**Installations on Linux Machine:**

\* I have used EC2 (Ubuntu 24.04 ) in my AWS account.

\* Install docker

<https://www.cherryservers.com/blog/install-docker-ubuntu>

\* clone the repository

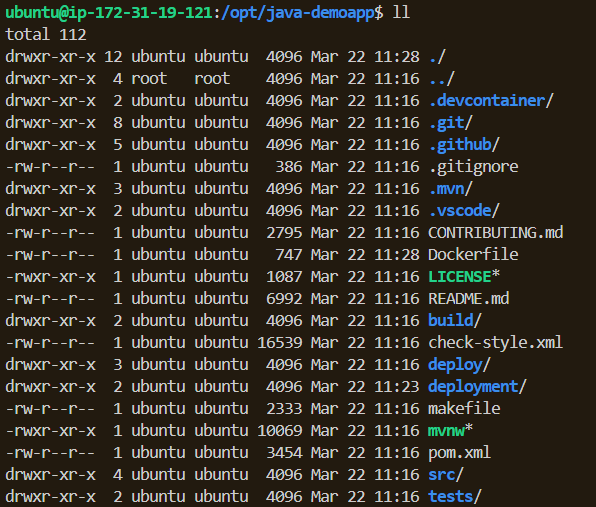
git clone https://github.com/benc-uk/java-demoapp/

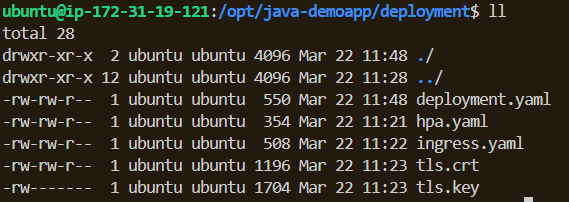
\* create “deployment” folder in this path.

sudo mkdir /opt/java-demoapp/deployment

\* copy Dockerfile to /opt/java-demoapp path

cp /opt/java-demoapp/build/Dockerfile /opt/java-demoapp/Dockerfile





1. **Minikube Setup:**

<https://minikube.sigs.k8s.io/docs/start/?arch=%2Flinux%2Fx86-64%2Fstable%2Fbinary+download>

curl -LO <https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64>

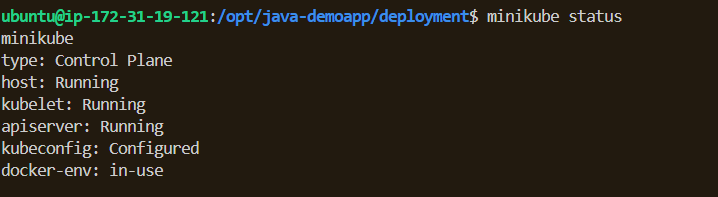
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64

minikube start

kubectl get po -A

minikube kubectl -- get po -A

alias kubectl="minikube kubectl --"



1. **Application Deployment:**

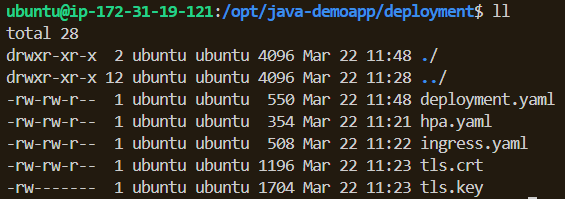
\* Create yaml files in/opt/java-demoapp/deployment . You can see at the repository.

<https://github.com/trkgul/java-demoapp/>

deployment.yaml

hpa.yaml

ingress.yaml



\* Create self-signed SSL certificate for HTTPS.

openssl req -x509 -nodes -days 365 -newkey rsa:2048 \

-keyout tls.key -out tls.crt \

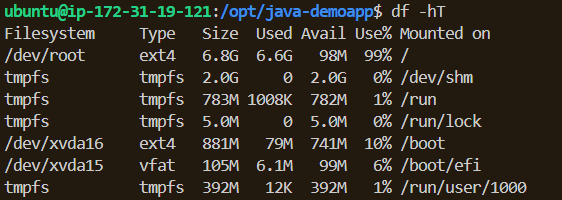
-subj "/CN=local.java-demo.com/O=java-demo"

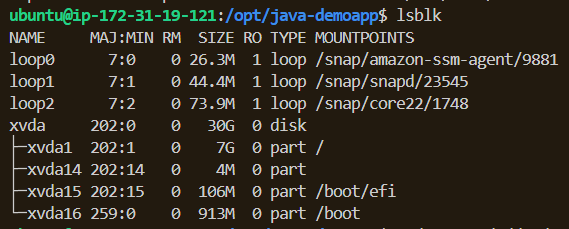
\* Create docker image according to Dockerfile in path /opt/java-demoapp/

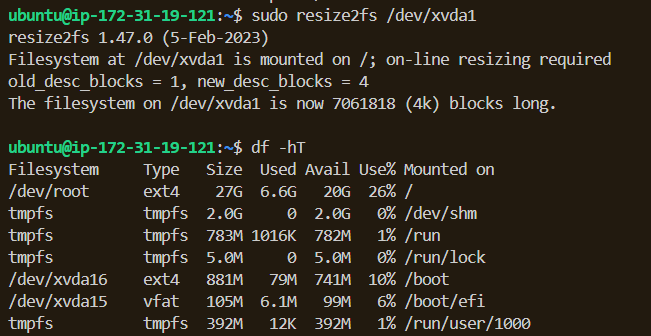
docker build -t java-demoapp .

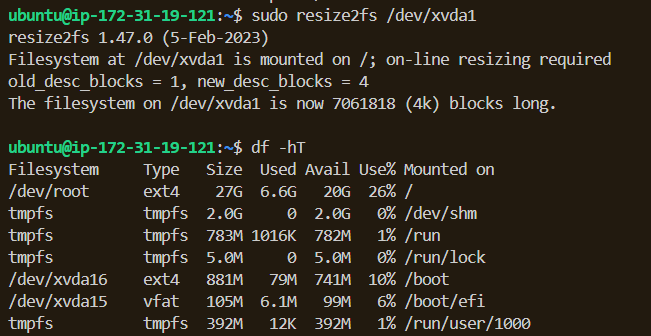
- I had this error : ERROR: failed to solve: failed to register layer: mkdir /opt/java/openjdk/conf/security/policy/limited: no space left on device

- When I check linux machine storage, I see that there is no enough space. I increased linux machine storage and config in the linux machine. Then I can create docker image.









\* I run the kubernetes deployment whit below commands.

kubectl apply -f deployment.yaml

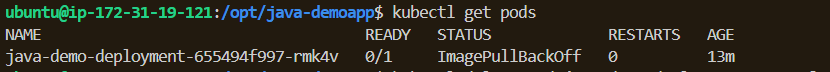
kubectl apply -f ingress.yaml

kubectl apply -f hpa.yaml

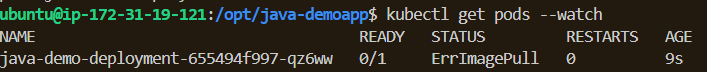
\* After this command I checked the pod. Is it running or not. It isn’t running. Status:Imagepullbackof

I researched and I saw that I should edit my deployment yaml file. I added **“imagePullPolicy: Never”**

row in “spec/containers” to pulling image only from my local.

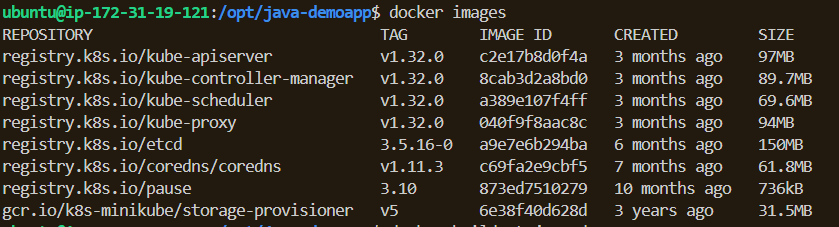


\* Then I recreated the deployment and I checked the pod. It’s not running.

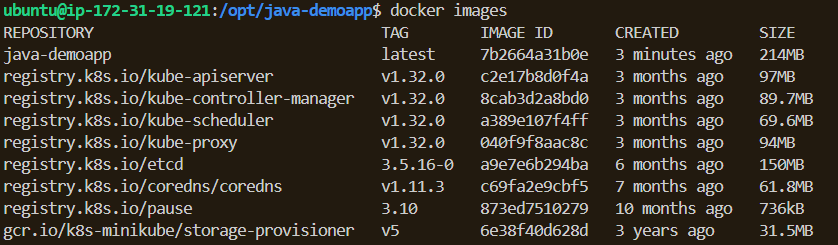


\* I checked the images. I couldn’t see my image. I researched and I found a solution.

With “eval $(minikube docker-env)” command I can see my image in minikube.

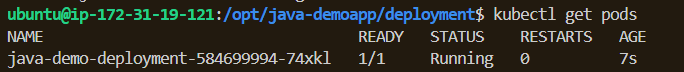






\* I redeployed the application.

kubectl rollout restart deployment java-demo-deployment







1. **Ingress configuration:**

\* Create secret from .crt and .key files.

kubectl create secret tls java-demo-tls --cert=tls.crt --key=tls.key

\* The “secretName=java-demo-tls” specified in the tls section of the ingress.yaml file points to the Kubernetes Secret where Ingress will find the TLS (SSL) certificate. This secret contains the valid SSL certificate and private key for the local.java-demo.com domain.

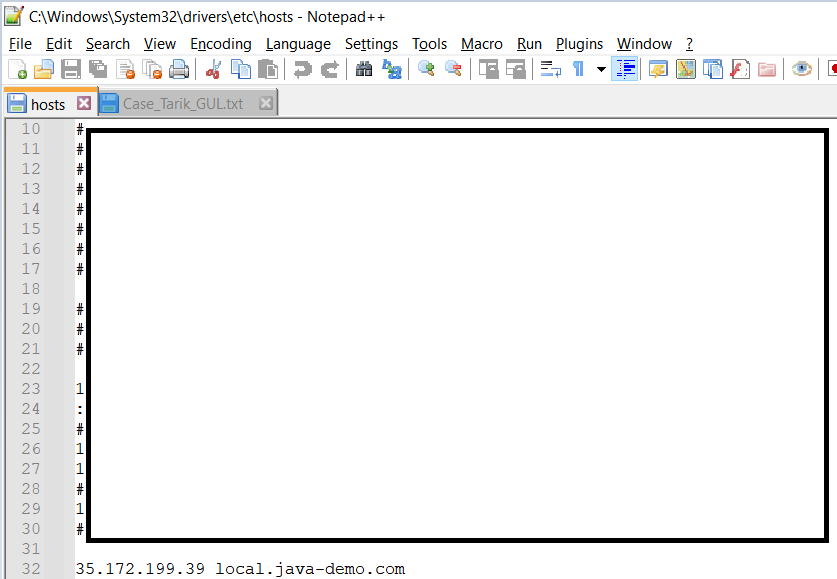
\*In ingress.yaml files Annotations section,

nginx.ingress.kubernetes.io/ssl-redirect: "true" --> HTTP requests from ingress controller forward to HTTPS

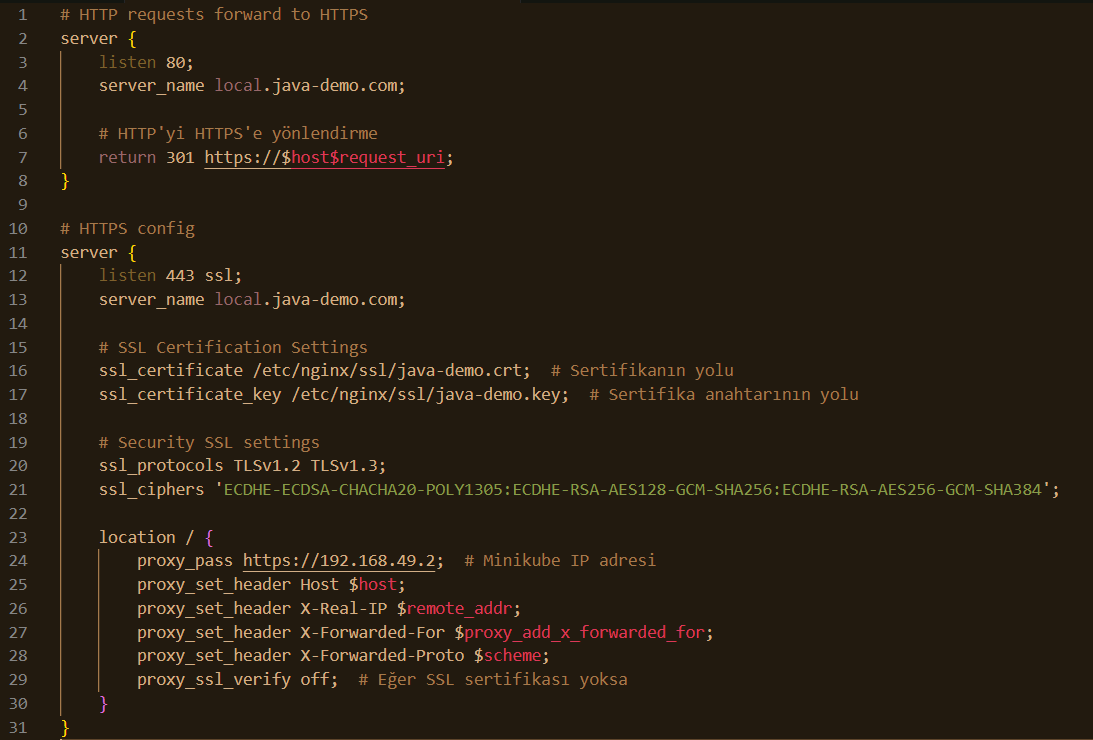
nginx.ingress.kubernetes.io/force-ssl-redirect: "true" --> Guaranteed to all traffic force to HTTPS

1. **Test and Validation:**

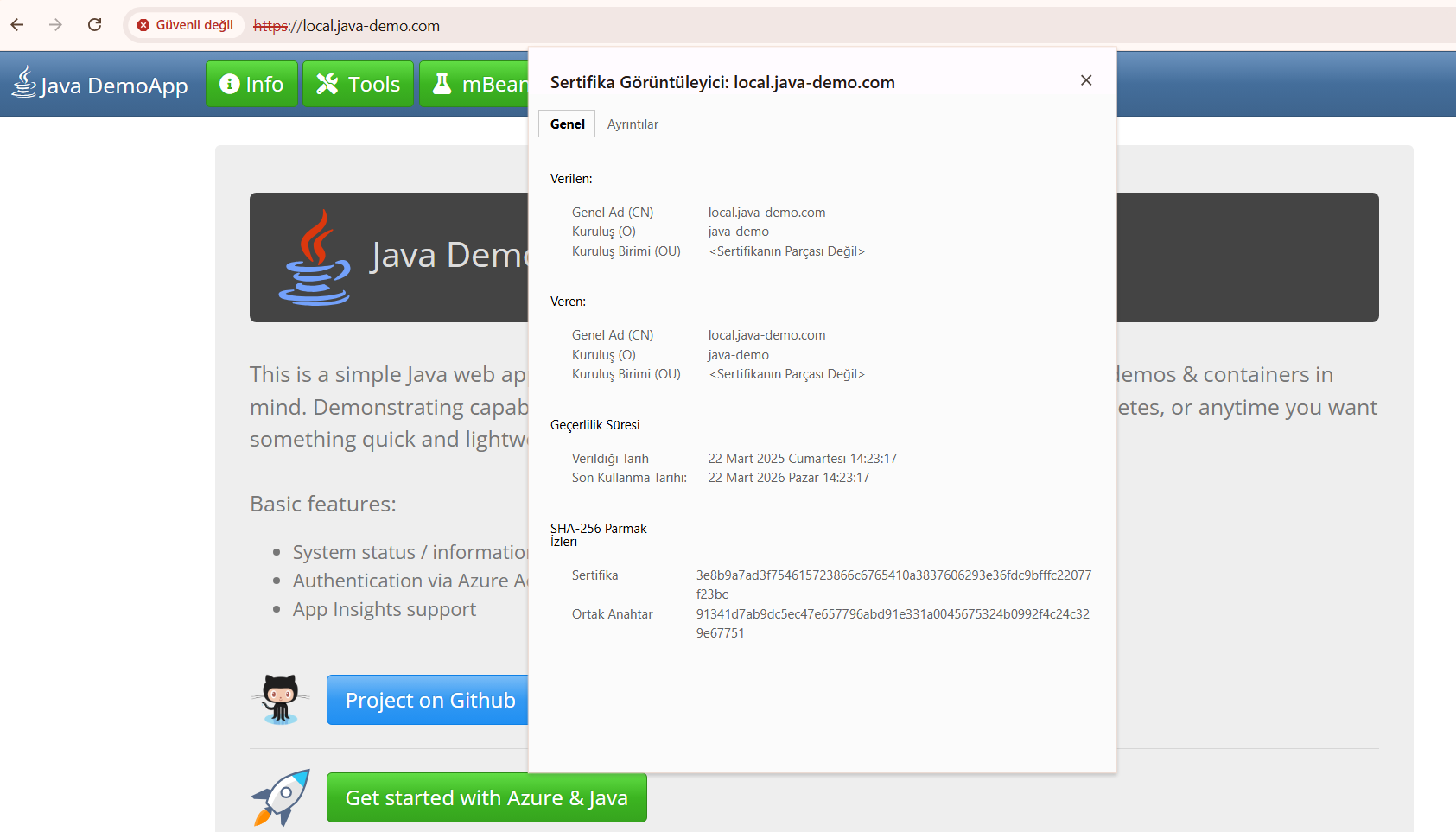
\* I added the my Linux machine Public IP and DNS in etc/hosts file in my local computer:

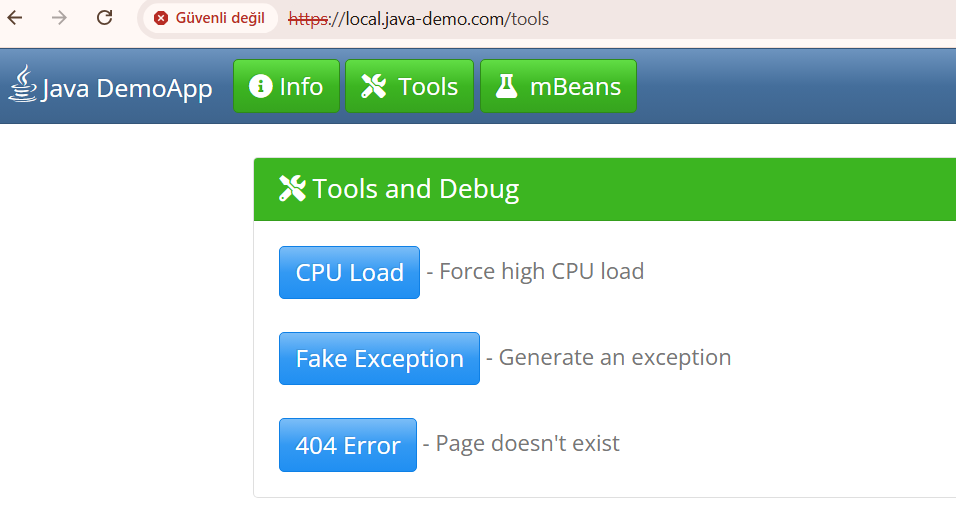


\* when I go to “local.java-demo.com” DNS in my local chrome browser. I can’t access the domain. Because I don’t have any webservice in my Linux Machine in AWS. I installed nginx and I created nginx config file according to https redirection.



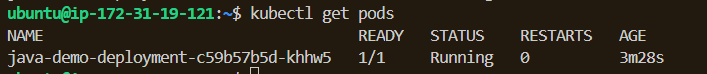
\* Then I go to DNS “local.java-demo.com”



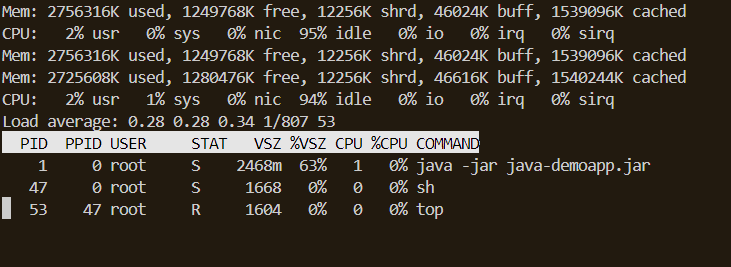


**Horizontal Pod Autoscaler (HPA):**

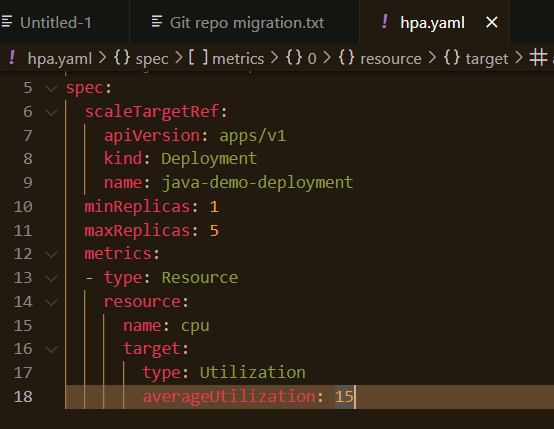
\* averageUtilization = 50 in hpa.yaml file. In this situation There are 1 running pod according to deployment.yaml file. replica=1



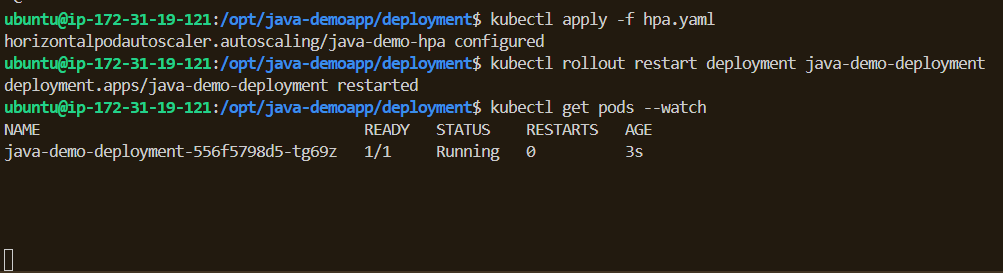
Cpu usage = %2 < 50



\* I changed averageUtilization value from 50 --> 15. It means when cpu usage > 15, New pods will be created.



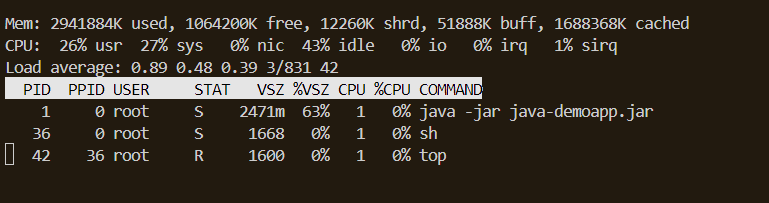
\* I applied to new hpa.yaml file. I restarted the deployment.



\* I started to load test sh script. CPU usage increased.

kubectl exec -it <pod-adı> -- sh -c 'while true; do :; done'





\* New pods created according to hpa.yaml file configuration.

