The Cloud Integration Offer

Realization document

Bachelor’s in applied IT
Application Development

Emile De Vlieger

Academic year 2023-2024

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

Table of Contents

1 INTRODUCTION 2

2 TUTORIALS 3

2.1 Ace APP on RedHat OpenShift 3

2.1.1 Prerequisites 3

2.1.2 Creating a namespace 3

2.1.3 Creating the configurations 4

2.1.4 Creating our runtime operator 8

2.1.5 Creating a route........ 10

2.1.6 Accessing our web UI 12

2.2 Deploy App Connect application with ArgoCD 14

2.2.1 Prerequisites 14

2.2.2 steps:........ 14

2.2.3 Links for GitLab repo’s 14

2.2.4 Creating the namespace 15

2.2.5 Creating the ArgoCD instance 15

2.2.6 Accessing our platform 16

2.2.7 Creating an application 19

2.2.8 Testing 21

2.3 How to create an Azure OpenShift Cluster 25

2.3.1 Prerequisite........ 25

2.3.2 Setting up the cluster 25

2.3.3 Introducing extra IBM operators 28

2.4 Powerpoint and Pitch 31

2.4.1 Powerpoint........ 31

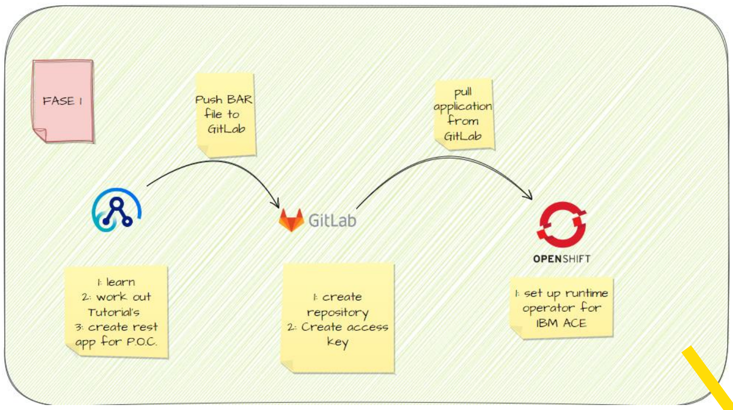
3 CONCLUSION 34

1 INTRODUCTION

In this document, I have outlined my achievements. As mentioned in my "Plan of Approach," my team uses Confluence for sharing all documentation and accomplishments. Confluence serves as a platform for sharing various documents and ideas among team members.

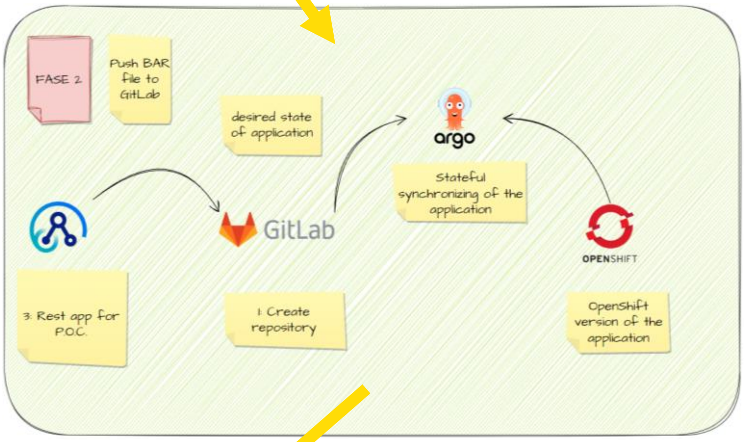
Given the relatively new nature of my role, I have created several tutorials to share with my colleagues, which I use as evidence of my achievements. In addition to this, my responsibilities included developing an offer, for which I created a PowerPoint presentation. I am attaching the slides along with a video recording of my pitch.

To give more context on how the tutorials are ordered:



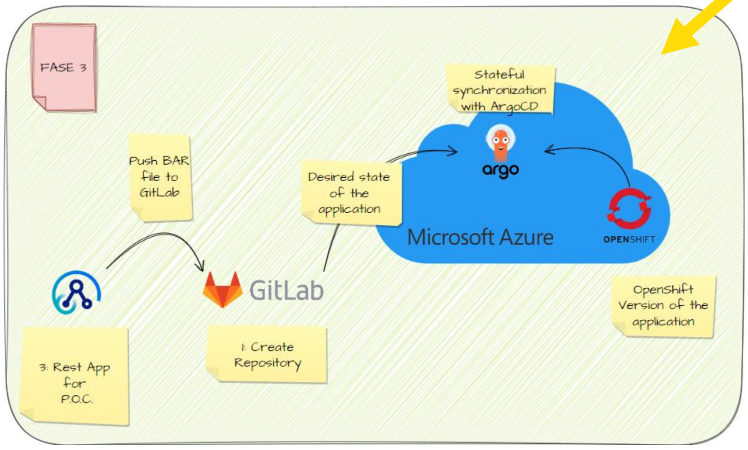
FASE 1
Push BAR file to GitLab
pull application from GitLab
GitLab
OPENSHIFT
1: learn 2: work out Tutorials 3: create rest app for P.O.C.
1: create repository 2: Create access key
1: set up runtime operator for IBM ACE

The first tutorial is the first picture. I push my application file (BAR file) to my repo on git lab and I pull that file with OpenShift to deploy the application.



FASE 2
Push BAR file to GitLab
desired state of application
argo
Stateful synchronizing of the application
GitLab
OPENSHIFT
3: Rest app for P.O.C.
1: Create repository
OpenShift version of the application

Second tutorial, I push my BAR file to git lab and ArgoCD pulls it to deploy it on OpenShift. Difference here is that with ArgoCD it remains synced with the GitLab repo to make sure the desired state of the application is always running.



The third tutorial is situated in the cloud. I give a step-by-step guide to deploy an azure OpenShift cluster and how to deploy your ACE application. The deployment process is very similar to the previous tutorial, but it covers how to properly set up your cloud environment and how to handle the few differences with the on-prem deployment process.

2 TUTORIALS

In this section, I will share all the tutorials that I have created on our sharing platform. These tutorials include step-by-step guides, with pictures and brief explanations of the processes that are happening behind the scenes.

2.1 Ace APP on RedHat OpenShift

In this tutorial, I'll be explaining the simplest method for deploying an App Connect Enterprise application using OpenShift. OpenShift is a deployment platform that is based on Kubernetes, but with an additional layer on top that makes everything easier.

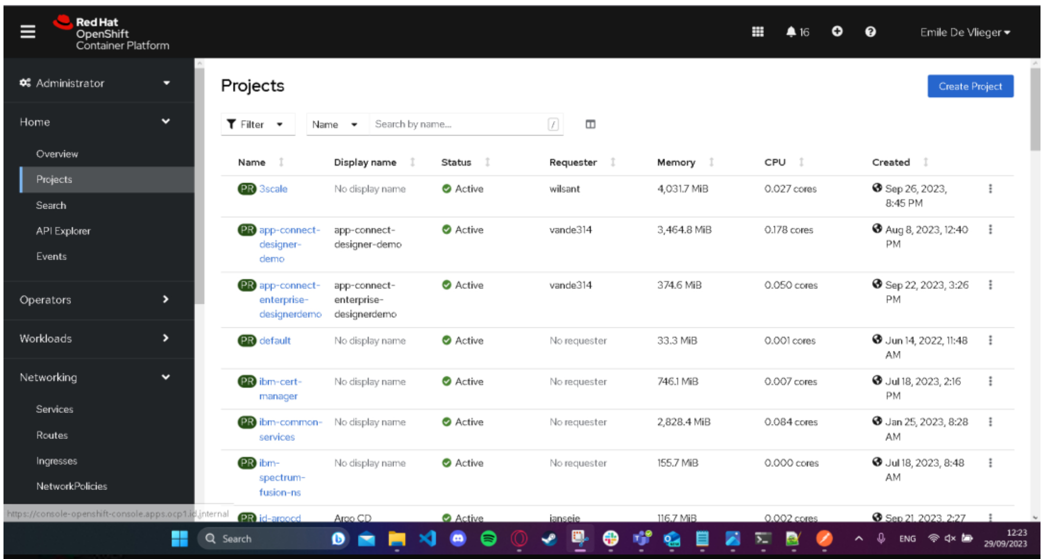
2.1.1 Prerequisites

- You need a git lab repo with a BAR file to deploy.

2.1.2 Creating a namespace

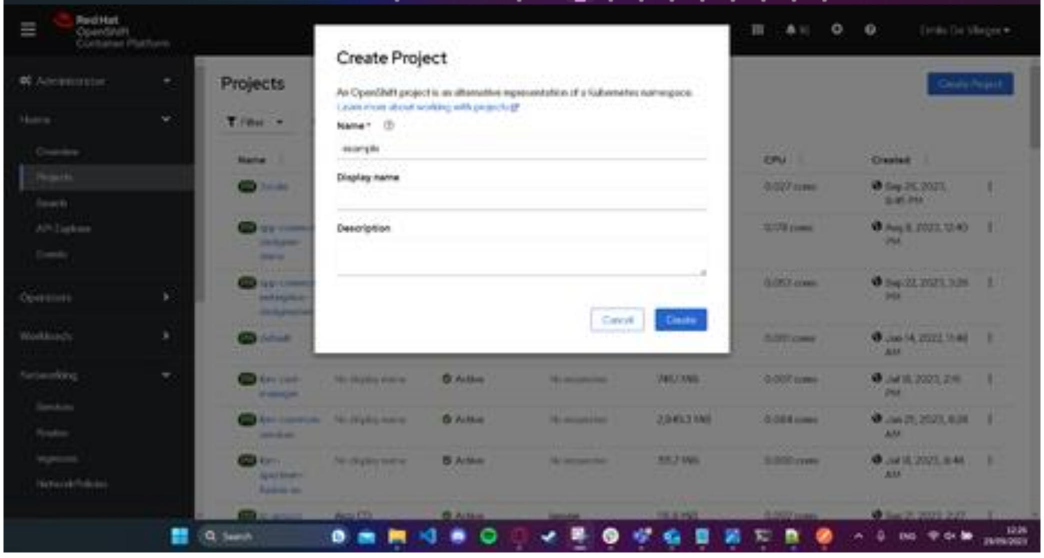
First up I made a namespace for myself which you can do by clicking home on ‘Home’ (top-left) => ‘Projects’ => ‘Create Project’ (top-right).

Then in the pop-up enter a name.



Projects

Name	Display name	Status	Requester	Memory	CPU	Created
3scale	No display name	Active	wilsant	4,031.7 MiB	0.027 cores	Sep 26, 2023, 8:45 PM
app-connect-designer-demo	app-connect-designer-demo	Active	vande314	3,464.8 MiB	0.178 cores	Aug 8, 2023, 12:40 PM
app-connect-enterprise-designerdemo	app-connect-enterprise-designerdemo	Active	vande314	374.6 MiB	0.050 cores	Sep 22, 2023, 3:26 PM
default	No display name	Active	No requester	33.3 MiB	0.001 cores	Jun 14, 2022, 11:48 AM
ibm-cert-manager	No display name	Active	No requester	746.1 MiB	0.007 cores	Jul 18, 2023, 2:16 PM
ibm-common-services	No display name	Active	No requester	2,828.4 MiB	0.084 cores	Jan 25, 2023, 8:28 AM
ibm-spectrum-fusion-ns	No display name	Active	No requester	155.7 MiB	0.000 cores	Jul 18, 2023, 8:48 AM
id-argocd	Argo CD	Active	janseje	116.7 MiB	0.002 cores	Sep 21, 2023, 2:27



Create Project

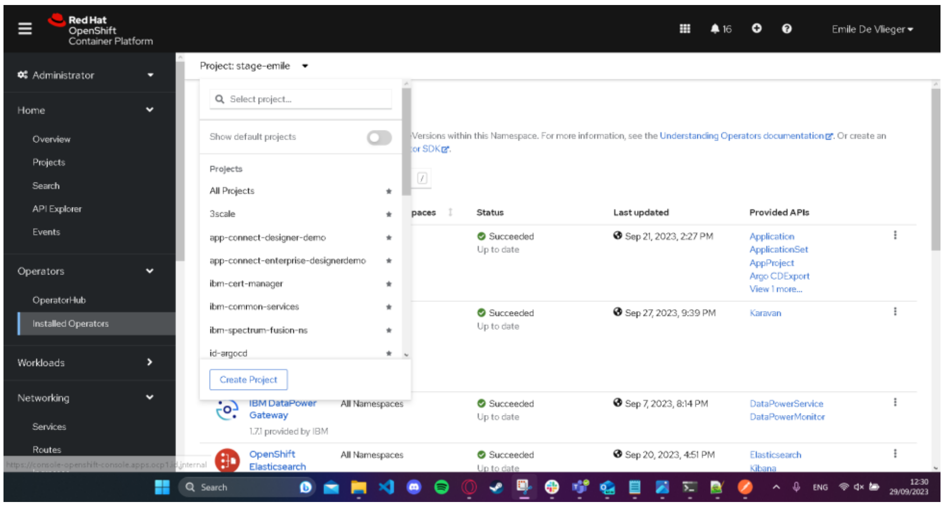
An OpenShift project is an alternative representation of a Kubernetes namespace.
Learn more about working with projects

Name * example

Display name

Description

Once you made your namespace, you will find it in the 'Projects' list. To work within this namespace you can either click it in the list or, when on other tabs, you can find it in the dropdown list:



2.1.3 Creating the configurations

Next up we will create 2 configurations:

- Barauth
    - We need this so OpenShift knows it can access our BAR-file on git.
- ServerConf
    - We need this for OpenShift to create a web-UI to access our application.

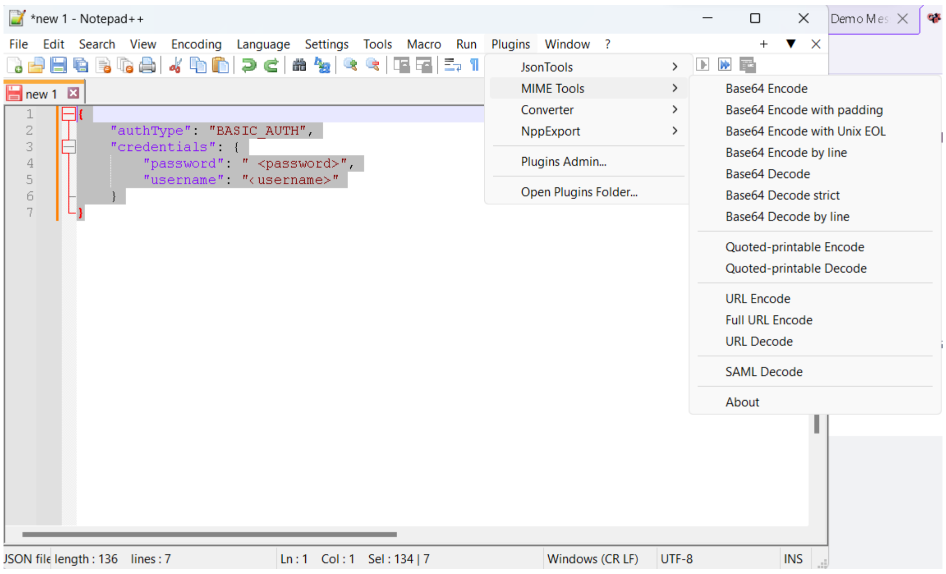
2.1.3.1 Creating the base65 encoded string to configure our "barauth"

The "barauth" configuration is where we store our credentials to be able to reach our BAR-file which we stored in the GitLab repository.

Open 'NotePad++' and copy this json text:

```
{
    "authType": "BASIC_AUTH",
    "credentials": {
        "password": " <password>",
        "username": "<username>"
    }
}
```

Now in 'notepadd++' select this piece of code and, in your toolbar, hover over 'plugins' => hover over 'MIMETOOLS' and select 'base64 encode'

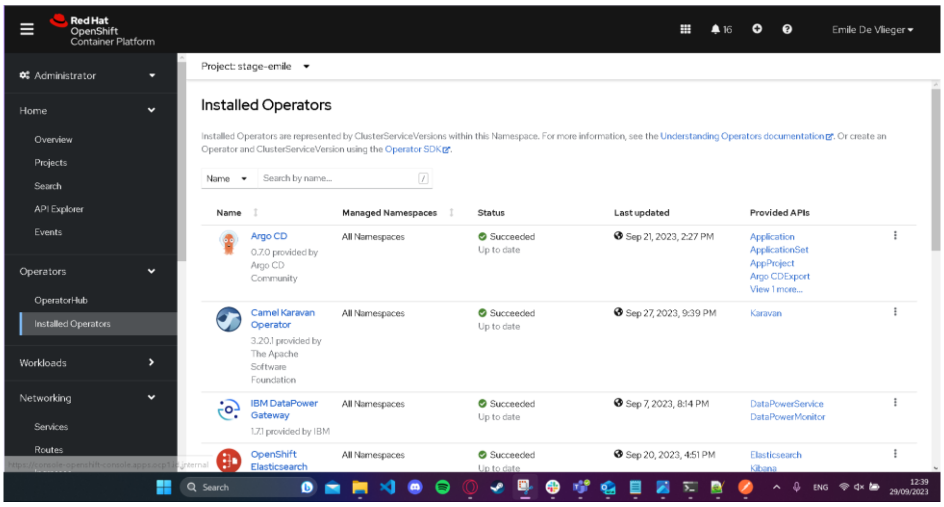


```
{
  "authType": "BASIC_AUTH",
  "credentials": {
      "password": " <password>",
      "username": "<username>"
    }
}
```

After this you should have base64 encoded json instead of the json we wrote.

2.1.3.2 Making our 'barauth' config

In OpenShift, select 'installed operators' under the 'Operators' tab:

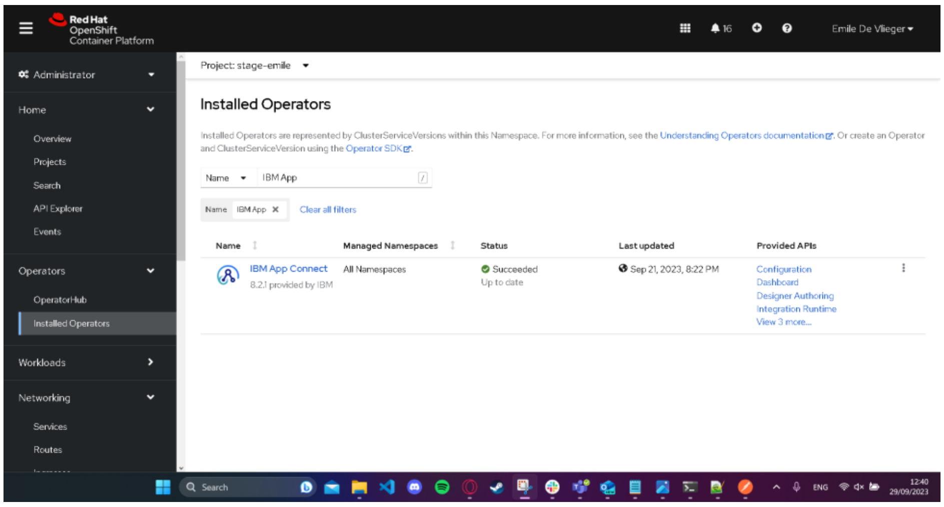


Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the Understanding Operators documentation. Or create an Operator and ClusterServiceVersion using the Operator SDK.

Name	Managed Namespaces	Status	Last updated	Provided APIs
Argo CD 0.7.0 provided by Argo CD Community	All Namespaces	Succeeded Up to date	Sep 21, 2023, 2:27 PM	Application ApplicationSet AppProject Argo CDExport View 1 more...
Camel Karavan Operator 3.20.1 provided by The Apache Software Foundation	All Namespaces	Succeeded Up to date	Sep 27, 2023, 9:39 PM	Karavan
IBM DataPower Gateway 1.7.1 provided by IBM	All Namespaces	Succeeded Up to date	Sep 7, 2023, 8:14 PM	DataPowerService DataPowerMonitor
OpenShift Elasticsearch	All Namespaces	Succeeded Up to date	Sep 20, 2023, 4:51 PM	Elasticsearch Kibana

In the search bar look for 'IBM App Connect'



Project: stage-emile

Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the Understanding Operators documentation. Or create an Operator and ClusterServiceVersion using the Operator SDK.

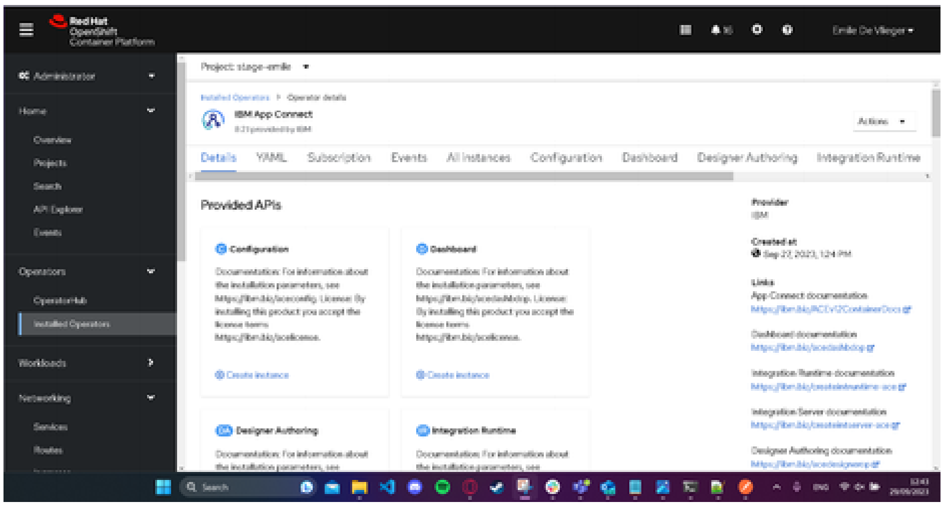
Name: IBM App

Clear all filters

Name	Managed Namespaces	Status	Last updated	Provided APIs
IBM App Connect 8.2.1 provided by IBM	All Namespaces	Succeeded Up to date	Sep 21, 2023, 8:22 PM	Configuration Dashboard Designer Authoring Integration Runtime View 3 more...

When in the App connect operator dashboard, we can see a range of possible operators to create. Later, we will create a runtime operator which we connect with GitLab to get our Bar URL from, which is why we are creating this 'barauth' configuration so that our operator can access this file.

Now click 'create instance' in the 'configuration' option:



Project: stage-emile

Installed Operators > Operator details

IBM App Connect
8.2.1 provided by IBM

Provided APIs

Configuration
Documentation: For information about the installation parameters, see https://ibm.biz/aceconfig. License: By installing this product you accept the license terms https://ibm.biz/acelicense.
Create instance

Dashboard
Documentation: For information about the installation parameters, see https://ibm.biz/acedashbdcp. License: By installing this product you accept the license terms https://ibm.biz/acelicense.
Create instance

Designer Authoring
Documentation: For information about the installation parameters, see

Integration Runtime
Documentation: For information about the installation parameters, see

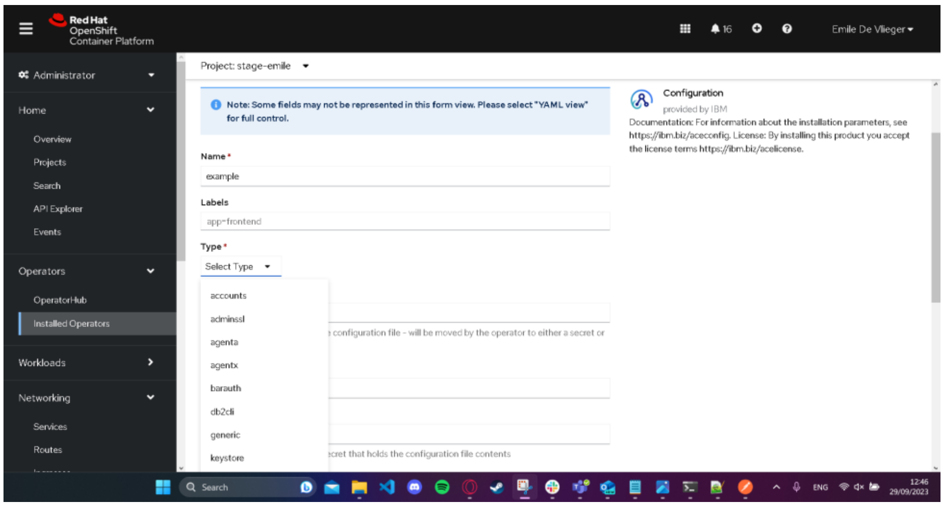
Provider
IBM

Created at
Sep 27, 2023, 1:24 PM

Links
App Connect documentation
https://ibm.biz/ACEV12ContainerDocs
Dashboard documentation
https://ibm.biz/acedashbdcp
Integration Runtime documentation
https://ibm.biz/createintruntime-ace
Integration Server documentation
https://ibm.biz/createintserver-ace
Designer Authoring documentation
https://ibm.biz/acedesignercp

Choose a name for the configuration.

Now on the 'type' dropdown, choose 'barauth':


Project: stage-emile

Note: Some fields may not be represented in this form view. Please select "YAML view" for full control.

Name *
example

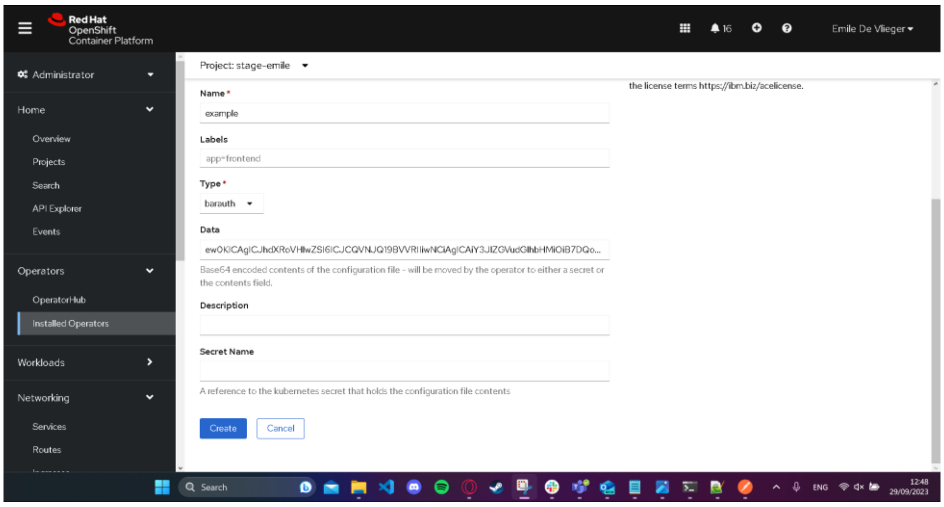
Labels
app=frontend

Type *
Select Type

accounts
adminssl
agenta
agentx
barauth
db2cli
generic
keystore

Configuration
provided by IBM
Documentation: For information about the installation parameters, see https://ibm.biz/aceconfig. License: By installing this product you accept the license terms https://ibm.biz/acelicense.

Next up, enter the previously created base 64 json in the 'DATA' field:


Project: stage-emile

Name *
example

Labels
app=frontend

Type *
barauth

Data
ewOKICAgICJhdXRoVHlwZSI6ICJCQVNJQ19BVVRIIiwNCiAgICAiY3JlZGVudGlhbHMiOiB7DQo...

Base64 encoded contents of the configuration file - will be moved by the operator to either a secret or the contents field.

Description

Secret Name

A reference to the kubernetes secret that holds the configuration file contents

Create Cancel

Click 'create'.

2.1.3.3 "Serverconf" Configuration:

Next up we need to create a configuration which will allow us to access the 'web ui' from our 'runtime operator'.

Again, choose the "App Connect" operator in the "installed operators" tab and click on configuration.

choose a name for the configuration.

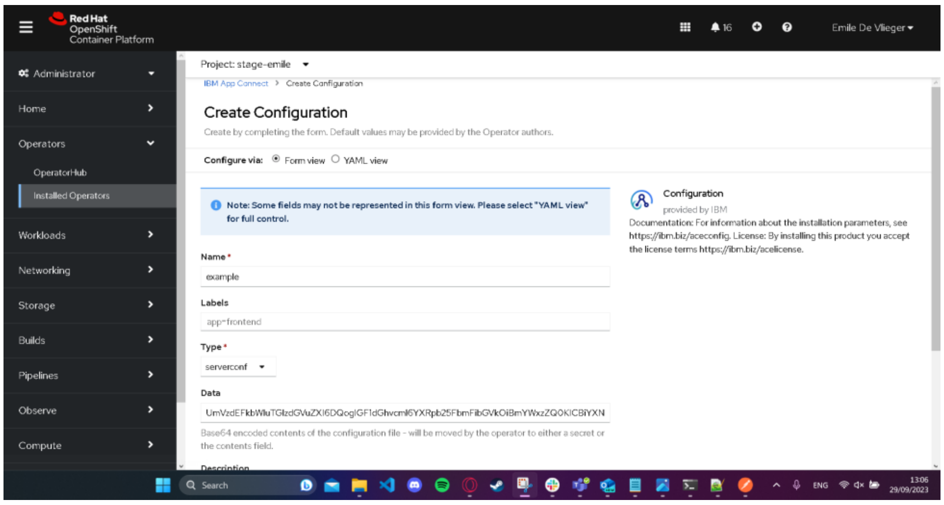
As type we will choose 'serverconf'.

Within our 'DATA' field we will enter a base64 string, derived from this configuration:

```
RestAdminListener:
  authorizationEnabled: false
  basicAuth: false
  port: 7600
  requireClientCert: false
```

You can copy it and convert it again in 'notepad++' to save time I will add the string here.

UmVzdEFkbWluTGlzdGVuZXI6DQogIGF1dGhvcml6YXRpb25FbmFibGVkOiBmYWxzZQ0KICBiYXNpY0F1dGg6IGZhbHNlDQogIHBvcnQ6IDc2MDANCiAgcmVxdWlyZUNsaWVudENlcnQ6IGZhbHNl


Red Hat OpenShift Container Platform

Project: stage-emile

IBM App Connect > Create Configuration

Create Configuration

Create by completing the form. Default values may be provided by the Operator authors.

Configure via: [✓ Form view] YAML view

Note: Some fields may not be represented in this form view. Please select "YAML view" for full control.

Name *: example

Labels: app=frontend

Type *: serverconf

Data: UmVzdEFkbWluTGlzdGVuZXI6DQogIGF1dGhvcml6YXRpb25FbmFibGVkOiBmYWxzZQ0KICBiYXN

Base64 encoded contents of the configuration file - will be moved by the operator to either a secret or the contents field.

Description

Configuration
provided by IBM
Documentation: For information about the installation parameters, see https://ibm.biz/aceconfig. License: By installing this product you accept the license terms https://ibm.biz/acelicense.

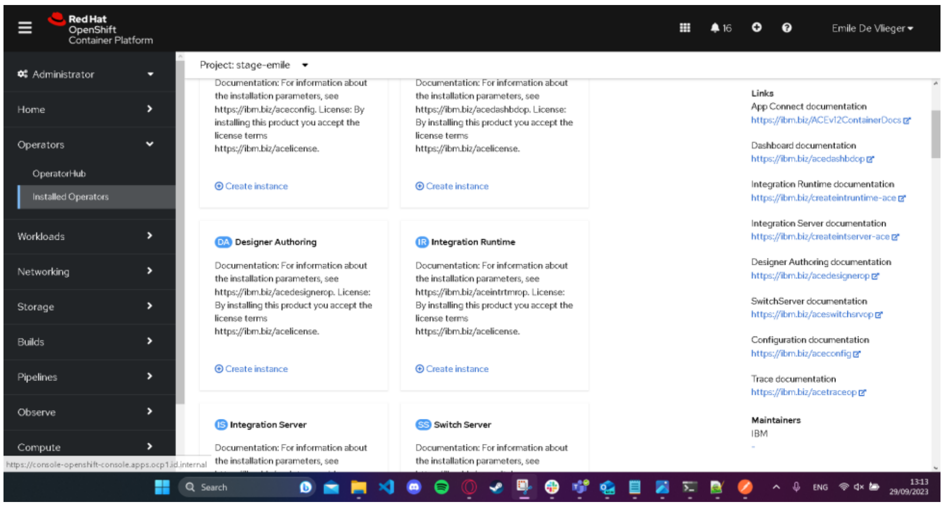
Click 'create'.

PS: To make the next step easier for you, I recommend copying the names of your configurations and keeping them on standby in notepad.

2.1.4 Creating our runtime operator

Navigate to the 'App Connect' operator dashboard.

This time we will choose to create an instance from the 'integration runtime' operator:


Red Hat OpenShift Container Platform

Project: stage-emile

Documentation: For information about the installation parameters, see https://ibm.biz/aceconfig. License: By installing this product you accept the license terms https://ibm.biz/acelicense.

Create instance

Documentation: For information about the installation parameters, see https://ibm.biz/acedashbdcp. License: By installing this product you accept the license terms https://ibm.biz/acelicense.

Create instance

DA Designer Authoring

Documentation: For information about the installation parameters, see https://ibm.biz/acedesignerop. License: By installing this product you accept the license terms https://ibm.biz/acelicense.

Create instance

IR Integration Runtime

Documentation: For information about the installation parameters, see https://ibm.biz/aceintrtmrop. License: By installing this product you accept the license terms https://ibm.biz/acelicense.

Create instance

IS Integration Server

Documentation: For information about the installation parameters, see

SS Switch Server

Documentation: For information about the installation parameters, see

Links
App Connect documentation
https://ibm.biz/ACEv12ContainerDocs
Dashboard documentation
https://ibm.biz/acedashbdcp
Integration Runtime documentation
https://ibm.biz/createintruntime-ace
Integration Server documentation
https://ibm.biz/createintserver-ace
Designer Authoring documentation
https://ibm.biz/acedesignerop
SwitchServer documentation
https://ibm.biz/aceswitchsrvop
Configuration documentation
https://ibm.biz/aceconfig
Trace documentation
https://ibm.biz/acetraceop

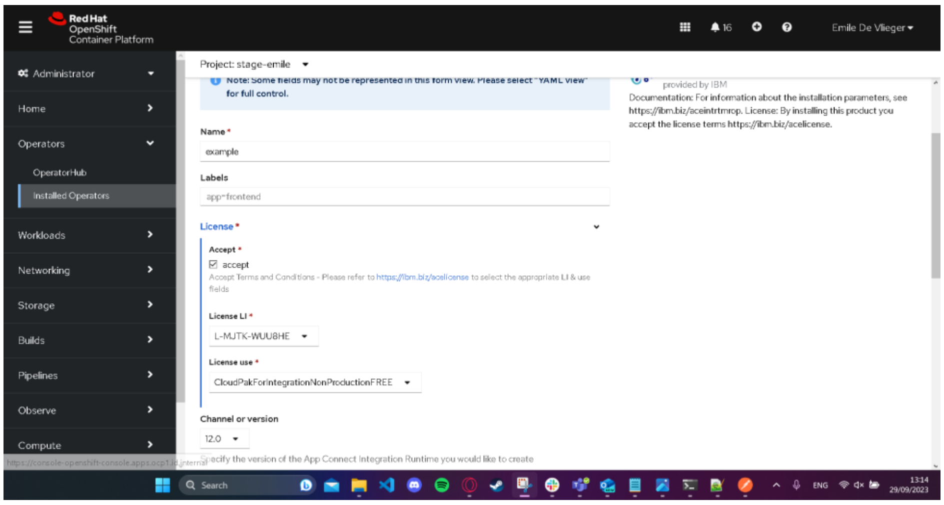
Maintainers
IBM

https://console-openshift-console.apps.ocp1.id.internal

When in the create tab:

Choose a name for your operator.

Check the 'accept' box in the 'license' dropdown:


Project: stage-emile

Note: Some fields may not be represented in this form view. Please select "YAML view" for full control.

Name *
example

Labels
app=frontend

License *

Accept *
☑ accept
Accept Terms and Conditions - Please refer to https://ibm.biz/acelicense to select the appropriate LI & use fields

License LI *
L-MJTK-WUU8HE

License use *
CloudPakForIntegrationNonProductionFREE

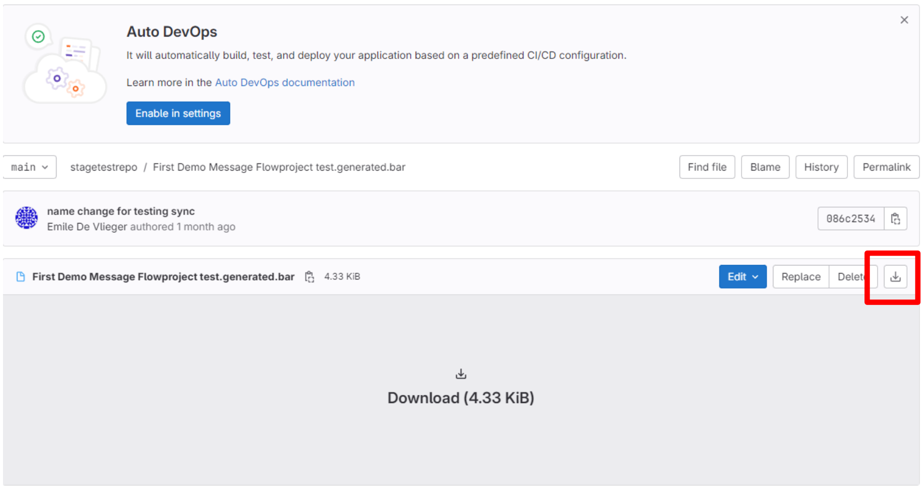
Channel or version
12.0
Specify the version of the App Connect Integration Runtime you would like to create

provided by IBM
Documentation: For information about the installation parameters, see https://ibm.biz/aceintrtmrop. License: By installing this product you accept the license terms https://ibm.biz/acelicense.

Open the Bar URL dropdown and click ‘add BAR URL’:

In your GitLab repo go to you BAR-file and copy the downloadable link:

(you can do this by right clicking the download logo on the right side of the page and choosing “copy link address”)


Auto DevOps

It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.

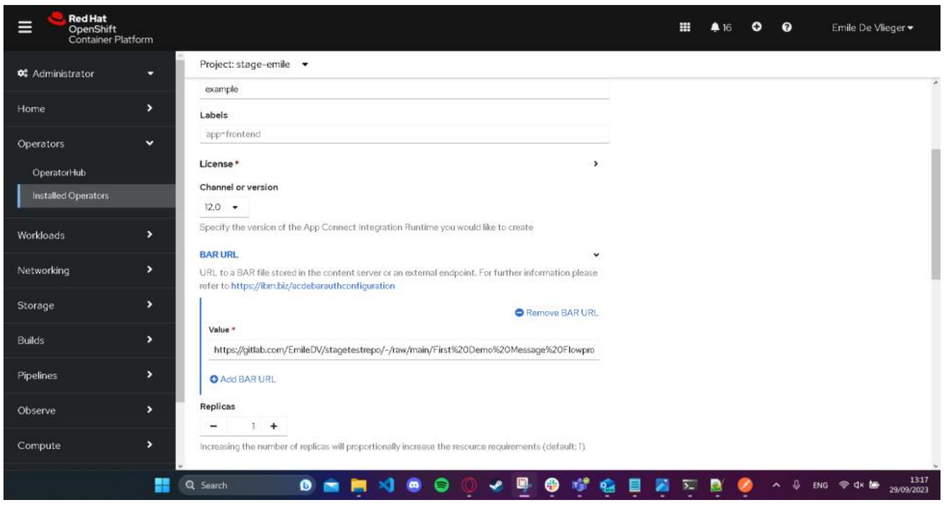
Learn more in the Auto DevOps documentation

main / stagetestrepo / First Demo Message Flowproject test.generated.bar

name change for testing sync
Emile De Vlieger authored 1 month ago
086c2534

First Demo Message Flowproject test.generated.bar 4.33 KiB

Download (4.33 KiB)


Project: stage-emile

example

Labels
app=frontend

License *

Channel or version
12.0
Specify the version of the App Connect Integration Runtime you would like to create

BAR URL
URL to a BAR file stored in the content server or an external endpoint. For further information please refer to https://ibm.biz/acdebarauthconfiguration

Remove BAR URL

Value *
https://gitlab.com/EmileDV/stagetestrepo/-/raw/main/First%20Demo%20Message%20Flowpro

Add BAR URL

Replicas
1
Increasing the number of replicas will proportionally increase the resource requirements (default: 1)

Open the ‘configuration’ dropdown:

Click ‘add configuration’ and insert the name of our ‘barauth’ configuration, which you stored in notepad 😉

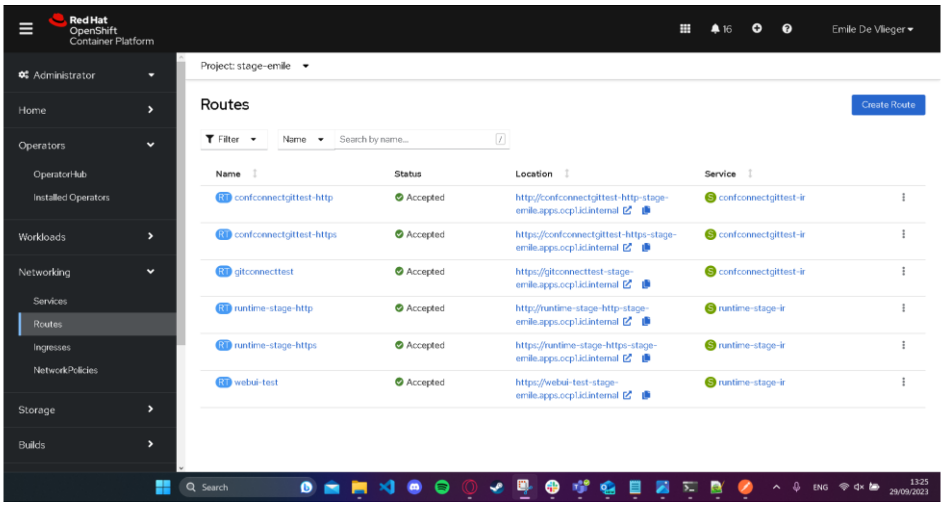
Repeat the previous step for the ‘serverconf’ configuration.

Click ‘create’

2.1.5 Creating a route

To make our lives easier we will now create a route which allows us to access the web UI just by clicking this link instead of having to ‘port forward’ our operator to our localhost.

In our menu on the left, choose Networking and click Routes:



Routes

Name	Status	Location	Service
RT confconnectgittest-http	Accepted	http://confconnectgittest-http-stage-emile.apps.ocp1.id.internal	S confconnectgittest-ir
RT confconnectgittest-https	Accepted	https://confconnectgittest-https-stage-emile.apps.ocp1.id.internal	S confconnectgittest-ir
RT gitconnecttest	Accepted	https://gitconnecttest-stage-emile.apps.ocp1.id.internal	S confconnectgittest-ir
RT runtime-stage-http	Accepted	http://runtime-stage-http-stage-emile.apps.ocp1.id.internal	S runtime-stage-ir
RT runtime-stage-https	Accepted	https://runtime-stage-https-stage-emile.apps.ocp1.id.internal	S runtime-stage-ir
RT webui-test	Accepted	https://webui-test-stage-emile.apps.ocp1.id.internal	S runtime-stage-ir

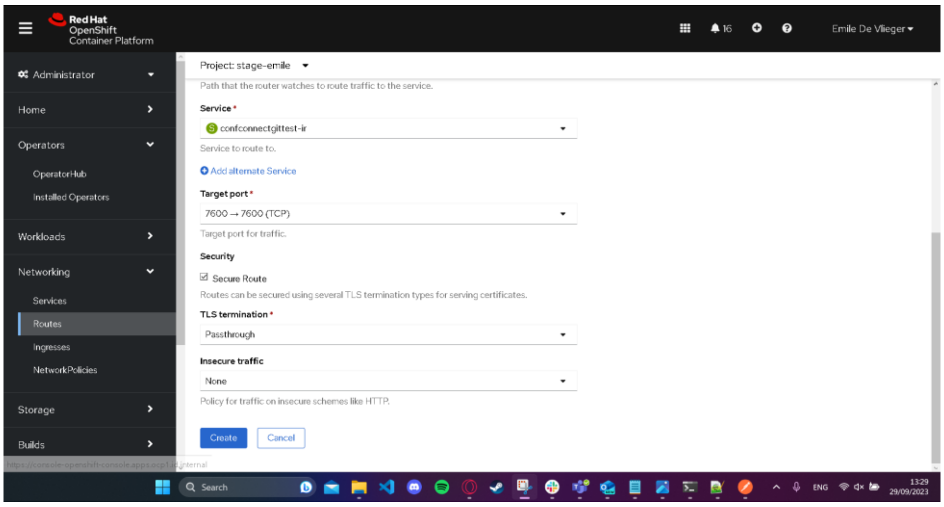
Next click ‘Create Route’

Enter a name for your route, click the ‘service’ dropdown list and choose your integration runtime.

(The name of your operator will be the name you gave it with ‘-ir’ behind it. Be careful not to choose the one with ‘-metrics’ behind it)

now under 'target port select port 7600.

Check the ‘secure’ checkbox and choose: ‘passthrough’ and ‘none’ withing theses dropdowns.



Project: stage-emile

Path that the router watches to route traffic to the service.

Service *

confconnectgittest-ir

Service to route to.

Add alternate Service

Target port *

7600 → 7600 (TCP)

Target port for traffic.

Security

☑ Secure Route

Routes can be secured using several TLS termination types for serving certificates.

TLS termination *

Passthrough

Insecure traffic

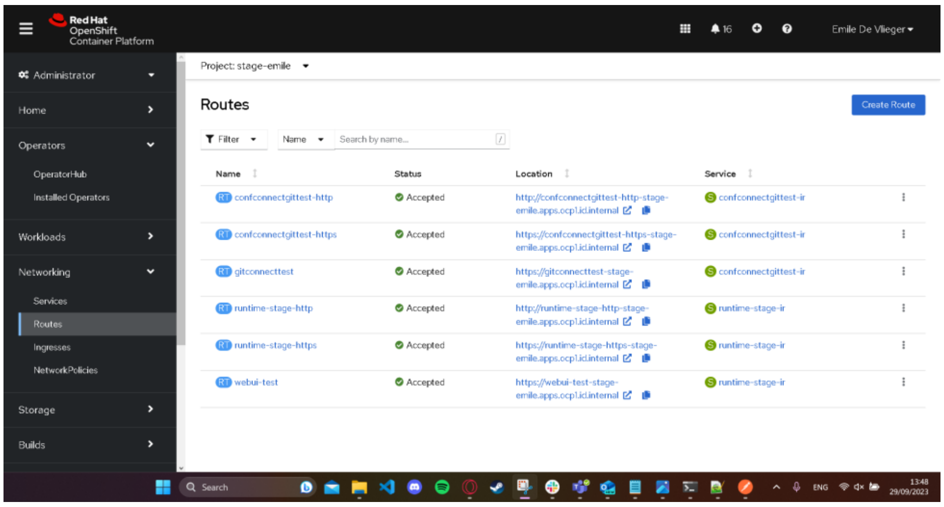
None

Policy for traffic on insecure schemes like HTTP.

Create Cancel

Click 'create'

You will see the link in the new rout we've just created.



Project: stage-emile

Routes

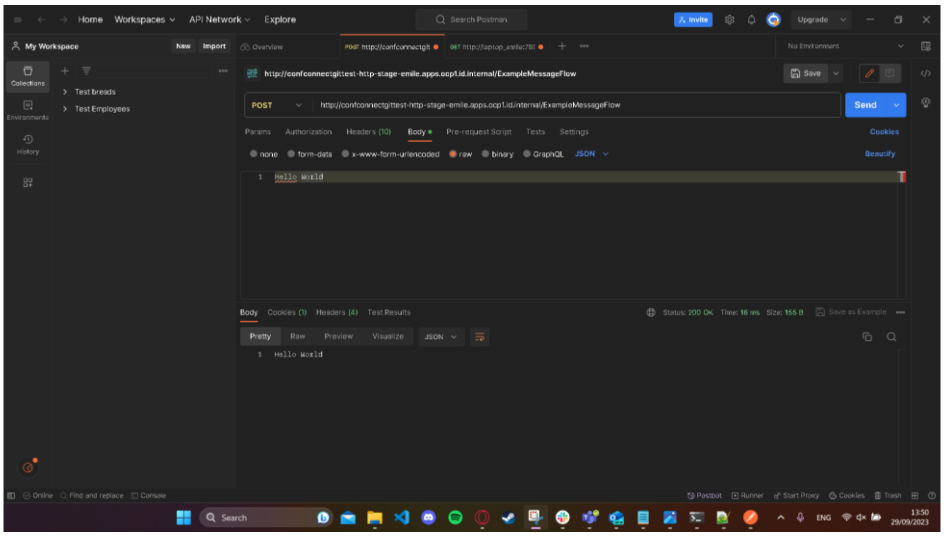
Name	Status	Location	Service
confconnectgittest-http	Accepted	http://confconnectgittest-http-stage-emile.apps.ocp1.id.internal	confconnectgittest-ir
confconnectgittest-https	Accepted	https://confconnectgittest-https-stage-emile.apps.ocp1.id.internal	confconnectgittest-ir
gitconnecttest	Accepted	https://gitconnecttest-stage-emile.apps.ocp1.id.internal	confconnectgittest-ir
runtime-stage-http	Accepted	http://runtime-stage-http-stage-emile.apps.ocp1.id.internal	runtime-stage-ir
runtime-stage-https	Accepted	https://runtime-stage-https-stage-emile.apps.ocp1.id.internal	runtime-stage-ir
webui-test	Accepted	https://webui-test-stage-emile.apps.ocp1.id.internal	runtime-stage-ir

For example:

For me to access the web UI of the project I made for this tutorial, I click the link next to 'gitconnectedtest'

ps: to do API requests to your application you can enter the 'HTTP' link of your route and add the app you want to access.

In this example I sent a post request with postman to see if my application is running and responding.



POST http://confconnectgittest-http-stage-emile.apps.ocp1.id.internal/ExampleMessageFlow

Body (raw, JSON):

```
Hello World
```

Status: 200 OK Time: 18 ms Size: 155 B

```
Hello World
```

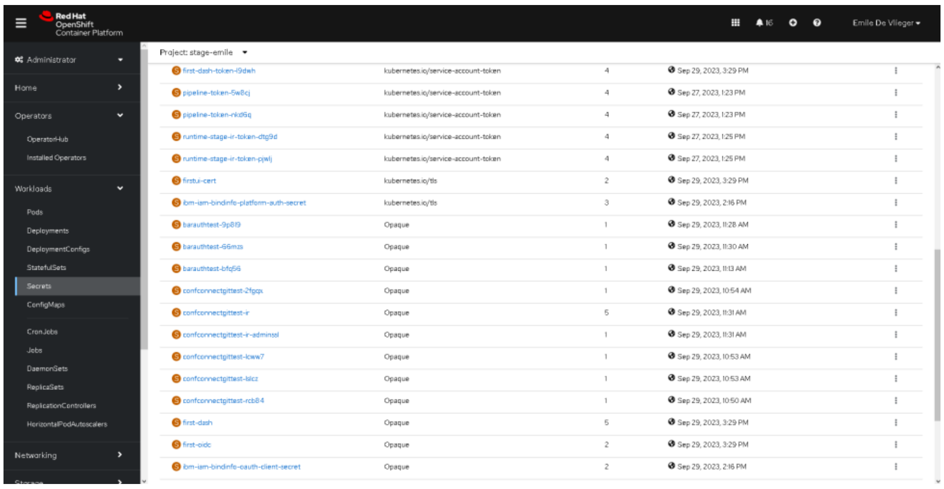
2.1.6 Accessing our web UI

When clicking on the link we created in the previous step we will be met with the login screen of the IBM web UI. However, we have not declared a user and/or password.

No worries as 'Secrets' come to the rescue!

In our menu on the left, click 'Operators' and select 'Secrets'.

Choose your project in the top left dropdown.



Project: stage-emile

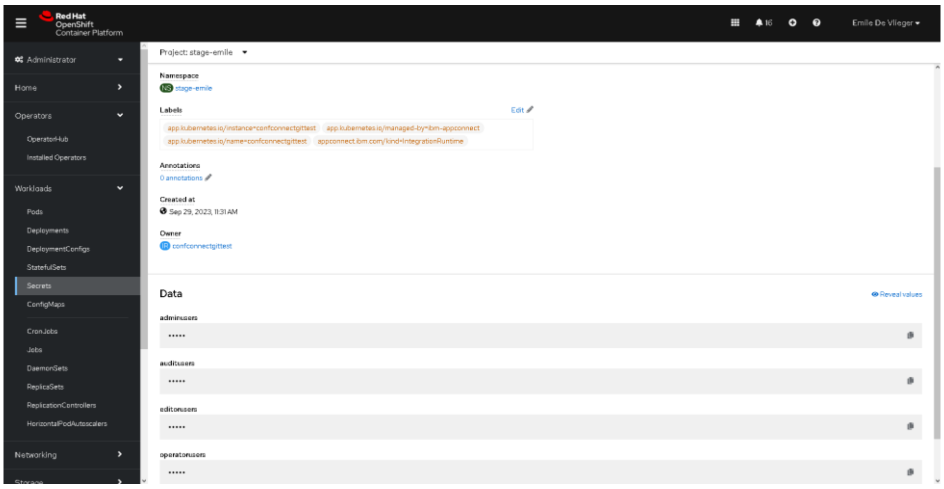
first-dash-token-l9dwh	kubernetes.io/service-account-token	4	Sep 29, 2023, 3:29 PM
pipeline-token-5w8cj	kubernetes.io/service-account-token	4	Sep 27, 2023, 1:23 PM
pipeline-token-nkd6q	kubernetes.io/service-account-token	4	Sep 27, 2023, 1:23 PM
runtime-stage-ir-token-dtg9d	kubernetes.io/service-account-token	4	Sep 27, 2023, 1:25 PM
runtime-stage-ir-token-pjwlj	kubernetes.io/service-account-token	4	Sep 27, 2023, 1:25 PM
firstui-cert	kubernetes.io/tls	2	Sep 29, 2023, 3:29 PM
ibm-iam-bindinfo-platform-auth-secret	kubernetes.io/tls	3	Sep 29, 2023, 2:16 PM
barauthtest-9p8l9	Opaque	1	Sep 29, 2023, 11:28 AM
barauthtest-66mzs	Opaque	1	Sep 29, 2023, 11:30 AM
barauthtest-bfq56	Opaque	1	Sep 29, 2023, 11:13 AM
confconnectgittest-2fgqx	Opaque	1	Sep 29, 2023, 10:54 AM
confconnectgittest-ir	Opaque	5	Sep 29, 2023, 11:31 AM
confconnectgittest-ir-adminssl	Opaque	1	Sep 29, 2023, 11:31 AM
confconnectgittest-lcww7	Opaque	1	Sep 29, 2023, 10:53 AM
confconnectgittest-blcz	Opaque	1	Sep 29, 2023, 10:53 AM
confconnectgittest-rcb84	Opaque	1	Sep 29, 2023, 10:50 AM
first-dash	Opaque	5	Sep 29, 2023, 3:29 PM
first-oidc	Opaque	2	Sep 29, 2023, 3:29 PM
ibm-iam-bindinfo-oauth-client-secret	Opaque	2	Sep 29, 2023, 2:16 PM

Your credentials will always be in a "opaque" type secret, so clicking the "type" will help you find it more easily

Next up choose the secret which is naming goes as follows: "<integration runtime-name> - ir". (not with "-adminssl")

Click on your secret.

Scroll down until you see the "adminusers" field and copy the credentials.



Project: stage-emile

Namespace
NS stage-emile

Labels
app.kubernetes.io/instance=confconnectgittest
app.kubernetes.io/managed-by=ibm-appconnect
app.kubernetes.io/name=confconnectgittest
appconnect.ibm.com/kind=IntegrationRuntime

Annotations
0 annotations

Created at
Sep 29, 2023, 11:31 AM

Owner
IR confconnectgittest

Data

adminusers
•••••

auditusers
•••••

editorusers
•••••

operatorusers
•••••

When pasting them in notepad, the first half is your username and the second half is your password.

Now go to our created route, click on the link and enter your credentials.

If you have done everything correctly, you should be in the webUI.

2.2 Deploy App Connect application with ArgoCD

In this tutorial we will create a deployed ACE application through ArgoCD. ArgoCD is a pipelining tool that manages your cluster to assure stateful deployment.

We create an ArgoCD instance, link it with a repo and deploy the right application.

2.2.1 Prerequisites

- Command line tool "**OC**"
- Command line tool "**kubectl**"

2.2.2 Steps:

- Links for GitLab repo's.
- Creating the namespace.
- Creating the ArgoCD instance.
- Creating an application in ArgoCD.
    1. Accessing our platform.
    2. Creating an ACE operand.
- Testing.

2.2.3 Links for GitLab repo's

https://gitlab.com/EmileDV/argocd-install

https://gitlab.com/EmileDV/argocd

Change files to your specific namespace

- Download the files within this git repo.
- Change all the "namespace" parameters within these files to your created namespace.
- Create a repo with the same structure but with the corrected namespaces.
- Use this new repo where we use the git repo listed above.

2.2.4 Creating the namespace

Sidenote:

The namespace that is used in this tutorial is already used so in order for others to follow this tutorial you will have to change the "namespace" parameter in almost all files that are used. The GitLab links above have downloadable files that you can change locally and push to you own repo.

All steps described in this tutorial are crucial and should be followed. So only the namespaces should be changed and the GitLab links should be altered to represent you own specific versions.

Above you will find a specific sections with instruction on where to change the namespaces.

In your cmd, login to OpenShift:

in the OpenShift console, in the top-right, you will find your name. click on it and click "copy login command".

Click the second option "openshift AD" and log in with your credentials and click "display token". Copy the first token and paste it in you command line.

Copy the following command and change the namespace (in this case "argocd") to your liking:

```
kubectl create namespace argocd
```

2.2.5 Creating the ArgoCD instance

We will create our ArgoCD instance through our cmd with the following command:

```
kubectl apply -n argocd -f https://gitlab.com/EmileDV/argocd-install/-
/raw/main/install.yml?ref_type=heads
```

Sidenote:

The file in this git repo is an altered version of the following original:

https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml

This file is altered as followed

When using the original, we found that the file requested restricted access. In order to change this we commented out 9 lines. These 9 lines are exactly the same and go as follows:

```
#seccompProfile:
        #type: RuntimeDefault
```

We have commented out these two lines 9 times which solves our access problems.

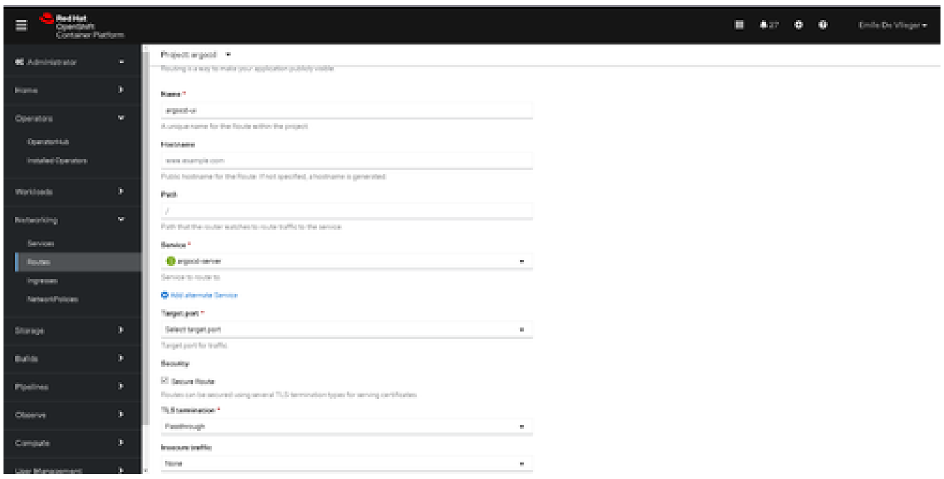
2.2.6 Accessing our platform

After all pods are created and deployed, we will make a route to access our argocd ui.

Navigate to the “Routes” map within “Networking” and click “Create Route”

Within this page enter “argocd-ui” as name, “argocd-server” as Service and “https” as target port.

Lastly enter “Passthrough” and “none” within the “TLS “fields:



Project: argocd

Routing is a way to make your application publicly visible

Name *
argocd-ui
A unique name for the Route within the project

Hostname
www.example.com
Public hostname for the Route. If not specified, a hostname is generated.

Path
/
Path that the router watches to route traffic to the service.

Service *
argocd-server
Service to route to.

Add alternate Service

Target port *
Select target port
Target port for traffic.

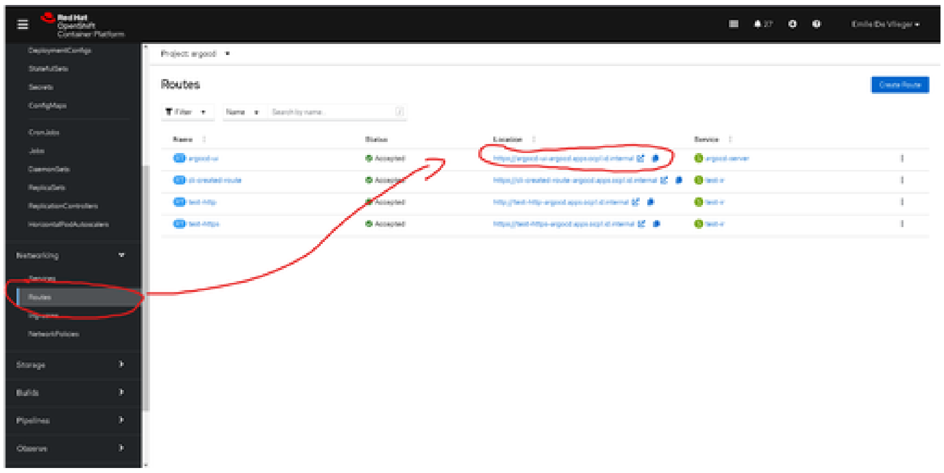
Security
Secure Route [✓]
Routes can be secured using several TLS termination types for serving certificates.

TLS termination *
Passthrough

Insecure traffic
None

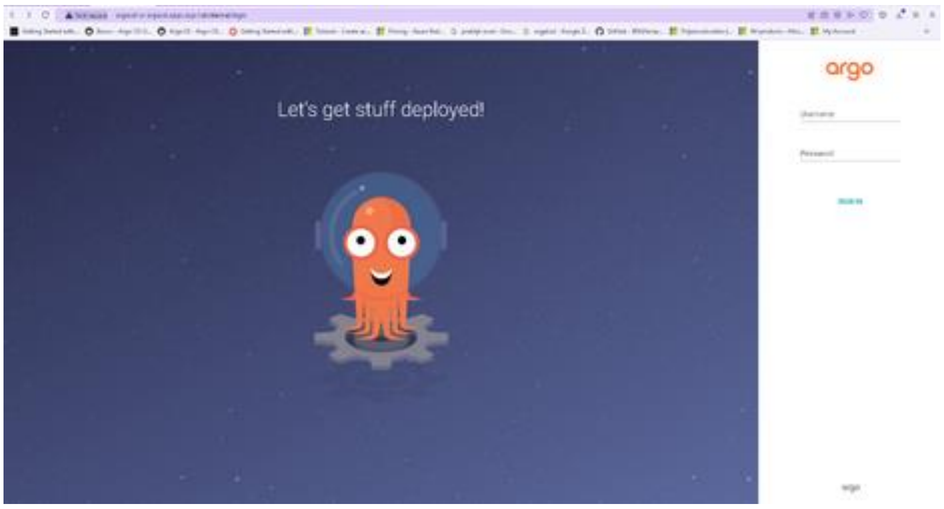
```
kind: Route
apiVersion: route.openshift.io/v1
metadata:
  name: argocd-ui
  namespace: argocd
  uid: 7f28e21e-4fed-46bb-9a87-af4376dca118
  resourceVersion: '515633455'
  creationTimestamp: '2023-10-26T09:50:36Z'
  labels:
    app.kubernetes.io/component: server
    app.kubernetes.io/name: argocd-server
    app.kubernetes.io/part-of: argocd
  annotations:
    openshift.io/host.generated: 'true'
  managedFields:
    - manager: Mozilla
      operation: Update
      apiVersion: route.openshift.io/v1
      time: '2023-10-26T09:50:36Z'
      fieldsType: FieldsV1
      fieldsV1:
        'f:metadata':
          'f:labels':
            .: {}
            'f:app.kubernetes.io/component': {}
            'f:app.kubernetes.io/name': {}
            'f:app.kubernetes.io/part-of': {}
        'f:spec':
          'f:port':
            .: {}
            'f:targetPort': {}
          'f:tls':
            .: {}
            'f:termination': {}
          'f:to':
            'f:kind': {}
            'f:name': {}
            'f:weight': {}
          'f:wildcardPolicy': {}
    - manager: openshift-router
      operation: Update
      apiVersion: route.openshift.io/v1
      time: '2023-10-26T09:50:36Z'
      fieldsType: FieldsV1
      fieldsV1:
        'f:status':
          'f:ingress': {}
      subresource: status
spec:
  host: argocd-ui-argocd.apps.ocp1.id.internal
  to:
    kind: Service
    name: argocd-server
    weight: 100
  port:
    targetPort: https
  tls:
    termination: passthrough
  wildcardPolicy: None
status:
  ingress:
    - host: argocd-ui-argocd.apps.ocp1.id.internal
      routerName: default
      conditions:
        - type: Admitted
          status: 'True'
          lastTransitionTime: '2023-10-26T09:50:36Z'
      wildcardPolicy: None
      routerCanonicalHostname: router-default.apps.ocp1.id.internal
```

Click the link found the "Location" header to navigate to the ArgoCD UI.



Name	Status	Location	Service
argocd-ui	Accepted	[illegible]	argocd-server
cli-created-route	Accepted	[illegible]	test-v
test-http	Accepted	[illegible]	test-v
test-https	Accepted	[illegible]	test-v

This will navigate you to the login screen of you ArgoCD instance



Now in your cmd enter the following command:

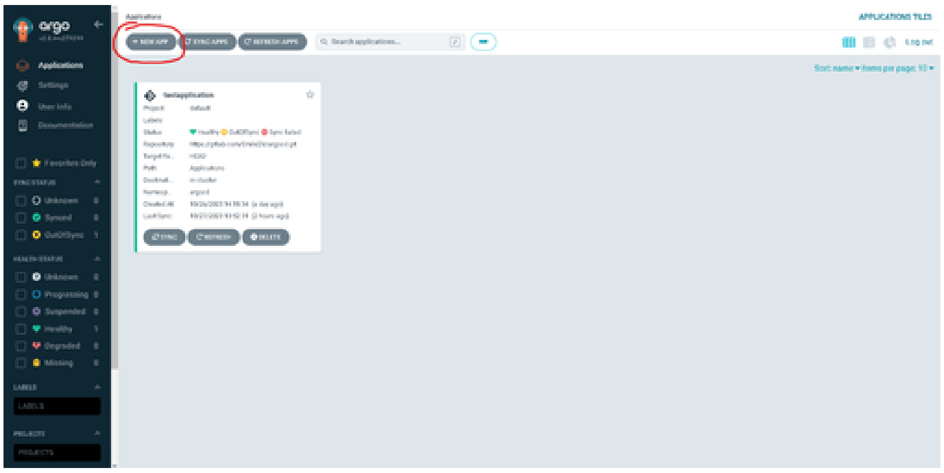
```
kubectl -n argocd get secret argocd-initial-admin-secret -o
jsonpath="{.data.password}" | base64 -d; echo
```

This will give you the password for the "admin" user for your ArgoCD platform.

Now enter "admin" as username and the give password.

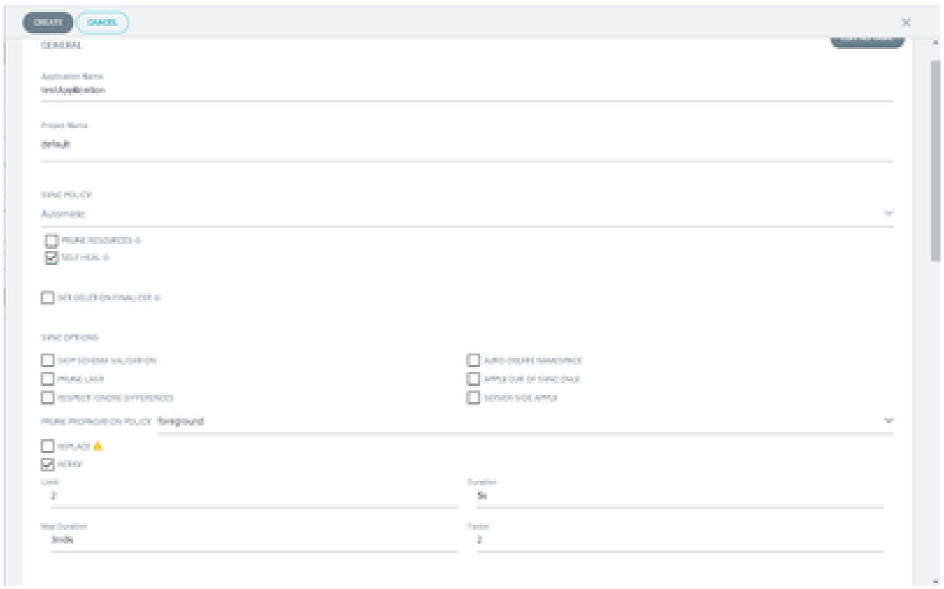
2.2.7 Creating an application

In your UI click "+ NEW APP" in the top left



In the "GENERAL" tab you can choose the name you want to give this application, for tutorial purposes we named it "testapplication"

These are the settings we chose for this version and recommend using these settings for the sake of simplicity.



GENERAL

Application Name
testApplication

Project Name
default

SYNC POLICY
Automatic

☐ PRUNE RESOURCES
☑ SELF HEAL

☐ SET DELETION FINALIZER

SYNC OPTIONS

☐ SKIP SCHEMA VALIDATION	☐ AUTO-CREATE NAMESPACE
☐ PRUNE LAST	☐ APPLY OUT OF SYNC ONLY
☐ RESPECT IGNORE DIFFERENCES	☐ SERVER-SIDE APPLY

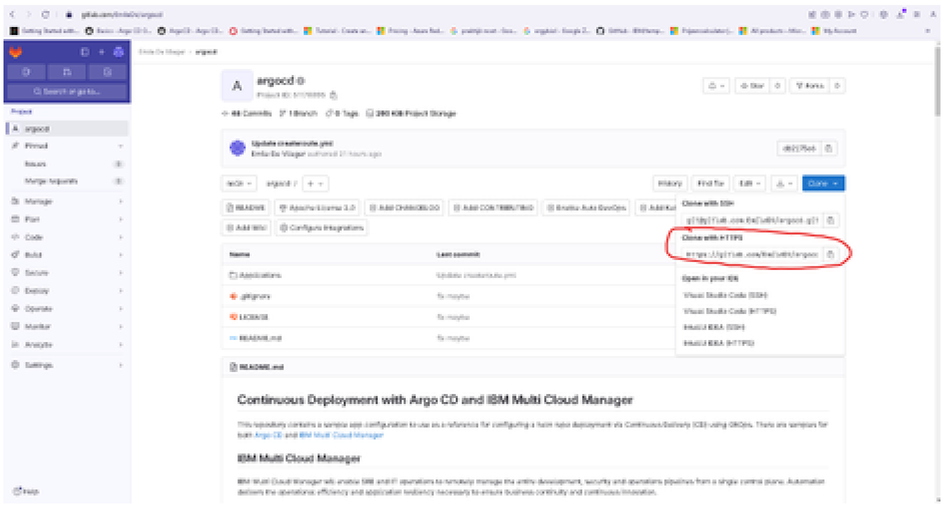
PRUNE PROPAGATION POLICY: foreground

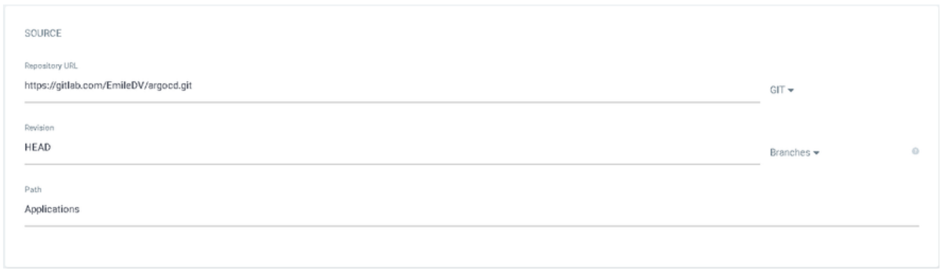
☐ REPLACE
☑ RETRY

Limit: 2	Duration: 5s
Max Duration: 3m0s	Factor: 2

In the "SOURCE" tab enter your Git repo under "Repository URL"

and the "Applications" map in the "PATH" input field



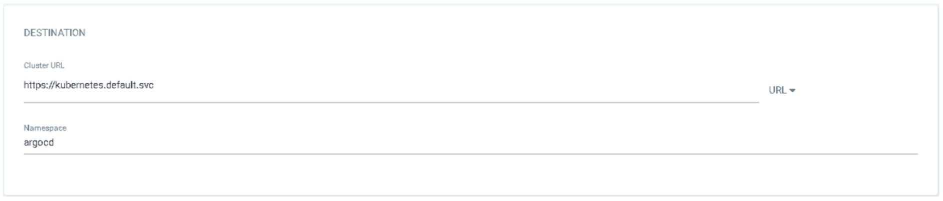

SOURCE

Repository URL
https://gitlab.com/EmileDV/argocd.git GIT

Revision
HEAD Branches

Path
Applications

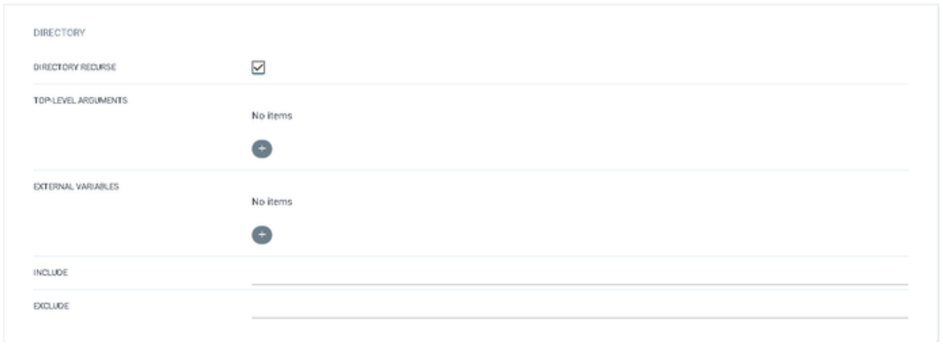
In the "DESTINATION" tab enter the following (with "namespace" being your created namespace)


DESTINATION

Cluster URL
https://kubernetes.default.svc URL

Namespace
argocd

In the "DIRECTORY" tab we check the "directory recurse" checkbox


DIRECTORY

DIRECTORY RECURSE ☑

TOP-LEVEL ARGUMENTS
No items

EXTERNAL VARIABLES
No items

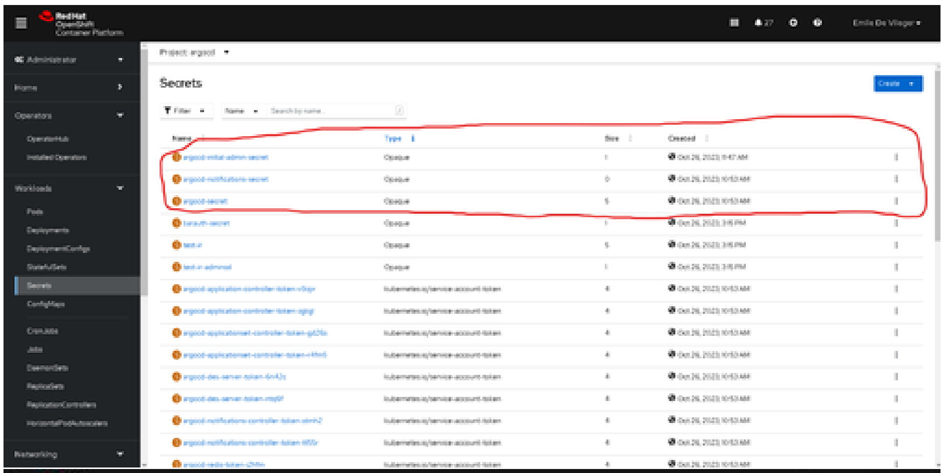
INCLUDE

EXCLUDE

2.2.8 Testing

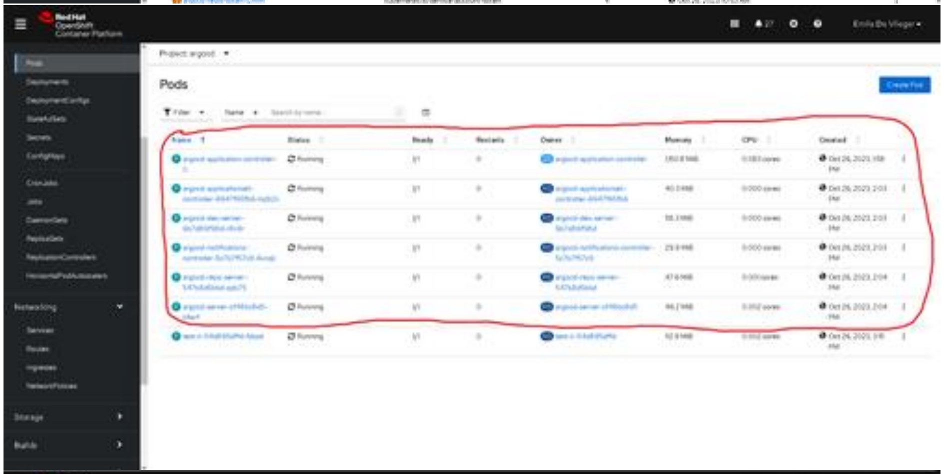
If everything has been done and created correctly you should see the following deployments on your openshift console.

2.2.8.1 What's created with ArgoCD deployment



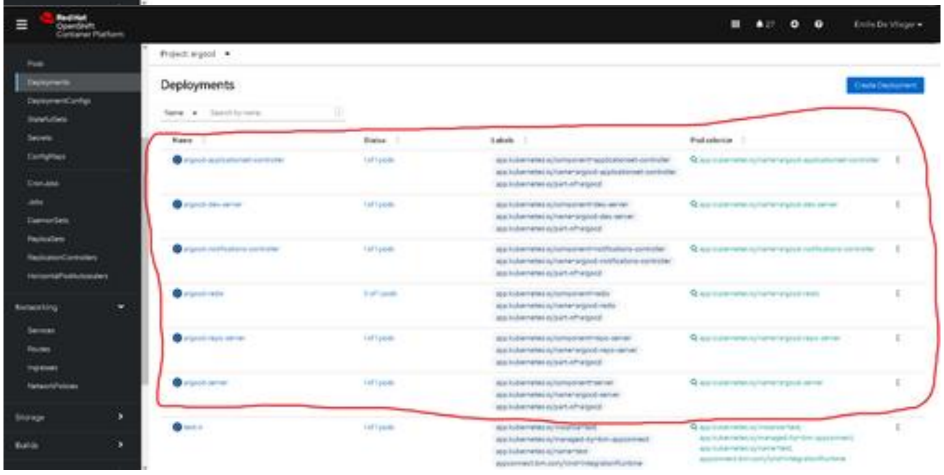
Secrets

Name	Type	Size	Created
argocd-initial-admin-secret	Opaque	1	Oct 26, 2023, 11:47 AM
argocd-notifications-secret	Opaque	0	Oct 26, 2023, 10:53 AM
argocd-secret	Opaque	5	Oct 26, 2023, 10:53 AM
[illegible]-secret	Opaque	1	Oct 26, 2023, 3:15 PM
test-ir	Opaque	5	Oct 26, 2023, 3:15 PM
test-ir-[illegible]	Opaque	1	Oct 26, 2023, 3:15 PM
argocd-application-controller-token-[illegible]	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-application-controller-token-[illegible]	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-applicationset-controller-token-[illegible]	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-applicationset-controller-token-[illegible]	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-dex-server-token-[illegible]	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-dex-server-token-[illegible]	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-notifications-controller-token-[illegible]	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-notifications-controller-token-[illegible]	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-redis-token-[illegible]	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM



Pods

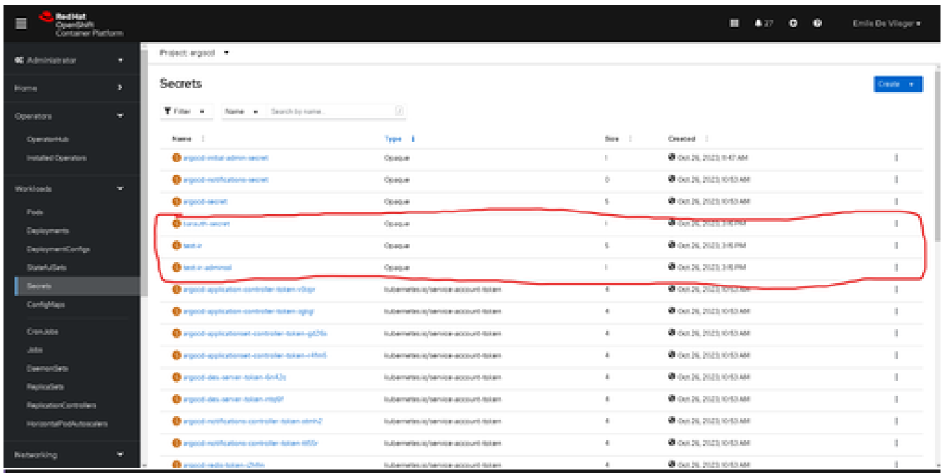
Name	Status	Ready	Restarts	Owner	Memory	CPU	Created
argocd-application-controller-0	Running	1/1	0	argocd-application-controller	[illegible] MiB	[illegible] cores	Oct 26, 2023, 1:58 PM
argocd-applicationset-controller-[illegible]	Running	1/1	0	argocd-applicationset-controller-[illegible]	[illegible] MiB	0.000 cores	Oct 26, 2023, 2:03 PM
argocd-dex-server-[illegible]	Running	1/1	0	argocd-dex-server-[illegible]	[illegible] MiB	0.000 cores	Oct 26, 2023, 2:03 PM
argocd-notifications-controller-[illegible]	Running	1/1	0	argocd-notifications-controller-[illegible]	[illegible] MiB	0.000 cores	Oct 26, 2023, 2:03 PM
argocd-repo-server-[illegible]	Running	1/1	0	argocd-repo-server-[illegible]	[illegible] MiB	[illegible] cores	Oct 26, 2023, 2:04 PM
argocd-server-[illegible]	Running	1/1	0	argocd-server-[illegible]	[illegible] MiB	[illegible] cores	Oct 26, 2023, 2:04 PM
test-ir-[illegible]	Running	1/1	0	test-ir-[illegible]	[illegible] MiB	[illegible] cores	Oct 26, 2023, 3:15 PM



Deployments

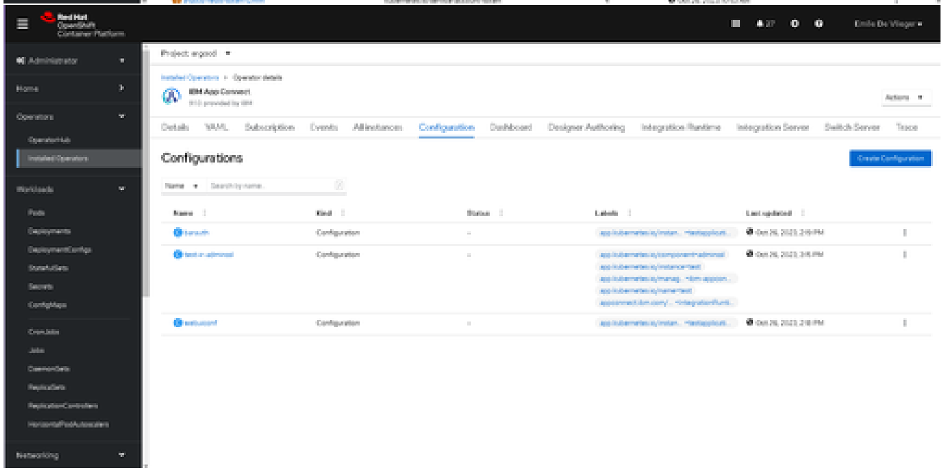
Name	Status	Labels	Pod selector
argocd-applicationset-controller	1 of 1 pods	app.kubernetes.io/component=applicationset-controller app.kubernetes.io/name=argocd-applicationset-controller app.kubernetes.io/part-of=argocd	app.kubernetes.io/name=argocd-applicationset-controller
argocd-dex-server	1 of 1 pods	app.kubernetes.io/component=dex-server app.kubernetes.io/name=argocd-dex-server app.kubernetes.io/part-of=argocd	app.kubernetes.io/name=argocd-dex-server
argocd-notifications-controller	1 of 1 pods	app.kubernetes.io/component=notifications-controller app.kubernetes.io/name=argocd-notifications-controller app.kubernetes.io/part-of=argocd	app.kubernetes.io/name=argocd-notifications-controller
argocd-redis	1 of 1 pods	app.kubernetes.io/component=redis app.kubernetes.io/name=argocd-redis app.kubernetes.io/part-of=argocd	app.kubernetes.io/name=argocd-redis
argocd-repo-server	1 of 1 pods	app.kubernetes.io/component=repo-server app.kubernetes.io/name=argocd-repo-server app.kubernetes.io/part-of=argocd	app.kubernetes.io/name=argocd-repo-server
argocd-server	1 of 1 pods	app.kubernetes.io/component=server app.kubernetes.io/name=argocd-server app.kubernetes.io/part-of=argocd	app.kubernetes.io/name=argocd-server
test-ir	1 of 1 pods	app.kubernetes.io/instance=test app.kubernetes.io/managed-by=ibm-appconnect app.kubernetes.io/name=test appconnect.ibm.com/kind=IntegrationRuntime	app.kubernetes.io/instance=test, app.kubernetes.io/managed-by=ibm-appconnect, app.kubernetes.io/name=test, appconnect.ibm.com/kind=IntegrationRuntime

What’s created with the ACE application



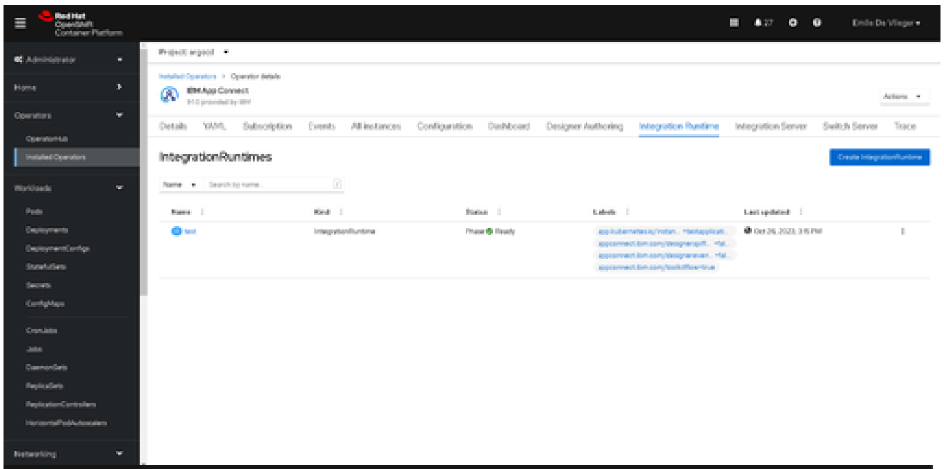
Secrets

Name	Type	Size	Created
argocd-initial-admin-secret	Opaque	1	Oct 26, 2023, 11:47 AM
argocd-notifications-secret	Opaque	0	Oct 26, 2023, 10:53 AM
argocd-secret	Opaque	5	Oct 26, 2023, 10:53 AM
basauth-secret	Opaque	1	Oct 26, 2023, 3:15 PM
test-ir	Opaque	5	Oct 26, 2023, 3:15 PM
test-ir-adminssl	Opaque	1	Oct 26, 2023, 3:15 PM
argocd-application-controller-token-v5bpr	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-application-controller-token-qjkgl	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-applicationset-controller-token-gd26x	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-applicationset-controller-token-r4hm5	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-dex-server-token-4n42s	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-dex-server-token-mtp9f	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-notifications-controller-token-ctmh2	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-notifications-controller-token-t65r	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-redis-token-z7kfm	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM



Configurations

Name	Kind	Status	Labels	Last updated
basauth	Configuration	-	app.kubernetes.io/instan... =testapplicati...	Oct 26, 2023, 2:19 PM
test-ir-adminssl	Configuration	-	app.kubernetes.io/component=adminssl app.kubernetes.io/instance=test app.kubernetes.io/manag... =ibm-appcon... app.kubernetes.io/name=test appconnect.ibm.com/... =integrationRunti...	Oct 26, 2023, 3:15 PM
[illegible]	Configuration	-	app.kubernetes.io/instan... =testapplicati...	Oct 26, 2023, 2:18 PM


Project: argocd

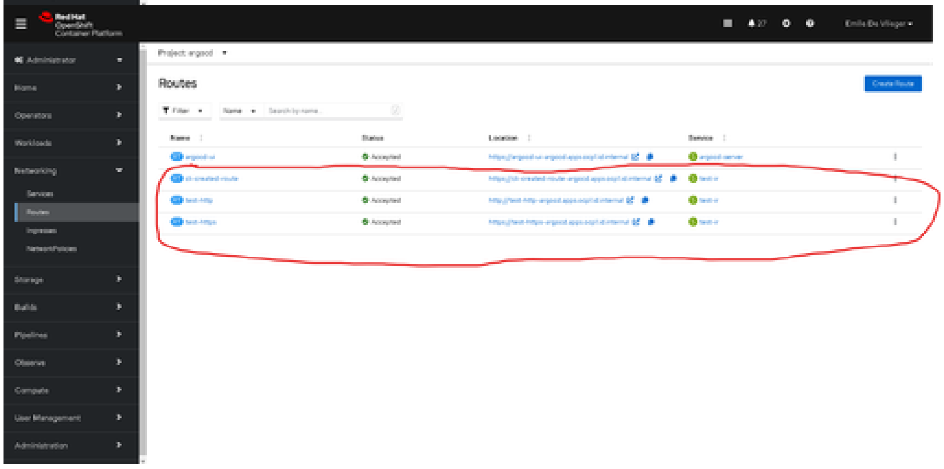
Installed Operators > Operator details

IBM App Connect

Details | YAML | Subscription | Events | All instances | Configuration | Dashboard | Designer Authoring | Integration Runtime | Integration Server | Switch Server | Trace

IntegrationRuntimes

Name	Kind	Status	Labels	Last updated
test	IntegrationRuntime	Phase: Ready	app.kubernetes.io/instan... =testapplicati... appconnect.ibm.com/designerapifl... =fal... appconnect.ibm.com/designerevan... =fal... appconnect.ibm.com/toolkitflow=true	Oct 26, 2023, 3:15 PM

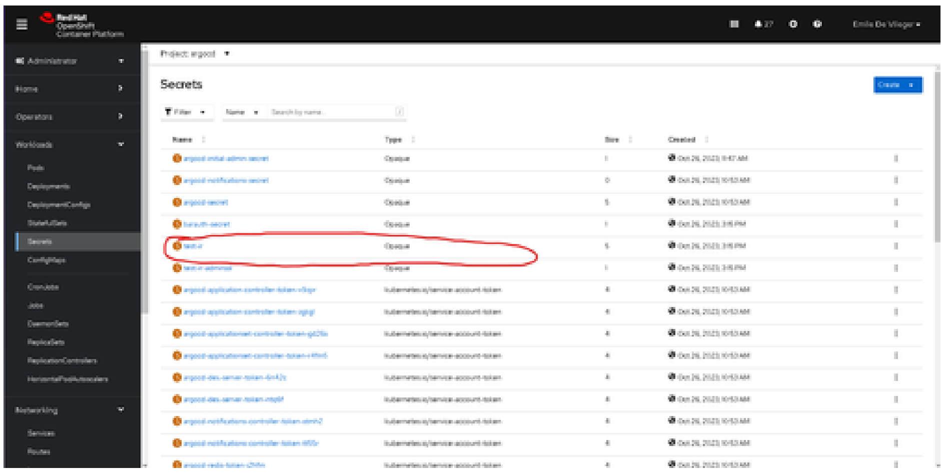

Project: argocd

Routes

Name	Status	Location	Service
argocd-ui	Accepted	https://argocd-ui-argocd.apps.ocp1.sl.internal	argocd-server
cli-created-route	Accepted	https://cli-created-route-argocd.apps.ocp1.sl.internal	test-ir
test-http	Accepted	http://test-http-argocd.apps.ocp1.sl.internal	test-ir
test-https	Accepted	https://test-https-argocd.apps.ocp1.sl.internal	test-ir

When clicking on the route "cli-created-route" under the "Location" header, you should be navigated to the login page.

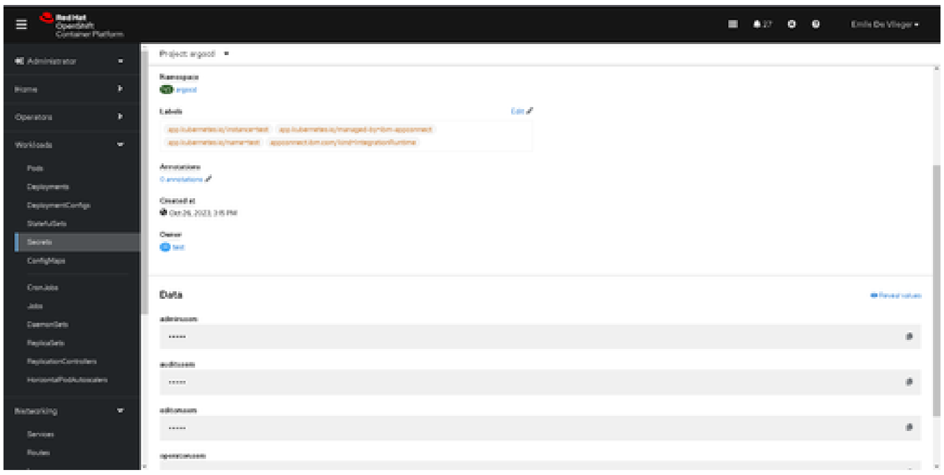
You login credentials are located within the secret "test-ir-".


Project: argocd

Secrets

Name	Type	Size	Created
argocd-initial-admin-secret	Opaque	1	Oct 26, 2023, 9:47 AM
argocd-notifications-secret	Opaque	0	Oct 26, 2023, 10:53 AM
argocd-secret	Opaque	5	Oct 26, 2023, 10:53 AM
basauth-secret	Opaque	1	Oct 26, 2023, 3:15 PM
test-ir	Opaque	5	Oct 26, 2023, 3:15 PM
test-ir-adminssl	Opaque	1	Oct 26, 2023, 3:15 PM
argocd-application-controller-token-x5kgr	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-application-controller-token-zgkgl	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-applicationset-controller-token-gd25s	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-applicationset-controller-token-r4fm5	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-dex-server-token-6m42s	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-dex-server-token-mtp9f	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-notifications-controller-token-xtmh2	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-notifications-controller-token-t65fr	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM
argocd-redis-token-z2hfm	kubernetes.io/service-account-token	4	Oct 26, 2023, 10:53 AM

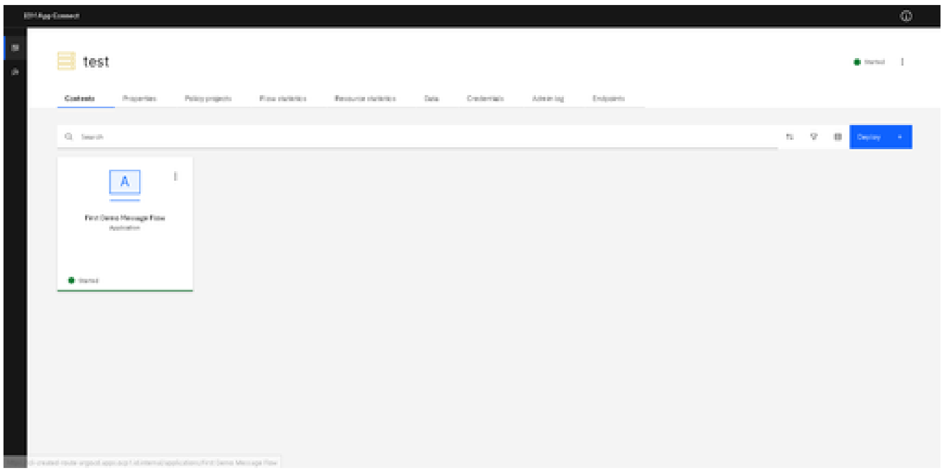
Under the “admin” field, copy the credentials and copy them in a notepad file.



Copy the first part and paste them as your “username”.

Copy the second part and paste them as your “password”.

Login and you will see your deployed application.



All done!

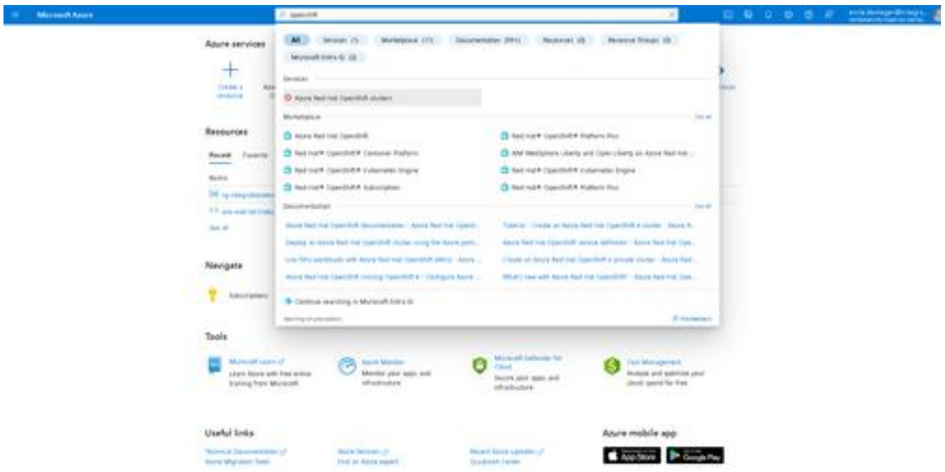
2.3 How to create an Azure OpenShift Cluster

2.3.1 Prerequisite

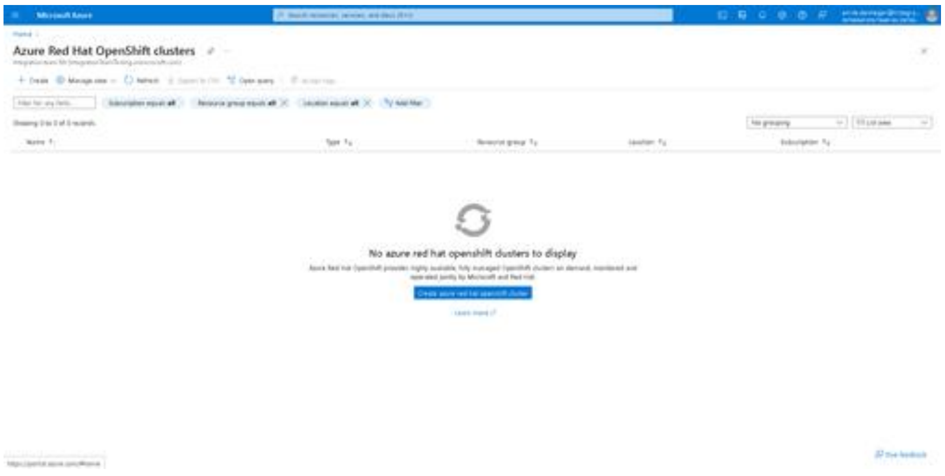
1. You should have an Azure subscription which allows you to create an OpenShift cluster (look at pricing, cheapest version is +- €1850)

2.3.2 Setting up the cluster

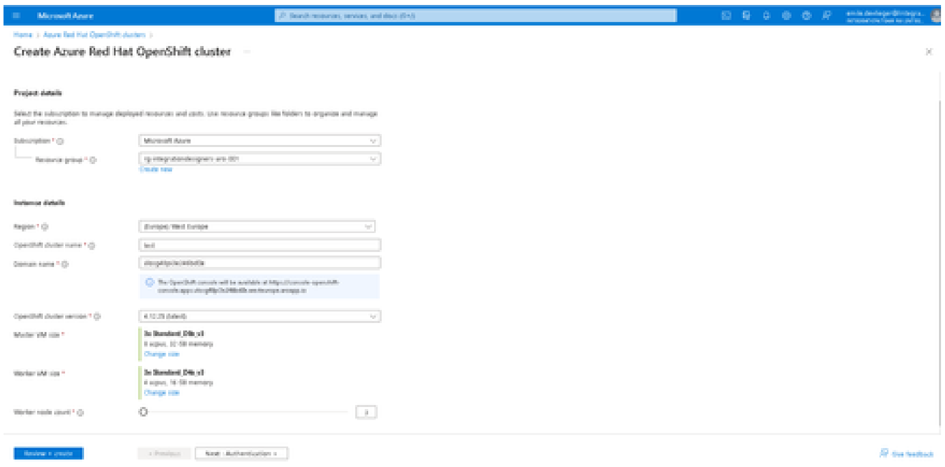
On the top of your window you will find a search bar. Type "openshift" and select the first option.



This will navigate you to the following page:


Microsoft Azure
Azure Red Hat OpenShift clusters
No azure red hat openshift clusters to display

Click "Create RedHat OpenShift cluster" or the "+ Create" button on the top left.


Microsoft Azure

Home > Azure Red Hat OpenShift clusters >

Create Azure Red Hat OpenShift cluster

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *: Microsoft Azure

Resource group *: [illegible]

Create new

Instance details

Region *: (Europe) West Europe

OpenShift cluster name *: test

Domain name *: [illegible]

The OpenShift console will be available at [illegible]

OpenShift cluster version *: [illegible]

Master VM size *: 3x Standard_D8s_v3, 8 vcpus, 32 GB memory, Change size

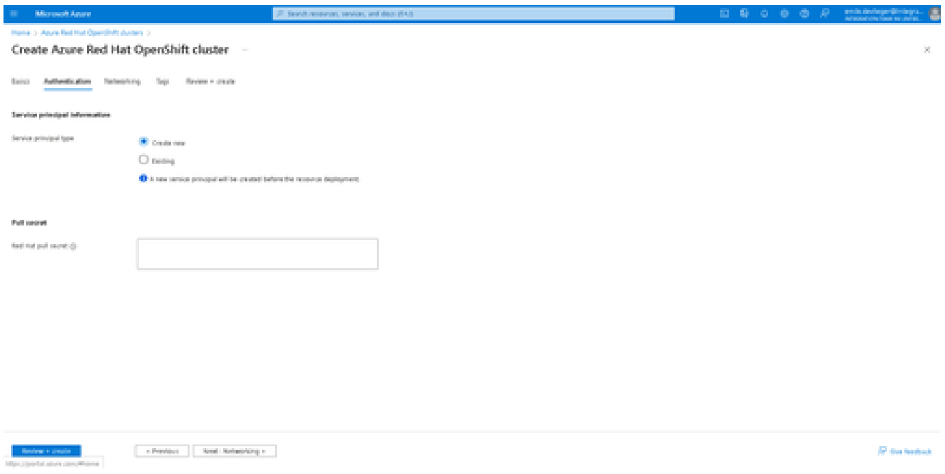
Worker VM size *: 3x Standard_D4s_v3, 4 vcpus, 16 GB memory, Change size

Worker node count *: 3

Review + create | < Previous | Next : Authentication >

In this example, where we will be working with the most light weight version of ARO, only enter the name you wish to give to your cluster. However when creating a more heavy weight cluster You can choose which master and worker nodes to select and how many.

Click "Next: Authentication"


Microsoft Azure

Home > Azure Red Hat OpenShift clusters >

Create Azure Red Hat OpenShift cluster

Basics | Authentication | Networking | Tags | Review + create

Service principal information

Service principal type: [✓ Create new] Existing

A new service principal will be created before the resource deployment.

Pull secret

Red Hat pull secret:

Review + create | < Previous | Next : Networking >

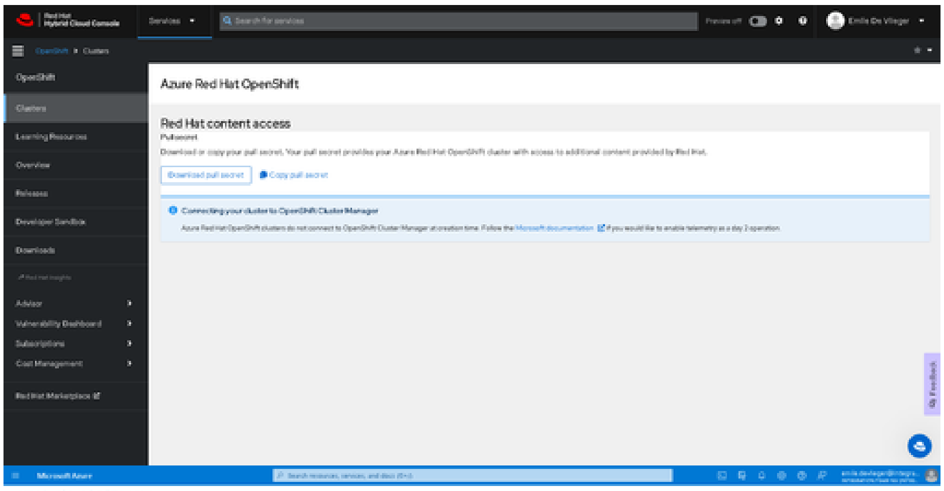
As you can see in the middle of your page. The setup asks for a "pull secret". This is very important as it allows you to access the "Operator Hub" on you OpenShift platform.

To get your pull secret go to the following link:

https://sso.redhat.com/auth/realms/redhat-external/protocol/openid-connect/auth?client_id=cloud-services&redirect_uri=https%3A%2F%2Fconsole.redhat.com%2Fopenshift%2Finstall%2Fazure%2Faro-provisioned&state=47544e44-bfea-4604-8663-b4798677c437&response_mode=fragment&response_type=code&scope=openid&nonce=d82740b4-8b59-4614-ba6f-09d97e288f62

If needed create an account and log in.

After login in you should see the option to "copy pull secret". Copy it and paste it in the input field in the setup wizard on Azure:



Azure Red Hat OpenShift

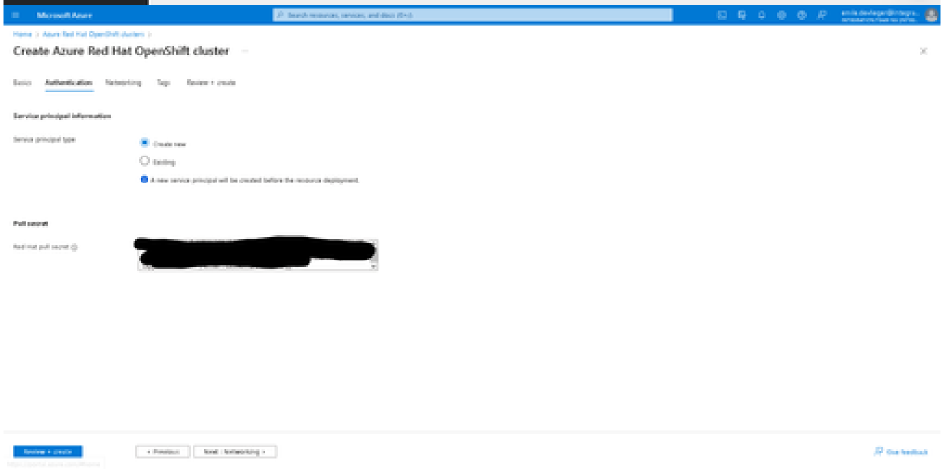
Red Hat content access

Pull secret

Download or copy your pull secret. Your pull secret provides your Azure Red Hat OpenShift cluster with access to additional content provided by Red Hat.

Connecting your cluster to OpenShift Cluster Manager

Azure Red Hat OpenShift clusters do not connect to OpenShift Cluster Manager at creation time. Follow the Microsoft documentation if you would like to enable telemetry as a day 2 operation.



Create Azure Red Hat OpenShift cluster

Service principal information

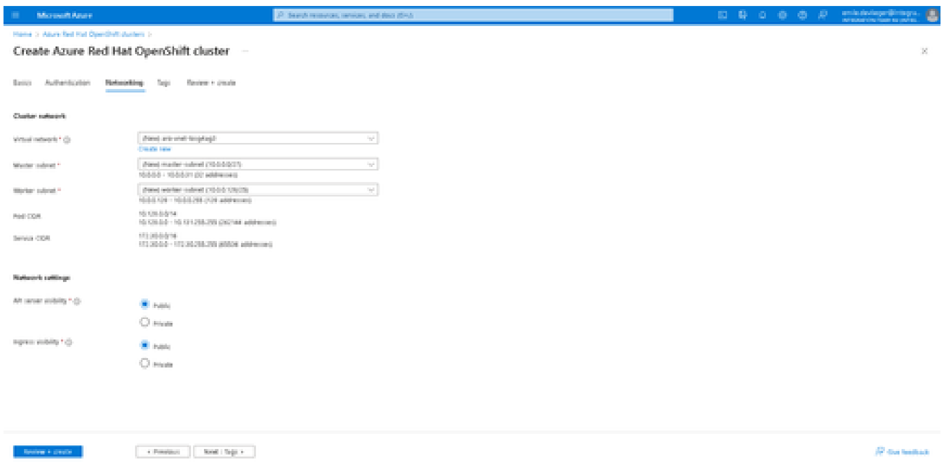
Service principal type: [✓ Create new] Existing

A new service principal will be created before the resource deployment.

Pull secret

Red Hat pull secret: [redacted]

Click “Next: Networking”



Create Azure Red Hat OpenShift cluster

Cluster network

Virtual network *	(New) aro-vnet-[illegible]
Master subnet *	(New) master-subnet (10.0.0.0/27) 10.0.0.0 - 10.0.0.31 (32 addresses)
Worker subnet *	(New) worker-subnet (10.0.0.128/25) 10.0.0.128 - 10.0.0.255 (128 addresses)
Pod CIDR	10.128.0.0/14 10.128.0.0 - 10.131.255.255 (262144 addresses)
Service CIDR	172.30.0.0/16 172.30.0.0 - 172.30.255.255 (65536 addresses)

Network settings

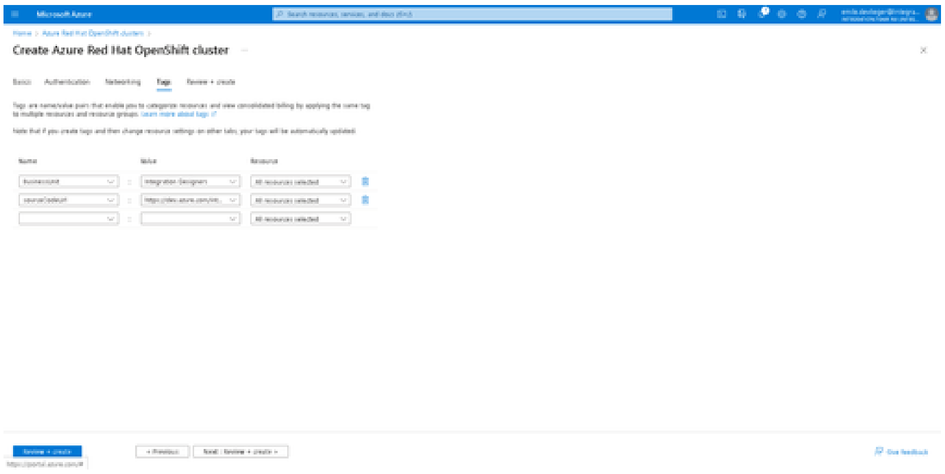
API server visibility *: [✓ Public] Private

Ingress visibility *: [✓ Public] Private

You can keep everything as it is or create a new Virtual network and subnets if you wish to do so.

(If you are using an existing virtual network, click the existing master subnet and worker subnet in the dropdowns.)

Click “Next: Tags”



You can read what Tags do in the link on the top of the page.

In this example I entered our BusinessUnit and a SourceCodeUrl (these are autocompleted)

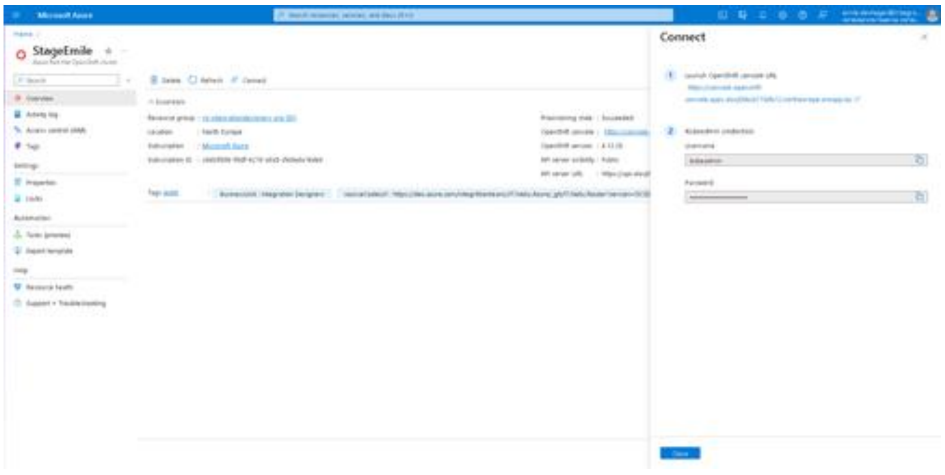
Click "Next: Review + Create"

On the next page, Azure will take some time to validate all given information. After this is done you will be able to click the **Create** Button on the bottom left.

Click "Create"

This deployment takes a good amount of time so go get yourself something to drink in the meantime.

After creation is complete you will find your credentials on the right when clicking on "connect".



Click the link above your credentials and enter your username and password.

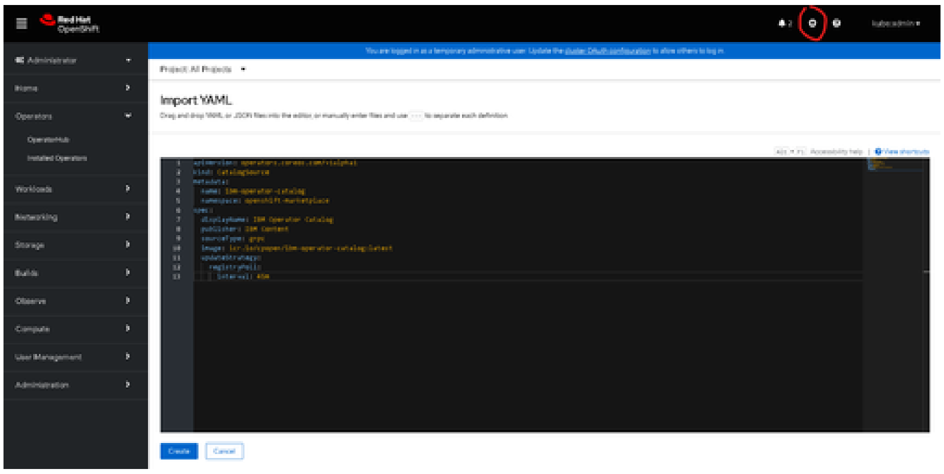
2.3.3 Introducing extra IBM operators

When looking for operator's like "IBM MQ" or "IBM App Connect" you will notice that these are not available in the operatorhub. This is because this is not a basic operator in any OpenShift subscription no matter the platform.

To get access to these operators follow this one easy step below.

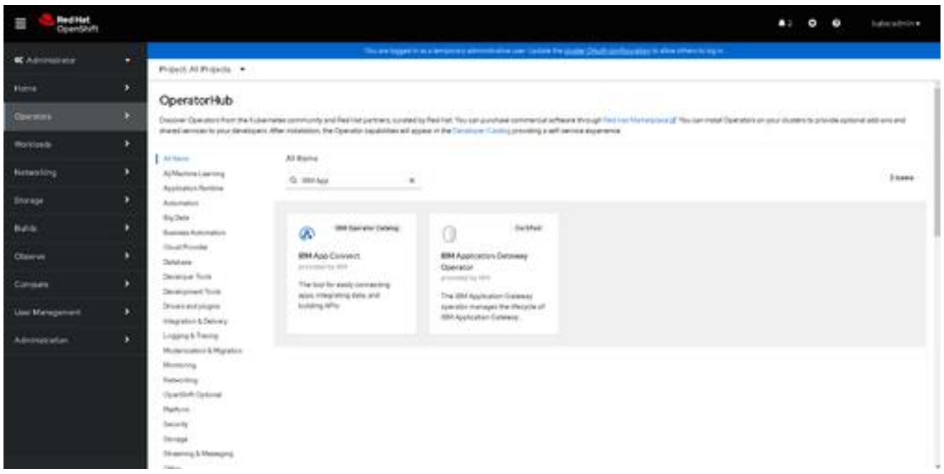
On the top right press the "+" sign and in the yaml code window enter the following code:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```



Click "Create"

After a couple minutes and a refresh you should see the added operators.



Installing the operator will be done through the installing wizard where, in this example, we install version 10.1. This version is auto selected but if not, install the auto selected version.

All done!

Recreating an ArgoCD instance in your ARO environment is exactly the same as doing it on our on-prem version (follow this guide).

Deploy App connect operand with through ArgoCD

!!!

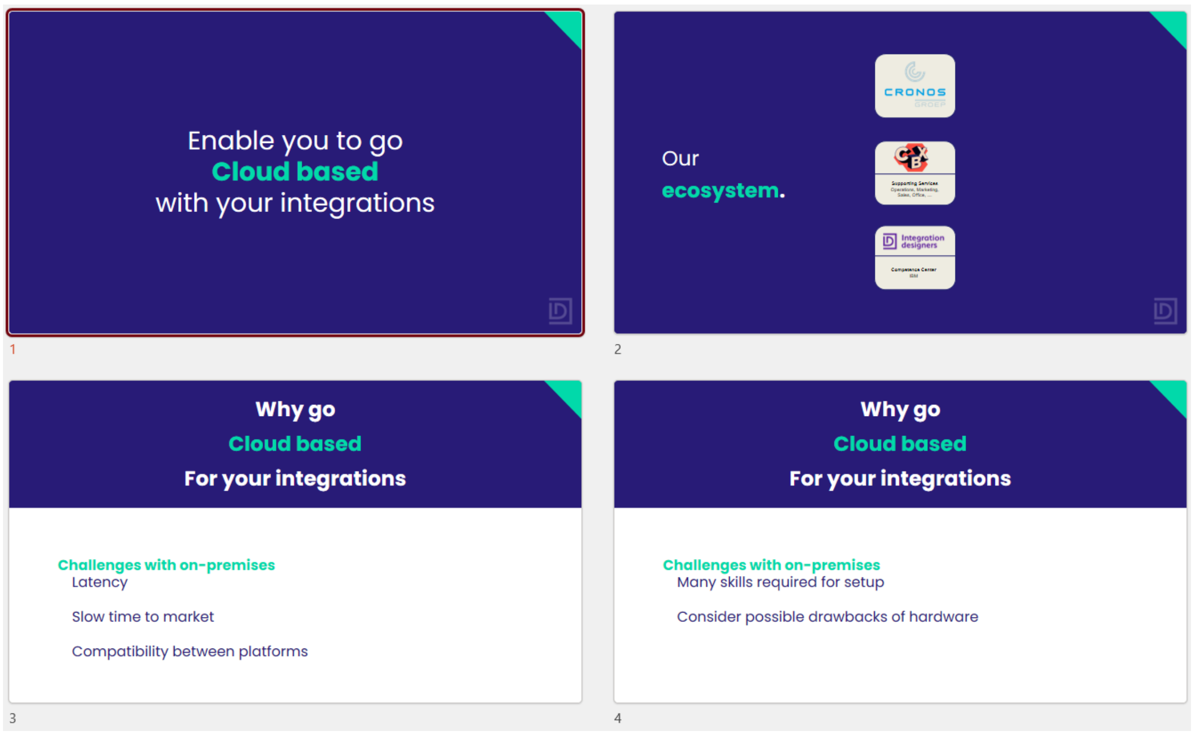
Instead of using the git repo with files for the on-prem version, use this repo.

https://gitlab.com/EmileDV/argocd-aro.git

(Do note that licenses to create an integration runtime operand differ in an ARO environment. The license to create and integration runtime has been changed here.)

2.4 Powerpoint and Pitch

2.4.1 Powerpoint



Enable you to go
Cloud based
with your integrations

1

Our
ecosystem.

2

Why go
Cloud based
For your integrations

Challenges with on-premises
Latency

Slow time to market

Compatibility between platforms

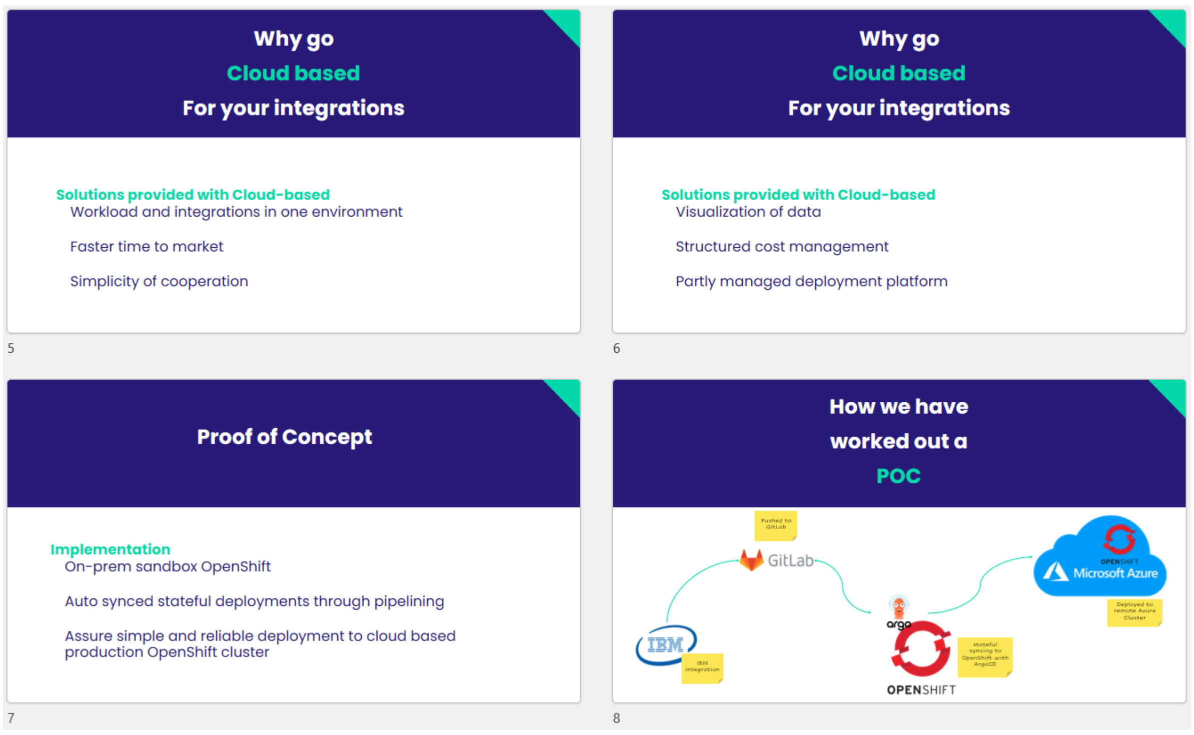
3

Why go
Cloud based
For your integrations

Challenges with on-premises
Many skills required for setup

Consider possible drawbacks of hardware

4



Why go
Cloud based
For your integrations

Solutions provided with Cloud-based
Workload and integrations in one environment

Faster time to market

Simplicity of cooperation

5

Why go
Cloud based
For your integrations

Solutions provided with Cloud-based
Visualization of data

Structured cost management

Partly managed deployment platform

6

Proof of Concept

Implementation
On-prem sandbox OpenShift

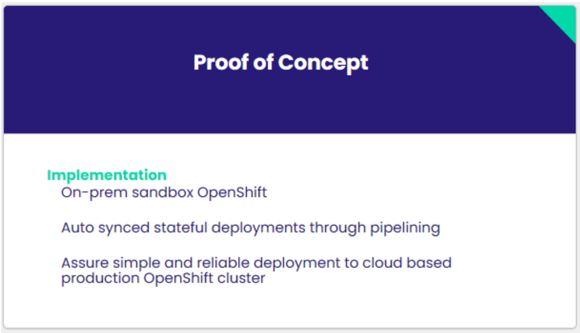
Auto synced stateful deployments through pipelining

Assure simple and reliable deployment to cloud based production OpenShift cluster

7

How we have
worked out a
POC

8



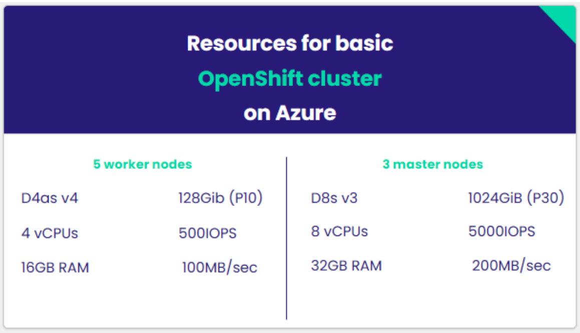
Proof of Concept

Implementation
On-prem sandbox OpenShift

Auto synced stateful deployments through pipelining

Assure simple and reliable deployment to cloud based production OpenShift cluster

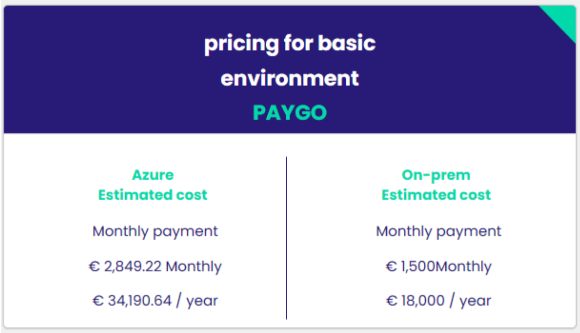
9



Resources for basic OpenShift cluster on Azure

5 worker nodes		3 master nodes	
D4as v4	128Gib (P10)	D8s v3	1024GiB (P30)
4 vCPUs	500IOPS	8 vCPUs	5000IOPS
16GB RAM	100MB/sec	32GB RAM	200MB/sec

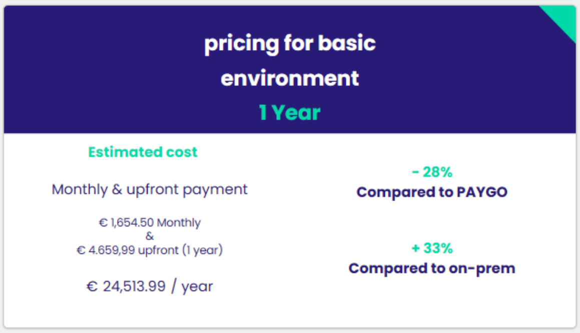
10



pricing for basic environment PAYGO

Azure Estimated cost	On-prem Estimated cost
Monthly payment	Monthly payment
€ 2,849.22 Monthly	€ 1,500Monthly
€ 34,190.64 / year	€ 18,000 / year

11



pricing for basic environment 1 Year

Estimated cost

Monthly & upfront payment

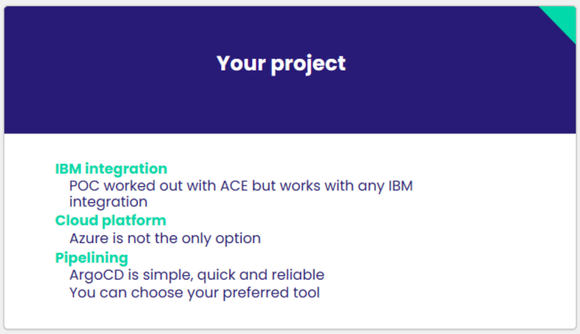
€ 1,654.50 Monthly
&
€ 4.659,99 upfront (1 year)

€ 24,513.99 / year

- 28%
Compared to PAYGO

+ 33%
Compared to on-prem

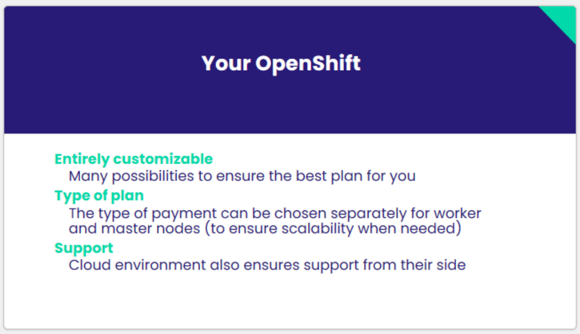
12



Your project

IBM integration
POC worked out with ACE but works with any IBM integration
Cloud platform
Azure is not the only option
Pipelining
ArgoCD is simple, quick and reliable
You can choose your preferred tool

13



Your OpenShift

Entirely customizable
Many possibilities to ensure the best plan for you
Type of plan
The type of payment can be chosen separately for worker and master nodes (to ensure scalability when needed)
Support
Cloud environment also ensures support from their side

14

Why Integration Designers

Leave your integration worries behind and gain control of your digital landscape.

Partnerships
As an IBM Business Partner, we operate at the **forefront** of IBM products.

Advice
We are not here solely to sell **licenses**. We are here to provide you with **personalized advice** and solutions that address your specific needs

Expertise
Ensuring that our knowledge is up to date with constant **knowledge sharing**, **certifications** and **badges**.

xx/xx/xxxx

15

Why Integration Designers

Size
Being part of CornerBriX means that we are part of the **biggest** integration competence center in **Belgium**

Approach
15+ years ensures a calculated approach working with **frameworks** and **best practices**

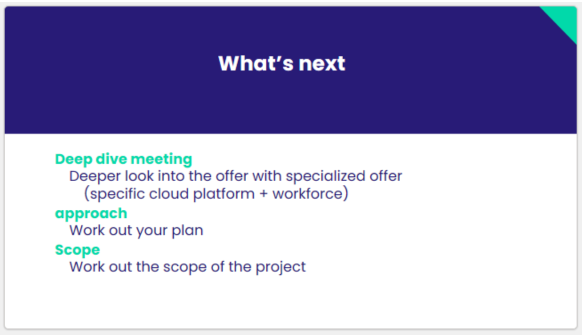
Network
Being part of CornerBriX means having a **strong network** of specialists. This ensures coverage of the **entire scope** of the project

16

Trusted By

HONDA
BNP PARIBAS
FEDNOT
LINEAS
La Mutualité Socialiste
Atlas Copco
VDAB
KBC
Q8
AGC
ABVV
Aquafin
D'Ieteren Auto
ArcelorMittal
ENGIE Electrabel

17



What's next

Deep dive meeting
Deeper look into the offer with specialized offer (specific cloud platform + workforce)
approach
Work out your plan
Scope
Work out the scope of the project

18

The Cloud Integration offer

Your own environment
Completely set up to your needs
Support
Setup and support by specialists
Managed
Managed and run to ensure health and efficiency

19

3 CONCLUSION

I was given a broad project to work on, but I believe I did a great job completing both sides of the process. I created a proof of concept that demonstrated the benefits of deploying on the cloud and how it can help in the field of integration.

Despite my background in application development and some doubts from my peers, I worked hard to finish this project. With the help of my mentor Kim, we developed a strong plan with clear checkpoints and goals, making each step of the process well-defined, challenging, and yet, still achievable.

I hope this document shows how I have grown and what I have accomplished.