

EVPN VxLAN Service Automation

References	3
EVPN VxLAN Service Provisioning	3
ATOM EVPN VxLAN Service Model Design	4
Configuration on Layer 3 Switch	4
Creating a EVPN VxLAN Service Definition	5
Deploying EVPN VxLAN Service configurations to devices	7
Working with EVPN VxLAN Service orders	12
Viewing EVPN VxLAN Services	12
Modifying EVPN VxLAN Services	Error! Bookmark not defined.
Deleting EVPN VxLAN Services	13
Additional Resources	13

Intended Audience

This document is intended for Network Administrators & Operators that are using ATOM to perform EVPN VxLAN Service Automation.

References

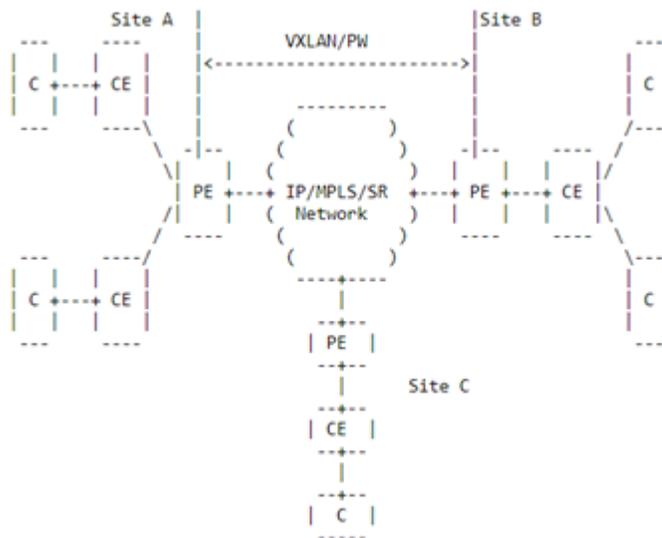
1. [Anuta ATOM Overview](#)
2. [ATOM User Guide](#)
3. ATOM Platform Guide - Discusses Service model, Device model and Workflow development

Please reach out to support@anutanetworks.com for more information on ATOM Documentation or Release Information.

EVPN VxLAN Service Provisioning

Anuta ATOM offers out-of-box support for EVPN VxLAN service provisioning. The model defines service configuration elements that can be used to maintain traffic separation across a layer 2 boundary.

The EVPN VxLAN service model is architected to cater to layer 2 segmentation by stretching the connectivity over an intervening Layer 3 network.



ATOM's EVPN VxLAN Service model offers lifecycle management through an abstracted interface to request, configure, and manage EVPN VxLAN Service components. The configuration of network elements may be done using the CLI or other southbound interfaces

such as NETCONF in conjunction with ATOM's device models based on CLI, Native YANG, or OpenConfig.

ATOM EVPN VxLAN Service Model Design

The EVPN VxLAN service model is structured to allow the user to configure the service from user form. The following section shows the configuration generated by EVPN VxLAN service.

Configuration on Switch

This module describes the configurations pushed to the devices to provision a EVPN VxLAN service.

Interface Provisioning

```
set interfaces irb unit 2105 virtual-gateway-accept-data
set interfaces irb unit 2105 description SDDC_EDU_NAT
set interfaces irb unit 2105 family inet address 172.18.143.228/21 preferred
set interfaces irb unit 2105 family inet address 172.18.143.228/21 virtual-gateway-address
172.18.136.1
```

Protocols Provisioning

```
set protocols igmp interface irb.2105
set protocols evpn vni-options vni 102105 vrf-target target:65001:102105
set protocols evpn extended-vni-list 102105
```

Policy-options Provisioning

```
set policy-options policy-statement OVERLAY_IN term IMPORT_VNI_102105 from community
COMM_VNI_102105
set policy-options policy-statement OVERLAY_IN term IMPORT_VNI_102105 then accept
insert policy-options policy-statement OVERLAY_IN term IMPORT_VNI_102105 before term
IMPORT_LEAF_ESI
set policy-options community COMM_VNI_102105 members target:65001:102105
```

Routing-instances Provisioning

```
set routing-instances OVERLAY_DC interface irb.2105
```

Vlan Provisioning

```
set vlans SDDC_EDU_NAT vlan-id 2105
set vlans SDDC_EDU_NAT l3-interface irb.2105
set vlans SDDC_EDU_NAT vxlan vni 102105
set vlans SDDC_EDU_NAT vxlan ingress-node-replication
```

Creating a EVPN VxLAN Service Definition

Use this procedure to create a EVPN VxLAN configuration on a Layer 3 switch.

Field	Action
Vlan-Id (Mandatory)	Enter an ID for VLAN Creation Note: Note: Range 1 through 4096
Description (Mandatory)	Enter a brief description that should appear on the interface Note: The field accepts only a string value.
Resource-Pool (Optional)	Select Resource Pool from the drop down to be used for the service creation
Create-Vrf	Select this field if a new VRF is to be created on the device. This will then require adding RD and RT values
Rd	Enter a Route Distinguisher that helps distinguish routes from others Note: The field accepts a numerical value or nn:nn format.
Rt	Enter a Route Target that will be used to import/export routes to the VRF Note: The field accepts a numerical value or nn:nn format.
Vrf (Mandatory)	Enter a name for the Virtual Routing Function
Cidr (Mandatory)	Select IP address pool name from the list
Virtual-Gateway-Address (Mandatory)	Enter a valid IP address



Create Evpn_vxlan:evpn-Vx-Lan



• -mandatory information



Vlan-Id •

Enter a VLAN ID



Description •

Provide interface description



Resource-Pool

Select site name / resource pool from the list

Create-Vrf

Create a VRF

Vrf •

Name of the existing VRF if the create-vrf is false (or) Name of the VRF to create if the ...

Cidr •

Select IP address pool name from the list

Virtual-Gateway-Address •

Must be a valid IP Address. Ex :172.16.1.24.

Owner •

Shared-With



Deploying EVPN VxLAN Service configurations to devices

To provision an EVPN VxLAN using Anuta ATOM perform the below actions:

1. On the ATOM instance navigate to **Services > Catalog** tab
2. Click on the Evpn-vx-lan to navigate to the **EVPN VxLAN** instances
3. From the Evpn-vx-lan instances Click on the + to create a new Evpn-vx-lan service
4. In the Create Evpn_vxlan:evpn-Vx-Lan form perform the following actions:
 - a. Enter the Vlan-Id for the Service creation
 - b. Provide a **Description** for the vlan. The vlan will be updated with a description
 - c. Select a Resource-Pool from the drop down
 - d. Select the check-box of Create-Vrf if a new VRF needs to be created
 - i. Enter an RD value to be used for VRF routes
 - ii. Enter an RT value that will be used in the VRF to import/export routes
 - e. Enter a **Vrf** name. A new VRF will be created on the device if option d is selected. If the VRF already exists, the same will be utilized.
 - f. In the Cider, Select IP address pool name from the list that will be used for the service provisioning
 - g. Provide the device Virtual-Gateway-IP on which the service will be provisioned.

Note: You can add only one device at a time

- h. Click on the  in the top right corner to start the device provisioning process.
- i. In the Tasks pane, check for **Create: evpn_vxlan**, and navigate to **Commands** to view the configurations..

The screenshot below shows the fields populated to provision an EVPN VxLAN service for a new VRF.



Edit Null



• -mandatory information



Cidr •

Select IP address pool name from the list

evpn_ipam



Create-Vrf

Create a VRF



Description •

Provide interface description

evpn-test



Rd •

Enter a RD

112:100



Resource-Pool

Select site name / resource pool from the list

rp



Rt •

Enter a RT

112:100



Virtual-Gateway-Address •

Must be a valid IP Address. Ex :172.16.1.24.

172.16.4.168



Vlan-Id •

Enter a VLAN ID

112



Vrf •

Name of the existing VRF if the create-vrf is false (or) Name of the VRF to create if the ...

evpn-test



The screenshot below shows the commands generated and pushed to the devices to provision a EVPN VxLAN.

Create: evpn_vxlan:evpn-vx-lan evpn-test,112 ✕

Task ID Jz63mpaIpjTTqY3qfm5CmeNg

User Name toshiba

Time Taken 01/04/2021, 15:01:46 - 01/04/2021, 15:01:59 (12 seconds)

Logs Summary **Commands**

Operation JunosQfxConfRoot:configuration

Device vQFX-RE-leafswitch_5.106 / 172.16.5.106

Name/IP

Status PROVISIONED

Commands

```
<configuration xmlns="http://yang.juniper.net/junos-qfx/conf/root"
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"> <!-- filler -->
<interfaces xmlns="http://yang.juniper.net/junos-qfx/conf/interfaces"> <!-- filler -->
  <!-- subtree:UPDATE /controller:devices/device=vQFX_172.16.5.106/atom-junos-
mount:juniperqfx181R1/junos-qfx-conf-root:configuration/junos-qfx-conf-interfaces:interfaces/interface=irb -->
  <!-- Last updated: 2021-04-01 09:31:47.07 -->
  <interface>
    <name>irb</name>
    <unit nc:operation="create">
      <name>112</name>
      <virtual-gateway-accept-data />
      <description>evpn-test</description>
      <family>
        <inet>
          <address>
            <name>172.16.2.1/24</name>
```

Create: evpn_vxlan:evpn-vx-lan evpn-test,112



Logs

Summary

Commands

```
<preferred />
<virtual-gateway-address>172.16.2.4</virtual-gateway-address>
</address>
</inet>
</family>
</unit>
</interface>
</interfaces> <!-- filler -->
<policy-options xmlns="http://yang.juniper.net/junos-qfx/conf/policy-options"> <!-- filler -->
  <!-- subtree:CREATE /controller:devices/device=vQFX_172.16.5.106/atom-junos-
mount:juniperqfx181R1/junos-qfx-conf-root:configuration/junos-qfx-conf-policy-options:policy-
options/community=COMM_VNI_10112 -->
  <!-- Last updated: 2021-04-01 09:31:47.236 -->
  <community nc:operation="create">
    <name>COMM_VNI_10112</name>
    <members>target:65001:10112</members>
  </community>
  <!-- subtree:CREATE /controller:devices/device=vQFX_172.16.5.106/atom-junos-
mount:juniperqfx181R1/junos-qfx-conf-root:configuration/junos-qfx-conf-policy-options:policy-options/policy-
statement=OVERLAY_IN -->
  <!-- Last updated: 2021-04-01 09:31:47.614 -->
  <policy-statement nc:operation="create">
    <name>OVERLAY_IN</name>
    <term>
      <name>IMPORT_VNI_10112</name>
```

```
name="10112" /name>
<then>
  <accept />
</then>
<from>
  <community>COMM_VNI_10112</community>
</from>
</term>
</policy-statement>
</policy-options> <!-- filler -->
<protocols xmlns="http://yang.juniper.net/junos-qfx/conf/protocols"> <!-- filler -->
  <!-- subtree:UPDATE /controller:devices/device=vQFX_172.16.5.106/atom-junos-
mount:juniperqfx181R1/junos-qfx-conf-root:configuration/junos-qfx-conf-protocols:protocols/evpn -->
  <!-- Last updated: 2021-04-01 09:31:46.941 -->
  <evpn>
    <extended-vni-list>10112</extended-vni-list>
    <vni-options> <!-- filler -->
      <!-- subtree:CREATE /controller:devices/device=vQFX_172.16.5.106/atom-junos-
mount:juniperqfx181R1/junos-qfx-conf-root:configuration/junos-qfx-conf-protocols:protocols/evpn/vni-
options/vni=10112 -->
      <!-- Last updated: 2021-04-01 09:31:47.518 -->
      <vni nc:operation="create">
        <name>10112</name>
        <vrf-target>
          <community>target:65001:10112</community>
        </vrf-target>
      </vni-operation>
    </vni-options>
  </evpn>
</protocols>
```

```
<community>target:112:100</community>
</vrf-target>
<route-distinguisher>
  <rd-type>112:100</rd-type>
</route-distinguisher>
<interface>
  <name>irb.112</name>
</interface>
</instance>
</routing-instances> <!-- filler -->
<vlangs xmlns="http://yang.juniper.net/junos-qfx/conf/vlangs"> <!-- filler -->
  <!-- subtree:CREATE /controller:devices/device=vQFX_172.16.5.106/atom-junos-
mount:juniperqfx181R1/junos-qfx-conf-root:configuration/junos-qfx-conf-vlangs:vlangs/vlan=evpn-test -->
  <!-- Last updated: 2021-04-01 09:31:46.825 -->
  <vlan nc:operation="create">
    <name>evpn-test</name>
    <vlan-id>112</vlan-id>
    <l3-interface>irb.112</l3-interface>
    <vxlan>
      <vni>10112</vni>
      <ingress-node-replication />
    </vxlan>
  </vlan>
</vlangs> <!-- filler -->
</configuration> <!-- filler -->
```

Working with EVPN VxLAN Service orders

Viewing EVPN VxLAN Services

1. Once the EVPN VxLAN services are created, the service instances can be seen under **Services > Catalog > Evpn-vx-lan**
2. Click on the Service instance to view the Service Details. An example is shown below.

Vlan-Id	Description	Resource-Pool	Create-Vif	Rfd	Rt	Vif	Cidr	Virtual-Gateway-Address
112	evpn-test	rp	<input checked="" type="checkbox"/>	112:100	112:100	evpn-test	evpn_ipam	172.16.4.168

Deleting EVPN VxLAN Services

1. To delete the EVPN VxLAN services created, navigate to **Services > Catalog > Evpn-vx-lan**
2. Check on the Service instance to be deleted and choose the delete icon.
3. The Task pane will capture the deletion progress.

Additional Resources

For detailed information on Anuta ATOM and its rich set of features, please refer to the [resources](#) section on anutanetworks.com