



Red Hat Application Interconnect 1.0

ポリシーを使用したサービスネットワークの保護

Red Hat Application Interconnect 1.0 で使用する場合 (限定利用)

Red Hat Application Interconnect 1.0 ポリシーを使用したサービスネットワークの保護

Red Hat Application Interconnect 1.0 で使用する場合 (限定利用)

法律上の通知

Copyright © 2023 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.

概要

このガイドでは、Application Interconnect サービスネットワークにポリシーシステムを追加して使用する方法について説明します。

目次

はじめに	3
第1章 ポリシーシステムについて	4
第2章 ポリシーシステム CRD のインストール	6
第3章 既存のサイトがあるクラスターにポリシーシステム CRD をインストールする	7
第4章 ポリシーシステムのポリシーの作成	8
4.1. 着信リンクを許可するポリシーを実装する	8
4.2. 特定のホストへの出力リンクを許可するポリシーを実装する	8
4.3. 特定のサービスを許可するポリシーを実装する	9
4.4. 特定のリソースを許可するポリシーを実装する	9
第5章 クラスターの現在のポリシーを調べる	11

はじめに

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#)をご覧ください。



注記

この限定利用リリースは、すべてのお客様が利用できるわけではありません。Application Interconnect の詳細は、[Red Hat の営業チーム](#) にお問い合わせください。

デフォルトでは、Application Interconnect には、サイト間のすべてのサービスネットワーク通信に相互 TLS を使用するなど、多くのセキュリティ機能が含まれています。デフォルトでは、ポリシーシステムをクラスターに適用すると、そのクラスターとの間のすべてのサービスネットワーク通信が防止されます。きめ細かいポリシーを指定して、必要なサービスネットワーク通信のみを許可します。



注記

ポリシーシステムは、[CLI を使用したアプリケーションインターコネクトサイトの設定](#)で説明されているように、アプリケーションインターコネクトサービスへのアクセスを現在の namespace に制限する **network-policy** オプションとは異なります。

サービスネットワークの各サイトは、アプリケーションインターコネクトルーターを実行し、専用の認証局 (CA) を持っています。サイト間の通信は相互 TLS で保護されているため、サービスネットワークは外部アクセスから分離され、横方向の攻撃、マルウェアの侵入、データの漏えいなどのセキュリティリスクを防ぎます。ポリシーシステムは、クラスター管理者がサービスネットワークへのアクセスを制御できるように、クラスターレベルで別のレイヤーを追加します。

このガイドは、次のアプリケーション相互接続の概念を理解していることを前提としています。

site

Application Interconnect がインストールされている namespace。

トークン (token)

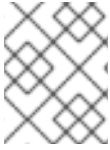
2 つのサイト間のリンクを確立するには、トークンが必要です。

サービスネットワーク

Application Interconnect を使用してサービスを公開した後、サービスネットワークを作成しました。

第1章 ポリシーシステムについて

クラスター管理者がカスタムリソース定義 (CRD) を使用してポリシーシステムをインストールした後、クラスター管理者は、**開発者**がサービスネットワーク上でサービスを作成および使用できるように1つ以上のポリシーを設定する必要があります。



注記

このガイドでは、**開発者**とは、namespace にアクセスできるが管理者権限を持たないクラスターのユーザーを指します。

クラスター管理者は、カスタムリソース (CR) を使用して、次の1つ以上の項目を設定して通信を有効にします。

入力リンクを許可する

allowIncomingLinks を使用して、開発者がトークンを作成し、入力リンクを設定できるようにします。

特定のホストへの出力リンクを許可する

allowedOutgoingLinksHostnames を使用して、開発者がリンクを作成できるホストを指定します。

サービスを許可する

allowedServices を使用して、開発者がサービスネットワーク上で作成または使用できるサービスを指定します。

リソースの公開を許可する

allowedExposedResources を使用して、開発者がサービスネットワーク上で公開できるリソースを指定します。



注記

クラスター管理者は、各ポリシー CR 設定を1つ以上の namespace に適用できます。

たとえば、次のポリシー CR は、以下を除くすべての namespace ですべてのアプリケーション相互接続機能を完全に許可します。

- **.example.com** で終わるドメインへの出力リンクのみを許可します。
- deployment/nginx リソースのみをサービスネットワークに公開できます。

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

```




注記

多くのポリシー CR を適用でき、許可された項目に競合がある場合は、最も許容度の高いポリシーが適用されます。たとえば、**allowedOutgoingLinksHostnames: []** という行を含む追加のポリシー CR を適用する場合は、ホスト名をリストしない場合でも、元の CR で許可されているため、***.example.com** への出力リンクは許可されます。

namespace

このポリシーが適用される namespace を指定するための1つ以上のパターン。[ラベルセレクター](#) を使用して namespace を一致させることができることに注意してください。

allowIncomingLinks

他のサイトが指定された namespace へのリンクを作成できるようにするには、**true** を指定します。

allowedOutgoingLinksHostnames

1つ以上のパターンを指定して、指定したネームスペースからリンクを作成できるホストを決定します。

allowedServices

1つ以上のパターンを指定して、指定された名前空間からサービスネットワークで許可されるサービスの許可される名前を決定します。

allowedExposedResources

指定された namespace から、サービスネットワークで許可されているリソースの1つ以上の許可された名前を指定します。パターンはサポートされていないことに注意してください。

ヒント

正規表現を使用して、パターンの一致を作成します。次に例を示します。

- **.*\\.com\$** は、**.com** で終わる任意の文字列に一致します。YAML の問題を回避するには、二重の円記号が必要です。
- **^abc\$** は文字列 **abc** と一致します。

特定の namespace の出力リンクを許可する別の CR を作成する場合、ユーザーはその名前空間からリンクを作成してサービスネットワークに参加できます。つまり、複数のポリシー CR のロジックは **OR** です。単一のポリシー CR で操作が許可されている場合、操作は許可されます。

第2章 ポリシーシステム CRD のインストール

ポリシーシステム CRD をインストールすると、クラスター管理者はサービスネットワークのポリシーを適用できます。



注記

クラスターに既存のサイトがある場合は、[3章 既存のサイトがあるクラスターにポリシーシステム CRD をインストールする](#)を参照して、サービスネットワークの中断を回避してください。

前提条件

- **cluster-admin** アカウントを使用したクラスターへのアクセス
- Skupper オペレーターがインストールされている

手順

1. **cluster-admin** アカウントを使用してクラスターにログインします。
2. CRD をダウンロードします。

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

3. CRD を適用します。

```
$ 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. ポリシーシステムがアクティブであることを確認するには、**skupper status** コマンドを使用して、出力に次の行が含まれていることを確認します。

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

第3章 既存のサイトがあるクラスターにポリシーシステム CRD をインストールする

クラスターがすでにアプリケーション相互接続サイトをホストしている場合は、CRD をインストールする前に次の点に注意してください。

- 既存の接続はすべて閉じられます。接続を再開するには、ポリシー CR を適用する必要があります。
- 既存のサービスネットワークサービスと公開されているリソースはすべて削除されます。これらのリソースを再度作成する必要があります。

手順

混乱を避けるために、以下を行います。

1. 適切な時期に CRD のデプロイメントを計画します。
2. クラスターでサイトを検索します。

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

3. サービスネットワーク上で公開されている各サービスとリソースを文書化します。
4. [2章 ポリシーシステム CRD のインストール](#)の説明に従って、CRD をインストールします。この手順により、接続が閉じられ、すべてのサービスネットワークサービスと公開されたリソースが削除されます。
5. **cluster-admin** によって作成されていないクラスターにアプリケーション相互接続サイトが存在する場合は、そのサイトがサービスネットワークからブロックされないように、開発者にポリシーを読み取るためのアクセス許可を付与する必要があります。
サイトの namespace ごとに、以下を行います。

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

ここで、**<namespace>** はサイトの名前空間です。

6. [4章 ポリシーシステムのポリシーの作成](#)で説明されているようにポリシー CR を作成します
7. 必要に応じて、サービスと公開されたリソースを再作成します。

第4章 ポリシーシステムのポリシーの作成

ポリシーを使用すると、クラスター管理者はクラスターからサービスネットワークを介した通信を制御できます。

前提条件

- **cluster-admin** アカウントを使用したクラスターへのアクセス。
- ポリシーシステム CRD がクラスターにインストールされます。



手順

通常、以下の手順の多くの要素を組み合わせたポリシー CR を作成します。CR の例は、[1章 ポリシーシステムについて](#)を参照してください。

1. 「着信リンクを許可するポリシーを実装する」
2. 「特定のホストへの出力リンクを許可するポリシーを実装する」
3. 「特定のサービスを許可するポリシーを実装する」
4. 「特定のリソースを許可するポリシーを実装する」

4.1. 着信リンクを許可するポリシーを実装する

allowIncomingLinks を使用して、開発者がトークンを作成し、入力リンクを設定できるようにします。

手順

1. このポリシーを適用する namespace を決定します。
2. **allowIncomingLinks** を **true** または **false** に設定して CR を作成します。
3. CR を作成して適用します。

たとえば、次の CR は、すべての namespace の入力リンクを許可します。

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

4.2. 特定のホストへの出力リンクを許可するポリシーを実装する

allowedOutgoingLinksHostnames を使用して、開発者がリンクを作成できるホストを指定します。**allowedOutgoingLinksHostnames** ポリシーを作成して、以前に許可されていた特定のホストを禁止することはできません。

1. このポリシーを適用する namespace を決定します。
2. 許可されたホストのパターンに設定された **allowedOutgoingLinksHostnames** を使用して CR を作成します。
3. CR を作成して適用します。

たとえば、次の CR は、すべての namespace の **example.com** のすべてのサブドメインへのリンクを許可します。

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

4.3. 特定のサービスを許可するポリシーを実装する

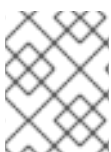
allowedServices を使用して、開発者がサービスネットワーク上で作成または使用できるサービスを指定します。以前に許可されていた特定のサービスを禁止する **allowedServices** ポリシーを作成することはできません。

手順

1. このポリシーを適用する namespace を決定します。
2. サービスネットワークで許可されるサービスを指定する **allowedServices** を設定した CR を作成します。
3. CR を作成して適用します。

たとえば、次の CR を使用すると、ユーザーはすべての namespace に対して接頭辞 **backend-** を持つサービスを公開および利用できます。

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



注記

サービスを公開するとき、ポリシーに合うようにサービスの名前を付けるため、**skupper** CLI の **--address <name>** パラメーターを利用することができます。

4.4. 特定のリソースを許可するポリシーを実装する

allowedExposedResources を使用して、開発者がサービスネットワーク上で公開できるリソースを指定します。以前に許可されていた特定のリソースを禁止する **allowedExposedResources** ポリシーを作成することはできません。

手順

1. このポリシーを適用する namespace を決定します。
2. **allowedExposedResources** を設定して CR を作成し、開発者がサービスネットワーク上で公開できるリソースを指定します。
3. CR を作成して適用します。

たとえば、次の CR を使用すると、すべての namespace の **nginx** デプロイメントを公開できます。

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



注記

allowedExposedResources の場合、各エントリは **type/name** の構文に準拠している必要があります。

第5章 クラスターの現在のポリシーを調べる

開発者は、特定のサイトに適用されているポリシーを確認することをお勧めします。

手順

1. アプリケーション相互接続サイトが初期化されているネームスペースにログインします。
2. 入力リンクが許可されているかどうかを確認します。

```
$ 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
```

この例では、入力リンクはポリシーによって許可されていません。

3. 他のポリシーを調べます。

```
$ 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
```

ここで示されているように、実行する操作を指定して各ポリシータイプをチェックするコマンドがあります。たとえば、nginx デプロイメントを公開できるかどうかをチェックします。

```
$ 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
```

「特定のリソースを許可するポリシーを実装する」で説明されているように、nginx のデプロイメントを許可した場合は、同じコマンドでリソースが許可されていることが示され、それを有効にしたポリシー CR の名前が表示されます。

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

改訂日時: 2022-07-03 00:28:49 +1000