**Jenkins on Kubernetes Assignment**

Objective: Deploy Jenkins as a master pod on Kubernetes, configure dynamic agent pods to run pipeline jobs, and verify execution.

## Jenkins Master Deployment

Deployment YAML

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: jenkins
  labels:
    app: jenkins
spec:
  replicas: 1
  selector:
    matchLabels:
      app: jenkins
  template:
    metadata:
      labels:
        app: jenkins
    spec:
      serviceAccountName: jenkins
      containers:
        - name: jenkins
          image: jenkins/jenkins:lts-jdk17
          ports:
            - containerPort: 8080
            - containerPort: 50000
          volumeMounts:
            - name: jenkins-home
              mountPath: /var/jenkins_home
      volumes:
        - name: jenkins-home
          emptyDir: {}
```

## NodePort Service YAML

Exposing port 30000 for web UI and 50000 for agent pods.

```
apiVersion: v1
kind: Service
metadata:
  name: jenkins
  labels:
    app: jenkins
```

```
spec:
  type: NodePort
  selector:
    app: jenkins
  ports:
    - name: http
      port: 8080
      targetPort: 8080
      nodePort: 30000
    - name: jnlp
      port: 50000
      targetPort: 50000
```

```
kubectl get pods -o wide
```

```
kubectl get svc jenkins
```

## ServiceAccount & RBAC

Created `ServiceAccount: jenkins` and `ClusterRoleBinding: jenkins` for permissions.

**jenkins-rbac.yaml**

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jenkins
  namespace: default
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: jenkins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
  - kind: ServiceAccount
    name: jenkins
    namespace: default
```

**jenkins-crb.yaml**

apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: jenkins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole

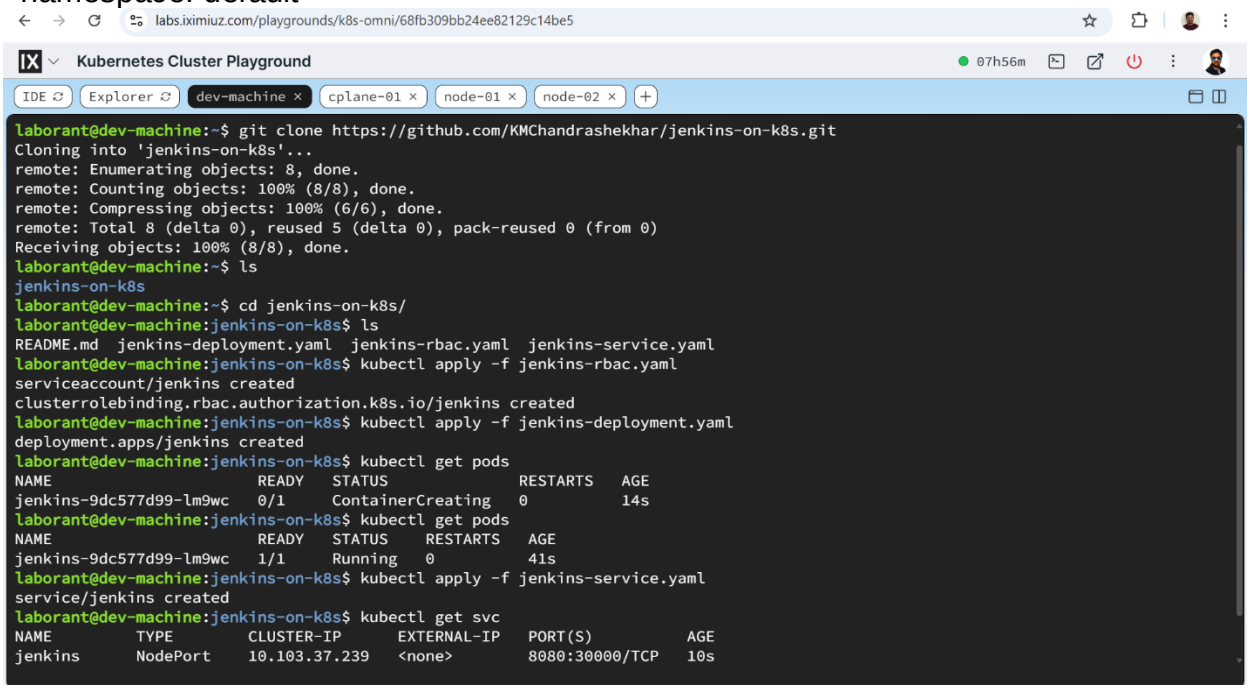
```
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: jenkins
  namespace: default
```

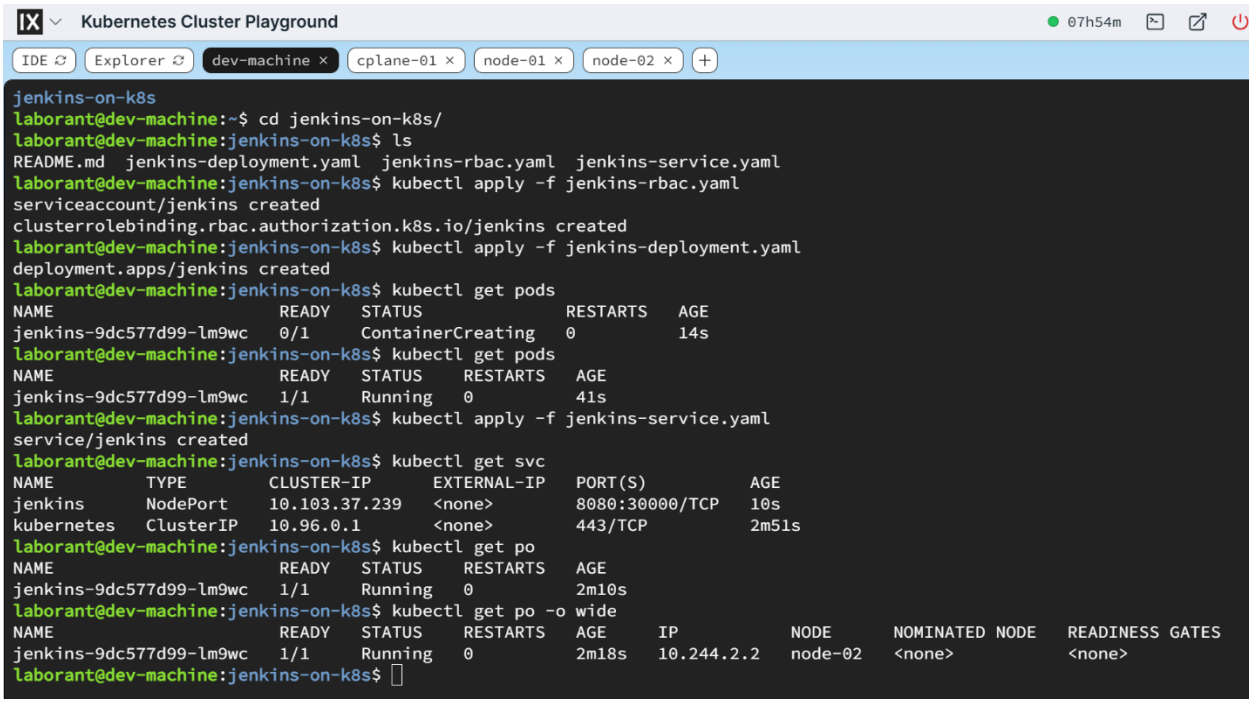
**jenkins-service.yaml**

```
apiVersion: v1
kind: Service
metadata:
  name: jenkins
spec:
  type: NodePort
  ports:
    - port: 8080
      targetPort: 8080
      nodePort: 30000
  selector:
    app: Jenkins
```

**jenkins-sa.yaml**

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jenkins
  namespace: default
```






## Expose HTTP(S) Ports

Access HTTP and HTTPS services running in the playground from your browser or local terminal. Handy for exploring web apps (e.g. Prometheus UI, Kubernetes Dashboard, etc.) or exposing HTTP APIs.

Machine: node-02

Port: 30000

HTTPS

EXPOSE

Advanced...

Only ports opened on the VM's primary network interface can be exposed. If you need to expose a port opened on a different interface (e.g., `localhost`), first forward it using `socat` or a similar tool.

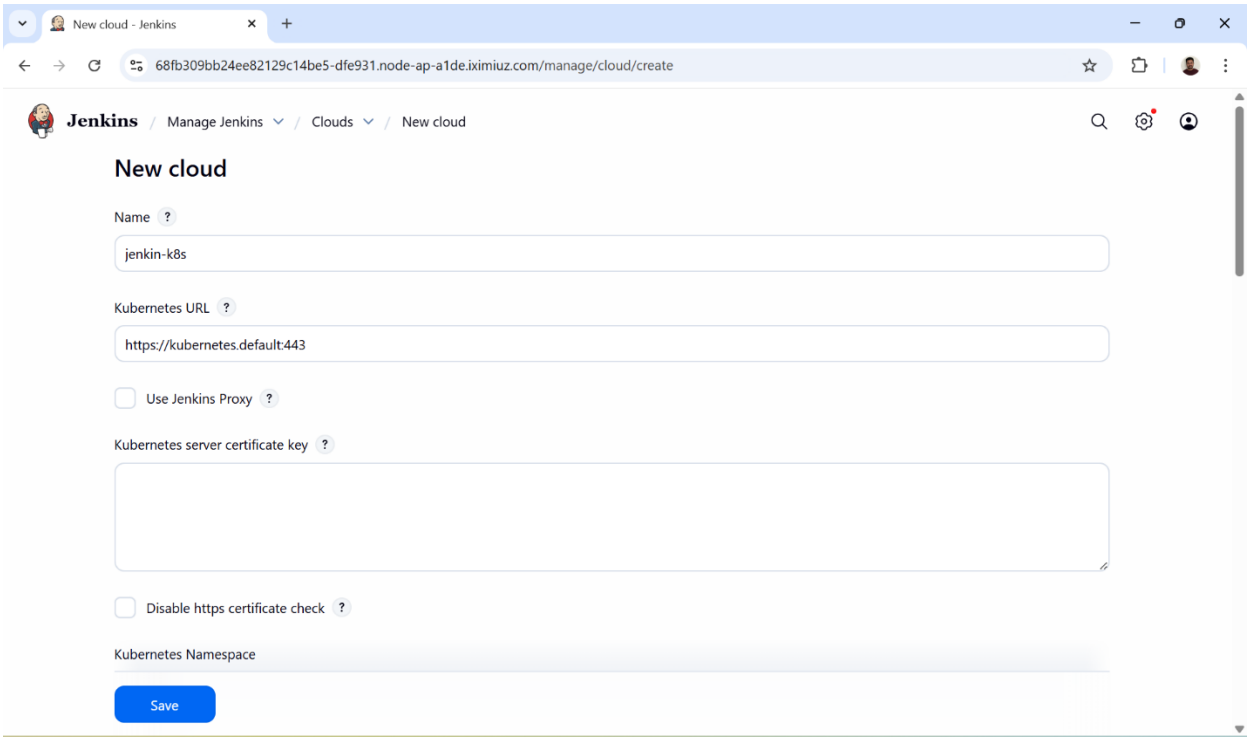
You can verify the target service is exposable using the following command from inside the playground:

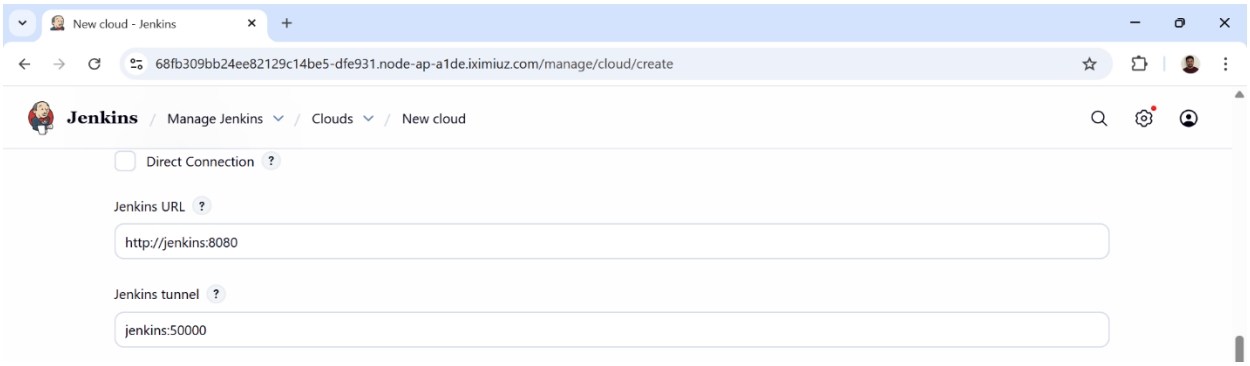
```
curl http://node-02:30000
```

| Port | URL | Host | Path | HTTPS | Access | Actions |
|---|---|---|---|---|---|---|
| node-02:30000 | 68...fe931.node-ap-a1de.iximiuz.com | - | - | No | private | |

# Configuring Jenkins Kubernetes Cloud

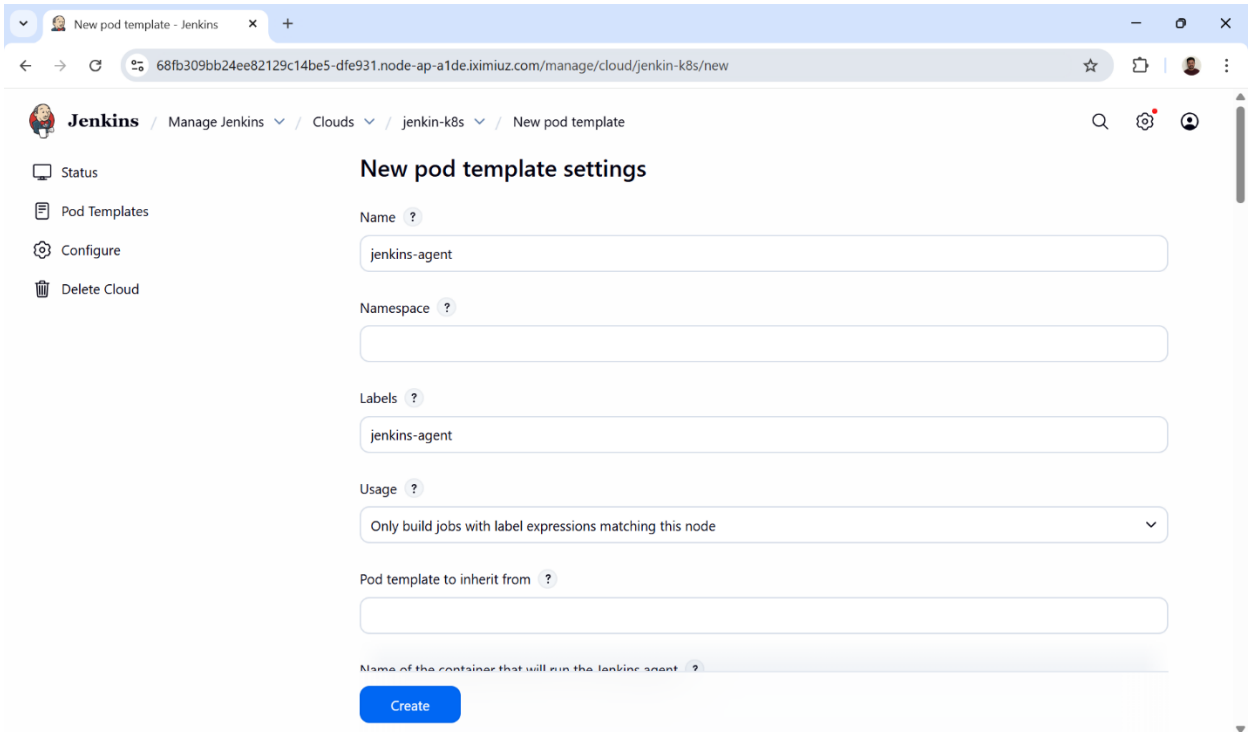
Steps: 1. Manage Jenkins → Configure System → Clouds → Add new cloud → Kubernetes 2. Fields: - Kubernetes URL: `https://kubernetes.default:443` - Jenkins URL: `http://jenkins:8080` - Jenkins Tunnel: `jenkins:50000` - Namespace: `default` - Credentials: Kubernetes Service Account - Skip TLS Verify:

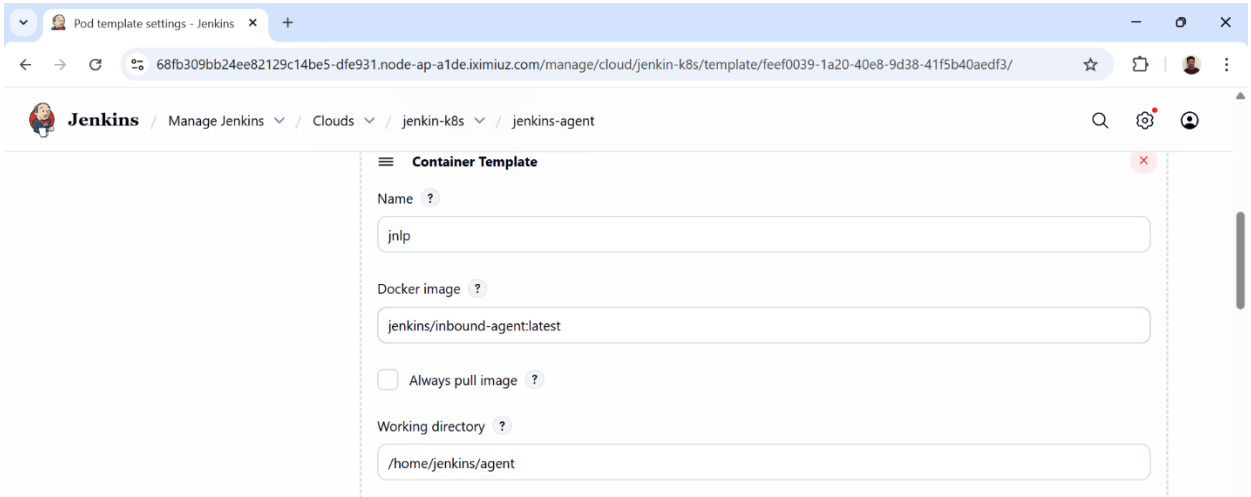


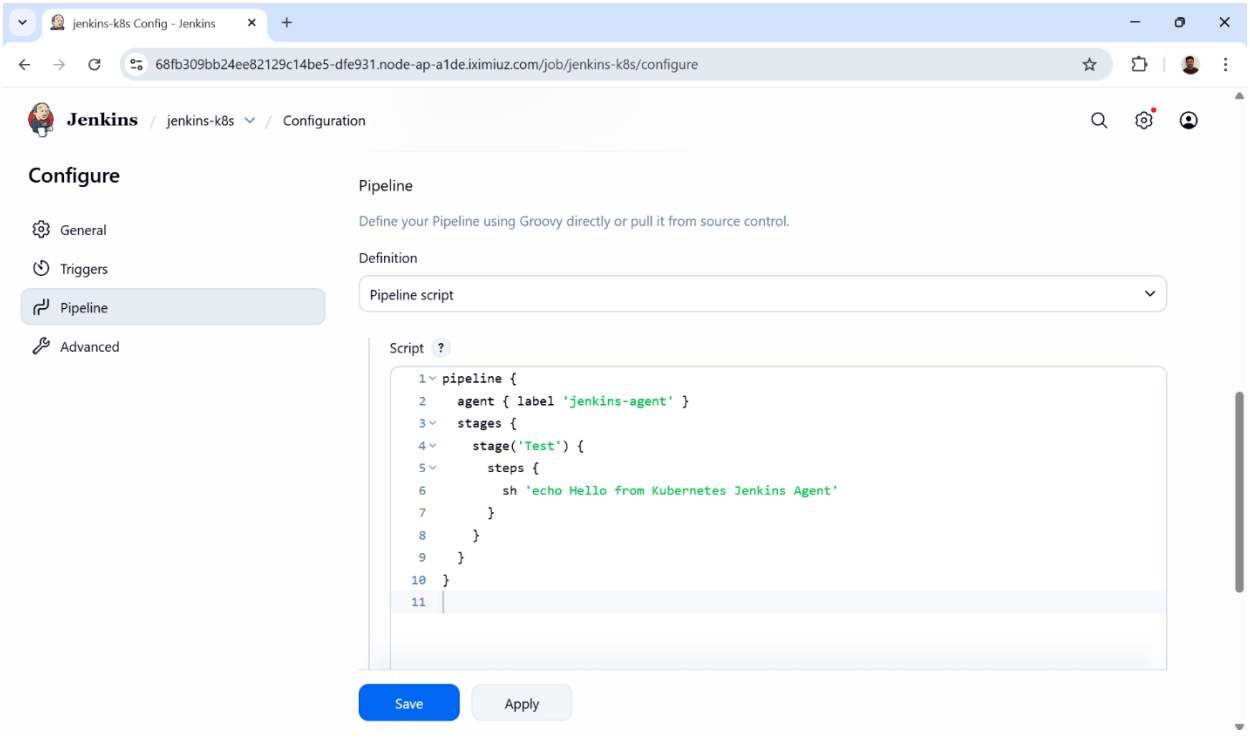
# Adding Pod Template for Agents

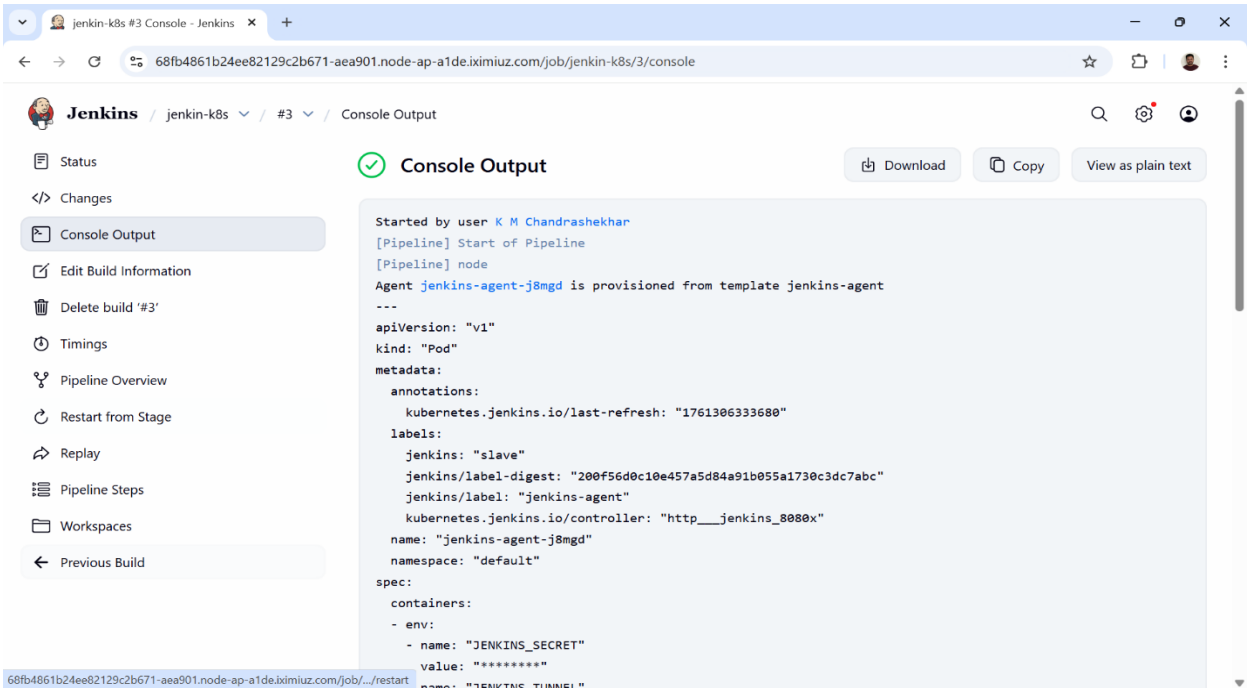
- Label: `jenkins-agent`
- Container Name: `jnlp`
- Image: `jenkins/inbound-agent:lts-jdk17`
- Command / Args: empty
- Working Directory: `/home/jenkins/agent`

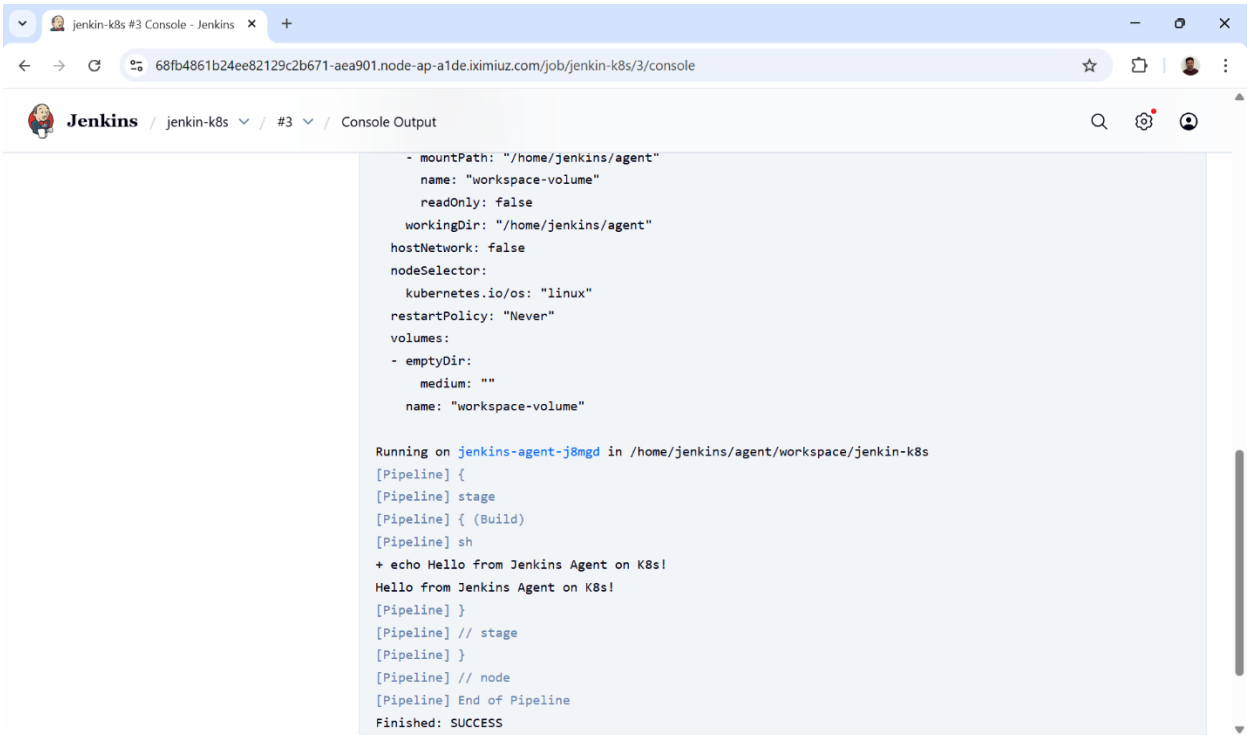


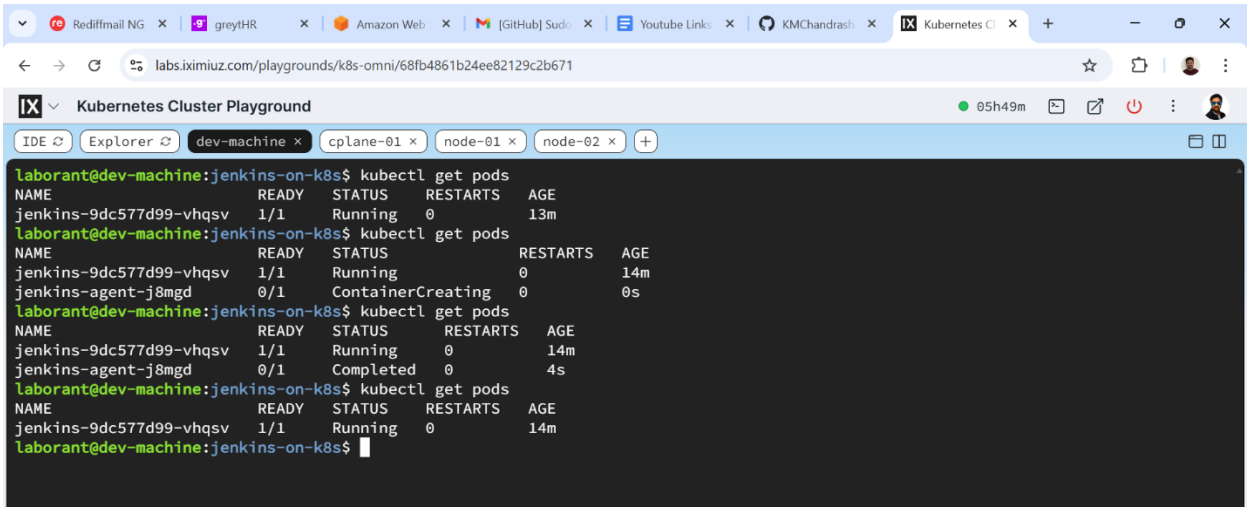



# Pipeline Job Execution






- Jenkins master running on Kubernetes
- Agent pods spin up dynamically to execute jobs
- Pipeline successfully executed using an agent pod
- NodePort service allowed web UI access