

Cisco Hypershield User Documentation

Hypershield user guide

Contents

- Introduction
- Concepts
  - General concepts
  - Segmentation design
  - Security policies
  - Enforcement points
  - Smart Switch
  - Timescape and SCC
  - Platform access control
  - Logging and observability
- Installations
  - Install on vSphere
    - Satisfy the prerequisites
    - Deploy the bootstrap OVA
    - Configure and install Leader and Follower clusters
    - Install Hypershield on Leader and Follower
    - Verify the installation
- Operate the environment
  - Register a leader cluster with SCC
  - Assign user permissions

© 2026 Cisco and/or its affiliates. All rights reserved.

- Manage followers
- Manage leaders
- Manage Smart Switches
  - Register a Smart Switch
  - Update Smart Switch software
  - Enable high availability on a Smart Switch
  - Move a Smart Switch to another follower
  - Remove a Smart Switch
- Manage network objects
- Manage security policies
- Validate policies with drafts
- Manage logging groups
- Manage telemetry destinations
- Remove a controller

Create security policies

Backup and restore

- Back up Hypershield
- Restore to original IP addressing
- Restore to new IP addressing
- Troubleshooting

Troubleshooting

- Hypershield general troubleshooting

© 2026 Cisco and/or its affiliates. All rights reserved.

Security policies

Smart Switches

Enable Gateway API post bootstrap

Collect troubleshooting information for Hypershield

Resources

Caveats and limitations

Failure scenarios

Scalability testing

Reference

CLI command usage

Timescape API usage

Glossary

Platform information and support

Open a support case

© 2026 Cisco and/or its affiliates. All rights reserved.

Hypershield

Hypershield is a distributed security architecture that provides hardware-accelerated security policy enforcement using DPU (Data Processing Unit) technology. It enables Kubernetes-native security policy management with enforcement at the network edge.

Hypershield serves network security architects and engineers who design and deploy high-performance, distributed security infrastructure, as well as security operations teams who need unified policy management and real-time traffic observability.

© 2026 Cisco and/or its affiliates. All rights reserved.

Hypershield concepts

This section introduces the main concepts, architecture, components, and deployment patterns used by the Hypershield platform.

© 2026 Cisco and/or its affiliates. All rights reserved.

General concepts

This page introduces the general concepts of Hypershield: its three architectural planes, the security policy flow, and its two operating modes.

Architecture

Hypershield separates policy administration, policy distribution, and policy enforcement into three planes. Each plane has a defined responsibility and interacts with the other planes through well-defined interfaces.

The architecture consists of three planes:

- Management plane

  This plane administers security policies and provides inventory management, observability, and management APIs. It consists of Timescape, the local management interface.

  Timescape can optionally be fronted by Cisco Security Cloud Control, which reaches Timescape through the Secure Connector (not shown in the diagram, for simplicity). The choice of management interface determines the operating mode; see Operating modes.

- Control plane

  This plane distributes approved security policies to managed enforcement points and reports deployment status. The on-prem controller performs this role.

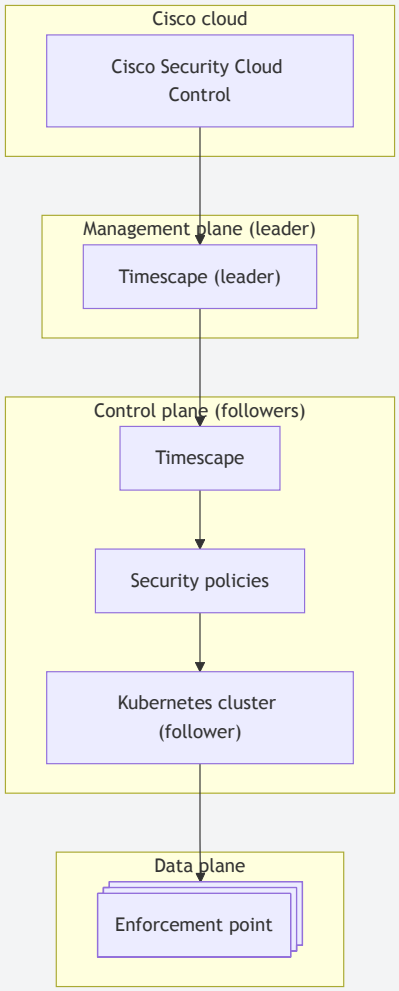
- Data plane

  This plane enforces security policy on traffic at supported enforcement points. See Enforcement points for the enforcement points supported today.

The following diagrams show the logical relationships between components; physical deployment may differ. The management plane runs on the leader cluster and the control plane runs on one or more follower clusters; see Leader and follower clusters.

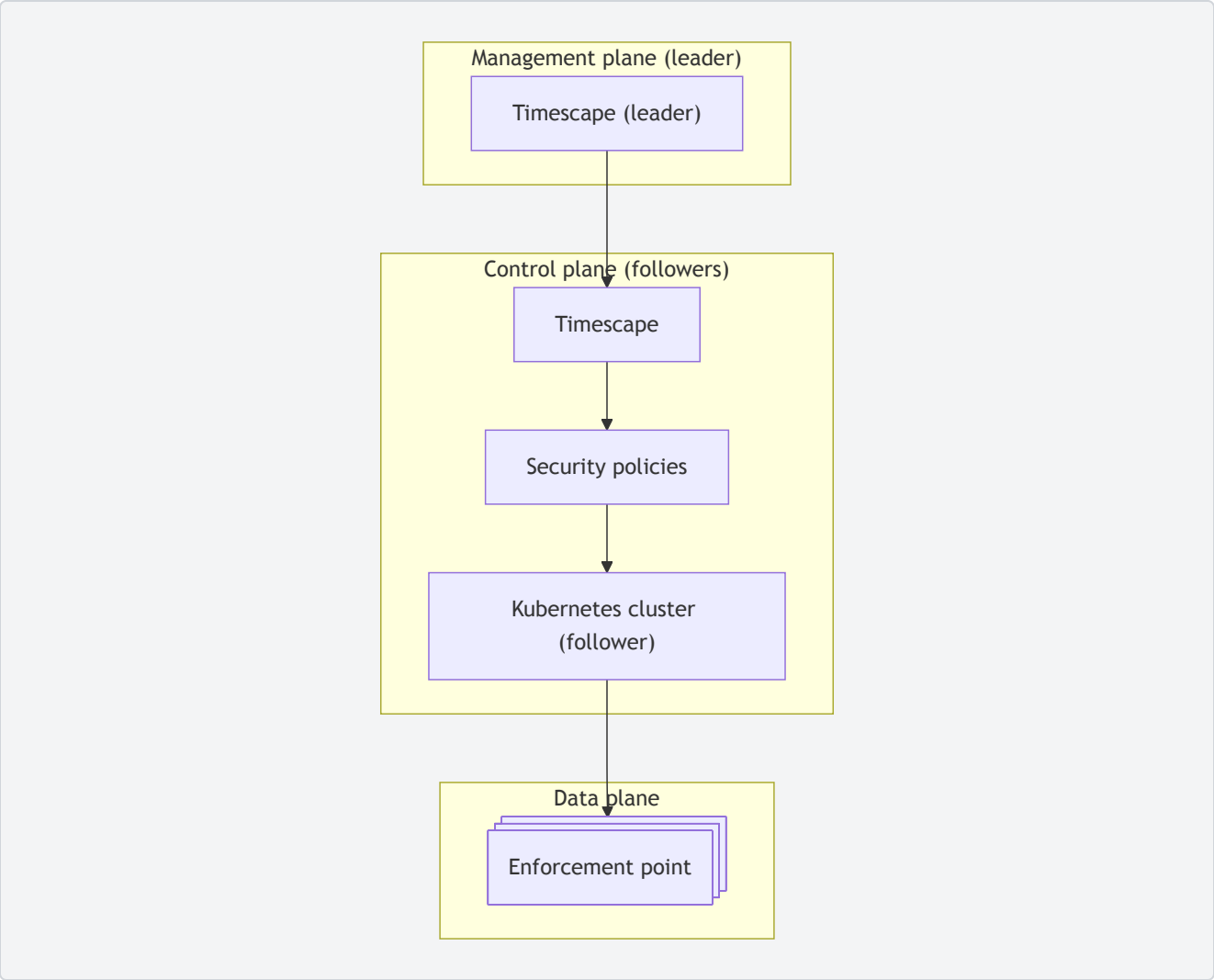
Connected mode

© 2026 Cisco and/or its affiliates. All rights reserved.



Disconnected mode

© 2026 Cisco and/or its affiliates. All rights reserved.



Security policy flow

Each architectural plane corresponds to one stage of the security policy flow:

- Policy administration (management plane)

  Administrators define, review, and approve security policies and policy objects that describe the protected resources. Policies represent administrative intent and do not affect network traffic until they are deployed.

- Policy deployment (control plane)

  After policy is approved, the control plane distributes it to the appropriate enforcement points and reports deployment status back to the management plane, allowing administrators to monitor policy rollout.

- Policy enforcement (data plane)

© 2026 Cisco and/or its affiliates. All rights reserved.

Each enforcement point applies its locally installed policy to network traffic. Policy decisions are made locally and continue to be enforced even if connectivity to the management or control plane is temporarily unavailable, until the policy is updated or removed.

Operating modes

Hypershield supports two operating modes: connected and disconnected. The operating mode determines where security policies are administered. It does not change how those policies are deployed or enforced: in both modes, the control plane is local and enforcement is local.

- Connected mode

  This mode uses Cisco Security Cloud Control (SCC) as the management interface, providing centralized policy administration across managed environments. Approved policy is distributed to the local leader (connected by the Secure Connector), and deployment status is reported back to Cisco Security Cloud Control.

- Disconnected mode

  This mode uses Timescape as the local management interface, providing local policy administration. Intended for environments that require local administration or that operate without connectivity to SCC, such as air-gapped or isolated deployments.

> **Note**
>
> The two modes are independent. That is, you can still access Timescape directly while in connected mode.

Leader and follower clusters

The management plane and the control plane are implemented as separate Kubernetes clusters. A Hypershield deployment includes one leader cluster and one or more follower clusters. Timescape coordinates security policy across the clusters.

- Leader

  This cluster hosts the management plane. Administrators interact with the leader to define, review, and approve security policy. It is the source of truth for policy state and distributes approved policy to the followers.

- Follower

  This cluster hosts the control plane for a set of enforcement points, typically the Smart Switches within a single failure domain such as a datacenter. It receives approved policy from the leader and is responsible for pushing that policy to its enforcement points and for reporting deployment status back.

© 2026 Cisco and/or its affiliates. All rights reserved.

Scalability

The leader/follower design allows the control plane to scale beyond what a single follower can manage. A single follower serves a bounded number of enforcement points; larger deployments distribute enforcement points across multiple followers, all coordinated by the same leader.

Availability

Separating the leader from the followers reflects the different availability requirements of the two planes:

- A follower shares the failure domain of the enforcement points it serves. If those enforcement points are unavailable, the follower has nothing to serve, so it does not need to be available either.
- The leader is not required for policy enforcement to continue. Enforcement points enforce their locally installed policy independently of the leader, so leader unavailability affects only new configuration changes.

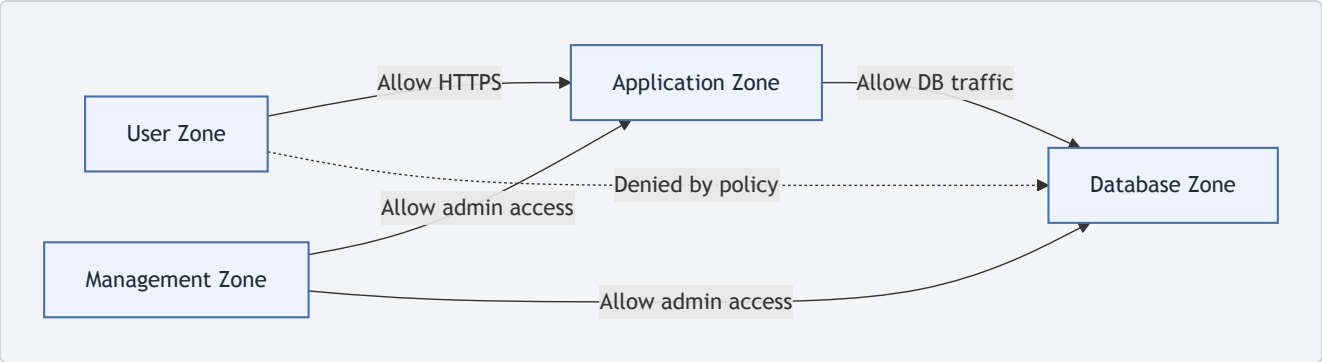
© 2026 Cisco and/or its affiliates. All rights reserved.

Segmentation design

Hypershield supports two complementary segmentation designs for L3/L4 policy: zone-based segmentation and workload-centric segmentation. Both are ways of expressing security intent, not statements about where the platform is deployed. You can apply either design across the supported Hypershield enforcement points, and you can combine them in the same environment.

Both designs organize security policy around policy domains. A policy domain is a portion of the environment that is treated as a single unit for security purposes - a set of systems that share a common trust profile or a common operational role. The two designs differ in how coarsely or finely they draw those domains.

Zone-based



Zone-based segmentation treats each policy domain as a broad security zone - an application tier, an environment such as production or development, a tenant or service domain, or a management or user network boundary. You define communication rules between zones rather than between individual workloads.

When to use it

Zone-based segmentation is usually the better fit when you want:

- broad controls between trust zones
- policy that aligns with existing network segmentation
- a scalable way to express macro-segmentation
- clear enforcement at environment, tier, or tenant boundaries

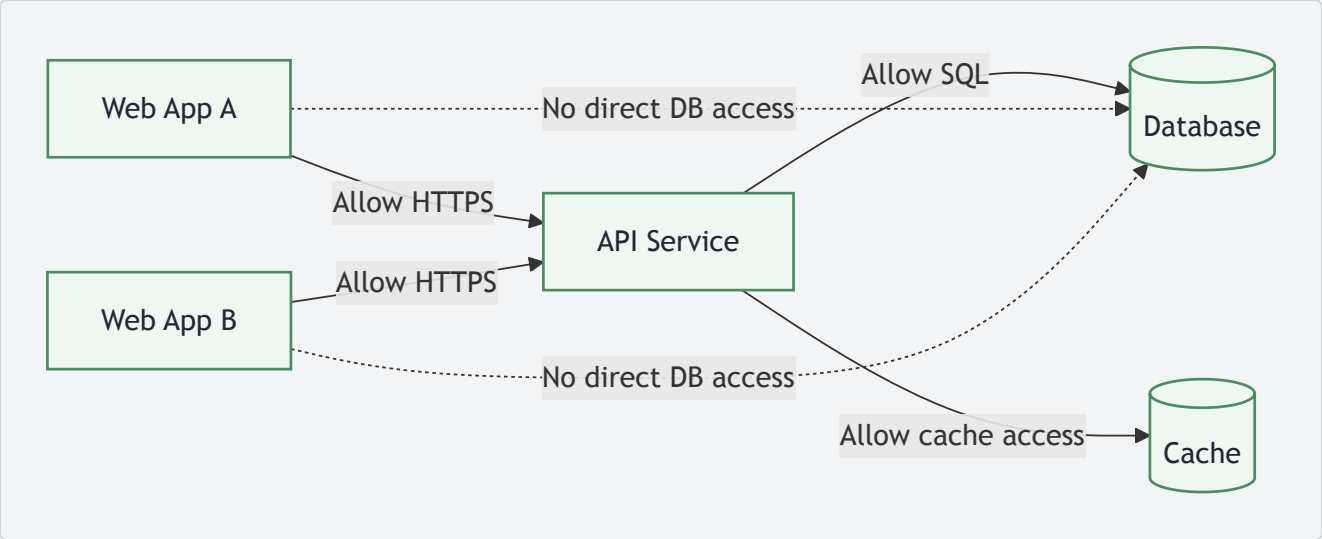
It is especially natural when the surrounding network already uses constructs such as VRFs, VLANs, routed environment boundaries, or tenant service domains, because policy can align with boundaries that operators already recognize.

Design considerations

© 2026 Cisco and/or its affiliates. All rights reserved.

Zone-based segmentation is strong at clarity and scale, and it defines the main security boundaries in an environment. It may not be precise enough for every application-level dependency; where more precision is needed inside a zone or between subsets of systems, you can add more detailed policy without abandoning the zone-based structure.

Workload-centric



Workload-centric segmentation treats each policy domain as a specific workload group defined by identity, placement, or role. You describe which applications, services, or workload sets should communicate with each other, rather than expressing rules only between broad zones.

When to use it

Workload-centric segmentation is usually the better fit when you want:

- granular access control between applications or services
- policy that reflects workload relationships rather than only network boundaries
- more precise control over east-west traffic
- tighter separation between systems that share the same broader environment

This design is often the right choice when a broader policy domain would be too coarse for the security controls you need.

Design considerations

© 2026 Cisco and/or its affiliates. All rights reserved.

Workload-centric segmentation gives you more precision, but it can also become more detailed to manage. The design works best when workload groupings are well understood, application dependencies are reasonably clear, and the added precision provides real security value.

Choosing a design

In practice, users usually choose between the two designs based on questions such as:

- Which unit is the security concern - a broader network boundary, or specific workload relationships?
- Does the existing network design already provide natural trust boundaries to align policy with?
- Do you need to distinguish access rules between workloads that share the same broader environment?

Design tradeoffs

The two designs carry different tradeoffs. Zone-based segmentation is strong at clarity and scale, but it is intentionally broader and less precise than workload-centric segmentation. Workload-centric segmentation gives you more precision, but becomes more detailed as the number of protected applications and service relationships grows.

Combining the two designs

The two designs are not mutually exclusive. A common approach is to use zone-based segmentation to define the main security boundaries in an environment, and then add workload-centric policy where you need more precision inside a zone or between subsets of systems.

This lets you keep a clear macro-level structure while still expressing the finer-grained rules that specific applications or services need.

© 2026 Cisco and/or its affiliates. All rights reserved.

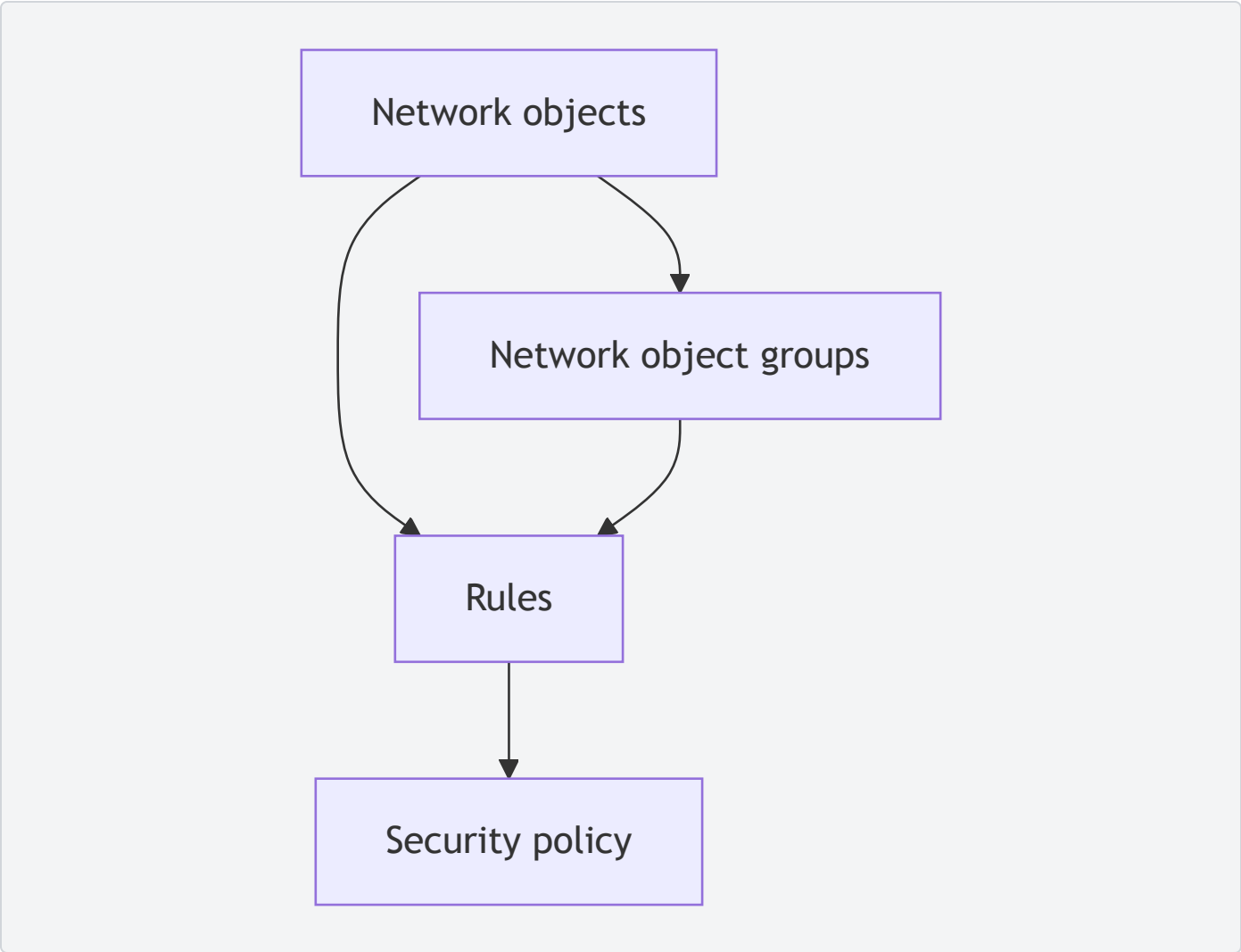
Security policies

In Hypershield, a security policy expresses the intended communication between protected resources: which traffic to allow, deny, or segment. A policy becomes effective only after it is deployed to one or more enforcement points.

This page describes how a security policy is composed, how it progresses from definition to enforcement, how the active set of policies and rules is evaluated, and briefly states how security policies are implemented internally.

Structure

A security policy is built from a small set of parts. Each policy contains one or more rules; rules reference network objects, either directly or grouped as network object groups. Separating these parts from policy rules lets multiple policies reuse the same definitions, reducing duplication and simplifying administration.



© 2026 Cisco and/or its affiliates. All rights reserved.

Rules

A security policy contains one or more rules. Each rule identifies the traffic to which it applies and the action to take when traffic matches. Rather than defining resources inline, rules reference reusable network objects.

Network objects

A network object is a reusable definition of a protected resource. It can represent an individual endpoint, a subnet or CIDR range, or another addressable network resource (VRF, VLAN). Rules reference network objects directly, or via network object groups.

Network object groups

A network object group bundles related network objects into a single reusable unit. A rule can reference a group instead of listing individual objects, which simplifies rule definition and reduces duplication.

Network object groups are maintained independently of the policies that reference them. Updating a group is automatically reflected in every rule that uses it.

Lifecycle

A security policy moves through the following stages, from definition to enforcement and later change:

- Define

  An administrator creates the policy that describes the intended communication behavior.

- Review

  Optionally, an administrator stages the change as a draft (Timescape) and reviews it before deployment. See Validate policies with drafts for the draft workflow.

- Validate

  Hypershield checks the policy for structural validity as a deployment-time gate. A failure rejects the deployment.

- Deploy

  Hypershield distributes the validated policy to the relevant enforcement points, where it becomes part of the active policy state.

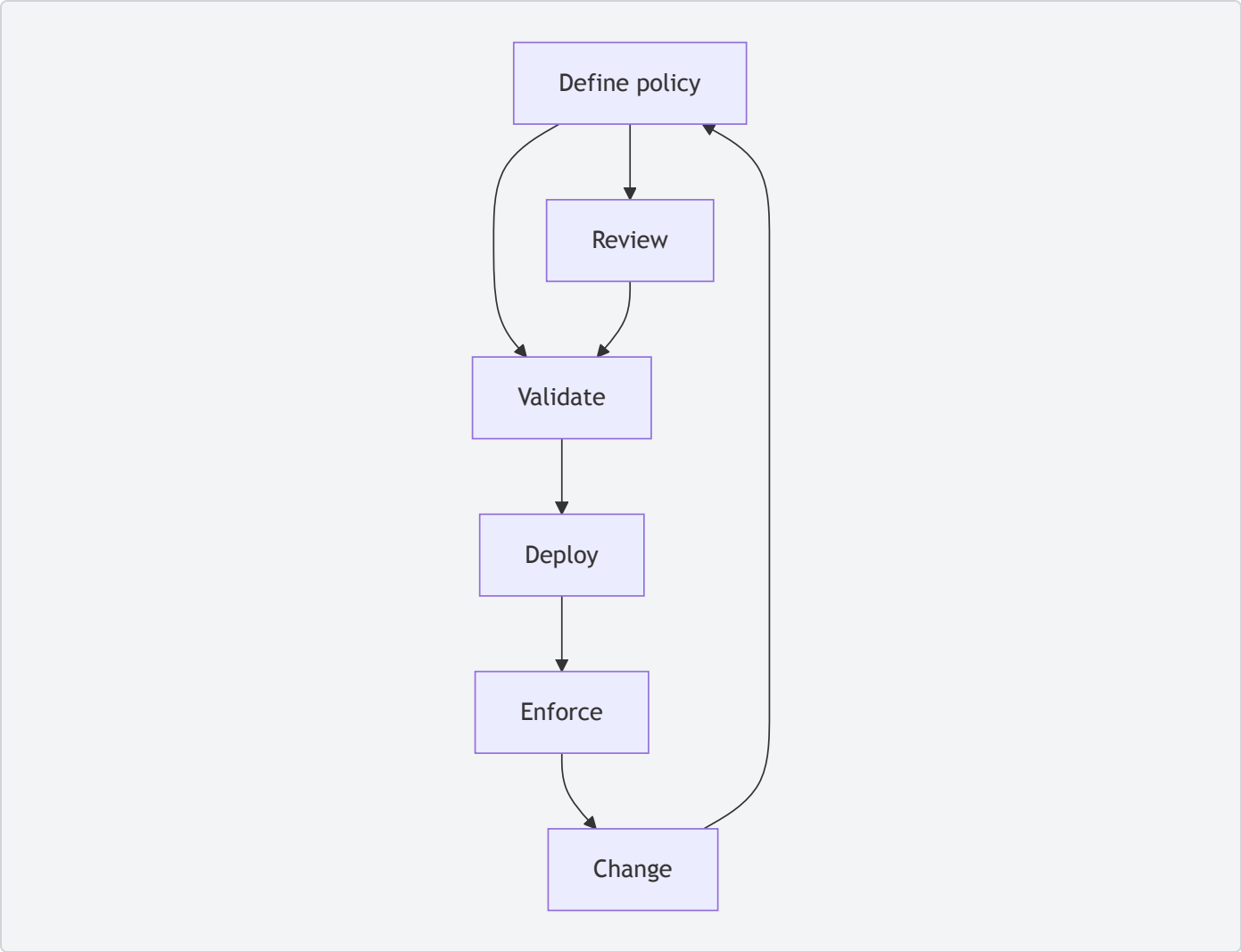
- Enforce

  Enforcement points evaluate traffic against the active policy state. Enforcement continues locally even if the management or control plane is temporarily unreachable.

- Change

© 2026 Cisco and/or its affiliates. All rights reserved.

An administrator updates, replaces, or removes the policy. Each change returns to validation and deployment before it becomes active.



Behavior on change

When a policy is changed (updated or removed), Hypershield distributes the new state to the affected enforcement points. Existing connections may continue according to their current connection state, depending on how the enforcement point handles live sessions.

For instance, on a Smart Switch, if the policy change does not potentially block a current session's traffic then the session will continue unaffected. However, if the change does represent a potential block, then the existing session will be interrupted after a pre-configured timer (60 seconds).

© 2026 Cisco and/or its affiliates. All rights reserved.

Evaluation and rule order

Order matters when multiple rules could apply to the same traffic. Broad rules, narrow rules, and overlapping rules can affect each other, so the final outcome depends on the active policy set rather than any single rule in isolation.

For example, a broad rule can make a narrower rule ineffective if the broad rule matches the same traffic first. This is sometimes called a shadowed rule. That kind of overlap can hide policy intent and make live behavior differ from what an administrator expects.

As policy sets grow, it is important to consider how rules interact, how changes affect the overall result, and whether a policy behaves as intended before it is enforced.

Kubernetes implementation

Internally, Hypershield represents security policies and network object groups as Kubernetes custom resources, such as `SmartSwitchNetworkPolicy` and `NetworkObjectGroup`. Administrators interact with these resources through Timescape and SCC, either directly or through Cisco Security Cloud Control (SCC), which fronts Timescape. Timescape also exposes a public REST API for automation; see Timescape API usage.

> **Caution**
>
> Do not create, modify, or delete security policies or network object groups directly through the Kubernetes API. Timescape coordinates policy changes across the Leader:Follower topology, and bypassing it can leave the deployment in an inconsistent state. Use only Timescape (or its REST API) for these changes.

© 2026 Cisco and/or its affiliates. All rights reserved.

Enforcement points

Hypershield enforces security policies at defined points in the network. This page describes the supported enforcement points.

Today, the Smart Switch is the only enforcement point. Additional enforcement points will be documented here as they become available.

Smart Switch

A Smart Switch operates as a managed enforcement point in a Hypershield deployment. It enforces security policies inline as traffic traverses the switch.

> **Note**
>
> Smart Switches are deployed as redundant pairs that function as a single enforcement point; see Smart Switch for details on how the Smart Switch works internally and how the pair provides high availability.

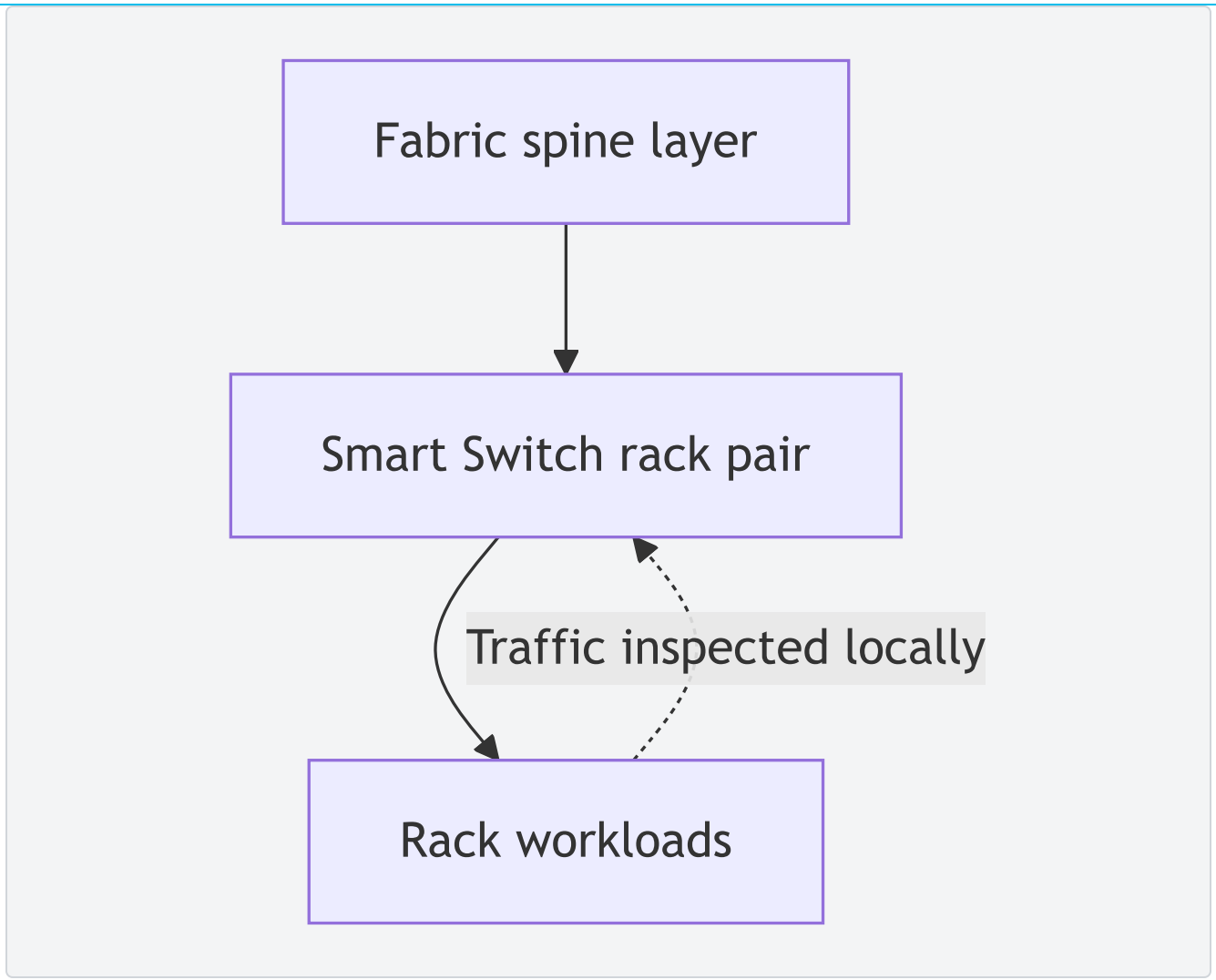
Placement

Hypershield supports two ways to place Smart Switch enforcement in the network: at the rack edge (top-of-rack) or at a shared inspection point in the fabric (services). These models describe where the Smart Switch pair sits and how traffic reaches it. They do not define security policy (see Security policies) or the segmentation design (see Segmentation design).

In a top-of-rack placement, each rack can have its own enforcement point - the Smart Switch sits at the rack edge as the access leaf that workloads attach to directly. In a services placement, traffic is steered to a shared enforcement layer elsewhere in the fabric - the Smart Switch pair sits at a common convergence point that traffic must cross.

Top-of-rack

© 2026 Cisco and/or its affiliates. All rights reserved.



Because the Smart Switch sits inline with local traffic, inspection happens without leaving the rack. This makes top-of-rack a localized placement model, with the enforcement point distributed across the fabric.

When to use it

Top-of-rack placement is usually the better fit when you want:

- east-west segmentation close to the workload
- enforcement at or near the rack edge
- lower dependence on a centralized inspection tier
- security placement that closely follows workload placement

Scaling

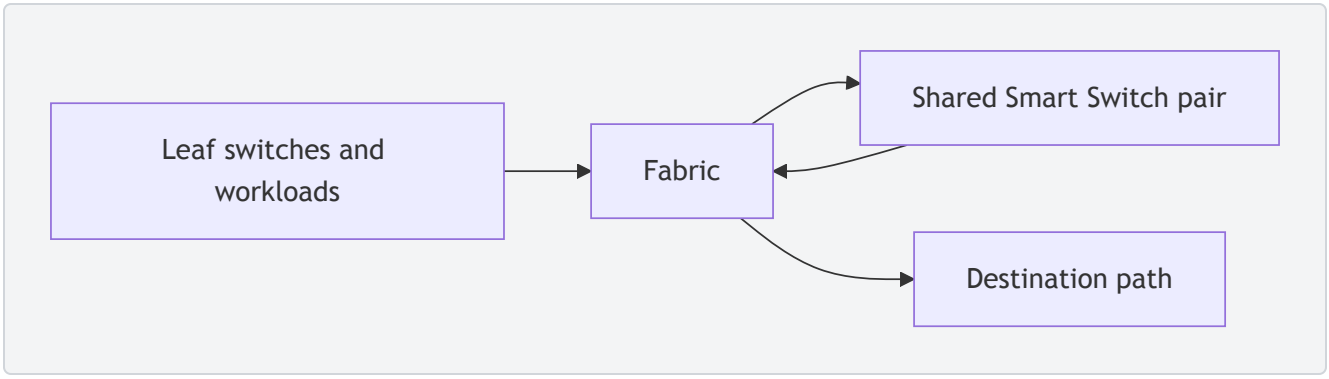
Enforcement capacity grows with the fabric: adding a protected rack adds a local enforcement point. This can be more predictable for environments where east-west traffic grows with workload count.

© 2026 Cisco and/or its affiliates. All rights reserved.

Design considerations

The Smart Switch must sit on the relevant traffic path. The surrounding network design must ensure that the traffic you want to inspect actually traverses the enforcement point. The model works best when workload locality and traffic patterns are well understood.

Services



Because traffic converges at the shared pair, inspection happens at a chosen point in the fabric rather than at every rack. This makes services a shared insertion model, with enforcement centralized relative to the workloads it protects.

When to use it

Services placement is usually the better fit when you want:

- shared enforcement for multiple traffic domains
- inspection at a natural convergence point in the network
- policy applied at inter-zone, inter-tier, or trust-boundary crossings
- a design in which enforcement is delivered as a network service

Scaling

Enforcement capacity is concentrated in the shared pair rather than distributed. This simplifies placement but means capacity does not scale automatically with every new rack.

Design considerations

© 2026 Cisco and/or its affiliates. All rights reserved.

Traffic does not reach the Smart Switch automatically; the surrounding network must steer relevant traffic through the shared pair. The model works best when trust boundaries and traffic convergence already align with a shared enforcement layer.

Choosing a placement

In practice, users usually choose a placement based on questions such as:

- Do you want enforcement close to each workload location or at a shared inspection point?
- Does your existing network design already provide a natural place for shared enforcement?
- Do you expect enforcement capacity to scale with each rack or with a smaller number of dedicated enforcement points?

The right placement depends on the environment. A good choice aligns policy intent with the network topology and with the way the organization wants to operate the platform.

© 2026 Cisco and/or its affiliates. All rights reserved.

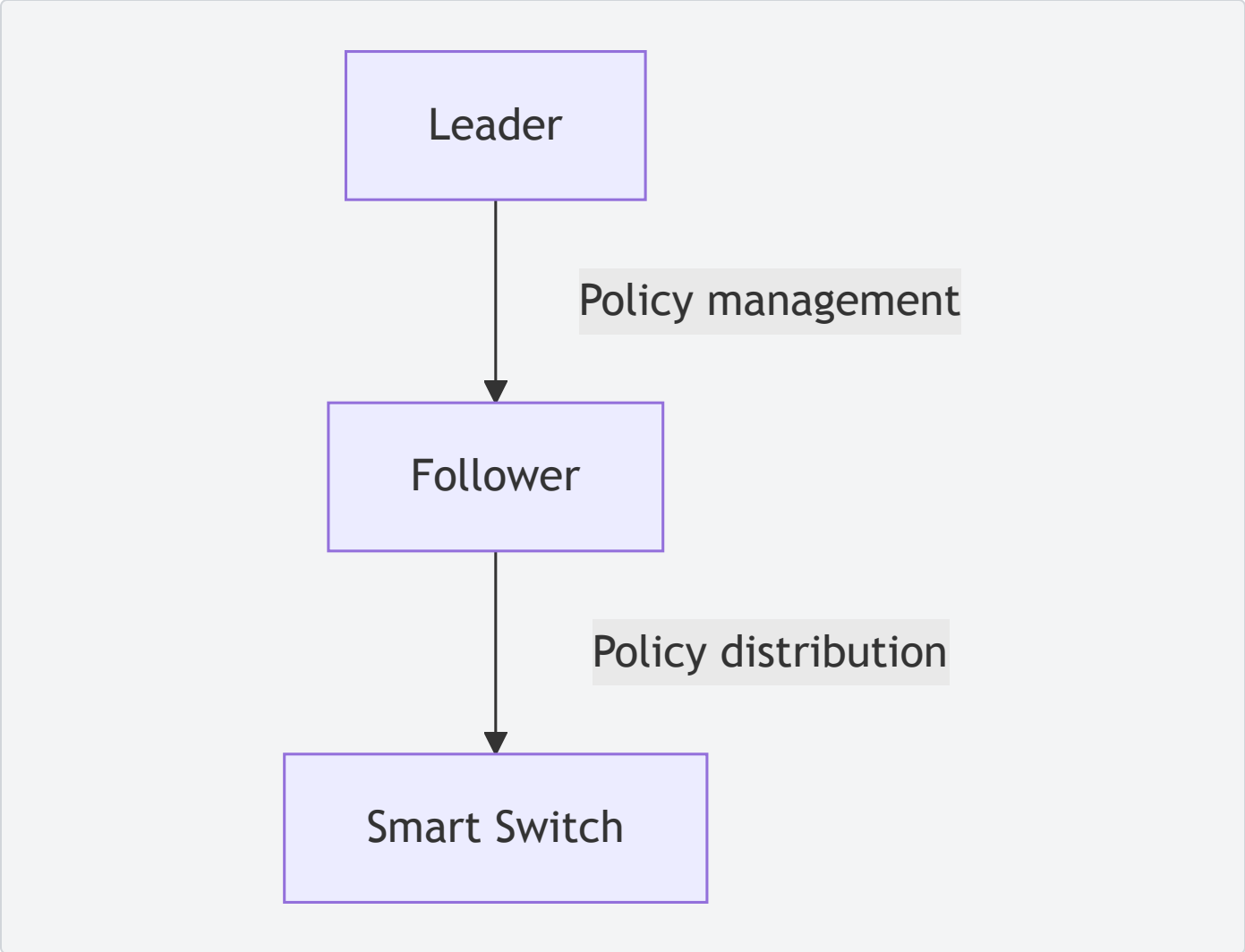
Smart Switch

A Smart Switch combines network switching capabilities with embedded Data Processing Units (DPUs). In addition to forwarding network traffic, it can inspect and enforce policies on selected traffic directly in the data path.

In Hypershield, the Smart Switch serves as a managed enforcement point. It inspects traffic traversing the switch and enforces centrally managed network policies.

> **Note**
>
> For the list of supported Smart Switch models and their hardware specifications, see Smart Switches in the reference doc set.

The following diagram shows how a Smart Switch receives security policies in a leader-follower deployment.



© 2026 Cisco and/or its affiliates. All rights reserved.

In a leader-follower deployment, each Smart Switch communicates with a single follower. The leader manages security policies, while followers distribute those policies to the Smart Switches they manage. Smart Switches do not communicate directly with the leader.

Each Smart Switch maintains a trusted connection with its follower to receive policy updates and report health, telemetry, and policy status.

For more information about registering, monitoring, updating, and migrating Smart Switches, see Manage Smart Switches.

Key components

Three hardware and software components work together within the Smart Switch to receive centrally managed policies and enforce them on network traffic.

Component	Role
Agent Gateway (AGW)	Coordinates policy deployment between the assigned follower and the DPUs.
Data Processing Unit (DPU)	Applies deployed security policies to network traffic.
Network Processing Unit (NPU)	Forwards network traffic through the switch.

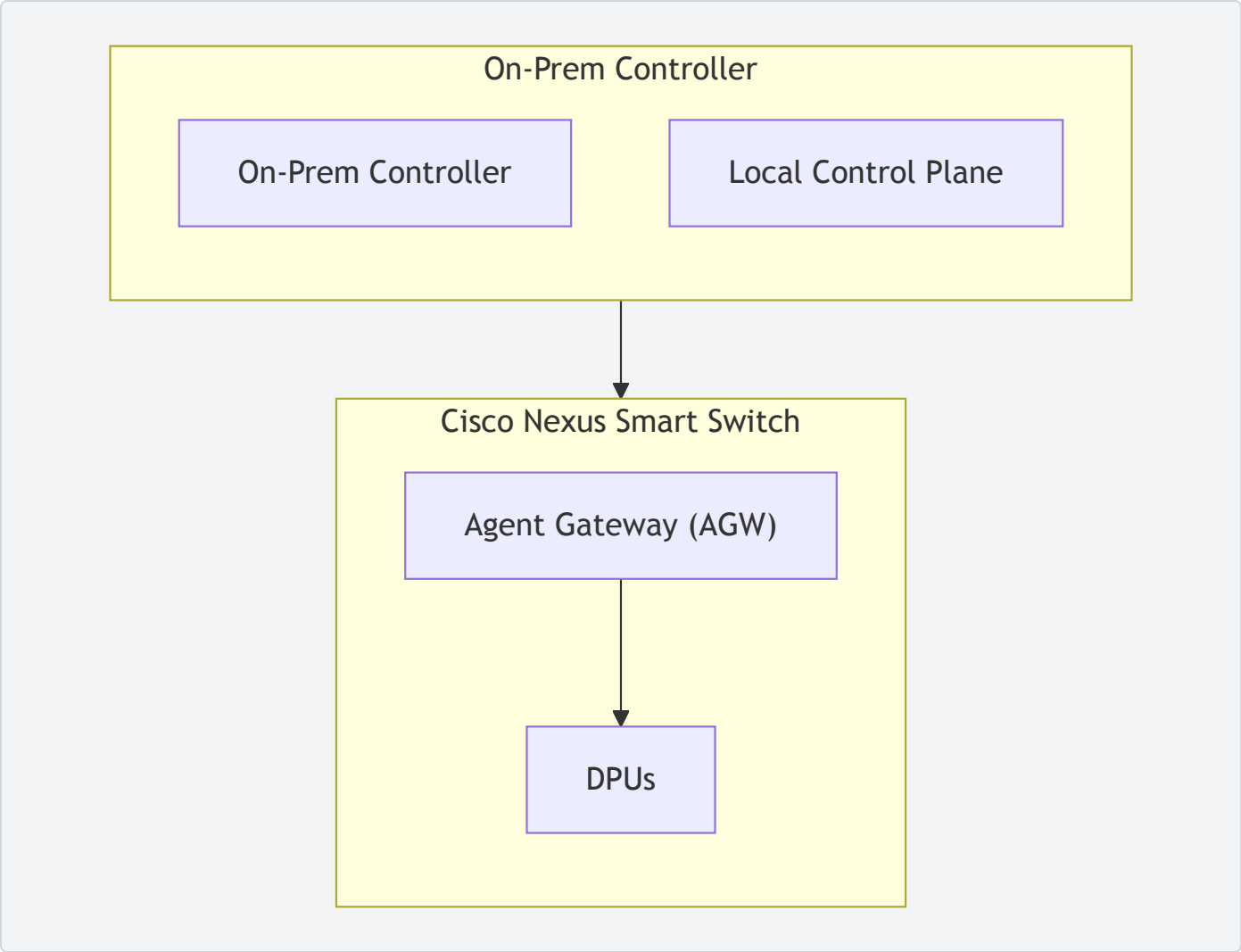
Data flows

The Smart Switch maintains separate flows for policy deployment and traffic processing. The deployment flow installs security policies on the switch; the enforcement flow applies those policies to network traffic.

Deployment

The AGW receives deployed policy from the assigned follower and coordinates policy programming on the DPUs. Once programmed, the DPUs hold the policy locally and continue enforcing it independently of follower connectivity.

© 2026 Cisco and/or its affiliates. All rights reserved.

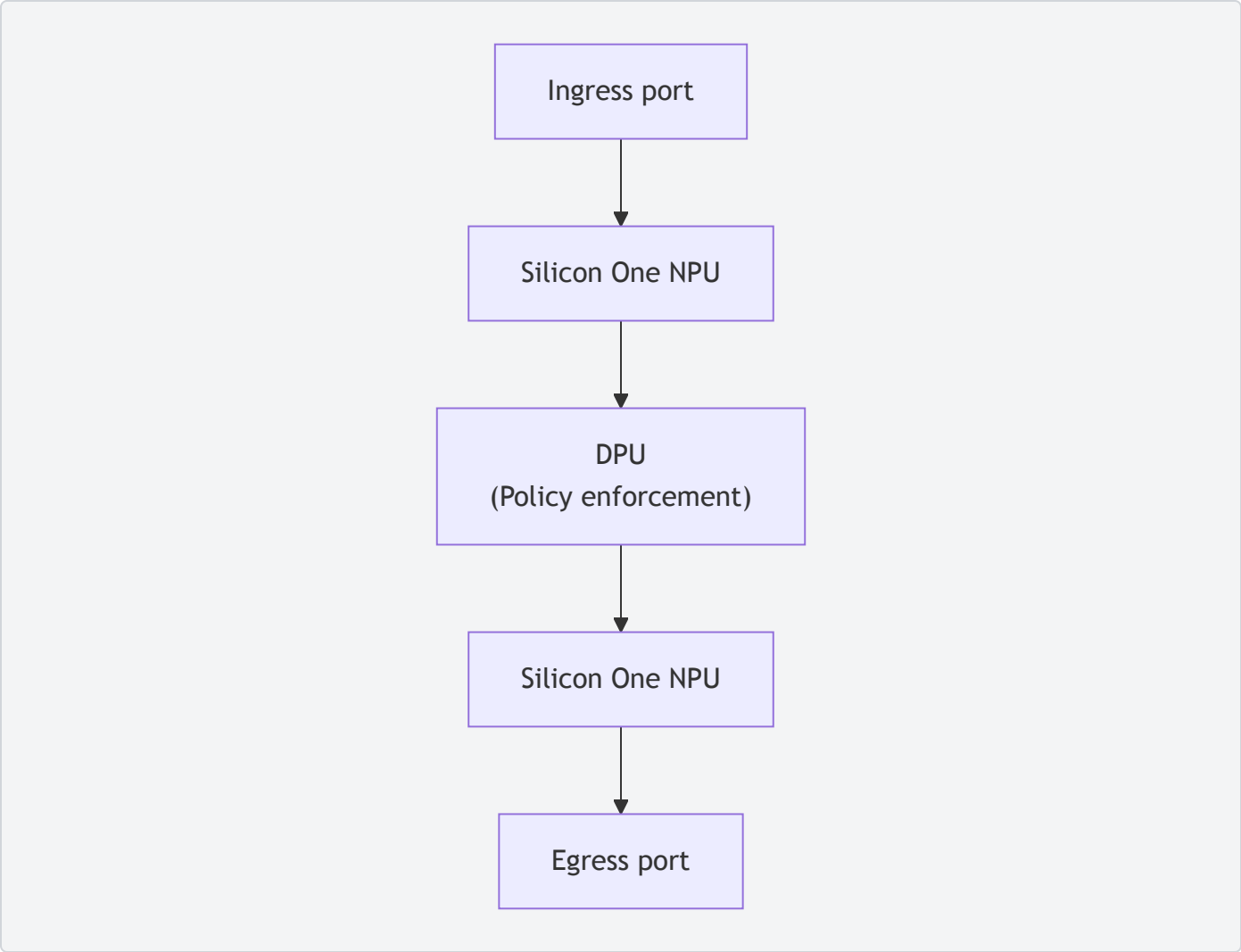


Enforcement

Traffic enters through the switch ports and is forwarded by the NPU. Traffic belonging to configured service VRFs is evaluated by the DPU against the locally installed policy before continuing through the forwarding path. Policy is evaluated within the context of the configured network topology, including constructs such as VRFs and VLANs.

Some traffic may traverse the DPU more than once depending on the forwarding scenario.

© 2026 Cisco and/or its affiliates. All rights reserved.



DPU traffic steering

DPU traffic steering is the NX-OS mechanism that redirects traffic from the NPU (the switch's normal forwarding path) to the DPU for inspection.

Policy is evaluated only for traffic that NX-OS steers to a DPU. Traffic that remains on the NPU forwarding path bypasses Hypershield policy evaluation. Steering configuration therefore determines the scope of enforcement, while policy definition determines how traffic within that scope is handled.

Dynamic and pinned

There are two DPU steering modes available:

- `dynamic module affinity`
- `pinned module affinity`

© 2026 Cisco and/or its affiliates. All rights reserved.

Mode `dynamic module affinity` has NX-OS assign each VRF to a specific DPU automatically based on the VRF name - the same VRF always lands on the same DPU, ensuring consistent stateful tracking without manual assignment.

Mode `pinned module affinity` is a legacy VRF steering mode where each VRF is manually assigned to a specific DPU module number.

> **Caution**
>
> Pinned mode is legacy and should only be used in consultation with your Cisco support representative.

Both supported switch placement options (Top-of-rack and Services) use the `dynamic module affinity` steering mode by default.

Policies: VRF-based vs. VLAN-based

Hypershield can match and enforce policy against traffic based on either VRF context and/or VLAN membership by defining the `SmartSwitchNetworkPolicy` object appropriately. VRF and VLAN designs have different scopes and implications:

Aspect	VRF-based policy	VLAN-based policy
scope	applies to all traffic within or between the named VRF(s)	applies to bridged Layer 2 traffic within the specified VLAN(s)
use case	inter-VLAN and inter-VRF segmentation; controlling routed traffic	intra-VLAN lateral movement control within a single broadcast domain
policy rule field	`vrf:` in source/destination	`vlan:` in source/destination
typical deployment	production/management VRF separation; tenant isolation	micro-segmentation within a server VLAN

Most production deployments will use VRF-based policy as the primary mechanism, with VLAN-based rules added where intra-VLAN lateral movement needs to be controlled. The two can also be combined (see Concurrent VRF and VLAN traffic below).

Steering: VRF-based vs. VLAN-based

The NX-OS `service firewall` setting controls which traffic is steered to the DPU. This is separate from policy definition.

- `vrf <name> module-affinity dynamic` - steers all routed traffic in the named VRF to the DPU. This is used for any VRF whose inter-VLAN or inter-VRF traffic you want inspected.

© 2026 Cisco and/or its affiliates. All rights reserved.

- `vlan id <range> bridged-traffic module-affinity dynamic` - steers Layer 2 bridged traffic within those VLANs to the DPU. Required if you want to enforce intra-VLAN policy (traffic that never crosses a VRF boundary and therefore would not be steered by a VRF entry).

Traffic in VRFs or VLANs not listed in `service firewall` bypasses the DPU entirely (it is forwarded normally by the NPU) - no policy is evaluated for it, regardless of what is configured in Timescape. This allows you to scope enforcement precisely.

Important

Steering a VRF captures inter-VLAN routed traffic within that VRF but does not capture bridged traffic that stays within a single VLAN inside that VRF. To enforce policy on intra-VLAN traffic, you must also add the relevant `vlan id` entries.

Concurrent VRF and VLAN traffic

A single source host can generate both routed (VRF) and bridged traffic (VLAN) simultaneously. When both are configured in `service firewall`, the DPU handles them in the correct context automatically:

- Traffic that is routed (crosses a subnet boundary via the SVI, whether inter-VLAN or inter-VRF) is steered through the VRF context and evaluated against VRF-scoped policy rules.
- Traffic that is bridged (stays within the same VLAN, never hits a router interface) is steered through the VLAN context and evaluated against VLAN-scoped policy rules.

This means you can enforce both inter-VLAN segmentation and intra-VLAN micro-segmentation on the same switch, for the same hosts, without conflict. The DPU distinguishes the two traffic types based on whether the NPU classified the packet as routed or bridged before steering it.

VRF management traffic

The assigned follower manages the switch by pushing policy to it. The VRF that carries this control-plane traffic can be the same as a data/workload VRF (for example, the default VRF) or a dedicated one.

When management traffic shares a VRF with data traffic and that VRF is steered to the DPU, the DPU's default-deny rule applies to the follower's own connection to the switch. This may result in a self-locking condition: the follower can no longer reach the switch to push the very policy that would unblock it.

Caution

Recovering from the self-locking condition requires console access to the switch.

Keeping management and data traffic in separate VRFs avoids the above class of problem by design, since the management VRF can be omitted from steering entirely.

© 2026 Cisco and/or its affiliates. All rights reserved.

If a shared VRF is used, appropriate allow rules are necessary to preserve reachability for the follower-to-switch channel and for related management flows such as syslog and IPFIX export, DNS, and any traffic to Timescape or the OIDC identity provider (such as Keycloak).

High availability

The Smart Switch continues enforcing previously deployed security policies during temporary management connectivity failures and supported network redundancy events.

New or updated policy can be deployed after follower connectivity is restored.

When deployed using supported network redundancy mechanisms, such as Virtual PortChannel (vPC), policy enforcement continues across the remaining forwarding path during a switch failure, provided the underlying network redundancy remains operational.

Note

For the procedure to enable high availability on a Smart Switch pair, see Enable high availability on a Smart Switch.

© 2026 Cisco and/or its affiliates. All rights reserved.

Timescape and SCC

Timescape is the management-plane component of Hypershield. It runs on the leader cluster with a coordinating instance on each follower, and is administered through a web-based dashboard. Optionally, Cisco Security Cloud Control (SCC) can front Timescape as the management interface (see Operating modes).

You can use the dashboard to perform the following operations:

- manage the leader and follower Kubernetes clusters
- work with network objects and network object groups
- create, change, and remove security policies
- manage logging and telemetry features
- review traffic visibility and security events
- automate supported network object and policy operations through the public REST API

Deployment topology

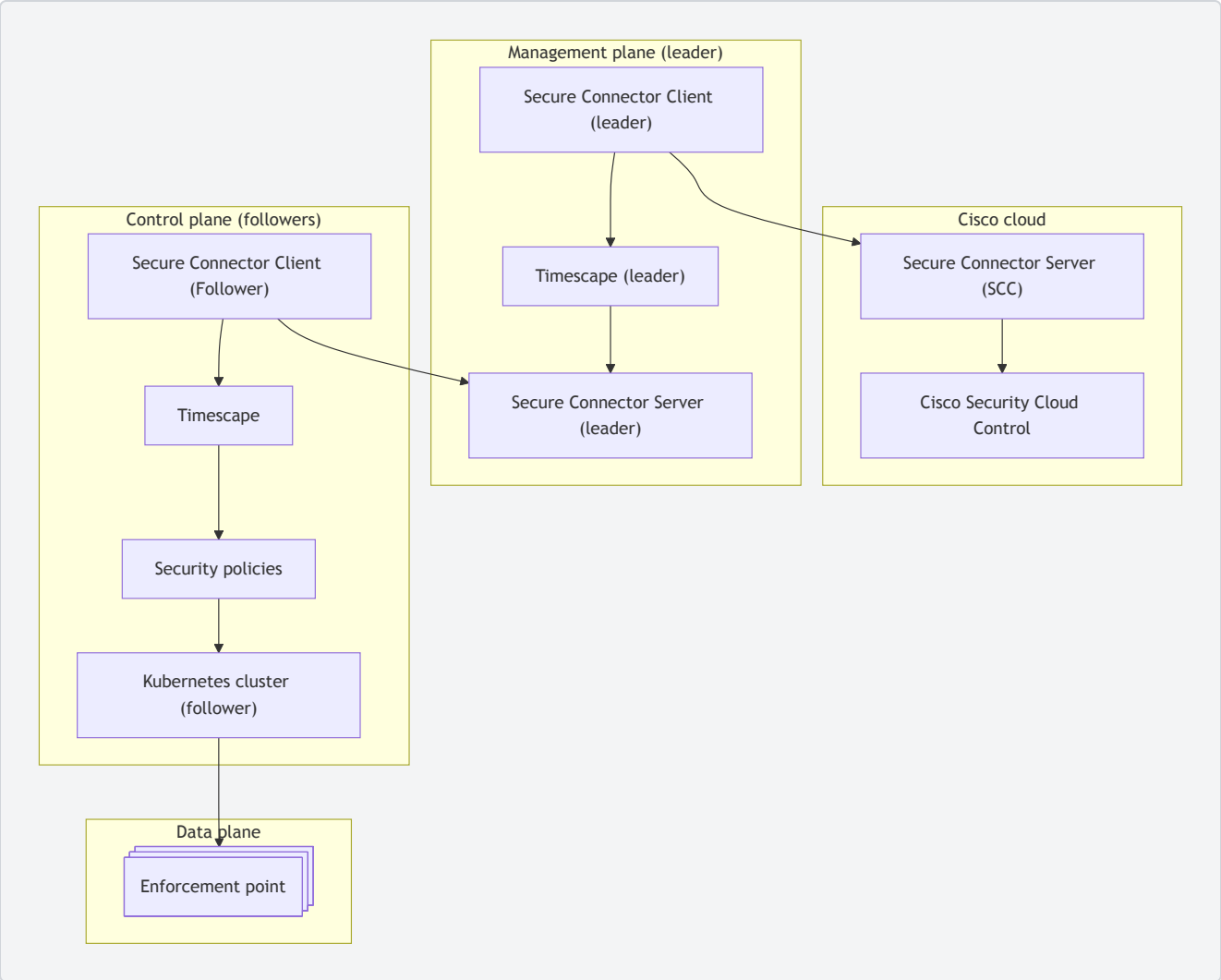
Timescape spans the leader and follower clusters. The leader hosts the primary Timescape instance where administrators define and approve policy; each follower runs a coordinating Timescape instance that receives approved policy from the leader and materializes it as security policy objects in the follower's Kubernetes cluster, which are then applied at the enforcement points.

Connectivity between clusters uses the Secure Connector, a client-server pair in which the client always initiates the connection. Two independent pairs are involved:

- Follower to leader. Each follower runs a Secure Connector client that connects to a Secure Connector server on the leader. This link carries coordination traffic between the follower's Timescape and the leader's Timescape.
- Leader to SCC (connected mode only). The leader runs a Secure Connector client that connects to a Secure Connector server in the Cisco cloud, enabling Cisco Security Cloud Control to front the leader's Timescape.

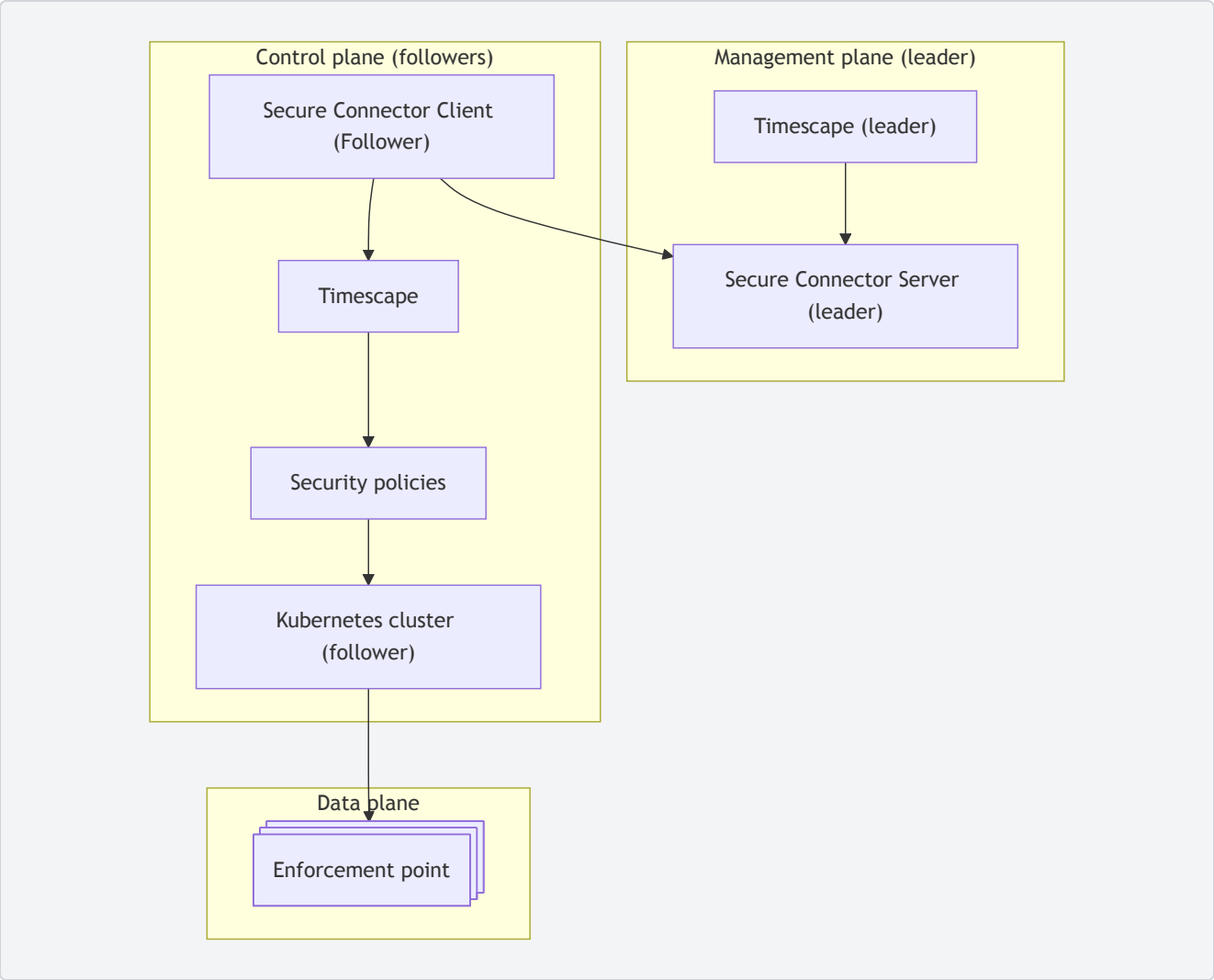
Connected mode

© 2026 Cisco and/or its affiliates. All rights reserved.



Disconnected mode

© 2026 Cisco and/or its affiliates. All rights reserved.



Access

Timescape is not exposed by default. When users or API clients need access, you, the administrator, expose Timescape by configuring ingress, TLS, and a deployment-specific URL for the web UI and API.

For authentication, Timescape supports OpenID Connect (OIDC), so authentication can be delegated to an existing identity provider. Combined with role-based access control, this determines what different users can view or change. OIDC also authenticates public REST API clients when API RBAC is enabled (see Public REST API).

User interface paths

You reach the user interface through one of two paths, one for each operating mode (see Operating modes):

- SCC (connected mode) - Timescape is embedded within the Cisco Security Cloud Control platform.

© 2026 Cisco and/or its affiliates. All rights reserved.

- directly (disconnected mode) - Timescape is accessed without going through SCC, with Isovalent Timescape branding.

SCC leader registration

The two paths expose the same Hypershield features but differ in one important way during initial setup. On the direct path, all features are available immediately.

On the SCC path, an existing on-prem leader cluster must be registered with SCC before any other feature becomes available. Until a leader is registered, the UI redirects all navigation to the Leaders page. For the registration procedure, see Register a leader cluster with SCC.

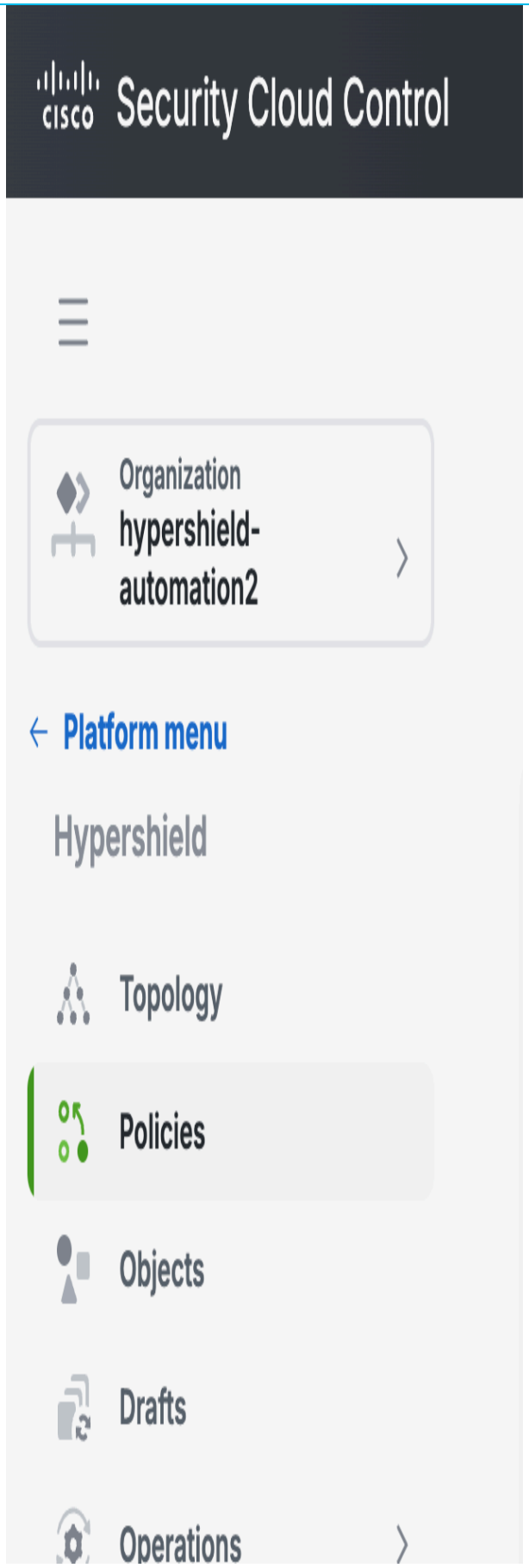
Web UI layout

The layout of the Timescape sidebar also differs between the two paths. With SCC, Hypershield appears under the Platform menu alongside the other SCC platform services. When accessing directly, Timescape presents its own sidebar grouped by function.

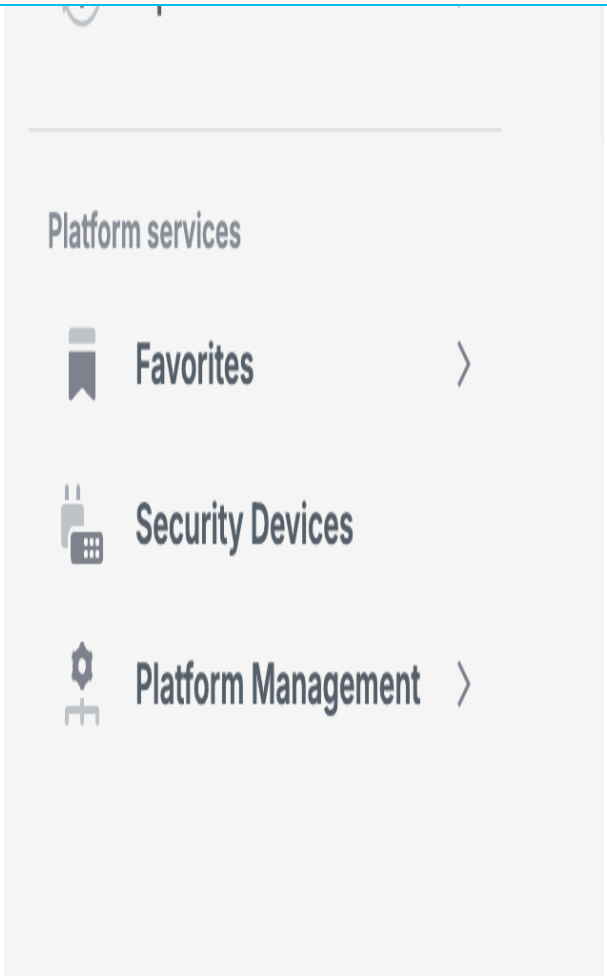
SCC (connected)

© 2026 Cisco and/or its affiliates. All rights reserved.

© 2026 Cisco and/or its affiliates. All rights reserved.



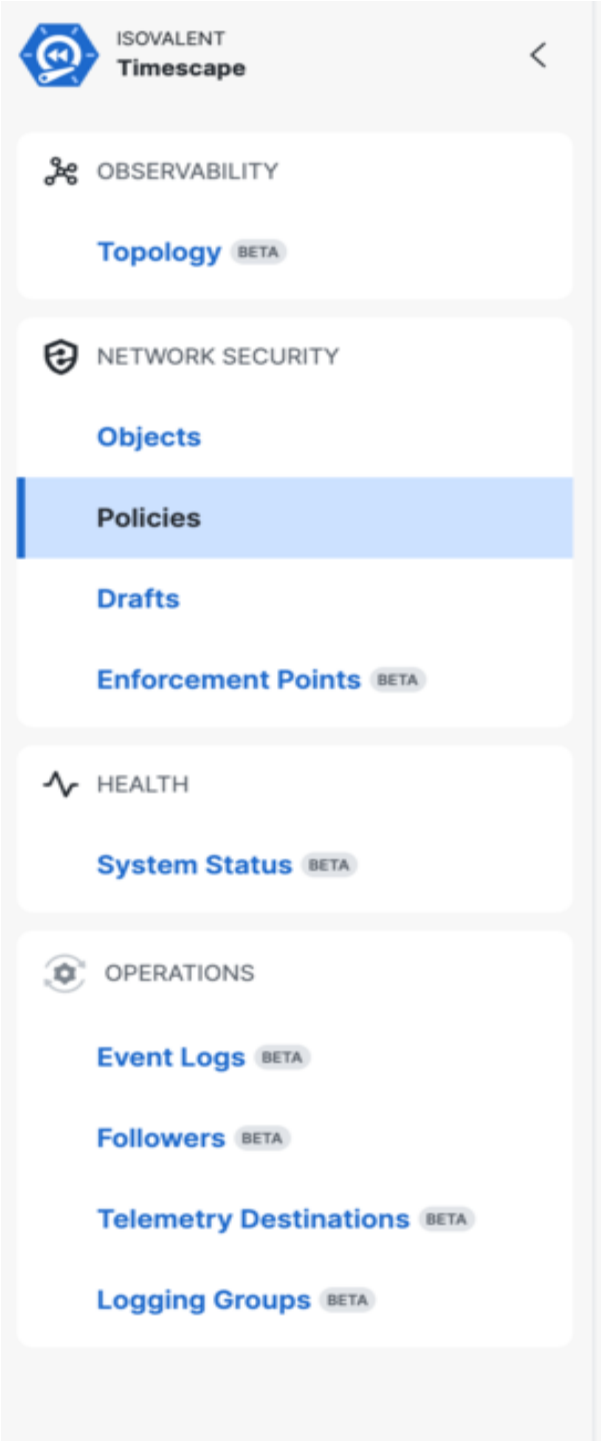
© 2026 Cisco and/or its affiliates. All rights reserved.



Timescape sidebar on the SCC path.

Direct (disconnected)

© 2026 Cisco and/or its affiliates. All rights reserved.



Timescape sidebar on the direct path.

© 2026 Cisco and/or its affiliates. All rights reserved.

Public REST API

The versioned public REST API supports customer automation for network object and security policy lifecycle operations. See Timescape API usage to configure access, then follow Manage network objects or Manage security policies for common workflows.

The public API records desired configuration. A successful policy mutation does not by itself confirm that every enforcement point has applied the change. Use policy deployment status to inspect the current state reported by enforcement agents, not as proof that they applied the desired revision. For the definitive endpoint and response contract, see the Timescape public API reference.

> **Note**
>
> To enable OIDC authentication and RBAC on the public REST API, see Public API RBAC.

© 2026 Cisco and/or its affiliates. All rights reserved.

Platform access control

Platform access control governs how users, API clients, and platform components authenticate to Hypershield and what they are allowed to do once authenticated. It covers identity, authentication methods such as OIDC, role-based access control (RBAC) for authorization, and the service identities that platform components use to communicate with each other.

This topic is about management access only. For workload identity used in security policy rules, see Security policies instead.

Management paths

The identity and access model applies across different management paths:

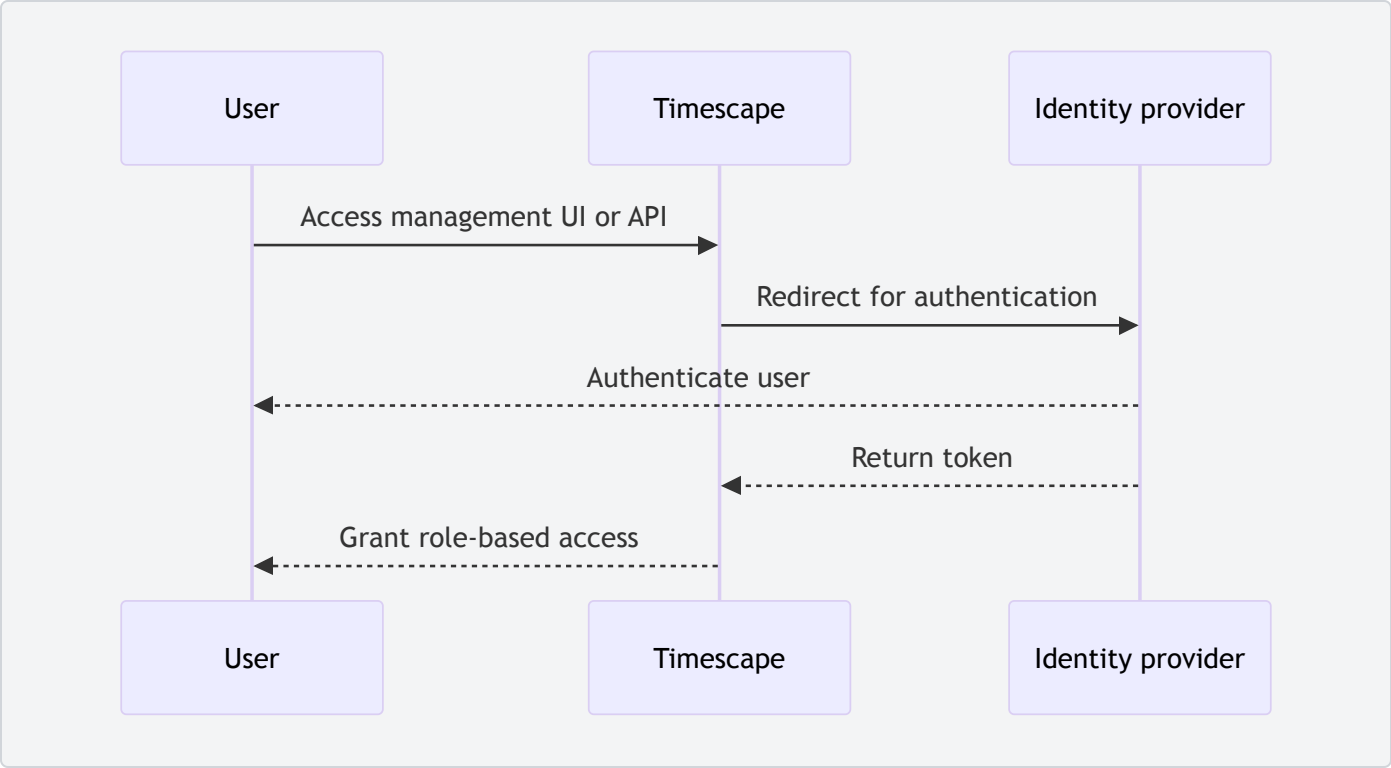
- user access to local management surfaces such as Timescape
- user access to centralized management surfaces such as Cisco Security Cloud Control (SCC)
- API-driven access for automation and integrations
- service-to-service access used by platform components

Each path defines which management boundary a user or system is crossing and which part of the platform is authoritative for that action.

Local management surfaces and OIDC

For a local management path, Timescape supports OpenID Connect (OIDC) so that authentication can be delegated to an external identity provider such as Keycloak.

© 2026 Cisco and/or its affiliates. All rights reserved.



This approach lets Hypershield integrate with an existing enterprise identity model and still use the same authentication framework across deployments.

Authorization and roles

For authentication and authorization, the platform needs to know:

- which identity is making the request
- which management surface or client the session is intended for
- which role or permission set applies to that identity

Management operations often span different responsibility levels. The RBAC approach separates those responsibilities and enables least-privilege operation. Tasks that demand varying levels of authorization include:

- manage platform settings
- create or change policy
- review objects, inventory, and status
- observe changes without modifying the environment

API access

The same identity model applies to APIs used for automation. API access still depends on authenticated sessions or tokens and the permissions associated with the calling identity. API-driven workflows need:

© 2026 Cisco and/or its affiliates. All rights reserved.

- a valid token or authenticated session
- the correct access context for the service being used
- a role that permits the requested operation

Hypershield exposes APIs on several management surfaces:

- the Kubernetes API server on the on-prem controller, which customers can access directly with `kubectl` for cluster-level operations
- the Public REST API for object and policy lifecycle operations
- the Cisco Security Cloud Control (SCC) API for centralized management

Kubernetes API vs. Timescape API

Although the Kubernetes API and the Timescape API are both available, they are not interchangeable. Timescape coordinates changes across the deployment, so bypassing it with `kubectl` can lead to an inconsistent state.

As a general guide:

- use the Timescape API for security policy and network object lifecycle operations (see Kubernetes implementation for the specific prohibition on direct Kubernetes API changes)
- use the Kubernetes API for read-only inspection when troubleshooting

Component and service identities

Platform components also need trusted identities when they communicate with other services. For example, local services, connectors, and enforcement-side components may need authenticated access to APIs, object stores, or management endpoints.

© 2026 Cisco and/or its affiliates. All rights reserved.

Logging and observability

Hypershield exposes operational visibility through several distinct streams: syslog, IPFIX, and the event history recorded by Timescape. Each stream has a specific purpose and a specific origin in the platform.

This page describes what each stream is for, where in the platform it originates, and how the streams relate to one another.

Visibility streams

Hypershield provides three primary visibility streams. They are complementary rather than interchangeable: a given operational question is usually best answered by one of them.

- Syslog

  Per-flow permit and deny records, exported from enforcement points to external collectors. Syslog is the primary input for SIEM and log-analysis platforms.

- IPFIX

  Connection-oriented flow records, exported from enforcement points to external collectors. IPFIX is used for traffic reporting and trend analysis rather than individual permit or deny decisions.

- Timescape event history

  A record of management activity - who created, modified, deployed, or removed a security policy, network object, or other managed resource. Timescape event history is reviewed in the Timescape UI (see Timescape and SCC) rather than exported.

Origins

The streams originate in different parts of the platform, which affects how they are configured and what they can report on:

- Syslog and IPFIX originate on the data plane. Enforcement points generate the records as they process traffic and export them directly to the configured collectors.
- Timescape event history originates on the management plane. It reflects administrative intent and deployment state, not live traffic.

Because syslog and IPFIX are exported to external infrastructure, collector reachability affects whether records can be delivered. Timescape event history is part of the management surface itself and is available wherever Timescape is available.

© 2026 Cisco and/or its affiliates. All rights reserved.

Related workflows

To configure syslog and IPFIX export, see Manage telemetry destinations and Manage logging groups.

© 2026 Cisco and/or its affiliates. All rights reserved.

Installations

© 2026 Cisco and/or its affiliates. All rights reserved.

Install Hypershield on VMware

This multi-part procedure shows how to install Hypershield on a VMware vCenter environment using a bootstrap process that creates Leader and Follower Kubernetes management clusters.

© 2026 Cisco and/or its affiliates. All rights reserved.

Satisfy the prerequisites

Before you install on VMware, confirm that your environment meets the infrastructure, capacity, and credential requirements described on this page. The installation uses a Leader/Follower cluster architecture and the bootstrap process provisions both clusters from a shared installer virtual machine (the management VM). Taking the time to verify each requirement now prevents interruptions during the multi-step installation procedure.

Gather the prerequisites

SSH key pair

You need an SSH key pair to authenticate with the management VM and the cluster nodes. Generate one if you do not already have a suitable key:

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

vCenter requirements

> **Warning**
>
> Hypershield requires vCenter Server 8.0 Update 3 or later. vCenter Server 7.x reached End of General Support as part of VMware vSphere 7.0 on October 2, 2025 and is not supported by Hypershield. Upgrade vCenter before you begin the installation. See the Broadcom End of General Support announcement.

Confirm that your vCenter environment meets the following requirements before proceeding:

- vCenter Server 8.0 Update 3 or later
- ESXi hosts with sufficient resources (see capacity planning below)
- VMFS or vSAN datastore with adequate free space
- Distributed Virtual Switch (DVS) configured

Minimum VMware and IP address capacity

The tables in this section assume `mgmtCluster.vmProfile: hypershield`, three control-plane nodes, two worker nodes, one leader cluster, and one follower cluster. The values are minimum allocations for planning purposes. Production capacity must also account for workload growth, persistent data, failure recovery, and the vSphere resource policy. One installer VM can deploy both clusters and is shown separately.

© 2026 Cisco and/or its affiliates. All rights reserved.

VM and resource requirements

Scope	VMs	Allocated vCPU	Allocated memory	Recommended datastore capacity
Leader management cluster: 3 control-plane + 2 worker	5	56	112 GiB	600 GiB
Follower management cluster: 3 control-plane + 2 worker	5	56	112 GiB	600 GiB
Shared installer VM	1	4	8 GiB	60 GiB thin-provisioned boot disk
Minimum combined deployment	**11**	**116**	**232 GiB**	**1.2 TiB plus installer boot disk**

Note

Each control-plane node uses 8 vCPU and 16 GiB RAM. Each Hypershield worker uses 16 vCPU and 32 GiB RAM. If separate installer VMs are retained for the leader and follower, add one VM, 4 vCPU, 8 GiB RAM, one 60 GiB thin-provisioned boot disk, and one primary-network IP address.

IP address requirements

Role and network	VM IPs	API VIP	Harbor VIP	Timescape Gateway VIP	Secure Connector VIP	Timescape ingester VIP	Minimum IPs	Recommended spare node IPs
Shared installer, primary network	1	0	0	0	0	0	1	0
Leader, primary network	5	1	1	1	1	0	9	1
Follower, primary network	5	1	1	0	0	0	7	1
Follower, SmartSwitch network	5	1	0	0	0	1	7	1
Minimum combined deployment	**16**	**3**	**2**	**1**	**1**	**1**	**24**	**3**

© 2026 Cisco and/or its affiliates. All rights reserved.

Tip

- The minimum combined total does not include the optional follower CLC, which requires one additional VIP on the SmartSwitch network.
- A single-network follower requires five node IPs, one primary API VIP, one Harbor VIP, and one Timescape ingester VIP on its primary network, for a minimum of eight follower addresses. It does not require a secondary API VIP.
- Reserve the recommended spare address on every node network to allow node replacement or scaling without expanding the IPAM pool.

Credentials required

You need vCenter administrator credentials to allow the bootstrap process to provision and manage VMs in your vSphere environment.

Verify the prerequisites

Before moving on to the installation steps, double-check the following:

- The SSH key pair exists and the public key is accessible.
- vCenter Server version is 8.0 Update 3 or later.
- ESXi hosts have at least 116 vCPU, 232 GiB RAM, and 1.2 TiB of datastore capacity available for the combined deployment.
- A Distributed Virtual Switch is configured.
- At least 24 IP addresses are reserved across the required networks (plus three recommended spare addresses).
- vCenter administrator credentials are available.

Next steps

The prerequisites are satisfied. Next, Deploy the bootstrap OVA.

© 2026 Cisco and/or its affiliates. All rights reserved.

Deploy the bootstrap OVA

This page shows how to download and deploy the Hypershield bootstrap OVA to your VMware vCenter environment. The resulting management VM serves as the installer host from which all subsequent installation steps are run.

All values shown are examples. Adjust them to match your environment.

> **Note**
>
> All steps on this page are performed from your local workstation unless otherwise noted.

Download the bootstrap OVA

Download the bootstrap OVA from the Cisco Software Download site:

1. Go to the Hypershield Bootstrap OVA download page and log in with your Cisco.com credentials.
2. Download the latest `hypershield-bootstrap-*.ova` file.

Deploy the OVA

Choose the deployment method that best suits your environment.

VMware easy mode (preferred)

1. Log in to vSphere Client at `https://<vcenter-ip>/ui`.
2. Right-click your datacenter or host and select **Deploy OVF Template**.
3. Browse to the downloaded OVA file or provide the URL of the HTTP server hosting the OVA file.
4. Configure the deployment:
   - **Name**: `hypershield-bootstrap`
   - **Folder**: select the destination folder
   - **Compute Resource**: select the cluster or host
   - **Storage**: select the datastore
   - **Network**: select the VM network (no DHCP)
5. In the **Customize template** section, under **Cloud-Init**:
   - Provide the hostname of the VM according to your network configuration.
   - Optionally add your public SSH key in the **SSH public keys** field. To display your public key:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
cat ~/.ssh/id_ed25519.pub
```

6. Under **Networking**, provide all network configuration details:
   - IP address
   - Subnet mask
   - Gateway
   - DNS servers (comma-separated)
   - NTP servers (comma-separated)
   - Proxy configuration (if applicable)
7. Click **Finish** to deploy.
8. Power on the VM and wait 2–3 minutes for cloud-init to complete.

Scripted

The scripted method uses the `import-ova.sh` helper script together with `metadata.yaml` and, optionally, `userdata.yaml`.

Before continuing, download the vSphere downloadable artifacts ZIP file and extract it. The following steps identify the exact extracted filename to use.

Prepare the metadata file

1. In the extracted `hypershield-downloadable-artifacts` folder, use `metadata.yaml` and `metadata.yaml.sha256`.
2. Verify the integrity of `metadata.yaml`:

   ```
   sha256sum -c metadata.yaml.sha256
   ```

   Expected output:

   ```
   metadata.yaml: OK
   ```

3. Open `metadata.yaml` and update all values marked with `# MODIFY THIS LINE` to match your IP configuration.

Prepare the userdata file (proxy environments only)

If you require a proxy configuration, complete the following steps:

© 2026 Cisco and/or its affiliates. All rights reserved.

In the extracted `hypershield-downloadable-artifacts` folder, use `userdata.yaml` and `userdata.yaml.sha256`. Verify the integrity of `userdata.yaml`:

```
sha256sum -c userdata.yaml.sha256
```

Expected output:

```
userdata.yaml: OK
```

Open `userdata.yaml` and update the proxy settings.

Prepare the import script

1. In the extracted `hypershield-downloadable-artifacts` folder, use `import-ova.sh` and `import-ova.sh.sha256`.
2. Verify the integrity of `import-ova.sh`:

   ```
   sha256sum -c import-ova.sh.sha256
   ```

   Expected output:

   ```
   import-ova.sh: OK
   ```

3. Make the script executable:

   ```
   chmod +x import-ova.sh
   ```

Configure the GOVC environment

Export the following environment variables to authenticate with vCenter:

```
export GOVC_URL='vcenter.example.com'
export GOVC_USERNAME='administrator@vsphere.local'
export GOVC_PASSWORD='your-password'
export GOVC_DATACENTER='Datacenter1'
export GOVC_DATASTORE='datastore1'
export GOVC_NETWORK='VM Network'
export GOVC_RESOURCE_POOL='*/Resources'
export GOVC_INSECURE=true   # Use with self-signed certificates
export GOVC_FOLDER="vm/hypershield"
```

Run the import

© 2026 Cisco and/or its affiliates. All rights reserved.

Basic usage:

```
./import-ova.sh hypershield-bootstrap-10.0.0.0.ova
```

Advanced usage with explicit network and SSH key:

```
./import-ova.sh \
  --network "vlan1357-access" \
  --ssh-key $HOME/.ssh/id_ed25519.pub \
  --name hsinstaller-ca2 \
  hypershield-bootstrap-10.0.0.0.ova
```

To see all available options:

```
./import-ova.sh -h
```

Sample output:

```
usage: ./import-ova.sh [options] <ova_path>

Options:
  --check-duplicate-ip    Enable duplicate IP conflict check
  -d, --debug             Enable debug messages
  -m, --metadata PATH     Path to OVA metadata.yaml
  -n, --name NAME         Name for deployed bootstrap VM
      --network NAME      VM Network to attach to
  -s, --ssh-key PATH      Path to SSH public key for access
  -u, --userdata PATH     Path to OVA userdata.yaml
  -h, --help              Show this help message

Arguments:
  ova_path                Path to OVA image to import
```

VMware advanced mode

1. Log in to vSphere Client at `https://<vcenter-ip>/ui`.
2. Right-click your datacenter or host and select **Deploy OVF Template**.
3. Browse to the downloaded OVA file or provide the URL of the HTTP server hosting the OVA file.
4. Configure the deployment:
   - **Name**: `hypershield-bootstrap`
   - **Folder**: select the destination folder
   - **Compute Resource**: select the cluster or host
   - **Storage**: select the datastore

© 2026 Cisco and/or its affiliates. All rights reserved.

- **Network**: select the VM network (no DHCP)

5. In the **Customize template** section:
   - Specify the hostname of the VM according to your naming convention.
   - Add your public SSH key in the **SSH public keys** field.
6. To configure the VM IP address directly, convert the contents of `metadata.yaml` to base64 and paste the value into **Base64-encoded network-config**:

```
cat metadata.yaml | base64 -w0
```

7. To configure the VM proxy directly, convert the contents of `userdata.yaml` to base64 and paste the value into **Base64-encoded user-data**:

```
cat userdata.yaml | base64 -w0
```

8. Click **Finish** to deploy.
9. Power on the VM and wait 2–3 minutes for cloud-init to complete.

Access the management VM

Tip

The default user is `iep` with password `iep`.

1. If you are using SSH agent, add your SSH key:

```
ssh-add ~/.ssh/id_ed25519
```

2. SSH to the management VM:

```
ssh -A iep@<management-vm-ip>
```

3. You are prompted to change the password on first login:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
WARNING: Your password has expired.
You must change your password now and login again!
Changing password for iep.
Current password:
New password:
Retype new password:
passwd: password updated successfully
Connection to 198.19.184.89 closed.
```

4. After the password is updated you are logged out. Reconnect with the new password.

Verify the management VM is accessible

After reconnecting, confirm that you can log in successfully and that the working environment is available:

```
ls /home/iep/
```

You should see the `bootstrap.yaml` template file and the `/opt/installer/` directory containing the node OVA and image archives.

Next steps

The management VM is deployed and accessible. Next, Configure and install Leader and Follower clusters.

© 2026 Cisco and/or its affiliates. All rights reserved.

Configure and install Leader and Follower clusters

This page shows how to prepare the configuration files for the Leader and Follower management clusters and then bootstrap both clusters using `isoctl`.

> **Note**
>
> All steps on this page are performed on the management VM unless otherwise noted.

Prepare the working directory

Create the working directories for configuration files and follower credentials:

```
mkdir -p /home/iep/hypershield
mkdir -p /home/iep/follower-credentials
cd /home/iep/hypershield
```

Configure the Leader bootstrap

Copy the bootstrap template and open it for editing:

```
cp /home/iep/bootstrap.yaml /home/iep/hypershield/bootstrap_leader.yaml
```

The following is an example of all available options that can be place in `bootstrap_leader.yaml`. Update all values to match your environment:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
# HTTP/HTTPS proxy for outbound connections (optional)
# Required if direct internet access is not available
# proxySettings:
#   http:
#     host: "http://proxy.example.com"
#     port: "80"
#   https:
#     host: "http://proxy.example.com"
#     port: "443"
#   noProxy:
#     - "localhost"
#     - ".example.com"

# Additional CA certificates (optional)
# caCerts:
#   files:
#     - hostPath: "/home/iep/custom-ca.crt"
#       name: "custom-CA"

kindNodeImage: "kindest/node:latest"

mgmtCluster:
  name: "hypershield-leader"
  vmProfile: hypershield
  gatewayAPI: true
  apiServer:
    address: "198.19.184.10"
    certSANs: []
  controlPlaneReplicas: 3
  workerReplicas: 2
  # podCIDR: "10.20.0.0/16"
  # serviceCIDR: "10.30.0.0/16"
  nodeAuthorizedKeys: []
  nodeAuthorizedKeysFromAgent: true
  imageArchives:
    - "/opt/installer/images.tar.gz"

registry:
  address: "198.19.184.11"
  localImageArchives:
    - "/opt/installer/images.tar.gz"

vsphere:
  endpoint: "vcenter.example.com"
  thumbprint: "AA:BB:CC:DD:EE:FF:..."
  datacenter: "Datacenter1"
  datastore: "datastore1"
  folder: "Hypershield-Leader"
  network: "VM Network"
  resourcePool: "*/Resources"
  template: "clusterX-node-template"
  ova: "/opt/installer/node.ova"
  username: "administrator@vsphere.local"
  password: "YourSecurePassword"

networks:
  - name: "VM Network"
    ipam:
      name: "leader-primary"
      # These addresses must not overlap with the apiServer or registry addresses
      addresses:
        - 198.19.184.12-198.19.184.20
      prefix: 24
      gateway: 198.19.184.1
      nameservers:
        - 8.8.8.8
        - 1.1.1.1
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Configure the Follower bootstrap

Copy the bootstrap template and open it for editing:

```
cp /home/iep/bootstrap.yaml /home/iep/hypershield/bootstrap_follower.yaml
```

Note

The main differences between the Leader and Follower bootstrap configurations are the cluster name, IP addresses, and network configuration. The Follower may also include a SmartSwitch network for dual-leg designs.

The following is an example of all available options that can be place in `bootstrap_follower.yaml`. Update all values to match your environment:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
# HTTP/HTTPS proxy for outbound connections (optional)
# proxySettings:
#   http:
#     host: "http://proxy.example.com"
#     port: "80"
#   https:
#     host: "http://proxy.example.com"
#     port: "443"
#   noProxy:
#     - "localhost"
#     - ".example.com"

# Additional CA certificates (optional)
# caCerts:
#   files:
#     - hostPath: "/home/iep/custom-ca.crt"
#       name: "custom-CA"

kindNodeImage: "kindest/node:latest"

mgmtCluster:
  name: "hypershield-follower"
  vmProfile: hypershield
  gatewayAPI: true
  apiServer:
    address: "198.19.184.30"
    certSANs: []
  controlPlaneReplicas: 3
  workerReplicas: 2
  # podCIDR: "10.20.0.0/16"
  # serviceCIDR: "10.30.0.0/16"
  nodeAuthorizedKeys: []
  nodeAuthorizedKeysFromAgent: true
  imageArchives:
    - "/opt/installer/images.tar.gz"

registry:
  address: "198.19.184.31"
  localImageArchives:
    - "/opt/installer/images.tar.gz"

vsphere:
  endpoint: "vcenter.example.com"
  thumbprint: "AA:BB:CC:DD:EE:FF:..."
  datacenter: "Datacenter1"
  datastore: "datastore1"
  folder: "Hypershield-Follower"
  network: "VM Network"
  resourcePool: "*/Resources"
  template: "clusterX-node-template"
  ova: "/opt/installer/node.ova"
  username: "administrator@vsphere.local"
  password: "YourSecurePassword"

networks:
  - name: "VM Network"
    ipam:
      name: "follower-primary"
      # These addresses must not overlap with the apiServer or registry addresses
      addresses:
        - 198.19.184.32-198.19.184.40
      prefix: 24
      gateway: 198.19.184.1
      nameservers:
        - 8.8.8.8
        - 1.1.1.1
  # SmartSwitch network (optional, for dual-leg design)
  - name: "SmartSwitch Network"
    # In a dual-legs design, the Kubernetes API is also exposed on the
```

© 2026 Cisco and/or its affiliates. All rights reserved.

```
    # switch network and thus requires a dedicated IP.
    # This IP is used by SmartSwitches to contact the controller.
    apiServerVIP: 198.19.185.30
    ipam:
      name: "follower-smartswitch"
      # These addresses must not overlap with the apiServerVIP of the second leg
      addresses:
        - 198.19.185.32-198.19.185.40
      prefix: 24
     #  # Additional nameservers for the secondary NIC (optional)
     #  nameservers:
     #    - 8.8.8.8
     #    - 1.1.1.1
```

Validate and install the Leader cluster

Validate the Leader bootstrap configuration before installing:

```
isoctl bootstrap validate -f /home/iep/hypershield/bootstrap_leader.yaml
```

Expected output:

```
Bootstrap config validation passed (all)
```

Install the Leader management cluster:

```
isoctl bootstrap install -f /home/iep/hypershield/bootstrap_leader.yaml --log-level=debug
```

This process creates the Leader management cluster with all necessary components. When it completes, configure `kubectl` to use the Leader kubeconfig:

```
export KUBECONFIG=/home/iep/hypershield-leader.yaml
kubectl get nodes
```

Expected output (node names will differ in your environment):

```
NAME                                  STATUS   ROLES           AGE   VERSION
hypershield-leader-infra-xxxx         Ready    control-plane   10m   v1.32.4
hypershield-leader-infra-md-xxxx-xxxx Ready    <none>          8m    v1.32.4
...
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Validate and install the Follower cluster

Validate the Follower bootstrap configuration:

```
isoctl bootstrap validate -f /home/iep/hypershield/bootstrap_follower.yaml
```

Install the Follower management cluster:

```
isoctl bootstrap install -f /home/iep/hypershield/bootstrap_follower.yaml --log-level=debug
```

Configure `kubectl` to use the Follower kubeconfig:

```
export KUBECONFIG=/home/iep/hypershield-follower.yaml
kubectl get nodes
```

Next steps

The Leader and Follower management clusters are installed. Next, Install Hypershield on Leader and Follower.

© 2026 Cisco and/or its affiliates. All rights reserved.

Install Hypershield on Leader and Follower

This page shows how to install Hypershield on the Leader management cluster, extract the credentials required by the Follower, and then install Hypershield on the Follower cluster.

> **Note**
>
> All steps on this page are performed on the management VM unless otherwise noted.

Install Hypershield on the Leader

> **Caution**
>
> Ensure the `KUBECONFIG` environment variable points to the Leader cluster before running any commands in this section.

Switch to the Leader cluster context:

```
export KUBECONFIG="/home/iep/hypershield-leader.yaml"
```

Create the Hypershield Leader configuration file at:

`/home/iep/hypershield/hypershield-install-leader.yaml`

The following is an example of all available options that can be place in `hypershield-install-leader.yaml`. Update all values to match your environment:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
# Hypershield Leader configuration
timescape:
  profile: leader
  ip: "198.19.184.12"   # Timescape Gateway load-balancer IP

  # OIDC proxy configuration (optional)
  # Uncomment if OIDC provider requires HTTP proxy
  # overrides:
  #   - 'gateway.extraEnv[0].name=HTTP_PROXY'
  #   - 'gateway.extraEnv[0].value=http://proxy.example.com:8080'
  #   - 'gateway.extraEnv[1].name=HTTPS_PROXY'
  #   - 'gateway.extraEnv[1].value=http://proxy.example.com:8080'
  #   - 'gateway.extraEnv[2].name=NO_PROXY'
  #   - 'gateway.extraEnv[2].value=localhost\,127.0.0.1\,.svc\,.cluster.local'

  server:
    hostname: "timescape-grpc.198.19.184.12.nip.io"
    rbac:
      oidcIssuerURL: "https://dex.198.19.184.50.nip.io:5556/dex"
      oidcClientID: "timescape"
      adminEmails:
        - "admin@example.com"
      oidcCAFile: "/home/iep/dex-server/dex.crt"

secureConnector:
  ip: "198.19.184.13"
  hostname: "secure-connector-leader.198.19.184.13.nip.io"

# Public API (optional)
publicAPI:
  enabled: true
  hostname: "timescape-api.198.19.184.12.nip.io"

ui:
  enabled: true
  hostname: "timescape.198.19.184.12.nip.io"
  auth:
    oidc:
      clientSecretFile: "/home/iep/dex-server/oidc-client-secret.txt"
      issuerCAConfigMapName: "oidc-ca"

# SCC (optional)
# scc:
#   enabled: true
#   tenantID: "00000000-0000-0000-0000-000000000000"
```

Install Hypershield on the Leader cluster:

```
isoctl hypershield install -f /home/iep/hypershield/hypershield-install-leader.yaml
```

Extract Leader credentials for the Follower

Before installing Hypershield on the Follower, extract the necessary credentials from the Leader cluster.

Ensure you are using the Leader kubeconfig:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
export KUBECONFIG="/home/iep/hypershield-leader.yaml"
```

Extract the Timescape CA certificate:

```
kubectl get secret hubble-timescape-server-cert -n hubble-timescape \
  -o jsonpath='{.data.ca\.crt}' | base64 --decode \
  > /home/iep/follower-credentials/leader-ca.crt
```

Extract the Secure Connector CA certificate:

```
kubectl get secret sc-server-root-ca-tls -n hubble-timescape \
  -o jsonpath='{.data.ca\.crt}' | base64 --decode \
  > /home/iep/follower-credentials/sc-server-ca.crt
```

Register the Follower in the Leader Timescape UI

1. Access the Timescape UI at the hostname you configured (for example, `https://timescape.198.19.184.12.nip.io`).
2. Navigate to the follower registration section.
3. Create a new follower with the name matching your Follower cluster name (for example, `hypershield-follower`).
4. Copy the generated join credential.
5. Save the follower join credential to a file:

   `/home/iep/follower-credentials/credential`

   Paste the credential from the Timescape UI and save the file.

6. Set secure permissions on the credential files:

   ```
   chmod 0600 /home/iep/follower-credentials/*
   ```

Install Hypershield on the Follower

> **Caution**
>
> Ensure the `KUBECONFIG` environment variable points to the Follower cluster before running any commands in this section.

Switch to the Follower cluster context:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
export KUBECONFIG="/home/iep/hypershield-follower.yaml"
```

Create the Hypershield Follower configuration file at:

`/home/iep/hypershield/hypershield-install-follower.yaml`

The following is an example of all available options that can be place in `hypershield-install-follower.yaml`. Update all values to match your environment:

```
# Hypershield Follower configuration
timescape:
  profile: follower
  follower:
    clusterName: "hypershield-follower"   # Must match the name registered in leader UI
    leaderAddress: "timescape-grpc.198.19.184.12.nip.io"
    k8sHost: "https://198.19.184.30:6443"   # Follower's Kubernetes API endpoint
    credentialFile: "/home/iep/follower-credentials/credential"
    leaderCAFile: "/home/iep/follower-credentials/leader-ca.crt"
  ingester:
    ip: "198.19.185.34"   # IP on SmartSwitch network

secureConnector:
  serverAddress: "secure-connector-leader.198.19.184.13.nip.io"
  caFile: "/home/iep/follower-credentials/sc-server-ca.crt"

# CLC (optional)
clc:
  enabled: true
  ip: "198.19.185.35"   # IP on SmartSwitch network
```

Install Hypershield on the Follower cluster:

```
isoctl hypershield install -f /home/iep/hypershield/hypershield-install-follower.yaml
```

Next steps

Hypershield is installed on both clusters. Next, Verify the installation.

© 2026 Cisco and/or its affiliates. All rights reserved.

Verify the installation

This page shows how to verify that both the Leader and Follower clusters are healthy and that the Follower is connected to the Leader.

> **Note**
>
> All steps on this page are performed on the management VM unless otherwise noted.

Verify the Leader cluster

Switch to the Leader cluster context and check the cluster state:

```
export KUBECONFIG="/home/iep/hypershield-leader.yaml"

# Check nodes
kubectl get nodes

# Check Timescape components
kubectl get pods -n hubble-timescape

# Check Gateway API
kubectl get gateways -A
```

All nodes should show a `Ready` status and all pods in the `hubble-timescape` namespace should show a `Running` status.

Verify the Follower cluster

Switch to the Follower cluster context and check the cluster state:

```
export KUBECONFIG="/home/iep/hypershield-follower.yaml"

# Check nodes
kubectl get nodes

# Check Timescape components
kubectl get pods -n hubble-timescape
```

All nodes should show a `Ready` status and all pods in the `hubble-timescape` namespace should show a `Running` status.

© 2026 Cisco and/or its affiliates. All rights reserved.

Verify the Follower connection in the Timescape UI

1. Access the Leader Timescape UI at the hostname you configured (for example, `https://timescape.198.19.184.12.nip.io`).
2. Navigate to the follower management section.
3. Confirm that the Follower shows a status of **Connected**.

If the Follower does not appear as connected, review the Install Hypershield on Leader and Follower page to ensure the credential file and CA files were correctly extracted and referenced in the Follower configuration.

Next steps

Hypershield is now installed in your VMware vCenter environment and you are now ready to add an enforcement point. For a Smart Switch enforcement point, see Register a Smart Switch.

© 2026 Cisco and/or its affiliates. All rights reserved.

Operate the environment

This section shows you how to operate the Hypershield platform after the initial environment is installed. It covers day-to-day tasks such as enrolling clusters, assigning permissions, managing network objects and policies, and observing traffic.

The network object and policy pages provide Timescape user interface workflows with supported public REST API alternatives in collapsible panels. For background on what Timescape is and how it fits into the platform, see Timescape and SCC.

© 2026 Cisco and/or its affiliates. All rights reserved.

Register a leader cluster with SCC

When Hypershield runs on Cisco Security Cloud Control (SCC), the Timescape UI requires at least one leader cluster to be enrolled before any other feature becomes available. Until a leader is registered, the UI redirects all navigation to the Leaders page.

This page describes how to enroll your leader through the Timescape UI.

Prerequisites

- Hypershield feature enabled on your SCC tenant
- Timescape is deployed and accessible (see Timescape and SCC).
- You have credentials for an identity provider configured with Timescape OIDC, or direct access to the UI.

Access the Leaders page

1. Ensure to have selected the correct organization where the Hypershield was enabled
2. On the left menu, navigate to **Hypershield**
3. When no leader is enrolled, the UI automatically redirects you to the Leaders page on first access. If a leader already exists, navigate to Operations ‣ Leaders in the sidebar.

Register a new leader

1. On the top of the leaders page, a provisioning ID is displyed, click **Copy** and save the value for future use
2. Click **+ Add leader cluster** in the middle of the screen.
3. Fill in the leader details:
    - **Leader name** – A human-readable name to identify this leader cluster.
    - **Description** (optional) – Additional context about the leader's purpose or environment.
4. Click **Save**.
5. The UI generates a one-time password (OTP) for the leader cluster to use during its initial connection. Copy this value before leaving the page.
6. The leader cluster uses this OTP to authenticate its first connection to Timescape. Once connected, the leader appears with an active status in the Leaders list.

Connect the local leader to the SCC

Perform these steps on the management VM.

© 2026 Cisco and/or its affiliates. All rights reserved.

1. Create a file with the value of the OTP gathered during the leader creation on the SCC, such as `/tmp/scc-bootstrap-token`.
2. Open file `/home/iep/hypershield/hypershield-install-leader.yaml`.
3. Under the `timescape` key, add the `scc` information

```
timescape:
  profile: leader
  ...
  scc:
    enabled: true
    tenantID: "<Provision ID gathered during the leader creation on the SCC>"
  ...
```

4. Still under the `timescape` key, under the `secureConnector` add a new key: `saasClient`.

```
timescape:
  profile: leader
  ...
  secureConnector:
    saasClient:
      enabled: true
      serverAddress: "secureconnector.us.prod.hypershield.cisco.com:443"
      bootstrapTokenFile: "/tmp/scc-bootstrap-token"
      # SassS Client proxy configuration (optional)
      # Uncomment if SaaS Client requires HTTP proxy and adpate configuration to your environement
      # overrides:
      #    - "extraEnv[0].name=HTTP_PROXY"
      #    - "extraEnv[0].value=http://proxy.company.com:8080"
      #    - "extraEnv[1].name=HTTPS_PROXY"
      #    - "extraEnv[1].value=http://proxy.company.com:8080"
      #    - "extraEnv[2].name=NO_PROXY"
      #    -
"extraEnv[2].value=localhost\\,127.0.0.1\\,::1\\,10.0.0.0/8\\,172.16.0.0/12\\,192.168.0.0/16\\,.cl
uster.local\\,.svc\\,.nip.io"
  ...
```

5. Update the hypershield install

```
isoctl hypershield install -f /home/iep/hypershield/hypershield-install-leader.yaml
```

6. Verify the connectivity to the SCC

```
kubectl logs -n hubble-timescape deployment/sc-saas-client
```

Expected output:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
time=2026-09-03T16:31:30.782Z level=INFO msg="starting sc-client (follower ID will be derived from
bootstrap token)" leaderAddress=secureconnector.us-east-2.prod.hypershield.cisco.com:443
socks5Address=127.0.0.1:1080
...
time=2026-09-03T16:31:31.646Z level=INFO msg="tunnel active, serving SOCKS5 connections"
subsys=connector followerId=06f019e6-c44c-4787-929f-eb43f8658bdd leader=secureconnector.us-east-
2.prod.hypershield.cisco.com:443
time=2026-09-03T16:32:30.783Z level=INFO msg="tunnel traffic summary" subsys=traffic role=follower
followerId=06f019e6-c44c-4787-929f-eb43f8658bdd connectionsCompleted=0 activeConnections=16
bytesSent=0 bytesReceived=0
...
```

7. On the SCC, navigate to **Policies**, the policies page should display the policies that you may already have

View leader details

Click a leader name in the list to view its detail page. The detail page shows:

- Leader metadata (name, description, creation time)

© 2026 Cisco and/or its affiliates. All rights reserved.

Assign user permissions

This page shows how to assign permissions to users. Permissions determine what authorization level a user has on the Hypershield platform. To grant certain permissions to a user, place the user in an appropriate user group.

Note

Internal to Hypershield, permissions are associated with a `role`.

Place a user in a group

To place a user in a group follow these steps:

1. Log into the Keycloak admin console: https://keycloak.<your-domain>/admin/master/console/
2. Select the `hypershield` realm
3. Navigate to **Users** → **<username>** → **Groups**
4. Add the user to the group

Permission changes take effect within approximately 30 seconds - existing sessions honor the change on their next API call. Neither a Pod restart nor a user re-login is required.

Note

The set of available groups is predefined:

- `admin`
- `policy_manager`
- `workload_agent_installer`
- `nbe_installer`
- `security`

For details on how identity and access management works, see conceptual page PLACEHOLDER.

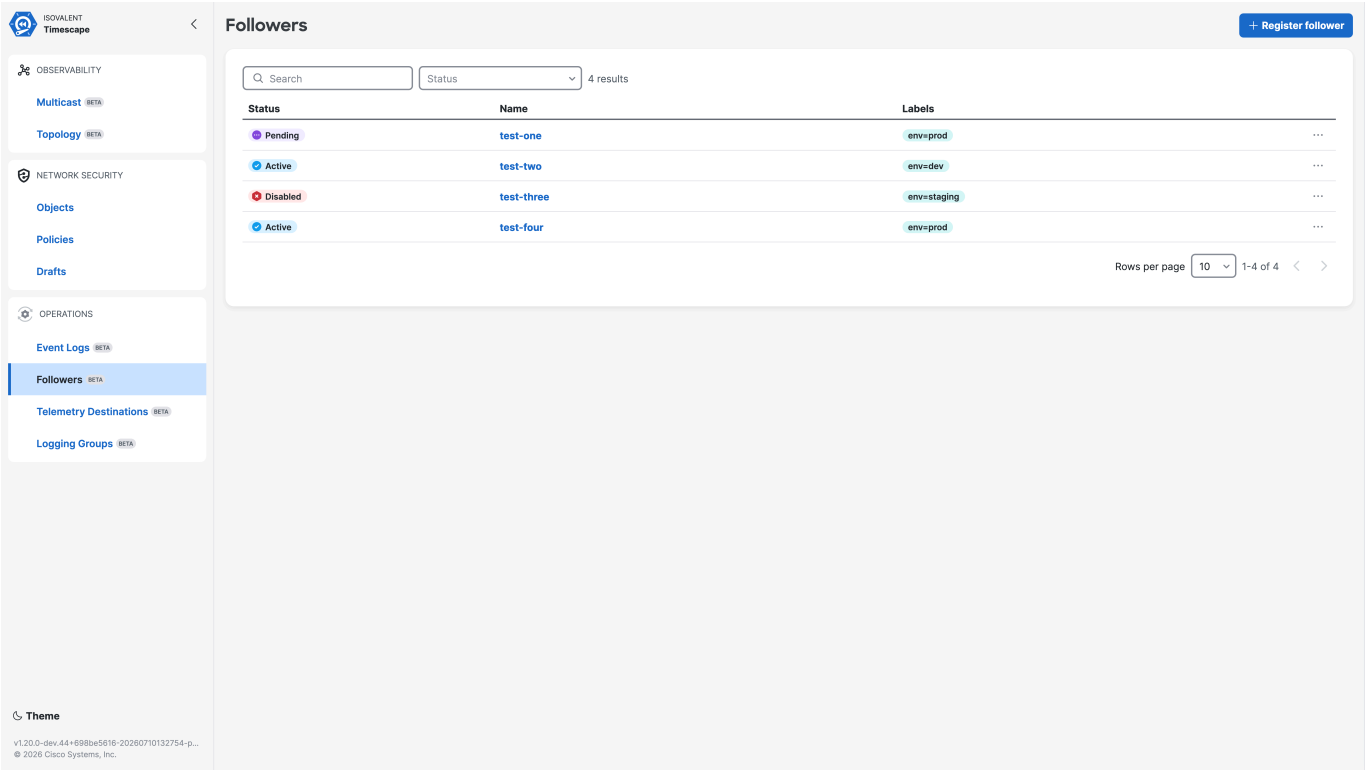
© 2026 Cisco and/or its affiliates. All rights reserved.

Manage followers

Followers (also called clusters in the API) are management clusters registered with Timescape. They receive segmentation policies from the leader, distribute those policies to the enforcement points they manage, and report policy synchronization and operational status back to the leader. In the Timescape UI, you manage them from Operations ▸ Followers.

View the followers list

Navigate to Operations ▸ Followers to see all registered clusters.



The followers list shows each cluster with its name, status, and labels.

The list supports:

- **Status filter** – Filter by Active, Disabled, or Pending status.
- **Search** – Filter clusters by name.
- **Actions menu** – Enable, disable, or delete clusters from the row action menu.

Cluster statuses:

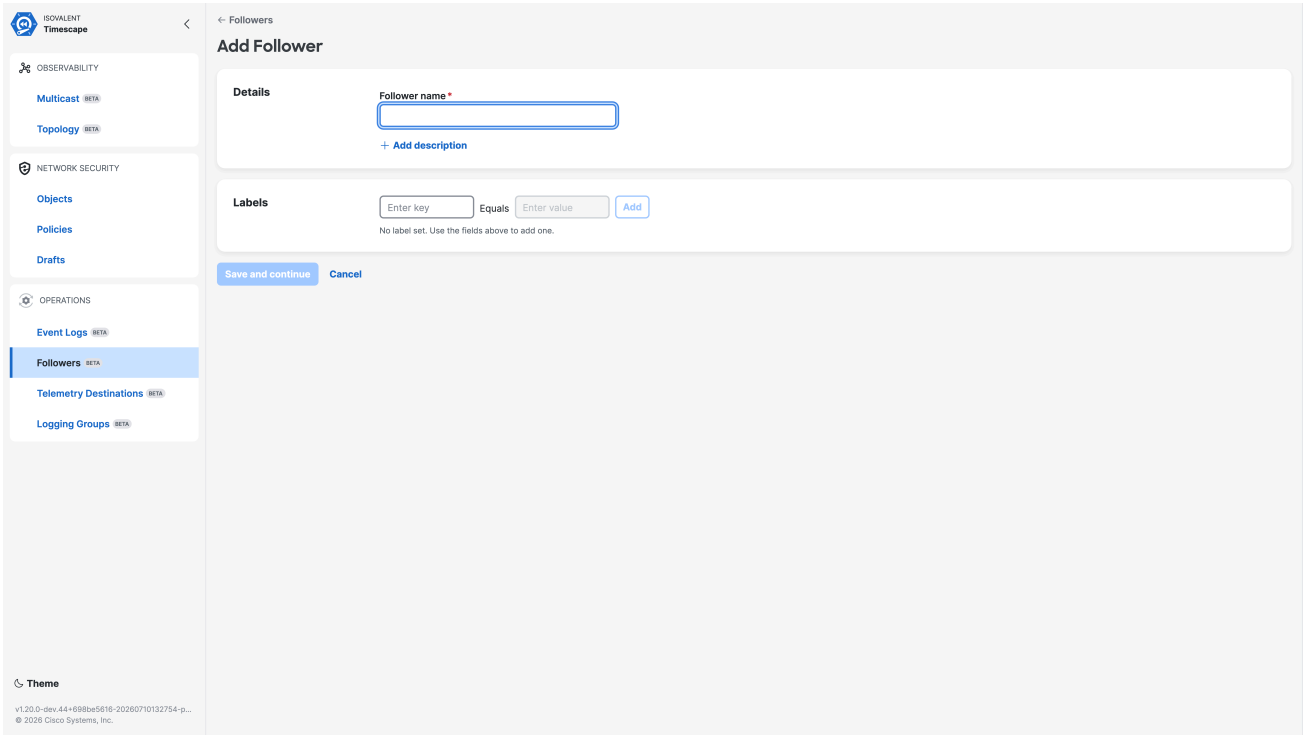
- **Active** – The cluster is connected and receiving policy updates.

© 2026 Cisco and/or its affiliates. All rights reserved.

- **Pending** – The cluster is registered but has not yet connected.
- **Disabled** – The cluster is paused and not receiving policy updates.

Register a new follower

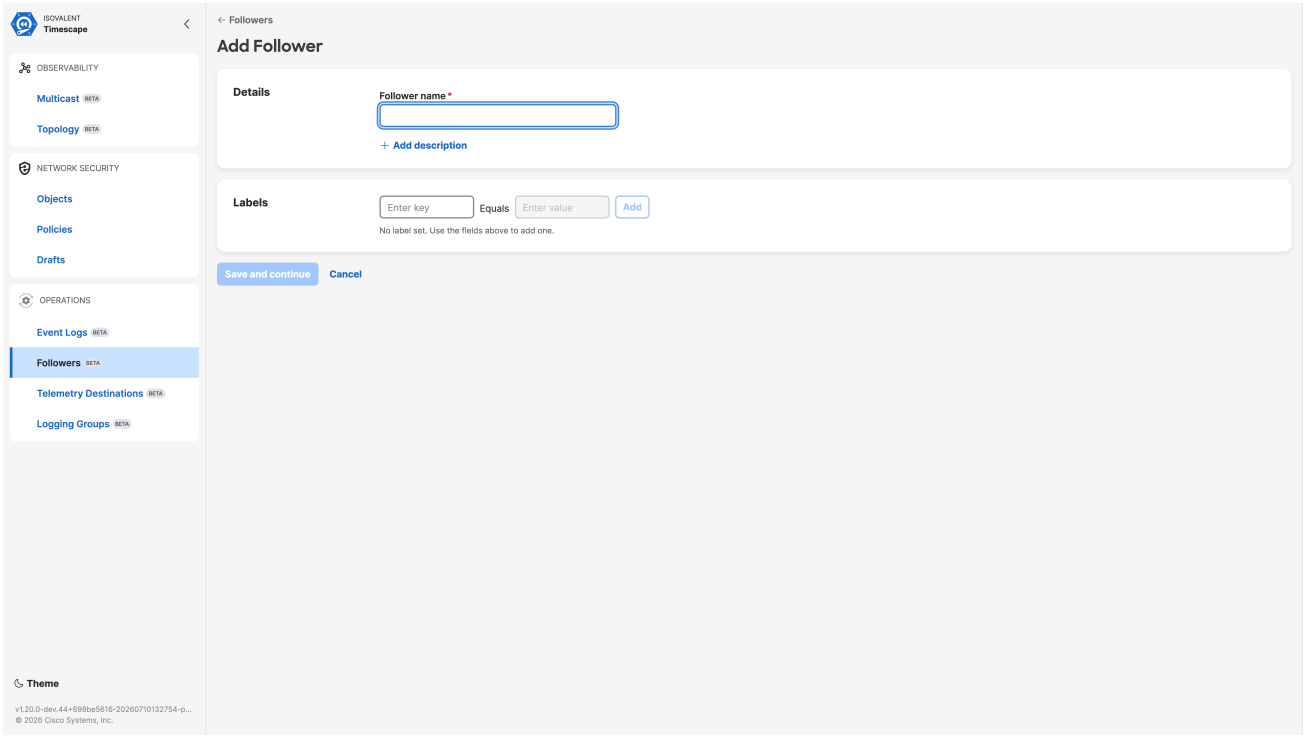
1. Click **Register follower**.
2. Fill in the cluster details:
   - **Name** – A unique name for the cluster.
   - **Labels** (optional) – Key-value pairs for organizing clusters (for example, by site, region, or role).



Enter the follower name and optional labels.

3. Click **Save**.
4. The UI generates connection credentials for the cluster. Provide these to the cluster administrator to complete the registration.

© 2026 Cisco and/or its affiliates. All rights reserved.



Copy the connection credentials and provide them to the cluster administrator.

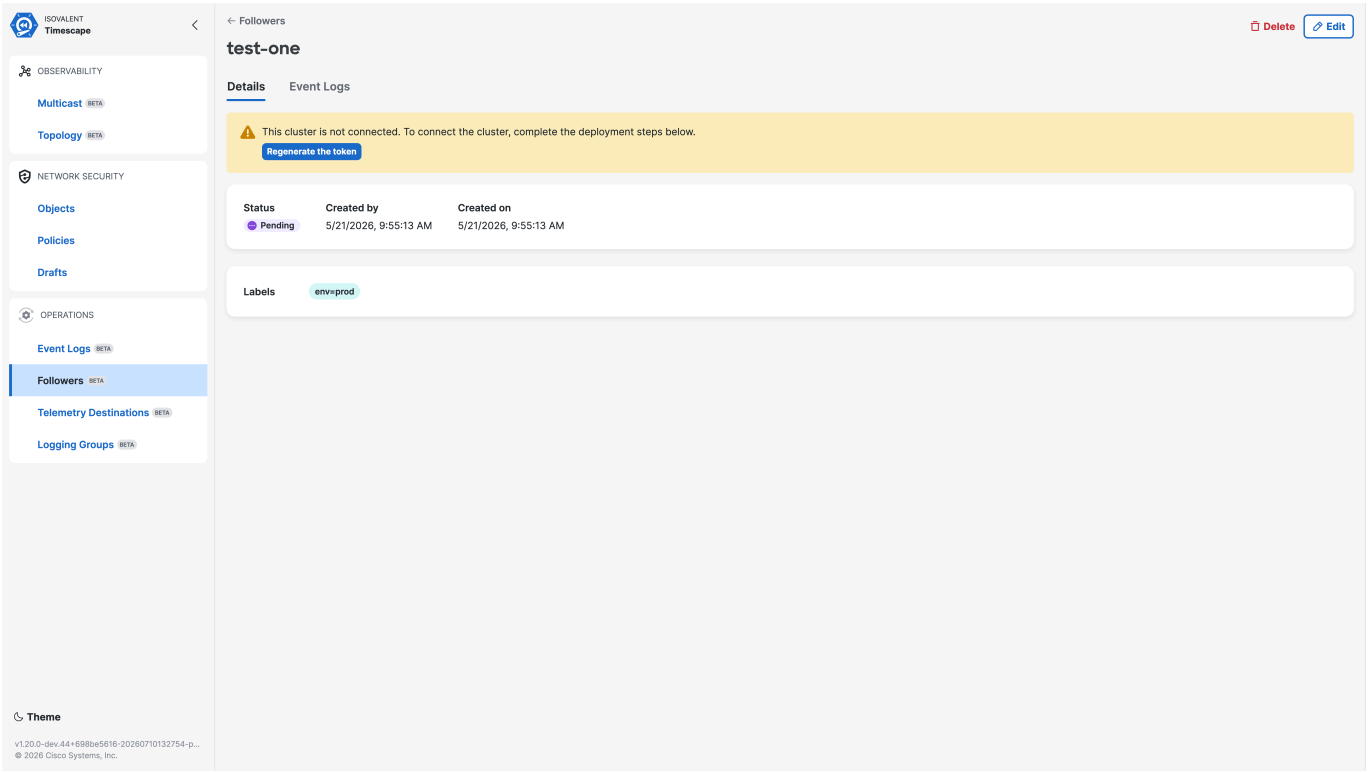
The cluster administrator uses these credentials to register the follower with the leader. After the follower is successfully registered, it becomes active and can manage enforcement points.

View follower details

Click a cluster name in the list to open its detail page. The detail page shows:

- Cluster metadata (name, labels, creation time)
- Current connection status
- Policy synchronization state

© 2026 Cisco and/or its affiliates. All rights reserved.

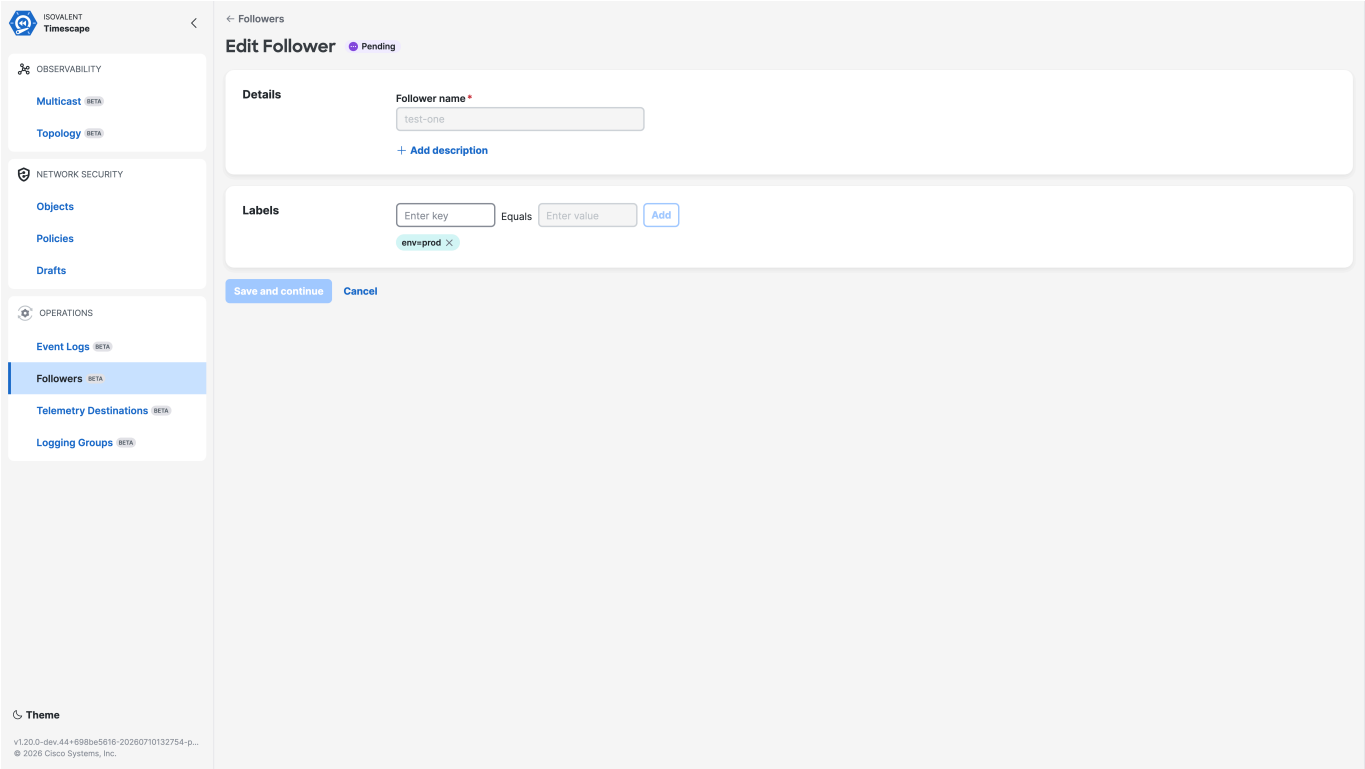


The detail page shows the cluster's connection state and metadata.

Edit a follower

From the detail page, click **Edit** to modify the cluster's name or labels.

© 2026 Cisco and/or its affiliates. All rights reserved.



Update the cluster name or labels.

Enable or disable a follower

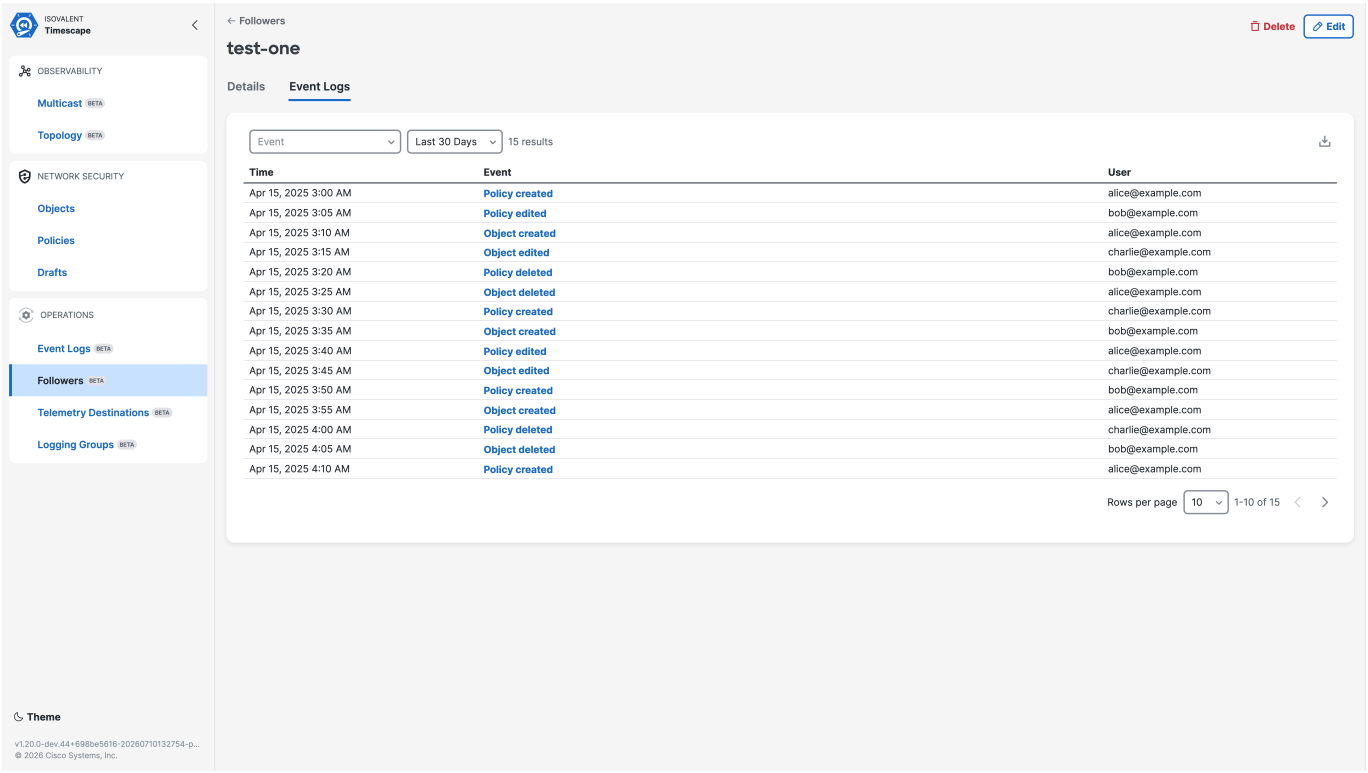
Use the actions menu on the list page to change a cluster's state:

- **Disable** – Pauses policy distribution to the cluster. The cluster retains its last-known policy state but stops receiving updates.
- **Enable** – Resumes policy distribution. The cluster receives all pending policy updates.

View follower event logs

From the cluster detail page, switch to the **Logs** tab to review the event history (registration, state changes, connection events).

© 2026 Cisco and/or its affiliates. All rights reserved.



The logs tab shows the cluster's lifecycle events.

Delete a follower

Use the actions menu to delete a cluster. Only clusters in Disabled or Pending status can be deleted.

> **Caution**
>
> Deleting a cluster removes it from Timescape permanently. The cluster stops receiving policy updates and must be re-registered to rejoin.

© 2026 Cisco and/or its affiliates. All rights reserved.

Manage leaders

Leaders are cluster controllers that coordinate platform operations across your Hypershield deployment. At least one leader must be enrolled for the Timescape UI to function. Leaders handle authentication, security policy distribution, and cluster orchestration.

For background information on leaders and their role in the architecture, see Leader and follower clusters.

Note

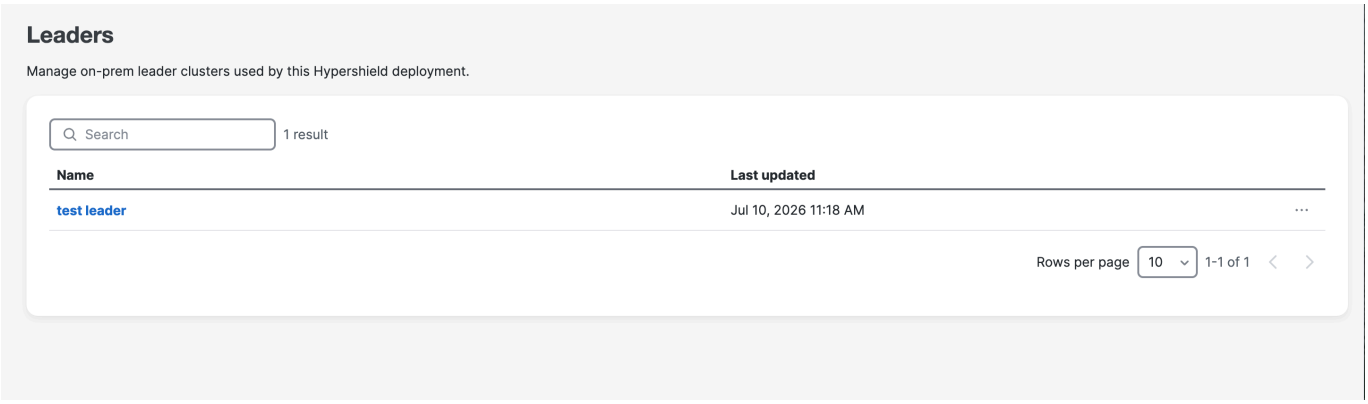
For initial leader enrollment during SCC configuration, see Register a leader cluster with SCC.

Using the user interface

In the Timescape UI, you manage leaders from the Operations ‣ Leaders page.

View the leaders list

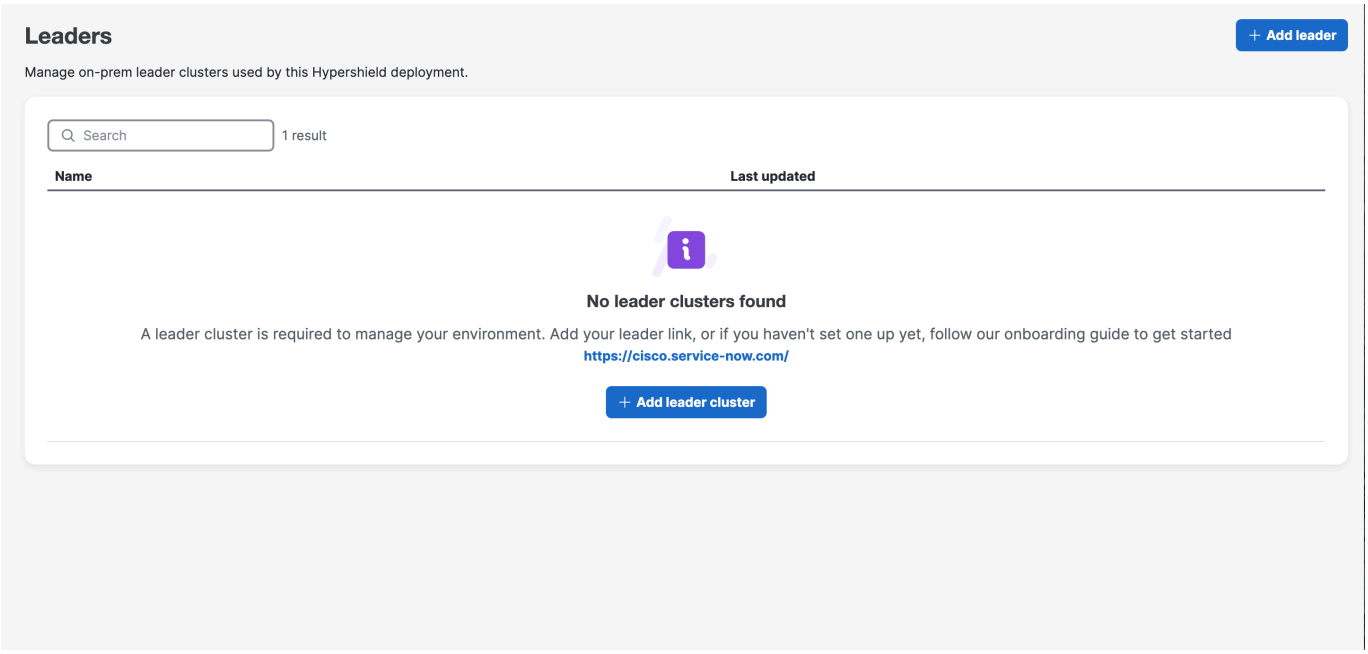
Navigate to Operations ‣ Leaders to see all registered leader clusters in your deployment.



The leaders list shows each leader with its name, status, and connection details.

When no leaders are enrolled yet, you'll see an empty state:

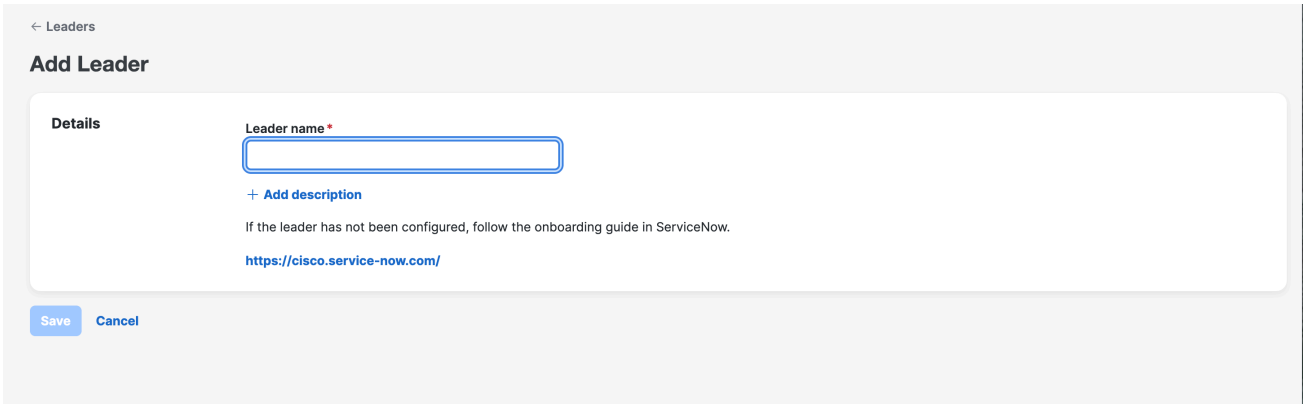
© 2026 Cisco and/or its affiliates. All rights reserved.



The Leaders page displays an empty state when no leader cluster is registered. Click **Add leader** to begin enrollment.

Register a new leader

1. Click **Add leader**.
2. Fill in the leader details in the form that appears:
   - **Leader name** – A unique, human-readable name to identify this leader cluster (must be valid as a Kubernetes resource name).
   - **Description** (optional) – Additional context about the leader's purpose, environment, or location.



Enter the leader name and optional description.

3. Click **Create** or **Save**.
4. After creation, you may receive a one-time password (OTP) or connection token for the leader cluster to use during its initial authentication. Copy this value before leaving the page, as it is shown only once.

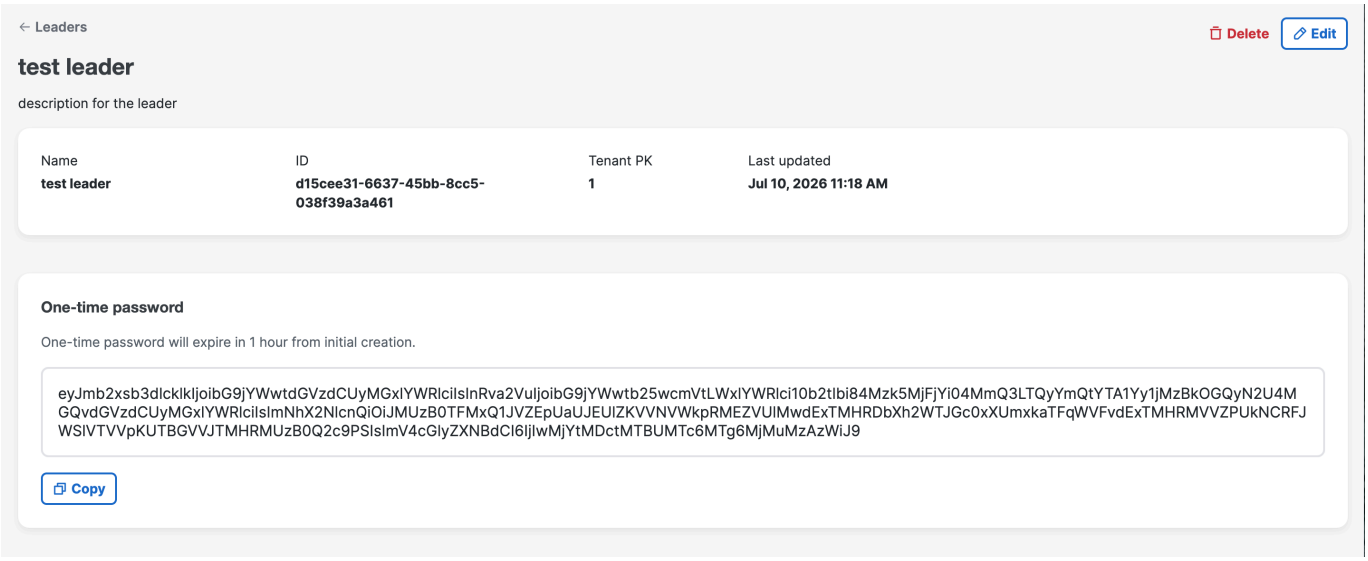
© 2026 Cisco and/or its affiliates. All rights reserved.

5. The leader cluster uses this credential to authenticate its first connection to Timescape. Once connected, the leader appears with an active status in the Leaders list.

View leader details

Click a leader name in the list to open its detail page. The detail page shows:

- Leader metadata (name, description, creation and modification times)
- Connection status and health
- Configuration details
- Associated clusters or resources managed by this leader



The leader detail page displays metadata, connection state, and configuration information.

Edit a leader

From the leader detail page or the leaders list:

1. Click **Edit** or use the actions menu (three-dot icon) on a leader row.
2. Modify the leader properties as needed.
3. Click **Save** to apply your changes.

Delete a leader

From the leaders list, use the actions menu (three-dot icon) on a leader row to delete it.

© 2026 Cisco and/or its affiliates. All rights reserved.

> **Caution**
>
> Deleting a leader disconnects its cluster from Timescape and may impact platform operations. If you delete the last remaining leader, all UI features become unavailable until a new leader is enrolled.

Troubleshooting

Leader not connecting

If a leader appears as disconnected or unhealthy:

- Verify network connectivity between the leader cluster and Timescape.
- Check that the leader cluster has the correct Timescape endpoint configuration.
- Review leader pod logs for authentication or connection errors.
- Ensure firewalls and network connectivity is in place to allow the required communication ports.

Multiple leaders showing conflicts

If multiple leaders show conflicting states:

- Verify that each leader has a unique name.
- Check for clock synchronization issues between leader clusters.
- Review the leader cluster logs for reconciliation errors.

For more troubleshooting guidance, see Hypershield general troubleshooting.

© 2026 Cisco and/or its affiliates. All rights reserved.

Manage Smart Switches

This section describes the operational tasks used to manage Smart Switches in a Hypershield deployment.

© 2026 Cisco and/or its affiliates. All rights reserved.

Register a Smart Switch

Register a Smart Switch to enable centralized policy management and status reporting. After registration, security policies can be distributed to the Smart Switch.

> **Note**
>
> This procedure assumes a Leader and Follower cluster deployment that is fully operational and error-free.

Prerequisites

- The Smart Switch is running a supported Hypershield NX-OS image (`nxos64-s1-dpu.10.6.3s.F.bin`, which reports as `10.6(3s)` in `show version`).
- You have administrative access to the Smart Switch.
- Network connectivity exists between the Smart Switch and the Follower cluster.

Obtain the Smart Switch registration token

First connect to the management VM and enter into the context of the Follower cluster:

```
export KUBECONFIG="/home/iep/hypershield-follower.yaml"
```

Then generate the registration token:

```
/home/iep/hypershield/get-token.sh hypershield smartswitch-token --ca-source secret 198.19.184.30
```

Where `198.19.184.30` is the Follower cluster's Kubernetes API endpoint.

Register the Smart Switch

Log in to the Smart Switch and access the switch console. Then register it with the Follower cluster by using the token generated in the previous step:

```
switch# service system hypershield register <registration-token>
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Verify the Smart Switch registration

Verify the registration on the switch by invoking the following commands on its console:

```
switch# service system hypershield test controller connection
switch# show service-acceleration status details
Service System: hypershield
 Source Interface: loopback101 (172.31.251.129)
 DPU Load-balance Mode: symmetric-hash
 Agent Status: firewall-ready,redirect-installed
 Agent Health Status: ok
 Controller Connection Status: success
 Agent HA Status: ha-ready
 Services:
   Firewall: in-service

   VRF              Operational State   Affinity(DPU)
   ==================================================
   vrf-lab          forwarding ready    1-4

   VLAN             Operational State   Affinity(DPU)
   ==================================================
   10               forwarding ready    1-4
```

Confirm that:

- `Controller Connection Status` reports a successful connection
- the switch appears in the list of managed switches

Verify the registration on the Follower cluster by running these commands on the management VM:

```
export KUBECONFIG="/home/iep/hypershield-follower.yaml"
kubectl get smartswitch -n hypershield
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Update Smart Switch software

Update Smart Switch software to install new features, enhancements, and fixes. Depending on your maintenance requirements, you can:

- Upgrade the complete NX-OS image.
- Upgrade the Agent Gateway (AGW) software independently.
- Upgrade the Data Processing Unit (DPU) firmware independently.

Note

The NX-OS image includes both the AGW software and the DPU firmware. In most cases, upgrade the complete NX-OS image. Upgrade the AGW software or DPU firmware independently only when you need to install specific fixes or enhancements.

Upgrade the NX-OS image

Prerequisites

- You have administrative access to the Smart Switch.
- You have obtained the appropriate Hypershield NX-OS image for your Smart Switch model.

Install the NX-OS image

1. Log in to the Smart Switch.
2. Copy the NX-OS image to the Smart Switch:

```
switch# copy scp://user@host//path/to/nxos64-s1-dpu.<version>.bin bootflash: vrf management use-
kstack
```

3. Save the running configuration:

```
switch# copy running-config startup-config
```

4. Install the image:

```
switch# install all nxos bootflash:nxos64-s1-dpu.<version>.bin
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Verify the NX-OS upgrade

Run the following command:

```
switch# show service-acceleration status detail
```

Verify that the Smart Switch returns to an operational state.

Upgrade AGW software

The AGW software is included in the NX-OS image. You can upgrade it independently to install recent fixes or enhancements without installing a new NX-OS image.

Install the AGW software

1. Log in to the Smart Switch.
2. Copy the AGW software package to the Smart Switch.
3. Install and activate the package:

   ```
   switch# install add bootflash:agent-<version>.rpm activate
   ```

Verify the AGW upgrade

1. Verify that the package is installed:

   ```
   switch# show install active detail
   ```

2. Verify that the Smart Switch is operational:

   ```
   switch# show service-acceleration status detail
   ```

Upgrade DPU firmware

The DPU firmware is included in the NX-OS image. You can upgrade it independently to install recent fixes or enhancements without installing a new NX-OS image.

© 2026 Cisco and/or its affiliates. All rights reserved.

Install the DPU firmware

1. Log in to the Smart Switch.
2. Install and activate the firmware package:

```
switch# install add bootflash:dpu_fw-<version>.rpm activate
```

> **Caution**
>
> The Smart Switch may reboot during the firmware upgrade.

Verify the DPU firmware upgrade

Run the following command:

```
switch# show service-acceleration status detail
```

Verify that the Smart Switch returns to an operational state after the reboot.

© 2026 Cisco and/or its affiliates. All rights reserved.

Enable high availability on a Smart Switch

This page shows you how to enable high availability (HA) on a Smart Switch via the NX-OS CLI. High availability flow state synchronisation is defined between two Smart Switches, referred to as an “HA pair” in this document.

Note

Review section Caveats and limitations before continuing.

Define a Loopback Address on each Smart Switch

From the Smart Switch console, in the default VRF define a Loopback address that is going to be used for HA communication between the smart switches. Make sure that the routing configuration is in place for this Loopback to have reachability to the Loopback of the other Smart Switch over a Port-Channel between the two switches.

```
switch(config)# interface Loopback101
switch(config-if)# ip address 192.0.2.2/32
```

Create the HA configuration

In service system hypershield, enter high-availability and specify the source interface for the HA traffic as well as the peer IP address, which is the IP address of the Loopback interface on the other smart switch that is part of the same HA pair. Do not confuse the source-interface configuration under service system hypershield with the source-interface configuration under high-availability, these are two different interfaces, the first one is used for the Smart Switch to connect to Hypershield, the second one is used for high availability between two smart switches.

```
switch(config)# service system hypershield
switch(config-svc-sys)# high-availability
switch(config-svc-sys-ha)# source-interface loopback101
switch(config-svc-sys-ha)# peer 192.0.3.2
switch(config-svc-sys-ha)# no shutdown
```

Verify HA enablement

You can verify the enablement of HA on the switch with the NX-OS command “show service-acceleration high-availability status”

© 2026 Cisco and/or its affiliates. All rights reserved.

```
switch# show service-acceleration high-availability status
Service System: hypershield
HA Source Interface: loopback101 (192.0.2.2)
HA Admin State: no-shutdown
Agent Status for Service Firewall: ready
Agent HA Status: ready
Peers:
PeerIP Peer Service State HA State with Peer Reason
192.0.3.2 ready ha-ok all criteria met
```

Possible status information includes:

- peer state
- last keepalive
- bulk sync status
- statistics

To view session flows:

```
switch# show tech-support dpu module all
```

Resources

Caveats and limitations

- Single peer support - Only one HA peer is supported.
- No inline sync during bulk sync - Flows created during bulk sync are not synced; requires second bulk sync after completion.
- Split brain risk - Network partition creates active-active scenario with no fencing or quorum mechanism.
- No graceful shutdown - Planned maintenance treated as failure; a full re-sync is required on restart.
- Keepalive thread contention - Keepalive RX shares thread with data path; heavy traffic can delay keepalive processing and cause false timeouts
- Limited configuration - Manual peer configuration is required; no auto-discovery; no dynamic peer addition (must restart)
- No statistics sync - Counters reset on failover; no historical data preserved
- TCP state limitations - Only basic TCP state sync is supported; no advanced TCP options synced; RST possible on complex TCP states
- No flow pruning - All flows synced regardless of age; stale flows may be synced.
- Event delivery not guaranteed - Unix socket can drop events if FWA is busy; no retry for event delivery.

© 2026 Cisco and/or its affiliates. All rights reserved.

Scalability

Keepalive overhead

Metric	Value
message size	20 bytes
frequency	every 5 seconds per peer
bandwidth	~32 bps per peer (negligible)
CPU	minimal (single threaded, event-driven)

Bulk sync performance

Metric	Value
flow entry size	68 bytes
batch size	up to 10 flows = 680 bytes + headers
ACK overhead	8 bytes per batch + headers
estimated sync time	~1-5 seconds for 10K flows

Scalability

Metric	Value
maximum flows	limited by session table capacity
tested	up to 100K flows in bulk sync
peer limit	1 peer (single HA pair)
thread model	single HA thread + per-DPU RX workers

Performance considerations

- batch transmission reduces per-message overhead
- ACK-based flow control prevents receiver overwhelm
- cutoff timestamp prevents stale flow sync
- compressed flow representation (68 bytes per flow)

© 2026 Cisco and/or its affiliates. All rights reserved.

Disable

Disable HA on a Smart Switch by entering “shutdown” under “high-availability”:

```
switch(config)# service system hypershield
switch(config-svc-sys)# high-availability
switch(config-svc-sys-ha)#shutdown
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Move a Smart Switch to another follower

Move a Smart Switch to another follower when replacing a follower, migrating a site, or changing the management topology. Re-registering the Smart Switch establishes a trusted connection with the new follower and enables it to receive policy updates from that follower.

Prerequisites

- The destination follower is deployed and operational.
- Obtain a registration token for the smart switch from the destination follower.
- You have administrative access to the Smart Switch.
- Network connectivity exists between the Smart Switch and the destination follower.

Move the Smart Switch

1. Log in to the Smart Switch.
2. Register the Smart Switch with the destination follower:

   ```
   switch# service system hypershield register <registration-token>
   ```

   Replace `<registration-token>` with the registration token generated for the destination follower. Re-registering the Smart Switch replaces the existing registration with the new follower.

Verify the follower connection

1. On the Smart Switch, verify that the connection to the destination follower was established:

   ```
   switch# show service-acceleration status detail
   ```

   Verify that `Controller Connection Status` reports a successful connection.

2. In the Timescape UI, verify that the Smart Switch is connected to the destination follower.
3. Confirm that the Smart Switch receives the latest segmentation policies from the destination follower.

© 2026 Cisco and/or its affiliates. All rights reserved.

Remove a Smart Switch

To remove a Smart Switch, unregister the switch and then remove its Hypershield configuration.

Unregister the switch

On the management VM, list all switches and then unregister the Smart Switch from the Hypershield control plane by specifying its serial number:

```
kubectl get smartswitches -n hypershield
kubectl delete smartswitch -n hypershield <serial-number>
```

The delete command also clears security policies from the DPUs.

Remove the Hypershield configuration

On the Smart Switch console, enter configuration mode and remove the Hypershield configuration:

```
switch# configure terminal
switch(config)# no feature service-acceleration
```

This command also stops the agent container and powers down the DPUs.

Verify the removal

On the management VM, verify that the Smart Switch is no longer listed:

```
kubectl get smartswitches -n hypershield
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Manage network objects

Network objects represent groups of IP addresses (CIDRs) that you reference in security policies. For background on the data model, see Security policies

In the Timescape UI, you manage network objects from the Network Security ‣ Objects page.

Note

This page includes API alternatives to web UI actions. See Timescape API usage for getting started with the Timescape public REST API or go directly to the API reference.

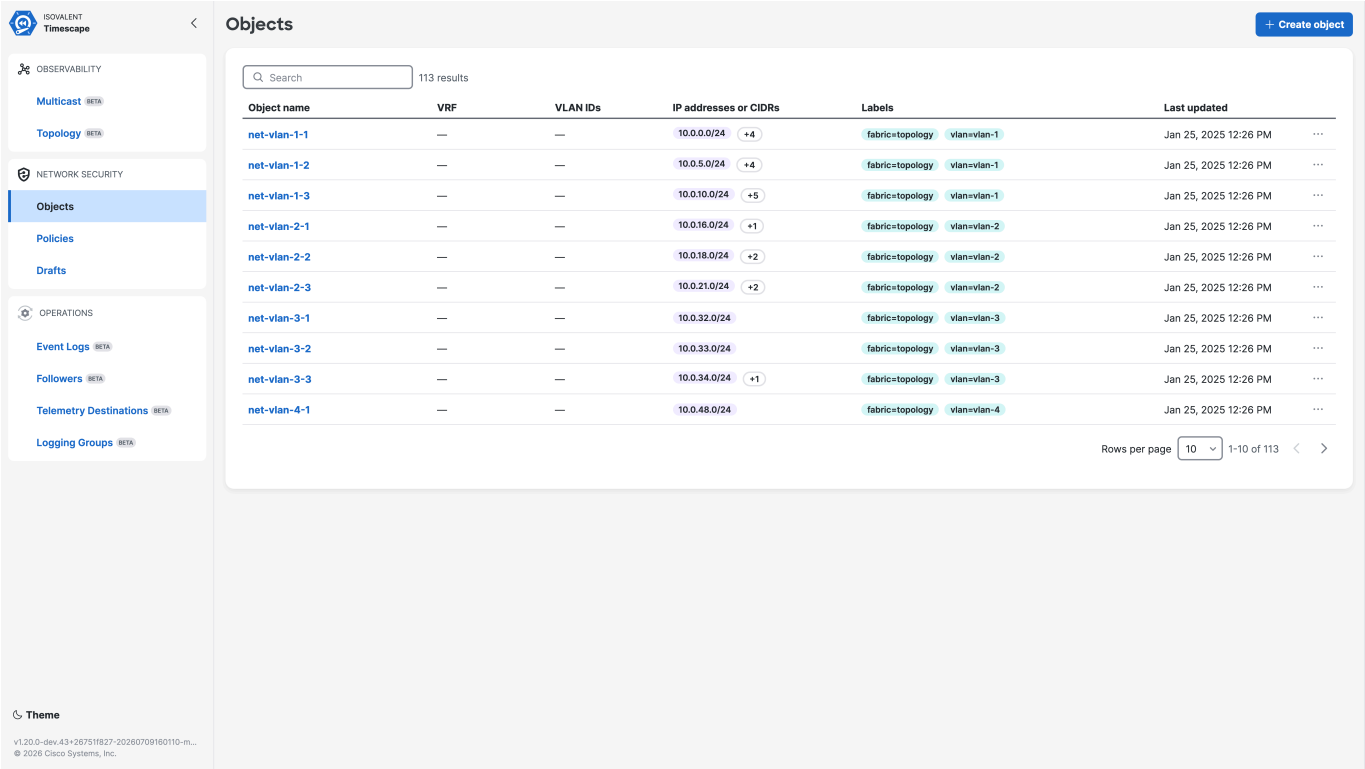
Public API mutations write directly to desired state. They cannot add changes to a Timescape draft.

View the objects list

Use the API

Navigate to Network Security ‣ Objects to see all network object groups in your deployment.

© 2026 Cisco and/or its affiliates. All rights reserved.



The objects list shows each network object group with its name, labels, CIDR count, and last modification time.

The list supports:

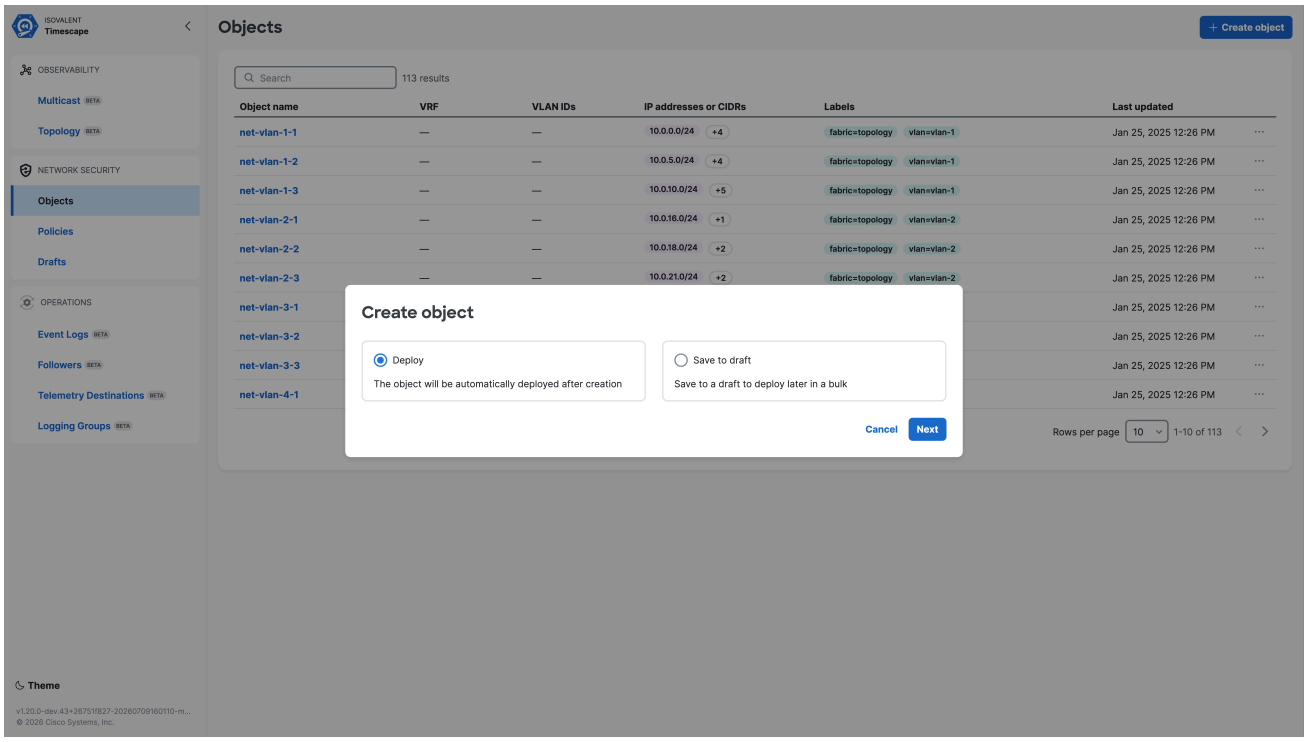
- **Search** – Filter objects by name, labels or IP addresses.
- **Pagination** – Navigate through large object sets.
- **Actions menu** – Edit or delete objects from the row action menu.

Create a network object

Use the API

1. Click **Create object** .
2. Select a deployment target in the modal that appears. The deployment target determines which draft or live environment receives the new object (see the draft section).

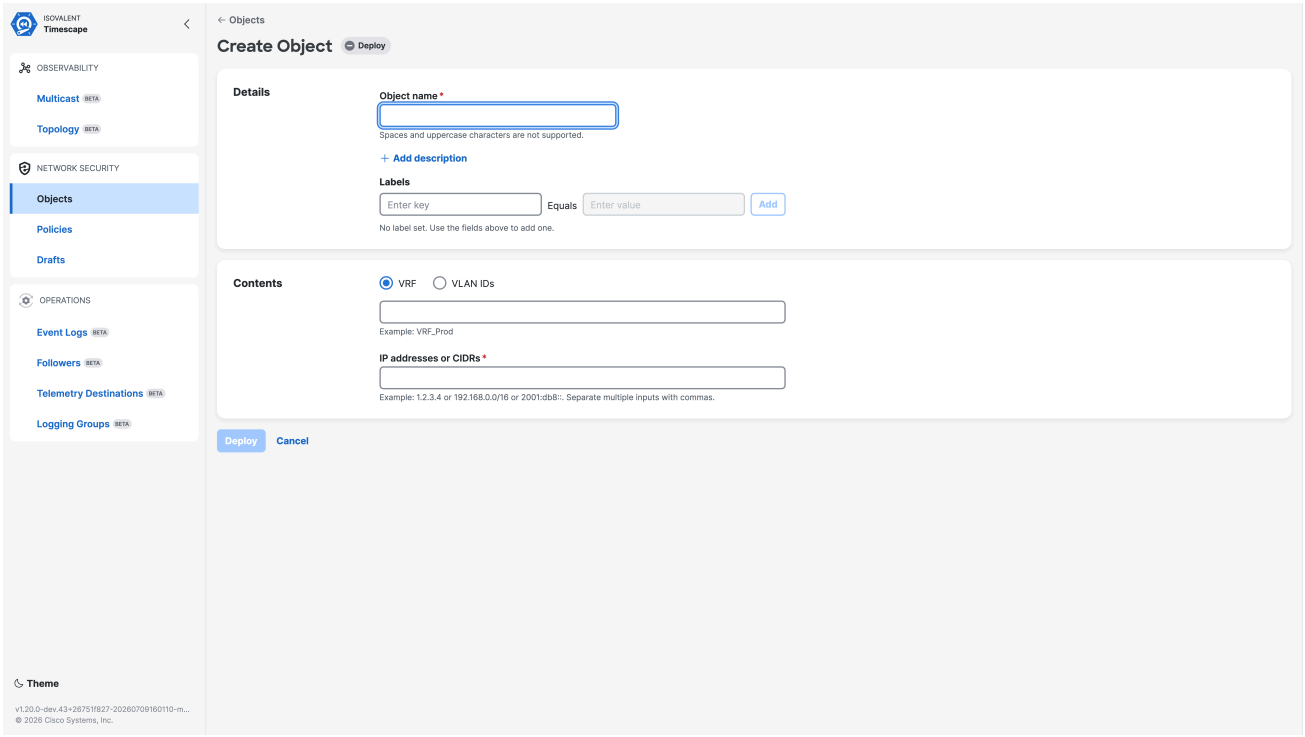
© 2026 Cisco and/or its affiliates. All rights reserved.



Choose whether to save the object directly or add it to an existing draft.

3. Fill in the object details:
    - **Name** – A unique name for the network object group (must be valid as a Kubernetes resource name).
    - **Labels** (optional) – Key-value pairs for organizing and filtering objects.
    - **CIDRs** – One or more IP address ranges that this object represents.

© 2026 Cisco and/or its affiliates. All rights reserved.



Enter the object name, optional labels, and one or more CIDR ranges.

4. Click **Deploy**.

The object is created according to your selected deployment target (live or draft).

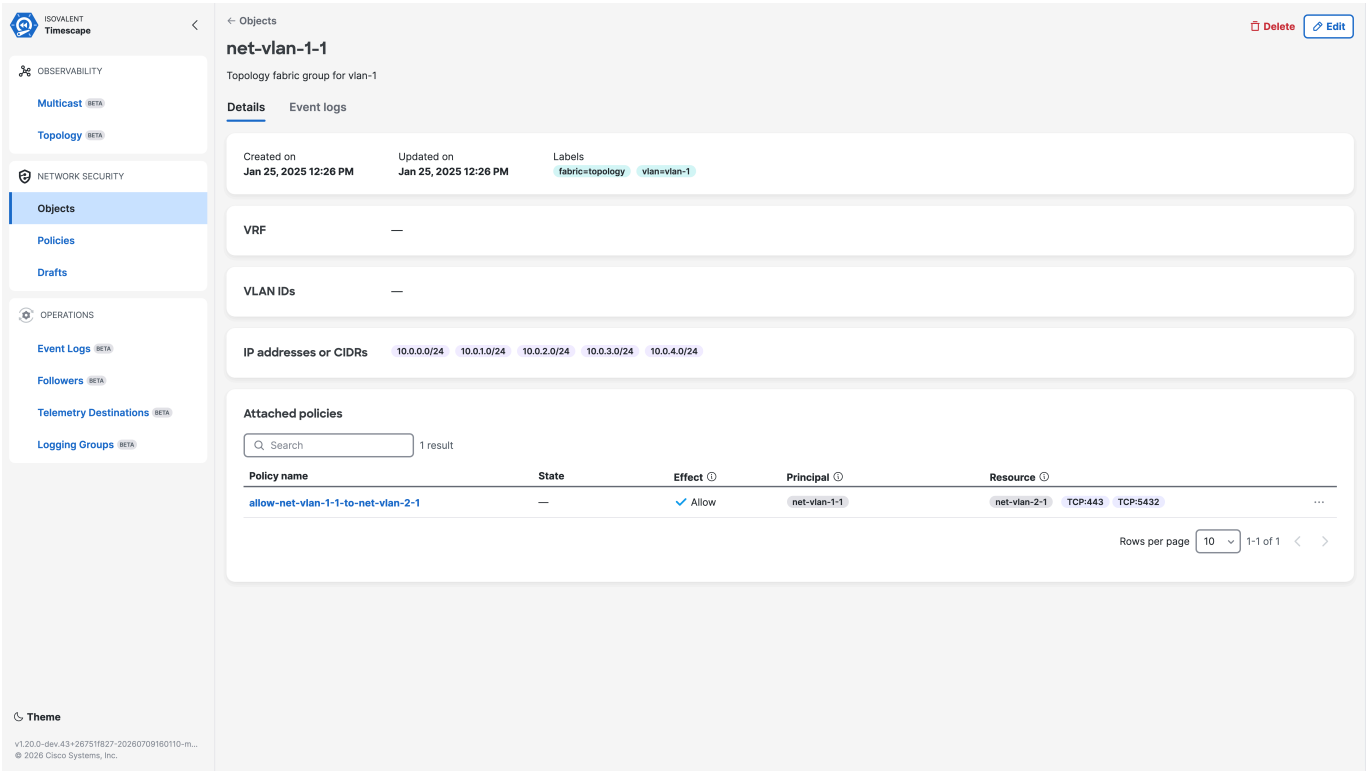
View object details

Use the API

Click an object name in the list to open its detail page. The detail page shows:

- Object metadata (name, namespace, labels, creation and modification times)
- CIDR list
- Attached policies that reference this object

© 2026 Cisco and/or its affiliates. All rights reserved.



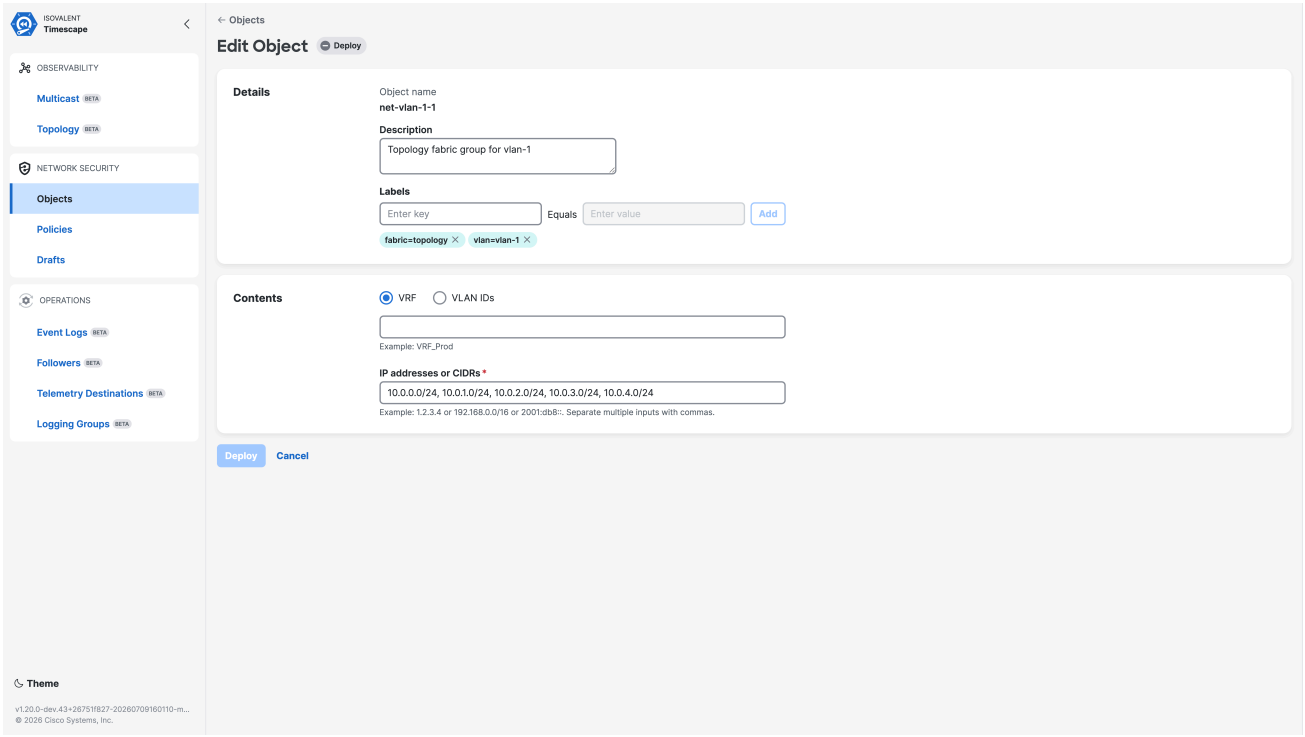
The detail page displays the object's CIDRs and lists all policies that reference it.

Edit a network object

Use the API

1. From the detail page, click **Edit**, or use the actions menu in the list page.
2. Select a deployment target in the modal that appears. The deployment target determines which draft or live environment receives the updated object (see the draft section).
3. Modify the object's labels or CIDRs as needed.

© 2026 Cisco and/or its affiliates. All rights reserved.



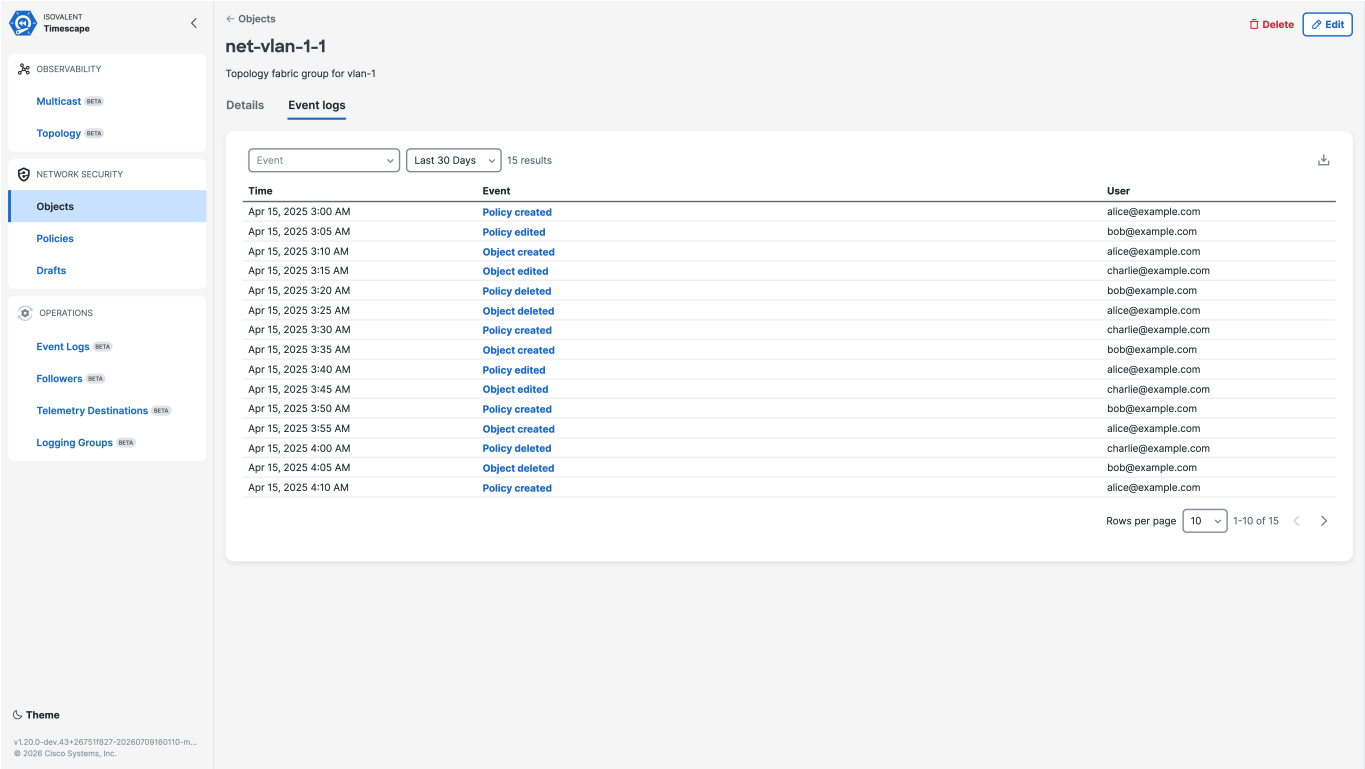
Edit the CIDRs or labels of an existing network object.

4. Click **Deploy**.

View object event logs

From the object detail page, switch to the **Logs** tab to review the event history for this object (creation, modifications, deletions).

© 2026 Cisco and/or its affiliates. All rights reserved.



The logs tab shows the change history for the object.

Delete a network object

Use the API

Use the actions menu on the list page or the detail page to delete an object.

> **Caution**
>
> Delete all security policies that reference this object first. Deleting an object that is still referenced by a policy is not allowed and will result in an error.

© 2026 Cisco and/or its affiliates. All rights reserved.

Manage security policies

Hypershield security policies (`SmartSwitchNetworkPolicy`) define which traffic is allowed between network objects. You can manage them through the Timescape user interface or the Timescape public REST API.

For background on the policy data model and lifecycle, see Security policies and Lifecycle.

In the Timescape UI, you manage security policies from the Network Security ‣ Policies page.

Note

This page includes API alternatives to web UI actions. See Timescape API usage for getting started with the Timescape public REST API or go directly to the API reference.

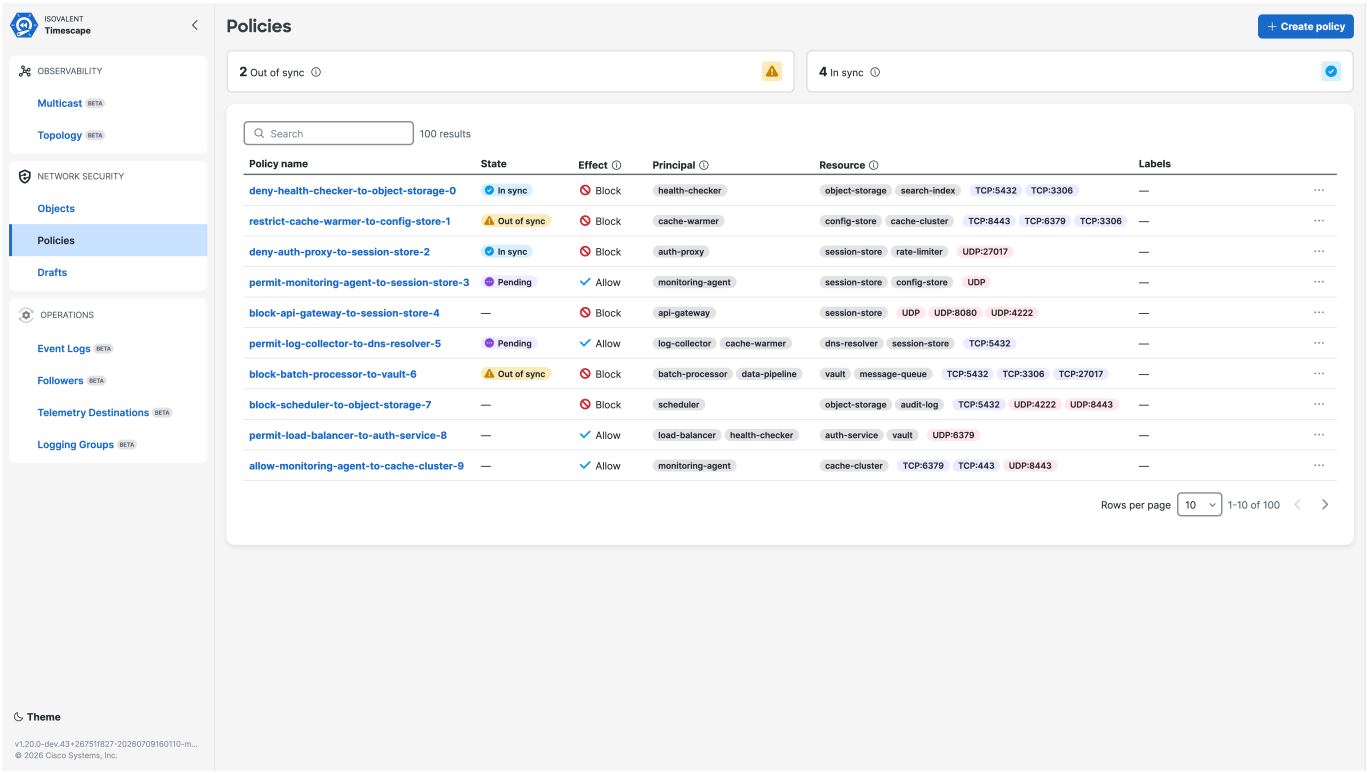
Public API mutations write directly to desired state. They cannot add changes to a Timescape draft.

View the policies list

Use the API

Navigate to Network Security ‣ Policies to see all Hypershield policies.

© 2026 Cisco and/or its affiliates. All rights reserved.



The policies list shows each policy with its name, enforcement state, and summary information.

The page includes summary cards at the top showing:

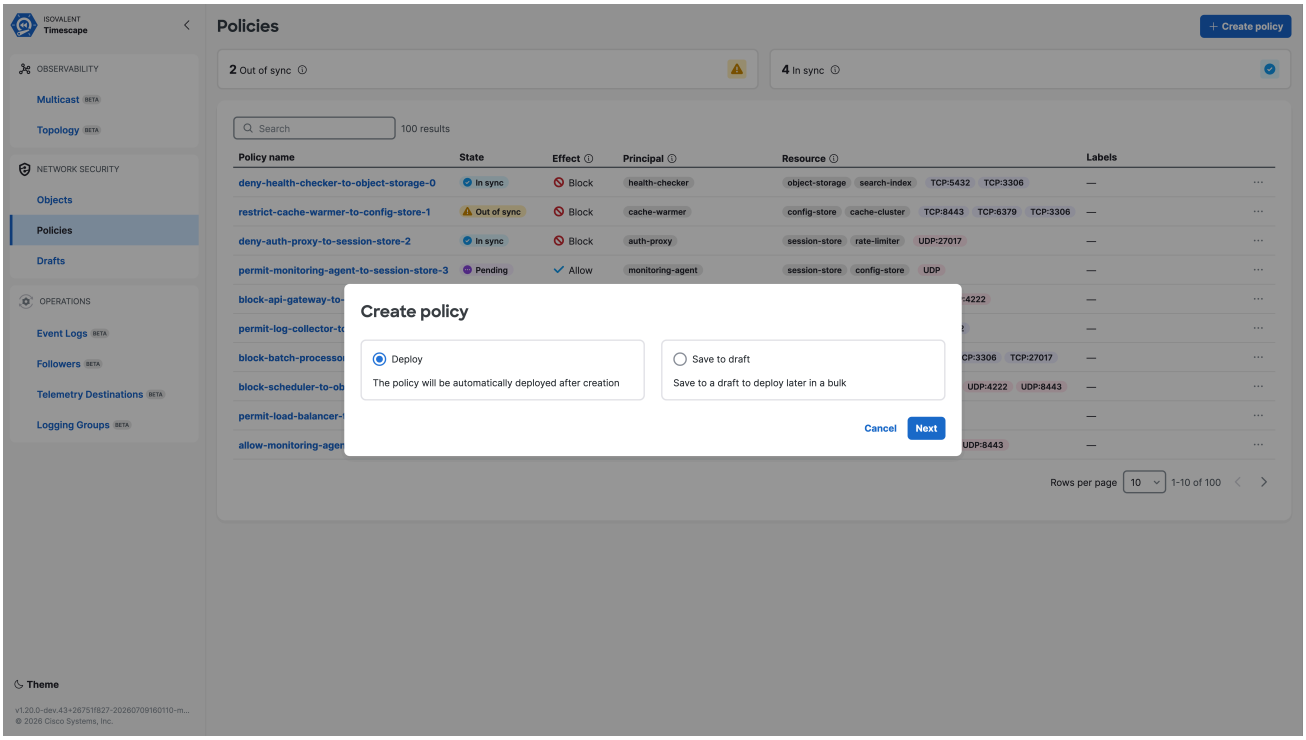
- Policies in sync with enforcement points
- Policies out of sync (pending deployment or with errors)

Create a security policy

Use the API

1. Click **Create policy**.
2. Select a deployment target (save directly or add to a draft, see the draft section).

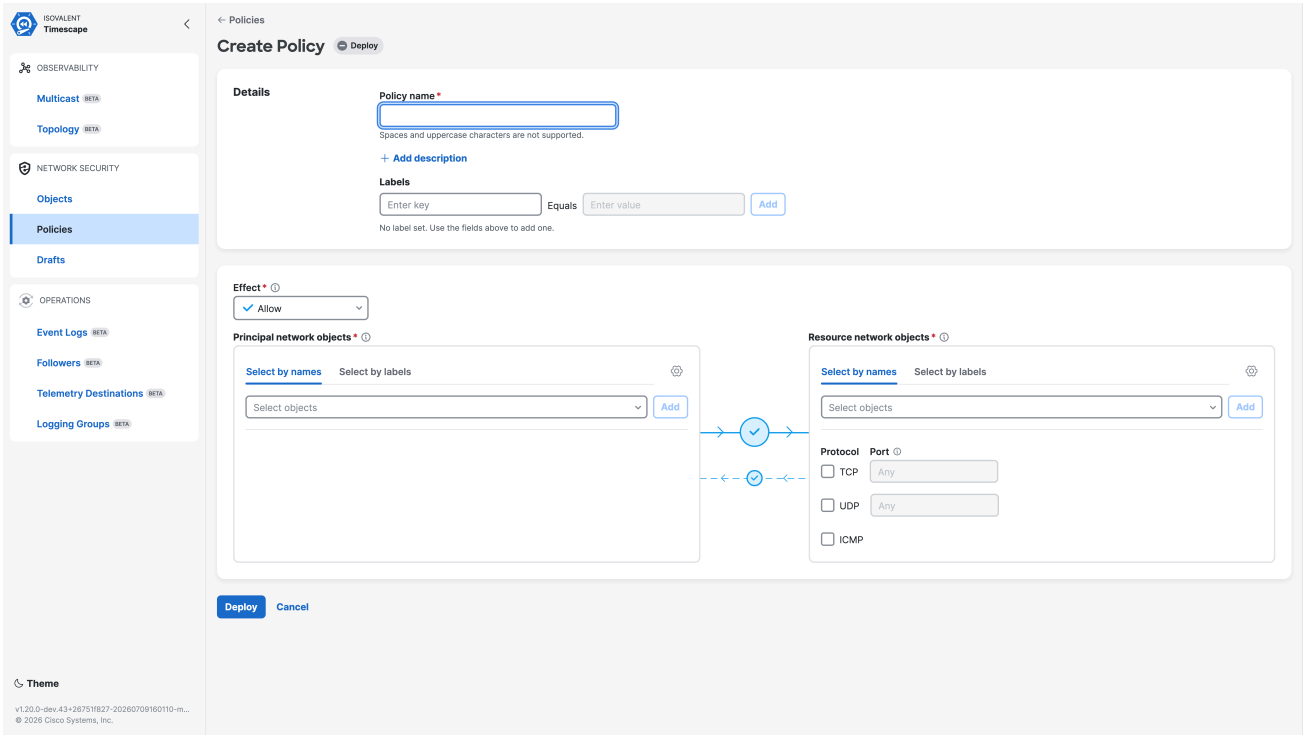
© 2026 Cisco and/or its affiliates. All rights reserved.



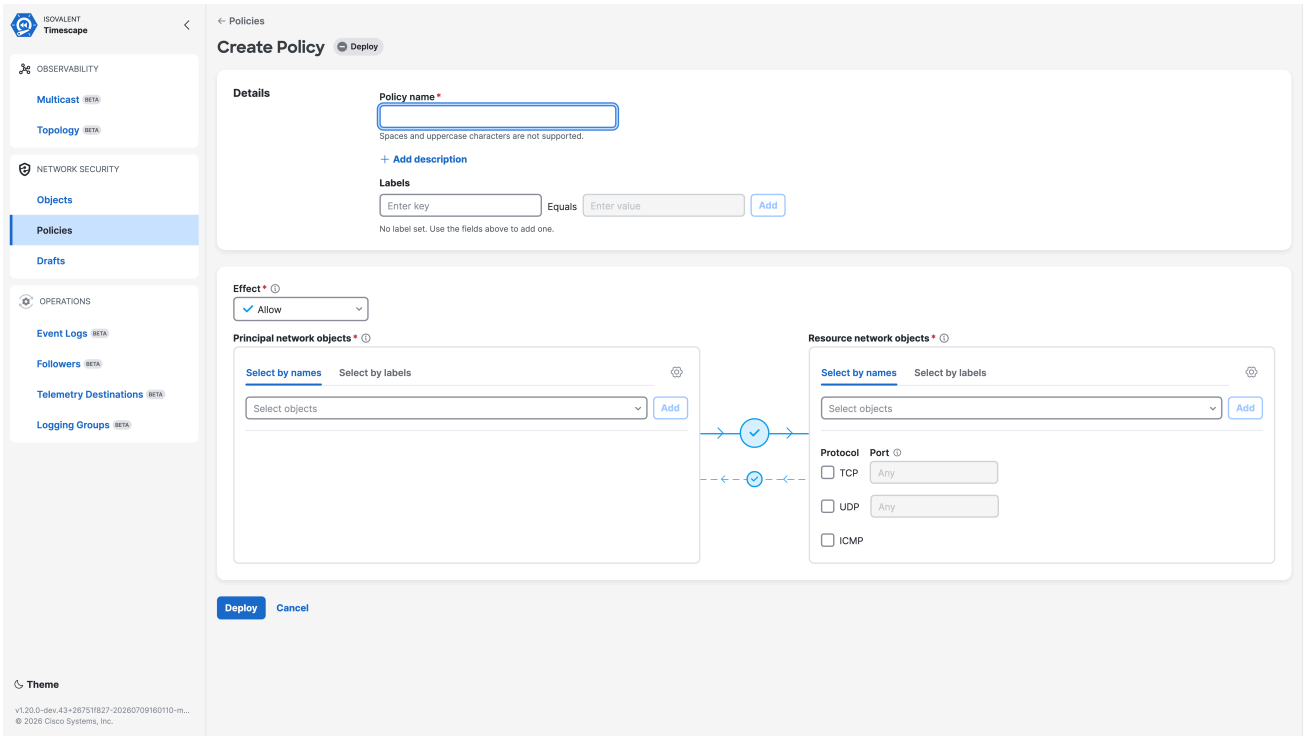
Choose whether to save the policy directly or add it to a draft for staged review.

3. Fill in the policy details:
    - **Name** – A unique name for the policy.
    - **Rules** – Define one or more rules, each with:
        - **Action** – Allow or deny.
        - **Source** – Select a network object or specify an inline CIDR.
        - **Destination** – Select a network object or specify an inline CIDR.
        - **Protocol and ports** – TCP or UDP with specific port numbers.

© 2026 Cisco and/or its affiliates. All rights reserved.



Define rules by selecting source and destination objects and specifying allowed protocols and ports.



The object selector lets you pick from existing network objects or enter inline CIDRs.

4. Click **Deploy**.

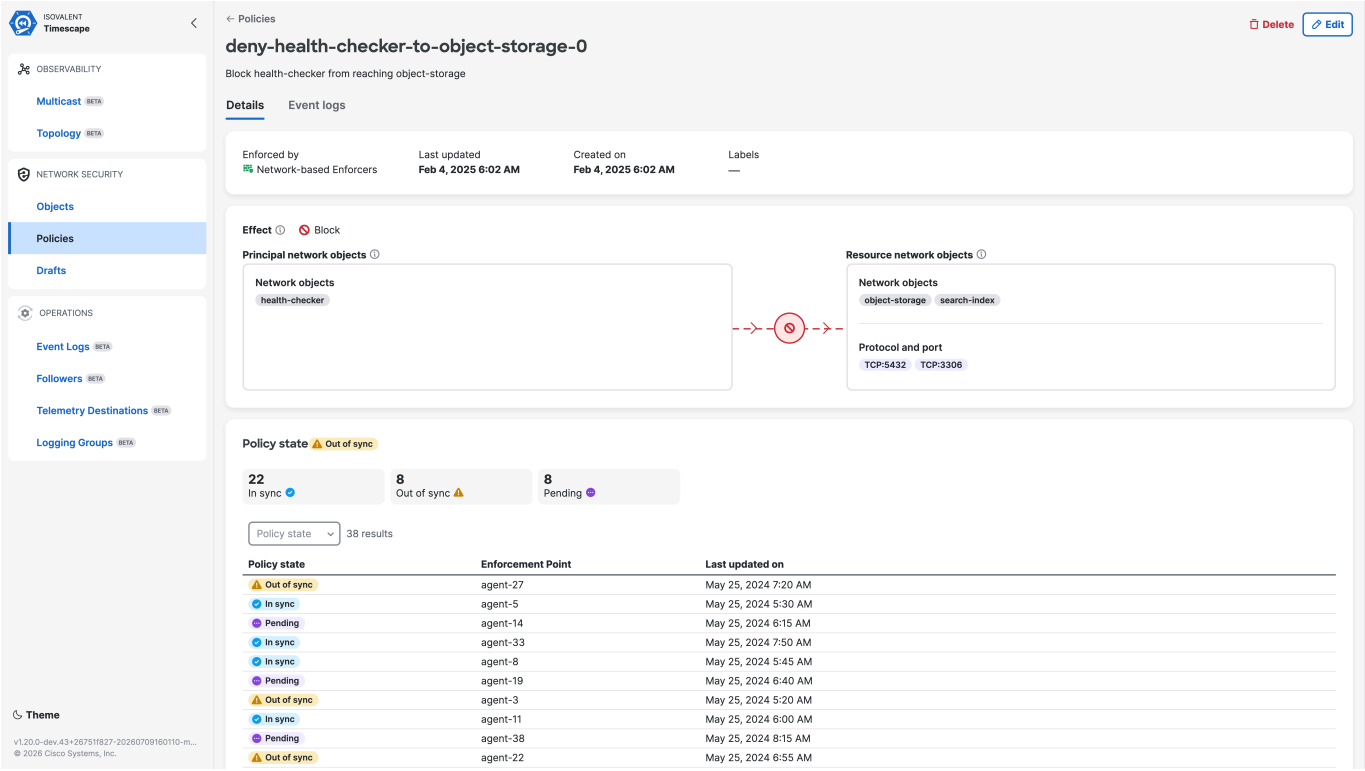
© 2026 Cisco and/or its affiliates. All rights reserved.

View policy details

Use the API

Click a policy name in the list to open its detail page. The detail page shows:

- Policy metadata (name, namespace, creation and modification times)
- Enforcement state across all enforcement points
- The complete rule set



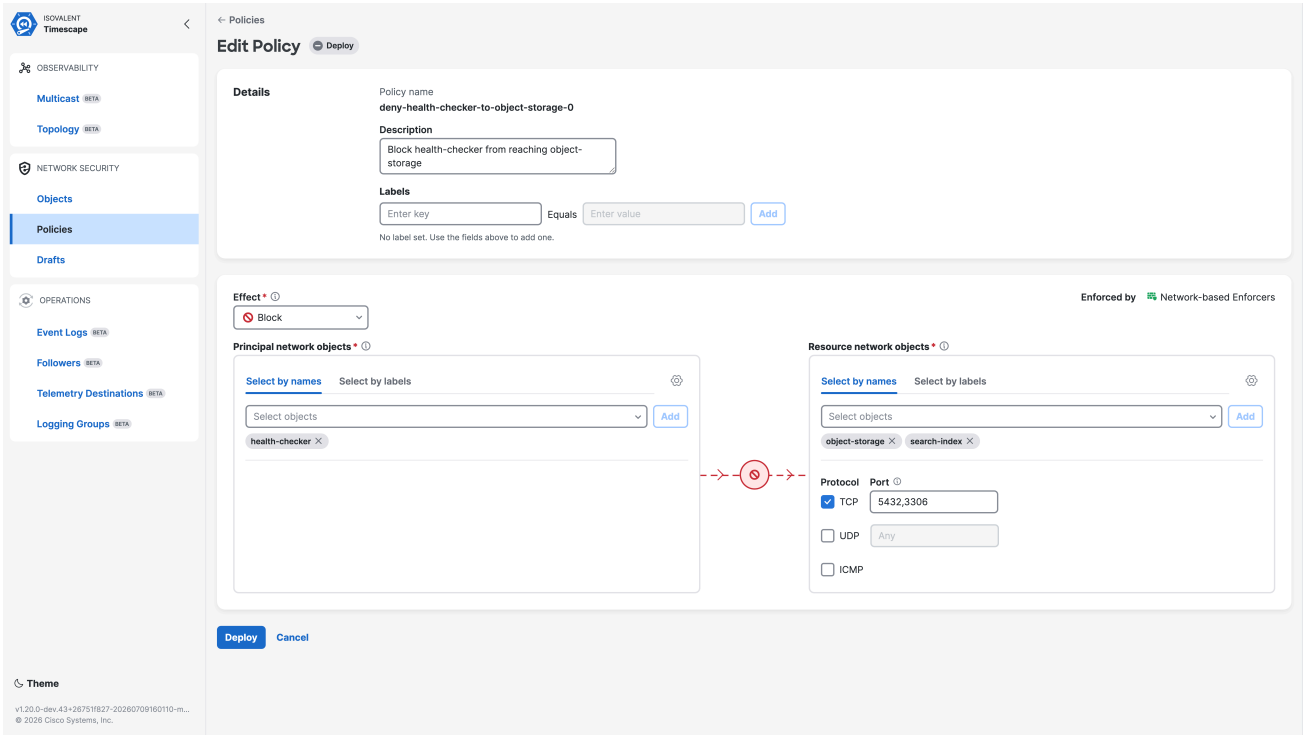
The detail page shows the policy rules and their enforcement state.

Edit a security policy

Use the API

1. From the detail page, click **Edit**, or use the actions menu in the list page.
2. Modify rules, sources, destinations, or ports as needed.

© 2026 Cisco and/or its affiliates. All rights reserved.



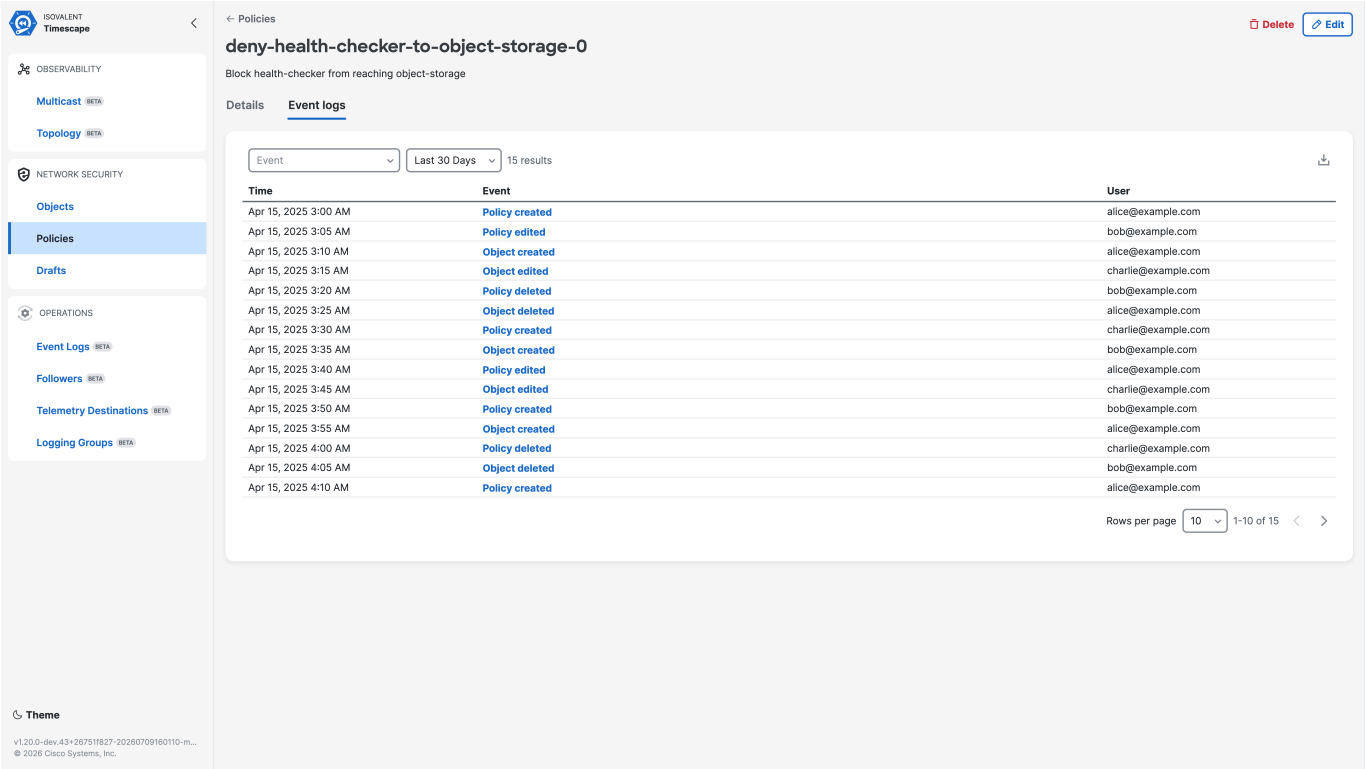
Edit the rules of an existing policy.

3. Click **Deploy**.

View policy event logs

From the policy detail page, switch to the **Logs** tab to review the event history (creation, modifications, deployments, deletions).

© 2026 Cisco and/or its affiliates. All rights reserved.



The logs tab shows the change and deployment history for the policy.

Delete a security policy

Use the API

Use the actions menu on the list page or the detail page to delete a policy.

Caution

Deleting a policy can change which traffic is allowed. Review the effect of the deletion before deleting the policy through either interface.

© 2026 Cisco and/or its affiliates. All rights reserved.

Validate policies with drafts

Before deploying policy changes to live enforcement, use Timescape drafts to stage, review, and validate them as a group.

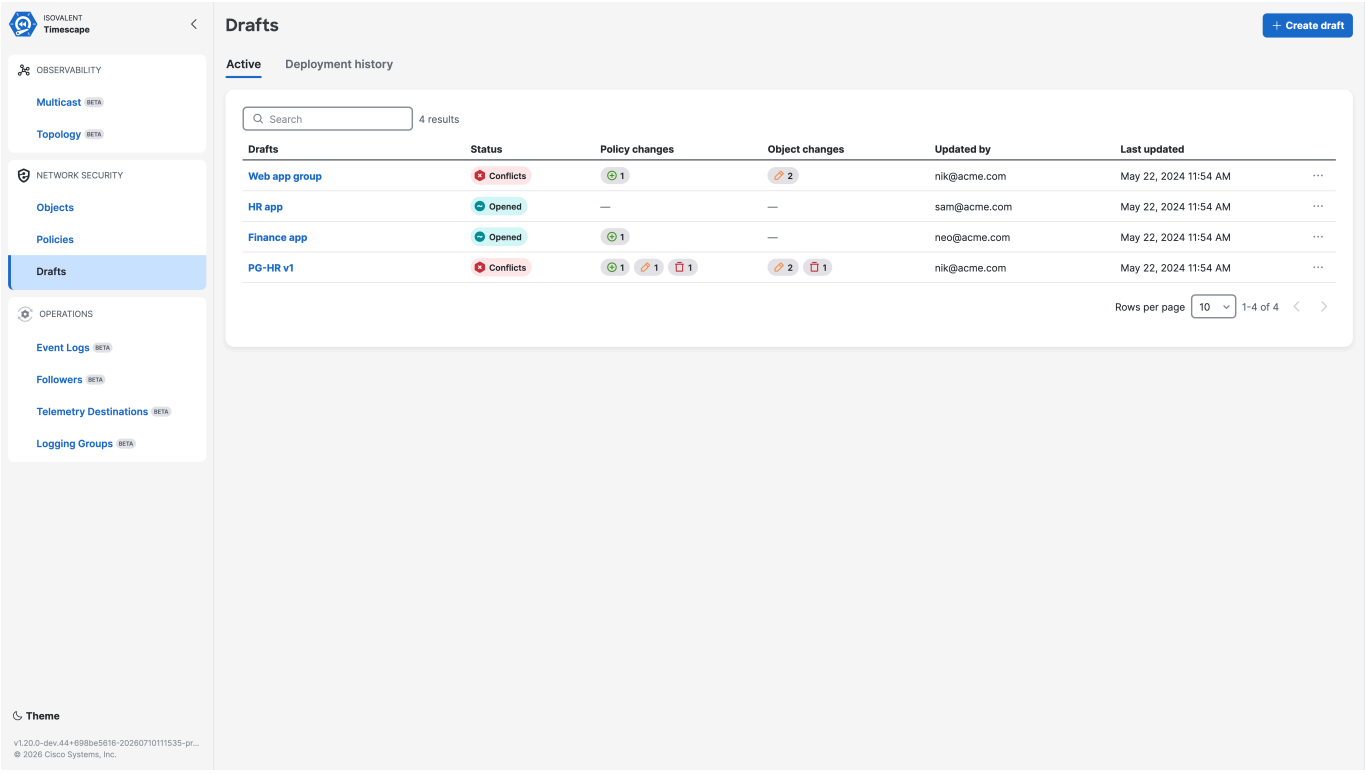
For background on the staged policy lifecycle, see Lifecycle

Drafts let you stage policy and object changes before deploying them to live enforcement. A draft groups one or more changes (creates, edits, deletes) into a single deployment unit that you can review, validate, and deploy as a batch.

Navigate to Network Security ▸ Drafts to access the drafts feature.

View active drafts

The Drafts page opens on the **Active** tab, which lists all drafts that have not yet been deployed or discarded.

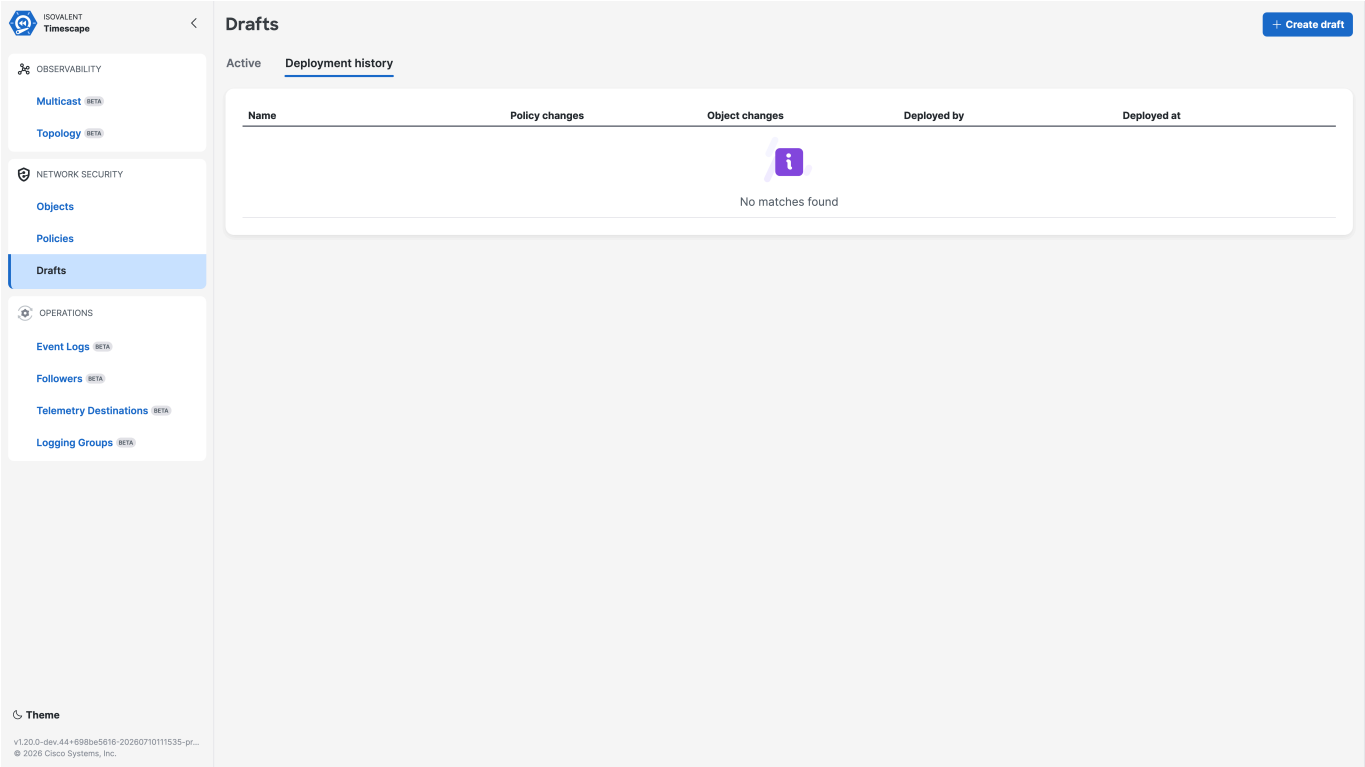


The Active tab shows all in-progress drafts with their name, change count, and status.

View deployment history

Switch to the **Deployment history** tab to see previously deployed drafts and their outcomes.

© 2026 Cisco and/or its affiliates. All rights reserved.

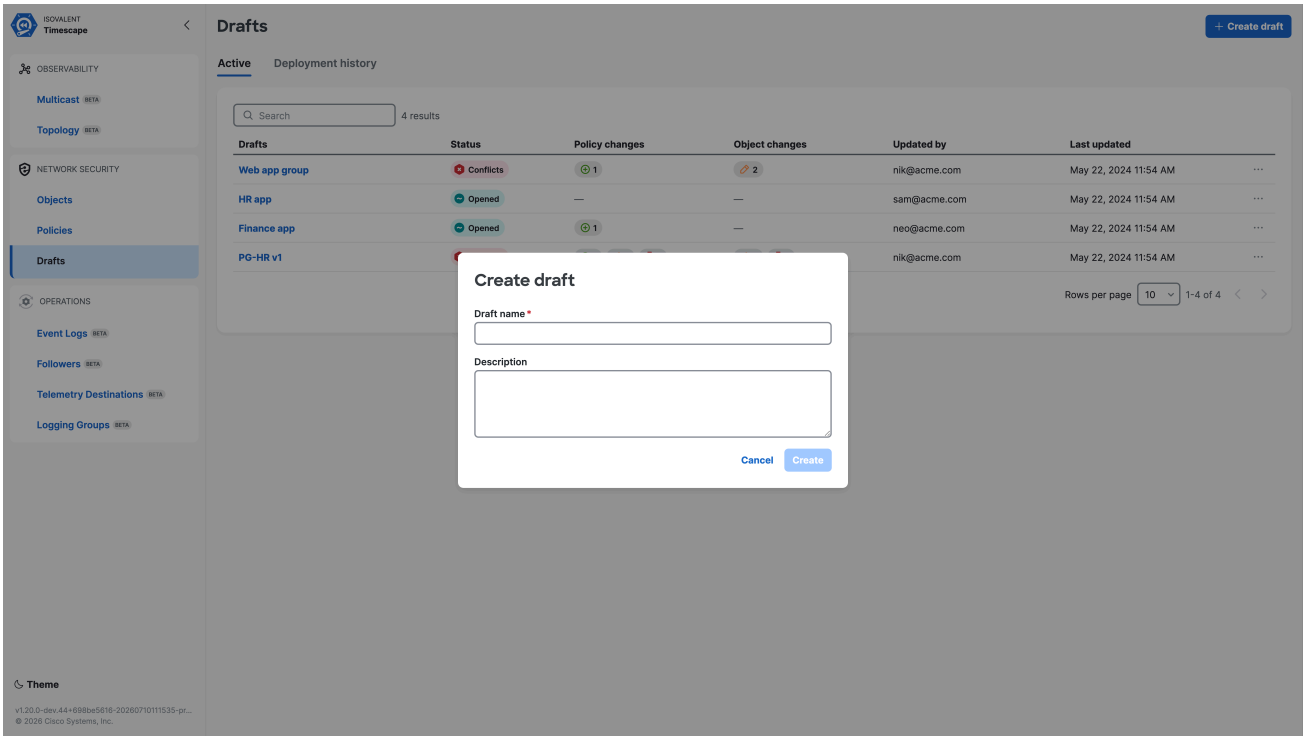


The Deployment history tab lists past deployments with their timestamp and result.

Create a draft

1. Click **Create draft** .
2. In the modal, enter a name and optional description for the draft.

© 2026 Cisco and/or its affiliates. All rights reserved.



Provide a name and optional description for the new draft.

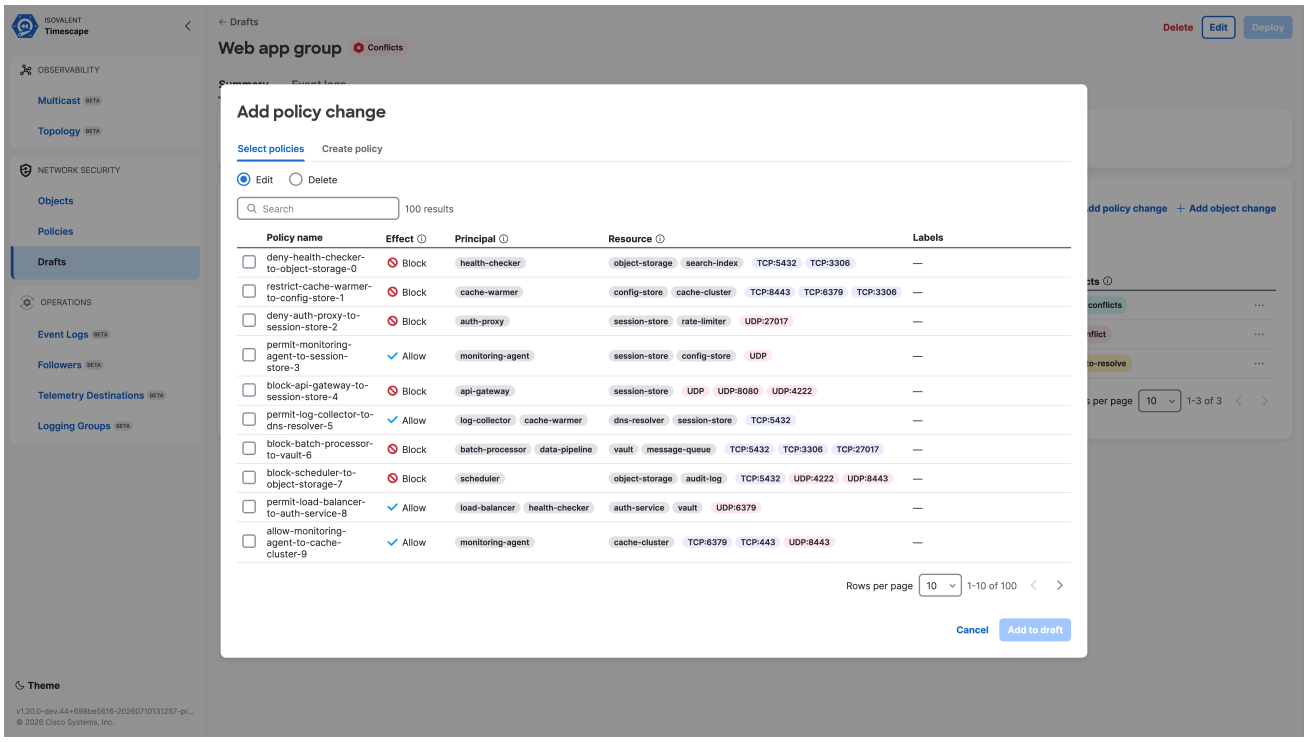
3. Click **Create**. You are redirected to the draft detail page.

Add changes to a draft

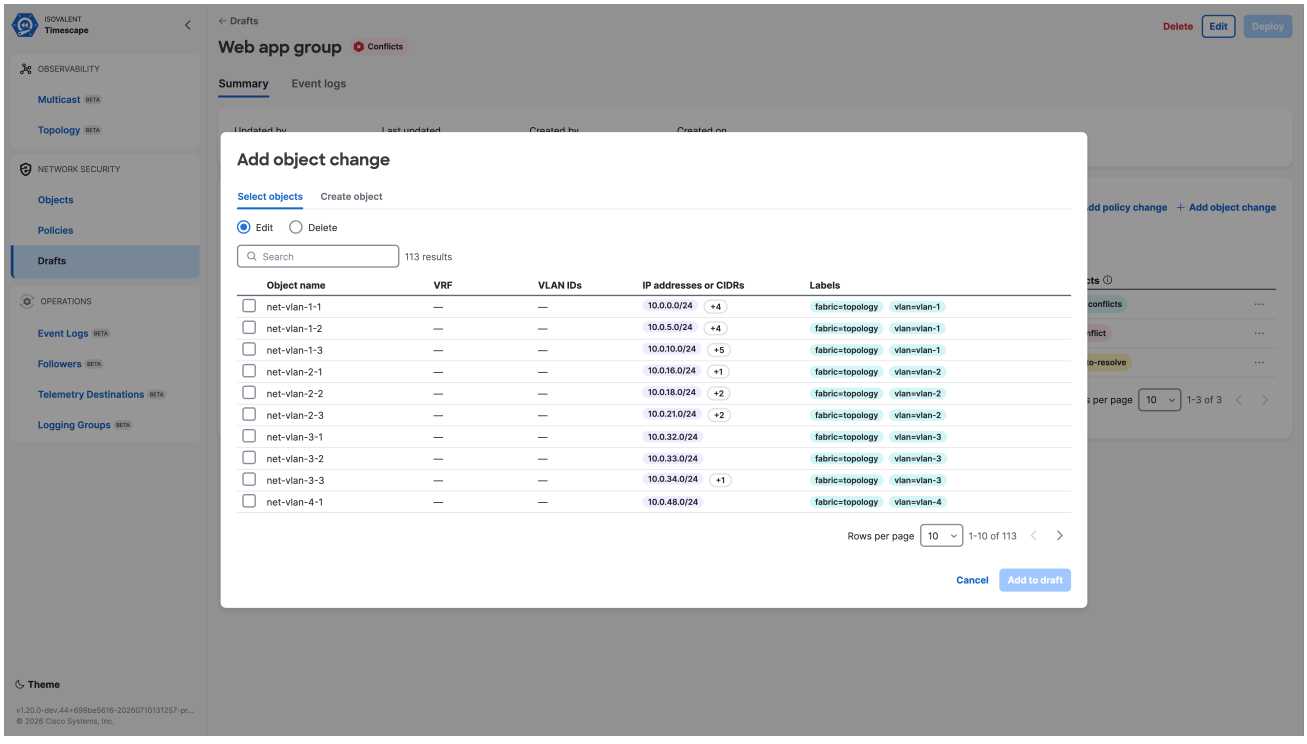
From the draft detail page, you can add policy and object changes:

1. Click **Add policy change** or **Add object change**.
2. In the modal that appears, select an existing resource to modify or delete, or create a new one.

© 2026 Cisco and/or its affiliates. All rights reserved.



Select a policy to modify or create a new policy within this draft.



Select an object to modify or create a new object within this draft.

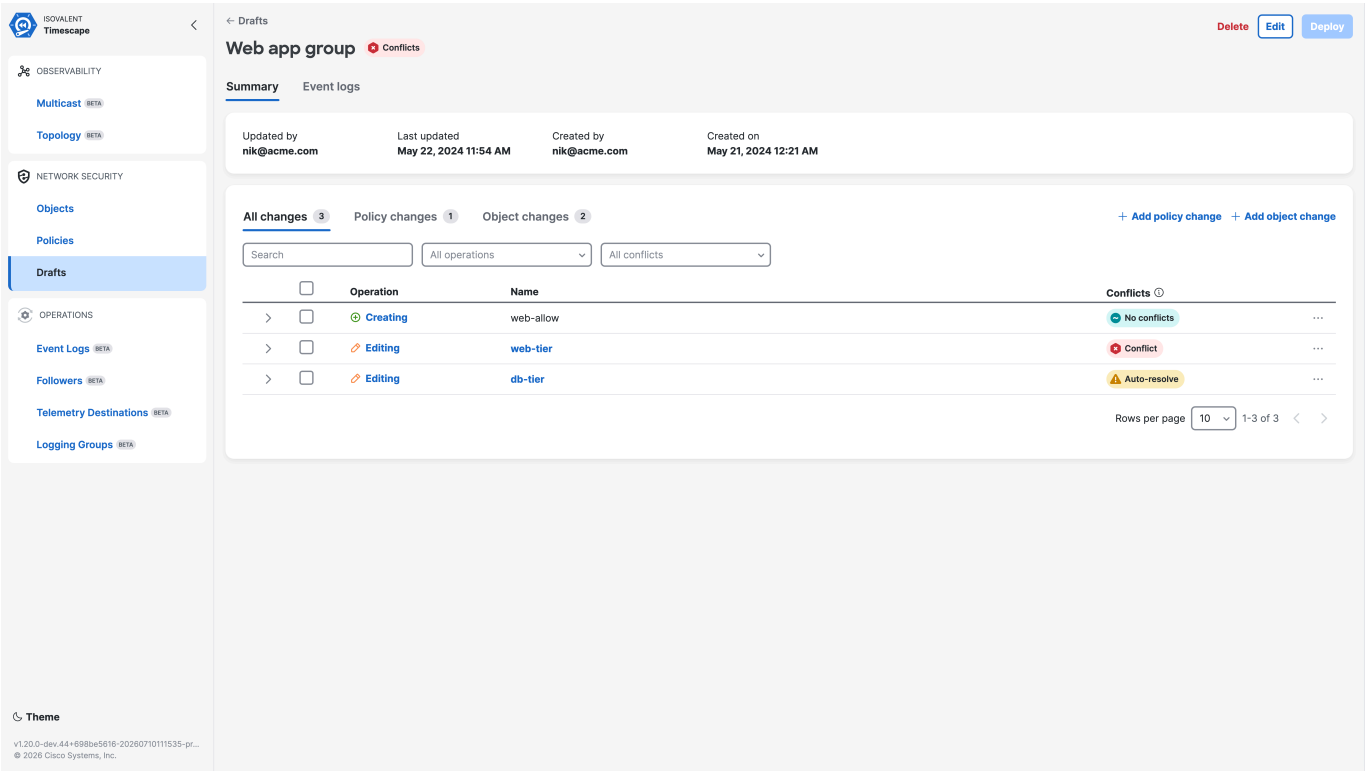
3. Make your changes in the form and save.

© 2026 Cisco and/or its affiliates. All rights reserved.

The draft detail page shows all pending changes in a table.

Review draft changes

The draft detail page displays all changes grouped by type:

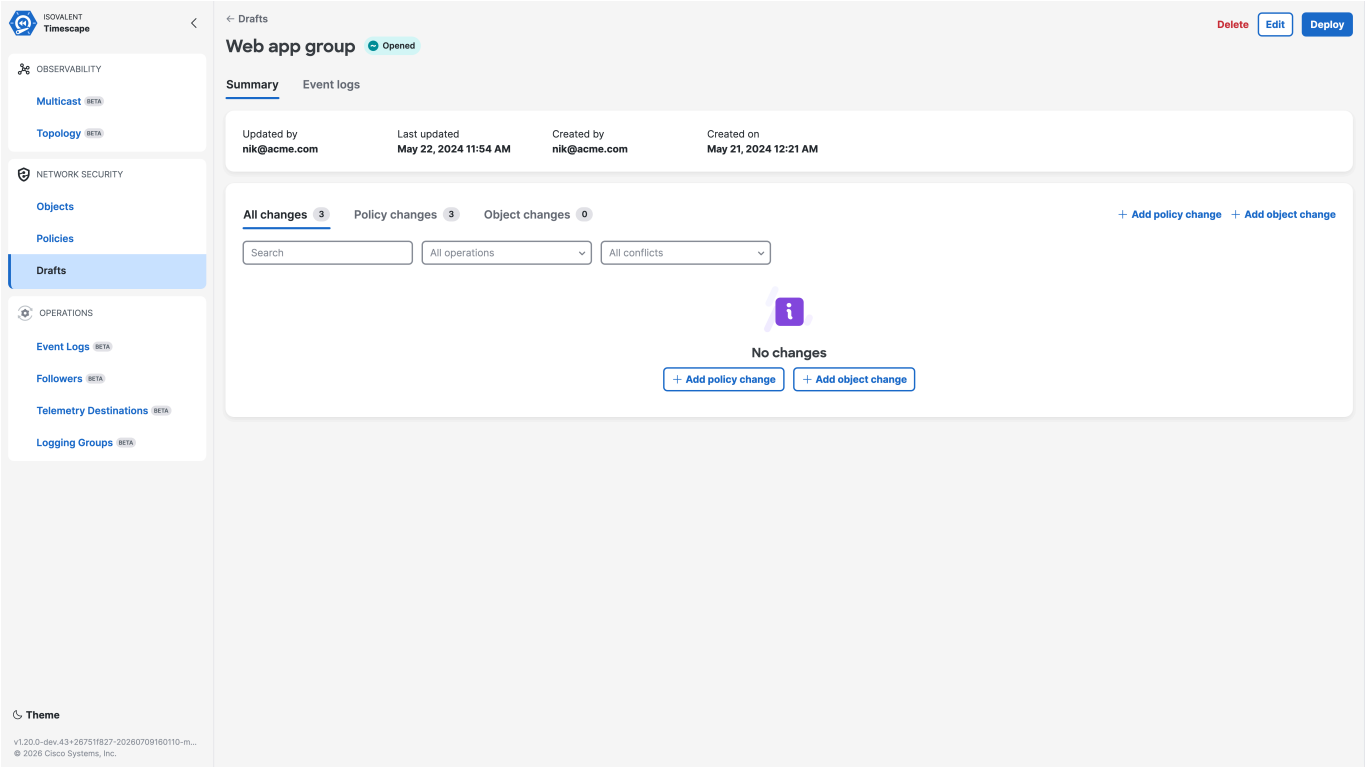


The changes table shows each pending modification with its operation type (create, edit, delete) and resource name.

You can filter changes by:

- **Kind** – Show only policies, only objects, or all changes.
- **Operation** – Filter by create, edit, or delete operations.
- **Conflicts** – Show changes that conflict with the current live state (another user modified the same resource since the draft was created).
- **Search** – Filter by resource name.

© 2026 Cisco and/or its affiliates. All rights reserved.



Use filters to focus on conflicting changes within a large draft.

Resolve conflicts

When a resource in a draft has been modified in the live environment since the draft was created, the change shows a conflict indicator. Conflicts can be:

- **Auto-resolvable** – The draft change and live change do not overlap; the system can merge them automatically.
- **Manual resolution** – The changes overlap and require you to choose which version to keep.

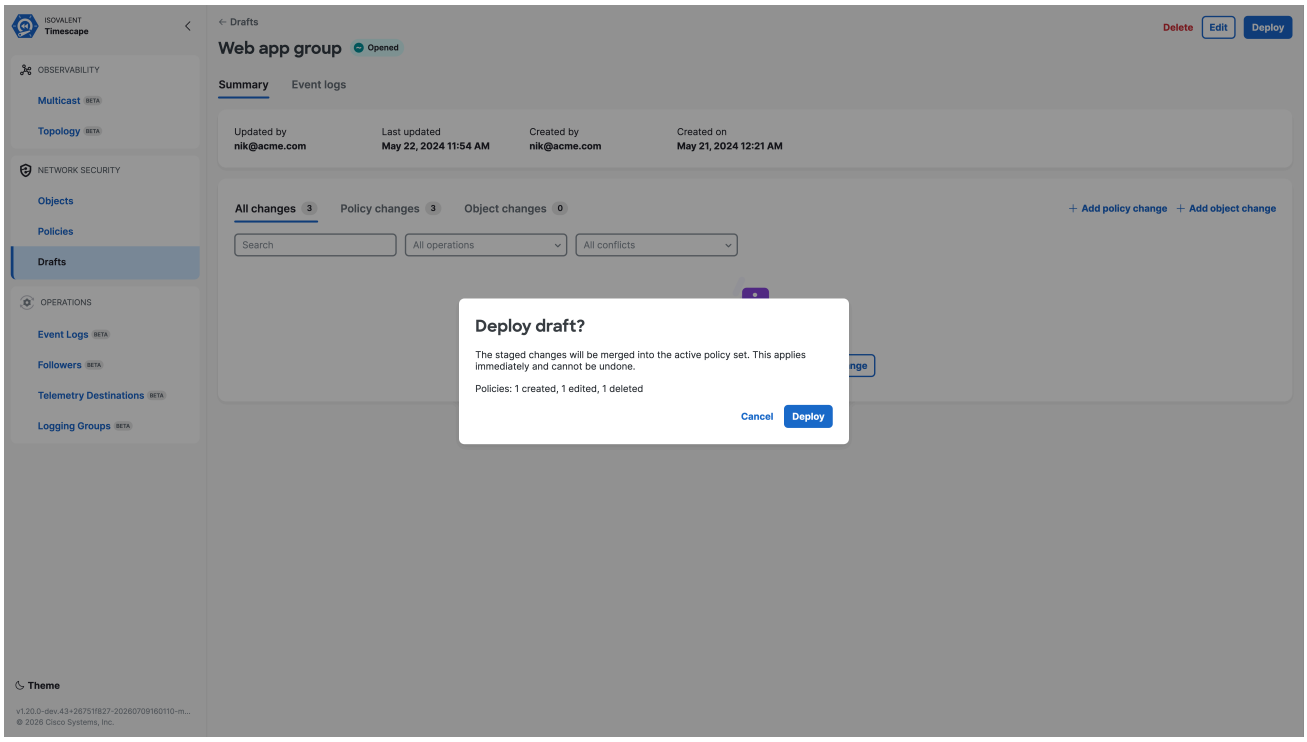
Review conflicting changes and resolve them before deploying the draft.

Deploy a draft

When all changes are reviewed and conflicts are resolved:

1. Click **Deploy** in the draft header.
2. Confirm the deployment in the modal.

© 2026 Cisco and/or its affiliates. All rights reserved.



Review the deployment summary and confirm to apply all changes to live enforcement.

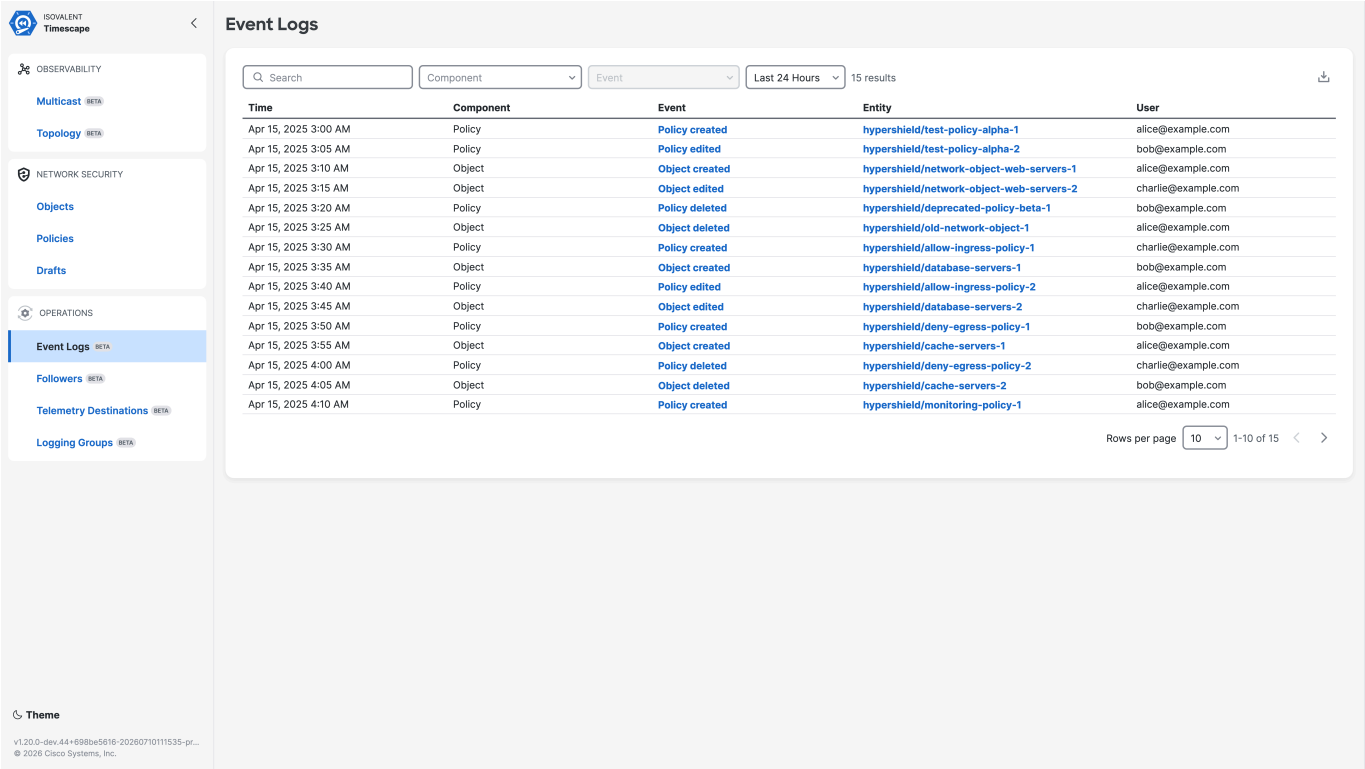
3. The draft moves to the deployment history with a success or failure status.

> **Note**
>
> Deploying a draft applies all its changes atomically. If any change fails validation, the entire deployment is rejected.

View draft event logs

From the draft detail page, switch to the **Logs** tab to see the event history for this draft (creation, changes added, deployment attempts).

© 2026 Cisco and/or its affiliates. All rights reserved.



The logs tab shows the complete activity history for the draft.

Delete a draft

Use the delete action in the draft header or from the drafts list. Deleting a draft discards all its pending changes without applying them.

© 2026 Cisco and/or its affiliates. All rights reserved.

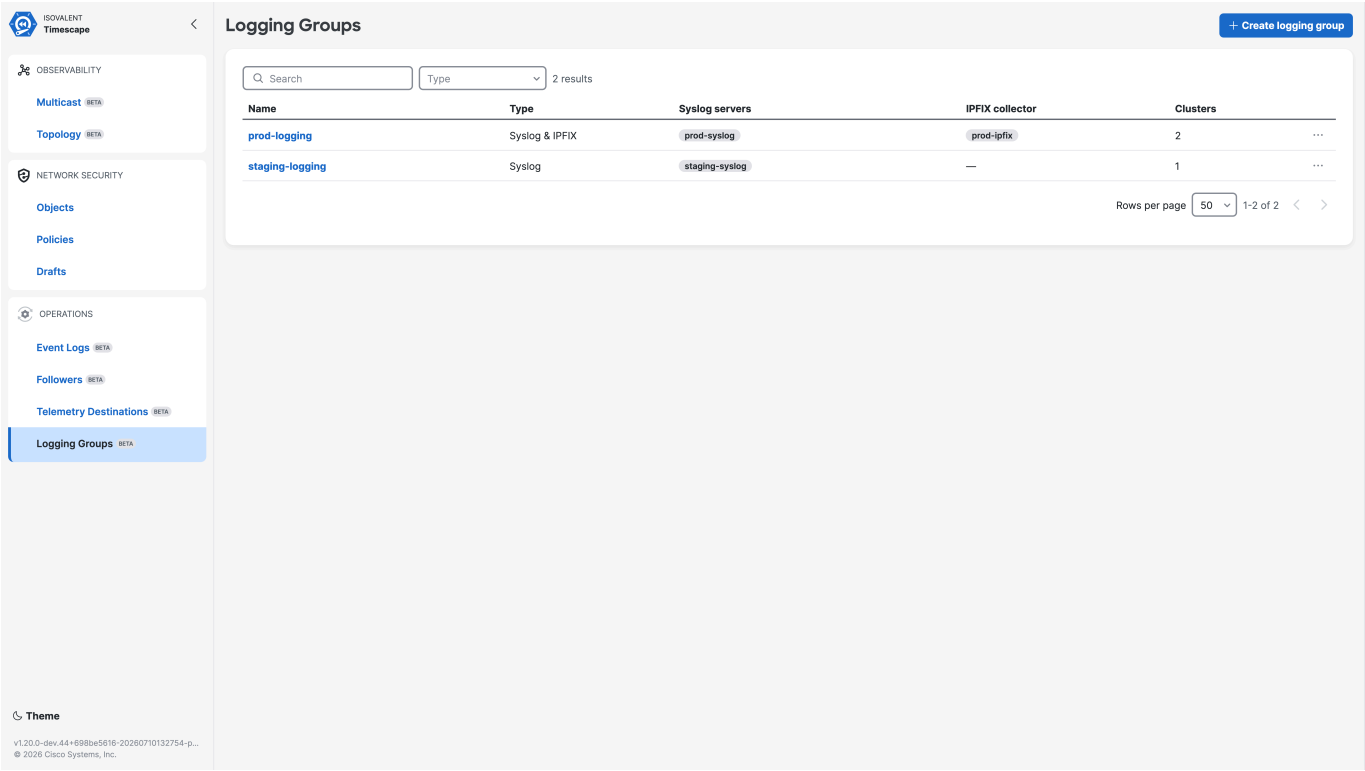
Manage logging groups

Logging groups define which telemetry destinations receive data from which follower clusters. They connect clusters to syslog servers and IPFIX collectors configured as telemetry destinations. In the Timescape UI, you manage them from Operations ▸ Logging Groups.

For background on logging and observability, see Logging and observability.

View the logging groups list

Navigate to Operations ▸ Logging Groups to see all configured logging groups.



The logging groups list shows each group with its name, type, assigned telemetry destinations, and cluster count.

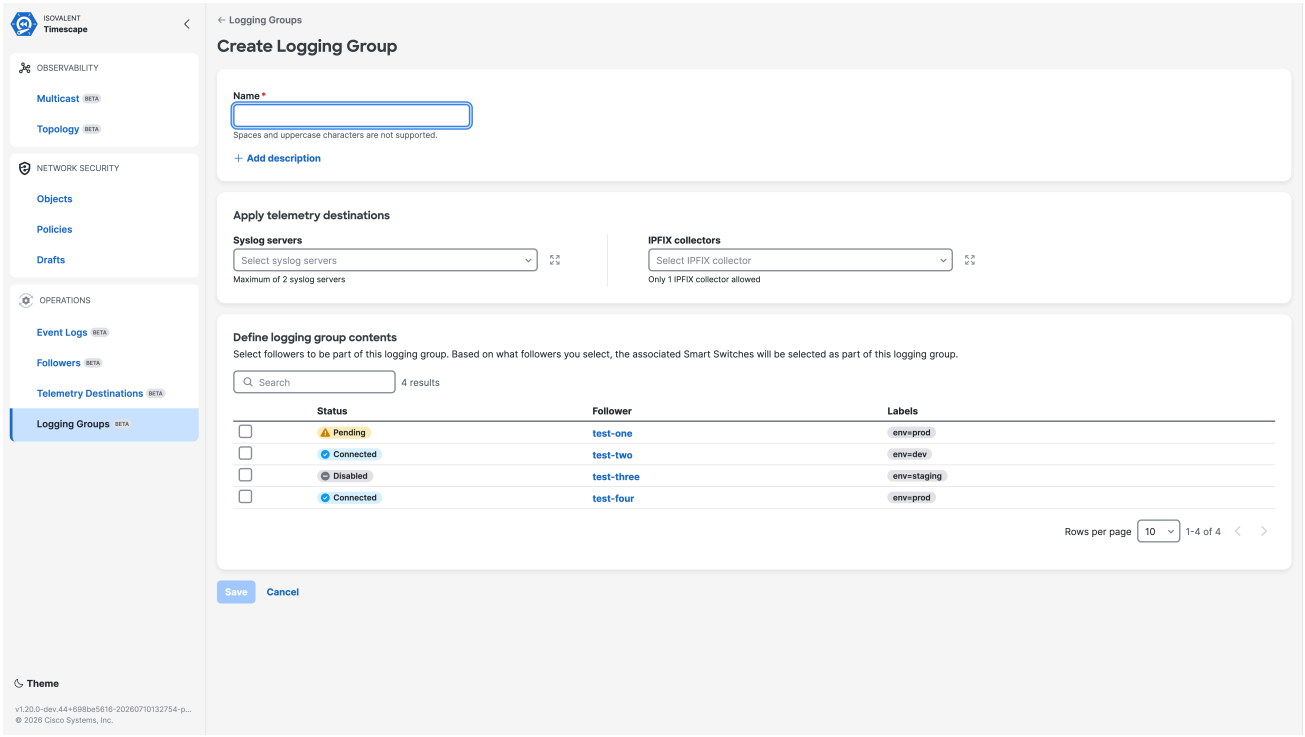
The list supports:

- **Type filter** – Filter by Syslog or IPFIX type.
- **Search** – Filter groups by name.
- **Actions menu** – Edit or delete groups from the row action menu.

© 2026 Cisco and/or its affiliates. All rights reserved.

Create a logging group

1. Click **Create logging group**. You will see the logging group creation form.



Enter the group name and select the telemetry type.

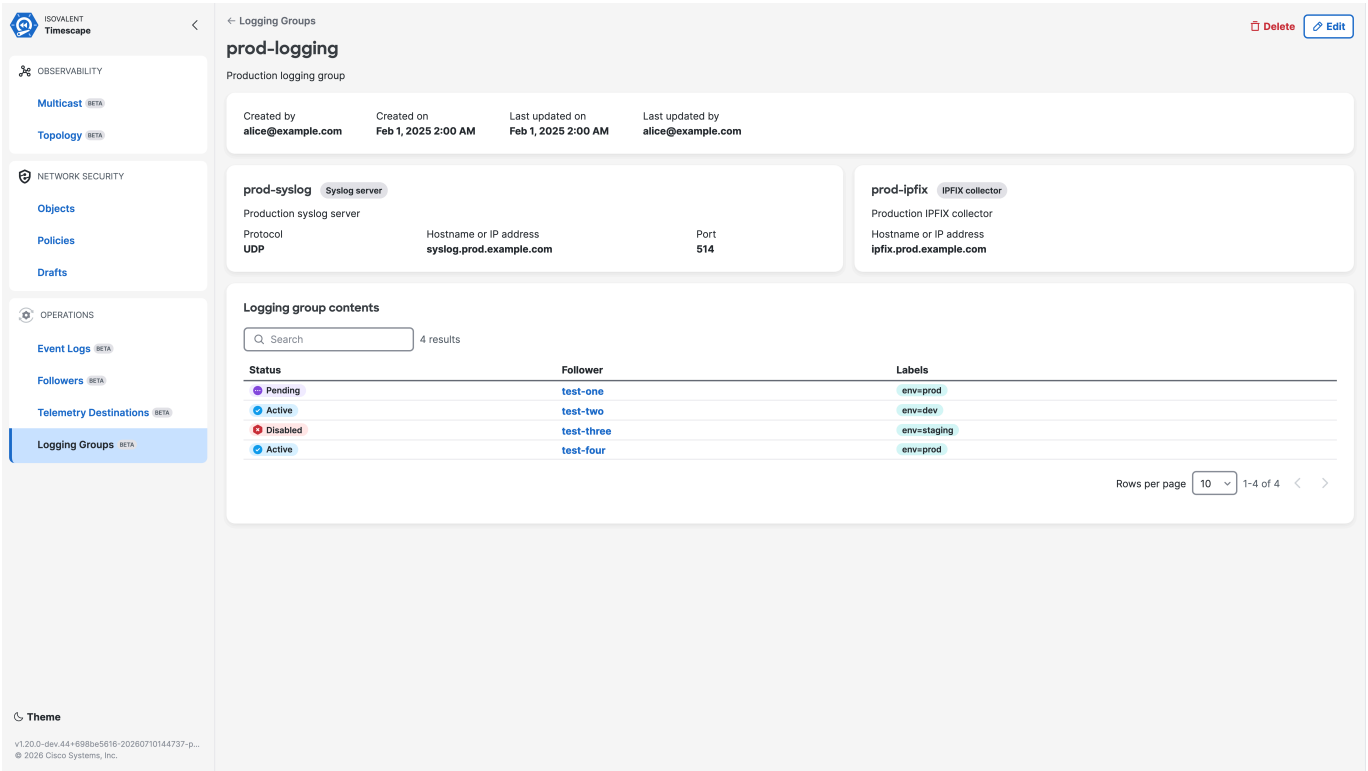
2. Fill in a unique name for the logging group.
3. **Assign the telemetry destination this group sends data to. These are**
   the servers that will receive data from clusters in this group. These are the servers that will receive data from clusters in this group and can be of type Syslog or IPFIX.
4. **Select which follower clusters belong to this logging group. Those**
   clusters will export telemetry to the assigned destinations.
5. Click **Save**.

View logging group details

Click a group name in the list to open its detail page. The detail page shows:

- Group metadata (name, type, creation time)
- Assigned telemetry destinations
- Member clusters

© 2026 Cisco and/or its affiliates. All rights reserved.

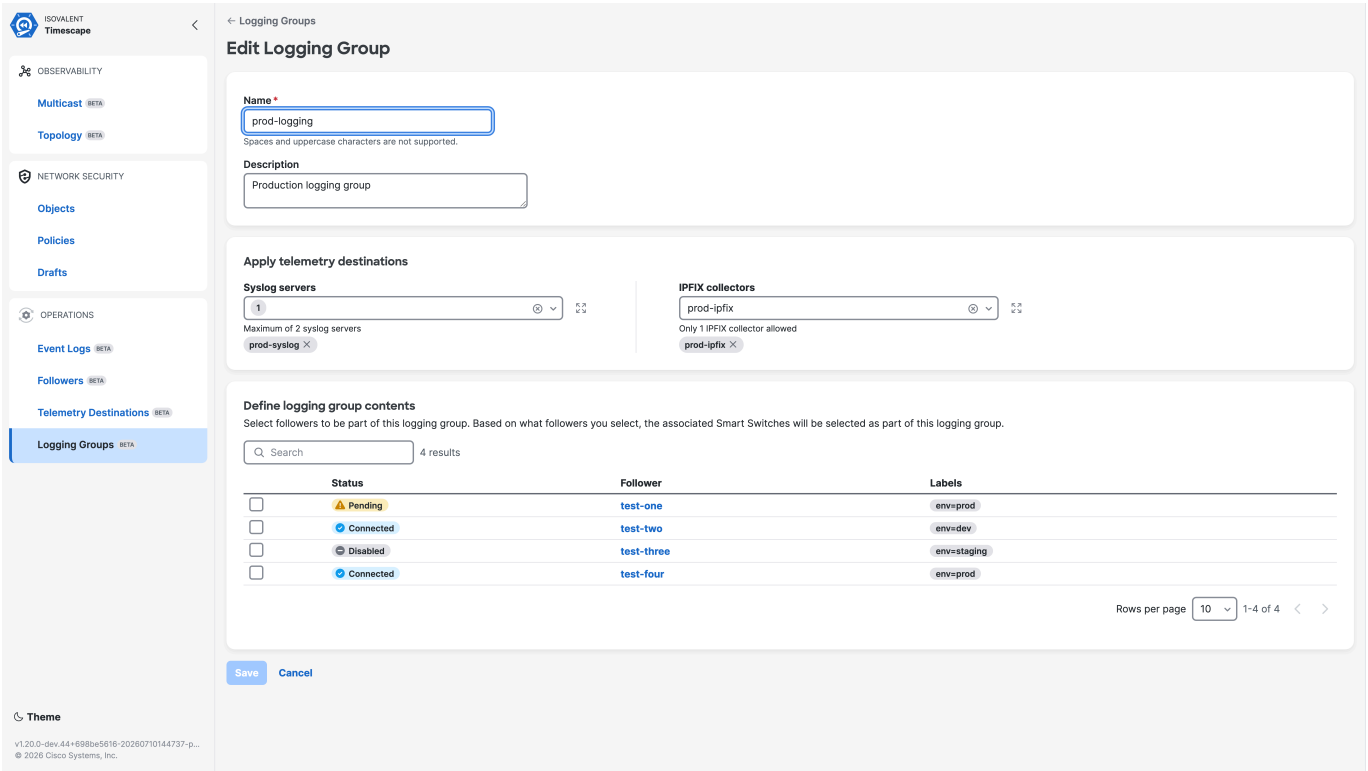


The detail page shows the group's destinations and assigned clusters.

Edit a logging group

From the detail page, click **Edit** to modify the group's destinations or cluster assignments.

© 2026 Cisco and/or its affiliates. All rights reserved.



Modify destination assignments or cluster membership.

Delete a logging group

Use the actions menu on the list page or the detail page to delete a group. Deleting a logging group stops telemetry export for the affected clusters and destinations.

© 2026 Cisco and/or its affiliates. All rights reserved.

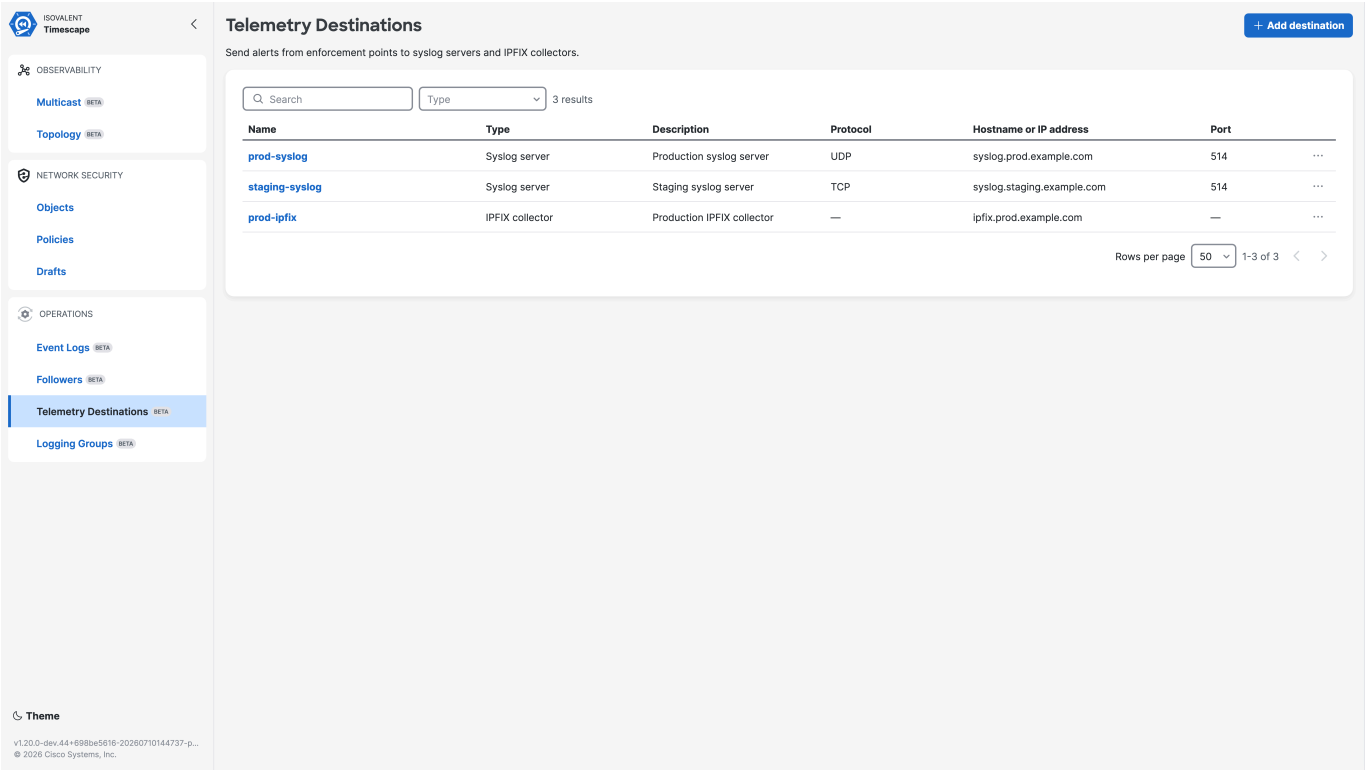
Manage telemetry destinations

Telemetry destinations define where enforcement points send syslog messages and IPFIX flow records. Each destination represents a single collector endpoint (a syslog server or an IPFIX collector). In the Timescape UI, you manage them from Operations ▸ Telemetry Destinations.

For background on telemetry export, see Logging and observability.

View the telemetry destinations list

Navigate to Operations ▸ Telemetry Destinations to see all configured destinations.



The destinations list shows each endpoint with its name, type, protocol, hostname, and port.

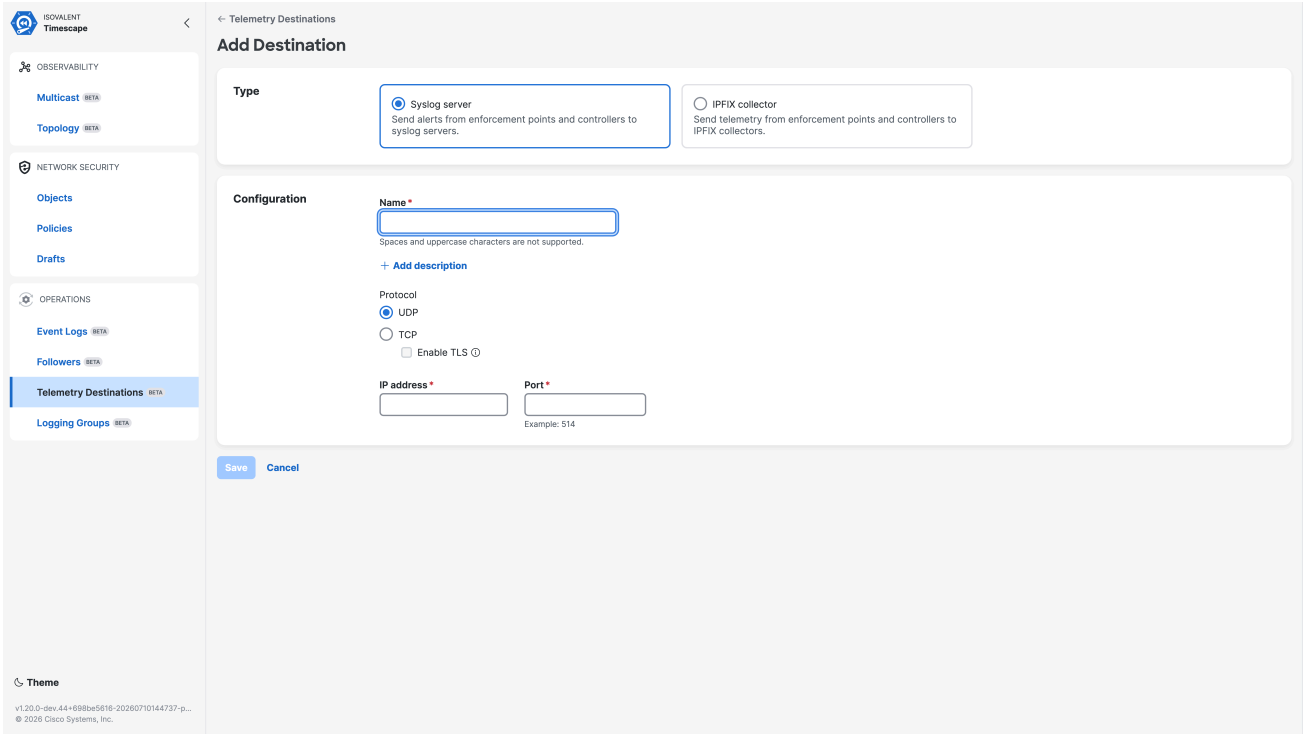
The list supports:

- **Type filter** – Filter by Syslog or IPFIX type.
- **Search** – Filter destinations by name.
- **Actions menu** – Edit or delete destinations from the row action menu.

© 2026 Cisco and/or its affiliates. All rights reserved.

Create a telemetry destination

1. Click **Create telemetry destination**.
2. Fill in the destination details:
   - **Name** – A unique name for the destination.
   - **Description** (optional) – Additional context about the destination.
   - **Type** – Select Syslog or IPFIX.
   - **Protocol** – The transport protocol (TCP or UDP).
   - **Hostname / IP** – The collector's address.
   - **Port** – The collector's listening port.



Enter the destination name, type, protocol, hostname, and port.

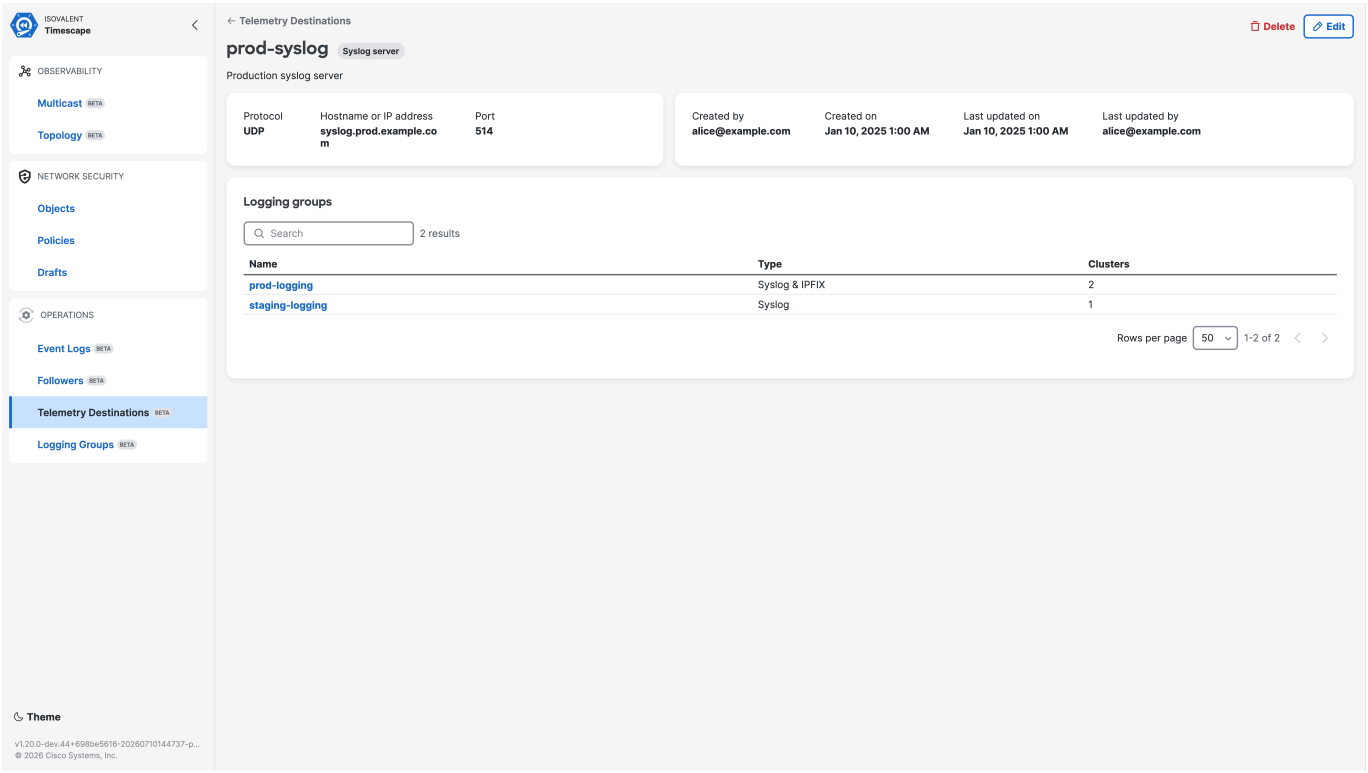
3. Click **Save**.

View destination details

Click a destination name in the list to open its detail page. The detail page shows:

- Destination metadata (name, type, protocol, hostname, port)
- Logging groups that reference this destination

© 2026 Cisco and/or its affiliates. All rights reserved.

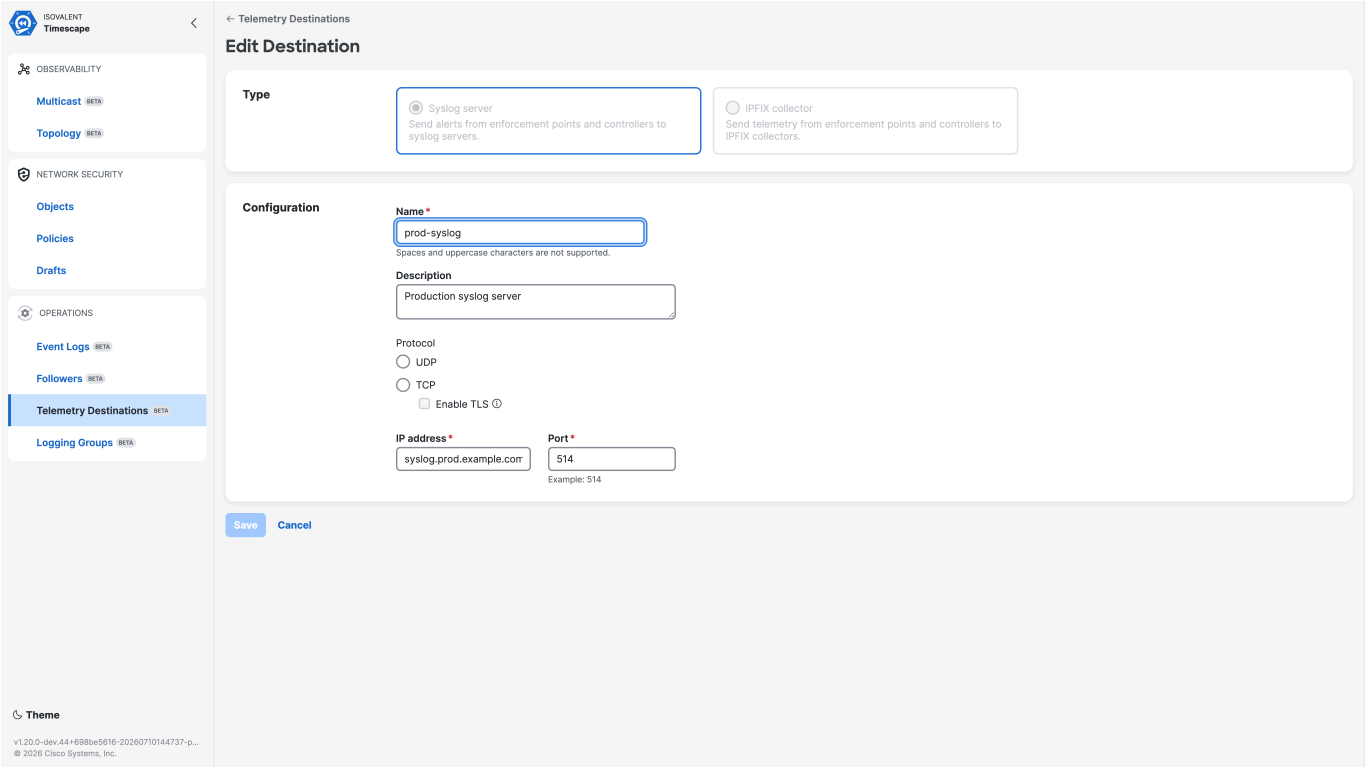


The detail page shows the destination's configuration and which logging groups reference it.

Edit a telemetry destination

From the detail page, click **Edit** to modify the destination's connection parameters.

© 2026 Cisco and/or its affiliates. All rights reserved.



Update the destination's protocol, hostname, port, or description.

Delete a telemetry destination

Use the actions menu on the list page or the detail page to delete a destination.

> **Caution**
>
> Remove the destination from all logging groups before deleting it. Deleting a destination that is still referenced by a logging group will leave that group with an unresolvable reference.

© 2026 Cisco and/or its affiliates. All rights reserved.

Remove a controller

When running, the Hypershield controller (Leader or Follower cluster) creates various disks to persist internal data.

The `isoctl` tool ensures to delete all VMs and associated disks from your VMware environment.

Warning

Uninstalling a controller will have significant adverse effects on your infrastructure. Beyond the loss of controller data, you must prepare for the following:

- If a Leader controller is removed, enforcement points (such as Smart Switches) will no longer receive security policy changes. This is because Follower clusters, which manage enforcement points, will no longer receive updates from the missing Leader. Important: Existing policies will continue to be enforced.
- If a Follower cluster is removed, associated enforcement points will no longer receive security policy changes made on the Leader. Important: Existing policies will continue to be enforced.

Make a data backup of the target controller before proceeding with its removal. See Back up Hypershield management and control planes for instructions.

Prerequisites

- The Hypershield controller is running.
- You have access to the bootstrap YAML files.
- You have access to the Kubconfig file.

Select the target controller

Perform these steps on the management VM.

Note

The instructions below target a Follower cluster. File paths and names are used as examples, so adapt them according to your environment.

Switch to the Follower cluster context:

```
export KUBECONFIG="/home/iep/hypershield-follower.yaml"
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Set up your GOVC environment variables according to your environment

```
export GOVC_URL='vcenter.example.com'
export GOVC_USERNAME='administrator@vsphere.local'
export GOVC_PASSWORD='your-password'
export GOVC_DATACENTER='Datacenter1'
export GOVC_INSECURE=true   # Use with self-signed certificates
```

Tip

Use the snippet below to read the vSphere configuration from your bootstrap file and set the GOVC variables. The management VM ships with the `yq` utility by default.

> Ensure that the `BOOTSTRAP_PATH` variable matches your bootstrap file path.
>
> ```
> export BOOTSTRAP_PATH="/home/iep/hypershield/bootstrap_follower.yaml"
>
> export GOVC_URL="$(yq '.vsphere.endpoint' $BOOTSTRAP_PATH)"
> export GOVC_USERNAME="$(yq '.vsphere.username' $BOOTSTRAP_PATH)"
> export GOVC_PASSWORD="$(yq '.vsphere.password' $BOOTSTRAP_PATH)"
> export GOVC_INSECURE=1
> export GOVC_DATACENTER="$(yq '.vsphere.datacenter' $BOOTSTRAP_PATH)"
> ```

Delete the controller cluster

Uninstalling a controller means deleting its Kubernetes cluster. The following command deletes the cluster (and associated resources). Before taking effect, you must confirm the operation by typing the cluster name.

```
isoctl mgmtcluster delete
```

Expected output:

```
Nuke: powering off VM /Datacenter1/vm/hypershield/hypershield-follower-1-infra-6t9nc
(source=vspheremachine)
Nuke: destroying VM /Datacenter1/vm/hypershield/hypershield-follower-1-infra-6t9nc
Nuke: destroyed VM /Datacenter1/vm/hypershield/hypershield-follower-1-infra-6t9nc
...
Nuke: VM destroy complete (destroyed=5 not_found=0)
Deleting 6 FCD volumes from vSphere...
Deleted PV pvc-0b939827-b5c7-4ff7-8e7e-719448c6e1b5 (handle=e376ee68-76ed-4be2-bc2a-88547c3ce6d3
claim=registry/data-hypershield-follower-1-registry-harbor-redis-0)
...
```

Delete the Kubeconfig file

© 2026 Cisco and/or its affiliates. All rights reserved.

```
rm -rf /home/iep/hypershield-follower.yaml
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Create security policies

Smart Switch policies

Smart Switch as VTEP leaf in a VXLAN fabric

The Smart Switch can be used as a VTEP leaf in a VXLAN fabric, and this case, when an inter-VRF policy is created with two VXLAN VRFs, AGW (Hypershield Agent Gateway) on the Smart Switch will automatically add **two** additional intra-VRF rules for VXLAN VRFs (one intra-VRF rule for each VXLAN VRF) to enable layer-3 routing between the VTEP leaf Smart Switches in the VXLAN fabric. This automatic addition of two additional intra-VRF rules for VXLAN VRFs is done in AGW when the action in the inter-VRF policy is either `Allow` or `Deny`.

To enforce a `Deny` inter-VRF policy with VXLAN VRFs on the DPU, it is recommended not to create such a policy, as it will automatically be enforced in the DPU using the implicit/default `Deny` rule.

Below are the example NXOS configurations for a VXLAN VRF on the Smart Switch:

```
configure terminal
feature nv overlay

vrf context vxlanVRF
  vni 101 l3

interface nve1
  host-reachability protocol bgp
  member vni 101 associate-vrf

interface <interface number>
  vrf member vxlanVRF

show running-config nv overlay
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Back up and restore Hypershield

This section covers backing up and restoring the Hypershield management and control planes. This topic is separate from similar operations that may be applied to enforcements points like a Smart Switch.

Important

Backup and restore operations are executed as the root user (through sudo). Use the commands with care.

© 2026 Cisco and/or its affiliates. All rights reserved.

Back up Hypershield management and control planes

Hypershield management and control planes are backed up by targeting the appropriate Kubernetes cluster. For the Leader and Follower deployment model:

- management plane: Leader cluster
- control plane: Follower cluster

The backup script used in the procedures is called `hs-backup.sh` (it resides on the management VM by default). Use the `--dry-run` option for testing. See the dropdown for usage details.

`hs-backup.sh`

Here is what is captured in each backup archive:

Content	Leader cluster	Follower cluster
Backup metadata (`manifest.json`)	Yes	Yes
Checksums (`checksums.sha256`)	Yes	Yes
Kubernetes Secrets (encrypted)	Both namespaces	`hypershield` namespace only
PostgreSQL database dump	Yes	No
CRD policy objects	Reference snapshot	Primary intent
ConfigMaps and RBAC	Both namespaces	`hypershield` namespace only
Helm versions	Yes	Yes
Pod logs (with `--include-logs`)	Timescape pods	HyperShield pods

Caution

When creating a backup you are prompted for a passphrase (twice). Store the passphrase separately from its archive as the passphrase decrypts the Kubernetes Secrets bundle. Also store the archive in a safe place as it contains unencrypted sensitive information.

Back up the Leader cluster

Run the following to back up the Leader cluster:

```
sudo KUBECONFIG="<leader kubeconfig>" \
  hypershield/scripts/hs-backup.sh --appliance-type leader
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Example successful output:

```
[21:10:44] INFO  Checking available disk space...
...
Enter backup passphrase:
...
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
Backup complete (leader): ./hypershield/hypershield-backup-20260603-211044-leader.tar.gz
SHA256: 8e9290d9493669e8860798cfae16ac1f220cfe94831e35c1e76d5e69d9a7ba81
...
```

Back up the Follower cluster

Run the following to back up the Follower cluster:

```
sudo KUBECONFIG="<follower kubeconfig>" \
  hypershield/scripts/hs-backup.sh --appliance-type follower
```

Example successful output:

```
[21:10:44] INFO  Checking available disk space...
...
Enter backup passphrase:
...
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
Backup complete (follower): ./hypershield/hypershield-backup-20260603-211044-follower.tar.gz
SHA256: 8e9290d9493669e8860798cfae16ac1f220cfe94831e35c1e76d5e69d9a7ba81
...
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Restore Hypershield to original IP addressing

This page shows you how to restore the Hypershield management and control planes where the new underlying infrastructure makes use of the same IP addresses and DNS names as the original infrastructure.

Warning

Restore is a destructive operation and must be performed with due caution.

These instructions apply to a topology where the management plane is backed by a single `leader` Kubernetes cluster and the control plane is backed by one or more `follower` Kubernetes clusters. See Leader and follower clusters for deployment concepts.

There are three scenarios covered in this document:

1. leader-only restore
2. follower-only restore
3. leader and follower restore

Note

You will need an understanding of Leader and Follower cluster YAML configuration. See Install Hypershield on Leader and Follower for details.

The restore script used in the procedures is called `hs-restore.sh` (it resides on the management VM by default). Use the `--dry-run` option for testing. See the dropdown for usage details.

`hs-restore.sh`

1. Leader-only restore

This section shows you how to restore a lost leader cluster from a previously made backup onto a freshly-deployed replacement cluster that uses the original cluster's IP addressing and DNS records.

Prerequisites

The following prerequisites are needed to complete the steps:

- a leader cluster backup file is available

© 2026 Cisco and/or its affiliates. All rights reserved.

- the follower cluster is running and pointing at the (now-lost) leader cluster
- a new base Kubernetes cluster is deployed with the original leader cluster's IP address and DNS hostname

1.1 Deploy Hypershield onto the new leader cluster

On the management VM, run the following command to deploy a new leader cluster:

```
isoctl hypershield install -f /home/iep/hypershield/hypershield-install-new-leader.yaml
```

See troubleshooting tips.

1.2 Apply the backup file

Apply the leader cluster backup file:

```
sudo KUBECONFIG=<leader kubeconfig> ./hs-restore.sh --file <backup.tar.gz> \
  --mode rollback --appliance-type leader --force
```

See troubleshooting tips.

1.3 Verify the environment

Confirm all leader Kubernetes pods reach a Running/Ready status:

```
kubectl get pods -n hubble-timescape
```

Sample output:

```
NAME                                          READY   STATUS      RESTARTS      AGE
chi-hubble-timescape-hubble-data-0-0-0        1/1     Running     0             46m
hubble-timescape-analyzer-8488ffcc67-mbjc4    1/1     Running     0             7m13s
hubble-timescape-gateway-5ff76557ff-2h28n     1/1     Running     0             15m
hubble-timescape-ingester-798459d5c9-tcnrx    1/1     Running     0             7m12s
hubble-timescape-migratepg-b9e9a4391b-5n4sr   0/1     Error       0             62m
hubble-timescape-migratepg-b9e9a4391b-dmkdv   0/1     Completed   0             60m
hubble-timescape-pg-1                         1/1     Running     0             60m
hubble-timescape-pg-2                         1/1     Running     0             57m
hubble-timescape-public-api-575b47bcff-gjhcn  1/1     Running     0             12s
hubble-timescape-server-cdb4c4b74-kq4qb       1/1     Running     0             7m13s
hubble-timescape-trimmer-29753340-c2xlv       0/1     Completed   0             40m
hubble-timescape-ui-6ff788668d-zdcs5          1/1     Running     0             15m
sc-server-574669d455-knpsn                    1/1     Running     7 (46m ago)   52m
sc-server-migratepg-833b59182b-7czh5          0/1     Completed   0             52m
```

© 2026 Cisco and/or its affiliates. All rights reserved.

1.4 Reconnect the follower

On the restored leader, export fresh TLS material:

```
kubectl --kubeconfig=<new leader kubeconfig> get secret \
  -n hubble-timescape sc-server-root-ca-tls \
  -o jsonpath='{.data.ca\.crt}' | base64 -d > sc-server-ca.crt

kubectl --kubeconfig=<new leader kubeconfig> get secret \
  -n hubble-timescape hubble-timescape-server-cert \
  -o jsonpath='{.data.ca\.crt}' | base64 -d > leader-ca.crt
```

In Timescape, generate a new join credential (Operations → Followers).

On the follower, force the follower-join job to re-run by deleting and recreating it from the Helm release hooks (this pair of commands may need to be re-run):

```
isoctl hypershield install -f hypershield-install-follower.yaml
kubectl delete job -n hubble-timescape hubble-timescape-follower-join
```

Sample successful output:

```
level=INFO msg="Follower-join completed successfully" leader-address=hs10-leader.pseudo.co:443 cluster-
id=1f5723c3-80a0-426c-ac3c-e463dc01da05
```

Still on the follower, delete the sc-client-bootstrap secret and restart sc-client:

```
kubectl delete secret -n hubble-timescape sc-client-bootstrap
kubectl rollout restart deployment -n hubble-timescape sc-client
```

Sample successful output:

```
level=INFO msg="tunnel active, serving SOCKS5 connections" subsys=connector leader=198.18.42.158:443
```

1.5 Validate the results in Timescape

In the Timescape web UI, verify that all restored data is present and correct:

- Objects
- Policies
- Event Logs
- Followers

© 2026 Cisco and/or its affiliates. All rights reserved.

- Telemetry Destinations
- Logging Groups

The follower cluster should show as **Active** and any Smart Switches should show as **Connected**.

Below is a screenshot of the Objects page:



2. Follower-only restore

This section shows you how to restore a lost follower cluster from a previously made backup onto a freshly-deployed replacement cluster that uses the original cluster's IP addressing and DNS records.

Prerequisites

The following prerequisites are needed to complete the steps:

- a follower cluster backup file is available
- the leader cluster is running and pointing at the (now-lost) follower cluster
- a new Kubernetes cluster is deployed with the original follower cluster's IP address and DNS hostname
- physical access to Smart Switches previously registered to the (now-lost) follower cluster

2.1 Deploy Hypershield onto the new follower cluster

On the original leader, export fresh TLS material:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
kubectl --kubeconfig=<leader kubeconfig> get secret -n hubble-timescape sc-server-root-ca-tls \
  -o jsonpath='{.data.ca\.crt}' | base64 -d > sc-server-ca.crt

kubectl --kubeconfig=<leader kubeconfig> get secret -n hubble-timescape hubble-timescape-server-cert \
  -o jsonpath='{.data.ca\.crt}' | base64 -d > leader-ca.crt
```

In Timescape, generate a new one-time join credential (OTP) from the leader UI (Operations → Followers).

On the original leader, configure the newly deployed follower cluster and apply the follower cluster backup file:

```
isoctl hypershield install -f hypershield-install-follower.yaml

sudo KUBECONFIG=<follower kubeconfig> ./hs-restore.sh --file <backup.tar.gz> \
  --mode rollback --appliance-type follower --force
```

See troubleshooting tips.

On the follower, reconnect sc-client to the leader:

```
isoctl hypershield install -f hypershield-install-follower.yaml
```

On the follower, force the follower-join job to re-run by deleting and recreating it from the Helm release hooks (this command may need to be re-run):

```
kubectl delete job -n hubble-timescape hubble-timescape-follower-join
```

Sample successful output:

```
Condition transition: SecureConnectorReady=True reason=SecureConnectorReady
Hypershield installed successfully
```

2.2 Re-register all Smart Switches

On the follower, generate a Kubernetes API token:

```
./get-token.sh hypershield smartswitch-token --ca-source secret
```

On each Smart Switch console, register the switch with the token generated in the previous command:

```
switch# service system hypershield register <token>
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Review Smart Switch commands for the next commands.

Wait 15 minutes for internal processes to settle.

2.3 Validate the results in Timescape

In the Timescape web UI, verify that the restored follower cluster is present and behaving normally. Under Enforcement Points, the switches should show **Connected** under the new follower cluster and **Disconnected** under the old follower cluster.

3. Leader and follower restore

This section shows you how to restore both a lost leader cluster and a lost follower cluster from previously made backups onto freshly-deployed replacement clusters that uses the original clusters' IP addressing and DNS records.

Prerequisites

The following prerequisites are needed to complete the steps:

- a leader cluster backup file is available
- a follower cluster backup file is available
- a new Kubernetes cluster is deployed with the original leader cluster's IP address and DNS hostname
- a new Kubernetes cluster is deployed with the original follower cluster's IP address and DNS hostname
- physical access to Smart Switches previously registered to the (now-lost) follower cluster

Leader restore

Follow these previously documented steps:

- 1.1 Deploy Hypershield onto the new leader cluster
- 1.2 Apply the backup file
- 1.3 Verify the environment

Do not perform any steps related to the follower cluster.

Ensure the leader cluster is healthy before continuing.

Follower restore

Follow these previously documented steps:

© 2026 Cisco and/or its affiliates. All rights reserved.

- 2.1 Deploy Hypershield onto the new follower cluster
- 2.2 Re-register all Smart Switches
- 1.5 Validate the results in Timescape

In the above steps, any mention of “original leader cluster” should be understood as being the newly restored leader cluster.

© 2026 Cisco and/or its affiliates. All rights reserved.

Restore Hypershield to new IP addressing

This page shows you how to restore the Hypershield management and control planes where the final underlying infrastructure has new IP addresses and DNS names compared to the original infrastructure.

Warning

Restore is a destructive operation and must be performed with due caution.

These instructions apply to a topology where the management plane is backed by a single `leader` Kubernetes cluster and the control plane is backed by one or more `follower` Kubernetes clusters. See Leader and follower clusters for deployment concepts.

There are three scenarios covered in this document:

4. leader-only restore
5. follower-only restore
6. leader and follower restore

The numbering is a continuation of the analogous scenarios (1, 2, and 3) covered in the companion document: Restore Hypershield to original IP addressing.

Note

You will need an understanding of Leader and Follower cluster YAML configuration. See Install Hypershield on Leader and Follower for details.

The restore script used in the procedures is called `hs-restore.sh` (it resides on the management VM by default). Use the `--dry-run` option for testing. See the dropdown for usage details.

`hs-restore.sh`

Leader-only restore

This section shows you how to restore a lost leader cluster from a previously made backup onto a freshly-deployed replacement cluster that uses new IP addressing and DNS records as compared to the original cluster.

Prerequisites

The following prerequisites are needed to complete the steps:

© 2026 Cisco and/or its affiliates. All rights reserved.

- a leader cluster backup file is available
- the follower cluster is running and pointing at the (now-lost) leader cluster
- a new base Kubernetes cluster is deployed with a new IP address and DNS hostname
- DNS A records exist for the new leader/UI/API hostnames
- a TLS certificate is valid for the new UI hostname (wildcard or SAN match)
- an OIDC provider app is updated such that Sign-In Redirect URLs are added for the new hostnames on the same client ID(s) used by the original leader

4.1 Deploy Hypershield onto the new leader cluster

On the management VM, run the following command to deploy a new leader cluster:

```
isoctl hypershield install -f hypershield-install-leader.yaml
```

Sample successful output:

```
Condition transition: TimescapeReady=True reason=TimescapeReady
Condition transition: SecureConnectorReady=True reason=SecureConnectorReady
Hypershield installed successfully
```

4.2 Apply the backup file

Apply the leader cluster backup file:

```
sudo KUBECONFIG=<new leader kubeconfig> ./hs-restore.sh --file <backup.tar.gz> \
  --mode rollback --appliance-type leader --force
```

Sample successful output:

```
[01:20:25] INFO  Checksums OK.
[01:20:26] INFO  Cluster reachable: mgmtcluster.isovalent.com
[01:20:38] INFO  AUDIT restore complete with k8s apply failures: count=1
```

See troubleshooting tips.

4.3 Validate the results in Timescape

In the Timescape web UI, verify that all restored data is present and correct:

© 2026 Cisco and/or its affiliates. All rights reserved.

- Objects
- Policies
- Event Logs
- Followers
- Telemetry Destinations
- Logging Groups

At this time, there are no registered Smart Switches.

Below is a screenshot of the Objects page:



4.4 Reconnect the follower

For the restored leader, export fresh TLS material:

```
kubectl --kubeconfig=<new leader kubeconfig> get secret \
  -n hubble-timescape sc-server-root-ca-tls \
  -o jsonpath='{.data.ca\.crt}' | base64 -d > sc-server-ca.crt

kubectl --kubeconfig=<new leader kubeconfig> get secret \
  -n hubble-timescape hubble-timescape-server-cert \
  -o jsonpath='{.data.ca\.crt}' | base64 -d > leader-ca.crt
```

Transfer the TLS material to the follower cluster host:

```
scp sc-server-ca.crt leader-ca.crt iep@<follower host>:follower-creds/<name>/
```

In Timescape, generate a new join credential (Operations → Followers).

© 2026 Cisco and/or its affiliates. All rights reserved.

Edit the follower cluster's configuration file `hypershield-install-follower.yaml` by updating `leaderAddress` to the new leader cluster's hostname, and `secureConnector.serverAddress` to the new leader cluster's SecureConnector IP address.

On the follower, force the follower-join job to re-run by deleting and recreating it from the Helm release hooks (this pair of commands may need to be re-run):

```
isoctl hypershield install -f hypershield-install-follower.yaml
kubectl delete job -n hubble-timescape hubble-timescape-follower-join
```

Sample successful output:

```
Condition transition: SecureConnectorReady=True reason=SecureConnectorReady
Hypershield installed successfully
```

Still on the follower, delete the sc-client-bootstrap secret and restart sc-client:

```
kubectl delete secret -n hubble-timescape sc-client-bootstrap
kubectl rollout restart deployment -n hubble-timescape sc-client
```

Sample successful output:

```
level=INFO msg="tunnel active, serving SOCKS5 connections" subsys=connector leader=198.18.42.129:443
```

4.5 Validate the results in Timescape

In the Timescape web UI, verify that all restored data is present and correct:

- Objects
- Policies
- Event Logs
- Followers
- Telemetry Destinations
- Logging Groups

The follower cluster should show as **Active** and any Smart Switches should show as **Connected**.

Below is a screenshot of the Objects page:

© 2026 Cisco and/or its affiliates. All rights reserved.



5. Follower-only restore

This section shows you how to restore a lost follower cluster from a previously made backup onto a freshly-deployed replacement cluster that uses new IP addressing and DNS records as compared to the original cluster.

Prerequisites

The following prerequisites are needed to complete the steps:

- a follower cluster backup file is available
- the leader cluster is running and pointing at the (now-lost) follower cluster
- a new base Kubernetes cluster is deployed with a new IP address and DNS hostname
- new local IP addresses are reserved for the follower's ingester and CLC endpoints
- physical access to Smart Switches previously registered to the (now-lost) follower cluster

5.1 Deploy Hypershield onto the new follower cluster

For the original leader, export fresh TLS material:

```
kubectl --kubeconfig=<leader kubeconfig> get secret -n hubble-timescape sc-server-root-ca-tls \
  -o jsonpath='{.data.ca\.crt}' | base64 -d > sc-server-ca.crt

kubectl --kubeconfig=<leader kubeconfig> get secret -n hubble-timescape hubble-timescape-server-cert \
  -o jsonpath='{.data.ca\.crt}' | base64 -d > leader-ca.crt
```

In Timescape, generate a new one-time join credential (OTP) from the leader UI (Operations → Followers).

© 2026 Cisco and/or its affiliates. All rights reserved.

Edit the follower cluster's configuration file `hypershield-install-follower.yaml` by updating `clusterName`, `ingester.ip` and `clc.ip` to the new follower cluster's ingester and CLC endpoint IP addresses.

On the original leader, configure the newly deployed follower cluster:

```
isoctl hypershield install -f hypershield-install-follower.yaml
```

Confirm that the new follower cluster has joined the leader:

```
kubectl get pods -n hubble-timescape
kubectl get pods -n hubble-timescape -l app.kubernetes.io/name=sc-client
```

Apply the follower cluster backup file:

```
sudo KUBECONFIG=<follower kubeconfig> ./hs-restore.sh --file <backup.tar.gz> \
  --mode rollback --appliance-type follower --force
```

See troubleshooting tips.

Replace the smartswitch-timescape-config ConfigMap's ingester IP address (namespace `hypershield`) with the new follower cluster's ingester IP address:

```
kubectl get cm -n hypershield smartswitch-timescape-config -o jsonpath='{.data.timescape_config}'
```

5.2 Validate the follower cluster in Timescape

In the Timescape web UI, verify that the leader's Followers page displays the restored follower cluster registers under its own name with no collision with any other still-connected follower cluster.

5.3 Re-register all Smart Switches

On the new follower, generate a Kubernetes API token:

```
./get-token.sh hypershield smartswitch-token --ca-source secret
```

On each Smart Switch console, register the switch with the token generated in the previous command:

```
switch# service system hypershield register <token>
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Review Smart Switch commands for the next commands.

Wait 15 minutes for internal processes to settle.

5.4 Validate the results in Timescape

In the Timescape web UI, verify that the restored follower cluster is present and behaving normally. Under Enforcement Points, the switches should show **Connected** under the new follower cluster and **Disconnected** under the old follower cluster.

6. Leader and follower restore

This section shows you how to restore both a lost leader cluster and a lost follower cluster from previously made backups onto freshly-deployed replacement clusters that uses new IP addressing and DNS records as compared to the original clusters.

Prerequisites

The following prerequisites are needed to complete the steps:

- a leader cluster backup file is available
- a follower cluster backup file is available
- a new Kubernetes cluster is deployed with the new leader cluster's IP address and DNS hostname
- a new Kubernetes cluster is deployed with the new follower cluster's IP address and DNS hostname
- physical access to Smart Switches previously registered to the (now-lost) follower cluster

Leader restore

Follow these previously documented steps:

- 4.1 Deploy Hypershield onto the new leader cluster
- 4.2 Apply the backup file
- 4.3 Validate the results in Timescape

Do not perform any steps related to the follower cluster.

Ensure the leader cluster is healthy before continuing.

© 2026 Cisco and/or its affiliates. All rights reserved.

Follower restore

Follow these previously documented steps:

- 5.1 Deploy Hypershield onto the new follower cluster

  Add new data in the follower cluster's configuration file. Examples: * `leaderAddress: "new-leader.pseudo.co:443"` * `secureConnector.serverAddress: "198.18.42.129:443"`

Change the ingester IP from the old follower cluster's IP address to the new follower cluster's ingester IP:

```
kubectl get cm -n hypershield smartswitch-timescape-config -o jsonpath='{.data.timescape_config}' >
/tmp/ts-config.json

sed -i 's/198\.18\.42\.168/198.18.42.108/' /tmp/ts-config.json   # old follower IP -> new follower IP

kubectl patch cm -n hypershield smartswitch-timescape-config --type=merge \
  -p "{\"data\":{\"timescape_config\": $(python3 -c 'import json; print(json.dumps(open("/tmp/ts-
config.json").read()))')}}"
```

- 5.2 Validate the follower cluster in Timescape
- 5.3 Re-register all Smart Switches
- 4.5 Validate the results in Timescape

In the above steps, any mention of "original leader cluster" should be understood as being the newly restored leader cluster.

© 2026 Cisco and/or its affiliates. All rights reserved.

Backup and restore troubleshooting

This page covers troubleshooting tips and methods when backing up and restoring the Hypershield management and control planes. Each section (and sub-section) map to an identical section/sub-section in a corresponding procedural page.

Restore Hypershield to original IP addressing

The corresponding procedural page for this section is here.

1.1 Deploy Hypershield onto the new leader cluster

After this command, a kubectl patch was noted to be needed on rare occasions.

```
isoctl hypershield install -f hypershield-install.yaml -timeout=30m
```

Apply this patch:

```
kubectl patch tlsroute -n hubble-timescape hubble-timescape-server \
  --type=json -p='[{"op":"add","path":"/spec/parentRefs/0/sectionName","value":"grpc"}]'
```

1.2 Apply the backup file

After this command, see the listed example errors and mitigation steps. Also ensure that logs are checked for errors.

```
sudo KUBECONFIG=<leader kubeconfig> ./hs-restore.sh --file <backup.tar.gz> \
  --mode rollback --appliance-type leader --force
```

Example errors:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
[01:20:31] INFO  Restoring critical secrets...
Error from server (Invalid): Secret "hubble-timescape-credential-encryption-keys" is invalid: data:
Forbidden: field is immutable when `immutable` is set
Error from server (Invalid): Secret "hubble-timescape-signing-keys" is invalid: data: Forbidden: field
is immutable when `immutable` is set
Error from server (Invalid): Secret "sc-client-kek" is invalid: data: Forbidden: field is immutable
when `immutable` is set
Error from server (Invalid): Secret "sc-server-kek" is invalid: data: Forbidden: field is immutable
when `immutable` is set
[01:20:33] WARN  Some critical secrets in hubble-timescape may not have applied cleanly.
...
K8s apply failures: 1
```

Mitigation steps - Delete the affected secrets and re-run the original command:

```
kubectl --kubeconfig=<kubeconfig> delete secret -n hubble-timescape \
  hubble-timescape-credential-encryption-keys \
  hubble-timescape-signing-keys \
  sc-client-kek \
  sc-server-kek

sudo KUBECONFIG=<kubeconfig> ./hs-restore.sh --file <backup.tar.gz> \
  --mode rollback --appliance-type leader --force
```

1.2a Check the logs for errors

In the logs, check for errors relating to:

- ClickHouse password-hash
- public-api client-tls
- public-api RBAC-path breakage

Do this for every leader restore, not just after isoctl re-runs.

Example errors:

```
level=WARN msg="unable to connect to DB" error="unable to connect to database: code: 516, message:
timescape_migrate: Authentication failed: password is incorrect, or there is no user with such name"
attempts=2 sleep-duration=4s
level=WARN msg="unable to connect to DB" ... attempts=8 sleep-duration=10s
context deadline exceeded
```

Mitigation steps - Sync ClickHouse password hashes:

```
hubble-timescape-*-creds Secrets, patch chi-hubble-timescape-common-usersd,
```

Then restart `server/analyzer/ingester/intent-syncer`.

© 2026 Cisco and/or its affiliates. All rights reserved.

Example errors:

```
level=WARN msg="Initial TLS configuration load failed, certificate files might not yet be available,
waiting on fswatcher update to become ready" config=tls-client reloadKeypairError="failed to load
keypair: open /var/lib/hubble-timescape-public-api/client-tls/tls.crt: no such file or directory"
reloadCAError=<nil>
error creating Timescape upstream clients: getting gRPC client options: cannot get TLS configuration:
context canceled
```

Mitigation steps - Restore public-api client-tls volume mount:

```
kubectl patch deployment -n hubble-timescape hubble-timescape-public-api --type=json -p='[
  {"op":"add","path":"/spec/template/spec/volumes/-",
   "value":{"name":"public-api-client-tls","projected":{"defaultMode":256,
     "sources":[{"secret":{"name":"hubble-timescape-public-api-cert"}}]}}},
  {"op":"add","path":"/spec/template/spec/containers/0/volumeMounts/-",
   "value":{"name":"public-api-client-tls",
     "mountPath":"/var/lib/hubble-timescape-public-api/client-tls","readOnly":true}}
]'
```

Example errors:

```
cannot create public API RBAC auth: failed to create public API policy authorizer: policy loader has no
current policy in place: multiLoader: error getting current policy from loader "config": open
/etc/hubble-timescape/rbac-policy.yaml: no such file or directory
```

Mitigation steps - Patch public-api RBAC policy-file path, then restart:

```
kubectl patch cm -n hubble-timescape hubble-timescape-public-api-config --type=merge \
  -p '{"data":{"config.yaml":"...policy-file: /etc/timescape-public-api-rbac/policy/hubble-timescape-
public-api-rbac-policy.yaml..."}}'
kubectl rollout restart deployment -n hubble-timescape hubble-timescape-public-api
```

Example problem and solution:

If the sc-server records a `missing-table` error, run the sc-server DB migration job:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
kubectl get deployment -n hubble-timescape sc-server -o json | python3 -c "
import sys, json
d = json.load(sys.stdin)
spec = d['spec']['template']['spec']; c = spec['containers'][0]
job = {'apiVersion': 'batch/v1', 'kind': 'Job',
  'metadata': {'name': 'sc-server-migratepg-manual', 'namespace': 'hubble-timescape'},
  'spec': {'template': {'spec': {'restartPolicy': 'Never',
    'containers': [{'name': 'migratepg', 'image': c['image'],
      'command': ['/usr/local/bin/sc-server'], 'args': ['migratepg'],
      'env': c.get('env', []), 'volumeMounts': c.get('volumeMounts', [])}],
    'volumes': spec.get('volumes', []), 'serviceAccountName': spec.get('serviceAccountName', '')}}}
print(json.dumps(job, indent=2))" | kubectl apply -f -

kubectl wait job/sc-server-migratepg-manual -n hubble-timescape --for=condition=complete --timeout=60s

kubectl rollout restart deployment -n hubble-timescape sc-server
```

2.1 Deploy Hypershield onto the new follower cluster

After this command, see the listed example errors and mitigation steps.

```
sudo KUBECONFIG=<follower kubeconfig> ./hs-restore.sh --file <backup.tar.gz> \
  --mode rollback --appliance-type follower --force
```

Example errors:

```
The Secret "sc-client-kek" is invalid: data: Forbidden: field is immutable when `immutable` is set
```

Mitigation steps - Delete the affected secrets and re-run the original command:

```
kubectl --kubeconfig=<follower kubeconfig> delete secret -n hubble-timescape sc-client-kek

sudo KUBECONFIG=<follower kubeconfig> ./hs-restore.sh --file <backup.tar.gz> \
  --mode rollback --appliance-type follower --force
```

Restore Hypershield to new IP addressing

The corresponding procedural page for this section is here.

4.2 Apply the backup file

After this command, see the listed example errors and mitigation steps. Also ensure that logs are checked for errors.

© 2026 Cisco and/or its affiliates. All rights reserved.

```
sudo KUBECONFIG=<leader kubeconfig> ./hs-restore.sh --file <backup.tar.gz> \
  --mode rollback --appliance-type leader --force
```

Example errors:

```
Error from server (Invalid): Secret "hubble-timescape-credential-encryption-keys" is invalid: data:
Forbidden: field is immutable when `immutable` is set
Error from server (Invalid): Secret "hubble-timescape-signing-keys" is invalid: data: Forbidden: field
is immutable when `immutable` is set
Error from server (Invalid): Secret "sc-client-kek" is invalid: data: Forbidden: field is immutable
when `immutable` is set
Error from server (Invalid): Secret "sc-server-kek" is invalid: data: Forbidden: field is immutable
when `immutable` is set
```

Mitigation steps - Delete the affected secrets and re-run the original command:

```
kubectl --kubeconfig=<kubeconfig> delete secret -n hubble-timescape \
  hubble-timescape-credential-encryption-keys \
  hubble-timescape-signing-keys \
  sc-client-kek \
  sc-server-kek
```

4.2a Check the logs for errors

In the logs, check for errors relating to:

- ClickHouse password-hash
- public-api client-tls
- public-api RBAC-path breakage

Example errors:

```
timescape_migrate: Authentication failed: password is incorrect, or there is no user with such name
open /var/lib/hubble-timescape-public-api/client-tls/tls.crt: no such file or directory
policy loader has no current policy in place ... open /etc/hubble-timescape/rbac-policy.yaml: no such
file or directory
```

Mitigation steps - Patch hubble-timescape-ui-config's oauth2.oidc.redirect-url from the old hostname to the new hostname and restart hubble-timescape-ui:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
CURRENT=$(kubectl get cm -n hubble-timescape hubble-timescape-ui-config -o
jsonpath='{.data.config\.yaml}')

FIXED=$(echo "$CURRENT" | sed 's|redirect-url: https://hs10-leader-
ui.pseudo.co/oauth2/callback|redirect-url: https://test-leader-ui.pseudo.co/oauth2/callback|')

kubectl patch cm -n hubble-timescape hubble-timescape-ui-config --type=merge \
  -p "{\"data\":{\"config.yaml\": $(echo \"$FIXED\" | python3 -c 'import sys,json;
print(json.dumps(sys.stdin.read()))')}}"

kubectl rollout restart deployment -n hubble-timescape hubble-timescape-ui
```

5.1 Deploy Hypershield onto the new follower cluster

After this command, see the listed example errors and mitigation steps.

```
sudo KUBECONFIG=<follower kubeconfig> ./hs-restore.sh --file <backup.tar.gz> \
  --mode rollback --appliance-type follower --force
```

Example errors:

```
The Secret "sc-client-kek" is invalid: data: Forbidden: field is immutable when `immutable` is set
```

Mitigation steps - Delete the affected secrets and re-run the original command:

```
kubectl --kubeconfig=<follower kubeconfig> delete secret -n hubble-timescape sc-client-kek

sudo KUBECONFIG=<follower kubeconfig> ./hs-restore.sh --file <backup.tar.gz> \
  --mode rollback --appliance-type follower --force
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Troubleshooting

This section collects resources that can help in troubleshooting various aspects of the Hypershield platform.

© 2026 Cisco and/or its affiliates. All rights reserved.

Hypershield general troubleshooting

This section gathers together general troubleshooting commands and techniques pertinent to the Hypershield platform.

> **Note**
>
> See CLI command usage for guidance on how and where to invoke certain types of CLI commands.

General health check

Invoke these commands on the management VM:

```
kubectl get nodes                          # all Nodes 'Ready'
kubectl get pods -n hypershield            # all Hypershield Pods 'Running', no restarts accumulating
kubectl get pods -n hubble-timescape       # all Timescape Pods 'Running'
kubectl get smartswitches -n hypershield   # all switches 'Connected'
etcdctl endpoint health --cluster          # all etcd members healthy
```

Timescape UI is not reachable

1. Confirm that the Timescape Pod is running:

   ```
   kubectl get pods -n hubble-timescape
   ```

2. Confirm that the Load Balancer IP address is assigned:

   ```
   kubectl get svc -n hubble-timescape
   ```

3. Test connectivity to the Load Balancer IP address:

   ```
   curl -sk https://<timescape-lb-ip>/healthz
   ```

4. If the Timescape login page loops back to Keycloak without granting access, verify the Keycloak realm configuration:
   - realm name must be exactly: `hypershield`
   - client ID must be exactly: `timescape`
   - redirect URI must match your Timescape FQDN

© 2026 Cisco and/or its affiliates. All rights reserved.

API returns `401` or `403`

`401` - the token is missing or was issued with the wrong audience

- re-acquire token with: `client_id=timescape`
- decode your JWT at jwt.io and confirm: aud contains “timescape”

`403` - the assigned Keycloak group provides insufficient privileges

- check Keycloak group membership for the account
- group `policy_manager` or `admin` is required for CRUD operations
- a read-only role gets 403 on any mutation

© 2026 Cisco and/or its affiliates. All rights reserved.

Security policies troubleshooting

This section gathers together troubleshooting commands and techniques pertinent to security policies.

> **Note**
>
> See CLI command usage for guidance on how and where to invoke certain types of CLI commands.

Compare policy intent to DPU state

There is an important distinction between:

- policy intent - what you have declared in `SmartSwitchNetworkPolicy` objects
- DPU state - what is actually programmed into the enforcement engine

Ideally, these are identical. However, when troubleshooting unexpected permit or deny behavior, it is worth verifying:

To view policy intent, run these commands on the management VM:

```
kubectl get smartswitchnetworkpolicies -n hypershield
kubectl describe smartswitchnetworkpolicy <policy-name> -n hypershield
```

If the two views diverge, a policy has not been pushed to the DPU. Check the Timescape UI on the Policy Status Update provided by SmartSwitch

```
kubectl logs -n hypershield deployment/hypershield-controller --tail=100 | grep <switch-name>
```

Policy not applied

Possible symptoms: `SmartSwitchNetworkPolicy` shows an error status

Run the below commands.

On the management VM:

```
kubectl describe smartswitchnetworkpolicy -n hypershield <policy-name>
```

On the switch console:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
switch# show tech-support service-acceleration
```

Common causes:

Error	Cause	Solution
`VRF not found`	VRF name not in mapping	add active VRF to SmartSwitchNetworkPolicy
`Validation failed`	invalid CIDR or port	check YAML syntax
`Capacity exceeded`	too many rules	optimize or split policy

Traffic is blocked that should be allowed

1. Confirm the policy exists on the controller. On the management VM:

   ```
   kubectl get smartswitchnetworkpolicies -n hypershield
   ```

2. Confirm the rule reached the DPU. Within the switch:

   ```
   switch# show tech-support service-acceleration
   ```

   Check the ‘policies info’ for the policy name in the above collected output.

   ```
   switch# show tech-support dpu module all
   ```

   Check the ‘hs policies’ for the policy and rules in the above collected output.

   If the rule is absent, check switch connectivity:

   ```
   switch# show service-acceleration status details
   Controller Connection Status: success
   ```

3. Confirm the traffic matches the policy rules exactly:
   - the source IP address must fall within the policy’s source CIDR
   - the destination IP address must fall within the policy’s destination CIDR
   - the protocol and port must match exactly (no wildcards)
   - VRF of the traffic must match the VRF steered to the DPU
4. Confirm the correct VRFs are steered to the DPU:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
switch# show service-acceleration status details
```

VRF count should include the VRF the traffic traverses

VRF resolution errors

Possible symptoms: policy shows the `VRF not found` error

Run the below commands on the switch console.

Check which VRFs are available on the AGW:

```
switch# show tech-support service-acceleration
```

Check the ‘vrf show’ for the VRF name in the above collected output.

Check VRFs on a switch console:

```
switch# show vrf
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Smart Switch troubleshooting

This section gathers together troubleshooting commands and techniques pertinent to Smart Switches.

> **Note**
>
> See CLI command usage for guidance on how and where to invoke certain types of CLI commands.

Switch shows status `Pending` or `Disconnected`

1. If registration of the switch was very recent, wait at least 60 seconds - the initial TLS handshake can take some time.
2. Check network reachability from the controller to the switch's source-interface address:

```
kubectl exec -n hypershield <controller-pod> -- \
   curl -sk https://<switch-source-interface-ip>/
```

3. Confirm the source-interface address is configured on the switch:

```
switch# show interface loopback <N>
```

4. If the switch was previously `Connected` and is now `Disconnected`, check for interface events on the switch:

```
switch# show logging last 100 | include eth1/48
```

General health check

Invoke the below commands on the management VM.

Check the status of all switches:

```
kubectl get smartswitches -n hypershield
```

Check the security policy status:

```
kubectl get smartswitchnetworkpolicy -n hypershield
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Invoke this command from the switch console:

```
switch# show service-acceleration status details
Agent Status: firewall-ready,redirect-installed
Agent Health Status: ok
Controller Connection Status: success
```

Check DPU registration status

Verify that the DPUs are registered with the AGW:

```
switch# show tech-support service-acceleration
```

Check the show_dpu in the collected output.

Sample output:

```
lastPing   healthy   dpu   uid             hardware   agent                          datapath
dpuReboot   policySync
1s         true      1     169.254.151.1   giglio     v1.19.0-pre.8-150-ge2ba52cc   1.150.2    13
true
1s         true      2     169.254.159.1   giglio     v1.19.0-pre.8-150-ge2ba52cc   1.150.2    13
true
```

Key health indicators:

Field	Healthy value	Description	Unhealthy sign
`healthy`	`true`	DPU is operational	`false` — DPU has failed health checks
`lastPing`	1-2s	Recent heartbeat received	>10s — communication issues between DPU and AGW
`policySync`	`true`	Policies are synchronized	`false` — policy distribution failure

DPU health checks

Various checks to perform on a DPU:

```
switch# show tech-support dpu module all
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Switch not discovered

Possible symptom: a SmartSwitch CR remains in a pending state

Run these commands on the management VM.

```
kubectl describe smartswitch -n hypershield <switch-name>
kubectl logs -n hypershield deployments/hubble-cpc | grep <switch-name>
```

AGW not connected to control plane

Possible symptoms:

- a Smart Switch CR is not created or remains in a pending state
- a Smart Switch does not receive updated policies or configurations
- policy changes on the control plane is not reflected on DPUs

An ordered series of diagnostic steps are recommended.

Perform all commands on a switch console.

Step 1: Test controller connectivity

On a switch console:

```
switch# service system hypershield test controller connection
```

Sample successful output:

```
==============================
Running config checks
==============================
Using source-interface loopback0 for controller connection
Controller connection status: [success]
Controller reason: [connected ok with Hypershield controller]
Controller endpoint: 10.62.156.185:6443

================================================
Starting network connectivity checks on the switch
================================================

Not using proxy for controller connection

Using curl to check connection to 10.62.156.185 port 6443
Curl successfully connected to 10.62.156.185:6443.
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Things to check if the above fails:

- verify network routing to control plane
- check firewall rules for port 6443
- verify source interface configuration

Step 2: Check AGW logs

View the AGW logs for authentication or connection errors:

```
switch# show tech-support service-acceleration
```

Check the show_log logs in the above collected output.

DPU issues

Diagnostic commands to perform on a switch console:

```
switch# show dpu internal info
switch# show dpu internal msgs
switch# show dpu internal event-history errors
switch# show dpu internal event-history msgs
```

Capacity issues

Policy capacity exceeded

Possible symptoms: new policies fail with capacity errors

Error codes:

Error code	Description
`ERR_CODE_POLICY_MAX_RULES_EXCEEDED (4096)`	Single policy exceeds 1K/4K limit
`ERR_CODE_POLICY_SRC_IP_MAX_TREE_NODES_EXCEEDED (129024)`	Too many source IP prefixes
`ERR_CODE_POLICY_DST_IP_MAX_CLASSES_EXCEEDED (129029)`	Too many destination IP groups
`ERR_CODE_POLICY_DST_PORT_PROTO_MAX_TREE_NODES_EXCEEDED (129034)`	Too many port combinations

© 2026 Cisco and/or its affiliates. All rights reserved.

Run the below command on the switch console to check the AGW logs:

```
switch# show tech-support service-acceleration
```

Check the show_log logs for “exceeded” in the above collected output.

A possible solution is to optimize the policies:

Technique	Avoid	Prefer
Use larger CIDR prefixes	256 x /32 routes	1 x /24 network
Use port ranges	1001 individual ports	8000-9000 range

Upgrade issues

AGW error: `package already exists`

When installing a new AGW and activating it in the same operation, you may encounter an error that states the package already exists:

```
switch# install add bootflash:agent-10.0.0-10.6.3s.x86_64.rpm activate upgrade
```

Example error output:

```
Adding the patch (/agent-10.0.0-10.6.3s.x86_64.rpm)
[####################] 100%
Install operation 16 failed because this package already exists. at Thu Jun  4 23:04:34 2026
```

Try to re-activate it:

```
switch# install activate agent-10.0.0-10.6.3s.x86_64.rpm
```

Sample successful output:

```
Groups-RPM Database                               58 MB/s |  60 kB     00:00
Local RPM Database                               1.5 MB/s | 1.5 kB     00:00
Local RPM Database                               1.4 MB/s | 1.4 kB     00:00
Thirdparty RPM Database                          1.5 MB/s | 1.5 kB     00:00
Groups-RPM Database                              2.2 MB/s | 2.3 kB     00:00
[####################] 100%
Install operation 17 completed successfully at Thu Jun  4 23:07:43 2026
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Enable Gateway API post bootstrap

During the installation of Hypershield on vSphere, a Gateway API enablement check was performed internally. Use the guidance on this page to enable Gateway API in case that check was not successful.

Patch the `mgmtcluster` to enable Gateway API:

```
kubectl patch mgmtcluster -n mgmt "${CLUSTER_NAME}" -p '{"spec":{"components":{"gatewayAPI":
{"enabled":true}}}}' --type=merge
```

The expected output is:

```
mgmtcluster.platform.isovalent.com/control-plane patched
```

Wait for the change to propagate. This takes approximately five minutes.

Then restart Cilium:

```
kubectl delete pod -l app.kubernetes.io/part-of=cilium -n kube-system
```

Verify Cilium configuration

L2 Announcement

```
kubectl -n kube-system exec ds/cilium -c cilium-agent -- cilium-dbg config --all | grep
EnableL2Announcements
```

The expected output is:

```
EnableL2Announcements             : true
```

If not true, restart Cilium again:

```
kubectl delete pod -l app.kubernetes.io/part-of=cilium -n kube-system
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Gateway class

```
kubectl get gatewayclasses.gateway.networking.k8s.io cilium
```

The expected output is:

```
NAME     CONTROLLER                     ACCEPTED   AGE
cilium   io.cilium/gateway-controller   True       4m
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Collect troubleshooting information for Hypershield

When you encounter unexpected behaviour with Hypershield, collecting troubleshooting information and providing it to Cisco Support is an important first step in having your concerns addressed in the most efficient way possible.

This pages shows how to collect three distinct bundles of information:

- one from the environment's control plane
- two from an individual Smart Switch

> **Tip**
>
> You will need to specify the path of the output file for each bundle of information. Use a meaningful filename that includes a timestamp.

Collect control plane data

Access the management VM and run the following:

```
isoctl sysdump \
  --kubeconfig /home/iep/hypershield/control-plane.yaml \
  --hypershield \
  --output-filename isoctl-sysdump-hypershield-control-plane-2026-10-31
```

The above captures the following:

- Pod logs
- CRD state
- CNPG diagnostics
- Hypershield-specific telemetry

The command places all this information into a single file named `isoctl-sysdump-hypershield-control-plane-2026-10-31` .

Collect switch data

Access an affected switch and run the given commands on its console. There are two data bundles to collect from a switch.

First bundle:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
switch# show tech-support service-acceleration > bootflash:show-tech-svc-accel.txt
```

The above captures:

- AGW logs
- DPU logs
- core dumps
- end-to-end diagnostic state

The command places all this information into a single file named `bootflash:show-tech-svc-accel-2026-10-31.txt` .

Second bundle:

```
switch# show tech-support dpu module all > bootflash:show-tech-dpu-all.txt
```

The above captures:

- DPU firmware versions
- hardware health
- memory utilization
- per-module operational state

The command places all this information into a single file named `bootflash:show-tech-dpu-all-2026-10-31.txt` .

Provide the output files

You should now have three files:

- `isoctl-sysdump-hypershield-control-plane-2026-10-31`
- `bootflash:show-tech-svc-accel-2026-10-31.txt`
- `bootflash:show-tech-dpu-all-2026-10-31.txt`

Provide these files to the Cisco TAC team.

© 2026 Cisco and/or its affiliates. All rights reserved.

Hypershield resources

These pages provide the background you need to plan, operate, and reason about Hypershield deployments in production. The material here complements the conceptual, procedural, and reference documentation.

Use these pages to:

- Understand the **caveats and limitations** of the current release, so you can plan deployments around known constraints and their workarounds.
- Anticipate **failure scenarios** and understand how Hypershield behaves when individual components or dependencies fail.
- Consult **scalability testing** results to size deployments against validated limits.

© 2026 Cisco and/or its affiliates. All rights reserved.

Caveats and limitations

This page describes the current caveats and limitations of the Hypershield platform, along with best practices to help guide the Operator.

Description	Component	Impact	Workaround
Policy and network object payload size	Hypershield Controller	A SmartSwitchNetworkPolicy or NetworkObjectGroup serialized above 1.5 MB can fail when the follower Kubernetes API materializes it.	Keep each object below 1.5 MB. Split large policies or object groups into smaller reusable objects.
OIDC refresh-token flow	Hypershield Controller	Users must authenticate again when the ID token expires, even when the identity provider supplies offline access.	Plan for periodic re-authentication and choose an appropriate ID-token lifetime in the identity provider.
Event-history CSV export	Hypershield Controller	A CSV export contains only the events displayed on the current page, not the complete filtered result set.	Export each page and combine the files, or review the complete history in Timescape.
Leader enrollment event visibility	Hypershield Controller	Leader enrollment events are not shown in event history, so enrollment does not create a visible audit entry in the UI.	Record leader enrollment through the organization change-control process.
Telemetry configuration replacement	Hypershield Controller / Hypershield Agent Gateway	A pre-existing statically named syslog or IPFIX ConfigMap can prevent a new UI-managed configuration from reaching previously configured Smart Switches.	Remove the obsolete ConfigMap from the affected follower cluster, then reapply the telemetry configuration and verify delivery.
VM live migration	Hypershield Network Enforcer	Live migration of a protected workload VM resets established TCP sessions that traverse DPU stateful inspection.	Drain connections or schedule migration during a low-traffic maintenance window.
Policy updates during control-plane disconnection	Hypershield Agent Gateway	A disconnected Smart Switch continues enforcing its last installed policy in a fail-closed posture, but it cannot receive new policy.	Restore follower-to-switch connectivity before deploying policy changes.
Policy restore after Smart Switch power cycle	Hypershield Network Enforcer / Agent Gateway	A power cycle clears in-memory DPU policy. Traffic is fail-closed until the switch reconnects and the control plane restores the current policy.	Use a correctly configured HA partner to maintain enforcement while the rebooted switch recovers.
Broadcast, unknown-unicast, and multicast traffic	Hypershield Network Enforcer	BUM traffic bypasses the DPU and is not evaluated by Hypershield policy.	Use NX-OS controls such as storm control or IGMP snooping; do not rely on Hypershield policy for BUM filtering.
Bidirectional policy on L2-only switches	Hypershield Network Enforcer	On a switch without routed interfaces, policy can be unidirectional if both source and destination VLANs are not steered to the DPU.	Configure both VLAN IDs in the service firewall block and validate traffic in both directions.

© 2026 Cisco and/or its affiliates. All rights reserved.

Description	Component	Impact	Workaround
Inter-VRF mode counters	Hypershield Network Enforcer	Inter-VRF mode is enabled by default. The first packet makes two NPU-to-DPU passes, which can double-count dropped packets on the service Ethernet interface.	Account for the double count in monitoring. Disable inter-VRF mode only for an intra-VRF-only design and only through an approved procedure.
Very large policy deployment	Hypershield Network Enforcer / Agent Gateway	Applying more than 500,000 rules can temporarily reduce throughput while the receiving Smart Switch programs the policy.	Schedule very large deployments during lower traffic and monitor switch health and policy status until synchronization completes.
Policy references a missing VRF	Hypershield Controller	The policy remains Pending and rules that depend on the missing VRF are not enforced on that Smart Switch.	Configure the referenced VRF. If its absence is intentional, treat Pending as expected for that target.
Policy references a missing VLAN	Hypershield Controller	The policy remains Pending and rules that depend on the missing VLAN are not enforced on that Smart Switch.	Configure the referenced VLAN. If its absence is intentional, treat Pending as expected for that target.
Mixed IPv4/IPv6 endpoint rule	Hypershield Controller	A policy that pairs principal and resource from different IP families remains Pending and is not enforced.	Use IPv4-to-IPv4 or IPv6-to-IPv6 source and destination pairs.
Policies reference mismatched VLAN IDs in Network Object Groups	Hypershield Controller	Policies using VLAN-based objects are only enforced when principal and resource VLANs match. Use VRF-based objects for inter-VLAN traffic.	For inter-VLAN enforcement, create policies using network object groups that leverage VRF, not VLAN, IDs.

© 2026 Cisco and/or its affiliates. All rights reserved.

Failure scenarios

This page lists failure scenarios for the Hypershield platform. Failure scenarios are known situations that negatively impact a deployment. In particular, how using a feature in a specific manner can cause problems. Each entry includes preventative strategies and/or mitigation steps.

#	Description	Impact	Workaround
1	Policy changes cannot be pushed to a switch that is disconnected from the controller. The switch enforces its existing policy set in a fail-closed posture during the outage, but no new policies can be deployed until connectivity is restored.	Policy operations are suspended for the disconnected switch only.	Restore controller-to-switch connectivity before attempting policy updates. Do not plan policy rollouts during maintenance windows when the switch will be disconnected.

© 2026 Cisco and/or its affiliates. All rights reserved.

Scalability testing

This page lists the results of scalability testing performed for the Hypershield platform.

Network Enforcer (per switch)

Metric	Published target	Notes
Maximum throughput	95% of available PCIe bandwidth	two DPUs per switch; target applies per DPU
Maximum new connections per second	1 million	CA2 published limit
Maximum policy rules at switch	1 million	CA2 published limit
Maximum concurrent connections	to be confirmed	contact Cisco for current engineering guidance
Latency added by DPU inspection	to be confirmed	contact Cisco for current engineering guidance

Controller scale

Metric	Limit
Controllers per tenant	1
Maximum policies per policy group	300 policies / 260,000 policy rules
Total policy rules (SCC MP + Controller)	8.7 million (based on 100 policies per group, 130,000 rules each)
Number of ToR switches per fabric	to be confirmed by engineering
Number of VRFs supported	to be confirmed by engineering
Number of Network Object Groups supported	to be confirmed by engineering

© 2026 Cisco and/or its affiliates. All rights reserved.

Hypershield reference

This section provides reference information for the Hypershield platform.

© 2026 Cisco and/or its affiliates. All rights reserved.

CLI command usage

This page helps orientate you in terms of how and where some CLI commands are invoked when working with Hypershield.

Command prompts

The command prompt (or a lack of one) provided in commands indicates in what context the command should be run:

Prompt	Context
`bash-5.2#`	the NX-OS Bash shell on a switch
`switch#`	the switch console

The lack of a specialized prompt represents the management VM.

Controller commands

Controller commands, typically the `kubectl` command but also general Linux commands, are invoked on the management VM (or any host that has access to the Kubernetes control plane).

Smart Switch commands

To query for data on a switch, you can interact with the NX-OS Bash shell.

The Bash shell is called from the switch console:

```
switch# run bash sudo su
bash-5.2# <command>
bash-5.2# uname -a
bash-5.2# exit
switch#
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Timescape API usage

Timescape provides a versioned public REST API for supported network object and security policy operations. Public API paths use `/api/hs/v1` on your deployment's Timescape URL.

> **Note**
>
> Documentation pages give preference to the web UI, but where an API alternative exists, it accompanies the web UI instructions in a collapsible **Use the API** section.

Timescape public API reference

For the complete contract, see the Timescape public API reference. The reference also describes authentication and authorization, requests and responses, and errors and status codes.

Prerequisites

Before you use the examples, ask your Timescape administrator for the HTTPS URL and any deployment-specific credentials. You might also need:

- a certificate authority (CA) certificate if your system does not trust the deployment's CA
- a client certificate and private key if the deployment requires mutual TLS (mTLS)
- a JSON Web Token (JWT) bearer token if public API role-based access control (RBAC) is enabled

The administrator must expose the API and configure TLS before clients can connect. Configure the deployment URL without a path. The examples use Bash arrays:

```
export TIMESCAPE_URL="https://timescape.example.com"
export TIMESCAPE_API="${TIMESCAPE_URL%/}/api/hs/v1"
TIMESCAPE_CURL_ARGS=(--fail-with-body --silent --show-error)
```

For a private CA, add its certificate to your system trust store or pass it to `curl`:

```
TIMESCAPE_CURL_ARGS+=(--cacert "/path/to/timescape-ca.pem")
```

If the deployment requires mTLS, add the client certificate and private key:

© 2026 Cisco and/or its affiliates. All rights reserved.

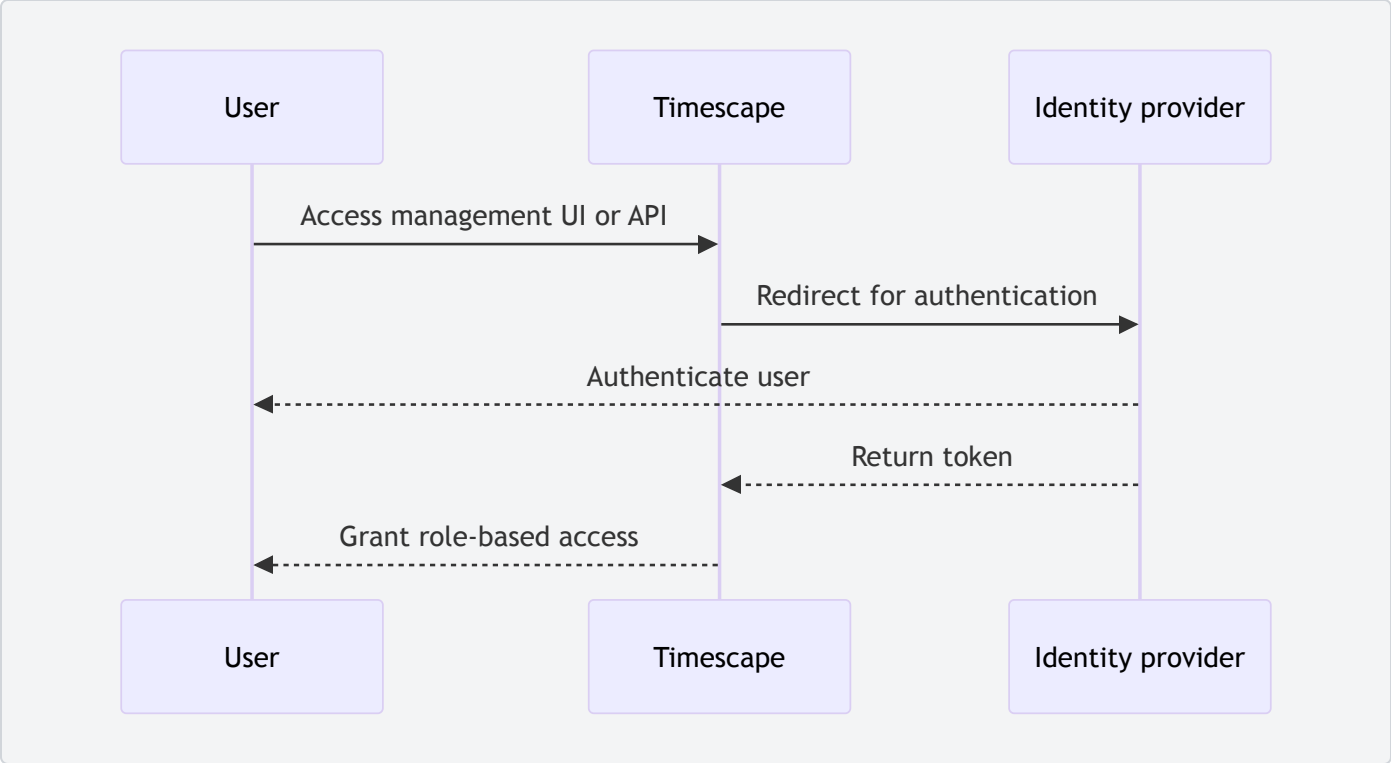
```
TIMESCAPE_CURL_ARGS+=(
  --cert "/path/to/timescape-client.pem"
  --key "/path/to/timescape-client-key.pem"
)
```

Store the client certificate and private key in protected files. Do not use `--insecure` to bypass certificate validation.

Public API RBAC

A client certificate does not replace a bearer token when public API RBAC is enabled.

When public API RBAC is enabled, API clients send an identity-provider-issued JSON Web Token (JWT) that the deployment accepts as a bearer token. For details, see Authentication and authorization in the API reference.



Read the token without adding it to your shell history, then add the authorization header:

```
read -r -s -p "Timescape API token: " TIMESCAPE_TOKEN
echo
TIMESCAPE_CURL_ARGS+=(
  --header "Authorization: Bearer ${TIMESCAPE_TOKEN}"
)
```

When public API RBAC is disabled, Timescape does not require its bearer token. A gateway or ingress can still require other credentials. Confirm the access requirements with your administrator, then verify access by requesting the Timescape build information:

© 2026 Cisco and/or its affiliates. All rights reserved.

```
curl "${TIMESCAPE_CURL_ARGS[@]}" \
  --header "Accept: application/json" \
  "${TIMESCAPE_API}/meta/build-info"
```

Clean the shell environment

When you're finished working with the API, remove the token and request arguments from your shell:

```
unset TIMESCAPE_TOKEN TIMESCAPE_CURL_ARGS
```

© 2026 Cisco and/or its affiliates. All rights reserved.

Glossary

Terms used throughout this guide, listed alphabetically. Where a term has a precise meaning specific to the Hypershield platform - distinct from its general networking meaning - that distinction is noted.

active/active

A high-availability model in which two Smart Switches operate simultaneously, each processing live traffic. Neither switch is idle or in standby. Contrast with active/standby, which Hypershield does not use. See also: HA peer link.

AGW

Agent Gateway. The software component that runs on a Nexus Smart Switch and serves as the bridge between the Hypershield control plane and the AMD DPU. The AGW receives compiled policy from the control plane over gRPC, translates it into DPU forwarding rules, and programs the DPU datapath. The AGW also collects DPU status, health metrics, and log export state and reports them back to the control plane.

CIDR

Classless Inter-Domain Routing. An IP address range expressed in prefix notation (for example: 10.10.20.0/24). CIDRs are the primary way to identify source and destination endpoints in `SmartSwitchNetworkPolicy` rules and `NetworkObjectGroup` definitions. Also referred to as ipBlock in the policy schema.

ClickHouse

The time-series database that backs Timescape. ClickHouse stores event log data, policy history, and observability records. It is deployed as part of the Timescape installation and does not require separate administration under normal operations.

ConfigMap

A Kubernetes object used to deliver configuration data to workloads. In Hypershield, two `ConfigMap` objects are used to configure DPU logging behavior: smartswitch-log-config (syslog export settings) and smartswitch-flow-export-config (IPFIX export settings). The Agent Gateway reads both objects and programs the DPU accordingly.

Connected

The expected steady-state status of a registered Smart Switch as seen from the control plane. A switch in Connected state has an active gRPC channel to the control plane, has received its policy set, and is enforcing. Pending indicates the registration handshake has not completed. Disconnected indicates the gRPC channel has been severed.

control plane link

The IP path between the Hypershield control plane and a Smart Switch's management interface (the source-interface loopback). The control plane uses this path to push policy to the switch. If this link goes down, the switch enters policy hold-down mode. Not to be confused with the HA peer link, which connects the two switches in an HA pair.

© 2026 Cisco and/or its affiliates. All rights reserved.

CRD

Custom Resource Definition. A Kubernetes extension mechanism that defines new object types. Hypershield uses three CRDs: `SmartSwitch`, `SmartSwitchNetworkPolicy`, and `NetworkObjectGroup`. These are installed on a control plane cluster and are the primary objects an admin interacts with via kubectl.

default-deny

The implicit policy installed on the DPU datapath at switch registration. Hypershield is deny-by-default - you build an allow-list on top of this baseline. The default-deny rule cannot be removed; it is the foundation of the fail-closed security posture.

DPU

Data Processing Unit. The DPU performs stateful L3/L4 connection tracking and per-flow policy evaluation at line rate inside the switch, eliminating the need to forward traffic to an external firewall. The DPU is the actual enforcement point - the control plane and AGW exist to program and manage it.

DPU traffic steering

The NX-OS mechanism that redirects traffic from the NPU (the switch's normal forwarding path) to the DPU for inspection. Steering is configured per VRF and per VLAN using the service firewall block. Traffic in VRFs or VLANs that are not listed in service firewall is not steered and bypasses the DPU entirely. Steering is implemented via a module-affinity mode; see dynamic module affinity and pinned module affinity.

dynamic module affinity

This mode has NX-OS assign each VRF to a specific DPU automatically based on the VRF name - the same VRF always lands on the same DPU, ensuring consistent stateful tracking without manual assignment. This mode is the default steering mode and is implemented with the symmetric hashing algorithm. Contrast with `pinned module affinity`.

ECMP

Equal-Cost Multi-Path. A routing behavior where traffic is distributed across multiple equal-cost paths in the fabric. In an active/active Smart Switch HA pair, ECMP ensures that if one switch loses an uplink, traffic reroutes through the fabric to reach the partner switch, without the need for an explicit failover event.

enforcement point

Any location, such as a Smart Switch, where security policies are enforced.

etcd

The distributed key-value store that is the persistent data layer of the Hypershield platform. etcd stores all Kubernetes object state including `SmartSwitchNetworkPolicy` and `NetworkObjectGroup` resources. The control plane cluster runs a three-node etcd quorum. At least two of the three etcd members must be healthy for the control plane to accept new API requests.

etcdctl

The command-line tool for inspecting etcd cluster state. Used primarily to verify quorum health. Quorum requires at least two of three members to report healthy.

© 2026 Cisco and/or its affiliates. All rights reserved.

fail-closed

The behavior of the DPU when it lacks explicit policy guidance for a traffic flow - it drops the traffic rather than permitting it. This applies in three scenarios: (1) at switch registration, before any allow policies are created; (2) during the reboot window before policy is restored from the control plane; (3) when the DPU has no match for a flow against any rule (default-deny). Hypershield never fails open.

feature service-acceleration

The NX-OS feature flag that must be enabled on a Smart Switch before the Hypershield service can be configured. This enables the NX-OS service acceleration subsystem that manages DPU steering and integration.

gRPC

The remote procedure call protocol used for communication between the Hypershield control plane and the Smart Switches. These channels use mutual TLS (mTLS) for authentication.

HA peer link

The dedicated network connection between the two Smart Switches in an active/active HA pair. Carries HA heartbeat traffic and DPU state synchronization. Not to be confused with a vPC peer-link, which is the NX-OS vPC control path.

hold-down

The behavior of the DPU when the control plane link to the Hypershield control plane is severed. The DPU continues enforcing the complete policy set it held at the time the link went down. It does not flush rules, relax policy, or fail open. New policy changes cannot be pushed to the switch until connectivity is restored. This is the intended behavior - a management outage must never degrade the security posture.

hubble-rbac

A JWT claim carrying the caller’s role(s). The configured OIDC provider is expected to include this claim in every issued token. Timescape reads the claim value to determine which API operations the caller may perform; a missing/invalid/expired token produces an HTTP 401, whereas a token lacking appropriate authorization results in a 403.

in-service

The NX-OS keyword that activates the Hypershield service firewall configuration and begins traffic steering to the DPU. Must be present in the service firewall block for any VRF or VLAN steering to take effect. Without it, the DPU is configured but not receiving traffic.

ipBlock

The policy schema field name for an inline CIDR source or destination in a `SmartSwitchNetworkPolicy` rule. An alternative to networkRef. Use ipBlock for one-off rules; use networkRef with a `NetworkObjectGroup` for subnets that appear in multiple policies.

isoctl

The Hypershield lifecycle management CLI tool. Used for control plane cluster bootstrap, component installation, and future upgrades.

© 2026 Cisco and/or its affiliates. All rights reserved.

JWT

JSON Web Token. A compact, signed token format standardised in RFC 7519. When public API RBAC is enabled, Timescape accepts JWTs issued by a configured OIDC provider for public API authentication; clients present them using the Bearer authentication scheme. Token issuance, claims, and lifetime are deployment-specific.

Keycloak

An open-source identity and access management server. Keycloak is one supported OIDC provider for Hypershield; any standards-compliant OIDC provider can be used.

management VM

The virtual machine deployed from the Hypershield bootstrap OVA into vSphere. It serves as the installer host from which all bootstrap and installation steps are run using `isoctl`. Deploying the management VM is the first step of a vSphere installation.

mTLS

mutual TLS. The authentication mechanism used for the gRPC channel between the Hypershield control plane and each Smart Switch. Both the control plane and the switch present certificates during the handshake, establishing bidirectional trust. The initial registration token bootstraps the certificate exchange when the switch is registered.

NetworkObjectGroup

A Kubernetes Custom Resource that defines a named, reusable collection of network objects. This CRD is referenced by name in `SmartSwitchNetworkPolicy` rules. When a `NetworkObjectGroup` is updated, all policies that reference it automatically incorporate the change on the next policy push. The kubectl resource name is `networkobjectgroup`.

network object

A network object represents an individual network parameter such as an endpoint, a CIDR range, VRF, or VLAN. Rules reference network objects directly, or collectively with a `NetworkObjectGroup`.

networkRef

The policy schema field that references a `NetworkObjectGroup` by name as a source or destination in a policy rule. Using networkRef makes policies more readable and easier to maintain than embedding CIDRs directly.

NPU

Neural Processing Unit. A specialized processor designed to accelerate AI and machine learning tasks. An NPU is designed to minimize power consumption for single small operations that run hundreds of times per minute.

NX-OS

The operating system that runs on Cisco Nexus switches. NX-OS hosts the Agent Gateway (AGW) process, manages the DPU hardware, and provides the service system hypershield configuration block through which the switch is connected to the control plane.

OIDC

© 2026 Cisco and/or its affiliates. All rights reserved.

OpenID Connect. The identity federation protocol used by Hypershield for user authentication. Hypershield supports any standards-compliant OIDC provider; the operator must choose and configure one.

OVA

Open Virtualization Appliance. The VMware virtual machine image format used to deliver the Hypershield Installer Appliance. Deployed into vSphere using the scripted `import-ova.sh` helper or through the vSphere UI in either easy or advanced mode.

PCIe bandwidth

The internal bus bandwidth available to each DPU within the Smart Switch. Hypershield's throughput target is expressed as a percentage of available PCIe bandwidth (95%). This value is hardware-dependent and varies by switch model and DPU configuration.

pinned module affinity

A legacy VRF steering mode where each VRF is manually assigned to a specific DPU module number. Contrast with `dynamic module affinity`.

policy domain

A portion of the environment that is treated as a single unit for security purposes. Segmentation designs are different ways of structuring policy across policy domains; zone-based segmentation defines broad policy domains, while workload-centric segmentation defines narrow ones.

policy hold-down

See hold-down.

policy intent

The desired state of security policy as declared in `SmartSwitchNetworkPolicy` objects on the control plane. Distinct from DPU state, which is the actual set of rules programmed into the DPU datapath. In normal operation these are identical. During a control plane disconnect, DPU state is frozen at the last-pushed intent.

protoPorts

The policy schema field that specifies the protocol and port constraints for a policy rule. Each entry in the protoPorts array has a protocol (TCP, UDP, or ICMP) and an optional port number. A rule without protoPorts matches all traffic between the specified source and destination regardless of protocol or port.

registration token

A one-time credential generated on the control plane and submitted from the Smart Switch NX-OS CLI during registration. The token bootstraps the mTLS trust between the control plane and the switch. Tokens are single-use - generate a new token if registration is interrupted before the switch reaches Connected status.

SCC

Security Cloud Control. Cisco's cloud-based management platform. Hypershield can be managed either via SCC or via Timescape.

service firewall

© 2026 Cisco and/or its affiliates. All rights reserved.

The NX-OS configuration that specifies which VRFs and VLANs should have their traffic steered to the DPU for inspection. Only traffic in VRFs or VLANs listed here is inspected; all other traffic bypasses the DPU. Steering is activated when `in-service` is present at the end of the configuration block.

shadowed rule

A policy rule that can never be evaluated because an earlier, more-general rule in the same policy matches all traffic that the shadowed rule would match. The shadowed rule is effectively dead. Fix it by reordering rules so more-specific rules appear before broader rules.

Smart Switch

A Cisco Nexus N9300 Series switch that contains embedded AMD DPUs. In the context of Hypershield, “Smart Switch” specifically means a DPU-capable switch that can be used as a Hypershield enforcement point.

SmartSwitch

The Kubernetes Custom Resource (CRD) that represents a registered Smart Switch on the control plane. Each registered switch has a corresponding `SmartSwitch` object. Its status field reflects the switch’s connection state (Connected, Pending, or Disconnected).

SmartSwitchNetworkPolicy

SSNP. The Kubernetes Custom Resource (CRD) that represents a Hypershield security policy. Each SSNP contains one or more rules, each with an action (allow or deny), a source, a destination, and optional protocol/port constraints. When an SSNP is created or modified, the control plane compiles it into DPU forwarding rules and pushes those rules to all registered Smart Switches. The kubectl resource name is `smartswitchnetworkpolicy` (plural: `smartswitchnetworkpolicies`).

source-interface

The NX-OS loopback interface whose IP address the Smart Switch uses as its source address when communicating with the Hypershield control plane. The control plane must be able to reach this IP address. Configured in the service system hypershield block. A separate source-interface under high-availability specifies the IP used for HA heartbeat traffic between the switch pair.

SSNP

Abbreviation for `SmartSwitchNetworkPolicy`. Used informally in this guide and in team communications.

stateful inspection

The DPU’s connection tracking capability. Rather than evaluating each packet in isolation, the DPU maintains a state table of active TCP/UDP connections and evaluates policy at the connection level. New connections are evaluated against the policy set; established connections that matched an allow rule continue to be forwarded without per-packet re-evaluation. Stateful inspection is the primary DPU enforcement capability.

symmetric hashing

The algorithm used in `dynamic module affinity` DPU traffic steering mode. It distributes individual flows across DPUs based on the connection’s 5-tuple (source/destination IP, port, protocol). Configured in the NX-OS service firewall block.

© 2026 Cisco and/or its affiliates. All rights reserved.

Timescape

The on-prem web UI and public REST API for Hypershield. Timescape provides policy and network object management, event log browsing, and observability. By default, Timescape is not exposed outside the control plane. An administrator configures a supported exposure method with TLS; the resulting URL is deployment-specific. User authentication is typically integrated with OpenID Connect (OIDC).

Top-of-rack

A deployment topology in which the Smart Switch serves as the access-layer leaf switch, with workloads connected directly to it. All traffic entering or leaving those workloads passes through the DPU for inspection. Contrast with Zone topology.

verdict

The policy evaluation outcome for a traffic flow: allow or deny.

vPC

Virtual Port Channel. A Cisco NX-OS feature that allows two physical switches to appear as a single logical switch to downstream devices, enabling dual-homed server connections. In a Top-of-rack topology Hypershield deployment, the two Smart Switches in the HA pair are typically configured as a vPC pair, with the vPC peer-link carrying vPC control traffic. The Hypershield HA peer link is separate from the vPC peer-link.

VRF

Virtual Routing and Forwarding. A network virtualization construct that creates multiple independent routing tables within a single switch. In Hypershield, VRFs are the primary unit of traffic steering - you specify which VRFs should have their traffic inspected by the DPU. Policy rules can target specific VRFs to control inter-VRF traffic flows (e.g., production VRF to management VRF).

VXLAN

Virtual Extensible LAN. A network encapsulation protocol that tunnels Ethernet frames inside UDP packets to extend Layer 2 networks over a Layer 3 underlay. In Zone topology deployments, traffic arrives at the Smart Switch via VXLAN from the border leaf. The VXLAN encapsulation is terminated at the border leaf before the traffic is handed to the Smart Switch.

Zone topology

A deployment topology in which the Smart Switch sits inline behind a border leaf switch. Traffic arrives at the Smart Switch via VXLAN or VLAN handoff from the border leaf, passes through the DPU, and returns to the border leaf. Contrast with Top-of-rack topology.

© 2026 Cisco and/or its affiliates. All rights reserved.

Platform information and support

© 2026 Cisco and/or its affiliates. All rights reserved.

Open a support case

Your first point of contact for Hypershield support is the Cisco TAC team (Technical Assistance Center).

Contact technical support

To submit a support case, you can either use the online support portal or speak with a support representative by phone.

The online portal is the Support Case Manager.

The telephone numbers are listed here:

Region	Phone number
US	1-800-553-2447
Europe	+32 2 704 5555
Asia-Pacific	+61 2 8446 7411

For other locations, consult Cisco Worldwide Support Contacts.

Gather information

When opening a support case of a technical nature, provide as much detail as possible about the issue, including any relevant logs, configurations, and steps to reproduce the problem.

See Collect troubleshooting information for Hypershield, and either attach it to the case, or provide it using a secure file transfer service recommended by your support agent.

© 2026 Cisco and/or its affiliates. All rights reserved.

Legal information

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries.

To view a list of Cisco trademarks, go to www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners.

The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers.

Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2026 Cisco Systems, Inc. All rights reserved.