

## Network Configuration Example

---

Two-Member QFX Series Virtual Chassis  
Upgrade Procedure



Published  
2025-12-17

Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, California 94089  
USA  
408-745-2000  
[www.juniper.net](http://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.

*Network Configuration Example Two-Member QFX Series Virtual Chassis Upgrade Procedure*  
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

## [Example: Two-Member QFX Series Virtual Chassis Upgrade](#)

[Upgrading Two-Member QFX Series Virtual Chassis](#) | 2

[About This Network Configuration Example](#) | 2

[Use Case Overview](#) | 2

[Technical Overview](#) | 3

[Configuration Example](#) | 4

[Requirements](#) | 4

[Overview](#) | 5

[Configuration](#) | 6

[Conclusion](#) | 13

# About This Guide

Use this network configuration example to manually upgrade a two-member QFX Series Virtual Chassis.

# 1

CHAPTER

## Example: Two-Member QFX Series Virtual Chassis Upgrade

---

### IN THIS CHAPTER

- [Upgrading Two-Member QFX Series Virtual Chassis | 2](#)

---

# Upgrading Two-Member QFX Series Virtual Chassis

## IN THIS SECTION

- [About This Network Configuration Example | 2](#)
- [Use Case Overview | 2](#)
- [Technical Overview | 3](#)
- [Configuration Example | 4](#)
- [Conclusion | 13](#)

## About This Network Configuration Example

This network configuration example (NCE) shows how to upgrade a two-member QFX Series Virtual Chassis when the nonstop software upgrade (NSSU) process is either not available or undesirable. This process minimizes service disruption and has minimal impact on data center workloads. The NSSU feature for the QFX Series is supported between specific releases that can be found in the QFX Series section of the Junos Release Notes.

## SEE ALSO

[\*Understanding QFX Series Virtual Chassis\*](#)

[\*Upgrading Software on a Virtual Chassis and Mixed Virtual Chassis Using Nonstop Software Upgrade\*](#)

## Use Case Overview

The Virtual Chassis capabilities are important aspects of the QFX Series portfolio. A common Virtual Chassis use case in data centers is aggregating multiple top-of-rack switches into a single logical entity for simplicity in management and operations of high-availability pairs. In this use case, racks of servers are multihomed to two top-of-rack QFX Series switches. The switches are configured into a Virtual Chassis pair and provide resiliency to the network path if one of the QFX Series devices fails.

When these devices need software updates, you will generally use the NSSU capabilities of the Virtual Chassis to upgrade the devices. The NSSU upgrade selectively upgrades the Virtual Chassis member devices in an intelligent order to minimize service disruption to the connected servers.

However, there are certain upgrade scenarios where the “from” release and “to” release do not support the NSSU upgrade process. When upgrading in these scenarios, we can achieve a similar result through a series of manual operations. This use case covers the non-NSSU upgrade path between two releases.

## Technical Overview

The process to manually upgrade a two-member Virtual Chassis closely mimics the steps taken by the automated NSSU process. The sequence leverages the high-availability design to systematically remove one device from service to perform the upgrade and reboot. When the server nodes are dual homed to each of the devices, the network can withstand the removal of one of the Virtual Chassis members during the upgrade window. There is a reduction of overall network bandwidth during the process, but the network remains available.

The Virtual Chassis feature uses a primary/backup concept to keep the device state synchronized between the members of the Virtual Chassis. While one device handles the traffic, we take the other device offline and upgrade it. To upgrade both devices, we take the following steps:

1. First, we shift all traffic to the primary device.
2. Once the backup device is no longer handling server traffic, we break apart the Virtual Chassis.
3. With the backup device completely isolated, we upgrade the software on the backup device and reboot it. The backup device will keep a copy of the original network configuration.
4. After the upgraded backup comes online, we shift server traffic from the primary device to the backup device. Once the backup is handling the network load, we upgrade and reboot the primary device.
5. After the primary device comes online, we shift traffic back to the primary device.
6. Finally, we re-enable the Virtual Chassis links between the two devices to re-create the Virtual Chassis pair running the new software version.

## Configuration Example

### IN THIS SECTION

- [Requirements | 4](#)
- [Overview | 5](#)
- [Configuration | 6](#)

This configuration example shows how to upgrade a two-member Virtual Chassis from Junos OS Release 14.1X53-D49.1 to Junos OS Release 18.1R2.6. As it happens, this is not a supported combination for the NSSU feature, so we will use the manual process outlined below.

This example uses a basic Virtual Chassis configuration, but the process here is adaptable to a number of different use cases.

### Requirements

Use this procedure to upgrade both members of a two-member Virtual Chassis consisting of QFX5100, QFX5110, QFX5220, or QFX5200 switches to the same Junos OS Release version. We strongly recommend that both members of the Virtual Chassis are the same platform, like in this example.

Before you begin:

- Make sure the Virtual Chassis is comprised of two members, where one is configured as a primary member and the other is configured as a backup member.
- Configure the Virtual Chassis in Virtual Chassis mode (that is, not Virtual Chassis Fabric mode)
- Make sure the Virtual Chassis is performing Layer 2 functions only (that is, no IRBs or routing protocols)

This example uses the following hardware and software components:

- Two QFX5100-48S-6Q devices running Junos OS Release 14.1X53-D49.1
- Junos OS Release 18.1R2.6
- Test server running Ubuntu Linux 16.04

## Overview

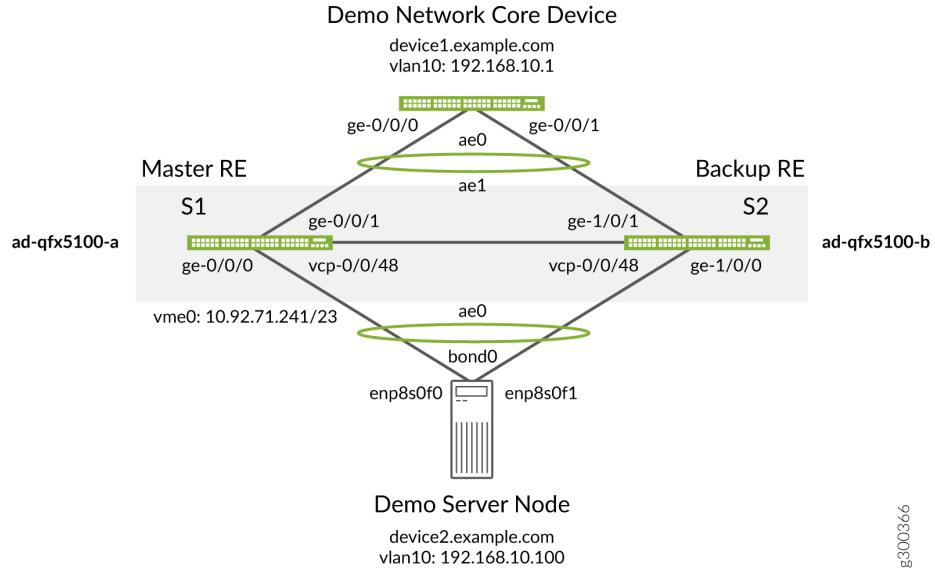
### IN THIS SECTION

- Topology | 5

The upgrade between releases requires a specific sequence of steps coordinated among the network elements to ensure a minimum of downtime during the transition. As indicated in the diagram, the general procedure will leverage the high availability characteristics of modern servers with redundant connections to the Virtual Chassis during the transition.

At the start of the upgrade, we begin with a functional two-member Virtual Chassis. Our goal is to upgrade to a new Junos OS release with minimum traffic disruption. To achieve this, we will break apart the Virtual Chassis and upgrade the member devices as standalone units. After the devices have been upgraded, we will re-connect them and re-establish the Virtual Chassis.

## Topology



## Configuration

### IN THIS SECTION

- [Procedure | 6](#)

## Procedure

### Step-by-Step Procedure

To upgrade the devices:

1. Verify the Virtual Chassis state. Check the parameters of the Virtual Chassis and verify you are working with a two-member Virtual Chassis that is operational.

```
{master:0}
user@qfx5100-a> show virtual-chassis

Preprovisioned Virtual Chassis
Virtual Chassis ID: 235e.5bda.52ca
Virtual Chassis Mode: Enabled

                                         Mstr          Mixed Route Neighbor List
Member ID  Status  Serial No    Model      prio  Role      Mode  Mode ID  Interface
0 (FPC 0)  Prsnt  TA3714120212 qfx5100-48s-6q 129  Master*    N  Virtual Chassis  1
Virtual Chassisip-255/0/48
1 (FPC 1)  Prsnt  TA3714120217 qfx5100-48s-6q 129  Backup      N  Virtual Chassis  0
Virtual Chassisip-255/0/48
```

2. Upload the new software to the Virtual Chassis members. Copy the new software to **/var/tmp** on the Virtual Chassis primary and backup devices. This step stages software on both switches for the upgrade procedure. The copy operation will take some time to complete while it transfers the Junos OS images.

```
file copy http://server.example.com/volume/download/software/junos/18.1R2.6/jinstall-host-
qfx-5-18.1R2.6-signed.tgz /var/tmp/
```

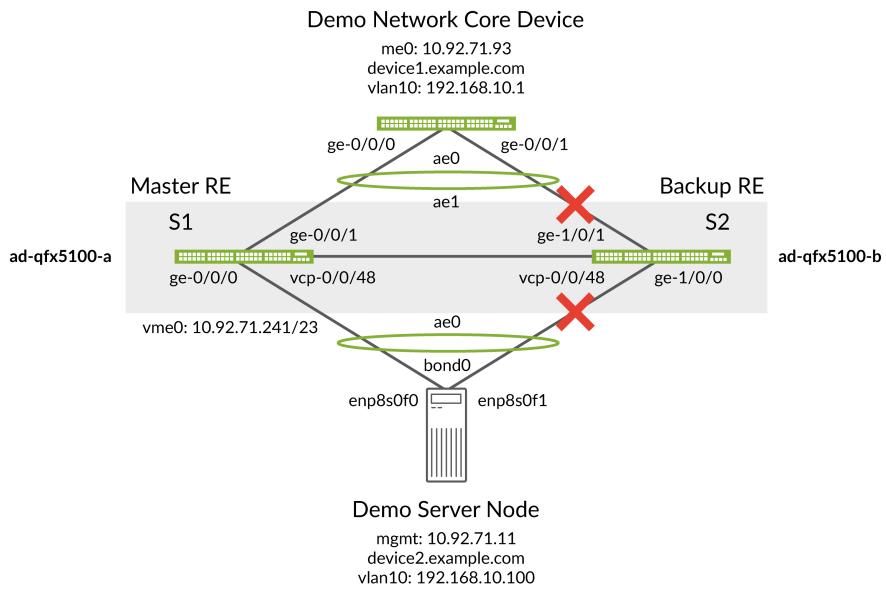
3. We recommend disabling split detection whenever you form a Virtual Chassis with only two members. If you do not disable split detection, the primary device may take on a linecard role and stop the control and data planes when you disable the backup Routing Engine later in this example.

Since you started this NCE with a fully configured Virtual Chassis, this option should already be configured. If it is not for any reason, configure it now.

```
user@qfx5100-a# set virtual-chassis no-split-detection
```

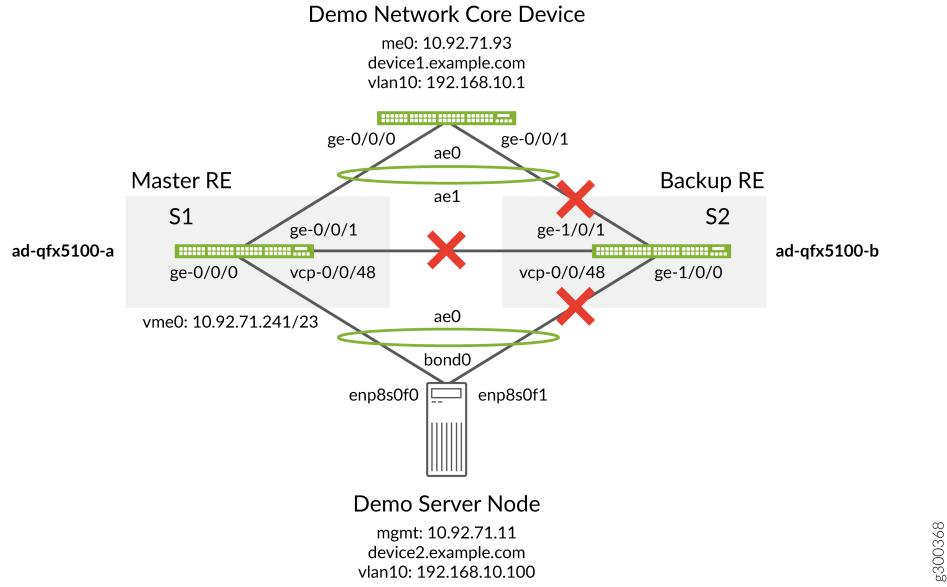
4. Disable server-facing ports on the backup Routing Engine to minimize disruption during switchover.

```
user@qfx5100-b# wildcard range set interfaces ge-1/0/[0-47] disable
wildcard range set interfaces xe-1/0/[0-47] disable
commit and-quit
```



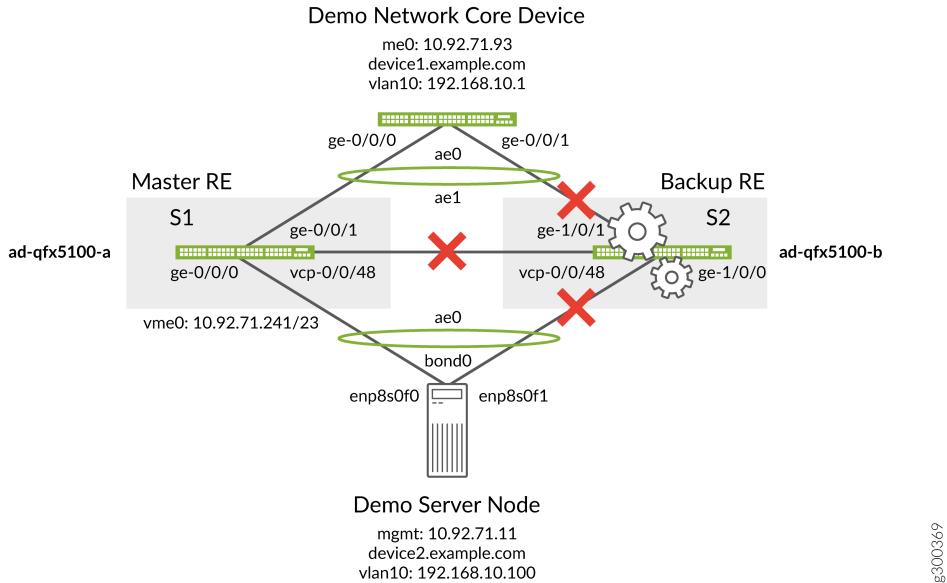
5. Disable VCP ports toward the backup Routing Engine. This breaks up the Virtual Chassis.

```
user@qfx5100-b> request virtual-chassis vc-port delete pic-slot 0 port 48 member 1
request virtual-chassis vc-port delete pic-slot 0 port 48 member 0
```



6. Upgrade the backup Routing Engine. When upgrading to a 18.2 or newer Junos release you should include the force-host option. This ensures that both the host OS and the Junos binaries are updated and remain matched.

```
{master:1}
user@qfx5100-b> request system software add /var/tmp/jinstall-host-qfx-5-18.1R2.6-
signed.tgz force-host reboot
```



7. Swap the server-facing ports by disabling the server-facing ports on the primary device and re-enabling the server-facing ports on the backup simultaneously. Implement the same configuration on the backup and primary devices to modify any configuration left over from when the two devices were part of the Virtual Chassis.

On the backup QFX, first disable the server-facing ports on the primary device. Do not commit the configuration:

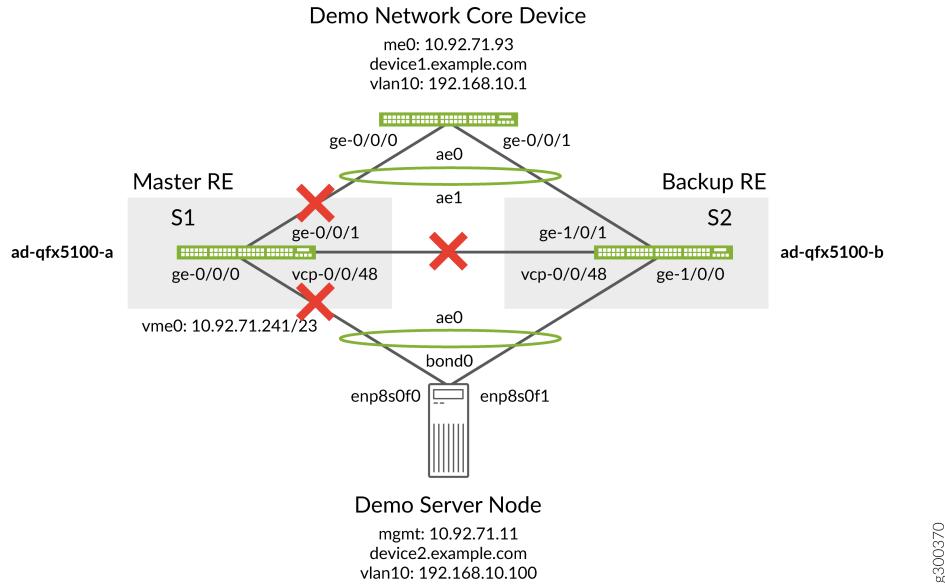
```
user@qfx5100-b# wildcard range set interfaces ge-0/0/[0-47] disable
wildcard range set interfaces xe-0/0/[0-47] disable
```

Then re-enable the server-facing ports on the backup by deleting the previous configuration. Commit the configuration:

```
wildcard range delete interfaces ge-1/0/[0-47] disable
wildcard range delete interfaces xe-1/0/[0-47] disable
commit and-quit
```

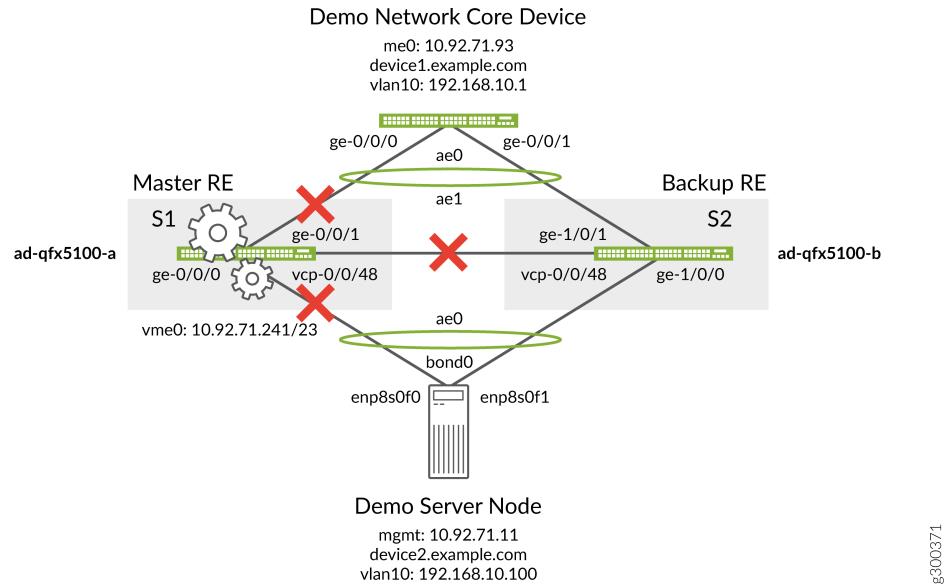
Repeat the configuration on the primary QFX:

```
user@qfx5100-a# wildcard range set interfaces ge-0/0/[0-47] disable
wildcard range set interfaces xe-0/0/[0-47] disable
wildcard range delete interfaces ge-1/0/[0-47] disable
wildcard range delete interfaces xe-1/0/[0-47] disable
commit and-quit
```



8. Upgrade the primary Routing Engine. When upgrading to a 18.2 or newer Junos release you should include the force-host option. This ensures that both the host OS and the Junos binaries are updated and remain matched.

```
{master:0}
user@qfx5100-a> request system software add /var/tmp/jinstall-host-qfx-5-18.1R2.6-
signed.tgz force-host reboot
```



- Swap the server facing ports back to the primary device. Re-enable the server-facing ports on the primary device to speed up LACP convergence when the Virtual Chassis comes back. Implement the same configuration on the backup and primary devices to modify any configuration left over from when the two devices were part of the Virtual Chassis.

On the backup QFX, first re-enable the server-facing ports on the primary device by deleting the previous configuration. Do not commit the configuration:

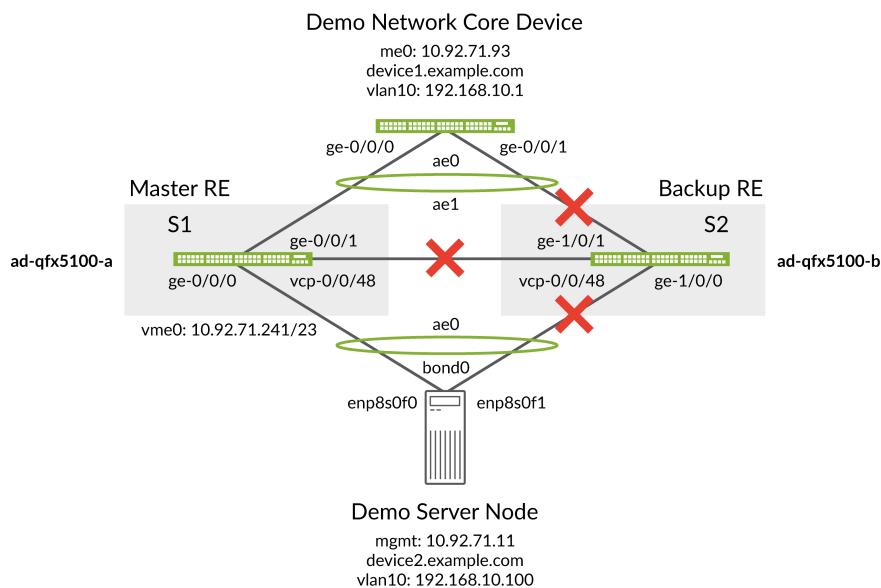
```
user@qfx5100-b# wildcard range delete interfaces ge-0/0/[0-47] disable
wildcard range delete interfaces xe-0/0/[0-47] disable
```

Then disable the server-facing ports on the backup and commit the configuration:

```
wildcard range set interfaces ge-1/0/[0-47] disable
wildcard range set interfaces xe-1/0/[0-47] disable
commit and-quit
```

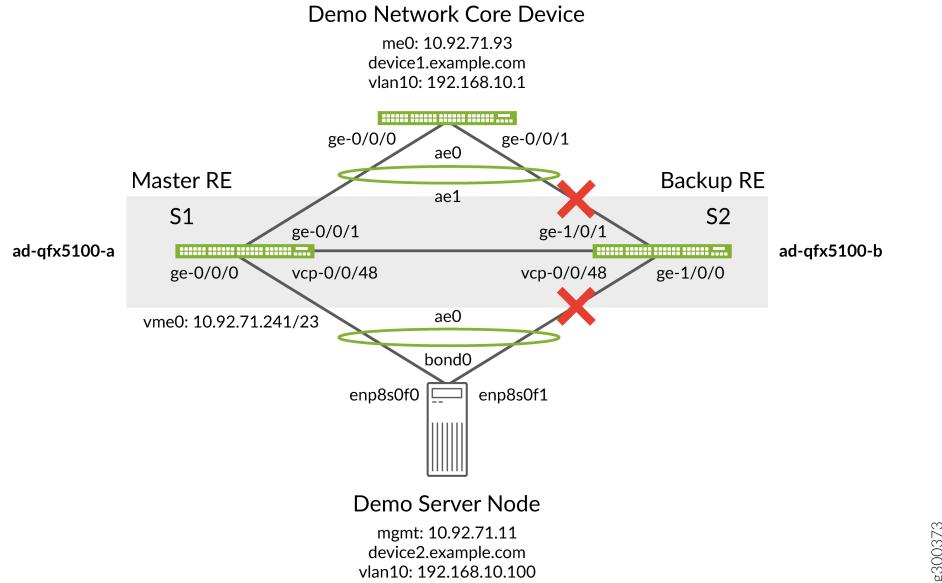
Repeat the configuration on the primary QFX:

```
user@qfx5100-a# wildcard range delete interfaces ge-0/0/[0-47] disable
wildcard range delete interfaces xe-0/0/[0-47] disable
wildcard range set interfaces ge-1/0/[0-47] disable
wildcard range set interfaces xe-1/0/[0-47] disable
commit and-quit
```



10. Re-enable the VCP ports on both boxes to re-establish the Virtual Chassis.

```
user@qfx5100-a> request virtual-chassis vc-port set pic-slot 0 port 48 member 0
request virtual-chassis vc-port set pic-slot 0 port 48 member 1
```



11. Verify you have re-established the Virtual Chassis.

```
user@qfx5100-a> show virtual-chassis
```

Preprovisioned Virtual Chassis

Virtual Chassis ID: 235e.5bda.52ca

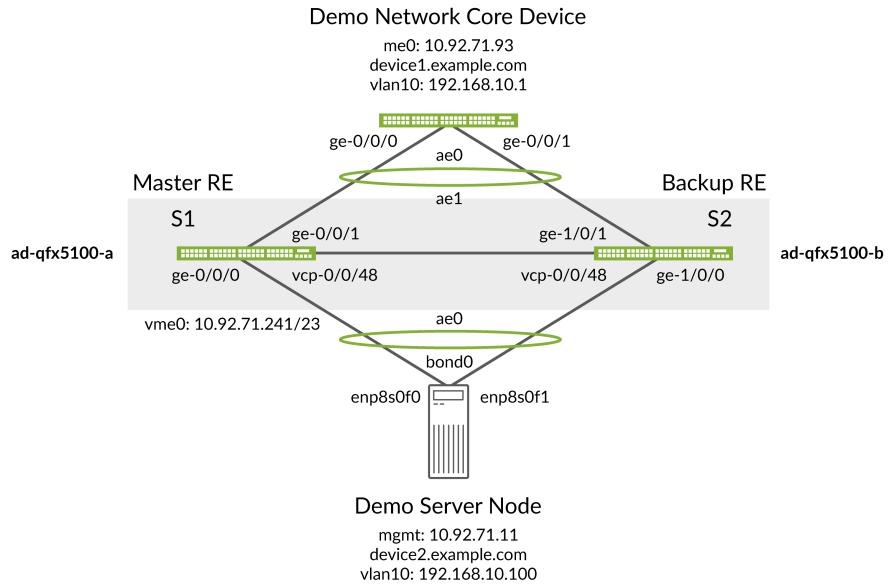
Virtual Chassis Mode: Enabled

| Member ID | Status | Serial No       | Model          | Mstr |         | Mixed Route Neighbor List |                 |           |
|-----------|--------|-----------------|----------------|------|---------|---------------------------|-----------------|-----------|
|           |        |                 |                | prio | Role    | Mode                      | Mode ID         | Interface |
| 0 (FPC 0) | Prsnt  | TA3714120212    | qfx5100-48s-6q | 129  | Master* | N                         | Virtual Chassis | 1         |
|           |        | Virtual Chassis | sp-255/0/48    |      |         |                           |                 |           |
| 1 (FPC 1) | Prsnt  | TA3714120217    | qfx5100-48s-6q | 129  | Backup  | N                         | Virtual Chassis | 0         |
|           |        | Virtual Chassis | sp-255/0/48    |      |         |                           |                 |           |

12. Enable access ports on both members. Now that the Virtual Chassis has been re-established, we need to re-establish the access ports so we can use the primary Routing Engine em0 address to communicate with the newly upgraded Virtual Chassis.

On the primary QFX:

```
user@qfx5100-a# wildcard range delete interfaces ge-1/0/[0-47] disable
wildcard range delete interfaces xe-1/0/[0-47] disable
commit and-quit
```



**NOTE:** If you intend to add more devices to your two-member Virtual Chassis, re-enable split detection.

You have upgraded your two-member Virtual Chassis.

## Conclusion

Virtual Chassis is an important architecture design for datacenter high availability. You now know how to manually upgrade a two-member QFX Series Virtual Chassis with minimal impact to your datacenter workloads. Use the procedure outlined in this document to upgrade any Virtual Chassis with a similar topology when NSSU is not available or not desirable.