**Private docker registry helm chart**

Create namespace:

kubectl create ns registry

Create a path for storing the authentication file:

mkdir ~/docker-registry/auth

Create a path as data for the image details:

mkdir ~/docker-registry/data

Generating password:

sudo apt install apache2-utils -y

cd ~/docker-registry/auth

Change the username to the username you want to login:

htpasswd -Bc registry.password username

Helm chart deployment :

helm install docker-registry . -n registry

**Push and pull images to the private docker regsitry**

On the main server, log in with the username and password you set up previously:

docker login http://domain-ip:30005

example: docker login 10.11.100.191:30005

Tag your existing image with the domain name as following

Example: docker pull nginx:latest

docker tag nginx:latest 10.11.100.191:30005/my-nginx:latest

Now your image tag will be changes and now try to push it to the private docker registry

docker push 10.11.100.191:30005/my-nginx:latest

Image will be pushed successfully.

Now for testing delete the image 10.11.100.191:30005/my-nginx:latest from your local repository and to pull it from private docker registry by following commands

docker pull 10.11.100.191:30005/my-nginx:latest

Now that you’ve tested pushing and pulling images, you’ve finished setting up a secure registry that you can use to store custom images.

Since we are not enabled secured (https) connection we need to mention the ip of our main server as insecure registry in the each node where we need to work with our private docker registry. Following are the steps to update it.

**Creating Insecure Registry:**

cd ~/etc/docker

Create a daemon.json file if it doesnot exist.

Add the below content into it.

        {

              “insecure-registries” : [“domain-ip:port”]

         }

Example:

  {

               “insecure-registries” : [“10.11.100.191:30005”]

          }

Save and exit the file .

Restart the docker for applying the changes.

**systemctl daemon-reload**

**systemctl restart docker**