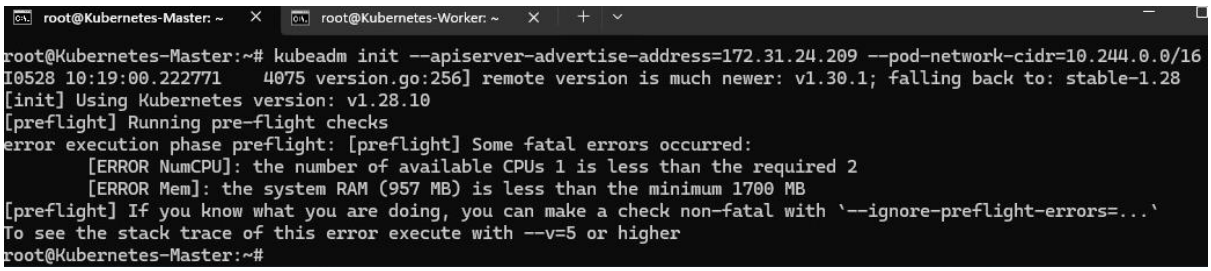
Kubernetes-DAY1

Module Agenda:

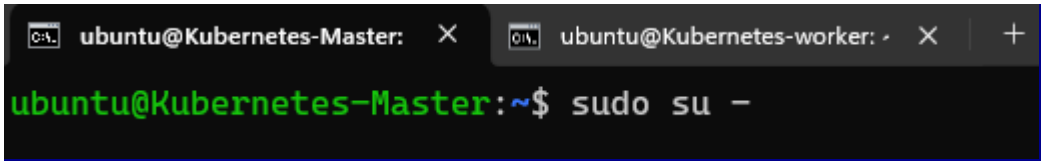
1. Introduction to Kubernetes
2. Docker Swarm vs Kubernetes
3. Kubernetes Architecture
4. Kubernetes Installation
5. Working of Kubernetes
6. Deployments in Kubernetes
7. Services in Kubernetes
8. Ingress in Kubernetes
9. Kubernetes Dashboard

Note: Launch t2.Medium for Kubernetes, don't go with t.2 micro, t2.small, t2.nano,

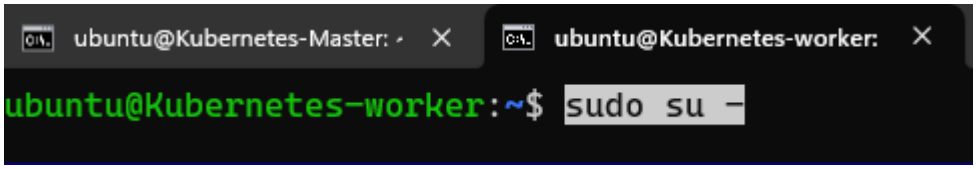


And After Launching t2.medium Login and change hostname as your wish

the user switching from a regular user account to the root account with sudo su.



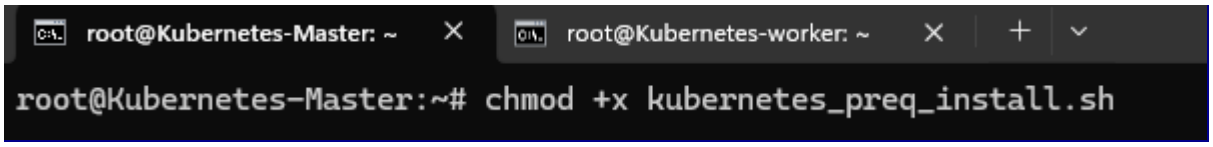
the user switching from a regular user account to the root account with sudo su.



commands to install Kubernetes, a container orchestration platform, on the system

```
ubuntu@Kubernetes-Master:~$ sudo su -
root@Kubernetes-Master:~# ls
kubernetes_preq_install.sh  snap
root@Kubernetes-Master:~# cat kubernetes_preq_install.sh
sudo apt update -y
sudo apt install curl apt-transport-https -y
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list
.d/kubernetes.list
sudo apt install -y kubeadm=1.28.1-1.1 kubelet=1.28.1-1.1 kubectl=1.28.1-1.1
sudo apt update -y
sudo apt install kubelet kubeadm kubectl -y
sudo apt-mark hold kubelet kubeadm kubectl
sudo swapoff -a
sudo sed -i '/ swap / s/^(.*)$/#\1/g' /etc/fstab
free -m
sudo tee /etc/modules-load.d/k8s.conf <<EOF
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter
sudo tee /etc/sysctl.d/kubernetes.conf<<EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF
sudo sysctl --system
sudo apt install -y curl gnupg2 software-properties-common apt-transport-https ca-certificates
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/docker-archive-keyring.gpg
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt update -y
sudo apt install -y containerd.io
sudo mkdir -p /etc/containerd
sudo containerd config default|sudo tee /etc/containerd/config.toml
sudo sed -i 's/SystemdCgroup \= false/SystemdCgroup \= true/g' /etc/containerd/config.toml
sudo systemctl restart containerd
sudo systemctl enable containerd
systemctl status containerd
root@Kubernetes-Master:~# kubernetes_preq_install.sh
```

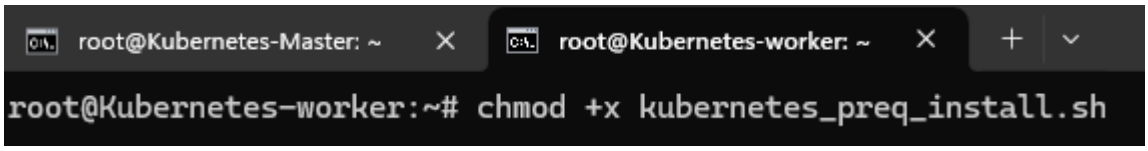
The script name is `kubernetes_preq_install.sh`. chmod is used to change the permissions of files and directories in Linux/Unix systems. IN MASTER MACHINE



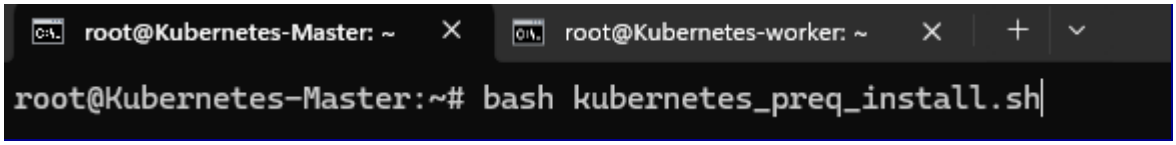
commands to install Kubernetes, a container orchestration platform, on the system IN WORKERS MACHINE

```
ubuntu@Kubernetes-worker:~$ sudo su -
root@Kubernetes-worker:~# ls
kubernetes_preq_install.sh  snap
root@Kubernetes-worker:~# cat kubernetes_preq_install.sh
sudo apt update -y
sudo apt install curl apt-transport-https -y
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list
.d/kubernetes.list
sudo apt install -y kubeadm=1.28.1-1.1 kubelet=1.28.1-1.1 kubectl=1.28.1-1.1
sudo apt update -y
sudo apt install kubelet kubeadm kubectl -y
sudo apt-mark hold kubelet kubeadm kubectl
sudo swapoff -a
sudo sed -i '/ swap / s/^(.*)$/#\1/g' /etc/fstab
free -m
sudo tee /etc/modules-load.d/k8s.conf <<EOF
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter
sudo tee /etc/sysctl.d/kubernetes.conf<<EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF
sudo sysctl --system
sudo apt install -y curl gnupg2 software-properties-common apt-transport-https ca-certificates
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/docker-archive-keyring.gpg
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt update -y
sudo apt install -y containerd.io
sudo mkdir -p /etc/containerd
sudo containerd config default|sudo tee /etc/containerd/config.toml
sudo sed -i 's/SystemdCgroup \= false/SystemdCgroup \= true/g' /etc/containerd/config.toml
sudo systemctl restart containerd
sudo systemctl enable containerd
systemctl status containerd
root@Kubernetes-worker:~#
```

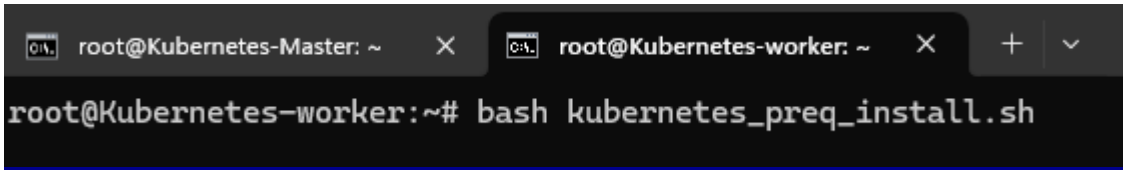
The script name is `kubernetes_preq_install.sh`. chmod is used to change the permissions of files and directories in Linux/Unix systems. IN WORKERS MACHINE



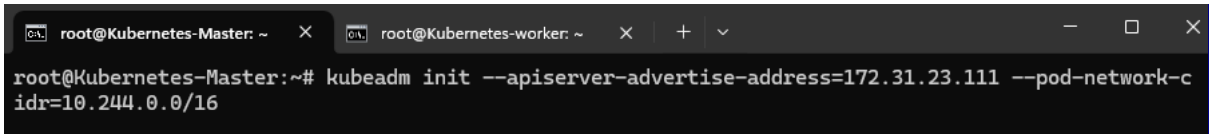
trying to install Kubernetes, a container orchestration platform, IN MASTER MACHINE



trying to install Kubernetes, a container orchestration platform, IN WORKERS MACHINE



initialization of a Kubernetes cluster on a machine with IP address 172.31.23.111. Kubernetes is a system for automating deployment, scaling, and management of containerized applications.



for joining a Kubernetes worker node to a cluster with the following details:

- API server address: 172.31.18.124:6443
- Discovery token: 610kcm.5emj9zn875dtw1v5
- Discovery token CA certificate hash: sha256:289b550db765d4f8bb4279c693a28b6a4feebc887b027a23c5fef64c4ed8af6

AND COPY THE TOKEN IN Worker node which I didn't captured

```
kubeadm join 172.31.18.124:6443 --token 6l0kcm.5emj9zn875dtw1v5 \
        --discovery-token-ca-cert-hash sha256:289b550db765d4f8bb4279c693a28b6a
4feebc887b027a23c5fef64c4ed8af6d
```

The three commands in the image set up a local Kubernetes cluster on your machine.

1. **Makes a directory:** Creates a directory named .kube in your home directory to store Kubernetes configuration files.
2. **Copies configuration file:** Copies the admin configuration file from the system directory to your .kube directory.
3. **Sets ownership:** Sets the ownership and permissions of the configuration file so only your user can access it.

```
To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:
```

command applies a configuration file from GitHub to set up Flannel for your Kubernetes cluster in one line:

Flannel is a network fabric implementation for Kubernetes that provides pod networking without a dedicated overlay network.

```
ubuntu@kubernetesmaster:~$ mkdir -p $HOME/.kube
ubuntu@kubernetesmaster:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/c
onfig
ubuntu@kubernetesmaster:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@kubernetesmaster:~$ kubectl apply -f https://raw.githubusercontent.com/
flannel-io/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@kubernetesmaster:~$
```

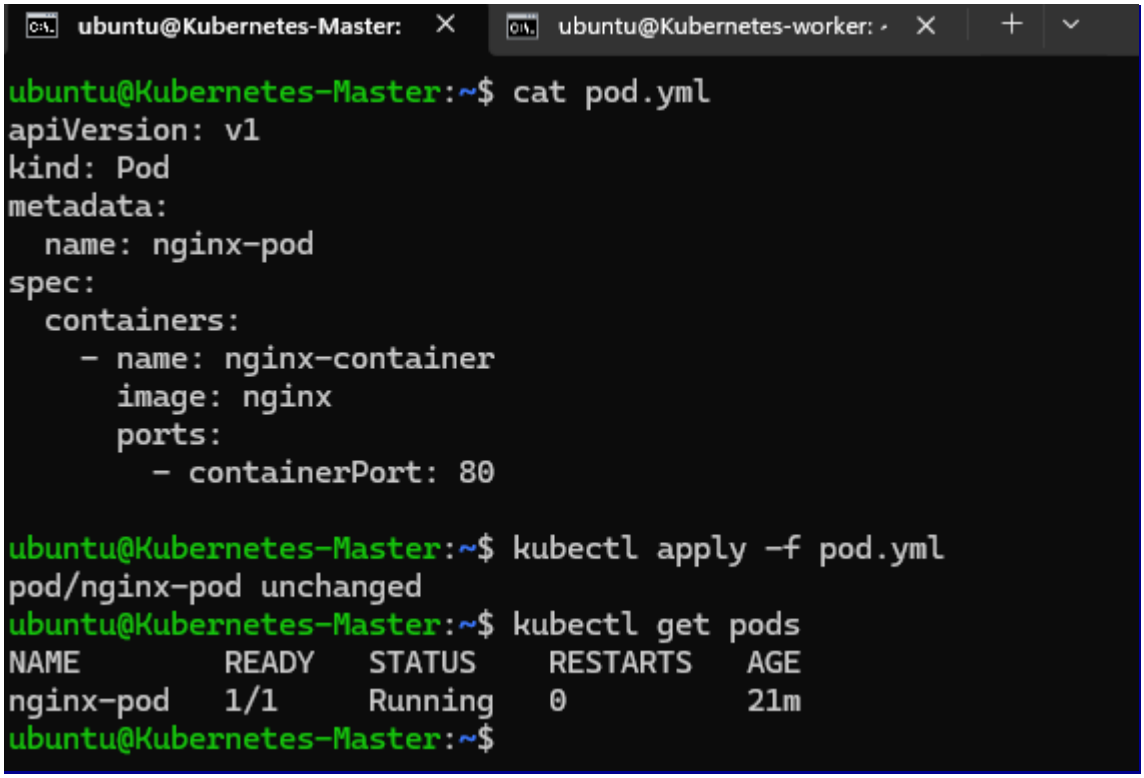
`kubectl get nodes` to list the nodes (computers) in a Kubernetes cluster and their current status.

```
ubuntu@Kubernetes-Master:  ubuntu@Kubernetes-worker:
ubuntu@Kubernetes-Master:~$ kubectl get nodes
NAME                STATUS   ROLES           AGE     VERSION
kubernetes-master   Ready    control-plane   3h20m   v1.28.10
kubernetes-worker   Ready    <none>          3h18m   v1.28.10
ubuntu@Kubernetes-Master:~$
```

Kubernetes-DAY2

- `kubectl`: This is the command-line tool for interacting with Kubernetes clusters.
- `apply`: This subcommand tells `kubectl` to create or update resources in your cluster.
- `-f`: This flag specifies that the following argument (`pod.yml`) is a file containing the resource definition.
- `pod.yml`: This is the YAML file that defines the pod configuration. It includes details like container image, ports, storage, etc.

The command `kubectl get pods` is used to retrieve information about the pods in a Kubernetes cluster



`kubectl describe pod nginx-pod` provides detailed information about a specific pod in your Kubernetes cluster named `nginx-pod`.

By running this command, you'll get a comprehensive view of the `nginx-pod`, including its:

- Status (Running, Pending, etc.)
- IP address
- Container details (image, ports, etc.)
- Events related to the pod
- Configuration details

- And more

```
ubuntu@Kubernetes-Master:~$ kubectl describe pod nginx-pod
Name:             nginx-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             kubernetes-worker/172.31.53.219
Start Time:       Wed, 29 May 2024 09:36:37 +0000
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.1.2
IPs:
  IP:  10.244.1.2
```

```
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  27m   default-scheduler  Successfully assigned default/nginx-pod to kubernete
s-worker
  Normal  Pulling    27m   kubelet            Pulling image "nginx"
  Normal  Pulled     27m   kubelet            Successfully pulled image "nginx" in 4.027s (4.027s
including waiting)
  Normal  Created    27m   kubelet            Created container nginx-container
  Normal  Started    27m   kubelet            Started container nginx-container
```

The command `kubectl logs nginx-pod` displays the logs generated by the containers running within the pod named `nginx-pod`

```
ubuntu@Kubernetes-Master:~$ kubectl logs nginx-pod
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/05/29 09:36:41 [notice] 1#1: using the "epoll" event method
2024/05/29 09:36:41 [notice] 1#1: nginx/1.25.5
2024/05/29 09:36:41 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/05/29 09:36:41 [notice] 1#1: OS: Linux 6.8.0-1008-aws
2024/05/29 09:36:41 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/05/29 09:36:41 [notice] 1#1: start worker processes
2024/05/29 09:36:41 [notice] 1#1: start worker process 30
2024/05/29 09:36:41 [notice] 1#1: start worker process 31
ubuntu@Kubernetes-Master:~$
```

`kubectl exec -it pod/nginx-pod -- /bin/bash`. This command allows you to establish an interactive shell session directly with a specific container inside a pod in your Kubernetes cluster

```
ubuntu@Kubernetes-Master:~$ kubectl exec -it pod/nginx-pod -- /bin/bash
root@nginx-pod:/# ls
bin   dev                  docker-entrypoint.sh  home  lib64  mnt  proc  run   srv  tmp  var
boot  docker-entrypoint.d  etc                   lib   media  opt  root  sbin  sys  usr
root@nginx-pod:/#
```

Kubectl delete pod/nginx-pod this command delete the pod

```
ubuntu@Kubernetes-Master:~$ kubectl get pod
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          30m
ubuntu@Kubernetes-Master:~$ kubectl delete pod/nginx-pod
pod "nginx-pod" deleted
ubuntu@Kubernetes-Master:~$ kubectl apply -f pod.yml
pod/nginx-pod created
ubuntu@Kubernetes-Master:~$ kubectl get pod
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          9s
```

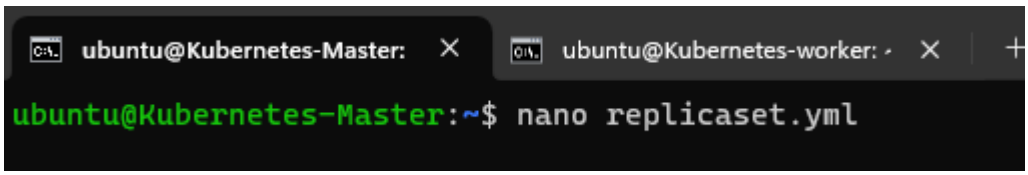
`kubectl get pod -o wide` retrieves information about pods in your Kubernetes cluster and displays it in a wider format with additional details.

- **Node:** The name of the Kubernetes node where the pod is running.
- **NOMINATED NODE:** The node the pod is nominated to run on (if scheduling is not yet complete).
- **READINESS GATES:** The readiness gates of the pod (if any).
- **AGE:** The age of the pod.
- **IP:** The IP address of the pod.

```
ubuntu@Kubernetes-Master:~$ kubectl get pod -o wide
NAME        READY   STATUS    RESTARTS   AGE   IP           NODE                NOMINATED NODE   R
EADINESS GATES
nginx-pod   1/1     Running   0          52s   10.244.1.3   kubernetes-worker   <none>           <
none>
ubuntu@Kubernetes-Master:~$
```

```
ubuntu@Kubernetes-Master:~$ kubectl get nodes
NAME                STATUS   ROLES           AGE     VERSION
kubernetes-master   Ready    control-plane   3h34m   v1.28.10
kubernetes-worker   Ready    <none>          3h32m   v1.28.10
ubuntu@Kubernetes-Master:~$ kubectl get nodes -o wide
NAME                STATUS   ROLES           AGE     VERSION    INTERNAL-IP     EXTERNAL-IP   OS-IMAGE
   KERNEL-VERSION   CONTAINER-RUNTIME
kubernetes-master   Ready    control-plane   3h34m   v1.28.10   172.31.62.130   <none>        Ubuntu 24.04 LTS
   6.8.0-1008-aws   containerd://1.6.32
kubernetes-worker   Ready    <none>          3h33m   v1.28.10   172.31.53.219   <none>        Ubuntu 24.04 LTS
   6.8.0-1008-aws   containerd://1.6.32
ubuntu@Kubernetes-Master:~$
```

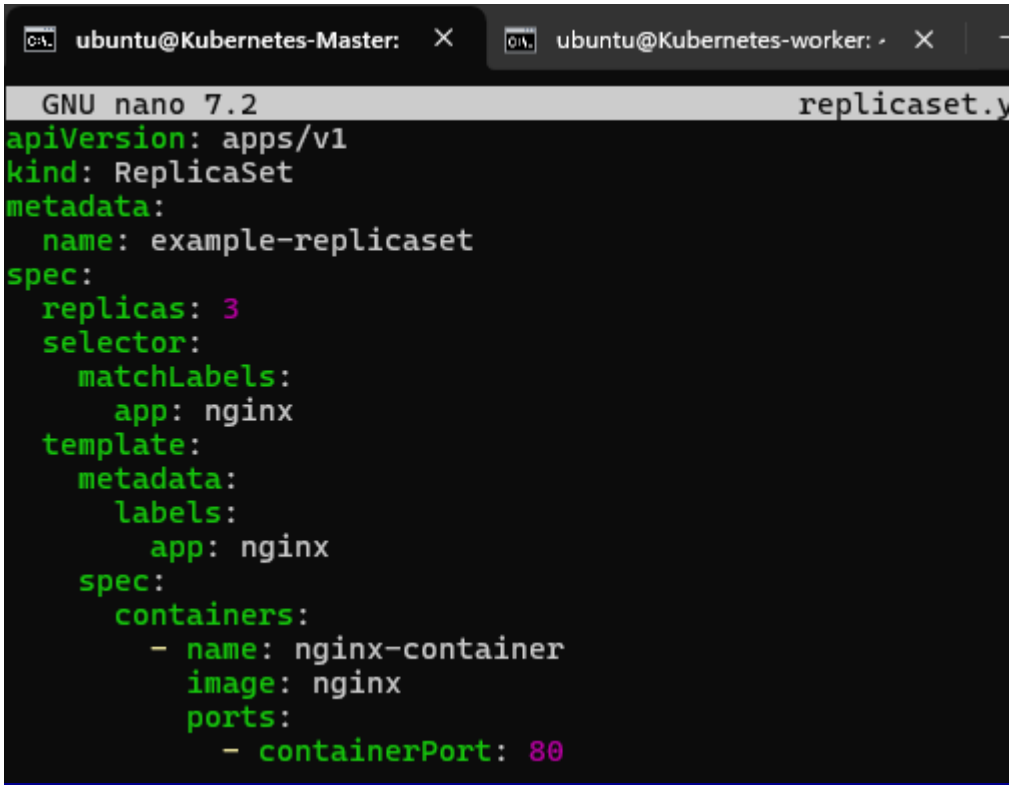
A ReplicaSet in Kubernetes ensures a specified number of identical pods are running at all times.



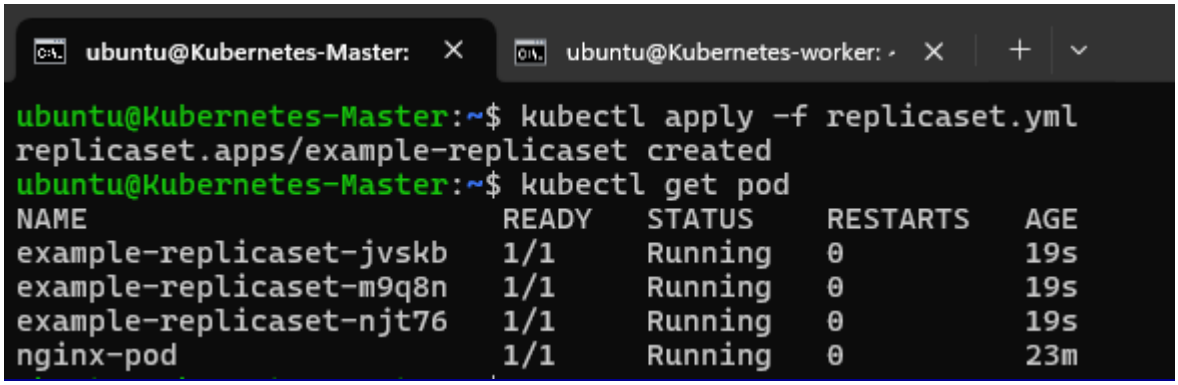
- **replicas: 3:** This sets the number of replicas (identical pods) that the ReplicaSet will manage.

- **app: nginx:** This defines a label selector that will match pods with the label `app: nginx`. The ReplicaSet will manage any pods that match this selector.

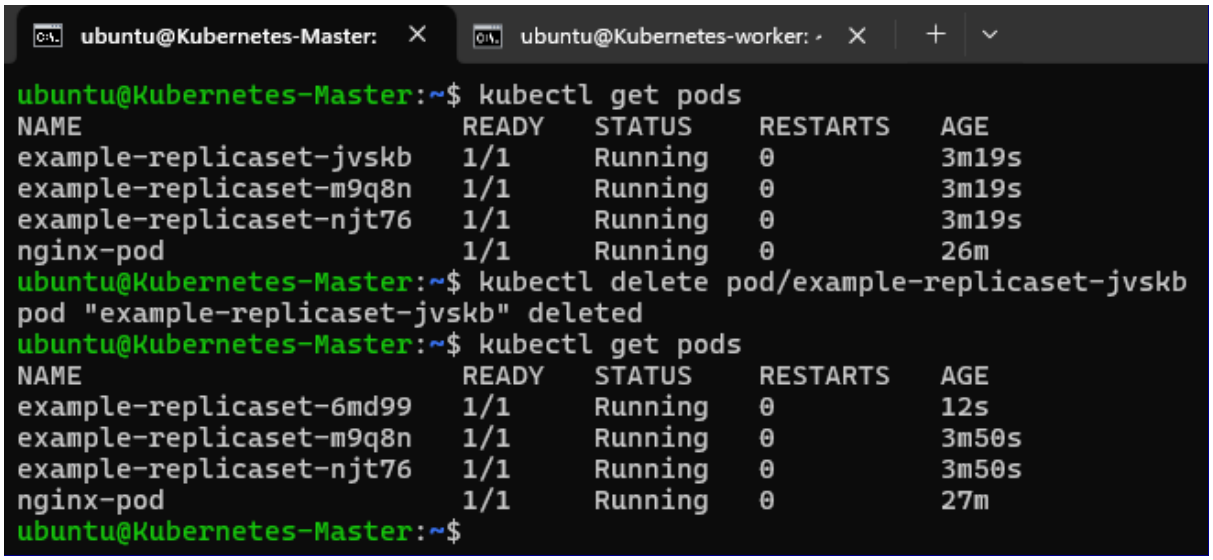
- **nginx-container:** This defines the container template that will be used to create the pods. The container image is set to `nginx` and the container port is set to `80`.



We Given 3 Replicaset so it will run 3 Pods



We Delete one Pod and automatically Kubernetes will its created another one



REPLICASET

SINGLE POD

ubuntu@Kubernetes-Master: ubuntu@Kubernetes-worker:

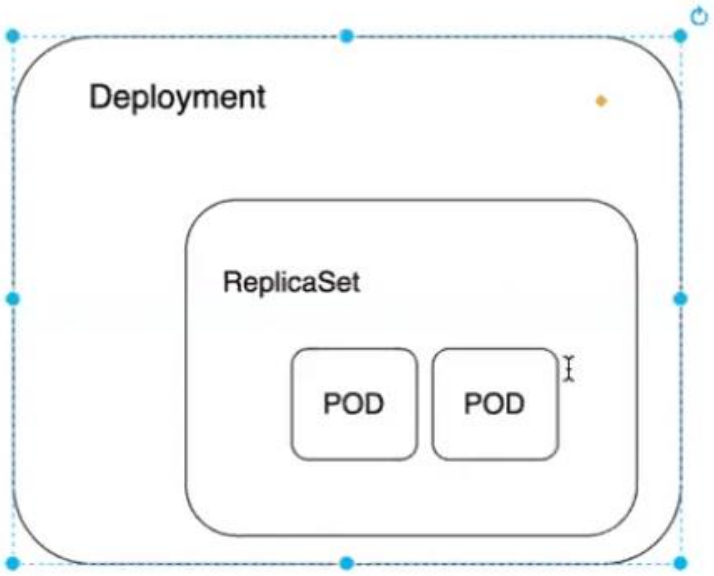
GNU nano 7.2 replicaset.y

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: example-replicaset
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx-container
          image: nginx
          ports:
            - containerPort: 80
```

ubuntu@Kubernetes-Master: ubuntu@Kuberne

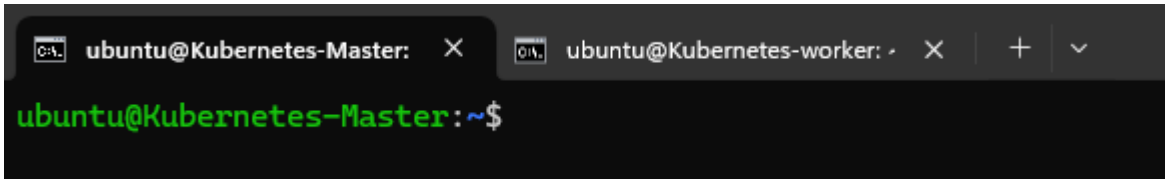
```
ubuntu@Kubernetes-Master:~$ cat pod.yml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
    - name: nginx-container
      image: nginx
      ports:
        - containerPort: 80
```

Kubernetes-DAY3

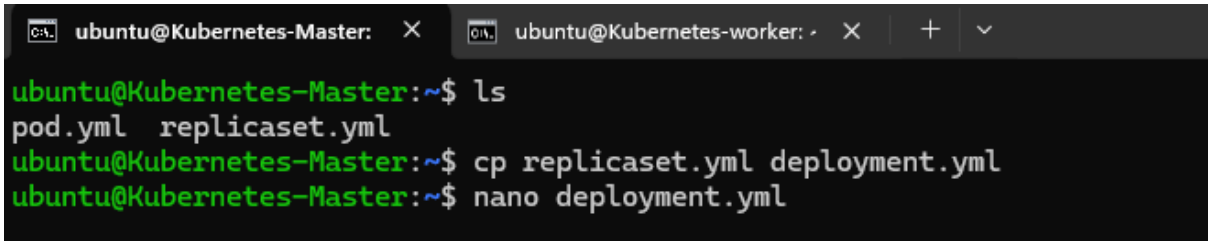


it is helpful in rollback

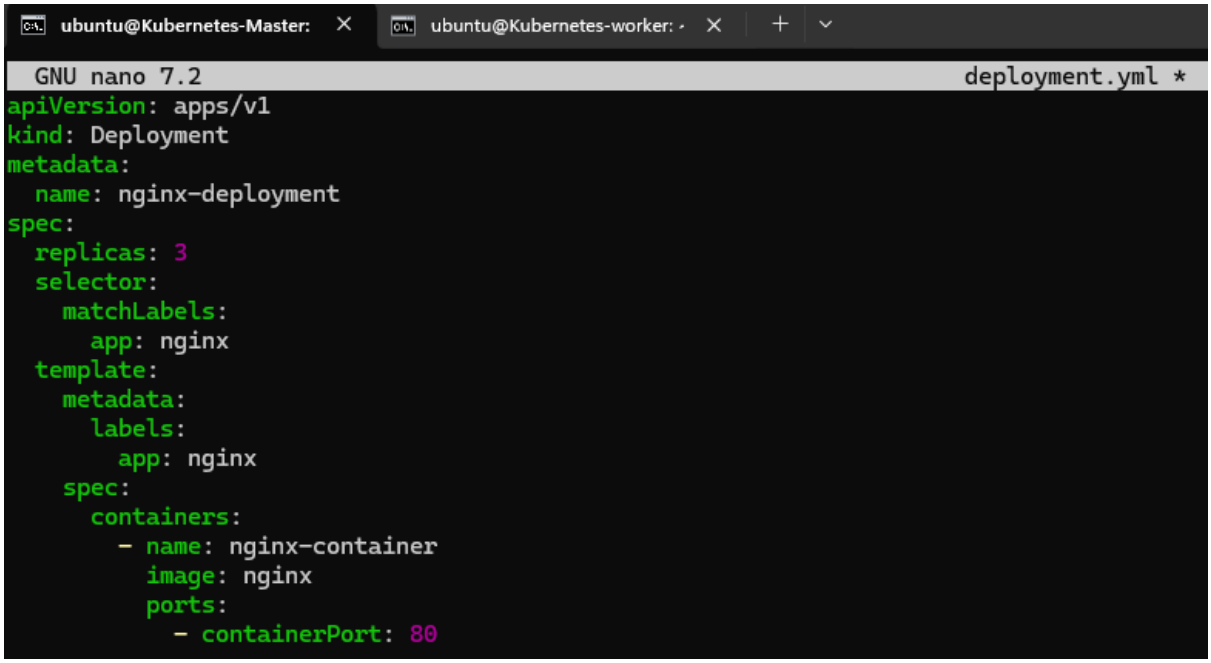
1. Launch Kubernetes- Master and Kubernetes- Slave



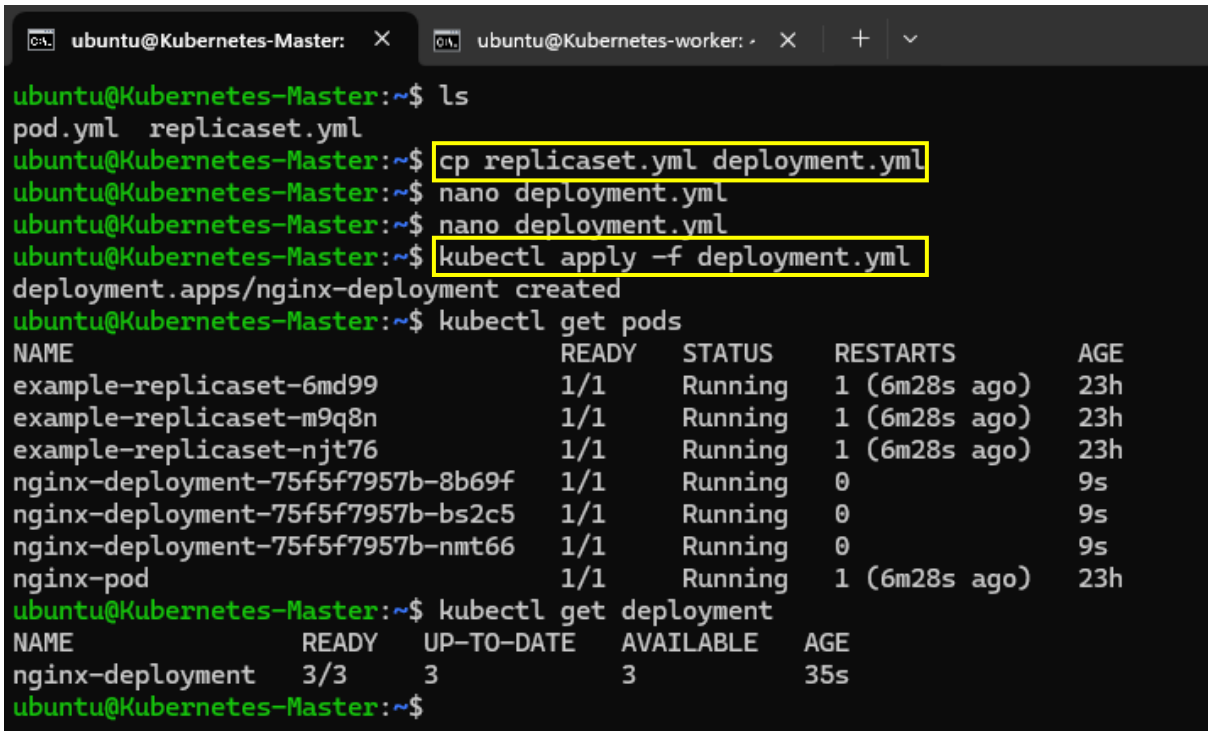
2. Now We Copied Replicaset script to deployment.yml



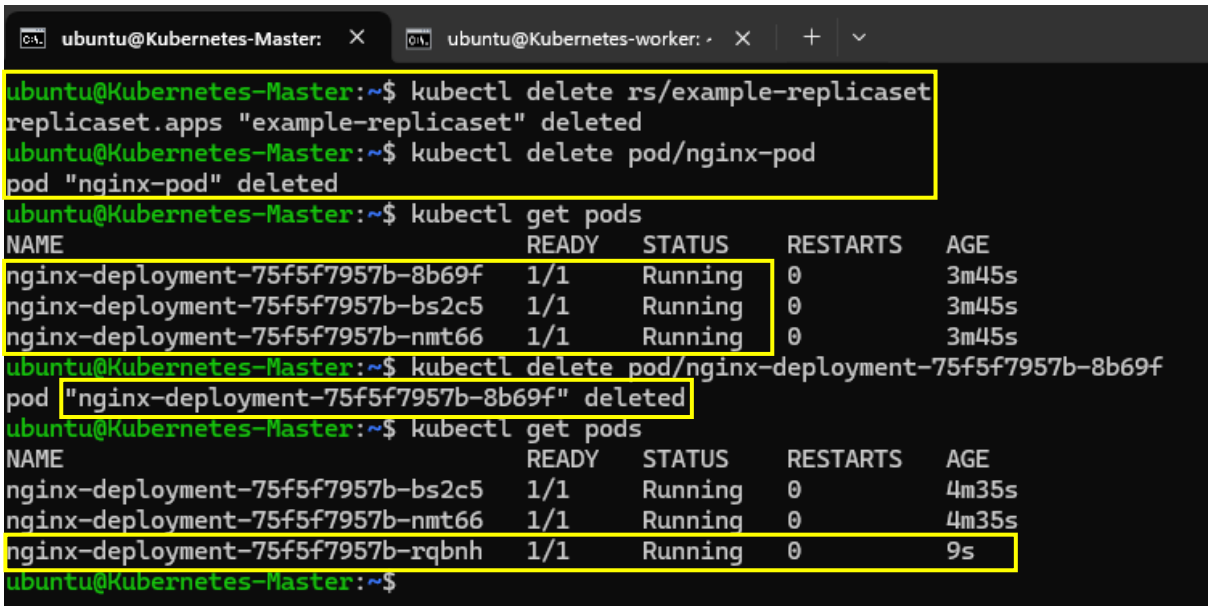
3. wrote the script



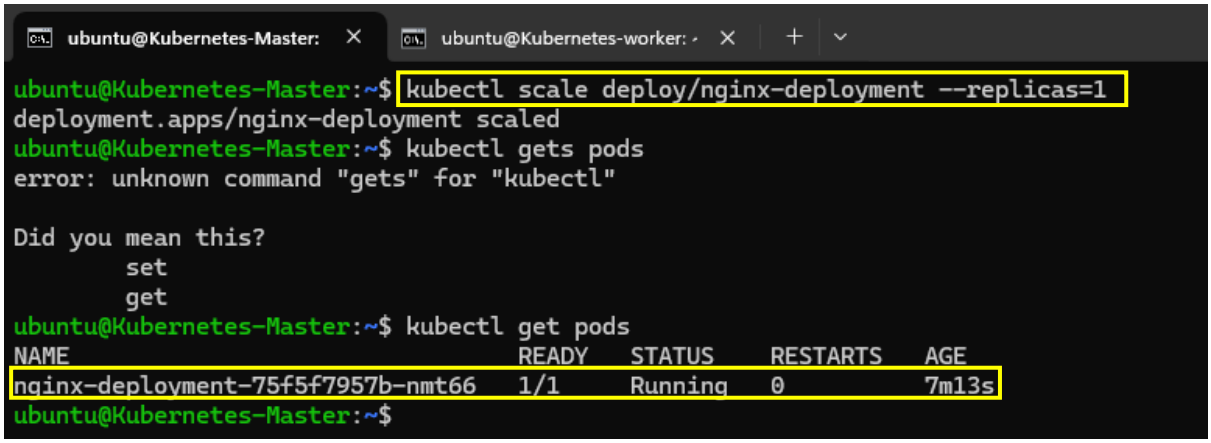
4. kubectl apply -f deployment.yml and its created. And 3 nginx deployment has created



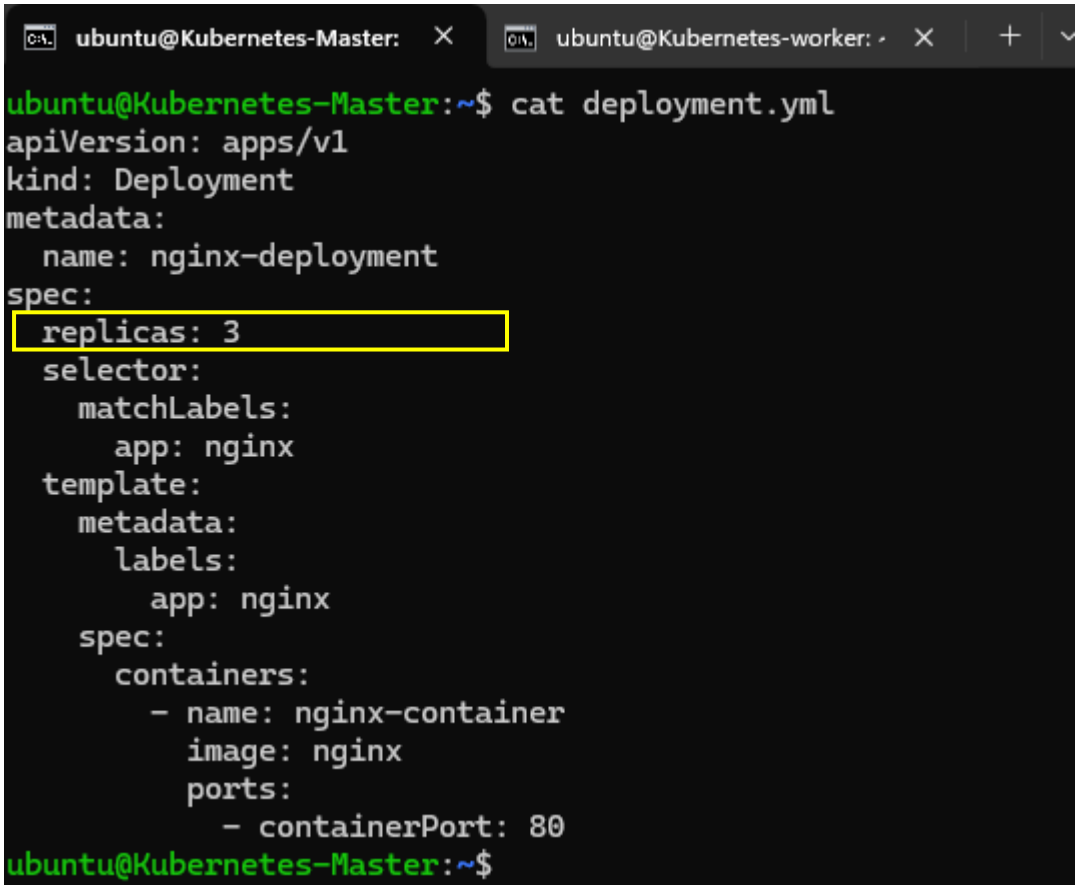
5. Deleting all rs/example-replicaset and deleting one pod/nginx-pod and only deployment is available and I am deleting one deployment and after that check pods replica created another deployment server because replicas is inside the deployment.



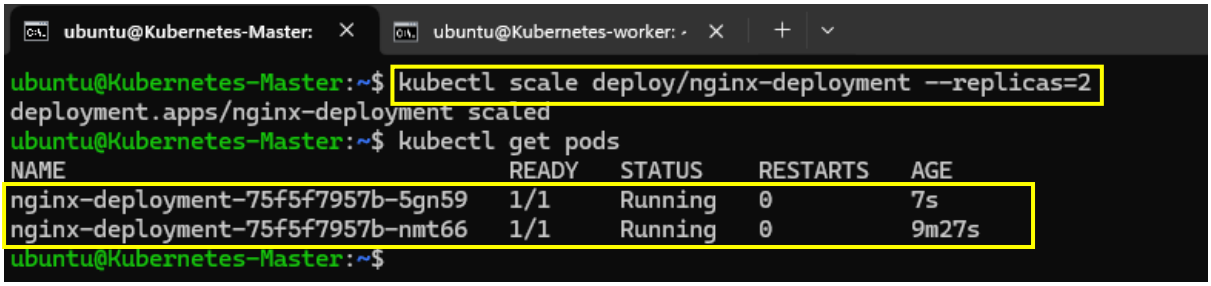
6. Here we scaled the only 1 replicas. We can see only one deployment server is available



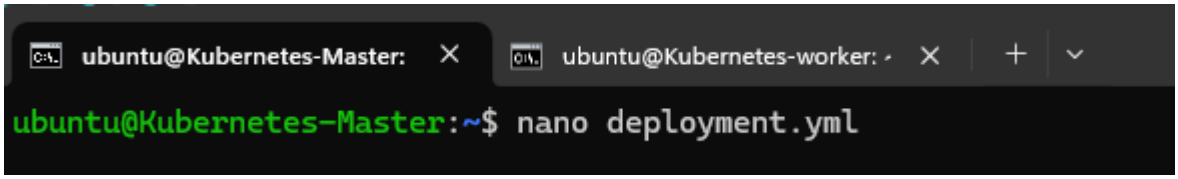
7. We can see inside the deployment.yml there is 3 replicas.



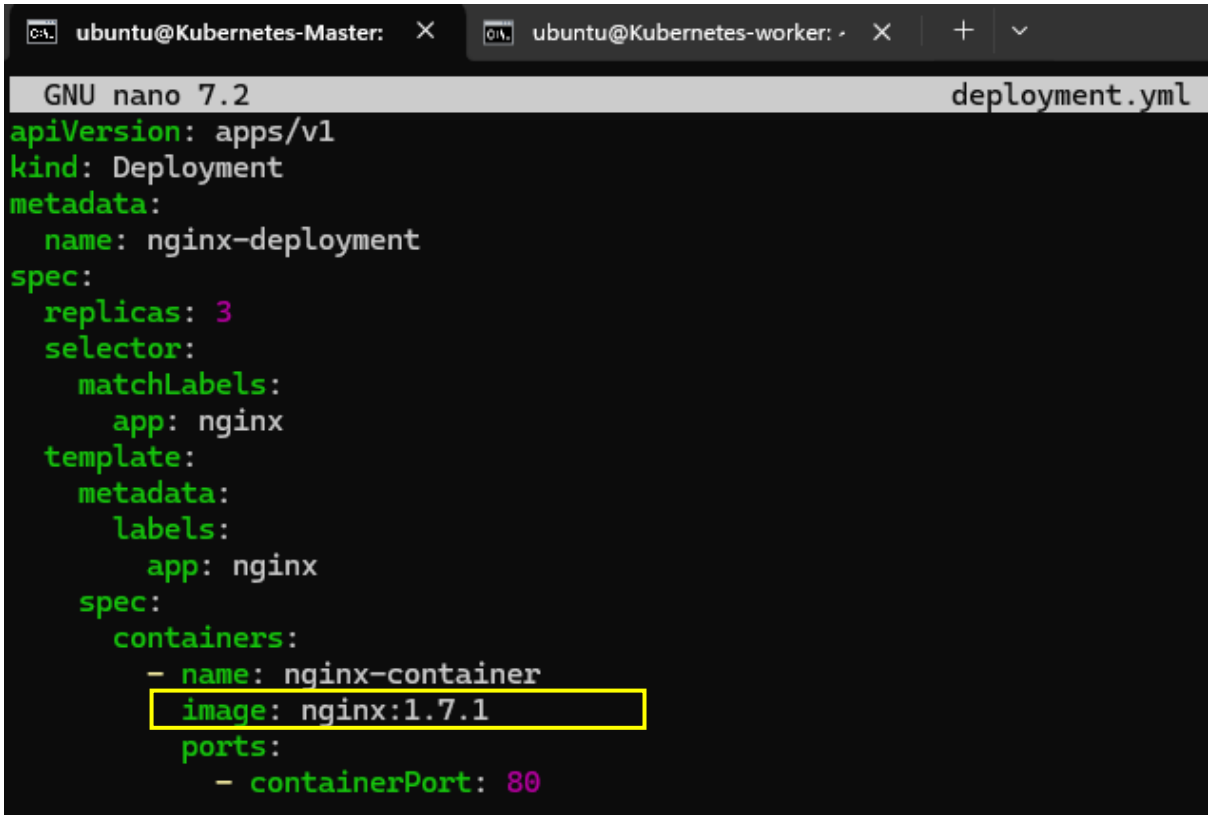
8. Now we scaled 2 replicas



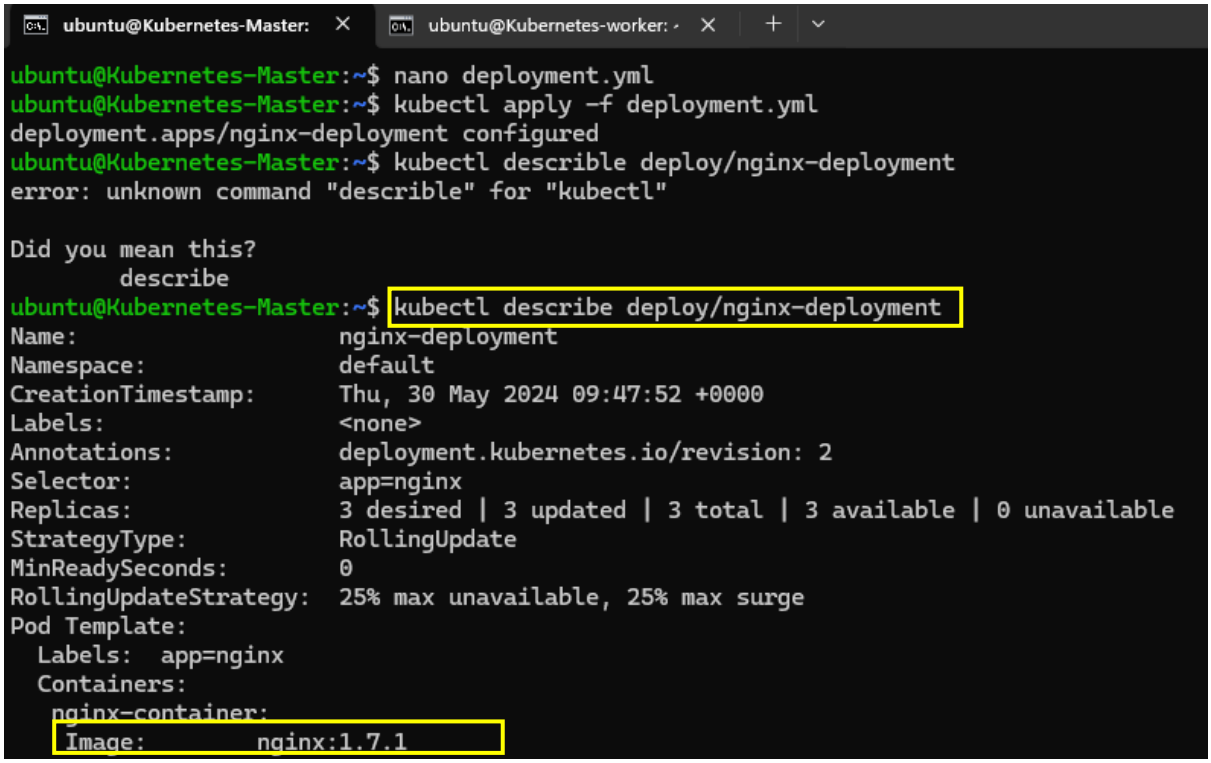
9. Now Again Edit the deployment.yml



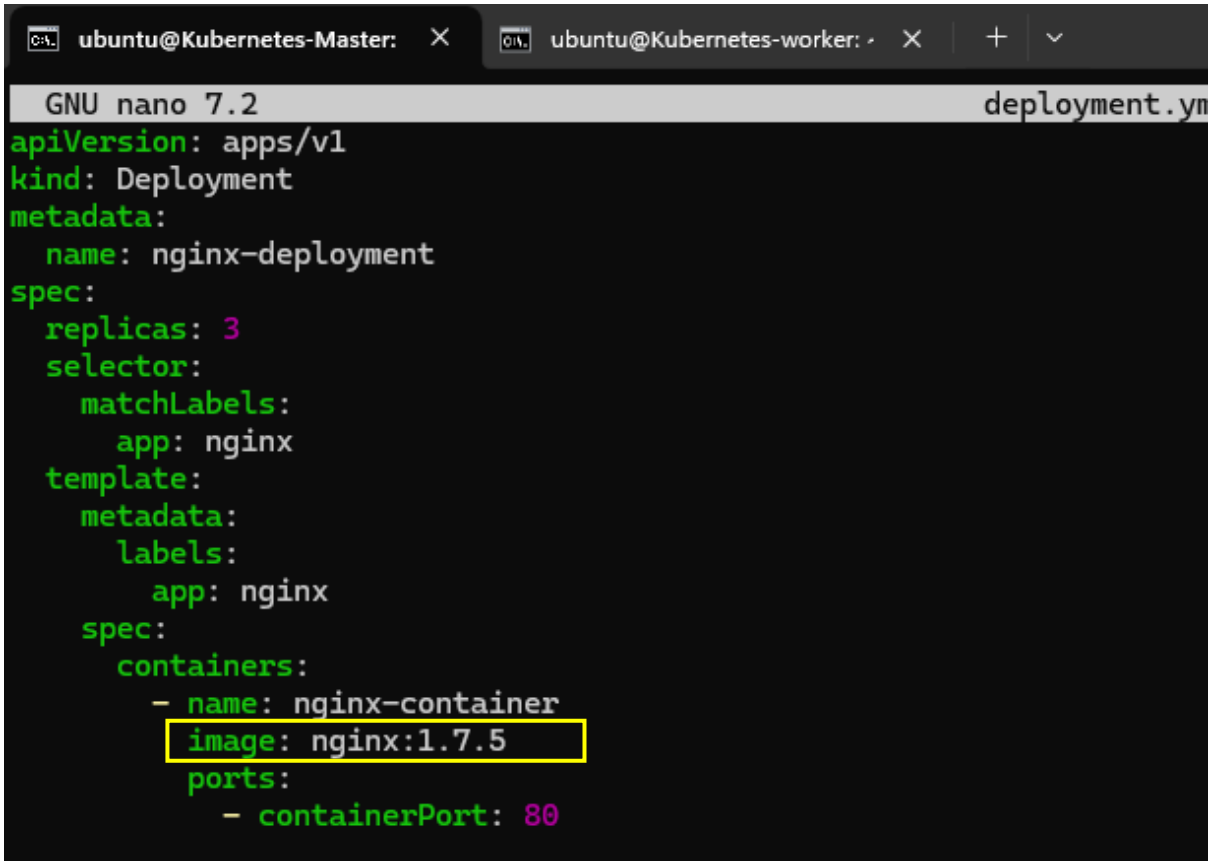
10. This time we are Modifying image nginx with version 1.7.1



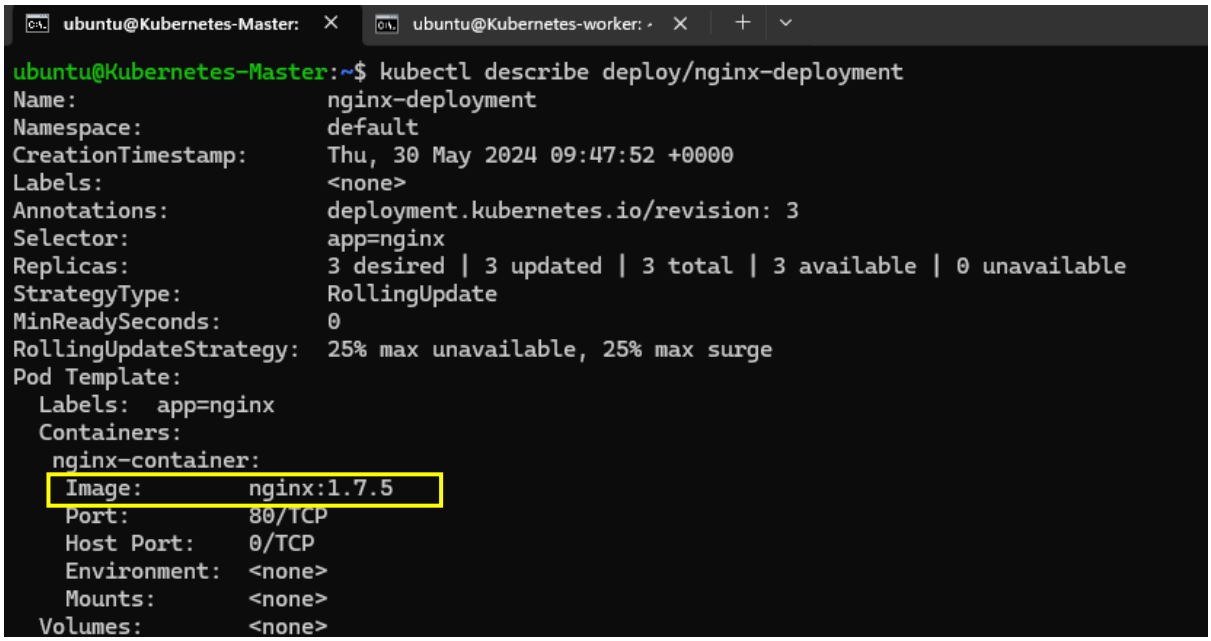
11. Apply and Describe it and we can see the image version is nginx:1.7.1



12. Again Modified



13. Now we can the updated one



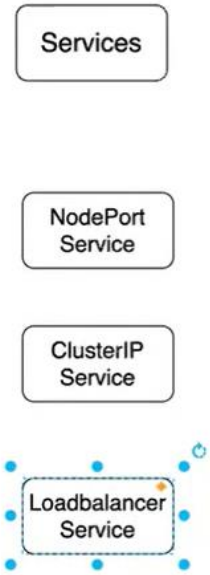
14. Rollout means it will go back and fetch the previous version

```
ubuntu@Kubernetes-Master:~$ kubectl rollout undo deployment/nginx-deployment
deployment.apps/nginx-deployment rolled back
ubuntu@Kubernetes-Master:~$ kubectl describe deploy/nginx-deployment
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Thu, 30 May 2024 09:47:52 +0000
Labels:                 <none>
Annotations:            deployment.kubernetes.io/revision: 4
Selector:               app=nginx
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx-container:
    Image:        nginx:1.7.1
```

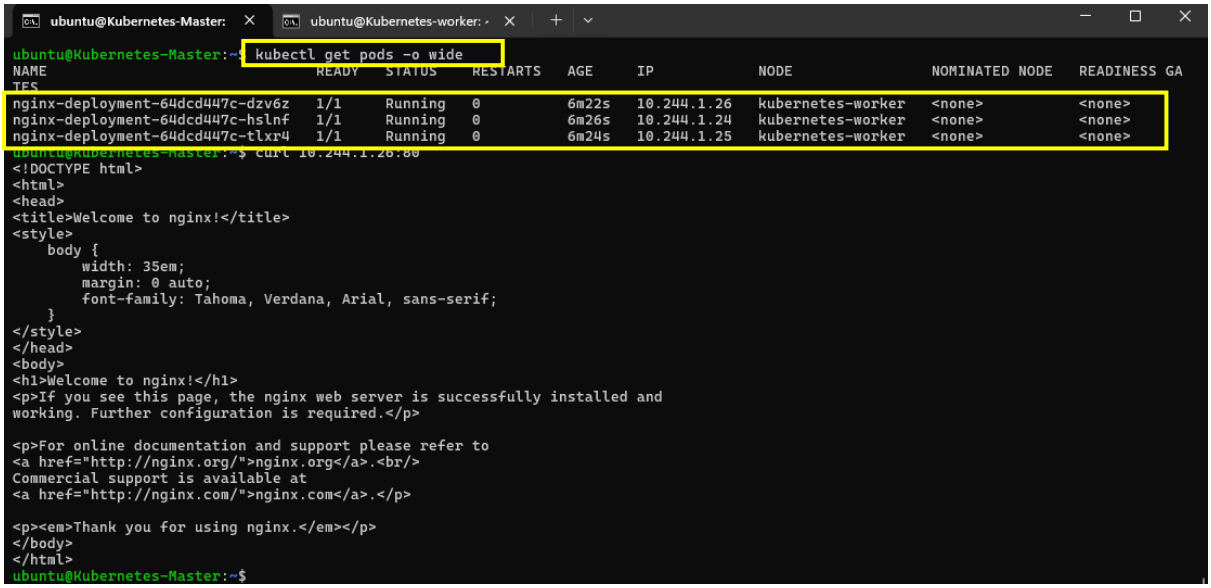
NodePort: Makes the service accessible on all nodes through a single port, exposing it externally via firewall configuration on the cluster's nodes.

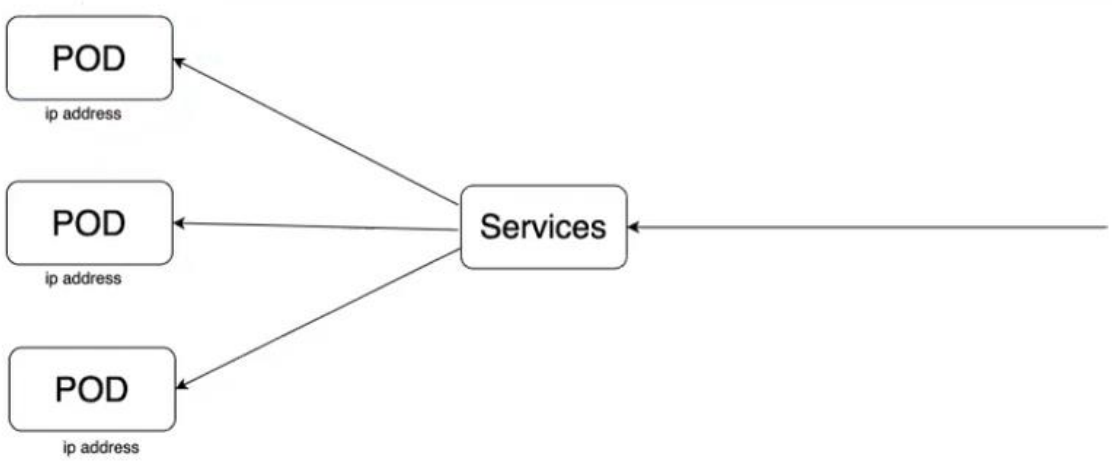
ClusterIP: Creates a service accessible only from within the Kubernetes cluster using a cluster-internal IP address.

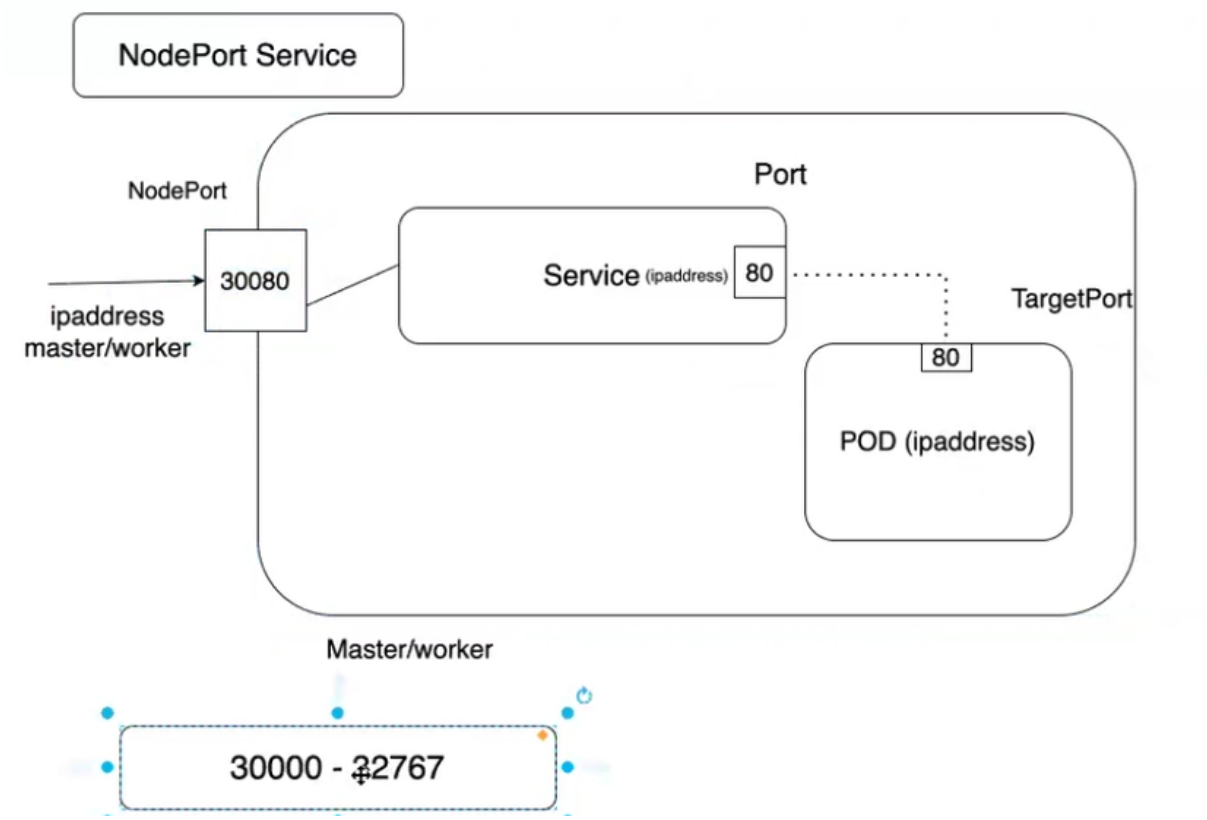
LoadBalancer: Exposes the service externally using a cloud provider's load balancer, typically with a dynamic IP address.



15. kubectl get pods -o wide(Displays detailed information about all pods in the current Kubernetes namespace.)

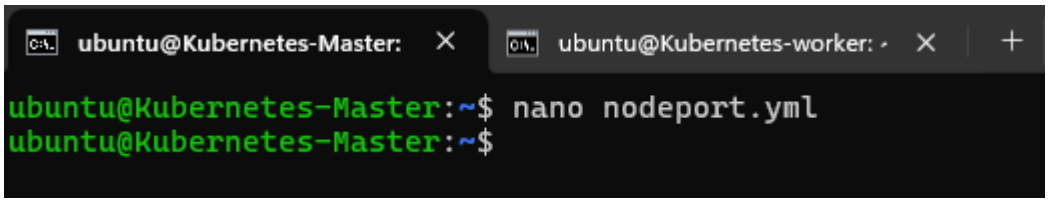






16. Now Creating nodeport.yml

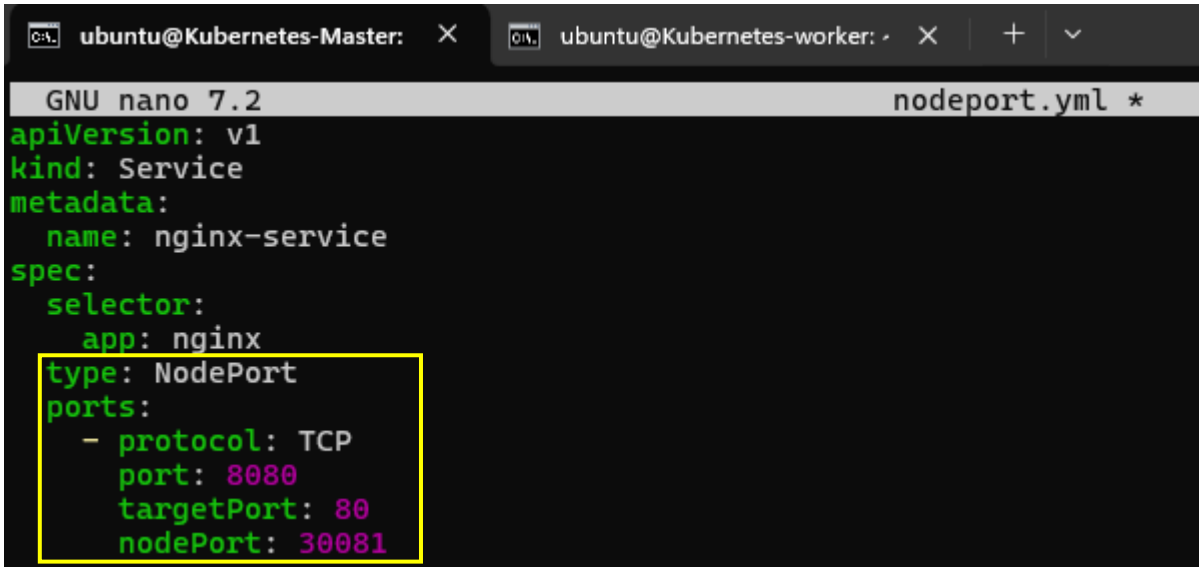
NodePort: Makes the service accessible on all nodes through a single port, exposing it externally via firewall configuration on the clfuster's nodes.



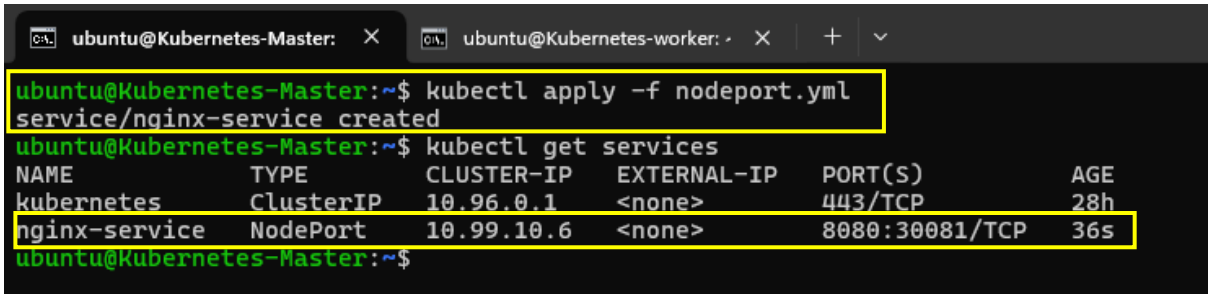
17. Copy the data from deployment.yml and edit it

```
ubuntu@Kubernetes-Master:~$ cat deployment.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx-container
          image: nginx:1.7.5
          ports:
            - containerPort: 80
```

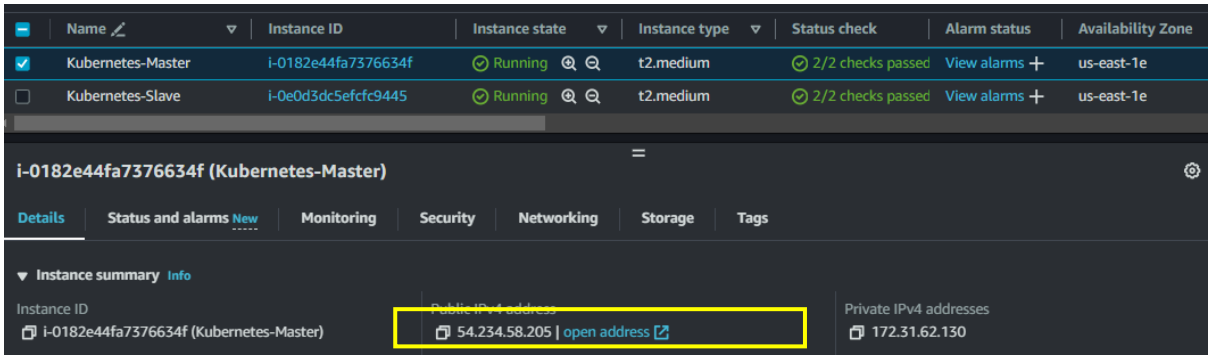
18. observe in the ports part



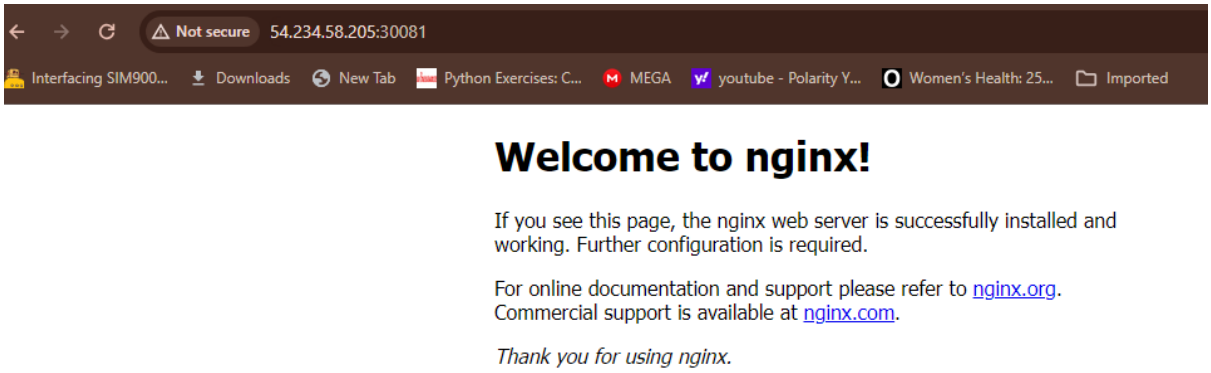
19. Apply the nodeport.yml and its created and launched the nginx server in the browser and port no 8080 and nodeport 30081



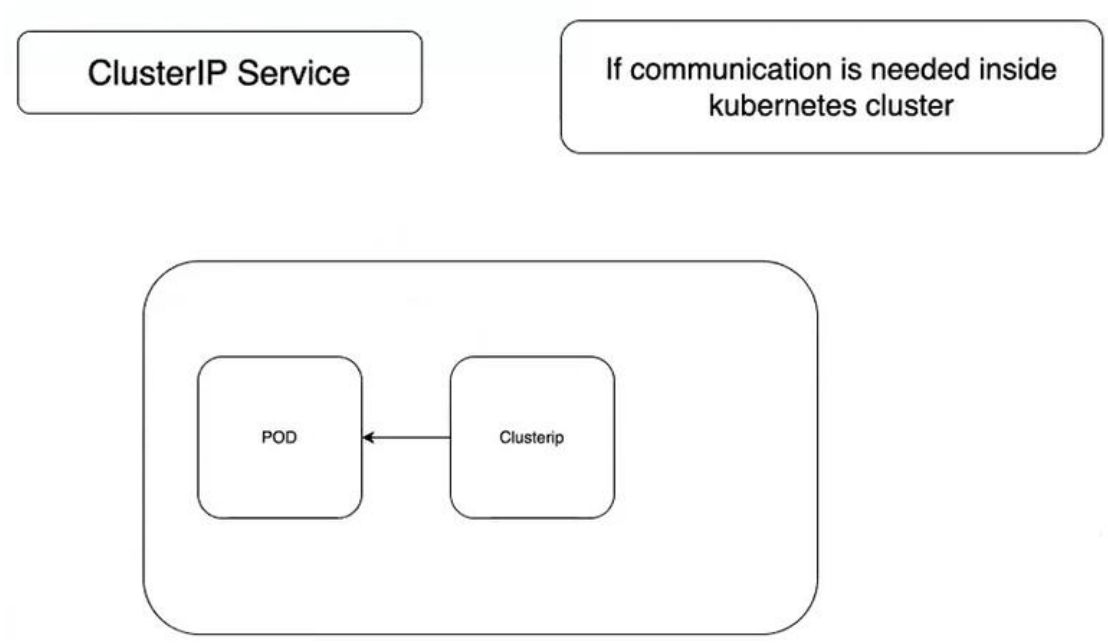
20. Copy the Kubernetes-Master Public-IP



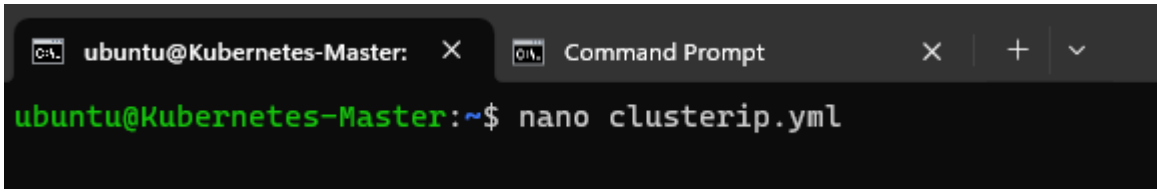
21. We can see the default nginx page



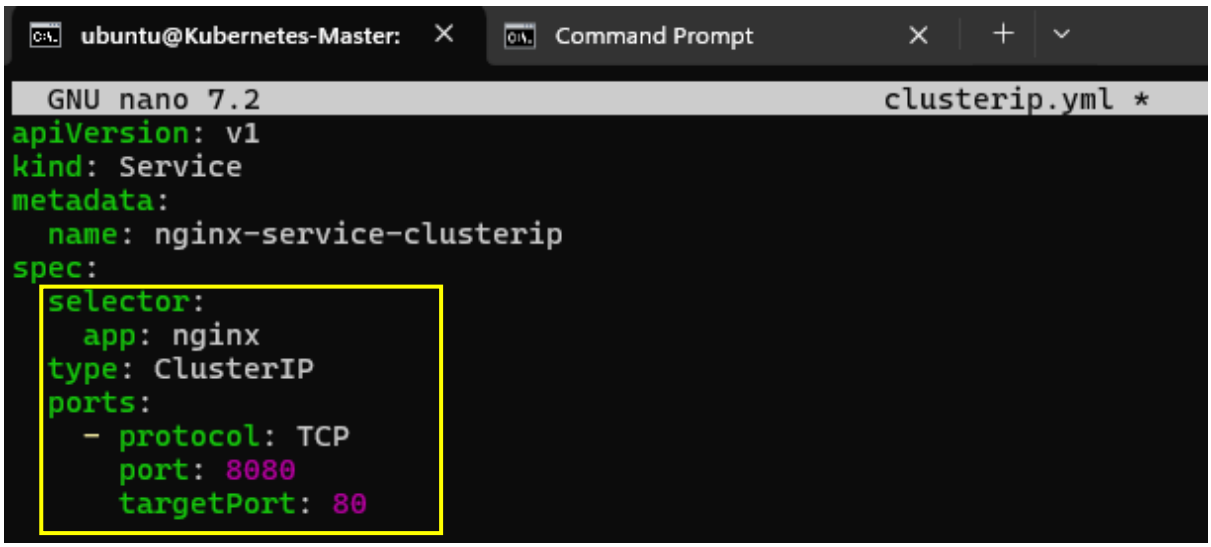
ClusterIP: Creates a service accessible only from within the Kubernetes cluster using a cluster-internal IP address.



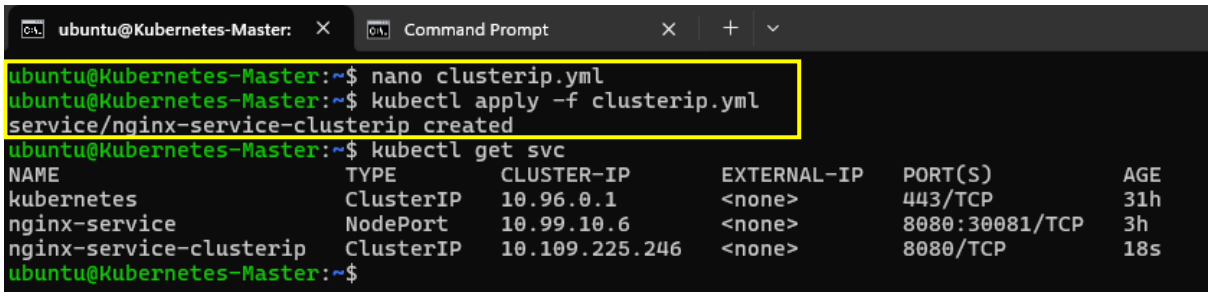
22. Creating Clusterip.yml



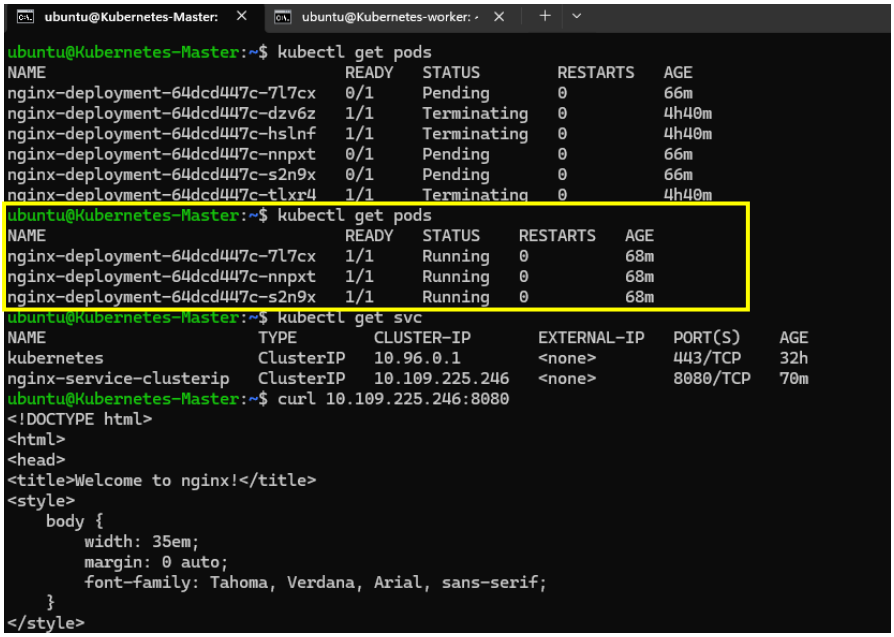
23. in the selector this time clusterIP and there is no clusteripport like nodeport this is only for internal connection.



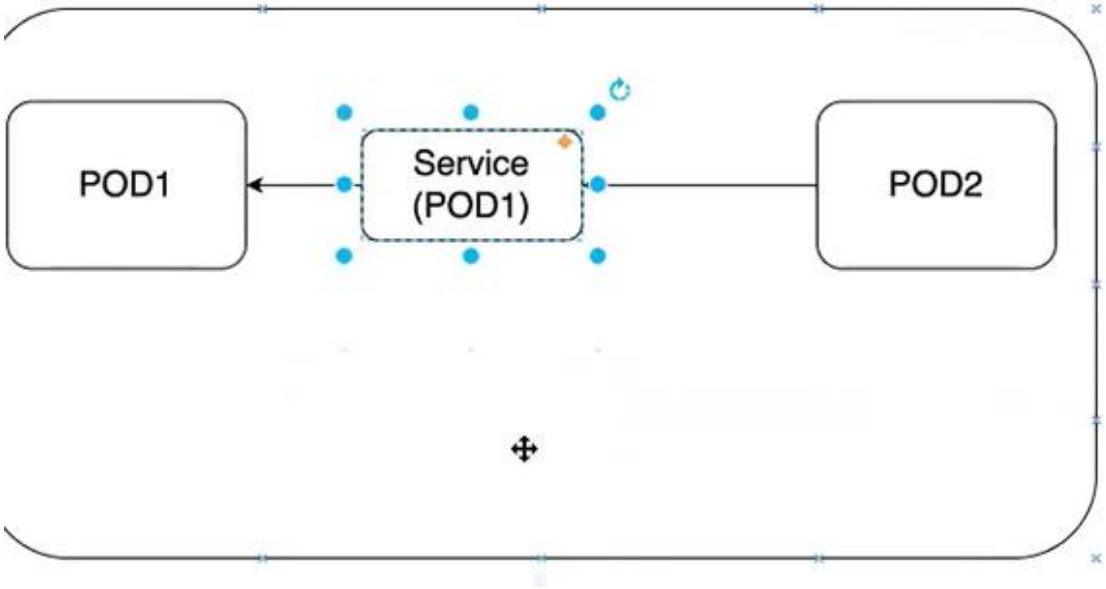
24. We can see the nginx-service-clusterip



25. its pending and terminating because I didn't launch Kubernetes-worker node and launch its terminated and running and we can see the curl 10.109.225.246:8080.

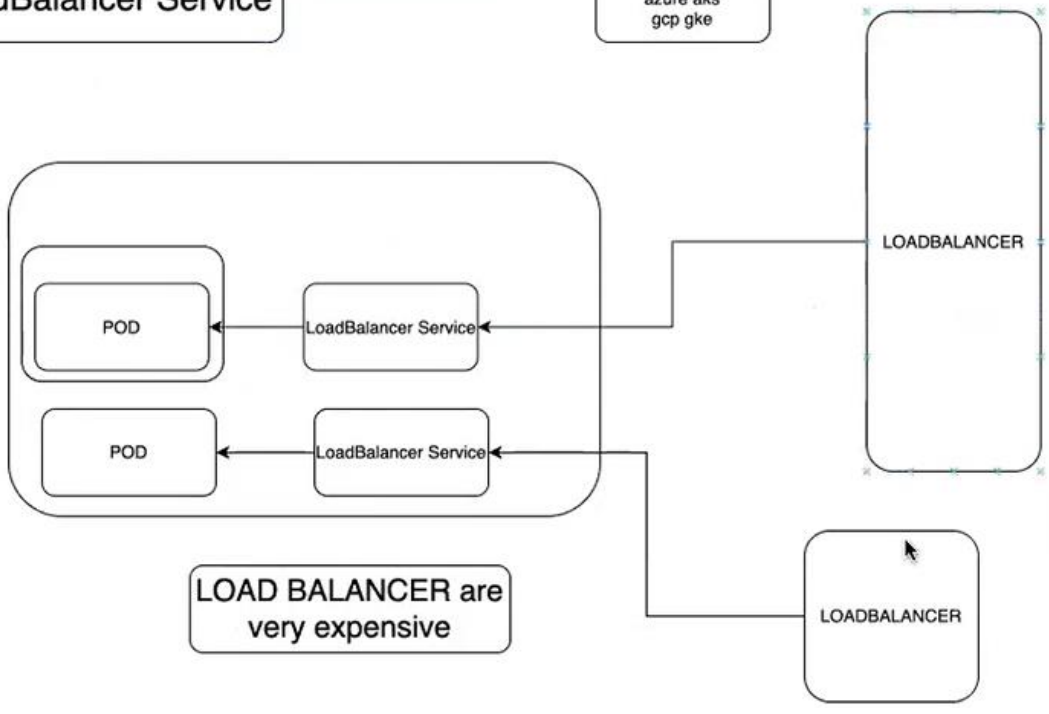


kubernetes cluster

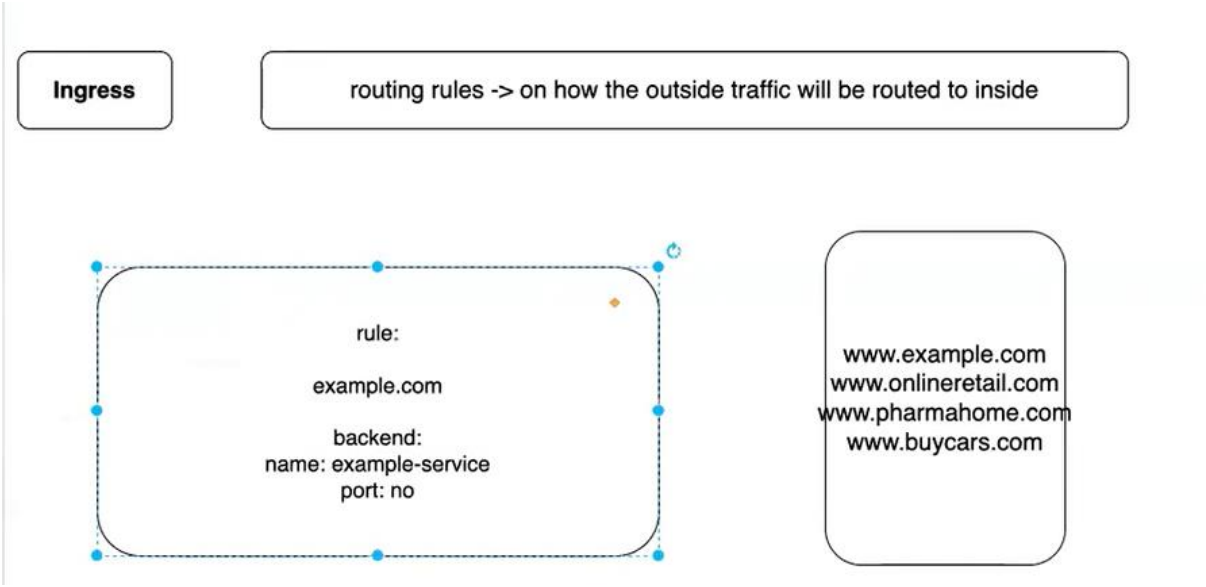


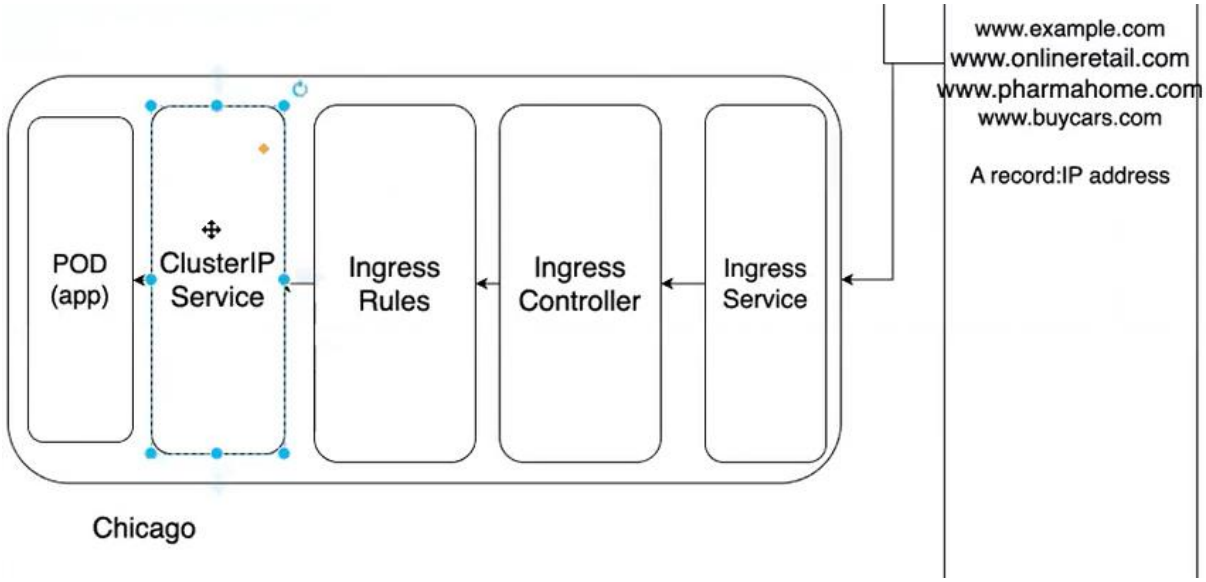
LoadBalancer Service

aws eks
azure aks
gcp gke



LOAD BALANCER are
very expensive





Kubernetes Ingress

Configuration

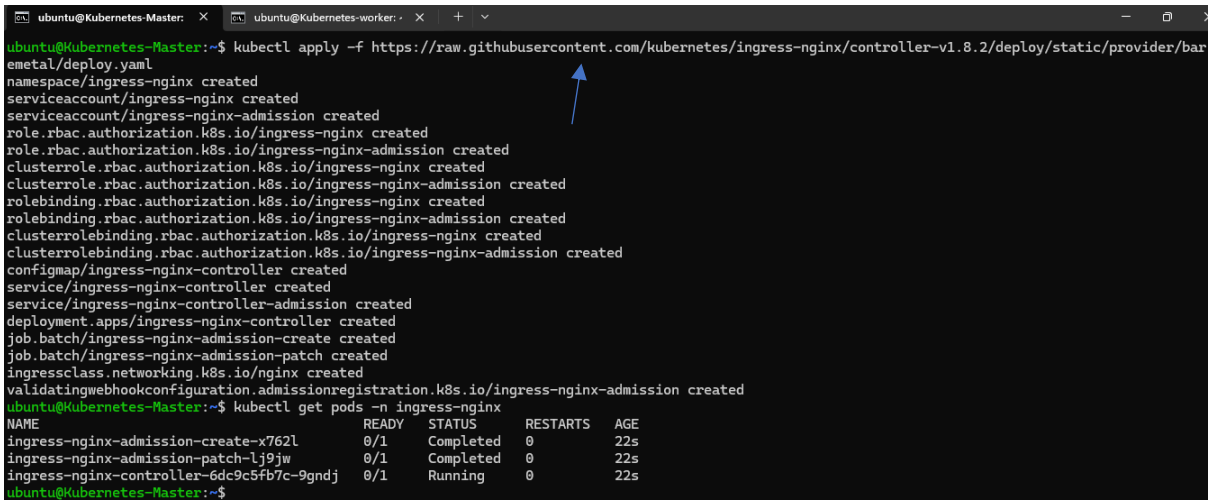
```
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.8.2/dep
loy/static/provider/baremetal/deploy.yaml
```

Ingress: In Kubernetes, ingress acts as an **entry point for external traffic, routing it to internal services based on defined rules**.

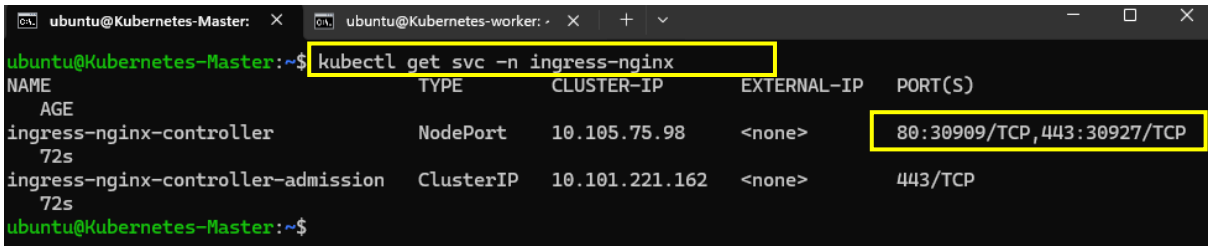
26. Deploying the Ingress

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controllerv1.8.2/deploy/static/provider/baremetal/deploy.yaml and

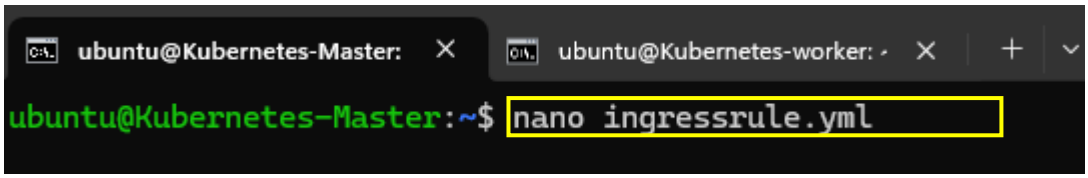
kubectl get pods -n ingress-nginx



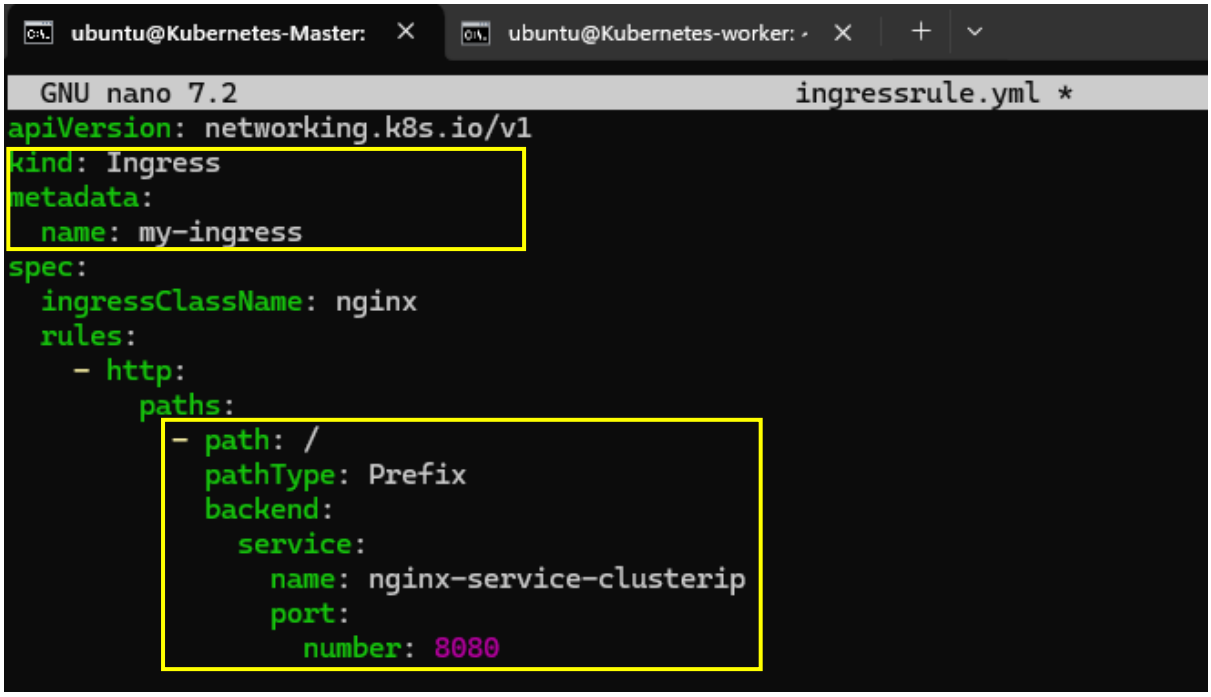
27. kubectl get svc -n ingress-nginx



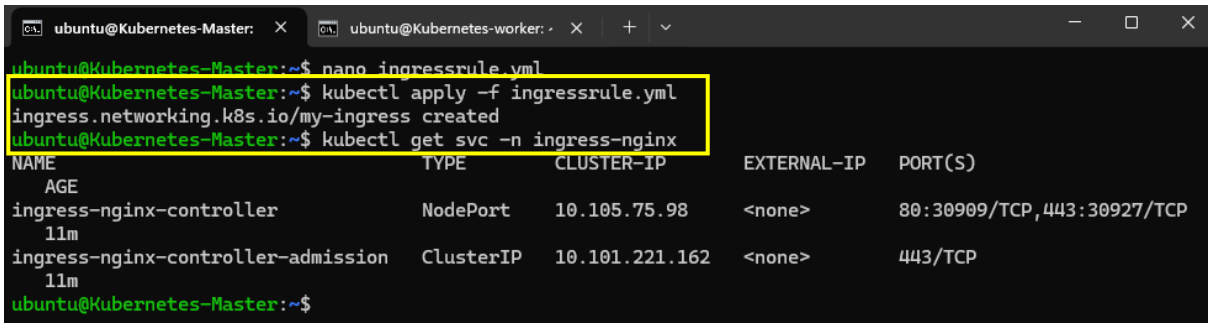
28. writing ingressrule.yml



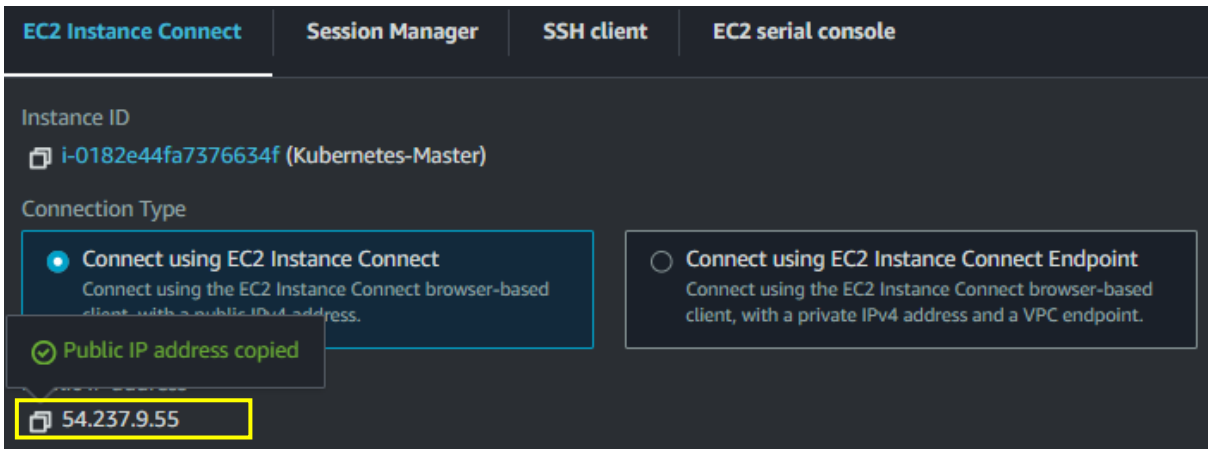
29. Observer the things



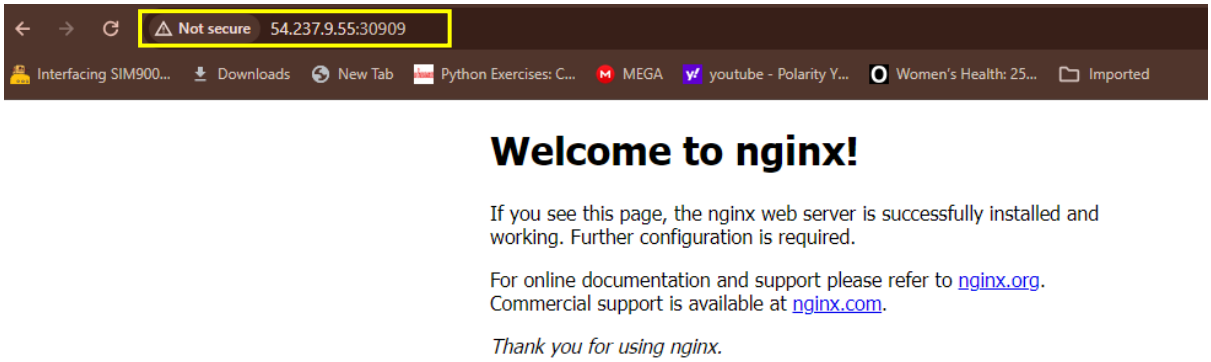
30. deploy ingressrule.yml



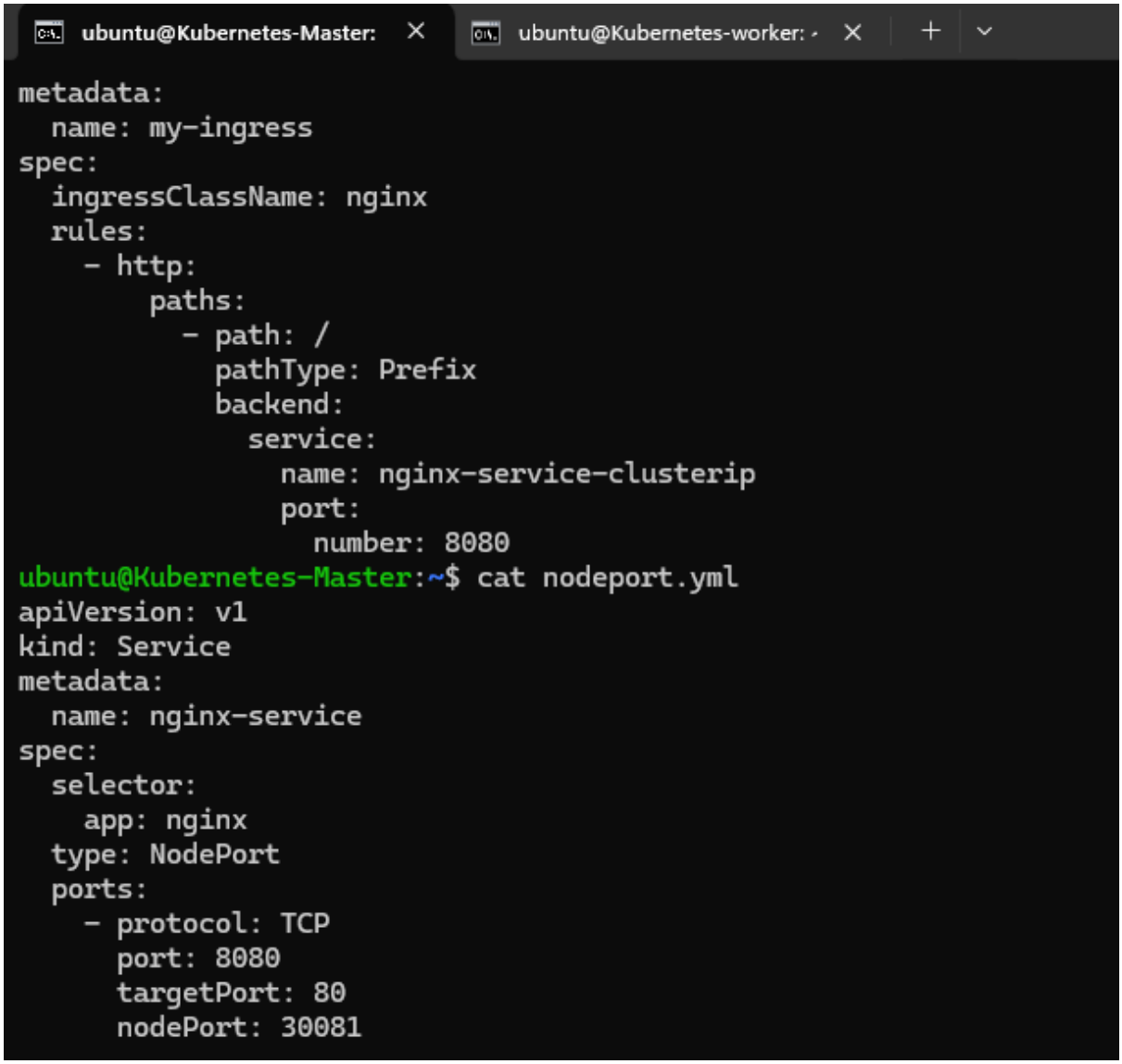
31. copy the public ip



32. now we can see the nginx server with the port no of 30909



33. observe the path of ingress and compare with nodeport

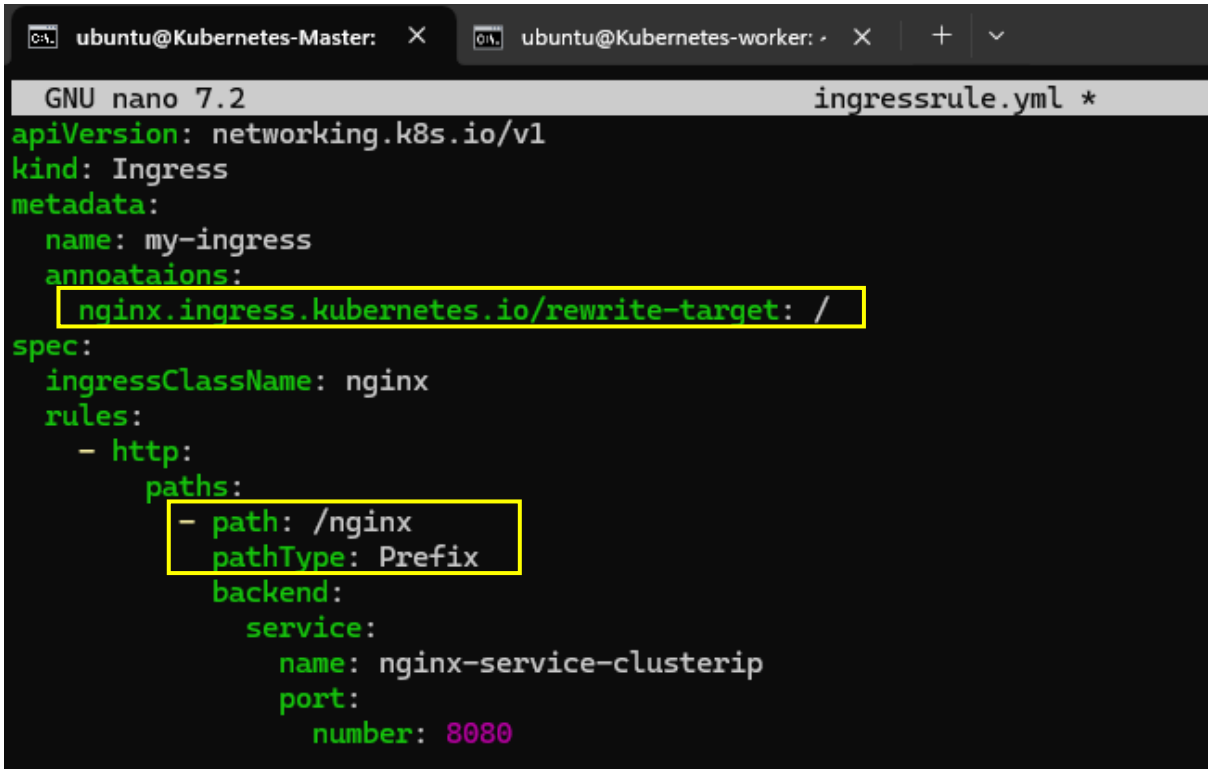


34. Check the ingress pods and checking the logs

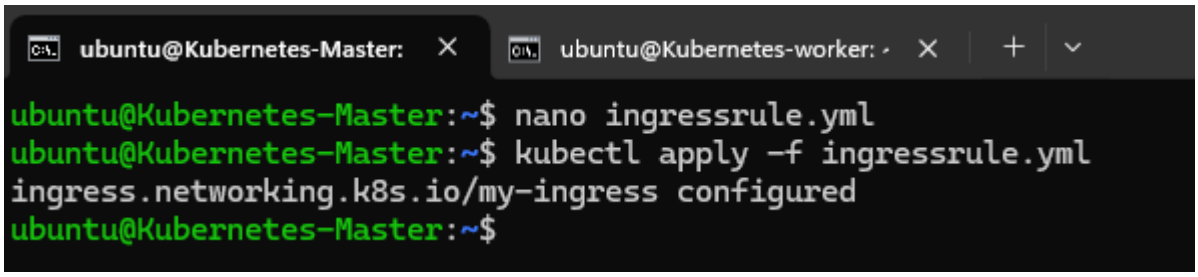
```
ubuntu@Kubernetes-Master:~$
ubuntu@Kubernetes-Master:~$ kubectl get pods -n ingress-nginx
NAME                                   READY   STATUS      RESTARTS   AGE
ingress-nginx-admission-create-x762l   0/1     Completed   0          17m
ingress-nginx-admission-patch-lj9jw    0/1     Completed   0          17m
ingress-nginx-controller-6dc9c5fb7c-9gndj   1/1     Running     0          17m
ubuntu@Kubernetes-Master:~$ kubectl logs ingress-nginx-controller-6dc9c5fb7c-9gndj -n ingress-nginx
-------------------------------------------------------------------------------
NGINX Ingress controller
  Release:       v1.8.1
  Build:         dc88dce9ea5e700f3301d16f971fa17c6cfe757d
  Repository:    https://github.com/kubernetes/ingress-nginx
  nginx version: nginx/1.21.6

-------------------------------------------------------------------------------

W0530 15:52:34.726477       7 client_config.go:618] Neither --kubeconfig nor --master was specified.  Using the inClusterConfig.  This might not work.
I0530 15:52:34.726793       7 main.go:209] "Creating API client" host="https://10.96.0.1:443"
I0530 15:52:34.734964       7 main.go:253] "Running in Kubernetes cluster" major="1" minor="28" git="v1.28.10" state="clean" commit="21be1d76a90bc00e2b0f6676a664bdf097224155" platform="linux/amd64"
I0530 15:52:35.182147       7 main.go:104] "SSL fake certificate created" file="/etc/ingress-controller/ssl/default-fake-certificate.pem"
I0530 15:52:35.200002       7 ssl.go:533] "loading tls certificate" path="/usr/local/certificates/cert" key="/usr/local/certificates/key"
I0530 15:52:35.211110       7 nginx.go:261] "Starting NGINX Ingress controller"
I0530 15:52:35.217923       7 event.go:285] Event(v1.ObjectReference{Kind:"ConfigMap", Namespace:"ingress-nginx", Name:"ingress-nginx-controller", UID:"fe675d76-25b9-4214-bd29-07437a1123a0", APIVersion:"v1", ResourceVersion:"24791", FieldPath:""}): type: 'Normal' reason: 'CREATE' ConfigMap ingress-nginx/ingress-nginx-controller
I0530 15:52:36.413126       7 nginx.go:304] "Starting NGINX process"
I0530 15:52:36.413364       7 leaderelection.go:248] attempting to acquire leader lease ingress-nginx/ingress-nginx-leader...
I0530 15:52:36.413630       7 nginx.go:324] "Starting validation webhook" address=":8443" certPath="/usr/local/certificates/cert" keyPath="/usr/local/certificates/key"
I0530 15:52:36.414108       7 controller.go:190] "Configuration changes detected, backend reload required"
I0530 15:52:36.422143       7 leaderelection.go:258] successfully acquired lease ingress-nginx/ingress-nginx-leader
I0530 15:52:36.422356       7 status.go:84] "New leader elected" identity="ingress-nginx-controller-6dc9c5fb7c-9gndj"
I0530 15:52:36.429184       7 status.go:215] "POD is not ready" pod="ingress-nginx/ingress-nginx-controller-6dc9c5fb7c-9gndj" node="kubernetes-worke
```

```
I0530 15:52:36.429184       7 status.go:215] "POD is not ready" pod="ingress-nginx/ingress-nginx-controller-6dc9c5fb7c-9gndj" node="kubernetes-worke
r"
I0530 15:52:36.498817       7 controller.go:207] "Backend successfully reloaded"
I0530 15:52:36.499460       7 controller.go:218] "Initial sync, sleeping for 1 second"
I0530 15:52:36.499627       7 event.go:285] Event(v1.ObjectReference{Kind:"Pod", Namespace:"ingress-nginx", Name:"ingress-nginx-controller-6dc9c5fb7
c-9gndj", UID:"1a3758f9-718d-411b-bc40-7b25f18b2d87", APIVersion:"v1", ResourceVersion:"24826", FieldPath:""}): type: 'Normal' reason: 'RELOAD' NGIN
X reload triggered due to a change in configuration
I0530 16:03:21.686521       7 admission.go:149] processed ingress via admission controller {testedIngressLength:1 testedIngressTime:0.044s rendering
IngressLength:1 renderingIngressTime:0s admissionTime:14.3kBs testedConfigurationSize:0.044}
I0530 16:03:21.686563       7 main.go:110] "successfully validated configuration, accepting" ingress="default/my-ingress"
I0530 16:03:21.695557       7 store.go:432] "Found valid IngressClass" ingress="default/my-ingress" ingressclass="nginx"
I0530 16:03:21.696032       7 controller.go:190] "Configuration changes detected, backend reload required"
I0530 16:03:21.696052       7 event.go:285] Event(v1.ObjectReference{Kind:"Ingress", Namespace:"default", Name:"my-ingress", UID:"62a17fb0-3b89-490e
-820c-857b0952baf8", APIVersion:"networking.k8s.io/v1", ResourceVersion:"25883", FieldPath:""}): type: 'Normal' reason: 'Sync' Scheduled for sync
I0530 16:03:21.778603       7 controller.go:207] "Backend successfully reloaded"
I0530 16:03:21.779080       7 event.go:285] Event(v1.ObjectReference{Kind:"Pod", Namespace:"ingress-nginx", Name:"ingress-nginx-controller-6dc9c5fb7
c-9gndj", UID:"1a3758f9-718d-411b-bc40-7b25f18b2d87", APIVersion:"v1", ResourceVersion:"24826", FieldPath:""}): type: 'Normal' reason: 'RELOAD' NGIN
X reload triggered due to a change in configuration
I0530 16:03:36.432655       7 status.go:300] "updating Ingress status" namespace="default" ingress="my-ingress" currentValue=null newValue=[{"ip":"1
72.31.53.219"}]
I0530 16:03:36.438917       7 event.go:285] Event(v1.ObjectReference{Kind:"Ingress", Namespace:"default", Name:"my-ingress", UID:"62a17fb0-3b89-490e
-820c-857b0952baf8", APIVersion:"networking.k8s.io/v1", ResourceVersion:"25909", FieldPath:""}): type: 'Normal' reason: 'Sync' Scheduled for sync
10.244.0.0 - - [30/May/2024:16:05:32 +0000] "\x16\x03\x01\x06\xE0\x01\x00\x06\xDC\x03\x03\x18<\xB7D~k%jU\xA6\xF2_\xA50z\x12\xA8!\xCF$w\xF8\xBD\xAC}oT+\x12\x93\x04_ pF\xB7\xEC\xA7xV\xD5\x95\x07s]\xE15\x9B\x7Fv\xF3\xB1\x11|\x9Bj\x8ACo\xECw\x80a\x01\xDB\x00 \xBA\xBA\x13\x01\x13\x02\x13\x03\xC0+\xC0/\xC0,\xC00\xCC\xA9\xCC\xA8\xC0\x13\xC0\x14\x00\x9C\x00\x9D\x00/\x005\x01\x00\x06sZZ\x00\x00\xFE" 400 150 "-" "-" 0 0.492 [] [] - - - - ff008d938ced739b0485d2f30d0c29a4
10.244.0.0 - - [30/May/2024:16:05:32 +0000] "\x16\x03\x01\x07\x00\x01\x00\x06\xFC\x03\x03<\xDB\x81\xA5YA\xFF\xD2&\xD1\xE5@Ib\x89\x11\xF6\xA6\x14\xA9\x09\xE9\xE8I\xDEVI\x08\x00\xD9\x18\xF2 \x8E\xA6\xDBz\xC8.\xFC\x08.\xE2\x05\x88\x96\xA2\xD7j\x8E\xD6\xF2\xA9/\x92M\xE4\xBD\xEF\xE4\x1A6\xEC\xCD\x83\x00 \x8A\x8A\x13\x01\x13\x02\x13\x03\xC0+\xC0/\xC0,\xC00\xCC\xA9\xCC\xA8\xC0\x13\xC0\x14\x00\x9C\x00\x9D\x00/\x005\x01\x00\x06\x93\x8A\x8A\x00\x00\xFE" 400 150 "-" "-" 0 0.233 [] [] - - - - e33fab4545216a1e968d4d9e0453d2d6
10.244.0.0 - - [30/May/2024:16:05:33 +0000] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.36" 432 0.002 [default-nginx-service-clusterip-8080] [] 10.244.1.30:80 612 0.002 200 b644f54c7fd807d223358853d0292583
10.244.0.0 - - [30/May/2024:16:05:33 +0000] "\x16\x03\x01\x06\xA0\x01\x00\x06\x9C\x03\x03f\x08\x8C\xE8\xD7h\x85\x9FV\x9Aoz\xE6\xC4e\x7F" 400 150 "-"
```

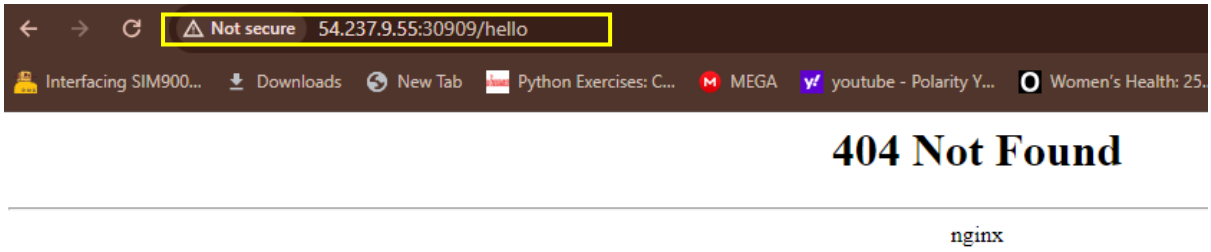
35. Now in the Ingress in the Path : /nginx we thought we are routing but its not.



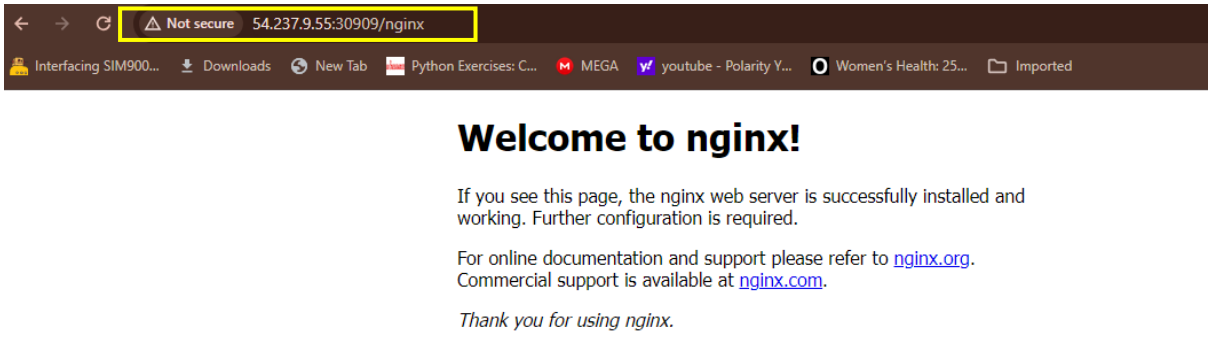
36. Apply it it will show configure



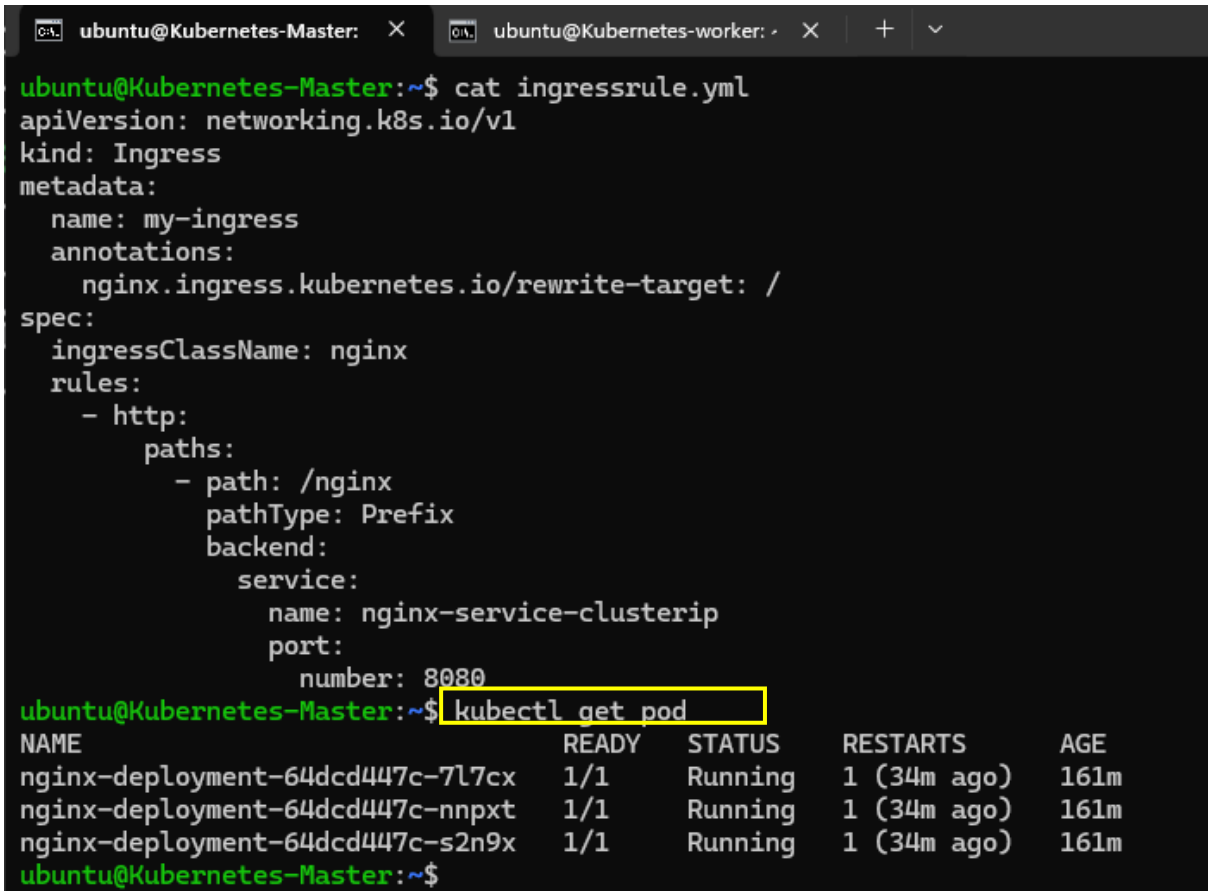
37. After ingress port no we enter /hello its show 404 error



38. but type /nginx it will show the welcome page



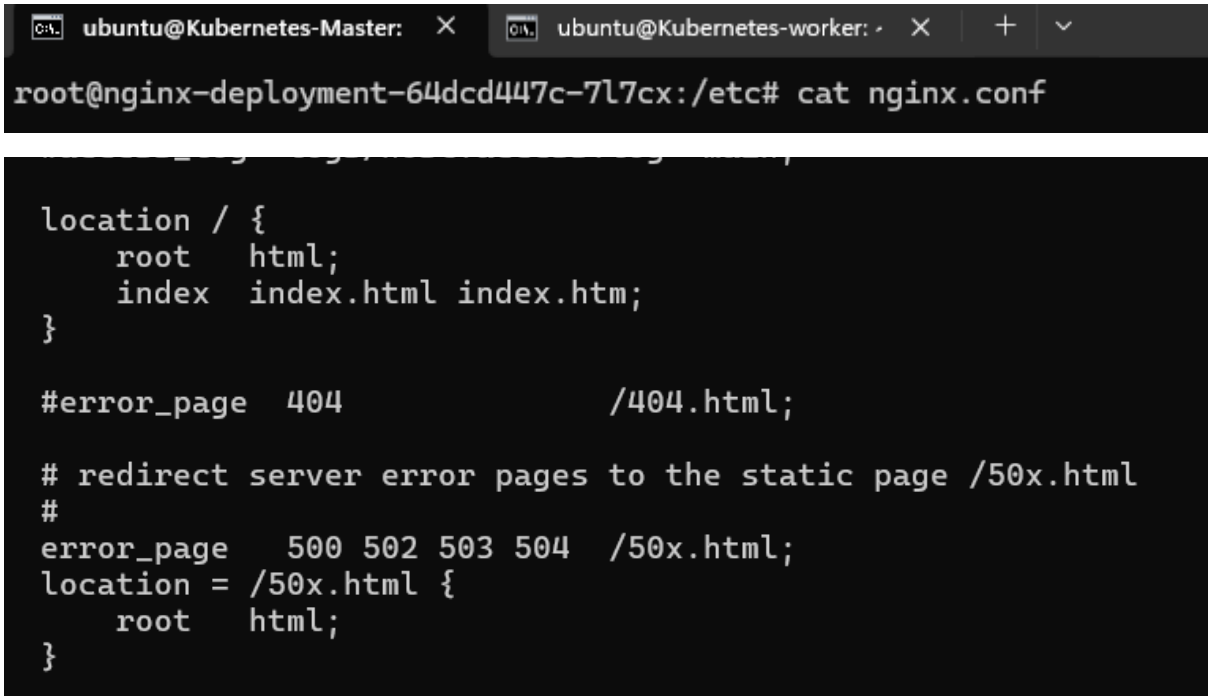
39. cat and get current pods



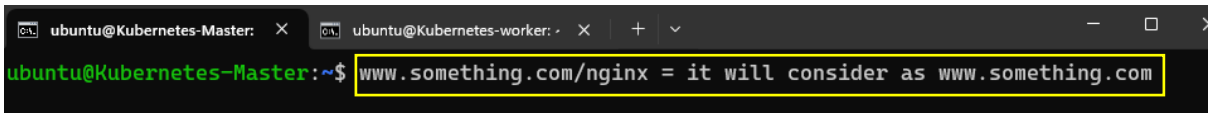
40. default html actually thing is what ever you /typeinfo it will show the default page its not routing

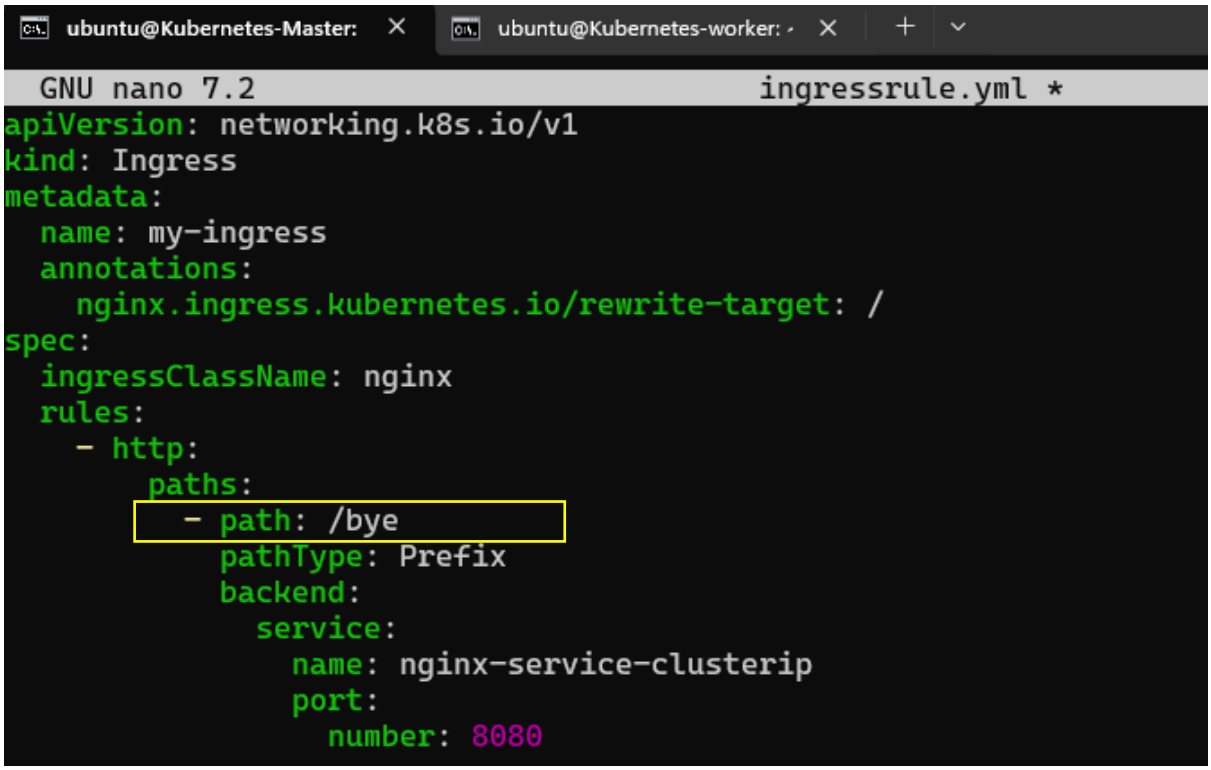
```
ubuntu@Kubernetes-Master:~$ kubectl exec -it pod/nginx-deployment-64dcd447c-7l7cx -- /bin/bash
root@nginx-deployment-64dcd447c-7l7cx:/usr/local/nginx/html# ls
50x.html  index.html
root@nginx-deployment-64dcd447c-7l7cx:/usr/local/nginx/html# cat index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@nginx-deployment-64dcd447c-7l7cx:/usr/local/nginx/html#
```

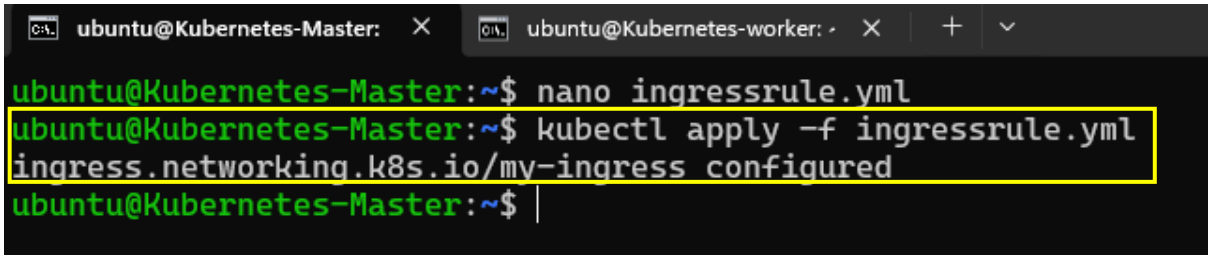
```
root@nginx-deployment-64dcd447c-7l7cx:/usr/local/nginx/html#
root@nginx-deployment-64dcd447c-7l7cx:/usr/local/nginx/html# cd ..
root@nginx-deployment-64dcd447c-7l7cx:/usr/local/nginx# cd ..
root@nginx-deployment-64dcd447c-7l7cx:/usr/local# ls
bin  etc  games  include  lib  man  nginx  sbin  share  src
root@nginx-deployment-64dcd447c-7l7cx:/usr/local# cd /etc
root@nginx-deployment-64dcd447c-7l7cx:/etc# ls
X11                     hostname            network             rpc
alternatives            hosts               networks            scgi_params
apt                     init                nginx.conf          scgi_params.default
bash.bashrc             init.d              nginx.conf.default  securetty
bash_completion.d       inittab             nsswitch.conf       security
bindresvport.blacklist  inputrc             opt                 selinux
cron.daily              insserv             os-release          services
debconf.conf            insserv.conf        pam.conf            sgml
debian_version          insserv.conf.d      pam.d               shadow
default                 iproute2            passwd              shadow-
dpkg                    issue               passwd-             shells
environment             issue.net           perl                skel
fastcgi.conf            kernel              profile             staff-group-for-usr-local
fastcgi.conf.default    koi-utf             profile.d           subgid
fastcgi_params          koi-win             protocols           subgid-
fastcgi_params.default  ld.so.cache         rc.local            subuid
fonts                   ld.so.conf          rc0.d               subuid-
fonts.conf.d            ld.so.conf.d        rc1.d               terminfo
fstab                   libaudit.conf       rc2.d               timezone
fstab.d                 localtime           rc3.d               ucf.conf
gai.conf                login.defs          rc4.d               uwsgi_params
group                   logrotate.d         rc5.d               uwsgi_params.default
group-                  mime.types          rc6.d               win-utf
gshadow                 mime.types.default  rcS.d               xml
gshadow-                mke2fs.conf         resolv.conf
host.conf               motd                rmt
root@nginx-deployment-64dcd447c-7l7cx:/etc#
```

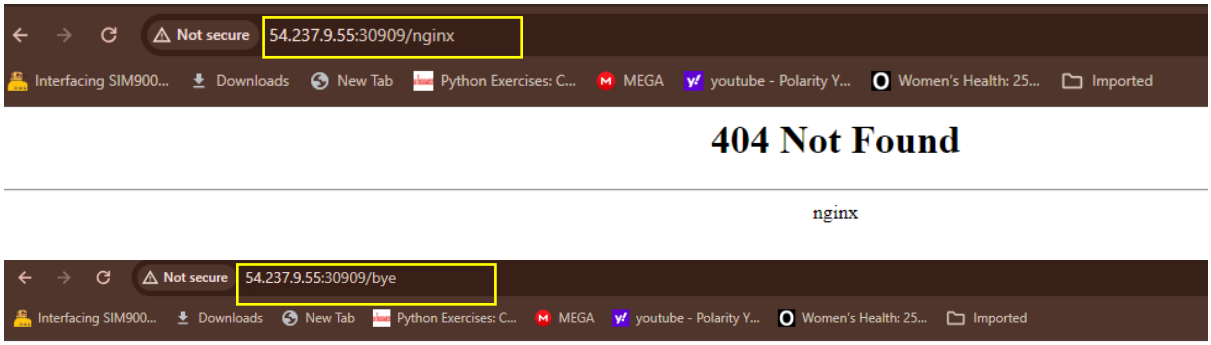


EXAMPLE OF INGRESS









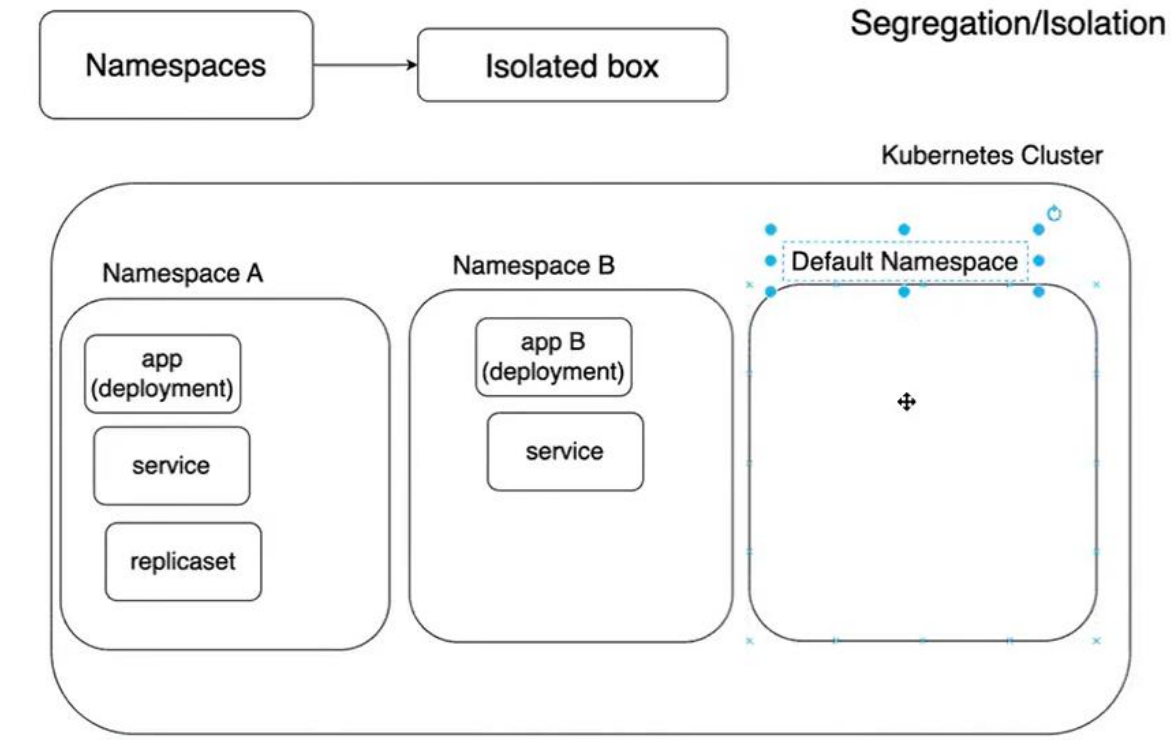
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Kubernetes-DAY4



NAME SPACE:

In Kubernetes, a namespace provides a mechanism for **isolating groups of resources within a single cluster**

```
ubuntu@Kubernetes-Master:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS      AGE
nginx-deployment-64dcd447c-7l7cx    1/1     Running   2 (84s ago)   20h
nginx-deployment-64dcd447c-nnpxt    1/1     Running   2 (84s ago)   20h
nginx-deployment-64dcd447c-s2n9x    1/1     Running   2 (84s ago)   20h
ubuntu@Kubernetes-Master:~$ kubectl get pod -n default
NAME                                READY   STATUS    RESTARTS      AGE
nginx-deployment-64dcd447c-7l7cx    1/1     Running   2 (2m13s ago)   20h
nginx-deployment-64dcd447c-nnpxt    1/1     Running   2 (2m13s ago)   20h
nginx-deployment-64dcd447c-s2n9x    1/1     Running   2 (2m13s ago)   20h
ubuntu@Kubernetes-Master:~$
```

Creating namespace named as web

```
ubuntu@Kubernetes-Master:~$ kubectl create ns web
namespace/web created
ubuntu@Kubernetes-Master:~$ ls
clusterip.yml  deployment.yml  ingressrule.yml  nodeport.yml  pod.yml  replicaset.yml
ubuntu@Kubernetes-Master:~$ cat pod.yml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
    - name: nginx-container
      image: nginx
      ports:
        - containerPort: 80

ubuntu@Kubernetes-Master:~$
```

Editing pod.yml

```
ubuntu@Kubernetes-Master:~$ nano pod.yml
```

In metadata namespace: web

```
 GNU nano 7.2                                                pod.yml *
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  namespace: web
spec:
  containers:
    - name: nginx-container
      image: nginx
      ports:
        - containerPort: 80
```

```
ubuntu@Kubernetes-Master:~$ cat pod.yml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  namespace: web
spec:
  containers:
    - name: nginx-container
      image: nginx
      ports:
        - containerPort: 80

ubuntu@Kubernetes-Master:~$
```

Created the pod

```
ubuntu@Kubernetes-Master:~$ kubectl apply -f pod.yml
pod/nginx-pod created
ubuntu@Kubernetes-Master:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS        AGE
nginx-deployment-64dcd447c-7l7cx    1/1     Running   2 (5m59s ago)   20h
nginx-deployment-64dcd447c-nnpxt    1/1     Running   2 (5m59s ago)   20h
nginx-deployment-64dcd447c-s2n9x    1/1     Running   2 (5m59s ago)   20h
ubuntu@Kubernetes-Master:~$
```

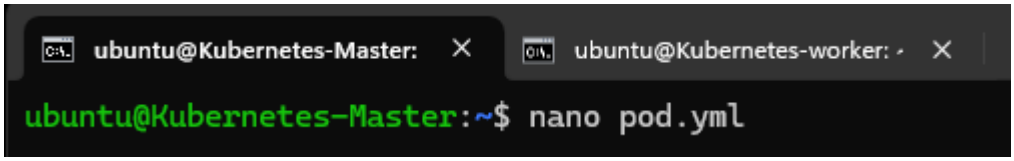
This is nginx-pod which in this we given namespace as web

```
ubuntu@Kubernetes-Master:~$ kubectl get pods -n web
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          55s
ubuntu@Kubernetes-Master:~$
```

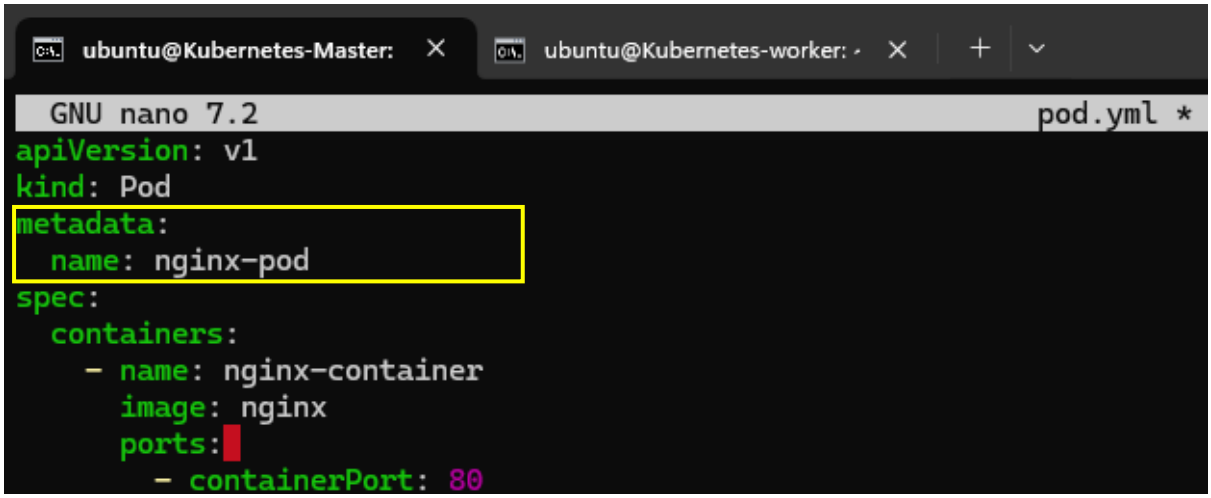
Deleted nginx-pod and showing no web namespace

```
ubuntu@Kubernetes-Master:~$ kubectl delete pod/nginx-pod -n web
pod "nginx-pod" deleted
ubuntu@Kubernetes-Master:~$ kubectl get pod -n web
No resources found in web namespace.
ubuntu@Kubernetes-Master:~$
```

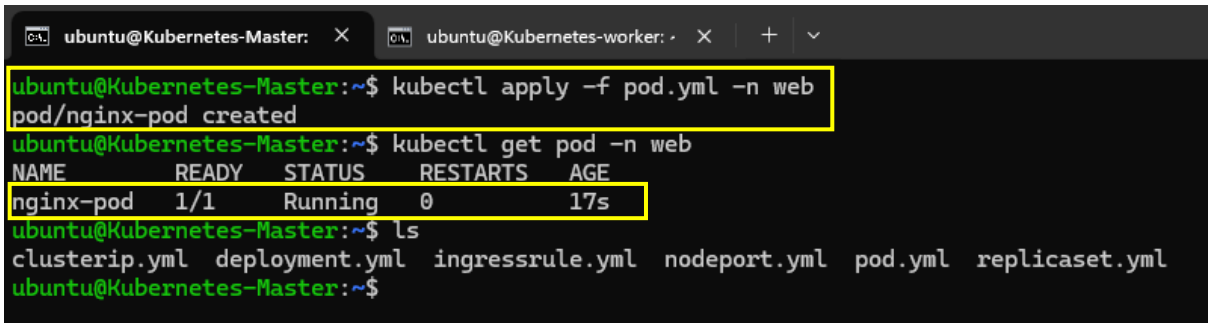
Go to edit pod



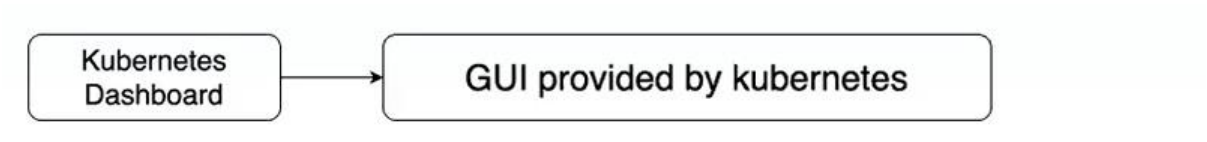
Remove namespace



Now apply -f pod.yml -n web and it will creates and kubectl get pod -n web and this command tell with the name of web, pod is running

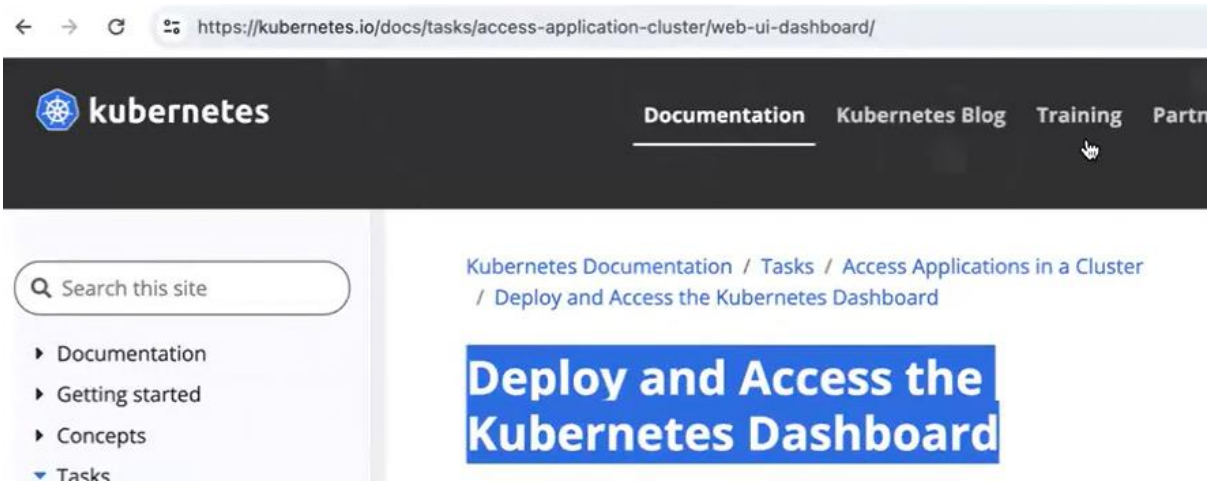


KUBERNETES Dashboard:



Step1: Deploy the kubernetes-dashboard manifest
Step2: made svc to nodeport (practice)
Step3: create serviceaccount
Step4: give permission to serviceaccount
Step5: create token for serviceaccount
Step6: paste the token in kubernetes-dashboard

Need to deploy



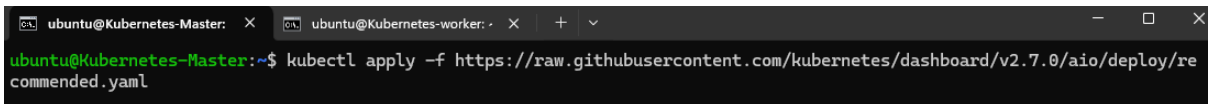
Kubernetes Dashboard

Configuration

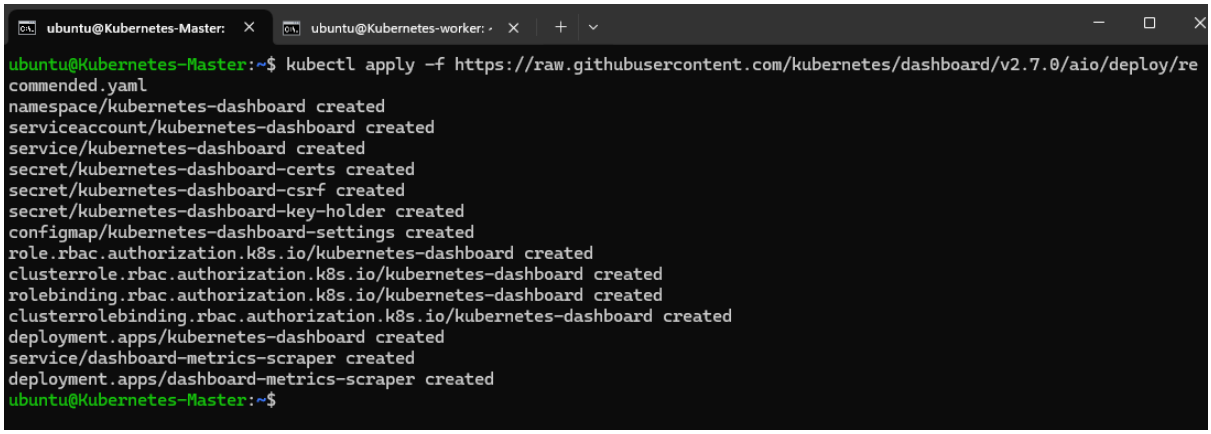
Step 1: Apply manifest
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml

https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml

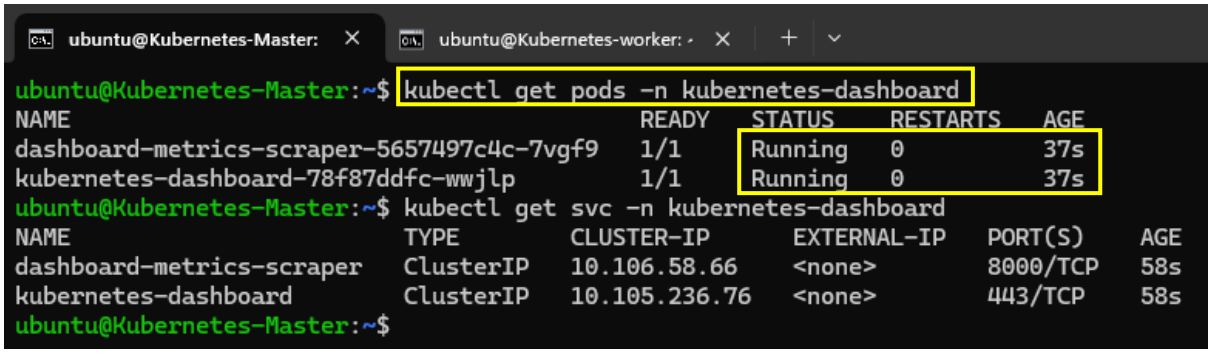
if you want to see the link, u can it will show script



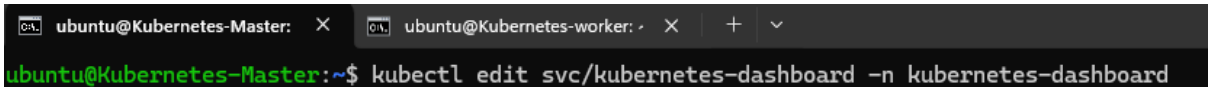
Follow it



Now we can see the pod with name of Kubernetes-dashboard we can see



Go edit this Kubernetes dashboard because it in clusterip, need to change nodeport



Scroll and edit

```
  selector:
    k8s-app: kubernetes-dashboard
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}

-- INSERT --
```

```
  selector:
    k8s-app: kubernetes-dashboard
  sessionAffinity: None
  type: NodePort
status:
  loadBalancer: {}
-- INSERT --
```

It will show edited and remember the 32473

```
ubuntu@Kubernetes-Master:~$ kubectl edit svc/kubernetes-dashboard -n kubernetes-dashboard
service/kubernetes-dashboard edited 35L, 1145B written
ubuntu@Kubernetes-Master:~$ kubectl get svc -n kubernetes-dashboard
NAME                        TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
dashboard-metrics-scraper   ClusterIP   10.106.58.66    <none>        8000/TCP         6m32s
kubernetes-dashboard        NodePort    10.105.236.76   <none>        443:32473/TCP    6m32s
ubuntu@Kubernetes-Master:~$
```

And copy the Kubernetes-master public ip

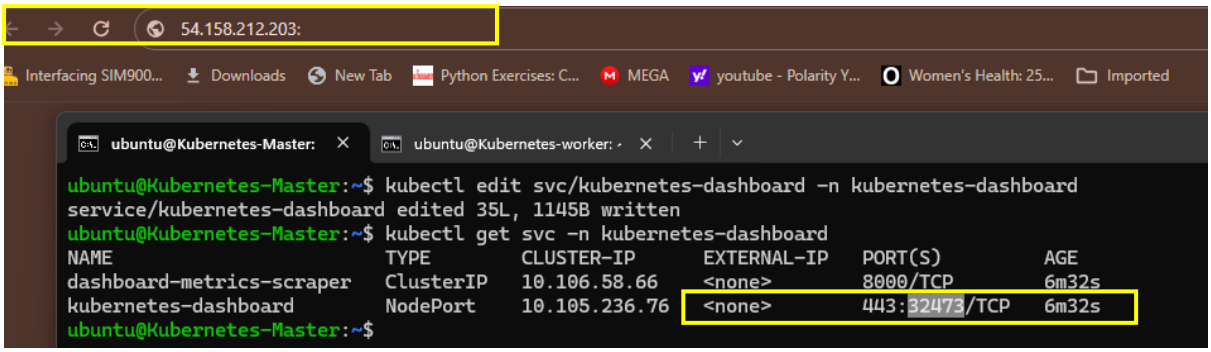
Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Kubernetes-Master	i-0182e44fa7376634f	Running	t2.medium	2/2 checks passed	View alarms
Kubernetes-Slave	i-0e0d3dc5efcfc9445	Running	t2.medium	2/2 checks passed	View alarms

i-0182e44fa7376634f (Kubernetes-Master)

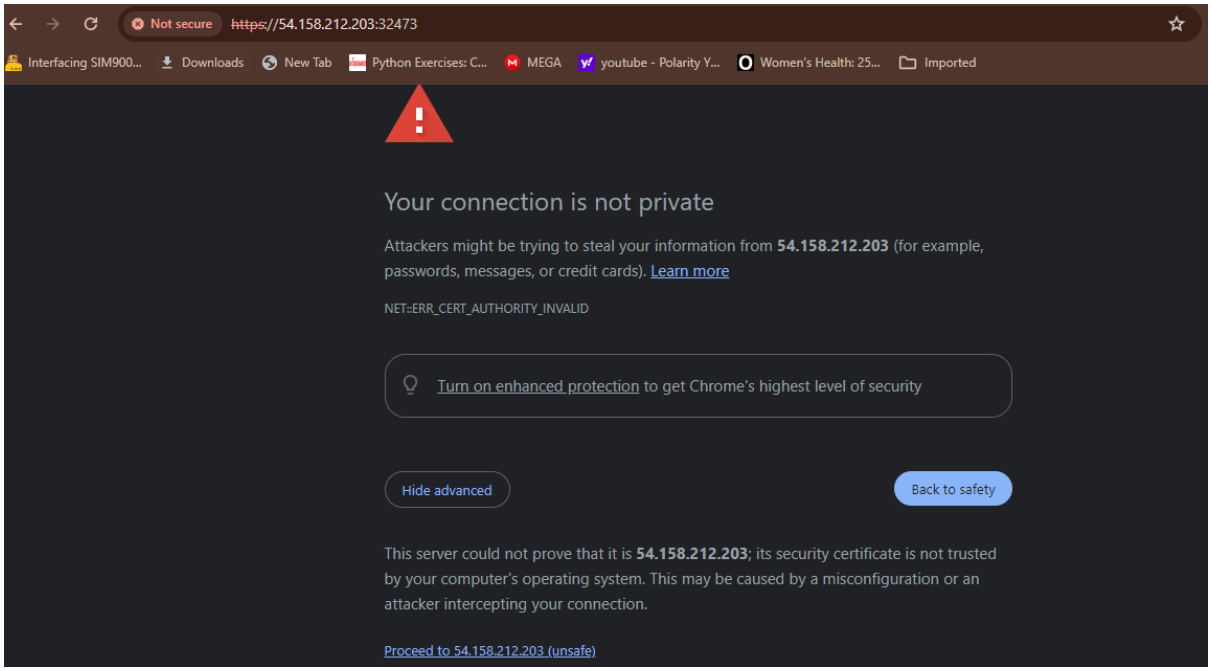
Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0182e44fa7376634f (Kubernetes-Master)	54.158.212.203 | open address	172.31.62.130

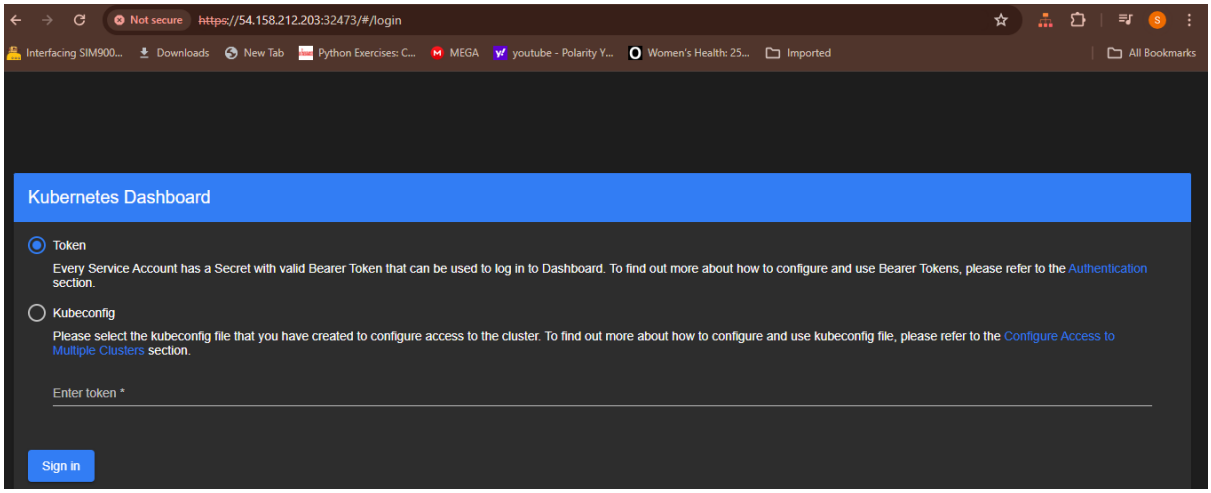
Copy it



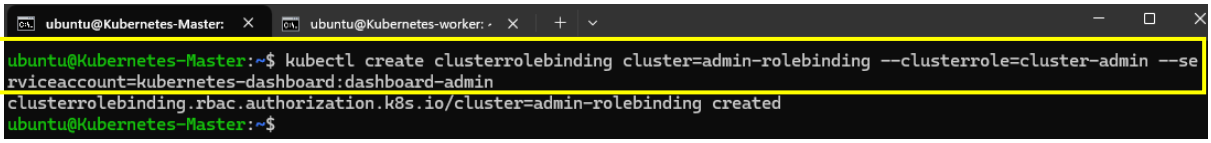
Paste it and to advance and hide advanced.



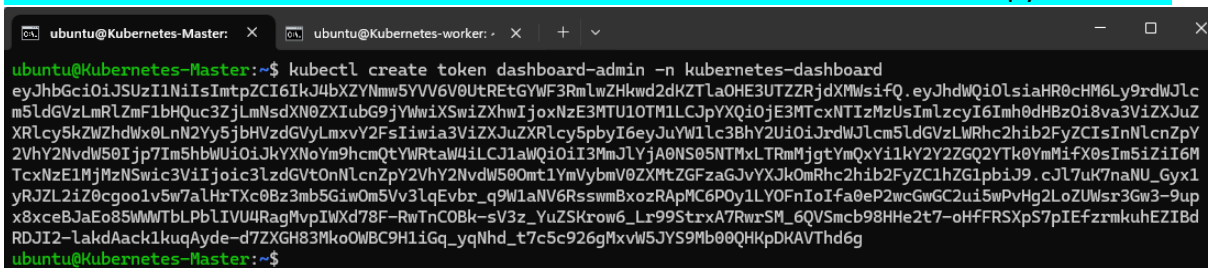
How to create a token?



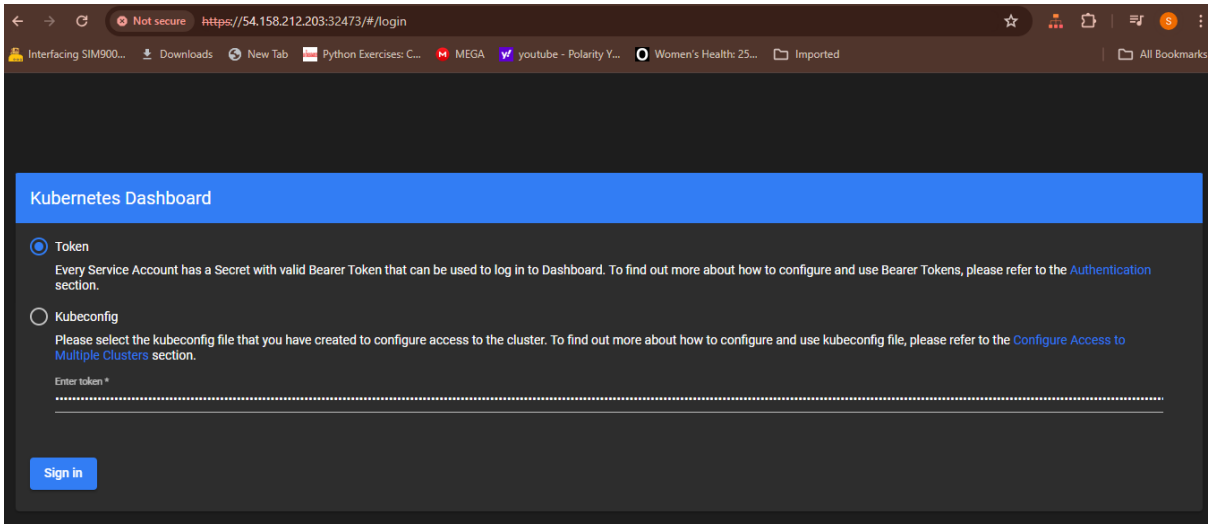
kubectl create clusterrolebinding cluster=admin-rolebinding --clusterrole=cluster-admin --serviceaccount=kubernetes-dashboard:dashboard-admin



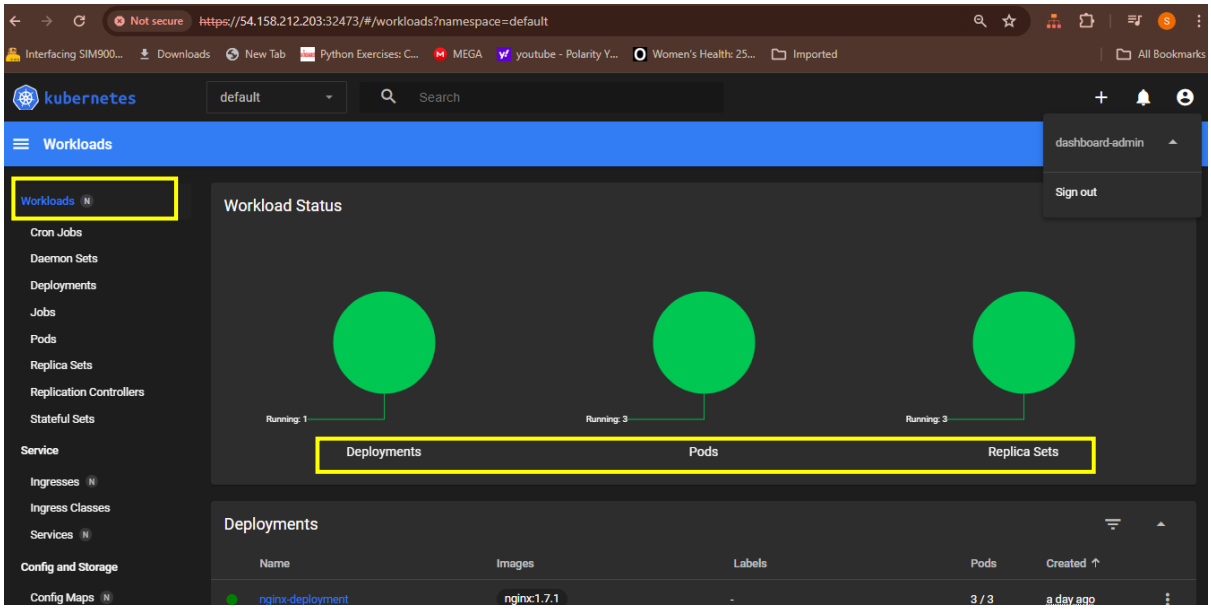
kubectl create serviceaccount dashboard-admin -n kubernetes-dashboard Copy the token



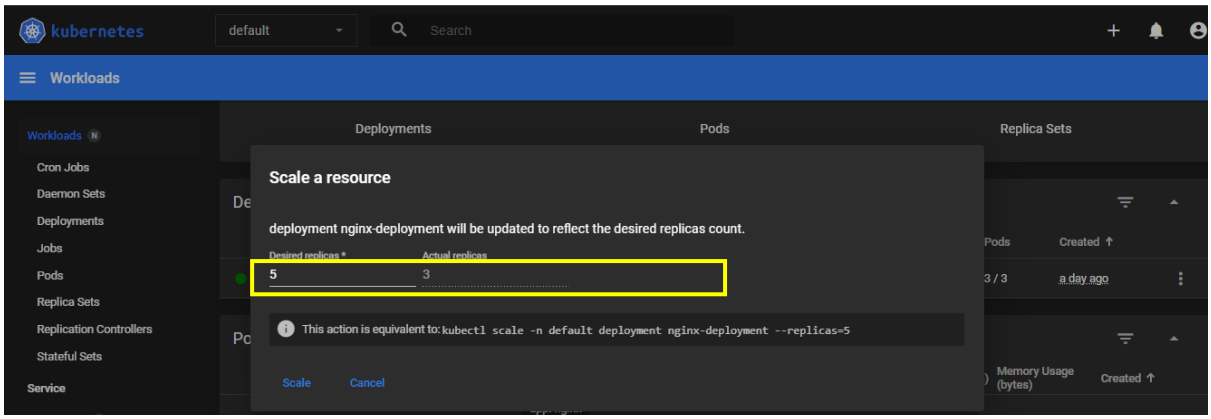
and paste it here



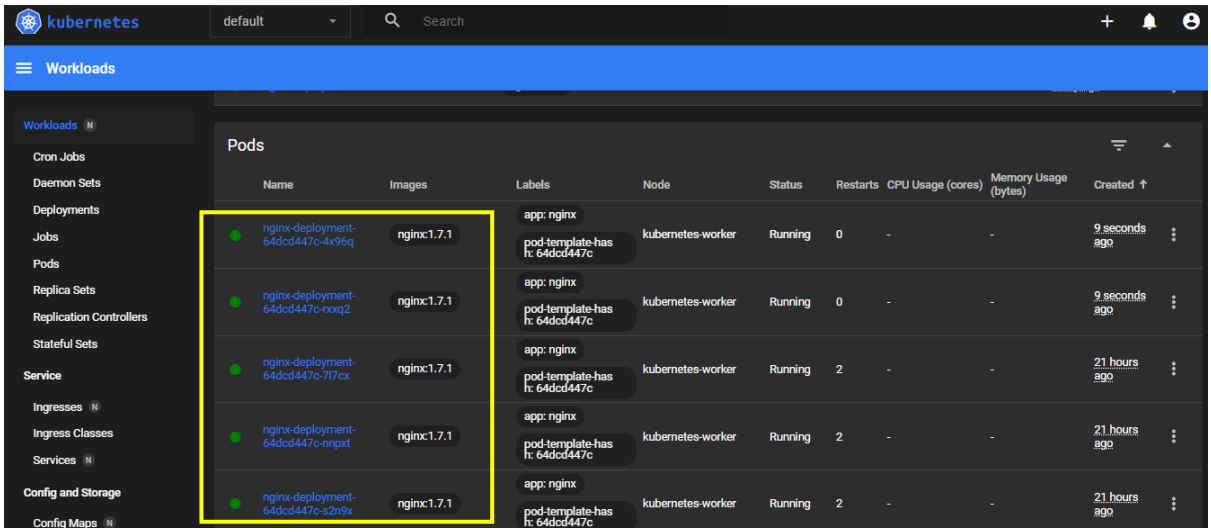
Successfully logged into Kubernetes dashboard

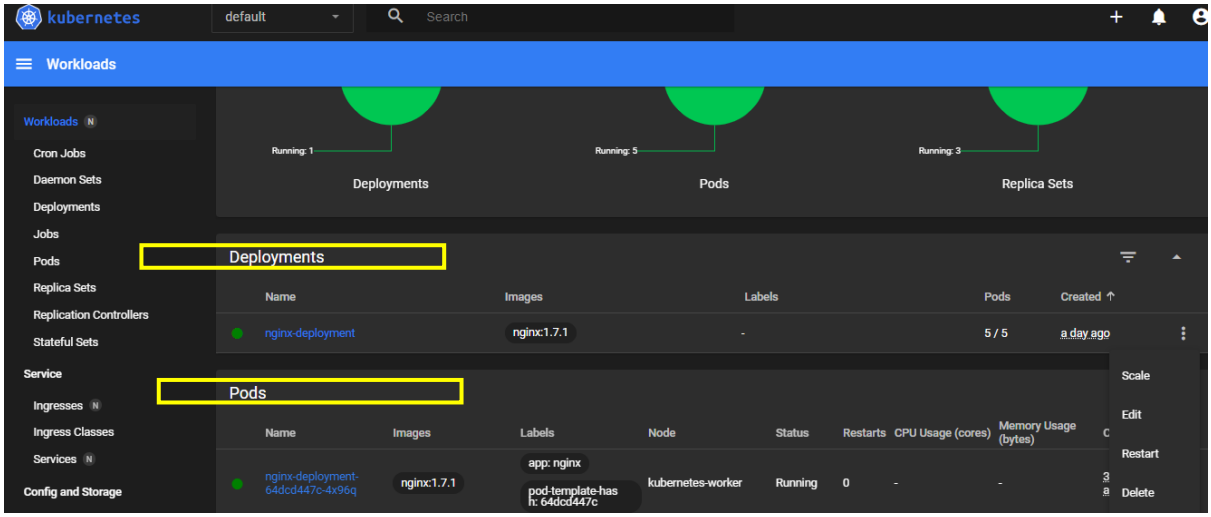


Edit the deployment and we creates 5 deployment

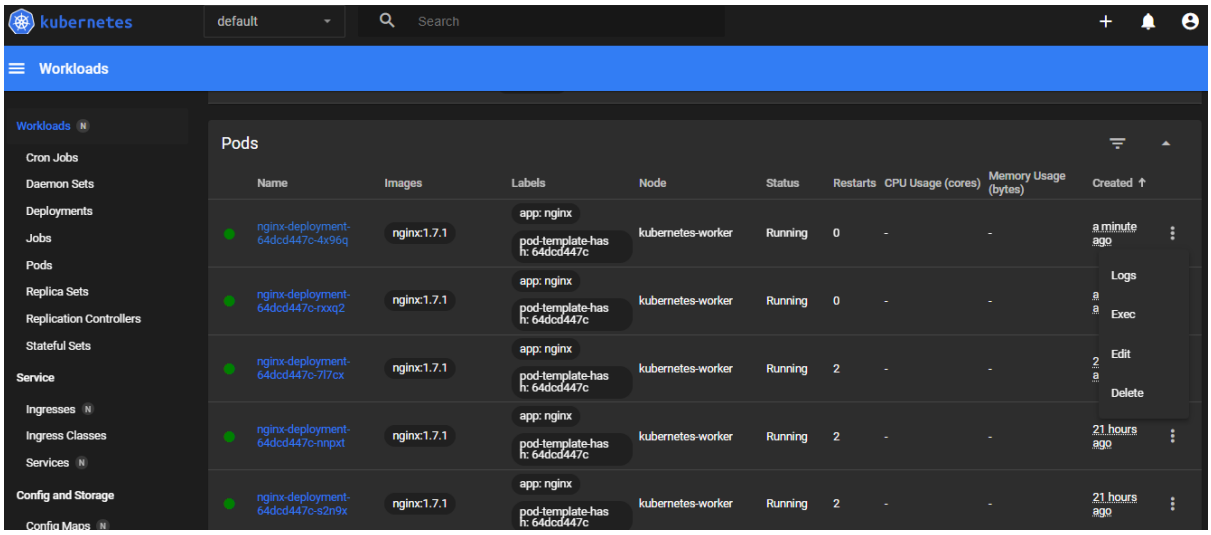


And its created

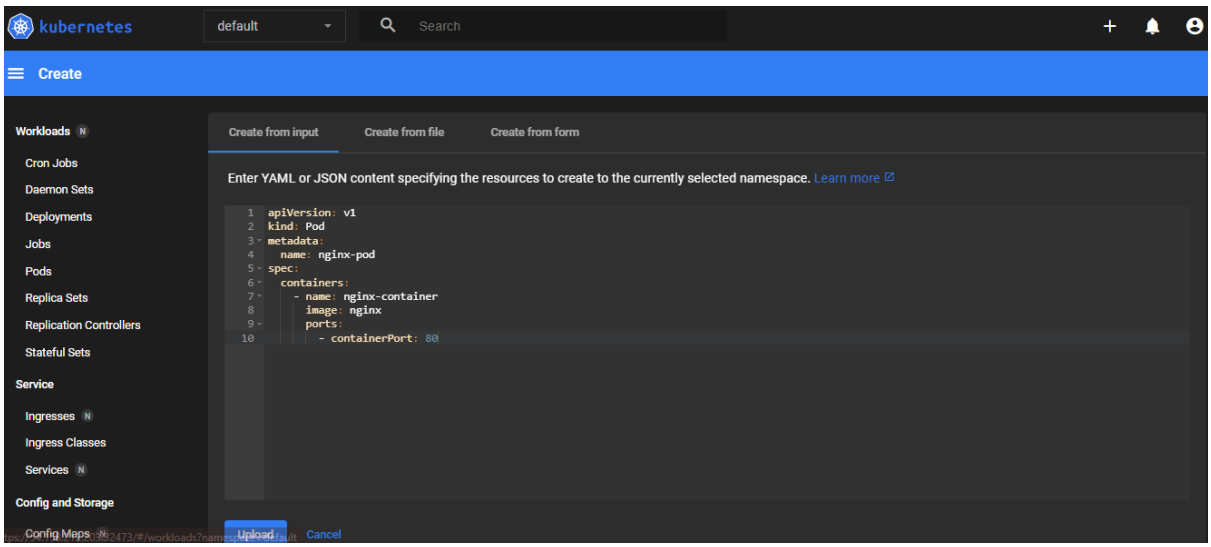




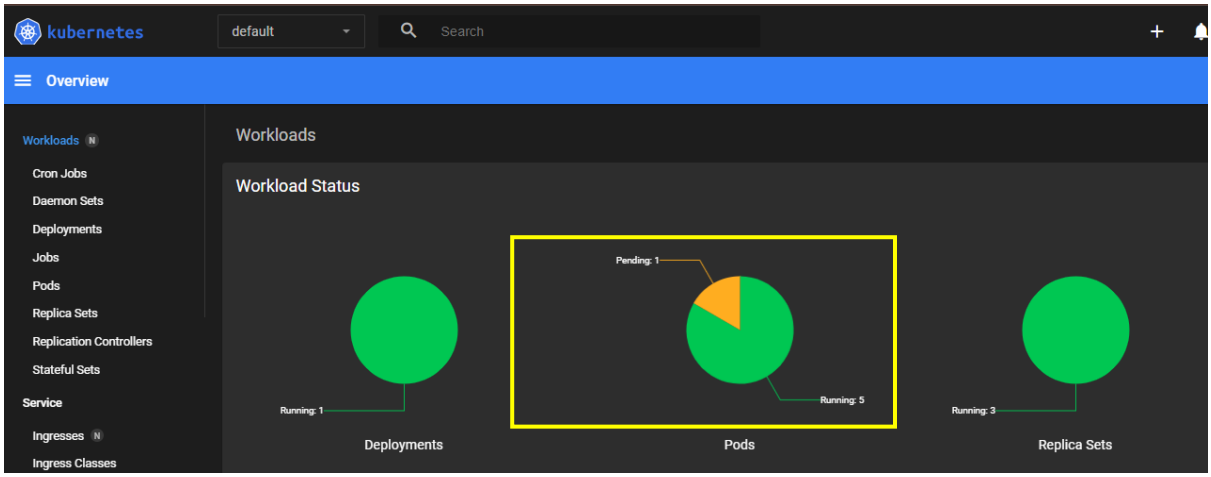
Edit the deployment

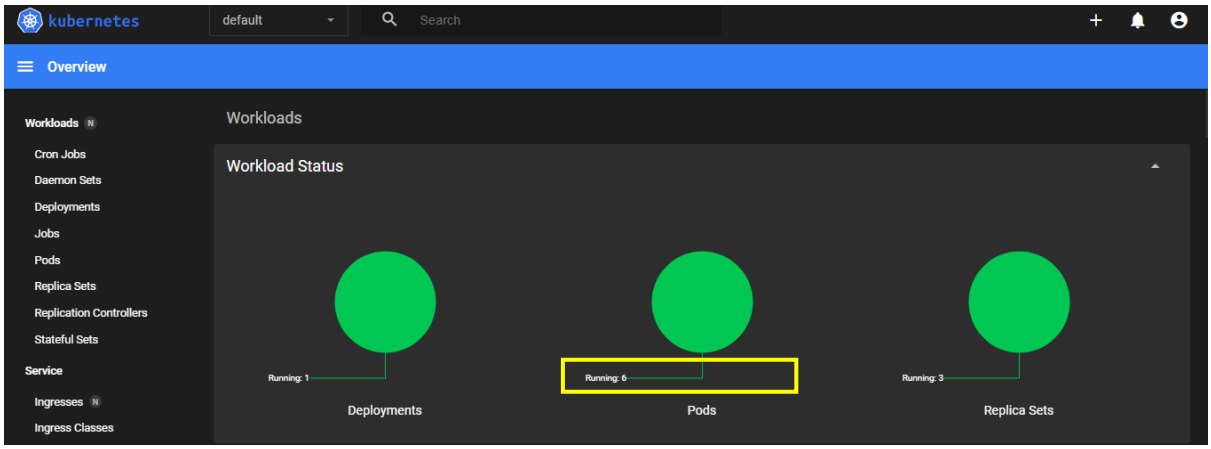


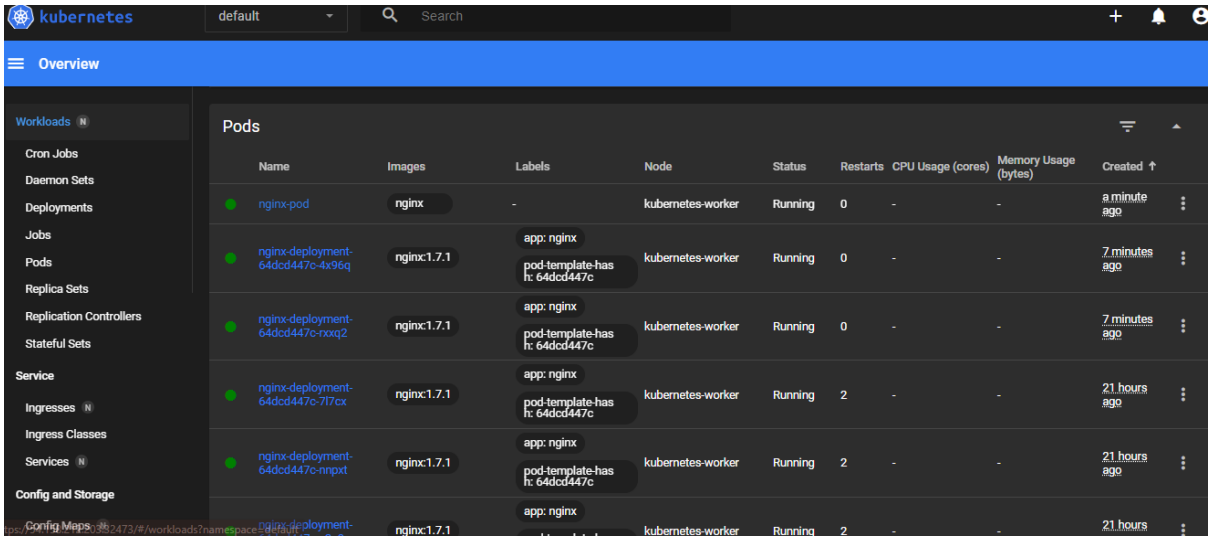
We can Edit the deployment here also



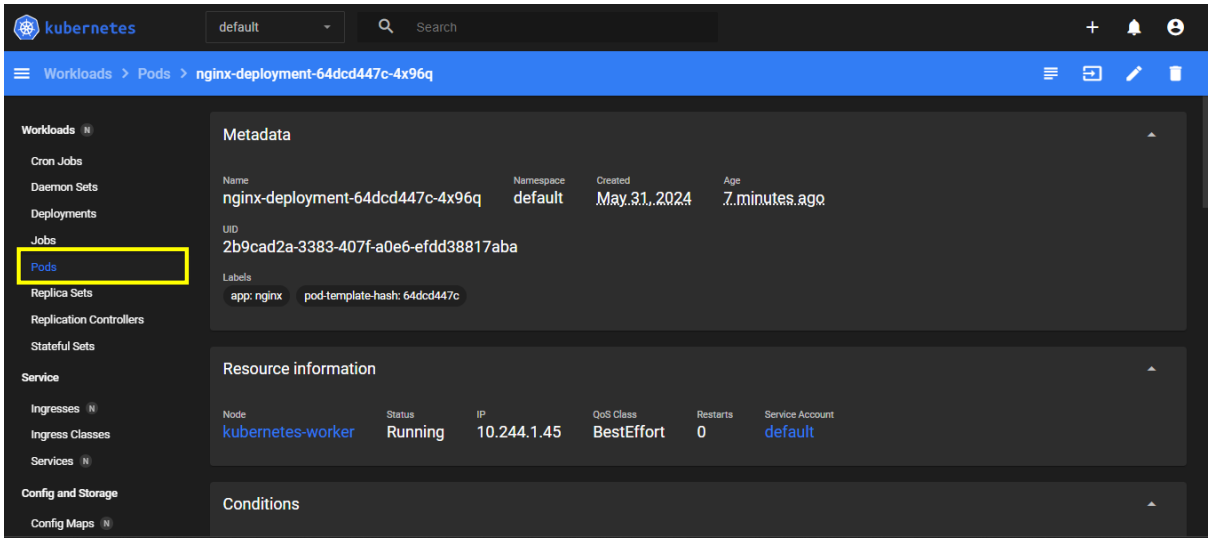
Here we can delete and create the pods, deployment



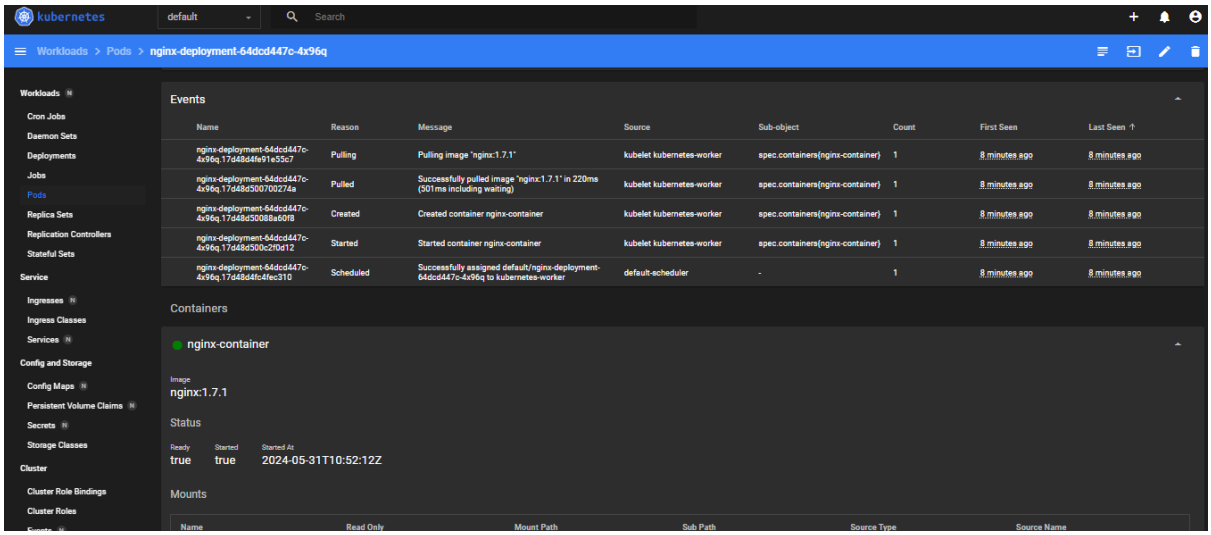




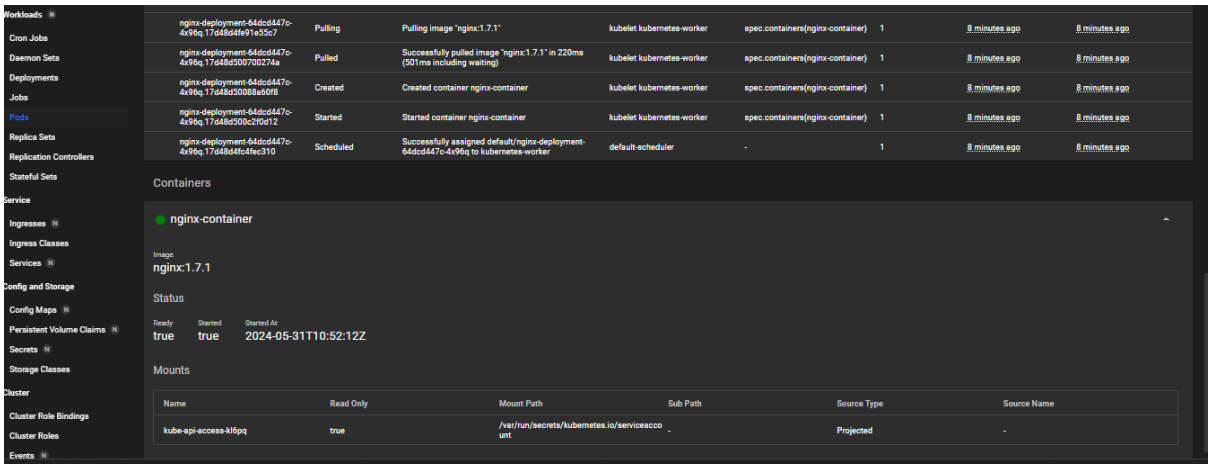
Inside the pod



Inside the pods > Conditions, controlled, replicaset



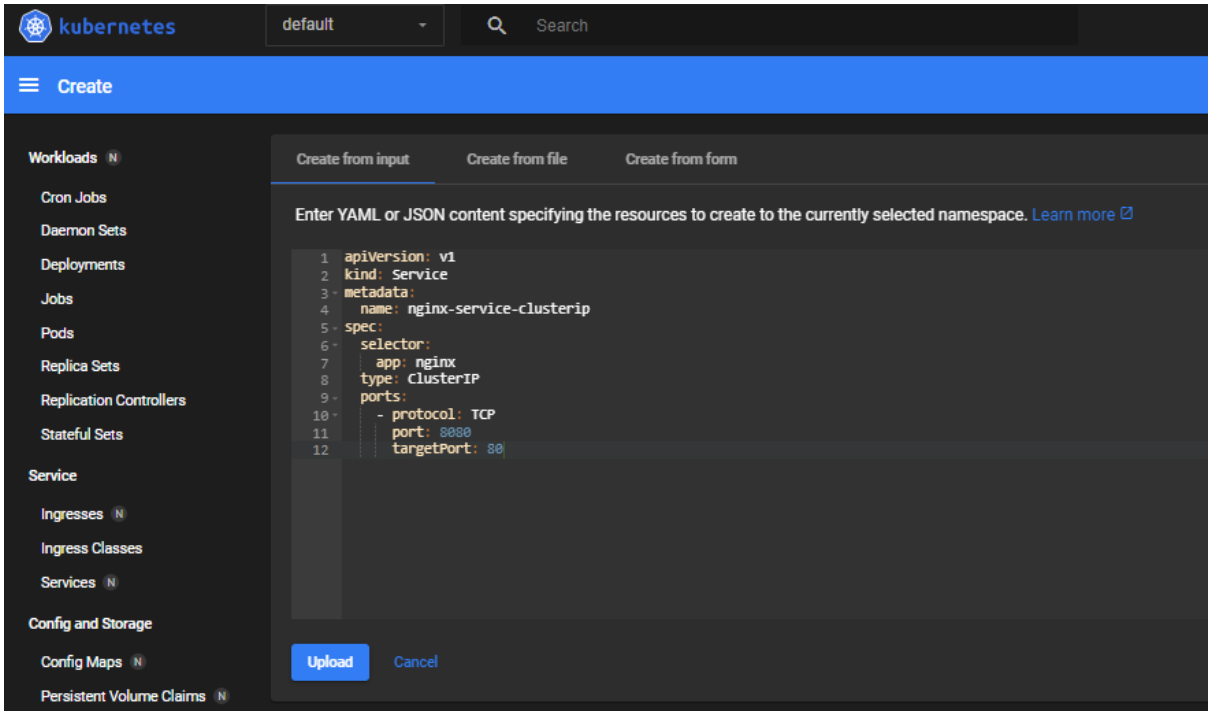
More details about pods



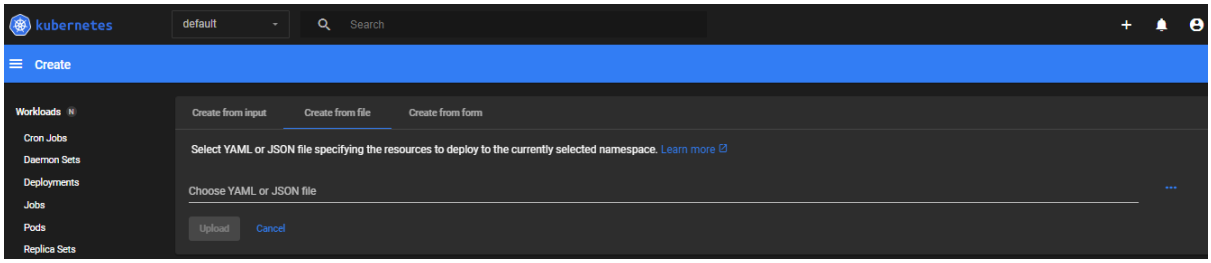
We can edit from the terminal

```
ubuntu@Kubernetes-Master:~$ ls
clusterip.yml  deployment.yml  ingressrule.yml  nodeport.yml  pod.yml  replicaset.yml
ubuntu@Kubernetes-Master:~$ cat clusterip.yml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service-clusterip
spec:
  selector:
    app: nginx
  type: ClusterIP
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 80

ubuntu@Kubernetes-Master:~$
```

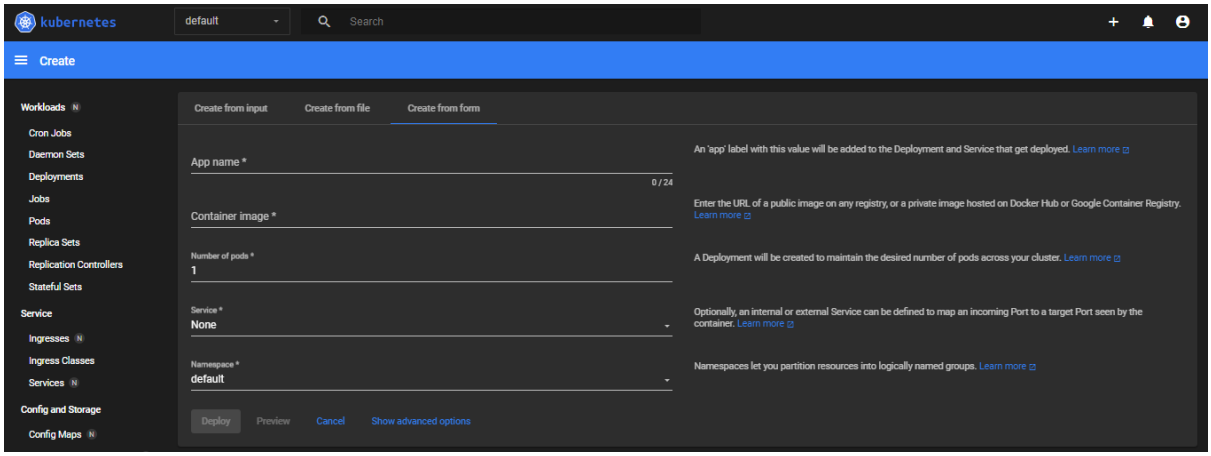
And also we can edit from the Kubernetes dashboard



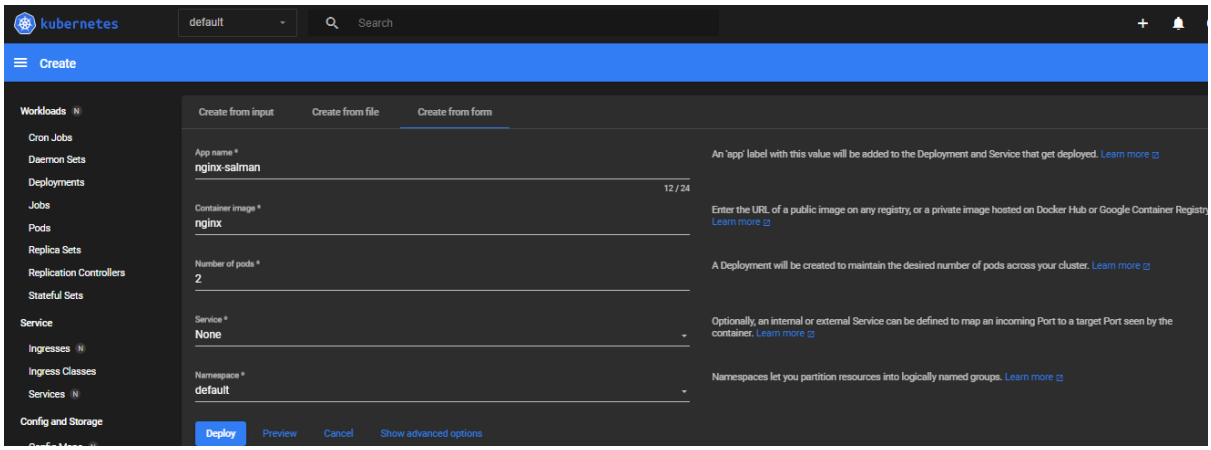
We can upload from into our local machine



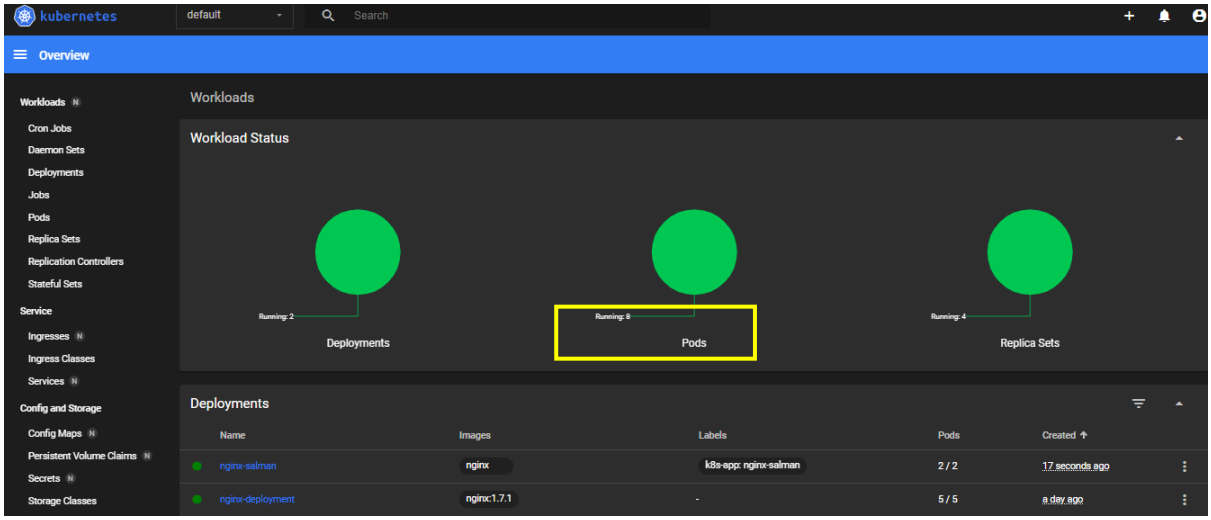
We can create app name, container image, number of pods through GUI(Kubernetes-dashboard)



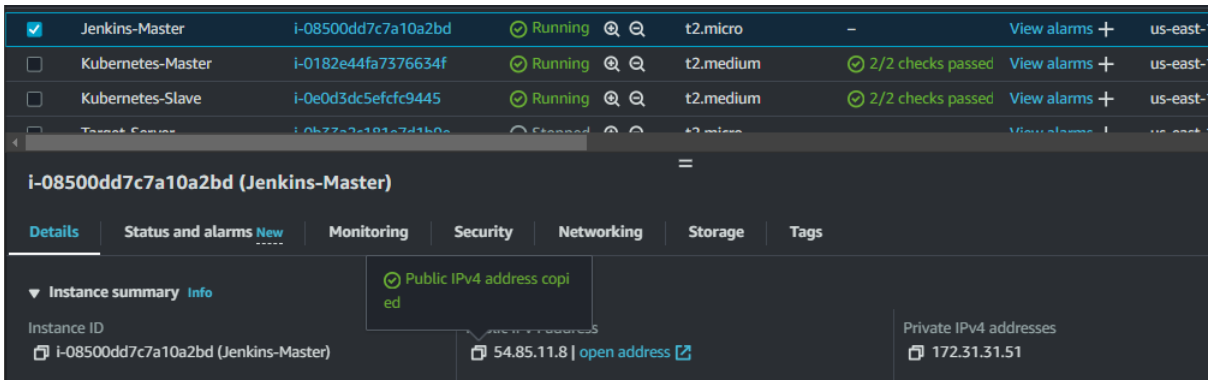
Creating sample



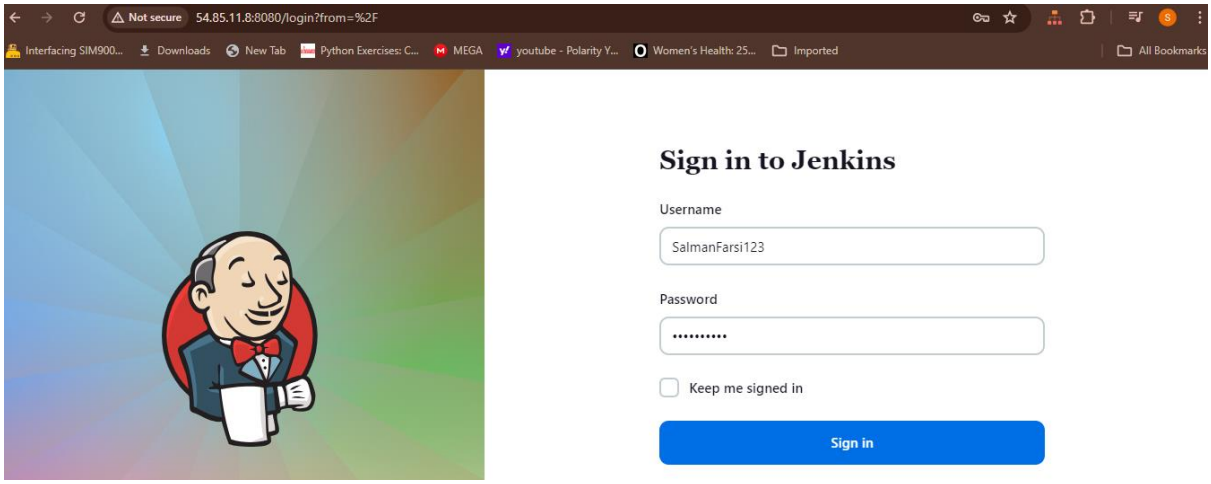
Now we have 8 pods we can see just now we created.



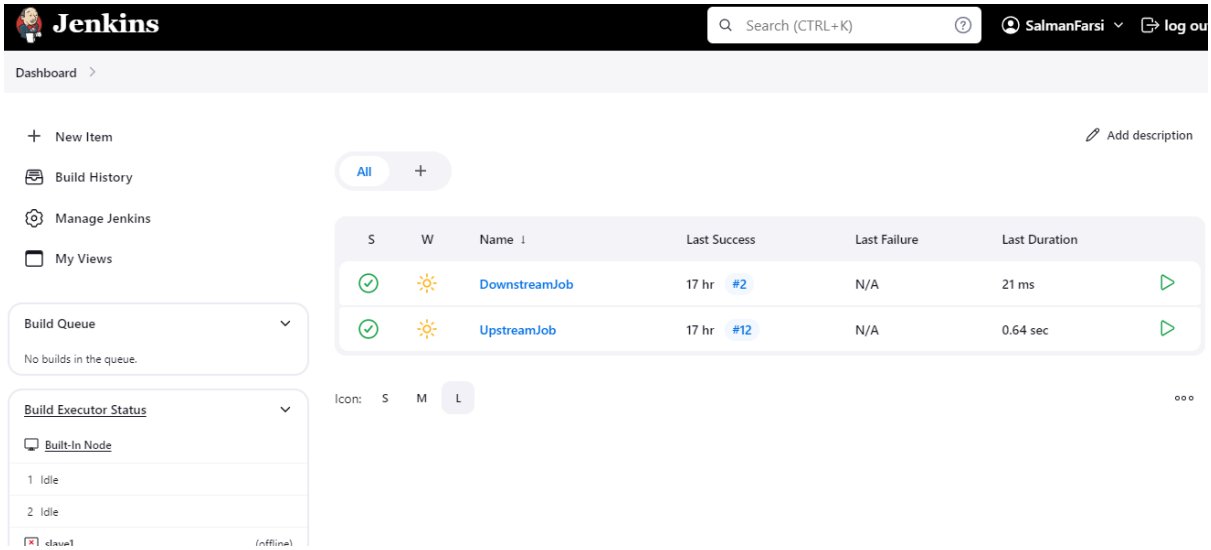
Now we are going to launch Jenkins-master. To deploy Kubernetes on jenkins

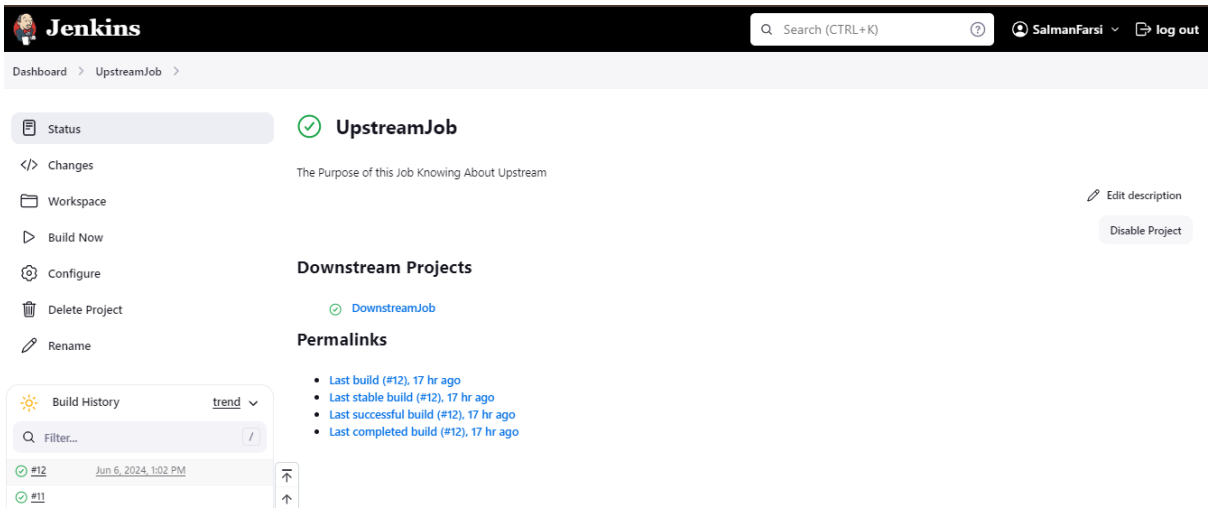


Sign in

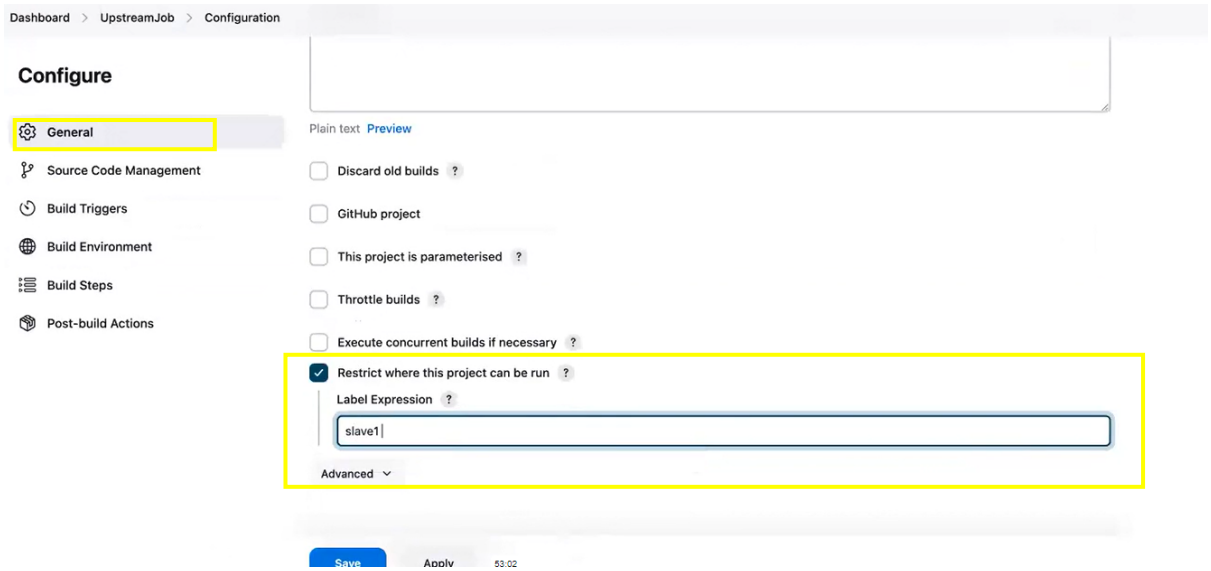


This is Jenkins demo, previously have practiced

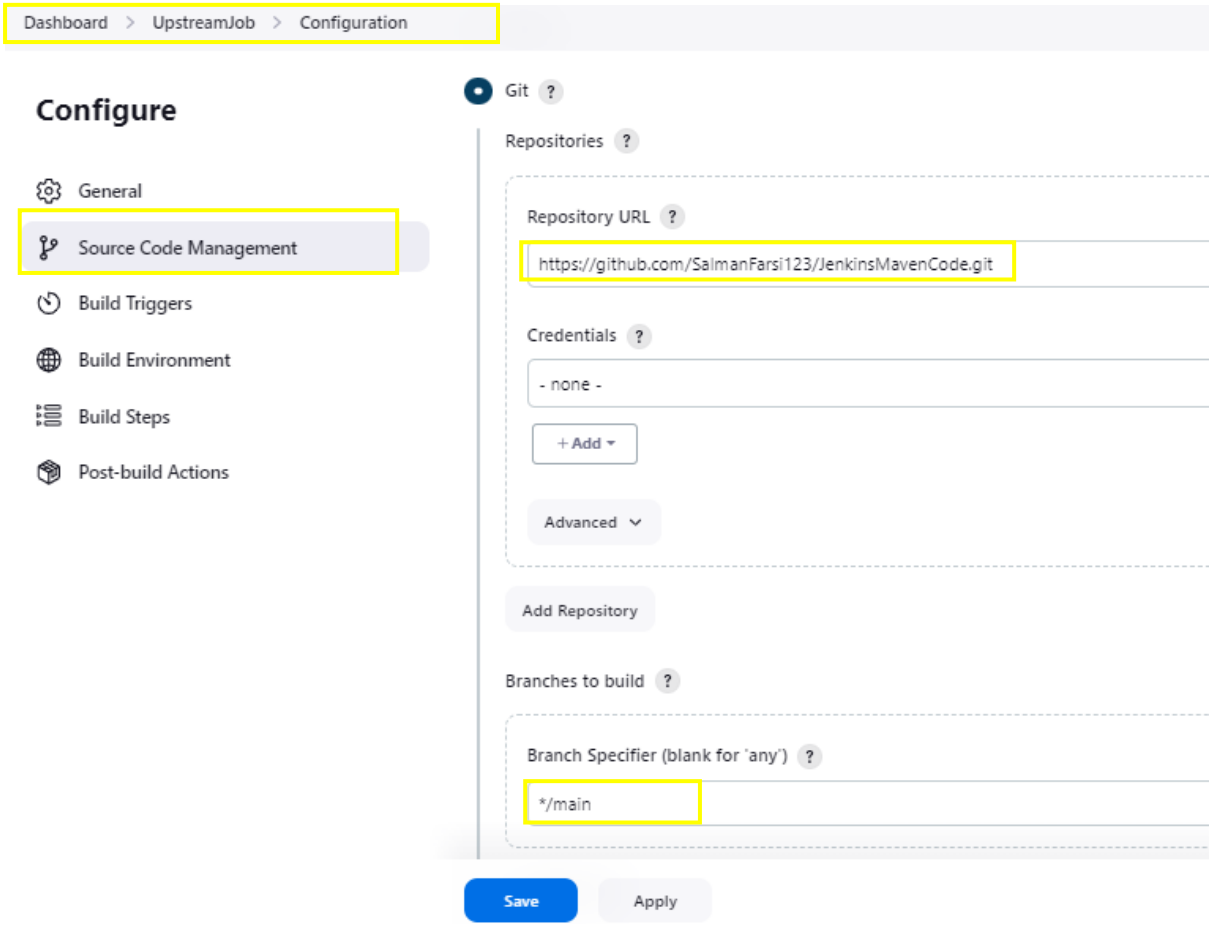




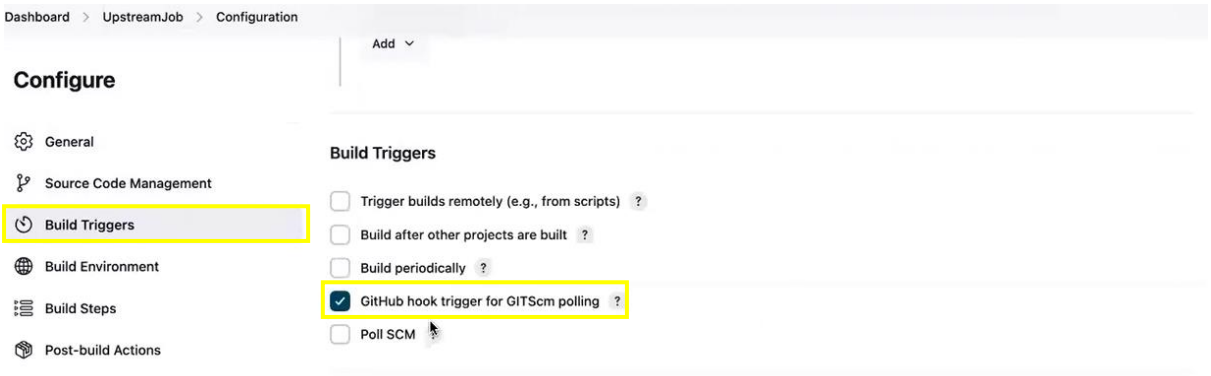
Build in slave1



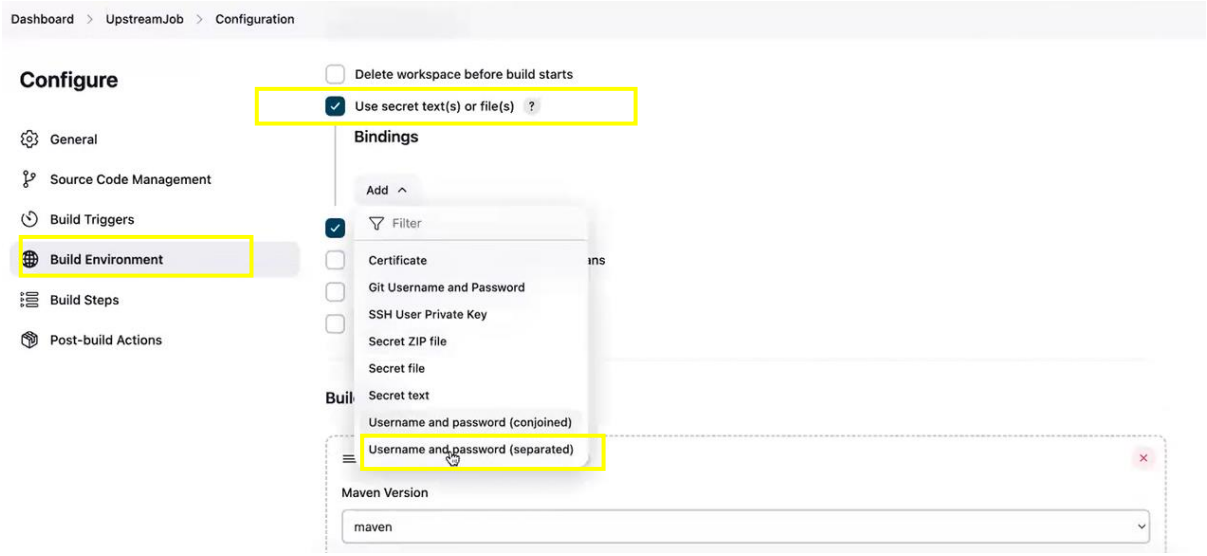
Cloning the repo in the slave1 from the main branch



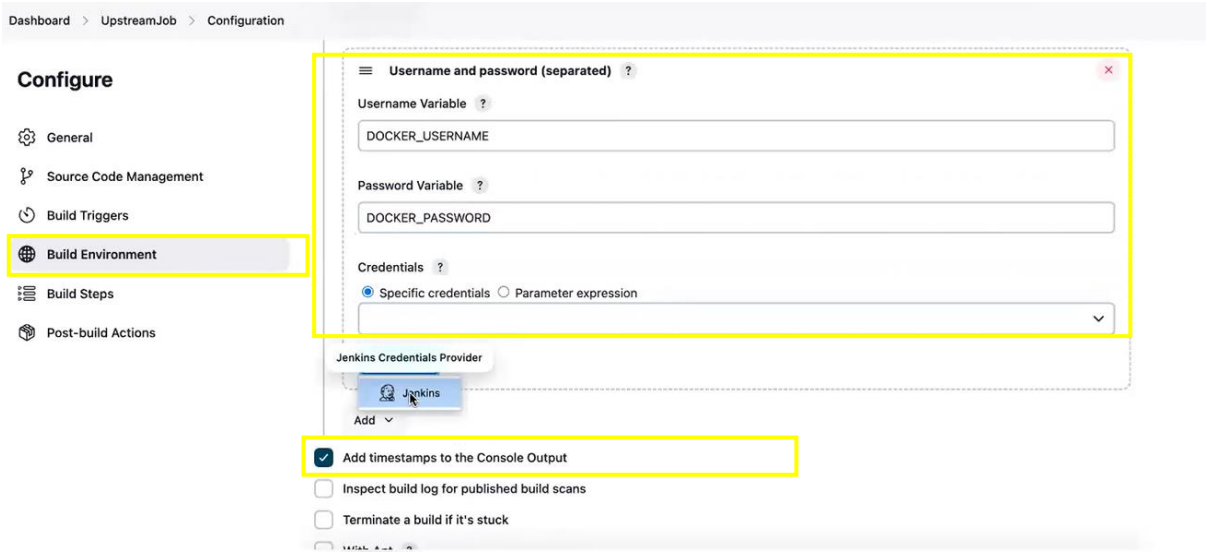
Enabling github hook trigger, if any changes and commit happens then it will going to Build the Job



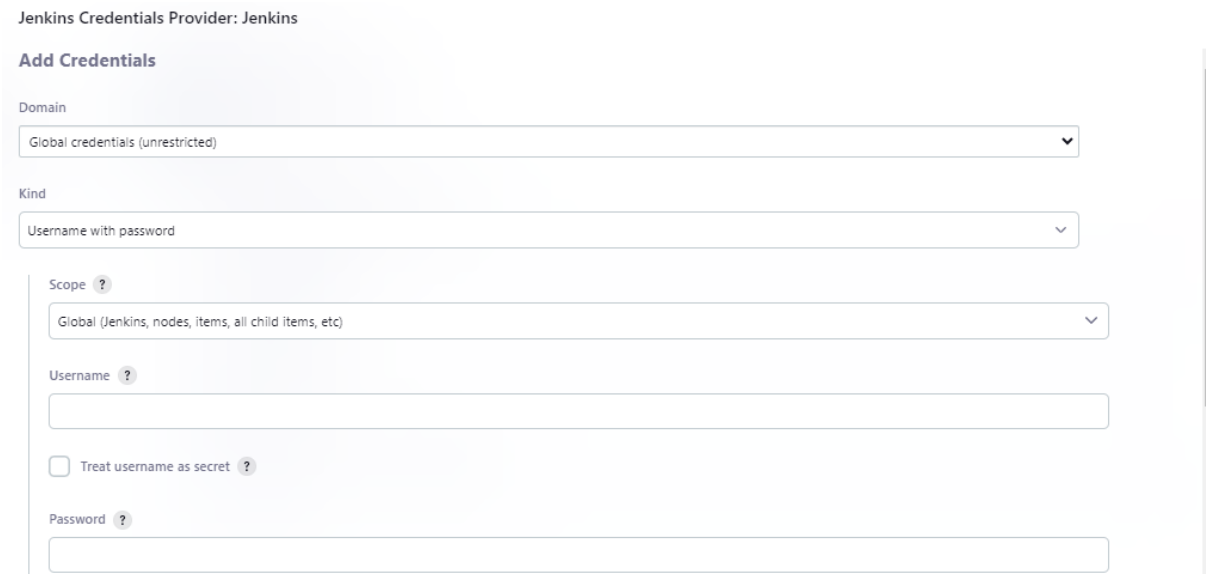
Use secret texts or files and using username and password(separated)



Click Add > Jenkins (Remember) U need to have dockerhub account and using variables hiding username and password purpose.

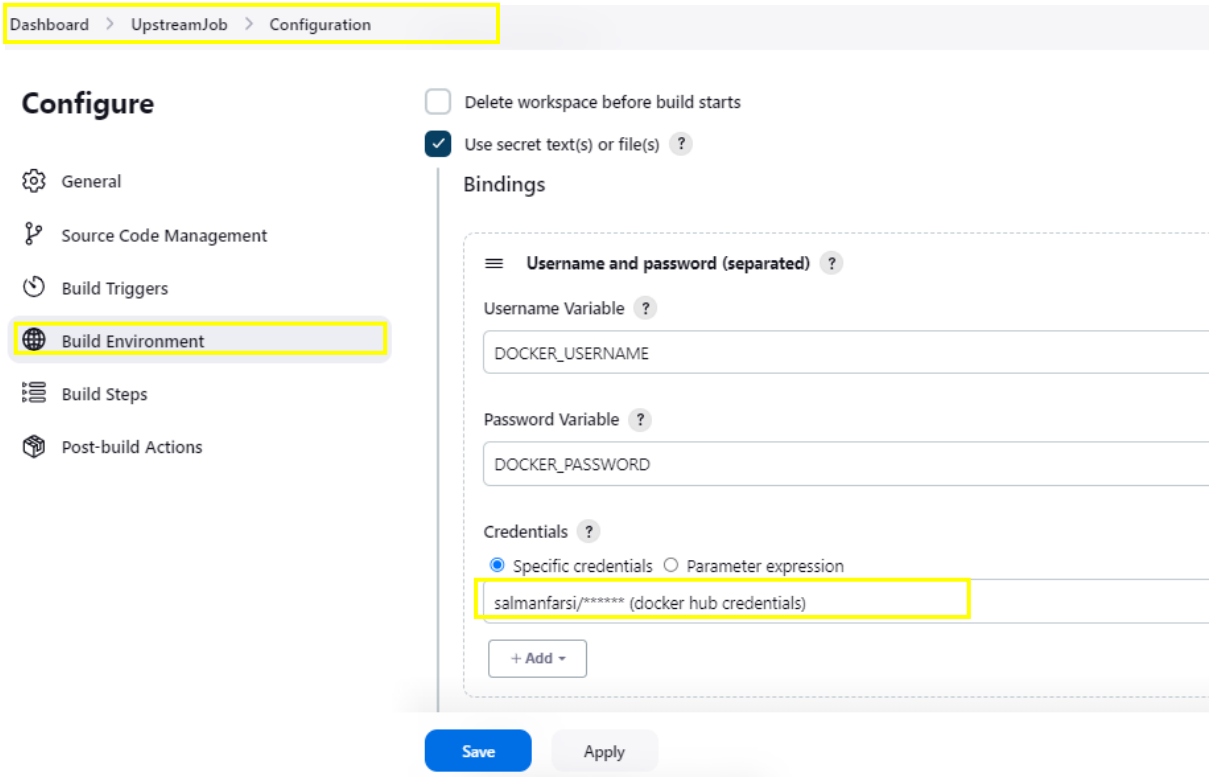


Kind: username and password and enter Your Dockerhub user name and password, ID and description, type anthing which u want.



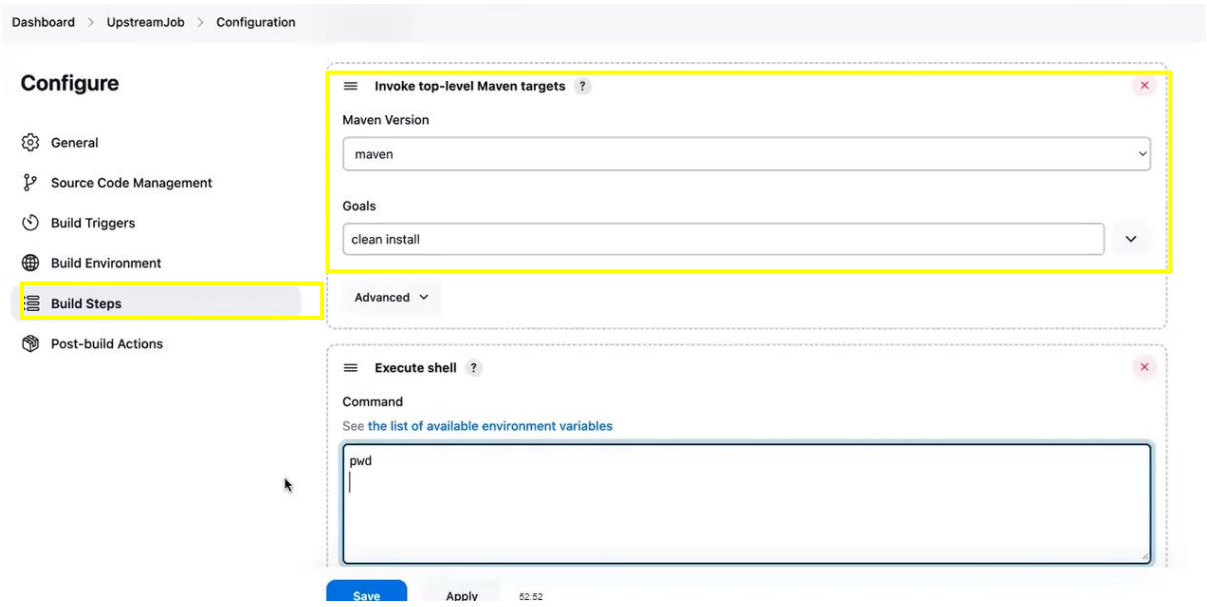
I add my credentials

We set up it.

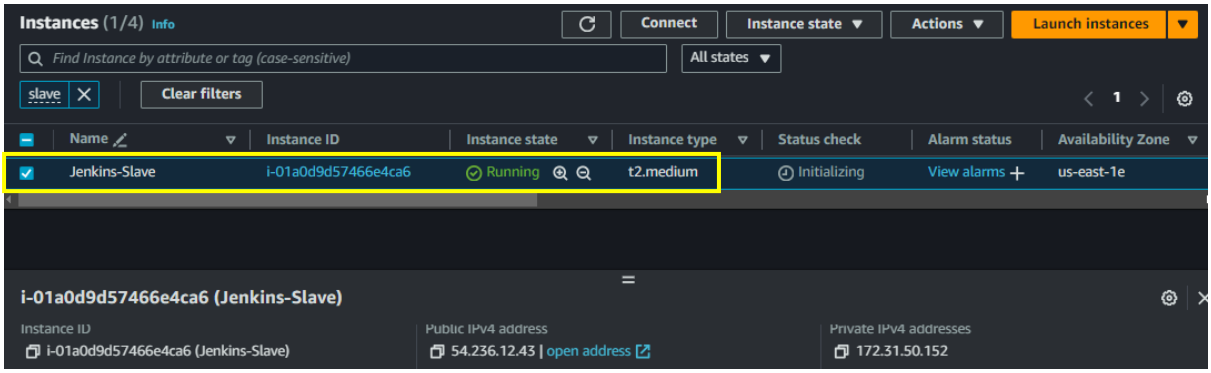


According to Jenkins task, u have keep this invoke top-level maven targets because anything developers add in the git hub, maven build going to change into executable format so keep it.

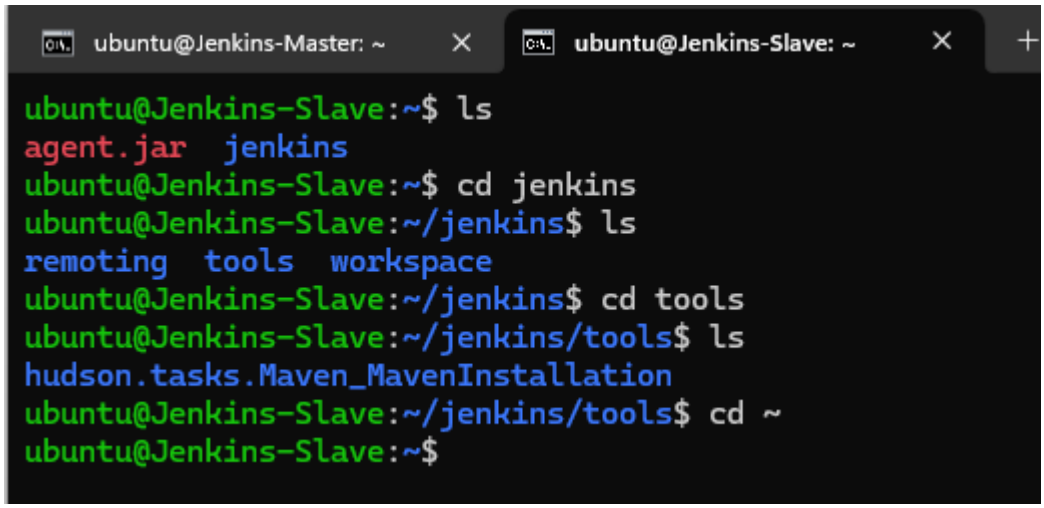
Now After come to Build steps u have to select execute shell and need to write script according to your project.



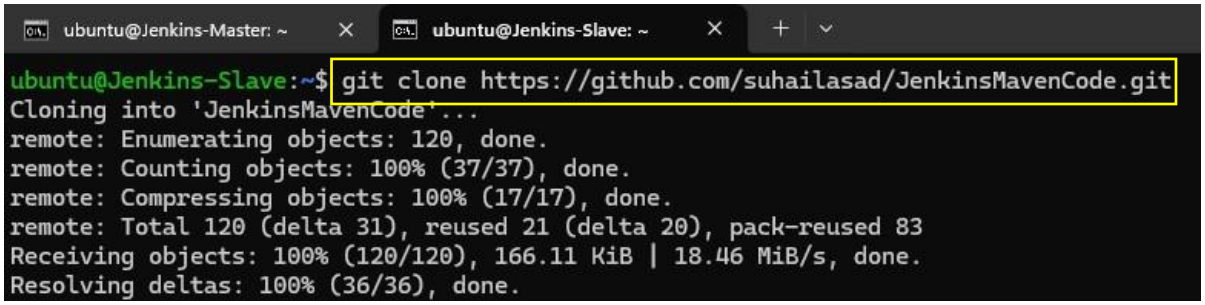
I am Launching my slave account, which is stopped, all build going to save and execute in this slave because I selected slave1



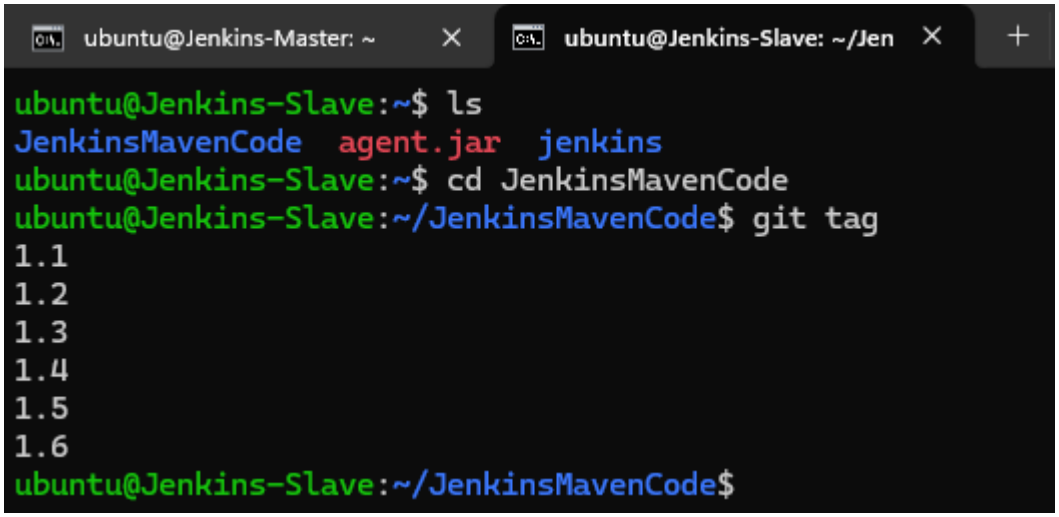
This was previous execution which I made in Jenkins demo



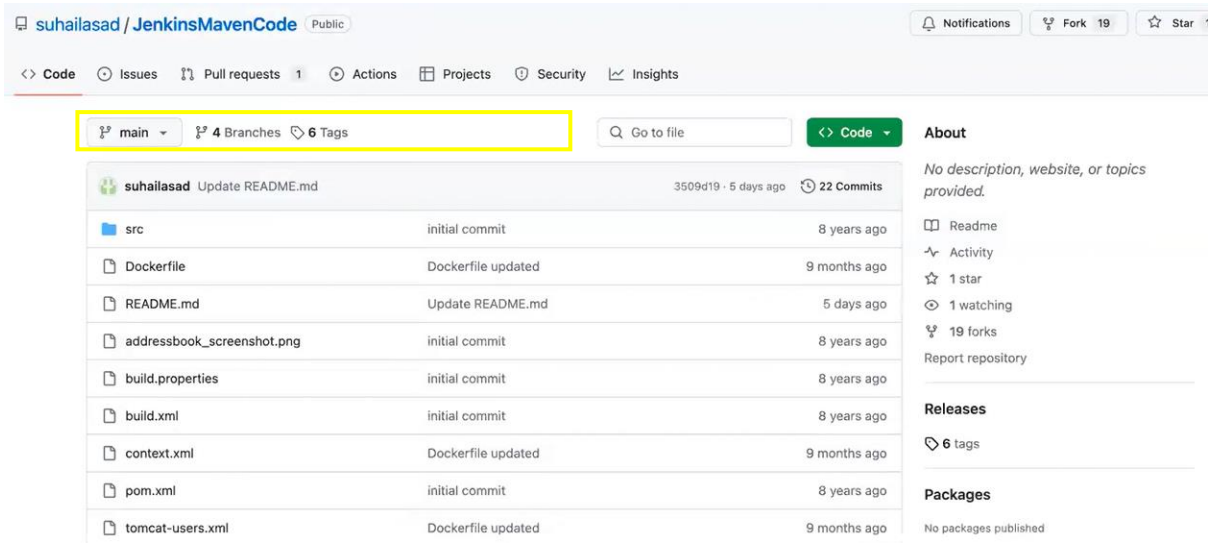
Copy Suhails Maven repository and Do Git Clone it will pull that all stuff of that repository.



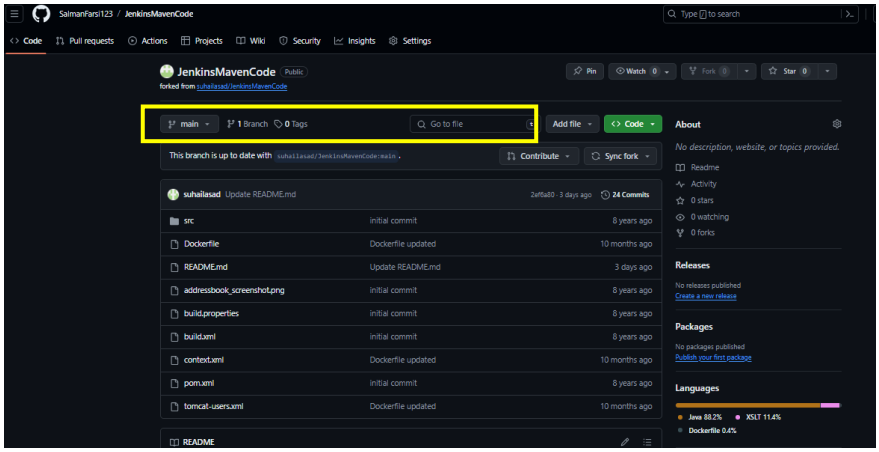
We can see the JenkinsMavenCode which is cloned from Suhail repo



We can in Suhail repo has 4 branches and 6 tags



But our repo have only 1 branch, we did this in Jenkins demo



Go to Inside the Maven and Follow all the commands

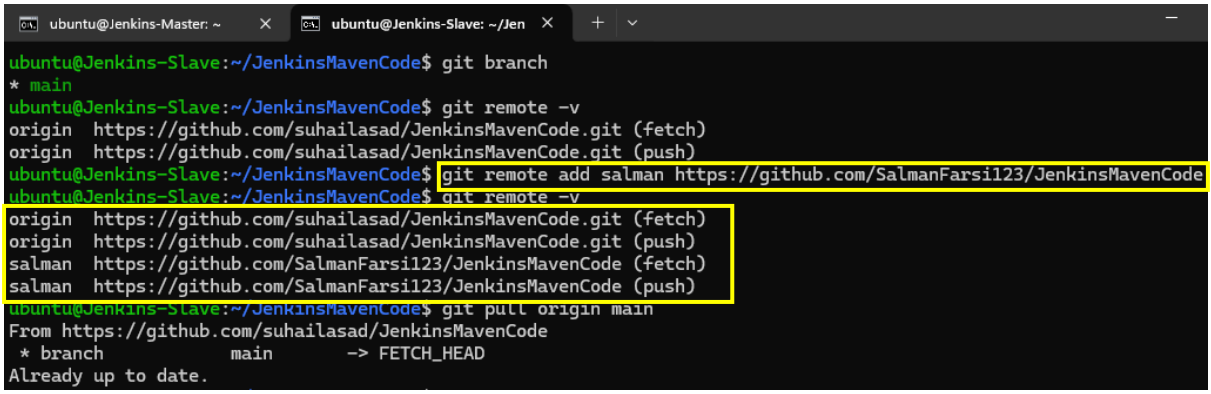
Check git remote -v which is we cloned suhailasad account it will shown

Now I am Adding My Github

Git remote add salman/replace your name Path of your github mavencode

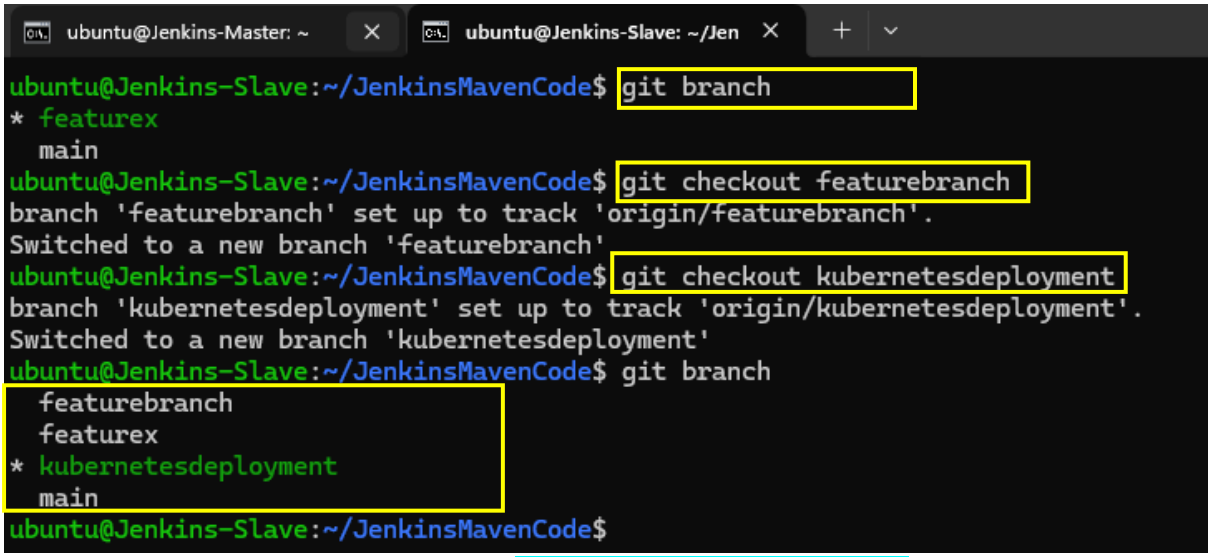
Check git remote -v

Git pull origin main it show already up to date(Because we already clone)

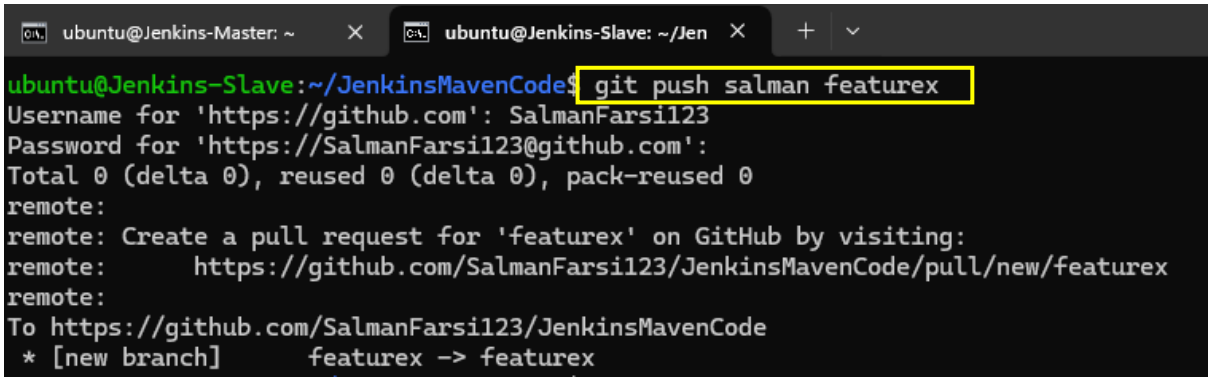


git branch git checkout feature git checkout featurebranch git checkout kubernetesdeployment

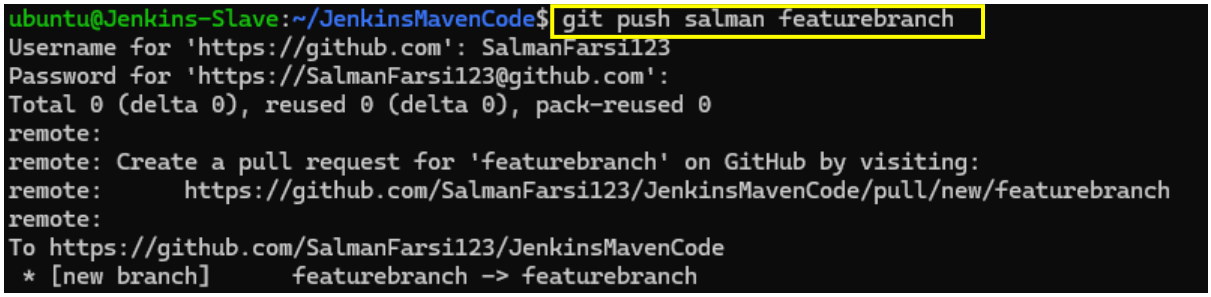
git branch will show all branches



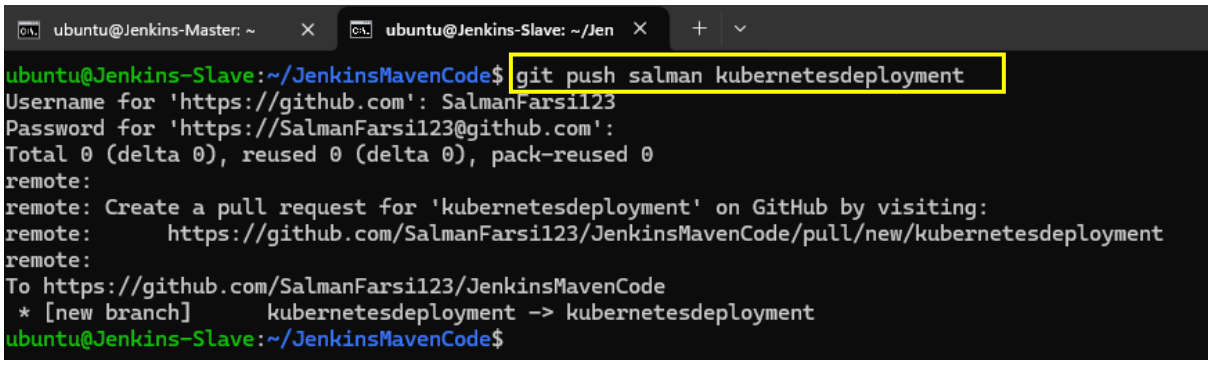
git push salman featurex and it will ask username and token of github



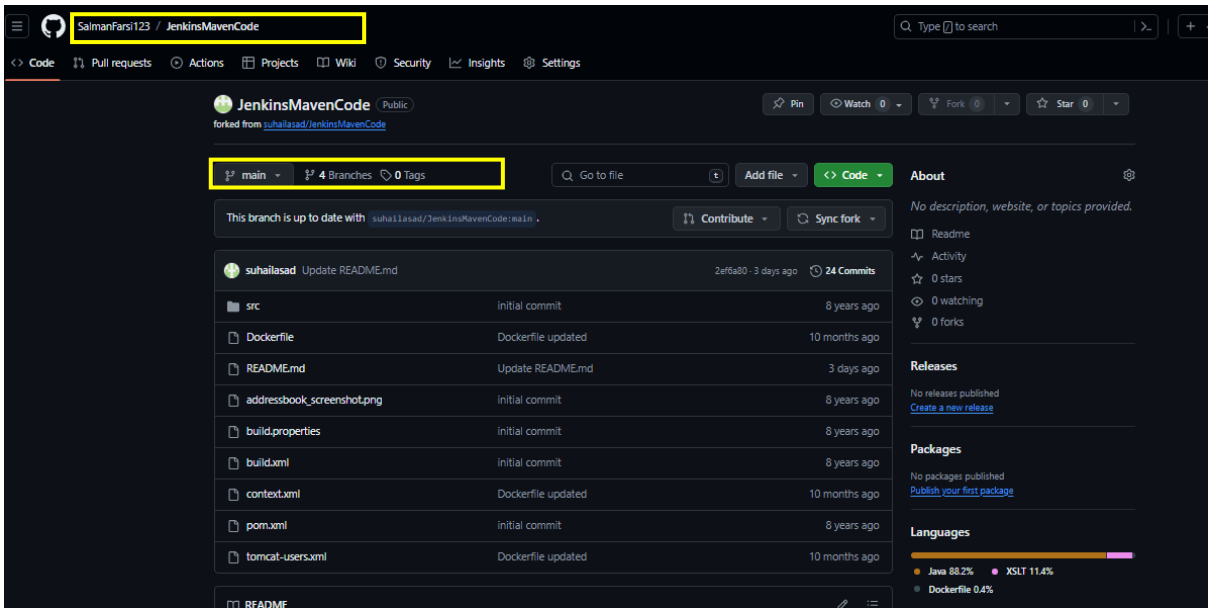
git push salman featurebranch and it will ask username and token of github



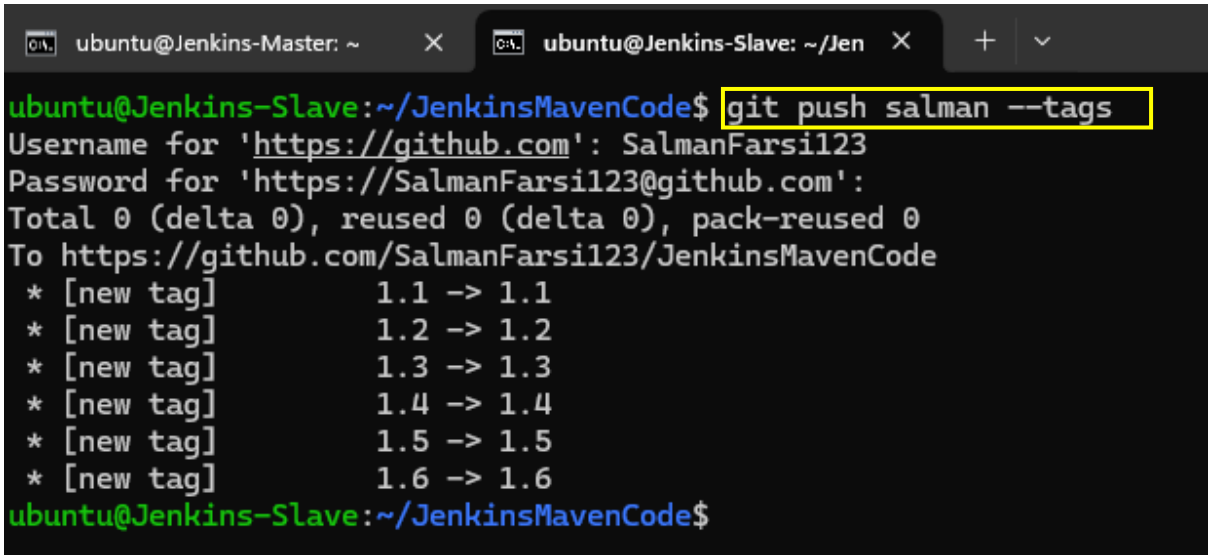
git push salman kubernetesdeployment and it will ask username and token of github and No need to Push Main branch because we already fork that branch.



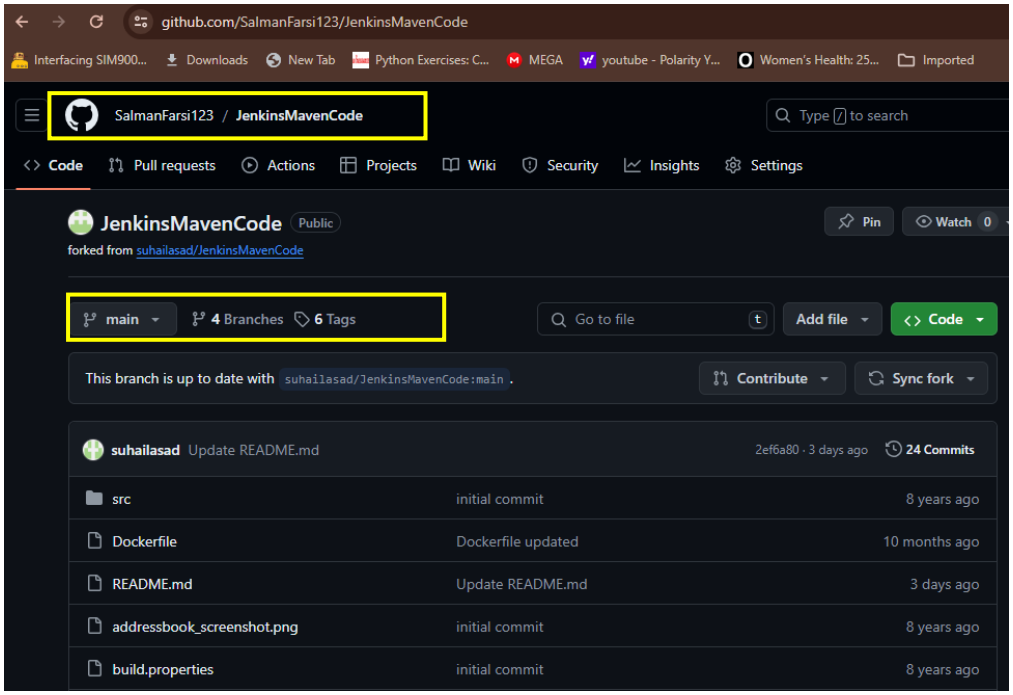
Now we can see in my account all 4 branches, but tags are missing



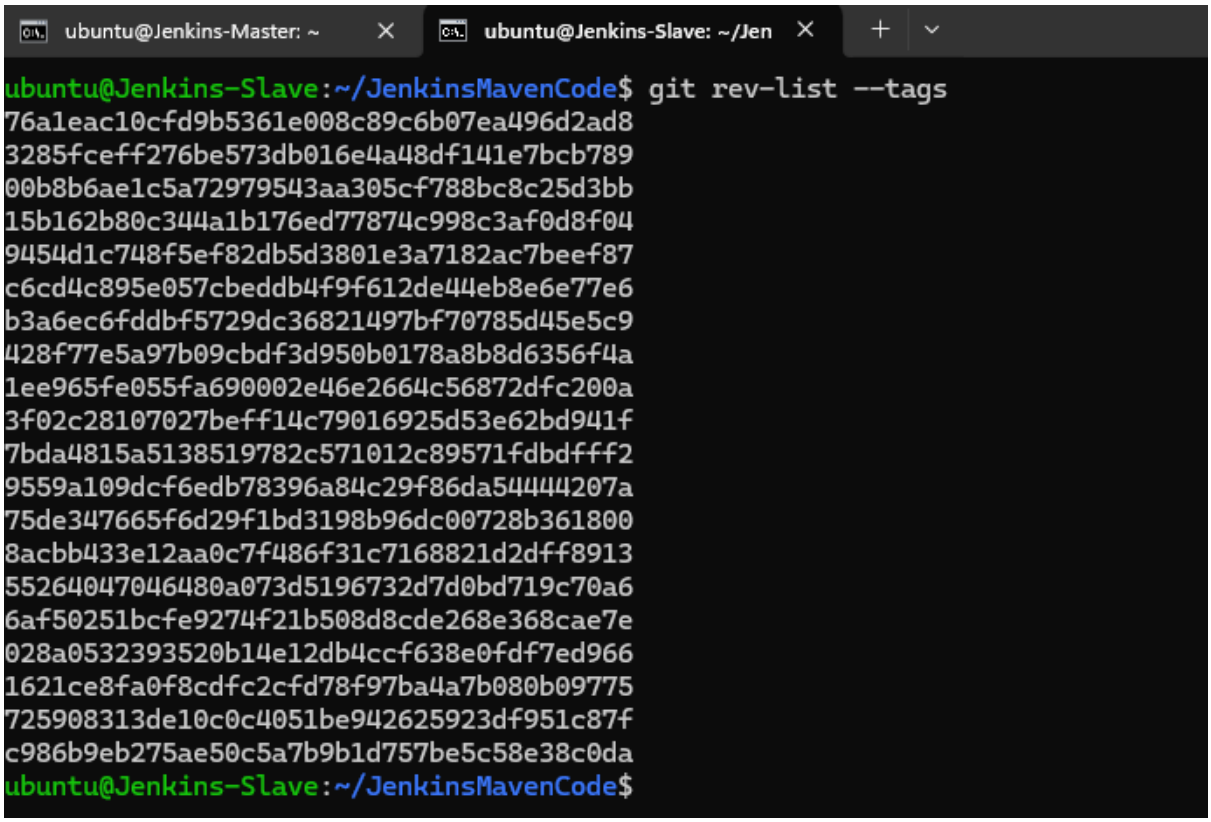
git push salman –tags and it will ask username and token of github and it will push all the tags.



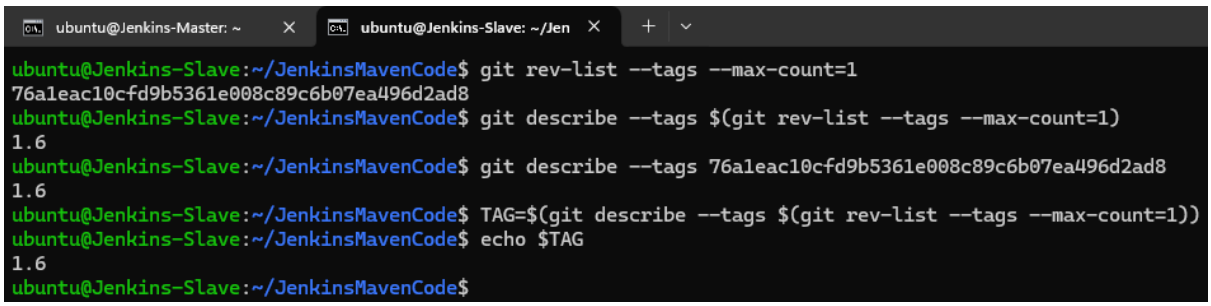
Now we are able to see all those stuff



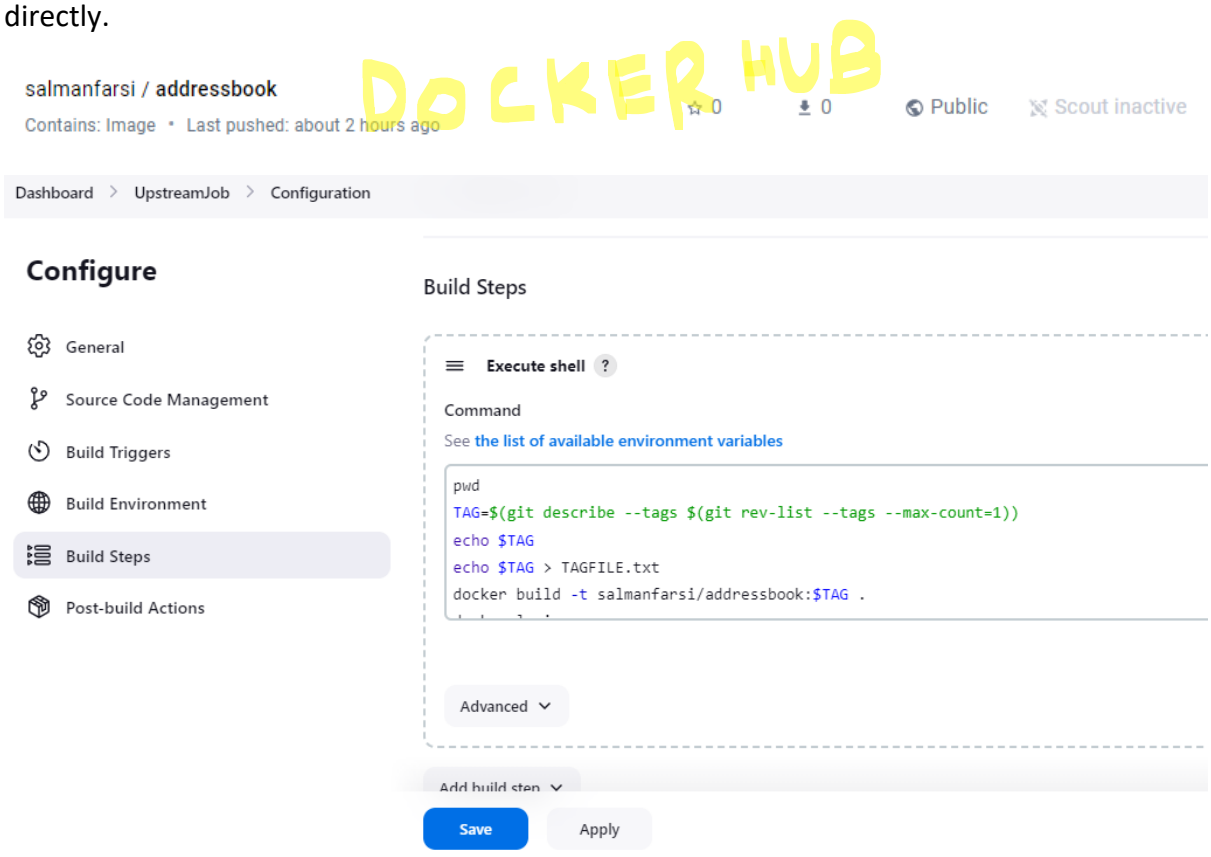
Git rev-list -tags



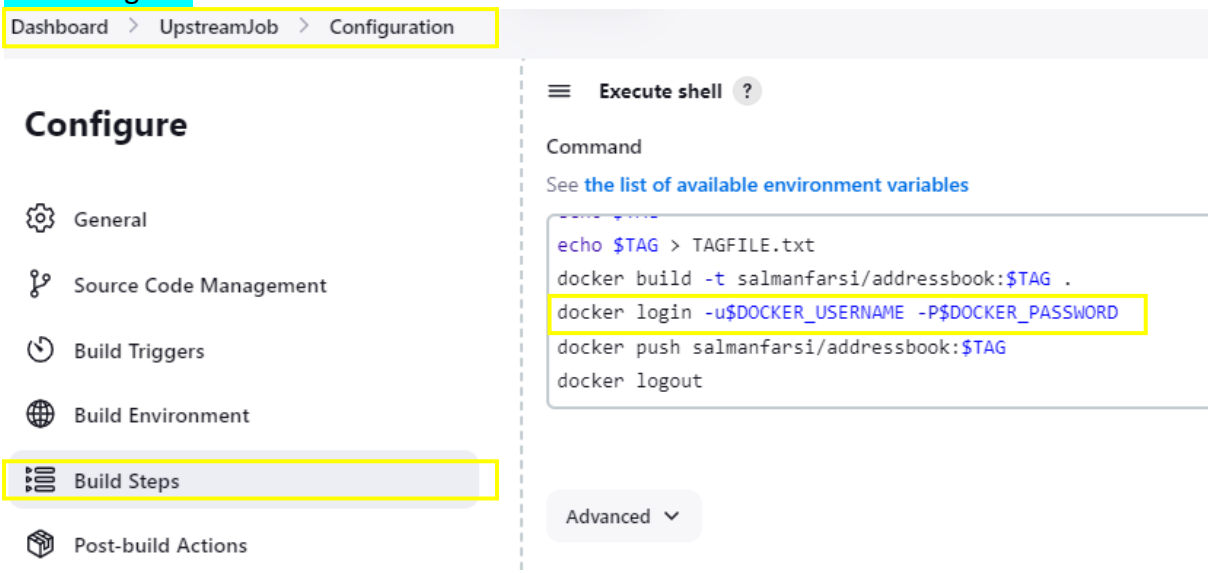
Now this git tag 1.6 and 76a1eac10cfd9b531e008c89c6b07ea496d2ad8 given name as TAG and the tag data stored in TAG with using linux commands



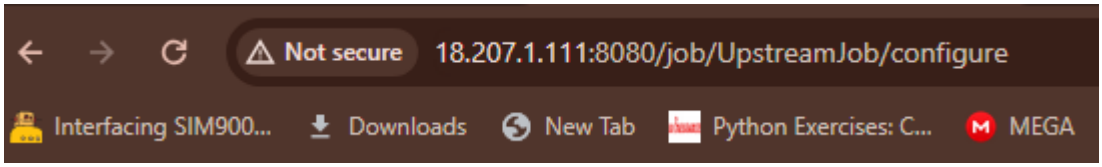
Go to Upstream > Configuration > Build Steps Paste the Commands first the TAG data will and then Tag data will be print and going to build docker and salmanfarsi/addressbook repository make sure u need to have dockerhub account with have salmanfarsi/addressbook this name repository and paste docker login and paste dockhub username and password Below mentioned how to cover that password instead of giving directly.



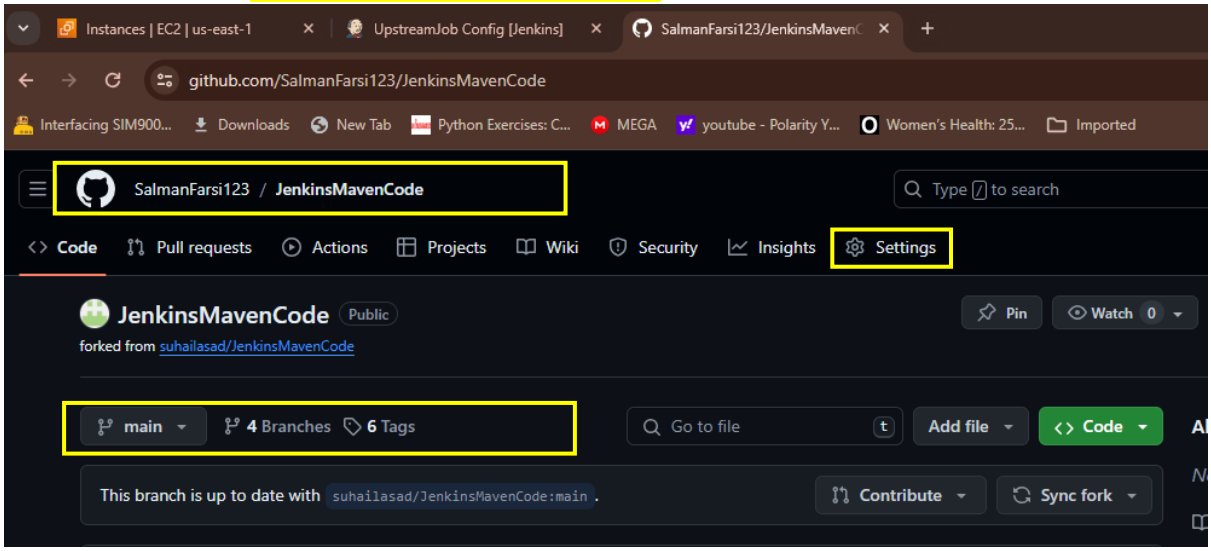
Now we can see username and password was covered. And pushing to dockerhub and docker logout.



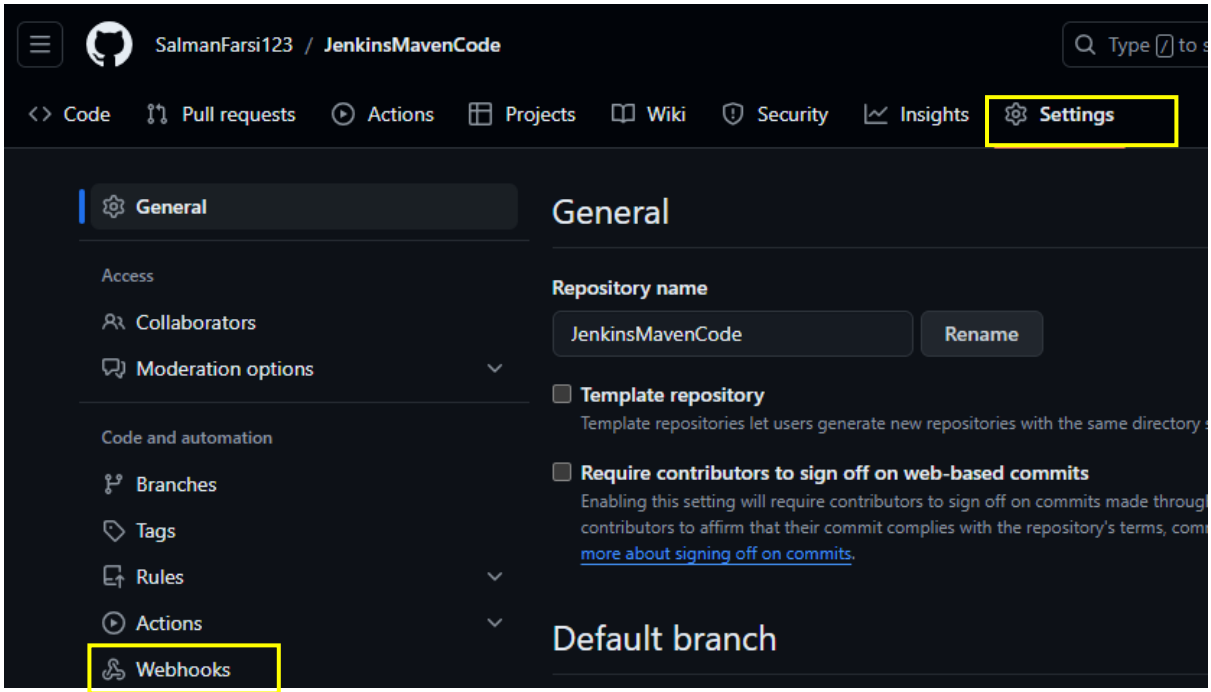
Copy the Public Ip of Jenkins Master and 8080 port



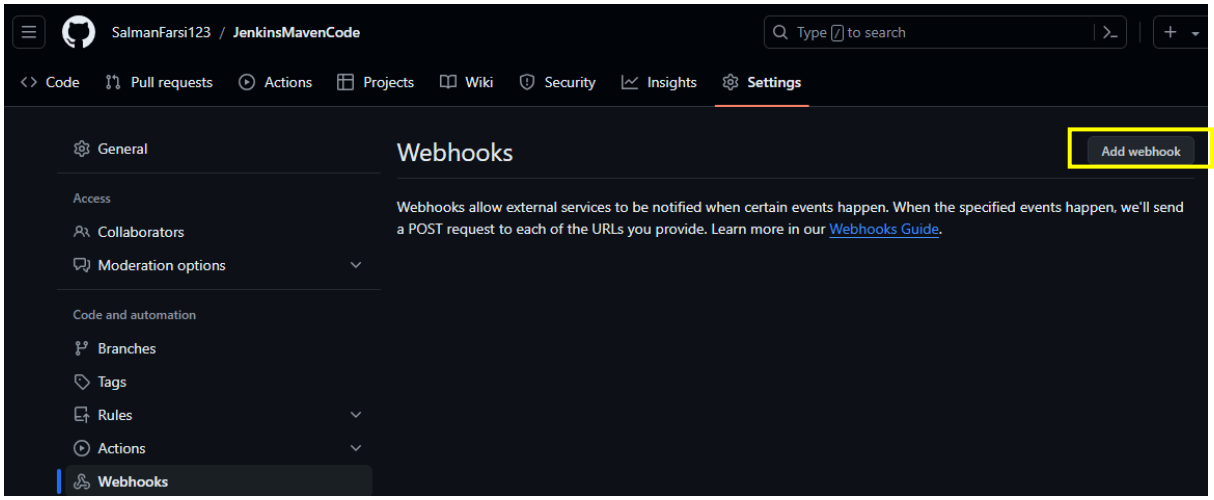
Go to Your GitHub Account > Settings > Webhooks



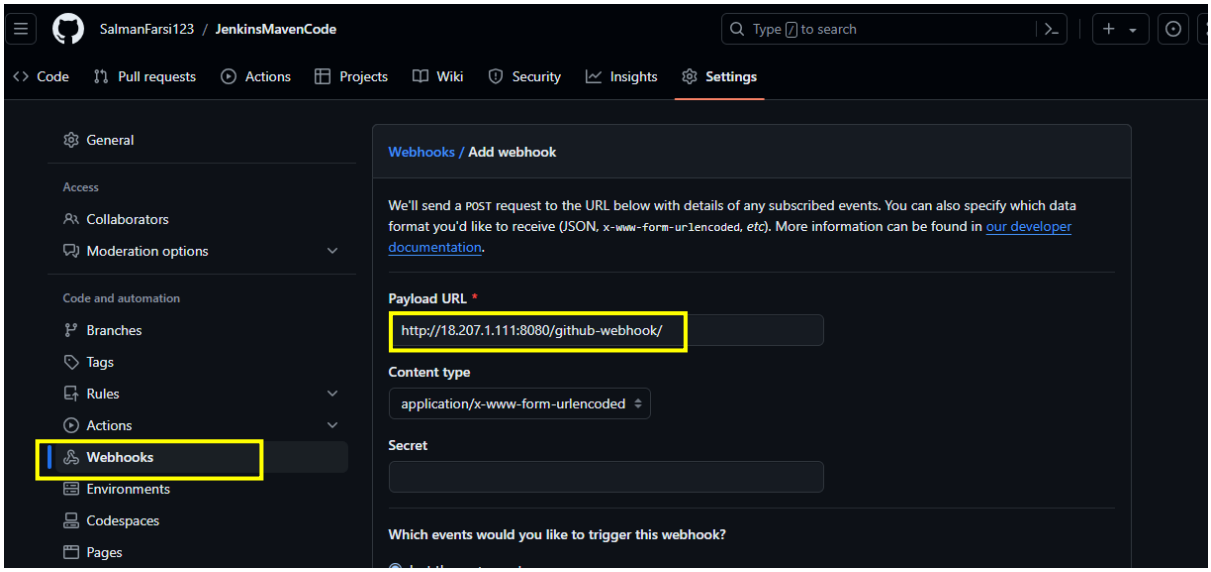
Go to Your GitHub Account > Settings > Webhooks



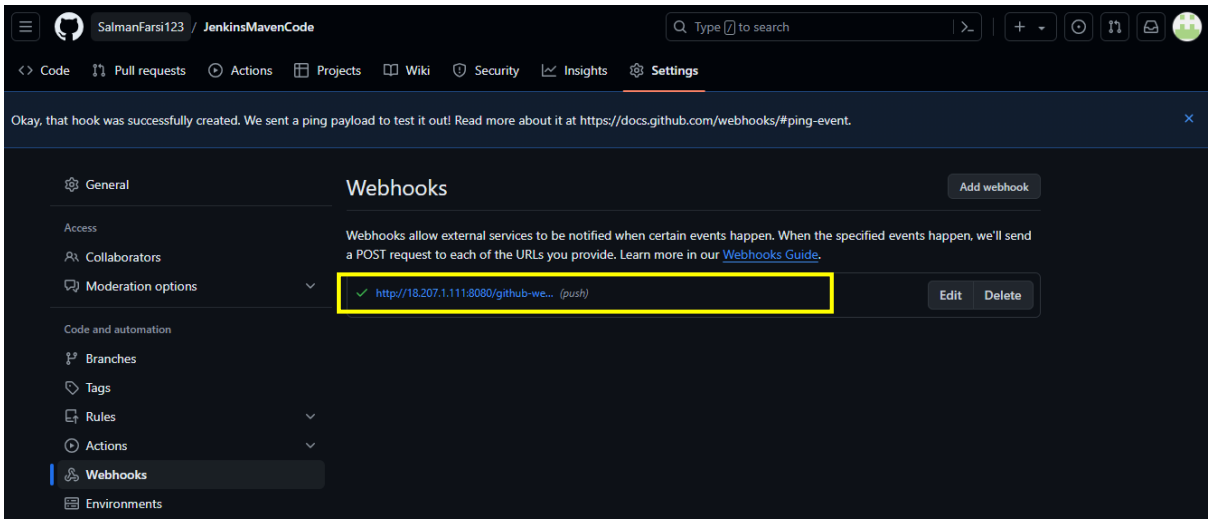
click Add Webhook in the right corner



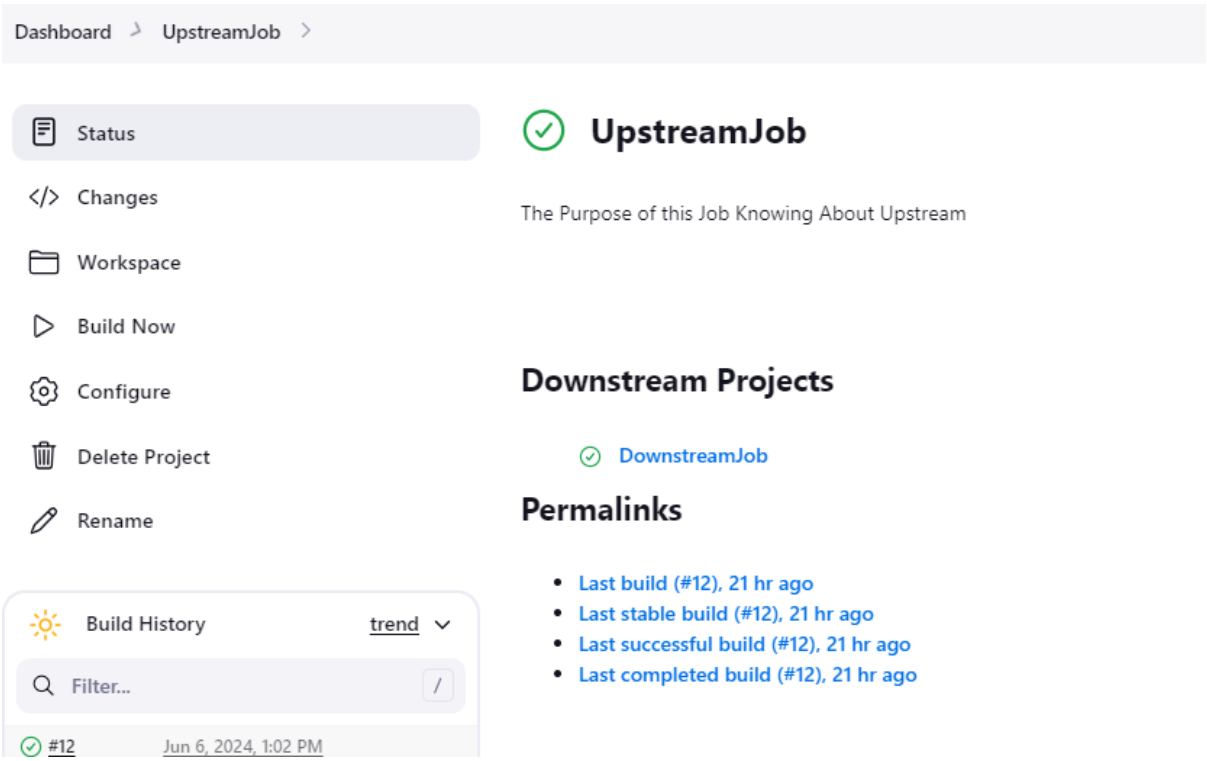
Paste the Jenkins Public-IP and port No of Jenkins 8080 and give github-webhook/



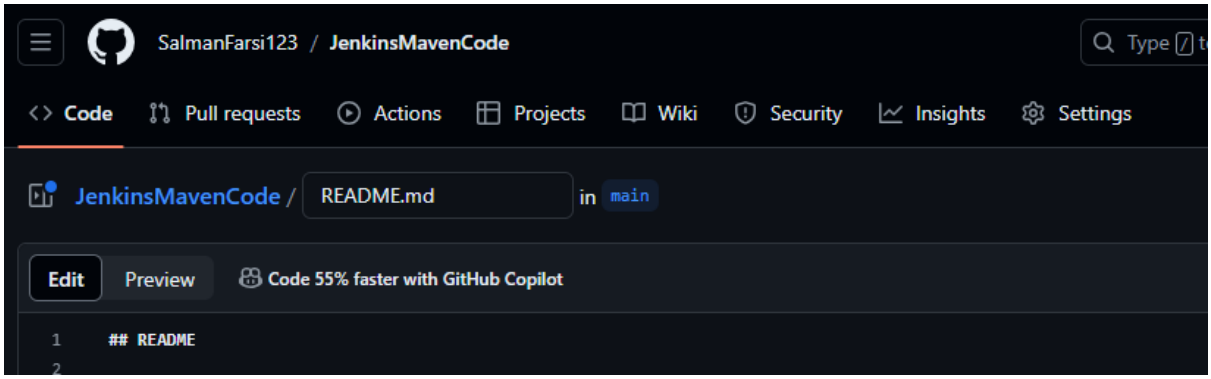
Add Successfully



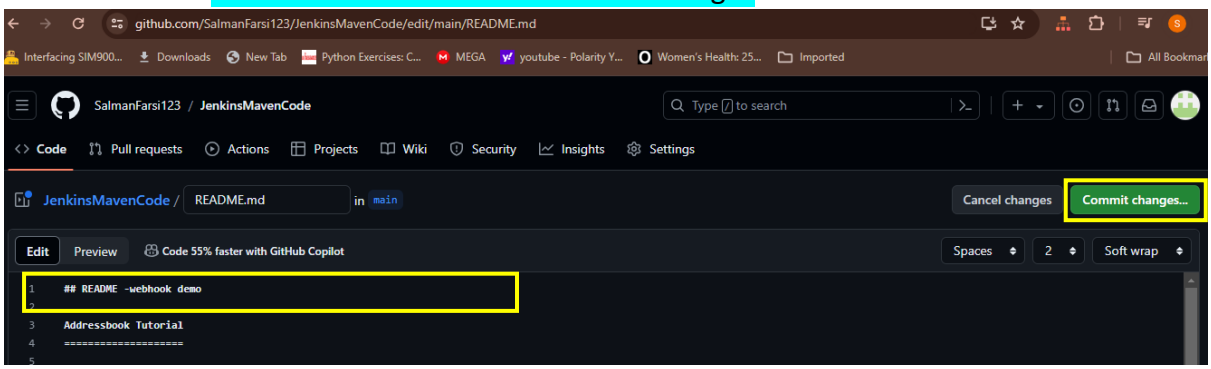
Now see the 12 Builds Made already



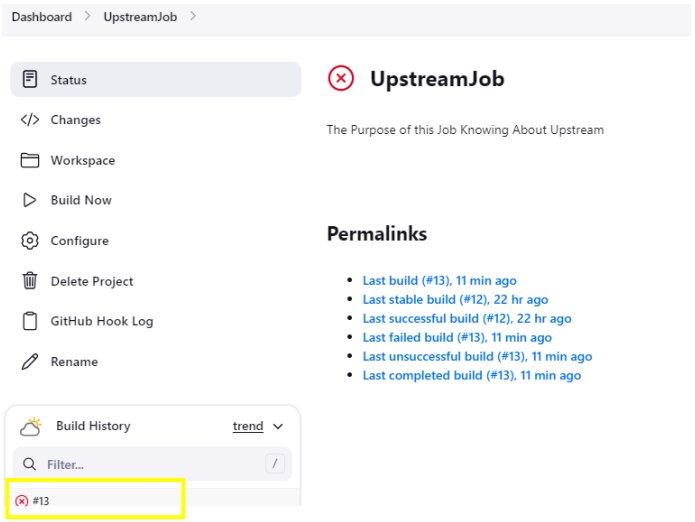
Now U can Make changes and I selected README.md

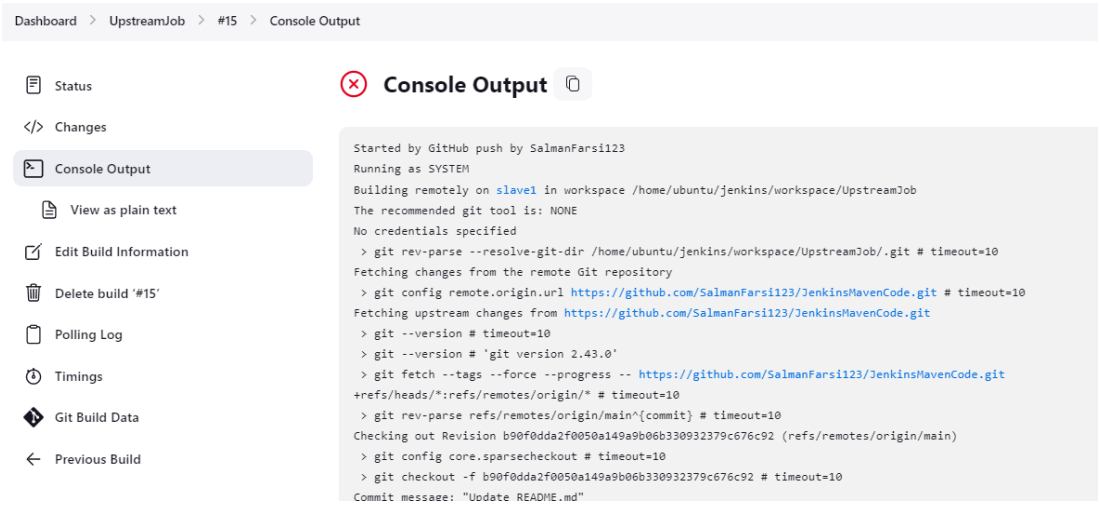


Now I removed webhook demo and click commit changes



Its Build but I Got an Error



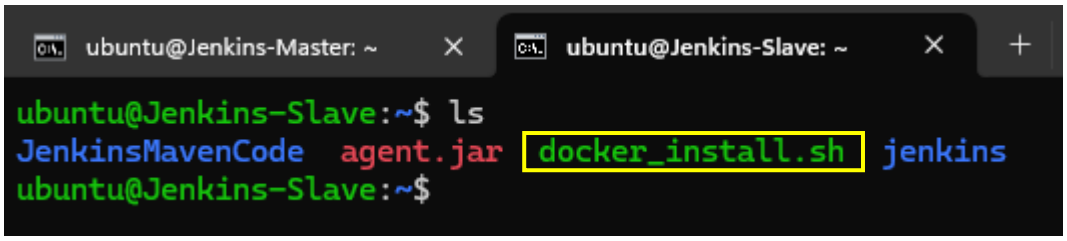



```
[WARNING]
[WARNING] Some problems were encountered while building the effective model for com.edurekademo.tutorial:addressbook:war:2.0
[WARNING] Reporting configuration should be done in <reporting> section, not in maven-site-plugin <configuration> as
reportPlugins parameter. @ line 298, column 40
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] ----------------< com.edurekademo.tutorial:addressbook >----------------
[INFO] Building Vaadin Addressbook example 2.0
[INFO]   from pom.xml
[INFO] --------------------------------[ war ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ addressbook ---
[INFO] Deleting /home/ubuntu/jenkins/workspace/UpstreamJob/target
[INFO]
[INFO] --- enforcer:1.0:enforce (enforce-versions) @ addressbook ---
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ addressbook ---
[INFO] skip non existing resourceDirectory /home/ubuntu/jenkins/workspace/UpstreamJob/src/main/resources
[INFO]
[INFO] --- compiler:3.2:compile (default-compile) @ addressbook ---
```

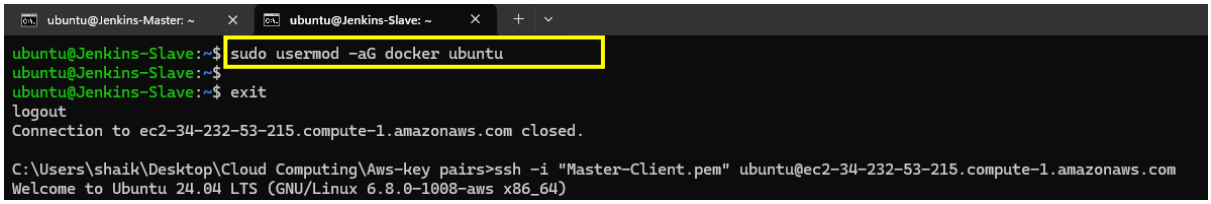
Error I got is docker not found

```
[UpstreamJob] $ /bin/sh -xe /tmp/jenkins12659865534336232270.sh
+ pwd
/home/ubuntu/jenkins/workspace/UpstreamJob
+ git rev-list --tags --max-count=1
+ git describe --tags 76a1eac10cfd9b5361e008c89c6b07ea496d2ad8
+ TAG=1.6
+ echo 1.6
1.6
+ echo 1.6
+ docker build -t salmanfarsi/addressbook:1.6 .
/tmp/jenkins12659865534336232270.sh: 6: docker: not found
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

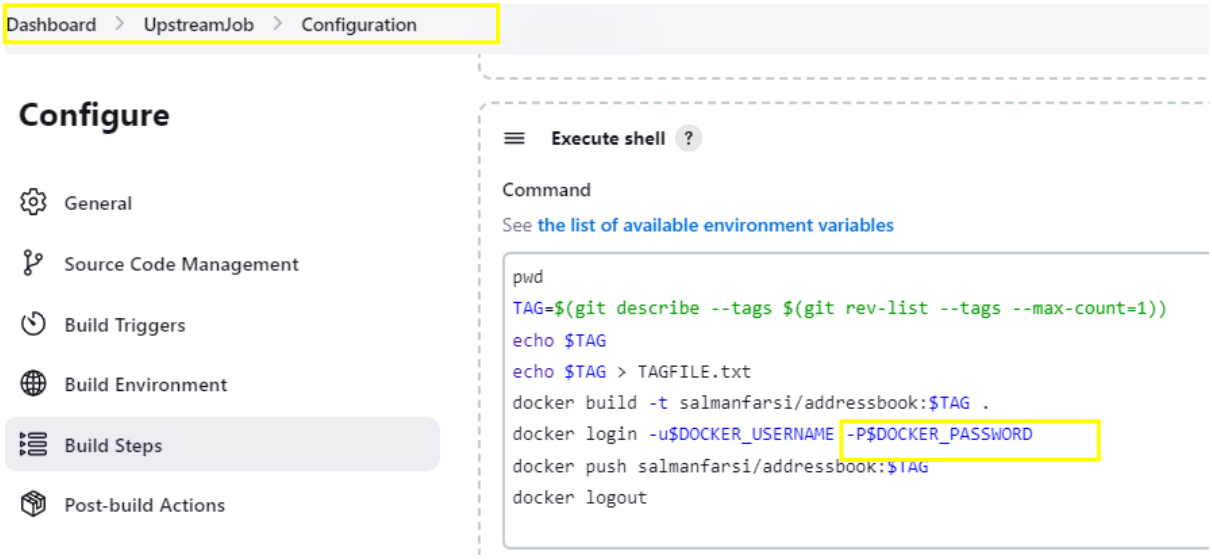
I installed dockers in Jenkins slave, gave permission and ran



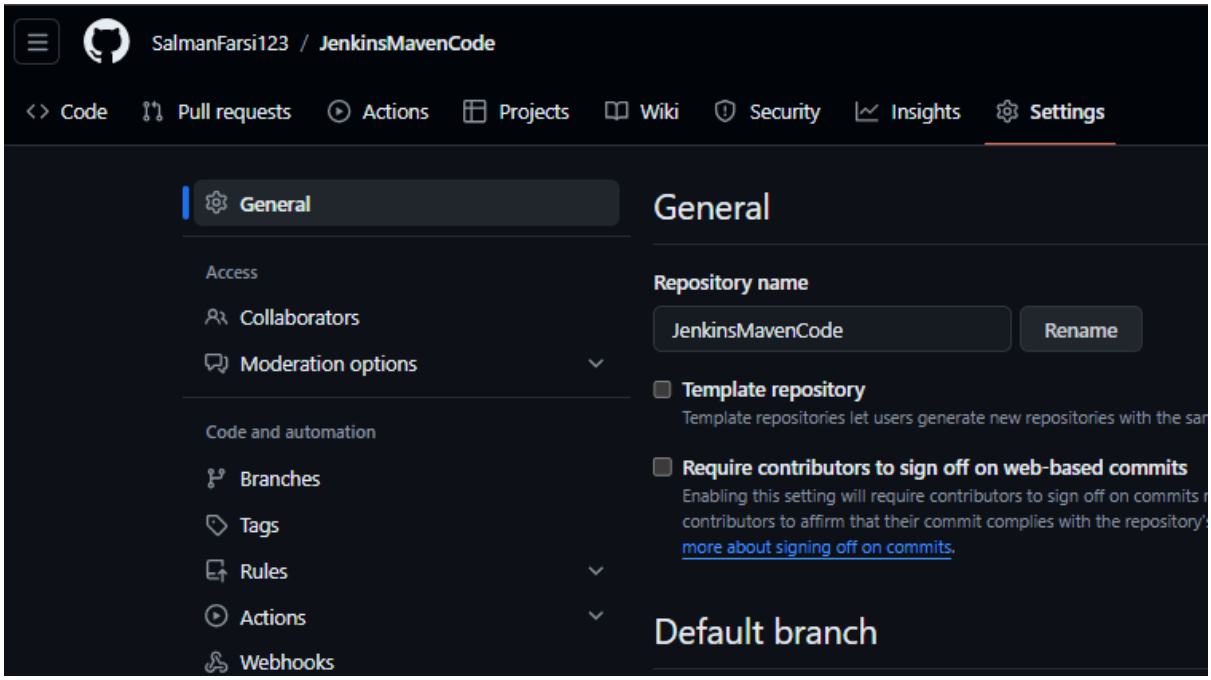
Giving docker permission to ubuntu user



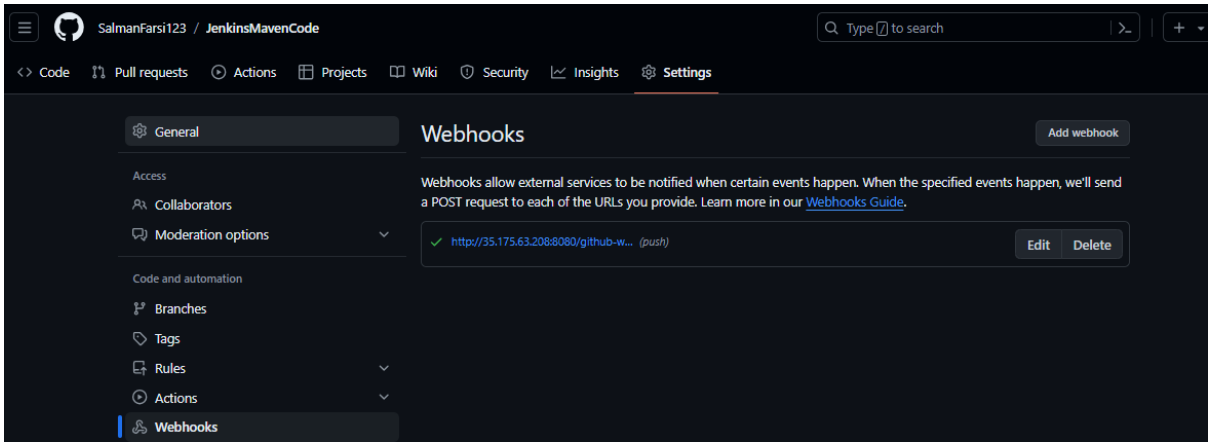
I Updated once again, I thought maybe password is incorrect too.



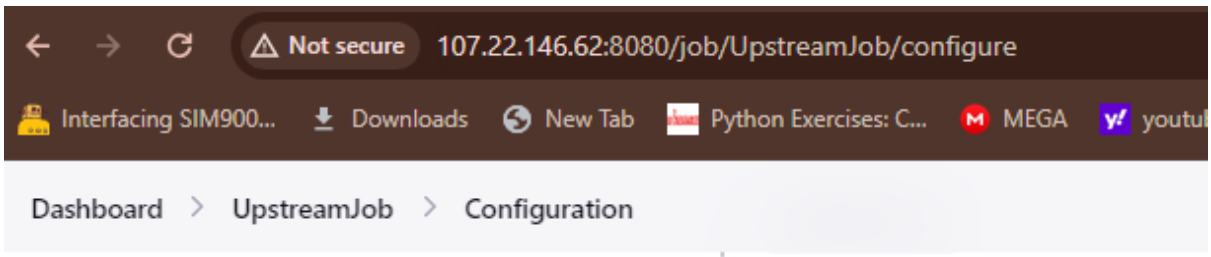
Again I updated because, I stopped the instances and again I launch then the public has been changed so, make sure on this



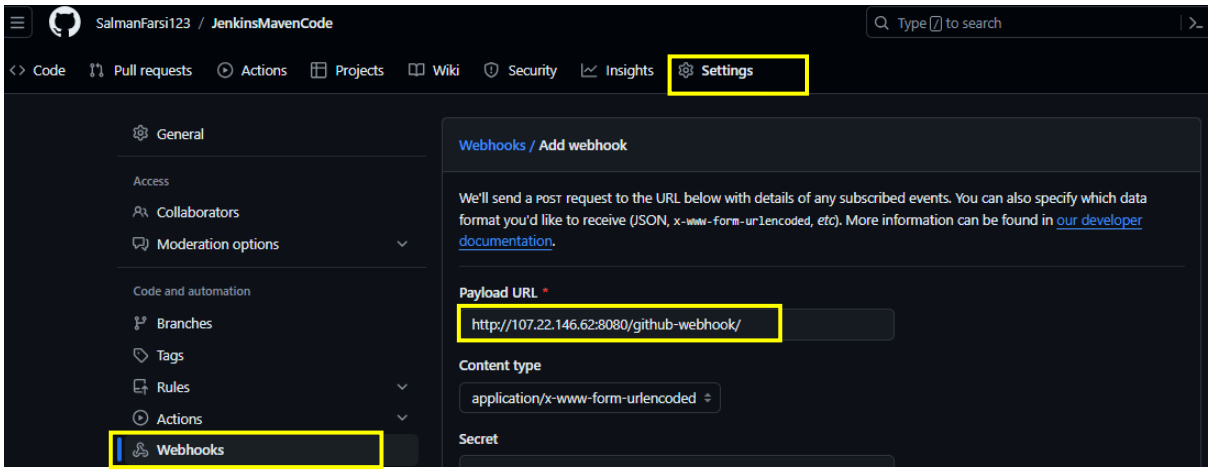
Old Jenkins master public IP



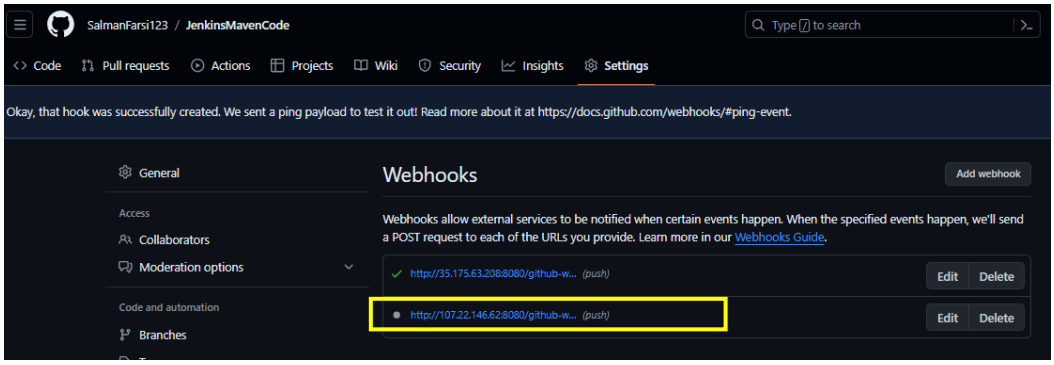
This is new public of Jenkins-Master



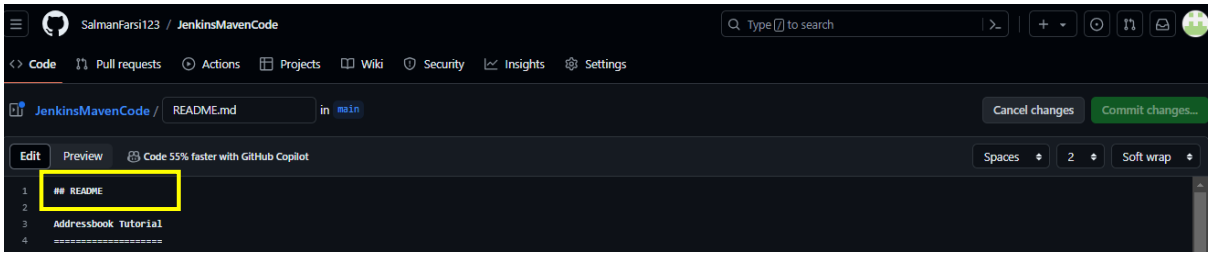
Updated Public Ip of Jenkins Master



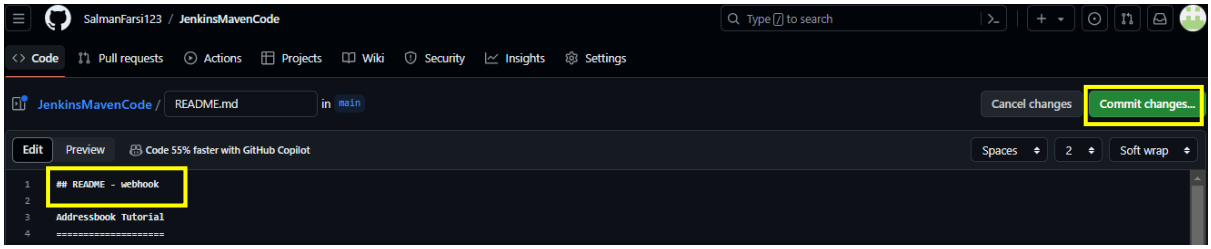
Updated



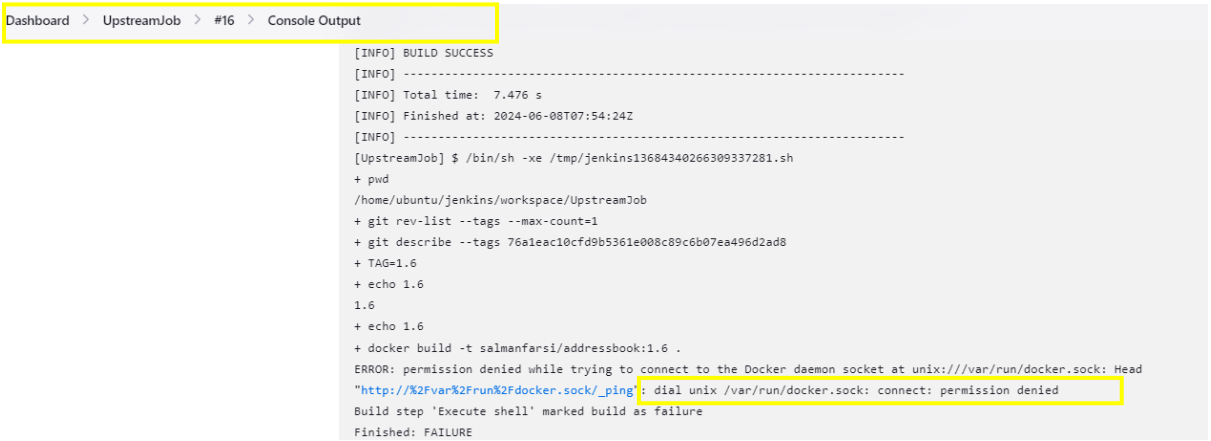
Like this to 👇



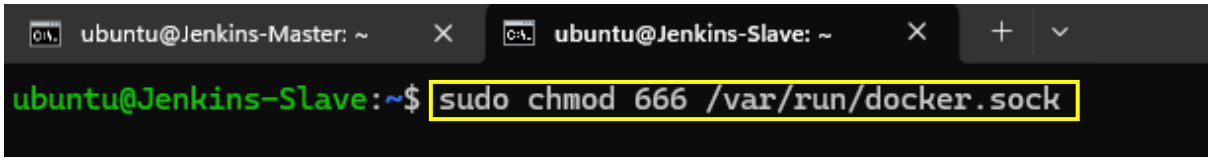
Like this I Make changes and commit changes



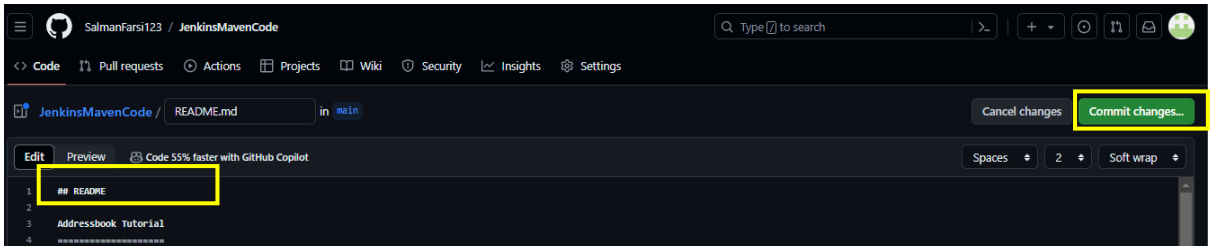
Again I got an error because of docker sock Permission



Go to Jenkins-slave and provide permisions



I Make changes and commit



Again got error because of capital ‘P’ in password part

Dashboard > UpstreamJob > #17 > Console Output

```
#8 DONE 0.0s

#9 [5/6] ADD tomcat-users.xml  /usr/local/tomcat/conf/tomcat-users.xml
#9 DONE 0.0s

#10 [6/6] ADD target/addressbook.war /usr/local/tomcat/webapps/
#10 DONE 0.1s

#11 exporting to image
#11 exporting layers
#11 exporting layers 0.2s done
#11 writing image sha256:6764785665d267188b17b112a9ee10908cbe2fe03a17893dc7bfb9155a3383ae done
#11 naming to docker.io/salmanfarsi/addressbook:1.6 done
#11 DONE 0.2s
+ docker login -usalmanfarsi -P****
unknown shorthand flag: 'P' in -P****
See 'docker login --help'.
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

Change into small p

Dashboard > UpstreamJob > Configuration

Configure

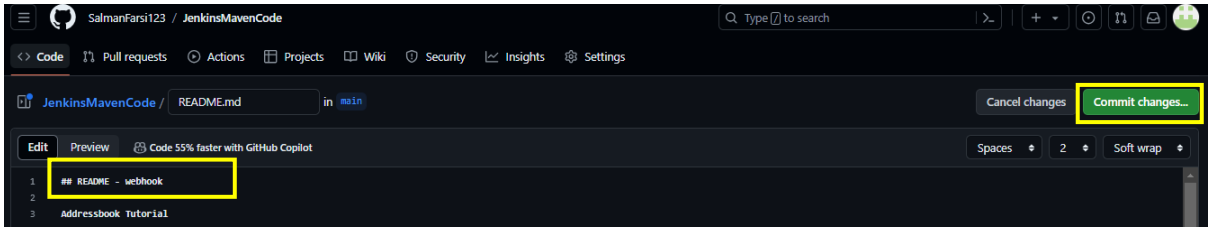
- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Command

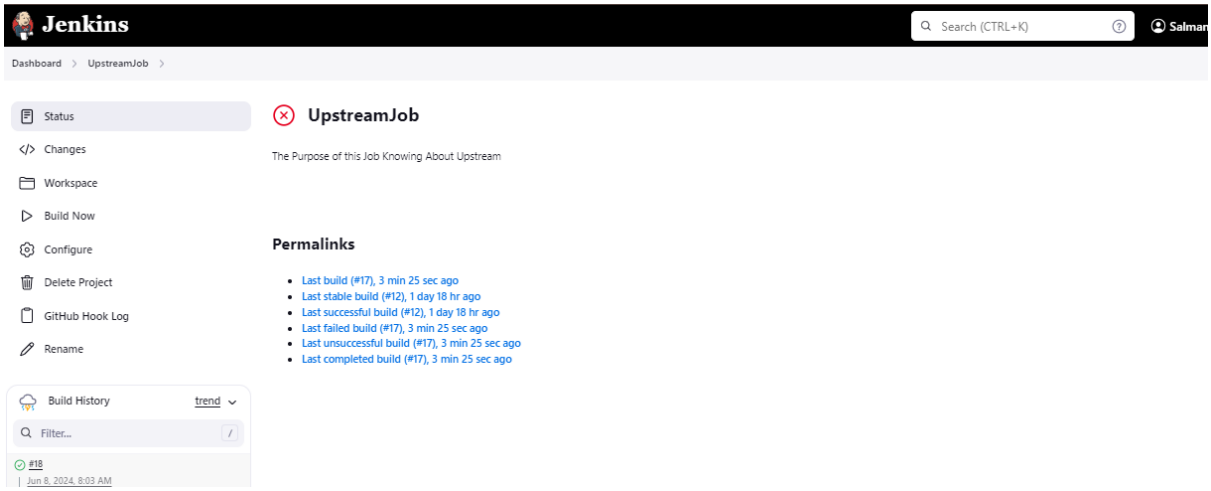
See the list of available environment variables

```
pwd
TAG=$(git describe --tags $(git rev-list --tags --max-count=1))
echo $TAG
echo $TAG > TAGFILE.txt
docker build -t salmanfarsi/addressbook:$TAG .
docker login -u$DOCKER_USERNAME -p$DOCKER_PASSWORD
docker push salmanfarsi/addressbook:$TAG
docker logout
```

Once again I make changes

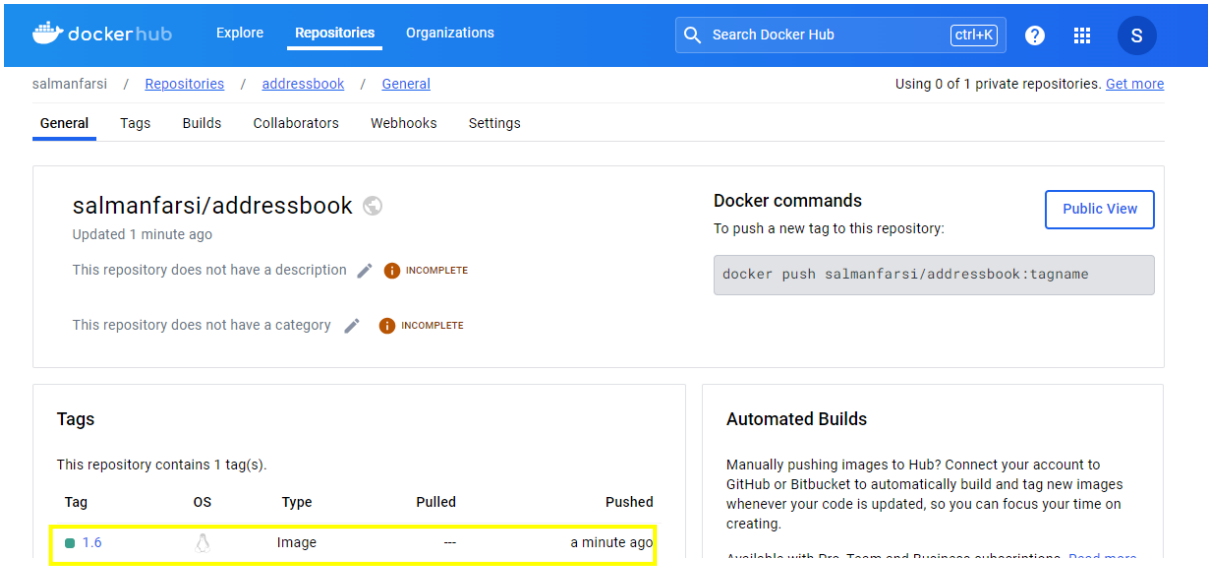


The job 18 ran successfully



Now its pushed to my docker hub account

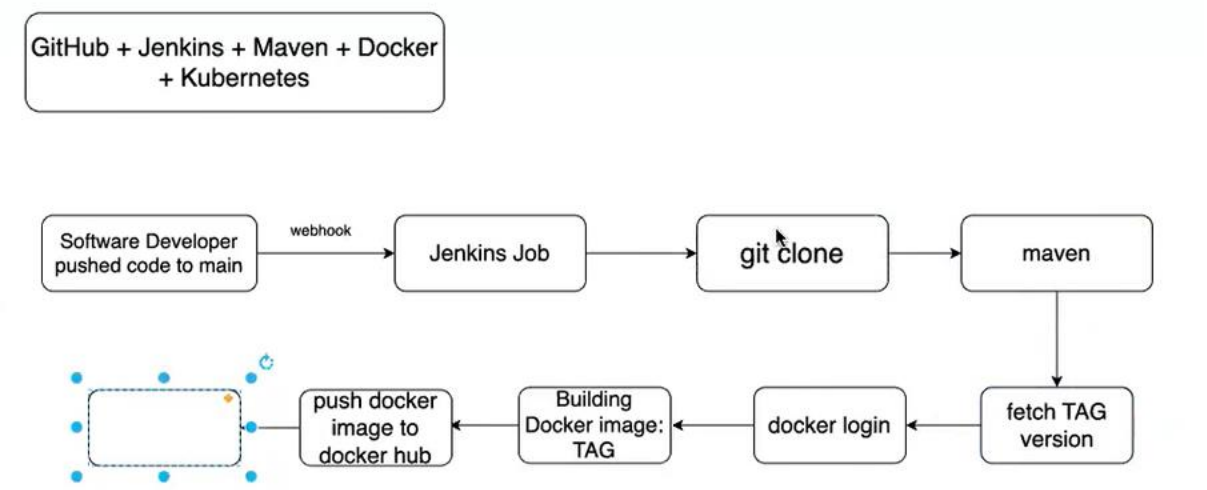
We can see the Image as been pushed into the docker Hub

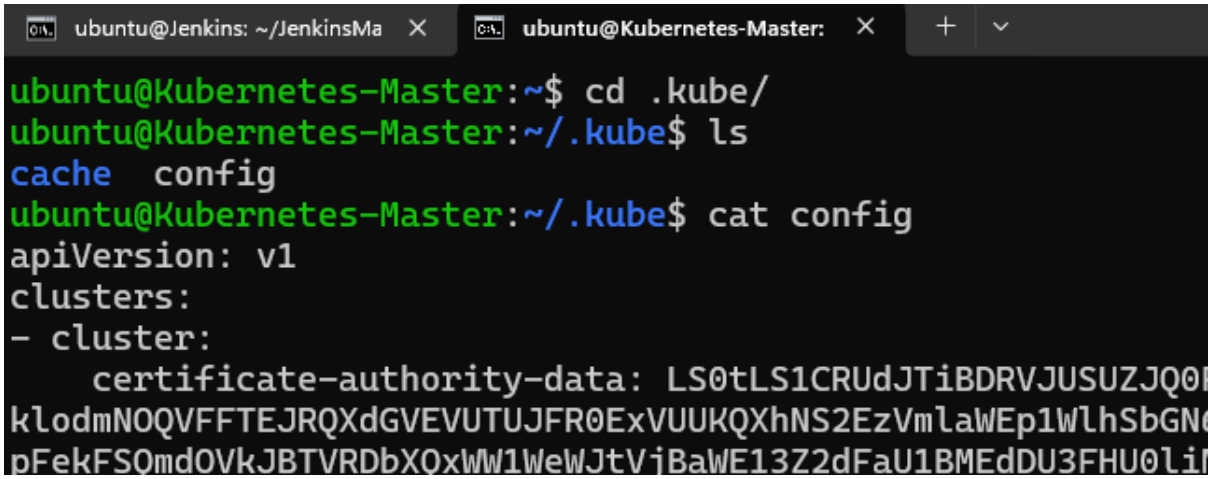


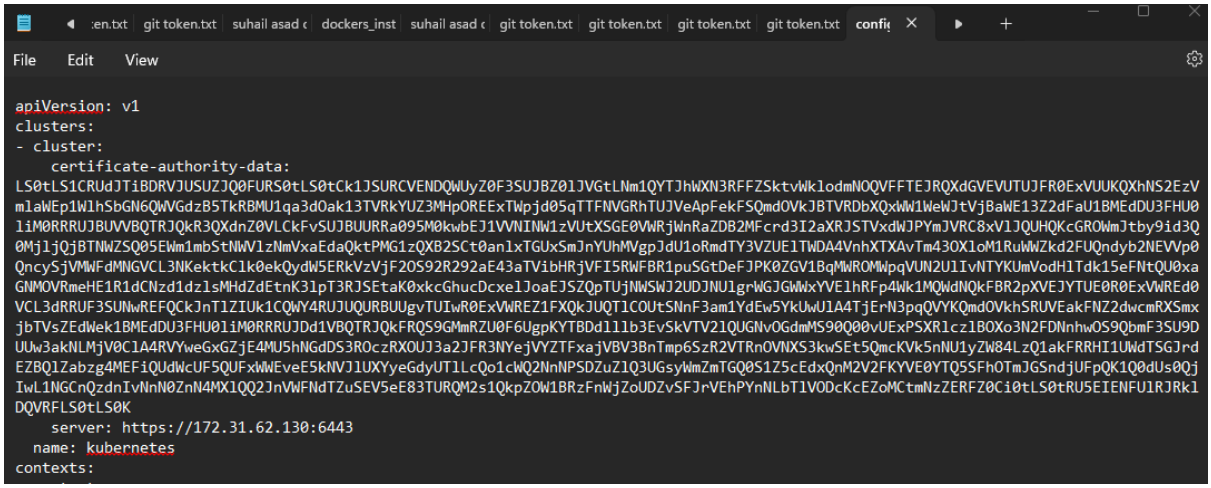
FOLLOW THIS IN ALL INTEGRATION TOOLS PDF:

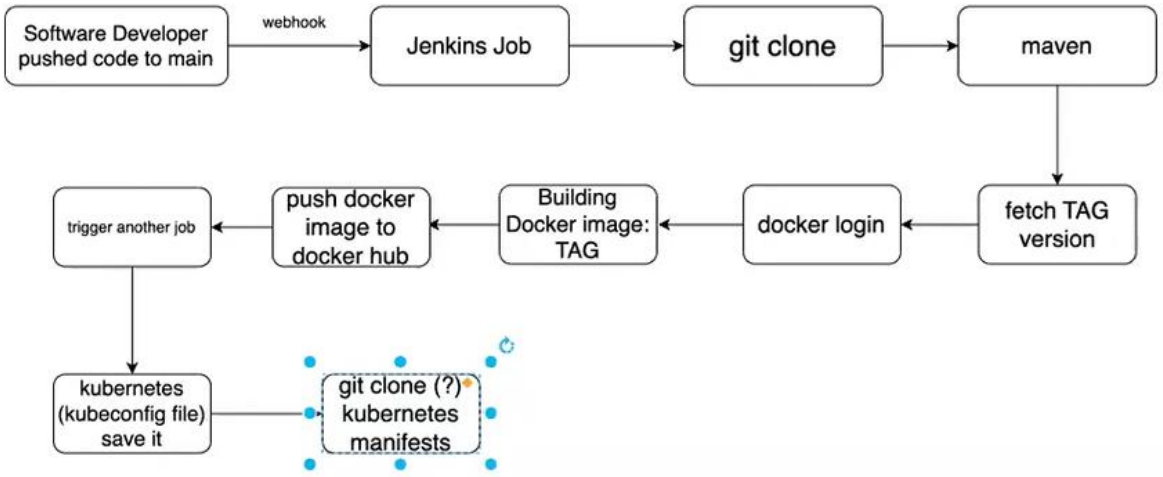
THIS IS INCOMPLETE TOPIC, FOLLOW THE NEXT CLASS

I Captured the Suhail practiced things but it was in complete

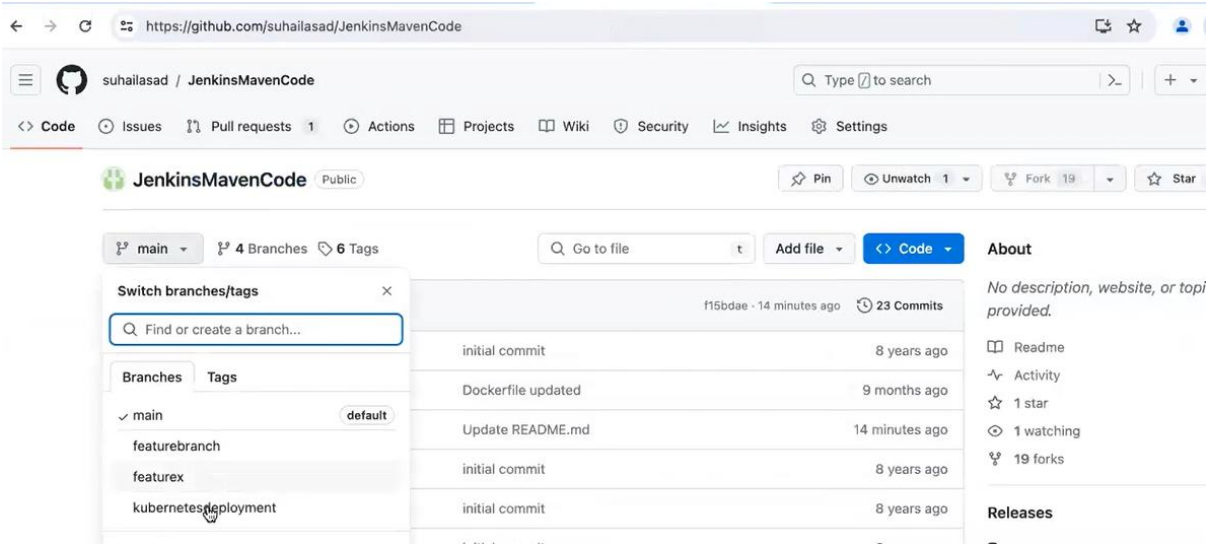


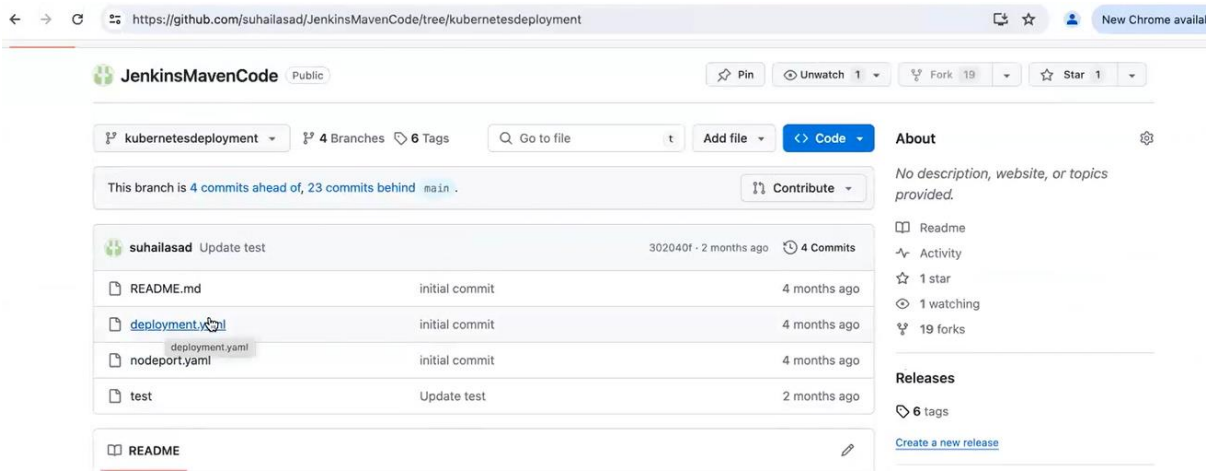

ubuntu@Jenkins: ~/JenkinsMa
ubuntu@Kubernetes-Master:
ubuntu@Kubernetes-Master:~$ cd .kube/
ubuntu@Kubernetes-Master:~/.kube$ ls
cache config
ubuntu@Kubernetes-Master:~/.kube$ cat config
apiVersion: v1
clusters:
- cluster:
certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0
klodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN
pFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0li


:en.txt
git token.txt
suhail asad c
dockers_inst
suhail asad c
git token.txt
git token.txt
git token.txt
git token.txt
confi
File
Edit
View
apiVersion: v1
clusters:
- cluster:
certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ01JVGtLNm1QYTJhWXN3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzV
mlaWEp1WlhSbGN6QWVGdzB5TkRBMU1qa3dOak13TVRkYUZ3MHpOREExTWpjd05qTTFNVGRhTUJVeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0
liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUURRa095M0kwbEJ1VVNINW1zVUtXSGE0VWRjWnRaZDB2MFcrd3I2aXRJSTVxdWJPYmJVRC8xV1JQUHQKcGROWmJtby9id3Q
0MjljQjBTNWZSQ05EWm1mbStNWV1zNmVxaEdaQktPMG1zQXB2SCt0anlxTGUxSmJnYUhMVgpJdU1oRmdTY3VZUE1TWDA4VnhXTXAvTm43OX1oM1RuWWZkd2FUQndyb2NEVVp0
QncySjVMWFdMNGVCL3NKektkC1k0ekQydW5ERkVzVjF2OS92R292aE43aTVibHRjVFI5RWFBR1puSGtDeFJPK0ZGV1BqMWROMWpqVUN2U1IvNTYKUmVodHlTdk15eFNtQU0xa
GNMOVRmeHE1R1dCNzd1dzlsMHdZdEtnK3lpT3RJSEtaK0xkcGhucDcxelJoaEJSZQpTUjNWSWJ2UDJNU1grWGJGWWxYVE1hRFp4Wk1MQWdNQkFBR2pXVEJYTUE0R0ExVWREd0
VCL3dRRUF3SUNwREFQCkJnT1ZIUk1CQWY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJUQTlCOUtSNnF3am1YdEw5YkUwU1A4TjErN3pqQVYKQmdOVkhSRUVEakFNZ2dwcmRXSmx
jbTVsZEdWek1BMEdDU3FHU01iM0RRRUJDd1VBQTRJQkFRQS9GMmRZU0F6UgpKYTBDdlllb3EvSkVTV21QUGNvOGdmMS90Q00vUExPSXR1czlBOXo3N2FDNnhwOS9QbmF3SU9D
UUw3akNLMjV0C1A4RVYweGxGZjE4MU5hNGdDS3ROczRXOUJ3a2JFR3NYejVYZTFxajVBV3BnTmp6SzR2VTRnOVNXS3kwSEt5QmcKVk5nNU1yZW84LzQ1akFRRHI1UWdTSGJrd
EZBQlZabzg4MEFiQUdWcUF5QUFxWWEveE5kNVJlUXYyeGdyUT1LcQo1cWQ2NnNPSDZuZlQ3UGsyWmZmTGQ0S1Z5cEdxQnM2V2FKYVE0YTQ5SFhOTmJGSndjUFpQK1Q0dUs0Qj
IwL1NGCnQzdnIvNnN0ZnN4MX1QQ2JnVWFNdTZuSEV5eE83TURQM2s1QkpZOW1BRzFnWjZoUDZvSFJrVEhPYnNLbT1VODcKcEZoMCtmNzZERFZ0Ci0tLS0tRU5EIENFU1RJRkl
DQVRFLS0tLS0K
server: https://172.31.62.130:6443
name: kubernetes
contexts:

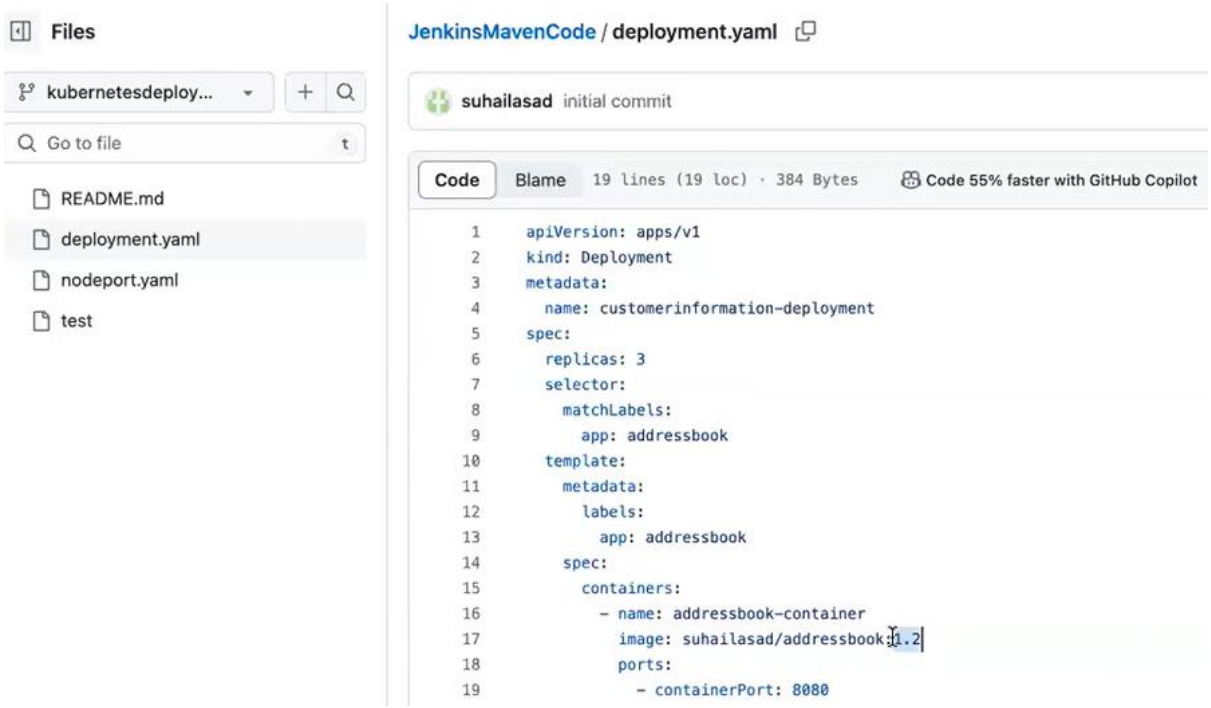


SUHAILASAD PRACTISED STUFF






https://github.com/suhailasad/JenkinsMavenCode/blob/kubernetesdeployment/deployment.yaml
README.md
deployment.yaml
nodeport.yaml
test
Code
Blame
19 lines (19 loc) · 384 Bytes
Code 55% faster with GitHub Copilot
apiVersion: apps/v1
kind: Deployment
metadata:
name: customerinformation-deployment
spec:
replicas: 3
selector:
matchLabels:
app: addressbook
template:
metadata:
labels:
app: addressbook
spec:
containers:
- name: addressbook-container
image: suhailasad/addressbook:1.2
ports:
- containerPort: 8080


Files
kubernetesdeploy...
Go to file
t
README.md
deployment.yaml
nodeport.yaml
test
JenkinsMavenCode / deployment.yaml
suhailasad initial commit
Code
Blame
19 lines (19 loc) · 384 Bytes
Code 55% faster with GitHub Copilot
apiVersion: apps/v1
kind: Deployment
metadata:
name: customerinformation-deployment
spec:
replicas: 3
selector:
matchLabels:
app: addressbook
template:
metadata:
labels:
app: addressbook
spec:
containers:
- name: addressbook-container
image: suhailasad/addressbook:1.2
ports:
- containerPort: 8080

https://github.com/suhailasad/JenkinsMavenCode/blob/kubernetesdeployment/nodeport.yaml

Files

kubernetesdeploy...

Go to file t

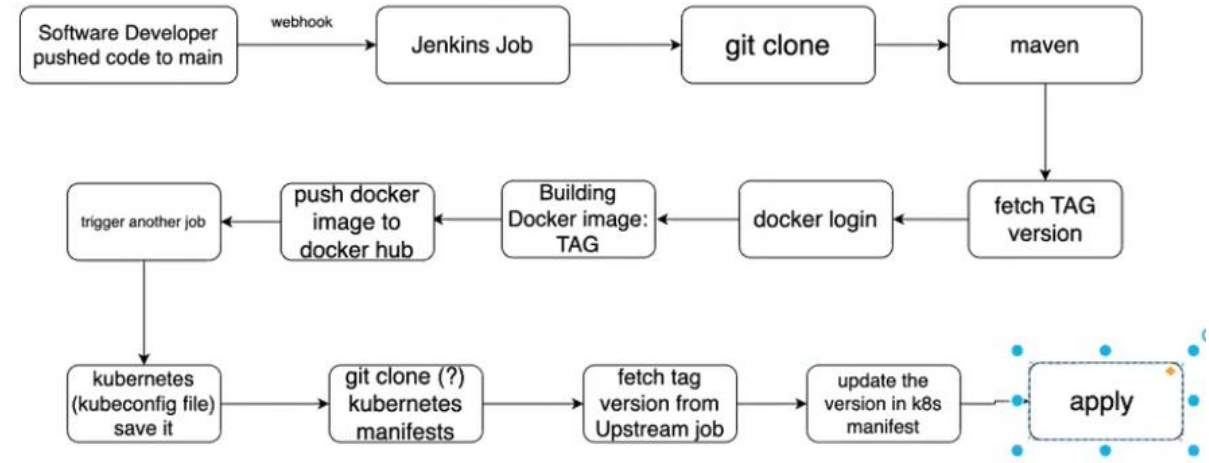
- README.md
- deployment.yaml
- nodeport.yaml
- test

JenkinsMavenCode / nodeport.yaml

suhailasad initial commit

Code | Blame | 13 lines (13 loc) · 221 Bytes | Code 55% faster with GitHub Copilot

```yaml
apiVersion: v1
kind: Service
metadata:
  name: customerinformation-nodeport
spec:
  selector:
    app: addressbook
  type: NodePort
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
      nodePort: 30089
```



Not Secure http://3.129.5.33:8080/job/UpstreamJob/8/console

New Chrome available

Dashboard > UpstreamJob > #8 > Console Output

Timestamps View as plain text

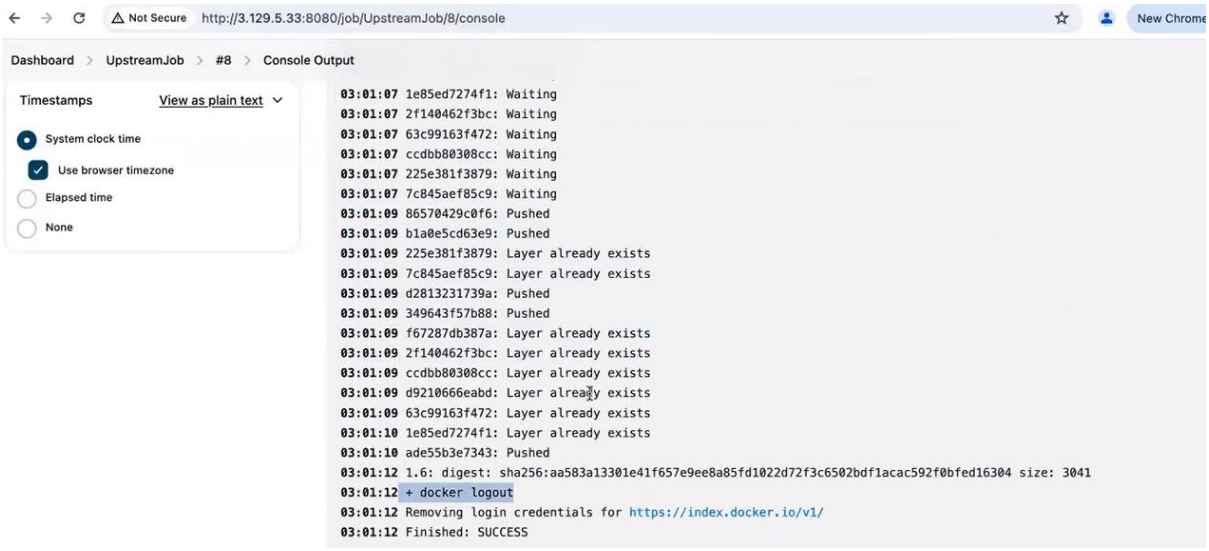
System clock time

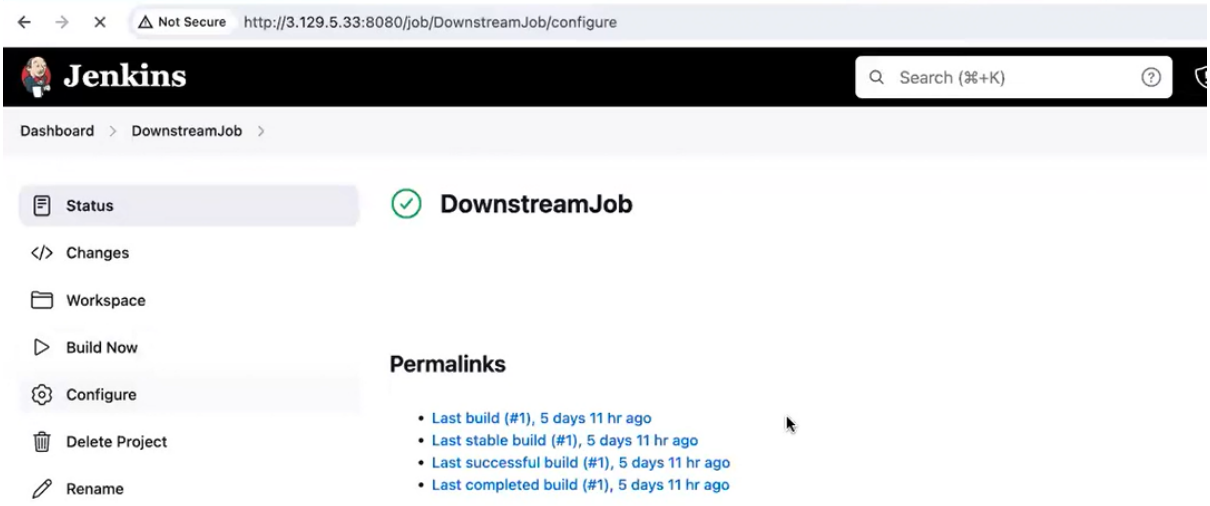
Use browser timezone

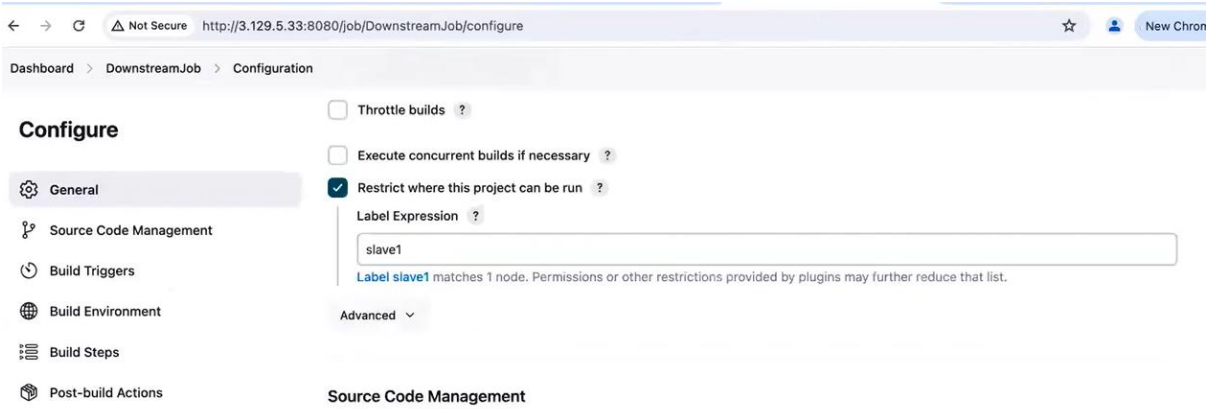
Elapsed time

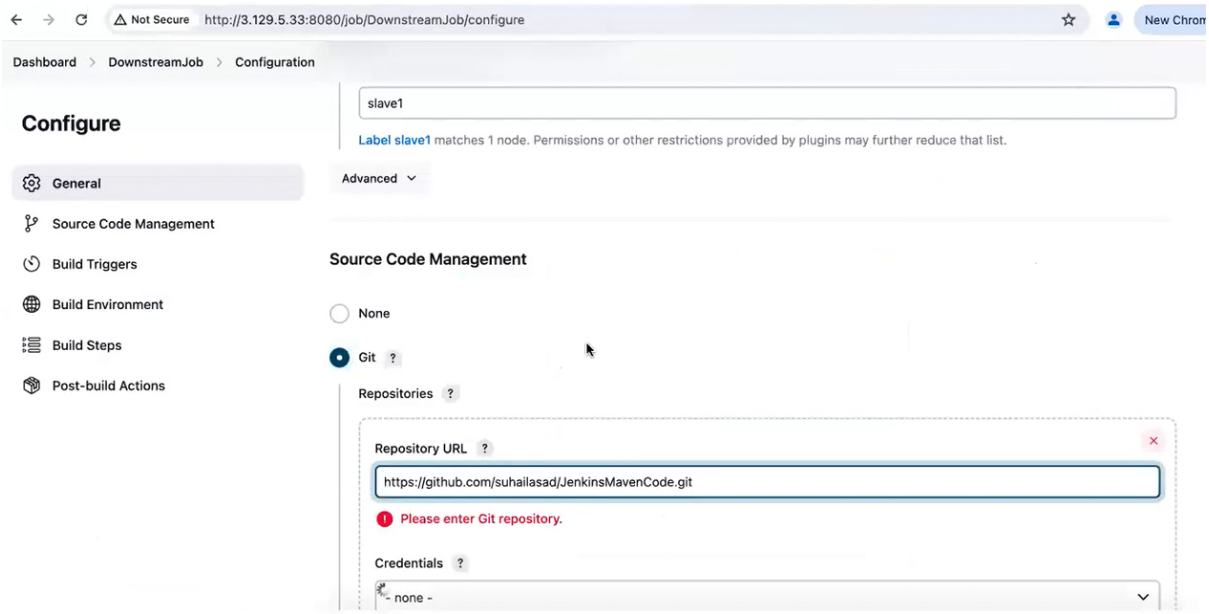
None

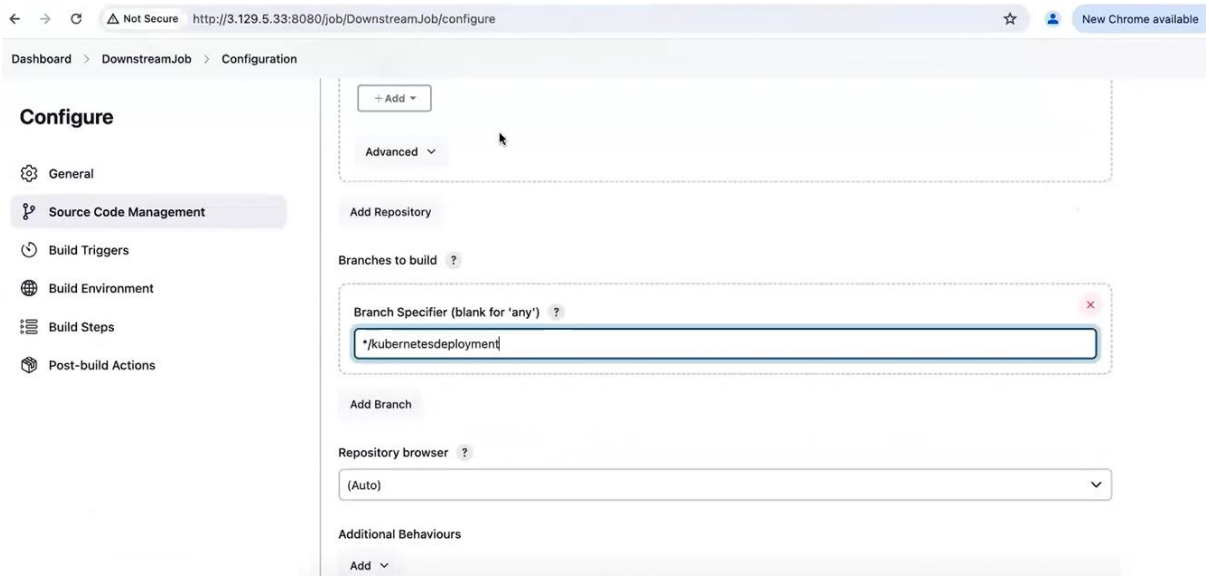
```
03:00:53 #4 [1/6] FROM docker.io/library/tomcat:9.0.45-jdk11-adoptopenjdk-
hotspot@sha256:b76071b553b248ea9560636e75c444b6b28b21c4520f6801ccceef9740633bf7
03:00:53 #4 resolve docker.io/library/tomcat:9.0.45-jdk11-adoptopenjdk-
hotspot@sha256:b76071b553b248ea9560636e75c444b6b28b21c4520f6801ccceef9740633bf7 0.0s done
03:00:53 #4 ...
03:00:53
03:00:53 #5 [internal] load build context
03:00:53 #5 transferring context: 16.54MB 0.2s done
03:00:53 #5 DONE 0.2s
03:00:53
03:00:53 #4 [1/6] FROM docker.io/library/tomcat:9.0.45-jdk11-adoptopenjdk-
hotspot@sha256:b76071b553b248ea9560636e75c444b6b28b21c4520f6801ccceef9740633bf7
03:00:53 #4 sha256:b76071b553b248ea9560636e75c444b6b28b21c4520f6801ccceef9740633bf7 1.21kB / 1.21kB done
03:00:53 #4 sha256:e7281e9ea44afc5e4c2e5b44f64d1d7fd0e4834d415c4e8d0bfaccc29b238e8d 2.00kB / 2.00kB done
03:00:53 #4 sha256:86ffac52d277c03cdacbc7f0af6fbc054218068d555977c23e00ddcdf561cb98 14.39kB / 14.39kB done
03:00:53 #4 sha256:345e3491a907bb7c6f1bdddcf4a94284b8b6ddd77eb7d93f09432b17b20f2bbe 9.44MB / 28.54MB 0.2s
03:00:53 #4 sha256:5e9250ddb7d0fa6d13302c7c3e6a0aa40390e42424caed1e5289077ee4054709 187B / 187B 0.2s done
03:00:53 #4 sha256:57671312ef6fdbecf340e5fed0fb0863350cd806c92b1fdd7978adbd02afc5c3 851B / 851B 0.1s done
03:00:53 #4 sha256:592ec2d7c1370aca6ef48ee1d20fbb3a4c148db400adc5dc575778c681f5dba0 0B / 16.03MB 0.2s
03:00:53 #4 sha256:ce0a2410aba2adb3fc36d0afef53fdb794de060dfa24ff52424735d165af5ee9 0B / 193.65MB 0.2s
03:00:53 #4 sha256:345e3491a907bb7c6f1bdddcf4a94284b8b6ddd77eb7d93f09432b17b20f2bbe 24.12MB / 28.54MB 0.4s
03:00:53 #4 sha256:592ec2d7c1370aca6ef48ee1d20fbb3a4c148db400adc5dc575778c681f5dba0 6.29MB / 16.03MB 0.4s
03:00:53 #4 sha256:345e3491a907bb7c6f1bdddcf4a94284b8b6ddd77eb7d93f09432b17b20f2bbe 28.54MB / 28.54MB 0.5s done
03:00:53 #4 sha256:592ec2d7c1370aca6ef48ee1d20fbb3a4c148db400adc5dc575778c681f5dba0 16.03MB / 16.03MB 0.6s
03:00:53 #4 sha256:ce0a2410aba2adb3fc36d0afef53fdb794de060dfa24ff52424735d165af5ee9 15.73MB / 193.65MB 0.6s
03:00:53 #4 sha256:7267948abfc47d400ccf2e9273d7a6ce5eeceea00dbb0d2c27d250e2200954da 0B / 172B 0.6s
03:00:53 #4 sha256:592ec2d7c1370aca6ef48ee1d20fbb3a4c148db400adc5dc575778c681f5dba0 16.03MB / 16.03MB 0.6s done
```

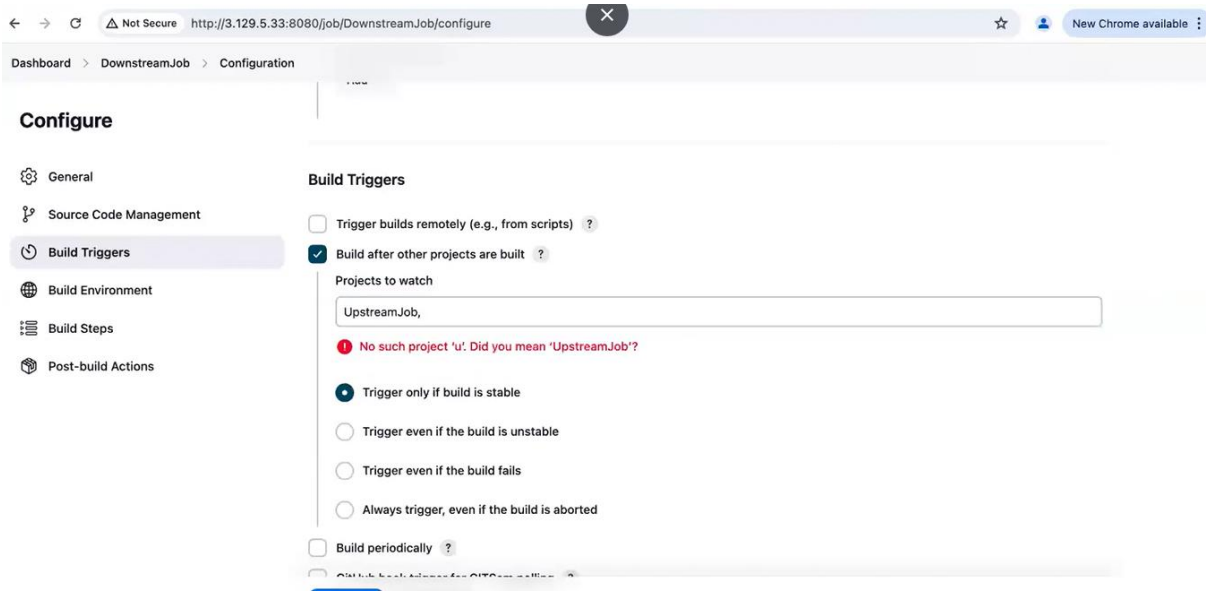

Not Secure http://3.129.5.33:8080/job/UpstreamJob/8/console
Dashboard > UpstreamJob > #8 > Console Output
Timestamps View as plain text
System clock time
Use browser timezone
Elapsed time
None
03:01:07 1e85ed7274f1: Waiting
03:01:07 2f140462f3bc: Waiting
03:01:07 63c99163f472: Waiting
03:01:07 ccdbb80308cc: Waiting
03:01:07 225e381f3879: Waiting
03:01:07 7c845aef85c9: Waiting
03:01:09 86570429c0f6: Pushed
03:01:09 b1a0e5cd63e9: Pushed
03:01:09 225e381f3879: Layer already exists
03:01:09 7c845aef85c9: Layer already exists
03:01:09 d2813231739a: Pushed
03:01:09 349643f57b88: Pushed
03:01:09 f67287db387a: Layer already exists
03:01:09 2f140462f3bc: Layer already exists
03:01:09 ccdbb80308cc: Layer already exists
03:01:09 d9210666eabd: Layer already exists
03:01:09 63c99163f472: Layer already exists
03:01:10 1e85ed7274f1: Layer already exists
03:01:10 ade55b3e7343: Pushed
03:01:12 1.6: digest: sha256:aa583a13301e41f657e9ee8a85fd1022d72f3c6502bdf1acac592f0bfed16304 size: 3041
03:01:12 + docker logout
03:01:12 Removing login credentials for https://index.docker.io/v1/
03:01:12 Finished: SUCCESS

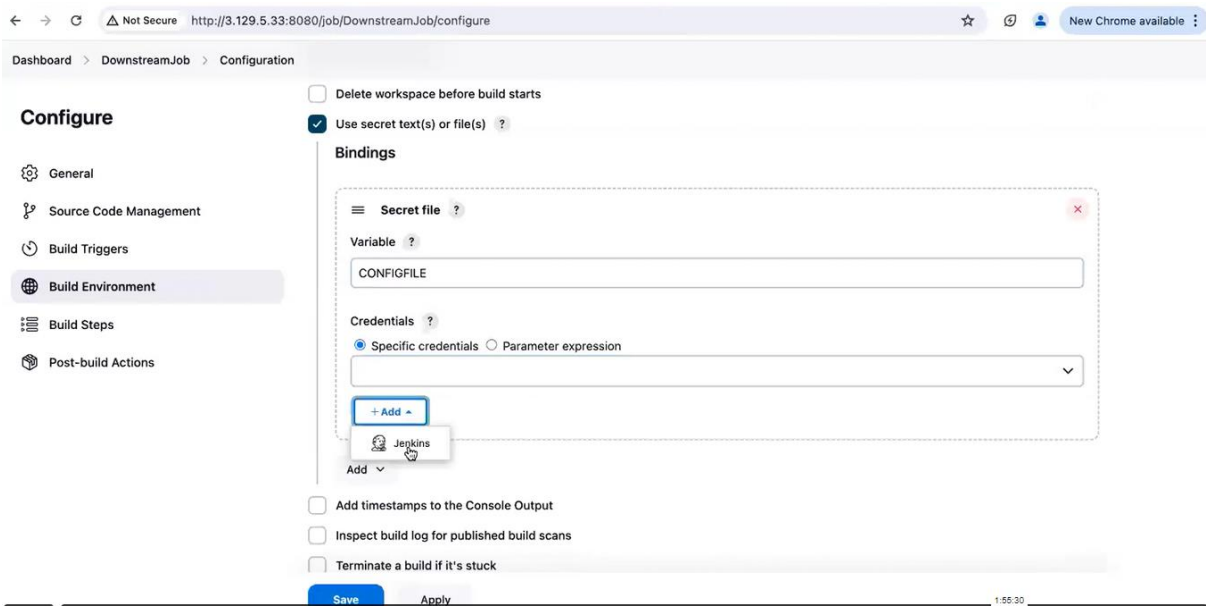

Not Secure http://3.129.5.33:8080/job/DownstreamJob/configure
Jenkins
Search (⌘+K)
Dashboard > DownstreamJob >
Status
Changes
Workspace
Build Now
Configure
Delete Project
Rename
DownstreamJob
Permalinks
Last build (#1), 5 days 11 hr ago
Last stable build (#1), 5 days 11 hr ago
Last successful build (#1), 5 days 11 hr ago
Last completed build (#1), 5 days 11 hr ago

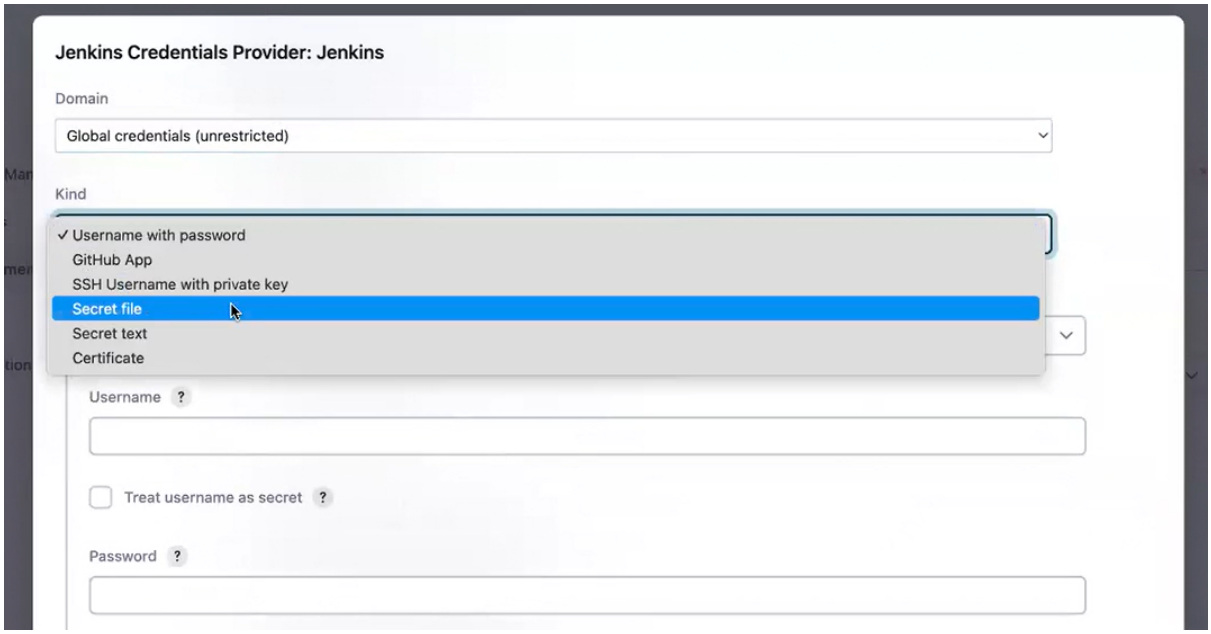

Not Secure http://3.129.5.33:8080/job/DownstreamJob/configure
Dashboard > DownstreamJob > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Throttle builds
Execute concurrent builds if necessary
Restrict where this project can be run
Label Expression
slave1
Label slave1 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.
Advanced
Source Code Management


Not Secure
http://3.129.5.33:8080/job/DownstreamJob/configure
Dashboard
DownstreamJob
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
slave1
Label slave1 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.
Advanced
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/suhailasad/JenkinsMavenCode.git
Please enter Git repository.
Credentials
- none -

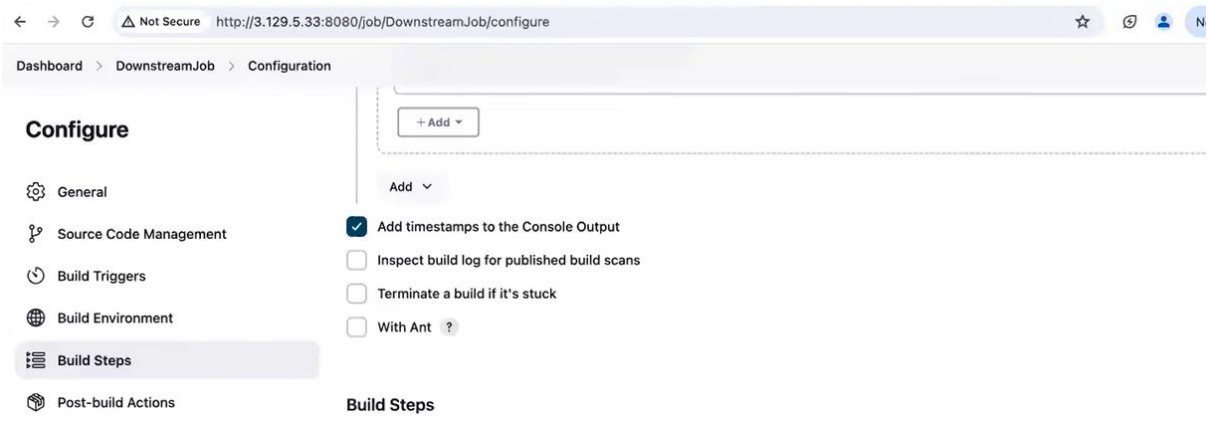

Not Secure
http://3.129.5.33:8080/job/DownstreamJob/configure
New Chrome available
Dashboard
DownstreamJob
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
+ Add
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/kubernetesdeployment
Add Branch
Repository browser
(Auto)
Additional Behaviours
Add

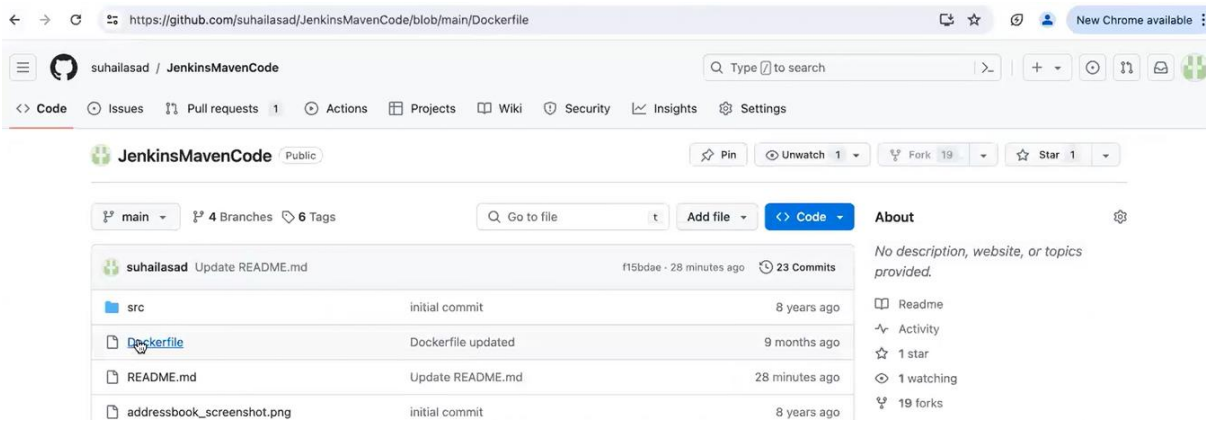

Not Secure
http://3.129.5.33:8080/job/DownstreamJob/configure
New Chrome available
Dashboard
DownstreamJob
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Build Triggers
Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Projects to watch
UpstreamJob,
No such project 'u'. Did you mean 'UpstreamJob'?
Trigger only if build is stable
Trigger even if the build is unstable
Trigger even if the build fails
Always trigger, even if the build is aborted
Build periodically

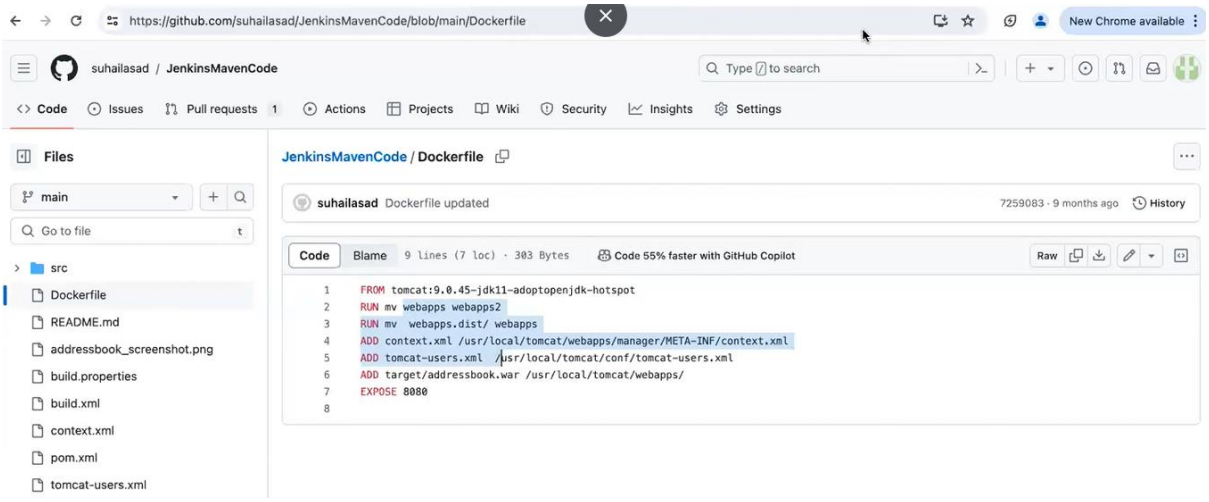

Not Secure
http://3.129.5.33:8080/job/DownstreamJob/configure
New Chrome available
Dashboard
DownstreamJob
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Delete workspace before build starts
Use secret text(s) or file(s)
Bindings
Secret file
Variable
CONFIGFILE
Credentials
Specific credentials
Parameter expression
+ Add
Jenkins
Add
Add timestamps to the Console Output
Inspect build log for published build scans
Terminate a build if it's stuck
Save
Apply


Jenkins Credentials Provider: Jenkins
Domain
Global credentials (unrestricted)
Kind
✓ Username with password
GitHub App
SSH Username with private key
Secret file
Secret text
Certificate
Username
Treat username as secret
Password


Jenkins Credentials Provider: Jenkins
Secret file
Scope
Global (Jenkins, nodes, items, all child items, etc)
File
Choose file
config
ID
KUbeconfig
Description
kubernetes kubeconfig file
Cancel
Add


Not Secure
http://3.129.5.33:8080/job/DownstreamJob/configure
Dashboard > DownstreamJob > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
+ Add
Add
Add timestamps to the Console Output
Inspect build log for published build scans
Terminate a build if it's stuck
With Ant
Build Steps


https://github.com/suhailasad/JenkinsMavenCode/blob/main/Dockerfile
New Chrome available
suhailasad / JenkinsMavenCode
Type / to search
Code
Issues
Pull requests 1
Actions
Projects
Wiki
Security
Insights
Settings
JenkinsMavenCode
Public
Pin
Unwatch 1
Fork 19
Star 1
main
4 Branches
6 Tags
Go to file
t
Add file
Code
About
No description, website, or topics provided.
suhailasad Update README.md
f15bdae · 28 minutes ago
23 Commits
src
initial commit
8 years ago
Dockerfile
Dockerfile updated
9 months ago
README.md
Update README.md
28 minutes ago
addressbook_screenshot.png
initial commit
8 years ago
Readme
Activity
1 star
1 watching
19 forks


https://github.com/suhailasad/JenkinsMavenCode/blob/main/Dockerfile
New Chrome available
suhailasad / JenkinsMavenCode
Type / to search
Code
Issues
Pull requests 1
Actions
Projects
Wiki
Security
Insights
Settings
Files
main
Go to file
t
src
Dockerfile
README.md
addressbook_screenshot.png
build.properties
build.xml
context.xml
pom.xml
tomcat-users.xml
JenkinsMavenCode / Dockerfile
suhailasad Dockerfile updated
7259083 · 9 months ago
History
Code
Blame
9 lines (7 loc) · 303 Bytes
Code 55% faster with GitHub Copilot
Raw
1 FROM tomcat:9.0.45-jdk11-adoptopenjdk-hotspot
2 RUN mv webapps webapps2
3 RUN mv webapps.dist/ webapps
4 ADD context.xml /usr/local/tomcat/webapps/manager/META-INF/context.xml
5 ADD tomcat-users.xml /usr/local/tomcat/conf/tomcat-users.xml
6 ADD target/addressbook.war /usr/local/tomcat/webapps/
7 EXPOSE 8080
8