

Schemas

- Schema Design Considerations, on page 1
- Concurrent Configuration Updates, on page 5
- Creating Schemas and Templates, on page 6
- Deploying Templates, on page 23
- Cloning Schemas, on page 25
- Cloning Templates, on page 27
- Migrating Objects Between Templates, on page 28
- Template Deployment View (Beta), on page 29
- Schema Overview and Deployment Visualizer, on page 31
- Shadow Objects, on page 34
- Configuration Drifts, on page 40

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 adds all VRFs, Bridge Domains, EPGs, Contracts and other elements to that template. You can then create a single application profile or multiple application profiles within the template and deploy it to one or more sites.

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

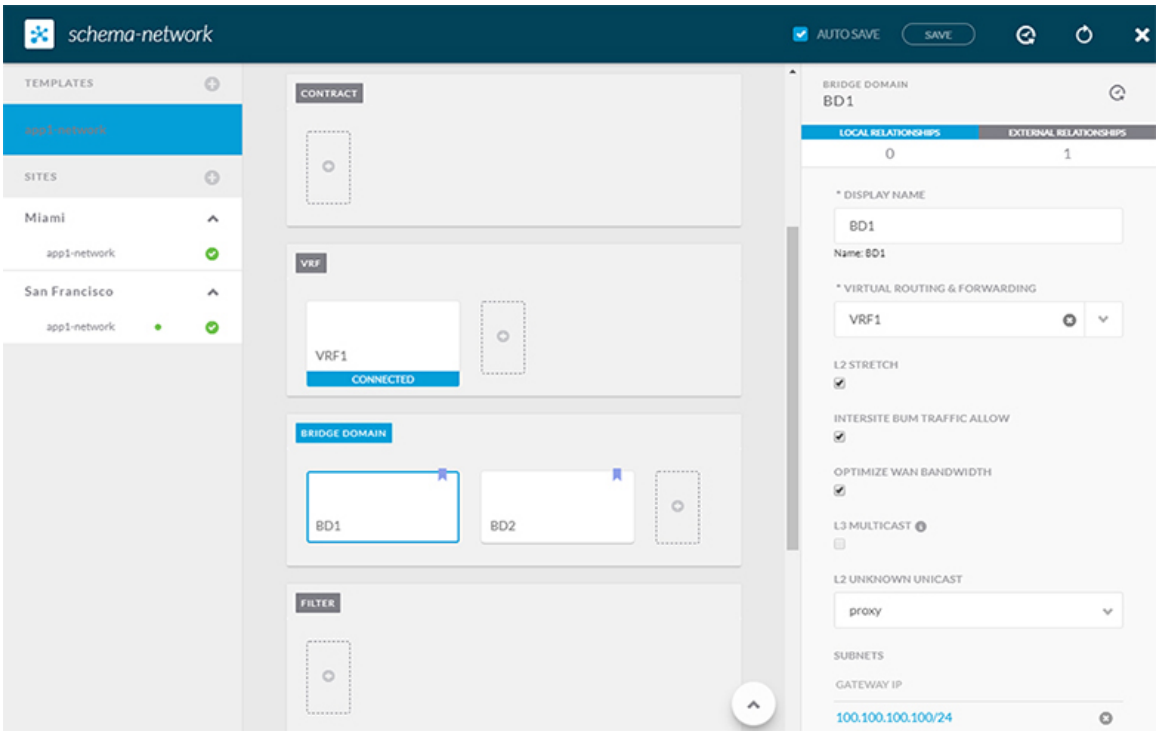
Multiple Schemas with Network Separation

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.

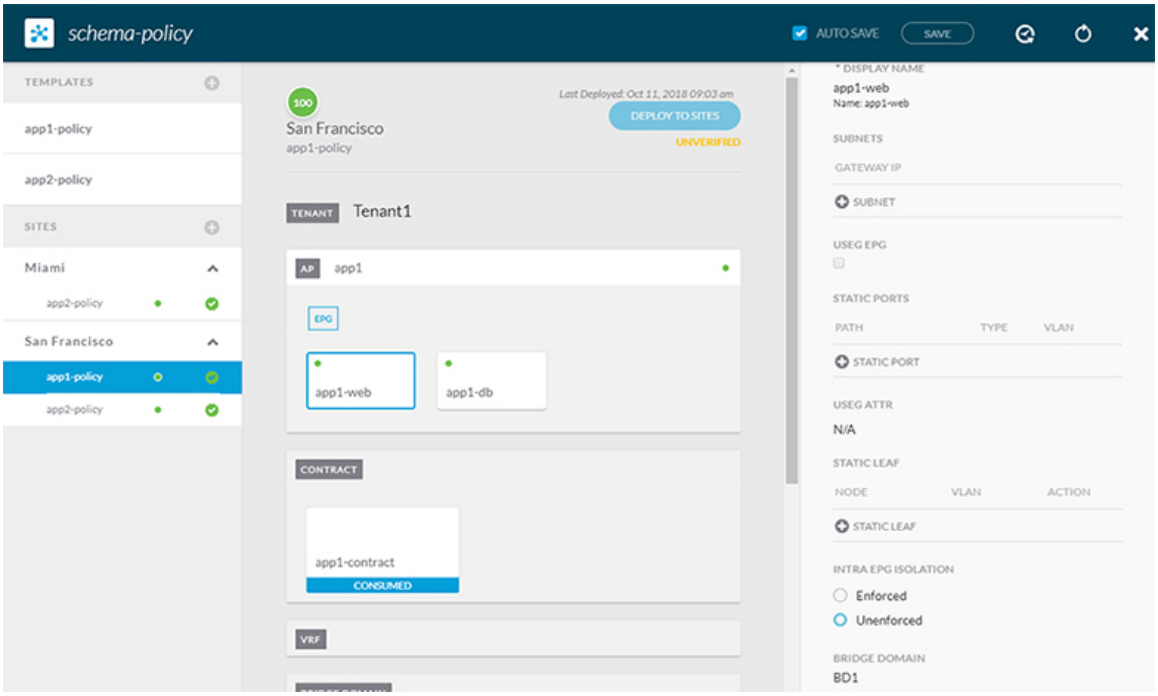
The following figure shows a single networking template configuration with VRF, BD, and subnets configured and deployed to two sites:

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



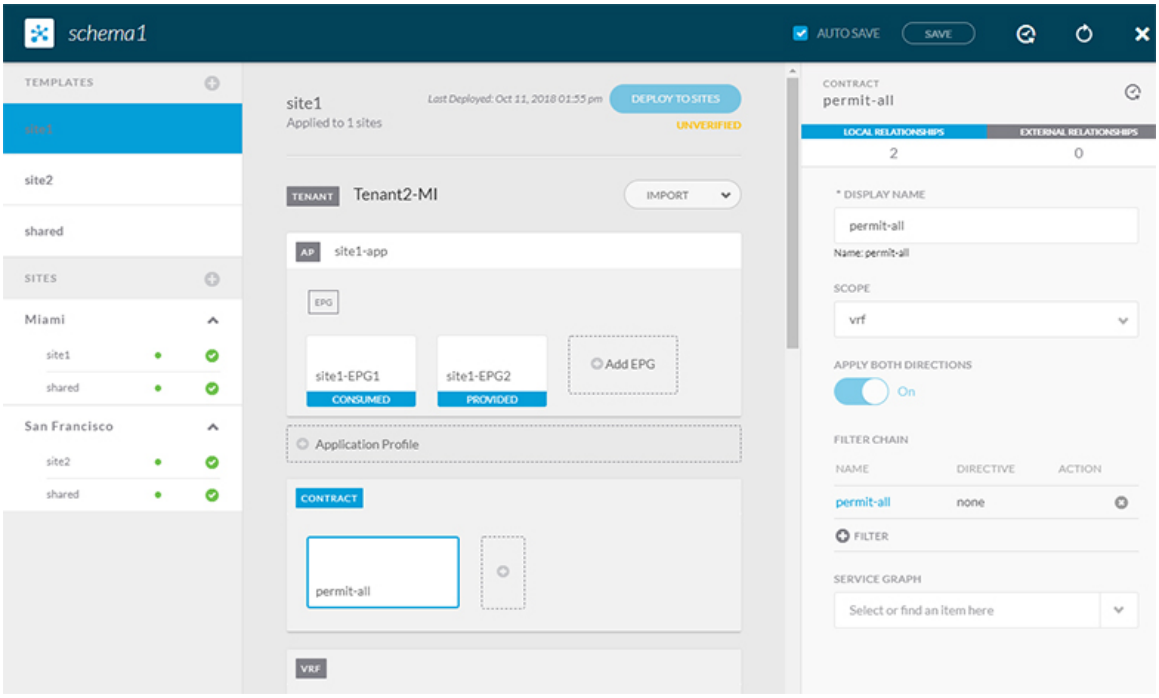
After creating and deploying the policy schemas and templates, the networking objects in the networking schema will display the number of external references by the policy schema elements. The object with external references will also be denoted by the ribbon icon as shown in the "Network Schema" figure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



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 MSO 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, MSO will not track any updates internally. And as a result, any configuration updates can be potentially overwritten by both subsequent API calls or GUI users.

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

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

Example

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

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

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

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



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

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

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

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

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

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

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

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

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

Creating Schemas and Templates

Before you begin

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

For more information, see the *User Access, Authentication, and Accounting* chapter in the *Cisco APIC Basic Configuration Guide*.

- You must have at least one available tenant that you want to incorporate into your site.

  For more information, refer to Adding Tenants.

Procedure

Step 1 Log in to your Nexus Dashboard and open the Multi-Site Orchestrator application.

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 right sidebar, specify the **Display Name**.
c) (Optional) Provide a **Description**.
d) 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.

e) 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 MSO, 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 MSO and a new object is imported with the **Include Relations** option enabled, MSO 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 MSO. 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 MSO 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 MSO.

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 6.

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.

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 application profile.

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 18. 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

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 Bridge Domains 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 8.
- 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 bridge domain.

For most fields, you will see the values you have configured at the template level.

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`—The subnet applies only within its tenant.
- `Advertised Externally`—The subnet can be exported to a routed connection.

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 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.

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

When using Cisco Multi-Site with this APIC fabric domain (site), indicates that the VRF, EPG, or BD using this subnet are mirrored from another site, which has a relationship to this site through a contract. Do not modify or delete the mirrored objects

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** and **Leaf** for the interface.

d) Provide the **Path** for the interface.

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.

For micro-segmentation, when a **Primary MICRO-SEG VLAN** is configured, the **Port Encap VLAN** is configured as an Isolated Secondary VLAN for the Primary VLAN. In this configuration, traffic is forwarded to the leaf that the

host is connected to. 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 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 6.

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 filter.

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 MSO, 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 6.
- You must have the VRF created as described in Configuring VRFs, on page 13

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.

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 virutal 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 MSO-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 MSO-orchestrated Multi-Site deployments.

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

The network visibility of the subnet.

- `Private to VRF`—The subnet applies only within its tenant.
- `Advertised Externally`—The subnet can be exported to a routed connection.

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

When using Cisco Multi-Site with this APIC fabric domain (site), indicates that the VRF, EPG, or BD using this subnet are mirrored from another site, which has a relationship to this site through a contract. Do not modify or delete the mirrored objects

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

Enables IGMP Snooping on the subnet

h) Click **Save**.

Step 5 (Optional) Enable **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 Domains Site-Local Properties, on page 17

Configuring Bridge Domains 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 14.
- 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.

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 enabled Host Based Routing on the bridge domain, individual host-routes (`/32` prefixes) are advertised from the border leaf switches. Border leaf switches along with the subnet advertise the individual end-point (EP) prefixes. The 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 advertise 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 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, MSO 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 **Bridge Domains** 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**.

d) 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.

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.

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 Multi-Site 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 6.
- You must have the VRF for the L3Out created as described in Configuring VRFs, on page 13

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 classification subnet.
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 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 the this option, **Shared Security Import** option becomes available, which allows access from the subnet to the endpoints within the fabric.

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.

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 Multi-Site Orchestrator GUI.

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.

Procedure

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

Step 2 Assign the templates to sites.

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

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

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

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

d) Click **Save**.

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

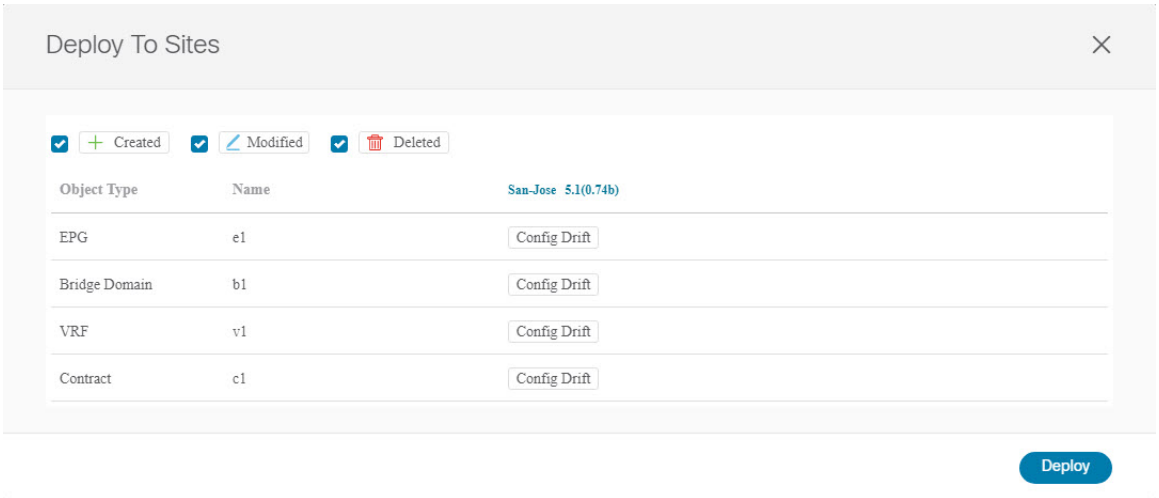
Step 4 In the top right of the template edit view, click **Deploy to sites**.

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

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

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

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



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

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.

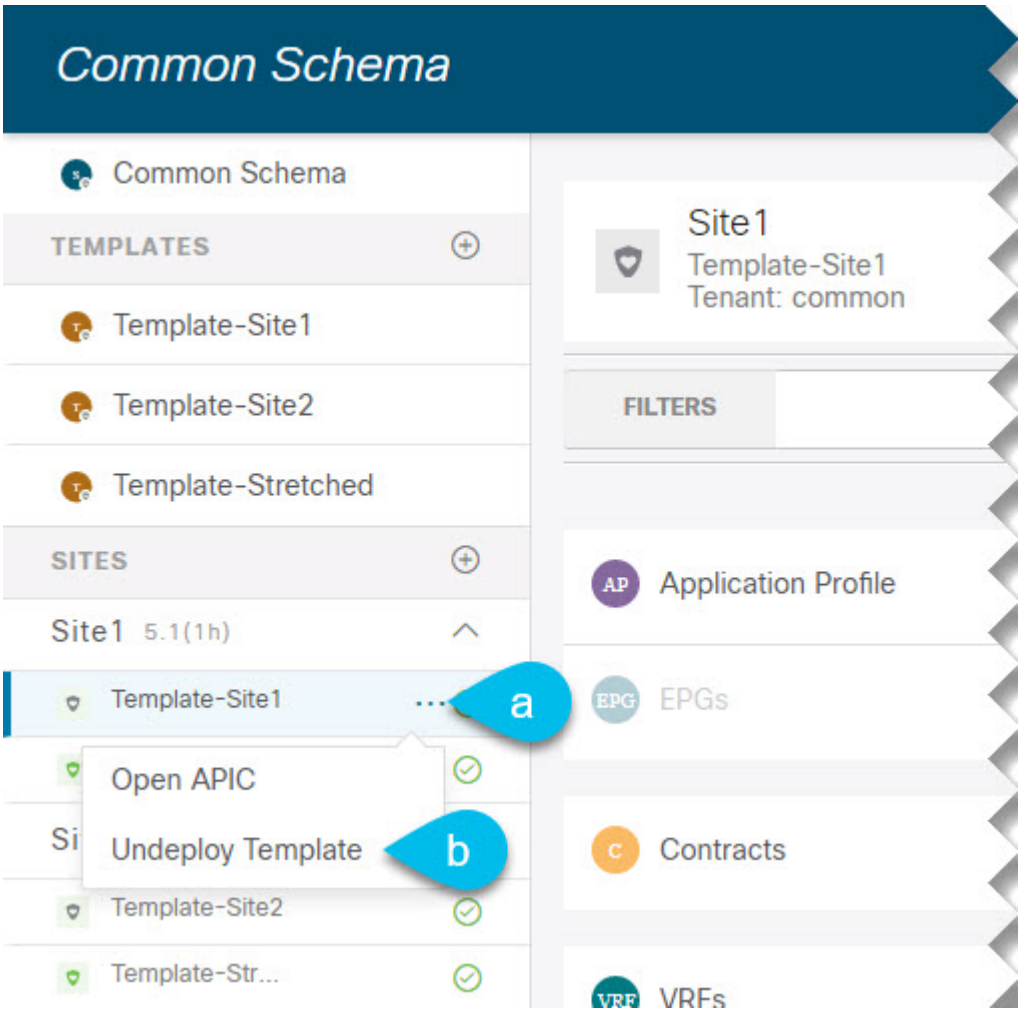
  Undeploying a template that was modified since it was last deployed would create a configuration drift because the set of objects deployed with the template would be different than the set of objects you try to undeploy after making changes to the template.

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**.

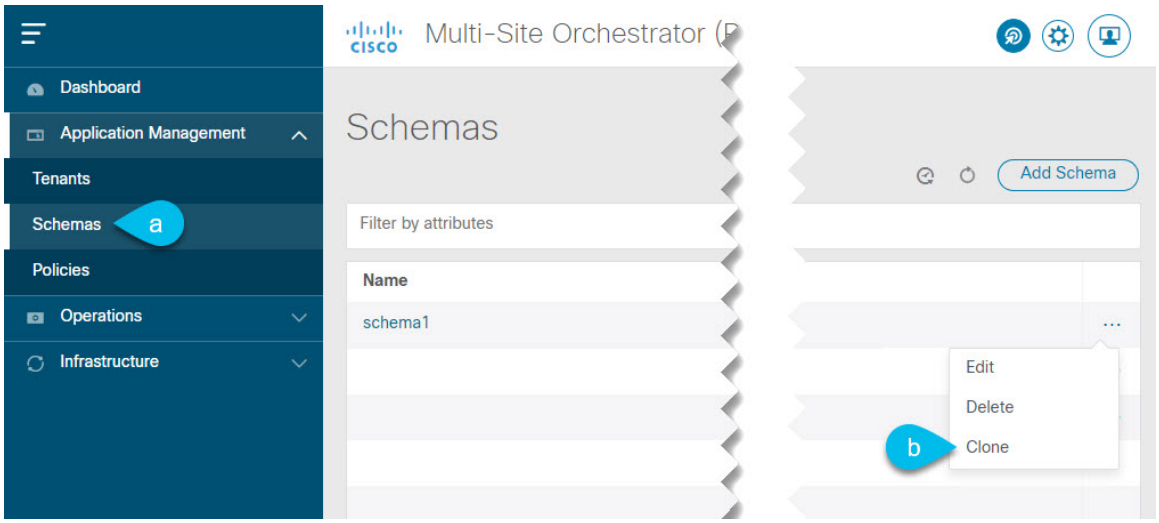
Cloning Schemas

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.

Procedure

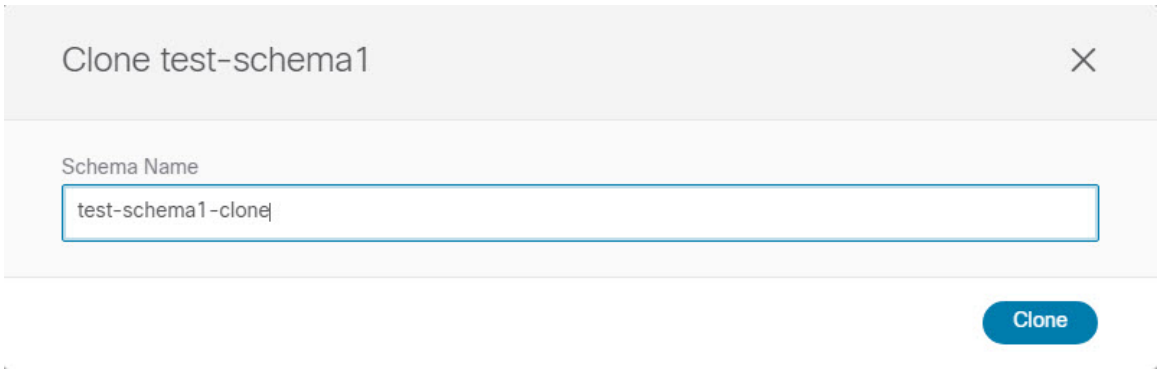
Step 1 Log in to your Multi-Site Orchestrator GUI.

Step 2 Choose the schema to clone..

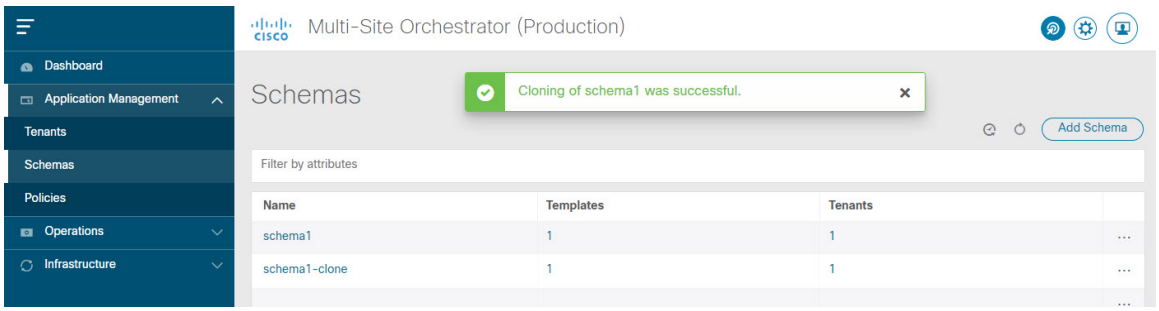


a) From the left navigation menu, select **Application Management** > **Schemas**.
b) From the **Actions** menu next to the name of the schema you want to clone, select **Clone Schema.**

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:

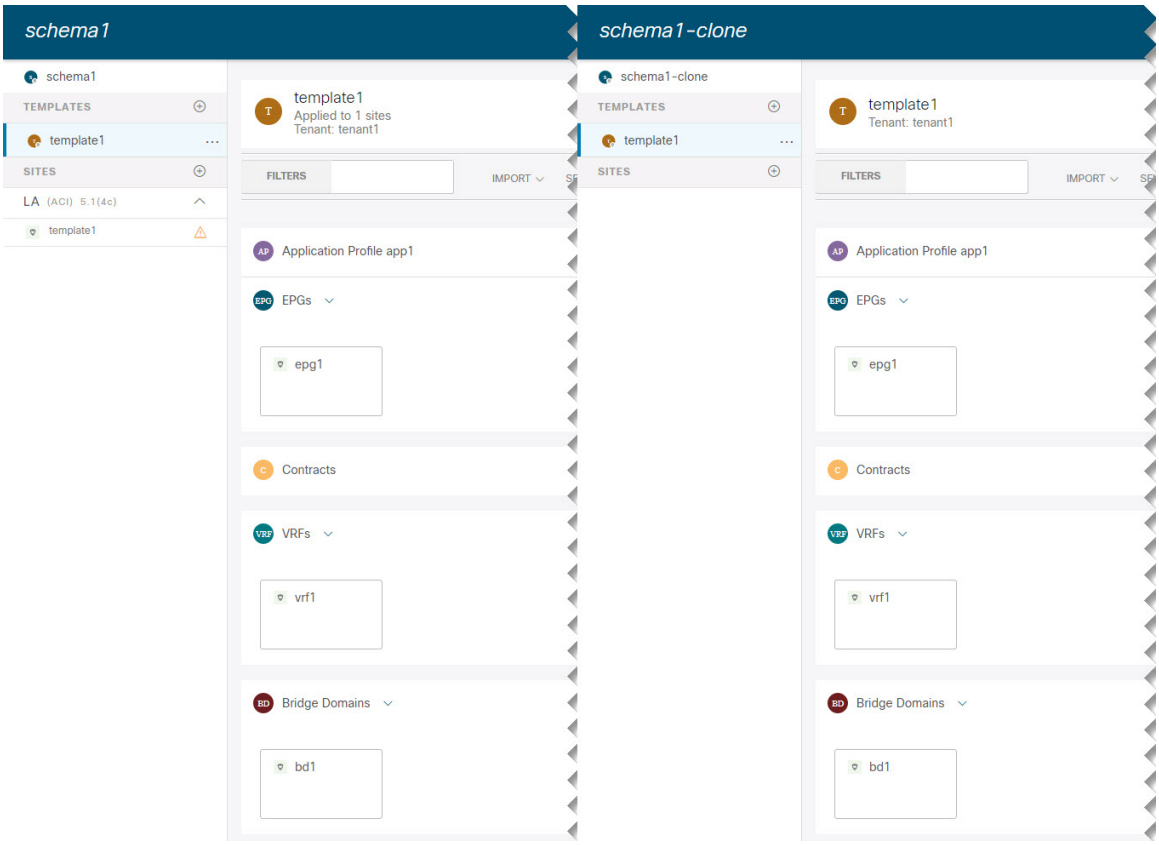


The new schema is created with the exact same templates (and their tenants' association), object, and policy configurations as the original schema.

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.

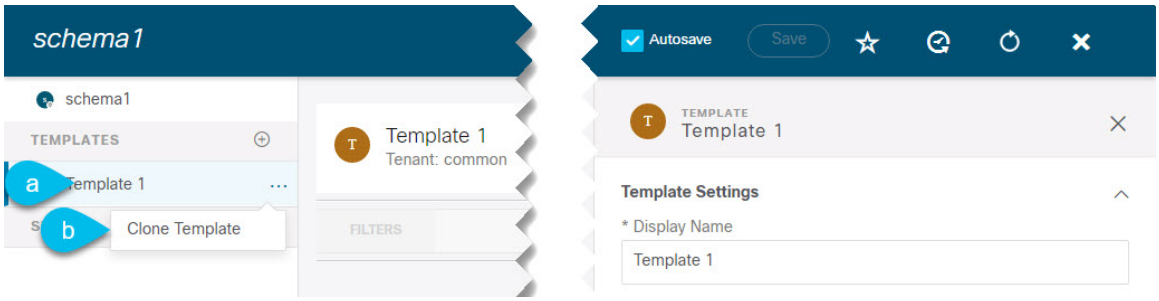
Procedure

Step 1 Log in to your Multi-Site 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 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.

If the template you are cloning has cross references to objects in other templates within its schema, then if you choose a different destination schema, the other templates will be cloned as well to preserve object relationships.

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

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

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

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

Migrating Objects Between Templates

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

- Only EPG and Bridge Domain (BD) objects can be moved between templates.
- Migrating objects to or from Cloud 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 Multi-Site 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.

Template Deployment View (Beta)

Starting with Release 3.3(1), you can view all objects currently deployed to sites from a specific template. This is a beta feature in this release.

Even though any given template can be deployed, undeployed, updated, and re-deployed any number of times, this API 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

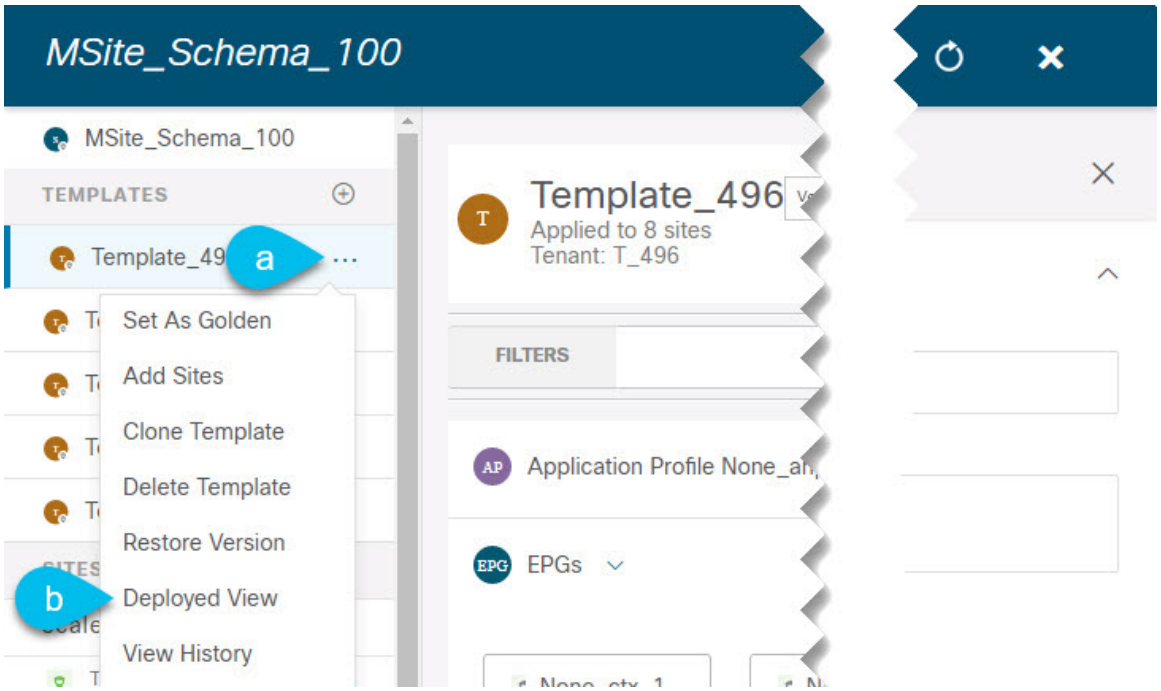
Step 1 Log in to your Multi-Site Orchestrator GUI.

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 **Deployment View** for the template.



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

b) Click **Deployment View**.

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

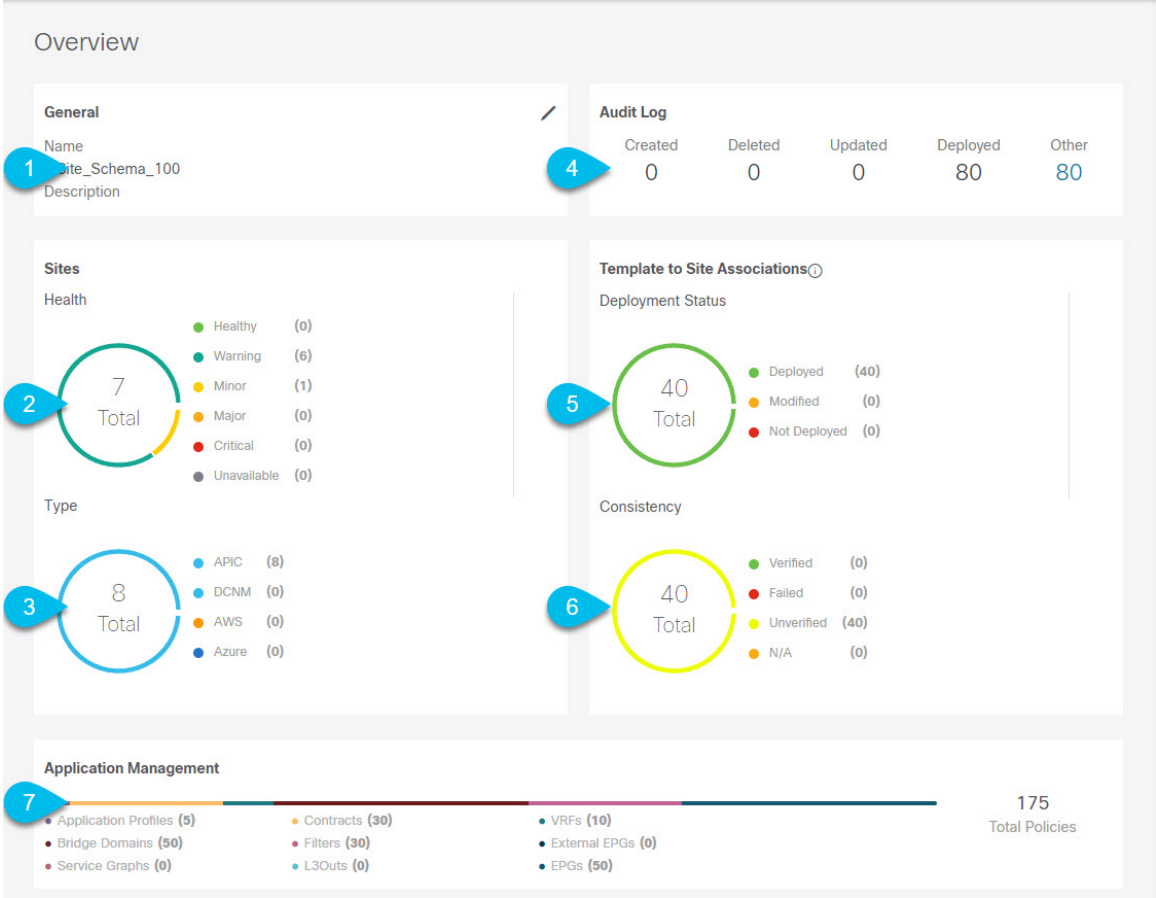
You will be able to see the XML content of all the objects that were pushed to the site when this specific template was last deployed.



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

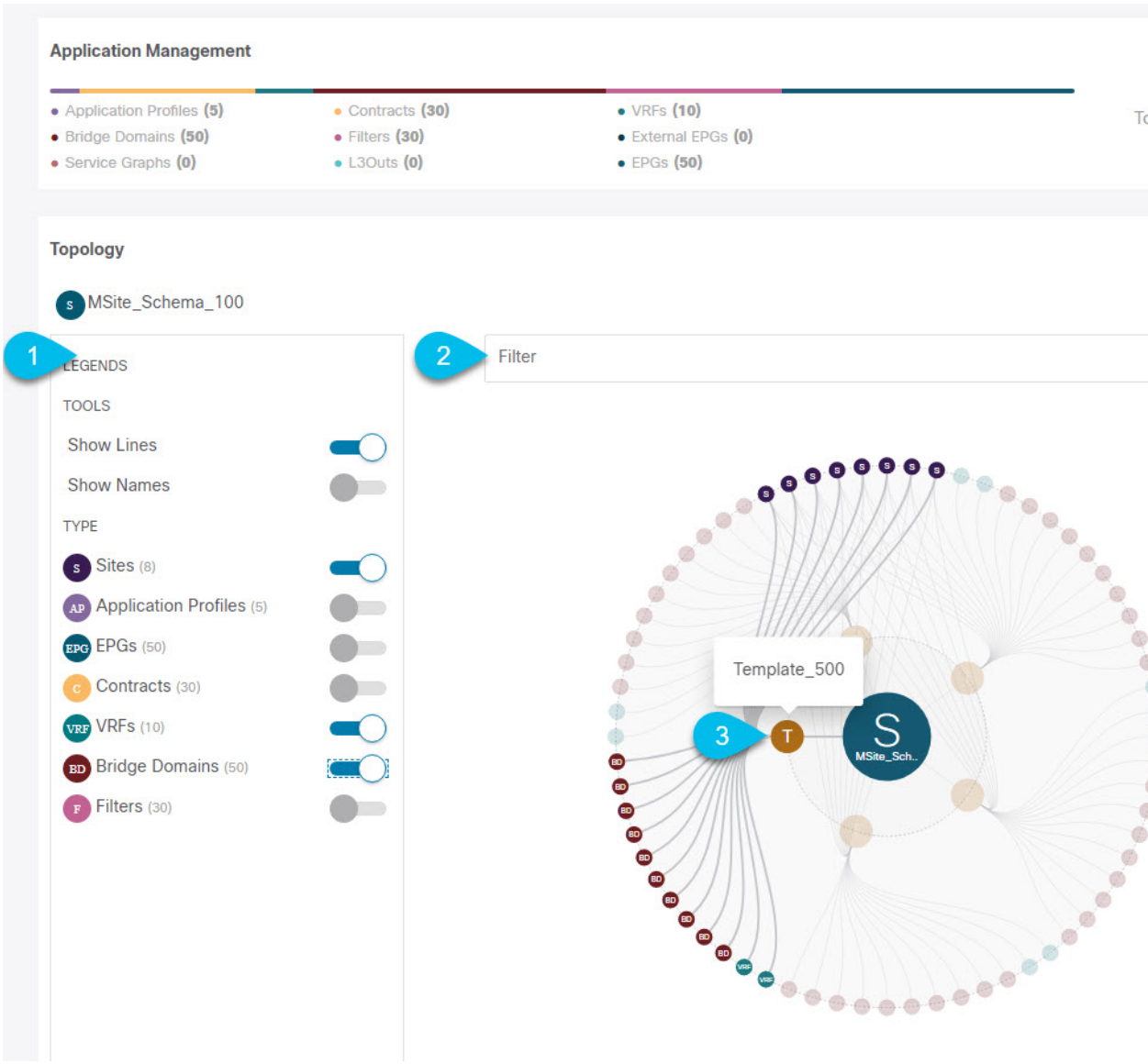


The following details are provided on this page:

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.

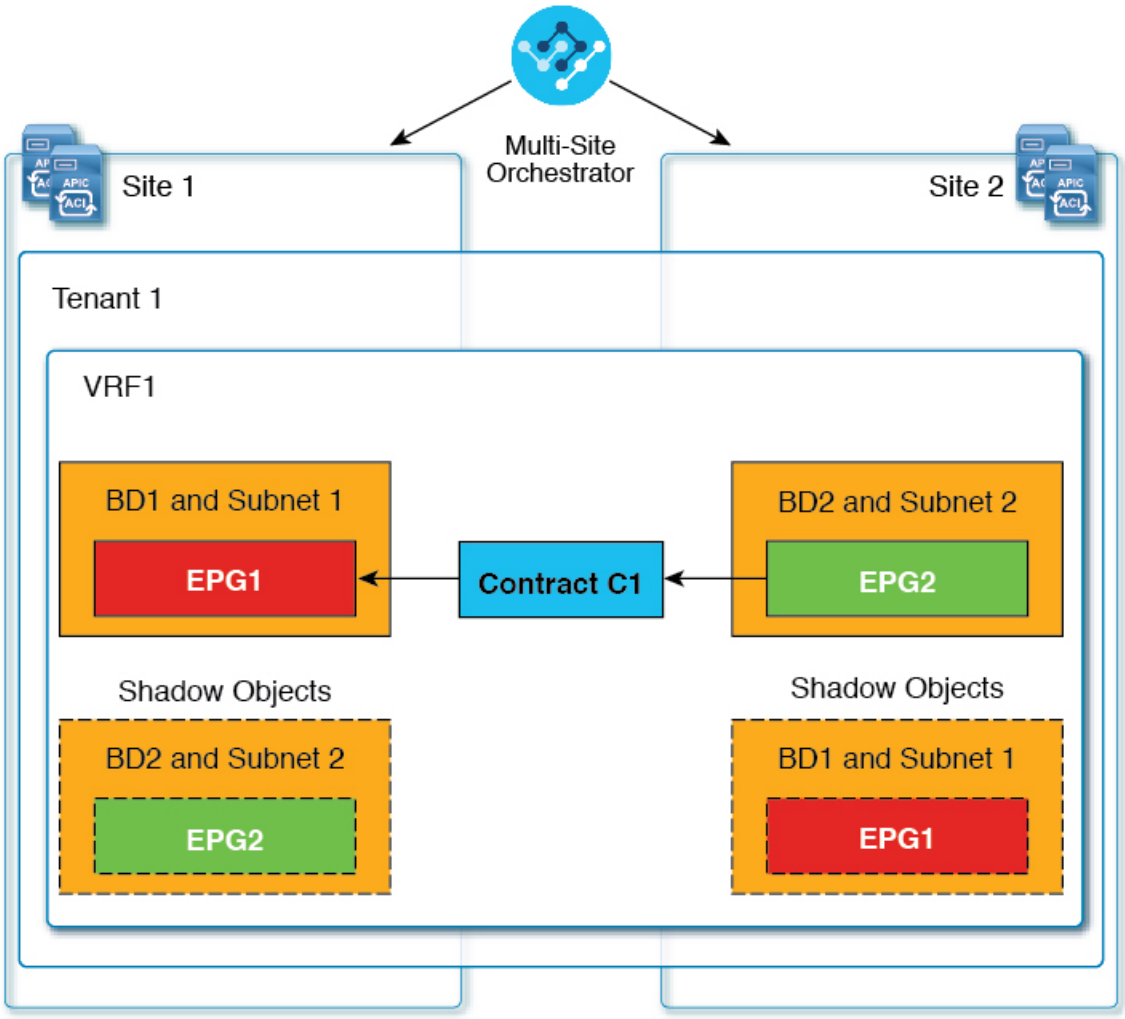
Shadow Objects

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 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)

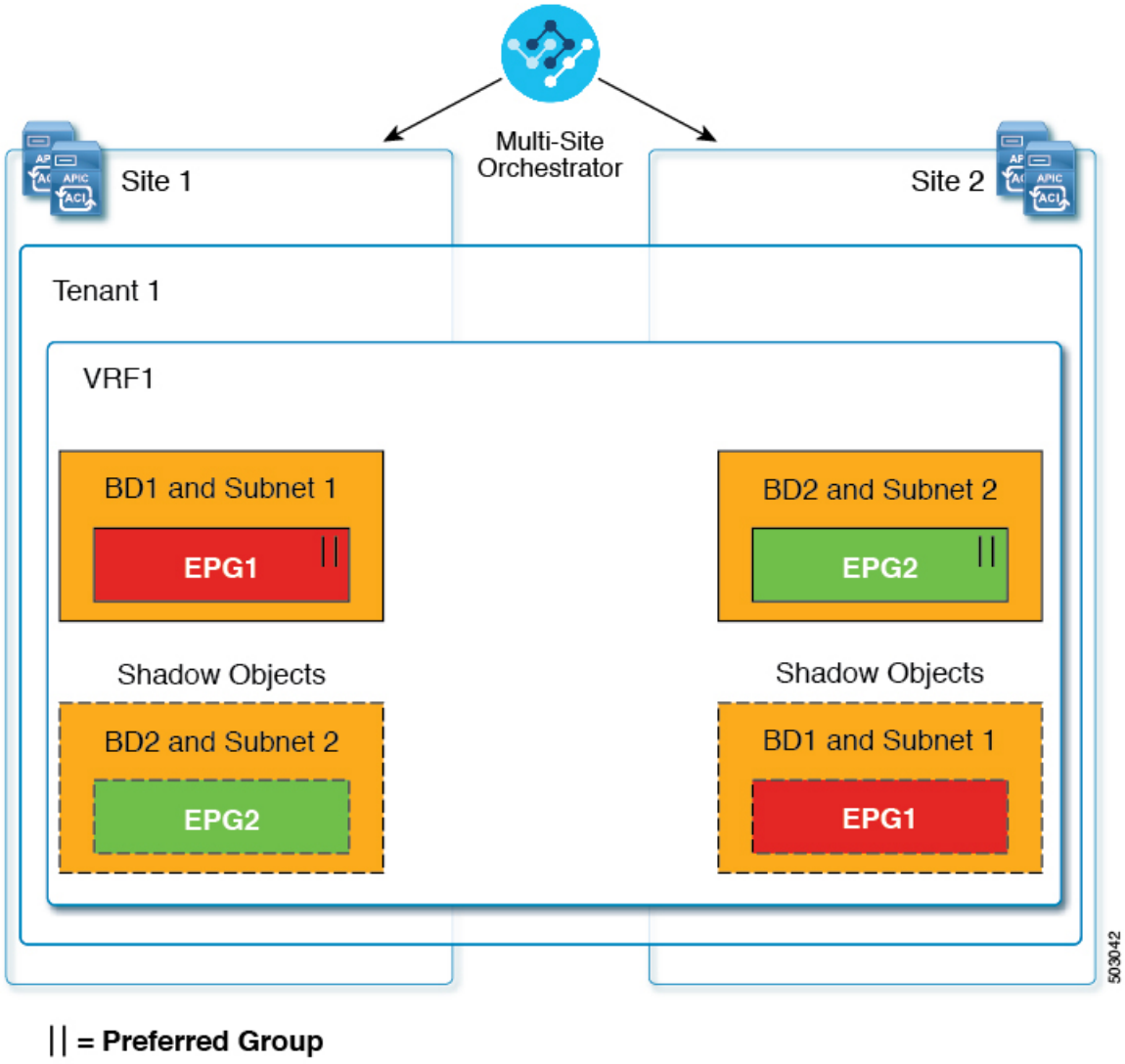
If your fabrics are running APIC Release 5.0(2) or later, when you select a shadow object in the APIC GUI, you will see a `This is a shadow object pushed by MSC to support intersite policies. Do not make any changes or delete this object.` warning at the top of main GUI pane. In addition, shadow

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. You can check for these options as described below:

Other Use Cases with Shadow Objects

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

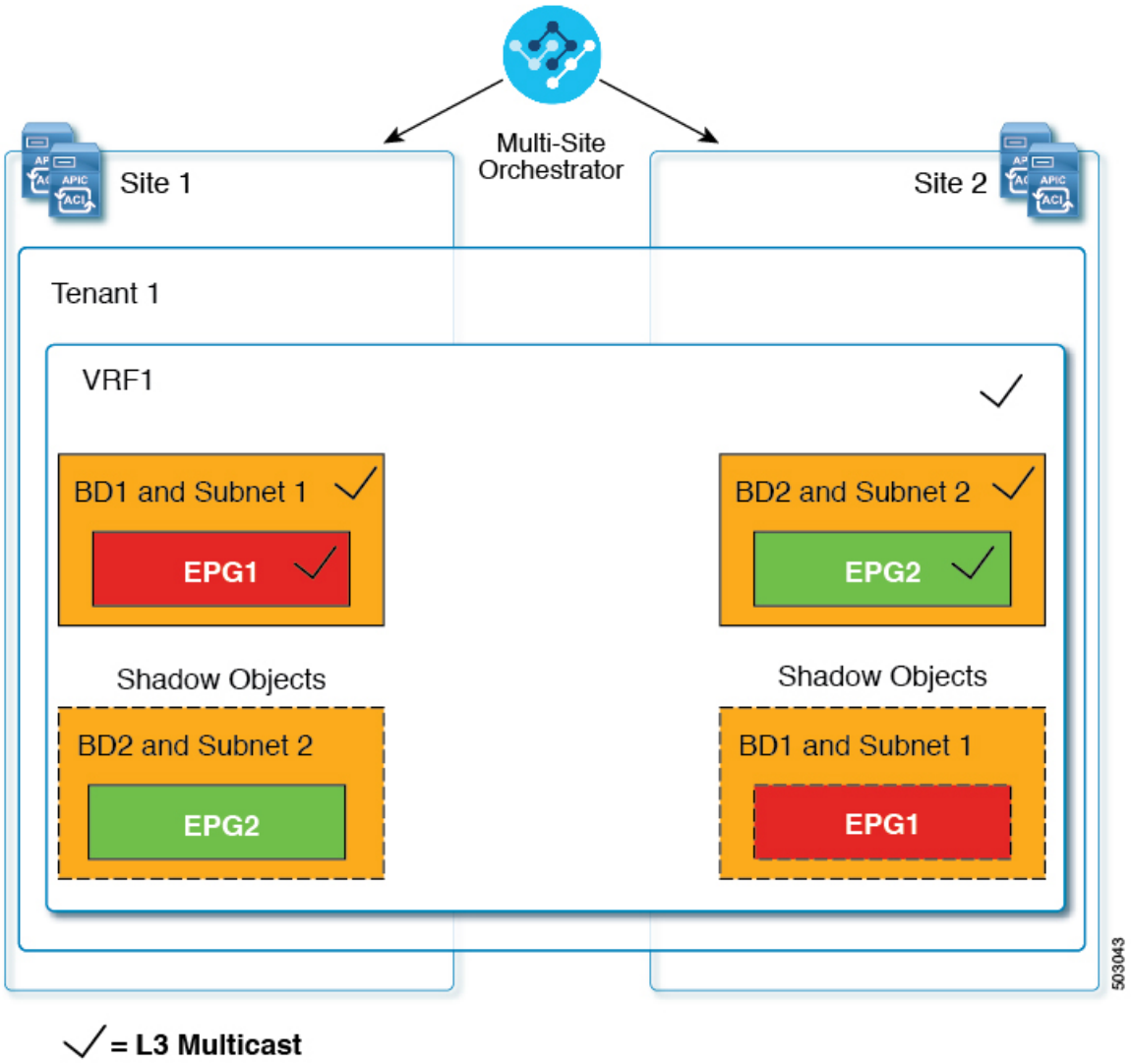
Figure 7: Preferred Group



|| = Preferred Group

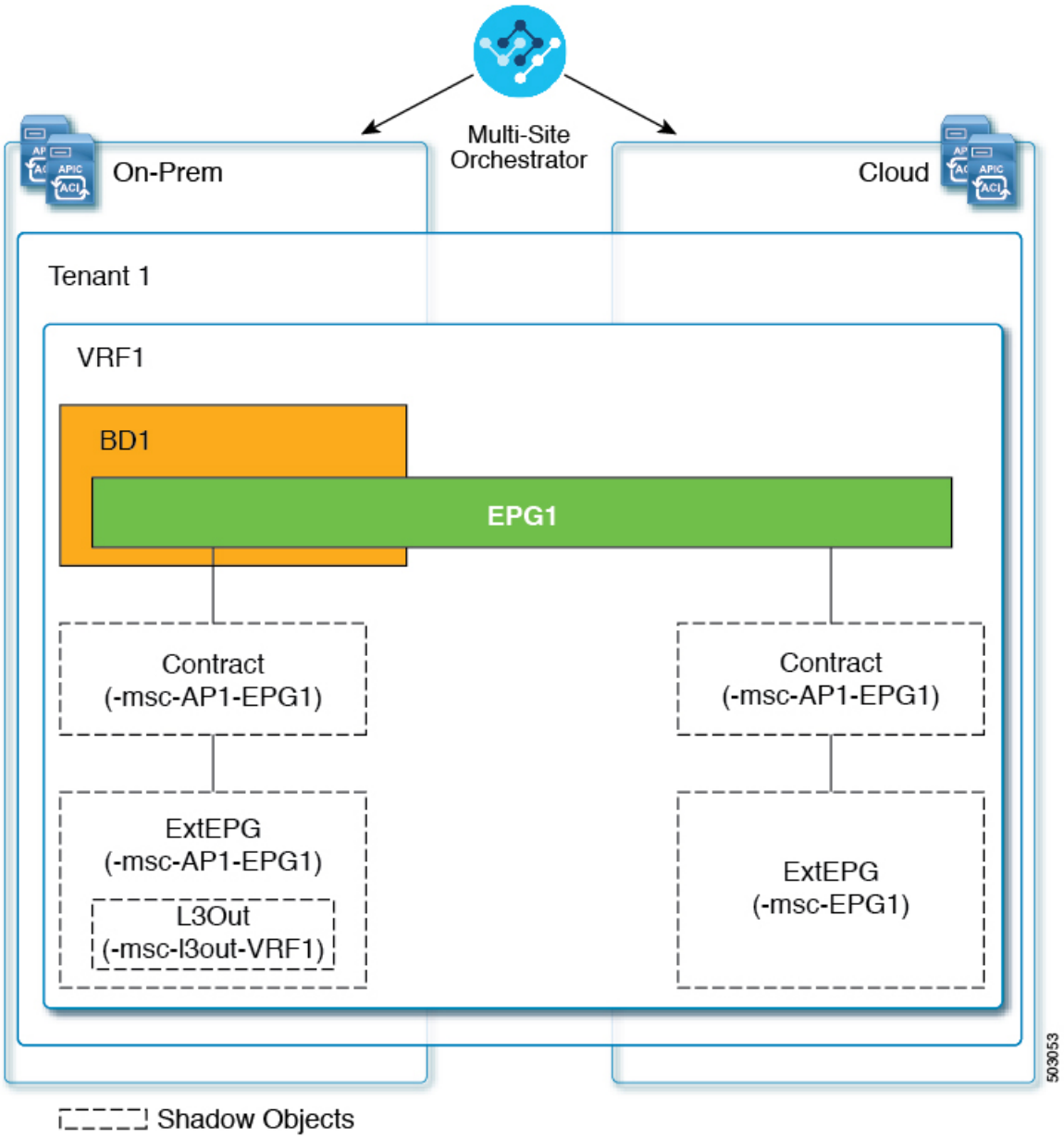
In case of multicast, the shadow objects are created only for EPGs/BDs that have multicast sources connected and the option explicitly configure at the EPG level.

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



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 Multi-Site 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 Multi-Site 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 Multi-Site Orchestrator.
  - When the Multi-Site 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.

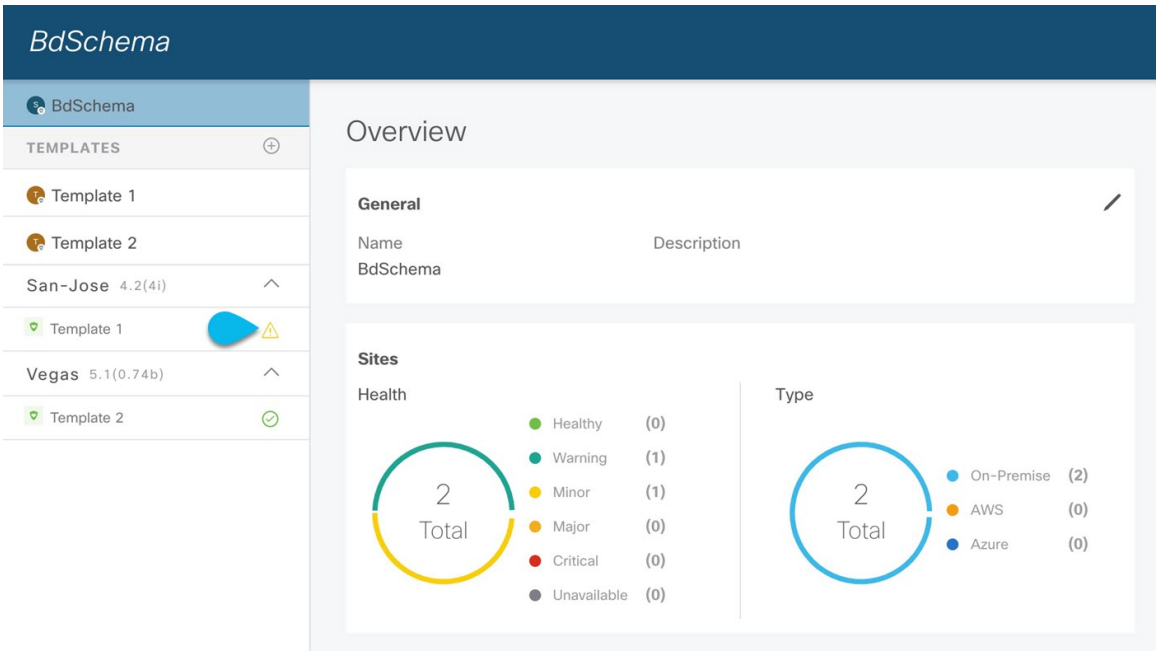
The setting is stored in the user profile and is enable or disabled separately for each user.

Step 4 Repeat the process for any additional APIC sites.

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 Multi-Site Orchestrator. These configuration discrepancies are referred to as **Configuration Drifts** and are indicated by the exclamation point next to the template name in the schema view as shown in the following figure:

Figure 10:



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

- **Configuration is modified in MSO**—when you modify a template in MSO 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 MSO 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.

  - If you want to undo the local changes, simply re-deploy the template from MSO to override any manual changes made at the site level.
  - If you want to preserve the local changes, import the modified objects from the site into the MSO template, save the schema, and re-deploy it back to the site.

- **MSO configuration is restored from backup**—restoring configuration from a backup in MSO 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 since the backup was created, restoring the backup would create a configuration drift.

- If you want to preserve the changes since the backup, import the modified objects from the site into the MSO template after configuration restore, save the schema, and re-deploy it back to the site.
- If you do not want to preserve the changes, simply re-deploy the template from MSO to override any configuration discrepancies at the site level.

  Note that this would allow you to override the configuration for objects and objects' properties that can be managed from MSO; it will not remove extra objects and properties configured at the APIC level that are not directly manageable from MSO.

- **MSO 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 in the site's APIC and the values are different from the defaults.
  - If you want to reset the object properties to the default values, simply re-deploy the template.
  - If you want to preserve the already configured custom values for the newly supported object properties, import the modified objects from the site into the MSO template, save the schema, and re-deploy the template back to the site.

- **MSO 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.
  - If you want to reset the object properties to the default values, simply re-deploy the template.
  - If you want to preserve the already configured custom values for the newly supported object properties, import the modified objects from the site into the MSO template, save the schema, and re-deploy the template back to the site.