Nexus 7000 Lab & Answers:

There are 16 labs contained in this document.

It is assumed that you are already proficient with Cisco IOS and the most of the concepts here are not new. These labs are meant to teach you how to configure the Nexus 7000 NX-OS switch and get you comfortable with some of the differences between IOS and NX-OS.

Items in BLACK print are tasks
Items in BLUE print are notes
Items in RED are the switch interface output and commands
Items in **RED BOLD** and **BLUE BOLD** are the commands or command references.

Lab 1:

Creating VDCs and assigning ports to the VDC

Tasks:

- Check to make sure there is only 1 VDC
- Create the following VDCs:
  VDC-2
  VDC-3
- Check to make sure that the VDCs are created
- Assign the following ports to VDC-2 on both switches
  - The first 10GB port group on Card 7 and on Card 8
  - The first 24 ports on Card 3
- Assign the following ports to VDC-3:
  - The second 10GB port group on Card 7 and Card 8
  - The last 24 Ports on Card 3
- Check to see if the ports are assigned correctly

To see what VDCs are on the switch already, ***show vdc*** *is the command*

```
N7K1(config)# sh vdc

vdc_id  vdc_name                          state               mac
------  --------                          -----               ----------
1       N7K1                              active              00:26:98:1b:f0:41
```

To create a VDC, enter configuration mode and use the ***VDC name*** *command*

```
N7K1(config)# vdc VDC-2
Note:  Creating VDC, one moment please ...
N7K1(config-vdc)# 2010 Jun 12 00:25:29 N7K1 %$ VDC-1 %$ %VDC_MGR-2-VDC_ONLINE:
   vdc 2 has come online
N7K1(config)# exit
N7K1(config)# vdc VDC-3
Note:  Creating VDC, one moment please ...
N7K1(config-vdc)# 2010 Jun 12 00:27:59 N7K1 %$ VDC-1 %$ %VDC_MGR-2-VDC_ONLINE:
   vdc 3 has come online

N7K1(config-vdc)# show vdc

vdc_id  vdc_name                          state               mac
------  --------                          -----               ----------
1       N7K1                              active              00:26:98:1b:f0:41
2       VDC-2                             active              00:26:98:1b:f0:42
3       VDC-3                             active              00:26:98:1b:f0:43

N7K1(config-vdc)#
```

Note: Notice you can enter SHOW command and such from configuration mode in NX-OS.

This is actually a trick question in some ways. You need to keep in mind that on a 32-port 10GB card that port groups run even and odd. So the first port-group for Card 7 they are ports e7/1, e7/3, e7/5, and e7/7 and on Card 8 they are ports e8/1, e8/3, e8/5, and e8/7.

In configuration mode, switch to the appropriate VDC and use the ***allocate interface*** *command.*

```
N7K1(config-vdc)# vdc vdc-2
N7K1(config-vdc)# allocate interface ethernet 7/1, e7/3, e7/5, e7/7
  Moving ports will cause all config associated to them in source vdc to be
  removed. Are you sure you want to move the ports (y/n)?  [yes] y
N7K1(config-vdc)# allocate interface ethernet 8/1, e8/3, e8/5, e8/7
  Moving ports will cause all config associated to them in source vdc to be
  removed. Are you sure you want to move the ports (y/n)?  [yes] y
N7K1(config-vdc)# allocate interface ether 3/1 – 24
  Moving ports will cause all config associated to them in source vdc to be
  removed. Are you sure you want to move the ports (y/n)?  [yes] y
```

To check to see what ports are assigned, use ***show vdc vdcname membership***

```
N7K1(config-vdc)# sh vdc vdc-2 membership
  vdc_id: 2 vdc_name: VDC-2 interfaces:
          Ethernet3/1           Ethernet3/2           Ethernet3/3
          Ethernet3/4           Ethernet3/5           Ethernet3/6
          Ethernet3/7           Ethernet3/8           Ethernet3/9
          Ethernet3/10          Ethernet3/11          Ethernet3/12
          Ethernet3/13          Ethernet3/14          Ethernet3/15
          Ethernet3/16          Ethernet3/17          Ethernet3/18
          Ethernet3/19          Ethernet3/20          Ethernet3/21
          Ethernet3/22          Ethernet3/23          Ethernet3/24
          Ethernet7/1           Ethernet7/3           Ethernet7/5
          Ethernet7/7           Ethernet8/1           Ethernet8/3
          Ethernet8/5           Ethernet8/7
```

Do the same for the other VDCs

```
N7K1(config-vdc)# vdc vdc-3
N7K1(config-vdc)# allocate interface ethernet 7/2, e7/4, e7/6, e7/8
  Moving ports will cause all config associated to them in source vdc to be
  removed. Are you sure you want to move the ports (y/n)?  [yes] y
N7K1(config-vdc)# allocate interface ethernet 8/2, e8/4, e8/6, e8/8
  Moving ports will cause all config associated to them in source vdc to be
  removed. Are you sure you want to move the ports (y/n)?  [yes] y
N7K1(config-vdc)# allocate interface ether 3/25 - 48
  Moving ports will cause all config associated to them in source vdc to be
  removed. Are you sure you want to move the ports (y/n)?  [yes] y
N7K1(config-vdc)# sh vdc vdc-3 membership
  vdc_id: 3 vdc_name: VDC-3 interfaces:
       Ethernet3/25          Ethernet3/26          Ethernet3/27
       Ethernet3/28          Ethernet3/29          Ethernet3/30
       Ethernet3/31          Ethernet3/32          Ethernet3/33
       Ethernet3/34          Ethernet3/35          Ethernet3/36
       Ethernet3/37          Ethernet3/38          Ethernet3/39
       Ethernet3/40          Ethernet3/41          Ethernet3/42
       Ethernet3/43          Ethernet3/44          Ethernet3/45
       Ethernet3/46          Ethernet3/47          Ethernet3/48
       Ethernet7/2           Ethernet7/4           Ethernet7/6
       Ethernet7/8           Ethernet8/2           Ethernet8/4
       Ethernet8/6           Ethernet8/8
```

Lab 2:

Setting up a Nexus 7000 VDC from scratch

Tasks:

- Setup VDC-2 using the following parameters:
    - Secure password standard
    - Password of ***NXos12345***
    - Enter the basic setup wizard
    - Do not create another login account
    - Do not configure any SNMP strings
    - Set the switch name to ***VDC-2***
    - Do not configure out-of-band (mgmt0)
    - No advanced IP options
    - Enable Telnet
    - Enable SSH
    - Use the default key and bits
    - Set the default interface to L3
    - Set the default switchport interface to shut
- Setup VDC-3 the same way except make the hostname ***VDC-3***

Note:In order to get to VDC-2, you need to ***switchto vdc vdcname*** *the VDC. Once you are there you will automatically be placed into the setup mode.*

```
N7K1# switchto vdc VDC-2

         ---- System Admin Account Setup ----
Do you want to enforce secure password standard (yes/no) [y]: y

    Enter the password for "admin": NXos12345
  Confirm the password for "admin": NXos12345

         ---- Basic System Configuration Dialog VDC: 2 ----
This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.

Please register Cisco Nexus7000 Family devices promptly with your
supplier. Failure to register may affect response times for initial
service calls. Nexus7000 devices must be registered to receive
entitled support services.
Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.
Would you like to enter the basic configuration dialog (yes/no): y
         Create another login account (yes/no) [n]: n
  Configure read-only SNMP community string (yes/no) [n]: n
  Configure read-write SNMP community string (yes/no) [n]: n
  Enter the switch name : VDC-2
  Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: n
  Configure advanced IP options? (yes/no) [n]: n
  Enable the telnet service? (yes/no) [n]: y
  Enable the ssh service? (yes/no) [y]: y
    Type of ssh key you would like to generate (dsa/rsa) [rsa]: rsa
    Number of rsa key bits <768-2048> [1024]: 1024
  Configure default interface layer (L3/L2) [L3]: L3
  Configure default switchport interface state (shut/noshut) [shut]: shut
```

```
The following configuration will be applied:
  password strength-check
  switchname VDC-2
  feature telnet
  ssh key rsa 1024 force
  feature ssh
  no system default switchport
  system default switchport shutdown
Would you like to edit the configuration? (yes/no) [n]: n
Use this configuration and save it? (yes/no) [y]: y
Configuration update aborted: This vdc has had a global configuration change since the
last saved config. Please save config in default vdc before proceeding
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2010, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
N7K1-VDC-2#
```

There is a good chance you are still in VDC-2, if that is the case you need to ***switchback*** *to the main VDC before* ***switchto vdc vdcname.***

```
N7K1-VDC-2# switchback
N7K1# switchto vdc vdc-3
         ---- System Admin Account Setup ----
Do you want to enforce secure password standard (yes/no) [y]: y
Enter the password for "admin": NXos12345
  Confirm the password for "admin": NXos12345
         ---- Basic System Configuration Dialog VDC: 3 ----
This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.
Please register Cisco Nexus7000 Family devices promptly with your
supplier. Failure to register may affect response times for initial
service calls. Nexus7000 devices must be registered to receive
entitled support services.
Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.
Would you like to enter the basic configuration dialog (yes/no): y
  Create another login account (yes/no) [n]: n
  Configure read-only SNMP community string (yes/no) [n]: n
  Configure read-write SNMP community string (yes/no) [n]: n
  Enter the switch name : VDC-3
  Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: n
  Configure advanced IP options? (yes/no) [n]: n
  Enable the telnet service? (yes/no) [n]: y
  Enable the ssh service? (yes/no) [y]: y
    Type of ssh key you would like to generate (dsa/rsa) [rsa]: rsa
    Number of rsa key bits <768-2048> [1024]: 1024
  Configure default interface layer (L3/L2) [L3]: L3
  Configure default switchport interface state (shut/noshut) [shut]: shut
The following configuration will be applied:
  password strength-check
  switchname VDC-3
  feature telnet
  ssh key rsa 1024 force
  feature ssh
  no system default switchport
  system default switchport shutdown
Would you like to edit the configuration? (yes/no) [n]: n
       Use this configuration and save it? (yes/no) [y]: y
Configuration update aborted: This vdc has had a global configuration change
       since the last saved config. Please save config in default vdc before
       proceeding
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2010, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php

N7K1-VDC-3#
```

Lab 3:

Creating layer 3 interfaces between the Nexus 7000 switches

Tasks:

- Set Port e8/2 on N7K1-VDC-3 into dedicated mode
- Set Port e8/2 on N7K2-VDC-3 into shared mode
- Set the IP address on N7K1-VDC-3 e8/2 to 192.168.1.1 / 24
- Set the IP address on N7K2-VDC-3 e8/2 to 192.168.1.2 / 24
- PING across the interfaces to test connectivity

Enter configuration mode and change to the interface. To set the rate mode, use the ***rate-mode dedicated*** *command. To assign the ip address, use the* ***ip address IP/SUBNET*** *command. Remember, that when you did the setup the default interface mode is SHUTDOWN, so be sure to* ***no shut*** *the interface to enable it.*

```
N7K1-VDC-3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
N7K1-VDC-3(config)# int e8/2
N7K1-VDC-3(config-if)# rate-mode dedicated
N7K1-VDC-3(config-if)# ip add 192.168.1.1/24
N7K1-VDC-3(config-if)# no shut
N7K1-VDC-3(config-if)# ^Z
N7K1-VDC-3#
```

Do the same for the other switch. Note that you do not need to set the ***rate-mode*** *as shared is the default configuration.*

```
N7K2-VDC-3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
N7K2-VDC-3(config)# int e8/2
N7K2-VDC-3(config-if)# ip add 192.168.1.2 255.255.255.0
N7K2-VDC-3(config-if)# no shut
N7K2-VDC-3(config-if)# ^Z
```

To test connectivity ***PING across the interfaces.***

```
N7K2-VDC-3# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
Request 0 timed out
64 bytes from 192.168.1.1: icmp_seq=1 ttl=254 time=1.353 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=254 time=0.596 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=254 time=0.59 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=254 time=0.598 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 4 packets received, 20.00% packet loss
round-trip min/avg/max = 0.59/0.784/1.353 ms
N7K2-VDC-3#
```

Note: The default rate-mode for a 10GB interface is SHARED mode. You will only need to change the rate-mode when you want to use DEDICATED mode.
Also note that the switch will take either a / type subnet for a subnet mask. i.e. - /24 for 255.255.255.0

Lab 4:

Creating VLANs, VLAN interfaces, and VTP

Tasks:

- Create the VTP domain of N7KLab and set it to transparent
- Create the following VLANS and assign the appropriate names:

VLAN	Name
100	VL100
110	VL110
120	VL120
130	VL130

- Assign the following IP addresses to the VLANS

VLAN	N7K1	N7K2
100	10.100.1.2/24	10.100.1.3/24
110	10.110.1.2/24	10.110.1.3/24
120	10.120.1.2/24	10.120.1.3/24
130	10.130.1.2/24	10.130.1.3/24

*We will need to first enable the **feature vtp** in order to configure VTP. Once enabled, you use **vtp** commands and **VLAN** commands just like IOS.*

```
N7K2-VDC-2(config)# feature vtp
N7K2-VDC-2(config)# vtp domain N7KLab
N7K2-VDC-2(config)# vtp mode transparent
N7K2-VDC-2(config)#

N7K1-VDC-2(config)# vlan 100
N7K1-VDC-2(config-vlan)# name VL100
N7K1-VDC-2(config-vlan)# vlan 110
N7K1-VDC-2(config-vlan)# name VL110
N7K1-VDC-2(config-vlan)# vlan 120
N7K1-VDC-2(config-vlan)# name VL120
N7K1-VDC-2(config-vlan)# vlan 130
N7K1-VDC-2(config-vlan)# name VL130

N7K2-VDC-2(config)# vlan 100
N7K2-VDC-2(config-vlan)# name VL100
N7K2-VDC-2(config-vlan)# vlan 110
N7K2-VDC-2(config-vlan)# name VL110
N7K2-VDC-2(config-vlan)# vlan 120
N7K2-VDC-2(config-vlan)# name VL120
N7K2-VDC-2(config-vlan)# vlan 130
N7K2-VDC-2(config-vlan)# name VL130
```

*In order to create VLAN interfaces, you need to enable the **feature interface-vlan**. Once enabled, you creat VLAN interfaces just like IOS.*

```
N7K1-VDC-2(config)# feature interface-vlan

N7K1-VDC-2(config)# int vlan 100
N7K1-VDC-2(config-if)# ip add 10.100.1.2/24
N7K1-VDC-2(config-if)# int vlan 110
N7K1-VDC-2(config-if)# ip add 10.110.1.2/24
```

```
N7K1-VDC-2(config-if)# int vlan 120
N7K1-VDC-2(config-if)# ip add 10.120.1.2/24
N7K1-VDC-2(config-if)# int vlan 130
N7K1-VDC-2(config-if)# ip add 10.130.1.2/24

And on the other switch
N7K2-VDC-2(config)# feature interface-vlan
N7K2-VDC-2(config)# int vlan 100
N7K2-VDC-2(config-if)# ip add 10.100.1.3/24
N7K2-VDC-2(config-if)# int vlan 110
N7K2-VDC-2(config-if)# ip add 10.110.1.3/24
N7K2-VDC-2(config-if)# int vlan 120
N7K2-VDC-2(config-if)# ip add 10.120.1.3/24
N7K2-VDC-2(config-if)# int vlan 130
N7K2-VDC-2(config-if)# ip add 10.130.1.3/24
N7K2-VDC-2(config-if)#
```

Lab 5:

Trunking with LACP and enabling UDLD

Tasks:

- Create a LACP etherchannel (use 10) on both Nexus 7000 VDC-2 on ports 7/1 and 8/1.
- Once created, check to make sure Spanning-Tree is operational on the newly created Ethernetchannel
- Set N7K1-VDC2 as root bridge and N7K2-VDC2 as secondary.
- Check to make sure that the correct switch is ROOT.
- Configure UDLD on Ports e7/1 and e8/1 on both switches

*First you should check the status of the interfaces to see if they are in use, routed, or in switch mode. Do you this with the **show interface eX/Y status**.*

```
N7K1-VDC-2# sh int e7/1 status
--------------------------------------------------------------------------------
Port           Name                Status    Vlan      Duplex  Speed   Type
--------------------------------------------------------------------------------
Eth7/1         --                  down      routed    full    auto    10g
N7K1-VDC-2#
N7K1-VDC-2# sh int e8/1 status
--------------------------------------------------------------------------------
Port           Name                Status    Vlan      Duplex  Speed   Type
--------------------------------------------------------------------------------
Eth8/1         --                  down      routed    full    auto    10g
```

*You will see that they have defauted to ROUTED mode. This is because when we ran through the setup we set the default interface to Layer 3 (L3). So, just like IOS you use the **switchport** command to convert them to switchport.*

```
N7K1-VDC-2(config)# int e7/1
N7K1-VDC-2(config-if)# switchport

N7K1-VDC-2# sh int e7/1 st
--------------------------------------------------------------------------------Port
Name                Status    Vlan      Duplex  Speed   Type
--------------------------------------------------------------------------------
Eth7/1         --                  down      1         full    auto    10g
N7K1-VDC-2#
```

*We can then set them to **switchport mode trunk**.*

```
N7K1-VDC-2(config-if)# switchport mode trunk
```

*Now, lets create the channel-group using LACP. Just like IOS, you use the **channel-group ## mode active** command.*

```
N7K1-tr(config-if)# channel-group 10 mode active
LACP process needs to be started before configuring active mode
N7K1-tr(config-if)#
```

*Notice, we need to enable the LACP feature before we can use ACTIVE mode. Just like any other feature, you use the **feature** command to enable.*

```
N7K1-tr(config-if)# exit
N7K1-tr(config)# feature lacp
N7K1-tr(config)# int e7/1
N7K1-tr(config-if)# channel-group 10 mode active
N7K1-tr(config-if)# no shutdown
```

Now, let us do the other port:

```
N7K1-tr(config)# int e8/1
N7K1-tr(config-if)# switchport
N7K1-tr(config-if)# switchport mode trunk
N7K1-tr(config-if)# channel-group 10 mode active
N7K1-tr(config-if)# no shutdown
```

Now let's set the port-channel into trunking mode as well

```
N7K1-tr(config-if)#
N7K1-tr(config-if)# int port-channel 10
N7K1-tr(config-if)# switchpo mode trunk
```

Now for the other switch:

```
N7K2-VDC-2(config)# feature lacp
N7K2-VDC-2(config)# int e7/1
N7K2-VDC-2(config-if)# switchport
N7K2-VDC-2(config-if)# switchp mode tru
N7K2-VDC-2(config-if)# channel-group 10 mode active
N7K2-VDC-2(config-if)# no shutdown
N7K2-VDC-2(config-if)# int e8/1
N7K2-VDC-2(config-if)# switchport
N7K2-VDC-2(config-if)# switchport mode tru
N7K2-VDC-2(config-if)# channel-group 10 mode active
N7K2-VDC-2(config-if)# no shutdown
N7K2-VDC-2(config-if)# int port-channel 10
N7K2-VDC-2(config-if)# switchport mode trunk
N7K2-VDC-2(config-if)#
```

To check the port-channel with the ***show port-channel summary*** *command:*

```
N7K1-VDC-2# sh port-channel summary
Flags:  D - Down        P - Up in port-channel (members)
        I - Individual  H - Hot-standby (LACP only)
        s - Suspended   r - Module-removed
        S - Switched    R - Routed
        U - Up (port-channel)
--------------------------------------------------------------------------------
Group Port-       Type     Protocol  Member Ports      Channel
--------------------------------------------------------------------------------
10    Po10(SU)    Eth      LACP      Eth7/1(P)    Eth8/1(P)
```

To check spanning-tree over the port-channel, use ***show spanning-tree root:***

```
N7K2-VDC-2# sh spanning-tree root
                                       Root  Hello Max Fwd
Vlan                   Root ID         Cost  Time  Age Dly  Root Port
----------------- -------------------- ----- ----- --- --- -----------------
VLAN0001         32769 0026.9807.95c2     0    2   20  15  This bridge is root
VLAN0100         32868 0026.9807.95c2     0    2   20  15  This bridge is root
VLAN0110         32878 0026.9807.95c2     0    2   20  15  This bridge is root
VLAN0120         32888 0026.9807.95c2     0    2   20  15  This bridge is root
VLAN0130         32898 0026.9807.95c2     0    2   20  15  This bridge is root
N7K2-VDC-2#

N7K1-VDC-2# sh spanning-tree root
                                        Root  Hello Max Fwd
Vlan                   Root ID          Cost  Time  Age Dly  Root Port
----------------- -------------------- ------ ----- --- ---  -----------------
VLAN0001         32769 0026.9807.95c2        1    2    20  15    port-channel10
VLAN0100         32868 0026.9807.95c2        1    2    20  15    port-channel10
VLAN0110         32878 0026.9807.95c2        1    2    20  15    port-channel10
VLAN0120         32888 0026.9807.95c2        1    2    20  15    port-channel10
```

```
VLAN0130         32898 0026.9807.95c2        1    2  20  15    port-channel10
N7K1-VDC-2#
```

To set the root, use the same commands that we use in IOS – ***spanning-tree vlan ### root primary/secondary***

```
N7K2-VDC-2(config)# spanning-tree vlan 100 root primary
N7K2-VDC-2(config)# spanning-tree vlan 110 root primary
N7K2-VDC-2(config)# spanning-tree vlan 120 root primary
N7K2-VDC-2(config)# spanning-tree vlan 130 root primary
N7K2-VDC-2(config)# sh spanning-tree root

                                       Root  Hello Max Fwd
Vlan                   Root ID         Cost  Time  Age Dly  Root Port
---------------- -------------------- ------ ----- --- ---  ----------------
VLAN0001         32769 0009.43ae.9ec0      1    2   20  15    port-channel20
VLAN0100         24676 0026.9807.95c2      0    2   20  15  This bridge is root
VLAN0110         24686 0026.9807.95c2      0    2   20  15  This bridge is root
VLAN0120         24696 0026.9807.95c2      0    2   20  15  This bridge is root
VLAN0130         24706 0026.9807.95c2      0    2   20  15  This bridge is root
N7K2-VDC-2(config)#
```

And on the other switch

```
N7K1-VDC-2(config)# spanning-tree vlan 130 root secondary
N7K1-VDC-2(config)# spanning-tree vlan 120 root secondary
N7K1-VDC-2(config)# spanning-tree vlan 110 root secondary
N7K1-VDC-2(config)# spanning-tree vlan 100 root secondary
```

To configure udld, use the command **udld** under the interface

```
N7K1-VDC-2(config)# int e7/1
N7K1-VDC-2(config-if)# udld
                       ^
% Invalid command at '^' marker.
```

As you can see, UDLD is not enabled. So, we need to enable the UDLD feature globally on the VDC

```
N7K1-VDC-2(config)# feature udld
```

Once enabled, we need to configure it under the interface:

```
N7K1-VDC-2(config)# int e7/1
N7K1-VDC-2(config-if)# udld ?
  aggressive  Enable UDLD aggressive mode for interface(s)
  disable     Disable UDLD for fiber interface(s)
  enable      Enable UDLD for non-fiber interface(s)
N7K1-VDC-2(config-if)# udld enable
```

Now for the other switch:

```
N7K2-VDC-2(config)# feature udld
N7K2-VDC-2(config)# int e7/1
N7K2-VDC-2(config-if)# udld enable
N7K2-VDC-2(config-if)# int e8/1
N7K2-VDC-2(config-if)# udld enable
N7K2-VDC-2(config-if)# exit
```

Now let's check to make sure UDLD is working using ***show udld neighbors***

```
N7K2-VDC-2(config)# show udld neighbors
Port             Device Name    Device ID   Port ID         Neighbor State
--------------------------------------------------------------------------
Ethernet7/1      JAF1414AKED    1           Ethernet7/1     bidirectional
Ethernet8/1      JAF1414AKED    1           Ethernet8/1     bidirectional
```

Lab 6:

Creating a vPC

Tasks:

- Check to make sure that vPC is supported on the hardware
- Create the following vPC domain 100 and assign the ports as follows

N7K1	N7K1	Type
7/1	7/1	vPC Peer (Po10)
8/2	8/2	vPC Peer (Po10)
3/1	3/1	vPC Keepalive

- The vPC keepalive should be between 10.10.10.1 and 10.10.10.2. Source the keepalive from the default VRF.

*Check to make sure your hardware supports vPC using the command **show hardware feature-capability***

```
N7K1-VDC-2# sh hardware feature-capability
Hardware Dependent Features:
   . = supported
   X = unsupported
                -- Module --
                 3   4   7   8
VPC              .   .   .   .
PVLAN Trunk      .   .   X   X
```

As you can see, MODs 3, 4, 7, and 8 all support vPC.

*Now to configure ip addresses on the interface, this is the same as any other IOS switch configuration. Just remember to **no shut** the interface.*

On N7K2:

```
N7K2-VDC-2(config-vpc-domain)# int e3/1
N7K2-VDC-2(config-if)# no sw
N7K2-VDC-2(config-if)# ip add 10.10.10.2 255.255.255.0
N7K2-VDC-2(config-if)# no shutdown
```

On N7K1:

```
N7K1-VDC-2(config)# int e3/1
N7K1-VDC-2(config-if)# no sw
N7K1-VDC-2(config-if)# ip add 10.10.10.1 255.255.255.0
N7K1-VDC-2(config-if)# no shut
```

*Now let's **ping** to make sure we have connectivity.*

```
N7K2-VDC-2# ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1): 56 data bytes
64 bytes from 10.10.10.1: icmp_seq=0 ttl=254 time=1.089 ms
64 bytes from 10.10.10.1: icmp_seq=1 ttl=254 time=0.592 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=254 time=0.474 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=254 time=0.474 ms
64 bytes from 10.10.10.1: icmp_seq=4 ttl=254 time=0.476 ms
```

```
--- 10.10.10.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.474/0.621/1.089 ms
N7K2-VDC-2#
```

*To create a vPC, you use the **vpc** command*

```
N7K1-VDC-2(config)# vpc ?
                     ^
% Invalid command at '^' marker.
```

As you can see, VPC is a feature that needs to be enabled

```
N7K1-VDC-2(config)# feature vpc
```

*Now to create the vPC domain using **vpc domain ##***

```
N7K1-VDC-2(config)# vpc domain 100
```

*And to check the vPC, do **show vpc brief***

```
N7K1-VDC-2(config-vpc-domain)# sh vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                   : 100
Peer status                     : peer link not configured
vPC keep-alive status           : Disabled
Configuration consistency status: failed
Configuration consistency reason: vPC peer-link does not exist
Type-2 consistency status       : failed
Type-2 consistency reason       : vPC peer-link does not exist
vPC role                        : none established
Number of vPCs configured       : 0
Peer Gateway                    : Disabled
Dual-active excluded VLANs      : -
N7K1-VDC-2(config-vpc-domain)#
```

*Before you can create a vPC between the switches, you must first configure the vPC peer-keepalive link under the **vpc domain** submenu. The peer link is configured using the **peer-keepalive** command set.*

```
N7K1-VDC-2(config)# vpc domain 100
N7K1-VDC-2(config-vpc-domain)# peer-keepalive destination 10.10.10.1 vrf default
source 10.10.10.2 vrf default
```

And on the other switch:

```
N7K2-VDC-2(config)# vpc domain 100
N7K2-VDC-2(config-vpc-domain)# peer-keepalive destination 10.10.10.2 vrf default
source 10.10.10.1 vrf default
```

*To check the status of the vPC peer, use **show vpc peer-keepalive***

```
N7K1-VDC-2(config-vpc-domain)# sh vpc peer-keepalive

vPC keep-alive status           : peer is alive
--Peer is alive for             : (18) seconds, (995) msec
--Send status                   : Success
--Last send at                  : 2010.06.14 21:14:52 964 ms
--Sent on interface             : Eth3/1
--Receive status                : Success
--Last receive at               : 2010.06.14 21:14:53 68 ms
--Received on interface         : Eth3/1
--Last update from peer         : (0) seconds, (805) msec

vPC Keep-alive parameters
--Destination                   : 10.10.10.2
```

```
--Keepalive interval            : 1000 msec
--Keepalive timeout             : 5 seconds
--Keepalive hold timeout        : 3 seconds
--Keepalive vrf                 : default
--Keepalive udp port            : 3200
--Keepalive tos                 : 192
N7K1-VDC-2(config-vpc-domain)#
```

To create the vpc peer-link you need to be under the port-channel interface.
*The command to designate this port-channel as a vpc peer is **vpc peer-link***

```
N7K2-VDC-2(config-vpc-domain)# int port-channel 10
N7K2-VDC-2(config-if)# vpc peer-link
Please note that spanning tree port type is changed to "network" port type on vPC
peer-link.
This will enable spanning tree Bridge Assurance on vPC peer-link provided the STP
Bridge Assurance
(which is enabled by default) is not disabled.
N7K2-VDC-2(config-if)#
```

And on the other switch:

```
N7K1-VDC-2(config-vpc-domain)# int port-channel 10
N7K1-VDC-2(config-if)# vpc peer-link
Please note that spanning tree port type is changed to "network" port type on vPC
peer-link.
This will enable spanning tree Bridge Assurance on vPC peer-link provided the STP
Bridge Assurance
(which is enabled by default) is not disabled.
N7K1-VDC-2(config-if)#
```

*To check the status of the vpc, use **show vpc brief***

```
N7K1-VDC-2(config-if)# sh vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                   : 100
Peer status                     : peer adjacency formed ok
vPC keep-alive status           : peer is alive
Configuration consistency status: success
Type-2 consistency status       : success
vPC role                        : secondary
Number of vPCs configured       : 0
Peer Gateway                    : Disabled
Dual-active excluded VLANs      : -

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po10   up     1,100,110,120,130
N7K1-VDC-2(config-if)#
```

Lab 7:

Trunking to another switch utilizing vPC on the Nexus 7000

Task:

- There is a Cisco 2950T pre-configured and attached to the Nexus 7000 switches on ports 4/1. Configure a single virtual port-channel (20) to the 2950T using e4/1 on both switches.
- Once created, ping the remote 2950T at IP address 10.100.1.254.

*First you should check the interface using **show int ethernet x/y***

```
N7K1-VDC-2# sh int e4/1
                    ^
Invalid range at '^' marker.
N7K1-VDC-2#
```

*You will notice, that you do not have an interface 4/1, that interface is still in the main VDC. So, let us **switchback** to the main VDC and assigned port 4/1 to our VDC.*

```
N7K1-VDC-2# switchback
N7K1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
N7K1(config)# vdc VDC-2
N7K1(config-vdc)# allocate interface e4/1
Moving ports will cause all config associated to them in source vdc to be removed. Are
you sure you want to move the ports (y/n)?  [yes] y
N7K1(config-vdc)# exit
N7K1(config)# switchto vdc VDC-2
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2010, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
N7K1-VDC-2# sh int e4/1 brief

--------------------------------------------------------------------------------
Ethernet      VLAN   Type Mode   Status  Reason                   Speed     Port
Interface                                                                   Ch #
--------------------------------------------------------------------------------
Eth4/1        --     eth  routed down    Administratively down      auto(D) --
N7K1-VDC-2#
```

*Now we can configure the interface trunk and channel like we normally would. The difference here is that we need to configure a **vpc** as well. That is done under the **port-channel** interface using the **vpc ##** command.*

```
N7K1-VDC-2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
N7K1-VDC-2(config)# int ethernet 4/1
N7K1-VDC-2(config-if)# switchport
N7K1-VDC-2(config-if)# switchport mode trunk
N7K1-VDC-2(config-if)# channel-group 20 mode active
N7K1-VDC-2(config-if)# no shutdown
N7K1-VDC-2(config-if)# int port-channel 20
```

```
N7K1-VDC-2(config-if)# vpc 20
```

To check the status of the vPC once created use the ***show vpc ##***

```
N7K1-VDC-2# sh vpc 20
vPC status
----------------------------------------------------------------------
id   Port   Status Consistency Reason                     Active vlans
--   ----   ------ ----------- -------------------------- ------------
20   Po20   up     success     success                    1,100,110,1
                                                          20,130
```

And to ***ping*** *to test connectivity*

```
N7K1-VDC-2# ping 10.100.1.254
PING 10.100.1.254 (10.100.1.254): 56 data bytes
64 bytes from 10.100.1.254: icmp_seq=0 ttl=254 time=2.079 ms
64 bytes from 10.100.1.254: icmp_seq=1 ttl=254 time=1.708 ms
64 bytes from 10.100.1.254: icmp_seq=2 ttl=254 time=1.719 ms
64 bytes from 10.100.1.254: icmp_seq=3 ttl=254 time=1.722 ms
64 bytes from 10.100.1.254: icmp_seq=4 ttl=254 time=1.922 ms

--- 10.100.1.254 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 1.708/1.83/2.079 ms
```

Now to check to see what the 2950T sees:

```
N7KLab_2950#sh int port-channel 1 etherchannel
Port-channel1   (Primary aggregator)

Age of the Port-channel   = 2d:22h:46m:59s
Logical slot/port   = 1/0           Number of ports = 2
HotStandBy port = null
Port state          = Port-channel Ag-Inuse
Protocol            =   LACP

Ports in the Port-channel:

Index   Load   Port     EC state        No of bits
------+------+------+------------------+-----------
  0     00     Gi0/1    Active             0
  0     00     Gi0/2    Active             0

Time since last port bundled:    0d:00h:54m:41s    Gi0/1


N7KLab_2950#
```

Lab 8:

Configure HSRP

Task:

- Configure HSRP between N7K1-VDC2 and N7K2-VDC on VLANs 100, 120, 130, and 140
- Use the VLAN number for the group number
- N7K1 should be primary for VLANs 100 and 120
- N7K2 should be primary for VLANs 110 and 130
- Configure the HSRP so that the correct switch is always active, even after a reboot
- Use the appropriate .1 for the HSRP address on the VLAN
- Used MD5 authentication with a password of NEXUS
- Once configured, ping the HSRP address from the 2950T.

Lets create a standby HSRP address under VLAN 100 first. Is the command ***standby*** *or* ***hsrp*** *in NX-OS?*

```
7K1-VDC-2(config)# int vlan 100
N7K1-VDC-2(config-if)# standby
                        ^
% Invalid command at '^' marker.
N7K1-VDC-2(config-if)# hsrp?
                       ^
% Invalid command at '^' marker.
N7K1-VDC-2(config-if)# exit
```

You will notice that standby is not an option and nor is HSRP. This is because HSRP is a feature that needs to be enabled.

```
N7K1-VDC-2(config)# feature hsrp
```

Now that HSRP is enabled, you are able to configure it. The command to configure hsrp in NX-OS is ***hsrp.*** *All HSRP commands and variables are configured under this sub-menu. IP, priority, and such are all configured similar to IOS.*

```
N7K1-VDC-2(config)# int vlan 100
N7K1-VDC-2(config-if)# hsrp 100
N7K1-VDC-2(config-if-hsrp)# ip 10.100.1.1
N7K1-VDC-2(config-if-hsrp)# priority 110
N7K1-VDC-2(config-if-hsrp)# authentication md5 key-string NEXUS
N7K1-VDC-2(config-if-hsrp)# preempt

N7K1-VDC-2(config-if-hsrp)# int vlan 110
N7K1-VDC-2(config-if)# hsrp 110
N7K1-VDC-2(config-if-hsrp)# ip 10.110.1.1
N7K1-VDC-2(config-if-hsrp)# prio 90
N7K1-VDC-2(config-if-hsrp)# authen md5 key-st NEXUS

N7K1-VDC-2(config-if-hsrp)# int vlan 120
N7K1-VDC-2(config-if)# hsrp 120
N7K1-VDC-2(config-if-hsrp)# ip 10.120.1.1
N7K1-VDC-2(config-if-hsrp)# prio 110
N7K1-VDC-2(config-if-hsrp)# auth md5 key-st NEXUS
N7K1-VDC-2(config-if-hsrp)# pre

N7K1-VDC-2(config-if-hsrp)# int vlan 130
```

```
N7K1-VDC-2(config-if)# hsrp 130
N7K1-VDC-2(config-if-hsrp)# ip 10.130.1.1
N7K1-VDC-2(config-if-hsrp)# prio 90
N7K1-VDC-2(config-if-hsrp)# auth md5 key-st NEXUS
N7K1-VDC-2(config-if-hsrp)# int vlan 120
```

And we will do the same on the other switch

```
N7K2-VDC-2#
N7K2-VDC-2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
N7K2-VDC-2(config)# feature hsrp
N7K2-VDC-2(config)# int vlan 100
N7K2-VDC-2(config-if)# hsrp 100
N7K2-VDC-2(config-if-hsrp)# ip 10.100.1.1
N7K2-VDC-2(config-if-hsrp)# prio 90
N7K2-VDC-2(config-if-hsrp)# authentication md5 key-st NEXUS
N7K2-VDC-2(config-if-hsrp)# exit
N7K2-VDC-2(config-if)# int vlan 110
N7K2-VDC-2(config-if)# hsrp 110
N7K2-VDC-2(config-if-hsrp)# ip 10.110.1.1
N7K2-VDC-2(config-if-hsrp)# pri 110
N7K2-VDC-2(config-if-hsrp)# pre
N7K2-VDC-2(config-if-hsrp)# authentication md5 key-st NEXUS
N7K2-VDC-2(config-if-hsrp)# int vlan 120
N7K2-VDC-2(config-if)# hsrp 120
N7K2-VDC-2(config-if-hsrp)# ip 10.120.1.1
N7K2-VDC-2(config-if-hsrp)# prio 90
N7K2-VDC-2(config-if-hsrp)# authentication md5 key-string NEXUS
N7K2-VDC-2(config-if-hsrp)# int vlan 130
N7K2-VDC-2(config-if)# hsrp 130
N7K2-VDC-2(config-if-hsrp)# ip 10.130.1.1
N7K2-VDC-2(config-if-hsrp)# priority 110
N7K2-VDC-2(config-if-hsrp)# preempt
N7K2-VDC-2(config-if-hsrp)# authentication md5 key-string NEXUS
N7K2-VDC-2(config-if-hsrp)# int vlan 120
```

And to check the status of HSRP use the ***show hsrp*** *command*

```
N7K2-VDC-2# sh hsrp
Vlan100 - Group 100 (HSRP-V1) (IPv4)
  Local state is Standby, priority 90 (Cfged 90)
    Forwarding threshold(for VPC), lower: 1 upper: 90
  Hellotime 3 sec, holdtime 10 sec
  Next hello sent in 2.259000 sec(s)
  Virtual IP address is 10.100.1.1 (Cfged)
  Active router is 10.100.1.2, priority 110 expires in 9.684000 sec(s)
  Standby router is local
  Authentication MD5, key-string "NEXUS"
  Virtual mac address is 0000.0c07.ac64 (Default MAC)
  1 state changes, last state change 00:02:04
  IP redundancy name is hsrp-Vlan100-100 (default)

Vlan110 - Group 110 (HSRP-V1) (IPv4)
  Local state is Active, priority 110 (Cfged 110), may preempt
    Forwarding threshold(for VPC), lower: 1 upper: 110
  Hellotime 3 sec, holdtime 10 sec
  Next hello sent in 0.878000 sec(s)
  Virtual IP address is 10.110.1.1 (Cfged)
  Active router is local
  Standby router is 10.110.1.2
  Authentication MD5, key-string "NEXUS"
  Virtual mac address is 0000.0c07.ac6e (Default MAC)
  1 state changes, last state change 00:01:53
  IP redundancy name is hsrp-Vlan110-110 (default)

Vlan120 - Group 120 (HSRP-V1) (IPv4)
  Local state is Standby, priority 90 (Cfged 90)
    Forwarding threshold(for VPC), lower: 1 upper: 90
  Hellotime 3 sec, holdtime 10 sec
  Next hello sent in 2.718000 sec(s)
  Virtual IP address is 10.120.1.1 (Cfged)
  Active router is 10.120.1.2, priority 110 expires in 8.393000 sec(s)
  Standby router is local
  Authentication MD5, key-string "NEXUS"
  Virtual mac address is 0000.0c07.ac78 (Default MAC)
  1 state changes, last state change 00:01:21
  IP redundancy name is hsrp-Vlan120-120 (default)

Vlan130 - Group 130 (HSRP-V1) (IPv4)
  Local state is Active, priority 110 (Cfged 110), may preempt
    Forwarding threshold(for VPC), lower: 1 upper: 110
  Hellotime 3 sec, holdtime 10 sec
  Next hello sent in 0.168000 sec(s)
  Virtual IP address is 10.130.1.1 (Cfged)
  Active router is local
  Standby router is 10.130.1.2
  Authentication MD5, key-string "NEXUS"
  Virtual mac address is 0000.0c07.ac82 (Default MAC)
  1 state changes, last state change 00:01:12
  IP redundancy name is hsrp-Vlan130-130 (default)

N7K2-VDC-2#
```

From the 2950 you should now be able to ***ping*** *the gateway HSRP IP addresses on the Nexus 7000.*

```
N7KLab_2950#ping 10.100.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
N7KLab_2950#ping 10.110.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.110.1.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/2/4 ms
N7KLab_2950#ping 10.120.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.120.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/202/1004 ms
N7KLab_2950#ping 10.130.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.130.1.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/4 ms
N7KLab_2950#
```

Lab 9:

Assigning ports to existing VDCs and configuring a L3 IP address

Task:

- There is a Cisco router connected to ports 4/2 on both switches. Allocate this port in the proper VDC.
- Configure N7K1-VDC-2 e4/2 with an IP address of 54.2.2.1/24
- Ping the router that is connected, IP of the router is 54.2.2.254

First you will need to switchback and allocate the e4/2 interface to your VDC on both switches.

```
N7K1-VDC-1# switchback
N7K1# config t
N7K1(config)# vdc vdc-2
N7K1(config-vdc)# allocate interface e4/2

Moving ports will cause all config associated to them in source vdc to be removed. Are
you sure you want to move the ports (y/n)?  [yes] y
N7K1(config-vdc)# exit
N7K1(config)# exit
N7K1# switcht vdc VDC-2
```

```
N7K2-VDC-2# switchback
N7K2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
N7K2(config)# vdc vdc-2
N7K2(config-vdc)# allocate interface e4/2
Moving ports will cause all config associated to them in source vdc to be removed. Are
you sure you want to move the ports (y/n)?  [yes] y
N7K2(config-vdc)# exit
N7K2(config)# exit
N7K2# switcht vdc VDC-2
```

Once allocated, you can configure it with an IP address

```
N7K1-VDC-2(config-if)# int e4/2
N7K1-VDC-2(config-if)# ip add 54.2.2.1 255.255.255.0
N7K1-VDC-2(config-if)# no shut
```

Once configured, you should be able to ***ping*** *the remote router*

```
N7K1-VDC-2# ping 54.2.2.254
PING 54.2.2.254 (54.2.2.254): 56 data bytes
Request 0 timed out
64 bytes from 54.2.2.254: icmp_seq=1 ttl=254 time=1.141 ms
64 bytes from 54.2.2.254: icmp_seq=2 ttl=254 time=0.819 ms
64 bytes from 54.2.2.254: icmp_seq=3 ttl=254 time=0.846 ms
64 bytes from 54.2.2.254: icmp_seq=4 ttl=254 time=0.847 ms

--- 54.2.2.254 ping statistics ---
5 packets transmitted, 4 packets received, 20.00% packet loss
round-trip min/avg/max = 0.819/0.913/1.141 ms
N7K1-VDC-2#
```

Lab 10:

Basic routing with RIP

Task:

- Configure RIP for network 54.2.2.0/24 and all VLAN interfaces on the Nexus
- Use a process name of NEXUS
- Ping the a remote IP address of 212.18.3.1

Let's configure RIP using ***router rip*** *command.*

```
N7K1-VDC-2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
N7K1-VDC-2(config)# router rip
                         ^
% Invalid command at '^' marker.
```

Again, this is a ***feature*** *that needs to be enabled.*

```
N7K1-VDC-2(config)# feature rip
```

Once enabled, we can configure it but notice it needs a process name now.

```
N7K1-VDC-2(config)# router rip
                              ^
% Incomplete command at '^' marker.

N7K1-VDC-2(config)# router rip ?
  WORD                         Process ID (Max Size 20)

N7K1-VDC-2(config)# router rip NEXUS
```

Once you are in the RIP submenu, the normal ***network*** *commands do not work. Routing protocols are now assigned on a per-interface level*

```
N7K1-VDC-2(config-router)# network 54.0.0.0
                            ^
% Invalid command at '^' marker.
N7K1-VDC-2(config-router)
N7K1-VDC-2(config-router)# int e4/2
N7K1-VDC-2(config-if)# router rip NEXUS
N7K1-VDC-2(config-router)# int vlan 100
N7K1-VDC-2(config-if)# router rip NEXUS
N7K1-VDC-2(config-router)# int vlan 110
N7K1-VDC-2(config-if)# router rip NEXUS
N7K1-VDC-2(config-router)# int vlan 120
N7K1-VDC-2(config-if)# router rip NEXUS
N7K1-VDC-2(config-router)# int vlan 130
N7K1-VDC-2(config-if)# router rip NEXUS
N7K1-VDC-2(config-router)#
```

To look at the routing table, just like IOS, you use ***show ip route***

```
N7K1-VDC-2# sh ip route
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

10.10.10.0/24, ubest/mbest: 1/0, attached
    *via 10.10.10.1, Eth3/1, [0/0], 02:45:23, direct
```

```
10.10.10.1/32, ubest/mbest: 1/0, attached
    *via 10.10.10.1, Eth3/1, [0/0], 02:45:23, local
10.100.1.0/24, ubest/mbest: 1/0, attached
    *via 10.100.1.2, Vlan100, [0/0], 02:36:45, direct
10.100.1.1/32, ubest/mbest: 1/0
    *via 10.100.1.1, Vlan100, [0/0], 00:26:56, hsrp
10.100.1.2/32, ubest/mbest: 1/0, attached
    *via 10.100.1.2, Vlan100, [0/0], 02:36:45, local
10.110.1.0/24, ubest/mbest: 1/0, attached
    *via 10.110.1.2, Vlan110, [0/0], 02:36:45, direct
10.110.1.1/32, ubest/mbest: 1/0
    *via 10.110.1.1, Vlan110, [0/0], 00:23:35, hsrp
10.110.1.2/32, ubest/mbest: 1/0, attached
    *via 10.110.1.2, Vlan110, [0/0], 02:36:45, local
10.120.1.0/24, ubest/mbest: 1/0, attached
    *via 10.120.1.2, Vlan120, [0/0], 01:15:09, direct
10.120.1.1/32, ubest/mbest: 1/0
    *via 10.120.1.1, Vlan120, [0/0], 00:26:32, hsrp
10.120.1.2/32, ubest/mbest: 1/0, attached
    *via 10.120.1.2, Vlan120, [0/0], 01:15:09, local
10.130.1.0/24, ubest/mbest: 1/0, attached
    *via 10.130.1.2, Vlan130, [0/0], 01:15:07, direct
10.130.1.1/32, ubest/mbest: 1/0
    *via 10.130.1.1, Vlan130, [0/0], 00:22:54, hsrp
10.130.1.2/32, ubest/mbest: 1/0, attached
    *via 10.130.1.2, Vlan130, [0/0], 01:15:07, local
54.2.2.0/24, ubest/mbest: 1/0, attached
    *via 54.2.2.1, Eth4/2, [0/0], 00:04:02, direct
54.2.2.1/32, ubest/mbest: 1/0, attached
    *via 54.2.2.1, Eth4/2, [0/0], 00:04:02, local
212.18.0.0/24, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [120/2], 00:03:01, rip-NEXUS, rip
212.18.1.0/24, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [120/2], 00:03:01, rip-NEXUS, rip
212.18.2.0/24, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [120/2], 00:03:01, rip-NEXUS, rip
212.18.3.0/24, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [120/2], 00:03:01, rip-NEXUS, rip
```

*Now we can **ping** remote host 212.18.3.1*

```
N7K1-VDC-2# ping 212.18.3.1
PING 212.18.3.1 (212.18.3.1): 56 data bytes
64 bytes from 212.18.3.1: icmp_seq=0 ttl=254 time=1.336 ms
64 bytes from 212.18.3.1: icmp_seq=1 ttl=254 time=0.878 ms
64 bytes from 212.18.3.1: icmp_seq=2 ttl=254 time=0.846 ms
64 bytes from 212.18.3.1: icmp_seq=3 ttl=254 time=0.848 ms
64 bytes from 212.18.3.1: icmp_seq=4 ttl=254 time=0.851 ms

--- 212.18.3.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.846/0.951/1.336 ms
N7K1-VDC-2#
```

Lab 11:

Basic routing with OSPF

Tasks:

- Disable RIP before proceeding
- Configure OSPF process ID 1
- Assign the 54.2.2.0/24 subnet in Area 0 as well as all VLAN interfaces
- To test OSPF, ping 20.11.0.1 from the Nexus

To disable RIP, just use the ***no feature rip*** *command. This will disable RIP and remove all related commands.*

```
N7K1-VDC-2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
N7K1-VDC-2(config)# no feature rip
N7K1-VDC-2(config)#
```

To configure OSPF you will need to enable the ***feature*** *as well. If you do not enable it, you will not be able to configure it.*

```
N7K1-VDC-2(config)# router ospf 1
                         ^

% Invalid command at '^' marker.
N7K1-VDC-2(config)# feature ospf
N7K1-VDC-2(config)# router ospf 1
N7K1-VDC-2(config-router)# exit
```

Again, all routing protocols are configured under the interface now.

```
N7K1-VDC-2(config)# int e4/2
N7K1-VDC-2(config-if)# ip router ospf 1 area 0
N7K1-VDC-2(config)# int vlan 100
N7K1-VDC-2(config-if)# ip router ospf 1 area 0
N7K1-VDC-2(config-if)# int vlan 110
N7K1-VDC-2(config-if)# ip router ospf 1 area 0
N7K1-VDC-2(config-if)# int vlan 120
N7K1-VDC-2(config-if)# ip router ospf 1 area 0
N7K1-VDC-2(config-if)# int vlan 130
N7K1-VDC-2(config-if)# ip router ospf 1 area 0
N7K1-VDC-2(config-if)#
```

Look at the routing table to see what routes are there using the same IOS commands ***show ip route***

```
N7K1-VDC-2# sh ip route
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

10.10.10.0/24, ubest/mbest: 1/0, attached
    *via 10.10.10.1, Eth3/1, [0/0], 03:42:45, direct
10.10.10.1/32, ubest/mbest: 1/0, attached
    *via 10.10.10.1, Eth3/1, [0/0], 03:42:45, local
10.100.1.0/24, ubest/mbest: 1/0, attached
    *via 10.100.1.2, Vlan100, [0/0], 03:34:07, direct
10.100.1.1/32, ubest/mbest: 1/0
    *via 10.100.1.1, Vlan100, [0/0], 01:24:18, hsrp
10.100.1.2/32, ubest/mbest: 1/0, attached
```

```
    *via 10.100.1.2, Vlan100, [0/0], 03:34:07, local
10.110.1.0/24, ubest/mbest: 1/0, attached
    *via 10.110.1.2, Vlan110, [0/0], 03:34:07, direct
10.110.1.1/32, ubest/mbest: 1/0
    *via 10.110.1.1, Vlan110, [0/0], 01:20:57, hsrp
10.110.1.2/32, ubest/mbest: 1/0, attached
    *via 10.110.1.2, Vlan110, [0/0], 03:34:07, local
10.120.1.0/24, ubest/mbest: 1/0, attached
    *via 10.120.1.2, Vlan120, [0/0], 02:12:31, direct
10.120.1.1/32, ubest/mbest: 1/0
    *via 10.120.1.1, Vlan120, [0/0], 01:23:54, hsrp
10.120.1.2/32, ubest/mbest: 1/0, attached
    *via 10.120.1.2, Vlan120, [0/0], 02:12:31, local
10.130.1.0/24, ubest/mbest: 1/0, attached
    *via 10.130.1.2, Vlan130, [0/0], 02:12:29, direct
10.130.1.1/32, ubest/mbest: 1/0
    *via 10.130.1.1, Vlan130, [0/0], 01:20:16, hsrp
10.130.1.2/32, ubest/mbest: 1/0, attached
    *via 10.130.1.2, Vlan130, [0/0], 02:12:29, local
20.0.0.1/32, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [110/401], 00:01:29, ospf-1, intra
20.1.0.1/32, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [110/401], 00:01:29, ospf-1, intra
20.2.0.1/32, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [110/401], 00:01:29, ospf-1, intra
20.3.0.1/32, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [110/401], 00:01:29, ospf-1, intra
20.4.0.1/32, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [110/401], 00:01:29, ospf-1, intra
20.5.0.1/32, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [110/401], 00:01:29, ospf-1, intra
20.6.0.1/32, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [110/401], 00:01:29, ospf-1, intra
20.7.0.1/32, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [110/401], 00:01:29, ospf-1, intra
20.8.0.1/32, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [110/401], 00:01:29, ospf-1, intra
20.9.0.1/32, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [110/401], 00:01:29, ospf-1, intra
20.10.0.1/32, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [110/401], 00:01:29, ospf-1, intra
20.11.0.1/32, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [110/401], 00:01:29, ospf-1, intra
54.2.2.0/24, ubest/mbest: 1/0, attached
    *via 54.2.2.1, Eth4/2, [0/0], 01:01:24, direct
54.2.2.1/32, ubest/mbest: 1/0, attached
    *via 54.2.2.1, Eth4/2, [0/0], 01:01:24, local
```

And now you should be able to ***ping 20.11.0.1***

```
N7K1-VDC-2# ping 20.11.0.1
PING 20.11.0.1 (20.11.0.1): 56 data bytes
64 bytes from 20.11.0.1: icmp_seq=0 ttl=254 time=1.266 ms
64 bytes from 20.11.0.1: icmp_seq=1 ttl=254 time=0.849 ms
64 bytes from 20.11.0.1: icmp_seq=2 ttl=254 time=0.845 ms
64 bytes from 20.11.0.1: icmp_seq=3 ttl=254 time=0.849 ms
64 bytes from 20.11.0.1: icmp_seq=4 ttl=254 time=0.85 ms

--- 20.11.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.845/0.931/1.266 ms
N7K1-VDC-2#
```

Lab 12:

Basic routing with EIGRP

Tasks:

- Remove OSPF conf0iguration from the switch
- Configure EIGRP AS 10
- Enable EIGRP for the 54.2.2.0/24 network and all VLAN interface subnets
- To test connectivity, PING 200.0.3.1

To disable OSPF, just use the ***no feature ospf*** *command. This will disable OSPF and remove all related commands.*

```
N7K1-VDC-2(config)# no feature ospf
```

Now we can configure EIGRP, but as like all else, you need to enable the ***feature.*** *Hopefully by now you are realizing that most things are features that need to be enbled.*

```
N7K1-VDC-2(config)# router eigrp
                         ^

% Invalid command at '^' marker.
N7K1-VDC-2(config)# feature eigrp
```

Once enabled, we can configure EIGRP, but as with all other routing protocols, this is done under the interface. There is no ***network*** *command anymore.*

```
N7K1-VDC-2(config)# router eigrp 10
N7K1-VDC-2(config-router)# network ?
% Incomplete command

N7K1-VDC-2(config)# int e4/2
N7K1-VDC-2(config-if)# ip router eigrp 10
N7K1-VDC-2(config-if)# int vlan 100
N7K1-VDC-2(config-if)# ip router eigrp 10
N7K1-VDC-2(config-if)# int vlan 110
N7K1-VDC-2(config-if)# ip router eigrp 10
N7K1-VDC-2(config-if)# int vlan 120
N7K1-VDC-2(config-if)# ip router eigrp 10
N7K1-VDC-2(config-if)# int vlan 130
N7K1-VDC-2(config-if)# ip router eigrp 10
N7K1-VDC-2(config-if)#
```

Let us look at the routing table.

```
N7K1-VDC-2# sh ip route
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

10.10.10.0/24, ubest/mbest: 1/0, attached
    *via 10.10.10.1, Eth3/1, [0/0], 03:46:29, direct
10.10.10.1/32, ubest/mbest: 1/0, attached
    *via 10.10.10.1, Eth3/1, [0/0], 03:46:29, local
10.100.1.0/24, ubest/mbest: 1/0, attached
    *via 10.100.1.2, Vlan100, [0/0], 03:37:51, direct
10.100.1.1/32, ubest/mbest: 1/0
    *via 10.100.1.1, Vlan100, [0/0], 01:28:02, hsrp
10.100.1.2/32, ubest/mbest: 1/0, attached
```

```
    *via 10.100.1.2, Vlan100, [0/0], 03:37:51, local
10.110.1.0/24, ubest/mbest: 1/0, attached
    *via 10.110.1.2, Vlan110, [0/0], 03:37:51, direct
10.110.1.1/32, ubest/mbest: 1/0
    *via 10.110.1.1, Vlan110, [0/0], 01:24:41, hsrp
10.110.1.2/32, ubest/mbest: 1/0, attached
    *via 10.110.1.2, Vlan110, [0/0], 03:37:51, local
10.120.1.0/24, ubest/mbest: 1/0, attached
    *via 10.120.1.2, Vlan120, [0/0], 02:16:15, direct
10.120.1.1/32, ubest/mbest: 1/0
    *via 10.120.1.1, Vlan120, [0/0], 01:27:38, hsrp
10.120.1.2/32, ubest/mbest: 1/0, attached
    *via 10.120.1.2, Vlan120, [0/0], 02:16:15, local
10.130.1.0/24, ubest/mbest: 1/0, attached
    *via 10.130.1.2, Vlan130, [0/0], 02:16:13, direct
10.130.1.1/32, ubest/mbest: 1/0
    *via 10.130.1.1, Vlan130, [0/0], 01:24:00, hsrp
10.130.1.2/32, ubest/mbest: 1/0, attached
    *via 10.130.1.2, Vlan130, [0/0], 02:16:13, local
54.2.2.0/24, ubest/mbest: 1/0, attached
    *via 54.2.2.1, Eth4/2, [0/0], 01:05:08, direct
54.2.2.1/32, ubest/mbest: 1/0, attached
    *via 54.2.2.1, Eth4/2, [0/0], 01:05:08, local
200.0.0.0/24, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [90/153856], 00:00:11, eigrp-10, internal
200.0.1.0/24, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [90/153856], 00:00:11, eigrp-10, internal
200.0.2.0/24, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [90/153856], 00:00:11, eigrp-10, internal
200.0.3.0/24, ubest/mbest: 1/0
    *via 54.2.2.254, Eth4/2, [90/153856], 00:00:11, eigrp-10, internal
N7K1-VDC-2#
```

We should now be able to ***ping 200.0.3.1***

```
N7K1-VDC-2# ping 200.0.3.1
PING 200.0.3.1 (200.0.3.1): 56 data bytes
64 bytes from 200.0.3.1: icmp_seq=0 ttl=254 time=1.358 ms
64 bytes from 200.0.3.1: icmp_seq=1 ttl=254 time=0.964 ms
64 bytes from 200.0.3.1: icmp_seq=2 ttl=254 time=0.846 ms
64 bytes from 200.0.3.1: icmp_seq=3 ttl=254 time=0.849 ms
64 bytes from 200.0.3.1: icmp_seq=4 ttl=254 time=0.85 ms

--- 200.0.3.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.846/0.973/1.358 ms
N7K1-VDC-2#
```

Lab 13:

Routing with RIP and authentication

Tasks:

- Disable EIGRP and shutdown interface e4/2 N7K1-VDC-2 and remove the IP address configuration
- Configure N7K2-VDC-2 e4/2 with an IP address of 54.1.1.1/24
- Configure RIP on the switch, using NEXUS as the domain
- Enable RIP for network 54.0.0.0 and all VLAN interfaces
- The router attached to e4/2 is expecting MD5 authentication with a password of CISCO
- Once configured, ping 212.18.3.1 to test connectivity

Lets remove the EIGRP feature and IP address. To remove the EIGRP feature, just like before use the ***no feature eigrp*** *command. Then continue to remove the IP and* ***shutdown*** *the e4/2 interface.*

```
N7K1-VDC-2(config)# no feature eigrp
N7K1-VDC-2(config)# int e4/2
N7K1-VDC-2(config-if)# no ip add
N7K1-VDC-2(config-if)# shut
```

Now on the other switch, lets configure the IP address like we do in IOS

```
N7K2-VDC-2(config)# int e4/2
N7K2-VDC-2(config-if)# ip add 54.1.1.1 255.255.255.0
N7K2-VDC-2(config-if)# no shutdown
N7K2-VDC-2(config-if)# ^Z
N7K2-VDC-2#
```

Now we need to enable the ***feature*** *and assign the interfaces to the RIP process*

```
N7K2-VDC-2(config)# feature rip
N7K2-VDC-2(config)# router rip NEXUS
N7K2-VDC-2(config)# int e4/2
N7K2-VDC-2(config-if)# ip router rip NEXUS
N7K2-VDC-2(config-if)# int vlan 100
N7K2-VDC-2(config-if)# ip router rip NEXUS
N7K2-VDC-2(config-if)# int vlan 110
N7K2-VDC-2(config-if)# ip router rip NEXUS
N7K2-VDC-2(config-if)# int vlan 120
N7K2-VDC-2(config-if)# ip router rip NEXUS
N7K2-VDC-2(config-if)# int vlan 130
N7K2-VDC-2(config-if)# ip router rip NEXUS
```

With RIP, you need a ***key-chain*** *to configure authentication. This is just like IOS and applied to the interface that needs authentication.*

```
N7K2-VDC-2(config-if)# exit
N7K2-VDC-2(config)# key chain RIP
N7K2-VDC-2(config-keychain)# key 1
N7K2-VDC-2(config-keychain-key)# key-string CISCO
7K2-VDC-2(config)# int e4/2
N7K2-VDC-2(config-if)# ip rip authentication mode md5
N7K2-VDC-2(config-if)# ip rip authentication key-chain RIP
N7K2-VDC-2(config-if)# ^Z
```

Let's look at the route table like we do in IOS

```
N7K2-VDC-2# sh ip route
```

```
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

10.10.10.0/24, ubest/mbest: 1/0, attached
    *via 10.10.10.2, Eth3/1, [0/0], 04:14:31, direct
10.10.10.2/32, ubest/mbest: 1/0, attached
    *via 10.10.10.2, Eth3/1, [0/0], 04:14:31, local
10.100.1.0/24, ubest/mbest: 1/0, attached
    *via 10.100.1.3, Vlan100, [0/0], 04:05:54, direct
10.100.1.1/32, ubest/mbest: 1/0
    *via 10.100.1.1, Vlan100, [0/0], 01:53:25, hsrp
10.100.1.3/32, ubest/mbest: 1/0, attached
    *via 10.100.1.3, Vlan100, [0/0], 04:05:54, local
10.110.1.0/24, ubest/mbest: 1/0, attached
    *via 10.110.1.3, Vlan110, [0/0], 04:05:54, direct
10.110.1.1/32, ubest/mbest: 1/0
    *via 10.110.1.1, Vlan110, [0/0], 01:52:54, hsrp
10.110.1.3/32, ubest/mbest: 1/0, attached
    *via 10.110.1.3, Vlan110, [0/0], 04:05:54, local
10.120.1.0/24, ubest/mbest: 1/0, attached
    *via 10.120.1.3, Vlan120, [0/0], 04:05:54, direct
10.120.1.1/32, ubest/mbest: 1/0
    *via 10.120.1.1, Vlan120, [0/0], 01:52:42, hsrp
10.120.1.3/32, ubest/mbest: 1/0, attached
    *via 10.120.1.3, Vlan120, [0/0], 04:05:54, local
10.130.1.0/24, ubest/mbest: 1/0, attached
    *via 10.130.1.3, Vlan130, [0/0], 04:05:54, direct
10.130.1.1/32, ubest/mbest: 1/0
    *via 10.130.1.1, Vlan130, [0/0], 01:52:13, hsrp
10.130.1.3/32, ubest/mbest: 1/0, attached
    *via 10.130.1.3, Vlan130, [0/0], 04:05:54, local
54.1.1.0/24, ubest/mbest: 1/0, attached
    *via 54.1.1.1, Eth4/2, [0/0], 00:03:50, direct
54.1.1.1/32, ubest/mbest: 1/0, attached
    *via 54.1.1.1, Eth4/2, [0/0], 00:03:50, local
212.18.0.0/24, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [120/2], 00:03:50, rip-NEXUS, rip
212.18.1.0/24, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [120/2], 00:03:50, rip-NEXUS, rip
212.18.2.0/24, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [120/2], 00:03:50, rip-NEXUS, rip
212.18.3.0/24, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [120/2], 00:03:50, rip-NEXUS, rip
N7K2-VDC-2#
```

We should now be able to ***PING 212.18.3.1***

```
N7K2-VDC-2# ping 212.18.3.1
PING 212.18.3.1 (212.18.3.1): 56 data bytes
64 bytes from 212.18.3.1: icmp_seq=0 ttl=254 time=1.338 ms
64 bytes from 212.18.3.1: icmp_seq=1 ttl=254 time=0.967 ms
64 bytes from 212.18.3.1: icmp_seq=2 ttl=254 time=0.969 ms
64 bytes from 212.18.3.1: icmp_seq=3 ttl=254 time=0.975 ms
64 bytes from 212.18.3.1: icmp_seq=4 ttl=254 time=0.973 ms

--- 212.18.3.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.967/1.044/1.338 ms
N7K2-VDC-2#
```

Lab 14

Routing with OSPF and authentication

Tasks:

- Disable the RIP feature and enable OSPF
- Configure OSPF Process 10
- Place the 54.2.2.0/24 network in area 0
- Place all VLAN interfaces in Area 0
- The neighbor on e4/2 is expecting MD5 authentication with a password of CISCO
- To test, ping 20.11.0.1

Disable RIP and enable OSPF

```
N7K2-VDC-2(config)# no feature rip
N7K2-VDC-2(config)# feature ospf
N7K2-VDC-2(config)#
```

OSPF authentication is configured the same as IOS, under the interface.

```
N7K2-VDC-2(config-keychain-key)# int e4/2
N7K2-VDC-2(config-if)# ip router ospf 10 area 0
N7K2-VDC-2(config-if)# ip ospf authentication
N7K2-VDC-2(config-if)# ip ospf authentication message-digest
N7K2-VDC-2(config-if)# ip ospf authentication-key CISCO
N7K2-VDC-2(config-if)# int vlan 100
N7K2-VDC-2(config-if)# ip router ospf 10 area 0
N7K2-VDC-2(config-if)# int vlan 110
N7K2-VDC-2(config-if)# ip router ospf 10 area 0
N7K2-VDC-2(config-if)# int vlan 120
N7K2-VDC-2(config-if)# ip router ospf 10 area 0
N7K2-VDC-2(config-if)# int vlan 130
N7K2-VDC-2(config-if)# ip router ospf 10 area 0
N7K2-VDC-2(config-if)#
```

Look at the routing table

```
N7K2-VDC-2# sh ip route
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

10.10.10.0/24, ubest/mbest: 1/0, attached
    *via 10.10.10.2, Eth3/1, [0/0], 04:22:32, direct
10.10.10.2/32, ubest/mbest: 1/0, attached
    *via 10.10.10.2, Eth3/1, [0/0], 04:22:32, local
10.100.1.0/24, ubest/mbest: 1/0, attached
    *via 10.100.1.3, Vlan100, [0/0], 04:13:55, direct
10.100.1.1/32, ubest/mbest: 1/0
    *via 10.100.1.1, Vlan100, [0/0], 02:01:26, hsrp
10.100.1.3/32, ubest/mbest: 1/0, attached
    *via 10.100.1.3, Vlan100, [0/0], 04:13:55, local
10.110.1.0/24, ubest/mbest: 1/0, attached
    *via 10.110.1.3, Vlan110, [0/0], 04:13:55, direct
10.110.1.1/32, ubest/mbest: 1/0
    *via 10.110.1.1, Vlan110, [0/0], 02:00:55, hsrp
10.110.1.3/32, ubest/mbest: 1/0, attached
    *via 10.110.1.3, Vlan110, [0/0], 04:13:55, local
10.120.1.0/24, ubest/mbest: 1/0, attached
```

```
    *via 10.120.1.3, Vlan120, [0/0], 04:13:55, direct
10.120.1.1/32, ubest/mbest: 1/0
    *via 10.120.1.1, Vlan120, [0/0], 02:00:43, hsrp
10.120.1.3/32, ubest/mbest: 1/0, attached
    *via 10.120.1.3, Vlan120, [0/0], 04:13:55, local
10.130.1.0/24, ubest/mbest: 1/0, attached
    *via 10.130.1.3, Vlan130, [0/0], 04:13:55, direct
10.130.1.1/32, ubest/mbest: 1/0
    *via 10.130.1.1, Vlan130, [0/0], 02:00:14, hsrp
10.130.1.3/32, ubest/mbest: 1/0, attached
    *via 10.130.1.3, Vlan130, [0/0], 04:13:55, local
20.0.0.1/32, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [110/401], 00:00:08, ospf-10, intra
20.1.0.1/32, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [110/401], 00:00:08, ospf-10, intra
20.2.0.1/32, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [110/401], 00:00:08, ospf-10, intra
20.3.0.1/32, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [110/401], 00:00:08, ospf-10, intra
20.4.0.1/32, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [110/401], 00:00:08, ospf-10, intra
20.5.0.1/32, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [110/401], 00:00:08, ospf-10, intra
20.6.0.1/32, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [110/401], 00:00:08, ospf-10, intra
20.7.0.1/32, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [110/401], 00:00:08, ospf-10, intra
20.8.0.1/32, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [110/401], 00:00:08, ospf-10, intra
20.9.0.1/32, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [110/401], 00:00:08, ospf-10, intra
20.10.0.1/32, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [110/401], 00:00:08, ospf-10, intra
20.11.0.1/32, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [110/401], 00:00:08, ospf-10, intra
54.1.1.0/24, ubest/mbest: 1/0, attached
    *via 54.1.1.1, Eth4/2, [0/0], 00:11:51, direct
54.1.1.1/32, ubest/mbest: 1/0, attached
    *via 54.1.1.1, Eth4/2, [0/0], 00:11:51, local
N7K2-VDC-2#
```

And ***ping 20.11.0.1***

```
N7K2-VDC-2# ping 20.11.0.1
PING 20.11.0.1 (20.11.0.1): 56 data bytes
64 bytes from 20.11.0.1: icmp_seq=0 ttl=254 time=1.291 ms
64 bytes from 20.11.0.1: icmp_seq=1 ttl=254 time=0.967 ms
64 bytes from 20.11.0.1: icmp_seq=2 ttl=254 time=0.84 ms
64 bytes from 20.11.0.1: icmp_seq=3 ttl=254 time=0.848 ms
64 bytes from 20.11.0.1: icmp_seq=4 ttl=254 time=0.866 ms

--- 20.11.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.84/0.962/1.291 ms
N7K2-VDC-2#
```

Lab 15

Routing EIGRP with authentication, non-stop forwarding, and passive interfaces

Tasks:

- Disable OSPF and enable EIGRP on the N7K2-VDC-2
- Configure EIGRP for AS10
- Enable EIGRP non-stop forwarding with the following values:
  - Converge – 180 seconds
  - Route hold – 200 seconds
- Enable routing for the 54.1.1.0/24 network and all VLANs
- The neighbor on e4/2 is excepting a MD5 authentication password of Cisco
- Place all VLANs interfaces in PASSIVE mode for EIGRP AS 10
- Ping 200.0.3.1 to test connectivity

Disable OSPF and enable EIGRP

```
N7K2-VDC-2(config)# no feature ospf
N7K2-VDC-2(config)# feature eigrp
```

Configure EIGRP, but to enable NSF you need to change to the ***address-family ipv4 unicast*** *and configure* ***graceful-restart*** *and associated* ***timers****.*
Authenticaiton is configured the same as IOS with ***key-chain*** *commands.*

```
N7K2-VDC-2(config)# router eigrp 10
N7K2-VDC-2(config-router)# address-family ipv4 unicast
N7K2-VDC-2(config-router-af)# graceful-restart
N7K2-VDC-2(config-router-af)# timers nsf converge 180
N7K2-VDC-2(config-router-af)# timers nsf route-hold 200
N7K2-VDC-2(config)# key chain EIGRP
N7K2-VDC-2(config-keychain)# key 1
N7K2-VDC-2(config-keychain-key)# key-string CISCO
N7K2-VDC-2(config-keychain-key)#
N7K2-VDC-2(config)# int e4/2
N7K2-VDC-2(config-if)# ip router eigrp 10
N7K2-VDC-2(config-if)# ip authentication mode eigrp 10 md5
N7K2-VDC-2(config-if)# ip authentication key-chain eigrp 10 EIGRP
N7K2-VDC-2(config-if)# exit
```

To place an interface in passive, under the interface use the ***ip passive-interface eigrp AS#*** *command*

```
N7K2-VDC-2(config)# int vlan 100
N7K2-VDC-2(config-if)# ip router eigrp 10
N7K2-VDC-2(config-if)# ip passive-interface eigrp 10
N7K2-VDC-2(config-if)# int vlan 110
N7K2-VDC-2(config-if)# ip router eigrp 10
N7K2-VDC-2(config-if)# ip passive-interface eigrp 10
N7K2-VDC-2(config-if)# int vlan 120
N7K2-VDC-2(config-if)# ip router eigrp 10
N7K2-VDC-2(config-if)# ip passive-interface eigrp 10
N7K2-VDC-2(config-if)# int vlan 130
N7K2-VDC-2(config-if)# ip router eigrp 10
N7K2-VDC-2(config-if)# ip passive-interface eigrp 10
N7K2-VDC-2(config-if)#
```

Look at the routing table

```
N7K2-VDC-2# sh ip route
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

10.10.10.0/24, ubest/mbest: 1/0, attached
    *via 10.10.10.2, Eth3/1, [0/0], 04:41:46, direct
10.10.10.2/32, ubest/mbest: 1/0, attached
    *via 10.10.10.2, Eth3/1, [0/0], 04:41:46, local
10.100.1.0/24, ubest/mbest: 1/0, attached
    *via 10.100.1.3, Vlan100, [0/0], 04:33:09, direct
10.100.1.1/32, ubest/mbest: 1/0
    *via 10.100.1.1, Vlan100, [0/0], 02:20:40, hsrp
10.100.1.3/32, ubest/mbest: 1/0, attached
    *via 10.100.1.3, Vlan100, [0/0], 04:33:09, local
10.110.1.0/24, ubest/mbest: 1/0, attached
    *via 10.110.1.3, Vlan110, [0/0], 04:33:09, direct
10.110.1.1/32, ubest/mbest: 1/0
    *via 10.110.1.1, Vlan110, [0/0], 02:20:09, hsrp
10.110.1.3/32, ubest/mbest: 1/0, attached
    *via 10.110.1.3, Vlan110, [0/0], 04:33:09, local
10.120.1.0/24, ubest/mbest: 1/0, attached
    *via 10.120.1.3, Vlan120, [0/0], 04:33:09, direct
10.120.1.1/32, ubest/mbest: 1/0
    *via 10.120.1.1, Vlan120, [0/0], 02:19:57, hsrp
10.120.1.3/32, ubest/mbest: 1/0, attached
    *via 10.120.1.3, Vlan120, [0/0], 04:33:09, local
10.130.1.0/24, ubest/mbest: 1/0, attached
    *via 10.130.1.3, Vlan130, [0/0], 04:33:09, direct
10.130.1.1/32, ubest/mbest: 1/0
    *via 10.130.1.1, Vlan130, [0/0], 02:19:28, hsrp
10.130.1.3/32, ubest/mbest: 1/0, attached
    *via 10.130.1.3, Vlan130, [0/0], 04:33:09, local
54.1.1.0/24, ubest/mbest: 1/0, attached
    *via 54.1.1.1, Eth4/2, [0/0], 00:31:05, direct
54.1.1.1/32, ubest/mbest: 1/0, attached
    *via 54.1.1.1, Eth4/2, [0/0], 00:31:05, local
200.0.0.0/24, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [90/153856], 00:00:11, eigrp-10, internal
200.0.1.0/24, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [90/153856], 00:00:11, eigrp-10, internal
200.0.2.0/24, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [90/153856], 00:00:11, eigrp-10, internal
200.0.3.0/24, ubest/mbest: 1/0
    *via 54.1.1.254, Eth4/2, [90/153856], 00:00:11, eigrp-10, internal
N7K2-VDC-2#
```

And ping 200.0.3.1

```
N7K2-VDC-2# ping 200.0.3.1
PING 200.0.3.1 (200.0.3.1): 56 data bytes
64 bytes from 200.0.3.1: icmp_seq=0 ttl=254 time=1.49 ms
64 bytes from 200.0.3.1: icmp_seq=1 ttl=254 time=0.959 ms
64 bytes from 200.0.3.1: icmp_seq=2 ttl=254 time=0.973 ms
64 bytes from 200.0.3.1: icmp_seq=3 ttl=254 time=0.967 ms
64 bytes from 200.0.3.1: icmp_seq=4 ttl=254 time=0.974 ms

--- 200.0.3.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.959/1.072/1.49 ms
```

```
N7K2-VDC-2#
```

Side notes for non-stop forwarding:

During a RP Failover, the EIGRP neighbor does not go down and packets continue to forward.

```
N7KLab_RTR#sh debug

EIGRP:
  EIGRP NSF Events debugging is on
N7KLab_RTR#sh ip eigrp neighbors
IP-EIGRP neighbors for process 10
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   54.1.1.1                Fa0/0             14 00:01:29   11   200  0  7
N7KLab_RTR#
*Jun 15 17:42:54.836: EIGRP: AS10. Peer resync 54.1.1.1 at 00:00:00.
N7KLab_RTR#
*Jun 15 17:43:14.796: EIGRP: NSF: AS10. Receive EOT from 54.1.1.1
*Jun 15 17:43:14.800: EIGRP: NSF: AS10. route hold timer expiry
*Jun 15 17:43:14.800: DUAL: Search for outdated routes from 54.1.1.1
N7KLab_RTR#sh ip eigrp neighbors
IP-EIGRP neighbors for process 10
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   54.1.1.1                Fa0/0             11 00:02:23   13   200  0  4
N7KLab_RTR#
```

As you can see from the output above, the EIGRP neighbor was notified of the failover of the supervisors and continued to maintain the neighbor state.

Lab 16:

Failing over the Supervisors

Tasks:

- Failover from Sup-1 to Sup-2
  - o You should connect to both Supervisors slots via Console while doing this task.
- Once Sup-1 as recovered, fail back.

Do a ***system switchover***

```
N7K2# system switchover
The primary supervisor will now reboot:
  <CR>

N7K2# system switchover
N7K2#  writing reset reason 7,

NX7 SUP Ver 3.17.0
Serial Port Parameters from CMOS
PMCON_1: 0x200
PMCON_2: 0x0
PMCON_3: 0x3a
PM1_STS: 0x101
Performing Memory Detection and Testing
Testing 1 DRAM Patterns
 [----[ Information Snipped ]----]
```

On the supervisor that was in STANDBY will display:

```
N7K2(standby) login: 2010 Jun 15 21:24:35 N7K2 %$ VDC-1 %$ %KERN-2-SYSTEM_MSG:
Switchover started by redundancy driver - kernel
2010 Jun 15 21:24:35 N7K2 %$ VDC-1 %$ %SYSMGR-2-HASWITCHOVER_PRE_START: This
supervisor is becoming active (pre-start phase).
2010 Jun 15 21:24:35 N7K2 %$ VDC-1 %$ %SYSMGR-2-HASWITCHOVER_START: This supervisor is
becoming active.
2010 Jun 15 21:24:36 N7K2 %$ VDC-1 %$ %SYSMGR-2-SWITCHOVER_OVER: Switchover completed.
```

To fail back to the primary, use the command ***system switchover****. If you try to failover before the primary is ready, you will receive this message:*

```
N7K2# system switchover
Failed to switch over (standby not ready to take over)
N7K2# system switchover
```

Once this message appears on the ACTIVE supervisor, you can should switch back to the primary soon.

```
2010 Jun 15 21:28:05 N7K2 %$ VDC-1 %$ %CMPPROXY-STANDBY-2-LOG_CMP_UP: Connectivity
Management processor(on module 5) is now UP
```

If you try to switchover before all the VDCs are booted, you will see this message:

```
N7K2# system switchover
Failed to switch over (standby not ready to take over in vdc 2)
N7K2#
N7K2# system switchover
N7K2#  writing reset reason 7,

NX7 SUP Ver 3.17.0
Serial Port Parameters from CMOS
```

Lab 17:

Creating a checkpoint and rolling back

Tasks:

- On VDC-2
  - Create a checkpoint called NEXUSLAB
  - Give it a description of Lab Training
  - Create VLAN 500 and name it CHECKPOINT
  - Assign ports 3/12-24 using interface ranges
  - Create a 2nd checkpoint called NEXUSLAB2
  - Save the this checkpoint to the bootflash
  - Rollback your config to the first checkpoint (NEXUSLAB) and verify.

To create a checkpoint, use the command ***checkpoint NAME description DESC***

```
N7K2-VDC-2# checkpoint NEXUSLAB desc Lab Training
Done
```

Now we can create the VLAN and assign the ports. To do an interface range on NX-OS you do not need to do the ***range*** *command like IOS. Just add a* **–** *and the NX-OS is smart enough to know what you mean. Remember, that in order to assign a port to a VLAN it must be a* ***switchport****.*

```
N7K2-VDC-2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
N7K2-VDC-2(config)# vlan 500
N7K2-VDC-2(config-vlan)# name CHECKPOINT
N7K2-VDC-2(config-vlan)#
N7K2-VDC-2(config-vlan)# interface e3/12 - 24
N7K2-VDC-2(config-if-range)# switchport
N7K2-VDC-2(config-if-range)# sw mode access
N7K2-VDC-2(config-if-range)# switchport access vlan 500
```

Now we can check the VLAN port allocation via ***show vlan***

```
N7K2-VDC-2(config-if-range)# sh vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Po10, Po20
100  VL100                            active    Po10, Po20
110  VL110                            active    Po10, Po20
120  VL120                            active    Po10, Po20
130  VL130                            active    Po10, Po20
500  CHECKPOINT                       active    Po10, Po20, Eth3/12, Eth3/13
                                                Eth3/14, Eth3/15, Eth3/16
                                                Eth3/17, Eth3/18, Eth3/19
                                                Eth3/20, Eth3/21, Eth3/22
                                                Eth3/23, Eth3/24
```

To create a checkpoint on the flash, just designate the ***file***

```
N7K2-VDC-2# checkpoint file bootflash:NEXUSLAB2
Done
N7K2-VDC-2# dir
       7008     Jun 16 01:28:10 2010  NEXUSLAB2

Usage for bootflash://
  373317632 bytes used
 1436581888 bytes free
 1809899520 bytes total
```

*To rollback the config, use the command **rollback running-config checkpoint CHECKPOINTNAME***

```
N7K2-VDC-2# rollback running-config checkpoint NEXUSLAB
Note: Applying config parallelly may fail Rollback verification
Collecting Running-Config
Generating Rollback Patch
Executing Rollback Patch
Generating Running-config for verification
Generating Patch for verification
```

Now we can check the VLANs and see that VLAN 500 is no longer there

```
N7K2-VDC-2# sh vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Po10, Po20
100  VL100                            active    Po10, Po20
110  VL110                            active    Po10, Po20
120  VL120                            active    Po10, Po20
130  VL130                            active    Po10, Po20
```

Lab 18:

Deleting a VDC

Tasks:

- On N7K1 and N7K2, delete all the non-default VDCs

In order to delete the VDCs, you need to be in VDC1. First, let's see what VDCs have been created. To do this, use ***SHOW VDC***

```
N7K2# sh vdc

vdc_id  vdc_name                          state               mac
------  --------                          -----               ----------
1       N7K2                              active              00:26:98:07:95:c1
2       VDC-2                             active              00:26:98:07:95:c2
3       VDC-3                             active              00:26:98:07:95:c3

N7K2#
```

Here we can see that we have two VDCs on the switch -> VDC-2 and VDC-3. To remove, use the ***no vdc VDCNAME*** *command in config mode.*

```
N7K2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
N7K2(config)# no vdc vdc-2
Deleting this vdc will remove its config. Continue deleting this vdc (y/n)?  [no] y
Note:  Deleting VDC, one moment please ...
2010 Jun 16 20:55:03 N7K2-VDC-2 %$ VDC-2 %$ %VPC-2-PEER_KEEP_ALIVE_RECV_FAIL: In
domain 100, VPC peer keep-alive receive has failed
N7K2(config)#
2010 Jun 16 20:55:26 N7K2 %$ VDC-1 %$ %VDC_MGR-2-VDC_OFFLINE: vdc 2 is now offline
N7K2(config)#
N7K2(config)# no vdc vDC-3
Deleting this vdc will remove its config. Continue deleting this vdc (y/n)?  [no] y
Note:  Deleting VDC, one moment please ...
N7K2(config)# 2010 Jun 16 20:56:18 N7K2 %$ VDC-1 %$ %VDC_MGR-2-VDC_OFFLINE: vdc 3 is
now offline
N7K2(config)#
```

And to confirm, lets do ***show vdc***

```
N7K2(config)# show vdc

vdc_id  vdc_name                          state               mac
------  --------                          -----               ----------
1       N7K2                              active              00:26:98:07:95:c1

N7K2(config)#
```

Now for the other switch:

```
N7K1# sh vdc

vdc_id  vdc_name                          state               mac
------  --------                          -----               ----------
1       N7K1                              active              00:26:98:1b:f0:41
2       VDC-2                             active              00:26:98:1b:f0:42
3       VDC-3                             active              00:26:98:1b:f0:43

N7K1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
N7K1(config)# no vdc vdc-2
Deleting this vdc will remove its config. Continue deleting this vdc (y/n)?  [no] y
```

```
Note:  Deleting VDC, one moment please ...
N7K1(config)# 2010 Jun 16 20:52:41 N7K1 %$ VDC-1 %$ %VDC_MGR-2-VDC_OFFLINE: vdc 2 is
now offline

N7K1(config)# no vdc vdc-3
Deleting this vdc will remove its config. Continue deleting this vdc (y/n)?  [no] y
Note:  Deleting VDC, one moment please ...
N7K1(config)# 2010 Jun 16 20:54:48 N7K1 %$ VDC-1 %$ %VDC_MGR-2-VDC_OFFLINE: vdc 3 is
now offline

N7K1(config)#
```

Note: Once you delete a vdc, there is no undo command. The VDC and all associated configs are gone. This is a permanent change.