

Assignment: Dockerizing a Java Application and Deploying it on Kubernetes

Name: Akash Nadigepu

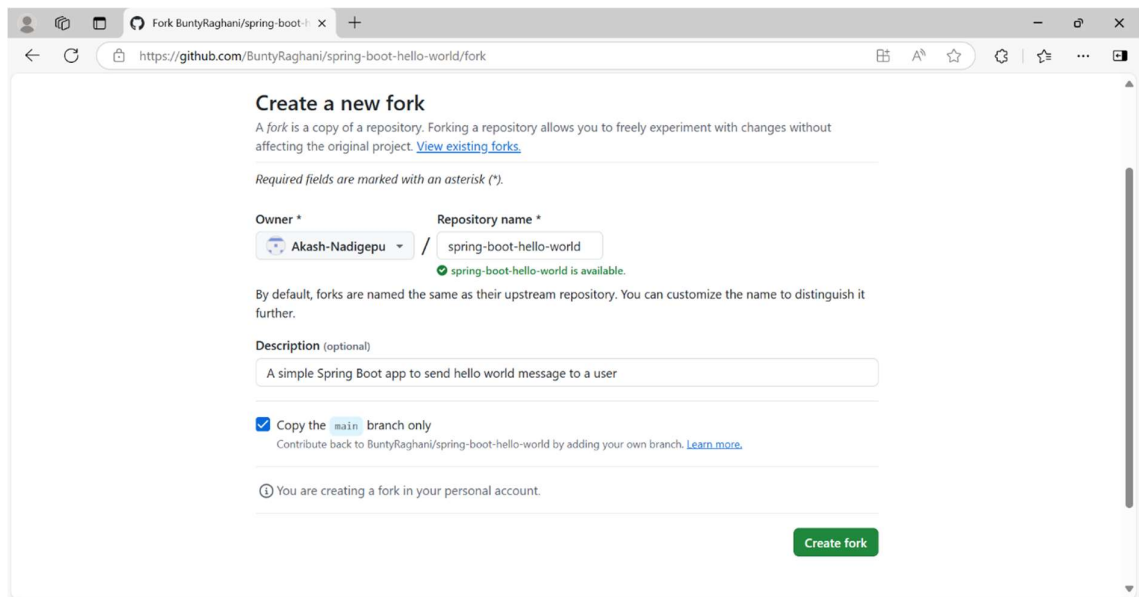
ID: 290396

Github repo: [Akash-Nadigepu/spring-boot-hello-world: A simple Spring Boot app to send hello world message to a user](#)

DockerHub repo: [akash6637/spring-boot-hello-world general | Docker Hub](#)

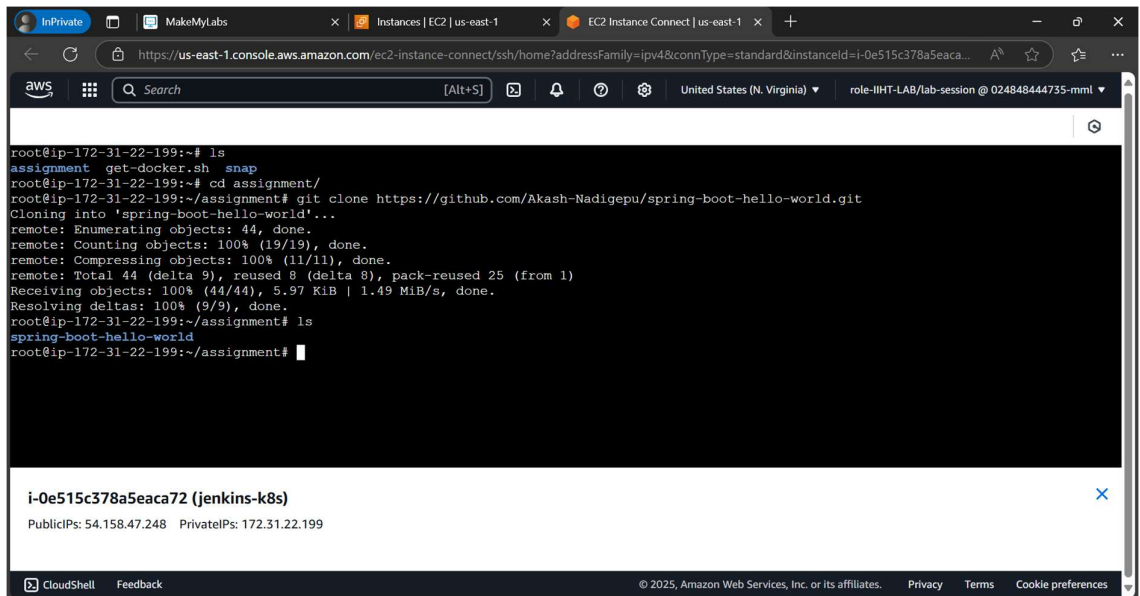
Task 1: Fork and Clone Java Repo

1. Fork the repository on GitHub.



The screenshot shows the GitHub 'Create a new fork' page. The browser address bar displays 'https://github.com/BuntyRaghani/spring-boot-hello-world/fork'. The page title is 'Create a new fork'. Below the title, a brief explanation of forking is provided, along with a link to 'View existing forks'. A note states 'Required fields are marked with an asterisk (*)'. The 'Owner' field is set to 'Akash-Nadigepu' and the 'Repository name' field is 'spring-boot-hello-world', with a green checkmark indicating 'spring-boot-hello-world is available'. A descriptive text explains that forks are named the same as their upstream repository by default. The 'Description (optional)' field contains the text 'A simple Spring Boot app to send hello world message to a user'. The 'Copy the main branch only' checkbox is checked, with a note to 'Contribute back to BuntyRaghani/spring-boot-hello-world by adding your own branch' and a link to 'Learn more'. A footer note states 'You are creating a fork in your personal account.' A green 'Create fork' button is located at the bottom right.

2. Cloned it to into local machine.

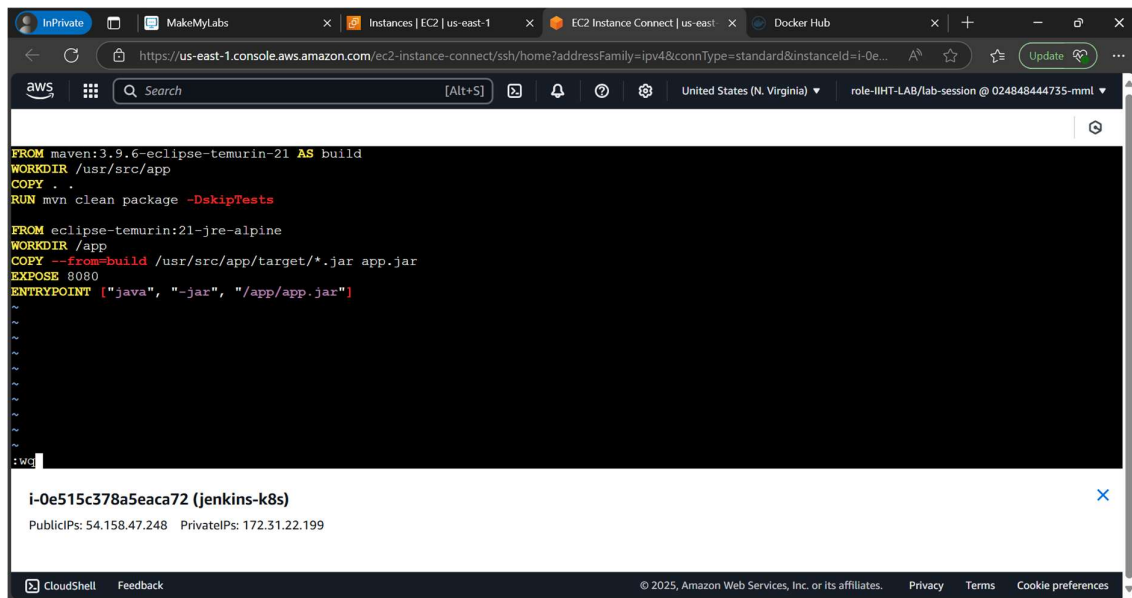


The screenshot shows a terminal window in the AWS CloudShell interface. The user is logged in as 'role-IHHT-LAB/lab-session @ 024848444735-mm1'. The terminal output shows the following commands and their results:

```
root@ip-172-31-22-199:~# ls
assignment  get-docker.sh  snap
root@ip-172-31-22-199:~# cd assignment/
root@ip-172-31-22-199:~/assignment# git clone https://github.com/Akash-Nadigepu/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 | 1.49 MiB/s, done.
Resolving deltas: 100% (9/9), done.
root@ip-172-31-22-199:~/assignment# ls
spring-boot-hello-world
root@ip-172-31-22-199:~/assignment#
```

Below the terminal output, the instance ID is displayed: **i-0e515c378a5eaca72 (jenkins-k8s)**. The public IP is 54.158.47.248 and the private IP is 172.31.22.199.

Task 2: Define a Dockerfile



The screenshot shows a terminal window in the AWS CloudShell interface. The user is logged in as 'role-IHHT-LAB/lab-session @ 024848444735-mm1'. The terminal output shows the following Dockerfile content:

```
FROM maven:3.9.6-eclipse-temurin-21 AS build
WORKDIR /usr/src/app
COPY . .
RUN mvn clean package -DskipTests

FROM eclipse-temurin:21-jre-alpine
WORKDIR /app
COPY --from=build /usr/src/app/target/*.jar app.jar
EXPOSE 8080
ENTRYPOINT ["java", "-jar", "/app/app.jar"]
```

Below the terminal output, the instance ID is displayed: **i-0e515c378a5eaca72 (jenkins-k8s)**. The public IP is 54.158.47.248 and the private IP is 172.31.22.199.

Task 3: Build and Push Docker Image

1. Build the image:

```
root@ip-172-31-22-199:~/assignment/spring-boot-hello-world# docker build -t akash6637/spring-boot-hello-world:latest .
[+] Building 24.1s (15/15) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 305B
=> [internal] load metadata for docker.io/library/eclipse-temurin:21-jre-alpine
=> [internal] load metadata for docker.io/library/maven:3.9.6-eclipse-temurin-21
=> [auth] library/maven:pull token for registry-1.docker.io
=> [auth] library/eclipse-temurin:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [build 1/4] FROM docker.io/library/maven:3.9.6-eclipse-temurin-21@sha256:8d63d4c1902cb12d9e79a70671b18ebe26358cb592561af33ca1808f00d 11.5s
=> => resolve docker.io/library/maven:3.9.6-eclipse-temurin-21@sha256:8d63d4c1902cb12d9e79a70671b18ebe26358cb592561af33ca1808f00d 0.0s
=> => sha256:62ed8d83373096a7ba983899af5dfbe2bd29c099913c0aa74c6217132e90195d 2.41kB / 2.41kB
=> => sha256:c38802599f183312ded6259ff07d25b4d2ac6b9e8bb0bac3bf315415540dea6 7.63kB / 7.63kB
=> => sha256:4a023cab5400feb5c1ab725beb8345ddb0e3200314004b56677a5eee2e8c86cf 30.44MB / 30.44MB
=> => sha256:5e5d1bccc5440d3a24f4a620704b9e687b4163c6c872fcc8e812e200c9bbac58 17.46MB / 17.46MB
=> => sha256:afd7ada947ce7661ad85815ab478851ccf7b0d064cde9c9195bafb2bd499b29e 158.51MB / 158.51MB
=> => sha256:8d63d4c1902cb12d9e79a70671b18ebe26358cb592561af33ca1808f00d935cb 979B / 979B
=> => sha256:478e5694bb3e0cbd54a331514092737b8a8de5a4fe7069a3f00e6eb1b98fd605 178B / 178B
=> => sha256:64fc845fc4fc373053c33ba34eb50ffbf689714b0ae2ef7729f42c6ce138f12e 19.00MB / 19.00MB
=> => sha256:d8fa38685928ae1a460295486cb01d7d2db1eff47fb68c8ac6c936e89af1641 731B / 731B

i-Oe515c378a5eaca72 (jenkins-k8s)
PublicIPs: 54.158.47.248 PrivateIPs: 172.31.22.199
```

2. Run the Container Locally

```
root@ip-172-31-22-199:~# docker run -p 9090:8080 akash6637/spring-boot-hello-world:latest

:: Spring Boot :: (v2.7.18)

2025-03-29 06:03:40.741 INFO 1 --- [main] c.e.helloworld.HelloWorldApplication : Starting HelloWorldApplication v1.0.2-SNAPSHOT using Java 21.0.6 on 3b0d24da6dda with PID 1 (/app/app.jar started by root in /app)
2025-03-29 06:03:40.752 INFO 1 --- [main] c.e.helloworld.HelloWorldApplication : No active profile set, falling back to 1 default profile: "default"
2025-03-29 06:03:42.337 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2025-03-29 06:03:42.353 INFO 1 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-03-29 06:03:42.353 INFO 1 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.83]
2025-03-29 06:03:42.477 INFO 1 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
```

3. Checking if it's running:

```
aws Search [Alt+S] United States (N. Virginia) role-IIHT-LAB/lab-session @ 02484844735-mml

root@ip-172-31-22-199:~# docker run -d -p 9090:8080 akash6637/spring-boot-hello-world:latest
4747c44f5f384c8d7f41431dad96e587bebe5ba6e1dd6101ffe3a4f846ad317b
root@ip-172-31-22-199:~# curl http://localhost:9090/hello
Hello, World!root@ip-172-31-22-199:~#
```

Worked! It is printing **Hello, World!**

4. Push to DockerHub:

aws console

https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0e... United States (N. Virginia) role-IHNT-LAB/lab-session @ 024848444735-mm1

```
root@ip-172-31-22-199:~# docker run -d -p 9090:8080 akash6637/spring-boot-hello-world:latest
4747c44f5f384c8d7f41431dad96e587bebe5ba6e1dd6101ffe3a4f846ad317b
root@ip-172-31-22-199:~# curl http://localhost:9090/hello
Hello, World!root@ip-172-~#
root@ip-172-31-22-199:~# docker push akash6637/spring-boot-hello-world:latest
The push refers to repository [docker.io/akash6637/spring-boot-hello-world]
270788f4e4e5: Pushed
37ff237e4508: Pushed
4a31297e6baa: Mounted from library/eclipse-temurin
822032205b9c: Mounted from library/eclipse-temurin
93509ae705ea: Mounted from library/eclipse-temurin
8f5df01935a3: Mounted from library/eclipse-temurin
08000c18d16d: Mounted from library/eclipse-temurin
latest: digest: sha256:94a8e13d1ad53ea788945fe78fa58cd704b067f28a6276be92f2a0b6c63cad75 size: 1785
root@ip-172-31-22-199:~#
```

i-0e515c378a5eaca72 (jenkins-k8s)

PublicIPs: 54.158.47.248 PrivateIPs: 172.31.22.199

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

dockerhub

https://hub.docker.com/repositories/akash6637

New Introducing our new CEO Don Johnson - Read More →

Explore My Hub Search Docker Hub Ctrl+K ? ? ? ? A

akash6637 Docker Personal

Repositories Settings Default privacy Notifications Billing Usage Pulls Storage

Repositories

All repositories within the akash6637 namespace.

Search by repository name All content Create a repository

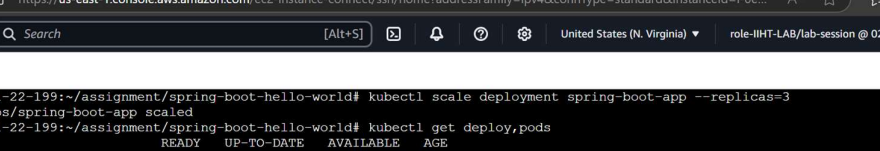
Name	Last Pushed ↑	Contains	Visibility	Scout
akash6637/spring-boot-hello-world	6 minutes ago	IMAGE	Public	Inactive
akash6637/java-web-app-cicd	about 19 hours ago	IMAGE	Public	Inactive
akash6637/my-springboot-app	7 days ago	IMAGE	Public	Inactive
akash6637/ubuntu	about 1 month ago	IMAGE	Public	Inactive
akash6637/maven_docker	about 1 month ago	IMAGE	Public	Inactive

Task 4: Create a Kubernetes Deployment

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and a list of services including CloudShell, Feedback, and a footer with copyright information. The main content area displays the configuration for a deployment named "i-0e515c378a5eaca72 (jenkins-k8s)". The configuration is shown in a YAML format, detailing the deployment type, metadata, spec, and template. The spec section indicates a single replica of the "spring-boot-app" service. The template section defines the container image as "akash6637/spring-boot-hello-world:latest" and maps port 8080 of the container to port 20 of the host. The deployment is running on the "PublicIPs: 54.158.47.248" and "PrivateIPs: 172.31.22.199" subnets. The bottom status bar shows the deployment is in a "Completed" state with a "20, 0-1" status.

```
root@ip-172-31-22-199:~/assignment/spring-boot-hello-world# vi deployment.yaml
root@ip-172-31-22-199:~/assignment/spring-boot-hello-world# kubectl apply -f deployment.yaml
deployment.apps/spring-boot-app created
root@ip-172-31-22-199:~/assignment/spring-boot-hello-world# kubectl get pods
NAME                                READY    STATUS    RESTARTS   AGE
spring-boot-app-646c4cb966-clptb    1/1      Running   0           11s
root@ip-172-31-22-199:~/assignment/spring-boot-hello-world# kubectl get deploy
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
spring-boot-app                    1/1      1              1            26s
root@ip-172-31-22-199:~/assignment/spring-boot-hello-world#
```

Task 5: Scale Up and Scale Down



The screenshot shows a terminal window with the following content:

```
root@ip-172-31-22-199:~/assignment/spring-boot-hello-world# kubectl scale deployment spring-boot-app --replicas=3
deployment.apps/spring-boot-app scaled
root@ip-172-31-22-199:~/assignment/spring-boot-hello-world# kubectl get deploy,pods
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/spring-boot-app    3/3      3              3            6m58s

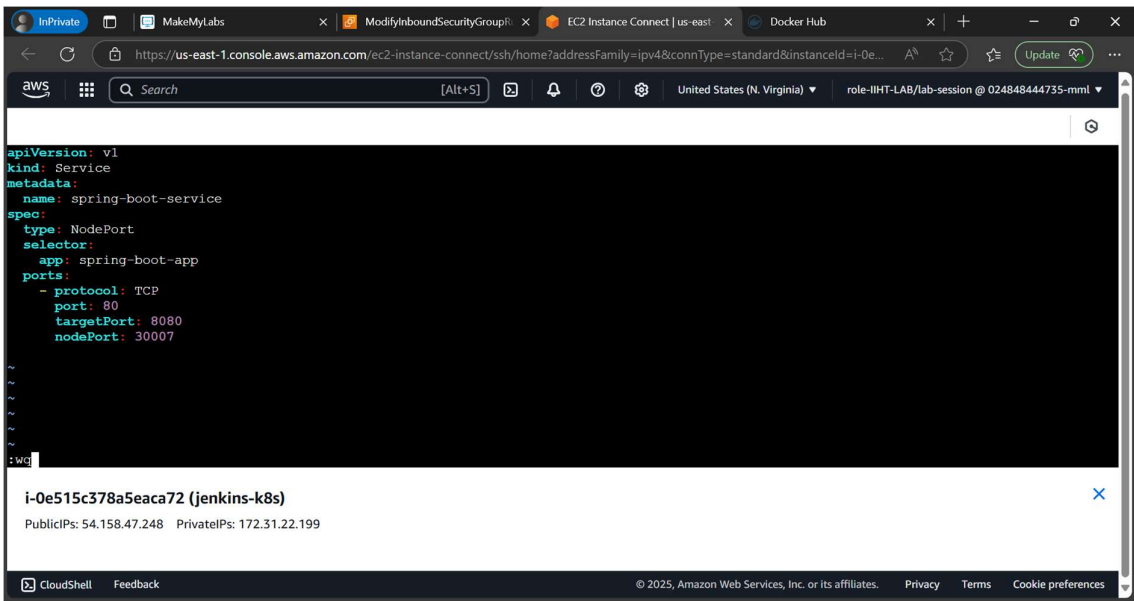
NAME                                STATUS    RESTARTS    AGE
pod/spring-boot-app-646c4cb966-clptb 1/1      Running     0            6m58s
pod/spring-boot-app-646c4cb966-czt6c 1/1      Running     0            11s
pod/spring-boot-app-646c4cb966-lkqmw 1/1      Running     0            11s
root@ip-172-31-22-199:~/assignment/spring-boot-hello-world# kubectl scale deployment spring-boot-app --replicas=1
deployment.apps/spring-boot-app scaled
root@ip-172-31-22-199:~/assignment/spring-boot-hello-world# kubectl get deploy,pods
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/spring-boot-app    1/1      1              1            7m51s

NAME                                STATUS    RESTARTS    AGE
pod/spring-boot-app-646c4cb966-clptb 1/1      Running     0            7m51s
root@ip-172-31-22-199:~/assignment/spring-boot-hello-world#
```

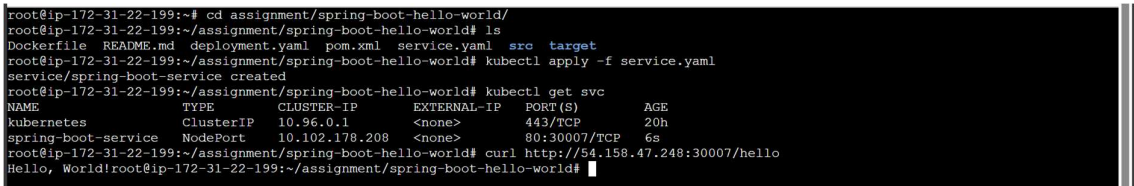
Below the terminal output, there is a Jenkins job ID and its public IP address:

i-0e515c378a5eaca72 (jenkins-k8s)
PublicIPs: 54.158.47.248 PrivateIPs: 172.31.22.199

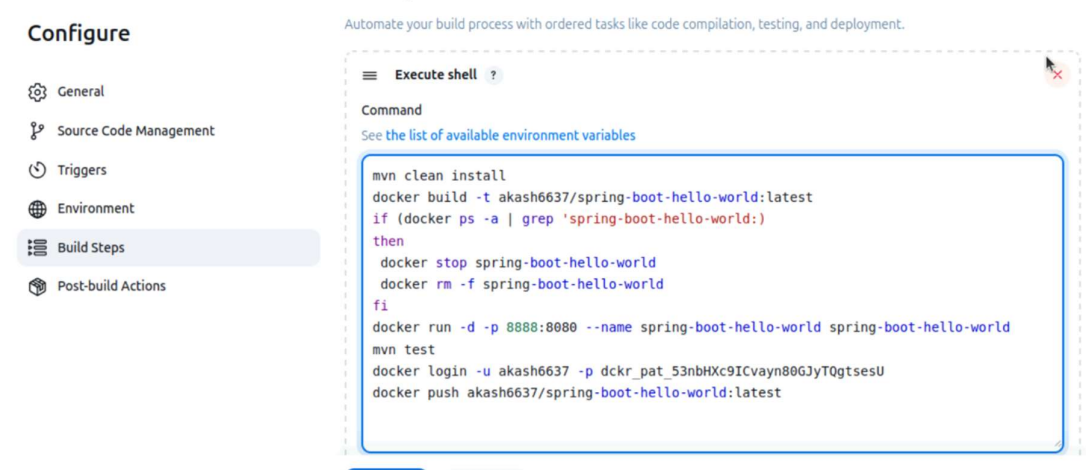
Task 6: Expose the Service Using NodePort



Accessing the Application using NodePort :30007



Task 7: Automate Deployment Using Jenkins



Status

</> Changes

Workspace

Build Now

Configure

Delete Project

Rename

✓ java-hello-world

Add description

Permalinks

- [Last build \(#2\), 2 hr 2 min ago](#)
- [Last stable build \(#2\), 2 hr 2 min ago](#)
- [Last successful build \(#2\), 2 hr 2 min ago](#)
- [Last failed build \(#1\), 2 hr 3 min ago](#)
- [Last unsuccessful build \(#1\), 2 hr 3 min ago](#)
- [Last completed build \(#2\), 2 hr 2 min ago](#)

Builds

*** ↗

Filter

/

Today

✓ #2 5:40 AM

▼