



# NetBrain® Integrated Edition 8.0 Release Notes

1. Summary .....	10
2. New Features and Feature Enhancements.....	13
2.1. Data View 2.0.....	13
2.1.1. Main End User Flow .....	14
2.1.2. Define Data View Templates .....	18
2.1.3. Share Data View Templates.....	28
2.1.4. Built-in Data View Templates .....	28
2.1.5. Schedule Data Retrieval for DVT.....	38
2.1.6. Message Golden Baseline Alerts to More Users.....	40
2.2. Static and Map Data View .....	44
2.2.1. Static Data View.....	44
2.2.2. Map Data View .....	45
2.3. Golden Baseline .....	46
2.3.1. Manually Define Golden Baseline.....	47
2.3.2. Auto Calculate Golden Baseline.....	50
2.3.3. View Golden Baseline Analysis Result.....	53
2.4. Visualize Third-Party System Data through Single Pane of Glass .....	56
2.4.1. Define and Apply a SPOG URL.....	57
2.4.2. Pre-Define a Generic Variable for SPOG URL.....	58
2.5. New Multicast Reverse Path Calculation .....	59
2.5.1. Calculate Multicast Reverse Path.....	59
2.5.2. View Path Legend on Map .....	60
2.5.3. Gateway for Multicast Reverse Path .....	61
2.6. Application Assurance Module .....	61
2.6.1. Manage Applications and Paths.....	62
2.6.2. Verify Application Paths in Batch.....	64

2.6.3.	Verify Application Paths via Runbook Automation.....	69
2.7.	Smart CLI.....	71
2.7.1.	Basic Functions of Smart CLI Tool .....	71
2.7.2.	Auto Login to Device.....	73
2.7.3.	Document CLI Results .....	74
2.7.4.	Collaboration Between Multiple Smart CLI Consoles .....	78
2.8.	New Event-driven Automation Framework.....	79
2.8.1.	Configure Third-party System to Send Events to NetBrain via APIs .....	80
2.8.2.	Define Event Templates to Process Events .....	83
2.8.3.	View Results in System Automation Manager .....	85
2.9.	Search Engine Extension.....	86
2.9.1.	Search for Network Context Map .....	87
2.9.2.	Search for One-IP Table .....	87
2.9.3.	Search for A/B Paths across a Specific Hop.....	88
2.9.4.	Search for Event .....	89
2.9.5.	Search for ACI Endpoint Table .....	89
2.9.6.	Other Enhancements .....	89
2.10.	Export Dynamic Map to Word.....	90
2.11.	Document Network Change and Runbook Automation in Word.....	92
2.12.	New Change Analysis .....	93
2.12.1.	Define Analysis Settings.....	95
2.12.2.	Execute Change Analysis Automatically.....	95
2.12.3.	Run Change Analysis Report .....	96
2.12.4.	View Change Details .....	96
2.12.5.	Export Change Analysis Report.....	97
2.13.	Ansible Integration.....	98
2.13.1.	Define Ansible Playbook .....	100
2.13.2.	Execute Ansible Playbook .....	102
2.13.3.	NetBrain Ansible Deployment.....	103
2.14.	Enhanced Supportability for Customer Service Ticketing .....	105

2.14.1.	Auto-Collected Data in Ticketing.....	106
2.14.2.	Ticketing Samples .....	107
2.15.	Map Enhancements.....	110
2.15.1.	Customize Device Icon for Device Type and Vendor Model .....	110
2.15.2.	Define and Customize Topology Link Styles on Map .....	111
2.15.3.	Use Built-in or Customized Visio Template to Export Visio Maps .....	114
2.15.4.	Separately Highlight Link and Port on Map.....	115
2.15.5.	Export Map to Image.....	116
2.15.6.	Auto Unlink Interfaces in a Device Group.....	117
2.15.7.	Narrow Down Auto-link Scope and Count.....	118
2.15.8.	Custom Node Icons .....	119
2.15.9.	Free Text for Network Design/Troubleshooting Annotation .....	119
2.15.10.	Copy, Cut and Paste Stencil Icons and Shapes .....	120
2.15.11.	Map One-IP Table Entries with Drag-and-Drop.....	120
2.15.12.	Enhancements to Embedded Map.....	121
2.16.	Runbook Enhancements.....	123
2.16.1.	Personalize Default Runbook.....	124
2.16.2.	Show Alert Icon for Execution Results .....	124
2.16.3.	Merge Same-Type Action Nodes.....	124
2.16.4.	Move Action Nodes.....	125
2.16.5.	Limit Max. Node Count .....	126
2.16.6.	New Action Nodes .....	126
2.16.7.	Enhanced Compare Node.....	127
2.16.8.	Enhanced CLI Node .....	128
2.16.9.	Keyboard Shortcuts .....	128
2.17.	Site Enhancements .....	129
2.17.1.	Exclude Specified Device Types from Site Assignment.....	129
2.17.2.	Allow Hiding Neighbor Sites from a Site Map .....	131
2.17.3.	Allow User Input for Site Type.....	132
2.17.4.	Move Site Name Underneath Site Icon to Show Full Site Name .....	132

2.18.	Enhancements to Change Management .....	133
2.18.1.	Use Variables in Config-let.....	133
2.18.2.	Schedule Change Management Task via REST APIs .....	135
2.19.	More Feature Enhancements.....	135
2.19.1.	Auto Authenticate User Accounts during Login.....	135
2.19.2.	Allow Users to Run Automation Actions Directly on an Unsaved Map .....	136
2.19.3.	Loosen Map Editing Rights to Enhance Collaboration.....	136
2.19.4.	Enhance Device Group to Group and Map Interfaces.....	137
2.19.5.	Support the Display of Multiple Device Data Dialogues Concurrently.....	139
2.19.6.	Enhanced Device Data Comparison Flow .....	140
2.19.7.	Enhancements to API Triggered Diagnosis .....	141
2.19.8.	Visualize and Search Audit Logs .....	145
2.19.9.	Custom Rules for User Inputs of CLI Commands .....	145
2.19.10.	Enhancements to Service Monitor .....	146
2.19.11.	Enhancements to Event Console .....	150
3.	Platform Enhancements.....	151
3.1.	Enhanced Platform Framework.....	151
3.1.1.	Topology Dependency.....	151
3.1.2.	Driver Enhancements for Virtualization.....	152
3.1.3.	New Multi-CLI Mode Support .....	154
3.1.4.	Build/Retrieve Configuration File for SDN Nodes via TechSpec .....	156
3.1.5.	NCT Framework Enhancements .....	156
3.1.6.	L2 Topology Framework Enhancements .....	158
3.1.7.	Use Plugins to Customize for Data Accuracy .....	159
3.1.8.	Script Manager .....	162
3.2.	Extension Based on Enhanced Platform Framework .....	163
3.2.1.	Technology Support Summary .....	163
3.3.	Path Enhancements .....	164
3.3.1.	Enhancements to Path Gateway.....	164
3.3.2.	Calculate Overlay and Underlay Path based on Topology Dependency.....	168

3.3.3.	QoS Path Support .....	170
3.3.4.	Cisco Wireless L2 Path Support .....	171
3.3.5.	Enhancements to Path Calculation for Unknow End Systems .....	173
3.3.6.	Path Visualization.....	173
3.3.7.	Pin the Path Result Pane.....	175
3.3.8.	Other Path Enhancements .....	175
3.3.9.	Supportability .....	177
3.4.	Fully Extensible MPLS Cloud Framework.....	178
3.4.1.	Add Cloud Type .....	178
3.4.2.	Define Specific Clouds for a Cloud Type .....	180
3.4.3.	Build Cloud Topology and NCT Data for Path Calculation .....	181
3.4.4.	Calculate Path Across a Cloud.....	181
3.5.	Knowledge Cloud .....	182
3.5.1.	Knowledge Cloud Management (NetBrain Internal Use Only) .....	183
3.5.2.	Common Repo Release Management .....	184
3.5.3.	Customized Repo Release Management.....	185
3.5.4.	Update from IE System .....	186
3.6.	Enhancements to Variable Mappings for Multi-Vendor Support.....	187
3.6.1.	Manage Namespace and Variable Mappings.....	188
3.6.2.	View Variable Mappings for DVT.....	189
3.7.	SDN Enhancements.....	190
3.7.1.	ACI Multi-POD Support.....	190
3.7.2.	Calculate Path in ACI Environment.....	192
3.7.3.	Context Map Improvement .....	196
3.7.4.	Enhanced L2 Topology of SDN Nodes .....	196
3.8.	Enhancements to Multi-Vendor Support.....	198
3.8.1.	Enhanced Customization of Login Script and Supported Keys .....	198
3.8.2.	More Drivers.....	200
3.9.	Enhancements to GDR.....	201
3.9.1.	Allow to Predefine Enum Values.....	201

3.9.2.	Allow the Deletion of GDR Property Assigned with Value .....	202
3.10.	Parser Enhancements .....	202
3.11.	Qapp Enhancements .....	204
3.11.1.	Enhancements to Qapp Output.....	204
3.11.2.	Enhancements to Qapp Debugging .....	208
3.11.3.	Usability Enhancements.....	209
3.12.	L3 Topology Support for Interfaces without Masks .....	209
3.13.	L2 Topology Improvements.....	210
3.13.1.	Auto Clean Unknown End Systems .....	210
3.14.	Enhancements to APIs.....	211
3.14.1.	North-bound APIs .....	211
3.14.2.	System APIs.....	212
4.	Admin & MISC .....	214
4.1.	New Subscription License Model with Modularization .....	214
4.1.1.	Subscription License Model.....	214
4.1.2.	Modularized Product.....	218
4.1.3.	License Adjustment to SDN Module.....	219
4.2.	System Architecture Enhancements .....	223
4.2.1.	System Architecture Summary .....	223
4.2.2.	Component Changes and New Components .....	226
4.2.3.	Optimize Installation Process.....	227
4.2.4.	System Switchover Across Multiple Data Centers.....	228
4.3.	Administering NetBrain and Others.....	229
4.3.1.	Domain Health Report .....	229
4.3.2.	Granular Policy for Device Access.....	233
4.3.3.	Detect and Resolve Hostname Duplication.....	235
4.3.4.	Support Internet Cloud .....	238
4.3.5.	Resolve Undiscoverable Devices.....	238
4.3.6.	Restore Backup Maps.....	240
4.3.7.	Auto Clean Execution Logs and More Data.....	241

4.3.8.	Monitor MongoDB Disk Usage with Email Alerts .....	242
4.3.9.	Monitor System Usage and Health.....	243
4.3.10.	Enhancements to System/Tenant Administration .....	244
4.3.11.	Enhancements to Domain Setup/Maintenance Process .....	247
4.3.12.	Enhancements to Benchmark.....	250
4.3.13.	Enhancements to Discovery.....	256
4.3.14.	Enhancements to Duplicate IP and Subnet.....	257
4.3.15.	Collect Usage Log.....	259
4.3.16.	Patch Version Management .....	260
4.4.	Enhancements to System Security .....	260
4.4.1.	New Keystore with Enhanced Hashing and Encryption Algorithms .....	260
4.4.2.	Enforce Strong Password Policy.....	261
4.4.3.	Single Source for Website Base URL .....	262
4.4.4.	Upgrade Third-Party Components to the Latest Versions .....	263
4.4.5.	Allow the Deletion of Built-in Admin Account.....	264
4.4.6.	Start Services with Restricted Privileges .....	264
4.4.7.	Lock Accounts After Too Many Unsuccessfully Attempts of Password Reset .....	264
4.4.8.	Limit Guest User's Privilege .....	265
4.4.9.	Validate Data Access Privilege for Extended Device Scope.....	265
4.4.10.	Minor Enhancements .....	266
4.5.	Performance Enhancements.....	266
5.	Appendix.....	270
5.1.	Built-in Data View Templates in Detail.....	270
5.2.	Built-in Plugins.....	278
5.3.	Application Weight.....	279
5.4.	SDN Context Maps.....	280
5.5.	Technology Support Details .....	287
5.5.1.	VXLAN .....	287
5.5.2.	OTV.....	290
5.5.3.	VPLS .....	293

5.5.4.	HA/Cluster .....	298
5.5.5.	Transparent Device.....	301
5.5.6.	MC-LAG.....	302
5.5.7.	FEX Dual-Homed .....	305
5.5.8.	Port Channel.....	307
5.5.9.	Unified Sub Interface in Topology Calculation.....	308
5.5.10.	End System .....	308
5.5.11.	Wireless.....	313
5.5.12.	Management Route .....	316
5.5.13.	SPB .....	317
5.5.14.	HSRP Improvement .....	318
5.5.15.	Checkpoint Firewall R80.....	319

# 1. Summary

NetBrain Integrated Edition 8.0 (IEv8.0) streamlines end users' workflow with dynamic data view, drill-down action, golden baseline, and smart CLI, and enhances the platform support by building Knowledge Cloud capabilities to manage all software resources and platform plug-in to enable the platform expansion. Further, many new features such as Application Assurance, Change Analysis, and more feature improvements such as Path enhancements are added into IEv8.0.

## Highlights

---

### 1. Streamline end users' workflow

- [Dynamic Data View \(Data View 2.0\)](#): a view of the design and operational data on a dynamic map with the visual alert against the golden baseline and recommended drill-down actions. Each data represents a network design or a troubleshooting scenario.
- [Golden Baseline](#): the reference standard of a normal network state which can be auto-calculated from historical data or manually defined, and serve as a basis or foundation for network management analysis and comparison.
- [Single Pane of Glass \(SPOG\)](#): a dynamic link back to the web page of a 3<sup>rd</sup>-party tool related to the data retrieved from this 3<sup>rd</sup>-party in the dynamic data view, which is usually defined as a drill-down action of the data view.
- [Smart CLI](#): a Telnet/SSH client that can document CLI results in a dynamic map or executable Runbook besides the capability of logging into a network device and executing CLI commands.

### 2. Other New Features

- [New Multicast Reverse Path Calculation](#): calculate a multicast path from a receiver to a source.
- [New Application Assurance Module \(AAM\)](#): an add-on module to manage applications and underlined network paths, and periodically verify the application paths to detect path changes and failures.
- [New Event-driven Automation Framework](#) to proactively integrate with 3<sup>rd</sup>-party systems.
- [Change Analysis](#): analyze the change of network data, such as configurations, so that the change history and difference details of every change record can be reported.
- [Ansible Integration](#): integrate Ansible Playbook as a Runbook node so that a dynamic map can be used as the Playbook input, and the results can be managed through Runbook.
- [Export Dynamic Map to Word](#): create a Microsoft document from a Qmap to include Visio map, device inventory data, configurations, and interface information.
- [Document Network Change and Runbook Automation in Word](#)

- [Customer Service Ticketing](#): directly create a ticket to NetBrain Support Team from a specific feature with the associated data auto-collected and attached.
3. Feature Enhancements, including:
- [12 Map Enhancements](#): customization of device icon according to device type and vendor/model, customization of link styles, export to Visio with a template applied, export map to image...
  - [9 Runbook Enhancements](#)
  - [4 Site Enhancements](#)
  - [2 Enhancements to Change Management](#)
  - [More...](#)
4. Platform Enhancements:
- [Topology and Path Dependency](#): create a model for topology dependency and automatically discover the underlay and overlay paths.
  - [Driver Enhancements for Virtualization](#)
  - [New Multi-CLI Mode Support](#)
  - [NCT Framework Enhancements](#)
  - [L2 Topology Framework Enhancements](#)
  - [Platform Plugin Framework](#): a framework to allow plugins which can be any type of executable scripts to be inserted into different stages of discovery, benchmark, or any type of scheduled task, which enables platform expansion to support multi-vendor and new technologies.
  - [10 Path Enhancements](#): calculate Overlay and Underlay Path based on Topology Dependency, enhancements to Path calculation, Path Gateway, and Path Visualization, etc.
  - [Fully Extensible MPLS Cloud Framework](#): a fully scalable framework to support more cloud deployment scenarios and technologies, such as VPLS, IPv6.
  - [13 New Technology Support](#), such as VXLAN, OTV, VPLS, SPB, Wireless, etc.
  - [Knowledge Cloud](#): a central place to manage all built-in and customer-specific software resources and plugins, such as vendor/model, drivers, Qapps, Runbooks, data view templates, and platform plugins.
  - [4 Enhancements to Variable Mapping for Multi-Vendor Support](#): manage variable mappings through Namespace and Global Variable, etc.
  - [4 SDN Enhancements](#): Cisco ACI Multi-Pod Support, Contract Path, and Logical Node Support.
  - [Enhancements to Multi-vendor Support](#): enhanced customization of login scripts and support of sending keys commands.
  - [3 Enhancements to Qapp Output](#)

- [18 New North-Bound APIs and 2 Enhancements](#)

5. NetBrain System Administration and Others

- [New Subscription License Model with Modularization](#)
- [16 Enhancements to Administration](#)
- [10 Security Enhancements](#)
- [4 Enhancements to System Architecture](#)
- [31 Performance Enhancements](#)

## 2. New Features and Feature Enhancements

### 2.1. Data View 2.0

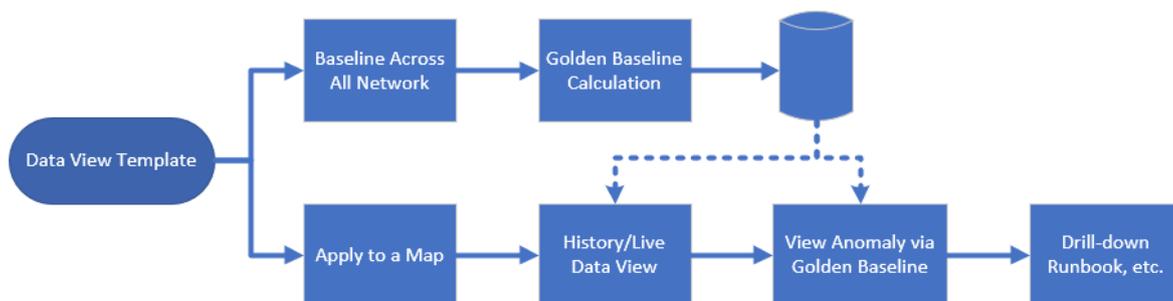
Dynamic Data View, as known as Data View 2.0 (differentiated from Static Data View), streamlines the normal users' workflow by incorporating viewing the network design or operational status, comparing the data against the historical or [Golden Baseline](#) data, and recommending drill-down actions into a central concept, Data View Template. Each data view template can represent a network design or a troubleshooting scenario.

When a user opens or creates a map, the system will apply all data view templates against the devices on this map and all qualified data views will be listed beside the map. When a data view is selected and applied, the historical or live data is displayed to show the network design or operational status. The data is auto-compared against the golden baseline rule, and the data which does not match the rule is highlighted. The user can execute drill-down actions associated with a variable or a data view, such as executing CLI commands, Qapps, Single Pane of Glass, and so on.

To achieve the goal of the end-user flow, an admin or power user needs to define the following:

- Define Data View Template (DVT): besides the variables and the device or interface positions to display these variables, DVT definition also includes drill-down actions, supporting variables, and condition-based notes and highlights.
- Benchmark all variables defined in the DVT. The data will be stored in the data engine and displayed while the end user applies the corresponding DVT to a map.
- Enable Golden Baseline calculation for all or selected variables in the DVT. A golden baseline rule represents a normal network state and can be calculated from a set of historical data.

### Reference Flow for Dynamic Data View

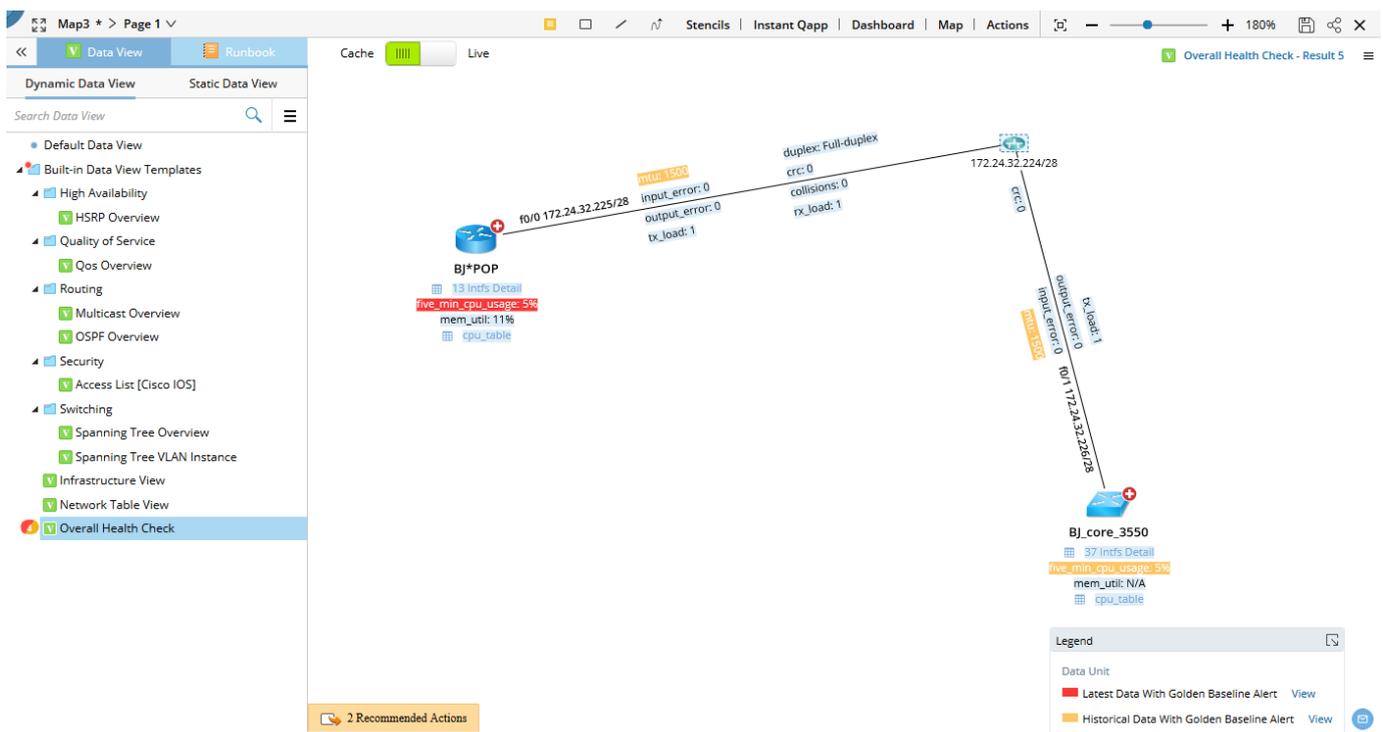


## 2.1.1. Main End User Flow

1. [Apply a Dynamic Data View](#)
2. [View Data and Golden Baseline Alert](#)
3. [View the Detail Data Pane](#)
4. [Execute Drill-Down Actions](#)

### 1. Apply a Dynamic Data View

Dynamic Data View refers to Data View Template (DVT), which organizes device and interface data based on a network technology or a troubleshooting scenario. When users open a map, all DVTs will be checked and qualified by the devices on the map, and only the qualified DVTs can be listed in the Data View pane beside the map. Users can instantly know whether there are any recent or historical Golden Baseline Alerts per Data View Template, and how many of them. Then they can select a proper DVT to apply according to the actual user task, or based on anomalies.



Example of Indicator	Explanation
<p><b>6</b></p> <p>(Latest GB Alerts)</p>	<p>This data view has Golden Baseline Alerts for <b>6</b> device/interface variables, detected in the latest run.</p> <p><b>Note:</b> If the latest run was one month ago, these alerts will be treated as Historical GB Alerts.</p>

Example of Indicator	Explanation
 (Historical GB Alerts)	This data view has Golden Baseline Alerts for <b>1</b> device/interface variable, detected in the last six months.
 (A Combination of Latest and Historical GB Alerts)	<ul style="list-style-type: none"> <li>▪ This data view has Golden Baseline Alerts for <b>X</b> device/interface variables, detected in the latest run.</li> <li>▪ This data view has Golden Baseline Alerts for <b>Y</b> device/interface variables, detected in the last six months.</li> <li>▪ <math>X + Y = 4</math></li> </ul>

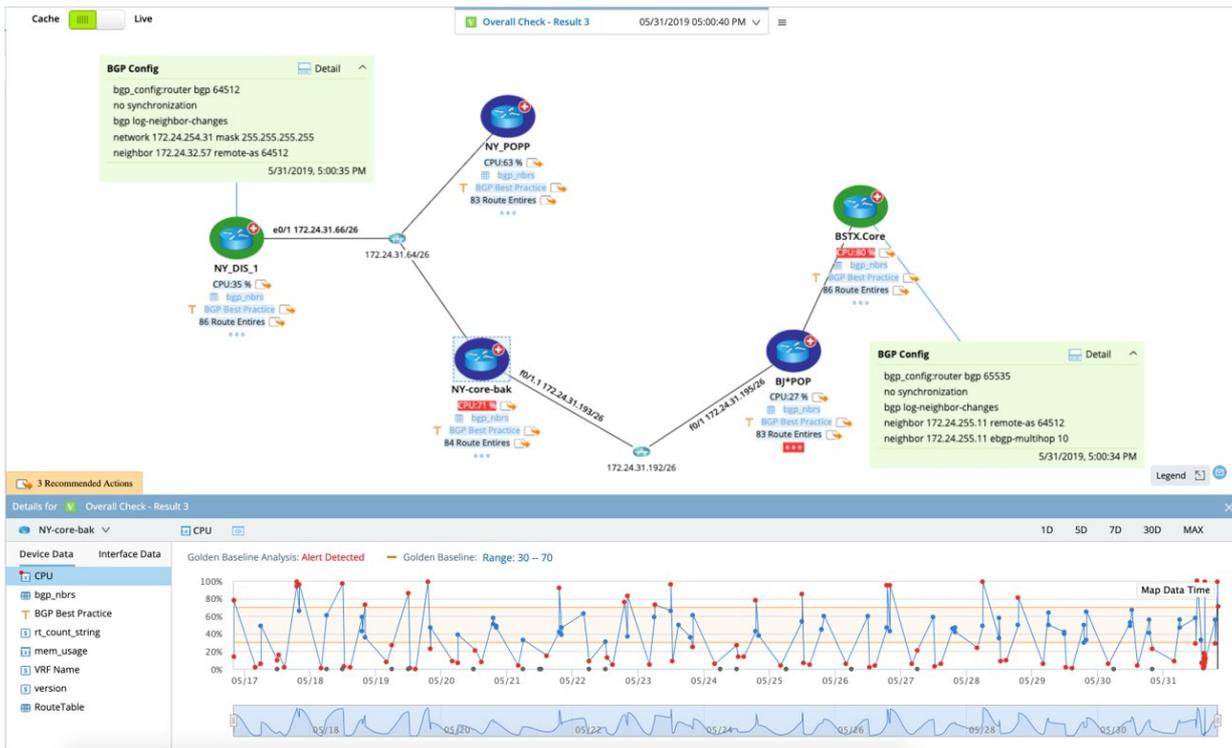
**Note:** DVT supports the qualification definition, such as using configuration contains “router bgp” and “Device Type equals to Cisco Router and Cisco IOS Switch” as filter conditions. Thus, a DVT will only be executed on the qualified devices, which makes applying DVT more efficient.

**Note:** Some of the DVTs allow users to interactively input values to generate a specific data view, such as inputting a VRF name equals to “Blue” and a VLAN ID equals “10”. See [Define Input Variable](#) for more details.

## 2.a View Data and Golden Baseline Alert

When a DVT is applied, the system will retrieve data from the database or live network based on the pre-defined data source in a DVT and display the data on the map. The data can be displayed on the positions under a device icon or along an interface link. It can also be a note or highlight for the devices or interfaces. The golden baseline alert will be shown if the data doesn't match the golden baseline rule. Meanwhile, the pre-

defined drill-down actions for DVT and data unit will also be displayed.



## 2.b View Notes

The device/interface note can be plain texts or contain variables. The display mode is different:

### Note with Alerts

Details for Overall Check - Result 16

NY\_DIS\_1 | bgp\_config

Golden Baseline Analysis: **Alert Detected** Golden Baseline: Equals: router bgp 64512 no synchronization ...

211 items

bgp_config	Time
router bgp 64512 no synchronization	06/03/2019 10:02:24 AM (Map Data Time)
router bgp 64512 no synchronization	06/03/2019 07:58:12 AM
router bgp 64512 no synchronization	06/03/2019 07:57:06 AM
router bgp 64512 no synchronization	06/03/2019 07:56:32 AM

### Note containing variables

Details for Overall Check - Result 16

NY\_DIS\_1 | BGP Config

Last Update Time: 06/03/2019 01:45:14 PM

Device Data	Interface Data
<ul style="list-style-type: none"> <li>CPU</li> <li>\$bgp_nbr_count</li> <li>BGP Best Practice</li> <li>\$rt_count_string</li> <li>mem_usage</li> <li>VRF Name</li> <li>version</li> <li>ospf_nbrs</li> <li>BGP Config</li> <li>bgp_config</li> </ul>	<pre> bgp_config:router bgp 64512 no synchronization bgp log-neighbor-changes network 172.24.254.31 mask 255.255.255.255 neighbor 172.24.32.57 remote-as 64512 no auto-summary         </pre>

### 3. View the Detail Pane for More Data

Click a data unit to open the Data View Detail pane, which provides different styles for different data types. With the Detail pane, users can view both historical data trends, golden baseline and alerts.

#### Detail Pane for Number Data



All the variables in the current data view are listed on the left side of the Detail Pane. The right side of the Detail Pane shows the comparative analysis for historical data points. The Golden Baseline Analysis field shows the current analysis result targeting all data points. Once there is an alert, "Alert Detected" will be shown, and red dots represent the values that don't match the Golden Baseline. If the Golden Baseline Rule is manually changed, the system will re-analyze all historical data according to the new rule to address the anomaly, and refresh the result.

#### Detail Pane for String Data

The screenshot shows the 'Details for Overall Check - Result 3' window. The left sidebar lists variables under 'Device Data' and 'Interface Data', with 'version' selected. The main area displays a table of version data. The table has two columns: 'version' and 'Time'. The 'version' column shows '12.4(3)' for all entries. The 'Time' column shows various timestamps from 05/31/2019. A 'Golden Baseline Analysis: Alert Detected' message is shown at the top, along with 'Golden Baseline: Equals: 15.0'. A '330 items' indicator is also present.

version	Time
12.4(3)	05/31/2019 04:01:47 PM (Map Data Time)
12.4(3)	05/31/2019 04:01:17 PM
12.4(3)	05/31/2019 04:01:16 PM
12.4(3)	05/31/2019 03:01:48 PM
12.4(3)	05/31/2019 03:01:14 PM

## Detail Pane for Table Data

Details for **All Variable Types - Result 1**

PE-3600X-01 | bgp\_nbrs | Table Details | Row Count History

Device Data | Interface Data | 06/03/2019 01:41:05 PM (Map Data Time) | Golden Baseline Analysis: **Alert Detected** | Golden Baseline: Equals: Table...

five\_sec\_cpu\_usage | 4 items | Compare | Type to find... | Export

nbr_id	remote_as	nbr_type	version	remote_rid	state	up_time	last_read	last_write	holdtime
10.88.250.9	64553	external link	4	10.88.255.51	Established	2y43w	00:00:22	00:00:37	180
10.88.250.27	64556	external link	4	10.88.7.4	Established	3y43w	00:00:12	00:00:00	90
10.88.255.5	64550	internal link	4	10.88.255.5	Established	2y10w	00:00:17	00:00:19	180
10.88.255.41	64554	external link	4	10.88.255.41	Established	00:11:17	00:00:06	00:00:48	180

## 4. Execute Drill-Down Actions

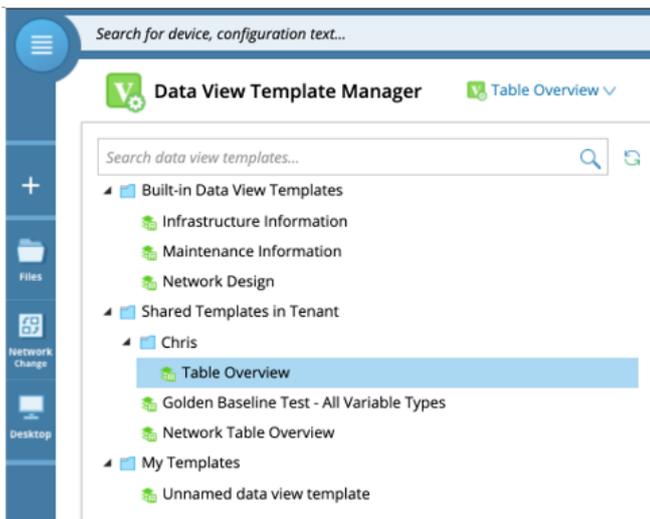
[Drill-down Actions](#) refer to a set of pre-defined actions to analyze and troubleshoot a specific network issue. A drill-down action can belong to the template level or device level.

- **DVT Level** – the action is executed on all devices on a map.
- **Device Level** – the action is only executed on one device.

The selected drill-down actions will be added to a Runbook for users to execute later.

### 2.1.2. Define Data View Templates

Data View Templates are defined, modified, and deleted in the Data View Template Manager.



- [Define a Qualification](#)
- [Define a Variable on a Device/Interface Position](#)
- [Define a Drill-Down Action](#)

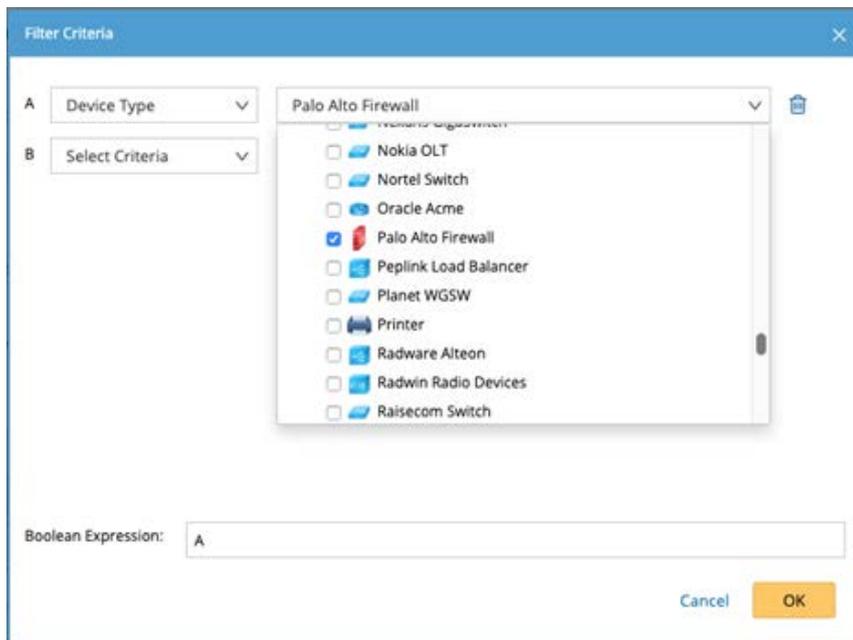
- [Define a Condition to Display the Drill-Down Action](#)
- [Define Script-Based Action Input and Device Input for the Drill-Down Action](#)
- [Define a Note](#)
- [Define a Highlight](#)
- [Define a Compound Variable](#)
- [Define an Input Variable](#)

See also: [Built-in Data View Templates](#).

## Define a Qualification

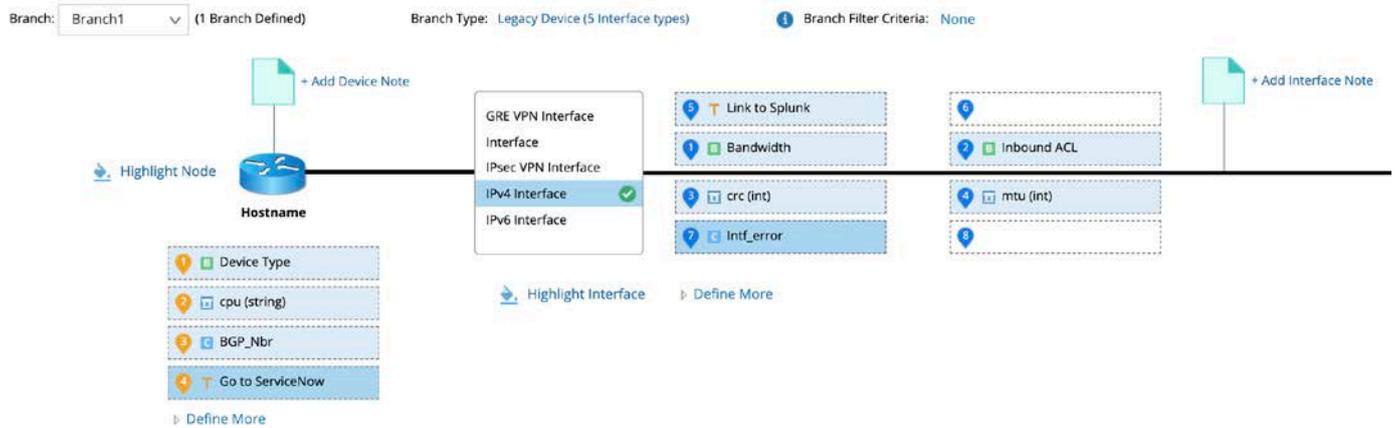
The qualification for a DVT can be used to 1) qualify which devices this DVT can apply to; 2) show different data views for different devices by setting multiple branches; each branch has its conditions.

When you expand the dynamic data view pane beside the map, only the applicable DVTs will be displayed. In other words, the devices on the current map have been prequalified whether they meet the qualifications of the DVTs. For example, to define a DVT for both Palo Alto firewalls and BGP devices, you can create two branches with different conditions: Device Type equal to Palo Alto Firewall **and** Configuration contains router bgp.



## Define a Variable on a Device or Interface Position

When defining a DVT, users can select data to add to 4 or more positions under a device icon and 8 or more positions along an interface link.



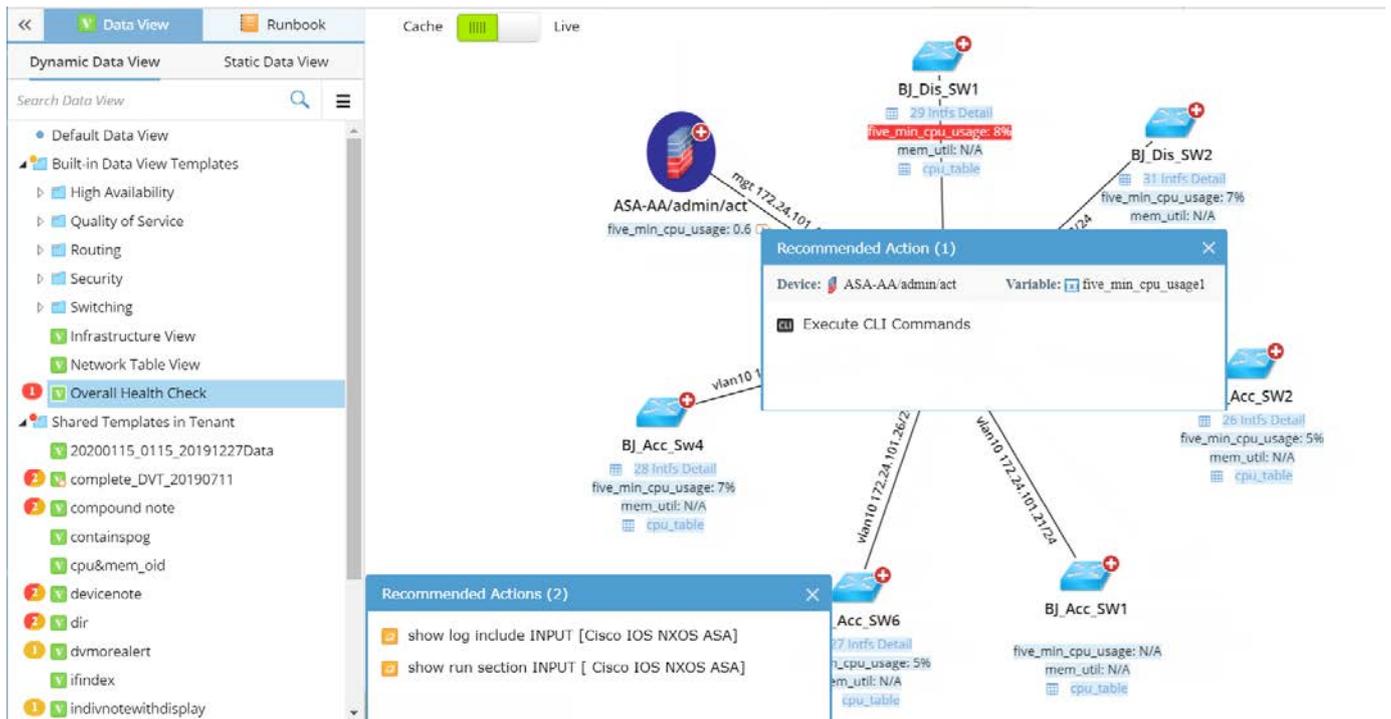
The data can be selected from:

- Built-in Data (GDR Property)
- CLI/SNMP/API/Configuration Parser Variable
- [Compound Variable](#)
- [Input Variable](#)
- Text

## Define Drill-Down Actions

A power user can recommend one or more actions that represent the best practice or in-depth analysis for a data unit or the whole data view. When end users apply a DVT to a map, the drill-down actions will prompt as

recommendations to assist end users for further troubleshooting or analysis.



## 14 Types of Available Drill-Down Actions

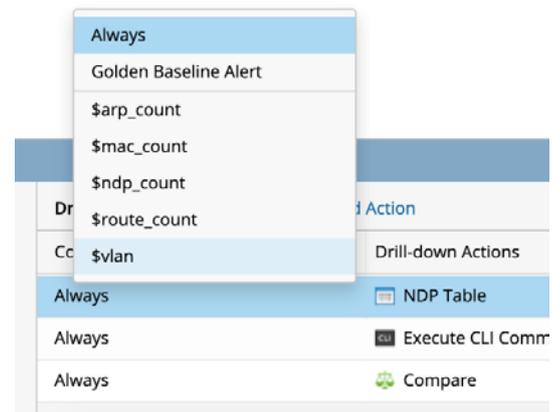
Action	Description	DVT Level Supported	Device/Interface Level Supported
Data View Template	Apply a DVT to a map	✓	✓
Qapp	Run a Qapp	✓	✓
Gapp	Run a Gapp	✓	✓
Overall Health Monitor	Run the "Overall Health Monitor" Qapp	✓	✓
Execute CLI Commands	Execute CLI commands	✓	✓
View Data	Visualize the built-in data, such as configurations, route tables, etc.		✓
Ping	Perform the Ping action		✓
Traceroute	Perform the Traceroute command		✓
Path	Calculate a Path between two end points		✓

Action	Description	DVT Level Supported	Device/Interface Level Supported
Compare	Compare the built-in data or CLI command results	√	√
<a href="#">SPOG URL</a>	Open the designated SPOG URL of the 3 <sup>rd</sup> -party system		√
URL	Open the predefined URL in Web Browser	√	√
Runbook Template	Add the nodes of a Runbook Template to the current Runbook	√	√
Verify Application	Run a <a href="#">Verify Application</a> node	√	√

## Define Conditions to Display the Drill-Down Action

The following conditions can be used to define whether a drill-down action can be displayed or not.

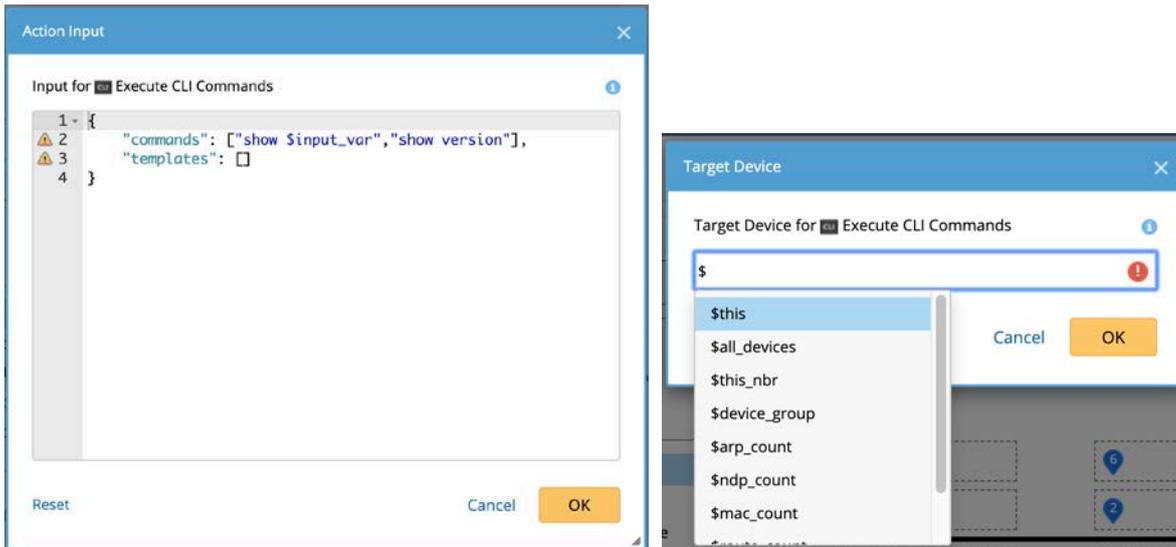
- **Always** – this drill-down action will display unconditionally.
- **Golden Baseline Alert** – this drill-down action will display when the data doesn't match the Golden Baseline.
- **By Supporting Variable** – this drill-down action will display only when the variable from the supporting variable list is not null (for string type), equals to **true** (for Boolean type), or not equal to **0** (for number type).



## Define Script-Based Action Input and Device Input for the Drill-Down Action

To customize for specific use scenarios, power users can define input parameters and target devices for each drill-down action via scripts. For example, to define a drill-down action for interface CRC errors, power users can: 1) run a Qapp to monitor with a predefined alert threshold; 2) run a Qapp to check interface

speed/duplex mismatch for a device and its neighbor devices.



See [Online Help](#) for more reference.

## Define Notes

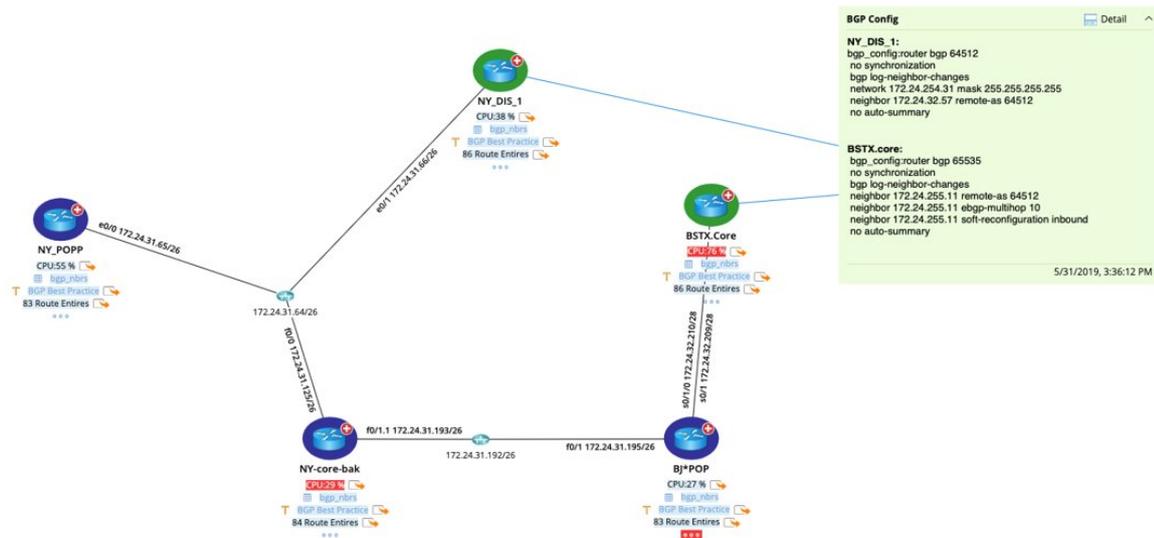
Besides the plain text, the built-in data (GDR properties) and parser variables can be added as a device or interface note. Further, users can define the condition to display the note.

There are three types of notes:

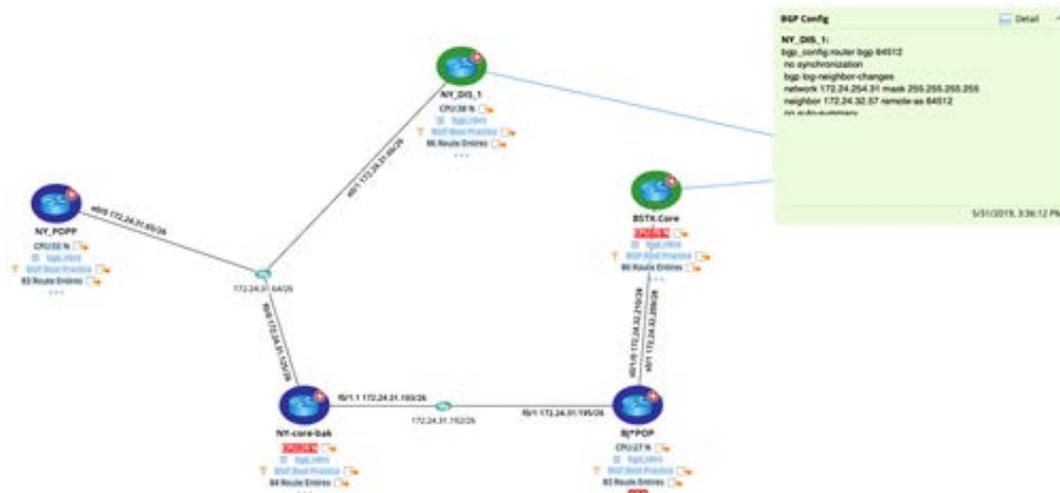
- **Individual Device Note** — a note linked to only one device.



- **Compound Note (Append)** — a note linked to multiple devices, and its content is merged with the notes of these devices. Each note uses the hostname of the device as its prefix.



- **Compound Note (Overwrite)** — a note linked to multiple devices, but its content is the latest note for one device. The previous note will be overwritten by a new one.



## Define Highlights

Like a drill-down action, a condition can be defined for whether to highlight a variable, an interface or a device. Supporting variables can be used to define the condition for a highlight. For interface highlight, the line shape and width can be customized besides the color.

## Define Compound Variables

If the existing built-in data (GDR property) or parser variables cannot satisfy a complex semantic, a Compound Variable can be defined from a set of GDR or parser variables.

For example, the BGP neighbor table can be added to a device position as a variable. As shown in the following figure **Before**, the variable is added traditionally: each device position displays the same content **BGP Nbr**, and there is no valuable information or details. Alternatively, as shown in the figure **After**, a compound variable is displayed with the number of rows, which makes the data view more useful.



**Note:** Both the CPU/Memory utilization and Interface traffic utilization often require a transformation of percentages, for example,  $\text{value}/\text{total} * 100\%$ .

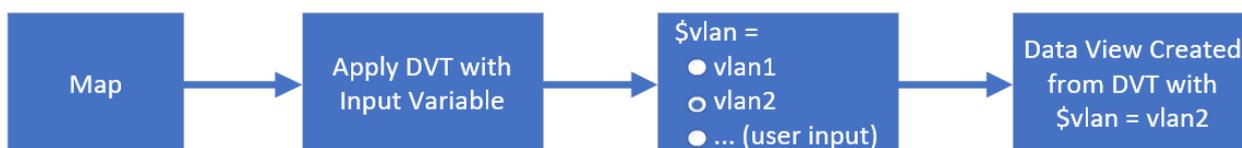
Compound Variable supports three types: string, number, and bool. The supported formula syntax may differ depending on variable types. Refer to [Online Help](#) for more details.

## Define Input Variables

A lot of network technologies and configurations contain more than one instance. I Ev8.0 introduces the "Input Variable" feature for power users to define multiple value options for a variable in the data view template, so that end users can select a proper one or even manually input one when applying the data view template.

For example, each VLAN is corresponding to a Spanning Tree, when users apply a data view template that contains the "VLAN ID" variable to a map, the generated data view can be customized by specifying different values for variables.

### Reference Flow of DVT with Input Variables

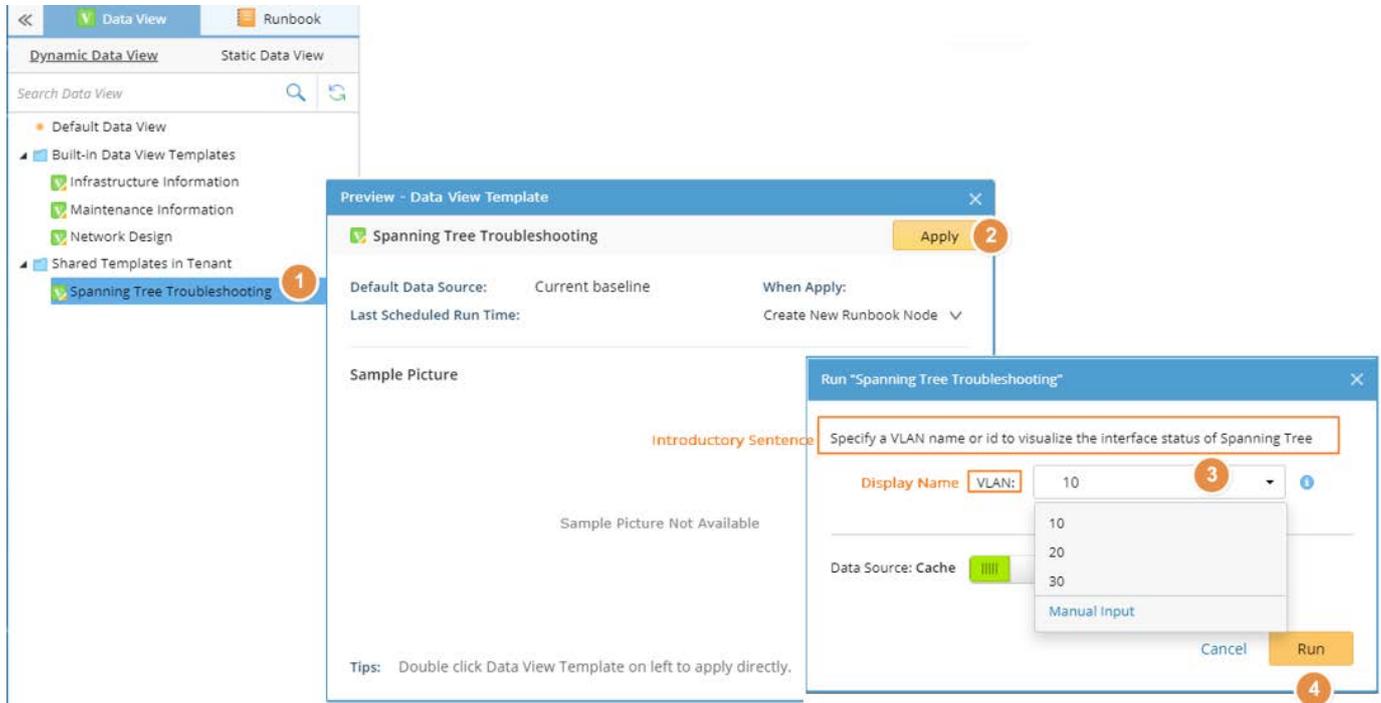


### More Use Cases:

- Manually input a value for multicasting source/group pair to generate an expected data view for a Multicasting Distribution Tree.
- Use a device property from GDR as an input variable. For example, use the GDR property of VLAN as the input value to generate an expected data view for a Spanning Tree.

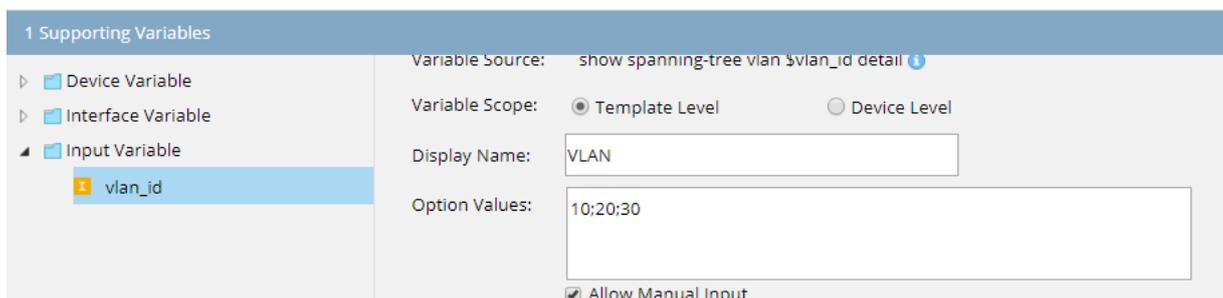
## Apply Data View Template with Input Value to Create Data View

When a user applies a data view template with an input variable, an input dialog will pop up and the user can either select or enter a value. Take “VLAN ID” for example:

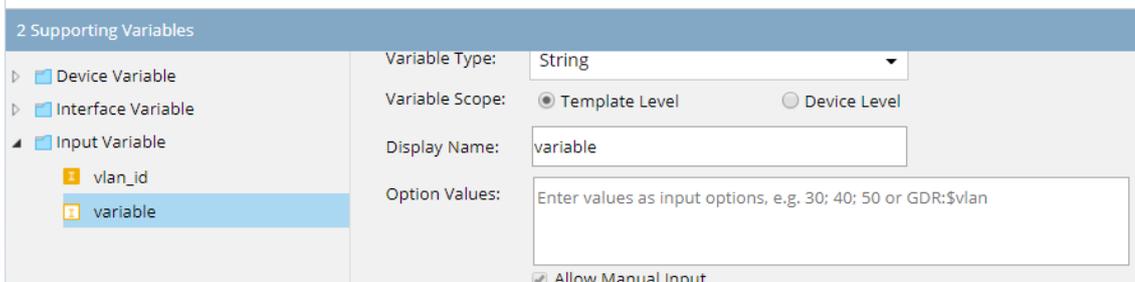
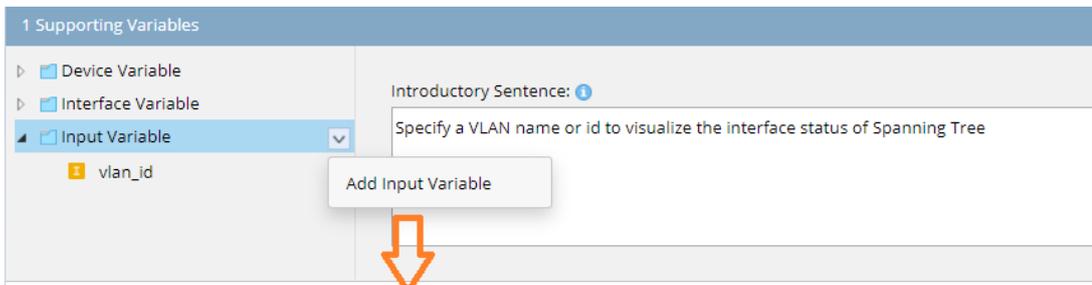


## Define Data View Template with Input Variable

If a CLI command itself includes a variable, such as “show spanning-tree vlan \$vlan\_id detail”, then the variable of this CLI command, **\$vlan\_id** in this example, will be auto-added as an input variable.



You can also manually define an input variable as follows.



For more details about the configurable properties of an Input Variable, refer to [Online Help](#).

### Use Data View Template with Input Variable in Triggered Automation

Data view template with input variables can be added to an API-triggered diagnosis task through a Runbook Template, and the first option value of the input variable will be treated as the default value, which cannot be modified in the task definition. However, other values can be defined manually through scripts in the third-party API server and passed to the NetBrain system for execution.

Variable Input \* Required

Type	Variable Name	Default Value	
▶ API Service Stub			
▶ Map Creation			
◀ My Runbook Templates/Tri...			
☑ ◀ <span style="color: green;">V</span> Highlight STP for Specifi...			Settings
	* \$device		
	* \$data_source	Pull Live data once	
	* VLAN	10	

**Note:** If no option value is defined in the data view template for the input variable and no value is defined in the third-party script, then the data view template node will not be executed during the execution of API-triggered task.

### 2.1.3. Share Data View Templates

There are two ways to share DVTs:

- **Publish DVT Resources Through Knowledge Cloud**

[NetBrain Knowledge Cloud](#) is a new primary media to deliver resources to customers. Data View Template (DVT) is one of the resources, and all associated files referenced in DVT will be shared with customers.

- **Export and Import DVTs**

Exporting/Importing is a standard method to share a DVT file with more users within an enterprise/organization. The NetBrain Service Team usually delivers IEV7.x DVTs to customers in this way.

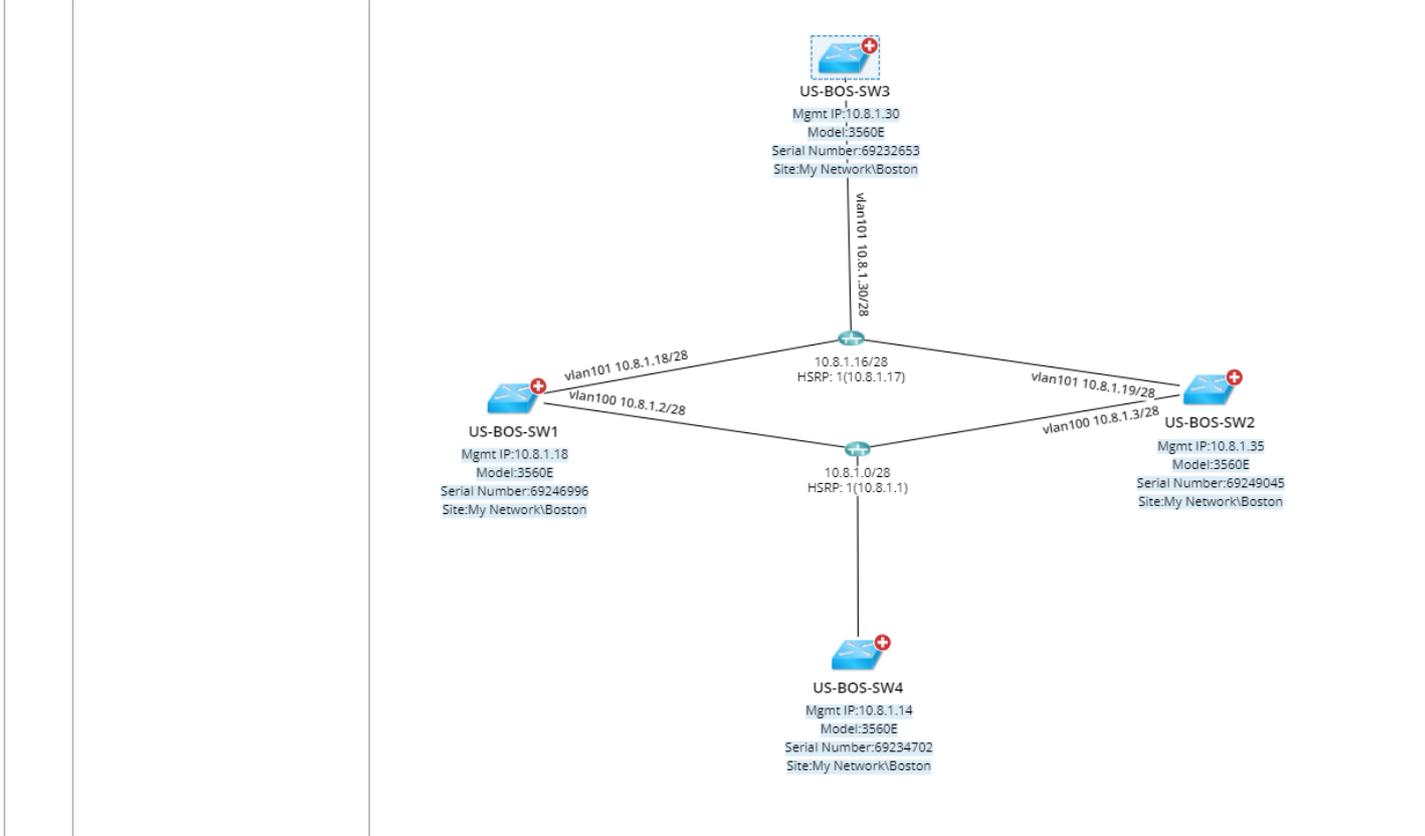
**Limitation:** All Qapp/Gapp/Runbook Template/DVT files referenced as the drill-down actions of a DVT cannot be exported. As a result, it is very likely that the imported DVTs cannot address the corresponding action to execute.

### 2.1.4. Built-in Data View Templates

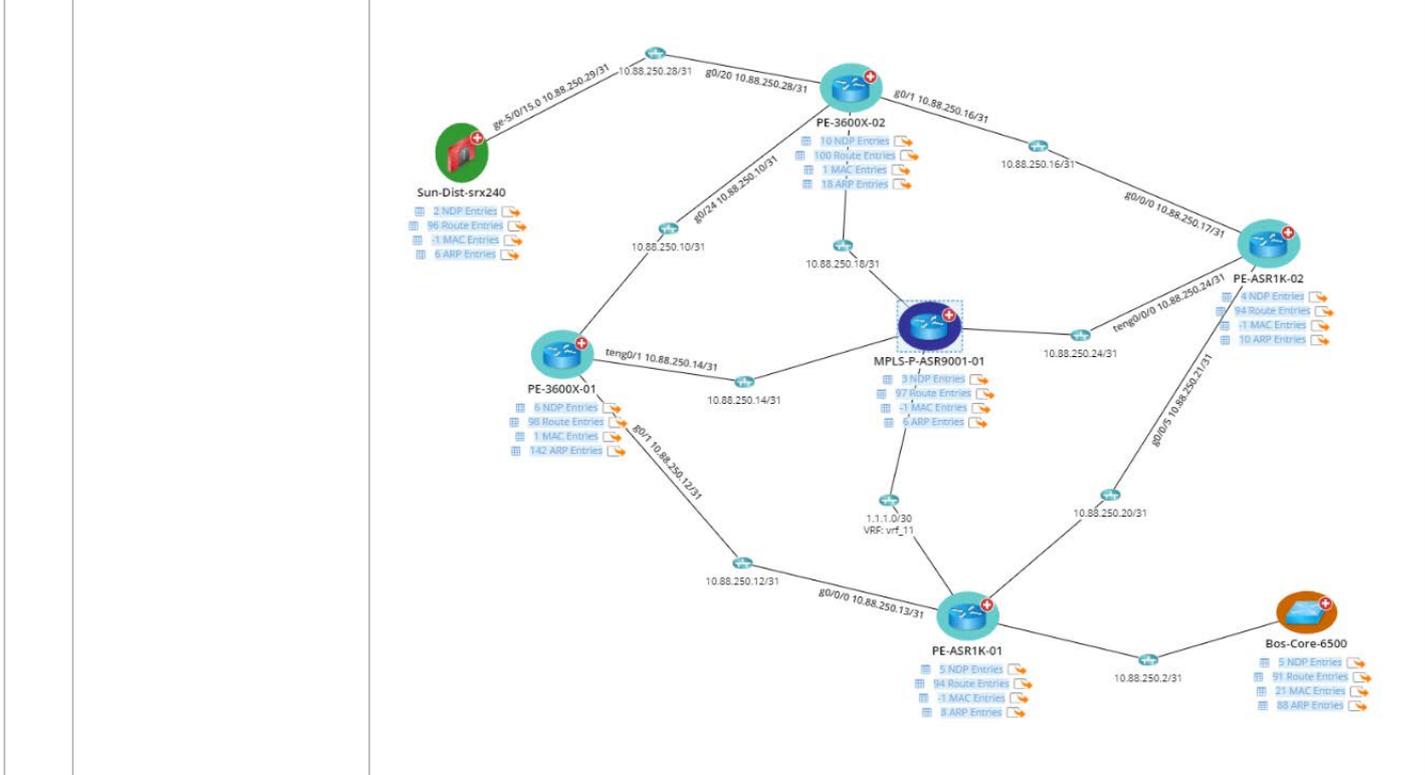
IEV8.0 provides 20 built-in data view templates for various purposes and scenarios, including routing and switching, security, quality of service, high availability and so on. To know more details about supporting variables and drill-down actions defined in each template, refer to the [Appendix](#).

No.	Built-in DVT Name	Description and Sample Image
1	Infrastructure View	Infrastructure maintenance information:

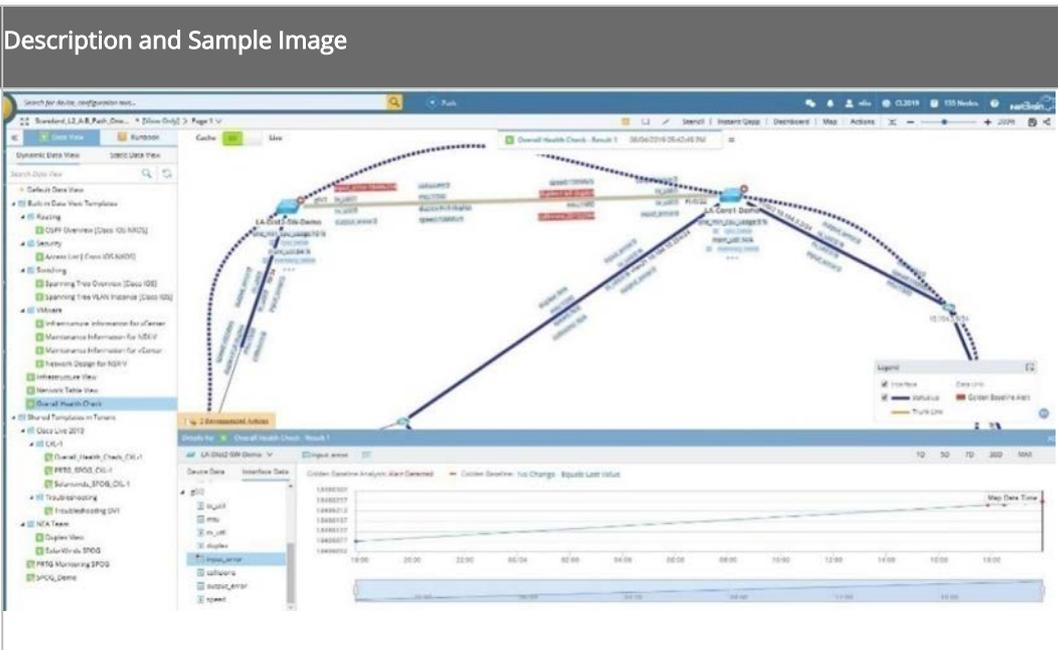
No.	Built-in DVT Name	Description and Sample Image
-----	-------------------	------------------------------

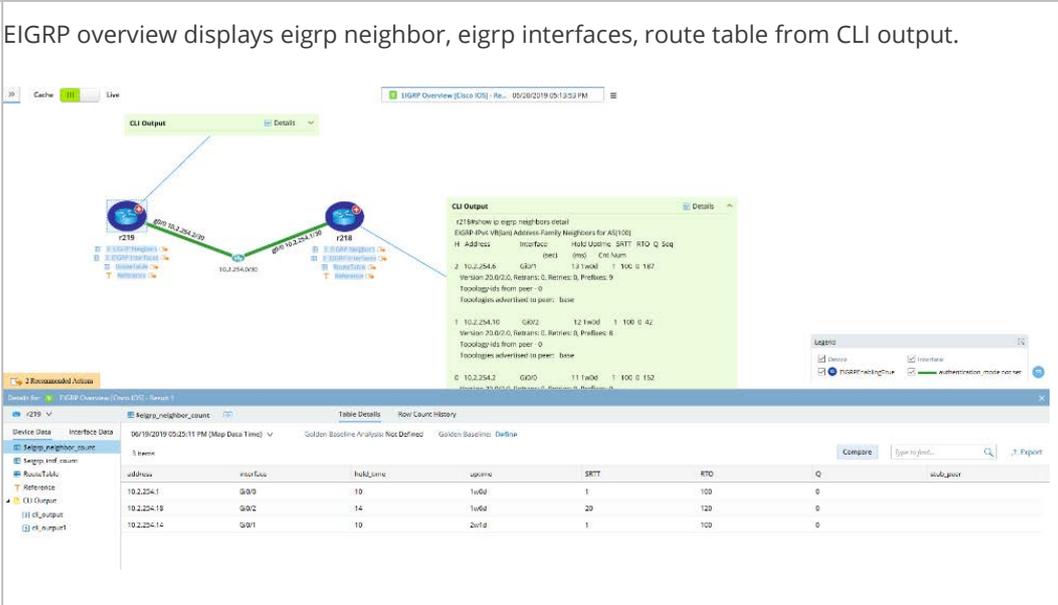


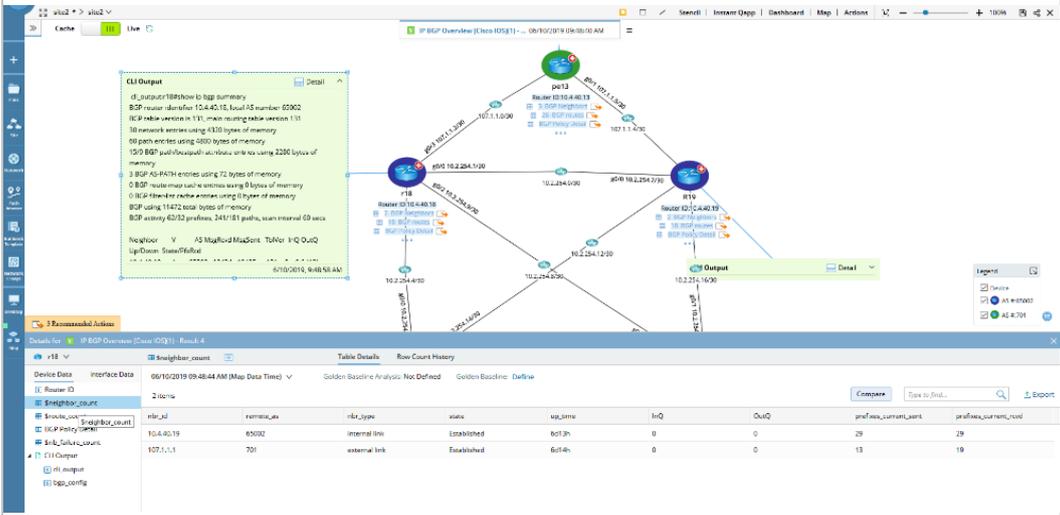
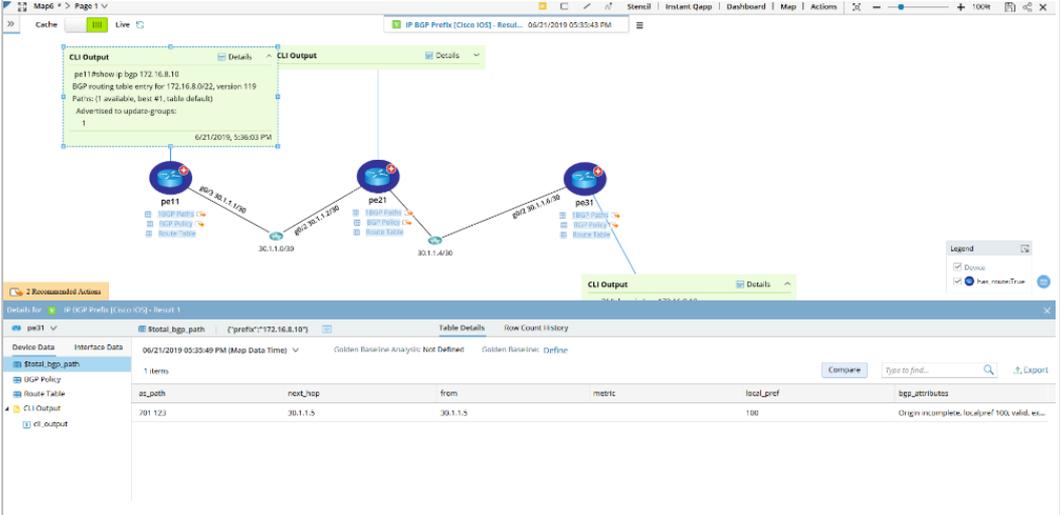
2	Network Table View	NetBrain Built-in Network Table Overview:
---	--------------------	---

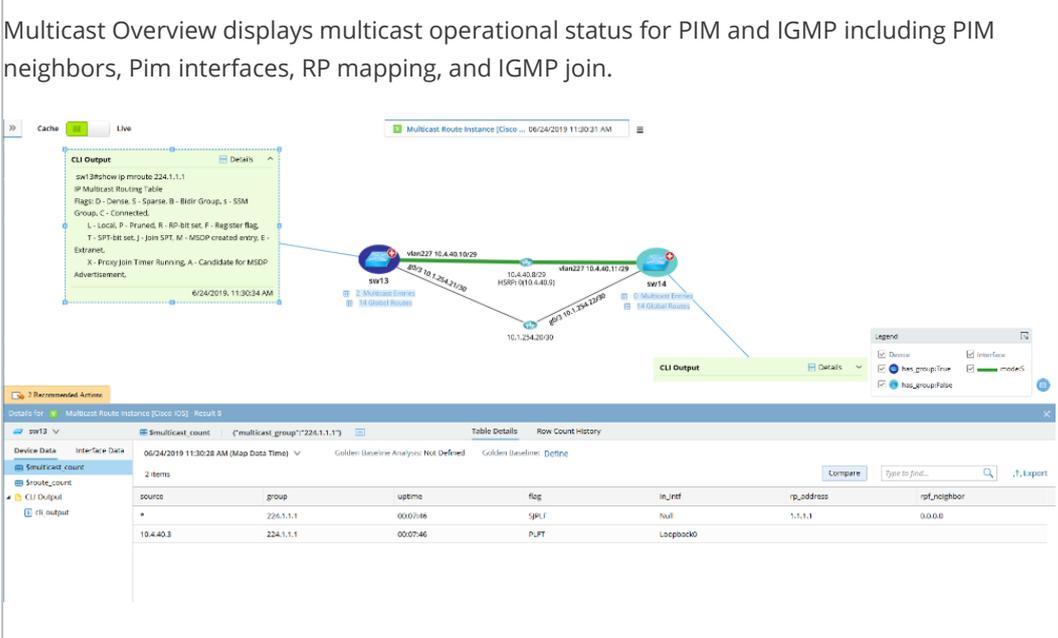


3	Overall Health Check	Overall Health information from CLI:
---	----------------------	--------------------------------------

No.	Built-in DVT Name	<p>Description and Sample Image</p> 
-----	-------------------	--

4	EIGRP Overview	<p>EIGRP overview displays eigrp neighbor, eigrp interfaces, route table from CLI output.</p>  <table border="1" data-bbox="558 1187 1500 1321"> <thead> <tr> <th>address</th> <th>interface</th> <th>nbrName</th> <th>uptime</th> <th>SRTT</th> <th>RTO</th> <th>Q</th> <th>adjPeer</th> </tr> </thead> <tbody> <tr> <td>10.2.224.1</td> <td>G0/0</td> <td>10</td> <td>1w0d</td> <td>1</td> <td>100</td> <td>0</td> <td></td> </tr> <tr> <td>10.2.224-13</td> <td>G0/2</td> <td>14</td> <td>1w0d</td> <td>20</td> <td>120</td> <td>0</td> <td></td> </tr> <tr> <td>10.2.224-14</td> <td>G0/1</td> <td>10</td> <td>2w0d</td> <td>1</td> <td>100</td> <td>0</td> <td></td> </tr> </tbody> </table>	address	interface	nbrName	uptime	SRTT	RTO	Q	adjPeer	10.2.224.1	G0/0	10	1w0d	1	100	0		10.2.224-13	G0/2	14	1w0d	20	120	0		10.2.224-14	G0/1	10	2w0d	1	100	0	
address	interface	nbrName	uptime	SRTT	RTO	Q	adjPeer																											
10.2.224.1	G0/0	10	1w0d	1	100	0																												
10.2.224-13	G0/2	14	1w0d	20	120	0																												
10.2.224-14	G0/1	10	2w0d	1	100	0																												

No.	Built-in DVT Name	Description and Sample Image																											
5	IP BGP Overview	<p>BGP IPv4 Unicast Overview displays BGP operational status including BGP neighbors, BGP policy filter, Route Tables, Rib failure from CLI output.</p>  <p>The screenshot shows the 'IP BGP Overview' interface for router r18. It features a network diagram with three routers (r18, r19, r20) and their interconnections. A 'CLI Output' window is open, displaying the following text:</p> <pre> r18#show ip bgp summary BGP router identifier 10.4.42.15, local AS number 65002 BGP table version is 111, main routing table version 111 38 network entries using 4133 bytes of memory 69 path entries using 3020 bytes of memory 159 BGP path/attribute entries using 2280 bytes of memory 3 BGP AS PATH entries using 72 bytes of memory 0 BGP route-map cache entries using 0 bytes of memory 0 BGP filter-ter cache entries using 0 bytes of memory BGP using 1442 total bytes of memory BGP activity 62/32 prefixes, 241/81 paths, scan interval 60 secs  Neighbor  Y  N  InQ/OutQ  TtlNo  I/O Qx/Qz Up/Down: State/Failed ... </pre> <p>Below the CLI output, a table shows BGP neighbor details for r18:</p> <table border="1" data-bbox="478 739 1519 851"> <thead> <tr> <th>Neighbor ID</th> <th>Remote AS</th> <th>Neighbor Type</th> <th>Status</th> <th>Up Time</th> <th>InQ</th> <th>OutQ</th> <th>Prefixes Current/Max</th> <th>Prefixes Current/Used</th> </tr> </thead> <tbody> <tr> <td>10.4.40.19</td> <td>65002</td> <td>Internal link</td> <td>Established</td> <td>0ct/3h</td> <td>0</td> <td>0</td> <td>29 / 29</td> <td>29</td> </tr> <tr> <td>107.1.1.1</td> <td>701</td> <td>external link</td> <td>Established</td> <td>0ct/4h</td> <td>0</td> <td>0</td> <td>13 / 19</td> <td>19</td> </tr> </tbody> </table>	Neighbor ID	Remote AS	Neighbor Type	Status	Up Time	InQ	OutQ	Prefixes Current/Max	Prefixes Current/Used	10.4.40.19	65002	Internal link	Established	0ct/3h	0	0	29 / 29	29	107.1.1.1	701	external link	Established	0ct/4h	0	0	13 / 19	19
Neighbor ID	Remote AS	Neighbor Type	Status	Up Time	InQ	OutQ	Prefixes Current/Max	Prefixes Current/Used																					
10.4.40.19	65002	Internal link	Established	0ct/3h	0	0	29 / 29	29																					
107.1.1.1	701	external link	Established	0ct/4h	0	0	13 / 19	19																					
6	IP BGP Prefix Instance	<p>IPv4 BGP data view with prefix input displays BGP attributes for BGP paths, highlight prefix existing in BGP table or not.</p>  <p>The screenshot shows the 'IP BGP Prefix Instance' interface for prefix 172.16.8.10. It features a network diagram with three routers (r11, r21, r31) and their interconnections. A 'CLI Output' window is open, displaying the following text:</p> <pre> r11#show ip bgp 172.16.8.10 BGP routing table entry for 172.16.8.0/22, version 119 Paths: (1 available, best #1, table default)   Advertised to update-groups:     1 ... </pre> <p>Below the CLI output, a table shows BGP path details for r11:</p> <table border="1" data-bbox="478 1411 1519 1523"> <thead> <tr> <th>AS_Path</th> <th>Next_Hop</th> <th>From</th> <th>Metric</th> <th>Local_Pref</th> <th>BGP_Attributes</th> </tr> </thead> <tbody> <tr> <td>701 123</td> <td>30.1.1.5</td> <td>30.1.1.5</td> <td></td> <td>100</td> <td>Origin incomplete, local pref 100, valid, est...</td> </tr> </tbody> </table>	AS_Path	Next_Hop	From	Metric	Local_Pref	BGP_Attributes	701 123	30.1.1.5	30.1.1.5		100	Origin incomplete, local pref 100, valid, est...															
AS_Path	Next_Hop	From	Metric	Local_Pref	BGP_Attributes																								
701 123	30.1.1.5	30.1.1.5		100	Origin incomplete, local pref 100, valid, est...																								

No.	Built-in DVT Name	Description and Sample Image																												
7	ISIS Overview	<p>ISIS Overview displays ISIS operations including ISIS neighbors, ISIS interfaces, LSP databases, ISIS processes from CLI output.</p>  <p>The screenshot shows the ISIS Overview interface. At the top, there is a network diagram with four routers: r316 (10.3.254.8/30), r317 (10.3.254.6/30), r318 (10.3.254.4/30), and r321 (10.3.254.2/30). A 'Recommended Actions' dialog box is open, showing 'Execute CLI Commands' for device r318 with variable \$this_neighbo... Below the diagram is a table titled 'Neighbor count' for device r318.</p> <table border="1"> <thead> <tr> <th>system_id</th> <th>type</th> <th>interface</th> <th>ip_address</th> <th>state</th> <th>holdtime</th> <th>circuit_id</th> </tr> </thead> <tbody> <tr> <td>r317</td> <td>L2</td> <td>Gib0</td> <td>10.3.254.2</td> <td>UP</td> <td>25</td> <td>00</td> </tr> <tr> <td>r318</td> <td>L2</td> <td>Gib01</td> <td>10.3.254.6</td> <td>UP</td> <td>28</td> <td>00</td> </tr> <tr> <td>r321</td> <td>L2</td> <td>Gib02</td> <td>10.3.254.10</td> <td>UP</td> <td>25</td> <td>00</td> </tr> </tbody> </table>	system_id	type	interface	ip_address	state	holdtime	circuit_id	r317	L2	Gib0	10.3.254.2	UP	25	00	r318	L2	Gib01	10.3.254.6	UP	28	00	r321	L2	Gib02	10.3.254.10	UP	25	00
system_id	type	interface	ip_address	state	holdtime	circuit_id																								
r317	L2	Gib0	10.3.254.2	UP	25	00																								
r318	L2	Gib01	10.3.254.6	UP	28	00																								
r321	L2	Gib02	10.3.254.10	UP	25	00																								
8	Multicast Overview	<p>Multicast Overview displays multicast operational status for PIM and IGMP including PIM neighbors, Pim interfaces, RP mapping, and IGMP join.</p>  <p>The screenshot shows the Multicast Overview interface. At the top, there is a network diagram with two switches: sw13 (10.4.40.11/29) and sw14 (10.4.40.11/29). A 'Recommended Actions' dialog box is open, showing 'Execute CLI Commands' for device sw13 with variable \$this_group... Below the diagram is a table titled 'Smulticast count' for device sw13.</p> <table border="1"> <thead> <tr> <th>source</th> <th>group</th> <th>uptime</th> <th>flag</th> <th>rp_ref</th> <th>rp_address</th> <th>rp_neighbor</th> </tr> </thead> <tbody> <tr> <td>*</td> <td>224.1.1.1</td> <td>00:07:46</td> <td>SPLT</td> <td>null</td> <td>1.1.1.1</td> <td>0.0.0.0</td> </tr> <tr> <td>10.4.40.3</td> <td>224.1.1.1</td> <td>00:07:46</td> <td>P,PT</td> <td>Loopback0</td> <td></td> <td></td> </tr> </tbody> </table>	source	group	uptime	flag	rp_ref	rp_address	rp_neighbor	*	224.1.1.1	00:07:46	SPLT	null	1.1.1.1	0.0.0.0	10.4.40.3	224.1.1.1	00:07:46	P,PT	Loopback0									
source	group	uptime	flag	rp_ref	rp_address	rp_neighbor																								
*	224.1.1.1	00:07:46	SPLT	null	1.1.1.1	0.0.0.0																								
10.4.40.3	224.1.1.1	00:07:46	P,PT	Loopback0																										

No.	Built-in DVT Name	Description and Sample Image
-----	-------------------	------------------------------

9	OSPF Overview	<p>OSPF Neighbor; OSPF Interface; OSPF Process.</p> <p><b>OSPF CLI Output</b></p> <pre> cl_output:CA-TOR-R1#show ip ospf neighbor Neighbor ID Pri State DeadTime Address Interface 10.8.3.9 1 FULL/BDR 00:00:30 10.8.3.6 Ethernet0/1 10.8.3.10 1 FULL/BDR 00:00:39 10.8.3.2 Ethernet0/0 10.10.10.10 0 FULL/- 00:01:40 10.96.1.1 Tunnel0 ***** cl_output:CA-TOR-R1#show ip ospf interface brief Interface PID Area IP Address/Mask Cost State Nbrs FIC Lo0 1 51 10.8.2.0/24 1 L0CP 0/0 Et0/0 1 51 10.8.3.5/30 10 DR 1/1 Et0/1 1 51 10.8.3.1/30 10 DR 1/1 Tu0 1 51 10.96.1.51/24 10 P2MP 1/1 </pre> <p><b>Recommended Actions (4)</b></p> <ul style="list-style-type: none"> <li>Compare</li> <li>show log include ZINPUT [Cisco IOS NXOS ASA]</li> <li>show run section INPUT [Cisco IOS NXOS ASA]</li> <li>Troubleshooting OSPF</li> </ul> <p><b>Details for CA-TOR-R1</b></p> <table border="1"> <thead> <tr> <th>nbr_id</th> <th>priority</th> <th>state</th> <th>dead</th> <th>nbr_addr</th> <th>intf</th> </tr> </thead> <tbody> <tr> <td>10.8.3.9</td> <td>1</td> <td>FULL/BDR</td> <td>00:00:35</td> <td>10.8.3.6</td> <td>Ethernet0/1</td> </tr> <tr> <td>10.8.3.10</td> <td>1</td> <td>FULL/BDR</td> <td>00:00:34</td> <td>10.8.3.2</td> <td>Ethernet0/0</td> </tr> <tr> <td>10.10.10.10</td> <td>0</td> <td>FULL/-</td> <td>00:01:55</td> <td>10.96.1.1</td> <td>Tunnel0</td> </tr> </tbody> </table>	nbr_id	priority	state	dead	nbr_addr	intf	10.8.3.9	1	FULL/BDR	00:00:35	10.8.3.6	Ethernet0/1	10.8.3.10	1	FULL/BDR	00:00:34	10.8.3.2	Ethernet0/0	10.10.10.10	0	FULL/-	00:01:55	10.96.1.1	Tunnel0
nbr_id	priority	state	dead	nbr_addr	intf																					
10.8.3.9	1	FULL/BDR	00:00:35	10.8.3.6	Ethernet0/1																					
10.8.3.10	1	FULL/BDR	00:00:34	10.8.3.2	Ethernet0/0																					
10.10.10.10	0	FULL/-	00:01:55	10.96.1.1	Tunnel0																					

10	Spanning Tree Overview	<p>Spanning Tree with VLAN and Root information:</p> <p><b>show spanning-tree summary</b></p> <pre> cl_output:Sci-Dist-3750-01#show spanning-tree summary Switch is in pvst mode Root bridge for: VLAN0001, VLAN0044, VLAN0042, VLAN0044, VLAN0046, VLAN0049 Extended system ID is enabled PortFast Default is disabled PortFast BPDU Guard Default is disabled PortFast BPDU Filter Default is disabled Loopguard Default is disabled EtherChannel misconfig guard is enabled UplinkFast is disabled BackboneFast is disabled Configured Pathcost method used is short Name Blocking Listening Learning Forwarding STP Active ----- VLAN0001 0 0 0 3 3 VLAN0042 0 0 0 3 3 VLAN0044 0 0 0 3 3 </pre> <p><b>Details for Sci-Dist-3750-01</b></p> <table border="1"> <thead> <tr> <th>vlan_id</th> <th>vlan_name</th> <th>vlan_status</th> <th>ports</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>default</td> <td>active</td> <td>Fa1/0/1, Fa1/0/2, Fa1/0/3, Fa1/0/4, Fa1/0/5, Fa1/0/6, Fa1/0/...</td> </tr> <tr> <td>40</td> <td>DATA</td> <td>active</td> <td></td> </tr> <tr> <td>42</td> <td>VOICE</td> <td>active</td> <td></td> </tr> <tr> <td>44</td> <td>WIRELESS</td> <td>active</td> <td></td> </tr> <tr> <td>46</td> <td>VLAN0046</td> <td>active</td> <td></td> </tr> <tr> <td>49</td> <td>VLAN0049</td> <td>active</td> <td></td> </tr> <tr> <td>1002</td> <td>fddi-default</td> <td>act/linstp</td> <td></td> </tr> </tbody> </table>	vlan_id	vlan_name	vlan_status	ports	1	default	active	Fa1/0/1, Fa1/0/2, Fa1/0/3, Fa1/0/4, Fa1/0/5, Fa1/0/6, Fa1/0/...	40	DATA	active		42	VOICE	active		44	WIRELESS	active		46	VLAN0046	active		49	VLAN0049	active		1002	fddi-default	act/linstp	
vlan_id	vlan_name	vlan_status	ports																															
1	default	active	Fa1/0/1, Fa1/0/2, Fa1/0/3, Fa1/0/4, Fa1/0/5, Fa1/0/6, Fa1/0/...																															
40	DATA	active																																
42	VOICE	active																																
44	WIRELESS	active																																
46	VLAN0046	active																																
49	VLAN0049	active																																
1002	fddi-default	act/linstp																																

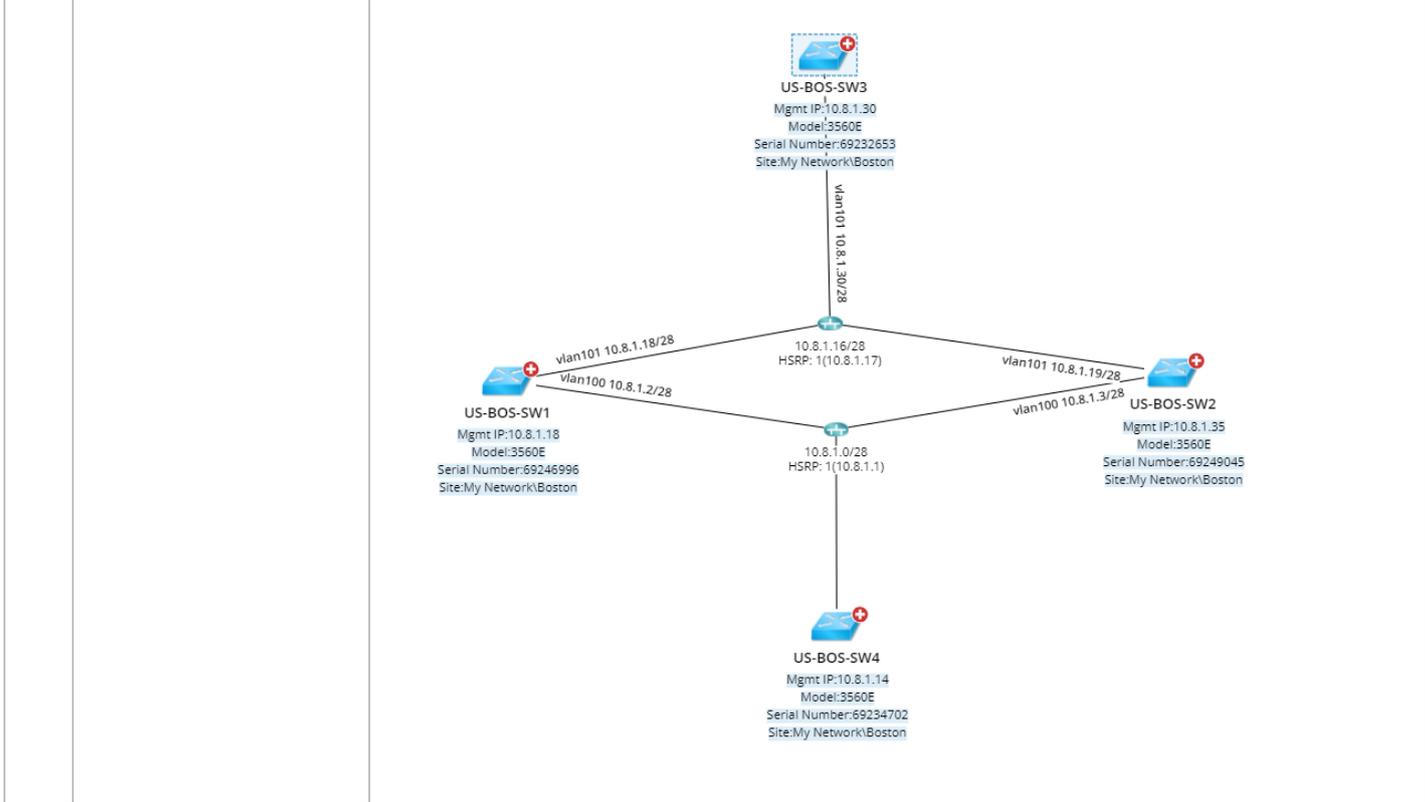
No.	Built-in DVT Name	Description and Sample Image																										
11	Spanning Tree VLAN Instance	<p>Input a VLAN ID, and this DVT will map the VLAN based STP view.</p>																										
12	HSRP Overview	<p>Display all HSRP interface status, and highlight HSRP interface states active/standby.</p> <table border="1"> <thead> <tr> <th>interface</th> <th>group</th> <th>state</th> <th>virtual_ip</th> <th>active_router</th> <th>standby_router</th> <th>priority</th> <th>configured_priority</th> <th>track</th> <th>preemption</th> <th>authentication</th> <th>hello_time</th> <th>hold_time</th> </tr> </thead> <tbody> <tr> <td>Vlan217</td> <td>0</td> <td>Standby</td> <td>10.4.40.120</td> <td>10.4.40.202</td> <td>local</td> <td>110</td> <td>110</td> <td></td> <td></td> <td></td> <td>3</td> <td>10</td> </tr> </tbody> </table>	interface	group	state	virtual_ip	active_router	standby_router	priority	configured_priority	track	preemption	authentication	hello_time	hold_time	Vlan217	0	Standby	10.4.40.120	10.4.40.202	local	110	110				3	10
interface	group	state	virtual_ip	active_router	standby_router	priority	configured_priority	track	preemption	authentication	hello_time	hold_time																
Vlan217	0	Standby	10.4.40.120	10.4.40.202	local	110	110				3	10																
13	Access List [Cisco IOS]	<p>Display access-list on the devices with detail information such as access-list name, hit count, inbound/outbound access-list applied to an interface.</p>																										

No.	Built-in DVT Name	Description and Sample Image																																				
		<table border="1"> <thead> <tr> <th>ad_rule</th> <th>match</th> <th>ad_type</th> <th>ad_name</th> </tr> </thead> <tbody> <tr> <td>10 permit 10.8.1.26 (1519 matches)</td> <td>1519</td> <td>Standard IP</td> <td>1</td> </tr> <tr> <td>20 permit 10.8.2.200 (91 matches)</td> <td>91</td> <td>Standard IP</td> <td>1</td> </tr> <tr> <td>30 permit 10.8.3.200 (395 matches)</td> <td>395</td> <td>Standard IP</td> <td>1</td> </tr> <tr> <td>10 permit ip host 10.8.1.26 any</td> <td></td> <td>Extended IP</td> <td>101</td> </tr> <tr> <td>10 permit tcp any host 10.1.1.1 eq telnet</td> <td></td> <td>Extended IP</td> <td>190</td> </tr> <tr> <td>10 permit top any any gr 1023 established (10453 matches)</td> <td></td> <td>Extended IP</td> <td>191</td> </tr> <tr> <td>10 permit top any any eq www</td> <td></td> <td>Extended IP</td> <td>192</td> </tr> <tr> <td>10 permit top any host 10.1.1.1 eq telnet</td> <td></td> <td>Extended IP</td> <td>193</td> </tr> </tbody> </table>	ad_rule	match	ad_type	ad_name	10 permit 10.8.1.26 (1519 matches)	1519	Standard IP	1	20 permit 10.8.2.200 (91 matches)	91	Standard IP	1	30 permit 10.8.3.200 (395 matches)	395	Standard IP	1	10 permit ip host 10.8.1.26 any		Extended IP	101	10 permit tcp any host 10.1.1.1 eq telnet		Extended IP	190	10 permit top any any gr 1023 established (10453 matches)		Extended IP	191	10 permit top any any eq www		Extended IP	192	10 permit top any host 10.1.1.1 eq telnet		Extended IP	193
ad_rule	match	ad_type	ad_name																																			
10 permit 10.8.1.26 (1519 matches)	1519	Standard IP	1																																			
20 permit 10.8.2.200 (91 matches)	91	Standard IP	1																																			
30 permit 10.8.3.200 (395 matches)	395	Standard IP	1																																			
10 permit ip host 10.8.1.26 any		Extended IP	101																																			
10 permit tcp any host 10.1.1.1 eq telnet		Extended IP	190																																			
10 permit top any any gr 1023 established (10453 matches)		Extended IP	191																																			
10 permit top any any eq www		Extended IP	192																																			
10 permit top any host 10.1.1.1 eq telnet		Extended IP	193																																			

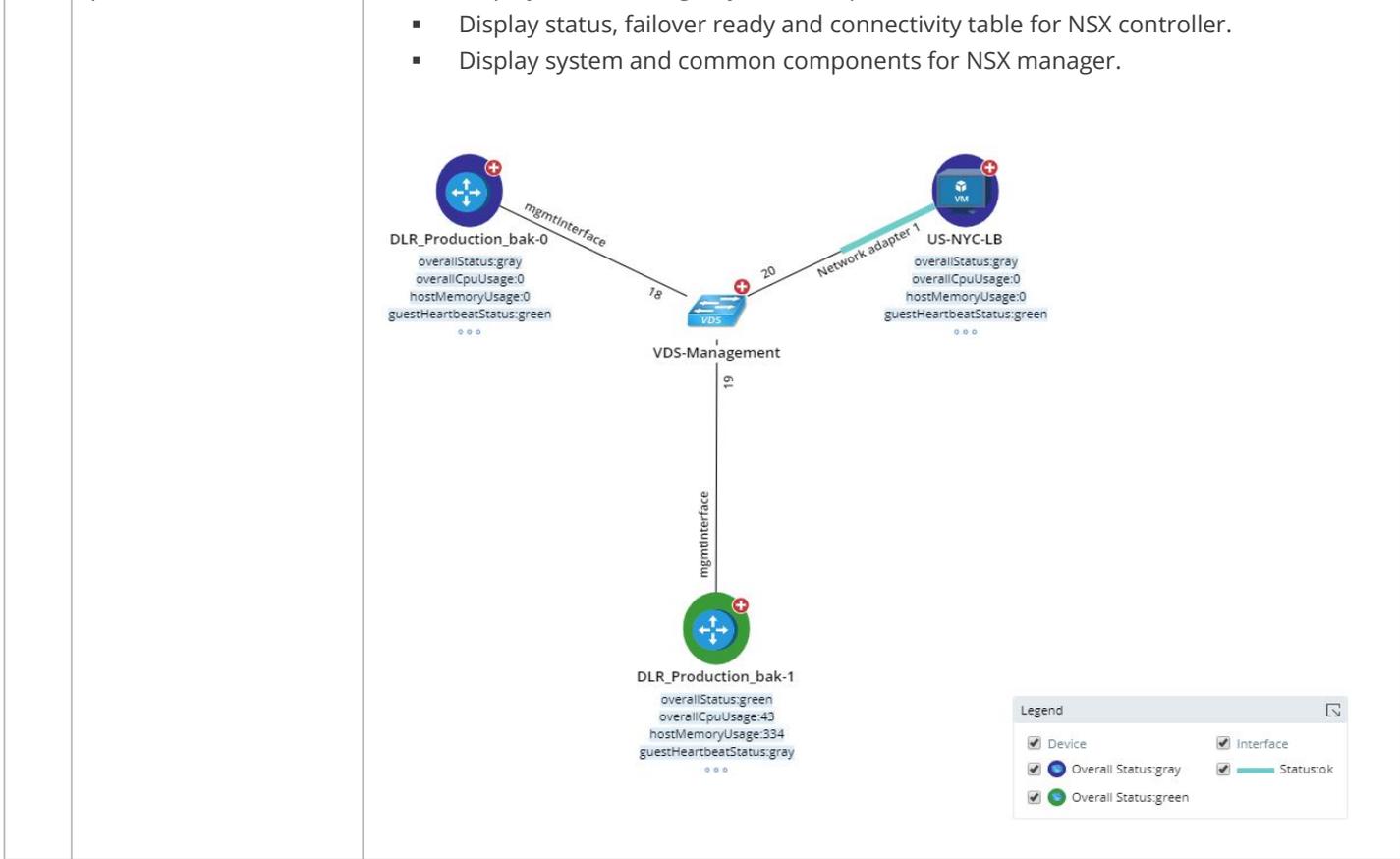
14	QoS Overview	<p>Display policy-map, class-map, mls QoS with detail information such as class-map with policy map, match information (dscp, access-list), inbound/outbound policy-map applied to interfaces.</p> <table border="1"> <thead> <tr> <th>class_map</th> <th>match_type</th> <th>match</th> <th>packets</th> <th>min5_drop_rate</th> <th>direction</th> <th>policy</th> <th>intf</th> </tr> </thead> <tbody> <tr> <td>video</td> <td>match-any</td> <td>ip dscp af41 (34)</td> <td>600545</td> <td>0</td> <td>input</td> <td>qos</td> <td>Ethernet0/2</td> </tr> <tr> <td>signal</td> <td>match-all</td> <td>access-group 190</td> <td>0</td> <td>0</td> <td>input</td> <td>qos</td> <td>Ethernet0/2</td> </tr> <tr> <td>voice</td> <td>match-all</td> <td>access-group 191</td> <td>10577</td> <td>0</td> <td>input</td> <td>qos</td> <td>Ethernet0/2</td> </tr> <tr> <td>data</td> <td>match-all</td> <td>access-group 192</td> <td>3105818</td> <td>0</td> <td>input</td> <td>qos</td> <td>Ethernet0/2</td> </tr> <tr> <td>general</td> <td>match-all</td> <td>access-group 193</td> <td>0</td> <td>0</td> <td>input</td> <td>qos</td> <td>Ethernet0/2</td> </tr> <tr> <td>class-default</td> <td>match-any</td> <td>any</td> <td>254809</td> <td>0</td> <td>input</td> <td>qos</td> <td>Ethernet0/2</td> </tr> </tbody> </table>	class_map	match_type	match	packets	min5_drop_rate	direction	policy	intf	video	match-any	ip dscp af41 (34)	600545	0	input	qos	Ethernet0/2	signal	match-all	access-group 190	0	0	input	qos	Ethernet0/2	voice	match-all	access-group 191	10577	0	input	qos	Ethernet0/2	data	match-all	access-group 192	3105818	0	input	qos	Ethernet0/2	general	match-all	access-group 193	0	0	input	qos	Ethernet0/2	class-default	match-any	any	254809	0	input	qos	Ethernet0/2
class_map	match_type	match	packets	min5_drop_rate	direction	policy	intf																																																			
video	match-any	ip dscp af41 (34)	600545	0	input	qos	Ethernet0/2																																																			
signal	match-all	access-group 190	0	0	input	qos	Ethernet0/2																																																			
voice	match-all	access-group 191	10577	0	input	qos	Ethernet0/2																																																			
data	match-all	access-group 192	3105818	0	input	qos	Ethernet0/2																																																			
general	match-all	access-group 193	0	0	input	qos	Ethernet0/2																																																			
class-default	match-any	any	254809	0	input	qos	Ethernet0/2																																																			

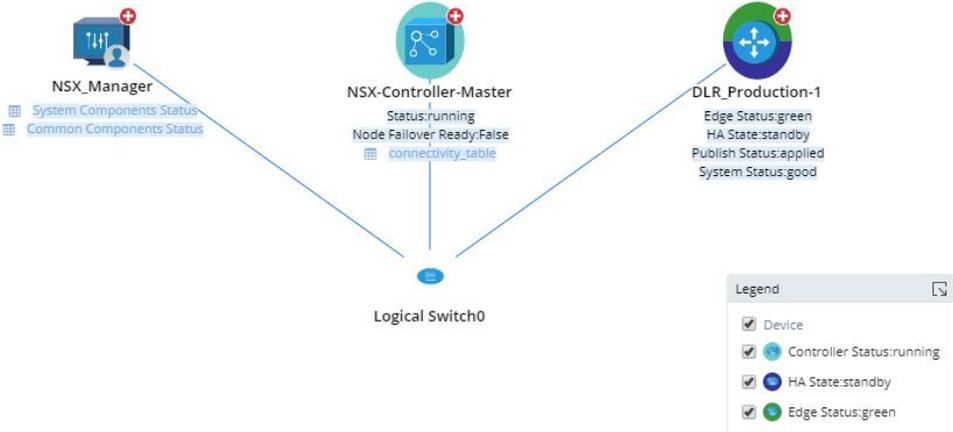
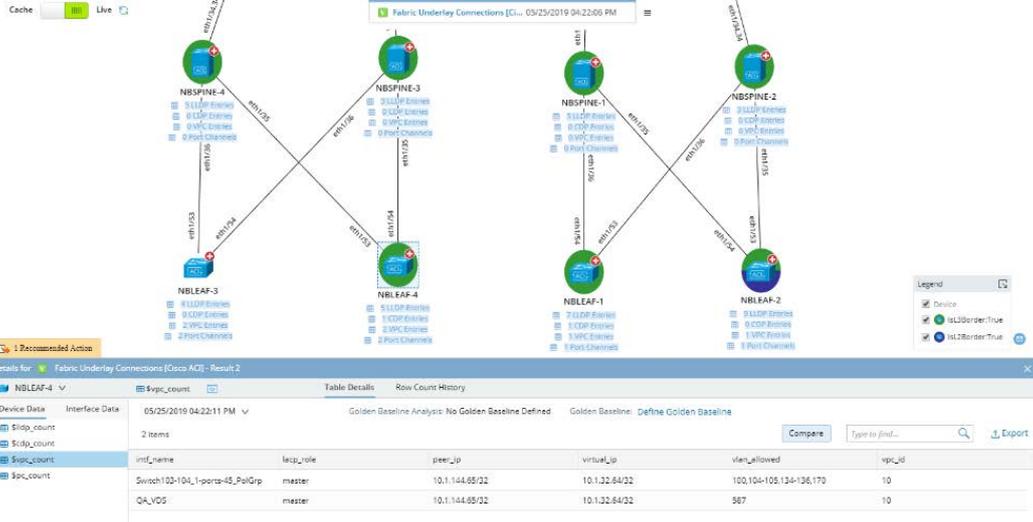
15	Infrastructure View [vCenter]	<ul style="list-style-type: none"> <li>Provide host, memory, vCPU and storage information along with interface connection status for NSX controller, manager, DLR, ESG and VMhost.</li> <li>Display interface information like port-group, vlan, connection status, etc (including VDS interfaces).</li> </ul>
----	-------------------------------	--

No.	Built-in DVT Name	Description and Sample Image
-----	-------------------	------------------------------



16	Operation View [NSX-V]	<ul style="list-style-type: none"> <li>Display HA state, edge, system and publish status for NSX DLR and ESG.</li> <li>Display status, failover ready and connectivity table for NSX controller.</li> <li>Display system and common components for NSX manager.</li> </ul>
----	------------------------	--



No.	Built-in DVT Name	Description and Sample Image																		
17	Operation View [vCenter]	<ul style="list-style-type: none"> <li>Display status and CPU &amp; memory usage along with heartbeat status for NSX controller, manager, DLR, ESG and VMhost.</li> <li>Display status of host for VDS.</li> </ul> 																		
18	Fabric Underlay Connections [Cisco ACI]	<p>Display Cisco Fabric underlay connection information, including LLDP and CDP neighbor table, port-channel and virtual port-channels configured on a device.</p>  <table border="1" data-bbox="603 1541 1508 1682"> <thead> <tr> <th>intf_name</th> <th>lacp_role</th> <th>peer_ip</th> <th>virtual_ip</th> <th>vni_allowed</th> <th>vpc_id</th> </tr> </thead> <tbody> <tr> <td>Sw0ch103-104_1-port2-45_PolGrp</td> <td>master</td> <td>10.1.144.85/22</td> <td>10.1.32.64/32</td> <td>100,104-108,134-136,170</td> <td>10</td> </tr> <tr> <td>QA_VDS</td> <td>master</td> <td>10.1.144.85/22</td> <td>10.1.32.64/32</td> <td>587</td> <td>10</td> </tr> </tbody> </table>	intf_name	lacp_role	peer_ip	virtual_ip	vni_allowed	vpc_id	Sw0ch103-104_1-port2-45_PolGrp	master	10.1.144.85/22	10.1.32.64/32	100,104-108,134-136,170	10	QA_VDS	master	10.1.144.85/22	10.1.32.64/32	587	10
intf_name	lacp_role	peer_ip	virtual_ip	vni_allowed	vpc_id															
Sw0ch103-104_1-port2-45_PolGrp	master	10.1.144.85/22	10.1.32.64/32	100,104-108,134-136,170	10															
QA_VDS	master	10.1.144.85/22	10.1.32.64/32	587	10															
19	Fabric Health and Faults [Cisco ACI]	Cisco ACI Fabric nodes health score and faults with critical logs.																		

No.	Built-in DVT Name	Description and Sample Image																																																																		
		<p>The screenshot displays the 'Fabric Health and Faults' interface for Cisco ACI. At the top, there's a 'Cache' section with 'Live' status. Below is a network topology diagram showing spine and leaf nodes (NBSPINE-1 to NBSPINE-4 and NBLEAF-1 to NBLEAF-4) connected in a mesh. A table below the diagram lists fault details for NBLEAF-1.</p> <table border="1"> <thead> <tr> <th>ack</th> <th>cause</th> <th>cisco_fault_url</th> <th>code</th> <th>created</th> <th>descr</th> <th>domain</th> <th>lastTransition</th> <th>severity</th> <th>subject</th> <th>type</th> </tr> </thead> <tbody> <tr> <td>no</td> <td>interface-physical...</td> <td>https://pubhubde...</td> <td>F0546</td> <td>2019-04-03T21:06...</td> <td>Port is down, reas...</td> <td>access</td> <td>2019-04-03T21:08...</td> <td>warning</td> <td>port down</td> <td>communications</td> </tr> <tr> <td>no</td> <td>interface-physical...</td> <td>https://pubhubde...</td> <td>F0532</td> <td>2019-05-01T19:56...</td> <td>Port is down, reas...</td> <td>access</td> <td>2019-05-01T19:58...</td> <td>critical</td> <td>port down</td> <td>communications</td> </tr> <tr> <td>no</td> <td>configuration-failed</td> <td>https://pubhubde...</td> <td>F0467</td> <td>2019-03-01T19:45...</td> <td>Configuration fail...</td> <td>tenant</td> <td>2019-03-01T19:47...</td> <td>minor</td> <td>management</td> <td>config</td> </tr> <tr> <td>no</td> <td>protocol-ntp-sync...</td> <td>https://pubhubde...</td> <td>F1700</td> <td>2018-11-15T07:21...</td> <td>ntp configuration ...</td> <td>infra</td> <td>2018-11-15T07:23...</td> <td>minor</td> <td>ntp</td> <td>operational</td> </tr> <tr> <td>no</td> <td>configuration-failed</td> <td>https://pubhubde...</td> <td>F1295</td> <td>2018-11-15T07:21...</td> <td>Dateline Policy Co...</td> <td>infra</td> <td>2018-11-15T07:23...</td> <td>minor</td> <td>management</td> <td>config</td> </tr> </tbody> </table>	ack	cause	cisco_fault_url	code	created	descr	domain	lastTransition	severity	subject	type	no	interface-physical...	https://pubhubde...	F0546	2019-04-03T21:06...	Port is down, reas...	access	2019-04-03T21:08...	warning	port down	communications	no	interface-physical...	https://pubhubde...	F0532	2019-05-01T19:56...	Port is down, reas...	access	2019-05-01T19:58...	critical	port down	communications	no	configuration-failed	https://pubhubde...	F0467	2019-03-01T19:45...	Configuration fail...	tenant	2019-03-01T19:47...	minor	management	config	no	protocol-ntp-sync...	https://pubhubde...	F1700	2018-11-15T07:21...	ntp configuration ...	infra	2018-11-15T07:23...	minor	ntp	operational	no	configuration-failed	https://pubhubde...	F1295	2018-11-15T07:21...	Dateline Policy Co...	infra	2018-11-15T07:23...	minor	management	config
ack	cause	cisco_fault_url	code	created	descr	domain	lastTransition	severity	subject	type																																																										
no	interface-physical...	https://pubhubde...	F0546	2019-04-03T21:06...	Port is down, reas...	access	2019-04-03T21:08...	warning	port down	communications																																																										
no	interface-physical...	https://pubhubde...	F0532	2019-05-01T19:56...	Port is down, reas...	access	2019-05-01T19:58...	critical	port down	communications																																																										
no	configuration-failed	https://pubhubde...	F0467	2019-03-01T19:45...	Configuration fail...	tenant	2019-03-01T19:47...	minor	management	config																																																										
no	protocol-ntp-sync...	https://pubhubde...	F1700	2018-11-15T07:21...	ntp configuration ...	infra	2018-11-15T07:23...	minor	ntp	operational																																																										
no	configuration-failed	https://pubhubde...	F1295	2018-11-15T07:21...	Dateline Policy Co...	infra	2018-11-15T07:23...	minor	management	config																																																										
20	Logic Node View [Cisco ACI]	<p>Cisco ACI Contract, EPG, Bridge Domain, L3 Out, L2 Out basic</p> <p>The screenshot displays the 'Logic Node View' interface for Cisco ACI. It shows a network topology with various nodes like L3OUT-246-External-EPG, L3Out-ASA, APP-L3OUT-EPG, WEB-EPG, DB-EPG, APP-EPG, vsig_DB-EPG, and APP_Web-EPG. Below the diagram is a table listing details for WEB-EPG.</p> <table border="1"> <thead> <tr> <th>ep_name</th> <th>ip</th> <th>mac</th> <th>vlan</th> <th>vmmGrp</th> </tr> </thead> <tbody> <tr> <td>00:50:56:0c:30:7b</td> <td>20.0.0.10</td> <td>00:50:56:0c:30:7b</td> <td>vlan=135</td> <td>dis</td> </tr> <tr> <td>00:50:56:0c:33:39</td> <td>20.0.0.11</td> <td>00:50:56:0c:33:39</td> <td>vlan=135</td> <td>dis</td> </tr> <tr> <td>00:50:56:0c:1b:50</td> <td>20.0.0.20</td> <td>00:50:56:0c:1b:50</td> <td>vlan=135</td> <td>dis</td> </tr> <tr> <td>00:50:56:0c:33:af</td> <td>20.0.0.21</td> <td>00:50:56:0c:33:af</td> <td>vlan=135</td> <td>dis</td> </tr> </tbody> </table>	ep_name	ip	mac	vlan	vmmGrp	00:50:56:0c:30:7b	20.0.0.10	00:50:56:0c:30:7b	vlan=135	dis	00:50:56:0c:33:39	20.0.0.11	00:50:56:0c:33:39	vlan=135	dis	00:50:56:0c:1b:50	20.0.0.20	00:50:56:0c:1b:50	vlan=135	dis	00:50:56:0c:33:af	20.0.0.21	00:50:56:0c:33:af	vlan=135	dis																																									
ep_name	ip	mac	vlan	vmmGrp																																																																
00:50:56:0c:30:7b	20.0.0.10	00:50:56:0c:30:7b	vlan=135	dis																																																																
00:50:56:0c:33:39	20.0.0.11	00:50:56:0c:33:39	vlan=135	dis																																																																
00:50:56:0c:1b:50	20.0.0.20	00:50:56:0c:1b:50	vlan=135	dis																																																																
00:50:56:0c:33:af	20.0.0.21	00:50:56:0c:33:af	vlan=135	dis																																																																

### 2.1.5. Schedule Data Retrieval for DVT

IEv8.0 adds a new type of scheduled task to retrieve and parse network data for data view templates or parsers. When a Data View Template is applied to a map, the cached data can be instantly used to create a Data View instance.

**Tip:** This new task type can also be triggered through Benchmark to adapt to Benchmark frequency.

## Reference Flow



**Note:** When a data view template that contains an auto-generated [input variable](#) is scheduled and executed, the system will pull all optional values of the input variable and pass each of them to the corresponding parser to generate CLI command instances. The **Max Command Instances of a Parameterized Parser for Each Device** field is used to limit the generated CLI command instances for each device included in this task. If the selected parsers have parameters, assign a smaller value to avoid devices overloaded due to the execution of too many CLI commands. The default value is 32.

Task Name: Demo Description:

Frequency Device Scope Select Data View Template/Parser

▼ Data View Template ⓘ  
+ Add

No.	Folder/File	Location
1	Infrastructure Information	Data View Templates/Built-in Data View...
2	Maintenance Information	Data View Templates/Built-in Data View...
3	Network Design	Data View Templates/Built-in Data View...

> Parser ⓘ

Max Command Instances of a Parameterized Parser for Each Device: 32 ⓘ

Cancel Submit

## 2.1.6. Message Golden Baseline Alerts to More Users

To warn end users about network data deviations when they are not monitoring, power users can message alerts via NetBrain system alerts, or email alerts, or both. As a result, end users can timely respond by analyzing the root cause on an addressed dynamic map and performing drill-down troubleshooting.

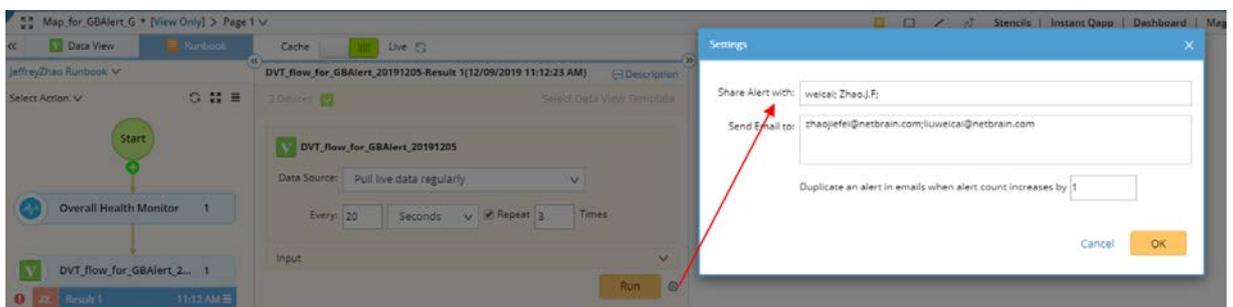
By default, only the action executor can receive system alerts for golden baseline deviations. The executor can configure whether to message more users proactively. Detected alerts will be categorized as errors in the Event Console.

### Use Flow

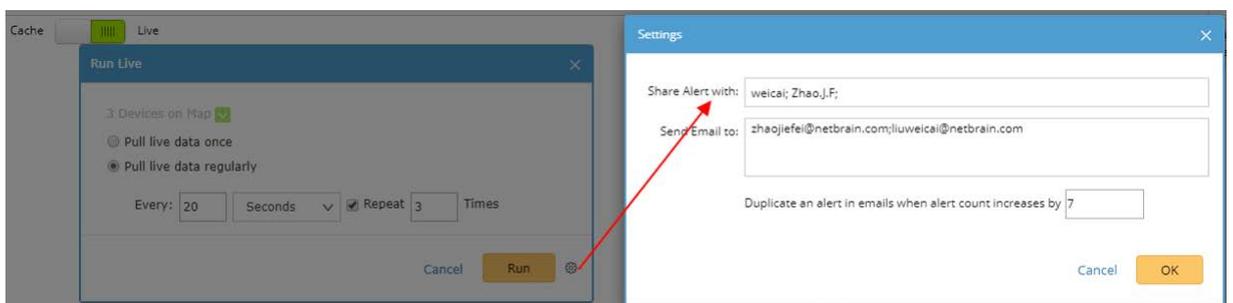


1. Configure the alert settings for a Data View Template by sharing with specified domain user accounts, or sending email alerts to specified email addresses, or both. For example:

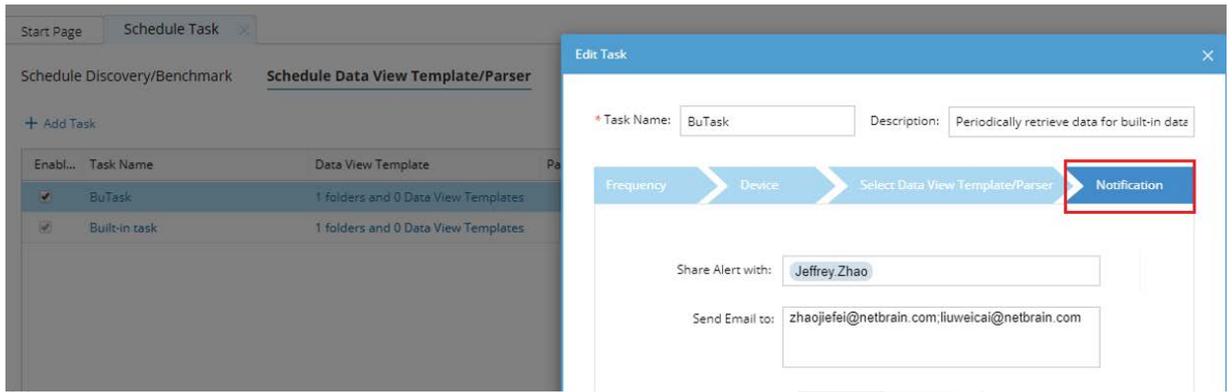
- Before running a Data View Template Node in a Runbook:



- When toggling to apply a Data View Template with Live Data to a map:



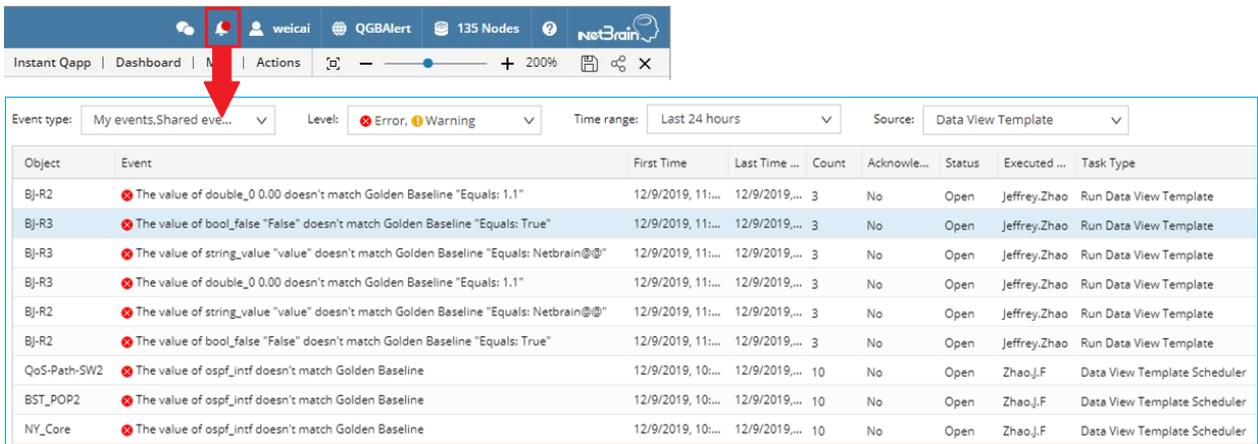
- When scheduling a Data View Template Task:



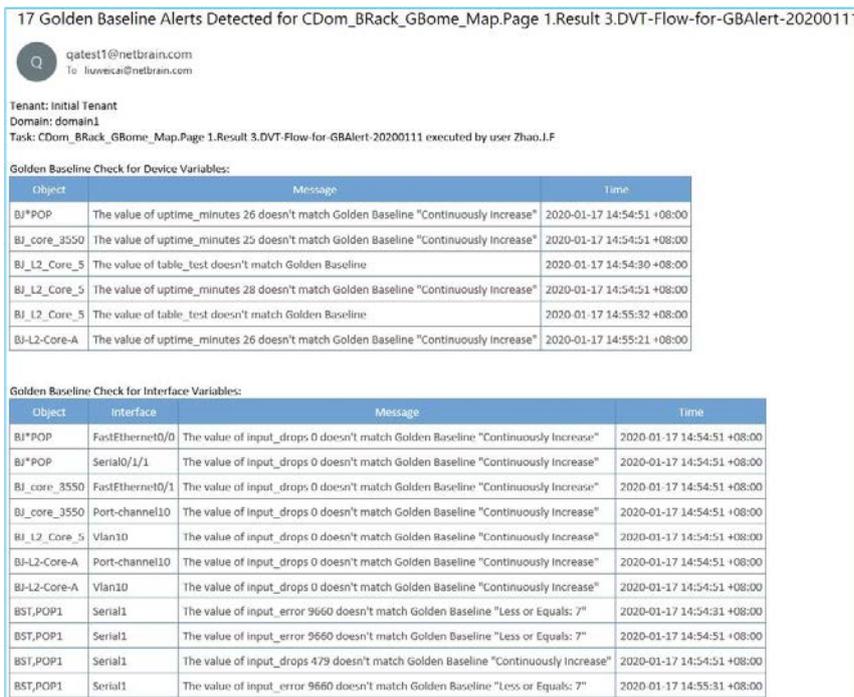
For more details about the email alerting rules, refer to [Alerting Rules](#).

- Receive alert messages from system alerts, or email alerts, or both.

**Example 1: Golden Baseline Alerts in System Event Console**



**Example 2: Golden Baseline Alert Messages in Email**



### 3. Open the map to troubleshoot.

The screenshot displays the NetBrain interface. At the top, a table lists several alerts:

Object	Event	First Time	Last Time	Count	Acknowledge	Status	Executed	Task Type	From Task
BJ-R2	The value of double_0 0.00 doesn't match Golden Baseline "Equals: 1.1"	12/9/2019, 11:...	12/9/2019, ...	3	No	Open	Jeffrey.Zhao	Run Data View Template	<Map>DVT_flow_for_GBAlert_20191205
BJ-R3	The value of bool_false "False" doesn't match Golden Baseline "Equals: True"	12/9/2019, 11:...	12/9/2019, ...	3	No	Open	Jeffrey.Zhao	Run Data View Template	_flow_for_GBAlert_20191205
BJ-R3	The value of string_value "value" doesn't match Golden Baseline "Equals: Netbrain@@"	12/9/2019, 11:...	12/9/2019, ...	3	No	Open	Jeffrey.Zhao	Run Data View Template	_flow_for_GBAlert_20191205
BJ-R3	The value of double_0 0.00 doesn't match Golden Baseline "Equals: 1.1"	12/9/2019, 11:...	12/9/2019, ...	3	No	Open	Jeffrey.Zhao	Run Data View Template	_flow_for_GBAlert_20191205
BJ-R2	The value of string_value "value" doesn't match Golden Baseline "Equals: Netbrain@@"	12/9/2019, 11:...	12/9/2019, ...	3	No	Open	Jeffrey.Zhao	Run Data View Template	flow_for_GBAlert_20191205
BJ-R2	The value of bool_false "False" doesn't match Golden Baseline "Equals: True"	12/9/2019, 11:...	12/9/2019, ...	3	No	Open	Jeffrey.Zhao	Run Data View Template	flow_for_GBAlert_20191205

Below the table, a network map is shown. A red arrow points to the 'Open Map' button in the context menu. The map displays a network topology with nodes like BJ-R2 and BJ-R3. A detailed view of a Golden Baseline Alert is shown, indicating that the value of 'double\_0' (0.00) does not match the Golden Baseline 'Equals: 1.1'. The alert details include:

- Golden Baseline Analysis: Alert Detected
- Golden Baseline: Equals: 1.1
- Alert Message: The value of double\_0 0.00 doesn't match Golden Baseline "Equals: 1.1"
- Alert Time: 11:12:30 AM

**Note:** If the alert is detected in a scheduled DVT task, the alert message will be attached to the map as a device note.

The screenshot shows a network map with several devices: LA.DIS,1 (Cisco Router), LA\_POP (Cisco Router), BJ\*POP (Cisco Router), and XE-MGMT (Cisco IOS Switch). A yellow alert box is attached to the BJ\*POP device, displaying the following information:

**Golden Baseline Alert**

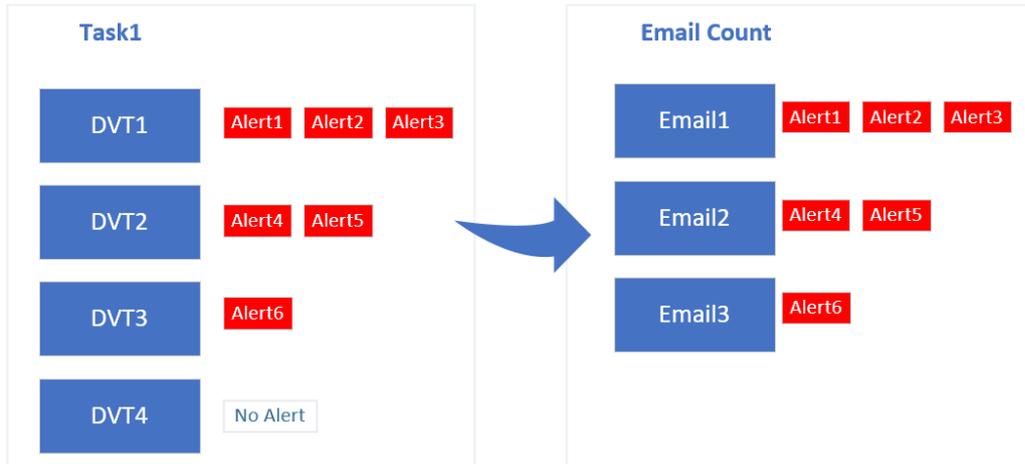
The value of five\_min\_cpu\_usage 11 doesn't match Golden Baseline "Equals: 4"

Jeffrey.Zhao 12/11/2019, 5:16:23 PM

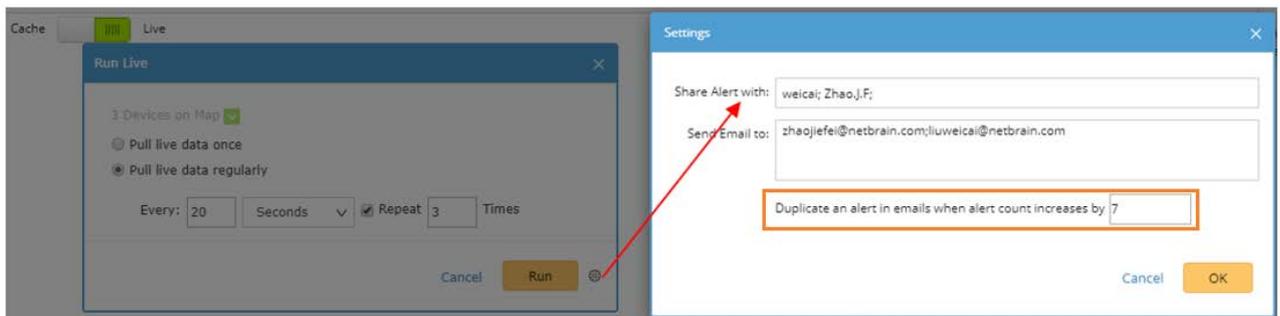
## Emailing Rules for Golden Baseline Alerts

The system provides three general rules to avoid duplicate and excessive email alerts.

1. By default, the system sends alert emails every 5 minutes.  
The frequency is configurable at **System Management > Email Settings**.
2. The system emails GB Alerts per Data View Template.  
For example, if a task (on-demand or scheduled) contains 4 Data View Templates, and 3 of them have Golden Baseline deviations, the system will send out 3 emails separately.



3. When the data source of an on-demand DVT task is "Pull live data regularly", the system will duplicate an alert message in one email if it reaches a specified threshold. It is configurable in the alert settings, and the default threshold value is 7.



## 2.2. Static and Map Data View

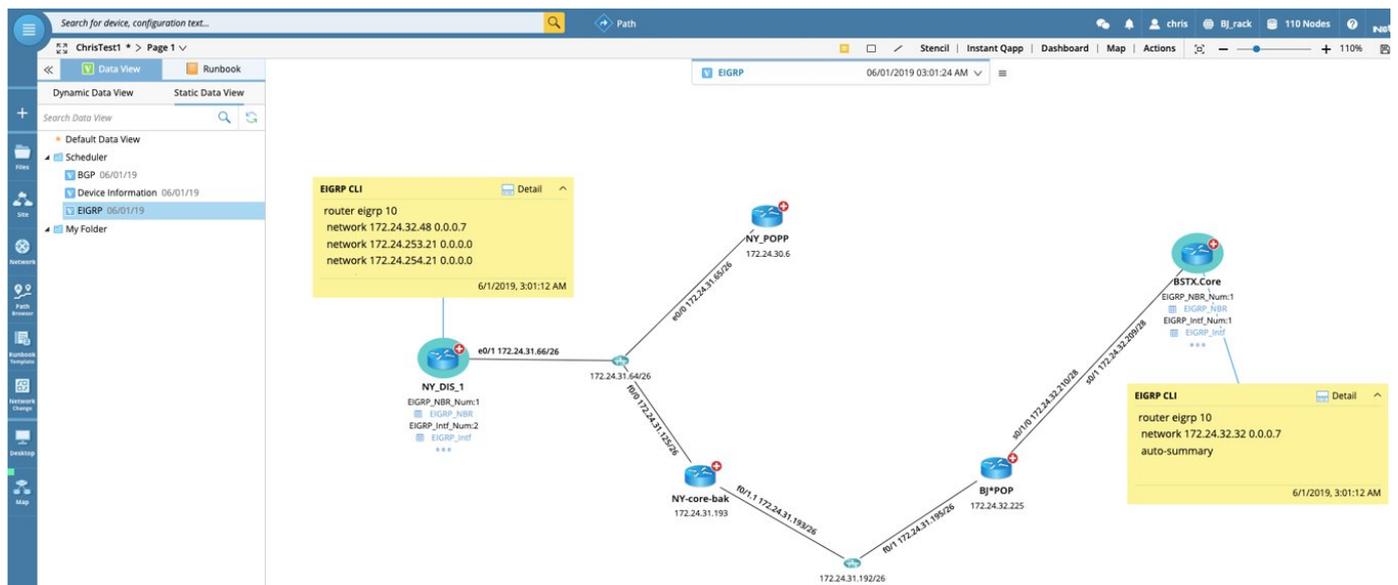
### 2.2.1. Static Data View

Static Data View (SDV, sometimes also called Global Data View) is different from Data View Template (DVT). DVT is only a template without instantiating to a specific device. SDV is a data view instance. The data of an SDV can be out-of-date unless users manually modify it.

1. [Apply a Static Data View](#)
2. [Edit a Static Data View](#)

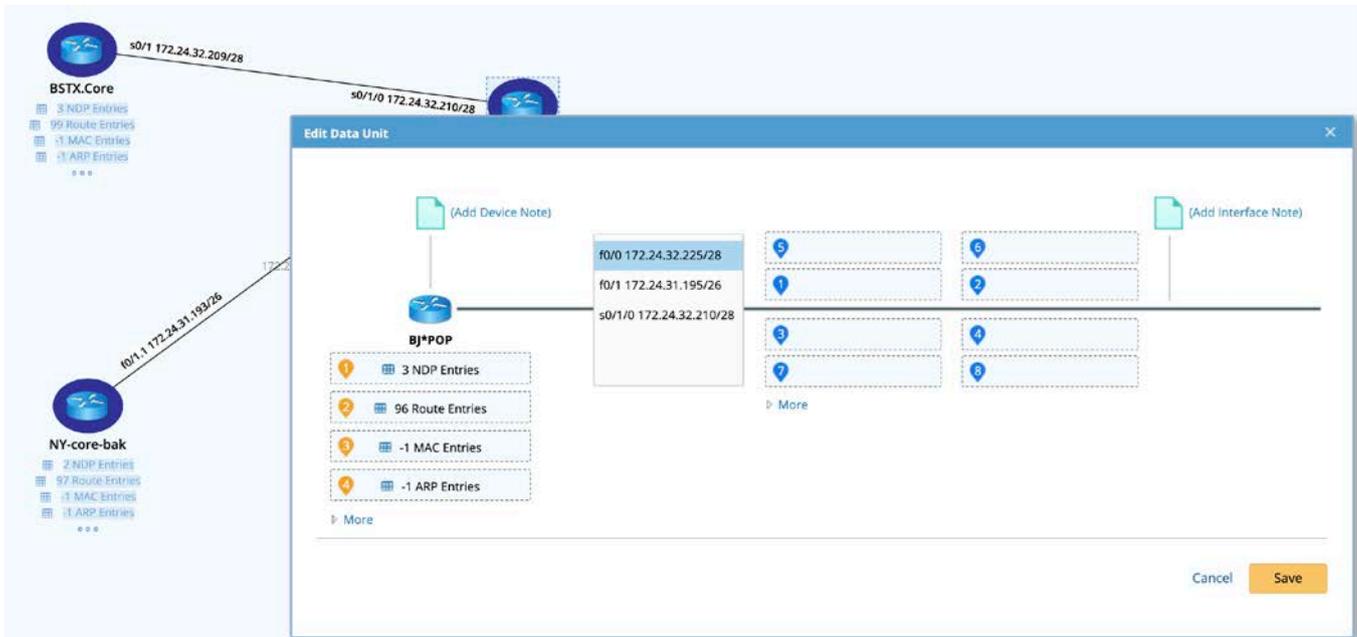
### Apply a Static Data View

The Static Data Views related to devices on a map are listed when a user opens the map. The user can select an SDV to apply to the map based on the actual user task.



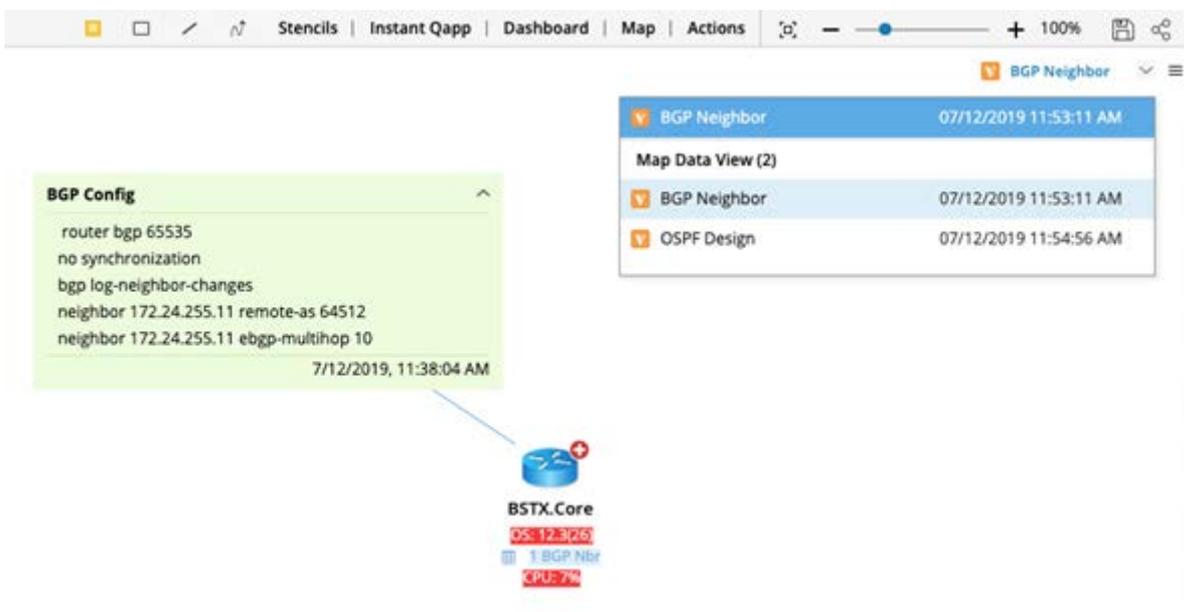
### Edit Static Data View

The Static Data View can be used to document and share the device data across different maps. Users are allowed to edit a Static Data View to add the data which cannot be retrieved from the live network.



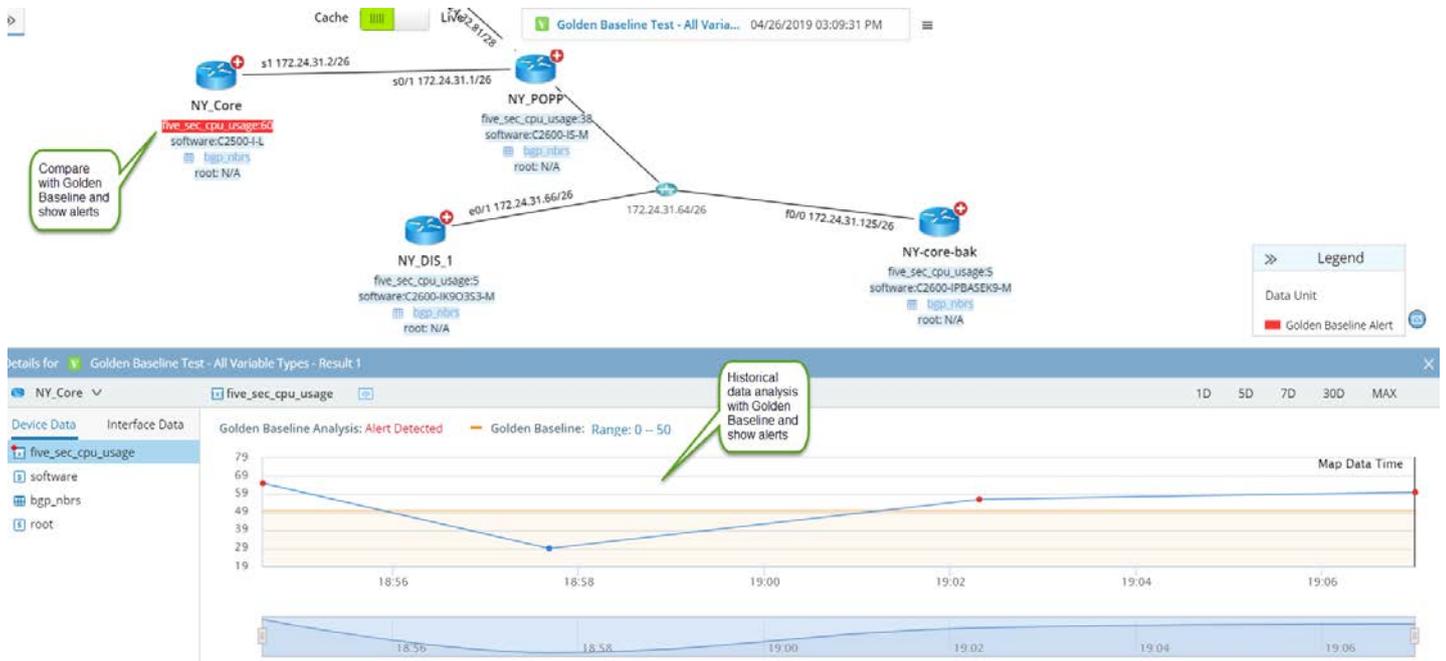
## 2.2.2. Map Data View

Map Data View can be recorded and displayed only on the current map, which is a method to document data within the map itself. However, it will be affected if the map file is deleted or the devices on the map change. The Map Data View is listed on the top of the map, instead of the Data View pane beside the map.



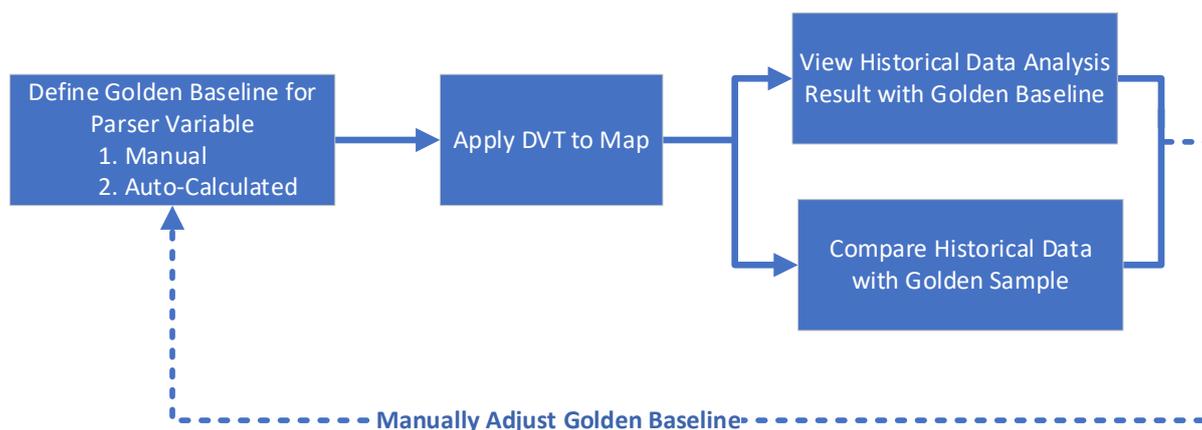
## 2.3. Golden Baseline

Golden Baseline serves as the reference standard of “healthy” status for various types of data. It can be auto-calculated based on historical statistics or manually defined, and can be further used as a basis or foundation for network management analysis and comparison, e.g., troubleshooting.



**Note:** IeV8.0 only provides a golden baseline for parser variables of legacy devices.

## Reference Flow



1. [Manually Define Golden Baseline](#)
2. [Auto Calculate Golden Baseline](#)
3. [View Golden Baseline Analysis Result](#)

## 2.3.1. Manually Define Golden Baseline

The system provides two ways to manually define Golden Baseline for parser variables:

- [Define golden baseline for parser variables on a map.](#)
- [Define golden baseline for parser variables in Golden Baseline Manager.](#)

## Golden Baseline Rule Definition

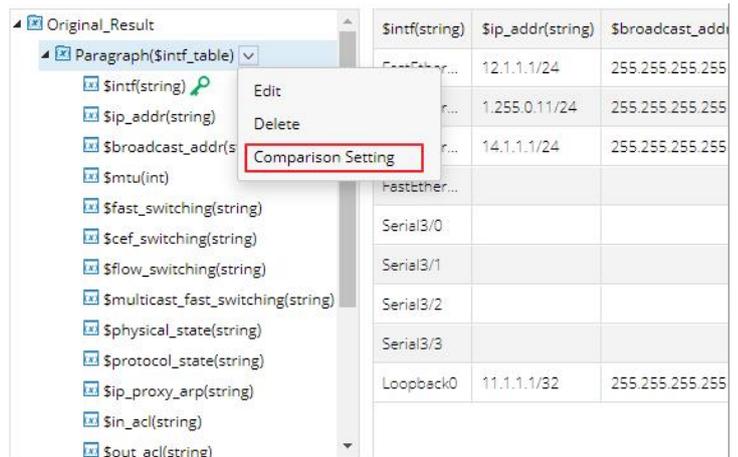
The definition of Golden Baseline rules for different parser variable types are as follows:

Variable Type	Golden Baseline Rule	Example
Number	Equals	"\$BGP_neighbor_count" equals 3
	Equals any of	"\$mtu" equals 1500 or 1514
	Not Equals to	"\$In_traffic" not equals to 0
	Range	"\$BGP_routes" range (400, 420)
	Less or Equals	"\$interface_utilization" less than or equals 50%
	Greater or Equals	"\$tunnel_counts" greater than or equals 3
	No Change - Equals Last Value	"\$CRC_error" <b>Note:</b> Use this rule if the expected behavior for the variable should be kept as it is and should not change.
	Continuously Increase	"\$device_up_time"

Variable Type	Golden Baseline Rule	Example
		<b>Note:</b> Use this rule if the expected behavior for the variable should keep increasing.
String	Equals	"\$version_number" equals 12.1
	Equals any of	"\$ospf_neighbor_status" equals any of "2 way", "established"
	Not Equals to	"\$Interface_status" Not equals to "Down"
	Regex	Enter a regular expression and the system will search the entire parser variable results to see if there's a match.
Table *)	Equals	"\$BGP_neighbor" equals XXX

**\*) Prerequisites to Define Golden Baseline for Table-Type Variables:**

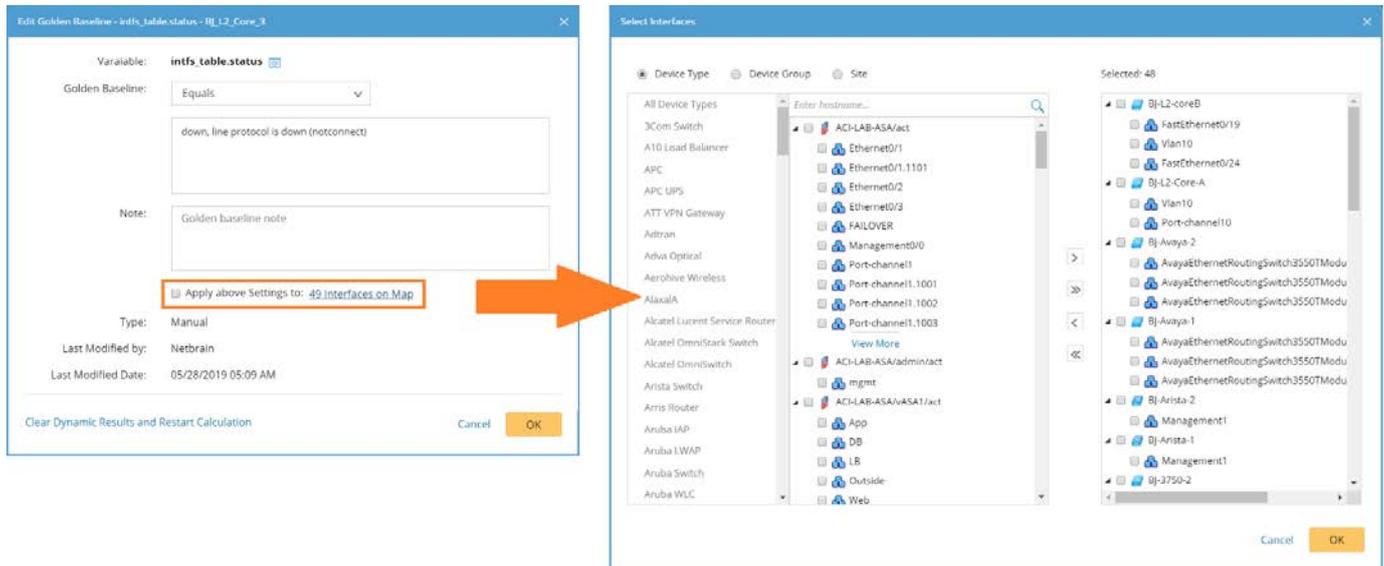
1. An Interface Key or Table Key is required for each table-type parser variable. Otherwise, neither manual definition nor dynamic calculation can be done to set the Golden Baseline for table-type variables.
2. The Golden Baseline definition for table-type variables allows selecting only part of table columns, which can be done in parser definition. It means unimportant columns or constantly changing columns can be ignored. For example, the "BGP Neighbor", "Version" and "AS Number" in the following BGP Table is defined as Golden, and the other three columns are not involved in Golden Baseline calculation.



BGP Table Key			Compared Columns			Ignored Columns		
BGP Neighbor	Version	AS	MsgRCD	MsgSent	InQ			
10.100.1.1	4	200	26	22	0			
10.200.1.1	4	200	51	23	0			

## Define Golden Baseline Rules on Map

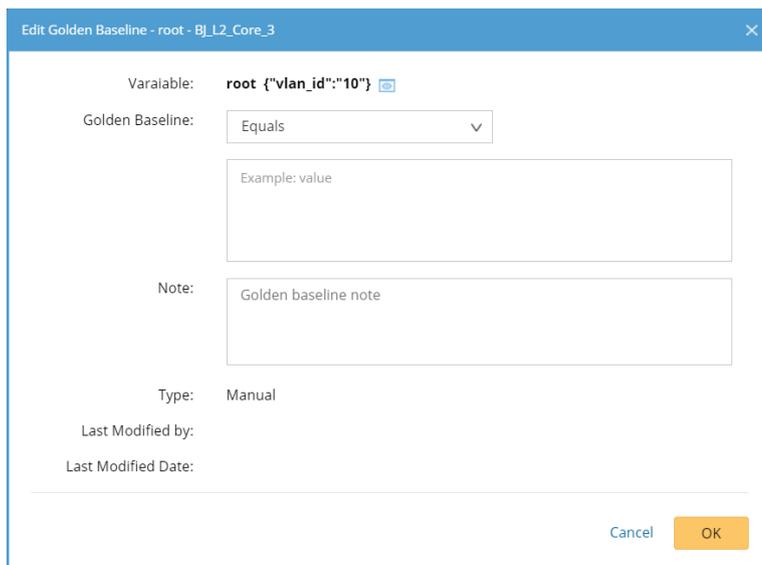
After applying a Data View Template to a map, users can define Golden Baseline Rules for parser variables according to their variable types, and apply the rules to devices/interfaces on the map.



Moreover, the system allows users to apply the rules to multiple devices/interfaces, even for those not in the map or not of the same device type.

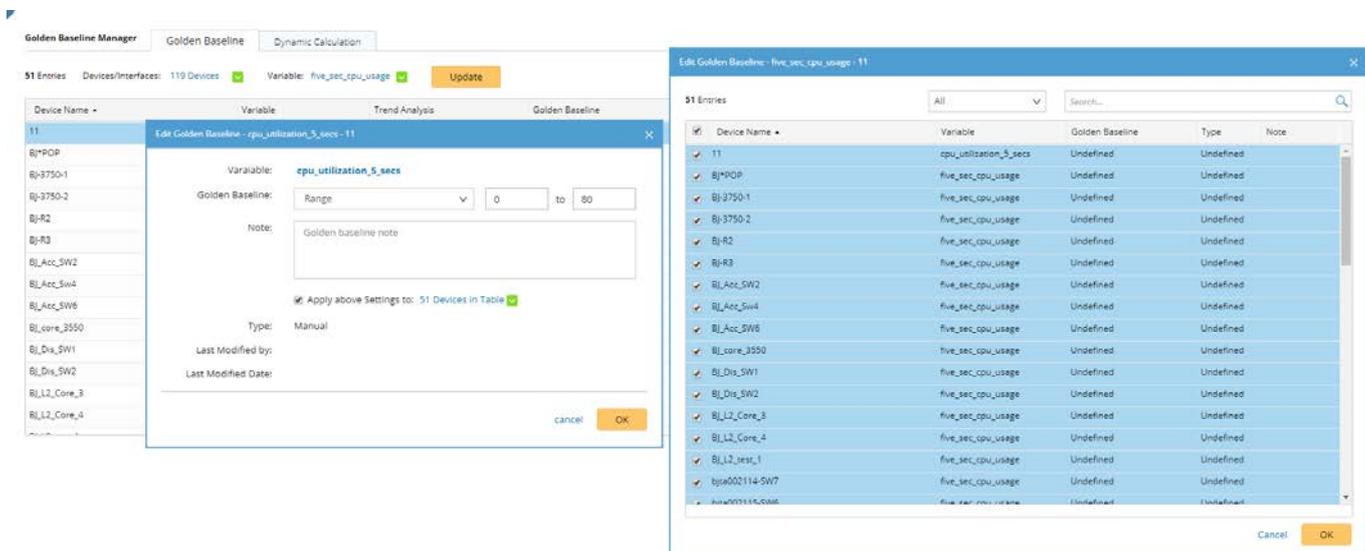
**Note:** When the rules are applied to other device types, the system will look up variable mappings to find out the corresponding variables of those devices.

**Note:** When applying a DVT that includes an [Input Variable](#), users can input different values for different devices. However, the Golden Baseline Rule for the Input Variable cannot be applied to other devices/interfaces.



## Define Golden Baseline Rules in Golden Baseline Manager

In Golden Baseline Manager, users can define Golden Baseline Rules for parser variables to a batch of devices/interfaces. Through this method, users can batch define, modify, and delete Golden Baselines for parser variables.



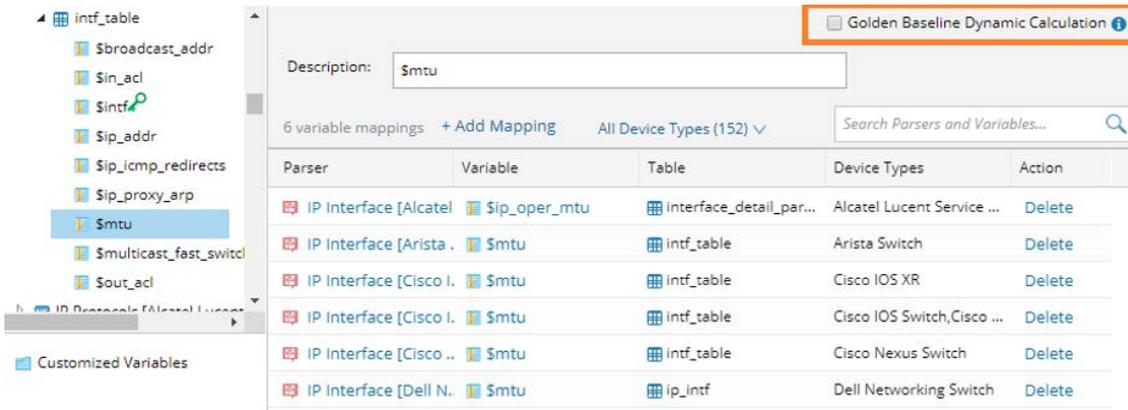
The batch of devices/interfaces can be selected per Device Type, Device Group, or Site. IEv8.0 introduces [Enhanced Device Group](#) to enable users to batch define Golden Baseline for a group of interfaces.

### 2.3.2.Auto Calculate Golden Baseline

With Machine Learning, IEv8.0 provides an auto-calculation mechanism to set Golden Baseline for parser variables based on historical data generated through scheduled tasks. Once a Golden Baseline is manually modified with customized rules, its auto-calculation function will be disabled. To re-enable dynamic calculation for it, users can clear the manually defined Golden Baseline Rule.

### Enable/Disable Dynamic Calculation

An option to enable/disable the dynamic calculation of Golden Baseline for a parser variable is offered through **Variable Mapping > Global Variable**.



**Note:** Only the dynamic calculation of single-value variables, table variables, and interfaces' table column variables can be enabled.

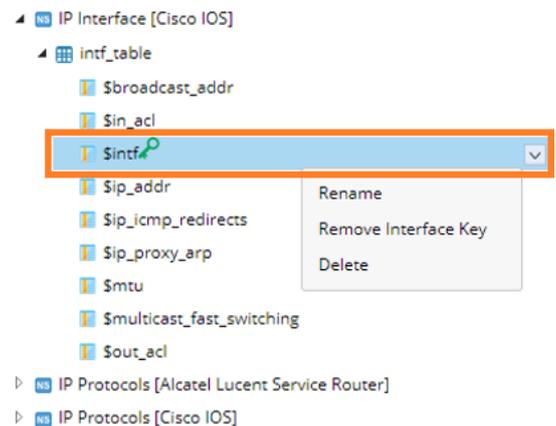
**Note:** To enable the dynamic calculation for interfaces' table column variables, you need to enable an interface key.

▪ **Enable/Disable Dynamic Calculation Globally:**

The Golden Baseline Manager provides an option to disable the dynamic calculations for all Golden Baselines. As a result, all the existing dynamic calculation process will be disabled.

▪ **Clear Dynamic Calculation Results:**

The system allows users to clear all the existing dynamic calculation results for a Golden Baseline and will re-calculate with new data.



## Dynamic Calculation Logic for Different Variable Types

Golden Baseline for string/number/table-type variables can be dynamically calculated.

**1. String Type:** The Golden Baseline for the following two string-type variables can be dynamically calculated.

- **Equals** – at least requires data at 4 different time points for calculation, and the first three values must be exactly the same. See the following table for the example of the IOS version.
- **Equals any of** – at least requires data at 10 different time points for calculation, and the count of instance status must be no greater than 3.

	1 <sup>st</sup> Time	2 <sup>nd</sup> Time	3 <sup>rd</sup> Time	4 <sup>th</sup> Time	5 <sup>th</sup> Time	6 <sup>th</sup> Time	7 <sup>th</sup> Time	8 <sup>th</sup> Time
Data (IOS Version)	12.1	12.1	12.1	12.1	12.2	12.2	12.2	12.2

	1 <sup>st</sup> Time	2 <sup>nd</sup> Time	3 <sup>rd</sup> Time	4 <sup>th</sup> Time	5 <sup>th</sup> Time	6 <sup>th</sup> Time	7 <sup>th</sup> Time	8 <sup>th</sup> Time
<b>Golden Baseline</b>	Calculating	Calculating	Calculating	Equals 12.1	Equals 12.1	Equals 12.1	Equals 12.1	Equals 12.2
<b>Alert</b>	N/A	N/A	N/A	No Alerts	Alert Detected	Alert Detected	Alert Detected	No Alerts

**2. Number Type:** The Golden Baseline for the following number-type variables can be dynamically calculated:

Available Type	Required Data Points
<b>Equals</b>	at least requires data at 4 different time points for calculation, and the first three values must be exactly the same.
<b>Equals any of Range</b>	at least requires data at 10 different time points for calculation.
<b>No Change – Equals Last Value</b>	
<b>Continuously Increase</b>	

**3. Table Type:**

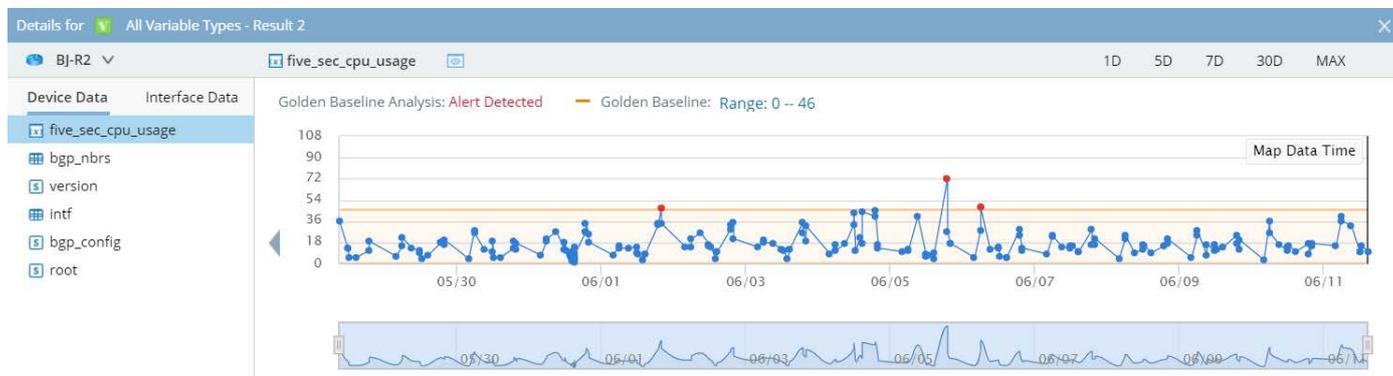
According to the predefined compared columns, the system generates hash values in parsers, and convert into single-value strings for comparison and analysis. The analysis logic is the same as that for string-type variables.

Take the following table for example, if the “BGP Neighbor”, “Version” and “AS Number” columns are predefined as compared columns, then only the corresponding hash values for these three columns will be generated for comparison.

Compared Columns			Ignored Columns		
<b>BGP Neighbor</b>	<b>Version</b>	<b>AS</b>	MsgRCD	MsgSent	InQ
10.100.1.1	4	200	26	22	0
10.200.1.1	4	200	51	23	0

## Initial Value for Golden Baseline Calculation

Automation Team will accelerate the convergence for the Golden Baseline dynamic calculation of range-type variables by specifying an initial value, for example, "0 - 90%". With the specified initial value, the Golden Baseline can be gradually calculated with convergence ever since the first cycle of data retrieval.



**Note:** Only the data retrieved through scheduled tasks can be involved in Golden Baseline dynamic calculation, such as scheduled Qapp/Gapp task, or scheduled DVT/Parser task. The data generated by on-demand tasks will not be involved in the Golden Baseline calculation.

### 2.3.3.View Golden Baseline Analysis Result

The analysis and comparison result between historical data and Golden Baseline can be visualized by applying a Data View Template to a map. The analysis results between the last copy of historical data and Golden Baseline are displayed on the map, and the Detail Pane offers the analysis results between more historical data and Golden Baseline. See [Apply a Dynamic Data View](#) for more details.

### Apply DVT and View Comparison Result with Golden Baseline

After a user applies a Data View Template to a map, the last copy of historical data and Golden Baseline is displayed. When the data doesn't match the Golden Baseline Rule, it will be highlighted in red color to alert



point that matches Golden Baseline Rule.

The screenshot shows the Golden Baseline Analysis interface for the table 'bgp\_nbrs'. The table contains 4 items. A 'Compare' button is highlighted with an orange box and an orange arrow. Below the table is a 'Compare Result' dialog box. The dialog box shows a comparison between two snapshots: '2019-06-11 10:39:16 AM (Map Data Time)' and '2019-05-30 07:56:25 AM (Golden Sample)'. The comparison result shows 4 modified items, which are highlighted in blue in the table below. The legend at the bottom of the dialog box indicates: 0 Added, 0 Removed, 4 Modified.

nrbr_id	remote_as	nrbr_type	version	remote_rid	state	up_time	last_read	last_write	holdtime
10.88.250.9	64553	external link	4	10.88.255.51	Established	20	00:00:43	00:00:39	180
10.88.250.27	64556	external link	4	10.88.7.4	Established	3	00:00:16	00:00:11	90
10.88.255.5	64550	internal link	4	10.88.255.5	Established	2	00:00:16	00:00:11	90
10.88.255.41	64554	external link	4	10.88.255.41	Established	2	00:00:16	00:00:11	90

- For table-type variables, only the table columns selected for comparison will be compared. The table columns for comparison are selected in the comparison setting of parser definition, and users also have a chance to temporarily adjust the desired columns to view the comparison result.

The screenshot shows the 'Parser Compare Settings' dialog box. The dialog box has a title bar 'Parser Compare Settings' and a close button. Below the title bar is a section 'Select variables to compare'. Under this section, the variable 'bgp\_nbrs' is expanded, and a list of variables is shown with checkboxes. The variables are: \$nrbr\_id, \$remote\_as, \$nrbr\_type, \$version, \$remote\_rid, \$state, \$up\_time, \$last\_read, \$last\_write, \$holdtime, \$keepalive, \$session\_active, and \$route\_refresh. The checkboxes for \$nrbr\_id, \$remote\_as, \$nrbr\_type, \$version, \$remote\_rid, \$state, \$up\_time, \$last\_read, \$holdtime, and \$keepalive are checked. The checkboxes for \$last\_write, \$session\_active, and \$route\_refresh are unchecked. At the bottom of the dialog box are 'OK' and 'Cancel' buttons.

## 2.4. Visualize Third-Party System Data through Single Pane of Glass

With NetBrain, network data from different sources and third-party systems can be visualized on a single map, which enables a map to serve as a single pane of glass (SPOG) for all valuable network data. In most SPOG scenarios, either the link to access a third-party webpage appears in the drill-down action list, or the detailed information retrieved from third-party systems via APIs is displayed in a table format with generated URL. With one click, users can directly go to the predefined webpage for further investigations and operations.

In previous versions, the only way to define a SPOG URL was to manually add it to a data view template, which was static and labor-consuming and lacks extension capabilities. Iev8.0 introduces the definition of SPOG URLs to resolve the limitation. Various variables and combinations are supported in SPOG URLs, including customized variables, parser variables, built-in variables (properties in GDR), and generic variables. Once defined, SPOG URLs can be directly referenced in all dynamic data views and will appear in the drill-down action list if the predefined conditions are met.

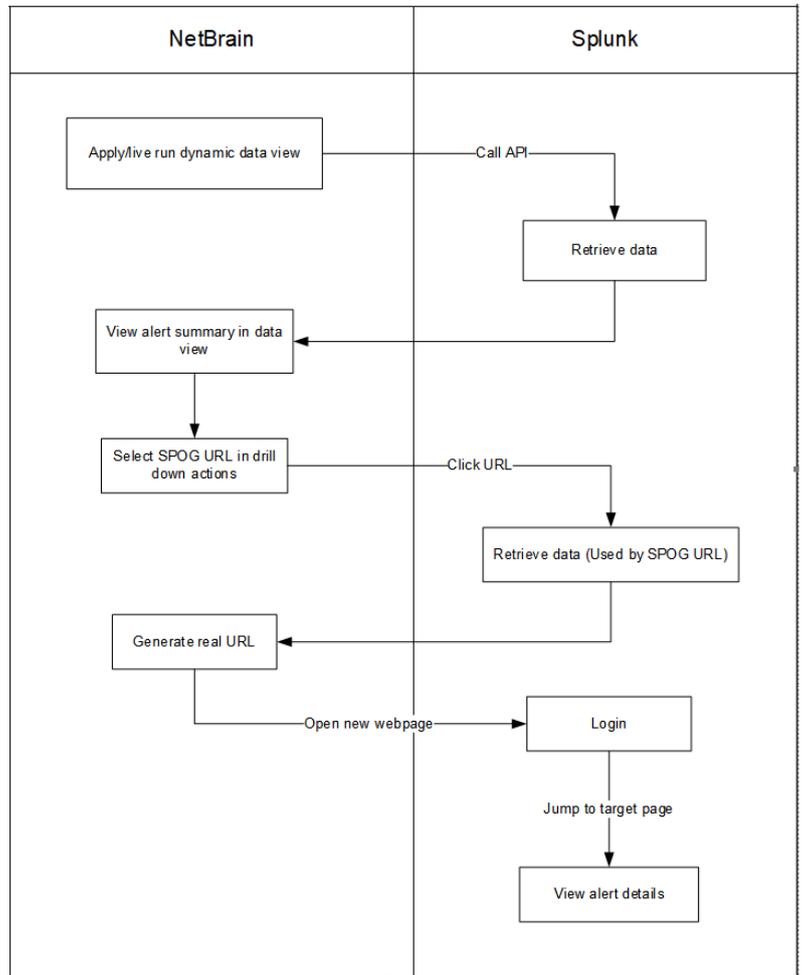
The screenshot displays the NetBrain interface. At the top, there is a search bar and navigation tabs for 'Map2', 'Page 1', 'Data View', and 'Runbook'. The main area shows a network map with three nodes: GW2Lab, BJ-R1, and BJ-R2. GW2Lab is connected to BJ-R1 (80/1 172.24.10.2/30) and BJ-R2 (10/0 172.24.10.10/29). BJ-R1 is also connected to BJ-R2 (80/0 172.24.10.9/29). A 'Recommended Action (1)' box is overlaid on the map, highlighting 'Device\_Status (PRTG)'. An orange arrow points from this box to a PRTG monitoring dashboard. The dashboard shows a gauge chart and a table of data. The sidebar on the left contains a search bar and a list of data view templates, including 'Display STP Detail', 'InputTest', 'position condition parser', 'support variable - compound action', 'support variable - compound case', 'support variable - device built-in data', 'support variable - device parser api', 'support variable - input variable', 'support variable - input variable data', 'support variable - interface built-in data', 'support variable - interface built-in ref', 'support variable - interface compound', 'support variable - interface parser', 'test', 'test1', 'Run Mode', 'Cache', 'Live(Run once)', 'Live2(Frequency)', 'zlh', 'om demand', 'Original - dvt', and 'My Templates'.

## Reference Flow



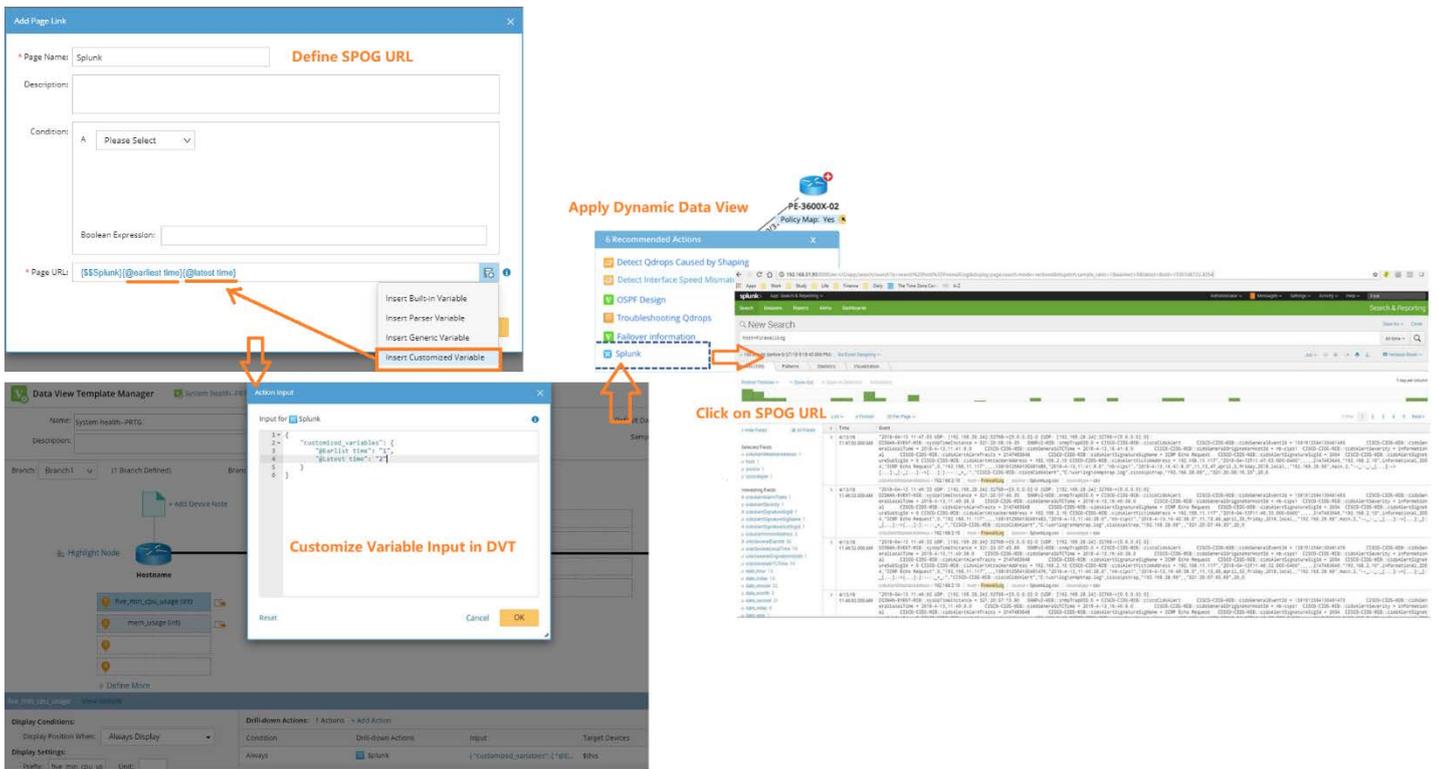
## Key Use Case

End users apply a dynamic data view (including API parser variable from Splunk) to a map, and the event information retrieved via APIs will be displayed in a table-type data unit. Click on the Event ID in a specific cell of the table to generate the SPOG URL. Then end users will be redirected to the target Splunk website for event details.



### 2.4.1. Define and Apply a SPOG URL

Take how to visualize a SPOG URL for Splunk firewall log search in a data view for example.

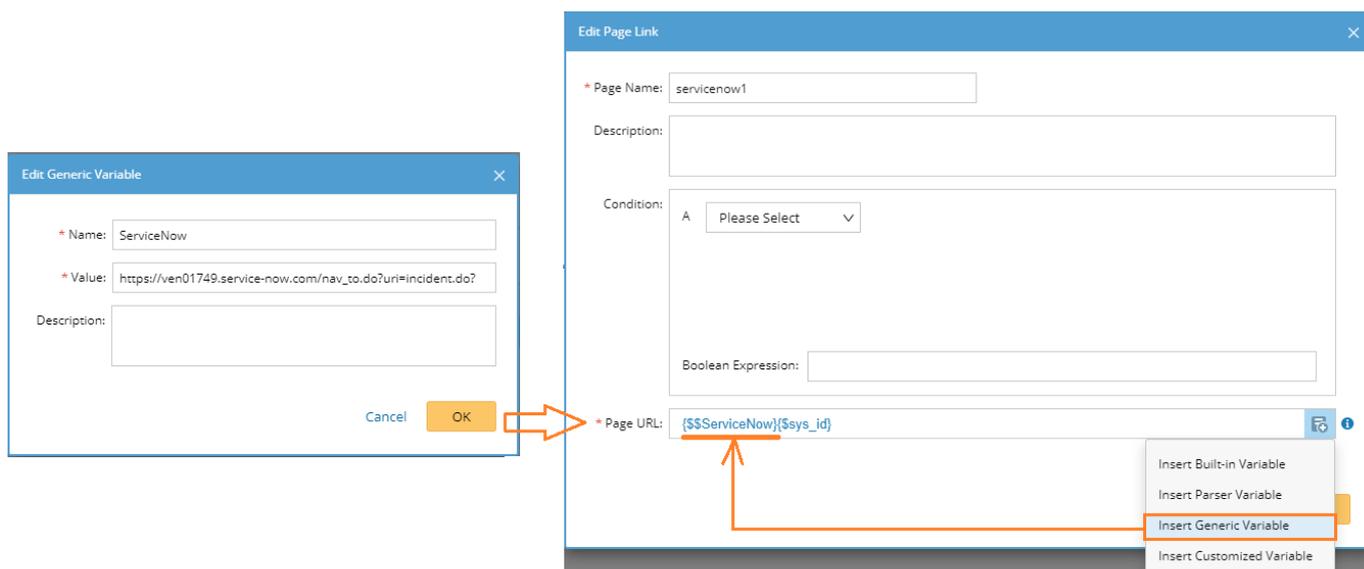


1. During the SPOG URL definition process, users may have no idea about the earliest and latest time for the search period. So, they can compose the URL by inserting two customized variables “earliest time” and “latest time” as placeholders.
2. During specific DVT definition process, assign values to “earliest time” and “latest time” in the input script.
3. Apply the Dynamic Data View (template), and the SPOG will appear in the drill-down action list.
4. Click on the SPOG URL to go to the Splunk page for the target firewall log search.

## 2.4.2.Pre-Define a Generic Variable for SPOG URL

To make common and static resources (such as a base URL) easy to reuse and maintain in SPOG URL management, users can pre-define generic variables before defining a SPOG URL.

For example, add "<https://ven01749.service-now.com/>" as a generic variable. Then it can be directly inserted when defining SPOG URLs for ServiceNow.



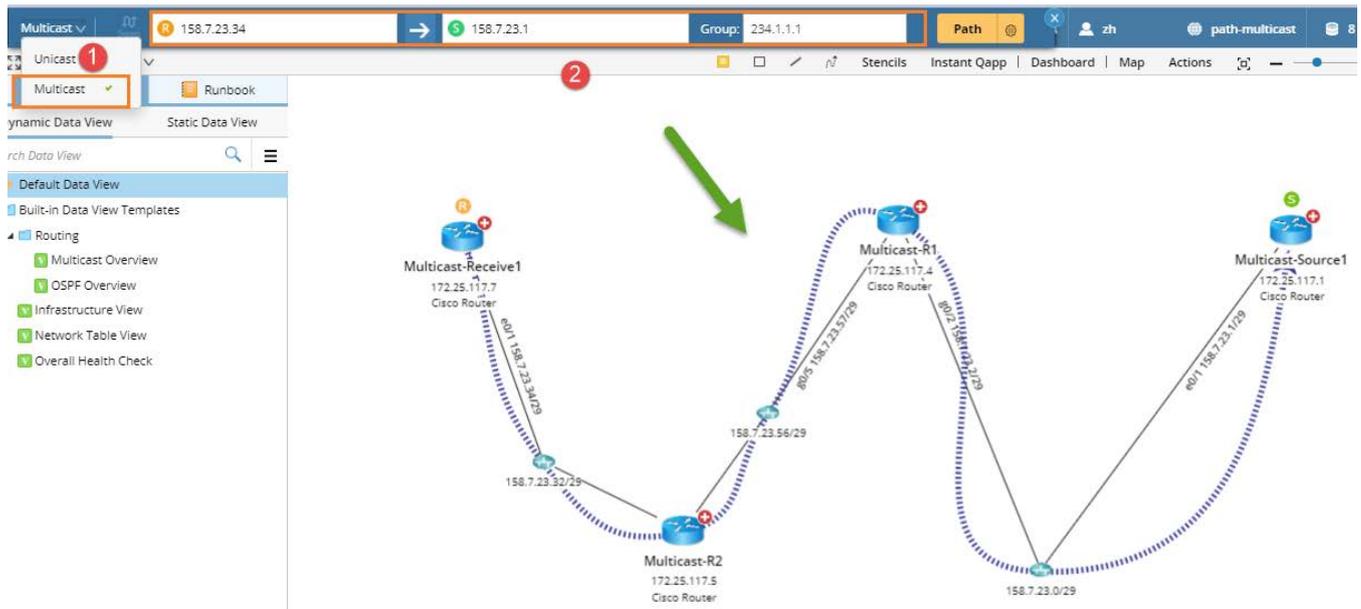
## 2.5. New Multicast Reverse Path Calculation

### 2.5.1. Calculate Multicast Reverse Path

When a problem occurs in a multicast network, users often need to troubleshoot the connection from a receiver to a source based on a specific group. In the previous versions, the system helped users troubleshoot multicast issues by drawing multicast paths via Qapps.

The system has extended the path framework, which enables users to calculate multicast paths. End users can specify a multicast group, source IP, and receiver IP to calculate a unidirectional multicast path from a receiver to a source.

A multicast reverse path includes four parameters: receiver, source, group, and gateway.



## Multicast Path Calculation Logic

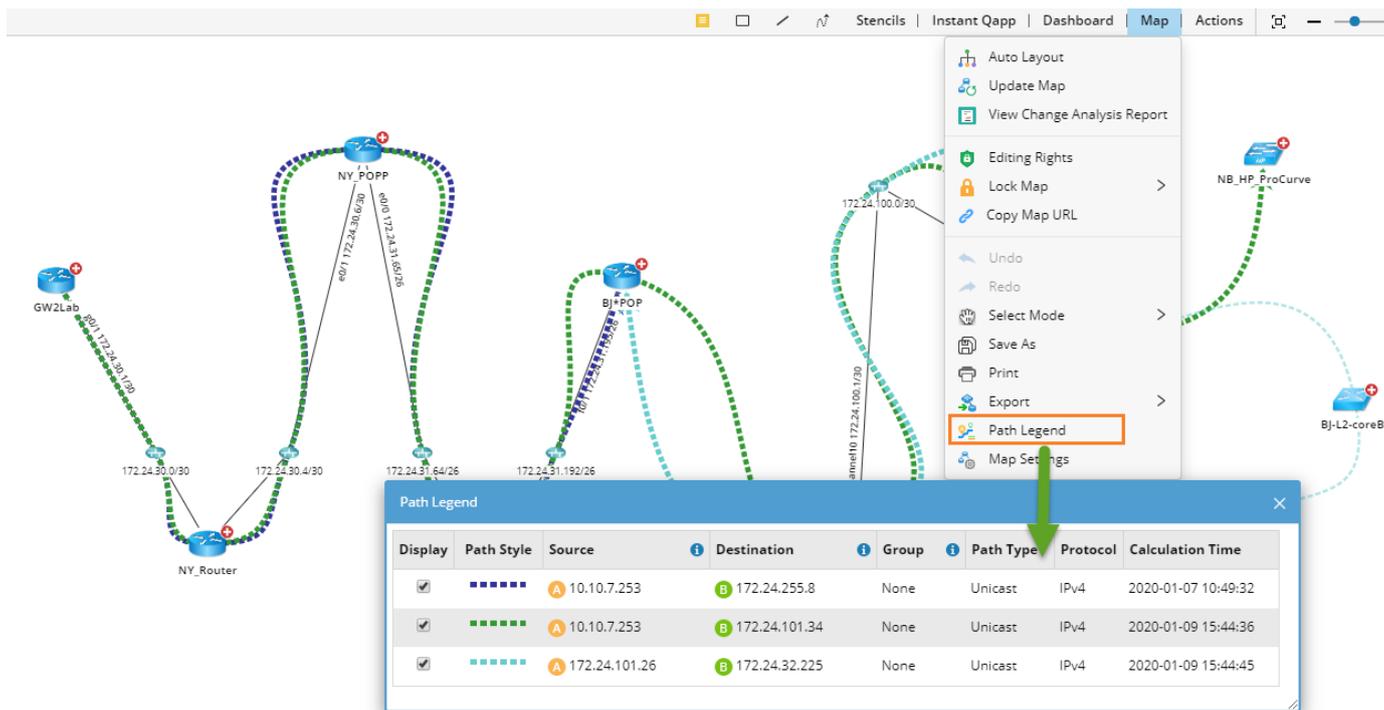
Compared with a unicast path, the following calculation logics are added for a multicast path.

- The system adds an NCT “Multicast Route Table” and uses it to look up L3 next-hop devices during a path calculation.  
**Note:** For the first-hop device, the system looks up its next-hop device based on the original routing table.
- When checking ACL/Policy on interfaces, the system checks whether the group IP as a destination is matched and continues path calculation based on the matching result.

### 2.5.2.View Path Legend on Map

IEv8.0 introduces the Path Legend function to differentiate paths on a map based on styles and types.

By default, the Path Legend pane is invisible. To view path legends, click **Map > Path Legend**.



### 2.5.3. Gateway for Multicast Reverse Path

IEv8.0 uses the unicast path gateway logic to look up the gateway for a multicast path, but adds filter mechanism to the logic:

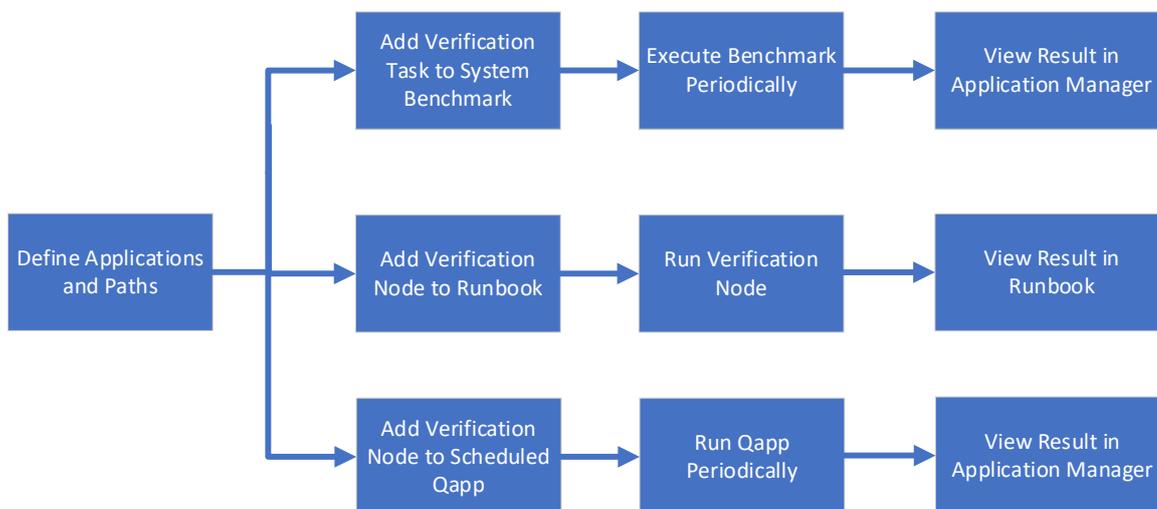
- If a candidate gateway is not configured with multicast, it will be removed from the gateway list.
- If devices associated with a particular technology such as HSRP, VXLAN anycast gateway, and VIP, is configured with multicast, they will be added to the gateway list.
- If the device that an input IP belongs to is a network device, the device will be added to the gateway list no matter it is configured with multicast or not.

## 2.6. Application Assurance Module

Application Assurance Module (AAM) feature is designed to manage paths based on applications and periodically verify the application paths to detect network changes.

**Note:** Like Change Management, AAM is a feature that requires separate subscription. Only users who have subscribed to this feature can use the related functions.

## Reference Flow



1. [Verify Application Paths in Benchmark](#)
2. [Verify Application Paths in Runbook](#)
3. [Monitor Application Paths in Qapp Scheduler](#)

### 2.6.1. Manage Applications and Paths

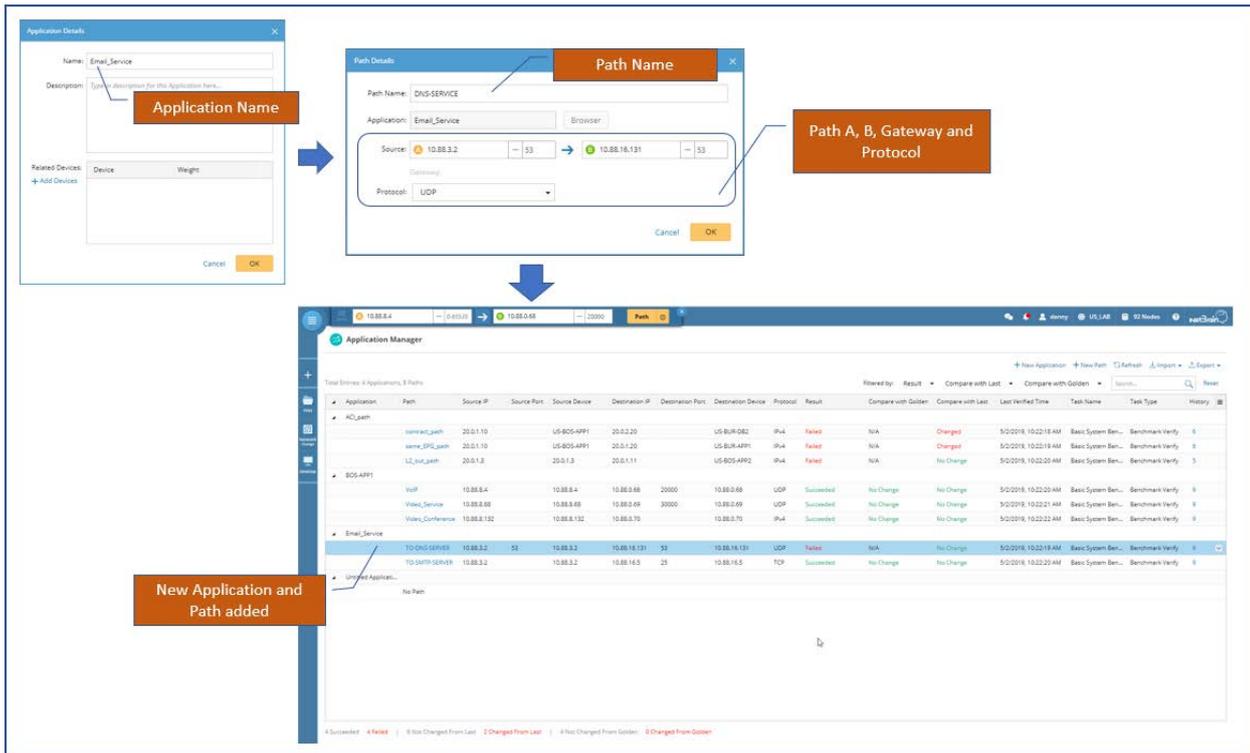
The Application Manager is used to manage applications and paths as well as to show the path verification results.

Application	Path	Source	Source Port	Destination	Destination Port	Protocol	Result	Compare with Golden	Compare with Last	Last Verified Tim...	Task Name
Email Services (3)	Client to DNS	10.22.3.4		10.22.3.4		IPv4	Succeeded	N/A	Changed	09:24am, 2018/09/18	Manually Verified by John
	Client to SMTP	10.22.3.4	1050	10.22.3.4	8080	TCP	Failed	Changed	No Change	09:18am, 2018/09/18	Basic Benchmark
	POP3 to Client	10.22.31.4	9099	10.22.31.4	53	UDP	Succeeded	Changed	No Change	09:24am, 2018/09/18	HR Application Verified
Web Services (3)	External to Web					IPv4	Succeeded	Changed	Changed		nce Application
	Web to App	10			1050	TCP	Succeeded	No Change	No Change		ified by Tim
	Application to Database	10				IPv4	Failed	N/A	No Change		on Verified

## Define Applications and Paths

Before using the Application Manager, you need to first create applications and add paths to the corresponding applications in the Application Manager. For details on defining application and paths, refer to [online help](#).

**Example1:** Manually define applications and paths in Application Manager.

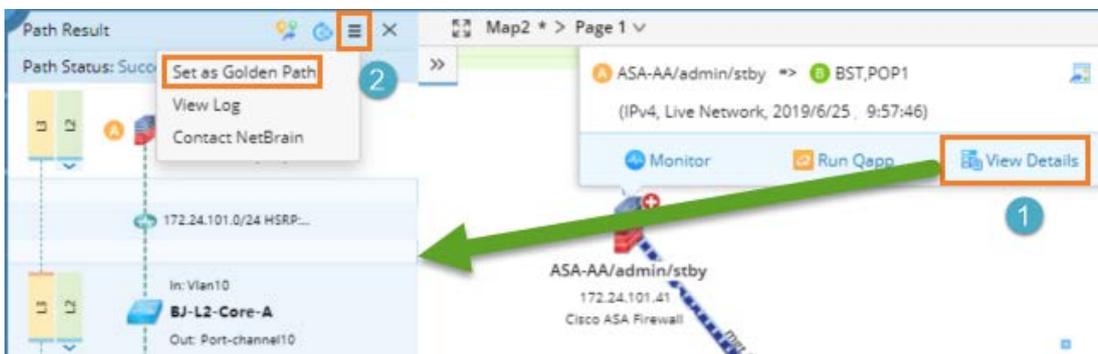


## Define Golden Paths

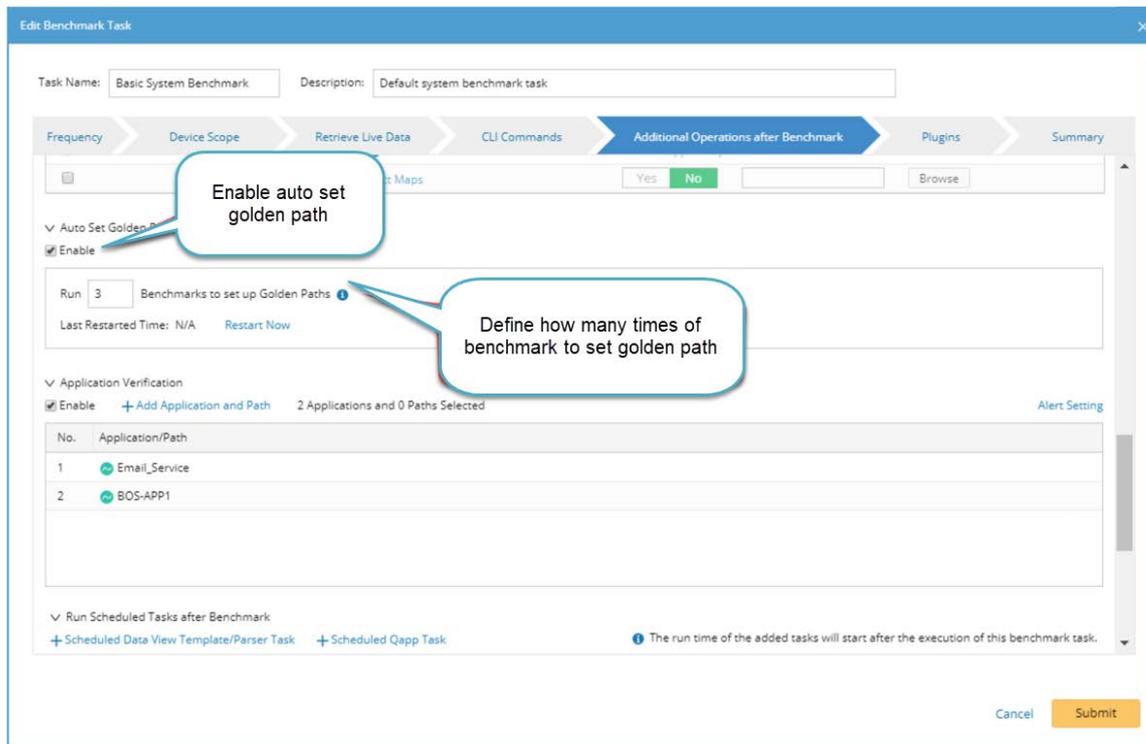
Golden Path refers to a calculated path in the NetBrain system that goes exactly as the traffic goes in a real network. You can set a golden path for a path calculation, and each time when the system periodically verifies or monitors the path, it will automatically compare the results with the Golden Path to detect changes.

A golden path can be set manually or through the Auto-Set Golden Path function that enables the system to automatically set a golden path by analyzing and comparing several consecutive path results. For details about how to define golden paths, refer to [online help](#).

**Example1:** Manually Define a Golden Path.



## Example2: Auto Set Golden Path in Benchmark.



**Note:** The auto-set golden path function is only supported in the benchmark function.

### The logic for Auto-Set Golden Path

The Auto-Set Golden Path function (Run “x” Benchmarks to set up a Golden Path) will auto set the last calculation result of a path as its Golden Path if the results of benchmarking the path continuously for “x” times are all successful and consistent. Otherwise, its Golden Path will be left empty.

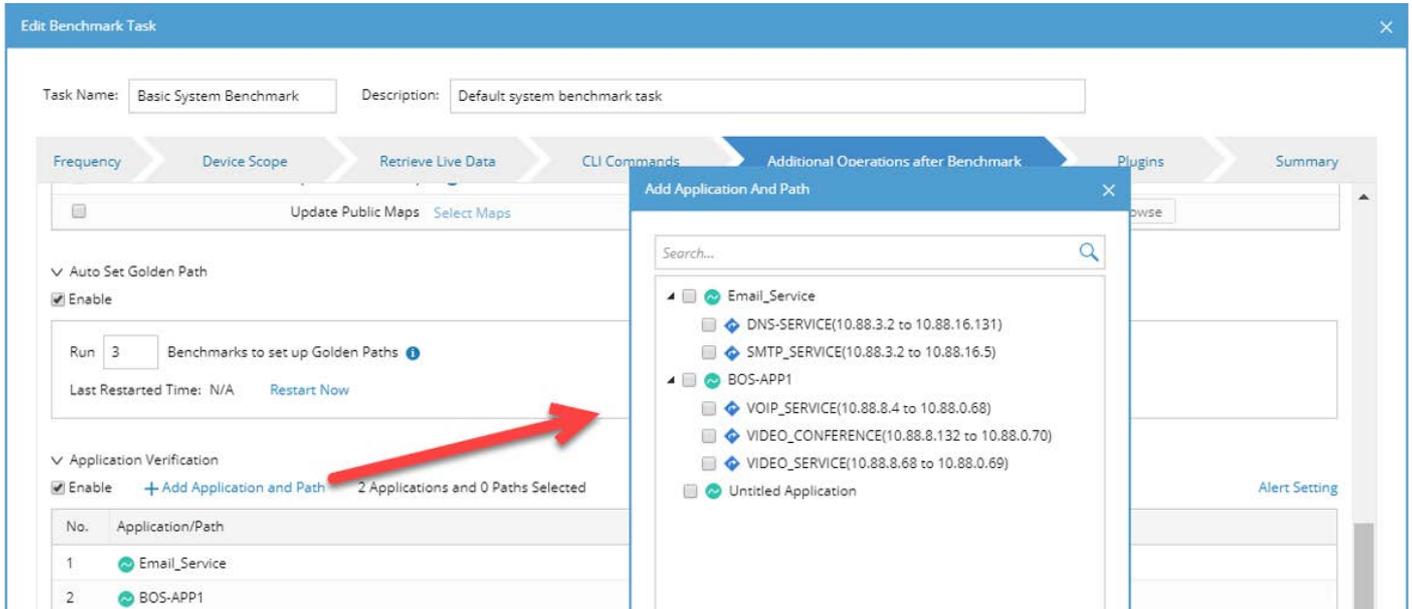
## 2.6.2. Verify Application Paths in Batch

You can set your NetBrain system to monitor or verify the application paths periodically and sent your alerts and emails when the paths change.

- [Verify Application Paths via a Benchmark Task](#)
- [Monitor Application Paths via Qapp Scheduler](#)
- [View Results](#)

## Verify Application Paths through Benchmark

You can add the application paths to a benchmark task. Each time when the benchmark is executed, the application paths will be verified accordingly.



**Note:** To improve the efficiency of path calculation, the system uses the data retrieved from the benchmark task to perform path calculation with cache data.

### Path Result Logic

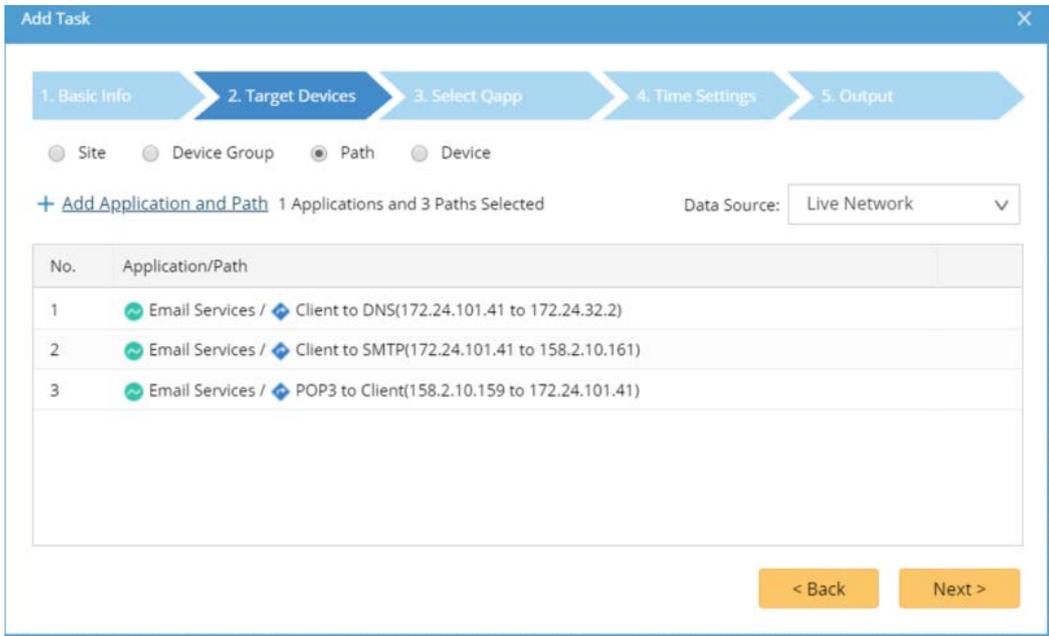
Path (Full Path) can be divided into Segmented Paths. In the Application Manager, the result of a Full Path is displayed as "Successful" or "Failed". If the Full Path is a load-balance path, only when all segmented paths are successfully calculated, the Full Path is considered as "successful"; otherwise, it is considered as "failed".

### The logic for comparing with the Last Path and Golden Path

When the system compares the current path with the Last Calculated Path and Golden Path, it will compare the device, incoming interface, and outgoing interface of each hop of the Full Path. If it is consistent, the system considers the path unchanged; otherwise, the path is considered changed.

# Monitor Application Paths via Qapp Scheduler

You can add the application paths to a Qapp Scheduler to periodically monitor the paths. Once the paths have changed, the system can detect changes and notify the changes via emails and alerts.

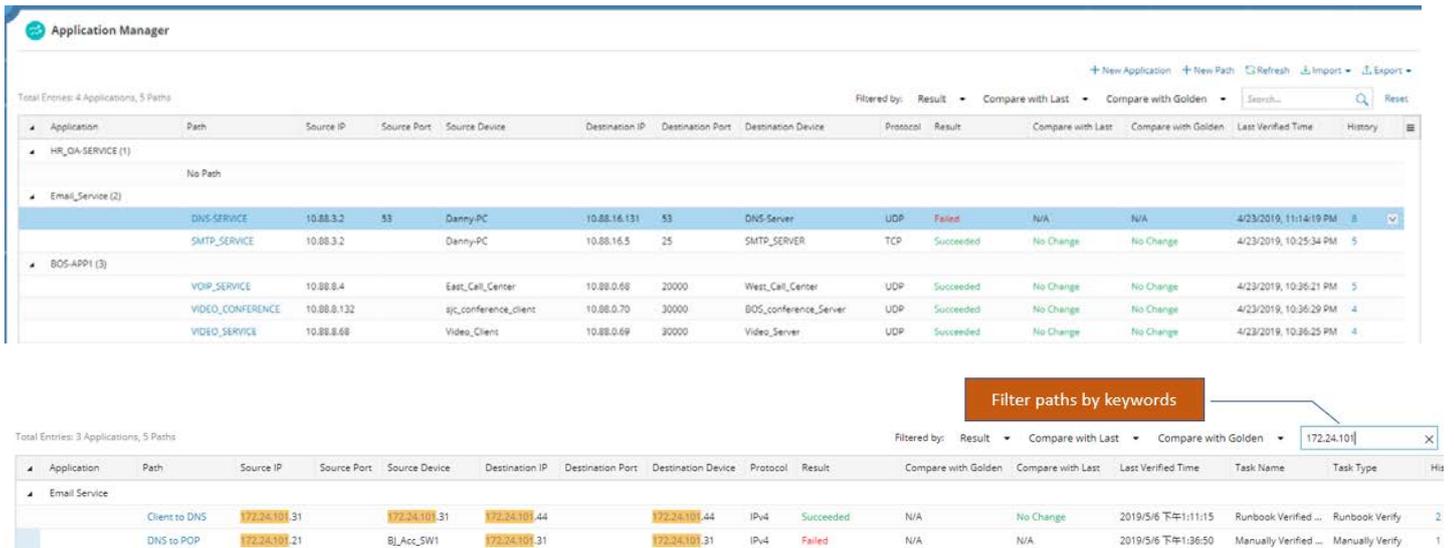


## View Results

When you monitor application paths in a Qapp Scheduler or benchmark, the verification results will be recorded and displayed in the Application Manager.

### View Latest Results

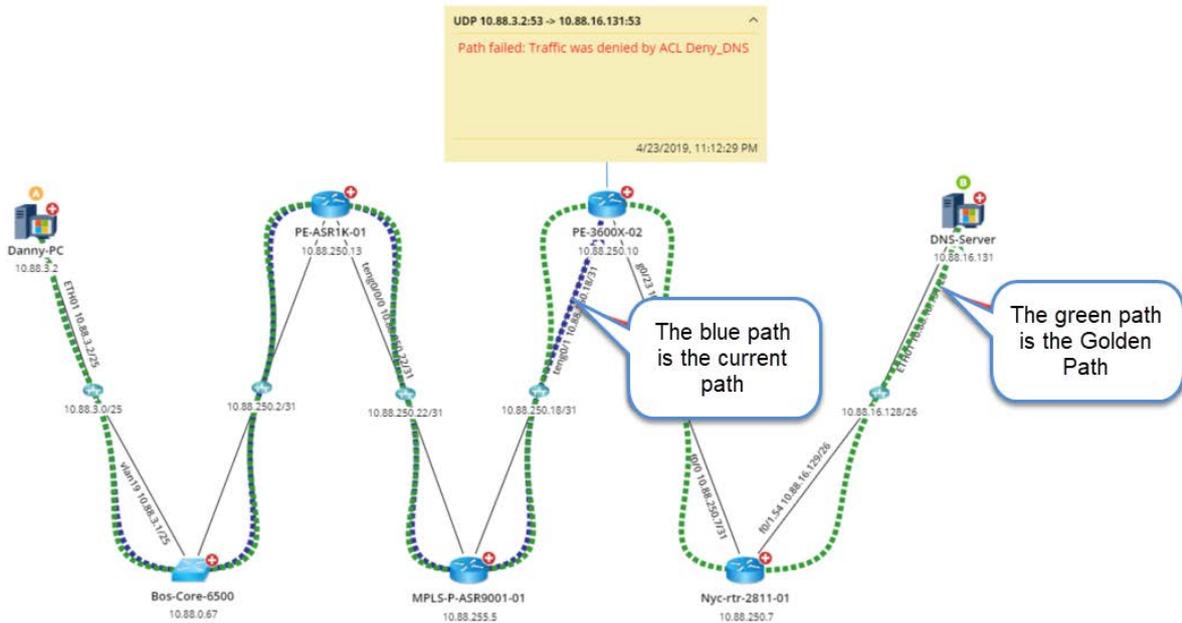
The Application Manager shows the latest results and supports a variety of filter methods to filter the result.



- Export the latest results in the Application Manager to a CSV file.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Application	Path	Source IP	Source Port	Source Device	Destination IP	Destination Port	Destination Device	Protocol	Result	Compare with Last	Compare with Golden	Last Verified Time	Problem Device
HR_OA-SERVICE													
Email_Service	DNS_SERVICE	10.88.3.2	53	Danny-PC	10.88.16.131	53	DNS-Server		Failed	N/A	N/A	4/24/2019 3:14	PE-3600X-02
Email_Service	SMTP_SERVICE	10.88.3.2		Danny-PC	10.88.16.5	25	SMTP_SERVER		Succeeded	No Change	No Change	4/24/2019 2:25	
BOS-APP1	VOIP_SERVICE	10.88.8.4		East_Call_Center	10.88.0.68	20000	West_Call_Center		Succeeded	No Change	No Change	4/24/2019 2:36	
BOS-APP1	VIDEO_CONFERENCE	10.88.8.132		sjc_conference_client	10.88.0.70	30000	BOS_conference_Server		Succeeded	No Change	No Change	4/24/2019 2:36	
BOS-APP1	VIDEO_SERVICE	10.88.8.68		Video_Client	10.88.0.69	30000	Video_Server		Succeeded	No Change	No Change	4/24/2019 2:36	
Untitled Application													

- Draw the current path and Golden Path on a map to view the differences.



## View History Results

The Application Manager stores the last calculated path and earlier history paths. You can open a history path on a map.

The screenshot illustrates the process of viewing history results. On the left, a table displays verification history with columns for 'Compare with Last', 'Compare with Golden', 'Last Verified Time', 'Task Name', 'Task Type', and 'History'. A callout box points to the 'History' column with the text 'Click the Number in History Column'. An arrow points to a 'History Path List' dialog box on the right, which shows a list of paths with columns for 'Last Verified', 'Result', and 'Last Verified Time'. A callout box points to the 'Open Path in Map' button in the dialog. Below the dialog, a network map shows the path between 'Danny-PC' and 'DNS-Server' through various routers, with a callout box pointing to the map and the text 'Map History Path'.

Compare with Last	Compare with Golden	Last Verified Time	Task Name	Task Type	History
No Change	No Change	4/23/2019, 5:32:58 PM	Manually Verified ...	Manually Verify	4
No Change	No Change	4/23/2019, 5:32:50 PM	Manually Verified ...	Manually Verify	3
No Change	No Change	4/23/2019, 5:32:41 PM	Manually Verified ...	Manually Verify	3
No Change	No Change	4/23/2019, 5:32:31 PM	Manually Verified ...	Manually Verify	2
No Change	No Change	4/23/2019, 5:32:21 PM	Manually Verified ...	Manually Verify	2

Last Verified	Result	Last Verified Time	Set as Golden Path
Success	Success	4/23/2019, 10:46:47 PM	(P)
Success	Success	4/23/2019, 10:25:20 PM	(P)
Success	Success	4/23/2019, 5:32:55 PM	(P)
Success	Success	4/23/2019, 5:31:29 PM	(P)
Success	Success	4/23/2019, 5:30:45 PM	(P)
Success	Success	4/23/2019, 5:26:14 PM	(P)

## View Verification Results via Emails

The Application Manager supports sending verification results to specified users via emails.

1. Set Email Sending in the AAM Alert Settings of Qapp Scheduler or Benchmark.

The screenshot shows the 'Alert Settings' dialog box. The 'Share Alert with:' field contains 'Eddy'. The 'Send Email to:' field contains 'admin@netbrain.com', with a callout box pointing to it and the text 'Add Email Address'. The 'Error' and 'Warning' checkboxes are checked. The 'Cancel' and 'OK' buttons are visible at the bottom.

## 2. View the verification result in Email.

[NetBrain]Application Path Verification Result 2019-04-09.csv  
1 KB

CSV Report Attached

Hi admin,

Application path verification results for 2019-04-09 are below:  
Task Type: Server Benchmark  
Task Name: TestApp03  
Execution Time: 2019-04-09 03:21:07 +08:00  
Total Entries: 2 Applications, 5 Paths  
3 paths succeeded;  
2 paths failed;  
1 paths changed with Golden Path;  
0 paths changed with Previous Path.

You can see the details in [Application Manager](#).

The paths that failed or changed are listed here:

Application	Path	Source IP	Source Port	Source Device	Destination IP	Destination Port	Destination Device	Protocol	Result	Compare with Golden	Compare with Last	Last Verified Time	Task Name
lhx	path1	10.10.0.23	1	10.10.0.23	172.24.31.195	2	BJ*POP	TCP	Failed	Changed	No Change	2019-04-09 11:22:37 +08:00	TestApp03
lhx	SearchPath30001	10.10.0.29		10.10.0.29	172.24.31.195		BJ*POP	IPv4	Succeed	No Change	No Change	2019-04-09 11:22:39 +08:00	TestApp03
lhx	path2	10.10.10.2	1	qapp-c3560-2	172.24.30.6	1	NY_POPP	TCP	Failed	N/A	No Change	2019-04-09 11:22:50 +08:00	TestApp03
lhx	TestBug55407	10.10.0.23		10.10.0.23	10.10.0.24		10.10.0.24	IPv4	Succeed	No Change	No Change	2019-04-09 11:22:37 +08:00	TestApp03
NB IE	TestBenchmark0001	172.24.30.2		NY_Router	172.24.31.125		NY-core-bak	IPv4	Succeed	No Change	No Change	2019-04-09 11:22:38 +08:00	TestApp03

Summary Information

Table Report

### 2.6.3. Verify Application Paths via Runbook Automation

The Runbook in IEv8.0 adds a **Verify Application** node to enable you to verify applications and paths related to devices on a map via runbook automation.

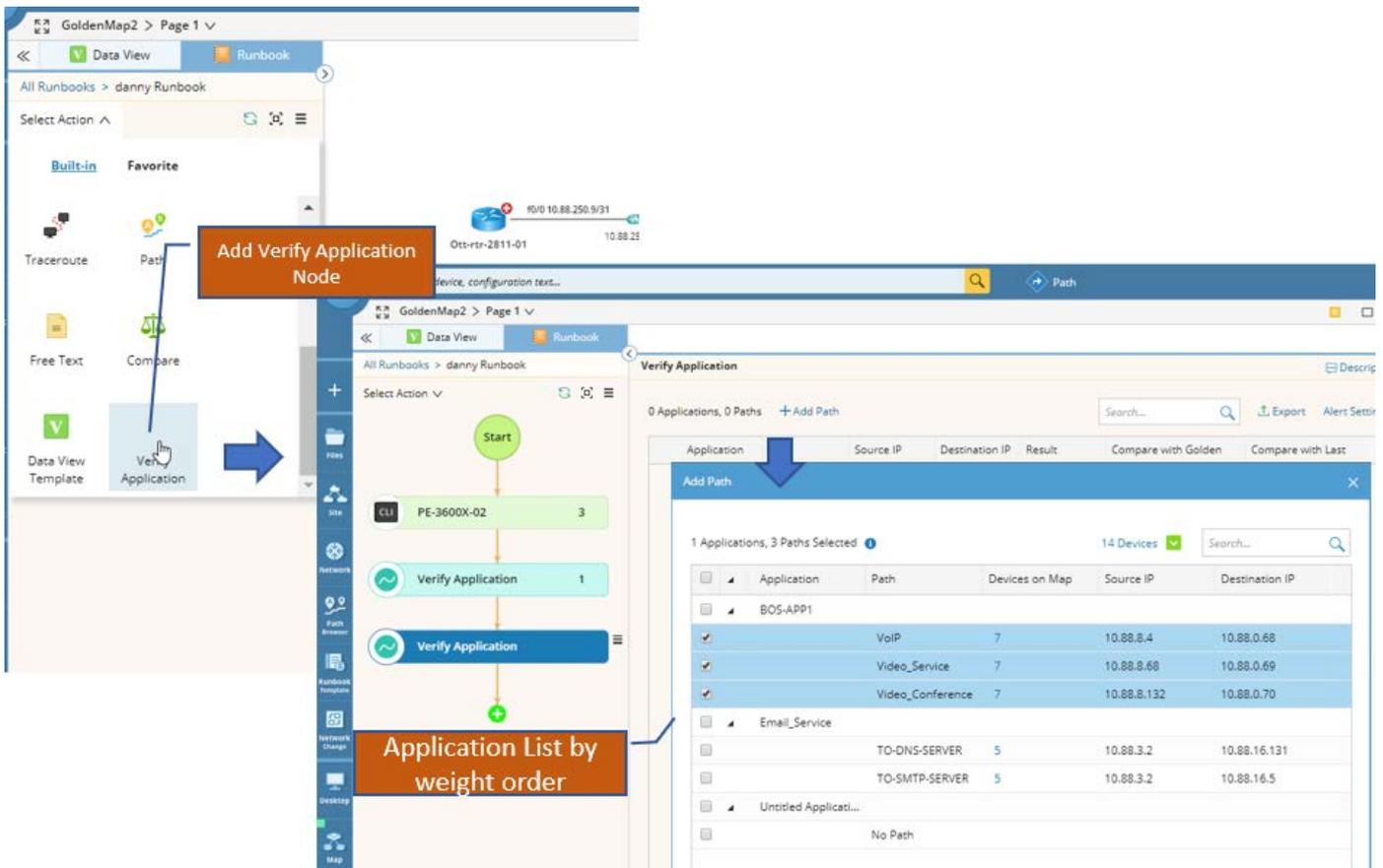
**Note:** Only users who have purchased the AAM feature can see and use the Verify Application Node in the Runbook Action.

#### Key Use Case

- In a network change flow, users can define which applications need to be verified after the network change through the runbook node and ensure that the network change does not affect these critical applications.
- During troubleshooting, users can verify application paths via Runbook automation, and the system can automatically filter out the application paths related to devices on a map.

#### Add “Verify Application” Node

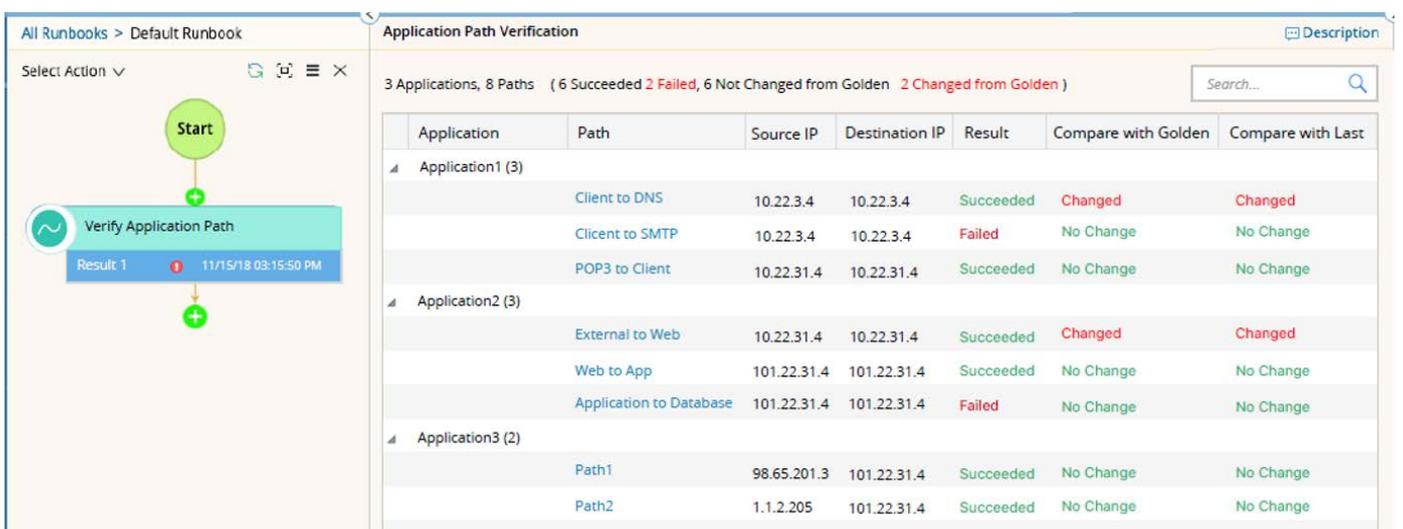
On a map, open the default runbook, add a **Verify Application** node to the runbook and then click Add Paths to add applications and paths that you want to verify.



The applications in the Add Path pane are sorted and ordered based on application weight. For details on the rules about various weights, refer to the [Appendix](#).

## Execute "Verify Application" Node

After verifying the application paths in the runbook, you can view the verification results in the **Application Path Verification** Pane.

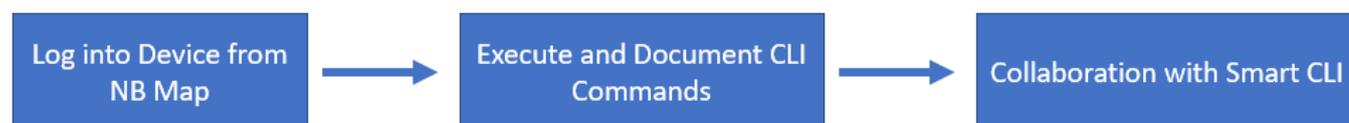


## 2.7. Smart CLI

When network problems occur, most engineers usually use Telnet/SSH terminals to connect devices and type commands to troubleshoot the problems. NetBrain Smart CLI is a Telnet/SSH client for the Windows platform.

NetBrain Smart CLI can retrieve device credentials from NetBrain IE and automatically log in to devices to execute CLI Command. Furthermore, it can send CLI results to Map and Runbook for documentation and analysis.

### Reference Flow



- [Auto Login to Device](#)
- [Document CLI Results](#)
- [Collaboration Between Multiple Smart CLI Consoles](#)

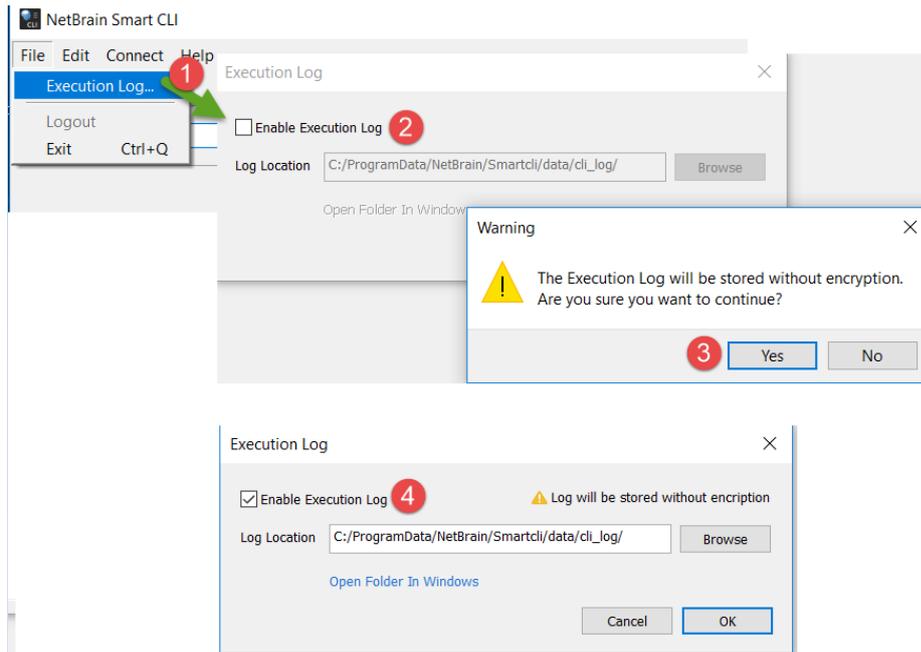
### 2.7.1. Basic Functions of Smart CLI Tool

The Smart CLI Tool is a rich client that can be installed independently on the Windows operating systems and provide CLI interaction like other Telnet/SSH tools.

The basic functions of this tool are as follows:

- Create a Telnet/SSH connection to Network Devices.
- Save and manage the connections created manually by users.

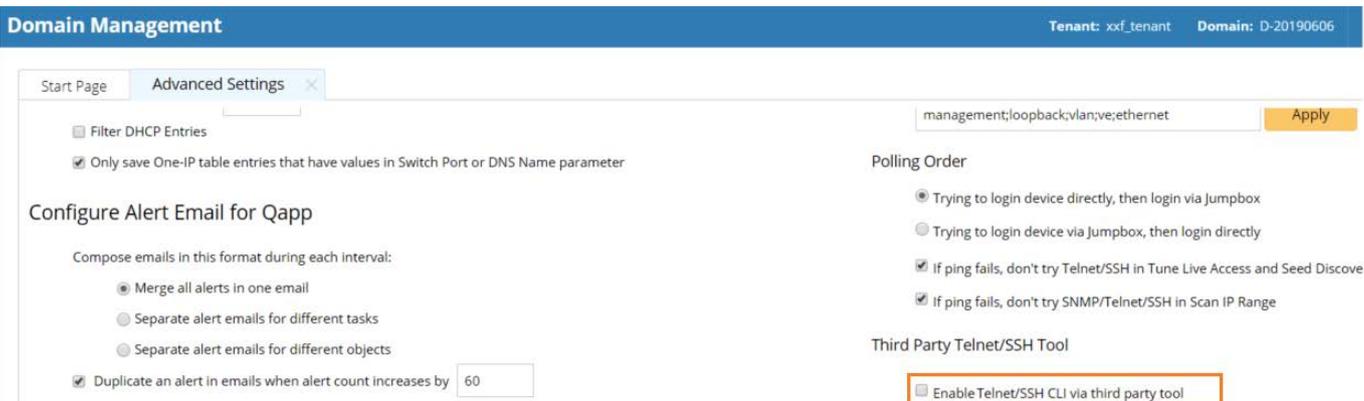
- Record the execution logs of all login devices. The log recording function is disabled by default for security considerations because the logs are saved in a clear text.



**Note1:** The Windows operating systems that the tool supports are as follows:

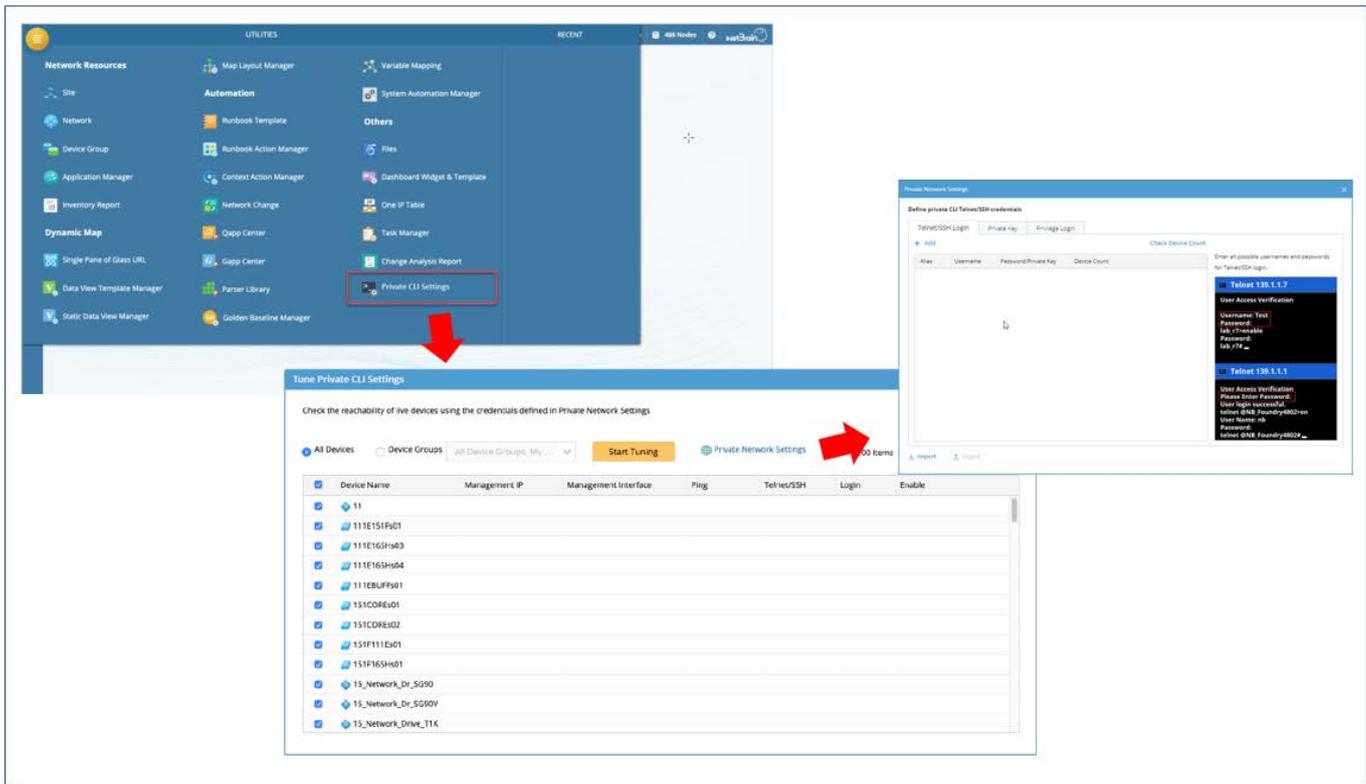
- Windows 10
- Windows Server 2012 (Standard/Datacenter Edition)
- Windows Server 2012 R2 (Standard/Datacenter Edition)
- Windows Server 2016 (Standard/Datacenter Edition)
- Windows Server 2019 (Standard/Datacenter Edition)

**Note2:** Both Smart CLI and Telnet/SSH CLI are remote connection tools in the system, which may confuse users for choices. IEv8.0 adds an option to enable/disable the Telnet/SSH CLI feature. By default, this option is disabled.



## 2.7.2.Auto Login to Device

Before logging to devices with the Smart CLI (without requiring the Domain Management privilege), add and tune the Private CLI settings.

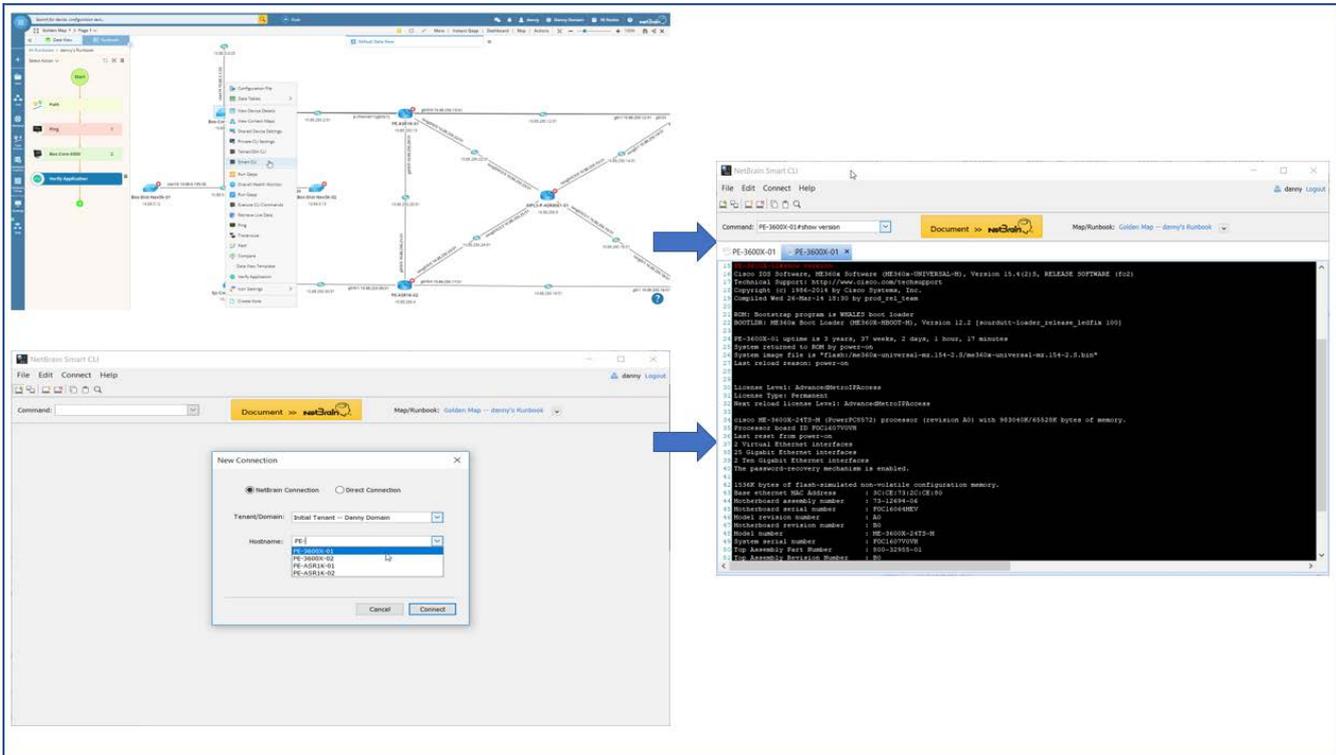


Then you can access a network device via Smart CLI with configured credentials after logging into NetBrain IE.

- [Auto Login to Device from NetBrain Map](#)
- [Auto Login to Device from Smart CLI Terminal](#)

## Auto Login to Device from NetBrain Map

When you try to log into a device by using Smart CLI from a NetBrain Map, the Smart CLI Tool will retrieve the credentials of the device to log into the device.



**Note:** The Auto Login to Device function supports login script as well as the login via jump box.

## Auto Login to Device from Smart CLI Window

After logging into NetBrain IE from the Smart CLI window, Smart CLI supports searching a device through the hostname and management IP and auto login to the device.

**Note:** The prerequisite of auto login is that you have logged in to NetBrain IE and the device exists in the target domain.

### 2.7.3.Document CLI Results

You can save a CLI command result to NetBrain Runbook and then apply parsers or comparisons to detect network changes.

- [Save CLI Command Result to Runbook Node](#)
- [Compare CLI Command Node to Address Changes](#)
- [Generate Map Note from the Selected Result](#)

## Save CLI Command Result to Runbook Node

You can send the result of a CLI Command for a device to a specified map and runbook in NetBrain IE. The Runbook will automatically generate a corresponding CLI Command node.

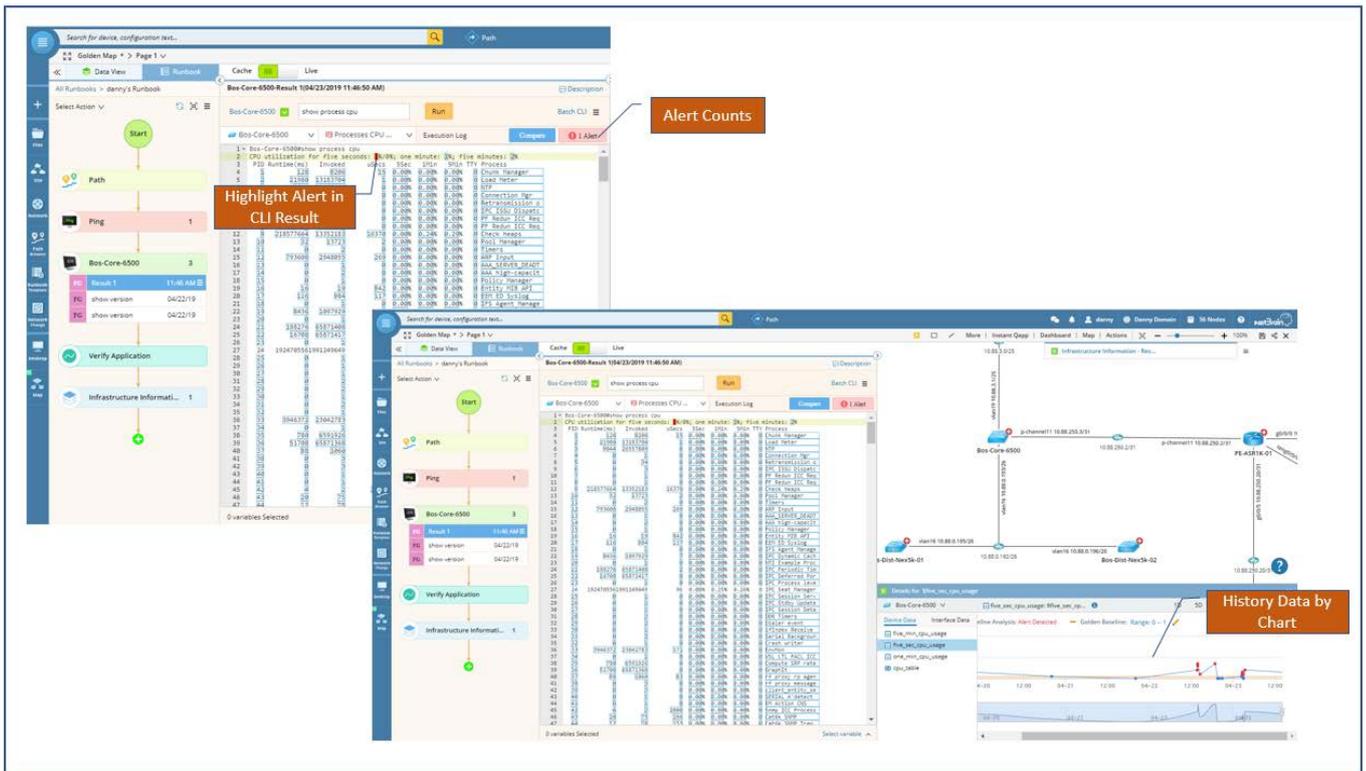
The image illustrates the process of saving CLI command results to a NetBrain Runbook. It is divided into three main sections:

- Top Left:** A screenshot of the NetBrain Smart CLI interface. A CLI command is entered: `PE-3600X-01 show version`. The output shows device details for PE-3600X-01, including software versions and hardware information. A blue arrow points from this output to the 'Document to IE' dialog.
- Top Right:** A 'Document to IE' dialog box. It allows selecting a 'Map/Runbook' (Golden Map - danny's Runbook) and a 'Device' (PE-3600X-01). A 'Send to NetBrain IE' button is visible at the bottom.
- Bottom:** A screenshot of the NetBrain IE Runbook interface. The 'CLI Command' node is highlighted, showing the command `PE-3600X-01 show version` and its corresponding output. A 'CLI Command Node and Result' label points to this node. The background shows a network map with various nodes and connections.

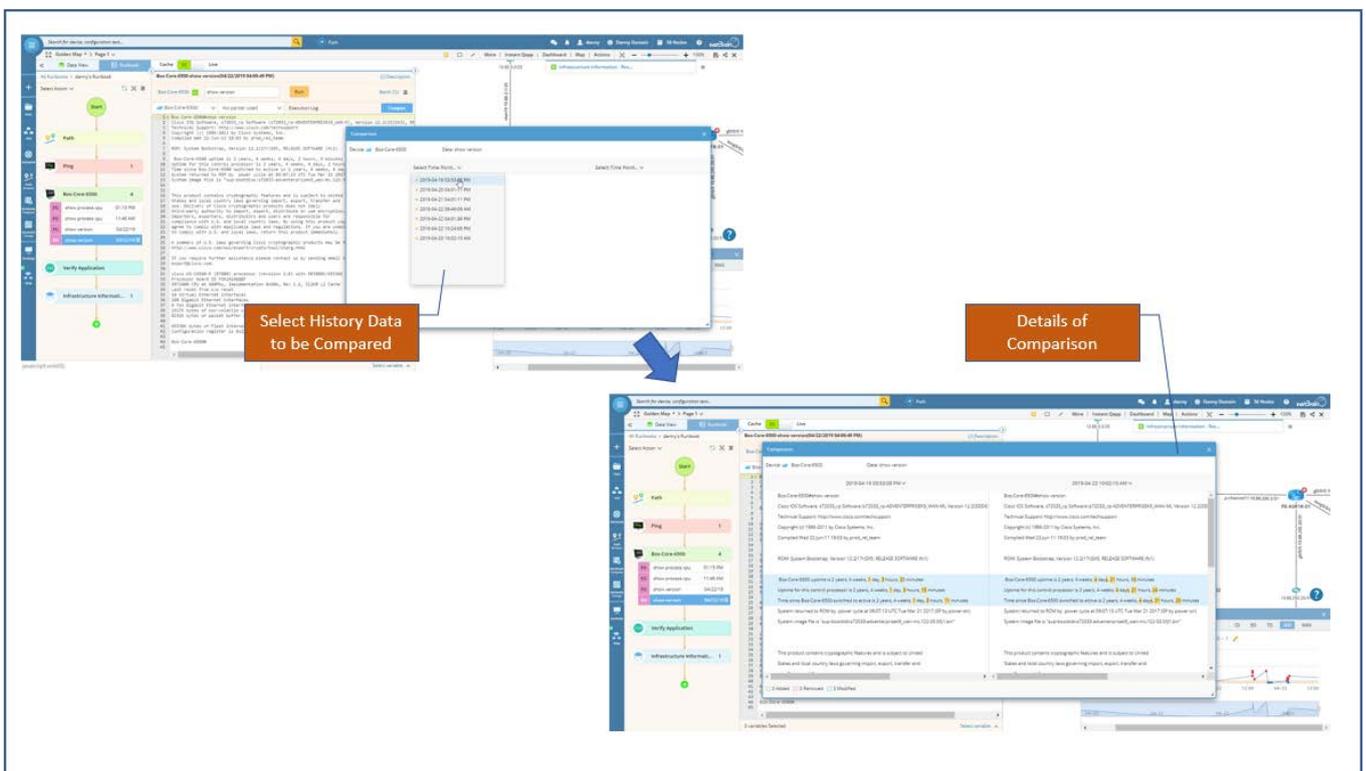
## Compare CLI Command Node to Address Changes

In the CLI command node of the Runbook, you can compare the CLI command result with NetBrain IE history results and Golden Baseline and view history details.

1. Compare the current CLI result with Golden Baseline and highlight alerts in the CLI result. You can view the history data of each changed variable.



2. Compare the current CLI result with history results to see what has changed.



# Generate Map Note from the Selected Result

You can select a CLI Command result in the Smart CLI window and send it to a specified map in NetBrain IE to generate a map note.

**Select CLI result and Click note icon**

**Add note here**

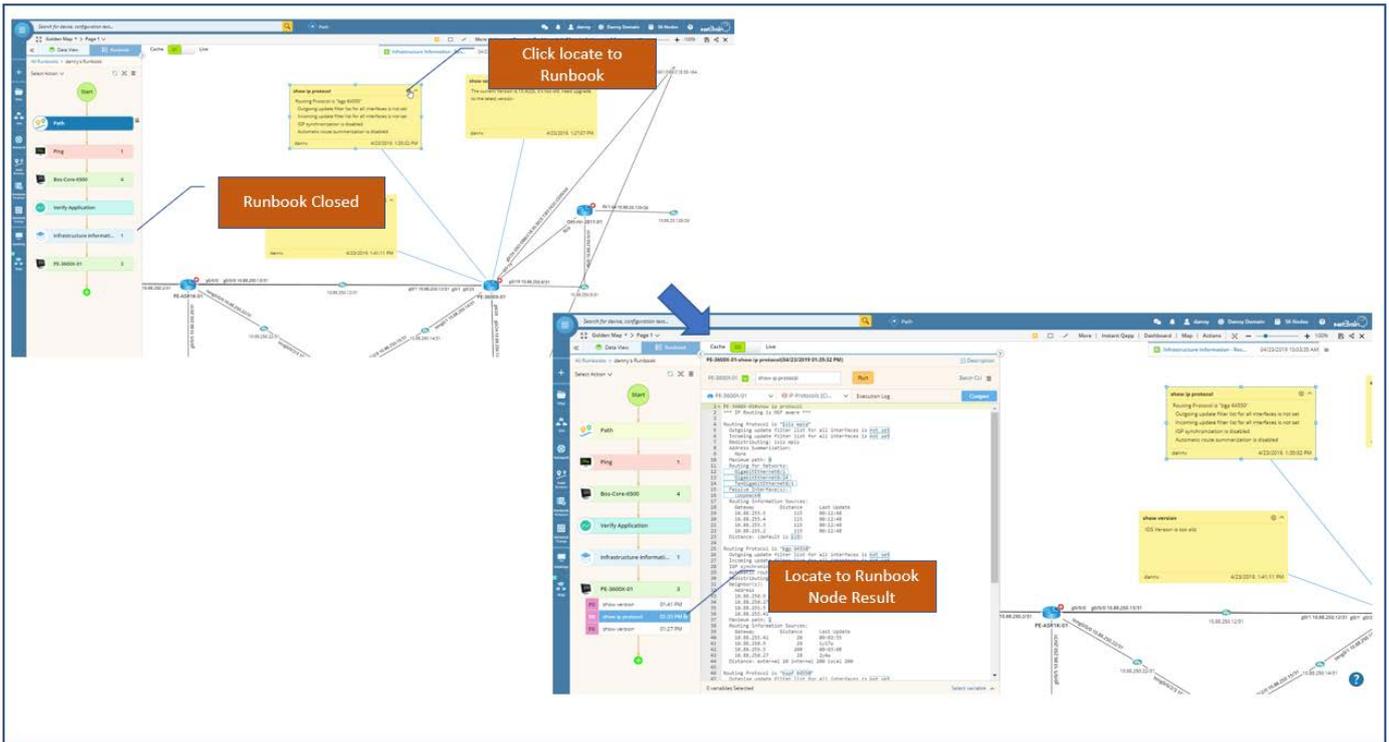
**Enable add map note**

**CLI result to be document**

**Draw map note on map**

```
PE-3600X-01 show ip protocols
1:
2: IP Routing is ON for all interfaces
3: *** IP Routing is NSF aware ***
4: Routing Protocol is "ISIS ISIS"
5: Outgoing update filter list for all interfaces is not set
6: Incoming update filter list for all interfaces is not set
7: Automatic route summarization is disabled
8: Redistribution: connected, static
9: Address Summarization:
10:  None
11: Maximum paths: 1
12: Routing for Services:
13:  GigabitEthernet0/1
14:  GigabitEthernet0/24
15:  Tunnel-Interface0/1
16: Passive Interface(s):
17:  Loopback0
18:
19:
20:
21:
22:
23:
24:
25:
26:
27:
28:
29:
30:
31:
32:
33:
34:
35:
36:
37:
38:
39:
40:
41:
42:
43:
44:
45:
46:
47:
48:
49:
50:
51:
52:
53:
54:
55:
56:
57:
58:
59:
60:
61:
62:
63:
64:
65:
66:
67:
68:
69:
70:
71:
72:
73:
74:
75:
76:
77:
78:
79:
80:
81:
82:
83:
84:
85:
86:
87:
88:
89:
90:
91:
92:
93:
94:
95:
96:
97:
98:
99:
100:
101:
102:
103:
104:
105:
106:
107:
108:
109:
110:
111:
112:
113:
114:
115:
116:
117:
118:
119:
120:
121:
122:
123:
124:
125:
126:
127:
128:
129:
130:
131:
132:
133:
134:
135:
136:
137:
138:
139:
140:
141:
142:
143:
144:
145:
146:
147:
148:
149:
150:
151:
152:
153:
154:
155:
156:
157:
158:
159:
160:
161:
162:
163:
164:
165:
166:
167:
168:
169:
170:
171:
172:
173:
174:
175:
176:
177:
178:
179:
180:
181:
182:
183:
184:
185:
186:
187:
188:
189:
190:
191:
192:
193:
194:
195:
196:
197:
198:
199:
200:
201:
202:
203:
204:
205:
206:
207:
208:
209:
210:
211:
212:
213:
214:
215:
216:
217:
218:
219:
220:
221:
222:
223:
224:
225:
226:
227:
228:
229:
230:
231:
232:
233:
234:
235:
236:
237:
238:
239:
240:
241:
242:
243:
244:
245:
246:
247:
248:
249:
250:
251:
252:
253:
254:
255:
256:
257:
258:
259:
260:
261:
262:
263:
264:
265:
266:
267:
268:
269:
270:
271:
272:
273:
274:
275:
276:
277:
278:
279:
280:
281:
282:
283:
284:
285:
286:
287:
288:
289:
290:
291:
292:
293:
294:
295:
296:
297:
298:
299:
300:
301:
302:
303:
304:
305:
306:
307:
308:
309:
310:
311:
312:
313:
314:
315:
316:
317:
318:
319:
320:
321:
322:
323:
324:
325:
326:
327:
328:
329:
330:
331:
332:
333:
334:
335:
336:
337:
338:
339:
340:
341:
342:
343:
344:
345:
346:
347:
348:
349:
350:
351:
352:
353:
354:
355:
356:
357:
358:
359:
360:
361:
362:
363:
364:
365:
366:
367:
368:
369:
370:
371:
372:
373:
374:
375:
376:
377:
378:
379:
380:
381:
382:
383:
384:
385:
386:
387:
388:
389:
390:
391:
392:
393:
394:
395:
396:
397:
398:
399:
400:
401:
402:
403:
404:
405:
406:
407:
408:
409:
410:
411:
412:
413:
414:
415:
416:
417:
418:
419:
420:
421:
422:
423:
424:
425:
426:
427:
428:
429:
430:
431:
432:
433:
434:
435:
436:
437:
438:
439:
440:
441:
442:
443:
444:
445:
446:
447:
448:
449:
450:
451:
452:
453:
454:
455:
456:
457:
458:
459:
460:
461:
462:
463:
464:
465:
466:
467:
468:
469:
470:
471:
472:
473:
474:
475:
476:
477:
478:
479:
480:
481:
482:
483:
484:
485:
486:
487:
488:
489:
490:
491:
492:
493:
494:
495:
496:
497:
498:
499:
500:
501:
502:
503:
504:
505:
506:
507:
508:
509:
510:
511:
512:
513:
514:
515:
516:
517:
518:
519:
520:
521:
522:
523:
524:
525:
526:
527:
528:
529:
530:
531:
532:
533:
534:
535:
536:
537:
538:
539:
540:
541:
542:
543:
544:
545:
546:
547:
548:
549:
550:
551:
552:
553:
554:
555:
556:
557:
558:
559:
560:
561:
562:
563:
564:
565:
566:
567:
568:
569:
570:
571:
572:
573:
574:
575:
576:
577:
578:
579:
580:
581:
582:
583:
584:
585:
586:
587:
588:
589:
590:
591:
592:
593:
594:
595:
596:
597:
598:
599:
600:
601:
602:
603:
604:
605:
606:
607:
608:
609:
610:
611:
612:
613:
614:
615:
616:
617:
618:
619:
620:
621:
622:
623:
624:
625:
626:
627:
628:
629:
630:
631:
632:
633:
634:
635:
636:
637:
638:
639:
640:
641:
642:
643:
644:
645:
646:
647:
648:
649:
650:
651:
652:
653:
654:
655:
656:
657:
658:
659:
660:
661:
662:
663:
664:
665:
666:
667:
668:
669:
670:
671:
672:
673:
674:
675:
676:
677:
678:
679:
680:
681:
682:
683:
684:
685:
686:
687:
688:
689:
690:
691:
692:
693:
694:
695:
696:
697:
698:
699:
700:
701:
702:
703:
704:
705:
706:
707:
708:
709:
710:
711:
712:
713:
714:
715:
716:
717:
718:
719:
720:
721:
722:
723:
724:
725:
726:
727:
728:
729:
730:
731:
732:
733:
734:
735:
736:
737:
738:
739:
740:
741:
742:
743:
744:
745:
746:
747:
748:
749:
750:
751:
752:
753:
754:
755:
756:
757:
758:
759:
760:
761:
762:
763:
764:
765:
766:
767:
768:
769:
770:
771:
772:
773:
774:
775:
776:
777:
778:
779:
780:
781:
782:
783:
784:
785:
786:
787:
788:
789:
790:
791:
792:
793:
794:
795:
796:
797:
798:
799:
800:
801:
802:
803:
804:
805:
806:
807:
808:
809:
810:
811:
812:
813:
814:
815:
816:
817:
818:
819:
820:
821:
822:
823:
824:
825:
826:
827:
828:
829:
830:
831:
832:
833:
834:
835:
836:
837:
838:
839:
840:
841:
842:
843:
844:
845:
846:
847:
848:
849:
850:
851:
852:
853:
854:
855:
856:
857:
858:
859:
860:
861:
862:
863:
864:
865:
866:
867:
868:
869:
870:
871:
872:
873:
874:
875:
876:
877:
878:
879:
880:
881:
882:
883:
884:
885:
886:
887:
888:
889:
890:
891:
892:
893:
894:
895:
896:
897:
898:
899:
900:
901:
902:
903:
904:
905:
906:
907:
908:
909:
910:
911:
912:
913:
914:
915:
916:
917:
918:
919:
920:
921:
922:
923:
924:
925:
926:
927:
928:
929:
930:
931:
932:
933:
934:
935:
936:
937:
938:
939:
940:
941:
942:
943:
944:
945:
946:
947:
948:
949:
950:
951:
952:
953:
954:
955:
956:
957:
958:
959:
960:
961:
962:
963:
964:
965:
966:
967:
968:
969:
970:
971:
972:
973:
974:
975:
976:
977:
978:
979:
980:
981:
982:
983:
984:
985:
986:
987:
988:
989:
990:
991:
992:
993:
994:
995:
996:
997:
998:
999:
1000:
1001:
1002:
1003:
1004:
1005:
1006:
1007:
1008:
1009:
1010:
1011:
1012:
1013:
1014:
1015:
1016:
1017:
1018:
1019:
1020:
1021:
1022:
1023:
1024:
1025:
1026:
1027:
1028:
1029:
1030:
1031:
1032:
1033:
1034:
1035:
1036:
1037:
1038:
1039:
1040:
1041:
1042:
1043:
1044:
1045:
1046:
1047:
1048:
1049:
1050:
1051:
1052:
1053:
1054:
1055:
1056:
1057:
1058:
1059:
1060:
1061:
1062:
1063:
1064:
1065:
1066:
1067:
1068:
1069:
1070:
1071:
1072:
1073:
1074:
1075:
1076:
1077:
1078:
1079:
1080:
1081:
1082:
1083:
1084:
1085:
1086:
1087:
1088:
1089:
1090:
1091:
1092:
1093:
1094:
1095:
1096:
1097:
1098:
1099:
1100:
1101:
1102:
1103:
1104:
1105:
1106:
1107:
1108:
1109:
1110:
1111:
1112:
1113:
1114:
1115:
1116:
1117:
1118:
1119:
1120:
1121:
1122:
1123:
1124:
1125:
1126:
1127:
1128:
1129:
1130:
1131:
1132:
1133:
1134:
1135:
1136:
1137:
1138:
1139:
1140:
1141:
1142:
1143:
1144:
1145:
1146:
1147:
1148:
1149:
1150:
1151:
1152:
1153:
1154:
1155:
1156:
1157:
1158:
1159:
1160:
1161:
1162:
1163:
1164:
1165:
1166:
1167:
1168:
1169:
1170:
1171:
1172:
1173:
1174:
1175:
1176:
1177:
1178:
1179:
1180:
1181:
1182:
1183:
1184:
1185:
1186:
1187:
1188:
1189:
1190:
1191:
1192:
1193:
1194:
1195:
1196:
1197:
1198:
1199:
1200:
1201:
1202:
1203:
1204:
1205:
1206:
1207:
1208:
1209:
1210:
1211:
1212:
1213:
1214:
1215:
1216:
1217:
1218:
1219:
1220:
1221:
1222:
1223:
1224:
1225:
1226:
1227:
1228:
1229:
1230:
1231:
1232:
1233:
1234:
1235:
1236:
1237:
1238:
1239:
1240:
1241:
1242:
1243:
1244:
1245:
1246:
1247:
1248:
1249:
1250:
1251:
1252:
1253:
1254:
1255:
1256:
1257:
1258:
1259:
1260:
1261:
1262:
1263:
1264:
1265:
1266:
1267:
1268:
1269:
1270:
1271:
1272:
1273:
1274:
1275:
1276:
1277:
1278:
1279:
1280:
1281:
1282:
1283:
1284:
1285:
1286:
1287:
1288:
1289:
1290:
1291:
1292:
1293:
1294:
1295:
1296:
1297:
1298:
1299:
1300:
1301:
1302:
1303:
1304:
1305:
1306:
1307:
1308:
1309:
1310:
1311:
1312:
1313:
1314:
1315:
1316:
1317:
1318:
1319:
1320:
1321:
1322:
1323:
1324:
1325:
1326:
1327:
1328:
1329:
1330:
1331:
1332:
1333:
1334:
1335:
1336:
1337:
1338:
1339:
1340:
1341:
1342:
1343:
1344:
1345:
1346:
1347:
1348:
1349:
1350:
1351:
1352:
1353:
1354:
1355:
1356:
1357:
1358:
1359:
1360:
1361:
1362:
1363:
1364:
1365:
1366:
1367:
1368:
1369:
1370:
1371:
1372:
1373:
1374:
1375:
1376:
1377:
1378:
1379:
1380:
1381:
1382:
1383:
1384:
1385:
1386:
1387:
1388:
1389:
1390:
1391:
1392:
1393:
1394:
1395:
1396:
1397:
1398:
1399:
1400:
1401:
1402:
1403:
1404:
1405:
1406:
1407:
1408:
1409:
1410:
1411:
1412:
1413:
1414:
1415:
1416:
1417:
1418:
1419:
1420:
1421:
1422:
1423:
1424:
1425:
1426:
1427:
1428:
1429:
1430:
1431:
1432:
1433:
1434:
1435:
1436:
1437:
1438:
1439:
1440:
1441:
1442:
1443:
1444:
1445:
1446:
1447:
1448:
1449:
1450:
1451:
1452:
1453:
1454:
1455:
1456:
1457:
1458:
1459:
1460:
1461:
1462:
1463:
1464:
1465:
1466:
1467:
1468:
1469:
1470:
1471:
1472:
1473:
1474:
1475:
1476:
1477:
1478:
1479:
1480:
1481:
1482:
1483:
1484:
1485:
1486:
1487:
1488:
1489:
1490:
1491:
1492:
1493:
1494:
1495:
1496:
1497:
1498:
1499:
1500:
1501:
1502:
1503:
1504:
1505:
1506:
1507:
1508:
1509:
1510:
1511:
1512:
1513:
1514:
1515:
1516:
1517:
1518:
1519:
1520:
1521:
1522:
1523:
1524:
1525:
1526:
1527:
1528:
1529:
1530:
1531:
1532:
1533:
1534:
1535:
1536:
1537:
1538:
1539:
1540:
1541:
1542:
1543:
1544:
1545:
1546:
1547:
1548:
1549:
1550:
1551:
1552:
1553:
1554:
1555:
1556:
1557:
1558:
1559:
1560:
1561:
1562:
1563:
1564:
1565:
1566:
1567:
1568:
1569:
1570:
1571:
1572:
1573:
1574:
1575:
1576:
1577:
1578:
1579:
1580:
1581:
1582:
1583:
1584:
1585:
1586:
1587:
1588:
1589:
1590:
1591:
1592:
1593:
1594:
1595:
1596:
1597:
1598:
1599:
1600:
1601:
1602:
1603:
1604:
1605:
1606:
1607:
1608:
1609:
1610:
1611:
1612:
1613:
1614:
1615:
1616:
1617:
1618:
1619:
1620:
1621:
1622:
1623:
1624:
1625:
1626:
1627:
1628:
1629:
1630:
1631:
1632:
1633:
1634:
1635:
1636:
1637:
1638:
1639:
1640:
1641:
1642:
1643:
1644:
1645:
1646:
1647:
1648:
1649:
1650:
1651:
1652:
1653:
1654:
1655:
1656:
1657:
1658:
1659:
1660:
1661:
1662:
1663:
1664:
1665:
1666:
1667:
1668:
1669:
1670:
1671:
1672:
1673:
1674:
1675:
1676:
1677:
1678:
1679:
1680:
1681:
1682:
1683:
1684:
1685:
1686:
1687:
1688:
1689:
1690:
1691:
1692:
1693:
1694:
1695:
1696:
1697:
1698:
1699:
1700:
1701:
1702:
1703:
1704:
1705:
1706:
1707:
1708:
1709:
1710:
1711:
1712:
1713:
1714:
1715:
1716:
1717:
1718:
1719:
1720:
1721:
1722:
1723:
1724:
1725:
1726:
1727:
1728:
1729:
1730:
1731:
1732:
1733:
1734:
1735:
1736:
1737:
1738:
1739:
1740:
1741:
1742:
1743:
1744:
1745:
1746:
1747:
1748:
1749:
1750:
1751:
1752:
1753:
1754:
1755:
1756:
1757:
1758:
1759:
1760:
1761:
1762:
1763:
1764:
1765:
1766:
1767:
1768:
1769:
1770:
1771:
1772:
1773:
1774:
1775:
1776:
1777:
1778:
1779:
1780:
1781:
1782:
1783:
1784:
1785:
1786:
1787:
1788:
1789:
1790:
1791:
1792:
1793:
1794:
1795:
1796:
1797:
1798:
1799:
1800:
1801:
1802:
1803:
1804:
1805:
1806:
1807:
1808:
1809:
1810:
1811:
1812:
1813:
1814:
1815:
1816:
1817:
1818:
1819:
1820:
1821:
1822:
1823:
1824:
1825:
1826:
1827:
1828:
1829:
1830:
1831:
1832:
1833:
1834:
1835:
1836:
1837:
1838:
1839:
1840:
1841:
1842:
1843:
1844:
1845:
1846:
1847:
1848:
1849:
1850:
1851:
1852:
1853:
1854:
1855:
1856:
1857:
1858:
1859:
1860:
1861:
1862:
1863:
1864:
1865:
1866:
1867:
1868:
1869:
1870:
1871:
1872:
1873:
1874:
1875:
1876:
1877:
1878:
1879:
1880:
1881:
1882:
1883:
1884:
1885:
1886:
1887:
1888:
1889:
1890:
1891:
1892:
1893:
1894:
1895:
1896:
1897:
1898:
1899:
1900:
1901:
1902:
1903:
1904:
1905:
1906:
1907:
1908:
1909:
1910:
1911:
1912:
1913:
1914:
1915:
1916:
1917:
1918:
1919:
1920:
1921:
1922:
1923:
1924:
1925:
1926:
1927:
1928:
1929:
1930:
1931:
1932:
1933:
1934:
1935:
1936:
1937:
1938:
1939:
1940:
1941:
1942:
1943:
1944:
1945:
1946:
1947:
1948:
1949:
1950:
1951:
1952:
1953:
1954:
1955:
1956:
1957:
1958:
1959:
1960:
1961:
1962:
1963:
1964:
1965:
1966:
1967:
1968:
1969:
1970:
1971:
1972:
1973:
1974:
1975:
1976:
1977:
1978:
1979:
1980:
1981:
1982:
1983:
1984:
1985:
1986:
1987:
1988:
1989:
1990:
1991:
1992:
1993:
1994:
1995:
1996:
1997:
1998:
1999:
2000:
2001:
2002:
2003:
2004:
2005:
2006:
2007:
2008:
2009:
2010:
2011:
2012:
2013:
2014:
2015:
2016:
2017:
2018:
2019:
2020:
2021:
2022:
2023:
2024:
2025:
2026:
2027:
2028:
2029:
2030:
2031:
2032:
2033:
2034:
2035:
2036:
2037:
2038:
2039:
2040:
2041:
2042:
2043:
2044:
2045:
2046:
2047:
2048:
2049:
2050:
2051:
2052:
2053:
2054:
2055:
2056:
2057:
2058:
2059:
2060:
2061:
2062:
2063:
2064:
2065:
2066:
2067:
2068:
2069:
2070:
2071:
2072:
2073:
2074:
2075:
2076:
2077:
2078:
2079:
2080:
2081:
2082:
2083:
2084:
2085:
2086:
2087:
2088:
2089:
2090:
2091:
2092:
2093:
2094:
2095:
2096:
2097:
2098:
2099:
2100:
2101:
2102:
2103:
2104:
2105:
2106:
2107:
2108:
2109:
2110:
2111:
2112:
2113:
2114:
2115:
2116:
2117:
2118:
2119:
2120:
2121:
2122:
2123:
2124:
2125:
2126:
2127:
2128:
2129:
2130:
2131:
2132:
2133:
2134:
2135:
2136:
2137:
2138:
2139:
2140:
2141:
2142:
2143:
2144:
2145:
2146:
2147:
2148:
2149:
2150:
2151:
2152:
2153:
2154:
2155:
2156:
2157:
2158:
2159:
2160:
2161:
2162:
2163:
2164:
2165:
2166:
2167:
2168:
2169:
2170:
2171:
2172:
2173:
2174:
2175:
2176:
2177:
2178:
2179:
2180:
218
```

NetBrain also supports the reverse positioning from a map note to a specific CLI Command node result.



## 2.7.4. Collaboration Between Multiple Smart CLI Consoles

Based on Map/Runbook, multiple users can work collaboratively on a troubleshooting task by using the document function of Smart CLI.

### **Collaboration Sample Flow:**

1. Level1 Engineer A shares the document results of Smart CLI with other engineers via a runbook or map.
2. Engineers B views the received notifications and opens the map to view the document results shared by Engineer A.
3. Engineers B accesses the device from the map with Smart CLI and starts troubleshooting.
4. Engineers B documents the results of a CLI command and shares the document results with Engineer A and others.
5. All these engineers can receive notifications and open the map to view the latest document results.

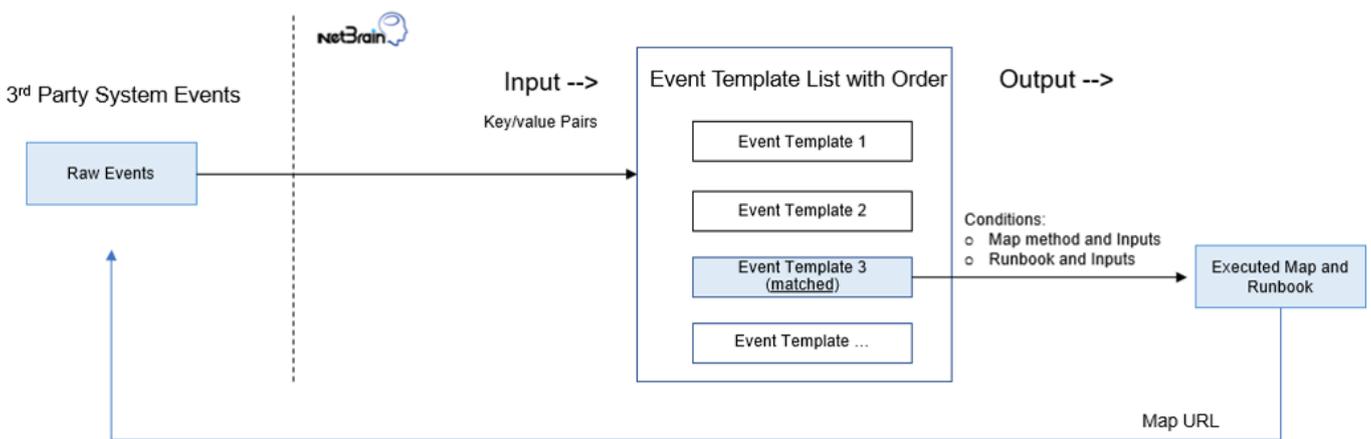


capability, users only need to define general event templates in the NetBrain system, and they will process event analysis when receiving the 3<sup>rd</sup>-party event.

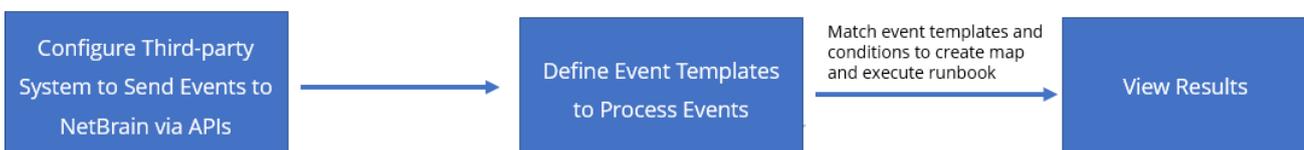
Event-driven Automation brings benefits as follows:

- It eases the burden for power users and 3<sup>rd</sup>-party system admin. They do not have to write lots of scripts in a 3<sup>rd</sup>-party system to call NetBrain maps or runbooks event by event. Instead, they only need to enable the 3<sup>rd</sup>-party system to send events to their NetBrain system directly and track event status by using REST APIs.
- It only has a minimal requirement of script supportability for a 3<sup>rd</sup>-party system. A system that can call REST APIs to send event data to NetBrain is ready to use NetBrain’s automation capabilities.
- It improves the NetBrain system capability to process a large number of events by transferring task execution from Web API Server to Worker Server. It also enables horizontal scalability to improve event processing capacity.

**Figure:** Event-driven Automation Data Flow



## Use Flow



### 2.8.1. Configure Third-party System to Send Events to NetBrain via APIs

The system provides two REST APIs to send events from a 3<sup>rd</sup>-party system to NetBrain and track its status.

- [Drive Events](#) – used to send all events generated in a 3<sup>rd</sup>-party system to a NetBrain domain.

- [Track Event Status](#) – used to get the execution status of an event.

**Example:** Define business rule in ServiceNow to send events to NetBrain IE.

```
import requests
import json
import time
import requests.packages.urllib3 as urllib3
urllib3.disable_warnings()
# Need to install requests package for python
# pip install requests

user = "admin" # account to log in to your NetBrain Domain
pwd = "admin" # password
host_url = "http://10.10.0.29" # The URL of your NetBrain Domain
headers = {'Content-Type': 'application/json', 'Accept': 'application/json'}
headers1 = {'Content-Type': 'application/json', 'Accept': 'application/json'}
'''
Get token for netbrain
'''
TENANT = 'Initial_Tenant'
DOMAIN = 'domain1'
def getTokens(user,password):
    login_api_url = r"/ServicesAPI/API/V1/Session"
    Login_url = host_url + login_api_url
    data = {
        "username": user,
        "password": password
    }
    token = requests.post(Login_url, data=json.dumps(
        data), headers=headers, verify=False)
    if token.status_code == 200:
        print(token.json())
        return token.json()["token"]
    else:
        return "error"
# get token
token = getTokens(user,pwd)
headers["Token"] = token

def get_tenant_domain_id():
    tenant_id_url = '/ServicesAPI/API/V1/CMDB/Tenants'
    full_url = host_url + tenant_id_url
    data = requests.get(full_url,headers=headers,verify=False)
    # tenant_id = '78a825ef-24bd-729d-f56f-alad2b79f2ff'
    # domain_id = '36700aff-c585-4f23-95eb-8ea00214b778'
    print(data.json())
    if data.status_code == 200:
        for tenant in data.json()['tenants']:
            if TENANT == tenant['tenantName']:
                tenant_id = tenant['tenantId']
        if tenant_id:
            domain_id_url = '/ServicesAPI/API/V1/CMDB/Domains'
            full_domain_url = host_url +domain_id_url
            domain_data =
requests.get(full_domain_url,params={'tenantId':tenant_id},headers=headers,verify=False)
```

```

        print(domain_data.json())
        if domain_data.status_code == 200:
            for domain in domain_data.json()['domains']:
                if DOMAIN == domain['domainName']:
                    domain_id = domain['domainId']
            return tenant_id, domain_id
        else:
            return tenant_id, domain_id

tenant_id, domain_id = get_tenant_domain_id()
print(tenant_id, domain_id)
headers["TenantGuid"] = tenant_id
headers["DomainGuid"] = domain_id

def Logout():
    logout_url = "/ServicesAPI/API/V1/Session"
    time.sleep(2)
    full_url = host_url + logout_url
    body = {
        "token": token
    }
    result = requests.delete(full_url, data=json.dumps(body), headers=headers, verify=False)
    print('Logout: ' + str(result.json()))
    if result.status_code == 200:
        print("LogOut success...")
    else:
        data = "errorCode" + "LogOut API test failed... "
        return result.json()
# Trigger API function

def PublishEvent(Event_Data):
    # Trigger API url
    API_URL = r"/ServicesAPI/API/V1/CMDB/EventDriven/Events"
    # Trigger API payload
    print(headers)
    api_full_url = host_url + API_URL
    print('api_full_url: ' + api_full_url)
    api_result = requests.post(api_full_url, data=json.dumps(Event_Data), headers=headers,
    verify=False)
    if api_result.status_code == 200:
        return api_result.json()
    else:
        return api_result.json()

if __name__ == "__main__":
    #tenant_id, domain_id = get_tenant_domain_id()
    #print(tenant_id, domain_id)
    # tenant_id = '0b7eb490-d9cf-aacc-672c-ff9d58a47032'
    # domain_id = '53e4b108-086e-4b6f-95b8-ee23bd7d142a'
    Event_Data = {
        "parent": "",
        "u_path_analysis_set": "",
        "made_sla": "true",
        "cause_by": "",
        "watch_list": "",
        "u_nb_task": "",
        "upon_reject": "cancel",

```

```

"sys_updated_on": "2019-06-27 15:54:14",
"child_incidents": "0",
"approval_history": "",
"skills": "",
"number": "INC0011879",
"u_destination_port": "1234",
"u_source_ip_new": "1.1.1.1",
"resolved_by": "chris.zhao",
"opened_by": {
    "link": "https://ven01749.service-now.com/api/now/table/sys_user/22121da321adf",
    "value": "232s9i2asko92asdf232322d098s"
},
"user_input": "",
"state": "2",
"knowledge": "false",
"active": "true"
}
print(PublishEvent(Event_Data))

```

## 2.8.2. Define Event Templates to Process Events

An event template is a custom executor in the NetBrain system to process events from a specified 3<sup>rd</sup>-party system. After NetBrain receives an event, it will check the qualification defined in each event template and use the qualified event template to decide whether to map and execute a runbook for the event based on conditions.

An event template is defined through python scripts and contains the following functions:

- **Define Qualification** – define the criteria that an event template can be applied to an event. There are two methods to define a qualification:

- **Regular expression**

```
"type": .*?servicenow.*?
```

- **Def qualify**

```

def qualify(raw_event):
    raw_event_json = nbjson.loads(raw_event)
    if "number" in raw_event_json and "INC" in
raw_event_json["number"]:
        return True
    else:
        return False

```

- **Define Condition and Automation (def translate)** – used to translate an event to a NetBrain task and define conditions and corresponding drill-down actions. The actions include:
  - Parse the values of variables required for map and runbook automation.

```
map_setting = mapSettingHelper.build_device_map_setting("device_name", True,
"L3_Topo_Type" )
```

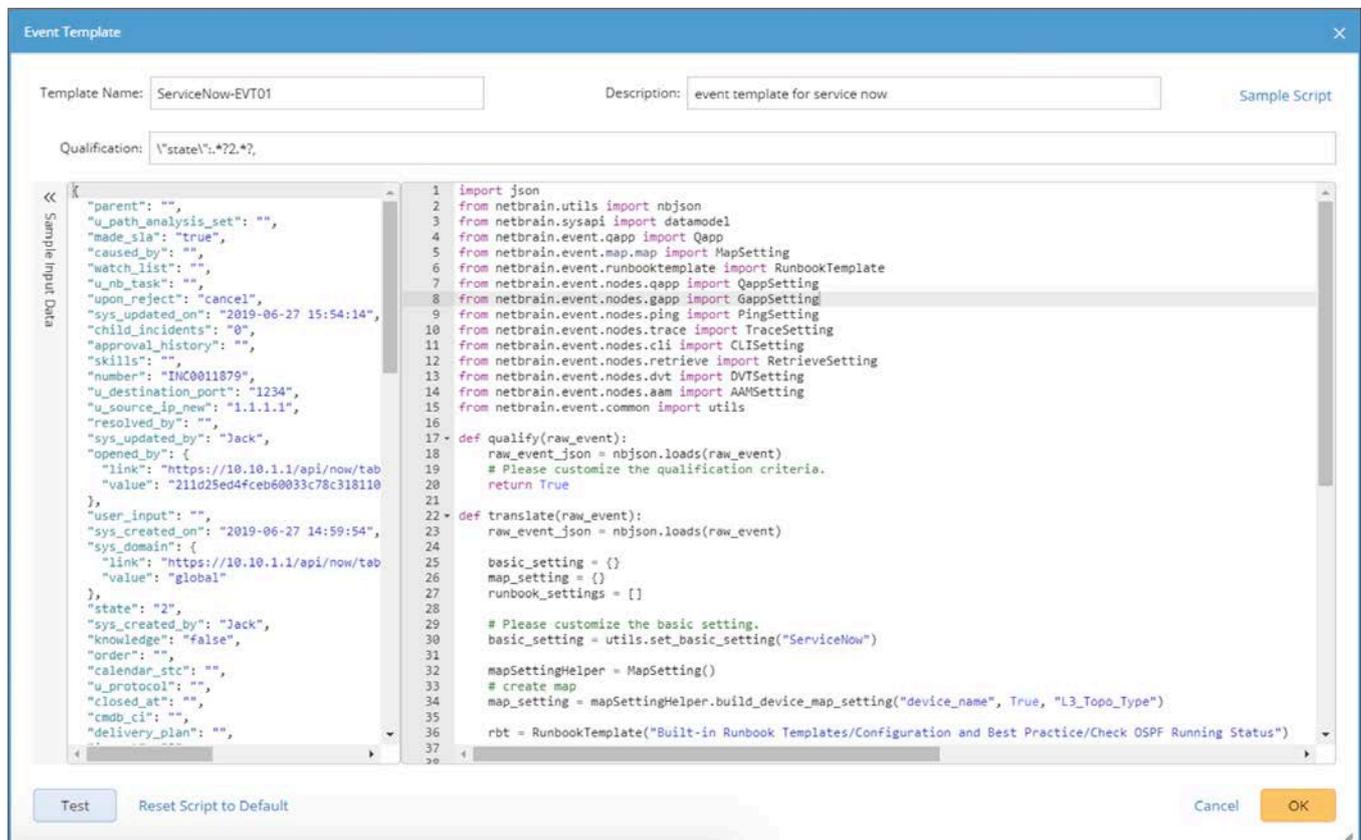
- o Define the conditions for automation.

```
if True:
    mapSettingHelper = MapSetting()
    map_setting = mapSettingHelper.build_site_map_setting("device_name", False)
    Pass
```

- o Define the input values required to draw a map or execute a runbook, and variable mapping.

```
# node name: 2. Retrieving the CLI commands of Failover, type: Execute CLI Commands
Node
rb_node = rbt.get_rbt_node("632653e6-4676-4f94-927f-86a6318f71e5")
cli_command_node = CLISetting(rb_node)
#cli_command_node.set_cli_commands(["show failover", "show monitor-interface", "show
failover interface"])
rbt.update(cli_command_node.value())
```

**Example:** Define an event template for events from ServiceNow.



## Event Process Logic:

After receiving an event, the system will:

1. Check all event templates in order until finding the one matching the event conditions.
2. Execute data parsing, mapping and automation actions defined in the matched event template.

- Return the result to the 3<sup>rd</sup>-party system in the form of a Map URL, and generates the corresponding task in NetBrain IE.

For more details on how to define an event template, refer to [Online Help](#).

### 2.8.3.View Results in System Automation Manager

All 3<sup>rd</sup>-party events and results are recorded in the System Automation Manager of NetBrain IE. Once an event task is completed, end users can view the map as well as the data collected by the Runbook.

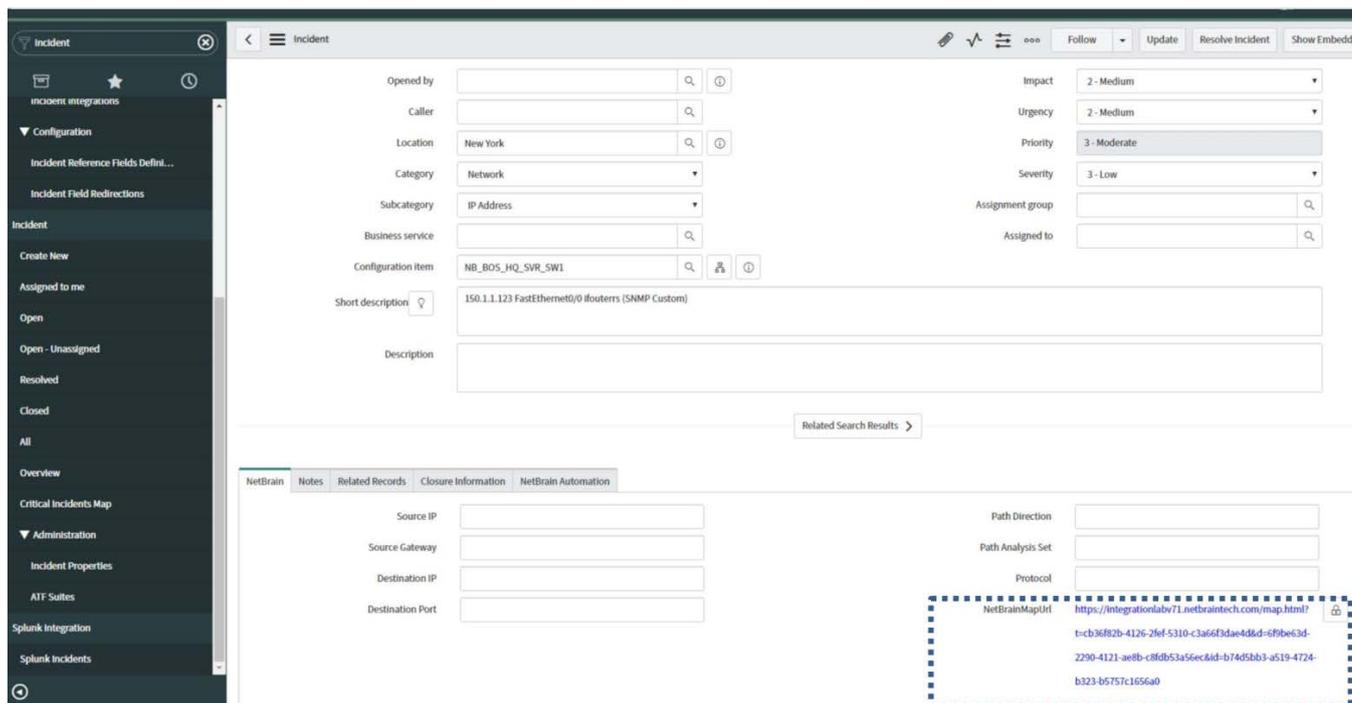
The screenshot displays the 'Event Template Manager' interface. At the top, there are tabs for 'API Triggered Tasks', 'API Stub Manager', and 'Event Template Manager'. Below the tabs, a search bar and a 'Time range' dropdown (set to 'Last 24 hours') are visible. The main area contains a table of tasks with columns: Task ID, Task/Event, Map, Status, Type, Triggered By, Triggered Time, Start Time, End Time, and Mode.

Task ID	Task/Event	Map	Status	Type	Triggered By	Triggered Time	Start Time	End Time	Mode
c383b197-3ec2-4b70-b6...	ServiceNow	ServiceNow_20191204053208576	Finished	Event Template	ServiceNow	12/4/2019, 1:32:06 PM	12/4/2019, 1:32:06 PM	12/4/2019, 1:32:15 PM	Real-Time
b9e0528b-1ec4-458b-bc...	ServiceNow_Trigger	ServiceNow_Trigger-20191204053048525	Finished	Event Template	ServiceNow	12/4/2019, 1:30:46 PM	12/4/2019, 1:30:46 PM	12/4/2019, 1:31:54 PM	Real-Time
bcb34c96-3e2d-4dc3-b2...	ServiceNow_Trigger	ServiceNow_Trigger-20191204053048525	Failed	Event Template	ServiceNow	12/4/2019, 11:05:05 AM	12/4/2019, 11:05:05 AM	12/4/2019, 11:05:07 AM	Real-Time
9a505087-a9c5-4066-be...	test	test-20191204030449657	Finished	Event Template	ServiceNow	12/4/2019, 11:04:47 AM	12/4/2019, 11:04:47 AM	12/4/2019, 11:04:52 AM	Real-Time

Below the table, a network map is displayed. On the left, a 'Leagy1' sidebar shows a 'Select Action' dropdown and a list of actions: Retrieve Live Data (1), Ping, Traceroute (1), Path, Free Text, Compare, Overall Health Check (1), and Result 1 (10:13 AM). The network map shows several nodes: BJ\*POP, NY\_DIS\_1, NY\_POPP, and NY-core-bak. Each node has associated data such as 'five\_min\_cpu\_usage', 'mem\_util', and 'cpu\_util'. A green arrow points from the task ID 'ServiceNow\_Trigger-20191204053048525' in the table to the 'NY-core-bak' node in the map.

The map URL of an event will be sent back to the 3<sup>rd</sup>-party system as a response result.

**Example:** Map URL of ServiceNow incident:



## 2.9. Search Engine Extension

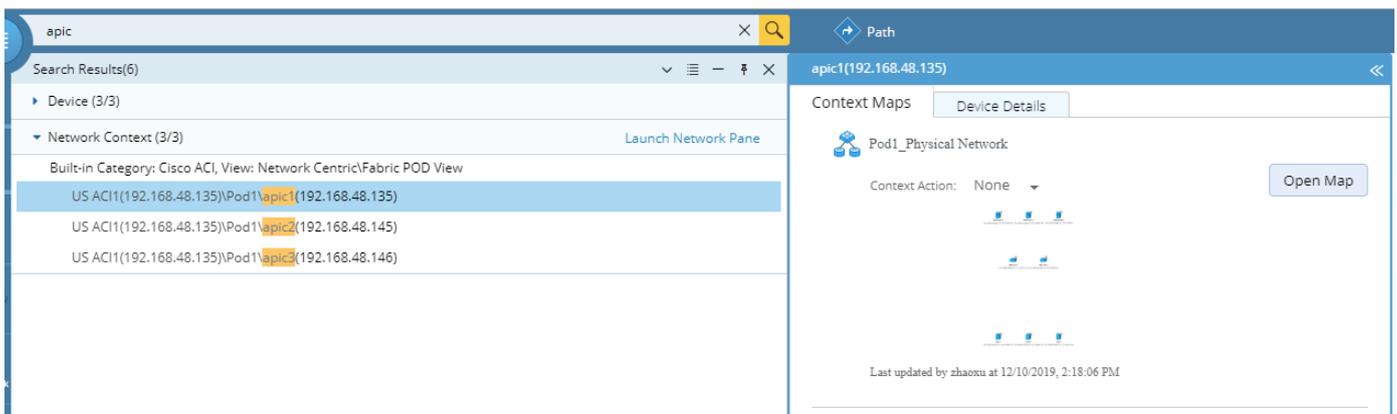
The scope for visual search is expanded, and multiple search terms are supported for each new category.

Search Object	Supported Search Terms
<a href="#">Network Context</a>	<ul style="list-style-type: none"> <li>Network Tree Node Name</li> </ul>
<a href="#">One-IP Table</a>	<ul style="list-style-type: none"> <li>IP Address</li> <li>LAN Segment</li> <li>MAC Address</li> <li>Vendor</li> <li>Switch Port</li> <li>DNS Name</li> <li>Description</li> </ul>
<a href="#">Path</a>	<ul style="list-style-type: none"> <li>Hop IP/Hostname</li> <li>Source or Destination</li> <li>Application Name</li> <li>Path Name</li> </ul>

Search Object	Supported Search Terms
<a href="#">Event</a>	<ul style="list-style-type: none"> <li>Object (Device or Path)</li> <li>Event Message</li> </ul>
<a href="#">Endpoint Table</a>	<ul style="list-style-type: none"> <li>APIC Domain</li> <li>End Point</li> <li>IP address</li> <li>MAC Address</li> <li>Interface Name</li> <li>VLAN</li> <li>Learning Source</li> <li>EPG</li> <li>Reporting Controller</li> <li>Multicast Address</li> </ul>

## 2.9.1. Search for Network Context Map

IEv8.0 extends the capability to show network contexts in the search results for an SDN node. With the context map, users can quickly understand the network design and perform drill-down actions.



## 2.9.2. Search for One-IP Table

IEv8.0 adds the capability to show the One-IP table in the search results for an IP address. Users can directly view a device's L2 neighbors from the Switch Port column and map them out.

10.138.158\*

Search Results(22)

Device (11/11)

One-IP Table (11/11) [Launch One-IP Table](#)

IP Address	LAN Segment	MAC Address	Vendor	Switch Port	DNS Name	Description
10.138.158.2	10.138.158.0/24	748E.F8B0.DAC0	Brocade C...	BIOM-E024-R...		LAWB-3:
10.138.158.3	10.138.158.0/24	748E.F89E.F4A4	Brocade C...	BIOM-E024-R...		LAWB-3:
10.138.158.8	10.138.158.0/24	748E.F89E.F4C8	Brocade C...	BIOM-E024-R...		LAWB-3:
10.138.158.7	10.138.158.0/24	748E.F8...		IM-E024-R...		LAWB-3:
10.138.158.11	10.138.158.0/24	748E.F89D.04BC	Brocade C...	BIOM-E024-R...		LAWB-3:
10.138.158.9	10.138.158.0/24	748E.F879.F228	Brocade C...	BIOM-E024-R...		LAWB-3:
10.138.158.10	10.138.158.0/24	748E.F89E.F4DC	Brocade C...	BIOM-E024-R...		LAWB-3:
10.138.158.4	10.138.158.0/24	748E.F879.EE64	Brocade C...	BIOM-E024-R...		LAWB-3:
10.138.158.6	10.138.158.0/24	748E.F879.F050	Brocade C...	BIOM-E024-R...		LAWB-3:
10.138.158.12	10.138.158.0/24	748E.F89D.0664	Brocade C...	BIOM-E024-R...		LAWB-3:

### 2.9.3. Search for A/B Paths across a Specific Hop

IPv8.0 adds the capability to search for all A/B paths crossing a specific hop by using the device name or IP address of the hop as the search term. For example, users can quickly address all the paths impacted by a problematic device, and map them out for drill-down troubleshooting.

158.7.19.5

Search Results(22)

Device (2/2)

Map (2/2)

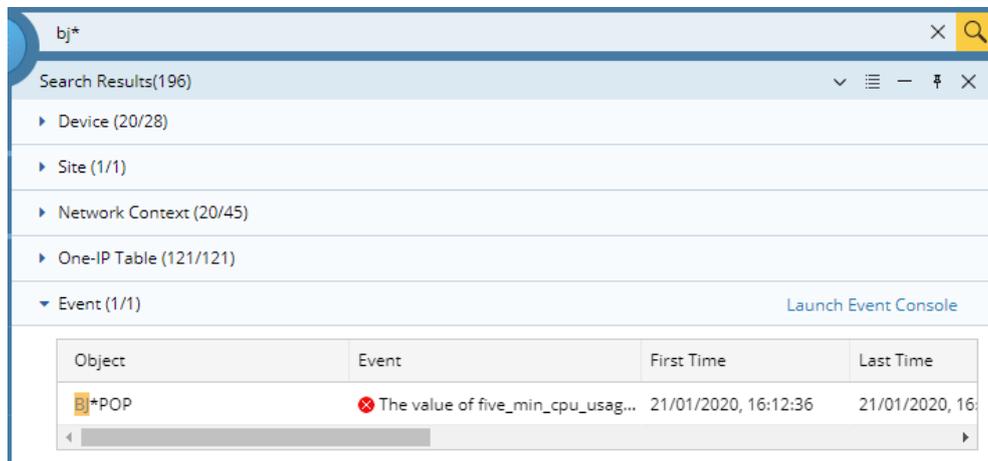
Path (18/18) [Launch](#)

Path	Application	Match Hops	Source	Destination
P_6_parameter_little	Untitled Applicati...	Y	158.7.19.1	158.7.19.62
P_7	Untitled Applicati...	Y	158.7.19.11	158.7.19.62
4	Untitled Applicati...	Y	158.7.19.7	158.7.19.62
5	Untitled Applicati...	Y	158.7.19.8	158.7.19.62
P_8_dscp_default	Untitled Applicati...	Y	158.7.19.12	158.7.19.62
P_9_dscp_empty	Untitled Applicati...	Y	158.7.19.13	158.7.19.62
P_3_tcp_Aport_miss	Untitled Applicati...	Y	158.7.19.6	158.7.19.62
2	Untitled Applicati...	Y	158.7.19.4	158.7.19.62
3	Untitled Applicati...	Y	158.7.19.6	158.7.19.62
P_2	Untitled Applicati...	Y	158.7.19.4	158.7.19.62

**Note:** The search scope for A/B paths only includes those saved in the Path Browser.

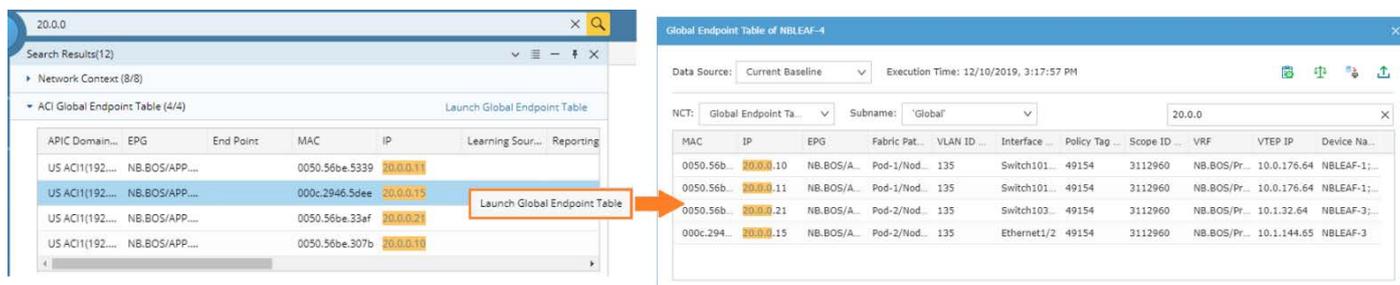
## 2.9.4. Search for Event

IEv8.0 adds the capability to search for events by using a keyword in the message as the search term. Users can start from search to map out the devices with alerts for drill-down troubleshooting.



## 2.9.5. Search for ACI Endpoint Table

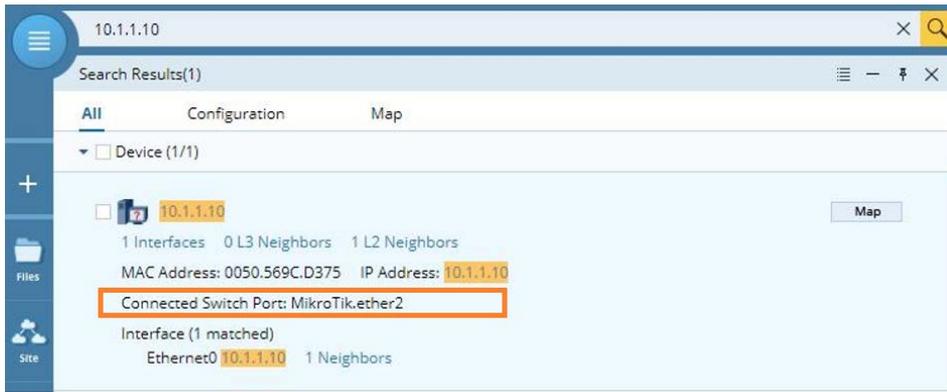
Global Endpoint Table is mainly dedicated to path calculation for a Cisco ACI network, listing all endpoint information collected from an ACI fabric. IEv8.0 adds the capability to show the related endpoint table entry in the search results for an IP address. Users can quickly launch the full ACI Endpoint Table to drill down.



## 2.9.6. Other Enhancements

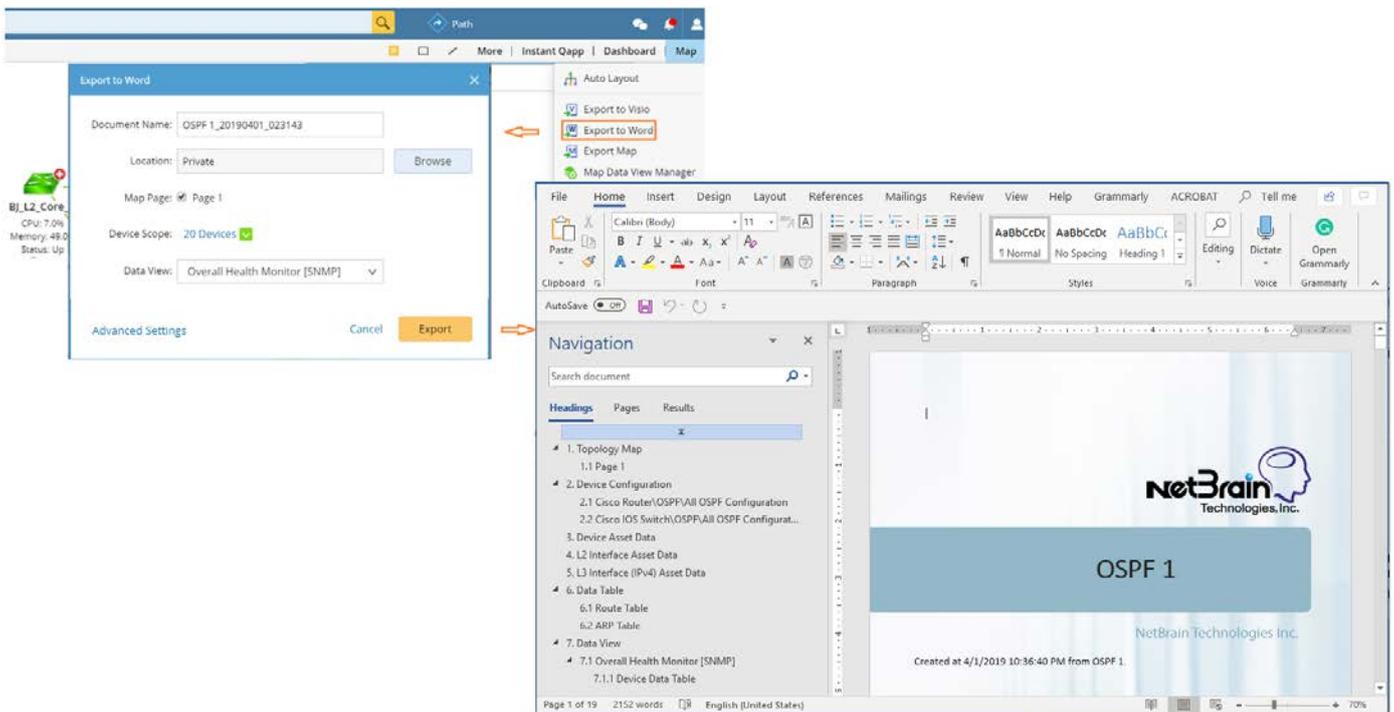
- Increase the default weight of hostname when prioritizing search results. In general, devices with matched hostnames will be displayed at the top. The value of weight is configurable at the back end.

- Display “Connected Switch Port” in the search result preview for Unknown End Systems.

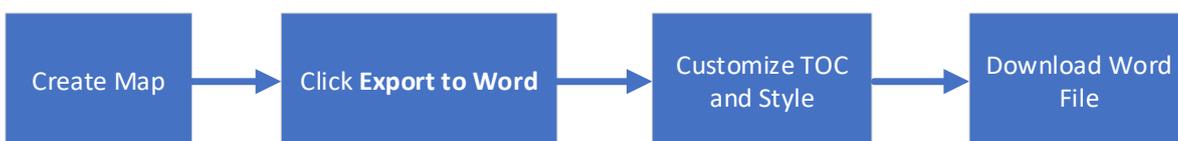


## 2.10. Export Dynamic Map to Word

IEv8.0 adds the capability to export Word documents directly from dynamic maps. Network data that can be exported contains topology maps, device configurations, data views and so on. Users can also customize advanced settings to output the desired document, such as styles, table of contents, and Word templates.



## Reference Flow and Golden Sample





OSPF

1\_20190401\_0213

Check out the golden sample file:

## Customize Contents and Styles

Before clicking the Export button, users can decide which style/template to adopt and which content to include in the exported Word document. The whole table of contents can be customized. Customized content settings can be exported and imported for future reuse.

Export to Word - Advanced Settings

Word Template: default

Content Settings: default

1. Topology Map

1.1 Page 1

2. Device Configuration

2.1 Cisco Router\OSPF\All OSPF Confi...

2.2 Cisco IOS Switch\OSPF\All OSPF C...

3. Device Asset Data

4. L2 Interface Asset Data

5. L3 Interface (IPv4) Asset Data

6. Data Table

6.1 Route Table

6.2 ARP Table

7. Data View

7.1 Overall Health Monitor [SNMP]

Recycle Bin

Template Property: default

Property	Value
Author	Kang Shaotun
Company	NetBrain
Company Address	Beijing
Title	OSPF 1

Cancel Save

Template Settings

Import Word Template

default

Name	Font
Content	Calibri 11 B I A
Heading1	Calibri Light 16 B I A
Heading2	Calibri Light 13 B I A
Heading3	Calibri Light 12 B I A
Heading4	Calibri Light 11 B I A
Heading5	Calibri Light 11 B I A

Save

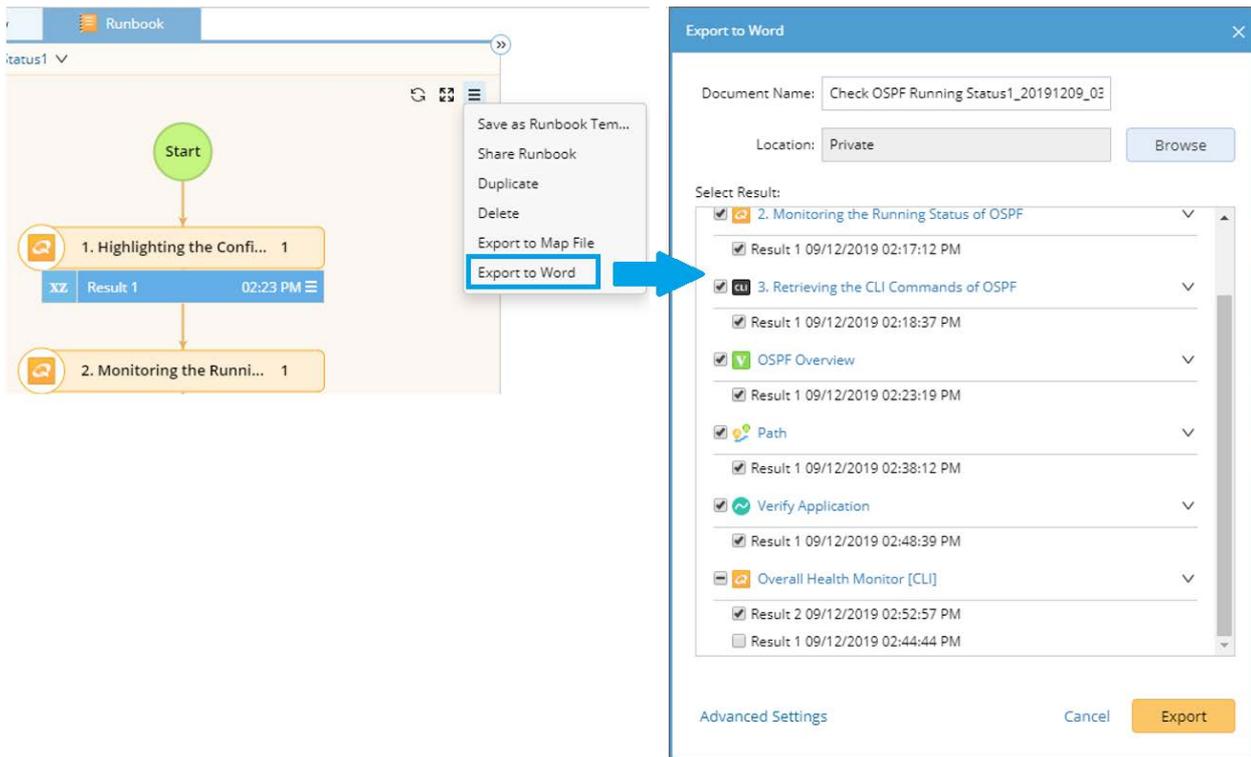
Cancel OK

By default, the following content types can be exported to a Word document:

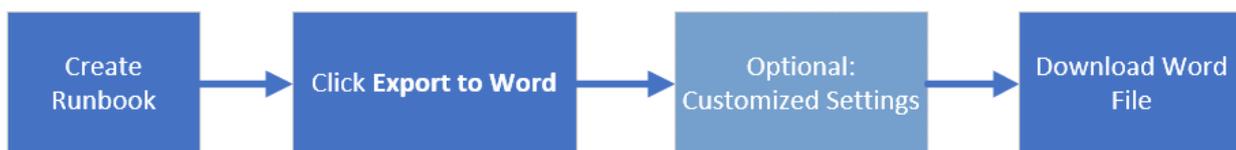
- Topology Map, including topology image and Visio map.
- Path Data, including path property, Path Overview tables.
- Device Configurations, including full configuration or Configlet.
- Device Properties and Interface Properties (IPv4 and IPv6).
- Data Table, including system tables and NCTs.
- Data View, including data tables and monitor charts with errors or warnings.

## 2.11. Document Network Change and Runbook Automation in Word

Accurate documentation is critical for managing a reliable network. I Ev8.0 adds the documentation feature for two important automation features: Change Management and Runbook. With the “Export to Word” function, each network change task can be documented in a controlled and consistent manner for archive and collaboration throughout the planning, approval, implementation and verification process, as well as other Runbook actions and results.



### User Flow to Document Runbook



Check out the golden sample files exported from two Runbooks:

#### General Runbook

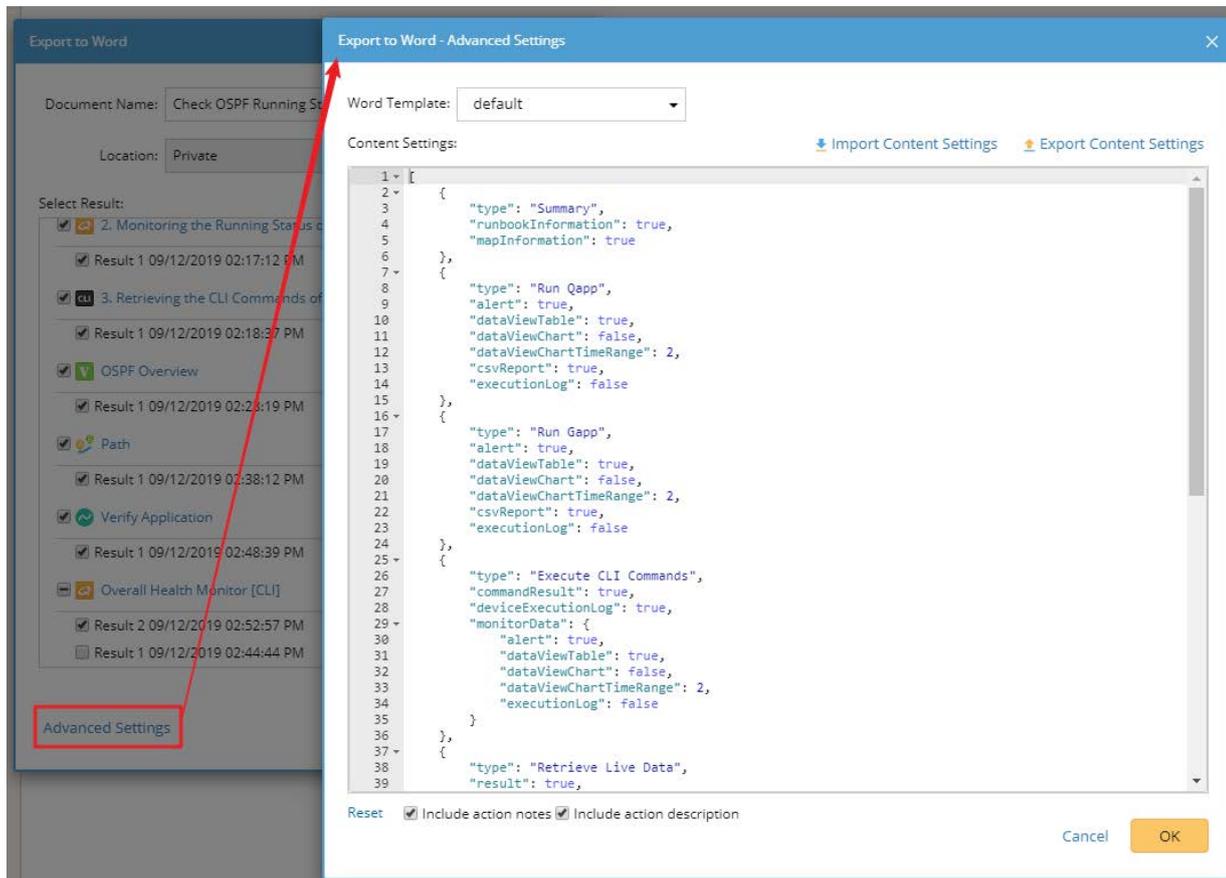


#### Change Management Runbook



## Custom Content Settings

Besides Runbook result selection, the Runbook content that can be exported is configurable in the Advanced Settings.



The default settings for the exported Runbook content can be referenced at [online help](#).

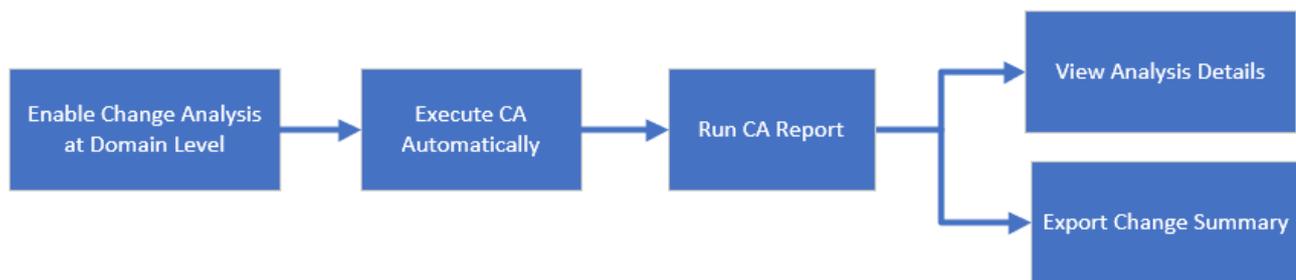
## 2.12. New Change Analysis

Change Analysis (CA) enables data comparison between the latest network data<sup>1)</sup> and previous network data<sup>1)</sup> retrieved by NetBrain scheduled tasks. The analysis results will be presented in the CA Report, where users can customize their desired time range/device scope/data type, and quickly view the change history and difference details of every change record.

**Note:**<sup>1)</sup> Network data includes Configuration File, Route Table, customized NCT Table, etc. See [Define Analysis Settings](#) for more details about supported data types.

## Reference Flow

---

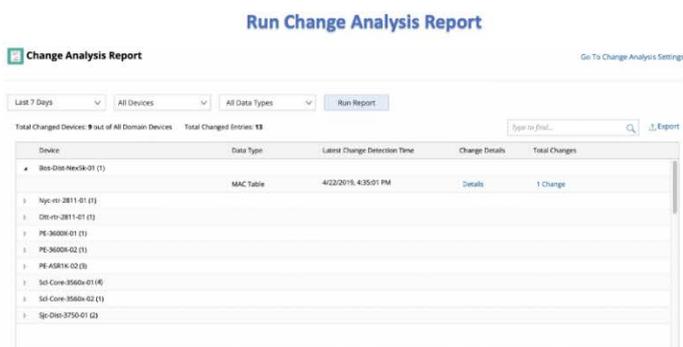


1. [Define Analysis Settings](#)
2. [Execute Change Analysis Automatically](#)
3. [Run Change Analysis Report](#)
4. [View Analysis Details](#)
5. [Export Change Analysis Report](#)

## Use Case

---

- You can leverage the global Change Analysis Report to track the data changes of your entire network such as Configuration Files and Route Tables, view the latest change details, and export the Change Summary as CSV file.
- You can benefit from the map-based Change Analysis Report and focus on the data changes of your target site map.



Details



Address	IP Address	Vendor	Port	IP Address	Vendor	Port	IP Address	Vendor	Port
10.10.10.10	10.10.10.10	Cisco Systems	eth0/24	10.10.10.10	Cisco Systems	eth0/24	10.10.10.10	Cisco Systems	eth0/24
10.10.10.10	10.10.10.10	Cisco Systems	eth0/24	10.10.10.10	Cisco Systems	eth0/24	10.10.10.10	Cisco Systems	eth0/24



Export

Device Name	Data Type	Latest Change Detection Time (Pacific Daylight Time)	Total Changes
Bos-Core-5500	MAC Table	04/22/2019, 1:35:04 PM	1
Bos-Core-6500	Route Table	04/23/2019, 3:00:05 AM	3
Bos-Dist-Nex5k-01	MAC Table	04/22/2019, 1:35:04 PM	1
Bos-Dist-Nex5k-02	MAC Table	04/22/2019, 1:35:04 PM	1
Bos-F3_LB-ASA-SW	MAC Table	04/22/2019, 1:35:00 AM	1
MPLS-P-ASR9001-01	Route Table	04/23/2019, 3:00:03 AM	2
Nyc-rt-2811-01	Route Table	04/23/2019, 3:00:03 AM	3
Orl-rt-2811-01	Route Table	04/23/2019, 3:00:03 AM	3
PE-3600X-01	BGP Advertised-route Table	04/22/2019, 1:34:59 PM	3
PE-3600X-01	Route Table	04/22/2019, 10:00:02 PM	2
PE-3600X-02	BGP Advertised-route Table	04/22/2019, 1:35:02 PM	2
PE-ASR1K-01	BGP Advertised-route Table	04/22/2019, 1:35:04 PM	1
PE-ASR1K-01	Route Table	04/23/2019, 3:00:06 AM	2
PE-ASR1K-02	BGP Advertised-route Table	04/22/2019, 1:35:03 PM	1
ScI-Core-3560x-01	Route Table	04/23/2019, 1:34:59 PM	1
ScI-Core-3560x-02	Route Table	04/23/2019, 8:30:02 AM	14
ScI-Dist-3750-01	Route Table	04/22/2019, 1:35:03 PM	1
ScI-Dist-3750-02	Route Table	04/22/2019, 1:35:04 PM	1
Sjc-Core-3560x-01	Route Table	04/23/2019, 3:00:03 PM	3
Sjc-Core-3560x-02	Route Table	04/22/2019, 1:35:02 PM	1
Sjc-Dist-3750-01	Route Table	04/22/2019, 1:34:59 PM	1
Sjc-Dist-3750-01	MAC Table	04/22/2019, 1:34:59 PM	1
Sjc-Dist-3750-02	Route Table	04/22/2019, 7:00:08 PM	2
Sun-Dist-srx240	Route Table	04/23/2019, 5:30:02 AM	11

## 2.12.1. Define Analysis Settings

In Domain Management, you can enable/disable Change Analysis and select the desired data type (see the following table) for comparison.

Data Type	Details
Built-in Data	<ul style="list-style-type: none"> <li>Configuration</li> <li>Route Table</li> <li>ARP Table</li> <li>MAC Table</li> <li>NDP Table</li> <li>STP Table</li> <li>BGP Advertised-route Table</li> </ul>
NCT Table	GRE Tunnels, VXLAN Peer Table, etc.

## 2.12.2. Execute Change Analysis Automatically

Change Analysis comparison is executed automatically after data is retrieved.

If you enable the option 'Auto update all selected data in Current Baseline' when configuring on-demand live data retrieval, all the associated data will be added to CA comparison.

The system will hash the data and compare the latest hash values with the previous ones to determine if the data change has occurred. The change record will be logged to the CA result table and available when the user runs the CA report.

### 2.12.3. Run Change Analysis Report

**From Global**

**From Map**

Device	Data Type	Latest Change Detection Time	Change Details	Total Changes
Bios-Disk-Nex5k-01 (1)	MAC Table	4/22/2019, 4:35:01 PM	Details	1 Change
Nyc-rtr-2811-01 (1)				
Or-rtr-2813-01 (1)				
PE-3600K-01 (1)				
PE-3600K-02 (1)				
PE-ASR1K-02 (B)				
Srl-Core-3560a-01 (A)				
Srl-Core-3560a-02 (1)				
Srl-Dev-3750-01 (2)				

1. Launch Change Analysis Report from the system menu (or from a map).
2. Specify the time range, device, and data type.
3. Run Report to get a specified change analysis summary.

### 2.12.4. View Change Layout Details

Two types of data (string data and table data) are supported by Change Analysis. You can view details of either data type as below:

- **String Data:**

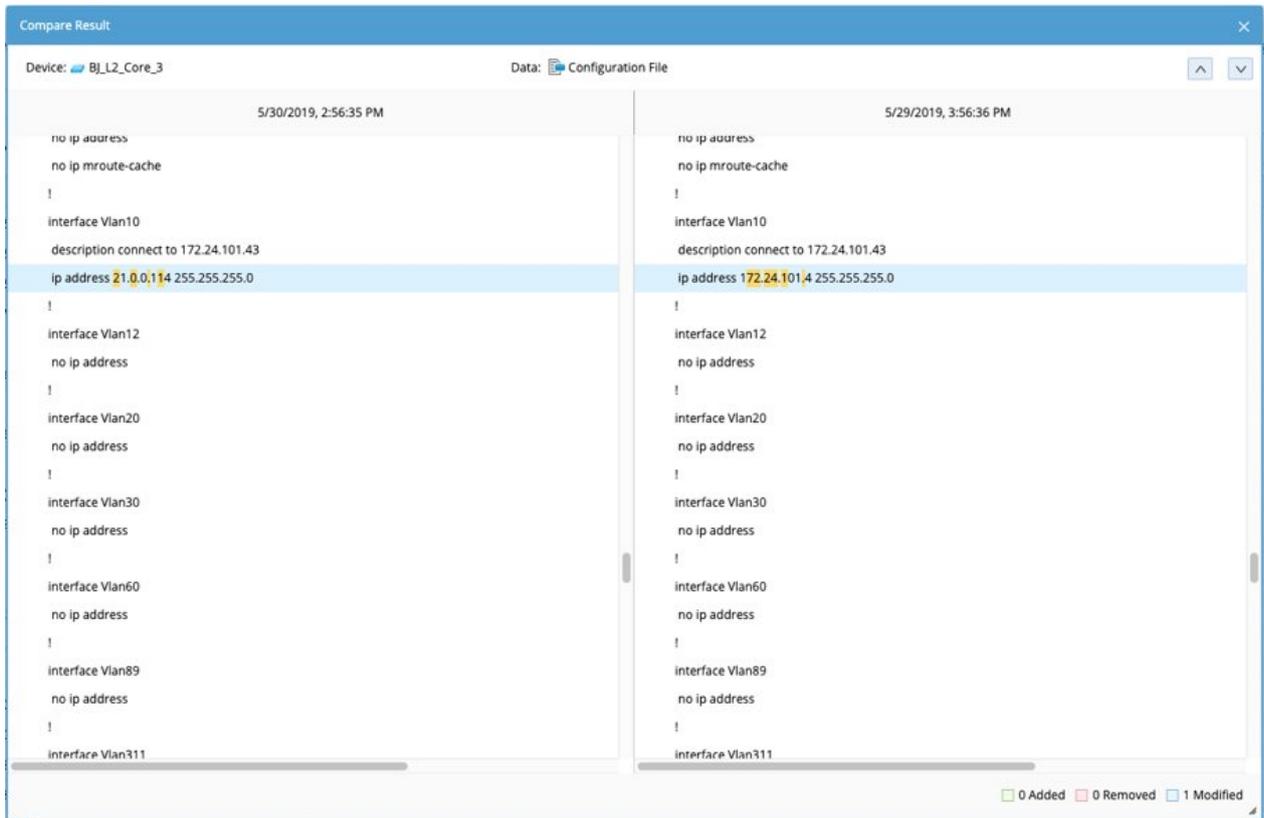
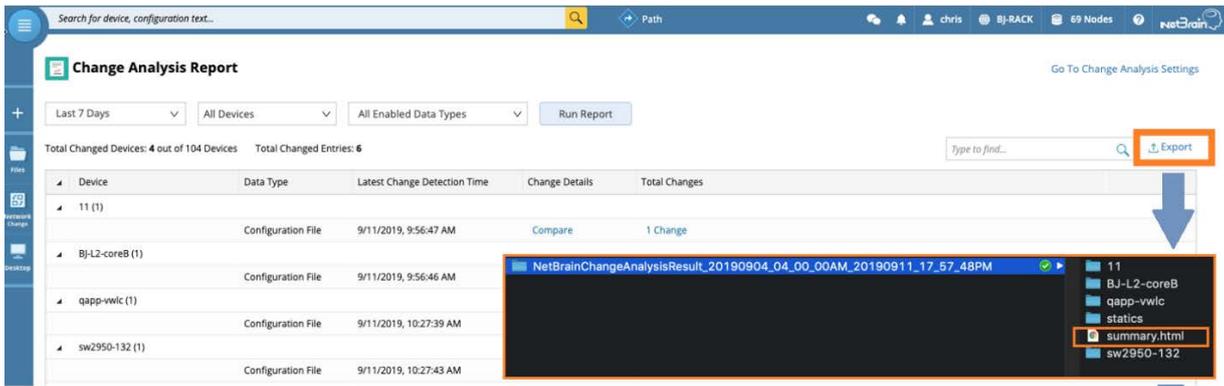


Table Data:

4/24/2019, 5:34:20 PM							4/24/2019, 5:34:11 PM							
Alg	Dest.Addr	Mask	Distance	Metric	Interface	Next Hop IP	Alg	Dest.Addr	Mask	Distance	Metric	Interface	Next Hop IP	Next Hop Dev...
Added (2 of 2)														
B	194.168.46.0	24	95	0		139.179.13.169								
B	133.190.0.8	32	95	0		139.179.13.169								
Removed (2 of 2)														
B	194.168.106.99	32	95	0		139.179.13.169	B	194.168.106.99	32	95	0		139.179.13.169	Sa
B	194.168.46.0	29	95	0		139.179.13.169	B	194.168.46.0	29	95	0		139.179.13.169	Sa
Modified (6 of 6)														
B	218.98.190.144	29	95	0		139.179.13.169	B	218.98.190.144	29	95	0		139.179.13.169	Sa
B	218.98.190.144	29	99	9		139.179.13.199	B	218.98.190.144	29	99	9		139.179.13.199	Sa
B	70.139.22.176	30	95	0		139.179.13.169	C	70.139.22.176	30	95	0		139.179.13.169	Sa
B	139.183.36.24	29	95	0		139.179.13.169	B	139.183.36.24	29	199	1		139.179.13.169	Sa
D	157.129.88.0	25	90	3072	GigabitEthernet0/0	133.190.192.178	D	157.129.88.0	25	90	3072	GigabitEthernet0/9	133.190.192.178	Sa
D	12.220.128.140	30	90	13056512	GigabitEthernet0/0	133.190.192.178	D	12.220.128.140	30	90	13056512	GigabitEthernet0/0	145.1.196.31	usb001945

## 2.12.5. Export Change Analysis Report

Change Analysis Report summarizes the changes based on specified time range/devices/data types. It can be exported to a zip file. Besides the change summary, users can view the change details (before and after) of each device side by side in the exported report, such as configuration files, route tables, and so on.



Tenant: Initial Tenant Domain: BJ-RACK  
 Mode: Change Analysis  
 Input: 4 Device(s), 4 Data; Dataset1 - 09/04/2019 04:00:00 AM UTC, Dataset2 - 09/04/2019 04:00:00 AM UTC  
 Result: 4 of 4 Device(s) Changed, 4 of 4 Data Changed

Y: Change N: No Change

Hostname	Configuration File
11	Y
BJ-L2-coreB	Y
qapp-vwlc	Y
sw2950-132	Y

```

banner motd ^C
SNMPv3 cfg:
snmp-server user snmpv3 snmp3 v3 auth md5 nbv3authmd5 priv aes 192 netbraincryaes192

^C
!
line con 0
line vty 0 4
exec-timeout 5 0
transport preferred ssh
transport input ssh

line con 4
line vty 4
line vty 5
  
```

**Note:** Unzip the exported file and double-click the **summary.html** file to view the portable report. For each device, only the last change during the specified time range will be exported.

## 2.13. Ansible Integration

Ansible Integration is an add-on feature to Change Management Module. Users can leverage the Ansible Task node (in Change Management Runbook) to define and execute an Ansible playbook, document the execution result and utilize NetBrain Change Management flow or 3<sup>rd</sup>-party system integration to manage the network change approval process, and perform change verification automation.

**Example:** Ansible Task Node in Runbook

Data View | Runbook | Instant Qapp

All Runbooks > Test 1

Select Action ▾

```

graph TD
    Start((Start)) --> Planning{Planning}
    Planning --> DefineChange[Define Change]
    DefineChange --> BenchmarkBefore[Benchmark Before]
    BenchmarkBefore --> Execute[Execute]
    Execute --> AnsibleTask[Ansible Task]
    AnsibleTask --> BenchmarkAfter[Benchmark After]
    BenchmarkAfter --> Compare[Compare]
    Compare --> End(( ))
  
```

**Ansible Task** Description

SDN Device ▾ | Jay\_repo ▾ | Branch/tag/Commit | Sync

Playbook: cisco\_aci\_query.yml | Commit: 02f390a

Playbook | Inventory | Dry-Run Log

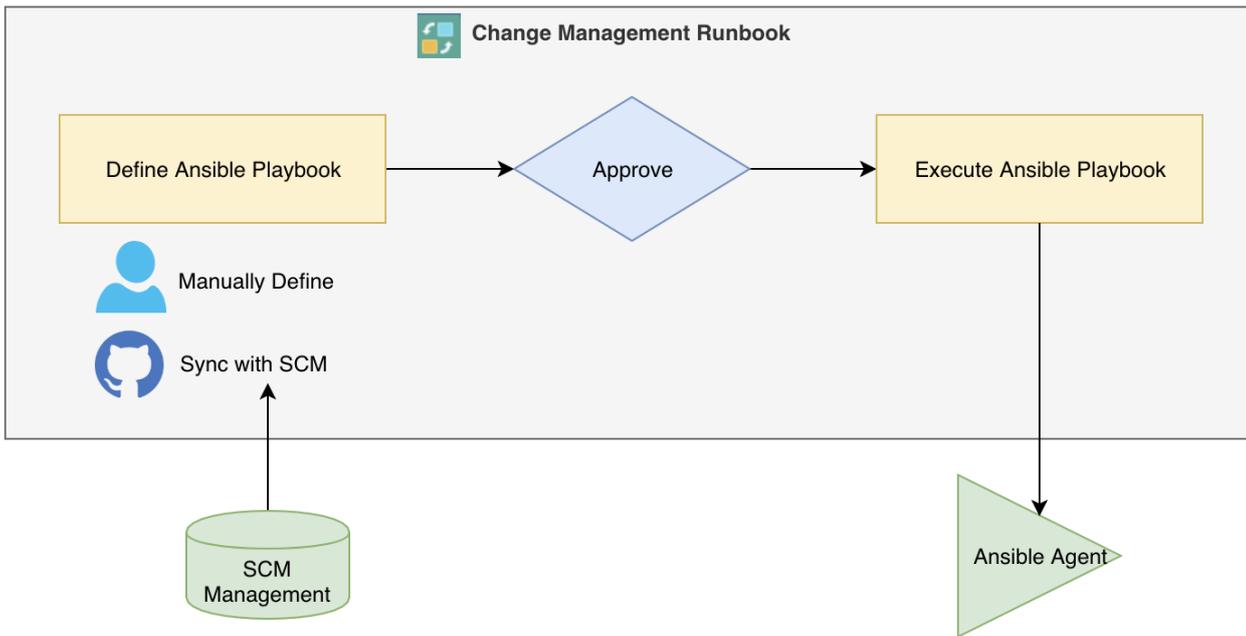
```

1 - name: "Cisco ACI Query Tenant/EPG"
2 hosts: all
3 gather_facts: no
4 connection: local
5
6 tasks:
7 - name: ACI query all tenant
8   aci_tenant:
9     hostname: "{{ inventory_hostname }}"
10    username: "{{ username }}"
11    password: "{{ password }}"
12    validate_certs: no
13    state: query
14
15 - name: ACI query all EPG
16   aci_epg:
17     hostname: "{{ inventory_hostname }}"
18     username: "{{ username }}"
19     password: "{{ password }}"
20     validate_certs: no
21     state: query
  
```

Define Extra Variables

Dry-Run Run

## Reference Flow



1. [Define Ansible Playbook](#)
2. [Execute Ansible Playbook](#)
3. [NetBrain Ansible Deployment](#)

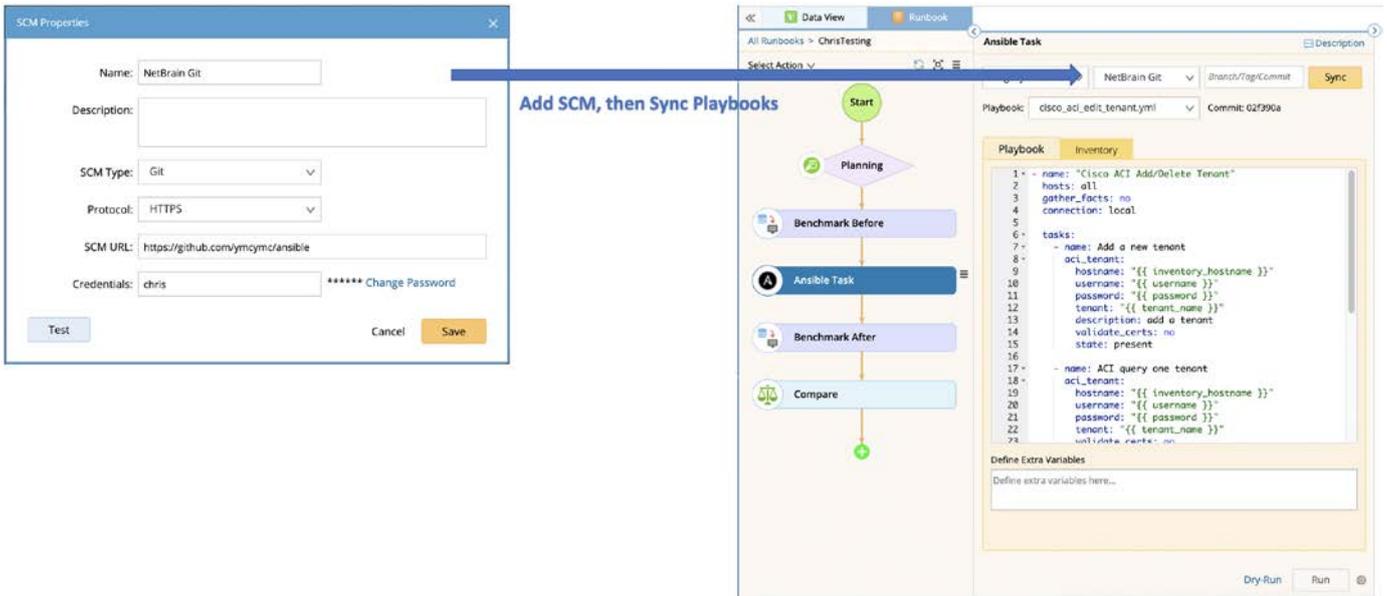
### 2.13.1. Define Ansible Playbook

Playbook and Inventory can be defined in the Ansible Task node.

- [Integrate with SCM to Sync Up Playbook](#)
- [Manually Define Playbook](#)
- [Inventory Definition](#)

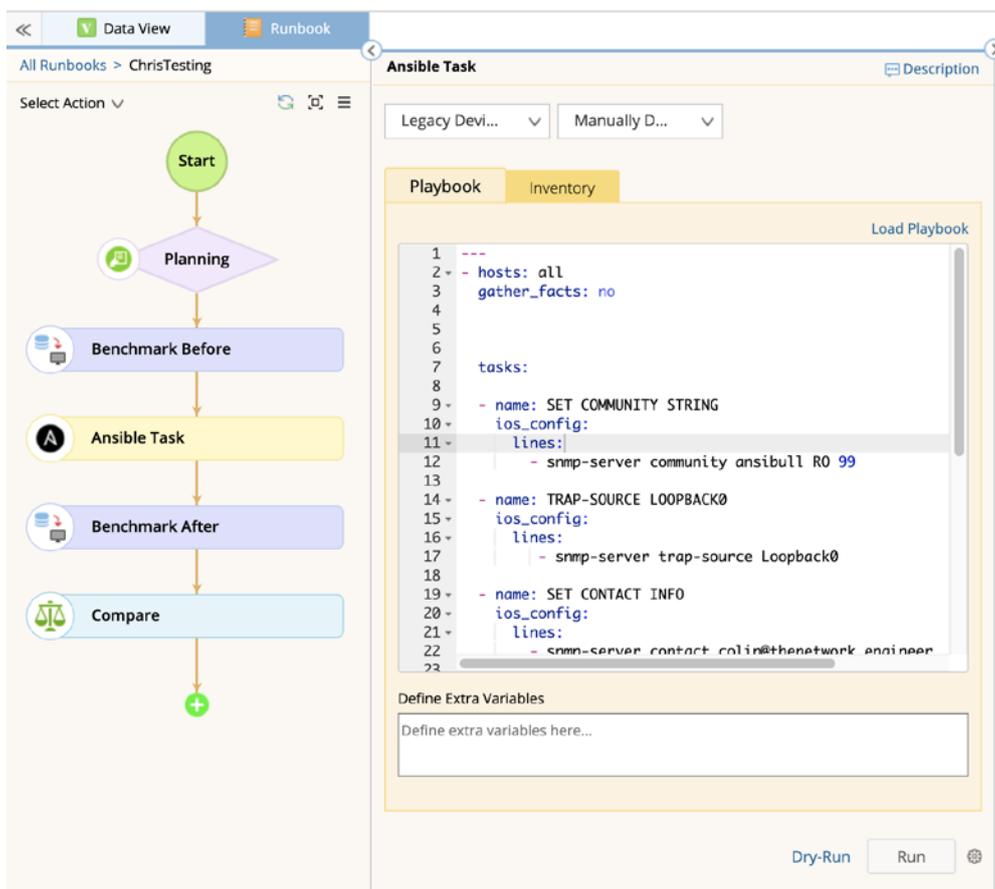
### Integrate with SCM to Sync Up Playbook

Most Ansible users will leverage SCM (Source Code Management) system, such as GitHub, to manage their playbooks. NetBrain can provide seamless integration with the mainstream SCM system, enabling effortless playbook sync-up with the SCM systems.



## Manually Define Playbook

For those users who do not use an SCM system to manage the playbooks, they have the option to create playbook in Ansible Task node with standard YAML syntax.



## Inventory Definition

---

NetBrain can act as a CMDB to provide the necessary information (such as host group, device login credentials, interface list, and visible interface list) for the Ansible Inventory Template definition. The built-in parameters provided by NetBrain enable a more flexible and convenient user experience of defining the inventory template, which will be used to generate inventory instance.

The table below summarizes the built-in parameters supported by IEv8.0 for inventory usage:

Category	Parameter
Legacy Device	<code>device.property_name</code> : device property, e.g. <code>\$device.vendor</code> <code>\$credential.username</code> : username to login current device (host) <code>\$credential.password</code> : password to login current device (host) <code>\$credential.enable_username</code> : username to enter into privilege mode <code>\$credential.enable_password</code> : password to enter into privilege mode <code>\$credential.ssh_port</code> : port number of SSH access <code>\$credential.enable_cmd</code> : command to enter into privilege mode <code>\$credential.quit_cmd</code> : command to quit privilege mode
SDN Device	<code>\$sdn_controller.controller_name</code> : e.g, <code>\$sdn_controller.apic_1</code> <code>\$credential.username</code> <code>\$credential.password</code>
Interface	<code>\$interface._all</code> : all interfaces of current device (host) <code>\$interface._visible</code> : visible interface of device shown on map <code>\$interface.management_interface</code> : management interface of current device (host)

---

### 2.13.2. Execute Ansible Playbook

- [Approval Process](#)
- [Execute Playbook \(and options\)](#)

#### Approval Process

---

As a component of Change Management Runbook, Ansible Task follows the Change Management Approval flow. Without approval, no playbook can be executed.

Two approval methods are supported by NetBrain Change Management:

- **Via NetBrain:** use NetBrain system to define roles including Network Change creator, approver, executor, and viewer. The approver is the only role that has the privilege to approve a network change request.

- **Via External System:** leverage third-party system (e.g., ServiceNow) integration to enable status synchronization of network change approval flow.

## Execute Playbook

Ansible playbook can be executed once the CM Runbook is approved. The detailed execution log is available to view the playbook execution result.

Ansible Node Description

Legacy Device Manually Defined

Playbook Inventory Execution Log

```

Using /etc/ansible/ansible.cfg as config file

PLAY [show ip interface brief]
*****

TASK [run show ip interface brief IOS]
*****
ok: [192.168.180.107] => {"changed": false, "stdout_lines": [{"Interface
Address OK? Method Status Protocol\nGigabitEthernet0/0
192.168.180.107 YES NVRAM up up \nGigabitEthernet0/1
10.10.2.2 YES NVRAM up up \nGigabitEthernet0/2
10.10.3.2 YES NVRAM up up \nGigabitEthernet0/3
unassigned YES unset up up \nGigabitEthernet0/4
unassigned YES unset administratively down down \nGigabitEthernet0/5
unassigned YES unset administratively down down \nLoopback0
2.2.2.2 YES NVRAM up up"}, {"Interface
IP-Address OK? Method Status Protocol", "GigabitEthernet0/0
192.168.180.107 YES NVRAM up up ", "GigabitEthernet0/1
10.10.2.2 YES NVRAM up up ", "GigabitEthernet0/2
10.10.3.2 YES NVRAM up up ", "GigabitEthernet0/3
unassigned YES unset up up ", "GigabitEthernet0/4
unassigned YES unset administratively down down ", "GigabitEthernet0/5
unassigned YES unset administratively down down ", "Loopback0
2.2.2.2 YES NVRAM up up"}]}}

PLAY RECAP
*****
**
192.168.180.107 : ok=1 changed=0 unreachable=0 failed=0

```

Execution started by Chris at 08/22/2018 10:47:26 AM

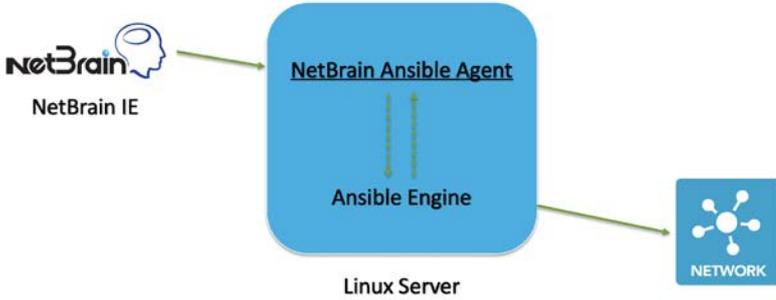
Dry-Run Terminate

**Note:** Ansible supports the Dry-run mode to simulate the Ansible Node execution process. Clicking **Dry-Run** hyperlink will generate Dry-run Log containing Dry-run results.

### 2.13.3. NetBrain Ansible Deployment

The following prerequisites must be met to enable Ansible integration:

- NetBrain IEv8.0 with Change Management module license (provided by NetBrain)
- NetBrain Ansible Agent (provided by NetBrain)
- Ansible Engine server (provided by customers)



## Install NetBrain Ansible Agent

NetBrain provides a standalone installation package for Ansible Agent. Ansible Agent must be installed on the same machine where Ansible Engine is installed.

**Note:** Customers need to deploy the Ansible Engine before installing NetBrain Ansible Agent. Refer to <https://www.ansible.com/products/engine> for more information about Ansible Engine.

**Note:** Installation package and documentation of NetBrain Ansible Agent can be downloaded from Ansible Agent Manager.

Domain Management Tenant: Initial Tenant Domain: BJ\_rack Operations Chris.Zhao@net...

---

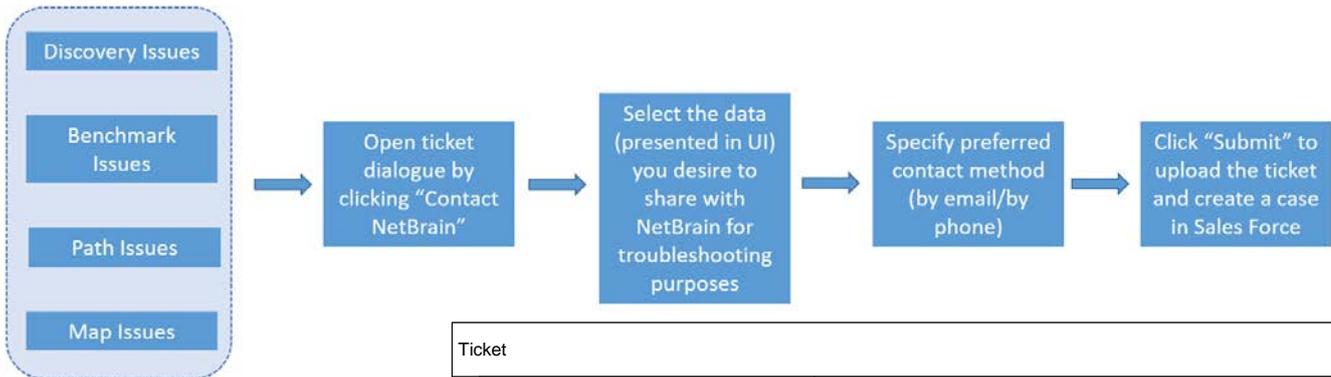
Start Page Ansible Agent Manager

0 Items + Add Download Ansible Agent

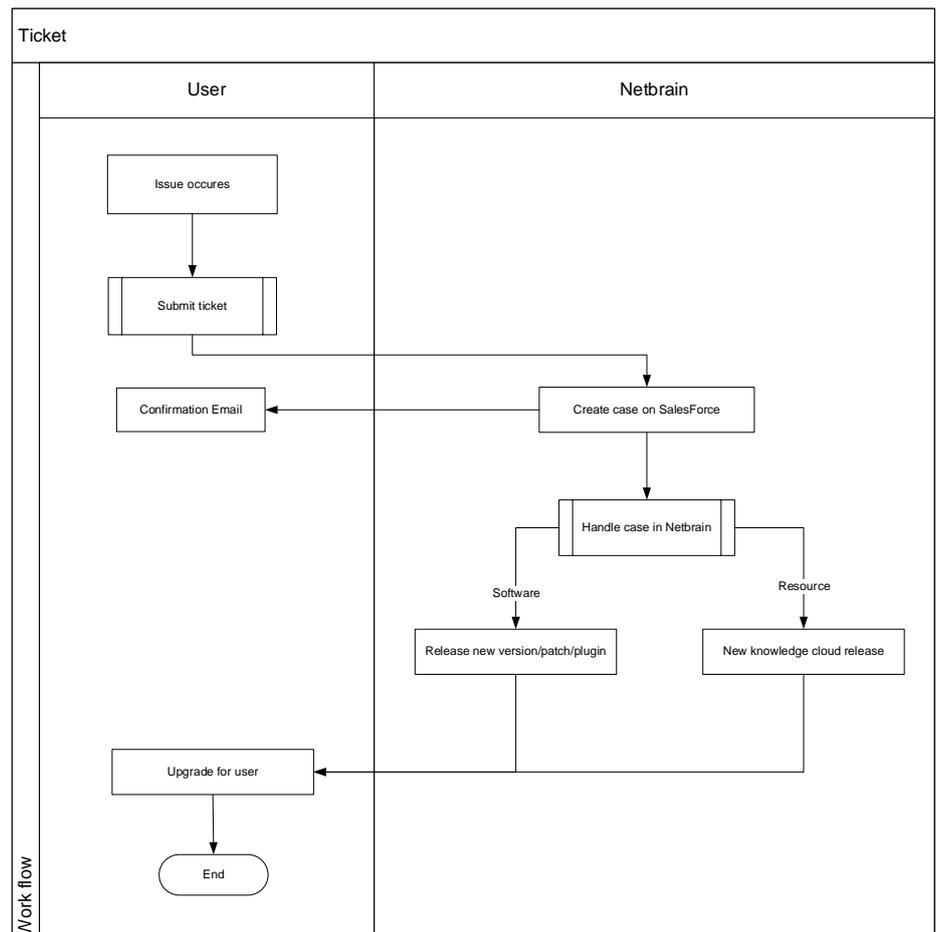
Agent Name	Description	Endpoints	Front Server

## 2.14. Enhanced Supportability for Customer Service Ticketing

NetBrain ticketing system enables centralized management of customer requests including bug investigation requests, unknown object ID investigation requests, and generic requests. The ticketing system automates the data collection process and provides Support Engineers with the necessary network environment information to perform troubleshooting. Successful implementation of the ticketing system will simplify the current troubleshooting process by reducing both time and cost.



**Note:** The above reference flow excludes [Submit a Ticket for Unknown Object ID](#) and [Submit a Generic Issue](#).



## 2.14.1. Auto-Collected Data in Ticketing

The following table summarizes the supported ticket types and their required data for troubleshooting purposes:

Ticket Type	Required Data
Discovery Issue	<ul style="list-style-type: none"><li>▪ Discovery Settings<ul style="list-style-type: none"><li>○ Network Settings</li><li>○ Network Definitions</li><li>○ Do-Not-Scan List</li><li>○ Live Access Settings</li></ul></li><li>▪ Discovery Report (latest discovery)</li><li>▪ Execution Log (latest discovery)</li><li>▪ NetBrain Server Status</li><li>▪ NetBrain Server Log (Worker server, Web server Task engine and Rabbit MQ server)</li></ul>
Benchmark Issue	<ul style="list-style-type: none"><li>▪ Schedule Discovery/ Benchmark Configuration</li><li>▪ Latest Task Result<ul style="list-style-type: none"><li>○ Execution log</li><li>○ Device Log</li><li>○ Plugin Log</li></ul></li><li>▪ NetBrain Server Status</li><li>▪ NetBrain Server Log (Worker server, Web server Task engine and Rabbit MQ server)</li></ul>
Map Issue	<ul style="list-style-type: none"><li>▪ Device/Interface Properties (from all the map devices)</li><li>▪ Configuration Files (from all the map devices)</li><li>▪ System Table<ul style="list-style-type: none"><li>○ Route Table</li><li>○ Mac Table</li><li>○ ARP Table</li><li>○ NDP Table</li><li>○ STP Table</li></ul></li><li>▪ NCT Table (from all the map devices)</li><li>▪ Topology Information</li><li>▪ Duplicate IP Result</li><li>▪ Latest Benchmark Result</li></ul>
Path Issue	<ul style="list-style-type: none"><li>▪ Device/Interface Properties (from all the map devices)</li><li>▪ Configuration Files (from all the map devices)</li><li>▪ System Table<ul style="list-style-type: none"><li>○ Route Table</li><li>○ Mac Table</li><li>○ ARP Table</li><li>○ NDP Table</li><li>○ STP Table</li></ul></li></ul>

Ticket Type	Required Data
	<ul style="list-style-type: none"> <li>▪ NCT Table (from all the map devices)</li> <li>▪ CLI Result</li> <li>▪ CheckPoint Manager Data</li> <li>▪ Path Data               <ul style="list-style-type: none"> <li>○ Device Log</li> <li>○ Traffic Status</li> <li>○ Execution Log</li> </ul> </li> <li>▪ Topology Information</li> <li>▪ Latest Benchmark Result</li> </ul>

## 2.14.2. Ticketing Samples

### Submit a Ticket for A/B Path

When the path calculation fails, you can launch a path bug investigation request by clicking the **Contact NetBrain** button in the path result pane.

The screenshot displays the NetBrain interface with a path calculation failure. A dialog box titled "Contact NetBrain - Path Issues" is open, prompting the user to submit information for a path bug investigation. The dialog box includes the following fields and options:

- Subject:** A text input field.
- Description:** A text input field with a placeholder: "Please confirm if it reflected correct path, apart from the data collected from devices, please also specify 1) Correct device sequence on this path. 2) A traceroute result from the source device".
- Path:** A section with the instruction: "If there are other devices should be included in path, please drag and drop on map before you create issue." Below this is a small map showing the path.
- Relevant Data:** A section with the text: "Retrieved data / log / xmap for troubleshooting purposes only, we will view the data in a secure encryption environment." Below this are two checked checkboxes: "Device/Interface Properties" and "Configuration File".
- Buttons:** "Export", "Cancel", and "Submit".

A red circle with the number 1 highlights the "Contact NetBrain" button in the path result pane on the left side of the interface.

## Submit a Ticket for Dynamic Map

When an error occurs during a map generation process, you can launch a map bug investigation request by clicking the **Contact NetBrain** button on the lower right side of the screen.

The screenshot displays the NetBrain interface with a 'Contact Netbrain - Map Issues' dialog box open. The dialog box includes the following sections:

- Checkboxes:**  NCT Table,  Duplicate IP Result,  Latest Benchmark Result, and  Delete data when issue resolved.
- Attachments:** A 'Browse...' button and a text area for attachments (0M/20M).
- Contact Information:** Fields for First Name (John), Last Name (John), Email (john@netbrain.com), and Phone Number (+1 9888346). A radio button for Preferred Contact Method is set to Email.
- Buttons:** Export, Cancel, and Submit.

The background shows a network map with various nodes and connections. A red arrow points from a red circle containing the number '1' to the 'Submit' button in the dialog box.

## Submit a Ticket for Unknown Object ID

When an unknown Object ID is discovered, you can choose to submit this Object ID to NetBrain so the platform team can update your built-in resource accordingly in the [Knowledge Cloud](#).

SysObjectID	Discovery Source	Discovery Time
1.3.6.1.4.1.18334.1.1.1.2.1.104.2.5	192.168.20.7	10/29.2018. 11:09:30 AM
1.3.6.1.4.1.30065.1.2759	172.25.94.2	

**Add Vendor Model**

OID: 1.3.6.1.4.1.18334.1.1.1.2.1.104.2.5

Device Type: 3Com Switch

Vendor: 3Com

Model:

Driver: 3Com-HP Comware Switch

CPU OID:

Memory OID:

Auto Update

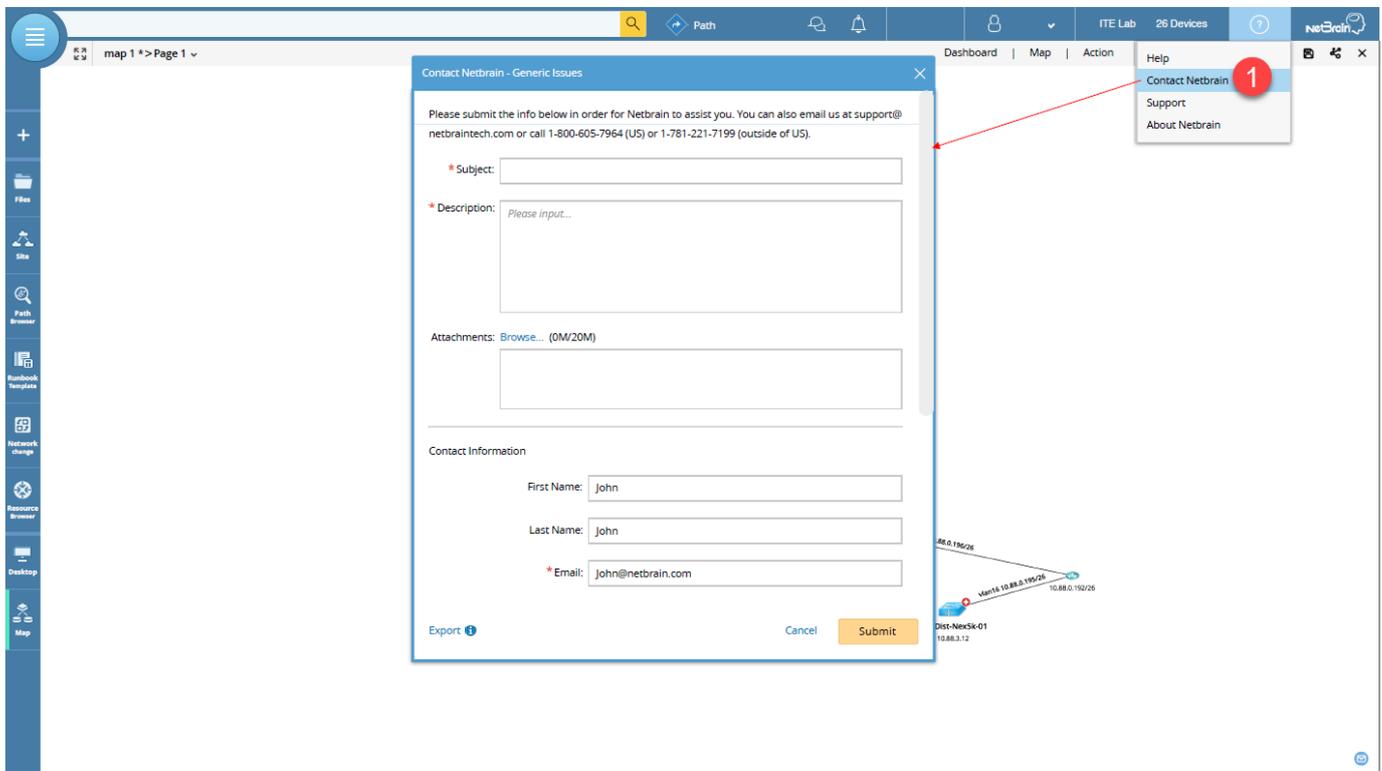
**Submit Object ID to Netbrain** ⓘ

Cancel Submit

- Discover Selected Device(s)
- Add to Vendor Model Table**
- Delete from Table

## Submit a Ticket for Generic Issues

To submit a generic issue that is not covered by a specific ticket type (path, map, benchmark, etc.), you can navigate to **Help > Contact NetBrain** to launch a generic issue request.

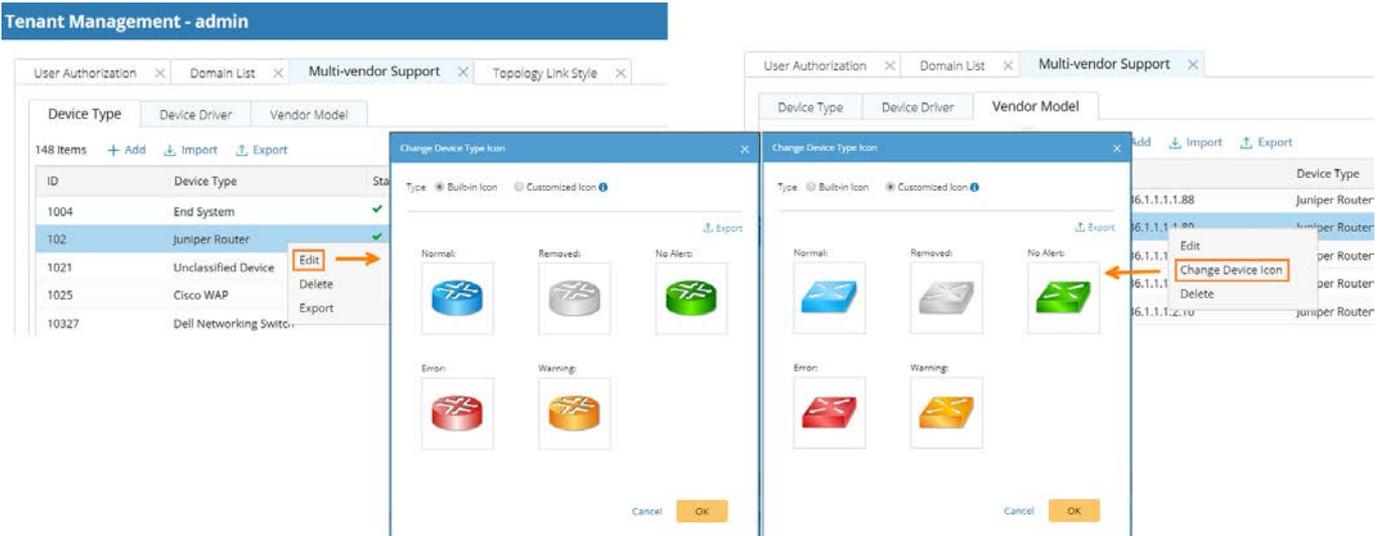


## 2.15. Map Enhancements

### 2.15.1. Customize Device Icon for Device Type and Vendor Model

In previous versions, the system provided a set of built-in icons for device types. IEv8.0 system allows the tenant admin to customize device icons for device types. As one device type contains more than one device

model, IEv8.0 also supports changing device icons for device models.

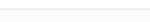
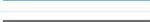
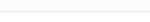


## 2.15.2. Define and Customize Topology Link Styles on Map

IEv8.0 visualizes the definitions of built-in topology link styles and allows admin to modify color, line shape, thickness, and conditions to apply these styles. Moreover, admin can define new styles and the conditions to apply the new style. Moreover, the system adds the capability to allow end users to change link styles on a map instantly.

## Modify Built-in Link Style Definition

Iev8.0 provides the following 10 built-in topology link styles. The definition of each style contains color, shape, thickness, and condition to apply this style on a map, all of which can be modified by the tenant admin.

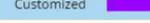
Style Name	Type	Link Color	Link Style	Link Thickness	Create Legend
Default	Built-in			1 px	
Trunk Link	Built-in			2 px	
MC-LAG Peer Link	Built-in			2 px	
MC-LAG Member Link	Built-in			2 px	
Fabric Path Link	Built-in			2 px	
Port Channel Link	Built-in			2 px	
Tunnel Link	Customized			2 px	
Wireless Link	Built-in			2 px	
HA Link	Built-in			1 px	
L2 Overlay Link	Built-in			1 px	

**Note:** To view the change on an existing map, users need to update the map first.

## Pre-Define New Link Styles for More Technologies

Besides modifying the built-in topology link styles, tenant admin can define new styles and related policies for more technologies. For example, you can add a new Access Link.

**Tenant Management - Initial\_Tenant** Operations

Style Name	Link Type	Link Color
Default	Built-in	
Trunk Link	Built-in	
MC-LAG Peer Link	Built-in	
MC-LAG Member Link	Built-in	
Fabric Path Link	Built-in	
Port Channel Link	Built-in	
Tunnel Link	Built-in	
Wireless Link	Built-in	
Access Link	Customized	

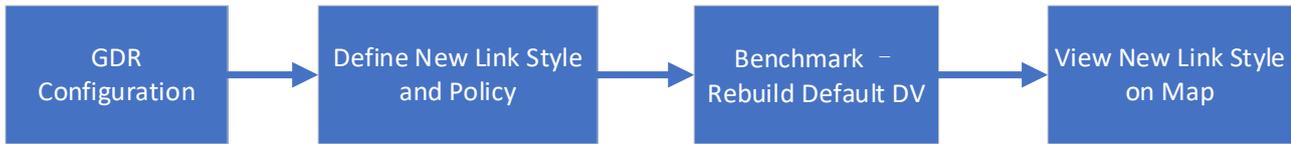
Search for device, configuration text. Map24 \* > Page 1 Dashboard | Map | Actions

BJ\*POP 172.24.255.8 f0/1 f0/23 BJ\_Acc\_Sw4 172.24.101.24

Legend

- Interface
- Default
- Access Link

## Reference Flow



- **GDR Configuration:** Interface properties referenced in policy definitions are sourced from Global Data Repository (GDR). The option “Record into Default Data View” of interface properties is required to be selected in the GDR configuration.

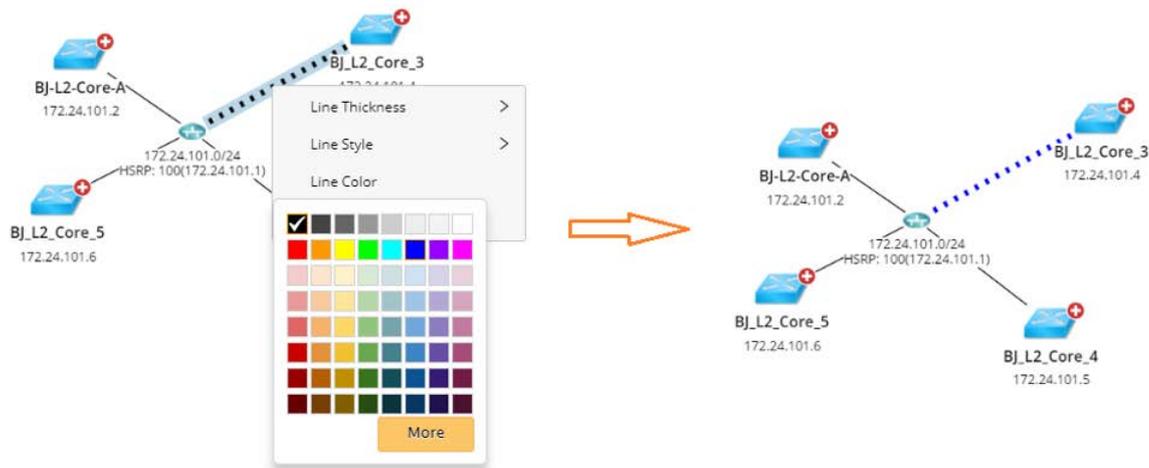
The screenshot shows the 'GDR Data Configuration' window with the 'Interface Properties' tab selected. Under 'Interface Type and Properties', a tree view shows 'Physical Interface' expanded to 'name'. To the right, the 'Record into Default Data View' checkbox is checked and highlighted with an orange box. Other options include 'Visible in Device Details Pane' (checked), 'Visible in Data View' (unchecked), 'Apply to' (All Types), 'Type' (System), and 'Description' (The interface name).

- **Rebuild Default Data View:** Enablement of the new link style on a map requires a refresh through the benchmark, and the option “Build Default Device Data View” must be checked.

The screenshot shows the 'Edit Benchmark Task' dialog. The 'Additional Operations after Benchmark' step is active. A table lists operations with checkboxes: 'Recalculate Dynamic Device Groups', 'Recalculate Site', and 'Build Default Device Data View'. The 'Build Default Device Data View' checkbox is checked and highlighted with an orange box. The 'Submit' button is visible at the bottom right.

## Customize Link Styles on Map

Users can change a link style on a map by right-clicking the link, and the customization applies to the single map only.



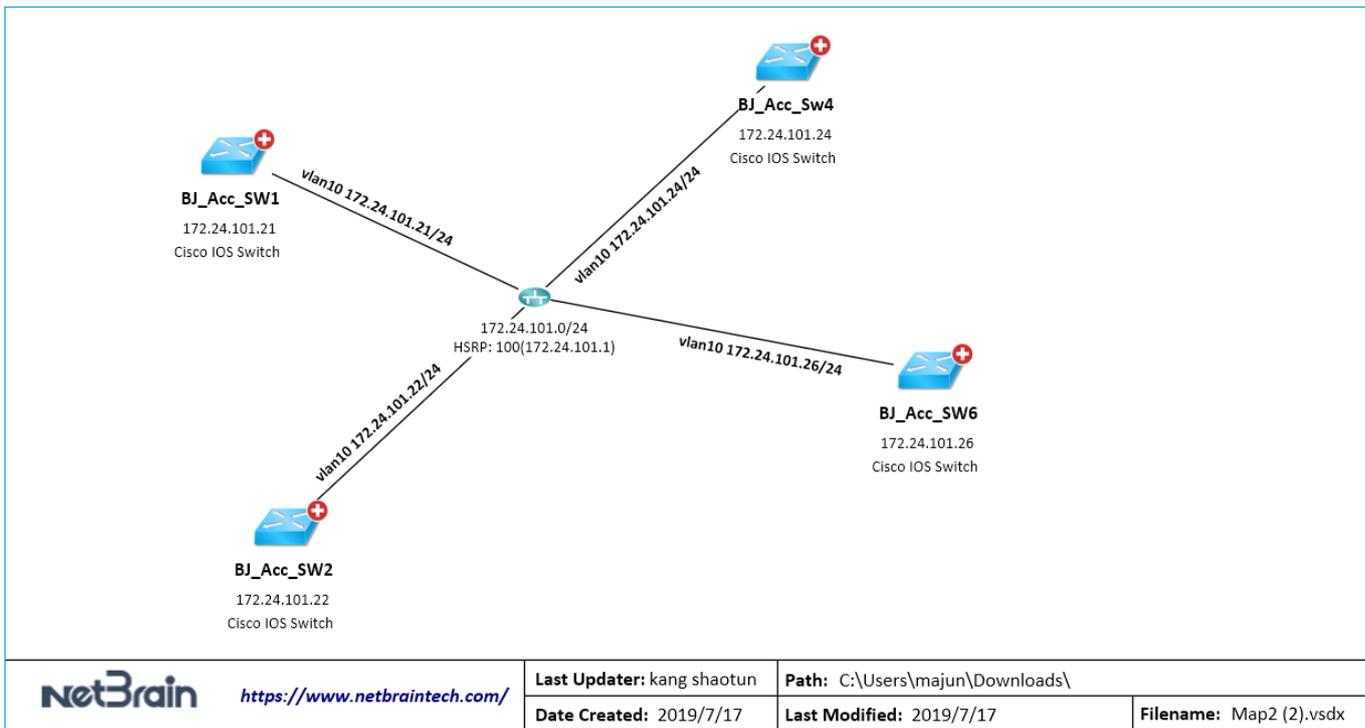
### Key Logic:

- The priorities of the following three types of link styles are: **highlight in data view** > **custom link style** > **built-in/pre-defined link style**.
- Only the user who has the map editing rights can save the link style changes, and the changes will not be lost after the map is refreshed or updated.

### 2.15.3. Use Built-in or Customized Visio Template to Export Visio Maps

IEv8.0 provides a built-in Visio template to export Visio maps. Each tenant can only have one Visio template, which will apply to all exported Visio maps for this tenant. Users are also allowed to import a customized Visio

template.



## 2.15.4. Separately Highlight Link and Port on Map

IEv8.0 allows users to separately highlight port and link of a device in the expanded style.

- Only Highlight Link

Map Settings

Example:

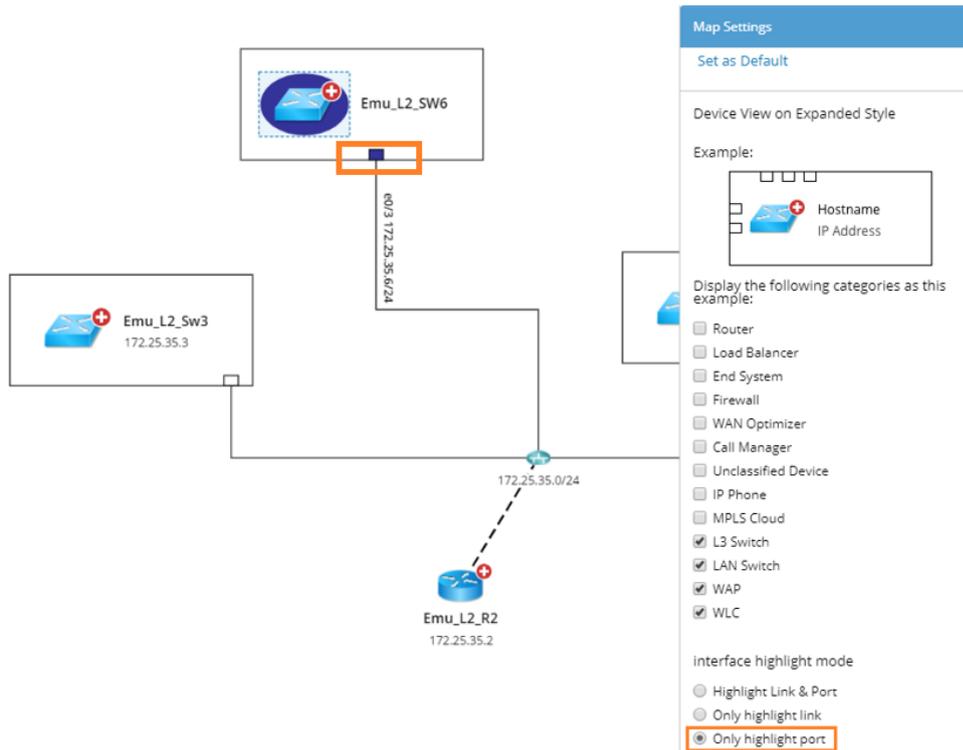
Display the following categories as this example:

- Router
- Load Balancer
- End System
- Firewall
- WAN Optimizer
- Call Manager
- Unclassified Device
- IP Phone
- MPLS Cloud
- L3 Switch
- LAN Switch
- WAP
- WLC

interface highlight mode

- Highlight Link & Port
- Only highlight link
- Only highlight port

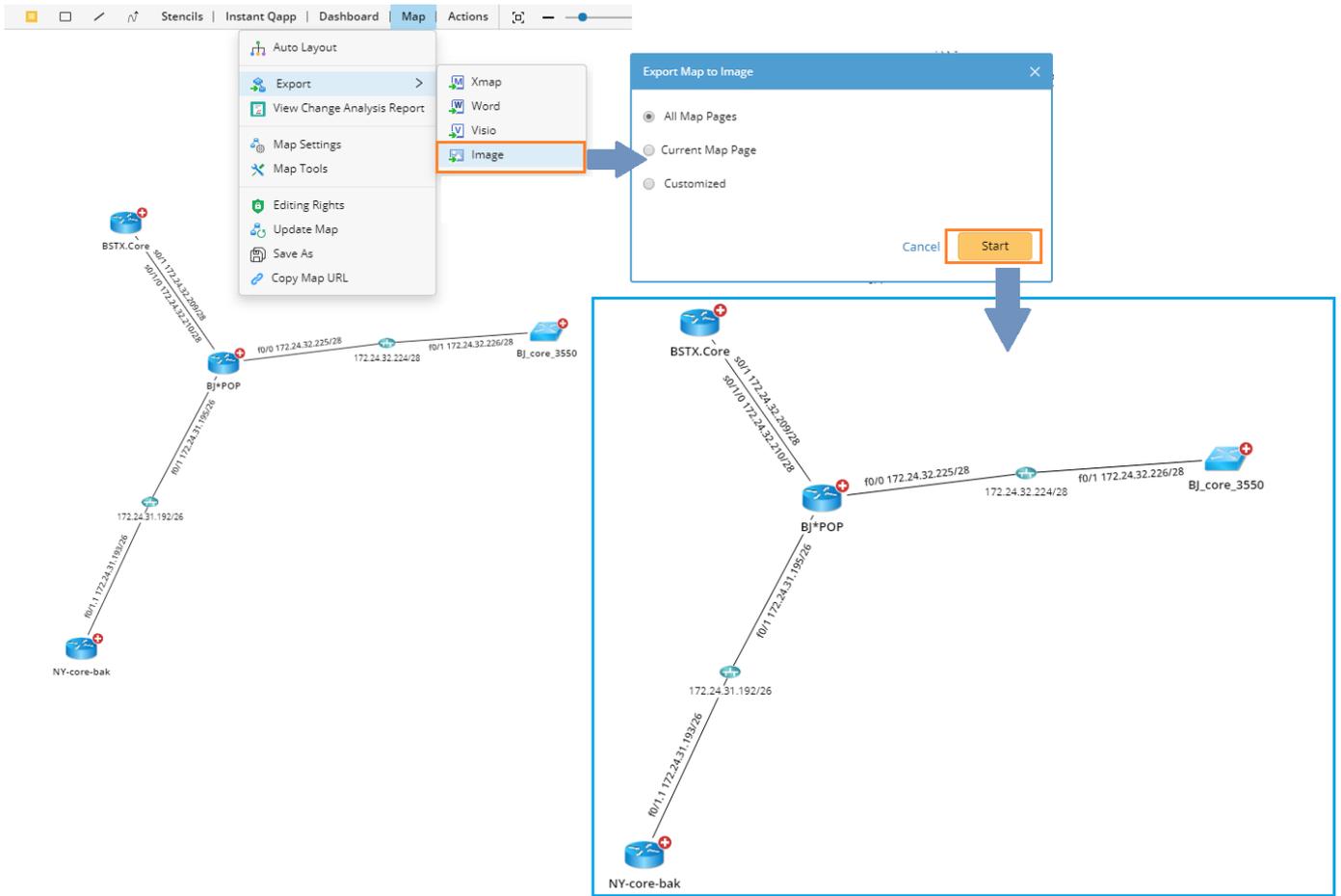
- Only Highlight Port:



## 2.15.5. Export Map to Image

IEv8.0 adds the capability to export a NetBrain map to an image file (.jpeg) in the local disk of users. Before starting the export process, users can specify the map pages to export.

By default, the system will export all map pages of the current map. JPG files will be compressed in a ZIP file.

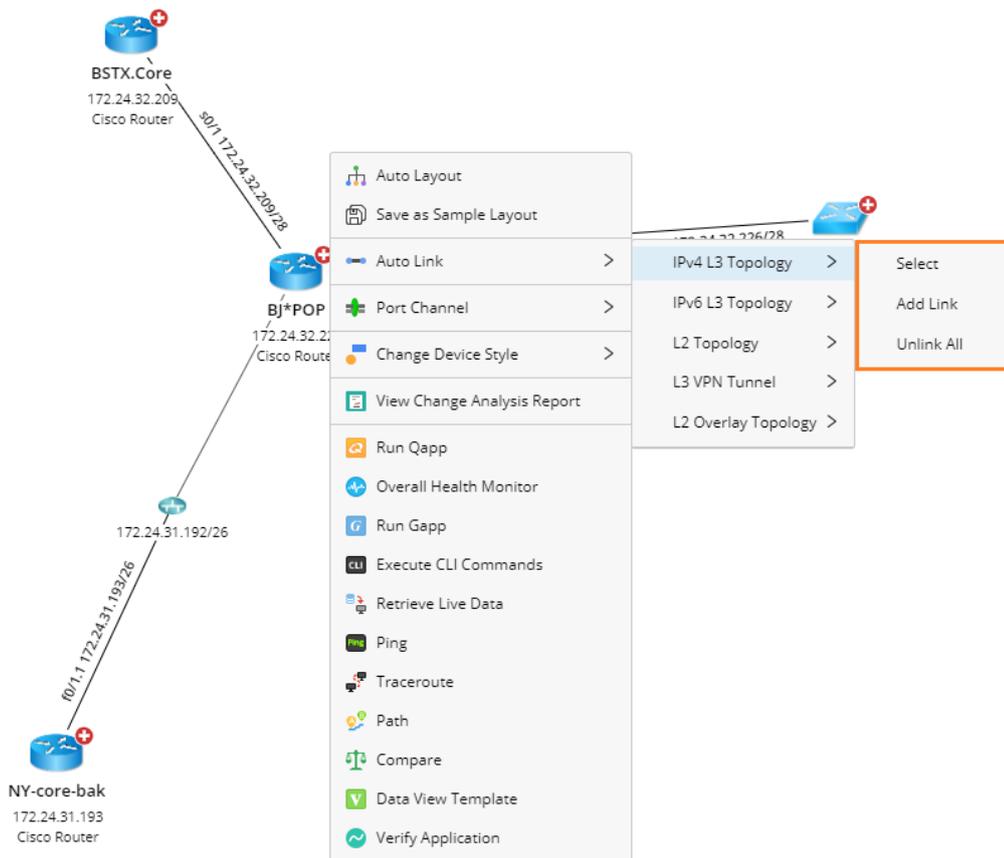


## 2.15.6. Auto Unlink Interfaces in a Device Group

In previous versions, it's labor-consuming for users to manually remove links from a map when they toggle to view links of a specific topology type by hiding the links of others.

IEv8.0 adds the Auto Unlink option in the context menu and introduces a few more changes:

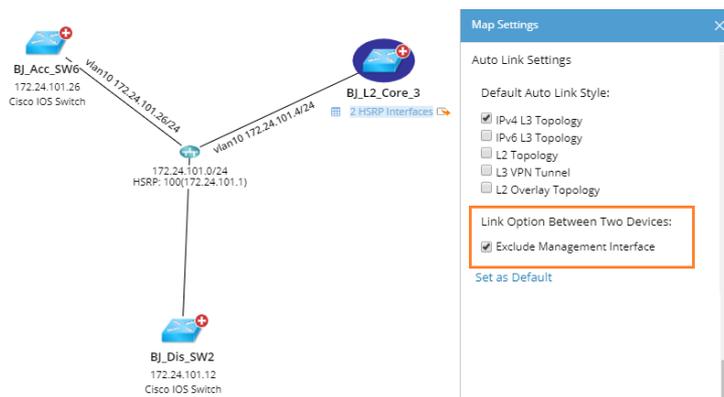
- **Unlink All** <sup>New</sup> — remove a batch of links for a specific topology type.
- **Select** <sup>New</sup> — select a batch of links for a topology type. With this option, users can identify which links belong to this topology type.
- **Add Link** — add a batch of links for a topology type, which is the same as the original Auto Link function.



## 2.15.7. Narrow Down Auto-link Scope and Count

### Enable to Exclude Management Links from a Map

IEv8.0 allows users to exclude management links from a map when using the auto-link function. By default, this option is enabled.



**Note:** Considering management links are usually managed in one subnet, IEv8.0 provides a new built-in plugin “recognize\_management\_interface” to identify management links by subnet matching. This plugin is executed along with

the Basic System Benchmark. Those interfaces within the specified subnets will be identified as management interfaces, and their interface property `isMgmtIntf` will be updated to “true” in the GDR.

## Limit the Number of One-Time Auto Links

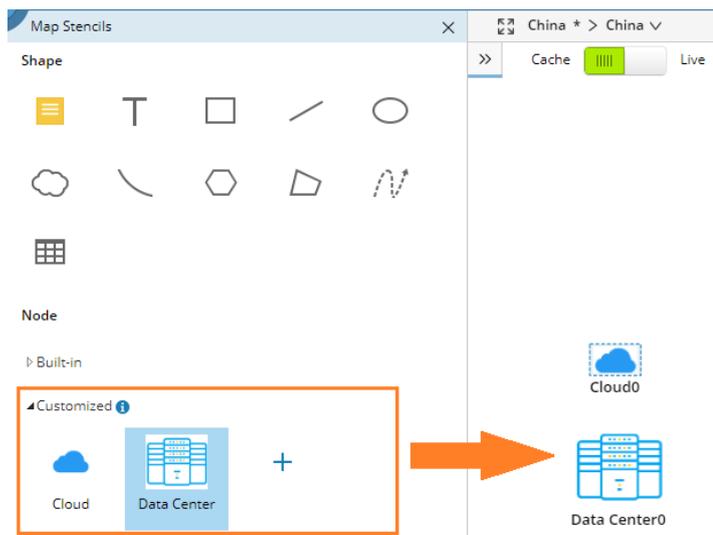
To improve the performance for the auto-link function, IEv8.0 adds control to limit that at most 50 devices can be auto-linked at one time on a map. Users have to repeat the auto-link action to link more devices.

**Tip:** This threshold is configurable in back-end config files.

## 2.15.8. Custom Node Icons

To meet the diverse needs for network mapping, IEv8.0 allows users to upload custom node icons to the Map Stencils pane. The uploaded node icons can be directly used for mapping with drag-and-drop.

**Note:** Only PNG is allowed.



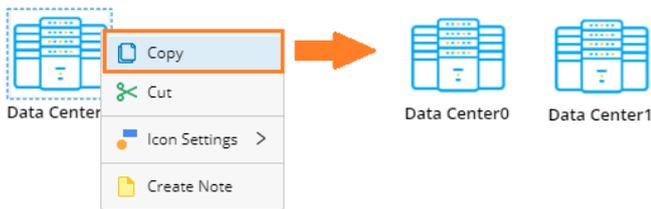
## 2.15.9. Free Text for Network Design/Troubleshooting Annotation

IEv8.0 adds the shape of Text to the Map Stencil pane. The style of the text can be customized, such as font, size, color, alignment, etc.



## 2.15.10. Copy, Cut and Paste Stencil Icons and Shapes

IEv8.0 adds the copy, cut, and paste functions for map components, including icons and shapes.



The following shortcut keys are allowed:

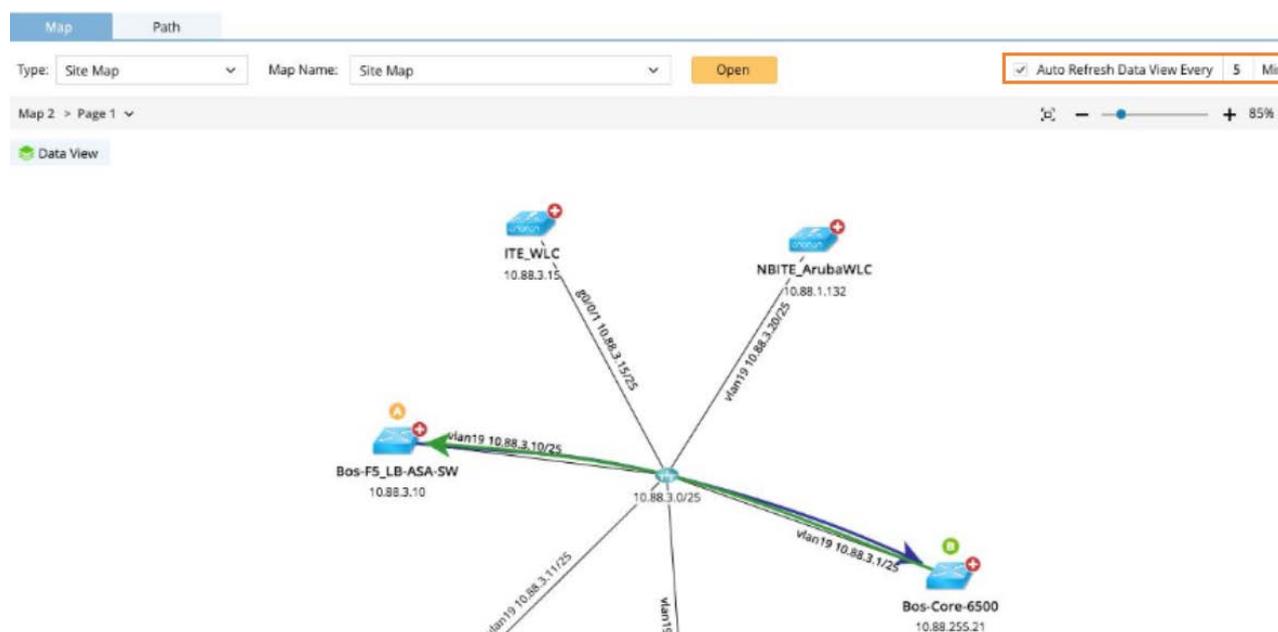
- **Copy** (CTL+C)
- **Cut** (CTL+X)
- **Paste** (CTL+V)

## 2.15.11. Map One-IP Table Entries with Drag-and-Drop

Launching the One-IP Table from the start menu will open a pane rather than a separate tab, so that users can map out devices and their L2 topology links with easy drag-and-drop, or compare table entries with the



automatically.



**Note:** Using live data to apply Dynamic Data View (DVT) is not supported in IEv8.0.

For more technical details about Data View 2.0 support, refer to the [API Signature Reference Document](#) (prepared by Automation Team and published at GitHub).

## API Enhancements to Path Calculation

Multiple logics have been optimized in the IEv8.0 path calculation. The display of path result has also been adjusted according to the IE design improvements. For more details, refer to [Enhancements to Path Calculation](#).

Moreover, the path calculation API has been adjusted as follows:

- A new API endpoint has been added to resolve device gateway;
- Calculate path API has been redesigned to:
  - Consume the response parameters (sourceGateway.type, sourceGateway.gatewayName, sourceGateway.payload) passed from the resolve device gateway API;
  - Configure advanced path settings, such as calculating L3 Active Path, continuing to calculate path when denied by policy/ACL rule, Enabling path fix-up logic, etc.

## Allow Creation of an Embedded Map via Triggered Automation

In IEv8.0, you can define a triggered-API automation task to use Qapp to create a map, call the task in your embedded map function and create an on-demand map in your portal using the input parameters in the

## Qapp.

The screenshot displays the 'API Service Stub' configuration window. On the left, the 'API Stub Name' is 'stub1' and the 'Trigger Option' is 'Real-time'. A dropdown menu under 'How would you like to create the map?' is set to 'Use Qapp to Create a Map'. Below this, a diagram shows a network path with a red alert icon. A code editor window is overlaid, showing a JSON configuration for 'triggerAPIConfig' with fields for 'stubName', 'userName', 'deviceName', and 'input' parameters. The 'input' section defines three parameters: 'receiver' (172.24.30.2), 'sourceIP' (172.19.43.2), and 'groupIP' (239.70.8.1). At the bottom, the 'Multicasting Reverse Path' tab is active, showing input fields for 'Receiver IP: 172.24.30.2', 'Source IP: 172.19.43.2', and 'Group IP: 239.70.8.1'. The interface also shows 'Tenant: xxf\_tenant' and 'Domain: domain\_20190527'.

```
var $triggerAPIConfig = {
  "stubName": "reversePath",
  "userName": "admin",
  "deviceName": "GW2Lab",
  //Qapp input_variable_parameters
  "input": [{
    "desc": "Multicast Receiver IP Address",
    "label": "Receiver IP",
    "name": "receiver",
    "value": "172.24.30.2"
  }, {
    "desc": "Multicast Source IP Address",
    "label": "Source IP",
    "name": "sourceIP",
    "value": "172.19.43.2"
  }, {
    "desc": "Multicast Group IP Address",
    "label": "Group IP",
    "name": "groupIP",
    "value": "239.70.8.1"
  }
]}
```

## 2.16. Runbook Enhancements

IEv8.0 introduces the following new features and enhancements to Runbook automation, including:

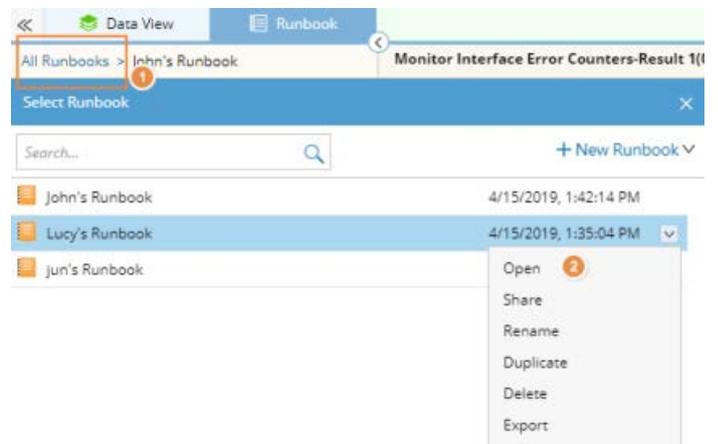
- [Personalize Default Runbook](#)
- [Show Alert Icon for Execution Results](#)
- [Merge Same-Type Action Nodes](#)
- [Move Action Nodes](#)
- [Max Node Count Limitation](#)
- [New Action Nodes](#)
- [Enhanced Compare Node](#)
- [Enhanced CLI Node](#)
- [Keyboard Shortcuts](#)

## 2.16.1. Personalize Default Runbook

In previous versions, a Default Runbook is used to accommodate activities performed on a map when there is no runbook intendedly created. As a result, activities performed by different users cannot be distinguished.

In IEv8.0, to make each user have a personal runbook to exclusively document his or her own activities, the default runbook is renamed and personalized, for example, John's Runbook.

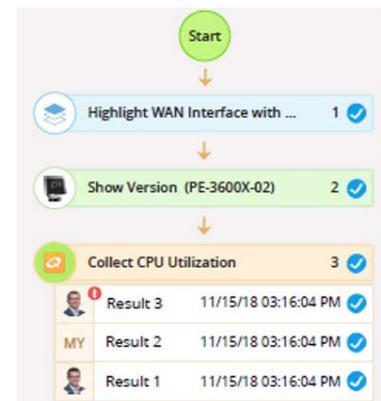
**Compatibility Note:** All Default Runbooks in IEv7.x are auto-renamed to **Auto Saved Runbook** in IEv8.0.



## 2.16.2. Show Alert Icon for Execution Results

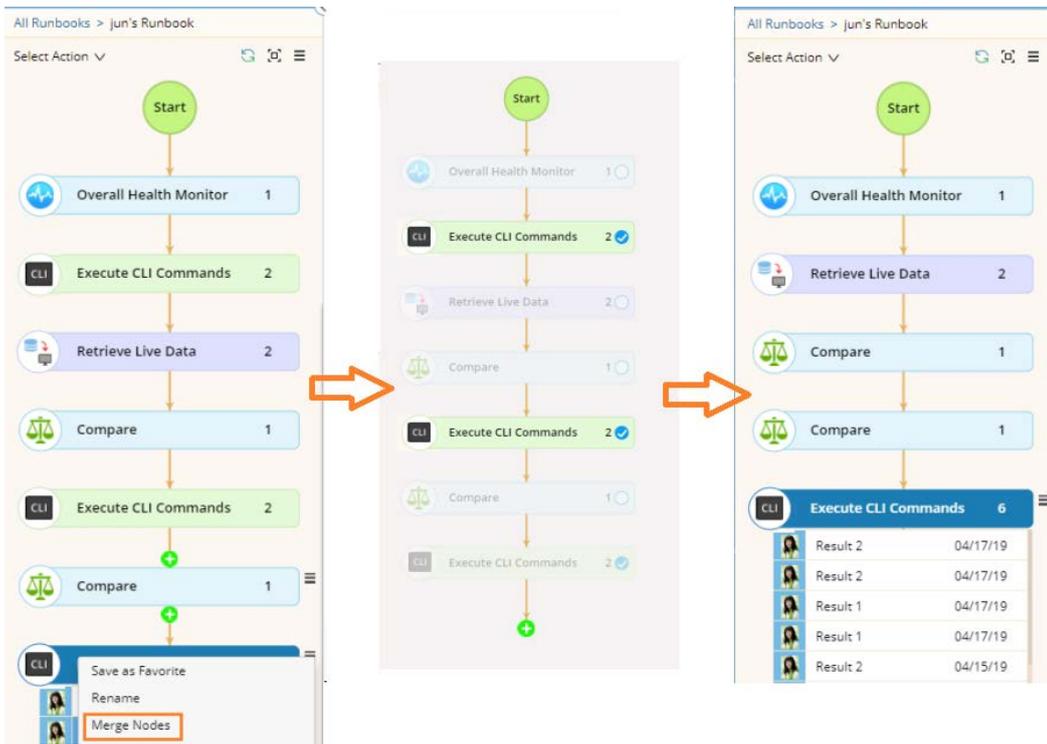
In IEv8.0, an alert icon is displayed on the execution results of an Action node to notify users that alerts have been generated during the execution cycle. Moreover, the executor's avatar is also displayed on each action execution result.

As soon as an alert is acknowledged, the alert icon will be cleared.



## 2.16.3. Merge Same-Type Action Nodes

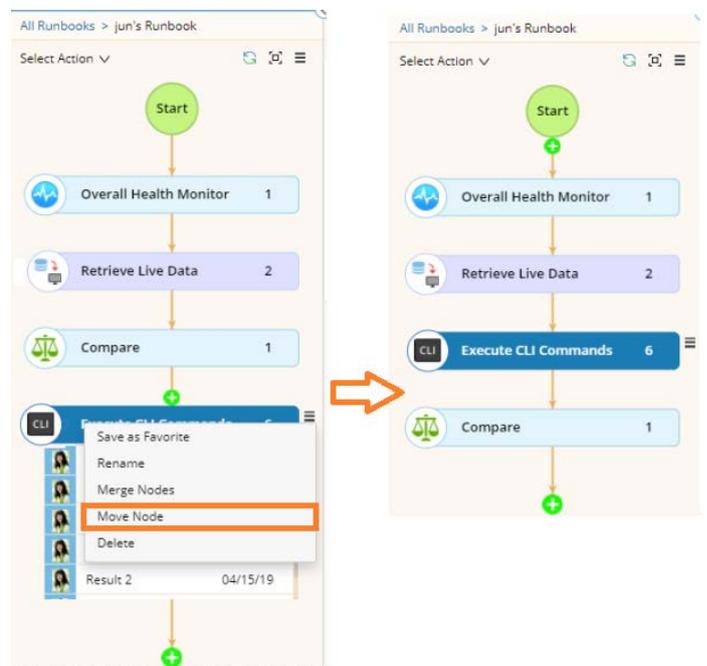
To make a runbook flowchart more organizable, IEv8.0 allows users to merge same-type action nodes. After the merging, all historical execution results are re-ranked by execution timestamp.



**Tip:** To avoid duplicate result names under one action node, users can rename execution results before merging nodes.

## 2.16.4. Move Action Nodes

A runbook workflow accurately reflects the sequential order of all executed actions. To allow users to flexibly adjust the workflow order rather than starting it over, IEv8.0 adds a Move button.



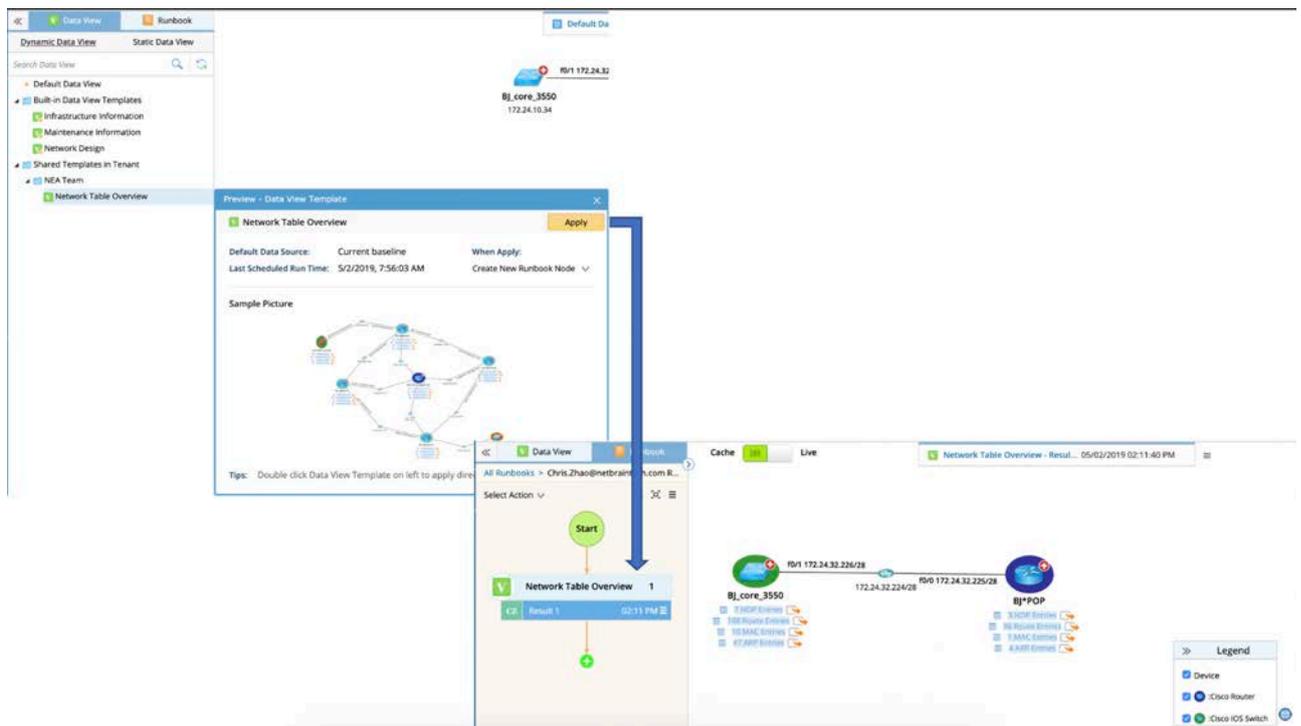
## 2.16.5. Limit Max. Node Count

When a Runbook contains too many action nodes, the intention to document user activities and results are not strong enough somehow, and users will also get lost there due to poor readability. To improve this, IEv8.0 limits the maximum count of action nodes inside a Runbook to **200**.

## 2.16.6. New Action Nodes

To adapt to the new end-user troubleshooting flow, three new types of Action Nodes are added to a Runbook.

- [Verify Application](#) — used to verify applications and application-aware paths.
- [Ansible Task](#) — used to integrate with Ansible to define, execute playbooks and visualize results in Change Management Runbooks.
- [Data View Template](#) — the most popular use case for this node is: when a dynamic data view is applied to a map, it will be auto-added to the current runbook. This node can also be manually added to a runbook and can be executed.



## 2.16.7. Enhanced Compare Node

The comparison feature is mainly used to compare network data between two periods to check whether a network change occurs. With this feature, users can quickly find out the network changes to identify the root cause of a network problem.

### Visualize Data Existence and View Details

In IEv8.0, users can view the data information about a specific device.

Input		Dataset ①	Dataset ②
Compare Same Device		Current Baseline	Default Live
All Devices	☑ All Data		
BJ-L2-Core-A	☑ Configuration File	☑ View	✗
BJ-L2-coreB	☑ Route Table	☑	✗
BJ_core_3550	☑ ARP Table	☑	✗
hostname_change_test	☑ MAC Table	☑	☑
qapp-force10	☑ NDP Table	☑	✗
	☑ STP Table	☑	☑

- Visualize data existence by using the two icons: ☑ and ✗.

For example, **BJ-L2-Core-A** has an ARP table in the current baseline but not in the default live. Users can know the ARP table has changed before comparing.

- View device data in detail.

For example, users can click the **View** button to see the current baseline of the configuration file for **BJ-L2-Core-A**.

### Export Comparison Result

To enhance the collaboration flow of Runbook Automation, IEv8.0 adds the capability to export the Runbook comparison result to a portable file, so that users can directly share the comparison result with other colleagues. For example: compare the retrieved live data at different time points, and export the comparison result for sharing.

The exported files are compressed in a zip file. Unzip it and double-click the **summary.html** file to view the summary report. Then click on “Y” to view changes in detail.

**Comparison Input:** 4 Device(s), 9 Data Type(s): Dataset1 - Current Baseline, Dataset2 - Retrieve Live Data - Result 1  
**Comparison Result:** 4 of 4 Device(s) Changed, 17 of 36 Data Type(s) Changed  
 Tenant: Initial Tenant Domain: NYTest Map: Private/Map2 Runbook Node: zhaoxu Runbook/Compare/Result 1(09/27/2019 03:22:19 AM UTC)

Hostname	Configuration File	Route Table	ARP Table	MAC Table	NDP Table	STP Table	BGP Advertised-route Table	NAT Table	NAT Table[Real-time]
BJ*POP	N	N	Y	Y	N	N	N	N	N
BJ_core_3550	N	Y	Y	Y	Y	Y	N	N	N
BSTX-Core	N	Y	Y	Y	N	N	Y	Y	Y
NY-core-bak	N	Y	Y	Y	Y	N	N	N	N

## 2.16.8. Enhanced CLI Node

To better serve the new [NetBrain Smart CLI](#) feature/tool, the following enhancements have been done to CLI Command node:

- Add the Compare function to compare CLI command outputs between two baselines.
- Allow users to select texts in CLI command outputs to create a map note.
- Auto compare parsed variables in CLI command outputs with Golden Baseline and highlight the mismatched ones.
- Auto display parsed variables on a map. In previous versions, users have to manually drag these variables to a map.

## 2.16.9. Keyboard Shortcuts

The following keyboard shortcuts are enabled for user operations in a Runbook.

Keyboard Shortcut	Action
Up Arrow (↑) Down Arrow (↓)	Toggle between execution results in an action node.

Keyboard Shortcut	Action
Left Arrow (→) Right Arrow (←)	Toggle between action nodes.
Delete	Delete an action or an execution result.

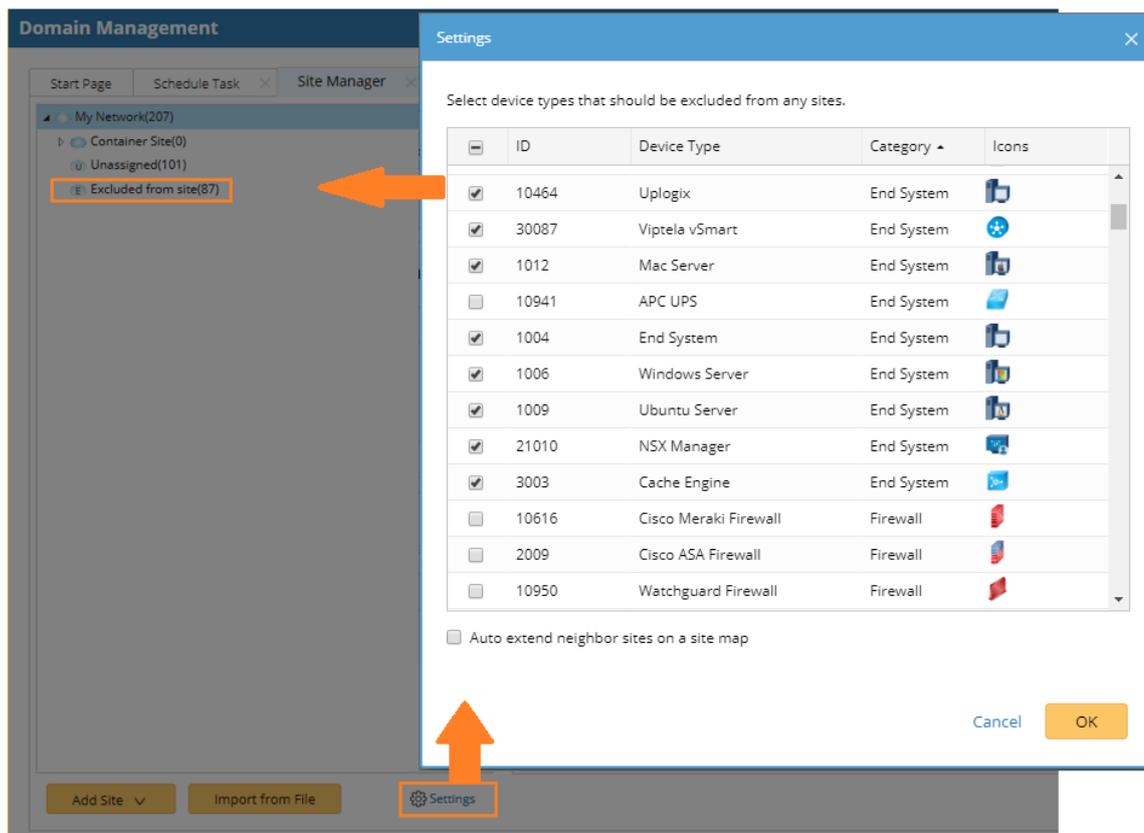
## 2.17. Site Enhancements

### 2.17.1. Exclude Specified Device Types from Site Assignment

During a domain setup process, one of the power users' tasks is to complete site management by assigning many "unassigned" devices to target sites they belong to. However, in real-world cases, many End Systems and WAPs, such as PCs and printers, are not involved in any site build or topology build. It is usually time-consuming to go through the long "unassigned" list to get rid of them.

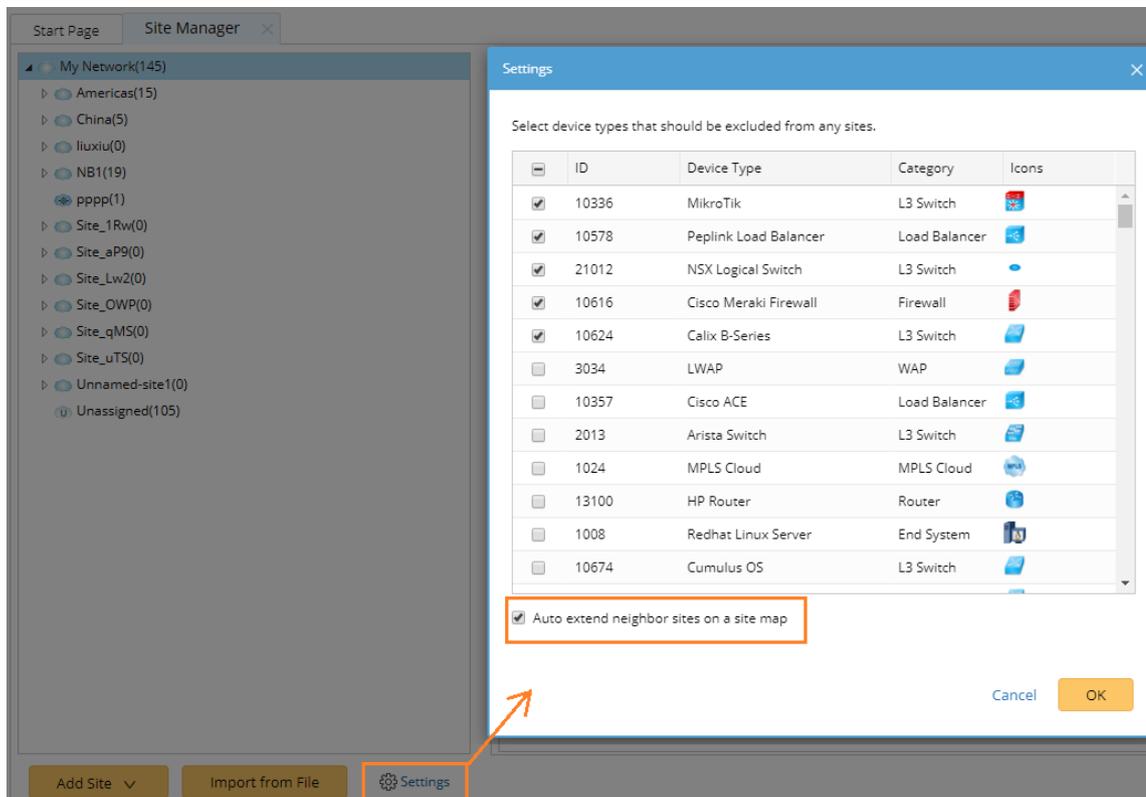
To elevate the operation efficiency of site management, IEv8.0 allows power users to exclude specific device types from the scope at the beginning. With this setting configured, devices of the specified devices types are

categorized into a new category “Excluded from site” in the site tree and cannot be assigned to any site.



## 2.17.2. Allow Hiding Neighbor Sites from a Site Map

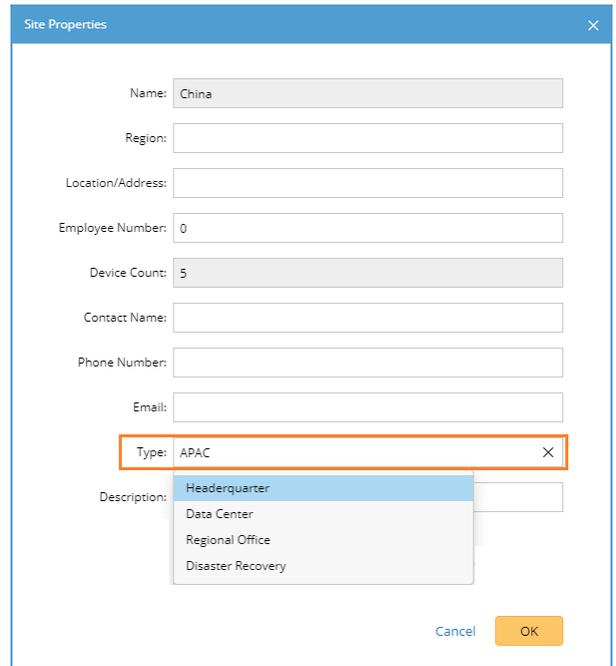
Iv8.0 adds control to hide neighbor sites from a site map to avoid mapping unnecessary neighbors. By default, this function is enabled.



To unhide neighbor sites, go to **Site Manager > Settings** and select the **Auto extend neighbor sites on a site map** check box.

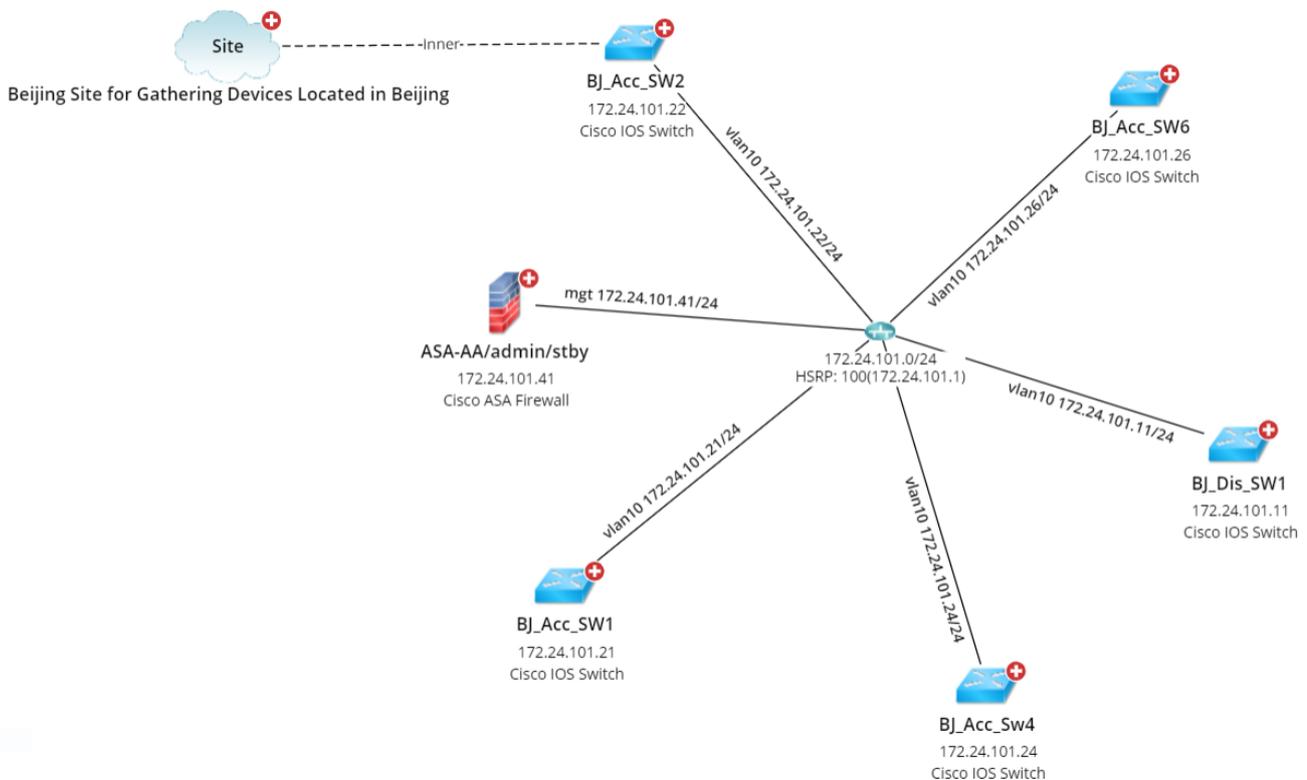
### 2.17.3. Allow User Input for Site Type

The “Site Type” property is used to label the usage of a site, such as headquarter, data center, regional office, or disaster recovery. Besides these built-in options, IEv8.0 allows custom user input to support more definitions, such as by location, by network architecture or by branch.



### 2.17.4. Move Site Name Underneath Site Icon to Show Full Site Name

To fully display a site name on a map despite the scale of a site icon, IEv8.0 changes the visual design of the site icon by moving the site name underneath the site icon.



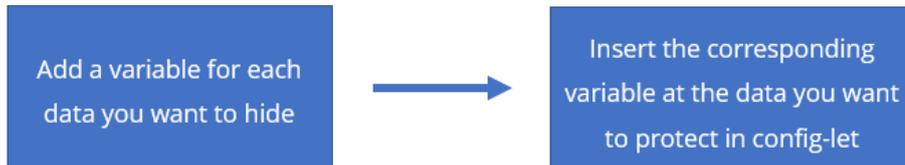
## 2.18. Enhancements to Change Management

### 2.18.1. Use Variables in Config-let

The change configs in Change Management (CM) are displayed in plain text, so there are security risks in some cases. For example, when users modify the device login password in batches through a CM task, it is not secure because passwords are displayed in plain text. Another example is that when users perform ACL security configuration for traffic access control through a CM task, they face the risks of security configuration reveal.

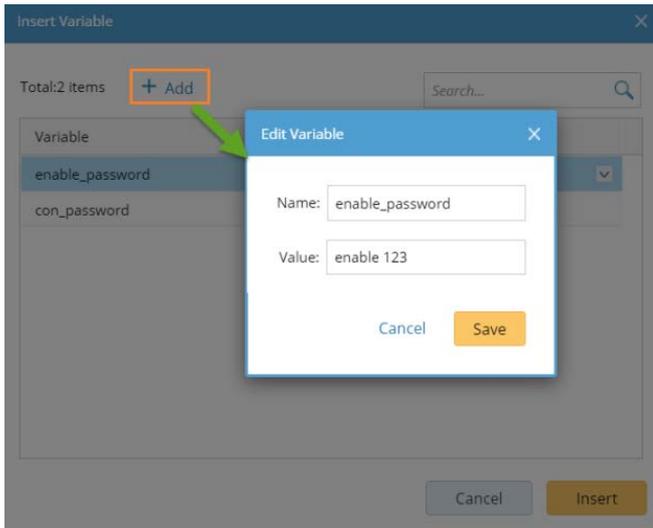
To resolve the security vulnerability and make change config more flexible, IEv8.0 enables users to use variables in config-let. The creator can hide sensitive data with variables when defining a Change node. For example, they can use variables to hide the passwords in a config-let. Only users with specific privileges can view or edit the values of the variables.

#### Use Flow



# 1. Add Variables

At the **Define Change** node, click **Insert Variable** to add variables and assign values.

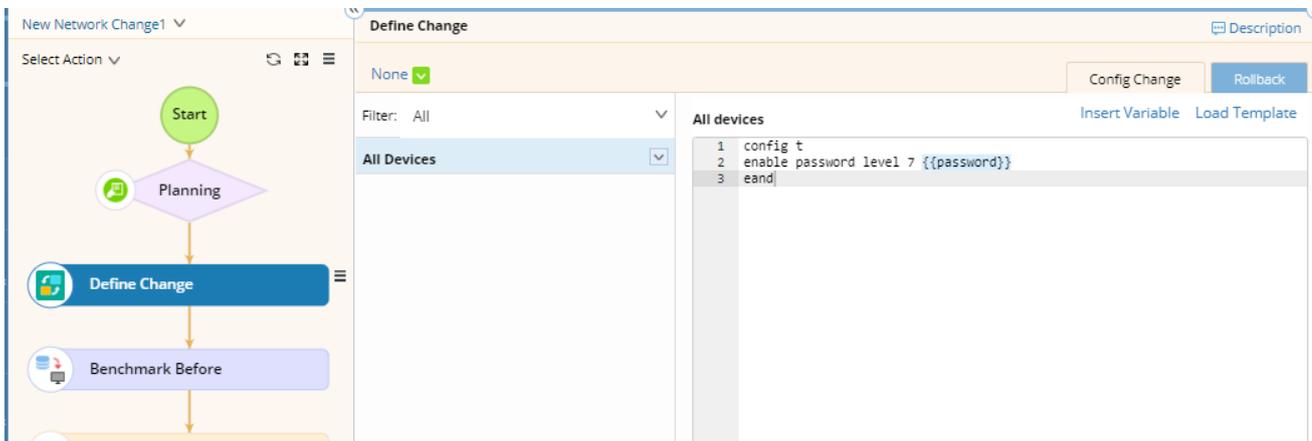


# 2. Replace Data with Variables

Insert the corresponding variable to replace sensitive data you want to protect. The value will be decrypted to form the actual commands sent to the device when the system executes the changes. Enter `{{` and select the corresponding variable.



The system only shows variable names and hide specific data in the commands. The execution logs also hide the specific data.



Only users with specific privileges can edit or view the variables and values.

Privilege	View Variable	Add/Delete/Edit Variable
Create Network Change	Yes	Yes
Approve Network Change	Yes	No
Execute Network Change	Yes	No
View Network Change	No	No

## 2.18.2. Schedule Change Management Task via REST APIs

IEv8.0 allows users to schedule a network change task via Restful APIs. See [North-bound API](#) for more details.

**Note:** A change management task can only be scheduled to run after being approved.

## 2.19. More Feature Enhancements

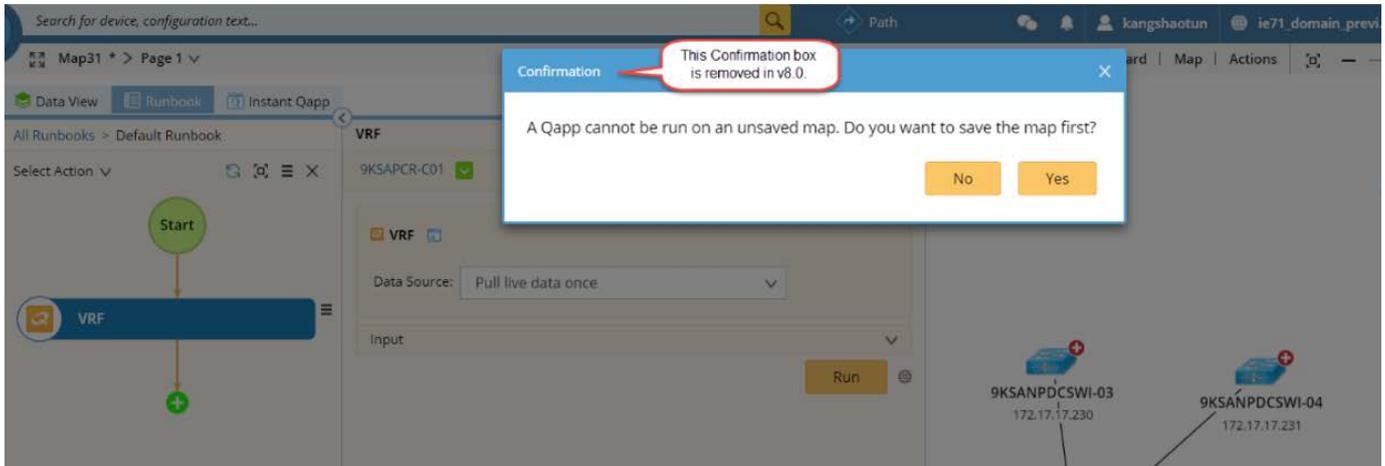
### 2.19.1. Auto Authenticate User Accounts during Login

To simplify user input during a login process, the authentication server selection, for example, LDAP/AD, TACACS+, or Local (NetBrain), is removed from IEv8.0. Instead, the system will auto-poll user account credentials of all available authentications to validate the user input.

If users are sure of the authentication server, adding “ServerName\” as a suffix to their usernames will accelerate the validating process. As shown in the figure, the user account “admin” is manually created in the system, so “netbrain\” can be added as the suffix.

## 2.19.2. Allow Users to Run Automation Actions Directly on an Unsaved Map

In previous versions, users had to save current maps first when executing on-demand actions. To improve the user experience, IeV8.0 allows users to execute actions directly on unsaved maps.

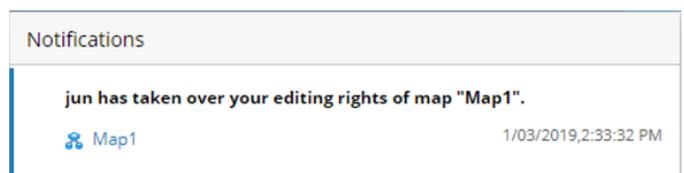


## 2.19.3. Loosen Map Editing Rights to Enhance Collaboration

Each map file can only have one editor. If the editor of a map file does not release or transfer the editing rights before taking a vacation or resigning, the map file can no longer be modified, updated, and deleted by any other users in the system. This scenario not only hampers team collaboration but also causes many useful map files outdated and many junk map files saved in the database.

To improve this user experience, map editing rights is loosened from the following aspects:

- Users with the "Domain Management" privilege can directly delete any junk public map files of which editing rights are unclaimed or occupied by other users. The map editors will be notified by a system message or by an email.
- Users with the "Domain Management" privilege can take over the editing rights of any public map file without waiting for approval. The map editors will be notified by a system message or by an email.



- The editing rights of a site map and a system device group map will be released immediately after the system admin forces the map editor to log out.

## 2.19.4. Enhance Device Group to Group and Map Interfaces

To offer a batch of interfaces to serve other features, such as Dynamic Map, Golden Baseline, Scheduled Qapps and etc., I Ev8.0 allows users to group interfaces based on common interface properties. For example, a device group for interfaces configured with OSPF 10.

The screenshot shows a web interface for configuring a Device Group named 'OSPF 10'. The interface includes a search bar and a table listing devices and their associated interfaces.

Hostname	Mgmt IP	Interface	Interface IP
BJ*POP	172.24.31.195	FastEtherne...	172.24.31...
BJ*POP	172.24.31.195	Loopback8...	172.24.255...
BJ*POP	172.24.31.195	Serial0/1/0	172.24.32.2...
BJ_L2_test_1	172.24.33.10	Vlan10	172.24.33.1...
BJ_L2_test_1	172.24.33.10	Vlan30	172.24.34.6...
BJ_L2_test_1	172.24.33.10	Vlan50	172.24.34.1...
BJ_L2_test_1	172.24.33.10	Vlan40	172.24.34.1...
Emu_MV_XR	172.25.43.3	GigabitEthe...	172.25.43.3...
GW2Lab	10.10.7.253	GigabitEthe...	172.24.30.1...
LA.DIS,1	172.24.32.66	FastEtherne...	172.24.32.6...

To do so, use the dynamic search function and set the condition: Routing Protocol contains OSPF 10. The result will be dynamically searched out as follows.

The screenshot shows the 'Dynamic Search' configuration interface. The 'Boolean Expression' is set to 'A', and the search criteria are defined as 'Routing Protocol' contains 'OSPF 10'. The search results are displayed in a table below.

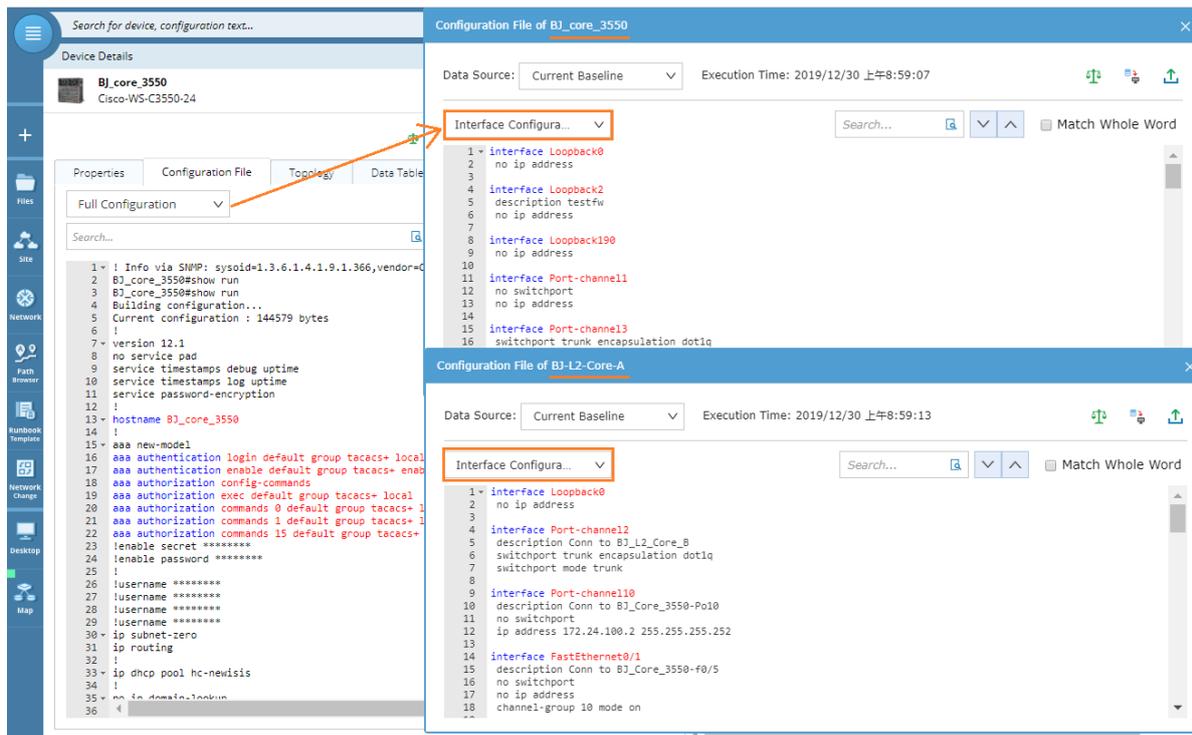
Device Name	Vendor	Model	Management IP	Interface	Interface IP
BJ_L2_test_1	Cisco	catalyst37xxStack	172.24.33.10	Vlan10	172.24.33.10/26
BJ*POP	Cisco	2811	172.24.31.195	FastEthernet0/1	172.24.31.195/26
NXB\Router/3	Cisco	2501	172.24.32.82	Serial0	172.24.32.82/28
LA.DIS,1	Cisco	2621	172.24.32.66	FastEthernet0/0	172.24.32.66/26
NY_DIS_1	Cisco	2611	172.24.31.66	Loopback90000	172.24.255.12/32



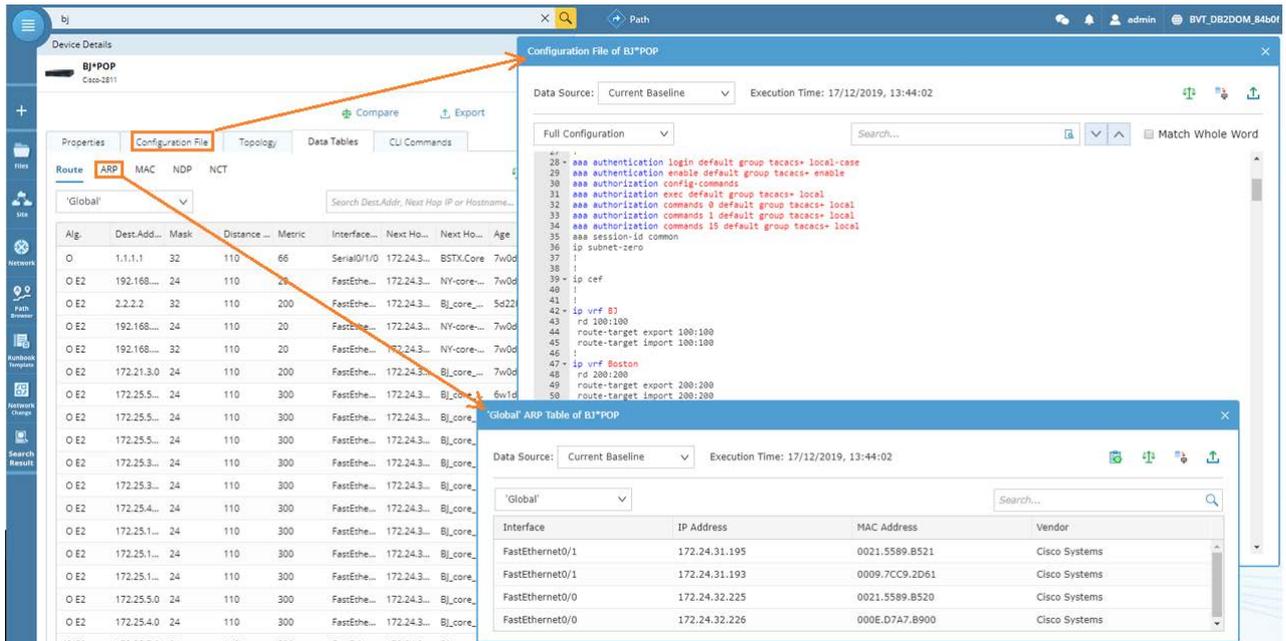
## 2.19.5. Support the Display of Multiple Device Data Dialogues Concurrently

IEv8.0 enables users to concurrently launch multiple dialogues from the Device Detail pane by introducing non-modal dialogue (resizable). As a benefit, users can view information from multiple device data tables in one screen under the following circumstances:

- Compare the interface configurations between a pair of neighbor devices on a map to troubleshoot the IP mismatch issue.



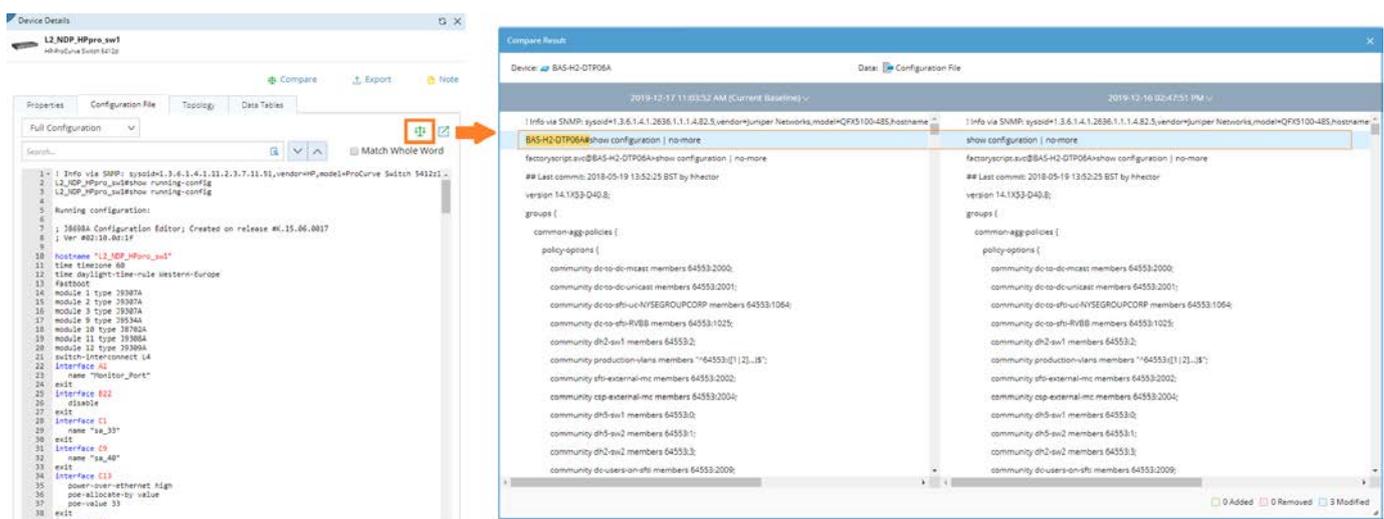
- Check the configuration file, route table and ARP table for a single problematic device.



## 2.19.6. Enhanced Device Data Comparison Flow

IEv8.0 enables users to directly view the comparison result for device data from the Device Detail pane, including configurations and a variety of data tables.

When users click the Compare button on the Configuration File or Data Table tab, the system directly displays the comparison result rather than launching a Runbook, which requires a few more clicks.



## 2.19.7. Enhancements to API Triggered Diagnosis

### Open Context Map

Based on input device types, the following methods are supported to create context maps:

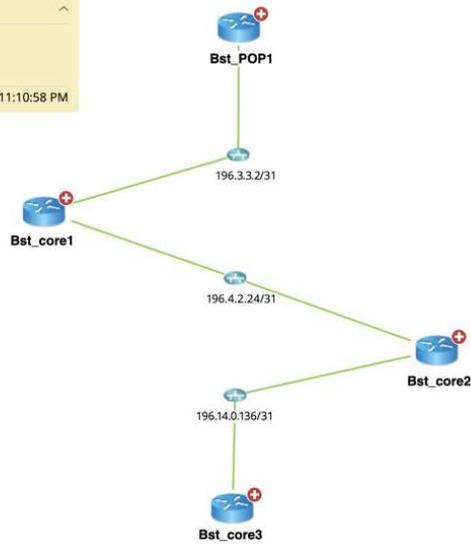
- If the input device is a legacy device, the corresponding context map can be created based on the input device name.
- If the input device is a Cisco ACI node, the corresponding context map can be created based on input APIC/node, etc.

Type	Variable Name	Default Value
API Service Stub		
*	\$stub_name	stub1
	\$mode	Real-Time
	\$max_waiting_hours	1
*	\$domain_name	BJ_rack
Map Creation		
*	\$apic	N/A
	\$device	N/A
	\$pod_id	N/A
	\$tenant_name	N/A
	\$svrf_name	N/A
	\$application_name	N/A

### Create Map by Qapp

API-triggered Qapp provides a flexible option to create advanced maps, such as a Multicasting tree map or a VPLS map, as complex logic can be easily defined in Qapp.

(S, G)  
(1.1.1.1, 239.77.1.1)  
Chris.Zhao@netbraintech.c... 6/27/2019, 11:10:58 PM



## Support Definition and Execution of New Runbook Nodes

Two new node types (Data View Template node/Verify Application node) have been added to IEv8.0 Runbook. The definition and execution of these new nodes are fully supported by API-Triggered Diagnosis.

For more details about the usage of the two nodes in IEv8.0, refer to [Enhancements to Runbook](#).

**Example:** New Data View Template Node under Runbook Template

API Service Stub

API Stub Name: stub1

Description:

Trigger Option: Real-time Edit

Task will be run automatically when triggered. Max wait time: 1 Hrs

1 How would you like to create the map?

Map Device and Its Neighbors

Include Device's Neighbors IPv4 L3 Topology

Sample

2 Would you like to add a Runbook? (Optional)

My Runbook Templates/DVT node?userId=... Browse Runbooks

Type	Variable Name	Default Value
API Service Stub		
Map Creation		
My Runbook Templates/DV...		
<input checked="" type="checkbox"/> BGP Overall	* \$device	
	* \$data_source	Pull live data once

Cancel Save

## Example: New Verify Application Node under Runbook Template

The screenshot shows the 'API Service Stub' configuration window. On the left, the 'API Stub Name' is 'stub1'. Under 'How would you like to create the map?', 'Include Device's Neighbors' is checked, and 'IPv4 L3 Topology' is selected. A sample map shows a device connected to an interface. On the right, the 'Variable Input' table lists variables for the 'Verify Application' node, which is highlighted with a red box.

Type	Variable Name	Default Value
API Service Stub		
Map Creation		
My Runbook Templates/DV...		
<input checked="" type="checkbox"/>	Verify Application	Advanced Settings
*	\$device	
*	\$data_source	live network
	\$max_relevant_application	3
	\$max_path_of_application	5
	\$share_alert	

## Two New Ways to Create a Map via API-Triggered Task

- Draw multiple devices and their neighbors and auto-link them on a map.

The screenshot shows the 'API Service Stub' configuration window. Under 'How would you like to create the map?', 'Auto Link All Devices' is checked, and 'IPv4 L3 Topology' is selected. A sample map shows three devices (Device 1, Device 2, Device 3) connected in a triangle. On the right, the 'Variable Input' table lists variables for the 'API Service Stub' and 'Map Creation' nodes.

Type	Variable Name	Default Value
API Service Stub		
*	\$stub_name	stub1
	\$mode	Real-Time
	\$max_waiting_hours	1
*	\$domain_name	BVT_DB2DOM_cdc8
Map Creation		
*	\$devices	N/A
	\$auto_link	true
	\$auto_link_type	L3_Topo_Type
	\$include_neighbor	false
	\$neighbor_type	

- Improve the [path API](#) to draw multicast reverse paths. Set `$path_type` to multicast, assign the multicast receiver address value to `$destination`, multicast source value to the `$source`, group value

to \$group, and call APIs to draw the multicast path.

API Service Stub
✕

API Stub Name:

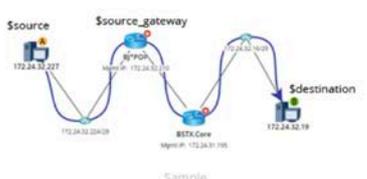
Description:

Trigger Option: Real-time Edit

Task will be run automatically when triggered. Max wait time: 1 Hrs.

**1** How would you like to create the map?

Map a Path



Sample

**2** Would you like to add a Runbook? (Optional)

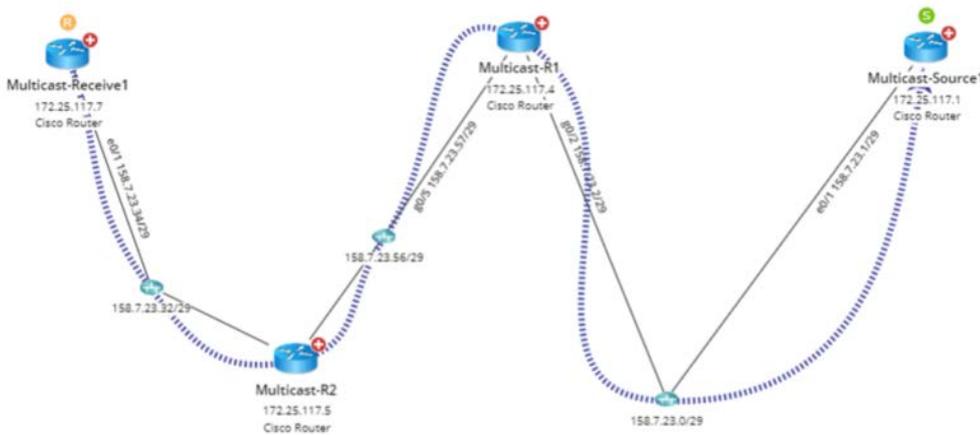
Browse Runbooks

Cancel Save

Variable Input \* Required

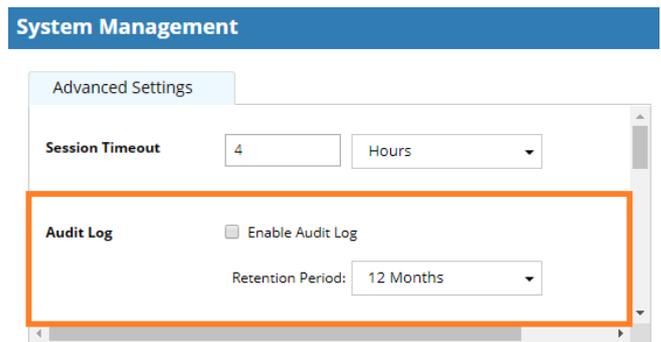
Type	Variable Name	Default Value	
<b>API Service Stub</b>			
	* \$stub_name	stub1	
	\$mode	Real-Time	
	\$max_waiting_hours	1	
	* \$domain_name	BVT_DB2DOM_db634	
<b>Map Creation</b> <span style="float: right;">Settings</span>			
	* \$source	auto	
	* \$source_gateway	auto	
	* \$destination	auto	
	* \$path_type	unicast	
	\$group	auto	
	\$destination_gateway	auto	
	\$protocol	IPv4	
	\$source_port		
	\$destination_port		
	\$data_source	Live Network	
	\$direction	false	
	\$use_config_in_current...	true	
	\$is_active_path	false	

A multicast path is shown as follows:

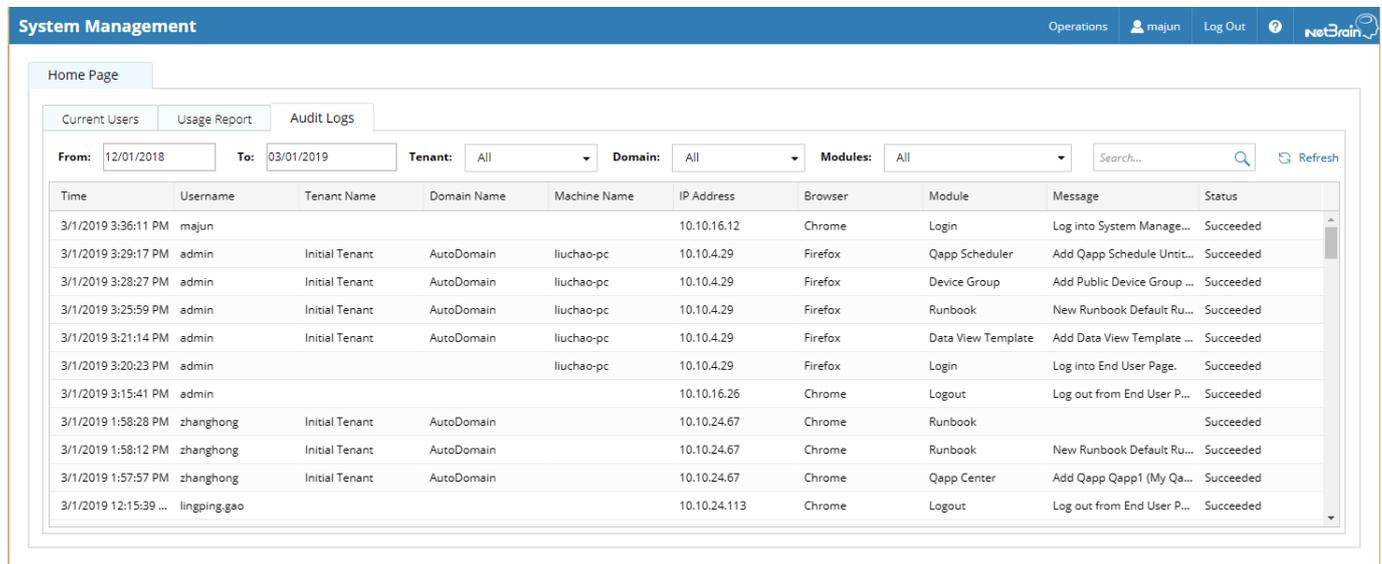


## 2.19.8. Visualize and Search Audit Logs

In the previous version, the system added the capability to record user operations for auditing purposes, but only at the back-end (location on Web Server: C:\ProgramData\Netbrain\AuditLog). By default, the function is disabled. Admin decides whether to enable the audit log function and configure the retention period of these logs.



To visualize the demand for viewing audit logs within the system and get prepared for cloud-hosting solutions, IEv8.0 adds a new tab page to show audit logs within the pre-defined retention period.



- Provide multiple filters to narrow down audit logs by customized period, tenant, domain, or multiple feature modules.
- Add the search capability for audit logs in the last 12 months, by entering keywords in specific columns, including Username, Machine Name, IP Address, and Message.

## 2.19.9. Custom Rules for User Inputs of CLI Commands

In Runbook Automation and benchmark tasks, the system retrieves device data by executing CLI commands. The rules to define executable CLI commands can prevent illegal operations, such as changing device

configuration files. IEV8.0 provides three types of rules and allows command customization by modifying keywords in each rule in the database.

## Supported Rules

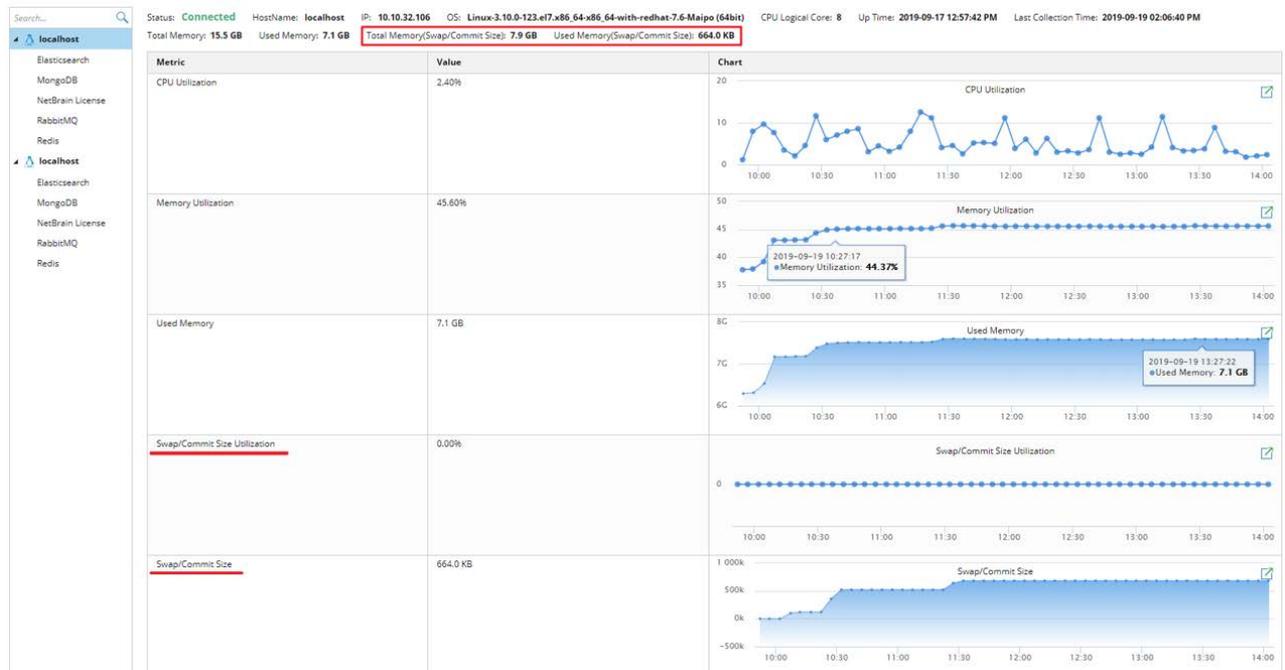
The rule that each CLI command must meet: (A or B) and C.

Rule	Name	Default Value of Keyword	Description
A	Start with	ping, trace, traceroute, telnet, ssh, b, list	The CLI command started with 'ping', 'trace', 'traceroute', 'telnet', 'ssh', 'b', or 'list' is valid.
B	Include	show, sh, get, display, running-config	The CLI command contained 'show', 'sh', 'get', 'display', or 'running-config' is valid.
C	Black List (Exclude)		CLI commands cannot contain the keyword defined in this rule.

## 2.19.10. Enhancements to Service Monitor

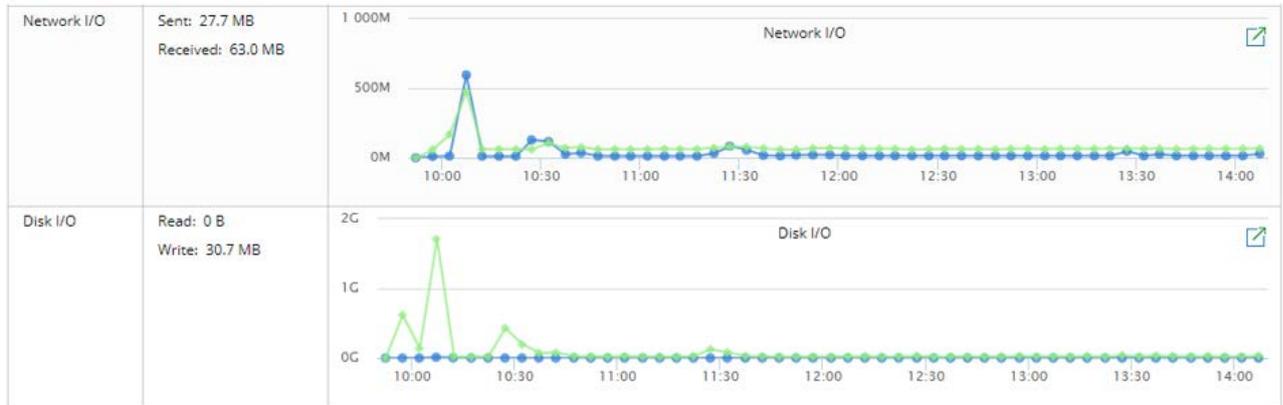
### New Metrics for All Servers

- Add 2 Static Metrics and 2 Tendency Metrics:



- Total Memory(Swap/Commit Size)
- Used Memory(Swap/Commit Size)

- Swap/Commit Size Utilization
- Swap/Commit Size
- Change the chart type for “Network I/O” and “Disk I/O” from Total Tendency to Incremental Tendency.



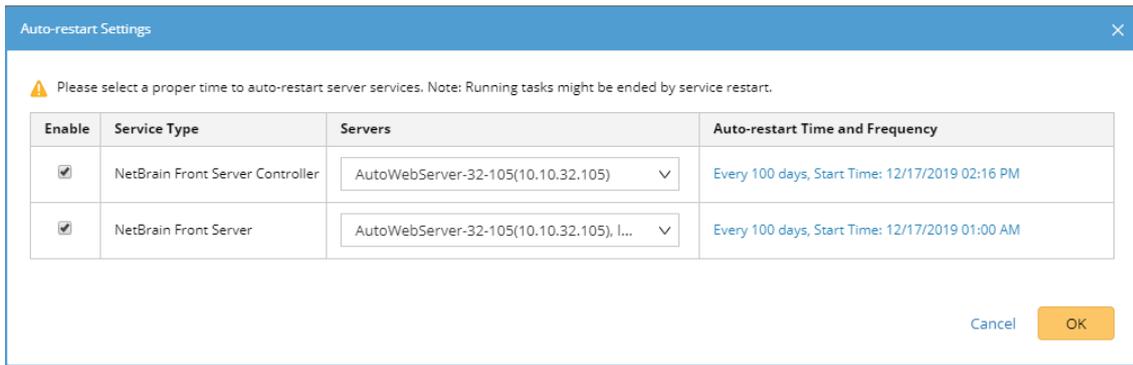
## New Metrics for RabbitMQ

- Number of Ready Messages for RMClientCallback Queue
- Number of Ready Messages for flowengine health monitor Queue
- Number of Ready Messages for flowengine leader Queue
- Number of Ready Messages for flowengine task status Queue
- Number of Ready Messages for taskengine event Queue
- Number of Ready Messages for xfagent task Queue
- Number of Ready Messages for prepared task Queue
- Number of Ready Messages for FSC trigger Queue

## Schedule Service Auto-Restart for Front Server & Controller

- The longer time the system has been running for, the more memory usage it will consume, and the more junk data it will generate. IEv8.0 adds the capability to schedule an auto-restart of the service for Front Server and Front Server Controller recurringly, to solve some complicated issues with can be fixed by a restart button.

- By default, the auto-restart settings are disabled. Users can enable an auto-restart on specified servers and select a proper time point and frequency.

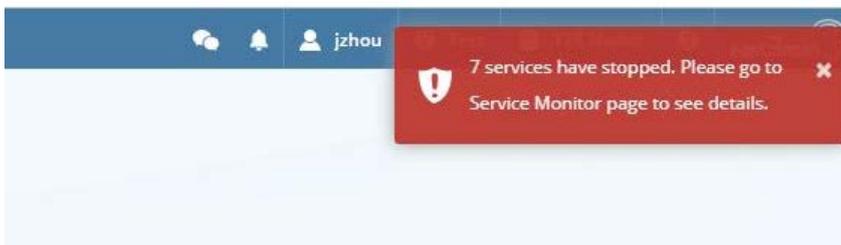


**Note:** Running tasks might be ended by a service restart.

## Alert for Stopped Services

Service anomaly may cause functionality issues in the Thin Client. It usually takes too much time to debug an issue about functions, but finally, it turns out to be caused by stopped services.

IEv8.0 can alert users when any system service has stopped. Users can timely take actions if required, such as restarting these services.



## Log Analysis for System Health

In previous versions, the system added the capability to collect Support Logs for troubleshooting purposes. To proactively check system health, IEv8.0 Service Monitor adds the capability to analyze a large number of logs for various NetBrain services and even Windows Event Logs, and extract key words out of them to summarize the addressed issue types and causes. As a benefit, both the Support Team and admin users are able to look into the detailed logs to drill down and resolve the issues before end users might encounter them.

Server: All Type: All From: Click to select a date/time... To: Click to select a date/time... Log Analyze

Tasks: 12/31/2019, 3:20:51 PM Download 12/14/2019, 2:40:00 PM 1/4/2020, 2:40:00

Server	Service	Known	Unknown
DB31	RabbitMQ	1	21
DB31	Redis	0	4
WIN-RT25HL0D3UV	NetBrain Worker Server	7	6978
WIN-RT25HL0D3UV	NetBrain Front Server Controller	0	3
WIN-RT25HL0D3UV	Event	9	0
WIN-RT25HL0D3UV	NetBrain Task Engine	1	0
WIN-V86T8P82MUJ	IE Web	4	68
WIN-V86T8P82MUJ	NetBrain Front Server	1	2
WIN-V86T8P82MUJ	NetBrain Front Server Controller	0	143
WIN-V86T8P82MUJ	Event	28	0
WIN-V86T8P82MUJ	Knowledge Cloud Proxy	0	3
WIN-V86T8P82MUJ	NetBrain Task Engine	1	0

Issues

Category	Description
start	OnStart
start	fully started
task	SubmitTaskFlow timeout
task	SubmitTaskFlow error
BuildL3MultiDeviceMediaIpMask	redisHelper.LockMedia failed
BuildL3MultiDeviceMediaIpMask	redisHelper.AddMedia2Edges failed
task	send result to XAgent

Issues

Category	Description
UnknownError	failed to load DLA task
UnknownError	parser table not exist.

## Other Enhancements

- The timestamp for both the start time and the end time of the collected support log is automatically translated to the local time zone of users based on UTC (universal time code).

NetBrain Service Monitor\_Support Log

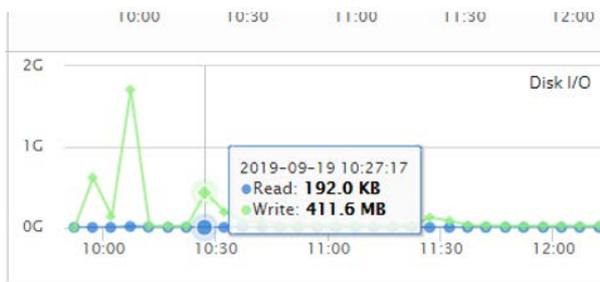
MongoDB Disk Alert Rules admin

Server: All Type: All From: 2019-09-01 14:21 To: 2019-09-19 14:21 Collecting Cancel View Last Result

Selected Logs Size: 0 B Download

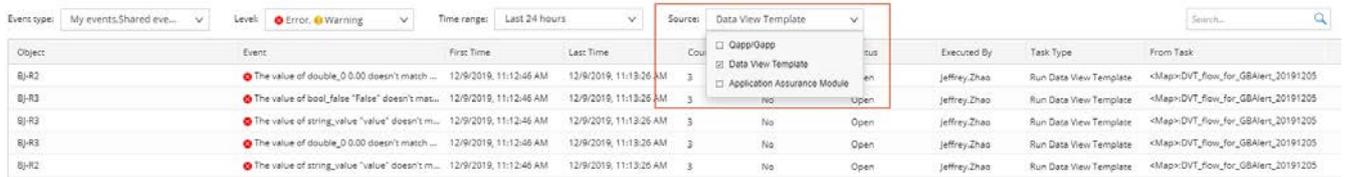
Server	Type	Log	Size	Status	Created Time	Last Updated Time
localhost	Elasticsearch	/var/log/netbrain/elasticsearch/elasticsearch_deprecation.log	4.2 MB	✓	2019-09-16 07:49:27 AM	2019-09-20 02:19:47 PM
localhost	Elasticsearch	/var/log/netbrain/elasticsearch/elasticsearch_index_indexing_slowlog.log	0 B	✓	--	2019-09-16 07:49:24 AM
localhost	Elasticsearch	/var/log/netbrain/elasticsearch/elasticsearch_index_search_slowlog.log	0 B	✓	--	2019-09-16 07:49:24 AM
localhost	Elasticsearch	/var/log/netbrain/elasticsearch/elasticsearch-2019-09-16-1.log.gz	29.1 KB	✓	2019-09-16 00:00:00 AM	2019-09-17 09:14:46 AM

- When a chart contains multiple legends, hovering on a data point will display the values of all legends.

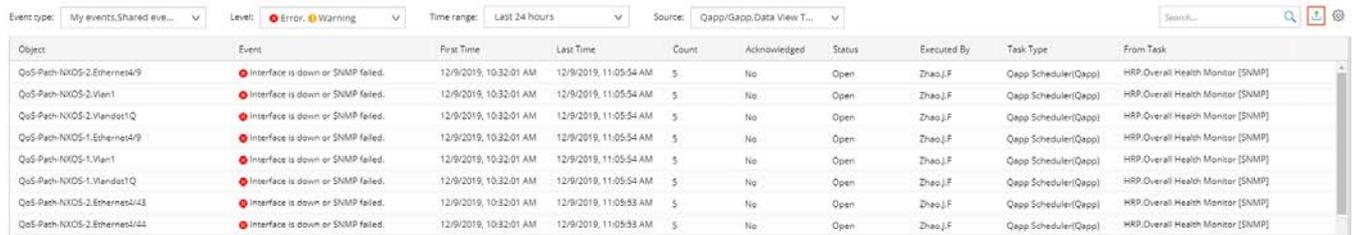


## 2.19.11. Enhancements to Event Console

- Use three filters to narrow events by task source.

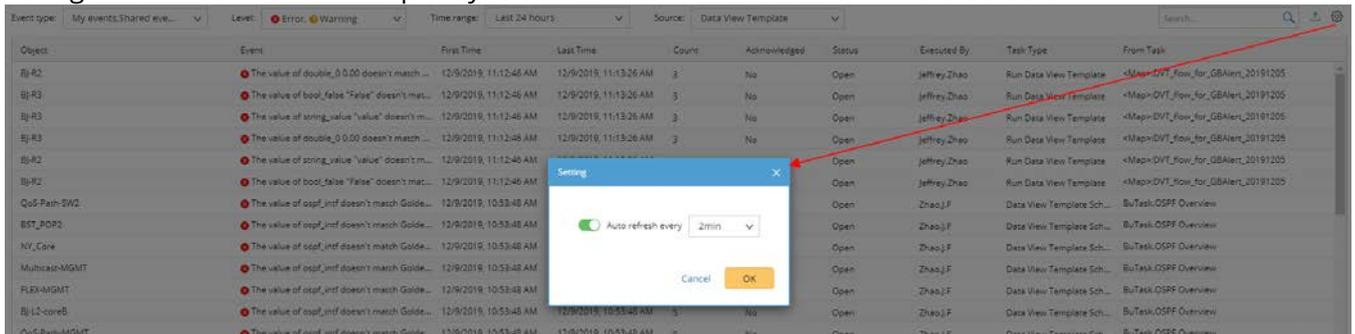


- Export events into a CSV file by clicking the Export button.

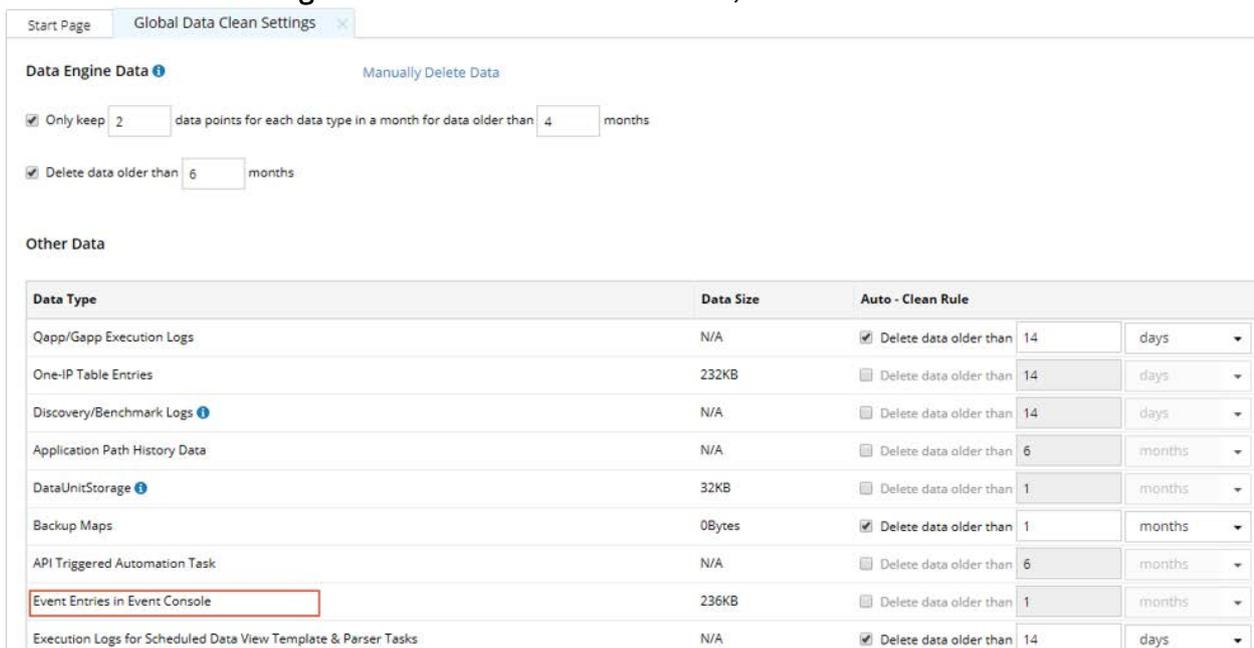


- Events are highlighted and differentiated through Error/Warning icons, rather than font colors.

- Configure the Auto Refresh Frequency to auto-refresh event entries in Event Console.



- Auto-clean event entries regularly by enabling and setting the corresponding rule (Domain Management > Global Data Clean Settings > Event Entries in Event Console).



## 3. Platform Enhancements

IEv8.0 enhances the platform frameworks by building a Platform Plug-in Framework to enable platform expansion to allow the customization for data accuracy, creating Knowledge Cloud to manage all software resources, improving Path framework to discover the dependent underlay and overlay paths besides many other enhancements.

### 3.1. Enhanced Platform Framework

#### 3.1.1. Topology Dependency

In an overlay and underlay network, the overlay network is created on top of the underlay network. To demonstrate the topology relationship between the overlay network and the underlay network, IEv8.0 introduces the **Topology Dependency** concept.

The system records topology dependency of different network-layer types (such as L3 and L2, VXLAN overlay topology and L3 or L2) at the interface level. For example, the interface `vne1` which belongs to the VXLAN topology depends on the ethernet interface `e1` which belongs to IPv4 topology.

In the Data Model, IEv8.0 records the dependency of an interface between the overlay and the underlay (including the dependency between the topologies in the system) in the GDR. The GDR values are filled via Driver and the format is as follows:

GDR ID	Type	Description	Sample
<code>intfs.topoDep.topoType</code>	string	The topology types of the current interface.	<pre>" intf_name": " F 0/0",   "topologies": [     {       "topology_type": "Ipv4 L3 Topology",       "dependency": {         "topology_type": "L2 Topology",         "interface": "F0/0"</pre>

In IEv8.0, the topology dependency is mainly used in path calculation. During a path calculation, the system automatically calculates a path and underlay paths at a hop based on interface topology dependency information. For more details, see [Calculate Underlay and Overlay Path](#).

---

### 3.1.2.Driver Enhancements for Virtualization

In previous versions, NetBrain supports retrieving data from virtual devices and building topology and maps for them. However, limitations with the support still exist, such as:

- Retrieving the data table for child devices is not support.
- Can not calculate L2 topology for virtual device.

IEv8.0 improves network visualization support to address the limitations including:

- [Optimize Discovery and Benchmark for Virtual Child Devices](#)
- [Support Live Access of Virtual Devices](#)

---

#### Optimize Discovery and Benchmark for Virtual Child Devices

IEv8.0 optimizes the discovery and data retrieving for virtual child devices in drivers. After adding virtual devices to a domain via Discovery, the system can retrieve various data tables of virtual child devices via Benchmark to maintain virtual child device data.

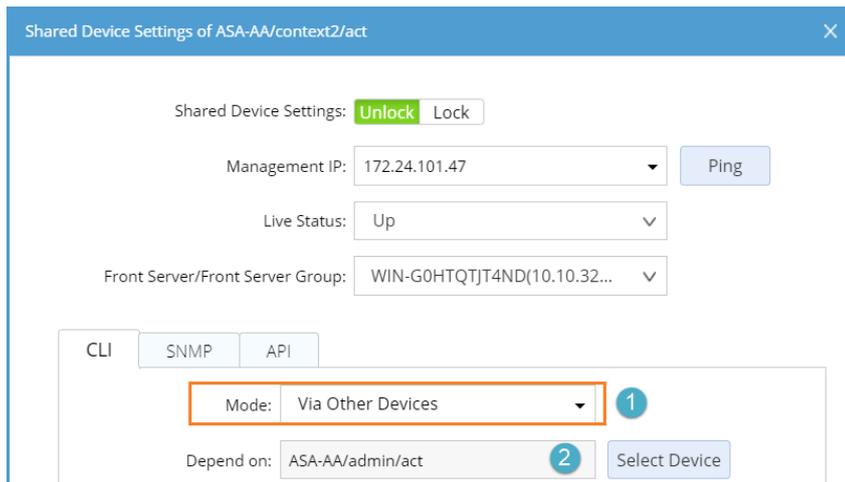
---

#### Support Live Access of Virtual Devices

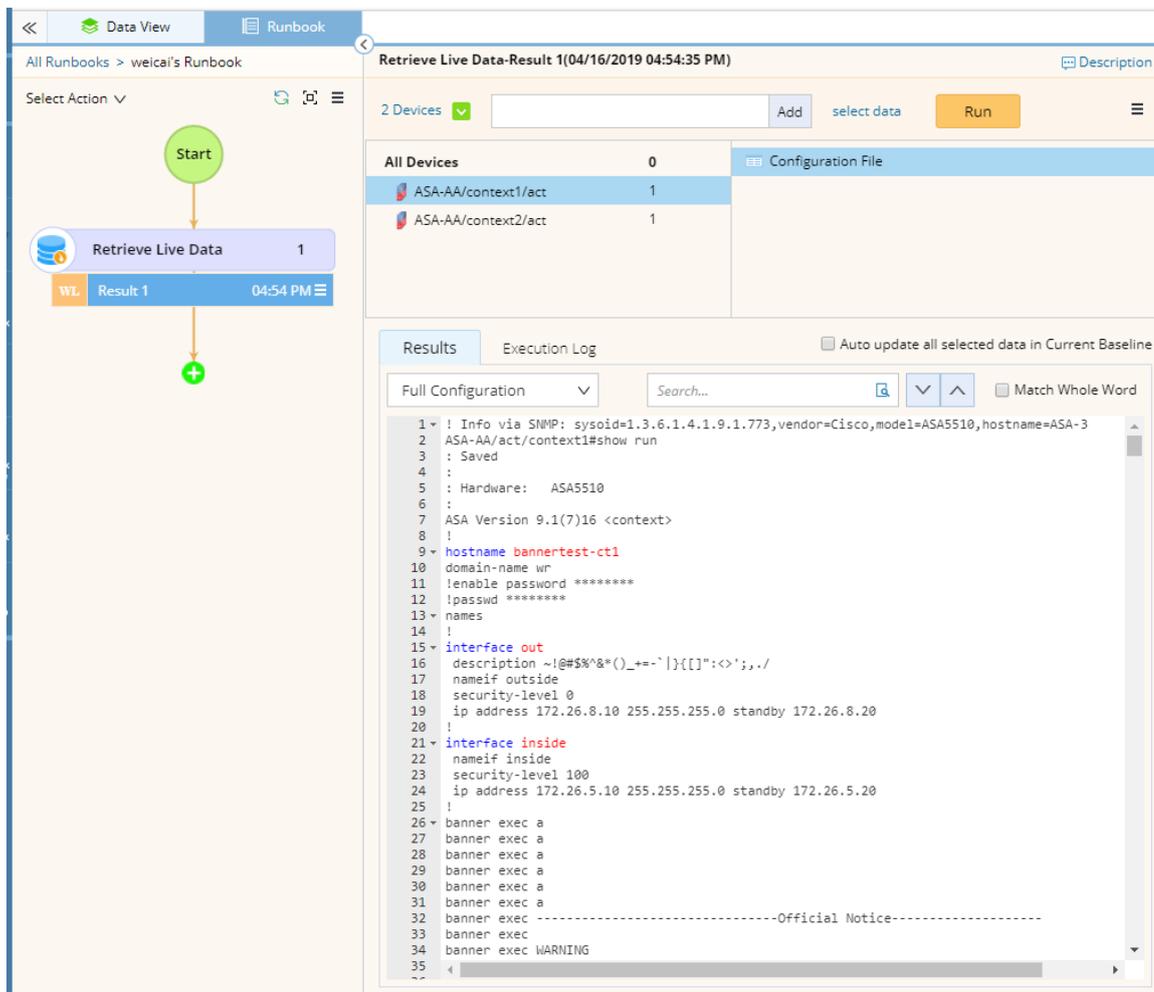
In the previous versions, the system is not able to retrieve the data for a virtual device on demand because the virtual device does not have its own device setting. IEv8.0 offers device settings for virtual devices and allows the on-demand live access to virtual devices via their parent devices.

IEv8.0 adds the **Mode** field for both CLI and SNMP access in the shared device setting and for a virtual device the Mode is set to Via Other Devices and Depend On field is set to be its parent device, which is automatically

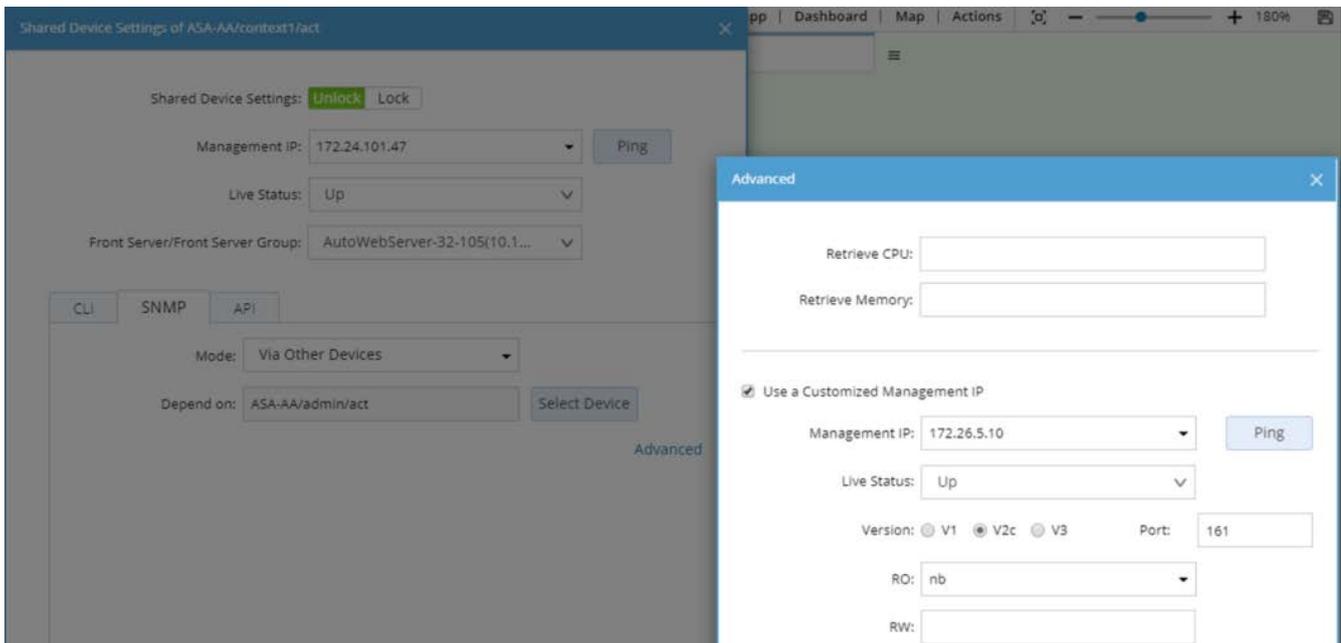
set in the discovery process and can be manually modified.



When a user retrieves the data from a virtual device, the system will log in to the parent device defined in the Shared Device Settings, switch to the virtual device (such as an ASA context) and issue the CLI command to retrieve the data.



Further customization of SNMP information can be set in the Shared Device Settings of a virtual child device in case virtual child devices have independent SNMP attributes from their parent devices, such as ASA contexts.



### 3.1.3. New Multi-CLI Mode Support

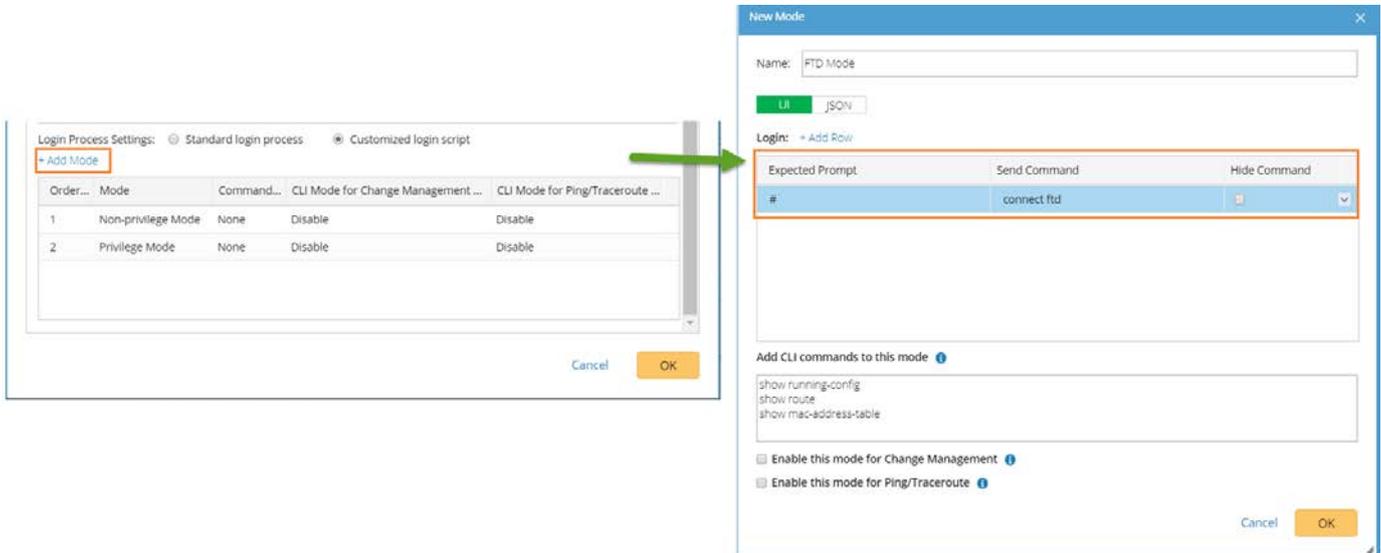
In previous versions, the system allowed users to customize login prompts and commands based on the privilege and non-privilege modes. However, this customization is not applicable when a device type has other CLI modes. For example, a Cisco Firepower includes three modes: FXOS mode, FTD mode, and ASA mode, so the customization based on the privilege and non-privilege modes does not apply to this scenario.

To support more CLI modes and enable users to define commands dedicated to a mode, IEv8.0 optimizes the multi-CLI mode function.

## Define More CLI Modes in Driver

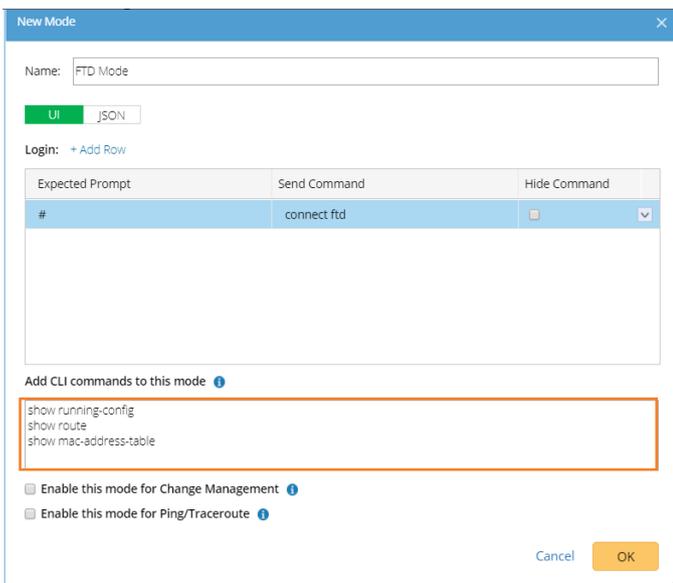
Power users can customize to add multiple CLI modes for a device type in the device driver. The customization includes expected prompts and commands for each mode, and the dedicated CLI commands. During the live access to a device, the system attempts CLI modes from top to bottom until addressing the right one.

**Example1:** Define a mode and the corresponding prompts



**Note:** After defining CLI modes in a driver, users can modify the login scripts in the advanced settings of the Shared Device Settings, but the modified scripts will not be synchronized to the driver.

If users specify CLI commands dedicated to a mode, the system will send the commands in the specified mode.



**Note:** If multiple modes are associated with the same command, the system does not try the command in the subsequent modes when it is executed successfully in a mode.

---

### 3.1.4. Build/Retrieve Configuration File for SDN Nodes via TechSpec

In previous versions, the system did not retrieve configuration files for some SD-WAN devices after discovering them via APIs (TechSpec), which caused the problem that Policy and ACL check based on configuration files cannot be implemented in the system.

IEv8.0 improves the configuration retrieval for SDN nodes in the following aspects:

- Enable the configuration file retrieval in TechSpec.
- Add the logic to retrieve configuration files via **API > CLI > SNMP** when retrieving them in the benchmark/live path/parser/device configuration pane.

---

### 3.1.5. NCT Framework Enhancements

In IEv8.0 the definition of Network Control Table (NCT) can be written with Python scripts instead of Executable Procedures which can be only edited in the early EE version and not supported in IE and the management of System NCT is moved to device drivers instead of the OG (Object Group). Retrieving NCT can be triggered in a benchmark task or in the process of retrieving live data. This enhancement eases writing and upgrading of NCT as well as NCT extension for new technology support in path calculation.

Two new tabs "Json" and "NCT" are added in the device driver to define and manage NCT.

- The Json tab is used to enter global Json scripts and define what types of data tables can be supported.

Device Driver Properties

Driver name: Cisco IOS XR

Device type: Cisco IOS Switch

Author: Cisco IOS XR

Live Access | Config File | **Table** | Advanced

**Json** | Route Table | MAC Table | ARP Table | NDP Table | BGP Advertised Route Table | NCT Table

Use json to define data tables. Apply

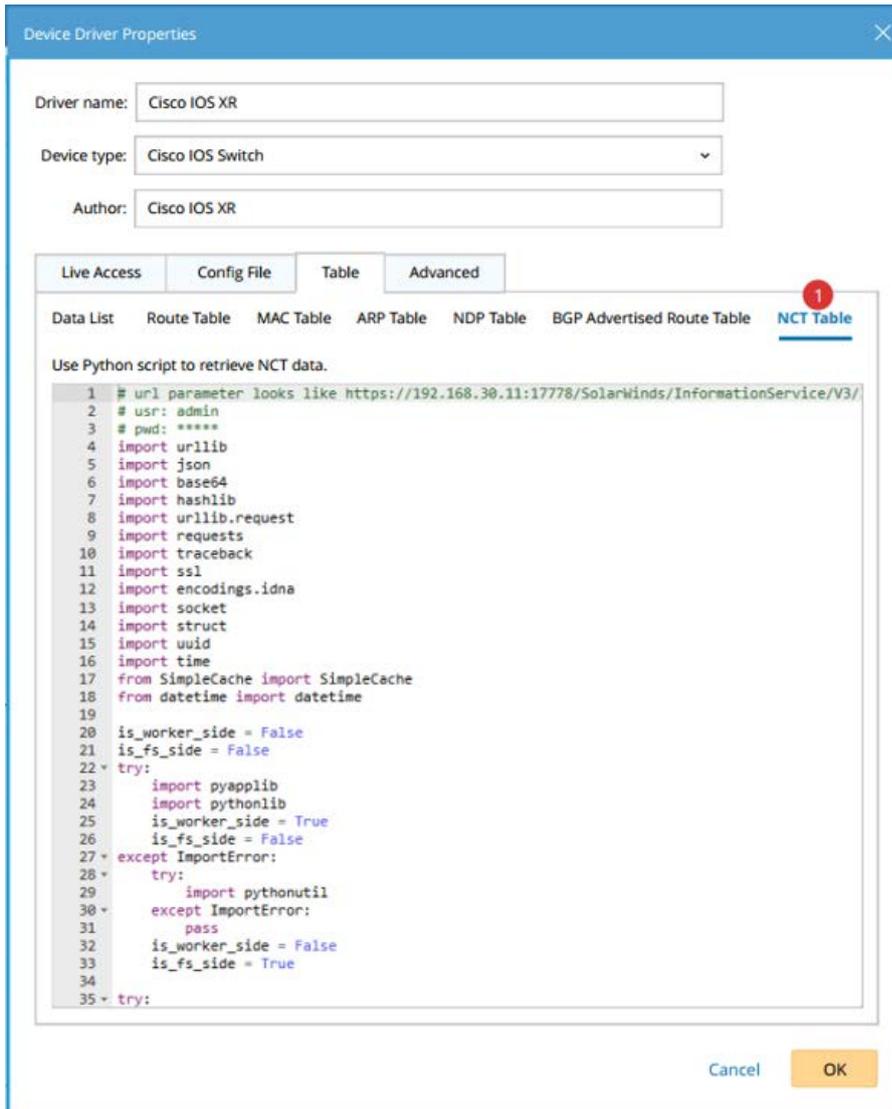
```

1  {
2  "scriptContent": "",
3  "moduleName": "aci_discovery",
4  "generalApiResponseMethod": "aci_generic_api_resp",
5  "discoverMethod": "aci_discover_devices",
6  "actions": [
7  {
8  "schema": "ACI.topRoot.children.fabricTopology.children.fabricPod.children.fabric
9  "nodeName": "Fabric Node",
10 "dataTables": [
11 {
12 "name": "Route Table",
13 "type": "routeTable",
14 "displayedTableType": "Route",
15 "isSystem": true,
16 "isNct": false,
17 "qualifyBy": "GDR.xxxx",      ##option
18 "CLIComands": "show IP route"
19 "CLIComands_VRF": "show IP route &VRF"
20 "subNamesMethod": [
21
22 {
23 "type": "API",
24 "Method": "aci_discovery.get_Route_sub_names_API"},
25 {
26 "type": "CLI",
27 "Method": "aci_discovery.get_Route_sub_names_CLI"},
28 {
29 "type": "SNMP",
30 "Method": "aci_discovery.get_Route_sub_names_SNMP"
31 }
32 ]
33 "retrieveMethod": [
34 {
35 "type": "API",
36 "Method": "aci_discovery.get_Route_table_API"},
37 {

```

Cancel OK

- The NCT Table tab is used to enter Python scripts for retrieving and parsing NCT data.



### 3.1.6.L2 Topology Framework Enhancements

- Improve the method of determining the interface category. IEv8.0 uses the config file parser defined in the Driver to judge whether an interface is sub interface, VLAN interface or port-channel interface, instead of the previous hardcoded interface name in the L2 topology calculation module.
- Use a sub interface's main interface instead of the sub interface itself to establish the L2 topology link with neighbors.
- Attempt to establish an L2 topology link for port-channel interfaces. In the previous versions, the system only generated topology links for the port-channel interfaces configured with Cisco VPC.

- No longer provide a separate calculation logic for Cisco vPC and instead provide a generic MC-LAG (Multi-Chassis Link Aggregation Group) Model to support the MC-LAG technology. This model not only supports Cisco vPC but also supports similar technologies from other vendors such as Arista MLAG.
- Use the topology Qapp/Plugin to calculate topology for transparent firewalls and remove the previous hardcoded calculation logic in the L2 topology calculation module. The topology Qapp adds the support for Riverbed WAN Optimizer.
- Use the SSID as the interface, instead of the previous hardcoded Dot11ratio interface, to establish the L2 topology link for wireless LWAP/WAP.
- Provide a general calculation model "BPE (Bridge Port Extension)", to support Cisco FEX dual-homed technology. This model provides the ability to support more similar technologies from other vendors via customization.
- Remove the logic that L2 topology links calculated via NDP tables do not include interfaces configuring IP addresses.
- Provide a new topology type "L2 Overlay Topology" to support underlay/overlay technologies, including:
  - VXLAN
  - OTV
  - VPLS

Refer to [Technology Support Summary](#) for more details.

---

### 3.1.7. Use Plugins to Customize for Data Accuracy

**Platform plugin** is used to fix different types of data accuracy and incompleteness issues, which can be common to all customers or just special to one customer. The plugin can be inserted in different stages of discovery, benchmark or any type of scheduled task. For example, a plugin can be inserted after the general discovery process to import configurations of devices that may not be accessed. Platform Plugin may be implemented in Python which calls the system APIs to modify the underline network and data model and can be automatically updated via Knowledge Cloud. The user can enable or disable a plugin and define the execution points and order.

#### Use Case

---

- **Solve the data incompleteness issue due to the inaccessible devices**

By plugins, users can import the data such as configurations, route table, MAC table and ARP table from other systems.

- **Complement data model logics**

By plugins, users can fix the limitations or defects of the method to create the data model, such as fix up missing L2 links between two neighbor interfaces due to missing interface index.

- **Support new functions**

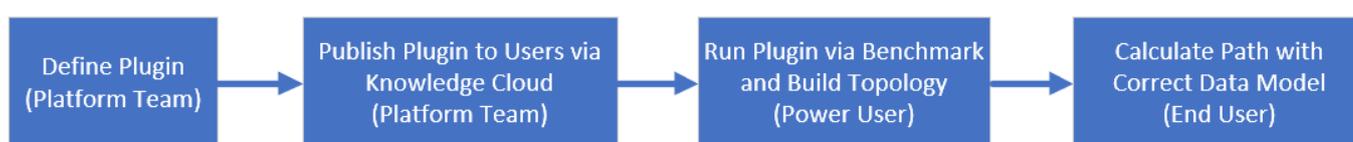
By using plugins, NetBrain can deploy new technology support on the system without having to release new patches, such as supporting new device types or vendors in IeV8.0.

- **Maintain the domain data automatically**

By plugins, users can manage and operate in the domain automatically, such as executing plugins to solve duplicate IP issues via moving them to different zones, create sites and device groups, and manage the device host changes.

## Reference Flow

A typical flow chart of using plugins is as follows:



## Define Plugins

The plugin is defined and managed in the Plugin Manager. A Plugin is an executable program that can be implemented by Python and has its input parameters. The system provides a set of APIs for the plugin to query and modify the underline network and data model.

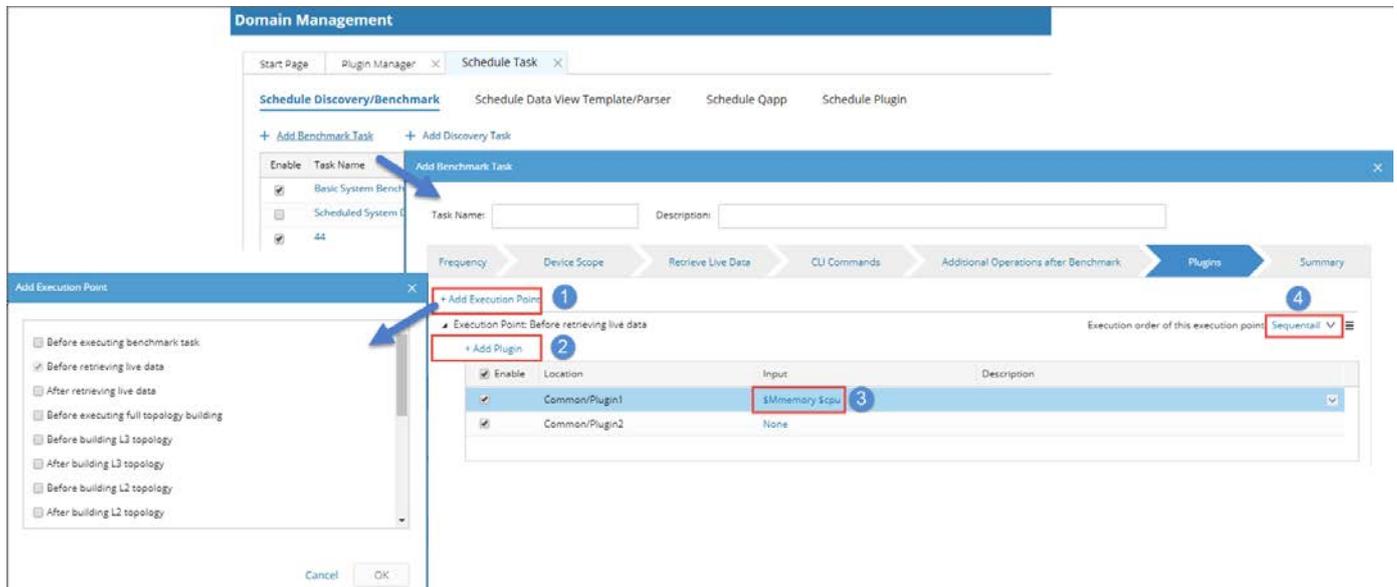
Name:   Recommendation Settings

Description Input main.py +

```
1 from netbrain.sysapi import datamodel
2 from netbrain.sysapi import devicedata
3 from netbrain.sysapi import pluginfw
4 import time
5 # from netbrain.sysapi.DataModelAPI import *
6
7 #mgmtIP vendor
8 def run(pluginParam):
9     pluginfw.AddLog("begin run plugin,param:%s" % pluginParam)
10    a= datamodel.GetDeviceProperty(pluginParam,'BJ*POP')
11    try:
12        pluginfw.AddLog("GetDeviceProperty :%s"%a)
13
14
15    except Exception as error:
16        pluginfw.AddLog('test API GetDeviceProperty with error: %s' % error, pluginfw.ERROR)
```

## Execute Plugins

In a scheduled task such as Benchmark and Discovery, you can add the execution points (the time to execute plugins) under the Plugin tag and select plugins to be executed in these points. You can also define the execution order for plugins at the same execution point.



## Plugin Framework

The following introduces the concepts related to the plugin framework.

### Plugin Input

The plugin input is flexible and can be text in any format (txt, json, etc.); All inputs will be passed as text by the framework to plugins during execution.

### Plugin Qualification

The system will perform a qualification check (filter) based on device groups at the framework level; A plugin will be executed only when the target device group is not empty.

### Installation Setting of Plugin

The Installation Setting is used to control whether a plugin can be automatically installed when being updated through the Knowledge Cloud. This definition of installation is set by the Platform team.

A plugin can be specified to execute at which execution point and whether enabled by default by using the Installation Settings. The settings only work for the built-in tasks, including Basic System Benchmark.

### Plugin Category

The plugins can fall into the following categories:

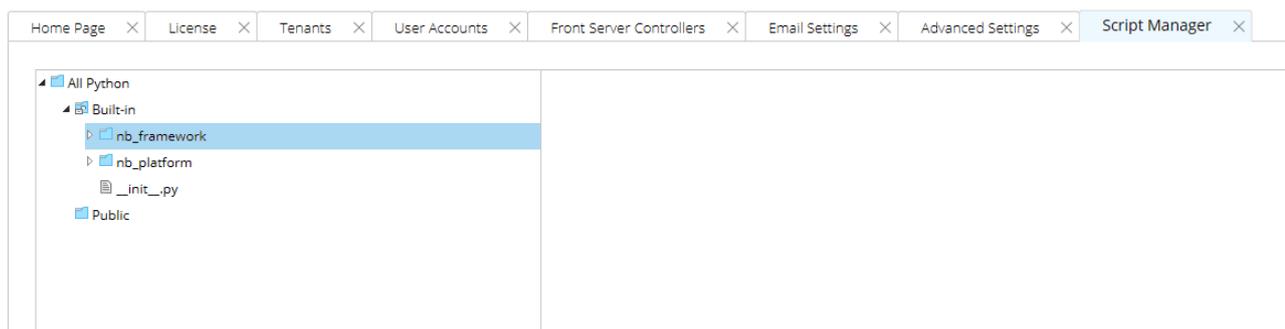
No.	Category	Description
1	Discovery Plugin	The plugins used to intervene with discovery. The corresponding execution points of the discovery plugin includes as follows: <ul style="list-style-type: none"> <li>▪ Before executing a discovery task</li> <li>▪ Before executing a discovery task</li> <li>▪ After executing a discovery task</li> </ul>
2	Tune Plugin	The plugins used to intervene with the device tune. The tuning of a plugin can be executed on-demand or in Schedule Discovery and Schedule Plugin.
3	Benchmark Plugin	The plugins used to intervene with the benchmark. The corresponding execution points of the benchmark plugin includes as follows: <ul style="list-style-type: none"> <li>▪ Before executing a benchmark task</li> <li>▪ After executing a benchmark task</li> </ul>
4	NCT/System Table Plugin	The plugins for NCT and System tables modification and creation. These plugins are executed after retrieving live data and before building topology.
5	Topology Building Plugin	The plugins used for topology building.
6	Data Model Plugin	The plugins used for data model building, such as MPLS Cloud, Internet Cloud, Device/interface property and device relationship like virtualization.

See also: [Built-in Plugins](#).

### 3.1.8. Script Manager

**Script Manager** is a new function in IEv8.0, which provides a front-end UI to enable users to manipulate python scripts, including the addition, deletion and update of python scripts in the system. Any modifications to the python scripts in the Script Manager will be automatically synchronized to system worker servers.

#### System Management



The Script Manager provides two types of scripts:

- **Built-in** — This folder contains all built-in python data of NetBrain IE in different features and modules. All the python files in this category can be imported for use in Qapp/Path/Plugin/Driver.
- **Public** — This folder is used to contain the private python data that you will create for your NetBrain System. Like the built-in scripts, the scripts in this folder can be called in the NetBrain system.

## 3.2. Extension Based on Enhanced Platform Framework

This section will use each technology as examples to introduce the extension based on the platform framework enhanced in IEv8.0, and several [Built-in Plugins](#) to customize for data accuracy.

### 3.2.1. Technology Support Summary

In IEv8.0, the support of the following technologies is newly added and enhanced for specific vendors. Click on each technology name to view more details.

Technology Name	Data Model Support	Topology Support	Path Support	Supported Device Type
<a href="#">VXLAN</a>	√	L2 Overlay Topology	√	<ul style="list-style-type: none"> <li>▪ Cisco Nexus Switch</li> </ul>
<a href="#">OTV</a>	√	L2 Overlay Topology	√	<ul style="list-style-type: none"> <li>▪ Cisco Nexus Switch</li> <li>▪ Cisco IOS Switch</li> </ul>
<a href="#">VPLS</a>	√	L2 Overlay Topology	√	<ul style="list-style-type: none"> <li>▪ Juniper Router</li> </ul>
<a href="#">HA/Cluster</a>	√	L3 Topology	√	<ul style="list-style-type: none"> <li>▪ Cisco ASA Firewall</li> <li>▪ Cisco Router</li> <li>▪ Juniper SRX</li> <li>▪ Palo Alto Firewall</li> </ul>
<a href="#">Transparent Device</a>	√	L2 Topology	√	<ul style="list-style-type: none"> <li>▪ Riverbed WAN Optimizer</li> <li>▪ Cisco ASA Firewall</li> <li>▪ Palo Alto Firewall (Virtual Wire)</li> </ul>
<a href="#">MC-LAG</a>	√	L2 Topology	√	<ul style="list-style-type: none"> <li>▪ Cisco Nexus Switch</li> <li>▪ Arista Switch</li> </ul>

Technology Name	Data Model Support	Topology Support	Path Support	Supported Device Type
				<ul style="list-style-type: none"> <li>▪ Dell Force10 Switch</li> <li>▪ Checkpoint Firewall</li> <li>▪ Cisco ASA Firewall</li> </ul>
<a href="#">FEX Dual-Homed</a>	√	L2 Topology	-	<ul style="list-style-type: none"> <li>▪ Cisco Nexus Switch</li> </ul>
<a href="#">Port-Channel</a>	√	L2 Topology	√	-
<a href="#">End System</a>	√	L3 Topology	√	<ul style="list-style-type: none"> <li>▪ Ubuntu Server</li> <li>▪ Windows Server</li> <li>▪ Mac Server</li> <li>▪ Printer</li> </ul>
<a href="#">Wireless</a>	√	L2 Topology	√	<ul style="list-style-type: none"> <li>▪ Cisco LWAP</li> <li>▪ Cisco WLC</li> </ul>
<a href="#">Management Route</a>	√	-	√	<ul style="list-style-type: none"> <li>▪ Dell Force 10</li> <li>▪ F5 Load Balancer</li> </ul>
<a href="#">SPB</a> (Shortest Path Bridging)	√	-	√	<ul style="list-style-type: none"> <li>▪ Avaya Switch</li> <li>▪ Avaya VSP</li> </ul>
<a href="#">HSRP</a>	√	L3 Topology	√	<ul style="list-style-type: none"> <li>▪ Cisco Nexus Switch</li> </ul>

## 3.3. Path Enhancements

### 3.3.1. Enhancements to Path Gateway

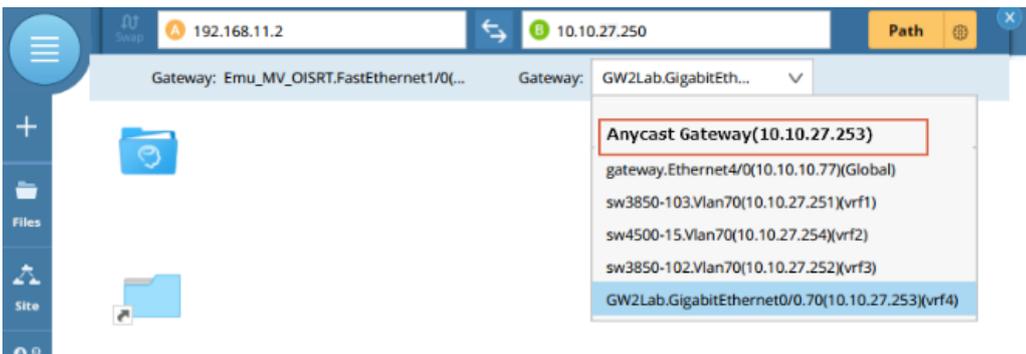
The enhancements to Path Gateway Framework in IEv8.0 include:

- The path gateway list is completely controlled by the scripts rather than the framework, which is more flexible to handle possible gateway issues in the future. For example, with scripts, it is easier to support the new format or HSRP configuration of HSRP gateway.
- The system is capable of processing different technologies to display the gateway in the specific formats in the list. After users make choices, the system can determine which script can be used to parse the correct gateway based on the choices. For example, a VXLAN Anycast Gateway (10.1.27.253). Users can find the correct gateway based on the display format at ease.

## Customizable Gateway Based on Technology

In some network configurations (such as HSRP and VXLAN anycast gateway), the network traffic uses a certain network device as the gateway. Since the NetBrain system does not know the cable connection, it can only list all the possible gateways that match the input IP for users to choose from.

To help users find the correct gateway in case of VXLAN anycast gateway or HSRP, IeV8.0 enhances the algorithm to select the gateway: when a user inputs a source or destination IP, the technology behind a gateway will be listed, such as Anycast Gateway (10.1.27.253). After the technology is selected, the system will automatically find the appropriate gateway interface.



## VXLAN Topology and Path Logic in Anycast-Gateway Deployment

- **Anycast Gateway Selection.** When an IP is entered, the system checks whether the VRF of the IP interface belongs to the L3 VNI Interface. If yes, the input IP will be treated as an endpoint of anycast gateway and displayed as a default option in the gateway list [the display format is **anycast gateway (IP)**]. If anycast gateway is selected, the system will look up the ARP tables of all alternate gateway lists and find the matching item with the input source/destination as the gateway.
- **Enable Duplicate IP for Anycast-Gateway Configuration.** When VXLAN is configured with anycast-gateway, it will be recognized as duplicate IP in the system by default, which affects the L3 topology and path calculation. To solve this problem, IeV8.0 adds a GDR tag for the corresponding interface to mark the interface IP as a non-duplicate IP. While calculating L3 topology, the system identifies this tag and treats the interface IP with this tag as a non-duplicate IP and connects all interfaces with the same IP to the same L3 LAN segment.

## Fix-Up Path Source/Destination IP and Gateway

Gateway is required for the NetBrain system to calculate an A-B path. In the current path framework, when users input a source or destination, the system will automatically look up the gateway interfaces within the

same subnet as the input source or destination. However, the gateway lookup method based on the device interface and the same subnet does not work in some cases.

IPv8.0 introduces the **Path IP and Gateway Fix-up** feature, which enables users to manually define gateways based on destination and source (IPs). When users calculate a path, the system will use the predefined gateway if the input source and destination IPs match rules in the Path IP and Gateway Fix-up Manager.

### Key Use Case

- A gateway device has no gateway interface. For example, a device's gateway is a VIP on an F5 load balancer.
- A gateway device is not in the same subnet with the input destination or source.
- A gateway device is not in the domain.
- Specify a gateway for all IPs in a network segment.

For example, define a path gateway fix-up rule so that all IP address in a specific subnet will use the specified gateway for path calculation: In the **Source** field, enter the subnet (for example, 10.10.16.0/24), **\*** in the **Destination** field (\* means any) and 10.10.19.253 in the **Source Gateway** field. The sample fix-up rule means that when users input an IP within the subnet 10.10.16.0/24 in the Source field when calculating a path, and no matter what they input in the Destination field, the path will use 10.10.16.253 as the source gateway.

Dialog box titled "Edit IP and Gateway Fix-up entry" with a close button (X).

Section: Specify the original Source and Destination

- \*Source: 10.10.16.0/24
- \*Destination: \*

Section: Specify the Replacements and Gateways (You must fill at least one item below)

- Source Replacement: *IP, Subnet or leave it blank*
- Source Gateway: 10.10.19.253
- Destination Replacement: *IP, Subnet or leave it blank*
- Destination Gateway: *IP, Interface Name or leave it blank*

Description:

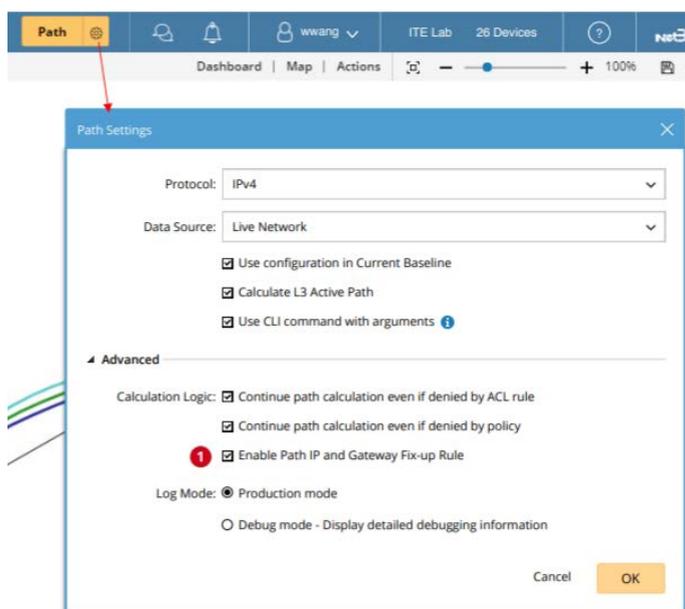
Buttons: Cancel, OK

When users start a path calculation, the system will check if the destination and source match one fix-up rule defined in the gateway fix-up manager. If matched, the system will use the fix-up gateway or replacement destination/source (if defined in the fix-up rule) to calculate the path.

For example, calculate a path from 10.10.16.15 to 172.11.1.12. The source IP is in the range of the subnet 10.10.16.0/24, this path calculation matches the rule defined in the above sample gateway fix-up rule and the system will use the specified gateway 10.10.19.253 to calculate the path.



The gateway fix-up rule is enabled by default. Users can disable it before calculating a path in the Path Settings.



## Path Gateway Fix-Up Logic

The Path IP and Gateway Fix-up function at the domain level enables power users to define specific rules to affect the gateway (or source and destination) selection when end users calculate paths.

When an end user calculates a path, if a rule is matched, the system will trigger a python script to replace the gateway (or source and destination) based on the source and destination and the path will be calculated based on the information after replacement;

After calculating the path, the system returns the path result and the replacement information to the front-end UI.

## Set Default Path Gateway

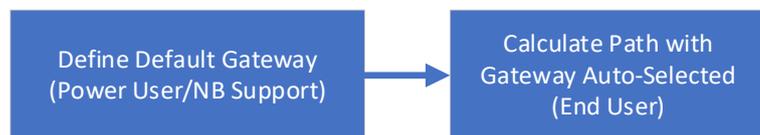
---

To some customers, the gateway device IPs or names of end systems and servers in their networks are regular. For example, always use the largest/minimum address in a network segment or always use the device with a hostname containing GW as gateways. While NetBrain system automatically uses the first gateway in the gateway list as the target gateway by default. To solve the accuracy issue of the gateway selection, IEv8.0 offers a domain-level setting to allow power users to specify the rules of the default gateway.

**Note:** The system does not provide a UI for this setting and users need to specify the rule in the database.

### Key Use Case and Flow

Accurately set gateways for end systems and servers based on path gateway rules in customers' networks.



### Implementation Logic

After a user inputs the source/destination, the gateway list is generated based on the gateway list calculation logic, and then the system determines which gateway item should be put on the first top and selected as the default gateway based on the defined default gateway rule.

**Note:** If the input IP matches an end system whose path gateway attribute is predefined in Device Details Pane, the logic of the default gateway rule will not be triggered, and the system always uses the path gateway defined in end system properties as a top priority.

---

## 3.3.2. Calculate Overlay and Underlay Path based on Topology Dependency

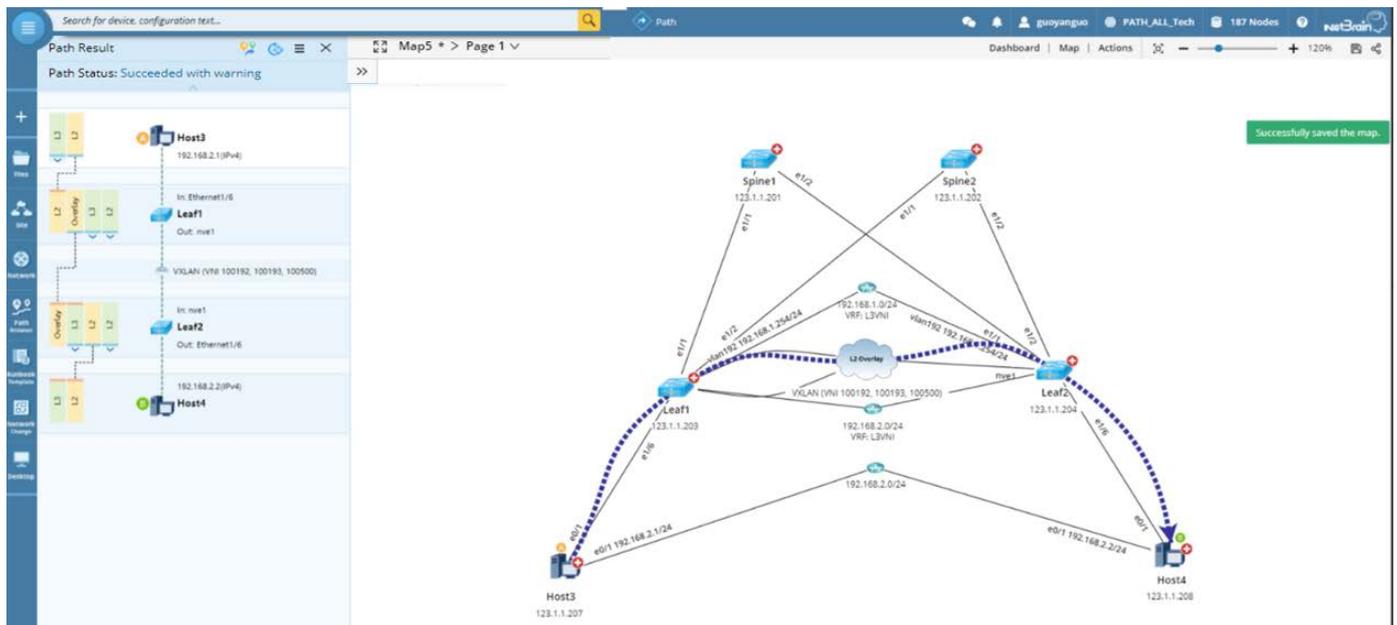
In hybrid data centers with both underlay and overlay deployments, users care about not only the connection of the underlay network but also that of the overlay network built on top of the underlay network.

IEv8.0 can calculate both underlay and overlay paths of networks with overlay deployment, including VXLAN, VPLS, and OTV.

## Calculate Path based on Topology Dependency

The system automatically calculates a path and underlay paths at a hop based on interface topology dependency information. For example, the interface `vne1` which belongs to the VXLAN topology depends on the ethernet interface `e1` which belongs to IPv4 topology. While discovering the path, users do not need to select the path type (L3 or L2). Instead, the system will automatically discover the corresponding path and underlay paths. When the source and destination IPs are in the same subnet, the system discovers the L2 path first; otherwise, the system discovers the L3 path first. At each hop, if the outbound interface's topology depends on another underlay topology type, the system automatically discovers the underlay path.

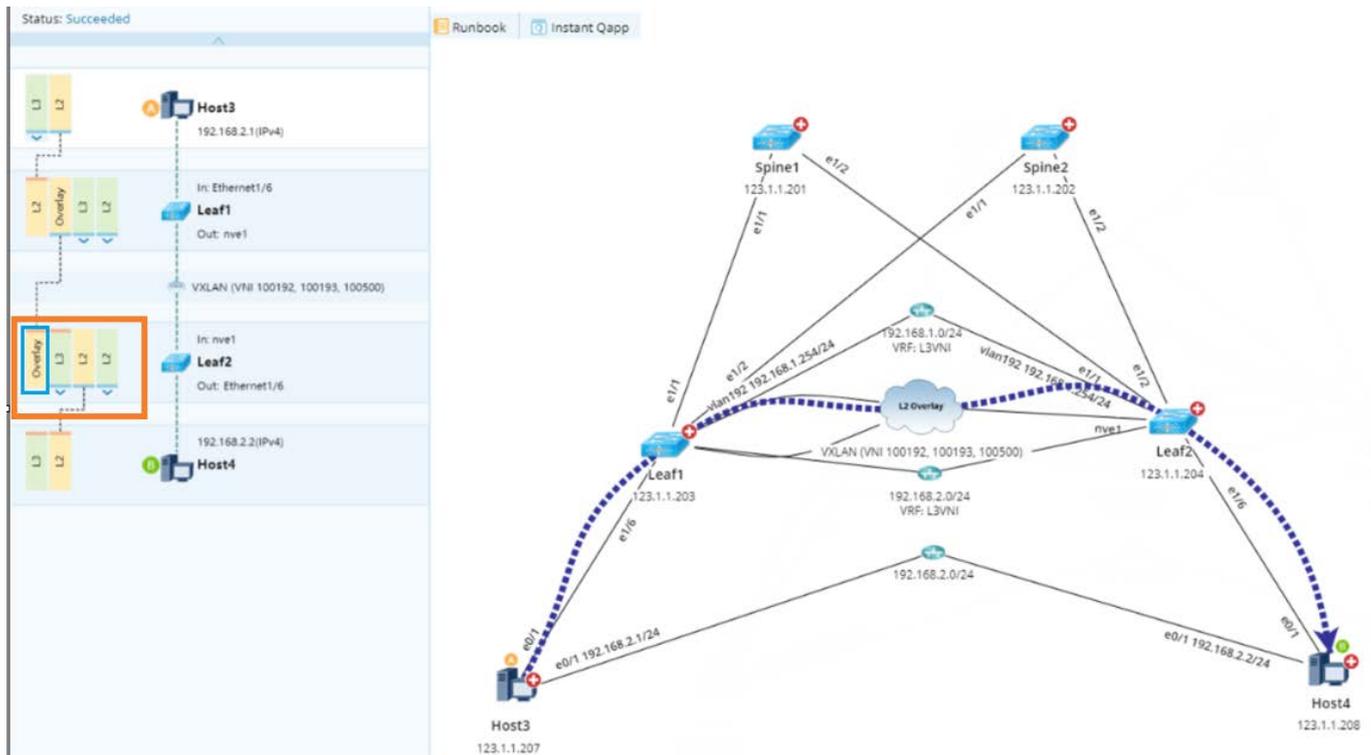
### Example: A/B Path Crosses VXLAN Overlay Network



## Visualize Topology Dependency in Path Result Panel

The Path Result Pane shows hop nodes along an A-B path, including inbound and outbound interfaces at each hop. At each hop, the Path Result Pane displays all path types. Users can select a path type to display it on a

map.



### 3.3.3. QoS Path Support

IPv8.0 system can calculate L3 QoS paths. The system retrieves QoS parameters such as DSCP, IP Precedence and QoS-Group based on the server-policy information configured on an interface. It automatically calculates an L3 QoS path based on the QoS policies.

The screenshot shows the NetBrain interface for a QoS path calculation. The path is from 158.7.19.5 to 158.7.19.62. The path steps are:

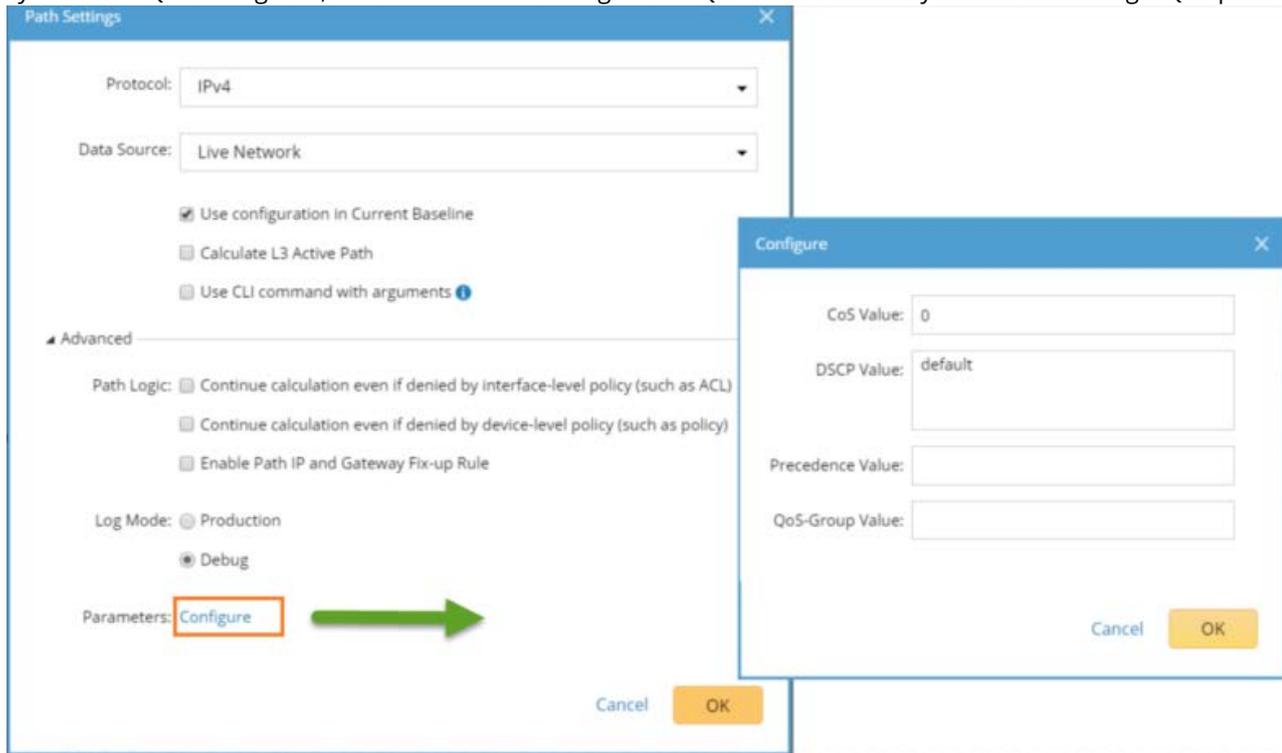
- 158.7.19.5 (158.7.19.5/29 HSRP: 102(158.7.19.3))
- In:Ethernet0/1.102
- QoS-Path-R3 (Out:Ethernet0/2.250)
- 158.7.19.8/29
- In:Ethernet0/3.250
- QoS-Path-R1 (Out:Ethernet0/2)
- 158.7.19.40/29
- In:Ethernet0/1
- QoS-Path-R2 (Out:Ethernet0/3.1)
- 158.7.19.56/29 HSRP: 11(158.7.19.59))
- 158.7.19.62 (158.7.19.62/29)

The device logs for QoS-Path-R3 show the following steps:

1. Check Ingress QoS (Matched)
  - Retrieving QoS via configuration
  - Ingress QoS <DPS-QPPB-POLICY> was found at the input interface Ethernet0/1.102
  - qos-group was set to 10
  - dscp was set to cs7
11. Look up Routing Table (Matched)
  - Retrieving global routing table
  - Prefix: 158.7.19.56, Mask: 255.255.255.248, Output interface: Ethernet0/2.250, Next hop IP: 158.7.19.11
18. Find Next Hop (Found)
  - Next hop is QoS-Path-R1
19. Check Egress QoS (Matched)
  - Retrieving QoS via configuration
  - Egress QoS <output> was found at the output interface Ethernet0/2.250
  - dscp cs7 was switched to af12
  - cos was set to 0
  - qos-group 10 was switched to 1023

The network diagram highlights the path from 158.7.19.5 to 158.7.19.62 through QoS-Path-R1 and QoS-Path-R2.

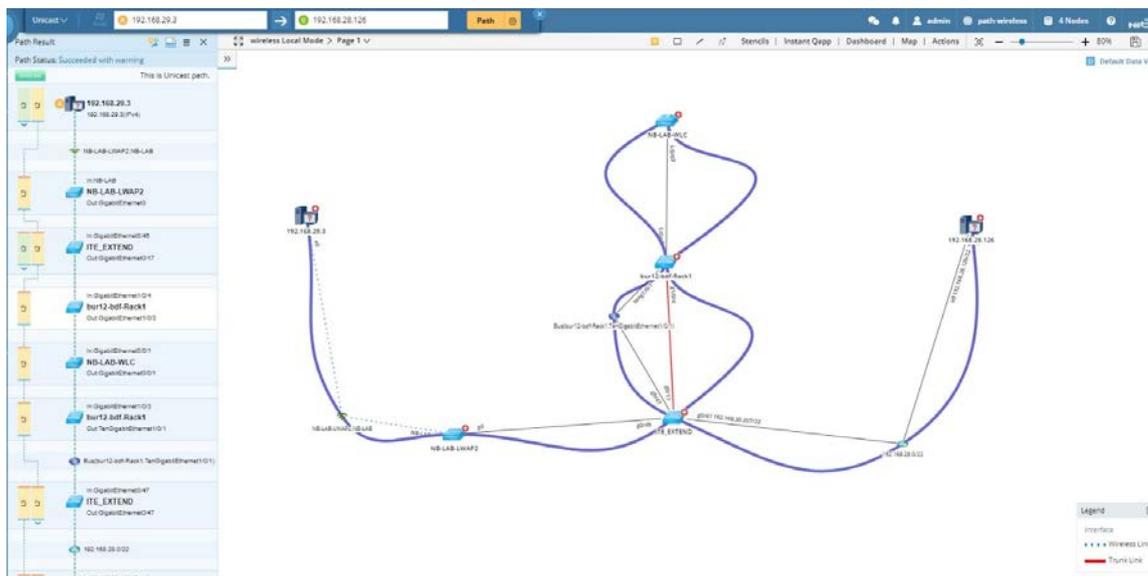
**Note:** The system cannot recognize and retrieve the QoS information of an end system. When the source device is an end system with QoS configured, end users need to configure the QoS values manually before calculating a QoS path.



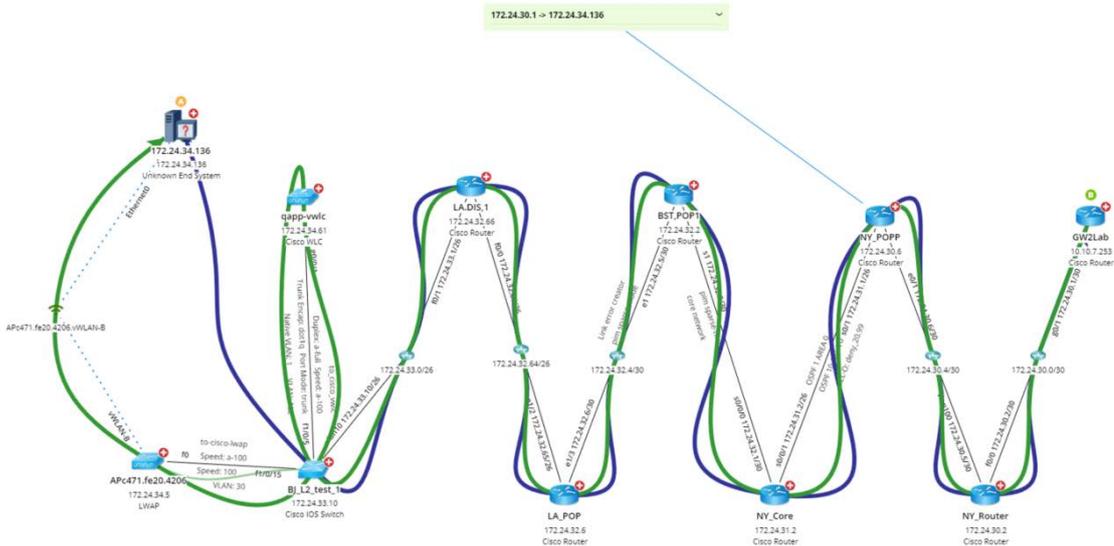
### 3.3.4. Cisco Wireless L2 Path Support

Iev8.0 system improves the support of the L2 path across wireless devices and supports an L2 path calculation across Cisco LWAPs in the Local and Flexconnect mode.

#### L2 Path across LWAPs in Local Mode:



## L2 Path across LWAPs in Flexconnect Mode:



## Path Calculation Logic for Additional Path Parameters:

The traffic state adds DSCP, COS, QOS-Group, and Precedence options to record the corresponding information of traffic packets. When the system checks that an interface is configured with policies during a path calculation, it changes the corresponding value in the traffic state to determine routing decisions if the value in the packet is changed.

Take the path calculation in service policy configured network for example:

1. The system checks whether the in/out interfaces are configured with service policy.
2. If configured, the system checks whether the DSCP/COS/TOS value is modified in the service policy.
3. If modified, the system checks whether the class-map content is matched with the simulated content, including IP, port, DSCP, and simulated packet.
4. If matched, the system modifies the DSCP value in the traffic state and passes the value to the next-hop device for further processing.

**Note:** The change of the Path Settings affects all modules that call the Path feature, such as Application Manager and Runbook.

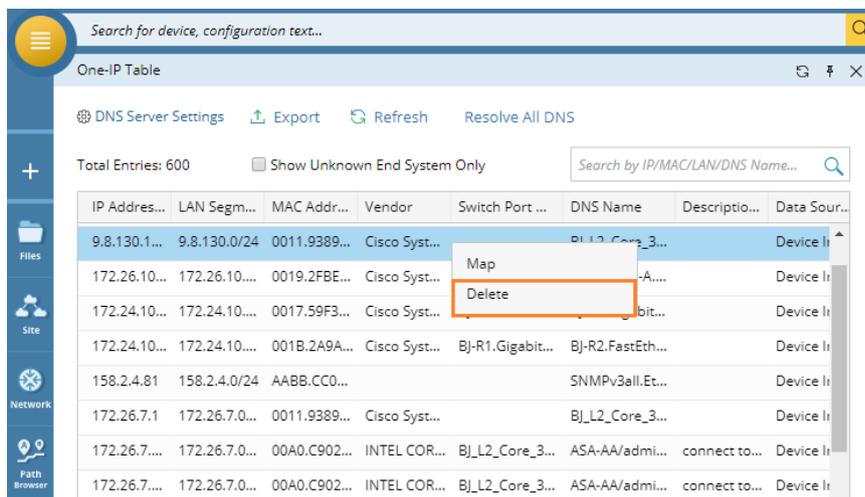
### 3.3.5. Enhancements to Path Calculation for Unknown End Systems

In previous versions, a path related to unknown end systems failed when

- Duplicate unknown end systems were found as a next-hop device during a path calculation.
- Many existing virtual IP addresses were recognized as unknown end systems.

To resolve these issues, I Ev8.0 enhances path calculation for unknown end systems:

- Use the latest unknown end system as the next-hop device when the system finds multiple ones during a path calculation.
- Exclude more virtual MAC addresses for Cisco, Juniper, and Fortinet devices, so that the system will not recognize them as unknown end systems.
- Allow users to delete One-IP table entries and unknown end systems together manually.



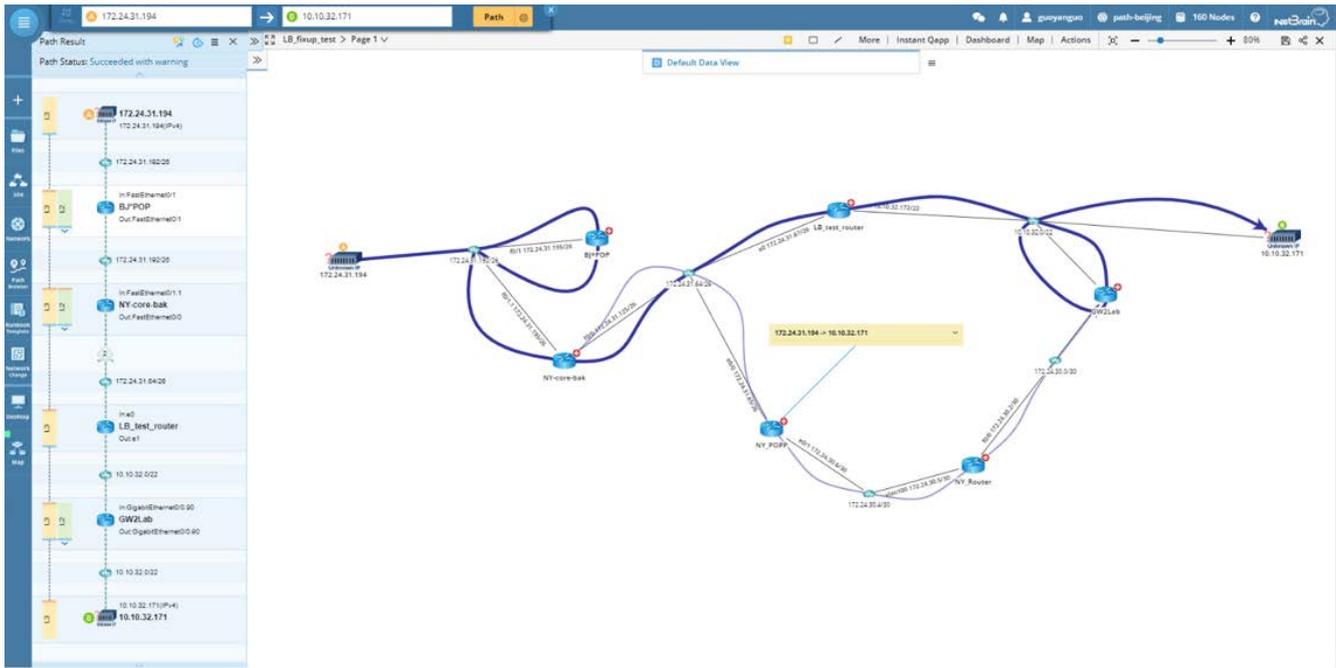
### 3.3.6. Path Visualization

- [Show/Hide Load Balance Branches in Path Detail Pane](#)
- [Visualize All Path Details](#)

#### Show/Hide Load Balance Branches in Path Detail Pane

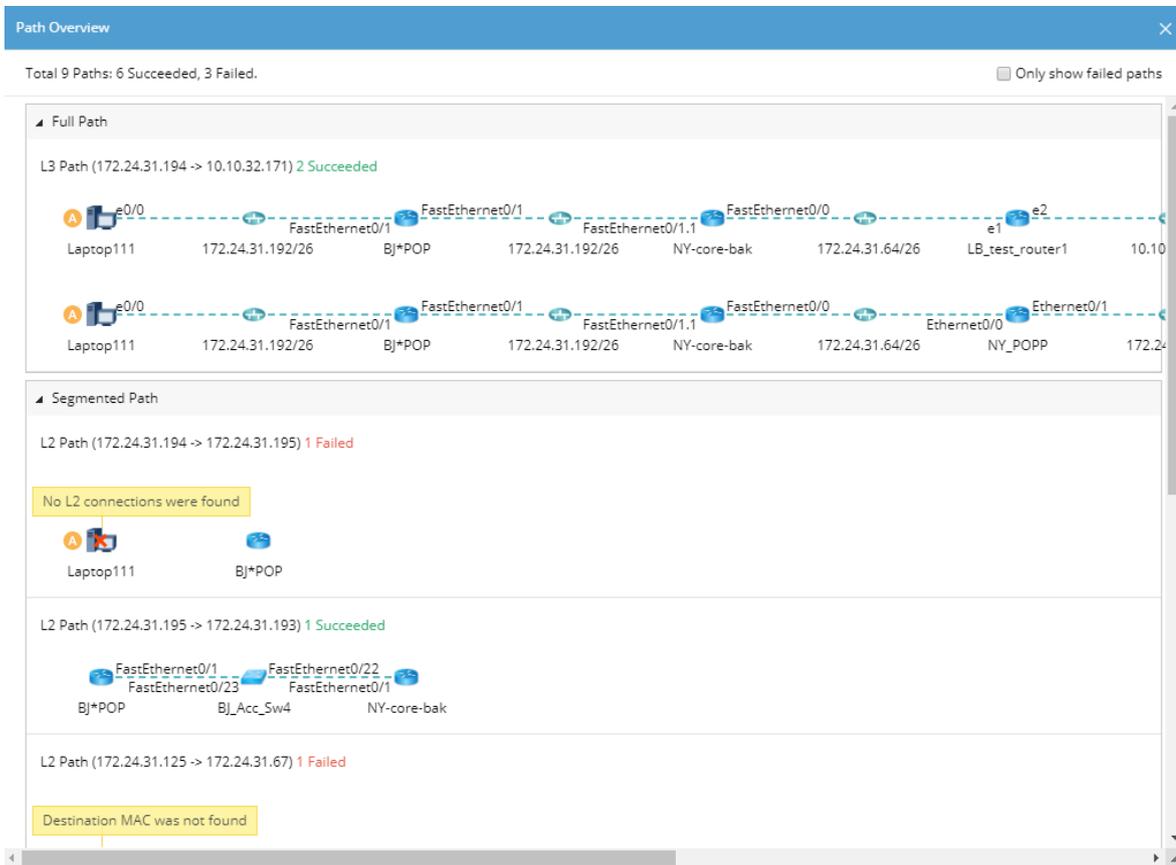
I Ev8.0 provides a **Show all load balances** option in Load Balance Settings to control whether to display all redundant paths on a map. By default, this option is checked, which means to draw all load balance branches

on a map. When it is unchecked, only the active path is drawn in the map.



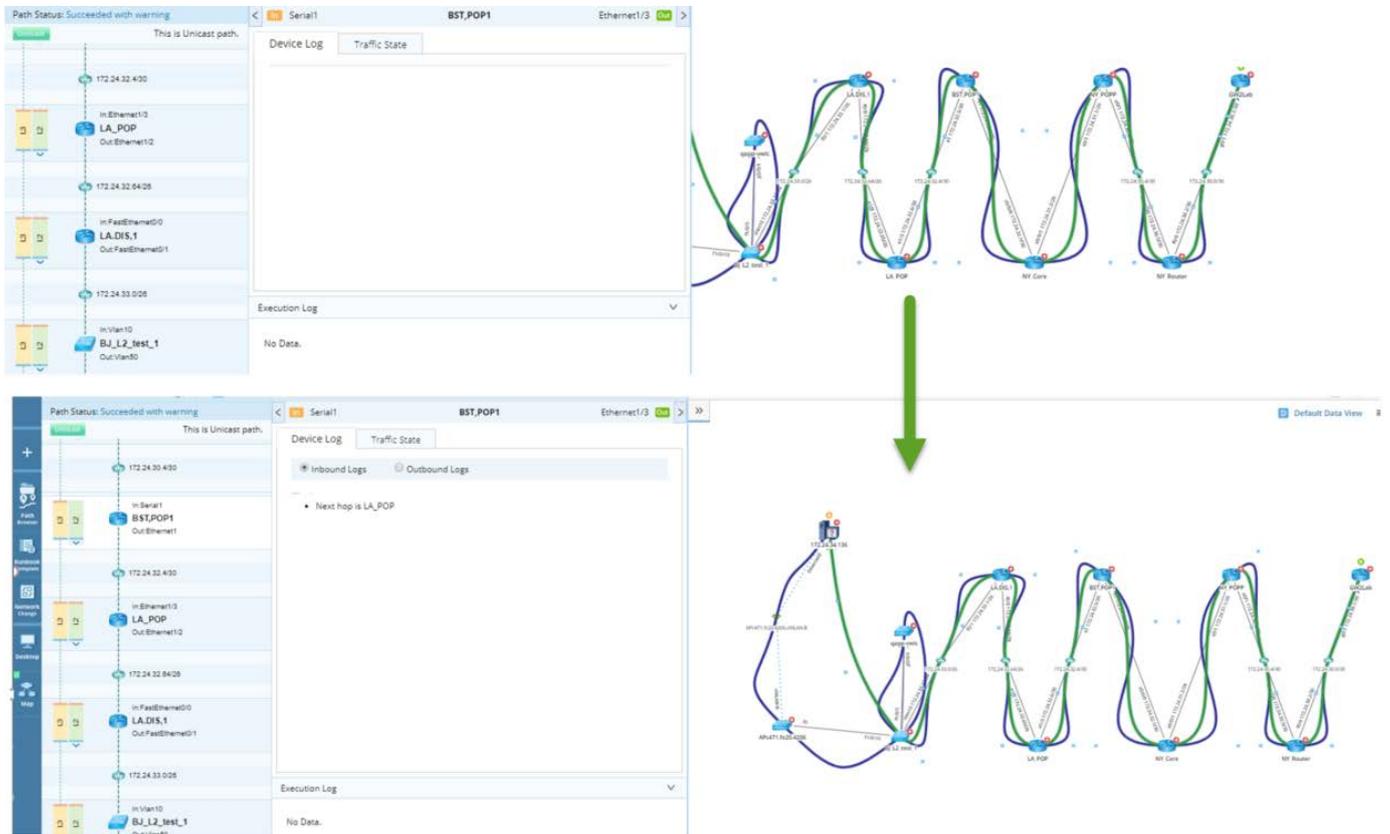
## Visualize All Path Details

With the topology dependency, a path may contain different types of paths, such as L2, L3, and VXLAN. IEx8.0 adds the Path Overview Pane to display the summary of all types of paths. In the pane, users can view the full path and its status (succeeded or failed) and all segmented paths.



### 3.3.7. Pin the Path Result Pane

IEv8.0 adds a Pin function. After users pin the Path Result pane, it will squeeze the map so that the map content can be fully displayed.



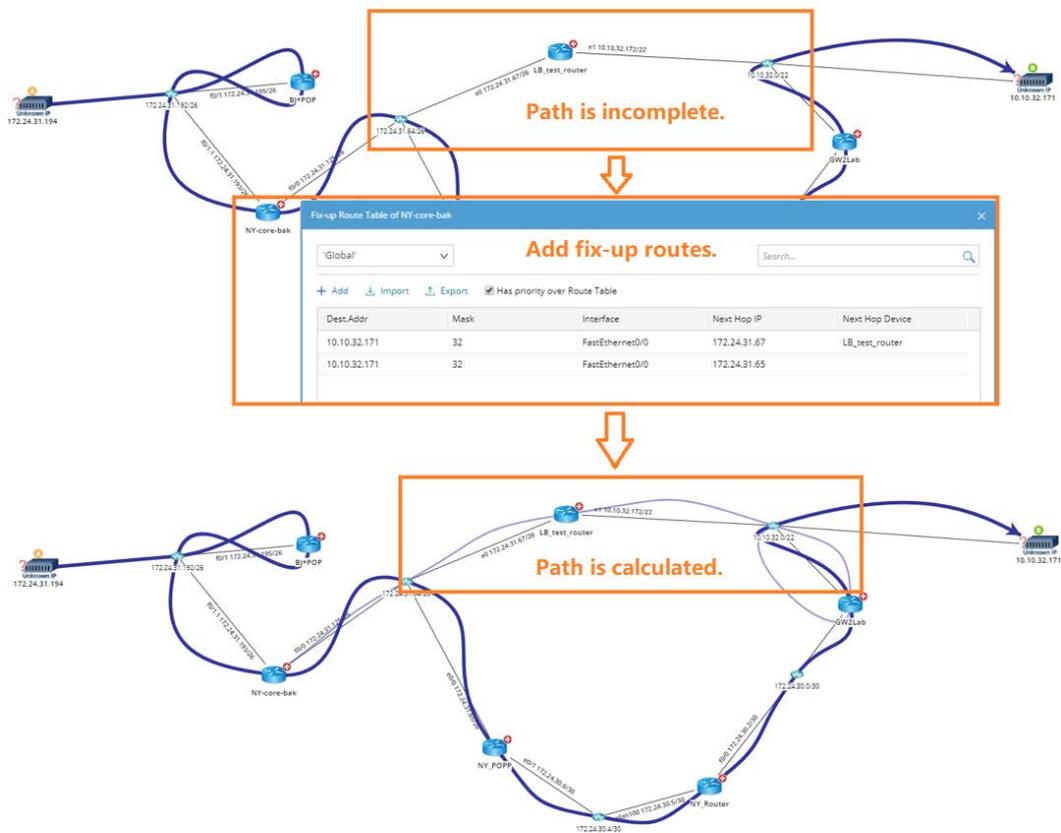
### 3.3.8. Other Path Enhancements

- [Support Fix-Up Route Table](#)
- [Add 2 Options to Control Whether to enable ACL/Policy Checking](#)
- [Support Internet Cloud](#)

#### Fix-Up Route Table

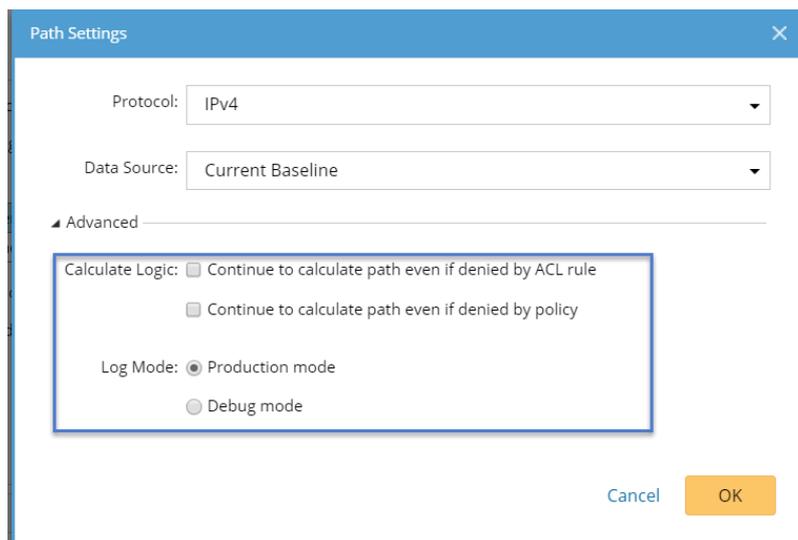
IEv8.0 allows users to manually fix up route tables in case that the system is not able to get the route table from the live network. Further users can define route tables based on different VRFs to support VRF-related cases.

This is an example of completing path calculation by adding a fix-up route table.



## Add 2 Options to Control Whether to enable ACL/Policy Checking

IEv8.0 adds two options in Path Settings to control whether to enable ACL or policy checking before calculating a path. If enabled, the system will ignore ACL or policies check at a path calculation.

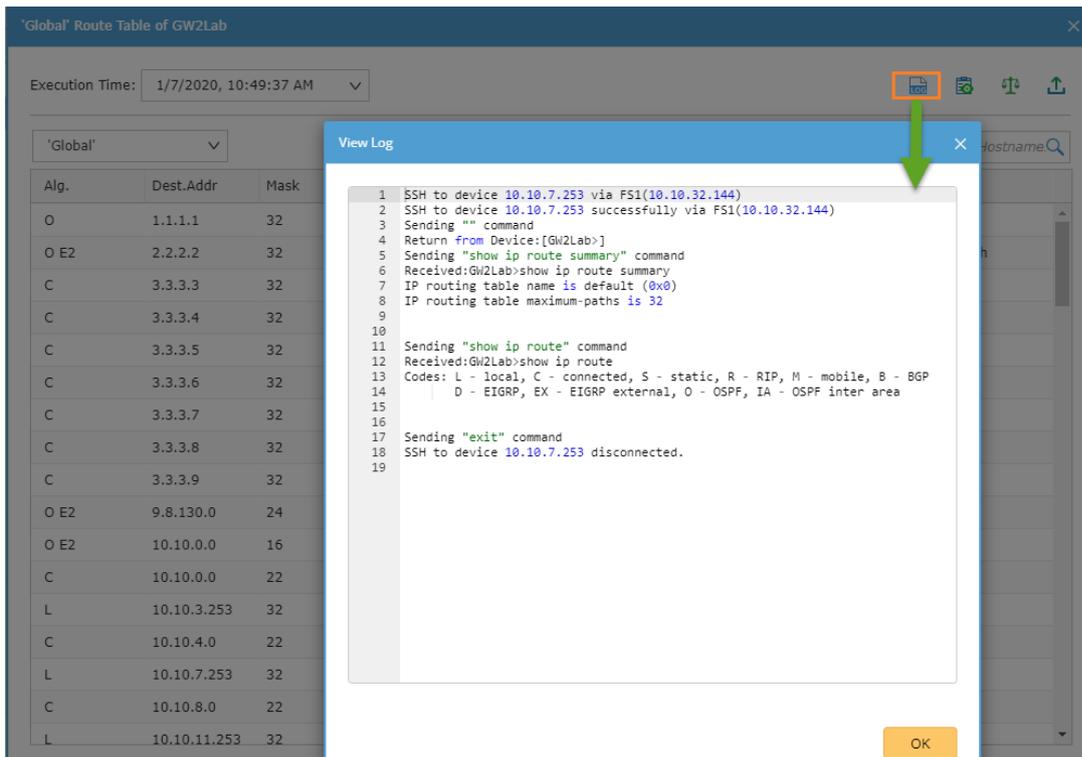


**Note:** IEv8.0 removes the **Path Analysis Set** option (used to select L2/L3 path) because it automatically selects path types to calculate based on interface dependency topology types during a path calculation.

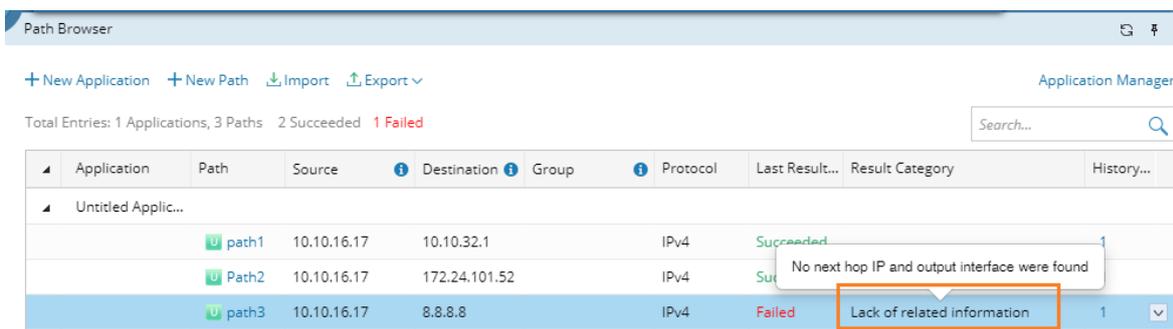
### 3.3.9.Supportability

- Add a **Debug Mode** option in Path Settings. When this option is checked, the Path will print system debugging messages to help troubleshoot path issues.
- Add logs for retrieving data tables and CLI commands during path calculation.

**Figure:** Execution log of route tables



- Add path message categories and error tips. The information in the result category can be exported along with the path exports in the Application Manager and Path Browser.



- Path logic (NCT, path script) dedicated to customers' specific usage can be updated and synced through Knowledge Cloud. Refer to [Knowledge Cloud](#) for more details.
- Support of Path Accuracy Ticketing. Refer to [Ticketing Sample](#) for more details.

## 3.4. Fully Extensible MPLS Cloud Framework

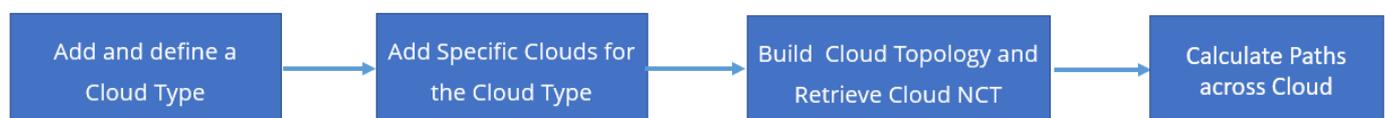
In the previous versions, MPLS Cloud was used to simulate an MPLS network that carries traffics for customer networks at different regions, which enabled users to calculate the traffics across the MPLS Cloud to make their network complete in the NetBrain system. However, as more cloud deployment scenarios and technologies increase, the limitations of the MPLS Cloud framework arise:

- It cannot simulate PE-side redundancy deployment scenarios because the CE interface cannot be configured to connect to multiple PE IPs.
- It cannot simulate CE-side redundancy deployment scenarios because multiple CE interfaces cannot be configured to connect to the same PE IP.
- It cannot add CE interfaces configured with IP unnumbered configuration.
- It is not extensible to support the deployment scenarios of Carrier Supporting Carriers (CSC) and Inter-Provider.
- It is not extensible to support other cloud technologies, such as VPLS.

To resolve the above limitations, IEv8.0 has redesigned the Cloud framework to make it a generic with high scalability. The new Cloud framework enables power users and NetBrain Platform Team to define a cloud-based via JSON and Python so that they can flexibly adjust and develop cloud functions based on network situations.

Besides MPLS Cloud, the new Cloud framework uses the Cloud concept to represent the network connecting to users' networks but not managed by the users, for different technologies such as VPLS and IPv6.

### Use Flow



#### 3.4.1. Add Cloud Type

Each cloud type is composed of four components, which are used to build the data model for cloud topology and path calculation:

Component	Description
Property	Define which parameters need to be filled in. For an MPLS Cloud, the parameters include PE devices, CE interfaces, and other parameters.
NCT	Define the NCT needed by a cloud to calculate the path crossing cloud. For example, the virtual route table for an MPLS Cloud, the virtual MAC table for a VPLS cloud.
Topology	Define the logic to calculate the topology for a cloud and its neighbor devices.
Icon	Define the Cloud icon displayed in the system.

**Note:** After adding or modifying a cloud type, you need to add or adjust the path scripts used to calculate paths for this cloud type.

To enable users to use the MPLS Cloud function directly, the system offers a built-in “MPLS L3 VPN” cloud type and power users can define an MPLS cloud as usual based on this cloud type.

**Example:** Built-in Cloud Type for MPLS L3 VPN

The screenshot shows the 'Tenant Management - Topology' interface. At the top, there are tabs for 'User Authorization', 'Domain List', and 'Cloud Type Definition'. The 'Cloud Type Definition' tab is active, showing a form for defining a cloud type. The form has a 'Name' field with 'MPLS L3 VPN' and an 'ID' field with '1024'. Below the form, there are tabs for 'Property', 'NCT Table', 'Topology', and 'Icon'. The 'Property' tab is selected, and a 'JSON' editor is open. The JSON editor contains the following code:

```

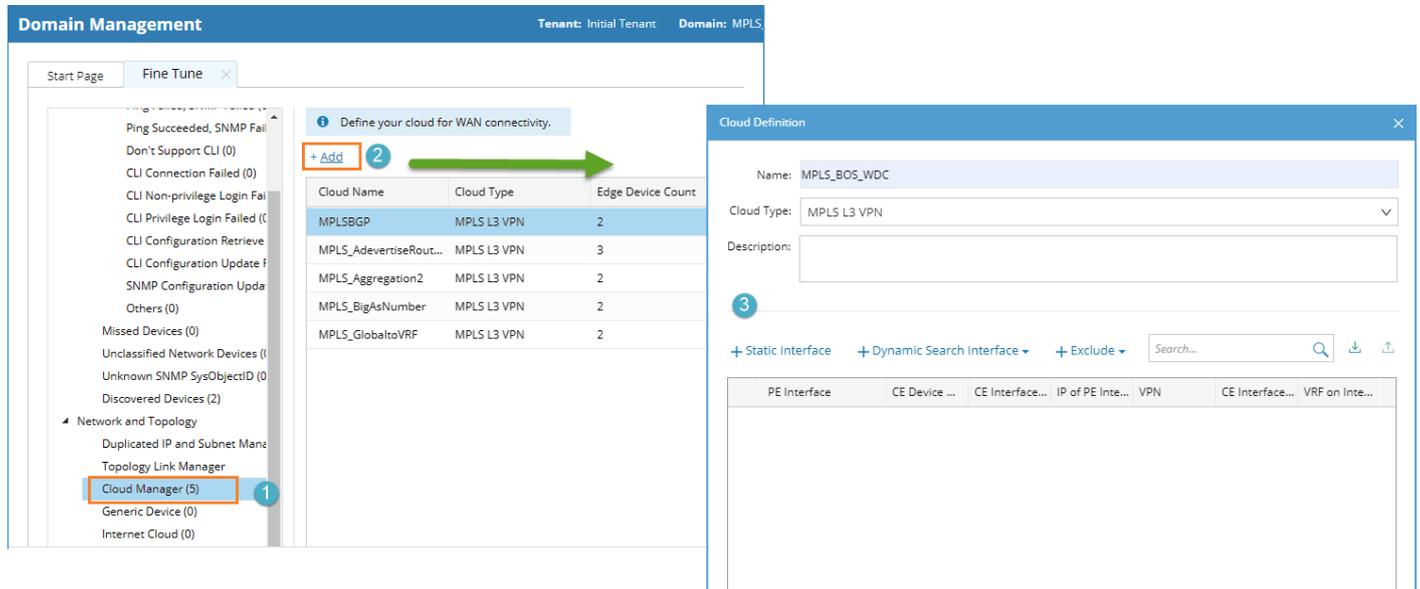
1 [
2   {
3     "name": "cloud_interface",
4     "displayName": "PE Interface",
5     "dataType": "string",
6     "required": true
7   },
8   {
9     "name": "cloud_ip",
10    "displayName": "IP of PE Interface",
11    "dataType": "ip",
12    "required": true
13  },
14  {
15    "name": "cloud_vpn",
16    "displayName": "VPN",
17    "dataType": "string",
18    "required": false
19  },
20  {
21    "name": "edge_device",

```

A 'Save' button is located at the bottom right of the JSON editor.

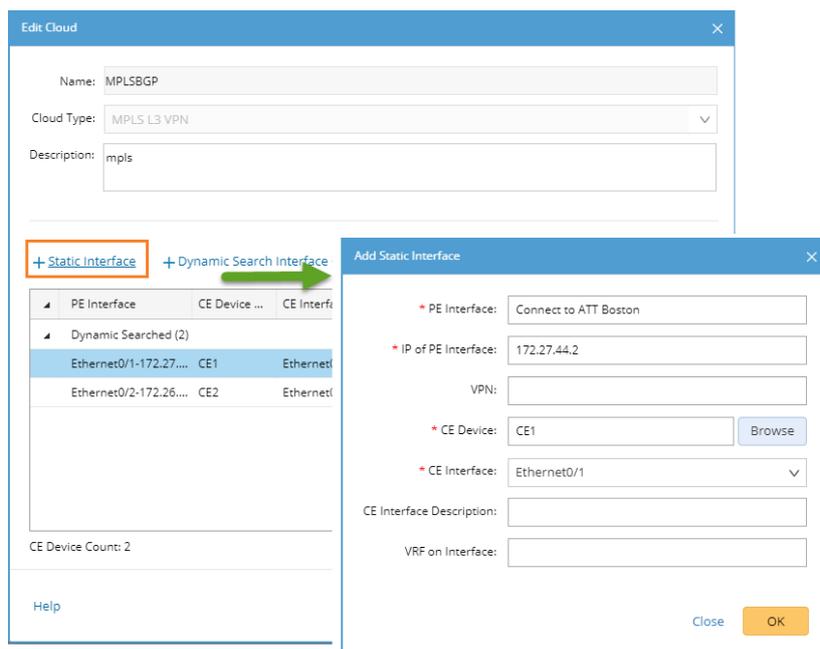
## 3.4.2. Define Specific Clouds for a Cloud Type

After defining a cloud type, power users can define specific clouds in their domain. After they select a cloud type, the system automatically loads the parameters defined in the cloud type.



Compared with the previous versions, there are some changes in the MPLS Cloud definition:

1. Select **MPLS L3 VPN** as the cloud type.
2. Fill in the PE interface instead of the routing protocol when manually defining an MPLS cloud via Static Interface
3. Fill in any content based on display needs on a map for the configuration of a PE interface. For example, fill in the PE interface "Connect to ATT Boston", then they will see that MPLS Cloud uses this interface to connect to the CE device on the map.



### 3.4.3. Build Cloud Topology and NCT Data for Path Calculation

After defining a cloud, power users need to build the connection topology between a cloud and its neighbors and retrieve the NCT data used for path calculation through a benchmark task.

Take MPLS L3 VPN for example:

- Build the MPLS topology via the **IPv4 L3 Topology** option.

Build Topology	
Enable	Operation Name
<input type="checkbox"/>	IPv4 L3 Topology
<input type="checkbox"/>	IPv6 L3 Topology

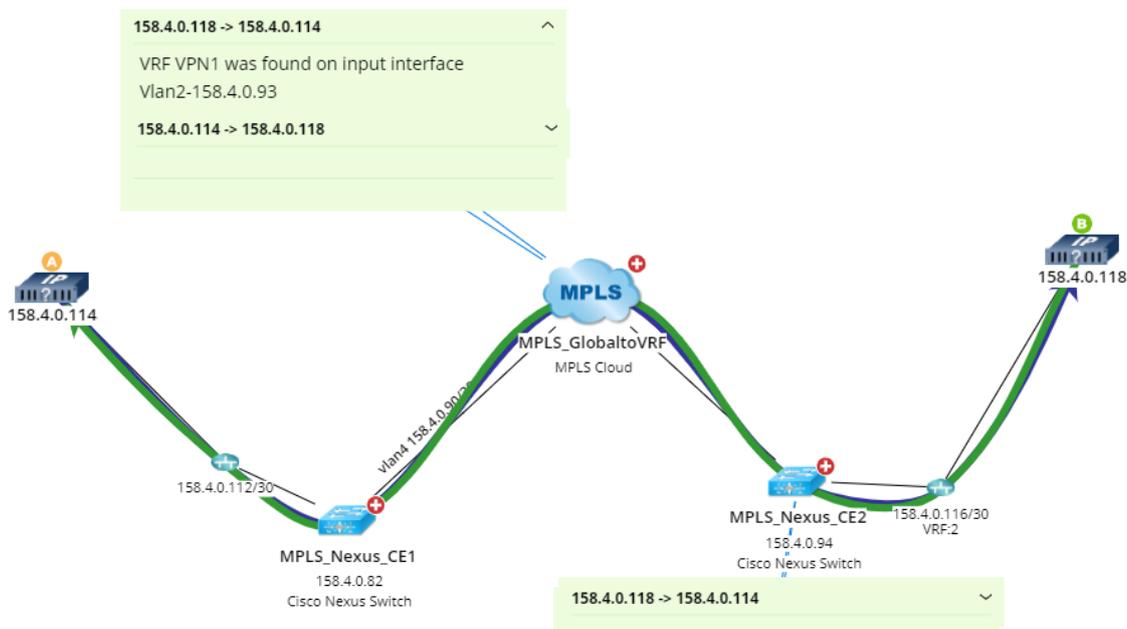
- Calculate the virtual route table via the **Recalculate Cloud NCT Table** option.

Update Cloud	
Enable	Operation Name
<input type="checkbox"/>	Recalculated Cloud
<input type="checkbox"/>	Recalculate Cloud NCT Table

### 3.4.4. Calculate Path Across a Cloud

After building the topology and NCT data, end users can calculate a path across the defined cloud.

**Example:** A Path across MPLS Cloud



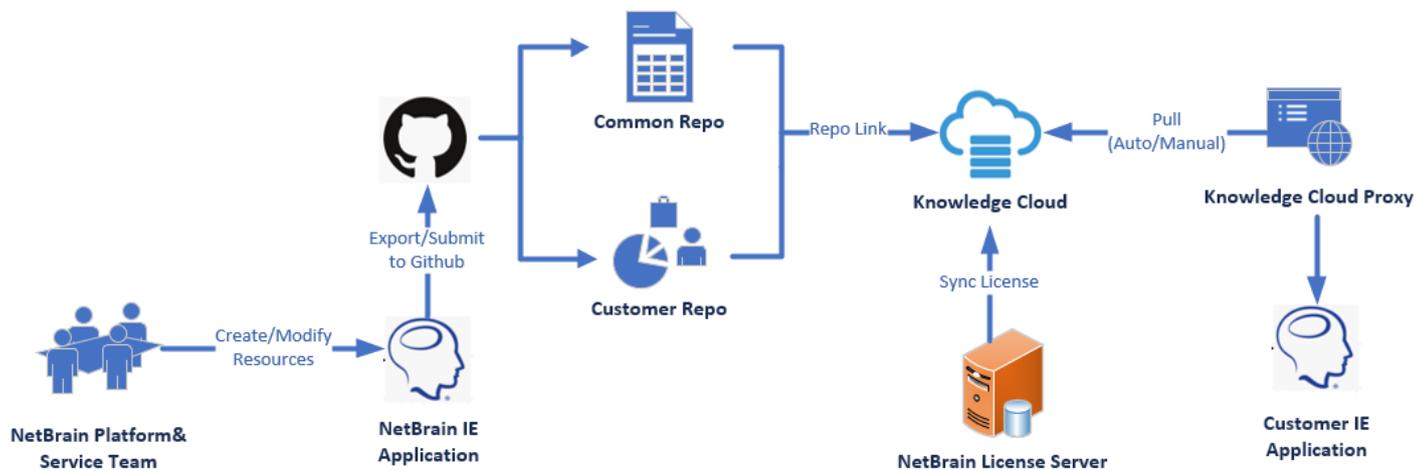
## 3.5. Knowledge Cloud

Knowledge Cloud is a centralized resource base housing many types of NetBrain resources. Auto-updating resources via Knowledge Cloud is aiming to provide support for the following scenarios:

1. Auto-update varieties of resources (DVT, Runbook, Platform Plugin, New Tech, Device Type, Driver, etc.) that apply to the user's specific IE version constantly.
2. Auto-update patches for built-in resources. Customized resources (Driver/Path Scripts/Platform Plugin) that are dedicated to customers' specific usage can be reapplied automatically after the system upgrade.
3. Auto-update customized resources (DVT, Runbook) according to the request.

**Note:** Access to Knowledge Cloud Domain (<https://knowledgecloud.netbraintech.com/>) is required on Customers' Web Servers.

### Reference Flow



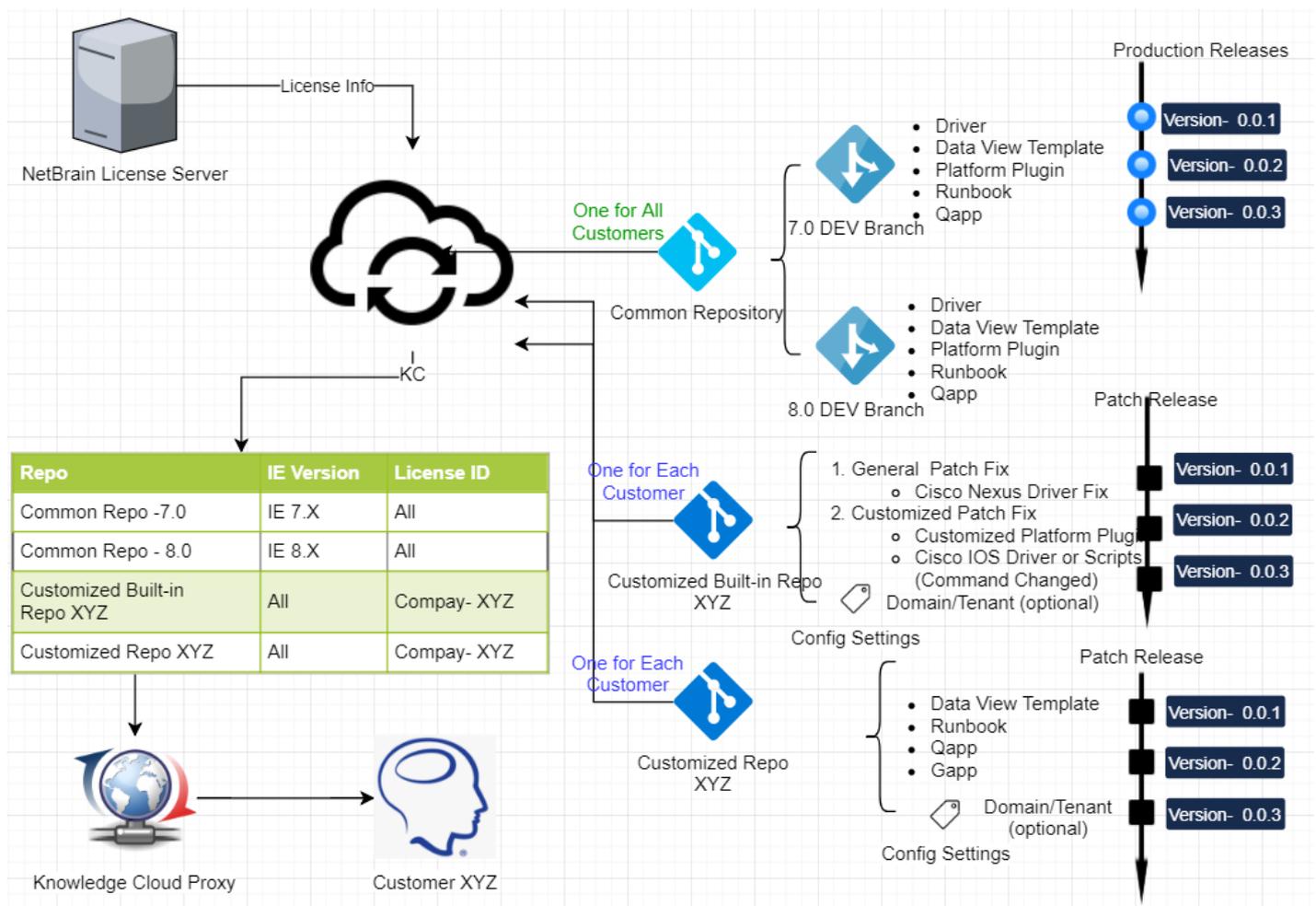
Below is a full list of resources currently supported by Knowledge Cloud:

- Qapp/Gapp
- Runbook Template/Data View Template
- Parser Library
- Driver/Device Type/Vendor Model Table
- Device Icon/Image/Topology Link Type (Ipv4, Ipv6, etc.)
- Media Type

- GDR Properties
- Tech Spec/Schema/Visual Space
- API Plugin/Platform Plugin
- SPOG URL
- Device Group
- Global Python Scripts (including Path Scripts)
- Variable Mapping & Global Variable
- Golden Baseline Dynamic Analysis Logic
- Interface Type Translation
- Default Data View Template
- Network Tree Category and Different Views

### 3.5.1. Knowledge Cloud Management (NetBrain Internal Use Only)

The diagram below depicts the storage structure of various resources managed by Knowledge Cloud:



GitHub Repositories house following three types of repositories for the consumption of different types of customers:

- **Common Repository:** a repository storing universal software resources that are applicable to all customers. This repository contains the branches of major software versions. For example:
  - **IEv7.0 DEV Branch:** resources of this branch can be consumed by one or multiple minor software versions (e.g., IEv7.0a, IEv7.0b, IEv7.1, IEv7.1a, etc.)
  - **IEv8.0 DEV Branch:** includes unique IEv8.0 system resources (e.g., Data View Template/Runbook) which are incompatible with previous major versions. Resources of this branch can be consumed by one or multiple minor software versions (e.g., IEv8.0, IEv8.0a, IEv8.1, etc.)
- **Customized Built-in Repository - \$Customer name:** a repository storing customized resources based on individual customer's special needs. The software patches in this repository are version-agnostic (applicable to all software versions owned by the specific customer). There are two types of resources in this repository:
  - **General Patch Fix:** these general patch resources will be removed from the repository after each major release or fix.
  - **Customized Patch Fix:** these patch resources include customized logics (e.g., Customized Platform Plugin). They need to be compatible with the user's software version and always remain in Customized Build-in Repository.
- **Customized Repository:** a repository storing DVT/Runbook/Qapp/Gapp resources. The major difference between Customized Repository and Customized Build-in Repository is that Customized Repository is designed to be managed by Network Automation Team, while Customized Build-in Repository is designed to be managed by Platform Team.

---

### 3.5.2.Common Repo Release Management

The following workflow applies to the common repo release aiming to be consumed by all customers:

1. **Release in GitHub:** release relevant resources in GitHub
2. **Refresh in Knowledge Cloud:** refresh data in Knowledge Cloud to view latest version information
3. **Select QA License to Test:** QA Team to select a portion of QA Licenses to test the resources and deployment flow
4. **Request Publish:** NetBrain internal user (with request publish permission) to select specific or all customers to request resource publish

5. **Approve Publish:** NetBrain internal user (with common repo release approval permission) to approve the publish requests, so end users can retrieve the latest software update info when checking update from IE
6. **Customer Retrieves Updated Resources:** Customer connects the IE system with Knowledge Cloud; retrieving latest software update info (IE sends a request to Knowledge Cloud to retrieve system update info every 12 hours)

**Note:** NetBrain internal users without permission to visit Knowledge Cloud have the option to manually export the resources from Knowledge Cloud and import the resources to the IE system to help end users perform resource updates.

### 3.5.3. Customized Repo Release Management

The following workflow depicts the release procedures for Data View Templates. Platform Engineer/Automation Engineer can follow a similar flow to conduct other customized repo release when receiving users' update requests:

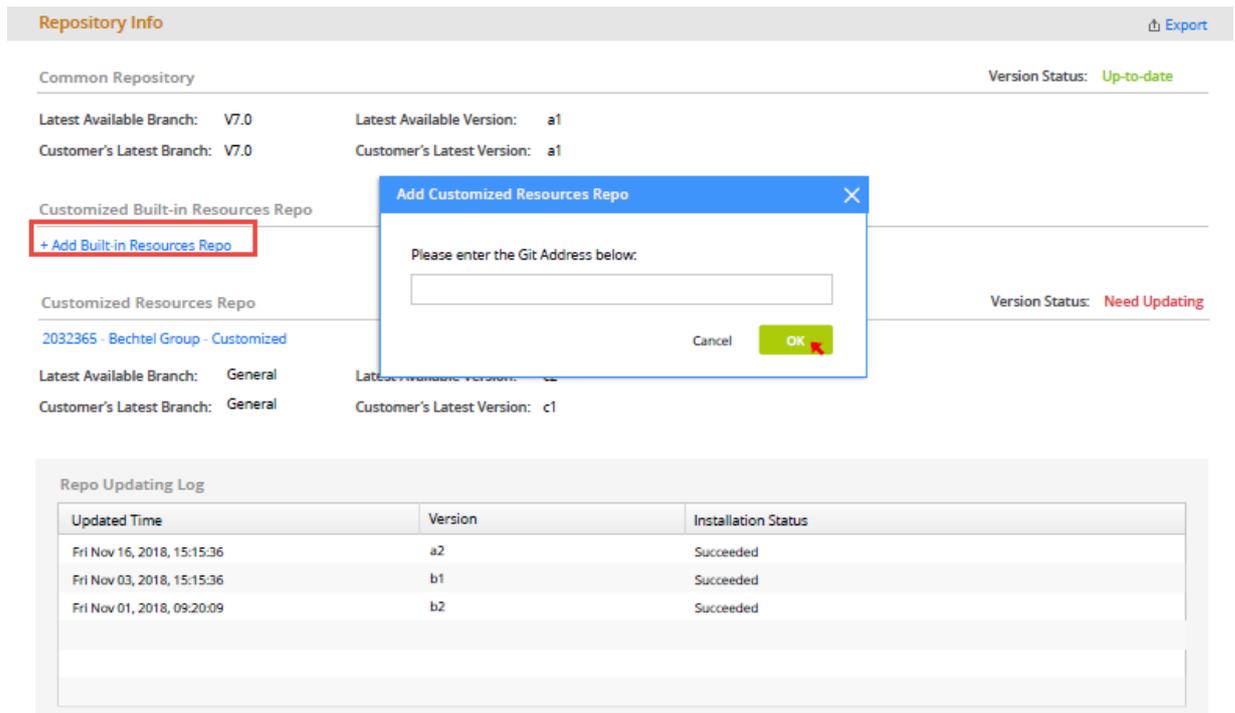
1. **Create/Test DVT Template in IE:** Platform Engineer/Automation Engineer creates corresponding Data View Template in the IE system, and configures the associated Parsers, Variable Mappings, and Golden Baseline Settings.
2. **Export DVT Template, including all dependencies (Parser/DVT/GDR/Variable Mapping):** Platform Engineer exports the corresponding DVT and its dependencies by using the Export Tool prepared by DEV team.
3. **View Customer/Repo relationship from KC:** Service Engineer to verify if the current customer has associated customized repo by searching for License ID or Customer Name in Customer Manager interface:

- If yes, take notes of the repository name and address;

Customer Manager

License ID	Customer Name	Version	Common Repo Version	Customized Built-in Repo	Customized Repo	Maintenance Info
58625564	AT&T	IE 7.1a - c1049207921e	7.1a-a1	20326436 - ATT - Built-in	20324643 - ATT - Customized	Valid
25455653	Verizon	IE 8.0 - a265607921e	7.1a-a2	N/A	N/A	Expired
48635569	T1	IE 8.0 - c056667921f	7.1a-a3	N/A	N/A	Valid
John@att.com	TR	N/A	N/A			Valid

- If not, create the customized repository on GitHub; open the Customer Detail interface, and add the corresponding repo by clicking the hyperlink 'Add Customized & Built-in Repo'.



4. **Service Engineer Commits the Code to GitHub:** Data View Template will be pushed to all Tenants by default; config file can be created to push resources to the target Tenant/Domain.
5. **Refresh in Knowledge Cloud:** Refresh data in Knowledge Cloud to view the latest version information.
6. **Request Publish:** NetBrain internal user (with request publish permission) to select specific or all customers to request resource publish.
7. **Approve/Reject Publish:** NetBrain internal user (with custom repo release approval permission) will receive the publish request and decide to approve or reject the request.

### 3.5.4. Update from IE System

There are three ways to perform resource updates (powered by Knowledge Cloud) from an end user perspective:

1. **Auto Update from NetBrain:** NetBrain IE fetches the latest resources from Knowledge Cloud constantly (time interval every 12 hours, start time between 12 AM and 3 AM). To take advantage of this feature, end user needs to ensure **Auto Update from NetBrain** option is checked in the System Management page.
2. **Check Update Now:** In some circumstances, end user would desire to apply the latest resources immediately to his/her IE system, he/she can click **Check Update Now** button in System Management interface to manually initiate the auto-update process.

3. **Import Resources:** Some end users can only perform off-line updates due to security concerns. In this scenario, NetBrain Support Team will download the update resources from the Knowledge Cloud and forward these resources to the end users so they can click **Import Resources** button in System Management interface and upload the corresponding update package for the IE system to process.

**Resource Update** [Proxy Settings](#)

Last Updated on: 11/27/2018

Executor System: [View Update History](#)

Auto Update from NetBrain

Share variable analysis results with NetBrain

[Check Update Now](#) [Import Resources](#)

The current resources are the newest available resources.

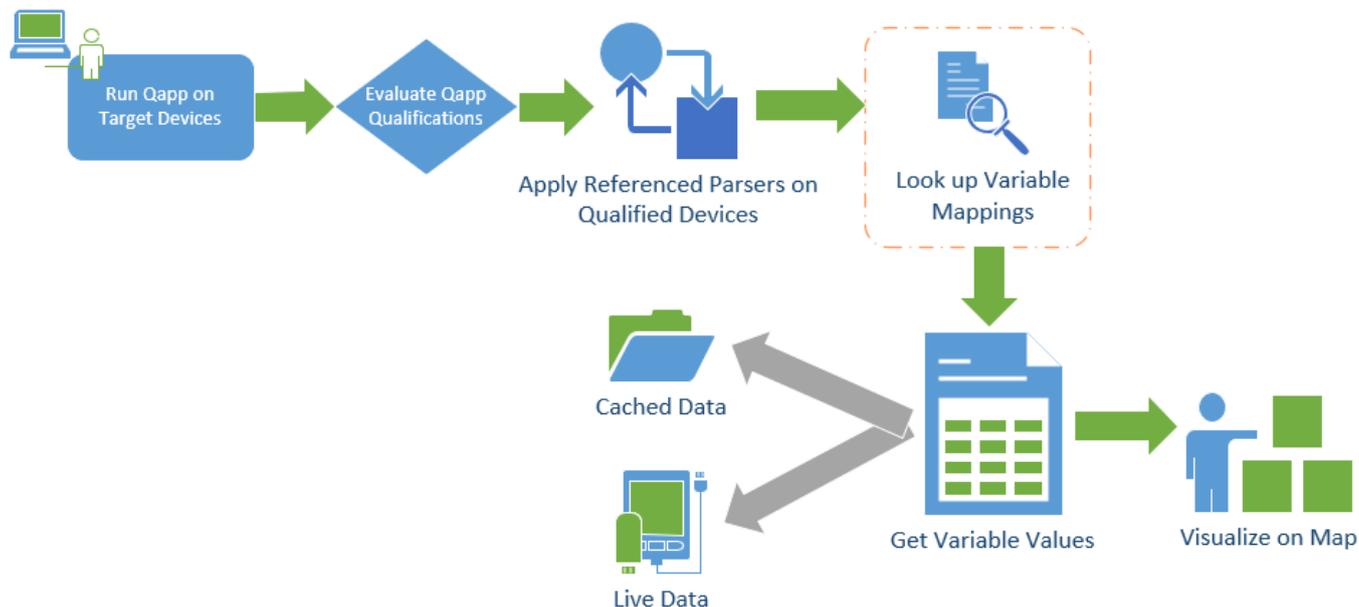
**Note:** The following restrictions have been introduced to avoid compatibility issues caused by modification of Built-in Resources:

- Built-in Resources can only be modified by NetBrain staff;
- End users can view (but not modify) their Built-in Resources and save them as Customized Resources.

## 3.6. Enhancements to Variable Mappings for Multi-Vendor Support

The essence of Variable Mapping is to define mapping relationships of different parser variables among different vendors and different data sources. If one variable for a specific vendor or source is supported in NetBrain Automation, other variables for the rest of the vendors or sources can also be supported.

Variable mappings are automatically applied in a “single pane of glass” view for dynamic mapping and automation. The following flowchart describes how Variable Mapping applies to Qapp Automation.



Variable Mapping was initially introduced in IEv7.1. The system predefined a series of built-in variable mappings and also allowed users to customize new mappings. However, it was still not easy for users to define, manage and maintain such a huge matrix of mapping relationships when there was no global alias at all. The import and export of mappings were also complex.

To resolve the limitations, IEv8.0 adds more capability and enhancements to variable mapping definitions, including:

- Introduce a new definition “Global Variable” to serve as the alias to map parser variables.
- Introduce a new definition “Namespace” to host global variables and their mapping relationships.
- Divide global variable resources into two categories, built-in and customized, to simplify resources sharing via Knowledge Cloud. Built-in variables are predefined and maintained by NetBrain, and can be saved as customized variables.
- The original Variable Mapping page becomes a read-only table to view mapping details. Add a button to view mapping details of parser variables by selecting a Data View Template.

### 3.6.1. Manage Namespace and Variable Mappings

Each Namespace is composed of a group of variables that have similar network semantics. Generally, every variable inside a Namespace is parsed through a single CLI command, for a single device type. With the

organization by Namespace, Global Variable can be further defined, as the global alias of Variable Mappings for multi-vendor.

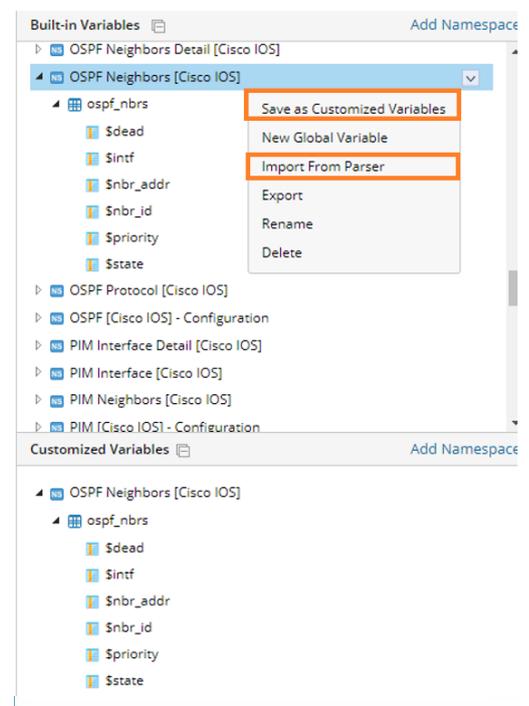
Global Variable contains two categories: built-in variables and customized variables. The system allows a single parser variable to appear in each category only once at most.

- As long as a parser variable appears in the customized category, the system will only look up variable mappings for multi-vendor and multi-source in the customized category.
- If a parser variable only appears in the built-in category, the system will look up variable mappings for multi-vendor and multi-source in the built-in category only.

## Add Variable Mapping by Importing Parser

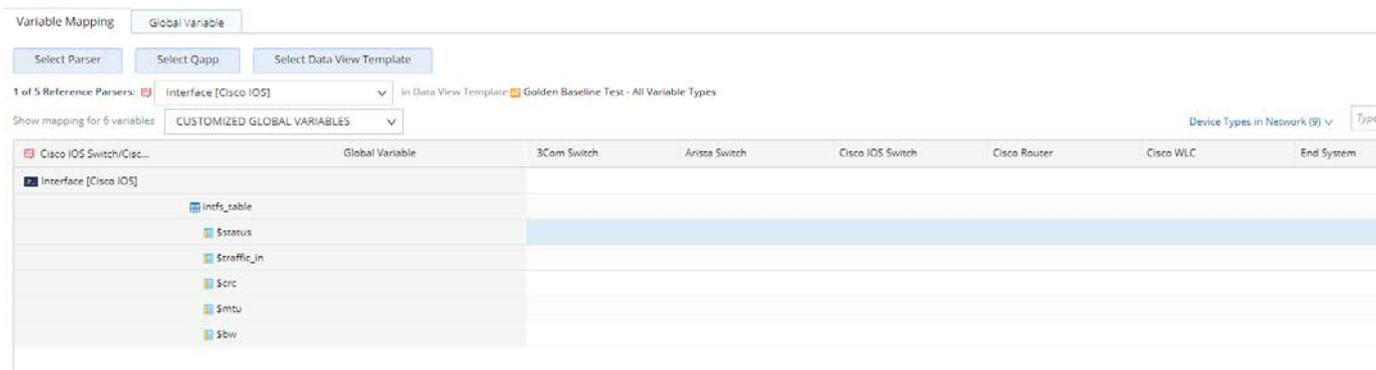
If all variables for a single parser are required as global variables, users can import the parser to a Namespace so that all variables for this parser can be added as global variables at one time.

**Note:** Users are not allowed to modify built-in global variables. If there is any request to add new vendors or add new sources for existing vendors, they can either submit tickets to NetBrain Support Team, or save a Built-in Namespace as a Customized Namespace to extend the support by themselves.



### 3.6.2.View Variable Mappings for DVT

IEv8.0 allows users to select a Data View Template to view the variable mappings inside among multiple device types, so that the missing mappings for required device types can be addressed quickly.



## 3.7. SDN Enhancements

Cisco ACI-NetBrain integration is aimed to provide enhanced network operations to application-centric data centers enabled by Cisco ACI. Following improvements have been introduced in NetBrain IEv8.0 to further streamline NetOps's day-to-day operational workflow:

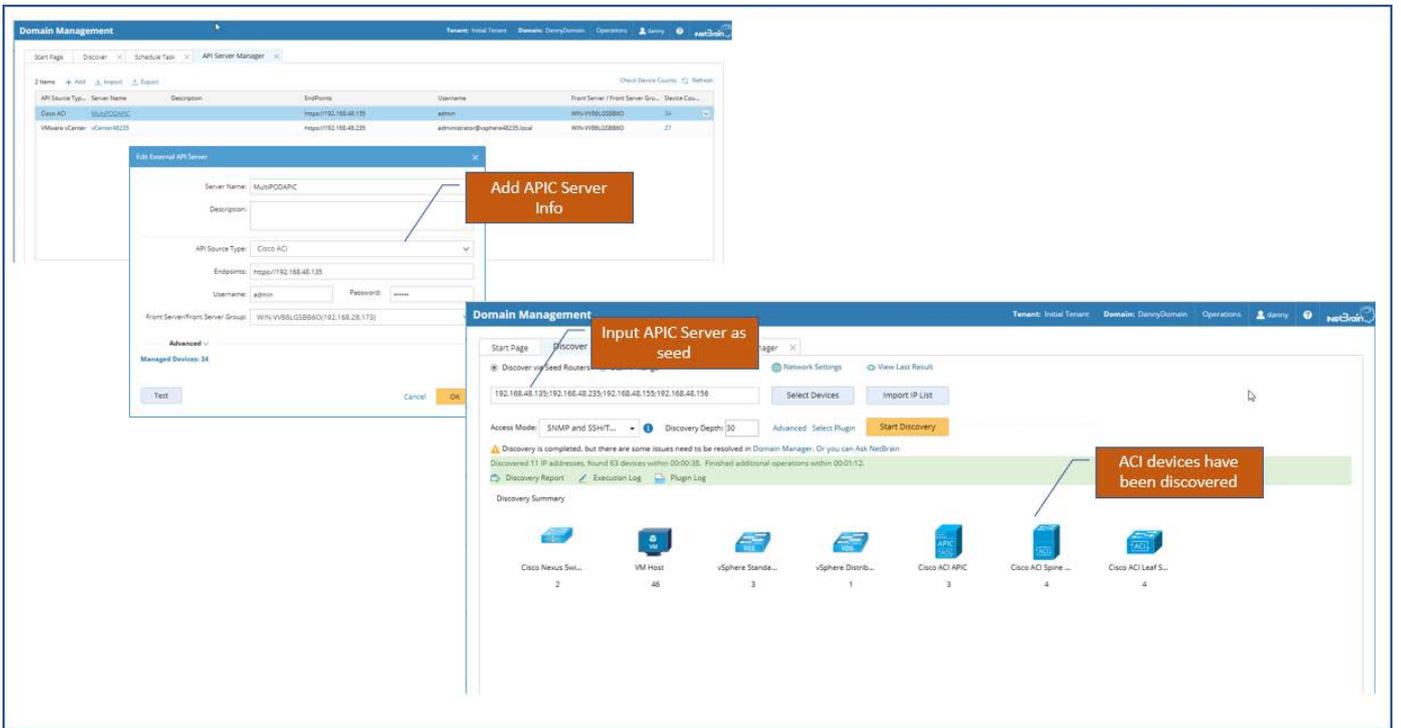
- [ACI Multi-POD Support](#)
- [Calculate Path in ACI Environment](#)
- [Context Map Improvement](#)

### 3.7.1. ACI Multi-POD Support

IEv8.0 SDN Module provides comprehensive support for discovering and mapping Cisco ACI Multi-Pod fabric.

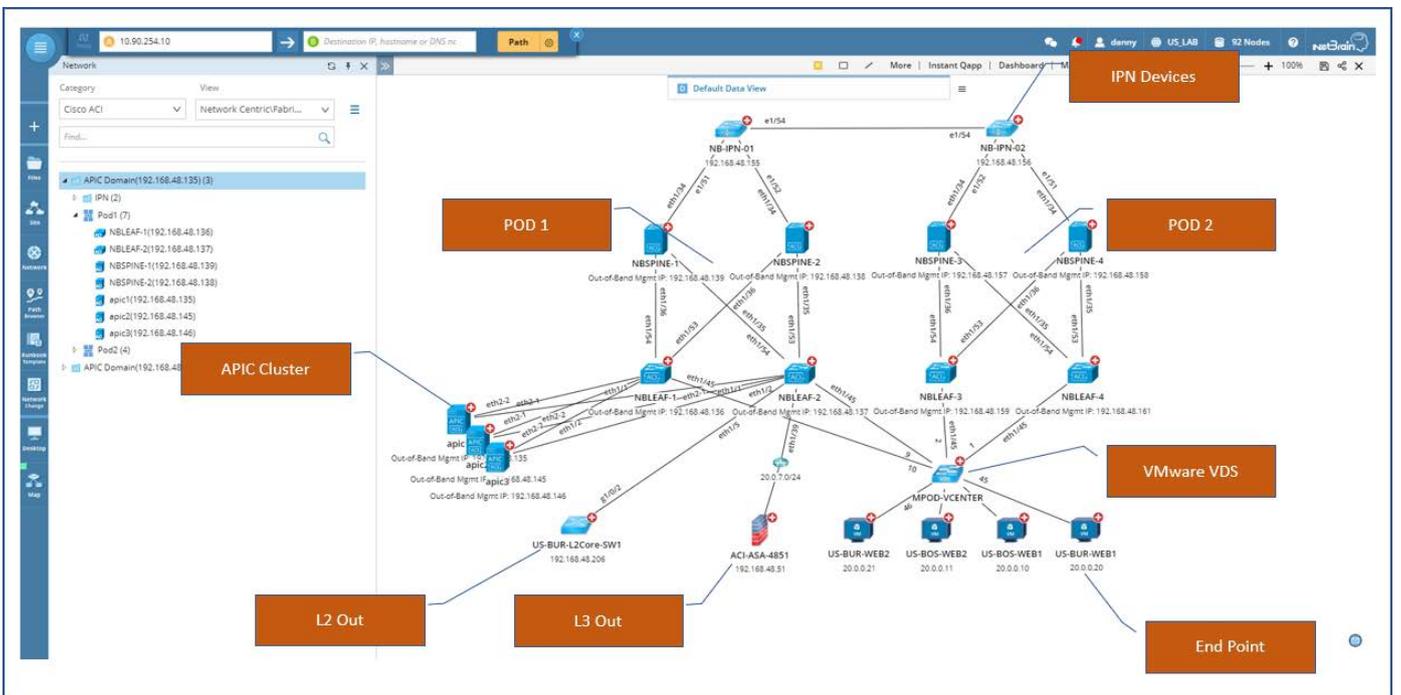
#### Discovery

The following example demonstrates the NetBrain discovery interface in a Cisco ACI Multi-POD environment, including IPN devices that are connected to the Spine switch.



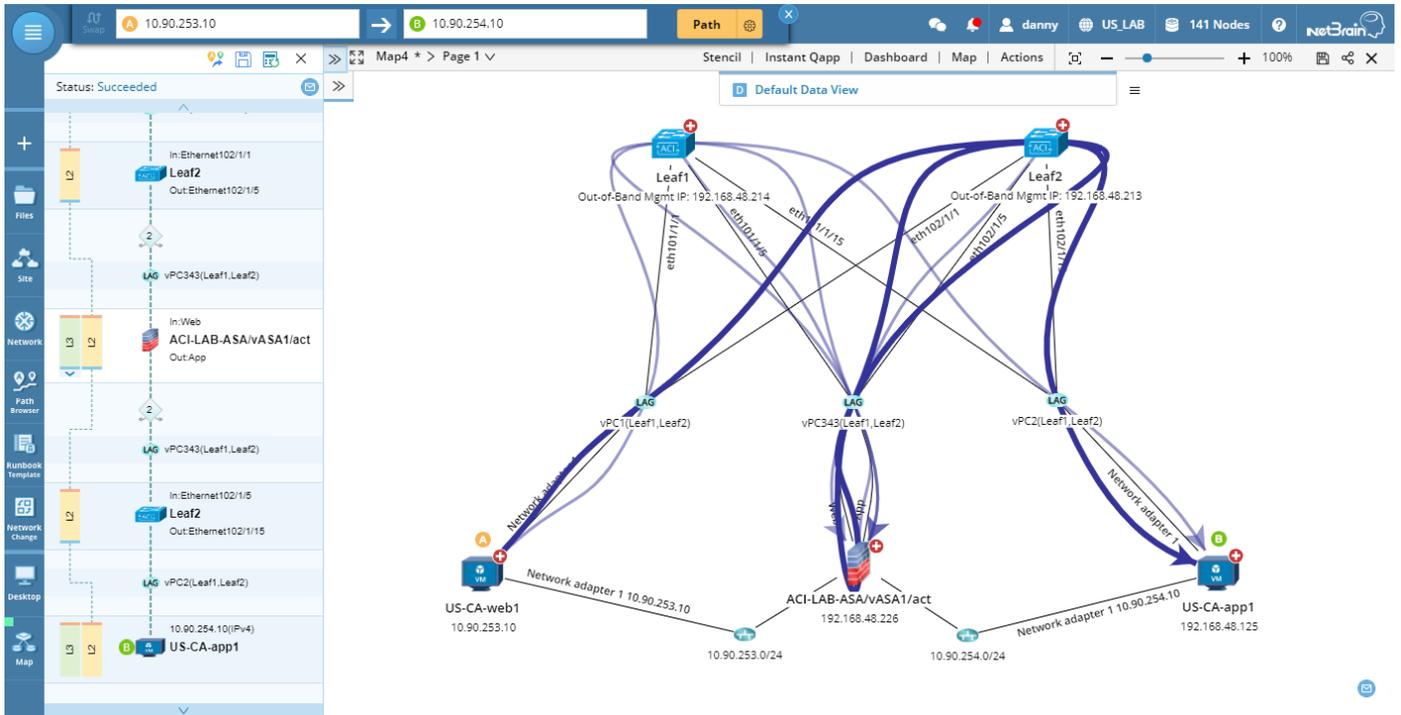
## Mapping

The following example demonstrates the topology map including all devices (IPN, Spine, Leaf, APIC, L3 Out, L2 Out, End Point) and their connectivity in a Multi-POD environment:



## 3.7.2. Calculate Path in ACI Environment

### Gateway on Legacy Device

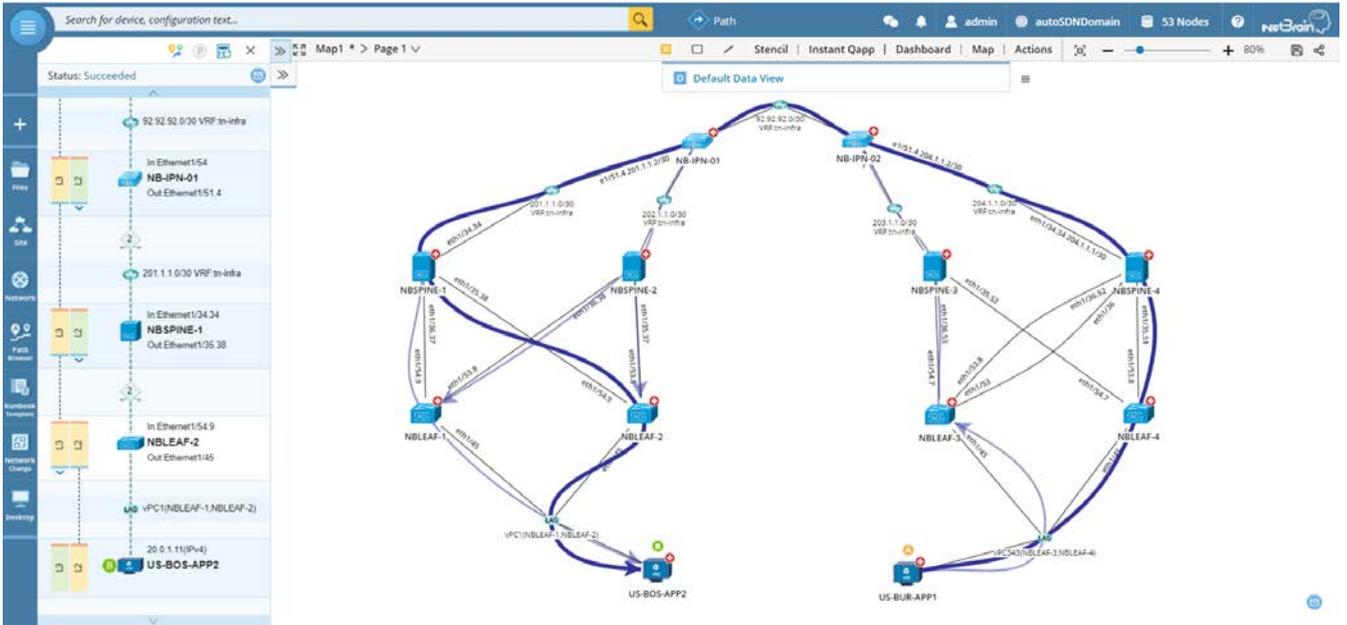


In the above scenario, the ACI Fabric is treated as an L2 switch in a legacy network.

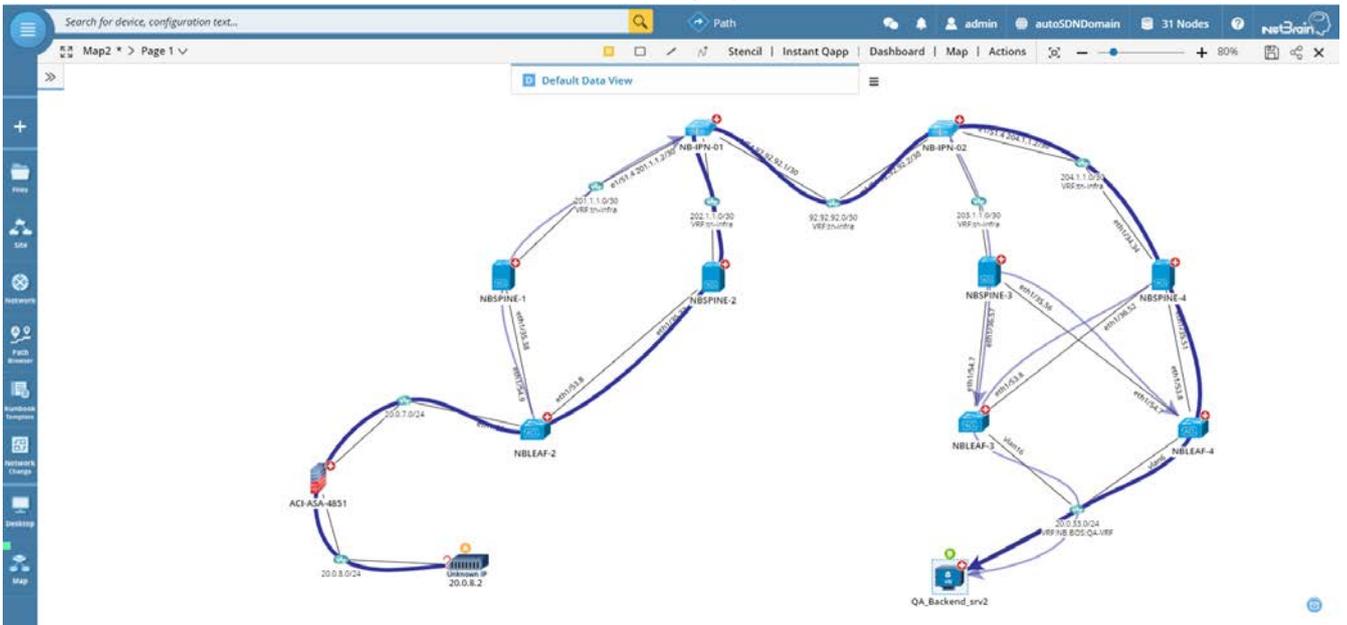
### Anycast Gateway within ACI

Anycast Gateway Path in ACI deployment is supported in the following scenarios:

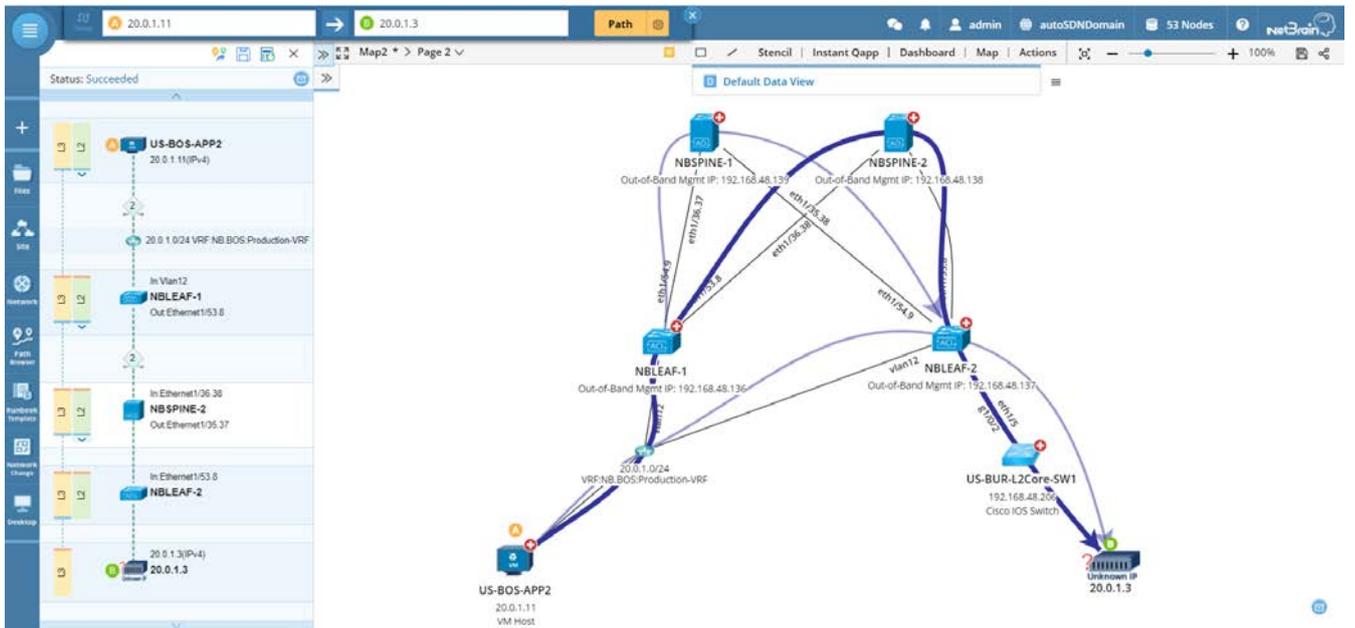
- 1) Source and Destination End Points are in the same ACI Fabric.



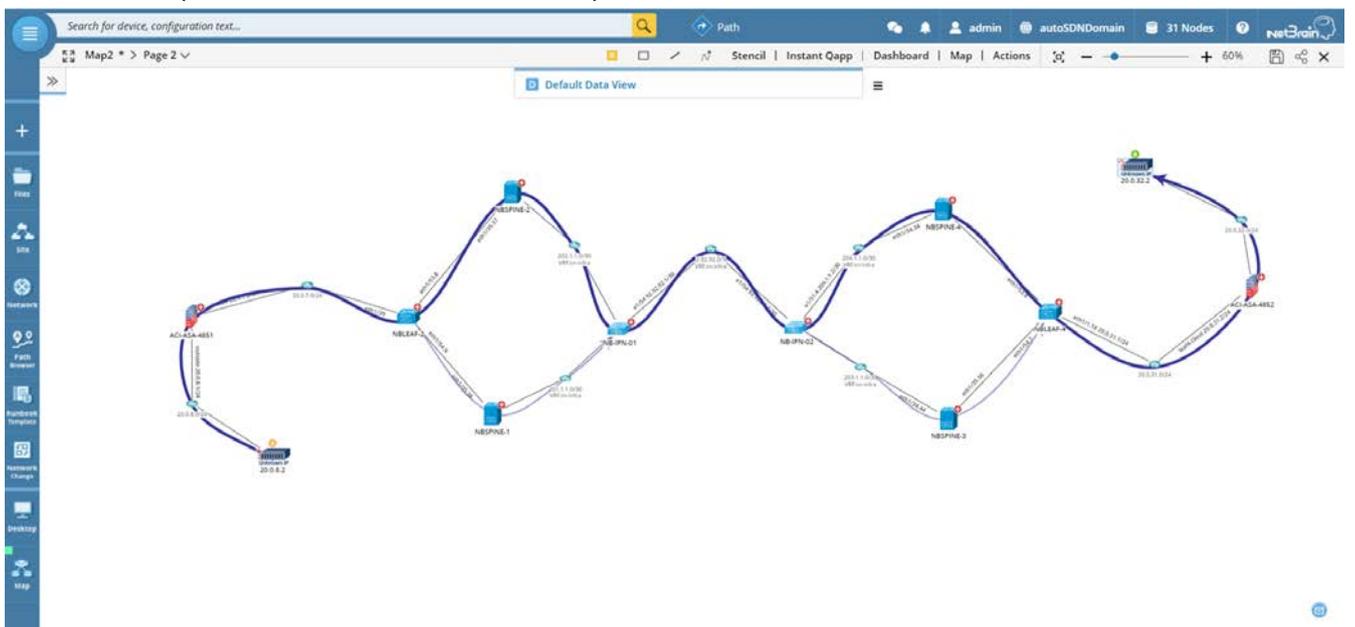
2) Source or Destination End Point in ACI Fabric (from a legacy network to the inside of ACI, or vice versa).



3) Source or Destination End Point uses L2 Out to connect to ACI Fabric (might not be fully supported in IEv8.0).



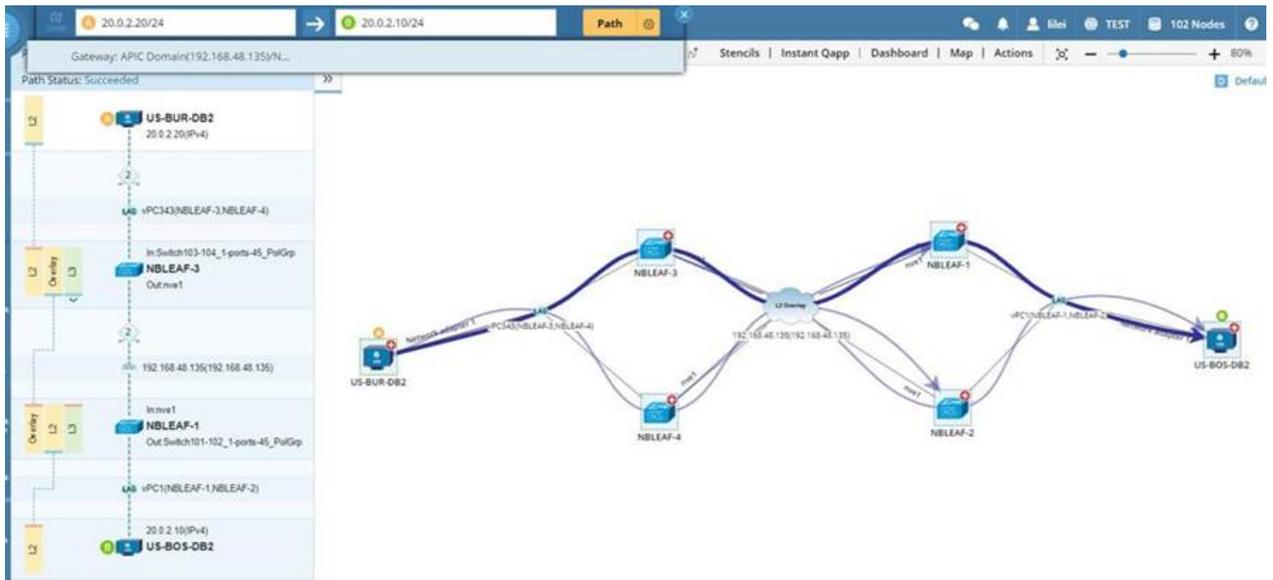
4) Transit Route (ACI Fabric likes a transit network).



## Calculate ACI L2 Overlay Path

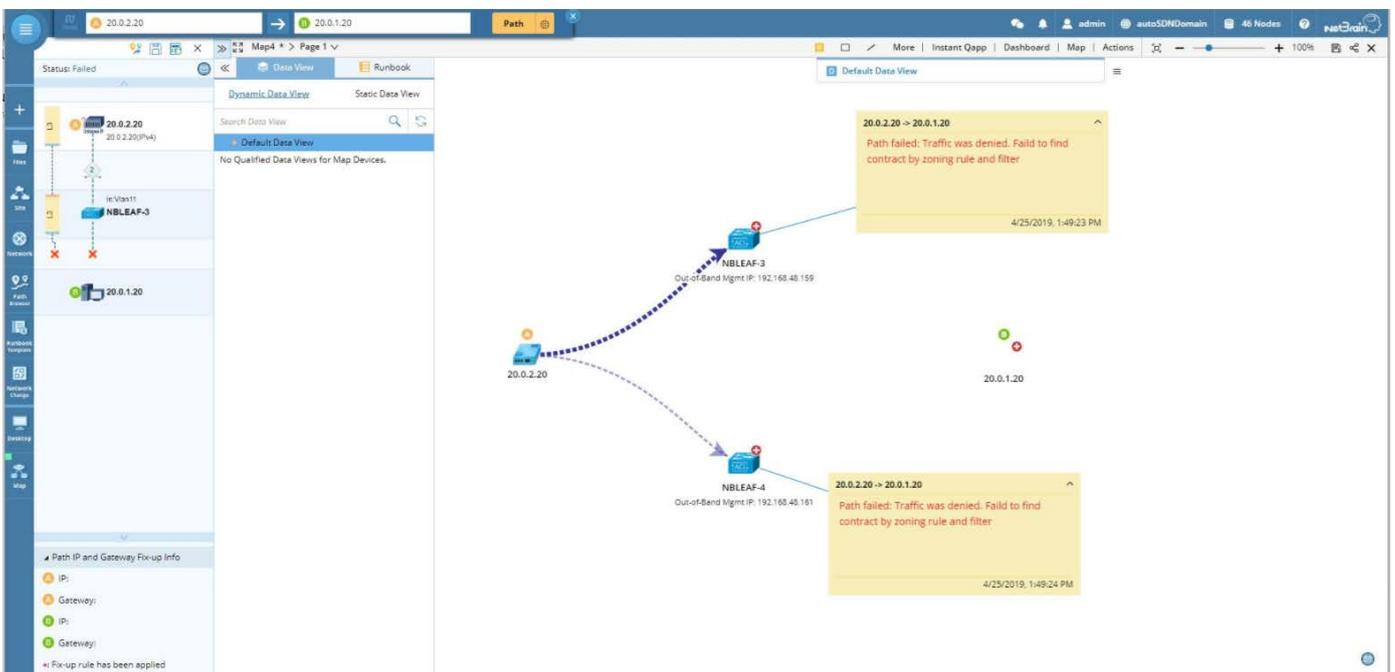
- Connect all leaf devices in the same ACI Fabric that have an L2 overlay connection to one L2 Overlay Media.

- When calculating a pure L2 path crossing an ACI Fabric, the system calculates L2 overlay and underlay path based on topology dependency.



## Consider Contract in Path Calculation

NetBrain will check if the contract denies the traffic while executing path calculation. Detailed information about the contract will be displayed in Path Log and Tip pane.



### 3.7.3.Context Map Improvement

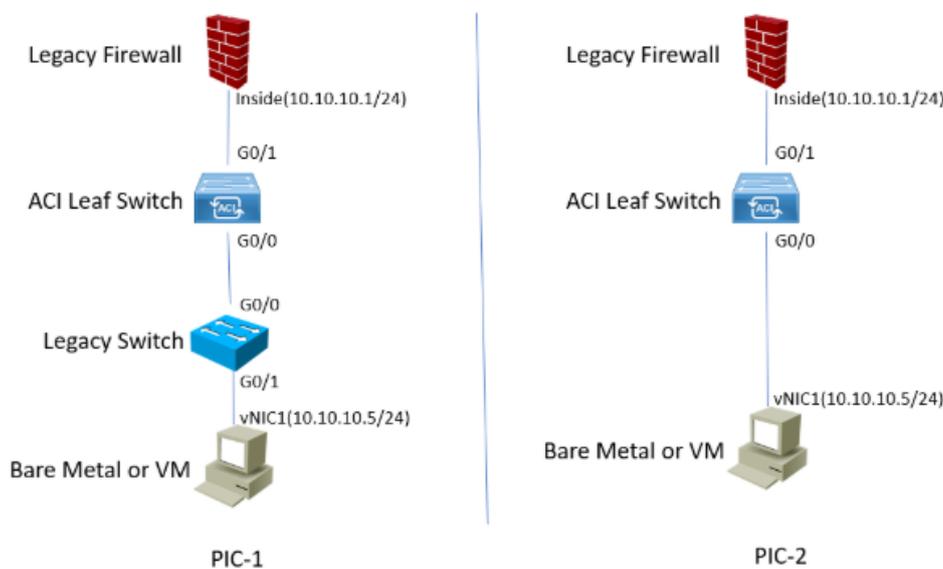
For the details about SDN context map improvements in IEv8.0, refer to the [Appendix](#).

### 3.7.4.Enhanced L2 Topology of SDN Nodes

IEv8.0 has enhanced the L2 topology algorithm to improve the accuracy of L2 Topology in the virtualization, legacy and SDN hybrid network.

#### Gateway of a VM host in Legacy Network

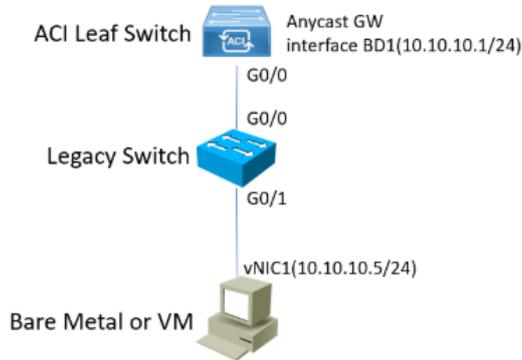
In the previous versions, when the gateway of a VM host was a legacy device and the ACI Leaf Switch connected to the gateway worked as a pure L2 switch, the system cannot calculate the L2 path in this network scenario because of topology issues.



IEv8.0 has resolved the L2 topology connection issues and supported the path calculation by optimizing topology calculation logic: When generating an end system, if an entry in the Global Endpoint Table has only MAC Address without IP Address, the system uses the MAC Address to find the IP + MAC entry without a switchport in the One-IP Table, generates a legacy end system and then save the L2 Topology relationship to ACI L2 Topology Table.

## Gateway in ACI and End Point Connecting to legacy Switch

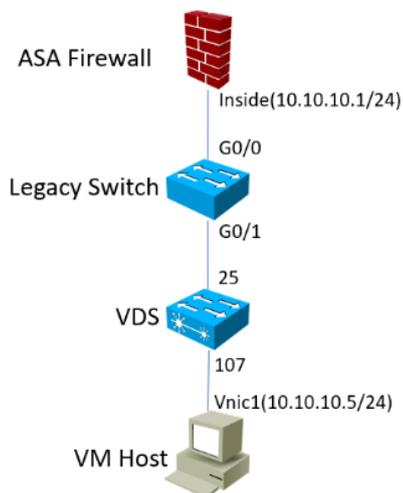
In the previous versions, when the gateway of a VM host was an ACI Leaf Switch and there was a legacy device between the gateway device and VM host, the system cannot calculate the connection between the VM host and gateway device.



IEv8.0 has fixed this issue by optimizing the L2 topology calculation logic in this scenario.

## VM Host's L3 Neighbor is ASA Firewall

An ASA Firewall does not support NDP. In the previous versions, when a VM host connected to VDS → Legacy Device → ASA Firewall (shown as the figure below), the system would generate two L2 topology connections for the VM host, one calculated based on legacy topology logic (VM Host Vnic 1 ↔ Legacy Switch G0/1), the other based on SDN L2 topology logic (VM Host Vnic 1 ↔ VDS107).



IEv8.0 adds the legacy topology to Multi-Source Topology Priority Rules and relies on the following priority to select topology when a topology is generated from different sources: NSX > vCenter > Legacy > ACI.

## Duplicate VRF and IP Subnet

In the previous versions, when ACI Fabric Instances deployed in different regions were configured with the same VRF and Any Gateway, these ACI instances would connect to the same one LAN Media when NetBrain calculated topology for them.

IEv8.0 has resolved the issue by generating and assigning different zones for ACI Fabric instances.

## Virtual Network Device installed on VM Host via Image

In the previous versions, when a virtual network device (such as a virtual ASA) was generated on a VM host via installing network device image, the system treated the virtual network device and host as two devices, and L2 topology generated in this system were not incorrect in this scenario.

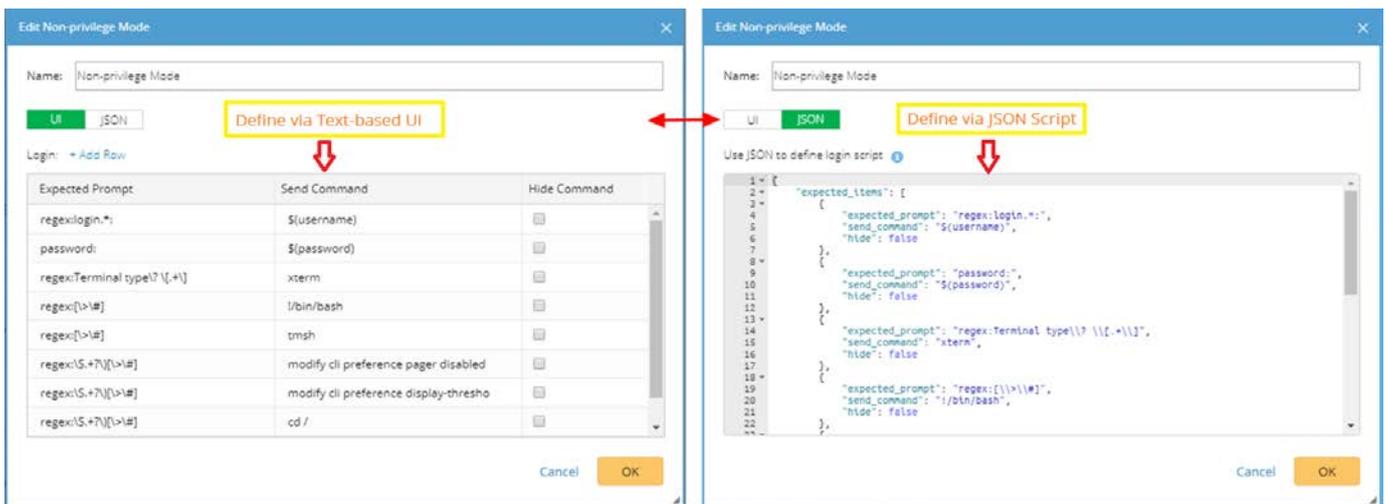
IEv8.0 has improved the topology logic to discover them as one device.

## 3.8. Enhancements to Multi-Vendor Support

### 3.8.1. Enhanced Customization of Login Script and Supported Keys

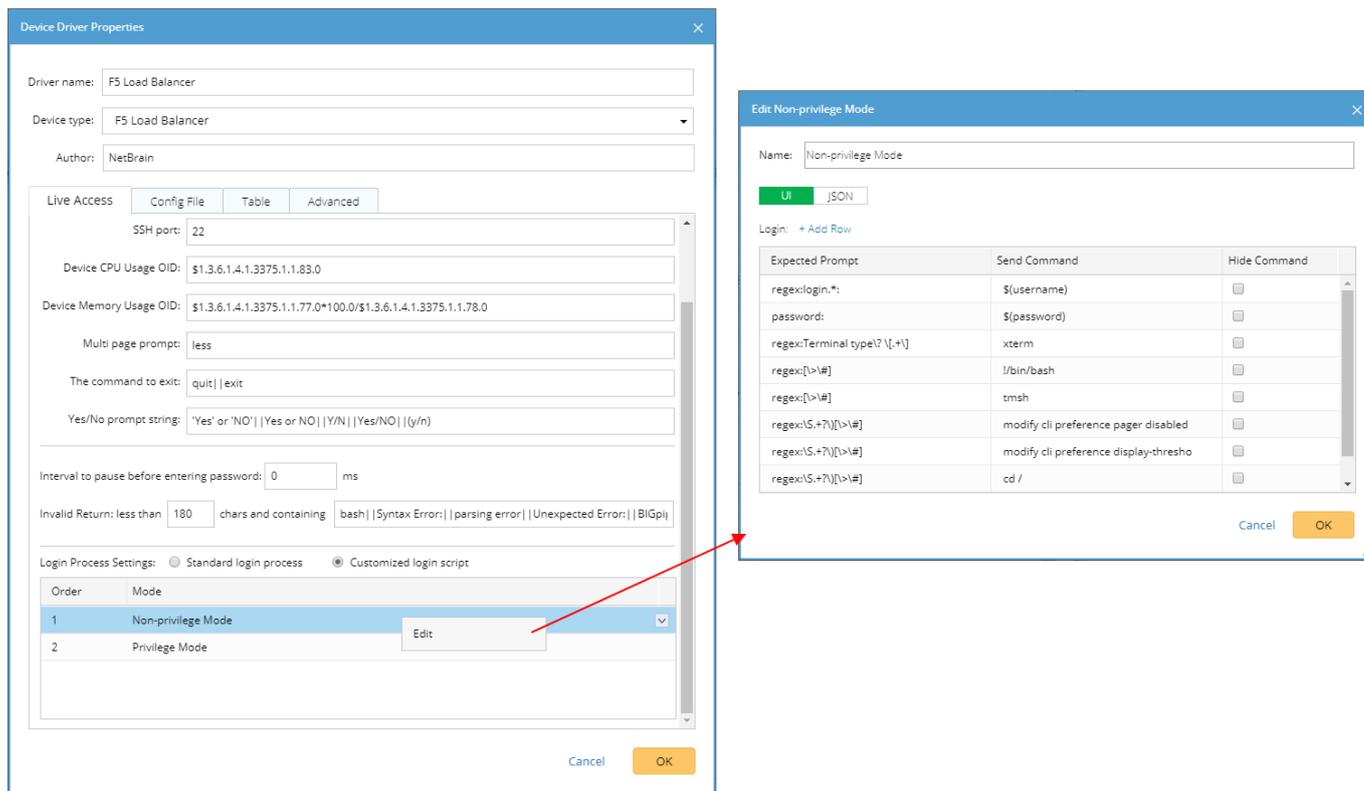
#### Login Script Customization

IEv8.0 enables you to define a customized login script via text-based UI or via JSON scripts.



The content defined on the UI tab and JSON tab is associated and synchronized with each other. When you configure prompts and commands on the UI tab, the corresponding JSON scripts of your configurations are automatically generated on the JSON tab, and vice versa.

**Tip:** You can access login script definition UI in shared device settings or drivers. The login script defined in a driver applies to all device types that use this driver, while that customized in the shared device settings of a device can be applied to a device or device group.



## More Key and Combination Keys

With the continuous development of network technology, network devices are constantly being updated. In order to cope with the more complex login interaction scenarios that NetBrain may encounter in the Multi-vendor Support, IEx8.0 supports more key and combination keys.

No.	Key/Combination Key	Description
1	[Ctrl+Y]	The <b>Ctrl+Y</b> command is sent. Y is a letter in a lower or upper case from A to Z.
2	[Shift+Y]	The <b>Shift+Y</b> command is sent.
3	[Alt+Y]	The <b>Shift+Y</b> command is sent.
4	[Ctrl+Shift+Y]	The <b>Ctrl+Shift+Y</b> command is sent.
5	[Ctrl+Alt+Y]	The <b>Ctrl+ALT+Y</b> command is sent.
6	[Shift+Alt+Y]	The <b>Shift+ALT+Y</b> command is sent.

7	[Ctrl+Y, X]	This command includes pressing the <b>Ctrl+X</b> combination key and then pressing the <b>Y</b> key separately (without holding <b>Ctrl+X</b> ). <ul style="list-style-type: none"> <li>▪ X and Y are both letters in a lower or upper case from A to Z.</li> <li>▪ <b>Ctrl+Y</b>, in this case, can be any type of combination keys from No.1 to No.6, such as [Ctrl+Shift+Y, X].</li> </ul>
8	[Ctrl+Y, Ctrl+X]	The <b>Ctrl +Y+X</b> command is sent. <ul style="list-style-type: none"> <li>▪ <b>Ctrl +Y or Ctrl+X</b>, in this case, can be any type of combination keys from No.1 to No.6, such as [Ctrl+Shift+Y, Ctrl+X].</li> </ul>
9	[Esc]	The <b>Esc</b> command is sent.
10	[Enter]	The <b>Enter</b> command is sent.
11	[Space]	The <b>Space</b> command is sent.
12	[Y]	The <b>Y</b> command is sent.
13	[Ctrl+\]]	The <b>Ctrl+] </b> command is sent.  The system uses [ ] to include a key or combination key, so the escape character \ is used to avoid the conflict of the ] character.

**Tip:** The following two combination keys can be used to customize whether to send an additional Enter key when sending a command.

- **+*r***: send a command with the Enter key. For example, [Ctrl+Y, X] +*r*.
- **-*r***: send a command without the Enter key. For example, [Ctrl+Y] -*r*.

## 3.8.2. More Drivers

IEv8.0 introduces new drivers to support more device types and also improves some existing drivers.

Driver Name	Description
Cisco Web Security Application <sup>New</sup>	Can add configuration files via SNMP only.
Imperva WAF <sup>New</sup>	Can add configuration files and ARP tables.
Secure64 <sup>New</sup>	Can add configuration files via SNMP only.
NetScaler SD-WAN <sup>New</sup>	Can add configuration files and ARP tables.

Driver Name	Description
Checkpoint XOS <sup>New</sup>	Can add configuration files, ARP tables, route tables, and virtual device information.
Cisco ENCS <sup>New</sup>	Can add configuration files, NDP tables, and MAC tables.
AeroHive Switch <sup>New</sup>	Can add configuration files, ARP tables, MAC tables, and route tables.
XetaWave Ethernet Radio <sup>New</sup>	Can add configuration files, ARP tables, and route tables.
AudioCodes Appliance <sup>New</sup>	Can add configuration files, ARP tables, and route tables.
RackSwitch <sup>New</sup>	Can add configuration files, MAC tables, NDP tables, and route tables.
Fortinet FortiGate Firewall	Remove root as virtual node to be discovered.
Cisco IOS Router	Support traceroute for version IOS 16.
Citrix-B Series	Add MAC table and NDP Table.
Cisco WAAS	Add module information.
Alcatel Lucent Server Router	Add NCT tables, including Management ARP Table, Management Route Table, BFD Table, ARP Table[Learning], LDP Table, BGP Neighbor Table, OSPF Neighbor Table.
Viptela	Visualize IPsec tunnel topology for Viptela devices

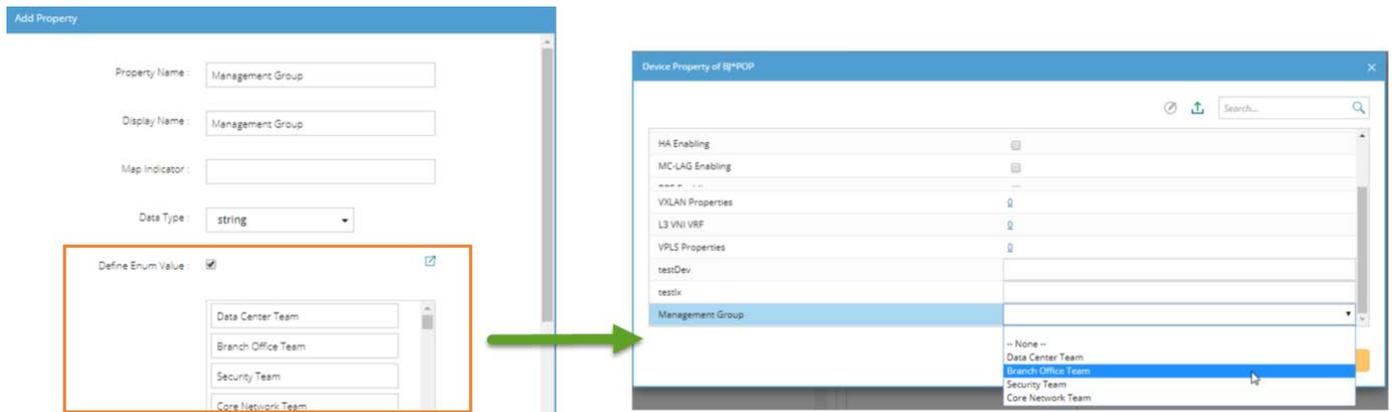
## 3.9. Enhancements to GDR

### 3.9.1. Allow to Predefine Enum Values

In previous versions, a GDR property can only be assigned with value via manual input in the Device Detail Panel. When multiple network engineers fill in the values of these GDR properties for different devices, the same values may be filled in different formats.

IEv8.0 provides the enum value function that enables you to predefine values when defining a single-valued GDR property, such as int, double, and string. The predefined values will be displayed as a list for you to select

when you assign a value for a GDR property in the Device Details Pane.



### 3.9.2. Allow the Deletion of GDR Property Assigned with Value

In previous versions, if a GDR property was filled in with a value, then this GDR cannot be deleted. IEv8.0 allows you to delete a GDR property that has been assigned with a value in the GDR configuration. When the GDR property value is deleted, the corresponding value is also removed from the database.

## 3.10. Parser Enhancements

The Parser feature has been enhanced in IEv8.0 to support the following use cases:

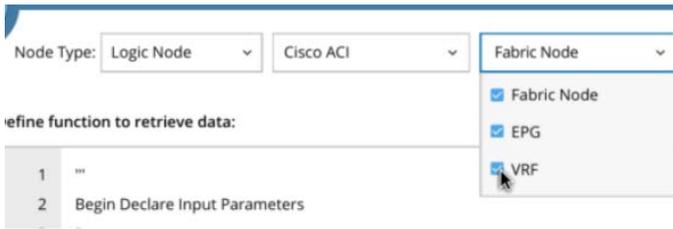
- **Retrieve and Parse Data of Logic Nodes to Enhance SDN Visualization**  
Define a parser for SDN logic nodes, reference the parser in a data view template and then apply it to visualize the data of SDN logic nodes on a map.
- **Allow Users to Set a Key Field for Table Comparison**  
Set a Key for a table-type variable to compare its values at two different periods.

### Retrieve and Parse Data for Logic Nodes to Enhance SDN Visualization

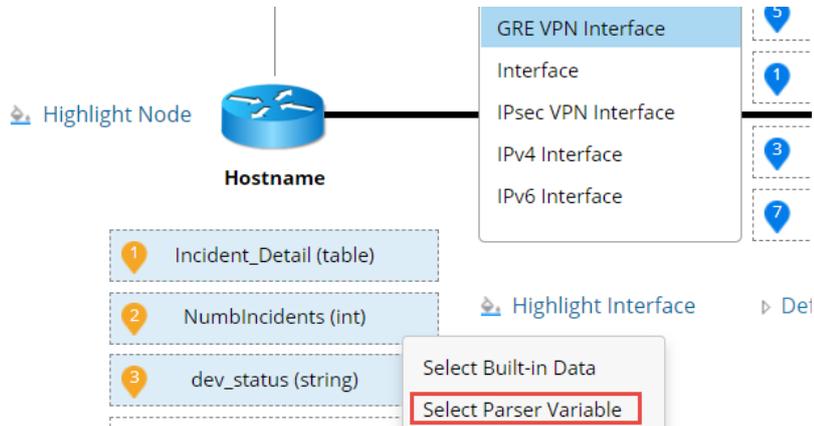
IEv8.0 allows users to define a parser to retrieve the data of any supported SDN logic nodes. The diagram below illustrates the flow:



1. Define an API parser for logic nodes.



2. When defining a Data View Template, reference the API parser.

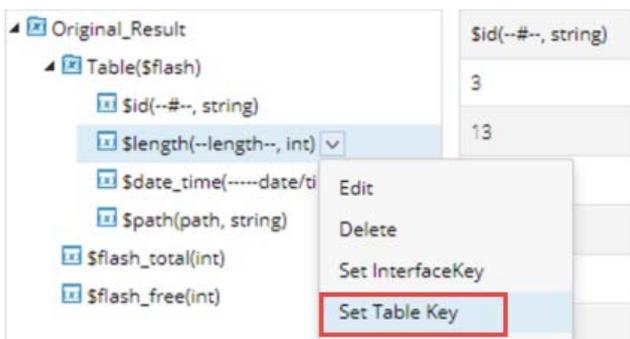


3. Apply the Data View Template to a map with logic nodes, to view the data.

**Note:** API parsers for logic nodes are invisible when building a Qapp or using Instant Qapp, because Qapp can only run on legacy devices.

## Allow Users to Set a Key Field for Table Comparison

The system requires a key to compare table-type parser variables. To better serve Golden Baseline comparison between different time points, IEv8.0 allows users to set a field as the table key to compare two tables.



## 3.11. Qapp Enhancements

### 3.11.1. Enhancements to Qapp Output

To make users more aware of Qapp execution progress and values of various outputs, especially alerts, IEv8.0 shows the elapsed runtime, auto prompts the [Consolidated Output Console](#) and improves the [Email Alerting Mechanism](#).

The screenshot displays the NetBrain interface for configuring and monitoring a Qapp named 'Overall Health Monitor'. On the left, a workflow diagram shows a 'Start' node leading to the 'Overall Health Monitor' node, which has a 'Result 1' output at 10:31 AM. The main panel shows the configuration for 'Overall Health Monitor [SNMP]' with a data source of 'Pull live data regularly', a frequency of 'Every: 10 Seconds', and a repeat count of '10 Times'. Below this is a table of thresholds:

Alert Name	Threshold
High Output Traffic Utilization - Warn...	>= 60
High CPU Utilization - Error	>= 90
High CPU Utilization - Warning	>= 70
High Memory Utilization - Error	>= 80
High Memory Utilization - Warning	>= 60

At the bottom, the 'Elapsed Time:0:2:12' is shown next to a 'Stop' button. A 'Show Output Console' button is also present. On the right, a 'Consolidated Output Console' window is open, showing a summary of outputs: 'Alert (13)' (including 'Interface Down: 13'), 'Data View (1)' (Overall Health Monitor [SNMP]), and 'CSV Report (0)'. The console also shows 'Output Summary', 'Alert', and 'Overall Health Monitor' icons.

### Consolidate Output Console

During Qapp execution, various types of outputs might be generated, depending on Qapp definitions, including Alerts, Data Views, CSV Reports, and Execution Logs. In previous versions, these outputs were scattered in individual panels, and would not prompt for users to explore after Qapp execution. In IEv8.0, the Output Console auto prompts to show available outputs in a single panel, and introduces more UX/UI improves to each output type.

## Show Output Summary

Output Summary is a new tab added to provide a general view of Qapp outputs with statistics.

The screenshot shows the 'Output Summary' tab selected. The top navigation bar includes tabs for 'Output Summary', 'Alert', 'Highlight BGP', 'Overall Health Monitor [SNMP]', 'FRU Report', 'InterfaceErrorCounterReport', and 'Execution Log'. The main content area is divided into two columns. The left column contains sections for 'Alert (3)' (with a red 'X' icon and 'Interface Down: 3'), 'Data View (2)' (with green checkmark icons for 'Highlight BGP' and 'Overall Health Monitor [SNMP]'), and 'CSV Report (2)' (with green document icons for 'FRU Report' and 'InterfaceErrorCounterReport'). The right column is titled 'Execution Log' and lists statistics: 'Unqualified Devices: 2', 'None Parser: 0', 'None Data: 0', 'Printed Messages: 3', 'System Errors: 0', and 'Duration: 0:2:24'.

The following information is highlighted:

- Generated output type and its count.
- How many outputs each output type has.
- Key execution information, including devices' qualifications, variable mapping, etc., to provide a basic analysis for further troubleshooting.

## Show Alert Outputs and Counting Statistics

Alerts triggered at both device and interface levels are shown in the Alert Output tab. To eliminate duplicate alert items, the Count column is added in IEv8.0.

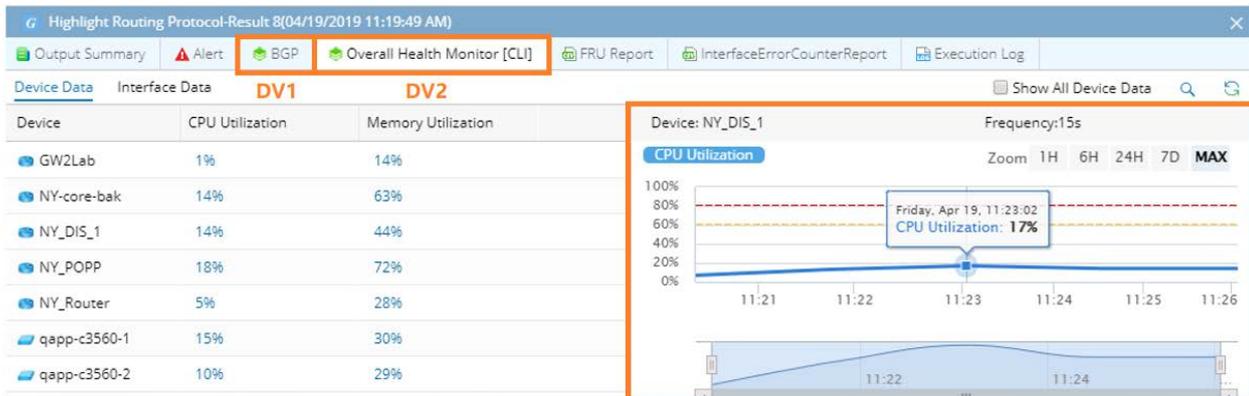
The screenshot shows the 'Alert' tab selected. The top navigation bar includes tabs for 'Output Summary', 'Alert', 'BGP', 'Overall Health Monitor [CLI]', 'FRU Report', 'InterfaceErrorCounterReport', and 'Execution Log'. Below the navigation bar, there are filters for 'Alert Level: All Alerts' and 'Device: All Devices'. The main content area is a table with the following data:

Device	Message	Last Time	Count
NY_DIS_1.Ethernet0/1	✖ Input Errors: 54	4/19/2019, 12:32:01 PM	34
NY-core-bak	⚠ Memory utilization is 64% >= 60%.	4/19/2019, 12:32:00 PM	35
NY_POPP	⚠ Memory utilization is 72% >= 60%.	4/19/2019, 12:31:59 PM	35
NY_POPP.Ethernet0/0	✖ Input Errors: 1	4/19/2019, 11:41:45 AM	1

For more enhancements about Qapp alert emails, refer to [Enhanced Email Alerting Mechanism](#).

## Show Data View Outputs

In IEv8.0, each Data View output has its individual tab to show detailed information, including metrics of devices and interfaces. For numerical metrics, users can view their historical values and trend in a chart.



## Show CSV Report Outputs

In IEv8.0, each CSV report has its individual tab to show detailed information. For recurring Qapps, users can further locate the exact report instance via Qapp execution time.

The screenshot displays the 'Highlight Routing Protocol-Result 11' window. The 'FRU Report' tab is active. A table shows report instances for Report1 and Report2. The table includes columns for Device, Execution Time, Description, PID, VID, SN, Location, and Management IP.

Device	Execution Time	Description	PID	VID	SN	Location	Management IP
GW2Lab	04/19/2019 12:33:23 PM	CISCO3945-CHASSIS	CISCO3945-CHASSIS	V02	FGL183810W1		10.10.7.253
GW2Lab	04/19/2019 12:31:48 PM	Cisco Services Perf...	C3900-SPE250/K9	V05	FOC183584XM		10.10.7.253
GW2Lab	04/19/2019 12:28:56 PM	C3900 AC Power Su...	PWR-3900-AC	V04	QCS18261DX3		10.10.7.253
NY_Router	04/19/2019 12:27:37 PM	2811 chassis	CISCO2811	V08	FHK1423F1YS		172.24.30.2
NY_Router	04/19/2019 12:26:09 PM	Port FE Switch	HWIC-4ESW	V01	FOC12021ABY		172.24.30.2
qapp-c3560-2	04/19/2019 12:24:36 PM	WS-C3560G-24TS	WS-C3560G-24TS-S	V03	FOC1241Y01S		172.27.0.2
NY_DIS_1	04/19/2019 12:23:16 PM	2611 chassis	2611 chassis, Hw S...	0x202	JAB024704TX (3591...		172.24.31.66

## Show Detailed Execution Log

IEv8.0 makes improvements on Qapp execution logs:

The screenshot displays the 'Monitor Interface Error Counters-Result 1' window. The 'Execution Log' tab is active. A dropdown menu for 'Log Type' is open, showing options: All, Unqualified Devices, None parser, None Data, Printed Messages, and System Errors. The log entries show CLI timeout errors.

```
1 4:3 All c_sw1 + Interface [Cisco IOS] + 2019-04-29 16:34:08".CLI timed out due to mismatched prompt.Error Code 131000010.
2 4:3 Unqualified Devices c_sw1 + Interface [Cisco IOS] + 2019-04-29 16:34:48".CLI timed out due to mismatched prompt.Error Code 131000010.
3 4:3 None parser c_sw1 + Interface [Cisco IOS] + 2019-04-29 16:35:28".CLI timed out due to mismatched prompt.Error Code 131000010.
4 4:3 None Data c_sw1 + Interface [Cisco IOS] + 2019-04-29 16:36:08".CLI timed out due to mismatched prompt.Error Code 131000010.
```

- Show a more detailed description with error codes for debugging.
- Add a few filters to narrow down log entries by log type and device.

**Tip:** To auto clean Qapp execution logs, refer to [Auto Clean Execution Logs and More Data](#).

## Enhanced Email Alerting Mechanism

IEv8.0 unifies and improves the Qapp email alerting mechanism to allow users to customize their email alerting rules based on specific needs.

### Alert Email Composition

IEv8.0 allows domain admin to select among three email compositions to meet their specific needs. For example, two devices get an alert separately in two tasks. Check out the following sample emails to see differences among three compositions:

#### 1. Merge All Alerts in One Email New and Default

- Sample Email:

Object	Severity	Message	From Task	User	Time
NY_POPP	Warning	Memory utilization is 72% >= 60%.	Map5.Page 1.Result 2.Overall Health Monitor [CLI]	weicai	2019-04-19 12:50:56 +08:00
NY_POPP	Warning	Memory utilization is 72% >= 60%.	Map6.Page 1.Result 1.Overall Health Monitor [CLI]	weicai	2019-04-19 12:21:45 +08:00
NY-core-bak	Warning	Memory utilization is 64% >= 60%.	Map5.Page 1.Result 2.Overall Health Monitor [CLI]	weicai	2019-04-19 12:50:57 +08:00
NY-core-bak	Warning	Memory utilization is 64% >= 60%.	Map6.Page 1.Result 1.Overall Health Monitor [CLI]	weicai	2019-04-19 12:21:47 +08:00

#### 2. Separate Alert Emails for Different tasks New

- Sample Email1 for Task1:

Object	Severity	Message	From Task	User	Time
NY_POPP	Warning	Memory utilization is 72% >= 60%.	Map5.Page 1.Result 2.Overall Health Monitor [CLI]	weicai	2019-04-19 12:55:56 +08:00
NY-core-bak	Warning	Memory utilization is 64% >= 60%.	Map5.Page 1.Result 2.Overall Health Monitor [CLI]	weicai	2019-04-19 12:56:02 +08:00

- Sample Email2 for Task2:

Object	Severity	Message	From Task	User	Time
NY_POPP	Warning	Memory utilization is 72% >= 60%.	Map6.Page 1.Result 1.Overall Health Monitor [CLI]	weicai	2019-04-19 11:56:30 +08:00
NY-core-bak	Warning	Memory utilization is 64% >= 60%.	Map6.Page 1.Result 1.Overall Health Monitor [CLI]	weicai	2019-04-19 11:56:32 +08:00

#### 3. Separate Alert Emails for Different Devices (or paths...)

- Sample Email1 for Device1:

Object	Severity	Message	From Task	User	Time
NY_POPP	Warning	Memory utilization is 72% >= 60%.	Map5.Page 1.Result 2.Overall Health Monitor [CLI]	weicai	2019-04-19 12:53:26 +08:00

- Sample Email2 for Device2:

Object	Severity	Message	From Task	User	Time
NY-core-bak	Warning	Memory utilization is 64% >= 60%.	Map5.Page 1.Result 2.Overall Health Monitor [CLI]	weicai	2019-04-19 12:53:27 +08:00

To change the alert email composition, go to **Domain Management > Advanced Settings > Configure Alert Email for Qapp**.

### Configure a Delta Threshold to Avoid Excessive Emails

To avoid excessive emails, IEv8.0 adds a checkbox to allow users to configure a delta threshold: only when an alert is repetitively recurring and increasingly reaches the value, the system will email the alert.

By default, the value is 60. To configure it, go to **Domain Management > Advanced Settings > Configure Alert Email for Qapp**.

## 3.11.2. Enhancements to Qapp Debugging

When being executed with Current Baseline, a Qapp actually uses the parsed data in the Current Baseline (data generated the scheduled parser/DVT task) to analyze results, instead of the raw data of the Current Baseline. It brings a problem that NetBrain engineers cannot use the raw data provided by users to debug a Qapp beyond the users' domains.

To resolve this problem, IEv8.0 introduces a function that allows running a Qapp with raw data directly for the debugging purpose.

Settings

Share Alert with: *Click to select username or email address.*

Send Email to: *Separate multiple addresses with a semicolon.*

Send Email for:  Error  Warning

Run Mode:  Production mode *Only show error and warning message.*  
 Debug mode *Show error,warning,information,and system message.*

Run Qapp with raw data on Current Baseline

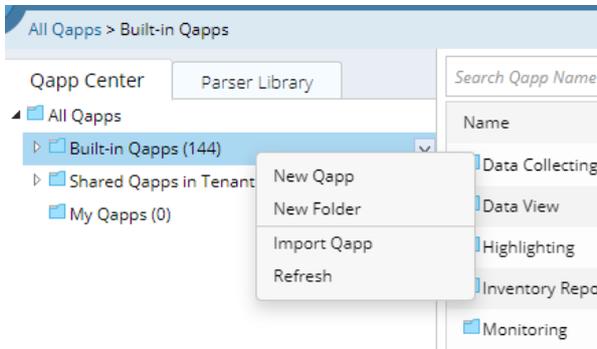
Cancel OK

**Note:** This option is available only when the data source is **Current Baseline** and the **Debug Mode** is enabled.

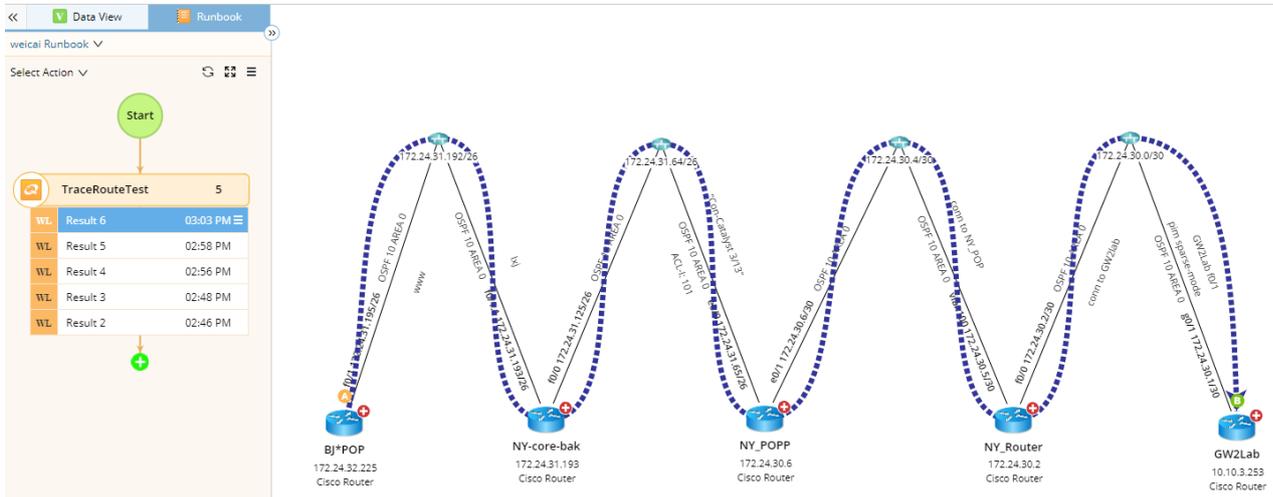
It is not recommended for customers to use this function, because it is designed for NetBrain Engineers to debug a Qapp for customers. Variable Mapping will not be applied when the option is checked.

### 3.11.3. Usability Enhancements

- Use the shortcut key **CTRL+ S** to save a Qapp/Gapp/Parser file.
- Add a right-click menu for folders in the Parser Library, Qapp Center and Gapp Center.



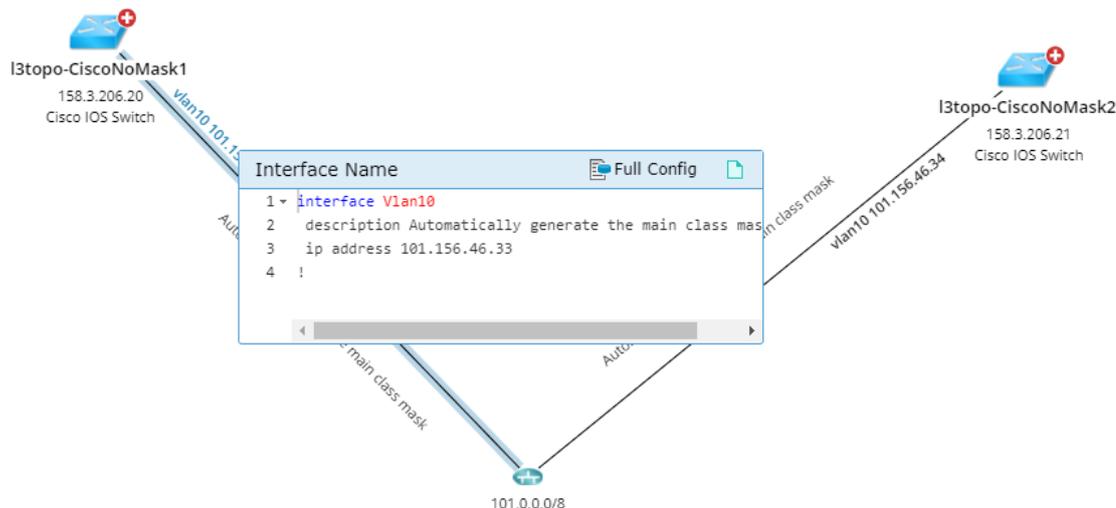
- Add a **DrawTraceRouteHops()** API in the script output to draw the traceroute result hop by hop on a map.



### 3.12. L3 Topology Support for Interfaces without Masks

IPv8.0 supports calculating L3 Topology for interfaces without masks. For example, the interface IPs of Meraki devices retrieved via APIs do not have masks. The system will try to find the LAN where the interface belongs to and use the mask of the LAN to calculate the topology for the interface. If the LAN for the interface cannot be found, the system will use the corresponding mask of the IP main class of the interface IP to calculate the

topology.



To enable this function, go to **Domain Management > Advanced Settings > Build L3 Topo Option** and check the option **Use the main class mask to calculate L3 topology for an IP without mask**.

## 3.13. L2 Topology Improvements

### 3.13.1. Auto Clean Unknown End Systems

For a network using DHCP, an IP address may be assigned to different PCs at different times. This causes multiple unknown end systems with the same IP address, but different MAC addresses existed in the domain. When the system calculates a path across this kind of unknown end system, the path will fail due to duplicated MAC entries.

To resolve this issue, IeV8.0 automatically cleans outdated unknown end systems when a One-IP table entry is deleted via either of the following methods

- When a One-IP table entry older than the specified days (auto clean) is automatically deleted from the system, the unknown end systems generated by the One-IP table entry will be deleted.

Data Type	Data Size	Auto - Clean Rule		
Qapp/Gapp Execution Logs	N/A	<input checked="" type="checkbox"/> Delete data older than	14	days
One-IP Table Entries	4KB	<input type="checkbox"/> Delete data older than	14	days

- When One-IP table entries are deleted at L2 topology rebuilding after enabling the DHCP Filter function, the unknown end systems generated by the One-IP table entries will be deleted.

### Build L2 Topology Option

Minimum subnet mask:

Filter DHCP Entries

Only save One-IP table entries that have values in Switch Port or DNS Name parameter

**Tip:** When the DHCP filter is enabled, the system filters out the outdated DHCP entries and only keeps the latest IP addresses when building build L2 topology.

## 3.14. Enhancements to APIs

### 3.14.1. North-bound APIs

Since IEv8.0, all north-bound APIs have been published at [GitHub](#), bringing the following changes:

- 18 New APIs:
  - [Get Device Data](#)
  - [Add Discovery Task](#)
  - [Edit Discovery Task](#)
  - [Delete Discovery Task](#)
  - [Get Event Console](#)
  - [Acknowledge Event Alert](#)
  - [Delete Event Alert](#)
  - [Get Device Access Policies](#)
  - [Add Device Access Policy](#)
  - [Update Device Access Policy](#)
  - [Delete Device Access Policy](#)
  - [Get User Device Access Policy](#)
  - [Set User Device Access Policy](#)
  - [Assign Device Access Policies to User](#)
  - [Remove Device Access Policies from User](#)
  - [Get Users of Device Access Policy](#)
  - [Get License Node Info](#)

- [Get Production Info](#)
- [Schedule Change Management Task](#)
- [Get a Scheduled CM Task](#)
- [Update a Scheduled CM Task](#)
- [Delete a Scheduled CM Task](#)
- 2 Enhanced APIs:
  - [Get Device and Device Attributes](#)
  - [Get Device Neighbors with Topology Type](#)
  - [Get One-IP Table](#)
  - [Get path calculation result](#)
  - [API-triggered diagnosis](#): add more input parameters to support more functions, such as mapping a path, using Qapp to generate a map, supporting DVT and Application Verification node, and so on.

### 3.14.2. System APIs

IEv8.0 introduces 2 new system APIs, which can be called in Plugin, Qapp or other modules in the system.

System API	Description
<code>addDevicesToSite(sitename,devicelist)</code>	Add devices to a site. The API can be called in the Qapp/plugin/path feature.
<code>DrawTraceroute()</code>	Draw the result of a traceroute operation hop by hop on a map. This API can be called in the Qapp feature.
<code>ExcuteSharedTuneDevice()</code> <code>ExcuteTuneOneDevice()</code> <code>ExcutePrivateTuneDevices()</code> <code>ExcutePrivateTuneOneDevice()</code> <code>GetTuneDeviceResultByDevIds()</code> <code>GetPrivateTuneDeviceResultByDevIds()</code>	Tune devices and report tune results.
<code>GetHostnameChangeList()</code> <code>UpsertHostnameChangeList()</code> <code>KeepLastChangeDevice()</code> <code>SubmitHostNameChangeChildTask()</code>	Detect hostname change and keep the latest hostnames.

**Tip:** For more APIs in the system, click this [link](#).



## 4. Admin & MISC

### 4.1. New Subscription License Model with Modularization

The following changes have been introduced to IEv8.0 licensing functionalities:

1. Transform from Perpetual License Model to [Subscription License Model](#)
2. [Product Modularization](#) based on different feature functionalities and value propositions
3. [License Adjustment to SDN Module](#) by using new conversion ratios

#### 4.1.1. Subscription License Model

In IEv8.0, NetBrain is moving from the perpetual license model toward the subscription license model to achieve high revenue predictability, a more loyal customer base and an opportunity to upsell its value-added services. Successful implementation of the subscription license model will provide a strong impetus for NetBrain's continuous growth in the future.

NetBrain subscription license includes multiple terms. Each term is defined license parameters including start date, end date, node, seat, etc. After the customer purchases the subscription license, a new term will be created in NetBrain License Server.

As the license expiry date (defined by term parameter- end date) approaches, customers will be notified to renew their license (by adding new terms). To further improve the license renewal experience, NetBrain also offers a grace period to the customers after the license expires, allowing them to continue using the software within a certain amount of time.

#### User Flow

1. New Customers: [Purchase, Activate & Renew License](#)
2. Existing Customers: [Migrate Perpetual License to Subscription License](#)
3. [Unbind License](#)

#### Purchase, Activate & Renew License

For new customers, the new subscription license model provides the 'purchase-activate-renew' three-step flow to maintain their subscription licenses:

1. [Purchase License](#)
2. [Activate License](#)
3. [Renew License](#)

## Purchase License

When a purchase order is submitted to NetBrain with all negotiated licensing details, such as the network size and the number of concurrent users, the Order Fulfillment Team will log into License Server to define the subscription license for the customer.

The workflow of defining a subscription license by Order Fulfillment Team:

1. Define Contact Information and Delivery Information
2. Define Order Name and Payment Information
3. Define Parameters of Essential Module
4. Define Parameters of AAM/CM/SDN Modules

Refer to [Product Modularization](#) for more information about module license features.

## Activate License

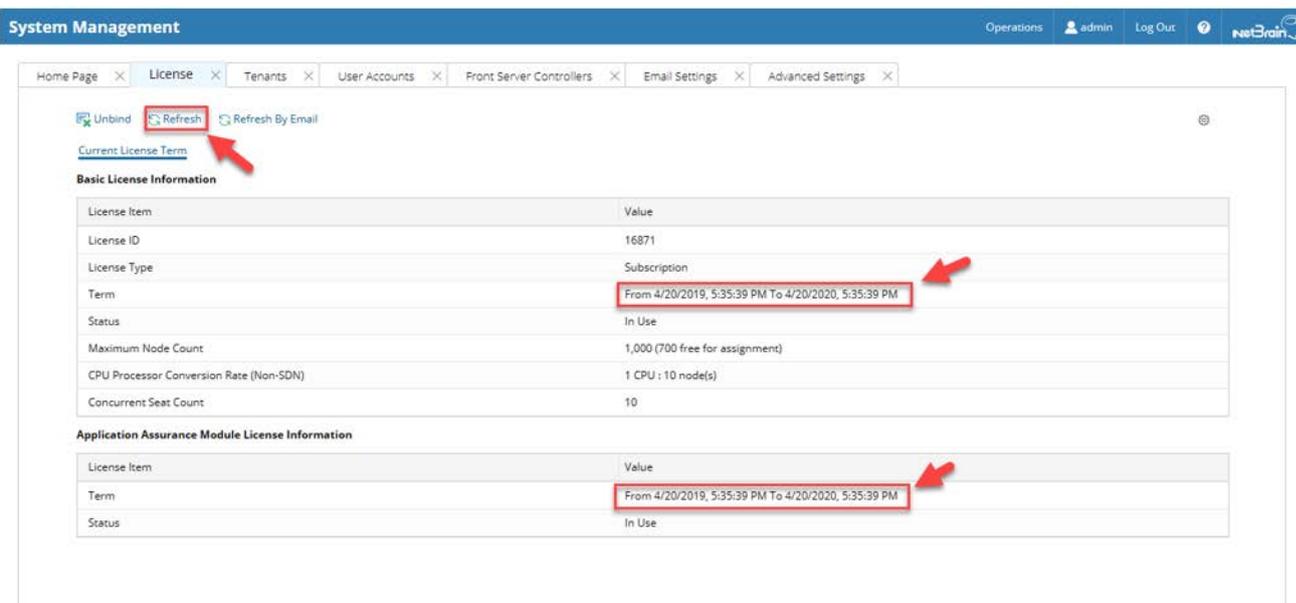
When customers complete the NetBrain installation process, they will need to log in to the System Management page with the license ID and activation key to activate the software license. To accommodate different network conditions, NetBrain supports both online activation and offline activation (by email).

## Renew License

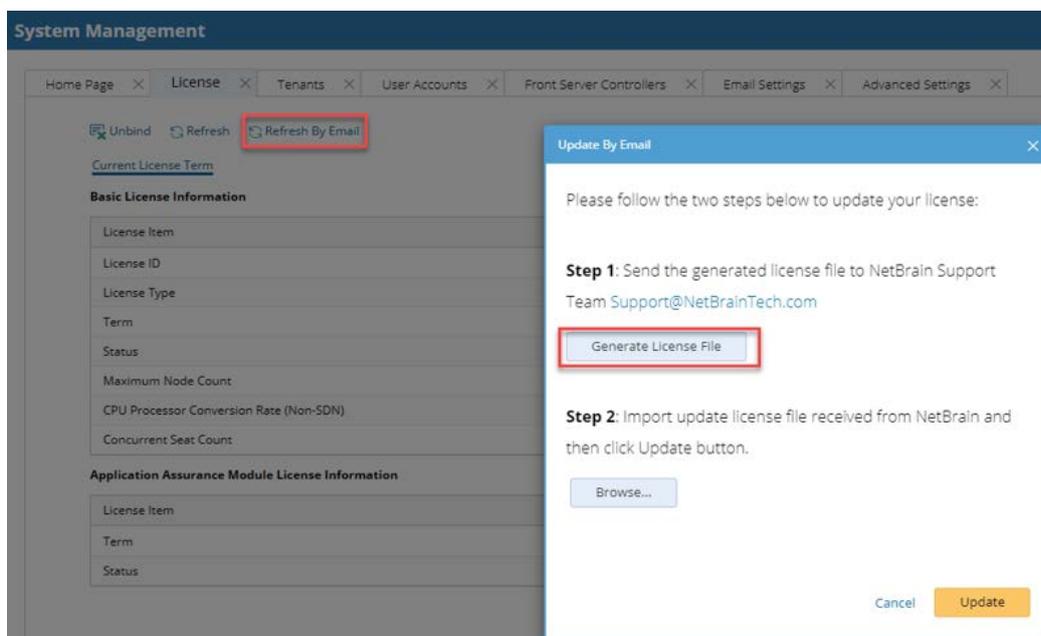
User needs to renew their subscription license when the license expiry date is approaching to avoid account suspension or deactivation of certain advanced features. NetBrain IE will keep the end user well-informed during the entire license renewal cycle.

If the user's license is about to expire or already expired, the corresponding notification will be displayed when the user logs into the IE end user interface. If the user intends to renew the license, he/she can contact the sales staff to negotiate the renewal details.

1. **Online Refresh License (Recommend)**



## 2. Offline Refresh License



### Notes:

- Essential module, Application Assurance Module, Change Management Module and SDN Module each has its own license parameters including start date, end date, expiry notification period and grace period settings.
- If the subscription license (Essential Module) expires (and is beyond the grace period), end user will not be able to log into the IE end user system. If the feature license (Assurance Module/Change Management Module/SDN Module) expires (and is beyond the grace period), the relevant features will be disabled, but the relevant user data will not be deleted.
- If the node count (purchased with new subscription term) is fewer than the number of devices being added to the IE system, end user will not be able to log into the IE end user system. In this scenario, NetBrain Support Engineer

can temporarily increase the customer's license node count, deleting the unused devices from the customer's IE system, and restore the node count.

## Migrate Perpetual License to Subscription License

For users who wish to upgrade the system from IEv7.x, they'll need to migrate the IEv7.x perpetual license to the IEv8.0 subscription license as follows:

1. **Migrate Perpetual License to Subscription License** (by Order Fulfillment Team)

Order Fulfillment team will complete the license migration on NetBrain license server as follows:

- a. Change the license version to 8.0 in the perpetual license editing page.
- b. Define subscription license parameters in the new page.

2. **Upgrade to 8.0 and Refresh License** (by Admin User)

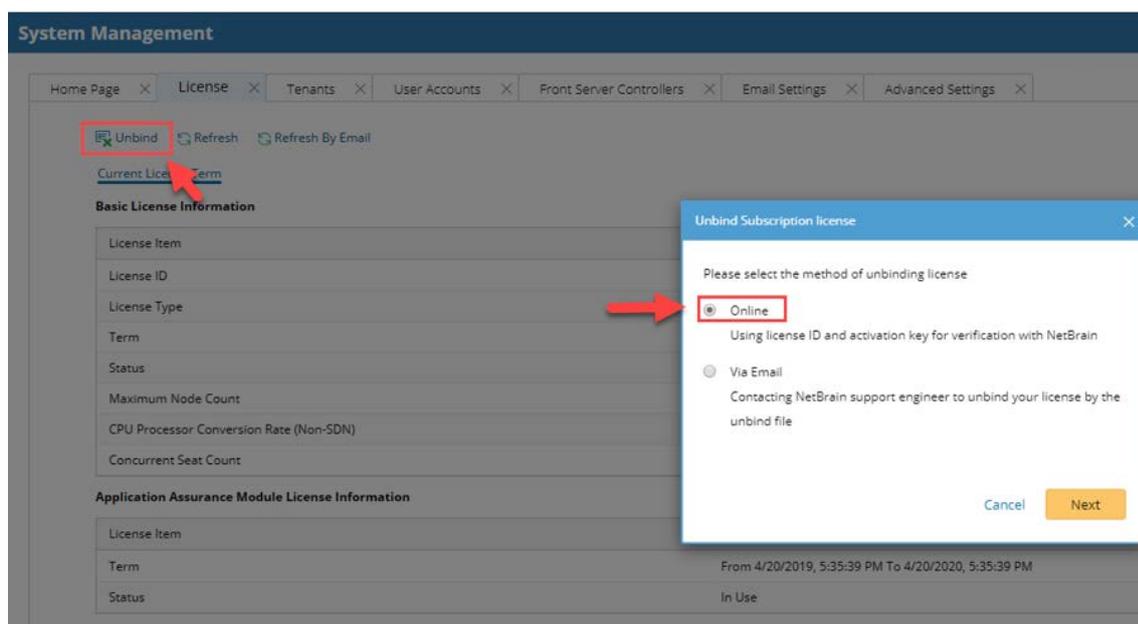
Admin user needs to refresh the license (by navigating to **System Management > License**) to complete the migration from a perpetual license to a subscription license.

## Unbind License

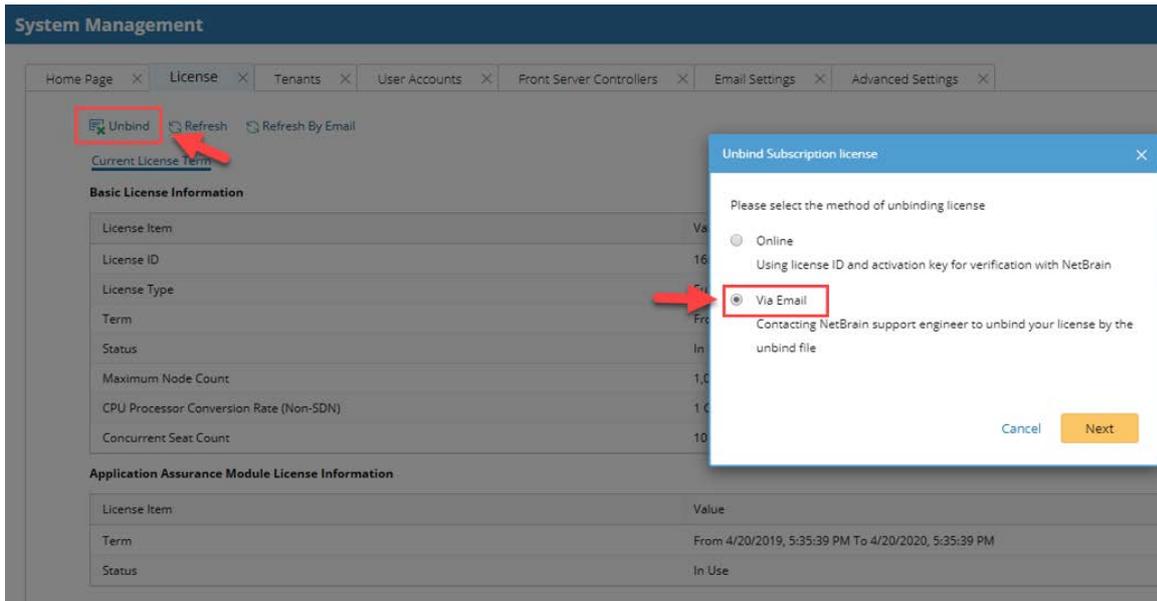
To migrate the NetBrain license to a new server, users need to unbind the license on the old server before reactivating the license on the new server. In IEv8.0, the unbinding flow has been optimized from the perspective of system security:

1. Unbind License (by Admin User/NetBrain Support Engineer)

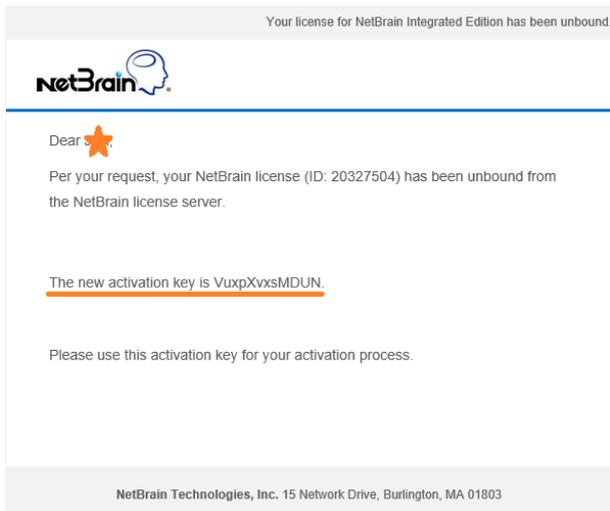
- a. **Online Unbind (Recommended)**



b. *Offline Unbind*



2. Receive New Activation Key (by Admin User).



3. [Activate License](#) (by Admin User/NB Support).

## 4.1.2. Modularized Product

NetBrain I Ev8.0 includes the following four modules:

**Essential Module** is intended to provide customers with the core product features (excludes the value-added features offered in other advanced modules) to enable the solution of network visibility and automation.

**Application Assurance Module (AAM)** is intended for customers who mainly focus on application-aware path management. It provides an integrated solution to manage, monitor and troubleshoot network applications

by combining the application-aware paths with NetBrain automation features. Value-added features of AAM include:

- Application Manager
- Golden Path
- Application-Aware Monitor

**Change Management Module (CMM)** is intended to provide customers with a comprehensive network change management solution including the ability to create, approve, deploy and validate complex network changes.

Value-added features of CMM include:

- Change Management Runbook
- Schedule and Triggered Network Change

**SDN Module** is intended to provide customers with advanced capability to monitor, troubleshoot, and visualize the SDN network via NetBrain dynamic map. Value-added features of SDN module include:

- Cisco ACI
- VMware NSX-V

**Note:** For more details about the license parameter definition of each module, refer to [SDN Module License Enhancements](#).

**Note:** Triggered Diagnosis (sub-feature previously offered in IE7.x Machine-driven Task Module) will be merged into the Essential Module in IEv8.0.

---

### 4.1.3. License Adjustment to SDN Module

IEv8.0 introduces the following two enhancements to the SDN module license:

- Unify SDN Licensed Node Count
- Dedicated SDN Module License for SDN Technologies

These enhancements will not only tremendously reduce the manual workload of Support Engineers when processing migration from a legacy network to an SDN network for customers, but also ease their learning curve of understanding the NetBrain software.

## Unify SDN Licensed Node Count

In IEv8.0, the Licensed Node Count becomes the single metric to measure the quantities for both legacy devices and SDN nodes hosted by NetBrain system. CPU Process Count and Port Count, as a result, will no longer be used for SDN nodes.

The following two parameters have been introduced in IEv8.0 to enable the unification of SDN node count:

- **CPU Processor Conversion Rate** is the ratio used to convert the number of CPU consumptions to the number of node consumptions. In IEv8.0, CPU Processor Conversion Rate (SDN) is provided for the usage of NSX node calculation.
- **Port Conversion Rate** is the ratio used to convert the number of port consumptions to the number of node consumptions. In IEv8.0, Port Conversion Rate (SDN) is provided for the usage of Cisco ACI node calculation.

**Example:** SDN Conversion Rate definition in License Server (SDN) interface

The screenshot displays the NetBrain License Server interface. The top navigation bar includes links for Orders, Users, Products, Trial Users, Coupons, Licenses, Request Enterprise Edition List, Log Info, Activation, Update, Unbind, Management, Advertisement, Download Visit Info, My Profile, and Logout. The main interface is divided into two panels: 'Basic Information' on the left and 'Feature' on the right. The 'Feature' panel is further divided into tabs: Essential, Administration, Application Assurance, SDN (highlighted), and Change Management. The 'SDN' tab contains the following configuration fields:

- Order Name: 4/20/2016 - 4/20/2022 (Secondary Year)
- Status: ON
- Term (Boston Time): From 04/20/2019 17:35:39 To 04/20/2020 17:35:39
- Grace Period: 60 (Allow user to continue to use in xxx day(s) when license expired)
- Expiry Notification Period: 30 (Notice user the license will expire before xxx day(s))
- Activate Time: [Empty field]
- Change Time: 2019-04-24T17:03:25
- CPU Conversion Rate: 1 CPUs Convert to 10.0 Nodes
- Port Conversion Rate: 1 Ports Convert to 0.5 Nodes
- Payment Info: Contract
- End Date of Contract: [Empty field]
- Description: [Empty text area]

At the bottom right of the interface, there are 'Save' and 'Go Back' buttons.

**Example:** SDN Conversion Rate definition in System Management (SDN Module License) interface

The screenshot shows the 'System Management' interface with the 'License' tab selected. The interface includes a navigation bar with 'Operations', 'admin', 'Log Out', and 'NetBrain' logo. Below the navigation bar, there are tabs for 'Home Page', 'License', 'Tenants', 'User Accounts', 'Front Server Controllers', 'Email Settings', and 'Advanced Settings'. The main content area displays 'Current License Term' and 'Basic License Information'.

**Basic License Information**

License Item	Value
License ID	16871
License Type	Subscription
Term	From 4/20/2019, 5:35:39 PM To 4/20/2020, 5:35:39 PM
Status	In Use
Maximum Node Count	1,000 (700 free for assignment)
CPU Processor Conversion Rate (Non-SDN)	1 CPU : 10 node(s)
Concurrent Seat Count	10

**SDN Module License Information** (highlighted with a red box)

License Item	Value
Term	From 4/20/2019, 5:35:39 PM To 4/20/2020, 5:35:39 PM
Status	In Use
Port Conversion Rate (SDN)	1 port : 0.5 node(s)
CPU Processor Conversion Rate (SDN)	1 CPU : 10 node(s)

**Application Assurance Module License Information**

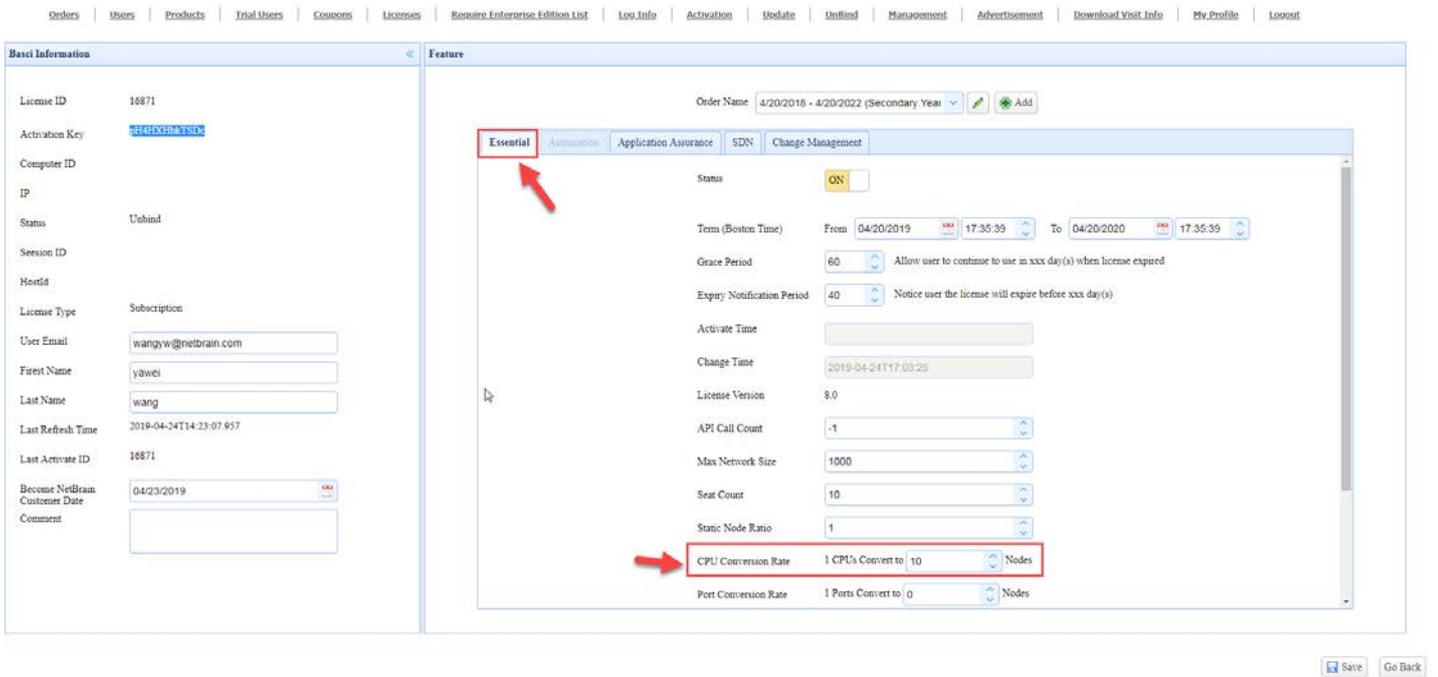
License Item	Value
Term	From 4/20/2019, 5:35:39 PM To 4/20/2020, 5:35:39 PM
Status	In Use

## Dedicated SDN Module License for SDN Technologies

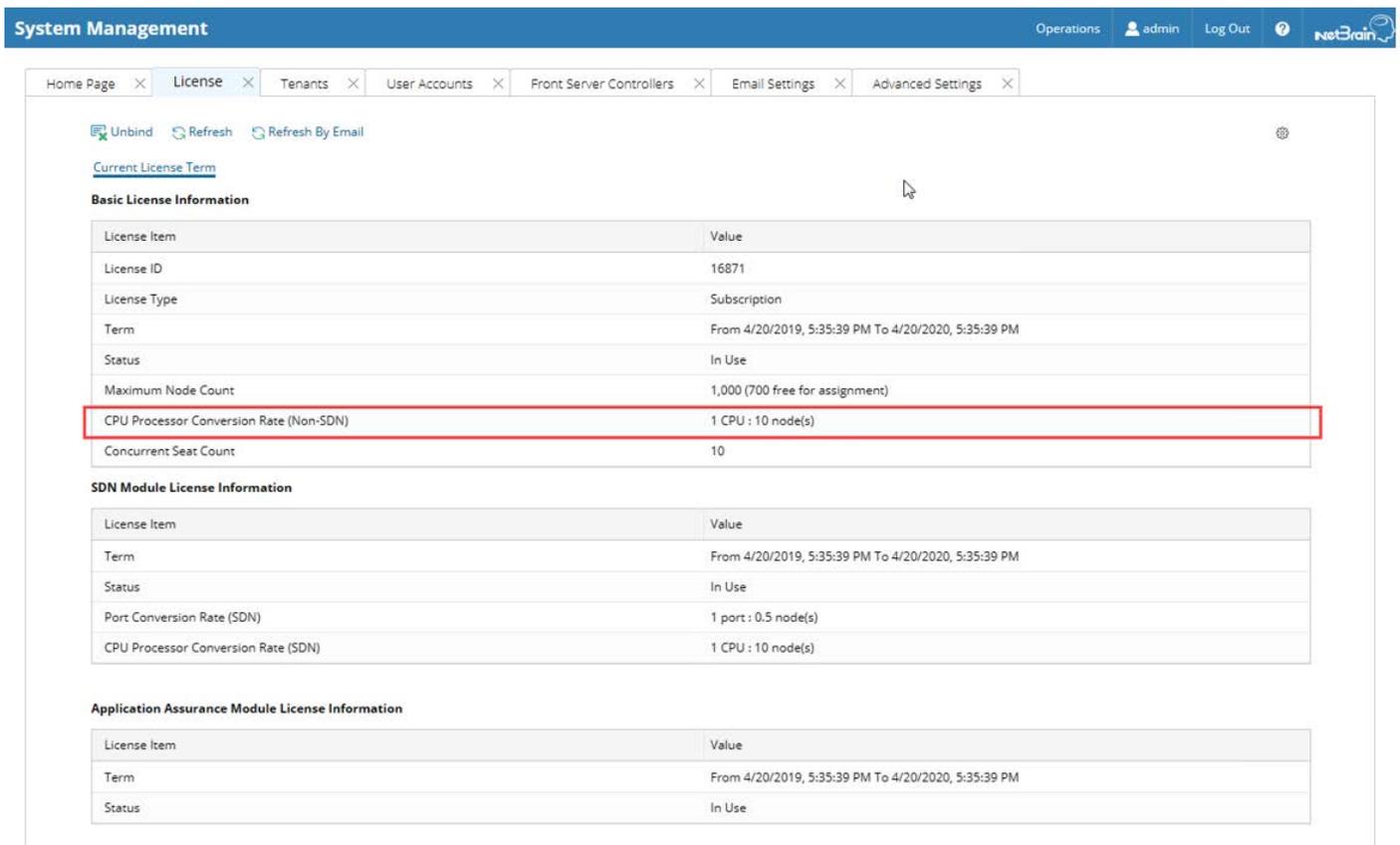
In IEV8.0, only SDN technologies, such as VMware NSX-V and Cisco ACI, will require SDN licensed nodes. Tech Spec technologies including Cisco Meraki, VMware vCenter (ESXi) and Checkpoint R80 will be fully supported by Essential Module, which means no extra fee is required for SDN Module.

**Example:** CPU Processor Conversion Rate definition in License Server (Essential) interface

Similar to NSX-V, CPU Processor Conversion Rate is used in vCenter node calculation (Host Virtualization) to convert the number of CPU consumptions to the number of node consumptions.



**Example:** CPU Processor Conversion Rate definition in System Management (Basic License) interface



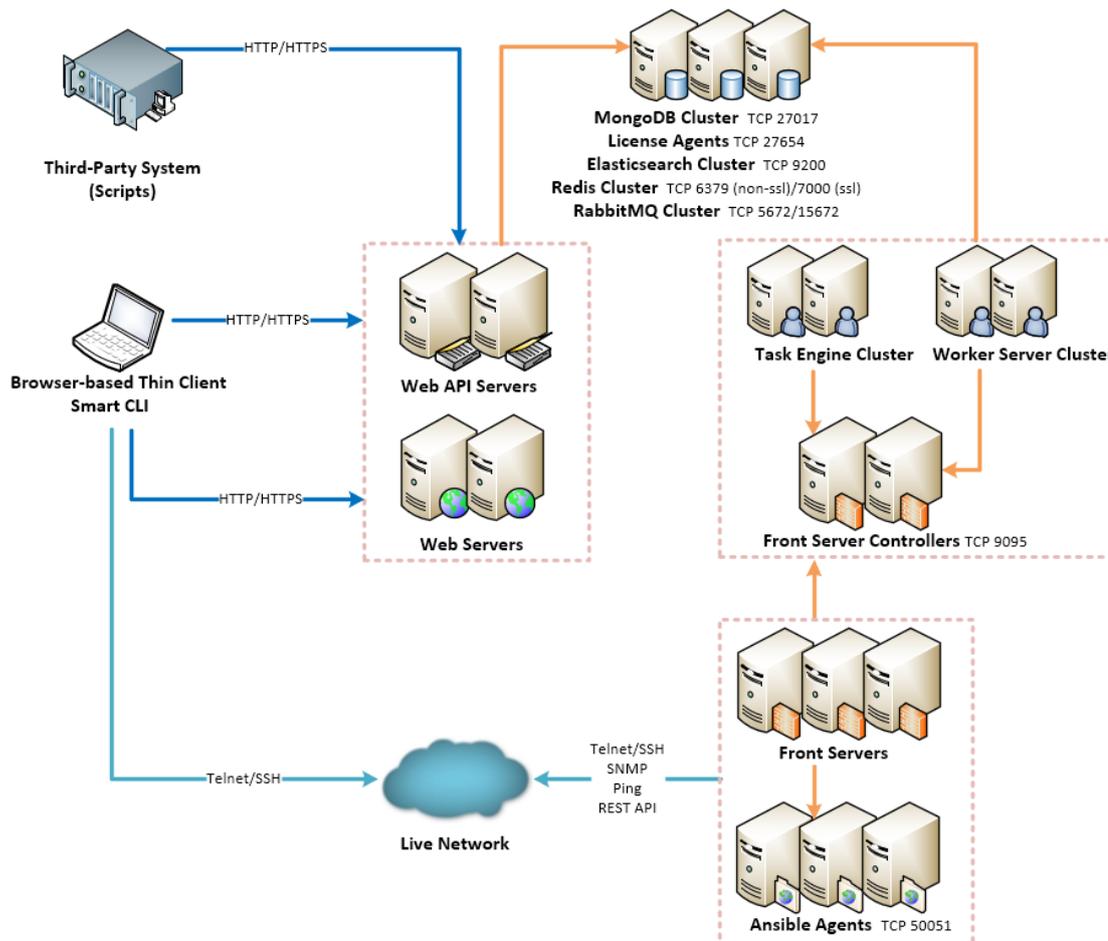
**Note:** In an ESXi/NSX hybrid environment, to avoid repetitive resource calculation, NetBrain will only use the NSX CPU Processor consumption to calculate the license usage.

## 4.2. System Architecture Enhancements

### 4.2.1. System Architecture Summary

The system components include both third-party components and NetBrain components.

#### V8.0 Distributed Deployment Architecture (HA)



**Note:** Knowledge Cloud Agent <sup>New</sup> (not illustrated in the above diagram) is installed with Web API Server, serving as a proxy to synchronize NetBrain IE resources from NetBrain Knowledge Cloud, including Qapp, Data View Templates, Runbook Templates, Drivers, etc.

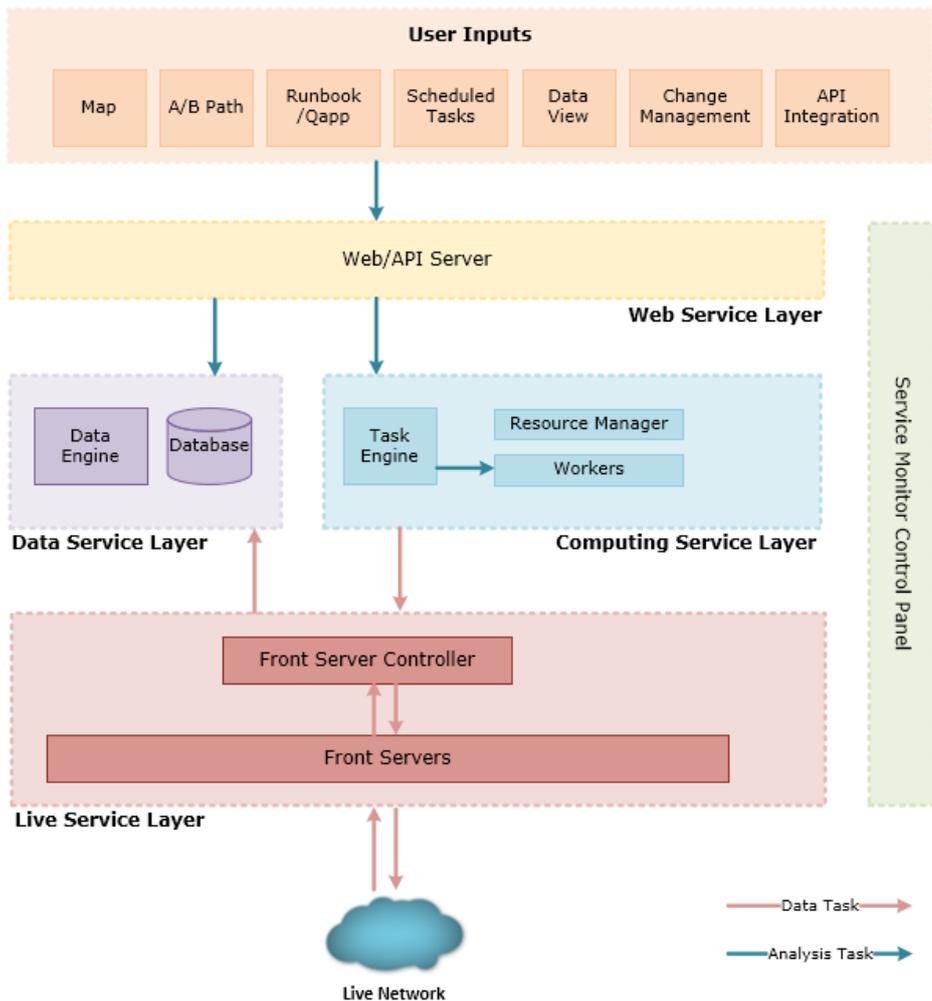
Component		Description
Third-Party Components	MongoDB	serves as a Database Server to store system data.
	Elasticsearch	serves as an Index Server to provide full-text search and analytics engine in a distributed multi-user environment, and will be used for: <ul style="list-style-type: none"> <li>Full Search</li> <li>Dynamic Search</li> </ul>

Component	Description
	<ul style="list-style-type: none"> <li>▪ Service Monitor</li> </ul>
<b>Redis</b>	serves as a Cache Server to provide memory cache for the system, and will be used for: <ul style="list-style-type: none"> <li>▪ Discovering/Building Topology</li> <li>▪ Selecting Leader for FSC HA</li> <li>▪ System Setting Cache</li> <li>▪ XFAgent\WorkerShell\EVShellCharp</li> <li>▪ Qapp (Delta Operator and Monitor Summary)</li> </ul>
<b>RabbitMQ</b>	serves as a Message Server to prioritize and forward requested tasks, and will be used for: <ul style="list-style-type: none"> <li>▪ Multi-DC Switchover</li> <li>▪ XF (execution framework)</li> <li>▪ Communication among multiple Web Servers</li> <li>▪ FSC</li> </ul>
<b>NetBrain Components</b>	<p><b>Web Server</b> serves static content such as HTML, JavaScript and CSS resources, which serves as the user interface of the Thin Client. It will be used for:</p> <ul style="list-style-type: none"> <li>▪ Extending Neighbors</li> <li>▪ Calculating Topology Links</li> <li>▪ Applying Data Views</li> <li>▪ Data View Template</li> <li>▪ Exporting Map to Word</li> <li>▪ Exporting to Visio</li> <li>▪ Building Sites</li> <li>▪ Processing Search Data</li> </ul> <p><b>Web API Server</b> provides the front-end web applications to support the browser-based Thin Clients, and serves RESTful API calls from third-party applications for integration.</p> <p><b>Worker Server</b> serves as a resource manager to support computing tasks. It relies on both Redis and RabbitMQ to work. It will be used for:</p> <ul style="list-style-type: none"> <li>▪ Discovering/Benchmarking/Tuning</li> <li>▪ Building Topology/Site/Device Group/MPLS Cloud VRT/Default DV/Visual Space</li> <li>▪ On-demand/Schedule Run Qapp/Gapp</li> <li>▪ On-demand Path Calculation</li> <li>▪ Executing CLI Commands/Ping/Traceroute/Retrieving Live Data</li> <li>▪ Exporting/Updating Map</li> </ul> <p><b>Task Engine</b> coordinates computing tasks.</p> <p><b>Front Server Controller</b> serves to coordinate and communicate with Front Servers and other components.</p> <p><b>Front Server</b> serves as a polling server to collect and parse live network data. It is the only component required to access the live network.</p> <p><b>Service Monitor Agent</b> monitors the health of your NetBrain Servers with operations management of related services.</p>

Component		Description
	<b>Ansible Agent</b> (add-on) <span style="color: red;">New</span>	integrates with Ansible to define, execute playbooks and visualize results in Change Management Runbooks.
	<b>Smart CLI</b> (add-on) <span style="color: red;">New</span>	provides a Telnet/SSH client to connect to devices from Windows, and can be integrated with NetBrain workflows.

## Layered System Architecture

The system components can be organized in the following four layers.



## 4.2.2.Component Changes and New Components

Component		Changes in IEv8.0
Third-Party Components	MongoDB	Upgrade from v3.6.4 to v4.0.6.
	Elasticsearch	Upgrade from v6.5.2 to v6.7.2.
	Redis	<ul style="list-style-type: none"> <li>▪ Replace Windows Redis with Linux Redis</li> <li>▪ Support Redis HA installed in two servers</li> <li>▪ Add Redis for redis cluster function</li> <li>▪ Upgrade to 5.0.4 from 3.0.504</li> </ul>
	RabbitMQ	<ul style="list-style-type: none"> <li>▪ Replace Windows RabbitMQ with Linux RabbitMQ</li> <li>▪ Improve RabbitMQ network partition</li> <li>▪ Upgrade from 3.7.7 to 3.7.14.</li> </ul>
NetBrain Components	Web Server	No changes.
	Web API Server	Add Knowledge Cloud Agent Service.
	Task Engine Worker Server	<ul style="list-style-type: none"> <li>▪ Add the mechanism to auto-register Worker Server to Task Engine.</li> <li>▪ Add the compression and cleaning mechanism for logs in RMAgent.</li> <li>▪ Regularly clean up Flowengine database.</li> <li>▪ Resume Multi-DC switchover.</li> <li>▪ Optimize task consistency between XF and XFAgent.</li> <li>▪ Allow adding memory limit and exclusive use of memory when submitting a task.</li> <li>▪ Fix multi-threading issues for XF.</li> <li>▪ Support of configuration file upgrade.</li> <li>▪ Enable path related services.</li> <li>▪ Add the interface of waitTaskFlow.</li> </ul>
	Front Server Controller	Save and sync Golden Baseline data.
	Front Server	Save and sync Golden Baseline data.
	Service Monitor Agent	<ul style="list-style-type: none"> <li>▪ Collect 4 basic metrics for new Knowledge Cloud, including CPU/memory usage, memory usage rate, thread count.</li> <li>▪ Collect metrics for multiple MongoDB instances.</li> <li>▪ Collect metrics for data/log directory and size.</li> <li>▪ Optimize encapsulation of sending HTTP requests, add new logics and reuse long connections to avoid frequently change of port numbers.</li> <li>▪ Upgrade python from v to v3.7.2.</li> </ul>
	Ansible Agent (add-on) <b>New</b>	New component.

Component		Changes in IEv8.0
	Smart CLI (add-on) <b>New</b>	New component.

### 4.2.3. Optimize Installation Process

IEv8.0 implements the following enhancements to installation:

- **New Operating System Support**

NetBrain Windows components can be deployed on Windows Server 2019 Standard/Datacenter Edition, while, NetBrain Linux components can only be deployed on CentOS 7.5/7.6/7.7, and RHEL 7.5/7.6.7.7. If the versions of CentOS and RHEL are lower than 7.5, upgrading the operating system is required first.

- **Linux RabbitMQ and Redis**

RabbitMQ and Redis are migrated from Windows to Linux. Cluster deployments for RabbitMQ (Master/Slave) and Redis (Master/Slave/Sentinel) are also supported. When installing a slave Redis node on a Linux server, a sentinel Redis node is automatically installed on this Linux server.

- **Simplify All-in-Two Deployment**

Both Redis and RabbitMQ are installed with Application Server (Linux) at one shot, while Service Monitor Agent (Windows) is installed with Application Server (Windows) at one shot. Besides, to reduce the times to input connection credentials when installing Application Server, only one username and password is created and applied to MongoDB/Elasticsearch/RabbitMQ/Redis.

- **Simplify Task Engine Installation**

Auto register Worker Server to Task Engine and remove the Worker Server connection configuration (Worker Hostname) from the Installation Wizard of Task Engine.

- [Start Services with Restricted Privileges](#)

Create a Windows user account with controlled privileges and use it to run the services of Task Engine/Front Server Controller/Service Monitor Agent/IE to enhance security.

- [Upgrade Third-Party Components to the Latest](#)

Upgrade MongoDB/Elasticsearch/Redis/RabbitMQ/Python to a newer official released version to apply the updates. Click the above link to see more details.

- **Add Logrotate Configuration File for MongoDB/Redis**

Add the **mongod.conf** and **redis.conf** files under the **/etc/logrotate.d** directory to allow users to configure the logrotate settings for MongoDB and Redis. Cron tasks are also created for timely run logrotate for MongoDB and Redis.

- **Add KVAP Setting for Web Server/Web API Server/Worker Server/Front Server Controller**

[KVAP](#) (Keyvault Administration Passphrase) is designed to create a passphrase to initialize the system keyvault and is required for KeyVault Administration tasks, such as key rotation, and maintenance tasks like adding new hosts. This passphrase is not stored in NetBrain and cannot be recovered by any means. If users lose or forget this passphrase, they will have to re-install this product to gain access to the KeyVault, which will result in the data loss.

- **Print More Installation Logs for Linux Components**

To help users troubleshoot installation failures and locate the causes easier and quicker for Linux components, more installation logs are collected and printed on the console during the installation process.

- **Rewrite Installation Scripts for All Linux Packages in Uniform Standard**

- Rename the installation configuration file to **setup.conf** for all NetBrain Linux components.
- Add License Agreement to all NetBrain Linux components.
- Use **systemd** to manage all NetBrain Linux services, such as starting and stopping services.
- Back up the **uninstall.sh** script of all NetBrain Linux components under the **/usr/lib/netbrain/installer/<component\_name>** directory.
- Store the logs generated during the installation, uninstallation, and upgrade of all Linux components under the **/var/log/netbrain/installationlog** directory.

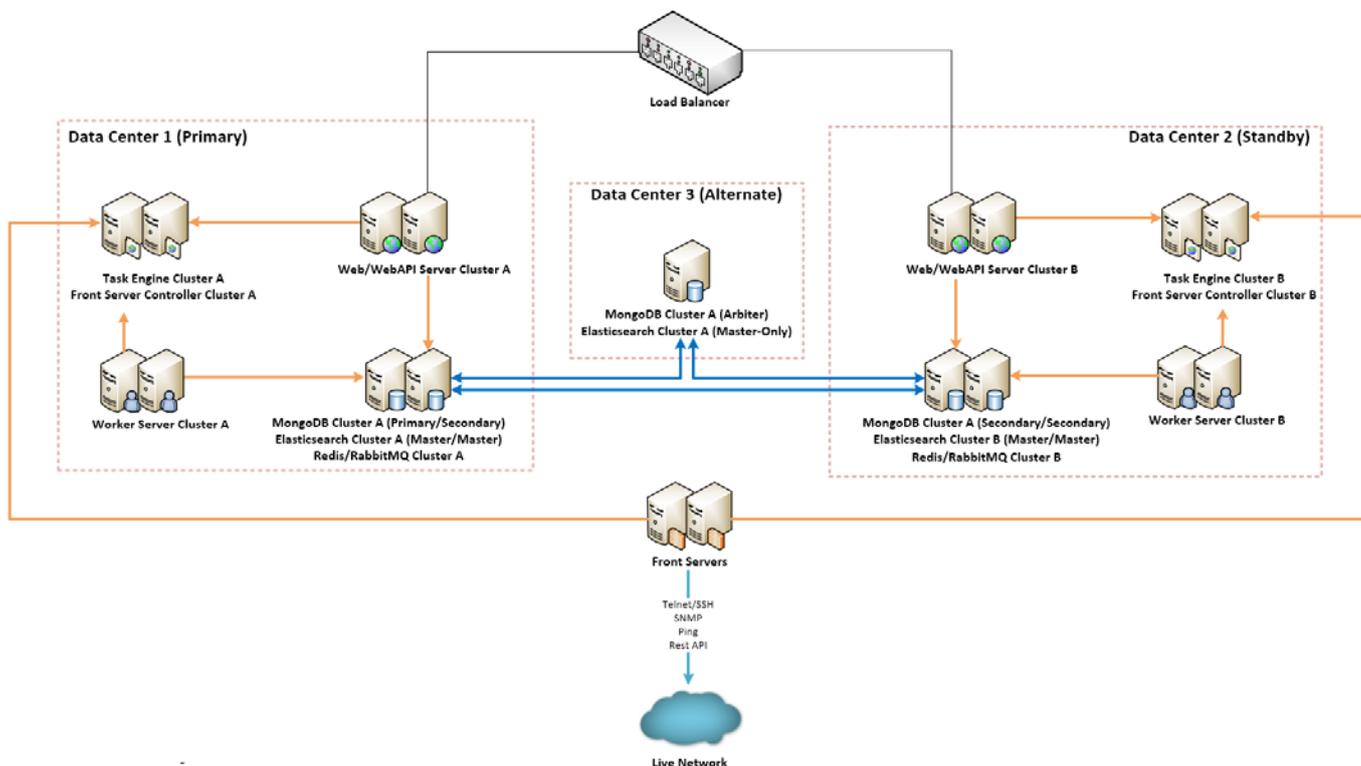
---

## 4.2.4. System Switchover Across Multiple Data Centers

In IEv7.0b, NetBrain system supported deployments across multiple data centers and manual switchover among these data centers, to offer high availability and failover in case of power outages, network

interruptions, natural disasters, or other emergencies in one data center. However, the multi-DC switchover solution was unavailable since IEv7.1 for some reason.

The 8.0 system brings the multi-DC switchover solution back online by adapting to the architecture changes. Customers can deploy two systems in multi-DC and manually switchover with one click.



## 4.3. Administering NetBrain and Others

### 4.3.1. Domain Health Report

To accelerate the tune-up phase (customization for topology/path data accuracy) before delivering a fine-tuned domain to end users, IEv8.0 provides a report with more detailed statistics about domain health. As a benefit, the domain administrators can proactively focus on the items that need their attention and resolve the potential problems existing in a domain as earlier as possible.

The domain health report is generated on-demand and contains both a summary and a variety of categories.

Domain Management Tenant: Initial Tenant Domain: domain16 Operations kang

Report Generated Time: 12/18/2019 01:15:30 PM Refresh Create Health Report

Basic Network Settings: 6 need attention Discovery Status: 5 need attention Path: 2 failed Others: 11 need attention Export

**Report Summary**

**Driver Associated Device:** 22 Driver Applied, 148 Devices, 4887 Interfaces

Device Driver	Associated Device Count
Cisco IOS Switch	50
End System	45
Cisco Router	26
Cisco ASA Firewall	5
Cisco Nexus Switch	2
Avaya Switch	2
Arista Switch	2
Juniper EX Switch	2
Dell Sonicwall	1

**Basic Network Settings Completeness:**

Attention	Index	Count
	Stand-alone Front Server (defined)	3
!	Stand-alone Front Server (unused)	2

Report Category	Description and Example																				
Summary	This summary shows the number of items that need the domain admin to pay attention to.																				
Driver Associated Device	<p>This category shows the number of devices that are discovered by drivers.</p> <p>22 Driver Applied, 148 Devices, 4887 Interfaces</p> <table border="1"> <thead> <tr> <th>Device Driver</th> <th>Associated Device Count</th> </tr> </thead> <tbody> <tr><td>Cisco IOS Switch</td><td>50</td></tr> <tr><td>End System</td><td>45</td></tr> <tr><td>Cisco Router</td><td>26</td></tr> <tr><td>Cisco ASA Firewall</td><td>5</td></tr> <tr><td>Cisco Nexus Switch</td><td>2</td></tr> <tr><td>Avaya Switch</td><td>2</td></tr> <tr><td>Arista Switch</td><td>2</td></tr> <tr><td>Juniper EX Switch</td><td>2</td></tr> <tr><td>Dell Sonicwall</td><td>1</td></tr> </tbody> </table>	Device Driver	Associated Device Count	Cisco IOS Switch	50	End System	45	Cisco Router	26	Cisco ASA Firewall	5	Cisco Nexus Switch	2	Avaya Switch	2	Arista Switch	2	Juniper EX Switch	2	Dell Sonicwall	1
Device Driver	Associated Device Count																				
Cisco IOS Switch	50																				
End System	45																				
Cisco Router	26																				
Cisco ASA Firewall	5																				
Cisco Nexus Switch	2																				
Avaya Switch	2																				
Arista Switch	2																				
Juniper EX Switch	2																				
Dell Sonicwall	1																				
Basic Network Setting Completeness	This category shows the summary of the predefined Network Settings, including the number of Front Server, Front Server Group, Private Key, Jumpbox, Telnet/SSH Login, Privilege Login, SNMP																				

Report Category	Description and Example
-----------------	-------------------------

String, API Server, CheckPoint OPSEC and their usage.

Attention	Index	Count
	Stand-alone Front Server (defined)	3
!	Stand-alone Front Server (unused)	2
	Stand-alone Front Server (over 5000 devices)	0
	Front Server Group (defined)	0
	Front Server Group (unused)	0
	Front Server Group (over 5000 devices per ...)	0
	Private Key (defined)	6
!	Private Key (unused)	6
	Jumpbox (defined)	3
!	Jumpbox (unused)	3
	Telnet/SSH Login (defined)	44
!	Telnet/SSH Login (unused)	44
	Privilege Login (defined)	21
!	Privilege Login (unused)	21
	SNMP String (defined)	18
!	SNMP String (unused)	18
	API Server	11
	CheckPoint OPSEC	0

**Note:** The attention ( ! ) is a reminder, but not an alert. If the configuration is not necessary, users can ignore the attention.

Discovery Status	Description and Example
------------------	-------------------------

This category shows a summary of the discovery report, license usage, and issues encountered in the discovery process.

Attention	Index	Count
	Discovered Devices	Legacy Device: 102
	Licensed Node Usage	102 out of 10000 Licensed Nodes (Usage: 1.02%)
!	Discovered by SNMP	81
!	Unknown IP	82
	Missed Devices	0
!	Unclassified Network Devices	1
!	Unknown SNMP SysObjectID	1
!	Subnet with Conflicted IPs	5
	Zone	4
	Hostname Change	0

Report Category	Description and Example																																																			
Site Definition Completeness	<p>This category shows the usage summary of each site.</p> <p>Latest Build Time: 12/18/2019 01:15:30 PM</p> <table border="1"> <thead> <tr> <th>Attention</th> <th>Index</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>!</td> <td>Container Site</td> <td>0</td> </tr> <tr> <td>!</td> <td>Leaf Site</td> <td>0</td> </tr> <tr> <td></td> <td>Leaf Site (with 0 device)</td> <td>0</td> </tr> <tr> <td></td> <td>Leaf Site (over 100 devices)</td> <td>0</td> </tr> <tr> <td>!</td> <td>Unassigned Devices</td> <td>148</td> </tr> </tbody> </table>	Attention	Index	Count	!	Container Site	0	!	Leaf Site	0		Leaf Site (with 0 device)	0		Leaf Site (over 100 devices)	0	!	Unassigned Devices	148																																	
Attention	Index	Count																																																		
!	Container Site	0																																																		
!	Leaf Site	0																																																		
	Leaf Site (with 0 device)	0																																																		
	Leaf Site (over 100 devices)	0																																																		
!	Unassigned Devices	148																																																		
Benchmark Task Health	<p>This category shows the latest two execution summaries of each benchmark task.</p> <p><a href="#">Basic System Benchmark</a></p> <table border="1"> <thead> <tr> <th>Attention</th> <th>Execution Time</th> <th>Result</th> <th>Duration</th> <th>Configuration Retrieval Success Rate</th> </tr> </thead> <tbody> <tr> <td></td> <td>12/16/2019 12:58:31 PM</td> <td>Succeeded</td> <td>8 mins 34 secs</td> <td>99.32% (145/146 devices)</td> </tr> </tbody> </table> <p><a href="#">Benchmark1</a></p> <table border="1"> <thead> <tr> <th>Attention</th> <th>Execution Time</th> <th>Result</th> <th>Duration</th> <th>Configuration Retrieval Success Rate</th> </tr> </thead> <tbody> <tr> <td></td> <td>12/18/2019 9:39:13 AM</td> <td>Succeeded</td> <td>37 secs</td> <td>40% (2/5 devices)</td> </tr> </tbody> </table>	Attention	Execution Time	Result	Duration	Configuration Retrieval Success Rate		12/16/2019 12:58:31 PM	Succeeded	8 mins 34 secs	99.32% (145/146 devices)	Attention	Execution Time	Result	Duration	Configuration Retrieval Success Rate		12/18/2019 9:39:13 AM	Succeeded	37 secs	40% (2/5 devices)																															
Attention	Execution Time	Result	Duration	Configuration Retrieval Success Rate																																																
	12/16/2019 12:58:31 PM	Succeeded	8 mins 34 secs	99.32% (145/146 devices)																																																
Attention	Execution Time	Result	Duration	Configuration Retrieval Success Rate																																																
	12/18/2019 9:39:13 AM	Succeeded	37 secs	40% (2/5 devices)																																																
Path Calculation Health	<p>This category shows the number of paths that have calculation failures.</p> <p>Total 5 paths, 3 succeeded, 0 not executed, 2 failed</p>																																																			
Disk Management Setting Completeness	<p>This category shows the summary of the predefined auto-clean rules for global data and the disk alert rules for MongoDB.</p> <p><a href="#">Global Data Clean Settings</a></p> <table border="1"> <thead> <tr> <th>Attention</th> <th>Data Type</th> <th>Data Clean</th> </tr> </thead> <tbody> <tr> <td></td> <td>Data Engine Data</td> <td>Enabled</td> </tr> <tr> <td></td> <td>Qapp/Gapp Execution Logs</td> <td>Enabled</td> </tr> <tr> <td>!</td> <td>One-IP Table Entries</td> <td>Disabled</td> </tr> <tr> <td>!</td> <td>Discovery/Benchmark Logs</td> <td>Disabled</td> </tr> <tr> <td>!</td> <td>Application Path History Data</td> <td>Disabled</td> </tr> <tr> <td>!</td> <td>DataUnitStorage</td> <td>Disabled</td> </tr> <tr> <td></td> <td>Backup Maps</td> <td>Enabled</td> </tr> <tr> <td>!</td> <td>API Triggered Automation Task</td> <td>Disabled</td> </tr> <tr> <td>!</td> <td>Event Entries in Event Console</td> <td>Disabled</td> </tr> <tr> <td></td> <td>Execution Logs for Scheduled Data View Template &amp; Parser Tasks</td> <td>Enabled</td> </tr> <tr> <td></td> <td>Dashboard Activity Data Table</td> <td>Enabled</td> </tr> </tbody> </table> <p><a href="#">MongoDB Disk Alert Rules: (System Settings)</a></p> <table border="1"> <thead> <tr> <th>Attention</th> <th>Type</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>!</td> <td>Send emails when alert detected</td> <td>Disabled</td> </tr> <tr> <td>!</td> <td>Send emails and delete Data Engine data when alert detected</td> <td>Disabled</td> </tr> <tr> <td></td> <td>Send emails and disable write permission to MongoDB when alert detected</td> <td>Enabled</td> </tr> <tr> <td>!</td> <td>Email Address</td> <td>None</td> </tr> </tbody> </table>	Attention	Data Type	Data Clean		Data Engine Data	Enabled		Qapp/Gapp Execution Logs	Enabled	!	One-IP Table Entries	Disabled	!	Discovery/Benchmark Logs	Disabled	!	Application Path History Data	Disabled	!	DataUnitStorage	Disabled		Backup Maps	Enabled	!	API Triggered Automation Task	Disabled	!	Event Entries in Event Console	Disabled		Execution Logs for Scheduled Data View Template & Parser Tasks	Enabled		Dashboard Activity Data Table	Enabled	Attention	Type	Status	!	Send emails when alert detected	Disabled	!	Send emails and delete Data Engine data when alert detected	Disabled		Send emails and disable write permission to MongoDB when alert detected	Enabled	!	Email Address	None
Attention	Data Type	Data Clean																																																		
	Data Engine Data	Enabled																																																		
	Qapp/Gapp Execution Logs	Enabled																																																		
!	One-IP Table Entries	Disabled																																																		
!	Discovery/Benchmark Logs	Disabled																																																		
!	Application Path History Data	Disabled																																																		
!	DataUnitStorage	Disabled																																																		
	Backup Maps	Enabled																																																		
!	API Triggered Automation Task	Disabled																																																		
!	Event Entries in Event Console	Disabled																																																		
	Execution Logs for Scheduled Data View Template & Parser Tasks	Enabled																																																		
	Dashboard Activity Data Table	Enabled																																																		
Attention	Type	Status																																																		
!	Send emails when alert detected	Disabled																																																		
!	Send emails and delete Data Engine data when alert detected	Disabled																																																		
	Send emails and disable write permission to MongoDB when alert detected	Enabled																																																		
!	Email Address	None																																																		

Report Category	Description and Example		
Map Layout Settings Completeness	This category shows the number of custom tags.		
	Attention	Index	Count
	!	Customized Layout	0
		Devices with Tags Associated	0
	!	Devices without Tags Associated	148
		Site Maps with Layouts Associated	0
		Site Maps without Layouts Associated	0

For more reference about the report content, refer to [online help](#).

### 4.3.2. Granular Policy for Device Access

In previous versions, the NetBrain IE system only provides the general control on what NetBrain features users can perform and use, so the domain administrator can only grant users access to either all network devices or no device access at all. To meet security requirements, IEv8.0 introduces the granular controls to allow specified users to have access privileges to specified devices.

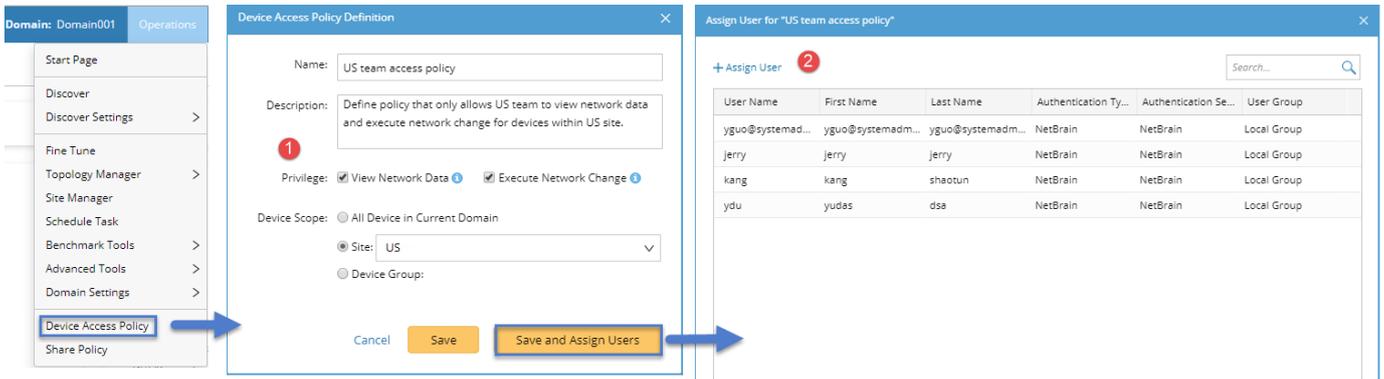
#### Key Use Case

Large customers usually divide and manage their networks geographically, e.g., China, U.S., EMEA, and have a strict security policy that each network management team can only view the data of their managed devices. For example, the U.S. team can only view the data of the devices in the U.S. site. Moreover, another common situation is that a specific team can view the data of a specific device type. For example, only the security team can view the configuration file of the firewalls.

With the granular device access policy, the domain administrator can assign users in different regions to comply with different device access policies. These policies determine whether users can be authorized to view the data of specific devices or execute changes based on their associated policies.

#### Key Use Flow

The use flow to define a Device Access Policy contains two steps:



**Prerequisites:** The function privilege of “Share Policy Management” is required for defining Device Access Policies.

1. Define a Device Access Policy by selecting device scope and at least one of the following privileges:

- **View Network Data**

The network data includes: Configuration File, Route Table, ARP Table, MAC Table, NDP Table, STP Table, BGP Advertised Table, NCT Table, GDR Property, CLI/SNMP/API Original Result, Parsed Variable and Configlet.

- **Execute Network Change**, including:

- Run the Execute node in a Change Management Runbook.
- Run the Ansible Task node in a Runbook.

**Note:** The function privileges of “Access to Live Network” and “View Network Change” are also required for users who execute network changes.

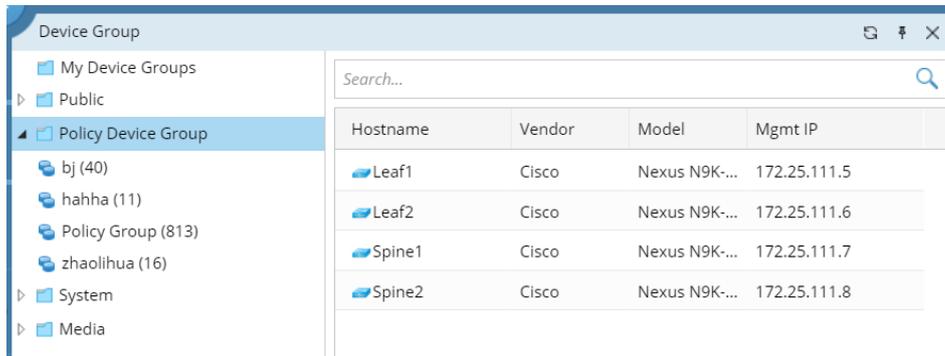
2. Assign Users to comply with the Policy.

Alternatively, the user assignment can also be done when configuring the domain share policy or when adding a system user account.

**Note:** Each user can be associated with multiple policies. If a user hasn’t been associated with any policy, he/she will not be authorized to view the detailed data of any device in a domain or execute changes on any device in a domain.

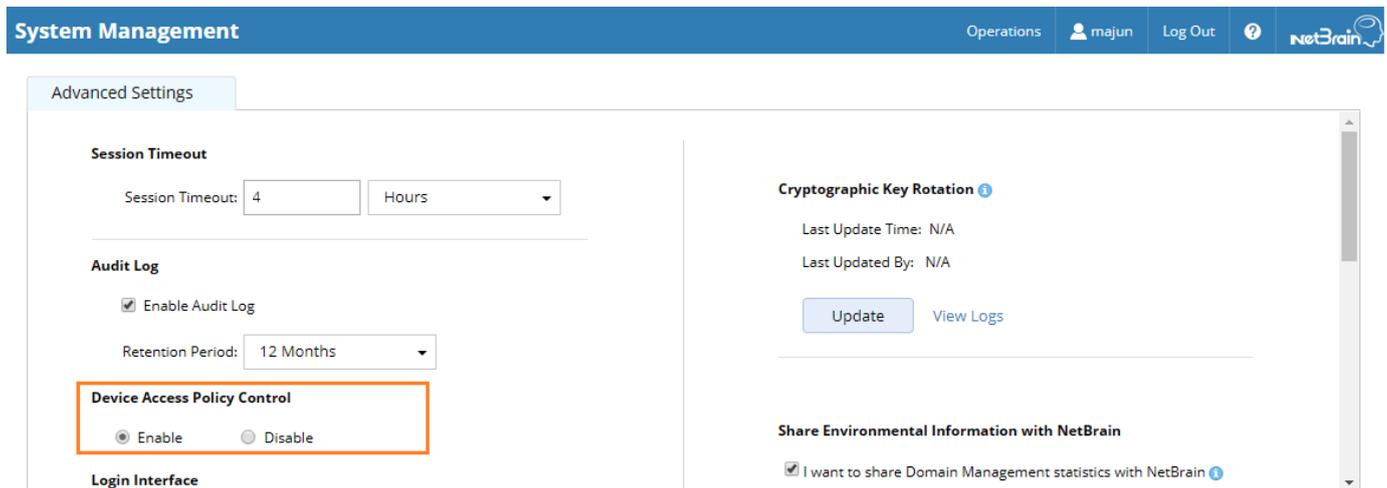
## Policy Device Group

Policy Device Group is a new type of device group dedicated to defining device access policies. Only the users with the Share Policy Management privilege can create, edit, and delete a Policy Device Group.



## Enable Device Access Policy Control

Before using this feature, the system administrator must enable the “Device Access Policy Control”.



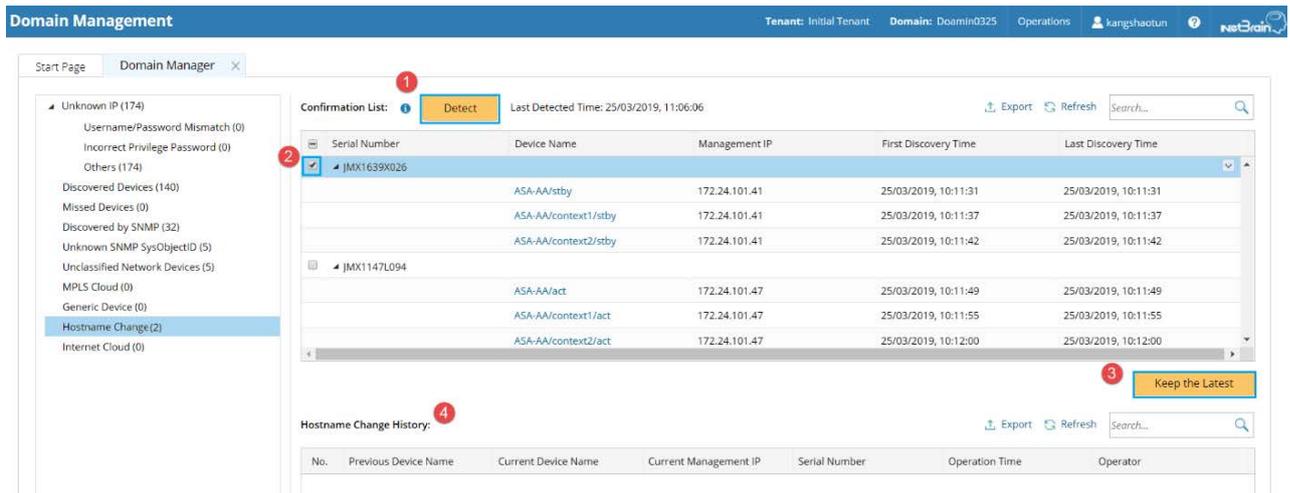
### 4.3.3. Detect and Resolve Hostname Duplication

NetBrain system uses the hostname as the unique identifier for devices in live network discovery, so a hostname-changed device will be recognized as a new one and added to a domain one more time. Even though the possible hostname changes can be found by tuning live access, there is no further automating mechanism to help users correct or clean up them quickly, which will cause duplicate IPs, topology accuracy issues, obsolete maps, etc.

IEv8.0 offers the capability of resolving device with hostname-changes as a supplement to topology accuracy. The system scans device serial numbers and lists the devices with duplicate serial numbers as possible candidates with hostname changes. Then domain administrators can view the report after each discovery task and fix these issues.

## Resolve Hostname Changes

1. In the Domain Manager, check duplicate devices with hostname-change possibilities by clicking **Detect**.



2. Review and confirm the hostname-change possibilities sorted by serial number.
3. Click **Keep the Latest**. The system will use the original GUID of the new device, and use the hostname of the new device. The current baseline data of the new device will be merged with the old device, and duplicate devices will be removed.
4. The latest entry will be logged in the Hostname Change History, and then verify the result on a map.

## What Has Been Updated to Resolve Hostname Change

The new device hostname will be updated in the following features/files:

Feature/File	Description
Config File	The configuration file of the device will be updated to the new one.
One IP Table	The old device name will be updated to the new name.
GDR	<ul style="list-style-type: none"> <li>Device: The data of the new device will overwrite the data of the old one.</li> <li>Interface: Keep the interface information of the old device, and delete the information of the new device.</li> </ul>
Share/Private Device Settings	The data of the new device will overwrite the data of the old one.

Feature/File	Description
Device Group	Old device hostnames in the device group will be updated to the new hostnames.
Site	Old device hostnames in the site will be updated to the new hostnames.
MPLS CE List and VRT	Old hostnames of CE devices will be updated to the new hostnames.
Topology	Rebuild the topology manually to ensure its accuracy after clicking Keep the Latest.
Map	The old device names will be changed to the new names after updating the map.
Site Map	The old device names will be changed to the new names after updating the map.
Default Data View	The default data views of the old and new devices will be deleted, and the rebuilding of the data view will be triggered when the map is open next time.
Data View	Old device hostnames in a data view will be updated to the new hostnames.
Customized Device Icon	If the users customize the device icons for the devices with old hostnames, the correspondence remains unchanged between the customized icons and the devices with new hostnames.
Context Map	Old hostnames in the context map will be updated to new hostnames.
Other Features	The data saved in some other features will also be updated.

## Hostname Change Detect for VDC Devices

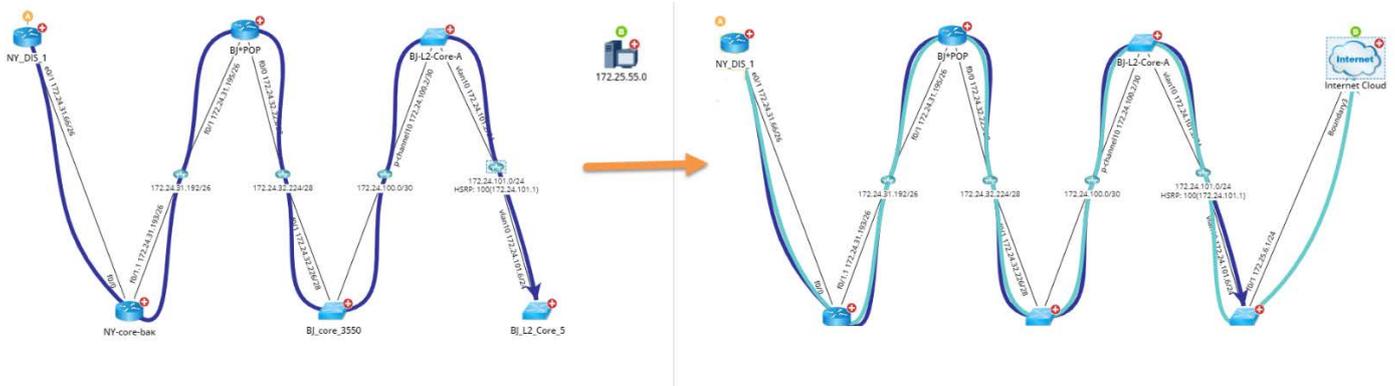
Because each VDC device in a group is a unique device with the same SN, the general hostname change logic does not apply to VDC devices. For VDV devices, the system uses the enhanced hostname change logic:

- The system adds a GDR property **VDC\_MAC** to differentiate VDC devices in a group.
- The system checks both SN and MAC address during the hostname change check for VDC devices. Only when two devices have the same SN and MAC address, the system determines that one device is the other one's hostname-changed device.

## 4.3.4.Support Internet Cloud

In previous versions, when a traffic path goes through the Internet, the path will be labeled as a failure.

IEv8.0 introduces the concept of Internet Cloud. Internet Cloud is an emulated device with a randomly assigned interface, which is used to build the connection between the boundary device and the Internet.



## Define Internet Cloud at Domain Level

The Internet boundary interface is selected when the Internet Cloud is defined by Power User. The selected interface will have **isInternetBoundary** property to be true.

## Enhance Path Calculation Logic for Internet Cloud

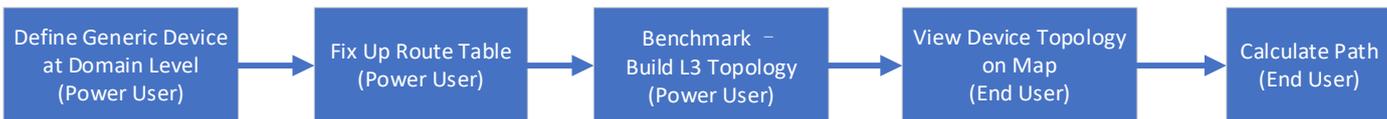
The calculation of a path through an Internet Cloud will be triggered if the outbound interface of the next-hop device has **isInternetBoundary** property to be true.

## 4.3.5.Resolve Undiscoverable Devices

IEv8.0 allows a user to manually add the devices which are not accessible as a generic device to ensure the topology and data model completeness.

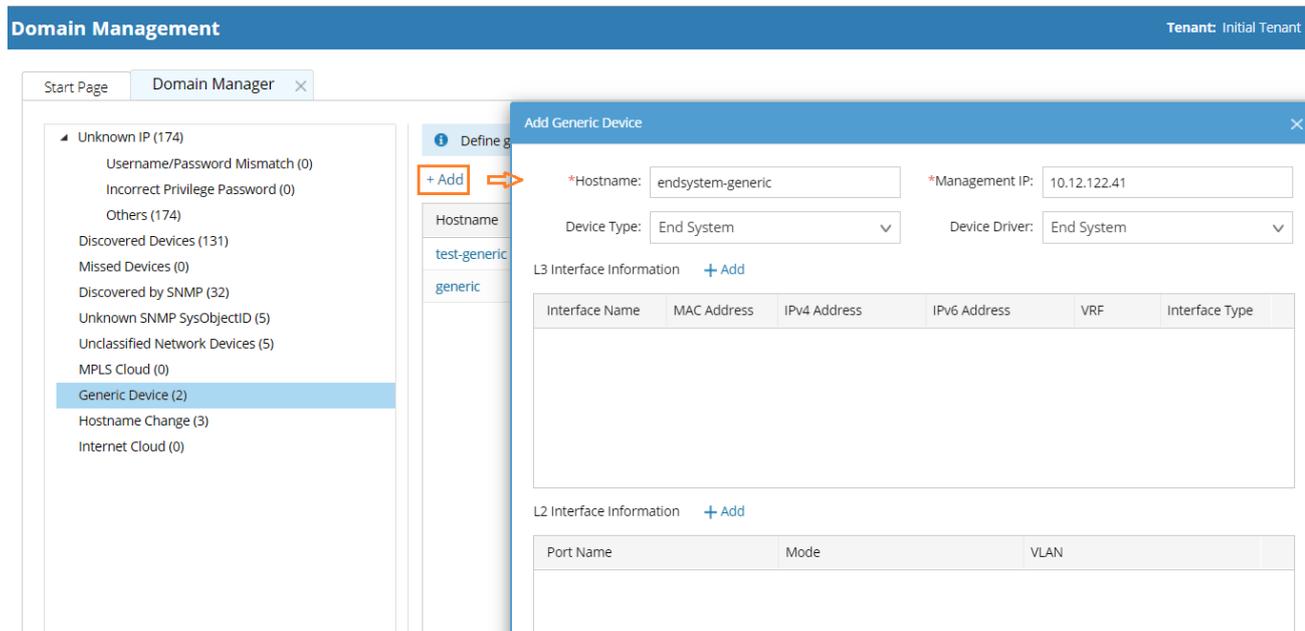
## Reference Flow

A reference flow of using a generic device is as follows:

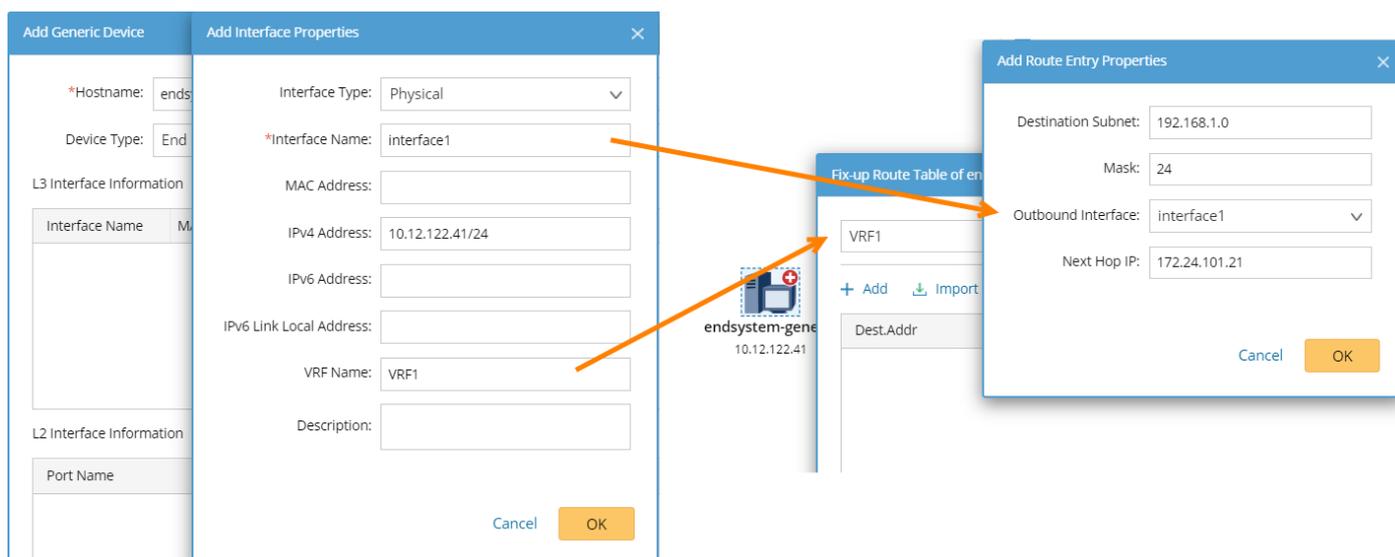


## Define Generic Device

IEv8.0 allows users to manually add any undiscoverable devices to a domain by adding their hostnames, management IPs, and device types.



To help the added device establish a correct connection with other devices, users can define its interfaces, such as interface name and VRF name.



## Define Fix-up Routes for Generic Device

To make the added generic devices participate in path calculation, users need to define fix-up route tables for them. See [Fix-up Route Table](#) for more details.

### 4.3.6.Restore Backup Maps

To help the domain administrator deal with unexpected map update errors and roll back to the previous version of a map, I Ev8.0 adds the capability to restore the maps that have been backed up through benchmark tasks.

## Back Up Maps through Benchmark Settings

By default, the “back up map” function is disabled. To enable it, go to **Benchmark > Additional Operations after Benchmark > Update Maps** and slide the button to select “Yes”.

Update Maps Go to Operations > Domain Maintenance > Update Map Manager to see update results.

Enable	Map	Export to Visio	Back Up Maps
<input checked="" type="checkbox"/>	Update Site Maps <a href="#">Select Maps</a>	Yes No <input type="text"/> Browse	Yes No
<input checked="" type="checkbox"/>	Update Shared Device Group Maps <a href="#">Select Maps</a>	Yes No <input type="text"/> Browse	Yes No
<input type="checkbox"/>	Update Context Maps <span>!</span>	Not Applicable	Not Applicable
<input checked="" type="checkbox"/>	Update Public Maps <a href="#">Select Maps</a>	Yes No <input type="text"/> Browse	Yes No

**Note:** To free your disk, backup maps can be deleted on a customized frequency. Go to **Domain Management > Global Data Clean Settings**.

## Restore Backup Maps

The domain administrator can either restore a single map or restore a batch of maps in the Update Map Manager by selecting the timestamp of backups:

The screenshot shows the 'Update Map Manager' interface. At the top, there is a 'Start Page' tab and an 'Update Map Manager' tab. Below this, there is a 'Update Source' dropdown set to 'update1', and 'Map Type' checkboxes for 'Site Maps', 'Device Group Maps', and 'Public Maps'. A 'Restore All' button is highlighted with an orange box. To the right, there is a search bar and a 'Refresh' button. Below these elements is a table with columns: 'Map Name', 'Path', 'Update Source', 'Update Log', 'Map Restore History', and 'Back Up Maps'. The table contains three rows: 'lc', 'lxj', and 'test1', all with 'Device Group\Public' as the path. The 'Back Up Maps' column has values 'No', 'Yes', and 'Yes' respectively. The 'Restore' button for the 'lxj' row is highlighted with an orange box. A 'Restore Map' dialog box is open in the center, with a blue border. It contains the text: 'Only maps with backups can be restored. Once restored, the current map will be overwritten.' Below this text is a list of timestamps: '09/16/2019 11:06:29 AM', '09/16/2019 11:04:47 AM', '09/16/2019 11:03:14 AM', and '09/16/2019 11:01:39 AM'. The third timestamp is selected. At the bottom of the dialog are 'Cancel' and 'Restore' buttons. Orange arrows point from the 'Restore All' button to the 'Restore' button in the dialog, and from the 'Restore' button in the dialog to the 'Restore' button in the table.

**Note:** Only maps with backups can be restored. For a batch restore, all the maps that have backups will be restored with one-click.

### 4.3.7. Auto Clean Execution Logs and More Data

IEv8.0 adds segmented controls to the interface of Global Data Clean Settings so that domain admin can flexibly define rules to auto clean by data type, including Qapp/Gapp Execution Logs, Discovery/Benchmark Logs, Path Data, One-IP Table Entries, and historical data views generated by Qapp.

Domain Management Tenant: Initial Tenant Domain: MVS Operations kang NetBrain

Start Page Network Settings Schedule Task **Global Data Clean Settings**

**Data Engine Data** Manually Delete Data

Only keep  data points for each data type in a month for data older than  months

Delete data older than  months

**Other Data** Refresh Data Size

Data Type	Data Size	Auto - Clean Rule
Qapp/Gapp Execution Logs	N/A	<input checked="" type="checkbox"/> Delete data older than <input type="text" value="14"/> <input type="text" value="days"/>
One-IP Table Entries	21MB	<input type="checkbox"/> Delete data older than <input type="text" value="14"/> <input type="text" value="days"/>
Discovery/Benchmark Logs	N/A	<input type="checkbox"/> Delete data older than <input type="text" value="14"/> <input type="text" value="days"/>
Application Path History Data	N/A	<input type="checkbox"/> Delete data older than <input type="text" value="6"/> <input type="text" value="months"/>
DataUnitStorage	36KB	<input type="checkbox"/> Delete data older than <input type="text" value="1"/> <input type="text" value="months"/>

Save

### 4.3.8. Monitor MongoDB Disk Usage with Email Alerts

If a large proportion of MongoDB disk capacity is consumed, the system response might become slow or even shut down if MongoDB is running out of disk space.

As one solution to tackle this challenge, IEv8.0 adds the email alerting capability to warn users about MongoDB usage when it reaches pre-defined thresholds. Moreover, corresponding actions can be defined and performed to protect the system database from being corrupted.

For example, users will receive both email alerts and in-place warning in the system:

Warning ×



The following database servers have been running out of disk space since 4/1/2019 6:05:17 AM:

localhost.localdomain (10.10.32.106).

Please contact your NetBrain administrator to avoid potential data loss.

Three rules are designed for MongoDB disk alert. Each rule defines different actions with two conditions of disk usage threshold. The corresponding action will be triggered when either condition is met.

Rule	Actions	Condition 1	Condition 2
A (optional)	Send Email Alerts Only	Usage Percentage Reaches <u>80%</u>	Only <u>20GB</u> Free Space Left

B (optional)	Send Email Alerts Delete Historical Data <sup>1)</sup>	Usage Percentage Reaches <u>90%</u>	Only <u>10GB</u> Free Space Left
C (enabled by default)	Send Email Alerts Disable Writing Permissions to MongoDB <sup>2)</sup>	Usage Percentage Reaches <u>93%</u> <sup>3)</sup>	Only <u>5GB</u> <sup>3)</sup> Free Space Left

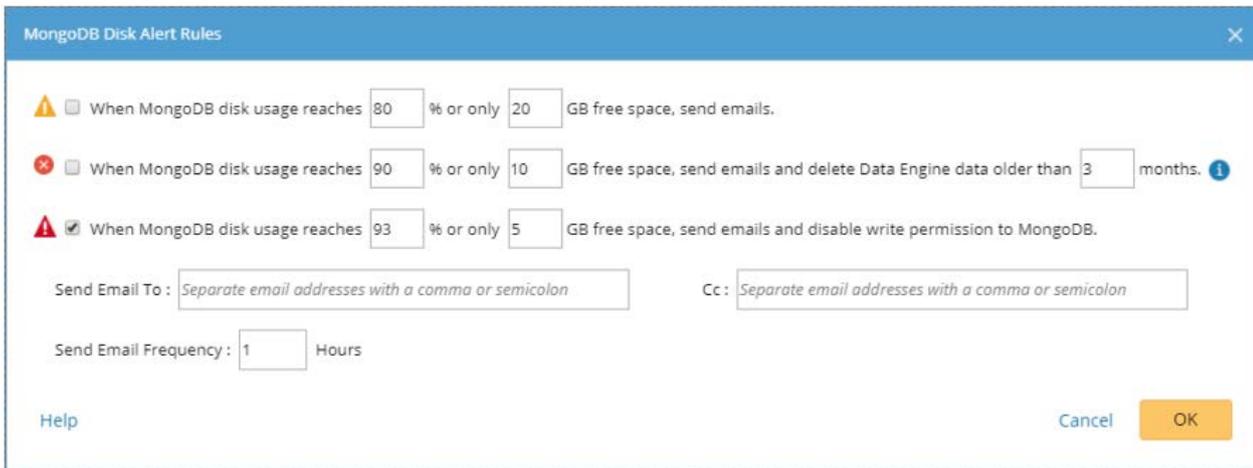
**Notes:**

<sup>1)</sup> The historical data that can be automatically deleted includes configuration files, data tables, CLI commands and parser data for all domains in this MongoDB instance.

<sup>2)</sup> The disabled writing permissions include: saving maps, saving data as the current baseline, and so on.

<sup>3)</sup> The default thresholds for Rule C are recommended values, depending on the configurations of Elasticsearch since it shares the disk with MongoDB. Refer to <https://www.elastic.co/guide/en/elasticsearch/reference/6.7/disk-allocator.html> for more details.

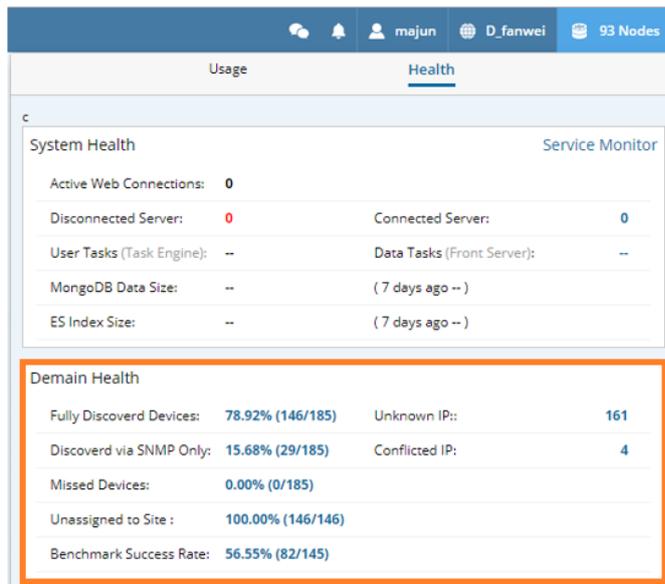
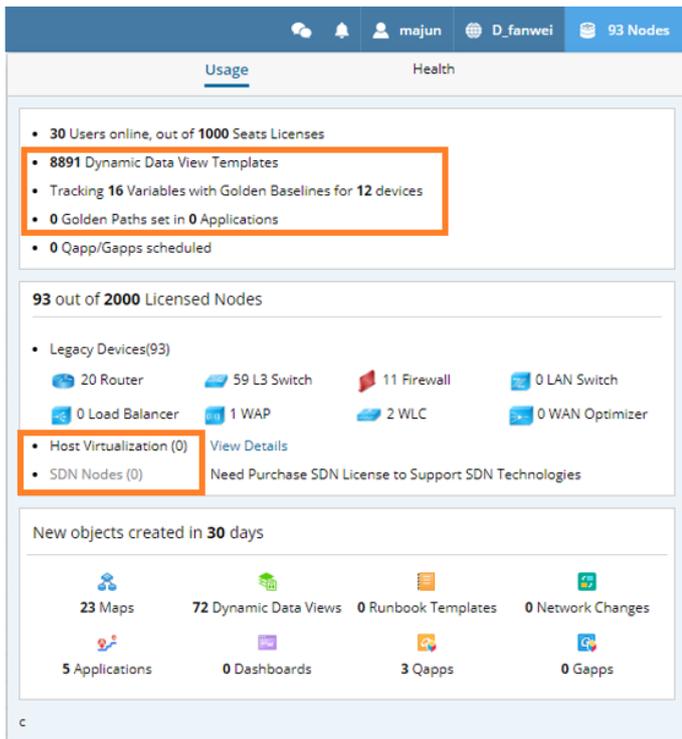
Admin can enable either a single rule to assign quotas, or multiple rules to assign progressive quotas, and add more users to get notified via emails. The thresholds are all configurable.



### 4.3.9. Monitor System Usage and Health

To provide more detailed summary information about the system, the usage and health summary panel is redesigned, including:

- Highlight key metrics in the top area, especially the count of DVT, Golden Baseline Variables and Golden Paths.
- Unify licensed node counts, such as Host Virtualization and [SDN Node Count](#).
- Add more domain statistics in a separate “Health” tab, including various types of device count and rate.



### 4.3.10. Enhancements to System/Tenant Administration

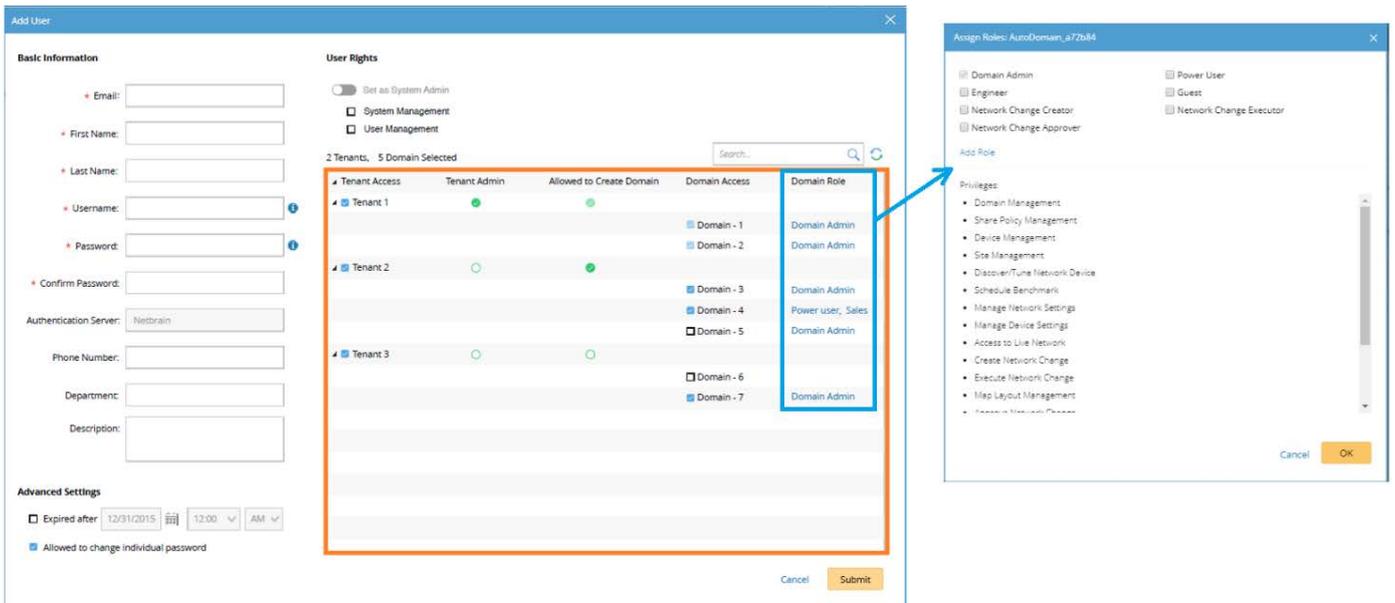
In previous versions, the tenant/domain access assignment of users was configured by different administrators (system admin and domain admin) in different setup stages, and lacked a consolidated control. Moreover, both system management and tenant management page might become dis-organized after more and more features were piled up.

To resolve the above limitations, IEv8.0 introduces the following improvements:

- Assign Domain Access and Roles When Adding a User Account
- Assign Domain Access and Roles When Adding an Email Suffix for User Sign-Up
- Show Maximum/Used Node Count for Each Tenant
- Distinguish Login Sessions via Web/API/Smart CLI
- Show All Tenants and Domains in Domain Dropdown
- Flexible Menu On/Off

#### Assign Domain Access and Roles When Adding a User Account

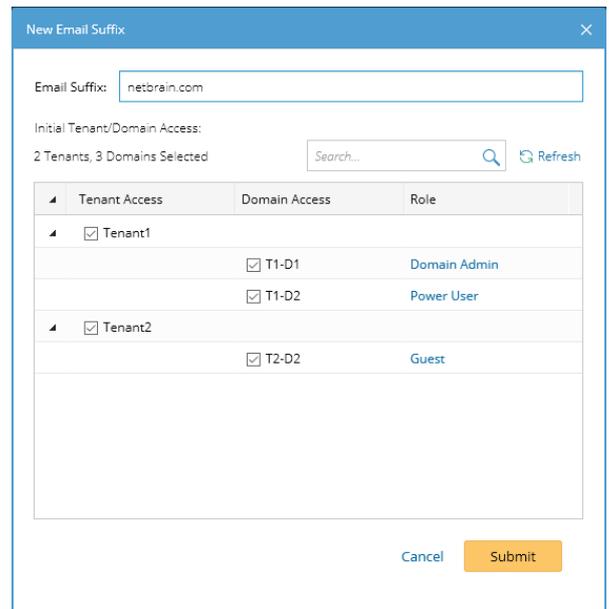
IEv8.0 provides detailed access controls to multiple domains for each user account, and user roles can also be assigned respectively in different domains. System Admin account has access to all domains with Domain Admin role assigned by default.



## Assign Domain Access and Roles When Adding an Email Suffix for User Sign-Up

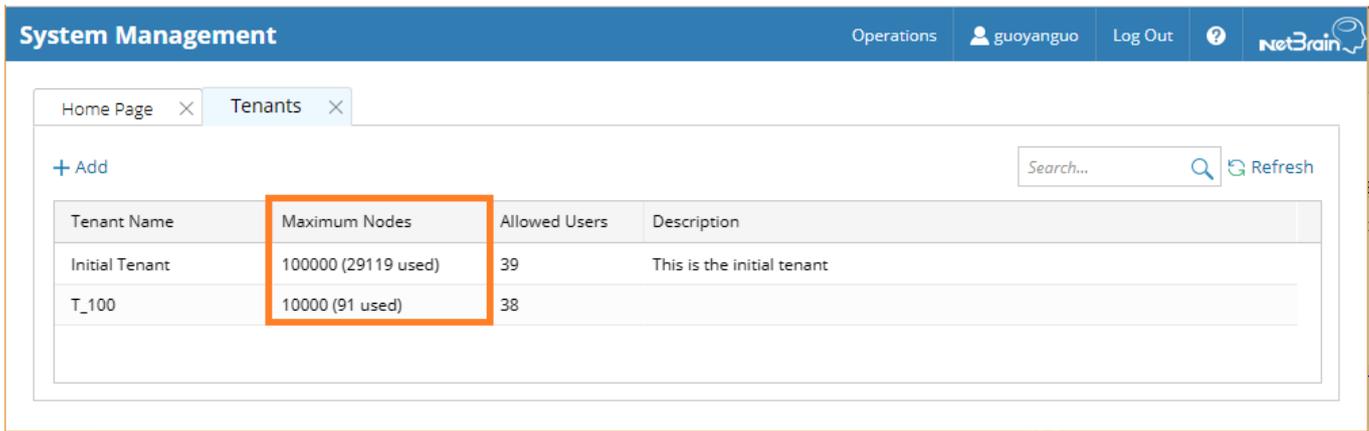
IEv8.0 provides access control to different domains for user accounts with one Email Suffix, and assign user roles by domain.

Moreover, **Email Domain** is renamed to **Email Suffix**, to avoid confusion with **NetBrain Domain**.



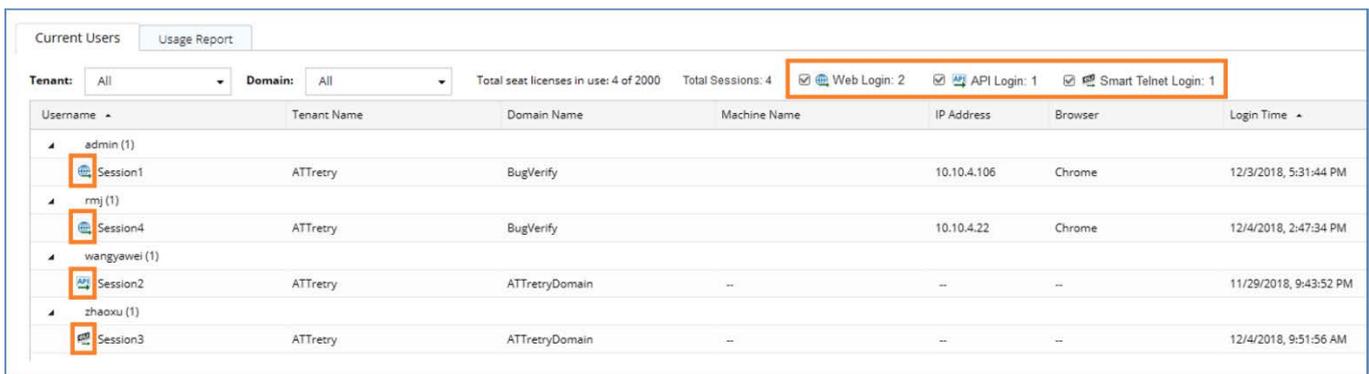
## Show Maximum/Used Node Count for Each Tenant

To make admin aware of the maximum node count and node usage for each tenant, IEv8.0 adds the information in the Tenant List.



## Distinguish Login Sessions via Web/API/Smart CLI

Besides the current login ways: via GUI and API calls, I Ev8.0 introduces a new login type via Smart CLI. To distinguish these three types of user sessions, individual filters and icons are added to User List, in the entrance of both System Management and Domain Management page. The icons also appear in the Usage Report tab to distinguish different login types.

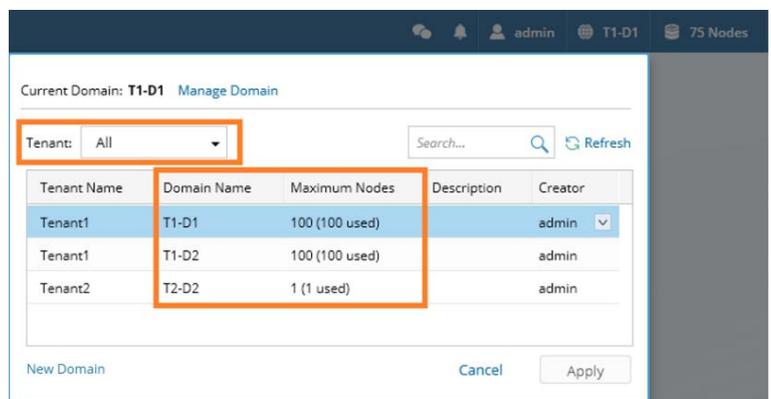


## Show All Tenants and Domains in Domain Dropdown

The domain dropdown is the entrance to other domains, or to the Domain Management page if the current user has the required privilege.

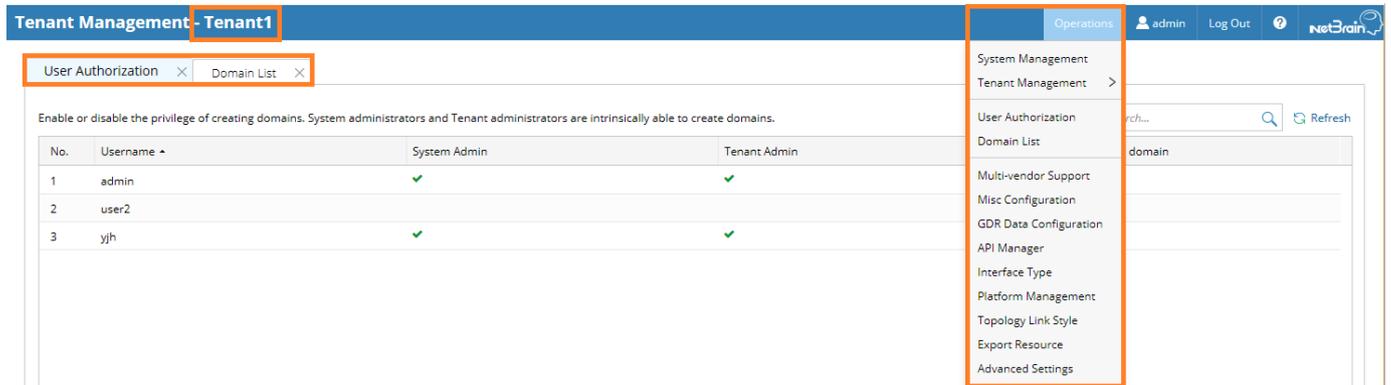
I Ev8.0 adds an All filter to show a full list of domains under all tenants, so that users can get an overall picture and make a quick switchover.

Besides, Max Node Count and Node Usage for each domain are displayed.



## Flexible Menu On/Off

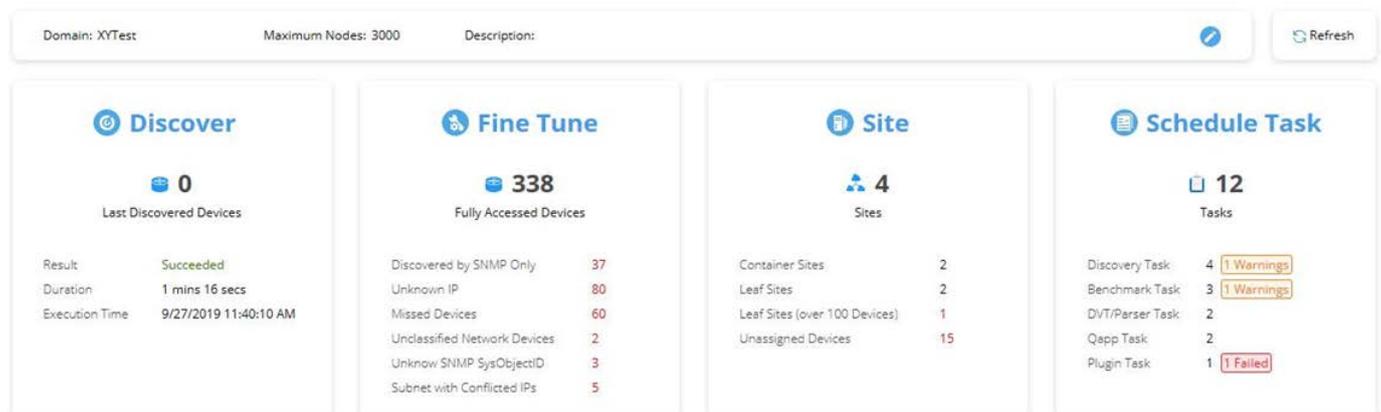
IEv8.0 adds a Close button on each tab of System/Tenant Management for a flexible turn-off. Moreover, the dropdown menu of management operations is sorted by priorities.



The Tenant Management page default to display the User Authorization List and Domain List only, and show the current tenant name in the banner.

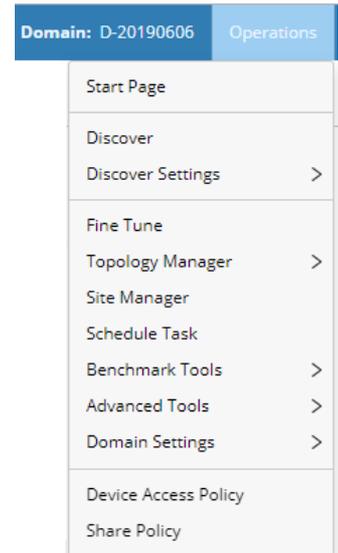
### 4.3.11. Enhancements to Domain Setup/Maintenance Process

To solve the pain point that the domain administrator may not be aware of the problems existing in a domain among too many focuses, the domain setup/maintenance process is optimized in IEv8.0 to provide a summary report about the domain health, including a guidance of to-do that the domain administrator needs to care about and perform.



The redesigned start page of Domain Management lists the most important 4 categories of the domain setup/maintenance process:

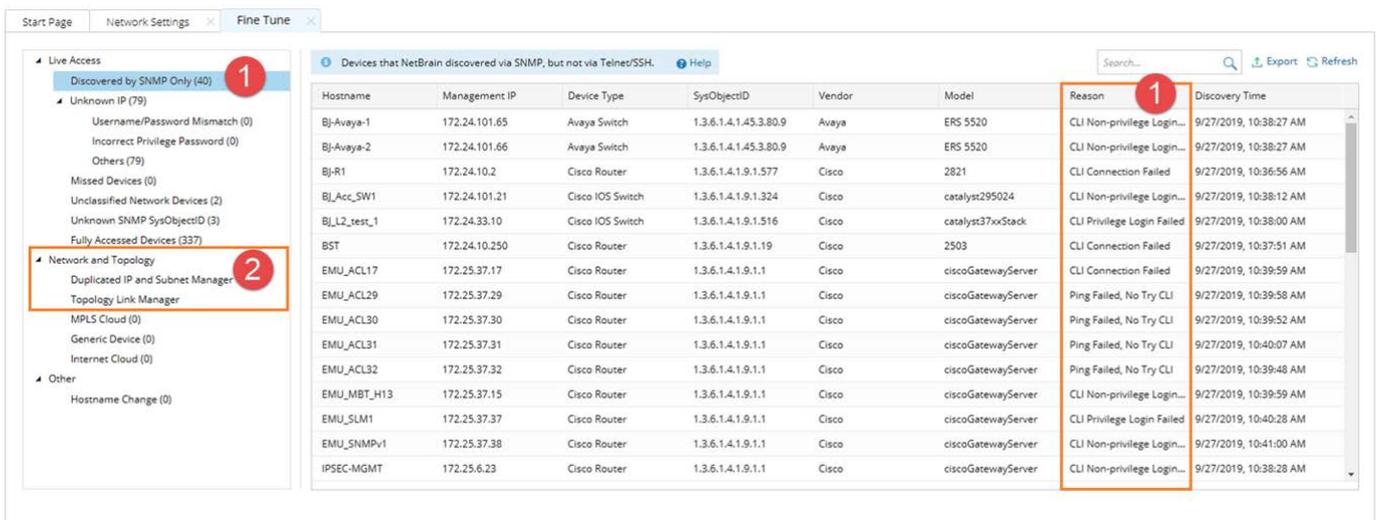
- **Discovery** — shows the record of the last on-demand discovery task performed by the current logged-in domain administrator.
- **Fine Tune** — renamed “Domain Manager”, highlighting the problems that the domain administrator must pay attention to.
- **Site** — highlights the count of leaf sites that contain more than 100 devices and the count of unassigned devices.
- **Schedule Task** — highlights the count of scheduled tasks that the last result may not be healthy.



To help the domain administrator better understand the purpose of domain setup/maintenance operations, the hamburger menu of Operations on the quick access toolbar has been re-organized in I Ev8.0.

## Fine Tune

The “Domain Manager” in previous versions is renamed to “Fine Tune” to help users better understand the feature, introducing the following enhancements:



1. Identify root causes for the category “Discovered by SNMP Only”.

Reason	Access via CLI
Ping Failed, No Try CLI	Unattempted
Don't Support CLI	Not applicable
CLI Connection Failed	Connection Failed

Reason	Access via CLI
CLI Non-Privilege Login Failed	Connected but non-privilege login failed
CLI Privilege Login Failed	Connected but privilege login failed
CLI Configuration Retrieval Failed	Connected and logged in but cannot retrieve configurations.
CLI Configuration Update Failed	Connected and logged in but cannot update configurations.
Others	E.g. "SNMP only" is selected in the Discovery Settings.

2. Add a new category "Network and Topology", and group "Duplicated IP and Subnet Manager" and "Topology Link Manager" inside.

## Categorize Unknown IP by Reason

IEv8.0 splits 10 sub-categories for the Unknown IP category in the Fine Tune module to sort discovered IP addresses by reason.

The screenshot shows the 'Fine Tune' module in NetBrain. On the left, a navigation tree shows the 'Unknown IP (84)' category expanded, with sub-categories like 'Ping Failed, SNMP Failed (47)', 'Ping Succeeded, SNMP Failed (44)', etc. The main table displays a list of IP addresses with columns for IP Address, Source Device, Source Interface, Interface Description, Collection Source, Description, Reason, and Discovery Time. The 'Reason' column shows various failure reasons such as 'Ping Succeeded, SNMP Failed' and 'Ping Failed, SNMP Failed'.

Category (by Reason)	Description
<b>Ping Failed, SNMP Failed</b>	It contains the discovered IP addresses that the system failed to ping or access via SNMP.

Category (by Reason)	Description
<b>Ping Succeeded, SNMP Failed</b>	It contains the discovered IP addresses that the system can ping successfully but cannot access via SNMP.
<b>Don't Support CLI</b>	It contains the discovered IP addresses that don't support Telnet/SSH access.
<b>CLI Connection Failed</b>	It contains the discovered IP addresses that the system failed to access via both Telnet/SSH and SNMP.
<b>CLI Non-Privilege Login Failed</b>	It contains the discovered IP addresses that the system can access via Telnet/SSH but failed to log in.
<b>CLI Privilege Login Failed</b>	It contains the discovered IP addresses that the system can log in via Telnet/SSH in Non-privilege mode, but failed in Privilege mode.
<b>CLI Configuration Retrieval Failed</b>	It contains the discovered IP addresses that the system can log in via Telnet/SSH but failed to retrieve CLI configurations.
<b>CLI Configuration Update Failed</b>	It contains the discovered IP addresses that the system can log in via Telnet/SSH but failed to update the retrieved CLI configurations.
<b>SNMP Configuration Update Failed</b>	It contains the discovered IP addresses that the system can access via SNMP but failed to access via Telnet/SSH and update SNMP configurations fully.
<b>Others</b>	None of the above.

For more details, refer to [online help](#).

## 4.3.12. Enhancements to Benchmark

### Allow to Exclude Device Group from Benchmark

To blacklist specific devices, especially those devices which are accessible or have a slow response time, users can group them first, and exclude this group from basic benchmark task. Another use case is for the devices

with big data such as route tables, and you can create a separate benchmark task for those devices.

Task Name:  Description:

Frequency **Device Scope** Retrieve Live Data CLI Commands Additional Operations after Benchmark Summary

Select Device

All Devices  Device Group  Site

MPLS Cloud(2)  
Router(17)

Exclude Device Group: group1, group2

Select external API servers to retrieve data of SDN nodes

API Source Type	Server Name	Description	Endpoints
<input type="checkbox"/> Cisco ACI	ABCD	Test	https://192.168.10.2:8080
<input type="checkbox"/> VMware vCenter	SDN Controller		http://www.cisco.com/controller

Cancel Submit

## Email Alerts for Benchmark Execution Failures and Config Retrieval Failures

Admin can enable the email alert function for benchmark execution failures or with warnings, or configuration retrieval failures. The full execution log will be attached to the email. By default, the option is disabled.

Task Name: Basic System Benchmark Description: Default system benchmark task

Frequency Device Scope Retrieve Live Data CLI Commands **Additional Operations after Benchmark** Summary

Update Maps Go to Operations > Domain Maintenance > Update Map Manager to see update results.

Enable	Map	Export to Visio
<input type="checkbox"/>	Update Site Maps <a href="#">Select Maps</a>	Yes No <input type="text"/> Browse
<input type="checkbox"/>	Update Shared Device Group Maps <a href="#">Select Maps</a>	Yes No <input type="text"/> Browse
<input type="checkbox"/>	Update Context Maps	Do not support Export to Visio
<input type="checkbox"/>	Update Public Maps <a href="#">Select Maps</a>	Yes No <input type="text"/> Browse

Email Alerts Email alerts to specified users for task execution failures or configuration retrieval failures.

Enable

To:

Cc:

Cancel Submit

**Note:** When task execution failure is caused by system unsteadiness, emails might not be sent because servers are down/off.

## Show Retrieval Success Rate for Configuration Files in Email and Benchmark Log

IEv8.0 adds the success rate of configuration retrieval both in emails and in the benchmark log. The calculation of the success rate only involves legacy network devices.

**Note:** SDN nodes, IP Phones, End Systems and Call Managers are not included in the success rate calculation.

Start Time	Duration	Task Result	Configuration Retrieval Success Rate	Retrieved Data Size (MB)	Retrieved Data Items	Log
27/03/2019, 14:38:02	13 min	Succeeded	93.59% (73/78 devices)	1.35	393	

**Prerequisites:** The “Retrieve Live Data > Configuration File” option is checked in benchmark settings. If not checked or no legacy devices involved, “N/A” will be displayed in this column.

## Show Retrieval Time and Filter Devices with Config Retrieval Failures in Device Log

In the device log of a benchmark task, admin can view the time to retrieve data for each device, and further narrow down the scattered results to only view the devices which have config retrieval failures.

Device Name	Retrieval Time (seconds)	Configuration	Route Table	ARP Table	MAC Table	NDP Table	STP Table	BGP Advertisements
MPLScloudCE1	2	Succeed	N/A	N/A	N/A	N/A	N/A	N/A
bjta002443-SW12	2	Succeed	N/A	N/A	N/A	N/A	N/A	N/A
Qapp-Test1	2	Succeed	N/A	N/A	N/A	N/A	N/A	N/A
IPSEC-MGMT	2	Succeed	N/A	N/A	N/A	N/A	N/A	N/A
EMU_ACL31	2	Succeed	N/A	N/A	N/A	N/A	N/A	N/A
bjta002439-SW10	1	Succeed	N/A	N/A	N/A	N/A	N/A	N/A
EMU_ACL17	1	Succeed	N/A	N/A	N/A	N/A	N/A	N/A
EMU_SLM1	1	Succeed	N/A	N/A	N/A	N/A	N/A	N/A
EMU_UNKNOWNSysDI...	1	Succeed	N/A	N/A	N/A	N/A	N/A	N/A
MPLScloudPE1	2	Succeed	N/A	N/A	N/A	N/A	N/A	N/A

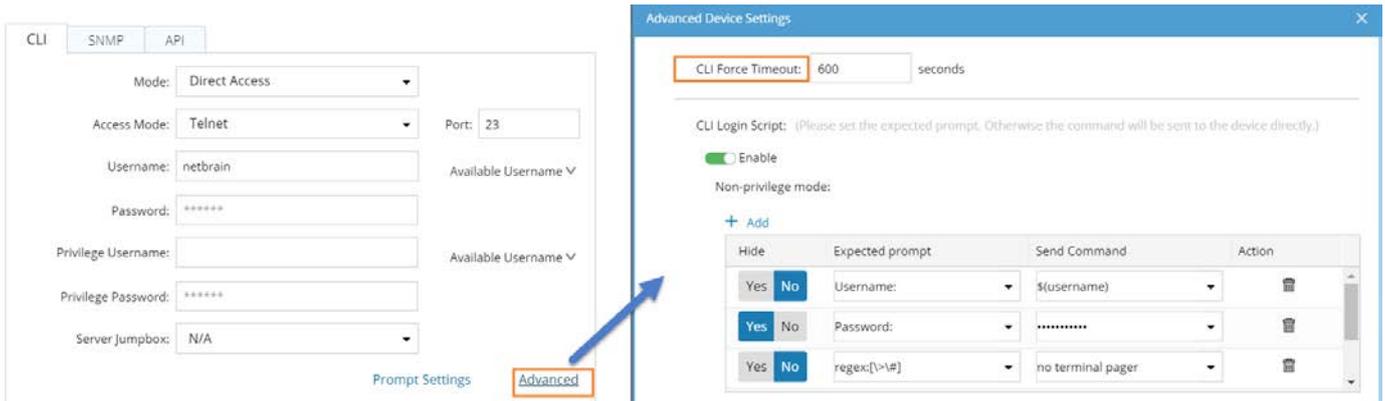
Live Access Log of MPLScloudCE1:

```
14:38:06 Begin data retrieving task
14:38:06 The Device Type of this device does not support live CLI
14:38:08 SNMP get device interface and IP successfully, time:1.47 second(s)
14:38:08 Update configuration file of MPLScloudCE1 successfully,(0.015s)
14:38:08 Began retrieving the NCT Table "Bridging Group"
14:38:08
```

Progress: (Not Running) Close

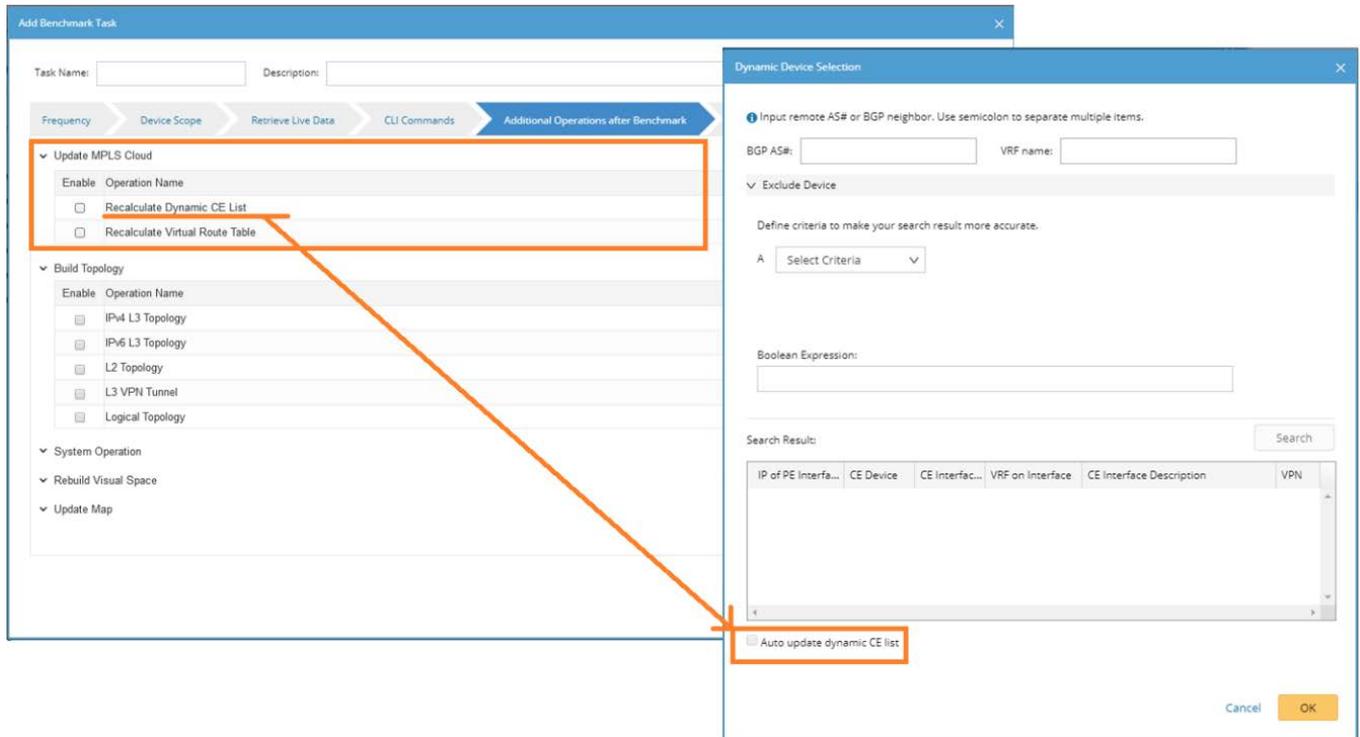
If the system takes too long to retrieve data from a device via CLI for some reasons, such as too large data size, users can decide whether to modify the CLI access timeout for this device. This setting limits the

maximum retrieval time for a single device.



## Visualize the Option to Update MPLS Cloud CE List During Benchmark

IPv8.0 adds a new option “Recalculate Dynamic CE List” as additional operations during the benchmark and makes the “Update MPLS Cloud” function optional, which can be enabled/disabled by users.

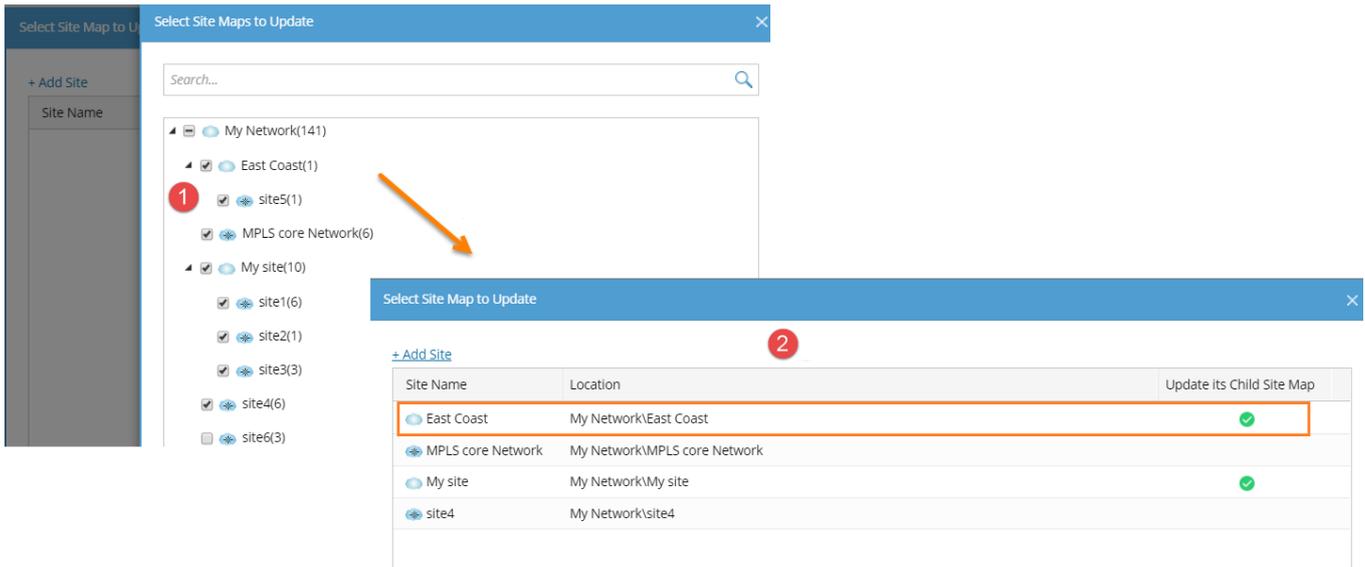


**Note:** It doesn't correlate to any MPLS Clouds added to benchmark device scope, but it will have an impact on MPLS Clouds throughout the domain, as long as the “Auto update dynamic CE list” option is checked for an MPLS Cloud.

## Auto-Update Site Maps for Leaf Sites Through Benchmark

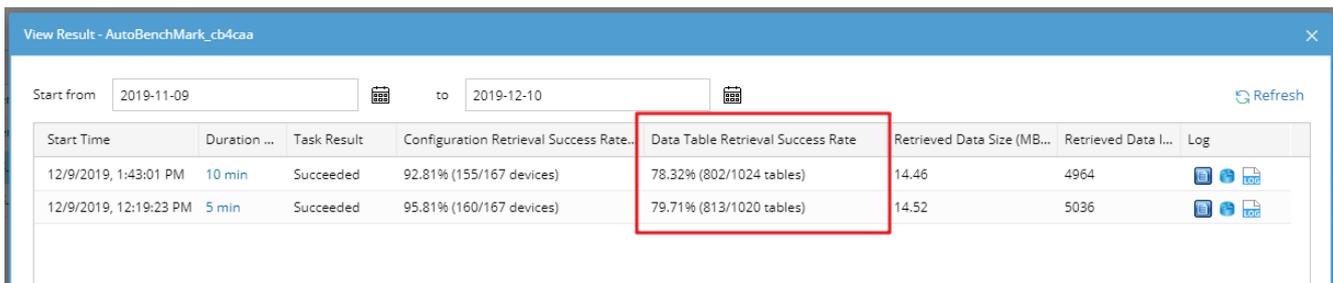
Site maps can be selected to update to sync with benchmark results. In previous versions, if a selected container site has newly added leaf sites, the site maps for those leaf sites would not be involved in the update scope.

IEv8.0 adds the capability to sync child site changes for site map updating.



## Show Data Table Retrieval Success Rate in Benchmark Report

To provide a hint for path analysis, IEv8.0 adds a column for the retrieval success rate of data tables to each benchmark result, including system tables and network control tables (NCT).



## Enhanced Discovery/Benchmark Execution Log

To enable the lookup for the most time-consuming sub-task in a discovery/benchmark task, IEv8.0 separates a new column for Total Time Spent in the task execution log. This column shows the time spent on each sub-

task and can be sorted by value size.

Date & Time	Messages	Total Time Spent
12/9/2019, 1:43:01 PM	Begin: retrieve devices data.	
12/9/2019, 1:50:33 PM	End: retrieve devices data.	0 hrs 7 mins 32 secs
12/9/2019, 1:50:33 PM	There are no MPLS Cloud devices in your domain.	
12/9/2019, 1:50:33 PM	Begin:build topology	
12/9/2019, 1:50:33 PM	Try to build topology IPv4 L3 Topology	
12/9/2019, 1:50:57 PM	End: build IPv4 L3 Topology with 417 links.	0 hrs 0 mins 24 secs
12/9/2019, 1:50:57 PM	Try to build topology IPv6 L3 Topology	
12/9/2019, 1:51:04 PM	End: build IPv6 L3 Topology with 36 links.	0 hrs 0 mins 6 secs
12/9/2019, 1:51:04 PM	Try to build topology L2 Topology	

## Enhanced Benchmark Device Log

IEv8.0 introduces the following usability enhancements to the benchmark device log:

Device Name	Device Type	Retrieval Time (seconds...)	Configuration	Route Table	ARP Table ...	MAC Table	NDP Table...
sw2960-121	Cisco IOS Switch	9	Succeeded via SNMP	Failed	Succeeded	Failed	Succeeded
sw2960-105	Cisco IOS Switch	9	Succeeded via SNMP	Failed	Succeeded	Failed	Succeeded
sw2960-107	Cisco IOS Switch	9	Succeeded via SNMP	Failed	Succeeded	Failed	Succeeded
sw2960-106	Cisco IOS Switch	9	Succeeded via SNMP	Failed	Succeeded	Failed	Succeeded
FLEX-MGMT	Cisco IOS Switch	11	Succeeded via CLI	Succeeded	Succeeded	Succeeded	Succeeded
sw2960-130	Cisco IOS Switch	10	Succeeded via SNMP	Failed	Succeeded	Failed	Succeeded
VRF-CE4	Cisco Router	14	Succeeded via CLI	Succeeded	Succeeded	Failed	Succeeded
BJ-R1	Cisco Router	11	Succeeded via SNMP	Failed	Succeeded	Failed	Succeeded
Multicast-R1	Cisco Router	14	Failed	Succeeded	Succeeded	Failed	Succeeded
IPSEC-Router	Cisco Router	15	Succeeded via CLI	Succeeded	Succeeded	Failed	Succeeded

Live Access Log of sw2960-130:

```

13:43:08 Begin data retrieving task
13:43:08 Prepare retrieving command.
13:43:08 Can not Telnet/SSH to the device.
13:43:13 Retry to get device command
13:43:13 Can not Telnet/SSH to the device.
13:43:13 Begin to retrieve NDP table via SNMP.
  
```

- Add device icons in the column of Device Name and add a column for Device Type to distinguish the vendor/model.
- Freeze the two columns: Device Name and Device Type to ensure users won't be lost when scrolling

horizontally.

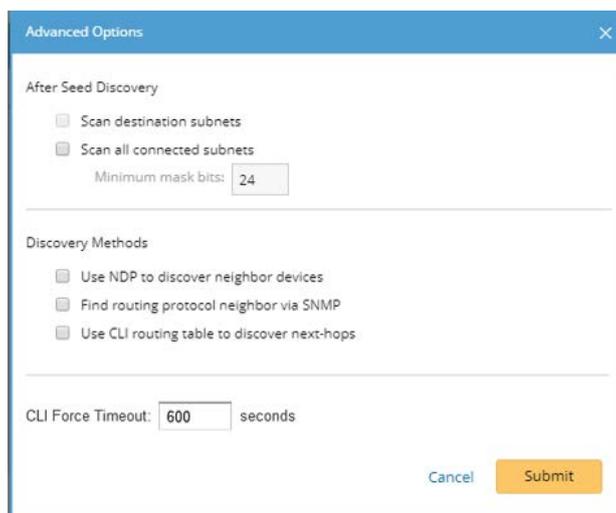
- Change the font color for failures to red.

### 4.3.13. Enhancements to Discovery

#### Add CLI Force Timeout for New Devices in Discovery Settings

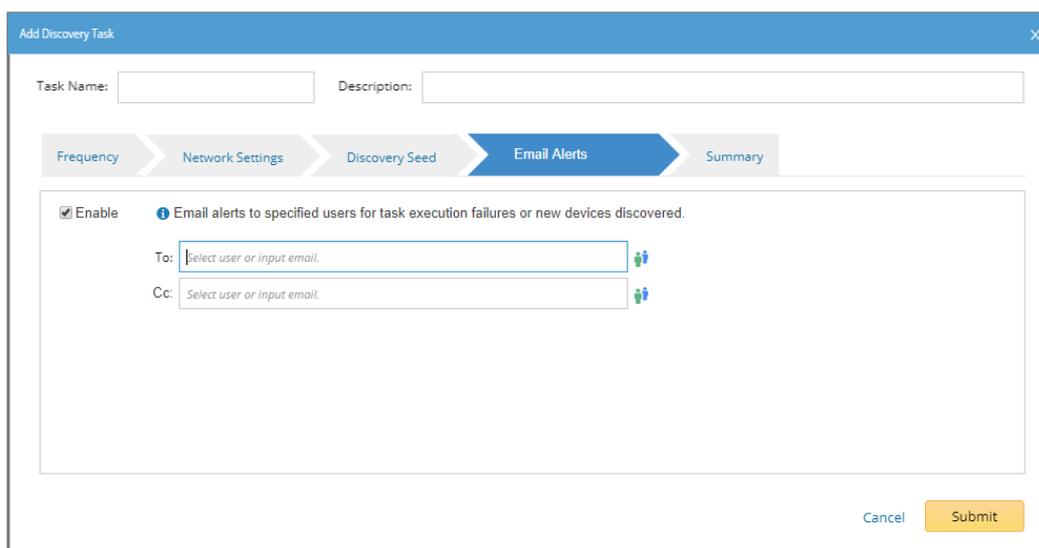
To limit the time spent for retrieving live data of each new device via CLI commands, admin can customize the CLI Force Timeout setting. The default value is 600 seconds.

**Note:** This setting for discovery tasks has a lower priority than that individual setting at the device level. Moreover, its enablement may differ based on different discovery modes, such as Seed Router or Scan IP.



#### Email Alerts for Discovery Execution Failures and Newly Discovered Devices

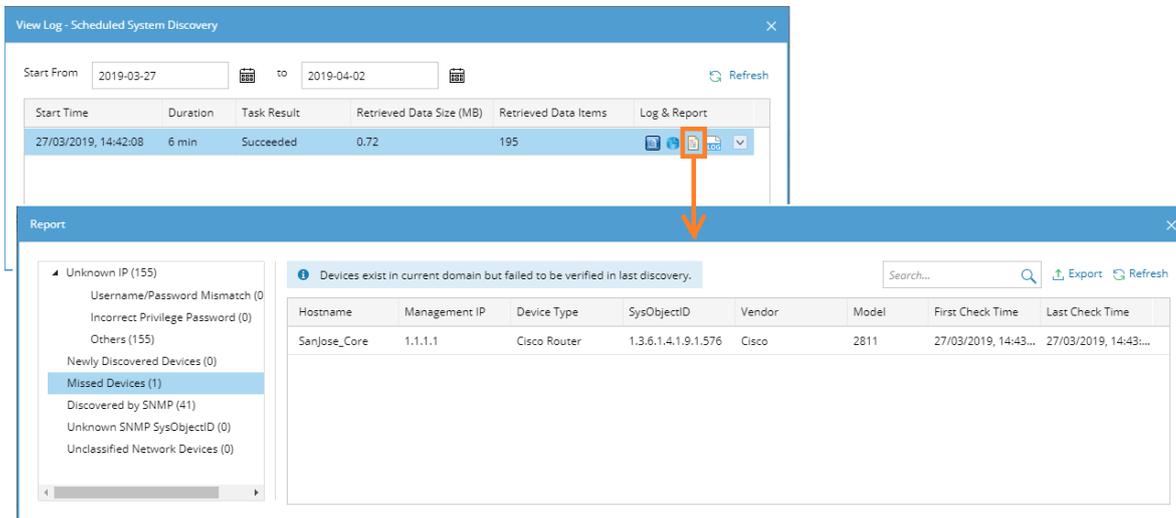
Admin can enable the email alert function for either discovery execution failures or newly discovered devices, or both. The full execution log or new device list will be attached in the email.



**Note:** When task execution failure is caused by system unsteadiness, emails will not be sent because servers are off.

## View Newly Discovered Devices and More Reported Issues

Admin can directly check the discovery report when viewing discovery logs. The report is only for the current discovery, but the categories are almost the same with Domain Manager (provides reports for all historical discovery tasks), except for “MPLS” and “Hostname Change”.



The screenshot shows two windows. The top window, titled "View Log - Scheduled System Discovery", has a date range from 2019-03-27 to 2019-04-02. It contains a table with the following data:

Start Time	Duration	Task Result	Retrieved Data Size (MB)	Retrieved Data Items	Log & Report
27/03/2019, 14:42:08	6 min	Succeeded	0.72	195	[Icons]

An orange arrow points from the "Log & Report" icon in the second row to the "Report" window below. The "Report" window has a left sidebar with a tree view containing categories like "Unknown IP (155)", "Missed Devices (1)", etc. The main area shows a message: "Devices exist in current domain but failed to be verified in last discovery." Below this is a table with the following data:

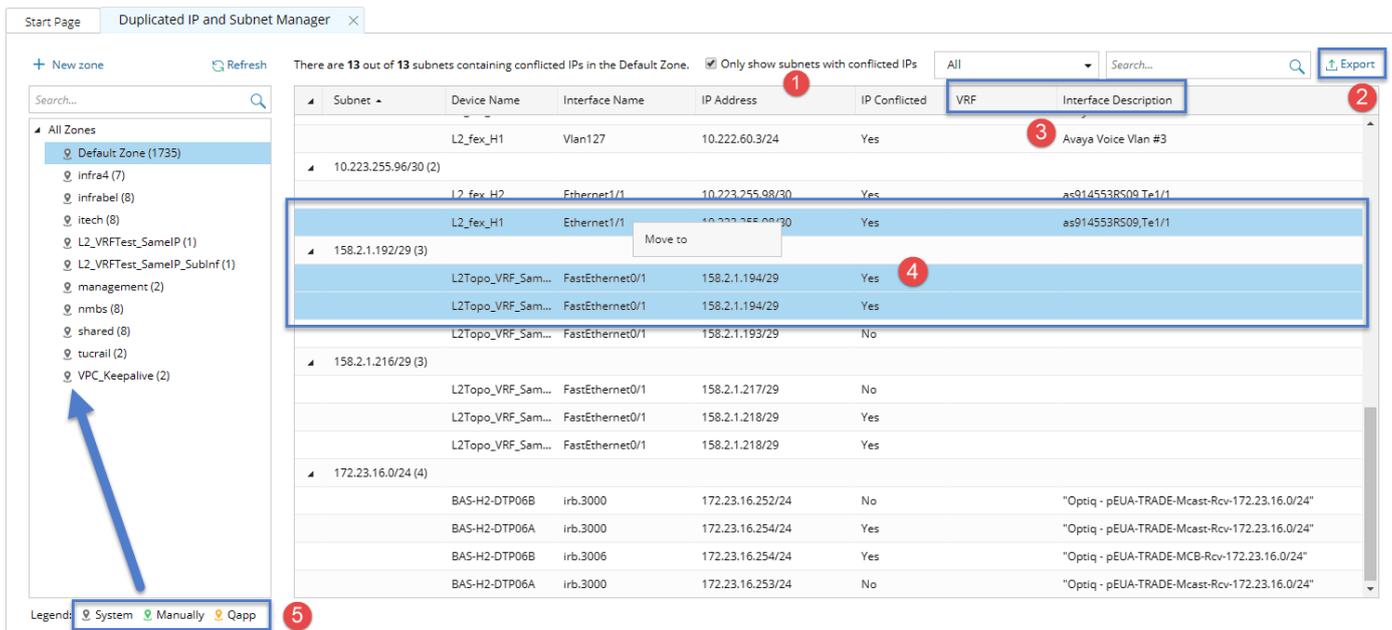
Hostname	Management IP	Device Type	SysObjectID	Vendor	Model	First Check Time	Last Check Time
SanJose_Core	1.1.1.1	Cisco Router	1.3.6.1.4.1.9.1.576	Cisco	2811	27/03/2019, 14:43...	27/03/2019, 14:43...

### 4.3.14. Enhancements to Duplicate IP and Subnet

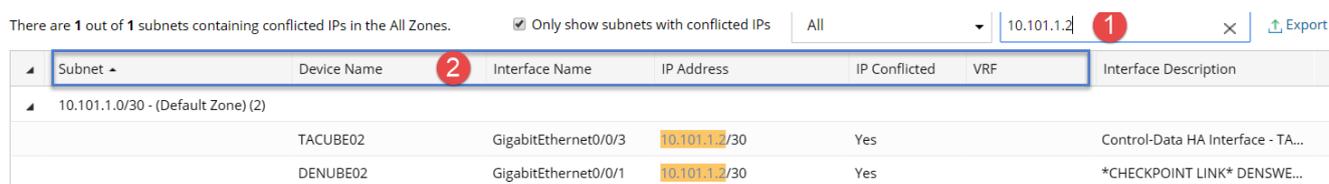
Duplicate IP and Subnet Manager is used to manage the interfaces configured with the same IP address as well as the subnets in a NetBrain domain. With this feature, users can enable all duplicated IPs to join the Ipv4 L3 topology link calculations by moving interfaces with duplicated IPs to different zones.

## UX/UI Enhancements and New Functions

IEv8.0 introduces many enhancements to Duplicated IP and Subnet Manager from a UX/UI perspective, and adds a new function “Export to CSV”.



1. **Show Subnets and Duplicate IPs in one Page** — In IEv8.0, “All Zones” is added as the root category. When users select “All Zones”, all subnets and duplicate IPs in a domain are displayed, so that users can understand their relationships at one glance. To quickly look up subnets with conflicted IPs, enable the filter “Only show subnets with conflicted IPs”.
2. **Export to CSV** New Function — With one-click, users can export information of zones from the NetBrain system to a CSV file.
3. **Display VRF and Interface Description** — VRF and interface columns are added for users to understand the VRF and descriptions of an interface.
4. **Batch Operations** — Move multiple interfaces into a zone or delete multiple zones at one time.
5. **Optimize the Search Function** — In IEv8.0, the search function is performed throughout the whole database rather than lazy-loading. Also, users can search for contents within any column.



6. **Display Zone Types** — Per generation methods, zone types include System (built-in), Qapp and manually added. Each zone type has its color legend for users to distinguish in the tree list.

## Miscellaneous Improvements

- **Optimize Interface Display**

In previous versions, an interface is displayed as **IPv4 + interface** form, e.g., **192.168.3.1 e0/1**. Users are

not able to search based on the interface name or IP. I Ev8.0 displays the physical interface name and IP address separately.

- **Auto Delete Empty Zones Created by System**

Whenever a system zone is auto-created, I Ev8.0 checks all zones created by the system and auto delete the zones having no interface.

- **Auto Refresh Zones**

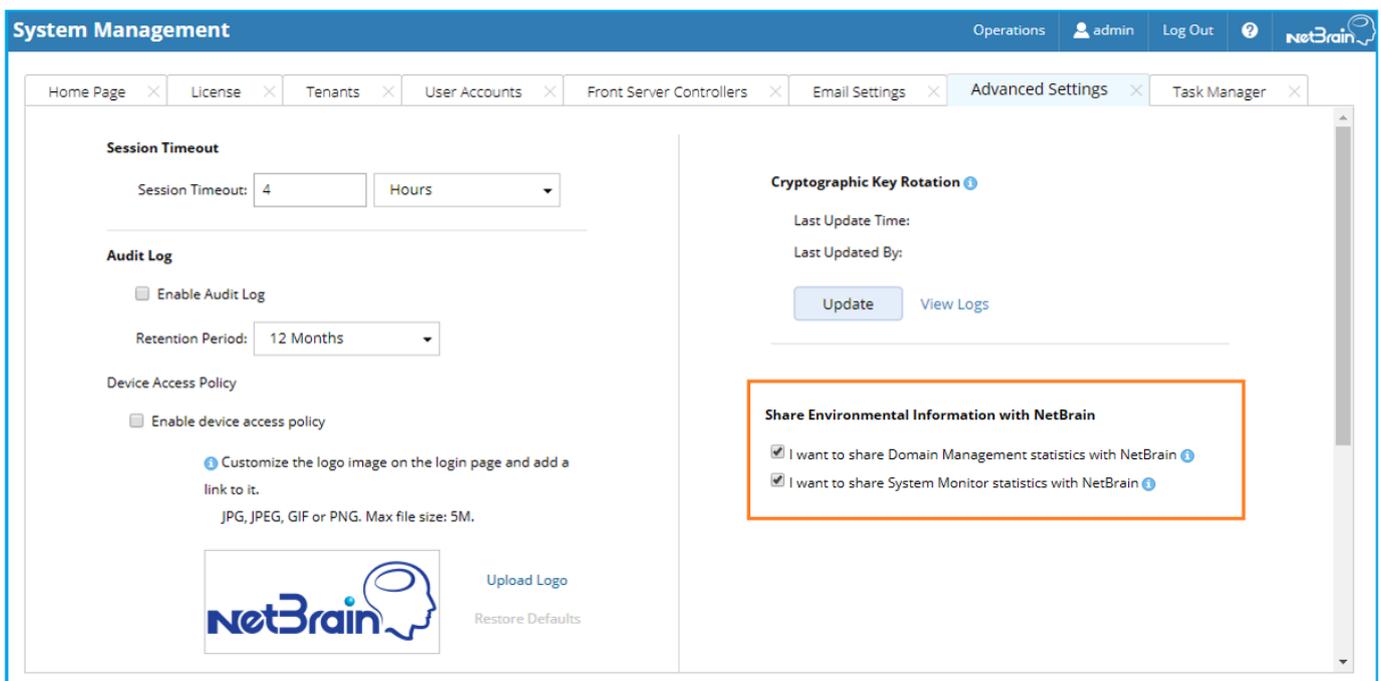
After users move a duplicated IP to another zone, the system automatically detects if there are still duplicate IPs in a zone. Moreover, the feature provides a Refresh button for users to refresh the conflicted IP display manually.

### 4.3.15. Collect Usage Log

To help NetBrain engineers better assist customers with more solid statistics, I Ev8.0 adds options to send the NetBrain feature usage information via logs to the NetBrain team.

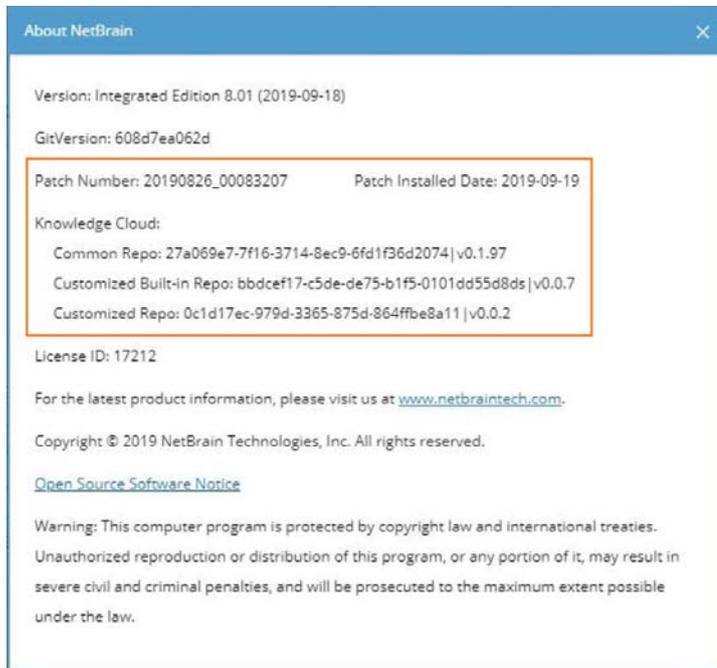
- **I want to share Domain Management statistics with NetBrain** — send the usage statistics about Tenant and Domain to NetBrain, including device discovery and advanced feature metrics.
- **I want to share System Monitor statistics with NetBrain** — send the usage statistics about Service Monitor to NetBrain, including operating system information and the health of NetBrain services.

By default, these two options are enabled.



## 4.3.16. Patch Version Management

To better manage the patch release and also elevate the troubleshooting process, I Ev8.0 adds both the patch version number and Knowledge Cloud version number to the About NetBrain dialog.



**Note:** Patch version number will also be recorded in the [Usage Log](#) to share with NetBrain.

## 4.4. Enhancements to System Security

### 4.4.1. New Keystore with Enhanced Hashing and Encryption Algorithms

The proper management of cryptographic keys is essential to the effective use of cryptography for security. Securely storing and retrieving these keys as needed is a major security enhancement.

To address a significant FIPS requirement and to enhance the solution's security, I Ev8.0 builds a new keystore in the database, as a repository to store cryptographic keys, and also adopts enhanced hashing and encryption algorithms.

Algorithm	Used in IEv7.x	Adopted in IEv8.0
Non-Cryptographic Hashing	MD5	SHA256 Spooky 128
Password Hashing	MD5/SHA256	PBKDF2
Encryption/Decryption	DES	AES-256-CBC

**Note:** This upgrade of hashing and encryption algorithms has backward compatibility with user data in IEv7.x, except for Network Settings. A convert tool can be used to adapt existing Network Settings to IEv8.0.

---

#### 4.4.2.Enforce Strong Password Policy

To address security requirements, stronger password policy is enforced in IEv8.0, including:

- Enforce “Require Password Change at First Login” for users whose accounts are created by admin.
- Enforce “Password cannot be the same as username” and “Password must meet at least three requirements”.
- Default to enable “New password can only contain at most 2 consecutive characters of the old one”.

To configure these settings, go to **System Management > User Accounts > Password Policy**.

The screenshot shows the 'System Management' interface with the 'User Accounts' section selected. The 'Password Policy' tab is active, displaying the following configuration options:

- Minimum password length:  characters (6-128 characters)
- Password expires after  days
- New password can only contain at most 2 consecutive characters of the old one

Password must meet at least three of the following requirements:

- Includes uppercase letters (A - Z)
- Includes lowercase letters (a - z)
- Includes a number (0 - 9)
- Includes a non-alphabetic character (such as ! \$ # %)

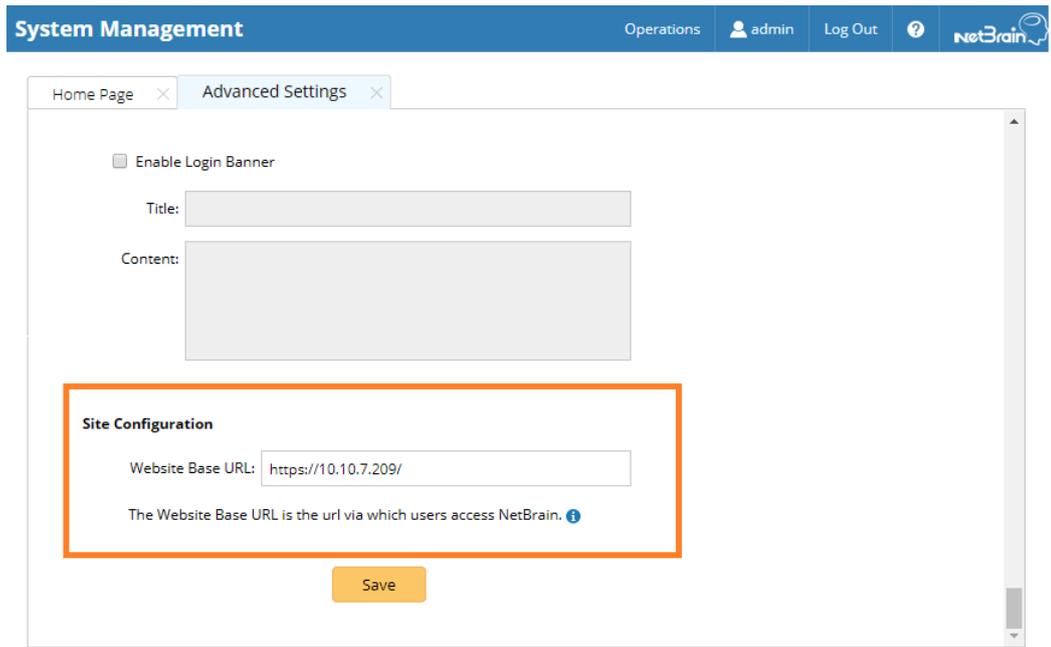
Password cannot be same as username

### 4.4.3. Single Source for Website Base URL

In previous versions, when users wanted to share a map or reset a password, the system provided an assembled URL for redirection by directly obtaining the domain URL from the client's web browser cache, which may have potential security risks.

To fix this vulnerability, IEv8.0 enforces the definition for a base URL and uses it as a single source for website URL assembly. Every time a user requests a website URL, the system uses the predefined base URL to assemble it, rather than obtaining the base URL from the user's web browser cache.

IEv8.0 guides the system administrator upon login to configure the base URL at **System Management > Advanced Settings > Website Base URL**.



#### 4.4.4. Upgrade Third-Party Components to the Latest Versions

To ensure the longevity of support and the most up-to-date code from a security standpoint, many components have been upgraded to the latest version in the IEv8.0 release.

Component	Old Version	Updated Version in IEv8.0
MongoDB	3.6.4	4.0.6
Elasticsearch	6.0.0 6.5.2 (v7.1a2)	6.7.2
Redis	3.0.504	5.0.4
RabbitMQ	3.7.7	3.8.1
Python	3.6.2	3.7.5
JDK	JDK1.8.131 OpenJDK 11 (v7.1a2)	OpenJDK 12.0.1
Gojs	1.6.9	2.0.18

Component	Old Version	Updated Version in IEv8.0
Node.js	8.2.1	9.3.0
OpenSSL (Windows + Linux)	1.0.2p	1.0.2t
psutil	5.5.1	5.6.7

---

#### 4.4.5.Allow the Deletion of Built-in Admin Account

Privileged accounts may pose potential security risks if not managed. They usually have broad access to underlying customer information that resides in applications and databases. And passwords for these accounts are often embedded and stored in unencrypted text files, a vulnerability that is replicated across multiple servers to provide greater fault tolerance for applications.

To eliminate this risk, IEv8.0 allows deleting the default administrator account.

**Note:** Before the deletion of the admin account, make sure there is at least one active user account with user management privilege in the system.

---

#### 4.4.6.Start Services with Restricted Privileges

To reduce the risk of elevated privileges, IEv8.0 enforces to launch NetBrain related services with restricted privileges when interacting with both Windows and Linux. Startup accounts with restricted privileges will be either created or configured during the system installation, rather than using privileged accounts of operating systems.

---

#### 4.4.7.Lock Accounts After Too Many Unsuccessfully Attempts of Password Reset

In previous versions, user accounts with too many unsuccessfully login attempts would be locked, to protect user information confidentiality. However, the vulnerability still existed in the Password Reset function.

IEv8.0 adds the same control to the Password Reset function. When users are attempting to reset their passwords via GUI or API calls, entering incorrect passwords for too many times will lock their user accounts to protect user-information confidentiality.

**Note:** By default, locked user accounts will be available again in 1 hour. In previous versions, the default value is 12 hours.

---

#### 4.4.8.Limit Guest User's Privilege

By default, the system only grants Guest Users with limited access to system resources by disabling their privilege of Shared Resource Management. In previous versions, the privilege of Shared Resource Management only covered most resource types in the system, such as Qapps, Gapps, Parsers, Runbook Templates, etc., but not all of them. It might cause potential security risk that Guest Users can modify or delete public resources, such as public maps and A-B paths.

To limit Guest User's privileges, IEv8.0 expands the privileged scope for Shared Resource Management to cover more resources, including:

- One-IP Table
- Inventory Report
- Device Property
- CLI Commands Template
- My Files/Desktop
- Path Browser Pane
- Network Pane
- Application Manager
- Static Data View Manager
- Context Action Manager
- System Automation Task Manager

---

#### 4.4.9.Validate Data Access Privilege for Extended Device Scope

The device access policy introduced in the previous version allows specified users to have the privilege of data access to specified devices. However, it might be out of control when it comes to Qapp execution because the

device scope of Qapp might be expanded to neighbor devices depending on the device input settings for a Qapp.

To fix this vulnerability, IEv8.0 starts to validate the data access privilege on the extended device scope for Qapp execution. If the executor is not authorized to view the device data for any extended neighbor device, the system will not run Qapp on that device.

#### 4.4.10. Minor Enhancements

- Encrypt the configuration keys for “requirepass” and “masterauth” to securely store Redis credentials.
- Update the search guard to use appropriate cipher suites only.
- Remove the support for older versions of TLS (before TLS 1.2, including SSL).
- Remove all instances of clear-text and hard-coded passwords.
- Clear Browser Temporary Data after User Logout.  
**Note:** Refreshing webpage will not clear browser temp data.
- Disable Insecure Communication Protocols (SSL and TLS < 1.2) Unless Required.
- Protect Redis Sentinel by Authentication and Rename Default Command.
- Remedy Aged Libraries for Redis.
- Fix REST API Vulnerability about Roles and Privileges.

### 4.5. Performance Enhancements

Compared to IEv7.1, the performance improvements realized in IEv8.0 are listed as follows.

Feature	Test Scenario	In IEv7.1	In IEv8.0
Search	Search results include 3000 maps and each map has 1~3 pages, including 50 devices with notes	11~13 seconds	1~3 seconds
Semantic Search	E.g., search for “OSPF 1 AREA 0” with 20 out of 646 devices returned	151 seconds	30 seconds
MPLS Dynamic Search	Search results include 724 CE devices and 1.3k interfaces	4.1 hour	0.1 hour

Feature	Test Scenario	In IEv7.1	In IEv8.0
Data View	Build Default Data Views for 10k interfaces	219 minutes	1.5 minutes
Map	Auto link 50 devices on a new map	3.27 seconds	0.6 seconds
	Extend 19k neighbors for a device	283 seconds	24 seconds
Schedule Map Update	Device Group Map #OSPF 10 with 2k+ devices	485 seconds	140 seconds
	Device Group Map #OSPF 20 with 2k+ devices	513 seconds	31 seconds
	Device Group Map #OSPF 100 with 749 devices	619 seconds	25 seconds
Rebuild Site	10+ sites, 130k devices in total, with many end systems	180 minutes	2 minutes
Path	Calculate 5000 complex paths (up to 28 hops) with cache data	325 minutes	61 minutes
Path Gateway	45k devices and 3415k interfaces	2~30 seconds (depending on cases)	2~3 seconds
Big Data	Parse system NCT table exceeding 64MB	Not supported	Support of parsing ARP/MAC table exceeding 64MB
	Build L2 topology for 45k devices	3.7 hours	1.9 hours
	Build L2 topology for 90k devices	29 hours	6.5 hours
	Build topology for 10k interfaces	Not supported	Supported
	Build L3 topology for 10k devices and 463k interfaces, containing 100k+ Class A IP addresses (without mask) connected to a single media <b>Note:</b> The <b>Use the main class mask to calculate L3 topology for an IP without mask</b> option is checked.	18 hours	0.21 hour
	Extend neighbors for a device with 250k ARP entries	Failed after 15 minutes loading	2 minutes
	Update a map with 2 devices and 10k interfaces	810 seconds	1 second

Feature	Test Scenario	In IEv7.1	In IEv8.0
	Retrieve live data for 10k GRE Tunnels + 10k IPsec Tunnels	102 minutes	1.5 minutes
	250k ARP table entries + 250k MAC table entries + 10k NDP neighbors + 100MB routes + big config (10k interfaces)	76 minutes	2 minutes
<b>Parser</b>	Retrieve configuration file (65MB)	420 seconds	45 seconds
<b>Execute CLI Commands</b>	Retrieve routes exceeding 100MB	34 seconds	26 seconds
	Retrieve and parse routes exceeding 100MB	114 seconds	72 seconds
<b>CheckPoint Firewall</b>	Benchmark 2 CheckPoint firewalls configured with 500 policies, 500 NAT, and 4000k IPsec VPN	82 minutes	5~10 minutes
<b>File Upload</b>	Upload 2k Data View Template files.	Failed	Succeeded
<b>Load Site Tree</b>	Load a tree with 9000 sites (5 devices per site) in: <ul style="list-style-type: none"> <li>▪ Site Panel</li> <li>▪ Site Manager</li> <li>▪ Dynamic Search</li> <li>▪ Select Device/Interface</li> <li>▪ Inventory Report</li> </ul>	3~4 minutes or failed	1~3 seconds
<b>Load Device Group Tree</b>	Load a tree with 3000 device groups (50 devices per group) in: <ul style="list-style-type: none"> <li>▪ Device Group Panel</li> <li>▪ Dynamic Search</li> <li>▪ Select Device/Interface</li> <li>▪ Network Panel</li> <li>▪ Plugin Manager</li> <li>▪ Search Result</li> <li>▪ Add to Device Group</li> <li>▪ Scheduled Task</li> <li>▪ Tune</li> </ul> <p><b>Note:</b> This improvement is not significant in <b>Shared Device Settings &gt; Apply to Device Group.</b></p>	5~20 seconds or failed	1 second
<b>SDN</b>	Preview context map in Fabric Pod View: <ul style="list-style-type: none"> <li>▪ Legacy Device Count: 20k</li> <li>▪ ACI Tenant Count: 1.2k</li> <li>▪ ACI BD Count: 4.8k</li> <li>▪ ACI VRF Count: 4.8k</li> <li>▪ ACI EPG Count: 144k</li> <li>▪ ACI ANP Count: 48k</li> </ul>	18 seconds	8.5 seconds

Feature	Test Scenario	In IEv7.1	In IEv8.0
	Preview context map of ANP node in Application Centric View: <ul style="list-style-type: none"> <li>▪ Legacy Device Count: 20k</li> <li>▪ ACI Tenant Count: 1.2k</li> <li>▪ ACI BD Count: 4.8k</li> <li>▪ ACI VRF Count: 4.8k</li> <li>▪ ACI EPG Count: 144k</li> <li>▪ ACI ANP Count: 48k</li> </ul>	27 seconds	9 seconds
	Load a Network Tree <ul style="list-style-type: none"> <li>▪ ACI Tenant Count: 876</li> <li>▪ ACI BD Count: 4k+</li> <li>▪ ACI VRF Count: 4k+</li> <li>▪ ACI EPG Count: 4k+</li> <li>▪ ACI ANP Count: 1k</li> <li>▪ ACI EP Count: 142k</li> </ul>	<ul style="list-style-type: none"> <li>▪ The loading speed in IEv8.0 is 1.6 times faster than IEv7.1</li> <li>▪ The consumed memory in IEv8.0 is 40%~60% lower than IEv7.1</li> </ul>	

**Note:** The above comparison test between IEv7.1a1 and IEv8.0 was performed with the same hardware spec.

## 5. Appendix

### 5.1. Built-in Data View Templates in Detail

The supporting variables and drill-down actions defined in each built-in data view template are listed in the following table, including branch conditions.

Category	Built-in DVT Name	Branch Criteria	Supporting Variables	Drill Down Actions	
Cisco ACI	Fabric Health and Faults [Cisco ACI]	<ul style="list-style-type: none"> <li>- Legacy Device</li> <li>- Device Type contains Cisco ACI APIC; Cisco ACI Leaf Switch; Cisco ACI Spine Switch</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- fault_details<sup>1)</sup></li> <li>- fault_critical</li> <li>- health_timestamp</li> <li>- health_change</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- <a href="#">View the Cisco APIC Faults Online Guide<sup>1)</sup></a></li> </ul>	
	Fabric Underlay Connections [Cisco ACI]	<ul style="list-style-type: none"> <li>- Legacy Device</li> <li>- Device Type contains Cisco ACI APIC; Cisco ACI Leaf Switch; Cisco ACI Spine Switch</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- LLDP_Detail</li> <li>- CDP_Detail</li> <li>- port_channel_detail</li> <li>- vpc_detail</li> </ul>	Template Level: <ul style="list-style-type: none"> <li>- <a href="#">View the Introduction to Cisco ACI</a></li> </ul>	
	Logic Node View [Cisco ACI]	Branch: BD	<ul style="list-style-type: none"> <li>- Legacy Device</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- tn_name</li> <li>- bd_mac</li> <li>- subnets</li> </ul>	Template Level: <ul style="list-style-type: none"> <li>- <a href="#">View the ACI Policy Model</a></li> </ul>
		Branch: Contract	<ul style="list-style-type: none"> <li>- Contract</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- provider_epg</li> <li>- consumer_epg</li> <li>- contract_detail</li> <li>- filters_detail</li> </ul>	
Branch: EPG		<ul style="list-style-type: none"> <li>- EPG</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- tn_name1</li> <li>- ap_name</li> <li>- ep_detail</li> </ul>		
Branch: L2Out			Device Level:		

Category	Built-in DVT Name	Branch Criteria	Supporting Variables	Drill Down Actions
		- L2Out	- BD_extended - node_intf_table1	
		Branch: L3Out - L3Out	Device Level: - ospf_info - node_intf_table	
High Availability	HSRP Overview	- Legacy Device (IPv4 Interface) - Config File contains 'standby' and Device Type contains Cisco IOS Switch; Cisco Router	Device Level: - hsrp_groups <sup>1)</sup> - ip_int_brief <sup>2)</sup> Interface Level: - group1 - authentication - priority	Template Level: - Compare the Configuration File in two data sources  - <a href="#">Understand and Troubleshoot HSRP Problems in Catalyst Switch Networks</a>  - View the Spanning Tree Overview  Device Level:  - Execute CLI Commands <b>show standby</b> ; Ping the device <sup>1)</sup>  - Execute CLI Commands <b>show ip int brief</b> and <b>show ip int   in line   list</b> ; View the ARP Table <sup>2)</sup>
Quality of Service	QoS Overview	- Legacy Device (IPv4 Interface; Physical Interface) - Config File contains 'service-policy' and Config File contains 'policy-map' and Device Type contains Cisco IOS Switch; Cisco Router	Device Level: - service_policy <sup>1)</sup> - class_map <sup>2)</sup> - mls_qos_intfs Interface Level: - direction	Template Level: - Compare the Configuration File in two data sources  - <a href="#">View Enterprise QoS Design</a>  Device Level:  - Execute CLI Commands <b>show policy-map</b> <sup>1)</sup>  - Execute CLI Commands <b>show class-map</b> <sup>2)</sup>
Routing	EIGRP Overview	- Legacy Device (IPv4 Interface) - Config File contains 'router eigrp' and Device Type contains Cisco IOS Switch; Cisco Router	Device Level: - eigrp_neighbor_detail <sup>1)</sup> - eigrp_intfs (paragraph) <sup>2)</sup> - RouteTable <sup>3)</sup>	Template Level: - Compare the Configuration File in two data sources  - Show log include INPUT [Cisco IOS NXOS ASA]  - View the <a href="#">EIGRP Troubleshoot Guide</a>

Category	Built-in DVT Name	Branch Criteria	Supporting Variables	Drill Down Actions
			Interface Level: <ul style="list-style-type: none"> <li>- peers (int)</li> <li>- split_horizon (string)</li> <li>- pending_routes (string)</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- Execute CLI Commands <b>show ip eigrp neighbor detail</b>; Ping the device; View the MAC Table<sup>1)</sup></li> <li>- View the ARP Table; Execute CLI Commands <b>show ip eigrp interface</b><sup>2)</sup></li> <li>- Execute CLI Commands <b>show ip eigrp topology</b><sup>3)</sup></li> </ul>
	IP BGP Overview	<ul style="list-style-type: none"> <li>- Legacy Device</li> <li>- Config File contains 'router bgp' and Device Type contains Cisco IOS Switch; Cisco Router</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- rid</li> <li>- bgp_neighbors<sup>1)</sup></li> <li>- bgp_filter_detail<sup>2)</sup></li> <li>- bgp_rib_failure<sup>3)</sup></li> </ul>	Template Level: <ul style="list-style-type: none"> <li>- Compare the Configuration File in two data sources</li> <li>- Show log include INPUT [Cisco IOS NXOS ASA]</li> <li>- <a href="#">Troubleshoot BGP</a></li> </ul> Device Level: <ul style="list-style-type: none"> <li>- Ping the device; Use Traceroute to discover and map a routing path; Execute CLI Commands <b>show ip bgp neighbors, show tcp brief all</b><sup>1)</sup></li> <li>- Execute CLI Commands <b>show ip bgp neighbors, show access-list, show ip prefix-list, and show route-map</b><sup>2)</sup></li> <li>- <a href="#">Understand BGP RIB-failure</a><sup>3)</sup></li> </ul>
	IP BGP Prefix Instance	<ul style="list-style-type: none"> <li>- Legacy Device</li> <li>- Config File contains 'router bgp' and Device Type contains Cisco IOS Switch; Cisco Router</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- bgp_paths<sup>1)</sup></li> <li>- bgp_filter_detail<sup>2)</sup></li> <li>- RouteTable</li> </ul>	Template Level: <ul style="list-style-type: none"> <li>- Compare the Configuration File in two data sources</li> <li>- <a href="#">Troubleshoot BGP</a></li> </ul> Device Level: <ul style="list-style-type: none"> <li>- View the BGP Advertised-route Table<sup>1)</sup></li> <li>- Execute CLI Commands <b>show access-list, show ip prefix-list, and show route-map</b><sup>2)</sup></li> </ul>

Category	Built-in DVT Name	Branch Criteria	Supporting Variables	Drill Down Actions
	ISIS Overview	<ul style="list-style-type: none"> <li>- Legacy Device (IPv4 Interface)</li> <li>- Config File contains 'router isis' and Device Type contains Cisco IOS Switch; Cisco Router</li> </ul>	<p>Device Level:</p> <ul style="list-style-type: none"> <li>- clns_neighbors<sup>1)</sup></li> <li>- clns_interfaces<sup>2)</sup></li> <li>- level_1sp_database<sup>3)</sup></li> <li>- isis_protocol_process<sup>4)</sup></li> </ul> <p>Interface Level:</p> <ul style="list-style-type: none"> <li>- level_1_metric</li> <li>- level_2_metric</li> <li>- l1_adj_count</li> <li>- l2_adj_count</li> </ul>	<p>Template Level:</p> <ul style="list-style-type: none"> <li>- Compare the Configuration File and Route Table in two data sources</li> <li>- Show log include INPUT [Cisco IOS NXOS ASA]</li> </ul> <p>Device Level:</p> <ul style="list-style-type: none"> <li>- Ping the device; View the ARP Table; Execute CLI Commands <b>show isis neighbor<sup>1)</sup></b></li> <li>- Execute CLI Commands <b>show clns interface<sup>2)</sup></b></li> <li>- Execute CLI Commands <b>show isis database<sup>3)</sup></b></li> <li>- Execute CLI Commands <b>show isis process<sup>4)</sup></b></li> </ul>
	Multicast Overview	<ul style="list-style-type: none"> <li>- Legacy Device (IPv4 Interface)</li> <li>- Config File contains 'ip pim' and Device Type contains Cisco IOS Switch; Cisco Router</li> </ul>	<p>Device Level:</p> <ul style="list-style-type: none"> <li>- pim_neighbors<sup>1)</sup></li> <li>- pim_intf<sup>2)</sup></li> <li>- rp_mapping<sup>3)</sup></li> <li>- igmp_groups<sup>4)</sup></li> </ul> <p>Interface Level:</p> <ul style="list-style-type: none"> <li>- version</li> <li>- dr_priority</li> <li>- dr1</li> <li>- nbr_count</li> </ul>	<p>Template Level:</p> <ul style="list-style-type: none"> <li>- <a href="#">Troubleshoot Multicast</a></li> <li>- Compare Configuration File in two data sources</li> <li>- Show log include INPUT [Cisco IOS NXOS ASA]</li> </ul> <p>Device Level:</p> <ul style="list-style-type: none"> <li>- Execute CLI Commands <b>show ip pim neighbor<sup>1)</sup></b>; View the Route Table</li> <li>- Execute CLI Commands <b>show ip pim interface<sup>2)</sup></b></li> <li>- Execute CLI Commands <b>show ip pim rp mapping</b> and <b>show access-list<sup>3)</sup></b></li> <li>- Execute CLI Commands <b>show ip igmp group<sup>4)</sup></b></li> </ul>
	OSPF Overview	<ul style="list-style-type: none"> <li>- Legacy Device (IPv4 Interface)</li> <li>- Config File contains 'router ospf' and Device</li> </ul>	<p>Device Level:</p> <ul style="list-style-type: none"> <li>- ospf_nbrs<sup>1)</sup></li> <li>- ospf_intf<sup>2)</sup></li> </ul>	<p>Template Level:</p> <ul style="list-style-type: none"> <li>- Compare the Configuration File in two data sources</li> </ul>

Category	Built-in DVT Name	Branch Criteria	Supporting Variables	Drill Down Actions
		Type contains Cisco IOS Switch; Cisco Router	<ul style="list-style-type: none"> <li>- ospf_process<sup>3)</sup></li> <li>- RouteTable<sup>4)</sup></li> </ul> Interface Level: <ul style="list-style-type: none"> <li>- cost</li> <li>- nbrs_fc</li> <li>- intf_state</li> </ul>	<ul style="list-style-type: none"> <li>- Show log include INPUT [Cisco IOS NXOS ASA]</li> <li>- <a href="#">Troubleshoot OSPF</a></li> </ul> Device Level: <ul style="list-style-type: none"> <li>- Ping the device; Execute CLI Commands <b>show ip ospf neighbor detail<sup>1)</sup></b></li> <li>- Execute CLI Commands <b>show ip interface brief, show ip ospf interface</b>, and <b>show ip int   in list   line<sup>2)</sup></b></li> <li>- Execute CLI Commands <b>show ip ospf</b> and <b>show ip ospf database self-originate<sup>3)</sup></b></li> <li>- Execute CLI Commands <b>show ip route summary</b> and <b>show ip route ospf</b>; Show ip route INPUT [ Cisco IOS NXOS ASA]<sup>4)</sup></li> </ul>
Security	Access List [Cisco IOS]	<ul style="list-style-type: none"> <li>- Legacy Device (IPv4 Interface)</li> <li>- Config File contains 'access-list' and Device Type contains Cisco IOS Switch; Cisco Router</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- acl_list<sup>1)</sup></li> <li>- acl_detail</li> <li>- intf_table<sup>2)</sup></li> <li>- ACL Tips</li> </ul> Interface Level: <ul style="list-style-type: none"> <li>- in_acl</li> <li>- mtu</li> <li>- out_acl</li> <li>- nat</li> </ul>	Template Level: <ul style="list-style-type: none"> <li>- Compare the Configuration File in two data sources</li> <li>- <a href="#">Troubleshoot Cisco ACL</a></li> </ul> Device Level: <ul style="list-style-type: none"> <li>- Execute CLI Commands <b>show access-list<sup>1)</sup></b></li> <li>- Execute CLI Commands <b>show ip interface<sup>2)</sup></b></li> </ul>
Switching	Spanning Tree Overview	<ul style="list-style-type: none"> <li>- Legacy Device</li> <li>- Config File contains 'switchport' and Config File contains 'spanning' and Device Type contains Cisco IOS Switch</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- root_bridge</li> <li>- spanning_tree_root</li> <li>- vlan_intfs</li> <li>- stp_features</li> </ul>	Template Level: <ul style="list-style-type: none"> <li>- Compare the Configuration File and STP Table in two data sources</li> <li>- View the Spanning Tree VLAN Instance</li> </ul>
	Spanning Tree VLAN Instance	<ul style="list-style-type: none"> <li>- Legacy Device (Physical Interface)</li> <li>- Config File contains 'switchport' and Config</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- bridge_address</li> <li>- bridge_priority</li> </ul>	N/A

Category	Built-in DVT Name	Branch Criteria	Supporting Variables	Drill Down Actions
		File contains 'spanning' and Device Type contains Cisco IOS Switch	<ul style="list-style-type: none"> <li>- stp_vlan</li> </ul> Interface Level: <ul style="list-style-type: none"> <li>- role</li> <li>- type</li> <li>- cost</li> </ul>	
VMware	Infrastructure View [vCenter]	Branch: Legacy <ul style="list-style-type: none"> <li>- Legacy Device (Physical Interface)</li> <li>- Device Type contains NSX Controller; NSX Distributed Logical Router; NSX Edge Security Gateway; NSX Manager; VM Host</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- ESXi Host Name</li> <li>- Memory Size</li> <li>- vCPU Number</li> <li>- UsedStorage_MB</li> </ul> Interface Level: <ul style="list-style-type: none"> <li>- Connected</li> <li>- Start Connected</li> </ul>	N/A
		Branch: VDS <ul style="list-style-type: none"> <li>- VMware Distributed Virtual Switch (Port)</li> </ul>	Interface Level: <ul style="list-style-type: none"> <li>- Port Group</li> <li>- State</li> <li>- Connectee</li> <li>- Vlan</li> </ul>	
	Operation View [NSX-V]	Branch: Branch1 <ul style="list-style-type: none"> <li>- Legacy Device</li> <li>- Device Type contains NSX Distributed Logical Router; NSX Edge Security Gateway</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- edge_status (string)</li> <li>- ha_state (string)</li> <li>- publish_status (string)</li> <li>- system_status (string)</li> </ul>	N/A
		Branch: Branch2 <ul style="list-style-type: none"> <li>- Legacy Device</li> <li>- Device Type contains NSX Controller</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- status (string)</li> <li>- nodeFailoverReady (bool)</li> <li>- connectivity_table (table)</li> </ul>	
		Branch: Branch3 <ul style="list-style-type: none"> <li>- Legacy Device</li> <li>- Device Type contains NSX Manager</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- componentsByGroup_SYSTEM_components</li> </ul>	

Category	Built-in DVT Name	Branch Criteria	Supporting Variables	Drill Down Actions
			- componentsByGroup_COMMON_components	
	Operation View [vCenter]	Branch: Branch1 - Legacy Device (Physical Interface) - Device Type contains NSX Controller; NSX Distributed Logical Router; NSX Edge Security Gateway; NSX Manager; VM Host	Device Level: - overallStatus (string) - overallCpuUsage (int) - hostMemoryUsage (int) - guestHeartbeatStatus (string) Interface Level: - status (string) - startConnected (bool) - connected (bool)	N/A
		Branch: Branch2 - Legacy Device (Physical Interface) - Device Type contains vSphere Distributed Switch	Device Level: - hosts_status_table Interface Level: - intf_status (string)	
Generic	Infrastructure View	Branch: ACI - Fabric Node	Device Level: - Node Id - Model - Serial Number - Version	Template Level: - Compare the Configuration File in two data sources - Monitor the overall health of all devices via SNMP - Learn more about <a href="#">how to use NetBrain</a>
		Branch: Legacy - Legacy Device - Device Main Type contains Router; L3 Switch; Firewall; WAN Optimizer; LAN Switch; Load Balancer; Call Manager; WAP; WLC	Device Level*: - Mgmt IP <sup>1)</sup> - Model - Serial Number - Site	Device Level*: - Ping the device; Use Traceroute to discover and map a routing path <sup>1)</sup>
	Network Table View	Branch: Legacy - Legacy Device - Device Main Type contains Router; L3	Device Level: - NDPTable <sup>1)</sup> - RouteTable <sup>12)</sup>	Template Level: - Monitor the overall health of all devices via SNMP

Category	Built-in DVT Name	Branch Criteria	Supporting Variables	Drill Down Actions
		Switch; Firewall; WAN Optimizer; LAN Switch; Load Balancer; Call Manager; WLC	<ul style="list-style-type: none"> <li>- ARPTable1<sup>3)</sup></li> <li>- MACTable1<sup>4)</sup></li> </ul>	<ul style="list-style-type: none"> <li>- View the <a href="#">training video</a> to see how NetBrain delivers deeper network visibility, faster troubleshooting diagnoses and enhanced collaboration within IT teams</li> </ul> <p>Device Level:</p> <ul style="list-style-type: none"> <li>- View the NDP Table; Compare the NDP Table in two data sources<sup>1)</sup></li> <li>- View the Route Table; Compare the Route Table in two data sources<sup>2)</sup></li> <li>- View the ARP Table; Compare the ARP Table in two data sources<sup>3)</sup></li> <li>- View the MAC Table; Compare the MAC Table in two data sources<sup>4)</sup></li> </ul>
Overall Health Check		Branch: CiscoASA <ul style="list-style-type: none"> <li>- Legacy Device (IPv4 Interface)</li> <li>- Device Type contains Cisco ASA Firewall</li> </ul>	Device Level*: <ul style="list-style-type: none"> <li>- one_min_cpu_usage<sup>1)</sup></li> <li>- used_mem_util<sup>2)</sup></li> <li>- failover_unit<sup>3)</sup></li> <li>- failover_interfaces<sup>4)</sup></li> </ul> Interface Level: <ul style="list-style-type: none"> <li>- input_errors (int)</li> <li>- crc (int)</li> <li>- output_error (int)</li> <li>- collections (int)</li> <li>- one_min_input_rate_pkts</li> <li>- one_min_output_rate_pkts</li> <li>- one_min_drop_rate_pkts</li> </ul>	Template Level: <ul style="list-style-type: none"> <li>- Show log include INPUT [Cisco IOS NXOS ASA]</li> <li>- Show run section INPUT [Cisco IOS NXOS ASA]</li> </ul> Device Level*: <ul style="list-style-type: none"> <li>- Execute CLI Commands <b>show cpu usage</b><sup>1)</sup></li> <li>- Execute CLI Commands <b>show memory</b><sup>2)</sup></li> <li>- Show log include INPUT [Cisco IOS NXOS ASA]<sup>3)</sup></li> <li>- Ping the device; Execute CLI Commands <b>show failover</b><sup>4)</sup></li> </ul>
		Branch: CiscoIOS <ul style="list-style-type: none"> <li>- Legacy Device (IPv4 Interface; Physical Interface)</li> </ul>	Device Level: <ul style="list-style-type: none"> <li>- intfs_table</li> <li>- five_min_cpu_usage</li> <li>- mem_util</li> </ul>	

Category	Built-in DVT Name	Branch Criteria	Supporting Variables	Drill Down Actions
		- Device Type contains Cisco IOS Switch; Cisco Router	- cpu_table Interface Level: - input_errors (double) - crc (int) - output_error (double) - collisions (int) - mtu (int) - duplex (string) - tx_load (int) - rx_load (int)	

## 5.2. Built-in Plugins

IEv8.0 provides the following 6 built-in Plugins to customize for data accuracy.

Plugin Name	Use Case	Execution Point
<b>Topology Stitching</b>	To modify the topology links between device interfaces. This plugin can be added to a benchmark task at the execution point after building L3 topology.  <b>Note:</b> This plugin might need further customization to meet customers' requirements.	After building L3 topology
<b>Import End System</b>	To add end systems to the system and set path their gateways for path calculation. This plugin can be added to a discovery task at the execution point before executing a discovery.  <b>Note:</b> This plugin might need further customization to meet customers' requirements.	Before executing a discovery task
<b>Import Configuration</b>	To add devices and device interfaces to the system. This plugin can be added to a discovery task at the execution point before executing a discovery.	Before executing a discovery task

Plugin Name	Use Case	Execution Point
	<b>Note:</b> This plugin might need further customization to meet customers' requirements.	
<b>Separate ASA Firewall Failover Interfaces into Different Zones</b>	To add failover interfaces of ASA Firewalls with the same IP addresses to different zones so that the failover interfaces can participate in L3 topology calculation. This plugin can be added to a benchmark task at the execution point before building the network's L3 topology.	Before building L3 topology
<b>Separate Checkpoint Firewall Sync Interfaces into Different Zones</b>	To add the sync interfaces of Checkpoint firewalls with the same IP addresses to different zones so that the sync interfaces can participate in L3 topology calculation. This plugin can be added to a benchmark task at the execution point before building the network's L3 topology.	Before building L3 topology
<b>Checkpoint Warp Topology Stitching</b>	To establish the Ipv4 L3 topology for Checkpoint virtual system and Checkpoint virtual router/virtual switch. This plugin can be added to a benchmark task at the execution point after building the network's L3 topology.	After building L3 topology

### 5.3. Application Weight

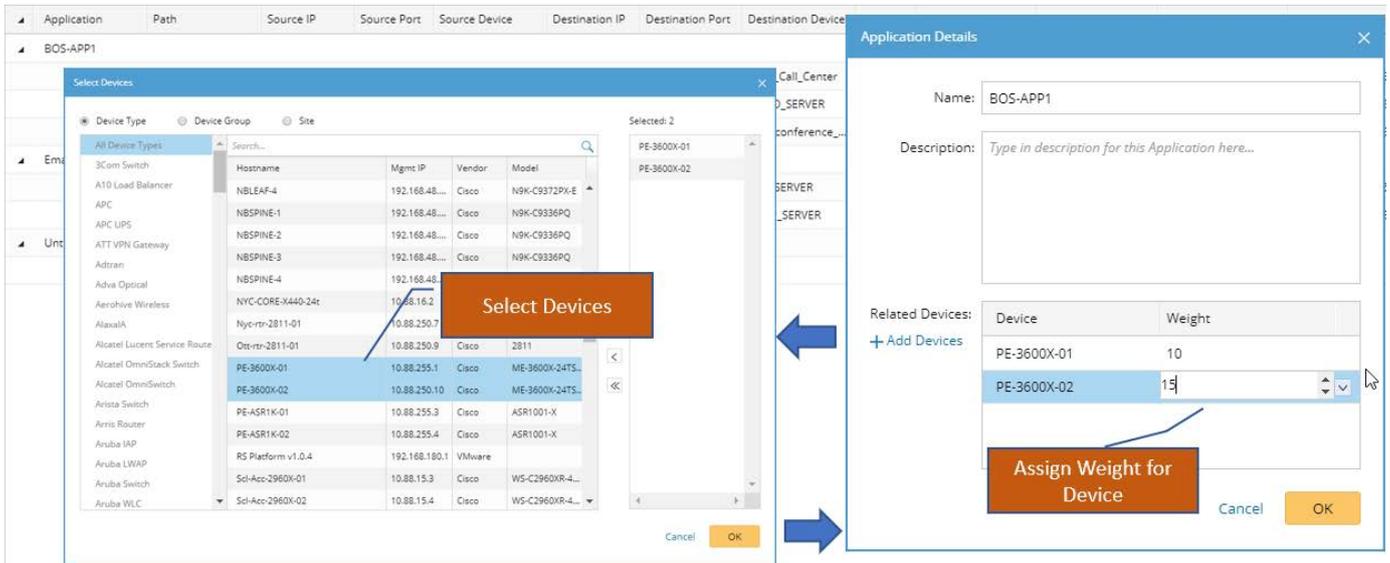
When you select paths to add to the Verify Application node of a runbook, the applications will be listed and sorted by weight. The weight of an application is the sum of the following two parts:

- **Device Weight:** the weight of all involved devices
- **Path Weight:** the weight of all paths contained in the application

#### Device Weight

The device weight of an application is the sum of the weights of all devices involved in the application.

When assigning devices to an application, you can manually assign a weight to each device.



## Path Weight

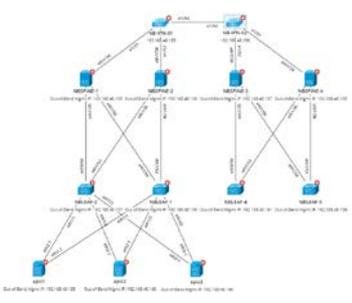
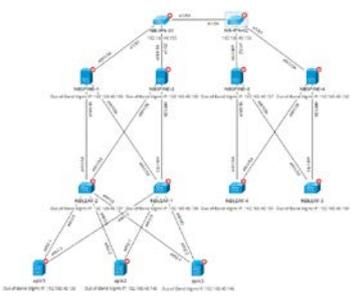
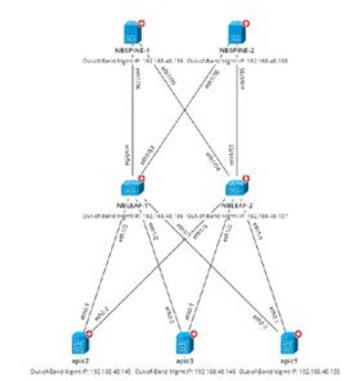
The path weight of an application is the sum of the weights of all the paths contained in the application. For example, APP1 has two paths, and the weight of Path1 is 3, the weight of Path2 is 5, then the path weight of the App1 is 8.

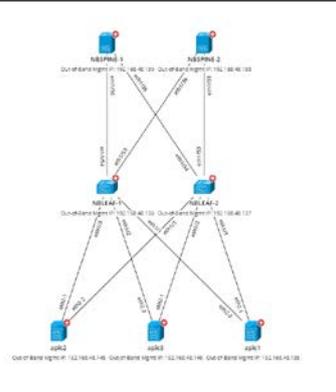
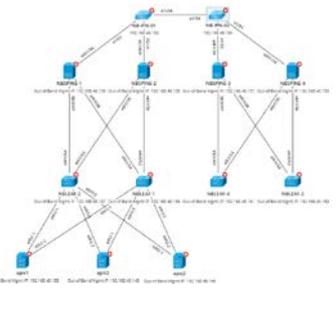
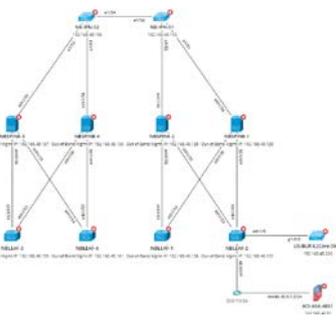
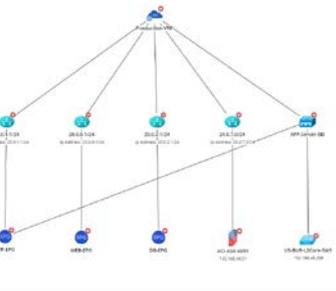
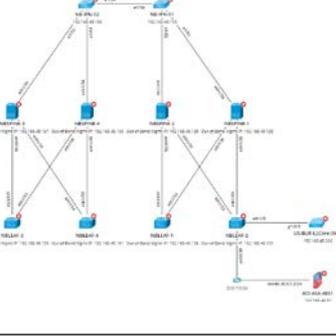
The weight of a path depends on how many devices that the path crosses on a map, and one device weighs 1. For example, a path crosses 5 devices but only 3 of 5 devices appear on the map, then the path weight is 3.

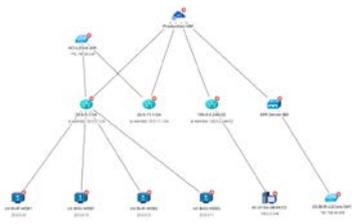
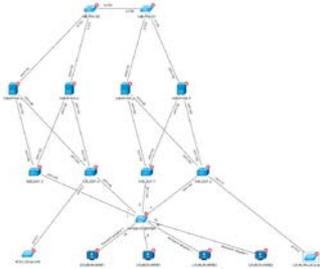
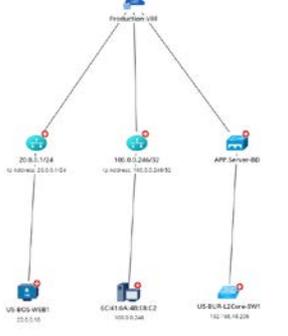
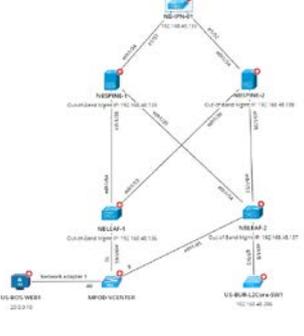
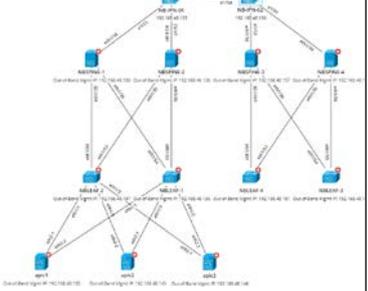
### Notes:

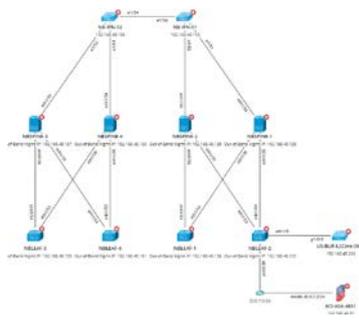
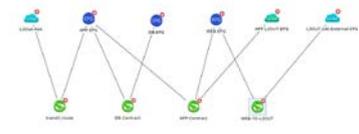
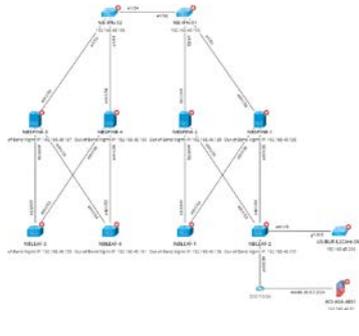
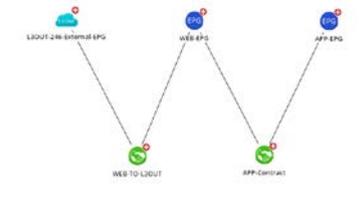
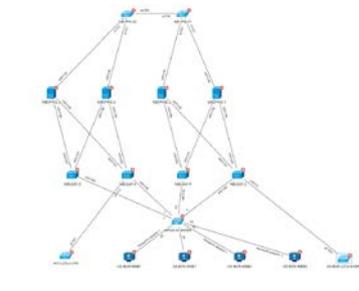
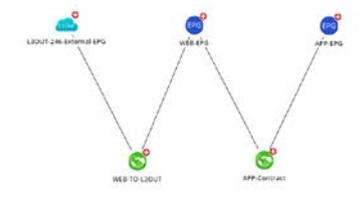
- Unknown devices and unknown IP on a map do not join the weight calculation.
- When finding the path that a device belongs to, the system will first search it in the Golden Path, and then look it up in the latest successful path if there is no Golden Path.

## 5.4. SDN Context Maps

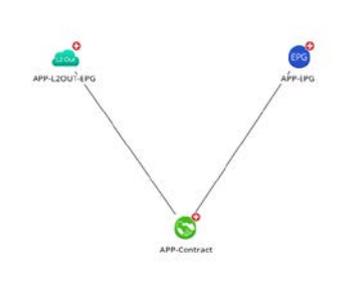
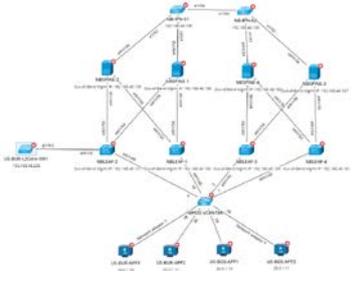
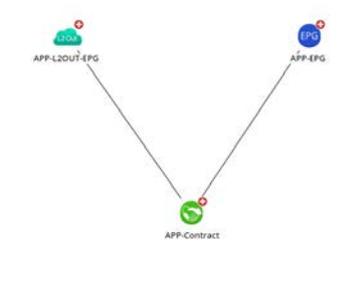
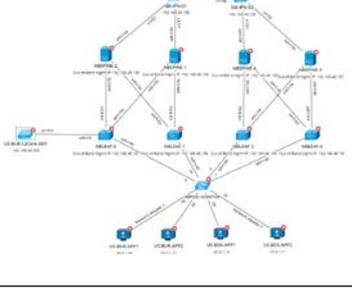
View	Object	Context Map	Sample	Improvement
Network Centric View/Fabric POD View	APIC Domain	Overlay Map	N/A	
Network Centric View/Fabric POD View	APIC Domain	Underlay Map		Display whole fabric underlay map
Network Centric View/Fabric POD View	IPN Folder	Overlay and Underlay Map	N/A	
Network Centric View/Fabric POD View	IPN Device	Overlay Map	N/A	
Network Centric View/Fabric POD View	IPN Device	Underlay Map		Same as APIC Domain Underlay Map
Network Centric View/Fabric POD View	POD	Underlay Map		Display Spine, Leaf and APIC in that POD only

Network Centric View/Fabric POD View	Spine/Leaf/APIC	Underlay Map		Inherit POD underlay Map
Network Centric View/Tenant View	APIC Domain	Underlay Map		Display whole fabric underlay map
Network Centric View/Tenant View	Tenant	Overlay Map	N/A	Removed from IEv8.0 due to performance issue
Network Centric View/Tenant View	Tenant	Underlay Map		Display Spine, Leaf and all L3 Out and L2 Out related to this Tenant
Network Centric View/Tenant View	VRF	Overlay Map		Display VRF, Subnet, BD, EPG and L3 Out, L2 Out for the VRF.
Network Centric View/Tenant View	VRF	Underlay Map		Display all Spine and Leaf switches belong to the VRF, if the VRF across Multi-POD show IPN devices as well. All L3 Out and L2 Out related to this VRF will be included.

Network Centric View/Tenant View	EPG	Overlay Map		Display logic relationship from EPG to Subnet/BD to VRF. If there's any L3/L2 Out related to this EPG should be shown here as well
Network Centric View/Tenant View	EPG	Underlay Map		Display all spine and leaf switches cover this EPG, if the EPG across Multi-POD will be shown IPN devices as well.
Network Centric View/Tenant View	End Point	Overlay Map		Inherit EPG Overlay Map but use current End Point as filter criteria.
Network Centric View/Tenant View	End Point	Underlay Map		Inherit EPG Underlay Map but use current End Point as filter criteria.
Application Centric View	APIC Domain	Overlay Map	N/A	
Application Centric View	APIC Domain	Underlay Map		Display the whole fabric underlay map

Application Centric View	Tenant	Logical Map	N/A	
Application Centric View	Tenant	Underlay Map		Display IPN, Spine, Leaf and L3/L2 Out devices that current Tenant resource covered.
Application Centric View	ANP	Logical Map		Display all EPGs, Contracts and L3/L2 Outs belong to current ANP.
Application Centric View	ANP	Underlay Map		Display IPN, Spine, Leaf and L3/L2 Out devices belong to current ANP.
Application Centric View	EPG	Logical Map		Display all EPGs and L3/L2 Outs have contract with current EPG
Application Centric View	EPG	Underlay Map		Display IPN, Spine, Leaf and L3/L2 Out devices related to current EPG
Application Centric View	End Point	Logical Map		Inherit EPG's Logical Map

Application Centric View	End Point	Underlay Map		Display IPN, Spine, Leaf and L3/L2 Out devices related to current End Point.
Application Centric View	L3 Out Instance	Logical Map		Display the connectivity with L3 Out, Contract and EPG
Application Centric View	L3 Out Instance	Underlay Map		Display the physical resource for L3 Out Device and Endpoint related to the EPG that has contract with L3 Out.
Application Centric View	L3 Out Device	Logical Map		Inherit L3 Out instance
Application Centric View	L3 Out Device	Underlay Map		Inherit L3 Out instance

Application Centric View	L2 Out Instance	Logical Map		Display the connectivity with L2 out, Contract and related EPG
Application Centric View	L2 Out Instance	Underlay Map		Display the physical resource of L2 out and related EPG
Application Centric View	L2 Out Device	Logical Map		Inherit L2 Out instance
Application Centric View	L2 Out Device	Underlay Map		Inherit L2 Out Instance

**Note:** For performance considerations, the underlay context map will only display end points in the following scenarios:

- EPG or end point under EPG is selected;
- End point number below 200.

## 5.5. Technology Support Details

### 5.5.1. VXLAN

Virtual Extensible LAN (VXLAN) is a technology to provide layer 2 overlay networks on top of a layer 3 network. It is one of the most popular and effective overlay networking technologies for building data center networks.

IEv8.0 can build a VXLAN data model and calculate VXLAN overlay topology and path.

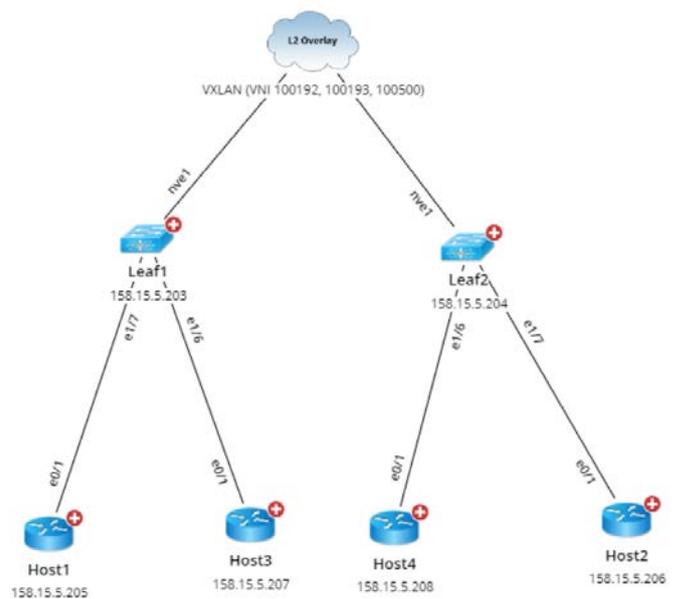
**Note:** IEv8.0 only supports VXLAN configured on Cisco Nexus Switch. More vendors or VXLAN scenarios can be supported via drivers.

### Calculate Overlay Topology of L2 Overlay Network

IEv8.0 introduces a new media type “L2 Overlay” to interconnect VXLAN edge devices on maps, and a new topology type “L2 Overlay Topology” to present the virtual L2 topology of L2 Overlay Network.

**Example:** An L2 Overlay Topology of VXLAN.

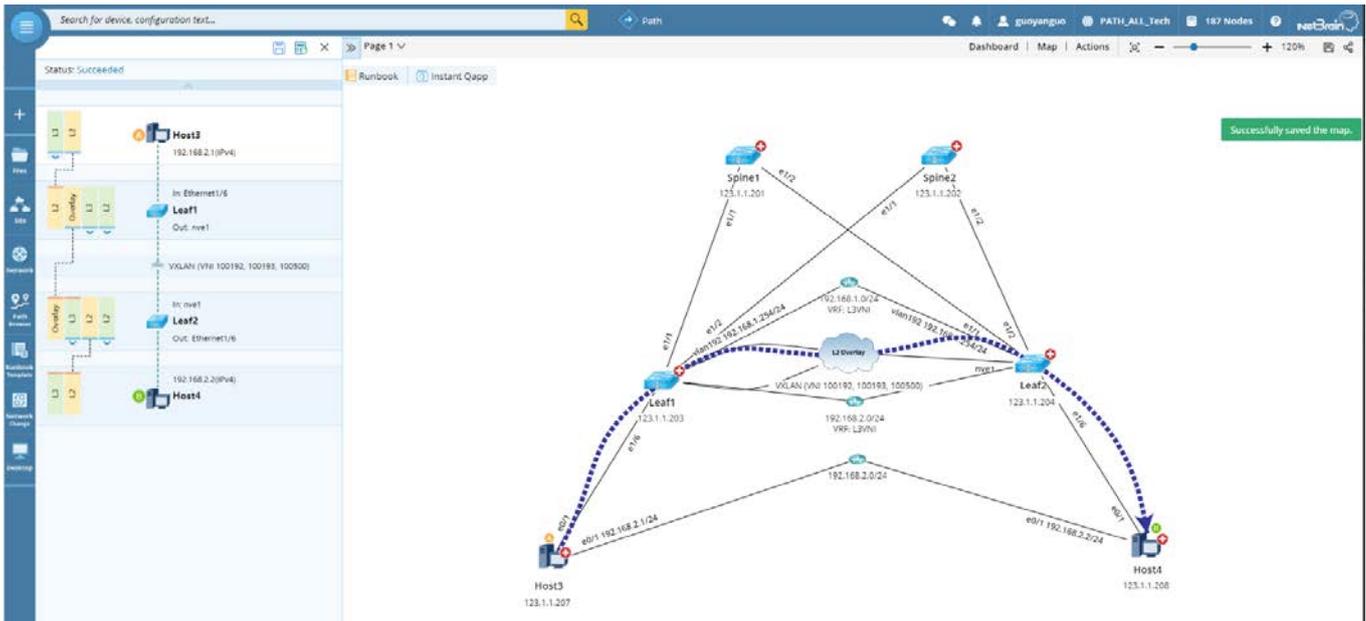
To understand the data model of VXLAN topology and how it is built, refer to [Build a Data Model for VXLAN](#) for details.



### Calculate Path Across VXLAN Overlay Network

IEv8.0 is able to discover the VXLAN path and its underlay L3 path.

## Example: A-B Path across VXLAN Overlay Network.



## Build a Data Model for VXLAN

To calculate the topology and path for VXLAN, IEx8.0 creates the following GDR and NCT mostly via drivers.

### Topology Dependency GDR:

The table below describes the GDR information about topology dependency, which is used for VXLAN path calculation.

GDR ID	Type	Description	Sample
intfs.topoDep.topoType	string	The topology types of the current interface.	<pre>"intf_name": " F 0/0", "topologies": [ { "topology_type": "IPv4 L3 Topology", "dependency": { "topology_type": "L2 Topology", "interface": "F0/0"</pre>
intfs.topoDep.dep	object		
intfs.topoDep.dep.topoType	string	The topology type of one dependent interface of the current interface.	
intfs.topoDep.dep.intf	string	The dependent interface of the current interface.	

## VXLAN Device GDR:

The table below describes the VXLAN device GDR information used for the VXLAN topology calculation.

GDR ID	Type	Description	Configuration Example
hasVXLANConfig	bool	Determine whether a device is configured with VXLAN. The judgment logic varies on vendors and needs to be defined in each device type driver. For Cisco Nexus Switch, the logic is that config file contains <i>interface nve</i> .	
VXLAN	list	A list of VXLAN objects.	
VXLAN.intf	string	The name of nve interface.	<pre>interface nve1 source-interface loopback1 member vni 100192 no suppress-arp no suppress-unknown-unicast mcast-group 224.1.1.192</pre>
VXLAN.VNI	string	A list of VNIs configured under a nve interface. e.g., the parsed VNI value of the configuration example will be: '100192; '100193'	<pre>interface nve1 source-interface loopback1 member vni 100192 no suppress-arp no suppress-unknown-unicast mcast-group 224.1.1.192 member vni 100193 no suppress-arp no suppress-unknown-unicast mcast-group 224.1.1.192</pre>
VXLAN.IP	string	The IP address of the source interface of nve interface.	<pre>interface nve1 source-interface loopback1 member vni 100192 no suppress-arp no suppress-unknown-unicast mcast-group 224.1.1.192  interface loopback1 description Overlay ip address 10.1.255.11/32 ip router ospf 255 area 0.0.0.0 ip pim sparse-mode no shutdown</pre>

## Topology Calculation Logic

The system determines which devices will join topology calculation through the GDR property "hasVXLANConfig", retrieve the VXLAN peers of the devices having VXLAN configuration through the following NCT table "VXLAN peer table", and then connects a VXLAN device and its peers to the same L2 Overlay media. As for the name of the L2 Overlay media, its format is "VXLAN" <VNI>.

## New NCT for Path and Topology Calculation

---

The table below shows the VXLAN peer table in the system.

Peer-ID	Interface	Peer-IP	State
1	nve1	10.1.255.12	Up

---

### 5.5.2.OTV

Overlay transport virtualization (OTV) is a Cisco proprietary technology to extend Layer 2 applications across distributed Data Centers.

IEv8.0 can build a data model, calculate topology and path for OTV.

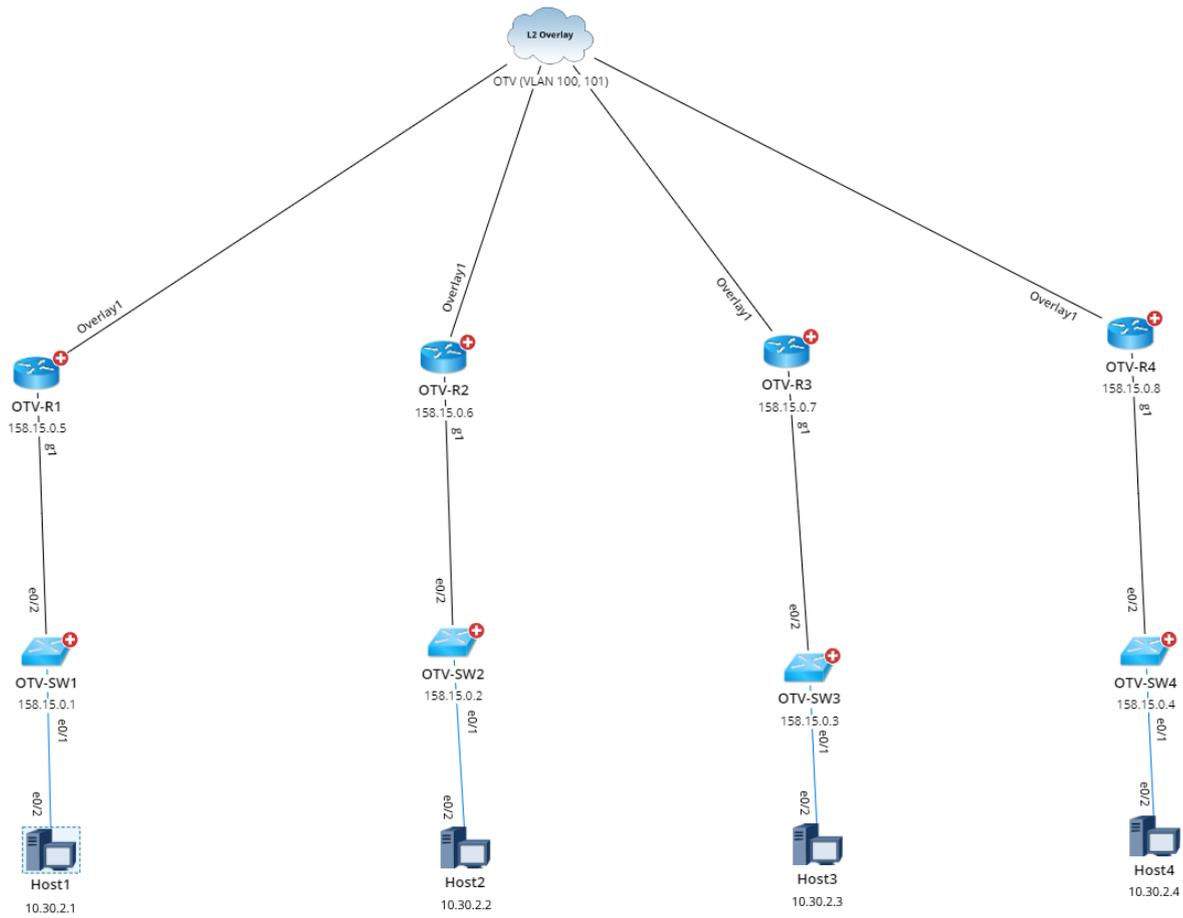
**Note:** IEv8.0 only supports the OVT on Cisco Router (XE) and Cisco Nexus Switch. More vendors can be supported by drivers.

### Calculate Overlay Topology of L2 Overlay Network

---

IEv8.0 introduces a media type "L2 Overlay" to interconnect OTV edge devices on maps, and a new topology type "L2 Overlay Topology" to present the virtual L2 topology of L2 Overlay Network.

**Example:** An L2 Overlay Topology of OTV.

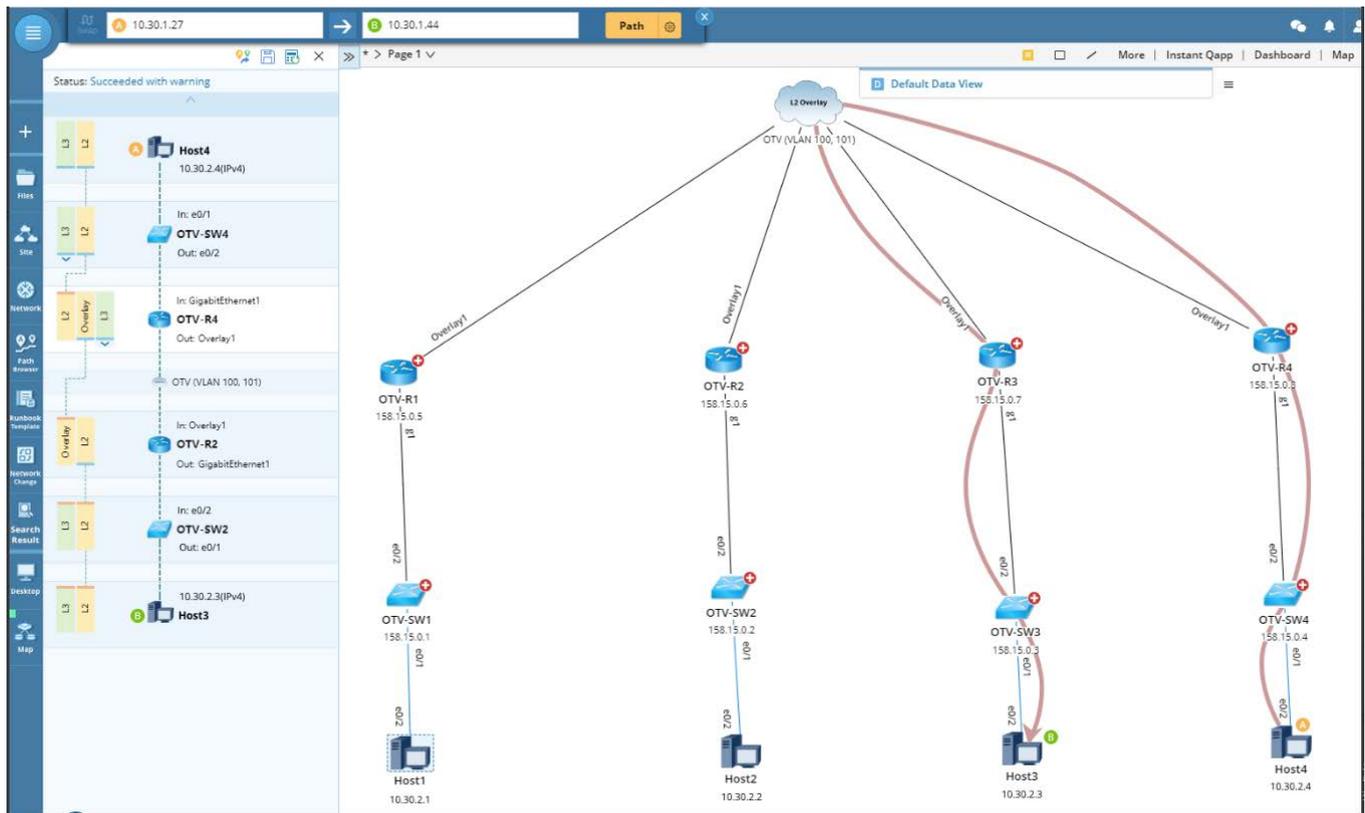


To understand the data model of VXLAN topology and how it is built, refer to [Build a Data Model for OTV](#) for details.

## Calculate Path across OTV Overlay Network

IEv8.0 is able to discover the OTV path and its underlay L3 path.

**Example:** A-B Path across OTV Overlay Network.



## Build a Data Model for OTV

To calculate the topology and path for OTV, IEx8.0 creates the following GDR and NCT mostly via drivers.

### OTV Device GDR:

The table below describes the OTV device GDR information used for topology calculation.

GDR ID	Type	Description	Configuration Example
hasOTVConfig	bool	Determine whether a device is configured with OTV. The judgment logic varies on vendors and needs to be defined in each device type driver. For Cisco Nexus Switch, the logic is that the config file contains <b><i>Interface Overlay</i></b> .	
OTV	Table		
OTV.intf	string	The name of OTV interface. If the interface configuration starts with <b><i>Overlay</i></b> , add the specific overlay to OTV.intf.	interface <b>Overlay1</b>  no description  spanning-tree port-priority 128  spanning-tree cost auto

GDR ID	Type	Description	Configuration Example
OTV.extendVLAN	string	Display the extend VLANs for OTV interface.	Cisco Nexus Switch:  interface Overlay1  otv extend-vlan 100-101

**Overlay Topology Calculation Logic:**

The systems determine which devices will participate in the calculation through the GDR property "hasOTVConfig", finds OTV interface with the same extend VLANs through the OTV.extendVLAN property and connects them to the same L2 Overlay media. As for the name of the L2 Overlay media, its format is "OTV" <extend VLAN>.

**NCT:**

IEv8.0 records the OTV route information used for path calculation in NCT.

The table below describes the OTV route table in the system.

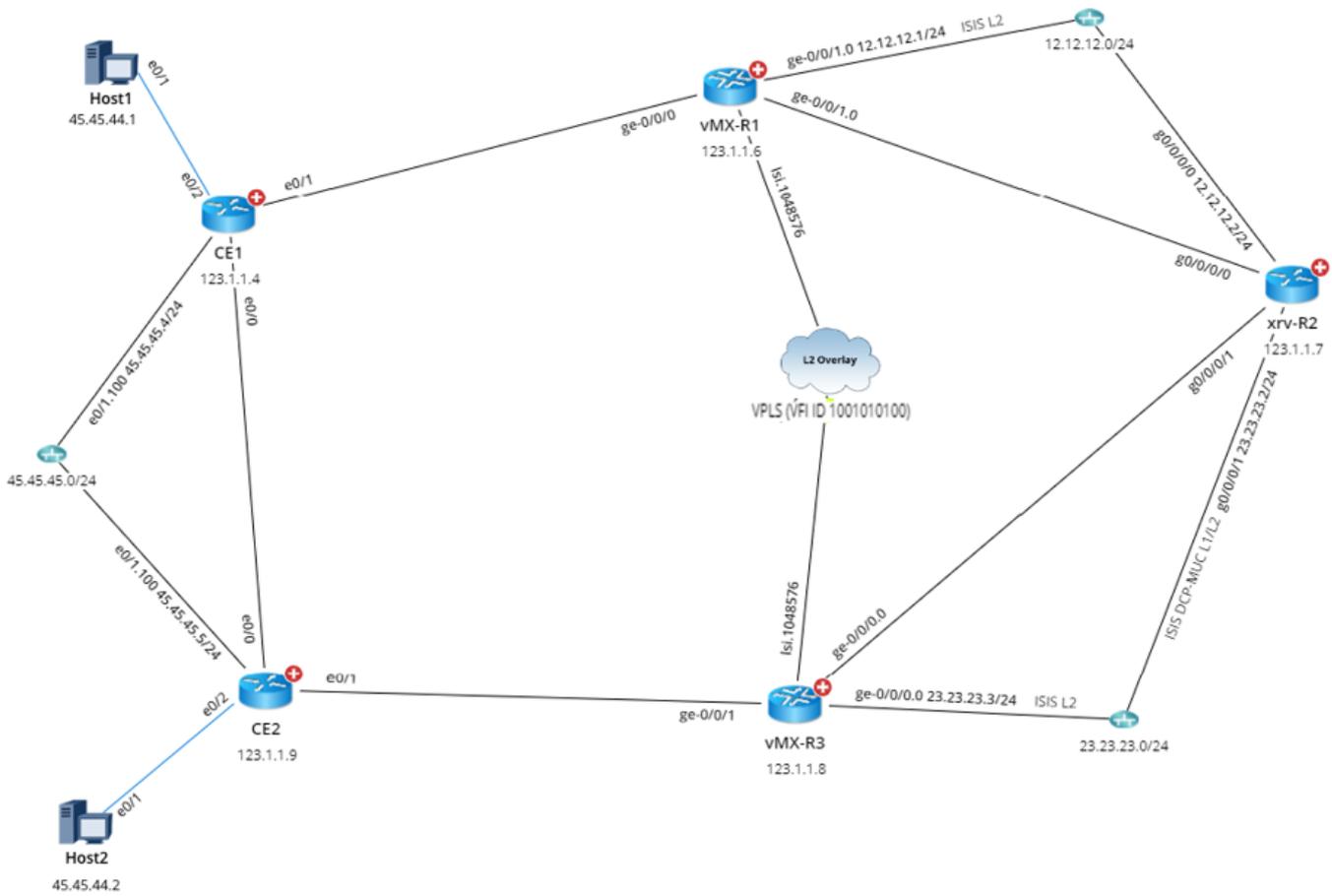
VLAN	MAC Address	Overlay	Output Interface	Destination Address	Next Device
100	aabb.cc00.9000	Overlay1	GigabitEthernet2		
100	aabb.cc00.b000	Overlay1	Overlay1	172.16.64.84	202-OTV-CSR3
101	aabb.cc00.a000	Overlay1	GigabitEthernet2		
101	aabb.cc00.c000	Overlay1	Overlay1	172.16.64.84	202-OTV-CSR3

**5.5.3.VPLS**

Virtual private LAN service (VPLS) is an Ethernet-based point-to-multipoint Layer 2 virtual private network (VPN) that connects geographically dispersed Ethernet local area network (LAN) sites across an MPLS backbone. IEv8.0 can build a data model for a VPLS network and calculate topology and path.

**Note:** IEv8.0 only supports the VPLS on Juniper Routers with CE and PE both discovered in NetBrain. More vendors can be supported by drivers.

**Example:** Visualized topology of a VPLS network.

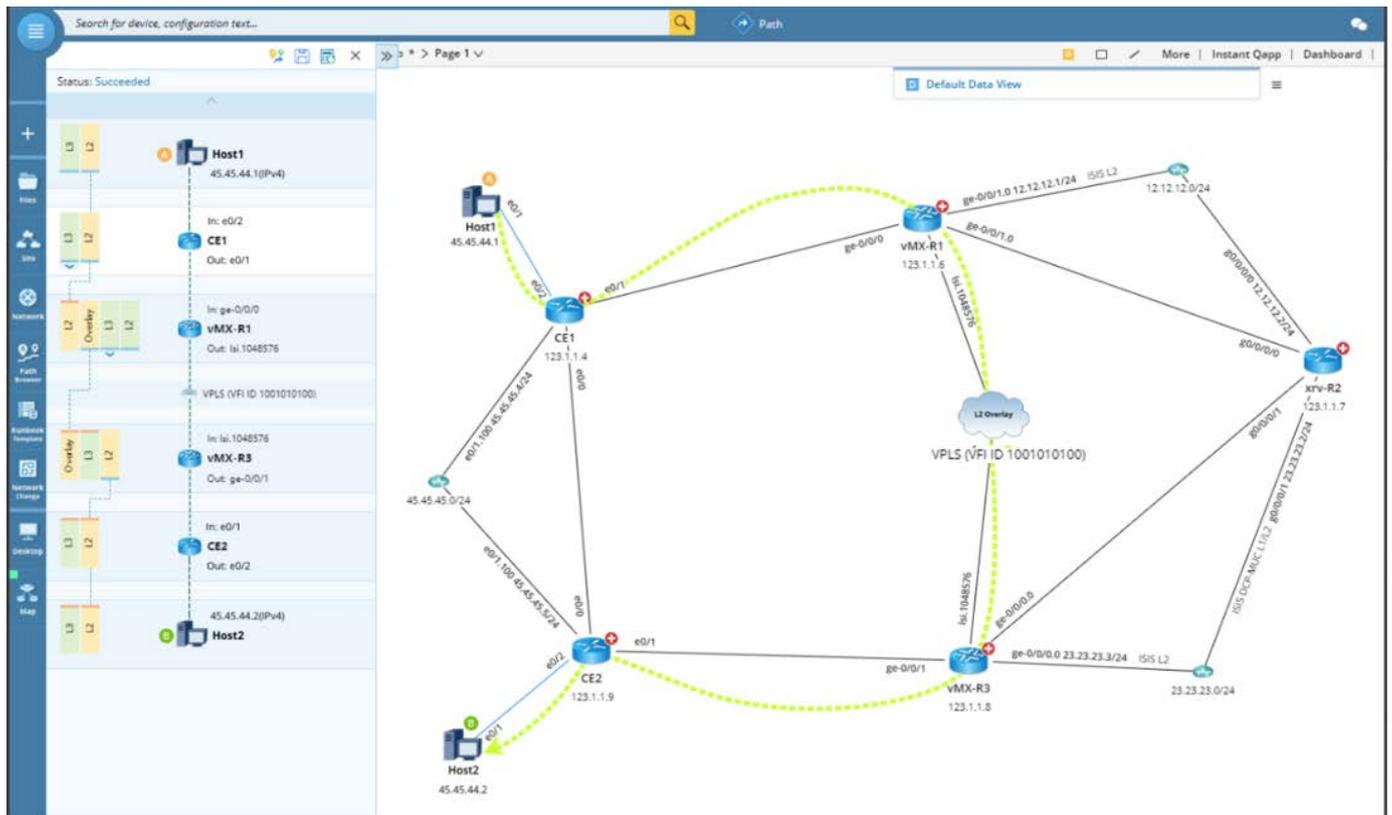


To understand the data model of VPLS topology and how it builds these data, refer to [Build a Data Model for VPLS](#) for details.

## Calculate Path across VPLS Network

IEv8.0 is able to discover a VPLS path and its underlay L3 path.

**Example:** A/B Path across a VPLS Overlay Network.



## Build a Data Model for VPLS

To calculate the topology and path for VPLS, IEv8.0 creates the following GDR and NCT mostly via drivers.

### VPLS Device GDR:

The table below describes the VPLS device GDR information used for topology calculation.

GDR ID	Type	Description	Configuration Example
hasVPLSConfig	bool	Determine whether a device is configured with VPLS. The judgment logic varies on vendors and needs to be defined in each device type driver. For Juniper router, the logic is that the config file contains <i>instance-type vpls</i> .	
VPLS	list	A list of VPLS VFIs.	
VPLS.PWID	string	The ID of a VPLS virtual forwarding instance (VFI).	routing-instances { v1 { instance-type vpls; interface ge-0/0/0.100; protocols { vpls {

GDR ID	Type	Description	Configuration Example
			<pre>no-tunnel-services; vpls-id 1001010100; neighbor 1.49.89.3; } } }</pre>
VPLS.in_intf	string	The input interface for the VPLS VFI	<pre>routing-instances { v1 { instance-type vpls; interface ge-0/0/0.100; protocols { vpls { no-tunnel-services; vpls-id 1001010100; neighbor 1.49.89.3; } } } }</pre>
VPLS.lsi_intf	string	The lsi interface for the VPLS VFI, which will be used to build L2 Overlay Topology. This interface is found by searching the output of the command 'show vpls connections'.	<pre>nb@vMX-R3&gt;show vpls connections  Instance: v1 VPLS-id: 1001010100 Neighbor Type St Time last up # Up trans 1.49.89.1(vpls-id 1001010100) rmt Up Apr 1 08:53:13 2019 1 Remote PE: 1.49.89.1, Negotiated control- word: No Incoming label: 262145, Outgoing label: 262145 Negotiated PW status TLV: No Local interface: lsi.1048576, Status: Up, Encapsulation: ETHERNET Description: Intf - vpls v1 neighbor 1.49.89.1 vpls-id 1001010100 Flow Label Transmit: No, Flow Label Receive: No</pre>
VPLS.neighbor	string	The neighbor IP of the VPLS VFI.	<pre>routing-instances { v1 { instance-type vpls; interface ge-0/0/0.100; protocols { vpls { no-tunnel-services; vpls-id 1001010100; neighbor 1.49.89.3; } } } }</pre>

GDR ID	Type	Description	Configuration Example
VPLS.out_intf	string	The output interface for the VPLS VFI, which will be used to support topology dependency. This interface is found by searching the output of command 'show route table I2circuit.0 all protocol vpls detail.'	<pre>nb@vMX-R3&gt;show route table I2circuit.0 all protocol vpls detail  I2circuit.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden) 1.49.89.1:NoCtrlWord:5:1001010100:Local/96 (1 entry, 1 announced) *VPLS Preference: 7 Next hop type: Indirect Address: 0x97403ac Next-hop reference count: 2 Next hop type: Router Next hop: 23.23.23.2 via ge-0/0/0.0, selected Label operation: Push 24001 Label TTL action: prop-ttl Load balance label: Label 24001: None; Session Id: 0x0 Protocol next hop: 1.49.89.1 Indirect next hop: 0x975c000 - INH Session ID: 0x0 State: &lt;Active Int&gt; Age: 14:54:55 Metric2: 1 Validation State: unverified Task: I2 circuit Announcement bits (1): 0-LDP AS path: I VC Label 262145, MTU 1500, Flow Label T Bit 0, Flow Label R Bit 0</pre>
intfs.VPLSPEIntf	bool	<p>Indicate if an interface is a VPLS PE interface.</p> <p>If the input interface of a VFI is a sub interface, only the parent interface will be marked as VPLS PE interface.</p>	

### VPLS L2 Topology Calculation Logic:

IEv8.0 uses the following logic to calculate VPLS L2 topology: If one same interface in the NDP table of a device has two entries, one is connected to the VPLS PE device, the other is connected the non-VPLS PE device, the system uses the entry connected to the VPLS PE device to generate the L2 topology.

### VPLS Overlay Topology Calculation Logic

IEv8.0 uses the following logic to calculate VPLS L2 Overlay Topology: Determine which devices to join the overlay topology calculation by looking up the "hasVPLSConfig" GDR property, find out VPLS devices and their neighbors through the NCT table "L2VPN Forwarding Table[Real-time]" and then connect the NVE interfaces of

a VPLS device and its neighbors to the same L2 overlay media. As for the name of the L2 Overlay media, its format is "VPLS" < lsi of connected device >

**NCT:**

IEv8.0 records the VPLS Peer information used for path calculation in NCT.

The table below describes the VPLS peer table in the system.

MAC Address	Type	Learned From/Filtered on	Neighbor	Circuit	Next Hop	Out Interface	Tunnel Label	VC Label
aabb.cc00.7110	D	lsi.1048576	1.49.89.3	1001010100	12.12.12.2	ge-0/0/1.0	24000	262145
aabb.cc00.9110	D	ge-0/0/0.100						

**VPLS Underlay Path Calculation Logic**

When calculating the L2 path on a VPLS enabled juniper router, if the matching entry of the MAC Address has a value for the 'Neighbor', use the LSI interface as an output interface to activate topology dependency on L2 Overlay Topology. Record the IP of Neighbor to special info as 'L2 Overlay Destination Address', push Tunnel Label to the L2 packet, and record the original Next Hop address to special info as 'VPLS Next Hop'.

When initializing an L3 path, in case the path is created by topology dependency with a parent traffic state, and there is an L2 Overlay Destination Address in special info, and the device type is Juniper Router, set the first traffic state of the L3 path to have:

- eth\_type as ETH\_TYPE\_MPLS\_LABEL
- next hop as VPLS Next Hop in special info
- MPLS labels as the labels in the parent traffic state

---

## 5.5.4.HA/Cluster

In previous versions, NetBrain already supported HA/Cluster. However, there is an obvious flaw with the previous logic: to avoid the problem that both active and standby devices parsed into the system cause duplicate IP issues, some interfaces and IPs of standby devices are filtered out when being parsed. This processing logic causes that the standby devices do not have their own topology and do not join path calculation.

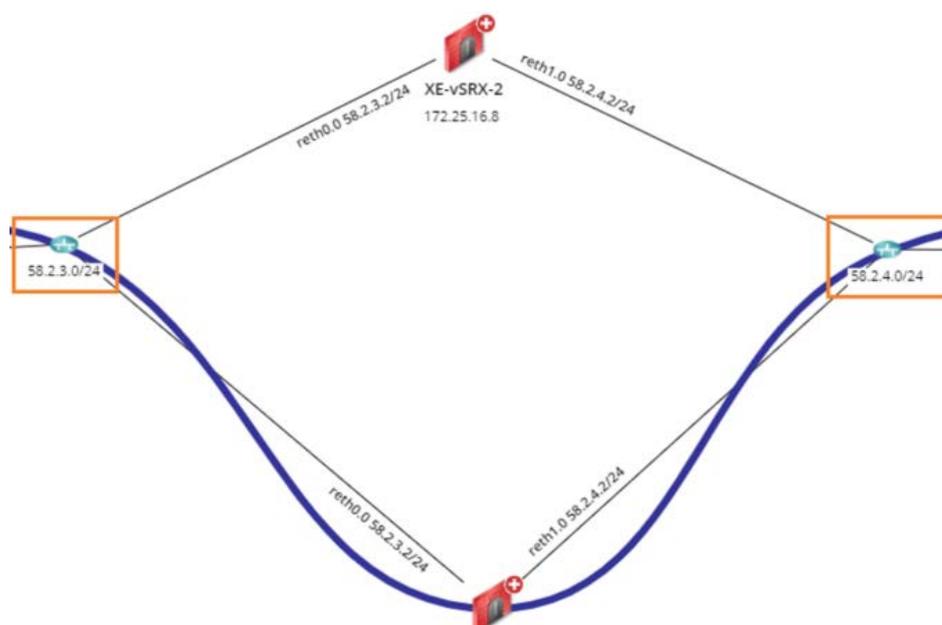
To fix the issue, I Ev8.0 improves the HA/Cluster support in terms of Data Model, topology and path.

**Note:** I Ev8.0 only supports the HA/Cluster on the following device types:

- Cisco ASA Firewall
- Cisco Router
- Juniper SRX
- Palo Alto Firewall

## Calculate L3 Topology for HA/Cluster

I Ev8.0 supports the topology calculation of duplication IP addresses in HA/Cluster and enables both active and standby devices to join the topology calculation. When calculating a topology, NetBrain allows all HA/Cluster duplicate IP addresses to join the L3 topology calculations, and these duplicate IP addresses in a NetBrain domain are connected to the same LAN media.

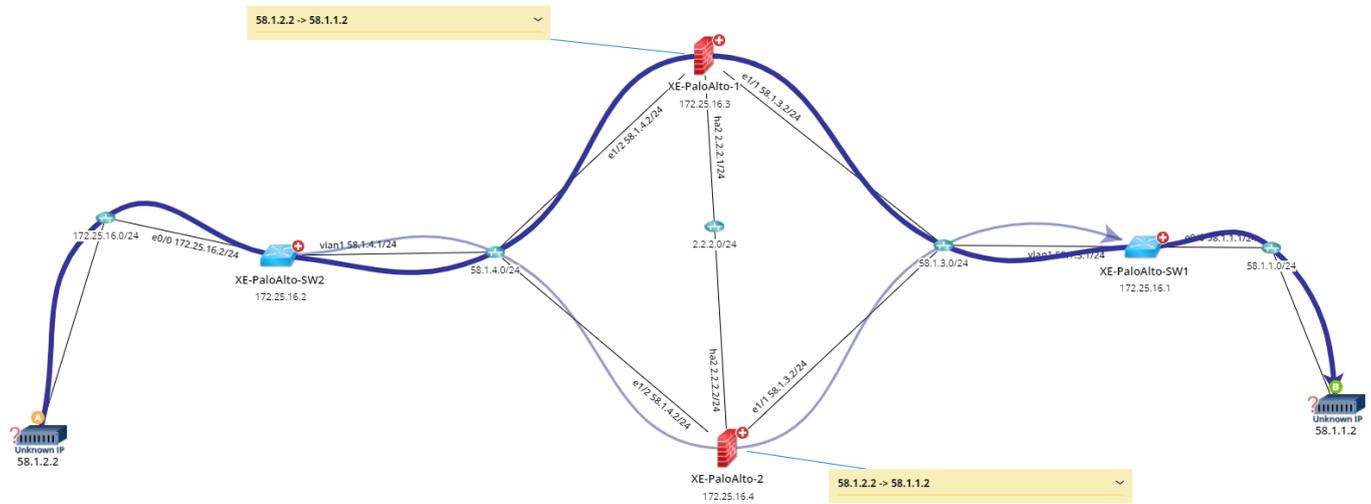


**Note:** The above logic is dedicated to duplicate IPs in HA/Cluster. If a duplicated IP exists in both HA/Cluster and other places of a network, users need to resolve the duplicate IP issues by assigning the duplicated IPs to different zones through the [Duplicated IP and Subnet Manager](#) feature.

To understand the data model of HA/Cluster and how it builds, refer to [Build a Data Model for HA/Cluster](#) for details.

## Calculate Path Across HA/Cluster

IEv8.0 is able to calculate a path across HA/Cluster. When calculating a path across HA/Cluster, the system can find the active device and the path traffic will always go through the active device.



## Build a Data Model for HA/Cluster

To calculate the topology and path for VPLS, IEv8.0 creates the following GDR and NCT mostly via drivers.

### HA/Cluster Device GDR:

The table below describes the HA/Cluster device GDR information used for topology calculation.

GDR ID	Type	Display Name	Description
isHA	bool		Indicate whether a device is configured with HA.
HA	object	HA	
HA.role	String	Role	The role of HA. The role is Primary or Secondary. If no role is configured, this GDR is null.
HA.hostName	String	HA Hostname	The HA name of HA devices. For example, a pair of ASA firewalls share an HA name.
HA.failoverIntf	string	Failover Interface	The failover interface information configured in HA configuration.
Intfs.isFailover	bool		Indicate whether an interface is a failover interface.
isCluster	bool		Indicate whether a device is configured with Cluster.
cluster	Object	Cluster	

GDR ID	Type	Display Name	Description
<b>cluster.name</b>	string	Cluster Name	The name or ID of a Cluster.
<b>cluster.unitName</b>	string	Cluster Node Name	The local unit name of a Cluster.
<b>cluster.clusterIntf</b>	string	Cluster Interface	The Cluster Interface information of Cluster configuration;
<b>nonDuplicateIP</b>	bool		Indicate whether an interface IP is a duplicate IP.  All HA/Cluster interfaces with this GDR value as True will join L3 topology calculation.

### NCT:

IEv8.0 records the HA state information used for path calculation in NCT.

The table below shows an example of HA State Table in the system.

HA Node	HA State	Is Active
<b>this</b>	master	True
<b>LASLCFWL01B</b>	slave	False
<b>LASLCFWL01C</b>	slave	False

## 5.5.5.Transparent Device

In previous versions, NetBrain supported the L2 topology calculation for Cisco ASA Transparent Firewalls and Riverbed WAN Optimizers and visualized the links, but the topology and link of Riverbed WAN Optimizers maybe not accurate in some customers' environment.

IEv8.0 has enhanced the L2 topology calculation for transparent devices by recording properties of transparent devices in GDR and calculating topology based on transparent device type and GDR properties. To understand the data model of transparent topology and how it builds, refer to [Build a Data Model for Transparent device](#) for details.

With the enhancement, IEv8.0 can calculate topology and path for Riverbed WAN Optimizers and Palo Alto Firewall in Virtual Wire mode.

**Note:** IEv8.0 only supports the transparent technology on Riverbed WAN Optimizer, Cisco ASA Firewall and Palo Alto Firewall (Virtual Wire). More vendors can be supported by drivers.



## Build a Data Model for Transparent Device

To calculate the topology and path for transparent devices, IEv8.0 creates the following GDR and NCT via drivers.

### Transparent Device GDR:

The table below describes the transparent device GDR information used for topology calculation.

GDR ID	Type	Description
<b>isTransparent</b>	bool	Determine whether one device is a transparent device. The judgment logic varies on vendors and needs to be defined in each device type driver.
<b>intfs.isTransparent</b>	bool	Determine whether one interface is a transparent interface. The judgment logic varies on vendors and needs to be defined in each device type driver.

## Topology Calculation Logic

IEv8.0 uses the following logic to calculate the L2 topology of Riverbed optimizers: If the **isTransparent** property of a device is true and the device type is the Riverbed optimizer, look up the MAC address that belongs to one interface of a network device from the Riverbed optimizer MAC table, and then build a L2 topology link by connecting the interface of the network device and the Riverbed optimizer interface that has learned this MAC address (of the interface of the network device).

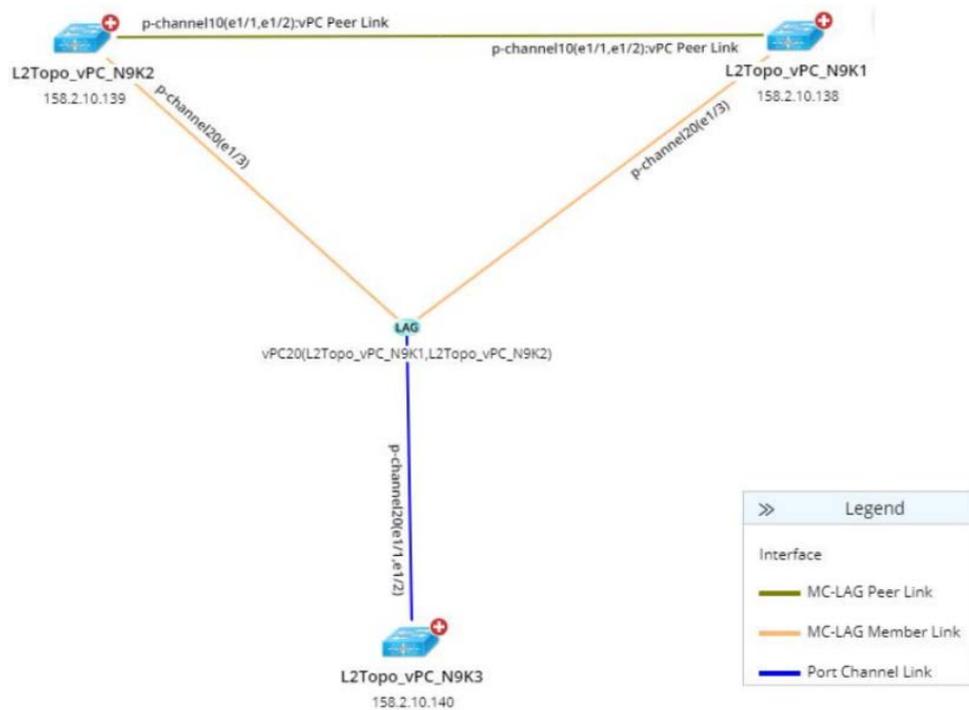
### 5.5.6.MC-LAG

MC-LAG (Multi-Chassis Link Aggregation Group) is a solution that allows link aggregation to form a logical LAG interface between two MC-LAG devices. In previous versions, the system supports only Cisco Nexus vPC, not

the MC-LAG technologies from other vendors. I Ev8.0 adds the support of the general MC-LAG technologies by adding a new media type “LAG” to show the connections between MC-LAG peers.

**Note:** I Ev8.0 only supports the following MC-LAG technology types. Other types of vendors can be added by NetBrain platform team.

- Cisco Nexus VPC
- Arista MC-LAG
- Dell VLT



## Build a Data Model for MC-LAG

To calculate the topology for MC-LAG devices, IEv8.0 builds the following GDR and NCT mostly via drivers.

### MC-LAG Device GDR:

The table below describes the MC-LAG device GDR information used for topology calculation.

GDR ID	Type	Description	Configuration Example
<b>hasMCLAGConfig</b>	bool	Determine whether one device is configured with MC-LAG. The judgment logic varies on vendors and needs to be defined in each device type driver. For Cisco Nexus Switch, the logic is that the interface configuration contains <i>vpc</i> .	
<b>intfs.MCLAG</b>	Object		
<b>intfs.MCLAG.imp</b>	String	The specific implementation name of the MC-LAG from a vendor. For example, the name is VPC for Cisco Nexus Switch.	
<b>intfs.MCLAG.number</b>	String	The associated number of MC-LAG member link.	interface port-channel89 switchport switchport mode trunk bandwidth 20000000 <i>vpc 89</i> no shutdown
<b>intfs.MCLAG.type</b>	String	Determine whether an interface is a peer link or member link.	interface port-channel100 description vpc-peer-link switchport switchport mode trunk bandwidth 80000000 <i>vpc peer-link</i> no shutdown

## MC-LAG Device Topology Calculation Logic

IEv8.0 uses the following logic to calculate the MC-LAG L2 topology: after the L2 topology link between devices is calculated through NDP tables, the system further determines whether the interfaces at both ends of a link is the MC-LAG peer link. If it is a peer link, the two devices are considered to belong to the same peer group. In

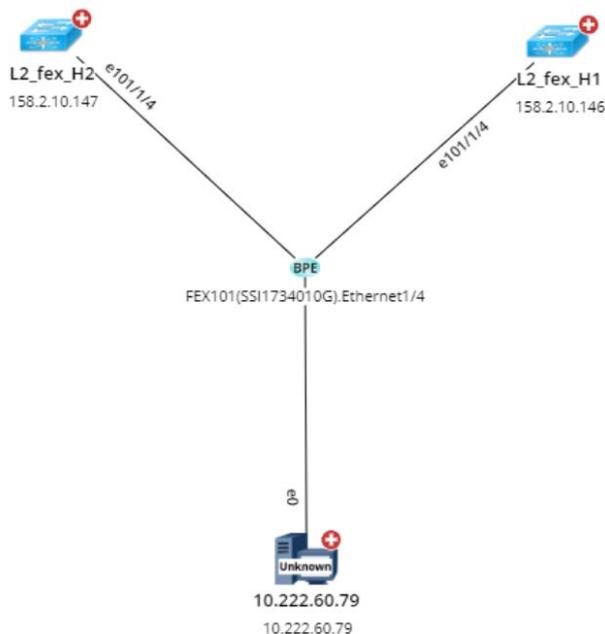
the peer group, the system generates a LAG media for the interfaces of the MC-LAG member link and connects the interfaces to the media. The more neighbor devices and interfaces of the MC-LAG member interfaces obtained via consequent NDP/MAC calculation will also be connected to the LAG media. The format of the LAG media is <intfs.MCLAG.imp> <Intfs.MCLAG.number> (peer group device names).

## 5.5.7.FEX Dual-Homed

IEv8.0 can calculate L2 topology and visualize the topology link for dual-homed Cisco Nexus FEX.

**Note:** IEv8.0 only supports the dual-homed configuration on Cisco Nexus FEX. More vendors can be supported by drivers.

It uses a new media type “BPE” (Bridge Port Extension) to show the topology connections of dual-homed devices.



## Build a Data Model for FEX Dual-Homed

To calculate the topology for FEX dual-homed devices, IEv8.0 creates the following GDR and NCT mostly via drivers.

### Dual-homed Device GDR

The table below describes the configuration information of the FEX dual-homed device in the GDR.

Schema Name	Type	Comments	Configuration Example
hasBPEConfig	bool	Determine whether one device is configured with bridge port extension. The judgment logic varies on vendors	

Schema Name	Type	Comments	Configuration Example
		and needs to be defined in each device type driver. For Cisco Nexus Switch, it is determined by the command " <i>show fex detail</i> ".	
BPE	Table		
BPE.imp	String	The specific implementation name of the MC-LAG from a vendor. For example, the name is FEX for Cisco Nexus Switch.	
BPE.number	INT	The number assigned to an extender device.	<pre>switch# show fex detail  FEX: 100 Description: FEX0100 state: Online  FEX version: 4.2(1)N1(1) [Switch version: 4.2(1)N1(1)]  FEX Interim version: 4.2(1)N1(0.309)</pre>
BPE.sn	String	The serial number of an extender device.	<pre>switch# show fex detail  FEX: 100 Description: FEX0100 state: Online  FEX version: 4.2(1)N1(1) [Switch version: 4.2(1)N1(1)]  FEX Interim version: 4.2(1)N1(0.309)  Switch Interim version: 4.2(1)N1(0.309)  Extender Model: N5K-C5110T-BF-1GE, Extender Serial: JAF1237ABSE</pre>
intfs.isBPE	bool	Determine whether an interface is a BPE interface	
intfs.BPENumber	INT	The number that a BPE interface belongs to.	
intfs.BPEIntf	String	The BPE name of an interface.	

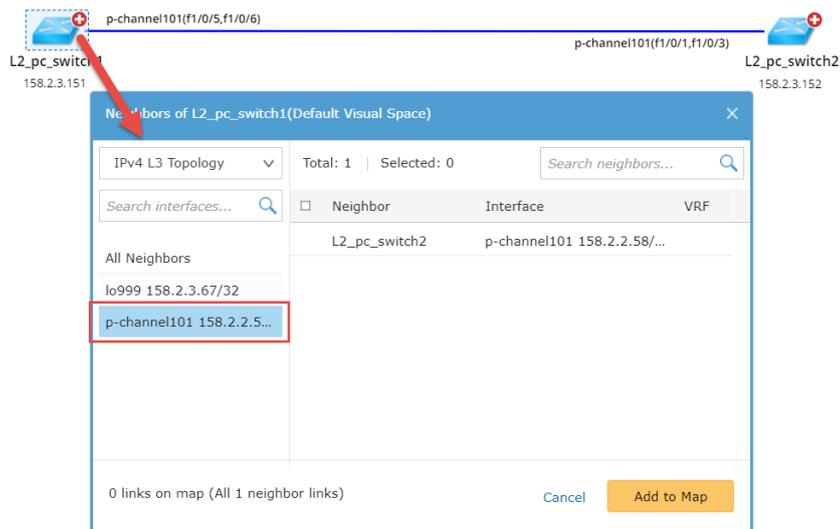
## Dual-Homed Device Topology Calculation Logic

IEv8.0 uses the following logic to calculate the FEX L2 topology for Cisco Nexus switches: after the L2 topology between an end system and its neighbor switch is calculate through MAC tables, the system determines whether the interface of the neighbor switch is the BPE interface. If true, the system generates a BPE media to connect the end system and the neighbor switch interface. As for the name of BPE media, its format is <BPE.IMP> <BPE.Number> (BPE.SN). <intfs.BPEIntf>.

### 5.5.8.Port Channel

A port channel is an aggregation of multiple physical interfaces that create a logical interface to increase the throughputs for improved network resiliency. In previous versions, NetBrain only supported the L2 topology calculation for physical interfaces.

IEv8.0 can calculate and visualize the topology for port-channel interfaces.



### Port-Channel Topology Calculation Logic

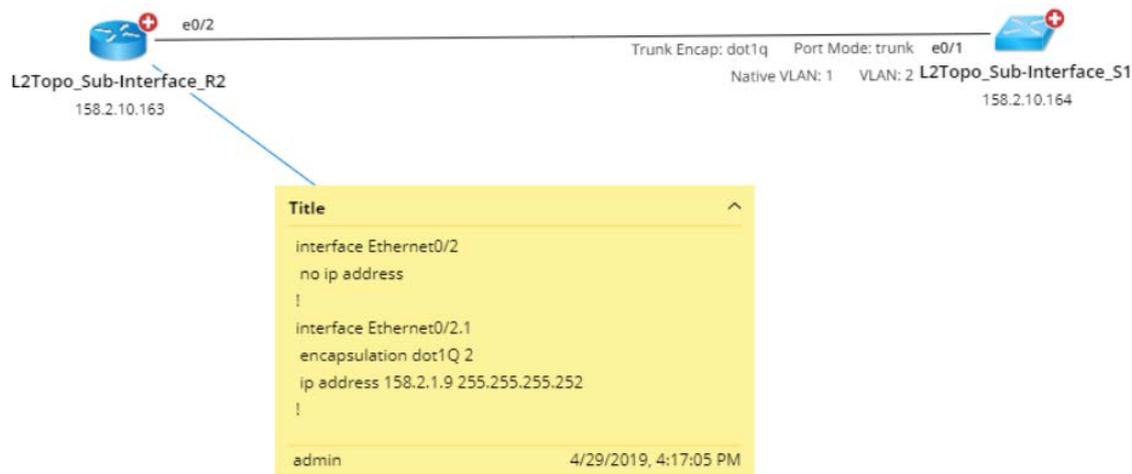
IEv8.0 uses the following logic to calculate the L2 topology for port channel interfaces: after the L2 topology of the physical interfaces is calculated through NDP or MAC table, the system further checks whether the interfaces at both ends of the topology link belong to a port channel interface. If true, the system establishes an L2 topology link for the port channel interfaces.

## 5.5.9. Unified Sub Interface in Topology Calculation

In previous versions, there was not a uniform standard about how to define the subinterface connection in an L2 topology calculation.

IEv8.0 unifies the subinterface topology calculation logic as follows:

- Use the main interface of a sub interface, instead of the sub interface itself, to generate the L2 topology link on maps.



- Use the neighbor interface of the main interface to populate the switchport that a sub interface (configured with IP address) connects to in One-IP table.

## Sub Interface Topology Calculation Logic

After calculating the L2 topology between a sub interface (with IP configured) and its neighbor switch through MAC tables, the system will use the main interface of the sub interface, instead of the sub interface itself, to generate the L2 topology link with the neighbor switch.

## 5.5.10. End System

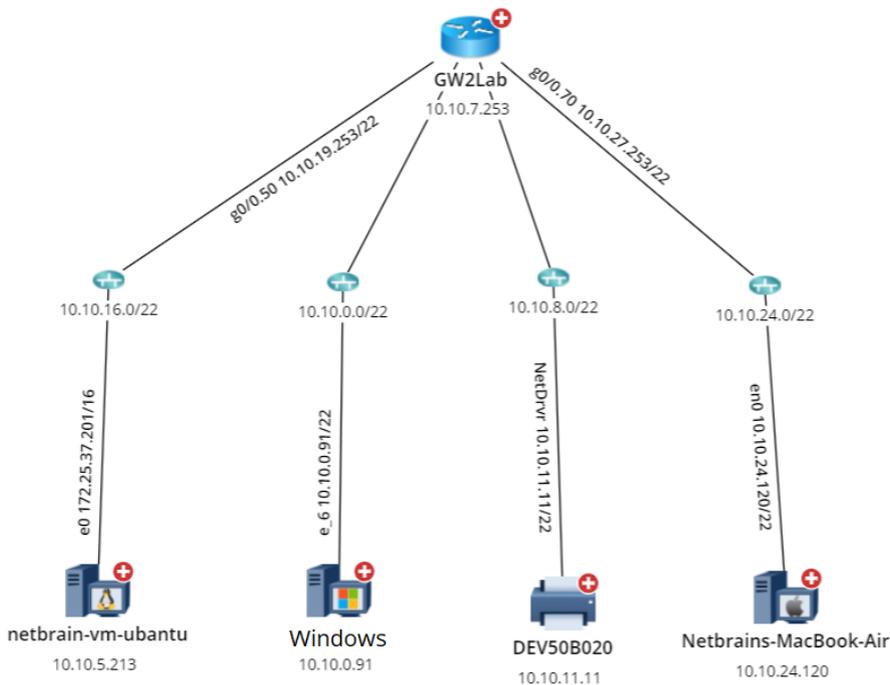
End System in NetBrain refers to an endpoint device (such as a printer, server, and computer) added to domains through a discovery. IEv8.0 improves the end system accuracy from the following aspects:

- [Subdivide End System Types](#)
- [Enable CLI Access Capability for End System running Linux](#)
- [Support Building L3 Topology for Unknown End Systems](#)

- [Optimize SNMP discovery of End Systems](#)
- [Assign IPv4 Gateway to End System for Path](#)
- [Visualize Applications Running on End Systems](#)
- [Optimize End System Management in One-IP Table](#)
- [Add Controls to Remove One-IP Entries without Port Values](#)

## Subdivide End System Types

IEv8.0 subdivides the device type of End Systems, such as Printer, Windows, Ubuntu, and Mac. Users need to enable SNMP on end systems because the system retrieves the sub device type of an end system by using SNMP OID.

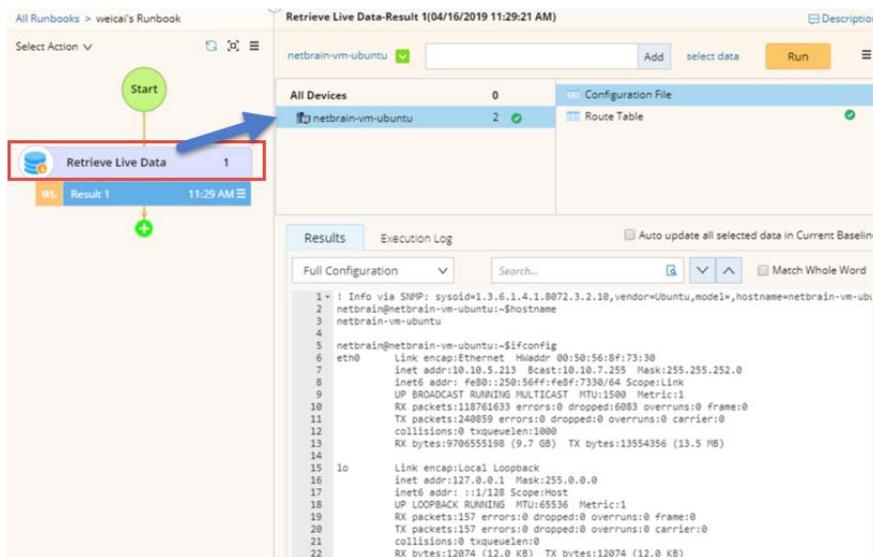


**Tip:** IEv8.0 offers one built-in Linux-based device type Ubuntu and has the capability of supporting more Linux-based device types per customer requests, such as RedHat and CentOS.

## Enable CLI Access Capability for End Systems Running Linux

IEv8.0 allows CLI access to End Systems running Linux (currently only supports Ubuntu) via the device driver, which enables users to obtain device configuration, system tables and more device/interface properties of End

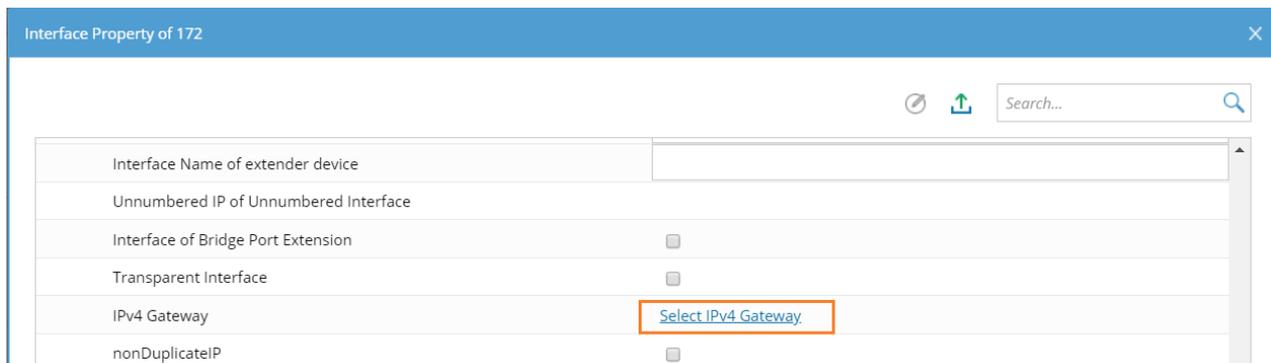
Systems for dynamic mapping and automation.



## Assign IPv4 Gateway to End System for Path Calculation

The A/B path calculation is mostly from an end to another end. When end users calculate paths, they need to select a gateway for an end system. While in most cases it is difficult for them to determine which gateway is the right one.

To enable power users to batch predefine the IPv4 gateways for end systems so that the system can automatically load the correct gateway during a path (including path calculation in AAM), IEv8.0 adds an interface property for gateway setting.



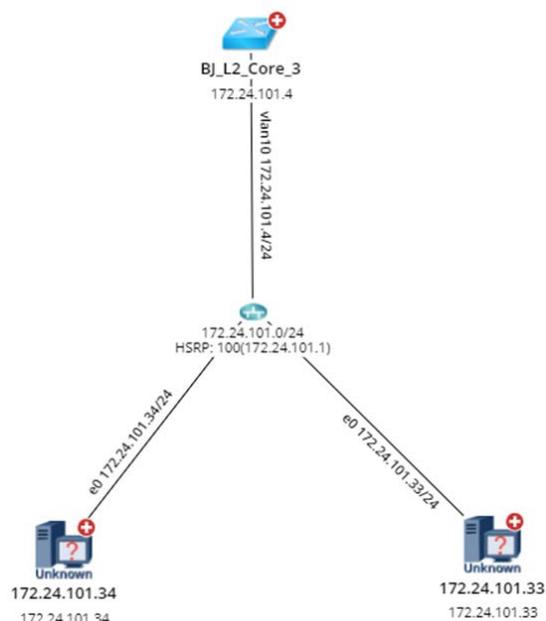
The table below shows the Gateway GDR of an End System.

GDR ID	Type	Description
intfs.ipgateway	string	Define the IP gateway of an end system interface for path calculation.

## Support Building L3 Topology for Unknown End Systems

To enhance the accuracy of path and topology calculation involving Unknown End Systems, IEv8.0 adds an option to allow building their L3 topology. If this option is enabled, the system will calculate L3 topology for unknown end systems when it builds L3 topology in a benchmark task.

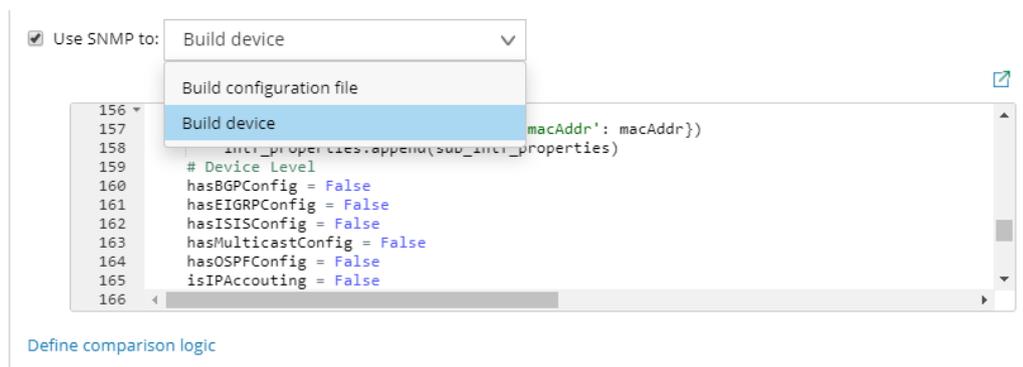
This option is disabled by default. To enable it, go to **Domain Management > Operations > Advanced Settings > Build L3 Topology Option**.



## Optimize SNMP Discovery of End Systems

In previous versions, when using SNMP to add an end system to a domain, the device driver retrieved SNMP data with python scripts and then parsed it with another programming language. This processing method had bad extensibility and caused the difficulty of adding more information to end systems.

IEv8.0 has improved SNMP discovery for end systems. It introduces the **Build Device** function in the driver that users can write python scripts to retrieve device data, build a device data model and finally add a device to domain with this function.

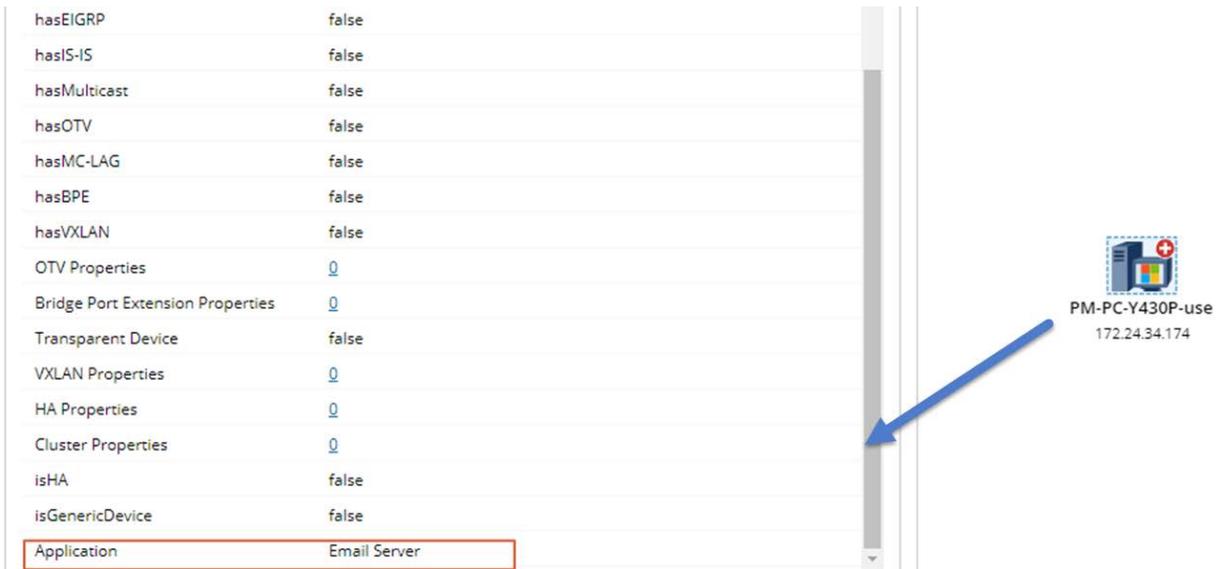


## Visualize Application Service Running on End System

IEv8.0 adds a new property "Application" in the GDR of an end system, which enables users to record the application or service running on the end system. Users can view or edit the application deployed on an end

system in Device Details Pane.

hasEIGRP	false
hasIS-IS	false
hasMulticast	false
hasOTV	false
hasMC-LAG	false
hasBPE	false
hasVXLAN	false
OTV Properties	<a href="#">Q</a>
Bridge Port Extension Properties	<a href="#">Q</a>
Transparent Device	false
VXLAN Properties	<a href="#">Q</a>
HA Properties	<a href="#">Q</a>
Cluster Properties	<a href="#">Q</a>
isHA	false
isGenericDevice	false
Application	Email Server



**Example:** End System Gateway GDR.

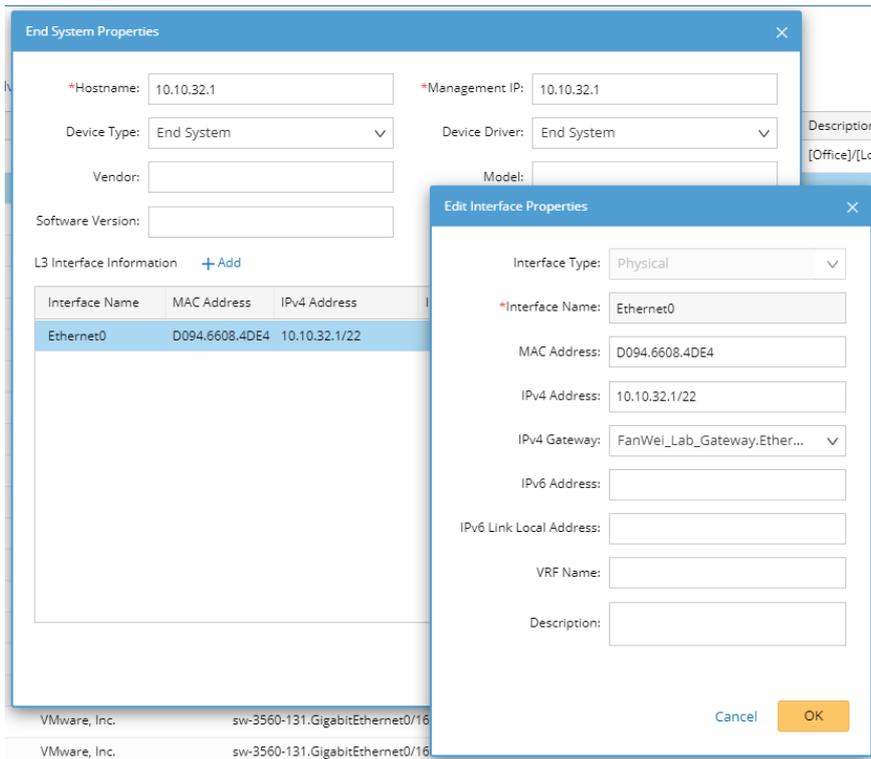
GDR ID	Type	Description
application	string	Define the application service running on the end system.

**Note:** The system will not automatically populate the application value and users need to set it manually.

## Optimize End System Management in One-IP Table

- Add a **Show Unknown End System Only** filter to only show unknown End Systems in the One-IP table.

- Support to convert an Unknown End System to an End System. When adding an unknown end system to end system entries, users can predefine gateways for it.



## Add Controls to Remove One-IP Entries without Port Values

When calculating L2 topology, the system may save lots of entries without switch ports and DNS names in the One-IP table. To decrease the storage size and increase the query efficiency of the One-IP table, the system adds an **Only save One-IP table entries that have values in Switch Port or DNS Name** parameter option.

To set this option, go to **Domain Management > Operations > Advanced Settings > Build L2 Topology Option**.

### 5.5.11. Wireless

IEv8.0 optimizes the discovery, topology and path calculation for wireless devices.

**Note:** IEv8.0 only supports Cisco WAP/WLC/LWAP devices.

- Discover LWAPs via WLC device drivers.

IEv8.0 can flexibly obtain and manage LWAP data in domains and provide the ability to support more LWAP types. It defines the function to discover LWAPs in WLC driver Advanced Script. In the Advanced Script, the system can obtain the accurate Ethernet interface for LWAP, and parse the SSID connected

to the wireless endpoint in the LWAP as the wireless interface of the LWAP, and add the interface property (intfs.isWlan) to the SSID wireless interface.

**Example:** Wireless Interface Property.

GDR ID	Type	Description
intfs.isWlan	bool	Identify if the interface is a WLAN interface.

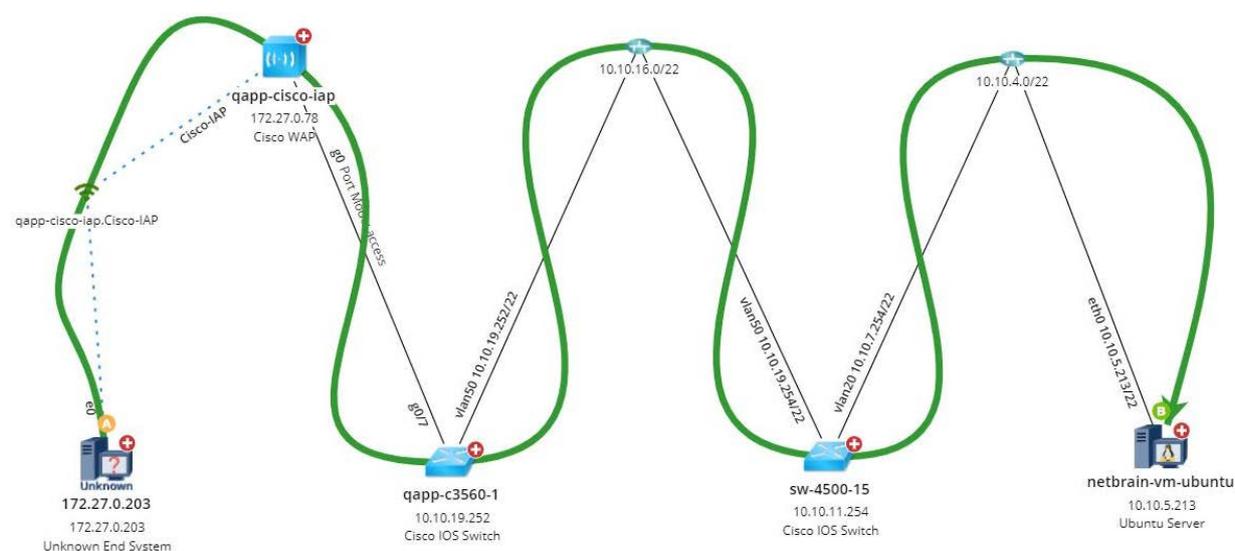
**Note:** The system does not discover LWAP devices by default. To discover the LWAPs, it needs to uncheck the LWAP device type in Do-Not-Scan.

- Support to retrieve the data of LWAPs via WLC in a Benchmark or Scheduled DVT/Parser task.

The screenshot shows a workflow in NetBrain. On the left, a workflow diagram starts with a green 'Start' circle, followed by a blue 'Retrieve Live Data' node with a '1' next to it, and a blue 'Result 1' node with a timestamp of '09:28 AM'. On the right, the '2 Devices' section shows two devices: 'APc471.fe20.4206' and 'APc471.feb0.b5fc', both with a green checkmark. Below this, the 'Results' section shows a table with the following data:

Local Interface	Device Name	Interface Name	Interface Address
FastEthernet0	BJ_L2_test_1.netbrain.com	FastEthernet1/0/15	172.24.34.62

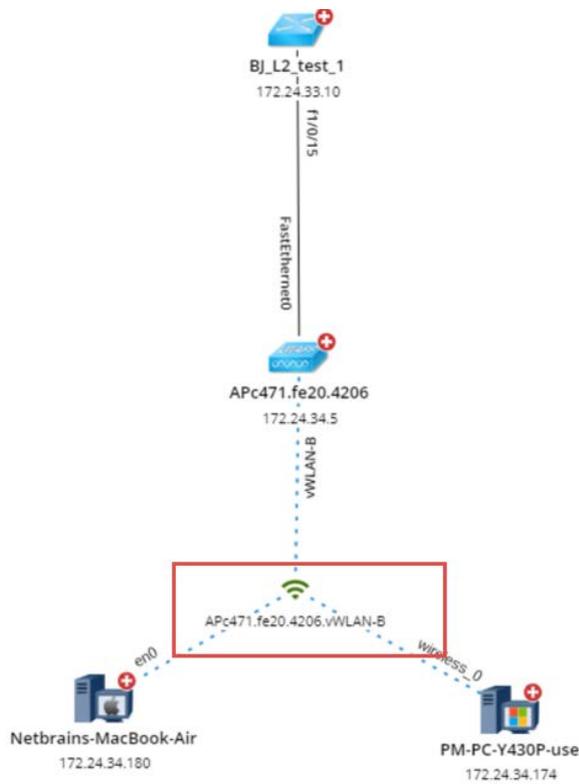
- Visualize the WAP device that an L2 path cross when users calculate A-B path between two wireless endpoint devices.



**Note:** IEx8.0 only supports the wireless path calculation across Cisco WAP devices.

- Visualize the L2 link between a wireless endpoint and its AP (LWAP/ WAP) via a WLAN media named after LWAP/WAP and SSID name.

**Note:** When the system builds L2 topology during a benchmark, it calls L2 Wireless Topology Qapp to create an L2 topology link between LWAP/WAP and wireless endpoint.



- Add the LWAP Summary NCT table for WLCs to help users view the information of APs connected to a WLC in a domain.

**qapp-vwlc**  
Cisco-Virtual Wireless LAN Controller

Compare Export Note

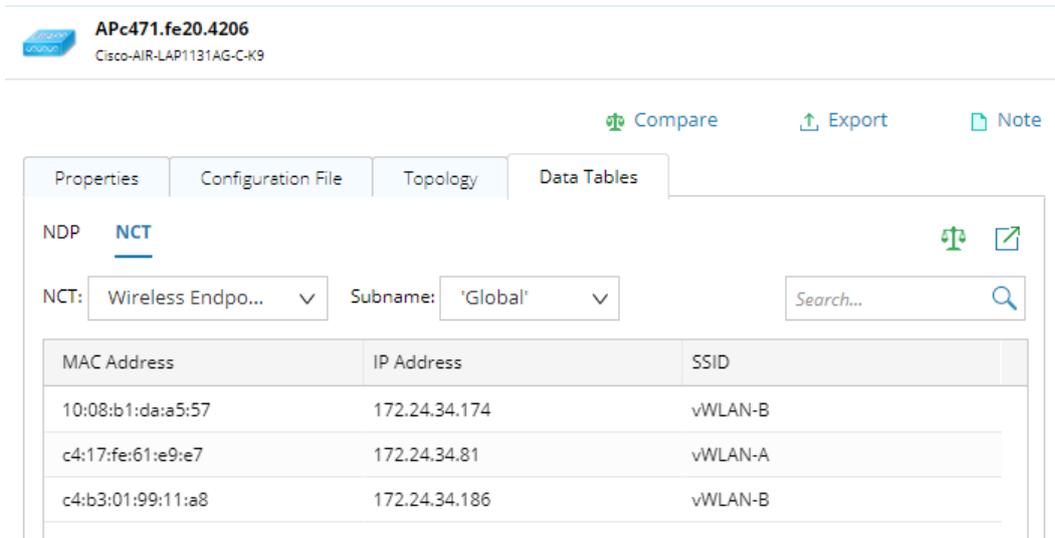
Properties Configuration File Topology Data Tables CLI Commands

Route ARP NDP **NCT**

NCT: LWAP Summar... Subname: 'Global' Search...

LWAP Device	LWAP MAC	LWAP IP Address
APc471.fe20.b5fc	c4:71:fe:b0:b5:fc	172.24.34.59
APc471.fe20.4206	c4:71:fe:20:42:06	172.24.34.5

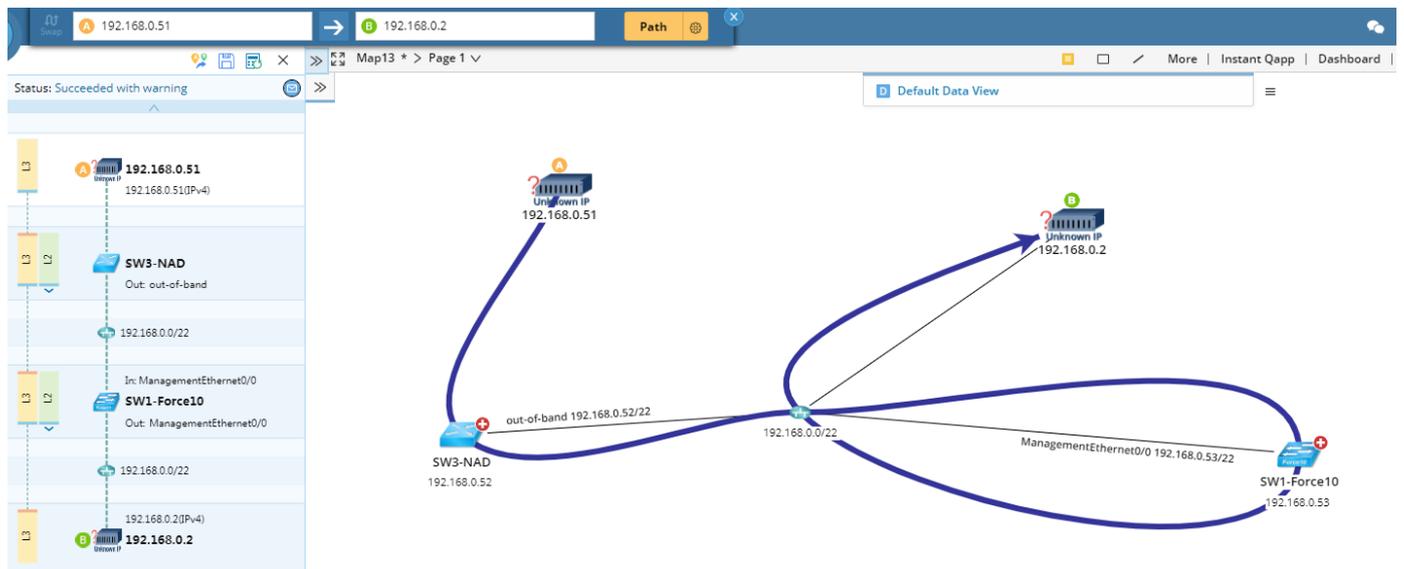
- Add Wireless Endpoint NCT for AP to help users view the information of wireless endpoints connected to LWAP/WAP in a domain.



**Note:** The Wireless Endpoint Summary Table affects the L2 topology calculation between LWAP/WAP and Wireless, and the L2 path calculation of WAP will also call this table.

## 5.5.12. Management Route

IEv8.0 can retrieve the management routes of Dell Force 10 and F5 Load Balancer and calculate paths across the devices with the management routes.



The screenshot shows a network management console for 'SW1-Force10'. On the left, a 'Device Log' panel lists several diagnostic steps. Step 5, 'Look up Routing Table (Management Route)', is highlighted with an orange box and an arrow pointing to the main table. The main table, titled 'Management Route Table of SW1-Force10', displays the following data:

Destination	Interface	Next Hop IP	State	Route Source
0.0.0.0/0		192.168.0.1	Active	Static
192.168.0.0/22	ManagementEthernet0/0		Connected	Connected

**Note:** In this case, the fix-up route table is required to be manually set up for the device “SW3-NAD” as follows.

The screenshot shows a configuration window for a 'Fix-up Route Table of SW3-NAD'. It includes a search bar, a dropdown menu set to 'Global', and a table with the following entry:

Dest.Addr	Mask	Interface	Next Hop IP	Next Hop Device
192.168.0.0	24	out-of-band	192.168.0.53	SW1-Force10

The table below shows an example of the Management Route Table in the system.

Destination	Interface	Next Hop IP	State	Route Source
1.1.1.1/24	ManagementEtherent 0/0	N/A	Connected	Connected
0.0.0.0/0	N/A	3.3.3.3	Active	Static
2.2.2.2/16	N/A	3.3.3.3	Active	Static

### 5.5.13. SPB

Shortest Path Bridging (SPB), the replacement for the older Spanning Tree Protocol, is a layer 2 technology intended to simplify the creation and configuration of networks while enabling multipath routing.

IEv8.0 can retrieve the SPB MAC addresses and calculate L2 paths based on SPB MAC addresses in a Layer 2 network adopting the SPB technology.

**Note:** I Ev8.0 only supports the SPB on Avaya Switch and Avaya VSP.

The screenshot displays the NetBrain interface. On the left, the 'Device Details' window for 'SW\_redNet\_54' is open, showing the 'NCT' (Network Configuration Table) tab. The table lists MAC addresses, VLAN IDs, BVLAN IDs, interfaces, and destination system MACs. On the right, a network topology diagram shows a central switch 'SW\_redNet\_54' connected to other devices like 'CH-GN-DC-01-02\_CH-GN-Webint-Sued-RN' and 'fs-Itmgn-hsdc-02'.

MAC Address	VLAN ID	BVLAN ID	Interface	Destination System...
0000.0002.6400	901		2/1	0010.0255.38F
0000.0002.6401	901		MLT-48	-
0000.0007.8900	901		2/1	0010.0255.38F
0000.0007.8901	901		MLT-47	-
001c.7f00.0264	901		MLT-48	-
001c.7f00.0789	901		2/1	0010.0255.38F
001c.7f42.bc33	901		MLT-47	-
001c.7f42.bfb3	901		2/1	0010.0255.38F
001c.7f67.7c39	901		2/1	0010.0255.38F
001c.7f67.f3b1	901		MLT-48	-
001c.7f69.5806	901		MLT-48	-
0000.0000.8900	999		2/1	0010.0255.40F
0000.0000.8901	999		MLT-41	-
0000.0001.8900	999		2/1	0010.0255.40F
0000.0001.8901	999		MLT-41	-

The table below shows an example of the SPB MAC Table in the system.

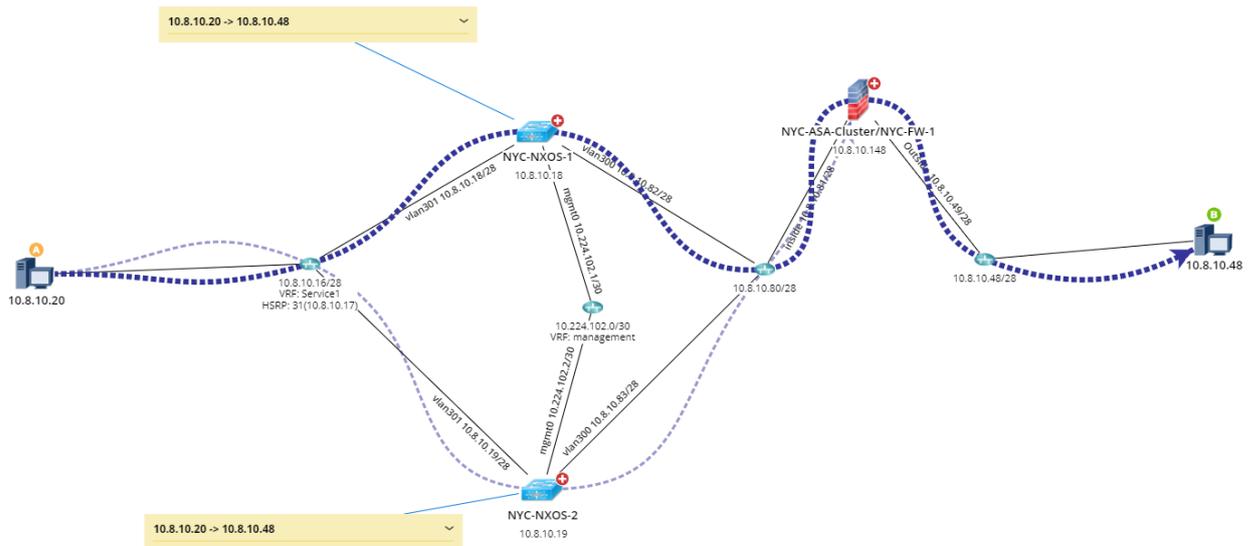
MAC Address	VLAN ID	BVLAN ID	Interface	Destination System MAC
<b>0102.0304.0506</b>	N/A	0010	1/00	0605.0403.0201

When looking up the destination MAC during path calculation, the systems will iterate the SPB MAC Table until a destination MAC is matched and then start the SPB path calculation.

## 5.5.14. HSRP Improvement

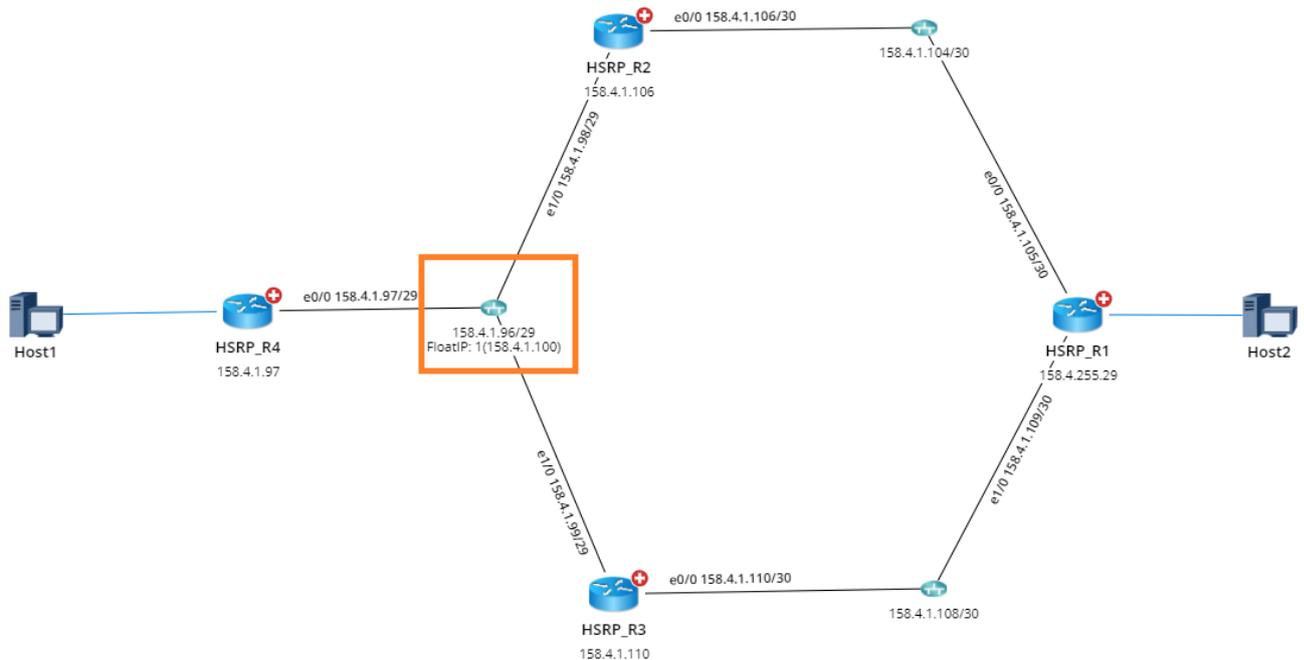
I Ev8.0 improves the HSRP technology support on Cisco Nexus Switch in the following aspects:

- Support to calculate a path in HSRP Active/Active with vPC configured. When calculating a path, the system can recognize both active links and calculate paths for both.



- Support customizing the media name and specify any information in the media name.

**Example:** Customize to show the floating IP in the HSRP media name.



## 5.5.15. Checkpoint Firewall R80

Since the data of the CheckPoint Firewall cannot be retrieved comprehensively based on the original OPSEC protocol after upgrading to the R80 version, it is recommended to use the API for more information. I EV8.0 supports retrieving the following NCT data of Checkpoint Firewall R80 via APIs:

- NAT table
- IPsec VPN table
- Bridging Group

## Discover CheckPoint Firewall R80 by Both Driver and TechSpec

---

In order to retrieve the NCT data of the CheckPoint Firewall R80 via APIs and solve the problem that the device discovered first by TechSpec or Driver, IEv8.0 supports the discovery of the CheckPoint Firewall R80 based on both TechSpec and driver: TechSpec discovery only fills the device name, management IP and other basic data, while Driver discovery will populate more information. The following issues will occur when users only use one of the two discovery methods:

- When CheckPoint Firewall R80 is only discovered via TechSpec, the CLI and SNMP information in Shared Device Setting will be missing. It is required to execute a tune to obtain the information.
- When CheckPoint Firewall R80 is only discovered via driver, the API Server information will be missing in the Shared Device Setting; It is required to execute the TechSpec discovery to obtain the information.

In summary, it is recommended to execute CLI discovery first and then execute API discovery when the CLI discovery is completed. After the API discovery is done, execute a benchmark to obtain the data of Checkpoint R80.

**Note:** IEv8.0 distinguishes Checkpoint Firewall R80 and previous Checkpoint Firewall as two device types, because the benchmark to obtain NCT data of Checkpoint Firewall in previous versions is based on the logic of OPSEC. If the device type of a device is CheckPoint Firewall during the benchmark, the system will always retrieve all data based on OPSEC; If not to distinguish them as different device types, the system will also retrieve data of Checkpoint Firewall R80 based on OPSEC besides retrieving the data based on APIs.

## Retrieve NCT by TechSpec

---

IEv8.0 supports retrieving CheckPoint Firewall R80 via API, CLI, and SNMP. To control which method is used to obtain data, IEv8.0 has defined the priority of these three ways to obtain data: API>CLI>SNMP. This priority settings enable NCT data to be always retrieved via APIs and other system tables that do not support API retrieved via CLI/SNMP.

**Note:** The priority of these three ways to obtain data also solves the problem that the NCT data of Leaf devices can be only retrieved via CLI.

## Merge the Display of TechSpec and Driver NCT in Benchmark Data Retrieval

---

The NCT control options to retrieve NCT via TechSpec in IEv7.1a are listed separately in Benchmark data retrieval based on TechSpec types (and the data obtained via Techspec is no longer supported in Runbook's Retrieve live data). IEv8.0 combines all NCTs under the NCT category, no longer distinguishing NCTs by TechSpec types.