

Juniper Apstra Cloud Services Edge Setup Guide

Published
2025-08-27

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Juniper Apstra Cloud Services Edge Setup Guide
Copyright © 2025 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <https://support.juniper.net/support/eula/>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide | iv

1

Introduction

Introduction | 2

2

Prerequisites

Prerequisites | 4

3

Adopt the Edge

Adopt the Juniper Apstra Cloud Services Edge and Enable Juniper Apstra Flow | 7

4

Deploy

Deploy the Docker Edge Container | 12

5

Import a CPU and Memory probe in Juniper Apstra for Apstra Cloud Services

Import a CPU and Memory Probe in Juniper Apstra for Apstra Cloud Services | 20

6

Post-Setup

Juniper Apstra Cloud Services Edge Post-Setup: Config Changes and Troubleshooting | 25

7

Replace the SSL Certificate of Juniper Apstra's Nginx Controller

Replace the SSL Certificate of Juniper Apstra's Nginx Controller | 28

8

Internal Variables

Internal Variables for Juniper Apstra Cloud Services Edge Configuration | 31

About This Guide

Use this guide to adopt the Juniper Apstra Edge into your Apstra-managed fabric. The Juniper Apstra Edge is a hardware-agnostic virtual device that runs within a container in the data center. The Edge functions like a proxy device in an Apstra-managed fabric. It receives DC network event and anomaly data from Juniper Apstra and Apstra Flow Server, and forwards that data to Apstra Cloud Services. This guide walks you through the process of adopting the Juniper Apstra Edge and downloading and installing the Edge container.

Juniper Apstra Cloud Services supports Flow Data. Flow Data, installed in an Apstra Flow VM, is a comprehensive, scalable, high-performance flow collector and analyzer. It includes an array of network flow analytics that helps you understand application performance and usage across your Apstra-managed network. Flow data features out-of-the-box dashboards and advanced analytics capabilities such as fine-grain filtering and customizable charts.

1

CHAPTER

Introduction

IN THIS CHAPTER

- [Introduction](#) | 2
-

Introduction

This setup guide walks you through the process of “adopting” the Juniper Apstra Edge to enable Juniper Apstra Cloud Services (ACS), and enabling [Flow Data](#). It then guides you through the process of spinning up the Juniper Apstra Edge Docker container.



NOTE: This Edge setup guide is for Juniper Apstra version 5.0 and earlier. If you are using Juniper Apstra 5.1, you can install the Edge from within Apstra. For more information, see [ACS Edge Installation Workflow](#).

The Apstra Edge is a hardware-agnostic virtual device that runs within a container in the data center. The Apstra Edge is a required component for enabling ACS and its supported features, like [Marvis Virtual Network Assistant for Data Center](#), Service Awareness, and Impact Analysis. For more information about the features of Juniper ACS, see the [Release Notes: Juniper Apstra Cloud Services](#), and the [Juniper Apstra User Guide](#).

The Edge functions like a proxy device in an Apstra-managed fabric. It receives DC network event and anomaly data from Juniper Apstra and Apstra Flow Server, and forwards that data to Apstra Cloud Services. Apstra Cloud Services analyzes and visualizes the data, enabling root-cause analysis and a proactive response. The Edge is typically installed onto a Juniper Apstra server. However, you can also install it onto a different Linux host that has reachability to both the Juniper Apstra Server and Juniper ACS.

Apstra Flow Data, installed on an Apstra Flow VM, collects and analyzes DC network flow traffic. Apstra Flow Data integrates seamlessly into your organization, providing visibility and highly granular insight into your network traffic.

2

CHAPTER

Prerequisites

IN THIS CHAPTER

- [Prerequisites](#) | 4
-

Prerequisites

Before setting up the Juniper Apstra Edge for Apstra Cloud Services, ensure that you have the following:



NOTE: Although it is not required, we strongly recommend that you install Juniper Apstra Flow. You can still use Apstra Cloud Services without Apstra Flow installed, but you will not benefit from new features like Service Awareness, Impact Analysis, or Dashboard. Additionally, new and exciting features in future releases require Juniper Apstra Flow.

For information about how to install Juniper Apstra Flow, see the [Juniper Apstra Flow Installation Guide](#).

- Your Apstra-managed fabric is fully operational and running Juniper Apstra version 4.2.1 or later.



NOTE: Juniper Apstra version 4.2.1 supports the latest ACS features. If you are using Juniper Apstra 4.2.0, you can enjoy Marvis Virtual Network Assistant for Data Center, but you will not benefit from the latest ACS features.

For information about the latest Apstra Cloud Services features, see the [Juniper Apstra Cloud Services Release Notes](#).

- Active Juniper Apstra Cloud Services account.

For information about how to activate a Juniper Apstra Cloud Services account, see the [User Activation and Login](#) section of the Juniper Apstra Cloud Services User Guide.

- Active Marvis subscription.

For information about Marvis subscription options, see [Subscriptions for Marvis](#).

- Active Juniper Apstra license.

For information about Juniper Apstra software licenses, see the [Juniper Apstra Datasheet](#).

- Internet connectivity that provides the Edge container with access to Juniper Apstra Cloud Services.
- Download the the Juniper Apstra Edge image files ([Edge image](#)), under Application Tools, onto the host where you plan on running the installation. This host must be able to reach the Juniper Apstra controller.
- The RECEIVER_PORT (default 9595) and the EDGE_SERVER_PORT (default 8081) do not have existing services using those ports on the host where the apstra-edge container will be installed.

In the unlikely event that there is a port conflict, you can manually change these port numbers in the `docker-compose.yml` file of the .tgz Edge image download. Internal Edge services use these ports to stream vital data.

3

CHAPTER

Adopt the Edge

IN THIS CHAPTER

- [Adopt the Juniper Apstra Cloud Services Edge and Enable Juniper Apstra Flow | 7](#)
-

Adopt the Juniper Apstra Cloud Services Edge and Enable Juniper Apstra Flow

Follow these steps to Adopt the Juniper Apstra Cloud Services Edge and enable Juniper Apstra Flow for your Apstra-managed fabric. Note that the Juniper Apstra Edge and Flow Server are separate VMs. After you enter the information for each VM, the aos-edge configuration is updated and the Edge can communicate with the Apstra Flow VM.



NOTE: This guide assumes that you have Juniper Apstra Flow installed. For information about how to install Juniper Apstra Flow, see the [Juniper Apstra Flow Installation and Upgrade Guide](#).

1. Using the Juniper Apstra Cloud Services URL that was provided during the entitlement process, access your organization using the appropriate login credentials. Alternatively, to access the GUI as the first admin user without an invite:
 - a. Access the GUI directly at <https://dc.ai.juniper.net>.
 - b. Click **Create Account** and fill in the required information.
 - c. Click **Create Account**.
ACS sends a verification e-mail to activate your account.
 - d. Click **Validate me!** in the e-mail body.
The New Account page appears.
 - e. Click **Create Organization**.
 - f. Type a unique name for your organization and click **Create**.
The New Account page appears.
 - g. Click the organization on the New Account page.
The Marvis Actions page displays.
2. On the left side of the page, select **Organization > Inventory**.
 - a. From the **Inventory** tab, select **Adopt Apstra Edge** on the top right of the page.
 - b. Fill in the following fields:
 - Edge Name: The name of the Juniper Apstra Edge instance
 - Management URL: The management URL of your Juniper Apstra Cloud Services UI
 - Username: The username you use to access the Juniper Apstra Cloud Services UI

- Password: The password you use to access the Juniper Apstra Cloud Services UI
- Data Center Edge Download Location: The Juniper Downloads URL for the latest Apstra Edge distribution. Use the following Juniper Downloads link to download the latest Apstra Edge version: <https://support.juniper.net/support/downloads/?p=apstra>.

c. Select the toggle for **Flow Server Details** and fill in the following fields:

- Flow Server Name: The name of the Apstra Flow Server instance
- Management URL: The IP address or hostname of your Apstra Flow VM, with the management port.

`https://<ip-or-hostname-of-flow-data-vm>:9200/`

.

- Username: The username you use to access the Juniper Apstra Flow Server dashboard UI. The default username is **admin**.
- Password: The password you use to access the Juniper Apstra Flow Server dashboard UI. The default password is **Apstra-Flow5**.



NOTE: We recommend that you change the default password after your log in.

d. Select **Adopt**.

Adopt Apstra Edge ✕

Edge Name ⓘ

Management URL ⓘ

Username ⓘ

Password ⓘ

Flow Server Details ☒


Flow Server Name ⓘ

Management URL ⓘ

Username ⓘ

Password ⓘ

Apstra Edge Download Location



Adopt

Cancel

3. Retrieve the registration code for the Juniper Apstra Cloud Services Edge instance.
The registration code associates the Juniper Apstra Edge instance with Marvis.
 - a. Select the box next to your Edge component and select **More > Get Registration Code** at the top of the page.

- b. Copy the code and select **Close**.



NOTE: This registration code registers the Apstra Edge with Juniper Apstra Cloud Services, which uses the code to retrieve unique organization ID, secret, and device ID during Edge installation. These IDs must be stored securely as they cannot be retrieved after the initial setup is complete. is required to initialize the Juniper Apstra Cloud Services Edge container.

4. Save the registration code as it is needed for the next steps.

RELATED DOCUMENTATION

[Juniper Apstra Cloud Services User Guide](#)

[Juniper Apstra Flow Installation Guide](#)

4

CHAPTER

Deploy

IN THIS CHAPTER

- [Deploy the Docker Edge Container | 12](#)
-

Deploy the Docker Edge Container

Follow these steps to set up the container environment and install the Edge container using a local image. Note that the latest Apstra Edge version is 0.0.78. This Edge version supports the latest Apstra Cloud Services features, such as Service Awareness, Impact Analysis, and Dashboard.



NOTE: This setup process uses Juniper Apstra Edge distribution version 0.0.78. The version that you download might be a more recent version. For information about ACS compatibility, see [Juniper Apstra Cloud Services Compatibility](#).

1. Untar the tar.gz file that you downloaded.



NOTE: You can download and install the Edge file on the same VM as the Apstra Server, or a separate VM. For this example, the Edge file is installed on the same VM as the Apstra Server. The ACS Edge must have connectivity to both Juniper Apstra and ACS.

This creates the following directory: `apstra-edge-0.0.78`.

```
root@user:~# tar -xvzf apstra-cloud-services-edge_0.0.78.tar.gz
apstra-edge-0.0.78/
apstra-edge-0.0.78/docker-compose-0.0.78.yml
apstra-edge-0.0.78/ssl-keys/
apstra-edge-0.0.78/ssl-keys/ep-term.ai.juniper.net.cer
apstra-edge-0.0.78/apstra-edge-container-0.0.78.tgz
root@user:~#
```

2. Modify the following lines in the `docker-compose.yml` file. Enter `ep-term.ai.juniper.net` for the `CLOUD_TERM`.

```
root@user:~# vi docker-compose-0.0.78.yml
- REGISTRATION_KEY=<registration-code>
- CLOUD_TERM=ep-term.ai.juniper.net
```

This registration code is retrieved from Step 3 in the "[Adopt the Juniper Apstra Cloud Services Edge and Enable Juniper Apstra Flow](#)" on page 7 section, after adopting the Edge. The registration code registers the Apstra Edge with Juniper Apstra Cloud Services.



NOTE: The Juniper Apstra Edge uses the registration code to retrieve unique organization ID, secret, and device ID during Edge installation. These IDs must be stored securely as they cannot be retrieved after the initial setup is complete.

The CLOUD_TERM is the service that runs in the cloud, which is the entry point for any connectivity for any Edge component to communicate with the cloud.

For a list of internal configuration variables and their functions, see "[Internal Variables for Juniper Apstra Edge Cloud Services Configuration](#)" on page 31.



NOTE: These variables are meant for internal use and are not required to set up the Edge component. We do not recommend overriding these variables in production.

3. Create a directory for installing the Edge container. Copy the docker-compose-0.0.78.yml file from the tar.gz. and rename it to docker-compose.yml.

```
root@user:~# mkdir apstra_edge
root@user:~# cp apstra-edge-0.0.78/docker-compose-0.0.78.yml apstra_edge/docker-compose.yml
```

4. Copy the container file into the apstra_edge directory.

```
root@user:~# cp ~/apstra-edge-0.0.78/apstra-edge-container-0.0.78.tgz ~/apstra_edge
```

5. (Optional) Verify that no existing image is present.

```
root@user:~# cd apstra_edge
root@user:/apstra_edge# docker images |grep edge
```

6. Load the Docker image.

```
root@user:/apstra_edge# docker load < apstra-edge-container-0.0.78.tgz
dd3a0446c8dc: Loading layer [=====] 2.048kB/
2.048kB
20926e4376db: Loading layer [=====] 19.25MB/
19.25MB
91a5e17f426c: Loading layer [=====] 3.072kB/
3.072kB
70f18eed95b4: Loading layer [=====] 4.096kB/
4.096kB
```

```

437b361ffd18: Loading layer [=====>] 20.55MB/
20.55MB
fa960967b411: Loading layer [=====>] 20.56MB/
20.56MB
b2db1e6c6bcd: Loading layer [=====>] 2.56kB/
2.56kB
ed13056991f4: Loading layer [=====>] 6.144kB/
6.144kB
e051cc879cb0: Loading layer [=====>] 3.256MB/
3.256MB
4c412efeef84: Loading layer [=====>] 20.56MB/
20.56MB
a9547726ed61: Loading layer [=====>] 2.56kB/
2.56kB
Loaded image: apstra-edge:0.0.78

```

7. Verify that the apstra-edge image is loaded.

```

root@user:/apstra_edge# docker images |grep edge
apstra-edge          0.0.78                c183eb098689    8 days ago
273MB

```

8. We recommend that you replace the Juniper Apstra self-signed certificate with a publicly-signed certificate. To proceed with a publicly-signed certificate, follow the directions at, ["Replace the SSL Certificate of Juniper Apstra's Nginx Controller" on page 28.](#)

The Juniper Apstra Edge connects to the Juniper Apstra controller using the management URL provided in the Juniper Apstra Cloud Services entitlement process. This management URL is configured with an IP address. The Apstra controller requires an SSL certificate with a Subject Alternative Name containing this IP.

9. (Optional) To proceed with the self-signed certificate on the Apstra controller instead of a publicly-signed certificate, you must add the following line (- AOS_INSECURE_SKIP_VERIFY=true) to the docker-compose.yml file after the two environment variables that you previously entered (REGISTRATION_KEY=<registration-code>, CLOUD_TERM=ep-term.ai.juniper.net):

```

root@user:~# vi docker-compose.yml
- REGISTRATION_KEY=<registration-code>
- CLOUD_TERM=ep-term.ai.juniper.net

- AOS_INSECURE_SKIP_VERIFY=true

```



NOTE: You must perform Steps 9 and 10 regardless of whether you use a self-signed certificate or a publicly-signed certificate for the Apstra controller.

10. Copy the EP-Term SSL certificate to the correct directory, and add read/write permissions. This enables the Juniper Apstra Cloud Services Edge to validate SSL certs for server authentication.

```
root@user:~# cd /etc/ssl/certs
root@user:/etc/ssl/certs# sudo cp ~/apstra-edge-0.0.78/ssl-keys/ep-term.ai.juniper.net.cer .
root@user:/etc/ssl/certs# sudo chmod 644 ep-term.ai.juniper.net.cer
```

11. Update the certificates.

```
root@user:/etc/ssl/certs# sudo update-ca-certificates
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
```

12. Spin up the Docker Edge container from the `apstra_edge` directory.

The `docker compose up -d` command initializes the services listed in `docker-compose.yml` file. Applying the `.yml` file also creates a volume named `apstra_edge_apstra_edge_store/`. This volume is mounted at `/var/lib/docker/volumes/apstra_edge_apstra_edge_store/_data/`.



NOTE: The Juniper Apstra Cloud Services instance supports both `docker compose` and `docker-compose` commands.



NOTE: You must run the `docker compose up -d` command from within the same directory where the `docker-compose.yml` is located. The file must also be named `docker-compose.yml` or the command will not work as intended.



NOTE: Ensure that you back up `/var/lib/docker/volumes/apstra_edge_apstra_edge_store/_data/`. The Edge instance uses this mount to restart connectivity in the event of a system crash.

```
root@user:~/apstra_edge$ docker compose up -d
[+] Running 1/1
```

Container apstra-edge
Started

0.2s



NOTE: IMPORTANT: During the first boot of the Juniper Apstra Cloud Services Edge container, it will perform a one-time registration process using the provided registration code. This process generates a unique secret necessary for authentication between the Edge instance and the CLOUD_TERM service. After this initial registration process, the registration code is invalid. Subsequent starts or restarts of the Juniper Apstra Cloud Services Edge instance use the secret to connect to the CLOUD_TERM service.

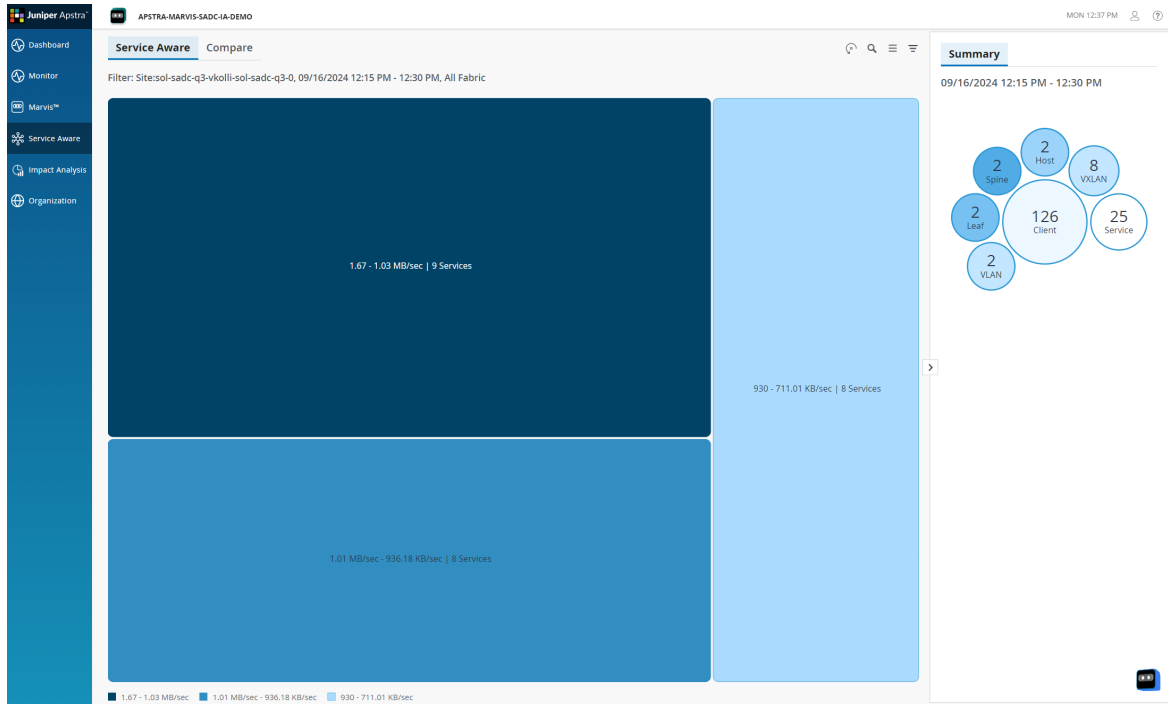
13. From the UI, verify the following statuses:



NOTE: If any of these statuses show **Disconnected**, an error message appears when you hover over the status text.

- Cloud Connectivity is **Connected**: The connection status between the Juniper Apstra Cloud Services Edge container and Juniper Apstra Cloud Services.
- Apstra Connectivity is **Connected**: The connection status of the Juniper Apstra Cloud Services Edge container to the Juniper Apstra Edge instance.

14. Verify that Juniper Apstra Cloud Services is querying the new Edge instance with Apstra Flow. In the ACS UI, a Service Aware and Impact Analysis tab should be visible. Note that these tabs might not appear immediately.



15. (Optional) Verify that the Docker volume was successfully created.

```
root@user:~/apstra_edge# docker volume ls | grep apstra_edge
local      apstra_edge_apstra_edge_store
```

16. (Optional) View event logs.

```
root@user:~/apstra_edge# docker logs --tail 10 -f apstra-edge
```

17. To stop the container, use the following command:

```
root@user:~/apstra_edge# docker compose down
Stopping apstra-edge ... done
Removing apstra-edge ... done
```

18. To restart a stopped Edge container, run the following command:

```
root@user:~/apstra_edge# docker compose up -d
Creating apstra-edge ... done
```

The Juniper Apstra Cloud Services Edge container is initialized, and the Edge instance is running. For additional information about the features and operation of the Apstra Edge Instance, see the Juniper Apstra Cloud Services User Guide.

For post-setup verification, actions, and troubleshooting, see ["Juniper Apstra Cloud Services Edge Post-Setup: Config Changes and Troubleshooting" on page 25](#).

5

CHAPTER

Import a CPU and Memory probe in Juniper Apstra for Apstra Cloud Services

IN THIS CHAPTER

- Import a CPU and Memory Probe in Juniper Apstra for Apstra Cloud Services | 20
-

Import a CPU and Memory Probe in Juniper Apstra for Apstra Cloud Services

SUMMARY

Follow these steps to import a CPU and Memory probe in Juniper Apstra. Note that you must import a probe per blueprint on your Apstra Cloud Services instance for CPU and Memory metrics streaming.

1. From within your blueprint in the Juniper Apstra GUI, navigate to **Analytics > Probes > Create Probe > Import Probes**.



NOTE: Before proceeding, you must create a blueprint if you don't already have one. For more information, see [Create Blueprint](#).

The screenshot shows the Juniper Apstra GUI interface. The breadcrumb navigation at the top indicates the path: Blueprints > zz-darynl-evpn.vqfx_offbox.2485377892354-1085018912 - evpn-vqfx_offbox-virtual > Analytics > Probes. The main navigation bar includes tabs for Dashboard, Analytics, Staged, Uncommitted, Active, and Time Voyager. Below this, there are sub-tabs for Dashboards, Anomalies, Probes, Reports, Root Causes, and Flow Data. The 'Probes' sub-tab is active, displaying a table of existing probes. A 'Create Probe' button is located in the top right corner, with a dropdown menu that includes options: New Probe, Instantiate Predefined Probe, and Import Probes. The 'Import Probes' option is highlighted with a red rectangular box. The table below lists various probes such as Device System Health, Device Telemetry Health, Device Traffic, Drain Traffic Anomaly, ECMP Imbalance (Fabric Interfaces), ESI Imbalance, EVPN VXLAN Type-3 Route Validation, EVPN VXLAN Type-5 Route Validation, LAG Imbalance, and MAC Monitor. Each row shows the probe name, its anomaly status (No anomalies), its state (Disabled or Operational), the user who updated it, and whether it is enabled.

| Name | Anomalies | State | Updated By | Enabled |
|------------------------------------|--------------|-------------|-------------------------|---------|
| Device System Health | No anomalies | Disabled | admin - an hour ago | OFF |
| Device Telemetry Health | No anomalies | Operational | System - an hour ago | ON |
| Device Traffic | No anomalies | Operational | System - an hour ago | ON |
| Drain Traffic Anomaly | No anomalies | Operational | System - 42 minutes ago | ON |
| ECMP Imbalance (Fabric Interfaces) | No anomalies | Operational | System - an hour ago | ON |
| ESI Imbalance | No anomalies | Operational | System - an hour ago | ON |
| EVPN VXLAN Type-3 Route Validation | No anomalies | Operational | admin - 37 minutes ago | ON |
| EVPN VXLAN Type-5 Route Validation | No anomalies | Operational | admin - 38 minutes ago | ON |
| LAG Imbalance | No anomalies | Operational | System - an hour ago | ON |
| MAC Monitor | No anomalies | Operational | admin - 37 minutes ago | ON |

2. Upload a JSON file with the following content and click **Import**.

```
{
  "label": "Stream CPU and Memory Utilization",
  "description": "Probe used to stream cpu and memory utilization from a blueprint in
Apstra to the Cloud",
  "processors": [
    {
      "name": "System cpu utilization data",
      "type": "extensible_data_collector",
      "properties": {
        "execution_count": "-1",
        "service_name": "resource_util",
        "query_expansion": {},
        "service_interval": "10",
        "enable_streaming": true,
        "system_id": "system.system_id",
        "query_tag_filter": {
          "filter": {},
          "operation": "and"
        },
      },
      "graph_query": "node('system', name='system', role=is_in(['leaf', 'access',
'spine', 'superspine']), deploy_mode=is_in(['deploy', 'drain']))",
      "keys": [
        "metric"
      ],
      "query_group_by": [],
      "ingestion_filter": {},
      "data_type": "number",
      "service_input": "",
      "metric": "'system_cpu_utilization'"
    },
    {
      "name": "System memory utilization data",
      "type": "extensible_data_collector",
      "properties": {
        "execution_count": "-1",
```

```

        "service_name": "resource_util",
        "query_expansion": {},
        "service_interval": "10",
        "enable_streaming": true,
        "system_id": "system.system_id",
        "query_tag_filter": {
            "filter": {},
            "operation": "and"
        },
        "graph_query": "node('system', name='system', role=is_in(['leaf', 'access',
'spine', 'superspine']), deploy_mode=is_in(['deploy', 'drain']))",
        "keys": [
            "metric"
        ],
        "query_group_by": [],
        "ingestion_filter": {},
        "data_type": "number",
        "service_input": "",
        "metric": "'system_memory_utilization'"
    },
    "inputs": {},
    "outputs": {
        "out": "System memory utilization data"
    }
}
],
"stages": [
    {
        "description": "System memory utilization percentage",
        "enable_metric_logging": false,
        "retention_size": 0,
        "name": "System memory utilization data",
        "graph_annotation_properties": {},
        "retention_duration": 2592000,
        "units": {
            "value": "%"
        }
    },
    {
        "description": "System cpu utilization percentage",
        "enable_metric_logging": false,
        "retention_size": 0,
        "name": "System cpu utilization data",

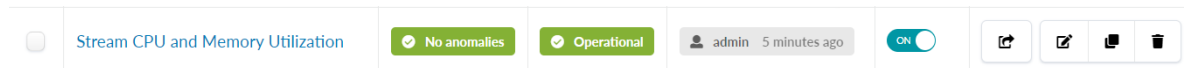
```

```

    "graph_annotation_properties": {},
    "retention_duration": 2592000,
    "units": {
      "value": "%"
    }
  }
]
}

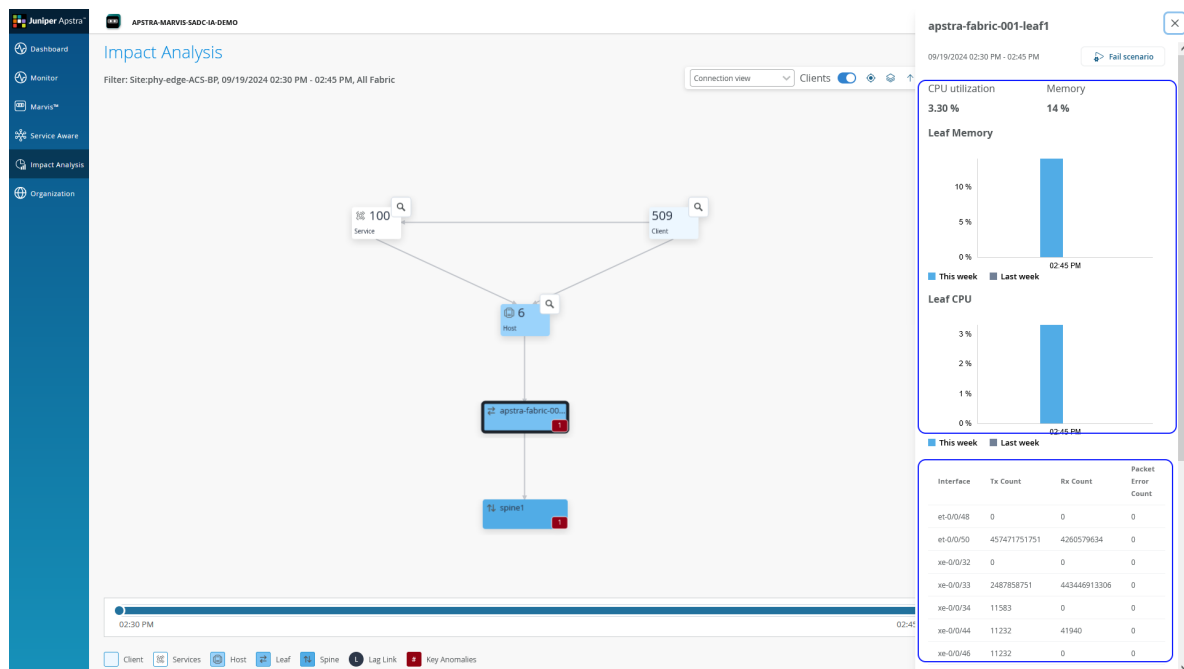
```

The new probe appears in the Probes view.



The probe automatically starts streaming metrics data to the Juniper Apstra Edge. The Edge forwards that information for use by Apstra Cloud Services.

The CPU and memory metrics should now be visible in the Apstra Cloud Services GUI in the Service Aware topology view. Select a topology device to view CPU and memory metrics.



RELATED DOCUMENTATION

[Import Probe](#)

6

CHAPTER

Post-Setup

IN THIS CHAPTER

- [Juniper Apstra Cloud Services Edge Post-Setup: Config Changes and Troubleshooting | 25](#)
-

Juniper Apstra Cloud Services Edge Post-Setup: Config Changes and Troubleshooting

SUMMARY

This document provides information about how the Juniper Apstra Edge for Apstra Cloud Services component handles potential configuration changes, connection failures, and system crashes.

IN THIS SECTION

- [Configuration Changes | 25](#)
- [Connection Failures | 25](#)
- [Juniper Apstra Edge Version Upgrade | 26](#)
- [Handling Direct Modifications | 26](#)
- [Crash Recovery | 26](#)

Configuration Changes

The Juniper Apstra Edge instance collects important information such as Username, Password, and IP Address of the Edge instance during the onboarding process in the Juniper Apstra Cloud Services UI. It stores this data in the cloud database for retrieval through Portal API (PAPI) endpoints. This is crucial for maintaining a stable connection and ensuring that the Edge container can communicate with the Edge instance.

Connection Failures

Connection failures are monitored by the Edge controller, which sends periodic status updates, including connection statuses, to the Juniper Apstra Cloud Services UI. These updates are important for identifying and addressing south-bound (Edge to instance) and north-bound (Edge to Apstra Cloud Services connection failures.

Juniper Apstra Edge Version Upgrade

The Edge container also plays a role in managing version upgrades by retrieving anomalies from the Edge instance upon requests from the alert-collector service. This means that the Edge version must be current to ensure accurate API requests for anomaly data collection.

Handling Direct Modifications

In scenarios where the Juniper Apstra Edge instance credentials are changed without updating them in the UI, the Juniper Apstra Edge container does the following:

- **Detects disruption:** Upon detecting a password change, the Edge container's connection to the Edge instance is disrupted, prompting an immediate status update to the UI.
- **Attempts to reconnect:** The Edge container attempts to re-establish the connection up to three times before halting.

In this instance, manual intervention is required. You must update the new password in the Juniper Apstra Cloud Services UI, triggering a config-push from the `edge-collector` service.

- **Re-establishes connection:** With the new credentials, the Edge container attempts to reconnect, continuously retrying until successful or a new config push occurs.

A successful connection leads to a status update in the UI. A failed connection results in continuous retry attempts until the connection is re-established or a new config push is received.

Crash Recovery

If the Edge instance crashes, a recovery process occurs. Critical Edge information is stored during the startup event and should be backed up outside the host machine. This information is then used in the `docker-compose.yml` file for the Edge container during recovery, allowing for authentication with Juniper Apstra Cloud Services and the continuation of anomaly data forwarding from the Juniper Apstra Edge instance to the cloud.

RELATED DOCUMENTATION

[Juniper Apstra Cloud Services User Guide](#)

7

CHAPTER

Replace the SSL Certificate of Juniper Apstra's Nginx Controller

IN THIS CHAPTER

- [Replace the SSL Certificate of Juniper Apstra's Nginx Controller | 28](#)
-

Replace the SSL Certificate of Juniper Apstra's Nginx Controller

SUMMARY

The Juniper Apstra Edge connects to Juniper Apstra Cloud Services using the management URL provided during the Juniper Apstra Cloud Services onboarding process. This management URL is configured with an IP address, but the default SSL cert configured in Juniper Apstra's Nginx controller only lists DNS:apstra.com as the Subject Alternative Name (SAN). Use these steps to generate a new SSL cert with a SAN containing a Juniper Apstra management IP.

To replace the SSL cert of Juniper Apstra's Nginx controller:

1. Follow the steps in the following link to replace certs in Juniper Apstra [Replace SSL Certificate on Apstra Server with Self-Signed One](#).



NOTE: During Step 3, add an extra SAN name with the desired management IP. The following command creates a cert with a validity of 3 years. Replace -days based on your time requirement.

```
root@user:~# cd /etc/aos/nginx.conf.d/
```

```
root@user:/etc/aos/nginx.conf.d# openssl req -newkey rsa:2048 -nodes -keyout nginx.key -x509 -days 1095 -out nginx.crt -addext extendedKeyUsage=serverAuth -addext subjectAltName=DNS:apstra.com,IP:<IP-address-of-Juniper-Cloud-Services-URL>
```



NOTE: To check the new cert, use the following command:

```
root@user:/etc/aos/nginx.conf.d# openssl x509 -in nginx.crt -text -noout
```


2. Generate a new CA cert.

```
root@user:/etc/aos/nginx.conf.d# openssl x509 -in nginx.crt -inform PEM -outform PEM -out <ip-of-apstra>_ca.cert.pem -days 1095
```

3. Copy the generated CA cert into the host directory where you plan to install the Juniper Apstra Edge container.

```
cp /tmp/<ip-of-apstra>_ca.cert.pem /etc/ssl/certs
```

4. Continue with Step 7 of the Juniper Apstra Edge Container Deployment process.

8

CHAPTER

Internal Variables

IN THIS CHAPTER

- [Internal Variables for Juniper Apstra Cloud Services Edge Configuration | 31](#)
-

Internal Variables for Juniper Apstra Cloud Services Edge Configuration

SUMMARY

This topic lists additional Juniper Apstra Edge for Apstra Cloud Services environment variables meant for internal use for advanced users only. Aside from `REGISTRATION_KEY` and `CLOUD_TERM`, these variables are not required to set up the Apstra Edge instance. These variables should only be reconfigured in specialized situations by expert users.

The following is an example `docker-compose-extended.yml` that lists the Juniper Apstra Edge environment variables.



NOTE: Aside from `REGISTRATION_KEY` and `CLOUD_TERM`, these variables are meant for internal use only. We strongly recommend that you do not alter these variables in production environments in most scenarios.

```
version: '3.0'
volumes:
  apstra_edge_store:
services:
  apstra-edge:
    # Name of the edge container
    container_name: apstra-edge
    # The image to be used for the edge container
    image: svl-artifactory.juniper.net/cdo-docker/aide-jcloud/aos-edge:latest
    # The restart policy for the container
    restart: always
    # pull_policy is set to always to ensure that the latest image is always used
    pull_policy: always
    logging:
      driver: "json-file"
      options:
        max-size: "30m"
        max-file: "10"
```

```

# List of volumes to be mounted to the container
volumes:
  # Allows the container to access the host's SSL certificates
  - /etc/ssl/certs:/etc/ssl/certs
  # Allows the container to access the host's /etc/hosts file
  - /etc/hosts:/etc/hosts
  # Allows apstra-edge to store auth data retrieved from the cloud during registration
  # This volume is used to persist the data across container restarts
  # User must backup this volume to avoid data loss
  - apstra_edge_store:/var/lib/aos-edge
network_mode: "host"
environment:
  # The registration key of the apstra-edge registered in the PAPI/UI
  # mandatory
  - REGISTRATION_KEY=<registration-key-from-papi>
  # The hostname of the cloud endpoint, EPterm
  # mandatory
  - CLOUD_TERM=<hostname-of-epterm>
  # The log level for the edge
  # optional, default is info
  - LOG_LEVEL=debug
  # ORG_ID is the organization ID of the cloud
  # optional, Use when you want to override registration process
  # not recommended in production
  - ORG_ID=<org-id>
  # SECRET is the secret of the apstra-edge registered in the PAPI/UI
  # optional, use when you want to override registration process
  # not recommended in production
  - SECRET=<secret>
  # DEVICE_ID is the device ID of the apstra-edge registered in the PAPI/UI
  # optional, only required when you want to override registration process
  # not recommended in production
  - DEVICE_ID=<device-id>
  # AOS_BLUEPRINT_QUERY_INTERVAL is the interval at which the edge queries the AOS server
for Blueprints
  # Default is 300 seconds
  # optional, only required when you prefer not to use the default interval
  - AOS_BLUEPRINT_QUERY_INTERVAL=<aos-blueprint-query-interval>
  # AOS_PROBE_QUERY_INTERVAL is the interval at which the edge queries the AOS server for
Probes
  # Default is 290 seconds
  # optional, only required when you prefer not to use the default interval
  - AOS_PROBE_QUERY_INTERVAL=<aos-probe-query-interval>

```

```

# EDGE_SERVER_PORT is the port on which the edge local http server listens
# Default is 8081
# optional, only required when you prefer not to use the default port
- EDGE_SERVER_PORT=<edge-server-port>
# AOS_RECEIVER_IP is the IP of the AOS streaming config
# Default value is the IP address of the interface used to connect to the AOS server
# optional, only required when you want to override default value
# not recommended in production
- AOS_RECEIVER_IP=<aos-receiver-ip>
# AOS_RECEIVER_PORT is the port of the AOS streaming config
# optional, only required when you want to override default value, 9595
# not recommended in production
- AOS_RECEIVER_PORT=<aos-receiver-port>
# AOS_URL is the URL of the AOS server
# optional, only required when you want to override fetching from PAPI
# not recommended in production
# Acceptable formats:
# - AOS_URL=https://<hostname>:<port>
# - AOS_URL=https://<hostname>
- AOS_URL=<aos-url>
# AOS_USERNAME is the username of the AOS server
# optional, only required when you want to override fetching from PAPI
# not recommended in production
- AOS_USERNAME=<aos-username>
# AOS_PASSWORD is the password of the AOS server
# optional, only required when you want to override fetching from PAPI
# not recommended in production
- AOS_PASSWORD=<aos-password>
# CLOUD_CA_CERT_PATH is the path to the CA certificate of the cloud
# Path should a path in the mounted volume so aos-edge can access it
# optional, only required when you prefer not to fetch from host system
# not recommended in production
- CLOUD_CA_CERT_PATH=<cloud-ca-cert-path>
# AOS_CA_CERT_PATH is the path to the CA certificate of the AOS server
# Path should a path in the mounted volume so aos-edge can access it
# optional, only required when you prefer not to fetch from host system
# not recommended in production
- AOS_CA_CERT_PATH=<aos-ca-cert-path>
# AOS_INSECURE_SKIP_VERIFY is a flag to skip the verification of the AOS server's
certificate
# optional, only required when you prefer not to verify the AOS server's certificate
# not recommended in production
- AOS_INSECURE_SKIP_VERIFY=<true/false>

```

```

# AUTH_STORE_DIR is the directory where the auth data is stored
# Path should a path in the mounted volume so aos-edge can access it
# Default is /var/lib/aos-edge
# optional, only required when you prefer not to use the default path
# not recommended in production
- AUTH_STORE_DIR=<auth-store-dir>
# AUTH_STORE_FILE_NAME is the name of the file where the auth data is stored
# Default is aos-edge-auth.json in the AUTH_STORE_DIR
# optional, only required when you prefer not to use the default file name
# not recommended in production
- AUTH_STORE_FILE_NAME=<auth-store-file-name>
# DISABLE_PUSH_MODE is a flag to disable the push mode
# optional, only required when you prefer to disable the push mode
# when set to true, the edge will not push Blueprints, Anomalies, Probes etc to the Cloud
# and will only receive/respond to the requests from the Cloud
- DISABLE_PUSH_MODE=<true/false>
# RemoteServer configuration
# Edge can be configured to forward request received from the Cloud to a remote server
# Each RemoteServer configuration require the following set of environment variables
# REMOTE_SERVER_URL_<index> is the URL of the remote server. The index starts from 1
# mandatory
- REMOTE_SERVER_URL_1=<remote-server-url>
# REMOTE_SERVER_USERNAME_<index> is the username of the remote server. The index starts
from 1
# mandatory
- REMOTE_SERVER_USERNAME_1=<remote-server-username>
# REMOTE_SERVER_PASSWORD_<index> is the password of the remote server. The index starts
from 1
# mandatory
- REMOTE_SERVER_PASSWORD_1=<remote-server-password>
# REMOTE_SERVER_NAME_<index> is the name of the remote server. The index starts from 1
# Optional, default is the hostname:port from the URL
- REMOTE_SERVER_NAME_1=<remote-server-name>
# REMOTE_SERVER_TYPE_<index> is the type of the remote server. The index starts from 1
# Optional
- REMOTE_SERVER_TYPE_1=<remote-server-type>
# AOS_RECEIVER_WATCH_INTERVAL is the interval at which the edge watches the AOS streaming
config object
# optional, Default is 30 seconds
- AOS_RECEIVER_WATCH_INTERVAL=<aos-receiver-watch-interval>
# WEBSOCKET_RETRY_INTERVAL is the interval edge waits before retrying connect to the Cloud
via websocket
# optional, Default is 5 seconds

```

```
- WEBSOCKET_RETRY_INTERVAL=<websocket-retry-interval>
# MAX_STREAM_MESSAGE_QUEUE_SIZE is the size of the message queue to receive messages from
the AOS streaming object while forwarding to the Cloud
# optional, Default is 1024
- MAX_STREAM_MESSAGE_QUEUE_SIZE=<max-stream-message-queue-size>
```