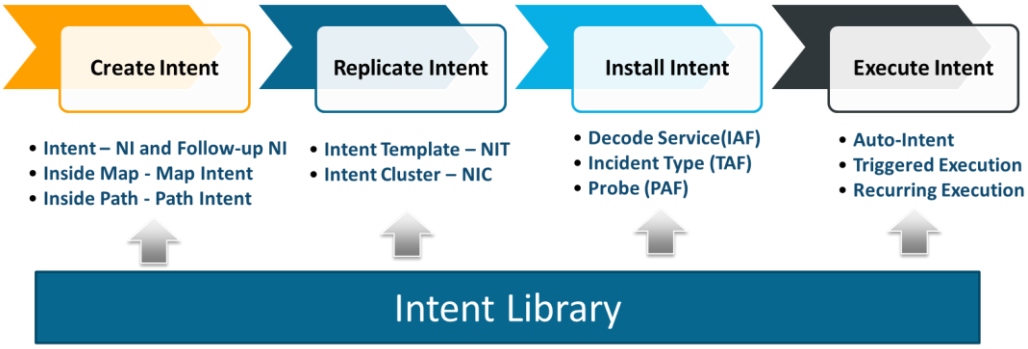
NetBrain® Release 11

Release Note

Table of Contents

1 Key New Features and Enhancements

NetBrain R11 greatly enhances the Intent-based **Problem Diagnosis Automation System (PDAs),** which automates the Diagnosis of repetitive problems and enforces preventive measures across the entire network. Especially, R11 improves the flow of building a PDAS system on a large scale with the following new features and enhancements:



- Create Intent

  R11 continuously improves the NI editor: allows using the verified parsers, supports NIC as the follow-up diagnosis, adds the inline expression, provides a better table-based diagnosis, etc. Also, a standalone Intent Manager is added to manage the common intents, path intents and newly added map intents.

- NIC Enhancements

  R11 adds two new features of the Intent Cluster (NIC), auto mode and auto test. The Auto Mode reduces NIC definition from 7 steps to three simple steps, and the auto test simplifies the definition of the device classifier and eigen grouping.

- Intent Template (NIT) New

R11 introduces NIT, a breakthrough that greatly simplifies the intent's replication. Users can define NIT for any NI with the single-device diagnosis by setting the device qualifications and the critical variables to be auto-tested. The intent with the template configured can be cloned on the fly after it is decoded and installed in the Intent Library to be executed interactively (IAF), Triggered Automation (TAF), and Preventive Automation (PAF).

- Intent Library New

  The intent library is the central console to install Intent and NIT in the PDAS system: enable the backend decoding service to qualify the devices against the template setting; update the baseline data; enable the auto intent to create the intents in the map (Auto Intent for the IAF); install the intent in TAF and PAF; execute the configure orchestration files downloaded from NetBrain KC to download and install the intents automatically.

- Interactive Automation (IAF)

  R11 greatly improves the Intent pane of the map with three new features: Auto Intent New, creating the intents from the qualified NIT for the map devices on the fly; Published Intent New, displaying the published intents for the map devices; Map Intent New, a dedicated NI for the map.

- Triggered Automation (TAF)

  R11 introduces a new Incident Type, Virtual Incident Type (VIT), to encapsulate the detailed logic of Hash Tag and remove the hurdle of the diagnosis by the Hash Tag, which can be associated with the incoming Incident Type to match intents by the Hash Tag. The self-service applications can use VIT directly.

- Preventive Automation (PAF)

  The intent timer is added to the PAF, and the intent template can be installed in PAF.

- Execute and Use Intent

R11 greatly improves the Diagnosis Tree: add device nodes between the diagnosis node and NI node and add the detail pane for each node, such as the device and diagnosis pane. The intent results can also be displayed as a special data view on a map.

- Self-Service Tools

  Besides improving the Incident Pane/portal, R11 creates a new self-service, Teams Chatbot [New], as a simple way to run NetBrain automation functions and view the results in teams.

- Incident-based Collaboration Workflow

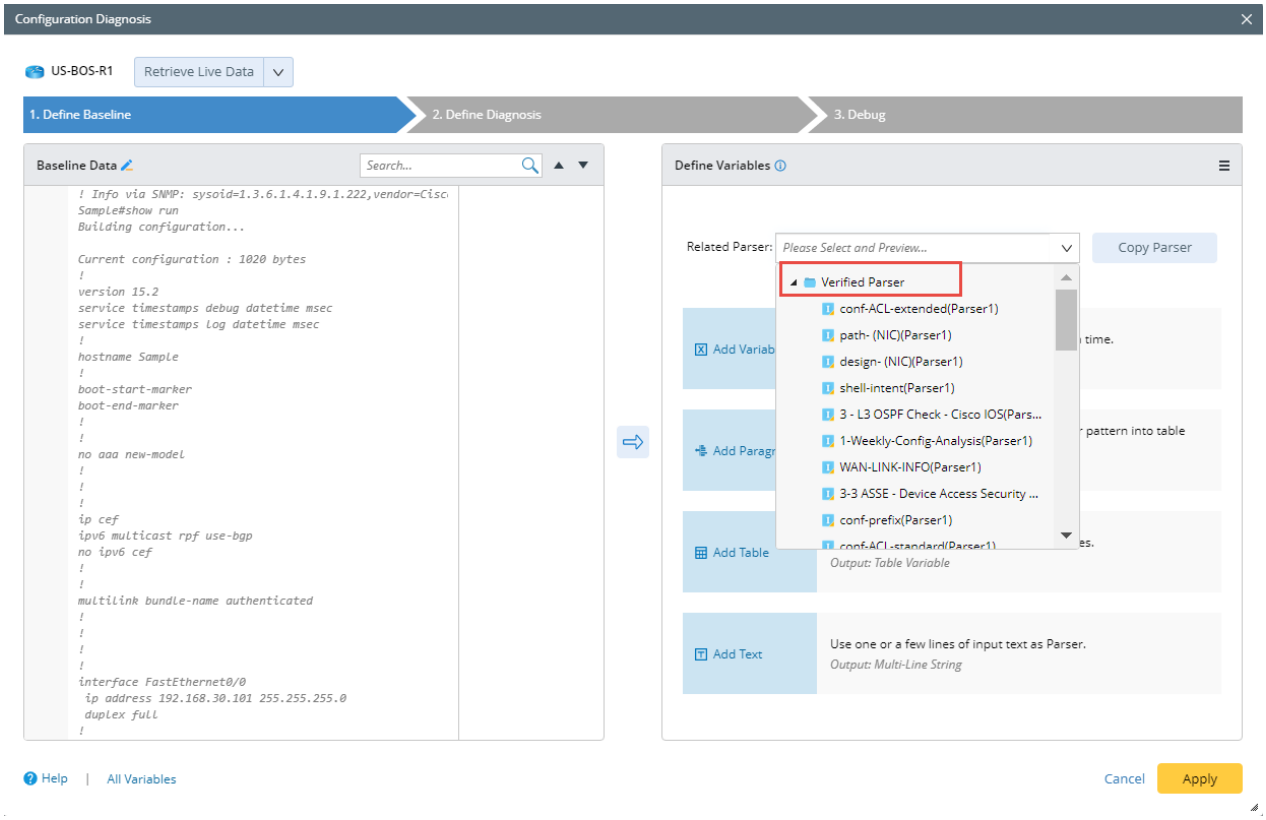
  R11 Improves the search IntelliSense pane to show more related results and allows users to create the map directly from an entry. A new Incident entry is added to the map.

2 Create Intent

The first step in building the PDAS system is to define the base NI. R11 makes many improvements to ease configuring diagnosis and better support the table formatted data.

2.1 Copy Verified Parser

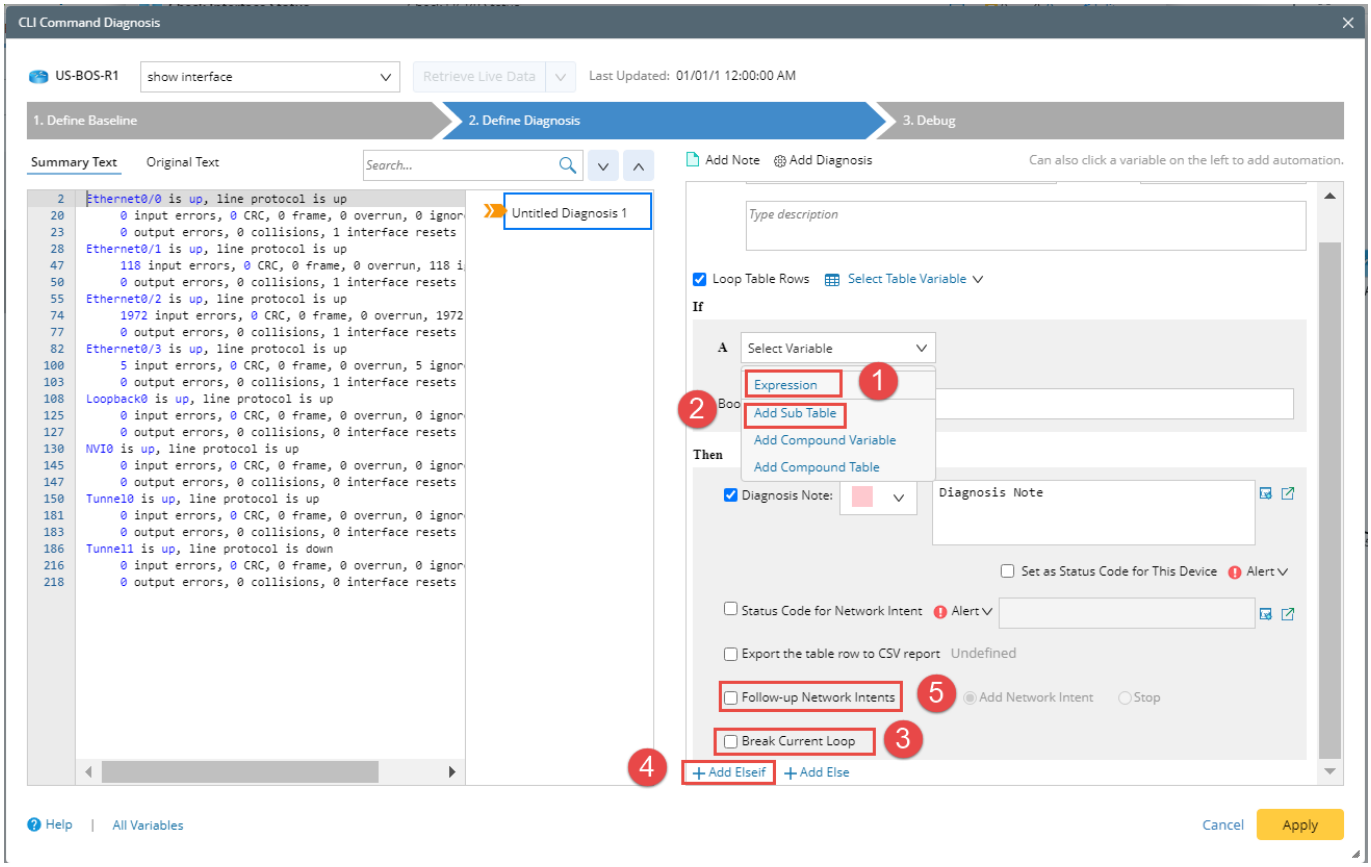
In the first step of **Configuration Diagnosis, Define Baseline**, R11 support copying the Verified Parser besides the library. The verified parsers are from the results of the Intent Template (NIT) decode engine. The system already verifies the historical commands and the corresponding parser.



2.2 Enhance Diagnosis Definition

In the second step of **Configuration Diagnosis**, **Define Diagnosis**, R11 provides many enhancements such as:

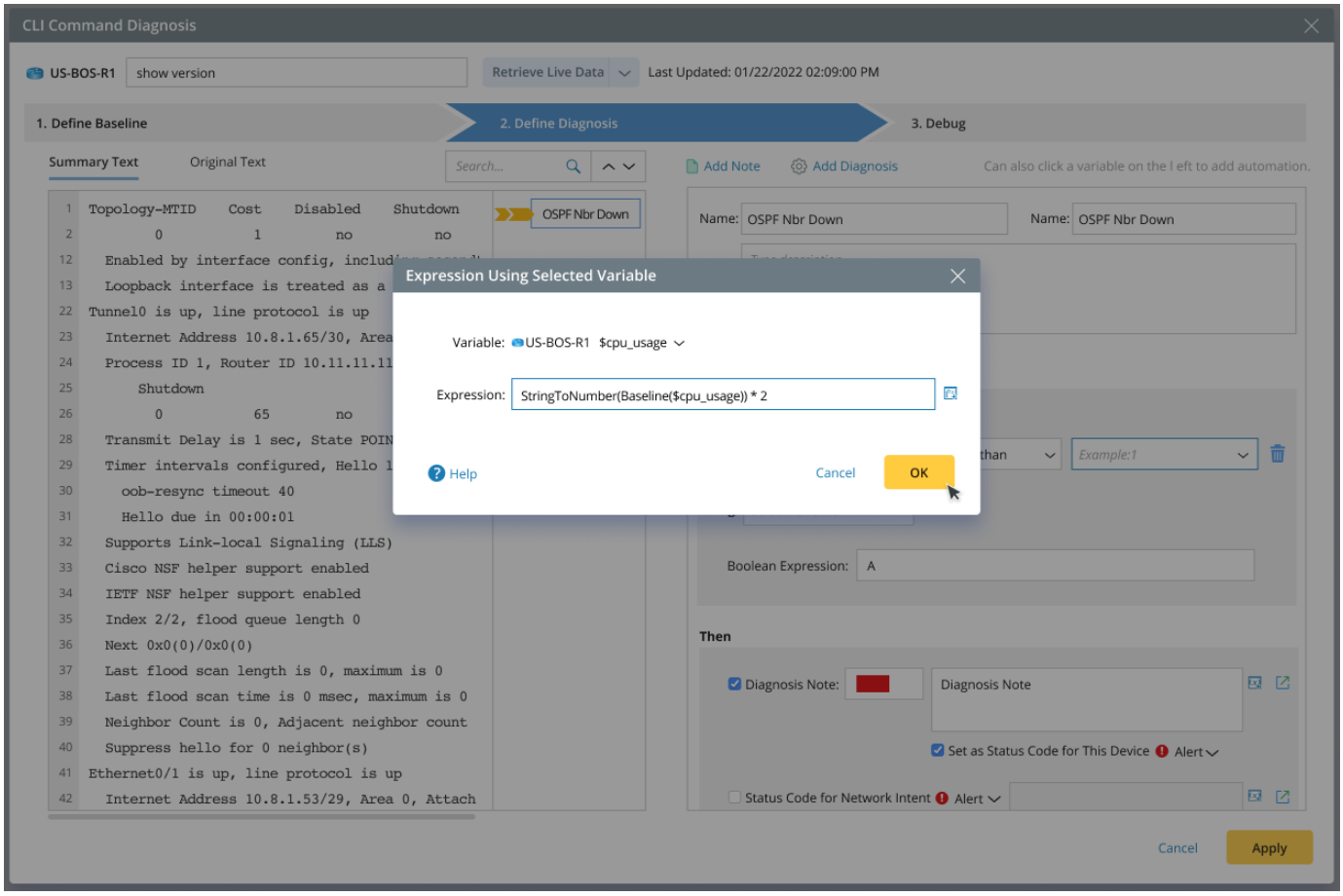
1. Support the inline expression for the variables.
2. Support the sub table creation (when the option **loop table rows** is enabled) and the whole table operations (when the option **loop table rows** is disabled).
3. Support break the current loop of a table.
4. Support Elseif branch.
5. Support NIC as the follow-up intent.



2.2.1 Support Inline Expression

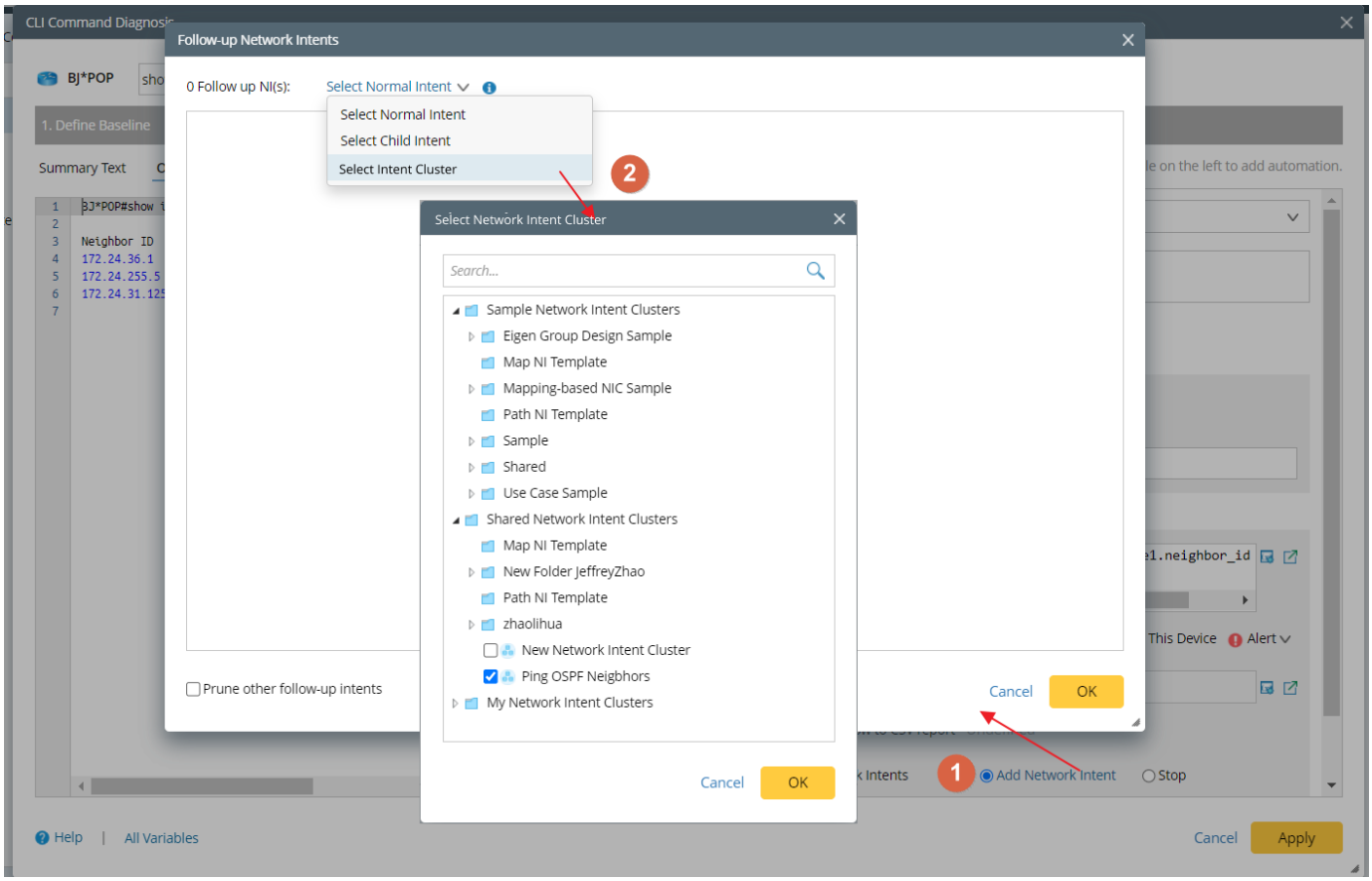
Users can use the inline expression in the definition of a diagnosis instead of defining a compound variable and then using that variable in the diagnosis.

The inline variable expression supports the simple math operation (**+**, **-**, *****, and **/**)and built-in functions, which are expanded to manipulate the string, IP address, MAC address, interface name, etc.

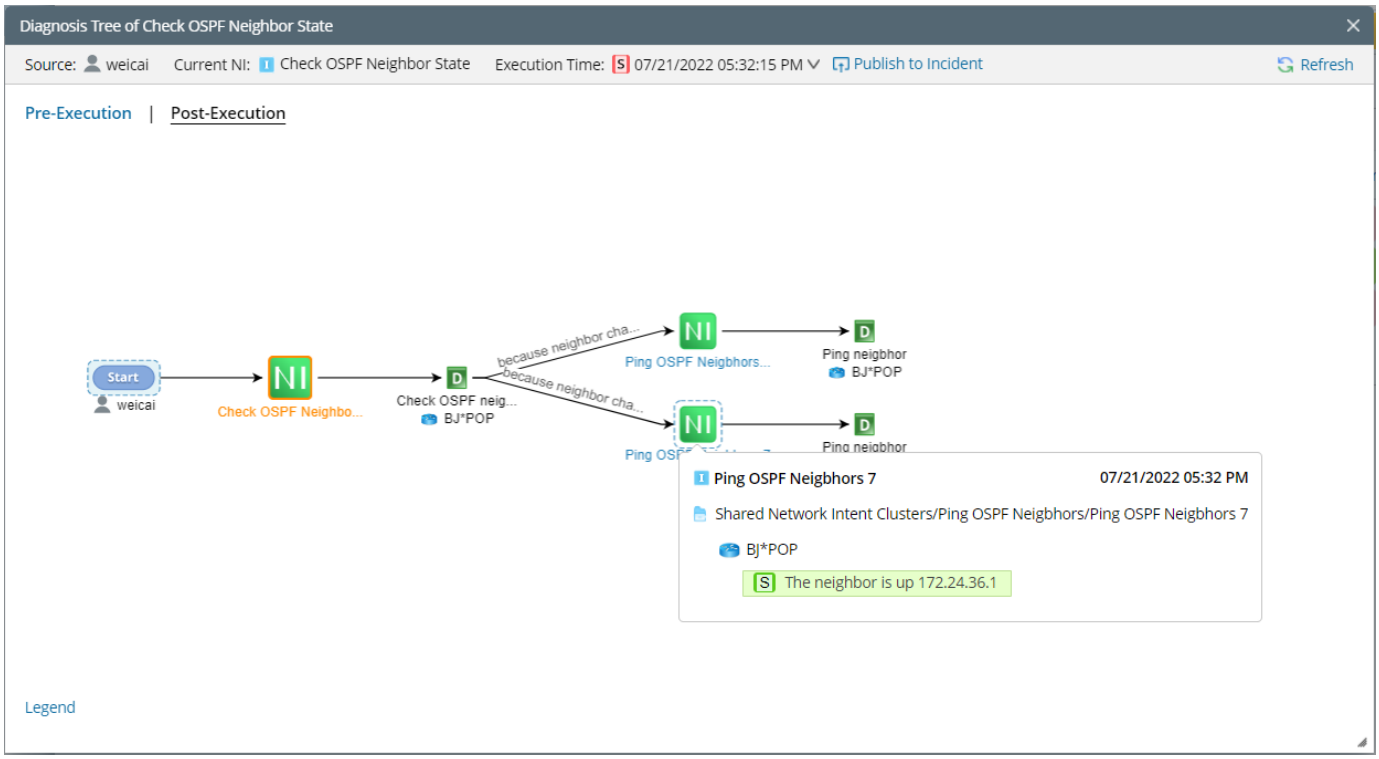


2.2.2 Support NIC as the follow-up NI

In the real world, much of the diagnosis logic is done at the interface/port level, and users could have many interface-based NI per device. Therefore, the follow-up NI should be dynamically selected from a NIC by parameters to fill the gap that NI cannot use the parameter to drive the “next-step” execution. For example, for each entry changed in OSPF neighbors, a user wants to do a “Ping to state changed neighbor” in follow-up NIC with macro variable and pass the obtained value to command: `ping $neighbor_ip`.

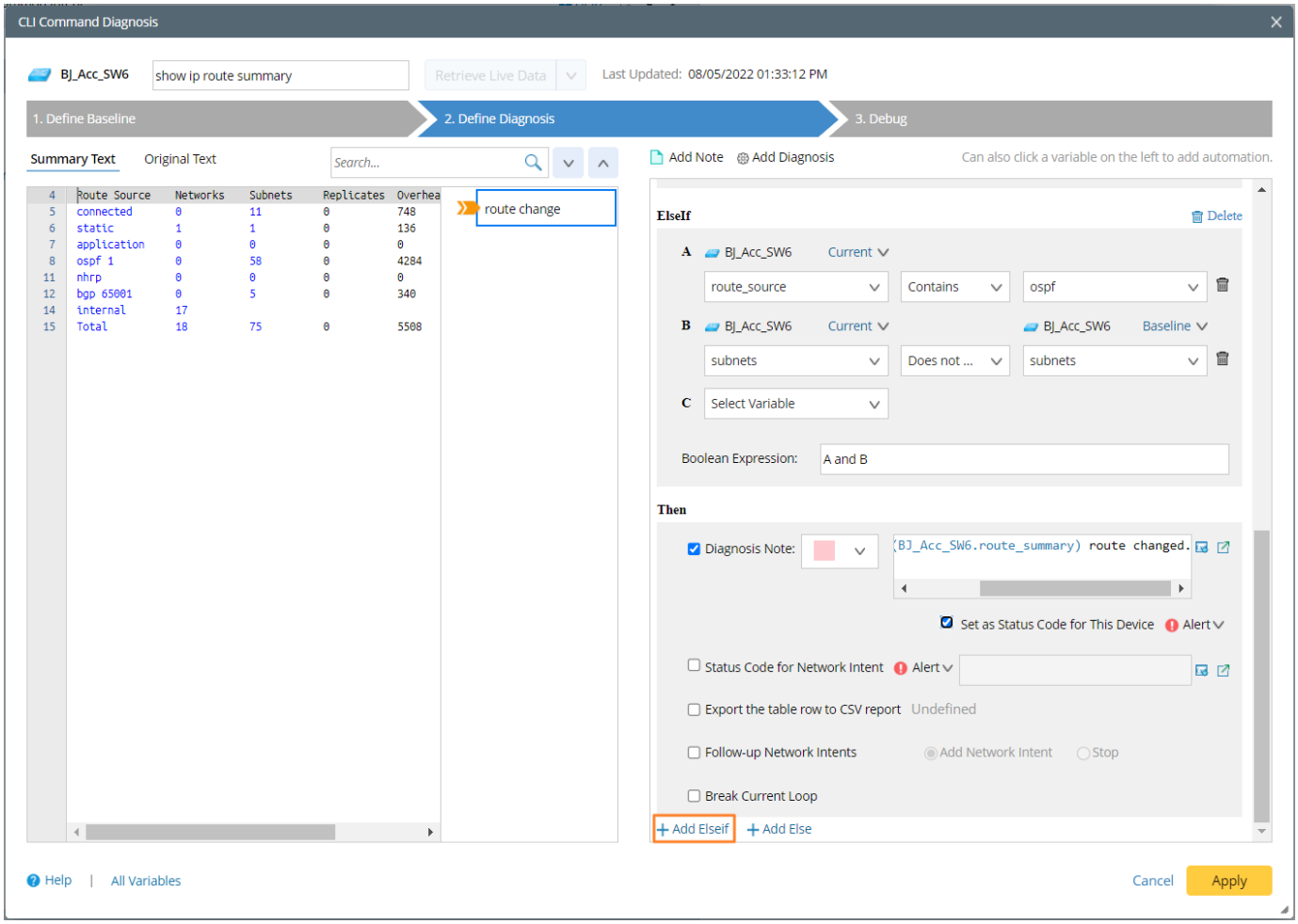


The matched member NIs in the follow-up NIC will be displayed in the Diagnosis Tree (Post-Execution).



2.2.3 Support Elseif branch in diagnosis

R11 adds the **Elseif** branch under the **If** branch of the diagnosis. For example, it can be used to judge whether the subnet of the BGP route in the route summary has changed. If the subnet has not changed, check whether the subnet of the OSPF route has changed.

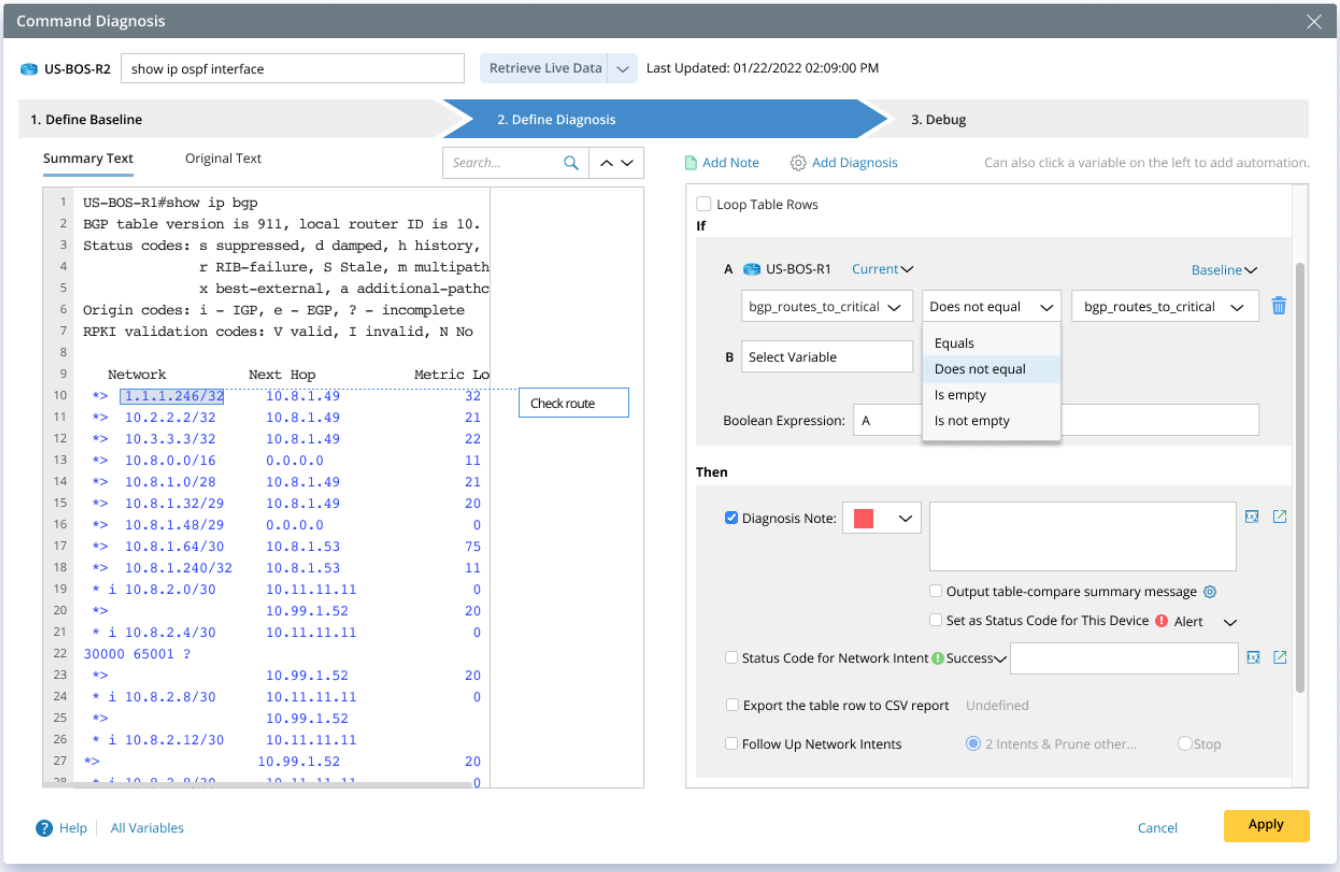


2.2.4 Better support for the table-based diagnosis

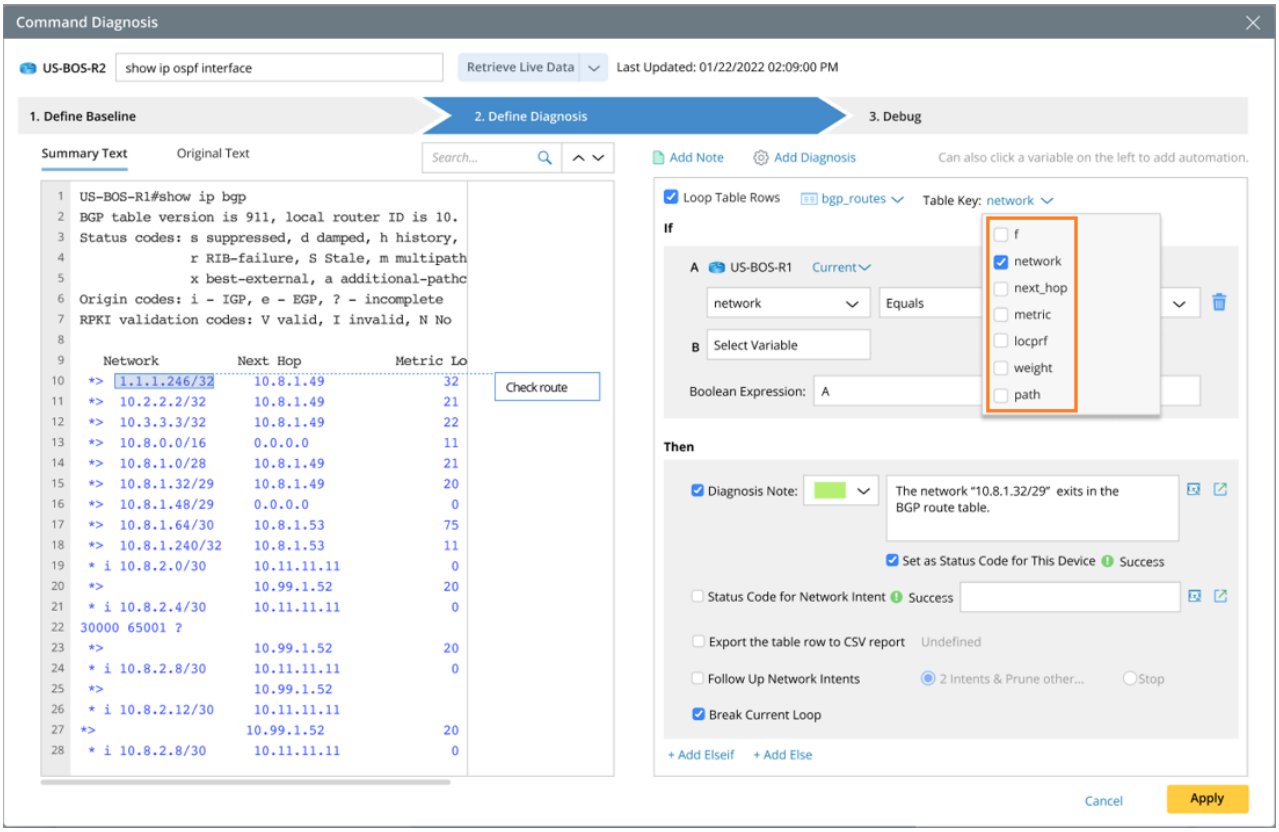
R11 provides better support for the table-based diagnosis, including:

- The **Whole Table-based Diagnosis** to compare the current table with its last or baseline value to detect changes.

When the Loop Table Rows option is not selected, a user can select a table, an operator (Equals, Does not equal, Is empty, and Is not empty), and the table to be compared.

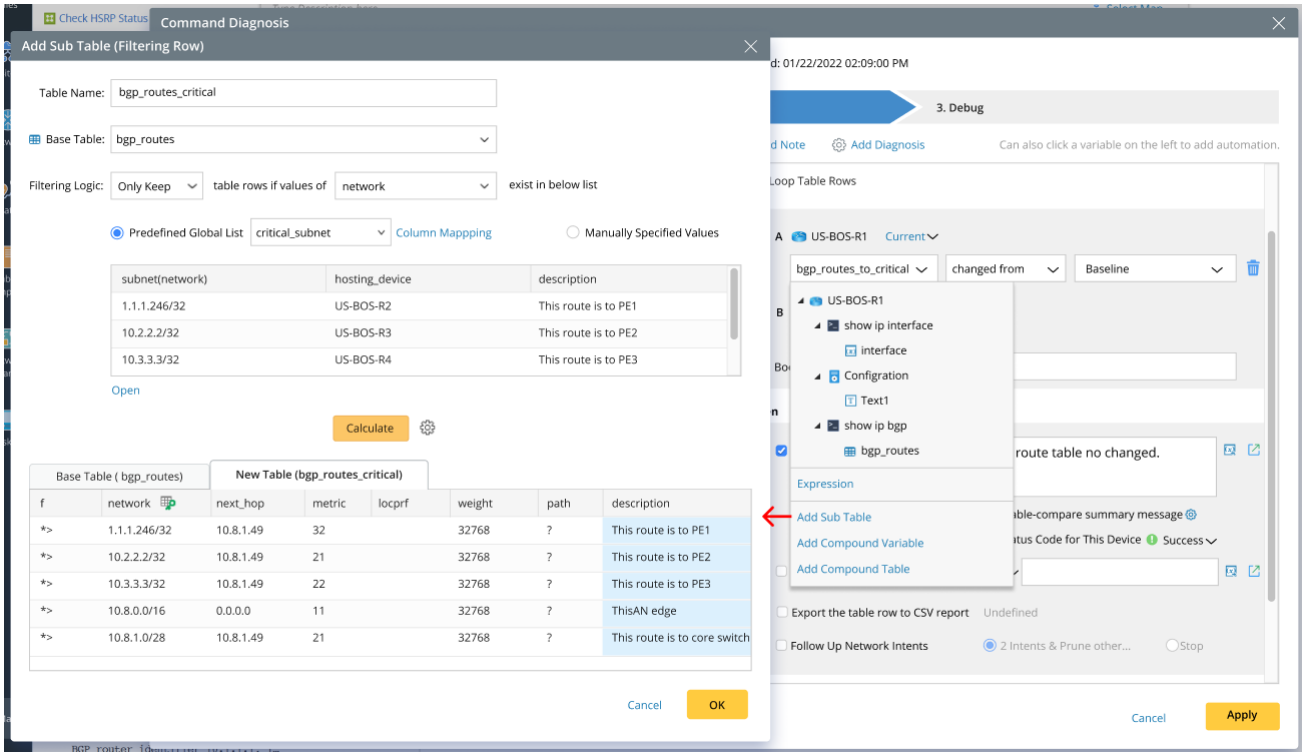


- Set the table key in the diagnosis



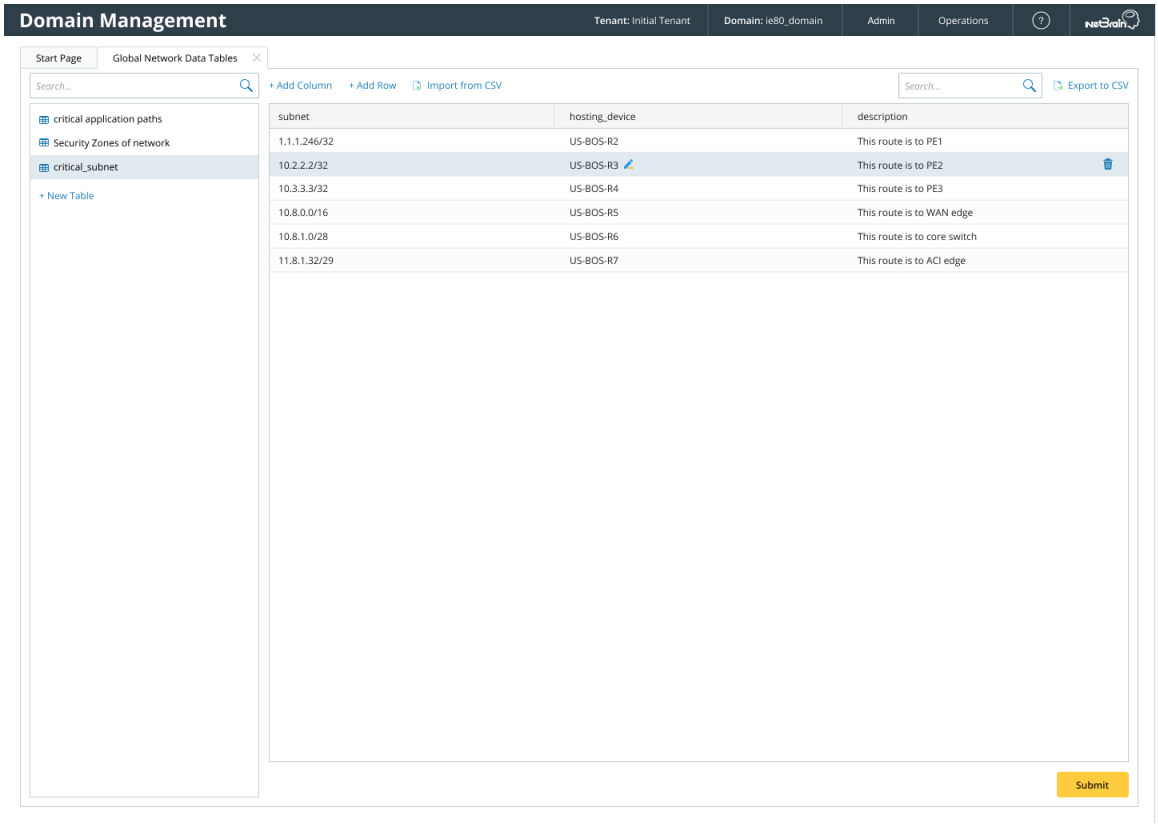
- Support the sub table creation

R11 supports defining a sub table to define a table-based batch comparison diagnosis, which can reduce the complexity of creating a table-based diagnosis. The sub table can be created by either keeping or removing the value at the specific columns, which can be used to compare only the desired data in a table. For example, users only want to analyze the route entries related to the critical application subnet in the device route table and ignore the changes of other dynamic routes, reducing the noise alert that may appear in diagnosis.



- Support the global network data tables

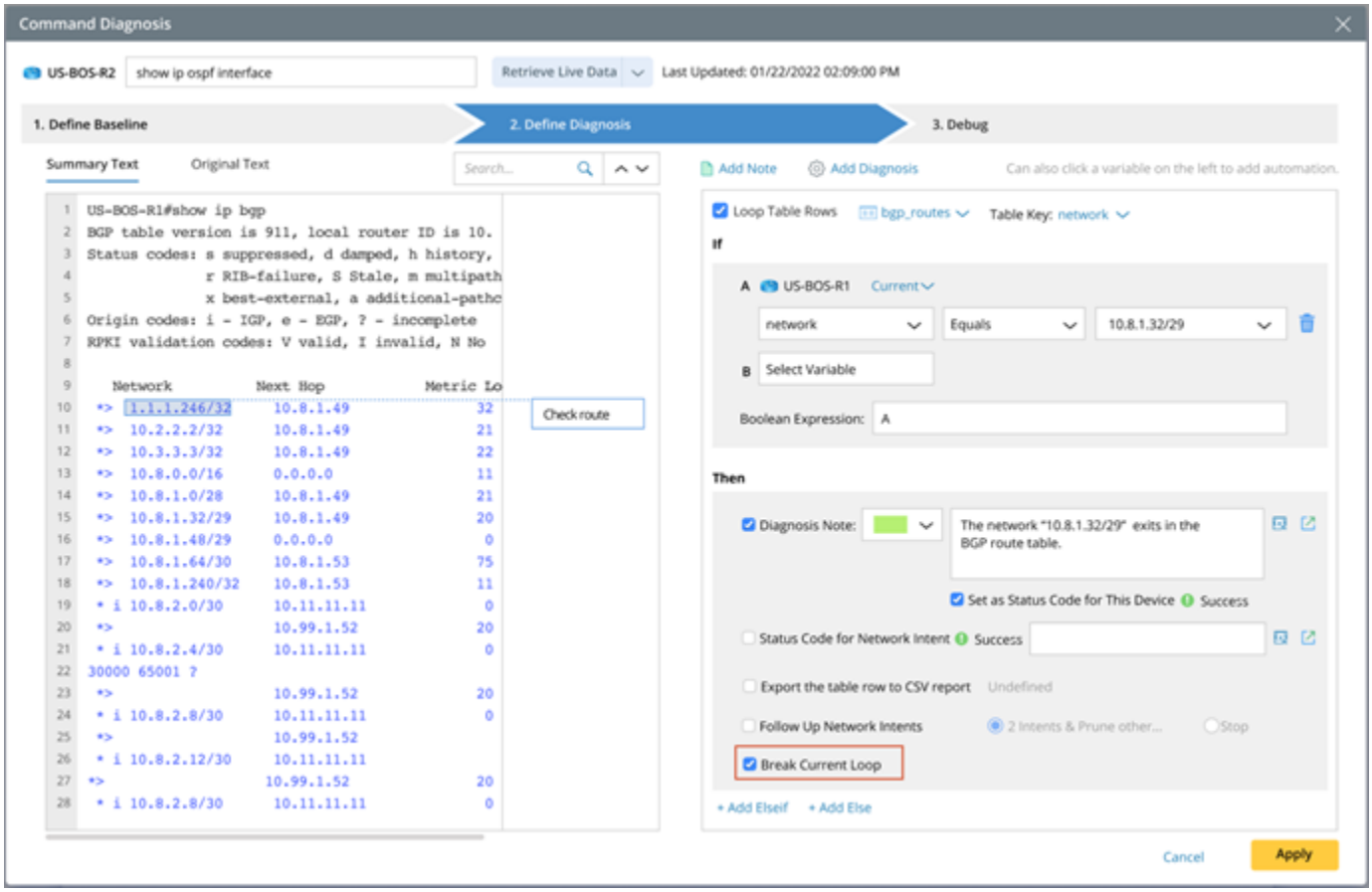
R11 supports predefining **Global Network Data Tables** as the data source for Sub Table to define white or block list. It can also be used for **Mapping Macro Variable** in NIT definition.



- Break table loop

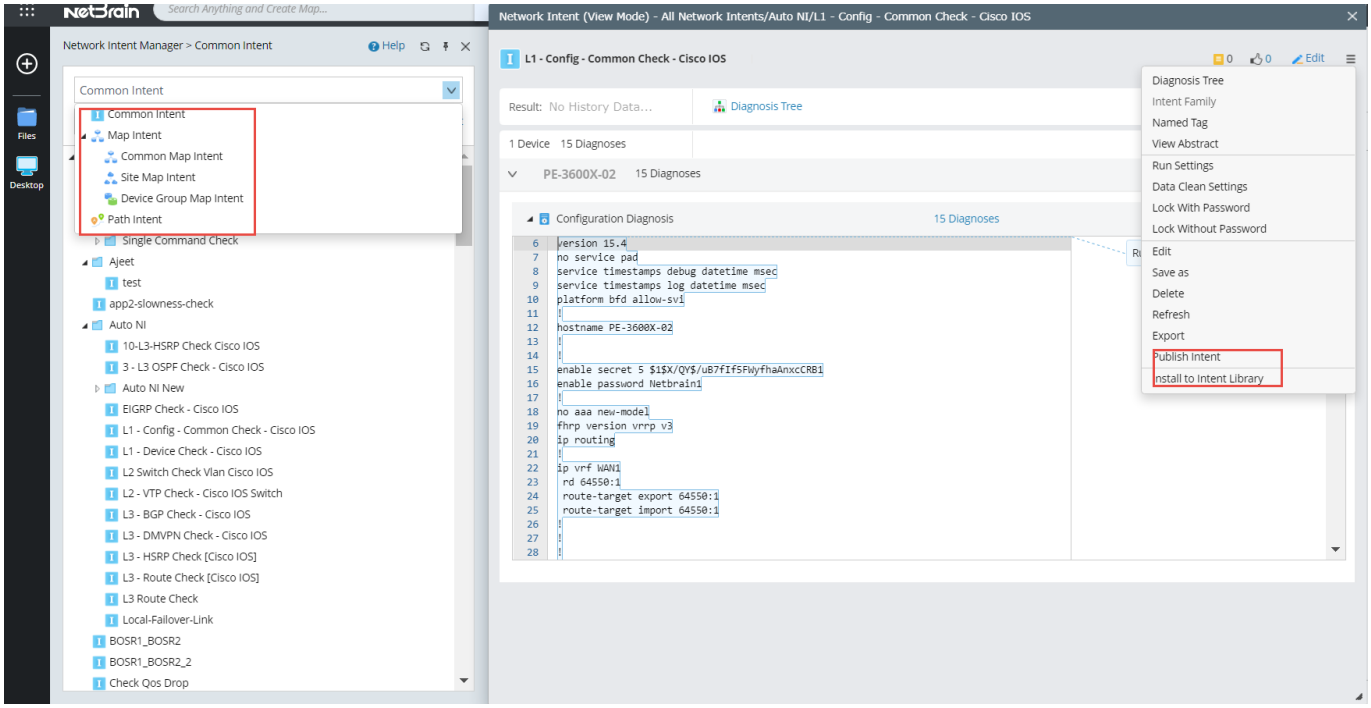
  A new Output (Break Table Loop) can be added to the diagnosis output. Beak Table Loop can better control the flow of the loop table row. When the table row loops, the current loop table will be

terminated to reduce the calculation of the redundant loop if the condition is satisfied.



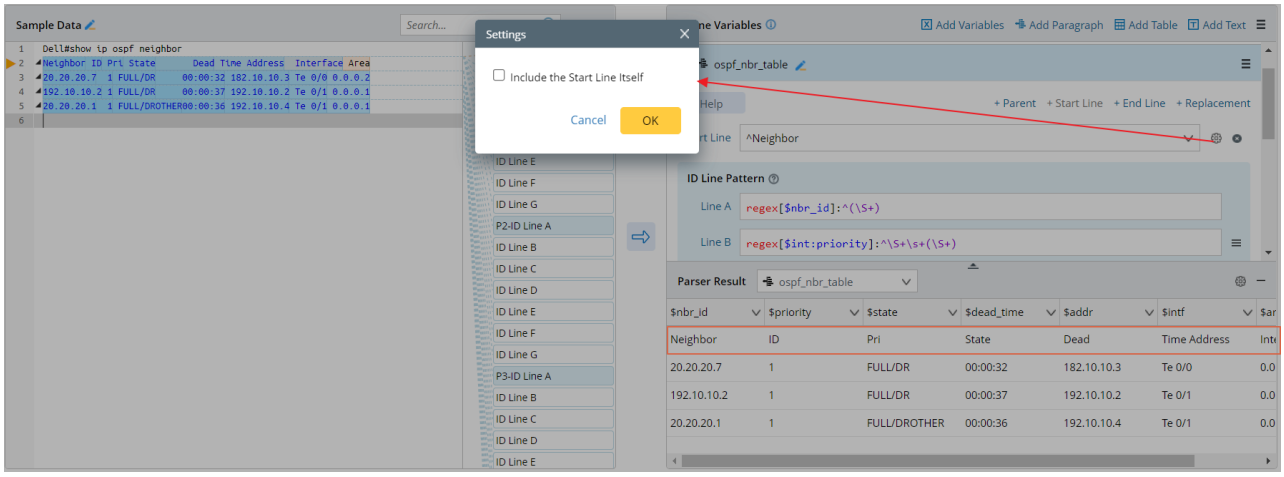
2.3 Intent Manager

R11 provides a standalone Intent Manager to manage all intents, including common intents, Map Intents and Path Intents. Besides editing, creating, and deleting functions, users can publish and install an intent to the library, two new intent functions.

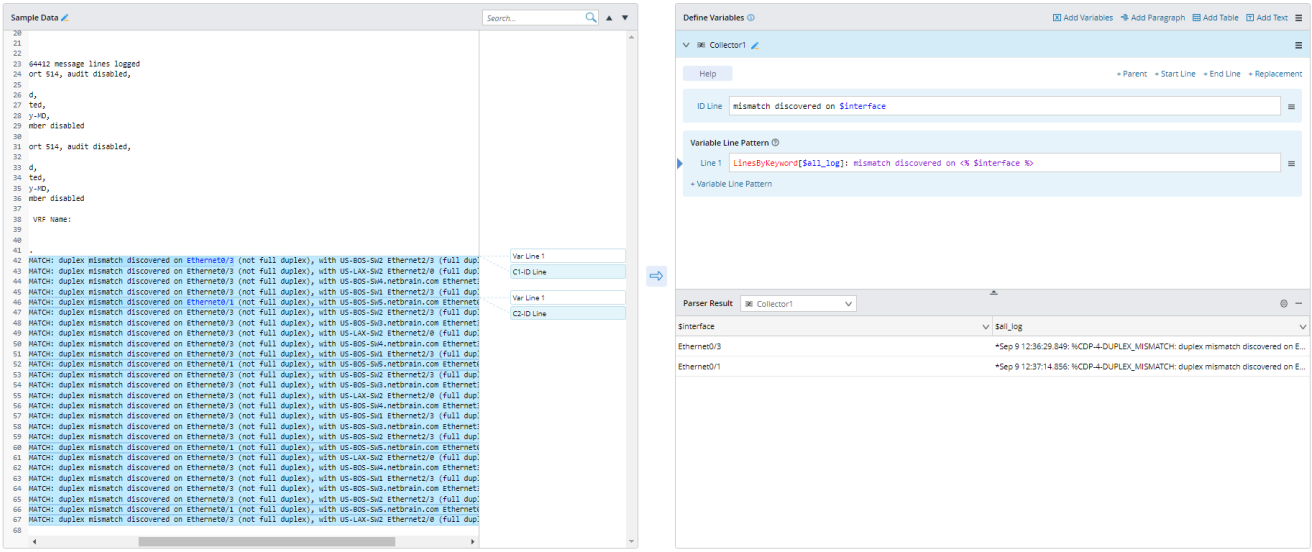


2.4 Visual Parser Enhancements

NetBrain R11 supports the setting **Include the Star Line Itself** for the Start Line and **Include the End Line Itself** for the End Line, determining whether the Start Line and the End Line are included in the matched text.



During the Collector Parser definition, the ***<% $var %>*** is supported in the LinesByKeyword expression syntax in Variable Line. The specific var1 in syntax <% $var1 %> refers to the ID variable defined in the ID Line. With this improvement, users can define multiple lines of configlets simultaneously.



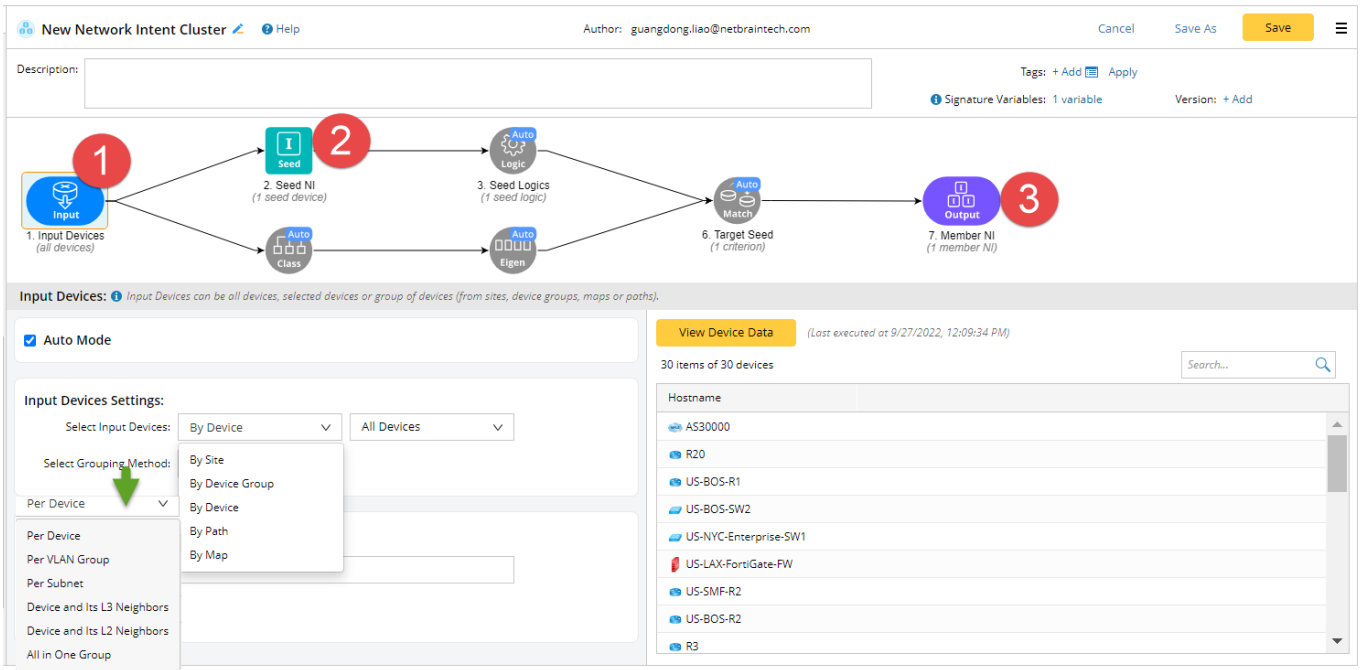
3 Replicate Intent

Replicating the logic of a NI (seed NI) to the whole network is essential to build PDAS on a large scale. R11 introduces a simpler method**, Intent Cluster (NIT)**, to replicate the intent while improving the Intent Cluster (NIC), adding the **Auto Mode** to reduce 7 steps to 3 steps and the **Auto Test** feature to simplify the definition of the device classifier and eigen grouping and create the member NIs only for the qualified devices.

3.1 NIC Improvements

3.1.1 Auto Mode

The Auto Mode (enabled by default) reduces NIC definition from 7 steps to three simple steps:



1. Select the input devices.

   R11 adds two ways to select the input devices: **by path** and **by map**. When users select inputting the device **by device**, they must select the method to create the group, which can be **per device, per VLAN group, per subnet, device and L3 neighbors, device and its L2 neighbors, and all in one group**.
2. Select Seed NIs.

   The auto mode only supports the single device diagnosis. The system will ask users to disable the auto mode if the Seed Intent contains a cross-device diagnosis.
3. Create the member NIs.

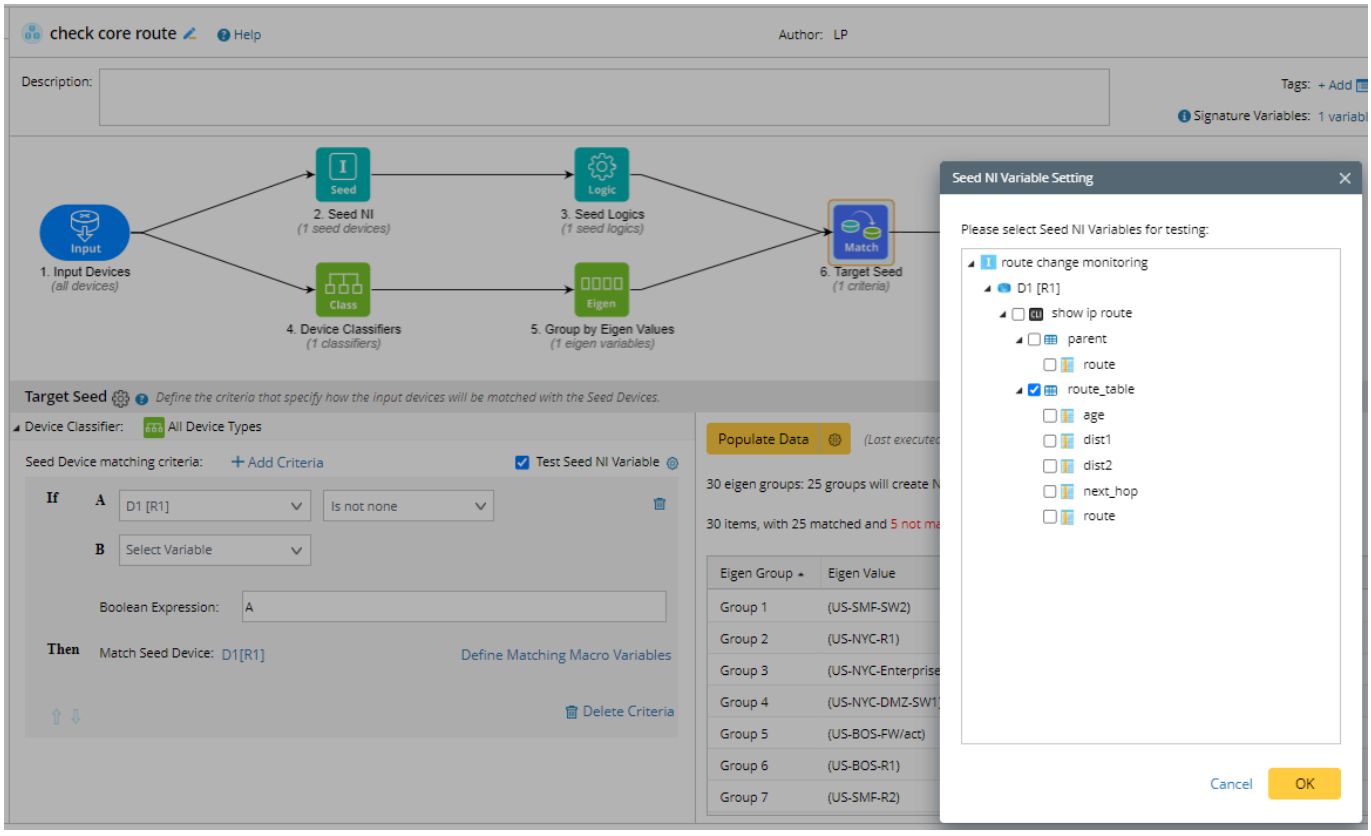
   The member NIs will be created by the type of the input devices or the method to create the group:
   - **By map**: all devices on the same map will belong to a member NI.
   - **By Site**: all devices of a site will belong to a member NI.
   - **By Device Group**: all devices in a device group will belong to a member NI.
   - **By Path**: all devices in a path will belong to a member NI.
   - **By Device**:
     - **Per device**: a member NI will be created for each device.
     - **Per VLAN group**: a member NI will be created for all devices belonging to a VLAN group.
     - **Per subnet**: a member NI will be created for all devices belonging to a subnet.
     - **Device and L3 neighbors**: a member NI will be created for the device and its L3 neighbors.
     - **Device and its L2 neighbors**: a member NI will be created for the device and its L2 neighbors.
     - **All in one group**: only one member NI is created to include all devices.

The system automatically creates other nodes. Users can disable the auto nodes and edit these nodes.

NIC auto mode can be used to batch create the map intents (input the devices by map so that the map name will be used as eigenvalue) and Path intents (input the devices by path so that the path name will be used as eigenvalue).

3.1.2 Auto Test

R11 adds an option, **Test Seed NI variable**, to node **Target Seed**. With this option enabled, users can select the **seed NI variables**. If one of these variables is not retrieved or parsed successfully from a device, the system will not create member NI for this device. With this option, users can create the member NIs that do not create meaningful results and simplify the definition of Device Classifiers (node 4) and Group by Eigen Values (node 5) when multiple vendors or commands are involved.

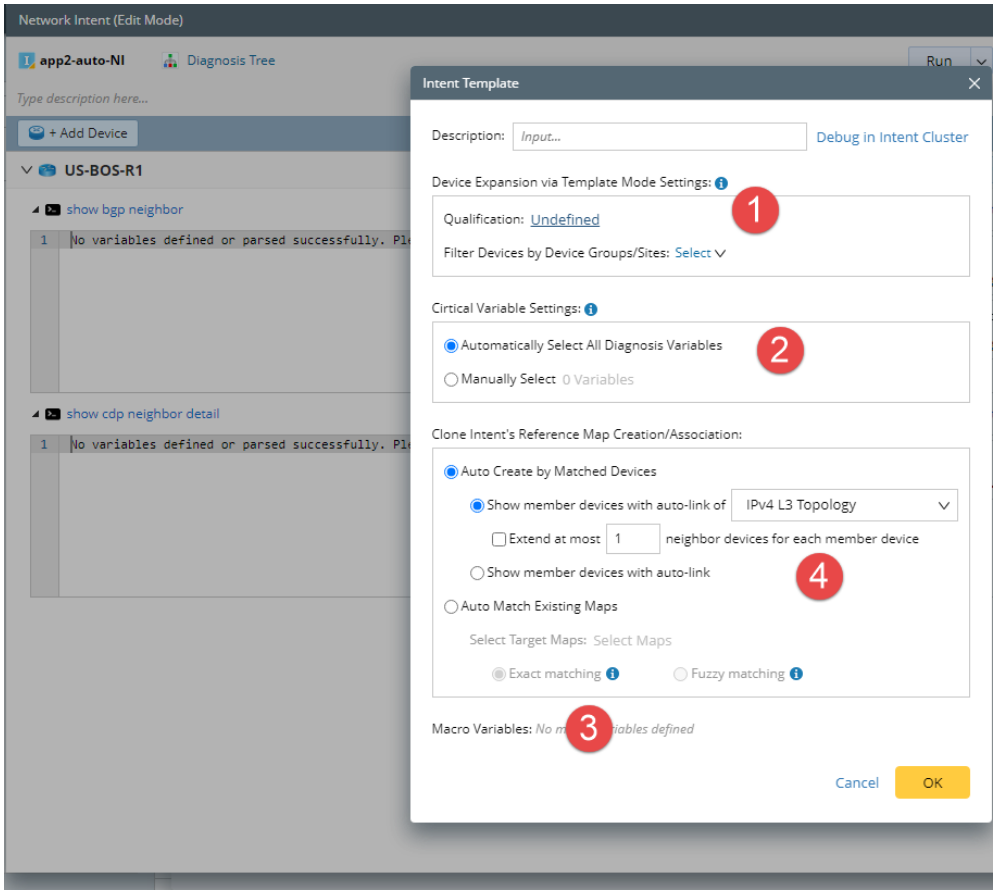


3.2 Intent Template (NIT)

R11 further simplifies the replication of the intent by inventing the **Intent Template (NIT)**. A user can define NIT for any NI with the single-device diagnosis. Like NIC auto mode, if an Intent contains cross-device diagnosis, it cannot be replicated by NIT.

3.2.1 Define NIT

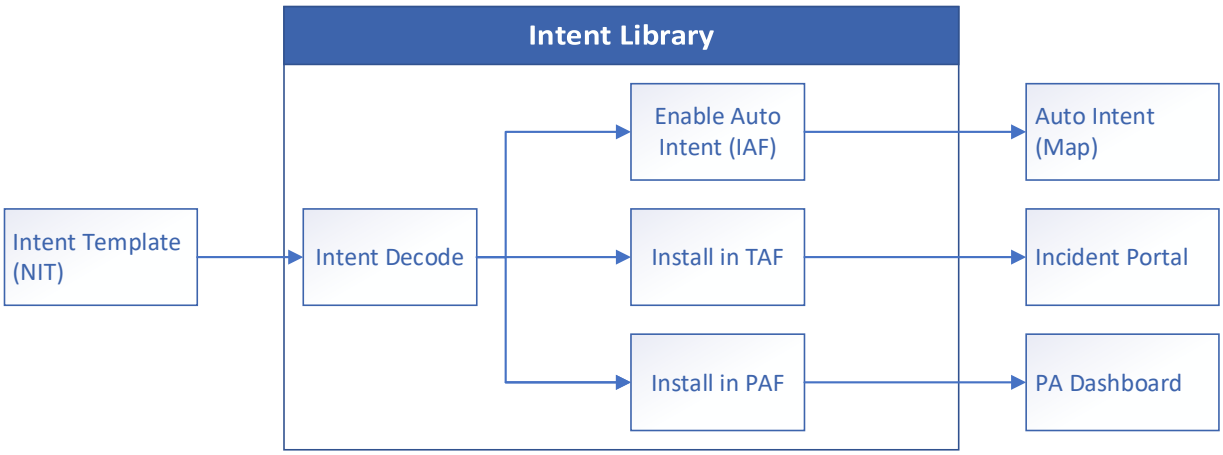
Users can define an NIT in four easy steps:



1. Define the qualified devices: Cisco IOS devices with OSPF configuration. Further, you can filter devices by device group or sites.
2. Define critical variable settings: The system selects all diagnosis variables by default. You can manually set a subset of the variables. A member NI will only be created for a device if all critical variables are retrieved and parsed successfully.
3. If NI has macro variables, you may define macro variables.
4. Define how to create the reference map.

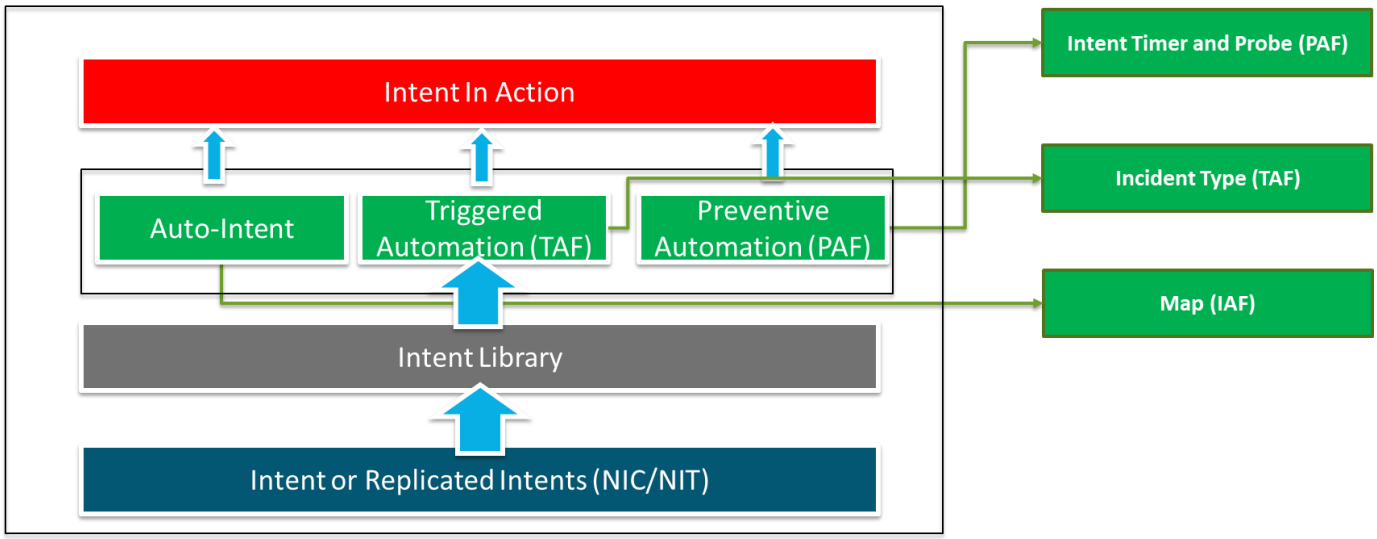
3.2.2 Install and Use NIT

After NIT is defined, it can be installed and used throughout the system. First, you must instruct the system to decode the intent, a process to decide whether to create the member NI based on the template definition. Then, you can enable the intent to be displayed in the map under the auto intent; install it in TAF so that it can be triggered; install it in PAF so that the Probe can trigger it. R11 add Intent Library as a central console for these operations.



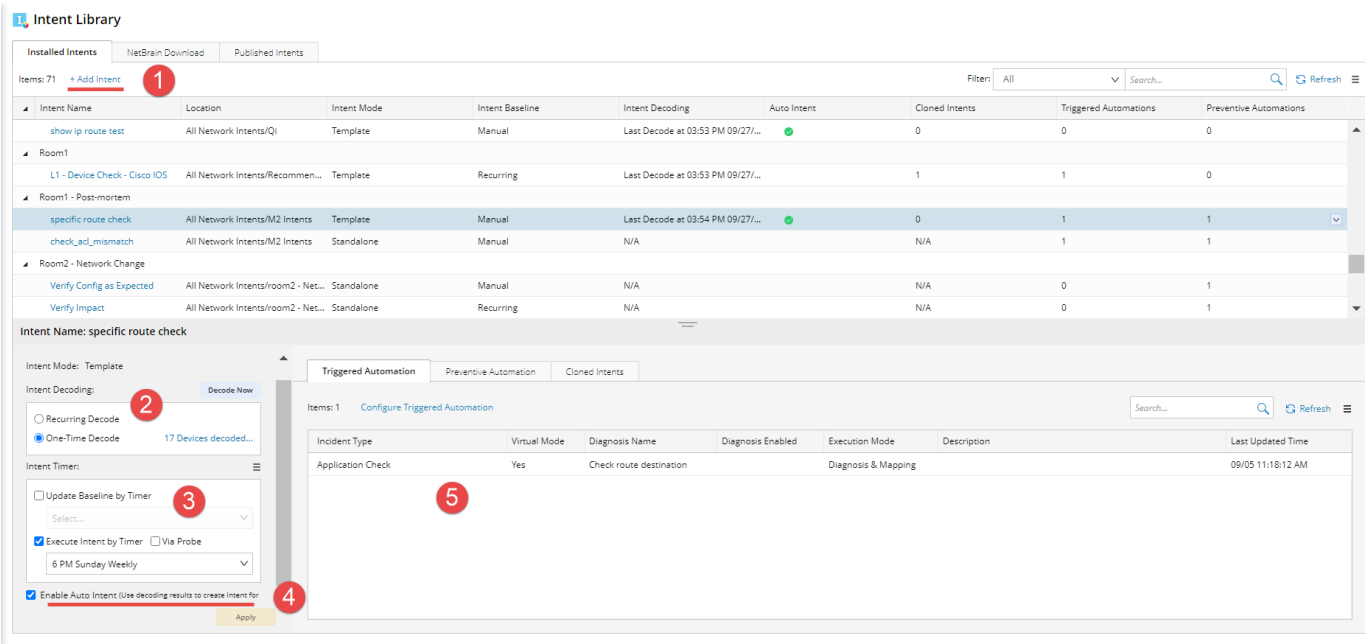
4 Install Intent

Intent or replicated Intent (NIT or NIC) can be executed manually from the map or path (Interactive Automation, IAF), triggered by a 3rd party system (Triggered Automation, TAF), or triggered by the Probe (Preventive Automation, PAF).

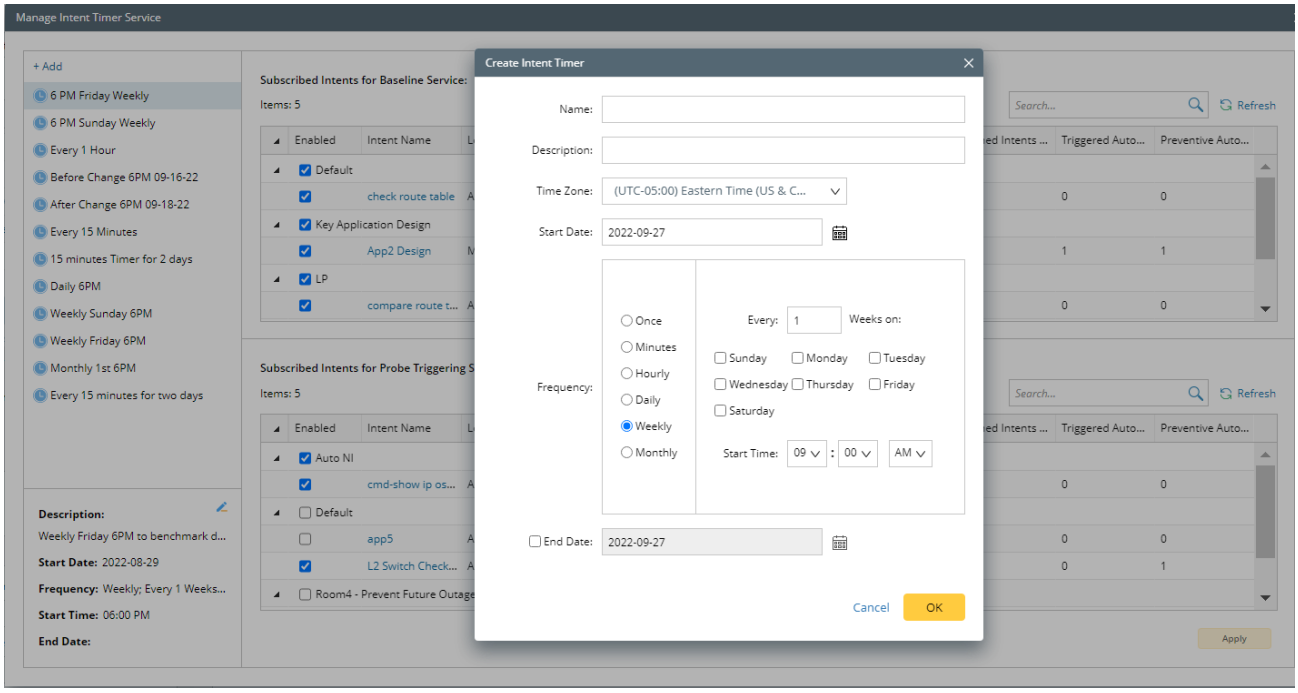


4.1 Intent Library

The intent library is the central console to install an intent:



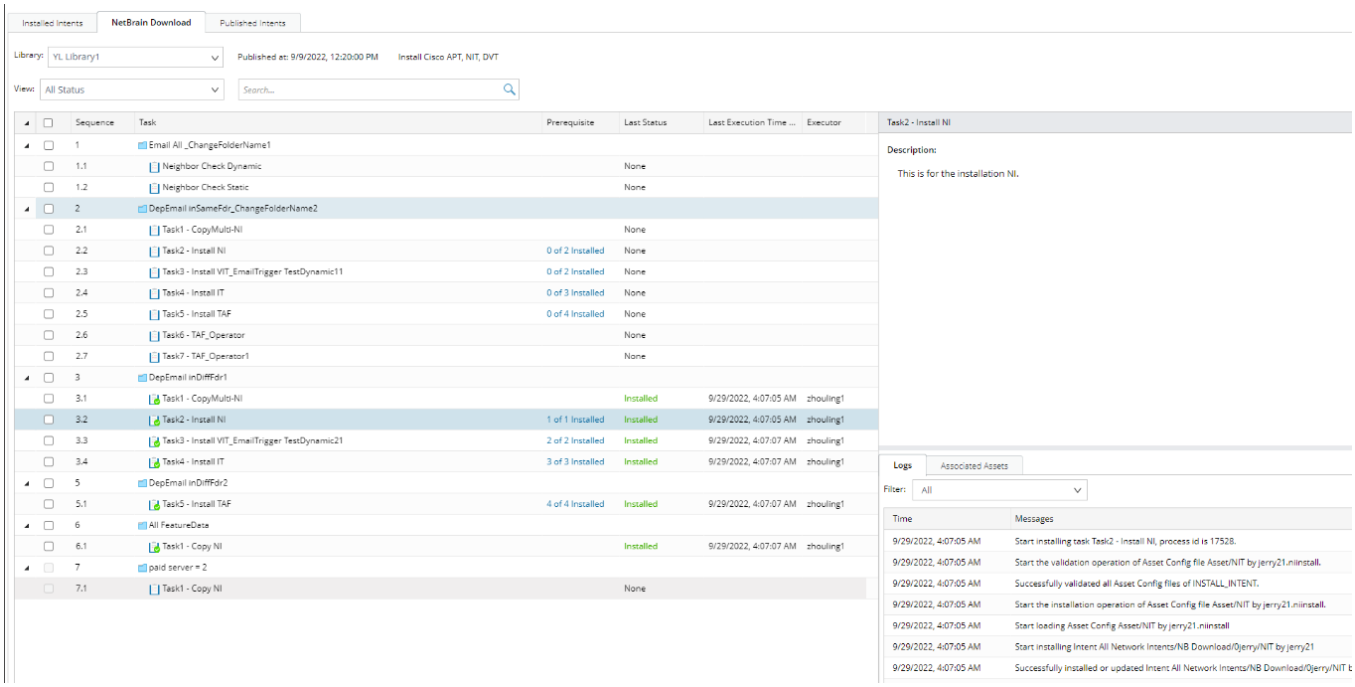
1. Add an intent as the standalone or the template.
2. Configure the intent decoding if the intent is installed as a template. Users can select one-time or recurring decoding. The NIT decoding service creates a list of qualified devices based on the NIT setting. These qualified devices will be used later to clone NIs on the fly.
3. Configure Intent Timers. Configure the timer to update the intent baseline and execute the intent. The execution timer is used by PAF, which has two options, **Execute by Timer** (always execute when the timer is up) and **via Probe** (only when the condition of Probe is satisfied).



4. Enable auto intent so the intent decode service can clone intents for the map.
5. Install NI/NIT to PAF and TAF.

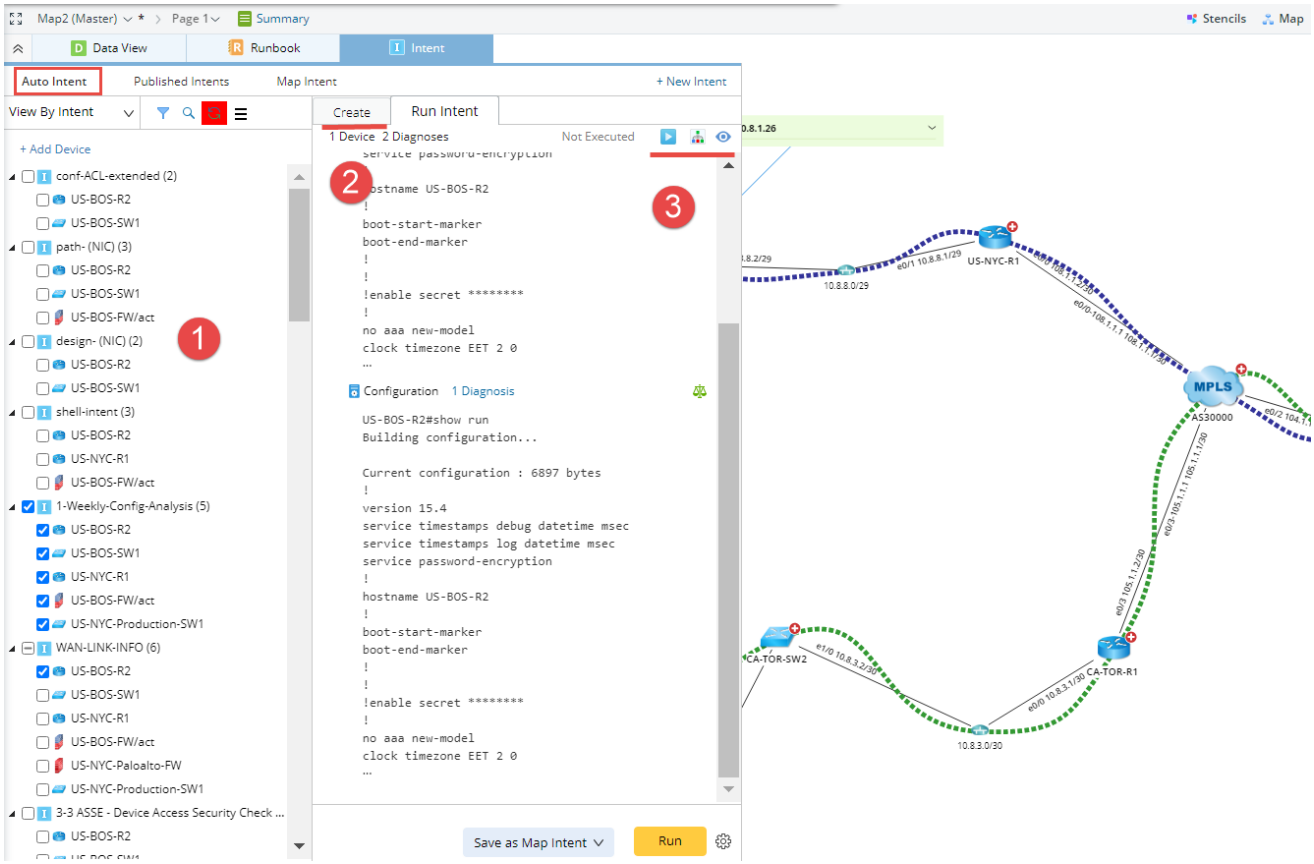
Besides the installed intent, NetBrain downloaded tasks and published intents are also displayed in the Intent Library.

NetBrain team continuously creates the intents for common network problems and publishes them through Knowledge Cloud, which can be automatically downloaded to your system. The downloaded files include a configuration Orchestration File for each task, which can be executed to download and install the related intents.


Installed Intents
NetBrain Download
Published Intents
Library: YL Library1
Published at: 9/9/2022, 12:20:00 PM
Install Cisco APT, NIT, DVT
View: All Status
Search...
Sequence
Task
Prerequisite
Last Status
Last Execution Time ...
Executor
1 Email All _ChangeFolderName1
1.1 Neighbor Check Dynamic None
1.2 Neighbor Check Static None
2 DepEmail inSameFdr_ChangeFolderName2
2.1 Task1 - CopyMulti-NI None
2.2 Task2 - Install NI 0 of 2 Installed None
2.3 Task3 - Install VIT_EmailTrigger TestDynamic11 0 of 2 Installed None
2.4 Task4 - Install IT 0 of 3 Installed None
2.5 Task5 - Install TAF 0 of 4 Installed None
2.6 Task6 - TAF_Operator None
2.7 Task7 - TAF_Operator1 None
3 DepEmail inDiffFdr1
3.1 Task1 - CopyMulti-NI Installed 9/29/2022, 4:07:05 AM zhouling1
3.2 Task2 - Install NI 1 of 1 Installed Installed 9/29/2022, 4:07:05 AM zhouling1
3.3 Task3 - Install VIT_EmailTrigger TestDynamic21 2 of 2 Installed Installed 9/29/2022, 4:07:07 AM zhouling1
3.4 Task4 - Install IT 3 of 3 Installed Installed 9/29/2022, 4:07:07 AM zhouling1
5 DepEmail inDiffFdr2
5.1 Task5 - Install TAF 4 of 4 Installed Installed 9/29/2022, 4:07:07 AM zhouling1
6 All FeatureData
6.1 Task1 - Copy NI Installed 9/29/2022, 4:07:07 AM zhouling1
7 paid server = 2
7.1 Task1 - Copy NI None
Task2 - Install NI
Description:
This is for the installation NI.
Logs
Associated Assets
Filter: All
Time
Messages
9/29/2022, 4:07:05 AM Start installing task Task2 - Install NI, process id is 17528.
9/29/2022, 4:07:05 AM Start the validation operation of Asset Config file Asset/NIT by jerry21.niinstall.
9/29/2022, 4:07:05 AM Successfully validated all Asset Config files of INSTALL_INTENT.
9/29/2022, 4:07:05 AM Start the installation operation of Asset Config file Asset/NIT by jerry21.niinstall.
9/29/2022, 4:07:05 AM Start loading Asset Config Asset/NIT by jerry21.niinstall
9/29/2022, 4:07:05 AM Start installing Intent All Network Intents/NB Download/0jerry/NIT by jerry21
9/29/2022, 4:07:05 AM Successfully installed or updated Intent All Network Intents/NB Download/0jerry/NIT b

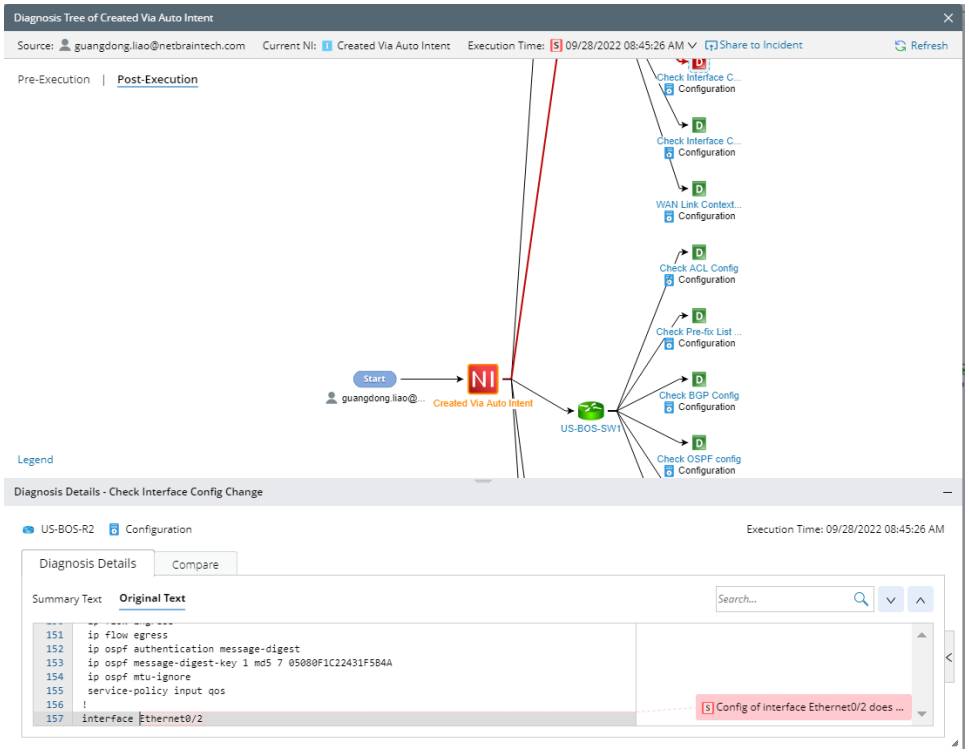
4.2 Auto Intent

When users open a map, the automatically decoded Intents for the map devices are listed under the Auto Intents by device or intent. Users can select one or multiple auto intents:



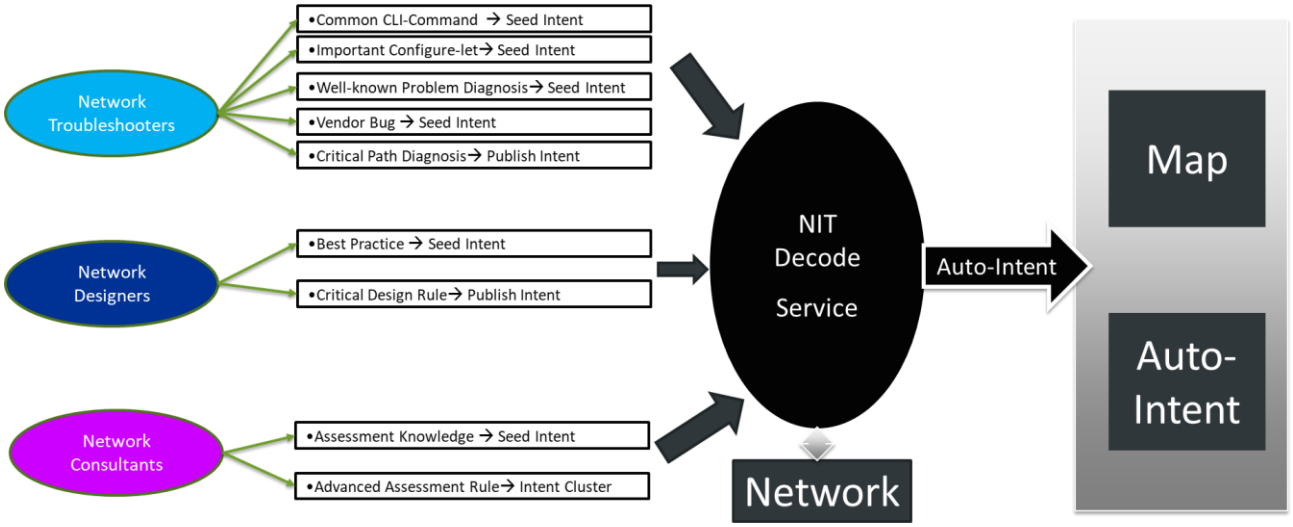
1. Select and view the auto intents.
2. Create or recreate the selected intents. The intent decoding services (configured in the **Intent Library**) do not create the intents. Instead, the auto intents are created here.

3. Run the selected auto intents. The results can be viewed in the **post-execution Diagnosis Tree**.



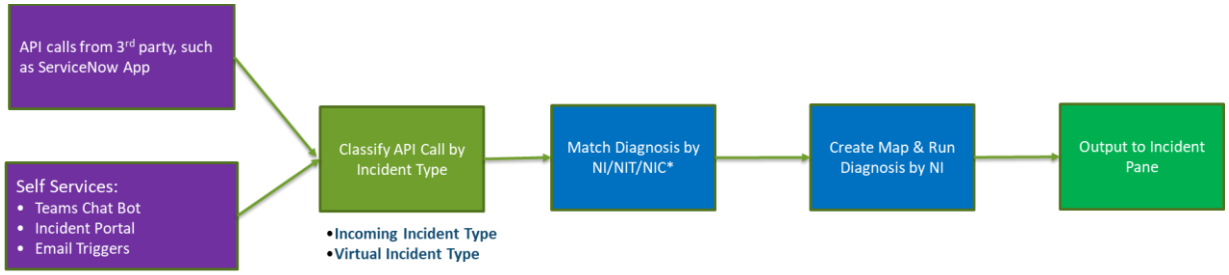
The results can also be viewed on the map by a special data view.

NIT (easy to clone NI), Auto Intent decoding service, and Auto Intent can scale intents to a large network without the users' intervention.



4.3 Triggered Automation (TAF)

The triggered Automation (TAF) is a framework to connect the Automation (NI/NIT/NIC) with a network problem by the following steps:

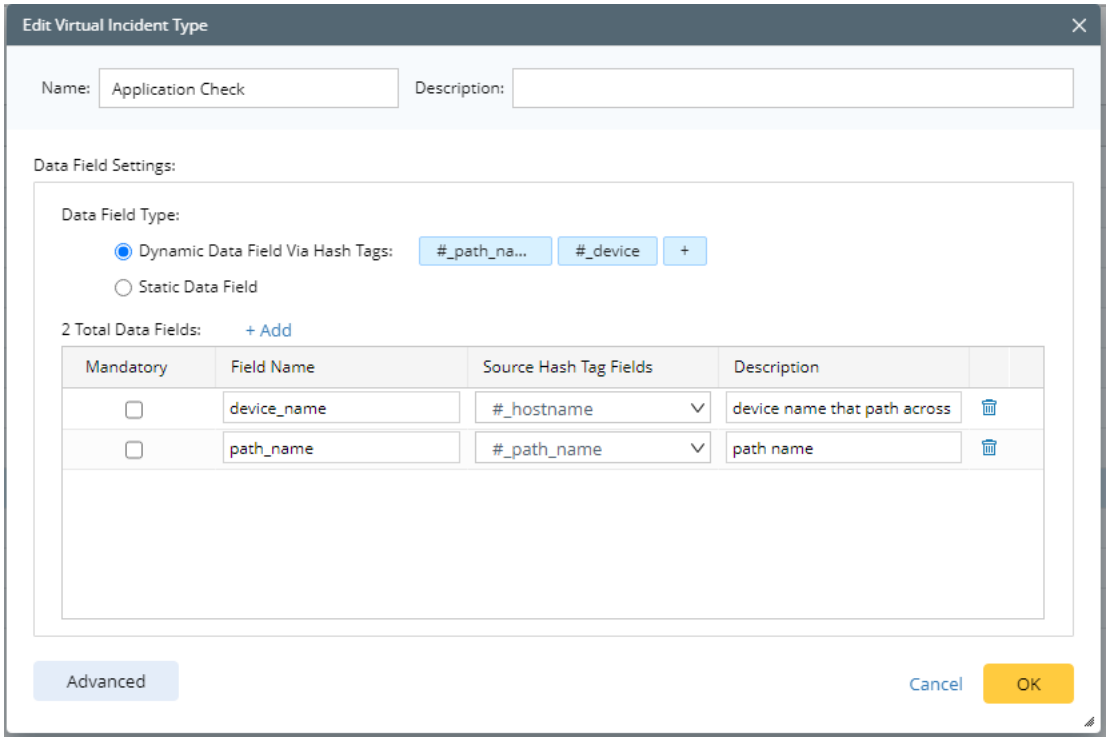


- Receive the API calls from 3rd party system or the self-service application (Team chatbot, Emails, and Incident Portals)
- Classify the call to an Incident Type.

- Match the standalone NI or member NIs with the static data field of the API call (static diagnosis) or the hash tags (dynamic diagnosis).
- Execute NIs to create the map and/or run the diagnosis.
- Output the maps and diagnosis results to the Incident Portals.

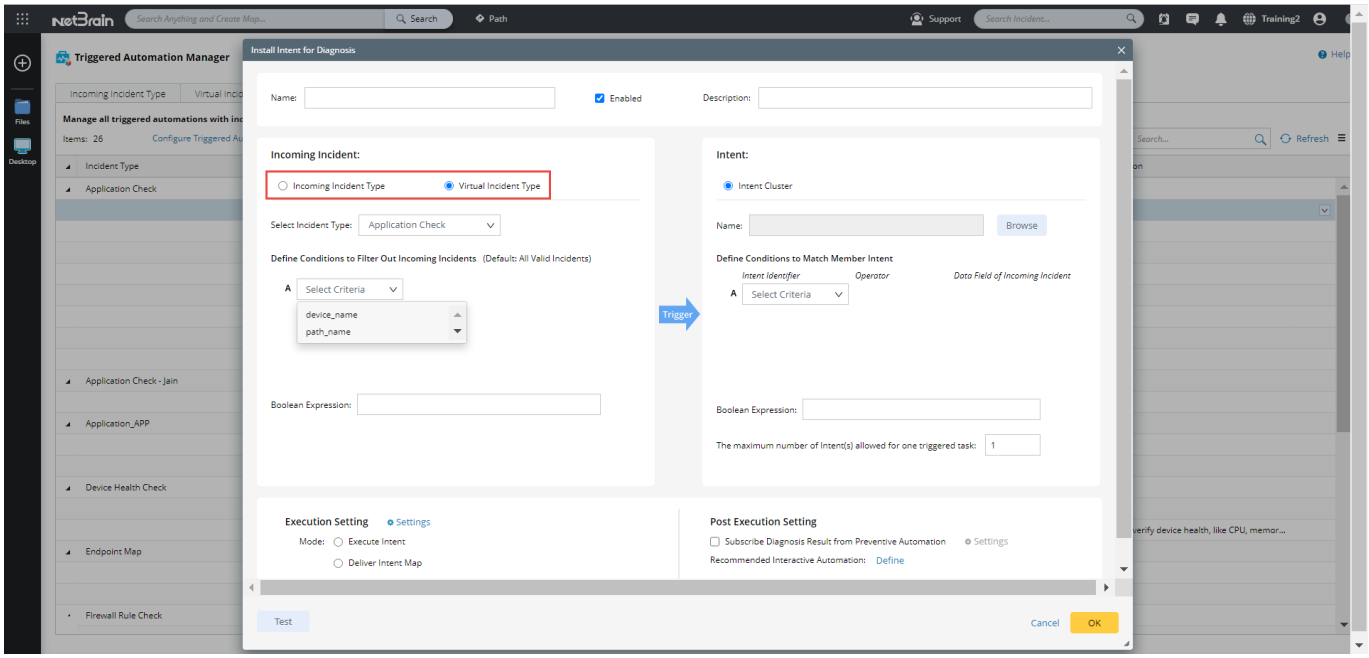
R11 introduces a new Incident Type, Virtual Incident Type (VIT) and renames the previous Incident Type as the Incoming Incident Type. VIT encapsulates the detailed logic of Hash Tag and removes the hurdle of the diagnosis by the Hash Tag (the Dynamic Diagnosis in the previous version).

VIT has a set of data fields to describe a network incident which can be either dynamically filled via Hash Tag or statically defined.

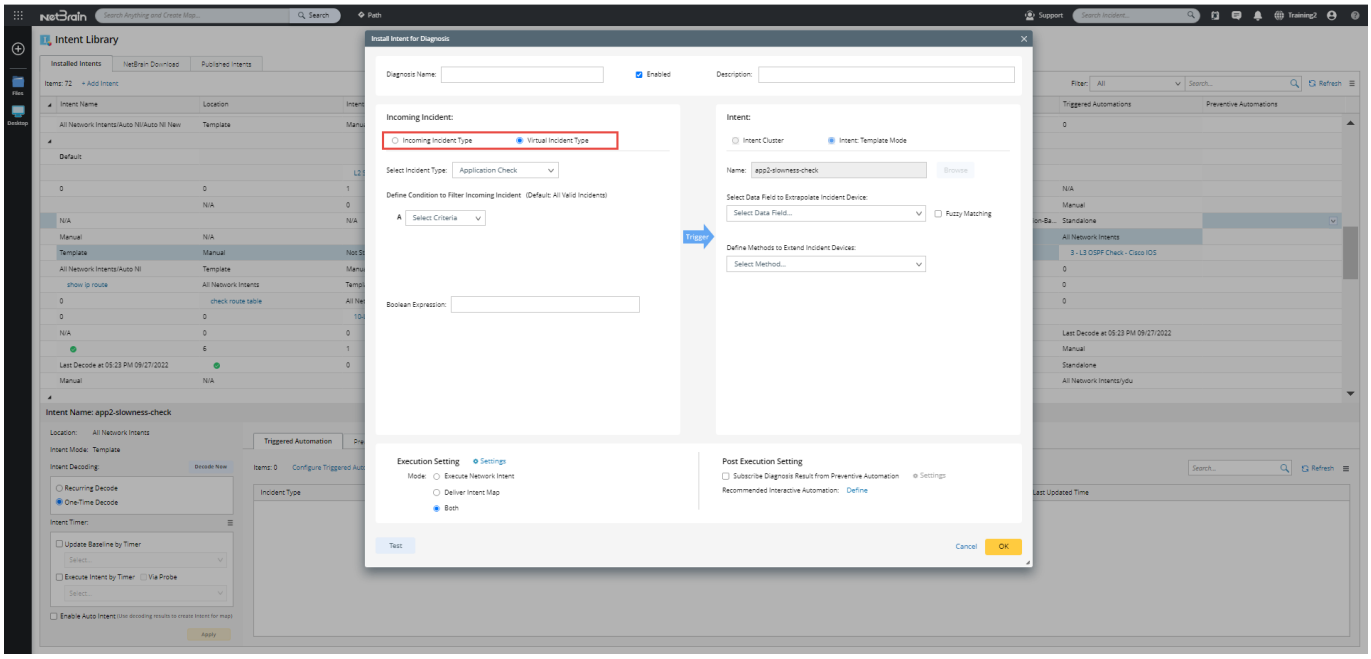


The NI/NIT/NIC can be installed in both incident types with the same interface (in the current version, you can only install the **intent cluster** from the **Triggered Automation Manager** and install NI/NIC in the **Intent Library**):

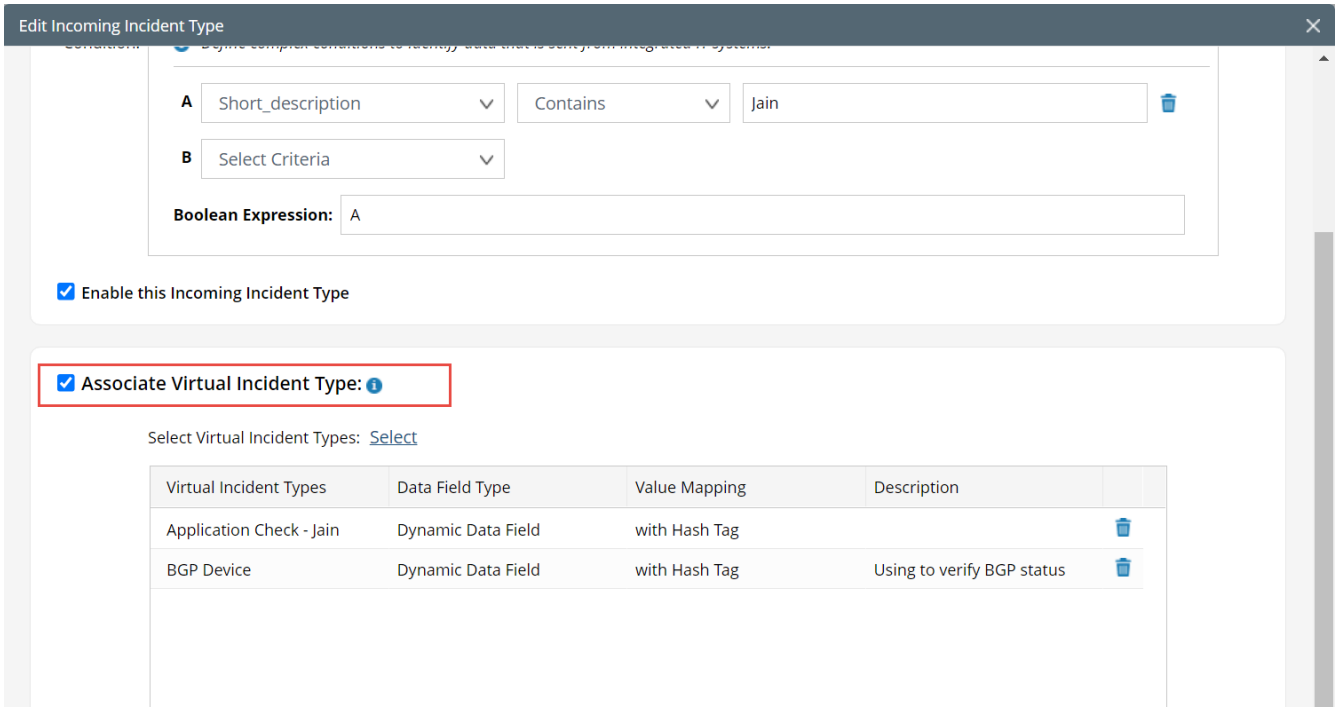
- Install NIC from the **Trigger Automation Manager**:



- Install NI/NIT from the **Intent Library**:



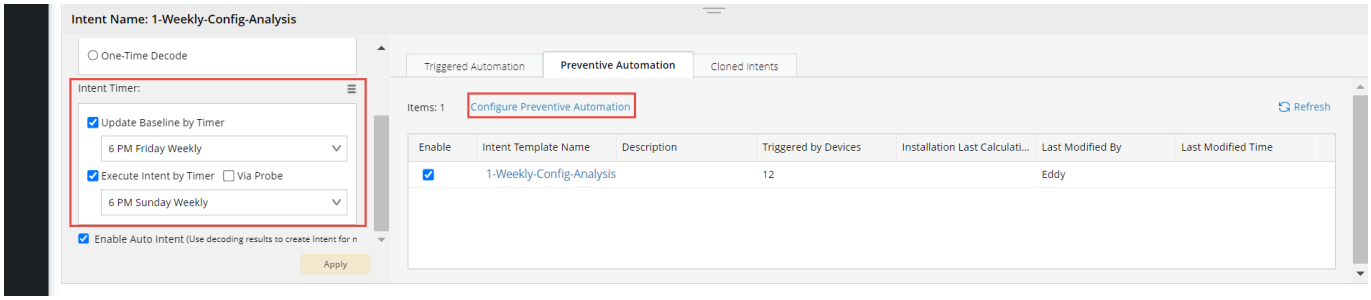
An incoming Incident Type can be associated with one or multiple VITs and inherits the automations installed for these VITs. This one-to-many relationship can help scale TAF.



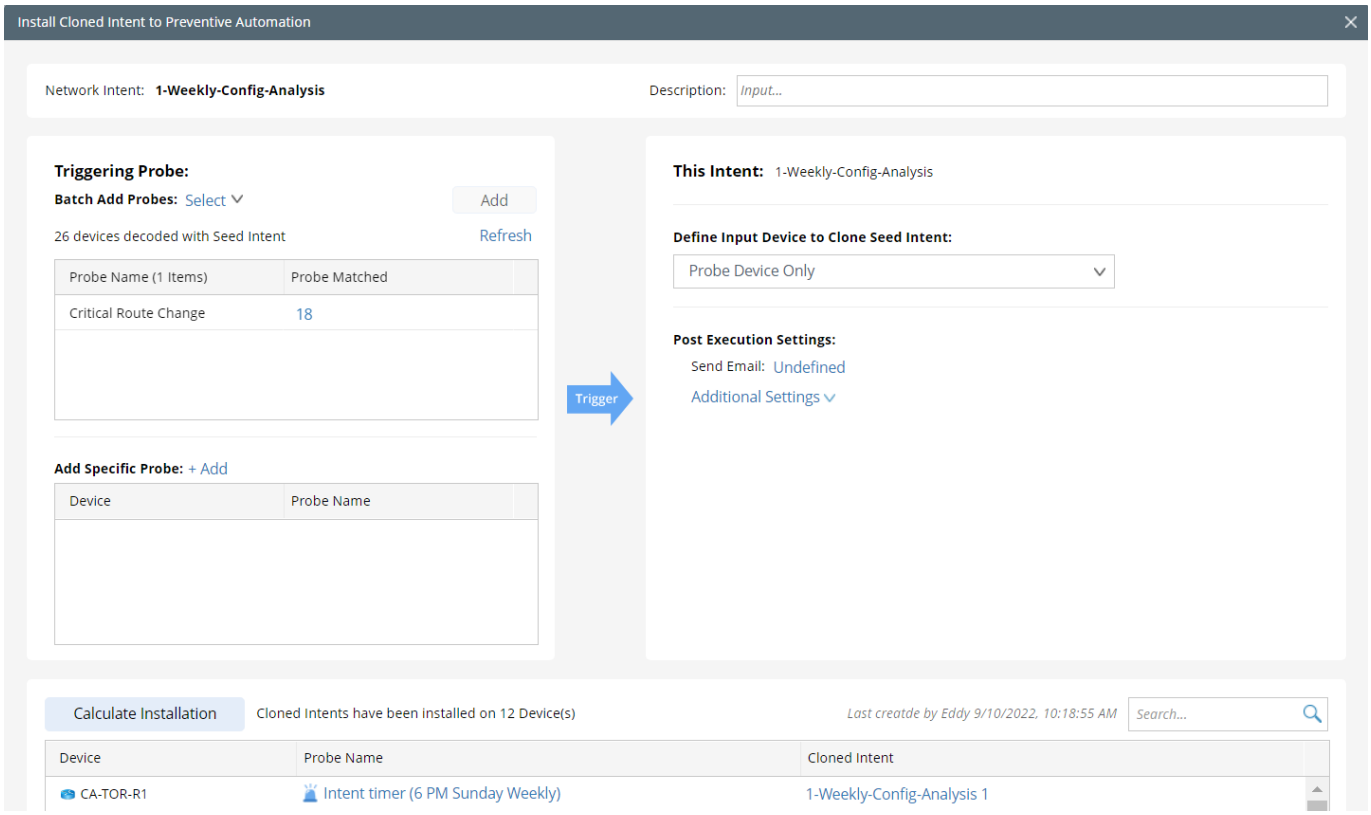
4.4 Preventive Automation (PAF)

PAF uses the two-level probes (primary and secondary probes), which can further trigger any NI/NIT/NIC execution. The results are displayed in PA Dashboard.

Users can configure the timer to execute the intent for PAF, which has two options, **Execute by Timer** (always execute when the timer is up) and **via Probe** (only when the condition of Probe is satisfied).



Users can add probes to trigger this intent in the Intent Library and define the input devices to clone the seed intent.

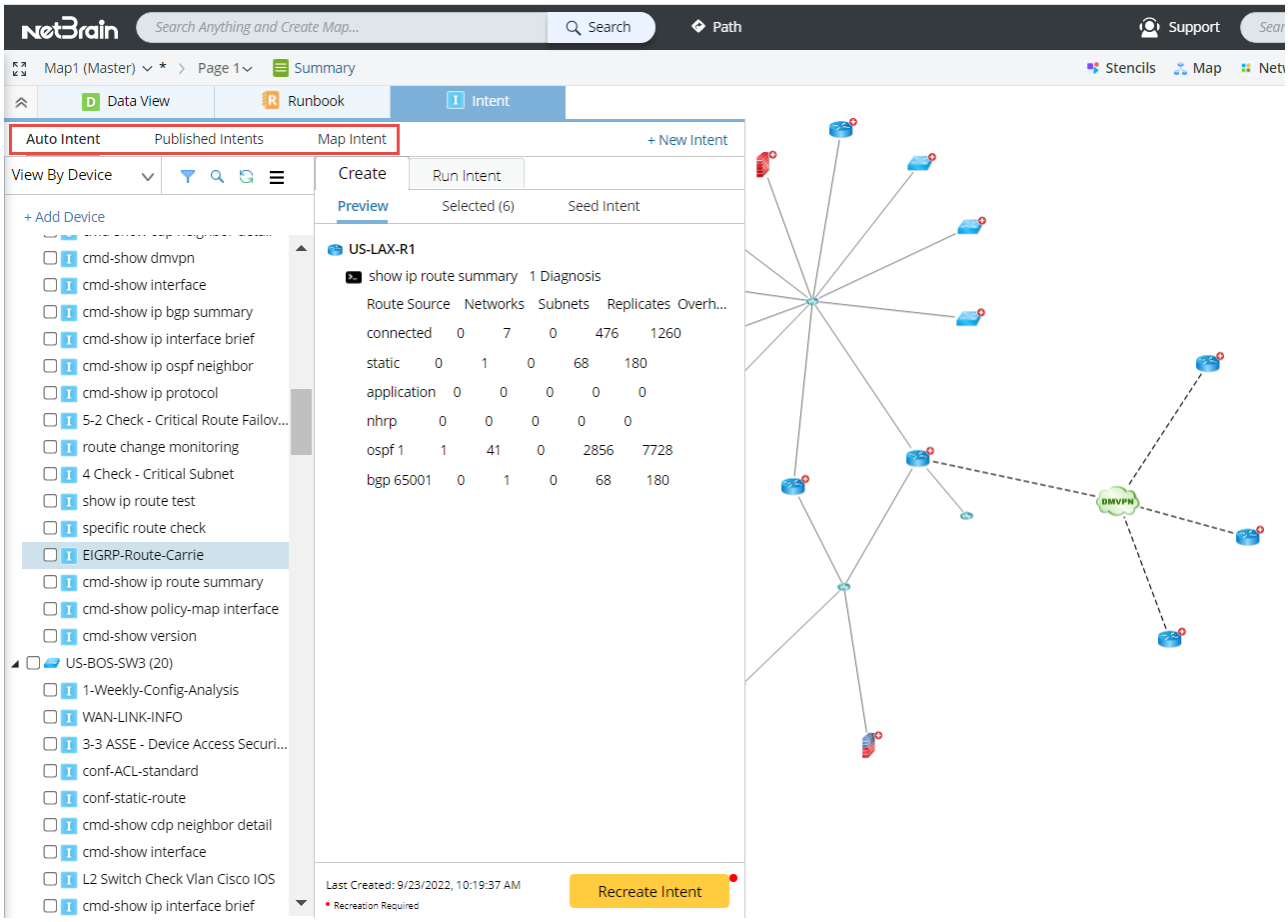


5 Use and Execute Intent

5.1 Clone and Execute NI from Map (IAF)

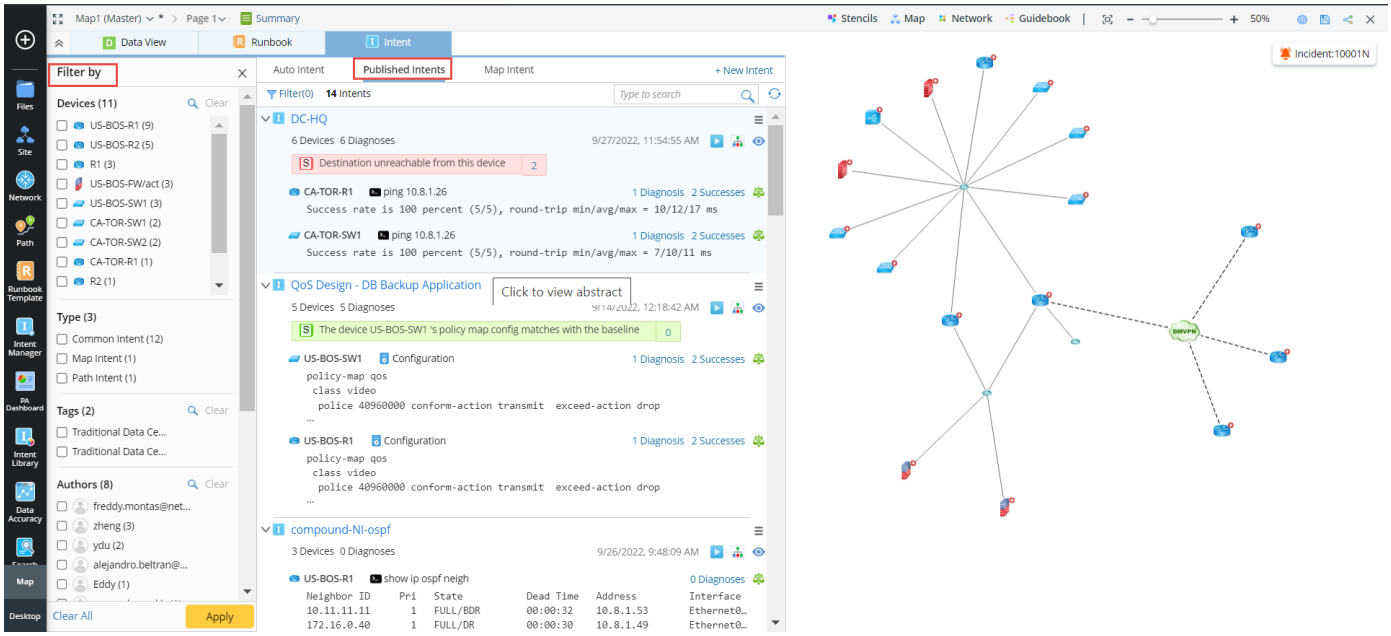
The Intent Pane of a map is fully redesigned with three tabs:

- Auto Intent: the automatically decoded Intents for the map devices.
- Published Intents: only listing the intents published for the map devices.
- Map Intent: an intent dedicated to this map.

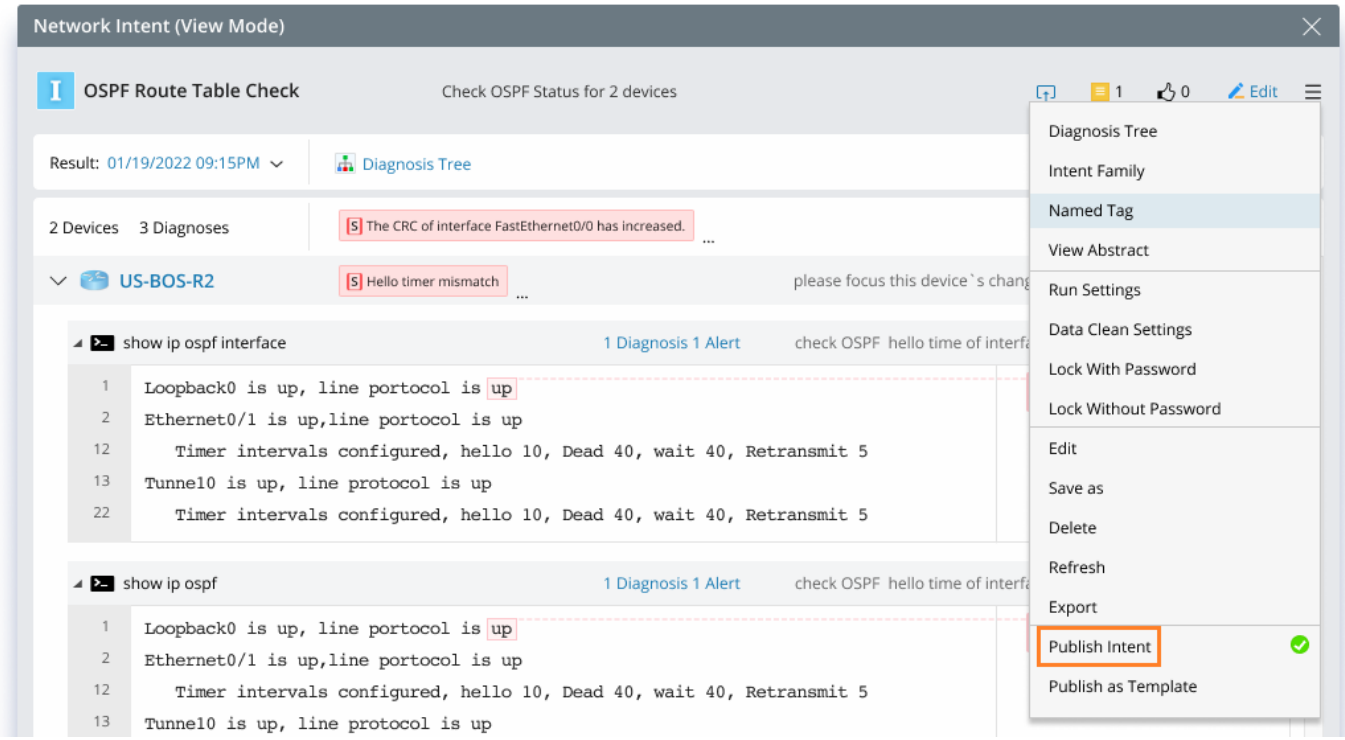


5.1.1 Published Intents

Instead of showing all intents filtered by map devices, which can be too many, R11 only shows the published intents in the map intent pane. The user can filter the published intents further by device, type, tags, and authors.



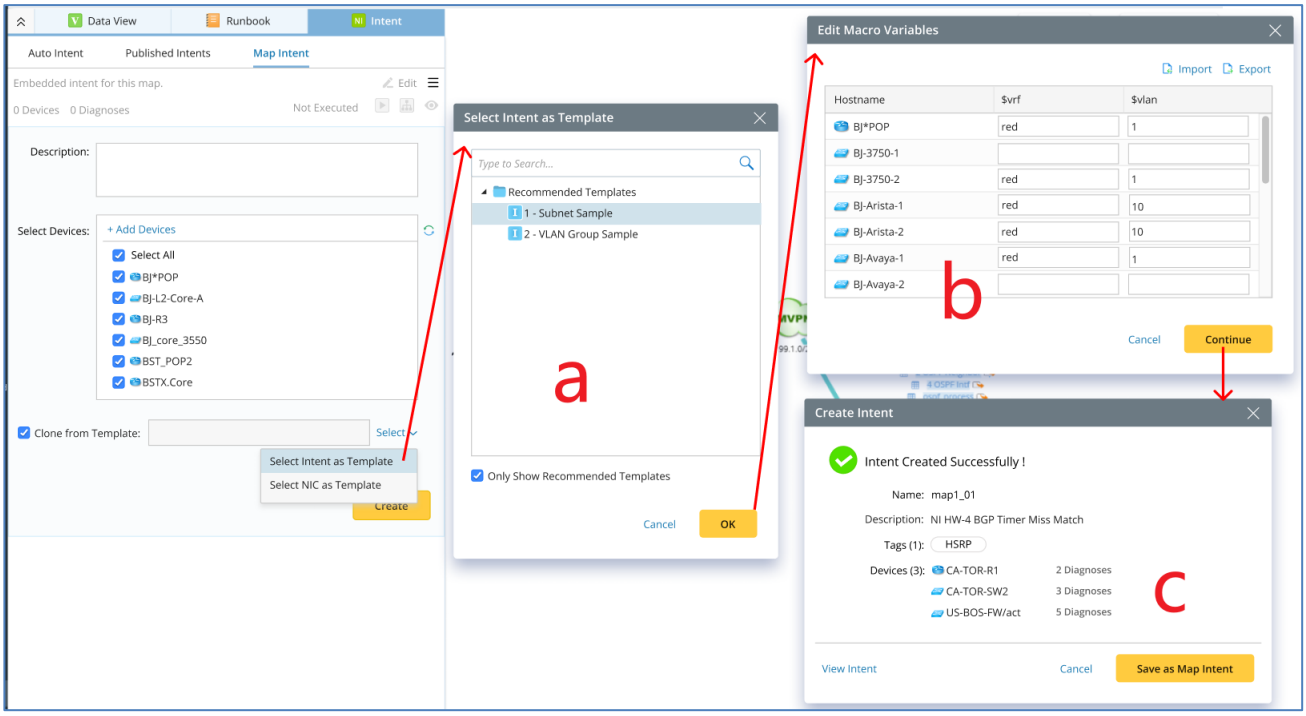
The user can publish an intent from an NI editor or view mode.



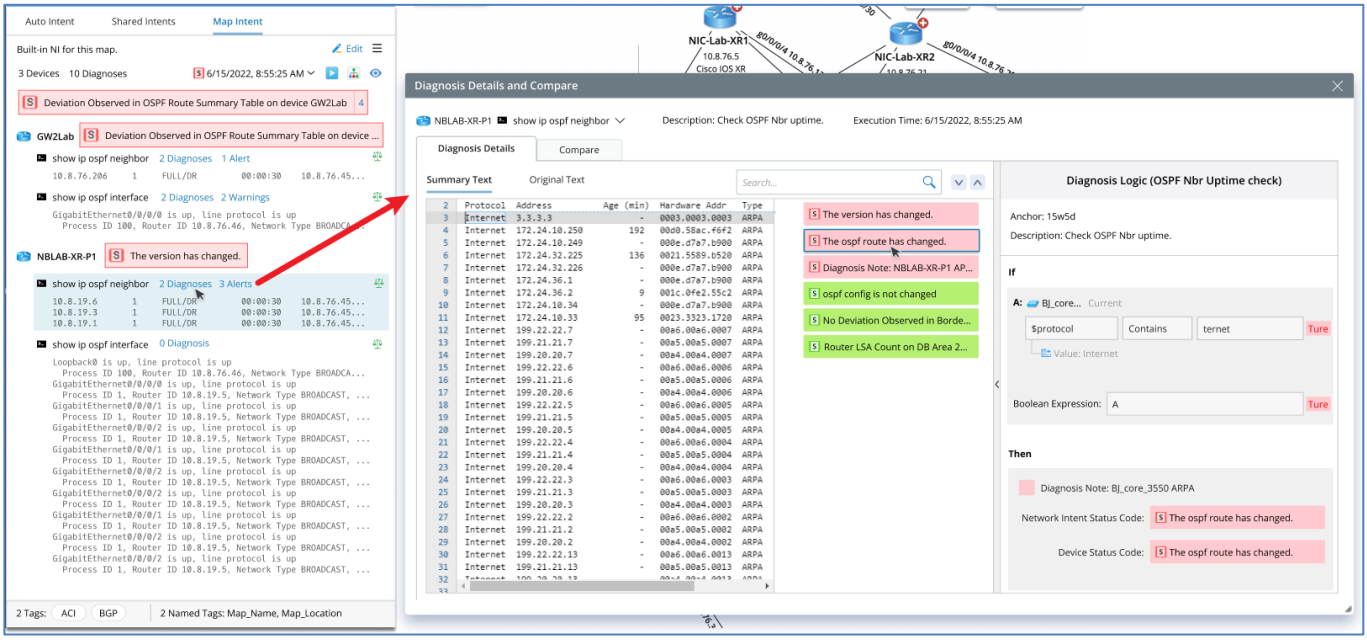
5.1.2 Map Intent

By embedding an intent inside a map (Map Intent), users can have map and intent work side-by-side with an intuitive workflow: create intent along with the map, execute intent while working on the map, and view the intent results in the map.

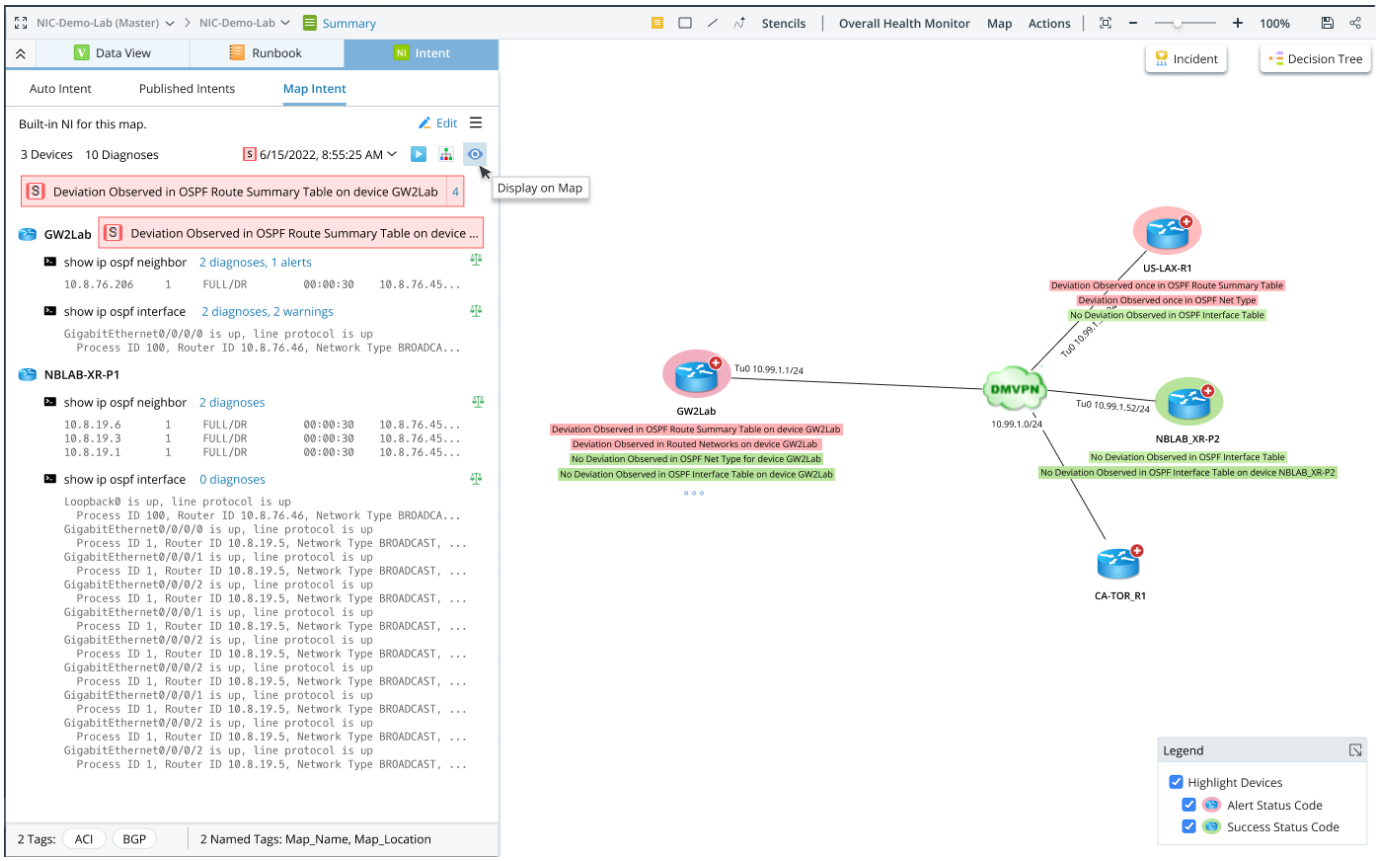
Users can create a map intent from scratch, associate an existing NI as the map intent, or clone from NIT or NIC.



Users can execute the Map Intent, view the diagnosis details, and compare the original data with the data.



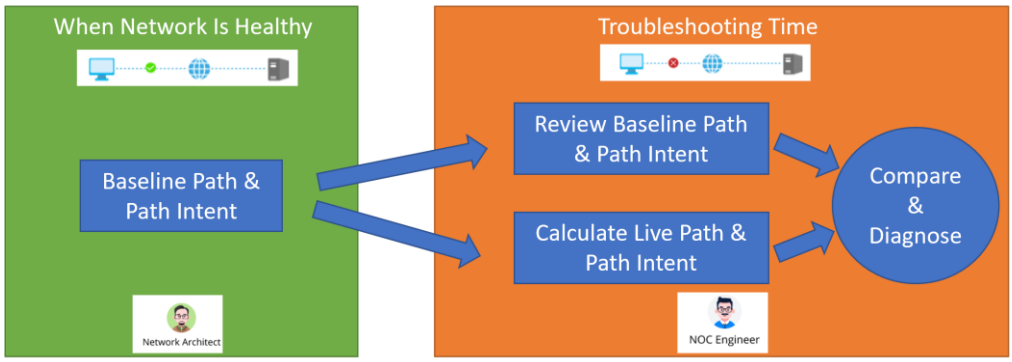
The intent results can also be displayed like a special data view on the map. The devices with alerts will be highlighted and alerted.



5.2 Path Intent

The Path-based Troubleshooting Flow (PBTF) is an essential part of PDAS. With PBFT, the network architect can baseline the critical application and path intents when the network is healthy. Path-related baseline data and diagnosis logic can be programmatically defined inside Path Intent, which can be added into a NIC as static member NIs. Then, this NIC can be used in the trigger diagnosis of TAF with the filter of member NI defined with the path source and destination and the application name. While troubleshooting the application, users

can review the baseline path and path intent results, rerun the associated path NI with the live network data and check the diagnosis results.

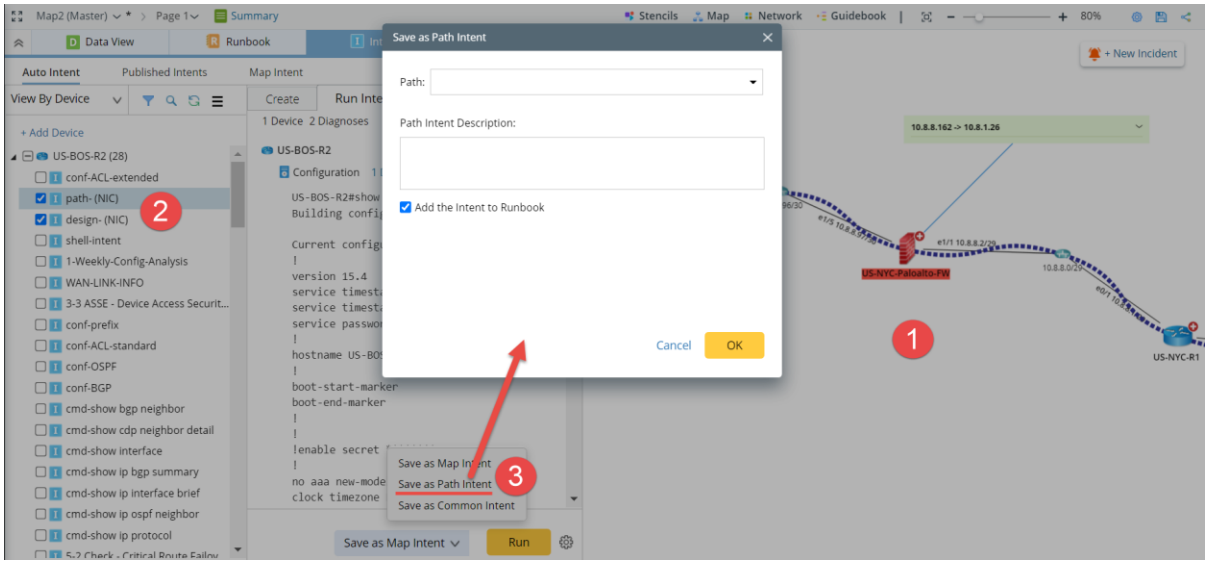


R11 provides more efficient ways to create Path Intent:

- Integrate Auto Intent for Path Intent
- Clone from NIC/NIT Template

5.2.1 Create a Path Intent Via Auto Intent

With the Auto Intent feature, users can create the Path Intent with three steps:



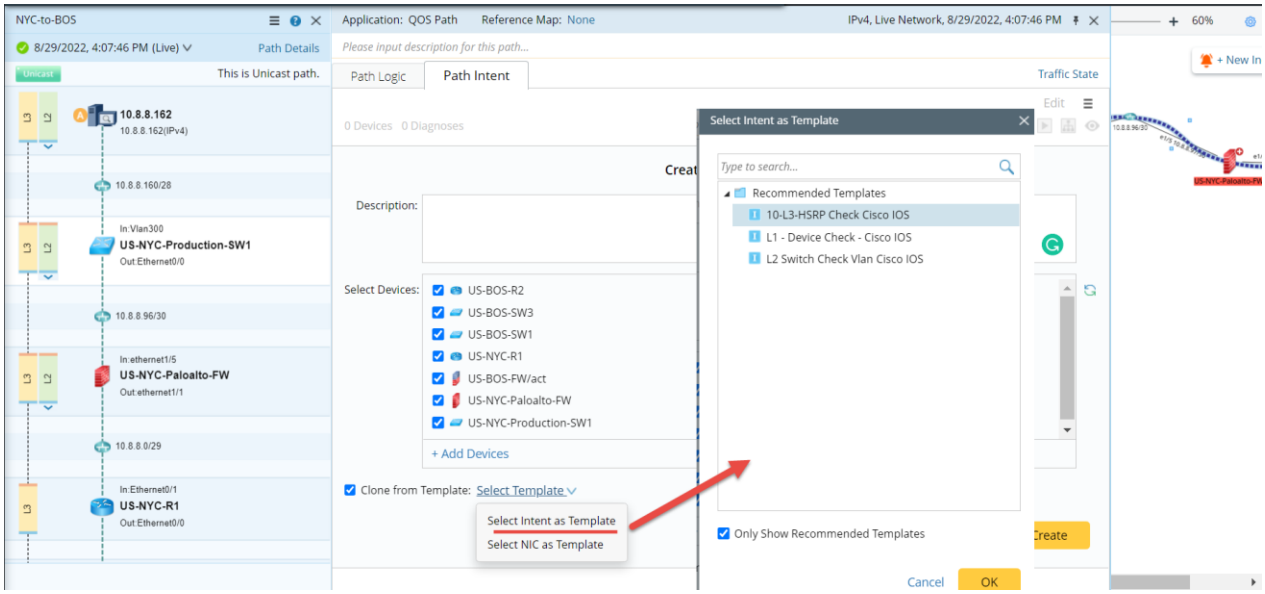
1. Map an application path when the network is healthy.

2. Create an intent via Auto Intent.

3. Save the created intent as the path intent.

5.2.2 Create a Path Intent by Cloning from NIC/NIT

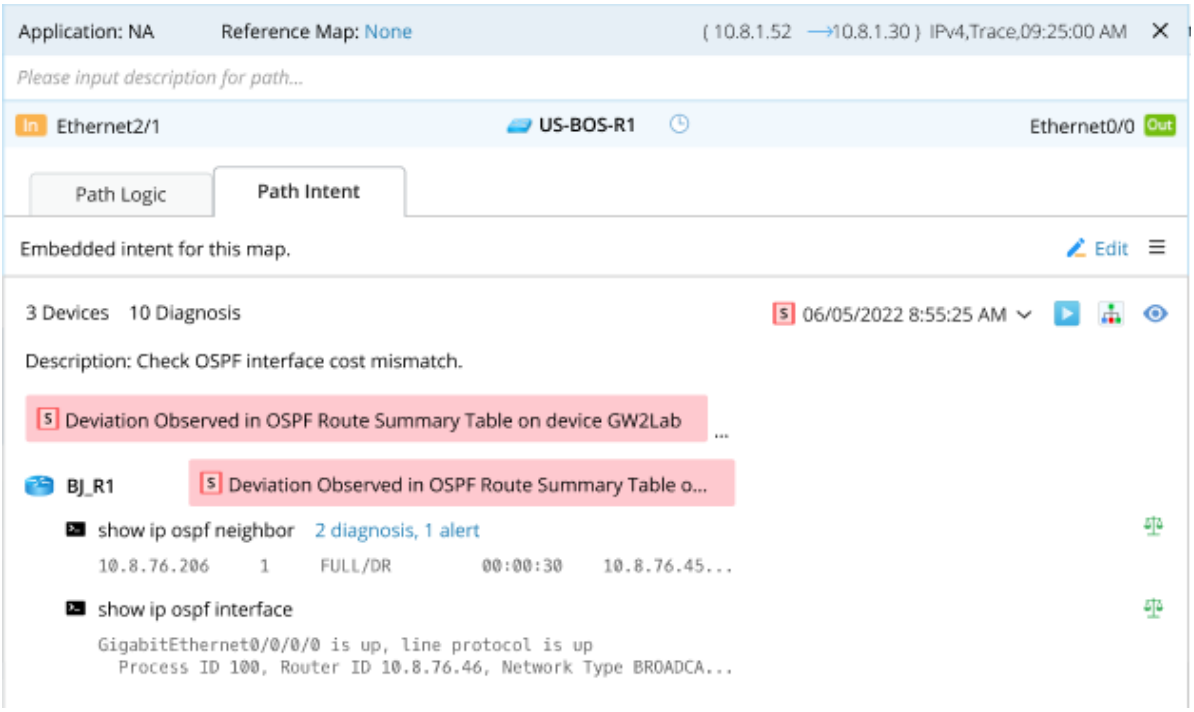
Path intent can also be automatically created by using a suitable intent template or cluster.



5.2.3 Run Path NI and View Results

The Path NI pane is improved to support a smooth workflow. Users can

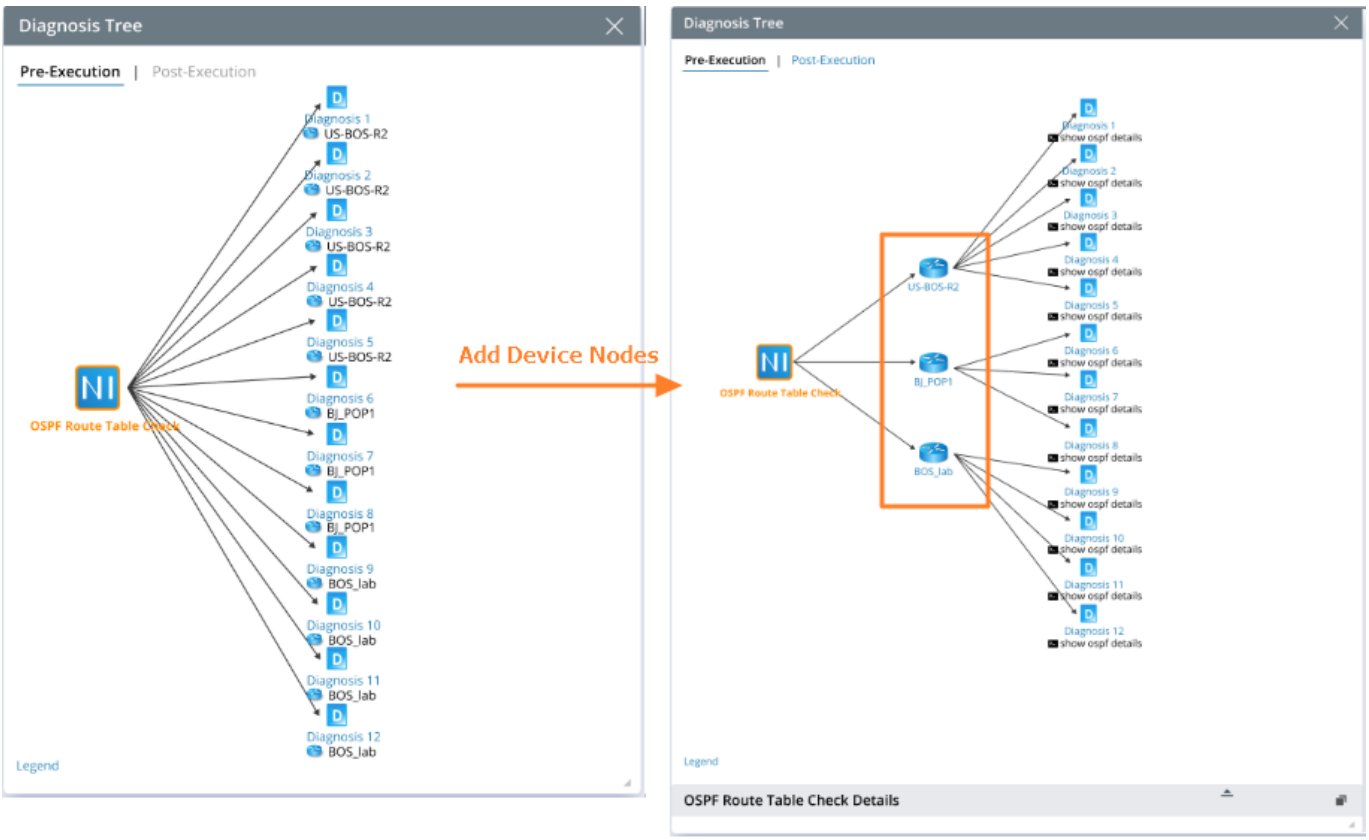
- Execute the Path NI.
- View the diagnosis tree from the Path NI pane.
- Display the Path NI results on the map.



5.3 Diagnosis Tree

R11 improves the diagnosis tree to display more data and add more types of nodes so that the diagnosis tree can be used as a single pane of glass to learn and use NI for diagnosis.

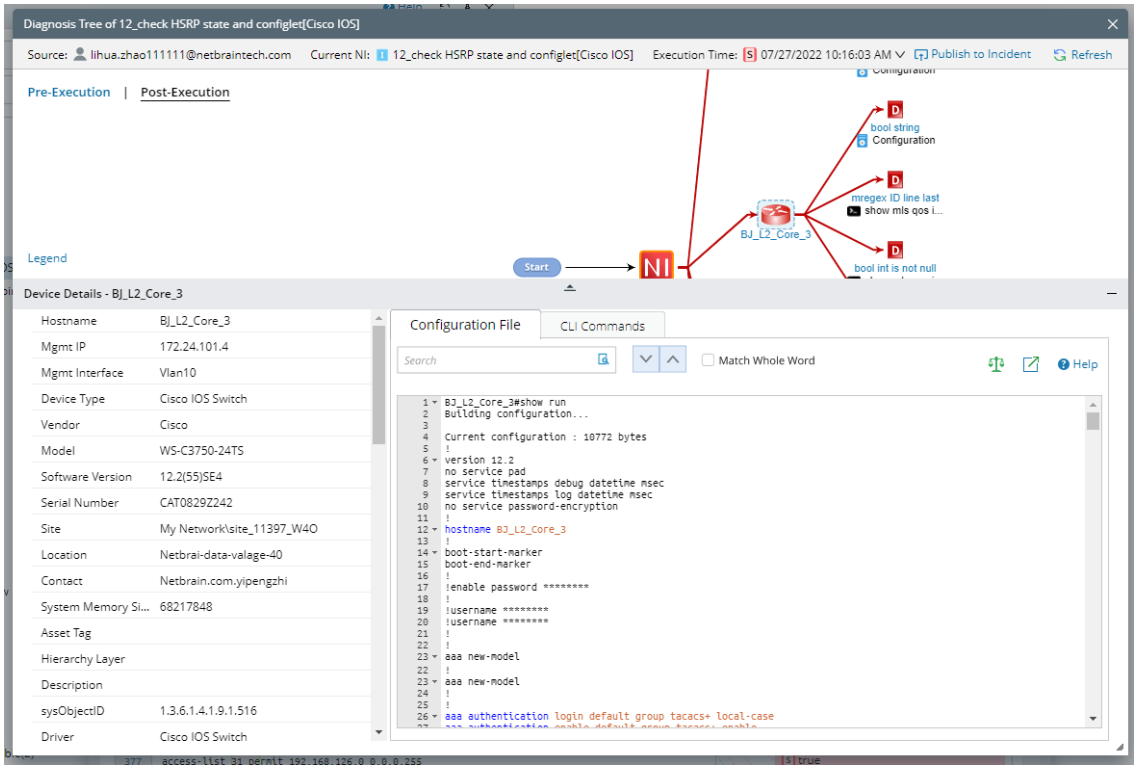
- Add device nodes between the diagnosis node and NI node

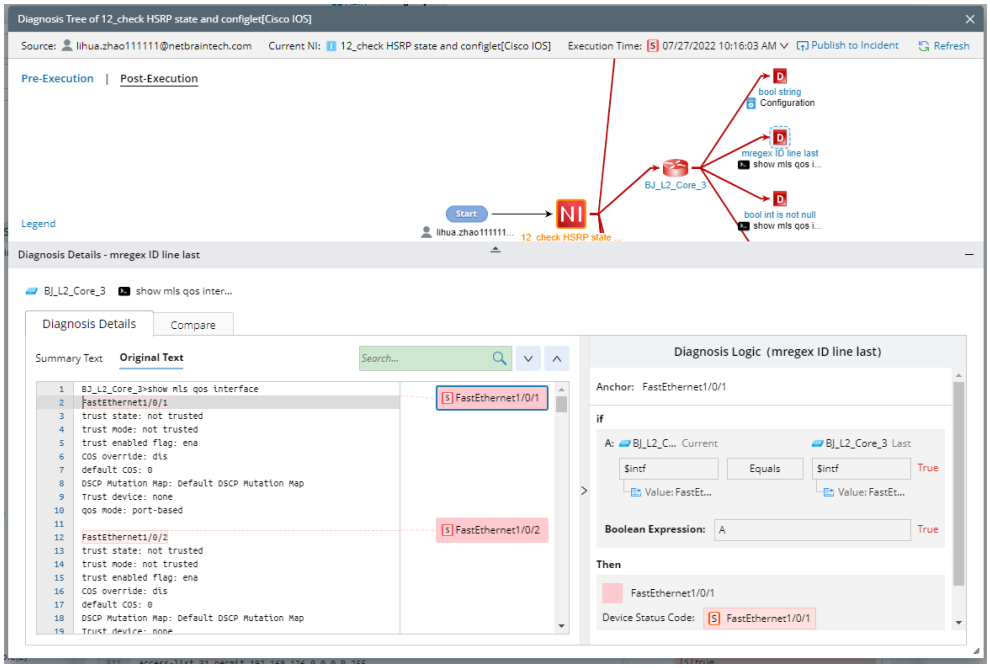


In the previous version, when multiple diagnoses are defined under one NI, users are hard to find out the relationship between devices and diagnoses in the diagnosis tree. Adding the device node between the NI node and the diagnosis node optimizes the diagnosis tree layout, providing better readability.

- Provide Detail Pane for All Nodes

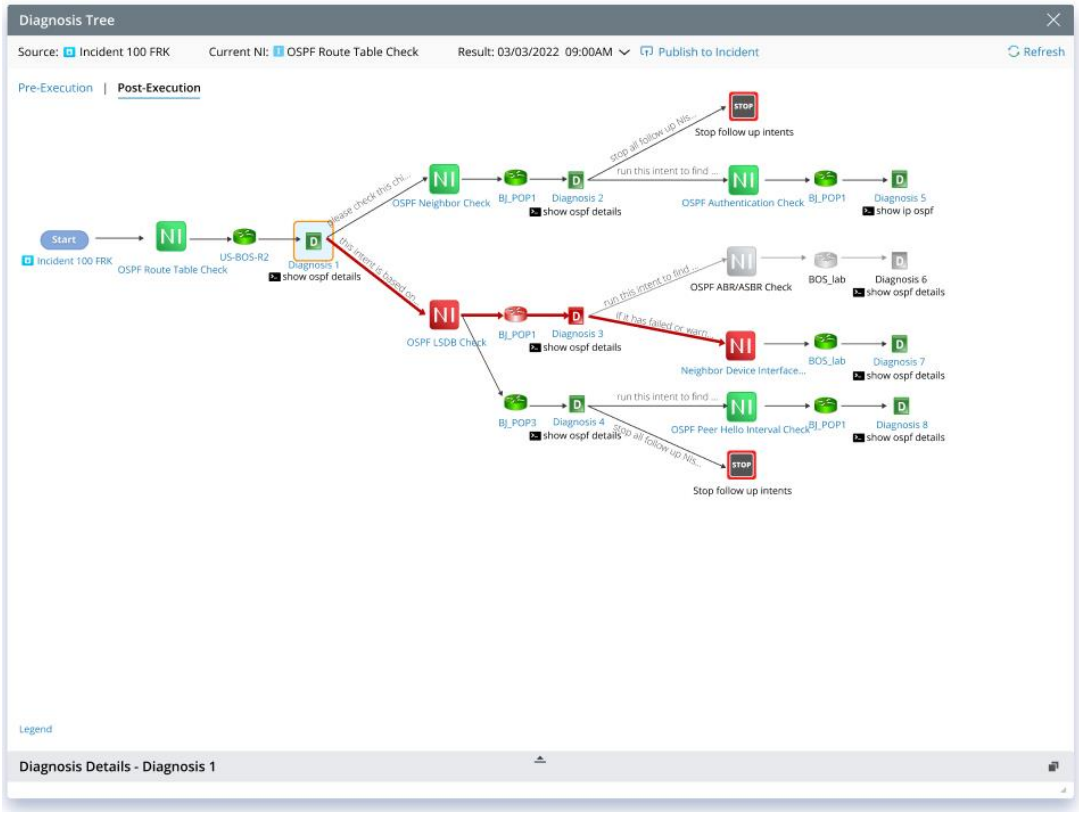
A detail pane is provided so the user can view the node details, such as device and diagnosis details.





- Highlight Nodes and Links with Alerts in Red

  In the previous version, the diagnosis node in the diagnosis tree does not display the alert status. If an alert is generated, the NI node will turn red, but the diagnosis node will not. Therefore, R11 makes All NI nodes, device nodes, and diagnosis nodes with alerts will turn red.



5.4 Self-Service Tools

PDAS Triggered Automation can be consumed through Machine-to-machine API calls or by a human being through Self-service Tools:

- Incident Portal: each NetBrain incident portal can be enabled as the self-service launcher.

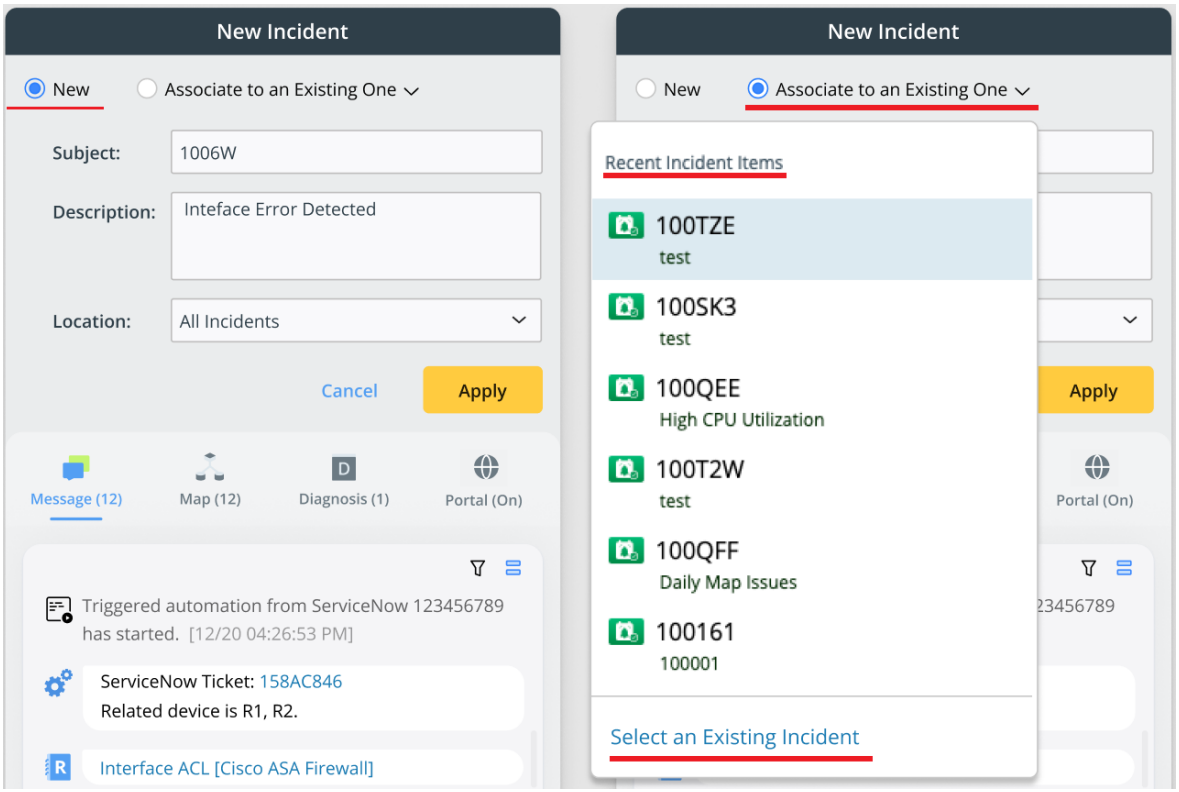
- Email Trigger: fill in an email template and send it to NetBrain to trigger Automation.
- ServiceNow NetBrain App: enable ServiceNow Application to access NetBrain automation.
- Teams Chatbot: deployed through Microsoft Teams store.

Besides improving the Incident Pane/portal, R11 creates a new self-service, **Teams Chatbot**, as a simple way to run NetBrain automation functions and view the results in teams.

5.4.1 Incident Pane/Portal Enhancements

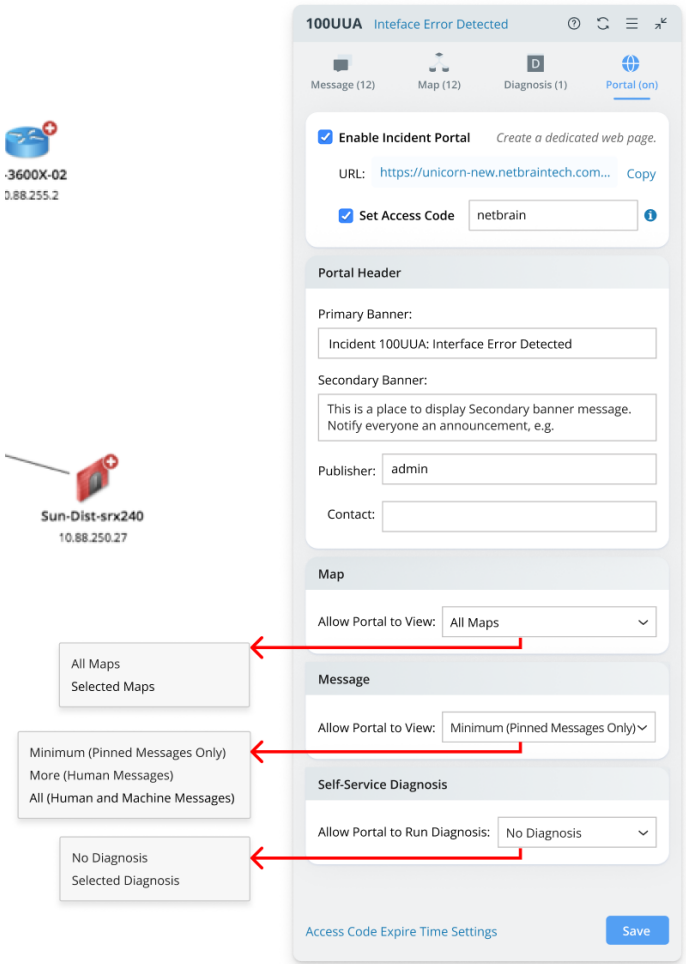
The Incident feature is further improved to provide End Users with a simple flow of using Incident and Portal:

- Redesign UI to create a new incident: simplify the entrance of creating a new incident and clearly shows two ways to create a new incident: from scratch or associate with an existing one.



- Reserve **Interactive Self-Service Diagnosis Only** and Improve the View of Diagnosis Results. To fully present the interactive automation feature of Incident, the operations performed in the Diagnosis Pane will focus on Self-Service Diagnosis only. The triggered diagnosis results are not shown in the Message pane.

- Emphasize Portal settings: Simplify the flow of setting the incident portal. Users will be more likely to set up and use the portal for smooth collaborative troubleshooting.



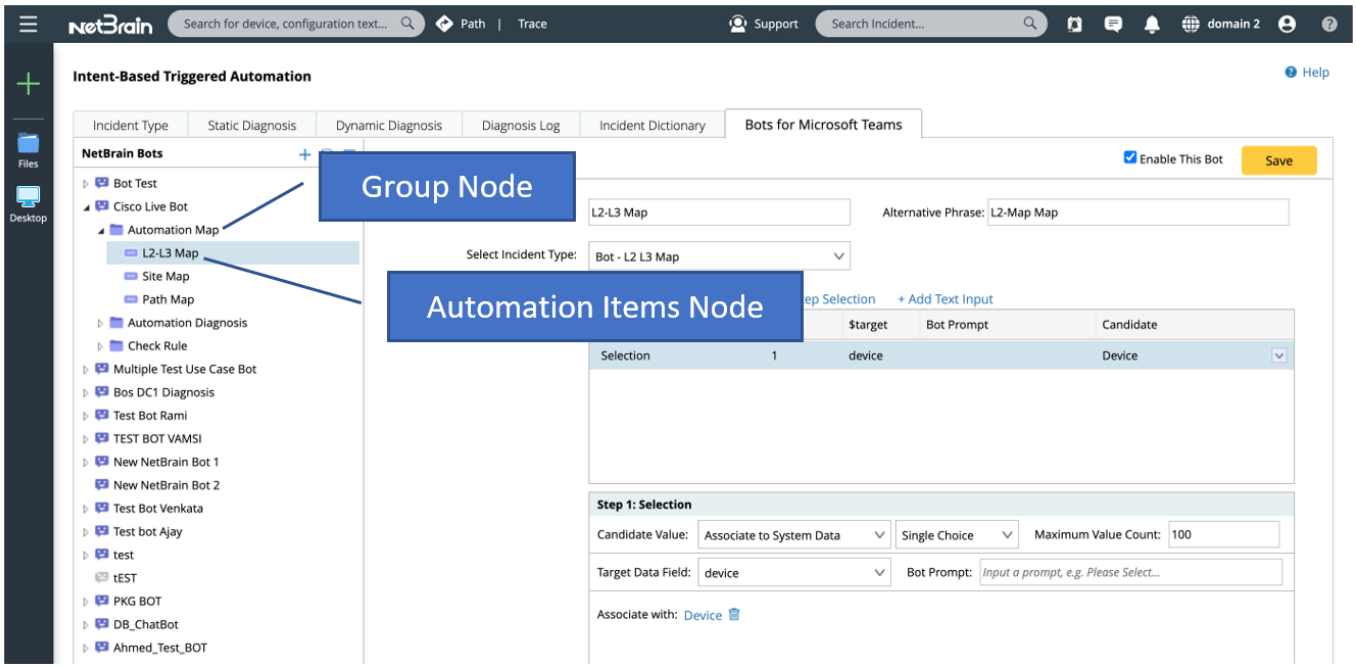
- Share Personal Copy to Incident Portal. Users can also share a personal copy with portal users.

5.4.2 Teams Chatbot

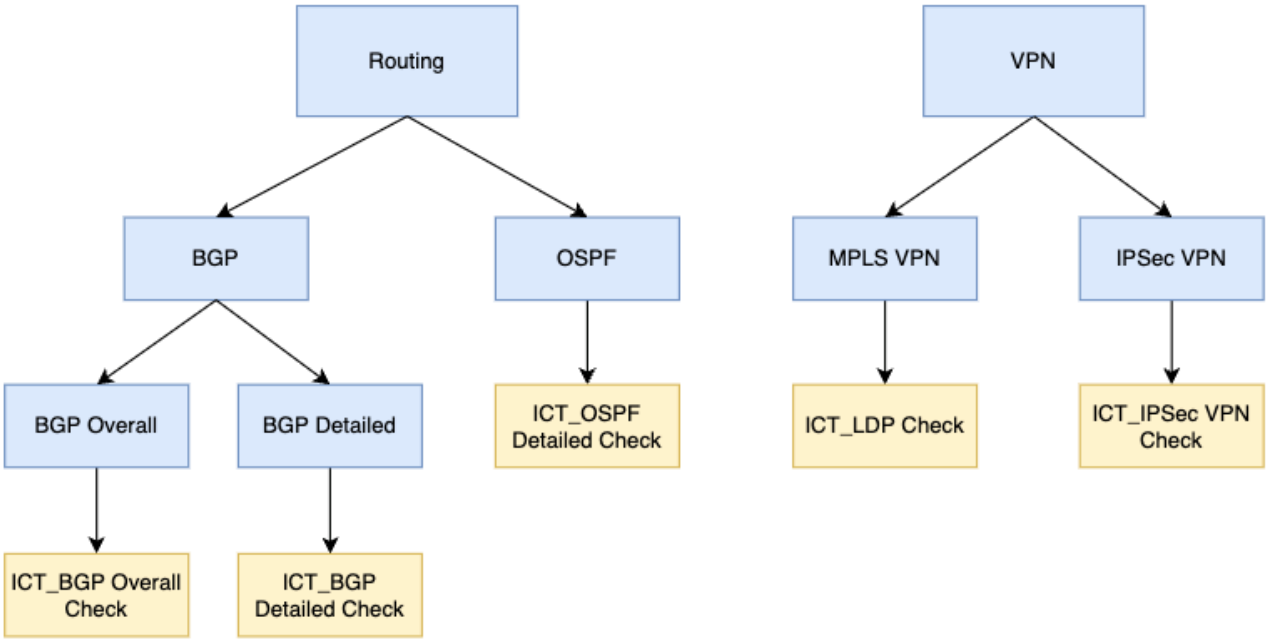
R11 creates a Microsoft Teams Chatbot as a simple way to run NetBrain automation functions and view the results in teams, including:

- Create a network map
- Automate network problem diagnosis
- Enforce design rules and best practices, fully customizable without any coding.

The bot conversation flow can be easily customized in the NetBrain system. The group and automation node form a tree structure to guide the user through the chat.



A sample bot for troubleshooting the routing issue:


Routing
BGP
OSPF
BGP Overall
BGP Detailed
ICT_OSPF Detailed Check
ICT_BGP Overall Check
ICT_BGP Detailed Check
VPN
MPLS VPN
IPSec VPN
ICT_LDP Check
ICT_IPSec VPN Check

6 Other Enhancements

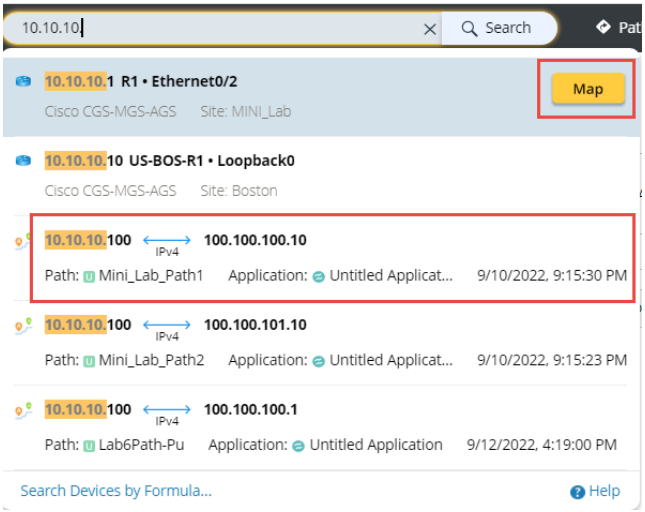
6.1 Incident-based Collaboration Workflow Enhancement

One common interactive collaboration workflow is: Search the digital twin (e.g., IP address), create the map from the search results, open the Incident pane or create a new Incident, and share the incident portal with others.

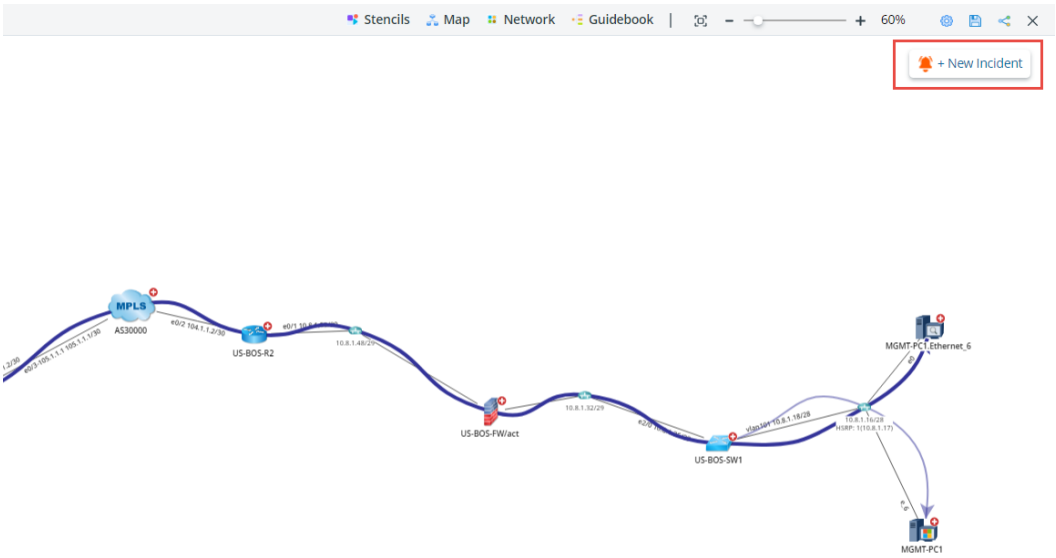
Along this workflow, R11 makes the following improvements:

- Improve the search IntelliSense pane to show more related results and allow users to create the map directly from an entry.

  For example, entering an IP address or partial IP address, all device interfaces with IP and historical A/B paths matching the IP address are displayed. You can directly map the interface or the path.



- After mapping the searched result, users can create a new incident from the map, which can be associated with an existing Incident.



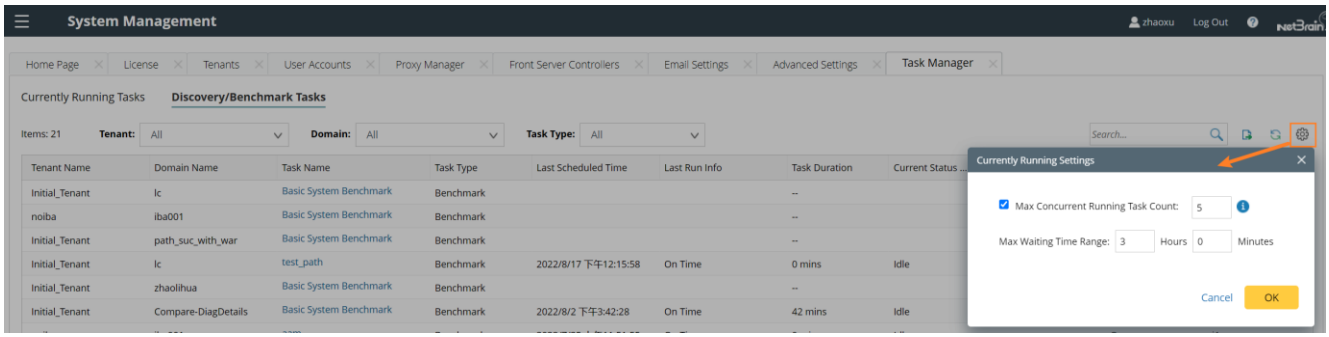
- From the Incident pane, users can enable the Incident Portal to share the map and automation results with others. See Incident Pane/Portal Enhancements for details.

6.2 Benchmark Schedule Enhancement

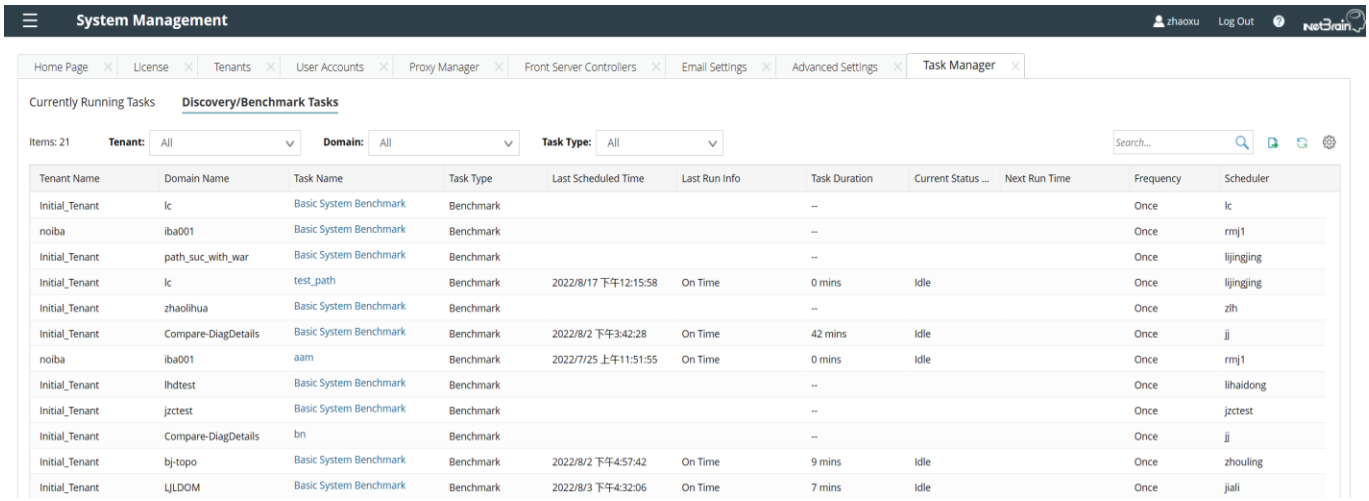
If the system has many domains, such as Service Providers, it is hard to set up the benchmark tasks to limit the concurrent background tasks. NetBrain R11 allows users to configure rules of the concurrently running tasks and view all the scheduled benchmark/discovery tasks on the System Management page to adjust the frequency more easily.

- Configure Rules of Concurrently Running Tasks

Users can set the maximum number of concurrently running tasks to avoid conflicts and specify the maximum waiting time (for running tasks to improve task execution efficiency. The task will be skipped and not executed if the waiting time exceeds the specified waiting time range.



- A **Discovery/Benchmark Tasks** tab is added to the Task Manager on the System Management page to centrally display all the scheduled Discovery/Benchmark tasks across all domains.



- Edit the Task Frequency to Avoid Conflicts

  Users can further adjust the task frequency based on the running information to avoid schedule overlapping.

6.3 Email Change Analysis Report

The global change analysis report allows users to track the data change of the entire network and presents the result in the CA Report. In the previous version, if users need to view, save or share a CA report, the only way is to log in to the IE system and manually export the CA report to the local machine. To simplify the process and report network changes via email daily, R11 enables users to receive the CA report via email at a certain frequency.

The scheduled email alert will inform the email recipient of a summary of the CA report, such as the comparison inputs, results, and the involved tenant. Also, a copy of the CA report can be attached to the email as needed.