Source: index.md

Refdog

Concepts	**Resources**	**Commands**
Concept overview Concept index	Resource overview Resource index	Command overview Command index

Source: toc.md

Table of Contents

- commands/site/generate.md
- commands/site/index.md
- commands/site/reload.md
- commands/site/start.md
- commands/site/status.md
- commands/site/stop.md
- commands/site/update.md
- commands/system/apply.md
- commands/system/index.md
- commands/system/install.md
- commands/system/reload.md
- commands/system/start.md
- commands/system/status.md
- commands/system/stop.md
- commands/system/uninstall.md
- commands/token/index.md
- commands/token/issue.md
- commands/token/redeem.md
- concepts/access-token.md
- concepts/application.md
- concepts/component.md
- concepts/connector.md
- concepts/index.md
- concepts/link.md
- concepts/listener.md
- concepts/network.md
- concepts/overview.md
- concepts/platform.md
- concepts/routing-key.md
- concepts/site.md
- concepts/workload.md
- resources/access-grant.md
- resources/access-token.md
- resources/attached-connector-binding.md
- resources/attached-connector.md
- resources/connector.md
- resources/index.md
- resources/link.md
- resources/listener.md
- resources/overview.md
- resources/router-access.md
- resources/site.md
- topics/application-tls.md
- topics/attached-connectors.md
- topics/components.md
- topics/controller-configuration.md
- topics/debug-dumps.md
- topics/high-availability.md
- topics/index.md
- topics/individual-pod-services.md
- topics/large-networks.md
- topics/load-balancing.md
- topics/resource-settings.md
- topics/resource-status.md
- topics/router-tls.md
- topics/service-exposure.md
- topics/site-configuration.md
- topics/site-linking.md

- topics/skupper-overview.md
- topics/system-namespaces.md
- topics/system-tls-credentials.md

Source: commands/index.md

Skupper commands

Site operations

Site	Overview of site commands
Site create	Create a site
Site update	Change site settings
Site delete	Delete a site
Site status	Display the status of a site
Site generate	Generate a Site resource

Site linking

Token	Overview of token commands
Token issue	Issue a token file redeemable for a link to the current site
Token redeem	Redeem a token file in order to create a link to a remote site

Link	Overview of link commands
Link update	Change link settings
Link delete	Delete a link
Link status	Display the status of links in the current site
Link generate	Generate a Link resource for use in a remote site

Service exposure

Listener	Overview of listener commands
Listener create	Create a listener
Listener update	Update a listener
Listener delete	Delete a listener
Listener status	Display the status of listeners in the current site
Listener generate	Generate a Listener resource

Connector	Overview of connector commands

Connector create	Create a connector
Connector update	Update a connector
Connector delete	Delete a connector
Connector status	Display the status of connectors in the current site
Connector generate	Generate a Connector resource

System operations

System	Overview of system commands
System install	Install the Skupper components
System uninstall	Remove the Skupper components
System start	Start up the Skupper components for the current site
System stop	Shut down the Skupper components for the current site
System reload	Reload the site configuration
System status	Display the status of the system

Debugging operations

Debug	Overview of debug commands
Debug check	Run diagnostic checks
Debug dump	Generate a debug dump file

Other operations

Version	Display versions of Skupper components

Source: commands/overview.md

Skupper command overview

Skupper uses the `skupper` command as its command-line interface (CLI) for creating and operating Skupper networks.

Capabilities

In its primary role, the Skupper CLI is a thin layer on top of the standard Skupper resources. Its job is to configure sites, listeners, and connectors. It additionally provides commands for site linking, system operation, and troubleshooting.

- **Resource configuration:** Create, update, and delete Skupper resources.
- **Resource status:** Display the current state of Skupper resources.
- **Resource generation:** Produce Skupper resources in YAML or JSON format.

- **Site linking:** Use tokens to set up site-to-site links.
- **System operation:** Install and operate Skupper runtime components.
- **Troubleshooting:** Use debugging tools to identify and fix problems.

By design, the Skupper CLI does not do everything the Skupper resources can do. We encourage you to use the resources directly for advanced use cases.

Usage

```
skupper [command] [subcommand] [options]
```

- `command`: A resource type or functional area.
- `subcommand`: The specific operation you want to perform.
- `options`: Additional arguments that change the operation's behavior.

Context

Skupper commands operate with a current platform and namespace (with a few exceptions). On Kubernetes, there is additionally a current kubeconfig and context. You can use CLI options or environment variables to change the current selection.

Context	Default	CLI option	Environment variable
Platform	`kubernetes`	`--platform`	`SKUPPER_PLATFORM`
Namespace	*From kubeconfig*	`--namespace`	*None*
Kubeconfig context	*From kubeconfig*	`--context`	*None*
Kubeconfig	`~/.kube/config`	`--kubeconfig`	`KUBECONFIG`

On Docker, Podman, and Linux, the current namespace defaults to `default`.

Blocking

On Kubernetes, resource operations block until the desired outcome is achieved, an error occurs, or the timeout is exceeded. You can change the wait condition and the timeout duration using the `--wait` and `--timeout` options.

- Site and link operations block until the resource is ready.
- Listener and connector operations block until the resource is configured.
- All resource delete operations block until deletion is complete.

On Docker, Podman, and Linux, resource operations do not block. Instead, they place the resources in the input location. Changes are applied when the user invokes `skupper system reload`.

Errors

The Skupper CLI returns a non-zero exit code indicating an error when:

- User input is invalid.
- Referenced resources are not found.
- The operation fails or times out.

Resource commands

```
skupper <resource-type> create <resource-name> [options]
skupper <resource-type> update <resource-name> [options]
skupper <resource-type> delete <resource-name> [options]
skupper <resource-type> status [resource-name] [options]
skupper <resource-type> generate <resource-name> [options]
```

These commands operate on Skupper sites, links, listeners, and connectors.

Resource properties are set using one or more `--some-key some-value` command-line options. YAML resource options in camel case (`someKey`) are exposed as hyphenated names (`--some-key`) when used as options.

The `create`, `update`, and `delete` commands control the lifecycle of Skupper resources and configure their properties.

The `status` commands display the current state of resources. If no resource name is specified, they list the status of all resources of the given type.

The `generate` commands produce Skupper resources as YAML or JSON output. They are useful for directing the output to files or other tools.

Token commands

```
skupper token issue <token-file> [options]
skupper token redeem <token-file> [options]
```

These commands use access tokens to create links between sites.

The `token issue` command creates an access token for use in remote sites. The `token redeem` command uses an access token to create a link to the issuing site.

System commands

```
skupper system install [options]
skupper system uninstall [options]
skupper system start [options]
skupper system stop [options]
skupper system reload [options]
skupper system status [options]
```

These commands configure and operate the Skupper runtime components for Docker, Podman, and Linux sites.

Debug commands

```
skupper debug check [options]
skupper debug dump [options]
```

These commands help you troubleshoot problems.

Version command

```
skupper version
```

The `version` command displays the versions of Skupper components.

Source: commands/version.md

Version command

```
skupper version [options]
```

Display versions of Skupper components.

Platforms	Kubernetes, Docker, Podman, Linux

Examples

```
# Show component versions
$ skupper version
COMPONENT         VERSION
cli               2.0.0
controller        2.0.0
router            3.0.0

# Show version details in YAML format
$ skupper version --output yaml
components:
  cli:
    version: 2.0.0
  controller:
    version: 2.0.0
    images:
      controller:
        name: quay.io/skupper/controller:2.0.0
        digest:
sha256:663d97f86ff3fcce27a3842cd2b3a8e32af791598a46d815c07b0aec07505f55
  router:
    version: 3.0.0
    images:
      router:
        name: quay.io/skupper/router:3.0.0
        digest:
sha256:dc5e27385a1e110dd2db1903ba7ec3e0d50b57f742aa02d7dd0a7b1b68c34394
      kube-adaptor:
        name: quay.io/skupper/kube-adaptor:2.0.0
        digest:
sha256:4dc24bb3d605ed3fcec2f8ef7d45ca883d9d87b278bfedd5fcca74281d617a5e
```

Primary options

–output

(-o) <format>

Produce verbose structured output.

Choices	`json` Produce JSON output `yaml` Produce YAML output
Platforms	Kubernetes, Docker, Podman, Linux

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

<table>
<tr><th>Default</th><td><code>kubernetes</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>kubernetes</code></th><td>Kubernetes</td></tr>
<tr><th><code>docker</code></th><td>Docker</td></tr>
<tr><th><code>podman</code></th><td>Podman</td></tr>
<tr><th><code>linux</code></th><td>Linux</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes, Docker, Podman, Linux</td></tr>
<tr><th>See also</th><td>Platform concept</td></tr>
</table>

–help

(-h) boolean
global

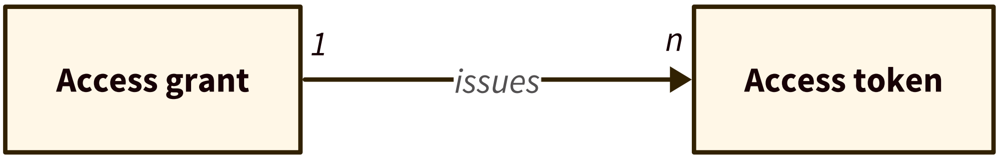
Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

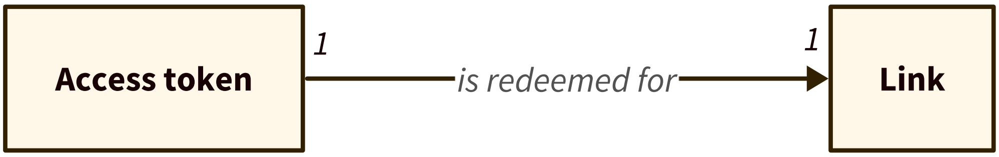
Source: concepts/access-token.md

Access token concept

An access token is a short-lived credential used to create a link. An access token contains the URL and secret code of a corresponding *access grant*.



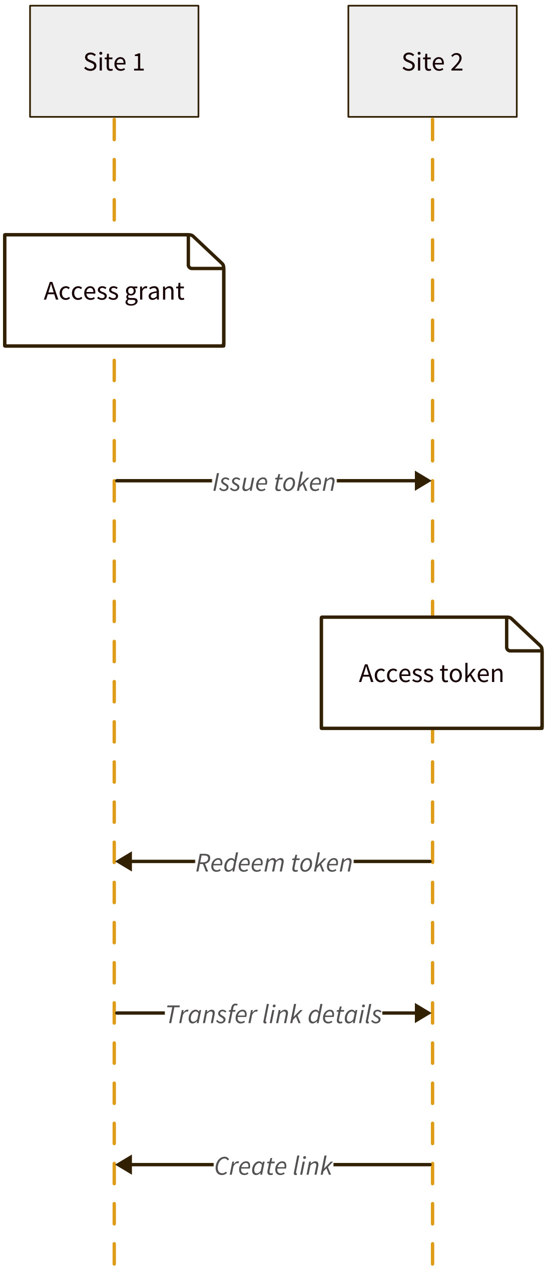
Issuing tokens



Redeeming tokens

Access tokens are issued from access grants. A grant issues zero or more tokens. Tokens are redeemed for links.

Access tokens have limited redemptions and limited lifespans. By default, they can be redeemed only once, and they expire 15 minutes after being issued. You can set custom limits by configuring the access grant.



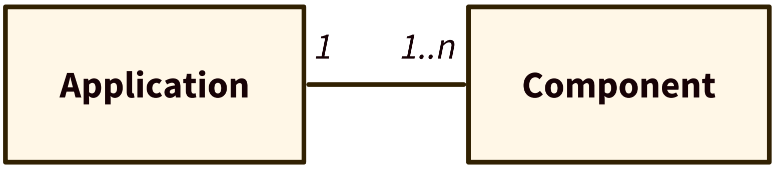
The sequence for issuing and redeeming access tokens

- A site wishing to accept a link (site 1) creates an access grant.
- It uses the access grant to issue a corresponding access token and transfers it to a remote site (site 2).
- Site 2 submits the access token to site 1 for redemption.
- If the token is valid, site 1 sends site 2 the TLS host, port, and credentials required to create a link to site 1.

Source: concepts/application.md

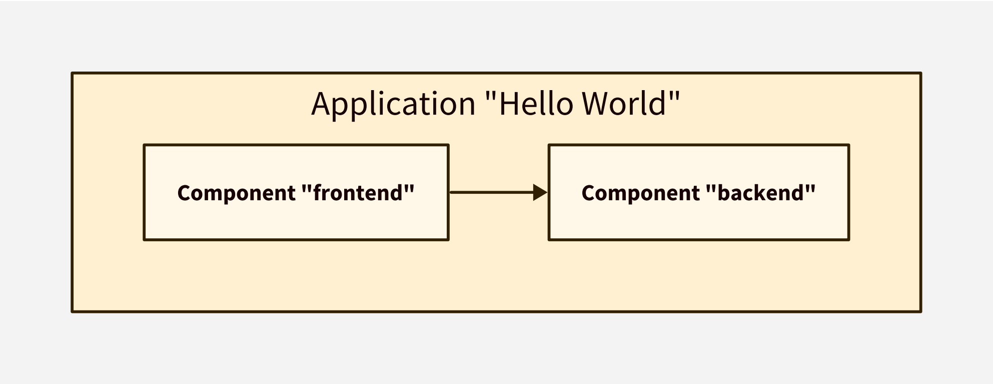
Application concept

An application is a set of components that work together. A Skupper network is dedicated to one application.

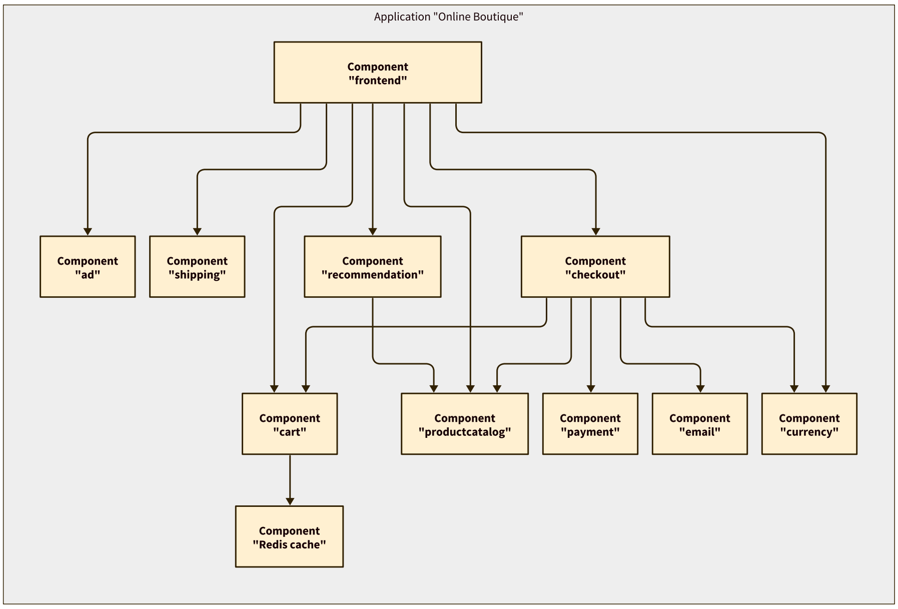


The application model

An application has one or more components.



A simple application with two components

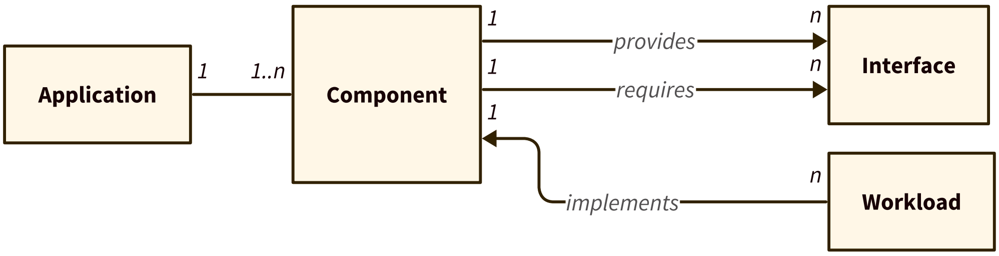


The components of the Online Boutique example application

Source: concepts/component.md

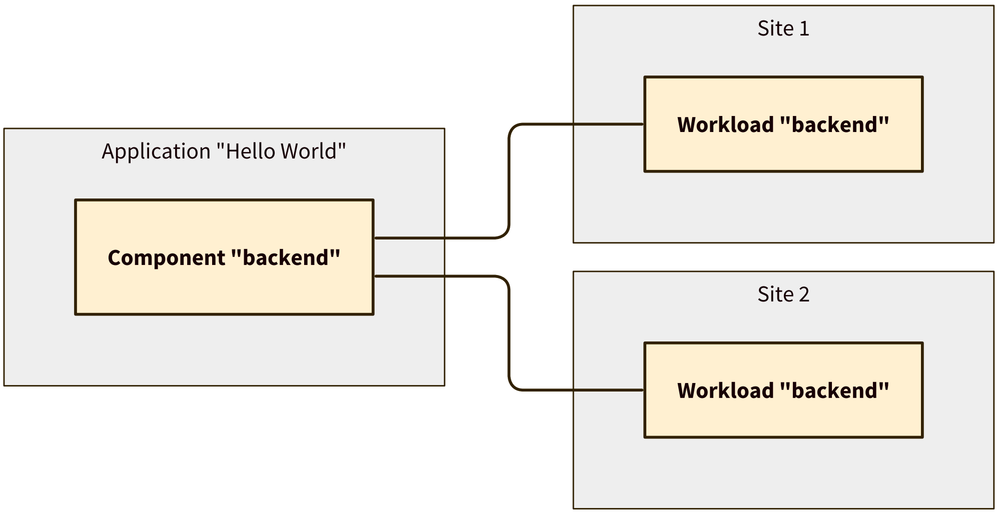
Component concept

A component is a logical part of an application. Each component has a set of responsibilities in achieving the goals of the application. Components provide and require *interfaces* such as REST APIs or database listeners. A component is implemented by workloads.

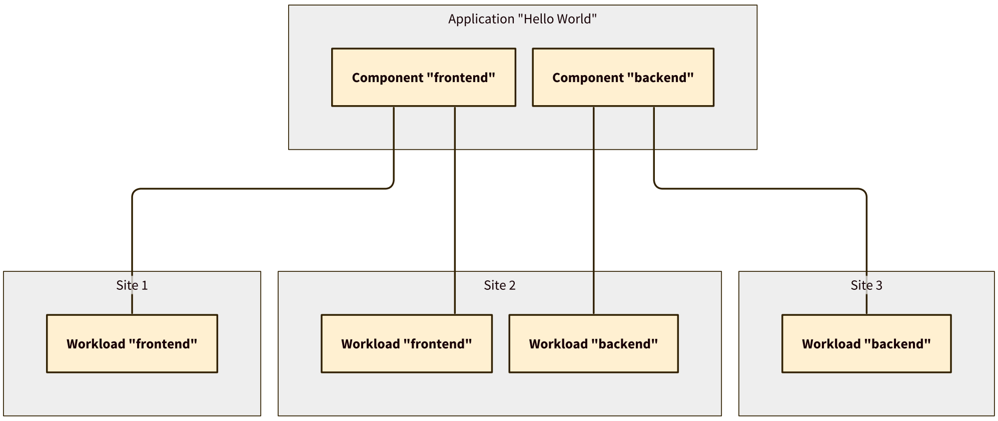


The component model

An application has one or more components. Each component provides and requires zero or more interfaces. Each component is implemented by zero or more workloads.



A component with workloads in two different sites

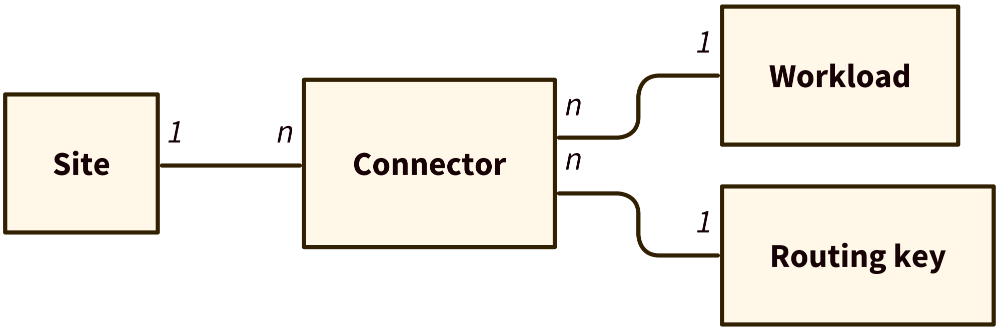


Hello World with its components implemented by workloads in three different sites

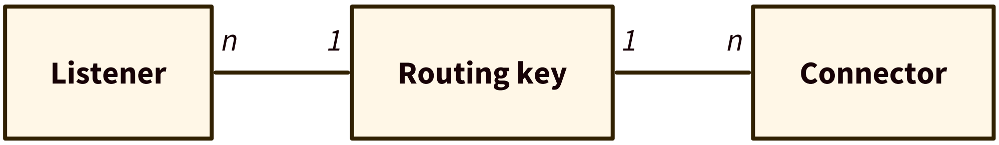
Source: concepts/connector.md

Connector concept

A connector binds a local workload to listeners in remote sites. Listeners and connectors are matched using routing keys.



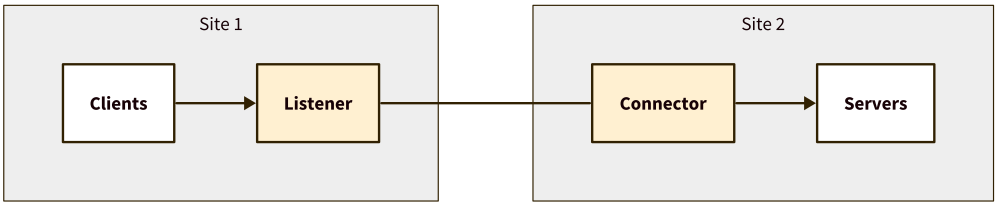
The connector model



The routing key model

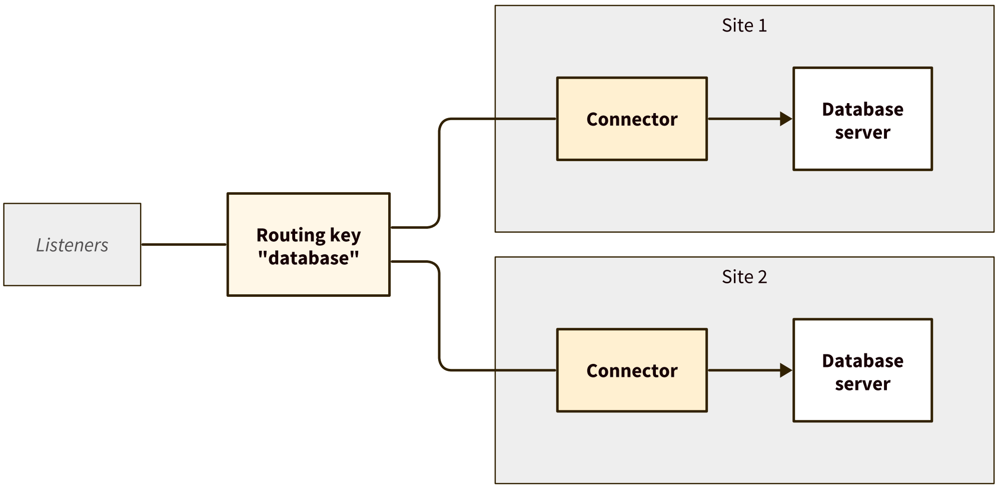
A site has zero or more connectors. Each connector has an associated workload and routing key. The workload can be specified as a Kubernetes pod selector or as the host and port of a local network service. The routing key is a string identifier that binds the connector to listeners in remote sites.

On Kubernetes, the workload is usually specified using a pod selector. On Docker, Podman, and Linux, it is specified using a host and port.



Client connections forwarded to servers

Skupper routers forward client connections across the network from listeners to connectors with matching routing keys. The connectors then forward the client connections to the workload servers.



A database service with connectors in two sites

Source: concepts/index.md

Skupper concepts

Sites

Site	A site is a place on the network where application workloads are running
Workload	A workload is a set of processes running on a platform
Platform	A platform is a system for running application workloads

Networks

Network	A network is a set of sites joined by links
Link	A link is a channel for communication between sites
Access token	An access token is a short-lived credential used to create a link

Services

Listener	A listener binds a local connection endpoint to connectors in remote sites
Connector	A connector binds a local workload to listeners in remote sites
Routing key	A routing key is a string identifier for matching listeners and connectors

Applications

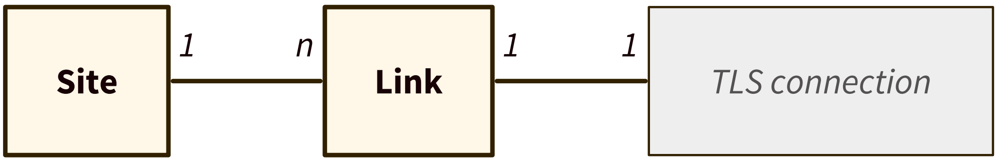
Application	An application is a set of components that work together
Component	A component is a logical part of an application

Source: concepts/link.md

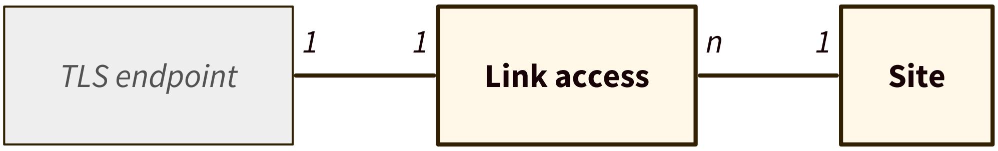
Link concept

A link is a channel for communication between sites. Links carry application connections and requests. A set of linked sites constitutes a network.

To create a link to a remote site, the remote site must enable *link access*. Link access provides an external access point for accepting links.



The link model

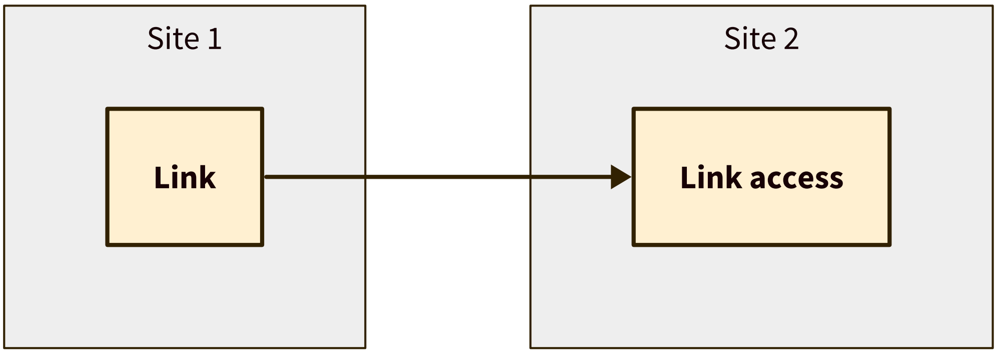


The link access model

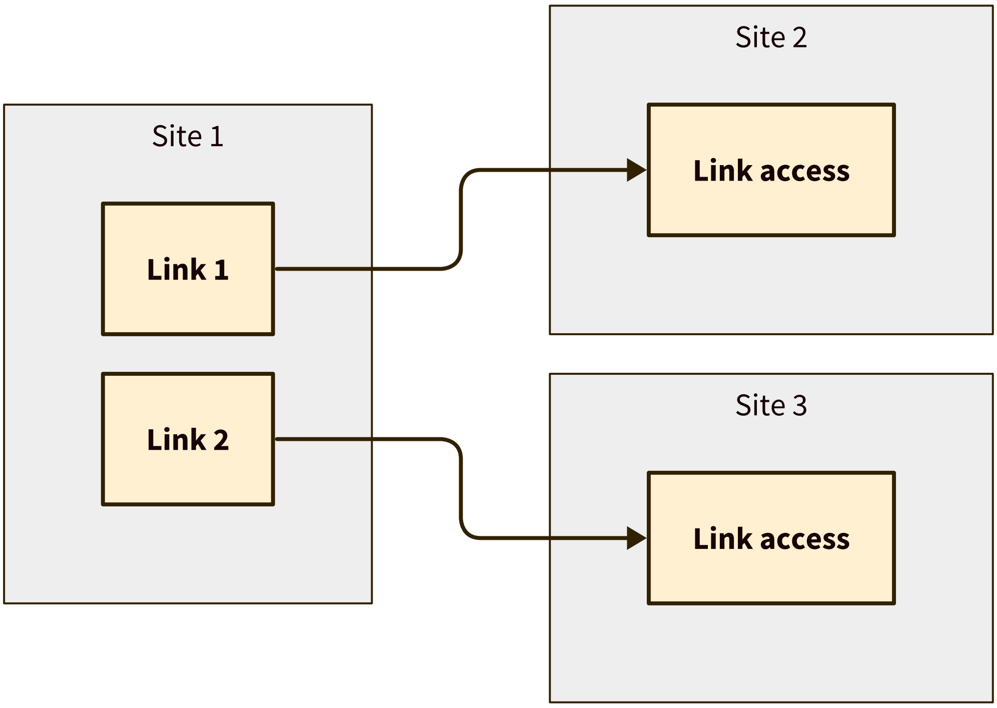
A site has zero or more links. Each link has a host, port, and TLS credentials for making a mutual TLS connection to a remote site. In addition, a site has zero or more link accesses. Usually only one is needed per site. Each link access has a host, port, and TLS credentials for exposing a TLS endpoint that accepts connections from remote sites.

Application connections and requests flow across links in both directions. A linked site can communicate with any other site in the network, even if it is not linked directly. Links can be added and removed dynamically.

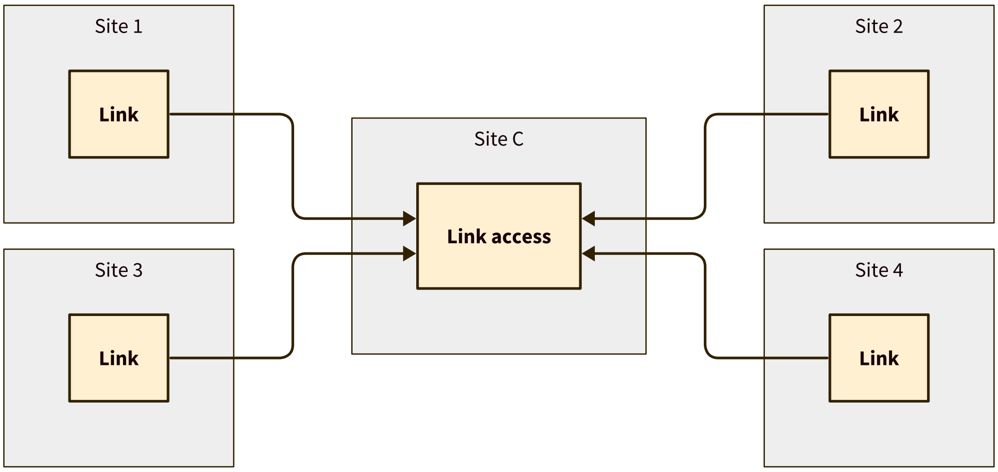
You can use access tokens to securely exchange the connection details required to create a link.



A link joining two sites to create a simple network



A site with two links, to two remote sites

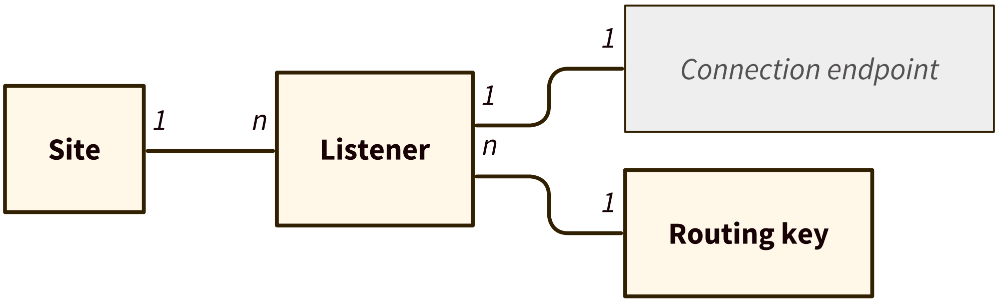


A larger network with links to a central hub site

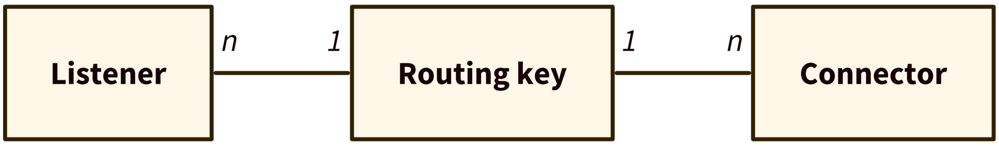
Source: concepts/listener.md

Listener concept

A listener binds a local connection endpoint to connectors in remote sites. Listeners and connectors are matched using routing keys.



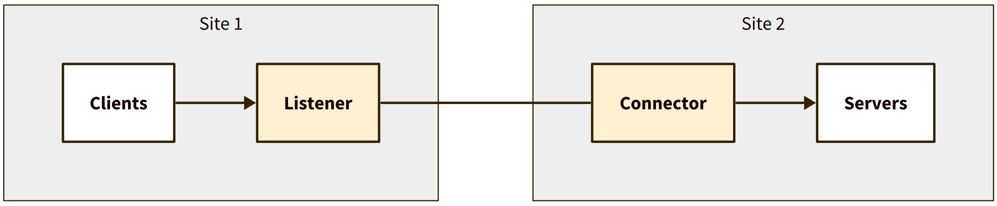
The listener model



The routing key model

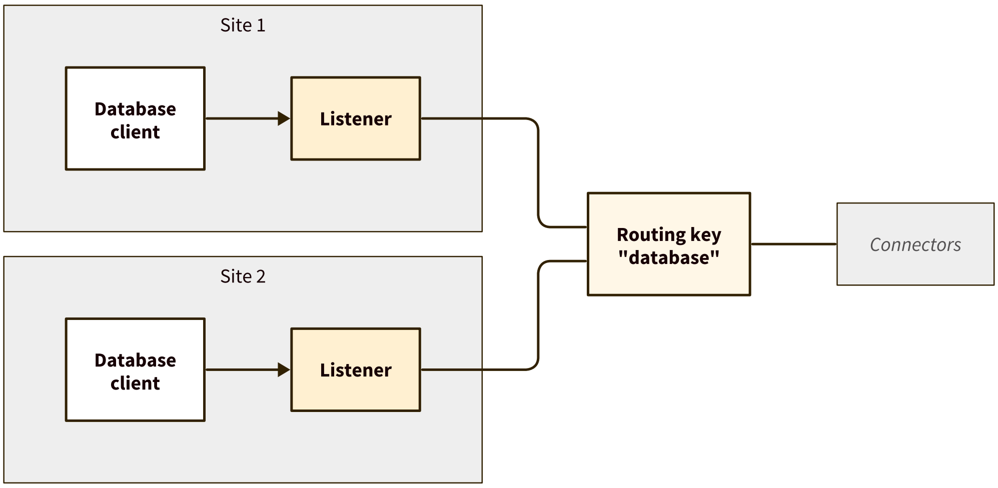
A site has zero or more listeners. Each listener has an associated connection endpoint and routing key. The connection endpoint exposes a host and port for accepting connections from local clients. The routing key is a string identifier that binds the listener to connectors in remote sites.

On Kubernetes, a listener is implemented as a Service. On Docker, Podman, and Linux, it is a listening socket bound to a local network interface.



Client connections forwarded to servers

Skupper routers forward client connections across the network from listeners to connectors with matching routing keys. The connectors then forward the client connections to the workload servers.

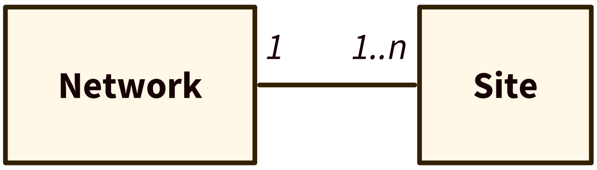


A database service with listeners in two sites

Source: concepts/network.md

Network concept

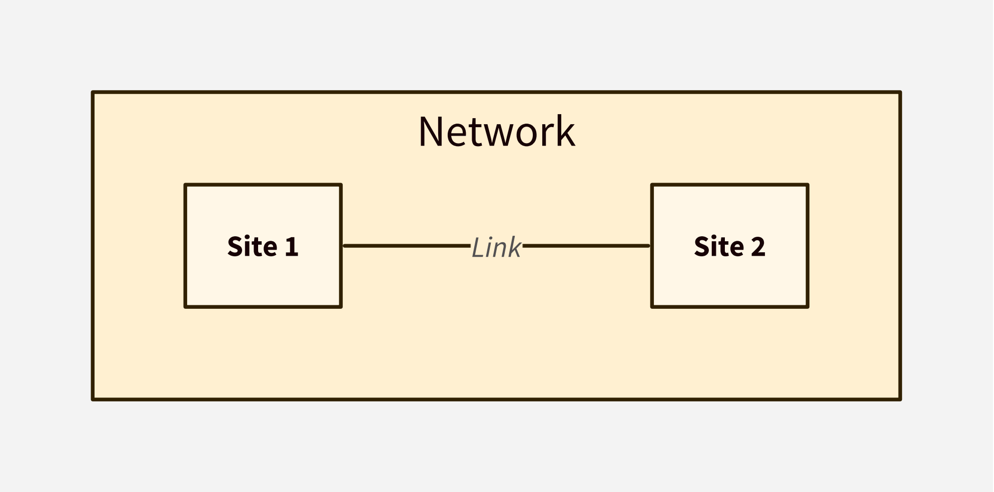
A network is a set of sites joined by links. A Skupper network is also known as an application network or virtual application network (VAN).



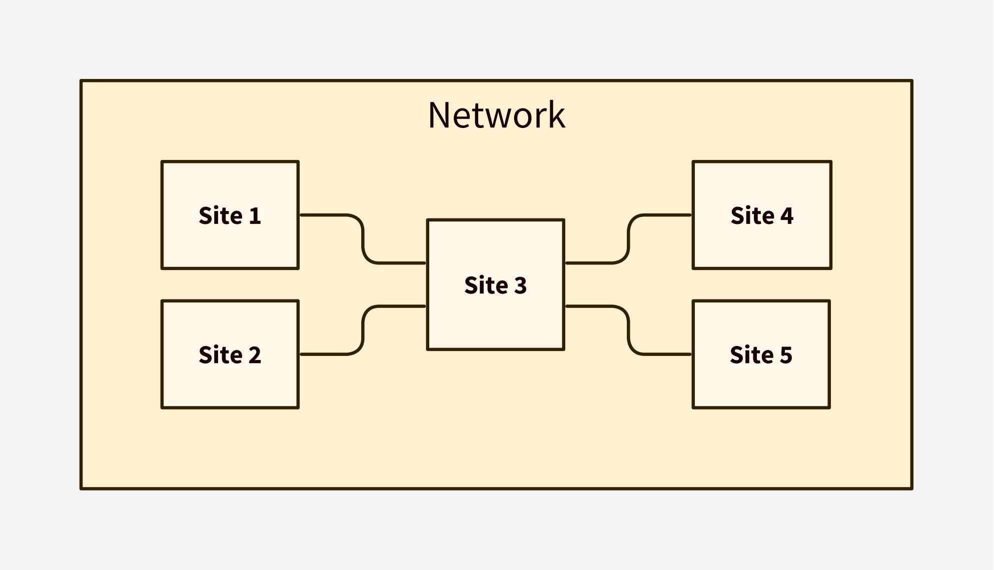
The network model

A network has one or more sites. Each site belongs to only one network.

Each site in the network can expose services to other sites in the network. In turn, each site in the network can access those exposed services. Each network is meant for one distributed application. This provides isolation from other applications and networks.



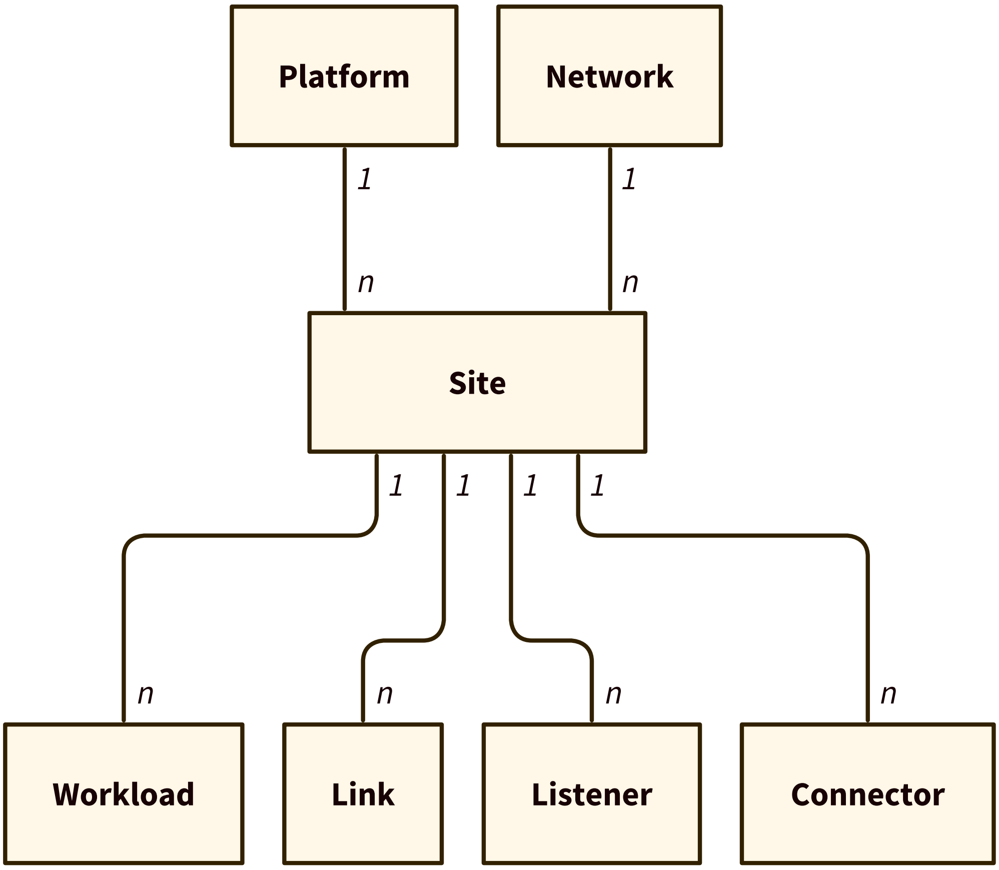
A simple network with two sites



A larger network

Source: concepts/overview.md

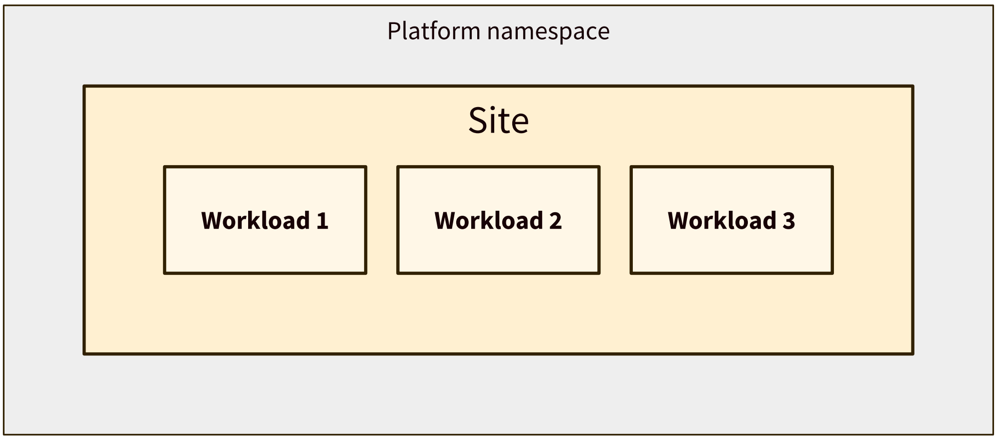
Skupper concept overview



The primary concepts in the Skupper model

Sites

Skupper's job is to provide connectivity for applications that have parts running in multiple locations and on different platforms. A site represents a particular location and a particular platform. It's a place where you have real running workloads. Each site corresponds to one platform namespace, so you can have multiple sites on one platform.

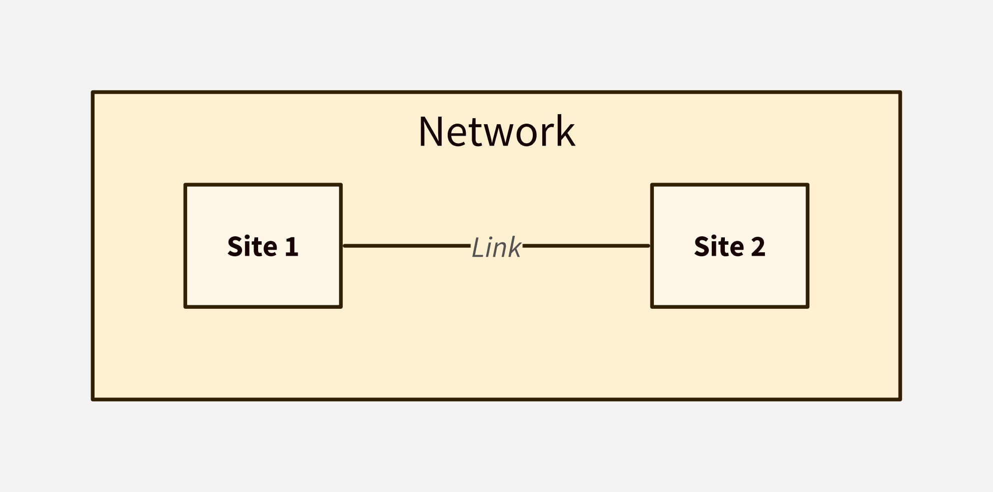


A site with three workloads

Networks

In a distributed application, those workloads need to communicate with other workloads in other sites. Skupper uses links between sites to provide site-to-site communication. Links are always secured using mutual TLS authentication and encryption.

When a set of sites are linked, they function as one application-focused network. You can use short-lived access tokens to securely create links.



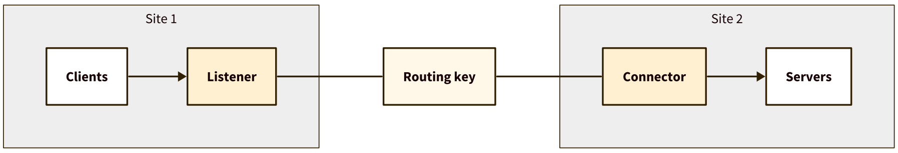
A simple network with two sites

Services

Site-to-site links are distinct from application connections. Links form the transport for your network. Application connections are carried on top of this transport. Application connections can be established in any direction and to any site, regardless of how the underlying links are established.

Services are exposed on the network by creating corresponding listeners and connectors. A listener in one site provides a connection endpoint for client workloads. A connector in another site binds to local server workloads.

The listener and connector are associated using a routing key. Skupper routers use the routing key to forward client connections to the sites where the server workload is running.

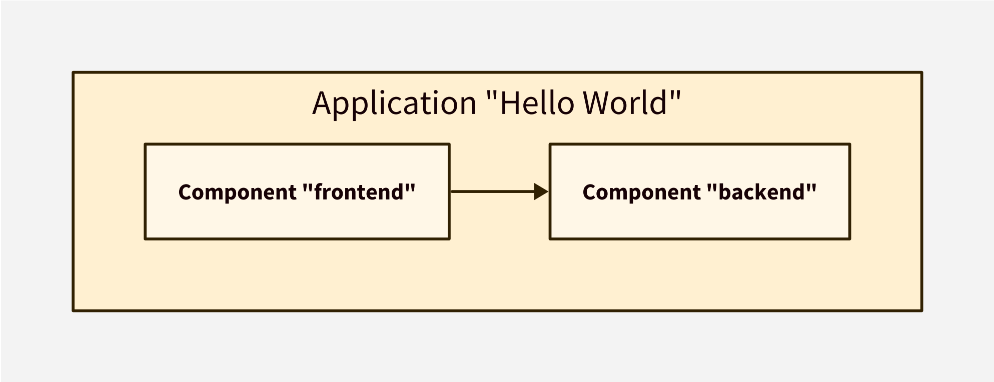


A workload exposed as a service in a remote site

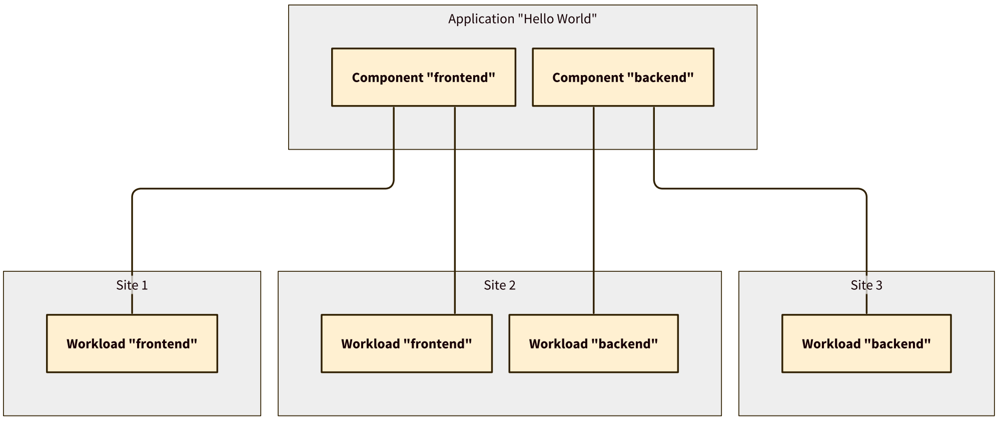
Applications

An application is a set of components that work together to do something useful. A *distributed* application has components that can be deployed as workloads in different locations. Distributed applications are often built with a multitier, service-oriented, or microservices architecture.

Because Skupper makes communication transparent to the application, the location of the running workloads is a concern independent of the application’s design. You can deploy your application workloads to locations that suit you today, and you can safely change to new locations later.



A simple application with two components



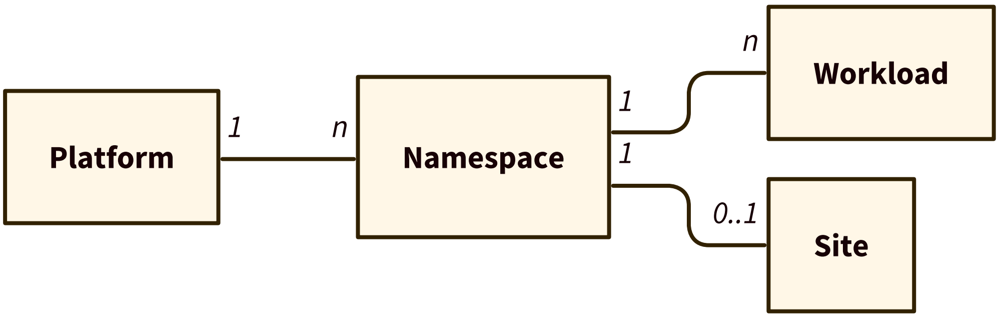
Hello World with its components implemented by workloads in three different sites

Source: concepts/platform.md

Platform concept

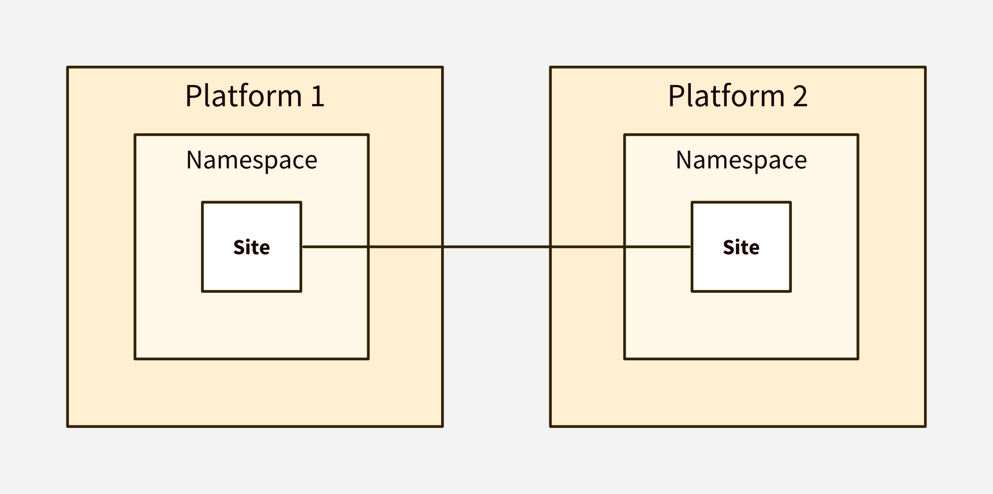
A platform is a system for running application workloads. A platform hosts sites. Skupper supports Kubernetes, Docker, Podman, and Linux. Each site in a network can run on any supported platform.

Platforms provide *namespaces* for related workloads and resources. Skupper uses namespaces to host multiple independent sites on one instance of a platform. Each site on a platform can belong to a distinct application network.

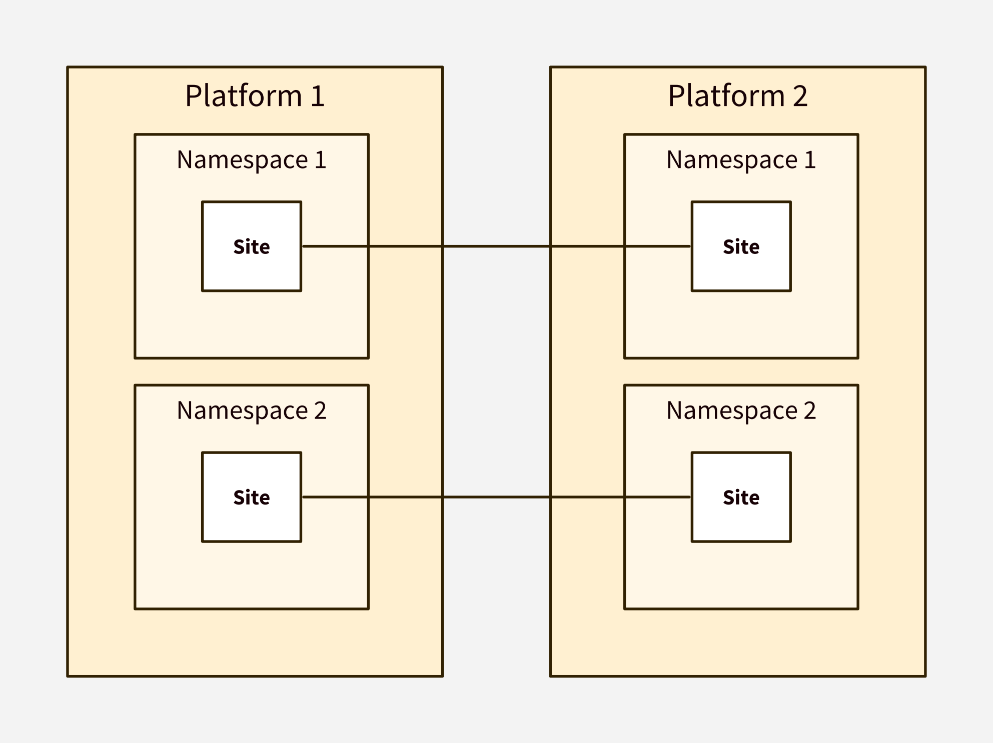


The platform model

A platform has zero or more namespaces. Each namespace is associated with zero or more workloads. A namespace may be associated with a site.



A simple network with sites on two different platforms

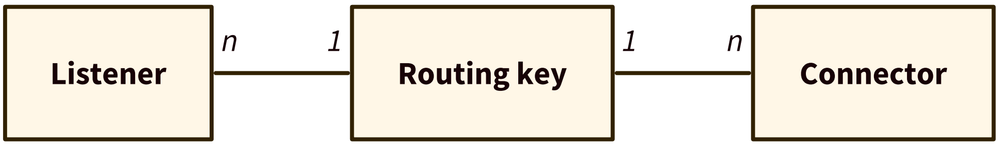


Two different networks spanning two platforms

Source: concepts/routing-key.md

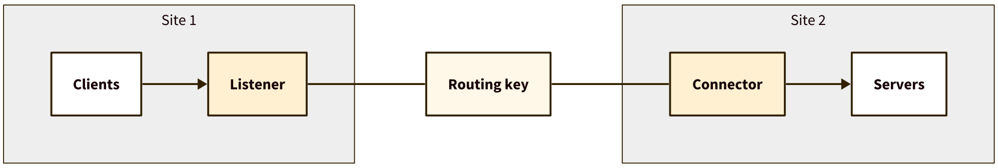
Routing key concept

A routing key is a string identifier for matching listeners and connectors.

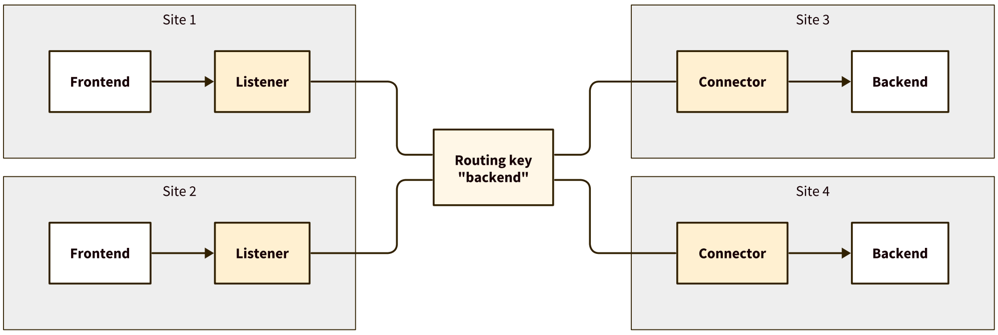


The routing key model

A routing key has zero or more listeners and zero or more connectors. A service is exposed on the application network when it has at least one listener and one connector, matched by routing key.



A workload exposed as a service in a remote site

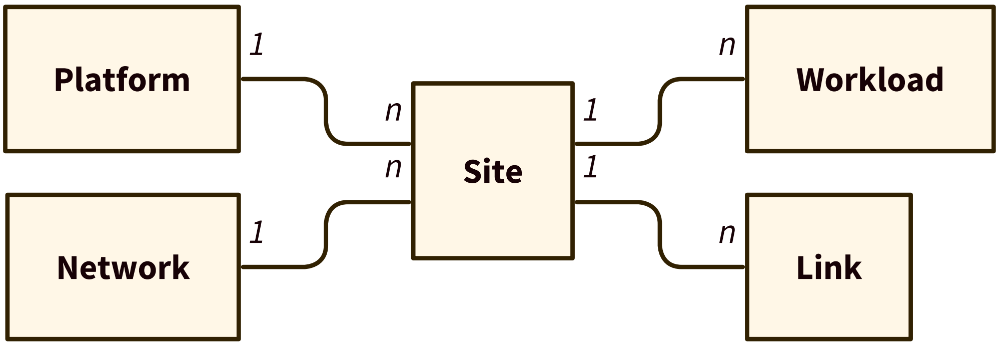


A routing key with two listeners and two connectors

Source: concepts/site.md

Site concept

A site is a place on the network where application workloads are running. Sites are joined by links.

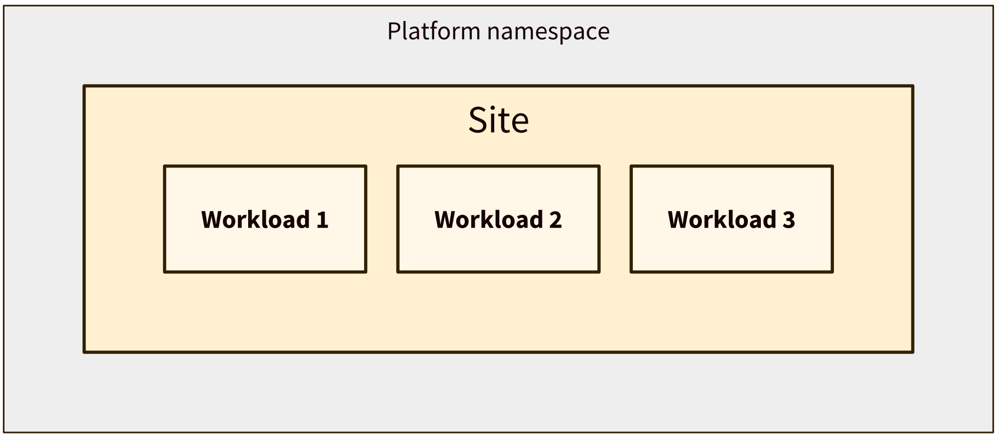


The site model

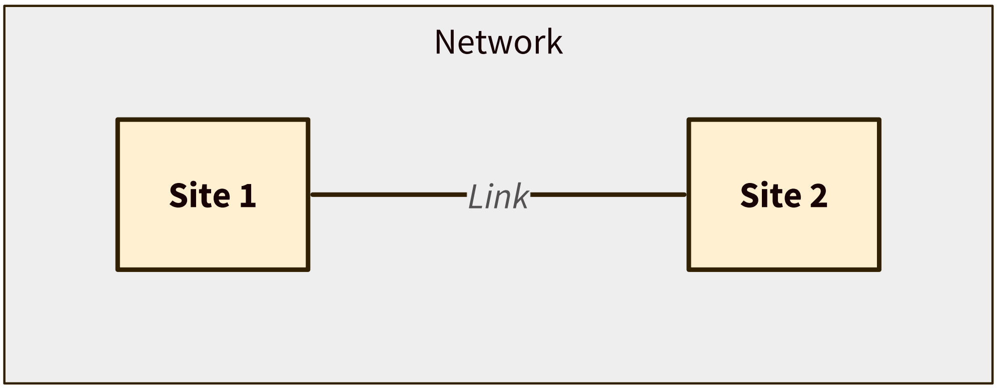
A site is associated with one platform and one network. Each site has zero or more workloads and zero or more links.

Sites operate on multiple platforms. One site corresponds to one namespace in a platform instance. Sites can be added to a network and removed from a network dynamically.

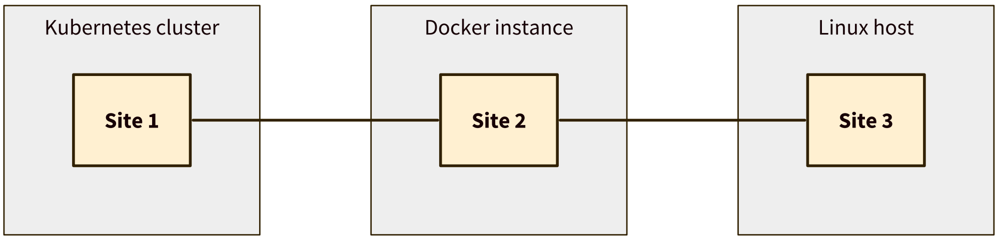
Each site has a Skupper router which is responsible for communicating with the local workloads and forwarding traffic to routers in remote sites.



A site with three workloads



Two sites linked to form a network

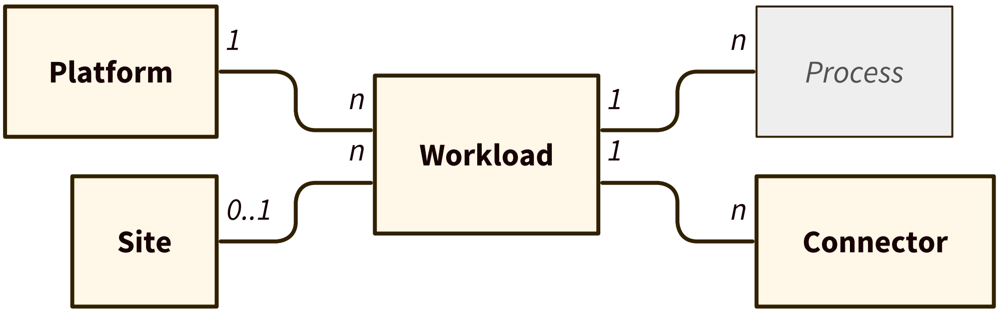


A network with sites on three different platforms

Source: concepts/workload.md

Workload concept

A workload is a set of processes running on a platform. A *process* is a pod, container, or system process. Workloads in a site are exposed as services on the network using connectors.

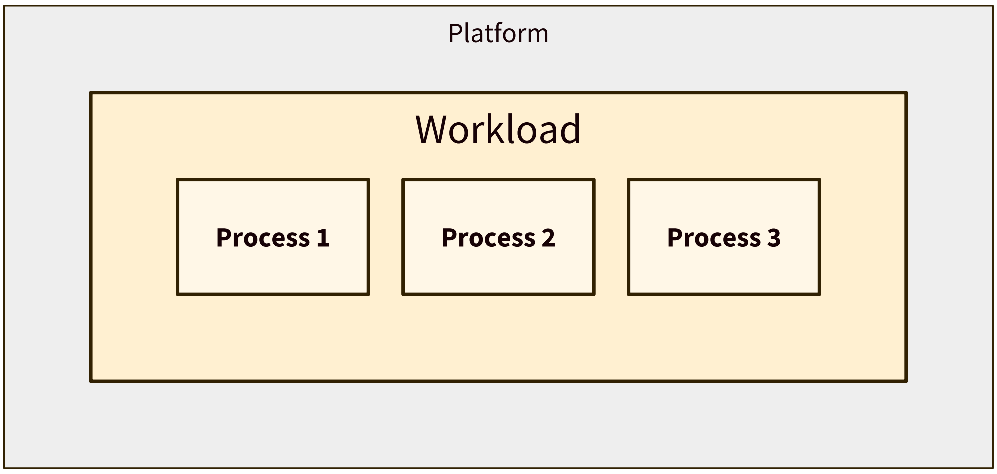


The workload model

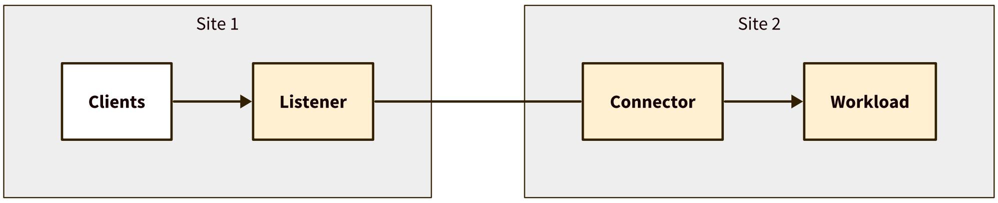
A platform has zero or more workloads. A site also has zero or more workloads. Each workload has zero or more processes and zero or more connectors.

A workload implements one part of an application by providing a network interface (for example, an API) that other parts of the application use. A workload can be both a client and a server.

On Kubernetes, a workload is a Deployment, StatefulSet, or DaemonSet. On Docker or Podman, a workload is a set of containers. On Linux, a workload is a set of system processes.



A workload with three processes



A workload exposed as a service using a connector

Source: resources/access-grant.md

AccessGrant resource

Permission to redeem access tokens for links to the local site. A remote site can use a token containing the grant URL and secret code to obtain a certificate signed by the grant’s certificate authority (CA), within a certain expiration window and for a limited number of redemptions.

The `code`, `url`, and `ca` properties of the resource status are used to generate access tokens from the grant.

Metadata properties

name

string
required

The name of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

namespace

string

The namespace of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept, Kubernetes namespaces, System namespaces

Spec properties

redemptionsAllowed

integer

The number of times an access token for this grant can be redeemed.

Default	1
Platforms	Kubernetes, Docker, Podman, Linux

expirationWindow

string (duration)

The period of time in which an access token for this grant can be redeemed.

Default	`15m`
Platforms	Kubernetes, Docker, Podman, Linux

code

string
advanced

The secret code to use to authenticate access tokens submitted for redemption.

If not set, a value is generated and placed in the `code` status property.

Platforms	Kubernetes, Docker, Podman, Linux

issuer

string
advanced

The name of a Kubernetes secret used to generate a certificate when redeeming a token for this grant.

If not set, `defaultIssuer` on the Site rsource is used.

Platforms	Kubernetes
See also	Router TLS, Kubernetes TLS secrets

settings

object
advanced

A map containing additional settings. Each map entry has a string name and a string value.

Note: In general, we recommend not changing settings from their default values.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource settings

Status properties

status

string

The current state of the resource.

- `Pending`: The resource is being processed.
- `Error`: There was an error processing the resource. See `message` for more information.
- `Ready`: The resource is ready to use.

Platforms	Kubernetes, Docker, Podman, Linux

See also	Resource status

message

string

A human-readable status message. Error messages are reported here.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

redemptions

integer

The number of times a token for this grant has been redeemed.

Platforms	Kubernetes, Docker, Podman, Linux

expirationTime

string (date-time)

The point in time when the grant expires.

Platforms	Kubernetes, Docker, Podman, Linux

url

string

The URL of the token-redemption service for this grant.

Platforms	Kubernetes, Docker, Podman, Linux

ca

string

The trusted server certificate of the token-redemption service for this grant.

Platforms	Kubernetes, Docker, Podman, Linux

code

string

The secret code used to authenticate access tokens submitted for redemption.

Default	*Generated*
Platforms	Kubernetes, Docker, Podman, Linux

conditions

array
advanced

A set of named conditions describing the current state of the resource.

- `Processed`: The controller has accepted the grant.
- `Resolved`: The grant service is available to process tokens for this grant.
- `Ready`: The grant is ready to use. All other conditions are true.

Platforms	Kubernetes
See also	Resource status, Kubernetes conditions

Source: resources/access-token.md

AccessToken resource

A short-lived credential used to create a link. An access token contains the URL and secret code of a corresponding access grant.

Note: Access tokens are often issued and redeemed using the Skupper CLI.

Metadata properties

name

string
required

The name of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

namespace

string

The namespace of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept, Kubernetes namespaces, System namespaces

Spec properties

url

string
required

The URL of the grant service at the remote site.

Platforms	Kubernetes, Docker, Podman, Linux

code

string
required

The secret code used to authenticate the token when submitted for redemption.

Platforms	Kubernetes, Docker, Podman, Linux

ca

string

The trusted server certificate of the grant service at the remote site.

Platforms	Kubernetes, Docker, Podman, Linux

linkCost

integer

The link cost to use when creating the link.

Default	1
Platforms	Kubernetes, Docker, Podman, Linux
See also	Load balancing

settings

object
advanced

A map containing additional settings. Each map entry has a string name and a string value.

Note: In general, we recommend not changing settings from their default values.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource settings

Status properties

redeemed

boolean

True if the token has been redeemed. Once a token is redeemed, it cannot be used again.

Default	False
Platforms	Kubernetes, Docker, Podman, Linux

status

string

The current state of the resource.

- `Pending`: The resource is being processed.
- `Error`: There was an error processing the resource. See `message` for more information.
- `Ready`: The resource is ready to use.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

message

string

A human-readable status message. Error messages are reported here.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

conditions

array
advanced

A set of named conditions describing the current state of the resource.

- `Redeemed`: The token has been exchanged for a link.

Platforms	Kubernetes
See also	Resource status, Kubernetes conditions

Source: resources/attached-connector-binding.md

AttachedConnectorBinding resource

A binding to an attached connector in a peer namespace.

Metadata properties

name

string
required

The name of the resource.

The name must be the same as that of the associated AttachedConnector resource in the connector namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

namespace

string

The namespace of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept, Kubernetes namespaces, System namespaces

Spec properties

connectorNamespace

string
required

The name of the namespace where the associated AttachedConnector is located.

Platforms	Kubernetes, Docker, Podman, Linux

routingKey

string

required

The identifier used to route traffic from listeners to connectors. To expose a local workload to a remote site, the remote listener and the local connector must have matching routing keys.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True
See also	Routing key concept

exposePodsByName

boolean
advanced

If true, expose each pod as an individual service.

Default	False
Platforms	Kubernetes
See also	Individual pod services

settings

object
advanced

A map containing additional settings. Each map entry has a string name and a string value.

Note: In general, we recommend not changing settings from their default values.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource settings

Status properties

status

string

The current state of the resource.

- `Pending`: The resource is being processed.
- `Error`: There was an error processing the resource. See `message` for more information.
- `Ready`: The resource is ready to use.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

hasMatchingListener

boolean

True if there is at least one listener with a matching routing key (usually in a remote site).

Default	False
Platforms	Kubernetes, Docker, Podman, Linux
See also	Routing key concept

conditions

array
advanced

A set of named conditions describing the current state of the resource.

Platforms	Kubernetes
See also	Resource status, Kubernetes conditions

Source: resources/attached-connector.md

AttachedConnector resource

A connector in a peer namespace.

Metadata properties

name

string
required

The name of the resource.

The name must be the same as that of the associated AttachedConnectorBinding resource in the site namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

namespace

string

The namespace of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept, Kubernetes namespaces, System namespaces

Spec properties

siteNamespace

string
required

The name of the namespace in which the site this connector should be attached to is defined.

Platforms	Kubernetes, Docker, Podman, Linux

port

integer
required

The port on the target server to connect to.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

selector

string
required

A Kubernetes label selector for specifying target server pods. It uses `<label-name>=<label-value>` syntax.

On Kubernetes, either `selector` or `host` is required.

Platforms	Kubernetes
Updatable	True
See also	Kubernetes label selectors

includeNotReadyPods

boolean
advanced

If true, include server pods in the `NotReady` state.

Default	False

Platforms	Kubernetes

tlsCredentials

string
advanced

The name of a bundle of TLS certificates used for secure router-to-server communication. The bundle contains the trusted server certificate (usually a CA). It optionally includes a client certificate and key for mutual TLS.

On Kubernetes, the value is the name of a Secret in the current namespace. On Docker, Podman, and Linux, the value is the name of a directory under `input/certs/` in the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Application TLS, Kubernetes TLS secrets, System TLS credentials

settings

object
advanced

A map containing additional settings. Each map entry has a string name and a string value.

Note: In general, we recommend not changing settings from their default values.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource settings

Status properties

status

string

The current state of the resource.

- `Pending`: The resource is being processed.
- `Error`: There was an error processing the resource. See `message` for more information.
- `Ready`: The resource is ready to use.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

conditions

array
advanced

A set of named conditions describing the current state of the resource.

Platforms	Kubernetes
See also	Resource status, Kubernetes conditions

selectedPods

array
advanced

Platforms	Kubernetes, Docker, Podman, Linux

Source: resources/connector.md

Connector resource

A connector binds a local workload to listeners in remote sites. Listeners and connectors are matched by routing key.

On Kubernetes, a Connector resource has a selector and port for specifying workload pods.

On Docker, Podman, and Linux, a Connector resource has a host and port for specifying a local server. Optionally, Kubernetes can also use a host and port.

Examples

A connector in site East for the Hello World backend service:

```yaml
apiVersion: skupper.io/v2alpha1
kind: Connector
metadata:
  name: backend
  namespace: hello-world-east
spec:
  routingKey: backend
  selector: app=backend
  port: 8080
```

Metadata properties

name

string

required

The name of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

namespace

string

The namespace of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept, Kubernetes namespaces, System namespaces

Spec properties

routingKey

string
required

The identifier used to route traffic from listeners to connectors. To expose a local workload to a remote site, the remote listener and the local connector must have matching routing keys.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True
See also	Routing key concept

port

integer
required

The port on the target server to connect to.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

selector

string
frequently used

A Kubernetes label selector for specifying target server pods. It uses `<label-name>=<label-value>` syntax.

On Kubernetes, either `selector` or `host` is required.

Platforms	Kubernetes
Updatable	True
See also	Kubernetes label selectors

host

string
frequently used

The hostname or IP address of the server. This is an alternative to `selector` for specifying the target server.

On Kubernetes, either `selector` or `host` is required.

On Docker, Podman, or Linux, `host` is required.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

includeNotReadyPods

boolean
advanced

If true, include server pods in the `NotReady` state.

Default	False
Platforms	Kubernetes

exposePodsByName

boolean
advanced

If true, expose each pod as an individual service.

Default	False
Platforms	Kubernetes
See also	Individual pod services

tlsCredentials

string
advanced

The name of a bundle of TLS certificates used for secure router-to-server communication. The bundle contains the trusted server certificate (usually a CA). It optionally includes a client certificate and key for mutual TLS.

On Kubernetes, the value is the name of a Secret in the current namespace. On Docker, Podman, and Linux, the value is the name of a directory under `input/certs/` in the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Application TLS, Kubernetes TLS secrets, System TLS credentials

useClientCert

boolean
advanced

Send the client certificate when connecting in order to enable mutual TLS.

Default	False
Platforms	Kubernetes, Docker, Podman, Linux
See also	Application TLS

verifyHostname

boolean
advanced

If true, require that the hostname of the server connected to matches the hostname in the server’s certificate.

Default	False
Platforms	Kubernetes, Docker, Podman, Linux
See also	Application TLS

settings

object
advanced

A map containing additional settings. Each map entry has a string name and a string value.

Note: In general, we recommend not changing settings from their default values.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource settings

Status properties

status

string

The current state of the resource.

- `Pending`: The resource is being processed.
- `Error`: There was an error processing the resource. See `message` for more information.
- `Ready`: The resource is ready to use.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

message

string

A human-readable status message. Error messages are reported here.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

hasMatchingListener

boolean

True if there is at least one listener with a matching routing key (usually in a remote site).

Default	False
Platforms	Kubernetes, Docker, Podman, Linux
See also	Routing key concept

conditions

array
advanced

A set of named conditions describing the current state of the resource.

- `Configured`: The connector configuration has been applied to the router.
- `Matched`: There is at least one listener corresponding to this connector.
- `Ready`: The connector is ready to use. All other conditions are true.

Platforms	Kubernetes, Docker, Podman, Linux

selectedPods

array
advanced

Platforms	Kubernetes, Docker, Podman, Linux

Source: resources/index.md

Skupper resources

Primary resources

Site	A site is a place on the network where application workloads are running
Link	A link is a channel for communication between sites
Listener	A listener binds a local connection endpoint to connectors in remote sites
Connector	A connector binds a local workload to listeners in remote sites

Sites and site linking

Site	A site is a place on the network where application workloads are running
Link	A link is a channel for communication between sites
AccessGrant	Permission to redeem access tokens for links to the local site
AccessToken	A short-lived credential used to create a link
RouterAccess	Configuration for secure access to the site router

Service exposure

Listener	A listener binds a local connection endpoint to connectors in remote sites
Connector	A connector binds a local workload to listeners in remote sites
AttachedConnector	A connector in a peer namespace
AttachedConnectorBinding	A binding to an attached connector in a peer namespace

Source: resources/link.md

Link resource

A link is a channel for communication between sites. Links carry application connections and requests. A set of linked sites constitutes a network.

A Link resource specifies remote connection endpoints and TLS credentials for establishing a mutual TLS connection to a remote site. To create an active link, the remote site must first enable *link access*. Link access provides an external access point for accepting links.

Note: Links are not usually created directly. Instead, you can use an access token to obtain a link.

Metadata properties

name

string
required

The name of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

namespace

string

The namespace of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept, Kubernetes namespaces, System namespaces

Spec properties

endpoints

array
required

An array of connection endpoints. Each item has a name, host, port, and group.

Platforms	Kubernetes, Docker, Podman, Linux

cost

integer

The configured routing cost of sending traffic over the link.

Default	1
Platforms	Kubernetes, Docker, Podman, Linux
See also	Load balancing

tlsCredentials

string

The name of a bundle of certificates used for mutual TLS router-to-router communication. The bundle contains the client certificate and key and the trusted server certificate (usually a CA).

On Kubernetes, the value is the name of a Secret in the current namespace.

On Docker, Podman, and Linux, the value is the name of a directory under `input/certs/` in the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Router TLS, Kubernetes TLS secrets, System TLS credentials

settings

object
advanced

A map containing additional settings. Each map entry has a string name and a string value.

Note: In general, we recommend not changing settings from their default values.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource settings

Status properties

status

string

The current state of the resource.

- `Pending`: The resource is being processed.
- `Error`: There was an error processing the resource. See `message` for more information.
- `Ready`: The resource is ready to use.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

message

string

A human-readable status message. Error messages are reported here.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

remoteSiteId

string

The unique ID of the site linked to.

Platforms	Kubernetes, Docker, Podman, Linux

remoteSiteName

string

The name of the site linked to.

Platforms	Kubernetes, Docker, Podman, Linux

conditions

array
advanced

A set of named conditions describing the current state of the resource.

- `Configured`: The link configuration has been applied to the router.
- `Operational`: The link to the remote site is active.
- `Ready`: The link is ready to use. All other conditions are true.

Platforms	Kubernetes
See also	Resource status, Kubernetes conditions

Source: resources/listener.md

Listener resource

A listener binds a local connection endpoint to connectors in remote sites. Listeners and connectors are matched by routing key.

A Listener resource specifies a host and port for accepting connections from local clients. To expose a multi-port service, create multiple listeners with the same host value.

Examples

A listener in site West for the Hello World backend service in site East:

```
apiVersion: skupper.io/v2alpha1
kind: Listener
```

```yaml
metadata:
  name: backend
  namespace: hello-world-west
spec:
  routingKey: backend
  host: backend
  port: 8080
```

Metadata properties

name

string
required

The name of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

namespace

string

The namespace of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept, Kubernetes namespaces, System namespaces

Spec properties

routingKey

string
required

The identifier used to route traffic from listeners to connectors. To enable connecting to a service at a remote site, the local listener and the remote connector must have matching routing keys.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True
See also	Routing key concept

host

string
required

The hostname or IP address of the local listener. Clients at this site use the listener host and port to establish connections to the remote service.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

port

integer
required

The port of the local listener. Clients at this site use the listener host and port to establish connections to the remote service.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

exposePodsByName

boolean
advanced

If true, expose each pod as an individual service.

Default	False
Platforms	Kubernetes
See also	Individual pod services

tlsCredentials

string
advanced

The name of a bundle of TLS certificates used for secure client-to-router communication. The bundle contains the server certificate and key. It optionally includes the trusted client certificate (usually a CA) for mutual TLS.

On Kubernetes, the value is the name of a Secret in the current namespace. On Docker, Podman, and Linux, the value is the name of a directory under `input/certs/` in the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Application TLS, Kubernetes TLS secrets, System TLS credentials

settings

object
advanced

A map containing additional settings. Each map entry has a string name and a string value.

Note: In general, we recommend not changing settings from their default values.

- `observer`: Set the protocol observer used to generate traffic metrics.
  Default: `auto`. Choices: `auto`, `none`, `http1`, `http2`.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource settings

Status properties

status

string

The current state of the resource.

- `Pending`: The resource is being processed.
- `Error`: There was an error processing the resource. See `message` for more information.
- `Ready`: The resource is ready to use.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

message

string

A human-readable status message. Error messages are reported here.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

hasMatchingConnector

boolean

True if there is at least one connector with a matching routing key (usually in a remote site).

Default	False
Platforms	Kubernetes, Docker, Podman, Linux
See also	Routing key concept

conditions

array
advanced

A set of named conditions describing the current state of the resource.

- `Configured`: The listener configuration has been applied to the router.
- `Matched`: There is at least one connector corresponding to this listener.
- `Ready`: The listener is ready to use. All other conditions are true.

Platforms	Kubernetes
See also	Resource status, Kubernetes conditions

Source: resources/overview.md

Skupper resource overview

Skupper provides custom resource definitions (CRDs) that define the API for configuring and deploying Skupper networks. Skupper uses custom resources not only for Kubernetes but also for Docker, Podman, and Linux. The Skupper resources are designed to provide a uniform declarative interface that simplifies automation and supports integration with other tools.

Capabilities

- **Site configuration:** Create and update Skupper sites.
- **Site linking:** Create and update site-to-site links.
- **Service exposure:** Expose application workloads on Skupper networks.

Controller

The Skupper controller is responsible for taking the desired state expressed in your Skupper custom resources and producing a corresponding runtime state. It does this by generating platform-specific output resources that configure the local site and router.

For example, a Site input resource on Kubernetes results in the following output resources:

- A Deployment and ConfigMap for the router
- A ServiceAccount, Role, and RoleBinding for running site components
- A Secret containing a signing CA for site linking

Operations

On Kubernetes:

- *Create and update:* `kubectl apply -f <yaml-file>`
- *Delete:* `kubectl delete -f <yaml-file>`

On Docker, Podman, and Linux:

- *Create and update:* `skupper system apply -f <yaml-file>`
- *Delete:* `skupper system delete -f <yaml-file>`

On Docker, Podman, and Linux, resources are stored on the local filesystem under `~/.local/share/skupper/namespaces/default/input/resources`.

The Skupper CLI provides additional commands to help create and configure Skupper resources.

Common properties

- spec.settings
- spec.tlsCredentials
- status.Status
- status.Message
- status.Conditions

Labels and annotations

{{lipsum()}}

Primary resources

- **Site:** {{lipsum(10)}}
- **Link:** {{lipsum(10)}}
- **Listener:** {{lipsum(10)}}
- **Connector:** {{lipsum(10)}}

These are the main resources you interact with. The others are for less common scenarios.

Site is the heart of things. The Site resource represents a location in a Skupper network. It carries all the configuration for the site. The starting point. The parent of other Skupper resources.

Links…. The Link resource is usually not created directly. Instead, you use access tokens.

Listener and connector are the key resources for service exposure.

Site linking resources

- **Link:** {{lipsum(10)}}
- **AccessGrant:** {{lipsum(10)}}
- **AccessToken:** {{lipsum(10)}}
- **RouterAccess:** {{lipsum(10)}}

You may want to use the CLI (or some other automation) to do the linking part.

Service exposure resources

- **Listener:** {{lipsum(10)}}
- **Connector:** {{lipsum(10)}}
- **AttachedConnector:** {{lipsum(10)}}
- **AttachedConnectorBinding:** {{lipsum(10)}}

Source: resources/router-access.md

RouterAccess resource

Configuration for secure access to the site router. The configuration includes TLS credentials and router ports. The RouterAccess resource is used to implement link access for sites.

Metadata properties

name

string
required

The name of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

namespace

string

The namespace of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept, Kubernetes namespaces, System namespaces

Spec properties

roles

array
required

The named interfaces by which a router can be accessed. These include “inter-router” for links between interior routers and “edge” for links from edge routers to interior routers.

Platforms	Kubernetes, Docker, Podman, Linux

tlsCredentials

string
required

The name of a bundle of TLS certificates used for mutual TLS router-to-router communication. The bundle contains the server certificate and key and the trusted client certificate (usually a CA).

On Kubernetes, the value is the name of a Secret in the current namespace.

On Docker, Podman, and Linux, the value is the name of a directory under `input/certs/` in the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Router TLS, Kubernetes TLS secrets, System TLS credentials

generateTlsCredentials

boolean

Default	False
Platforms	Kubernetes, Docker, Podman, Linux

issuer

string

Platforms	Kubernetes, Docker, Podman, Linux

accessType

string

<table>
<tr><th>Default</th><td><em>On OpenShift, the default is <code>route</code>. For other Kubernetes flavors, the default is <code>loadbalancer</code>.</em></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>route</code></th><td>Use an OpenShift route. <em>OpenShift only.</em></td></tr>
<tr><th><code>loadbalancer</code></th><td>Use a Kubernetes load balancer.</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes</td></tr>
</table>

bindHost

string

The hostname or IP address of the network interface to bind to. By default, Skupper binds all the interfaces on the host.

Default	`0.0.0.0`
Platforms	Docker, Podman, Linux

subjectAlternativeNames

array

The hostnames and IPs secured by the router TLS certificate.

Default	*The current hostname and the IP address of each bound network interface*
Platforms	Docker, Podman, Linux

settings

object
advanced

A map containing additional settings. Each map entry has a string name and a string value.

Note: In general, we recommend not changing settings from their default values.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource settings

Status properties

status

string

The current state of the resource.

- `Pending`: The resource is being processed.
- `Error`: There was an error processing the resource. See `message` for more information.
- `Ready`: The resource is ready to use.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

message

string

A human-readable status message. Error messages are reported here.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

conditions

array
advanced

A set of named conditions describing the current state of the resource.

- `Configured`: The router access configuration has been applied to the router.
- `Resolved`: The connection endpoints are available.
- `Ready`: The router access is ready to use. All other conditions are true.

Platforms	Kubernetes
See also	Resource status, Kubernetes conditions

endpoints

array
advanced

An array of connection endpoints. Each item has a name, host, port, and group.

Platforms	Kubernetes, Docker, Podman, Linux

Source: resources/site.md

Site resource

A site is a place on the network where application workloads are running. Sites are joined by links.

The Site resource is the basis for site configuration. It is the parent of all Skupper resources in its namespace. There can be only one active Site resource per namespace.

Examples

A minimal site:

```
apiVersion: skupper.io/v2alpha1
kind: Site
metadata:
  name: east
  namespace: hello-world-east
```

A site configured to accept links:

```
apiVersion: skupper.io/v2alpha1
kind: Site
```

```
metadata:
  name: west
  namespace: hello-world-west
spec:
  linkAccess: default
```

Metadata properties

name

string
required

The name of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

namespace

string

The namespace of the resource.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept, Kubernetes namespaces, System namespaces

Spec properties

linkAccess

string
frequently used

Configure external access for links from remote sites.

Sites and links are the basis for creating application networks. In a simple two-site network, at least one of the sites must have link access enabled.

Default	none	
Choices	**none**	No linking to this site is permitted.

`default`	Use the default link access for the current platform. On OpenShift, the default is `route`. For other Kubernetes flavors, the default is `loadbalancer`.
`route`	Use an OpenShift route. *OpenShift only.*
`loadbalancer`	Use a Kubernetes load balancer.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True
See also	Link concept, Site linking

ha

boolean

Configure the site for high availability (HA). HA sites have two active routers.

Note that Skupper routers are stateless, and they restart after failure. This already provides a high level of availability. Enabling HA goes further and reduces the window of downtime caused by restarts.

Default	False
Platforms	Kubernetes
Updatable	True
See also	High availability

defaultIssuer

string
advanced

The name of a Kubernetes secret containing the signing CA used to generate a certificate from a token. A secret is generated if none is specified.

This issuer is used by AccessGrant and RouterAccess if a specific issuer is not set.

Default	`skupper-site-ca`
Platforms	Kubernetes
Updatable	True
See also	Router TLS, Kubernetes TLS secrets

edge

boolean
advanced

Configure the site to operate in edge mode. Edge sites cannot accept links from remote sites.

Edge mode can help you scale your network to large numbers of sites. However, for networks with 16 or fewer sites, there is little benefit.

Currently, edge sites cannot also have HA enabled.

Default	False
Platforms	Kubernetes, Docker, Podman, Linux
See also	Large networks

serviceAccount

string
advanced

The name of the Kubernetes service account under which to run the Skupper router. A service account is generated if none is specified.

Default	*Generated*
Platforms	Kubernetes
See also	Kubernetes service accounts

settings

object
advanced

A map containing additional settings. Each map entry has a string name and a string value.

Note: In general, we recommend not changing settings from their default values.

- `routerDataConnections`: Set the number of data connections the router uses when linking to other routers.
  Default: *Computed based on the number of router worker threads. Minimum 2.*
- `routerLogging`: Set the router logging level.
  Default: `info`. Choices: `info`, `warning`, `error`.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource settings

Status properties

status

string

The current state of the resource.

- `Pending`: The resource is being processed.
- `Error`: There was an error processing the resource. See `message` for more information.
- `Ready`: The resource is ready to use.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

message

string

A human-readable status message. Error messages are reported here.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Resource status

conditions

array
advanced

A set of named conditions describing the current state of the resource.

- `Configured`: The output resources for this resource have been created.
- `Running`: There is at least one router pod running.
- `Resolved`: The hostname or IP address for link access is available.
- `Ready`: The site is ready for use. All other conditions are true.

Platforms	Kubernetes
See also	Resource status, Kubernetes conditions

defaultIssuer

string
advanced

The name of the Kubernetes secret containing the active default signing CA.

Platforms	Kubernetes
See also	Router TLS, Kubernetes TLS secrets

endpoints

array
advanced

An array of connection endpoints. Each item has a name, host, port, and group.

These include connection endpoints for link access.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Link concept, Site linking

network

array
advanced

Platforms	Kubernetes, Docker, Podman, Linux

sitesInNetwork

integer
advanced

Platforms	Kubernetes, Docker, Podman, Linux
See also	Network concept

Source: topics/application-tls.md

Application TLS

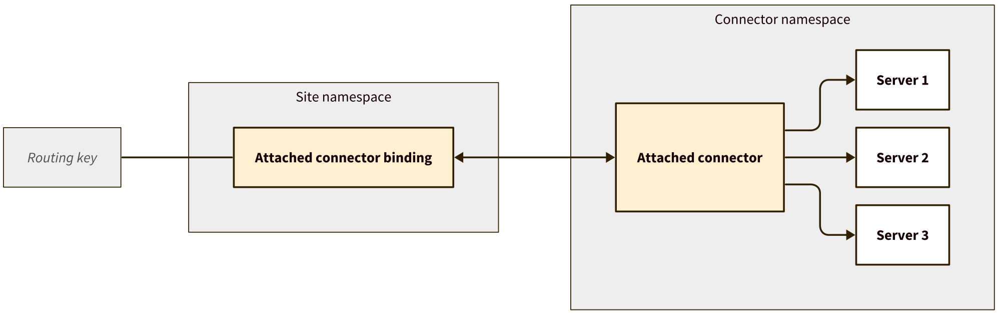
- Client-to-router and router-to-server TLS.
- Hop-by-hop TLS, not end-to-end TLS.
- An alternative to purely application-level end-to-end TLS.
- Simplifies certificate management.

Source: topics/attached-connectors.md

Attached connectors

- An attached connector is one not directly in the site namespace but in a peer namespace.
- Useful for sharing services across networks.
- Requires the router namespace and the workload namespace to opt in to the attachment.
- The router side controls the routing key. The workload side controls the selector.
- siteNamespace and connectorNamespace must correspond.
- AttachedConnector and AttachedConnectorBinding must have matching names.
- The connector side is responsible for selecting pods, while the binding side controls the routing key.
- If you want to expose a workload (say a database) in multiple networks, you need multiple AttachedConnectors, one for each corresponding binding that resides in a particular site belonging to a network.
- You can’t create attached connectors with the CLI. You have to use YAML resources.

An *attached connector* is a connector in a peer namespace.



XXX

Source: topics/components.md

Components

- The controller is focused on interacting with the Kube API
- The controller is all about reconciling input and output resources within the Kube API
- All direct interaction with the router is the job of “kube-adaptor”

Source: topics/controller-configuration.md

Controller configuration

The controller configuration controls two aspects at present: the access types supported and their configuration, and whether the grant server is enabled and how it is configured.

Access type configuration:

Option	Description
`-default-access-type`	The default access type.
`-enabled-access-types`	The access types which should be enabled for sites to choose from. (default `local,loadbalancer,route`)
`-cluster-host`	The hostname or IP address through which the cluster can be reached. Required for configuring nodeport as an access type.
`-ingress-domain`	The domain to use in constructing the fully qualified hostname for Ingress resources, through which the ingress controller can be reached. Only used when selecting `ingress-nginx` as an access type.
`-http-proxy-domain`	The domain to use in constructing the fully qualified hostname for contour HttpProxy resources, through which the contour controller can be reached. Only used when selecting `contour-http-proxy` as an access type.
`-gateway-class`	The class of Gateway to use. This is required to enable `gateway` as an access type.
`-gateway-domain`	The domain to use in constructing the fully qualified hostname for TLSRoutes resources. Only used when selecting `gateway` as an access type.
`-gateway-port`	The port the Gateway should be configured to listen on. This is only used if `gateway` is enabled as an access type. (default 8443)

Grant server configuration:

Option	Description
`-enable-grants`	Enable use of AccessGrants.
`-grant-server-autoconfigure`	Automatically configure the URL and TLS credentials for the AccessGrant Server.
`-grant-server-base-url`	The base url through which the AccessGrant server can be reached.
`-grant-server-port`	The port on which the AccessGrant server should listen. (default 9090)
`-grant-server-tls-credentials`	The name of a secret in which TLS credentials for the AccessGrant server are found. (default `skupper-grant-server`)
`-grant-server-podname`	The name of the pod in which the AccessGrant server is running (default `$HOSTNAME`)

Source: topics/debug-dumps.md

Debug dumps

- The purpose of a debug dump is to package up the details of a site so another party can identify and fix a problem.
- A dump is a tarball containing various files with the site details.
- Key elements include site resources and status; component versions, config files, and logs; and info about the environment where Skupper is running.
- Should we include workloads in the namespace? Services, deployments, pods?
- .txt file summaries for some things?
- What details about the overall network should we get?
  - Links from other sites?

```
# Same as the output of 'skupper version -o yaml'
version.yaml

# Same as the output of 'kubectl -n <site-namespace> get <kind>/<name>
-o yaml'
resources/<kind>-<name>.yaml

# Same as the output of 'kubectl -n <skupper-namespace> get
<kind>/<name> -o yaml'
resources/<kind>-<name>.yaml
```

Source: topics/high-availability.md

High availability

- Multiple routers, not controllers.
- HA is two routers, each with its own link access.
- Reduces the time to recover after a router restarts.

Source: topics/index.md

Topics

{{directory_nav(page)}}

Source: topics/individual-pod-services.md

Individual pod services

- Directly connect to individual pods across a Skupper network.
- Uses the pod name to create each service.
- This is for Kubernetes only.

Source: topics/large-networks.md

Large networks

- Skupper can scale up to networks with many sites.
- Beyond 16 sites, you may want to configure some sites to be edge sites.
- But you should not try to put a bunch of applications on one big network. It’s less secure and less performant.

Source: topics/load-balancing.md

Load balancing

- Skupper load balances connections (not requests) across connectors for the same routing key.
- The load balancing is not round robin. It is balanced according to capacity.
- The capacity calculation can be adjusted using link cost.

Source: topics/resource-settings.md

Resource settings

- Each Skupper resource has a set of additional settings.

- These are key-value pairs, where the key and the value are strings.
- These are less frequently used and exist to handle more marginal scenarios.
- Normally it’s best if users leave this at their default values.

Source: topics/resource-status.md

Resource status

Source: topics/router-tls.md

Router TLS

- Routers always communicate using mutual TLS.
- By default, the certs for this are automatically generated.
- You can provide your own certs if you wish.

Source: topics/service-exposure.md

Service exposure

- To expose multi-port services, create multiple listeners and connectors, one for each port (and using the same host).

Source: topics/site-configuration.md

Site configuration

Source: topics/site-linking.md

Site linking

- Using tokens and the CLI
- Using tokens and YAML
- Token distribution methods
- Using link generation
- Using a network-scoped CA
- Special concerns for non-Kube sites

Using kubectl to generate an access token from an access grant

```
kubectl -n sk1 get accessgrant/<grant-name> -o template --template '
apiVersion: skupper.io/v2alpha1
kind: AccessToken
metadata:
  name: <token-name>
spec:
  code: "{{{ .status.code }}}"
  ca: {{{ printf "%q" .status.ca }}}
  url: "{{{ .status.url }}}"
' > token.yaml
```

Source: topics/skupper-overview.md

Skupper overview

Source: topics/system-namespaces.md

System namespaces

- Kubernetes already has namespaces. This is for non-Kubernetes platforms: Docker, Podman, and Linux.
- Filesystem path: ~/.local/share/skupper/namespaces/
- The default namespace is named `default`.
- Each namespace contains…

Source: topics/system-tls-credentials.md

System TLS credentials

- Kubernetes already has secrets. The Docker, Podman, and Linux platforms use a directory in a well-known location.
- Location: /input/certs and /input/issuers
- Also: /runtime/certs and issuers
- Each directory has the files…

Source: commands/connector/create.md

Connector create command

```
skupper connector create <name> <port> [options]
```

Create a connector.

Platforms	Kubernetes, Docker, Podman, Linux
Waits for	Configured

Examples

```
# Create a connector for a database
$ skupper connector create database 5432
Waiting for status...
Connector "database" is configured.

# Set the routing key and selector explicitly
$ skupper connector create backend 8080 --routing-key be1 --selector
app=be1

# Use the workload option to select pods
$ skupper connector create backend 8080 --workload deployment/backend
```

Primary options

<name>

string
required

The name of the resource to be created.

The name is the default routing key if the `--routing-key` option is not specified. On Kubernetes, the name defines the default pod selector if the `--selector` and `--workload` options are not specified.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

<port>

integer
required

The port on the target server to connect to.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–routing-key

<string>
frequently used

The identifier used to route traffic from listeners to connectors. To expose a local workload to a remote site, the remote listener and the local connector must have matching routing keys.

Default	*Value of name*
Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–workload

<resource>
frequently used

A Kubernetes resource name that identifies a workload. It uses `<resource-type>/<resource-name>` syntax and resolves to an equivalent pod selector.

This is an alternative to setting the `--selector` or `--host` options.

Platforms	Kubernetes
See also	Kubernetes workloads

–selector

<string>

A Kubernetes label selector for specifying target server pods. It uses `<label-name>=<label-value>` syntax.

This is an alternative to setting the `--workload` or `--host` options.

Default	`app=[value-of-name]`
Platforms	Kubernetes
Updatable	True
See also	Kubernetes label selectors

–host

<string>

The hostname or IP address of the server. This is an alternative to `selector` for specifying the target server.

This is an alternative to setting the `--selector` or `--workload` options.

Default	*Value of name*
Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–wait

<status>

Wait for the given status before exiting.

<table>
<tr><th>Default</th><td><code>ready</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>none</code></th><td>Do not wait.</td></tr>
<tr><th><code>configured</code></th><td>Wait until the configuration is applied.</td></tr>
<tr><th><code>ready</code></th><td>Wait until the resource is ready to use.</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes</td></tr>
<tr><th>See also</th><td>Resource status</td></tr>
</table>

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes
See also	Duration format

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

<table>
<tr><th>Default</th><td><code>kubernetes</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>kubernetes</code></th><td>Kubernetes</td></tr>
<tr><th><code>docker</code></th><td>Docker</td></tr>
<tr><th><code>podman</code></th><td>Podman</td></tr>
<tr><th><code>linux</code></th><td>Linux</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes, Docker, Podman, Linux</td></tr>
<tr><th>See also</th><td>Platform concept</td></tr>
</table>

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/connector/delete.md

Connector delete command

```
skupper connector delete <name> [options]
```

Delete a connector.

Platforms	Kubernetes, Docker, Podman, Linux
Waits for	Deletion

Examples

```
# Delete a connector
$ skupper connector delete database
```

```
Waiting for deletion...
Connector "database" is deleted.
```

Primary options

<name>

string
required

The name of the resource to be deleted.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes

–wait

boolean

Wait for deletion to complete before exiting.

Default	true
Platforms	Kubernetes

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

<table>
<tr><th>Default</th><td><code>kubernetes</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>kubernetes</code></th><td>Kubernetes</td></tr>
<tr><th><code>docker</code></th><td>Docker</td></tr>
<tr><th><code>podman</code></th><td>Podman</td></tr>
<tr><th><code>linux</code></th><td>Linux</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes, Docker, Podman, Linux</td></tr>
<tr><th>See also</th><td>Platform concept</td></tr>
</table>

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/connector/generate.md

Connector generate command

```
skupper connector generate <name> <port> [options]
```

Generate a Connector resource.

Platforms	Kubernetes, Docker, Podman, Linux

Examples

```
# Generate a Connector resource and print it to the console
$ skupper connector generate backend 8080
apiVersion: skupper.io/v2alpha1
kind: Connector
metadata:
  name: backend
spec:
  routingKey: backend
  port: 8080
  selector: app=backend

# Generate a Connector resource and direct the output to a file
$ skupper connector generate backend 8080 > backend.yaml
```

Primary options

<name>

string
required

The name of the resource to be generated.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

<port>

integer
required

The port on the target server to connect to.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–routing-key

<string>
frequently used

The identifier used to route traffic from listeners to connectors. To expose a local workload to a remote site, the remote listener and the local connector must have matching routing keys.

Default	*Value of name*
Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–workload

<resource>
frequently used

A Kubernetes resource name that identifies a workload. It uses `<resource-type>/<resource-name>` syntax and resolves to an equivalent pod selector.

This is an alternative to setting the `--selector` or `--host` options.

Platforms	Kubernetes
See also	Kubernetes workloads

–selector

<string>

A Kubernetes label selector for specifying target server pods. It uses `<label-name>=<label-value>` syntax.

This is an alternative to setting the `--workload` or `--host` options.

Default	`app=[value-of-name]`
Platforms	Kubernetes
Updatable	True

See also	Kubernetes label selectors

–host

<string>

The hostname or IP address of the server. This is an alternative to `selector` for specifying the target server.

This is an alternative to setting the `--selector` or `--workload` options.

Default	*Value of name*
Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–wait

<status>

Wait for the given status before exiting.

<table>
<tr><th>Default</th><td><code>configured</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>none</code></th><td><i>Do not wait</i></td></tr>
<tr><th><code>configured</code></th><td>Configured</td></tr>
<tr><th><code>ready</code></th><td>Ready</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes, Docker, Podman, Linux</td></tr>
</table>

–output

(-o) <format>

Select the output format.

<table>
<tr><th>Default</th><td><code>yaml</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>json</code></th><td>Produce JSON output</td></tr>
</table></td></tr>
</table>

	`yaml`	Produce YAML output
Platforms	Kubernetes, Docker, Podman, Linux	

Global options

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`	
Choices	`kubernetes`	Kubernetes
	`docker`	Docker
	`podman`	Podman
	`linux`	Linux
Platforms	Kubernetes, Docker, Podman, Linux	
See also	Platform concept	

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/connector/index.md

Connector command

`skupper connector [subcommand] [options]`

Platforms	Kubernetes, Docker, Podman, Linux

Subcommands

Connector create	Create a connector
Connector update	Update a connector
Connector delete	Delete a connector
Connector status	Display the status of connectors in the current site
Connector generate	Generate a Connector resource

Source: commands/connector/status.md

Connector status command

`skupper connector status [name] [options]`

Display the status of connectors in the current site.

Platforms	Kubernetes, Docker, Podman, Linux

Examples

```
# Show the status of all connectors in the current site
$ skupper connector status
NAME       STATUS   ROUTING-KEY   SELECTOR         HOST     PORT
LISTENERS
backend    Ready    backend       app=backend      <none>   8080
true
database   Ready    database      app=postgresql   <none>   5432
true

# Show the status of one connector
$ skupper connector status backend
Name:                         backend
Status:                       Ready
Message:                      <none>
```

```
Routing key:              backend
Selector:                 app=backend
Host:                     <none>
Port:                     8080
Has matching listeners:   1
```

Primary options

[name]

string
optional

An optional resource name. If set, the status command reports status for the named resource only.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes
See also	Duration format

–output

(-o) <format>

Print status to the console in a structured output format.

Choices	**`json`** Produce JSON output **`yaml`** Produce YAML output
Platforms	Kubernetes, Docker, Podman, Linux

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`
Choices	**`kubernetes`** Kubernetes **`docker`** Docker

`podman`	Podman
`linux`	Linux

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/connector/update.md

Connector update command

```
skupper connector update <name> <port> [options]
```

Update a connector.

Platforms	Kubernetes, Docker, Podman, Linux
Waits for	Configured

Examples

```
# Change the workload and port
$ skupper connector update database --workload deployment/mysql --port
3306
Waiting for status...
Connector "database" is configured.

# Change the routing key
$ skupper connector update backend --routing-key be2
```

Primary options

<name>

string
required

The name of the resource to be updated.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

<port>

integer
required

The port on the target server to connect to.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–routing-key

<string>
frequently used

The identifier used to route traffic from listeners to connectors. To expose a local workload to a remote site, the remote listener and the local connector must have matching routing keys.

Default	*Value of name*
Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–workload

<resource>
frequently used

A Kubernetes resource name that identifies a workload. It uses `<resource-type>/<resource-name>` syntax and resolves to an equivalent pod selector.

This is an alternative to setting the `--selector` or `--host` options.

Platforms	Kubernetes
See also	Kubernetes workloads

–selector

<string>

A Kubernetes label selector for specifying target server pods. It uses `<label-name>=<label-value>` syntax.

This is an alternative to setting the `--workload` or `--host` options.

Default	`app=[value-of-name]`
Platforms	Kubernetes
Updatable	True
See also	Kubernetes label selectors

–host

<string>

The hostname or IP address of the server. This is an alternative to `selector` for specifying the target server.

This is an alternative to setting the `--selector` or `--workload` options.

Default	*Value of name*
Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–wait

<status>

Wait for the given status before exiting.

<table>
<tr><th>Default</th><td><code>ready</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>none</code></th><td><em>Do not wait</em></td></tr>
<tr><th><code>configured</code></th><td>Configured</td></tr>
<tr><th><code>ready</code></th><td>Ready</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes</td></tr>
</table>

See also	Resource status

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

<table>
<tr><th>Default</th><td><code>kubernetes</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>kubernetes</code></th><td>Kubernetes</td></tr>
<tr><th><code>docker</code></th><td>Docker</td></tr>
<tr><th><code>podman</code></th><td>Podman</td></tr>
<tr><th><code>linux</code></th><td>Linux</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes, Docker, Podman, Linux</td></tr>
<tr><th>See also</th><td>Platform concept</td></tr>
</table>

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/debug/check.md

Debug check command

```
skupper debug check [options]
```

Run diagnostic checks.

Platforms	Kubernetes, Docker, Podman, Linux

Primary options

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

<table>
<tr><th>Default</th><td><code>kubernetes</code></td></tr>
<tr><th>Choices</th><td><table><tr><th><code>kubernetes</code></th><td>Kubernetes</td></tr></table></td></tr>
</table>

`docker`	Docker
`podman`	Podman
`linux`	Linux

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/debug/dump.md

Debug dump command

```
skupper debug dump [file] [options]
```

Generate a debug dump file. Debug dumps collect the details of a site so another party can identify and fix a problem.

Platforms	Kubernetes, Docker, Podman, Linux

Examples

```
# Generate a dump file
$ skupper debug dump
Debug dump file: /home/fritz/skupper-dump-west-2024-12-09.tar.gz

# Generate a dump file to a particular path
$ skupper debug dump /tmp/abc.tar.gz
Debug dump file: /tmp/abc.tar.gz
```

Primary options

[file]

string
optional

The name of the file to generate.

The command exits with an error if the file already exists.

Default	`skupper-dump-<site-name>-<date>.tar.gz`
Platforms	Kubernetes, Docker, Podman, Linux

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`	
Choices	**`kubernetes`**	Kubernetes
	`docker`	Docker
	`podman`	Podman
	`linux`	Linux
Platforms	Kubernetes, Docker, Podman, Linux	
See also	Platform concept	

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/debug/index.md

Debug command

```
skupper debug [subcommand] [options]
```

Platforms	Kubernetes, Docker, Podman, Linux

Subcommands

Debug check	Run diagnostic checks
Debug dump	Generate a debug dump file

Source: commands/link/delete.md

Link delete command

```
skupper link delete <name> [options]
```

Delete a link.

Platforms	Kubernetes, Docker, Podman, Linux
Waits for	Deletion

Examples

```
# Delete a link
$ skupper link delete west-6bfn6
Waiting for deletion...
Link "west-6bfn6" is deleted.
```

Primary options

<name>

string
required

The name of the resource to be deleted.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes

–wait

boolean

Wait for deletion to complete before exiting.

Default	true
Platforms	Kubernetes

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>

global

Set the Skupper platform.

Default	`kubernetes`
Choices	**`kubernetes`** Kubernetes **`docker`** Docker **`podman`** Podman **`linux`** Linux
Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/link/generate.md

Link generate command

```
skupper link generate [name] [options]
```

Generate a Link resource for use in a remote site.

Generating a link requires a site with link access enabled. The command waits for the site to enter the ready state before producing the link.

Platforms	Kubernetes, Docker, Podman, Linux
Waits for	Site resource ready

Examples

```
# Generate a Link resource and print it to the console
$ skupper link generate
apiVersion: skupper.io/v2alpha1
kind: Link
metadata:
  name: south-ac619
spec:
  endpoints:
    - group: skupper-router-1
      host: 10.97.161.185
      name: inter-router
      port: "55671"
    - group: skupper-router-1
      host: 10.97.161.185
      name: edge
      port: "45671"
  tlsCredentials: south-ac619
---
apiVersion: v1
kind: Secret
type: kubernetes.io/tls
metadata:
  name: south-ac619
data:
  ca.crt: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURKekNDQWcrZ0F3SUJB
[...]
  tls.crt: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURORENDQWh5Z0F3SUJ
[...]
  tls.key: LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb3dJQkFBS0N
[...]

# Generate a Link resource and direct the output to a file
$ skupper link generate > link.yaml
```

Primary options

[name]

string
optional

The name of the resource to be generated. A name is generated if none is provided.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–cost

<integer>

The configured routing cost of sending traffic over the link.

Default	1
Platforms	Kubernetes, Docker, Podman, Linux
See also	Load balancing

–output

(-o) <format>

Select the output format.

Default	`yaml`
Choices	**`json`** Produce JSON output **`yaml`** Produce YAML output
Platforms	Kubernetes, Docker, Podman, Linux

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

<table>
<tr><th>Default</th><td><code>kubernetes</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>kubernetes</code></th><td>Kubernetes</td></tr>
<tr><th><code>docker</code></th><td>Docker</td></tr>
<tr><th><code>podman</code></th><td>Podman</td></tr>
<tr><th><code>linux</code></th><td>Linux</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes, Docker, Podman, Linux</td></tr>
<tr><th>See also</th><td>Platform concept</td></tr>
</table>

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/link/index.md

Link command

```
skupper link [subcommand] [options]
```

Platforms	Kubernetes, Docker, Podman, Linux

Subcommands

Link update	Change link settings
Link delete	Delete a link
Link status	Display the status of links in the current site
Link generate	Generate a Link resource for use in a remote site

Source: commands/link/status.md

Link status command

```
skupper link status [name] [options]
```

Display the status of links in the current site.

Platforms	Kubernetes, Docker, Podman, Linux

Examples

```
# Show the status of all links in the current site
$ skupper link status
NAME          STATUS   COST
west-6bfn6    Ready    1
south-ac619   Error    10

Links from remote sites:

<none>

# Show the status of one link
$ skupper link status west-6bfn6
Name:     west-6bfn6
Status:   Ready
```

```
Message:  <none>
Cost:     1
```

Primary options

[name]

string
optional

An optional resource name. If set, the status command reports status for the named resource only.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes
See also	Duration format

–output

(-o) <format>

Print status to the console in a structured output format.

Choices	**`json`** Produce JSON output **`yaml`** Produce YAML output
Platforms	Kubernetes, Docker, Podman, Linux

Global options

–context

<name>

global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

<table>
<tr><th>Default</th><td><code>kubernetes</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>kubernetes</code></th><td>Kubernetes</td></tr>
<tr><th><code>docker</code></th><td>Docker</td></tr>
<tr><th><code>podman</code></th><td>Podman</td></tr>
<tr><th><code>linux</code></th><td>Linux</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes, Docker, Podman, Linux</td></tr>
</table>

See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/link/update.md

Link update command

```
skupper link update <name> [options]
```

Change link settings.

Platforms	Kubernetes, Docker, Podman, Linux
Waits for	Ready

Examples

```
# Change the link cost
$ skupper link update west-6bfn6 --cost 10
Waiting for status...
Link "west-6bfn6" is ready.
```

Primary options

<name>

string
required

The name of the resource to be updated.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–cost

<integer>

The configured routing cost of sending traffic over the link.

Default	1
Platforms	Kubernetes, Docker, Podman, Linux
See also	Load balancing

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes

–wait

<status>

Wait for the given status before exiting.

<table>
<tr><th>Default</th><td><code>ready</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>none</code></th><td><em>Do not wait</em></td></tr>
<tr><th><code>configured</code></th><td>Configured</td></tr>
<tr><th><code>ready</code></th><td>Ready</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes</td></tr>
<tr><th>See also</th><td>Resource status</td></tr>
</table>

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`	
Choices	**`kubernetes`**	Kubernetes
	`docker`	Docker
	`podman`	Podman
	`linux`	Linux
Platforms	Kubernetes, Docker, Podman, Linux	
See also	Platform concept	

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/listener/create.md

Listener create command

```
skupper listener create <name> <port> [options]
```

Create a listener.

Platforms	Kubernetes, Docker, Podman, Linux
Waits for	Configured

Examples

```
# Create a listener for a database
$ skupper listener create database 5432
Waiting for status...
Listener "database" is configured.

# Set the routing key and host explicitly
$ skupper listener create backend 8080 --routing-key be1 --host
apiserver
```

Primary options

<name>

string
required

The name of the resource to be created.

The name is the default routing key and host if the `--routing-key` and `--host` options are not specified.

Platforms	Kubernetes, Docker, Podman, Linux

See also	Kubernetes object names

<port>

integer
required

The port of the local listener. Clients at this site use the listener host and port to establish connections to the remote service.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–routing-key

<string>
frequently used

The identifier used to route traffic from listeners to connectors. To enable connecting to a service at a remote site, the local listener and the remote connector must have matching routing keys.

Default	*Value of name*
Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–host

<string>
frequently used

The hostname or IP address of the local listener. Clients at this site use the listener host and port to establish connections to the remote service.

Default	*Value of name*
Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–wait

<status>

Wait for the given status before exiting.

Default	`ready`

Choices	**`none`**	Do not wait.
	`configured`	Wait until the configuration is applied.
	`ready`	Wait until the resource is ready to use.
Platforms	Kubernetes	
See also	Resource status	

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes
See also	Duration format

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

<table>
<tr><th>Default</th><td><code>kubernetes</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>kubernetes</code></th><td>Kubernetes</td></tr>
<tr><th><code>docker</code></th><td>Docker</td></tr>
<tr><th><code>podman</code></th><td>Podman</td></tr>
<tr><th><code>linux</code></th><td>Linux</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes, Docker, Podman, Linux</td></tr>
<tr><th>See also</th><td>Platform concept</td></tr>
</table>

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/listener/delete.md

Listener delete command

```
skupper listener delete <name> [options]
```

Delete a listener.

Platforms	Kubernetes, Docker, Podman, Linux
Waits for	Deletion

Examples

```
# Delete a listener
$ skupper listener delete database
Waiting for deletion...
Listener "database" is deleted.
```

Primary options

<name>

string
required

The name of the resource to be deleted.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes

–wait

boolean

Wait for deletion to complete before exiting.

Default	true
Platforms	Kubernetes

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`
Choices	**`kubernetes`** Kubernetes **`docker`** Docker

`podman`	Podman
`linux`	Linux

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/listener/generate.md

Listener generate command

```
skupper listener generate <name> <port> [options]
```

Generate a Listener resource.

Platforms	Kubernetes, Docker, Podman, Linux

Examples

```
# Generate a Listener resource and print it to the console
$ skupper listener generate backend 8080
apiVersion: skupper.io/v2alpha1
kind: Listener
metadata:
  name: backend
spec:
  routingKey: backend
  port: 8080
  host: backend

# Generate a Listener resource and direct the output to a file
$ skupper listener generate backend 8080 > backend.yaml
```

Primary options

<name>

string
required

The name of the resource to be generated.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

<port>

integer
required

The port of the local listener. Clients at this site use the listener host and port to establish connections to the remote service.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–routing-key

<string>
frequently used

The identifier used to route traffic from listeners to connectors. To enable connecting to a service at a remote site, the local listener and the remote connector must have matching routing keys.

Default	*Value of name*
Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–host

<string>
frequently used

The hostname or IP address of the local listener. Clients at this site use the listener host and port to establish connections to the remote service.

Default	*Value of name*
Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–wait

<status>

Wait for the given status before exiting.

Default	`configured`	
Choices	**`none`**	*Do not wait*
	`configured`	Configured
	`ready`	Ready
Platforms	Kubernetes, Docker, Podman, Linux	

–output

(-o) <format>

Select the output format.

Default	`yaml`	
Choices	**`json`**	Produce JSON output
	`yaml`	Produce YAML output
Platforms	Kubernetes, Docker, Podman, Linux	

Global options

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`

Choices	`kubernetes`	Kubernetes
	`docker`	Docker
	`podman`	Podman
	`linux`	Linux
Platforms	Kubernetes, Docker, Podman, Linux	
See also	Platform concept	

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/listener/index.md

Listener command

```
skupper listener [subcommand] [options]
```

Platforms	Kubernetes, Docker, Podman, Linux

Subcommands

Listener create	Create a listener
Listener update	Update a listener
Listener delete	Delete a listener
Listener status	Display the status of listeners in the current site
Listener generate	Generate a Listener resource

Source: commands/listener/status.md

Listener status command

`skupper listener status [name] [options]`

Display the status of listeners in the current site.

Platforms	Kubernetes, Docker, Podman, Linux

Examples

```
# Show the status of all listeners in the current site
$ skupper listener status
NAME       STATUS   ROUTING-KEY   HOST       PORT   CONNECTORS
backend    Ready    backend       backend    8080   true
database   Ready    database      database   5432   true

# Show the status of one listener
$ skupper listener status backend
Name:                        backend
Status:                      Ready
Message:                     <none>
Routing key:                 backend
Host:                        backend
Port:                        8080
Has matching connectors:     true
```

Primary options

[name]

string
optional

An optional resource name. If set, the status command reports status for the named resource only.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes
See also	Duration format

–output

(-o) <format>

Print status to the console in a structured output format.

Choices	`json`	Produce JSON output
	`yaml`	Produce YAML output
Platforms	Kubernetes, Docker, Podman, Linux	

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

<table>
<tr><th>Default</th><td><code>kubernetes</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>kubernetes</code></th><td>Kubernetes</td></tr>
<tr><th><code>docker</code></th><td>Docker</td></tr>
<tr><th><code>podman</code></th><td>Podman</td></tr>
<tr><th><code>linux</code></th><td>Linux</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes, Docker, Podman, Linux</td></tr>
<tr><th>See also</th><td>Platform concept</td></tr>
</table>

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/listener/update.md

Listener update command

```
skupper listener update <name> [options]
```

Update a listener.

Platforms	Kubernetes, Docker, Podman, Linux
Waits for	Configured

Examples

```
# Change the host and port
$ skupper listener update database --host mysql --port 3306
Waiting for status...
Listener "database" is configured.

# Change the routing key
$ skupper listener update backend --routing-key be2
```

Primary options

<name>

string
required

The name of the resource to be updated.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–host

<string>
frequently used

The hostname or IP address of the local listener. Clients at this site use the listener host and port to establish connections to the remote service.

Default	*Value of name*
Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–port

<integer>
frequently used

The port of the local listener. Clients at this site use the listener host and port to establish connections to the remote service.

Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–routing-key

<string>
frequently used

The identifier used to route traffic from listeners to connectors. To enable connecting to a service at a remote site, the local listener and the remote connector must have matching routing keys.

Default	*Value of name*
Platforms	Kubernetes, Docker, Podman, Linux
Updatable	True

–wait

<status>

Wait for the given status before exiting.

<table>
<tr><th>Default</th><td><code>ready</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>none</code></th><td><em>Do not wait</em></td></tr>
<tr><th><code>configured</code></th><td>Configured</td></tr>
<tr><th><code>ready</code></th><td>Ready</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes</td></tr>
<tr><th>See also</th><td>Resource status</td></tr>
</table>

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

<table>
<tr><th>Default</th><td><code>kubernetes</code></td></tr>
<tr><th>Choices</th><td><table><tr><th><code>kubernetes</code></th><td>Kubernetes</td></tr><tr><th><code>docker</code></th><td>Docker</td></tr></table></td></tr>
</table>

`podman`	Podman
`linux`	Linux

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/site/create.md

Site create command

```
skupper site create <name> [options]
```

Create a site.

Platforms	Kubernetes, Docker, Podman, Linux
Waits for	Ready

Examples

```
# Create a site
$ skupper site create west
Waiting for status...
Site "west" is ready.

# Create a site that can accept links from remote sites
$ skupper site create west --enable-link-access
```

Primary options

<name>

string
required

A name of your choice for the Skupper site. This name is displayed in the console and CLI output.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–enable-link-access

boolean
frequently used

Allow external access for links from remote sites.

Sites and links are the basis for creating application networks. In a simple two-site network, at least one of the sites must have link access enabled.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Link concept, Site linking

–link-access-type

<type>

Configure external access for links from remote sites.

Sites and links are the basis for creating application networks. In a simple two-site network, at least one of the sites must have link access enabled.

Default	`default`	
Choices	**`default`**	Use the default link access. On OpenShift, the default is `route`. For other Kubernetes flavors, the default is `loadbalancer`.
	`route`	Use an OpenShift route. *OpenShift only.*
	`loadbalancer`	Use a Kubernetes load balancer. *Kubernetes only.*
Platforms	Kubernetes	

Updatable	True
See also	Site linking

–enable-ha

boolean

Configure the site for high availability (HA). HA sites have two active routers.

Note that Skupper routers are stateless, and they restart after failure. This already provides a high level of availability. Enabling HA goes further and reduces the window of downtime caused by restarts.

Default	False
Platforms	Kubernetes
Updatable	True
See also	High availability

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes
See also	Duration format

–wait

<status>

Wait for the given status before exiting.

Default	`ready`	
Choices	**`none`**	Do not wait.
	`configured`	Wait until the configuration is applied.
	`ready`	Wait until the resource is ready to use.
Platforms	Kubernetes	

See also	Resource status

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`	
Choices	**`kubernetes`**	Kubernetes

`docker`	Docker
`podman`	Podman
`linux`	Linux

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Errors

- **A site resource already exists**

  There is already a site resource defined for the namespace.

Source: commands/site/delete.md

Site delete command

```
skupper site delete [name] [options]
```

Delete a site.

Platforms	Kubernetes, Docker, Podman, Linux
Waits for	Deletion

Examples

```
# Delete the current site
$ skupper site delete
Waiting for deletion...
```

```
Site "west" is deleted.

# Delete the current site and all of its associated Skupper resources
$ skupper site delete --all
```

Primary options

[name]

string
optional

The name of the site resource.

If not specified, the name is that of the site associated with the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–all

boolean
frequently used

In addition the site resource, delete all of the Skupper resources associated with the site in the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes

–wait

boolean

Wait for deletion to complete before exiting.

Default	true
Platforms	Kubernetes

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`
Choices	**`kubernetes`** Kubernetes **`docker`** Docker

`podman`	Podman
`linux`	Linux

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Errors

- **No site resource exists**

  There is no existing Skupper site resource to delete.

Source: commands/site/generate.md

Site generate command

```
skupper site generate <name> [options]
```

Generate a Site resource.

Platforms	Kubernetes, Docker, Podman, Linux

Examples

```
# Generate a Site resource and print it to the console
$ skupper site generate west --enable-link-access
apiVersion: skupper.io/v2alpha1
kind: Site
metadata:
  name: west
spec:
  linkAccess: default

# Generate a Site resource and direct the output to a file
$ skupper site generate east > east.yaml
```

Primary options

<name>

string
required

The name of the resource to be generated.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–enable-link-access

boolean
frequently used

Allow external access for links from remote sites.

Sites and links are the basis for creating application networks. In a simple two-site network, at least one of the sites must have link access enabled.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Link concept, Site linking

–output

(-o) <format>

Select the output format.

Default	`yaml`
Choices	**`json`** Produce JSON output **`yaml`** Produce YAML output
Platforms	Kubernetes, Docker, Podman, Linux

–link-access-type

<type>

Configure external access for links from remote sites.

Sites and links are the basis for creating application networks. In a simple two-site network, at least one of the sites must have link access enabled.

Default	`default`
Choices	**`default`** Use the default link access. On OpenShift, the default is `route`. For other Kubernetes flavors, the default is `loadbalancer`. **`route`** Use an OpenShift route. *OpenShift only.* **`loadbalancer`** Use a Kubernetes load balancer. *Kubernetes only.*
Platforms	Kubernetes
Updatable	True
See also	Site linking

–enable-ha

boolean

Configure the site for high availability (HA). HA sites have two active routers.

Note that Skupper routers are stateless, and they restart after failure. This already provides a high level of availability. Enabling HA goes further and reduces the window of downtime caused by restarts.

Default	False
Platforms	Kubernetes
Updatable	True
See also	High availability

Global options

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`	
Choices	**`kubernetes`**	Kubernetes
	`docker`	Docker
	`podman`	Podman
	`linux`	Linux
Platforms	Kubernetes, Docker, Podman, Linux	
See also	Platform concept	

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/site/index.md

Site command

```
skupper site [subcommand] [options]
```

Platforms	Kubernetes, Docker, Podman, Linux

Subcommands

Site create	Create a site
Site update	Change site settings
Site delete	Delete a site

Site status	Display the status of a site
Site generate	Generate a Site resource

Source: commands/site/reload.md

Site reload command

Reload the site configuration.

Platforms	Docker, Podman, Systemd

Usage

```
skupper site reload [options]
```

Source: commands/site/start.md

Site start command

Start running the Skupper components for the current site.

Platforms	Docker, Podman, Systemd

Usage

```
skupper site start [options]
```

Source: commands/site/status.md

Site status command

```
skupper site status [name] [options]
```

Display the status of a site.

Platforms	Kubernetes, Docker, Podman, Linux

Examples

```
# Show the status of the current site
$ skupper site status
Name:      west
Status:    Ready
Message:   -
```

Primary options

[name]

string
optional

The name of the site resource.

If not specified, the name is that of the site associated with the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes
See also	Duration format

–output

(-o) <format>

Print status to the console in a structured output format.

Choices	`json` Produce JSON output `yaml` Produce YAML output
Platforms	Kubernetes, Docker, Podman, Linux

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`
Choices	**`kubernetes`** Kubernetes **`docker`** Docker **`podman`** Podman **`linux`** Linux
Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/site/stop.md

Site stop command

Shut down the Skupper components for the current site.

Platforms	Docker, Podman, Systemd

Usage

```
skupper site stop [options]
```

Source: commands/site/update.md

Site update command

```
skupper site update [name] [options]
```

Change site settings.

Platforms	Kubernetes, Docker, Podman, Linux
Waits for	Ready

Examples

```
# Update the current site to accept links
$ skupper site update --enable-link-access
Waiting for status...
Site "west" is ready.

# Update multiple settings
$ skupper site update --enable-link-access --service-account alice
```

Primary options

[name]

string
optional

The name of the site resource.

If not specified, the name is that of the site associated with the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes object names

–enable-link-access

boolean
frequently used

Allow external access for links from remote sites.

Sites and links are the basis for creating application networks. In a simple two-site network, at least one of the sites must have link access enabled.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Link concept, Site linking

–link-access-type

<type>

Configure external access for links from remote sites.

Sites and links are the basis for creating application networks. In a simple two-site network, at least one of the sites must have link access enabled.

Default	`default`	
Choices	**`default`**	Use the default link access. On OpenShift, the default is `route`. For other Kubernetes flavors, the default is `loadbalancer`.
	`route`	Use an OpenShift route. *OpenShift only.*
	`loadbalancer`	Use a Kubernetes load balancer. *Kubernetes only.*
Platforms	Kubernetes	
Updatable	True	
See also	Site linking	

–enable-ha

boolean

Configure the site for high availability (HA). HA sites have two active routers.

Note that Skupper routers are stateless, and they restart after failure. This already provides a high level of availability. Enabling HA goes further and reduces the window of downtime caused by restarts.

Default	False
Platforms	Kubernetes
Updatable	True
See also	High availability

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes

–wait

<status>

Wait for the given status before exiting.

Default	`ready`
Choices	**`none`** *Do not wait* **`configured`** Configured **`ready`** Ready
Platforms	Kubernetes
See also	Resource status

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`
Choices	**`kubernetes`** Kubernetes **`docker`** Docker **`podman`** Podman **`linux`** Linux
Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Errors

- **No site resource exists**

  There is no existing Skupper site resource to update.

Source: commands/system/apply.md

System apply command

Apply resource configuration from files or standard input.

Platforms	Docker, Podman, Linux

Usage

```
skupper system apply [options]
```

Primary options

Global options

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`
Choices	**`kubernetes`** Kubernetes **`docker`** Docker **`podman`** Podman

`linux`	Linux

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/system/index.md

System command

```
skupper system [subcommand] [options]
```

Platforms	Docker, Podman, Linux

Subcommands

System install	Install the Skupper components
System uninstall	Remove the Skupper components
System start	Start up the Skupper components for the current site
System stop	Shut down the Skupper components for the current site
System reload	Reload the site configuration
System status	Display the status of the system

Source: commands/system/install.md

System install command

```
skupper system install [options]
```

Install the Skupper components.

This creates the router configuration, TLS certificates, and systemd unit file for the current namespace. On Docker or Podman, it also creates containers for Skupper components.

Platforms	Docker, Podman, Linux

Primary options

Global options

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`	
Choices	**`kubernetes`**	Kubernetes
	`docker`	Docker
	`podman`	Podman
	`linux`	Linux
Platforms	Kubernetes, Docker, Podman, Linux	
See also	Platform concept	

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/system/reload.md

System reload command

```
skupper system reload [options]
```

Reload the site configuration.

This restarts the systemd service for the current namespace.

Platforms	Docker, Podman, Linux

Source: commands/system/start.md

System start command

```
skupper system start [options]
```

Start up the Skupper components for the current site.

This starts the systemd service for the current namespace.

Platforms	Docker, Podman, Linux

Primary options

Global options

–namespace

(-n) <name>

global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`
Choices	**`kubernetes`** Kubernetes **`docker`** Docker **`podman`** Podman **`linux`** Linux
Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/system/status.md

System status command

```
skupper system status [options]
```

Display the status of the system.

Platforms	Docker, Podman, Linux

Primary options

Global options

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`	
Choices	**`kubernetes`**	Kubernetes
	`docker`	Docker
	`podman`	Podman
	`linux`	Linux
Platforms	Kubernetes, Docker, Podman, Linux	
See also	Platform concept	

–help

(-h) boolean

global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/system/stop.md

System stop command

```
skupper system stop [options]
```

Shut down the Skupper components for the current site.

This stops the systemd service for the current namespace.

Platforms	Docker, Podman, Linux

Primary options

Global options

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`
Choices	**`kubernetes`** Kubernetes

`docker`	Docker
`podman`	Podman
`linux`	Linux

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/system/uninstall.md

System uninstall command

```
skupper system uninstall [options]
```

Remove the Skupper components.

This removes the router configuration, TLS certificates, and systemd unit file for the current namespace. On Docker or Podman, it also removes the containers for Skupper components.

Platforms	Docker, Podman, Linux

Primary options

Global options

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`	
Choices	**`kubernetes`**	Kubernetes
	`docker`	Docker
	`podman`	Podman
	`linux`	Linux
Platforms	Kubernetes, Docker, Podman, Linux	
See also	Platform concept	

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Source: commands/token/index.md

Token command

```
skupper token [subcommand] [options]
```

Platforms	Kubernetes, Docker, Podman, Linux

Subcommands

Token issue	Issue a token file redeemable for a link to the current site
Token redeem	Redeem a token file in order to create a link to a remote site

Source: commands/token/issue.md

Token issue command

```
skupper token issue <file> [options]
```

Issue a token file redeemable for a link to the current site.

This command first creates an access grant in order to issue the token.

Issuing a token requires a site with link access enabled. The command waits for the site to enter the ready state before producing the token.

Platforms	Kubernetes
Waits for	Ready

Examples

```
# Issue an access token
$ skupper token issue ~/token.yaml
Waiting for status...
Access grant "west-6bfn6" is ready.
Token file /home/fritz/token.yaml created.

Transfer this file to a remote site. At the remote site,
create a link to this site using the 'skupper token
redeem' command:

    $ skupper token redeem <file>

The token expires after 1 use or after 15 minutes.

# Issue an access token with non-default limits
$ skupper token issue ~/token.yaml --expiration-window 24h --
redemptions-allowed 3
```

```
# Issue a token using an existing access grant
$ skupper token issue ~/token.yaml --grant west-1
```

Primary options

<file>

string
required

The name of the token file to create.

Platforms	Kubernetes, Docker, Podman, Linux

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes, Docker, Podman, Linux

–expiration-window

<duration>

The period of time in which an access token for this grant can be redeemed.

Default	`15m`
Platforms	Kubernetes, Docker, Podman, Linux

–redemptions-allowed

<integer>

The number of times an access token for this grant can be redeemed.

Default	1
Platforms	Kubernetes, Docker, Podman, Linux

–grant

<name>
advanced

Use the named access grant instead of creating a new one.

Platforms	Kubernetes, Docker, Podman, Linux

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

Default	`kubernetes`
Choices	**`kubernetes`** Kubernetes

`docker`	Docker
`podman`	Podman
`linux`	Linux

Platforms	Kubernetes, Docker, Podman, Linux
See also	Platform concept

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux

Errors

- **Link access is not enabled**

  Link access at this site is not currently enabled. You can use “skupper site update –enable-link-access” to enable it.

Source: commands/token/redeem.md

Token redeem command

```
skupper token redeem <file> [options]
```

Redeem a token file in order to create a link to a remote site.

Platforms	Kubernetes, Docker, Podman, Linux

Examples

```
# Redeem an access token
$ skupper token redeem ~/token.yaml
Waiting for status...
```

```
Link "west-6bfn6" is active.
You can now safely delete /home/fritz/token.yaml.
```

Primary options

<file>

string
required

The name of the token file to use.

Platforms	Kubernetes, Docker, Podman, Linux

–timeout

<duration>

Raise an error if the operation does not complete in the given period of time.

Default	`60s`
Platforms	Kubernetes, Docker, Podman, Linux

–link-cost

<integer>

The link cost to use when creating the link.

Default	1
Platforms	Kubernetes, Docker, Podman, Linux
See also	Load balancing

Global options

–context

<name>
global

Set the kubeconfig context.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–kubeconfig

<file>
global

Set the path to the kubeconfig file.

Platforms	Kubernetes
See also	Kubernetes kubeconfigs

–namespace

(-n) <name>
global

Set the current namespace.

Platforms	Kubernetes, Docker, Podman, Linux
See also	Kubernetes namespaces, System namespaces

–platform

<platform>
global

Set the Skupper platform.

<table>
<tr><th>Default</th><td><code>kubernetes</code></td></tr>
<tr><th>Choices</th><td><table>
<tr><th><code>kubernetes</code></th><td>Kubernetes</td></tr>
<tr><th><code>docker</code></th><td>Docker</td></tr>
<tr><th><code>podman</code></th><td>Podman</td></tr>
<tr><th><code>linux</code></th><td>Linux</td></tr>
</table></td></tr>
<tr><th>Platforms</th><td>Kubernetes, Docker, Podman, Linux</td></tr>
<tr><th>See also</th><td>Platform concept</td></tr>
</table>

–help

(-h) boolean
global

Display help and exit.

Platforms	Kubernetes, Docker, Podman, Linux