Red Hat Service Interconnect 1.5

Release notes

Latest information about features and issues in this release

Last Updated: 2024-06-24

Red Hat Service Interconnect 1.5 Release notes

Latest information about features and issues in this release

Legal Notice

Abstract

These release notes contain the latest information about new features, enhancements, fixes, and issues contained in the Red Hat Service Interconnect 1.5 release. Red Hat Service Interconnect is a Red Hat build of the open source Skupper project.

Table of Contents

CHAPTER 1. NEW AND CHANGED FEATURES

The Red Hat build of Skupper is now known as Red Hat Service Interconnect.

The following features were designated Technology Preview in version 1.4, but are now fully supported:

Podman sites

With this release of Red Hat Service Interconnect, you can create sites on RHEL hosts as well as Kubernetes namespaces. See Using Skupper podman for more information.

Console

With this release of Red Hat Service Interconnect, you can optionally provision a console to monitor traffic flows across the service network. See Using the Skupper console for more information. This early version of the console uses an in-memory prometheus instance to populate traffic visualizations. The prometheus instance and metrics are not intended for public use, Red Hat encourage you to test and provide feedback on the console features only.

CHAPTER 2. SUPPORTED CONFIGURATIONS

Command-line interface

- Red Hat Enterprise Linux 8 x86-64 and aarch64
- Red Hat Enterprise Linux 9 x86-64 and aarch64
- Linux x86-64 and aarch64
- macOS x86-64 (Technical Preview feature)
- Windows x86-64 (Technical Preview feature)

Router

For use in Kubernetes-based sites and as a gateway for containers or machines.

- Red Hat Enterprise Linux 8 x86-64 and aarch64
- Red Hat Enterprise Linux 9 x86-64 and aarch64

NOTE

Red Hat Service Interconnect is not supported for standalone use as a messaging router.

Red Hat Service Interconnect Operator

The operator is supported with OpenShift 4.x only.

OpenShift versions

- OpenShift 3.11
- OpenShift 4.12, 4.13, 4.14 and 4.15
- ROSA and ARO
- OpenShift Container Platform and OpenShift Dedicated

Installing Red Hat Service Interconnect in a disconnected network by mirroring the required components to the cluster is supported.

Ingress types

- LoadBalancer
- OpenShift Routes

CPU architecture

- x86-64 and aarch64

Multiarch sites:

Sites created on aarch64 architecture cluster or Linux hosts are a Technical Preview feature.

Windows and macOS CLI

The Skupper CLI for Windows and macOS is a Technical Preview feature.

Kubernetes distributions

Red Hat provides assistance running Red Hat Service Interconnect on any CNCF-certified distribution of Kubernetes.

Note, however, that Red Hat Service Interconnect is tested only on OpenShift.

Ingress types

- Contour
- Nginx - This requires configuration for TLS passthrough
- NodePort

Upgrades

Red Hat supports upgrades from one downstream minor version to the next, with no jumps. While Red Hat aims to have compatibility across minor versions, we recommend upgrading all sites to latest version.

> **NOTE**
>
> If you have applications that require long lived connections, for example Kafka clients, consider using a load balancer as ingress instead of a proxy ingress such as OpenShift route. If you use an OpenShift route as ingress, expect interruptions whenever routes are configured.

For information about the latest release, see Red Hat Service Interconnect Supported Configurations .

CHAPTER 3. TECHNOLOGY PREVIEW FEATURES

Some features in this release are currently in Technology Preview. This section describes the Technology Preview features in Red Hat Service Interconnect 1.5.

IMPORTANT

Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

Windows and macOS

With this release of Red Hat Service Interconnect, you can use the **skupper** CLI on Windows and macOS.

Multiarch

With this release of Red Hat Service Interconnect, you can create sites on 64-bit ARM clusters and Linux hosts (aarch64).

CHAPTER 4. DEPRECATED FEATURES

This section describes features that are supported, but have been deprecated from Red Hat Service Interconnect.

Protocols

The **http** and **http2** protocols are deprecated and will be removed in a future release when a feature that provides similar observability becomes available. Red Hat recommends using the **tcp** protocol unless **http** or **http2** observability is required.

CHAPTER 5. UPGRADING SITES

This release of Red Hat Service Interconnect is compatible with previous versions, however Red Hat recommends upgrading all sites to version 1.5.

NOTE

Update all sites to ensure the same version of Service Interconnect is running across your service network. You can expect some minimal downtime during the update process.

To upgrade a site:

```
$ skupper update
```

See the Installation for instructions about upgrading Operator based sites.

NOTE

If you enabled the console using a release before 1.4, which was the default, the upgraded site will also have the console enabled. The default for Service Interconnect 1.4 and later is that the console is not enabled.

To upgrade a gateway, delete the gateway and recreate it.

Port negotiation limitation

If your protocol negotiates the communication port, for example active FTP, you cannot use that protocol to communicate across a service network.

CHAPTER 6. FIXED ISSUES

See Red Hat Service Interconnect 1.5.x Resolved Issues for a list of issues that have been fixed in patch releases.

CHAPTER 7. KNOWN ISSUES

- **SKUPPER-1802 - skupper service status reports a service that doesn't exist on the current site**
  If you disabled **service-sync** on your network and expose a service on one site, the **skupper service status** command reports those services as existing on all other sites even though that service does not exist on thoses sites.

  Workaround

  To workaround this issue use **skupper service status -v** to understand exposed services and availability.

  For example, on the **west** site:

  ```
  $ skupper service status

  Services exposed through Skupper:
  ╰─ backend:8080 (tcp)
  ```

  However, that service does not exist on the **west** site. Use the verbose output to show that the service only exists on the **east** site:

  ```
  $ skupper service status -v

  Services exposed through Skupper:
  ╰─ backend:8080 (tcp)
     ╰─ Sites:
        ╰─ 316fbe31-299b-490b-9391-7b46507d76f1(east)
           │ policy: disabled
           ╰─ Targets:
              ╰─ backend:8080 name=backend-9d84544df-rbzj
  ```

- **SKUPPER-869 - Enable idle connection timeouts for the TCP transport**
  If an endpoint is terminated, for example a client is killed, the other endpoint observes a half-closed connection. If the other endpoint does not close the connection or attempt to send data to the connection, the Skupper router does not release the memory allocated to that connection

  Workaround

  Avoid client server configurations that use this behavior if possible. For example, if a server automatically closes dormant connections, or attempts to communicate with client, the Skupper router frees the memory when a client is terminated.

- **SKUPPER-805 - skupper init doesn't work for ordinary user on OCP 3.11**
  Workaround

  Two workarounds are available:

  - Use YAML to configure a site.
  - Create a service account with the following permissions to run the **skupper** CLI:

  ```
  ---
  ```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: skupper-non-admin
rules:
- apiGroups:
  - ""
  resources:
  - configmaps
  - pods
  - pods/exec
  - services
  - secrets
  - serviceaccounts
  verbs:
  - get
  - list
  - watch
  - create
  - update
  - delete
- apiGroups:
  - apps
  resources:
  - deployments
  - statefulsets
  - daemonsets
  verbs:
  - get
  - list
  - watch
  - create
  - update
  - delete
- apiGroups:
  - route.openshift.io
  resources:
  - routes
  verbs:
  - get
  - list
  - watch
  - create
  - delete
- apiGroups:
  - networking.k8s.io
  resources:
  - ingresses
  - networkpolicies
  verbs:
  - get
  - list
  - watch
  - create
  - delete
- apiGroups:
  - projectcontour.io
  resources:
  - httpproxies
  verbs:
  - get
  - list
  - watch
  - create
  - delete
- apiGroups:
  - rbac.authorization.k8s.io
  resources:
  - rolebindings
  - roles
  verbs:
  - get
  - list
  - watch
  - create
  - delete
```

You can save the YAML above to **role.yaml**, apply it and bind the role to a username using:

```
$ oc apply -f role.yaml
$  oc policy add-role-to-user skupper-non-admin <username> -n <namespace-name> --role-namespace=<namespace-name>
```

APPENDIX A. ABOUT SERVICE INTERCONNECT DOCUMENTATION

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 .

Revised on 2024-06-24 15:01:24 UTC