

EXAM

1. **Github repository:** <https://github.com/Abhiramikannan/spring-boot-hello-world.git>
2. **Dockerhub image:** [abhiramikannan/javaimage general | Docker Hub](#)
3. **Yaml files used:** deployment.yaml, service.yaml
4. **Deployment.yaml:**

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world-deployment
  labels:
    app: java-application
spec:
  replicas: 3
  selector:
    matchLabels:
      app: java-application
  template:
    metadata:
      labels:
        app: java-application
    spec:
      containers:
        - name: java-application
          image: abhiramikannan/javaimage:latest
          ports:
            - containerPort: 8080
```

5. **Service.yaml:**

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: NodePort
  selector:
    app: java-application
  ports:
    - port: 8080
      # By default and for convenience, the `targetPort` is set to
      # the same value as the `port` field.
      targetPort: 8080
      # Optional field
      # By default and for convenience, the Kubernetes control plane
      # will allocate a port from a range (default: 30000-32767)
      nodePort: 30007
```

6. **Screenshots:**

- **Building image:** `docker build -t abhiramikannan/javaimage .`

K L L R C O D A

Kubernet1.32

This playground will always have the latest Kubeadm Kubernetes version a few weeks after release.

You have access to an empty Kubeadm Kubernetes cluster with two 2GB nodes. The controlplane node has taint removed to be

Editor Tab1

Initialising Kubernetes... done

controlplane:~\$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1-24.04.1
controlplane:~\$ git clone https://github.com/Abhiramkannan/spring-boot-hello-world.git
Cloning into 'spring-boot-hello-world'...
remote: Enumerating objects: 44, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 44 (delta 9), reused 8 (delta 8), pack-reused 25 (from 1)
Receiving objects: 100% (44/44), 5.97 KiB | 1.99 MiB/s, done.
Resolving deltas: 100% (9/9), done.
controlplane:~\$ cd spring-boot-hello-world/
controlplane:~/spring-boot-hello-world\$ ls
README.md pom.xml src
controlplane:~/spring-boot-hello-world\$ nano Dockerfile
controlplane:~/spring-boot-hello-world\$ docker build -t abhiramkannan/javaimage .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 90.11kB
Step 1/11 : FROM maven:3.8.8-eclipse-temurin-17 AS builder
3.8.8-eclipse-temurin-17: Pulling from library/maven
3dd181f9be59: Pull complete
0f838805bddf: Pull complete
dfab58ae8b86: Pull complete
c6cc1bdab538: Pull complete
cc590439cbf2: Pull complete
14f3e5e9a252: Pull complete

6 32°C

Mostly sunny

Search

ENG IN

14:34

03-04-2025

Kubernet1.32

This playground will always have the latest Kubeadm Kubernetes version a few weeks after release.

You have access to an empty Kubeadm Kubernetes cluster with two 2GB nodes. The controlplane node has taint removed to be

Editor Tab1

[INFO] Total time: 10.230 s
[INFO] Finished at: 2025-04-03T09:05:14Z
[INFO] -----
--> Removed intermediate container d9892e422d78
--> f8f081d6e591
Step 7/11 : FROM openjdk:17-jdk-slim
17-jdk-slim: Pulling from library/openjdk
1fe172e4850f: Pull complete
44d3aa8d0766: Pull complete
6ce99fdf16e8: Pull complete
Digest: sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3f62f9774
Status: Downloaded newer image for openjdk:17-jdk-slim
--> 37cb44321d04
Step 8/11 : WORKDIR /app
--> Running in 795231e4573d
--> Removed intermediate container 795231e4573d
--> 2b1db8f8cd4b
Step 9/11 : COPY --from=builder /app/target/*.jar app.jar
--> 2c3b11ce6ee5
Step 10/11 : EXPOSE 8080
--> Running in c75f19200438
--> Removed intermediate container c75f19200438
--> 4aabcd9796f2
Step 11/11 : ENTRYPOINT ["java", "-jar", "app.jar"]
--> Running in 199518646cc1
--> Removed intermediate container 199518646cc1
--> adce24b786eb
Successfully built adce24b786eb
Successfully tagged abhiramkannan/javaimage:latest
controlplane:~/spring-boot-hello-world\$

6 32°C

Mostly sunny

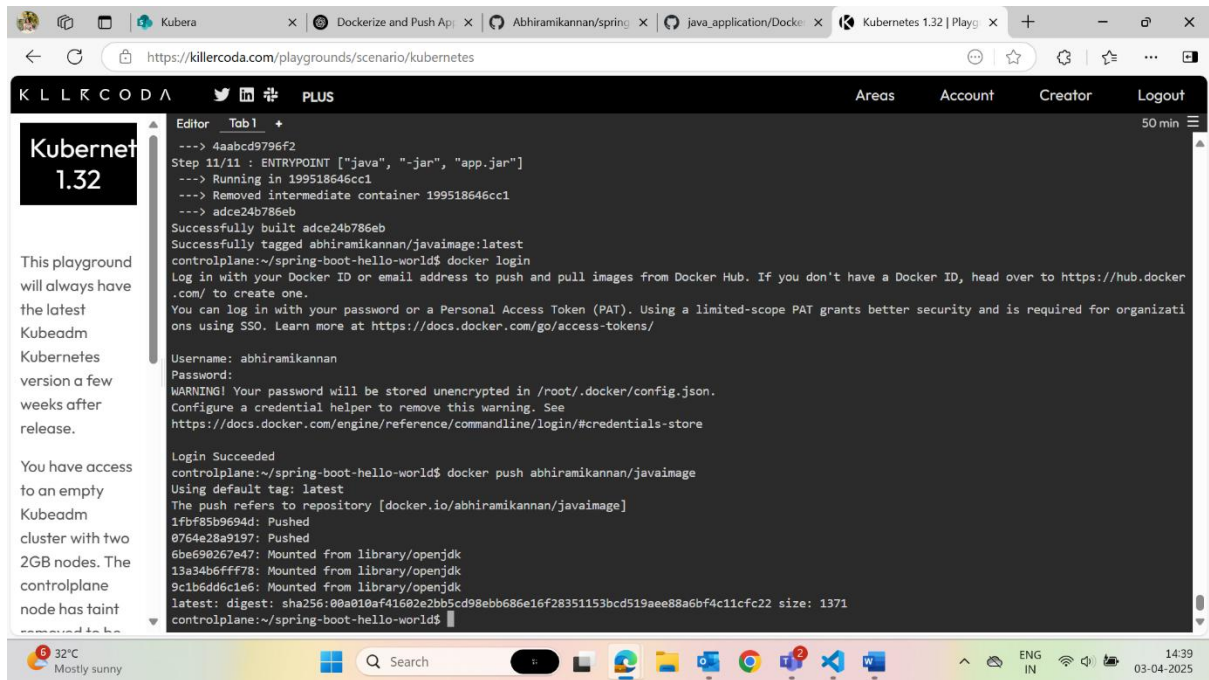
Search

ENG IN

14:35

03-04-2025

- Docker login and pushed image to docker hub



K L L R C O D A PLUS Areas Account Creator Logout

Kubernetes 1.32

This playground will always have the latest Kubeadm Kubernetes version a few weeks after release.

You have access to an empty Kubeadm cluster with two 2GB nodes. The controlplane node has taint removed to be

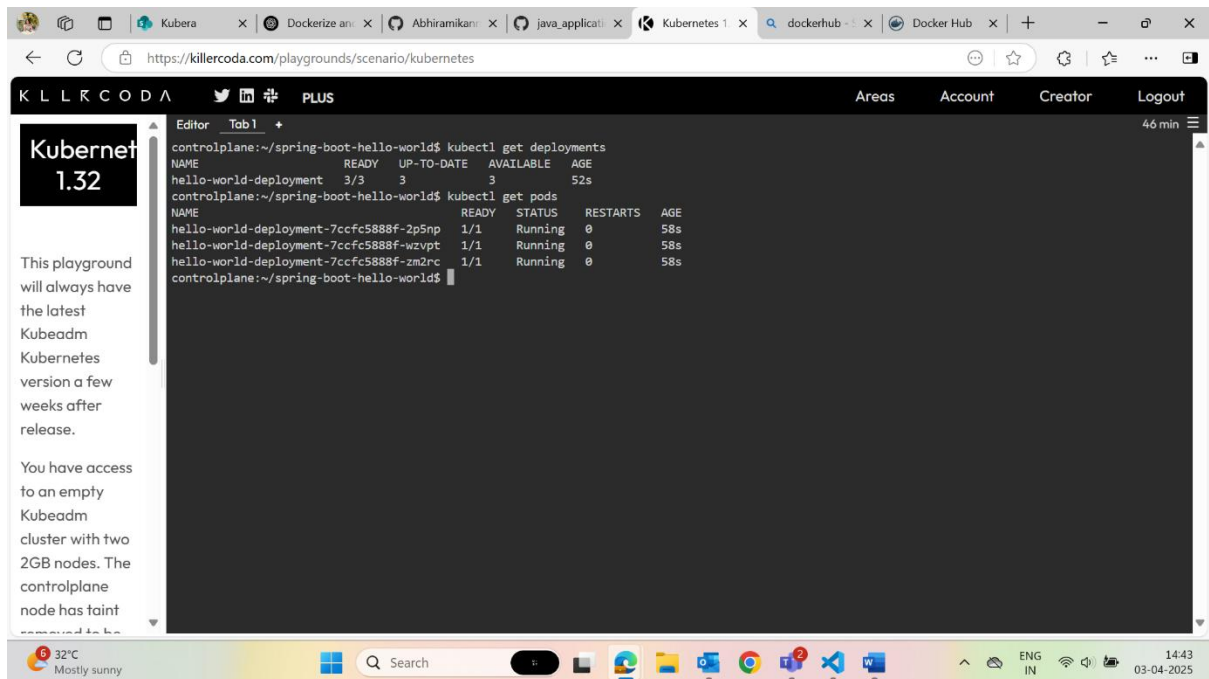
```
Editor Tab1 +
---- 4aabcd9796f2
Step 11/11 : ENTRYPOINT ["java", "-jar", "app.jar"]
---- Running in 199518646cc1
---- Removed intermediate container 199518646cc1
---- adce24b786eb
Successfully built adce24b786eb
Successfully tagged abhiramkannan/javaimage:latest
controlplane:~/spring-boot-hello-world$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username: abhiramkannan
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
controlplane:~/spring-boot-hello-world$ docker push abhiramkannan/javaimage
Using default tag: latest
The push refers to repository [docker.io/abhiramkannan/javaimage]
1fbf85b9694d: Pushed
0764a28a0197: Pushed
6be690267e47: Mounted from library/openjdk
13a34b6fff78: Mounted from library/openjdk
9c1b6dd6c166: Mounted from library/openjdk
latest: digest: sha256:00a010af41602e2bb5cd98ebb686e16f28351153bcd519aee88a6bf4c11cfc22 size: 1371
controlplane:~/spring-boot-hello-world$
```

32°C Mostly sunny Search 14:39 03-04-2025

- Created deployment



K L L R C O D A PLUS Areas Account Creator Logout

Kubernetes 1.32

This playground will always have the latest Kubeadm Kubernetes version a few weeks after release.

You have access to an empty Kubeadm cluster with two 2GB nodes. The controlplane node has taint removed to be

```
Editor Tab1 +
controlplane:~/spring-boot-hello-world$ kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
hello-world-deployment  3/3     3             3           52s
controlplane:~/spring-boot-hello-world$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
hello-world-deployment-7ccfc5888f-2p5np  1/1     Running    0           58s
hello-world-deployment-7ccfc5888f-wzvpt  1/1     Running    0           58s
hello-world-deployment-7ccfc5888f-zm2rc  1/1     Running    0           58s
controlplane:~/spring-boot-hello-world$
```

32°C Mostly sunny Search 14:43 03-04-2025

- Created service, scaled the deployment as replicas=3 and replicas=1

Kubernetes 1.32

This playground will always have the latest Kubeadm Kubernetes version a few weeks after release. You have access to an empty Kubeadm Kubernetes cluster with two 2GB nodes. The controlplane node has tainted...

```
controlplane:~/spring-boot-hello-world$ nano service.yaml
controlplane:~/spring-boot-hello-world$ kubectl apply -f service.yaml
service/my-service created
controlplane:~/spring-boot-hello-world$ kubectl get service
NAME         TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes   ClusterIP   10.96.0.1     <none>         443/TCP          11d
my-service    NodePort    10.107.241.147 <none>         8080:30007/TCP   15s
controlplane:~/spring-boot-hello-world$ curl 10.107.241.147:30007
Hello, World!
controlplane:~/spring-boot-hello-world$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
hello-world-deployment 3/3      3            3           5m26s
controlplane:~/spring-boot-hello-world$ kubectl scale deployment hello-world-deployment -- replicas=1
error: required flag(s) "replicas" not set
controlplane:~/spring-boot-hello-world$ kubectl scale deployment hello-world-deployment --replicas=3
deployment.apps/hello-world-deployment scaled
controlplane:~/spring-boot-hello-world$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
hello-world-deployment-7ccfc5888f-2p5np 1/1     Running   0          6m44s
hello-world-deployment-7ccfc5888f-wzvpv 1/1     Running   0          6m44s
hello-world-deployment-7ccfc5888f-zm2rc 1/1     Running   0          6m44s
controlplane:~/spring-boot-hello-world$ kubectl scale deployment hello-world-deployment --replicas=1
deployment.apps/hello-world-deployment scaled
controlplane:~/spring-boot-hello-world$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
hello-world-deployment-7ccfc5888f-2p5np 1/1     Running   0          7m13s
controlplane:~/spring-boot-hello-world$
```

- Curled using clusterip because no public ip there.

```
controlplane:~/spring-boot-hello-world$ curl 10.107.241.147:8080/hello
Hello, World!
controlplane:~/spring-boot-hello-world$
```

- Pushed into github by using ssh

Kubernetes 1.32

This playground will always have the latest Kubeadm Kubernetes version a few weeks after release. You have access to an empty Kubeadm Kubernetes cluster with two 2GB nodes. The controlplane node has tainted...

```
controlplane:~/spring-boot-hello-world$ git config --global user.email "abhiabhirami242@gmail.com"
controlplane:~/spring-boot-hello-world$ git config --global user.name "Abhiramikannan"
controlplane:~/spring-boot-hello-world$ git commit -m "added dockerfile,deployment,service files"
[main a50efd2] added dockerfile,deployment,service files
3 files changed, 72 insertions(+)
create mode 100644 Dockerfile
create mode 100644 deployment.yaml
create mode 100644 service.yaml
controlplane:~/spring-boot-hello-world$ git push
Username for 'https://github.com': Abhiramikannan
Password for 'https://github.com': 
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/Abhiramikannan/spring-boot-hello-world.git/'
controlplane:~/spring-boot-hello-world$ git push origin
Username for 'https://github.com': Abhiramikannan
Password for 'https://github.com': 
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/Abhiramikannan/spring-boot-hello-world.git/'
controlplane:~/spring-boot-hello-world$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
```

```
controlplane:~$ cd spring-boot-hello-world/
controlplane:~/spring-boot-hello-world$ git push git@github.com:Abhiramikannan/spring-boot-hello-world.git
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.35 KiB | 1.35 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Abhiramikannan/spring-boot-hello-world.git
   3d3464b..a50efd2  main -> main
controlplane:~/spring-boot-hello-world$
```

- The deployment file

```
controlplane:~/spring-boot-hello-world$ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world-deployment
  labels:
    app: java-application
spec:
  replicas: 3
  selector:
    matchLabels:
      app: java-application
  template:
    metadata:
      labels:
        app: java-application
    spec:
      containers:
      - name: java-application
        image: abhiramikannan/javaimage:latest
        ports:
        - containerPort: 8080
```

- The service file

```
controlplane:~/spring-boot-hello-world$ cat service.yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: NodePort
  selector:
    app: java-application
  ports:
    - port: 8080
      # By default and for convenience, the `targetPort` is set to
      # the same value as the `port` field.
      targetPort: 8080
      # Optional field
      # By default and for convenience, the Kubernetes control plane
      # will allocate a port from a range (default: 30000-32767)
      nodePort: 30007
```