SETUP CONTINUES DEPLOYMENT

CREATE & PUBLISH DOCKER IMAGES

✓ Create Docker file Instructions.

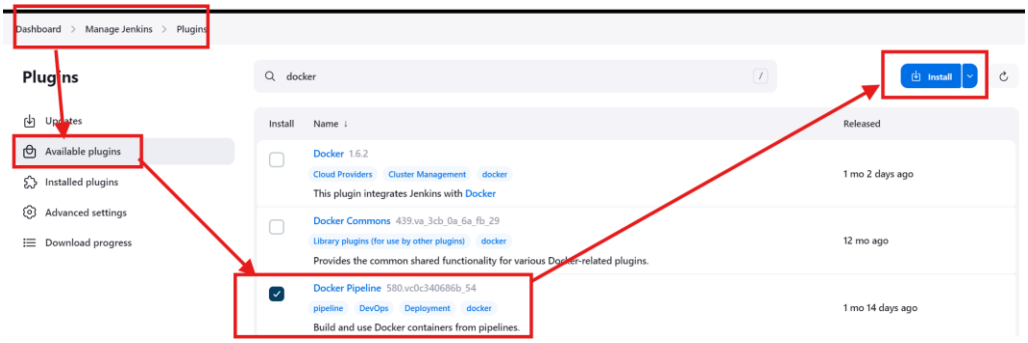
- Create a new file **Dockerfile** in local repository.
- Add the following instructions

```
FROM adoptopenjdk/openjdk11

EXPOSE 8080

ENV APP_HOME /usr/src/app

COPY target/*.jar $APP_HOME/app.jar

WORKDIR $APP_HOME

CMD ["java", "-jar", "app.jar"]
```

- And commit to the repository and push to GITHUB.

✓ Install Docker Plugin for Jenkins

- In Jenkins Dashboard head over to **Manage Jenkins > Plugins > Available Plugins.**
- Search for **Docker Pipeline > Install > Restart** (*if needed).

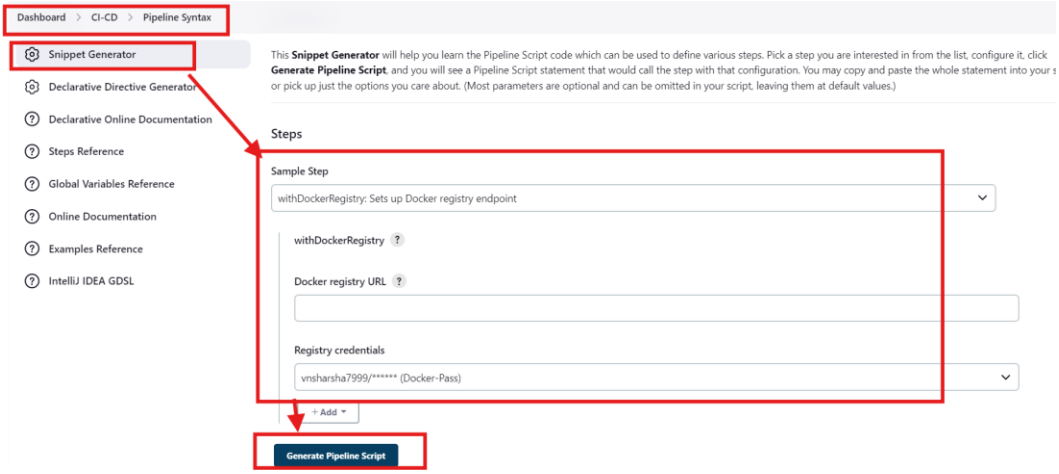


✓ Setup Docker Hub credentials

- In Jenkins Dashboard head over to **Manage Jenkins > Credentials > Global Credentials > Create Credentials.**
- Select type as **Username and Password > Enter the username and password > Give unique ID > Add.**

✓ Generate pipeline script.

- In Jenkins Dashboard head over to **New Item > Select Pipeline > Give a name > Create**
- Scroll down to the bottom of the page **Select Pipeline Syntax > Declarative Snippet Generator.**
- Select the **withDockerRepo step > Give Repo URL > Select docker credential from dropdown > Generate pipeline syntax.**



➢ Copy the step, and add it to the pipeline.

```
stage ("CREATE AND PUSH DOCKER IMAGES"){
            steps{
                withDockerRegistry(credentialsId: 'Docker-Pass') {
                  sh "docker build -t <Docker-registry-name>/App ."
                  sh "docker push <Docker-registry-name> /App"
             }
               }
        }
```

SETUP KUBERNETES DEPLOYMENT

✓ Create Kubernetes Manifest.

- Create a new file **deployment-service.yaml** in local repository.
- Add the following contents.

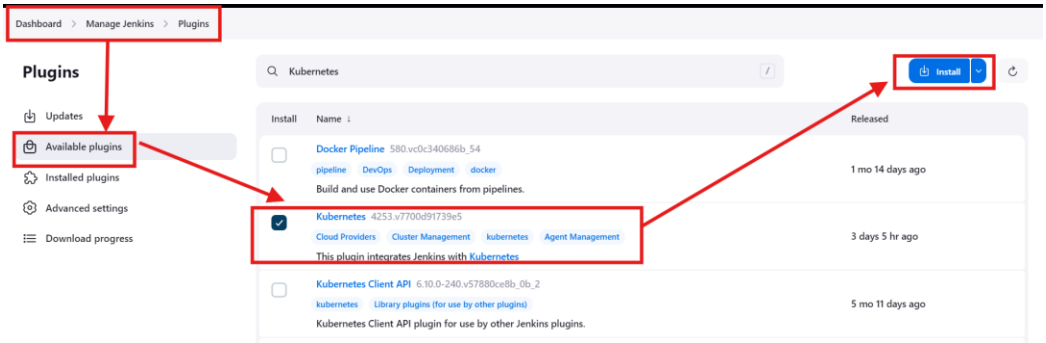
```
apiVersion: apps/v1
kind: Deployment # Kubernetes resource kind we are creating
metadata:
  name: boardgame-deployment
spec:
  selector:
    matchLabels:
      app: boardgame
  replicas: 2 # Number of replicas that will be created for this
deployment
  template:
    metadata:
      labels:
        app: boardgame
    spec:
      containers:
        - name: boardgame
          image: adijaiswal/boardgame:latest # Image that will be
used to containers in the cluster
          imagePullPolicy: Always
          ports:
            - containerPort: 8080 # The port that the container is
running on in the cluster
```

```yaml
---

apiVersion: v1 # Kubernetes API version
kind: Service # Kubernetes resource kind we are creating
metadata: # Metadata of the resource kind we are creating
  name: boardgame-ssvc
spec:
  selector:
    app: boardgame
  ports:
    - protocol: "TCP"
      port: 8080
      targetPort: 8080
type: LoadBalancer # type of the service.
```

- Add and commit to GITHUB.

✓ Install Kubernetes Plugin

- In Jenkins Dashboard head over to **Manage Jenkins > Plugins > Available Plugins.**
- Search for the following pulgins
  - **Kubernetes**
  - **Kubernetes CLI**
  - **Kubernetes Credential**
- **Install > Restart** (*if needed).



✓ Setup Cluster credentials

- SSH to the minikube server, and use the following command to extract the cluster credentials file.

Cat .kube/config

```
ubuntu@ip-172-31-91-57:~$ cat .kube/config
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /home/ubuntu/.minikube/ca.crt
    extensions:
    - extension:
        last-update: Thu, 11 Jul 2024 11:58:42 UTC
        provider: minikube.sigs.k8s.io
        version: v1.33.1
      name: cluster_info
    server: https://192.168.49.2:8443
  name: minikube
contexts:
- context:
    cluster: minikube
    extensions:
    - extension:
        last-update: Thu, 11 Jul 2024 11:58:42 UTC
        provider: minikube.sigs.k8s.io
        version: v1.33.1
      name: context_info
    namespace: default
    user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: /home/ubuntu/.minikube/profiles/minikube/client.crt
    client-key: /home/ubuntu/.minikube/profiles/minikube/client.key
```

- Get the CA file details and get the credentials.

```
Last login: Thu Jul 11 12:11:12 2024 from 49.205.84.211
ubuntu@ip-172-31-91-57:~$ cat /home/ubuntu/.minikube/ca.crt
-----BEGIN CERTIFICATE-----
MIIDBjCCAe6gAwIBAgIBATANBgkqhkiG9w0BAQsFADAVMRMwEQYDVQQDEwptaW5p
a3ViZUNBMB4XDTI0MDcxMDExNTgyN1oXDTM0MDcwOTExNTgyN1owFTETMBEGA1UE
AxMKbWluaWt1YmVDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMOc
qcNWC/YqLEjfYDHXwmUfj1L1B0tIhPj4PHInr5E8eCx2TOD6w/JXFPbBYI0cmL9v
6x5xTxSq71BYHil2oZLHnA+qajcb3VxHvWKjRhOUzf6E7X8EE0zSpr0Aiml5UfeC
yJQPcZQUZgcAagTAy6iA3AokyK5y6sidQfLP2CQEKXWKPAcMduPxdUBARMHnpouc
PS8gtPZEI7GYlYD5PkCqhrDqdmFSMFahQnIkDZewMx840dGWbr14o+DZd7BGe1VN
ockgOgDMfn5T2Wtgzl6gPiDuTuSXSeLHb7RYygIR3jKZ7UcV6fKARVQPJvoR4cYn
Ee1l+EBzwFDgL6faftMCAwEAAaNhMF8wDgYDVR0PAQH/BAQDAgKkMB0GA1UdJQQW
MBQGCCsGAQUFBwMCBggrBgEFBQcDATAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQW
BBTOV52DZD+DnbhObM8bREIm2Kyf7TANBgkqhkiG9w0BAQsFAAOCAQEATytWhlTD
qCTOM1Ak+D0JQ7HOVzolzQI+XyM+AT8gG4j4P9mnmbjiid7yxYQM5LK5KQ7vmjmL
3fKdUxeVrRzwZXbjvvnGDuG4mJ+aq7SZ0ib29wcR8Woe6qrPW99Un12oamHHoWx0
NWVX0ljb35wwLqrzZZDScQHBkNqV7yP66pDzUBNlPUh2onAm+v+LWgUpz16E5+yR
tPLX9ZFUTpZaIzqlBh8Xt+15PFtWReVkg/K82Eusu4vUaGdL8vvXhs8yBXoBI0To
hVmWZO5X5c/hOe/ScIHP85PsIuo1p+1hL9prCVqqYcWVhOpoFWira74BO5Tzfu3u
QG66HufpVMQmug==
-----END CERTIFICATE-----
```

- **Copy the cluster certificate contents**
- In Jenkins Dashboard head over to **Manage Jenkins > Credentials > Global Credentials > Create Credentials.**

- Select type as **Kubernetes Configuration > Paste the CA cert > Give unique ID > Save.**

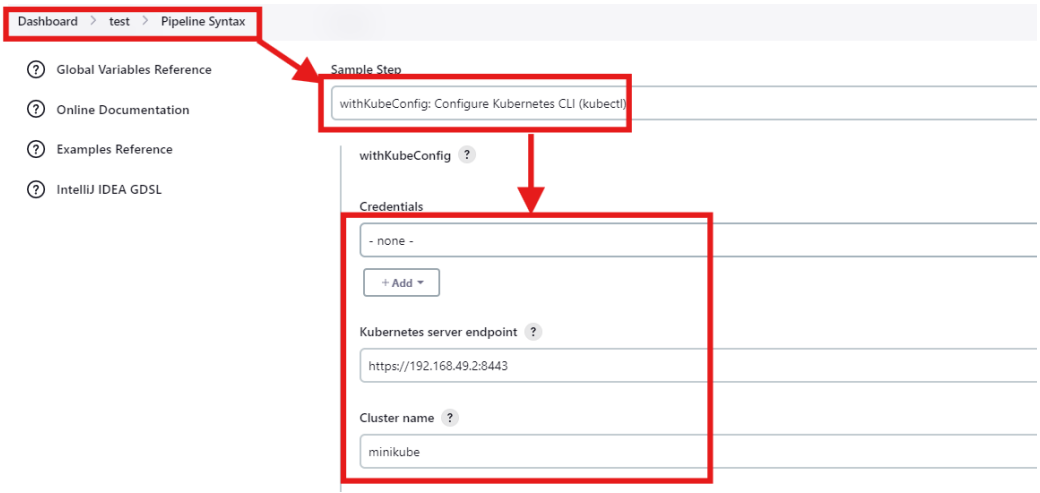
✓ Install Kubectl

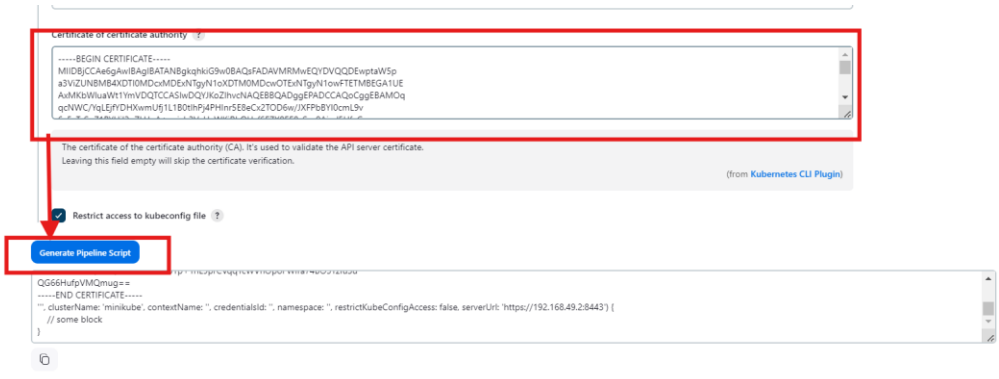
- SSH to the Jenkins server
- Run the following commands.

```
sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl
gnupg
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/deb/Release.key |
sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg
sudo apt-get update
sudo apt-get install -y kubectl
```

✓ Generate pipeline steps

- In Jenkins Dashboard head over to **New Item > Select Pipeline > Give a name > Create**
- Scroll down to the bottom of the page **Select Pipeline Syntax > Declarative Snippet Generator.**
- Select the **withKubeConfig step > Give the details > Generate pipeline Syntax.**





➢ Cope the step and add to the pipeline.

```
stage ("DEPLOY TO KUBERNETES"){
            steps{
                withKubeConfig(clusterName: 'minikube', contextName: '', credentialsId: '', namespace: '', restrictKubeConfigAccess: false, serverUrl: 'https://192.168.49.2:8443') {
                    sh "kubectl apply -f deployment-service.yaml"
                }
            }
        }
```