is stable	07/29/2024 03:55:05 PM
	4	BJ_core_3550	HSRP Status Check BJ_core_3550	0	BJ_core_3550 HSRP None is stable	07/29/2024 03:55:07 PM
	5	Bur-isp-gw1	HSRP Status Check Bur-isp-gw1	0	Bur-isp-gw1 HSRP hsrp-Gi0/0/0-200 is stab	le 07/29/2024 03:55:07 PM
	6	IPv6Lab-SW8	HSRP Status Check IPv6Lab-SW8	0	IPv6Lab-SW8 HSRP hsrp-Et0/1-1 is stable	07/29/2024 03:55:05 PM
	7	IPv6Lab-SW9	HSRP Status Check IPv6Lab-SW9	0	IPv6Lab-SW9 HSRP hsrp-Et0/1-1 is stable	07/29/2024 03:55:07 PM
	8	PE-3600X-01	HSRP Status Check PE-3600X-01	0	PE-3600X-01 HSRP hsrp-Gi0/24-2 is stable	07/29/2024 03:55:07 PM
	9	PE-3600X-02	HSRP Status Check PE-3600X-02	0	PE-3600X-02 HSRP hsrp-Gi0/13-2 is stable	07/29/2024 03:55:07 PM
	10	PE-ASR1K-01	HSRP Status Check PE-ASR1K-01	0	PE-ASR1K-01 HSRP hsrp-Te0/0/0-15 is stabl	e 07/29/2024 03:55:05 PM
	11	PE-ASR1K-02	HSRP Status Check PE-ASR1K-02	0	PE-ASR1K-02 HSRP hsrp-Gi0/0/5-10 is stabl	e 07/29/2024 03:55:05 PM
	12	Sjc-Dist-3750-02	HSRP Status Check Sjc-Dist-3750-02	0	Sjc-Dist-3750-02 HSRP hsrp-Vl30-0 is stable	07/29/2024 03:55:07 PM
	13	US-BOS-SW1	HSRP Status Check US-BOS-SW1	0	US-BOS-SW1 HSRP hsrp-Vl100-1 is stable	07/29/2024 03:55:07 PM
	14	US-BOS-SW2	HSRP Status Check US-BOS-SW2	0	US-BOS-SW2 HSRP hsrp-Vl100-1 is stable	07/29/2024 03:55:05 PM
	15	bjta002237-SW2	HSRP Status Check bjta002237-SW2	0	bjta002237-SW2 HSRP hsrp-Vl481-1 is stab	le 07/29/2024 03:55:07 PM
	16	bjta002238-SW3	HSRP Status Check bjta002238-SW3	0	bjta002238-SW3 HSRP hsrp-Vl481-1 is stab	le 07/29/2024 03:55:05 PM
	17	bjta002444-SW13	HSRP Status Check bjta002444-SW13	0	bjta002444-SW13 HSRP hsrp-Et1/1-1 is stat	07/29/2024 03:55:05 PM
	18	bur-isp-gw2	HSRP Status Check bur-isp-gw2	0	bur-isp-gw2 HSRP hsrp-Gi0/0/0-200 is stabl	
	19	gapp-c3560-2	HSRP Status Check gapp-c3560-2	O	qapp-c3560-2 HSRP hsrp-Gi0/15-0 is stable	

# 8.3.4 Replicate the intent to all HSRP devices

Replicate the intent to all HSRP devices using the **Intent Replication Wizard** from the network intent window. In the intent replication wizard, input the seed intent and target devices as follows:

- 1. In the first step (**Seed Intent**), Validate the default seed intent and go to the next section.
- 2. In the second step (**Define ADT**), provide the basic input to create a new ADT, such as name (HSRP Pair failover), ADT location and target device group.
- 3. In the third step, define the **intent qualification** with device group **HSRP Devices** to include devices for the seed intent to replicate on.
- In the fourth step, Replicate Intent, modify the name of the Replicated Intent In the ADT Columns section and add other columns like Intent Status Code and Last Execution Time as needed in the final ADT.

	Define ADT	Replication Settings	Replicate Intent
ADT Columns:			Additional Columns ~
Column Data	Column Name	Тад	Replicated Intent
Replicated Intent	Replicated Intent	0 tags	Intent Message
S Intent Status Code	Intent Status Code		D Intent Status Cod
() Last Execution Time	Last Execution Time	You can edit the column	Device Status Coo
K	·/	names as required	Intent Devices
			Intent CLI Comma
			☐ Intern CEI Comma
		Columns will	
		appear on left as	
		appear on left as	
		appear on left as	
		appear on left as	✓ Lest Execution Tir
		appear on left as	

5. Select **Save and Replicate** to save all the settings and create ADT. An option **Open Output ADT** will appear. Click it to open the table in the **ADT Manager**.

Seed Intent	Define ADT	Replication Settings	Replicate Intent
DT Columns:			Additional Colu
Column Data	Column Name	Tag	
Replicated Intent	HSRP Pair failover check	0 tags	
s Intent Status Code	Intent Status Code		
🕔 Last Execution Time	Last Execution Time		
	AD	ck the link to open T in new window Request submitted at: 08/02/2024	Save and Replicate

#### Review the new Intent columns that are added to the table and results.

Search Q C «	🚯 HS	RP Route Failure	Table Builder	Last Updat	ed at: 07/30/2024 12:57 PM 🛛 🍕 Rebuild Table	Add Data Manually 🗸 📃
Shared Tables (1201) My Tables (5)	Descri	ption: Type description here				
<ul> <li>Ping (1)</li> <li>BGP Config Change Diagnosis</li> </ul>	Items:	19 Rows 4 Columns			Search	Q T Advanced Filter: Undefined
👪 Day 1 Lab	No.	Device	HSRP failover		S Intent Status Code	() Last Execution Time
HSRP Route Failure	1	BJ-L2-coreB	HSRP Status Check BJ-L2-coreB	0	BJ-L2-coreB HSRP hsrp-VI10-100 is stable	07/29/2024 03:55:07 PM
III NTP	2	BJ_L2_Core_3	HSRP Status Check BJ_L2_Core_3	0	BJ_L2_Core_3 HSRP hsrp-Fa1/0/9-10 is stable	07/29/2024 03:55:07 PM
	3	BJ_L2_Core_4	HSRP Status Check BJ_L2_Core_4	0	BJ_L2_Core_4 HSRP hsrp-Fa2/0/3-0 is stable	07/29/2024 03:55:05 PM
	4	BJ_core_3550	HSRP Status Check BJ_core_3550	0	BJ_core_3550 HSRP None is stable	07/29/2024 03:55:07 PM
	5	Bur-isp-gw1	HSRP Status Check Bur-isp-gw1	0	Bur-isp-gw1 HSRP hsrp-Gi0/0/0-200 is stable	07/29/2024 03:55:07 PM
	6	IPv6Lab-SW8	HSRP Status Check IPv6Lab-SW8	0	IPv6Lab-SW8 HSRP hsrp-Et0/1-1 is stable	07/29/2024 03:55:05 PM
	7	IPv6Lab-SW9	HSRP Status Check IPv6Lab-SW9	0	IPv6Lab-SW9 HSRP hsrp-Et0/1-1 is stable	07/29/2024 03:55:07 PM
	8	PE-3600X-01	HSRP Status Check PE-3600X-01	0	PE-3600X-01 HSRP hsrp-Gi0/24-2 is stable	07/29/2024 03:55:07 PM
	9	PE-3600X-02	HSRP Status Check PE-3600X-02	•	PE-3600X-02 HSRP hsrp-Gi0/13-2 is stable	07/29/2024 03:55:07 PM
	10	PE-ASR1K-01	HSRP Status Check PE-ASR1K-01	0	PE-ASR1K-01 HSRP hsrp-Te0/0/0-15 is stable	07/29/2024 03:55:05 PM
	11	PE-ASR1K-02	HSRP Status Check PE-ASR1K-02	0	PE-ASR1K-02 HSRP hsrp-Gi0/0/5-10 is stable	07/29/2024 03:55:05 PM
	12	Sjc-Dist-3750-02	HSRP Status Check Sjc-Dist-3750-02	0	Sjc-Dist-3750-02 HSRP hsrp-Vl30-0 is stable	07/29/2024 03:55:07 PM
	13	US-BOS-SW1	HSRP Status Check US-BOS-SW1	0	US-BOS-SW1 HSRP hsrp-VI100-1 is stable	07/29/2024 03:55:07 PM
	14	US-BOS-SW2	HSRP Status Check US-BOS-SW2	0	US-BOS-SW2 HSRP hsrp-VI100-1 is stable	07/29/2024 03:55:05 PM
	15	bjta002237-SW2	HSRP Status Check bjta002237-SW2	0	bjta002237-SW2 HSRP hsrp-Vl481-1 is stable	07/29/2024 03:55:07 PM
	16	bjta002238-SW3	HSRP Status Check bjta002238-SW3	0	bjta002238-SW3 HSRP hsrp-Vl481-1 is stable	07/29/2024 03:55:05 PM
	17	bjta002444-SW13	HSRP Status Check bjta002444-SW13	0	bjta002444-SW13 HSRP hsrp-Et1/1-1 is stable	07/29/2024 03:55:05 PM
	18	bur-isp-gw2	HSRP Status Check bur-isp-gw2	0	bur-isp-gw2 HSRP hsrp-Gi0/0/0-200 is stable	07/29/2024 03:55:07 PM
	19	qapp-c3560-2	HSRP Status Check gapp-c3560-2	0	qapp-c3560-2 HSRP hsrp-Gi0/15-0 is stable	07/29/2024 03:55:07 PM

#### 8.3.5 View and Run the intents in ADT

Navigate to the ADT you have just created, and we shall run the intent as follows:

- 1. Go to the Intents column, hover the mouse on the column header, and click **Run**.
- 2. Click **Rebuild** to refresh the results in the **Intent Status code**.

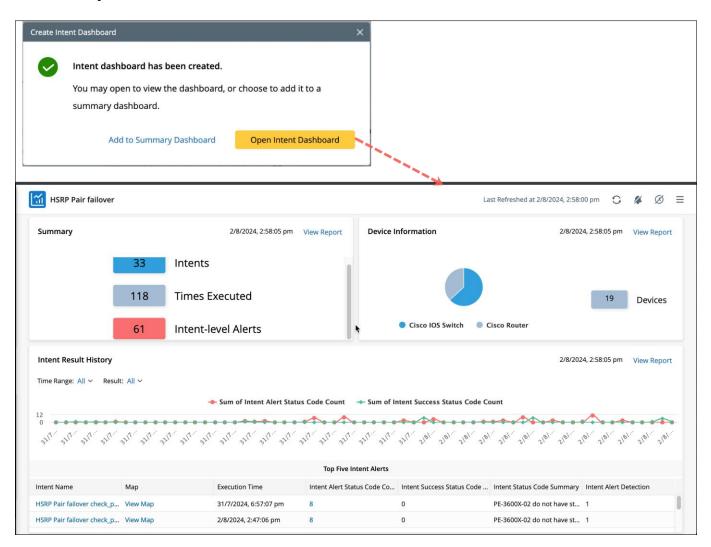
earch	Q	<mark>℃</mark> ≪	ш	RP Pair failover	Table Builder	ast Undated at: (	08/02/2024 03:36 P	Rebuild Table	
Shared Tables (1208)			а <u>0</u> пр	Image: HSRP Pair failover     Table Builder     Last Updated at: 08/02/2024 03:36 Pt     Rebuild Table					
📄 My Tables (6)			Descri	ption: Type description here					
Ping (1)			Items:	19 Rows 4 Columns				Search	
BGP Config Change I	Diagnosis								
👪 Day 1 Lab			No.	Device	I HSRP Pair failover c	Details	s Intent Status Code		
HSRP Pair failover			1	BJ-L2-coreB	HSRP Pair failover check_parent BJ-L2-co	reB 💿	BJ-L2-coreB do not have sta	andby HSRP	
<ul> <li>明 HSRP Route Failure</li> <li>明 NTP</li> </ul>	Route Failure		2	BJ_L2_Core_3	HSRP Pair failover check_parent BJ_L2_Co	ore_3 💿	BJ_L2_Core_3 do not have s	tandby HSF	
		3	BJ_L2_Core_4	HSRP Pair failover check_parent BJ_L2_Co	ore_4 💿	BJ_L2_Core_4 do not have s	tandby HSI		
				4	BJ_core_3550	HSRP Pair failover check_parent BJ_core_	3550 💿	BJ_core_3550 do not have s	tandby HSI
			5	Bur-isp-gw1	HSRP Pair failover check_parent Bur-isp-	gw1 💿	Bur-isp-gw1 do not have st	andby HSR	
			6	IPv6Lab-SW8	HSRP Pair failover check_parent IPv6Lab	-SW8 💿	IPv6Lab-SW8 do not have s	tandby HSF	
			7	IPv6Lab-SW9	HSRP Pair failover check_parent IPv6Lab	-SW9 💿			
			8	PE-3600X-01	HSRP Pair failover check_parent PE-3600	X-01 💿	PE-3600X-01 do not have st	andby HSR	
			9	PE-3600X-02	HSRP Pair failover check_parent PE-3600	X-02 💿	PE-3600X-02 do not have st	andby HSR	
			10	PE-ASR1K-01	HSRP Pair failover check_parent PE-ASR1	K-01 😐	PE-ASR1K-01 do not have s	tandby HSF	
			11	PE-ASR1K-02	HSRP Pair failover check_parent PE-ASR1	к-02 💿	PE-ASR1K-02 do not have s	tandby HSI	
			12	Sjc-Dist-3750-02	HSRP Pair failover check_parent Sjc-Dist-	375 💿	Sjc-Dist-3750-02 do not hav	e standby	
			13	US-BOS-SW1	HSRP Pair failover check_parent US-BOS-	-SW1 💿	US-BOS-SW1 do not have s	tandby HSI	
			14	US-BOS-SW2	HSRP Pair failover check_parent US-BOS-	SW2 💿	US-BOS-SW2 do not have s	tandby HSI	
			15	bjta002237-SW2	HSRP Pair failover check parent bita0022	237 💿	bjta002237-SW2 do not hav	e standby	

## 8.3.6 Create the Dashboard

Let us create the intent dashboard from the ADT. Open the **New Intent Dashboard** from the intent HSRP Failover column  $\equiv$  menu.

	Q         ○         ≪           Tables (1208)         Ins (f)	B HSRP Pair failover	ere	Table Builder	Last Upda	
▲ 📄 My Tabl ▷ 📄 Ping 職 BGP		Items: 19 Rows 4 Columns				
e Intent Dashbo	ard	×	I HSRP P	I HSRP Pair failover check		
			HSRP Pai	Run Intents Once		
Create Inten	t Dashboard for ADT 'HSRP Pair fa	ilover'	HSRP Pai	Run Intents via Timer		
			HSRP Pai	i Open Seed Intent		
Name:	HSRP Pair failover		HSRP Pai Rebuild In		uild Intent-related Column Grou	
			HSRP Pai	Remove Empty Wrapper	Intent	
Location:	My Dashboards 🗸 4		) HSRP Pai	Enable Auto Intent		
🗌 Use Templa	te		HSRP Pai	Export Diagnosis Result	to CSV	
			HSRP Pai	View Summary Report		
Data Source	Automation Data Table	~	(	Export Intent Output Ma	р	
			HSRP Pai	Debug Empty Cells		
Automation Da	ta Table: HSRP Pair failover		HSRP Pai	Tag Current Column		
Include Trig	gered Follow-up Intent Results		HSRP Pai			
			HSRP Pai	Edit Delete		
Intent Column:	HSRP Pair failover check 🛩		HSRP Pai	Set as Table Key		
🔲 Filter Intent	by Devices		HSNP Pai			
_	,		HSRP Pa	Submit Related Commar		
			HSRP Pai	New Intent Dashboard	2	
			MALE AND	failover.check.pareot.bits	002444	
Time Range	Last 7 Days	~				
	-					

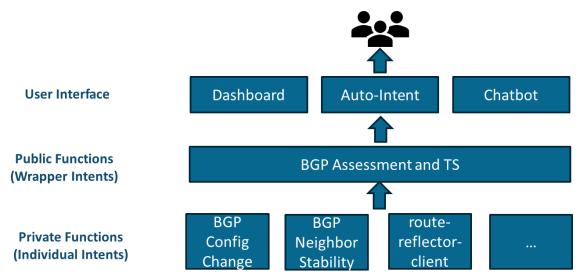
In the **Create Intent Dashboard** dialog, **Open Intent Dashboard** to view the results summary and result history.



# 9 Collaborative Troubleshooting

In Chapter 4, you learned how an end user can troubleshoot a network problem with the auto intent and Chabot. You learned how to use the follow up intent to encapsulate a set of intents into a wrapper intent, which hides the detailed intent implementation from the end user and can function as a public interface to the end user, which will be accessed as an auto intent or as a chatbot.

In this chapter, we will dive deeper into the troubleshooting. We will use the BGP as an example to illustrate how to apply intent-based automation to collaborative troubleshooting. You will create and include all BGP-related intents, such as checking BGP config change, BGP neighbor stability, and route-reflector-client, into a wrapper intent. They will be exposed to the end user interface, such as Dashboard, Auto-intent, and Chatbot.



The following are covered in this chapter:

- 1. <u>Document BGP Devices and Configurations</u>
- 2. <u>Create a wrapper intent for TS BGP</u>
- 3. Add more BGP troubleshooting intents
- 4. <u>Create a chatbot for the BGP TS wrapper intent</u>

# 9.1 Document BGP Devices and Configurations

In this section, you will create an ADT with the base table to include network BGP devices, AS number, Router id, and BGP configurations. This ADT base table can be used as the base to replicate any BGP-related intent and the foundation of troubleshooting the BGP issues. The ADT will be created by the pre-replicated intent with the following steps:

- 1. <u>Create a dynamic device group to include all network devices.</u>
- 2. <u>Create a seed intent to parse BGP AS numbers and configurations.</u>
- 3. Define the replication settings and install/decode it.
- 4. <u>Create an ADT, building the base table with built-in device properties and signature variables</u> (As number and BGP configurations).

### 9.1.1 Create BGP Device Group

Create a device group (BGP Devices) with the dynamic criteria using **Device Property** (BGP Enabling) and **Vendor** contains **Cisco**.

- 1. Click **Select Criteria > Device Property** and select **BGP Enabling** from the dropdown.
- 2. Add the condition **True**.
- 3. Click Search.
- 4. Click **OK** to save and close the dialog.

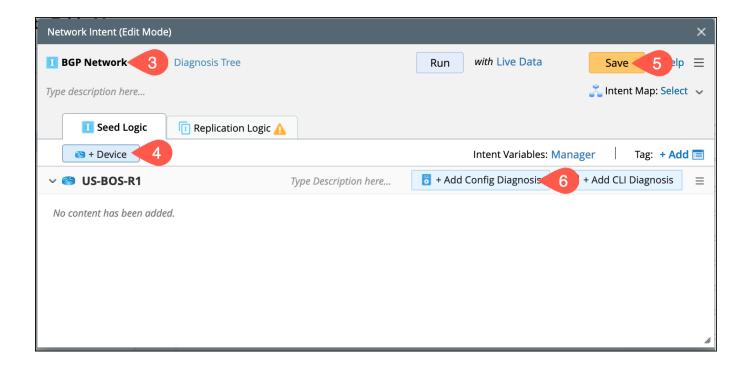
Dynami	ic Search Device				×
	All Devices	~			
A	Select Criteria BPE Enabling	✓ is	✓ True 2	~ [	<u>ال</u>
Boo	HA Enabling VPLS Enabling Multicast Enabling BGP Enabling MC-LAG Enabling QoS Enabling VXLAN Enabling IPv6 Enabling			Search	3
Hos	NAT Enabling	Vendor	Model	Management IP	
	R-R206-Forti200F-1	Fortinet	200F	192.168.0.100	
BUR	R-R206-Forti200F-1_(Cloud	Fortinet	200F	192.168.0.100	
BUR	R-R206-Forti200F-1_(Demo	Fortinet	200F	192.168.0.100	
Berl	lin-vEdge	Cisco	WS-C4500X-32	192.168.0.1	
Bur	-Netbond(Primary)(East-RG	Microsoft	Azure MSEE		
				Cancel	ок 4

# 9.1.2 Create an intent to parse BGP AS numbers and configuration

#### 9.1.2.1 Select a device

Define the basic information of the intent, such as name, description, and device and save it to the intent manager as follows:

- 1. Click 🕑 icon from your desktop > **New Intent.**
- 2. A window **Network Intent** in edit mode will appear.
- 3. Enter the title and a brief description (optional) of the intent.
- 4. Add the seed devices by selecting the option **+Device**. You should select a **Cisco** device with BGP configured.
- 5. Click **Save** to save the intent to the Intent Manager.
- 6. Click **+Add CLI Diagnosis** to open the corresponding window and proceed to the next section to parse the variables and define the diagnosis.



#### 9.1.2.2 Retrieve Data and Parse Variables

- 1. Click **Retrieve** to collect the data from Live Data.
- 2. Search for bgp data using keywords in the search bar.
- 3. Select the bgp text **65001** and click **Parse Variable** in the tip window

**Note**: You can also double-click the text to get the same result.

Configuration Diagnosis
Contraction of the second seco
1. Define Variable 2. Define Diagnosis
Format1 V +
Critical Variable (0)     Type: Single Mu
Current Device V 08/06/2024 04:09:03 PM bgp 2 X A V
229 area 51 range 10.0.0.0 255.255.0.0 230 redistribute bgp 65001 subnets route-map AWS-Ohio-M 231 network 10.8.1.0 0.0.0.255 area 0
232 default-information originate always 233 distribute-list prefix BOS-TOR-Traffic in
234 ! 235 router bgp 65001 3
236 bgp router id 1 Parse Variable
238 redistribute ospf 1 match internal external 1 exter 239 neighbor 6.7.8.9 remote-as 12345

4. In the right pane **Var Line 1** field, update the variable name to *router bgp \$int:bgp\_as*.

	2. Define Diagnosis Change the parser name to		
}	bgp using pen icon	Test on Devices: 0	≡
ariable (0)	🖬 Pattern1 🗸 Z Type: Single 🍘 🕂 New Pattern 🗸		=
<b>* *</b>	Var Line 1 router bgp \$int:bgp_as 4 235 router bgp 65001	≡ > 1 Line	l
	+ Field		
	Output     + Parse Lines     5       \$bgp_as (int)     =     65001		

- 5. To parse the router id, select text **10.10.10** and click **Parse Variable** in the tip window.
- 6. Parsing BGP configuration: To parse the multiple lines into one variable, use the function *LineByVariables*:
  - a) Click + Parse Lines.
  - b) Enter Variable name *bgp\_config*.
  - c) Check **The line between** the radio button and parse the lines between **\$bgp\_as** to **End**.
  - d) Review the **Pattern** *LinesByVariable[\$bgp\_config]:\$bgp\_as-* and click **Apply**.

Name: bgp_config     • he line of variable: sect     • he line of variable: bgp_as     • he line of vari	Parse Lines				×					
<ul> <li>The line of variable:</li> <li>Select</li> <li>The line between:</li> <li>Sbgp_as</li> <li>to</li> <li>End</li> <li>C</li> <li>The line contains keyword:</li> <li>Inter keyword</li> </ul> Outpute:   Supp_config=   router bgp 65300   bgp router-id 192.168.20.254   bgp log-neighbor-changes   redistribute eigrp 1 route-map E2B   neighbor 100.30.0.1 remote-as 10000   !   ip forward-protocol nd   Pattern: LinesByVariable[\$bgp_config]: \$bgp_as-	Name: bgp config	b								
<ul> <li>The line between: \$bgp_as to end to be and step configered by the line contains keyword</li> <li>The line contains keyword</li> <li>Output:</li> <li>\$bgp_configerered by the line by t</li></ul>					~					
<pre>c Inte Betweent program ( progr</pre>										
Output: <b>\$bgp_config=</b> router bgp 65300 bgp router-id 192.168.20.254 bgp log-neighbor-changes redistribute eigrp 1 route-map E2B neighbor 100.30.0.1 remote-as 10000 ! ip forward-protocol nd Pattern: LinesByVariable[\$bgp_config]:\$bgp_as-	⊙ The line between:	\$bgp_as	✓ <sup>to</sup> End	C	~					
<pre>\$bgp_config= router bgp 65300 bgp router-id 192.168.20.254 bgp log-neighbor-changes redistribute eigrp 1 route-map E2B neighbor 100.30.0.1 remote-as 10000 ! ip forward-protocol nd Pattern: LinesByVariable[\$bgp_config]:\$bgp_as-</pre>	O The line contains keyw	ord: Enter keyword			(X)					
<pre>\$bgp_config= router bgp 65300 bgp router-id 192.168.20.254 bgp log-neighbor-changes redistribute eigrp 1 route-map E2B neighbor 100.30.0.1 remote-as 10000 ! ip forward-protocol nd Pattern: LinesByVariable[\$bgp_config]:\$bgp_as-</pre>	Output:									
router bgp 65300 bgp router-id 192.168.20.254 bgp log-neighbor-changes redistribute eigrp 1 route-map E2B neighbor 100.30.0.1 remote-as 10000 ! ip forward-protocol nd Pattern: LinesByVariable[\$bgp_config]:\$bgp_as-	\$bgp_config=									
bgp log-neighbor-changes redistribute eigrp 1 route-map E2B neighbor 100.30.0.1 remote-as 10000 ! ip forward-protocol nd Pattern: LinesByVariable[\$bgp_config]:\$bgp_as-										
<pre>redistribute eigrp 1 route-map E2B neighbor 100.30.0.1 remote-as 10000 ! ip forward-protocol nd Pattern: LinesByVariable[\$bgp_config]:\$bgp_as-</pre>	bgp router-id 192.168.2	0.254								
neighbor 100.30.0.1 remote-as 10000   !   ip forward-protocol nd   Pattern: LinesByVariable[\$bgp_config]:\$bgp_as-	bgp log-neighbor-chang	ges								
<pre>! ip forward-protocol nd Pattern: LinesByVariable[\$bgp_config]:\$bgp_as-</pre>	redistribute eigrp 1 rout	te-map E2B	Pattern appea	rs from						
<pre>ip forward-protocol nd Pattern: LinesByVariable[\$bgp_config]:\$bgp_as-</pre>	neighbor 100.30.0.1 ren	note-as 10000	step b and ste	рс						
Pattern: LinesByVariable[\$bgp_config]:\$bgp_as-	!	1								
Lifesbyvariabite[sbgp_config].sbgp_as=	ip forward-protocol nd									
Cancel Apply C	Pattern: LinesByVaria	<pre>able[\$bgp_config]</pre>	:\$bgp_as-							
Cancel Apply C				_						
				Cancel	Apply d					

e) Back in the Configuration Diagnosis window, from the menu  $\equiv$  add **Start Line** and **End Line** to limit the parsing lines to the BGP configuration data.

	<ul> <li>2. Define Diagnos</li> </ul>	is		
}			Test on	Devices: 0 😑
ariable (0)	🖾 bgp	✓ ✓ Type: Single ⑦ + New Pattern ✓		≡
{ <b>▲</b> ▼		e	Add Start L	ine
	Var Line 1	router bgp \$int:bgp_as	Add End Lir	ne
		146 router bgp 65300	Define Rep	lacement
{			Rename	
Į	Var Line 2	Line Rule might of the section of the sec	Delete	
}	Var Line 2	<pre>LinesByVariable[\$bgp_config]:\$bgp_as-</pre>	Insert Patte	ern >
		146 router bgp 65300	>	1 Line
	+ Field			
}	Output	+ Parse Lines		•
{				
}			Cancel	Apply

f) Enter the starting line and end line text into these two fields: **Start Line** and **End Line**.

Format1	1 ~	+					Test on Devi	ces: (
Double-click a v	variable to pars	e. Select multiple lines to par	rse a table.	Critical Variable (0)	🖾 bgp	✓ Z Type: Single Ø + New Pattern >	/	
	er eigrp 1	07/02/2024 12:19:03 A 8.20.0 0.0.0.3	Search	Q • •	Start Line:	router bgp	~ ©	Û
143 redi	istribute b	gp 65300 metric 100 1	L 255 255 1500 ro		End Line:	I	~ @	Û
147 148 149	bgp rou bgp log redistr	gp 65300 ter-id 192.168 -neighbor-chan ibute eigrp 1 r 100.30.0.1	nges route-map E2		Var Line 1	router bgp <pre>\$int:bgp_as</pre> - 146 router bgp 65300	> 1 Line	] ≡ ]
151 I	low-export				Var Line 2	LinesByVariable[\$bgp_config]:\$bgp_as- 146 router bgp 65300	> 1 Line	]≡
158 no ip 159 no ip	o http servo http secu oute 172.16		0.30.0.1		+ Field Output	+ Parse Lines		

- 7. Parsed variables will be listed under the **Output** section in the right pane. Validate the variables.
- 8. Click **Apply** and proceed to the **Define Diagnosis** section. We will not define any diagnosis and status code for this intent since we only export the variables to the ADT base table.

2. Define Diag	gnosis		
		Test on Dev	ices: 0
e (0) 🖾 Pattern1	✓ Z Type: Single ? + New Pattern ✓		Ξ
Start Lir	router bgp	~ (6)	ŧ
End Lir	ne: !	~ @	Ē
Var Line	235 router bgp 65001	> 1 Line	]=
Var Line	<pre>2 bgp router-id \$router_id 236 bgp router-id 10.10.10.10</pre>	> 1 Line	]= ]
Var Line	235 router bgp 65001	> 1 Line	]=
\$router_id (	+ Parse Lines = 65001 string) = 10.10.10.10 g (string) = router bgp 65001 bgp router-id 10.10.10.10 bgp log	-neighbor-chan	ges r
		Can 8	Apply

9. Exit the **Visual Parser** window.

# 9.1.3 Define the replicate settings and decode

Replicate the settings to all the BGP devices using the device group created in Section 9.1.1. And define the signature variables to add them as columns in ADT:

- 1. From the menu  $\equiv$ , select **Full Settings for Template**.
- 2. Enable the field **Serve as Template** for **Deviced-based Replication**.
- 3. In the **Intent Qualification** tab, select the device group created in earlier Section 9.1.1.

	Network Inte	nt (Edit Mode											×
	🚺 BGP Netv	vork 🔒	Diagnosis Tree					Run	with Liv	e Data	Save	😮 Help	=
	Type descriptio	on here								Intent Settings	;		
			_							Intent Variable			
		ed Logic	<u> Replication L</u>	ogic 🤣						<ul> <li>Full Settings for Lock Settings</li> </ul>	or remplate		
	a 💛 😵 + De	evice							ntent Var	Add Intent Dia	gnosis Block		-
	v 🤭 US-B	OS-R1		Type Des	cription here			+ Add Cor	nfig Diagi	Add Diagnosis		ent	
Full Settings for I	Intent Template								×	Switch Devices			
										Named Tag	ct		
2 Serve as 1	Template for: 🤇	Device-bas	ed Replication	🔘 Path-b	ased Replication	🗌 Enable N	Neighbor Pair Re	plication		Export			-11
Intent Ou	unlification		(autobla )	Critical Variabl		- Catting				Save as			
	ualification	Macro \	/ariable 0	Critical Variabi	e Advanc	e Settings			$\rightarrow$	View Original	-		
Device Qu	ualification:									View Summary Publish Intent		gnoses	-11
💿 via	Device Group	3 Dev	ice Groups/Folder	rs 🗐						Intent Replicat			- 11
										Auto Intent Wi			-10
🔾 via	Dynamic Search	n: Undefine	d										
Command	d Qualification:	0/1											
								_					4
				_			Cance	OK					-
					Add Device Group	)			×				
				ſ									
					All Dev	vice Groups Device Groups			ш.				
						3GP Devices (94)							
						Cisco Routers (4)							
					_≤ (	CPU check (6)							
						Day 1 Lab (21)							
						HSRP DEVICES (19) ared Device Group							
						icy Device Groups	-						
				l	<b>_ ~</b> #	#BGP 200 (2)							
					L 🗧 ‡	#BGP 23089 (173)							
							Cancel	OK					
							Cancel	OK					
				L									

- 4. Go to the **Advanced Settings** tab and click **+Add via Single Variables** to open the corresponding window.
- 5. Enter a name for the variable and select the variable from the dropdown menu. Add three signature variables: **BGP Configuration**, **BGP ASN**, and **Router ID**.

Full Settings for In	itent Templat	e				×				
C Serve as Te	mplate for:	Oevice-based Replication	O Path-based	Replication 🗌 Ena	ble Neighbor Pair Replication					
Intent Qua	lification	Macro Variable	Critical Variable	Advance Settings						
Match Target De	vice with See	d Devices:		а -						
O Match 1 S	Seed Device O	only								
💿 Try to Ma	tch All Seed D	Devices								
Name Rule: 🚯	Name Rule: () \$nit_name \$device_name									
Automation Tag	for Cloned In	tents: + Add 🥅								
Signature Variab	les:									
0 Items + A		Columns + Add via Sing	le Variables							
Signature	Variables Nan	ne Type	Refe	renced Parser Variables						
		Add Signature Var	iables		×					
		Name:	BGP Configuration							
			h							
		C Select Variables:	bgp_config	~						
			✓ Configuration	on.	Cancel OK					
		How it works.	□ coniigurau			Â				
			🗌 🖾 route							
			✓ 🖾 bgp_d	config						

- 6. Add all the variables and click **OK**.
- 7. Close the **Full settings for Intent Template** window.

Full Settings for Intent Template											
•	C Serve as Te	mplate for: (	Device-based Replic	ation 🔿 Path-ba	sed Replication	on 🗌 Enat	ole Neighbor	Pair Replication			
	Intent Qualification Macro Variable Critical Variable Advance Settings										
I	Match Target Device with Seed Devices:										
O Match 1 Seed Device Only											
	💿 Try to Ma	tch All Seed D	evices								
	Name Rule: )       \$nit_name \$device_name         Automation Tag for Cloned Intents: + Add =										
9	Signature Variab 3 Items + A	les: dd via Table (	Columns + Add v	ia Single Variables							
	Signature	Variables Nan	пе Туре		Re	ferenced Parser Va	riables				
	BGP ASN		Singl	e Variables	1 V	/ariable					
	Router ID		Singl	e Variables	1 V	/ariable		6			
	BGP Configuration Singl			e Variables	Variable						
	<b>`</b>										
								Са 7 ОК			

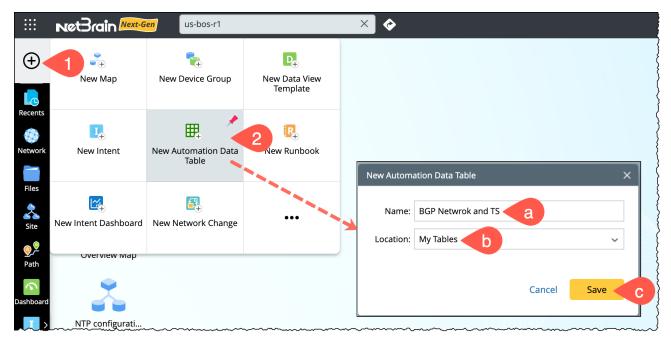
- 8. Install and decode the intent in the Intent Based Automation Center.
  - a) Open the **Intent Based Automation Center** and click **+Add an Intent Template**. Select the intent you just saved.
  - b) Click **Decode Now** to decode the intents.
  - c) Wait for the system to finish replicating and decoding intents. You can see the number of devices already decoded. Click it to view the execution log.

Inst	alled Intent Templ	ates Published	Intents Auto In	ntent	Auto Intent	t Profile	NetBrain Download				
a	+ Add Inte	ent Template Filler:	All	~	Search		Q G Refre	sh ≡	Intent Tem	olate Name: BGP Network	
4	ntent Template	Location	Intent Decoding	Decoded D	Device	Auto Intent	Cloned Intents		Basic	Cloned Intents	
<b>a</b> 0	-BGP Demo										
	Check BGP Co	All Network Inten	Last Decoded at	45		0	138		Intent	Decoding	b Decode Now
	Check BGP Sta	All Network Inten	Last Decoded at	24		0	30		⊖ Re	curring Decode	
	Check Device	All Network Inten	Last Decoded at	2		0	87		Sel	ect	~
	Troubleshoot	All Network Inten	Last Decoded at	38		0	0			Update Intent Baseline Periodically	
	Map47	Map Intent/Com	Last Decoded at	45		0	7		obr	e-Time Decode	C 53 Devices decoded
	Lab 1 Map	Map Intent/Com	Last Decoded at	127		0	0		ைற	ecoding Settings	Apply
	Map1	Map Intent/Com	Last Decoded at	69		0	0				
	draw-map	All Network Inten	Last Decoded at	0		0	0				
	ACL Config -foll	All Network Inten	Last Decoded at	123		0	12				
	Depend on sa	All Network Inten	Last Decoded at	52		_	0				

# 9.1.4 Build Base ADT

Build an ADT with the base table containing the built-in device properties and signature variables (*AS number* and *BGP configurations*) as columns:

- 1. From the desktop, click the icon 🕑 > New Automation Data Table.
- 2. In the **New Automation Data Table** popup, enter the name and select a folder to store the table.



- 3. Click **Table Builder** to open ADT Builder and define the Base Group.
- 4. Base Group tab settings: Under the **Base** tab, define the following settings:
  - a) Input **Description** for the base table to describe its use and function.
  - b) Select the method **Pre-replicated Intent Template** from the dropdown to build the base table.
  - c) Click the **Select** link to choose the intent **BGP Network** created in the previous section from **Intent Based Automation Center**.
  - d) Click **OK** to confirm the selection and close the Select Intent template window.

🖽 Automation Data Table Manager											
Search	Q 0 «	BGP Network and TS			Table Builder	3 t Updated a	t: N/A 🔍 Rebuild Table				
Shared Tables (1253)											
🔺 📄 My Tables (7)		Description: Type description	here								
Ping (1)	Automatica Data Tal			<u> </u>				Search			
BGP Network	Automation Data Tal	ble Builder									
BGP Network and TS				Selec	t Intent Template		L. L.	×			
🖽 Day 1 Lab	Column Header:							~			
HSRP Pair failover				Sele	ct Intent Template from: 🔘 A	ll Intents 🧿 In:	stalled Intents				
<ul> <li>HSRP Route Failure</li> <li>NTP</li> </ul>	Base 4		Cc	lten	ns: 1381	Filter by:	Device-based ~ Search.	Q			
	Description:				Intent Name	Location	Replication Mode				
	BGP Network <	a			Map1	Map Intent	Device-based Replication				
t i i i i i i i i i i i i i i i i i i i	Select Method to	Select Method to Build Base Table:			draw-map	All Network	Device-based Replication				
,	Pre-replicated I	I Intent Template		ACL Config -follow up	All Network	Device-based Replication					
	Intent Template:		>		Depend on sample	All Network	Device-based Replication				
	Built-in Fields:				BGP Network	All Network	Device-based Replication				
		vice 🕋 🖉		4	0-IntErrors						
	Dev				NI_Check_int_errors	All Network	Device-based Replication				
	Int	ent L									
	Intent Output:			Last	Decoded at 05:52 PM 08/06/20	024	C	Cancel OK d			
	Intent M	Message S						4			
	Intent Sta	atus Code s	~~~~	Aconor	dokan saluan kendar frasiliken	TDDAL GENE OF BRO	ilable.data field.d				

- 5. Build an ADT table with the columns **Device**, *BGP ASN*, *Router ID*, and *BGP Configuration*.
  - a) From the Built-in fields, drag and drop the **Device** field into the **Column Group (Base)** pane.
  - b) From the Device Signature Variables, drag and drop the BGP ASN, Router ID, and BGP Configuration into the Column Group (Base) pane.
  - c) Click **Save and Build** and the **Build Table** dialog appears.
  - d) Choose the settings as per your preferences and then click **Build** to save all the settings.

		Reset All
c4     Image: C4     Image: C4     Image: C4       Device     BGP ASN	C1         S         C2         S           Router ID         BGP Configuration	
Base +	Column Group (Base):	Select Column V
Description: 	Router ID BGP Configuration	c4 The second se
	BGPASN BBGPASN	
Built-in Fields:		
Intent	(Drag and drop colure (Drag and drop colure)	
Device Signature Variable: BGP ASN	Filter Row	Only show major execution process log
BGP Configuration     S	Filter devices of clone	Show all the detailed log
Intent Output:		Cancel Build

The final ADT output will appear with all the configured columns:

Search	Q 0 «	B BG	P Network and TS	Table Builder Last Upd	ated at: 08/06 🍳 Rebuild Table 🗛	dd Data Manually 🗸 📃 🛛
Shared Tables (1253)		-	iption: Type description here			
<ul> <li>My Tables (7)</li> <li>Ping (1)</li> <li>BGP Network</li> </ul>			53 Rows 4 Columns		Search Q Y Adv	vanced Filter: Undefined
BGP Network and TS		No.	管 Device	i BGP ASN	S Router ID	S BGP Configuration
🔀 Day 1 Lab		1	NIC-Lab-XR1	65501	10.8.76.200	router bgp 65501
<ul> <li>HSRP Pair failover</li> <li>HSRP Route Failure</li> </ul>		2	NIC-Lab-PE2	65501	10.8.76.203	router bgp 65501
■ NTP		3	bjta002440-SW11	65145	10.88.192.6	router bgp 65145
		4	US-SFO-R2	65001	10.8.2.250	router bgp 65001
		5	US-SFO-R1	65001	10.9.9.9	router bgp 65001
		6	US-BOS-R1	65001	10.10.10.10	router bgp 65001
		7	bur-isp-gw2	64661		router bgp 64661
		8	VRF-PE2	100		router bgp 100
		9	Internet	80001	124.1.1.1	router bgp 80001

# 9.2 Create a wrapper intent for BGP Troubleshooting

In this section, you will create an intent to check the change of the BGP configuration and a wrapper intent with follow up intent execution and then replicate to all the devices in the BGP device group created in the previous section:

- 1. Create an intent to check changes in BGP configuration
- 2. Create a Wrapper Intent

# 9.2.1 Create an intent to Check BGP Configuration

#### 9.2.1.1 Select a device and Parse variables

Create a new intent, **Check BGP Config Change**, select a BGP device, and **Add Config Diagnosis**.

Network Intent (Edit Moo	de)					×
I Check BGP Config ch	hange 2 jiagnosis Tree		Run	with Live Data	Save 2	● =
Type description here					💦 Intent Map: S	elect 🗸
I Seed Logic	🗊 Replication Logic 🤣					
😂 + Device	3	Intent	Variables: Ma	nager   Use A	ADT: 1   Tag: + A	dd 🔳 📗
v 🤭 US-BOS-R1	Туре	Description	👌 + Add Cor	nfig Diagnosis	5 Add CLI Diagnosis	≡
Configuration Di	agnosis 🗭 Type	Description here			Edit Diagnosis	; ≡
3 4 Current co 5 ! 6 ! Last con 7 ! 8 version 15 9 service til 10 service til	onfiguration nfiguration : 13636 byte figuration change at 21: .4 mestamps debug datetime ms mestamps log datetime ms ssword-encryption S-BOS-R1 -marker	48:46 EST Wed J msec	ul 31 2024	by nb		

Parse the bgp AS number and bgp configuration using the function **LinesByVariables**. Refer to Section 9.1.2 for details.

)	🖬 bgp	✓ Z Type: Single		
	- V6P			
	Start Line:	router bgp	~ 0	Û
	End Line:	I	~ @	Û
	Var Line 1	<pre>router bgp \$int:bgp_as</pre>		≡
		146 router bgp <mark>65300</mark>	> 1 Line	
	Var Line 2	<pre>LinesByVariable[\$bgp_config]:\$bgp_as-</pre>		≡
		146 router bgp 65300	> 1 Line	]
	+ Field			
	Output	+ Parse Lines		-
	\$bgp_as (int)	= 65300		
	\$bgp_config (st	ring) = router bgp 65300 bgp router-id 192.168.20.254 bgp log	-neighbor-ch	nange

#### 9.2.1.2 Add ADT Table as Intent Variable

In the diagnosis, you will compare the parsed *bgp configuration* against the configuration defined in the ADT. In order to refer to the ADT elements, you need to add the ADT table as an **Intent Variable**:

1. Go to **All Intent Variables** > **Use Automation Data Table** > **+ Automation Data Tables** to select ADT.

沓 US-BOS	-R1 Retrieve v with Live Data		
1. Define V	riable		2. Define
Fo	rmat1 🗸 +		
<li>Double-</li>	click a variable to parse. Select multiple lines to parse a table.	Critical Variable (0)	M Patter
	Intent Variables for Seed Logic		
Current			
> 235			tar
236	Intent Variable Use Automation Data Table	Task Variable	
237			
238	+ Automation Data Tables		En
239			
240			
241 242			ar
242			a
245			
245			
246			
247			
248			ar
> 249			u
250			
251			
252		Close	
253			11
	ip riow-export designation 172.10.9.0 99		+ Field
	ip flow-top-talkers		
256	ton 10		Output
257			

2. In the **Select Automation Data Table** window, select the **BGP Network** ADT and then **OK**.

Select Automation Data Tables		×
Search		Q
D Shared Tables		
🔺 😑 🖿 My Tables		
▷ <b>□</b> Ping		
☑ 冊 BGP Network		
□ 冊 BGP Network and TS		
🔲 🖽 Day 1 Lab		
—		
	Cancel	ок 🛃

#### 9.2.1.3 Define Diagnosis

In the diagnosis, compare the configurations against the configuration defined in the ADT (**BGP Network**) using two functions:

- **Match Pattern (MP)**: To compare two strings line by line and report the differences, including the changed, added and removed lines.
- **Get\_Table\_Cell:** To retrieve a cell value from an ADT table.
- 1. Define basic information such as name and anchor.
- 2. Define the **If** condition **A**:
  - a) Select Variable bgp\_config.
  - b) Select **Match Pattern** from the dropdown, validate the default settings and then click **OK**.

NOTE: Change the default rule name if you have multiple match pattern rules in one diagnosis.

🕨 2. Defin	ne Diagnosis				
📄 Add No	ote D Add Diagnosis	Can also click a var	iable on the left to add au	automation.	
Name:	Check bgp configuration chang	ge <b>1</b> Anch	or: \$bgp_as	~	
	Type description of the diagnosis	5	Match Pattern		×
🗌 Loop	Table Rows		Rule Name:	Rule1	
~ If	2		Compare Paragraph by:	Exact Match	~
A	res US-BOS-R1 Current ↓	Feuele		<ol> <li>Current string and pattern have identical matched lines</li> <li>Ignore Order of lines</li> </ol>	
в	bgp_config a ~ Select Variable ~	Equals ~ Does not equal Contains	Compare Each Line By:	Equal Contain     The line is idential to an expanded pattern line	
∼ Then		Does not contain Is empty	Learn more about Match	h Pattern rule Cancel OK	
	gnosis Message:	Is not empty Subnet contains In subnet			
<u> </u>	Set Status Code for Intent	Match pattern	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~		

c) To retrieve the configuration from the ADT table, select **Expression** > + **Get\_Table\_Cell** function.

		ne Diagnosis						
	📄 Add N	ote D Add Di	agnosis	Can a	lso click a	variable	on the left to add a	utomation.
	Name:	Check bgp configu	ration change	2	А	nchor:	\$bgp_as	~
		Type description of	the diagnosis					
	□ Loop ∨ If	o Table Rows						
	A	😁 US-BOS-R1	Current 🗸					
		bgp_config	~	MP(Rule1)	~		````	/ 🟛
	В	Select Variable	~			US 🍋 🖌 5 🖌	-BOS-R1 Configuration	
	∽ Then						🙁 bgp_as	
	📄 Dia	gnosis Message:	n				i bgp_config ▲ 😫 Rule1 The Result	t =
		Set Status Code for I Set Status Code for I					Matched_li	
						Expres		
	Add Log	ic ~	~~~~~~	·····		Define	Variable	
pression					×			
xpression:								
	+ 💽 Variabl	e + 🔃 Function + 🎟 Ge	t_Table_Cell	C				
Help				Cancel	ОК			

Expre

- d) In the **Get Table Cell** window, select the ADT table, column, and condition to retrieve a table cell:
  - i. Select the table **BGP\_Network**.
  - ii. Select the Column **BGP\_Configuration**.
  - iii. Define Row Matching Condition A: Device | Equals | this\_device.

The Expression will be: Get\_Table\_Cell (BGP\_Network, BGP\_Configuration, <Condition: A>)

Get Ta	able Cell			×
Retri	ieve Cell Value from:			
Sele	ct Table: BGP_Network ┥	1		~
Fro	om Column: BGP_Configu	iration 🥑		~
Fro	om the Row Matching Cond	ition:		
A	Device	✓ Equals	↓ this_devi	ce 🗸 💼
в	Select Variable	~		
Во	olean Expression: A			
Expr	ression: Get_Table_Cell (BGF	_Network, BGP_Co	nfiguration, <condition< td=""><td>on: A&gt;)</td></condition<>	on: A>)
			~	аг і ок
			Ca	UK UK

3. Then: In case If logic is true, define the color (green), status (Success), and:

**Message**: *BGP* configuration did not change.

4. Check the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.

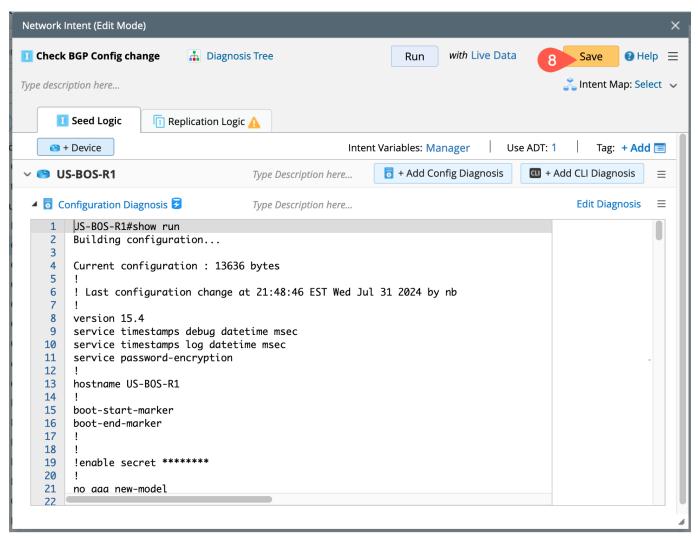
5. **Else**: In case **If** logic is not true, define the color (**red**), status (**Error**) and:

**Message**: *BGP* configuration changed. *Missing lines*: *\$Rule1.Unmatched\_lines*. *Extra lines*: *\$Rule1.Unused\_pattern\_lines*.

- 6. Check the selection boxes of the **Set Status Code for Device** and **Set Status Code for Intent** to duplicate the message to the Device Status Code.
- 7. Click **Apply** to save the settings and then close the window.

2. Define Diagnosis	Add Diagnosis	Can also click a variable on the left to add automatio
A _ JP-TYO-CF		
bgp_config	~	MP(Rule1) ~ Get_Table_Cell(BG ~ 📾
B Select Varia	ble 🗸	
<ul> <li>✓ Then</li> <li>☐ Diagnosis Message</li> </ul>	ge:	Save to Incident 😑
	configuration did	d not change
Image: Set Status Control       Image: Set Set Status Control       Image: Set Set Set Set Status Control       Image: Set		tion did not change
Set Status Co		tion did not change
Else		🛅 Delete
5 Diagnosis Messag		□ Save to Incident
Set Status Co		
<ul> <li>● Error </li> <li>● Error </li> <li>● S Set Status Co</li> </ul>		tion changes. Missing lines: \$Rule1.Unmatched_lines. Extra lin
Error	BGP configurat	tion changes. Missing lines: \$Rule1.Unmatched_lines. Extra lin
Add Logic ~		/
+ Add Elself		
{		Can 7 Apply

8. In the Network Intent (Edit Mode) dialog, click Save.



- 9. Without closing the window, replicate the intent to the ADT as detailed in the next section.
- 10. Run the intent to test its correctness. Often, your BGP configuration did not change. You can modify the ADT table to emulate a changed BGP to test your intent. The following is an example result.

### 9.2.1.4 Replicate the Intent to the existing ADT

Use the **Intent Replication Wizard** to replicate the **Check BGP config change** intent to the ADT table BGP\_Network created in Section 9.1.1.

- 1. Launch the **Intent Replication Wizard** from the  $\blacksquare$  menu.
- 2. The seed intent will appear by default as you have launched it from the **Network Intent** window. Click **Next.**

Intent Settings   Intent Settings   Intent Variables:   Intent Variables: </th <th>Network Intent (Edit Mode)</th> <th>×</th>	Network Intent (Edit Mode)	×
10 service timestamps log datetime msec 11 service password-encryption 12 !	<pre>ype description here  Seed Logic  Period  Period Period  Period  Period  Period  Period  Period  Period  Period Perio</pre>	Intent Variables: Manager Intent Variables: Manager The total of the total of total of the total of to
		<b>2</b> Ne

3. In the **Define ADT** ribbon, go to **Use an Existing ADT** > **BGP Network** ADT >**OK**.

Intent Replication Wizard	- Check BGP Config change		×
Seed Intent	Define ADT	Replication Settings	Replicate Intent
	Create a New ADT	Use a	n Existing ADT
Select Automation Data	Table	, second second	×
Search		Q	
<ul> <li>Shared Tables</li> <li>My Tables</li> </ul>			~
▷ <b>■</b> Ping <b>■</b> BGP Network			
Bay 1 Lab			
HSRP Route F	allure		Previous Next
		Cancel OK	

4. Replicate Intent to: select **New Column Group** and enter the name **Check BGP Config Change** and click **Next**.

Intent Replication Wizard - Check BGP Config cha	nge X
Define ADT	Replication Settings Replicate Intent
Create a New AD	Use an Existing ADT
Automation Data Table:	BGP Network
Replicate Intent to:	New Column Group 🗸 Check BGP Config Change
Replicate on Device Column:	Device ~
<ol> <li>Selection Mode: Device-based Replication.</li> </ol>	Previous 4 Next

5. Add the BGP device group created in Section 9.1.4 and click **Next.** 

ent Replication Wizard - HSR	P Pair failover check_pa	rent				
Seed Intent	Define ADT	Replication Set	tings	Replicate	e Intent	
				🛞 Full :	Settings for Template	
Intent Qualification:	via Device Groups/Si	ites: 1 Device Groups/	Folders 💼	🔘 via Dynan	nic Search: Undefined	
Define Macro Variable ltem: 1	s and Rules for Their S	ubstitution:	Add the B device gro			
Seed Device	See	ed Command			Macro Variables	
	Please click	to select an entry	from the abo	ove table.		
More Replication Settin	igs 🗸					
Selection Mode: Device-bas	sed Replication, ADT: BC	SP Network.			Previous 5	Next

6. In the **Replication Settings**, modify the replicated intent **Column Name** and add additional columns such as **Intent Status Code** and **Device Status Code**.

Seed Intent	Define ADT	Replication Settin	gs	Replicate Intent
ADT Columns:				Additional Columns ~
Column Data	Column Name		Тад	Replicated Intent
Replicated Intent	Check BGP config change		0 tags	Intent Message
	·			Intent Status Code
				🗌 Device Status Code
				Intent Devices
		Add columns	s to	🔲 Intent Map
		the table as		🔲 Intent CLI Comma.
		needed		Last Execution Tim

7. After selecting **Save and Replicate** to save all the settings and create ADT.

Seed Intent	Define ADT Replication Settings	Replicate Intent
ADT Columns:		Additional Colum
Column Data	Column Name	Тад
I Replicated Intent	Check BGP config change	0 tags
s Intent Status Code	Intent Status Code	
s Device Status Code	Device Status Code	
		7 Save and Replicate

The table will now be populated with devices and the replicated Intents (Check BGP config Change).

👪 BG	iP Network	Table	Builder Last Updated at	: 08/08/2024 01:39 PM 🛛 🔍 Rebuil	ld Table Add Data Manu
Descr	iption: Type description here		Rep	olicated	
ltems:	53 Rows 7 Columns		Inte	ent Sear	rch Q 🍸 Advanced Filter: U
No.	Device	BGP ASN	S Router ID	<b>S</b> BGP Configuration	Check BGP config change
1	US-SFO-R1	65001	10.9.9.9	router bgp 65001	Check BGP Config change US-SF 💿
2	US-BOS-R1	65001	10.10.10.10	router bgp 65001	Check BGP Config change US-BO •
3	US-SFO-R2	65001	10.8.2.250	router bgp 65001	Check BGP Config change US-SF 💿
4	NIC-Lab-PE2	65501	10.8.76.203	router bgp 65501	Check BGP Config change NIC-La 💿
5	ip-172-26-0-114	64513		router bgp 64513	Check BGP Config change ip-172 •
6	PE-3600X-01	64550	10.88.255.1	router bgp 64550	Check BGP Config change PE-36 ④
7	VRF-PE2	100		router bgp 100	Check BGP Config change VRF-PE2 ④
8	bjta002440-SW11	65145	10.88.192.6	router bgp 65145	Check BGP Config change bjta00 ④

### 9.2.1.5 Execute and view Results

Run the replicated intent in the ADT once and rebuild the table. Review the intent result in the column **Intent Status Code**.

₽	🖥 Automation Data Table Manager									
»	🖥 BGI	P Network	Table Builder	Last Updated at: 08/08/2	024 01:39 PM 🛛 🌯 Rebuild Tabl	e				
	Descri	ption: Type description here								
	ltems:	53 Rows 7 Columns				[	Search Q			
	No.	Device	I BGP ASN	S Router ID	S BGP Configuration	Check B Run Details	S Intent Status Code			
1		US-SFO-R1	65001	10.9.9.9	router bgp 65001	Check BGP Config change US-SF •				
2		US-BOS-R1	65001	10.10.10.10	router bgp 65001	Check BGP Config change US-BO 👁				
3		US-SFO-R2	65001	10.8.2.250	router bgp 65001	Check BGP Config change US-SF •				
4		NIC-Lab-PE2	65501	10.8.76.203	router bgp 65501	Check BGP Config change NIC-La 👁				
5		ip-172-26-0-114	64513		router bgp 64513	Check BGP Config change ip-172 •				
6		PE-3600X-01	64550	10.88.255.1	router bgp 64550	Check BGP Config change PE-36 •				
7		VRF-PE2	100		router bgp 100	Check BGP Config change VRF-PE2				

No.DeviceIBGP ASNRouter IDIBGP ConfigurationI Check BGP Config changeIntent Status CodeI Device Status1US-SPO-R16500110.9.9.9router bgp 65001Check BGP Config change US-SPO	-	P Network ption: Type description here	Table Bu	ilder Last Updated at:	08/08/2024 02:12 PM 🛛 🔍 Rebui	Result as defined in the intent diagnosis			Add Data Manually $\sim$
Image: Construct of the second seco	Items:	53 Rows 7 Columns					Se	arch Q	T Advanced Filter: Undefine
1         1	No.	Cevice	BGP ASN	S Router ID	S BGP Configuration	Check BGP config change		S Intent Status Code	S Device Status Code
Indext and the sector of th	1	US-SFO-R1	65001	10.9.9.9	router bgp 65001	Check BGP Config change US-SFO	0	BGP did not change	US-SFO-R1
3       US-SFO-R2       65001       10.8.2.250       router bgp 65001       Check BGP Coning change US-SFO       BGP did not change       BGP did not change         44       NIC-Lab-PE2       65501       10.8.76.203       router bgp 64513       Check BGP Coning change NIC-Lab       BGP did not change       PE-3600X-01         5       Ip-172-266-0114       64550       10.88.255.1       router bgp 64550       Check BGP Config change VRF-PE2       BGP did not change       VRF-PE2       BGP did not change       VRF-PE2       BGP did not change       VRF-PE2       BGP did not change       NIC-Lab-XR1       BGP did not change       Sic Core-3560X       BGP di	2	US-BOS-R1	65001	10.10.10.10	router bgp 65001	Check BGP Config change US-BOS	0	BGP did not change	
44       NIC-Lab-PE2       65501       10.8.76.203       router bgp 65501       Check BGP Config change NIC-Lab       BGP did not change       ip:722-26-0114       64513       Feadows	3	US-SFO-R2	65001	10.8.2.250	router bgp 65001	Check BGP Config change US-SFO	0	BGP did not change	ip-172-26-0-114
And the second secon	4	NIC-Lab-PE2	65501	10.8.76.203	router bgp 65501	Check BGP Config change NIC-Lab	0	BGP did not change	BGP did not change
All of the the term of the term of term	5	ip-172-26-0-114	64513		router bgp 64513	Check BGP Config change ip-172-2	•	BGP did not change	ip-172-26-0-114
All       bita00240-SW11       65145       10.88.192.6       router bgp 65145       Check BGP Config change bjta002       BGP did not change       bjta00240-SW         99       bur-isp-gw2       64661       router bgp 656145       Check BGP Config change bur-isp       BGP did not change       bjta00240-SW         100       NIC-Lab-XR1       65501       10.8.76.200       router bgp 65601       Check BGP Config change bita002       BGP did not change       bjta00240-SW         111       bjta00230-SW9       65000       10.61.4.25       router bgp 65600       Check BGP Config change bita002       BGP did not change       bjta00230-SW         122       Sjc-Core-3560x-01       64552       10.88.255.81       router bgp 4552       Check BGP Config change IPv6Lab       BGP did not change       bjta00230-SW         133       IPv6Lab-R4       4       4.4.4.4       router bgp 102       Check BGP Config change IPv6Lab       BGP did not change       IPv6Lab-R4         144       VRF-MCE1       102       router bgp 80001       Check BGP Config change Internet       BGP did not change       NF-MCE1         155       Internet       80001       124.1.1.1       router bgp 80001       Check BGP Config change Internet       BGP did not change       NF-MCE1         161       VRF-PE1       <	6	PE-3600X-01	64550	10.88.255.1	router bgp 64550	Check BGP Config change PE-3600	•	BGP did not change	PE-3600X-01
9       burisp-gw2       64661       router bgp 64661       Check BGP Config change burisp       BGP did not change       burisp-gw2         10       NIC-Lab-XR1       65501       10.8.7.6.200       router bgp 65601       Check BGP Config change NIC-Lab       BGP did not change       NIC-Lab-XR1         11       bja002303-SW9       65000       10.61.4.25       router bgp 65501       Check BGP Config change NIC-Lab       BGP did not change       bja0023.03-SW         12       Sjc-Core-3560x-01       64552       10.88.255.81       router bgp 64552       Check BGP Config change IPv6Lab       BGP did not change       bja002303-SW         13       IPv6Lab-R4       4       4.4.4.4       router bgp 102       Check BGP Config change IPv6Lab       BGP did not change       IPv6Lab-R4         14       VRF-MCE1       102       router bgp 80001       Check BGP Config change INF-MCE1       BGP did not change       NF-MCE1         15       Internet       80001       124.1.1.1       router bgp 80001       Check BGP Config change INF-MCE1       BGP did not change       NF-MCE1         16       VRF-PE1       100       router bgp 80001       Check BGP Config change INF-MEE1       BGP did not change       NF-MCE1         17       Iv6Lab-MLS       5.5.5.5       router bgp 5 <t< td=""><td>7</td><td>VRF-PE2</td><td>100</td><td></td><td>router bgp 100</td><td>Check BGP Config change VRF-PE2</td><td>0</td><td>BGP did not change</td><td>VRF-PE2</td></t<>	7	VRF-PE2	100		router bgp 100	Check BGP Config change VRF-PE2	0	BGP did not change	VRF-PE2
Interpret	8	bjta002440-SW11	65145	10.88.192.6	router bgp 65145	Check BGP Config change bjta002	0	BGP did not change	bjta002440-SW11
Marked	9	bur-isp-gw2	64661		router bgp 64661	Check BGP Config change bur-isp	0	BGP did not change	bur-isp-gw2
2       Sic-Core-3560x-01       64552       10.88.255.81       router bgp 64552       Check BGP Config change Sic-Core       BGP did not change       Sic-Core-3560x         3       IPv6Lab-R4       4       4.44.4       router bgp 4       Check BGP Config change Sic-Core       BGP did not change       IPv6Lab-R4         4       VRF-MCE1       102       router bgp 102       Check BGP Config change VRF-MCE1       BGP did not change       VRF-MCE1         5       Internet       80001       124.1.1.1       router bgp 100       Check BGP Config change VRF-MCE1       BGP did not change       Internet         6       VRF-PE1       100       router bgp 100       Check BGP Config change VRF-PE1       BGP did not change       VRF-PE1         6       VRF-PE1       100       router bgp 100       Check BGP Config change VRF-PE1       BGP did not change       VRF-PE1         7       IPv6Lab-MPLS       5       5.5.5.5       router bgp 5       Check BGP Config change IPv6Lab       BGP did not change       IPv6Lab-MPLS	0	NIC-Lab-XR1	65501	10.8.76.200	router bgp 65501	Check BGP Config change NIC-Lab	0	BGP did not change	NIC-Lab-XR1
13       IP6Lab-R4       4       4.4.4.4       router bgp 4       Check BGP Config change IPv6Lab ●       BGP did not change       IPv6Lab-R4         14       VRF-MCE1       102       router bgp 102       Check BGP Config change VRF-MCE1 ●       BGP did not change       VRF-MCE1         15       Internet       8001       124.1.1.1       router bgp 100       Check BGP Config change VRF-MCE1 ●       BGP did not change       VRF-MCE1         16       VRF-PE1       100       router bgp 100       Check BGP Config change VRF-PE1 ●       BGP did not change       VRF-PE1         17       IP6Lab-MPLS       5       5.5.5       router bgp 5       Check BGP Config change IPv6Lab ●       BGP did not change       IPv6Lab-MPLS	1	bjta002303-SW9	65000	10.61.4.25	router bgp 65000	Check BGP Config change bjta002	0	BGP did not change	bjta002303-SW9
Internet       NRF-MCE1       Internet       Internet <th< td=""><td>12</td><td>Sjc-Core-3560x-01</td><td>64552</td><td>10.88.255.81</td><td>router bgp 64552</td><td>Check BGP Config change Sjc-Core</td><td>0</td><td>BGP did not change</td><td>Sjc-Core-3560x-01</td></th<>	12	Sjc-Core-3560x-01	64552	10.88.255.81	router bgp 64552	Check BGP Config change Sjc-Core	0	BGP did not change	Sjc-Core-3560x-01
Internet       80001       124.1.1.1       router bgp 80001       Check BGP Config change Internet ●       BGP did not change       Internet         16       VRF-PE1       100       router bgp 100       Check BGP Config change VRF-PE1 ●       BGP did not change       VRF-PE1         17       IPv6Lab-MPLS       5.5.5.5       router bgp 5       Check BGP Config change IPv6Lab●       BGP did not change       IPv6Lab-MPLS	3	IPv6Lab-R4	4	4.4.4.4	router bgp 4	Check BGP Config change IPv6Lab	0	BGP did not change	IPv6Lab-R4
Market     Note of the second se	4	VRF-MCE1	102		router bgp 102	Check BGP Config change VRF-MCE1	0	BGP did not change	VRF-MCE1
7     IPv6Lab-MPLS     5     5.5.5.5     router bgp 5     Check BGP Config change IPv6Lab Implementation of the second	5	Internet	80001	124.1.1.1	router bgp 80001	Check BGP Config change Internet	0	BGP did not change	Internet
	6	VRF-PE1	100		router bgp 100	Check BGP Config change VRF-PE1	•	BGP did not change	VRF-PE1
8 NBLAB-XR-PE1 65100 10.8.19.1 router bgo 65100 Check BGP Config change NBLAB-XR-PE1	7	IPv6Lab-MPLS	5	5.5.5.5	router bgp 5	Check BGP Config change IPv6Lab	0	BGP did not change	IPv6Lab-MPLS
	8	NBLAB-XR-PE1	65100	10.8.19.1	router bgp 65100	Check BGP Config change NBLAB	0	BGP did not change	NBLAB-XR-PE1

#### 9.2.1.6 Set the intent tag as BGP.

Add a tag **BGP** to the intent column:

- Help Add Data Manually  $\lor \equiv \mathbf{G}$ 🔍 Rebuild Table Search... C Check BGP config change ►≡ S Intent Status Code S Device Status Code Ξ Tag Current Column PA Once BGP did not change US-SFO-R1 PA via Timer BGP did not change US-BOS-R1 + Add 📰 Intent BGP did not change US-SFO-R2 tent-related Column Group BGP did not change NIC-Lab-PE2 npty Wrapper Intent Cancel OK BGP did not change ip-172-26-0-114 o Intent BGP did not change PE-3600X-01 Export Diagnosis Result to CSV p 100 BGP did not change VRF-PE2 View Summary Report p 65 45 BGP did not change bjta002440-SW11 Export Intent Output Map p 64661 BGP did not change bur-isp-gw2 Debug Empty Cells p 65501 BGP did not change NIC-Lab-XR1 Tag Current Column (p 65000 BGP did not change bjta002303-SW9 Edit p 64552 BGP did not change Sjc-Core-3560x-01 Delete p 4 BGP did not change IPv6Lab-R4 Set as Table Key p 102 VRF-MCE1 BGP did not change Submit Related Commands to Benchmark 80001 p BGP did not change Internet New Intent Dashboard p 100 BGP did not change VRF-PE1 Check BGP Config change IPv6Lab... • р5 BGP did not change IPv6Lab-MPLS Check BGP Config change NBLAB-... • p 65100 BGP did not change NBLAB-XR-PE1
- 1. Go to the intent column 🗏 menu > Tag Current column:

2. The added tag can be seen in the intent column header.

🐻 Au	tomation Data Ta	ble Mana	ger				
🖽 BG	iP Network		Table Builder         Last Update	d at: 08/08/2024 02:12 PM 🛛 🔍 Rebuild T	īable		
Descr	iption: Type description her	re Та	ag Current Column	×			
ltems	: 53 Rows 7 Columns					Search	Q
No.	Device	īΒ	1 Tag: BGP +		Check BGP config change	🖹 Detail	■ SInte
1	US-SFO-R1	650			Check BGP Config change US-SFO-R1	1 Tag:	li
2	US-BOS-R1	650		Cancel OK	Check BGP Config change US-BOS-R1	BGP	li
3	US-SFO-R2	650			Check BGP Config change US-SFO-R2		
4	NIC-Lab-PE2	65501	10.8.76.203	router bgp 65501	Check BGP Config change NIC-Lab-PE2	۲	BGP di
5	ip-172-26-0-114	64513		router bgp 64513	Check BGP Config change ip-172-26-0-114	4 <b>O</b>	BGP di
an	- AF-26AQXAA		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~			~~~~	RCD

### 9.2.2 Create a Wrapper Intent

Create a wrapper intent as the public interface of intents to troubleshoot BGP-related problems. You will define the follow up intents logic as all the intent columns from the ADT with the tag BGP:

- 1. <u>Create a Wrapper Intent</u>
- 2. <u>Replicate and Execute the Wrapper Intent</u>

#### 9.2.2.1 Create a Wrapper Intent

From the **Intent Manager**, create a new intent and name it as **Wrapper intent for BGP Troubleshooting.** 

- 1. Click +Device and add a device from the BGP device group as the seed Intent, e.g., US-BOS-R1.
- 2. Click 🔳 located beside the add diagnosis buttons and then **Add Device Diagnosis Block**.

Network Intent (Edit Mode)	i de la companya de l				×
Wrapper intent for BGI	P Troubleshooting	🔒 Diagnosis Tr	ree Run	with Live Data	Save 🛛 🕄 Help 🚍
Type description here					💦 Intent Map: Select 🗸 🗸
Seed Logic	TReplication Logic	<b>A</b>			
😤 + Device 🚽 1				Intent Variables: N	Nanager 🕴 Tag: + Add 🧮
v 🤭 US-BOS-R1		Type Description	here 📴 + Ad	d Config Diagnosis	the second
No content has been added	ι.			Add AP Add Pin Add Tra Add De Add Ac	IMP Diagnosis PI Diagnosis age Diagnosis evice Diagnosis Block 2 tion > ate Section

- 3. Add a description to the diagnosis about the follow up action.
- 4. In the **If** condition, select **True** from the Select Variable dropdown.
- 5. Remove the **Diagnosis Message** and add **Follow-up Intent** from the **Add logic** dropdown.

6. Click the **Network Intent** link to add the follow up intent from the ADT.

×
=
ОК

#### 7. Click Select Intents from ADT.

ow-up Intents										_
Follow-up Intents:	+ Follow-up	• 0								
	Select Inter	nt (Standalone)	-							
		nt Template								
		nts from ADT nt Templates via A	ADT							
	elect Inter									
		Select Automatio	on Data Table					×		
	1									
		Search					C	2		
		Share								
		My Ta	ables							
			SP Network							
			GP Network and T	S						
			ay 1 Lab							
						Cancel	ОК	a		
		2								
Prune other follow-	in intents						-	ancol	5-11-0	
Prune other follow-u	up intents						C	Cancel	Save	
Prune other follow-u	up intents						C	Cancel	Save	
	up intents + Follow-up ~	• •	_		-		(	Cancel	Save	
ow-up Intents	+ Follow-up ~			-	-			Cancel	Save	Ē
ow-up Intents Follow-up Intent:	+ Follow-up ~ (Automation Dat							Cancel	Save	
ow-up Intents Follow-up Intent:	+ Follow-up ~ (Automation Dat							Cancel	Save	
ow-up Intents Follow-up Intent: ~ 📆 BGP Network ( Description: Whe	+ Follow-up ~ (Automation Dat	ta Table)	ntent Variables (	Device or Device V	'ariables):		(	Cancel	Save	
ow-up Intents Follow-up Intent: -  -  -  -  -  -  -  -  -  -  -  -  -	+ Follow-up ~ (Automation Daten mation Assets (	ta Table)		Device or Device V	'ariables):			Cancel	Save	
ow-up Intents Follow-up Intent:	+ Follow-up ~ (Automation Daten mation Assets (	ta Table) (ADT Rows) by In	tical asset.	Device or Device V				Cancel	Save	
ow-up Intents Follow-up Intent: -  -  -  -  -  -  -  -  -  -  -  -  -	+ Follow-up ~ (Automation Daten mation Assets (	ta Table) (ADT Rows) by In	tical asset.		'ariables):			Cancel	Save	
ow-up Intents Follow-up Intent:	+ Follow-up ~ (Automation Date en mation Assets (	ta Table) (ADT Rows) by In properties of crit	tical asset.					Cancel		
ow-up Intents Follow-up Intent: Follow-up Intent: BGP Network ( Description: Whe Find Critical Auto Find critical ass A Device B Select	+ Follow-up ~ (Automation Date en mation Assets (	ta Table) ( <b>ADT Rows) by In</b> properties of crit	tical asset.					Cancel		
ow-up Intents Follow-up Intent: Follow-up Intent: BGP Network ( Description: Whe Find Critical Auto Find critical ass A Device B Select	+ Follow-up ~ (Automation Dat en mation Assets ( sets by device, or	ta Table) ( <b>ADT Rows) by In</b> properties of crit	tical asset.					Cancel		
ow-up Intents Follow-up Intent:	+ Follow-up ~ (Automation Date en mation Assets ( sets by device, or pression: A	ta Table) (ADT Rows) by In	tical asset.					Cancel		
ow-up Intents Follow-up Intent: Follow-up Intent: BGP Network ( Description: Whe Find Critical Auto Find critical ass A Device B Select	+ Follow-up ~ (Automation Date en mation Assets ( sets by device, or pression: A	ta Table) (ADT Rows) by In	tical asset.					Cancel		
ow-up Intents Follow-up Intent:	+ Follow-up ~ (Automation Date) en mation Assets ( ets by device, or ets by device, or pression: A of Found Critic	ta Table) (ADT Rows) by In	tical asset.					Cancel		
ow-up Intents Follow-up Intent:  Follow-up Intent:  George BGP Network ( Description: Whe Find Critical Autor Find Critical Autor Find critical ass A Device B Select Boolean Exp Select Intents A Intents	+ Follow-up ~ (Automation Date) en mation Assets ( ets by device, or ets by device, or pression: A of Found Critic s	ta Table) (ADT Rows) by In properties of critical al Automation A	tical asset.					Cancel		
ow-up Intents Follow-up Intent: Comparison BGP Network ( Description: Whe Find Critical Autor Find critical ass A Device B Select Boolean Exp Select Intents	+ Follow-up ~ (Automation Date) en mation Assets ( ets by device, or pression: A of Found Critic s	ta Table) (ADT Rows) by In properties of critical al Automation A	tical asset.					Cancel		
ow-up Intents Follow-up Intent:  Follow-up Intent:  BGP Network  Find Critical Auto Find Critical Auto Find critical ass A Device B Select Boolean Exp Select Intents All Intents Selected I	+ Follow-up ~ (Automation Date) en mation Assets ( ets by device, or ets by device, or pression: A of Found Critic s	ta Table) (ADT Rows) by In properties of critical al Automation A t	tical asset.					Cancel		
ow-up Intents Follow-up Intent:  Follow-up Intent:  BGP Network  Find Critical Auto Find Critical Auto Find critical ass A Device B Select Boolean Exp Select Intents All Intents Selected I	+ Follow-up (Automation Dat en mation Assets ( sets by device, or pression: A of Found Critic s Intents: Selec	ta Table) (ADT Rows) by In properties of critical al Automation A t	Assets to Execut		this_device			Cancel		

- 8. Back in the Network Intent (Edit Mode) dialog, click **Save**.
- 9. Click **Run** to execute the intent.

Network Intent (Edit Mode)		×
I Wrapper intent for BGP Troubl	Lagnosis Tree	9 Run with Live Data 8 Save 8 Help =
Type description here		🏅 Intent Map: Select 🗸
Seed Logic     Replicatio	n Logic 🛕	
🚳 + Device		Intent Variables: Manager   Tag: + Add 🗐
v 🤭 US-BOS-R1	Type Description here	o+ Add Config Diagnosis■■+ Add CLI Diagnosis■
Cevice Diagnosis Block	Type Description here	Edit Diagnosis 🛛 🚍
t i		
1		
		4

10. Open the diagnosis tree to see the results.

Network Intent (View Mode) - All Network Intents/Arun Belide/Wrapper i	ntent for BGP Troubleshooting		×
I Wrapper intent for BGP Trouble		🚨 Open 📄 0 🔥 0 🗶 Ed	dit =
Result: 08/08/2024 04:02 Pt 10 a @ Market State	You can View Execution Log	Run vith Live	Data
S BGP did not change 2	C C	Ň	View ~
V C US-BOS-R1 1 Diagnosis			
Device Diagnosis Block Diagnosis Tree of Wrapper intent for BGP Troubleshooting	1 Diagnosis 1 Follow-up NI		×
Source: 💄 ArunKumar.Belide@netbraintech.com Current NI	: 💶 Wrapper intent for BGP Troubleshooting	Execution Time: 08/08/2024 04:02:58 PM 🗸	🚳 View Settings   抗 Auto Layout   😋 🔳
	→ DBecause	→閱Ⅰ	
ArunKumar.Belide Wrapper intent for BG US-BOS-R1	BGP Config check Device Diagno Follow up	BGP Network Check BGP Config cl	h US-BOS-R1 Check bgp config
	intent from ADT with BGP tag is executed		
Network Intent Details - Wrapper intent for BGP Troubleshooti	ng		e7 /

#### 9.2.2.2 Replicate and Execute the Wrapper Intent

Now, you can use the **Intent Replication Wizard** to replicate the **wrapper** intent to all the devices in the ADT table **BGP Network**. Refer to Section 9.2.1.4 for details. In the second step (**Define ADT**), select the existing ADT, **BGP Network**. In the third step, define the **intent qualification** with device group **BGP Devices** to include devices for the wrapper intent to replicate on.

rch Q O	K BGP Network	Table Builder	lasti	Ipdated at: 08/08/2024 06:55 PM	Pobuild Table	Add Data Manually 🗸 🗉	_
Syed_interface_CRC_check_l1_intent_DE		Table Builder	Last L	paalea al: 08/08/2024 08:55 PM	Rebuild Table	Add Data Manually V	- 1
🖶 test	Description. Type description here			<b>Deplicated</b> wronner			
🖶 test(1)	Items: 53 Rows 8 Columns			Replicated wrapper intent	Search	🔍 🛛 🗑 Advanced Filter: Undefined	d
Test_Joey	S BGP Configuration	Check BGP config change	0	Intent	e Status Code	TS BGP (wrapper)	
Test1	router bgp 65001	Check BGP Config change US-SF	0	BGP uiu not change US		Wrapper intent for BGP Tro •	
4 teST32323	router bgp 65001	Check BGP Config change US-BO			OS-R1	Wrapper intent for BGP Tro •	
🗄 testEW	router bgp 65001	Check BGP Config change US-SF			FO-R2	Wrapper intent for BGP Tro	
👪 testka		Check BGP Config change NIC-La	-			Wrapper intent for BGP Tro	
Teststetre	router bgp 65501				ab-PE2	1	
🖏 testzile 🖏 uptime_dashboard	router bgp 64513	Check BGP Config change ip-172	-	<b>·</b>	2-26-0-114	Wrapper intent for BGP Tro •	
uptime_days	router bgp 64550	Check BGP Config change PE-36		BGP did not change PE-36	500X-01	Wrapper intent for BGP Tro •	
Verify Interface State	router bgp 100	Check BGP Config change VRF-PE2	0	BGP did not change VRF-F	PE2	Wrapper intent for BGP Tro 💿	
version checks_test01	router bgp 65145	Check BGP Config change bjta00	0	BGP did not change bjta0	02440-SW11	Wrapper intent for BGP Tro ④	
Viptela Assessment Library	router bgp 64661	Check BGP Config change bur-is	0	BGP did not change bur-is	sp-gw2	Wrapper intent for BGP Tro 💿	
B VPN Device	router bgp 65501	Check BGP Config change NIC-La	0	BGP did not change NIC-L	.ab-XR1	Wrapper intent for BGP Tro •	
B Wireless Controller	router bgp 65000	Check BGP Config change bjta00	0	BGP did not change bita0	02303-SW9	Wrapper intent for BGP Tro •	
WLC_AP_Count Jyyyzzz	router bgp 64552	Check BGP Config change Sjc-Co	0	BGP did not change Sic-Co	ore-3560x-01	Wrapper intent for BGP Tro •	
The second secon	router bgp 4	Check BGP Config change IPv6La			ab-R4	Wrapper intent for BGP Tro •	
My Tables (7)	router bgp 102	Check BGP Config change VRF-M		BGP did not change VRF-N		Wrapper intent for BGP Tro •	
Ping (1)		Check BGP Config change Internet				Wrapper intent for BGP Tro	
BGP Network	router bgp 80001			BGP did not change Inter			
BGP Network and TS	router bgp 100	Check BGP Config change VRF-PE1	o	BGP did not change VRF-F	PE1	Wrapper intent for BGP Tro 💿	
🖏 Day 1 Lab	router bgp 5	Check BGP Config change IPv6La	0	BGP did not change IPv6L	ab-MPLS	Wrapper intent for BGP Tro 💿	
HSRP Pair failover	router bgp 65100	Check BGP Config change NBLA	0	BGP did not change NBLA	B-XR-PE1	Wrapper intent for BGP Tro 💿	
HSRP Route Failure     NTP	router bgp 65523	Check BGP Config change West	0	BGP did not change West	-CSR1000v	Wrapper intent for BGP Tro 💿	

# 9.2.3 Execute the Auto Intent BGP Wrapper

Let us enable the auto intent feature for wrapper intent on a map and execute it on the devices in the map:

- 1. Click the 🔳 icon in the intent column **TS BGP (wrapper)**, then click **Enable Auto Intent**.
- 2. In the Enable Auto Intent dialog, validate the Auto Intent default name **BGP Network TS BGP** (wrapper).
- 3. Click Select > Select Profile > Shared Profiles > [<Your profile>] > click OK.
- 4. Click **OK** to complete the creation of the Auto Intent for the replicated **TS BGP (wrapper).**

Ta	able Builder	Last Updated at: 08/08	/2024 11:34 PM 🛛 🔍 Rebuild Tab	ble Add Data Manually >	∕≡ 🖬
ription here					
umns			Search	Q T Advanced Filter: Undef	ined C
Check BGP Config Char	ige 🔊	S Intent Status Code	S Device Status Code	TS BGP (Wrapper)	=
Check BGP Config change	US-SF 💿	BGP did not change	US-SFO-R1	Run Intents Once	<b>(A)</b>
Check BGP Config change	US-BO 💿	BGP did not change	US-BOS-R1	Run Intents via Timer	<b>PA</b>
Enable Auto Intent - TS BGP (V	Vrapper)		-R2 -PE2	Open Seed Intent Rebuild Intent-related Column Gr Remove Empty Wrapper Intent	roup
Auto Intent Name:	BGP Network -	TS BGP (Wrapper)	26-0-114 0X-01	Enable Auto Intent	
Select Device Column:	Device		~ 2	Export Diagnosis Result to CSV View Summary Report	
Add to Auto Intent Profile:			Select 3.gw2	Export Intent Output Map Debug Empty Cells	
1 Enabled Auto Intent 💙		Cancel	OK :303-SW9	Tag Current Column	
Check BGP Config change	Sjc-Co 💿	BGP did not change	Sic-Core-3560x-01	Edit	
Check BGP Config change	IPv6La 💿	BGP did not change	IPv6 ab-R4	Delete Set as Table Key	
Check BGP Config change	Notification			nands to Benc	hmark
Check BGP Config change	En.	abled the Auto Intent succes	sfully!	d	
Check BGP Config change				GP Tro 💿	
	View Det	ails		ОК 4	

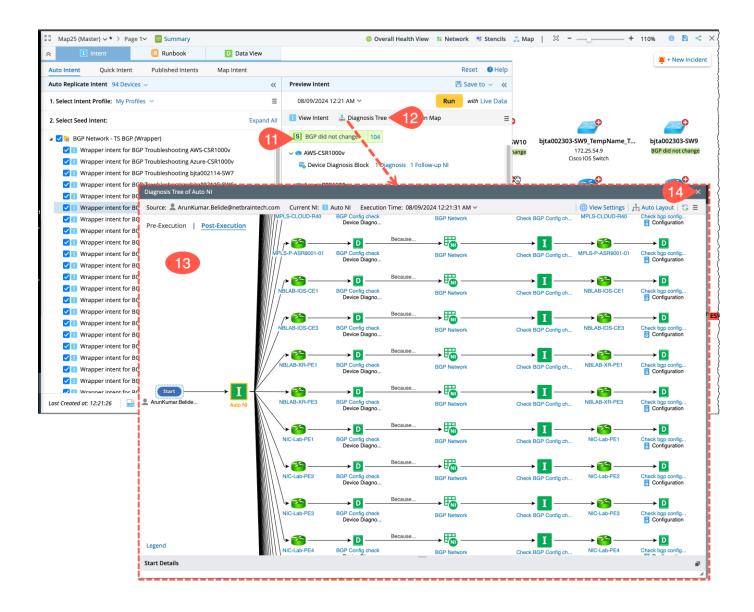
5. Go to **Device Group**> Navigate to **BGP Device Group** >**Draw Devices on Map** and close the device group pane.

Device Group > My Device Groups > BGP D	evices			😮 Help 😋 🕴	🗙 Health View 🗰 Network 🤻 Stencils 🎿 Map   🗵 – + 90% 💿 🖺
Search Q	Search			(	🎉 + New In
🔺 🧮 My Device Groups	Hostname	Vendor	Model	Mgmt IP	
SGP Devices (94)	AWS-CSR1000v	Cisco	CSR1000V	18.223.162.247	
Edit	Azure-CSR1000v	Cisco	CSR1000V	52.255.228.32	Pv6Lab-MPLS
Move	😥 Berlin-vEdge	Cisco Viptela	vedge-cloud	10.8.12.33	IPv6Lab-MPLS
Delete Open Group Map	σ bjta002114-SW7	Cisco	3560E	172.25.54.7	
Draw Devices on Map	a bjta002114-SW	Cisco	3560E	172.25.54.7	ANT TTT REAL TRANSPORT
Draw Group on Map	a bjta002115-5w6	Cisco	3560E	172.25.54.6	eig and a start way
AG_IOS_Nexus	a bjta002302-SW8	Cisco	3560E	172.25.54.8	GOUNDA 172-16-5-15-72 DELAB-XR-PET
🛛 🖿 Ajeet	a bjta002303-SW9	Cisco	3560E	172.25.54.9	000003172.00
Assessment Reference Libr	💋 bjta002303-SW	Cisco	3560E	172.25.54.9	U.
Automation Library     Automation-Tutorial	😁 bjta002439-SW10	Cisco	CGS-MGS-AGS	172.25.54.10	1.81/22
Biel Top 10	💿 bjta002439-SW	Cisco	CGS-MGS-AGS	172.25.54.10	172.16.8.0/22 e0/2 172.16.8.78/22
📄 BT Sample Library	📨 bjta002440-SW11	Cisco	3560E	172.25.54.11	-10.8.78/22
Cisco ACI Devices     Cloud Assessment Referenc	💿 bjta002444-SW13	Cisco	CGS-MGS-AGS	172.25.54.13	PE1
Cloud Assessment Referenc     Cloud Demo	a Bos-Core-6500	Cisco	WS-C3750-2	10.88.0.67	
▶ 🖿 CS	a BOS-N9K-L3OUT	Cisco	N9K-C9372T	192.168.50.21	in a start with the s
📄 Dan Pliilps	😁 Bur-isp-gw1	Cisco	ASR1002-X	192.168.0.27	S S
D GK	😁 bur-isp-gw2	Cisco	ASR1002-X	192.168.0.28	

- 6. In the upper-left corner of the map, go to **Intent** > **Auto Intent**.
- 7. In **Select Intent Profile**, click the down arrow and navigate to the folder location with the auto intent **BGP Network TS BGP (wrapper)** you saved in earlier steps 2 and 3.
- 8. Click the checkbox next to the **BGP Network TS BGP (wrapper).**
- 9. Click **Replicate** and review the list of created intent automation in the **Preview Intent** pane.
- 10. Click Run.

هم Map25 (Master) マ★ > Page 1 → 🗧 Summary		😔 Overall Health View 💠 Network 📑 Stencils 🦨	Map   🗵 – –
Runbook	/		
Auto Intent 6 ick Intent Published Intents Map Intent		Reset 🛛 9 Help	
Auto Replicate Intent 94 Devices 🗸	~~	Preview Intent 🔄 Save to 🗸 💦	
1. Select Intent Profile: My Profiles	= 1	Not Executed Run 10 ive Data	
			bjta002439-SW10
2. Select Seed Intent: Expa	and All	I View Intent 🛔 Diagnosis Tree 💿 Show in Map 📃	172.25 Cisco F
8 BGP Network - TS BGP (Wrapper)		~ ● AWS-CSR1000v	<b>(</b>
✓ I Wrapper intent for BGP Troubleshooting AWS-CSR1000v	- 18	🍣 Device Diagnosis Block 1 Diagnosis	
Vrapper intent for BGP Troubleshooting Azure-CSR1000v	- 18		bjta002114-SV 172.25.54.7
✓ 🚺 Wrapper intent for BGP Troubleshooting bjta002114-SW7	- 16	v 😁 Azure-CSR1000v	Cisco IOS Swite
✓ 1 Wrapper intent for BGP Troubleshooting bjta002115-SW6	- 18	Section 2 Device Diagnosis Block 1 Diagnosis	Cisco F
✓ 1 Wrapper intent for BGP Troubleshooting bjta002302-SW8	- 16	√ 🚅 bjta002114-SW7	
✓ 1 Wrapper intent for BGP Troubleshooting bjta002303-SW9	- 18		~
Vrapper intent for BGP Troubleshooting bjta002439-SW10	- 16	Section 2017 Contract	US-SA
✓ 1 Wrapper intent for BGP Troubleshooting bjta002440-SW11		v 🛹 bjta002115-SW6	10.8.7
✓ 1 Wrapper intent for BGP Troubleshooting bjta002444-SW13		Contraction of the second seco	Cisco F
✓ 1 Wrapper intent for BGP Troubleshooting Bos-Core-6500			
✓ 1 Wrapper intent for BGP Troubleshooting Bur-isp-gw1		√ 🛹 bjta002302-SW8	~
✓ 1 Wrapper intent for BGP Troubleshooting bur-isp-gw2		Source Diagnosis Block 1 Diagnosis	NIC-La
✓ 1 Wrapper intent for BGP Troubleshooting GCP-CSR1000v			10.8.7
✓ 1 Wrapper intent for BGP Troubleshooting gcp-csr1000v-2		v 🛹 bjta002303-SW9	Cisco
Wrapper intent for BGP Troubleshooting Internet	1	육 Device Diagnosis Block 1 Diagnosis	
✓ 1 Wrapper intent for BGP Troubleshooting ip-172-26-0-114			
✓ 1 Wrapper intent for BGP Troubleshooting ip-172-26-0-13		S bjta002439-SW10     Borke Diagnosis Block 1 Diagnosis	2 NB-VxLAN
✓ 1 Wrapper intent for BGP Troubleshooting IPv6Lab-MPLS	1	Solution Contraction Contracti	10.8.8
✓ 1 Wrapper intent for BGP Troubleshooting IPv6Lab-R_IPv4		v 🛹 bjta002440-SW11 :h	Cisco Nex
✓ 1 Wrapper intent for BGP Troubleshooting IPv6Lab-R4		Contraction Contra	
☑ Wrapper intent for BGP Troubleshooting IPv6Lab-R7			<b>S</b>
Wranner intent for RGP Troubleshooting MPI S-CLOUD-P40		v 💿 bjta002444-SW13	IPv6La
Last Created at: 12:12:22	ate	Centre Diagnosis Block 1 Diagnosis	10.1.

- 11. Validate the successful display of the follow up intent result.
- 12. In the Preview Intent page, click **Diagnosis Tree** to trace the execution flow.
- Review the Diagnosis Tree and validate to yourself that you can properly trace the execution of the Auto Intent > Wrapper Intent > Automation Data Tables > Replicated Intents > Diagnosis Logic.
- 14. Close the Diagnosis Tree.



# 9.3 Create frequently used BGP Troubleshooting Intents

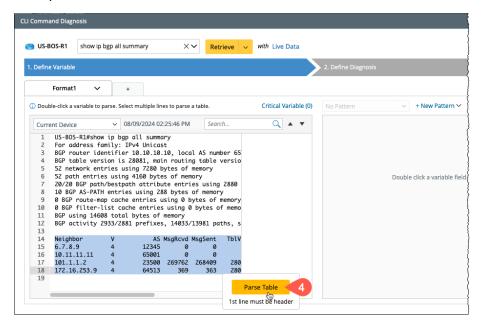
#### 9.3.1 Create NI to Check BGP Neighbor Status Check

In this section, you will create an intent to check the change in BGP Neighbor status:

- 1. Create an Intent from the Intent Manager with Name, **BGP Neighbor Stability Check**.
- 2. Add a seed device from the BGP device group, e.g., **US-BOS-R1**.
- 3. Click +Add CLI Diagnosis to parse the variables and define the diagnosis.

SP Neighbor Stability Check Diagnosis Tree   Run with Live Data   description here     Intent Map: Select     Seed Logic     Replication Logic A   Intent Variables: Manager Tag: + Add
Image: Seed Logic Image: Replication Logic ▲         Image: How the second se
an + Device 2 Intent Variables: Manager   Tag: + Add
US-BOS-R1 Type Description here 🖥 + Add Config Diagnosis 💷 + Add CLI Diagnosi
US-BOS-R1 Type Description here 🖥 + Add Config Diagnosis 🛛 🖾 + Add CLI Diagnos

4. The command will be: *show ip bgp all summary.* The required neighbor data is in tabular format, which can be parsed by the **Table Parser**. Select the table data and click **Parse Table** in the tip window.



5. In the parsed variables **output**, set the variable *sneighbor* as Table key and click **Apply**. The table key will be used to compare two tables, such as the last and baseline tables.

	2. Define Diagnosis									
							Te	est on Dev	vices:	0 ≡
Critical Variable (0)	bgp_nbrs	~ 🖌	Тур	e: Table 🕜	+ New I	Pattern 🗸	/			≡
		PfxRcd ∠	AS emicol	Rcvd MsgS Ion to separate co 65001 0	olumn.	TblVer	InQ Ou	tQ Up/Dov		
	∽ Set Table Colur	nn					-			
Table Header	Column	~	⇔	Variable \$neighbor			Type string	~	ŧ	
First Row										
Last NOW	V	~	⇒	\$v			int	~	Ť	
	AS	~	⇔	\$as			int	~	Û	
	Rcvd	~	⇔	\$rcvd			int	~	Ť	
	MsgSent	~	⇔	\$msgsent			int	~	Û	
	TblVer Output	~	⇔	\$tblver			int	~	ŧ	_
	\$neighbor 🗸	\$v	~	\$as	~	\$rcvd	`	<ul> <li>\$msg</li> </ul>	sent	
Remo		4		65001		0		0		
	s Interface Key s Table Key	4		30000		504848		5048	13	
							Ca	ancel	Арр	oly

- 6. Define Diagnosis:
  - a) If condition: A: Current bgp\_nbrs | Does not equal | Baseline bgp\_nbrs
  - b) **Then**: In case **If** logic is true, define the color (**red**), status (**Error**), and **Message**: *\$this\_device BGP changed from the baseline*!
  - c) Else: In case If logic is not true, define the color (green), status (Success), and Message: \$this\_device BGP did not change.

📄 Add No	ote D	Add Diagnosis	Can al	so click a var	iable on the left to a	dd auto
Name:	Check whe	ther BGP Neighbo	r changed	Anchor	bgp_nbrs.\$neigi	nbor -
	Type descrip	ntion of the diagnos	5İS			
	Table Rows					_
v If	😁 US-BOS-F	R2 Current ~			Baselir	
	bgp_nbrs	v current v	Does not equal	→ bgp	_nbrs	
в	Select Varia			- or		
В	Select varia	ble ~				
∽ Then						
📄 Diag	nosis Messa				Save to Inc	ident
	\$this_device BGP neighbor changed from the baseline!           Image: Set Status Code for Device:					
	Error v \$this_device BGP neighbor changed from the baseline					
		ode for Intent:		16		
•	Error 🗸	stnis_device BC	5P neighbor change	d from the b	aseline	
Add Logi	C~					
✓ Else						<b>D</b>
📄 Diag	nosis Messa	ge:			Save to Inc	ident
	∽ \$th	is_device BGP nei	ghbor did not chang	e.		
<b>~</b> -5	Set Status Co	ode for Device:				
0	Success 🗸	\$this_device BC	5P neighbor did not	change.		
<mark>~</mark> -5	Set Status Co	ode for Intent:				
0	Success $\checkmark$	\$this_device BC	6P neighbor did not	change.		
Add I ==						
Add Logi						
+ Add E	lself					

# 9.3.2 Replicate the Intent to ADT

Launch the **Intent Replication Wizard** from the Network Intent window to replicate the intent:

- 1. In the Seed Intent tab, select the Path-based Replication option.
- 2. **Define ADT**: **Use an Existing ADT** and select **BGP Network**. Enter a relevant Intent group name in the **Replicate Intent to** field.

Intent Replication Wizard - BGP Neighbor Stability Cl	neck	×
Seed Intent Define Al	Replication Settings	Replicate Intent
	Create a New ADT	Use an Existing ADT
Automation Data Table:	BGP Network	Add column
Replicate Intent to:	New Column Group 🛛 🗸 Neighbor Check 🥌	name
Replicate on Device Column:	Device	~
<ol> <li>Selection Mode: Device-based Replication.</li> </ol>		Previous Next

3. Replication Settings:

ent Replication Wizard -	BGP Neighbor Stabili	ty Check		
Seed Intent	Define A	DT Replication	on Settings	Replicate Intent
				Full Settings for Templat
Intent Qualificatio	on: 🧿 via Device Gro	oups/Sites: 1 Device Groups/Fold	ers 面 🛛 🔿 via Dynan	nic Search: Undefined
Define Macro Var	ables and Rules for 1	Their Substitution:		
Item: 1			Add the	
Seed Device		Seed Command	BGP device	Macro Variables
			group	

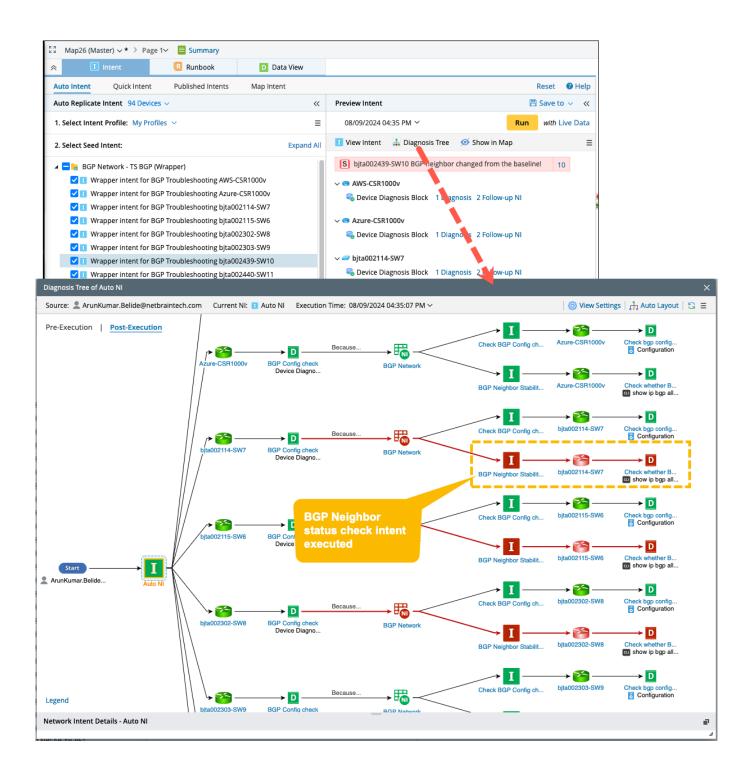
4. Define the column name and tag for the Intent. The added automation column will be:

Seed Intent	Define ADT	Replication	n Settings		Replicate Intent
ADT Columns:					Additional Columns
Column Data	Column Name		Tag		
Replicated Intent	BGP Neighbor Check		1 tag		
		the name to hbor Check		Click and add the tag BGP	

The table will now be populated with devices and the replicated Intents (**BGP Neighbor Check**).

BGP Network Description: Type	Table Bui	Ider Last Updated at:	: 08/09/2024 04:23 PM 🛛 🔌 Rebi	uild Table	BGP tag	~ = d
Items: 53 Rows	) Columns			Intent replicated to all the devices		efined
Configuration	Check BGP Config Change	S Intent Status Code	S Device Status Code	in ADT	BGP Neighbor Check	⋗≡
bgp 65001	Check BGP Config change US-SF	BGP did not change	US-SFO-R1	W	BGP Neighbor Stability Chec 💿	·
bgp 65001	Check BGP Config change US-BO •	BGP did not change	US-BOS-R1	Wrapper intent for BGP Troubles 🧿	BGP Neighbor Stability Chec 🔍	*
bgp 65001	Check BGP Config change US-SF •	BGP did not change	US-SFO-R2	Wrapper intent for BGP Troubles 💿	BGP Neighbor Stability Chec 🧿	•
bgp 65501	Check BGP Config change NIC-La •	BGP did not change	NIC-Lab-PE2	Wrapper intent for BGP Troubles 💿	BGP Neighbor Stability Chec 💿	•
bgp 64513	Check BGP Config change ip-172 •	BGP did not change	ip-172-26-0-114	Wrapper intent for BGP Troubles	BGP Neighbor Stability Chec	•
bgp 64550	Check BGP Config change PE-36 •	BGP did not change	PE-3600X-01	Wrapper intent for BGP Troubles	BGP Neighbor Stability Chec	•
bgp 100	Check BGP Config change VRF-PE2 ④	BGP did not change	VRF-PE2	Wrapper intent for BGP Troubles	BGP Neighbor Stability Chec ④	F
bgp 65145	Check BGP Config change bjta00 •	BGP did not change	bjta002440-SW11	Wrapper intent for BGP Troubles	BGP Neighbor Stability Chec 💿	×.
bgp 64661	Check BGP Config change bur-is •	BGP did not change	bur-isp-gw2	Wrapper intent for BGP Troubles	BGP Neighbor Stability Chec 💿	•
bgp 65501	Check BGP Config change NIC-La •	BGP did not change	NIC-Lab-XR1	Wrapper intent for BGP Troubles		
bgp 65000	Check BGP Config change bjta00 •	BGP did not change	bjta002303-SW9	Wrapper intent for BGP Troubles	BGP Neighbor Stability Chec ④	

5. Now, execute the wrapper intent on BGP devices with auto intent, repeating the same detailed in Section 9.2.3 to verify the **BGP Neighbor Check** (tagged BGP) is also executed.



## 9.3.3 Create NI to Check BGP Route Reflector Client

In this section, you will create an intent to check whether the status of the route reflector client is configured, and if so, ping each reflector client:

- 1. Create an Intent from the Intent Manager with Name, **BGP Route Reflector Client Configured**.
- 2. Add a seed device from the BGP device group, e.g., **ISP-P02**.
- 3. Click +Add Config Diagnosis to parse the variables and define the diagnosis.

Network Intent (Edit Mode	2)			×
BGP Route Reflector	Client Configured	Diagnosis Tree	Run with Live Data	Save 🛛 🕄 Help 📃
Type description here				🏅 Intent Map: Select 🗸
I Seed Logic	🔟 Replication Logic 🛕			
🗃 + Device			Intent Variables: Mar	nager   Tag: + Add 🥅
🗸 💿 ISP-P02 🔁		Type Description here	🗟 + Add Config Diagnosis	+ Add CLI Diagnosis
No content has been adde	d.			

- 4. Click **Retrieve** to collect the data from Live Data. The required neighbor data is in paragraph format. Create the Paragraph Parser to parse the neighbor:
  - a) Define the ID Line as ^ neighbor \$neighbor route-reflector-client.

Define Variable					2. Define Diagr			
Format1	~	+						Test on Devices: 0
Double-click a v	ariable to pars	e. Select multiple lines to pa	rse a table.	Critical Variable (0	) - Froute_refle	ctor_clients 👻 者 Type: Paragra	ph 🍘 🛛 + New Patte	ern ~ =
Current Device		₩ 06/14/2024 06:36:04	PM Search	ا م ا	D Line A 🔷	neighbor <b>\$neigh</b>	or route-re	eflector-clien
91 neig	nbor 12.0.0	0.2 remote-as 10000 0.2 remote-as 10000 0.2 remote-as 10000				96 neighbor 4.0.0.1 route-	reflector-client	> 4 Lines
96 🔺	neighbo	r 4.0.0.1 act r 4.0.0.1 rou	te-reflector-	-client	+ Field ~			
98 🔺 🛛	neighbo	r 11.0.0.2 ac r 11.0.0.2 ro	ute-reflector	-client	Output	+ Parse Lines	<b>^</b>	_
		r 12.0.0.2 ac <sup>.</sup> r 12.0.0.2 ro		r-client	\$neighbor 4.0.0.1			~
		r 13.0.0.2 ac		r-client	11.0.0.2			
		ress-familv			12.0.0.2			

- b) Parse all the multiple lines with *route-reflector-client* using **+Parse Lines** to check whether the route reflector client is configured.
- c) Add the name: *route\_reflector\_client\_config*.
- d) Parse the lines using the **line of variable**: select the variable **\$neighbor.**
- e) Click **Apply** to save and close the **Parse Lines** dialog.

1. Define Va	ariable			2. Define	Diagnosis	
Fo	ormat1 🗸 +				Test on Devices: 0	≡
i Double	Parse Lines			×	reflector_clients 👻 者 Type: Paragraph 🚷 🔹 + New Pattern 🗠	=
Current 90 91	Name: route_reflector	_client_config			Line A ^ neighbor \$neighbor route-reflector-client	
d	• The line of variable:	\$neighbor		~	96 neighbor 4.0.0.1 route-reflector-client > 4 Lines	
94 95	O The line between:		✓ to Select	~		
96 4 97	O The line contains key	word: Enter keyword				
98 99 100	Output:				+ Parse Lines b -	-
101	\$neighbor	~	<pre>\$route_reflector_client_config</pre>	~		
103 104	4.0.0.1		neighbor 4.0.0.1 route-reflector-client	t	A CONTRACT OF	
105	11.0.0.2		neighbor 11.0.0.2 route-reflector-clien	nt	¥.	
107	12.0.0.2		neighbor 12.0.0.2 route-reflector-clien	nt		
	13.0.0.2		neighbor 13.0.0.2 route-reflector-clier	nt		
	Pattern: LinesByVar	<pre>iable[\$route_refle</pre>	<pre>ctor_client_config]:\$neighbo</pre>	r		
			¢ e	Apply		

5. Rename the parser as **route\_reflector\_clients**, and the final output would be:

			2. Define Diagnosis
+			Test on Devices: 0
elect multiple lines to parse a ta	ble.	Critical Variabl	5 route_reflector_clients 🗸 Z Type: Paragraph 👔 🛛 + New Pattern 🗡 🚍
06/14/2024 06:36:04 PM	Search	۹ 🔺	
route-reflector-clien 2 activate 2 route-reflector-clien 2 activate 2 route-reflector-clien 2 activate 2 route-reflector-clien ly 1 nd	nt nt	P1-ID Line A Var Line 1 P2-ID Line A Var Line 1 P3-ID Line A Var Line 1	ID Line A       neighbor \$neighbor route-reflector-client       =         96       neighbor 4.0.0.1 route-reflector-client       > 4 Lines         Var Line 1       LinesByVariable[\$route_reflector_client_config]:\$n       =         96       neighbor 4.0.0.1 route-reflector_client       > 4 Lines
server		P4-ID Line A Var Line 1	+ Field ~ Output + Parse Lines
server nity *******			\$neighbor     \$route_reflector_client_config       4.0.0.1     neighbor 4.0.0.1 route-reflector-client
			11.0.0.2 neighbor 11.0.0.2 route-reflector-client
			12.0.0.2 neighbor 12.0.0.2 route-reflector-client
			13.0.0.2 neighbor 13.0.0.2 route-reflector-client

- 6. Define Diagnosis 1: Check whether the route reflector client is configured as follows:
  - a) **If** condition: A: *route\_reflector\_clients* | Is not empty.
  - b) **Then**: In case **If** logic is true, define the color (**green**), status (**Success**), and Message is *Route reflector client is configured*.
  - c) **Else:** In case **If** logic is not true, define the color (**grey**), status (**Error**), and Message is *Route reflector client is not configured*.

		2. Define Diagnosis
earch	۹ 🗸 م	Add Note • Add Diagnosis Can also click a variable on the left to add automation
-client pr-client pr-client	Check whether rout	Name: Check whether route reflector client is configured Anchor: route_reflector_clie v
r-client	·	Type description of the diagnosis
		☐ Loop Table Rows ~ If
	a	A  ISP-P02 Current  Is not empty  Is not emp
		B Select Variable
		✓ Then     ☐ Diagnosis Message: Save to Incident ≡
		✓ Route reflector client is configured
	b	✓ Set Status Code for Device:
		Success V Route reflector client is configured
}		✓ Is Set Status Code for Intent:         ✓ Success ✓         Route reflector client is configured
<pre>}</pre>		
}		Add Logic ~
<pre>}</pre>		✓ Else
{		Diagnosis Message:
}		✓ Route reflector client is not configured
}	С	Set Status Code for Device:
}		Error      Route reflector client is not configured
ł		Set Status Code for Intent:
{		Error      Route reflector client is not configured
}		Add Logic ∼

- 7. Define Diagnosis 2: **Ping** the route reflector client if it is configured:
  - a) **If** condition: A: *neighbor* | Is not empty.
  - b) **Then**: Add the *Ping* **<***destination ip* **> NIT created in Chapter 4 as a follow-up Intent.**

		Diagnosis			on the left to add a	utomation
Name:	Ping the route ref	flector client		Anchor:	route_reflector_c	lie 🗸
	Type description o	f the diagnosis				
✓ Loop ✓ If	Table Rows 📕	route_reflector_client	s 🗸 🛛 Tabl	e Key: Pleas	e Select 👻 🤨	Ø
A	s ISP-P02	Current 🗸				
	neighbor	✓ Is not e	empty	~		Î
В	Select Variable	~				
✓ Then						
	w-up Intent: O No	etwork Intent emplate: Ping Check	ent Intent (	(Self) 🔿 S	Stop	≡
Add Logi	c~					
+ Add E	ilself + Add I	Follow-up Intents	•			
		0 Follow-up Intents	: + Follow	v-up 🗸 🚯		_
			Select	t Intent (Stan	dalone)	_
				t Intent Tem		
				t Intents fror	n ADT blates via ADT	
				t Intent Clust		

- 8. In the Follow-up Intents window:
  - a) Define Replicate Current Intent to: **Device by Varible** | **this\_device**.
  - b) Set macro Variable **dest\_ip** to be the Variable \$*neighbor.*

	v-up Inten		-				
~ 1	Ping Che	<b>ck</b> (Intent Template)				Ē	l
Des	cription:	When					
	Replicatio	on Settings:					
	Replicate	Current Intent to:	Device by Variable	~	this_device ~	1	
Set Macro Variables of Seed Ir							
Seed Device		ed Device	Macro Variable	Туре	Set Variable		
	▲ Seed Device						
		b	dest_ip	string	neighbor (route_reflector_clients)		
						1	
	Merge	multiple replicated	intents into one		0/1 Device Key Set for Selected Table		
1		Execution: Setting	S				
	Follow-up						
		llow-up intents			_		

9. Close the diagnosis window and go back to the **Network Intent** window to replicate to **ADT BGP Network**.

# 9.3.4 Replicate the Intent to ADT

Launch the **Intent Replication Wizard** from the Network Intent window to replicate the intent:

- 1. In the Seed Intent tab, select the Path-based Replication option.
- 2. **Define ADT**: **Use an Existing ADT** and select **BGP Network**. Enter a relevant Intent group name in the **Replicate Intent to** field.

Seed Intent Define ADT Replication Settings Replicate Intent
Create a New ADT Use an Existing ADT Add the column
Automation Data Table: BGP Network
Replicate Intent to: New Column Group V Reflector Client Status Check
Replicate on Device Column: Device 🗸
Selection Mode: Device-based Replication.     Previous     Next

3. Replication Settings:

ent Replication Wizard - BGP Route Reflect	or Client Configured		
Seed Intent	Define ADT		Replication Settings
Intent Qualification: 🧿 via Device (	roups/Sites: 1 Device Groups/	Folders 🖻 🛛 🔿	via Dynamic Search: Undefined
Define Macro Variables and Rules fo ltem: 1	r Their Substitution:	Add the BG device grou	
Seed Device	Seed Command		Macro Variables
US-BOS-R1	o Configuration		
			~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

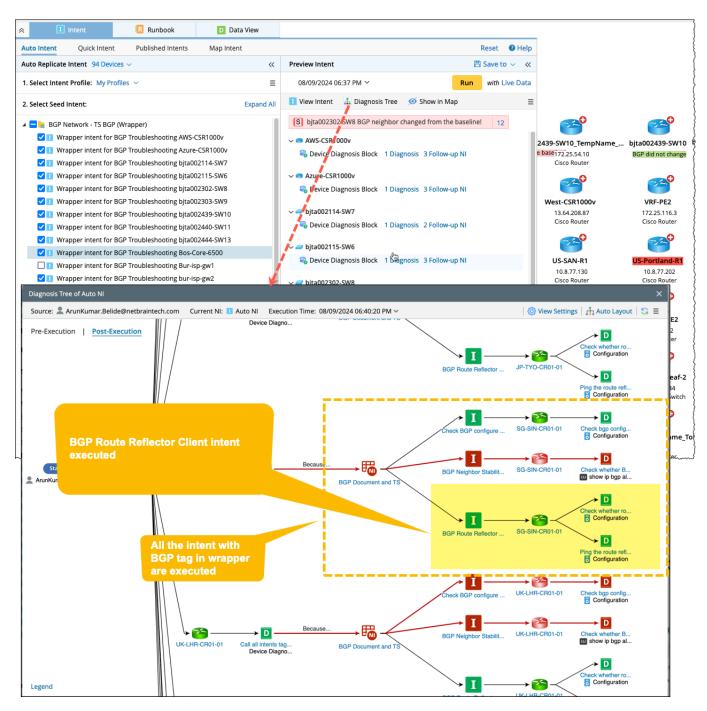
4. Define the column name and tag for the Intent. The added automation column will be:

t Replication Wizard - BGP I	Route Reflector Client Configured			
Seed Intent	Define ADT	Replication Setti	ngs	Replicate Intent
ADT Columns:				Additional Columns ~
Column Data	Column Name		Тад	
I Replicated Intent	Check Reflector Client		1 tag	
s Intent Status Code	Route Reflector Client Configur	ed or Not		
	Default colum modifie			Added the tag BGP

The table will now be populated with devices and the replicated Intent (*Check Reflector Client*).

🚯 BGP N	etwork	Table Builder	st Update	d at: 08/09/2024 06:33 PM	Rebuild Table			Add Data Manually 🗸 📃	- <b>-</b>
Descriptio	on: Type description here					BGP			
Items: 53	Rows 11 Columns					tag Seurch	9.1	Advanced Filter: Undefined	
s Code	S Device Status Code	TS BGP (Wrapper)		E DCD Mainthear Charle	D	Check Reflector Client	⊘ ≡	S Route Reflector Clie	=
ange	US-SFO-R1	Wrapper intent for BGP Troubles.	Re	olicated Intent to		BGP Route Reflector Client Conf	ï 💿		
ange	US-BOS-R1	Wrapper intent for BGP Troubles	the	ADT	s 💿	BGP Route Reflector Client Conf	ì 💿		
ange	US-SFO-R2	Wrapper intent for BGP Troubles		BGP Neighbor Stability Check	US 💿	BGP Route Reflector Client Conf	ĩ 💿		
ange	NIC-Lab-PE2	Wrapper intent for BGP Troubles.	. •	BGP Neighbor Stability Check	NI 💿	BGP Route Reflector Client Conf	ĩ 💿		
ange	ip-172-26-0-114	Wrapper intent for BGP Troubles.	. •	BGP Neighbor Stability Check	ip 💿	BGP Route Reflector Client Conf	î 💿		
ange	PE-3600X-01	Wrapper intent for BGP Troubles.	💿	BGP Neighbor Stability Check	PE 💿	BGP Route Reflector Client Conf	ĩ 💿		
ange	VRF-PE2	Wrapper intent for BGP Troubles.	💿	BGP Neighbor Stability Check	VR 💿	BGP Route Reflector Client Conf	ĩ 💿		
ange	bjta002440-SW11	Wrapper intent for BGP Troubles.	. •	BGP Neighbor Stability Check	bjt 💿	BGP Route Reflector Client Conf	ĩ 💿		
ange	bur-isp-gw2	Wrapper intent for BGP Troubles.	. •	BGP Neighbor Stability Check	bu 💿	BGP Route Reflector Client Conf	î 💿		
ange	NIC-Lab-XR1	Wrapper intent for BGP Troubles.	💿			BGP Route Reflector Client Conf	ĩ 💿		
ange	bjta002303-SW9	Wrapper intent for BGP Troubles.	💿	BGP Neighbor Stability Check	bjt 💿	BGP Route Reflector Client Conf	ĩ 💿		
ange	Sjc-Core-3560x-01	Wrapper intent for BGP Troubles.	. •	BGP Neighbor Stability Check	Sjc 💿	BGP Route Reflector Client Conf	ĩ 💿		
ange	IPv6Lab-R4	Wrapper intent for BGP Troubles	. 0	BGP Neighbor Stability Check	IPv 💿	BGP Route Reflector Client Conf	i 💿		

5. Now, execute the wrapper intent on BGP devices with auto intent, repeating the same detailed in Section 9.2.3 to verify the **BGP Route Reflector client** (tagged BGP) is also executed.



# 9.4 Create a chatbot for the BGP TS wrapper intent

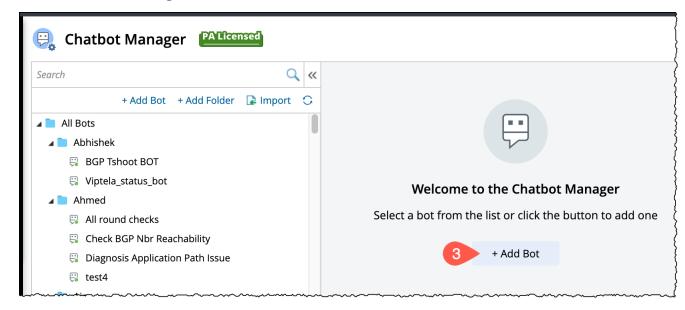
Chatbot allows power users to build interactive chatbots, so end users can execute intent-based automation to solve real-world challenges without accessing NetBrain system UI. Building a chatbot flow is straightforward.

#### 9.4.1 Create the Netbrain Chatbot

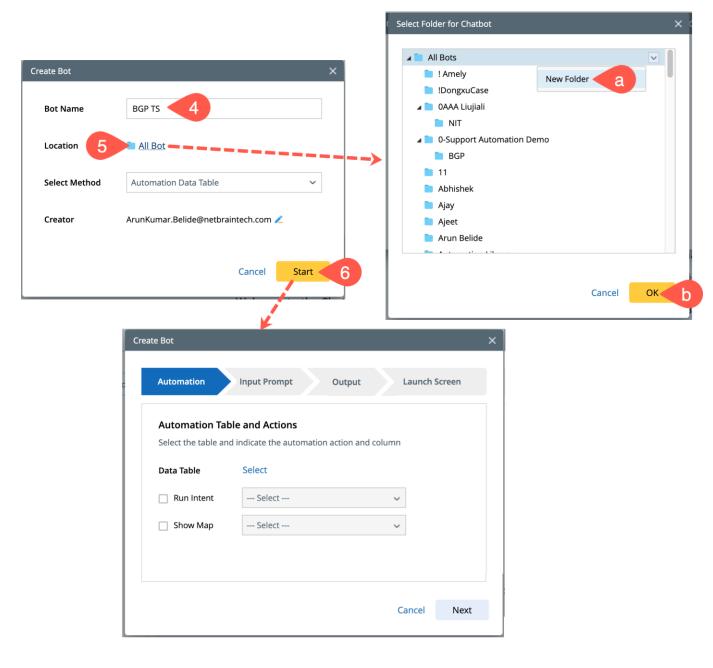
- 1. In the Upper-Left corner of the desktop, click <sup>III</sup>.
- 2. Select Chatbot Manager under the Intent-Based Automation tab.

::: <b>1</b> etBrain	
Search App	🧠 Domain Management
Network Map-Based Automation	Intent-Based Automation Incident 8
Intent	Execute Intent
🎝 Intent Manager	4 Intent Based Automation Center
🔩 Intent Cluster Manager	🗟 Triggered Automation Manager
🖽 Automation Data Table (ADT)	💪 Preventive Automation Manager
🗔 Task Variable Manager	Schedule Automation
	2 🔍 Chatbot Manager 🔶 📌
	😫 Preventive Automation Console

3. In the **Chatbot Manager**, click **+ Add Bot**.



- 4. Name the bot **BGP TS.**
- 5. Location: To choose a location for the bot, click **All Bot**.
  - a) In the dialog, **Select Folder for Chatbot**, select All Bots or create a folder under it using the option **New Folder** from the dropdown menu.
  - b) Click **OK.**
- 6. Click **Start** to begin the **Chatbot creation wizard**.



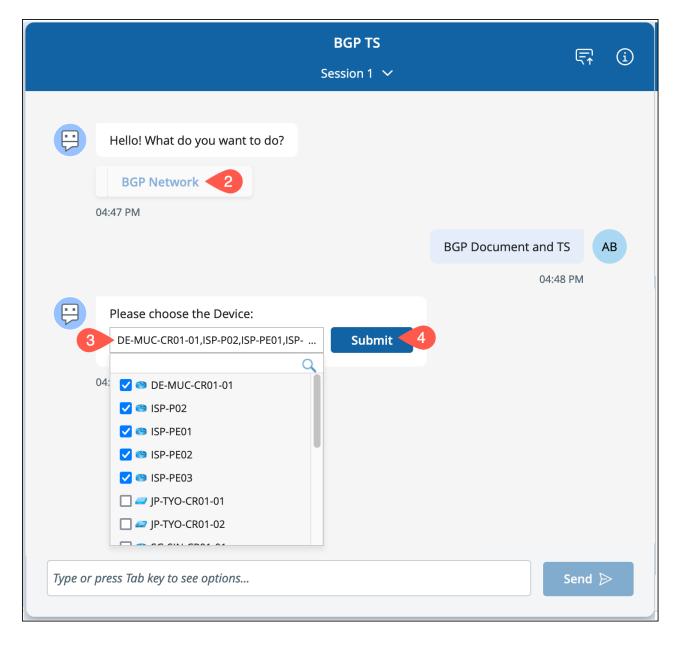
- 7. In the first step **Automation**, click **Select** located next to **Data Table** and choose the ADT **BGP Network.** 
  - a) Check the selection box of the **Run Intent** field and select the intent **TS BGP (Wrapper)** from the dropdown.
  - b) Click **Next** to go to the next ribbon Output.

Creat	te Bot				×
	Automation	Input Prompt	Output	Launch Screen	
	Automation Tal		d actument		
	Data Table	d indicate the automation action and My Tables/BGP Network			
8	Run Intent	Check BGP Config Change	~		
	Show Map	Check BGP Config Change			
		TS BGP (Wrapper) BGP Neighbor Check			
		Check Reflector Client			
				Cancel Next	
				Cancel Next •	9

- 8. In the second step **Input prompt**, check the **Device** selection box for input and enable the multiple selection option.
- 9. In the third step **Input prompt**, review the automation output message and customize it if needed using the pencil icon.
- 10. The fourth step provides the option to personalize the logo and description on the launch screen. Do the modifications if needed or leave the default set up and click **OK**.

## 9.4.2 Use Chatbot

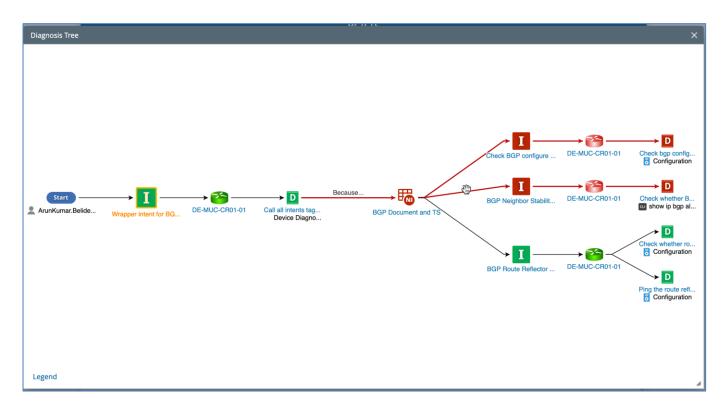
- 1. Click Launch Bot > START.
- 2. Click the Intent **BGP Network** (appears as the ADT table is BGP Network).
- 3. Under **Please choose the Device**, select multiple devices from the drop-down as required.
- 4. Click **Submit** to execute the intent.



5. The chatbot will execute the intent and provide the diagnosis output.

NOTE: Click the **Map** and **View Diagnosis tree** links to see the error details.

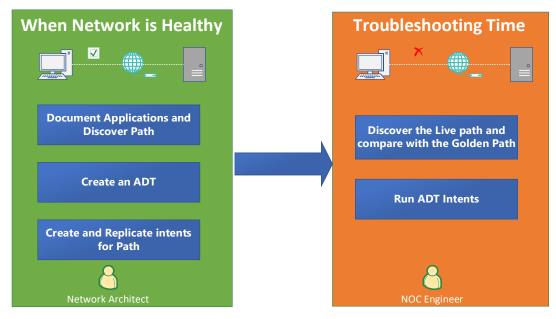
B	GP TS 등 (
Devoices selected for wrapper execution	DE-MUC-CR01-01, ISP-P02 AB
	04:55 PM
<ul> <li>"Wrapper intent for BGP Troubleshooting ISP-P02" ex 1 errors found on device ISP-P02.</li> <li>YES BGP does not change ISP-P02 BGP neighbor changed from the baseline Ping succeeded</li> </ul>	Automation output message
Map from Intent: <u>Intent Output Map</u> View Diagnosis Tree: <u>Diagnosis Tree</u>	
"Wrapper intent for BGP Troubleshooting DE-MUC-CR 2 errors found on device DE-MUC-CR01-01.	01-01" executed.
DE-MUC-CR01-01 BGP neighbor changed from the baseli NO BGP configuration changes. Missing lines: None. Extra lin bgp router-id 192.168.11.254 bgp log-neighbor-changes redistribute ospf 1 route-map O2B neighbor 100.20.1.1 remote-as 10000 neighbor 100.20.1.1 description Connection_to_ISP-PE02 neighbor 100.20.1.1 weight 200	Click Diagnosis Tree to see intents and its follow up in tree view
Map from Intent: <u>Intent Output Map</u> View Diagnosis Tree: <u>Diagnosis Tree</u>	
This is the last step. You can type "restart" to start over. 04:55 PM	
pe "/" for shortcut commands	Send <b>&gt;</b>



6. Close all the browser tabs except for the **NetBrain desktop.** 

# **10 Application Path Assessment and Troubleshooting**

This chapter explains how to use Intents to troubleshoot path-related network issues. Firstly, you document the network path when it is functioning correctly to establish a Golden Baseline. Next, you will create an ADT for critical applications. Then, you replicate the intents that you think are useful to monitor and troubleshoot the applications to the ADT. And you can create a wrapper intent as the user interface. Finally, when you troubleshoot the path-related issues, you will discover the path and compare it with the Golden baseline for any difference and run the wrapper intent to find any error.



The intent-based path assessment can answer the following questions:

- Is the Path changed visually? Compare the cached path (when the network is healthy), golden baseline path, and live Path in the troubleshooting stage.
- Is the Path failing over programmatically? Use the intent to check for the routing next-hop change and the CAM table (for L2 failover) change.
- Is the Path healthy performance-wise? Use the intent to analyze utilization change, CPU/memory change, link error change, QoS buffer drop change, etc.
- Is Path configured properly? Use the intent to check the QoS configuration consistency across devices of the path and configuration consistency between the failover device pair for the Path.

In this chapter, you will learn how to replicate a seed intent with **path-based replication logic** (you use the device-based replication logic up to now).

# 10.1 Document the application path when the network is healthy

Note: To use the full functions of the **Path Browser**, you need to purchase an **AAM license**. Otherwise, you cannot create a new application and only can save the path in "Untitled Application".

This section teaches how to define a map path for an application using **Path Browser**. You can manage this path based on the Application in the **Path Browser**.

Follow the step-by-step instructions to add paths in the **Path Browser**.

- 1. In the quick access toolbar, click **Path** to open the **Path Browser** window.
- 2. Create an Application to add paths in the application.

	NetBr	ain Next-Gen	Search Anything and Cl	reate Map	ي ال						
$(\pm)$	Path Bro	wser	2			(9 Help 'G ₹ X					
	+ Nev	w Application	v Path 🕞 Import 🔓	Export 🗸		Applica	tion Manager				
Recents	Total E	Entries: 5 Applications Application Details	9 Paths, 6 Succeeded	d 1 Failed		Search ×					
Network	4	Name:				Last Result	Result Cat				
Files		Description:	Type in description for	this Application here		O Running					
Site	4					Succeeded	Result Cat				
<mark>⊘</mark> ⁰ Path	1					Succeeded					
	<b>-</b>					<u> </u>	/				
		Related Devices: + Add Devices	Device	Weight							
		T Add Devices									
				Canc	el OK	C					

- a) Click **New Application** to open the **Application details** window.
- b) Enter the application Name, i.e., *Cookbook*. The Description is optional.
- c) Click **OK** to close the window. The **Cookbook application** will be created in the **ApplicationManager**. You can create a path under this Application.
- 3. Click **New Path** to define the path in the **Path Details** window.

$\oplus$	Path Browser		Help 5	* × }	
Recents	+ New Application + New Path 3 or Total Entries: 5 Applications 9 Paths, 6 Succe	t 🕞 Export 🗸	Application Mai	anager	
Network Files	Applicati Path Source     Cookbook     No Path	Destina     Group     Protoc	ol Last Result Resu	Select Application	×
Path Details	Patabase_path		×	Cookbook Testing SampleApp Top Critical Apps	
Application: C	ookbook	Browse		2 Ontitled Application	
	ath Type:  Unicast  Multicast US-BOS-Host1	→ 3 10.101.1.100			
	ateway: HSRP.1(10.1.1.1)				
Parameters: Co	nfigure			+ New Application Cancel OK	
			Cancel OK e		

- a) Enter **Path Name, i.e.,** Database\_path.
- b) Select an application (**Cookbook**) for the path using the **Browse** button.
- c) Enter Source IP and Destination IP Addresses (e.g., Source IP: 10.1.1.100, and Dest IP: 10.101.1.100). The related gateways will be auto-identified per your input. You can select the target one from the Gateway list if the device has multiple gateways.
- d) Select a **Protocol** from the **dropdown** list.
- e) Click **OK** to create a path. You can view your created path (**Database\_path**) under the **Cookbook** application.

4. Right-click on the created Path and click **Calculate Live Path** from the menu.

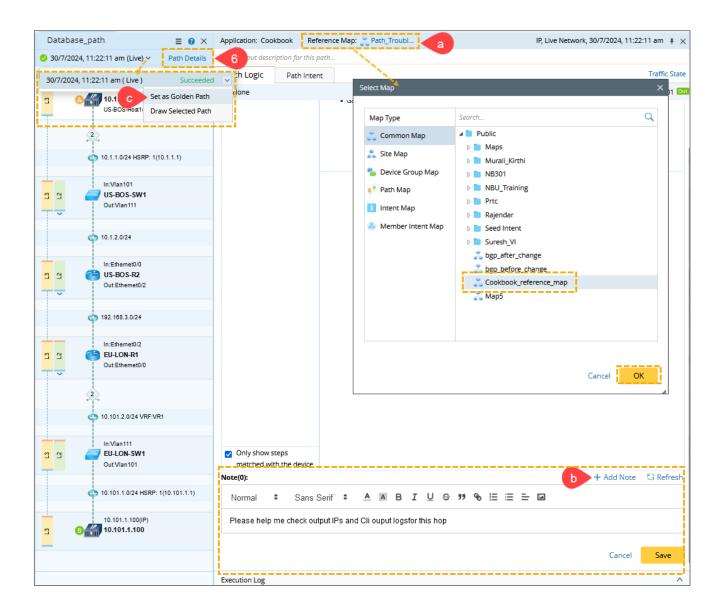
P	ath Brow	wser							<b>(</b> ) H	elp E	3 8	
	+ Nev	v Application	+ New Path 🔓 Import 🔓	🖁 Export 🗸	,				Applica	tion M	lana	ger
Ì	Total E	ntries: 5 Appl	lications, 10 Paths, 7 Succeed	led 1 Faile	ed			Sea	rch		(	2
		Applicati	Path	Source	0	Destina 🔒	Group	0	Protocol	Last	Resi	ult
1		Cookbook										
Ē			U Database_path	US-BOS-	н	10.101.1.1	Draw Lat	test P	ath on Map			d
5	4	Testing							Path on Map			
			U Path1	192.168.	3	19 4	Calculate	e Live	Path			d
ł			U Reference Path	US-BOS-	S	US-BOS-S	Calculate	e Pati	n by Advanced	Optio	ns	d
		SampleApp					Resolve I	Path	Gateway			
			No Path Defined				Add to P	ath C	alculation Bar			
		Top Critic					View Pat	h Ove	erview			
			U BOS-Host1 to LON-Ho	10.1.1.10	00	10.101.1.1	View Pat	h His	tory			d
L.		Untitled					Edit					
\$ }			U test	1.1.1.1		EU-MAD-R1	Move to					d
ł			US-EU	10.4.1.10	)5	10.104.1.1	Delete					d
5			US-EU2	10.4.1.10	00	10.104.1.1	Open Re	ferer	ice Map			d
			U test2	1.1.1.1		10.8.86.131	-		ence Map			
~				MIS-BOS-	H	.8.8.8.8	Delete R	efere	nce Map			L

You can wait to load the path until you see the **Succeeded** status in the **Last Result** column.

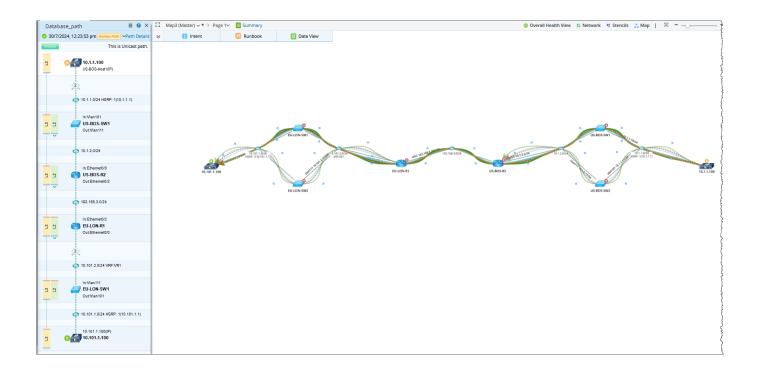
5. Double-click the path to check whether the path is healthy.

h Bro	wser					0	Help 🕄 🖡 🗙		base_pa	
										24, 19:43:51 (Live) Y Path Deta
- Ne	w Application	+ New Path 🗣 Import	Export ~			Appli	cation Manager	Unic	əst	This is Unicast pa
Total I	Entries: 5 Appli	ications, 10 Paths, 7 Succeed	ed 1 Failed		1	Search	Q	3		10.1.1.100
	Applicati	Path	Source 0	Destina 🜖	Group	Protocol	Last Result	-		US-BOS-Host1(IP)
						-				
4	Cookbook					IP				2
	5	🕨 🛛 Database_path	US-BOS-H	10.101.1.1		IP	Succeeded			10.1.1.0/24 HSRP: 1(10.1.1.1)
4	Testing									
		U Path1	192.168.3	192.168.3		IP	Succeeded	3	2	In:Vlan101
		u Reference Path	US-BOS-S	US-BOS-S		IP	Succeeded	-	-	Out:Vlan111
4	SampleApp								~	
		No Path Defined								10.1.2.0/24
4	Top Critic									
		BOS-Host1 to LON-Ho	10.1.1.100	10.101.1.1		IP	Succeeded			In:Ethernet0/0
	Untitled	-						ព	2	Out:Ethernet0/2
		U test	1.1.1.1	EU-MAD-R1		IP	Succeeded	T	~	
		US-EU		10.104.1.1		IP	Succeeded			192.168.3.0/24
		_								T
		US-EU2		10.104.1.1		IP	Succeeded			In:Ethernet0/2
		u test2	1.1.1.1	10.8.86.131		IP	N/A	ព	21	EU-LON-R1 ! Out.Ethernet0/0
		U chaman	US-BOS-H	8.8.8.8		IP	Failed	1	~	
		U Sa_test	10.1.1.100	10.101.1.1		IP	N/A			2
										10.101.2.0/24 VRF:VR1
									_	In:Vlan111
								5	5	EU-LON-SW1
								1	-	Out:Vlan101
										10.101.1.0/24 HSRP: 1(10.101.1.1)
										10.101.1.100/-
								3	6	10.101.1.100(IP)
								-		- MARY

- 6. Click **Path Details** to view the path details for each hop and define the reference map.
  - a) Select the **Reference Map** from the **Select Map** window.
  - **Note**: You can define a reference map for a saved path. The path reference map allows you to draw the golden baseline path on the map and use device notes to draw key configlets or troubleshooting recommendations on the map.
  - b) Click + Add Note in the bottom-right corner, enter a note and click Save.
  - c) Set this path as a **Golden Path**.



Now, you can see the **Golden Path** drawn on the map.



# **10.2Build and Manage Automations for Path**

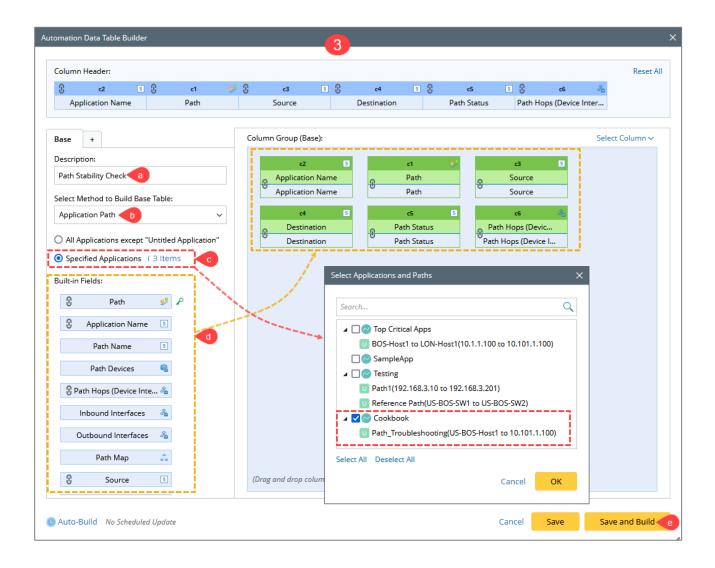
In this section, you will create a base ADT from the Application path. Furthermore, you will use the **Intent Replication Wizard** to add several Intents and a wrapper Intent to the ADT. Additionally, you will create a wrapper Intent for path-related automation and enable the auto intent functionality.

## **10.2.1 Create the ADT base table from the path browser**

The system provides the method **Application Path** for you to import the application and path data into the ADT base table.

Follow the step-by-step instructions to build the base table with the data in devices:

- 1. Build the new base table by entering the Name, Path Stability Check and select the Location as per your preference.
- 2. Click **Table Builder** and define the base table using the method **Application Path**.
- 3. You can build an ADT table with the following columns: **Application Name**, **Path**, **Source**, **Destination**, **Path Devices**, **Path Status**, and **Path Hops**.



#### The ADT will be:

🛱 Path Stability Check Table Builder Lost Updated at: 08/09/2024 12:57 PM 🔍 Rebuild Table Add Data Manual									
Descri	Description: Type description here								
Items:	Items: 4 Rows 11 Columns Search Q 🕎 Advanced Filter: Un								
No.	S Application Name	9º Path	号 Path Devices	Source	S Destination	S Path Status	🖧 Path Hops (Device In }		
1	Cookbook	Database_path	10.1.1.100	US-BOS-Host1	10.101.1.100	Succeeded	EU-LON-R1 - Ethernet0/		
2	Testing	Path1	192.168.3	192.168.3.10	192.168.3.201	Succeeded	192.168.3.10 - Ethernet		
3	Testing	Reference Path	US-BOS-S	US-BOS-SW1	US-BOS-SW2	Succeeded	US-BOS-SW1 - Loopback		
4	Top Critical Apps	BOS-Host1 to LON-Host1	10.1.1.100	10.1.1.100	10.101.1.100	Succeeded	EU-LON-R1 - Ethernet0/		

## **10.2.2 Replicate the Intents of Device Health Check**

In this section, you will replicate all the intents created in Chapter 3 for Basic Device Health Check. The intents you will replicate in the ADT will be:

- CPU usage check (Cisco IOS).
- Interface status check (Cisco IOS).
- Uptime check (Cisco IOS).

NOTE: Ensure when you use Intent Replication Wizard to replicate intent, define Path-based Replication instead of Device based replication.

#### 10.2.2.1 Replicate CPU Usage Check (Cisco IOS) Intent to the ADT

- 1. From the Intent, go to the menu and click **Intent Replication Wizard**.
- 2. Seed Intent. Select the **Path-based Replication** option.

Intent Replication Wizard - CPU Usage Check (Cisco IOS)		
Seed Intent	Define ADT	Replication Settings
Seed Intent:	CPU Usage Check (Cisco IOS)	Select Last Moo
Intent Template for:	O Device-based Replication	Path-based Replication

3. Define ADT.

ntent Replication Wizard - CPU Usage Check (Cisco IOS)	
Seed Intent Define	ne ADT 3 Replication Settings
Create a New Al	ADT Use an Existing ADT a
Automation Data Table:	e: Path Stability Check
Replicate Intent to:	o: New Column Group 🗸 check cpu 🕑
Replicate on Path Column:	n: Path v 9º Path C
Select Automation	on Data Table X
Search	٩
> Assess	Tables Micro_Learning ssment Reference ADT d Assessment Reference ADT
Cookb	
▷ Instruc ▷ Instruc	uctor Materials _Materials
	Cancel OK

# 4. Replication Settings.

Intent	Replication Wizard - CPU Usage Check (Ci	sco IOS)		×			
	Seed Intent	Define ADT	Replication Settings	Replicate Intent			
				② Full Settings for Template			
	Define Macro Variables and Rules for Their Substitution: Item: 1						
	Seed Device	Seed Command	Macro Variables	Command Qualification			
	ave US-BOS-SW1	show process cpu		Defined 🗑			
	More Replication Settings 🗸						
🚹 Sel	lection Mode: Path-based Replication, AD	T: Path Stability Check.		Previous Next			

5. Replicate Intent and add the tag application or path.

Inten	t Replication Wizard - CPU Usage Check (C	isco IOS)		×
	Seed Intent	Define ADT	Replication Settings	Replicate Intent 5
,	ADT Columns:			Additional Columns 🗸
	Column Data	Column Name	Tag	
	Path Intent	a CPU Usage	b 1 tag	
		Tag Current Column	×	
		1 Tag: path +		C Save and Replicate
			Cancel OK	tted at: 07/30/2024 04:31 PM d Open Output ADT
<b>()</b> S	election Mode: Path-based Replication, AE	T: Path Stability Check, 0 Macro Variables.		Previous Finish

### The ADT after adding **CPU Usage** Intent:

🐻 Pa	th Stability Check	Table Builde	r Last Updated at: 08/09/	2024 12:57 PM 🛛 🍳 Rebuild Ta	ible		Add Data Manually 🗸 📃 🧃
Descr	iption: Type description here						
Items	: 4 Rows 8 Columns					Search	Q T Advanced Filter: Undefined
	9 <sup>9</sup> Path	Path Devices	s Source	s Destination	S Path Status	🔏 Path Hops (Device In	I CPU Usage 🔊
	Database_path	10.1.1.100	US-BOS-Host1	10.101.1.100	Succeeded	EU-LON-R1 - Ethernet0/	CPU Usage Check (Cisco IOS 💿
	Path1	192.168.3	192.168.3.10	192.168.3.201	Succeeded	192.168.3.10 - Ethernet	CPU Usage Check (Cisco IOS 💿
	Reference Path	US-BOS-S	US-BOS-SW1	US-BOS-SW2	Succeeded	US-BOS-SW1 - Loopback	CPU Usage Check (Cisco IOS 💿
	BOS-Host1 to LON-Host1	10.1.1.100	10.1.1.100	10.101.1.100	Succeeded	EU-LON-R1 - Ethernet0/	CPU Usage Check (Cisco IOS 💿
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# Similarly, add another two Intents (**Interface Status Check** and **Uptime check**) via Intent Replication Wizard.

Table Builder	Last Updated at: 08/09/2024 1	2:57 PM 🛛 🍳 Rebuild Table				Add Data Manually	~ =	9
				Search		Q T Advanced Filter: Under	efined	0
Status	🔏 Path Hops (Device In	CPU Usage	0	Uptime Check	Ø	Interface Status	Ø	≡
ded	EU-LON-R1 - Ethernet0/	CPU Usage Check (Cisco IOS) EU	•	Uptime check (Cisco IOS) EU	-LO 🗿	Interface Status Check (Cisc	>	*
ded	192.168.3.10 - Ethernet	CPU Usage Check (Cisco IOS) EU	•	Uptime check (Cisco IOS) US	-во 💿	Interface Status Check (Cisc 🧿	>	
ded	US-BOS-SW1 - Loopback	CPU Usage Check (Cisco IOS) US	•	Uptime check (Cisco IOS) US	-BO 💿	Interface Status Check (Cisc 🧕	>	
ded	EU-LON-R1 - Ethernet0/	CPU Usage Check (Cisco IOS) EU	•	Uptime check (Cisco IOS) EU	-LO 💿	Interface Status Check (Cisc 🧕	>	
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~	ᠵᠬᢦᢉᠯᡐᠺᢦᢉᢑᡘ᠋ᠴᡗ᠁᠋ᠧ᠕ᢣᡄᠬᢦ᠆ᡐᡘ᠇ᠧᢦᠻ	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	ᢣᡗᡔᠬᢛᡒᡨᠣᡇᡐᡐᡐᠧ᠆᠆ᡔ᠇ᡐᠿᠼᡊᢏᡔᡊ᠇ᡐᢧ᠆᠆᠆᠆ᡔᢊ᠊		~~~~

6. Open a replicated intent to check whether all devices in the path are replicated.

Path Stability Check	Table	Builder Last Updated	at: 07/30/2024	4 04:40 PM 🔍 Re	build Table		Ad	dd Data Manually	/
scription: Type desci	ription here								
ns: 1 Row 9 Colum	ins				Search		Q T Adv	vanced Filter: Und	defined
Path Status	🖧 Path Hops (Device In	Uptime Check	♥ =	CPU Usage		0	Interface S	Status	2
ceeded	EU-LON-R1 - Ethernet0/	Uptime check (Cisco IOS) EU	-LO 🗿	CPU Usage Chec	k (Cisco IOS) El	J 🗿	Interface Stat	tus Check (Cisc	0
<b>I</b> Uptime check	(Cisco IOS) EU-LO				🚨 Open 🛛 📄	0 🖒 0	🖌 Edit 🛛 📃		
Result: 07/30/2024	04:48 PM ✓ 🔒 👁 📓		n Log		Ru	in w	ith Live Data		
This intent executio		h 0 errors. You can View Executio	in Log		R	in w	rith Live Data View ~		
This intent executio	on is finished at 07/30/2024 04:48 PM with uptime is 12 weeks, 3 days, 20 hours, 8 r	h 0 errors. You can View Executio		om Seed Device: US-B		in w			
This intent execution	on is finished at 07/30/2024 04:48 PM with uptime is 12 weeks, 3 days, 20 hours, 8 r W1 S EU-LON-SW1	h 0 errors. You can View Execution	1 Frc	om Seed Device: US-B	OS-SW1.	in w			
This intent execution	on is finished at 07/30/2024 04:48 PM with uptime is 12 weeks, 3 days, 20 hours, 8 r W1 S EU-LON-SW2 W2 S EU-LON-SW2	h 0 errors. You can View Execution minutes 8 1 uptime is 12 weeks, 3 days, 20 h	1 Frc		OS-SW1. OS-SW1.	IN W			

7. Run Intent for all the Intent Columns and Rebuild Table.

#### **10.2.3 Create Path Intent for Path Next Hop Stability check**

In this section, you will create an intent to check whether the next hop changes for the path. You will use two seed devices of the different types to support the multi-vendor.

- 1. Create an Intent from the Intent Manager with Name, *Path Next Hop Stability Check*.
- 2. Add Seed devices.

Let us add two types of network devices, **Cisco Firewall** and **Cisco IOS Device**, to learn how to support the multi-vendor. Different CLI commands are used to retrieve the next hop for these devices.

I Path Next Hop Stability Check 🧹	1 Diagnosis Tree	Run with Live Data	Save 🕜 Help 🗄
ype description here			💦 Intent Map: Select 🥆
I Seed Logic	tion Logic 🤣	🛕 Neighb	or pair replication is disable
🚳 + Device 🛛 🙎		Intent Variables: Mar	ager   Tag: + Add
> 🔭 CA-TOR-R1	Type Description here		
> 🦸 US-BOS-FW/act	Type Description here		

- 3. Define the Variable for the **Cisco Firewall** device.
  - a) Define parser using **CLI Diagnosis**. The command will be: *show route 10.1.1.100*.
  - b) The Var Line 1 will be: **\* \*** *smstring:next\_hop\_ip, from*.

1. Define Variable		2. Define Diagnosis
Format1 🗸 +		Test on Devices: 0
① Double-click a variable to parse. Select multiple lines to parse a	table. Critical Variable (0)	🖾 next_hop_ip_FW 🛛 🖌 Type: Single 👔 🔸 New Pattern 🌱
Current Device         07/30/2024 06:59:56 PM           1         JS-B0S-FMWBhow route 10.1.1.180           2         Routing entry for 10.1.1.0 255.255.255.0           4         Knoun via "ospf 1", distance 110, metric 1           5         Last update from 10.1.2.11 on inside, 255.           6         Routing bescriptor Blocks:           7         * 10.1.2.11, from 10.1.50.10, 255:48:55 ag           8         Route metric is 11, traffic share cour           10         10.1.2.10, from 10.1.50.10, 255:48:55 ag           10         Route metric is 11, traffic share cour	48:55 ago o, via inside t is 1 o, via inside	<pre>^ * \$mstring:next_hop_ip, from b 7 * [10.1.2.11], from 10.1.50.10. 255:48:55 ago, via inside &gt; 1 Line + Field Output + Parse Lines - Snext_hop_ip (mstring) = 10.1.2.11</pre>

- 4. Define Variable for the **Cisco IOS Device**.
  - a) Define parser using CLI Diagnosis. The command will be: *show ip route 10.1.1.100*.
  - b) The Var Line 1 will be: **^ \* \$mstring:next\_hop\_ip, from**

CLI Command Diagnosis	(4) ×
CA-TOR-R1 show ip route 10.1.1.100	with Live Data
1. Define Variable	2. Define Diagnosis
Format1 V +	Test on Devices: 0 🗧
Ouble-click a variable to parse. Select multiple lines to parse a table.     Current Device	Critical Variable (0) 🔽 Next_hop_ios 🗸 🖌 Type: Single 🌒 + New Pattern 🗸 🚍
1 [A-TOR-R1#show ip route 10.1.1.100 2 Routing entry for 10.1.1.0/24 3 Known via "ocpf1", distance 110, metric 12, type intra a 4 Last update from 192.168.1.1 on Ethernet0/1, 3w5d ago 7 Routing Descriptor Blocks: 6 * 192.168.1.1, from 10.1.50.10, 3w5d ago, via Ethernet0/1 7 Route metric is 12, traffic share count is 1 8 8 8 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9	Var Line 1 ^ * \$mstring:next_hop_ip, from U I Sectorement from Total and a sector of the sector of
Help   All Intent Variables	Snext_hop_ip (mstring) = 192.168.1.1 Cancel Apply

5. Define Diagnosis for **Cisco Firewall**.

The condition **A** will be: The **Current** *next\_hop\_ip does not equal* **the last** *next\_hop\_ip*. And Define the Diagnosis message.

2. Defir	e Diagnosis 5	
📄 Add No	Le Add Diagnosis Can also click a variable on the left to add automation.	
Name:	check next hop ip for FW Anchor:	
	Type description of the diagnosis	
	Table Rows	
√ If		
А	US-BOS-F Current V	
	next hop ip v Does not equal v next_hop_ip v 🔟	urrent aseline
в	Select Variable V	ast

6. Define Diagnosis for **Cisco IOS Device**.

The **If** condition **A** will be: The **Current** *next\_hop\_ip dose not equal* the last *next\_hop\_ip*. And Define the Diagnosis message.

Add N	D Add Diagnosis	Can also (	click a variable on th	ne left to add automat	tion.
Name:	Next hope Ip check for ios devic	e	Anchor:	~	A
	Type description of the diagnosis.				
🗌 Loop	Table Rows				
□ Loop ∨ If	Table Rows				
√ If	Table Rows			Last 🗸	Current
√ If	🤋 CA-TOR-R1 Current ∽	Does not equal	v next_hop_ip		Current Baseline

#### 10.2.3.1 Replicate the Intent to ADT

In this section, you will use the **Intent Replication Wizard** to replicate the intent, **Path Stability Check**. In the replication settings tabs, you will define the *destination IP* as **Macro Variable** and set it as the path destination.

- 1. In the Seed Intent tab, select the **Path-based Replication** option.
- 2. Define ADT: **Use an Existing ADT** and select **Path Stability Check**. Enter a relevant Intent group name here.
- 3. Replication Settings:

Full Settings for Intent Templat			Replicate Inten	Template a	eighbor Pair Replication
Intent Qualification	Macro Variable	Critical Var	iable Adva	ance Settings	
0 Items + Device Variable	·				
Seed Device     CA-TOR-R1	Macro Variable	Source	Type Define Command Va	Default Value	Look up Data for Device
CI show ip route 1			Command: Variable Pattern: \$dest_ip: Description:	? Tip: You can define a me	
			Type: * Default Value: Prompt for Input:	string 10.101.1.100 d dest_ip	•
	Cancel	ок	Hint:	Set value	Cancel OK e

- a) Click Full Settings for Template to define Macro Variable for the Cisco Device.
- b) Click + **Command Variable** to open the **Add Command Variable** window, and double click the CLI command, i.e., **show ip route 10.1.1.100.**
- c) Define the command variable by replacing the command, *show ip route 10.1.1.100* with *show ip route \$dest\_ip*.
- d) Enter the Default Valu,e, i.e., destination IP, **10.101.1.100**.
- e) Click **OK** to add Macro Variable.
- f) Select **Destination** from the dropdown to define look-up data for the device.

Full Sett	ings for Intent Templat	te				×
<b>()</b> S	erve as Template for:	O Device-based Replication	Path-based R	eplication	🗌 Enab	e Neighbor Pair Replication
l	ntent Qualification	Macro Variable	Critical Variable	Advar	ice Settings	
1 Item	+ Device Variable	+ Command Variable				
	Seed Device	Macro Variable	Source	Туре	Default Value	Look up Data for Device
4	CA-TOR-R1					
		🛍 ip_macro	show ip route \$ip_ma	string	10.1.1.100	Select 🗸 🛣
	US-BOS-FW/act					•      • Path Variable     • Source     • Destination     • SourcePort     • DestinationPort     • Protocol     • PathType     •     Select ADT Column     Clear

g) Similarly, define Macro Variable for Cisco Firewall. The added macro variable will be:

/ 20	rve as Template for: (	) Device-based Replicatio	n 💿 Path-based R	eplication	Enable Neighbor Pai	r Replication	
Int	ent Qualification	Macro Variable	Critical Variable	Advar	nce Settings		
ems	+ Device Variable	+ Command Variable					
	Seed Device	Macro Variable	Source	Туре	Default Value	Look up Data for Device	
	CA-TOR-R1						
		🖽 dest_ip	show ip route \$dest_ip	string	10.101.1.100	Destination	~
	💋 US-BOS-FW/act						
		🖽 dest_ip	show route \$dest_ip	string	10.101.1.100	Destination	~

#### 4. Replicate Intent:

Define the column name and tag for the Intent. The added automation column will be:

Automation Data	Table Manager						<b>0</b> F	Help
🐻 Path Stability Check	Table Buil	ilder	Last Updated at: 08/09/2024 01:	05 PM	🌯 Rebuild Table		Add Data Manually 🗸 🗏	
Description: Type description	n here							
Items: 4 Rows 11 Columns					Search	Q	T Advanced Filter: Undefined	1
🔏 Path Hops (Device In	CPU Usage	0	Uptime Check	Ø	Interface Status	•	Next Hop Stability Check	0
EU-LON-R1 - Ethernet0/	CPU Usage Check (Cisco IOS) EU	0	Uptime check (Cisco IOS) EU-LO	0	Interface Status Check (Cisco IOS	⊙ Pa	ath Next Hop Stability Che 💿	
192.168.3.10 - Ethernet	CPU Usage Check (Cisco IOS) EU	•	Uptime check (Cisco IOS) US-BO	•	Interface Status Check (Cisco IOS	o Pa	ath Next Hop Stability Check 💿	
US-BOS-SW1 - Loopback	CPU Usage Check (Cisco IOS) US	•	Uptime check (Cisco IOS) US-BO	•	Interface Status Check (Cisco IOS	⊙ Pa	ath Next Hop Stability Check 💿	
EU-LON-R1 - Ethernet0/	CPU Usage Check (Cisco IOS) EU	•	Uptime check (Cisco IOS) EU-LO	•	Interface Status Check (Cisco IOS	0 Da	ath Next Hop Stability Che 💿	

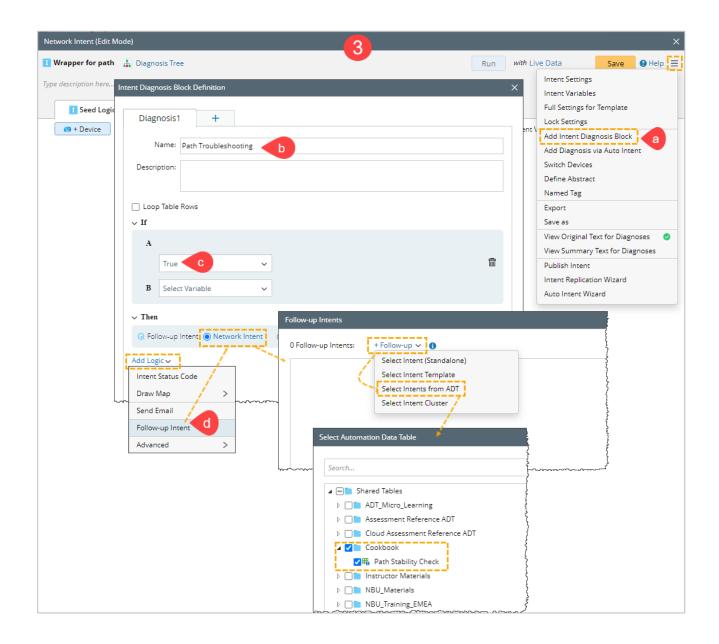
### **10.2.4 Create Wrapper Intent and Enable Auto Intent**

In this section, you will create a Wrapper Intent as a public interface for the path-related intents. For detailed steps on how to create a wrapper intent, please refer to Section 4.2. A key difference here is that you will use the **Intent Diagnosis** method instead of the Device Diagnosis method since the Device Diagnosis method will not replicate all path devices. Also, you need to create a separate ADT for the **Wrapper Intent**.

1. Create an ADT for Wrapper Intent using the **Application Path** Method. Add the **Path** and **Path Devices** built-in fields to the ADT.

urch 🔍 🔉 🔇	<	Table Builder Last L	Underted	at: 08/02/2024 0	B Pobuild 1	Table Add Data Manually 🗸 🔳	
Shared Tables (97)			opuuteu	ac. 00/02/2024 0	- Rebuild I		
ADT_Micro_Learning (3)	Descr	iption: Type description here					
> 📄 Assessment Reference ADT (33)	Items	: 4 Rows 2 Columns		Search	Q	Y Advanced Filter: Undefined	
> Cloud Assessment Reference ADT (41)					~	•	
a 📄 Cookbook (2)	No.	9 <sup>9</sup> Path	9	Path Devices			
Path Stability Check	1	Database path	10	0.1.1.100			
How Wrappers	-			2462.2			
> 💼 Instructor Materials (2)	2	Path1	19	92.168.3			
NBU_Materials (1)	3	Reference Path	U	S-BOS-S			
NBU_Training_APAC (0)	4	BOS-Host1 to LON-Host1	1(	0.1.1.100			
NBU_Training_EMEA (7)							
> 💼 NBU_Training_NALA (1)							
> 📄 Workshops (1)							

- 2. Create Wrapper Intent from Intent Manager with name, Wrapper for Path.
- 3. Add Follow-up intent by selecting Select Intents from ADT option.



4. In the Follow-up Intents window:

Follow-up Intents		×
1 Follow-up Intent: + Follow-up 🗸 🚯		
∼ ∰ Path Stability Check (Automation Data Table)	Ē	-
Description: When		
Find Critical Automation Assets (ADT Rows) by Intent Variables (Device or Device Variables): Find critical assets by device, or properties of critical asset.		
A     Application Name     V     Is not none     Image: Comparison of the second s		
B Select ~		
Boolean Expression: A		
Select Intents of Found Critical Automation Assets to Execute:		
O All Intents		
O Selected Intents: Select ~		
Intents with Tags: Match all      path		-
Prune other follow-up intents Car C	Save	

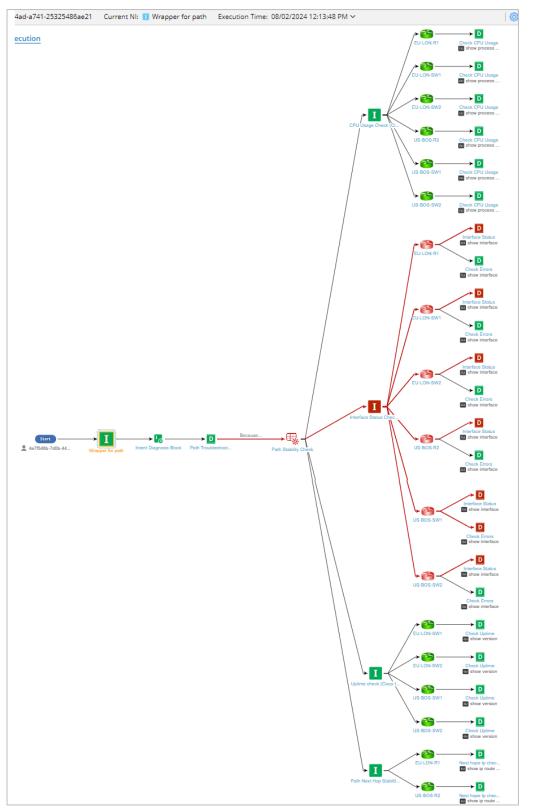
- a) Define condition **A**: *Application Name is not none*.
- b) Select the **Intents with Tags** option to Match all **path** tags.
- c) Click **Save** to save the wrapper intent.
- 5. Replicate this Wrapper Intent to the ADT using **Path-based Replication**.

Automation Data Table Mana	ger					•	🗿 Help	
Search Q C «	-	appers Tal	🌯 Rebuild 1	= d				
<ul> <li>ADT_Micro_Learning (3)</li> <li>Assessment Reference ADT (33)</li> <li>Cloud Assessment Reference ADT (41)</li> </ul>		4 Rows 3 Columns		Search	۹		ed C	
🔺 🖿 Cookbook (2)	No.	🞐 Path	Path Devices			Wrapper Intent		
🖁 Path Stability Check	1	Database path	10.1.1.100		Wrapper for	r path Path_Troubles 💿		
H Wrappers								
Instructor Materials (2)	2	Path1	192.168.3		Wrapper for	r path Path1 💿		
NBU_Materials (1)	3	Reference Path	US-BOS-S		Wrapper for	r path Reference Path 🛛 💿		
NBU_Training_APAC (0)	4	BOS-Host1 to LON-Host1	10.1.1.100		Wrapper for	r path BOS-Host1 to 💿		
NBU_Training_EMEA (7)								
NBU_Training_NALA (1)								
👌 🗅 Workshops (1)								
Land the second				~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~	᠕᠂᠆᠆᠆᠆᠆᠆᠆᠆᠂᠆᠂		

6. **Run** the Wrapper Intent and check the diagnosis tree for all the replicated devices.

Network Intent (View Mode) - All Network Intents/Cookbook/New Wrapper			×
I Wrapper for path	💦 Open	<u>∎</u> o ⊀	0 🖌 Edit 🚍
Result: 08/02/2024 12:26 PM 🗸 👔 💿 🔛	R	un   ~	with Live Data
This intent execution is finished at 08/02/2024 12:26 PM with 0 errors. You can View Execution Log			
S on EU-LON-R1, Interface Ethernet0/3 is down, Current L1 status is administratively down and L2 status is down			View 🗸
Diagnosis Message (08/02/2024 12:26 PM)			
Intent: 1 Wrapper for path			
With 4 Follow-up Intents S 66 Alerts S 326 Successes S 196 Messages	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	৵ <b>৵৽৽৽৵৵৾৾৽</b> ৾৻৾৻৵৴৽৾৾৽৾৻ৢ৵৵৽৾৽৽৽৾
View by Intent View by Device			
<ul> <li>✓ ■ Wrapper for path</li> </ul>			
>  Path Stability Check / CPU Usage Check (Cisco IOS) EU-LO 12 Successes 6 Messages			
>  Path Stability Check / Interface Status Check 66 Alerts 302 Successes 184 Messages			
>  Path Stability Check / Uptime check (Cisco IOS) EU-LON-SW1 8 Successes 4 Messages			
>  Path Stability Check / Path Next Hop Stability Check EU-LON 4 Successes 2 Messages			

7. Click the Diagnosis Tree 📠 icon to view the results in a tree layout.



8. Enable Auto Intent for Wrapper Intent.

NOTE: Before Enabling **Auto Intent** for the Wrapper Intent column, you need to define **Auto Intent Profile** in the **IBA Center**.

a) Enable Auto Intent via ADT.

🎝 Intent Based Automation Ce	nter							
Installed Intent Templates Published Int	ents Au	ito Intent	Auto Intent Profile	NetBrain Download				
+Add Folder	G Autor	nation Data Tab	le: Path Stability C	heck Description:				
Search		Intent Name: F	Path Stability Check ·	- Wrapper Intnet				
Construction Content via ADT     Construction     Construction Content via ADT     Construction     Construction	alth A B alt	Path Devices Select		✓ Matches		Hostname		
BGP_Route_check     Device Information		: intents/maps/pa : in Auto Intent	aths to be listed in A	uto Intent and set the dis				
I IP Route Navigator				1	Display Name			
I MAC Address Navigator			😏 Path		\$Path 🖉 🛛 🏢			
I Map One-IP Table Entry			I Uptime		\$Path_Intent_2 🧪			
Wireless_Troubleshooting			I CPU Us	age	\$Path_Intent 🖉			
I NB302_Client_Summary [Cisco WLC]			Interfac	e Status	\$Path_Intent_1 🔁			
Suresh_WLC_Client_Summary			🚺 Next Ho	op Stability Check	\$Path_Intent_4 者			-
	Autor	nation Data Table	e Preview:					
	No	SApplicatio	on Name	9 <sup>9</sup> Path	Path Device:	5	s Source	S Destination
	1	Cookbook		Path_Troubleshooting	10.1.1.100		US-BOS-Host1	10.101.1.100

#### b) Define Auto Intent Profile.

Installed Intent Templates	Publishe	d Intents	Auto Intent	Auto Intent Profile	b Brain Download	
+Add Profile	G	Name	e: Database_pa	th_troubleshooting		
earch	Q	Descriptior	a lonut			
🛯 Shared Profiles		Description	1. <i>mpa</i>			
NBU_EMEA_Training Network Essential		Included N	ITs/ADTs: +Add			
🛯 Wireless_Troubleshoot	ing	NIT/ADT	and the second se			Location
🔁 My Profiles		📰 Path	× Stability Check -	Wrapper Intnet		/Enable Auto Intent via AD
path_troublshooting			-			

c) **Enable Auto Intent** from the ADT Wrapper Intent column.

	<pre>č</pre>		Search		Q T Advanced Filter: Under	fined C
		Next Hop	Stability Check	0	Wrapper Intnet	© ≡
Enable Auto Intent - Wrapper	' Intent		×	٥	Run Intents Once Run Intents via Timer	PA PA
Auto Intent Name:	Path Stability Check - Wrap	per Intnet			Open Seed Intent Rebuild Intent-related Column G	roup
Select Device Column:	Path Devices		~		Remove Empty Wrapper Intent Enable Auto Intent	
Add to Auto Intent Profile:	My Profiles/Database_path	_troubles	Select	~~~~~	Export Diagnosis Result to CSV View Summary CSV Report	
1 Enabled Auto Intent 💙		Cancel	ОК			

9. Run the Wrapper Intent and Create a Dashboard.

Path TS Wrapper				Last Refreshed at 02	/08/2024, 19:30:46 🗘	# ∅ ≡
Summary	02/08/2024, 19	:30:51 View Report I	Device Information		02/08/2024, 19:30:51	View Report
6 17 99	Intents Times Executed Intent-level Alerts		Cisco 105 Switch     Cisco Rout	ter • (null)	6 Devic	ces
Intent Result History Time Range: All Y Result: All Y					02/08/2024, 19:30:51	View Report
	🔶 Sum of Intent A	lert Status Code Count 🛛 🔶 S	Sum of Intent Success Status Code Count			
200 100 0 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,08,2024, 02,000,000,000,000,000,000,000,000,000	22/08/2024, 02/08/2024, 02/08/2024, 02/08/2024, 18:27:28 18:27:28	02/08/2024, 02/08/ 18:37:53 18:37	2024, 02/08/2024, 02/08/2024, 02/08/2024, 18.37/54 18.37/53 18.37/53	02/08/2024, 02/08/2024, 02/08/2024 19:09:22 19:09:24 19:09:24		02/08/2024, 19:09:26
		Top Five Intent	Alerts			
Intent Name Map	Execution Time	Intent Alert Status Code Cou			tent Alert Detection	
Interface Status Check (Cisco IOS) EU-LO View Map	02/08/2024, 18:27:27	33	151	on EU-LON-R1, Interface Ethernet0/3 is do 1		
Interface Status Check (Cisco IOS) EU-LO View Map Interface Status Check (Cisco IOS) EU-LO View Map	02/08/2024, 18:37:55 02/08/2024, 19:09:24	33	151	on EU-LON-R1, Interface Ethernet0/3 is do 1 on EU-LON-R1, Interface Ethernet0/3 is do 1		
Path Next Hop Stability Check EU-LON-R1 View Map	02/08/2024, 19:09:24	0	2	EU-LON-R1 next hop IP 10.101.2.11 has N 0		

#### Intent Dashboard:

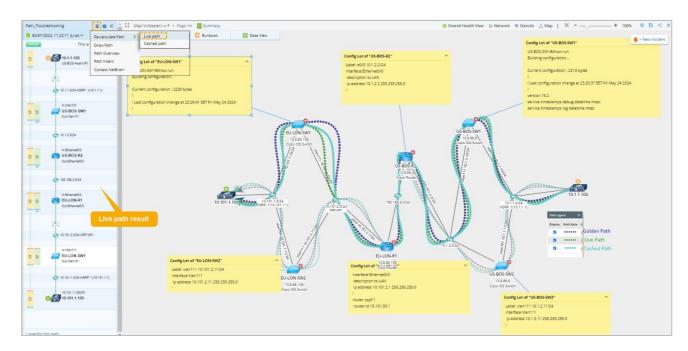
### Summary Dashboard:

Summary										
Number of Intents	Numbe	er of Devices		Numb	er of Alerts		N	umber of Successes		
4	0			19	9	978				
Path										
Number of Intents	Numbe	er of Devices		Numb	er of Alerts		N	Number of Successes		
4	0			19	8		ç	78		
		Device Results								
Dashboard and Intent Group	Intent Results	Total Device Results	EU-LON-R1	US-BOS-R2	US-BOS-SW1	EU-LON-SW1	US-BOS-SW2	EU-LON-SW2		
Path TS Wrapper	489 99	489 99	66 24	66 24	87 15	90 12	90 12	90 12		
Total Alert Count $\downarrow$			24	24	15	12	12	12		

## 10.3Troubleshoot the path issue

Calculate the live path and compare it with the golden or cached path to view the path differences.

1. Open your path from the path browser, draw the live path, and then draw the Golden or Cached path for comparison.



2. Run the Wrapper Intent and view the intent results in the dashboard.

Path Troubleshooting												
Summary												
Number of Intents	fDevices		1	Number of Successes								
4	0					462						
						▲ 264				2,282		
Path												
Number of Intents	1	Number of	f Devices			Numbe	er of Alerts		1	lumber of Successes		
4		0		462						2,282		
						▲ 264				1,304		
			Device Results									
Dashboard and Intent Group	Intent Results		Total Device Results	EU-LON-R1	US-BOS-R2		US-BOS-SW1	EU-LON-SW1	US-BOS-SW2	EU-LON-SW2		
Path TS Wrapper	1,141	231	1,141 231	154 56	154	56	203 35	210 28	210 28	210 28		
Total Alert Count $\downarrow$				56		56	35	28	28	28		

# **11 Intent-based Change Verification and Assessment**

NetBrain **Change Management** module is a comprehensive solution for controlling a network change process. Within a single Runbook, network engineers can define a change, deploy the configuration updates, and verify the impact of the changes in a fully auditable way and adherent to existing approval processes.

With intent-based automation, you can upgrade the change process with more accurate verifications of the change. One recommended change flow can be:

	Net3rain Next-G	en	SF	× 📀	<b>&gt;</b>				Ø	ncident Se	arch Incident	q 📑	<b>4</b> 6	Lab Dor	main 🔿
	្នឹង Map63 (Master) 🗸	* > Pi	age 1 🗸 📒 Summary				📀 Overall Health View	v 🚦 Network	📑 Stencils	💦 Map	🗵 -	-0	- + 10	0% 🚳	🖹 < X
$( \pm )$			Runbook	D Data View										( + New	v Incident
	Add a network subnet	(for Coo	kbook) 🗸	📋 Document	Auto Test Intent (Be	fore Change)		[Ţ] No	te (0) 🗙					- · · · ·	v meldene
Recents	Select Action			≡	Please input descriptio	n for this action.									
	2		Define Change		Items:4 + Add Inte	ent									
					Intent Name	Target Dev	Status Code	CSV Repor	Actions						
Files			Benchmark Before		Ping Check	1						•			
Site					Assess OSPF neighbo	or st 1				1		6			
Site				_	Check CPU	1						US-BOS-R2			
<u>9</u> 9	3		Auto Test Intent (Before		Cookbook Server Far	rm F 1	la la					10.8.1.240 Cisco Router			
Path															
Cashboard	4	0	Execute												
		-													
Intents	5		↓ What is Changed Intent		4				•						
Ę			what is changed intent		Items:0 Filter Intent	by Tags: type to	o select tags.	Filter			2				
Chatbot					Intent Name Targ	et Dev Status (	Code CSV R	epor Diagnosi	is Tr		11.20AL	172.16.8.0/22	Et2/3		
Data >	6		Auto Test Intent (After C							1	2.16.11.204/22		- /)	2.16.11.206/22	
Data										EBIT	, ,			-9/22	<u> </u>
			Benchmark After							<u>~~</u>					US-BOS-SW2
										BOS-SW3					10.8.1.242 Cisco IOS Switch
			¥							0.8.1.243 o IOS Switch					
			Compare												
Мар															
Desktop			0					Ru	n ©						
	🖹 Incident: 105XRX							Ru	6						

- 1. Map the devices you want to change and those that can be affected by the change, for example, the L3/L2/routing neighbors.
- 2. Define the change.
- Prepare the intents you want to run before and after the change. You can add as many as intents to verify that the change does not violate the designs and that important operation status is normal. At least you may want to run the batch ping for the key applications to ensure that they are still accessible. Run these intents before the change.
- 4. Execute the change.
- 5. After the change, run an intent to highlight what has changed and verify that your change is indeed successfully pushed to the devices.
- 6. Repeat the same intents as step 3 to verify that the change does not violate the designs and that important operation status is normal.

You may still use the "old" verification methods: benchmark the data before and after the change and compare these data to check the difference. The comparison is based on the text only, while the intents provide a more accurate way.

In this chapter, you will learn how to create the intent for the change flow. We will focus on two types of intents: the intent to show what has changed and the other basic auto-test. Certainly, all intents you have created up to now can be added to the CM workflow. For example, the general device health check you created in Chapter 3 can be useful to verify that the device is healthy before and after the change.

As an example, we will modify the OSPF configurations of a router to add a new subnet.

# **11.1Build Intents for Change Assessment**

# 11.1.1 What has Changed

To find out what has changed between two times (for example, the time when your network is good and when your network is troubled) is not just useful for the change process but also essential for the troubleshooting process and network assessment. In Chapters 5 and 6, you have learned how to parse a subset of configurations, such as AAA, ACL, etc. and compare the configuration to the golden template or the industry standard and best practices. You may run these intents before and after the change.

For a change process, you can create an intent to find out the change of the whole configurations before and after the change.

Create an intent, **Configuration Change**, select a seed device (for CM, you can select a device you are going to change configurations), and add a **Configuration Diagnosis**. A Cisco IOS device configuration will be like:

US-BOS-R2#show run

Building configuration...

### Current configuration : 12524 bytes

```
      !

      ! Last configuration change at 10:11:43 EST Wed Jul 31 2024 by nb

      !

      version 15.4

      service timestamps debug datetime msec
```

You may notice that the bold lines can often change, and so you want to ignore them while comparing the confiugrations. So, you define a variable **\$configurations**, by the function

**LinesByVariable,** starting from the line, *version \$float:version*, to the end. You can click the link **Parse Lines** to define the function **LinesByVariable.** Check the variable value to confirm its correctness.

fine Variable	2. Define Diagnosis
Format1  +  buble-click a variable to parse. Select multiple lines to parse a table.  Critical Variable (0)	Test on Devices: 0
<pre>vurrent Device v 07/31/2024 02:28:29 PM Search Q Var Line 1 7 8 version 15.4 9 service timestamps debug datetime msec 10 service timestamps log datetime msec 11 service timestamps log datetime msec 13 hostname US-BOS-R2 14 I 15 boot-start-marken 16 boot-start-marken 16 boot-start-marken 17 I 18 I 19 lenable secret ******* 14 I 19 lenable secret ******** 14 I 19 lenable secret ******** 14 I 19 lenable secret ******** 14 I 15 lenable secret ******** 15 lenable secret ************************************</pre>	Var Line 1       version \$float:version         8       version 15.4         Var Line 2       LinesByVariable[\$confiugrations]:\$version-         8       version 15.4         + Field         Output       + Parse Lines         \$version (float) = 15.4         \$conflugrations (string) = version 15.4 service timestamps debug datetime msec service timest.

In the diagnosis, you compare the current and baseline values of **\$configurations** using the operator **Pattern Match**. If they do not match, create the status code to include the added and missing lines:

Configurations changed. Added lines:

\$Rule1.Unmatched\_lines

Missing Lines:

#### \$Rule1.Unused\_pattern\_lines

Note:

- Select the **Pop up** option from the menu at the right of the message to pop up the status code editing window to enter the note with the multiple lines.
- You need to run the intent twice to have the last and current data.

1. Define Variable		2. Define Diagnosis	
Summary Text Original Text Search.		Add Note D Add Diagnosis	Can also click a variable on the left to add auto
<pre>8 version 15.4 9 service timestamps debug datetime msec 10 service timestamps log datetime msec 11 service password-encryption 12   13 hostname US-BOS-R2 14   15 boot-start-marken 16 boot-end-marken 17   18   19 lenable secret ********</pre>	What has Changed *	Loop Table Rows V If US-BOS-R2 Current Conflugrations VMP(Rule1) B Select Variable V	Last ~ v conflugrations v
20 ! 21 no aaa new-model sis Message of Intent Diagnosis	×	✓ Then Diagnosis Message:	□ Save to Incident =
osis Message: figurations changed. Added lines: Rule1.Unmatched_lines I ing Lines: Rule1.Unused_pattern_lines	+ 🖸 Variable 🛛 + 🗊 Expression	Configurations did not change     Set Status Code for Device:     Success      Configurations did not change     S Set Status Code for Intent:	
s Code for Device:	+ 🗷 Variable 🛛 + 🖪 Expression	Add Logic ✓	
figurations changed. Added lines: Jule1.Unmatched_lines sing Lines: Jule1.Unused_patters_lines	* *	Else  Jagnosis Message:  Configurations changed, Added lines:  Configurations changed. Added lines:	Save to Incident = \$Rule1.Unmatche Delete Delete
	Cancel OK	+ Add Elself	

#### Run the intent and check the results:

ĺ

Diagno	osis Details Compare	
ummar		Diagnosis Logic (What has Changed)
1	US-BOS-R2#show run	Anchor: version 15.4 service timestamps debug datetime msec service
2 3	Building configuration	if
4 5 6 7	Current configuration : 12566 bytes ! ! Last configuration change at 21:49:12 EST Wed Jul 3:	A: SUS-BO Current SUS-BOS-R2 Last \$confiugrations MP:Rule1 \$confiugrations Fals
7 8 9 10 11 12	version 15.4 service timestamps debug datetime msec service timestamps log datetime msec service password-encryption	Configurations changed. Added lines: Boolean Expressio Username ******** Missing Lines:
13 14 15 16	hostname US-BOS-R2 ! boot-sart-marker boot-end-marker	> Else
17 18 19 20	lenable secret *******	Diagnosis Message: Configurations changed. Adlyed lines:     Status Code for Device: S Configurations changed. Added li
21 22 23 24	no aaa new-model clock timezone EST -5 0 mmi polling-interval 60 no mmi auto-configure	
25 26 27 28 29	no mmi pvc mmi snmp-timeout 180 ! !	

### 11.1.2 Auto Test

While making the network change, you want to ensure no accidental impart. For example, the server farm is still accessible, and the traceroute to the critical server does not change, etc. You can develop a set of automations for auto tests.

Let us use the ping as an example. To ensure that the critical server still functions, you may ping the server IP addresses from a set of devices that should have routes to the server. Ping for many servers from hundreds and even thousands of devices manually is not feasible. However, you can have the intents to do this for you automatically.

First, create an ADT including all pairs of the device and server IP. You can create a CSV file and import the CVS file into the ADT table (make sure that you set the device name column as the data type **Device**). The following is a small set of sample data:

age	er									6	Help
< I	🖥 Bato	h Ping Device and Target IP	Table Builder	Last Update	d at: N/A	🌯 Rebuild Table				Add Data Manually $\sim$	≡ 🖬
(	Descrip	tion: Type description here									
ľ	tems:	4 Rows 3 Columns						Search	Q	Advanced Filter: Undefine	ed 🖸
1	NO.	s Device		s Ping	IP		s Ping	g Target			=
1		US-BOS-R2		10.10.1	0.1		East Se	erver Farm			4
2		US-BOS-R1		10.10.1	0.1		East Se	erver Farm			
3		US-SFO-R1		192.16	8.1.1		West S	erver Farm			$\mathbf{v}$
4		US-SFO-R1		192.16	8.1.3		Mail Se	erver		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	

Now, you can build a new Column Group, **Ping Check**:

- 1. Add a new group, **Ping Check**.
- 2. Select the **Ping Check** intent created in Chapter 2.
- 3. Select the Device Column.
- 4. Set the macro variable, *dest\_ip*, as the **Ping IP** column.
- 5. Add the intent output to the ADT.
- 6. Save and build.

c1   C2     Device   Ping IP	S C c3 S C Ping Target	-	cs S C	t1 O		Reset Al
Base Ping Check +	Column Group (Ping Check)	:				Select Column 🗸
Description: Select Method to Build Group Table:	e4 Ping Check Intent Replicated Intent	C5 Intent Status Coc Intent Status Coc				
Intent Template Intent Template: Ping Check	~					
Built-in Fields:	(Drag and drop column hee	der from the upper area or av	ailable data field from the left	here.)		
Intent Output:	Define Logic to Populate Nev	Columns for Each Row				
Intent Message S	Select Device Column to Re		~	-3		
🕄 Intent Status Code 🖻 🗾 5	Set Macro Variables of Seed	Intent Template:				
Device Status Code S	<ul> <li>Seed Device</li> </ul>	Macro Variable	Туре	Set Variable from AD	T Field	
Intent Devices 🛛 🍕	▲ 🚳 ISP-P02					
Intent Map 😤		dest_ip	string	Ping IP	✓ □ Use r 4	text as a string
Intent CLI Commands 5	Replication Settings Auto	o selantent Map for replicate	d intents: Settings			
C Last Execution Time			0			

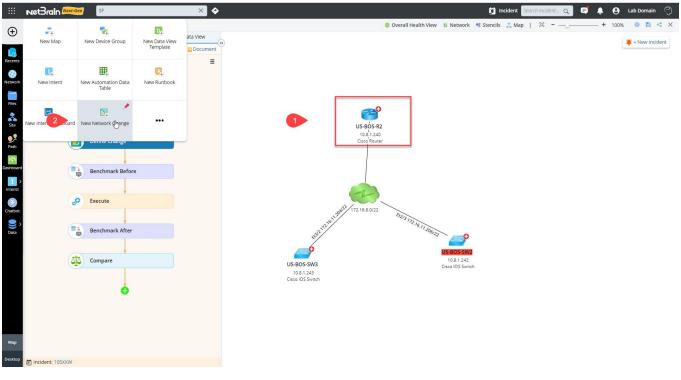
# Run the replicated intents once and check the results. You can also create a dashboard.

	× 🗞				🔯 Inc	ident Search Incident Q	• • •	Lab Domain	\$
şer								😗 Hel	lp
🐻 Ba	ich Ping	Table Builder	Last Updated at: 08/01/2024 02:49 PM	🌯 Rebuild Table			Add Data	a Manually $\sim~\equiv$	<b>B</b>
Descri	ption: Type description here								
Items:	4 Rows 6 Columns					Search	X Advanced	Filter: Undefined	0
No.	Device	s Ping IP	S Ping Target	Ping Check	≡	S Intent Status Code	() Last Execution	on Time	≡
1	US-BOS-R2	10.8.1.1	East Server Farm	Ping Check US-BOS-R2 1	۲	Ping succeeded	08/01/2024 02:	49:09 PM	^
2	US-BOS-R1	10.8.1.1	East Server Farm	Ping Check US-BOS-R1 1	۲	Ping succeeded	08/01/2024 02:4	49:09 PM	
3	US-SFO-R1	192.168.1.1	West Server Farm	Ping Check US-SFO-R1 1	۲	Ping failed with the success r	08/01/2024 02:4	49:09 PM	
4	US-SFO-R1	192.168.1.3	Mail Server	Ping Check US-SFO-R1	٥	Ping failed with the success r	08/01/2024 02:4	49:09 PM 🗸	

## **11.2Intent-Driven Change Management and Assessment**

In this section, we will walk you through the intent-driven change management and assessment flow with a simple change: we will add a new subnet at one router, **US-BOS-R2**, and add this subnet into the OSPF configuration.

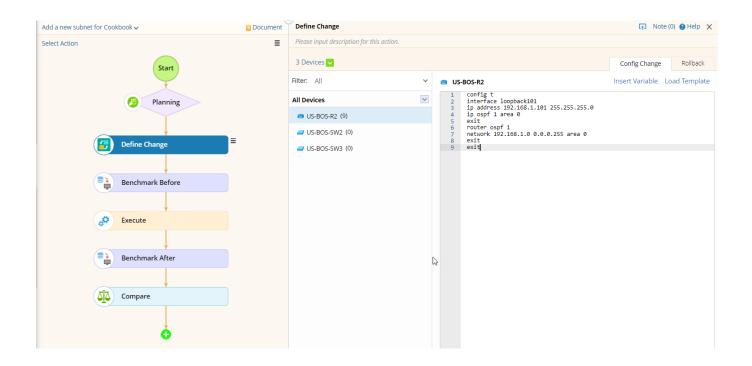
Map the devices you are going to make changes to and add the possibly affected devices, such as OSPF neighbors, in our example. Then, add a **New Network Change** task.



### 11.2.1 Define Change

Click the **Define Change** node, select the device, **US-BOS-R2**, and enter the **config change**, which adds a new loopback interface and the subnet into the **OSPF 1**.

```
config t
interface loopback101
ip address 192.168.1.101 255.255.255.0
ip ospf 1 area 0
exit
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
exit
exit
```



You may also enter the configurations to roll back your change in case the change causes unexpected results.

After you review the configurations to be changed, you can click the **Planning** node to request approval. The change can only be executed after the change is approved. You can assign yourself as the approver for the exercise.

### **11.2.2 Create Intent to Analyze Route Changes**

Since we are going to make changes to the OSPF configuration, it will affect the routing. So, we should add an intent to analyze the route changes.

Create a new intent, **Compare Dynamic Routes**, and select the device to which you will change configurations. Enter the CLI command *show ip route* and retrieve the data. At first glance, it is hard to parse the result. However, if you study the data closely, the routes have two types of formats: one for the dynamic routes and one for the directly connected routes. So, you can create two Paragraph Parsers to parse the dynamic and connected routes.

For the dynamic routes, copy a line to the **ID Line A** and replace the values with the variables (be cautious to keep the number of spaces intact):

\$mstring:protocol \$subnet [\$distance] via \$next\_hop, \$\_dummy

efine Variable		2. Define Diagnosis			
Format1 V +					Test on Devices
Double-click a variable to parse. Select multiple lines to parse a table.	Critical Variable (0)	) 📲 dynamic_routes 🕓	🖌 者 Type: Paragraph 💡	+ New Pattern 🗸	
Current Device	. Q 🔺 🔻				
14 #0*E2 0.0.0.0/0 [110/1] via 10.8.1.51, 7w0d, Ethernet0/1	P1-ID Line A	ID Line A \$mstring:	protocol \$subnet [\$dis	ance] via \$next_hop, \$_	dummy
15 1.0.0.0/32 is subnetted, 1 subnets	P2-ID Line A	14 000	2 0.0.0.0/0 [110/1] via 10.8.1	51 Zu Orl Ethernet0/1	> 49 Lines
10 IA 1.1.1.246 [110/32] via 10.8.1.49, 5w5d, Ethernet0/1 8.0.0.0/32 is subnetted, 1 subnets		14 012		.51, /wod, Etherneto/1	7 49 Lines
18 AD 8.8.8.8 [110/11] via 10.8.1.51, 7w8d, Ethernet0/1	P3-ID Line A				
19 10.0.0.0/8 is variably subnetted, 43 subnets, 7 masks	P4-ID Line A	+ Field 🗸			
20 40 10.2.2.2/32 [110/21] via 10.8.1.49, 5w5d, Ethernet0/1	P5-ID Line A				
21 40 10.3.3.3/32 [110/22] via 10.8.1.49, 5w5d, Ethernet0/1	-6-ID Line A				
22         0         IA         10.8.0.0/16         [110/30]         via 10.8.1.51, 7w0d, Ethernet0/1           23         0         10.8.1.0/28         [110/21]         via 10.8.1.49, 5w5d, Ethernet0/1					
23 40 10.8.1.0/28 [110/21] Via 10.8.1.49, 5w5d, Ethernet0/1	P7-ID Line A				
25 40 10.8.1.32/29 [110/20] via 10.8.1.49, 5w5d, Ethernet0/1	P8-ID Line A				
26 C 10.8.1.48/29 is directly connected, Ethernet0/1	PO ID II - A				
27 L 10.8.1.53/32 is directly connected, Ethernet0/1	P9-ID Line A			<b>^</b>	
28 C 10.8.1.64/30 is directly connected, Tunnel0	P10-ID Line A	Output + Parse Line	25	_	
29 L 10.8.1.65/32 is directly connected, Tunnel0	P11-ID Line A				
30 C 10.8.1.240/32 is directly connected, Loopback2	P12-ID Line A	\$protocol 🗸	\$subnet 📭 🛛 🗸 🗸	\$distance V	\$next_hop ~
B         10.8.2.0/30 [20/0] via 104.1.1.1, 1w0d           B         10.8.2.4/30 [20/0] via 104.1.1.1, 5w5d		O*E2	0.0.0/0	110/1	10.8.1.51
33 4B 10.8.2.8/30 [20/0] via 104.1.1.1, 5w5d	P13-ID Line A	0 12	0.0.0.0/0	110/1	10.0.1.51
34 ⊿B 10.8.2.12/30 [20/0] via 104.1.1.1, 5w5d	P14-ID Line A	O IA	1.1.1.246	110/32	10.8.1.49
35 4B 10.8.2.16/29 [20/0] via 104.1.1.1, 5w5d	P15-ID Line A				
36 B 10.8.2.52/30 [20/0] via 104.1.1.1, 5w5d		0	8.8.8.8	110/11	10.8.1.51
37 AB 10.8.2.128/25 [20/0] via 104.1.1.1, 5w5d	P16-ID Line A	0	10.2.2.2/32	110/21	10.8.1.49
38         10.8.2.250/32         [20/0]         via 104.1.1.1, 5w5d           39         4B         10.8.3.4/30         [20/0]         via 104.1.1.1, 5w5d	P17-ID Line A		10.2.2.2/02	110/21	10.0.1.49
40 48 10.8.3.4/30 [20/0] Via 104.1.1.1, 5W5d	P18-ID Line A	0	10.3.3.3/32	110/22	10.8.1.49
41 40 E2 10.8.5.0/24 [110/20] via 10.8.1.51, 2d02h, Ethernet0/1	P19-ID Line A				
40 ·····	Pla-ID Line A	O IA	10.8.0.0/16	110/30	10 8 1 51
42	P20-ID Line A				

Review the table to confirm its correctness and completeness.

You can parse the connected routes similarly. We will leave it for your homework.

You can create a simple diagnosis to compare the current and last route tables without looping each route entry. Create a message when the table changes. The system will automatically add the summary message to show how many entries are changed, added, and removed. You can configure the number of entries to be included by modifying the **Table Compare Settings**.

> 2. Defin	ne Diagnosis				
📄 Add Ne	ote D Add Diagnosis		Can also click a variable on t	the left to add automation	n.
Name:	Compare dynamic routes		Anchor:	×	
	Type description of the diagnosis				
□ Loop ~ If	Table Rows				
А	e US-BOS-R2 Current	,		Last 🗸	Table Compare Settings X
в	dynamic_routes ~ Select Variable ~	Does not equal	✓ dynamic_routes	~ 🗎	Output table-compare summary message Compare Table dynamic_routes of SUS-BOS-R2
~ Then					Between Current and Last
<b>-</b> 5	enosis Message:       V     Dynamic route changes       Set Status Code for Device:       Error     V   Dynamic route changes	es			Sample Summary Message e.g. 'table' of \$device has changed (Current vs Baseline): 30 changed, 20 added, 10 removed.
5	Set Status Code for Intent:				<ul> <li>✓ Include Top</li> <li>✓ Include Top</li> <li>20</li> <li>Added Entries</li> </ul>
Add Logi + Add I				•	Include Top 20 Removed Entries
				Cancel Apply	Cancel OK

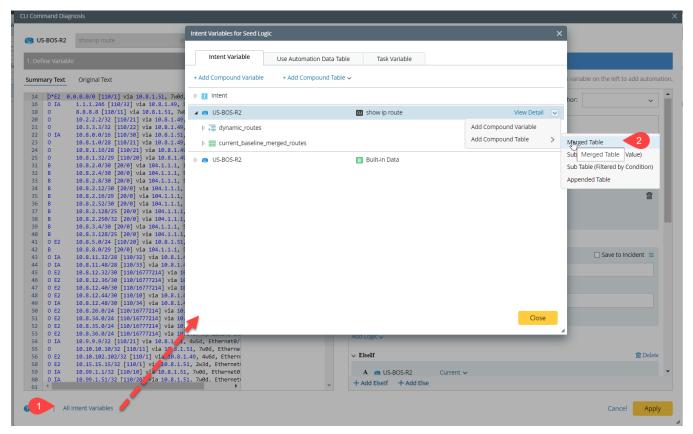
If you want to get more details, you can create a diagnosis to find out all the changed, added and missing entries. For this purpose, you can add a **Compound Table** to merge the current and last tables. The system supports different rules for merging tables, and the most common one is the **full join**:

		Returns a	all rows when th	here i	s a match in le	ft (table1)	or rig	ht (table2) t	able rows.		
		Left Ta	able (Table1)		<b>Right Table</b>	(Table2)			Merge	d Table	
		Columnal	Column2					L. Column1	L. Column2	R. Column1	R. Column2
Full Join	Table1 Table2	Column1	a	C	Column1 🗊	Column2		1	a		
		1		-	ь	х		2	b	b	×
		2	b	-	d	z	11	3	c		
		3	c							d	z

So, when you merge the current route table with the past route table using the subnet as the paired key, the entry of a missing route will only have the values in the columns corresponding to the past route table (the values of the columns corresponding to the current route table are empty) in the merged table while an added route will only have the values in the columns corresponding to the current route table.

To create a merged table,

- 1. Click the **All Intent Variables** link at the bottom of the Diagnosis window.
- 2. At the dropdown menu, select **Add Compound Table > Merged Table**.



- 3. In the Edit Merged Table window,
  - a) Enter an easy-to-understand name for the merged table, e.g., *current\_last\_merged\_routes*, implying that this table merges the current and last routes.
  - b) At the right side of **Table 1**, select the table from the pulldown menu, *dynamic\_routes*, and then select the **Current** table (which is the default option) above.
  - c) At the right side of **Table 2**, select the table from the pulldown menu, *dynamic\_routes*, and then select the **Last** table (which is the default option) above.
  - d) Select the **subnet** as the key for both tables.
  - e) In the **Advanced Setting**, you can select columns in the merged table and select the merging rule. Here, you can use the default value.
  - f) Check the merged table for its accuracy.

Table Name:	current_last_merged_rou	ites a				(Help =	
Input Tables:				<ul> <li>←</li> </ul>		Select 、	·
	US-BOS-R2		Current	~	OS-BOS-R2	Last 🗸	
Table 1:	dynamic_routes		~	Table 2	dynamic_routes	+ Add	
Paired Keys:						isis =	=
Key 1:	subnet (US-BOS-R2.dynai	mic routes)	d ~	Key 1	subnet (US-BOS-R2.dynamic_routes)	✓ 💼 Iosis E	
+ Add P	aired Keys		<b></b> c	alculate	e Au-anced Settings		<b>S-R2</b> .240
Table 1 (dyn	amic_routes) Table	2 (dynamic_routes) O	utput (current_last_me	erged_routes)	elect columns in merged table:		
				US-BOS-R2			
				ute , 📃 dynamic_routes	, Table 1:	Table 2:	
protocol (string)	✓ subnet (string)	✓ distance (string)	✓ next_hop (string)	<ul> <li>protocol (string)</li> </ul>	Columns	Columns	
		110/1	10.8.1.51	O*E2	protocol (US-BOS-R2.dynamic_routes)	protocol (US-BOS-R2.dynamic_	routes)
	0.0.0/0	440.000	40.04.40				
AI	1.1.1.246	110/32	10.8.1.49	O IA	✓ subnet (US-BOS-R2.dynamic_routes)	subnet (US-BOS-R2.dynamic_n	outes)
) IA		110/32 110/11 110/21	10.8.1.49 10.8.1.51 10.8.1.49	0 IA 0 0	<ul> <li>subnet (US-BOS-R2.dynamic_routes)</li> <li>distance (US-BOS-R2.dynamic_routes)</li> </ul>	subnet (US-BOS-R2.dynamic_n     distance (US-BOS-R2.dynamic_	
) IA ) )	1.1.1.246 8.8.8.8	110/11	10.8.1.51	0	distance (US-BOS-R2.dynamic_routes)	distance (US-BOS-R2.dynamic_	routes)
) IA ) )	1.1.1.246 8.8.8.8 10.2.2.2/32	110/11 110/21	10.8.1.51 10.8.1.49	0	-		routes)
AIA D D DIA	1.1.1.246 8.8.8.8 10.2.2.2/32 10.3.3.3/32	110/11 110/21 110/22	10.8.1.51 10.8.1.49 10.8.1.49	0	distance (US-BOS-R2.dynamic_routes)	distance (US-BOS-R2.dynamic_	routes)
AIA D D DIA D	1.1.1.246 8.8.88 10.2.2.2/32 10.3.3.3/32 10.8.0.0/16	110/11 110/21 110/22 110/30	10.8.1.51 10.8.1.49 10.8.1.49 10.8.1.51	0 0 0 0 0 0	distance (US-BOS-R2.dynamic_routes)	distance (US-BOS-R2.dynamic_	routes)
AI () () () () () () () () () () () () () (	1.1.1.246 8.8.8.8 10.2.2.2/32 10.3.3.3/32 10.8.0.0/16 10.8.1.0/28	110/11 110/21 110/22 110/30 110/21	10.8.1.51 10.8.1.49 10.8.1.49 10.8.1.51 10.8.1.49	0 0 0 0 IA 0	distance (US-BOS-R2.dynamic_routes)	distance (US-BOS-R2.dynamic_	routes)
A A D A A D A A D A A D A A D A A D A A D A A D A A D A A D A A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D A D	1.1.1.246 8.8.8.8 10.2.2.2/32 10.3.3.3/32 10.8.0.0/16 10.8.1.0/28 10.8.1.16/28	110/11 110/21 110/22 110/30 110/21 110/21	10.8.1.51 10.8.1.49 10.8.1.49 10.8.1.51 10.8.1.49 10.8.1.49	0 0 0 0 IA 0 0	distance (US-BOS-R2.dynamic_routes)	distance (US-BOS-R2.dynamic_	routes)
D*E2 D IA D D D D D IA D D D IA D D D B B B	1.1.1.246 8.8.8.8 10.2.2.2/32 10.3.3.3/32 10.8.0.0/16 10.8.1.0/28 10.8.1.16/28 10.8.1.32/29	110/11 110/21 110/22 110/30 110/21 110/21 110/20	10.8.1.51 10.8.1.49 10.8.1.49 10.8.1.51 10.8.1.49 10.8.1.49 10.8.1.49	0 0 0 0 1A 0 0 0	distance (US-BOS-R2.dynamic_routes)	distance (US-BOS-R2.dynamic_	routes)

Now, you can define the diagnosis by looping through the merged table to find the missing, added, and changed routes:

- Loop through the merged table, *current\_last\_merged\_routes*, and select both **subnet** fields as the table key. Here, the pulldown options include the column tables from two tables (**table 1** for the **current** route table and **Table 2** for the **last** route table). The column names are identical, with the top half corresponding to **Table 1** and the lower half corresponding to **Table 2**. The same rule applies to all other pulldown options.
- In the **If** condition, select the **current subnet** (the top one) and check whether it is empty. If so, add a diagnosis message: *The entry for \$subnet1 is missing*. You should use the last subnet in the message.

📄 Add N	ote D Add Diagnosis	Can also click a variable on	the left to add automation.
Name:	Compare current and last routes	Anchor:	~ <b>`</b>
	Type description of the diagnosis		
	Table Rows III current_last_merged_routes V Table i	Key: subnet, subnet 🗸	©
∨ If		protocol	
A	e US-BOS-R2 Current ✓	✓ subnet	<ul> <li>Table 1 (current</li> </ul>
2	subnet (current) Is empty		
в	Select Variable	protocol	
_		🗹 su <sup>ll</sup> et	<ul> <li>Table 2 (last)</li> </ul>
~ Then		distance	
📄 Dia	gnosis Message:	next_hop	Jent =
	<ul> <li>The entry for \$subnet1 is missing.</li> </ul>		-for star
<b>2</b> -5	Set Status Code for Device:		
	Error V The entry for \$subnet1 is missing.		

- 3. Add an **elseif** condition to check whether a route entry is added.
- 4. Add another **elseif** condition to check whether the next hop changed.

2. Define Diagnosis	
Add Note D Add Diagnosis	Can also click a variable on the left to add automation.
∽ ElseIf	🗊 Delete 🔺
A 😁 US-BOS-R2 Current 🗸	
3 subnet Last ~ Is empty B Select Variable ~	~
∽ Then	
Diagnosis Message:	□ Save to Incident 🔳
A new route for s\$subnet is added	Ł
Set Status Code for Device:	
● Error	s added
Set Status Code for Intent:	
● Error ∨ A new route for s\$subnet is	s added
Add Logic 🗸 😽	
∽ ElseIf	<u> </u> Delete
A 😁 US-BOS-R2 Current 🗸	Current 🗸
4 next_hop Current Does not e	equal ~ next_hop Last ~ 🖻
B Select Variable ∽	
∽ Then	
📄 Diagnosis Message:	□ Save to Incident 📃
The next hop of \$subnet changes	from \$next_hop1to \$next_hop!
✓-S Set Status Code for Device:	
+ Add Elself + Add Else	

5. Save and run the intent. Check the result for its correctness.

### 11.2.3 Organize and Run Automations Before Change

After you have all intents ready for a network change task, you need to decode/replicate the intents so that it can be run for the change and possibly affected devices. You can use **Auto Intent Wizard** so that users can replicate and run these intents under the **Auto Intent** pane in the map for the network change. Refer to Chapter 4 for the detailed instructions. Another way (maybe more organized and easier to manage) is to use ADT as follows:

- 1. Create a device group for the devices that will be changed or affected by the change. You can create the device group from the map directly.
- 2. Create an ADT, *For Add Subnet CM Task*, for the change management task, and build the base from the device group created in step 1.
- 3. Use the **Intent Replication Wizard** to replicate the intents (such as the **Configuration Change** you created in 2.1.1 and **Compare Dynamic Routes** you just created in 2.2.3) to the ADT created in step 2. You can select the device group created in step 1 as the intent qualification. And add the additional fields into the ADT such as intent/device status code, and last execution time.

The ADT will be like this:

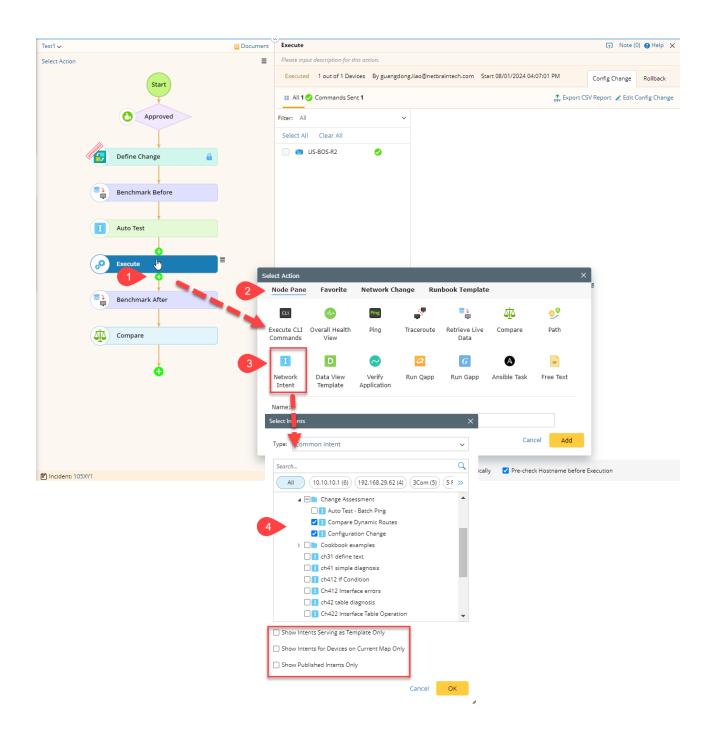
電, For Add Subnet CM Task Table Builder Loss Updated at: 08/02/2024 02:23 PM 🔍 Rebuild Table					/02/2024 02:23 PM 🔍 Rebuild Table				Add Data Manually 🗸 📃 💼
Descrip	ption: Type description here								
Items:	3 Rows 8 Columns							Search	Q T Advanced Filter: Undefined C
No.	Device	s Vendor		s Model	S Software Version	Configuration Change Intent	0	S Device Status Code	Route Change Intent
	US-BOS-R2	Cisco		CGS-MGS-AGS	15.4(2)T4	Configuration Change US-BOS-R2	0	Intent 1	Compare Dynamic Routes US: "
2	US-BOS-SW2	Cisco	Base	3560E	15.2(HI_20170202)FLO_D5G57	Configuration Change US-BOS-SW2	0	Intent 1	Compare Dynamic Routes US
	US-BOS-SW3	Cisco		3560E	15.2(HI 20170202)FLO DSGS7	Configuration Change US-BOS-SW3	0		2

You can set up an Intent Timer to run all replicated intents of this ADT, or you can just run once before and after the change.

The Auto Test, such as **Batch Ping**, has its own ADT, and you should also set up an Intent Timer to run these auto tests.

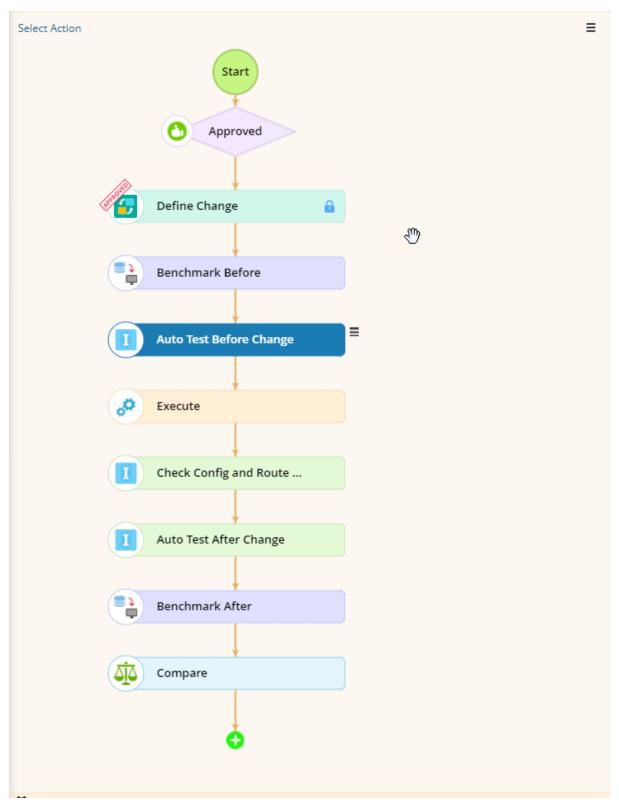
To remind yourself or others to run these intents, you can add intent nodes into the CM runbook. To add an intent node into the runbook,

- 1. Click the + sign before or after an existing node where you want to add a node.
- 2. In the **Select Action** window, select the **Node Pane** tag.
- 3. Select Network Intent.
- 4. In the **Select Intent** window, uncheck all checkboxes at the lower pane and select the intents you want to add. You can select more than one intent.



5. Since you cannot select the intents of an ADT, so, in the node **description**, you may remind the users of this runbook (which can be yourself or others) that they should run intents of a certain ADT instead of just intents selected here.

	I Intent	Runbook	D Data View	«				
est1 🗸			Document	Check Config a	nd Route Cha	nge		(1) Note (0)
elect Action			5	Please go to the	e ADT For Ad	d Subnet CM Task a	nd run all replicate	ed intents afte
				ltems:2 + Ad	ld Intent			
		Start		Intent Name	Target Dev	Status Code	CSV Repor	Actions
		Approved		Compare Dyna	1			
		Approved		Configuration (	: 1			
	1a	Ļ						
	STREET.	Define Change						
		¥ Benchmark Before						
				la contraction de la contracti				
		↓ ↓						
		Auto Test		Items:0 Filter Intent by Tags:	t una ta antant tanan		ilter	
		↓						
	<b>0</b> 0	Execute		Intent Name	Target Dev	Status Code	CSV Repor	Diagnosis T
	I	Check Config and Route	=					
		Benchmark After						
	Ţ							
		↓						
		Compare						
		Ó						
								Run
Incident: 1								



After adding all intents to the Runbook, you may end up with an intent-based CM flow:

# 11.2.4 Execute Change

Executing change is straightforward. Click the **Execute** node, select the device, and click the **Execute** button. You can select the execution mode: **one by one automatically, in batch automatically**, or **one by one manually**. After the execution, you can select a device and view the execution log.

Execute						[†] Note	e (0) 😮 Help
Please input description for this ad	ction.						
Executed 1 out of 1 Devices	By guangdong	liao@netbraintech.com,	Start 0	3/02/2024 03:58:57 PN	Л	Config Change	Rollback
📰 All 1 🤣 Commands Sent 1				1	Export CS	iV Report 🛛 🗶 Edit (	Ionfig Change
Filter: All	~	CS-BOS-R2					
Select All Clear All		Commands to be config t interface loopback1		device			
US-BOS-R2		<pre>ip address 192.168. ip ospf 1 area 0 exit router ospf 1 network 192.168.1.0 exit exit exit  Login to device Pre-check hostname: Retrieved hostname  Login to device Sending task(contai Dispatch the task t Telnet to device 10 User Access Verific Username: nb Password: US-BOS-R2&gt;enable Password: US-BOS-R2#enable US-BOS-R2#enable US-BOS-R2#enable US-BOS-R2#enable US-BOS-R2#config t Enter configuration US-BOS-R2(config)#i US-BOS-R2(config)if US-BOS-R2(config-if US-BOS-R2(config-ro US-BOS-R2(config-ro US-BOS-R2(config-ro</pre>	1.101 2 0 0.0.0. e and pr Succes "US-BOS e and se ining 9 to live 0.8.1.24 cation length n comman interfac b)#ip os b)#exit outer o buter)#n	255 area 0 e-check hostname sful -R2" matched nd commands commands) to Front thread 0 ds, one per line. e loopback101 dress 192.168.1.101 pf 1 area 0 spf 1	Server(ti End with 1 255.255.	CNTL/Z. 255.0	
0 Devices Selected	ecute	ঞ One by One Automat	tically	Pre-check Hostna	me before l	Execution	

### 11.2.5 Verify Change

After the change, you can run all auto-tests you want to, for example, **Batch Ping**, to verify that all critical servers are still pingable.

In our example, you can open the ADT *For Add Subnet CM Task*, and run all intents to check the configuration change and route change.

First, check the configuration changes in the change device to confirm that the configuration is indeed changed as expected.

US-BOS	5-R2 👩 Configuration		Execution Time: 08/02/2024 04:24:39
Diagno Summary	osis Details Compare y Text Original Text 08/02/2024 04:24:39 PM Search Q V	<b>`</b>	Diagnosis Logic(What has Changed)
1 2 3 4 5 6 7 7 8 9 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	US-805-R2#show run Building configuration Current configuration : 12685 bytes ! ! Last configuration change at 14:59:18 EST Fri Aug 2 ! version 15.4 service timestamps debug datetime msec service timestamps log datetime msec service password-encryption mostname US-80S-R2   boot-start-marker boot-start-marker lenable secret ******** no aaa new-model clock timezone EST -5 0 mmi polling-interval 60 no mmi auto-configure no mmi auto-configure no mmi pvc mmi snmp-timeout 180	>	Anchor: version 15.4 service timestamps debug datetime msec service if A: US-BO Sconfigurations changed. Added lines: Username ******* interface Loopback101 ip address 192.168.1.101 255.255.255.0 ip ospf 1 area 0 network 192.168.1.0 0.0.0.255 area 0 ! Missing Lines: Diagnosis Message: Configurations changed. Added times: Status Code for Device: S Configurations changed. Added times:

Secondly, check the dynamic route change. Here you should not just check the route changes of the change device, but also of the neighbor routers. For example, the following is the result of the intent, **Compare Dynamic Routes**, of a neighbor router, which shows that a new route is added as expected:

ork Intent (View Mode) - All N	etwork Intents/R&D/GD/Change Assessm	ent/Compare Dynamic Routes/C	Compare Dynamic Routes US-BOS-SW2	
Compare Dynamic Routes I	JS-BOS-SW2			🔔 Open 📄 0 🔥 0 🗶 Edit
sult: 08/02/2024 04:07 PM 🗸	i 💿 🔛			Run with Live Dat
s intent execution is finished	at 08/02/2024 04:07 PM with 0 errors. You	can View Execution Log		
Dynamic route changes	4			View
US-BOS-SW2	S Dynamic route changes dy	namic_routes of US 2	From Seed Device: US-BOS-R2.	
4 💷 show ip route		2 Diagnoses	From Seed Comma	nd: show ip route (Device: US-BOS-R2).
4 N1 - OSPF NS 5 E1 - OSPF es 6 i - IS-IS, s 7 ia - IS-IS, s 8 o - OOR, P - 9 a - applicat 10 + - replicat 11 12 Gateway of last res 13	X - EIGRP external, 0 - 0SPF, IA - SA external type 1, N2 - 0SPF NSA ternal type 1, E2 - 0SPF external t u - 15-15 summary, L1 - 15-15 level nter area, * - canddate default, U periodic downloaded static route, ion route ed route, % - next hop override ort is 10.8.1.18 to network 0.0.0.0 0/1] via 10.8.1.18, 6wld, Vlan101 0/1] via 10.8.1.2, 6wld, Vlan101	external type 2 ype 2 -1, L2 - IS-IS level-2 - per-user static route H - NHRP, 1 - LISP		Dynamic route changes dynamic_routes of US
	subnetted, 1 subnets			A new route for s192.168.1.101 is added

# **12 Map and Document Your Network**

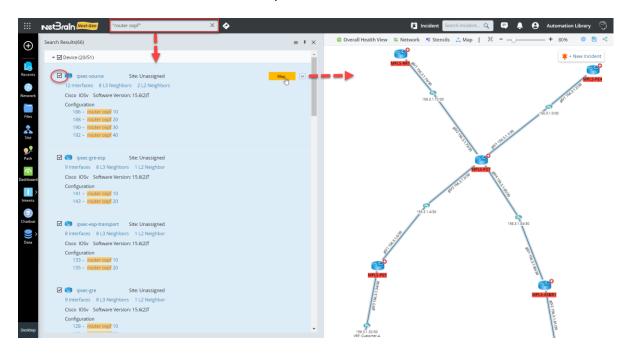
In previous chapters, you have learned how to create the report and document for your network via the ADT and intents. In this chapter, we will revisit these methods and cover how to create the map with the intent.

The chapter covers the following topics:

- Create a map with the intent
- Review the built-in inventory report
- Use ADT to build the inventory report
- Create the report from the CLI command results using the pre-replicated intent template
- Create the CSV report from an intent and ADT

### 12.1Create a Map

The NetBrain dynamic map provides full visibility of your network. There are many ways to create a map manually. For example, you can map the device group and site, enter two IP addresses to discover and map the path, or search and map the devices from the search results. For example, you can search all OSPF devices by the keyword *"router ospf"*. Select all devices or a subset of devices to draw them on the map.



The intent provides a programming way to create the map so that you can create a map according to the CLI command output and have more control over what is mapped, for example,

- Map OSPF neighbors
- Map the multicasting tree
- Map the routing path

### 12.1.1 Create Map with Intent

In this section, we will create an intent to map the OSPF neighbors with the following steps:

Starting with a seed device, use the command, **show ip ospf neighbor**, to retrieve all ospf neighbors. Use the table parser to parse all ospf neighbors from the CLI command output.

- 1. In the diagnosis, for each ospf neighbor,
  - a) Add a logic, **draw device**, to draw this device and the interface to the neighbor.
  - b) Add the other Logic, **follow-up self**, to call this intent for the neighbor devices. With this logic, the intent will be run on the neighbors of neighbors recursively.

12.1.1.1 Parse the OSPF Neighbor Table

From the Intent Manager, create a new intent and name it Map neighbor neighbors:

- 1. Select an OSPF device as the seed device. You can search the keyword "**router ospf**" (add the double quote around the keyword for an exact match) to find the routers with OSPF configured.
- 2. Add a CLI diagnosis.

Network Intent (Edit Mode)		×
I Map neighbor neighbors	VX 🚠 Diagnosis Tree	Run with Live Data Save 3 Help
Type description here		🏅 Intent Map: Select 🗸
I Seed Logic		
🔹 + Device 🚺		Intent Variables: Manager 🔰 Tag: + Add 🔳
V 🥔 US-BOS-SW3	Type Description here	🕫 + Add Config Diagnosis 🛛 🖾 + Add CLI Diagnosis 🔶 2

- 3. In the **CLI Command Diagnosis** window, enter the command, **show ip ospf neighbor**, and click the **Retrieve** button to retrieve the data from the live network.
- 4. Select the table header and first few lines, and click the **Parse Table** button. The system automatically parses the table and creates a table variable.
- 5. You may want to change the default table name, *Table1*, to a meaningful name, such as *ospf\_nbrs*.

CLI Command Diagnosis	
US-BC Show ip ospf neighbor X V Retrieve V	with Live Data
1. Define Variable	2. Define Diagnosis
Format1 V +	Test on Devices: 0
() Double-click a variable to parse. Select multiple lines to parse a table.	Critical Variable (0) Name: Table1 Type: Table Cancel Apply
Current Device     07/17/2024 01:56:36 PM     Search       1     US-BOS-SW1>show ip ospf neighbor       2     3     Neighbor ID       3     Neighbor ID     Pri   State Dead Time Address	Q       ▲       ✓         Header       Heighbor ID       Pri       State       Dead Time       Address       Interface       ∠         Find 6 columns, enter semicolon to separate column.       Find 6 columns, enter semicolon to separate column.       Interface       ∠
4         10.3.3.3         1         FULL/DR         00:00:31         10.8.1.19           5         10.3.3.3         1         FULL/BDR         00:00:39         10.8.1.19           6         155.16.178.124         1         FULL/DR         00:00:34         10.8         4           7         10.8.1.49         1         FULL/BDR         00:00:32         10.8.1.33         8	Image: Second
	Column Variable Type Neighbor ID Sneighbor_id String String Output
I	Sneighbor_id         Spri         State         Sdead_time         Saddress           10.3.3.3         1         FULL/DR         00:00:31         10.8.1.19           10.3.3.3         1         FULL/BDR         00:00:39         10.8.1.3           155.16.178.124         1         FULL/DR         00:00:34         10.8.1.14
4	Image: Pollow         Image: P
Help   All Intent Variables	Cancel Apply

### 12.1.1.2Map Device and Neighbors

After you apply the parser, add a diagnosis:

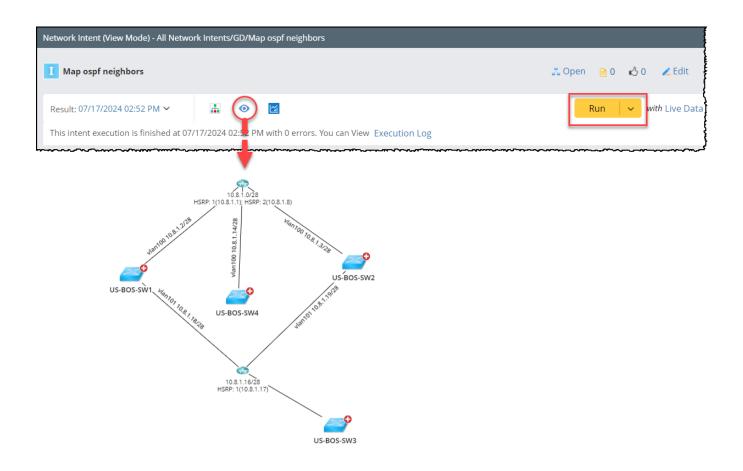
- 1. Change the diagnosis name to a meaning one, such as *Map ospf neighbors*.
- 2. Check the **Loop Table Rows**. Select the table *table ospf\_nbrs* and set the table key as *neighbor\_id*.
- 3. Add if condition as: **neighbor**\_*id* **is not empty**.
- 4. Add a logic: Device Map > Draw Device.

US-BOS-SW1 show ip ospf neighbor	× ~ Retrieve	✓ with Live Data	
1. Define Variable			2. Define Diagnosis
Summary Text Original Text	Search	Q ~ ^	Add Note O Add Diagnosis Can also click a variable on the left to add automation.
3         Neighbor ID         Pri         State           4         10.3.3.3         1         FULL/DR           5         10.3.3.3         1         FULL/BDR           6         155.16.178.124         1         FULL/DR           7         10.8.1.49         1         FULL/BDR	Dead Time Address 00:00:31 10.8.1.13 00:00:34 10.8.1.3 00:00:32 10.8.1.3 00:00:32 10.8.1.33	In VI VI VI Et	Name: Map ospf neighbors     Type description of the diagnosis     Image: Coop Table Rows     Optimized coopsignation of the diagnosis     Image: Coop Table Rows     Image: Coops: Coo
Help   All Intent Variables			Cancel Apply

- a) Under the **Draw Device**, select **this\_device**. The device will be drawn on the map.
- b) Select Include Interface Neighbor and select the *interface* variable of the table.

1. Define Variable		2. Define Diagnosis
Summary Text Original Text	Search Q 🗸	
3         Neighbor ID         Pri         State           4         10.3.3.3         1         FULL/DR           5         10.3.3.3         1         FULL/BR           6         155.16.178.124         1         FULL/BR           7         10.8.1.49         1         FULL/BDR	Dead Time         Addr           00:00:31         10.8           00:00:39         10.8           00:00:34         10.8           00:00:32         10.8	✓ Loop Table Rows          opt_nbrs ∨ Table Key: neighbor_id ∨          ✓ If         A       ✓ US-BOS-S         Current ∨         neighbor_id ∨       Is not empty ∨         B       Select Variable ∨         ∨ Then       >
		Image: International State       Save to Incident         Image: International State       Image: International State         Image: International State       Pop up         Image: International State       Move Down         Image: International State       Delete         Image: International State       Image: International State
		Select Device: E Select Device: this_device a Include: Interface neighbor Interface IPv4 L3 Topology b
4		Add Logic V + Add Elself + Add Else

- 5. Optionally, you can delete the message to make your intent more clear.
- 6. Save and run the intent. Click the icon **View Current Map** to confirm that you have the seed device, and its OSPF neighbors drawn in the map.



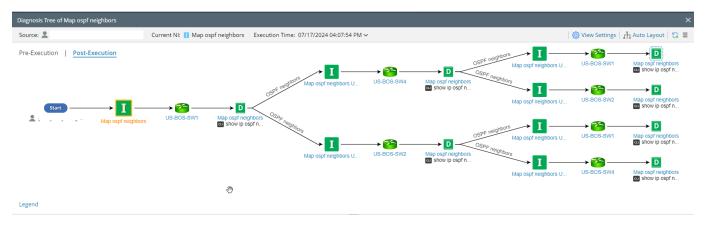
#### 12.1.1.3Map Neighbor Devices

Let us modify the intent to map the neighbor devices recursively. Add the logic, **follow up intent**, and select **Current Intent (Self)**. Click the **Current Intent (Self)** link.

- a) Replicate the current intent to the **Device by Variable**, and select the *neighbor\_id(ospf\_nbrs)*. The system will automatically transfer the IP address (*neighbor\_id*) to the hostname.
- b) Set the maximum depth of the self's execution. The default is 2.
- c) Change the annotations for the execution tree so that it can be easily understood.

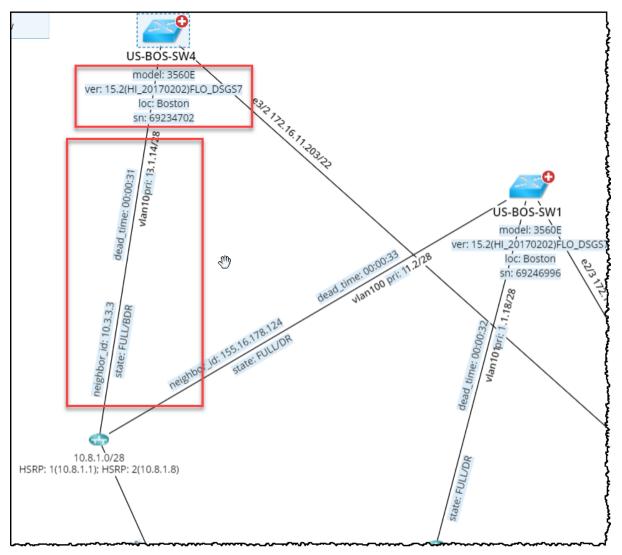
2. Define Diagnosis	Follow-up Self	>
Add Note  Add Diagnosis Can also click a variable on the left to add automation Type description of the diagnosis	Repeat current intent's logic in neighbor or next-hop device recursively.           Description:         When	
✓ Loop Table Rows III ospf_nbrs ~ Table Key: neighbor_id ~	Replication Settings:	
A     US-BOS-S     Current ~       neighbor_id     ~     Is not empty     *       B     Select Variable     ~	Replicate Current a point       Device by Variable       neighbor_id (ospf_nbrs)         Marge multiple replicated intents into one       Max Depth of Follow-up Self's Execution:       Point         Max Depth of Follow-up Self's Execution:       2       D       Follow-up Execution:	r: Settings
✓ Then ⇒ Draw Device:  ■	Draw Arrow from This Device to Next Settings	
Select Device this device  Select Device this device  Interface neighbor  Interface reighbor  Interface IPv4 L3 Topology	Annotation for Diagnosis Tree: * Pre-execution Mode: OSPF Neighbor	G
G Follow-up Intent: Network Intent     Current Intent (Self) Stop      Add Logic      + Add Elself + Add Else	Post-execution Mode: * If the intent is executed: OSPF Neighbor	La La
	Prune other follow-up intents     G	ancel Save

Save and run the intent. Click the icon **View Current Map** to confirm that the OSPF neighbors of the seed device's neighbors are drawn on the map. You can also view the diagnosis tree, which shows the intents are replicated for OSPF neighbors up to two depth levels.



#### 12.1.2 Intent Data View

You can enrich the map with the data view, which can be defined with the logic, **intent data view**. The following diagram is an example of the intent data view. Under the device and along the link, you can select what variables to be displayed. Besides the variables from the intent parser, you can also display the built-in device properties, such as **model**, **software version**, **serial number (sn)**, and **location** displayed here.

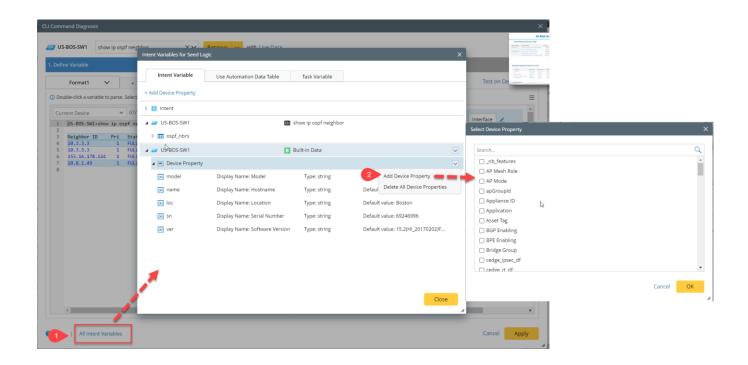


You follow two simple steps to add a data view:

- 1. Add the device properties to the intent variables so that the intent can use these properties.
- 2. Add a new logic, Intent Data View.

Let us edit the diagnosis in the previous session:

- 1. Click the link **All Intent Variables** at the left of the bottom.
- 2. In the **Intent Variables for Seed Logic** window, click **Add Device Property** and select all device properties you want to add. You can search the properties.



- 3. Under the **Define Diagnosis** tag, select **Intent Data View** from the **Add Logic** pull-down menus.
- 4. In the **Intent Data View** window, select the variables or properties you want to display for each position under the device and along the link. While you first add an interface variable along the link, the system may ask you to set the **interface key**, which is the variable *\$interface* of the OSPF neighbor table.

📄 Add Note 🛛 🖸 Add Diagn	osis Can also click	k a variable on the left to add au	omation.	
			A	
Type description of the a	iagnosis			
Loop Table Rows    ospf_	nbrs 🗸 🛛 Table Key: neighbo	or_id 🗸	•	
V If			_ {	
A 🥏 US-BOS-S Curr	ent 🗸		Į	
neighbor_id	✓ Is not empty ✓			
B Select Variable	~			
√ Then				
🍰 Draw Device:			≡	
Create Diagnosis Message	~			
Intent Data View 3 Draw Map >			<b>\$</b>	
Send Email	oor 🗸 Interface	✓ IPv4 L3 Topology ✓	ξ	
Set Intent Paseline				
Advanced >	< Intent	lf) 🔿 Stop	=	
Add Logic 🗸			- {	
+ Add Else				
		Cancel A	pply	
			~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	
L				
•				
•				
+ Device				
+ Device Note		9	<b>0</b>	
Note		🔊 🚺 neighbor_id(string)	<b>○</b>	
+ Device Highlight + Relation	IPv4 Interface 🗸	<b>v</b>		neighbor_id
+ Device Highlight Hostname	IPv4 Interface 🗸	neighbor_id(string)	2	Not
Device	IPv4 Interface 🗸	Ineighbor_id(string)	2 2 2	Not
+ Device + Relation Highlight Hostname	IPv4 Interface 🗸	neighbor_id(string)	2 2 2	🗄 🚺 pri
+ Device Highlight Hostname Model(string)	IPv4 Interface 🗸	Ineighbor_id(string)	2 2 2	neighbor_id pri state dead_time

Save and run the intent. Click the icon **View Current Map** to confirm that the data is displayed on the map. You may need to zoom in to see the data along the link.

#### 12.1.2.1Add OSPF Configurations to Device Note

Often, it is useful to see the relevant configurations while troubleshooting a network problem or documenting a network design, which can be done with the **intent data view**. Before we add the configurations to the intent data view as the device note, you need to parse OSPF configurations with the configuration diagnosis.

Continue editing the intent of the last section. Add a **Configuration Diagnosis**, in which you parse the OSPF configurations as a single variable, *ospf\_conf*:

Define Variable		2. Define Diagnosis	
Format1 V +			Test on Devices: 0
Double-click a variable to parse. Select multiple lines to parse a table.	Critical Variable (0)	🖾 ospf_conf 🗸 🖌 Type: Single 🕖 🕴 + New Pattern 🗸	
Current Device V 07/18/2024 06:00:48 AM route	× • •		
1/9 Interface Vianioi		Start Line: router ospf	× 💿 💼
180 ip flow ingress			
181 ip flow egress			
182 ip address 10.8.1.18 255.255.255.240 183 standby version 2		End Line: 1	✓ ③ 亩
184 standby 1 ip 10.8.1.17			
185 standby 1 timers 5 15			
standby 1 priority 105		Var Line 1 router ospf \$int:id	=
<pre>87 standby 1 preempt 88 ip ospf 1 area 0</pre>			
		190 router ospf 1	> 1 Line
190 <mark>route</mark> r ospf 1	Start Line		
191 router-id 10.2.2.2	Var Line 1		
192 <mark>1</mark> 193 ip forward-protocol nd	Var Line 2	Var Line 2 LinesByVariable[\$ospf_config]:\$id-	=
194 I	End Line		
195 ip flow-export source Vlan101	15	190 router ospf 1	> 1 Line
196 ip flow-export version 9			
197 ip flow-export destination 172.16.9.6 12356 198 !			
190 in the server		+ Field	
200 no ip http secure-server		Output + Parse Lines	
201 !		ouput France Enco	
202 ip access-list extended OUTSIDE		\$id (int) = 1	
203 permit ip 10.8.1.0 0.0.0.255 10.0.0.0 0.255.255.255 204 permit ip 10.8.1.0 0.0.0.255 20.0.0.0 0.0.7.255		<pre>\$ospf_config (string) = router ospf 1 router-id 10.2.2.2</pre>	
204 permit ip 10.0.1.0 0.0.0.255 20.0.0.0 0.0.7.255		sospi_comig(sumig) = router ospi i router id 10.2.2.2	
206 logging origin-id hostname			
207 logging host 172.16.9.225			
208 4	*		

Start Line	router ospf
End Line	vi
Var Line 1	router ospf <b>\$int:id</b>
Var Line 2	LinesByVariable[\$ospf_config]:\$id-

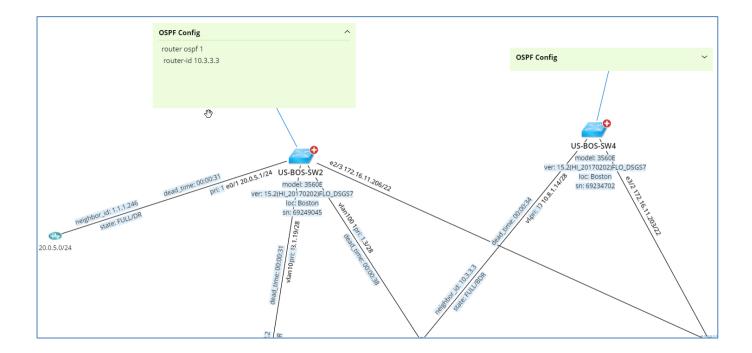
Close this diagnosis, navigate back to the CLI configuration diagnosis created in the last section and click the **Define** link at the left side of the intent data view to edit it.

In the Intent Data View window, click the **Device Note** icon and set the following parameters of the device note:

1 Device Notes	S terface 🗸 🚺 pri(int)	<ul> <li>tate(string)</li> </ul>	+ Interface Note
	I dead_time(string)	🍳 📔 neighbor_id(string)	
Hostname	<b>?</b>	8	nignight
🥺 📧 ver(string)	Define More		
😜 📖 loc(string)			
🍳 📖 sn(string)			
▷ Define More			
Device Note			Reset 🗙
Title:	OSPF Config		
Content:	] Insert Variable		
	ospf_config		
Note Type:	Overwrite 🗸		
Note View:	Collapsed		

- Title: enter a meaning title such as **OSPF Config**.
- Content: inset the variable *ospf\_config*
- Note Type: can be **Overwrite** or **Append**.
- Note View: can be collapsed or Expanded.

Save and run the intent. Click the icon **View Current Map** to confirm that the device note is displayed on the map. You may expand the note.



### **12.2Create the Inventory Report**

### 12.2.1 Built-in Inventory Reports

NetBrain system provides the following built-in inventory reports:

- **Device Report**: contains properties of devices in the current domain, such as management IP and device type.
- **Interface Report:** Contains the interface properties for all devices in the current domain, such as IPv4/6 address and bandwidth.
- **Module Report:** Contains the module properties for all devices in the current domain, such as module ports and slot numbers.
- **Site Report**: Contains the summary information for all sites in the current domain, such as location and device count.
- **Summary Report:** Contains the summary information for all devices in the current domain, such as module count and interface count.
- **FRU Report**: Contains the chassis hardware information about Cisco routers, Cisco IOS switches, Juniper routers, and Juniper EX switches in the current domain.

- **Stackable Switch Report**: Contains the stack information about Cisco Routers or IOS Switches in the current domain.
- **VSS Report**: Contains VSS modules in the current domain.
- **Router\_If\_Report**: Contains the interface properties for routers in the current domain.
- **Switch\_If\_Report**: Contains the interface properties for switches in the current domain.
- **Other\_If\_Report**: Contains the interface properties for other device types in the current domain.

It is easy to create a built-in inventory report:

- 1. Click the **start** menu and select **Inventory Report**.
- 2. Select the target report in the left pane to browse asset details in the report. Its asset details are displayed in the right table.
- 3. Click the  $\equiv$  icon and select **Export**.

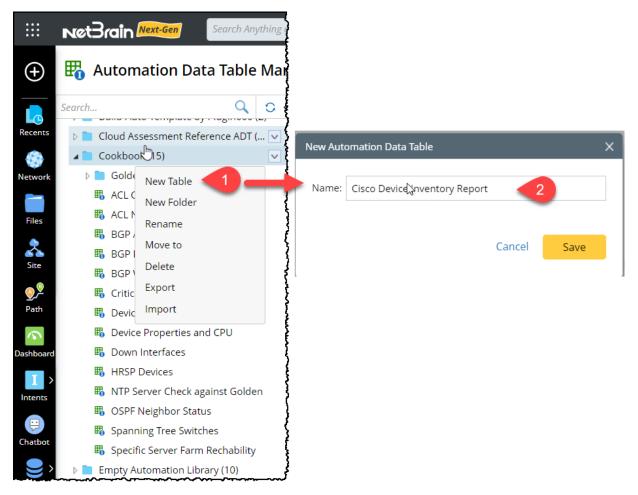
🔹 🚯 Net8: 🗙 😽 Inven 🗙	Built-in Report     Device Report	Total Count: 98 🚹	Filter by: My Net	work	✓ Search		Q	1	🗧 🔞 Help 🛛 Go To Manage Rep	ports Page >>
🔹 📭 NetBi 🗙 😝 Inven 🗙	Interface Report	Hostname	Mgmt IP	Mgmt Interface	Device Type	sysObjectID	Vendor	Model	Copy URL	Serial Num
← → C 📪 nextgen-tr	Module Report	CA-YYZ-AS01-01	10.10.2.3	Management1	Arista Switch	1.3.6.1.4.1.30065.1.2	Arista	VEOS	4.17.2F	System
ii wet3rain	Site Report	CA-YYZ-AS01-02	10.10.2.4	Management1	Arista Switch	1.3.6.1.4.1.30065.1.2	Arista	VEOS	4.17.2F	System
	Customized Reports	CA-YYZ-JR01-01	10.10.2.2	em0.0	Juniper Router	1.3.6.1.4.1.2636.1.1	Juniper	MX150	14.1R4.8	
Search App	🔺 ≽ Shared Reports	DE-MUC-CR01-01	10.20.1.2	Ethernet0/0	Cisco Router	1.3.6.1.4.1.9.1.1	Cisco	CGS-MGS-A	GS 15.4(2)T4	67108880
Network Map-Base	FRU Report     Other_If_Report	DE-MUC-CR01-02	10.20.1.3	Ethernet0/0	Cisco Router	1.3.6.1.4.1.9.1.1	Cisco	CGS-MGS-A	GS 15.4(2)T4	67108960
etwork Analysis	Router_If_Report	DE-MUC-CW01	10.20.1.4	Ethernet0/0	Cisco IOS Switch	1.3.6.1.4.1.9.1.1227	Cisco	3560E	15.2(20170809:1942	67109808
	Stackable Switch Report	DE-MUC-CW02	10.20.1.5	Ethernet0/0	Cisco IOS Switch	1.3.6.1.4.1.9.1.1227	Cisco	3560E	15.2(20170809:1942	67109584
Network	Switch_If_Report	DE-MUC-CW03	10.20.1.6	Ethernet0/0	Cisco IOS Switch	1.3.6.1.4.1.9.1.1227	Cisco	3560E	15.2(20170809:1942	67109664
One-IP Table	VSS Report Private Reports	FR-CDG-AS01-01	10.20.2.3	Management1	Arista Switch	1.3.6.1.4.1.30065.1.2	Arista	vEOS	4.17.2F	System
Device Group		FR-CDG-AS01-02	10.20.2.4	Management1	Arista Switch	1.3.6.1.4.1.30065.1.2	Arista	vEOS	4.17.2F	System
Site		FR-CDG-JR01-01	10.20.2.2	em0.0	Juniper Router	1.3.6.1.4.1.2636.1.1	Juniper	MX150	14.1R4.8	
Site Manager		ipsec-esp-trans	156.2.0.15	GigabitEthernet0/0	Cisco Router	1.3.6.1.4.1.9.1.1041	Cisco	IOSv	15.6(2)T	9BIYR03C
		ipsec-esp-tunnel	156.2.0.14	GigabitEthernet0/0	Cisco Router	1.3.6.1.4.1.9.1.1041	Cisco	IOSv	15.6(2)T	9UA6AS9
Inventory Report		ipsec-gre	156.2.0.12	GigabitEthernet0/0	Cisco Router	1.3.6.1.4.1.9.1.1041	Cisco	IOSv	15.6(2)T	939L4JXV
Change Analysis Report		ipsec-gre-esp	156.2.0.13	GigabitEthernet0/0	Cisco Router	1.3.6.1.4.1.9.1.1041	Cisco	IOSv	15.6(2)T	9QV0AIH
		🥏 ipsec-gw	156.2.0.1	Vlan1	Cisco IOS Switch	1.3.6.1.4.1.9.1.1227	Cisco	3560E	15.2(HI_20170202)FL	67117188
		ipsec-internal-g	156.2.0.18	GigabitEthernet0/0	Cisco Router	1.3.6.1.4.1.9.1.1041	Cisco	IOSv	15.6(2)T	9D9XLJ9C
		ipsec-internal-g	156.2.0.17	GigabitEthernet0/0	Cisco Router	1.3.6.1.4.1.9.1.1041	Cisco	IOSv	15.6(2)T	98GUKOE
		🌀 ipsec-lan	156.2.0.16	GigabitEthernet0/0	Cisco Router	1.3.6.1.4.1.9.1.1041	Cisco	IOSv	15.6(2)T	9FV5YFM
		ipsec-source	156.2.0.11	GigabitEthernet0/0	Cisco Router	1.3.6.1.4.1.9.1.1041	Cisco	IOSv	15.6(2)T	9JDUHOC
		ISP-P01	172.16.0.100	Ethernet0/3	Cisco Router	1.3.6.1.4.1.9.1.1	Cisco	CGS-MGS-A	GS 15.4(2)T4	67108944
		ISP-P02	4.0.0.2	Ethernet0/0	Cisco Router	1.3.6.1.4.1.9.1.1	Cisco	CGS-MGS-A	GS 15.4(2)T4	67109248

Though it is easy to export a built-in inventory report, you cannot customize the fields you want to export. To create an inventory report according to your requirements, build an ADT and export the ADT table to a CSV report.

### 12.2.2 Use ADT to Build Inventory Report

ADT provides a flexible way for you to build the inventory report according to your requirements:

- 1. In the **Automation Data Table Manager** window, right-click a folder and select the menu item **New Table**.
- 2. In the **New Automation Data Table** window, enter a name, such as the **Cisco Device Inventory Report** and click the **Save** button.



- 3. Select the table you just created.
- 4. Click the Table Builder to open the Automation Data Table Builder window.
  - a) Select the method **Devies of Device Group** as the method to build the base table.
  - b) Select a device group, such as a device group to include all Cisco Devices.
  - c) Drag the fields you are interested into the right pane. You can change the header name.
  - d) Click the Save and Builder.

Automation Data Table Mana	iger	
Secorch	Description: Type description here Items: 0 Rows 0 Columns Automation Data Table Builder	× set All
	Base       •       Column Group (Base):       Select Column         Description:       •	

5. Click the **Build** button in the pop-up window **Build Table**.

Build Table				×
Build the column groups:	Alm			~
Log:	Production Mode	Only show major execution	process log	
	○ Debug Mode	Show all the detailed log		
			Cancel Buil	d <b>5</b>

6. Wait for the system to finish building the table. Some data cells may be blank. Click the menu bar at the top right corner and select **Export to CSV Only** to export the data into a CSV file.

iearch Q O K	< 🖪 Ci	sco Device Inventory Report	Table Builde	Last Updated at: 07/15	/2024 03:11 PM 🔍 Rebuild Ta	ble	Add Data Manually 🗸	=
📄 Shared Tables (465)	^ ·	and the state of the state					Import from System Table	e
OPractice (149)	Descr	iption: Type description here					Lock Settings	
Automation Library (30)	Items	: 30 Rows 7 Columns				Search	Execution log	
<ul> <li>Automation Library Kunal (43)</li> <li>Automation Library PKG11 (0)</li> </ul>	No.	Device	S Mgmt IP	S Mgmt Interface	5 Model	Software Version	Export	Lo
<ul> <li>Build-Auto Template by Plugin (2)</li> </ul>			_ 0			6	Export to CSV Only	
<ul> <li>Build-Auto Template by Plugin (E)</li> <li>Build-Auto Template by Plugin PKG (2)</li> </ul>	1	Berlin-R1	172.16.8.60	Ethernet0/1	CGS-MGS-AGS	15.4(2)T4	67 Export Datasets to File	
Build-Auto Template by Plugin666 (2)	2	DE-MUC-CR01-01	10.20.1.2	Ethernet0/0	CGS-MGS-AGS	15.4(2)T4	67 Dataset Tag Settings	:-4
Cloud Assessment Reference ADT (28)	3	DE-MUC-CR01-02	10.20.1.3	Ethernet0/0	CGS-MGS-AGS	15.4(2)T4	67 Table Settings	:-4
🖌 📄 Cookbook (16)	4	ISP-P02	4.0.0.2	Ethernet0/0	CGS-MGS-AGS	15.4(2)T4	67109248	
👂 📄 Golden Template (1)	5	ISP-PE01	1.0.0.2	Ethernet0/0	CGS-MGS-AGS	15.4(2)T4	67109152	
ACL Configuration Drift	6	ISP-PE02	2.0.0.2	Ethernet0/0	CGS-MGS-AGS	15.4(2)T4	67109024	
B ACL Name List		ISP-PE03						
BGP Automation Table	7		3.0.0.2	Ethernet0/0	CGS-MGS-AGS	15.4(2)T4	67108992	
BGP Document and TS BGP Wrapper	8	JP-TYO-CR01-01	10.30.0.2	Ethernet0/0	CGS-MGS-AGS	15.4(2)T4	67109104	DC-3
GSP wrapper     GSP cisco Device Inventory Report	9	JP-TYO-CR01-02	10.30.0.3	Ethernet0/0	CGS-MGS-AGS	15.4(2)T4	67109088	DC-3
Critical Application	10	NORI30148WRA1A	10.87.122.5	TenGigabitEthernet0/11(1	ASR1001-HX	16.12.05	TKM21080352	US -
B Device Performance Data	11	NORI30148WRC0B	10.87.122.130	TenGigabitEthernet0/1/5	ASR1001-HX	16.12.05	TKM21110248	US -
Device Properties and CPU	12	RT01-VNDR-SYKE-ASH-V	10.78.252.1	Loopback0	ISR4431/K9	16.12.05	FJC2332A06R	
B Down Interfaces	13	RT01-VNDR-SYKE-CENT	10.78.253.132	GigabitEthernet0/0/0.100	ISR4431/K9	16.12.05	FJC2332A06Q	
HRSP Devices								
🚯 NTP Server Check against Golden	14	RT01-VNDR-TTEC-AUS-T	10.78.16.1	Loopback0	ASR1002-HX	16.12.05	FX52144Q06J	
B OSPF Neighbor Status	15	RT01-VNDR-TTEC-CENT	10.78.24.22	TenGigabitEthernet0/1/	ASR1002-HX	16.12.05	FXS2144Q065	
Spanning Tree Switches	16	RT02-VNDR-SYKE-ASH-V	10.78.252.133	GigabitEthernet0/0/1.100	ISR4431/K9	16.12.05	FJC2332A065	
Becific Server Farm Rechability	17	RT02-VNDR-SYKE-CENT	10.78.253.2	Loopback0	ISR4431/K9	16.12.05	FJC2332A06T	
<ul> <li>Empty Automation Library (10)</li> <li>Empty Staging Automation Library (24)</li> </ul>	18	RT02-VNDR-TTEC-AUS-T	10.78.16.133	TenGigabitEthernet0/1/	ASR1002-HX	16.12.05	FXS2222Q2XH	

Similarly, you can create an inventory report at the interface level. Follow the same steps and select the method **Interfaces of Device Group** at step 6a. In step 6b, you can only select the device group that is created with the static interface or dynamic interface search.

C	c1 😜	C	c2	s C	c3	s C	i1	I 0	c4	s ()	c5	S	сб	s 🕻	c7	5
[	Device	In	terface Name		IPv4 Address		Bandwidth		Speed		Duplex		Inbound ACL		Live Status	
ase	+			(	Column Group (Ba	se):									Select	Column 🗸
escriptior	n:				c1	*	1	:2	S	c3	5		ii 🔲		c4	S
					Devi			ice Name	£	IPv4 Addre		<u>n</u>	Bandwidth	0	Speed	
elect Met	hod to Build Bas	e Table:			Devio			ice Name		IPv4 Addre	255		Bandwidth	_	Speed	
Interfaces	s of Device Grou	р	~		c5 Duple	s	Live	:7 Status	S							
elect Inter	rfaces by Device	Group:	Shared Devic		Dupl		(1)	Status								
U	Bandwidth	L	^													
£	Speed	S														
£	Duplex	S					G									
£	Live Status	s														
1	MAC Address	S														
	Slot#	S														
1	Module Type	S														
	Description	S														
Ro	outing Protocol	S														
1	Tunnel Mode	s														
					(Drag and drop co	lumn heade	er from the upper o	area or availa	able data fiel	d from the lef	t here.)					
Mu	Ilticasting Mode	S	v													

You can view the interface data and export it to a CSV file.

👪 In	terface Level Inventory Report	Table Bu	uilder Last Updated at:	07/15/2024 03:39 PM 🛛 🔍 Reb	uild Table			Add Data Manually $\sim~\equiv$
Desci	iption: Type description here							
Items	: 325 Rows 8 Columns						Search Q	Advanced Filter: Undefined
No.	Device	S Interface Name	SIPv4 Address	Bandwidth	Speed	5 Duplex	S Live Status	옯Interface
21	ITE_EXTEND	GigabitEthernet0/45		1000000	a-1000	a-full	up/up	ITE_EXTEND - Gigab
2	ITE_EXTEND	GigabitEthernet0/44	=	1000000	a-1000	a-full	up/up	ITE_EXTEND - Gigab
3	ITE_EXTEND	GigabitEthernet0/43	J	10000	auto	auto	down/down	ITE_EXTEND - Gigab
4	ITE_EXTEND	GigabitEthernet0/42		10000	auto	auto	down/down	ITE_EXTEND - Gigab
5	ITE_EXTEND	GigabitEthernet0/41		10000	auto	auto	down/down	ITE_EXTEND - Gigał
6	ITE_EXTEND	GigabitEthernet0/40		1000000	a-1000	a-full	up/up	ITE_EXTEND - Gigal
7	ITE_EXTEND	GigabitEthernet0/39		10000	auto	auto	down/down	ITE_EXTEND - Gigal
8	ITE_EXTEND	GigabitEthernet0/38		1000000	a-1000	a-full	up/up	ITE_EXTEND - Gigat
9	ITE_EXTEND	GigabitEthernet0/37		1000000	a-1000	a-full	up/up	ITE_EXTEND - Gigab
)	ITE_EXTEND	GigabitEthernet0/36		1000000	a-1000	a-full	up/up	ITE_EXTEND - Gigal
1	ITE_EXTEND	GigabitEthernet0/35		10000	auto	auto	down/down	ITE_EXTEND - Gigal
2	ITE_EXTEND	GigabitEthernet0/34		1000000	a-1000	a-full	up/up	ITE_EXTEND - Gigal
3	ITE_EXTEND	GigabitEthernet0/33		1000000	a-1000	a-full	up/up	ITE_EXTEND - Gigab
4	ITE_EXTEND	GigabitEthernet0/32		1000000	a-1000	a-full	up/up	ITE_EXTEND - Gigał
5	ITE_EXTEND	GigabitEthernet0/31		10000	auto	auto	down/down	ITE_EXTEND - Gigal
5	ITE_EXTEND	GigabitEthernet0/30		10000	auto	auto	down/down	ITE_EXTEND - Gigal
7	ITE EXTEND	GigabitEthernet0/29		10000	auto	auto	down/down	ITE_EXTEND - Gigab

# **12.3Create the Report from CLI Command Results**

If the built-in data cannot satisfy your need, you may retrieve the data from the CLI commands and create a report from the output of the CLI commands. As you have learned from the previous chapters, there are two ways to create a report from the CLI command results:

- Create a CSV report from an intent and replicate the intent to an ADT. Then from the ADT, you can create a summary report, which combines the CSV reports of all replicated intents.
- Build an ADT base from Pre-Replicated Intent Templates. In the intent replication setting, define all variables you want to export to the ADT as the **signature variables**.

### 12.3.1 Create CSV Report from CLI Command Results

In section 5.2, you have learned how to create a CSV report of all ACL configurations. Let us review the key steps with a simple example: create a report for all interfaces which has input errors larger than 0.

Create an intent **Interface with Input Errors**. Issue the CLI command, *show interfaces*, and parse the interface name, status and input errors with the **Paragraph** Parser. The Paragraph Parser has the ID line, *\$if\_name is \$status, line protocol is,* and Var Line 1 as, *\$int:input\_err input errors*. You may add other variables such as *input rate, CRC error*, etc.

		with Live Data			
Define	2 Variable		Define Diagnosis		
	Format1 V +			Tes	st on Devices: 0
Doub	le-click a variable to parse. Select multiple lines to parse a table.	Critical Variable (0)	🕯 interfaces 🗸 🖌 🗸 Type: Paragraph	+ New Pattern 🗸	
Curre	ent Device  v 07/18/2024 02:46:47 PM Search	Q 🔺 🔻			
2	<pre>#Ethernet0/0 is up, line protocol is up (connected)</pre>	P1-ID Line A	ID Line A \$if_name is \$status, line p	rotocol is	=
3	Hardware is Ethernet, address is aabb.cc00.1400 (bia aa		2 Ethernet0/0 is up, line proto		> 18 Lines
4	MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec, reliability 255/255, txload 1/255, rxload 1/255		2 Ethernetoro is up, line proto	coris up (connected)	> 18 Lines
6	Encapsulation ARPA, loopback not set				
7	Keepalive set (10 sec)		Verlie 1 dt i tra		=
8	Half-duplex, Auto-speed, media type is RJ45		VarLine1 \$int:input_err input err	rors	=
9 10	input flow-control is off, output flow-control is unsup ARP type: ARPA, ARP Timeout 04:00:00		21 0 input errors, 0 CRC	C, 0 frame, 0 overrun, 0 ignor	N 4011
11	Last input 00:00:01, output 00:00:00, output hang never		21 Unput errors, 0 CRC	c, o frame, o overrun, o ignor	> 18 Lines
12	Last clearing of "show interface" counters never				
13	Input queue: 0/2000/0/0 (size/max/drops/flushes); Total		+ Field ~		
14	<pre>Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo</pre>		+ Field 🗸		
14 15	Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo Output queue: 0/0 (size/max)		A		
14	<pre>Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo</pre>				-
14 15 16 17 18	Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo Output queue: 0/0 (size/max) 5 minute input rate 0 bits/sec, 0 packets/sec 5 minute output rate 2000 bits/sec, 4 packets/sec 10275792 packets input, 8&4746524 bytes, 0 no buffer		A	Sinput_err	-
14 15 16 17 18 19	<pre>Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo Output queue: 0/0 (size/max) 5 minute unput rate 0 bits/sec, 0 packets/sec 5 minute output rate 2000 bits/sec, 4 packets/sec 10275792 packets input, 884746524 bytes, 0 no buffer Received 9493391 broadcasts (0 multicasts)</pre>		Output     + Parse Lines       Sif_name     >       \$status	✓ \$input_err	~
14 15 16 17 18 19 20	<pre>Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo Output queue: 0/0 (size/max) 5 minute input rate 0 bits/sec, 0 packets/sec 5 minute output rate 2000 bits/sec, 4 packets/sec 10275792 packets input, 884746524 bytes, 0 no buffer Received 9493391 broadcasts (0 multicasts) 0 runts, 0 glants, 0 throttles</pre>	Var Line 1	Output + Parse Lines		~
14 15 16 17 18 19 20 21	<pre>Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo Output queue: 0/0 (size/max) 5 minute input rate 0 bits/sec, 0 packets/sec 5 minute output rate 2000 bits/sec, 4 packets/sec 10275792 packets input, 88/746524 bytes, 0 no buffer Received 9493391 broadcasts (0 multicasts) 0 runts, 0 giants, 0 throttles 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored</pre>	Var Line 1	Output     + Parse Lines       Sif_name     >       \$status	✓ \$input_err	~
14 15 16 17 18 19 20	<pre>Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo Output queue: 0/0 (size/max) 5 minute input rate 0 bits/sec, 0 packets/sec 5 minute output rate 2000 bits/sec, 4 packets/sec 10275792 packets input, 884746524 bytes, 0 no buffer Received 9493391 broadcasts (0 multicasts) 0 runts, 0 glants, 0 throttles</pre>	Var Line 1	Output     + Parse Lines       bilf_name         thernet0/0     up	✓ \$input_err 0	~
14 15 16 17 18 19 20 21 22	Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo Output queue: 0/0 (size/max) 5 minute input rate 0 bits/sec, 0 packets/sec 5 minute output rate 2000 bits/sec, 4 packets/sec 10275792 packets input, 884746524 bytes, 0 no buffer Received 9493391 broadcasts (0 multicasts) 0 runts, 0 giants, 0 throttles 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored 0 input packets with dribble condition detected	Var Line 1	Output     + Parse Lines       bilf_name         thernet0/0     up	✓ \$input_err 0	~
14 15 16 17 18 19 20 21 22 23 24 25	<pre>Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo Output queue: 0/0 (size/max) 5 minute input rate 0 bits/sec, 0 packets/sec 5 minute output rate 2000 bits/sec, 4 packets/sec 10275792 packets input, 884746524 bytes, 0 no buffer Received 9493391 broadcasts (0 multicasts) 0 runts, 0 giants, 0 throttles 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored 0 input packets with dribble condition detected 80962979 packets output, 6955392766 bytes, 0 underru 0 output errors, 0 collisions, 0 interface resets 0 unput packets drouge 1</pre>	····-Var Line 1	Output     + Parse Lines       Sif_name P     Status       Sthernet0/0     up       Ethernet0/1     up       Ethernet0/2     up	✓ \$input_err 0 0 0 0	~
14 15 16 17 18 19 20 21 22 23 24 25 26	<pre>Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo Output queue: 0/0 (size/max) 5 minute input rate 0 bits/sec, 0 packets/sec 5 minute output rate 2000 bits/sec, 4 packets/sec 10275792 packets input, 8&amp;4746524 bytes, 0 no buffer Received 9493391 broadcasts (0 multicasts) 0 runts, 0 giants, 0 throttles 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored 0 input packets with dribble condition detected 80962979 packets output, 695392766 bytes, 0 underru 0 output errors, 0 collisions, 0 interface resets 0 unknown protocol drops 0 babbles, 0 late collision, 0 deferred</pre>	Var Line 1	Output     + Parse Lines       Slif_name IP <ul> <li>Status</li> <li>Ethernet0/0</li> <li>up</li> <li>Up</li> <li>Ethernet0/1</li> <li>up</li> <liup< li=""> <liup< li=""> <li>Up</li>         &lt;</liup<></liup<></ul>	\$input_err 0 0	~
14 15 16 17 18 19 20 21 22 23 24 25 26 27	<pre>Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo Output queue: 0/0 (size/max) 5 minute input rate 0 bits/sec, 0 packets/sec 5 minute output rate 2000 bits/sec, 4 packets/sec 10275792 packets input, 8847d524 bytes, 0 no buffer Received 9493391 broadcasts (0 multicasts) 0 runts, 0 giants, 0 throttles 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored 0 input packets with dribble condition detected 80962979 packets output, 6955392766 bytes, 0 underru 0 output errors, 0 collisions, 0 interface resets 0 unknown protocol drops 0 babbles, 0 late collision, 0 deferred 0 lost carrier, 0 no carrier</pre>		Output     + Parse Lines       Sif_name P     Status       Sthernet0/0     up       Ethernet0/1     up       Ethernet0/2     up	✓ \$input_err 0 0 0 0	~
14 15 16 17 18 19 20 21 22 23 24 25 26	<pre>Input queue: 0/2000/0/0 (size/max/drops/flushes); Total Queueing strategy: fifo Output queue: 0/0 (size/max) 5 minute input rate 0 bits/sec, 0 packets/sec 5 minute output rate 2000 bits/sec, 4 packets/sec 10275792 packets input, 8&amp;4746524 bytes, 0 no buffer Received 9493391 broadcasts (0 multicasts) 0 runts, 0 giants, 0 throttles 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored 0 input packets with dribble condition detected 80962979 packets output, 695392766 bytes, 0 underru 0 output errors, 0 collisions, 0 interface resets 0 unknown protocol drops 0 babbles, 0 late collision, 0 deferred</pre>	Var Line 1	Output     + Parse Lines       Slf_name     >       Sthernet0/0     up       Ethernet0/1     up       Ethernet0/2     up       Ethernet0/3     up	✓ \$input_err 0 0 0 0 0 0 0 0 0	~

- 1. Add a diagnosis to export the interfaces with the input errors.
  - a) Loop through the table variables defined in step 1, **interfaces**. Define the condition as, *If the input\_errs is greater than 0*, add a logic, **Export to CSV Report**.
  - b) Define the CSV report.
  - c) Click the link **Manage All CSV Reports** and add a CSV report file. Define the file name and column names, which are separated by the comma (,).
  - d) Define the mapping between the variables and column names.

2. Define Diagnosis	
Add Note • Add Diagnosis Can also click a variable on the left to add au	tomation
Name: Report Interface with input errors Anchor:	~
Type description of the diagnosis	
	{
✓ Loop Table Rows ↓ Interfaces ∨ Table Key: if_name ∨ ✓ If	©
A 🥥 US-BOS-S Current 🗸	
input_err v Greater than v 0 a v	
B Select Variable V	
✓ Then	
Export to CSV Report: Define	≡
Add Logic 🗸	
CLI Command Diagnosis	
Export Data to CSV for US-BOS-SW1	Intent Settings
Export the row data of the "interfaces" to CSV Report.	Intent Map and Data View Execution Settings Embedded Incident CSV Report Files Follow-up Intent
Interfaces with Input Errors 🔹 🗟 Manage All CSV Reports	Define the CSV report files
Define the column mapping from the "interfaces" to the selected CSV Report.	+ Add CSV
Hostname Interface name Intput errors	8
\$this_device	* CSV Name: Interfaces with Input Errors
	* Columns: Hostname, Interface name, Intput errors
	Ca
	* Location: Browse
	* CSV File:
	• The CSV file name will be reset as file name + intent name when this intent
	is used as intent template/intent cluster.
	Cancel OK

- 2. Replicate the seed intent to a new ADT. Select the **Intent Replication Wizard** from the intent menu bar and follow the wizard to create a new ADT:
  - a) In 1<sup>st</sup> step (Seed Intent), select the template for Device-based Replication.
  - b) In 2<sup>nd</sup> step (**Define ADT**), select **Create a New ADT**. Define the name and select the device group(s) to include deivces for the intent to replicate on.
  - c) In the 3<sup>rd</sup> step, define the **intent qualification**. You can qualify by the device group and select the same device groups as step b.
  - d) In the 4<sup>th</sup> step, build and replicate the intents.

Nizard - Interface	with Input Errors							
eed Intent		Define ADT		Re	plication Settings			Replicate Intent
	(	Create a New ADT			Use an Existing	ADT		
	Auto	mation Data Table						
	New ADT by De	evice Group	Ş				>	<
	Name:	Interface with Errors	20	Location:	Cookbook		~	
	Description:							
	Create from S	Selected Device Groups: 🤇	2 Device Groups 🗸	<b>a</b>				
	へ Additional	Columns						
						Cancel	Save	

- 3. Open the ADT created in step 3 and wait for the replication process to finish.
  - a) Select the cloned intent column and select **Run Intents Once**. Wait for the system to finish the executions.
  - b) Select the cloned intent column and select **View Summary Report**.

d Create I	Мар Q 📀		🌠 Incident 🛛 Search Incident 🔍 📑 🌲 😝 Top3 Solution Lab 📿
ger			Help
👪 In	terface Having Input Errors	Table Builder	Last Updated at: 07/18/2024 04:45 PM 🛛 🌯 Rebuild Table Add Data Manually 🗸 🚍 🛾
Desci	ription: Type description here		
Items	: 47 Rows 2 Columns		Search Q TAdvanced Filter: Undefined
No.	Device		Replicated Intent
29	UK-LHR-CR01-02		Interface with Input Errors UK-LHR-CR01-02 a Run Intents Once
30	UK-LHR-CW01-01		Interface with Input Errors UK-LHR-CW01-01 1 Run Intents Timer
31	UK-LHR-CW01-02		Interface with Input Errors UK-LHR-CW01-02 1 Open Seed Intent
32	US-BOS-CR01-01		Interface with Input Errors US-BOS-CR01-01 1 Rebuild Intent-related Column Group
33	US-BOS-CR01-02		Interface with Input Errors US-BOS-CR01-02 1 Remove Empty Wrapper Intent
34	US-BOS-CW01-01		Interface with Input Errors US-BOS-CW01-01 1
35	US-BOS-CW01-02		Interface with Input Errors US-BOS-CW01-02 1 Export Diagnosis Result to CSV
36	US-BOS-R1		b View Summary Report
37	US-BOS-R2		Interface with Input Errors US-BOS-R2 1 Debug Empty Cells
38	US-BOS-SW1		Interface with Input Errors US-BOS-SW1 1
39	US-BOS-SW2		Interface with Input Errors US-BOS-SW2 1 Tag Current Column
40	US-BOS-SW3		Interface with Input Errors US-BOS-SW3 1 Edit
41	US-BOS-SW4		Interface with Input Errors US-BOS-SW4 1 Delete
42	US-NYJ-CR01-01		Interface with Input Errors US-NYJ-CR01-01 1 Set as Table Key
43	US-NYJ-CR01-02		Interface with Input Errors US-NYJ-CR01-02 1 Submit Related Commands to Benchmark
44	US-NYJ-CW01-01		Interface with Input Errors US-NYJ-CW01-01 1 New Intent Dashboard
45	US-NYJ-CW01-02		Interface with Input Errors US-NYJ-CW01-02 1
46	VXLAN-MGMT		
47	Wireless_DHCP_Switch		

The following is an example of the summary report. You can export to a CSV file.

View Summary Report - Replicated Intent (28 inte	nts)		×	
Create summary report of all the CSV reports ge Only merge CSV reports generated in: 1 Interfaces with Inp		f 28 intent results filte	ered	
Items: 5 rows 3 columns			Search Q	
Hostname	Interface name	Intput errors		
ISP-PE02	Ethernet1/1	1328	A	
US-BOS-R2	Ethernet0/2	43768		
ISP-PE01	Ethernet1/0	2		
Berlin-R1	Ethernet0/0	469		
Berlin-R1	Ethernet0/3	3013	-	
Export			Close	

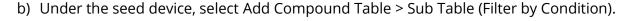
## 12.3.2 Create Report by Pre-Replicated Intent Template

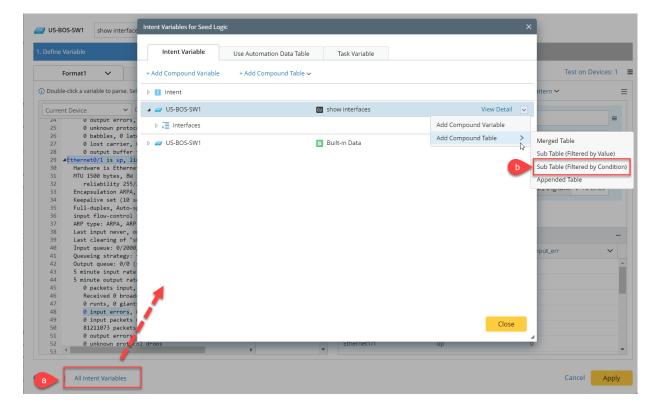
The other method to create the report is to build an ADT via the method **Pre-Replicated Intent Template**. The key concept is the **signature variables**, which can be defined in the **intent replication setting** and exported to the ADT base. For the signature variables to be imported, you have to install and decode the intent in the **Intent Based Automation Center**.

Let us reuse the intent in Section 2.3.1 to create an ADT to display all interfaces with the input errors. Edit the intent, keep the parser and diagnosis (optionally, you can remove the diagnosis to keep the intent clean), and then,

Add a subtable to filter interfaces with input errors.

a) Click the **All Intent Variables** link.





c) In the Add Sub Table window, enter a table name, select the base table (*interfaces*), and define the filtering logic: Only Keep table rows where *intput\_err* > 0. The filtered table is displayed at the lower half of the window.

Add Sub Table (Filte	tering Row by Condition)	×								
Table Name:	Interfaces_with_error									
🔠 Base Table:	interfaces 🗸									
Filtering Logic:	Only Keep $\sim$ table rows if values match the below condition									
	A input_err	~ 🛍								
	B Please Select V									
	$\Box$									
	Boolean Expression: A									
	Calculate 铰									
Base Table (inte	terfaces) New Table (Interfaces_with_error)									
	a show interfaces ■ interfaces									
if_name (string)		~								
🕐 Help	Cance	el OK								

- 4. Define the replication logic.
- 5. Under the Replication Logic tag,
  - a) Enable **Save as Template for** and select the option Device-based Replication.
  - b) Define the intent qualification **via Dynamic Search** and select the search criteria as the device type equal to **Cisco Router** or **Cisco IOS Switch**.
  - c) Define the signature variables in the **Advanced Setting**.

Network Intent (Edit Mode)			×
Interface with Input Errors	Run	with Live Data	Save 🕜 Help 😑
Type description here		20	Intent Map: Select 🗸
I Seed Logic			
Serve as Template for: Device-based Replication	plication 🗌 Enable Neighbor Pair Replication	a	
Intent Qualication: O via Device Groups/Sites: Select     O via Dyname	c Search: Defined 💼 🛛 🕒		Ξ
Define Rules to Replace Macro Variable in Seed Command 0/0	3 Defin	ne Critical Variable	ing C
V 🥔 US-BOS-SW1	6		
⊿ 💷 show interfaces Command Macro Variable: (+ Add)			

- d) Click Add via Table Columns and the subtable you just created, select Interfaces\_with\_error.
- e) In the **Add Signature Variable** window, enter a variable name and edit the display names.

Ill Settings for Intent Template			
Serve as Template for:      Device-based Report of the server of the	eplication O Path-based Replication	Enable Neighbor Pair Replication	
Intent Qualification Macro Variab	le Critical Variable Advan	ce Settings	
Match Target Device with Seed Devices:	6		
O Match 1 Seed Device Only		Select Table Columns	×
Try to Match All Seed Devices		Search	Q
		🖌 🚄 US-BOS-SW1	
Name Rule: () \$nit_name \$device_name		<ul> <li>Interfaces</li> <li>Interfaces</li> </ul>	
Automation Tag for Cloned Intents: + Add 📺			
		🗌 📔 status	
Signature Variables:	⊳	Interfaces_with_error	
d ms + Add via Table Columns -+ Add	ld via Single Variables 💼 💼 📂	✓ ₩ interfaces_with_error	
Signature Variables Name	Туре	🗸 📔 status	
		✓ II input_err	
		Cancel	NK IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
		Cancel	ж
		Cancel C	ж
		Cancel C	Cancel OK
		Cancel	
	Name:	Cancel	
	Name:	Cancel	
	Name: Selected Seed Devices and Tables	Cancel C	
3	Selected Seed Devices and Tables	Display Name of Column:	
	Selected Seed Devices and Tables Seed Device: Table:	Display Name of Column:	Cancel OK
	Selected Seed Devices and Tables Seed Device: Table: US-BOS-SW1 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Display Name of Column:	Error +
3	Selected Seed Devices and Tables Seed Device: Table:	Display Name of Column:	Error +
	Selected Seed Devices and Tables Seed Device: Table: US-BOS-SW1 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Display Name of Column:	Cancel OK
	Selected Seed Devices and Tables Seed Device: Table: US-BOS-SW1 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Display Name of Column:	Cancel OK
3	Selected Seed Devices and Tables Seed Device: Table: US-BOS-SW1 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Display Name of Column:	Cancel OK
	Selected Seed Devices and Tables Seed Device: Table: US-BOS-SW1 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Display Name of Column:	Cancel OK
	Selected Seed Devices and Tables Seed Device: Table: US-BOS-SW1 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Display Name of Column:	Cancel OK
	Selected Seed Devices and Tables Seed Device: Table: US-BOS-SW1 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Display Name of Column:	Cancel OK

- 6. Install and decode the intent in the **Intent Based Automation Center**.
  - a) Open the **Intent Based Automation Center** and click **+Add an Intent Template**. Select the intent you just saved.
  - b) Click **Decode Now** to decode the intents.
  - c) Wait for the system to finish replicating and decoding intents. You can see the number of devices already decoded. Click it to view the execution log.

nstalled Intent Templates	Published Intents	Auto Intent Aut	to Intent Profile NetBra	in Download				
a 59 + Add Intent Temp	late	Filter:	All 🗸	cookbook	🗙 😋 Refresh	∎ In	ntent Template Name: Interface with Input Errors	
Intent Template Name	Location	Intent Decoding	Decoded Device	Auto Intent	Cloned Intents		Basic Cloned Intents	
cookbook								
Spanning Tree Enabl	All Network Intents/GD	Last Decoded at 04:45	31		62		Intent Decoding	b Decode Now
Document BGP Net	All Network Intents/GD	Last Decoded at 02:29	15		15		○ Recurring Decode	
Interface with Input	All Network Intents/GD	Last Decoded at 02:59	58		116	~	Select	$\sim$
							Update Intent Baseline Periodically	
							One-Time Decode	58 Devices decoded
							Occoding Settings	Apply

- Create an ADT from the pre-replicated (installed) intent. Create a new ADT from the Automation Data Table Manager and click the Table Builder,
  - a) Select the Pre-replicated Intent Template as the method.
  - b) Select the intent template you just saved.
  - c) Drag and drop the signature variables to columns.
  - d) Click **Save and Build** to build the ADT.

Automation	i Data Table I	Builder				>
Column	Header:					Reset All
Û	c1	😑 ()	c2	s (} c3	s 🕃 c4 s	
	Device	Interfa	ces - Interface N	. Interfaces - status	Interfaces - Input Error	
Base	+			Column Group (Base):		Select Column 🗸
Descript Select N		ild Base Table:		c1 Device Device	c2     C3     C4       Interfaces - Interf     Interfaces - status       Interface_with_error     Interfaces - status	
Pre-re	plicated Inter	nt Template	a			
Built-in	Fields: Devic Intent Signature Var	۵	î	(Drag and drop colum	I In header from the upper area or available data field from the left here.)	
£ int	terface with	error - s s	C	Filter Row		
		_error - I S		Filter devices of cloned	d intent and member intent by Device Group/Sites: Select $\sim$	
Intent O	utput:		- 1			
	Intent Mes	sage S				
	Intent Status	Code S	~			
🕒 Auto-E	Build No Sc	heduled Updat	e		Cancel Sa d	Save and Build

A sample result is shown as follows. You can create a dashboard from this ADT or export it to a CSV file.

No.	Device		S Interfaces - Interface Name	SInterfaces - status	S Interfaces - Input Error	
41	ITE_EXTEND		GigabitEthernet0/9	up	137	$\checkmark$
42	ITE_EXTEND		GigabitEthernet0/11	up	93	
43	ITE_EXTEND		Port-channel2	up	137	
44	ITE_EXTEND		Port-channel3	up	93	
45	CP-SW3	ß	≡			V
46	UK-LHR-CW04-02					
47	DE-MUC-CW01-01					
48	US-BOS-CW04-02					
49	US-BOS-CR01-01					
50	JP-TYO-CW02-01					
51	US-BOS-CW02-02					
52	JP-TYO-CW01-01					
53	SG-SIN-CW02-01					
54	ISP-PE02		Ethernet1/1	up	1337	
55	ISP-P02					
56	ISP-PE01		Ethernet1/0	up	2	
57	Berlin-R1		Ethernet0/0	up	469	
58	Berlin-R1		Ethernet0/3	up	3021	