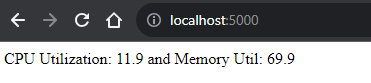
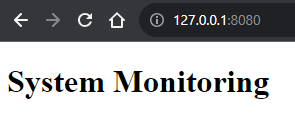
1. Working monitoring python app





# Run the container, port 8080 on the host



1. html for the monitoring app (templates folder)
2. Create Dockerfile to containerize the app and create the docker image

- docker build –t flask-app

- docker run –p 5000:5000 XXXXXXX = run the image = start a container from this image

1. https://boto3.amazonaws.com/v1/documentation/api/latest/index.html === boto3 documentation

# use boto3 to create/config/manage AWS services

We can either do it in AWS or use the python boto3 module (ecr.py)

create the ecr repo

1. Install aws cli = **pip3 install awscli**

aws configure = access key + secret + region

view push commands in the repo in AWS

# paste the commands in VSC

-- aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 921082554149.dkr.ecr.us-east-1.amazonaws.com

-- docker build -t martin\_abrashev\_repo .

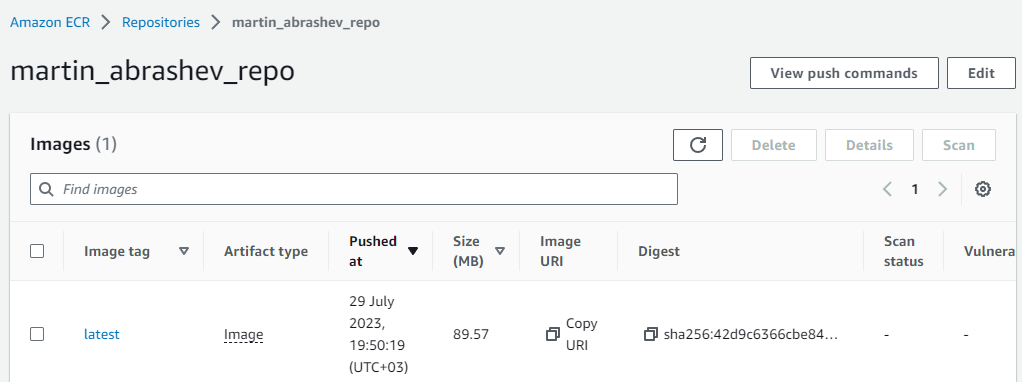
#build the docker image

-- docker tag martin\_abrashev\_repo:latest 921082554149.dkr.ecr.us-east-1.amazonaws.com/martin\_abrashev\_repo:latest

# tag the image so we can push it to the repo

-- docker push 921082554149.dkr.ecr.us-east-1.amazonaws.com/martin\_abrashev\_repo:latest

#push the image from local machine to aws ecr

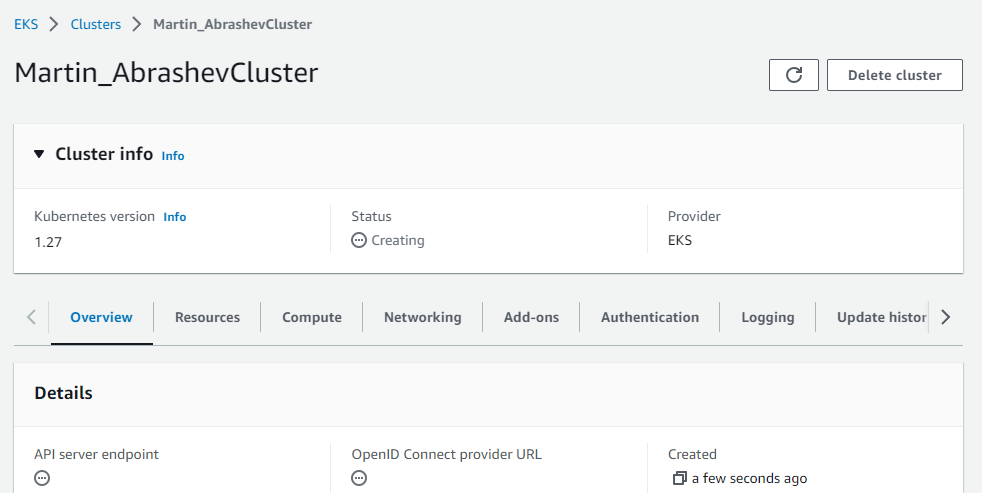


1. Go to EKS – clusters – create EKS cluster

\*the security group should have proper ports (5000 open in our case / 8080 /8000 )

- where the cluster is created we have to connect it with aws configure command

- create Node IAM role /must/



\* k8s will be using the deployment to maintain the state

- we need to have a service to expose our application to the outside of the cluster (so they can access it)

7) NEXT: create deployment + service using python(eks.py)

more kubectl

in the terminal = kubectl apply –f deployment.yaml

kubectl get deployment –n default (check what we have in the default namespace)

kubectl get svc –n default = check for services

kubectl get pods –n default = get the pods

kubectl port-forward svc/flask-service 5000:5000 = we want the app running on 5000 to be running on 5000 on our machine