Two-stage Configuration Commit

This chapter describes how to enable two-stage configuration commit mode on the Cisco NX-OS device.

This chapter includes the following sections:

- Two-stage configuration commit process, on page 1
- Guidelines and limitations for two-stage configuration commit, on page 2
- Configuring in Two-Stage Configuration Commit Mode, on page 3
- Abort the two-stage configuration commit mode, on page 8
- Display commit IDs, on page 9
- Rollback capability, on page 9
- View current session configurations, on page 10

Two-stage configuration commit process

Configuration Commit is the process of applying configuration changes to a network device. It determines when and how configuration changes take effect on the device. Based on how the configuration is committed, it divided into two modes—one-stage and two-stage.

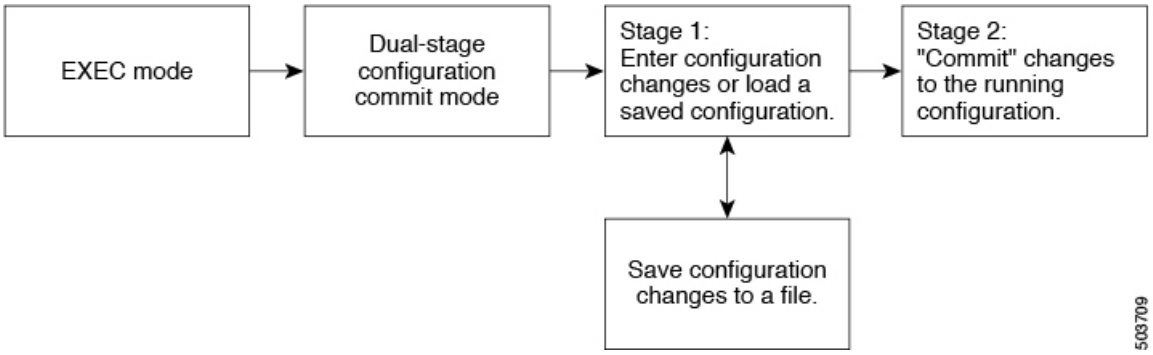
Summary

In an interactive session, when you run a command, it is executed and it changes the running configuration. This is known as one-stage configuration commit. In confirm-commit or two-stage configuration commit mode, changes in configurations are stored in a staging database. These changes do not affect the running configuration until you run the **commit** command.

The two-stage process creates a target configuration session, where you can make, edit, and verify configuration changes before committing them to the running state of the switch. You can also commit the changes for a time period you specify before you commit them permanently. After the specified time period, the switch reverts to the previous configuration if you do not run the **commit** command. When a commit is successful, you can view the commit information that includes the commit ID, username, and timestamp.

Workflow

Figure 1: Two-stage configuration commit process



The image depicts the two stages of the dual-stage configuration commit mode—configuration changes and commit changes.

Guidelines and limitations for two-stage configuration commit

This section lists the configuration guidelines and limitations divided into various categories for two-stage configuration commit.

Before you run any feature-related configuration commands, enable the feature using the **feature** command and commit it using the **commit** command.

Configuration

- This feature is supported only for a command line interface in a user-interactive session.
- Two-stage configuration commit mode does not support other modes like maintenance mode, scheduler mode, or virtual mode.
- When you are in the two-stage configuration commit mode, avoid editing configurations in one-stage configuration commit mode from different sessions at the same time.
- If the verification fails, edit and retry the commit.
- If the commit fails, the configuration rolls back to the previous configuration.
- Configurations that you do not commit are not saved after you reload the switch.
- This feature does not support commits with NX-API, EEM, PPM, and NETCONF.
- You can have only one active two-stage configuration commit session at a given time.

  An SSH connection can terminate unexpectedly without ending the associated two-stage configuration session. When this occurs, the session remains active, and a subsequent `configure dual-stage` command displays `Session is already in use`. Run `show configuration dual-stage sessions` to identify the active session.
- Only users with the network-admin role can use two-stage configuration commit.

Show configuration

- Review the configurations using the **show configuration** command before committing the changes.

- Show configuration displays the staged configs:
  - It displays the real difference, that is yes and no form of the same command will result in empty config.
  - It is recommended to issue the exact no form of the command to negate the configuration.

    For example, to negate **ip address** *x* config, user has to give **no ip address***x* instead of **no ip address**.
  - Interface layer change commands (switchport or no switchport) should be issued explicitly.
  - Any invalid configuration in the session should manually be removed by the user before attempting commit. If could not remove manually clear the session and start a new session.

Platform support

- N9K-C92348GC-X supports two-stage configuration commit from Cisco NX-OS Release 10.4(3)F.

Configuring in Two-Stage Configuration Commit Mode

To enable a feature in the two-stage configuration commit mode, perform the following steps:

Note In this procedure, the BGP feature is enabled as an example.

Procedure

	Command or Action	Purpose

Step 1	**configure dual-stage**
	Example:
	`switch# configure dual-stage`
	`switch(config-dual-stage)#`
		Creates a new target configuration session.
		Note The target configuration isn't a copy of
		entered during the target configuration

Step 2	**feature** *feature_name*
	Example:
	`switch(config-dual-stage)# feature bgp`
	`switch(config-dual-stage)#`
		Enables the feature.
		Note
		• You can enable the feature even be
		• You can't combine feature-related

Step 3	**commit** [**confirmed** *seconds*] **Example:** `switch(config-dual-stage-router)# commit confirmed 30`	Commits changes to the running configuration. • **confirmed**: Commits the changes to the running configuration.

	Command or Action	Purpose
	Verification Succeeded. Proceeding to apply configuration. This might take a while depending on amount of configuration in buffer. Please avoid other configuration changes during this time. Configuration committed by user 'admin' using Commit ID : 1000000001 switch(config-dual-stage)# switch(config-dual-stage)# commit Confirming commit for trial session. switch(config-dual-stage)# **Example:** switch(config-dual-stage)# hostname example-switch switch(config-dual-stage)# commit Verification Succeeded. Proceeding to apply configuration. This might take a while depending on amount of configuration in buffer. Please avoid other configuration changes during this time. Configuration committed by user 'admin' using Commit ID : 1000000002 example-switch(config-dual-stage)#	• *seconds*: Commits the configuration in global configuration mode on a trial basis for a minimum of 30 seconds and a maximum of 65535 seconds. **Note** If you enter a trial period, run the commit co command, the switch reverts to the previou
Step 4	**Example:** switch(config-dual-stage)# router bgp 64515.46 switch(config-dual-stage-router)# switch(config-dual-stage-router)# router-id 141.8.139.131 switch(config-dual-stage-router)#	Run any feature-related commands that are supported in this configuration mode.
Step 5	**show configuration** **Example:** switch(config-dual-stage-router)# show configuration ! Cached configuration ! router bgp 64515.46 router-id 141.8.139.131	Displays the target configuration. **Note** You can run this command only in the dual
Step 6	**commit** [**confirmed** *seconds*] **Example:** switch(config-dual-stage-router)# commit Verification Succeeded. Proceeding to apply configuration. This might take a while depending on amount of configuration in buffer. Please avoid other configuration changes during this time. Configuration committed by user 'admin' using Commit ID : 1000000003	Commits changes to the running configuration.

	Command or Action	Purpose

Step 7	(Optional) **show configuration commit** [**changes**] *commit-id*
	Example:
	`switch(config-dual-stage-router)# show configuration commit changes 1000000003`
	`*** /bootflash/.dual-stage/1000000003.tmp Fri Mar 19 10:59:00 2021`
	`--- /bootflash/.dual-stage/1000000003 Fri Mar 19 10:59:05 2021`
	`***************`
	`*** 378,383 ****`
	`--- 378,385 ----`
	`line console`
	`line vty`
	`boot nxos bootflash:/nxos64.10.1.1.44.bin`
	`+ router bgp 64515.46`
	`+ router-id 141.8.139.131`
	`xml server timeout 1200`
	`no priority-flow-control override-interface mode off`
	Example:
	`switch(config-dual-stage)# show configuration commit 1000000003`
	`feature bgp`
	`router bgp 64515.46`
	`router-id 141.8.139.131`
	`.`
	`.`
	`.`
		Displays commit-related information.
		Only the last 50 commits or the commit files stored in the reserved disk space are saved. The reserved disk space is 20 MB. All the commit sessions will be removed when you reload the switch. However, the commit IDs are not removed. Also, these commit IDs will not be removed in case of write, erase, and reload.
		Use the **show configuration commit changes** *commit-id* command to view only the changes in the current session of the commit you specify.
		Use the **show configuration commit** *commit-id* command to view the complete configurations in the commit that you specify, along with few class-map policies. These class-map policies are not new policies but hidden policies. To view the hidden policies, use the **show run all** command.

Step 8	(Optional) **save configuration** *filename* **Example:** `switch(config-dual-stage)# save configuration bootflash:test.cfg`	Saves the target configurations to a separate file without committing them to the running configuration. **Note** • You can load the target configurati • You can view the configuration file • Some of the user-specific informat • Configs saved in dual stage mode is **file** <> and not using **#show file** <>

Step 9	(Optional) **load** *filename*
	Example:
	`switch (config-dual-stage)# show configuration`
	`! Cached configuration`
	`switch (config-dual-stage)# load test.cfg`
	`switch (config-dual-stage-router)# show configuration`
	`! Cached configuration`
	`!`
		Loads a target configuration that you saved. After loading a file, you can modify it or commit it to the running configuration. To save the changes, use the **save configuration** *filename* command.
		You can load a target configuration that you saved using only the **save configuration** *filename* command.

	Command or Action	Purpose
	`router bgp 1` `switch(config-dual-stage-router)#`	

Step 10	(Optional) **clear configuration**
	Example:
	`switch(config-dual-stage)# show configuration`
	`! Cached configuration`
	`!`
	`router bgp 64515.46`
	`router-id 141.8.139.131`
	`switch (config-dual-stage)# clear configuration`
	`switch (config-dual-stage)# show configuration`
	`! Cached configuration`
	`switch (config-dual-stage)#`
		Clears changes made to the target configuration without terminating the configuration session. It deletes any configuration changes that aren't committed.

Step 11	**end**
	Example:
	`switch(config-dual-stage-if)# end`
	`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]`
		Exits the global dual stage configuration mode.
		If you end a configuration session without committing the configuration changes, you'll be prompted to save changes, discard changes, or cancel the action:
		• Yes: Commits the configuration changes and exit configuration mode
		• No: Exits the configuration mode without committing the configuration changes
		• Cancel: Remains in configuration mode without committing the configuration changes
		Note
		• If you choose to exit when a confirm c still chose to exit, the trial configurati
		• If the default session times out before the session. In this case, no warning m

Step 12	**show configuration dual-stage sessions**
	Example:
	`switch(config-dual-stage)# show configuration dual-stage sessions`
	`SNo. Session Line User Date`
	`---- ---------- ---------- ---------- ----------`
	`1 8671-17101913 /dev/ttyS0 admin Wed Feb 17 10:56:00 2021`
	`switch(config-dual-stage)# end`
	`switch# show configuration dual-stage sessions`
	`There are no active dual stage sessions`
	`switch#`
		Before you start a configuration session, you must check if there are other configuration sessions in progress. Only single user is allowed to enter the dual stage configuration mode. Therefore, you need to exit the previous session before starting a new one. There are as many as 32 interactive VSH sessions possible, and the show command displays the PID and line information of the dual stage session.
		Note Dual stage mode will be accessible only af

	Command or Action	Purpose

Step 13	**clear configuration commits diskspace**
	Example:
	`Southlake-2# clear configuration commits diskspace ?`
	`<1-20971> Number of Kilo Bytes of disk space to free`
	
	`Southlake-2# clear configuration commits diskspace 100`
	`Deleting 7 rollback points from '1000005557' to '1000005563'`
	`101 KB of disk space will be freed.`
	`Continue with deletion (yes/no)? [no] y`
	`Southlake-2#`
		You can delete the oldest configuration commitIDs by entering the clear configuration commits command. The clear configuration commits command must be followed by either the amount of disk space to reclaim or the number of commitIDs to delete. To reclaim disk space from the oldest commitIDs, enter the clear configuration commits command followed by the diskspace keyword and number of kilobytes to reclaim.

Step 14	**clear configuration commits oldest**
	Example:
	`switch(config-dual-stage)# clear configuration commits oldest 10`
	`Deleting 10 rollback points '1000000030' to '1000000039'`
	`125 KB of disk space will be freed.`
	`Continue with deletion (yes/no)? [no] n`
		To delete a specific number of the oldest commitIDs, enter the clear configuration commits command followed by the oldest keyword and number of commitIDs to delete.

Step 15	**Show configuration failed**
	Example:
	`switch(config-dual-stage-if)# commit`
	`Verification Succeeded.`
	
	`Proceeding to apply configuration. This might take a while depending on amount of configuration in buffer.`
	`Please avoid other configuration changes during this time.`
	`Failed to commit one or more configuration items.`
	`Commit Failed, Rolling back ...`
	`switch(config-dual-stage)#`
	`switch(config-dual-stage)# show configuration failed`
	`` `config terminal` ``
	`` `router bgp 100 ` ``
	`` `neighbor 2.2.2.2 ` ``
	`` `bfd ` ``
	`Syntax error while parsing 'bfd '`
	
	`` `neighbor 3.3.3.3 ` ``
	`` `bfd ` ``
	`Syntax error while parsing 'bfd '`
	
	`` `interface port-channel23 ` ``
	`` `bfd ` ``
	`Syntax error while parsing 'bfd '`
	
	`` `end` ``
	`` `end` ``
	`switch(config-dual-stage)#`
		Configuration changes are semantically verified during the commit operation, and begins the actual backend commit once the verification is succeeded. A message appears if one or more configuration entry fails while committing. To display an error message and description for a failed configuration, enter the show configuration failed command. This will display the configuration block that failed in last commit. Configuration block is to preserve the configuration context.

	Command or Action	Purpose

Step 16	**show configuration failed noerrors**
	Example:
	`switch(config-dual-stage)# show configuration failed noerror`
	`router bgp 100`
	`  neighbor 2.2.2.2`
	`    bfd`
	`  neighbor 3.3.3.3`
	`    bfd`
	`interface port-channel23`
	`  bfd`
	`switch(config-dual-stage)#`
		To display only the errored config (without a description) for a failed configuration block, enter the show configuration failed noerrors command.

Step 17	**load configuration failed commit**
	Example:
	`switch(config-dual-stage)# load configuration failed commit`
	`switch(config-dual-stage-if)# sh configuration`
	`! Cached configuration`
	`!`
	`router bgp 100`
	` neighbor 2.2.2.2`
	`    bfd`
	`!`
	`interface port-channel23`
	` bfd`
	`switch(config-dual-stage-if)#`
		If the router displays a verification failure message during commit, the configuration changes are not lost. You can modify the target configuration and commit again. But, if the router displays a configuration failure message (backend error) when you attempt to commit a configuration change, the configuration session will reset. But, while you remain in dual-stage configuration mode, you can reload the failed configuration block into the target configuration, correct the errors, and commit the changes.
		To load a failed configuration, enter the load configuration failed commit command. After recovery, correct and commit the configuration or save it to a file to avoid losing it. Please note that while loading, syntactically wrong configurations will get ignored. You can use ‘show configuration’ to view the target configuration.

Abort the two-stage configuration commit mode

When you abort a configuration session, uncommitted changes are discarded and the configuration session ends.

No warning appears before the configuration changes are deleted.

```
switch(config-dual-stage)# router bgp 1
switch(config-dual-stage-router)# neighbor 192.0.2.4
switch(config-dual-stage-router-neighbor)# remote-as 1
switch(config-dual-stage-router-neighbor)# show configuration
! Cached configuration
!
router bgp 1
neighbor 192.0.2.4
remote-as 1
switch(config-dual-stage-router-neighbor)# show run bgp

!Command: show running-config bgp
```

```
!Running configuration last done at: Wed Mar 17 16:17:40 2021
!Time: Wed Mar 17 16:17:55 2021

version 10.1(2) Bios:version
feature bgp


switch(config-dual-stage-router-neighbor)# abort
switch# show run bgp

!Command: show running-config bgp
!Running configuration last done at: Wed Mar 17 16:18:00 2021
!Time: Wed Mar 17 16:18:04 2021

version 10.1(2) Bios:version
feature bgp

switch#
```

Display commit IDs

At each successful commit, the commit ID is displayed in the syslog. The total number of commit IDs saved in the system depends on the configuration size and the disk space available. However, the maximum number of commit IDs stored at any given time is 50.

Use the **show configuration commit list** command to view information about the last 50 commit IDs. Each entry shows the user who committed configuration changes, the connection used to execute the commit, and commit ID timestamp.

```
switch# show configuration commit list
SNo. Label/ID     User     Line         Client     Time Stamp
~~~~ ~~~~~~~~~~~~ ~~~~~~~~ ~~~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~
1    1000000001   admin    /dev/ttyS0   CLI        Wed Jul 15 15:21:37 2020
2    1000000002   admin    /dev/ttyS0   Rollback   Wed Jul 15 15:22:15 2020
3    1000000003   admin    /dev/pts/0   CLI        Wed Jul 15 15:23:08 2020
4    1000000004   admin    /dev/pts/0   Rollback   Wed Jul 15 15:23:46 2020
```

Rollback capability

You can rollback the configuration to any of the previous successful commits.

Use the **rollback configuration** command to rollback to any of the last 50 commits.

```
switch# rollback configuration to ?
1000000015
1000000016
1000000017

:
:

switch#

Each commit ID acts as a (checkpoint or) rollback point. You can rollback to any given
commit ID. When you roll back the configuration to a specific rollback point, you undo
all configuration changes made during the session identified by the commitID for that
rollback point, and you undo all configuration changes made after that point. The
rollback process rolls back the configuration and commits the rolled-back configuration.
The rollback process also creates a new rollback point (commit ID)so that you can roll
```

```
back the configuration to the previous configuration.


switch(config-dual-stage)# rollback configuration to 1000000002
Rolling back to commitID :1000000002
ADVISORY: Rollback operation started...
Modifying running configuration from another VSH terminal in parallel
is not recommended, as this may lead to Rollback failure.

Configuration committed by rollback using Commit ID : 1000000004
switch(config-dual-stage)#
```

View current session configurations

You can view the current session configuration using the **show configuration** command.

This command is supported only in the dual-stage mode. The session configuration is cleared if a commit fails.

```
switch(config-dual-stage-cmap)# show configuration
! Cached configuration
!
class-map type control-plane match-any copp-s-ipmcmiss
class-map type control-plane match-any copp-s-l2switched
class-map type control-plane match-any copp-s-l3destmiss
switch(config-dual-stage-cmap)#

If there is no configuration, the following message appears:

switch(config-dual-stage)# show configuration
! Cached configuration
switch(config-dual-stage)# commit
No configuration changes to commit.
switch(config-dual-stage)#
```