

Schemas

- Schema Design Considerations, on page 1
- Concurrent Configuration Updates, on page 6
- Creating Schemas and Templates, on page 8
- Bulk Update for Template Objects, on page 24
- Assigning Templates to Sites, on page 27
- Template Versioning, on page 28
- Template Review and Approval, on page 33
- Deploying Templates, on page 36
- Undeploying Templates, on page 40
- Disassociating Template from Sites, on page 41
- Configuration Drifts, on page 42
- Cloning Schemas, on page 47
- Cloning Templates, on page 49
- Migrating Objects Between Templates, on page 50
- Viewing Currently Deployed Configuration, on page 51
- Schema Overview and Deployment Visualizer, on page 53
- Shadow Objects, on page 56

Schema Design Considerations

A schema is a collection of templates, which are used for defining policies, with each template assigned to a specific tenant. 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, 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.

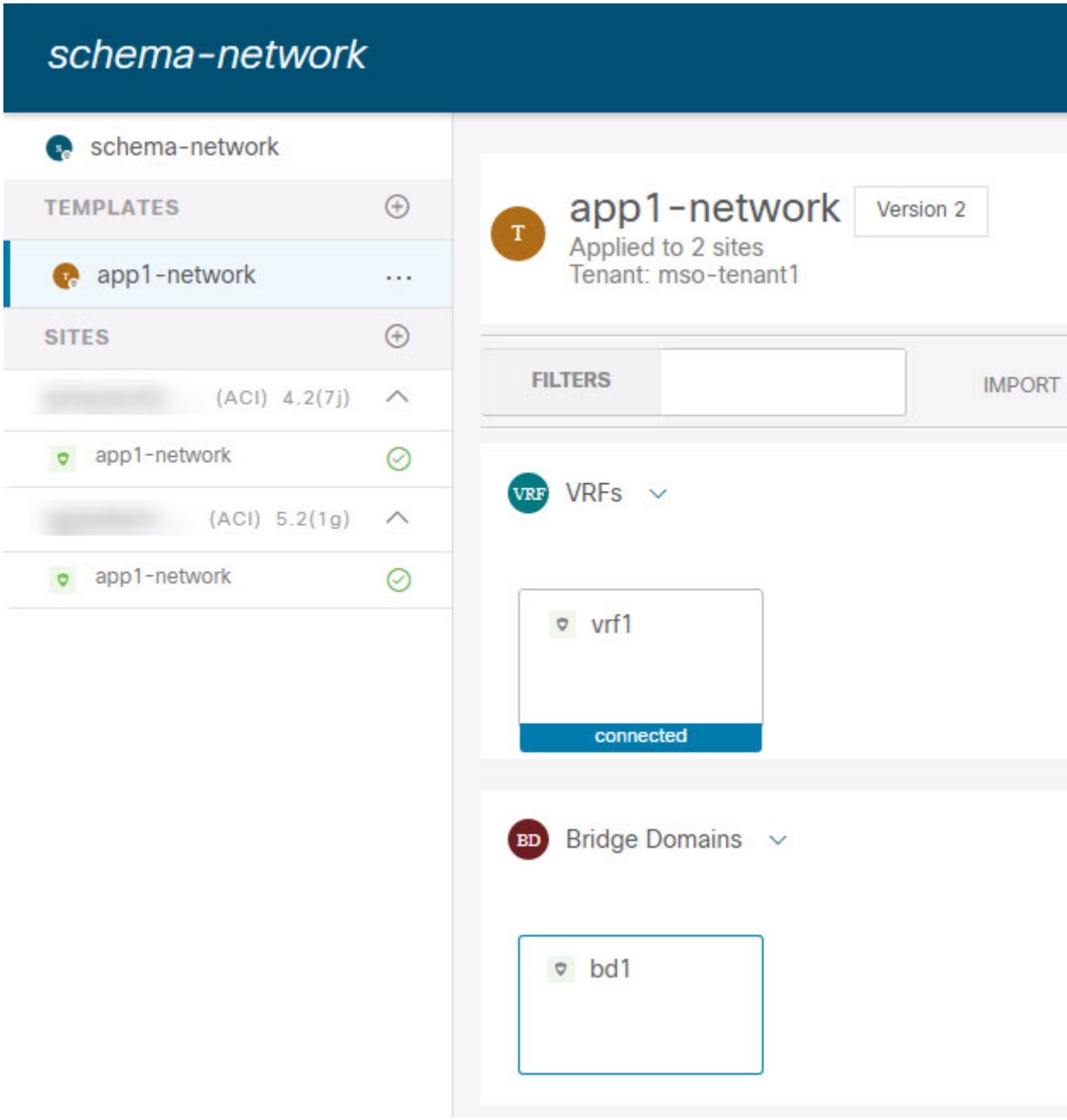
Multiple Schemas with Network Separation

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.

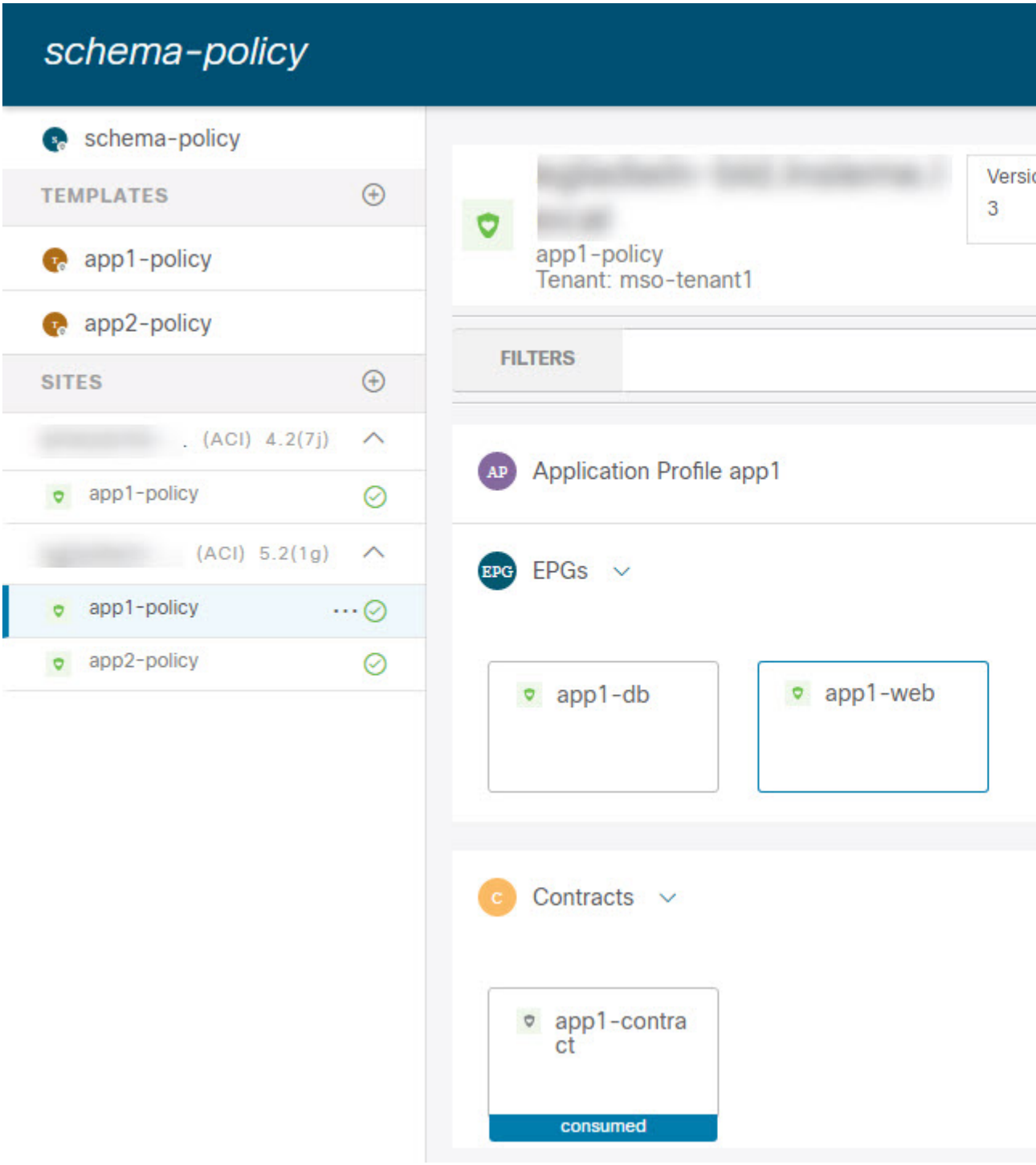
The following figure shows a single networking template configuration with VRF, BD, and subnets configured and deployed to two sites:

Figure 1: Network Schema



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. The following figure shows a policy schema that contains two application templates both of which reference the networking elements in an external schema. One of the applications is local to one site while the other is stretched across two sites:

Figure 2: Policy Schema



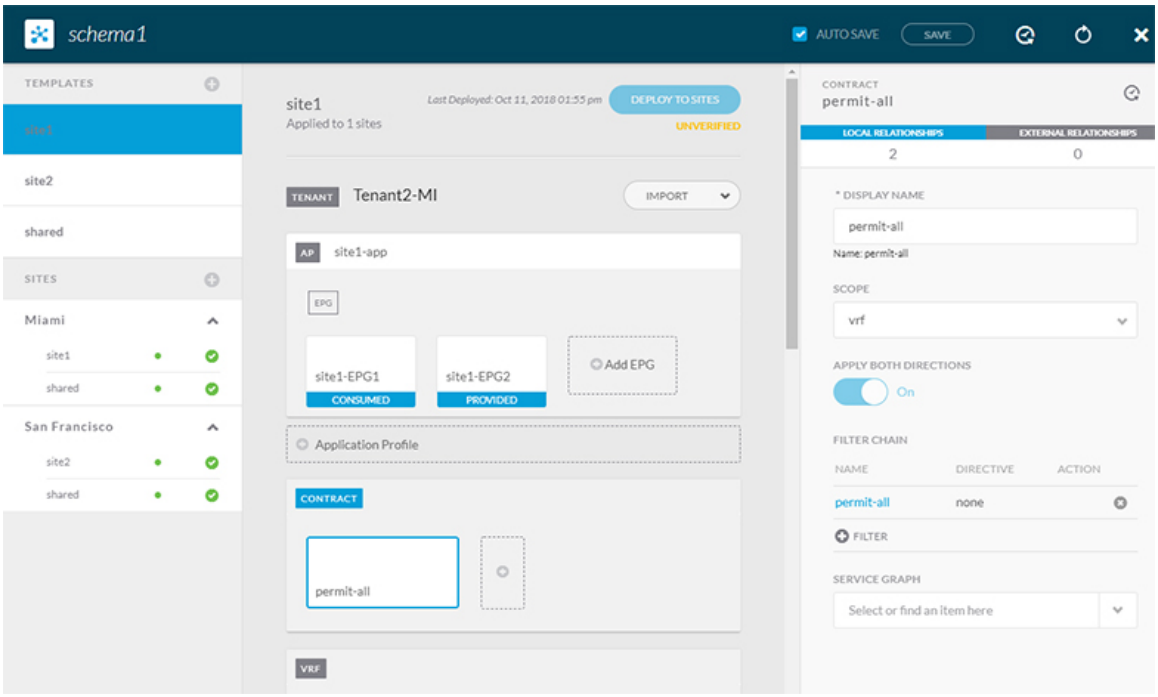
After creating and deploying the policy schemas and templates, the networking objects in the networking schema will display the number of external references by the policy schema elements. The object with external references will also be denoted by the ribbon icon as shown in the "Network Schema" figure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

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 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, including the VRFs, BDs, EPGs, Contracts, and Filters. And create different templates containing the shared objects.

Figure 3: One Template per Site



The **site1** template in the above figure contains only the objects that are local to Site1 and the template is deployed to only the Miami site. Similarly, the **site2** template contains only the object relevant to site2 and is deployed to the San Francisco site. Any change made to any object in either of these templates has no effect on the other one. The **shared** template 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.

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 at least one available tenant that you want to incorporate into your site.

  For more information, refer to Tenants.

Procedure

Step 1 Log in to your Nexus Dashboard and open the Nexus Dashboard Orchestrator service.

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 **ACI Multi-Cloud** for the template type.

- **ACI Multi-Cloud**—Templates used for Cisco ACI on-premises and cloud sites, which allow template and object stretching between multiple sites.

  The following sections focus primarily on this type of template.

- **Networking**—Templates designed for Cisco Nexus Dashboard Fabric Controller (formerly Data Center Network Manager) sites.

  This guide described Nexus Dashboard Orchestrator configurations for Cisco ACI and cloud ACI sites. For information on working with Cisco NDFC sites, see the *Cisco Nexus Dashboard Orchestrator Configuration Guide for DCNM Fabrics* instead.

- **SR-MPLS**—Template designed for SR-MPLS integration.

  For detailed information on the SR-MPLS use case, see "SD-WAN Integration" chapter of this guide.

- **Cloud Local**—Templates designed for specific Cisco Cloud ACI use cases, such as Google Cloud site connectivity, and cannot be stretched between multiple sites.

c) In the right sidebar, specify the **Display Name**.
d) (Optional) Provide a **Description**.
e) If you are configuring an SR-MPLS template, enable the **SR-MPLS** knob.

For detailed information on SR-MPLS templates, see Sites Connected via SR-MPLS.

f) From the **Select a Tenant** dropdown, select the Tenant for this template.

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. Associating a user account with a tenant is described in Adding Tenants.

Step 4 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 APIC 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 Multi-Site Orchestrator. This section describes how to import one or more existing objects, while creating new objects is described later on in this document.

When importing policies from APIC into NDO, the common practice is to import some objects, such as VRFs or contracts, into a stretched template and other objects, such as non-stretched EPGs or BDs, into site-local templates.

Prior to Release 3.1(1), importing an object into a site-local template that referenced another object that is part of a stretched template presented certain challenges, for example:

- If a referenced object already exists in NDO and a new object is imported with the **Include Relations** option enabled, NDO would throw an error when trying to deploy the site-local template because of object duplication since the referenced object already existed.
- However, not importing the referenced object (**Include Relations** option disabled) would require an administrator to perform manual mapping with the referenced object after the import.

Beginning with Release 3.1(1), when importing an object into a site-local template that has references with another object that is part of a different template (in the same or a different schema), the references are automatically resolved by NDO. In such cases, the **Import Relations** option will be grayed-out in the UI for the object that is being imported and a warning tooltip will provide additional info, such as: `[Referenced Object] already exists in [Template]. Existing relations are imported by default.` While such objects are imported with their relations by default, you can change the references once the import operation is completed, for example by re-mapping a BD to a different VRF. The new behavior applies to all configuration objects that can be imported.

To import one or more objects from sites:

Procedure

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 NDO 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.

Step 5 (Optional) Enable the **Import Relations** knob to import all related objects.

For example, when importing a BD, enabling the **Import Relations** knob will import the associated VRF as well.

Note
As described previously, the **Import Relations** knob will be enabled by default and cannot be disabled for objects whose related objects already exist in NDO.

Configuring VRFs

This section describes how to configure 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 8.

Procedure

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

a) In the main pane, select **+Create Object** > **VRF**.

Alternatively, you can scroll down to the **VRFs** area, mouse over the tile, and click **Add VRF**.

b) In the right pane, provide the **Display Name** for the VRF.

c) (Optional) Provide a **Description**.

Step 2 Configure the **On-Premises Properties** for the VRF.

a) Specify **Policy Control Enforcement Preference**.

Note that you cannot change the Policy Control Enforcement for newly created VRFs and the setting is locked to the `enforced` mode.

However, you can use this to transition any VRF that you import from an APIC site that is configured as `unenforced` to the `enforced` mode after importing it. A typical use case is for brown field deployments where existing VRFs must

be converted to `enforced` mode to support stretching them between sites. Once you have transitioned an imported VRF from `unenforced` to `enforced` in NDO, you will not be able to make further changes to this field.

- `Enforced`—Security rules (contracts) will be enforced.
- `Unenforced`—Security rules (contracts) will not be enforced.

b) (Optional) Enable **IP Data-Plane Learning**.

Defines if IP addresses are learned through data-plane packets for the VRF.

When disabled, IP addresses are not learned from the data-plane packets. Local and remote MAC addresses are still learned, but local IP addresses are not learned from data packets.

Regardless of whether this parameter is enabled or disabled, local IP addresses can still be learned from ARP, GARP, and ND.

c) (Optional) Enable **L3 Multicast** for the VRF.

For additional information, see Layer 3 Multicast.

d) (Optional) Enable **vzAny** for the VRF.

For additional information, see vzAny Contracts.

Configuring Bridge Domains

This section describes how to configure a Bridge Domain (BD).

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 8.
- You must have the VRF created as described in Configuring VRFs, on page 10

Procedure

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

Step 2 Create a bridge domain.

a) In the main pane, select **+Create Object** > **Bridge Domain**.

Alternatively, you can scroll down to the **Bridge Domains** area, mouse over the tile, and click **Add Bridge Domain**.

b) In the right pane, provide the **Display Name** for the bridge domain.

c) (Optional) Provide a **Description**.

Step 3 Configure **On-Premises Properties**.

a) From the **Virtual Routing & Forwarding** dropdown, select the VRF for this BD.

b) (Optional) Enable **L2 Stretch**.

c) (Optional) Enable **Intersite BUM Traffic Allow**.

This option becomes available if you enabled **L2 Stretch**.

d) (Optional) Enable **Optimized WAN Bandwidth**.

e) (Optional) Enable **Unicast Routing**.

If this setting is enabled and a subnet address is configured, the fabric provides the default gateway function and routes the traffic. Enabling unicast routing also instructs the mapping database to learn the endpoint IP-to-VTEP mapping for this bridge domain. The IP learning is not dependent upon having a subnet configured under the bridge domain.

f) (Optional) Enable **L3 Multicast** for the BD.

For additional information about Layer 3 multicast, see Layer 3 Multicast.

g) (Optional) Choose **L2 Unknown Unicast** mode.

By default, unicast traffic is flooded to all Layer 2 ports. If enabled, unicast traffic flooding is blocked at a specific port, only permitting egress traffic with MAC addresses that are known to exist on the port. The method can be `Flood` or `Hardware Proxy`.

When the BD has L2 Unknown Unicast set to Flood, if an endpoint is deleted the system deletes it from both the local leaf switches as well as the remote leaf switches where the BD is deployed, by selecting Clear Remote MAC Entries. Without this feature, the remote leaf continues to have this endpoint learned until the timer expires.

Note
Modifying the L2 Unknown Unicast setting causes traffic to bounce (go down and up) on interfaces to devices attached to EPGs associated with this bridge domain.

h) (Optional) Choose **Unknown Multicast Flooding** mode.

This is applicable for IPv4 unknown multicast traffic and is the node forwarding parameter for Layer 3 unknown multicast destinations.

- `Flood` (default)—Unknown IPv4 multicast traffic is flooded on all front panel ports attached with the EPGs associated with this bridge domain. Flooding is not restricted to only M-Router ports of the bridge domain.
- `Optimize Flood`—Send the data only to M-router ports in the bridge domain.

i) (Optional) Choose **IPv6 Unknown Multicast Flooding** mode.

This is applicable for IPv6 unknown multicast traffic and is the node forwarding parameter for Layer 3 unknown multicast destinations.

- `Flood` (default)—Unknown IPv6 multicast traffic is flooded on all front panel ports attached with the EPGs associated with this bridge domain. Flooding is not restricted to only M-Router ports of the bridge domain.
- `Optimize Flood`—Send the data only to M-router ports in the bridge domain.

j) (Optional) Choose **Multi-Destination Flooding** mode.

The multiple destination forwarding method for Layer 2 multicast and broadcast traffic.

- `Flood in BD`—Sends the data to all ports on the same bridge domain.
- `Drop`—Drops Packet. Never sends the data to any other ports.
- `Flood in Encapsulation`—Send the data to all the EPG ports with the same VLAN within the bridge domain, except for the protocol packets which are flooded to the entire bridge domain.

Note

This mode is supported only when the **L2 Stretch** option is disabled and is not supported for BDs that are stretched across sites.

k) (Optional) Enable **ARP Flooding**.

Enables ARP flooding, so that the Layer 2 broadcast domain maps IP addresses to the MAC addresses. If flooding is disabled, unicast routing will be performed on the target IP address.

Enables ARP flooding, so that ARP request will be flooded inside the Layer 2 broadcast domain. If the BD is stretched across sites, enabling ARP flooding is only possible in conjunction with enabling **Intersite BUM Traffic Allow**. When ARP flooding is disabled, the leaf receiving the ARP request from a locally connected endpoint will forward it directly to the remote leaf where the target endpoint of the ARP request is connected (if the IP for the remote endpoint is known in the endpoint table) or to the spines (if the IP for the remote endpoint is not known in the endpoint table).

If you set the **L2 Unknown Unicast** mode to `Flood`, the **ARP Flooding** cannot be disabled. If the **L2 Unknown Unicast** mode is set to `Hardware Proxy`, ARP flooding can be enabled or disabled.

l) (Optional) Provide **Virtual MAC Address**.

The BD virtual MAC address and the subnet virtual IP address must be the same for all ACI fabrics for that bridge domain. Multiple bridge domains can be configured to communicate across connected ACI fabrics. The virtual MAC address and the virtual IP address can be shared across bridge domains.

Note
Virtual MAC along with virtual IP subnet should used only for migration of individual sites to NDO-orchestrated multi-site fabric. Once the migration is completed, these flags can be disabled.

Step 4 Add one or more **Subnets** for the BD.

a) Click **+Add Subnet**.

An **Add New Subnet** window opens.

b) Enter the subnet's **Gateway IP** address and a **Description** for the subnet you want to add.

c) If necessary, enable **Treat as virtual IP address** option.

This option along with the **Virtual MAC Address** on the BD can be used for migration scenarios from individual Common Pervasive Gateway configuration to NDO-orchestrated Multi-Site deployments.

d) Select the **Scope** for the subnet.

The network visibility of the subnet.

- `Private to VRF`—Prevents the subnet from being announced via L3Out toward an external network domain.
- `Advertised Externally`—The subnet can be announced via L3Out toward an external network domain.

e) (Optional) Enable **Shared Between VRFs**.

`Shared between VRFs`—The subnet can be shared with and exported to multiple contexts (VRFs) in the same tenant or across tenants as part of a shared service. An example of a shared service is a routed connection to an EPG present in another context (VRF) in a different tenant. This enables traffic to pass in both directions across contexts (VRFs). An EPG that provides a shared service must have its subnet configured under that EPG (not under a bridge domain), and its scope must be set to advertised externally, and shared between VRFs.

Shared subnets must be unique across the contexts (VRF) involved in the communication. When a subnet under an EPG provides a Layer 3 external network shared service, such a subnet must be globally unique within the entire ACI fabric.

f) Leave the **No Default SVI Gateway** option unchecked.

Enabling this options means that only the proxy route (subnet route to spine proxy) is programmed on the leaf switches and no SVI is created, which means SVI cannot be used as the gateway.

We recommend that SVI is created by the BD subnet as the gateway and the **No Default SVI Gateway** option is enabled on the EPG instead because EPG subnets should only be used for route leaking.

g) (Optional) Enable **Querier** option.

Enables IGMP Snooping on the subnet

h) (Optional) Enable **Primary** option to designate the subnet as primary.

There can be one primary IPv4 subnet and one primary IPv6 subnet.

i) Click **Save**.

Step 5 (Optional) Add a **DHCP Policy**.

For additional information, see DHCP Relay.

Step 6 Configure the bridge domain's site-local properties as necessary.

In addition to the template-level configurations, you can also define one or more site-local properties for the bridge domain, as described in Configuring Bridge Domain's Site-Local Properties, on page 14

Configuring Bridge Domain's Site-Local Properties

In addition to the template-level properties you typically configure for the object when you create it in a template, you can also define one or more properties that are specific to each site to which you assign the template.

When you deploy the object to more than 1 site, the same template-level configurations are deployed to all sites, while the site-local configurations are deployed to those specific sites only.

Before you begin

You must have:

- Created the bridge domain and configured its template-level properties, as described in Configuring Bridge Domains, on page 11.
- Assigned the template that contains the bridge domain to one or more sites.

Procedure

Step 1 Open the schema that contains the template with the bridge domain.

Step 2 In the left sidebar, select the template that contains the bridge domain under the specific site that you want to configure.

Step 3 In the main pane, select the bridge domain.

For most fields, you will see the values you have configured at the template level, which you cannot edit here.

Step 4 Click **+L3Out** to add an L3Out.

This is required to advertise the BD subnet out of the remote L3Out and ensure that inbound traffic to the BD can be maintained even if the local L3Out failed. In this case, you would also need to configure the subnet with the `Advertised Externally` flag. For more information, see the Intersite L3Out use case.

Step 5 Enable **Host Route**.

This enables Host Based Routing on the bridge domain. When this knob is enabled, the border leaf switches will also advertise individual end-point (EP) host-routes (`/32` or `/128` prefixes) along with the subnet. The host-route information is advertised only if the host is connected to the local Pod. If the EP is moved away from the local Pod or once the EP is removed from EP database, the route advertisement is then withdrawn.

Step 6 If necessary, change the **SVI MAC Address**.

The SVI MAC addresses must be unique per site, when virtual MAC and virtual IP are enabled for Common Pervasive Gateway (CPG) scenario. This field can also be used when CPG is not enabled, which will change the default router MAC of the BD

Configuring Application Profiles and EPGs

This section describes how to configure an Application Profile and an EPG.

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 8.

This section also assume you have a Contract and a Bridge Domain created.

Procedure

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

Step 2 Create an application profile.

a) In the main pane, select **+Create Object** > **Application Profile**.

Alternatively, you can scroll down to the **Application Profile** area, mouse over the tile, and click **Add Application Profile**.

b) In the right pane, provide the **Display Name** for the application profile.

You can create application profiles with the same name in different templates without any conflicts. You cannot however create other objects (such as VRFs, BDs, EPGs) with the same name in different templates if they will be deployed to the same site and tenant.

c) (Optional) Provide a **Description**.

Step 3 Create an EPG.

a) In the main pane, select **+Create Object** > **EPG**, then select the application profile where you want to create the EPG.

   Alternatively, you can scroll down to the specific **Application Profile** area, mouse over the **EPGs** tile, and click **Add EPG**.

b) In the right pane, provide the **Display Name** for the EPG.
c) (Optional) Provide a **Description**.

Step 4 Add a contract for the EPG.

Creating contracts and filters is described in detail in Configuring Contracts and Filters, on page 20. If you already have a contract created:

a) Click **+ Contract**.
b) On the **Add Contract** dialog, enter the contract name and type.
c) Click **SAVE**.

Step 5 From the **Bridge Domain** dropdown, select the bridge domain for this EPG.

If you are configuring an on-premises EPG, you must associate it with a bridge domain.

Step 6 (Optional) Click **+ Subnet** to add a subnet to your EPG.

You may choose to configure a subnet on the EPG level rather than the bridge domain level, for example for a VRF route-leaking use-case.

a) On the **Add Subnet** dialog, enter the **Gateway IP** address and a description for the subnet you plan to add.
b) In the **Scope** field select either **Private to VRF** or **Advertised Externally**.
c) Click the check box for **Shared Between VRFs** if appropriate.
d) Click the check box for **No Default SVI Gateway** if appropriate.
e) Click **OK**.

Step 7 (Optional) Enable microsegmentation.

If you are configuring a microsegmentation EPG (uSeg), you must provide one or more uSeg attributes for matching endpoints to the EPG.

a) Check the **uSeg EPG** checkbox.
b) Click **+uSeg Attribute**.
c) Provide the **Name** and **Type** for the uSeg attribute.
d) Based on the attribute type you have selected, provide the attribute details.

   For example, if you have selected `MAC` for the attribute type, provide the MAC address to identify an endpoint in this EPG.

e) Click **SAVE**.

Step 8 (Optional) Enable intra-EPG isolation.

By default, endpoints in EPG can freely communicate with each other. If you would like to isolate the endpoints from each other, set the isolation mode to **Enforced**.

Intra-EPG endpoint isolation policies provide full isolation for virtual or physical endpoints; no communication is allowed between endpoints in an EPG that is operating with isolation enforced. Isolation-enforced EPGs reduce the number of EPG encapsulations required when many clients access a common service but are not allowed to communicate with each other.

Step 9 (Optional) Enable Layer 3 multicast for the EPG.

For additional information about Layer 3 multicast, see Layer 3 Multicast

Step 10 (Optional) Enable preferred group membership for the EPG.

The Preferred Group feature allows you to include multiple EPGs within a single VRF to allow full communication between them with no need for contracts to be created. For additional information about EPG preferred group, see EPG Preferred Groups Overview and Limitations

Step 11 Configure the EPG's site-local properties as necessary.

In addition to the template-level configurations, you can also define one or more site-local properties for the EPG, as described in Configuring EPG's Site-Local Properties, on page 17

Configuring EPG's Site-Local Properties

In addition to the template-level properties you typically configure for the object when you create it in a template, you can also define one or more properties that are specific to each site to which you assign the template.

When you deploy the object to more than 1 site, the same template-level configurations are deployed to all sites, while the site-local configurations are deployed to those specific sites only.

Before you begin

You must have:

- Created the application profile and EPG and configured the template-level properties, as described in Configuring Application Profiles and EPGs, on page 15.
- Assigned the template that contains the bridge domain to one or more sites.

Procedure

Step 1 Open the schema that contains the template with the EPG.

Step 2 In the left sidebar, select the template that contains the EPG under the specific site that you want to configure.

Step 3 In the main pane, select the EPG.

For most fields, you will see the values you have configured at the template level, which you cannot edit here.

Step 4 Add one or more **Subnets** for the EPG.

a) Click **+Add Subnet**.

An **Add New Subnet** window opens.

b) Enter the subnet's **Gateway IP** address and a description for the subnet you want to add.

c) Select the **Scope** for the subnet.

The network visibility of the subnet.

- `Private to VRF`—Prevents the subnet from being announced via L3Out toward an external network domain.
- `Advertised Externally`—The subnet can be announced via L3Out toward an external network domain.

d) (Optional) Enable **Shared Between VRFs**.

`Shared between VRFs`—The subnet can be shared with and exported to multiple contexts (VRFs) in the same tenant or across tenants as part of a shared service. An example of a shared service is a routed connection to an EPG present in another context (VRF) in a different tenant. This enables traffic to pass in both directions across contexts (VRFs). An EPG that provides a shared service must have its subnet configured under the BD (not under the EPG), and its scope must be set to advertised externally, and shared between VRFs.

Shared subnets must be unique across the contexts (VRF) involved in the communication. When a subnet under an EPG provides a Layer 3 external network shared service, such a subnet must be globally unique within the entire ACI fabric.

e) (Optional) Enable **No Default SVI Gateway**.

Enabling this options means that only the proxy route (subnet route to spine proxy) is programmed on the leaf switches and no SVI is created, which means SVI cannot be used as the gateway.

We recommend enabling this option on the EPG subnets, which should only be used for route leaking and leaving this option disabled on the BD subnets so that the SVI can be used as a gateway.

f) Click **Save**.

Step 5 Add one or more **Static ports**.

a) Click **+Static Port**.
b) From the **Path Type** dropdown, select the type of port.
c) If configuring a physical interface, select the **Pod**
d) Choose whether you want to configure a single port or a range of ports.

For the interface configuration, you will have an option to do it either by entering a single **Leaf** and a **Path** or by entering a range of **Leaf** eg 120-125 and **Path** eg1/17-20. You will also have an option to enter a range of **Leaf** and associate it with one single **Path**, or enter a range of **Path** for one single **Leaf**.

However, after the configuration it will still be displayed as individual ports in the UI and will require individual changes for any future updates.

e) Select the **Port Encap VLAN**.

When manually configuring the port encap on a domain for an EPG, the VLAN ID must belong to a static VLAN block within a dynamic VLAN pool.

If EPG is enabled for microsegmentation at the template level, when a **Primary MICRO-SEG VLAN** is configured, the **Port Encap VLAN** is configured as an Isolated Secondary VLAN for the Primary VLAN. Traffic is sent from the host to the leaf using the secondary VLAN and return traffic from the leaf to the host is sent using the primary VLAN.

f) (Optional) Select the **Primary MICRO-SEG VLAN**.

The VLAN identifier for microsegmentation

g) (Optional) Select the **Deployment Immediacy**.

Once policies are downloaded to the leaf nodes, deployment immediacy can specify when the policy is pushed into the hardware policy CAM:

- `Immediate`—Specifies that the policy is programmed in the hardware policy CAM as soon as the policy is downloaded in the leaf software.
- `On Demand`—Specifies that the policy is programmed in the hardware policy CAM only when the first packet is received through the data path. This process helps to optimize the hardware space.

h) (Optional) Select the **Mode**.

The mode of the static association with the path. EPG tagging refers to configuring a static path under an EPG:

- `Trunk`—The default deployment mode. Select this mode if the traffic from the host is tagged with a VLAN ID.
- `Access (802.1P)`—Select this mode if the traffic from the host is tagged with a 802.1P tag. When an access port is configured with a single EPG in native 802.1p mode, its packets exit that port untagged. When an access port is configured with multiple EPGs, one in native 802.1p mode, and some with VLAN tags, all packets exiting that access port are tagged VLAN 0 for EPG configured in native 802.1p mode and for all other EPGs packets exit with their respective VLAN tags. Note that only one native 802.1p EPG is allowed per access port.
- `Access (Untagged)`—Select this mode if the traffic from the host is untagged (without VLAN ID). When a leaf switch is configured for an EPG to be untagged, for every port this EPG uses, the packets will exit the switch untagged. Note that when an EPG is deployed as untagged, do not deploy that EPG as tagged on other ports of the same switch.

Step 6 Add one or more **Static Leaf** nodes.

a) Click **+Static Leaf**.
b) From the **Leaf** dropdown, select the leaf node you want to add.
c) (Optional) In the **VLAN** field, provide the VLAN ID for tagged traffic.

Step 7 Add one or more **Domains** nodes.

a) Click **+Domain**.
b) Select the **Domain Association Type**.

This is the type of the domain you are adding:

- `VMM`
- `Fibre Channel`
- `L2 External`
- `L3 External`
- `Physical`

c) Select the **Domain Profile** name.
d) Select the **Deployment Immediacy**.

Deployment immediacy can specify when the policy is pushed:

- `Immediate`—Specifies that the policy is programmed in the hardware policy CAM as soon as the policy is downloaded in the leaf software.
- `On Demand`—Specifies that the policy is programmed in the hardware policy CAM only when the first packet is received through the data path. This process helps to optimize the hardware space.

e) Select the **Resolution Immediacy**.

Specifies whether policies are resolved immediately or when needed. The options are:

- `Immediate`—Specifies that EPG policies are pushed to the leaf switch nodes upon hypervisor attachment to the VMware vSphere Distributed Switch (VDS). LLDP or OpFlex permissions are used to resolve the hypervisor to leaf node attachments.

- `On Demand`—Specifies that EPG policies are pushed to the leaf switch nodes only when a hypervisor is attached to VDS and a VM is placed in the port group (EPG).
- `Pre-provision`—Specifies that EPG policies are pushed to the leaf switch nodes even before a hypervisor is attached to the VDS. The download pre-provisions the configuration on the switch.

Configuring Contracts and Filters

This section describes how to configure a contract, a filter, and assign the filter to the contract. A filter is similar to an Access Control List (ACL), it is used to filter traffic through contracts associated to EPGs.

Procedure

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

You can create the contract in the same or different template as the objects (EPGs and external EPGs) to which you will apply it. If the objects that will use the contract are deployed to different sites, we recommend defining the contract in a template associated to multiple sites. However, this is not strictly required and even if the contract and filters are defined only as local objects in Site1, NDO will create those objects in a remote Site2 when a local EPG or external EPG in Site2 needs to consume or provide that contract.

Step 2 Create a filter.

a) In the main pane, select **+Create Object** > **Filter**.

Alternatively, you can scroll down to the **Filters** area, mouse over the tile, and click **Add Filter**.

b) In the right pane, provide the **Display Name** for the filter.

c) (Optional) Provide a **Description**.

Step 3 Create a filter entry.

a) In the right pane, click **+ Entry**.

The filter entry is a combination of network traffic classification properties. You can specify one or more options as described in the following step.

b) Provide the **Name** for the filter.

c) Choose the **Ether Type**.

For example, `ip`.

d) Choose the **IP Protocol**.

For example, `icmp`.

e) Choose the **Destination Port Range From** and **Destination Port Range To**.

The start and end of the destination ports range. You can define a single port by specifying the same value in both fields or you can define a range of ports from `0` to `65535`. You can also choose to specify one of the server types instead of specific port numbers, for example `http`.

f) Enable **Match only fragments** option.

When enabled, the rule applies to any IP fragment with an offset that is greater than `0` (all IP fragments except the first). When disabled, the rule will not apply to IP fragments with an offset greater than `0` because TCP/UDP port information can only be checked in initial fragments.

g) Enable **Stateful** option.

When this option is enabled, any traffic coming from the provider back to the consumer will always have to have the `ACK` bit set in the packet or else the packets will be dropped.

h) Specify **ARP flag** (Address Resolution Protocol).

The **ARP Flag** is used when creating a specific filter for ARP and allows you to specify ARP request or ARP reply.

i) Choose the **Source Port Range From** and **Source Port Range To**.

The start and end of the source ports range. You can define a single port by specifying the same value in both fields or you can define a range of ports from `0` to `65535`. You can also choose to specify one of the server types instead of specific port numbers, for example `http`.

j) Specify **TCP session rules**.

TCP session rules are used when creating a filter for TCP traffic and allow you to configure `stateful` ACL behavior.

k) Click **Save** to save the filter.

l) Repeat this step to create any additional filter entries for this filter.

You can create and assign multiple filter entries for each filter.

Step 4 Create a contract

a) In the main pane, select **+Create Object** > **Contract**.

Alternatively, you can scroll down to the **Contract** area, mouse over the tile, and click **Add Contract**.

b) In the right pane, provide the **Display Name** for the contract.

c) (Optional) Provide a **Description**.

d) Select the appropriate **Scope** for the contract.

Contract scope limits the contract's accessibility; the contract will not be applied to any consumer EPG outside the scope of the provider EPG:

- `application-profile`
- `vrf`
- `tenant`
- `global`

e) Toggle the **Apply both directions** knob if you want the same filter to apply for both consumer-to-provider and provider-to-consumer directions.

If you enable this option, you will need to provide the filters only once and they will apply for traffic in both directions. If you leave this option disabled, you will need to provide two sets of filter chains, one for each direction.

Note
If you create and deploy a contract with **Apply both directions** enabled, you cannot simply disable the option and re-deploy for the change to apply. To disable this option on an already deployed contract, you must delete the contract, deploy the template, then re-create the contract with the option disabled to correctly change the setting in your fabrics.

f) (Optional) From the **Service Graph** dropdown, select a service graph for this contract.
g) (Optional) From the **QoS Level** dropdown, select a value for this contract.

This value specifies the ACI QoS Level that will be assigned to the traffic using this contract. For more information, see QoS Preservation Across IPN.

If you leave this at `Unspecified`, the default QoS Level 3 is applied to the traffic.

Step 5 Assign the filters to the contract

a) In the right pane, scroll down to the **Filter Chain** area and click **+ Filter** to add a filter to the contract.
b) In the **Add Filter Chain** window that opens, select the filter you added in previous step from the **Name** dropdown menu.
c) Select the **Action** for the filter.

When adding filters, you can choose whether to permit or deny traffic that matches the filter criteria. For `deny` filters, you can set the priority of the filter to one of four levels: `default`, `low`, `medium`, or `high`; the `permit` filters always have the default priority. For more information on ACI contracts and filters, see *Cisco ACI Contract Guide*.

d) Click **Save** to add the filter to the contract.
e) If you disabled the `Apply both directions` option on the contract, repeat this step for the other filter chain.
f) (Optional) You can create and assign multiple Filters to each Contract.

If you want to create additional filter for the same contract:

- Repeat Step 2 and Step 3 to create another filter along with its filter entries.
- Then repeat this step to assign the new filter to this Contract.

Configuring On-Premises External Connectivity

Cisco ACI allows you to establish connectivity to the networks outside your on-premises ACI fabric through the border leaf switches. This connectivity is defined using two constructs, L3Out and External EPG, which provide the configuration options necessary to define security and route maps.

This section describes how to create an L3Out and external EPG in the Nexus Dashboard Orchestrator GUI. The Orchestrator then creates the objects on the APIC site where you deploy the template. Keep in mind that when creating an L3Out from the Orchestrator, only the L3Out container object is created in the APIC and you must still perform the full L3Out configuration (such as nodes, interfaces, routing protocols, and so on) directly in the site's APIC.

While in most cases the L3Out will be created directly at the APIC level and then associated to an external EPG that you create in the Orchestrator, it may be useful to create both here in order to directly associate the L3Out to a VRF which you have created from the 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 8.
- You must have the VRF for the L3Out created as described in Configuring VRFs, on page 10

Procedure

Step 1 Navigate to the Schema and Template you want to edit.

Step 2 Create an L3Out.

a) In the schema edit view, scroll down to the **L3Out** area and click **+** to add a new L3Out.
b) In the properties pane on the right, provide a display name for the L3Out.
c) From the **Virtual Routing & Forwarding** dropdown, select the VRF you created for this.

Step 3 Create an external EPG.

a) In the schema edit view, scroll down to the **External EPG** area and click **+** to add a new External EPG.
b) In the properties pane on the right, select **On-Prem** for the site type.

Step 4 Configure external EPG's basic properties.

a) In the right properties sidebar, provide a display name for the External EPG.
b) From the **Virtual Routing & Forwarding** dropdown, select the VRF you created.

This must be the same VRF that you associated with the L3Out.

c) Click **+Contract** to add a contract for the external EPG to communicate with other EPGs.

If you already have the contracts created, you can assign them now. Otherwise, you can come back to this screen to assign any Contracts you plan to create later.

When assigning Contracts:

- If you are associating a contract with the external EPG as provider, choose contracts only from the tenant associated with the external EPG. Do not choose contracts from other tenants.
- If you are associating the contract to the external EPG as consumer, you can choose any available contract.

Step 5 If configuring an external EPG for an on-premises fabric, set **Site Type** to `on-prem` and configure external EPG's on-premises properties.

a) From the **L3Out** dropdown, select the L3Out you created in a previous step.

Note
You can select the L3Out at the template or site-local level. We recommend configuring the L3Out for the external EPG at the site-local level. To do that, select the template in the left sidebar under the site to which it is assigned. Then select the external EPG and associate an L3Out with it.

b) Click **+Subnet** to add a subnet.

This may be a classification subnet or a subnet used for route-control.

c) In the **Add Subnet** window, provide the subnet prefix.
d) Select the required **Route Control** options.

You can choose one or more for the following options:

- **Export Route Control** enables a route map that allows external prefixes matching the specified subnet prefix to get advertised out of the L3Out. These are the prefixes learned from other L3Outs for transit routing use cases.

  If you are adding the `0.0.0.0/0` subnet and enable the export route control option, **Aggregate Export** option becomes available. This allows you to advertise all the external prefixes learned from other L3Outs. If you

choose to leave this option disabled, only the default route learned from other L3Out will be advertised out of this L3Out.

- **Import Route Control** configures an ingress route map to give you control over what prefixes you want to import from your L3Out into the fabric. The **Import Route Control** is available only when using BGP as the routing protocol on the L3Out.

  If you are adding the `0.0.0.0/0` subnet and enable the import route control option, **Aggregate Import** option becomes available. This works similar to the export route control case except for ingress routes.

- **Shared Route Control** is used for shared L3Out use case and allows prefixes learned from the external router to be advertised to the other VRF that will use this L3Out.

  If you enable the shared route control option, **Aggregate Shared Routes** option becomes available. Again, this functions similar to previous two aggregate routes options but is available for non-`0.0.0.0/0` subnets.

e) Select the **External EPG Classification** options.

You should check the **External Subnets for External EPG** option for the configured subnet(s) to be able to map external entities to this specific External EPG. This allows you to apply a security policy (contract) between those external networks and endpoints belonging to EPG(s) defined inside the fabric. Enabling this flag for a `0.0.0.0/0` prefix ensures that all the external destinations are considered part of this External EPG.

If you enable this option, **Shared Security Import** option becomes available, which allows access from the subnet to the endpoints within the fabric for the inter-VRF (shared services) use case.

For both of these options, access is still subject to contract rules.

Step 6 If configuring an external EPG for a cloud fabric, set **Site Type** to `cloud` and configure external EPG's cloud properties.

a) From the **Application Profile** dropdown, select the application profile.
b) Click **+Add Selector** to add a cloud endpoint selector for the EPG.

Step 7 (Optional) If you want to include this external EPG in the preferred group, check the **Include in preferred group** checkbox.

For more info about EPG Preferred Group, see EPG Preferred Groups Overview and Limitations.

Viewing Schemas

After you have created one or more schemas, they are displayed both on the Dashboard and the Schemas page.

You can use the functionality available on these two pages to monitor the usage and the health of your schemas when they are deployed. You can also access and edit specific areas of the implemented schema policies using the Nexus Dashboard Orchestrator GUI.

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.

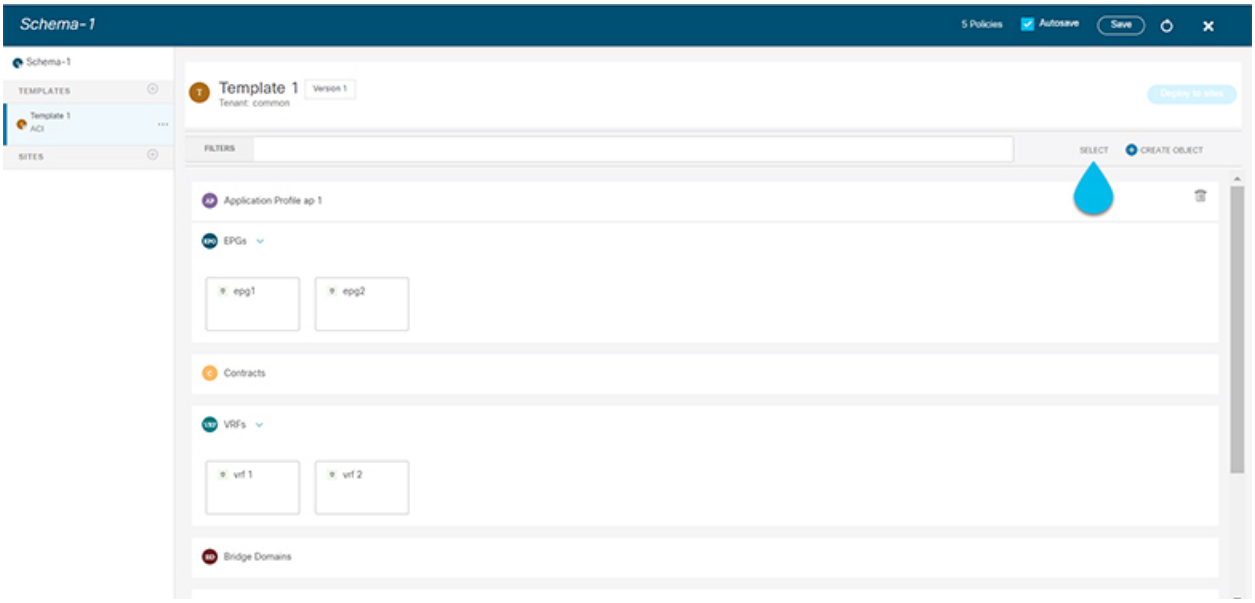
If the selected objects already have different property values configured on them, the update will overwrite those properties with the values you provide. This feature allows you update template-level object properties for on-premises; updating site-local properties and cloud properties are not supported.

Note This feature is supported for Cisco APIC and Cisco DCNM fabrics only; it is not supported for Cisco Cloud APIC sites.

Procedure

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

Step 2 From the upper right pane, choose "Select". It will allow you to choose multiple objects of same type.

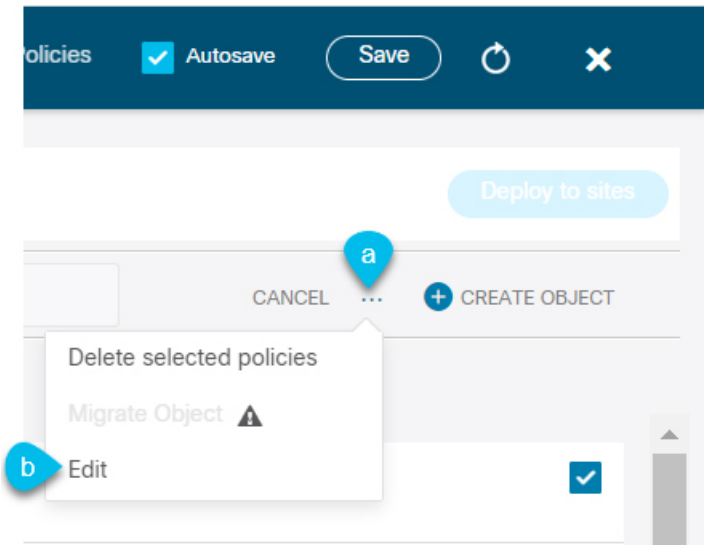


Step 3 After selecting all the objects that you want to update.

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.

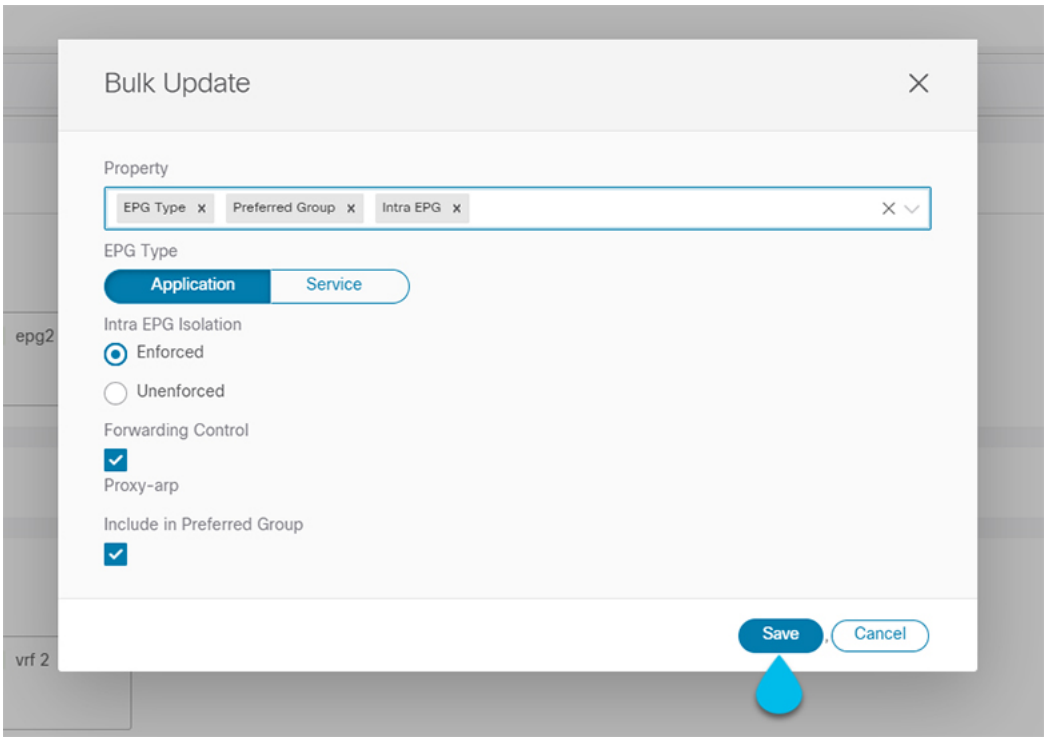
b. **Contracts**: Scope, Filter Chain, QOS Level.

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

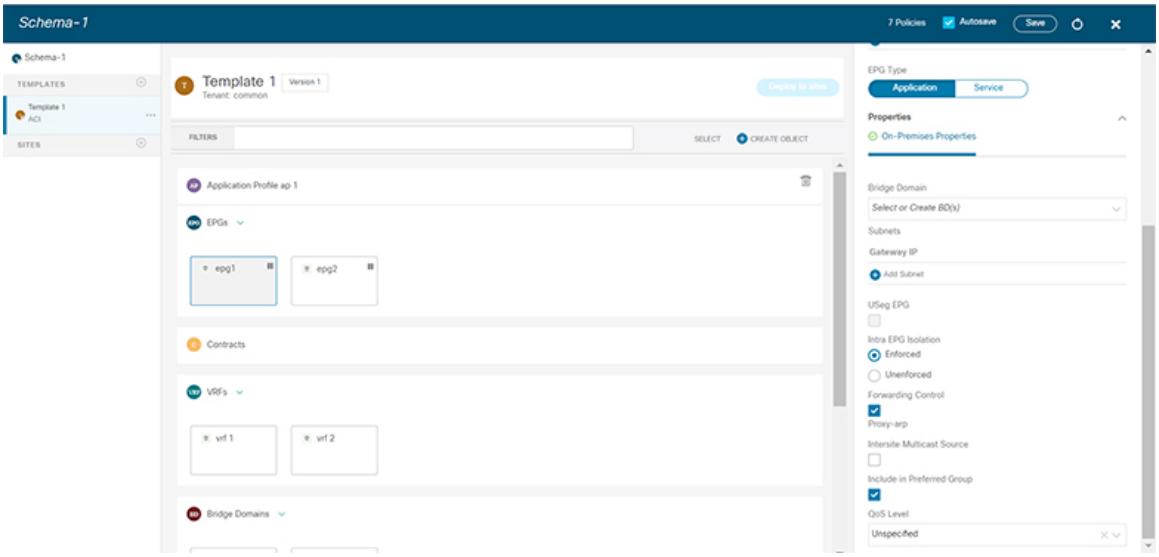
d. **Bridge Domain**: Virtual Routing and Fowrwarding, L2 Stretch, L2 Unknown Unicast, Unkown Multicast Flooding, IPv6 Unkown Multicast Flooding, Multi Destination Flooding, DHCP Policies, Unicast Routing.

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.



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 Click the **Actions** menu (**...**) next to the template name, and choose **Add Sites**.

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

Step 4 In the **Add 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 **Save**.

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

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.
- When initially upgrading from an earlier release that supported unlimited number of versions per template, all existing versions remain intact.

  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.

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 **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.

Procedure

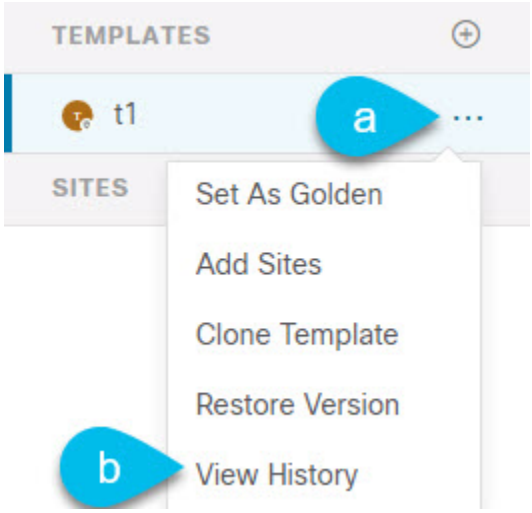
Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

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

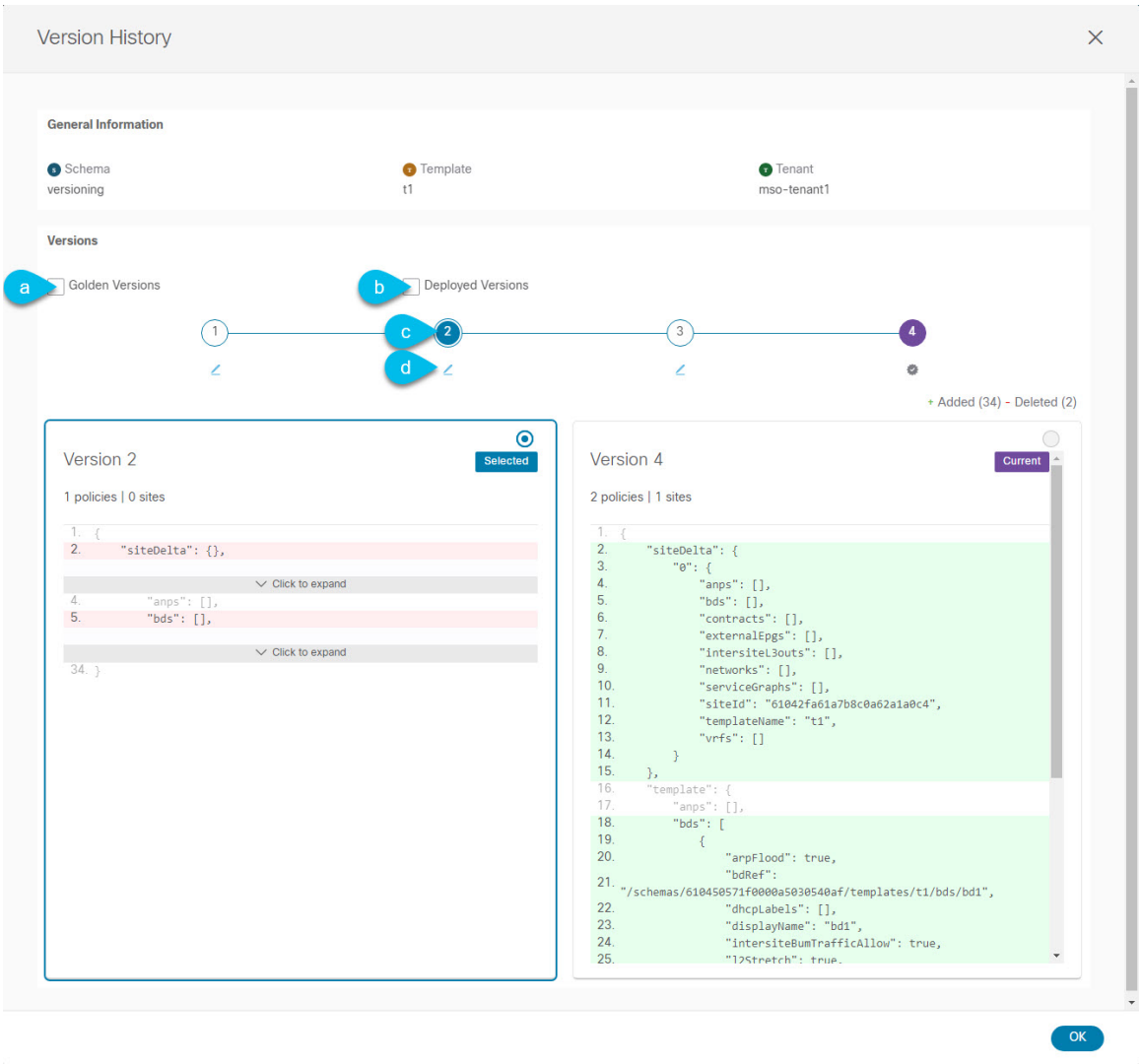
Step 3 Click the schema that contains the template you want to view.

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

Step 5 From the template's actions (**...**) menu, select **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 29.

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 33 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.

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, 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.

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 **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 33.

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 33.

- Created or updated users in Nexus Dashboard with `approver` and `deployer` roles, as described in Create Users with Required Roles, on page 34.
- 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 33.
- Created or updated users in Nexus Dashboard with `approver` and `deployer` roles, as described in Create Users with Required Roles, on page 34.

- 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 34.

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 29.

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 51.

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, 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 33.

Procedure

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

Step 2 In the left sidebar, select the template you want to deploy.

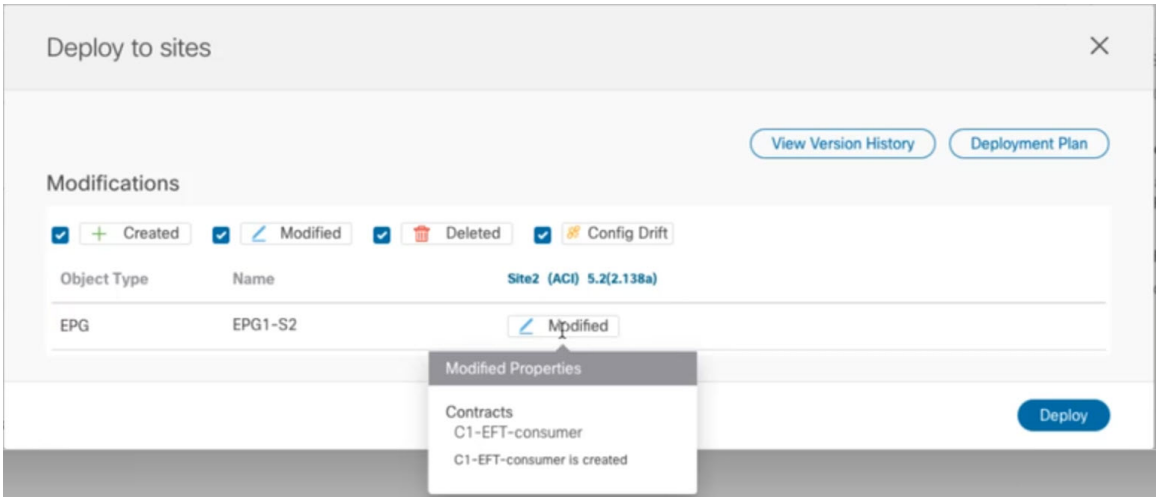
Step 3 In the top right of the template edit view, click **Deploy**.

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`.



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**, which 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 29.
- 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 Multi-Site 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.

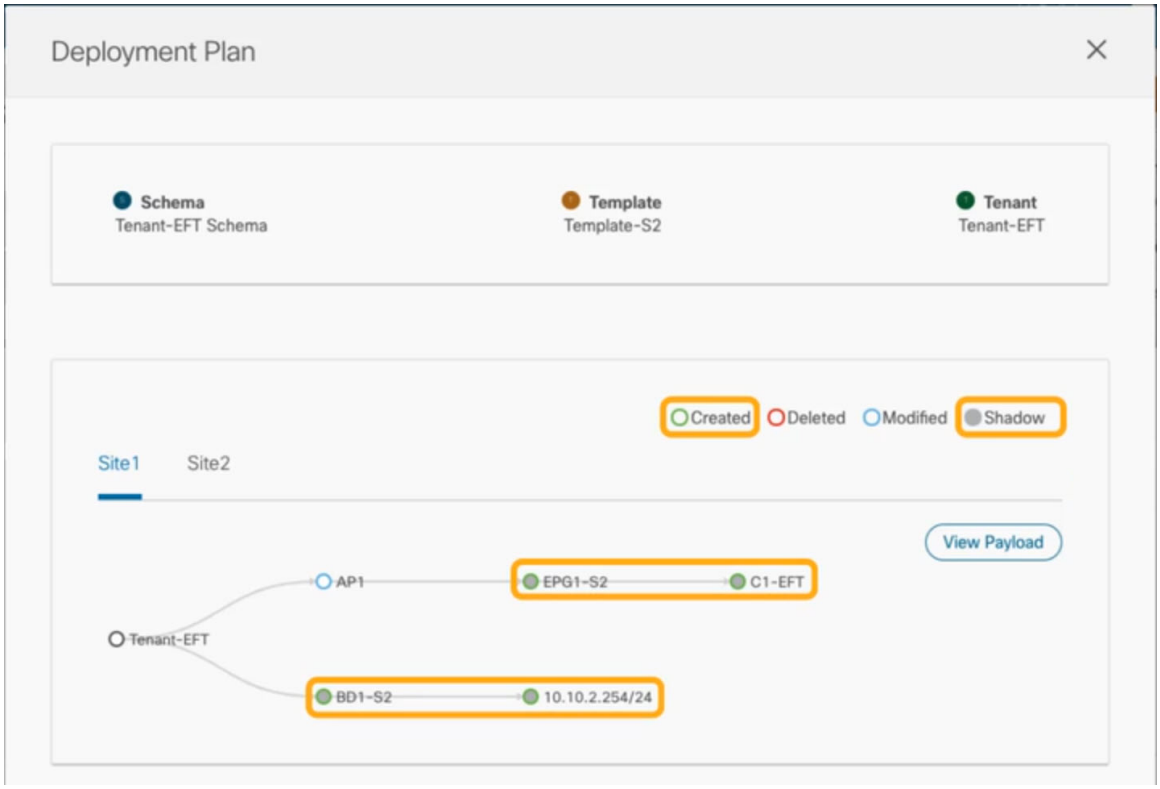
  Note
  We recommend verifying your changes using the Deployment Plan as described in this step before deploying the template. The visual representation of the configuration changes can help you reduce potential errors from deploying unintended configuration changes.

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`.

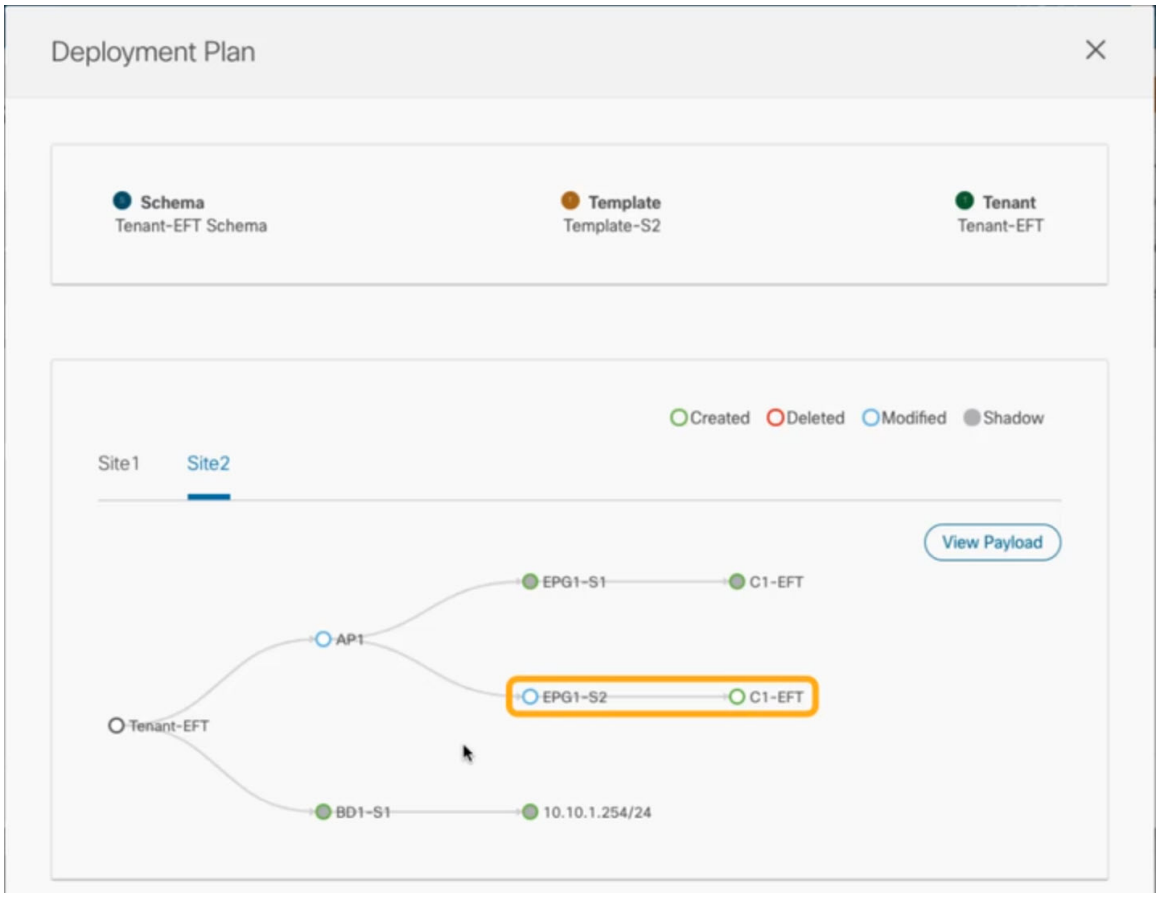
b) Verify your changes in the first listed site.

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`.



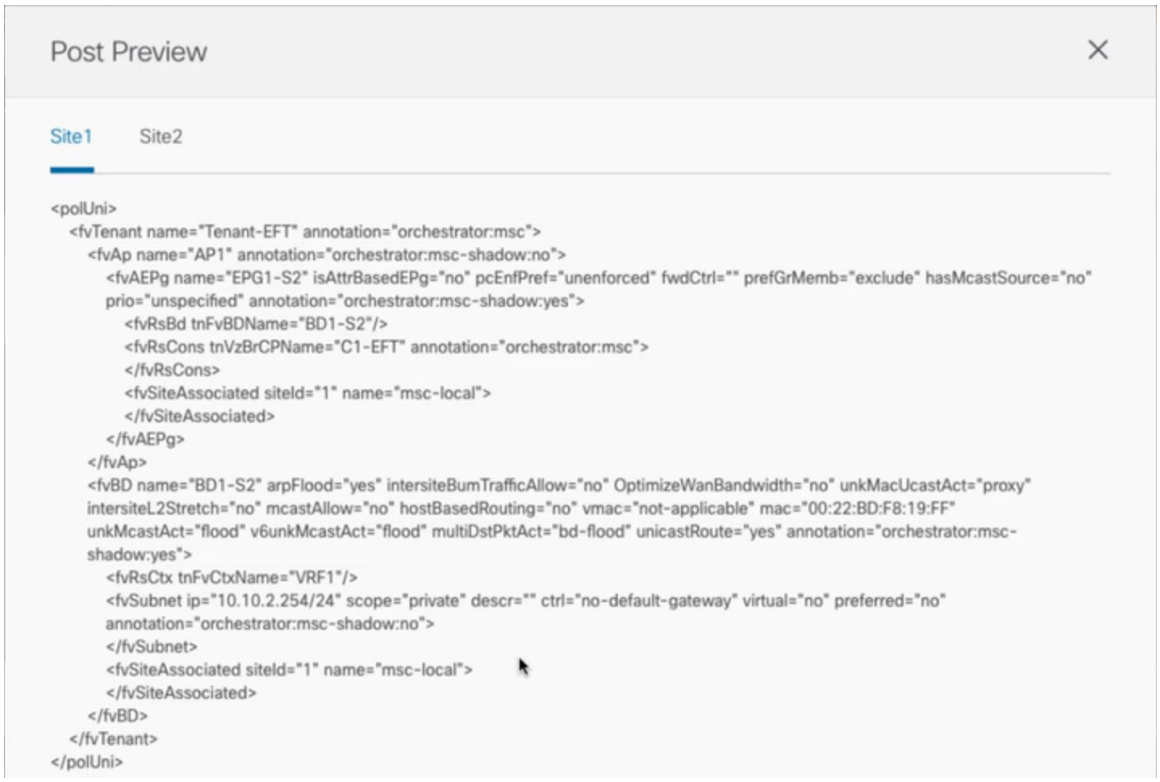
c) Repeat the previous substep to verify the changes in other sites

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.

Before you begin

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

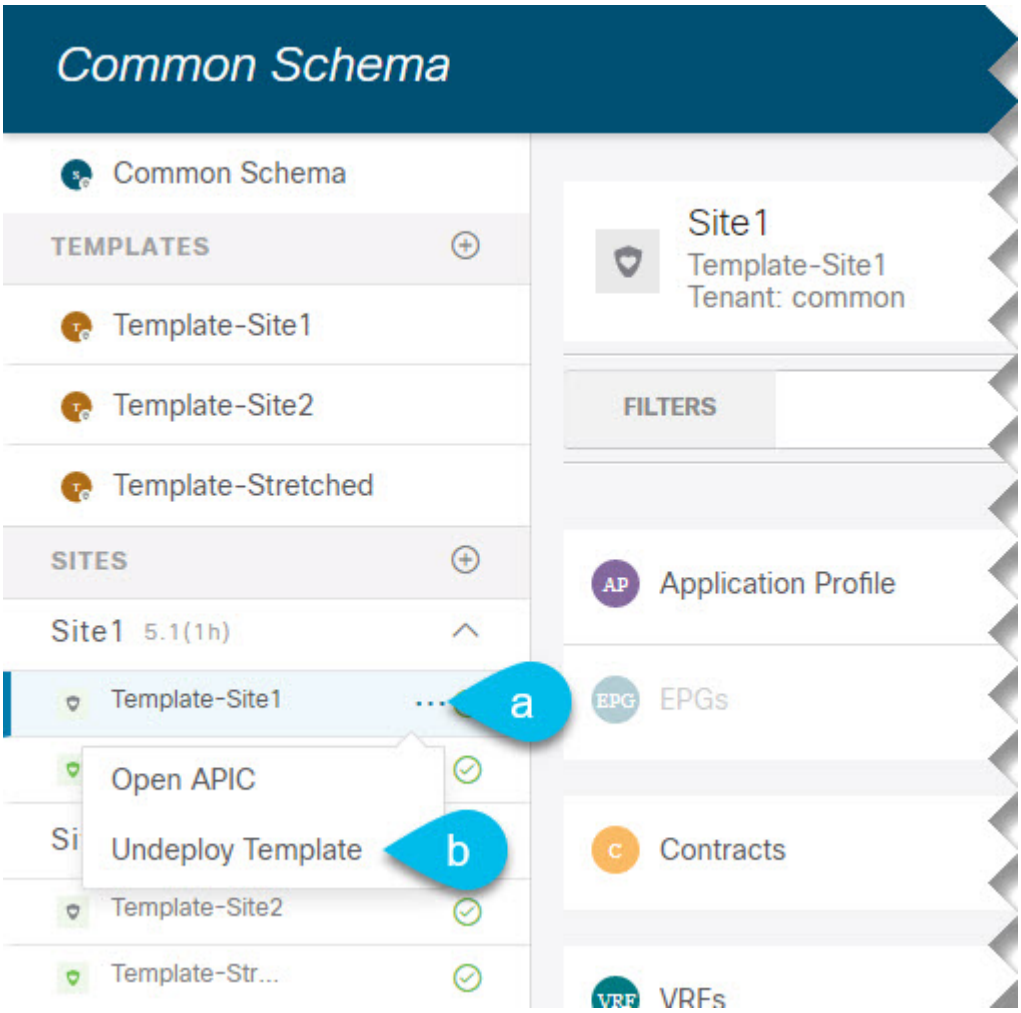
  Undeploying a template that was modified since it was last deployed may create a configuration drift because the set of objects deployed with the template would be different than the set of objects you try to undeploy after making changes to the template.

Procedure

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

Step 2 In the left sidebar under **SITES**, select the template you want to undeploy.

Step 3 Undeploy the template.



a) Click the **More options** (. . .) menu next to the template.
b) Click **Undeploy Template**.

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.

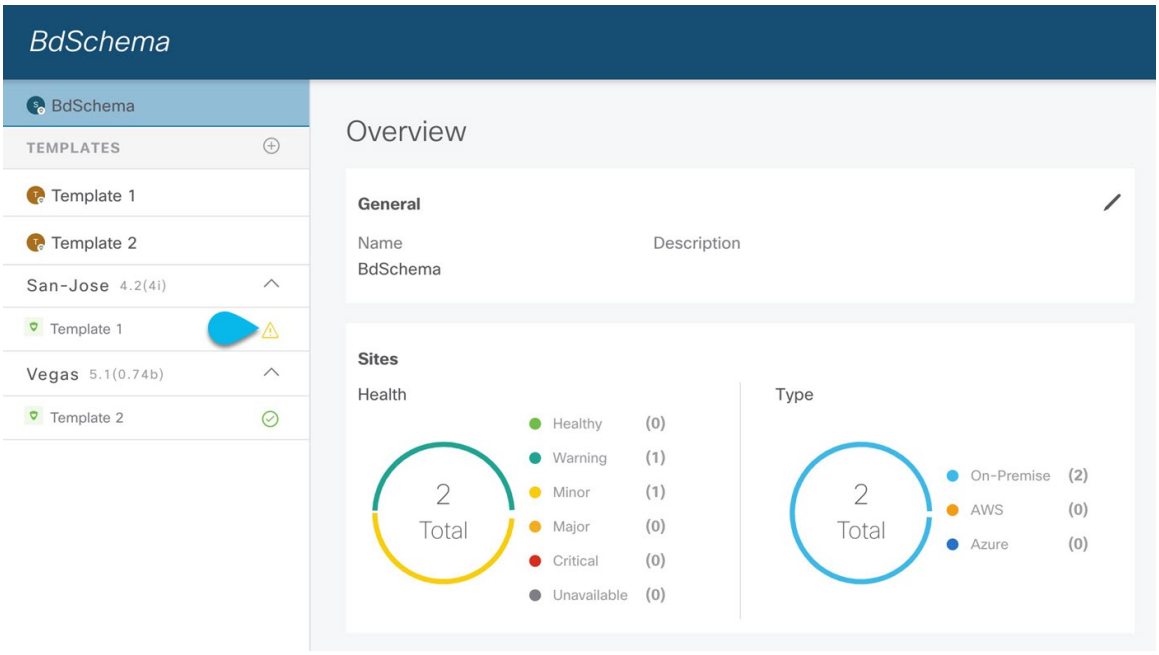
Step 4 In the Schema view, select the template under the specific site from which you want to disassociate it.

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

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

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. These configuration discrepancies are referred to as **Configuration Drifts** and are indicated by a yellow warning sign next to the template name in the schema view 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, on page 44.
- When you deploy a template from NDO, drift notification for objects in that template is disabled for 60 seconds.

  For fabric policies, NDO does not detect configuration drifts for policies for a certain period of time after the deployment. Any notification that is received during this timeout window will not be processed and the configuration drift will not be reported. You can calculate the length of this timeout window using the following formula:

  ```
  15 seconds + (600ms * number of policies in the template)
  ```
- In releases prior to 3.7(2g), you could assign a non-existent DHCP policy to a bridge domain and save the schema; this has been changed to allow only DHCP policies that already exist in NDO to be assigned to BDs. As a result, you may be unable to reconcile a configuration drift between NDO and APIC for a BD that has a DHCP policy label assigned to it in the APIC if that policy does not exist on NDO. Since importing DHCP labels is not supported in this release, you must first create the DHCP policy in NDO with the same name as it exists in the APIC and only then go through the drift reconciliation workflow. This allows you to keep certain DHCP policies only in the APIC and reconcile remaining configuration by creating only the policies you want to import in NDO.

  This NDO release supports only `tenant` scope DHCP policies, while APIC supports `infra` scope as well. So the drift reconciliation workflow will not import any DCHP policies with `infra` scope.

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.

  Starting from NDO release 3.6(1) it is recommended to run the "Reconcile Drift" workflow for the template, 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. For more information on the new drift reconciliation workflow in NDO 3.6, please refer to the "Reconciling Configuration Drifts" section below.

Reconciling Configuration Drifts

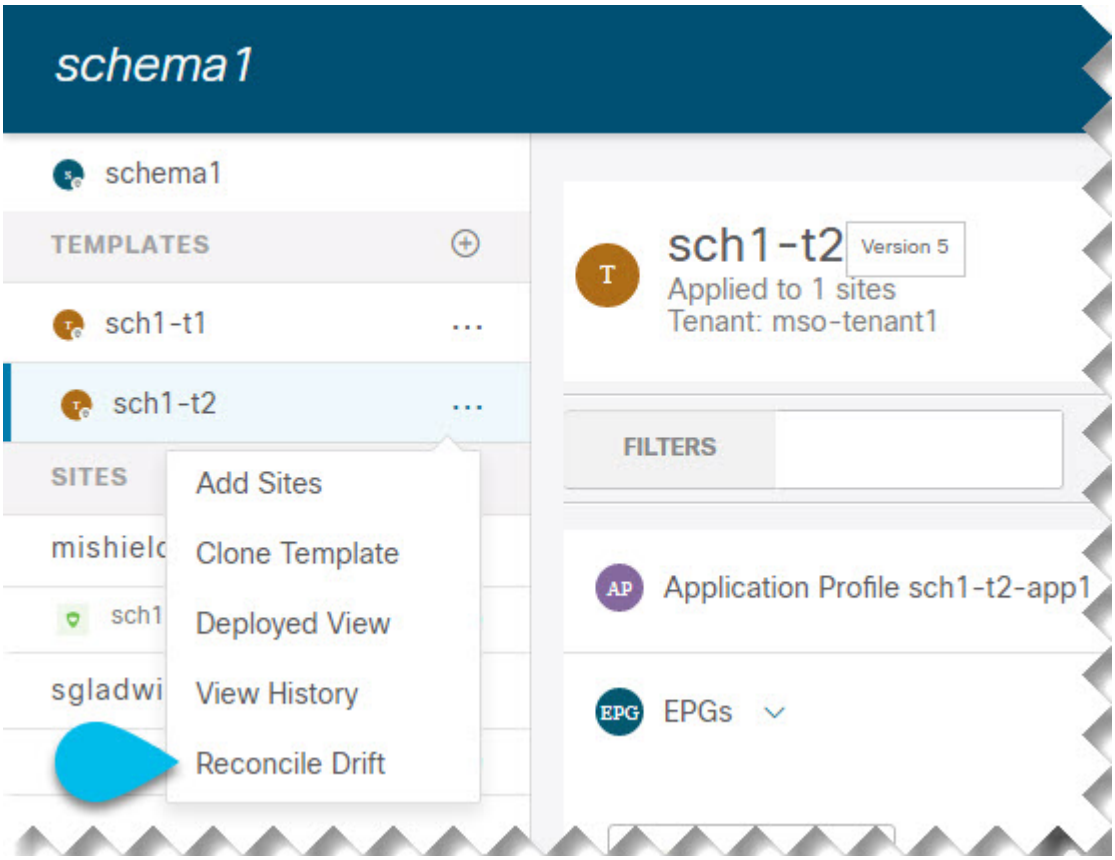
NDO release 3.6(1) introduces support for a drift reconciliation workflow that can be run to compare the template configuration defined on Nexus Dashboard Orchestrator and the configuration rendered in the APIC controllers of the sites part of the Multi-Site domain. This allows to provide more visibility into what causes the configuration drift (i.e. changes that have been made on Nexus Dashbaord Orchestrator or on APIC directly) and give the user the choice on how to reconcile the drift, as described in the steps below.

Note If you do not want the configurations you chose, you can close the schema and re-open. This will show the original configurations. You can re-trigger "Reconcile Drift" flow again if needed. The schema will get saved only after you choose Save or Deploy button.

Procedure

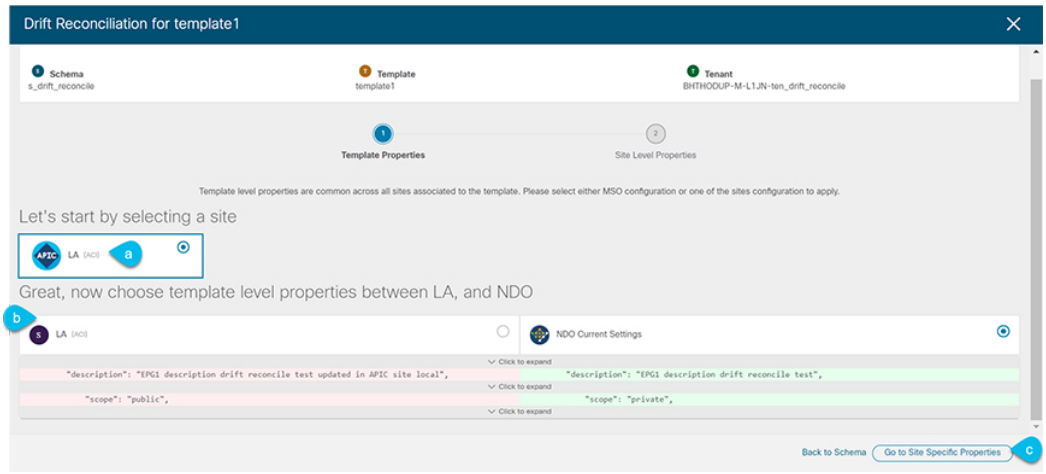
Step 1 Navigate to the schema that contains the template you want to check for configuration drifts.

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



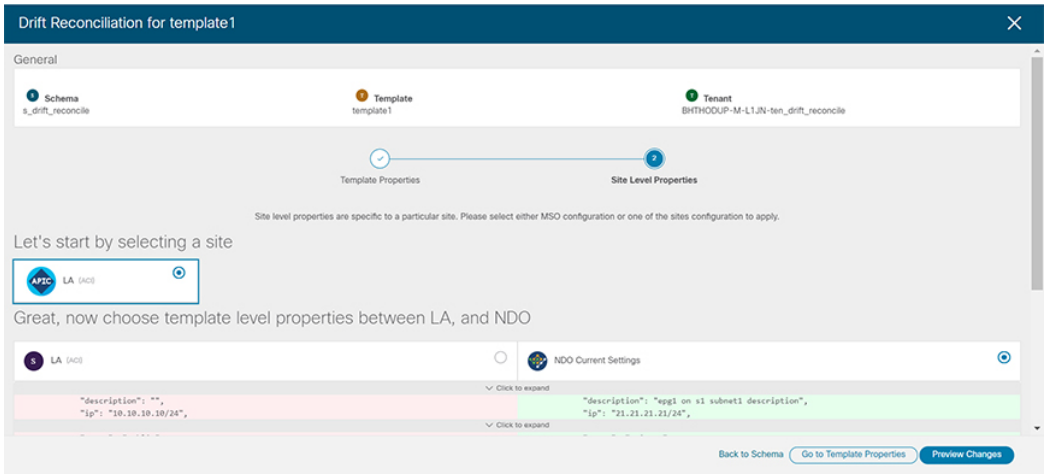
The **Drift Reconciliation** wizard opens.

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 Schemas

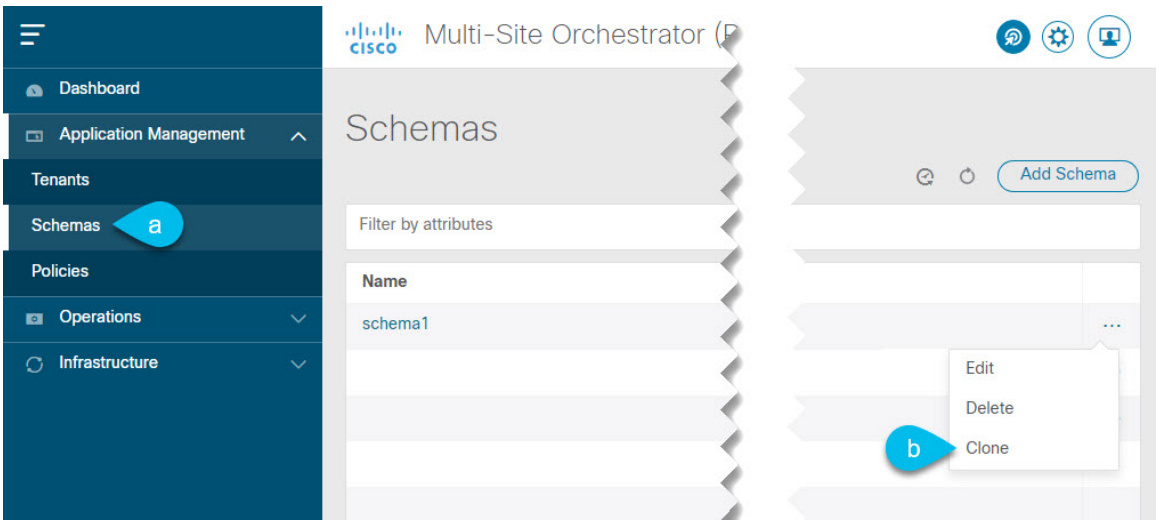
This section describes how to create a copy of an existing schema and all its templates using the "Clone Schema" feature in the **Schemas** screen.

Note If you clone a template and attempt to deploy it to the same site with a different configuration, the deployment may fail due to a duplicate name error. Changing the object name in the cloned template only updates the display name. It does not alter the database record, which causes the deployment to fail in this scenario.

Procedure

Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

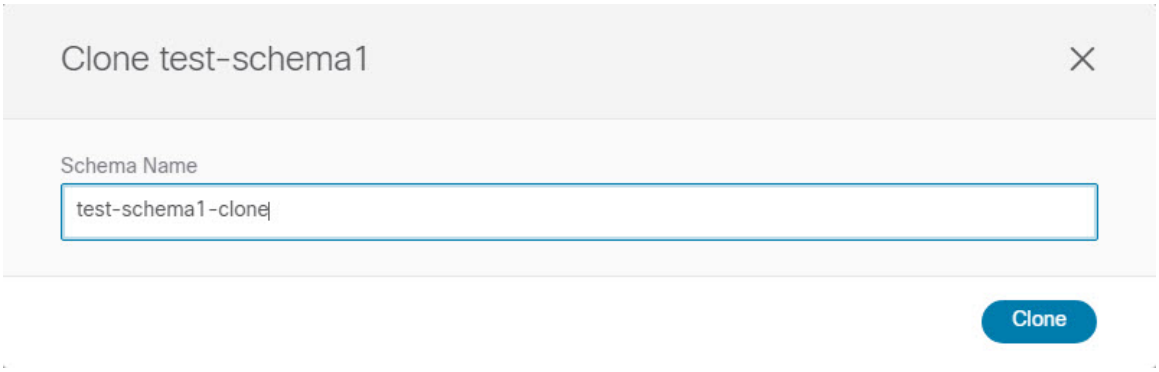
Step 2 Choose the schema to clone.



a) From the left navigation menu, select **Application Management** > **Schemas**.

b) From the **Actions** menu next to the name of the schema you want to clone, select **Clone Schema.**

Step 3 Provide the name for the new schema and click **Clone**.



After you click **Clone**, the UI will display `Cloning of <schema-name> was successful.` message and the new schema will be listed in the **Schemas** screen:

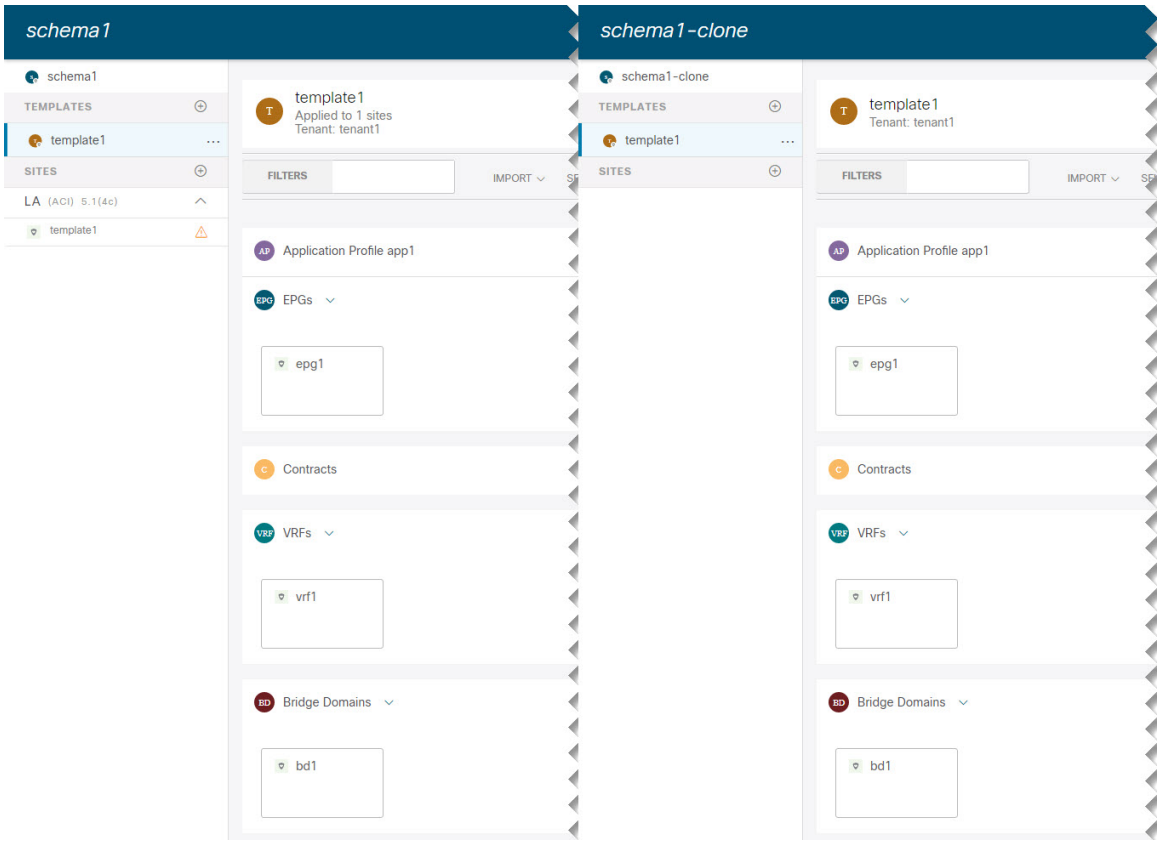


The new schema is created with the exact same templates (and their tenants' association), object, and policy configurations as the original schema.

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

Step 4 (Optional) Verify that the schema and all its templates were copied.

You can verify the operation completed successfully by comparing the two schemas:



Cloning Templates

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

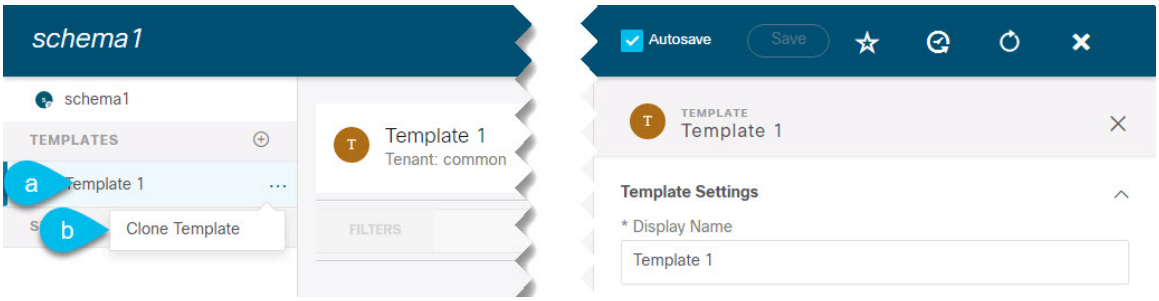
Procedure

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 In the Schema view, open the **Clone Template** dialog.



a) Select the Template you want to clone.

b) From the **Actions** menu, select **Clone Template**.

Step 5 Provide the clone destination details.



a) From the **Destination Tenant** dropdown, select the target tenant.

By default, the current template's tenant is selected. If you change the tenant, the new template will be assigned to the tenant you select instead.

Note that the destination tenant must already exist. If you want to clone the template and assign it to a new tenant, you must first create it in the **Tenants** page, then come back to the template to clone it.

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

b) 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.

c) In the **Cloned Template Name** field, provide the name for the new template.
d) 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 APIC 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

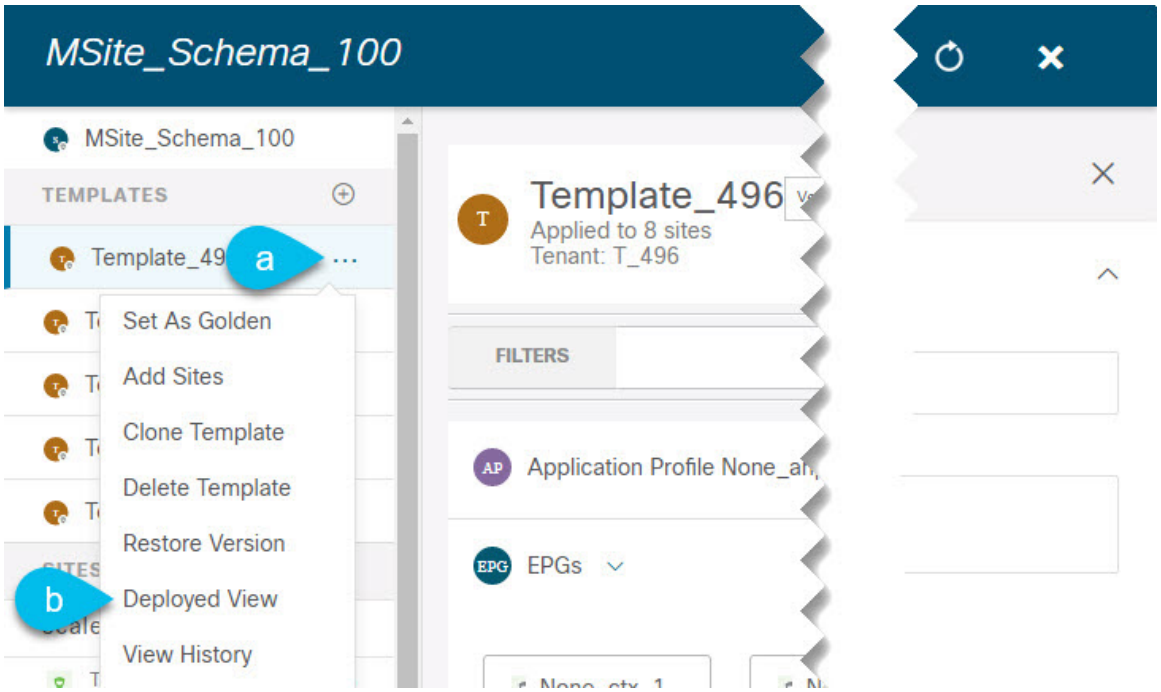
Step 1 Log in to your Nexus Dashboard Orchestrator GUI.

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.

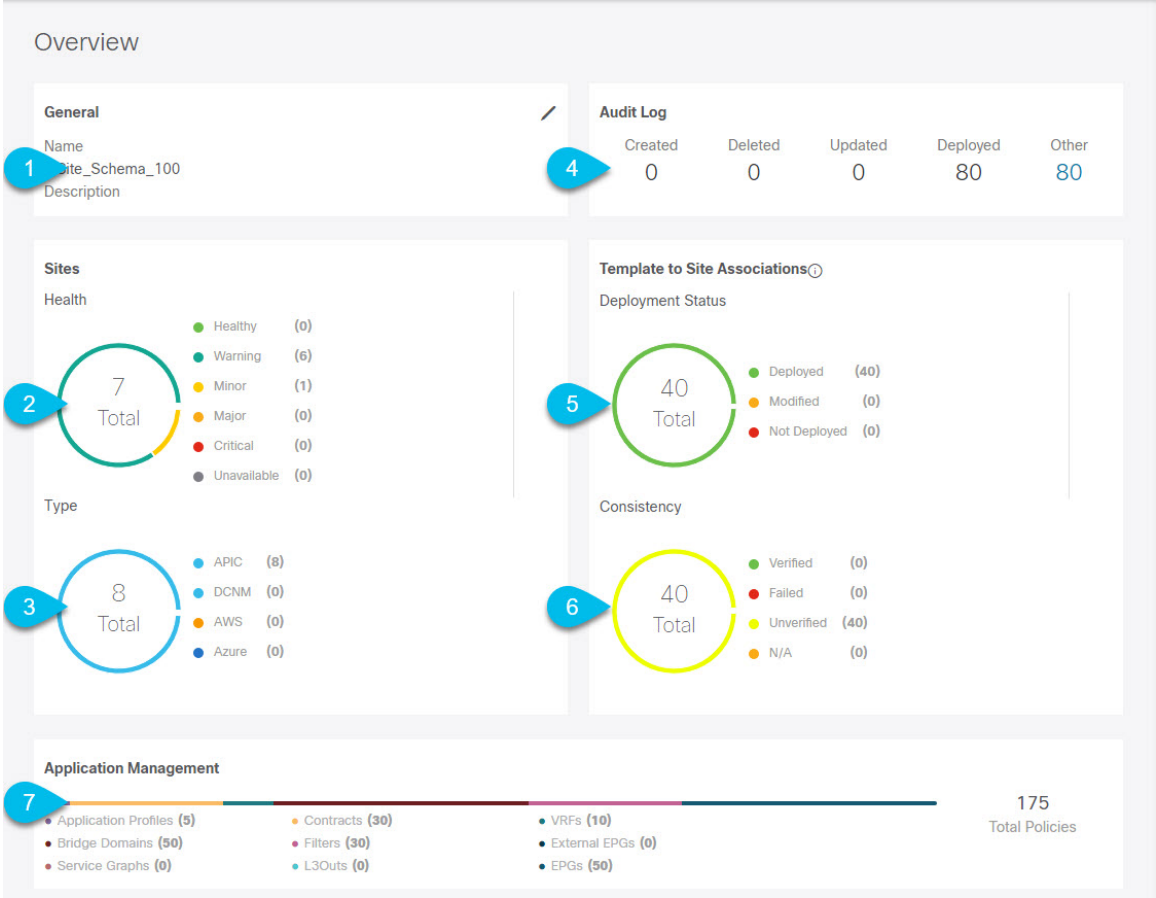
   If the latest version of the template is already deployed, the view will not contain any color-coded objects and will simply display the currently deployed configuration.

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 4: Schema Overview

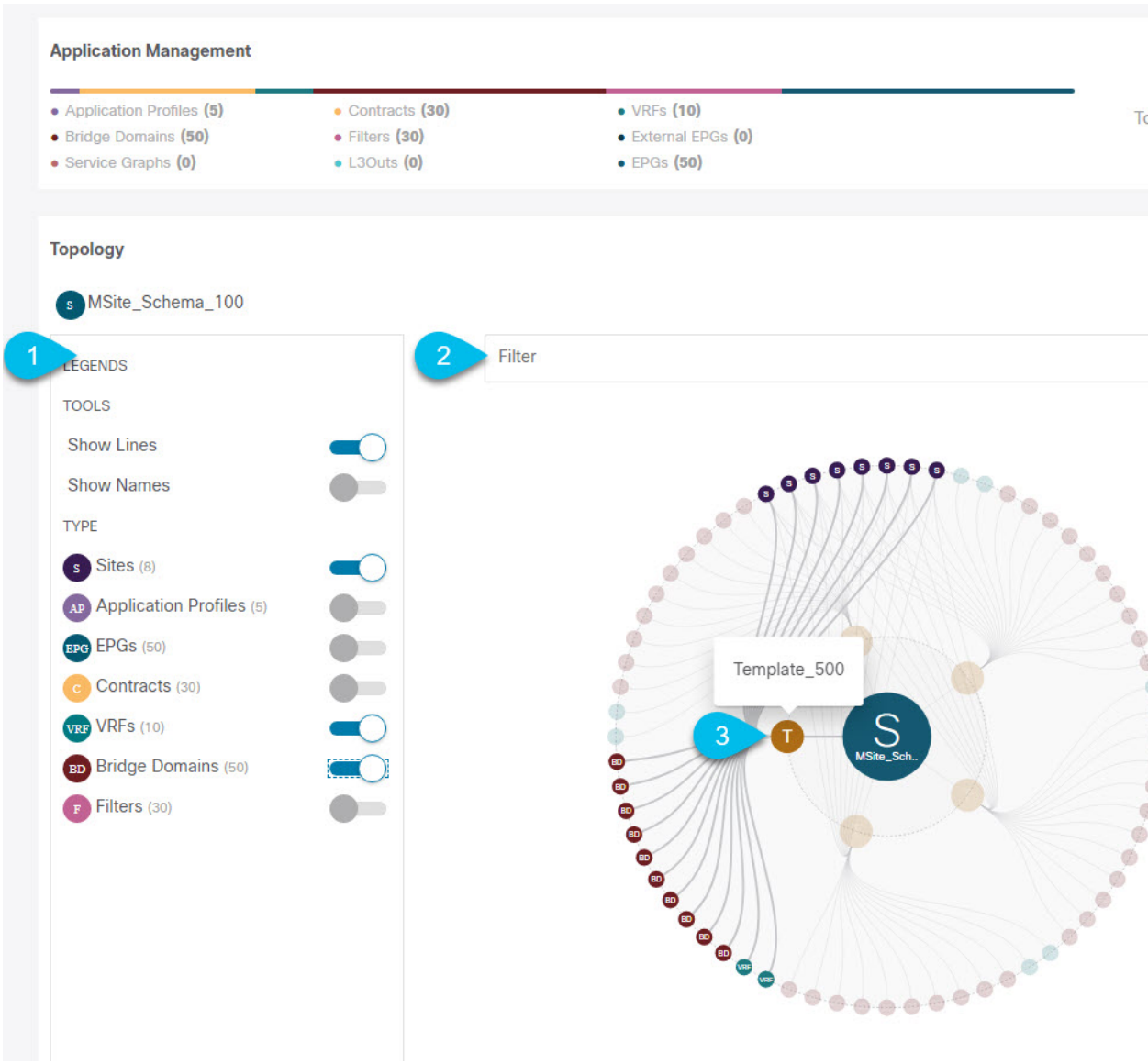


The following details are provided on this page:

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 5: 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.

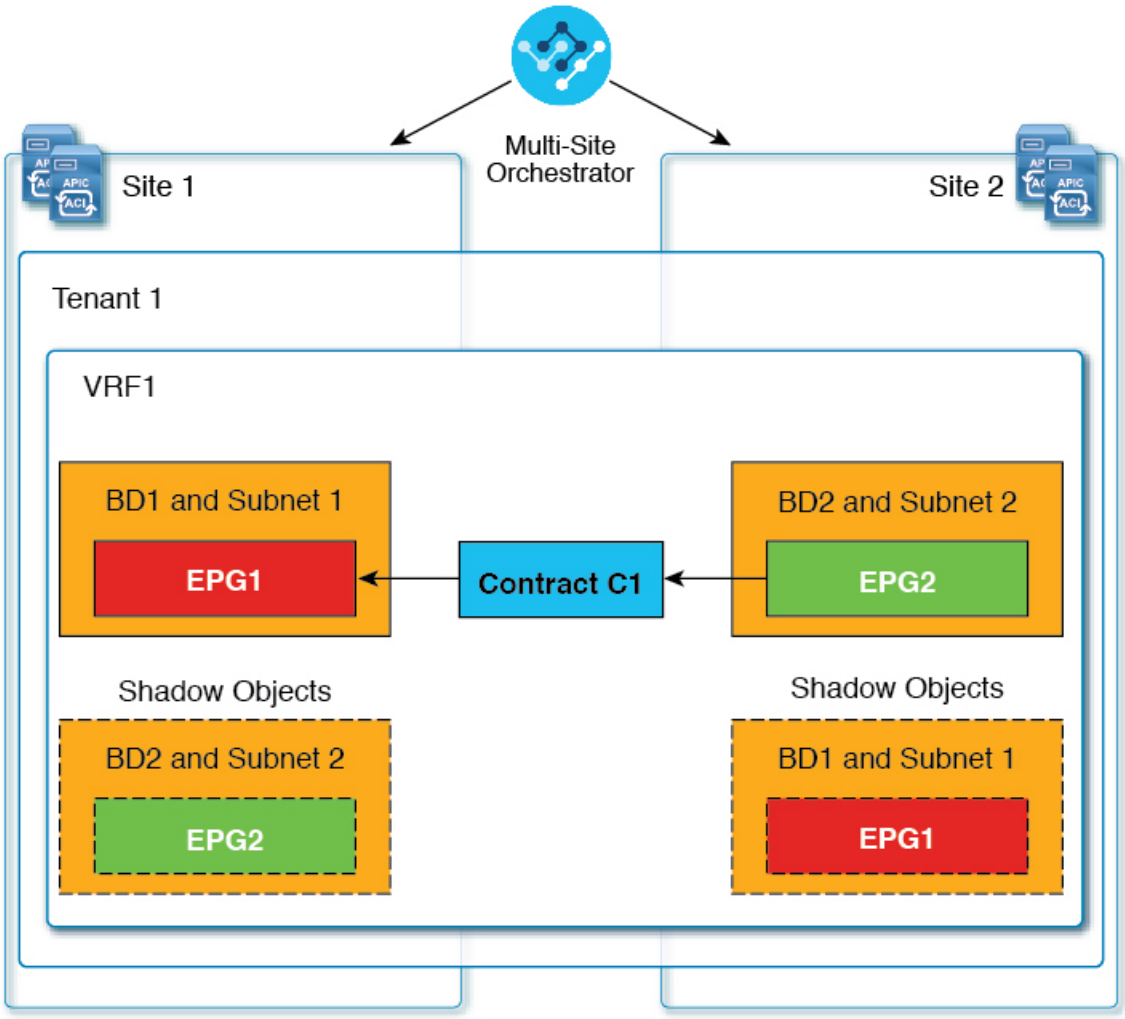
Shadow Objects

When a contract exists between site-local EPGs in stretched VRF or in Shared Services use-cases where provider and consumer are in different VRFs and communicate through Tenant contracts, the EPGs and bridge domains (BDs) are mirrored on the remote sites. The mirrored objects appear as if they are deployed in each of these sites' APICs, while only actually being deployed in one of the sites. These mirrored objects are called "shadow" objects.

Note Shadow objects should not be removed using the APIC GUI.

For example, if a tenant and VRF are stretched between Site1 and Site2, provider EPG and its bridge domain are deployed in Site2 only, and consumer EPG and its domain are deployed in Site1 only, then corresponding shadow bridge domains and EPGs will be deployed as shown in the figure below. They appear with the same names as the ones that were deployed directly to each site.

Figure 6: Basic Shadow EPG



The following objects can be shadowed:

- VRFs
- Bridge Domains (BDs)
- L3Outs
- External EPGs
- Application Profiles
- Application EPGs
- Contracts (Hybrid Cloud deployments)

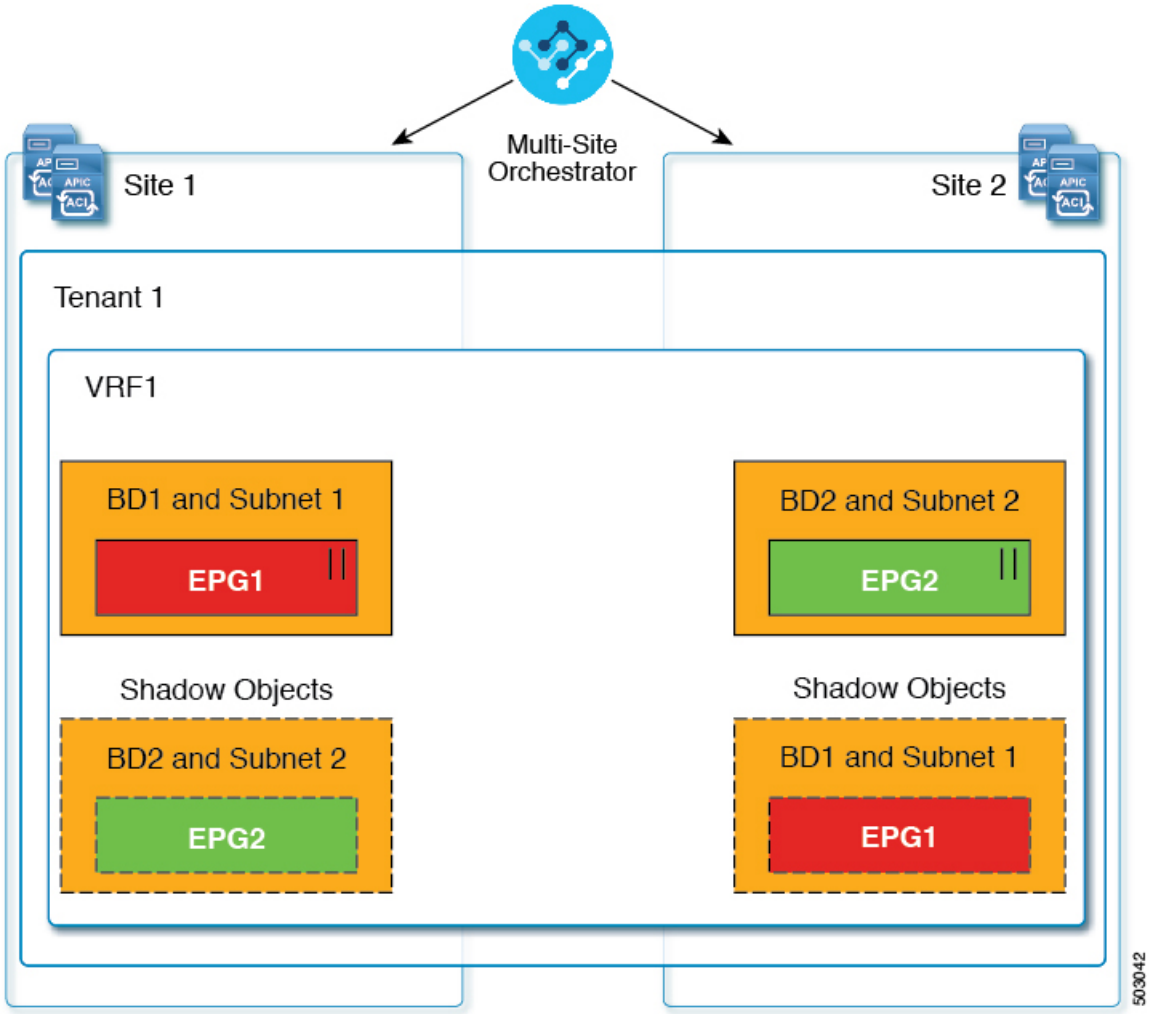
If your fabrics are running APIC Release 5.0(2) or later, when you select a shadow object in the APIC GUI, you will see a `This is a shadow object pushed by MSC to support intersite policies. Do not make any changes or delete this object.` warning at the top of main GUI pane. In addition, shadow

EPGs that are not part of a VMM domain will not have static ports, while shadow BDs will have **No Default SVI Gateway** option enabled in the APIC GUI.

Other Use Cases with Shadow Objects

Shadow objects are also create in a number of other use cases, such as Preferred Group, vzAny, and Layer 3 Multicast, and hybrid cloud, as shown in the figures below.

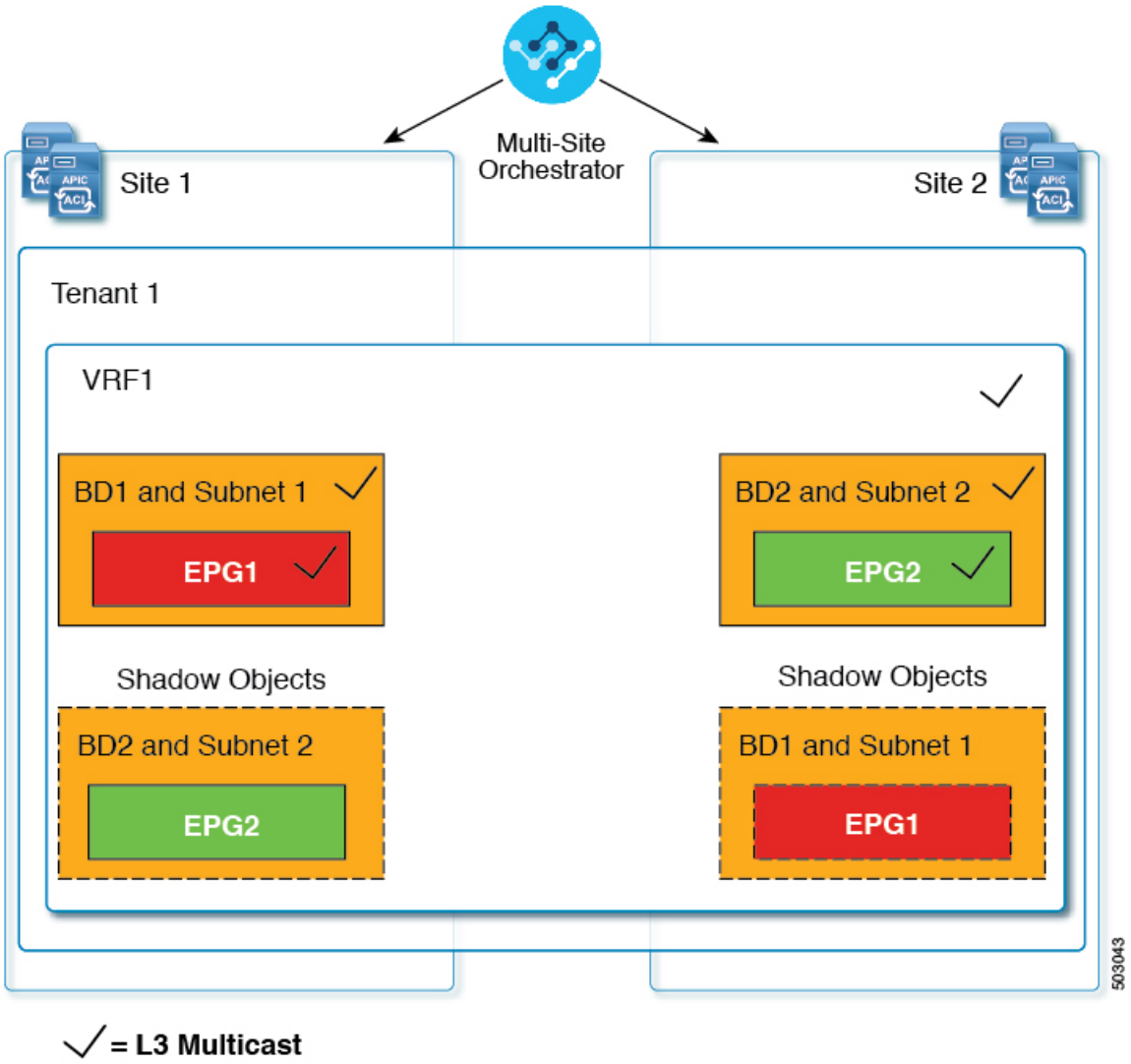
Figure 7: Preferred Group



|| = Preferred Group

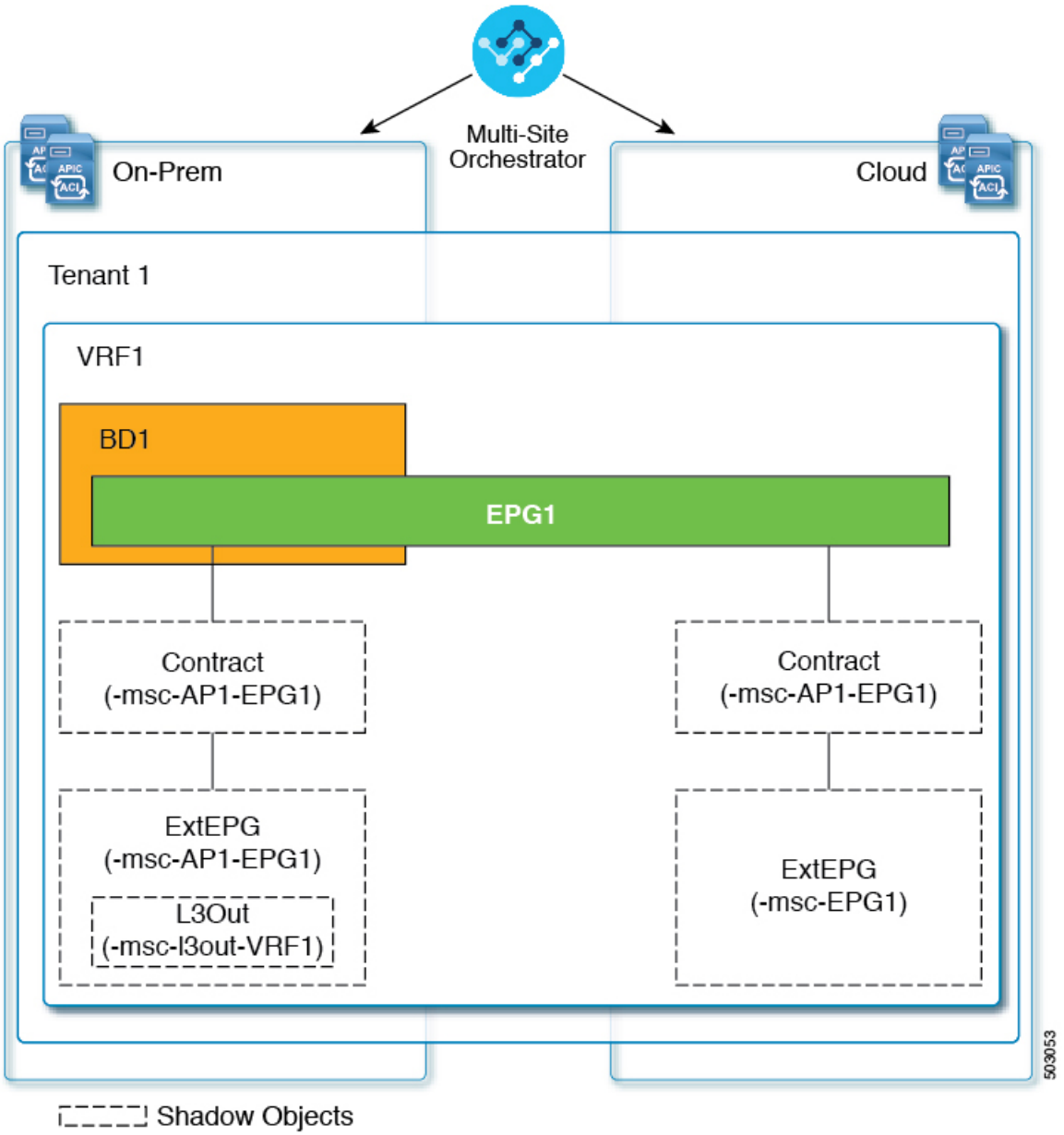
In case of multicast, the shadow objects are created only for EPGs/BDs that have multicast sources connected and the option explicitly configured at the EPG level.

Figure 8: L3 Multicast



In case of hybrid cloud deployments, even stretched objects will create shadow objects where implicit contracts exist. For example in the following case where an EPG is stretched between an on-premises and cloud sites, shadow external EPGs are created in each site with implicit shadow contracts between the stretched EPG and the shadow external EPGs.

Figure 9: Hybrid Cloud



Starting with Cisco APIC, Release 5.2(3), shadow objects are indicated by a unique icon in the Cisco APIC GUI. Regular Orchestrator-created objects are shown with a green cloud symbol, whereas the shadow objects will have a gray cloud icon.

Hiding Shadow Objects in APIC GUI

Starting with APIC Release 5.0(2), you can choose to show or hide the shadow objects created by the Nexus Dashboard Orchestrator in the on-premises site's APIC GUI. Shadow objects in Cloud APIC are always hidden.

If you want to hide shadow objects from the GUI, keep the following in mind:

- This option cannot be set globally from the Orchestrator and must be set directly in each site's APIC as described in this section.
- The option to show shadow objects is turned off by default for all new APIC Release 5.0(2) installations and upgrades, so previously visible objects may become hidden.
- Hiding shadow objects relies on a flag set by the Nexus Dashboard Orchestrator specifically for this feature, which is enabled from Orchestrator Release 3.0(2) and later:
  - If shadow objects are deployed by an earlier Orchestrator version, they will not have the required tag and will always be visible in the APIC GUI.
  - If shadow objects are deployed by Orchestrator version 3.0(2) or later, they will have the tag and can be hidden or shown using the APIC GUI setting.
  - We recommend upgrading each fabric to APIC Release 5.0(2) before upgrading the Nexus Dashboard Orchestrator.

    When the Nexus Dashboard Orchestrator is upgraded to Release 3.0(2), any objects deployed to sites running APIC Release 5.0(2) or later will be tagged with appropriate tags and can be shown or hidden using the APIC GUI without having to re-deploy them.

    If you upgrade the Orchestrator before the fabric's APIC, the site's objects will not be tagged and you will need to manually re-deploy the configuration after the fabric is upgraded for the flag to be set.
- If you ever downgrade your fabric to a release prior to Release 5.0(2), the shadow objects will no longer be hidden and you may see a different icon for them in the APIC GUI.



Procedure

Step 1 Log in to the site's APIC.

Step 2 In the top right corner, click the **Manage my profile** icon and choose **Settings**.

Step 3 In the **Application Settings** window, enable or disable the **Show Hidden Policies** checkbox.

The setting is stored in the user profile and is enable or disabled separately for each user.

Step 4 Repeat the process for any additional APIC sites.