

Templates Overview and Operations

- Schema and Template Design Considerations, on page 1
- Concurrent Configuration Updates, on page 5
- Assigning Templates to Sites, on page 7
- Disassociating Template from Sites, on page 7
- Deploying Templates, on page 8
- Undeploying Templates, on page 12
- Bulk Update for Template Objects, on page 13
- Template Versioning, on page 16
- Template Review and Approval, on page 20
- Configuration Drifts, on page 23
- Cloning Templates, on page 27
- Migrating Objects Between Templates, on page 28
- Viewing Currently Deployed Configuration, on page 29
- Schema Overview and Deployment Visualizer, on page 30

Schema and Template Design Considerations

Nexus Dashboard Orchestrator provides a number of policy templates that allow you to define one or more policies together and deploy them to one or more sites at the same time. These include Application templates, Tenant Policies templates, Fabric Policies and Fabric Resources Policies templates, and Monitoring templates. A schema is a collection of Application templates, which are used for defining application policies, with each template assigned to a specific tenant; schemas apply to Application templates only. There are multiple approaches you can take when it comes to creating the templates configurations specific to your deployment use case. The following sections describe a few simple design directions you can take when deciding how to define the schemas, templates, and policies in your Multi-Site domain.

Keep in mind that when designing schemas, you must consider the supported scalability limits for the number of schemas, templates, and objects per schema. Detailed information on verified scalability limits is available in the *Nexus Dashboard Orchestrator Verified Scalability Guides* for your release.

Application Templates

There are 3 types of schema templates, also known as application templates, available in Nexus Dashboard Orchestrator, each designed for a specific purpose:

- **ACI Multi-Cloud**—Templates used for Cisco ACI on-premises and cloud sites. This template supports two deployment types:
  - `Multi-Site` - The template can be associated to a single site (site-local policies) or to multiple sites (stretched policies) and the option should be selected for Multi-Site Network (ISN) or VXLAN intersite communication to allow template and object stretching between multiple sites.
  - `Autonomous` - The template can be associated to one or more sites that are operated independently and are not connected through an Inter-Site Network (no intersite VXLAN communication).

    Because autonomous sites are by definition isolated and do not have any intersite connectivity, there is no shadow object configuration across sites and no cross-programming of pctags or VNIDs in the spine switches for intersite traffic flow.

    The autonomous templates also allow for significantly higher deployment scale.

  The following sections focus primarily on this type of templates.
- **NDFC**—Templates designed for Cisco Nexus Dashboard Fabric Controller (formerly Data Center Network Manager) sites.

  This guide describes Nexus Dashboard Orchestrator configurations for on-premises Cisco ACI fabrics. For information on working with Cisco NDFC sites, see the *Cisco Nexus Dashboard Orchestrator Configuration Guide for NDFC Fabrics* instead.
- **Cloud Local**—Templates designed for specific Cloud Network Controller use cases, such as Google Cloud site connectivity, and cannot be stretched between multiple sites.

  This guide describes Nexus Dashboard Orchestrator configurations for on-premises Cisco ACI fabrics. For information on working with Cloud Network Controller fabrics, see the Nexus Dashboard Orchestrator use case library instead.

When creating schemas and application templates, you can choose to adopt one of the following simple approaches:

- **Single Template Deployment**

  The simplest schema design approach is a single schema, single template deployment. You can create a single schema with a single template within it and add all VRFs, Bridge Domains, EPGs, Contracts and other elements to that template and deploy it to one or more sites.

  This simplest approach to Multi-Site schema creation is to create all objects within the same schema and template. However, the supported number of schemas scalability limit may make this approach unsuitable for large scale deployments, which could exceed those limits.

  Note also that with this approach all the objects defined in the template become "stretched objects" and all changes made to the template are always simultaneously deployed to all the sites associated to such template.
- **Multiple Templates with Network Separation**

  Another approach to schema design is to separate the networking objects from the application policy configuration. Networking objects include VRFs, Bridge Domains, and subnets, while the application policy objects include EPGs, Contracts, Filters, External EPGs, and Service Graphs.

  You begin by defining a schema that contains the network elements. You can choose to create a single schema that contains all the network elements or you can split them into multiple schemas based on which applications reference them or which sites the network is stretched to.

You can then define one or more separate schemas which contain each application's policy objects. This new schema can reference the network elements, such as bridge domains, defined in the previous schema.

After creating and deploying the policy schemas and templates, the networking objects in the networking schema will display the number of external references by the policy schema elements. The object with external references will also be denoted by the ribbon icon.

Schemas designed this way provide logical separation of networking objects from the policy objects. However, this creates additional complexity when it comes to keeping track of externally referenced objects in each schema.

- **Multiple Templates Based On Object Relationships**

  When configuring multiple schemas with shared object references, it is important to be careful when making changes to those objects. For instance, making changes to or deleting a shared networking object can impact applications in one or more sites. Because of that, you may choose to create a template around each individual site that contains only the objects used by that site and its applications, including the VRFs, BDs, EPGs, Contracts, and Filters. And create different templates containing the shared objects.

  For example, you can create a `Site1` template that contains only the objects that are local to Site1 and the template is deployed to only that site. Similarly, the `Site2` template contains only the object relevant to site2 and is deployed to that site only. Any change made to any object in either of these templates has no effect on the other one. Then you can create a **`Shared`** template which contains objects that are shared between the sites.

  You can extend this scenario for an additional site with the following template layout:

  - Site 1 template
  - Site 2 template
  - Site 3 template
  - Site 1 and 2 shared template
  - Site 1 and 3 shared template
  - Site 2 and 3 shared template
  - All shared template

  Similarly, rather than separating objects based on which site they are deployed to, you can also choose to create schemas and templates based on individual applications instead. This would allow you to easily identify each application profile and map them to schemas and sites as well as easily configure each application as local or stretched across sites.

  However, as this could quickly exceed the templates per schema limit (listed in the Verified Scalability Guide for your release), you would have to create additional schemas to accommodate the multiple combinations. While this creates additional complexity with multiple additional schemas and templates, it provides true separation of objects based on site or application.

Fabric Policy Templates

In addition to the three types of application templates, Release 4.0(1) adds 3 new templates designed for fabric-wide policies:

- **Fabric Policies** templates can be used for managing the following fabric-wide policies:

- VLAN Pool
- Physical Domains
- SyncE Interface Policies
- Interface Settings
- Node Settings
- Pod Settings
- MACsec
- NTP Policies
- PTP Policies
- QoS DSCP Policies
- QoS SR-MPLS Policies
- QoS Class Policies

For additional information, see Creating Fabric Policies.

- **Fabric Resources Policies** templates can be used for managing the following fabric-wide policies:
  - Physical Interfaces
  - Port Channel Interfaces
  - Virtual Port Channel Interfaces
  - Node Profiles

  These templates reference policies are defined in the Fabric Policies templates, so those templates must be created and deployed first. For additional information, see Creating Fabric Resources Policies.

- **Monitoring Policy** templates can be used for managing `Tenant SPAN` or `Access SPAN` policies.

  For additional information, see Creating Monitoring Policies.

Template Design Best Practices

Beginning with Release 4.0(1), Nexus Dashboard Orchestrator validates and enforces a number of best practices when it comes to template design and deployment. Regardless of the type of template you are creating, keep in mind the following:

- All policy objects must be **deployed** in order according to their dependencies.

  For example, when creating a bridge domain (BD), you must associate it with a VRF. In this case, the BD has a VRF dependency so the VRF must be deployed to the fabric before or together with the BD. If these two objects are defined in the same template, then the Orchestrator will ensure that during deployment, the VRF is created first and associate it with the bridge domain.

  However, if you define these two objects in separate templates and attempt to deploy the template with the BD first, the Orchestrator will return a validation error as the associated VRF is not yet deployed. In this case, you must deploy the VRF template first, followed by the BD template.

- All policy objects must be **undeployed** in order according to their dependencies, or in the opposite order in which they were deployed.

  As a corollary to the point above, when you undeploy templates, you must not undeploy objects on which other objects depend. For example, you cannot undeploy a VRF before undeploying the BD with which the VRF is associated.

- No cyclical dependencies are allowed across multiple templates.

  Consider a case of a VRF (`vrf1`) associated with a bridge domain (`bd1`), which is in turn associated with an EPG (`epg1`). If you create `vrf1` in `template1` and deploy that template, then create `bd1` in `template2` and deploy that template, there will be no validation errors since the objects are deployed in correct order. However, if you then attempt to create `epg1` in `template1`, it would create a circular dependency between the two template, so the Orchestrator will not allow you to save `template1` addition of the EPG.

Concurrent Configuration Updates

The Nexus Dashboard Orchestrator GUI will ensure that any concurrent updates on the same site or schema object cannot unintentionally overwrite each other. If you attempt to make changes to a site or template that was updated by another user since you opened it, the GUI will reject any subsequent changes you try to make and present a warning requesting you to refresh the object before making additional changes; refreshing the template will lose any edits you made up to that point and you will have to make those changes again:



However, the default REST API functionality was left unchanged in order to preserve backward compatibility with existing applications. In other words, while the UI is always enabled for this protection, you must explicitly enable it for your API calls for NDO to keep track of configuration changes.

Note When enabling this feature, note the following:

- This release supports detection of conflicting configuration changes for Site and Schema objects only.
- Only `PUT` and `PATCH` API calls support the version check feature.
- If you do not explicitly enable the version check parameter in your API calls, NDO will not track any updates internally. And as a result, any configuration updates can be potentially overwritten by both subsequent API calls or GUI users.

To enable the configuration version check, you can pass the `enableVersionCheck=true` parameter to the API call by appending it to the end of the API endpoint you are using, for example:

`https://<mso-ip-address>/mso/api/v1/schemas/<schema-id>?`**`enableVersionCheck=true`**

Example

We will use a simple example of updating the display name of a template in a schema to show how to use the version check attribute with `PUT` or `PATCH` calls.

First, you would `GET` the schema you want to modify, which will return the current latest version of the schema in the call's response:

```
{
    "id": "601acfed38000070a4ee9ec0",
    "displayName": "Schema1",
    "description": "",
    "templates": [
        {
            "name": "Template1",
            "displayName": "current name",
            [...]
        }
    ],
    "_updateVersion": 12,
    "sites": [...]
}
```

Then you can modify the schema in one of two ways appending `enableVersionCheck=true` to the request URL:



Note You must ensure that the value of the `"_updateVersion"` field in the payload is the same as the value you got in the original schema.

- Using the `PUT` API with the entire updated schema as payload:

```
PUT /v1/schemas/601acfed38000070a4ee9ec0?enableVersionCheck=true
```

```
{
    "id": "601acfed38000070a4ee9ec0",
    "displayName": "Schema1",
    "description": "",
    "templates": [
        {
            "name": "Template1",
            "displayName": "new name",
            [...]
        }
    ],
    "_updateVersion": 12,
    "sites": [...]
}
```

- Using any of the `PATCH` API operations to make a specific change to one of the objects in the schema:

```
PATCH /v1/schemas/601acfed38000070a4ee9ec0?enableVersionCheck=true
```

```
[
    {
        "op": "replace",
        "path": "/templates/Template1/displayName",
        "value": "new name",
        "_updateVersion": 12
    }
]
```

When the request is made, the API will increment the current schema version by 1 (from `12` to `13`) and attempt to create the new version of the schema. If the new version does not yet exist, the operation will succeed and the schema will be updated; if another API call (with `enableVersionCheck` enabled) or the UI have modified the schema in the meantime, the operation fails and the API call will return the following response:

```
{
    "code": 400,
    "message": "Update failed, object version in the DB has changed, refresh your client
and retry"
}
```

Assigning Templates to Sites

This section describes how to assign a template to sites.

Before you begin

You must have the schema, template, and any objects you want to deploy to sites already created, as described in previous sections of this document.

Procedure

Step 1 Navigate to the schema that contains one or more templates that you want to deploy.

Step 2 In the left sidebar, select the template that you want to assign to sites.

Step 3 In the **Template Properties** view, click **Actions** and choose **Sites Association**.

The **Associate Sites to <template-name>** window opens.

Step 4 In the **Associate Sites** window, check the checkbox next to the sites where you want to deploy the template.

Note that some sites may not be available for assignment depending on the type of the template you selected and the intersite connectivity between sites:

- If you are assigning a `Cloud Local` template, you will be able to assign it only to a single cloud site.
- When assigning templates to multiple sites, the intersite connectivity between those sites must be established using BGP-EVPN protocol. If you select a site that has partial mesh connectivity, any site to which there is no intersite connectivity or intersite connectivity is established using BGP-IPv4 will be grayed out and unavailable for assignment.

Step 5 Click **Ok**.

You deploy one template at a time, so you must associate the template with at least one site before you can deploy it.

Disassociating Template from Sites

You can choose to disassociate a template from a site without undeploying it. This allows you to preserve any configuration deployed to the site from NDO while removing the template-site association in the schema. The managed object and policy ownership is transferred from NDO to the site's controller.

Before you begin

- The template and its configuration must already be deployed to a site.

- The template must be deployed to a single site only and not stretched across sites.
- The objects defined in the template must not be deployed as shadow objects in other sites.

Procedure

Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

Step 2 From the left navigation menu, select **Application Management** > **Schemas**.

Step 3 Click the schema that contains the template you want to disassociate.

Step 4 In the Schema view, select the template under the specific site from which you want to disassociate it.

Step 5 From the **Actions** menu, select **Disassociate Template**.

Step 6 In the confirmation window, click **Confirm Action**.

Deploying Templates

This section describes how to deploy new or updated policies to the ACI fabrics.

Before you begin

- You must have the schema, template, and any objects you want to deploy to sites already created and the templates assigned to one or more sites, as described in previous sections of this document.
- If template review and approval is enabled, the template must also be already approved by the required number of approvers as described in Template Review and Approval, on page 20.
- Ensure that you understand the required deployment order and object dependencies that are described in Schema and Template Design Considerations, on page 1.

Procedure

Step 1 Navigate to the schema that contains the template that you want to deploy.

Step 2 From the **View** dropdown menu, select the template you want to deploy.

Step 3 In the template view, click **Deploy to sites**.

The **Deploy to sites** window opens that shows the summary of the objects to be deployed.

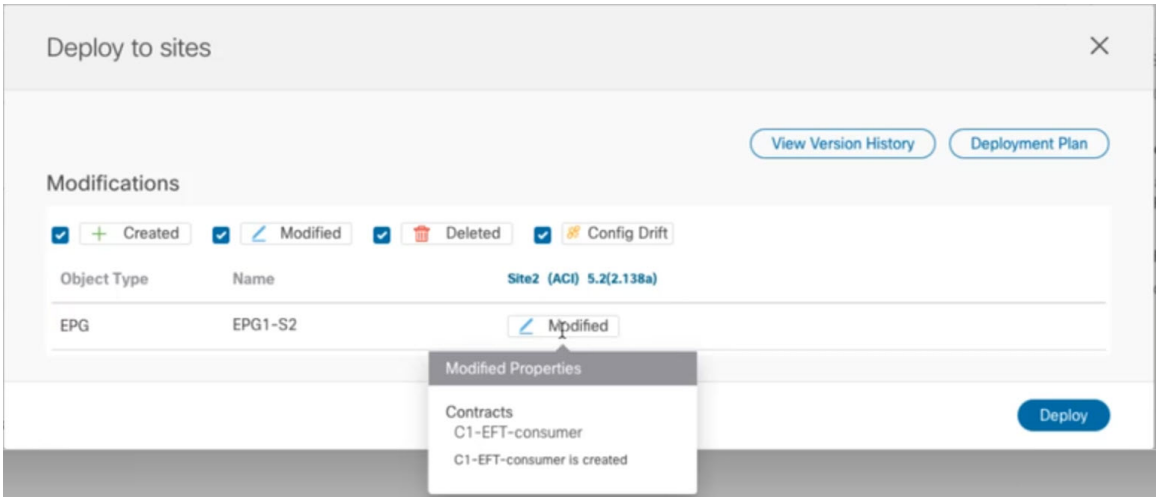
Step 4 If you have made changes to your template, review the **Deployment Plan** to verify the new configuration.

If you have previously deployed this template but made no changes to it since, the **Deploy** summary will indicate that there are no changes and you can choose to re-deploy the entire template. In this case, you can skip this step.

The **Deploy to sites** window will show you a summary of the configuration differences that will be deployed to sites. The following screenshots show a simple example of adding a `consumer` contract to an existing EPG (`EPG1-S2`) in `Site2`.

Note

In this case, only the difference in configuration is deployed to the sites. If you want to re-deploy the entire template, you must deploy once to sync the differences and then redeploy again to push the entire configuration as described in the previous paragraph.



You can also filter the view using the `Created`, `Modified`, and `Deleted` checkboxes for informational purposes, but keep in mind that all of the changes are still deployed when you click **Deploy**.

Here you can also choose to:

- **View Version History** shows the complete version history and incremental changes made between versions. Additional information about version history is available in Viewing History and Comparing Previous Versions, on page 17.
- Check the **Deployment Plan** to see a visualization and an XML payload of the configuration that will be deployed from this template.

  This feature provides better visibility into configuration changes that the Orchestrator will provision to the different fabrics that are part of your Multi-Site domain after you make a change to the template and deploy it to one or more sites.

  Unlike earlier releases of the Nexus Dashboard Orchestrator, which still provided a list of specific changes made to the template and site configuration, the Deployment Plan provides full visibility into all the objects that the deployment of the template would provision across the different fabrics. For example, depending on what change you make, shadow objects may be created in multiple sites even if the specific change is applied to only a single site.

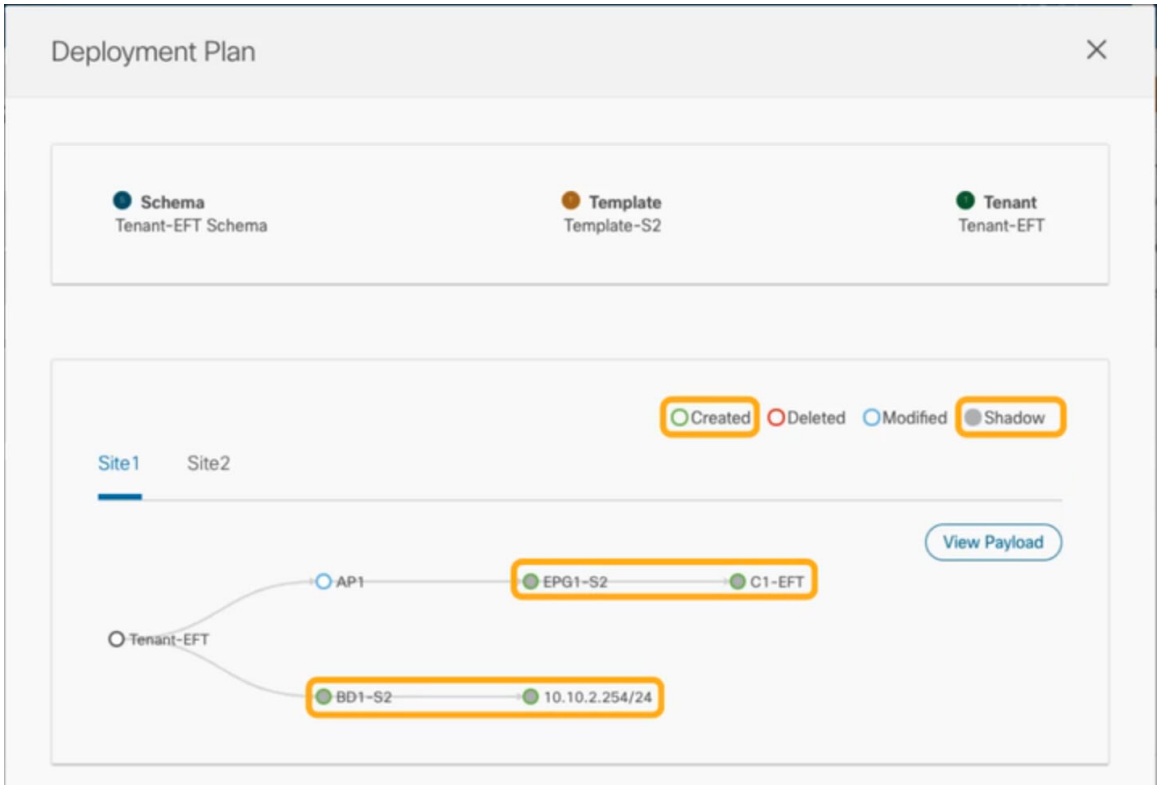
  Note
  We recommend verifying your changes using the Deployment Plan as described in this step before deploying the template. The visual representation of the configuration changes can help you reduce potential errors from deploying unintended configuration changes.

a) Click the **Deployment Plan** button.

   Continuing with the same example shown in the previous step, where a consumer contract was added to an existing EPG in `Site2`, the Deployment Plan allows you to also see that there are additional changes to be deployed to `Site1` as a result of the change to `Site2`.

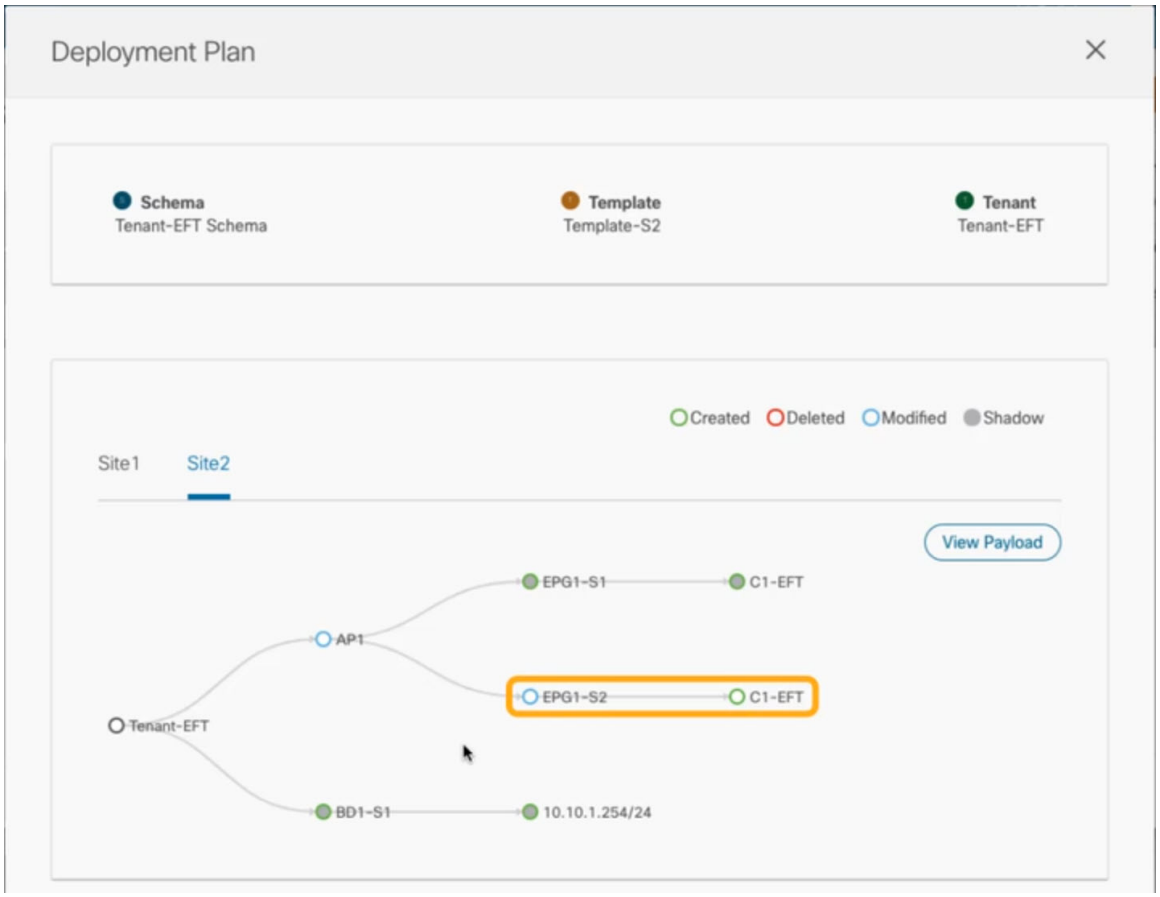
b) Verify your changes in the first listed site.

Based on the highlighted legend, you can see that the Orchestrator will create the shadow objects in `Site1` that are required by the contract you added to an EPG in `Site2`.



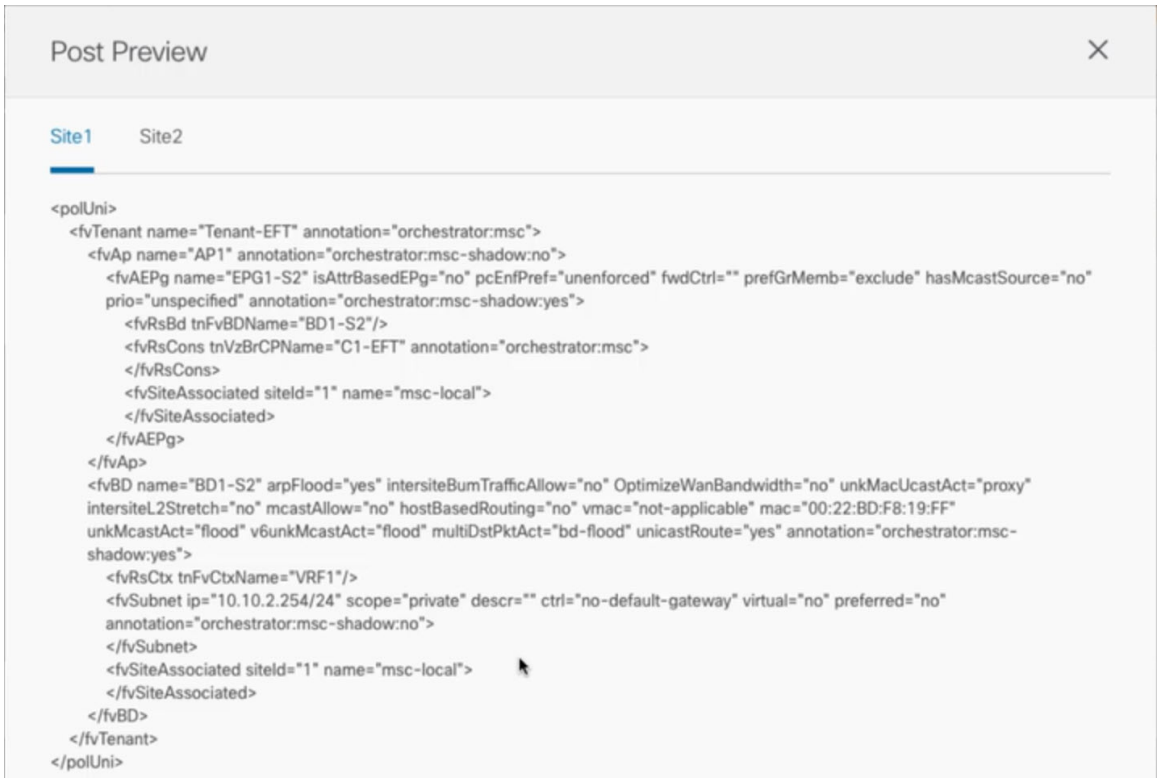
c) Repeat the previous substep to verify the changes in other sites

Here you can see the change you made explicitly to the EPG (`EPG1-S2`) in `Site2` when you assigned the contract (`C1-EFT`) to it, as well as the shadow objects for the EPG (`EPG1-S1`) in the other site, which is providing that contract.



d) (Optional) Click **View Payload** to see the XML payload for each site.

In addition to the visual representation of the new and modified objects, you can also choose to **View Payload** for the changes in each site:



e) After you are done verifying the changes, click the x icon to close the **Deployment Plan** screen.

Step 5 In the **Deploy to sites** window, click **Deploy** to deploy the template.

Undeploying Templates

This section describes how to undeploy a template from a site. Undeploying a template removes all configurations defined in that template from a specific site where the template is deployed.

Note This action removes managed objects (MOs) and their properties from the site's controller and can disrupt the network connectivity that depends on those configurations.

Before you begin

- Ensure that you have not made any changes to the template since you last deployed it.

  Undeploying a template that was modified since it was last deployed may create a configuration drift because the set of objects deployed with the template would be different than the set of objects you try to undeploy after making changes to the template.

- If you are undeploying a template that contains VRFs that are used in route leak configurations, the route leaks must be deleted before you can undeploy that template.

Procedure

Step 1 Select the schema that contains the template you want to undeploy.

Step 2 From the **View** dropdown, select the template you want to undeploy.

Step 3 From the **Actions** menu, click **Undeploy from all sites**.

Bulk Update for Template Objects

The bulk update feature allows you to update multiple properties on multiple different objects of the same type within a template at once. For example, you can enforce Infra EPG Isolation on two or more EPGs at the same time, instead of having to modify each object individually. When using this workflow, all selected objects must be of the same type, for example, you cannot choose to update an EPG and a BD simultaneously.

If the selected objects already have different property values configured on them, the update will overwrite those properties with the values you provide. This feature allows you update template-level object properties for on-premises; updating site-local properties and cloud properties are not supported.

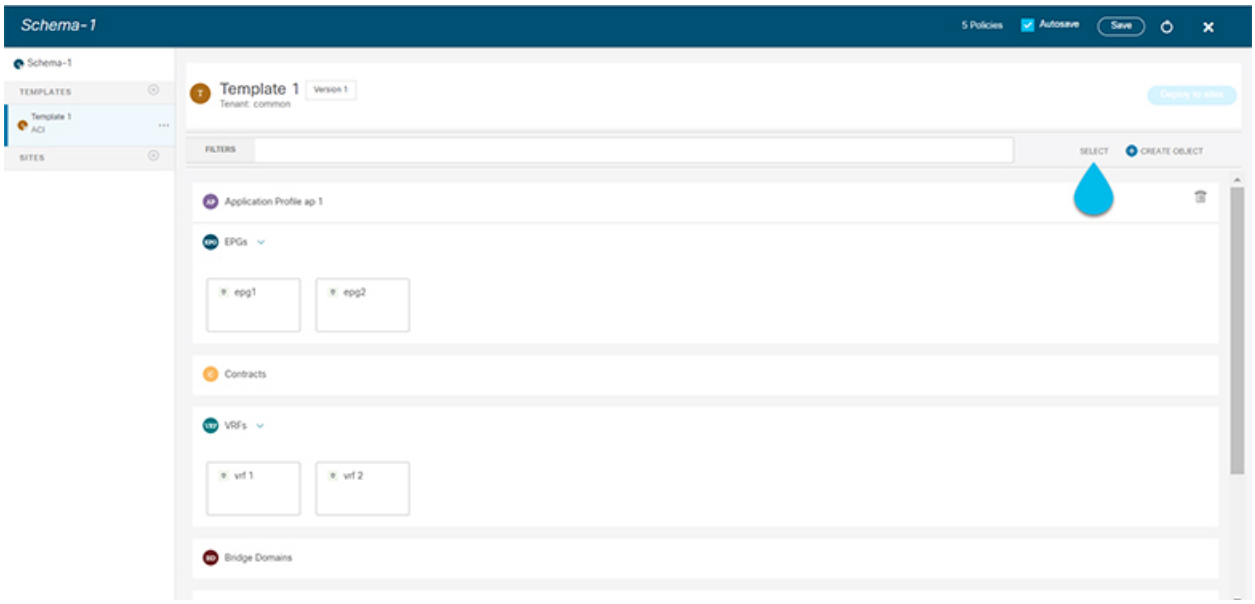


Note This feature is supported for Application templates only with Cisco APIC and Cisco NDFC fabrics only; it is not supported for other template types or Cisco Cloud Network Controller sites.

Procedure

Step 1 Navigate to the schema and template that contains the objects you want to update.

Step 2 From the main pane, choose **Select**. It will allow you to choose multiple objects of same type.

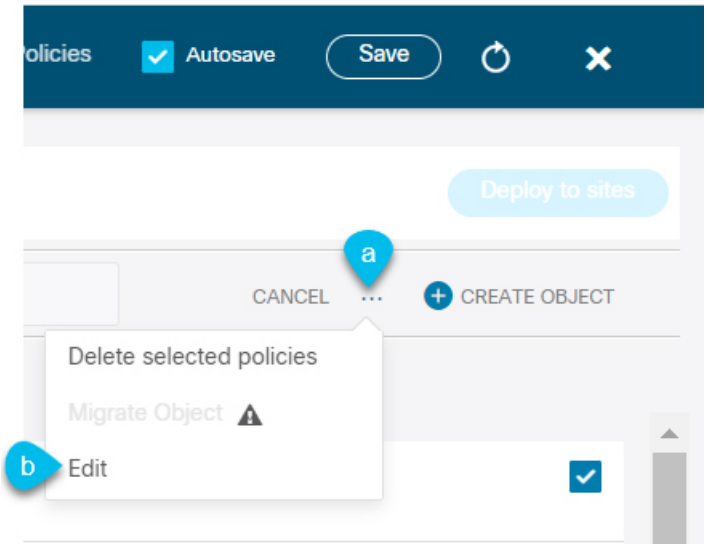


Step 3 After selecting all the objects that you want to update.

a) Choose "…" right next to the cancel option.

b) From the dropdown Choose "Edit".

If you choose objects of different type, you won't see the Edit option in the dropdown.



Step 4 After choosing "Edit", a pop-up will show up. It will show you a subset of the properties for the objects you selected.

You can update the following properties based on the type of objects you selected.

a. **EPG**: Bridge Domain, Contract, EPG Type, Infra EPG, Preferred Group.

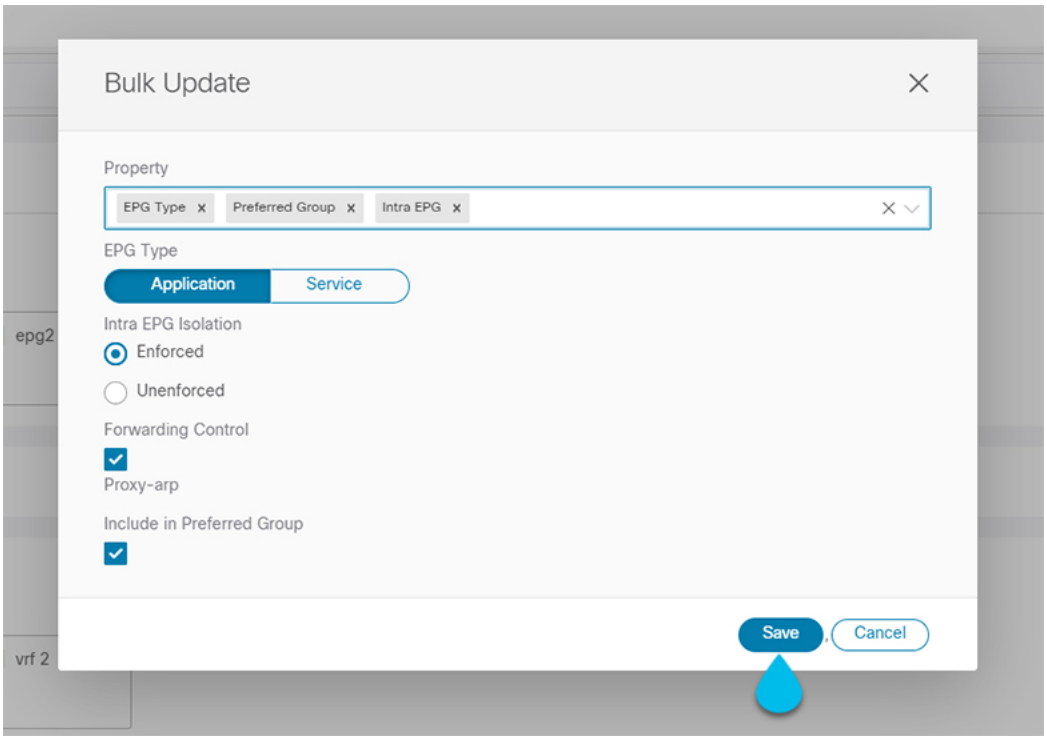
b. **Contracts**: Scope, Filter Chain, QOS Level.

c. **VRF**: IP Data-Plane Learning.

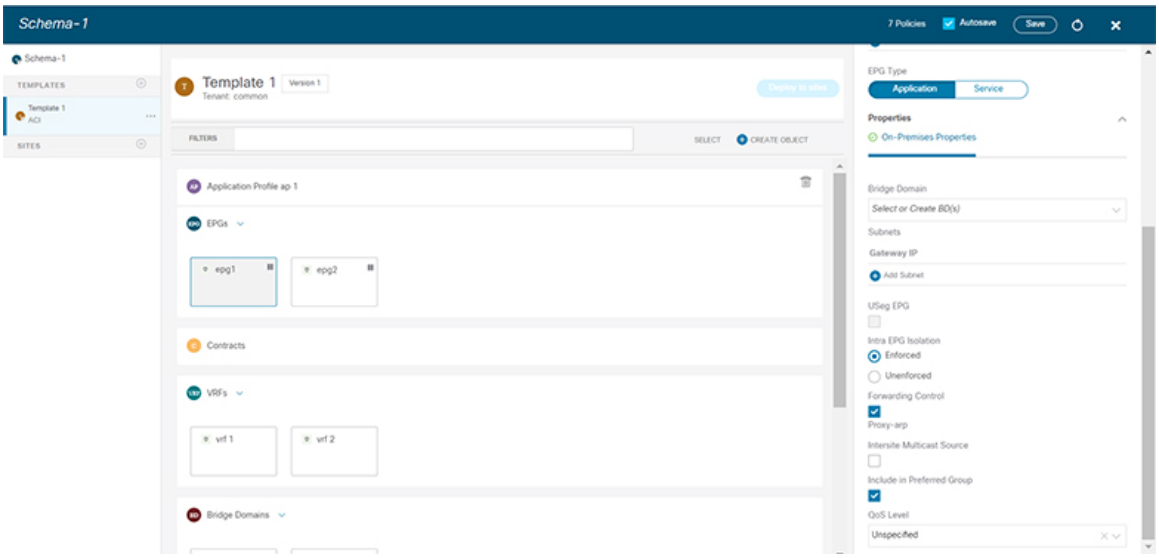
d. **Bridge Domain**: Virtual Routing and Fowrwarding, L2 Stretch, L2 Unknown Unicast, Unkown Multicast Flooding, IPv6 Unkown Multicast Flooding, Multi Destination Flooding, DHCP Policies, Unicast Routing.

e. **External EPG**: Contract, External EPG Type, Preferred Group.

Step 5 After selecting all the fields, you wish to update. Choose “Save” which will implement the update you just made.



Step 6 As you save the updates, you can see the changes you've made.



Template Versioning

A new version of the template is created every time it is saved. From within the NDO UI, you can view the history of all configuration changes for any template along with information about who made the changes and when. You can also compare any of the previous versions to the current version.

New versions are created at the template level, not schema level, which allows you to configure, compare, and roll back each template individually.

Template versions are created and maintained according to the following rules:

- All template versions are either `Deployed` or `Intermediate`.

  `Deployed`—versions of the template that have been deployed to sites.

  `Intermediate`—versions of the template that have been modified and saved, but not deployed to sites.

- A maximum of 20 `Deployed` and 20 `Intermediate` versions per template can be stored at any given time.
- When a new `Intermediate` version is created that would exceed the 20 version limit, the earliest existing `Intermediate` version is deleted.
- When a template is deployed and a new `Deployed` version is created, all `Intermediate` versions are deleted. If the new `Deployed` version exceeds the 20 version limit, the earliest existing `Deployed` version is deleted.
- Tagging a version `Golden` does not affect the number of stored template versions.
- A template that is tagged `Golden` cannot be deleted.

  You must untag the template first before you can delete it.

- When a template is modified and saved or deployed, any versions that exceed the 20 `Deployed` and 20 `Intermediate` scale are removed according to the above rules.
- When upgrading from a release prior to 4.0(1) to release 4.0(1) or later, only the latest versions of templates are preserved.

Tagging Templates

At any point you can choose to tag the current version of the template as "golden", for example for future references to indicate a version that was reviewed, approved, and deployed with a fully validated configuration.

Procedure

Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

Step 2 From the left navigation menu, select **Application Management** > **Schemas**.

Step 3 Click the schema that contains the template you want to view.

Step 4 In the Schema view, select the template you want to review.

Step 5 From the template's actions (**...**) menu, select **Tag**.

If the template is already tagged, the option will change to **Un-Tag** and allows you to remove the tag from the current version.

Any version that was tagged will be indicated by a star icon in the template's version history screen.

Viewing History and Comparing Previous Versions

This section describes how to view previous versions for a template and compare them to the current version.

Procedure

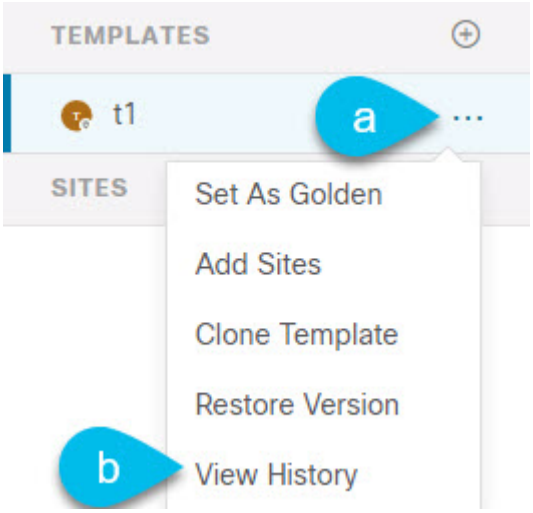
Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

Step 2 From the left navigation menu, select **Application Management** > **Schemas**.

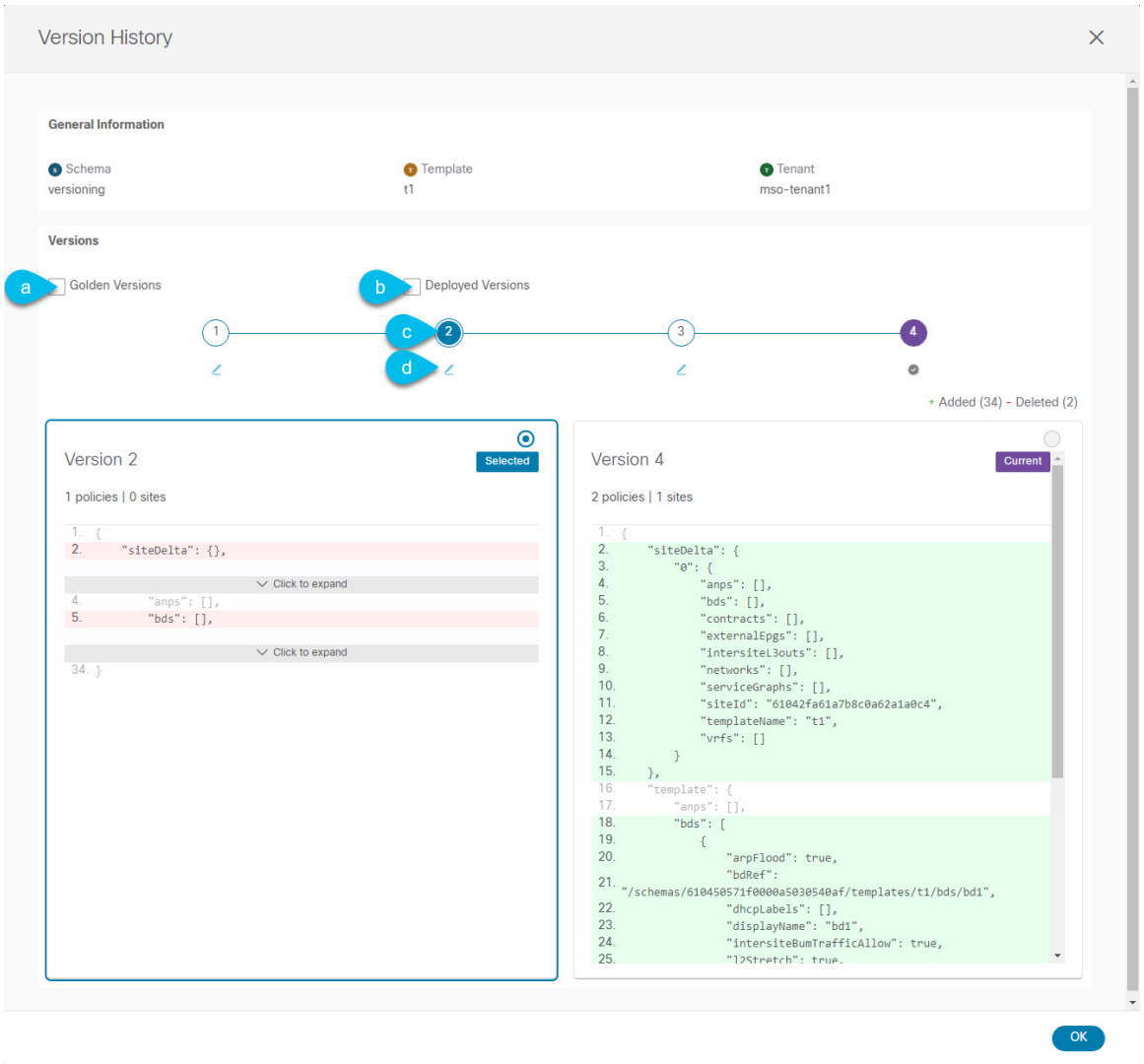
Step 3 Click the schema that contains the template you want to view.

Step 4 In the Schema view, select the template you want to review.

Step 5 From the template's actions (**...**) menu, select **History**.



Step 6 In the **Version History** window, make the appropriate selections.



a) Enable the **Golden Versions** checkbox to filter the list of previous versions to display only the versions of this template that had been marked as `Golden`.

   Tagging a template as "Golden" is described in Tagging Templates, on page 16.

b) Enable the **Deployed Versions** checkbox to filter the list of previous versions to display only the versions of this template that had been deployed to sites.

   A new template version is created every time the template is changed and the schema is saved. You can choose to only show the versions of the template that were actually deployed to sites at some point.

c) Click on a specific version to compare it to the current version.

   The version you select is always compared to the current version of the template. Even if you filter the list using the **Golden Versions** or **Deployed Versions** filters, the current version will always be displayed even if it was never deployed or tagged as golden.

d) Mouse over the **Edit** icon to see information about who created the version and when.

Step 7 Click **OK** to close the version history window.

Reverting Template to Earlier Version

This section describes how to restore a previous version of the template. When reverting a template, the following rules apply:

- If the target version references objects that are no longer present, restore operation will not be allowed.
- If the target version references sites that are no longer managed by NDO, restore operation will not be allowed.
- If the current version is deployed to one or more sites to which the target version was not deployed, restore operation will not be allowed.

  You must first undeploy the current version from those sites before reverting the template.
- If the target version was deployed to one or more sites to which the current version is not deployed, restore operation is allowed.

Procedure

Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

Step 2 From the left navigation menu, select **Application Management** > **Schemas**.

Step 3 Click the schema that contains the template you want to view.

Step 4 In the Schema view, select the template you want to review.

Step 5 From the **Actions** (**...**) menu, select **Rollback**.

Step 6 In the **Rollback** window, select one of the earlier versions to which you want to restore.

You can filter the list of versions using the **Golden Versions** and **Deployed Versions** checkboxes.

When you select a version, you can compare the template configuration of that version to the current version of the template.

Step 7 Click **Restore** to restore the selected version.

When you restore a previous version, a new version of the template is created with the same configuration as the version you selected in the previous step.

For example, if the latest template version is `3` and you restore version `2`, then version `4` is created that is identical to the version `2` configuration. You can verify the restore by browsing to the template version history and comparing the current latest version to the version you had selected during restore, which should be identical.

If template review and approval (change control) is disabled and your account has the correct privileges to deploy templates, you can deploy the version to which you reverted.

However, if change control is enabled, then:

- If you revert to a version that had been previously deployed and your account has the correct privileges to deploy templates, you can immediately deploy the template.

- If you revert to a version that had not been previously deployed or your account does not have the correct privileges to deploy templates, you will need to request template approval before the reverted version can be deployed.

  Additional information about review and approval process is available in the Template Review and Approval, on page 20 sections.

Template Review and Approval

Template review and approval (change control) workflow which allows you to set up designated roles for template designers, reviewers and approvers, and template deployers to ensure that the configuration deployments go through a validation process.

From within the NDO UI, a template designer can request review on the template they create. Then reviewers can view the history of all configuration changes for the template along with information about who made the changes and when, at which point they can approve or deny the current version of the template. If the template configuration is denied, the template designer can make any required changes and re-request review; if the template is approved, it can be deployed to the sites by a user with `Deployer` role. Finally, the deployers themselves can deny deployment of an approved template and restart the review process from the beginning.

The workflow is done at the template level, not schema level, which allows you to configure, review, and approve each template individually.

Enabling Template Approval Requirement

Before you can use the review and approval workflow for template configuration and deployment, you must enable the feature in the Nexus Dashboard Orchestrator's system settings.

Procedure

Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

Step 2 From the left navigation menu, select **Infrastructure** > **System Configuration**.

Step 3 On the **Change Control** tile, click the **Edit** icon.

Step 4 In the **Change Control** window, select **Enabled** to enable the feature.

Step 5 In the **Approvers** field, enter the number of unique approvals required before the templates can be deployed.

Step 6 Click **Save** to save the changes.

Create Users with Required Roles

Before you can use the review and approval workflow for template configuration and deployment, you must create the users with the necessary privileges in the Nexus Dashboard where the NDO service is deployed.

Procedure

Step 1 Log in to your Nexus Dashboard GUI.

Users cannot be created or edited in the NDO GUI, you must log in directly to the Nexus Dashboard cluster where the service is deployed.

Step 2 From the left navigation menu, select **Admin Console** > **Administrative** > **Users**.

Step 3 Create the required users.

The workflow depends on three distinct user roles: template designer, approver, and deployer. You can assign each role to a different user or combine the roles for the same user; users with `admin` privileges can perform all 3 actions.

There is no `Designer` role predefined on Nexus Dashboard, so the designer duties are assigned to any `Tenant Manager` or `Site Manager` user with write privileges, in addition to the default `Admin` user role:

- `Tenant Manager` should be used when the designer needs to make changes to templates associated only to a specific tenant (or a subset of tenants). In this case, the user should be mapped to the specific tenants.
- `Site Manager` should be used when the designer needs to make changes to templates that belong to different tenants.

In contrast to `Designer` role, there are pre-defined `Approver` and `Deployer` roles on the Nexus Dashboard that can be associated to the users. `Approver` and `Deployer` roles are not bound to specific tenant(s) by design. However, when creating a user role with both designer and approver (or designer and deployer) rights, follow the same guidelines as listed above.

Detailed information about configuring users and their privileges for local or remote Nexus Dashboard users is described in the Nexus Dashboard User Guide.

You must have at least as many unique users with `Approver` role as the minimum number of approvals required, which you configured in Enabling Template Approval Requirement, on page 20.

Note
If you disable the **Change Control Workflow** feature, any `Approver` and `Deployer` users will have read-only access to the Nexus Dashboard Orchestrator.

Requesting Template Review and Approval

This section describes how to request template review and approval.

Before you begin

You must have:

- Enabled the global settings for approval requirement, as described in Enabling Template Approval Requirement, on page 20.
- Created or updated users in Nexus Dashboard with `approver` and `deployer` roles, as described in Create Users with Required Roles, on page 20.
- Created a template with one or more policy configurations and assigned it to one or more sites.

Procedure

Step 1 Log in to your Nexus Dashboard Orchestrator GUI as a user with `Tenant Manager`, `Site Manager`, or `Administrator` role.

Step 2 If you assigned the `Tenant Manager` role, associate the user with the tenants.

If you used `Site Manager` or `Administrator` roles, skip this step.

If you assign the `Tenant Manager` role, you must also associate the user to the specific tenants they will manage.

a) From the left navigation menu, select **Application Management** > **Tenants**.
b) Select the tenant which the user will manage.
c) Check the box next to the designer user you created in Nexus Dashboard.
d) Repeat this step for all other tenants the user will manage.

Step 3 From the left navigation menu, select **Application Management** > **Schemas**.

Step 4 Click the schema that contains the template for which you want to request approval.

Step 5 In the schema view, select the template.

Step 6 In the main pane, click **Send for Approval**.

Note that the **Send for Approval** button will not be available in the following cases:

- The global change control option is not enabled
- The template has no policy configurations or is not assigned to any sites
- Your user does not have the right permissions to edit templates
- The template has already been sent for approval
- The template was denied by the approver user

Reviewing and Approving Templates

This section describes how to request template review and approval.

Before you begin

You must have:

- Enabled the global settings for approval requirement, as described in Enabling Template Approval Requirement, on page 20.
- Created or updated users in Nexus Dashboard with `approver` and `deployer` roles, as described in Create Users with Required Roles, on page 20.
- Created a template with one or more policy configurations and assigned it to one or more sites.
- Had the template approval requested by a schema editor, as described in Requesting Template Review and Approval, on page 21.

Procedure

Step 1 Log in to your Nexus Dashboard Orchestrator GUI as a user with `Approver` or `admin` role.

Step 2 From the left navigation menu, select **Application Management** > **Schemas**.

Step 3 Click the schema that contains the template you want to review and approve.

Step 4 In the schema view, select the template.

Step 5 In the main pane, click **Approve**.

If you have already approved or denied the template, you will not see the option until the template designer makes changes and re-sends the template for review again.

Step 6 In the **Approving template** window, review the template and click **Approve**.

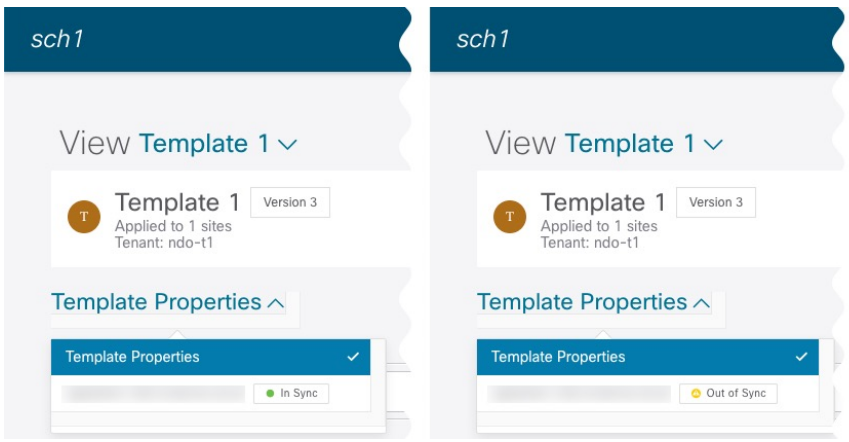
The approval screen will display all the changes which the template would deploy to the sites.

You can click **View Version History** to view the complete version history and incremental changes made between versions. Additional information about version history is available in Viewing History and Comparing Previous Versions, on page 17.

You can also click **Deployment Plan** to see a visualization and an XML of the configuration that would be deployed from this template. The functionality of the "Deployment Plan" view is similar to the "Deployed View" for already-deployed templates, which is described in Viewing Currently Deployed Configuration, on page 29.

Configuration Drifts

Occasionally, you may run into a situation where the configuration actually deployed to an APIC domain is different from the configuration defined for that domain in the Nexus Dashboard Orchestrator (NDO). These configuration discrepancies are referred to as **Configuration Drifts** and are indicated by an `Out of Sync` warning next to the site name in the template view page as shown in the following figure:



Note

- In certain cases, the template-level notification of a configuration drift shown above may not trigger if the configuration of properties of objects managed by NDO is modified directly in the site's controller. Specifically, addition (and subsequent removal) of the following properties do not show drift notification on NDO:
  - Subnets for EPGs or BDs
  - Bridge Domain DHCL Labels
  - Static Ports configuration for EPGs
  - Contract Relationships between EPGs

  In these cases, you can still check for configuration drift by manually running drift reconciliation workflow as described in Reconciling Configuration Drifts in Application Templates, on page 25.
- When you deploy a template from NDO, drift notification for objects in that template is disabled for 60 seconds.

Configuration Drift Causes

Configuration drifts can manifest due to a number of different reasons. Specific steps required to resolve a configuration drift depends on its cause. Most common scenarios and their resolutions are outlined below:

- **Configuration is modified in NDO**—when you modify a template in NDO GUI, it will show as configuration drift until you deploy the changes to the sites.

  To resolve this type of configuration drift, either deploy the template to apply the changes to the sites or revert the changes in the schema.
- **Configuration is modified directly in the site's APIC**—while the objects deployed from NDO are indicated by a warning icon and text in the site's APIC, an admin user can still make changes to them causing the configuration drift.

Note Every time an object is modified on APIC, APIC sends a notification to Nexus Dashboard Orchestrator. On receiving the notification, Nexus Dashboard Orchestrator starts a 30 second timer (waiting for further notifications to arrive) and at the expiration of such timer then makes API calls to APIC to retrieve detailed information about the changes made all the objects for which it received a notification. This allows the Nexus Dashboard Orchestrator to display the drift symbol on the UI for all the templates where those objects are defined. The only exception to this behavior is when Nexus Dashboard Orchestrator deploys the configuration for all (or of a subset of) the objects defined in a specific template. In that case, for 60 seconds Nexus Dashboard Orchestrator would ignore any notification received from APIC relative to those specific objects and, as a consequence, it would not be able to display the drift symbol on the UI.

- **NDO configuration is restored from backup**—restoring configuration from a backup in NDO restores only the objects and their state as they were when the backup was created, it does not automatically

re-deploy the restored configuration. As such, if there were changes made to the configuration and deployed on APIC since the backup was created, restoring the backup would create a configuration drift.

- **NDO configuration is restored from a backup created on an older release**—if the newer release added support for object properties which were not supported by the earlier release, these properties may cause configuration drift warning. Typically, this happens if the new properties were modified directly in the site's APIC GUI and the values are different from the defaults assumed by the Nexus Dashboard Orchestrator
- **NDO is upgraded from an earlier release**—this scenario is similar to the previous one where if new object properties are added in the new release, existing configuration may indicate a drift.

We recommend that you check for configuration drifts and, if necessary, run the "Reconcile Drift" workflow for templates, to have more visibility into the causes of the drift and be able to reconcile it. This recommendation applies to all the drift scenarios previously described in this section.

Reconciling Configuration Drifts in Application Templates

You can use the drift reconciliation workflow to compare the template's configuration as it is defined in Nexus Dashboard Orchestrator to the configuration rendered in the APIC controllers of the sites that are part of your Multi-Site domain. This provides better visibility into changes that may have been made in Nexus Dashboard Orchestrator or in APIC directly and give you an opportunity to correctly resolve those drifts.



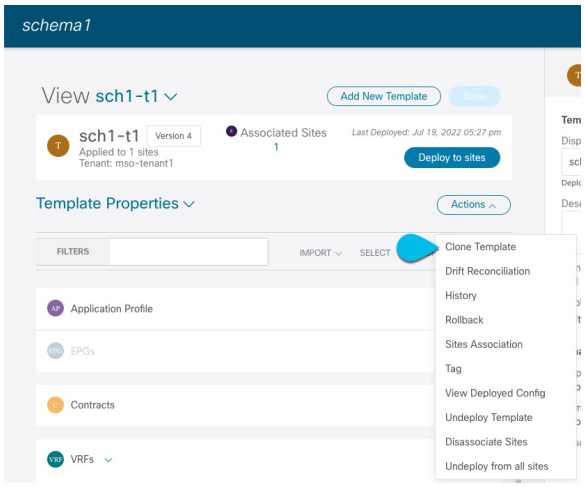
Note Configuration drift reconciliation is supported only for Application templates.

The templates are updated and saved only after you choose **Save** or **Deploy** at the end of the reconciliation workflow. If at any time during the workflow you want to undo the changes you already chose, you can close and re-open the schema to restore the original configurations. You can then re-run the workflow from the start.

Procedure

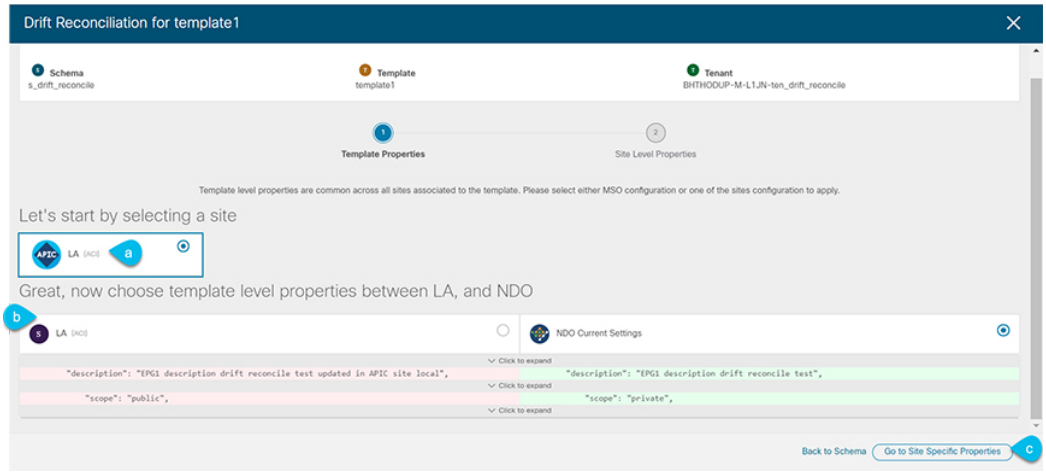
Step 1 Navigate to the schema that contains the template you want to check for configuration drifts.

Step 2 From the template's **Actions** menu, select **Reconcile Drift**.



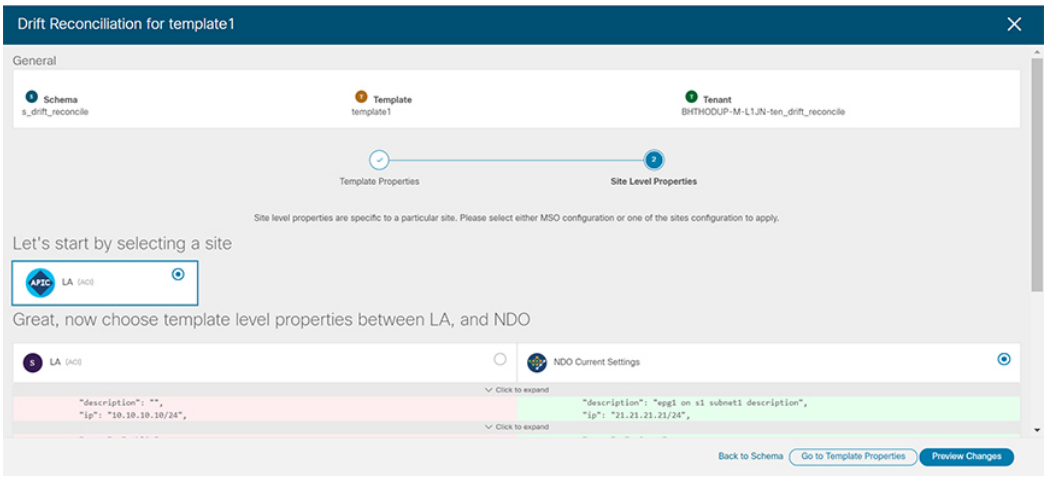
The **Drift Reconciliation** wizard opens.

Step 3 In the **Drift Reconciliation** screen, compare the template-level configurations for each site and choose the one you want.



Template-level properties are common across all sites associated to the template. You can compare the template level properties defined on Nexus Dashboard Orchestrator with the configuration rendered in each site and decide what should become the new configuration in the Nexus Dashboard Orchestrator template. Selecting the site configuration will modify those properties in the existing Nexus Dashboard Orchestrator template, whereas selecting the Nexus Dashboard Orchestrator configuration will keep the existing Nexus Dashboard Orchestrator template settings as is.

Step 4 Click **Go to Site Specific Properties** to switch to site-level configuration.



You can choose a site to compare that specific site's configuration. Unlike template-level configurations, you can choose either the Nexus Dashboard Orchestrator-defined or actual existing configurations for each site individually to be retained as the template's site-local properties for that site.

Even though in most scenarios you will make the same choice for both template-level and site-level configuration, the drift reconciliation wizard allows you to choose the configuration defined in the site's controller at the "Template Properties" level and the configuration defined in Nexus Dashboard Orchestrator at the "Site Local Properties" level or vice versa.

Step 5 Click **Preview Changes** to verify your choices.

The preview will display full template configuration adjusted based on the choices picked in the **Drift Reconciliation** wizard. You can then click **Deploy to sites** to deploy the configuration and reconcile the drift for that template.

Cloning Templates

This section describes how to create a copy of an existing template using the "Clone Template" feature in the Schema view.

Procedure

Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

Step 2 From the left navigation menu, select **Application Management** > **Schemas**.

Step 3 Click the schema that contains the template you want to clone.

Step 4 On the **View** menu, select a template to open it.

Step 5 From the **Actions** menu, select **Clone Template**.

Step 6 Provide the clone destination details.

a) From the **Destination Schema** dropdown, select the name of the Schema where you want to create the clone of the template.

You can select the same or a different schema to contain the clone of this template. If you want to clone the template into a schema that doesn't already exist, you can create a new schema by typing in the name of the schema and selecting `Create <schema-name>` option from the dropdown.

Note
When cloning across different schemas, the template must not have any objects that reference objects in other templates.

b) In the **Cloned Template Name** field, provide the name for the new template.
c) Click **Save** to create the clone.

A new template will be created in the destination schema, with the tenant you selected and the exact same object and policy configurations as the original template.

If the destination schema you chose was the same schema as the source template, the schema view will reload and the new template will be displayed in the left sidebar. If you chose a different schema, you can navigate to that schema to see and edit the new template.

Note that while the template objects and configurations are copied, the site association is not preserved and you will need to re-associate the cloned template with any sites where you want to deploy it. Similarly, you will need to provide any site-specific configurations for the template objects after you associate it with the sites.

Migrating Objects Between Templates

This section describes how to move objects between templates or schemas. When moving one or more objects, the following restrictions apply:

- Only EPG and Bridge Domain (BD) objects can be moved between templates.
- Migrating objects to or from Cloud Network Controller sites is not supported.

  You can migrate objects between on-premises sites only.
- The source and destination templates can be in the same schema or in different schemas, but the templates must be assigned to the same tenant.
- The destination template must have been created and assigned to at least one site.
- If the destination template is not deployed and has no other objects, the template will be automatically deployed after the objects are migrated.
- Once you initiate one object migration, you cannot perform another migration that involves the same source or target template. The migration is completed when the templates have been deployed to sites.

Procedure

Step 1 Log in to your Nexus Dashboard Orchestrator GUI.
Step 2 From the left navigation menu, select **Schemas**.
Step 3 Click the schema that contains the objects you want to migrate.
Step 4 In the Schema view, select the Template that contains the objects you want to migrate.

Step 5 In the top right of the main pane, click **Select**.

This allows you to select one or more objects to migrate.

Step 6 Click each object that you want to migrate.

Selected objects will display a check mark in their top right corner.

Step 7 In the top right of the main pane, click the actions (**...**) icon and choose **Migrate Objects**.

Step 8 In the **Migrate Objects** window, select the destination Schema and Template where you want to move the objects.

Only the templates with at least one site attached to them will appear in the list. If you don't see your target Template in the dropdown list, cancel the wizard and assign that template to at least one site.

Step 9 Click **OK** and then **YES** to confirm that you want to move the objects.

The objects will be migrated from the source template to the destination template that you selected. When you deploy your configuration, the objects will be removed from any site where the source Template is deployed and added to the site where the destination template is deployed.

Step 10 After the migration is completed, redeploy both, the source and the destination, templates.

If the destination template is not deployed and has no other objects, the template will be automatically deployed after the objects are migrated, so you can skip this step.

Viewing Currently Deployed Configuration

You can view all objects currently deployed to sites from a specific template. Even though any given template can be deployed, undeployed, updated, and re-deployed any number of times, this feature will show only the final state that resulted from all of those actions. For example, if `Template1` contains only `VRF1` object and is deployed to `Site1`, the API will return only `VRF1` for the template; if you then add `BD1` and redeploy, the API will return both objects, `BD1` and `VRF1`, from this point on.

This information comes from the Orchestrator database, so it does not account for any potential configuration drifts caused by changes done directly in the site's controller.

Procedure

Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

Step 2 From the left navigation menu, select **Application Management** > **Schemas**.

Step 3 Click the schema that contains the template you want to view.

Step 4 In the left sidebar, select the template.

Step 5 Open the **Deployed View** for the template.

a) Click the **Actions** menu next to the template's name.

b) Click **Deployed View**.

Step 6 In the **Deployed View** screen, select the site for which you want to view the information.

You will see a graphical representation of the template configuration comparison between what's already deployed to the site and what's defined in the template..



a) The color-coded legend indicates which objects would be created, deleted, or modified if you were to deploy the template at this time.

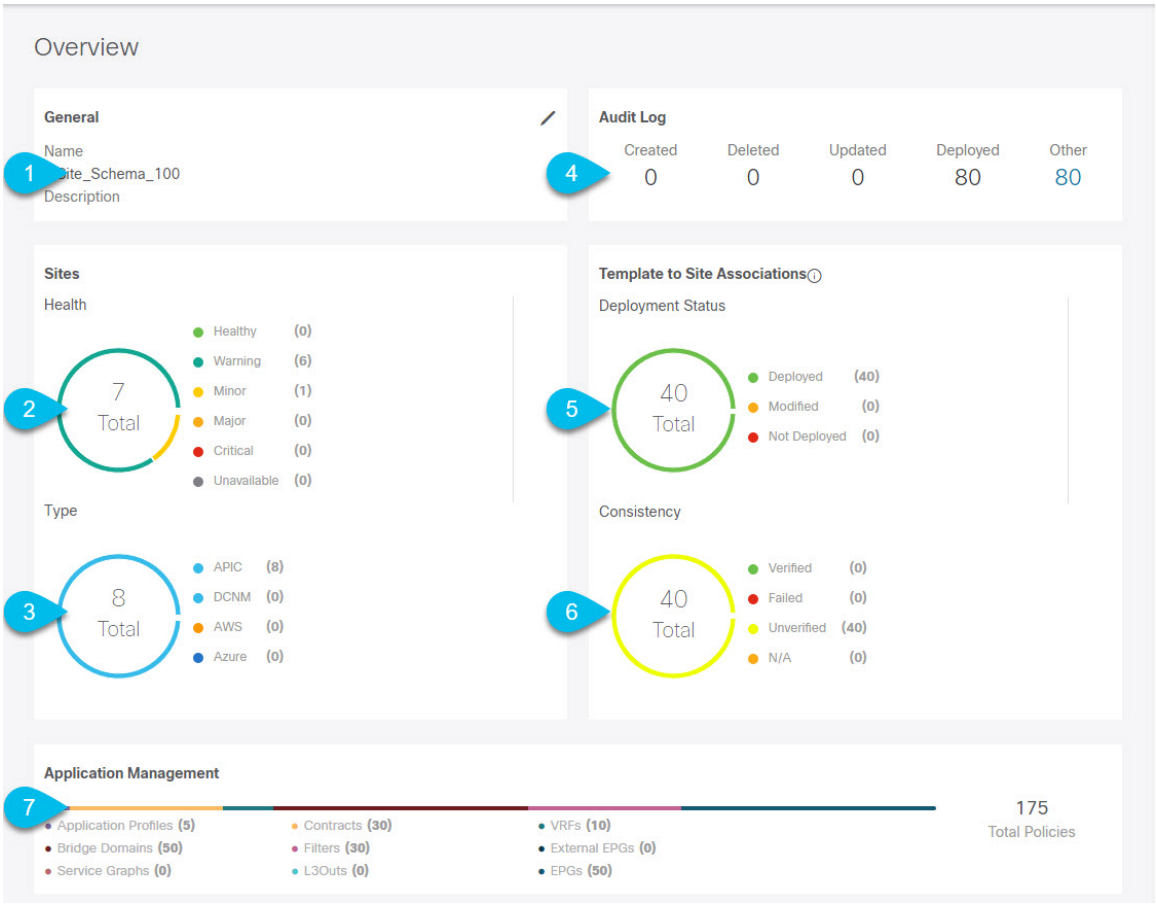
   If the latest version of the template is already deployed, the view will not contain any color-coded objects and will simply display the currently deployed configuration.

b) You can click on a site name to show configuration for that specific site.
c) You can click **View XML/JSON** to see the XML config of all the objects that are deployed to the selected site.

Schema Overview and Deployment Visualizer

When you open a schema with one or more objects defined and deployed to one or more ACI fabrics, the schema **Overview** page will provide you with a summary of the deployment.

Figure 1: Schema Overview

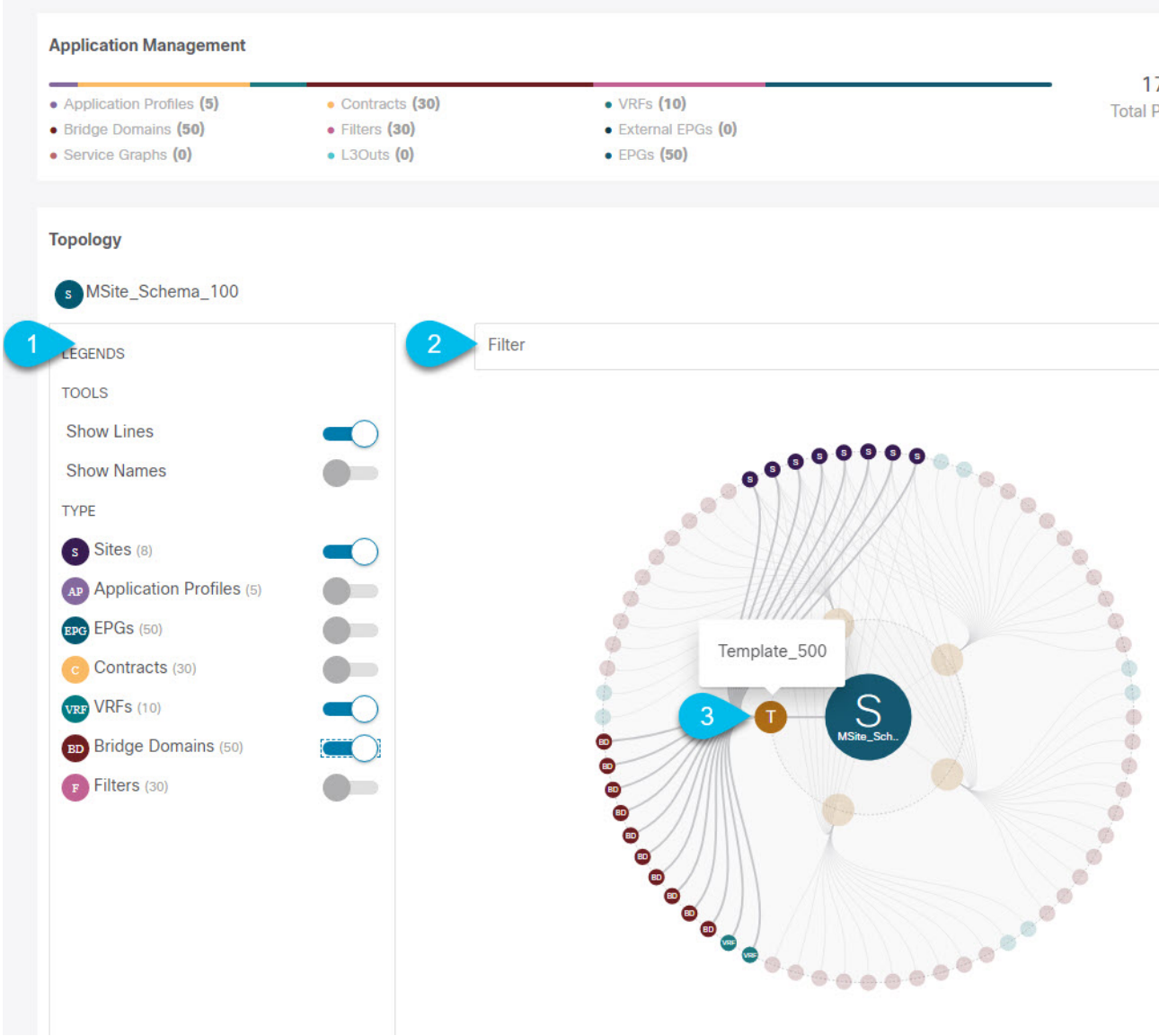


The following details are provided on this page:

1. **General**—Provides general information of the schema, such the name and description.
2. **Audit Log**—Provides audit log summary of the actions performed on the schema.
3. **Sites** > **Health**—Provides the number of sites associated with the templates in this schema sorted by the site's health status.
4. **Sites** > **Type**—Provides the number of sites associated with the templates in this schema sorted by the site's type.
5. **Template to Site Associations** > **Deployment Status**—Provides the number of templates in this schema that are associated with one or more sites and their deployment status.
6. **Template to Site Associations** > **Consistency**—Provides the number of consistency checks performed on the deployed templates and their status.
7. **Application Management**—Provides a summary of individual objects contained by the templates in this schema.

The **Topology** tile allows you to create a topology visualizer by selecting one or more objects to be displayed by the diagram as shown in the following figure.

Figure 2: Deployment Visualizer



1. **Legend**—Allows you to choose which policy objects to display in the topology diagram below.
2. **Filter**—Allows you to filter the displayed objects based on their names.
3. **Topology Diagram**—Provides visual representation of the policies configured in all of the Schema's templates that are assigned to sites.

   You can choose which objects you want to display using the **Configuration Options** above.

   You can also mouse over an objects to highlight all of its dependencies.

   Finally, you can click on any object in the diagram to zoom in to see only its relationships with other objects. For example, clicking a Template will display all objects within that specific template only.