IBM Cloud Pak Install on ICP running on top of Openshift

People worked on this task:

 Marcelo Gianini Novaes – mgnovaes@br.ibm.com

 Cassio Jose Santos Freire – cassiof@br.ibm.com

 Glauco dos Santos Reis – gsreis@br.ibm.com

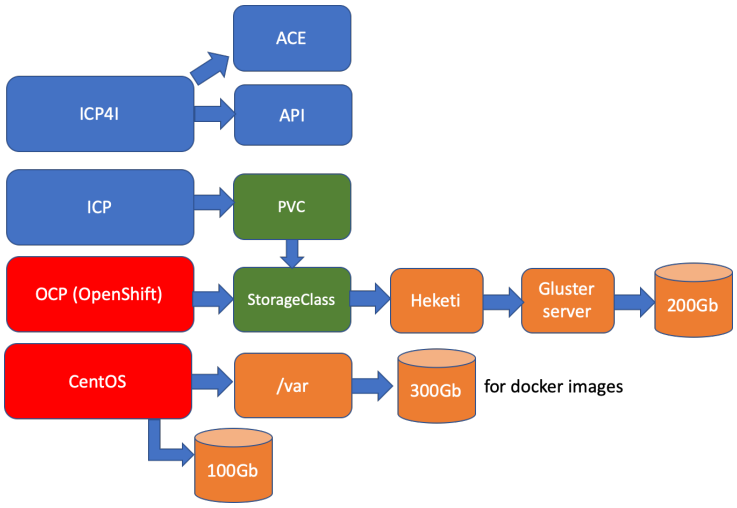
People that supported this task :

 James Hewith - james.hewitt@uk.ibm.com

 Chris Phillips - Chris.Phillips@uk.ibm.com – Should notice this guy made a lot of efforts even on his vacation! Thank you!

Introduction:

This document is a step by, ground up, of the process to install IBM Cloud PAK for Integration on top of Openshift. This were part of a collective effort to make possible a few POC on Brazilian customers moving to this new solution. The topology is shown below:



The decision was to use CentOS (less restrictive than RedHat):

- **CentOS: 7.x - Minimal Install (64 bit)**
- **32VCPUs + 128Gb**
- **100gb Disco SO + 200Gb Gluster + 300gb Docker**
- **PAK for Integration: 2019.3.1**
- **OCP: 3.11**
- **ICP: 3.2.0.1907 (Not used 1906 inside PAK 2019.3.1)**

Files needed (downloaded from IBM software downloads):

- **ibm-cloud-private-rhos-3.2.0.1907.tar.gz**
- **IBM_CLOUD_PAK_FOR_INTEGRATION_201.tar.gz**

(Suggestion is to move those files to /opt on CentOS)

Installation of OCP

Login on softlayer machine (as root):

ssh root@169.57.168.114 (change by your IP)

recommended to change root password

passwd root

Recommended to ensure that hostname resolves the machine IP address

```
hostname
ping <value_hostname>
vi /etc/hostname
vi /etc/hosts # ensure that 127.0.0.1 localhost and <hostname> <public_IP>
```

Prepare the disk for docker images

```
fdisk -l #get the 300Gb disk ID.  Example: /dev/xvde
```

Make the disk partition

```
fdisk /dev/xvde # the disk you selected before
# type n, and then choose all default options and w at the end
mkfs.ext4 /dev/xvde # format disk

# Move /var to 300Gb disk
yum install -y rsync
mkdir /mnt/newvar
mount /dev/xvde /mnt/newvar
rsync -aqxP /var/* /mnt/newvar
cd /
mv var var.old
mkdir /var
umount /dev/xvde
mount /dev/xvde /var
df
```

You should see at the end the var mapped to 300Gb

```
tmpfs           13184220    0          13184220     0%.   /run/user/0
/dev/xvde       309504832   323612     293436196    1%.   /var
```

We did a lot of experiments, and for some reason when the disk is not enough, some docker images for PAK's helm disappear. We suppose that when no space left, docker removes older or unused images. Since the PAK's image never used before, some of then that will be used in future where removed and the installation process by itself fail. We decided then to mount an entire new disk for /var. Decide what is best for your case… Anyway, make sure to watch the disk space using **df <disk_with_images_docker>**

To keep disks mounted between restart, edit file /etc/fstab put mapping there

/dev/xvde /var ext4 defaults 0 0

Docker's install

yum install -y docker
systemctl start docker
docker pull tomcat #not really needed, just to test is mount directory is mapped for this image

OpenShift's Install

yum install mlocate #not really needed
updated #not really needed
yum install telnet #not really needed

yum install git docker net-tools -y

Without these lines Openshift will not install correctly
sed -i 's/NM_CONTROLLED=no/NM_CONTROLLED=yes/' /etc/sysconfig/network-scripts/ifcfg-eth0
sed -i 's/NM_CONTROLLED=no/NM_CONTROLLED=yes/' /etc/sysconfig/network-scripts/ifcfg-eth1

cd /opt

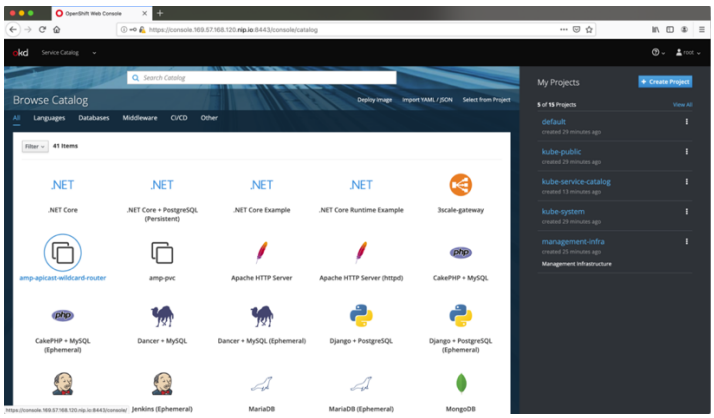
We used the script below to install Openshift. Some speriments were made using oc cluster up, but we found difficult to find some artifacts needed to start ICP. We suggest to follow this path for installation.

wget https://raw.githubusercontent.com/gshipley/installcentos/master/install-openshift.sh

chmod 777 install-openshift.sh
./install-openshift.sh

We leave all options default, except by security that we choose encryption disabled
Let's Encrypt only works if the IP is using publicly accessible IP and custom certificates.
This feature doesn't work with OpenShift CLI for now.
1) Yes
2) No
2

After all steps (that takes almost a half hour) you should have a
(1) URL: https://console.169.57.168.114.nip.io:8443
Credenciais: root/****



#validate that all pods are ok
oc get pods --all-namespaces

We have found that in extreme low memory cases (or disk) some pods do not show up. At this point, is very important to make sure the health of OpenShift is ok.

Gluster and Heketi install

The recommendation for RedHat is to use Gluster for persistent volumes. We started some tests using NFS, but we moved to Gluster when we found the RH recomendation.

yum install centos-release-gluster glusterfs-server thin-provisioning-tools -y
yum install centos-release-gluster -y
yum install glusterfs-server -y
systemctl start glusterd
ps -ef | grep glusterd
systemctl enable glusterd
gluster pool list

Now install heketi (a front end for Gluster)

cd /opt
wget https://github.com/heketi/heketi/releases/download/v9.0.0/heketi-v9.0.0.linux.amd64.tar.gz
tar -xzvf heketi-v9.0.0.linux.amd64.tar.gz
sudo chmod a+rw -R /opt/heketi
cd /opt/heketi

Seems there are differences from file versions of heteki. The file inside the bundle did not worked for us. And we provided some files that works.

To keep it simple, all needed files can be found on this link. Open the file and put on /opt
http://www.glaucoreis.com.br/yamfiles.zip

You can find files for security policies on this link also
https://github.com/IBM/cloud-pak

At the end you should have two directories /opt/yaml and /opt/cloud-pak-master

./heketi --config /opt/yaml/heketi.json
This line should start heketi server

By default port 8888 are going to used by heketi server. Make sure the port is not being used or change properly. The configuration is inside /opt/yaml/heketi.json.

For the rest of install leave this shell running and open another shell.

The recommendation is to create a service on /usr/lib/system/heketi.service, to grant that heketi starts on the Linux boot.

On the new shell prompt, check that Heketi server is available
curl http://localhost:8888/hello

It is not required, but the disk should be clear for Gluster
sudo wipefs -a -f /dev/xvdc # verificar o disco

Disk should be clear for Gluster's claim (no partitition and no format), so call wipefs to make sure this is OK.

Edit file /opt/yaml/topology.json, and make sure to update IP address and Cluster ID.

Check the sample bellow:

```
{
 "clusters": [
  {
   "nodes": [
    {
     "node": {
      "hostnames": {
       "manage": [
        "108.168.187.157"
       ],
       "storage": [
        "108.168.187.157"
       ]
      },
      "zone": 1
     },
     "devices": [
       {
          "name":"/dev/xvdc",
          "destroydata": false
       }
     ]
    }
   ]
  }
 ]
}
```

Call command line to create server

export HEKETI_CLI_SERVER=http://localhost:8888
/opt/heketi/heketi-cli topology load --json /opt/yaml/topology.json

You should see a couple of lines that the Cluster were created. Make sure you take note of ID, because you need it a few steps ahead

```
Creating cluster ... ID: 5fb774846110d6fe81cf17cc66c0ebcf
    Allowing file volumes on cluster.
    Allowing block volumes on cluster.
```

Create namespace for ICP
oc create namespace icp

Edit and change IP and ID you collected before
nano /opt/yaml/glustersc.yaml

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
 name: glustersc
 namespace: icp
provisioner: kubernetes.io/glusterfs
parameters:
 resturl: "http://108.168.187.157:8888"
 clusterid: "5fb774846110d6fe81cf17cc66c0ebcf "
```

```
restauthenabled: "false"
volumetype: none
```

Create storageclass with this yaml file

oc create -f /opt/heketi/yamls/glustersc.yaml

This step is not needed, but is recommended that you try to create some PVC to make sure it is being created
oc create -f /opt/heketi/yamls/my-pvc.yaml

At this point the PVC should be bounded to storageclass
oc get pvc -n icp

```
NAME    STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
my-pvc  Bound    pvc-23bd5724-c1de-11e9-8594-0616a651671b   50Gi       RWX            glustersc      10s
```

After this validation remove the provision an leave it available for ICP install

oc delete -f /opt/yaml/my-pvc.yaml -n icp
oc set volume deploymentconfigs/docker-registry --add --name=registry-storage -t pvc --claim-name=my-pvc --overwrite

Apply security roles

Create namespace
kubectl create namespace integration

Change role binfing
kubectl -n integration create rolebinding pod-security-policy-rolebinding --clusterrole=pod-security-policy-clusterrole --group=system:serviceaccounts:integration

Apply those polices (you already downloaded to /opt in previous steps)
cd /opt/cloud-pak-master/spec/security/scc
oc apply -f .
cd /opt/cloud-pak-master/spec/security/psp/
oc apply -f .

After some research, we find that theses polices appear to be applied, and we recommend to run those against OCP.

Install ICP

Load ICP images to docker (remember you downloaded it from software downloads)
tar -xf ibm-cloud-private-rhos-3.2.0.1907.tar.gz -O | sudo docker load

Open PAK install on /opt directory
cd / opt
tar -xf IBM_CLOUD_PAK_FOR_INTEGRATION_201.tar.gz

You should have a directory /opt/installer_files after this step

cd /opt/installer_files
sudo cp /etc/origin/master/admin.kubeconfig cluster/kubeconfig
#This file should be available as part of OCP install

Copy the file generated by ssh-keygen

We supposed this key were generated by OCP install, but anyway a valid test is type ssh root@<IP> and verify if ssh runs without ask for a password. If not works, get back and check previous steps.

cp /root/.ssh/id_rsa cluster/ssh_key
cd cluster

At this point you have these file updated :

```
├── config.yaml
├── hosts
├── kubeconfig
├── logs
└── ssh_key
```

Edit file hosts and place IP address on master, worker and proxy

```
[master]
169.57.168.117 #your IP

[worker]
169.57.168.117

[proxy]
169.57.168.117
```

Not sure if this steps is needed, but I have impression that on some of my previous attempts and error arises caused by this file (gsreis).

So, edit the file config.yaml

The fields **master, proxy e management receive the value of**
oc get node

```
cluster_nodes:
  master:
    - gsreis-icp.ibm.cloud
  proxy:
    - gsreis-icp.ibm.cloud
  management:
    - gsreis-icp.ibm.cloud
```

Field storageclass receive value of

oc get storageclass

```
storage_class: glustersc
```

console->host receives the host of OCP console – see (1)
console->port receives the port of OCP console – see (1)

```
  console:
    host: console.169.57.168.117.nip.io
    port: 8443
```

Insert this lines

```
default_admin_password: <password>
password_rules:
- '(.*)'
```

If ICP are going to be installed on same node of OCP master, change ingress IP

```
ingress_http_port: 3080
ingress_https_port: 3443
```

router->clusterhost receives icp-console.console.<your domain>
router->proxy-host receives icp-proxy.<your domain>

```
router:
  cluster_host: icp-console.console.169.57.168.117.nip.io
  proxy_host: icp-proxy.console.169.57.168.117.nip.io
```

 For sake of simplicity, we decided to use nip.io or xip.io as DNS. We did not try a real DNS for this experiments.

On tls->hostname, put the same value of proxy_host

```
    values:
      image:
        pullSecret: sa-integration
      tls:
        hostname: icp-proxy.console.169.57.168.117.nip.io # hostname of the ingress proxy to be configured
```

 I've tried same value of proxy or distinct values (gsreis) for this field and no problem on install anyway.

Apply this security policies

oc adm policy add-scc-to-group ibm-restricted-scc system:serviceaccounts:integration
oc adm policy add-scc-to-user ibm-restricted-scc system:serviceaccounts:integration

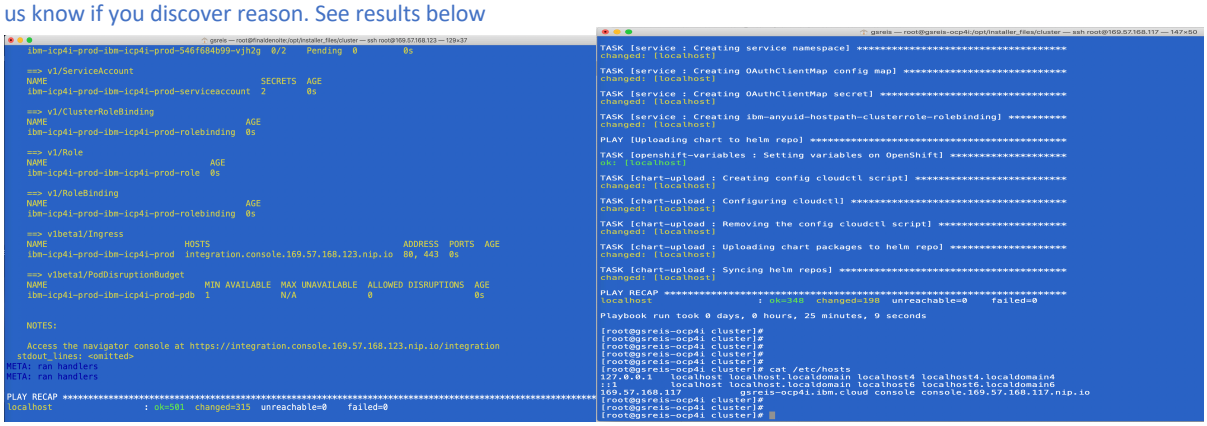
 Not sure if we need these policies. Keep them if you don't want to make additional experiments.

Run the installer

sudo docker run -t --net=host -e LICENSE=accept -v $(pwd):/installer/cluster:z -v /var/run:/var/run:z --security-opt label:disable ibmcom/icp-inception-amd64:3.2.0.1907-rhel-ee install-with-openshift

 Watch we are using the 1907 and not 1906 inside the original PAK. Seems that booth installs, but we prefer to use the newest one.

At this point, we found two results, one that seems to load all charts and another that finish fine as well, but with no packs loaded. We could not figure out the reasons to have one or another result, beside to make dozen of tests. Please let us know if you discover reason. See results below



The following steps will work for booth scenarios

Run post install comands

kubectl --kubeconfig /etc/origin/master/admin.kubeconfig get nodes

kubectl --kubeconfig /etc/origin/master/admin.kubeconfig patch scc icp-scc -p '{"allowPrivilegedContainer": true}'

kubectl --kubeconfig /etc/origin/master/admin.kubeconfig get scc icp-scc

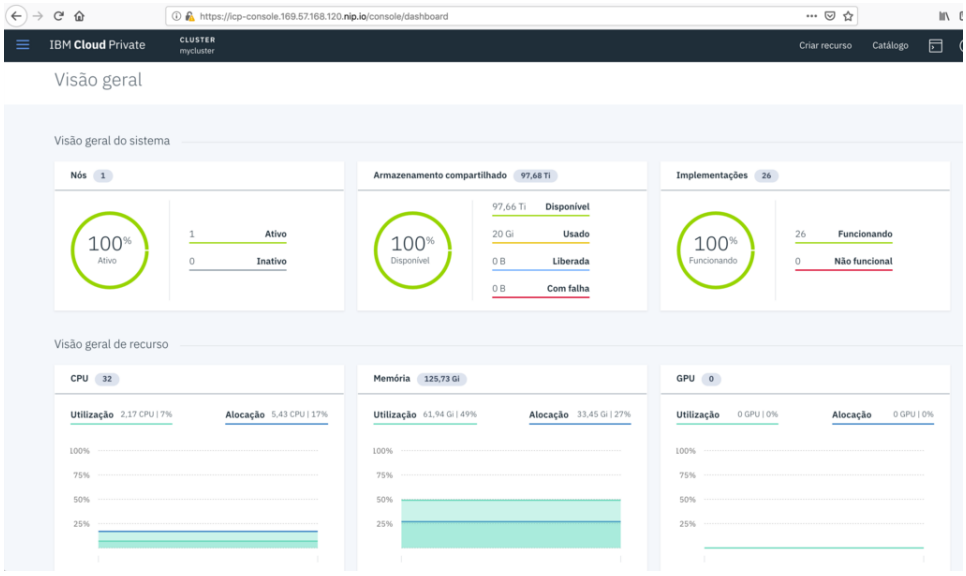
At this point you should have OCP and ICP console installed. Try to open console

cluster_host: icp-console.***169.57.168.120.nip.io***

https://icp-console.***169.57.168.120.nip.io***/console/welcome

user: admin

Password: password



At this point we strongly recommend you check if docker disk space, pods and memory are ok. An additional check is if route is properly configured. You should follow steps just if everything is OK.

docker system df

oc get pods –all-namespaces

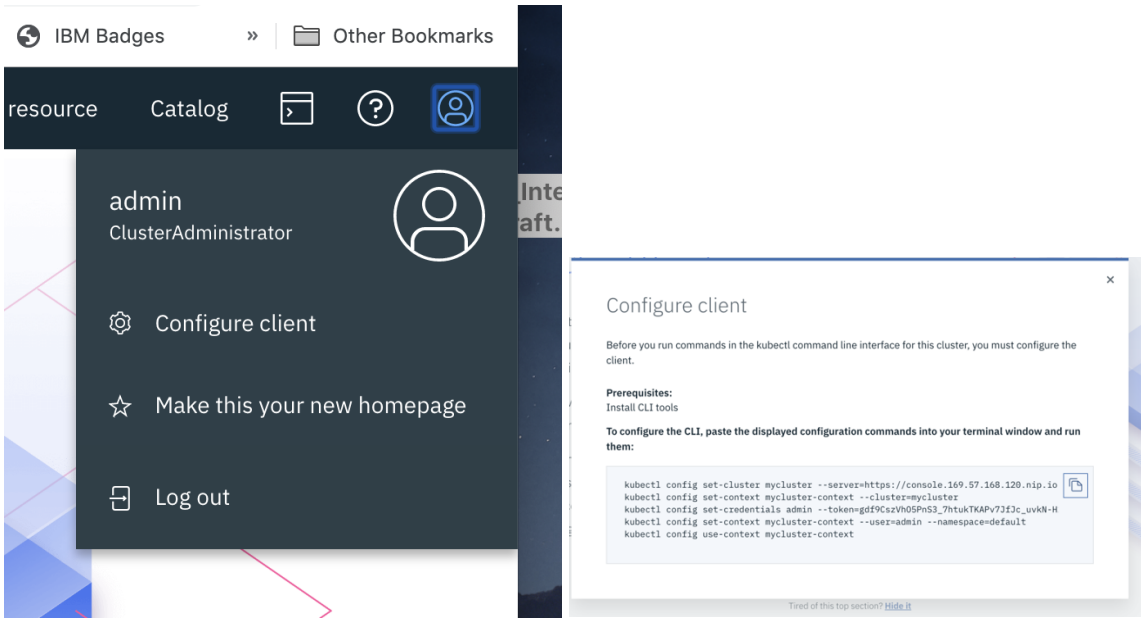
oc get route icp-proxy -n kube-system -o yaml

Install PAK

Create namespace for install

kubectl create namespace integration

Run configure client from ICP console



Paste these lines on shell and execute them

Download and install cloudctl

curl -kLo cloudctl-linux-amd64-v3.2.0-634 https://icp-console.<your_IP>.nip.io:443/api/cli/cloudctl-linux-amd64
sudo mv cloudctl-linux-amd64-v3.2.0-634 /bin/cloudctl
sudo chmod 777 /bin/cloudctl

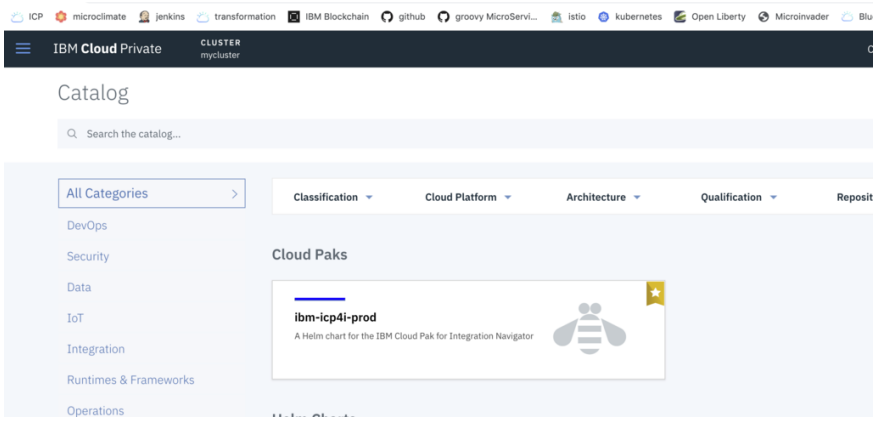
Log using CloudCtl and docker

cloudctl login -a https://icp-console.169.57.168.120.nip.io --skip-ssl-validation
docker login -u admin -p $(oc whoami -t) docker-registry.default.svc:5000

Not pretty sure if cloudctl or docker login should be called first each other (gsreis). If you load images and there is a problem to create a instance of helm, perhaps you should back to this point and change order.

cloudctl catalog load-archive --archive IBM-Cloud-Pak-for-Integration-2.0.0.tgz --registry docker-registry.default.svc:5000/integration

After finishing the process, you should have the ICP4I console available for install.



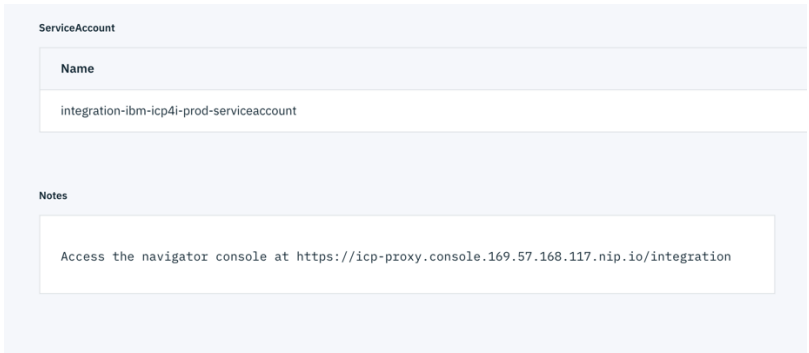
If the chart does not appear, get back and try cloudctl login and docker login again (perhaps change order).

Repeat the load process for other files on icp4icontent

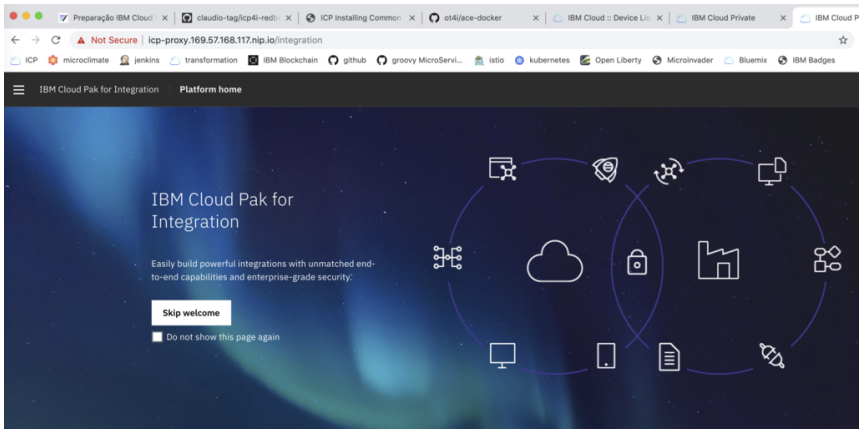
IBM-API-Connect-Enterprise-for-IBM-Cloud-Pak-for-Integration-1.0.1.tgz
IBM-App-Connect-Enterprise-for-IBM-Cloud-Pak-for-Integration-2.0.0.tgz
IBM-Aspera-High-Speed-Transfer-Server-for-IBM-Cloud-Pak-for-Integration-1.2.1.tgz
IBM-Cloud-Pak-for-Integration-2.0.0.tgz #already made
IBM-Cloud-Pak-for-Integration-Asset-Repository-2.0.0.tgz
IBM-DataPower-Virtual-Edition-for-IBM-Cloud-Pak-for-Integration-1.0.3.tgz
IBM-Event-Streams-for-IBM-Cloud-Pak-for-Integration-1.3.1-for-OpenShift.tgz
IBM-MQ-Advanced-for-IBM-Cloud-Pak-for-Integration-3.0.0.tgz

We strongly suggest to leave a shell open with watch docker system df and another with watch oc get pods –all-namespaces. Make sure the installation is healthy when upload files

Select helm ibm-icp4i-prod and configure it.
Select a name, a namespace with right policy (see below) and cluster.
In the field "**Hostname of the ingress proxy to be configured**" put the proxy-host of config.yaml file.
Submit and wait for provisioning.
After this, go to **hamburguer menu->helm releases** and select the name of instance you choose before. Go to very end of helm description and you find the URL of Integration console



Now pick up console URL and open on a new browser



Now you are ready to provision all other charts from Integration console

Links

Several links and information were used for this install

https://www.ibm.com/support/knowledgecenter/en/SSBS6K_3.2.0/supported_environments/openshift/install_openshift.html

https://docs.openshift.com/enterprise/3.1/install_config/persistent_storage/persistent_storage_nfs.html
https://medium.com/faun/openshift-dynamic-nfs-persistent-volume-using-nfs-client-provisioner-fcbb8c9344e

https://docs.openshift.com/container-platform/3.4/install_config/persistent_storage/dynamically_provisioning_pvs.html#change-default-storage-class

https://docs.openshift.com/container-platform/3.4/install_config/storage_examples/storage_classes_dynamic_provisioning.html#install-config-storage-examples-storage-classes-dynamic-provisioning

https://www.itzgeek.com/how-tos/linux/centos-how-tos/install-and-configure-glusterfs-on-centos-7-rhel-7.html

https://www.itzgeek.com/how-tos/linux/centos-how-tos/install-and-configure-glusterfs-on-centos-7-rhel-7.html/2

https://www.tecmint.com/add-new-disk-to-an-existing-linux/

https://www.linuxtechi.com/setup-glusterfs-storage-on-centos-7-rhel-7/

https://medium.com/@wilson.wilson/install-heketi-and-glusterfs-with-openshift-to-allow-dynamic-persistent-volume-management-89156340b2bd

https://docs.openshift.com/container-platform/3.7/install_config/storage_examples/containerized_heketi_with_dedicated_gluster.html

https://docs.openshift.com/container-platform/3.5/install_config/storage_examples/gluster_example.html

https://dzone.com/articles/debugging-kubernetes-common-errors-when-using-glus

https://www.scaleway.com/en/docs/how-to-configure-storage-with-glusterfs-on-ubuntu/

https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.3/html/container-native_storage_for_openshift_container_platform/chap-documentation-red_hat_gluster_storage_container_native_with_openshift_platform-heketi_cli

https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.1/html/container-native_storage_for_openshift_container_platform_3.4/chap-documentation-red_hat_gluster_storage_container_native_with_openshift_platform-cleaning_heketi_topology

https://docs.gluster.org/en/latest/CLI-Reference/cli-main/

https://www.server-world.info/en/note?os=CentOS_7&p=openshift311&f=1

https://github.com/IBM/cloud-pak

https://docs.openshift.com/container-platform/3.7/install_config/storage_examples/containerized_heketi_with_dedicated_gluster.html