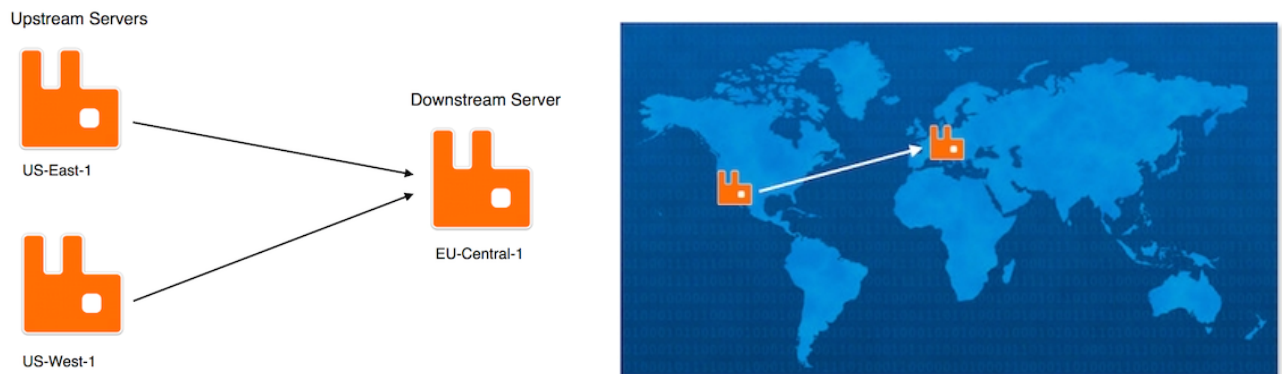


Rabbitmq Federation



Usually when RabbitMQ is used, messages are published to an exchange and the messages are then routed to the queue or the queues. With use of exchange federation it is possible to get RabbitMQ to distribute those messages to any other clusters. That means messages arriving to those federated exchanges will also be forwarded to the downstream clusters. Exchange federation consumes messages from an upstream cluster and republishes on it's own local exchange, as if the messages published on the upstream cluster were published on the local cluster. A situation where exchange federation could be used is when there are clusters in multiple regions, but you want to collect all messages to a central cluster where your consumers can process them. The federation act as an intelligent shovel, so it will create a queue on the upstream cluster, bind it to the exchange you're federating and then consume from that queue and republish them on the local exchange. If the connection is broken messages will queue up on the upstream queue and when the server reconnects again it will transfer all messages that were publish during the network outage.

Upstreams: each upstream defines how to connect to another broker.

Upstream sets: each upstream set groups together a set of upstreams to use for federation.

Policies: each policy selects a set of exchanges, queues or both, and applies a single upstream or an upstream set to those objects.

Enable Federation plugins

Commands:

rabbitmq-plugins enable rabbitmq_federation

rabbitmq-plugins enable rabbitmq_federation_management

Upstream and Downstream configuration

Add a new user, example “federation” with “management” tag



The screenshot shows the RabbitMQ Admin UI with the 'Admin' tab selected. The 'Users' section is active, displaying a table of users. The 'federation' user is highlighted, showing its tags as 'management'. The 'Add a user' form is visible below the table, with the 'Username' field set to 'federation' and the 'Tags' field set to 'management'. The 'Add user' button is at the bottom of the form.

Name	Tags	Can access virtual hosts	Has password
federation	management	*	*
foursource	administrator	./, staging6	*

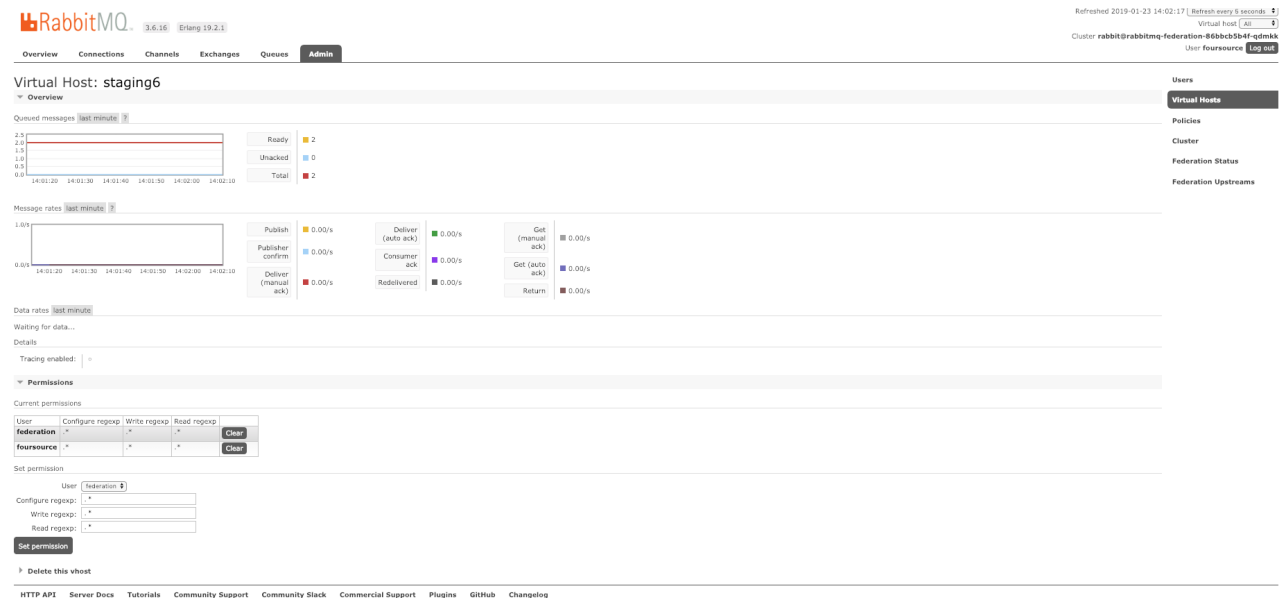
Add a virtual host, example “staging6”



The screenshot shows the RabbitMQ Admin UI with the 'Admin' tab selected. The 'Virtual Hosts' section is active, displaying a table of virtual hosts. The 'staging6' virtual host is highlighted, showing its configuration. The 'Add a new virtual host' form is visible below the table, with the 'Name' field set to 'staging6'. The 'Add virtual host' button is at the bottom of the form.

Name	Users	Ready	Unacked	Total	From client	To client	publish	deliver	/ get
/	foursource	NaN	NaN	NaN					
staging6	federation, foursource	2	0	2			0.00/s	0.00/s	

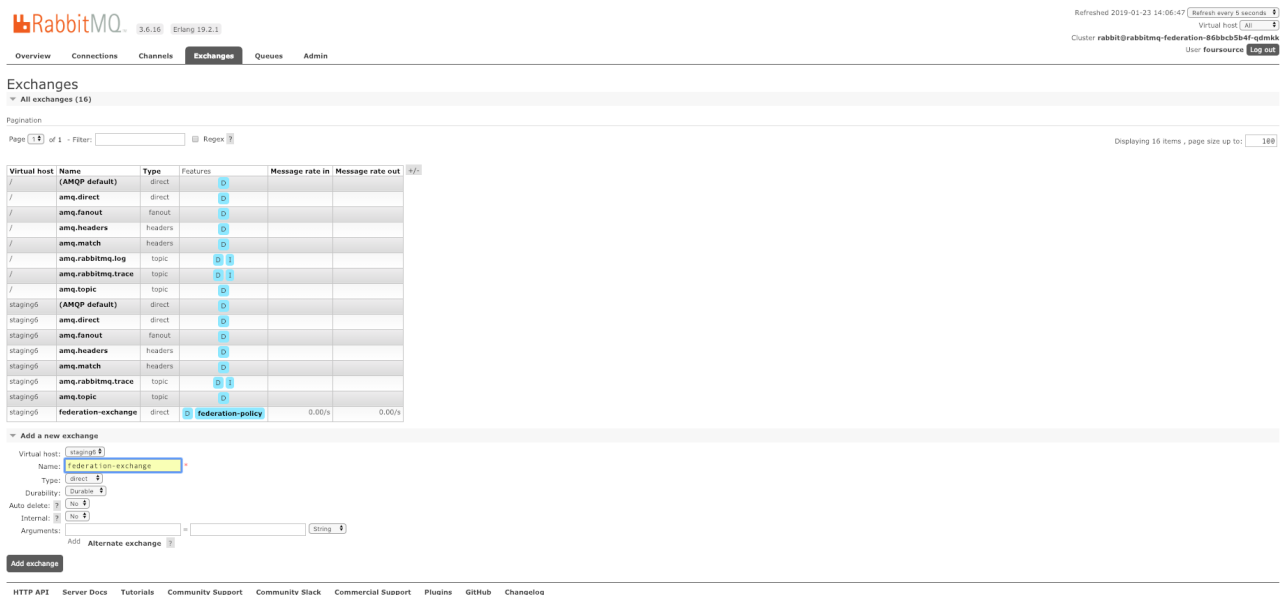
Add permissions on the virtual host “staging6” for the user “federation”



The screenshot shows the RabbitMQ Admin UI with the 'Admin' tab selected. The 'Virtual Host: staging6' page is active, showing the 'Permissions' section. The 'Permissions' table is displayed, showing the permissions for the 'federation' user. The 'Set permission' form is visible below the table, with the 'User' field set to 'federation' and the 'Configure regexp' field set to '*'. The 'Set permission' button is at the bottom of the form.

User	Configure regexp	Write regexp	Read regexp
federation	*	*	*
foursource	*	*	*

Create a new exchange, for example "federation-exchange"



The screenshot shows the RabbitMQ Management UI with the 'Exchanges' tab selected. The top navigation bar includes 'Overview', 'Connections', 'Channels', 'Exchanges', 'Queues', and 'Admin'. The 'Exchanges' section shows a list of 16 exchanges. The 'Add a new exchange' form is visible, with the following fields:

- Virtual host: staging6
- Name: federation-exchange
- Type: direct
- Durability: durable
- Auto delete: No
- Internal: No
- Arguments: (empty)
- Alternate exchange: (empty)

The 'Add exchange' button is at the bottom of the form. The footer contains links for HTTP API, Server Docs, Tutorials, Community Support, Community Slack, Commercial Support, Plugins, GitHub, and Changelog.

Add a new queue, for example "federation"

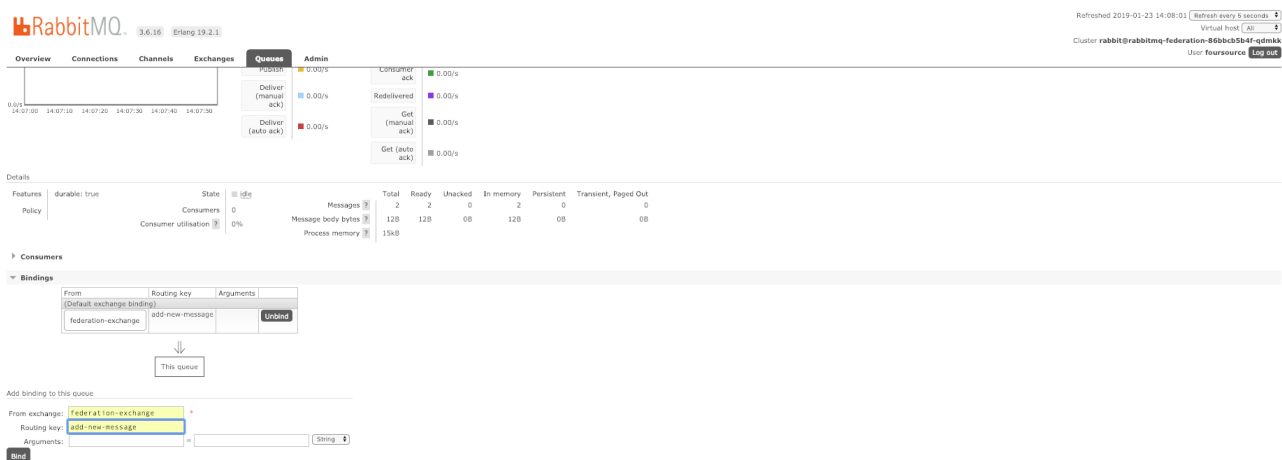


The screenshot shows the RabbitMQ Management UI with the 'Queues' tab selected. The top navigation bar includes 'Overview', 'Connections', 'Channels', 'Exchanges', 'Queues', and 'Admin'. The 'Queues' section shows a list of 1 queue. The 'Add a new queue' form is visible, with the following fields:

- Virtual host: staging6
- Name: federation
- Durability: durable
- Auto delete: No
- Arguments: (empty)

The 'Add queue' button is at the bottom of the form. The footer contains links for HTTP API, Server Docs, Tutorials, Community Support, Community Slack, Commercial Support, Plugins, GitHub, and Changelog.

Add new binding for the queue “federation”, from exchange “federation-exchange” routing key “add-new-message”



The screenshot shows the RabbitMQ Management UI with the 'Bindings' tab selected. The top navigation bar includes 'Overview', 'Connections', 'Channels', 'Exchanges', 'Queues', and 'Admin'. The 'Bindings' section shows a list of bindings. The 'Add binding' form is visible, with the following fields:

- From exchange: federation-exchange
- Routing key: add-new-message
- Arguments: (empty)

The 'Bind' button is at the bottom of the form. The footer contains links for HTTP API, Server Docs, Tutorials, Community Support, Community Slack, Commercial Support, Plugins, GitHub, and Changelog.

Downstream configuration

Create a new federation upstream, example “federation-upstream”, uri “amqp://federation:f4s0u45e@10.42.1.107:5672/staging6”

The screenshot shows the RabbitMQ Admin interface for the 'federation-upstream' virtual host. The 'Federation Upstreams' section is active, displaying a table with one upstream named 'federation-upstream' with URI 'amqp://federation:f4s0u45e@10.42.1.107:5672/staging6'. Below the table, the 'Add a new upstream' form is visible, showing fields for Name, URI, Prefetch count, Reconnect delay, Acknowledgement mode, Trust user ID, Exchange, Max hops, Expiry, Message TTL, HA Policy, and Queue. The 'Add upstream' button is at the bottom left of the form.

Create a new policy, example “federation-policy”, pattern “federation-exchange”, definition “federation-upstream” = “federation-upstream”

The screenshot shows the RabbitMQ Admin interface for the 'federation-policy' policy. The 'Policies' section is active, displaying a table with one policy named 'federation-policy' with pattern 'federation-exchange' and definition 'federation-upstream'. Below the table, the 'Add / update a policy' form is visible, showing fields for Name, Pattern, Apply to, Priority, Definition, HA mode, HA params, HA sync mode, HA mirror promotion on shutdown, HA mirror promotion on failure, Federation upstream set, Federation upstream, Queues, Message TTL, Auto expire, Max length, Max length bytes, Dead letter exchange, Dead letter routing key, Lazy mode, Master Locator, and Exchanges. The 'Add policy' button is at the bottom left of the form.

Check the federation status

The screenshot shows the RabbitMQ Admin interface for the 'federation-status' section. The 'Federation Status' section is active, displaying a table with one upstream named 'federation-upstream' with URI 'amqp://10.42.1.107:5672/staging6' and exchange 'federation-exchange'. The status is 'running'. Below the table, the 'Add policy' button is visible.

Try to send a message from the upstream exchange “federation-exchange” with routing key “add-new-message” to the downstream

RabbitMQ 3.6.16 Erlang 19.2.1

Refreshed 2019-01-23 14:17:46 Refresh every 5 seconds Virtual host: staging6 Cluster: rabbit@rabbitmq User: foursource Log out

Overview Connections Channels **Exchanges** Queues Admin

Exchange: federation-exchange in virtual host staging6

Overview

Message rates **Last minute**



Details

Type: topic
Features: durable: true
Policy:

Bindings

Publish message

Routing key: **add-new-message**
Delivery mode: **1 - Non-persistent**
Headers: **0**
Properties: **0**
Payload: **hello**

Publish message

Delete this exchange

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog

Check if the downstream received the message

RabbitMQ 3.6.16 Erlang 19.2.1

Refreshed 2019-01-23 14:19:20 Refresh every 5 seconds Virtual host: staging6 Cluster: rabbit@rabbitmq-federation-86bcb5b4f-qdmkk User: foursource Log out

Overview Connections Channels Exchanges **Queues** Admin

Exchange: federation-exchange
Routing Key: add-new-message
Redelivered: 0
Properties: delivery_mode: 1
headers: x-received-from: uri: amqp://10.42.1.107:5672/staging6
exchange: federation-exchange
redelivered: false
cluster-name: rabbit@rabbitmq

Payload: 4 bytes
Encoding: string
testa

Message 2

The server reported 1 messages remaining.

Exchange: federation-exchange
Routing Key: add-new-message
Redelivered: 0
Properties: delivery_mode: 1
headers: x-received-from: uri: amqp://10.42.1.107:5672/staging6
exchange: federation-exchange
redelivered: false
cluster-name: rabbit@rabbitmq

Payload: 4 bytes
Encoding: string
testa 2

Message 3

The server reported 0 messages remaining.

Exchange: federation-exchange
Routing Key: add-new-message
Redelivered: 0
Properties: delivery_mode: 1
headers: x-received-from: uri: amqp://10.42.1.107:5672/staging6
exchange: federation-exchange
redelivered: false
cluster-name: rabbit@rabbitmq

Payload: 4 bytes
Encoding: string
hello

Move messages
Delete
Purge
Runtime Metrics (Advanced)

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog