



Red Hat Application Interconnect 1.0

Securing a service network using policies

For Use with Application Interconnect 1.0 LIMITED AVAILABILITY

Red Hat Application Interconnect 1.0 Securing a service network using policies

For Use with Application Interconnect 1.0 LIMITED AVAILABILITY

Legal Notice

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This guide describes how to add and use a policy system on your Application Interconnect service network.

Table of Contents

PREFACE	3
CHAPTER 1. ABOUT THE POLICY SYSTEM	4
CHAPTER 2. INSTALLING THE POLICY SYSTEM CRD	6
CHAPTER 3. INSTALLING THE POLICY SYSTEM CRD ON A CLUSTER WITH EXISTING SITES	7
CHAPTER 4. CREATING POLICIES FOR THE POLICY SYSTEM	8
4.1. IMPLEMENT A POLICY TO ALLOW INCOMING LINKS	8
4.2. IMPLEMENT A POLICY TO ALLOW OUTGOING LINKS TO SPECIFIC HOSTS	8
4.3. IMPLEMENT A POLICY TO ALLOW SPECIFIC SERVICES	9
4.4. IMPLEMENT A POLICY TO ALLOW SPECIFIC RESOURCES	9
CHAPTER 5. EXPLORING THE CURRENT POLICIES FOR A CLUSTER	11

PREFACE

Making open source more inclusive

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).



NOTE

This Limited Availability release is not available to all customers. Contact [Red Hat Sales](#) if you are interested in learning more about Application Interconnect.

By default, Application Interconnect includes many security features, including using mutual TLS for all service network communication between sites. By default, applying the policy system to a cluster prevents all service network communication to and from that cluster. You specify granular policies to allow only the service network communication you require.



NOTE

The policy system is distinct from the **network-policy** option which restricts access to Application Interconnect services to the current namespace as described in [Configuring Application Interconnect sites using the CLI](#).

Each site in a service network runs a Application Interconnect router and has a private, dedicated certificate authority (CA). Communication between sites is secured with mutual TLS, so the service network is isolated from external access, preventing security risks such as lateral attacks, malware infestations, and data exfiltration. The policy system adds another layer at a cluster level to help a cluster administrator control access to a service network.

This guide assumes that you understand the following Application Interconnect concepts:

site

A namespace in which Application Interconnect is installed.

token

A token is required to establish a link between two sites.

service network

After exposing services using Application Interconnect, you have created a service network.

CHAPTER 1. ABOUT THE POLICY SYSTEM

After a cluster administrator installs the policy system using a Custom Resource Definition (CRD), the cluster administrator needs to configure one or more policies to allow *developers* create and use services on the service network.



NOTE

In this guide, *developers* refers to users of a cluster who have access to a namespace, but do not have administrator privileges.

A cluster administrator configures one or more of following items using custom resources (CRs) to enable communication:

Allow incoming links

Use **allowIncomingLinks** to enable developers create tokens and configure incoming links.

Allow outgoing links to specific hosts

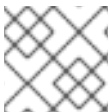
Use **allowedOutgoingLinksHostnames** to specify hosts that developers can create links to.

Allow services

Use **allowedServices** to specify which services developers can create or use on the service network.

Allow resources to be exposed

Use **allowedExposedResources** to specify which resources a developer can expose on the service network.



NOTE

A cluster administrator can apply each policy CR setting to one or more namespaces.

For example, the following policy CR fully allows all Application Interconnect capabilities on all namespaces, except for:

- only allows outgoing links to any domain ending in **.example.com**.
- only allows 'deployment/nginx' resources to be exposed on the service network.

```
apiVersion: skupper.io/v1alpha1
kind: SkupperClusterPolicy
metadata:
  name: cluster-policy-sample-01
spec:
  namespaces:
    - "*"
  allowIncomingLinks: true
  allowedExposedResources:
    - "deployment/nginx"
  allowedOutgoingLinksHostnames: [".*\\.example\\.com$"]
  allowedServices:
    - "*"

```




NOTE

You can apply many policy CRs, and if there are conflicts in the items allowed, the most permissive policy is applied. For example, if you apply an additional policy CR with the line **allowedOutgoingLinksHostnames: []**, which does not list any hostnames, outgoing links to ***.example.com** are still permitted because that is permitted in the original CR.

namespaces

One or more patterns to specify the namespaces that this policy applies to. Note that you can use [Label selectors](#) to match the namespaces.

allowIncomingLinks

Specify **true** to allow other sites create links to the specified namespaces.

allowedOutgoingLinksHostnames

Specify one or more patterns to determine which hosts you can create links to from the specified namespaces.

allowedServices

Specify one or more patterns to determine the permitted names of services allowed on the service network from the specified namespaces.

allowedExposedResources

Specify one or more permitted names of resources allowed on the service network from the specified namespaces. Note that patterns are not supported.

TIP

Use regular expressions to create pattern matches, for example:

- **.*\\.com\$** matches any string ending in **.com**. A double backslash is required to avoid issues in YAML.
- **^abc\$** matches the string **abc**.

If you create another CR that allows outgoing links for a specific namespace, a user can create a link from that namespace to join a service network. That is, the logic for multiple policy CRs is **OR**. An operation is permitted if any single policy CR permits the operation.

CHAPTER 2. INSTALLING THE POLICY SYSTEM CRD

Installing the policy system CRD enables a cluster administrator to enforce policies for service networks.



NOTE

If there are existing sites on the cluster, see [Chapter 3, *Installing the policy system CRD on a cluster with existing sites*](#) to avoid service network disruption.

Prerequisites

- Access to a cluster using a **cluster-admin** account
- The Skupper operator is installed

Procedure

1. Log in to the cluster using a **cluster-admin** account.
2. Download the CRD:

```
$ wget
https://raw.githubusercontent.com/skupperproject/skupper/1.0/api/types/crds/skupper_cluster_p
olicy_crd.yaml
```

3. Apply the CRD:

```
$ kubectl apply -f skupper_cluster_policy_crd.yaml

customresourcedefinition.apiextensions.k8s.io/skupperclusterpolicies.skupper.io created
clusterrole.rbac.authorization.k8s.io/skupper-service-controller created
```

4. To verify that the policy system is active, use the **skupper status** command and check that the output includes the following line:

```
Skupper is enabled for namespace "<namespace>" in interior mode (with policies).
```

CHAPTER 3. INSTALLING THE POLICY SYSTEM CRD ON A CLUSTER WITH EXISTING SITES

If the cluster already hosts Application Interconnect sites, note the following before installing the CRD:

- All existing connections are closed. You must apply a policy CR to reopen connections.
- All existing service network services and exposed resources are removed. You must create those resources again.

Procedure

To avoid disruption:

1. Plan the CRD deployment for an appropriate time.
2. Search your cluster for sites:

```
$ kubectl get pods --all-namespaces --selector=app=skupper
```

3. Document each service and resource exposed on the service network.
4. Install the CRD as described in [Chapter 2, Installing the policy system CRD](#). This step closes connections and removes all service network services and exposed resources.
5. If Application Interconnect sites exist in the cluster not created by **cluster-admin**, you must grant permissions to read policies to developers to avoid that site being blocked from the service network.

For each site namespace:

```
$ kubectl create clusterrolebinding skupper-service-controller-<namespace> --
clusterrole=skupper-service-controller --serviceaccount=<namespace>:skupper-service-
controller
```

where **<namespace>** is the site namespace.

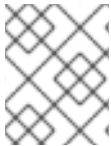
6. Create policy CRs as described in [Chapter 4, Creating policies for the policy system](#)
7. Recreate any services and exposed resources as required.

CHAPTER 4. CREATING POLICIES FOR THE POLICY SYSTEM

Policies allow a cluster administrator to control communication across the service network from a cluster.

Prerequisites

- Access to a cluster using a **cluster-admin** account.
- The policy system CRD is installed on the cluster.



PROCEDURE

Typically, you create a policy CR that combines many elements from the steps below. See [Chapter 1, *About the policy system*](#) for an example CR.

1. [Section 4.1, “Implement a policy to allow incoming links”](#)
2. [Section 4.2, “Implement a policy to allow outgoing links to specific hosts”](#)
3. [Section 4.3, “Implement a policy to allow specific services”](#)
4. [Section 4.4, “Implement a policy to allow specific resources”](#)

4.1. IMPLEMENT A POLICY TO ALLOW INCOMING LINKS

Use **allowIncomingLinks** to enable developers create tokens and configure incoming links.

Procedure

1. Determine which namespaces you want to apply this policy to.
2. Create a CR with **allowIncomingLinks** set to **true** or **false**.
3. Create and apply the CR.

For example, the following CR allows incoming links for all namespaces:

```
apiVersion: skupper.io/v1alpha1
kind: SkupperClusterPolicy
metadata:
  name: allowincominglinks
spec:
  namespaces:
    - "*"
  allowIncomingLinks: true
```

4.2. IMPLEMENT A POLICY TO ALLOW OUTGOING LINKS TO SPECIFIC HOSTS

Use **allowedOutgoingLinksHostnames** to specify hosts that developers can create links to. You cannot create a **allowedOutgoingLinksHostnames** policy to disallow a specific host that was previously allowed.

1. Determine which namespaces you want to apply this policy to.
2. Create a CR with **allowedOutgoingLinksHostnames** set to a pattern of allowed hosts.
3. Create and apply the CR.

For example, the following CR allows links to all subdomains of **example.com** for all namespaces:

```
apiVersion: skupper.io/v1alpha1
kind: SkupperClusterPolicy
metadata:
  name: allowedoutgoinglinkshostnames
spec:
  namespaces:
    - "*"
  allowedOutgoingLinksHostnames: ['.*\\.example\\.com']
```

4.3. IMPLEMENT A POLICY TO ALLOW SPECIFIC SERVICES

Use **allowedServices** to specify which services a developer can create or use on the service network. You cannot create a **allowedServices** policy to disallow a specific service that was previously allowed.

Procedure

1. Determine which namespaces you want to apply this policy to.
2. Create a CR with **allowedServices** set to specify the services allowed on the service network.
3. Create and apply the CR.

For example, the following CR allows users to expose and consume services with the prefix **backend-** for all namespaces:

```
apiVersion: skupper.io/v1alpha1
kind: SkupperClusterPolicy
metadata:
  name: allowedservices
spec:
  namespaces:
    - "*"
  allowedServices: ['^backend-']
```



NOTE

When exposing services, you can use the **--address <name>** parameter of the **skupper** CLI to name services to match your policy.

4.4. IMPLEMENT A POLICY TO ALLOW SPECIFIC RESOURCES

Use **allowedExposedResources** to specify which resources a developer can expose on the service network. You cannot create a **allowedExposedResources** policy to disallow a specific resource that was previously allowed.

Procedure

Procedure

1. Determine which namespaces you want to apply this policy to.
2. Create a CR with **allowedExposedResources** set to specify resources that a developer can expose on the service network.
3. Create and apply the CR.

For example, the following CR allows you to expose an **nginx** deployment for all namespaces:

```
apiVersion: skupper.io/v1alpha1
kind: SkupperClusterPolicy
metadata:
  name: allowedexposedresources
spec:
  namespaces:
    - "*"
  allowedExposedResources: ['deployment/nginx']
```



NOTE

For **allowedExposedResources**, each entry must conform to the **type/name** syntax.

CHAPTER 5. EXPLORING THE CURRENT POLICIES FOR A CLUSTER

As a developer you might want to check which policies are enforced for a particular site.

Procedure

1. Log into a namespace where a Application Interconnect site has been initialized.
2. Check whether incoming links are permitted:

```
$ kubectl exec deploy/skupper-service-controller -- get policies incominglink
ALLOWED POLICY ENABLED ERROR                                ALLOWED BY
false true           Policy validation error: incoming links are not allowed
```

In this example incoming links are not allowed by policy.

3. Explore other policies:

```
$ kubectl exec deploy/skupper-service-controller -- get policies
Validates existing policies

Usage:
  get policies [command]

Available Commands:
  expose      Validates if the given resource can be exposed
  incominglink Validates if incoming links can be created
  outgoinglink Validates if an outgoing link to the given hostname is allowed
  service     Validates if service can be created or imported
```

As shown, there are commands to check each policy type by specifying what you want to do, for example, to check if you can expose an nginx deployment:

```
$ kubectl exec deploy/skupper-service-controller -- get policies expose deployment nginx
ALLOWED POLICY ENABLED ERROR                                ALLOWED BY
false true           Policy validation error: deployment/nginx cannot be exposed
```

If you allowed an nginx deployment as described in [Section 4.4, “Implement a policy to allow specific resources”](#), the same command shows that the resource is allowed and displays the name of the policy CR that enabled it:

```
$ kubectl exec deploy/skupper-service-controller -- get policies expose deployment nginx
ALLOWED POLICY ENABLED ERROR                                ALLOWED BY
true true                                                    allowedexposedresources
```

Revised on 2022-06-24 16:33:30 UTC