Configuration Compliance, Release 12.2.1

Table of Contents

New and Changed Information

The following table provides an overview of the significant changes up to this current release. The table does not provide an exhaustive list of all changes or of the new features up to this release.

Release Version	Feature	Description
There were no major changes from the previous release.		

Configuration Compliance

The entire intent or expected configuration defined for a given switch is stored in NDFC. When you want to push this configuration down to one or more switches, the configuration compliance (CC) module is triggered. CC takes the current intent, the current running configuration, and then comes up with the set of configurations that are required to go from the current running configuration to the current expected configuration so that everything will be In-Sync.

When performing a software or firmware upgrade on the switches, the current running configuration on the switches is not changed. Post upgrade, if CC finds that the current running configuration does not have the current expected configuration or intent, it reports an Out-of-Sync status. There is no auto deployment of any configurations. You can preview the diffs that will get deployed to get one or more devices back In-Sync.

With CC, the sync is always from the NDFC to the switches. There is no reverse sync. So, if you make a change out-of-band on the switches that conflicts with the defined intent in NDFC, CC captures this diff, and indicates that the device is Out-of-Sync. The pending diffs will undo the configurations done out-of-band to bring back the device In-Sync. Note that such conflicts due to out-of-band changes are captured by the periodic CC run that occurs every 60 minutes by default, or when you click the RESYNC option either on a per fabric or per switch basis. From Cisco NDFC Release 12.1.1e, the periodic CC runs every 24 hours. You can configure the custom interval with the range of 30-3600 minutes. This configuration can be done by navigating to **Server > Server Settings > LAN-Fabric**. Note that you can also capture the out-of-band changes for the entire switch by using the CC REST API. For more information, see *Cisco NDFC REST API Guide*.

To improve ease of use and readability of deployed configurations, CC in NDFC has been enhanced with the following:

- All displayed configurations in NDFC are easily readable and understandable.
- Repeated configuration snippets are not displayed.
- Pending configurations precisely show only the diff configuration.
- Side-by-side diffs has greater readability, integrated search or copy, and diff summary functions.

Top-level configuration commands on the switch that do not have any associated NDFC intent are not checked for compliance by CC. However, CC performs compliance checks, and attempts removal, of the following commands even if there is no NDFC intent:

- configure profile
- apply profile
- interface vlan
- interface loopback
- interface Portchannel
- Sub-interfaces, for example, interface EthernetX/Y.Z
- fex
- vlan <vlan-ids>

CC performs compliance checks, and attempts removal, of these commands only when **Data Center VXLAN EVPN** and **BGP Fabric** templates are used. On **External_Fabric** and **Classic LAN** templates,

top-level configuration commands on the switch, including the commands mentioned above, that do not have any associated NDFC intent are not checked for compliance by CC.

We recommend using the NDFC freeform configuration template to create additional intent and deploy these commands to the switches to avoid unexpected behavior

Now, consider a scenario in which the configuration that exists on the switch has no relationship with the configuration defined in the intent. Examples of such configurations are a new feature that has not been captured in the intent but is present on the switch or some other configuration aspect that has not been captured in the intent. Configuration compliance does not consider these configuration mismatches as a diff. In such cases, Strict Configuration Compliance ensures that every configuration line that is defined in the intent is the only configuration that exists on the switch. However, configuration such as boot string, rommon configuration, and other default configurations are ignored during strict CC checks. For such cases, the internal configuration compliance engine ensures that these config changes are not called out as diffs. These diffs are also not displayed in the **Pending Config** window. But, the Side-by-side diff utility compares the diff in the two text files and does not leverage the internal logic used in the diff computation. As a result, the diff in default configurations are highlighted in red in the **Side-by-side Comparison** window.

In NDFC, the diffs in default configurations are not highlighted in the **Side-by-side Comparison** window. The auto-generated default configuration that is highlighted in the **Running config** window is not visible in the **Expected config** window.

Any configurations that are shown in the **Pending Config** window are highlighted in red in the **Side-by-side Comparison** window if the configurations are seen in the **Running config** window but not in the **Expected config** window. Also, any configurations that are shown in the **Pending Config** window are highlighted in green in the **Side-by-side Comparison** window if the configurations are seen in the **Expected config** window but not in the **Running config** window. If there are no configurations displayed in the **Pending Config** window, no configurations are shown in red in the **Side-by-side Comparison** window.

All freeform configurations have to strictly match the show running configuration output on the switch and any deviations from the configuration will show up as a diff during **Recalculate & Deploy**. You need to adhere to the leading space indentations.

You can typically enter configuration snippets in NDFC using the following methods:

- User-defined profile and templates
- Switch, interface, overlay, and vPC freeform configurations
- Network and VRF per switch freeform configurations
- Fabric settings for Leaf, Spine, or iBGP configurations

The configuration format should be identical to the **show running configuration** of the corresponding switch. Otherwise, any missing or incorrect leading spaces in the configuration can cause unexpected deployment errors and unpredictable pending configurations. If any unexpected diffs or deployment errors are displayed, check the user-provided or custom configuration snippets for incorrect values.

If NDFC displays the "Out-of-Sync" status due to unexpected pending configurations, and this configuration is either unable to be deployed or stays consistent even after a deployment, perform the following steps to recover:

1. Check the lines of config highlighted under the **Pending Config** tab in the **Config Preview** window.
2. Check the same lines in the corresponding **Side-by-side Comparison** tab. This tab shows whether the diff exists in "intent", or "show run", or in both with different leading spaces. Leading spaces are highlighted in the **Side-by-side Comparison** tab.
3. If the pending configurations or switch with an out-of-sync status is due to any identifiable configuration with mismatched leading spaces in "intent" and "running configuration", this indicates that the intent has incorrect spacing and needs to be edited.
4. To edit incorrect spacing on any custom or user-defined policies, navigate to the switch and edit the corresponding policy:
    a. If the source of the policy is **UNDERLAY**, you will need to edit this from the Fabric settings screen and save the updated configuration.
    b. If the source is blank, it can be edited from the **View/Edit policies** window for that switch.
    c. If the source of the policy is **OVERLAY**, but it is derived from a switch freeform configuration. In this case, navigate to the appropriate **OVERLAY** switch freeform configuration and update it.
    d. If the source of the policy is **OVERLAY** or a custom template, perform the following steps:
        i. Choose **Settings > Server settings**, set the **template.in_use.check** property to **false** and uncheck the **Template In-Use Override** check box and **Save**. This allows the profiles or templates to be editable.
        ii. Edit the specific profile or template from the **Manage > Templates > Edit template properties** edit window, and save the updated profile template with the right spacing.
        iii. Click **Recalculate & Deploy** to recompute the diffs for the impacted switches.
        iv. After the configurations are updated, set the **template.in_use.check** property to **true** and check the **Template In-Use Override** check box and **Save**, as it slows down the performance of the NDFC system, specifically for **Recalculate & Deploy** operations.

If NDFC displays "NA" in the Config Status, the following guidelines apply:

- It is expected when the switch 'Mode' is 'Migration'. This could be due to some of the NDFC work flows. Follow the associated work flow steps to get the switch mode to the 'Normal' state and associated Config Status.
- In all other cases, it may indicate a transient state where NDFC was not able to compute the correct 'Config Status'. Do the following:
    - If seen on one switch, then perform switch level **Preview** or **Deploy**.
    - If seen on multiple switches, then select those switches and perform **Preview** or **Deploy**.
    - If seen at a fabric level, then select all switches and perform **Preview** or **Deploy**.
    - R&D is also an option for fabric level but this does a **Config Save** operation as well which could take time in a large fabric.

To confirm that the diffs have been resolved, click **Recalculate & Deploy** after updating the policy to validate the changes.

NDFC checks only leading spaces, as it implies hierarchy of the command, especially in case of multi-command sequences. NDFC does not check any trailing spaces in command sequences.

Example 1: Configuration Compliance in Switch Freeform Policy

Let us consider an example with an incorrect spacing in the Switch Freeform Configuration field.

Create the switch freeform policy.

After deploying this policy successfully to the switch, NDFC persistently reports the diffs.

After clicking the **Side-by-side Comparison** tab, you can see the cause of the diff. The **ip pim rp-address** line has 2 leading spaces, while the running configuration has 0 leading spaces.

To resolve this diff, edit the corresponding Switch Freeform policy so that the spacing is correct.

After you save, you can use the **Push Config** or **Recalculate & Deploy** option to re-compute diffs.

The diffs are now resolved. The **Side-by-side Comparison** tab confirms that the leading spaces are updated.

Example 2: Resolving a Leading Space Error in Overlay Configurations

Let us consider an example with a leading space error that is displayed in the **Pending Config** tab.

In the **Side-by-side Comparison** tab, search for diffs line by line to understand context of the deployed configuration.

A matched count of 0 means that it is a special configuration that NDFC has evaluated to push it to the switch.

You can see that the leading spaces are mismatched between running and expected configurations.

Navigate to the respective freeform configs and correct the leading spaces, and save the updated configuration.

Navigate to **Fabric Overview** window for the fabric and click **Recalculate & Deploy**.

In the **Deply Configuration** window, you can see that all the devices are in-sync.

Configuration Compliance in External Fabrics

With external fabrics, any Nexus switches, Cisco IOS-XE devices, Cisco IOS XR devices, and Arista can be imported into the fabric, and there is no restriction on the type of deployment. It can be LAN Classic, VXLAN, FabricPath, vPC, HSRP, etc. When switches are imported into an external fabric, the configuration on the switches is retained so that it is non-disruptive. Only basic policies such as the switch username and mgmt0 interface are created after a switch import.

In the external fabric, for any intent that is defined in the Nexus Dashboard Fabric Controller, configuration compliance (CC) ensures that this intent is present on the corresponding switch. If this intent is not present on the switch, CC reports an Out-of-Sync status. Additionally, there will be a Pending Config generated to push this intent to the switch to change the status to In-Sync. Any additional configuration that is on the switch but not in intent defined in Nexus Dashboard Fabric Controller, will be ignored by CC, as long as there is no conflict with anything in the intent.

When there is user-defined intent added on Nexus Dashboard Fabric Controller and the switch has additional configuration under the same top-level command, as mentioned earlier, CC will only ensure that the intent defined in Nexus Dashboard Fabric Controller is present on the switch. When this user defined intent on Nexus Dashboard Fabric Controller is deleted as a whole with the intention of removing it from the switch and the corresponding configuration exists on the switch, CC will report an Out-of-Sync status for the switch and will generate **Pending Config** to remove the config from the switch. This **Pending Config** includes the removal of the top-level command. This action leads to removal of the other out-of-band configurations made on the switch under this top-level command as well. If you choose to override this behavior, the recommendation is that, you create a freeform policy and add the relevant top-level command to the freeform policy.

Let us see this behavior with an example.

1. A **switch_freeform** policy defined by the user in Nexus Dashboard Fabric Controller and deployed to the switch.
2. Additional configuration exists under **router bgp** in **Running config** that does not exist in user-defined Nexus Dashboard Fabric Controller intent **Expected config**. Note that there is no **Pending Config** to remove the additional config that exists on the switch without a user defined intent on Nexus Dashboard Fabric Controller.
3. The **Pending Config** and the **Side-by-side Comparison** when the intent that was pushed earlier via Nexus Dashboard Fabric Controller is deleted from Nexus Dashboard Fabric Controller by deleting the switch_freeform policy that was created in the Step 1.
4. A **switch_freeform** policy with the top-level **router bgp** command needs to be created. This enables CC to generate the configuration needed to remove only the desired sub-config which was pushed from Nexus Dashboard Fabric Controller earlier.
5. The removed configuration is only the subset of the configuration that was pushed earlier from Nexus Dashboard Fabric Controller.

   For interfaces on the switch in the external fabric, Nexus Dashboard Fabric Controller either manages the entire interface or does not manage it at all. CC checks interfaces in the following ways:

   - For any interface, if there is a policy defined and associated with it, then this interface is considered as managed. All configurations associated with this interface must be defined in the associated interface policy. This is applicable for both logical and physical interfaces.

Otherwise, CC removes any out-of-band updates made to the interface to change the status to **In-Sync**.

- Interfaces created out-of-band (applies for logical interfaces such as port-channels, sub interfaces, SVIs, loopbacks, etc.), will be discovered by Nexus Dashboard Fabric Controller as part of the regular discovery process. However, since there is no intent for these interfaces, CC will not report an **Out-of-Sync** status for these interfaces.
- For any interface, there can always be a monitor policy associated with it in Nexus Dashboard Fabric Controller. In this case, CC will ignore the interface's configuration when it reports the **In-Sync** or **Out-of-Sync** config compliance status.

Special Configuration CLIs Ignored for Configuration Compliance

The following configuration CLIs are ignored during configuration compliance checks:

- Any CLI having 'username' along with 'password'
- Any CLI that starts with 'snmp-server user'

Any CLIs that match the above will not show up in pending diffs and clicking Save & Deploy in the Fabric Builder window will not push such configurations to the switch. These CLIs will not show up in the Side-by-side Comparison window also.

To deploy such configuration CLIs, perform the following procedure:

1. Select **Manage > Fabrics**.

   Double click on the fabric name to view **Fabric Overview** screen.

2. On the Switches tab, double click on the switch name to view **Switch Overview** screen.

   On the Policies tab, all the policies applied on the switch within the chosen fabric are listed.

3. On the Policies tab, from the **Actions** drop-down list, select **Add Policy**.
4. Add a Policy Template Instances (PTIs) with the required configuration CLIs using the **switch_freeform** template and click **Save**.
5. Select the created policy and select **Push Config** from the **Actions** drop-down list to deploy the configuration to the switch(es).

Resolving Diffs for Case Insensitive Commands

By default, all diffs generated in NDFC while comparing intent, also known as the expected configuration and the running configuration, are case sensitive. However, the switch has many commands that are case insensitive, and therefore it may not be appropriate to flag these commands as differences. These are captured in the **compliance_case_insensitive_clis.txt** template that can be found under **Manage > Templates**.

From Cisco NDFC Release 12.0.1a, the **compliance_case_insensitive_clis.txt** file, along with the **compliance_strict_cc_exclude_clis.txt** and **compliance_ipv6_clis.txt** files are now part of the shipped templates. You can find all the templates in **Manage > Templates**. Modification of templates can be done after disabling **Template In-Use Override**.

There could be additional commands that are not included in the existing **compliance_case_insensitive_clis.txt** file that should be treated as case insensitive. If the pending configuration is due to the differences of cases between the expected configuration in NDFC and the running configuration, you can configure NDFC to ignore these case differences as follows:

1. Navigate to **Admin > System Settings > Server Settings > LAN-Fabric**, uncheck **Template In-Use Override**, and then click **Save**.
2. Navigate to **Manage > Templates** and search for the **compliance_case_insensitive_clis.txt** file.
3. Check **compliance_case_insensitive_clis.txt** and choose the **Actions > Edit** template content.

   An example of the entries in the **compliance_case_insensitive_clis.txt** file is displayed in the following figure.

4. Remove the entries highlighted in the figure and click **Finish**.

```
[root@dcnm98 model-config]# pwd
/usr/local/cisco/dcm/dcnm/model-config
[root@dcnm98 model-config]# cat compliance_case_insensitive_clis.txt
"^(no |)interface\s+Port(.)"
"^(no |)interface\s+Loo(.)"
"^(no |)interface\s+Eth(.)"
"^update-source\s+Loo(.)"
"^vrf\s+"
"^hardware profile portmode\s+"
"^(.*)route-map\s+(.)"
"^(.*)neighbor-policy(.)"
"(no |)encapsulation\s+(.)"
"(.*)alert-group\s+(.)"
"^streetaddress\s+(.)"
"^transport email\s+(.)"
"(no |)action\s+(.)"
"(no|)\s+\d*\s+remark.*"
[root@dcnm98 model-config]#
```

After upgrading to NDFC 12.2.1, you need to uncomment the following two lines in the **compliance_case_insensitive_clis** template:

```
==
```

```
" ^match\s+external-subnets\s+vrf\s+.*"
" ^match\s+connected-endpoints\s+vrf\s+.*"
==
```

The NDFC upgrade process adds these two line as comments in the **compliance_case_insensitive_clis** template. You need to uncomment these lines. If you do not uncomment these two lines, the security group access control lists (SGACL) always displays as **out-of-sync** due to a mismatch between the VRF name in NDFC and the Cisco NX-OS.

5. If newer patterns are detected during deployment, and they are triggering pending configurations, you can add these patterns to this file. The patterns need to be valid regex patterns.
6. Navigate to **Admin > System Settings > Server Settings > LAN-Fabric**, check **Template In-Use Override**, and then click **Save**.

   This enables NDFC to treat the documented configuration patterns as case insensitive while performing comparisons.

7. Click **Recalculate & Deploy** for fabrics to see the updated comparison outputs.

Resolving Configuration Compliance After Importing Switches

After importing switches in Cisco NDFC, configuration compliance for a switch can fail because of an extra space in the management interface (mgmt0) description field.

For example, before importing the switch:

```
interface mgmt0
  description SRC=SDS-LB-LF111-mgmt0, DST=SDS-LB-SW001-Fa0/5
```

After importing the switch and creating a configuration profile:

```
interface mgmt0
  description SRC=SDS-LB-LF111-mgmt0,DST=SDS-LB-SW001-Fa0/5
```

Navigate to Interface Manager and click the **Edit** icon after selecting the mgmt0 interface. Remove the extra space in the description.

Strict Configuration Compliance

Strict configuration compliance checks for diff between the switch configuration and the associated intent and generates no commands for the configurations that present on the switch but are not present in the associated intent. When you click **Recalculate and Deploy**, switch configurations that are not present on the associated intent are removed. You can enable this feature by choosing the **Enable Strict Config Compliance** check box under the **Advanced** tab in the **Create Fabric** or **Edit Fabric** window. By default, this feature is disabled.

The strict configuration compliance feature is supported on the Easy Fabric templates – **Data Center VXLAN EVPN** and **BGP Fabric**. To avoid generating diff for commands that are auto-generated by the switch, such as vdc, rmon, and so on, a file that has a list of default commands is used by CC to ensure that diffs are not generated for these commands. This file is maintained in **Manage > Templates**, **compliance_strict_cc_exclude_clis.txt** template.

Example: Strict Configuration Compliance

Let us consider an example in which the feature telnet command is configured on a switch but is not present in the intent. In such a scenario, the status of the switch is displayed as **Out-of-sync** after a CC check is done.

Now, click **Preview Config** of the out-of-sync switch. As the strict configuration compliance feature is enabled, the no form of the feature telnet command appears under **Pending Config** in the **Preview Config** window.

Click the **Side-by-side Comparison** tab to display the differences between the running configuration and the expected configuration. The **Re-sync** button is also displayed at the top right corner under the Side-by-side Comparison tab in the **Preview Config** window. Use this option to resynchronize NDFC state when there is a large scale out-of-band change, or if configuration changes do not register in the NDFC properly.

The re-sync operation does a full CC run for the switch and recollects " show run" and " show run all" commands from the switch. When you initiate the re-sync process, a progress message is displayed. During the re-sync, the running configuration is taken from the switch. The Out-of- Sync/In-Sync status for the switch is recalculated based on the intent defined in NDFC.

Now, close the **Preview Config** window and click **Recalculate and Deploy**. The strict configuration compliance feature ensures that the running configuration on the switch does not deviate from the intent by pushing the no form of the feature telnet command to the switch. The diff between the configurations is highlighted. The diff other than the feature telnet command are default switch and boot configurations and are ignored by the strict CC check.

You can right-click on a switch in the **Fabric Overview** window and select **Preview Config** to display the **Preview Config** window. This window displays the pending configuration that has to be pushed to the switch to achieve configuration compliance with the intent.

Custom freeform configurations can be added in NDFC to make the intended configuration on NDFC and the switch configurations identical. The switches are then in In-Sync status. For more information on how to add custom freeform configurations on NDFC, refer to Enabling Freeform Configurations on Fabric Switches.

Copyright