CCNPv7.1 SWITCH

Chapter 1 Lab - Preparing the Switch

1. Topology



1. Objectives
* Clear the configuration of **all** **the switches in your pod**
* Configure the database template used by **all the switches in your pod**
* Save a baseline configuration **for all the switches in your pod**
1. Background

When working with a switch that has been previously configured, any new commands entered will be merged with the existing configuration, causing unpredictable results. Additionally, if the switch is connected to other switches in the network, you can remove the VLANs but they might be relearned from another switch via VTP. In this lab you prepare your switches for use with future labs. This is accomplished by erasing the startup configuration from NVRAM and deleting the VLAN database. You also ensure that VLANs will not be relearned from another switch after the VLAN database has been deleted. Additionally, your switches may be required to support IPv6 traffic, which it does not by default. This is accomplished by changing the database template used by the Switch Database Manager.

**Note:** This lab uses Cisco Catalyst 3560 and 2960 switches running Cisco IOS 15.0(2)SE6 IP Services and LAN Base images, respectively. The 3560 and 2960 switches are configured with the SDM templates “dual-ipv4-and-ipv6 routing” and “lanbase-routing”, respectively. Depending on the switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab. Catalyst 3650 switches (running any Cisco IOS XE release) and Catalyst 2960-Plus switches (running any comparable Cisco IOS image) can be used in place of the Catalyst 3560 switches and the Catalyst 2960 switches.

Required Resources

You may use one of the following switches **or a comparable one** with this lab:

* Cisco 2960 with the Cisco IOS Release 15.0(2)SE6 C2960-LANBASEK9-M or comparable
* Cisco 3560v2 with the Cisco IOS Release 15.0(2)SE6 C3560-ipservicesK9-M or comparable
* Console Cable
* Computer with terminal emulation software
	1. Connect to the switch console port and enter privileged EXEC mode.

From a computer running a terminal emulation program, connect to the console port of the switch that you want to clear using a console cable. You should see a console prompt that includes the switch’s hostname, followed by a **>** or **#**. The default switch hostname is “Switch.”

Switch>

or

Switch#

If the prompt ends with a **>**, you are not in privileged EXEC mode. To enter privileged EXEC mode, type **enable**. This might require a password. If you are in a configuration mode, type **exit** or **end**.

If not enabled:

Switch> **enable**

Switch#

If in global configuration mode:

Switch(config)# **exit**

Switch#

* 1. Delete the VLAN database file, if present.

A VLAN database file named vlan.dat might exist in FLASH on the switch if it has been previously used in the network. This file holds information about VLANs created on the switch, their IDs, names, types and states, and it also stores the VTP settings. In privileged EXEC mode, type **dir** or **dir flash:** and press Enter. This will provide a directory listing of the files in FLASH. In particular, note two files in the output: the vlan.dat file that will be removed in this step, and the multiple-fs file that will be explained and removed in Step 3 below.

Switch#dir

Directory of flash:/

 3 drwx 512 Mar 1 1993 00:38:22 +00:00 c3560-ipservicesk9-mz.150-2.SE6

 522 -rwx 4889 Mar 2 1993 01:37:37 +00:00 startup-config

 560 -rwx 3096 Mar 1 1993 02:55:29 +00:00 multiple-fs

 561 -rwx 616 Mar 11 1993 23:00:09 +00:00 vlan.dat

In privileged EXEC mode, type **delete flash:vlan.dat** or **delete vlan.dat** (in the shorter form without the flash: prefix, the <TAB> key completion does not work) and press Enter. If you are asked to confirm, press Enter until you are back to the original prompt.

Switch# **delete vlan.dat**

Delete filename [vlan.dat]? ! Press Enter

Delete flash:vlan.dat? [confirm] ! Press Enter

Switch#

* 1. Erase the startup config from NVRAM.

On Cisco devices, NVRAM is the common location for configuration files. The goal of this step is to entirely erase the NVRAM contents so that on the next boot, the switch starts in a factory default configuration. There are, however, a few important facts you need to be aware of.

On the switch platforms used in these or similar labs, such as Catalyst 2950, 2960, 3550, 3560, 3650, 3750, 3850, the NVRAM is not truly physically present. Instead, a part of the FLASH memory is used to store the NVRAM contents. In other words, on these switch platforms, the NVRAM is only simulated using a part of the FLASH, as also evidenced by one of lines in the **show version** command output:

Switch# **show version | include volatile**

512K bytes of flash-simulated non-volatile configuration memory.

Files that appear to reside in NVRAM (use **dir nvram:** to display its contents) are in fact stored in FLASH. Some of them are stored as standalone files in FLASH, such as flash:config.text that maps to nvram:startup-config, or flash:private-config.text that maps to nvram:private-config and stores sensitive information such as RSA keys, master password encryption key etc. Deleting any of these files from FLASH will cause the corresponding mapped file in simulated NVRAM to also be deleted or its apparent length in NVRAM to be truncated to zero, and vice versa. Other files in NVRAM, such as self-generated X.509 certificates, are all stored in the flash:multiple-fs file.

The **erase startup-config** command commonly used to remove the stored configuration will remove the flash:config.text and flash:private-config.text. However, other contents of the simulated NVRAM, such as X.509 certificates that were automatically created for the HTTPS server run on the switch, will not be removed as they reside in the flash:multiple-fs file unaffected by the **erase startup-config** command. Therefore, to completely erase the simulated NVRAM contents, not only the **erase startup-config** command must be issued, but also the flash:multiple-fs file must be removed.

In privileged EXEC mode, issue the **delete flash:multiple-fs** (or simply **delete multiple-fs**) command, followed by the **erase startup-config** command. Press Enter on each prompt.

Switch# **delete multiple-fs**

Delete filename [multiple-fs]? ! Press Enter

Delete flash:/multiple-fs? [confirm] ! Press Enter

Switch# **erase startup-config**

Erasing the nvram filesystem will remove all configuration files! Continue? [confirm] ! Press Enter

[OK]

Erase of nvram: complete

Switch#

\*Mar 1 00:43:23.286: %SYS-7-NV\_BLOCK\_INIT: Initialized the geometry of nvram

**Note:** Historically, the **write erase** command was used to erase the startup configuration. This command has the same effect as **erase startup-config** command that was implemented in later IOS versions, and is still being used as its shortened version **wr e** is more convenient to write than **erase startup-config**.

* 1. Change the Switch Database Template.

The Cisco Switch Database Manager (SDM) provides various TCAM allocation templates that can be enabled to support specific roles, depending on how the switch is used in the network. By default the switch is using the “Default Desktop” template. This particular template divides the available TCAM up for use by the different processes and protocols in a manner that most likely supports standard IPv4 unicast and multicast traffic. Use the **show sdm prefer** command to see the details of the current template. Output may differ depending on the specific switch platform.

Switch# **show sdm prefer**

 The current template is "desktop default" template.

 The selected template optimizes the resources in

 the switch to support this level of features for

 8 routed interfaces and 1024 VLANs.

 number of unicast mac addresses: 6K

 number of IPv4 IGMP groups + multicast routes: 1K

 number of IPv4 unicast routes: 8K

 number of directly-connected IPv4 hosts: 6K

 number of indirect IPv4 routes: 2K

 number of IPv6 multicast groups: 0

 number of IPv6 unicast routes: 0

 number of directly-connected IPv6 addresses: 0

 number of indirect IPv6 unicast routes: 0

 number of IPv4 policy based routing aces: 0

 number of IPv4/MAC qos aces: 0.5K

 number of IPv4/MAC security aces: 1K

 number of IPv6 policy based routing aces: 0

 number of IPv6 qos aces: 20

 number of IPv6 security aces: 25

Switch#

Notice in the output that there is NO memory allocated to IPv6 operations.

There are several different SDM templates available for use, each with different amounts of TCAM allocated to different processes and protocols. Use the **show sdm prefer *template*** command to examine the details of a particular database template.

Switch# **show sdm prefer ?**

 access Access bias

 default Default bias

 dual-ipv4-and-ipv6 Support both IPv4 and IPv6

 routing Unicast bias

 vlan VLAN bias

 | Output modifiers

 <cr>

Switch# **show sdm prefer routing**

 "desktop routing" template:

 The selected template optimizes the resources in

 the switch to support this level of features for

 8 routed interfaces and 1024 VLANs.

 number of unicast mac addresses: 3K

 number of IPv4 IGMP groups + multicast routes: 1K

 number of IPv4 unicast routes: 11K

 number of directly-connected IPv4 hosts: 3K

 number of indirect IPv4 routes: 8K

 number of IPv6 multicast groups: 0

 number of IPv6 unicast routes: 0

 number of directly-connected IPv6 addresses: 0

 number of indirect IPv6 unicast routes: 0

 number of IPv4 policy based routing aces: 0.5K

 number of IPv4/MAC qos aces: 0.5K

 number of IPv4/MAC security aces: 1K

 number of IPv6 policy based routing aces: 0

 number of IPv6 qos aces: 20

 number of IPv6 security aces: 25

Switch#

Because our switching network will need to support IPv6 traffic, the SDM template must be changed to one of the dual-ipv4-and-ipv6 templates, using the **sdm prefer dual-ipv4-and-ipv6 *template*** global configuration command.

The template options for dual IPv4 and IPv6 operation vary based on the model of switch. As of this writing, the dual-ipv6-and-ipv4 template options on 3560s are **default**, **routing**, and **VLAN**, while the option on 2960s is **default**. The 2960's **lanbase-routing** template also supports connected IPv6 hosts.

**Note:** The **routing** template is not a valid selection on switches running the LANBASE feature set, even though **routing** may appear as an option at the command line.

Switch(config)# **sdm prefer dual-ipv4-and-ipv6 routing**

Changes to the running SDM preferences have been stored, but cannot take effect until the next reload.

Use 'show sdm prefer' to see what SDM preference is currently active.

Switch(config)#

As you can see in the output above, a restart is required for the change to be effective; the switch cannot re-allocate the TCAM on the fly. You will restart the switch at the end of this lab.

Configure your 3560 switches for the **dual-ipv4-and-ipv6 routing** template, and your 2960 switches for the **lanbase-routing** template.

**Note**: Several of the labs in the course end with instructions to reset the switch to its defaults. If you clear the switch (**delete vlan.dat**, **delete multiple-fs**, **write erase**, **reload**), the selected SDM template will return to the default, and could require reconfiguration (including a reboot).

* 1. Reload the device, but do not save the system configuration if prompted.

After clearing the switch configuration, reload the switch by typing **reload** and pressing Enter. If you are asked whether to save the current configuration, answer **no**. Press Enter to confirm. The switch starts reloading. Your output might look different depending on the switch model that you are using. This step might take a few minutes, because the switch needs time to reload.

Switch# **reload**

System configuration has been modified. Save? [yes/no]: **no**

Proceed with reload? [confirm]

\*Mar 11 23:03:06.985: %SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload command.

<output omitted>

* 1. Create a Baseline Configuration

To eliminate some of the redundant basic configurations, use a TCL script to build a configuration shell. This can be customized several ways, but the basic text below creates a file named BASE.CFG in FLASH that can be used at the beginning of labs after the switch has been completely cleared. This will save some of the mundane configuration steps

Use the script below and modify it to meet the particulars of the switch you are working on:

tclsh

puts [ open "flash:BASE.CFG" w+ ] {

hostname **DLS1**

ip domain-name CCNP.NET

no ip domain lookup

interface range f0/1-24 , g0/1-2

shutdown

exit

vtp mode transparent

line con 0

no exec-timeout

logging synchronous

exit

end

}

tclquit

* 1. Create a script to automate clearing and loading of the switch

Use TCL once more to create a script to automate the tasks involved in clearing the switch.

DLS1/DLS2:

tclsh

puts [ open "flash:reset.tcl" w+ ] {

typeahead "\n"

copy running-config startup-config

typeahead "\n"

erase startup-config

delete /force vlan.dat

delete /force multiple-fs

ios\_config "sdm prefer dual-ipv4-and-ipv6 routing"

typeahead "\n"

puts "Reloading the switch in 1 minute, type reload cancel to halt"

typeahead "\n"

reload in 1 RESET.TCL SCRIPT RUN

}

tclquit

ALS1/ALS2:

tclsh

puts [ open "flash:reset.tcl" w+ ] {

typeahead "\n"

copy running-config startup-config

typeahead "\n"

erase startup-config

delete /force vlan.dat

delete /force multiple-fs

ios\_config "sdm prefer lanbase-routing"

typeahead "\n"

puts "Reloading the switch in 1 minute, type reload cancel to halt"

typeahead "\n"

reload in 1 RESET.TCL SCRIPT RUN

}

tclquit

Now for any lab that requires clearing the switch, issue the privileged exec command **tclsh reset.tcl** and the switch will be completely cleared and reload in the proper state.

* 1. End of Lab

At this point, your switches should be at a factory default with the BASE.CFG file in FLASH for future use.