mbox-operator

Release 0.0.1

Jan 26, 2021

Contents:

CHAPTER 1

User Guide

1.1 koji-hub

This page documents the usage of koji-hub crd.

1.1.1 Dependencies

Koji-Hub Custom Resource Definition (CRD)

Koji-hub depends the following external components:

- postgresql
- fedora messaging

The operator does not deploy those components and it expects those to be are already available/deployed.

Sample deployment files are provided for development/example purposes:

- postgresql
- rabbitmq

1.1.2 Parameters

Name	Default Value	Type
image	quay.io/fedora/koji-hub:latest	string
replicas	1	int
persistent	true	boolean
host	koji-hub	string
configmap	koji-hub	string
ca_cert_secret	koji-hub-ca-cert	string
service_cert_secret	koji-hub-service-cert	string
postgres_secret	postgres	string
http_enabled	true	boolean
https_enabled	true	boolean
topic_prefix	mbox_dev	string
fedora_messaging_url		string
messaging_cert_cm	koji-hub-msg	string
ingress_backend	nginx	string
mbox	“”	string
httpd_pvc_name	koji-hub-httpd-pvc	string
httpd_pvc_size	1Gi	string
mnt_pvc_name	koji-hub-mnt-pvc	string
mnt_pvc_size	10Gi	string
web_client_cert_secret	koji-hub-web-client-cert	string
web_client_username	kojihub	string
admin_client_cert	koji-hub-admin-cert	string
admin_username	kojiadmin	string

image

The the full qualified image name to pull koji-hub from.

replicas

The amount of koji-hub replicas to deploy.

persistent

A boolean flag to enable/disable pvc creation.

Note: I will not create any external volumes if set to false.

host

The koji-hub hostname to be used on several config files and certificates such as httpd.

This property should be set to the public base url of koji on production environments.

configmap

The configmap name to use when deploying koji-hub.

This configmap object contains configuration files that are mounted in koji-hub pod filesystem.

ca_cert_secret

The root CA secret name to use or create.

It will skip its creation (self signed) if one is already present.

Secret format:

```
apiVersion: v1
kind: Secret
  metadata:
    name: mysecret
    namespace: default
    labels:
      app: koji-hub
  data:
    csr: -|
      fillme
    cert: -|
      fillme
    key: -|
      fillme
```

service_cert_secret

The httpd service secret name to use or create.

It will skip its creation (self signed) if one is already present.

It needs to be created and signed using the root CA certificate and private key.

Secret format:

```
apiVersion: v1
kind: Secret
metadata:
  name: myservice
  namespace: default
  labels:
    app: koji-hub
type: kubernetes.io/tls
data:
  tls.crt: -|
    fillme
  tls.key: -|
    fillme
```

postgres_secret

Postgresql secret used by koji-hub to connect to a psql instance.

Deployment will fail if this secret is not present.

Secret format:

```
apiVersion: v1
kind: Secret
metadata:
  name: postgres
  labels:
    app: postgres
data:
  POSTGRES_HOST: fillme
  POSTGRES_DB: fillme
  POSTGRES_USER: fillme
  POSTGRES_PASSWORD: fillme
```

http_enabled

A boolean flag that enables/disables http connections.

https_enabled

A boolean flag that enables/disables https connections.

topic_prefix

The fedora messaging topic prefix to use koji-hub config.

fedora_messaging_url

The fedora messaging url to use in koji-hub.

This is a required property with no default value.

messaging_cert_cm

A config map that contains fedora messaging certs to be mounted in koji-hub pod filesystem.

Those files are used to authenticate koji-hub to a fedora-messaging instance.

Config map format:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: koji-hub-msg
  namespace: default
  labels:
    app: koji-hub
data:
  koji.ca: |-
    fillme
  koji.crt: |-
```

(continues on next page)

(continued from previous page)

```
    fillme
  koji.key: |-
    fillme
```

ingress_backend

The kubernetes ingress backend to use when creating an ingress resource for koji-hub.

Available choices:

- nginx
- openshift

httpd_pvc_name

Name of the PersistentVolumeClaim for httpd server koji-hub will use.

If provided PVC doesn't exists, it creates its own.

httpd_pvc_size

Size of the PersistentVolumeClaim for httpd server koji-hub will create.

If httpd_pvc_name exists, this value is ignored.

mnt_pvc_name

Name of the PersistentVolumeClaim koji-hub will use.

If provided PVC doesn't exists, it creates its own.

mnt_pvc_size

Size of the PersistentVolumeClaim koji-hub will create.

If mnt_pvc_name exists, this value is ignored.

mbox

A Mbox resource name to retrieve shared data from (pvc volume and shared certs).

Koji-builder will use the following vars if this property is missing to create/use those shared resources:

- mnt_pvc_name (shared koji mnt volume)
- ca_cert_secret (root ca secret)
- postgres_secret (PSQL secret)

web_client_cert_secret

The koji-web secret name to use or create for koji-hub authentication.

It will skip its creation (self signed) if one is already present.

It needs to be created and signed using the root CA certificate and private key.

It should have one key "client.pem" to store both private key and public certificate.

The certificate's CN field will be used as username during authentication.

Secret format:

```
apiVersion: v1
kind: Secret
metadata:
  name: koji-hub-web-client-cert-secret
  namespace: default
  labels:
    app: koji-hub
data:
  client.pem: -|
    fillme
```

web_client_username

Koji web client username to be used when authenticating to koji-hub.

This property will be ignored if not using a self-signed certificate generated by the operator.

admin_client_cert

The admin koji-hub secret name to use or create for koji-hub authentication as the admin user.

It will skip its creation (self signed) if one is already present.

It needs to be created and signed using the root CA certificate and private key.

It should have one key "client.pem" to store both private key and public certificate.

The certificate's CN field will be used as username during authentication.

Secret format:

```
apiVersion: v1
kind: Secret
metadata:
  name: koji-hub-admin-client-cert-secret
  namespace: default
  labels:
    app: koji-hub
data:
  client.pem: -|
    fillme
```

admin_username

The koji-hub admin username.

The username should match the CN field from the "admin_client_cert" certificate.

1.1.3 Usage

Upstream file can be found here

Create a file containing the following content (modify as needed):

```
apiVersion: apps.fedoraproject.org/v1alpha1
kind: MBKojiHub
metadata:
  name: example
  labels:
    app: mbox
spec:
  image: quay.io/fedora/koji-hub:latest
  replicas: 1
  persistent: true
  host: koji-hub
  configmap: koji-hub
  ca_cert_secret: koji-hub-ca-cert
  service_cert_secret: koji-hub-service-cert
  postgres_secret: postgres
  http_enabled: true
  https_enabled: true
  topic_prefix: mbox_dev
  fedora_messaging_url: amqps://koji@messaging.url
  messaging_cert_cm: koji-hub-msg
  ingress_backend: nginx
```

Run the following command to create a koji-hub resource:

```
kubectl apply -f koji-hub-cr.yaml
```

You can check its status by running:

```
kubectl get mbkojihub/example -o yaml
```

1.2 koji-builder

This page documents the usage of koji-builder crd.

1.2.1 Description

MBox utilizes koji-builder to create a new repositories when needed.

1.2.2 Dependencies

Koji-Builder Custom Resource Definition (CRD)

Koji builder depends on koji-hub. This component is deployed as part of the operator deployment.

1.2.3 Parameters

Name	Default Value	Type
image	quay.io/fedora/koji-builder:latest	string
replicas	1	int
configmap	koji-builder-configmap	string
cacert_secret	koji-hub-ca-cert	string
client_cert_secret	koji-builder-client-cert	string
koji_hub_user	'koji-builder.mbox.dev'	string
koji_hub_host	'koji-hub'	string
koji_hub_port	8443	string
max_jobs	5	int
vendor	MBox	string
host_archs	["x86_64"]	[string]
host_name	koji-hub:8443	string
ssl_verify	true	boolean
shared_pvc	koji-hub-mnt-pvc	string
mbox	""	string
host_channels	["default", "createrepo"]	[string]

image

The the full qualified image name to pull koji-builder from.

replicas

The amount of koji-builder replicas to deploy.

configmap

The configmap name to use when deploying koji-builder.

This configmap object contains configuration files that are mounted in koji-builder pod filesystem.

cacert_secret

The root CA secret name to use.

If not provided it uses the one generated by koji-hub (self-signed).

client_cert_secret

The koji-hub client secret name to use or create.

It will skip its creation (self signed) if one is already present.

It needs to be created and signed using the root CA certificate and private key.

Secret format:

```
apiVersion: v1
kind: Secret
metadata:
  name: myservice
  namespace: default
  labels:
    app: koji-builder
type: kubernetes.io/tls
data:
  tls.crt: -|
    fillme
  tls.key: -|
    fillme
  tls.pem: -|
    This is a combination of tls.key and tls.crt separated by '\n' and encoded in␣
↪base64
    Example: "{{ (lookup('file', 'client_key.pem') + '\n' + lookup('file', 'client_
↪cert.pem')) | b64encode }}"
```

koji_hub_user

User to use when authenticating with koji-hub.

koji_hub_host

Hostname of the koji-hub server instance that koji-builder will connect to.

koji_hub_port

Port of the koji-hub server instance that koji-builder will connect to.

max_jobs

Max concurrent jobs the koji-builder should run in parallel.

vendor

Koji-builder vendor used in rpm headers.

host_archs

The list of supported koji builder host architectures.

Defaults to a single architecture of "x86_64".

host_name

The koji host name to be used when creating a koji host in koji-hub.

The name should be a qualified hostname address.

This name should be unique in koji and is also used as the koji-build client certificate CN field.

ssl_verify

A boolean flag used to tell koji-builder to verify ssl certs when connectiong to koji-hub.

It should be set to false if using self-signed certs.

shared_pvc

Name of the shared PersistentVolumeClaim koji-builder will use.

host_channels

A list of channels to add the koji-host to.

Defaults to the following channels: "default" and "createrepo".

mbox

A Mbox resource name to retrieve shared data from (pvc volume and shared certs).

Koji-builder will use the following vars if this property is missing:

- mnt_pvc_name (shared koji mnt volume)
- cacert_secret (root ca secret)

1.2.4 Usage

Upstream file can be found here

Create a file containing the following content (modify as needed):

```
apiVersion: apps.fedoraproject.org/v1alpha1
kind: MBKojiBuilder
metadata:
  name: example
  labels:
    app: mbox
spec:
  image: quay.io/fedora/koji-builder:latest
```

(continues on next page)

(continued from previous page)

```
  replicas: 1
  configmap: koji-builder-configmap
  cacert_secret: koji-hub-ca-cert
  client_cert_secret: koji-builder-client-cert
  koji_hub_user: 'koji-builder.mbox.dev'
  koji_hub_host: 'koji-hub'
  koji_hub_port: 8443
  max_jobs: 5
  vendor: MBox
  host_archs:
    - x86_64
  host_channels:
    - default
    - createrepo
  host_name: mbbox.default
  ssl_verify: false
  shared_pvc: koji-hub-mnt-pvc
```

Run the following command to create a koji-builder resource:

```
kubectl apply -f koji-builder-cr.yaml
```

You can check its status by running:

```
kubectl get mbkojibuilder/example -o yaml
```

1.3 kojira

This page documents the usage of kojira crd.

1.3.1 Description

Kojira is a stand-alone process which handles buildroot repos.

It is deployed in its own pod and shares a repo volume with other components such as koji-builder and koji-hub.

1.3.2 Dependencies

Kojira Custom Resource Definition (CRD)

Kojira depends on koji-hub. This component is deployed as part of the operator deployment.

1.3.3 Parameters

Name	Default Value	Type
image	quay.io/fedora/kojira:latest	string
replicas	1	int
configmap	kojira-config	string
hub_usermname	kojira	string
hub_host	koji-hub:8443	string
src	no	string
max_repo_tasks	15	int
repo_tasks_limit	15	int
shared_pvc	koji-hub-mnt-pvc	string
cacert_secret	koji-hub-ca-cert	string
client_cert_secret	kojira-client-cert	string
admin_secret	kojira-admin-cert	string
mbox	“”	string

image

The full qualified image name to pull kojira from.

replicas

The amount of kojira’s replicas to deploy.

configmap

The configmap name to use when deploying kojira.

This configmap object contains configuration files that are mounted in kojira pod filesystem.

hub_username

User to use when authenticating with koji-hub.

hub_host

Koji-hub hostname (includes port) for hub connections.

src

Indicates if kojira should include srpms in repos.

Possible choices are “yes” or “no”.

max_repo_tasks

The maximum/limit of newRepo tasks.

repo_tasks_limit

The maximum/limit of overall tasks.

shared_pvc

Name of the shared PersistentVolumeClaim kojira will use.

cacert_secret

The root CA secret name to use.

If not provided it uses the one generated by koji-hub (self-signed).

client_cert_secret

The koji-hub client secret name to use or create.

It will skip its creation (self signed) if one is already present.

It needs to be created and signed using the root CA certificate and private key.

Secret format:

```
apiVersion: v1
kind: Secret
metadata:
  name: myservice
  namespace: default
  labels:
    app: koji-builder
type: kubernetes.io/tls
data:
  tls.crt: -|
    fillme
  tls.key: -|
    fillme
  tls.pem: -|
    This is a combination of tls.key and tls.crt separated by '\n' and encoded in␣
↪base64
    Example: "{{ (lookup('file', 'client_key.pem') + '\n' + lookup('file', 'client_
↪cert.pem')) | b64encode }}"
```

admin_secret

A koji admin secret certificate.

An admin level certificate is needed to add all required permissions to the kojira user.

mbox

A Mbox resource name to retrieve shared data from such as shared pvc name.

Koji-builder will use the following var if this property is missing:

- shared_pvc (shared koji mnt volume)
- cacert_secret (root ca secret)

1.3.4 Usage

Upstream file can be found here

Create a file containing the following content (modify as needed):

```
apiVersion: apps.fedoraproject.org/v1alpha1
kind: MBKojira
metadata:
  name: mb-kojira
  labels:
    app: mb-kojira
spec:
  replicas: 1
  image: quay.io/fedora/kojira:latest
  configmap: kojira-config
  hub_username: kojira
  hub_host: koji-hub:8443
  src: 'no'
  max_repo_tasks: 15
  repo_tasks_limit: 15
  cacert_secret: koji-hub-ca-cert
  client_cert_secret: kojira-client-cert
  shared_pvc: koji-hub-mnt-pvc
```

Run the following command to create a koji-builder resource:

```
kubectl apply -f kojira-cr.yaml
```

You can check its status by running:

```
kubectl get mbkojira/example -o yaml
```

1.4 mbs-backend

This page documents the usage of mb-mbs-backend crd.

1.4.1 Description

The backend of the module-build-service

1.4.2 Dependencies

Mbox Module Build Service Backend Custom Resource Definition (CRD)

1.4.3 Parameters

Name	Default Value	Type
image	quay.io/fedora/mbs-backend:latest	string
replicas	1	int
hub_username	mbs	string
cacert_secret	mbs-ca-cert	string
client_cert_secret	mbs-client-cert	string
postgres_secret	postgres	string
mbs_configmap	mbs-configmap	string
fedora_versions	[‘32’]	[string]
hub_host	‘koji-hub:8443’	string
messaging_system	‘fedmsg’	string
topic_prefix	‘org.fedoraproject.dev’	string
scm_url	‘git+https://src.fedoraproject.org/modules/’	string
rpms_default_repository	‘git+https://src.fedoraproject.org/rpms/’	string
rpms_default_cache	‘https://src.fedoraproject.org/repo/pkgs/’	string
modules_default_repository	‘git+https://src.fedoraproject.org/modules/’	string
pdc_url	‘https://pdc.stg.fedoraproject.org/rest_api/v1’	string
oidc_required_scope	‘https://mbs.fedoraproject.org/oidc/submit-build’	string
shared_pvc	koji-hub-mnt-pvc	string
mbox	“”	string

image

The the full qualified image name to pull mbs-backend from.

replicas

The amount of mbs-backend replicas to deploy.

hub_username

User to use when authenticating with koji-hub.

cacert_secret

The root CA secret name to use.

If not provided it uses the one generated (self-signed).

client_cert_secret

The client secret name to use or create.

It will skip its creation (self signed) if one is already present.

It needs to be created and signed using the root CA certificate and private key.

Secret format:

```
apiVersion: v1
kind: Secret
metadata:
  name: myservice
  namespace: default
  labels:
    app: koji-builder
type: kubernetes.io/tls
data:
  tls.crt: -|
    fillme
  tls.key: -|
    fillme
  tls.pem: -|
    This is a combination of tls.key and tls.crt separated by '\n' and encoded in␣
↪base64
    Example: "{{ (lookup('file', 'client_key.pem') + '\n' + lookup('file', 'client_
↪cert.pem')) | b64encode }}"
```

postgres_secret

Postgresql secret used by MBS to connect to a psql instance.

Deployment will fail if this secret is not present.

Secret format:

```
apiVersion: v1
kind: Secret
metadata:
  name: postgres
  labels:
    app: postgres
data:
  POSTGRES_HOST: fillme
  POSTGRES_DB: fillme
  POSTGRES_USER: fillme
  POSTGRES_PASSWORD: fillme
```

configmap

The configmap name to use when deploying configuration shared between mbs-frontend and mbs-backend component.

This configmap contains configuration files that are shared between mbs-frontend and mbs-backend.

fedora_versions

The versions of the Fedora we need to generate module template for.

messaging_system

Messaging system to use when sending messages. Support for fedora messaging is not available in MBS for now.

topic_prefix

Prefix of the topic for messaging system.

config_scm_url

Source Code Management git URL for modules, should contain repositories for modules builds definitions.

rpms_default_repository

Default repository git URL for RPMS.

rpms_default_cache

Default cache URL for RPMS.

modules_default_repository

Default repository git URL for modules.

pdc_url

Product Definition Center URL.

oidc_required_scope

OIDC required scope URL.

shared_pvc

Name of the shared PersistentVolumeClaim mbs-backend will use.

mbox

A Mbox resource name to retrieve shared data from (pvc volume, shared certs and shared MBS configmap).

MBS Backend will use the following vars if this property is missing:

- shared_pvc (shared koji mnt volume)
- cacert_secret (root ca secret)
- postgres_secret (PSQL secret)
- configmap (shared configmap name)
- fedora_versions (versions of fedora for module templates)
- hub_host (Koji host URL)
- messaging_system (messaging system to use)
- topic_prefix (topic prefix for messaging system)

- scm_url (URL for SCM)
- rpms_default_repository (default URL for RPMS repositories)
- rpms_default_cache (default cache URL)
- modules_default_repository (default URL for modules repositories)
- pdc_url (URL for PDC)
- oidc_required_scope (OIDC required scope URL)

1.4.4 Usage

Upstream file can be found here

Create a file mbmbsbackend-cr.yaml containing the following content (modify as needed):

```
apiVersion: apps.fedoraproject.org/v1alpha1
kind: MBMbsBackend
metadata:
  name: example-mb-mbs-backend
spec:
  replicas: 1
  image: quay.io/fedora/mbs-backend:latest
  hub_username: mbs
  cacert_secret: koji-hub-ca-cert
  client_cert_secret: mbs-client-cert
  postgres_secret: postgres
  configmap: mbs-configmap
  fedora_versions: ['32']
  hub_host: 'koji-hub:8443'
  messaging_system: 'fedmsg'
  topic_prefix: 'org.fedoraproject.dev'
  scm_url: 'git+https://src.fedoraproject.org/modules/'
  rpms_default_repository: 'git+https://src.fedoraproject.org/rpms/'
  rpms_default_cache: 'https://src.fedoraproject.org/repo/pkgs/'
  modules_default_repository: 'git+https://src.fedoraproject.org/modules/'
  pdc_url: 'https://pdc.stg.fedoraproject.org/rest_api/v1'
  oidc_required_scope: 'https://mbs.fedoraproject.org/oidc/submit-build'
  shared_pvc: 'koji-hub-mnt-pvc'
  # mbox: example-mbox #uncomment to retrieve pvc and cert config from a mbox cr
```

Run the following command to create a mbs-backend resource:

```
kubectl apply -f mbmbsbackend-cr.yaml
```

You can check its status by running:

```
kubectl get mbmbsbackend/example -o yaml
```

1.5 mbs-frontend

This page documents the usage of mb-mbs-frontend crd.

1.5.1 Description

The frontend of the module-build-service

1.5.2 Dependencies

Mbox Module Build Service Frontend Custom Resource Definition (CRD)

1.5.3 Parameters

Name	Default Value	Type
replicas	1	int
image	quay.io/fedora/mbs-frontend:latest	string
configmap	mbs-frontend-configmap	string
https_enabled	true	boolean
postgres_secret	postgres	string
mbs_configmap	mbs-configmap	string
fedora_versions	[‘32’]	[string]
messaging_system	‘fedmsg’	string
topic_prefix	‘org.fedoraproject.dev’	string
scm_url	‘git+https://src.fedoraproject.org/modules/’	string
rpms_default_repository	‘git+https://src.fedoraproject.org/rpms/’	string
rpms_default_cache	‘https://src.fedoraproject.org/repo/pkgs/’	string
modules_default_repository	‘git+https://src.fedoraproject.org/modules/’	string
pdc_url	‘https://pdc.stg.fedoraproject.org/rest_api/v1’	string
oidc_required_scope	‘https://mbs.fedoraproject.org/oidc/submit-build’	string
ca_cert_secret	koji-hub-ca-cert	string
koji_hub_host	‘koji-hub:8443’	string
host	‘mbs.mbox.dev’	string
client_cert_secret	mbs-frontend-client-cert	string
service_cert_secret	mbs-frontend-service-cert	string
mbox	“”	string

image

The full qualified image name to pull mbs-frontend from.

replicas

The amount of mbs-frontend replicas to deploy.

configmap

The configmap name to use when deploying mbs-frontend

This configmap object contains mbs-frontend specific configuration files that are mounted in mbs-frontend pod filesystem.

https_enabled

A boolean flag that enables/disables https connections. If set to false http will be enabled.

postgres_secret

Postgresql secret used by MBS to connect to a psql instance.

Deployment will fail if this secret is not present.

Secret format:

```
apiVersion: v1
kind: Secret
metadata:
  name: postgres
  labels:
    app: postgres
data:
  POSTGRES_HOST: fillme
  POSTGRES_DB: fillme
  POSTGRES_USER: fillme
  POSTGRES_PASSWORD: fillme
```

mbs_configmap

The configmap name to use when deploying configuration shared between mbs-frontend and mbs-backend component.

This configmap contains configuration files that are shared between mbs-frontend and mbs-backend.

fedora_versions

The versions of the Fedora we need to generate module template for.

messaging_system

Messaging system to use when sending messages. Support for fedora messaging is not available in MBS for now.

topic_prefix

Prefix of the topic for messaging system.

config_scm_url

Source Code Management git URL for modules, should contain repositories for modules builds definitions.

rpms_default_repository

Default repository git URL for RPMS.

rpms_default_cache

Default cache URL for RPMS.

modules_default_repository

Default repository git URL for modules.

pdc_url

Product Definition Center URL.

oidc_required_scope

OIDC required scope URL.

ca_cert_secret

The root CA secret name to use.

If not provided it uses the one generated (self-signed).

koji_hub_host

Koji hub service name:port. This is used as common name for client certificate.

host

Hostname for MBS server. This is used as common name for server certificate.

client_cert_secret

The client secret name to use or create.

It will skip its creation (self signed) if one is already present.

It needs to be created and signed using the root CA certificate and private key.

Secret format:

```
apiVersion: v1
kind: Secret
metadata:
  name: myservice
  namespace: default
  labels:
    app: koji-builder
type: kubernetes.io/tls
data:
  tls.crt: -|
    fillme
```

(continues on next page)

(continued from previous page)

```
  tls.key: -|
    fillme
  tls.pem: -|
    This is a combination of tls.key and tls.crt separated by '\n' and encoded in␣
↪base64
    Example: "{{ (lookup('file', 'client_key.pem') + '\n' + lookup('file', 'client_
↪cert.pem')) | b64encode }}"
```

service_cert_secret

The httpd service secret name to use or create.

It will skip its creation (self signed) if one is already present.

It needs to be created and signed using the root CA certificate and private key.

Secret format:

```
apiVersion: v1
kind: Secret
metadata:
  name: myservice
  namespace: default
  labels:
    app: koji-hub
type: kubernetes.io/tls
data:
  tls.crt: -|
    fillme
  tls.key: -|
    fillme
```

mbox

A Mbox resource name to retrieve shared data from (pvc volume, shared certs and shared MBS configmap).

MBS Frontend will use the following vars if this property is missing:

- postgres_secret (PSQL secret)
- mbs_configmap (shared configmap name)
- fedora_versions (versions of fedora for module templates)
- messaging_system (messaging system to use)
- topic_prefix (topic prefix for messaging system)
- scm_url (URL for SCM)
- rpms_default_repository (default URL for RPMS repositories)
- rpms_default_cache (default cache URL)
- modules_default_repository (default URL for modules repositories)
- pdc_url (URL for PDC)
- oidc_required_scope (OIDC required scope URL)

- koji_hub_host (Koji host URL)
- cacert_secret (root ca secret)

1.5.4 Usage

Upstream file can be found here

Create a file mbmbsfrontend-cr.yaml containing the following content (modify as needed):

```
apiVersion: apps.fedoraproject.org/v1alpha1
kind: MBMbsFrontend
metadata:
  name: mb-mbs-frontend
  labels:
    app: mb-mbs-frontend
spec:
  replicas: 1
  image: quay.io/fedora/mbs-frontend:latest
  configmap: mbs-frontend-configmap
  https_enabled: true
  postgres_secret: postgres
  mbs_configmap: mbs-configmap
  fedora_versions: ['32']
  messaging_system: 'fedmsg'
  topic_prefix: 'org.fedoraproject.dev'
  scm_url: 'git+https://src.fedoraproject.org/modules/'
  rpms_default_repository: 'git+https://src.fedoraproject.org/rpms/'
  rpms_default_cache: 'https://src.fedoraproject.org/repo/pkgs/'
  modules_default_repository: 'git+https://src.fedoraproject.org/modules/'
  pdc_url: 'https://pdc.stg.fedoraproject.org/rest_api/v1'
  oidc_required_scope: 'https://mbs.fedoraproject.org/oidc/submit-build'

  ca_cert_secret: koji-hub-ca-cert
  koji_hub_host: 'koji-hub:8443'
  host: 'mbs.mbox.dev'
  client_cert_secret: mbs-frontend-client-cert
  service_cert_secret: mbs-frontend-service-cert
  service_name: 'mbs'
  ingress_backend: 'nginx'
  # mbox: example-mbox #uncomment to retrieve pvc and cert config from a mbox cr
```

Run the following command to create a mbs-frontend resource:

```
kubectl apply -f mbmbsfrontend-cr.yaml
```

You can check its status by running:

```
kubectl get mbmbsfrontend/example -o yaml
```

1.6 mbox

This page documents the usage of mbox crd.

1.6.1 Description

Mbox is a configuration resource that can be optionally used to defined shared configuration across other components.

1.6.2 Dependencies

Mbox Custom Resource Definition (CRD)

1.6.3 Parameters

Name	Default Value	Type
psql_secret_name	postgres	string
koji_pvc_name	mbox-koji-mnt	string
koji_pvc_size	10Gi	string
root_ca_secret_name	mbox-koji-root-ca	string
koji_hub_host	koji-hub:8443	string
mq_topic_prefix	org.fedoraproject.dev	string
mbs	{}	dict
mbs.scm_repo_url	git+https://src.fedoraproject.org/modules/	string
mbs.rpm_repo_url	git+https://src.fedoraproject.org/rpms/	string
mbs.pkg_repo_url	https://src.fedoraproject.org/repo/pkgs/	string
mbs.pdc_url	https://pdc.stg.fedoraproject.org/rest_api/v1	string
mbs.backend_config_messaging	fedmsg	string
mbs.fedora_versions	[‘32’]	[string]
mbs.topic_prefix	org.fedoraproject.dev	string
mbs.configmap	mbs-configmap	string

psql_secret_name

Postgresql secret used across many components to connect to a psql instance.

Deployment will fail if this secret is not present.

Secret format:

```
apiVersion: v1
kind: Secret
metadata:
  name: postgres
  labels:/
    app: postgres
data:
  POSTGRES_HOST: fillme
  POSTGRES_DB: fillme
  POSTGRES_USER: fillme
  POSTGRES_PASSWORD: fillme
```

koji_pvc_name

The koji pvc name to be used as shared volume across components.

It will not create a PVC if one with the same name is already present.

koji_pvc_size

The koji pvc size to be used as shared volume across components.

This value will be ignore if using an existing volume instead of creating one.

root_ca_secret_name

Root CA secret used to generate certificates across many components (koji clients, httpd, etc).

It will create a secret using self signed certs in case it does not exist.

```
apiVersion: v1
kind: Secret
  metadata:
    name: mysecret
    namespace: default
    labels:
      app: mbox
  data:
    csr: -|
      fillme
    cert: -|
      fillme
    key: -|
      fillme
```

koji_hub_host

The koji-hub internal service address (service name) and port to be used across koji and mbs components.

mq_topic_prefix

The MQ topic prefix to use when listening/emitting messages.

mbs

Shared config dictionary for both mbs frontend and backend.

This property is optional.

scm_repo_url

MBS scm repository git url to use.

This property is optional.

rpm_repo_url

MBS RPM repository git url.

This property is optional.

pkg_repo_url

MBS package repository url.

This property is optional.

pdc_url

MBS PDC rest API url.

This property is optional.

backend_config_messaging

Sets the mbs messaging system to use. We assume fedmsg as the default.

fedora_versions

The versions of the Fedora we need to generate module template for.

oidc_required_scope

MBS OIDC required scope URL.

topic_prefix

The MBS MQ topic prefix to use when listening/emitting messages.

configmap

The MBS config map name to use when creating one.

It will skip its creation and an existing one if it already exists.

1.6.4 Usage

Upstream file can be found here

Create a file containing the following content (modify as needed):

```
apiVersion: apps.fedoraproject.org/v1alpha1
kind: Mbox
metadata:
  name: example
spec:
  psql_secret_name: postgres
  koji_pvc_name: mbox-koji-mnt
  koji_pvc_size: 10Gi
  root_ca_secret_name: mbox-koji-root-ca
```

(continues on next page)

(continued from previous page)

```
  koji_hub_host: koji-hub:8443
  mq_topic_prefix: 'org.fedoraproject.dev'
  mbs:
    fedora_versions:
    - '32'
    scm_repo_url: 'git+https://src.fedoraproject.org/modules/'
    rpm_repo_url: 'git+https://src.fedoraproject.org/rpms/'
    pkg_repo_url: 'https://src.fedoraproject.org/repo/pkgs/'
    pdc_url: 'https://pdc.stg.fedoraproject.org/rest_api/v1'
    oidc_required_scope: 'https://mbs.fedoraproject.org/oidc/submit-build'
    config_system: koji
    backend_config_messaging: fedmsg
    hub_username: mbs
```

Run the following command to create a koji-builder resource:

```
kubectl apply -f mbox-cr.yaml
```

You can check its status by running:

```
kubectl get mbox/example -o yaml
```

CHAPTER 2

Deployment Guide

This guide describes manual deployment process of MBBox operator in OpenShift 4 and Kubernetes Cluster.

2.1 Requirements

To be able to deploy MBBox operator manually you need to have admin rights to cluster. Otherwise you will not be able to deploy CRDs, roles, role bindings or service accounts.

To be able to deploy just the CR files you only need admin rights to namespace where you want to deploy operator.

Some commands require the usage of the "kustomize" CLI tool which an be found here.

2.2 Makefile

We have a Makefile for you ready to be used.

2.3 Prepare MBBox deployment

To prepare cluster for MBBox deployment you just need to run following commands with the Makefile mentioned earlier in this guide.

```
make install
make deploy
```

This will apply CRDs files for MBBox and create roles, role bindings, service accounts and deploy the mbbox operator.

2.4 Create PVCs

For deployment of the MBBox operator you need to prepare 2 PVCs. In case you are allowed to create PVCs this could be configured in CR files for koji-hub and mbox itself and they will be generated automatically based on the configuration. Otherwise you need to create them manually beforehand.

Most of the components are using Koji shared mount point. Check if the name of PVC is same in each component CR file otherwise the deployment will fail.

2.5 Prepare PostgreSQL DB

In case you have PostgreSQL DB running in cluster you can skip this step and just use the existing one.

To deploy PostgreSQL DB you can use the one prepared by us. You can change anything in those files, especially secret file. To deploy it run the following.

```
kubectl apply -f components/psql -n <namespace>
```

2.6 Prepare RabbitMQ server

In case you have a running RabbitMQ server in your cluster, you can skip this step and just use the existing one.

To deploy RabbitMQ you can use the one prepared by us. You can change anything in those files, especially secret file. Refer to the *README.md* file for instructions about certificates. To deploy it run the following.

```
kubectl apply -f components/rabbitmq -n <namespace>
```

Note: Right now only Koji is emitting Fedora messaging messages, which needs the RabbitMQ server.

2.7 CR Deployment

Before deploying CR check the variables configuration. Please refer to *Contents:* for information about variables.

A full deployment needs to deploy a couple of CRs in order, *kustomize* can be used to achieve that:

```
kustomize build config/samples | kubectl apply -f -
```

2.8 Delete Operator deployment

To delete operator deployment simply run:

```
kustomize build config/samples | kubectl delete -f -
make undeploy # This will delete the operator
make uninstall # this will uninstall CRDs, roles, etc
```

CHAPTER 3

Contributing

Mbox welcomes contributions! Our issue tracker is located on GitHub.

3.1 Guidelines

When you make a pull request, someone from the fedora organization will review your code. Please make sure you follow the guidelines below:

3.1.1 Code Style

Make sure your yaml code passes our yamllint rules.

3.1.2 E2E Tests

Every change should be tested in molecule, which is the tool we use for E2E (end to end) testing.

Tests can be run using either molecule or operator-sdk cli (which uses molecule as well).

```
molecule test -s test-local #local tests, no need for a cluster
molecule test -s test-cluster #needs a remote cluster, minikube is enough
```

3.1.3 Debugging Local Tests

If you encounter any error when running test you can debug the issue by connecting to local instance of kubernetes running in docker:

```
molecule converge -s test-local #runs local test without destroy sequence
docker ps #find container named kind-test-local
docker exec -it <container_id> bash #<container_id> of container from previous command
kubectl config set-context --current --namespace=osdk-test #sets namespace to␣
↪operator-sdk
```

Here are few useful commands for debugging, for another commands look at *kubectl help*:

```
kubectl get all #returns all resources in the current namespace
kubectl logs <pod> #shows logs for specific <pod>
kubectl logs <mbox-operator-pod> ansible #shows ansible logs for <mbox-operator-pod>
kubectl describe <resource> #shows detailed information about specific <resource>
kubectl get ingress #returns all ingress resources, is not part of get all
```

3.1.4 Troubleshooting

During the development, we encountered some issues when debugging operator deployment. We will try to document them in this section, together with solutions.

Issue: Timeout in reconciliation task

This was caused by low space, because failing tests aren't removing docker volumes when they fails. To remove the volumes run following command *docker volume prune*.

3.2 Environment

We are providing a full development environment in Vagrant but you can use your host machine as long as you meet the following requirements:

- ansible >= 2.9
- molecule >= 3
- yamllint >= 1.20
- python kubernetes and openshift libraries
- operator-sdk >= 0.16
- docker >= 19

NOTE: make sure both ansible and molecule are system-wide installed using in the same python interpreter otherwise you may have issues running tests.

3.2.1 Setting Up Vagrant Environment

To start the vagrant operator SDK box, run the following in project root:

```
vagrant up #starts the vagrant VM, it could take a while
vagrant reload #this is needed to remount the sshfs mounts after reboot when cgroups␣
↪are changed to V1
vagrant ssh #connects you to the vagrant VM
```

In vagrant VM you can find project folder in *~/devel*. To run the tests do *cd ~/devel/mbox-operator* and follow *E2E Tests* section.

If you encounter any issue with *vagrant up* command, do *vagrant destroy* to be sure that there isn't any leftover from previous run.

Module Building in a Box (MBBOX) Kubernetes Operator

MBOX is a kubernetes operator used to set up a buildsystem that can be used for Fedora/RHEL Modular packages, based on Koji and Module Build Service.

The intention is for it to be trivially simple to get started with both, but also allow the same setup to be used for a production setup.