Juniper Apstra RedHat OpenShift Integration Guide

IN THIS GUIDE

About This Guide

This guide explains how to use OpenShift 4.17 and Red Hat Ansible Automation Platform 2.5 to automate workflows, streamline decision-making, and activate rulebooks to use for Juniper Apstra event-driven automation (EDA). This document also explains how to install and use Ansible Automation Platform with Automation Decisions and Automation Execution, and set up the environment to optimize the platform's features.

Overview

Juniper Apstra is now integrated with RedHat Ansible Event-Driven Automation (EDA). With this integration, Juniper Apstra supports Kubernetes SR-IOV (Single Root I/O Virtualization) traffic in an automated way. This functionality eliminates the complexities of manual configurations and ensures a more responsive, efficient, and scalable infrastructure.

Benefits

Juniper Apstra's integration with RedHat Ansible EDA provides the following benefits:

- Detects network SR-IOV traffic in Kubernetes clusters events in real time
- Automatically applies configuration updates to ensure optimal network performance
- Eliminates the need for manual intervention, which reduces operational complexity and errors
- Enhances visibility and control over Kubernetes workloads

Before You Start

Prerequisites

Before you can automate workflows, streamline decision-making, and activate rulebooks, you need to make sure you have the following software installed and/or configured:

- OpenShift 4.17 environment set up and configured
- Ansible Automation Platform 2.5 operator installed and configured
- Kubernetes NMState operator installed
- OpenShift SR-IOV Network operator installed
- Docker installed
- Docker registry available to publish images that OpenShift environment can use
- Host with access to the internet running the same architecture as the OpenShift environment
- Juniper Apstra 5.0 or 5.1
- Access to the Juniper public Git repository that contains the automation project files
- (Required) https://github.com/Juniper/eda-apstra-project
- (Optional) https://github.com/Juniper/apstra-ansible-collection

  Use to customize the solution, customize playbooks, and learn how to use modules with Apstra Ansible.

- (Optional) https://github.com/Juniper/k8s.eda

  Explains how to use the Kubernetes event source for Ansible. Also, you can use this repository to run events for other resources.

Notes

- Juniper Apstra EDA only recognizes objects that are labelled with type=eda.
- We require that you set Projects, Credentials, Apstra Blueprint name, and Rulebook Activations to run Juniper Apstra EDA as described in this guide.

RELATED DOCUMENTATION

- Automation Decisions
- Automation Execution Configuration
- Installing NMState Operator
- Installing SR-IOV Network Operator

Download and Installation of Environments

IN THIS SECTION

Follow these steps to download and install your Execution and Decision environments.

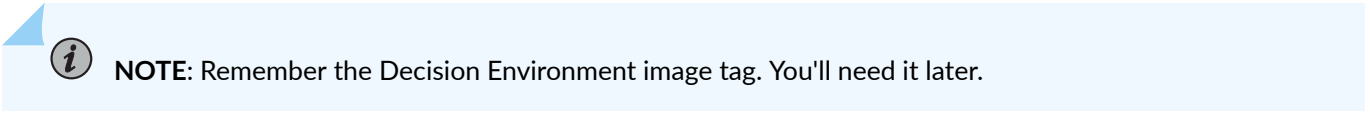
NOTE: Make sure you have the following items available:

- Docker is installed
- Docker registry available to publish images that the OpenShift environment can use

Download and Install the Decision Environment

1. Navigate to https://support.juniper.net/support/downloads/?p=apstra
2. Go to Application Tools.

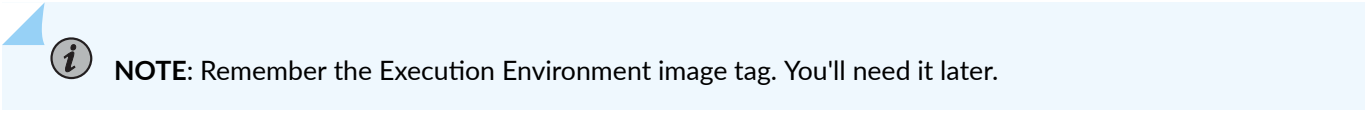
3. Download the image that matches your version of Apstra and the architecture of the server you're using--for example, **juniper-k8s-de-x86_64-1.4.4.image.tgz**
4. **docker load --input juniper-k8s-de-x86_64-1.4.4.image.tgz**
5. **docker tag juniper-k8s-de:latest s-artifactory.juniper.net/atom-docker/de/juniper-k8s-de-x86_64-1.4.4**
6. **docker push s-artifactory.juniper.net/atom-docker/de/juniper-k8s-de-x86_64-1.4.4**



NOTE: Remember the Decision Environment image tag. You'll need it later.

Download and Install the Execution Environment

1. Navigate to https://support.juniper.net/support/downloads/?p=apstra
2. Go to Application Tools.
3. Download the image that matches your version of Apstra and the architecture of the server you're using--for example, **apstra-ee-x86_64-1.0.1.image.tgz**
4. **docker load --input apstra-ee-x86_64-1.0.1.image.tgz**
5. **docker tag apstra-ee:latest s-artifactory.juniper.net/atom-docker/ee/apstra-ee-x86_64-1.0.1**
6. **docker push s-artifactory.juniper.net/atom-docker/ee/apstra-ee-x86_64-1.0.1**



NOTE: Remember the Execution Environment image tag. You'll need it later.

Automation Execution

IN THIS SECTION

Create OpenShift or Kubernetes API Bearer Token Credentials

You can create OpenShift or Kubernetes API Bearer token credential types. These credential types enable you create instance groups that point to a Kubernetes or OpenShift container. You can also use these credentials to access an OpenShift cluster from your automation jobs by using a service account.

For more information about how to create an OpenShift or Kubernetes API Bearer Token credential, see OpenShift or Kubernetes API Bearer Token.

Decision Automation

IN THIS SECTION

Configure SR-IOV Nodes

Enable LLDP on SR-IOV Nodes

Follow these steps to use NMState to enable LLDP on your SR-IOV nodes.

1. Change the interfaces and apply the YAML file below for the NodeNetworkConfigurationPolicy.

```
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: lldp-node-policy
spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ""   # Label the node role worker node if not already
  maxUnavailable: 3
  desiredState:
    interfaces:
      - name: enp4s0f0
        type: ethernet
        lldp:
          enabled: true
      - name: enp4s0f1
```

```
        type: ethernet
        lldp:
          enabled: true
```

2. Issue the following command to check the state of the NodeNetworkState, and you can see that the LLDP neighbors are visible.

```
kubectl get NodeNetworkState <nodeName> -o yaml

```yaml
      lldp:
        enabled: true
        neighbors:
```

Apply SR-IOV Network Node Policy

Create an SR-IOV network node policy to specify the SR-IOV network device configuration. The API object for the policy is part of the sriovnetwork.openshift.io API group.

Here's an example SR-IOV network node policy YAML file:

```
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  labels:
    cluster: rhocpdemo
  name: enp4s0f0-vfs
  namespace: openshift-sriov-network-operator
spec:
  deviceType: netdevice
  isRdma: false
  needVhostNet: true
  nicSelector:
    pfNames:
    - enp4s0f0
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  numVfs: 4
  priority: 99
  resourceName: enp4s0f0_vfs
```
```

Here's another example SR-IOV network node policy YAML file:

```
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodePolicy
metadata:
  labels:
    cluster: rhocpdemo
  name: enp4s0f1-vfs
  namespace: openshift-sriov-network-operator
spec:
  deviceType: netdevice
  isRdma: false
  needVhostNet: true
  nicSelector:
    pfNames:
    - enp4s0f1
  nodeSelector:
    feature.node.kubernetes.io/network-sriov.capable: "true"
  numVfs: 4
  priority: 99
  resourceName: enp4s0f1_vfs
```

See Configuring an SR-IOV Network Device for detailed information on each field in the SR-IOV network node policy.

Mappings of OpenShift Objects with Apstra Objects

The following table highlights what you can expect while creating various OpenShift objects.

Table 1: Mappings of OpenShift Objects with Apstra Objects

OpenShift Object	Apstra Object	Description
Project	Routing Zones (VRF)	Creating/Deleting a project will create Routing Zones (VRF) in Apstra.
SriovNetwork	Virtual Networks(VNET)	Creating/Deleting a SriovNetwork will create Virtual Networks(VNET) in Apstra.

Table 1: Mappings of OpenShift Objects with Apstra Objects *(Continued)*

OpenShift Object	Apstra Object	Description
Pod	Connectivity Template	Creating of VNET creates connectivity template automatically in Apstra. The pod is mapped to the respective nodes and ports in the connectivity templates dynamically.

Ansible Automation Platform

IN THIS SECTION

Ansible Role: apstra-aap-configure

You can use Ansible Role to configure Ansible Automation Controller (Ansible Tower) and Ansible Decisions (event-driven Ansible) for Juniper Apstra EDA.

NOTE: Ansible Role requires Ansible 2.15 or higher.

Role Variables

Variable	Required	Type	Comments
organization_name	yes	String	Name of the organization in Ansible Automation Plaform

(Continued)

Variable	Required	Type	Comments
project_url	yes	String	URL for the project where Playbooks and Rulebooks are available
project_scm_branch	yes	String	SCM branch for the project
apstra_blueprint_name	yes	String	Name of the Apstra blueprint
openshift_host	yes	String	Host address for OpenShift, for example: https://api.ocpapstra-lab.englab.juniper.net
automation_controller_host	yes	String	Ansible host controller URL. Go to operators->Ansible Automation Platform->All Instances-> Automation Controller-> URL
automation_controller_username	yes	String	Ansible host controller Username. Go to operators->Ansible Automation Platform->All Instances-> Automation Controller-> Username
automation_controller_password	yes	String	Ansible host controller URL. Go to operators->Ansible Automation Platform->All Instances-> Automation Controller-> Password
execution_environment_image_url	yes	String	URL where image for Execution environment is pushed
eda_controller_host	yes	String	Ansible EDA controller URL. Go to operators->Ansible Automation Platform->All Instances-> Automation EDA-> URL
eda_controller_username	yes	String	Ansible EDA controller Username. Go to operators->Ansible Automation Platform->All Instances-> Automation EDA-> Username
eda_controller_password	yes	String	Ansible EDA controller Password. Go to operators->Ansible Automation Platform->All Instances-> Automation EDA-> Password
controller_api	yes	String	API endpoint of Ansible controller, for example: https:// aap.apps.ocpapstra-lab.englab.juniper.net/api/controller/"
decision_environment_image_url	yes	String	URL where image for Decision environment is pushed
apstra_api_url	yes	String	URL for the Apstra API
apstra_username	yes	String	Username for Apstra
apstra_password	yes	String	Password for Apstra (sensitive)

NOTE: See Install and configure Ansible Automation Platform to learn how to obtain the Certificate Authority data and API authentication bearer token.

As a best practice, use Ansible Vault to encrypt these files.

Files

Name	Required to Change	Comments
cred_injector_config.json	No	This file requires you to create Apstra credential types in Ansible automation platform.
cred_input_config.json	No	This file requires you to create Apstra credential types in Ansible automation platform.
openshift-ca.crt	yes	Certificate Authority data for OpenShift Cluster.
openshift-sa.crt	yes	API authentication bearer token of Service Account of OpenShift.

NOTE: To learn how to obtain the Certificate Authority data and API authentication bearer token, see Install and configure Ansible Automation Platform.

As a best practice, use Ansible Vault to encrypt these files.

Example Playbook

You can run a playbook to configure Ansible Automation Platform.

Here's an example of how to use this role.

```
---
- name: Configure Ansible Automation Platform for Apstra EDA
  hosts: localhost
  gather_facts: false
  roles:
    - role: apstra-aap-configure
```

Verification and Testing

Follow these steps to verify and test the Juniper Apstra RedHat OpenShift Integration.

1. Validate the decision and execution workflows and rulebook activations by looking at your log files and dashboards in the Automation Controller and Automation Decision software.
2. Validate the projects so they synchronize properly.
3. Run sample YAML files located at https://github.com/Juniper/eda-apstra-project/tree/main/tests and then validate.
4. Create Routing Zones to include in a project in OpenShift.

   Check the project.yaml file to verify that you've created the project with Routing Zones correctly.
5. Verify that the automation job starts, and that the Routing Zone was created in Apstra.
6. Create the SR-IOV Network and verify that it was created correctly by checking the sriov-vn1.yaml file.
7. Verify that the automation job starts and that the Virtual Network was created in Apstra.
8. Verify that the connectivity templates were created.
9. Run the SR-IOV workloads (pod/deployment) on this Virtual Network. See the deployment-vn1.yaml file.
10. Verify that the automation job starts and that the node port is mapped in the Connectivity template.

RELATED DOCUMENTATION

- RedHat Ansible Automation Platform 2.5
- OpenShift Documentation