Messaging as a Service | AMQ ONLINE

Designing cloud-based messaging services for business and IoT

Rob Godfrey
Engineering Manager - AMQ Online
7th May 2019
Introduction

Rob Godfrey - Engineering Manager AMQ Online

- Red Hat since January 2017
- Previously Distinguished Engineer at JPMorganChase
Agenda

- Why is messaging important (but hard)?
- What do we mean by MaaS (and how does it help)?
- An Open Source MaaS solution - EnMasse
- The role of the Service Administrator
- Developing an application for AMQ Online
- Messaging and IoT
- Roadmap
The Software

- **EnMasse**
  - [https://enmasse.io](https://enmasse.io)
  - [https://github.com/EnMasseProject/enmasse](https://github.com/EnMasseProject/enmasse)

- **Hono**
  - [https://www.eclipse.org/hono/](https://www.eclipse.org/hono/)
  - [https://github.com/eclipse/hono](https://github.com/eclipse/hono)
THE PROBLEM(S)
Why use Messaging in 2019?

- Scalable distributed applications require communication between processes
- HTTP is not always the best option
- Reactive applications built on asynchronous message passing
- Integrate with existing applications
Setting up Messaging is Hard

- Complicated configuration
  - Security (Authn, Authz, TLS)
  - HA, Storage
  - Inter-broker links...

- Not cloud friendly
  - Large monolithic processes,
  - Vertical scaling,
  - Long startup times

- Sizing is an art
  - Users and Engineers rarely speak the same language
What about existing services?

- Tied to your provider
- Not available on-prem
- Proprietary APIs
- Non-standard protocols
MESSAGING AS A SERVICE
Messaging as a Service

- Messaging as Infrastructure
- Separate the use of Messaging from management of infrastructure
- Create messaging “on demand”
- Rapidly scale from small applications to huge volumes
- Declaratively define application messaging requirements
- Optimise resource usage by sharing infrastructure
User Personas

Service Provider
- Service Admin
- Service Operator

Tenant
- Application Owner
- Application Monitor
AMQ ONLINE
AMQ Online Architecture

- AMQ Online Operator
- Cluster-wide services
- Per-tenant services
- Messaging Infrastructure

Single infrastructure namespace

- Tenant namespace
- Tenant namespace
- Tenant namespace
Messaging as Resources

- Configuration of the service achieved using Custom Resources
  - Define messaging “plans” a tenant can choose from
  - Plans define sizing but also auth scheme, HA requirements, etc
- Tenants define their requirements using Custom Resources
  - An instance of messaging using a particular plan
SETTING UP A MESSAGING SERVICE
Configuring your AMQ Online Service

- AddressSpacePlan
- AddressPlan
- BrokeredInfraConfig
- StandardInfraConfig
- AuthenticationService
- ConsoleService
The AMQ Online Model

Address Space

- A collection of messaging endpoints (such as queues or topics) which can be accessed with the same set of credentials
- The unit of provisioning of AMQ Online

Address

- A messaging endpoint (such as a queue or topic) within an Address Space to which messages can be sent, or messages can be received.
The AMQ Online Model

Address Space Types

AMQ Online supports multiple *types* of Address Space.

- Different address space types
  - may support differing types of messaging endpoints
  - may support differing messaging protocols
  - may have different abilities to scale to high workloads
- The type reflects the underlying technology used
- The supported types are currently not user-definable
The AMQ Online Model

“Standard” Address Space Type
The AMQ Online Model

“Brokered” Address Space Type
The AMQ Online Model

Address Space Plans

The service admin defines a number of plans for each Address Space type

- A plan defines *how much* messaging you get
- A plan may enable or disable some features
The AMQ Online Model

Address Types

Each Address Space Type supports its own set of Address Types.

- Different address types represent different types of messaging endpoint, e.g. Queue, Topic
- The supported types are not user-definable
The AMQ Online Model

Address Plans

The service admin defines a number of plans for each Address type

- A plan defines *how much* messaging you get in terms of resource usage
- A plan may enable fine grained configuration of the address

The service admin defines which address plans are available in each address space plan
Configuring your AMQ Online Service

- AddressSpacePlan
- AddressPlan
- BrokeredInfraConfig
- StandardInfraConfig
- AuthenticationService
- ConsoleService
AMQ Online Resources

Address Space Plans

- An AMQ Online address space plan resource:

  ```yaml
  apiVersion: admin.enmasse.io/v1beta2
  kind: AddressSpacePlan
  metadata:
    name: example-plan
    labels:
      app: enmasse
  addressSpaceType: standard
  infraConfigRef: example-infra

  addressPlans:
  - example-queue
  - example-topic
  - example-anycast

  resourceLimits:
    router: 2.0
    broker: 2.0
    aggregate: 3.0
  ```
AMQ Online Resources

Infrastructure Config

- An AMQ Online infrastructure config resource looks like:

```
apiVersion: admin.enmasse.io/v1beta1
type: StandardInfraConfig
metadata:
  name: example-infra
spec:
  admin:
    resources:
      memory: 256Mi
  broker:
    resources:
      memory: 2Gi
      storage: 100Gi
    addressFullPolicy: PAGE
  router:
    resources:
      memory: 256Mi
      linkCapacity: 1000
      minReplicas: 1
    podTemplate:
      spec:
        affinity: {} 
        tolerations: []
    networkPolicy:
      ingress:
        from:
        - namespaceSelector:
          component: secure-ns
```
AMQ Online Resources

Address Plans

● An AMQ Online address plan resource:

```yaml
apiVersion: admin.enmasse.io/v1beta2
kind: AddressPlan
metadata:
  name: example-queue
  labels:
    app: enmasse
addressType: queue
```

● On creation of the address, the credits are compared to available resources (of broker and router) and new instances are created if necessary.
Effect of Address Resource Usages

- **broker-abcd-bb-0**
  - q1: 0.2
  - q2: 0.2
  - q4: 0.25

- **broker-efgh-dd-0**
  - q1: 0.2
  - q3: 0.2
  - q4: 0.25

- **broker-jklm-ee-0**
  - q5: 1.0

- **broker-npqr-gg-0**
  - q5: 1.0

Total: 2.0
An AMQ Online ‘standard’ authentication service resource looks like:

```yaml
apiVersion: admin.enmasse.io/v1beta1
kind: AuthenticationService
metadata:
  name: example-authservice
spec:
  type: standard
  standard:
    certificateSecret:
      name: my-authservice-cert
    credentialsSecret:
      name: my-authservice-admin-creds
---
storage:
  type: ephemeral | persistent-claim
  size: 5Gi
  class: glusterfs
resources: {}
datasource:
  type: postgresql // default is h2 (embedded)
  host: postgresql.example.com
  port: 5632
  database: authservice-db
  credentialsSecret:
    name: my-db-credentials
```
DEVELOPING AN APPLICATION
User Personas

Service Provider

Service Admin

Service Operator

Tenant

Application Owner

Application Monitor
The AMQ Online Model

Address Space
- A collection of messaging endpoints (such as queues or topics) which can be accessed with the same set of credentials
- The unit of provisioning of AMQ Online

Address
- A messaging endpoint (such as a queue or topic) within an Address Space to which messages can be sent, or messages can be received.
The AMQ Online Model

Users

- A User represents an identity which has access to an addressspace.
- Users are granted permissions within the addressspace to send/receive messages.
- Users can have one of two forms of authentication:
  - Addressspace specific username/password
  - an OpenShift service account token
Defining Your Messaging Needs

- AddressSpace
- Address
- MessagingUser
AMQ Online Resources

Address Space

- An AMQ Online address space:

```yaml
apiVersion: enmasse.io/v1beta1
type: brokered
kind: AddressSpace
metadata:
  name: myspace
spec:
  type: brokered
  plan: brokered-single-broker
  authenticationService:
    Name: standard-authsvc
  endpoints:
    - name: messaging
      service: messaging
      exports:
        - kind: ConfigMap
          name: messaging-config
          cert:
            provider: openshift
```
AMQ Online Resources

Address

- An AMQ Online address space:

```yaml
apiVersion: enmasse.io/v1beta1
kind: Address
metadata:
  name: myspace.myqueue
spec:
  address: myqueue
  type: queue
  plan: brokered-queue
```
AMQ Online Resources

User

- An AMQ Online user:

```yaml
apiVersion: enmasse.io/v1beta1
class: MessagingUser
metadata:
  name: myspace.user2
spec:
  username: system:serviceaccount:myapp:sa1
  authentication:
    type: serviceaccount
  authorization:
    - operations: ["send", "recv"],
      addresses: ["myqueue"]
```
MESSAGING AND IoT
WHAT IS ECLIPSE HONO™?
AMQ ONLINE & IoT

- MQTT
- HTTP
- LoRaWAN
- CoAP

Interconnect

Broker

Customer Applications

Red Hat
Data Grid

Red Hat
AMQ

Red Hat
OpenShift
AMQ ONLINE & IoT

MQTT
HTTP
LoRaWAN
CoAP

Kafka
Interconnect

Customer Applications

Red Hat Data Grid
Red Hat AMQ
Red Hat OpenShift
INTEGRATION

Customer Applications
- 3scale
- FUSE
- Decision Manager
- Custom Code

MQTT
HTTP
LoRaWAN
CoAP

Broker
Interconnect
Streams

Red Hat AMQ

Red Hat Integration
EVOLVING AMQ ONLINE
COMING SOON(ISH)

- Managed AMQ Streams / Kafka topics
- Shared Infrastructure for improved density
- Bridging between address spaces - wide area messaging
- Improved Auto-scaling
- Cross-cluster messaging / Cloud-bursting
- IoT integration
  - Kafka integration
  - Bring your own adapters
- Tooling for Service Admins
THANK YOU

linkedin.com/company/Red-Hat
youtube.com/user/RedHatVideos
facebook.com/RedHatInc
twitter.com/RedHat