Fabric Management

- Tenants, on page 1
- Schemas and Templates, on page 2
- Concurrent Configuration Updates, on page 3
- Creating Schemas and Templates, on page 5
- Template Versioning, on page 11
- Template Review and Approval, on page 15
- Deploying Templates, on page 18
- Viewing Currently Deployed Configuration, on page 19
- Schema Overview and Deployment Visualizer, on page 20

Tenants

A tenant is a logical container for application policies that enable an administrator to exercise domain-based access control. A tenant represents a unit of isolation from a policy perspective, but it does not represent a private network. Tenants can represent a customer in a service provider setting, an organization or domain in an enterprise setting, or just a convenient grouping of policies.

To manage tenants, you must have either `Power User` or `Site and Tenant Manager` read-write role.

Three tenants are pre-configured for you:

- `common`—A special tenant with the purpose of providing "common" services to other tenants in ACI fabrics. Global reuse is a core principle in the common tenant. Some examples of common services include shared L3Outs, DNS, DHCP, Active Directory, and shared private networks or bridge domains.
- `dcnm-default-tn`—A special tenant with the purpose of providing configuration for Cisco DCNM fabrics.
- `infra`—The Infrastructure tenant that is used for all internal fabric communications, such as tunnels and policy deployment. This includes switch to switch and switch to APIC communications. The `infra` tenant does not get exposed to the user space (tenants) and it has its own private network space and bridge domains. Fabric discovery, image management, and DHCP for fabric functions are all handled within this tenant.

When using Nexus Dashboard Orchestrator to manage Cisco DCNM fabrics, you will use the default `dcnm-default-tn` that is preconfigured for you and allows you to create and manage the following objects:

- VRFs

- Networks

Schemas and Templates

A schema is a collection of templates, which are used for defining networking configuration, with each template assigned to a specific tenant. A template is a set of configuration objects and their properties that you deploy all at once to one or more sites. There are multiple approaches you can take when it comes to creating schema and template 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 environment.

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 *Cisco Multi-Site Verified Scalability Guides* for your release.

Single Schema Deployment

The simplest schema design approach is a single schema deployment. You can create a single schema with all VRFs and Networks in that schema. You can then create a single application profile or multiple application profiles within the templates 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 or templates per schema scalability limit may make this approach unsuitable for large scale deployments, which could exceed those limits.

Multiple Schemas 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. And create different templates containing the shared objects.

For example, you can use the following templates for a configuration that you plan to deploy to 3 different sites:

- 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.

Template Design

In this release, we recommend creating separate templates for VRFs and Networks within each schema and then deploying the VRF templates first, followed by the templates that contain Networks. This way any VRFs required by the networks will be already created when you push Network configuration to the sites.

Similarly, when undeploying multiple networks and VRFs, we recommend undeploying the Networks template first, followed by the VRF templates. This will ensure that when VRFs are undeployed, there will be no conflicts with any existing Networks still using them.

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"
}
```

Creating Schemas and Templates

Before you begin

- You must have an administrative user account with full read/write privileges.
- You must have a tenant user account with read/write tenant policy privileges.

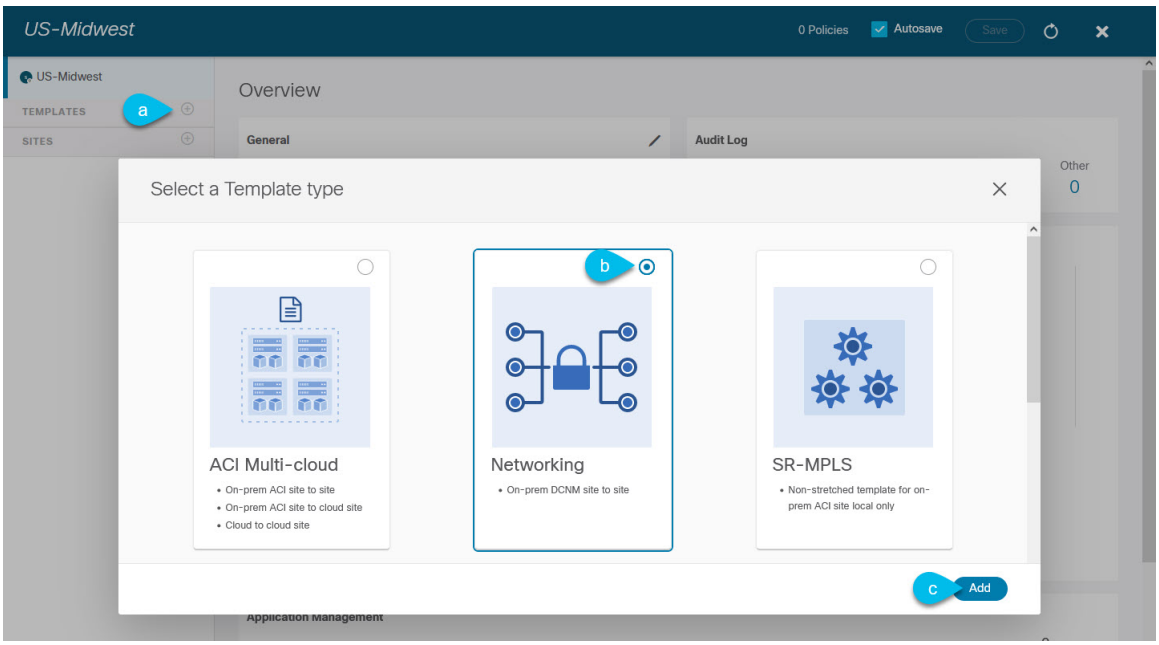
Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

Step 2 Create a new schema.

a) From the left navigation pane, choose **Application Management** > **Schemas**.
b) On the Schemas page, click **Add Schema**.
c) In the schema creation dialog, provide the **Name** and optional description for the schema.

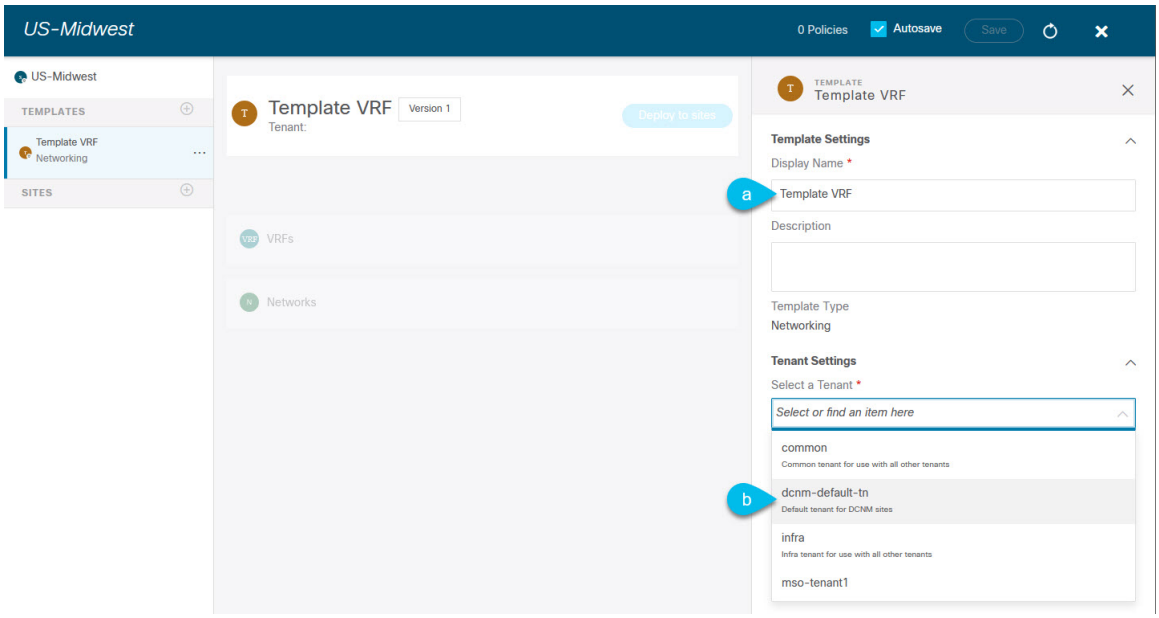
By default, the new schema is empty, so you need to add one or more templates.

Step 3 Create a template.



a) In the left sidebar under **Templates**, click the + sign to add a new template.
b) In the **Select a Template type** window, choose **Networking** for the template type.
c) Click **Add** to create the template.

Step 4 Provide the name and the tenant for the template.



a) In the right sidebar, specify the **Display Name** for the template.
b) From the **Select a Tenant** dropdown, select the `dcnm-default-tn` tenant.

   Keep in mind, the user account you're using to create a new schema must be associated with the tenant you are trying to add to it, otherwise the tenant will not be available in the drop-down menu.

Step 5 Assign the templates to sites.

You deploy one template at a time, so you need to associate the template with at least one site where you want to deploy the configuration.

a) In the left pane, click the + icon next to Sites
b) In the **Add Sites** window, check the checkbox next to the sites where you want to deploy the template.
c) From the **Assign to Template** dropdown next to each site, select one or more templates.

   While you deploy one template at a time to every site with which it is associated, you can associate multiple templates to a site at once.

d) Click **Save**.

Importing Schema Elements From DCNM Sites

You can create new objects and push them out to one or more sites or you can import existing site-local objects and manage them using the Nexus Dashboard Orchestrator. This section describes how to import one or more existing objects, while creating new objects is described later on in this document.

Step 1 Open the **Schema** where you want to import objects.
Step 2 In the left sidebar, select the **Template** where you want to import objects.
Step 3 In the main pane click the **Import** button and select the **Site** from which you want to import.

Step 4 In the **Import from <*site-name*>** window that opens, select one or more objects.

Note The names of the objects imported into the Nexus Dashboard Orchestrator must be unique across all sites. Importing different objects with duplicate names will cause a schema validation error and the import to fail. If you want to import objects that have the same name, you must first rename them.

Creating VRFs

This section describes how to create a VRF.

Before you begin

You must have the schema and template created and a tenant assigned to the template, as described in Creating Schemas and Templates, on page 5.

Step 1 Select the schema and template where you want to create VRF.

Step 2 In the schema edit view, choose **Create Object** > **VRF**.

Step 3 In the properties pane on the right, provide **Display Name** for the VRF.

Step 4 Configure the **DCNM Properties** for the VRF.

a) (Optional) Provide the **VRF ID**.

You can choose to specify the VNI of the VRF or leave the field empty and the VNI will be automatically allocated by the NDO from the ranges you specified in Configuring Infra: General Settings.

b) From the **VRF Profile** dropdown, select the VRF profile.

You can assign the `Default_VRF_Universal` profile or choose any available VRF Profile that had been previously created in DCNM. Any profiles created in DCNM are automatically imported into the NDO and are available for selection here.

c) From the **VRF Extension Profile** dropdown, select the extension profile.

You can assign the `Default_VRF_Extension_Universal` profile or choose any available VRF Extension Profile that had been previously created in the DCNM. Any profiles created in DCNM are automatically imported into the NDO and are available for selection here.

d) Provide the **Loopback Routing Tag**.

If a VLAN is associated with multiple subnets, then this tag is associated with the IP prefix of each subnet. Note that this routing tag is associated with overlay network creation too.

e) Provide the **Redistribute Direct Route Map**.

Specifies the route map name for redistribution of routes in the VRF.

f) (Optional) Check **Disable RT Auto-Generate** to disable automatic generation of route targets.

Note This feature is supported in Nexus Dashboard Orchestrator, Release 3.5(2) and later.

By default when this option is unchecked, the route targets (RTs) are generated by the switches and you can choose to generate custom RTs in addition to the existing auto-generated ones.If you enable this option, the automatic generation of RTs will be disabled and you can use only the custom RTs.

g) (Optional) Provide any custom route targets.

Note This feature is supported in Nexus Dashboard, Release 3.5(2) and later.

To provide custom RTs, enter one or more values for the following fields:

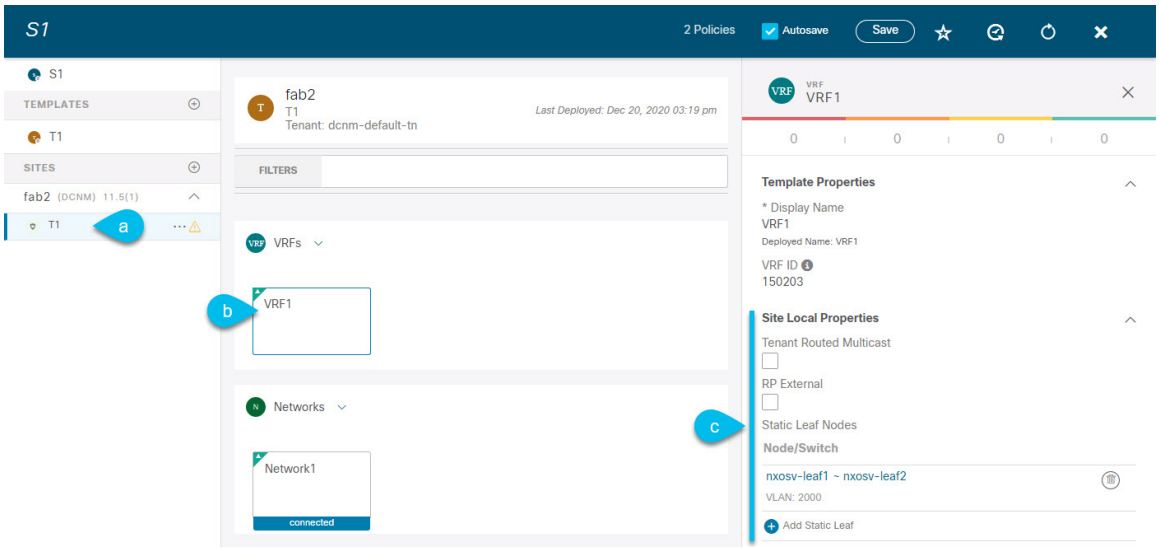
- **Import**—for VPN routes import
- **Export**—for VPN routes export
- **Import EVPN**—for EVPN routes import
- **Export EVPN**—for EVPN routes export

You must enter a valid value, for example `12.2.3.4:2200`. As you type in a value, the UI will validate it and once the format is correct, you will see a `Create "<value>"` option in the dropdown.

You can provide up to 10 custom route target values in total.

Step 5 Configure the VRF's site-local properties.

In addition to the network's general properties that apply to every site where the VRF is deployed, you can configure site-specific properties for this VRF individually for each site.



a) In the left sidebar under **SITES**, select the template where the network is defined.
b) In the main pane, select the network.
c) In the right **Properties** sidebar, provide the site-specific settings.

You can configure the following site-local properties:

- Enable **Tenant Routed Multicast**—Tenant Routed Multicast (TRM) enables multicast forwarding on the VXLAN fabric that uses a BGP-based EVPN control plane. TRM provides multi-tenancy aware multicast forwarding between senders and receivers within the same or different subnets local or across VTEPs.

  If you enable TRM, you must also provide the **RP Address** and **Overlay Multicast Group**.

- Enable **RP External** if the Rendezvous Point (RP) is external to the fabric.
- Click **Add Static Leaf** to select one or more leaf switches where the VRF will be configured.

In the **Add Static Leaf** window that opens, choose the leaf node and provide the VLAN ID for the VRF.

Creating Networks

This section describes how to configure a DCNM network from Nexus Dashboard Orchestrator.

Before you begin

- You must have the schema and template created and a tenant assigned to the template, as described in Creating Schemas and Templates, on page 5.
- You must have the VRF created as described in Creating VRFs, on page 7

Step 1 Select the schema and template where you want to create the application profile.

Step 2 In the schema edit view, choose **Create Object** > **Network**.

Step 3 In the properties pane on the right, provide **Display Name** for the network.

Step 4 (Optional) Provide the **Network ID**.

You can choose to specify the network ID or leave the field empty and the ID will be automatically allocated by the NDO when you save the schema.

Step 5 Choose whether or not this is a **Layer2 Only** network.

Step 6 From the **Virtual Routing & Forwarding** dropdown, select the VRF you created for this network.

This option will be unavailable if you enabled **Layer2 Only**.

Step 7 From the **Network Profile** dropdown, select the network profile.

You can assign the `Default_Network_Universal` profile or choose any available Network Profile that had been previously created in DCNM. Any profiles created in DCNM are automatically imported into the NDO and are available for selection here.

Step 8 From the **Network Extension Profile** dropdown, select the network extension profile.

You can assign the `Default_Network_Extension_Universal` profile or choose any available Network Extension Profile that had been previously created in the DCNM. Any profiles created in DCNM are automatically imported into the NDO and are available for selection here.

Step 9 Provide the **VLAN ID** for the network.

Step 10 Provide the **VLAN Name**.

Step 11 Add one or more **Subnets**.

This option will be unavailable if you enabled **Layer2 Only**.

a) Click **+Add Subnet**.

An **Add Subnet** window opens.

b) Click **+Add Gateway IP** and enter the subnet's **Gateway IP** address.

You can configure up to four gateway IPs.

c) Choose `Primary` for the first gateway you add.
d) Click the checkmark to save the gateway information.
e) Repeat the previous substeps to provide additional gateways.
f) Click **Add** to finish adding the subnet.

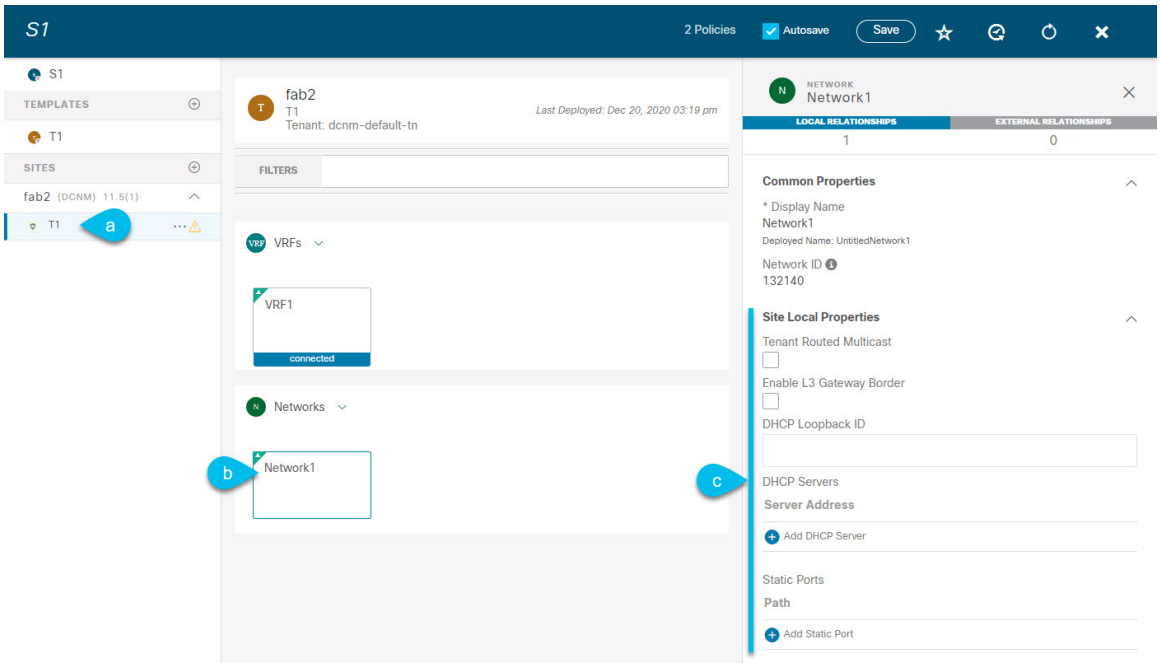
Step 12 Choose whether you want to **Suppress ARP**.

Step 13 Provide the **MTU** for this network.

Step 14 Provide the **Routing Tag**.

Step 15 Configure the network's site-local properties.

In addition to the network's general properties that apply to every site where the network is deployed, you can configure site-specific properties for this network individually for each site.



a) In the left sidebar under **SITES**, select the template where the VRF is defined.
b) In the main pane, select the VRF.
c) In the right **Properties** sidebar, provide the site-specific settings.

You can configure the following site-local properties:

- Enable **Tenant Routed Multicast**—Tenant Routed Multicast (TRM) enables multicast forwarding on the VXLAN fabric that uses a BGP-based EVPN control plane. TRM provides multi-tenancy aware multicast forwarding between senders and receivers within the same or different subnets local or across VTEPs.
- Check **Enable L3 Gateway Border** to enable Layer 3 SVI on the border gateways to allow connecting dual-attached hosts to it.
- Provide the **DHCP Loopback ID**.

  The value must be in the `0-1023` range.
- Click **+Add DHCP Server** to add one or more DHCP relay servers.

In the **Add DHCP Server** window that opens, provide the IP address of the DHCP relay and the VRF to which it belongs.

- Click **+Add Static Port** to add one or more ports to which the network's VLAN will be attached.

  In the **Add Static Port** window that opens, select the leaf switch that contains the port, provide the VLAN ID, and finally click **Add Port** to specify one or more ports for the network.

  Note that if you want to add multiple static ports from different leaf switches, you will need to repeat the process for each leaf switch separately.

Template Versioning

Release 3.4(1) adds support for 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.

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.

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 **Set as Golden**.

If the template is already tagged, the option will change to **Remove Golden** 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.

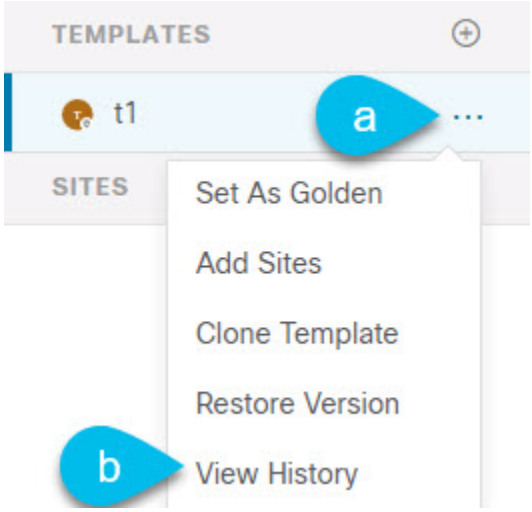
Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

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

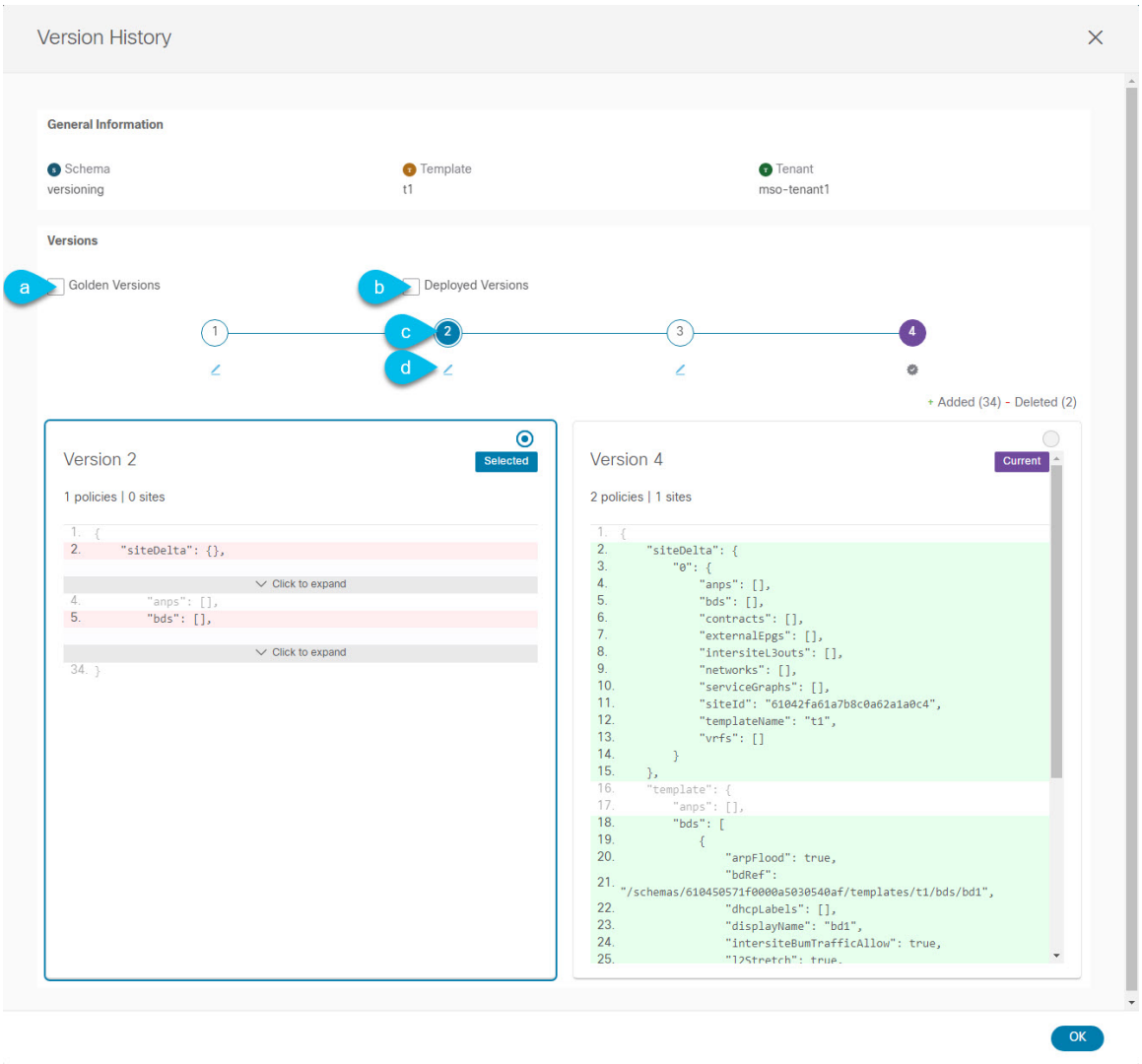
Step 3 Click the schema that contains the template you want to view.

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

Step 5 From the template's actions (**...**) menu, select **View 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 11.

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.

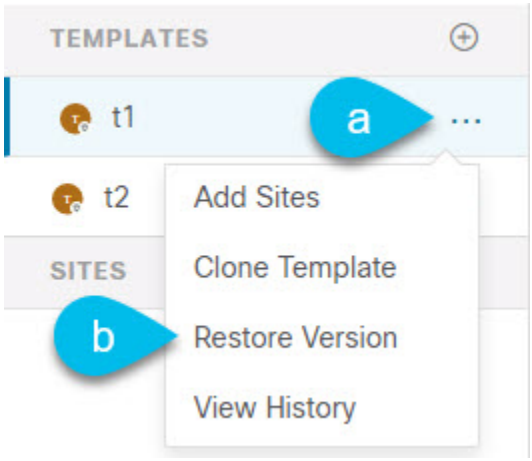
Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

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 **Restore Version**.



Step 6 In the **Restore Version** 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 as described in Deploying Templates, on page 18.

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 15 sections.

Template Review and Approval

Release 3.4(1) adds support for 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 deployer 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.

Note Because the review and approval workflow depends on user roles that are defined in Nexus Dashboard, you must be running Nexus Dashboard release 2.1(1) or later to use this feature. If you deployed your Nexus Dashboard Orchestrator in Nexus Dashboard release 2.0.2, the review and approval feature will be disabled until you upgrade your platform.

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.

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, check the **Change Control Workflow** checkbox 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.

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 **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.

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 15.

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 15.
- Created or updated users in Nexus Dashboard with `approver` and `deployer` roles, as described in Create Users with Required Roles, on page 16.

- Created a template with one or more policy configurations and assigned it to one or more sites.

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

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

Step 3 Click the schema that contains the template for which you want to request approval.

Step 4 In the schema view, select the template.

Step 5 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 15.
- Created or updated users in Nexus Dashboard with `approver` and `deployer` roles, as described in Create Users with Required Roles, on page 16.
- 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 16.

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 11.

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 19.

What to do next

After the template is reviewed and approved by the required number of approvers, you can deploy the template as described in Deploying Templates, on page 18.

Deploying Templates

This section describes how to deploy new or updated configuration to DCNM fabrics.

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.

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

Step 2 Assign the templates to sites.

If you have already assigned the templates to sites, skip this step.

You deploy one template at a time, so you need to associate the template with at least one site where you want to deploy the configuration.

a) In the left pane, click the + icon next to Sites
b) In the **Add Sites** window, check the checkbox next to the sites where you want to deploy the template.
c) From the **Assign to Template** dropdown next to each site, select one or more templates.

While you deploy one template at a time to every site with which it is associated, you can associate multiple templates to a site at once.

d) Click **Save**.

Step 3 In the left sidebar, select the template you want to deploy.

Step 4 In the top right of the template edit view, click **Deploy to sites**.

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

Step 5 Click **Deploy** to deploy the template.

- If this is the first time you are deploying this template or you have made changes to a previously deployed template, the **Deploy to Sites** summary will show the configuration difference that will be deployed to sites.

Note You can 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**.

- If you have previously deployed this template but made no changes to it since, the **Deploy to Sites** summary will allow you to re-deploy the entire template only the `Full Template` option and you can simply click **Deploy** to re-deploy the entire template.

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 `VRF2` and redeploy, the API will return both objects, `VRF1` and `VRF2`, 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.

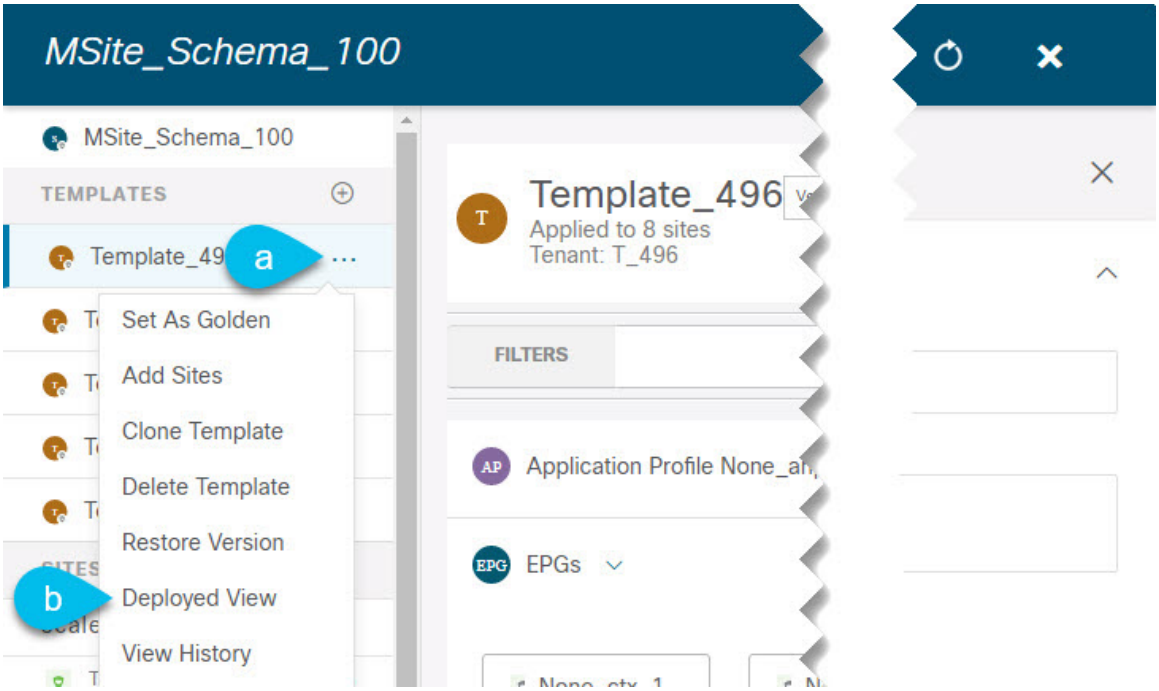
Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

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.

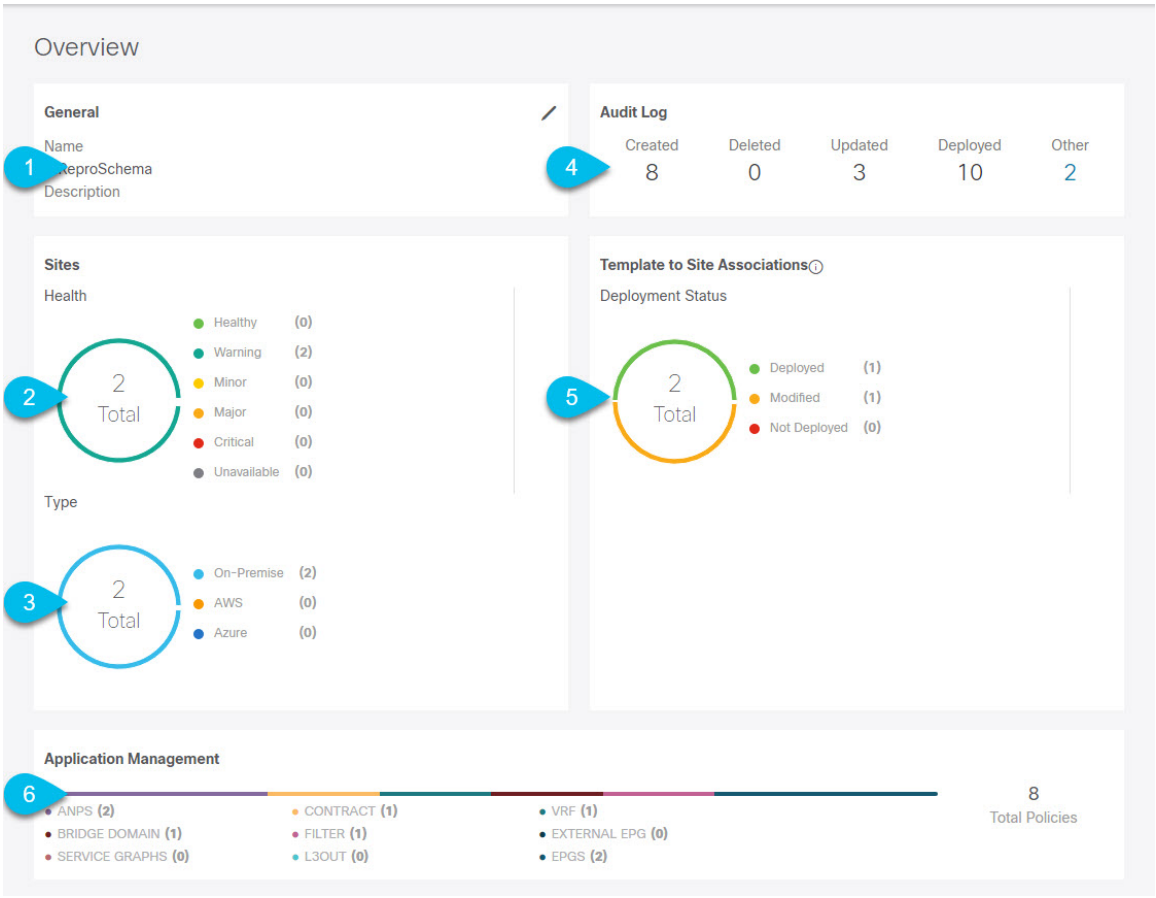
   If the latest version of the template is already deployed, the view will not contain any color-coded objects and will simply display the currently deployed configuration.

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 fabrics, the schema **Overview** page will provide you with a summary of the deployment.

Figure 1: Schema Overview

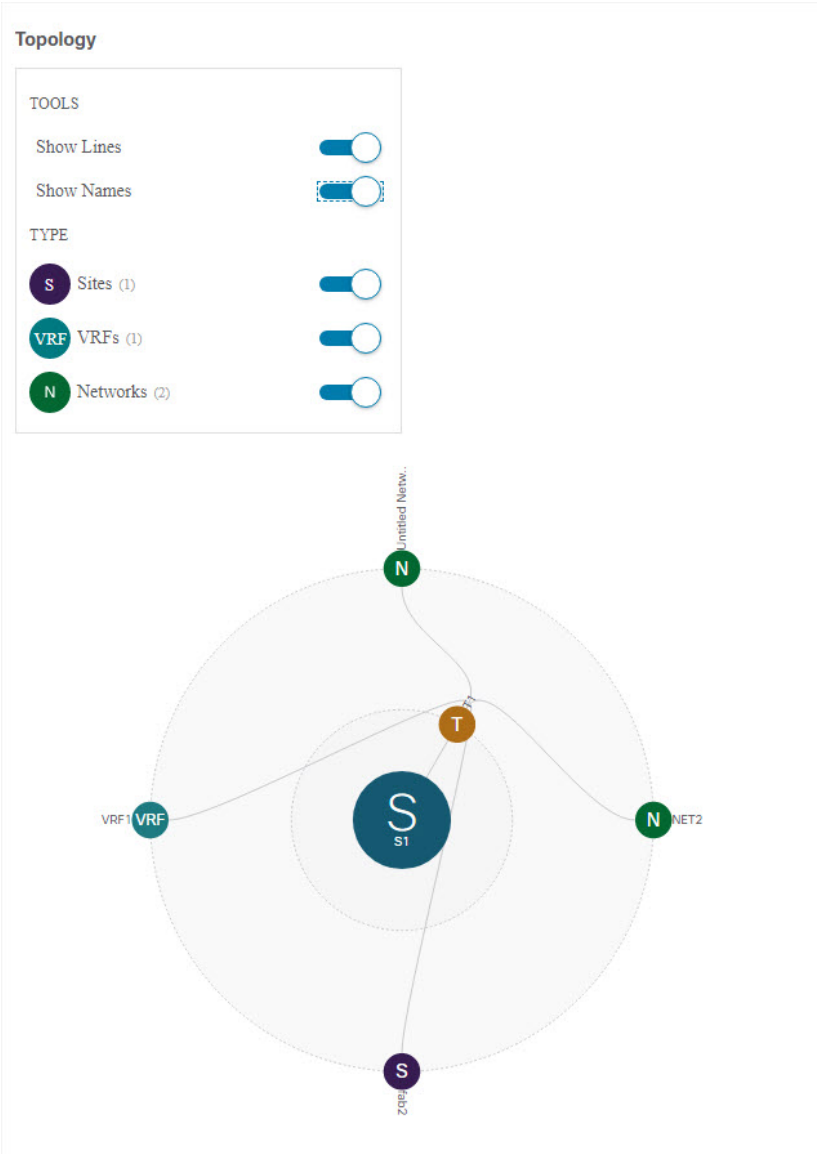


The following details are provided on this page:

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. **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. **Configuration Options**—Allows you to choose which policy objects to display in the topology diagram below.
2. **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.