JUNIPER NETWORKS | Engineering Simplicity

In Focus

Contrail Networking: How to Onboard a Fabric and Create an Overlay

IN THIS GUIDE

- About This In Focus Use Case | 1
- Onboard the Fabric | 4
- Create the Overlay Networks | 19
- Set Up a PNF Service Chain | 32
- Summary | 38

About This In Focus Use Case

Use Case	Use Contrail Command to onboard a fabric and create overlay networks.
Audience	Enterprise data center network administrator
Knowledge Level	General familiarity with data center architectures and overlay and underlay routing
Benefits	• Save time by using an intent-based UI to configure your data center network. • Reduce risk of error and misconfiguration through automation. • Use an industry-leading SDN solution to configure your overlay networks.

Products Used	• Contrail Networking Release 2003
	• QFX Series switches running Junos OS
	  ◦ tested on QFX10002 running Junos OS 18.4R2-S3 as a spine switch
	  ◦ tested on QFX5110 running Junos OS 18.4R2-S3 as a leaf switch
	• SRX Series Services Gateway running Junos OS
	  ◦ tested on SRX5400 running Junos OS 18.2R2-S3
	
	NOTE: Although this use case has been specifically validated against the hardware devices and Junos OS versions shown, you can choose to use any device and Junos OS version supported by this release of Contrail Networking as long as the devices and software support their assigned roles.

This use case demonstrates how you can onboard a data center fabric and build a simple overlay in minutes and is intended for the network administrator who is responsible for the data center network but not for the endpoint servers and compute devices. This use case therefore does not demonstrate the compute orchestration capabilities of Contrail Networking such as how servers and VMs are instantiated nor does it depend on any particular type or brand of compute orchestrator.

To get your data center network up and running, you first have to onboard the fabric, which means that you have to configure all your data center switches and routers to be part of the same IP network. Next, you create the separate overlay segmented networks that govern which endpoints are allowed to communicate with which endpoints. This underlay and overlay provisioning requires a considerable amount of configuration on each device, ranging from basic system turn-up to routing protocol configuration to interface provisioning. Depending on the size and complexity of your data center, this can easily run into dozens of CLI commands per device and take hours or days to complete.

With Contrail Command, you can get your data center network up and running in a fraction of this time while minimizing the risk of misconfiguration. Contrail Command provides an intent-based user interface that translates your high level intent into configuration commands that the devices understand. You provide basic configuration parameters for the overall fabric and specify the role that you want each device to play, and Contrail Command then performs the underlying configuration for you. Not only does this save you time, but it reduces the chance of error as you don't have to provision each device individually and risk creating mismatched configurations across devices.

Figure 1 shows the fabric that you are onboarding in this use case. The fabric consists of a set of QFX Series switches connected in a spine-and-leaf architecture with the Contrail Networking installation running on a regular compute device attached to a leaf switch. The switches themselves are connected to each other but have no configuration and are in zeroized (greenfield) state. All devices including the Contrail Networking installation are connected to an out-of-band management network over their management ports. The only prerequisite is that you've assigned an IP address to the Contrail Networking port that connects to the management LAN.

Figure 1: Fabric Underlay

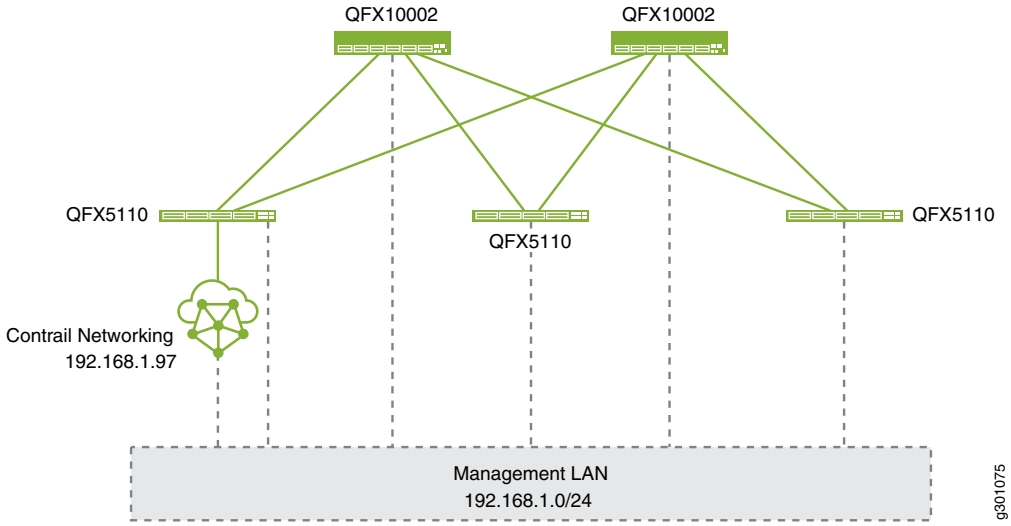
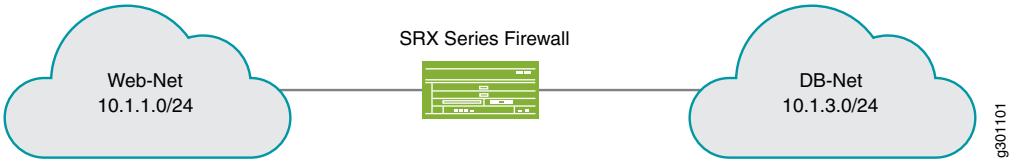


Figure 2 shows the overlay segmented networks you are creating in this use case. These represent nominally a web front-end network and a back-end database network in this fictitious example. These networks are separated by a firewall that ensures only legitimate database requests from qualified internal front-end servers are allowed to pass through. Configuration of the firewall is outside the scope of Contrail Networking and outside the scope of this use case.

Figure 2: Overlay Segmented Networks



This use case is sufficiently generic that you can apply the same principles to onboard fabrics and create overlays much more complex than that shown. Although this guide shows specific models of QFX Series switches, this use case works equally well with other QFX Series switches as long as the switches support the functions required by the roles they are given. For example, this use case creates a centrally-routed overlay architecture that requires the spine switches to support VXLAN routing, which is where the switch has the ability to decapsulate the VXLAN header and route traffic based on the inner overlay IP address. Refer to *Contrail Networking Supported Hardware Platforms and Associated Roles And Node Profiles* to see what other switches you can use in this scenario.

Moreover, this use case does not specify or presuppose any particular hardware module or interface on the switches. To make this use case work, you will need to ensure that the ports connecting neighboring switches have compatible optics so that their physical layers can come up, but this use case can work with different interface modules and speeds. In other words, this use case focuses purely on using Contrail Command to set up a fabric and overlay and leaves best practices on data center architectures to other documentation.

Onboard the Fabric

SUMMARY

In this section, you are using Contrail Command to configure your zeroized fabric devices to be part of a single physical underlay network.

IN THIS SECTION

- Explanation of Procedure | 4
- Configure the Device YAML File | 5
- Create Fabric | 9
- Device Discovery | 11
- Assign the Roles | 11
- Autoconfigure | 12
- Assign Telemetry Profiles | 14
- Verify Underlay Configuration on the Device (Optional) | 14

Explanation of Procedure

The recommended first step to onboarding a fabric is to tell Contrail Networking about the devices in the fabric. You do this by creating a YAML file containing the list of serial numbers of all devices in the fabric. For convenience, Contrail Command provides you with a template that you can use.

You then upload this file to Contrail Networking when you run the Create Fabric wizard, which guides you through fabric creation, including performing zero-touch-provisioning (ztp) of the devices in the list.

In a regular network, you typically onboard the fabric once at the beginning and whenever you add or remove devices from the fabric. In this latter case, you would be onboarding specific devices onto the fabric rather than creating a fabric from scratch.

> **NOTE:** All devices should be in zeroized state (**request system zeroize** from the CLI) before you perform the initial onboarding.

Configure the Device YAML File

The device YAML file is the file that contains the list of serial numbers of all fabric devices. The device YAML file also contains configuration that you want Contrail Networking to apply as part of onboarding the fabric. You create the device YAML file on the computer that you use to connect to Contrail Command. This can be your local computer as long as it has access to the management network where the Contrail Networking installation resides.

The YAML file must comply with the syntax specified in //yaml.org including spacing and indentations, which act as delimiters. By convention, a YAML file has a **.yaml** or **.yml** file extension.

For convenience, you can download a YAML file template from Contrail Command when you use the Create Fabric wizard (see step 4 in "Create Fabric" on page 9).

A very basic device YAML file just contains the chassis serial numbers of the devices and looks like this:

> **NOTE:** To get the chassis serial number from your device, issue the **show chassis hardware** command from the Junos CLI on the device. Alternatively, you can get the chassis serial number from the label affixed to your device. See *Locating the Serial Number on a QFX10000 Switch or Component* and *Locating the Serial Number on a QFX5100 Device or Component*.

```
device_to_ztp:
  - serial_number: '111111111111'
  - serial_number: ’222222222222’
  - serial_number: '333333333333’
  - serial_number: '444444444444'
  - serial_number: '555555555555'
```

It's a good idea, however, to include the hostnames of the devices. Otherwise, Contrail Command displays and refers to the devices by their serial numbers. The example below adds meaningful hostnames to identify the spine and leaf switches.

```
device_to_ztp:
  - serial_number: '111111111111'
    hostname: 'DC1-Access-Leaf-1'
  - serial_number: ’222222222222’
    hostname: 'DC1-Access-Leaf-2'
  - serial_number: '333333333333’
    hostname: 'DC1-Access-Leaf-3'
  - serial_number: '444444444444'
    hostname: 'DC1-Border-Spine-1'
  - serial_number: '555555555555'
    hostname: 'DC1-Border-Spine-2'
```

That's all you really need for the device YAML file. However, the YAML file in this use case contains a few more constructs to demonstrate its flexibility and power and to save you some additional configuration later on.

The YAML file used in this example is shown below:

```
supplemental_day_0_cfg:
  - name: '10k-spine-cfg'
    cfg: |
      set system location rack 10
  - name: '5kleaf-cfg'
    cfg: |
      set system location rack 5
device_to_ztp:
  - serial_number: '111111111111'
    supplemental_day_0_cfg: '5kleaf-cfg'
    hostname: 'DC1-Access-Leaf-1'
    device_functional_group: 'L2-Server-Leaf'
  - serial_number: ’222222222222’
    supplemental_day_0_cfg: '5kleaf-cfg'
    hostname: 'DC1-Access-Leaf-2'
    device_functional_group: 'L2-Server-Leaf'
  - serial_number: '333333333333’
    supplemental_day_0_cfg: '5kleaf-cfg'
    hostname: 'DC1-Access-Leaf-3'
    device_functional_group: 'L2-Server-Leaf'
  - serial_number: '444444444444'
    supplemental_day_0_cfg: '10k-spine-cfg'
    hostname: 'DC1-Border-Spine-1'
    device_functional_group: 'Centrally-Routed-Border-Spine'
  - serial_number: '555555555555'
    supplemental_day_0_cfg: '10k-spine-cfg'
    hostname: 'DC1-Border-Spine-2'
    device_functional_group: 'Centrally-Routed-Border-Spine'
```

Table 1 contains the definitions of the constructs used in this YAML file and Table 2 shows the resulting actions on each fabric device when you later apply this YAML file to the fabric.

Table 1: Device YAML File Definitions

Fields		Meaning
device_to_ztp		This mandatory section contains the fabric devices that you want to discover and configure. Each fabric device is organized in its own subsection. Each subsection is identified by the device’s serial number.

Table 1: Device YAML File Definitions *(continued)*

Fields		Meaning
	serial_number	This is the hardware serial number of the device. To see the serial number in Junos, issue the **show chassis hardware** command from the Junos CLI on the device. Copy the chassis serial number from the command output and paste it into this YAML file. This field is mandatory.
	supplemental_day_0_cfg	This field references the **supplemental_day_0_cfg** section within this file. This field is optional.
	hostname	This field sets the hostname on the device. This is equivalent to the Junos **set system host-name** command. This field is optional but highly recommended. If you do not specify the hostname, Contrail Command displays the device using its serial number.

	device_functional_group	This field applies the specified device functional group to the device. A device functional group is a user-defined node profile that automatically sets the roles for the device.
		
		This field is optional. If you do not specify a device functional group, you will need to specify the roles for the device explicitly when you create the fabric.
		
		To define the device functional groups used in this example, log in to Contrail Command and select **INFRASTRUCTURE>Fabrics** and then click on **Device Functional Groups**. Click **Create** to create the groups, as follows:
		
		• L2-Server-Leaf
		  ◦ physical role = leaf
		  ◦ routing roles = CRB-Access, AR-Client
		• Centrally-Routed-Border-Spine
		  ◦ physical role = spine
		  ◦ routing roles = Route-Reflector, CRB-Gateway, DC-Gateway, DCI-Gateway, PNF-Servicechain
		
		For more information on roles, see the *Contrail Networking Fabric Lifecycle Management Guide*.

Table 1: Device YAML File Definitions *(continued)*

Fields		Meaning
supplemental_day_0_cfg		This optional section contains additional configuration that you can apply to the discovered devices. The additional configuration is organized into subsections identified by name.
	name	This is the name of the additional configuration subsection. Devices in the **device_to_ztp** section refer to this subsection by this name.
	cfg	This contains the set of Junos CLI commands you want to execute on the device as part of device discovery.

Table 2: Device YAML File Resulting Actions

Device	Actions
111111111111	Sets the hostname to **DC1-Access-Leaf-1**. Sets the device functional group to **L2-Server-Leaf**, which sets the following: • physical role = leaf • routing/bridging roles = CRB-Access, AR-Client Issues the following CLI commands on the device: `set system location rack 5`
222222222222	Sets the hostname to **DC1-Access-Leaf-2**. Sets the device functional group to **L2-Server-Leaf**, which sets the following: • physical role = leaf • routing/bridging roles = CRB-Access, AR-Client Issues the following CLI commands on the device: `set system location rack 5`
333333333333	Sets the hostname to **DC1-Access-Leaf-3**. Sets the device functional group to **L2-Server-Leaf**, which sets the following: • physical role = leaf • routing/bridging roles = CRB-Access, AR-Client Issues the following CLI commands on the device: `set system location rack 5`

Table 2: Device YAML File Resulting Actions *(continued)*

Device	Actions
444444444444	Sets the hostname to **DC1-Border-Spine-1**. Sets the device functional group to **Centrally-Routed-Border-Spine**, which sets the following: • physical role = spine • routing/bridging roles = Route-Reflector, CRB-Gateway, DC-Gateway, DCI-Gateway, PNF-Servicechain Issues the following CLI commands on the device: `set system location rack 10`
555555555555	Sets the hostname to **DC1-Border-Spine-2**. Sets the device functional group to **Centrally-Routed-Border-Spine**, which sets the following: • physical role = spine • routing/bridging roles = Route-Reflector, CRB-Gateway, DC-Gateway, DCI-Gateway, PNF-Servicechain Issues the following CLI commands on the device: `set system location rack 10`

Once you have finished creating the YAML file and the device functional groups, you can proceed to the next step where you launch the Create Fabric wizard to start creating the fabric.

Create Fabric

This is the first stage of the Create Fabric wizard. You specify the basic underlay and overlay configuration parameters and you reference the device YAML file you created earlier.

1. Log in to Contrail Command.

2. Select **INFRASTRUCTURE>Fabrics** to bring up the Fabrics page and click **Create** to create the fabric.

   The **Select provisioning option** window appears.

3. Select **New Fabric** and click **Provision**.

   This launches the **Create Fabric** wizard for a new (greenfield) fabric.

4. Fill in the fields on this page as shown in Table 3.

Table 3: Fields in the Create Fabric Page

Fields	Meaning	Setting in this Example
Name	The name of the fabric.	DC1-Fabric
Device credentials	The root password that you want to set for all devices in the fabric. **NOTE:** Type the password carefully. Contrail Command does not ask you to verify or confirm the password that you type in.	*<password>*
Overlay ASN (iBGP)	The autonomous system number for the overlay iBGP network. All devices in the overlay belong to the same autonomous system.	65000
Device Info	The device YAML file. **NOTE:** To download the YAML file template, click the Download icon.	Upload the YAML file you created on your local computer.
Underlay ASNs (eBGP)	Specify the ASN range that you want to assign to the underlay devices. Contrail Networking assigns each device with its own AS number in the underlay.	65001 to 65099
Management subnets	Specify the management subnet and gateway for the underlay devices. Contrail Networking discovers all devices connected to the management subnet.	cidr: 192.168.1.0/24 gateway: 192.168.1.1
Fabric subnets	Specify the fabric subnet for the underlay. All fabric ports are on this subnet. Fabric ports are the main traffic-carrying ports on the devices.	192.168.11.0/24
Loopback subnets	Specify the loopback subnet. The loopback address is used by the overlay BGP.	192.168.111.0/24
PNF Servicechain subnets	Specify the physical network function (PNF) service chain subnets. These are the subnets for the PNF devices that you want to add to the service chain. You need a PNF service chain if you want to allow communications between devices on the different segregated networks that you later create.	192.168.100.0/24

5. Click **Next** to launch the device discovery process.

Device Discovery

This is the second stage of the Create Fabric wizard. Contrail Networking begins the discovery process by responding to DHCP requests on the management subnet and then using NETCONF to configure the devices that match the serial numbers listed in the YAML file, including:

- reading interface information, enabling LLDP, and bringing up interfaces
- applying the standard juniper-qfx10k and juniper-qfx5k node profiles
- assigning loopback IP addresses
- running the supplementary commands from the YAML file
- reading and storing inventory and other information from the devices
- building the topology

At the completion of this stage, Contrail Networking has brought up all the fabric devices and has built the fabric topology.

Click **Next** to progress to role assignment.

Assign the Roles

This is the third stage of the Create Fabric wizard. At this point, Contrail Networking has discovered and performed initial configuration of the fabric devices and knows how they are connected together but does not yet know what roles the devices play in the fabric.

When you assign a role to a device, you are telling Contrail Networking what the device's function is. A role is strictly a Contrail Networking construct and is an example of Contrail Networking's intent-based provisioning. It is not a Junos parameter that you configure on a switch. Instead, you tell Contrail Networking what role you want a device to assume, and Contrail Networking then performs all the underlying configuration on the device to enact that role.

There are two types of roles that you assign to a device. The physical (underlay) role describes whether a device is a spine device or a leaf device. The routing (overlay) role describes the overlay routing functions that a device supports. While a device can only have one physical role, it is common and normal for a device to have multiple routing roles.

The **Assign to devices** window shows the list of discovered devices along with their assigned roles. In this use case, you assigned the roles using the **device_functional_group** parameter in the device YAML file so you don't have to explicitly assign roles here.

Verify that the roles are as shown in Table 4 and Table 5. For more information on roles, see the *Contrail Networking Fabric Lifecycle Management Guide*.

Table 4: Roles Assigned to Leaf Switches

	Meaning	Setting in this Example
Role	The physical or underlay role of the device.	leaf

Table 4: Roles Assigned to Leaf Switches *(continued)*

	Meaning	Setting in this Example
Routing Roles	The routing/bridging or overlay role of the device.	CRB-Access
		AR-Client

Table 5: Roles Assigned to Spine Switches

	Meaning	Setting in this Example
Role	The physical or underlay role of the device.	spine
Routing Roles	The routing/bridging or overlay role of the device.	CRB-Gateway
		DC-Gateway
		DCI-Gateway
		PNF-Servicechain
		Route-Reflector

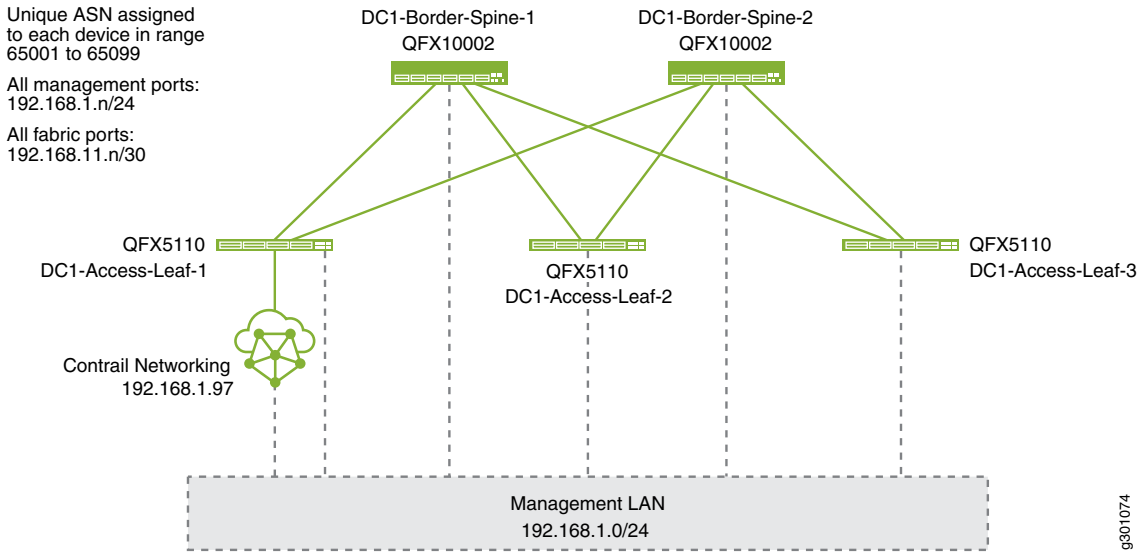
Click **Autoconfigure** to start the auto-configuration stage of the process.

Autoconfigure

This is the fourth stage of the Create Fabric wizard. Contrail Networking configures all the necessary parameters on the devices to support the roles they are given.

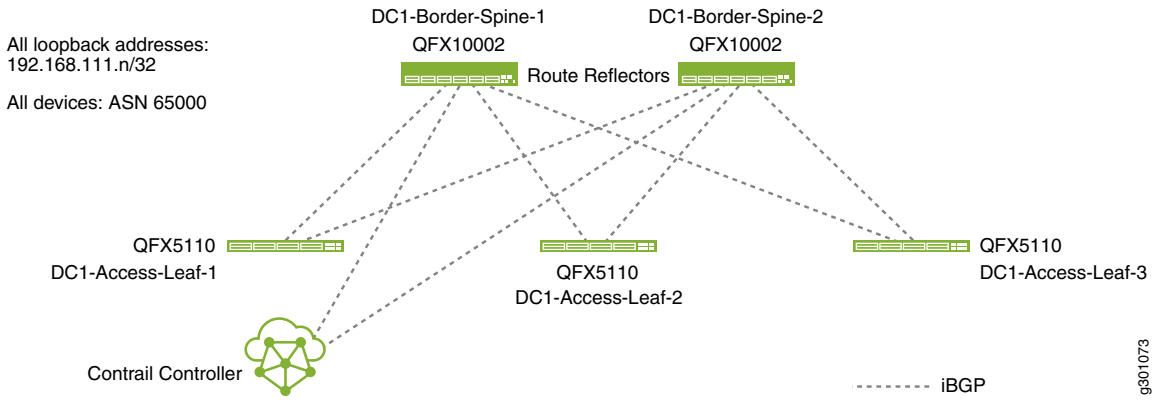
After auto-configuration is complete, the fabric is now up and running and ready for overlay configuration (Figure 3).

Figure 3: Underlay After Configuration



Additionally, the initial overlay BGP configuration is performed, identifying all overlay BGP speakers (including the Contrail Networking controller) and assigning the common overlay AS number (Figure 4).

Figure 4: Initial Overlay Configuration



Click **Next** to move to the next stage.

Assign Telemetry Profiles

This is the optional fifth and final stage of the Create Fabric wizard. You can assign telemetry profiles to any or all of the fabric devices in this stage. The Contrail Networking installation can include an sflow server for telemetry processing. When you assign a profile to a device, you are configuring the device to send telemetry data to this sflow server. For information on assigning telemetry profiles, see the *Contrail Networking Fabric Lifecycle Management Guide*.

This use case does not use telemetry profiles. Click **Finish** to exit the Create Fabric wizard.

Verify Underlay Configuration on the Device (Optional)

With just a few clicks, Contrail Networking has onboarded your fabric. If you want to see how Contrail Networking has configured your devices, you can log in to the individual devices to check.

1. Log in to DC1-Access-Leaf-1 using your favorite SSH client.
    a. Check out the interfaces configuration:

```
root@DC1-Access-Leaf-1> show interfaces brief
...
Physical interface: et-0/0/48, Enabled, Physical link is Up
  Link-level type: Ethernet, MTU: 9192, LAN-PHY mode, Speed: 40Gbps,
  Loopback: Disabled, Source filtering: Disabled, Flow control: Disabled,
  Media type: Fiber
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None

  Logical interface et-0/0/48.0
    Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
    inet  192.168.11.13/30

Physical interface: et-0/0/49, Enabled, Physical link is Up
  Link-level type: Ethernet, MTU: 9192, LAN-PHY mode, Speed: 40Gbps,
  Loopback: Disabled, Source filtering: Disabled, Flow control: Disabled,
  Media type: Fiber
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None

  Logical interface et-0/0/49.0
    Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
    inet  192.168.11.1/30
 ...
```

```
Physical interface: lo0, Enabled, Physical link is Up
  Type: Loopback, Link-level type: Unspecified, MTU: Unlimited,
  Clocking: Unspecified, Speed: Unspecified
  Device flags   : Present Running Loopback
  Interface flags: SNMP-Traps

  Logical interface lo0.0
    Flags: SNMP-Traps Encapsulation: Unspecified
    inet  192.168.111.248  --> 0/0
...

Physical interface: vme, Enabled, Physical link is Up
  Type: Mgmt-VLAN, Link-level type: Mgmt-VLAN, MTU: 1514, Clocking: Unspecified,
  Speed: Unspecified
  Device flags   : Present Running
  Interface flags: SNMP-Traps

  Logical interface vme.0
    Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
    inet  192.168.1.183/24
```

You can see the fabric interfaces (192.168.11.13 and 192.168.11.1) that connect from this leaf switch to the two spine switches, the loopback interface (192.168.111.248), and the management interface (192.168.1.183).

b. Check out the BGP configuration:

```
root@DC1-Access-Leaf-1> show bgp group
Group Type: External                              Local AS: 65005
  Name: IPCLOS_eBGP     Index: 0                  Flags: root@DC1-Access-Leaf-1> show
bgp group
Group Type: External                              Local AS: 65005
  Name: IPCLOS_eBGP     Index: 0                  Flags: <Export Eval>
  Export: [ IPCLOS_BGP_EXP ]
  Options: <LocalAS>
  Options: <VpnApplyExport>
  Holdtime: 0 Local AS: 65005 Local System AS: 65000
  Total peers: 2        Established: 2
  192.168.11.14+179
  192.168.11.2+61540
  inet.0: 10/18/18/0

Group Type: Internal    AS: 65000                 Local AS: 65000
  Name: _contrail_asn-65000 Index: 1              Flags: <Export Eval>
  Export: [ _contrail_ibgp_export_policy ]
  Options: <LocalAS>
  Options: <VpnApplyExport>
  Holdtime: 0 Local AS: 65000 Local System AS: 65000
```

```
  Total peers: 2        Established: 2
  192.168.111.251+179
  192.168.111.252+62234
  __default_evpn__.evpn.0: 0/0/0/0
  bgp.evpn.0: 0/0/0/0
  bgp.rtarget.0: 0/2/2/0
  default-switch.evpn.0: 0/0/0/0

Groups: 2  Peers: 4    External: 2    Internal: 2    Down peers: 0   Flaps: 1
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.rtarget.0
                       2          0          0          0          0          0
bgp.evpn.0
                       0          0          0          0          0          0
inet.0
                      18         10          0          0          0          0
default-switch.evpn.0
                       0          0          0          0          0          0
__default_evpn__.evpn.0
                       0          0          0          0          0          0
```

You can see that Contrail Networking has assigned ASN 65005 for the eBGP underlay and ASN 65000 for the iBGP overlay. The eBGP underlay has two peers, which are the two spine switches (identified by their fabric interface addresses and their AS numbers). The iBGP overlay also has two peers, which are the route reflectors on the two spine switches (identified by their loopback addresses).

2. Log in to DC1-Border-Spine-1 using your favorite SSH client.

   a. Check out the interfaces configuration:

```
root@DC1-Border-Spine-1> show interfaces brief
Physical interface: et-0/0/0, Enabled, Physical link is Up
  Link-level type: Ethernet, MTU: 9192, LAN-PHY mode, Speed: 40Gbps,
  Loopback: Disabled, Source filtering: Disabled, Flow control: Disabled,
  Media type: Fiber
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None

  Logical interface et-0/0/0.0
    Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
    inet  192.168.11.14/30
...
Physical interface: et-0/0/1, Enabled, Physical link is Up
  Link-level type: Ethernet, MTU: 9192, LAN-PHY mode, Speed: 40Gbps,
  Loopback: Disabled, Source filtering: Disabled, Flow control: Disabled,
```

```
  Media type: Fiber
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None

  Logical interface et-0/0/1.0
    Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
    inet  192.168.11.18/30
...
Physical interface: et-0/0/2, Enabled, Physical link is Up
  Link-level type: Ethernet, MTU: 9192, LAN-PHY mode, Speed: 40Gbps,
  Loopback: Disabled, Source filtering: Disabled, Flow control: Disabled,
  Media type: Fiber
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None

  Logical interface et-0/0/2.0
    Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
    inet  192.168.11.22/30
...
Physical interface: em0, Enabled, Physical link is Up
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Clocking: Unspecified,
  Speed: 1000mbps
  Device flags   : Present Running
  Interface flags: SNMP-Traps

  Logical interface em0.0
    Flags: Up SNMP-Traps 0x4000000 Encapsulation: ENET2
    inet  192.168.1.13/24
...
Physical interface: lo0, Enabled, Physical link is Up
  Type: Loopback, Link-level type: Unspecified, MTU: Unlimited,
  Clocking: Unspecified, Speed: Unspecified
  Device flags   : Present Running Loopback
  Interface flags: SNMP-Traps

  Logical interface lo0.0
    Flags: SNMP-Traps 0x4000 Encapsulation: Unspecified
    inet  192.168.111.251  --> 0/0
    inet6 fe80::82ac:ac0f:fc91:c400
```

You can see the fabric interfaces (192.168.11.14, 192.168.11.18, and 192.168.11.22) that connect from this spine switch to the three leaf switches, the loopback interface (192.168.111.251), and the management interface (192.168.1.13).

b. Check out the BGP configuration:

```
root@DC1-Border-Spine-1> show bgp group
Group Type: External                                Local AS: 65002
  Name: IPCLOS_eBGP     Index: 0                    Flags: <Export Eval>
  Export: [ IPCLOS_BGP_EXP ]
  Options: <LocalAS>
  Options: <VpnApplyExport>
  Holdtime: 0 Local AS: 65002 Local System AS: 65000
  Total peers: 3        Established: 3
  192.168.11.13+55479
  192.168.11.17+54145
  192.168.11.21+55618
  inet.0: 9/21/21/0

Group Type: Internal    AS: 65000                   Local AS: 65000
  Name: _contrail_asn-65000 Index: 1                Flags: <Export Eval>
  Export: [ _contrail_ibgp_export_policy ]
  Options: <Cluster LocalAS>
  Options: <VpnApplyExport>
  Holdtime: 0 Local AS: 65000 Local System AS: 65000
  Total peers: 3        Established: 3
  192.168.111.248+54191
  192.168.111.249+60777
  192.168.111.250+60518
  __default_evpn__.evpn.0: 0/0/0/0
  bgp.evpn.0: 0/0/0/0
  bgp.rtarget.0: 0/3/3/0
  default-switch.evpn.0: 0/0/0/0

Group Type: External                                Local AS: 65000
  Name: _contrail_asn-65000-external Index: 2       Flags: <Export Eval>
  Options: <Multihop LocalAS>
  Options: <VpnApplyExport>
  Holdtime: 0 Local AS: 65000 Local System AS: 65000
  Total peers: 1        Established: 0
  10.1.0.2

Group Type: Internal    AS: 65000                   Local AS: 65000
  Name: _contrail_asn-65000-rr Index: 3             Flags: <Export Eval>
  Export: [ _contrail_ibgp_export_policy ]
  Options: <LocalAS>
  Options: <VpnApplyExport>
  Holdtime: 0 Local AS: 65000 Local System AS: 65000
  Total peers: 1        Established: 1
  192.168.111.252+58935
  __default_evpn__.evpn.0: 0/0/0/0
  bgp.evpn.0: 0/0/0/0
  bgp.rtarget.0: 0/1/1/0
  default-switch.evpn.0: 0/0/0/0
```

```
Groups: 4  Peers: 8    External: 4    Internal: 4    Down peers: 1   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.rtarget.0
                       4          0          0          0          0          0
bgp.evpn.0
                       0          0          0          0          0          0
default-switch.evpn.0
                       0          0          0          0          0          0
__default_evpn__.evpn.0
                       0          0          0          0          0          0
inet.0
                      21          9          0          0          0          0
```

You can see that Contrail Networking has assigned ASN 65002 for the eBGP underlay and ASN 65000 for the iBGP overlay. The eBGP underlay has three peers, which are the three leaf switches (identified by their fabric interface addresses and their AS numbers). The iBGP overlay also has three peers, which are the three leaf switches (identified by their loopback addresses).

Create the Overlay Networks

SUMMARY

In this section, you are creating the overlay segmented networks on top of the physical fabric underlay.

IN THIS SECTION

- Explanation of Procedure | **19**
- Create Virtual Networks | **24**
- Create Logical Routers | **25**
- Create Virtual Port Groups | **27**
- Verify Routing Tables on the Spine Switch (Optional) | **29**

Explanation of Procedure

With the fabric onboarded, you can route traffic from any device to any device, which is what you want for the physical network. You are now ready to create the logical networks that ride on top of this physical network. These are the overlay segmented networks that dictate which endpoint device can communicate with which other endpoint device. For example,

you may want to create a segmented network for your finance department, another for your regular employee intranet, and yet another for your third-party contractors.

Creating overlay segmented networks, however, can be confusing especially if you're working with CLIs that don't explicitly distinguish between underlay and overlay parameters. Running the same routing protocols in both the underlay and the overlay, as is common in many data centers, can certainly add to the confusion.

Although you still need to know the difference between the underlay and the overlay, Contrail Command provides you with a simple user interface to create overlay segmented networks without requiring you to work with the corresponding underlay and overlay CLI commands. As with the underlay fabric, you work with the overlay segmented network as a whole rather than with each individual switch.

In order to minimize confusion between underlay (physical) and overlay (logical) terminology, Contrail Networking uses different terms for the overlay (Table 6).

NOTE: This document refers to a segmented network as a segregated network in general and not necessarily as a single LAN segment or VLAN. In other words, a segmented network can consist of multiple VLANs and multiple subnets. A segmented network can only communicate with another segmented network through a security policy.

NOTE: Contrail Networking was designed to serve both the enterprise (single-tenant) and service provider (multi-tenant) markets. To a new user of Contrail Networking, this flexibility can sometimes be confusing. To minimize confusion, this table explains the terminology in the context of a single-tenant network. The application of these concepts to multi-tenant networks may differ. In other words, Contrail Networking is a tool that allows you to abstract your network in different ways. How you abstract an enterprise network may be different from how you abstract a service provider network.

Table 6: Overlay Definitions for a Single-Tenant Network

Overlay Term	Meaning
Virtual network	A virtual network is an overlay bridged network. An endpoint device attached to a virtual network has layer 2 reachability to any other endpoint device attached to the same virtual network regardless of which physical (underlay) LAN the device resides on. Typically, a virtual network has a single subnet, but Contrail Networking allows you to define a virtual network with multiple subnets, which is akin to having a physical LAN or VLAN supporting secondary IP addressing.

Table 6: Overlay Definitions for a Single-Tenant Network *(continued)*

Overlay Term	Meaning
Logical router	A logical router is analogous to a VRF or routing instance and performs routing for a single segmented network. A segmented network consists of one or more virtual networks. In other words, a logical router can bridge and route within and between the virtual networks it is connected to. It cannot route to unconnected virtual networks, which are other segmented networks. You will learn how to route between segmented networks in a later step. The segmented network can have a single subnet or multiple subnets depending on whether you configure the logical router to route for one or more virtual networks. The same logical router is instantiated on every physical device that acts as a router in the overlay. In a centrally-routed model, these are the spine switches. In an edge-routed model, these are the leaf switches.
Virtual port group	A virtual port group allows you to multi-home your BMS or VM endpoints and consists of one or more switch ports grouped together and presented to the virtual network as a single entity, similar to a LAG or MC-LAG. For consistency, Contrail Networking requires an endpoint BMS or VM to attach to the virtual network through a virtual port group even if the attachment point is a single port, in which case the virtual port group consists of a single member.

Figure 5 shows the relationship between logical routers and virtual networks for a single-tenant network. A physical router can have multiple logical routers. Each logical router contains routes for a single segmented network, which is represented by one or more virtual networks. Each virtual network has a single subnet (typically) and attaches to endpoint devices through virtual port groups.

Figure 5: Logical Routers and Virtual Networks in a Single-Tenant Network

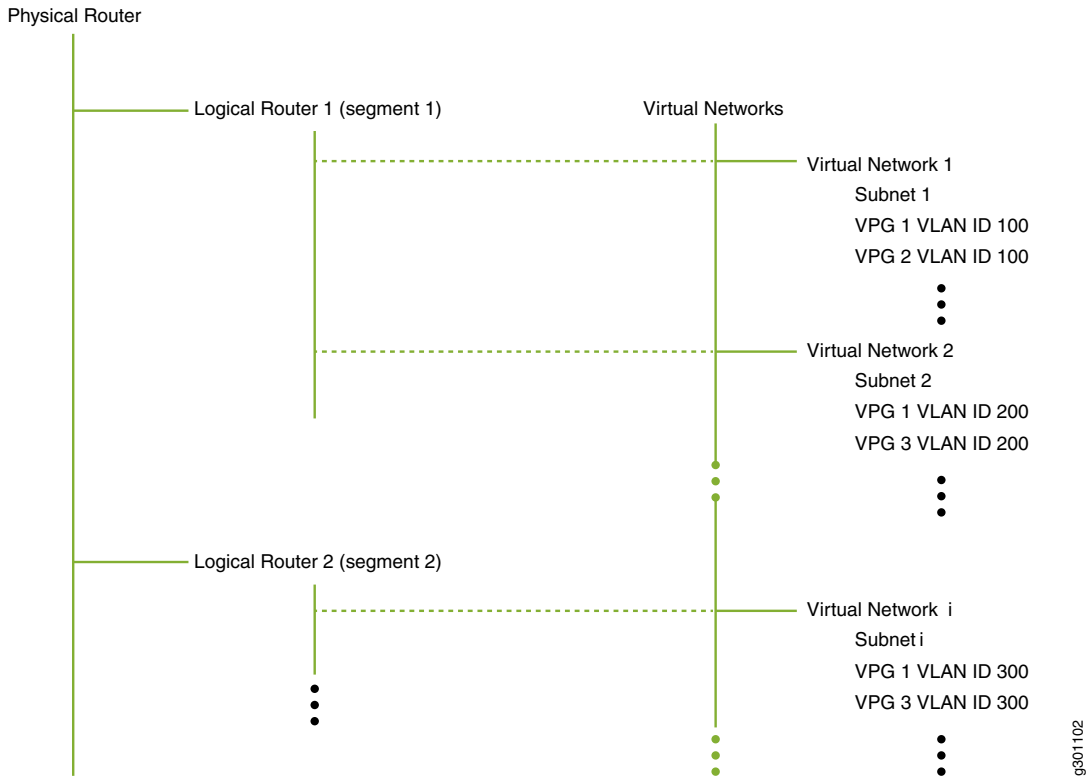
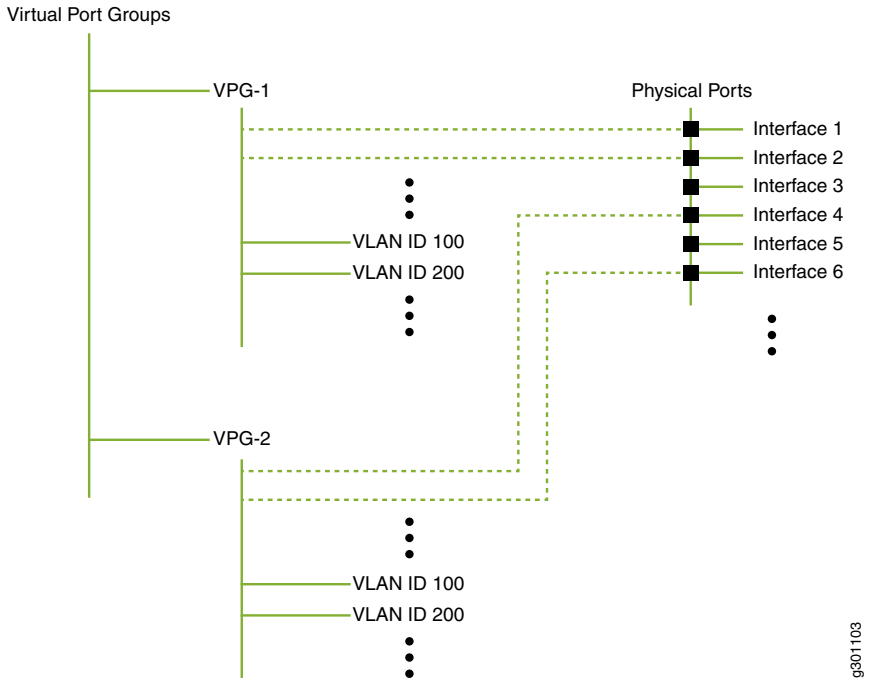


Figure 6 shows the relationship between virtual port groups and physical interfaces. A virtual port group consists of one or more associations with physical interfaces and can be defined with VLANs.

Figure 6: Virtual Port Groups



A number of observations can be made from the relationships shown for a single-tenant network:

- Endpoint devices (attached to a virtual port group) in virtual network 1 can freely communicate with endpoint devices in virtual network 2 because both of these virtual networks are connected to the same logical router. However, endpoint devices in virtual network 1 cannot communicate with endpoint devices in virtual network *i* unless you explicitly allow it because these virtual networks are connected to different logical routers.
- While a logical router can connect to multiple virtual networks, the reverse is not true. Just as a LAN can only belong to a single VRF, a virtual network can only connect to a single logical router. In the hierarchy above, since virtual network 1 is connected to logical router 1, it cannot also be connected to logical router 2.
- Furthermore, Contrail Networking allows you to define the scope of your VLANs and IP addresses. For enterprises, it is common for your VLANs and IP addresses to have enterprise-wide significance. For example, all devices using VLAN ID 100 are on the same VLAN even if they attach to different virtual port groups, and all IP addresses are unique even if they belong to different logical routers (routing instances).

Finally, when you create the overlay segmented networks, you have the option of defining a centrally-routed architecture or an edge-routed architecture:

- In a centrally-routed architecture, the spine switches perform the VXLAN routing. All user traffic is encapsulated and sent through VXLAN tunnels that terminate on the spine switches. The spine switches decapsulate and route packets based on the inner overlay IP header.
- In an edge-routed architecture, the leaf switches perform the VXLAN routing. The leaf switches decapsulate and route packets based on the inner overlay IP header.

To illustrate these concepts, this use case creates two overlay segmented networks and uses a centrally-routed model where each spine switch performs the routing. To do this, you define two virtual networks, one for each segmented

network. Next, you define the logical routers that connect to these virtual networks and instantiate the logical routers onto the physical spine switches. Each spine switch therefore has two logical routers, with each logical router responsible for its own segmented network. You then create virtual port groups that you attach to the virtual networks.

Create Virtual Networks

1. Select **OVERLAY>Virtual Networks** and click **Create**.

   The **Create Virtual Network** window appears.

2. Create the first virtual network (Web-Net) using the settings in Table 7. Leave all other settings at their default values.

   Table 7: Settings for Web-Net

Field	Meaning	Setting in this Example
Name	The name that you want to call this virtual network.	Web-Net
Subnets>Network IPAM	The IP address allocation instance to use.	default-domain:default:default-project:default-network-ipam This is the default IPAM instance.
Subnets>CIDR	The virtual network subnets.	10.1.1.0/24

   Click **Create**.

3. Create the second virtual network (DB-Net) using the settings in Table 8. Leave all other settings at their default values.

   Table 8: Settings for DB-Net

Field	Meaning	Setting in this Example
Name	The name that you want to call this virtual network.	DB-Net
Subnets>Network IPAM	The IP address allocation instance to use.	default-domain:default:default-project:default-network-ipam This is the default IPAM instance. By using the same IPAM instance for both networks, all IP addresses have global scope within the enterprise.
Subnets>CIDR	The virtual network subnets.	10.1.3.0/24

   Click **Create**.

4. Optionally, go to **MONITORING>Jobs** to bring up the Jobs page and click on a job to see details on the configuration being pushed to the device for that job.

You have now created the two virtual networks, each with a single subnet (Figure 7). These virtual networks are not associated with any physical devices yet.

Figure 7: Virtual Networks

Create Logical Routers

1. Select **OVERLAY>Logical Routers** to bring up the Logical Routers page and click **Create**.

   The **Create Logical Router** window appears.

2. Create the first logical router (Web-LR) using the settings in Table 9. Leave all other settings at their default values.

   Table 9: Settings for Web-LR

Field	Meaning	Setting in this Example
Name	The name you want to call this logical router.	Web-LR
Logical Router Type	The type of routing you want this logical router to perform.	VXLAN Routing
Choose Fabric		DC1-Fabric
Connected networks	The virtual network you want to connect to this logical router.	Web-Net
Extend to Physical Router	The actual physical routers where you want to instantiate this logical router.	DC1-Border-Spine-1 DC1-Border-Spine-2

   Click **Create**.

3. Create the second logical router (DB-LR) using the settings in Table 10. Leave all other settings at their default values..

   Table 10: Settings for DB-LR

Field	Meaning	Setting in this Example
Name	The name you want to call this logical router.	DB-LR

Table 10: Settings for DB-LR *(continued)*

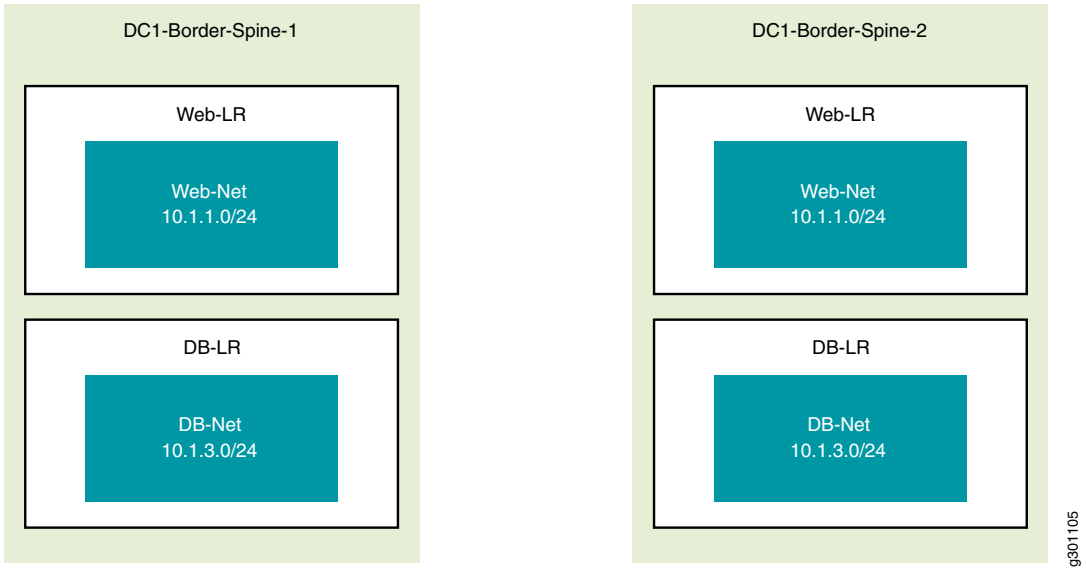
Field	Meaning	Setting in this Example
Logical Router Type	The type of routing you want this logical router to perform.	VXLAN Routing
Choose Fabric		DC1-Fabric
Connected networks	The virtual network you want to connect to this logical router.	DB-Net
Extend to Physical Router	The actual physical routers where you want to instantiate this logical router.	DC1-Border-Spine-1 DC1-Border-Spine-2

Click **Create**.

4. Optionally, go to **MONITORING>Jobs** to bring up the Jobs page and click on a job to see details on the configuration being pushed to the device for that job.

You have now created two logical routers and instantiated them on both border spine routers. You have also associated each logical router with one of the virtual networks you created in the previous procedure. Figure 8 shows logical router Web-LR instantiated on both spine switches and providing routing for the Web-Net virtual network and logical router DB-LR instantiated on both spine switches and providing routing for the DB-Net virtual network.

Figure 8: Logical Routers



Create Virtual Port Groups

A virtual port group is a network entity (not a server entity) and is the sole means by which you attach a port to a virtual network. It provides a consistent attachment point regardless of whether you are attaching a single port or a port group. Members of the group can be from the same switch or from across switches such as in a multi-homed configuration.

1. Select **OVERLAY>Virtual Port Group** to bring up the Virtual Port Group page and click **Create**.

   The **Create Virtual Port Group** page appears.

2. Configure the first virtual port group (VPG-1-Web) using the settings in Table 11. Leave all other settings at their default values.

   Table 11: Settings for VPG-1-Web

Field	Meaning	Setting in this Example
Virtual Port Group Name	The name you want to call this virtual port group.	VPG-1-Web
Fabric name	The fabric where this virtual port group resides.	DC1-Fabric
Physical Interface	The interface members of this virtual port group. The **Available Physical Interface** panel lists all the physical interfaces in the network that are available for assignment. Search for the interfaces using the case-insensitive search box and use the “>” at the end of the row to assign an interface to the group.	xe-0/0/2 on DC1-Access-Leaf-1 xe-0/0/2 on DC1-Access-Leaf-2
VLAN>Network	The virtual network to which you want to attach the virtual port group.	Web-Net
VLAN>VLAN ID	The VLAN ID to use for the specified virtual network. If this is the first time you are attaching any virtual port group to the specified virtual network, you need to specify the VLAN ID you want use. Each subsequent time you attach any other virtual port group to this same virtual network, Contrail Networking automatically assigns the same VLAN ID.	11

   Click **Create**.

3. Configure the second virtual port group (VPG-3-DB) using the settings in Table 12. Leave all other settings at their default values.

Table 12: Settings for VPG-3-DB

Field	Meaning	Setting in this Example
Virtual Port Group Name	The name you want to call this virtual port group.	VPG-3-DB
Fabric name	The fabric where this virtual port group resides.	DC1-Fabric
Physical Interface	The physical interfaces that you want to assign to this virtual port group. The **Available Physical Interface** panel lists all the physical interfaces in the network that are available for assignment. Search for the interfaces using the case-insensitive search box and use the “>” at the end of the row to assign an interface to the group.	xe-0/0/3 on DC1-Access-Leaf-2 xe-0/0/3 on DC1-Access-Leaf-3
Network	The virtual network to which you want to attach the virtual port group.	DB-Net
VLAN ID	The VLAN ID to use for the specified virtual network. If this is the first time you are attaching any virtual port group to the specified virtual network, you need to specify the VLAN ID you want use. Each subsequent time you attach any other virtual port group to this same virtual network, Contrail Networking automatically assigns the same VLAN ID.	13

Click **Create**.

4. Optionally, go to **MONITORING>Jobs** to bring up the Jobs page and click on a job to see details on the configuration being pushed to the device for that job.

5. Optionally, ping the 10.1.1.1 gateway from your 10.1.1.0/24 device and ping the 10.1.3.1 gateway from your 10.1.3.0/24 device.

You have now created two virtual port groups, one connecting to each virtual network.

Figure 9 and Figure 10 show the VPGs from the physical and logical perspective respectively. For context, these figures show the endpoint servers (compute devices), which have been configured outside of this use case.

Figure 9: Virtual Port Groups Physical View

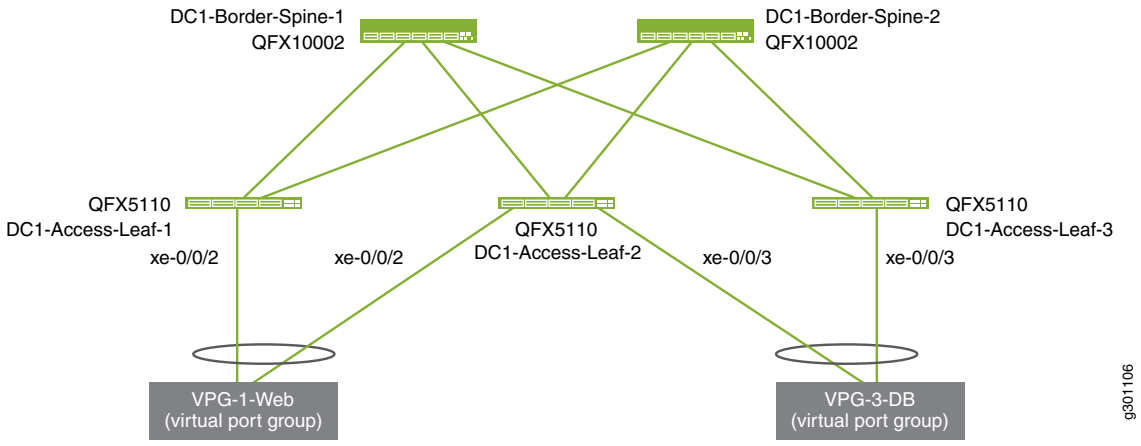
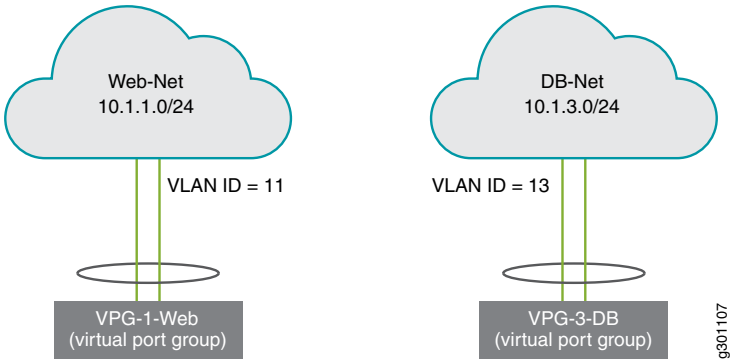


Figure 10: Virtual Port Groups Logical View



Now that you have created both segmented networks, endpoints on each segmented network have full connectivity within their respective network, but no connectivity to endpoints on the other segmented network. If that meets your needs, then you are done. If, however, you want to allow communication between the two segmented networks, then you need to set up a PNF service chain, which is the next step in this use case.

Verify Routing Tables on the Spine Switch (Optional)

1. Log in to DC1-Border-Spine-1 using your favorite SSH client.

2. Look at the routing table summaries.

```
root@DC1-Border-Spine-1> show route forwarding-table
...
Routing table: __contrail_Web-LR_2c1aca57-439b-4d62-a390-ff1847076c8f.inet
```

```
Internet:
Enabled protocols: Bridging, All VLANs,
Destination        Type RtRef Next hop           Type Index    NhRef Netif
default            perm     0                    rjct     1762     1
0.0.0.0/32         perm     0                    dscd     1756     2
10.1.1.0/24        intf     0                    rslv     1776     1 irb.11
10.1.1.0/32        dest     0 10.1.1.0           recv     1774     1 irb.11
10.1.1.1/32        intf     0 10.1.1.1           locl     1775     2
10.1.1.1/32        dest     0 10.1.1.1           locl     1775     2
10.1.1.4/32        intf     0 10.1.1.4           locl     1777     2
10.1.1.4/32        dest     0 10.1.1.4           locl     1777     2
10.1.1.5/32        dest     0 80:ac:ac:91:e4:0   ucst     1779     1 vtep.32769
10.1.1.255/32      dest     0 10.1.1.255         bcst     1773     1 irb.11
127.0.0.1/32       intf     0 127.0.0.1          locl     1778     1
172.16.0.73/32     user     0                    comp     1742     2
172.16.0.74/32     user     0                    dscd     1756     2
224.0.0.0/4        perm     0                    mdsc     1761     1
224.0.0.1/32       perm     0 224.0.0.1          mcst     1764     1
255.255.255.255/32 perm     0                    bcst     1765     1

Routing table: __contrail_DB-LR_ff58562c-8107-4a58-993b-02dff3579d47.inet
Internet:
Enabled protocols: Bridging, All VLANs,
Destination        Type RtRef Next hop           Type Index    NhRef Netif
default            perm     0                    rjct     1797     1
0.0.0.0/32         perm     0                    dscd     1790     2
10.1.3.0/24        intf     0                    rslv     1811     1 irb.12
10.1.3.0/32        dest     0 10.1.3.0           recv     1809     1 irb.12
10.1.3.1/32        intf     0 10.1.3.1           locl     1810     2
10.1.3.1/32        dest     0 10.1.3.1           locl     1810     2
10.1.3.4/32        intf     0 10.1.3.4           locl     1812     2
10.1.3.4/32        dest     0 10.1.3.4           locl     1812     2
10.1.3.5/32        dest     0 80:ac:ac:91:e4:0   ucst     1817     1 vtep.32769
10.1.3.255/32      dest     0 10.1.3.255         bcst     1808     1 irb.12
127.0.0.1/32       intf     0 127.0.0.1          locl     1813     1
172.16.0.73/32     user     0                    comp     1783     2
172.16.0.74/32     user     0                    dscd     1790     2
224.0.0.0/4        perm     0                    mdsc     1796     1
224.0.0.1/32       perm     0 224.0.0.1          mcst     1799     1
255.255.255.255/32 perm     0                    bcst     1800     1
```

You can see a routing table for each overlay segmented network, with routes for 10.1.1.0/24 and 10.1.3.0/24 directed out irb.11 and irb.12 respectively.

3. Look at the individual overlay routing tables:

```
root@DC1-Border-Spine-1> show route table
__contrail_Web-LR_2c1aca57-439b-4d62-a390-ff1847076c8f.inet.0

__contrail_Web-LR_2c1aca57-439b-4d62-a390-ff1847076c8f.inet.0: 6 destinations, 7 routes (5
active, 0 holddown, 1 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both

10.1.1.0/24        *[Direct/0] 00:11:16
                    >  via irb.11
                    [EVPN/170] 00:11:16
                       to 192.168.11.13 via et-0/0/0.0
                    >  to 192.168.11.17 via et-0/0/1.0
                       to 192.168.11.21 via et-0/0/2.0
10.1.1.1/32        *[Local/0] 00:11:16
                       Local via irb.11
10.1.1.4/32        *[Local/0] 00:11:16
                       Local via irb.11
172.16.0.73/32     *[EVPN/170] 00:11:16
                       to 192.168.11.13 via et-0/0/0.0
                    >  to 192.168.11.17 via et-0/0/1.0
                       to 192.168.11.21 via et-0/0/2.0
172.16.0.74/32     *[Static/5] 00:11:16
                       Discard
```

```
root@DC1-Border-Spine-1> show route table
__contrail_DB-LR_ff58562c-8107-4a58-993b-02dff3579d47.inet.0

__contrail_DB-LR_ff58562c-8107-4a58-993b-02dff3579d47.inet.0: 6 destinations, 7 routes (5
active, 0 holddown, 1 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both

10.1.3.0/24        *[Direct/0] 00:09:20
                    >  via irb.12
                    [EVPN/170] 00:09:19
                       to 192.168.11.13 via et-0/0/0.0
                    >  to 192.168.11.17 via et-0/0/1.0
                       to 192.168.11.21 via et-0/0/2.0
10.1.3.1/32        *[Local/0] 00:09:20
                       Local via irb.12
10.1.3.4/32        *[Local/0] 00:09:20
                       Local via irb.12
172.16.0.73/32     *[EVPN/170] 00:09:19
                       to 192.168.11.13 via et-0/0/0.0
```

```
                    >  to 192.168.11.17 via et-0/0/1.0
                       to 192.168.11.21 via et-0/0/2.0
172.16.0.74/32     *[Static/5] 00:09:20
                       Discard
```

You can see how the overlay routes (10.1.1.0/24 and 10.1.3.0/24) map onto the underlay (192.168.11.13, 192.168.11.17, 192.168.11.21).

Set Up a PNF Service Chain

SUMMARY

In this section, you are creating a path between the two overlay segmented networks by connecting them through an SRX firewall. You insert the firewall by setting up a service chain that includes the firewall.

IN THIS SECTION

- Explanation of Procedure | 32
- Onboard the PNF Device | 33
- Create PNF Service Chain | 34

Explanation of Procedure

A physical network function (PNF) service chain inserts a PNF device such as an SRX firewall between two segmented networks. The overlay you created previously consists of two segmented networks that have no logical connectivity to each other. When you add a PNF service chain, you are effectively connecting the two segmented networks through a firewall that enforces policies to govern what traffic can pass from one network to the other.

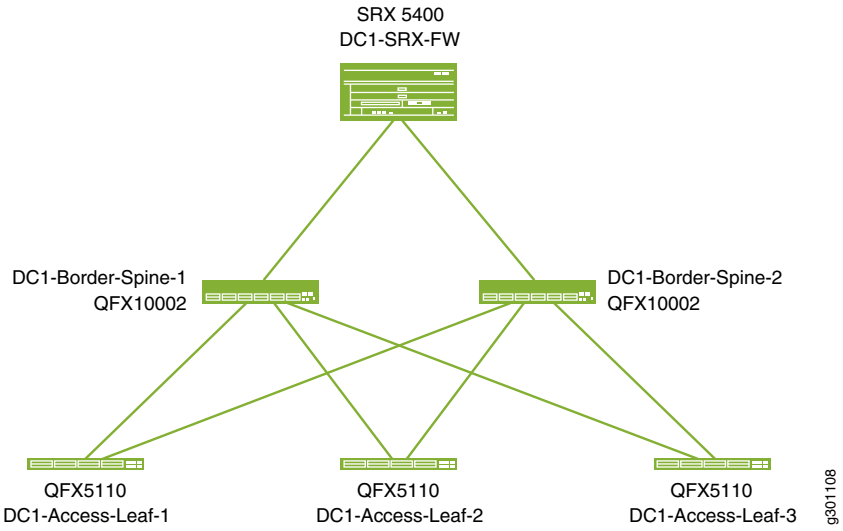
The PNF device exchanges routes with the logical routers over eBGP. It learns routes from one logical router and advertises them to the other logical router.

The first step to creating a service chain is to onboard the SRX. You do this by invoking the same Create Fabric wizard you used earlier, but this time you're discovering an existing (brownfield, already configured) device. After you onboard the SRX, you can then logically insert the SRX between the two segmented networks.

Onboard the PNF Device

Before you start, physically connect the SRX to the spine switches as shown in Figure 11. The SRX in this use case is already up and running and has a hostname DC1-SRX-FW. By onboarding this device, you are telling Contrail Networking to add this device to the fabric. Contrail Networking then configures the fabric port IP addresses on the SRX and the fabric port IP addresses on the spine switches.

Figure 11: PNF Physical View



1. Select **INFRASTRUCTURE>Fabrics** and click on the DC1-Fabric that you created previously.

   The **Fabric devices** window appears listing all the devices in the DC1-Fabric.

2. Select **Action>Brownfield wizard**.

   The **Create Fabric** wizard is launched, but you are not creating a fabric in this procedure. You are using the wizard to add a PNF device to the fabric you created earlier.

3. Fill in the fields on this page as shown in Table 3. Since this is an existing SRX, all you need to do is supply the login credentials and the management IP address. Leave all other fields at their defaults.

   Table 13: Fields in the Create Fabric Page

Fields	Meaning	Setting in this Example
Device credentials>Username	The username to log in to the device.	*<existing-username>*
Device credentials>Password	The password to log in to the device.	*<existing-password>*
Management subnets>CIDR	The IP address of the device on the management network.	192.168.1.147/32 This is the existing IP address of the device.

Table 13: Fields in the Create Fabric Page *(continued)*

Fields	Meaning	Setting in this Example
Management subnets>Gateway	The gateway (if it exists).	Delete the recommended gateway and leave blank.

Click **Next** to launch the device discovery process.

4. After Contrail Networking finishes device discovery, click **Next** to progress to role assignment.

5. Select the row for the PNF device you just discovered and click the **Assign Role** icon on the far right of the row.

   The **Assign role to devices** window appears.

6. Set the role for the SRX as shown in Table 14 and click **Assign**.

   Table 14: DC1-Access-Leaf-3 Roles

Role	Setting in this Example
Physical	pnf
Routing Bridging Roles	PNF-Servicechain

7. Click **Autoconfigure** to start auto-configuration.

8. When auto-configuration is complete, click **Next** and then click **Finish** to skip the assignment of telemetry profiles.

You have now onboarded the SRX and can now proceed to creating the service chain.

Create PNF Service Chain

In order to create the service chain, you first create a template that describes how the PNF device is connected to the fabric. The PNF device can be reused for multiple overlay applications. Creating a template saves you from configuring this information for future service chains.

1. Select **SERVICES>Catalog** to bring up the Catalog page.

2. Click the **PNF** tab to bring up the PNF Service Templates window and click **Create>Template**.

   The Create PNF Service Template page appears.

3. Fill in the fields on this page as shown in Table 15.

Table 15: PNF Service Template

Field	Meaning	Setting in this Example
Name	The name you want to call this service template.	DC1-SRX5400
PNF Device	The name of the device. This is the existing pre-configured hostname of the SRX.	DC1-SRX-FW
PNF Left Interface	The interface on one side of the SRX.	et-1/2/0
PNF Left Fabric	The name of the fabric on one side of the SRX.	DC1-Fabric
PNF Left Attachment Points>Physical Router	The router attached to the left interface of the SRX.	DC1-Border-Spine-1
PNF Left Attachment Points>Left Interface	The interface on the router attached to the left interface of the SRX.	et-0/0/3
PNF Right Interface	The interface on the other side of the SRX.	et-1/2/1
PNF Right Fabric	The name of the fabric on the other side of the SRX.	DC1-Fabric
PNF Right Attachment Points>Physical Router	The router attached to the right interface of the SRX.	DC1-Border-Spine-2
PNF Right Attachment Points>Right Interface	The interface on the router attached to the right interface of the SRX.	et-0/0/3

Click **Create** to create the template called DC1-SRX5400-template. Now you can create the service chain instance that uses this template in the next few steps.

4. Select **SERVICES>Deployments** to bring up the Deployments page.

5. Click the **PNF** tab to bring up the PNF Service Instances window and click **Create>Instance**.

   The Create PNF Service Instance page appears.

6. Fill in the fields on this page as shown in Table 16.

   Table 16: Create PNF Service Instance

Fields	Meaning	Setting in this Example
Name	The name you want to call this service chain instance.	Web-To-DB
Service Template	The service template you want to use. This is the name of the service template you created earlier.	DC1-SRX5400-template

Table 16: Create PNF Service Instance *(continued)*

Fields	Meaning	Setting in this Example
PNF eBGP ASN	The ASN for the PNF device. This is used by eBGP to exchange routes in the overlay.	65100
Left Tenant Logical Router	The logical router attached to the left interface. This assignment is arbitrary since each spine switch has the same two logical routers instantiated.	Web-LR
PNF Left BGP Peer ASN	The ASN for the left overlay network. There is one ASN for the entire overlay.	65000
Left Service VLAN	The VLAN ID for the left interface.	1001
Right Tenant Logical Router	The logical router attached to the right interface. This is the other logical router.	DB-LR
PNF Right BGP Peer ASN	The ASN for the right overlay network. There is one ASN for the entire overlay.	65000
Right Service VLAN	The VLAN ID for the right interface.	1002

When you finish filling in the fields, click **Create**.

You have now created the service chain. Routes between the two segmented networks are now exchanged, and inter-network traffic can now traverse the SRX. By default, Contrail Networking configures the SRX to be permissive (that is, an <any>-<any>-<any> permit policy). To change the policy, log in to the SRX and configure the policy as you normally do.

Figure 12 shows the physical connectivity of the SRX in the network. Note that typically you would have multiple SRX devices for redundancy and routing efficiency.

Figure 12: PNF Physical View After Configuration

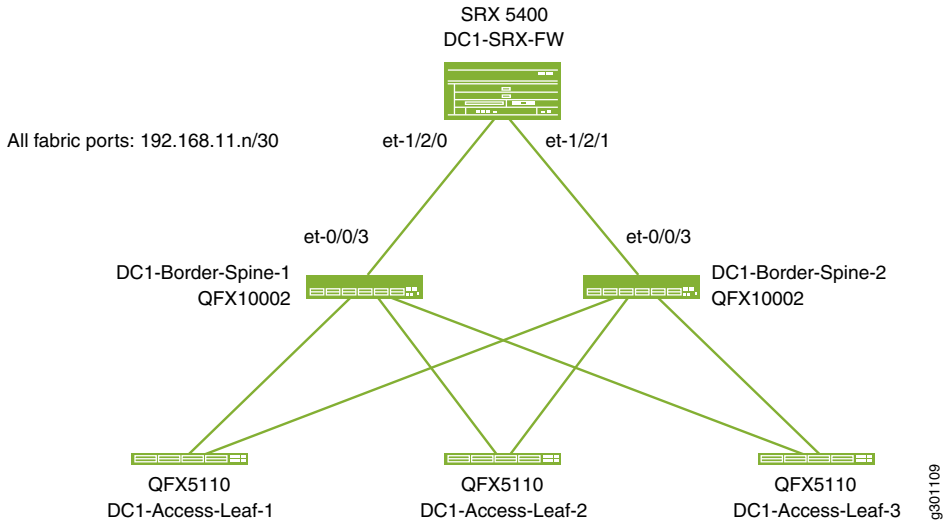
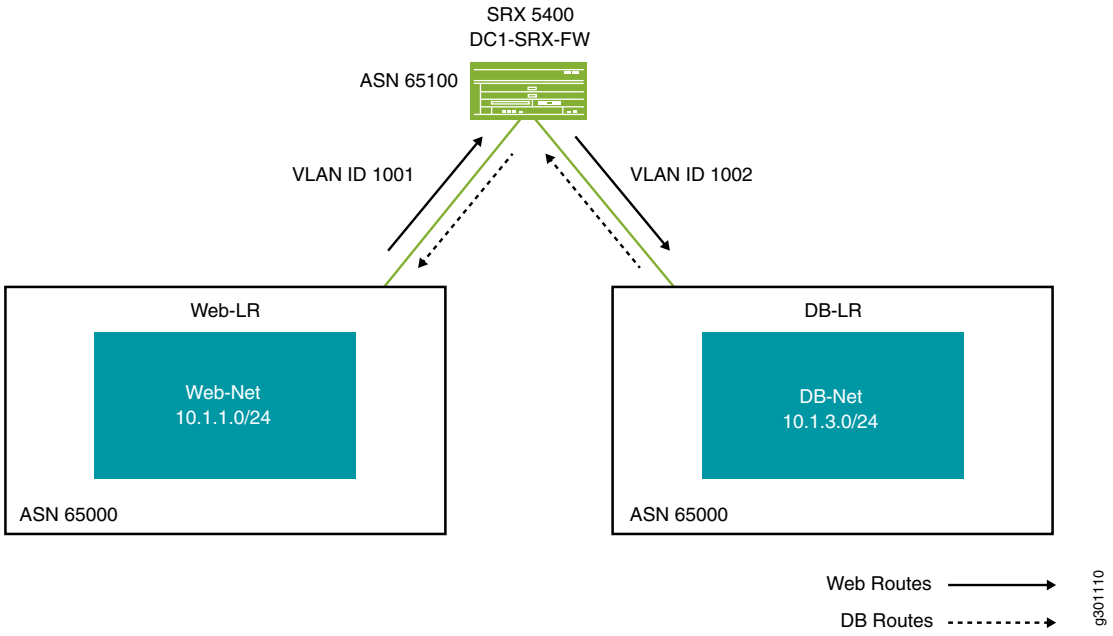


Figure 13 shows the SRX in the overlay. Routes from one segmented network are advertised across to the other segmented network through the firewall.

Figure 13: PNF Logical View After Configuration



Summary

In this use case, you used Contrail Command to onboard a fabric of QFX switches, configure segmented networks on top of this fabric, and set up a path between the segmented networks through an SRX firewall.

Although the example network in this use case is fairly basic, these same fundamental procedures apply equally well to cloud-scale data centers. The notion of configuring the fabric as a whole while leaving individual device configuration details to Contrail Networking allows these procedures to scale gracefully. With a well-crafted device YAML file, Contrail Networking can discover and onboard your fabric without requiring you to work with each fabric device individually.

While using Contrail Networking to onboard a fabric is relatively straightforward, it is just a prerequisite to what you really want to do, which is to use an SDN solution to create the overlay segmented networks on top of this fabric. Since you typically onboard a fabric once at the beginning and only occasionally thereafter, it is equally if not more important for Contrail Networking to provide an efficient way for you to set up your overlay segmented networks, which you perform on a regular basis. In this use case, you created overlay segmented networks by defining the network, the routing instance, and the endpoints, all from a network-wide point of view, again achieving scale that you would not normally achieve if you had to configure the network devices individually.

WHAT'S NEXT

> Now that you've created the segmented networks and connected them through an SRX firewall, you should log in to the SRX firewall and change the security policies to align with your security requirements. See SRX documentation for information on how to set up security policies.