Abstract

These release notes contain the latest information about new features, enhancements, fixes, and issues contained in the AMQ Interconnect 2.0 release.
# Table of Contents

**PREFACE** ................................................................. 3  
**CHAPTER 1. NEW AND CHANGED FEATURES** .............................. 4  
  1.1. INTRODUCTION TO AMQ INTERCONNECT 2.0 ......................... 4  
  1.2. TCP, HTTP 1 AND HTTP 2 PROTOCOLS ............................... 4  
  1.3. KNOWN ISSUES .................................................. 5
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.

IMPORTANT

AMQ Interconnect 2.0 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 https://access.redhat.com/support/offerings/techpreview.
CHAPTER 1. NEW AND CHANGED FEATURES

This release allows you connect services between OpenShift clusters, creating a service network. It also allows you connect services running on RHEL 8 using the **skupper gateway** command. See the following guides for more details:

- Creating a service network with OpenShift
- Configuring AMQ Interconnect sites using the CLI
- Using the Skupper console
- Creating AMQ Interconnect sites using the operator

The **Using the AMQ Interconnect router** describes how to create router networks using AMQP. If router networks are your goal, Red Hat recommends using the AMQ Interconnect 1 LTS release.

AMQ Interconnect 2 is tested on OpenShift Container Platform 4.6, 4.7, and 4.8.

1.1. INTRODUCTION TO AMQ INTERCONNECT 2.0

Interconnect 2.0 introduces a service network, linking services across the hybrid cloud. A service network enables communication between services running in different network locations. It allows geographically distributed services to connect as if they were all running in the same site.

For example, you can deploy your frontend in a public OpenShift cluster and deploy your backend in a private OpenShift cluster, then connect them into a service network.

A service network provides the following features:

- Security by default. All inter-site traffic is protected by mutual TLS using a private, dedicated certificate authority (CA).
- Easy connections between OpenShift clusters, even private clusters.
- A service network supports existing TCP-based applications without requiring modification.
- Monitor your application traffic spread across multiple OpenShift clusters using the service network console.

You deploy and manage a service network using the **skupper** CLI.

1.2. TCP, HTTP 1 AND HTTP 2 PROTOCOLS
This release of AMQ Interconnect supports using the following to create a service network:

- TCP and any protocol overlayed on TCP
- HTTP 1
- HTTP 2

Unlike the typical AMQ Interconnect use cases, this means that you can use existing applications without modification. The router provisioned by Skupper is not suitable for messaging applications. If you need to route AMQP messages, use the AMQ Interconnect 1 LTS release.

While router networks are still supported, if AMQP is your only goal, Red Hat recommends using the AMQ Interconnect 1 LTS release.

**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.

### 1.3. KNOWN ISSUES

- **ENTMQIC-3266** - Skupper gateway is failing to start on docker or podman against RHEL7 or RHEL8
  
  When creating a new skupper gateway on RHEL7 with docker or RHEL8 with podman, the container fails because of incorrect permissions given to some directories.

**Workaround**

1. Change the permissions using the following commands:

   ```bash
   $ cd $(HOME)/.local/share/skupper/cluster/
   $ find ./ -type d | xargs chmod 755
   
   2. Restart your container.

Revised on 2021-11-30 11:38:03 UTC