Prepare a independence server (call it as support server in this doc)

* + Install docker and docker-compose
  + Install gradle command on that server
  + This server can be accessible from ICP cluster

1. **Setup Docker Repository with Harbor** **(on support server)**

<https://github.com/goharbor/harbor/blob/master/docs/installation_guide.md>

        Quick Setup:

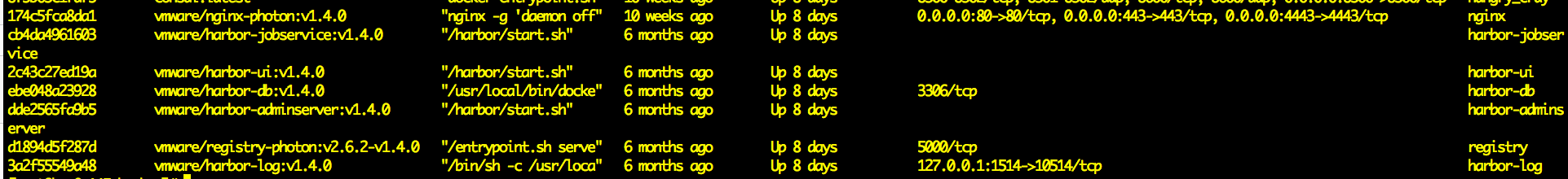
          1) Unzip package ( online or offline is okay, dependence on your server's network condition )

          2) Edit /harbor/harbor.cfg

                   \* set "hostname" to current server ip address

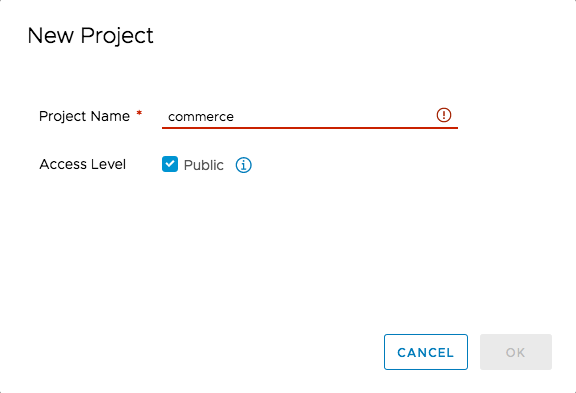
                   \* set "ui\_url\_protocol" to http

          3) Run /harbor/install.sh and waiting all docker start up success

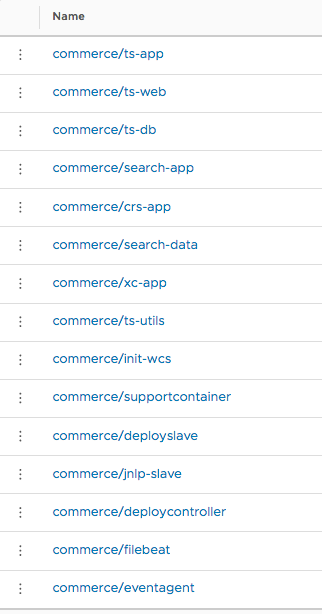


4) Access Harbor with url <http://IP_Address/harobr> ( admin/Harbor12345 )

5) Create a new project named as “commerce”



6) Upload All Commerce Docker Images



7) Make sure all Kubernetes node can pull docker image from Docker Harbor ( On all ICP Node )

* Add docker repository as insecurity mode
* Restart docker daemon
* Try to pull docker image from docker repository

1. Setup Nexus **(on support server)**

<https://hub.docker.com/r/sonatype/nexus/>

1. docker run -d -p 8081:8081 --name nexus sonatype/nexus
2. access nexus with url <http://IP_Address:8081/nexus> ( admin/admin123 )

1. Package customization package upload to Nexus **(on support server)**

* Create build.gradle file with below content

apply plugin: "maven"

apply plugin: "maven-publish"

publishing{

repositories {

maven {

url "http://<Nexus\_IP\_Address>:8081/nexus/content/repositories/releases"

credentials {

username 'admin'

password 'admin123'

}

}

}

publications {

zip(MavenPublication) {

groupId 'commerce.demo'

artifactId '<component\_name>'

version '<valida\_version>'

artifact '<path\_of\_customization\_package>/xxx.zip'

}

}

}

component\_name: search-app / ts-app / crs-app

version: you can name is as the data e.g 20181015

artifact: absolute path of zip package

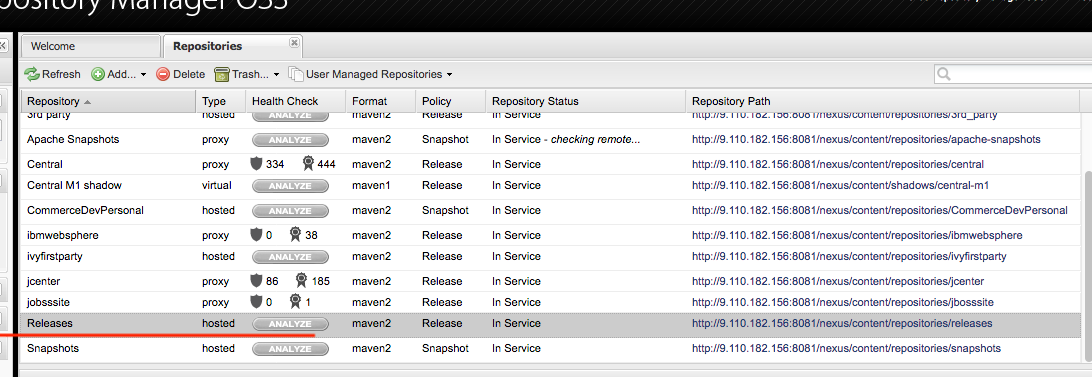
* Run gradle command to publish package to Nexus

#> gradle publish

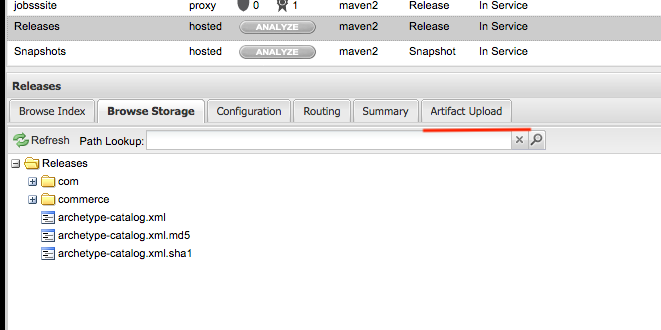
Run this command on the same directory with build.gradle

If you don’t know how to using Gradle to upload Assets, You can use Nexus UI to do that

1. Logon Nexus UI ( <http://NEXUS_IP:8081/nexus>
2. Login with default user ( admin/admin123 )
3. Select “Release” repository



1. Select “Artifact Upload” tab

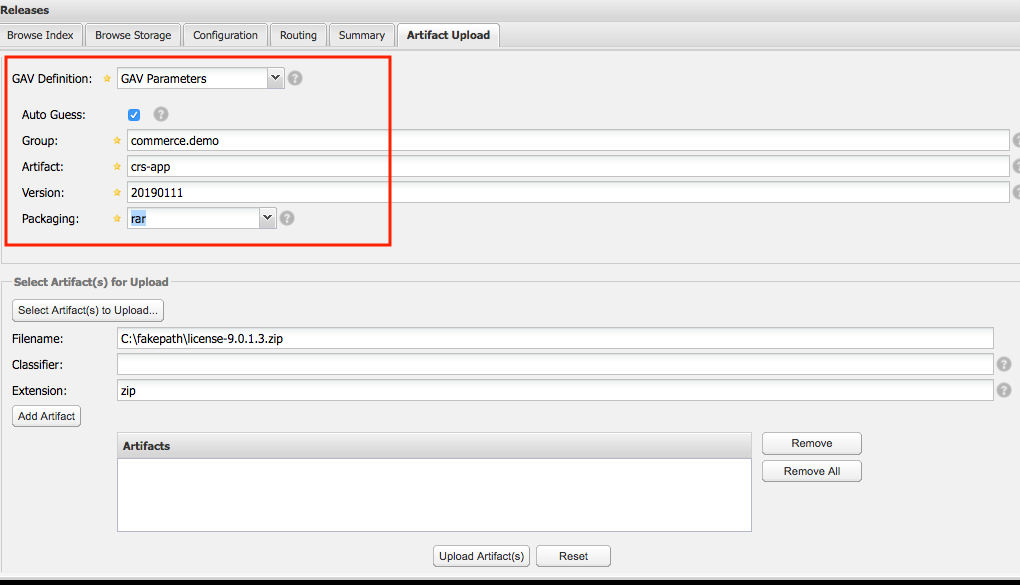


1. Input the field for assets version information like below :

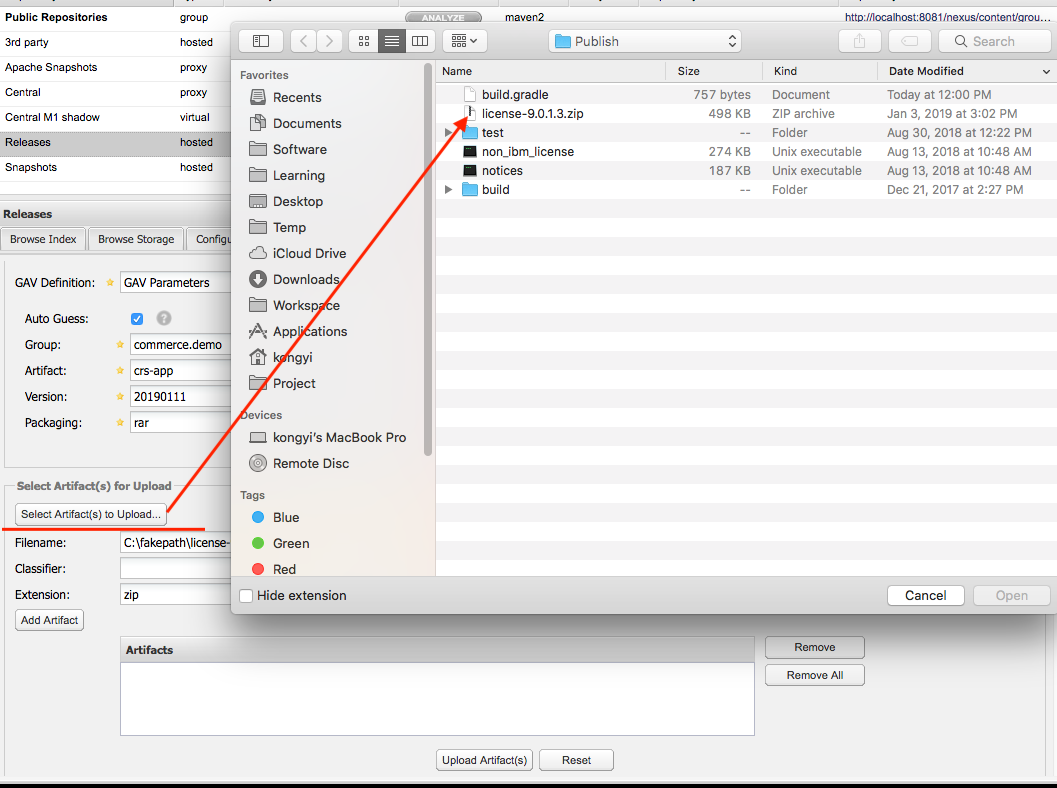
Change artifact to target component name

Change Group name to commernce.<target\_tenant\_name>

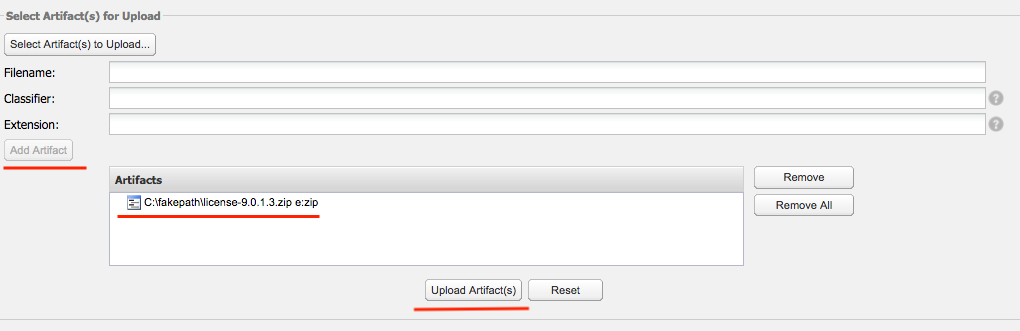
Change Version to your special version



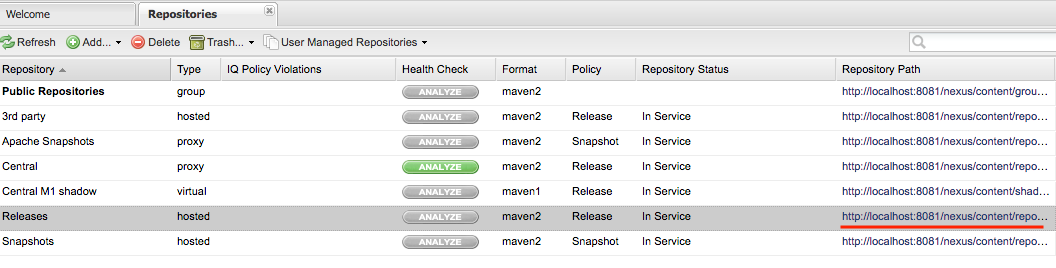
1. Select target Artifact you want to upload



1. Click “Add Artifact” Button then Click “Upload Artifact” Button to upload assets



1. Optional: You can click the release repository link and check if assets can be downloaded



When you set the Nexus server in deploy controller, we suggest you input the IP address, in

case the container in Kubernetes cluster can not parse the hostname

1. Setup the helm repository ( On ICP Master Node )

https://github.com/IBM/wc-helmchart/tree/master/WCSDevOps

The Quick way to setup Helm Charts Repository is leverage the Helm Client ( You can get help by run helm serve --help )

1. Logon Server Witch has installed Helm Client
2. Create Folder "HelmChartsRepo"
3. Upload Commerce V9 Helm Chart package "WCSV9" to ./HelmChartsRepo
4. Run Helm package command to package "WCSV9" folder

helm pacakge WCSV9

1. Run command to start Helm Serve service: helm serve --repo-path ./HelmChartsRepo --address ServerIPAddress:8879 ( replace the ServerIPAddress with real IP )

When you finish all steps you will see WCSV9-ChartVersion.tgz file and index.yaml file under ./HelmChartsRepo

you can check helm repo server by open index.yml. There should have one record like below:

apiVersion: v1

entries:

WCSV9:

- apiVersion: v1

created: 2018-08-02T17:06:31.80912053+08:00

description: Commerce Helm chart

digest: 8fac0a50d9c5831dc046b29645a98f3948aac97d8e5aecebe284830571fc4b38

name: WCSV9

urls:

- http://x.x.x.x:8879/WCSV9-0.1.x.tgz

version: 0.1.x

1. Build Docker Image for Tool Chain and upload to docker repository under “commerce” namespace

Tips:

* + If your ICP has enabled tls – when you run helm, you must add –tls, please follow this guide Please follow: https://github.com/IBM/wc-devops-utilities/tree/master/utilities/DeploySlave/BuildICPSlave
  + When you upload those tool chain to docker reposiroty, please tag them with latest tag. Because as default devops-utilities will use those docker image with latest tag~
  + When you build deployslave, if you are using ICP, you need get the helm client from ICP UI and put under the the root path of Dockerfile.

1. Deploy wc-devops-utilites on Kubernetes with helm chart ( On ICP Master Node )

<https://github.com/IBM/wc-helmchart/tree/master/WCSDevOps>

create cust-values.yaml with below content (correct the value with <>)

Vault / Consul docker image will pull from internet, if you already have them on private docker repository, you can change the image url point to address on your private repository

|  |
| --- |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| VaultConsul: |
|  |

|  |
| --- |
|  |

|  |
| --- |
| VaultImage: docker.io/vault |
|  |

|  |
| --- |
| VaultTag: 0.9.5 |
|  |

|  |
| --- |
| ConsulImage: docker.io/consul |
|  |

|  |
| --- |
| ConsulTag: 1.0.6 |
|  |

|  |
| --- |
| Enable: true |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| DeployController: |
|  |

|  |
| --- |
| Replicas: 1 |
|  |

|  |
| --- |
| Image: <Docker\_Harbor\_IP\_Address>/commerce/deploycontroller |
|  |

|  |
| --- |
| Tag: latest |
|  |

|  |
| --- |
| EnvParameters: |
|  |

|  |
| --- |
| InCluster: true |
|  |

|  |
| --- |
| VaultUrl: |
|  |

|  |
| --- |
| VaultToken: |
|  |

|  |
| --- |
| KubernetesUrl: |
|  |

|  |
| --- |
| BundleRepo: http://<Nexuse\_IP\_Address>:8081/nexus/content/repositories/releases/commerce |
|  |

|  |
| --- |
| DockerRepo: <Docker\_Harbor\_IP\_Address> |
|  |

|  |
| --- |
| DockerRepoPwd: Harbor12345 |
|  |

|  |
| --- |
| DockerRepoUser: admin |
|  |

|  |
| --- |
| HelmChartsRepo: http://<Master\_Node\_IP\_Address>:8879/charts |
|  |

Run Helm Deploy

Helm install –name=deploycontroller ./WCSDevOps -f cust-values.yaml

Access deploycontroller with url http://<IPC\_Ingress\_Prox\_IP\_Address>:31899 ( admin/admin )

1. Check if global configuration set correctly