

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
```

```
kubect1 get pods -o wide
```

```
kubect1 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
```

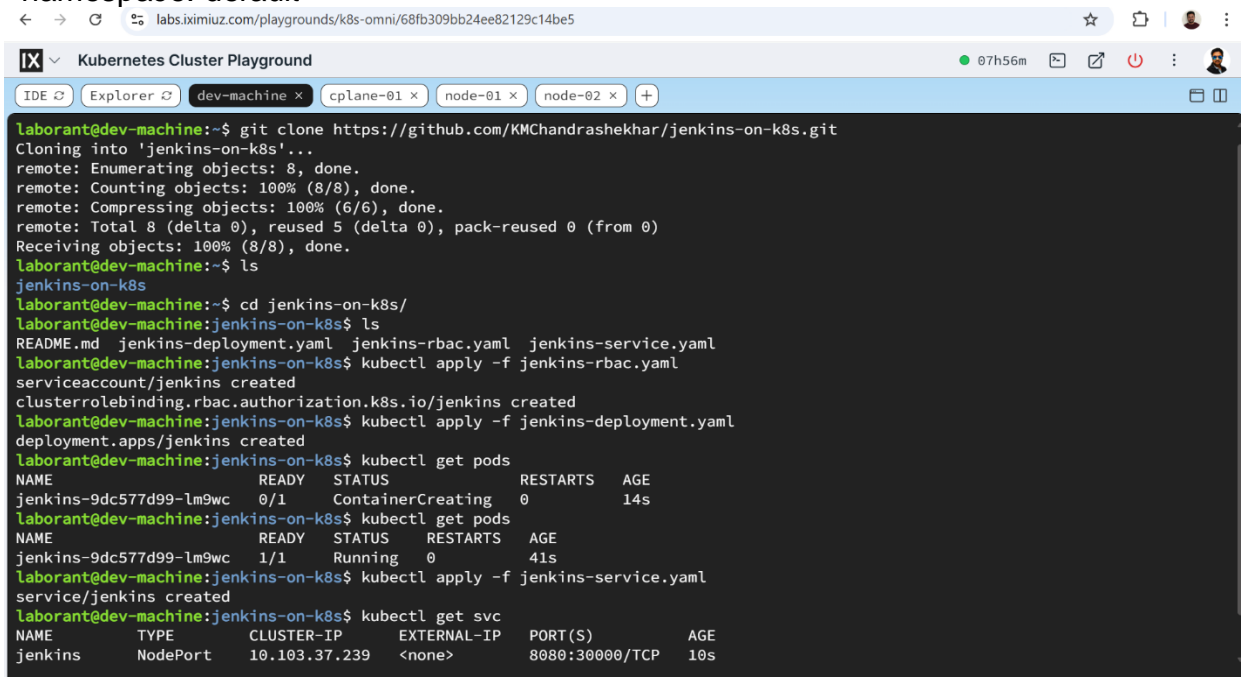
jenkins-crb.yaml

```
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
```



The screenshot shows a terminal window titled "Kubernetes Cluster Playground" with a tab for "dev-machine". The terminal output shows the following commands and results:

```
Laborant@dev-machine:~$ git clone https://github.com/KMChandrashekhhar/jenkins-on-k8s.git
Cloning into 'jenkins-on-k8s'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 8 (delta 0), reused 5 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (8/8), done.
Laborant@dev-machine:~$ ls
jenkins-on-k8s
Laborant@dev-machine:~$ cd jenkins-on-k8s/
Laborant@dev-machine:jenkins-on-k8s$ ls
README.md jenkins-deployment.yaml jenkins-rbac.yaml jenkins-service.yaml
Laborant@dev-machine:jenkins-on-k8s$ kubectl apply -f jenkins-rbac.yaml
serviceaccount/jenkins created
clusterrolebinding.rbac.authorization.k8s.io/jenkins created
Laborant@dev-machine:jenkins-on-k8s$ kubectl apply -f jenkins-deployment.yaml
deployment.apps/jenkins created
Laborant@dev-machine:jenkins-on-k8s$ kubectl get pods
NAME READY STATUS RESTARTS AGE
jenkins-9dc577d99-lm9wc 0/1 ContainerCreating 0 14s
Laborant@dev-machine:jenkins-on-k8s$ kubectl get pods
NAME READY STATUS RESTARTS AGE
jenkins-9dc577d99-lm9wc 1/1 Running 0 41s
Laborant@dev-machine:jenkins-on-k8s$ kubectl apply -f jenkins-service.yaml
service/jenkins created
Laborant@dev-machine:jenkins-on-k8s$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
jenkins NodePort 10.103.37.239 <none> 8080:30000/TCP 10s
```

```
IX Kubernetes Cluster Playground 07h54m
IDE Explorer dev-machine x cplane-01 x node-01 x node-02 x +
jenkins-on-k8s
laborant@dev-machine:~$ cd jenkins-on-k8s/
laborant@dev-machine:jenkins-on-k8s$ ls
README.md jenkins-deployment.yaml jenkins-rbac.yaml jenkins-service.yaml
laborant@dev-machine:jenkins-on-k8s$ kubectl apply -f jenkins-rbac.yaml
serviceaccount/jenkins created
clusterrolebinding.rbac.authorization.k8s.io/jenkins created
laborant@dev-machine:jenkins-on-k8s$ kubectl apply -f jenkins-deployment.yaml
deployment.apps/jenkins created
laborant@dev-machine:jenkins-on-k8s$ kubectl get pods
NAME READY STATUS RESTARTS AGE
jenkins-9dc577d99-lm9wc 0/1 ContainerCreating 0 14s
laborant@dev-machine:jenkins-on-k8s$ kubectl get pods
NAME READY STATUS RESTARTS AGE
jenkins-9dc577d99-lm9wc 1/1 Running 0 41s
laborant@dev-machine:jenkins-on-k8s$ kubectl apply -f jenkins-service.yaml
service/jenkins created
laborant@dev-machine:jenkins-on-k8s$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
jenkins NodePort 10.103.37.239 <none> 8080:30000/TCP 10s
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 2m51s
laborant@dev-machine:jenkins-on-k8s$ kubectl get po
NAME READY STATUS RESTARTS AGE
jenkins-9dc577d99-lm9wc 1/1 Running 0 2m10s
laborant@dev-machine:jenkins-on-k8s$ kubectl get po -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
jenkins-9dc577d99-lm9wc 1/1 Running 0 2m18s 10.244.2.2 node-02 <none> <none>
laborant@dev-machine:jenkins-on-k8s$
```

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.

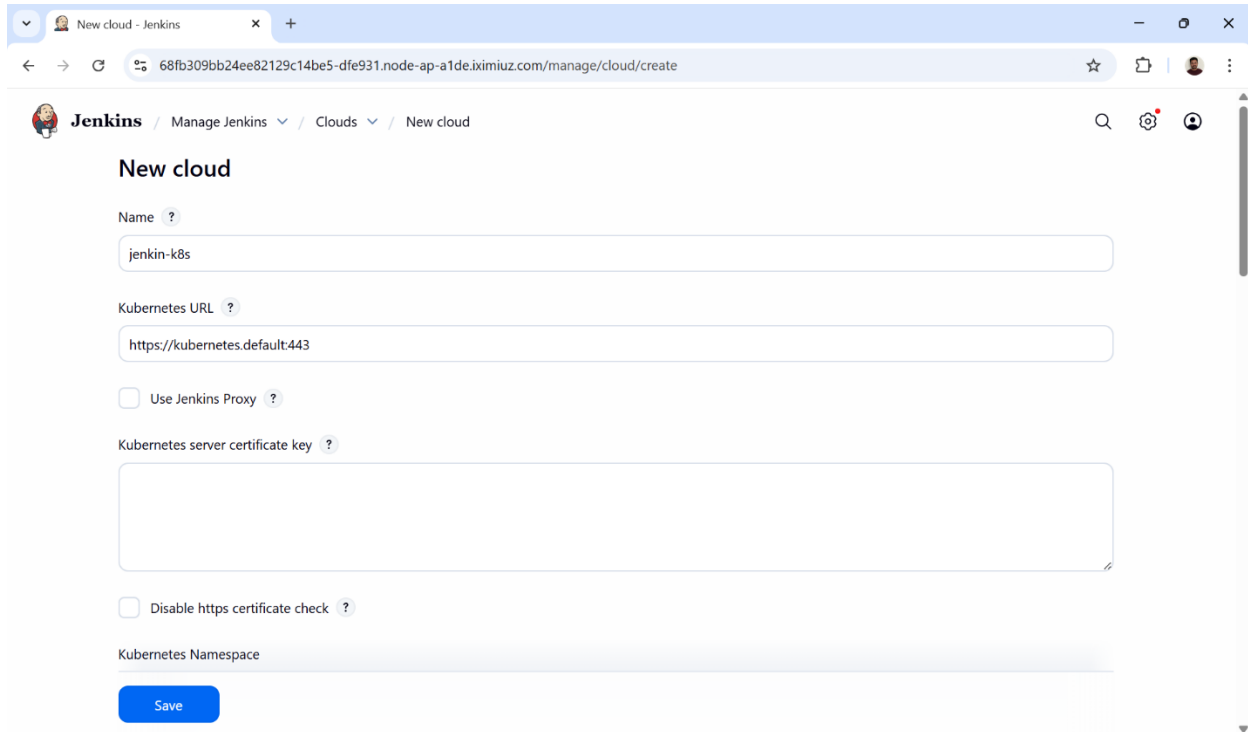
You can verify the target service is exposable using the following command from inside the playground:

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

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

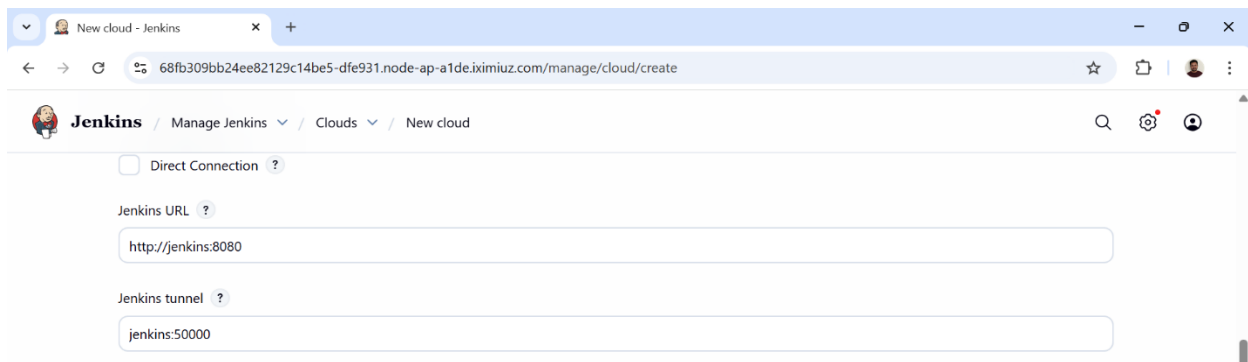
Configuring Jenkins Kubernetes Cloud

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:



The screenshot shows the Jenkins 'New cloud' configuration page. The browser address bar displays the URL: `68fb309bb24ee82129c14be5-dfe931.node-ap-a1de.iximiuz.com/manage/cloud/create`. The page title is 'New cloud'. The configuration fields are as follows:

- Name**: `jenkin-k8s`
- Kubernetes URL**: `https://kubernetes.default:443`
- ☐ **Use Jenkins Proxy**
- Kubernetes server certificate key**: (Empty text area)
- ☐ **Disable https certificate check**
- Kubernetes Namespace**: (Empty text field)
- Save** button



The screenshot shows the Jenkins 'New cloud' configuration page for a Direct Connection cloud. The browser address bar displays the URL: `68fb309bb24ee82129c14be5-dfe931.node-ap-a1de.iximiuz.com/manage/cloud/create`. The page title is 'New cloud'. The configuration fields are as follows:

- ☐ **Direct Connection**
- Jenkins URL**: `http://jenkins:8080`
- Jenkins tunnel**: `jenkins:50000`

Adding Pod Template for Agents

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

The screenshot shows the 'New pod template settings' page in the Jenkins web interface. The breadcrumb navigation at the top reads: Jenkins / Manage Jenkins / Clouds / jenkins-k8s / New pod template. On the left sidebar, there are links for Status, Pod Templates, Configure, and Delete Cloud. The main form contains the following fields:

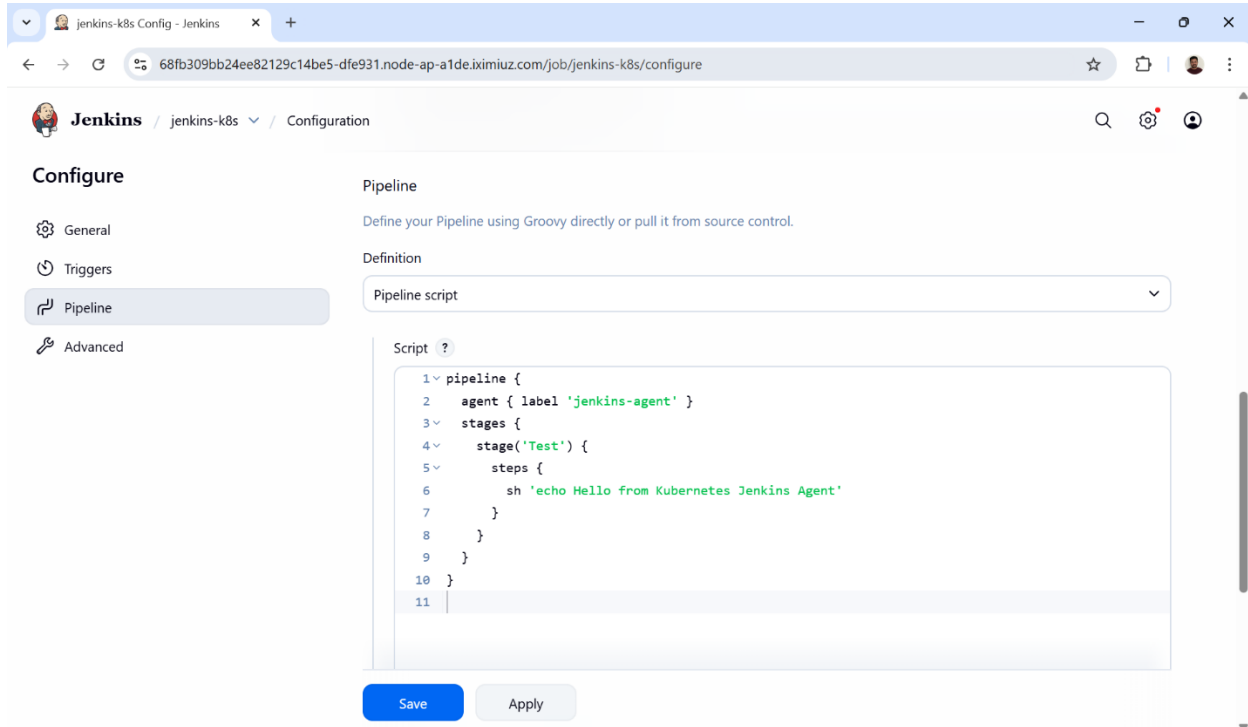
- Name**: jenkins-agent
- Namespace**: (empty)
- Labels**: jenkins-agent
- Usage**: Only build jobs with label expressions matching this node (dropdown menu)
- Pod template to inherit from**: (empty)
- Name of the container that will run the Jenkins agent**: (empty)

A blue 'Create' button is located at the bottom of the form.

The screenshot shows the 'Container Template' configuration page for the 'jenkins-agent' pod template. The breadcrumb navigation at the top reads: Jenkins / Manage Jenkins / Clouds / jenkins-k8s / jenkins-agent. The main form contains the following fields:

- Name**: jnlp
- Docker image**: jenkins/inbound-agent:latest
- Always pull image**: (checkbox, unchecked)
- Working directory**: /home/jenkins/agent

Pipeline Job Execution



The screenshot shows the Jenkins configuration page for a pipeline job named 'jenkins-k8s'. The left sidebar contains navigation links: General, Triggers, Pipeline (selected), and Advanced. The main area is titled 'Configure' and 'Pipeline'. It instructs the user to 'Define your Pipeline using Groovy directly or pull it from source control.' Below this, the 'Definition' section has a dropdown menu set to 'Pipeline script'. A 'Script' section contains a Groovy pipeline script. At the bottom, there are 'Save' and 'Apply' buttons.

Configure

- General
- Triggers
- Pipeline**
- Advanced

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

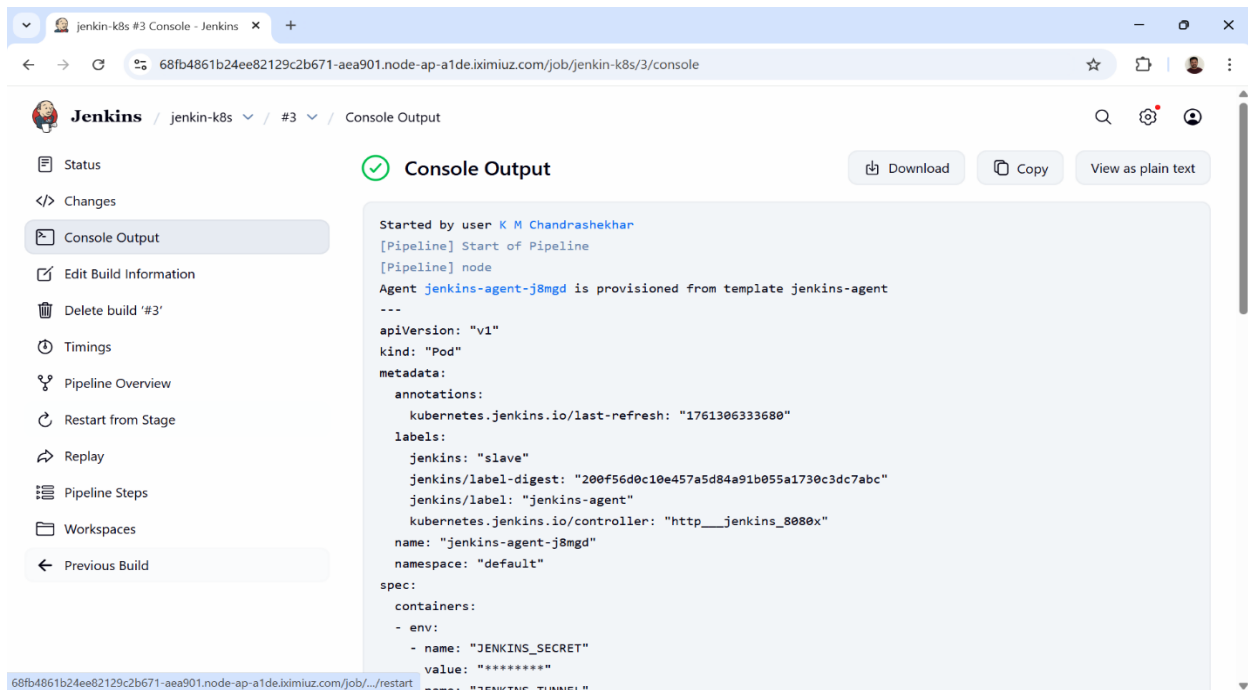
Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent { label 'jenkins-agent' }
3   stages {
4     stage('Test') {
5       steps {
6         sh 'echo Hello from Kubernetes Jenkins Agent'
7       }
8     }
9   }
10 }
11
```

Save Apply



The screenshot shows the Jenkins console output for the 'jenkins-k8s' job, build #3. The left sidebar contains navigation links: Status, Changes, Console Output (selected), Edit Build Information, Delete build '#3', Timings, Pipeline Overview, Restart from Stage, Replay, Pipeline Steps, Workspaces, and Previous Build. The main area is titled 'Console Output' and shows the output of the pipeline. The output includes the start of the pipeline, the agent 'jenkins-agent-j8mgd' being provisioned from the template 'jenkins-agent', and the pod's metadata and spec.

jenkins-k8s / #3 / Console Output

Console Output

Download Copy View as plain text

```
Started by user K M Chandrashekhara
[Pipeline] Start of Pipeline
[Pipeline] node
Agent jenkins-agent-j8mgd is provisioned from template jenkins-agent
---
apiVersion: "v1"
kind: "Pod"
metadata:
  annotations:
    kubernetes.jenkins.io/last-refresh: "1761306333680"
  labels:
    jenkins: "slave"
    jenkins/label-digest: "200f56d0c10e457a5d84a91b055a1730c3dc7abc"
    jenkins/label: "jenkins-agent"
    kubernetes.jenkins.io/controller: "http___jenkins_8080x"
  name: "jenkins-agent-j8mgd"
  namespace: "default"
spec:
  containers:
  - env:
    - name: "JENKINS_SECRET"
      value: "*****"
    name: "jenkins-agent-j8mgd"
```

The screenshot shows the Jenkins web interface for a job named 'jenkin-k8s'. The console output displays the configuration for a pod and the execution of a pipeline. The pod configuration includes a workspace volume mounted at '/home/jenkins/agent'. The pipeline consists of a single stage named 'Build' with a shell step that echoes a message. The output shows the message 'Hello from Jenkins Agent on K8s!' and a successful completion status.

```
- mountPath: "/home/jenkins/agent"
  name: "workspace-volume"
  readOnly: false
  workingDir: "/home/jenkins/agent"
hostNetwork: false
nodeSelector:
  kubernetes.io/os: "linux"
restartPolicy: "Never"
volumes:
- emptyDir:
    medium: ""
  name: "workspace-volume"

Running on jenkins-agent-j8mgd in /home/jenkins/agent/workspace/jenkin-k8s
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] sh
+ echo Hello from Jenkins Agent on K8s!
Hello from Jenkins Agent on K8s!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

The screenshot shows a terminal window with a Kubernetes Cluster Playground interface. The terminal displays the output of the 'kubectl get pods' command, showing the status of various pods. The pods include 'jenkins-9dc577d99-vhqsv' (Running), 'jenkins-agent-j8mgd' (ContainerCreating), and 'jenkins-agent-j8mgd' (Completed). The terminal also shows the output of the 'kubectl get pods' command again, showing the status of the pods after the job has completed.

```
laborant@dev-machine:jenkins-on-k8s$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
jenkins-9dc577d99-vhqsv  1/1     Running   0           13m
laborant@dev-machine:jenkins-on-k8s$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
jenkins-9dc577d99-vhqsv  1/1     Running   0           14m
jenkins-agent-j8mgd    0/1     ContainerCreating   0           0s
laborant@dev-machine:jenkins-on-k8s$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
jenkins-9dc577d99-vhqsv  1/1     Running   0           14m
jenkins-agent-j8mgd    0/1     Completed   0           4s
laborant@dev-machine:jenkins-on-k8s$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
jenkins-9dc577d99-vhqsv  1/1     Running   0           14m
laborant@dev-machine:jenkins-on-k8s$
```

- 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