redhat.

AMQ Online - Hackfest

Technical details

Rob Godfrey
22nd January 2019

AMQ Online

Core Concepts

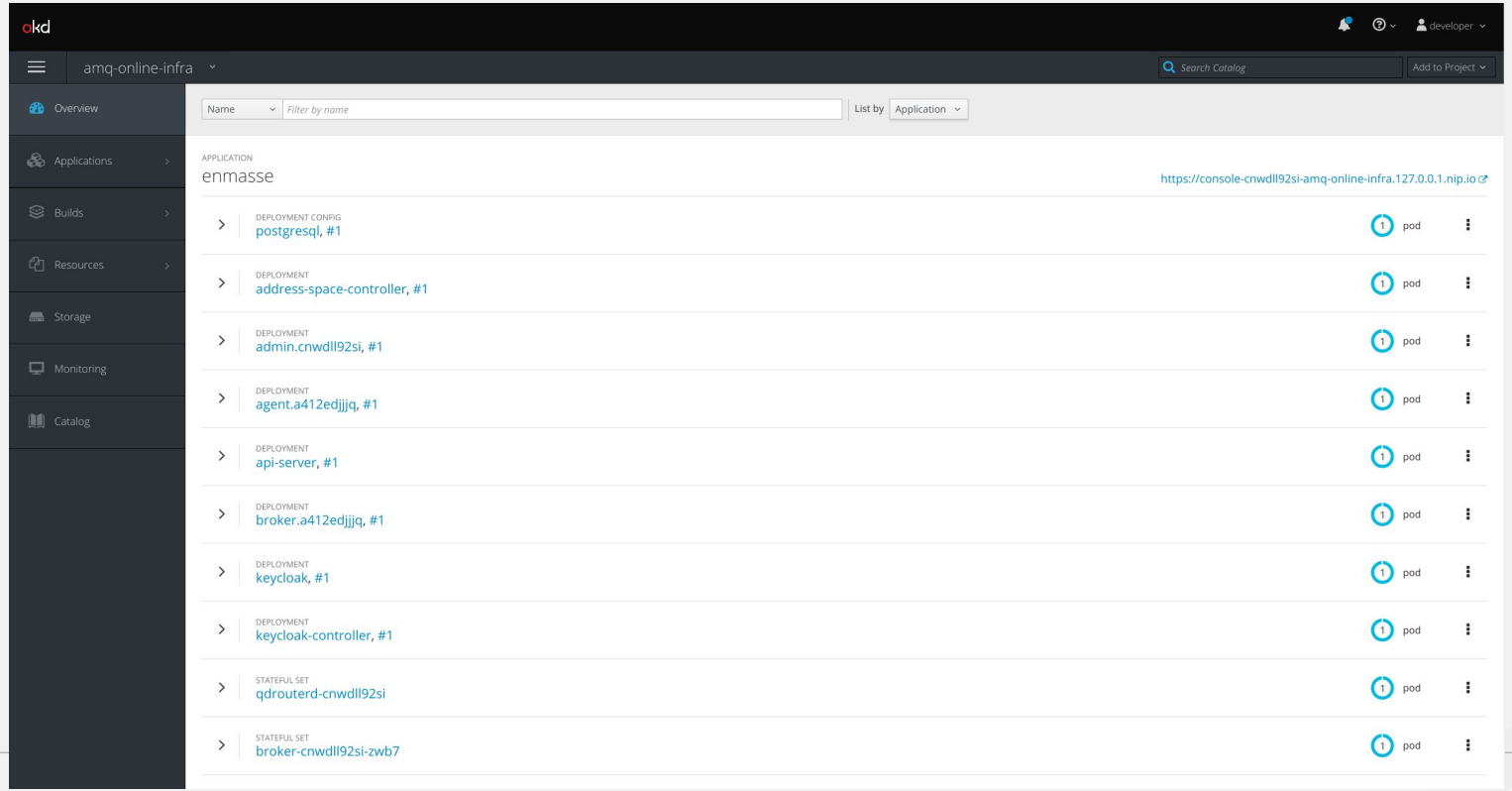
- Address Spaces
  - Address Space Types
  - Address Space Plans
- Addresses
  - Address Types
  - Address Plans
- Users
- Infrastructure Configurations

AMQ Online

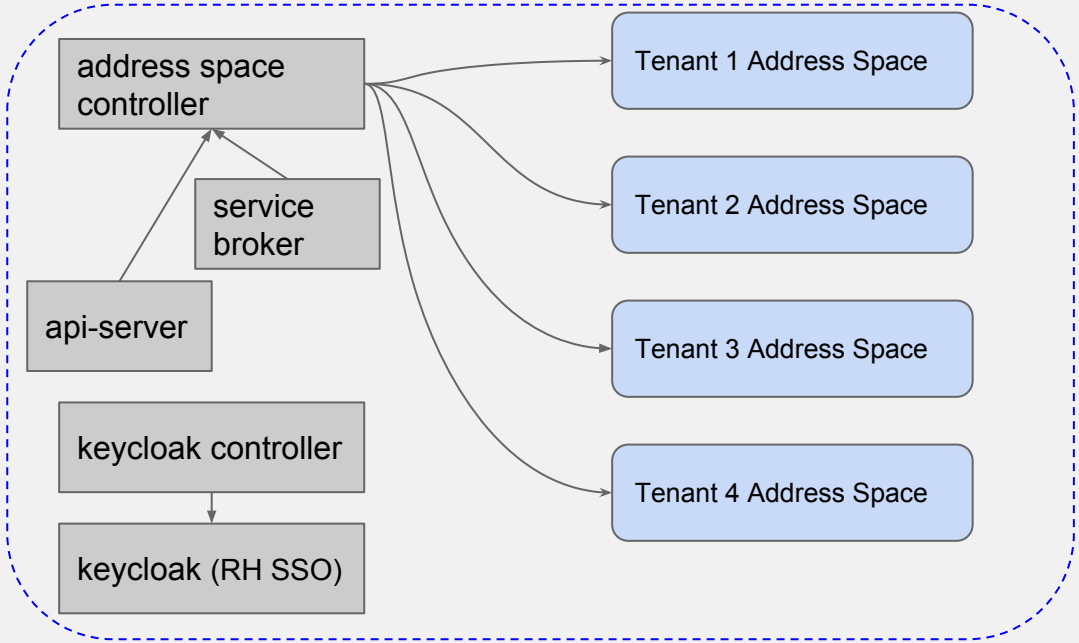
How does it work?

- AMQ Online installation is cluster-wide
  - A single OpenShift namespace owned by the System Admin
- Central infrastructure
  - Provisions / de-provisions messaging components
  - Creates OpenShift services / routes
  - Handles authentication / authorisation
  - Provides API / Custom Resources for messaging tenants
  - Provides Service Broker implementation

AMQ Online System Admin View



AMQ Online Components



AMQ Online Concepts

How does it work? - Address Space Controller

- Detects difference between desired state and actual state
- For each address space - verify existence / plan / infrastructure
  - Where the address space does not exist - create it
- For users (with standard auth service) - verify existence

Managing AMQ Online

- Provides custom resources manipulated through command line tools (oc)
- Resource kinds managed by the service admin
  - AddressSpacePlan
  - AddressPlan
  - BrokeredInfraConfig
  - StandardInfraConfig
- Resource kinds managed by the tenant:
  - AddressSpace
  - Address
  - MessagingUser

AMQ Online Concepts

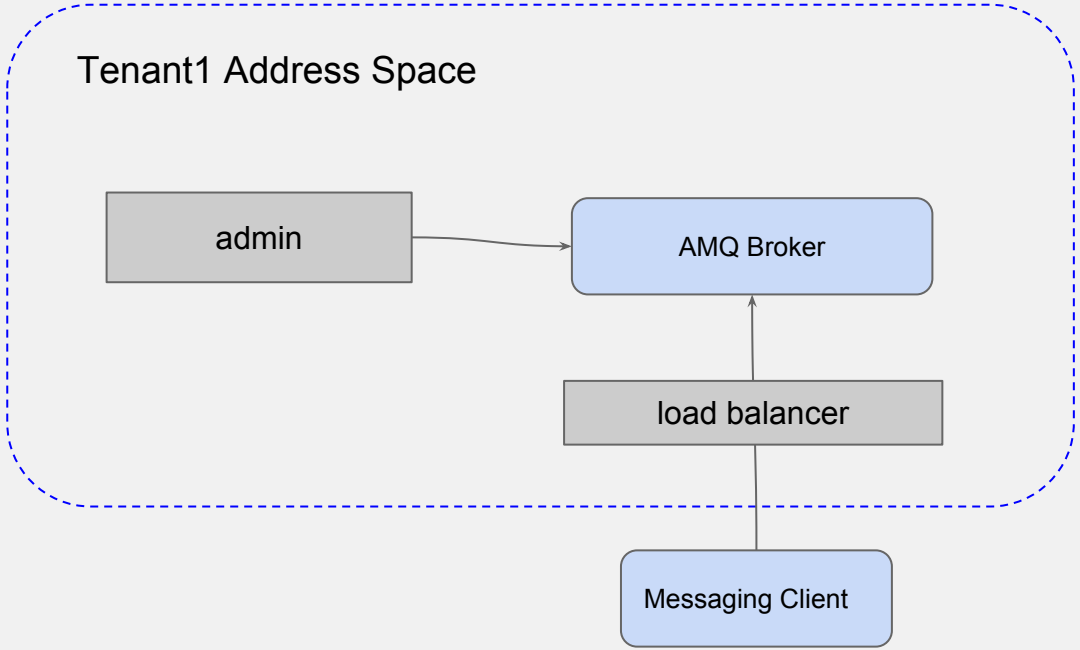
Address space

- An AMQ Online address space resource:

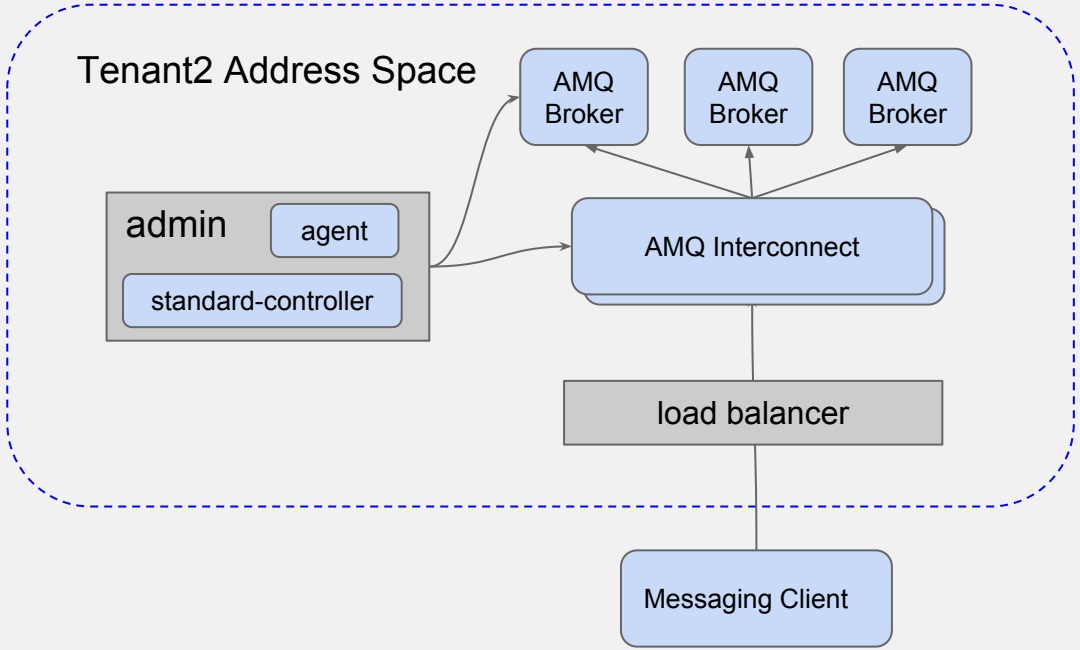
```
apiVersion: enmasse.io/v1beta1
kind: AddressSpace
metadata:
    name: myspace
spec:
    type: standard
    plan: example-plan
```

- On creation of an addressspace, the addresspace controller needs to create the necessary infrastructure

AMQ Online - Brokered address space



AMQ Online - Standard address space



AMQ Online Concepts

Address space plan

- An AMQ Online address space plan resource:

```
apiVersion: admin.enmasse.io/v1beta1
kind: AddressSpacePlan
metadata:
  name: example-plan
  labels:
    app: enmasse
  annotations:
    enmasse.io/defined-by: example-infra
displayName: Example Plan
displayOrder: 0
shortDescription: Example
longDescription: Example Plan
addressSpaceType: standard
    addressPlans:
    - example-queue
    - example-topic
    - example-anycast
    resources:
    - name: router
      max: 2.0
    - name: broker
      max: 2.0
    - name: aggregate
      max: 3.0
```

AMQ Online Concepts

Infrastructure Config

- An AMQ Online infrastructure config resource looks like:

```
apiVersion: admin.enmasse.io/v1beta1
kind: StandardInfraConfig
metadata:
  name: example-infra
spec:
  version: 0.26
  admin:
    resources:
      memory: 256Mi
  broker:
    resources:
      memory: 2Gi
      storage: 100Gi
    addressFullPolicy: PAGE
  router:
    resources:
      memory: 256Mi
    linkCapcity: 1000
    minReplicas: 1
  networkPolicy:
    ingress:
      - from:
        - namespaceSelector:
            component: secure-ns
```

AMQ Online Concepts

Address space

- AMQ Online address space resource:

```
apiVersion: enmasse.io/v1beta1
kind: AddressSpace
metadata:
    name: myspace
spec:
    type: standard
    plan: example-plan
```

- The addresspace controller needs to create one router (with 256Mi memory) and an agent (with 256Mi memory). No brokers will be created until they are necessay.

AMQ Online Concepts

Address

- An AMQ Online address resource:

```
apiVersion: enmasse.io/v1beta1
kind: Address
metadata:
    name: myspace.myqueue
spec:
    address: myqueue
    type: queue
    plan: example-queue
```

- On creation of an address, new infrastructure may need deploying, depending on the address plan

AMQ Online Concepts

Address plan

- An AMQ Online address plan resource:

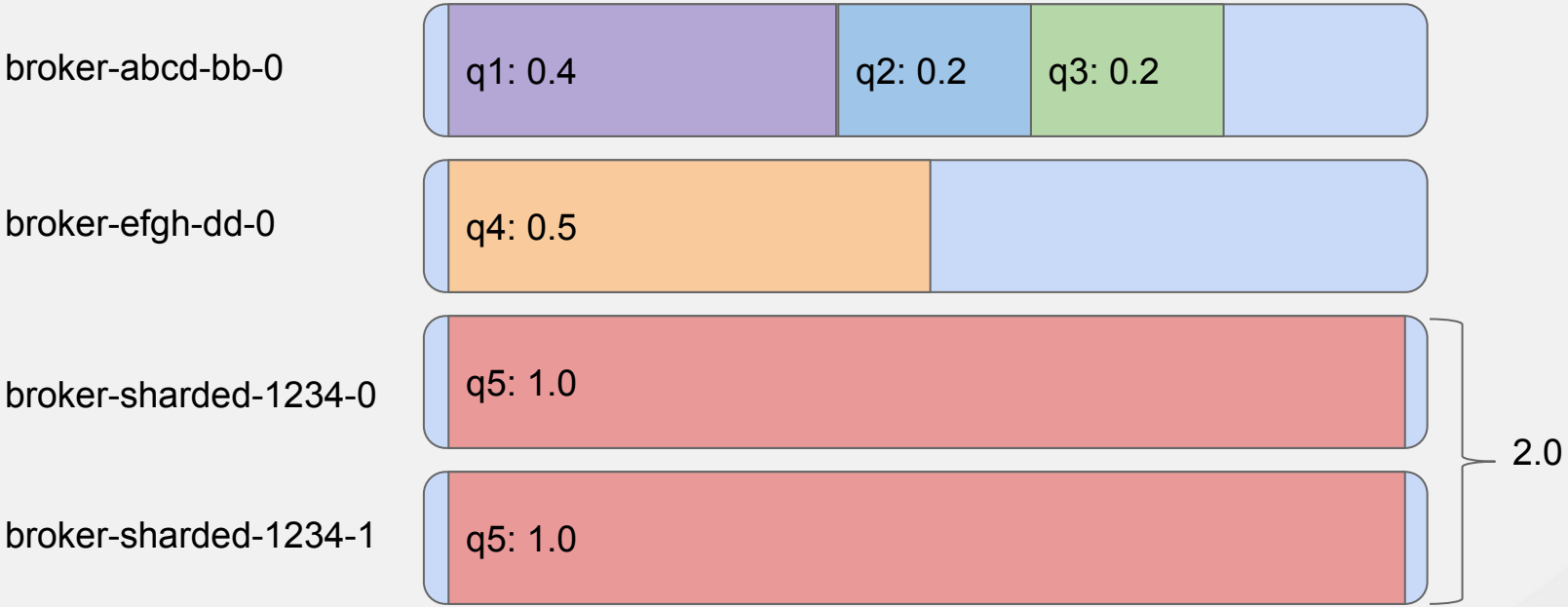
```
apiVersion: admin.enmasse.io/v1beta1
kind: AddressPlan
metadata:
  name: example-queue
  labels:
    app: enmasse
displayName: Example queue plan
displayOrder: 0
shortDescription: An example plan
longDescription: An example plan
addressType: queue
```

```
addressType: queue
requiredResources:
- name: router
  credit: 0.2
- name: broker
  credit: 0.3
```

- On creation of the address, the credits are compared to available resources (of broker and router) and new instances are created if necessary

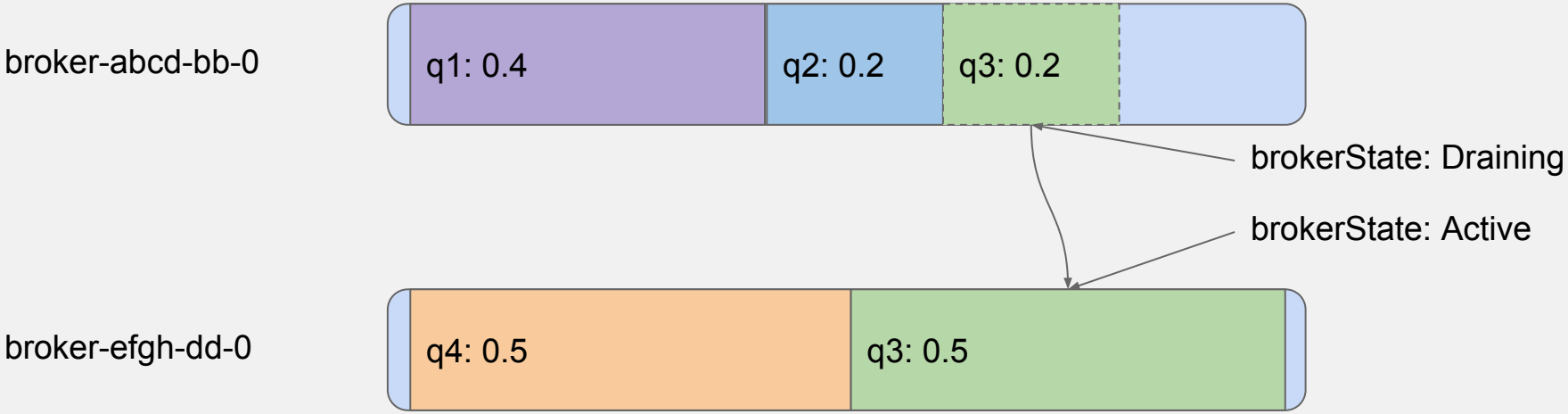
AMQ Online Concepts

Queue scheduling (standard address space)



AMQ Online Concepts

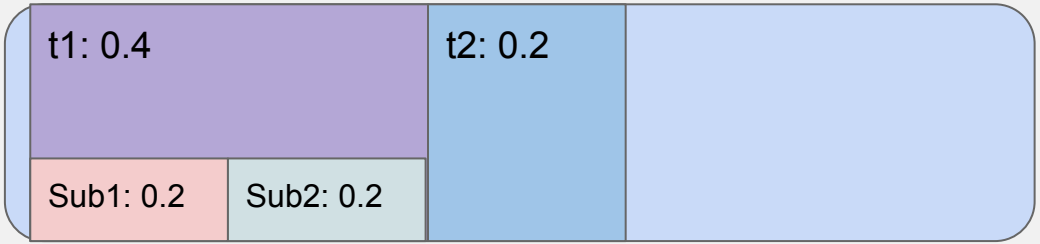
How addresses scale - Queues (Standard Space)



AMQ Online Concepts

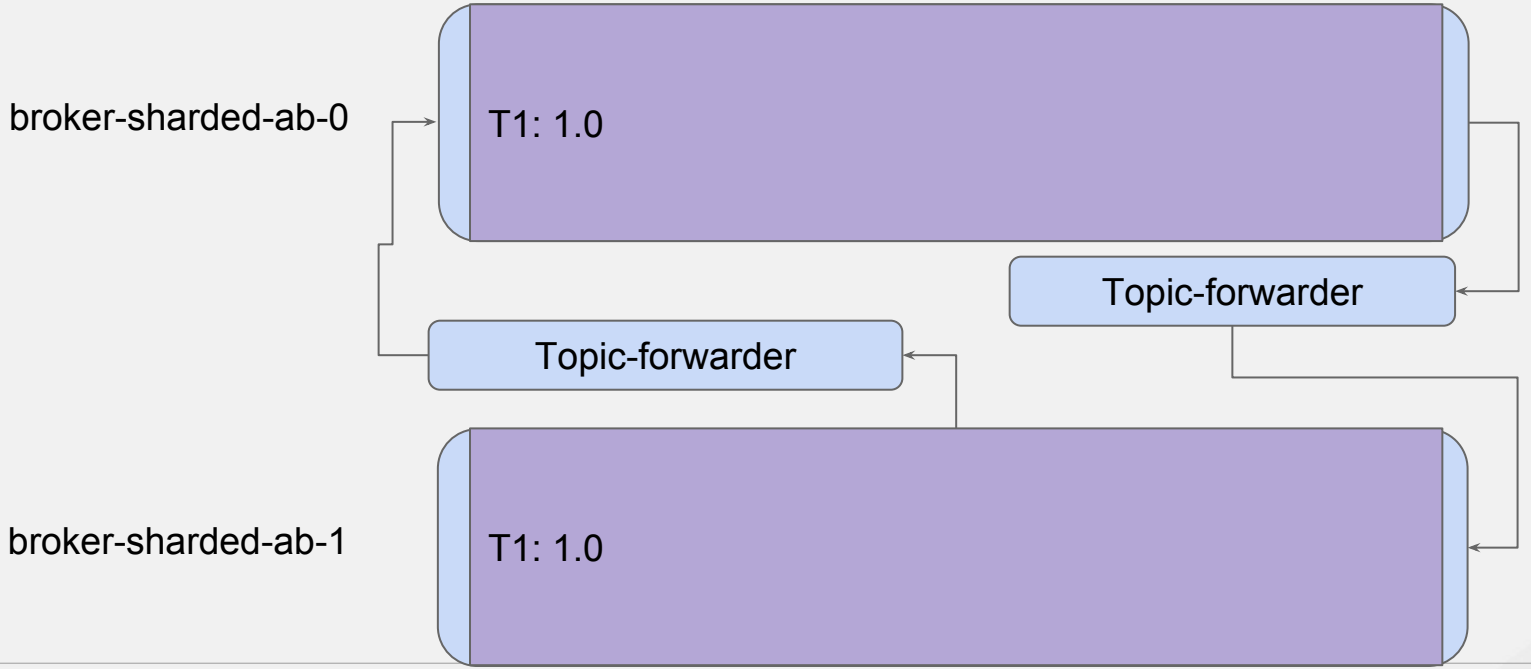
How addresses scale - Topics and durable subs(Standard Space)

broker-abcd-ef-0



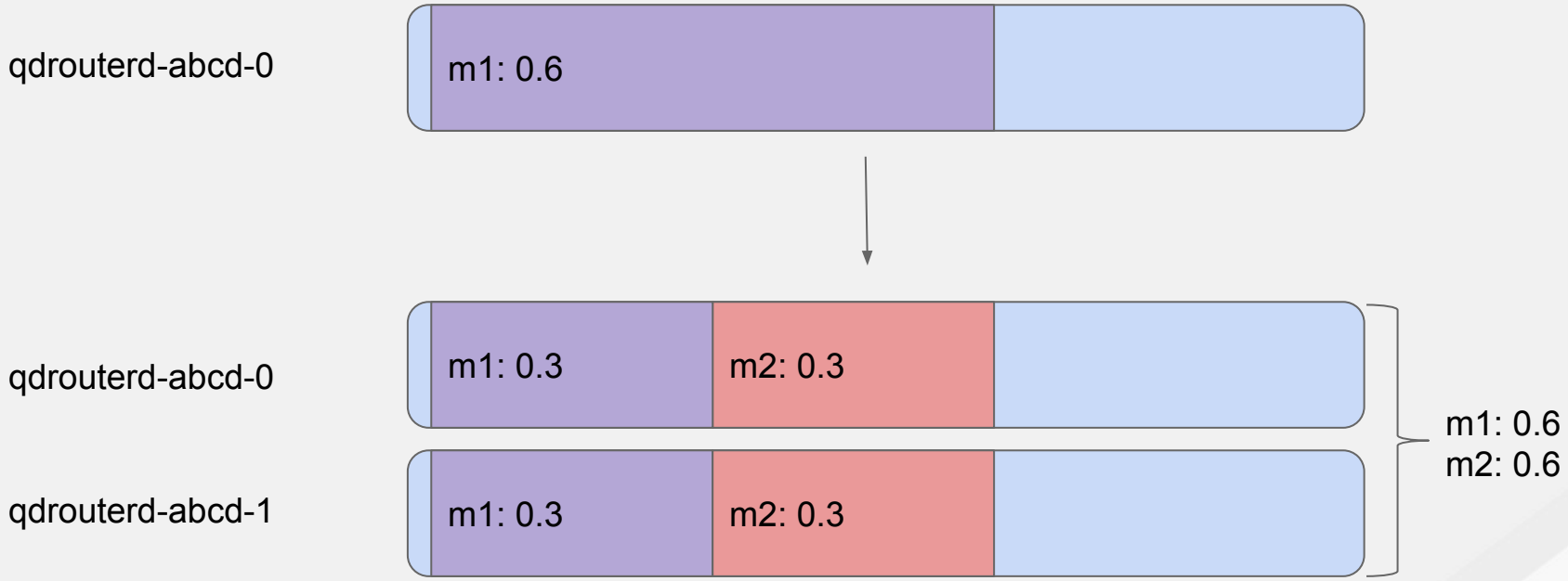
AMQ Online Concepts

How addresses scale - Sharded topics (Standard Space)



AMQ Online Concepts

How addresses scale - Anycast / Multicast (Standard Space)



AMQ Online Concepts

Scaling vs. Sharding

- For queues and topics of size <= 1.0 no sharding possible
- Sharding improves availability - ideally should be independent of size
- Sharding removes ordering guarantees

AMQ Online Concepts

What about the Brokered Space?

- In the Brokered space, all addresses share the broker
- Address plans can only be used to prevent “over provisioning”
- Future releases will trigger broker cluster scaleup
- Future releases may allow plans to set address specific behavior

Managing AMQ Online

- Provides custom resources manipulated through command line tools (oc)
- Resource kinds managed by the service admin
  - AddressSpacePlan
  - AddressPlan
  - BrokeredInfraConfig
  - StandardInfraConfig
- Resource kinds managed by the tenant:
  - AddressSpace
  - Address
  - MessagingUser

Managing AMQ Online (Tenant)

Create an address space

- Save the following YAML data to a file 'space.yaml':

```
apiVersion: enmasse.io/v1beta1
kind: AddressSpace
metadata:
    name: myspace
spec:
    type: standard
    plan: standard-small
```

- Create the address space using the command line:

```
oc create -f space.yaml
```

Managing AMQ Online (Tenant)

List address spaces

- You should now be able to list address spaces:

```
$ oc get addressspaces -o wide

NAME            TYPE       PLAN                       READY     AGE       STATUS
jms-example     brokered   brokered-single-broker   true      2m
vertx-example   standard   standard-small           false     2m        The following stateful
sets are not ready: [qdrouterd-qneo98mfsy]
```

Managing AMQ Online (Tenant)

Listing available plans

```
oc get addressspaceschema standard -o yaml
```

```
oc get addressspaceschema brokered -o yaml
```

Managing AMQ Online (Tenant)

Create an address

- Save the following YAML data to a file 'address.yaml' (**NOTE**: Prefixing the name with the address space name is required to ensure addresses from different address spaces do not collide):

```
apiVersion: enmasse.io/v1beta1
kind: Address
metadata:
    name: myspace.myqueue
spec:
    address: myqueue
    type: queue
    plan: standard-small-queue
```

- Create the address using the command line:

```
oc create -f address.yaml
```

Managing AMQ Online (Tenant)

Get addresses

- You should now be able to list addresses:

```
$ oc get addresses -o wide

NAME                     ADDRESS   ADDRESSSPACE    TYPE      PLAN                    READY     PHASE     AGE       STATUS
jms-example.myqueue      myqueue   jms-example     queue     brokered-queue          true      Active    3m
vertx-example.myqueue    myqueue   vertx-example   queue     standard-small-queue    true      Active    3m
vertx-example.mytopic    mytopic   vertx-example   topic     standard-small-topic    true      Active    3m
```

Managing AMQ Online (Tenant)

Users

- A `User` represents an identity which has access to an `addressspace`.
- `Users` are granted permissions within the `addressspace` to send/receive messages and/or to use the console.
- Users can either be
  - Federated from OpenShift identities
  - Authenticated against an OpenShift service account
  - `Addressspace` specific username/password

Managing AMQ Online (Tenant)

Create a user

- Save the following YAML data to a file user.yaml' (**NOTE**: Prefixing the name with the address space name is required to ensure users from different address spaces do not collide):

```
apiVersion: enmasse.io/v1beta1
kind: MessagingUser
metadata:
    name: myspace.user1
spec:
    username: user1
    authentication:
      type: password
      password: aGFja2Zlc3Q=
    authorization:
      - operations: [“send”, “recv”],
        addresses: [“myqueue”]
```

- Create the user using the command line:

```
oc create -f user.yaml
```

Managing AMQ Online (Tenant)

Create a service account user

- Save the following YAML data to a file user.yaml' (**NOTE**: Prefixing the name with the address space name is required to ensure users from different address spaces do not collide):

```
apiVersion: enmasse.io/v1beta1
kind: MessagingUser
metadata:
    name: myspace.user2
spec:
    username: system:serviceaccount:myapp:sa1
    authentication:
      type: serviceaccount
    authorization:
      - operations: [“send”, “recv”],
        addresses: [“myqueue”]
```

- Create the user using the command line:

```
oc create -f user.yaml
```

Managing AMQ Online (Tenant)

Get users

- You should now be able to list users:

```
oc get messagingusers
```

```
NAME                    USERNAME                                AGE
vertx-example.client    client                                  4m
vertx-example.tenant1   tenant1                                 4m
jms-example.client      system:serviceaccount:myapp:default     4m
jms-example.tenant1     tenant1                                 4m
```

Getting AMQ Online

Downstream builds

- AMQ Online
  - For pre-release images, configure registry brew-pulp-docker01.web.prod.ext.phx2.redhat.com:8888 (requires VPN)
  - Source of install.zip: https://github.com/jboss-container-images/amq-online-images/tree/amq-online-10-dev/templates

Getting AMQ Online

Upstream Project

- EnMasse
  - http://enmasse.io/
  - GitHub Project: https://github.com/EnMasseProject/enmasse

THANK YOU


plus.google.com/+RedHat


facebook.com/redhatinc


linkedin.com/company/red-hat


twitter.com/RedHat


youtube.com/user/RedHatVideos