Cisco Hypershield Reference Documentation

Hypershield reference

Contents

Introduction

Smart Switches

CLI commands

- agwctl
- dpctl
- isoctl

Timescape API v1

- Requests and responses
- Authentication and authorization
- Errors and status codes
- Rate limits
- Versioning and deprecation
- Endpoint reference

Hypershield reference

This is the central depot for Hypershield reference documentation.

Smart Switches

This page lists the supported Smart Switches for each version of Hypershield.

Hypershield 10.0.0

Supported models	Minimum NX-OS version	DPUs per switch	DPU type	Loopback interfaces
`Cisco N9324C-SE1U`	`10.6(3)r`	4	`AMD Elba` (32 GiB DDR4)	2
`Cisco N9348Y2C6D`	`10.6(3)r`	2	`AMD Giglio` (64 GiB DDR5)	2

The two loopback interfaces per switch are used for management (controller source) and HA heartbeat.

`agwctl` command line interface

A CLI tool for controlling the Hypershield SmartSwitch AGW agent over its local IPC socket. It sends control messages, inspects agent state, and applies policy directly to the AGW.

Usage:

```
agwctl [command]
```

Global flags:

Flag	Type	Description
`--cli-socket`	string	Path to the AGW CLI Unix socket
`--json`	bool	Print output in JSON format

`config add dpu`

Add a DPU configuration entry from a JSON file. This appends to the existing DPU set; it does not replace it.

Usage:

```
agwctl config add dpu [flags]
```

Flags:

Flag	Type	Description
`--file`, `-f`	string	Path to the DPU config JSON file

`config add ha`

Add the HA configuration to the AGW from a JSON file. Replaces any existing HA configuration.

Usage:

```
agwctl config add ha [flags]
```

Flags:

Flag	Type	Description
`--file`, `-f`	string	Path to the HA config JSON file

`config load_dpu_config`

Replace the AGW's DPU configuration with the contents of a JSON file. The file must be a single JSON object with the following fields:

- `service_ip`: string
- `service_mac`: string
- `port_low`: int
- `port_high`: int

Usage:

```
agwctl config load_dpu_config --file <file> [flags]
```

Flags:

Flag	Type	Description
`--file`, `-f`	string	Path to the DPU config JSON file

`config remove dpu`

Remove every DPU configuration entry currently held by the AGW.

Usage:

```
agwctl config remove dpu
```

`config remove ha`

Remove the HA configuration currently held by the AGW.

Usage:

```
agwctl config remove ha
```

`config show`

Show every AGW configuration object currently active, including DPU, HA, and syslog export configuration.

Usage:

```
agwctl config show
```

`device show`

Show the full contents of the AGW device gNMI data store.

Usage:

```
agwctl device show
```

`dpu debug`

Show repair state for every DPU peer, including:

- Repair-requested flag and last repair reason
- Reconnect count and last ack timestamp
- Per-policy repair fences currently blocking sync

Use `--dpu` to restrict output to a single DPU number.

Usage:

```
agwctl dpu debug [flags]
```

Flags:

Flag	Type	Description
`--dpu`, `-d`	int	Restrict output to a single DPU number (0 = all DPUs)

`dpu show`

Show the contents of the DPU gNMI data store. Use `--filter` to narrow the output to DPU names matching a regular expression.

Usage:

```
agwctl dpu show [flags]
```

Flags:

Flag	Type	Description
`--filter`	string	Regex filter on DPU name

`dpu status`

Show an aggregate summary of DPU fleet status. Reports counts only, without per-DPU details.

Usage:

```
agwctl dpu status
```

`dpu sync`

Show SyncPolicy convergence state for every DPU peer, including:

- Desired, installed, and pending rule counts
- Last ack timestamp and in-sync-since timestamp
- Reconnect count

Use `--dpu` to restrict output to a single DPU, and `--policy` to include a per-policy drift breakdown for DPUs that have not yet converged.

Usage:

```
agwctl dpu sync [flags]
```

Flags:

Flag	Type	Description
`--dpu`, `-d`	int	Restrict output to a single DPU number (0 = all DPUs)
`--policy`, `-p`	bool	Include a per-policy drift breakdown for DPUs that are mid-reconcile

`ha compatibility`

Show a side-by-side comparison of HA membership and adjacency compatibility fields between the local switch and each HA peer. Use `--peer` to restrict the comparison to a single peer.

Usage:

```
agwctl ha compatibility [flags]
```

Flags:

Flag	Type	Description
`--peer`	string	Restrict the comparison to a single peer by exact IP

`ha debug fail`

Set the `debug_override` HA criterion to false on the local node, forcing it into ha-switchover state. Run `agwctl ha debug ok` to restore normal operation.

Usage:

```
agwctl ha debug fail
```

`ha debug ok`

Set the `debug_override` HA criterion to true on the local node, allowing normal HA criteria evaluation to resume. Clears a failure previously injected by `agwctl ha debug fail`.

Usage:

```
agwctl ha debug ok
```

`ha debug peer-fail`

Inject a debug failure for a specific HA peer's membership or adjacency criterion. Exactly one of `--membership` or `--adjacency` must be set, along with `--peer`. Run `agwctl ha debug peer-ok` to clear the injected failure.

Usage:

```
agwctl ha debug peer-fail [flags]
```

Flags:

Flag	Type	Description
`--adjacency`	bool	Inject an adjacency failure (results in ha-degraded state)
`--membership`	bool	Inject a membership failure (triggers ha-takeover/ha-switchover)
`--peer`	string	Target peer IP address

`ha debug peer-ok`

Clear a debug failure previously injected via `agwctl ha debug peer-fail` for a peer's membership or adjacency HA criterion. Exactly one of the `--membership` or `--adjacency` flags must be set, along with `--peer`.

Usage:

```
agwctl ha debug peer-ok [flags]
```

Flags:

Flag	Type	Description
`--adjacency`	bool	Clear a previously injected adjacency debug failure
`--membership`	bool	Clear a previously injected membership debug failure
`--peer`	string	Target peer IP address

`ha debug reset`

Clear every local and peer-directed HA debug criterion in a single call and (optionally) block until the pair is stable. This replaces the ad-hoc sequence of calls that HA integration tests otherwise issue back-to-back:

- `agwctl ha debug ok`
- `agwctl ha debug peer-ok` `--membership`

- `agwctl ha debug peer-ok` `--adjacency`

Usage:

```
agwctl ha debug reset [flags]
```

Flags:

Flag	Type	Description
`--rounds`	int	Number of full keepalive rounds to observe once the pair is stable
`--wait-stable`	string	Wait for pair stability (Go duration, e.g. 30s); empty = no wait

`ha debug timers reset`

Restore production HA loop timer defaults

Usage:

```
agwctl ha debug timers reset
```

`ha debug timers set`

Override any subset of the HA loop timers. Unset flags leave the current value unchanged. At least one flag is required.

Usage:

```
agwctl ha debug timers set [flags]
```

Flags:

Flag	Type	Description
`--adj-timeout`	string	Adjacency + member expiry timeout (Go duration, e.g. 1s)
`--holddown`	string	Anti-flap recovery hold-down (Go duration, e.g. 2s)
`--keepalive`	string	Adjacency ticker interval (Go duration, e.g. 200ms)
`--policy-settling`	string	Peer policy mismatch settling window (Go duration, e.g. 5s)

`ha debug timers show`

Show the currently-active HA loop timers

Usage:

```
agwctl ha debug timers show
```

`ha gids`

Show VRFs with GID or preset allocations on the local switch and every HA peer. Use this to verify GID reconciliation across the HA cluster.

Usage:

```
agwctl ha gids
```

`ha info`

Show the static HA configuration, including HA IP and port, admin and operational state, leader status, and policy settings.

Usage:

```
agwctl ha info
```

`ha peers`

Show detailed HA state for every peer. Use `--filter` to narrow the output to peer IPs matching a regular expression.

Usage:

```
agwctl ha peers [flags]
```

Flags:

Flag	Type	Description
`--filter`	string	Regex filter on peer IP address

`ha show`

Show the HA cluster state, the local node state, and a per-peer summary with `[OK]`/`[FAIL]` indicators for membership, adjacency, and service.

Usage:

```
agwctl ha show
```

`ha vlans`

Show VLANs with ID allocations on the local switch and every HA peer. Use this to verify VLAN reconciliation across the HA cluster.

Usage:

```
agwctl ha vlans
```

`health`

Report the AGW agent's health as observed by the local IPC health probe.

Usage:

```
agwctl health
```

`load_syslog_cfg`

Load the syslog export configuration from a JSON file and push it to all DPU agents, replacing any existing configuration. The file must contain a JSON object keyed by export ID, where each entry has:

- `id`, `name`, `description`, `type`: string
- `config`: `host`, `port`, `mode` (string) and `tls` (bool)
- `secrets`: `token`, `ca`, `cert`, `key`, `keyPassword` (string)

Usage:

```
agwctl load_syslog_cfg <file>
```

`logging`

Set the AGW process log level. Supported values for `--level`:

- `debug`
- `info` (default)
- `warn`
- `error`

Usage:

```
agwctl logging [flags]
```

Flags:

Flag	Type	Description
`--level`, `-l`	string	Log level: debug, info, warn, or error

`metrics show`

Show a snapshot of current AGW runtime metrics, including memory usage, CPU usage, policy counts, and DPU insert/update/delete error counters.

Usage:

```
agwctl metrics show
```

`policies add`

Add a policy to the AGW from a YAML file.

WARNING: Adding policies locally bypasses the on-prem controller and may leave the AGW OUT OF SYNC with it.

Usage:

```
agwctl policies add [flags]
```

Flags:

Flag	Type	Description
`--file`, `-f`	string	Path to the policy YAML file

`policies clear`

Remove every policy currently loaded in the AGW.

WARNING: Clearing policies locally bypasses the on-prem controller and may leave the AGW OUT OF SYNC with it. To re-sync after clearing, restart `HypershieldAgent` from the NX-OS CLI (`no in-service` then `in-service`).

Usage:

```
agwctl policies clear
```

`policies info`

Display a summary of loaded policies including:

- Rule limits (global and per-policy; 0 = unlimited)
- Total number of policies and rules
- Number of allow vs deny rules
- Policies and rules per namespace
- VRFs and VLANs in use
- Protocol distribution

Flags:

- `--persisted`: Summarize the on-disk checkpoint via AGW's already-open DB.
- `--diff`: Report only ResourceIDs whose expanded rule count differs between the live handler and the persisted set. Identical entries collapse into a single count.

Usage:

```
agwctl policies info [flags]
```

Flags:

Flag	Type	Description
`--diff`	bool	Report only mismatches between the live handler and the persisted set
`--persisted`	bool	Summarize the on-disk checkpoint instead of the live handler

`policies limit`

Get or set the rule limits the AGW enforces on incoming policies. When either limit is exceeded, the incoming policy upsert is rejected with a rate-limit error. Set a limit to 0 to disable that specific check.

Running with no flags prints the current limits.

Defaults:

- `--set`: 1000000 (global, across all policies)
- `--set-per-policy`: 75000 (single policy)

Usage:

```
agwctl policies limit [flags]
```

Flags:

Flag	Type	Description
`--set`	string	Set the global maximum number of rules across all policies (0 disables the check)
`--set-per-policy`	string	Set the maximum number of rules a single policy may contain (0 disables the check)

`policies remove`

Remove a policy from the AGW, identified either by its ResourceID passed as a positional argument or by a YAML file supplied via `--file`.

WARNING: Removing policies locally bypasses the on-prem controller and may leave the AGW OUT OF SYNC with it. To re-sync after removal, restart `HypershieldAgent` from the NX-OS CLI (`no in-service` then `in-service`).

Usage:

```
agwctl policies remove [resourceId] [flags]
```

Flags:

Flag	Type	Description
`--file`, `-f`	string	Path to the policy YAML file identifying the policy to remove

`policies show`

Show the policies currently loaded in the AGW. Each policy's ResourceID is `kind/namespace/name` as it appears in Kubernetes. Use `--filter` with a regular expression to narrow the list.

Pass `--persisted` to stream the on-disk checkpoint (bbolt) instead of the live handler. AGW walks its already-open DB in-process, so this is safe to run while AGW is up. Rule counts reflect the post-expansion form so the output matches the live view.

Usage:

```
agwctl policies show [flags]
```

Flags:

Flag	Type	Description
`--filter`	string	Regex filter on policy ResourceID
`--persisted`	bool	Read from the on-disk checkpoint instead of the live handler

`policies status`

Stream one frame per policy with desired and installed rule counts, matching RuleNames, and active SyncPolicy conditions per DPU.

Use `--policy` to restrict output to a single Kubernetes policy, and use `--unhealthy-only` to emit only policies with active SyncPolicy conditions.

Usage:

```
agwctl policies status [flags]
```

Flags:

Flag	Type	Description
`--policy`, `-p`	string	Restrict output to a single Kubernetes policy (kind/namespace/name)
`--unhealthy-only`	bool	Emit only policies with active SyncPolicy conditions

`policies translate`

Translate policies into the JSON rule format consumed by the DPU dataplane.

By default the current AGW policy set is translated using the AGW's active VRFs. Use `--file` to translate policies from a YAML file instead, or `--no-vrfs` to translate every policy while ignoring active VRFs.

Usage:

```
agwctl policies translate [flags]
```

Flags:

Flag	Type	Description
`--file`, `-f`	string	Path to a policy YAML file to translate instead of the current AGW policy set
`--no-vrfs`	bool	Translate every policy, ignoring the AGW's active VRF configuration

`policies validate`

Walk the on-disk policy checkpoint database and report any entries that `RestorePolicyCheckpoint` would skip at startup — malformed payload, unsupported `schemaVersion`, or missing `kind`/`namespace`/`name`. On success, also prints an info-style summary of the accepted entries.

Runs entirely on this host — no AGW round-trip. AGW holds an exclusive lock on the checkpoint file while running, so this command only succeeds when the agent is stopped (or against a copy of the file inspected via `--dir`).

Usage:

```
agwctl policies validate [flags]
```

Flags:

Flag	Type	Description
`--dir`	string	override checkpoint directory (default: /iox_data/isovalent/policy/.checkpoint)

`policies wait`

Block until all policies report no active SyncPolicy conditions across all DPUs, or until the timeout elapses.

Use `--timeout` to cap the wait duration. Exits non-zero on timeout and exits 130 on `SIGINT`.

Usage:

```
agwctl policies wait [flags]
```

Flags:

Flag	Type	Description
`--timeout`, `-t`	duration	Maximum time to wait for convergence before exiting non-zero

`show_dpu`

Show DPU peer status in the legacy table format. Prefer `agwctl dpu show` for the current, richer output.

Usage:

```
agwctl show_dpu
```

`show_log`

Show a head and tail excerpt of the AGW process log buffer maintained by the agent on disk.

Usage:

```
agwctl show_log
```

`show_status`

Show the AGW agent's operational status, including uptime, IPC state, and DPU connectivity.

Usage:

```
agwctl show_status
```

`show_syslog_cfg`

Show the syslog export configuration the AGW has currently pushed to DPU agents.

Usage:

```
agwctl show_syslog_cfg
```

`show_tech`

Show a full AGW tech-support snapshot, aggregating status, metrics, policies, configuration, VRF/VLAN/DPU/HA state, and syslog output for support diagnostics.

This is a compatibility alias for `agwctl tech-support show`. Prefer the subcommand form in new scripts.

Usage:

```
agwctl show_tech
```

`show_timescape_cfg`

Show the Timescape client configuration used by the AGW, including:

- Client status (enabled or disabled)
- Server information: endpoint URL and username (passwords are redacted)
- gRPC configuration: enabled status and port

Usage:

```
agwctl show_timescape_cfg
```

`show_tokens`

Show metadata for the AGW's cached access and refresh tokens. The token value itself is only printed when JSON output is requested via `--json`.

Usage:

```
agwctl show_tokens
```

`tac_pac`

Collect a Cisco TAC support bundle on the local switch.

This is a compatibility alias for `agwctl tech-support bundle` that writes the bundle to the historical path `/iox_data/agw_logs.tgz`. The bundle contains AGW logs and configuration, the on-disk policy checkpoint, system state, and — when the AGW agent is reachable over IPC — its `show_tech` output plus a DPU log archive.

Core dumps are excluded (they can be multi-GB per file). Collect them separately with `agwctl tech-support cores` or via the NX-OS `show cores` command.

Prefer `agwctl tech-support bundle` for new integrations; it writes a timestamped file under `/data/volatile` and accepts flags for finer control.

Usage:

```
agwctl tac_pac
```

`tech-support bundle`

Collect a compressed tech-support bundle on the local switch.

The bundle always includes local artifacts (AGW logs and config, on-disk policy checkpoint, and system state such as `dmesg` / `ps` / `df`). It also attempts to contact the running AGW agent over its IPC socket to include `show_tech` output and a DPU log archive.

Core dumps are NOT included by default — cores may be multi-GB per file and are handled by `agwctl tech-support cores` on their own. Pass `--include-cores` to fold them into this bundle anyway.

If AGW is not running (crashed, restarting, or not deployed) the local artifacts are still collected — the IPC-derived sections are simply marked `skip` in the bundle's `MANIFEST.txt`. This is the intended behaviour: a tech-support bundle is most valuable exactly when the agent is unhealthy.

Bundle path defaults to `/data/volatile/agw_tech_support_<UTC-timestamp>.tar.gz`.

Usage:

```
agwctl tech-support bundle [flags]
```

Flags:

Flag	Type	Description
`--include-cores`	bool	Include core dumps in the bundle. Off by default — cores can be multi-GB.
`--output`, `-o`	string	Output path for the tech-support archive (default: /data/volatile/agw_tech_support_<timestamp>.tar.gz)
`--skip-dpu`	bool	Skip the DPU log collection over IPC. show_tech is still attempted.
`--skip-ipc`	bool	Skip the AGW IPC-mediated collection (show_tech and DPU logs). Use when AGW is known crashed.

`tech-support cores`

Collect a tarball containing only core dumps found on the local switch.

Cores are copied newest-first with a per-bundle budget (see `MANIFEST.txt` for details); anything past the budget is recorded in `core_dumps/_SKIPPED.txt` so support can request it out-of-band. No IPC calls are made — this command works whether or not AGW is running.

Output defaults to `/data/volatile/agw_cores_<UTC-timestamp>.tar.gz`.

Usage:

```
agwctl tech-support cores [flags]
```

Flags:

Flag	Type	Description
`--output`, `-o`	string	Output path for the cores tarball (default: /data/volatile/agw_cores_<timestamp>.tar.gz)

`tech-support dpu`

Collect a tech-support archive from every DPU peer. This subcommand requires the AGW agent to be running — it uses the agent's SSH credentials to reach each DPU.

If AGW is not running, use `agwctl tech-support bundle` instead: that collects local artifacts even when the agent has crashed.

Usage:

```
agwctl tech-support dpu
```

`tech-support show`

Print a full AGW tech-support snapshot to stdout, aggregating status, metrics, policies, configuration, VRF/VLAN/DPU/HA state, and syslog output. Requires the AGW agent to be running — this command talks to the agent's IPC socket.

Use `agwctl tech-support bundle` when AGW is crashed: it collects local artifacts and marks the IPC-derived sections as skipped in the bundle's `MANIFEST.txt`.

Usage:

```
agwctl tech-support show
```

`vlan info`

Show aggregate VLAN statistics, including total count, active count, ID assignment, and pinning distribution.

Usage:

```
agwctl vlan info
```

`vlan list`

Print a compact table of active VLANs, followed by a count summary.

Usage:

```
agwctl vlan list
```

`vlan show`

Show the contents of the VLAN gNMI data store. Use `--filter` to narrow the output to VLAN names matching a regular expression.

Usage:

```
agwctl vlan show [flags]
```

Flags:

Flag	Type	Description
`--filter`	string	Regex filter on VLAN name

`vrf gids`

Show the GID allocation table for every VRF currently holding a GID on the local switch.

Usage:

```
agwctl vrf gids
```

`vrf info`

Show aggregate VRF statistics, including total count, active count, GID usage, and pinning distribution.

Usage:

```
agwctl vrf info
```

`vrf list`

Print a compact table of active VRFs, followed by a count summary.

Usage:

```
agwctl vrf list
```

`vrf show`

Show the contents of the VRF gNMI data store. Use `--filter` to narrow the output to VRF names matching a regular expression.

Usage:

```
agwctl vrf show [flags]
```

Flags:

Flag	Type	Description
`--filter`	string	Regex filter on VRF name

`dpctl` command line interface

A CLI tool for controlling the AMD/Pensando dataplane services on a Hypershield DPU. It inspects and clears dataplane state, and drives debug operations against the local dataplane app over its gRPC or Unix domain socket transport.

Usage:

```
dpctl [flags] [command]
```

Global flags:

Flag	Type	Description
`--dsc-svc-ip`	string	Remote DSC's service URL
`--dsc-svc-port`	string	Remote DSC's service port

Flags:

Flag	Type	Description
`--toggle`, `-t`	bool	Help message for toggle

`clear collector-session statistics`

clear collector session statistics

Usage:

```
dpctl clear collector-session statistics [flags]
```

Flags:

Flag	Type	Description
`--session-id`, `-i`	uint32	Specify collector session ID (valid range 1-8192)

`clear flow`

clear flow object information

Usage:

```
dpctl clear flow
```

`clear pipeline statistics`

clear pipeline statistics

Usage:

```
dpctl clear pipeline statistics [command]
```

`clear pipeline statistics datapath-assist`

clear pipeline datapath assist statistics

Usage:

```
dpctl clear pipeline statistics datapath-assist
```

`clear pipeline statistics drop`

clear pipeline statistics

Usage:

```
dpctl clear pipeline statistics drop
```

`clear session`

clear a specific session by ID and its associated flows

Usage:

```
dpctl clear session [flags]
```

Flags:

Flag	Type	Description
`--high`	uint32	Specify high session id for range
`--id`	uint32	Specify session id to clear (single session)
`--low`	uint32	Specify low session id for range

`debug create collector-session`

create collector session

Usage:

```
dpctl debug create collector-session [flags]
```

Flags:

Flag	Type	Description
`--dst-ip`	string	Specify collector session destination IP address
`--dst-mac`	string	Specify collector session destination MAC address
`--forwarding-port`	uint32	Specify forwarding port (1 for UPLINK_1 and 2 UPLINK_2), default is UPLINK_1
`--session-id`, `-i`	uint32	Specify collector session ID (valid range 1-8192)
`--src-ip`	string	Specify collector session source IP address
`--src-mac`	string	Specify collector session source MAC address
`--tap-length`	uint32	Specify TAP length (valid range 64-9000), default is full packet length

`debug create telemetry-policy`

create telemetry policy

Usage:

```
dpctl debug create telemetry-policy [flags]
```

Flags:

Flag	Type	Description
`--capture-only`	bool	Specify whether received packets are for capture only
`--dst-ip-prefix`	string	Specify destination IP prefix. Default is any destination IP
`--dst-port`	uint32	Specify destination port. Default is any destination port
`--erspan-collector-session-id`	string	Specify ERSPAN collector session IDs in comma separated list in the range 1-8192
`--flowmon-collector-session-id`	string	Specify FlowMon collector session IDs in comma separated list in the range 1-8
`--ip-proto`	string	Specify IP protocol number. Default is any protocol
`--observation-domain-id`	uint32	IPFIX Observation Domain ID (RFC 7011) stamped on emitted IPFIX message headers. 0 = unspecified
`--packet-direction`	string	Specify the direction of traffic with respect to dsc - ingress|egress|all
`--policy-id`, `-i`	uint32	Specify telemetry policy ID (valid range 1-1024)
`--src-ip-prefix`	string	Specify source IP prefix. Default is any source IP
`--src-port`	uint32	Specify source port. Default is any source port

`debug datapath-assist`

datapath assist debug command

Usage:

```
dpctl debug datapath-assist [flags]
```

Flags:

Flag	Type	Description
`--num-cores`, `-c`	uint32	Specify number of cores assigned for datapath assist(1-12)

`debug delete collector-session`

delete collector session

Usage:

```
dpctl debug delete collector-session [flags]
```

Flags:

Flag	Type	Description
`--session-id`, `-i`	uint32	Specify collector session ID (valid range 1-8192)

`debug delete telemetry-policy`

delete telemetry policy

Usage:

```
dpctl debug delete telemetry-policy [flags]
```

Flags:

Flag	Type	Description
`--policy-id`, `-i`	uint32	Specify telemetry policy ID (valid range 1-1024)

`debug pipeline internal packet-trace-rule create`

add packet trace rule

Usage:

```
dpctl debug pipeline internal packet-trace-rule create [flags]
```

Flags:

Flag	Type	Description
`--dst-ip-prefix`	string	Specify destination IP prefix
`--dst-port`	uint32	Specify destination port
`--icmp-code`	uint32	Specify ICMP code
`--icmp-type`	uint32	Specify ICMP type
`--ip-proto`	string	Specify IP protocol number or any
`--key`	uint32	Specify unique rule key in the range of [0, 7]
`--src-ip-prefix`	string	Specify source IP prefix
`--src-port`	uint32	Specify source port
`--tcp-flags`	string	Specify TCP flags as comma separated list

Flag	Type	Description
`--vni`	uint32	Specify vni

`debug pipeline internal packet-trace-rule delete`

delete packet trace rule

Usage:

```
dpctl debug pipeline internal packet-trace-rule delete [flags]
```

Flags:

Flag	Type	Description
`--key`	uint32	Specify unique rule key in the range of [0, 7]

`debug port-counters clear`

Reset all port operation counters to zero

Usage:

```
dpctl debug port-counters clear
```

`debug port-counters show`

Display statistics about port operations in the policy engine

Usage:

```
dpctl debug port-counters show
```

`debug update`

debug update commands

Usage:

```
dpctl debug update
```

`hs fwa_message`

fwa_message

Usage:

```
dpctl hs fwa_message [flags]
```

Flags:

Flag	Type	Description
`--file`, `-f`	string	Message file path

`hs ha disable`

disable

Usage:

```
dpctl hs ha disable
```

`hs ha enable`

enable

Usage:

```
dpctl hs ha enable
```

`hs ha peers add`

add

Usage:

```
dpctl hs ha peers add [flags]
```

Flags:

Flag	Type	Description
`--dport`	uint32	Destination port (our service port)
`--dst-ip`	string	Destination IP (our service ip)
`--src-ip`	string	Source peer IP

`hs ha peers clear`

Clear HA flow sync statistics shown by hs ha peers show

Usage:

```
dpctl hs ha peers clear
```

`hs ha peers show`

Show HA peer status including local/peer IPs, ports, state (UP/DOWN/UNKNOWN), and keepalive sequence numbers

Usage:

```
dpctl hs ha peers show
```

`hs ha steering show`

Show HA flow sync steering table entries programmed for P4 qid steering

Usage:

```
dpctl hs ha steering show
```

`hs ha timer`

timer commands

Usage:

```
dpctl hs ha timer [command]
```

`hs ha timer delete_query`

Set HA delete query timeout in milliseconds

Usage:

```
dpctl hs ha timer delete_query <ms>
```

`hs ha timer delete_query_idle`

Set HA delete query idle timer in milliseconds

Usage:

```
dpctl hs ha timer delete_query_idle <ms>
```

`hs ha timer interval`

Set HA keepalive interval in milliseconds

Usage:

```
dpctl hs ha timer interval <ms>
```

`hs ha timer keepalive`

Set HA max missed keepalive threshold

Usage:

```
dpctl hs ha timer keepalive <count>
```

`hs ha timer keepalive_up_update`

Set periodic HA status replay interval in seconds. A value of 0 disables periodic replay.

Usage:

```
dpctl hs ha timer keepalive_up_update <sec>
```

`hs ha timer show`

Show HA timer settings

Usage:

```
dpctl hs ha timer show
```

`hs ha timer timeout`

Set HA keepalive timeout in milliseconds

Usage:

```
dpctl hs ha timer timeout <ms>
```

`hs log_export show`

Show log export status including configuration and collectors

Usage:

```
dpctl hs log_export show
```

`hs logging`

logging {error | info | verbose | debug | drops | timestamp {enable | disable}} If the level is omitted, the current log level will be printed. Use ‘timestamp enable’ or ‘timestamp disable’ to control timestamps. Use ‘drops’ to log only dropped packets with details.

Usage:

```
dpctl hs logging [command]
```

`hs logging categorize`

Enable or disable categorized logging. When enabled, logs are routed to separate files based on category (HA, Policy, Main).

Usage:

```
dpctl hs logging categorize [flags]
```

Flags:

Flag	Type	Description
`--enable`	bool	Enable categorized logging to separate files

`hs logging category`

Set log level for a specific category. When a category’s log level is set independently, logs for that category are automatically routed to a separate file.

Categories: main - General/main logs (dp-app.log) ha - HA-related logs (dp-app-ha.log) policy - Policy-related logs (dp-app-policy.log)

Log levels: error, info, debug, verbose, drop

Examples: dpctl hs logging category `--name` ha `--level` debug # Set HA logs to debug, route to dp-app-ha.log dpctl hs logging category `--name` policy `--level` info # Set Policy logs to info, route to dp-app-policy.log dpctl hs logging category `--name` ha `--reset` # Reset HA to global level, stop separate file dpctl hs logging category `--name` ha # Show current HA log level

Usage:

```
dpctl hs logging category [flags]
```

Flags:

Flag	Type	Description
`--level`	string	Log level: error, info, debug, verbose, drop
`--name`	string	Category name: main, ha, or policy (required)
`--reset`	bool	Reset category to use global log level

`hs logging drop clear`

Clear all captured packets from the buffer

Usage:

```
dpctl hs logging drop clear
```

`hs logging drop show`

Display all captured packets in the buffer

Usage:

```
dpctl hs logging drop show
```

`hs logging drop start`

Start capturing dropped packets with optional category filter.

Categories: all - Capture all drops (default) policy - Policy-related drops (explicit deny, default deny, no match) parse - Packet parsing/header drops (invalid header, unsupported ethertype) flow - Flow/session programming drops ha - HA-related drops (control messages, errors)

Usage:

```
dpctl hs logging drop start [flags]
```

Flags:

Flag	Type	Description
`--category`	string	Drop category filter (all, policy, parse, flow, ha)

`hs logging drop status`

Show packet drop logging status and statistics

Usage:

```
dpctl hs logging drop status
```

`hs logging drop stop`

Stop capturing dropped packets

Usage:

```
dpctl hs logging drop stop
```

`hs logging drop summary`

Show number of packets dropped per policy

Usage:

```
dpctl hs logging drop summary
```

`hs logging drop write`

Export captured packets from buffer to file

Usage:

```
dpctl hs logging drop write [flags]
```

Flags:

Flag	Type	Description
`--file`	string	Output file path
`--format`	string	Output format (text or json)

`hs pol disable`

disable

Usage:

```
dpctl hs pol disable
```

`hs pol enable`

enable

Usage:

```
dpctl hs pol enable
```

`hs policies clear`

clear

Usage:

```
dpctl hs policies clear
```

`hs policies limit`

Set the hypershield policy limit

Usage:

```
dpctl hs policies limit <count>
```

`hs policies show`

show

Usage:

```
dpctl hs policies show [flags]
```

Flags:

Flag	Type	Description
`--summary`	bool	Show policy list summary

`hs policies update`

update

Usage:

```
dpctl hs policies update [flags]
```

Flags:

Flag	Type	Description
`--file`, `-f`	string	Profile file path

`hs policy-update-timer`

policy update timer (500ms to 600000ms)

Usage:

```
dpctl hs policy-update-timer
```

`hs vrf add`

Add VRF information

Usage:

```
dpctl hs vrf add
```

`hs vrf del`

Delete VRF information

Usage:

```
dpctl hs vrf del
```

`hs vrf disable`

disable

Usage:

```
dpctl hs vrf disable
```

`hs vrf enable`

enable

Usage:

```
dpctl hs vrf enable
```

`hs vrf show`

show

Usage:

```
dpctl hs vrf show
```

`hs vrf show-map`

Display VRF Map table information

Usage:

```
dpctl hs vrf show-map
```

`show collector-session`

show collector session

Usage:

```
dpctl show collector-session [flags] [command]
```

Flags:

Flag	Type	Description
`--session-id`, `-i`	uint32	Specify collector session ID (valid range 1-8192)

`show collector-session statistics`

show collector session statistics

Usage:

```
dpctl show collector-session statistics [flags]
```

Flags:

Flag	Type	Description
`--session-id`, `-i`	uint32	Specify collector session ID (valid range 1-8192)

`show dpu-capacity`

show DPU table capacity and resource usage in hardware

Usage:

```
dpctl show dpu-capacity [command]
```

`show dpu-capacity summary`

show DPU capacity summary with table and worker thread usage

Usage:

```
dpctl show dpu-capacity summary
```

`show flow`

show flow object information

Usage:

```
dpctl show flow [flags]
```

Flags:

Flag	Type	Description
`--dst-ip`	string	Specify dsetination IP
`--dstport`	uint32	Specify flow destination port
`--ipproto`	uint32	Specify flow IP proto
`--src-ip`	string	Specify source IP
`--srcport`	uint32	Specify flow source port
`--summary`	bool	Display number of flows
`--vrfid`	uint32	Specify VRF ID

`show flow-analysis`

show detailed flow/session/policy analysis for a specific flow filtered by 5-tuple (supports both IPv4 and IPv6)

Usage:

```
dpctl show flow-analysis [flags]
```

Flags:

Flag	Type	Description
`--all`	bool	Show all matching flows (overrides `--limit`)
`--dst-ip`	string	Specify destination IP (IPv4 or IPv6)
`--dstport`	uint32	Specify flow destination port
`--ipproto`	uint32	Specify flow IP proto
`--limit`	uint32	Limit number of matches (default 1)
`--src-ip`	string	Specify source IP (IPv4 or IPv6)
`--srcport`	uint32	Specify flow source port

`show npu_address`

npu_address

Usage:

```
dpctl show npu_address
```

`show pipeline internal table constant`

show P4 pipeline table constant

Usage:

```
dpctl show pipeline internal table constant
```

`show pipeline internal table info`

show P4 pipeline table information

Usage:

```
dpctl show pipeline internal table info
```

`show pipeline internal table nacl`

show nacl information

Usage:

```
dpctl show pipeline internal table nacl
```

`show pipeline internal table nexthop`

show nexthop table entry information

Usage:

```
dpctl show pipeline internal table nexthop [flags]
```

Flags:

Flag	Type	Description
`--hw-id`	uint32	Specify the NextHop ID to display
`--json`	bool	display in json format
`--yaml`	bool	display in yaml format

`show pipeline internal table rewrite`

show rewrite table entry information

Usage:

```
dpctl show pipeline internal table rewrite [flags]
```

Flags:

Flag	Type	Description
`--hw-id`	uint32	Specify the Rewrite ID to display
`--json`	bool	display in json format
`--yaml`	bool	display in yaml format

`show pipeline statistics`

show pipeline statistics

Usage:

```
dpctl show pipeline statistics [command]
```

`show pipeline statistics datapath-assist`

Show pipeline datapath assist statistics

Usage:

```
dpctl show pipeline statistics datapath-assist [flags]
```

Flags:

Flag	Type	Description
`--cps`	bool	display connections/second history
`--cps_detail`	bool	display per thread connections/second history
`--detail`	bool	display statistics details
`--duration`	uint32	display connections/second history for the duration
`--summary`	bool	display statistics summary

`show pipeline statistics drop`

show pipeline drop statistics

Usage:

```
dpctl show pipeline statistics drop
```

`show session`

show session object information

Usage:

```
dpctl show session [flags]
```

Flags:

Flag	Type	Description
`--id`	uint32	Specify session id

`show telemetry-policy`

show telemetry policy

Usage:

```
dpctl show telemetry-policy [flags]
```

Flags:

Flag	Type	Description
`--policy-id`, `-i`	uint32	Specify telemetry policy ID (valid range 1-1024)

`show uplink-pinning`

show uplink pinning

Usage:

```
dpctl show uplink-pinning
```

`show version`

show version information

Usage:

```
dpctl show version
```

`tcp-state-check disable`

Disable TCP state machine compliance checking. When disabled, packets that violate the TCP state machine will not be dropped.

Usage:

```
dpctl tcp-state-check disable
```

`tcp-state-check enable`

Enable TCP state machine compliance checking. When enabled, packets that violate the TCP state machine will be dropped.

Usage:

```
dpctl tcp-state-check enable
```

`tcp-state-check show`

Show whether TCP state machine compliance checking is enabled in the P4 datapath.

Usage:

```
dpctl tcp-state-check show
```

`isoctl` command line interface

A CLI tool for managing the Isovalent Enterprise and Hypershield platforms.

Usage:

```
isoctl [command]
```

Global flags:

Flag	Type	Description
`--log-format`	string	Log format (“text” or “pretty”).
`--log-level`, `-l`	string	Log level (“debug”, “error” or “info”).

`bootstrap capi`

Install and initialize CAPI (with no infra providers) on the cluster.

Usage:

```
isoctl bootstrap capi
```

`bootstrap install`

Bootstrap and install a new management cluster.

Usage:

```
isoctl bootstrap install [flags]
```

Flags:

Flag	Type	Description
`--address`	string	IP address to be used as the management cluster control plane VIP.
`--cilium-version`	string	Version of cilium to install on the management cluster.
`--config`, `-f`	string	Path to BootstrapConfig config file

Flag	Type	Description
`--config-dry-run`	bool	Output derived BootstrapConfig config and exit
`--controlplane-replicas`	int	The number of control plane replicas to provision for the management cluster.
`--create-authorized-key`	bool	Create a key-pair for management cluster node authorization if no keys are provided.
`--image-archives`	list	Specify additional container image archives to be loaded onto the management cluster nodes.
`--ipam-addresses`	list	IPAM pool addresses
`--ipam-gateway`	string	**Deprecated**: IPAM pool gateway
`--ipam-nameservers`	list	**Deprecated**: IPAM pool nameservers
`--ipam-prefix`	uint8	**Deprecated**: IPAM pool prefix
`--ipampool-name`	string	**Deprecated**: IPAM pool name
`--manifest`, `-m`	string	Specify a MgmtCluster manifest to deploy.
`--name`	string	Name of the management cluster to be created.
`--namespace`	string	Namespace to create the MgmtCluster in. By default, it's the name of the management cluster.
`--node-authorized-keys`	list	Specify additional public keys to be authorized for SSH on the management cluster nodes.
`--node-authorized-keys-from-agent`	bool	Add management cluster node authorized keys from ssh-agent. $SSH_AUTH_SOCK will be used as the socket address.
`--node-user`	string	Name of the default user on all provisioned nodes. By default, it will be "ilb".
`--operator-image`	string	Override the isovalent-operator image URI to be used.
`--pod-cidr`	string	IP address range to be allocated for Pods.
`--seed-concurrency`	int	Number of images to push to the registry in parallel during seeding.
`--seed-from-local-image-archive`	list	Specify additional container image archives to seed the management cluster registry with.
`--service-cidr`	string	IP address range to be allocated for Services.
`--skip-cleanup`	bool	Skip cleanup of temporary resources after bootstrap is completed.
`--strict-config`	bool	Enable strict bootstrap config parsing (reject unknown fields)
`--timeout`, `-t`	duration	Maximum time to complete install process
`--validation-mode`	string	Validation mode: offline, online, or all
`--validation-trace`	bool	Enable detailed validation trace logs (for deep troubleshooting)
`--vsphere-datacenter`	string	Specify the datacenter node VMs should provision to.
`--vsphere-datastore`	string	Specify the datastore node VMs should allocate disks from.
`--vsphere-endpoint`	string	Endpoint for the target vSphere server.
`--vsphere-folder`	string	Configure the inventory folder to organize node VMs under.
`--vsphere-network`	string	Specify the default network for node VMs.
`--vsphere-password`	string	Password to use for authentication.

Flag	Type	Description
`--vsphere-resource-pool`	string	Specify the name of the resource pool node VMs should be scheduled to.
`--vsphere-template`	string	Provide a path to save the node OVA to. On bootstrap the CLI will upload the embedded node OVA to this path. All node VMs will be provisioned from this image.
`--vsphere-thumbprint`	string	Thumbprint of the vSphere server.
`--vsphere-username`	string	Username of the vSphere user that the control plane should assume the identity of.
`--worker-pool-name`	string	Name of the management cluster worker node pool.
`--worker-replicas`	int	The number of worker replicas to provision for the management cluster.

Examples:

```
# Install bootstrap cluster interactively
isoctl bootstrap install

# Install bootstrap cluster from config
isoctl bootstrap install -f bootstrap.yaml
```

`bootstrap validate`

Validate a bootstrap config file with offline and/or online checks.

Offline checks:

- required bootstrap fields (vSphere settings, networks)
- proxy settings and noProxy value format
- network/IPAM values (names, ranges, prefix, nameservers, optional gateway)
- no overlap between networks[*].ipam.addresses and effective pod/service CIDRs (mgmtCluster.podCIDR/serviceCIDR or defaults)
- network IPAM address capacity per configured segment against required node replicas (controlPlaneReplicas + workerReplicas)
- management cluster API and SSH auth strategy settings
- mgmtCluster.vmProfile value (must be one of: standard, hypershield)
- mgmtCluster.vmSizing bounds (numCPUs: 4..256, memoryMiB: 8192..1048576 when set)
- warning when both vmProfile and vmSizing are set (vmSizing overrides vmProfile/defaults per field; unset fields keep profile/default values)
- when using nodeAuthorizedKeysFromAgent as the only auth option: SSH_AUTH_SOCK must be set and ssh-agent must have at least one loaded private key
- archive paths and runtime prerequisites
- preflight check: bootstrap kind cluster name is not already in use

Online checks:

- vSphere connectivity/authentication
- vSphere inventory lookups (datacenter, datastore, resource pool, networks)
- management requested-IP preflight for mgmtCluster.apiServer.address (fail if the address responds to ICMP or already listens on TCP/6443)
- proxy endpoint TCP reachability (host/IP + configured port)
- container registry reachability (docker.io/registry.k8s.io/quay.io/ghcr.io), warning-only for potential airgap installs
- warning-only IPAM collision scan against visible VM guest IPs

Skipped during online checks:

- vSphere folder existence
- VM template existence

Usage:

```
isoctl bootstrap validate [flags]
```

Flags:

Flag	Type	Description
`--address`	string	IP address to be used as the management cluster control plane VIP.
`--cilium-version`	string	Version of cilium to install on the management cluster.
`--config`, `-f`	string	Path to BootstrapConfig config file
`--config-dry-run`	bool	Output derived BootstrapConfig config and exit
`--controlplane-replicas`	int	The number of control plane replicas to provision for the management cluster.
`--create-authorized-key`	bool	Create a key-pair for management cluster node authorization if no keys are provided.
`--image-archives`	list	Specify additional container image archives to be loaded onto the management cluster nodes.
`--ipam-addresses`	list	IPAM pool addresses
`--ipam-gateway`	string	**Deprecated**: IPAM pool gateway
`--ipam-nameservers`	list	**Deprecated**: IPAM pool nameservers
`--ipam-prefix`	uint8	**Deprecated**: IPAM pool prefix
`--ipampool-name`	string	**Deprecated**: IPAM pool name
`--manifest`, `-m`	string	Specify a MgmtCluster manifest to deploy.
`--name`	string	Name of the management cluster to be created.
`--namespace`	string	Namespace to create the MgmtCluster in. By default, it's the name of the management cluster.
`--node-authorized-keys`	list	Specify additional public keys to be authorized for SSH on the management cluster nodes.

Flag	Type	Description
`--node-authorized-keys-from-agent`	bool	Add management cluster node authorized keys from ssh-agent. $SSH_AUTH_SOCK will be used as the socket address.
`--node-user`	string	Name of the default user on all provisioned nodes. By default, it will be “ilb”.
`--operator-image`	string	Override the isovalent-operator image URI to be used.
`--pod-cidr`	string	IP address range to be allocated for Pods.
`--seed-concurrency`	int	Number of images to push to the registry in parallel during seeding.
`--seed-from-local-image-archive`	list	Specify additional container image archives to seed the management cluster registry with.
`--service-cidr`	string	IP address range to be allocated for Services.
`--skip-cleanup`	bool	Skip cleanup of temporary resources after bootstrap is completed.
`--strict-config`	bool	Enable strict bootstrap config parsing (reject unknown fields)
`--validation-mode`	string	Validation mode: offline, online, or all
`--validation-trace`	bool	Enable detailed validation trace logs (for deep troubleshooting)
`--vsphere-datacenter`	string	Specify the datacenter node VMs should provision to.
`--vsphere-datastore`	string	Specify the datastore node VMs should allocate disks from.
`--vsphere-endpoint`	string	Endpoint for the target vSphere server.
`--vsphere-folder`	string	Configure the inventory folder to organize node VMs under.
`--vsphere-network`	string	Specify the default network for node VMs.
`--vsphere-password`	string	Password to use for authentication.
`--vsphere-resource-pool`	string	Specify the name of the resource pool node VMs should be scheduled to.
`--vsphere-template`	string	Provide a path to save the node OVA to. On bootstrap the CLI will upload the embedded node OVA to this path. All node VMs will be provisioned from this image.
`--vsphere-thumbprint`	string	Thumbprint of the vSphere server.
`--vsphere-username`	string	Username of the vSphere user that the control plane should assume the identity of.
`--worker-pool-name`	string	Name of the management cluster worker node pool.
`--worker-replicas`	int	The number of worker replicas to provision for the management cluster.

Examples:

```
# Validate all checks (default)
isoctl bootstrap validate -f bootstrap.yaml

# Validate only local/offline checks
isoctl bootstrap validate -f bootstrap.yaml --validation-mode offline

# Validate only provider connectivity/resource checks
isoctl bootstrap validate -f bootstrap.yaml --validation-mode online
```

`charts extract`

Extract a Helm chart from the cluster

Usage:

```
isoctl charts extract [CHART_NAME...] [flags]
```

Flags:

Flag	Type	Description
`--all`	bool	Extract all available charts
`--manifest`	string	Path to custom manifest file (defaults to embedded manifest)
`--output`, `-o`	string	Output directory for extracted charts

Examples:

```
# Extract a single chart
isoctl charts extract harbor-1.17.2 -o /tmp/

# Extract multiple charts
isoctl charts extract harbor-1.17.2 cilium-1.18.10 infra-0.1.1 -o /tmp/

# Extract all charts
isoctl charts extract --all -o /tmp/

# Extract from custom manifest
isoctl charts extract harbor-1.17.2 --manifest /path/to/install.yaml -o /tmp/
```

`charts list`

List available Helm charts in the cluster

Usage:

```
isoctl charts list [flags]
```

Flags:

Flag	Type	Description
`--manifest`	string	Path to custom manifest file (defaults to embedded manifest)

`hypershield dev create`

Create a kind cluster wired for Hypershield and deploy Hypershield into it.

This mirrors a BYOVM install with no vSphere or Cluster API: it creates the kind cluster (default CNI and kube-proxy disabled) with the caller's Docker registry credentials mounted so the kubelet can pull enterprise images, applies the operator bundle, installs the Gateway API CRDs and enterprise Cilium, then runs the operator in-cluster and creates a self-targeting Hypershield CR.

With `--mode` leader the Hypershield CR is deployed in Timescape Leader mode with the web UI, Public API, a bespoke in-cluster Dex OIDC provider, and a host-exposed gateway. The UI and authenticated Public API are both reachable from the host.

With `--mode` follower the cluster joins an existing leader. First register the follower on the leader UI and copy back the join credential, then pass it here with `--credential` along with `--leader` (the leader kind cluster name). The cluster ID is read from the credential, and the leader address, CA, and the follower's k8s host are all auto-discovered. Multiple followers can join one leader; each is its own kind cluster. The follower name (`--follower-name`, or `--name`) must match the name registered on the leader.

Usage:

```
isoctl hypershield dev create [flags]
```

Flags:

Flag	Type	Description
`--cilium-version`	string	Cilium chart version (selects the cilium-<version> ConfigMap shipped in the operator bundle)
`--credential`	string	Follower join credential copied from the leader UI (required with `--mode` follower)
`--follower-name`	string	Follower name; MUST match the name registered on the leader (Follower mode; defaults to `--name`)
`--leader`	string	Name of the leader kind cluster; auto-discovers the leader address and CA (required with `--mode` follower)
`--load-operator-image`	bool	Load `--operator-image` from the local Docker daemon into the kind nodes (needed for locally-built images the kubelet cannot pull, e.g. an arm64 build on Apple Silicon)
`--mode`	string	Timescape deployment mode: leader or follower
`--name`	string	Name of the kind cluster to create
`--node-image`	string	kindest/node image to use
`--operator-image`	string	Override the operator manager image (e.g. a locally-built image for the host architecture)
`--public-api-hostname`	string	DNS hostname for the Timescape Public API Gateway listener (Leader mode; must differ from the server and UI hostnames; defaults to api.127-0-0-1.sslip.io)
`--registries`	list	Registries to resolve credentials for (default: all logged-in registries)

Flag	Type	Description
`--server-hostname`	string	DNS hostname for the Timescape server gRPC listener (Leader mode; defaults to a dev placeholder)
`--state-dir`	string	Directory for generated files (default: $XDG_DATA_HOME/isoctl/hypershield-dev or ~/.local/share/isoctl/hypershield-dev)
`--ui-hostname`	string	DNS hostname for the Timescape UI Gateway listener (Leader mode; must differ from `--server-hostname`; defaults to a sslip.io dev name)
`--workers`	int	Number of worker nodes (minimum 1 for L2 announcements)

Examples:

```
# Leader dev cluster (UI, Public API, Dex, and gateway are always on)
isoctl hypershield dev create --name hs-leader --mode leader

# Follower joined to the hs-leader cluster (credential copied from the leader UI)
isoctl hypershield dev create --name hs-follower-1 --mode follower \
  --leader hs-leader --credential "$CREDENTIAL"
```

`hypershield dev delete`

Delete the kind cluster created by ‘isoctl hypershield dev create’ and remove its per-cluster state directory (the resolved registry credentials, kubeconfig, and any leader/follower credential material such as the join credential and the Dex/leader CAs).

This is idempotent: deleting a cluster that does not exist is not an error.

Usage:

```
isoctl hypershield dev delete [flags]
```

Flags:

Flag	Type	Description
`--name`	string	Name of the kind cluster to delete
`--state-dir`	string	Directory holding generated files (default: $XDG_DATA_HOME/isoctl/hypershield-dev or ~/.local/share/isoctl/hypershield-dev)

Examples:

```
# Delete the default dev cluster
isoctl hypershield dev delete

# Delete a named cluster
isoctl hypershield dev delete --name hs-leader
```

`hypershield install`

Create or update the Hypershield CR and wait for all components to be ready.

Deployment modes:

CAPI (default):

- Resolves the MgmtCluster from the current kubeconfig context.
- Creates a cluster-scoped Hypershield CR using the MgmtCluster namespace as its management namespace.
- Uses the MgmtCluster's admin kubeconfig to deploy components.
- Uses vsphere-csi as the Timescape PVC storage class.

BYOVM:

- Select this mode with `--byovm` for an existing cluster not provisioned through CAPI.
- Requires `--admin-kubeconfig-secret` to identify an admin kubeconfig Secret.
- Creates a cluster-scoped Hypershield CR using isovalent-operator-system as its management namespace.
- Uses longhorn as the Timescape PVC storage class.

Components installed:

- ClickHouse operator
- CloudNativePG operator
- Control plane (CRDs and SmartSwitch configuration)
- Hubble Timescape (configure UI and Public API exposure via `-f` or `--timescape-*` flags)

The complete configuration is validated before any cluster resources are changed.

Usage:

```
isoctl hypershield install [flags]
```

Flags:

Flag	Type	Description
`--admin-kubeconfig-secret`	string	Name of the Secret in isovalent-operator-system containing an admin kubeconfig (required with `--byovm`)
`--byovm`	bool	Deploy on a pre-existing cluster not provisioned via CAPI (brownfield / BYOVM mode)
`--clc-enabled`	bool	Enable the Connection Logs Connector (receives SmartSwitch IPFIX and pushes ConnectionLogs to Timescape)
`--clc-ip`	string	External IP for the CLC IPFIX UDP LoadBalancer Service (required when CLC is enabled)
`--config`, `-f`	string	Path to HypershieldInstallConfig config file
`--config-dry-run`	bool	Output derived HypershieldInstallConfig config and exit
`--context`	string	Kubeconfig context name to use
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--strict-config`	bool	Enable strict Hypershield config parsing (reject unknown fields)
`--timeout`	duration	Maximum time to wait for Hypershield readiness
`--timescape-follower-ca-file`	string	Path to the Leader CA certificate PEM file (follower profile)
`--timescape-follower-cluster-name`	string	Human-readable name of this follower cluster (follower profile)
`--timescape-follower-credential-file`	string	Path to a file containing the opaque join credential from the Leader's GenerateJoinCredentials (follower profile)
`--timescape-follower-k8s-host`	string	Follower Kubernetes API server URL, e.g. https://192.168.1.10:6443 (follower profile)
`--timescape-follower-leader-address`	string	Leader Timescape gRPC address in host:port format (follower profile)
`--timescape-ingester-ip`	string	External LoadBalancer IP for SmartSwitch reporting to Timescape on HTTPS port 4260 and gRPC port 4261 (required in the follower profile)
`--timescape-ip`	string	External IP for the Gateway LoadBalancer (shared by gRPC server, Public API, and UI in the leader profile)
`--timescape-override`	stringArray	Helm `--set` style override for Timescape (repeatable, e.g. `--timescape-override` key=value)
`--timescape-profile`	string	Timescape role: leader or follower (required)
`--timescape-public-api-enabled`	bool	Enable the Timescape Public API (leader profile)
`--timescape-public-api-hostname`	string	DNS hostname for the Timescape Public API Gateway listener (leader profile)
`--timescape-sc-ca-file`	string	Path to the Leader Secure Connector server CA certificate PEM file (follower profile, required)
`--timescape-sc-hostname`	string	Optional DNS name for the Secure Connector server, added to its certificate SANs (leader profile)
`--timescape-sc-ip`	string	External LoadBalancer IP for the Secure Connector server gRPC endpoint (leader profile, required)
`--timescape-sc-server-address`	string	Leader Secure Connector server gRPC address in host:port format (follower profile, required)
`--timescape-scc-enabled`	bool	Enable production SCC integration (leader profile); requires `--timescape-scc-tenant-id`
`--timescape-scc-staging-enabled`	bool	Enable staging SCC integration (Cisco staging IdPs, QA/dev); requires `--timescape-scc-staging-tenant-id`

Flag	Type	Description
`--timescape-scc-staging-tenant-id`	string	Staging SCC tenant/product ID (required with `--timescape-scc-staging-enabled`)
`--timescape-scc-tenant-id`	string	Production SCC tenant/product ID pinned in the SCC token validation rules (required with `--timescape-scc-enabled`)
`--timescape-server-hostname`	string	DNS hostname for the Timescape gRPC server Gateway listener (leader profile)
`--timescape-server-rbac-oidc-ca-file`	string	Path to the OIDC provider CA certificate PEM file; isoctl creates a ConfigMap from it (leader profile, optional for public IdPs)
`--timescape-server-rbac-oidc-client-id`	string	OIDC client ID for Timescape server RBAC (leader profile)
`--timescape-server-rbac-oidc-issuer-url`	string	OIDC issuer base URL for Timescape server RBAC (leader profile)
`--timescape-ui-auth-oidc-ca-configmap-name`	string	Name of an existing ConfigMap in hubble-timescape containing the OIDC provider CA under key ca.crt (optional)
`--timescape-ui-auth-oidc-client-id`	string	Override the OIDC client ID for the UI (optional; defaults to server.rbac.oidcClientID via chart fallback)
`--timescape-ui-auth-oidc-client-secret-file`	string	Path to a file containing the OIDC application client secret for the UI; isoctl creates a Secret in the operator namespace from it (required in the leader profile when UI is enabled)
`--timescape-ui-auth-oidc-client-secret-name`	string	Name of the K8s Secret created in the operator namespace for the UI OIDC client secret
`--timescape-ui-auth-oidc-issuer-url`	string	Override the base OIDC issuer URL for the UI (optional; defaults to server.rbac.oidcIssuerURL via chart fallback)
`--timescape-ui-enabled`	bool	Enable the Timescape web UI
`--timescape-ui-hostname`	string	DNS hostname for the Timescape UI Gateway listener
`--timescape-ui-tls-cert-file`	string	Path to the server certificate PEM file for the UI TLS secret
`--timescape-ui-tls-key-file`	string	Path to the private key PEM file for the UI TLS secret
`--timescape-ui-tls-secret-name`	string	Name of the K8s TLS secret created in hubble-timescape namespace

Examples:

```
# Install on a CAPI-provisioned cluster
isoctl hypershield install

# Install on a pre-existing cluster (BYOVM / brownfield)
isoctl hypershield install --byovm --admin-kubeconfig-secret my-admin-kubeconfig

# Install with Timescape UI (operator manages TLS automatically)
isoctl hypershield install \
  --timescape-ui-enabled \
  --timescape-ip 10.0.0.100 \
  --timescape-ui-hostname timescape.example.com

# Install with Timescape UI using user-provided TLS certificates
isoctl hypershield install \
  --timescape-ui-enabled \
  --timescape-ip 10.0.0.100 \
  --timescape-ui-hostname timescape.example.com \
  --timescape-ui-tls-cert-file /path/to/tls.crt \
  --timescape-ui-tls-key-file /path/to/tls.key

# Install in the leader profile without initial identity bindings
isoctl hypershield install \
  --timescape-profile leader \
  --timescape-ip 10.0.0.200 \
  --timescape-sc-ip 10.0.0.201 \
  --timescape-server-hostname timescape-grpc.example.com \
  --timescape-public-api-enabled \
  --timescape-public-api-hostname api.example.com \
  --timescape-server-rbac-oidc-issuer-url https://my-account.okta.com/oauth2/default \
  --timescape-server-rbac-oidc-client-id 0oabc123def456 \
  --timescape-server-rbac-oidc-ca-file /path/to/oidc-ca.crt

# Install in the leader profile using a config file (see example.hypershield-install.yaml)
isoctl hypershield install -f hypershield-install.yaml
```

`hypershield upgrade`

Validate and apply a supported Hypershield release artifact.

The install path may be an upgrade .tar.zst bundle or the expanded directory installed by an isovalent-upgrade-hypershield Debian/RPM package. Installing the package only stages files; it does not change the cluster.

One immutable release contains the complete infrastructure and application payload. It may change either or both. The command executes their ordered internal stages and records the release only after the complete target converges.

Phase 1 supports CAPI-managed Hypershield installations.

Usage:

```
isoctl hypershield upgrade [flags]
```

Flags:

Flag	Type	Description
`--context`	string	Kubeconfig context name to use
`--install-path`, `-f`	string	Upgrade bundle or package-installed directory
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--ova-template`	string	Existing customer-imported vSphere VM template for a release node OVA
`--plan-only`	bool	Validate and display the upgrade plan without changing the cluster
`--seed-concurrency`	int	Number of images to push in parallel
`--wait-timeout`	duration	Maximum time to wait for target release convergence

`hypershield validate`

Validate a Hypershield installation config without changing cluster resources.

Validation checks:

- unknown or obsolete configuration fields
- a required Timescape leader or follower profile
- required leader and follower settings
- Timescape UI and Public API exposure and OIDC consistency
- referenced credential, certificate, and client-secret files
- optional component requirements

Usage:

```
isoctl hypershield validate [flags]
```

Flags:

Flag	Type	Description
`--clc-enabled`	bool	Enable the Connection Logs Connector (receives SmartSwitch IPFIX and pushes ConnectionLogs to Timescape)
`--clc-ip`	string	External IP for the CLC IPFIX UDP LoadBalancer Service (required when CLC is enabled)
`--config`, `-f`	string	Path to HypershieldInstallConfig config file
`--config-dry-run`	bool	Output derived HypershieldInstallConfig config and exit
`--strict-config`	bool	Enable strict Hypershield config parsing (reject unknown fields)
`--timescape-follower-ca-file`	string	Path to the Leader CA certificate PEM file (follower profile)
`--timescape-follower-cluster-name`	string	Human-readable name of this follower cluster (follower profile)

Flag	Type	Description
`--timescape-follower-credentials-file`	string	Path to a file containing the opaque join credential from the Leader's GenerateJoinCredentials (follower profile)
`--timescape-follower-k8s-host`	string	Follower Kubernetes API server URL, e.g. https://192.168.1.10:6443 (follower profile)
`--timescape-follower-leader-address`	string	Leader Timescape gRPC address in host:port format (follower profile)
`--timescape-ingester-ip`	string	External LoadBalancer IP for SmartSwitch reporting to Timescape on HTTPS port 4260 and gRPC port 4261 (required in the follower profile)
`--timescape-ip`	string	External IP for the Gateway LoadBalancer (shared by gRPC server, Public API, and UI in the leader profile)
`--timescape-override`	stringArray	Helm `--set` style override for Timescape (repeatable, e.g. `--timescape-override key=value`)
`--timescape-profile`	string	Timescape role: leader or follower (required)
`--timescape-public-api-enabled`	bool	Enable the Timescape Public API (leader profile)
`--timescape-public-api-hostname`	string	DNS hostname for the Timescape Public API Gateway listener (leader profile)
`--timescape-sc-ca-file`	string	Path to the Leader Secure Connector server CA certificate PEM file (follower profile, required)
`--timescape-sc-hostname`	string	Optional DNS name for the Secure Connector server, added to its certificate SANs (leader profile)
`--timescape-sc-ip`	string	External LoadBalancer IP for the Secure Connector server gRPC endpoint (leader profile, required)
`--timescape-sc-server-address`	string	Leader Secure Connector server gRPC address in host:port format (follower profile, required)
`--timescape-scc-enabled`	bool	Enable production SCC integration (leader profile); requires `--timescape-scc-tenant-id`
`--timescape-scc-staging-enabled`	bool	Enable staging SCC integration (Cisco staging IdPs, QA/dev); requires `--timescape-scc-staging-tenant-id`
`--timescape-scc-staging-tenant-id`	string	Staging SCC tenant/product ID (required with `--timescape-scc-staging-enabled`)
`--timescape-scc-tenant-id`	string	Production SCC tenant/product ID pinned in the SCC token validation rules (required with `--timescape-scc-enabled`)
`--timescape-server-hostname`	string	DNS hostname for the Timescape gRPC server Gateway listener (leader profile)
`--timescape-server-rbac-oidc-ca-file`	string	Path to the OIDC provider CA certificate PEM file; isoctl creates a ConfigMap from it (leader profile, optional for public IdPs)
`--timescape-server-rbac-oidc-client-id`	string	OIDC client ID for Timescape server RBAC (leader profile)
`--timescape-server-rbac-oidc-issuer-url`	string	OIDC issuer base URL for Timescape server RBAC (leader profile)
`--timescape-ui-auth-oidc-ca-configmap-name`	string	Name of an existing ConfigMap in hubble-timescape containing the OIDC provider CA under key ca.crt (optional)
`--timescape-ui-auth-oidc-client-id`	string	Override the OIDC client ID for the UI (optional; defaults to server.rbac.oidcClientID via chart fallback)
`--timescape-ui-auth-oidc-client-secret-file`	string	Path to a file containing the OIDC application client secret for the UI; isoctl creates a Secret in the operator namespace from it (required in the leader profile when UI is enabled)
`--timescape-ui-auth-oidc-client-secret-name`	string	Name of the K8s Secret created in the operator namespace for the UI OIDC client secret

Flag	Type	Description
`--timescape-ui-auth-oidc-issuer-url`	string	Override the base OIDC issuer URL for the UI (optional; defaults to server.rbac.oidcIssuerURL via chart fallback)
`--timescape-ui-enabled`	bool	Enable the Timescape web UI
`--timescape-ui-hostname`	string	DNS hostname for the Timescape UI Gateway listener
`--timescape-ui-tls-cert-file`	string	Path to the server certificate PEM file for the UI TLS secret
`--timescape-ui-tls-key-file`	string	Path to the private key PEM file for the UI TLS secret
`--timescape-ui-tls-secret-name`	string	Name of the K8s TLS secret created in hubble-timescape namespace

Examples:

```
# Validate a configuration file
isoctl hypershield validate -f hypershield-install.yaml

# Validate the equivalent CLI arguments before installing
isoctl hypershield validate \
  --timescape-profile leader \
  --timescape-ip 10.0.0.200 \
  --timescape-server-hostname timescape-grpc.example.com \
  --timescape-server-rbac-oidc-issuer-url https://idp.example.com \
  --timescape-server-rbac-oidc-client-id my-client-id
```

`hypervisor-agnostic validate`

Validate a hypervisor-agnostic Hypershield cluster config with offline and/or online checks.

Offline checks:

- required fields (podCIDR, serviceCIDR, controlPlaneReplicas, workerReplicas, apiServer.address)
- network names, IPAM addresses, gateway, and nameservers
- no overlap between networks[*].ipam.addresses and pod/service CIDRs
- network IPAM address capacity against required node replicas
- proxy settings and noProxy value format
- nodeAuthorizedKeys format when provided
- controller and worker IP resolution from networks[0].ipam.addresses

Online checks:

- proxy endpoint TCP reachability (host/IP + configured port)

Usage:

```
isoctl hypervisor-agnostic validate [flags]
```

Flags:

Flag	Type	Description
`--config`, `-f`	string	Path to hypervisor-agnostic cluster config file
`--strict-config`	bool	Enable strict config parsing (reject unknown fields)
`--validation-mode`	string	Validation mode: offline, online, or all
`--validation-trace`	bool	Enable detailed validation trace logs (for deep troubleshooting)

Examples:

```
# Validate all checks (default)
isoctl hypervisor-agnostic validate -f cluster-config.yaml

# Validate only local/offline checks
isoctl hypervisor-agnostic validate -f cluster-config.yaml --validation-mode offline

# Validate only network reachability checks
isoctl hypervisor-agnostic validate -f cluster-config.yaml --validation-mode online
```

`ilb accesslog`

Display load balancer T2 access log.

Usage:

```
isoctl ilb accesslog
```

`ilb applicationlog`

Display load balancer T2 application log.

Usage:

```
isoctl ilb applicationlog
```

ilb delete

Delete a load balancer by name.

Usage:

```
isoctl ilb delete NAME [flags]
```

Flags:

Flag	Type	Description
`--context`	string	Kubeconfig context name to use
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--name`	string	Name of the LoadBalancer
`--namespace`, `-n`	string	Kubernetes namespace
`--wait`	bool	Wait for load balancer to be deleted

Examples:

```
  # Delete LB and wait for completion
  isoctl ilb delete my-loadbalancer --wait

# Delete LB cluster and don't wait
  isoctl ilb delete my-loadbalancer
```

ilb install

Install a load balancer.

Usage:

```
isoctl ilb install [flags]
```

Flags:

Flag	Type	Description
`--address`	string	IP address to be used as the load balancer cluster control plane VIP.
`--aux-nodes`	int	The number of AUX replicas to provision for the load balancer.
`--cilium-version`	string	Version of cilium to install on the load balancer cluster.

Flag	Type	Description
`--context`	string	Kubeconfig context name to use
`--controlplane-replicas`	int	The number of control plane replicas to provision for the load balancer cluster.
`--create-authorized-key`	bool	Create a key-pair for load balancer node authorization if no keys are provided.
`--create-options`, `-f`	string	Path to CreateOptions config file
`--create-options-dry-run`	bool	Output derived CreateOptions config and exit
`--interactive`, `-i`	bool	Enabled interactive mode
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--manifest`, `-m`	string	Specify a LoadBalancer manifest to deploy.
`--name`	string	Name of the load balancer to be created.
`--namespace`	string	Namespace to create the LoadBalancer in. By default, it's the name of the load balancer.
`--node-authorized-keys`	list	Specify additional public keys to be authorized for SSH on the load balancer cluster nodes.
`--node-authorized-keys-from-agent`	bool	Add load balancer cluster node authorized keys from ssh-agent. $SSH_AUTH_SOCK will be used as the socket address.
`--node-user`	string	Name of the default user on all provisioned nodes. By default, it will be "ilb".
`--t1`	int	The number of T1 replicas to provision for the load balancer.
`--t2`	int	The number of T2 replicas to provision for the load balancer.

Examples:

```
# Install a basic load balancer from args using SSH_AGENT for node auth
   isoctl ilb install --name lb --address 10.53.53.249 --t1 2 --t2 3 --node-authorized-keys-from-agent

   # Install a load balancer from a MgmtCluster manifest
   isoctl ilb install -m loadbalancer.yaml

   # Install a load balancer from a config file
   isoctl ilb install -f config.yaml
```

`ilb kubeconfig`

Get the kubeconfig for a load balancer by name.

Usage:

```
isoctl ilb kubeconfig NAME [flags]
```

Flags:

Flag	Type	Description
`--context`	string	Kubeconfig context name to use
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--name`	string	Name of the LoadBalancer
`--namespace`, `-n`	string	Kubernetes namespace

Examples:

```
# Get the kubeconfig for a load balancer
isoctl ilb kubeconfig my-loadbalancer
```

`ilb service`

Display load balancer service status.

Usage:

```
isoctl ilb service
```

`ilb status`

Display load balancer status.

Usage:

```
isoctl ilb status
```

`ilb switch`

Switch between cluster contexts.

Usage:

```
isoctl ilb switch [NAME] [flags]
```

Flags:

Flag	Type	Description
`--context`	string	Kubeconfig context name to use
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--mgmt`	bool	switch to the management cluster
`--name`	string	Name of the CiliumCluster
`--namespace`, `-n`	string	Kubernetes namespace

Examples:

```
# Switch between contexts interactively
isoctl switch

# Switch to context by cluster name
isoctl switch my-loadbalancer

# Swicth to management cluster context
isoctl switch --mgmt
```

`ilb sysdump`

Collect information required to troubleshoot issues with Cilium and Hubble.

Usage:

```
isoctl ilb sysdump
```

`ilb test`

Run load balancer tests.

Usage:

```
isoctl ilb test
```

`ilb upgrade`

Upgrade the ILB cluster using the given install artifact tarball.

Usage:

```
isoctl ilb upgrade [flags]
```

Flags:

Flag	Type	Description
`--context`	string	Kubeconfig context name to use
`--install-path`, `-f`	string	Path to the install artifact tarball
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--seed-concurrency`	int	Number of images to push in parallel
`--wait-timeout`	duration	How long to wait for the cluster to become healthy after upgrade before rolling back

`images load`

Load container images from the given local image archive to the management cluster's registry, or to an explicitly provided registry.

Usage:

```
isoctl images load ARCHIVE [flags]
```

Flags:

Flag	Type	Description
`--context`	string	Kubeconfig context name to use
`--insecure-skip-tls-verify`	bool	Skip TLS verification for the explicitly provided registry
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--password`	string	Password for the explicitly provided registry
`--registry`	string	Registry address to load images into. When set, MgmtCluster registry discovery is skipped
`--seed-concurrency`	int	Number of images to push in parallel
`--username`	string	Username for the explicitly provided registry

Examples:

```
  # Load a local zst compressed image archive
isoctl images load my-images.tar.zst

# Load a local zst compressed image archive to an explicit registry
isoctl images load my-images.tar.zst --registry registry.example.com

# Load to an explicit registry with a self-signed certificate
isoctl images load my-images.tar.zst --registry registry.example.com --insecure-skip-tls-verify

# Load to an explicit registry with basic auth
isoctl images load my-images.tar.zst --registry registry.example.com --username myuser --password
mypass
```

`kubeconfig`

Get the kubeconfig for all clusters

Usage:

```
isoctl kubeconfig [flags]
```

Flags:

Flag	Type	Description
`--context`	string	Kubeconfig context name to use
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--update`	bool	Update the sourced kubeconfig file

`mgmtcluster delete`

Delete a management cluster.

The nuke strategy is intended for repeated CI/test environments where speed and full teardown are prioritized over graceful uninstall semantics.

Usage:

```
isoctl mgmtcluster delete [flags]
```

Flags:

Flag	Type	Description
`--context`	string	Kubeconfig context name to use
`--force`	bool	Skip interactive confirmation
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--strategy`	string	Delete strategy (currently supported: nuke)
`--timeout`	duration	Maximum time to complete deletion
`--vsphere-folder`	string	Optional vSphere inventory folder path used to enumerate VMs when VSphereMachine discovery is empty

`operator install`

Install the Isovalent Operator into an existing Kubernetes cluster (BYOVM).

The command applies the embedded operator bundle (CRDs, RBAC, Deployment, and related resources from install.yaml), then waits until the controller-manager Deployment is available.

The operator Deployment's manager container image comes from the embedded manifest (for example quay.io/isovalent-dev/isovalent-operator-ci:latest). Cluster nodes must be configured to pull from your registry when you are not using a public registry (registry mirrors, TLS trust, and credentials as required by your environment). You do not need to pass a registry host or Harbor project path in this command when node configuration already resolves the embedded image reference.

Use `--operator-image` only when you want to replace the manager container image with a different reference (for example <registry>/<project>/<image>:<tag>).

Flags:

- `-k`, `--kubeconfig`: Path to kubeconfig (default: standard kubeconfig path)
- `--context`: Kubeconfig context (default: mgmtcluster.isovalent.com)
- `--timeout`: Max wait for operator readiness (default: 10m)
- `--operator-image`: Override only the manager container image on the operator Deployment

Usage:

```
isoctl operator install [flags]
```

Flags:

Flag	Type	Description
`--context`	string	Kubeconfig context name to use

Flag	Type	Description
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--operator-image`	string	Image reference for the manager container (replaces the image in the embedded manifest)
`--timeout`	duration	Maximum time to wait for operator readiness

Examples:

```
# Install using current kubeconfig defaults
isoctl operator install

# Install with explicit kubeconfig and context (typical BYOVM admin cluster)
isoctl operator install --kubeconfig ~/.kube/config --context admin

# Install with a longer readiness wait
isoctl operator install --timeout 15m

# Optionally replace the manager container image
isoctl operator install --operator-image isovalent-dev/isovalent-operator-ci:latest
```

`switch`

Switch between cluster contexts.

Usage:

```
isoctl switch [NAME] [flags]
```

Flags:

Flag	Type	Description
`--context`	string	Kubeconfig context name to use
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--mgmt`	bool	switch to the management cluster
`--name`	string	Name of the CiliumCluster
`--namespace`, `-n`	string	Kubernetes namespace

Examples:

```
# Switch between contexts interactively
isoctl switch

# Switch to context by cluster name
isoctl switch my-loadbalancer

# Swicth to management cluster context
isoctl switch --mgmt
```

`sysdump`

Collect a sysdump from the cluster containing logs and diagnostics information for troubleshooting

Usage:

```
isoctl sysdump [flags]
```

Flags:

Flag	Type	Description
`--cilium-bugtool-flags`	stringArray	Optional set of flags to pass to cilium-bugtool command.
`--cilium-daemon-set-label-selector`	string	The labels used to target Cilium daemon set
`--cilium-envoy-label-selector`	string	The labels used to target Cilium Envoy pods
`--cilium-helm-release-name`	string	The Cilium Helm release name for which to get values. If not provided then the `--helm-release-name` global flag is used (if provided)
`--cilium-label-selector`	string	The labels used to target Cilium pods
`--cilium-namespace`	string	The namespace Cilium is running in. If not provided then the `--namespace` global flag is used (if provided)
`--cilium-node-init-selector`	string	The labels used to target Cilium node init pods
`--cilium-operator-label-selector`	string	The labels used to target Cilium operator pods
`--cilium-operator-namespace`	string	The namespace Cilium operator is running in. If not provided then the `--namespace` global flag is used (if provided)
`--cilium-spire-agent-selector`	string	The labels used to target Cilium spire-agent pods
`--cilium-spire-namespace`	string	The namespace Cilium SPIRE installation is running in
`--cilium-spire-server-selector`	string	The labels used to target Cilium spire-server pods
`--clustermesh-apiserver-label-selector`	string	The labels used to target ‘clustermesh-apiserver’ pods
`--cni-config-directory`	string	Directory where CNI configs are located
`--cni-configmap-name`	string	The name of the CNI config map
`--collect-logs-from-not-ready-agents`	bool	Whether to collect logs from not ready Cilium agent pods

Flag	Type	Description
`--context`	string	Kubeconfig context name to use
`--copy-retry-limit`	int	Retry limit for file copying operations. If set to -1, copying will be retried indefinitely. Useful for collecting sysdump while on unreliable connection.
`--debug`	bool	Whether to enable debug logging
`--detect-gops-pid`	bool	Whether to automatically detect the gops agent PID.
`--extra-label-selectors`	stringArray	Optional set of labels selectors used to target additional pods for log collection.
`--hubble-flows-count`	int64	Number of Hubble flows to collect. Setting to zero disables collecting Hubble flows.
`--hubble-flows-timeout`	duration	Timeout for collecting Hubble flows
`--hubble-generate-certs-labels`	string	The labels used to target Hubble UI pods
`--hubble-label-selector`	string	The labels used to target Hubble pods
`--hubble-relay-labels`	string	The labels used to target Hubble Relay pods
`--hubble-ui-labels`	string	The labels used to target Hubble UI pods
`--hypershield`	bool	Collect logs and resources from Hypershield and ClickHouse Operator namespaces
`--kubeconfig`, `-k`	string	Path to the mgmt kubeconfig to use
`--logs-limit-bytes`	int64	The limit on the number of bytes to retrieve when collecting logs
`--logs-since-time`	duration	How far back in time to go when collecting logs
`--node-list`	string	Comma-separated list of node IPs or names to filter pods for which to collect gops and logs
`--output-filename`	string	The name of the resulting file (without extension) '<ts>' can be used as the placeholder for the timestamp
`--profiling`	bool	Whether to enable scraping profiling data
`--quick`	bool	Whether to enable quick mode (i.e. skip collection of 'cilium-bugtool' output and logs)
`--tetragon-helm-release-name`	string	The Tetragon Helm release name for which to get values.
`--tetragon-label-selector`	string	The labels used to target Tetragon pods
`--tetragon-namespace`	string	The namespace Tetragon is running in
`--tetragon-operator-label-selector`	string	The labels used to target Tetragon operator pods
`--tracing`	bool	Whether to enable scraping tracing data
`--worker-count`	int	The number of workers to use NOTE: There is a lower bound requirement on the number of workers for the sysdump operation to be effective. Therefore, for low values, the actual number of workers may be adjusted upwards. Defaults to the number of available CPUs.

`version`

Print version.

Usage:

```
isoctl version
```

Timescape API v1

The Timescape public REST API provides supported automation for Hypershield network object groups and policies. It also exposes Timescape build information. All version 1 paths are relative to `/api/hs/v1` on the deployment-specific Timescape API origin.

This reference describes the behavior shared by the endpoints and the endpoint-specific contract for Timescape 1.20. Use the task-oriented Hypershield documentation for end-to-end operational workflows.

You can view the OpenAPI document, download its YAML source, or retrieve the JSON document bundled with a running Timescape deployment.

Requests and responses

Build each request URL from the Timescape API origin supplied by your administrator, the `/api/hs/v1` base path, and an endpoint path. For example, the build information path is `/api/hs/v1/meta/build-info`.

Media types

`POST` and `PUT` request bodies use `application/json`. Successful `GET`, `POST`, and `PUT` operations return `application/json`. Successful `DELETE` operations return `204 No Content` without a response body.

Most errors use `application/problem+json`. The `500 Internal Server Error` response has a distinct `application/json` envelope. See Errors and status codes for both formats.

Resource identity and revisions

A network object group or policy is identified by its Kubernetes `namespace` and resource `name`. Resource responses contain a `version` used for optimistic concurrency. They can also contain the `revisionTimestamp` for the returned revision.

List and get operations accept an optional `version` query parameter. Omit the parameter for the latest desired state. If you request `version=N`, the API returns desired state from the latest resource revision less than or equal to `N`. The returned resource `version` identifies the revision that was selected and can therefore be less than the requested value.

Filtering and pagination

List operations support these filters:

- `namespace` limits results to one namespace. If omitted, the operation is a cross-namespace list and requires access across namespaces. Timescape does not silently reduce an unfiltered list to individually authorized namespaces.
- `search` performs a case-insensitive substring search over the resource name and string fields.
- `labelSelector` filters resources by Kubernetes metadata labels.
- `version` selects a point-in-time resource version instead of the latest state.

Policy list operations also accept `includeStatus=true`. This parameter requires permission to read policy deployment status in addition to permission to list policies.

The default `pageSize` is 50 and the maximum is 1000. The service can return fewer results than requested. If a response contains `nextPageToken`, pass its value unchanged as `pageToken` in the next request. Pagination is complete when `nextPageToken` is absent. `totalCount` reports the total number of resources that match the

request, not only the current page size.

Create resources

A successful `POST` returns `201 Created`. The `Location` response header identifies the created resource, and the strong `ETag` response header contains its current resource version. Store the `ETag` if the client might replace the resource later.

Replace resources

`PUT` performs a full, idempotent replacement. Supply the complete desired configuration in the update request; omitted optional configuration is not preserved by a full replacement.

Every `PUT` requires one strong `If-Match` entity tag returned by an earlier `GET`, `POST`, or `PUT`. The API accepts one quoted value such as `"42"`. It rejects weak tags, wildcards, multiple tags, malformed values, and unquoted resource versions.

The API returns:

- `428 Precondition Required` when `If-Match` is missing
- `400 Bad Request` when the header is malformed or unsupported
- `412 Precondition Failed` when the supplied version is no longer current

After a `412` response, retrieve the latest resource, reconcile the intended replacement with that state, and submit the new `ETag`. Do not overwrite a concurrent update without reconciling it.

Delete resources

A successful `DELETE` returns `204 No Content`. A delete can return `409 Conflict` when the current resource state prevents deletion. Resolve the reported conflict before retrying the operation.

Retry requests safely

Use bounded retries for `408`, `429`, and transient server or network failures. Follow `Retry-After` after a `429` response. Add randomized delay when multiple clients can retry at the same time.

The request method and concurrency controls affect retry behavior:

- You can retry a `GET` without changing server state.
- If a `POST` result is uncertain, retrieve the named resource. If it exists, compare it with the intended resource and reconcile any differences. If it does not exist, retry only within the client's retry limit. Do not assume that `409 Conflict` proves that the first request succeeded.

- If a `PUT` result is uncertain, retrieve the resource. If it matches the intended replacement, treat the operation as successful. Otherwise, reconcile the intended replacement with the returned state and use its current `ETag` for any new update.
- If a `DELETE` result is uncertain, retrieve the named resource. Treat an absent resource as success. Do not retry the delete automatically if the resource exists; another client might have created a new resource with the same name. Delete it again only if the client can prove ownership, such as with an ownership label. Otherwise, report the uncertain outcome for reconciliation.

Desired state and deployment status

Policy mutation responses confirm that Timescape recorded the desired policy state. They do not confirm that every enforcement agent has applied that state.

Set `includeStatus=true` on a policy list or get request to include `deploymentStatus`. `overallStatus` can be `inSync`, `outOfSync`, `pending`, or `unknown`. The status also reports agent counts when those counts are available. Treat the resource configuration and revision as desired state, and treat `deploymentStatus` as observed deployment state.

`inSync` means that reporting enforcement agents are consistent with the highest policy version that any agent has reported. It does not compare that reported version with the desired resource `version` in this REST response. A successful mutation followed by `inSync` therefore does not, by itself, prove that the new desired revision is enforced.

If a request combines `version=N` and `includeStatus=true`, the resource configuration is historical desired state from the latest revision less than or equal to `N`, while `deploymentStatus` remains current observed state. Do not interpret the status as a historical observation from revision `N`.

Authentication and authorization

Timescape deployments can enable role-based access control (RBAC) for the public API. When RBAC is enabled, every operation declared in the OpenAPI document requires a JSON Web Token (JWT) that the deployment accepts. Obtain the API origin, token, and required access from your Timescape administrator. Token issuance and identity-provider configuration are deployment-specific.

`GET /api/hs/v1/openapi.json` is not a declared operation, and Timescape does not require a bearer token for it. A gateway or ingress can still require its own credentials.

Bearer authentication

Send the token with the HTTP `Bearer` authentication scheme:

```
Authorization: Bearer eyJhbGciOi...
```

Do not put tokens in URLs, request bodies, source files, or command history. Keep them in a protected credential store or short-lived environment variable, and rotate or revoke them according to your organization's policy.

When public API RBAC is disabled, the declared operations do not require a bearer token. Supplying a token to an RBAC-disabled deployment does not enable per-principal authorization. Confirm the deployment mode and access controls with your administrator instead of assuming that an endpoint is anonymous.

Authorization

Timescape authorizes each operation independently. A principal can have different access to list, get, create, replace, and delete resources. Access is also namespace-aware. A request scoped to one namespace requires access to that namespace. A list request that omits the `namespace` filter requires cross-namespace access; Timescape does not silently return a partial list.

Reading policy deployment status is a separate permission. A principal that can list or get policies might still receive `403 Forbidden` when requesting `includeStatus=true`.

The public API does not define stable built-in role names. Configure roles in the deployment and grant only the operations and namespaces required by the client.

Authentication failures

`401 Unauthorized` means authentication is required or the supplied credentials are invalid. Check that the `Authorization` header uses the `Bearer` scheme, and replace an expired, malformed, or revoked token.

`403 Forbidden` means Timescape authenticated the principal but the principal cannot perform the requested operation. Request the missing operation or namespace permission. Re-authenticating with the same identity does not resolve an authorization failure.

TLS and mutual TLS

Use HTTPS and validate the Timescape server certificate. Transport Layer Security (TLS) protects request credentials and data in transit.

A deployment can require mutual TLS (mTLS) at its gateway or ingress. mTLS authenticates the client connection, while a bearer token authenticates and authorizes the API principal inside Timescape. If both controls are configured, the client must satisfy both. A client certificate does not replace the bearer token unless the deployment explicitly disables public API RBAC.

Errors and status codes

Use the HTTP status code as the primary result of an API request. Do not parse human-readable error text to decide how a client behaves.

Success status codes

Status	Meaning
`200 OK`	A get, list, or replacement operation succeeded.
`201 Created`	A resource was created. Use the `Location` and `ETag` response headers for subsequent operations.
`204 No Content`	A resource was deleted. The response has no body.

Error status codes

Status	Meaning and response
`400 Bad Request`	The request is malformed or contains an invalid parameter, body, or precondition value. Correct the request before retrying.
`401 Unauthorized`	Authentication is required or the credentials are invalid. Supply or replace the bearer token.
`403 Forbidden`	The authenticated principal lacks access to the operation, namespace, or requested policy status.
`404 Not Found`	The named resource was not found.
`408 Request Timeout`	The server timed out while waiting for the request. Confirm the result before you retry the request safely.
`409 Conflict`	The requested mutation conflicts with the current resource state. Resolve the conflict before retrying.
`412 Precondition Failed`	The `If-Match` version does not match the current resource. Retrieve and reconcile the latest resource.
`413 Payload Too Large`	The request body exceeds the deployment's configured maximum size. Reduce the body before retrying.
`428 Precondition Required`	A replacement request omitted the required `If-Match` header.
`429 Too Many Requests`	The client exceeded a deployment-defined rate limit. Follow the returned retry and quota headers.
`500 Internal Server Error`	Timescape encountered an unexpected error. Confirm the result before you retry the request safely, and contact the administrator if the failure persists.

The endpoint reference lists the status codes declared for each operation.

Problem Details responses

All declared `4xx` responses use `application/problem+json` and the Problem Details format defined by RFC 9457. Every response includes `type`, `title`, and `status`. A response can also include `detail`, `instance`, and extension members.

Validation failures can include an `invalidParams` array. Each entry contains the invalid field `name` and a human-readable `reason`. It can also contain a stable application `code`. Clients can use that code for program logic and show the reason to a user.

For example:

```
{
  "type": "/errors/invalid-request",
  "title": "Invalid request",
  "status": 400,
  "detail": "One or more request parameters are invalid.",
  "instance": "/api/hs/v1/network-object-groups/prod/web",
  "invalidParams": [
    {
      "name": "spec.cidrs[0]",
      "reason": "must be a valid CIDR prefix",
      "code": "invalid_cidr"
    }
  ]
}
```

Internal server error responses

The `500` response does not use Problem Details. It uses `application/json` with this deliberately minimal envelope:

```
{
  "errorMessage": "Internal server error"
}
```

The message does not expose internal diagnostics. If the error persists, report the request method, path, approximate time, and Timescape build version to the administrator. Do not include bearer tokens in logs or support data.

Rate limits

Timescape rate limits are deployment-defined. The public API does not define a single numeric quota that applies to every deployment, principal, or endpoint. Ask the Timescape administrator for the limits configured for your deployment.

Rate limit response

When a client exceeds an applicable quota, Timescape returns `429 Too Many Requests` with an `application/problem+json` body. The response declares these headers:

Header	Meaning
`Retry-After`	Time to wait before another request.
`RateLimit`	Current service limit information for the applicable quota policy.
`RateLimit-Policy`	The quota policy consumed by the request.

`RateLimit` and `RateLimit-Policy` follow `draft-ietf-httpapi-ratelimit-headers-11`. Treat their policy names and numeric values as deployment data, not as a product-wide service-level guarantee.

Retry behavior

After a `429` response:

1. Stop sending requests covered by the exhausted quota.
2. Wait for the period specified by `Retry-After`.
3. Resume with bounded exponential backoff and randomized delay.
4. Reduce request concurrency if the client continues to receive `429`.

Do not retry in a tight loop. Coordinate retry behavior across workers that share a principal or deployment quota. For list-heavy clients, use pagination deliberately and avoid polling more frequently than the operational workflow requires.

Versioning and deprecation

Timescape exposes several independent version values. Use the value that matches the decision the client needs to make.

Version	Purpose
API family `v1`	Appears in the `/api/hs/v1` base path and identifies the public API major family.
OpenAPI contract version `1.0.0`	Appears in the OpenAPI document's `info.version` field and identifies this machine-readable contract revision.
Timescape build version	Comes from `GET /meta/build-info` and identifies the running Timescape public API build.
Resource `version`	Identifies a network object group or policy revision and supplies the value encoded in its `ETag`.

Clients should send requests to the explicit `v1` base path and use the contract published for that family. Do not infer the API family from the Timescape build version or a resource version.

When a read request supplies `version=N`, Timescape returns desired state from the latest resource revision less than or equal to `N`. The returned resource `version` identifies the revision that Timescape selected. If the request also supplies `includeStatus=true`, the policy configuration is that historical desired state, but `deploymentStatus` reports current observed state.

Deprecation signals

Successful operation responses can contain these lifecycle headers:

- `Deprecation` follows RFC 9745. Its structured field date indicates when an operation became or is expected to become deprecated.
- `Sunset` follows RFC 8594. Its HTTP date indicates when an operation is expected to become unavailable.

Check both headers on successful responses and surface them in client logs or monitoring. A `Deprecation` header does not by itself state when an operation is removed; use `Sunset` when it is present and consult the applicable release documentation.

The public API contract does not promise a fixed deprecation notice period. Do not derive one from example dates in the OpenAPI document.

Endpoint reference

Use this reference to find the parameters, request bodies, success responses, and declared errors for the 11 resource and metadata operations in the OpenAPI document. All paths are relative to `/api/hs/v1`.

See Errors and status codes for the response formats shared by API errors.

OpenAPI document

This page renders the OpenAPI document published with this documentation. You can also `download this copy as YAML` for use with compatible API tools.

To retrieve the contract bundled with a running Timescape deployment, send `GET /api/hs/v1/openapi.json` to the deployment-specific Timescape API origin. The endpoint returns the document as JSON with the `application/json` media type. For example:

```
curl --fail --silent --show-error \
  --output timescape-openapi.json \
  "https://timescape.example.com/api/hs/v1/openapi.json"
```

Timescape does not require a bearer token for this endpoint, even when public API RBAC is enabled. A gateway or ingress can still require deployment-specific credentials. The endpoint is served separately and does not appear in the rendered operation list below.

API operations

Meta

Meta API.

GET /meta/build-info

Get Timescape build information

Response JSON Object:

- **version** (*string, required*) – Public API build version.

Status Codes:

- 200 OK –

  Timescape build information

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "version": "1.25.0"
}
```

- 401 Unauthorized – Authentication is required or the supplied credentials are invalid.
- 403 Forbidden – The authenticated principal is not authorized to perform this operation.
- 408 Request Timeout – The server timed out waiting for the request.
- 429 Too Many Requests – The request rate limit was exceeded.
- 500 Internal Server Error – An unexpected internal server error occurred.

Response Headers:

- **Deprecation** (*string*) – Deprecation header as defined by RFC 9745. Date when the API became or will become deprecated, encoded as an HTTP structured field date.
- **Sunset** (*string:http-date*) – Sunset header as defined by RFC 8594. Date when the API is expected to become unavailable.
- **Retry-After** (*string*) – Time to wait before retrying the request.
- **RateLimit** (*string*) – RateLimit header as defined by draft-ietf-httpapi-ratelimit-headers-11. Current service limit available under a specific quota policy.
- **RateLimit-Policy** (*string*) – RateLimit-Policy header as defined by draft-ietf-httpapi-ratelimit-headers-11. Quota policy that client requests consume.

NetworkObjectGroups

Hypershield NetworkObjectGroup API.

GET /network-object-groups

List Hypershield network object groups

Lists active NetworkObjectGroup resources across namespaces.

Query Parameters:

- **namespace** (*string*) – Optional namespace filter. Omit to list across all authorized namespaces.
- **search** (*string*) – Case-insensitive substring search over the resource name and string fields.
- **labelSelector** (*string*) – Kubernetes label selector string used to filter resources by metadata labels.
- **version** (*string*) – Optional point-in-time resource version. Omit for the latest resource.
- **pageSize** (*integer:int32*) – Maximum number of results to return. The service may return fewer.
- **pageToken** (*string*) – Token from a previous list response.

Response JSON Object:

- **results[]** (*object*) – NetworkObjectGroup resource returned by Hypershield APIs.
- **results[].namespace** (*string, required*) – Namespace that owns this NetworkObjectGroup.

- **results[].name** (*string, required*) – NetworkObjectGroup resource name.
- **results[].labels** (*object*) – Kubernetes metadata labels.
- **results[].spec** (*object, required*) – Network locations that belong to a NetworkObjectGroup.
- **results[].spec.description** (*string*)
- **results[].spec.cidrs[]** (*string*)
- **results[].spec.virtualNetwork** (*object*) – Logical network identifiers.
- **results[].spec.virtualNetwork.vrfs[]** (*string*)
- **results[].spec.virtualNetwork.vlans[]** (*integer:int32*)
- **results[].clusterSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **results[].clusterSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **results[].clusterSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **results[].clusterSelector.matchExpressions[].key** (*string, required*)
- **results[].clusterSelector.matchExpressions[].operator** (*string:enum, required*)
- **results[].clusterSelector.matchExpressions[].values[]** (*string*)
- **results[].version** (*string, required*) – NetworkObjectGroup resource version used for optimistic concurrency.
- **results[].revisionTimestamp** (*string:date-time*) – Timestamp of the NetworkObjectGroup revision returned by this response.
- **nextPageToken** (*string*) – Token used to retrieve the next page of results.
- **totalCount** (*integer:int32, required*) – Total number of NetworkObjectGroups matching the request.

Status Codes:

- 200 OK –

  Network object groups

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "results": [
    {
      "namespace": "prod",
      "name": "web-servers",
      "labels": {
        "app": "web",
        "env": "prod"
      },
      "spec": {
        "description": "Production web servers",
        "cidrs": [
          "10.12.0.0/24"
        ],
        "virtualNetwork": {
          "vlans": [
            120
          ]
        }
      },
      "version": "42"
    }
  ],
  "totalCount": 1
}
```

- 400 Bad Request – The request is malformed or contains invalid parameters.
- 401 Unauthorized – Authentication is required or the supplied credentials are invalid.
- 403 Forbidden – The authenticated principal is not authorized to perform this operation.
- 408 Request Timeout – The server timed out waiting for the request.
- 429 Too Many Requests – The request rate limit was exceeded.
- 500 Internal Server Error – An unexpected internal server error occurred.

Response Headers:

- **Deprecation** (*string*) – Deprecation header as defined by RFC 9745. Date when the API became or will become deprecated, encoded as an HTTP structured field date.
- **Sunset** (*string:http-date*) – Sunset header as defined by RFC 8594. Date when the API is expected to become unavailable.
- **Retry-After** (*string*) – Time to wait before retrying the request.
- **RateLimit** (*string*) – RateLimit header as defined by draft-ietf-httpapi-ratelimit-headers-11. Current service limit available under a specific quota policy.
- **RateLimit-Policy** (*string*) – RateLimit-Policy header as defined by draft-ietf-httpapi-ratelimit-headers-11. Quota policy that client requests consume.

POST /network-object-groups

Create a Hypershield network object group

Creates a NetworkObjectGroup.

Request JSON Object:

- **namespace** (*string, required*) – Namespace where the NetworkObjectGroup will be created.
- **name** (*string, required*) – Name for the new NetworkObjectGroup.
- **labels** (*object*) – Kubernetes metadata labels.
- **spec** (*object, required*) – Network locations that belong to a NetworkObjectGroup.
- **spec.description** (*string*)
- **spec.cidrs[]** (*string*)
- **spec.virtualNetwork** (*object*) – Logical network identifiers.
- **spec.virtualNetwork.vrfs[]** (*string*)
- **spec.virtualNetwork.vlans[]** (*integer:int32*)
- **clusterSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **clusterSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **clusterSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **clusterSelector.matchExpressions[].key** (*string, required*)
- **clusterSelector.matchExpressions[].operator** (*string:enum, required*)
- **clusterSelector.matchExpressions[].values[]** (*string*)

```
POST /network-object-groups HTTP/1.1
Content-Type: application/json

{
  "namespace": "prod",
  "name": "web-servers",
  "labels": {
    "app": "web",
    "env": "prod"
  },
  "spec": {
    "description": "Production web servers",
    "cidrs": [
      "10.12.0.0/24"
    ],
    "virtualNetwork": {
      "vlans": [
        120
      ]
    }
  },
  "clusterSelector": {
    "matchLabels": {
      "region": "us-east"
    }
  }
}
```

Response JSON Object:

- **namespace** (*string, required*) – Namespace that owns this NetworkObjectGroup.
- **name** (*string, required*) – NetworkObjectGroup resource name.
- **labels** (*object*) – Kubernetes metadata labels.
- **spec** (*object, required*) – Network locations that belong to a NetworkObjectGroup.

- **spec.description** (*string*)
- **spec.cidrs[]** (*string*)
- **spec.virtualNetwork** (*object*) – Logical network identifiers.
- **spec.virtualNetwork.vrfs[]** (*string*)
- **spec.virtualNetwork.vlans[]** (*integer:int32*)
- **clusterSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **clusterSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **clusterSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **clusterSelector.matchExpressions[].key** (*string, required*)
- **clusterSelector.matchExpressions[].operator** (*string:enum, required*)
- **clusterSelector.matchExpressions[].values[]** (*string*)
- **version** (*string, required*) – NetworkObjectGroup resource version used for optimistic concurrency.
- **revisionTimestamp** (*string:date-time*) – Timestamp of the NetworkObjectGroup revision returned by this response.

Status Codes:

- 201 Created –

  Network object group created

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "namespace": "prod",
  "name": "web-servers",
  "spec": {
    "cidrs": [
      "10.12.0.0/24"
    ],
    "virtualNetwork": {
      "vlans": [
        120
      ]
    }
  },
  "version": "42"
}
```

- 400 Bad Request – The request is malformed or contains invalid parameters.
- 401 Unauthorized – Authentication is required or the supplied credentials are invalid.
- 403 Forbidden – The authenticated principal is not authorized to perform this operation.
- 408 Request Timeout – The server timed out waiting for the request.
- 409 Conflict – The requested change conflicts with the current resource.
- 413 Request Entity Too Large – The request body exceeds the configured maximum size.
- 429 Too Many Requests – The request rate limit was exceeded.
- 500 Internal Server Error – An unexpected internal server error occurred.

Response Headers:

- **Location** (*string:uri-reference*) – URI of the created resource.
- **ETag** (*string*) – Current resource version for optimistic concurrency, encoded as a strong quoted HTTP entity tag such as “42”.
- **Deprecation** (*string*) – Deprecation header as defined by RFC 9745. Date when the API became or will become deprecated, encoded as an HTTP structured field date.
- **Sunset** (*string:http-date*) – Sunset header as defined by RFC 8594. Date when the API is expected to become unavailable.
- **Retry-After** (*string*) – Time to wait before retrying the request.
- **RateLimit** (*string*) – RateLimit header as defined by draft-ietf-httpapi-ratelimit-headers-11. Current service limit available under a specific quota policy.
- **RateLimit-Policy** (*string*) – RateLimit-Policy header as defined by draft-ietf-httpapi-ratelimit-headers-11. Quota policy that client requests consume.

`GET /network-object-groups/{namespace}/{name}`

Get a Hypershield network object group

Parameters:

- **namespace** (*string, required*) – Kubernetes namespace that owns the Hypershield resource.
- **name** (*string, required*) – Kubernetes resource name.

Query Parameters:

- **version** (*string*) – Optional point-in-time resource version. Omit for the latest resource.

Response JSON Object:

- **namespace** (*string, required*) – Namespace that owns this NetworkObjectGroup.
- **name** (*string, required*) – NetworkObjectGroup resource name.
- **labels** (*object*) – Kubernetes metadata labels.
- **spec** (*object, required*) – Network locations that belong to a NetworkObjectGroup.
- **spec.description** (*string*)
- **spec.cidrs[]** (*string*)
- **spec.virtualNetwork** (*object*) – Logical network identifiers.
- **spec.virtualNetwork.vrfs[]** (*string*)
- **spec.virtualNetwork.vlans[]** (*integer:int32*)
- **clusterSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **clusterSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **clusterSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **clusterSelector.matchExpressions[].key** (*string, required*)
- **clusterSelector.matchExpressions[].operator** (*string:enum, required*)
- **clusterSelector.matchExpressions[].values[]** (*string*)
- **version** (*string, required*) – NetworkObjectGroup resource version used for optimistic concurrency.

- **revisionTimestamp** (*string:date-time*) – Timestamp of the NetworkObjectGroup revision returned by this response.

Status Codes:

- 200 OK –

  Network object group

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "namespace": "prod",
  "name": "web-servers",
  "labels": {
    "app": "web",
    "env": "prod"
  },
  "spec": {
    "description": "Production web servers",
    "cidrs": [
      "10.12.0.0/24"
    ],
    "virtualNetwork": {
      "vlans": [
        120
      ]
    }
  },
  "clusterSelector": {
    "matchLabels": {
      "region": "us-east"
    }
  },
  "version": "42",
  "revisionTimestamp": "2026-06-11T14:30:00Z"
}
```

- 400 Bad Request – The request is malformed or contains invalid parameters.
- 401 Unauthorized – Authentication is required or the supplied credentials are invalid.
- 403 Forbidden – The authenticated principal is not authorized to perform this operation.
- 404 Not Found – The requested resource was not found.
- 408 Request Timeout – The server timed out waiting for the request.
- 429 Too Many Requests – The request rate limit was exceeded.
- 500 Internal Server Error – An unexpected internal server error occurred.

Response Headers:

- **ETag** (*string*) – Current resource version for optimistic concurrency, encoded as a strong quoted HTTP entity tag such as “42”.
- **Deprecation** (*string*) – Deprecation header as defined by RFC 9745. Date when the API became or will become deprecated, encoded as an HTTP structured field date.
- **Sunset** (*string:http-date*) – Sunset header as defined by RFC 8594. Date when the API is expected to become unavailable.
- **Retry-After** (*string*) – Time to wait before retrying the request.
- **RateLimit** (*string*) – RateLimit header as defined by draft-ietf-httpapi-ratelimit-headers-11. Current service limit available under a specific quota policy.

- **RateLimit-Policy** (*string*) – RateLimit-Policy header as defined by draft-ietf-httpapi-ratelimit-headers-11. Quota policy that client requests consume.

PUT /network-object-groups/{namespace}/{name}

Update a Hypershield network object group

Idempotently replaces the NetworkObjectGroup configuration.

Request Headers:

- **If-Match** (*string, required*) – Required strong ETag returned by a prior GET, POST, or PUT. The Public API accepts exactly one quoted entity tag produced by this API, for example “42”. Weak ETags such as W/”42”, wildcard preconditions such as *, multiple entity tags, malformed values, and unquoted resource versions are rejected. A missing header returns 428 Precondition Required, while an invalid value returns 400 Bad Request.

Parameters:

- **namespace** (*string, required*) – Kubernetes namespace that owns the Hypershield resource.
- **name** (*string, required*) – Kubernetes resource name.

Request JSON Object:

- **labels** (*object*) – Kubernetes metadata labels.
- **spec** (*object, required*) – Network locations that belong to a NetworkObjectGroup.
- **spec.description** (*string*)
- **spec.cidrs[]** (*string*)
- **spec.virtualNetwork** (*object*) – Logical network identifiers.
- **spec.virtualNetwork.vrfs[]** (*string*)
- **spec.virtualNetwork.vlans[]** (*integer:int32*)
- **clusterSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **clusterSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **clusterSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **clusterSelector.matchExpressions[].key** (*string, required*)
- **clusterSelector.matchExpressions[].operator** (*string:enum, required*)
- **clusterSelector.matchExpressions[].values[]** (*string*)

```
PUT /network-object-groups/{namespace}/{name} HTTP/1.1
Content-Type: application/json

{
  "labels": {
    "app": "web",
    "env": "prod"
  },
  "spec": {
    "description": "Production web servers",
    "cidrs": [
      "10.12.0.0/24",
      "10.12.1.0/24"
    ],
    "virtualNetwork": {
      "vlans": [
        120
      ]
    }
  }
}
```

Response JSON Object:

- **namespace** (*string, required*) – Namespace that owns this NetworkObjectGroup.
- **name** (*string, required*) – NetworkObjectGroup resource name.
- **labels** (*object*) – Kubernetes metadata labels.
- **spec** (*object, required*) – Network locations that belong to a NetworkObjectGroup.
- **spec.description** (*string*)
- **spec.cidrs[]** (*string*)
- **spec.virtualNetwork** (*object*) – Logical network identifiers.
- **spec.virtualNetwork.vrfs[]** (*string*)
- **spec.virtualNetwork.vlans[]** (*integer:int32*)
- **clusterSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **clusterSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **clusterSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **clusterSelector.matchExpressions[].key** (*string, required*)
- **clusterSelector.matchExpressions[].operator** (*string:enum, required*)
- **clusterSelector.matchExpressions[].values[]** (*string*)
- **version** (*string, required*) – NetworkObjectGroup resource version used for optimistic concurrency.
- **revisionTimestamp** (*string:date-time*) – Timestamp of the NetworkObjectGroup revision returned by this response.

Status Codes:

- 200 OK –

  Network object group updated

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "namespace": "prod",
  "name": "web-servers",
  "spec": {
    "cidrs": [
      "10.12.0.0/24",
      "10.12.1.0/24"
    ],
    "virtualNetwork": {
      "vlans": [
        120
      ]
    }
  },
  "version": "43"
}
```

- 400 Bad Request – The request is malformed or contains invalid parameters.
- 401 Unauthorized – Authentication is required or the supplied credentials are invalid.
- 403 Forbidden – The authenticated principal is not authorized to perform this operation.
- 404 Not Found – The requested resource was not found.
- 408 Request Timeout – The server timed out waiting for the request.
- 409 Conflict – The requested change conflicts with the current resource.
- 412 Precondition Failed – The supplied resource version precondition does not match the current resource version.
- 428 Precondition Required – The If-Match header is required for this operation.
- 413 Request Entity Too Large – The request body exceeds the configured maximum size.
- 429 Too Many Requests – The request rate limit was exceeded.
- 500 Internal Server Error – An unexpected internal server error occurred.

Response Headers:

- **ETag** (*string*) – Current resource version for optimistic concurrency, encoded as a strong quoted HTTP entity tag such as “42”.
- **Deprecation** (*string*) – Deprecation header as defined by RFC 9745. Date when the API became or will become deprecated, encoded as an HTTP structured field date.
- **Sunset** (*string:http-date*) – Sunset header as defined by RFC 8594. Date when the API is expected to become unavailable.
- **Retry-After** (*string*) – Time to wait before retrying the request.
- **RateLimit** (*string*) – RateLimit header as defined by draft-ietf-httpapi-ratelimit-headers-11. Current service limit available under a specific quota policy.
- **RateLimit-Policy** (*string*) – RateLimit-Policy header as defined by draft-ietf-httpapi-ratelimit-headers-11. Quota policy that client requests consume.

`DELETE /network-object-groups/{namespace}/{name}`

Delete a Hypershield network object group

Deletes the NetworkObjectGroup.

Parameters:

- **namespace** (*string, required*) – Kubernetes namespace that owns the Hypershield resource.
- **name** (*string, required*) – Kubernetes resource name.

Status Codes:

- 204 No Content – Network object group deleted
- 400 Bad Request – The request is malformed or contains invalid parameters.
- 401 Unauthorized – Authentication is required or the supplied credentials are invalid.
- 403 Forbidden – The authenticated principal is not authorized to perform this operation.
- 404 Not Found – The requested resource was not found.
- 408 Request Timeout – The server timed out waiting for the request.
- 409 Conflict – The requested change conflicts with the current resource.
- 429 Too Many Requests – The request rate limit was exceeded.
- 500 Internal Server Error – An unexpected internal server error occurred.

Response Headers:

- **Deprecation** (*string*) – Deprecation header as defined by RFC 9745. Date when the API became or will become deprecated, encoded as an HTTP structured field date.
- **Sunset** (*string:http-date*) – Sunset header as defined by RFC 8594. Date when the API is expected to become unavailable.
- **Retry-After** (*string*) – Time to wait before retrying the request.
- **RateLimit** (*string*) – RateLimit header as defined by draft-ietf-httpapi-ratelimit-headers-11. Current service limit available under a specific quota policy.
- **RateLimit-Policy** (*string*) – RateLimit-Policy header as defined by draft-ietf-httpapi-ratelimit-headers-11. Quota policy that client requests consume.

Policies

Hypershield Policy API.

`GET /policies`

List Hypershield policies

Lists active SmartSwitchNetworkPolicy resources across namespaces.

Query Parameters:

- **namespace** (*string*) – Optional namespace filter. Omit to list across all authorized namespaces.
- **search** (*string*) – Case-insensitive substring search over the resource name and string fields.
- **labelSelector** (*string*) – Kubernetes label selector string used to filter resources by metadata labels.
- **version** (*string*) – Optional point-in-time resource version. Omit for the latest resource.
- **includeStatus** (*boolean*) – Include policy deployment status when true.

- **pageSize** (*integer:int32*) – Maximum number of results to return. The service may return fewer.
- **pageToken** (*string*) – Token from a previous list response.

Response JSON Object:

- **results[]** (*object*) – SmartSwitchNetworkPolicy resource returned by Hypershield APIs.
- **results[].namespace** (*string, required*) – Namespace that owns this SmartSwitchNetworkPolicy.
- **results[].name** (*string, required*) – SmartSwitchNetworkPolicy resource name.
- **results[].labels** (*object*) – Kubernetes metadata labels.
- **results[].spec** (*object, required*) – Ordered traffic rules for a SmartSwitchNetworkPolicy.
- **results[].spec.rules[]** (*object*) – Source, destination and action for one policy rule.
- **results[].spec.rules[].description** (*string*)
- **results[].spec.rules[].action** (*string:enum, required*)
- **results[].spec.rules[].source** (*object, required*) – Source selector. Specify ipBlocks or networkRefs, not both.
- **results[].spec.rules[].source.ipBlocks[]** (*object*) – Network locations that belong to a NetworkObjectGroup.
- **results[].spec.rules[].source.ipBlocks[].description** (*string*)
- **results[].spec.rules[].source.ipBlocks[].cidrs[]** (*string*)
- **results[].spec.rules[].source.ipBlocks[].virtualNetwork** (*object*) – Logical network identifiers.
- **results[].spec.rules[].source.ipBlocks[].virtualNetwork.vrfs[]** (*string*)
- **results[].spec.rules[].source.ipBlocks[].virtualNetwork.vlans[]** (*integer:int32*)
- **results[].spec.rules[].source.networkRefs[]** (*object*) – Reference to NetworkObjectGroups in the policy namespace. Specify name or labelSelector, not both.
- **results[].spec.rules[].source.networkRefs[].name** (*string*)
- **results[].spec.rules[].source.networkRefs[].labelSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **results[].spec.rules[].source.networkRefs[].labelSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **results[].spec.rules[].source.networkRefs[].labelSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **results[].spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].key** (*string, required*)
- **results[].spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].operator** (*string:enum, required*)
- **results[].spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].values[]** (*string*)
- **results[].spec.rules[].destination** (*object, required*) – Destination selector. Specify ipBlocks or networkRefs, not both.
- **results[].spec.rules[].destination.ipBlocks[]** (*object*) – Network locations that belong to a NetworkObjectGroup.
- **results[].spec.rules[].destination.ipBlocks[].description** (*string*)
- **results[].spec.rules[].destination.ipBlocks[].cidrs[]** (*string*)
- **results[].spec.rules[].destination.ipBlocks[].virtualNetwork** (*object*) – Logical network identifiers.

- **results[].spec.rules[].destination.ipBlocks[].virtualNetwork.vrfs[]** (*string*)
- **results[].spec.rules[].destination.ipBlocks[].virtualNetwork.vlans[]** (*integer:int32*)
- **results[].spec.rules[].destination.networkRefs[]** (*object*) – Reference to NetworkObjectGroups in the policy namespace. Specify name or labelSelector, not both.
- **results[].spec.rules[].destination.networkRefs[].name** (*string*)
- **results[].spec.rules[].destination.networkRefs[].labelSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **results[].spec.rules[].destination.networkRefs[].labelSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **results[].spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **results[].spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].key** (*string, required*)
- **results[].spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].operator** (*string:enum, required*)
- **results[].spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].values[]** (*string*)
- **results[].spec.rules[].destination.protoPorts[]** (*object*) – Transport protocol and optional port or port range.
- **results[].spec.rules[].destination.protoPorts[].protocol** (*string:enum, required*)
- **results[].spec.rules[].destination.protoPorts[].port** (*integer:int32*)
- **results[].spec.rules[].destination.protoPorts[].endPort** (*integer:int32*) – Inclusive end of a port range. Requires port and must be greater than port.
- **results[].clusterSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **results[].clusterSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **results[].clusterSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **results[].clusterSelector.matchExpressions[].key** (*string, required*)
- **results[].clusterSelector.matchExpressions[].operator** (*string:enum, required*)
- **results[].clusterSelector.matchExpressions[].values[]** (*string*)
- **results[].version** (*string, required*) – SmartSwitchNetworkPolicy version used for optimistic concurrency.
- **results[].revisionTimestamp** (*string:date-time*) – Timestamp of the SmartSwitchNetworkPolicy revision in this response.
- **results[].deploymentStatus** (*object*) – Aggregated deployment state for a SmartSwitchNetworkPolicy.
- **results[].deploymentStatus.overallStatus** (*string:enum, required*)
- **results[].deploymentStatus.totalAgents** (*integer:int64*)
- **results[].deploymentStatus.inSyncAgents** (*integer:int64*)
- **results[].deploymentStatus.outOfSyncAgents** (*integer:int64*)
- **results[].deploymentStatus.pendingAgents** (*integer:int64*)
- **nextPageToken** (*string*) – Cursor for the following page of policies.

- **totalCount** (*integer:int32, required*) – Total number of SmartSwitchNetworkPolicies matching the request.

Status Codes:

- 200 OK –

  Policies

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "results": [
    {
      "namespace": "prod",
      "name": "allow-web-to-checkout",
      "spec": {
        "rules": [
          {
            "action": "allow",
            "source": {
              "networkRefs": [
                {
                  "labelSelector": {
                    "matchLabels": {
                      "role": "web"
                    }
                  }
                }
              ]
            },
            "destination": {
              "networkRefs": [
                {
                  "name": "checkout-api"
                }
              ],
              "protoPorts": [
                {
                  "protocol": "TCP",
                  "port": 443
                }
              ]
            }
          }
        ]
      },
      "version": "17"
    }
  ],
  "totalCount": 1
}
```

- 400 Bad Request – The request is malformed or contains invalid parameters.
- 401 Unauthorized – Authentication is required or the supplied credentials are invalid.
- 403 Forbidden – The authenticated principal is not authorized to perform this operation.
- 408 Request Timeout – The server timed out waiting for the request.
- 429 Too Many Requests – The request rate limit was exceeded.
- 500 Internal Server Error – An unexpected internal server error occurred.

Response Headers:

- **Deprecation** (*string*) – Deprecation header as defined by RFC 9745. Date when the API became or will become deprecated, encoded as an HTTP structured field date.
- **Sunset** (*string:http-date*) – Sunset header as defined by RFC 8594. Date when the API is expected to become unavailable.
- **Retry-After** (*string*) – Time to wait before retrying the request.
- **RateLimit** (*string*) – RateLimit header as defined by draft-ietf-httpapi-ratelimit-headers-11. Current service limit available under a specific quota policy.
- **RateLimit-Policy** (*string*) – RateLimit-Policy header as defined by draft-ietf-httpapi-ratelimit-headers-11. Quota policy that client requests consume.

POST /policies

Create a Hypershield policy

Creates a SmartSwitchNetworkPolicy.

Request JSON Object:

- **namespace** (*string, required*) – Namespace where the SmartSwitchNetworkPolicy will be created.
- **name** (*string, required*) – Name for the new SmartSwitchNetworkPolicy.
- **labels** (*object*) – Kubernetes metadata labels.
- **spec** (*object, required*) – Ordered traffic rules for a SmartSwitchNetworkPolicy.
- **spec.rules[]** (*object*) – Source, destination and action for one policy rule.
- **spec.rules[].description** (*string*)
- **spec.rules[].action** (*string:enum, required*)
- **spec.rules[].source** (*object, required*) – Source selector. Specify ipBlocks or networkRefs, not both.
- **spec.rules[].source.ipBlocks[]** (*object*) – Network locations that belong to a NetworkObjectGroup.
- **spec.rules[].source.ipBlocks[].description** (*string*)
- **spec.rules[].source.ipBlocks[].cidrs[]** (*string*)
- **spec.rules[].source.ipBlocks[].virtualNetwork** (*object*) – Logical network identifiers.
- **spec.rules[].source.ipBlocks[].virtualNetwork.vrfs[]** (*string*)
- **spec.rules[].source.ipBlocks[].virtualNetwork.vlans[]** (*integer:int32*)
- **spec.rules[].source.networkRefs[]** (*object*) – Reference to NetworkObjectGroups in the policy namespace. Specify name or labelSelector, not both.
- **spec.rules[].source.networkRefs[].name** (*string*)
- **spec.rules[].source.networkRefs[].labelSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **spec.rules[].source.networkRefs[].labelSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].key** (*string, required*)

- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].operator** (*string:enum, required*)
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].values[]** (*string*)
- **spec.rules[].destination** (*object, required*) – Destination selector. Specify ipBlocks or networkRefs, not both.
- **spec.rules[].destination.ipBlocks[]** (*object*) – Network locations that belong to a NetworkObjectGroup.
- **spec.rules[].destination.ipBlocks[].description** (*string*)
- **spec.rules[].destination.ipBlocks[].cidrs[]** (*string*)
- **spec.rules[].destination.ipBlocks[].virtualNetwork** (*object*) – Logical network identifiers.
- **spec.rules[].destination.ipBlocks[].virtualNetwork.vrfs[]** (*string*)
- **spec.rules[].destination.ipBlocks[].virtualNetwork.vlans[]** (*integer:int32*)
- **spec.rules[].destination.networkRefs[]** (*object*) – Reference to NetworkObjectGroups in the policy namespace. Specify name or labelSelector, not both.
- **spec.rules[].destination.networkRefs[].name** (*string*)
- **spec.rules[].destination.networkRefs[].labelSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **spec.rules[].destination.networkRefs[].labelSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].key** (*string, required*)
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].operator** (*string:enum, required*)
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].values[]** (*string*)
- **spec.rules[].destination.protoPorts[]** (*object*) – Transport protocol and optional port or port range.
- **spec.rules[].destination.protoPorts[].protocol** (*string:enum, required*)
- **spec.rules[].destination.protoPorts[].port** (*integer:int32*)
- **spec.rules[].destination.protoPorts[].endPort** (*integer:int32*) – Inclusive end of a port range. Requires port and must be greater than port.
- **clusterSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **clusterSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **clusterSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **clusterSelector.matchExpressions[].key** (*string, required*)
- **clusterSelector.matchExpressions[].operator** (*string:enum, required*)
- **clusterSelector.matchExpressions[].values[]** (*string*)

```
POST /policies HTTP/1.1
Content-Type: application/json

{
  "namespace": "prod",
  "name": "allow-web-to-checkout",
  "labels": {
    "app": "web",
    "env": "prod"
  },
  "spec": {
    "rules": [
      {
        "description": "Allow web to checkout API",
        "action": "allow",
        "source": {
          "networkRefs": [
            {
              "labelSelector": {
                "matchLabels": {
                  "role": "web"
                }
              }
            }
          ]
        },
        "destination": {
          "networkRefs": [
            {
              "name": "checkout-api"
            }
          ],
          "protoPorts": [
            {
              "protocol": "TCP",
              "port": 443
            }
          ]
        }
      }
    ]
  }
}
```

Response JSON Object:

- **namespace** (*string, required*) – Namespace that owns this SmartSwitchNetworkPolicy.
- **name** (*string, required*) – SmartSwitchNetworkPolicy resource name.
- **labels** (*object*) – Kubernetes metadata labels.
- **spec** (*object, required*) – Ordered traffic rules for a SmartSwitchNetworkPolicy.
- **spec.rules[]** (*object*) – Source, destination and action for one policy rule.
- **spec.rules[].description** (*string*)
- **spec.rules[].action** (*string:enum, required*)
- **spec.rules[].source** (*object, required*) – Source selector. Specify ipBlocks or networkRefs, not both.
- **spec.rules[].source.ipBlocks[]** (*object*) – Network locations that belong to a NetworkObjectGroup.
- **spec.rules[].source.ipBlocks[].description** (*string*)
- **spec.rules[].source.ipBlocks[].cidrs[]** (*string*)
- **spec.rules[].source.ipBlocks[].virtualNetwork** (*object*) – Logical network identifiers.

- **spec.rules[].source.ipBlocks[].virtualNetwork.vrfs[]** (*string*)
- **spec.rules[].source.ipBlocks[].virtualNetwork.vlans[]** (*integer:int32*)
- **spec.rules[].source.networkRefs[]** (*object*) – Reference to NetworkObjectGroups in the policy namespace. Specify name or labelSelector, not both.
- **spec.rules[].source.networkRefs[].name** (*string*)
- **spec.rules[].source.networkRefs[].labelSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **spec.rules[].source.networkRefs[].labelSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].key** (*string, required*)
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].operator** (*string:enum, required*)
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].values[]** (*string*)
- **spec.rules[].destination** (*object, required*) – Destination selector. Specify ipBlocks or networkRefs, not both.
- **spec.rules[].destination.ipBlocks[]** (*object*) – Network locations that belong to a NetworkObjectGroup.
- **spec.rules[].destination.ipBlocks[].description** (*string*)
- **spec.rules[].destination.ipBlocks[].cidrs[]** (*string*)
- **spec.rules[].destination.ipBlocks[].virtualNetwork** (*object*) – Logical network identifiers.
- **spec.rules[].destination.ipBlocks[].virtualNetwork.vrfs[]** (*string*)
- **spec.rules[].destination.ipBlocks[].virtualNetwork.vlans[]** (*integer:int32*)
- **spec.rules[].destination.networkRefs[]** (*object*) – Reference to NetworkObjectGroups in the policy namespace. Specify name or labelSelector, not both.
- **spec.rules[].destination.networkRefs[].name** (*string*)
- **spec.rules[].destination.networkRefs[].labelSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **spec.rules[].destination.networkRefs[].labelSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].key** (*string, required*)
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].operator** (*string:enum, required*)
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].values[]** (*string*)
- **spec.rules[].destination.protoPorts[]** (*object*) – Transport protocol and optional port or port range.
- **spec.rules[].destination.protoPorts[].protocol** (*string:enum, required*)
- **spec.rules[].destination.protoPorts[].port** (*integer:int32*)

- **spec.rules[].destination.protoPorts[].endPort** (*integer:int32*) – Inclusive end of a port range. Requires port and must be greater than port.
- **clusterSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **clusterSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **clusterSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **clusterSelector.matchExpressions[].key** (*string, required*)
- **clusterSelector.matchExpressions[].operator** (*string:enum, required*)
- **clusterSelector.matchExpressions[].values[]** (*string*)
- **version** (*string, required*) – SmartSwitchNetworkPolicy version used for optimistic concurrency.
- **revisionTimestamp** (*string:date-time*) – Timestamp of the SmartSwitchNetworkPolicy revision in this response.
- **deploymentStatus** (*object*) – Aggregated deployment state for a SmartSwitchNetworkPolicy.
- **deploymentStatus.overallStatus** (*string:enum, required*)
- **deploymentStatus.totalAgents** (*integer:int64*)
- **deploymentStatus.inSyncAgents** (*integer:int64*)
- **deploymentStatus.outOfSyncAgents** (*integer:int64*)
- **deploymentStatus.pendingAgents** (*integer:int64*)

Status Codes:

- 201 Created –

  Policy created

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "namespace": "prod",
  "name": "allow-web-to-checkout",
  "spec": {
    "rules": [
      {
        "action": "allow",
        "source": {
          "networkRefs": [
            {
              "labelSelector": {
                "matchLabels": {
                  "role": "web"
                }
              }
            }
          ]
        },
        "destination": {
          "networkRefs": [
            {
              "name": "checkout-api"
            }
          ],
          "protoPorts": [
            {
              "protocol": "TCP",
              "port": 443
            }
          ]
        }
      }
    ]
  },
  "version": "17"
}
```

- 400 Bad Request – The request is malformed or contains invalid parameters.
- 401 Unauthorized – Authentication is required or the supplied credentials are invalid.
- 403 Forbidden – The authenticated principal is not authorized to perform this operation.
- 408 Request Timeout – The server timed out waiting for the request.
- 409 Conflict – The requested change conflicts with the current resource.
- 413 Request Entity Too Large – The request body exceeds the configured maximum size.
- 429 Too Many Requests – The request rate limit was exceeded.
- 500 Internal Server Error – An unexpected internal server error occurred.

Response Headers:

- **Location** (*string:uri-reference*) – URI of the created resource.
- **ETag** (*string*) – Current resource version for optimistic concurrency, encoded as a strong quoted HTTP entity tag such as “42”.
- **Deprecation** (*string*) – Deprecation header as defined by RFC 9745. Date when the API became or will become deprecated, encoded as an HTTP structured field date.
- **Sunset** (*string:http-date*) – Sunset header as defined by RFC 8594. Date when the API is expected to become unavailable.

- **Retry-After** (*string*) – Time to wait before retrying the request.
- **RateLimit** (*string*) – RateLimit header as defined by draft-ietf-httpapi-ratelimit-headers-11. Current service limit available under a specific quota policy.
- **RateLimit-Policy** (*string*) – RateLimit-Policy header as defined by draft-ietf-httpapi-ratelimit-headers-11. Quota policy that client requests consume.

`GET /policies/{namespace}/{name}`

Get a Hypershield policy

Parameters:

- **namespace** (*string, required*) – Kubernetes namespace that owns the Hypershield resource.
- **name** (*string, required*) – Kubernetes resource name.

Query Parameters:

- **version** (*string*) – Optional point-in-time resource version. Omit for the latest resource.
- **includeStatus** (*boolean*) – Include policy deployment status when true.

Response JSON Object:

- **namespace** (*string, required*) – Namespace that owns this SmartSwitchNetworkPolicy.
- **name** (*string, required*) – SmartSwitchNetworkPolicy resource name.
- **labels** (*object*) – Kubernetes metadata labels.
- **spec** (*object, required*) – Ordered traffic rules for a SmartSwitchNetworkPolicy.
- **spec.rules[]** (*object*) – Source, destination and action for one policy rule.
- **spec.rules[].description** (*string*)
- **spec.rules[].action** (*string:enum, required*)
- **spec.rules[].source** (*object, required*) – Source selector. Specify ipBlocks or networkRefs, not both.
- **spec.rules[].source.ipBlocks[]** (*object*) – Network locations that belong to a NetworkObjectGroup.
- **spec.rules[].source.ipBlocks[].description** (*string*)
- **spec.rules[].source.ipBlocks[].cidrs[]** (*string*)
- **spec.rules[].source.ipBlocks[].virtualNetwork** (*object*) – Logical network identifiers.
- **spec.rules[].source.ipBlocks[].virtualNetwork.vrfs[]** (*string*)
- **spec.rules[].source.ipBlocks[].virtualNetwork.vlans[]** (*integer:int32*)
- **spec.rules[].source.networkRefs[]** (*object*) – Reference to NetworkObjectGroups in the policy namespace. Specify name or labelSelector, not both.
- **spec.rules[].source.networkRefs[].name** (*string*)
- **spec.rules[].source.networkRefs[].labelSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **spec.rules[].source.networkRefs[].labelSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].key** (*string, required*)

- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].operator** (*string:enum, required*)
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].values[]** (*string*)
- **spec.rules[].destination** (*object, required*) – Destination selector. Specify ipBlocks or networkRefs, not both.
- **spec.rules[].destination.ipBlocks[]** (*object*) – Network locations that belong to a NetworkObjectGroup.
- **spec.rules[].destination.ipBlocks[].description** (*string*)
- **spec.rules[].destination.ipBlocks[].cidrs[]** (*string*)
- **spec.rules[].destination.ipBlocks[].virtualNetwork** (*object*) – Logical network identifiers.
- **spec.rules[].destination.ipBlocks[].virtualNetwork.vrfs[]** (*string*)
- **spec.rules[].destination.ipBlocks[].virtualNetwork.vlans[]** (*integer:int32*)
- **spec.rules[].destination.networkRefs[]** (*object*) – Reference to NetworkObjectGroups in the policy namespace. Specify name or labelSelector, not both.
- **spec.rules[].destination.networkRefs[].name** (*string*)
- **spec.rules[].destination.networkRefs[].labelSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **spec.rules[].destination.networkRefs[].labelSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].key** (*string, required*)
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].operator** (*string:enum, required*)
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].values[]** (*string*)
- **spec.rules[].destination.protoPorts[]** (*object*) – Transport protocol and optional port or port range.
- **spec.rules[].destination.protoPorts[].protocol** (*string:enum, required*)
- **spec.rules[].destination.protoPorts[].port** (*integer:int32*)
- **spec.rules[].destination.protoPorts[].endPort** (*integer:int32*) – Inclusive end of a port range. Requires port and must be greater than port.
- **clusterSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **clusterSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **clusterSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **clusterSelector.matchExpressions[].key** (*string, required*)
- **clusterSelector.matchExpressions[].operator** (*string:enum, required*)
- **clusterSelector.matchExpressions[].values[]** (*string*)
- **version** (*string, required*) – SmartSwitchNetworkPolicy version used for optimistic concurrency.
- **revisionTimestamp** (*string:date-time*) – Timestamp of the SmartSwitchNetworkPolicy revision in this response.
- **deploymentStatus** (*object*) – Aggregated deployment state for a SmartSwitchNetworkPolicy.

- **deploymentStatus.overallStatus** (*string:enum, required*)
- **deploymentStatus.totalAgents** (*integer:int64*)
- **deploymentStatus.inSyncAgents** (*integer:int64*)
- **deploymentStatus.outOfSyncAgents** (*integer:int64*)
- **deploymentStatus.pendingAgents** (*integer:int64*)

Status Codes:

- 200 OK –

  Policy

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "namespace": "prod",
  "name": "allow-web-to-checkout",
  "labels": {
    "app": "web",
    "env": "prod"
  },
  "spec": {
    "rules": [
      {
        "description": "Allow web to checkout API",
        "action": "allow",
        "source": {
          "networkRefs": [
            {
              "labelSelector": {
                "matchLabels": {
                  "role": "web"
                }
              }
            }
          ]
        },
        "destination": {
          "networkRefs": [
            {
              "name": "checkout-api"
            }
          ],
          "protoPorts": [
            {
              "protocol": "TCP",
              "port": 443
            }
          ]
        }
      }
    ]
  },
  "version": "17",
  "deploymentStatus": {
    "overallStatus": "inSync",
    "totalAgents": 12,
    "inSyncAgents": 12,
    "outOfSyncAgents": 0,
    "pendingAgents": 0
  }
}
```

- 400 Bad Request – The request is malformed or contains invalid parameters.
- 401 Unauthorized – Authentication is required or the supplied credentials are invalid.
- 403 Forbidden – The authenticated principal is not authorized to perform this operation.
- 404 Not Found – The requested resource was not found.
- 408 Request Timeout – The server timed out waiting for the request.
- 429 Too Many Requests – The request rate limit was exceeded.
- 500 Internal Server Error – An unexpected internal server error occurred.

Response Headers:

- **ETag** (*string*) – Current resource version for optimistic concurrency, encoded as a strong quoted HTTP entity tag such as “42”.
- **Deprecation** (*string*) – Deprecation header as defined by RFC 9745. Date when the API became or will become deprecated, encoded as an HTTP structured field date.
- **Sunset** (*string:http-date*) – Sunset header as defined by RFC 8594. Date when the API is expected to become unavailable.
- **Retry-After** (*string*) – Time to wait before retrying the request.
- **RateLimit** (*string*) – RateLimit header as defined by draft-ietf-httpapi-ratelimit-headers-11. Current service limit available under a specific quota policy.
- **RateLimit-Policy** (*string*) – RateLimit-Policy header as defined by draft-ietf-httpapi-ratelimit-headers-11. Quota policy that client requests consume.

`PUT /policies/{namespace}/{name}`

Update a Hypershield policy

Idempotently replaces the SmartSwitchNetworkPolicy configuration.

Request Headers:

- **If-Match** (*string, required*) – Required strong ETag returned by a prior GET, POST, or PUT. The Public API accepts exactly one quoted entity tag produced by this API, for example “42”. Weak ETags such as W/”42”, wildcard preconditions such as *, multiple entity tags, malformed values, and unquoted resource versions are rejected. A missing header returns 428 Precondition Required, while an invalid value returns 400 Bad Request.

Parameters:

- **namespace** (*string, required*) – Kubernetes namespace that owns the Hypershield resource.
- **name** (*string, required*) – Kubernetes resource name.

Request JSON Object:

- **labels** (*object*) – Kubernetes metadata labels.
- **spec** (*object, required*) – Ordered traffic rules for a SmartSwitchNetworkPolicy.
- **spec.rules[]** (*object*) – Source, destination and action for one policy rule.
- **spec.rules[].description** (*string*)
- **spec.rules[].action** (*string:enum, required*)
- **spec.rules[].source** (*object, required*) – Source selector. Specify ipBlocks or networkRefs, not both.

- **spec.rules[].source.ipBlocks[]** (*object*) – Network locations that belong to a NetworkObjectGroup.
- **spec.rules[].source.ipBlocks[].description** (*string*)
- **spec.rules[].source.ipBlocks[].cidrs[]** (*string*)
- **spec.rules[].source.ipBlocks[].virtualNetwork** (*object*) – Logical network identifiers.
- **spec.rules[].source.ipBlocks[].virtualNetwork.vrfs[]** (*string*)
- **spec.rules[].source.ipBlocks[].virtualNetwork.vlans[]** (*integer:int32*)
- **spec.rules[].source.networkRefs[]** (*object*) – Reference to NetworkObjectGroups in the policy namespace. Specify name or labelSelector, not both.
- **spec.rules[].source.networkRefs[].name** (*string*)
- **spec.rules[].source.networkRefs[].labelSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **spec.rules[].source.networkRefs[].labelSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].key** (*string, required*)
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].operator** (*string:enum, required*)
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].values[]** (*string*)
- **spec.rules[].destination** (*object, required*) – Destination selector. Specify ipBlocks or networkRefs, not both.
- **spec.rules[].destination.ipBlocks[]** (*object*) – Network locations that belong to a NetworkObjectGroup.
- **spec.rules[].destination.ipBlocks[].description** (*string*)
- **spec.rules[].destination.ipBlocks[].cidrs[]** (*string*)
- **spec.rules[].destination.ipBlocks[].virtualNetwork** (*object*) – Logical network identifiers.
- **spec.rules[].destination.ipBlocks[].virtualNetwork.vrfs[]** (*string*)
- **spec.rules[].destination.ipBlocks[].virtualNetwork.vlans[]** (*integer:int32*)
- **spec.rules[].destination.networkRefs[]** (*object*) – Reference to NetworkObjectGroups in the policy namespace. Specify name or labelSelector, not both.
- **spec.rules[].destination.networkRefs[].name** (*string*)
- **spec.rules[].destination.networkRefs[].labelSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **spec.rules[].destination.networkRefs[].labelSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].key** (*string, required*)
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].operator** (*string:enum, required*)
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].values[]** (*string*)

- **spec.rules[].destination.protoPorts[]** (*object*) – Transport protocol and optional port or port range.
- **spec.rules[].destination.protoPorts[].protocol** (*string:enum, required*)
- **spec.rules[].destination.protoPorts[].port** (*integer:int32*)
- **spec.rules[].destination.protoPorts[].endPort** (*integer:int32*) – Inclusive end of a port range. Requires port and must be greater than port.
- **clusterSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **clusterSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **clusterSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **clusterSelector.matchExpressions[].key** (*string, required*)
- **clusterSelector.matchExpressions[].operator** (*string:enum, required*)
- **clusterSelector.matchExpressions[].values[]** (*string*)

```
PUT /policies/{namespace}/{name} HTTP/1.1
Content-Type: application/json

{
  "labels": {
    "app": "web",
    "env": "prod"
  },
  "spec": {
    "rules": [
      {
        "description": "Allow web to checkout API",
        "action": "allow",
        "source": {
          "networkRefs": [
            {
              "labelSelector": {
                "matchLabels": {
                  "role": "web"
                }
              }
            }
          ]
        },
        "destination": {
          "networkRefs": [
            {
              "name": "checkout-api"
            }
          ],
          "protoPorts": [
            {
              "protocol": "TCP",
              "port": 443
            }
          ]
        }
      }
    ]
  }
}
```

Response JSON Object:

- **namespace** (*string, required*) – Namespace that owns this SmartSwitchNetworkPolicy.

- **name** (*string, required*) – SmartSwitchNetworkPolicy resource name.
- **labels** (*object*) – Kubernetes metadata labels.
- **spec** (*object, required*) – Ordered traffic rules for a SmartSwitchNetworkPolicy.
- **spec.rules[]** (*object*) – Source, destination and action for one policy rule.
- **spec.rules[].description** (*string*)
- **spec.rules[].action** (*string:enum, required*)
- **spec.rules[].source** (*object, required*) – Source selector. Specify ipBlocks or networkRefs, not both.
- **spec.rules[].source.ipBlocks[]** (*object*) – Network locations that belong to a NetworkObjectGroup.
- **spec.rules[].source.ipBlocks[].description** (*string*)
- **spec.rules[].source.ipBlocks[].cidrs[]** (*string*)
- **spec.rules[].source.ipBlocks[].virtualNetwork** (*object*) – Logical network identifiers.
- **spec.rules[].source.ipBlocks[].virtualNetwork.vrfs[]** (*string*)
- **spec.rules[].source.ipBlocks[].virtualNetwork.vlans[]** (*integer:int32*)
- **spec.rules[].source.networkRefs[]** (*object*) – Reference to NetworkObjectGroups in the policy namespace. Specify name or labelSelector, not both.
- **spec.rules[].source.networkRefs[].name** (*string*)
- **spec.rules[].source.networkRefs[].labelSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **spec.rules[].source.networkRefs[].labelSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].key** (*string, required*)
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].operator** (*string:enum, required*)
- **spec.rules[].source.networkRefs[].labelSelector.matchExpressions[].values[]** (*string*)
- **spec.rules[].destination** (*object, required*) – Destination selector. Specify ipBlocks or networkRefs, not both.
- **spec.rules[].destination.ipBlocks[]** (*object*) – Network locations that belong to a NetworkObjectGroup.
- **spec.rules[].destination.ipBlocks[].description** (*string*)
- **spec.rules[].destination.ipBlocks[].cidrs[]** (*string*)
- **spec.rules[].destination.ipBlocks[].virtualNetwork** (*object*) – Logical network identifiers.
- **spec.rules[].destination.ipBlocks[].virtualNetwork.vrfs[]** (*string*)
- **spec.rules[].destination.ipBlocks[].virtualNetwork.vlans[]** (*integer:int32*)
- **spec.rules[].destination.networkRefs[]** (*object*) – Reference to NetworkObjectGroups in the policy namespace. Specify name or labelSelector, not both.
- **spec.rules[].destination.networkRefs[].name** (*string*)
- **spec.rules[].destination.networkRefs[].labelSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.

- **spec.rules[].destination.networkRefs[].labelSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].key** (*string, required*)
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].operator** (*string:enum, required*)
- **spec.rules[].destination.networkRefs[].labelSelector.matchExpressions[].values[]** (*string*)
- **spec.rules[].destination.protoPorts[]** (*object*) – Transport protocol and optional port or port range.
- **spec.rules[].destination.protoPorts[].protocol** (*string:enum, required*)
- **spec.rules[].destination.protoPorts[].port** (*integer:int32*)
- **spec.rules[].destination.protoPorts[].endPort** (*integer:int32*) – Inclusive end of a port range. Requires port and must be greater than port.
- **clusterSelector** (*object*) – Kubernetes label selector. Requirements are ANDed together.
- **clusterSelector.matchLabels** (*object*) – Kubernetes metadata labels.
- **clusterSelector.matchExpressions[]** (*object*) – Single match expression in a Kubernetes label selector.
- **clusterSelector.matchExpressions[].key** (*string, required*)
- **clusterSelector.matchExpressions[].operator** (*string:enum, required*)
- **clusterSelector.matchExpressions[].values[]** (*string*)
- **version** (*string, required*) – SmartSwitchNetworkPolicy version used for optimistic concurrency.
- **revisionTimestamp** (*string:date-time*) – Timestamp of the SmartSwitchNetworkPolicy revision in this response.
- **deploymentStatus** (*object*) – Aggregated deployment state for a SmartSwitchNetworkPolicy.
- **deploymentStatus.overallStatus** (*string:enum, required*)
- **deploymentStatus.totalAgents** (*integer:int64*)
- **deploymentStatus.inSyncAgents** (*integer:int64*)
- **deploymentStatus.outOfSyncAgents** (*integer:int64*)
- **deploymentStatus.pendingAgents** (*integer:int64*)

Status Codes:

- 200 OK –

  Policy updated

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "namespace": "prod",
  "name": "allow-web-to-checkout",
  "spec": {
    "rules": [
      {
        "action": "allow",
        "source": {
          "networkRefs": [
            {
              "labelSelector": {
                "matchLabels": {
                  "role": "web"
                }
              }
            }
          ]
        },
        "destination": {
          "networkRefs": [
            {
              "name": "checkout-api"
            }
          ],
          "protoPorts": [
            {
              "protocol": "TCP",
              "port": 443
            }
          ]
        }
      }
    ]
  },
  "version": "18"
}
```

- 400 Bad Request – The request is malformed or contains invalid parameters.
- 401 Unauthorized – Authentication is required or the supplied credentials are invalid.
- 403 Forbidden – The authenticated principal is not authorized to perform this operation.
- 404 Not Found – The requested resource was not found.
- 408 Request Timeout – The server timed out waiting for the request.
- 409 Conflict – The requested change conflicts with the current resource.
- 412 Precondition Failed – The supplied resource version precondition does not match the current resource version.
- 428 Precondition Required – The If-Match header is required for this operation.
- 413 Request Entity Too Large – The request body exceeds the configured maximum size.
- 429 Too Many Requests – The request rate limit was exceeded.
- 500 Internal Server Error – An unexpected internal server error occurred.

Response Headers:

- **ETag** (*string*) – Current resource version for optimistic concurrency, encoded as a strong quoted HTTP entity tag such as “42”.

- **Deprecation** (*string*) – Deprecation header as defined by RFC 9745. Date when the API became or will become deprecated, encoded as an HTTP structured field date.
- **Sunset** (*string:http-date*) – Sunset header as defined by RFC 8594. Date when the API is expected to become unavailable.
- **Retry-After** (*string*) – Time to wait before retrying the request.
- **RateLimit** (*string*) – RateLimit header as defined by draft-ietf-httpapi-ratelimit-headers-11. Current service limit available under a specific quota policy.
- **RateLimit-Policy** (*string*) – RateLimit-Policy header as defined by draft-ietf-httpapi-ratelimit-headers-11. Quota policy that client requests consume.

`DELETE /policies/{namespace}/{name}`

Delete a Hypershield policy

Deletes the SmartSwitchNetworkPolicy.

Parameters:

- **namespace** (*string, required*) – Kubernetes namespace that owns the Hypershield resource.
- **name** (*string, required*) – Kubernetes resource name.

Status Codes:

- 204 No Content – Policy deleted
- 400 Bad Request – The request is malformed or contains invalid parameters.
- 401 Unauthorized – Authentication is required or the supplied credentials are invalid.
- 403 Forbidden – The authenticated principal is not authorized to perform this operation.
- 404 Not Found – The requested resource was not found.
- 408 Request Timeout – The server timed out waiting for the request.
- 429 Too Many Requests – The request rate limit was exceeded.
- 500 Internal Server Error – An unexpected internal server error occurred.

Response Headers:

- **Deprecation** (*string*) – Deprecation header as defined by RFC 9745. Date when the API became or will become deprecated, encoded as an HTTP structured field date.
- **Sunset** (*string:http-date*) – Sunset header as defined by RFC 8594. Date when the API is expected to become unavailable.
- **Retry-After** (*string*) – Time to wait before retrying the request.
- **RateLimit** (*string*) – RateLimit header as defined by draft-ietf-httpapi-ratelimit-headers-11. Current service limit available under a specific quota policy.
- **RateLimit-Policy** (*string*) – RateLimit-Policy header as defined by draft-ietf-httpapi-ratelimit-headers-11. Quota policy that client requests consume.

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.