



Performing Configuration Replace

This chapter includes the following sections:

- [About Configuration Replace and Commit-timeout, on page 1](#)
- [Overview, on page 1](#)
- [Guidelines and Limitations for Configuration Replace, on page 3](#)
- [Recommended Workflow for Configuration Replace, on page 5](#)
- [Performing a Configuration Replace, on page 6](#)
- [Verifying Configuration Replace, on page 8](#)
- [Examples for Configuration Replace, on page 8](#)

About Configuration Replace and Commit-timeout

The configuration replace feature enables you to replace the running configuration of the Cisco Nexus switch with the user provided configuration without reloading the device. The device reload may be required only when a configuration itself requires a reload. The running configuration file that is provided by the user should be taken using copy running file. Unlike **copy file: to running**, the configuration replace feature is not a merge operation. This feature replaces the entire running configuration with a new configuration that is provided by the user. If there is a failure in the configuration replace, the original configuration is restored in the switch. From Cisco NX-OS Release 9.3(1), **best-effort** option is introduced. This option enables the configuration replace to execute the full patch despite any error in the commands and the original configuration is not restored in the switch.

The commit-timeout feature enables you to rollback to the previous configuration after successfully performing the configuration replace operation. If the commit timer expires, the rollback operation is automatically initiated.



Note

- You must provide a valid running configuration that has been received with the Cisco NX-OS device. It should not be a partial configuration.
-

Overview

The configuration replace feature has the following operation steps:

- Configuration replace intelligently calculates the difference between the current running-configuration and the user-provided configuration in the Cisco Nexus switch and generates a patch file which is the difference between the two files. You can view this patch file which includes a set of configuration commands.
- Configuration replace applies the configuration commands from the patch file similarly to executing commands.
- The configuration rolls back to or restores the previous running configuration under the following situations:
 - If there is a mismatch in the configuration after the patch file has been applied.
 - If you perform the configuration operation with a commit timeout and the commit timer expires.
- The configuration does not roll back to or does not restore the previous running configuration when the best-effort option is used. This option enables the configuration replace to execute the full patch despite any error in the commands and will not roll back to the previous configuration.
- You can view the exact configuration that caused a failure using the **show config-replace log exec** command.
- Restore operations that fail while restoring the switch to the original configuration, are not interrupted. The restore operation continues with the remaining configuration. Use the **show config-replace log exec** command to list the commands that failed during the restore operation.
- If you enter the **configure replace commit** command before the timer expires, the commit timer stops and the switch runs on the user provided configuration that has been applied through the configuration replace feature.
- If the commit timer expires, roll back to the previous configuration is initiated automatically.
- In Cisco NX-OS Release 9.3(1), semantic validation support is added for the configuration replace. This semantic validation is done as part of the precheck in configuration replace. The patch gets applied only when the semantic validation is successful. After applying the patch file, configuration replace triggers the verification process. The configuration replace compares the running-configuration with the user configuration file during the verification process. If there is a mismatch, it restores the device to the original configuration.

The differences between configuration replace and copying a file to the running-configuration are as follows:

Configuration Replace	Copying a file
The configure replace <i><target-url></i> command removes the commands from the current running-configuration that are not present in the replacement file. It also adds commands that need to be added to the current running-configuration.	The copy <i><source-url></i> running-config command is a merge operation which preserves all the commands from, both the source file and the current running-configuration. This command does not remove the commands from the current running-configuration that are not present in the source file.
You must use a complete Cisco NX-OS configuration file as the replacement file for the configure replace <i><target-url></i> command.	You can use a partial configuration file as a source file for the copy <i><source-url></i> running-config command.

Benefits of Configuration Replace

The benefits of configuration replace are:

- You can replace the current running-configuration file with the user-provided configuration file without having to reload the switch or manually undo CLI changes to the running-configuration file. As a result, the system downtime is reduced.
- You can revert to the saved Cisco NX-OS configuration state.
- It simplifies the configuration changes by allowing you to apply a complete configuration file to the device, where only the commands that need to be added or removed are affected. The other service and configurations that are not modified remain untouched.
- If you configure the commit-timeout feature, you can rollback to the previous configuration even when the configuration replace operation has been successful.

Guidelines and Limitations for Configuration Replace

The configuration replace feature has the following configuration guidelines and limitations:

- The configuration replace feature is supported on Cisco Nexus 3000 Series and Cisco Nexus 9000 Series switches.
- Only one user can perform the configuration replace, checkpoint, and rollback operations, or copy the running-configuration to the startup configuration at the same time. Parallel operations such as operations via multiple Telnet, SSH, or NX-API sessions are not supported. The multiple configuration replace or rollback request is serialized, for example, only after the first request is completed, processing of the second request begins.
- You are not allowed to initiate another configuration replace operation when the commit timer is running. You must either stop the timer by using the **configure replace commit** command or wait until the commit timer expires before you initiate another configuration replace operation.
- For a successful configuration replace operation, sequence number must be present for all ACE entries in ACL in the target configuration file.
- The commit-timeout feature is initiated only if you perform the configuration replace operation with the commit-timeout. The timer value range is from 30 to 3600 seconds.
- The user provided configuration file must be the valid show running-configuration output that is taken from the Cisco NX-OS device (copy run file). The configuration cannot be a partial configuration and must include mandated commands, such as user admin and so on.
- We do not recommend a configuration replace operation that is performed on the configuration file that is generated across the software version because this operation could fail. A new configuration file must be regenerated whenever there is change in the software version.
- We recommend that you do not change any configuration from others sessions if the configuration replace operation is in progress because it could cause the operation to fail.
- Note the following about the configuration replace feature:
 - The configuration replace feature does not support features that require a reload. One such feature is: system vlan reserve.

- The configuration replace feature is not supported on Cisco Nexus 9500 platform switches with -R line cards.
 - The configuration replace feature could fail if the running configuration includes the **feature-set mpls** or the **mpls static range** commands and tries to move to a configuration without MPLS or modifies the label range.
 - The configuration replace feature does not support autoconfigurations.
- If the line card to which the configuration replace feature is applied is offline, the configuration replace operation fails.
 - An ITD service must be shut down (**shutdown**) prior to making ITD changes with the configuration replace feature.
 - Sequence number is mandatory for CLI **ip community-list** and **ip as-path access-list** commands. Without a sequence number, the configuration replace operation fails.
 - If your configurations demand reloading the Cisco NX-OS device in order to apply the configuration, then you must reload these configurations after the configuration replace operation.
 - The order of the commands in the user provided configuration file must be the same as those commands in the running configuration of the Cisco Nexus switch.
 - The user configuration file to which you need to replace the running configuration on the switch using CR should be generated from the running-config of the switch after configuring the new commands. The user configuration file should not be manually edited with the CLI commands and the sequence of the configuration commands should not be altered.
 - The semantic validation is not supported in 4-Gig memory platforms.
 - When different versions of a feature are present in the running configuration and user configuration (for example: VRRPv2 and VRRPv3), semantic validation option does not work as expected. This issue is a known limitation.
 - Beginning from Cisco NX-OS Release 10.3(1)F, the configuration replace feature does not support feature app-hosting.
 - Beginning from Cisco NX-OS Release 10.4(2)F, the configuration replace feature is supported for LDAP on Cisco NX-OS devices.
 - Beginning from Cisco NX-OS Release 10.4(2)F, for non-case sensitive commands, if there is a letter case distinction between the commands in running config and candidate-config files, then the output of **config replace show-patch** displays both the commands due to the difference in letter case.
 - Beginning from Cisco NX-OS Release 10.4(3)F, you can also use polymorphic commands in candidate configuration to perform configuration replace.
 - Clear text passwords are allowed in the case of configuration replace candidate-config file as the user database gets synced between SNMP and AAA (Security).

Guidelines and Limitations for Configuration Replace for PBR Commands

The content of this section is applicable from Cisco NX-OS Release 10.4(3)F.

None of the PBR commands can coexist under the same parent route-map. If the mutually exclusive PBR commands are given under the same route-map in the candidate config, the config-replace patch is generated only for the last command variant under the route-map and is applied after CR operation.

The following table depicts a few use cases.

Use Case	Candidate Config	Converted Candidate Config
<p>Use Case 1: Multiple command variants - Only the last command variant is retained</p> <p>The candidate config is automatically converted as shown in the third column before the CR patch is generated.</p>	<pre>route-map rmap1 permit 10 set ip next-hop 1.1.1.1 2.2.2.2 set ipv6 next-hop 3::3 set ip next-hop verify-availability 4.4.4.4 set ip next-hop verify-availability 5.5.5.5 set ip vrf green next-hop 6.6.6.6 set ip vrf blue next-hop 7.7.7.7 8.8.8.8</pre>	<pre>route-map rmap1 permit 10 set ip vrf green next-hop 6.6.6.6 set ip vrf blue next-hop 7.7.7.7 8.8.8.8</pre>
<p>Use Case 2: Commands comprising track IDs - Only the last command variant with same next-hop and different track ID is retained</p> <p>For the verify-availability commands, track ID cannot be modified for the same next-hop. The candidate config is automatically converted as shown in the third column before the CR patch is generated.</p>	<pre>route-map test permit 10 set ip next-hop verify-availability 1.1.1.1 track 1 set ip next-hop verify-availability 2.2.2.2 track 20 set ip next-hop verify-availability 2.2.2.2 track 30 set ip next-hop verify-availability 2.2.2.2 track 40 set ip next-hop verify-availability 3.3.3.3 track 3</pre>	<pre>route-map test permit 10 set ip next-hop verify-availability 1.1.1.1 track 1 set ip next-hop verify-availability 2.2.2.2 track 40 set ip next-hop verify-availability 3.3.3.3 track 3</pre>

Recommended Workflow for Configuration Replace

The following workflow is the recommended workflow for configuration replace:



Note

- This workflow needs to be the same in the candidate config.
- Default configuration in the candidate config is not supported.

1. Generate a configuration file by first applying the configurations on a Cisco Nexus Series device and then use the **show running-configuration** output as the configuration file. Use this file to make configuration modifications as required. Then use this generated or updated configuration file to perform configuration replace.
2. View and verify the patch file by executing the **configure replace <file> show-patch** command. This is an optional step.

3. Run the configuration replace file either using or skipping the **commit-timeout** *<time>* feature. Based on your requirements, you can perform one of the following steps:
 - Run **configure replace** *<file>* **verbose** to see the commands that get executed with configuration replace on the console.
 - Run the **configure replace** [**bootflash/scp/sftp**] *<user-configuration-file>* **verbose** **commit-timeout** *<time>* commands to configure the commit time.
4. Run the **configure replace commit** command to stop the commit timer. This step is necessary if you have run the configuration replace operation with the commit-timeout feature.
5. Configuration replace performs a precheck that includes the semantic validation of the configuration. The configuration replace operation fails if there is an error. Use the **show config-replace log verify** command to see the details of the failed configurations. After applying the patch file, configuration replace triggers the verification process. The configuration replace compares the running-configuration with the user configuration file during the verification process. If there is a mismatch, it restores the device to the original configuration. Use the **show config-replace log verify** command to see the mismatched configurations.
6. You can perform the following configuration replace operations in Cisco NX-OS Release 9.3(1):
 - Configuration replace without the semantic validation and without best-effort mode.
 - Configuration replace without the semantic validation and with best-effort mode.
 - Configuration replace with the semantic validation and without best-effort mode.
 - Configuration replace with the semantic validation and with best-effort mode.

Performing a Configuration Replace

To perform configuration replace, do the following:

SUMMARY STEPS

1. **configure replace** { *<uri_local>* | *<uri_remote>* } [**verbose** | **show-patch**]
2. **configure replace** [**bootflash / scp / sftp**] *<user-configuration-file>* **show-patch**
3. **configure replace** [**bootflash / scp / sftp**] *<user-configuration-file>* **verbose**
4. **configure replace** *<user-configuration-file>* [**best-effort**]
5. **configure replace** *<user-configuration-file>* [**verify-and-commit**]
6. **configure replace** *<user-configuration-file>* [**verify-only**]
7. (Optional) **configure replace** [**bootflash / scp / sftp**] *<user-configuration-file>* **verbose** **commit-timeout** *<time>*
8. (Optional) **configure replace** [**commit**]
9. (Optional) **configure replace** [**bootflash/scp/sftp**] *<user-configuration-file>* *non-interactive*

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>configure replace { < uri_local > < uri_remote > } [verbose show-patch]</code>	Performs configuration replace. If you make the configuration changes through any sessions when configuration replace is in progress, the configuration replace operation fails. If you send a configuration replace request when one configuration request is already in progress, then it gets serialized.
Step 2	<code>configure replace [bootflash / scp / sftp] < user-configuration-file > show-patch</code>	Displays the differences between the running-configuration and the user-provided configuration. Note <ul style="list-style-type: none"> • This command does not encrypt plain text password. • This command can still show a patch even after configuration replace is successful for CLI snmp-server traps commands.
Step 3	<code>configure replace [bootflash / scp / sftp] < user-configuration-file > verbose</code>	Replaces the configuration on the switch with the new user configuration that is provided by the user. Configuration replace is always atomic.
Step 4	<code>configure replace <user-configuration-file> [best-effort]</code>	Replaces the configuration on the switch with the new user configuration and enables the configuration replace with semantic validation. The best-effort option enables the configuration replace to execute the full patch despite any error in the commands and also make sure that the previous configuration is not rolled back.
Step 5	<code>configure replace <user-configuration-file> [verify-and-commit]</code>	Replaces the configuration on the switch with the new user configuration and enables the configuration replace with semantic validation. The verify-and-commit option is used for enabling the semantic validation. Patch will be executed only if semantic validation of the full patch gets passed. You can use the best-effort option or the verify-and-commit option or both the options at the same time.
Step 6	<code>configure replace <user-configuration-file> [verify-only]</code>	Shows only the patch and does Semantic validation on the patch, and display the results. The patch does not get applied to the system.
Step 7	(Optional) <code>configure replace [bootflash / scp / sftp] < user-configuration-file > verbose commit-timeout <time></code>	Configures the commit time in seconds. The timer starts after the configuration replace operation is successfully completed.

	Command or Action	Purpose
Step 8	(Optional) configure replace [commit]	Stops the commit timer and continues the configuration replace configuration. Note This step is applicable only if you have configured the commit-timeout feature. Note To rollback to the previous configuration, you must wait for the expiry of the commit timer. Once the timer expires, the switch is automatically rolled back to the previous configuration.
Step 9	(Optional) configure replace [bootflash/scp/sftp] <i><user-configuration-file> non-interactive</i>	There is no user prompt in maintenance mode. The yes user-confirmation is taken by default, and rollback proceeds. You can use the non-interactive option only in the maintenance mode.

Verifying Configuration Replace

To check and verify configuration replace and its status, use the commands that are outlined in the table:

Table 1: Verifying Configuration Replace

Command	Purpose
configure replace [bootflash/scp/sftp]<user-configuration-file> show-patch	Displays the difference between the running-configurations and user-provided configurations.
show config-replace log exec	Displays a log of all the configurations executed and those that failed. In case of an error, it displays an error message against that configuration.
show config-replace log verify	Displays the configurations that failed, along with an error message. It does not display configurations that were successful.
show config-replace status	Displays the status of the configuration replace operations, including in-progress, successful, and failure. If you have configured the commit-timeout feature, the commit and timer status and the commit timeout time remaining is also displayed.

Examples for Configuration Replace

See the following configuration examples for configuration replace:

- Use the **configure replace bootflash: <file> show-patch** CLI command to display the difference between the running-configurations and user-provided configurations.

```
switch(config)# configure replace bootflash:<file> show-patch
Collecting Running-Config
Converting to checkpoint file
#Generating Rollback Patch
!!
no role name abc
```

- Use the **configure replace bootflash: <file> verbose** CLI command to replace the entire running-configuration in the switch with the user-configuration.

```
switch(config)# configure replace bootflash:<file> verbose
Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Note: Applying config parallely may fail Rollback verification
Collecting Running-Config
#Generating Rollback Patch
Executing Rollback Patch
=====
config t
no role name abc
=====
Generating Running-config for verification
Generating Patch for verification

Rollback completed successfully.
```

Sample Example with adding of BGP configurations.

```
switch(config)# sh run | section bgp
switch(config)# sh file bootflash:file | section bgp
feature bgp
router bgp 1
  address-family ipv4 unicast
  neighbor 1.1.1.1
switch(config)#
switch(config)# configure replace bootflash:file verbose
Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Note: Applying config parallely may fail Rollback verification
Collecting Running-Config
#Generating Rollback Patch
Executing Rollback Patch
=====
config t
feature bgp
router bgp 1
address-family ipv4 unicast
neighbor 1.1.1.1
=====
Generating Running-config for verification
Generating Patch for verification

Rollback completed successfully.

switch(config)# sh run | section bgp
feature bgp
router bgp 1
  address-family ipv4 unicast
  neighbor 1.1.1.1
```

```

Sample Example with ACL
switch(config)# configure replace bootflash:run_1.txt
Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Note: Applying config parallelly may fail Rollback verification
Collecting Running-Config
#Generating Rollback Patch
Executing Rollback Patch
=====
config t
no ip access-list nexus-50-new-xyz
ip access-list nexus-50-new-xyz-jkl-abc
10 remark Newark
20 permit ip 17.31.5.0/28 any
30 permit ip 17.34.146.193/32 any
40 permit ip 17.128.199.0/27 any
50 permit ip 17.150.128.0/22 any
=====
Generating Running-config for verification
Generating Patch for verification

Rollback completed successfully.

switch(config)#

switch(config)# show run aclmgr | sec nexus-50-new-xyz-jkl-abc
ip access-list nexus-50-new-xyz-jkl-abc
 10 remark Newark
 20 permit ip 17.31.5.0/28 any
 30 permit ip 17.34.146.193/32 any
 40 permit ip 17.128.199.0/27 any
 50 permit ip 17.150.128.0/22 any

```

- Use the **configure replace bootflash:user-config.cfg verify-only** CLI command to generate and verify the patch semantically.

```

switch(config)# configure replace bootflash:user-config.cfg verify-only

Version match between user file and running configuration.
Pre-check for User config PASSED
Collecting Running-Config
Converting to checkpoint file
Generating Rollback Patch
Validating Patch
=====
`config t `
`interface Ethernet1/1`
`shutdown`
`no switchport trunk allowed vlan`
`no switchport mode`
`no switchport`
`exit`
Skip non dme command for CR validation
`interface Vlan1`
`shutdown`
`interface Ethernet1/1`
`shutdown`
`no switchport`
`ip address 1.1.1.1/24`
`exit`
Skip non dme command for CR validation
=====

```

```
Patch validation completed successful
switch(config)#
```

- Use the **configure replace bootflash:user-config.cfg best-effort verify-and-commit** CLI command to replace the switch running configuration with the given user configuration after performing the semantic validation on patch.

```
switch(config)# configure replace bootflash:user-config.cfg best-effort verify-and-commit
```

```
Version match between user file and running configuration.
Pre-check for User config PASSED
ADVISORY: Config Replace operation started...
Modifying running configuration from another VSH terminal in parallel
is not recommended, as this may lead to Config Replace failure.
```

```
Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Collecting Running-Config
Generating Rollback Patch
```

```
Validating Patch
Patch validation completed successful
Executing Rollback Patch
During CR operation, will retain L3 configuration
when vrf member change on interface
Generating Running-config for verification
Generating Rollback Patch
```

```
Configure replace completed successfully. Please run 'show config-replace log exec' to
see if there is any configuration that requires reload to take effect.
```

```
switch(config)#
```

- Use the **show config-replace log exec** CLI command to check all the configuration that is executed and failures if any.

```
switch(config)# show config-replace log exec
Operation          : Rollback to Checkpoint File
Checkpoint file name : .replace_tmp_28081
Scheme             : tmp
Rollback done By    : admin
Rollback mode       : atomic
Verbose             : enabled
Start Time          : Wed, 06:39:34 25 Jan 2017
```

```
-----
time: Wed, 06:39:47 25 Jan 2017
Status: SUCCESS
End Time           : Wed, 06:39:47 25 Jan 2017
Rollback Status    : Success
```

```
Executing Patch:
-----
switch#config t
switch#no role name abc
```

- Use the **show config-replace log verify** CLI command to check the failed configuration if any.

```
switch(config)# show config-replace log verify
Operation          : Rollback to Checkpoint File
Checkpoint file name : .replace_tmp_28081
Scheme             : tmp
```

```
Rollback done By      : admin
Rollback mode        : atomic
Verbose              : enabled
Start Time           : Wed, 06:39:34 25 Jan 2017
End Time             : Wed, 06:39:47 25 Jan 2017
Status               : Success
```

```
Verification patch contains the following commands:
```

```
-----
!!
! No changes
-----
```

```
time: Wed, 06:39:47 25 Jan 2017
Status: SUCCESS
```

- Use the **show config-replace status** CLI command to check the status of configuration replace.

```
switch(config)# show config-replace status
Last operation : Rollback to file
Details:
  Rollback type: atomic replace_tmp_28081
  Start Time: Wed Jan 25 06:39:28 2017
  End Time: Wed Jan 25 06:39:47 2017
  Operation Status: Success
switch(config)#
```

Configure Replace might fail when the manually created configuration is used instead of the configuration generated from the switch. The reason for possible failures is the potential difference in the default configuration that isn't shown in the show running configuration. Refer to the following examples:

If the power redundant command is the default command, it doesn't get displayed in the default configuration. But it's displayed when you use the **show run all** command. See the following example:

```
switch# show run all

!Command: show running-config all
!Running configuration last done at: Tue Nov 12 11:07:44 2019
!Time: Tue Nov 12 11:16:09 2019

version 9.3(1) Bios:version 05.39
power redundancy-mode ps-redundant
no hardware module boot-order reverse
<snip>
hostname n9k13
```

The power redundant command isn't shown in the show running configuration command out. See the following example:

```
!Command: show running-config
!Running configuration last done at: Tue Nov 12 11:07:44 2019
!Time: Tue Nov 12 11:17:24 2019

version 9.3(1) Bios:version 05.39
hostname n9k13
```

When the **power redundancy-mode ps-redundant** command is added in the user configuration for the configure replace; then the verification/commit might fail. See the following example:

```
switch# show file bootflash:test

!Command: show running-config
!Running configuration last done at: Tue Nov 12 10:56:49 2019
```

```
!Time: Tue Nov 12 11:04:57 2019

version 9.3(1) Bios:version 05.39
power redundancy-mode ps-redundant
hostname n9k13
```

The **power redundancy-mode ps-redundant** command will not be shown in the show running after configure replace; therefore it will be considered as “missing” and the CR will fail. An example is given below.

```
switch# config replace bootflash:test verify-and-commit
```

```
Version match between user file and running configuration.
Pre-check for User config PASSED
ADVISORY: Config Replace operation started...
Modifying running configuration from another VSH terminal in parallel
is not recommended, as this may lead to Config Replace failure.
```

```
Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Collecting Running-Config
.Generating Rollback Patch
```

```
Validating Patch
Patch validation completed successful
Executing Rollback Patch
During CR operation,will retain L3 configuration
when vrf member change on interface
Generating Running-config for verification
Generating Rollback Patch
Executing Rollback Patch
During CR operation,will retain L3 configuration
when vrf member change on interface
Generating Running-config for verification
Generating Patch for verification
Verification failed, Rolling back to previous configuration
Collecting Running-Config
Cleaning up switch-profile buffer
Generating Rollback patch for switch profile
Executing Rollback patch for switch profiles. WARNING - This will change the
configuration of switch profiles and will also affect any peers if configured
Collecting Running-Config
Generating Rollback Patch
Rollback Patch is Empty
Rolling back to previous configuration is successful
```

```
Configure replace failed. Use 'show config-replace log verify' or 'show config-replace
log exec' to see reasons for failure
```

```
n9k13# show config-replace log verify
Operation : Config-replace to user config
Checkpoint file name : .replace_tmp_31849
Scheme : tmp
Cfg-replace done By : agargula
Cfg-replace mode : atomic
Verbose : disabled
Start Time : Tue, 11:20:59 12 Nov 2019
Start Time UTC : Tue, 10:20:59 12 Nov 2019
-----
End Time : Tue, 11:21:28 12 Nov 2019
End Time UTC : Tue, 10:21:28 12 Nov 2019
Status : Failed
```

```
Verification patch contains the following commands:
-----
```

```
!!
Configuration To Be Added Missing in Running-config
=====
!
power redundancy-mode ps-redundant

Undo Log
-----
End Time : Tue, 11:21:32 12 Nov 2019
End Time UTC : Tue, 10:21:32 12 Nov 2019
Status : Success
n9k13#
```

In the above example, CR will consider the default commands that are missing and will therefore fail.