Create a pod defintion file to start nginx pod with a name webserver

**POD**

vim pod.yml

---

apiVersion: v1

kind: Pod

metadata:

name: nginx-pod

labels:

type: proxy

author: company

spec:

containers:

- name: webserver

image: nginx

...

Create pod from the above file

kubectl apply -f pod.yml

To check the list of pods

kubectl get pods

**Deployment**

**Creation of the nginx ( webserver)**

|  |
| --- |
|  |
| vi nginx.yaml | |
|  | | --- |
|  | | apiVersion: apps/v1 |
|  | | kind: Deployment |
|  | | metadata: |
|  | | name: nginx |
|  | | spec: |
|  | | selector: |
|  | | matchLabels: |
|  | | run: nginx |
|  | | replicas: 2 |
|  | | template: |
|  | | metadata: |
|  | | labels: |
|  | | run: nginx |
|  | | spec: |
|  | | containers: |
|  | | - name: nginx |
|  | | image: nginx:1.7.9 |
|  | | ports: |
|  | | - containerPort: 80  … |
| ============= | | ========================================================= |
|  | | $ kubectl apply -f nginx.yaml  Kubectl get pods  Kubectl get deployment |
|  | | ===================================================== |
|  | |  |
|  | | vi nginx.svs.yaml |
|  | | --- |
|  | | apiVersion: v1 |
|  | | kind: Service |
|  | | metadata: |
|  | | name: nginx |
|  | | labels: |
|  | | run: nginx |
|  | | spec: |
|  | | ports: |
|  | | - port: 80 |
|  | | protocol: TCP |
|  | | selector: |
|  | | run: nginx |
|  | | type: LoadBalancer  … |
|  | | === |
|  | | $ kubectl apply -f nginx.svc.yaml |
|  | | ===== |
|  | | $ kubectl get svc |
|  | | === |
|  | | $ kubectl delete pod (pod name) |
|  | | === |
|  | | $ kubectl delete svc (service name) |
| ========================= | | ============================================= |
|  | |  |

**Creation of the Jenkins installation:**

vi jenkins.yaml

======================

apiVersion: apps/v1

kind: Deployment

metadata:

name: jenkins

spec:

replicas: 1

selector:

matchLabels:

app: jenkins

template:

metadata:

labels:

app: jenkins

spec:

containers:

- name: jenkins

image: jenkins/jenkins:lts

ports:

- name: http-port

containerPort: 8080

- name: jnlp-port

containerPort: 50000

volumeMounts:

- name: jenkins-vol

mountPath: /var/jenkins\_vol

volumes:

- name: jenkins-vol

emptyDir: {}

. ..

=================================

$ kubectl apply -f jenkins.yaml

=====================================

vi jenkins-service.yaml

---

apiVersion: v1

kind: Service

metadata:

name: jenkins

spec:

type: NodePort

ports:

- port: 8080

targetPort: 8080

nodePort: 30000

selector:

app: jenkins

type: LoadBalancer

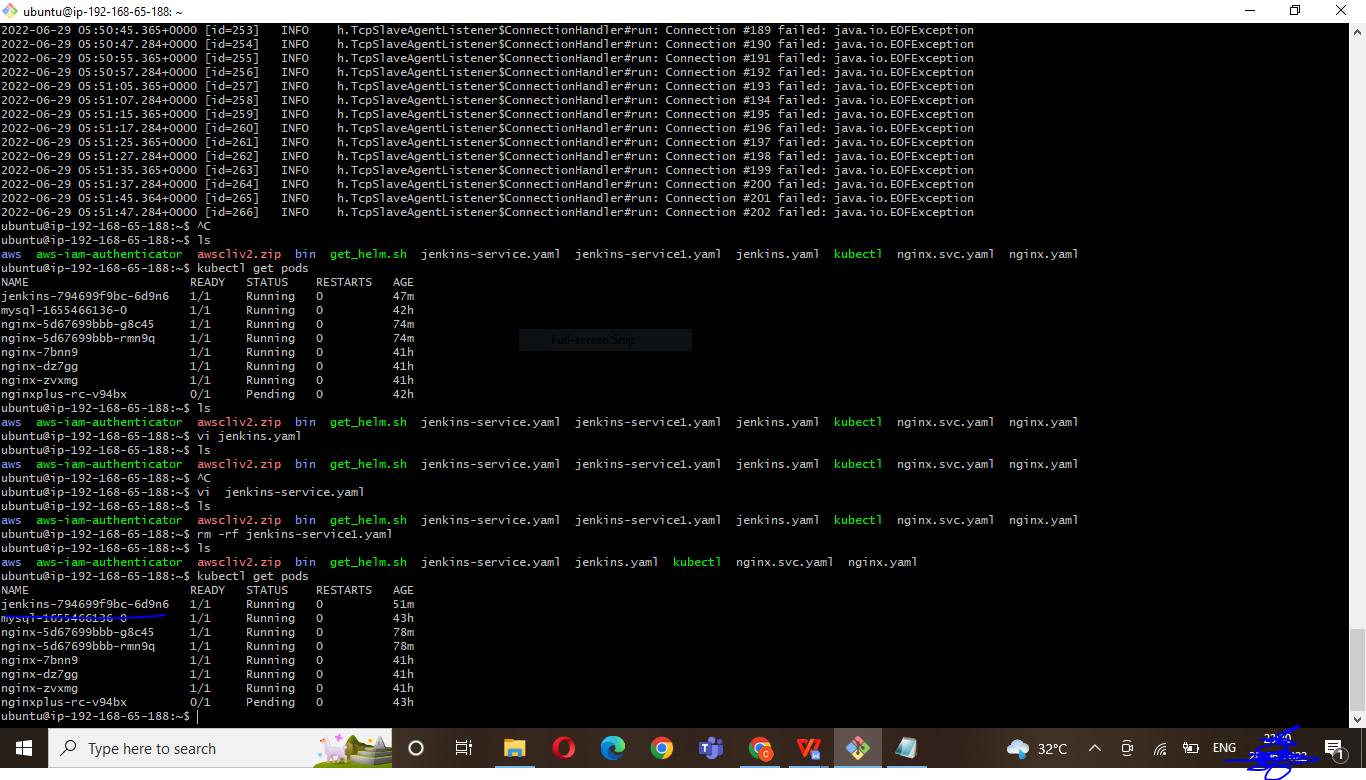
. ..

---

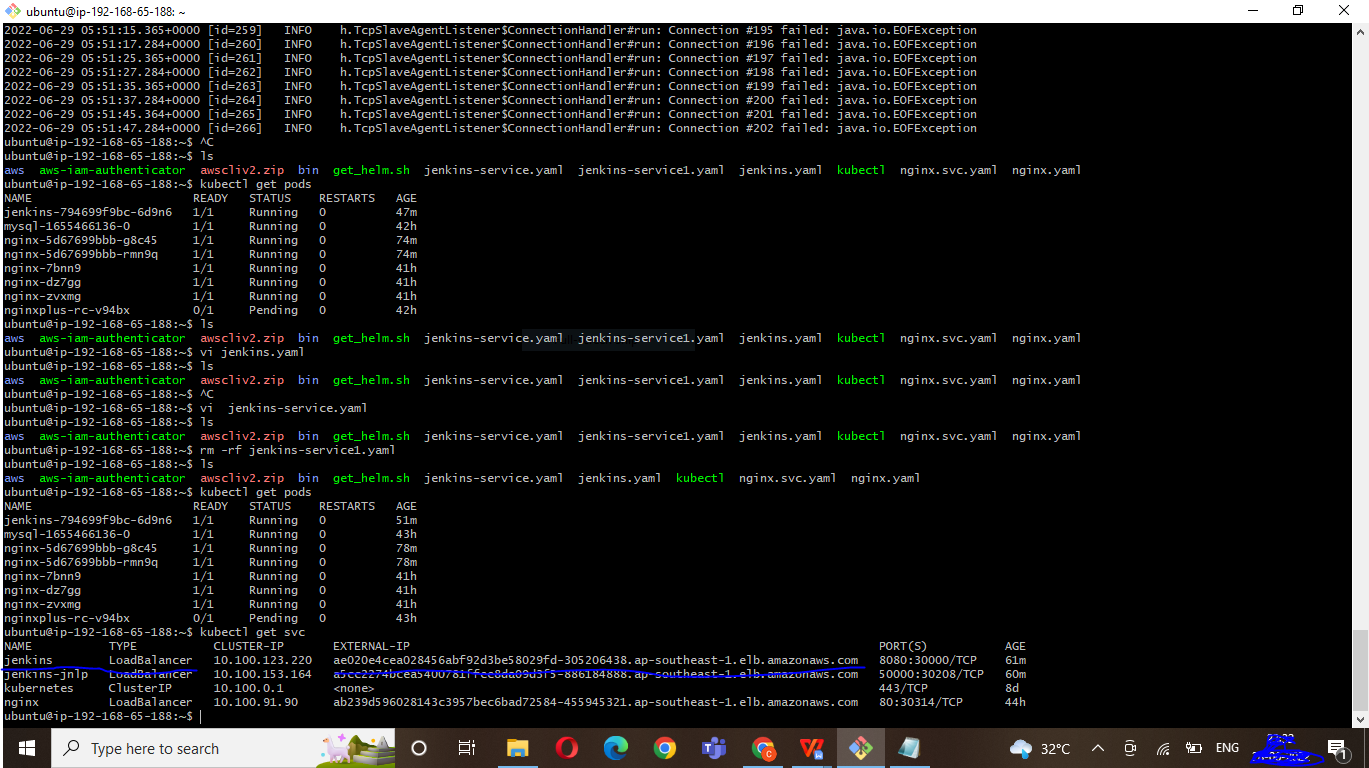
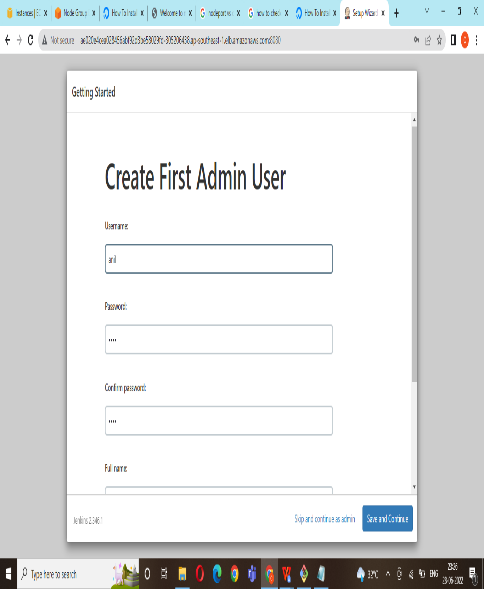
$ kubectl apply -f jenkins-service. yaml

$ kubectl get pods

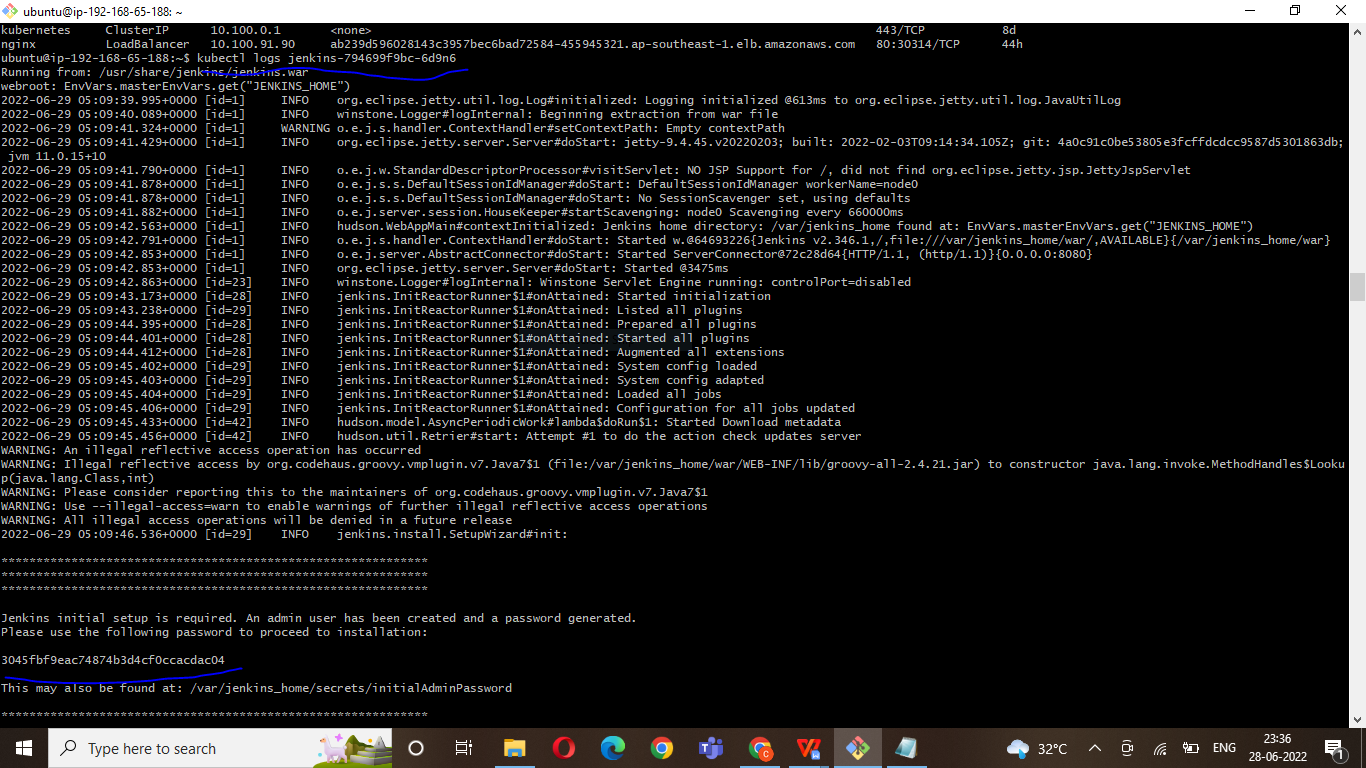
$ kubectl get pods -n jenkins



$ Kubectl get svc



$ kubectl logs jenkins-6fb994cfc5-twnvn ( give here jenkins pod full name)



==================================================================================================================================

**Creation of the SonarQube install**

|  |
| --- |
| Vi sonar.yaml  ------------------ |
|  | --- |
|  | apiVersion: apps/v1 |
|  | kind: Deployment |
|  | metadata: |
|  | name: sonarqube |
|  | spec: |
|  | replicas: 1 |
|  | selector: |
|  | matchLabels: |
|  | app: sonarqube |
|  | template: |
|  | metadata: |
|  | labels: |
|  | app: sonarqube |
|  | spec: |
|  | containers: |
|  | - name: sonarqube |
|  | image: sonarqube: latest |
|  | ports: |
|  | - name: sonar |
|  | containerPort: 9000  ... |
|  | ======================================= |
|  | kubectl apply -f sonar.yaml |
|  |  |
|  |  |
|  | ============================================= |
|  | vi sonar-service.yaml |
|  | ------------------------------  --- |
|  | apiVersion: v1 |
|  | kind: Service |
|  | metadata: |
|  | name: sonarqube-service |
|  | spec: |
|  | type: NodePort |
|  | ports: |
|  | - port: 9000 |
|  | targetPort: 9000 |
|  | nodePort: |
|  | selector: |
|  | app: sonarqube |
|  | type: LoadBalancer  ... |
|  | ============================= |
|  | kubectl apply -f sonar-service.yaml |
|  |  |
|  |  |
|  | kubectl get svc |
|  |  |
| ==================   |  | | --- | |  | |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | |  |  | | ========================================================  **Namespace**  To create Namespaces  ===========  vim namespace.yml  ---  apiVersion: v1  kind: Namespace  metadata:  name: test-ns  ...  kubectl apply -f namespace.yaml  To see the list of namespace  ================================  kubectl get namespace  Create a pod on that namespace  ===================================  vim pod-namespace.yml  ---  apiVersion: v1  kind: Pod  metadata:  name: jdk-pod  namespace: test-ns  labels:  author: company  spec:  containers:  - name: java  image: openjdk:12  ...  --------------------------  To create of the pod  Kubectl apply -f pod-namespace.yml  -----------------------------------------  To see list of pods in a namespace  ======================================  kubectl get pods -n test-ns  To delete a namespace  ===========================  kubectl delete namespace test-ns  =================================  **Replicationset**  Create a replicaset file to start 4 tomcat replicas and then perform scaling  vim replica-set.yml  ---  apiVersion: apps/v1  kind: ReplicaSet  metadata:  name: tomcat-rs  labels:  type: webserver  author: company  spec:  replicas: 4  selector:  matchLabels:  type: webserver    template:  metadata:  name: tomcat-pod  labels:  type: webserver  spec:  containers:  - name: mywebserver  image: tomcat  ports:  - containerPort: 8080  hostPort: 9090  …  ==========================================  To create the pods from the above file  kubectl create -f replica-set.yml  kubectl get pods  =============================================== |