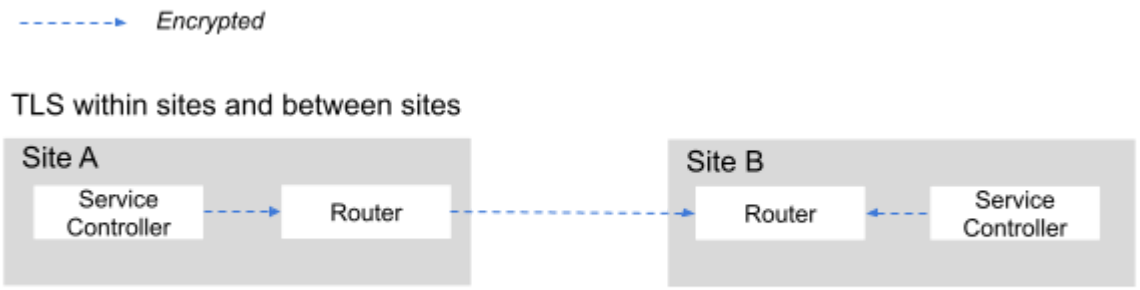
Understanding TLS certificates for Kubernetes sites

Table of Contents

The traffic between sites is encrypted using mutual TLS providing secure communication between Kubernetes clusters. If you do not provide certificates as described in this section, certificates are generated to create the mutual TLS connection. The traffic between Skupper components within a site is also encrypted using mutual TLS.



The Understanding TLS traffic between applications and routers section describes TLS traffic between a client and the router in a site, and then between the router and the server in a different site.

In TLS, two key checks occur during the handshake establishing the connection:

Trust of the peer's certificate

The certificate must be validated against a list of trusted Certificate Authorities (CAs). If the certificate is signed by a trusted CA and the certificate chain is valid, the peer's certificate is trusted.

Identity match

The peer's identity (for example, domain name) must match the Common Name (CN) or Subject Alternative Names (SANs) in the certificate. If there is a mismatch, the connection is considered insecure.

Both conditions must be met for the connection to be established.

This section describes the Kubernetes Secrets involved for various scenarios so that you can populate those Secrets using custom certificates if required.

Overview of TLS within and between sites

By default, Skupper creates Secrets to support TLS for the following traffic:

Mutual TLS within a site

Traffic flowing between a router and the service controller.

Mutual TLS between sites

Traffic flowing between routers, including between routers in different sites.

When running in Kubernetes, Skupper expects specific Secrets that support TLS for each of the above scenarios in each namespace where it is installed. These Secrets, which contain TLS keys and certificates for each of the scenarios above, are assigned predefined names.

When you create a site using the CLI or create a site declaratively using YAML, Skupper creates the required Secrets if they do not already exist.

If you use your certificates to populate the Secrets before the site is created, Skupper uses those certificates.

CA Secrets always have a name with the suffix `-ca`. These are only used to generate certificates if the associated Secrets do not already exist.

See your provider documentation for generating certificates. For example, Creating certificates for user workloads (https://docs.openshift.com/container-platform/4.17/security/cert_manager_operator/<code>cert-manager-creating</code>-certificate.html#<code>cert-manager-certificate</code>-mgmt_cert-<code>manager-creating-certificate</code>) if you use `cert-manager` on OpenShift.

Mutual TLS within a site

Within a Skupper site, the service controller pod needs to connect to the skupper router. This connection is secured using mutual TLS, and the required keys and certificates are stored in specific Secrets, all sharing the prefix **skupper-local-**:

`skupper-local-client`

Contains the key and certificate for the service controller, along with a list of trusted certificates used for verifying peer certificates.

`skupper-local-server`

Contains the key and certificate for the router, along with a list of trusted certificates used for verifying peer certificates.

If these Secrets do not exist, Skupper generates and signs those certificates using a self-signed CA certificate created during site setup, which is then stored in the `skupper-local-ca` Secret.

Skupper only uses the `skupper-local-ca` Secret if `skupper-local-server` and `skupper-local-client` are not populated.

Mutual TLS between sites

When two sites are linked, the routers communicate using mutual TLS and the required keys and certificates are stored in specific Secrets:

`skupper-site-server`

Contains the key, certificate, and the CA certificate used by the `skupper-router` when accepting links from other sites.

`<link-specific-name>`

Contains the key, certificate, and the CA certificate used by the `skupper-router` when creating links to other sites.

To establish a link, both routers must verify the peer's certificate was signed by a trusted CA.

The router on the linking site must verify the certificate of the peer matches the hostname or IP address used to establish the link.

If these Secrets do not exist, Skupper generates and signs those certificates using a self-signed CA certificate created during site setup, which is then stored in the `skupper-site-ca` Secret.

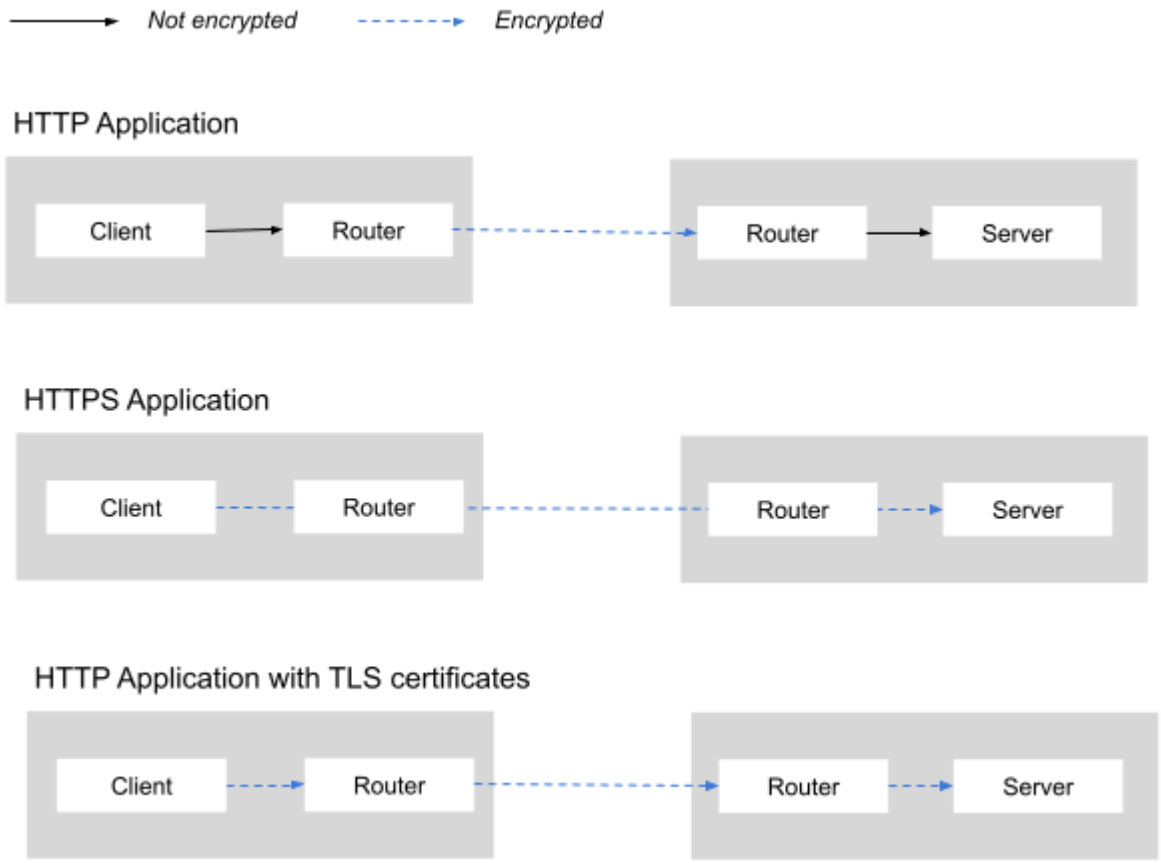
Skupper only uses the `skupper-site-ca` Secret if `skupper-site-server` is not populated.

Understanding TLS traffic between applications and routers

This section covers traffic between a client and the router in a site, and then between the router and the server in a different site.

If you need information about TLS traffic between sites or traffic between Skupper components within a site, see Understanding TLS certificates for Kubernetes sites.

Consider the following scenarios:



- An HTTP application that receives requests from a client in a remote site. The connection between the client and the router and the connection between the remote router and the server are unencrypted. The
- An HTTPS application where the traffic is encrypted by the client and unencrypted by the remote server.
- An HTTP application where traffic is encrypted at every stage: between the client and the router, between the routers, and between the router and the server at the remote site.

This section describes that third scenario, and the Secrets required to have TLS between the application and the router.

When a TLS connection from a client of a service is terminated and re-encrypted at the router, or when the router establishes a TLS connection to a pod implementing the service, additional Secrets are required.

By default, Skupper generates the following Secrets for this purpose, all sharing the prefix **skupper-service-**:

`skupper-service-client`

Contains credentials used for the TLS connection from the router to the pod implementing the service.

`skupper-service-ca`

Contains the trusted CA certificate used for validating client and server certificates in the TLS connection.

These Secrets can be provided by the user and specified through the `--tls-cert` and `--tls-trust` options to `skupper expose` or by using the equivalent annotations.

Summary of TLS related Secrets

Scenario	Secret Name	Notes
Mutual TLS within a site	`skupper-local-ca`	Certificate authority for signing `skupper-local-client` and `skupper-local-server` Secrets. Created by default. Not used if user provides other Secrets.
	`skupper-local-client`	The key and certificate for the service controller.
	`skupper-local-server`	The key and certificate for the router.
Mutual TLS between Skupper sites	`skupper-site-ca`	Certificate authority for signing certificates in `skupper-site-server` and client certificates for links. Created by default. Not used if user provides other Secrets.
	`skupper-site-server`	The key and certificate for securing incoming links from other sites.
	`<link-specific-Secret>`	The key and certificate for securing outgoing links to other sites. Labeled with `skupper.io/type=connection-token`
TLS between Skupper Router and Applications	`skupper-service-ca`	Certificate authority for signing `skupper-service-client`. Created by default. Not used if user provides `skupper-service-client` Secret.
	`skupper-service-client`	The key and certificate for securing connection between application and router.