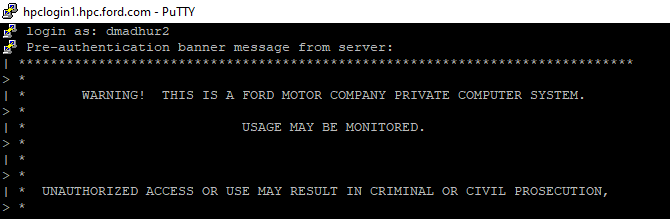
# Building Docker in Kubernetes Cluster and deploying into Mach1ML

Instructions:

1. Login to HPC account
2. Generate Github SSH key.
3. Add Github & Portus ssh keys into Kubernetes Secrets.
4. Create Docker-Enabled POD in Kubernetes Cluster
5. Build & Push Docker Image
   1. Clone git repo and create docker image
   2. Push the docker image to Docker Image Repository
6. Deployment in Mach1Ml

## Step 1: Login to HPC account:

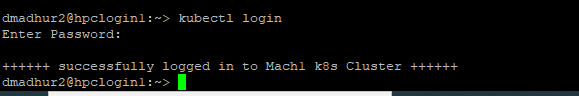
1. Login into HPC via Putty:



1. Login to Kubernetes

Command: kubectl login

Enter your hpc password



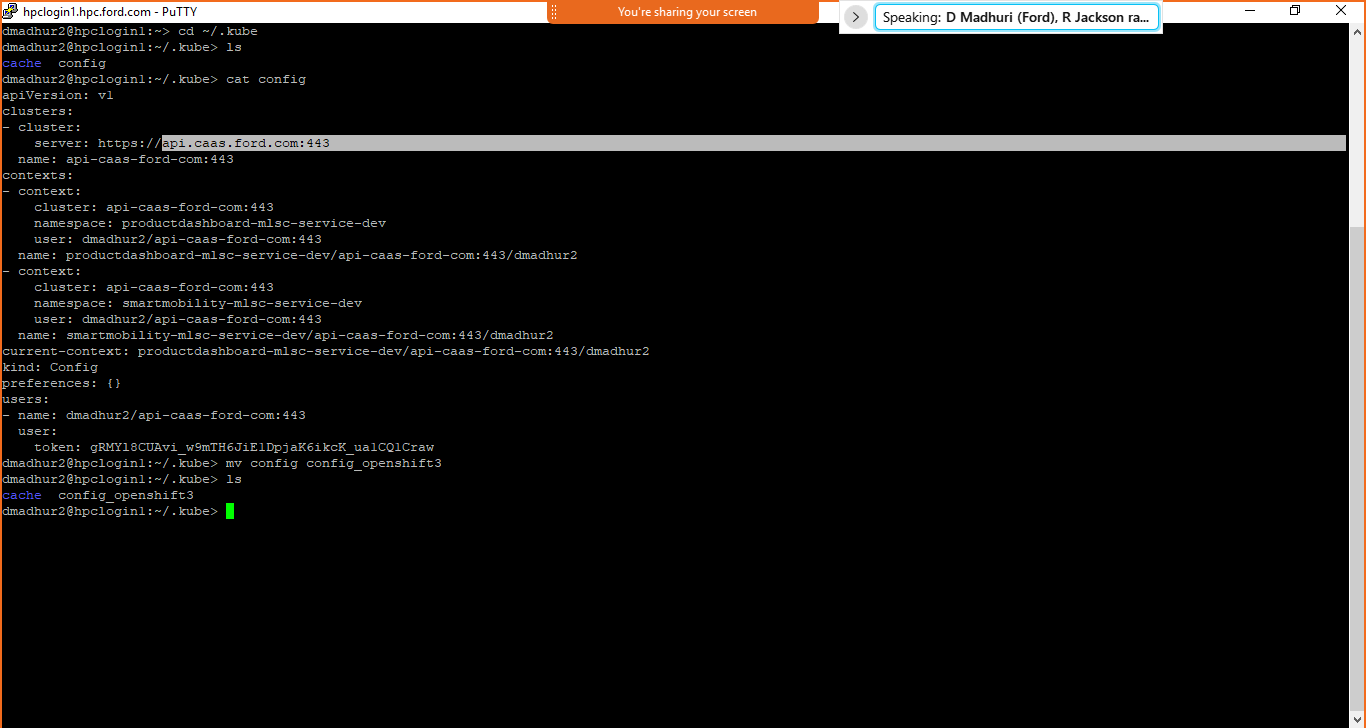
1. If you face any issues after login, then it might be kubernetes configuration is not pointing to Mach1 cluster. We need to recreate the configuration.

**Command**: cd ~/.kube

cat config

In the below example, the **config** file under **.kube** folder is pointing to “api-caas-ford.com” instead of Mach1 cluster.  
  
To recreate the config file:  
Command: mv config config\_openshift3

Here, we are renaming the config file to config\_openshift3. (You can rename the file to any name)



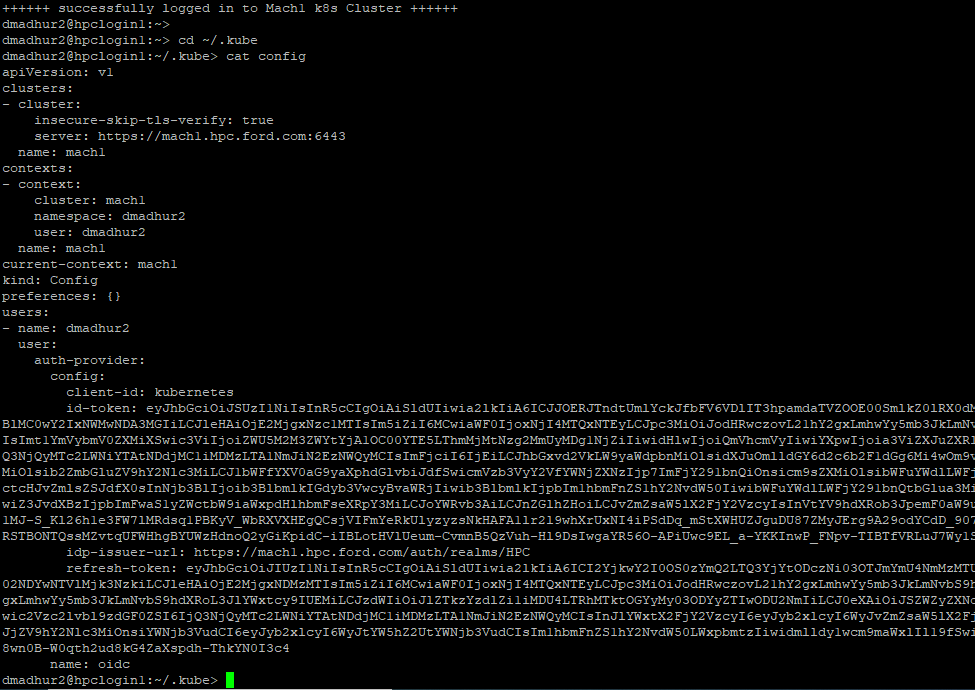
1. Exit hpc and login again.  
   Login to Kubernetes:  
   Command: kubectl login

Enter your hpc password  
  
Check the config file if its pointing to Mach1 cluster now.  
Command:

cd ~/.kube

cat config

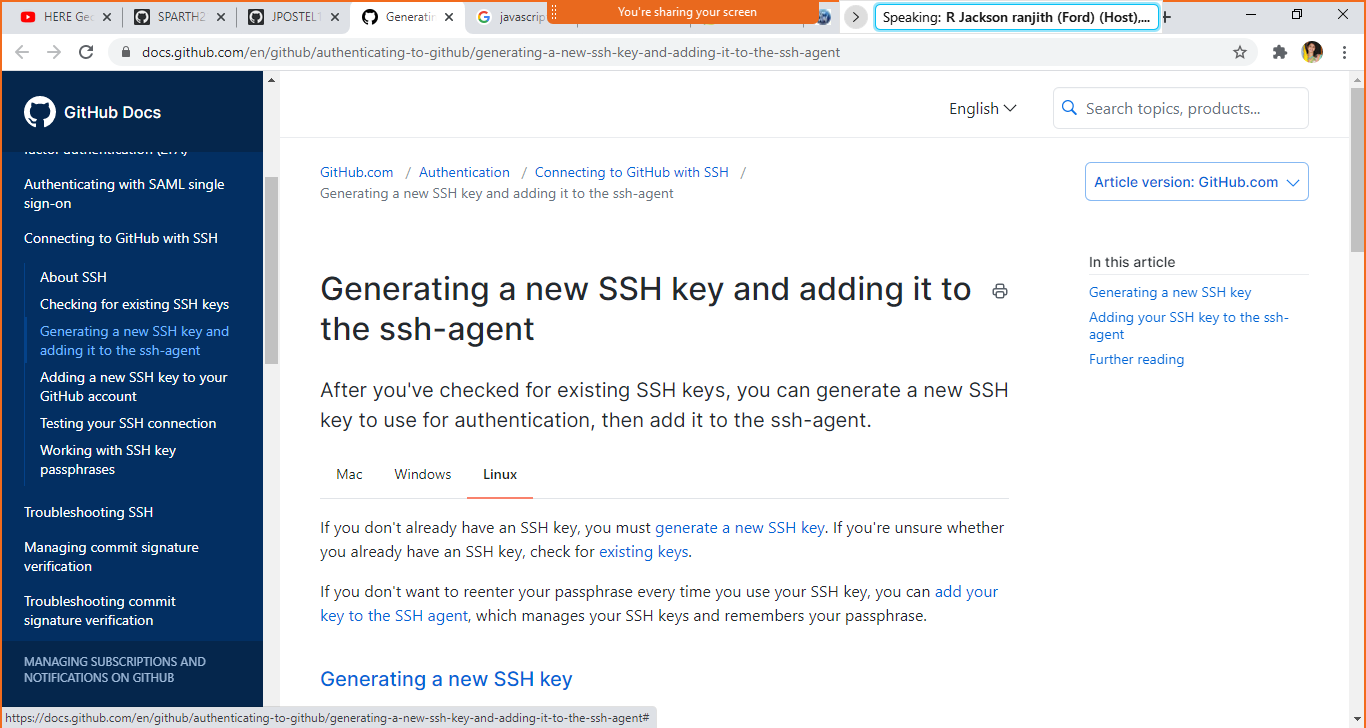
You can see from the below screenshot, its pointing to Mach1 cluster.



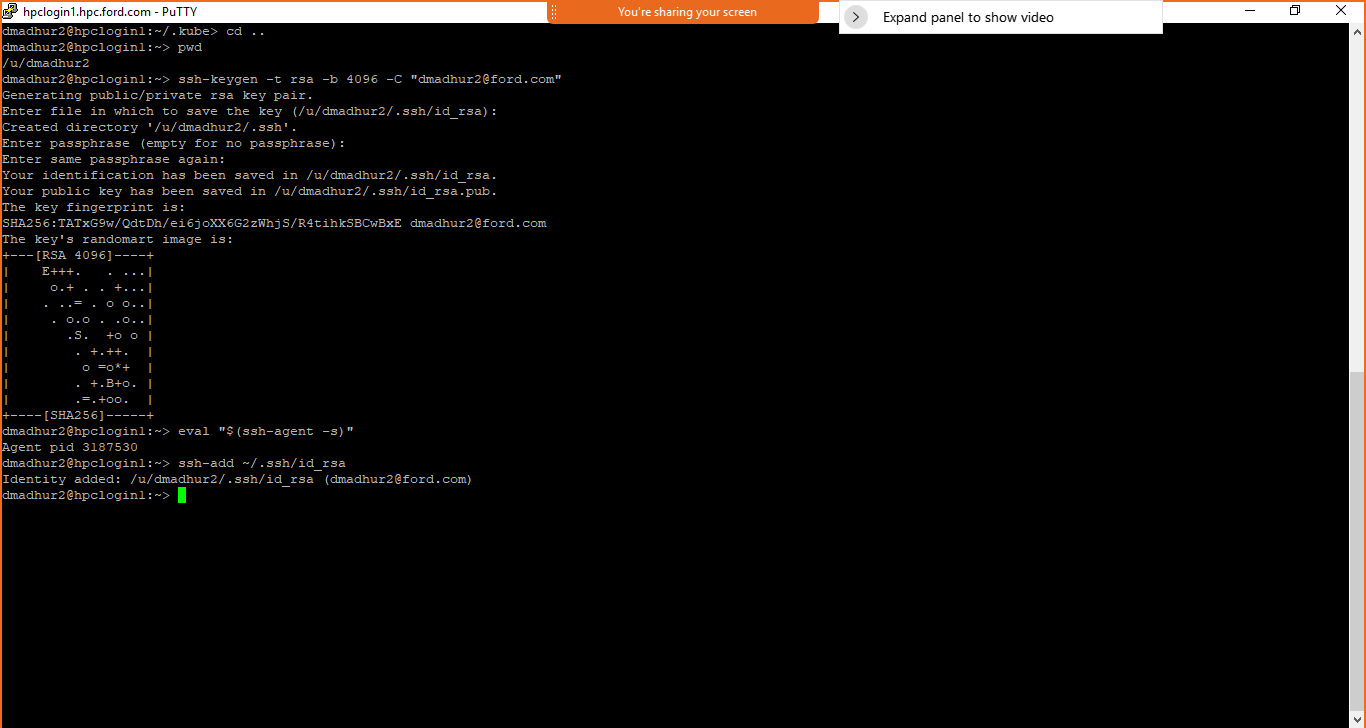
## Step 2: Generate and Add Github SSH key:

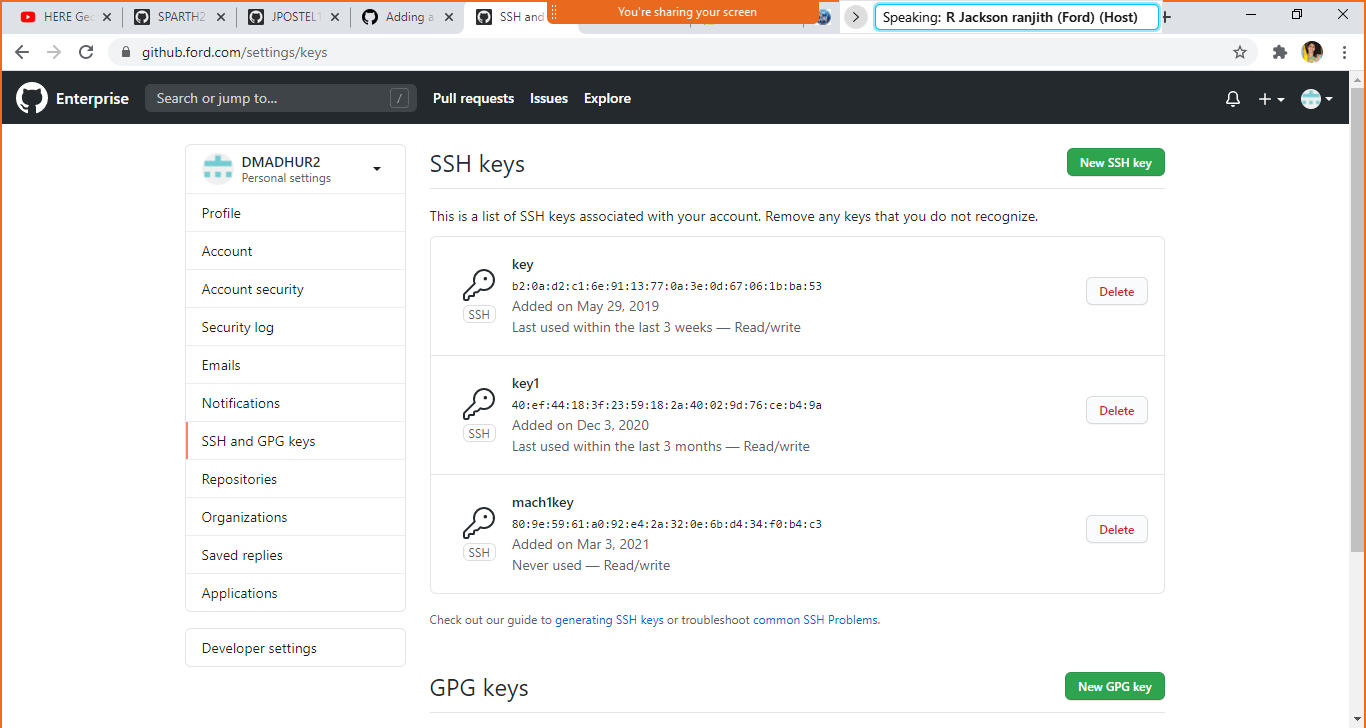
1. Create Github ssh key in HPC   
   Follow the instructions from the below link to generate ssh key and add the key to ssh-agent  
   <https://docs.github.com/en/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

Use the Linux OS commands  
*Note: Don’t give the passphrase while generating the key*



1. Adding ssh key to ssh-agent



1. Add the ssh key in Github settings  
   Go to settings under your profile. Click SSH and GPG keys.  
   Add the new HPC SSH Key and save.  
     
   

## Step 3: Add Github & Portus ssh keys into Kubernetes Secrets:

1. Create secret keys for Github and Portus

**Portus Secret**   
**Command**:  
kubectl create secret docker-registry portus-registry-credentials --docker-server=hpcregistry.hpc.ford.com --docker-username=your\_cdsid --docker-password=your\_hpc\_password

For Quay: download secret(service.yml) - ignore this step if you are able to create portus secret.

kubectl create -f service.yml --namespace=mobility-analytics

**Issue**: If you are not able to create or access secrets and facing below error

Error from server (Forbidden): error when creating "bdamayan-secret.yml": secrets is forbidden: User "bdamayan" cannot create resource "secrets" in API group "" in the namespace "mobility-analytics"

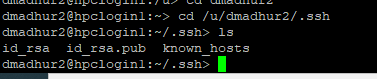
You need to raise a ITConnect ticket for docker access using below link

<https://azureford.sharepoint.com/sites/hadoopservice/deeplearning/Pages/Request-Docker-Access.aspx>

mention: I have access to hpc , need access to Kubenetes cluster and the error under description for Itconnect ticket.

**Github SSH secrets**

1. Go to your U drive in hpc - cd /u/dmadhur2/.ssh



1. List the files - ls
2. Check whether you have git public and private keys (Keys generated from Step 2 from this document.
3. Execute

Command:  
kubectl create secret generic ssh-keys --from-file=id\_rsa=$(readlink -f /u/dmadhur2/.ssh/id\_rsa) --from-file=id\_rsa.pub=$(readlink -f /u/dmadhur2/.ssh/id\_rsa.pub)

## Step4: Create Docker-Enabled pod:

Download the above file (remote-docker.yaml) and move this file to your hpc folder

Run the below command to spin the pod. This pod will contain docker-cli image.  
  
kubectl apply -f remote-docker.yaml  
  
To view the pods, use the below command

kubectl get pods

From the above screenshot, you can see the remote-docker pod is running  
Note: It may take a minute to spin the pod  
  
Get into the interactive shell of the pod  
  
Run the below command  
kubectl exec -it remote-docker bash

You will see the prompt as below  
root@remore-docker:/#

## Step 5: Build & Push Docker Image:

### Step 5.a: Clone git repo and create docker image

Clone your code from github using below command  
git clone [git@github.ford.com:xxxxx/xxxx.git](mailto:git@github.ford.com:xxxxx/xxxx.git)

git clone --single-branch --branch mobility-analytics-portal-adfs [git@github.ford.com:mobility-analytics/mobility-analytics-portal.git](mailto:git@github.ford.com:mobility-analytics/mobility-analytics-portal.git)

You can clone any or your github repo

After cloning go to your project folder



Next you can build docker image using the Dockerfile

**Command**: docker build -t dockerimagename .

After building docker image check the image using below command

**Command** : docker images

You can now run the docker command to build the docker image.

### Step 5.b: Push the docker image to Docker Image Repository

You can either push the docker image to Quay(registry.ford.com) or Portus(hpcregistry.hpc.ford.com)

Command to login to docker image repository

**HPC(Portus) :**

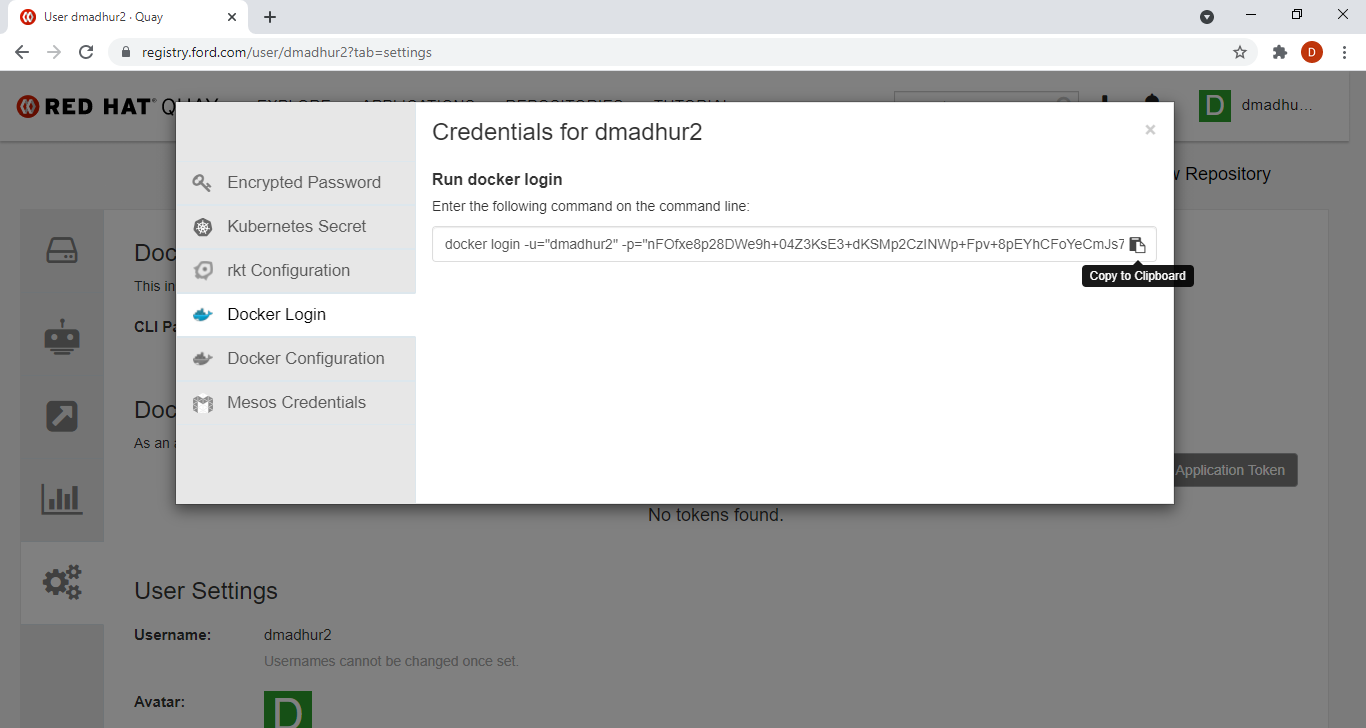
docker login hpcregistry.hpc.ford.com

**Quay:**

Login with your credentials and push the image.  
  
For Quay, please go through the below link  
<https://registry.ford.com/tutorial/>

docker login command copied from quay

Login to registry.ford.com. Go to account settings. Click "encrypted password". Enter the password and submit



**Tag the docker file**

docker tag dockerimagename:latest registry.ford.com/mobility-analytics/dockerimagename:latest

docker tag face-recognition:latest [hpcregistry.hpc.ford.com](https://hpcregistry.hpc.ford.com/rjack296/abc:latest)/dmadhur2/face-recognition:latest

**Push the docker file to registry.ford.com**

docker push registry.ford.com/ mobility-analytics /dockerimagename:latest

docker push [hpcregistry.hpc.ford.com](https://hpcregistry.hpc.ford.com/rjack296/abc:latest)/dmadhur2/face-recognition:latest

Check your image in Quay

## Step 6: Deployment in Mach1Ml

All projects deployment scrips are available under below git repo

<https://github.ford.com/mobility-analytics/deployment>

1. This deployment is done with a [deployment](https://kubernetes.io/docs/concepts/workloads/controllers/deployment/) workload in kubernetes, a [Service](https://kubernetes.io/docs/concepts/services-networking/service/), and a [VirtualService](https://istio.io/latest/docs/reference/config/networking/virtual-service/) to connect to the application running in the cluster.
2. In the [deployment.yaml](https://github.ford.com/SPARTH21/gdia-k8s-hands-on/blob/master/deployment/deployment.yaml) file we specify the docker image to be used, and the command to be run inside the docker image for starting the application.
3. Now modify the [service](https://github.ford.com/SPARTH21/gdia-k8s-hands-on/blob/master/deployment/service.yaml) file to expose the running application inside the pod as a service.
4. Finally, modify the [routes.yaml](https://github.ford.com/SPARTH21/gdia-k8s-hands-on/blob/master/deployment/route.yaml) or virtualservice.yml file to discover the service and connect to the same from a web browser.
5. Now for deploying the application, navigate to the deployment directory and execute the following in HPC:

kubectl apply -f deployment.yaml

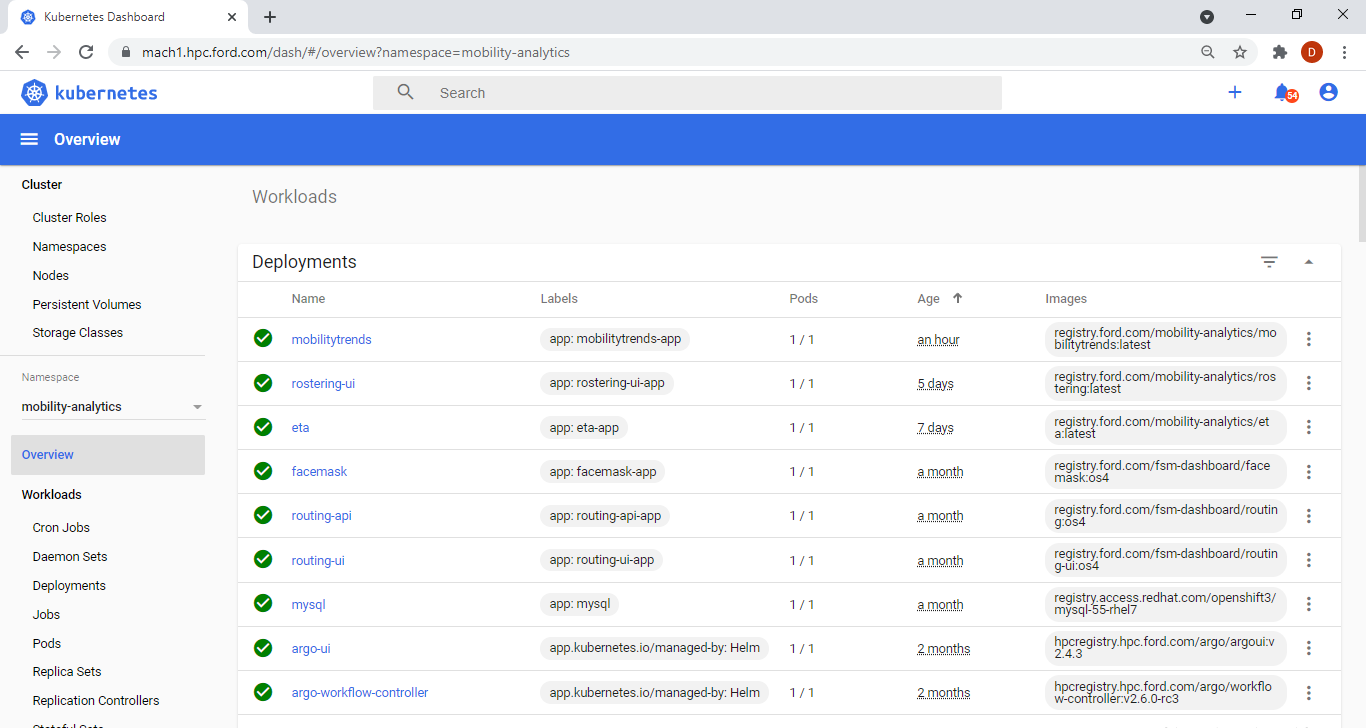
kubectl apply -f service.yaml

kubectl apply -f routes.yaml

mach1ml mobility-analytics namespace dashboard

<https://mach1.hpc.ford.com/dash/#/overview?namespace=mobility-analytics>

check your deployed app name as below.



**Ref:**  
<https://github.ford.com/SPARTH21/gdia-k8s-hands-on>