CONTAINERIZING A SPRING BOOT APPLICATION AND DEPLOYING IT ON KUBERNETES

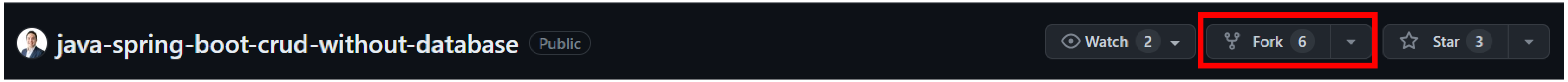
- **Description:**

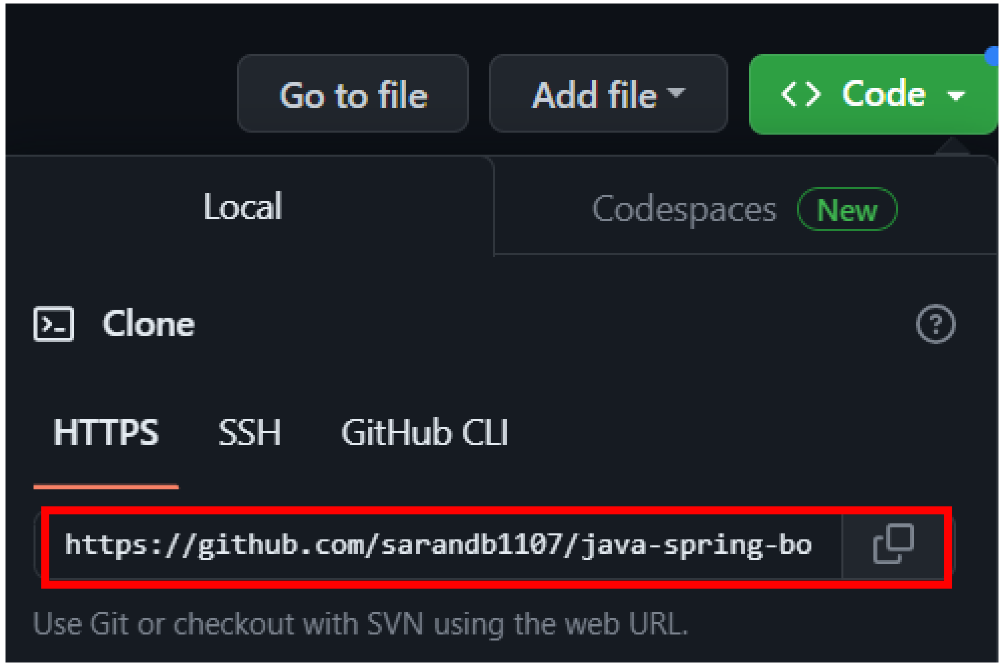
  Your task is to containerize a Spring boot application that serves HTTP requests.

- **Requirements:**

1. The application should be containerized with Docker. The application source can be found at https://github.com/adanyc/java-spring-boot-crud-without-database
2. The docker image build along with the maven build and test should be done in a CI pipeline of your choice.
3. The application should be deployed on Kubernetes
4. The CI pipeline should also be able to perform CD by deploying the new image to Kubernetes.
5. The application should be exposed with a service and optionally with an ingress if it is a cloud-managed Kubernetes cluster
6. The manifest files should be stored in a public git repository.

Creating a copy of the given repository in GitHub by clicking the fork.

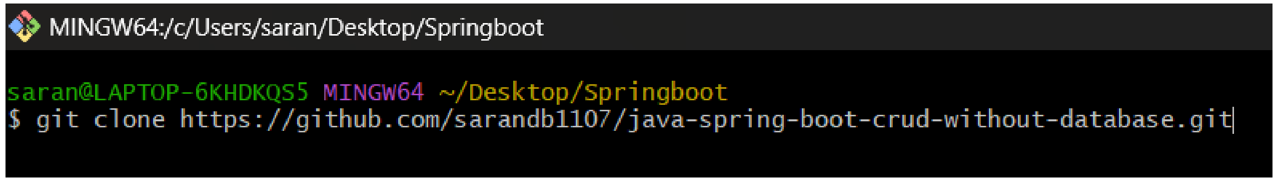




```
https://github.com/sarandb1107/java-spring-bo
```

Use Git or checkout with SVN using the web URL.

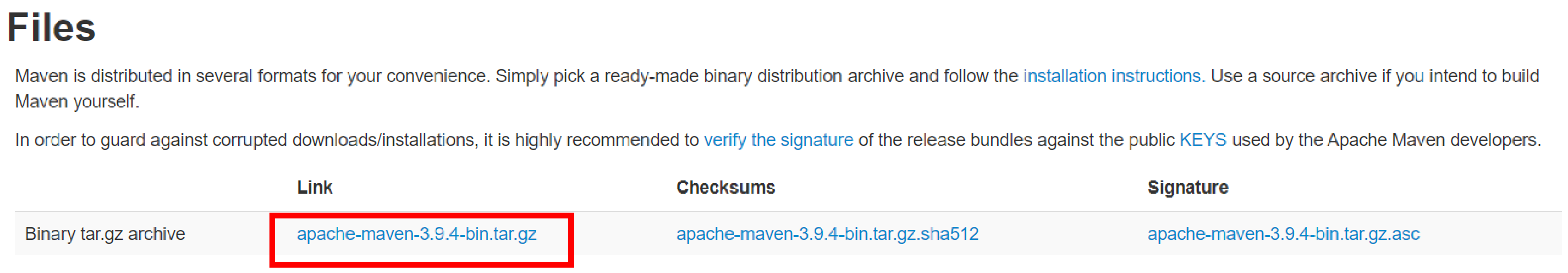
A copy of the repository is created in the local system by using



```
MINGW64:/c/Users/saran/Desktop/Springboot

saran@LAPTOP-6KHDKQS5 MINGW64 ~/Desktop/Springboot
$ git clone https://github.com/sarandb1107/java-spring-boot-crud-without-database.git
```

Maven Installation

Installing Maven by following the steps given in the Maven documentation.



Files

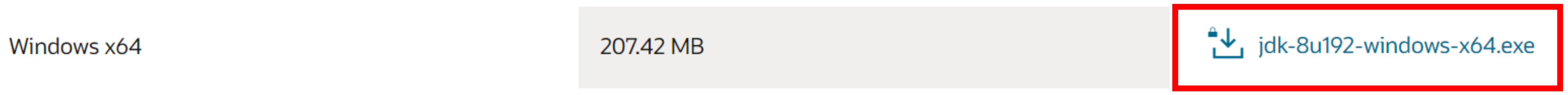
Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions. Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.9.4-bin.tar.gz	apache-maven-3.9.4-bin.tar.gz.sha512	apache-maven-3.9.4-bin.tar.gz.asc

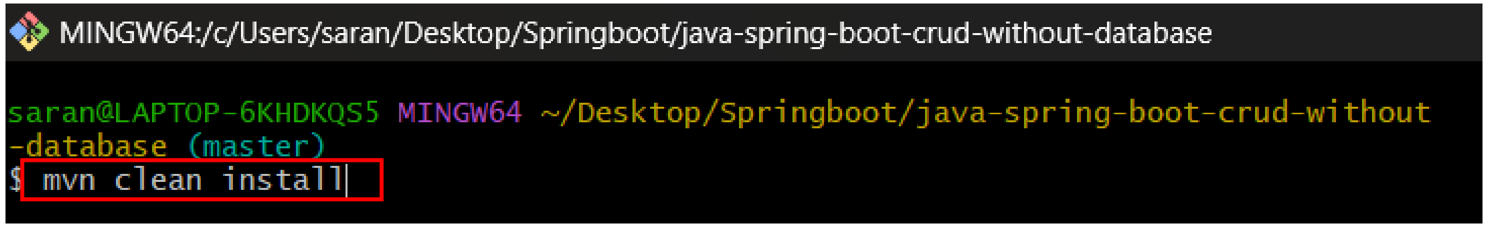
Java SE Development Kit 8u192

Installing JDK-8 by following the steps given in the Oracle documentation



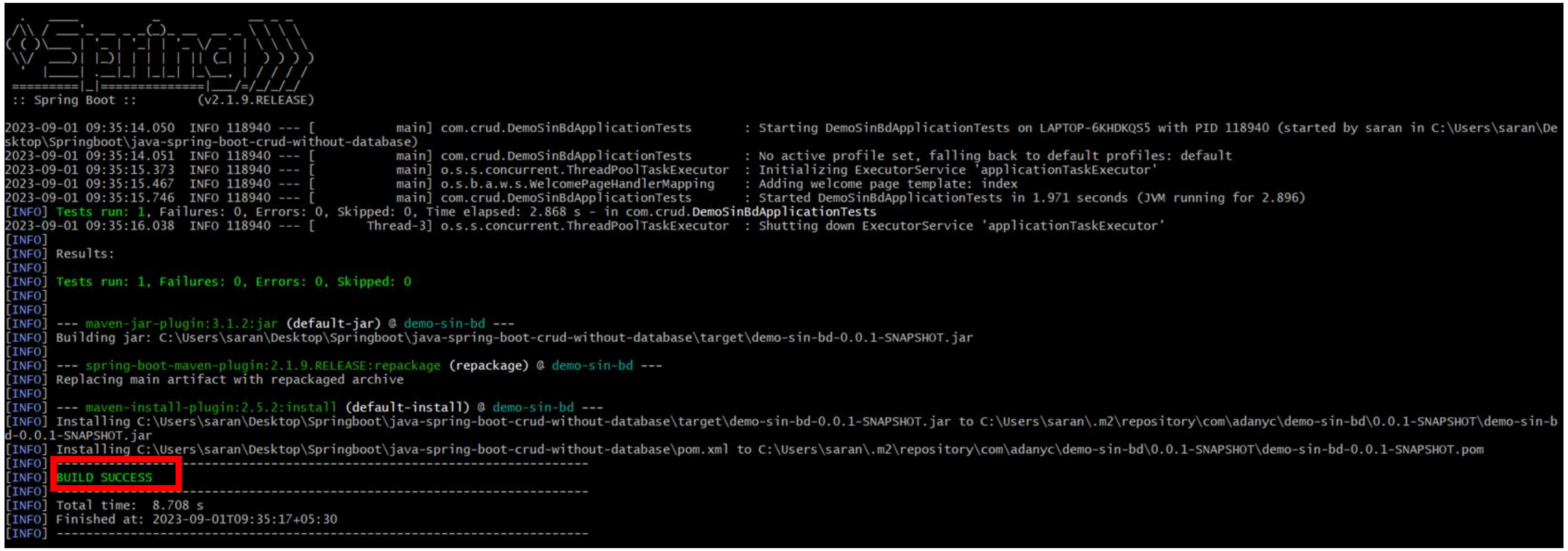
Windows x64	207.42 MB	jdk-8u192-windows-x64.exe

After the installation of Maven and Java, the Maven life cycle is executed.



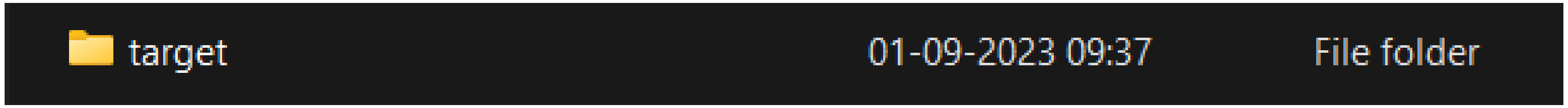
```
MINGW64:/c/Users/saran/Desktop/Springboot/java-spring-boot-crud-without-database

saran@LAPTOP-6KHDKQS5 MINGW64 ~/Desktop/Springboot/java-spring-boot-crud-without
-database (master)
$ mvn clean install
```

The Build is successfully Completed

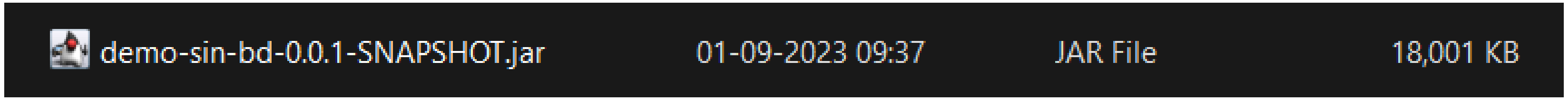


```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.1.9.RELEASE)

2023-09-01 09:35:14.050  INFO 118940 --- [           main] com.crud.DemoSinBdApplicationTests       : Starting DemoSinBdApplicationTests on LAPTOP-6KHDKQS5 with PID 118940 (started by saran in C:\Users\saran\De
sktop\Springboot\java-spring-boot-crud-without-database)
2023-09-01 09:35:14.051  INFO 118940 --- [           main] com.crud.DemoSinBdApplicationTests       : No active profile set, falling back to default profiles: default
2023-09-01 09:35:15.373  INFO 118940 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2023-09-01 09:35:15.467  INFO 118940 --- [           main] o.s.b.a.w.s.WelcomePageHandlerMapping    : Adding welcome page template: index
2023-09-01 09:35:15.746  INFO 118940 --- [           main] com.crud.DemoSinBdApplicationTests       : Started DemoSinBdApplicationTests in 1.971 seconds (JVM running for 2.896)
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.868 s - in com.crud.DemoSinBdApplicationTests
2023-09-01 09:35:16.038  INFO 118940 --- [       Thread-3] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shutting down ExecutorService 'applicationTaskExecutor'
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.1.2:jar (default-jar) @ demo-sin-bd ---
[INFO] Building jar: C:\Users\saran\Desktop\Springboot\java-spring-boot-crud-without-database\target\demo-sin-bd-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.1.9.RELEASE:repackage (repackage) @ demo-sin-bd ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ demo-sin-bd ---
[INFO] Installing C:\Users\saran\Desktop\Springboot\java-spring-boot-crud-without-database\target\demo-sin-bd-0.0.1-SNAPSHOT.jar to C:\Users\saran\.m2\repository\com\adanyc\demo-sin-bd\0.0.1-SNAPSHOT\demo-sin-b
d-0.0.1-SNAPSHOT.jar
[INFO] Installing C:\Users\saran\Desktop\Springboot\java-spring-boot-crud-without-database\pom.xml to C:\Users\saran\.m2\repository\com\adanyc\demo-sin-bd\0.0.1-SNAPSHOT\demo-sin-bd-0.0.1-SNAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  8.708 s
[INFO] Finished at: 2023-09-01T09:35:17+05:30
[INFO] ------------------------------------------------------------------------
```

After, the Execution of the Maven Life Cycle. The .jar file has been created.

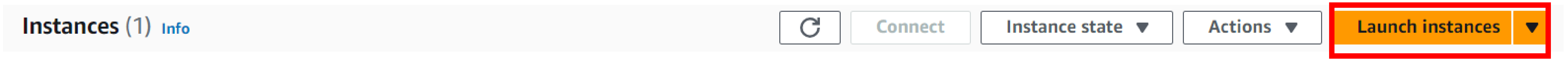


target 01-09-2023 09:37 File folder

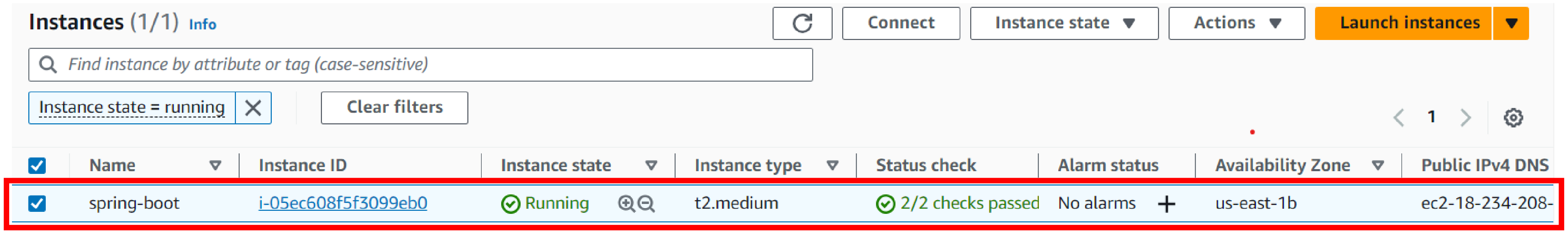


demo-sin-bd-0.0.1-SNAPSHOT.jar 01-09-2023 09:37 JAR File 18,001 KB

Launching an EC2 Instance with the help of AWS.

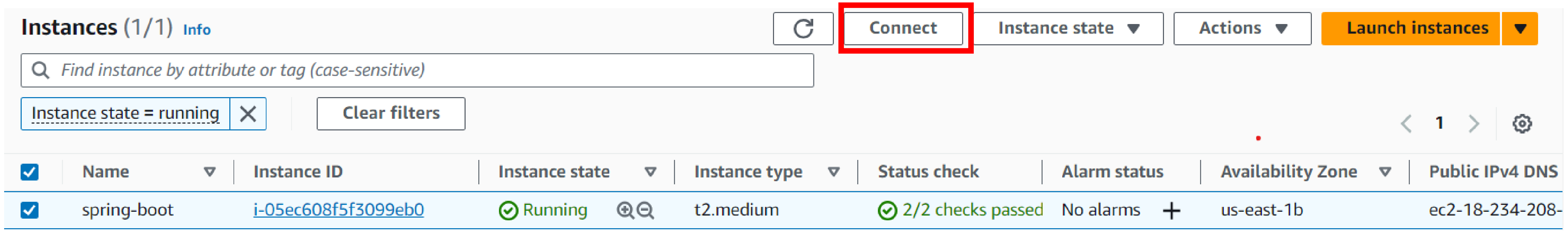


Spring-boot instances have been successfully launched.



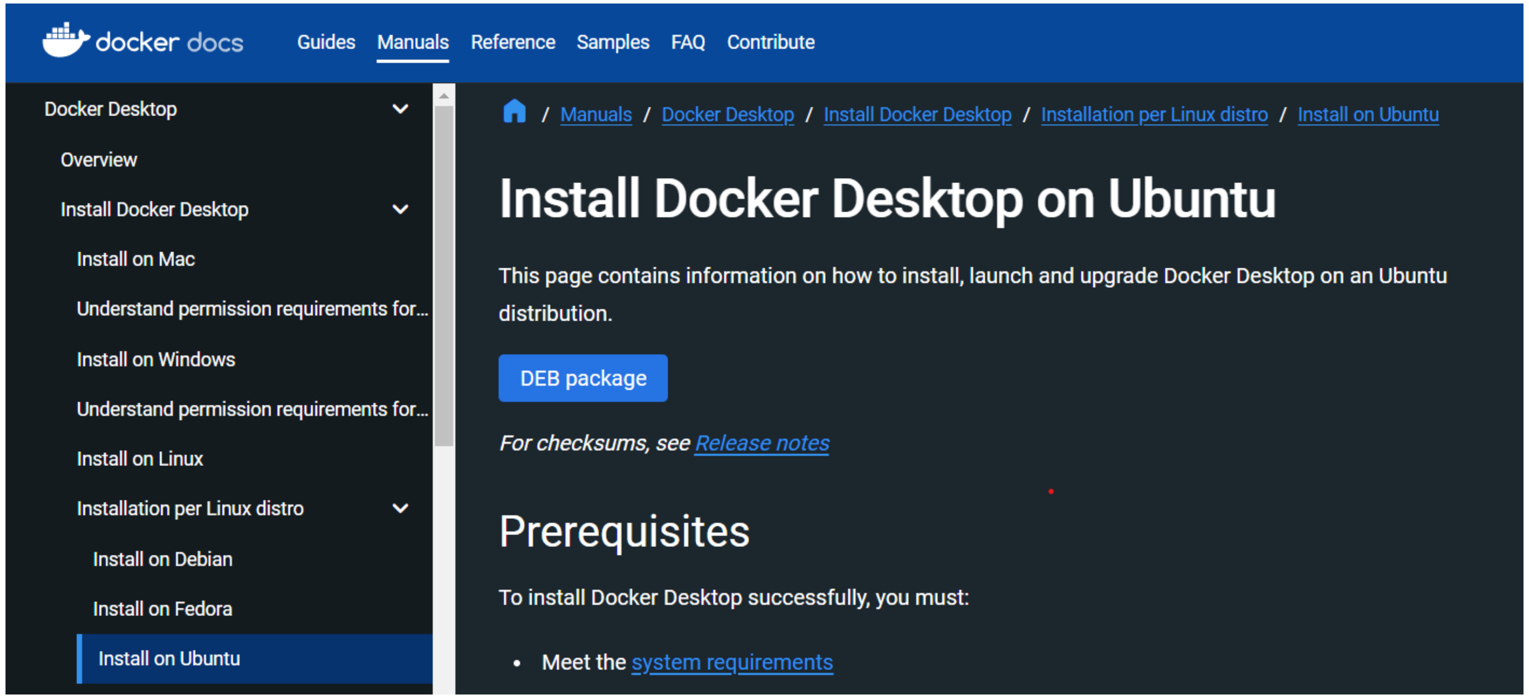
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
spring-boot	i-05ec608f5f3099eb0	Running	t2.medium	2/2 checks passed	No alarms	us-east-1b	ec2-18-234-208-

Connecting through AWS Connect



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
spring-boot	i-05ec608f5f3099eb0	Running	t2.medium	2/2 checks passed	No alarms	us-east-1b	ec2-18-234-208-

Installation of Docker Desktop on Ubuntu


docker docs
Guides
Manuals
Reference
Samples
FAQ
Contribute
Docker Desktop
Overview
Install Docker Desktop
Install on Mac
Understand permission requirements for...
Install on Windows
Understand permission requirements for...
Install on Linux
Installation per Linux distro
Install on Debian
Install on Fedora
Install on Ubuntu
/ Manuals / Docker Desktop / Install Docker Desktop / Installation per Linux distro / Install on Ubuntu
DEB package

Install Docker Desktop on Ubuntu

This page contains information on how to install, launch and upgrade Docker Desktop on an Ubuntu distribution.

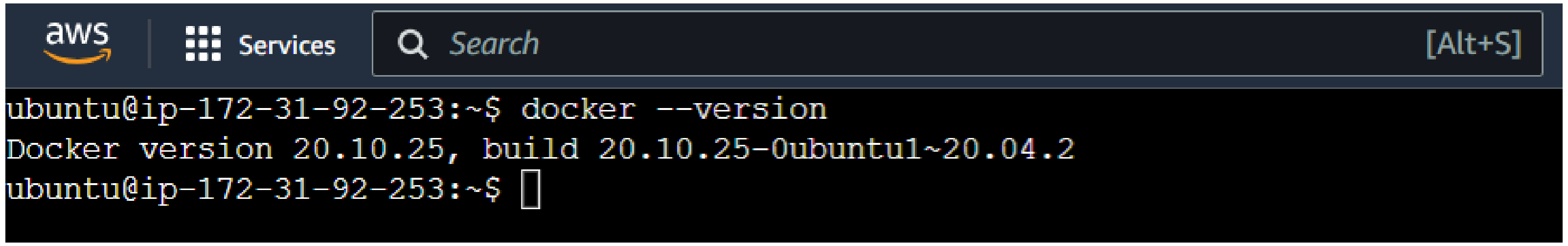
For checksums, see Release notes

Prerequisites

To install Docker Desktop successfully, you must:

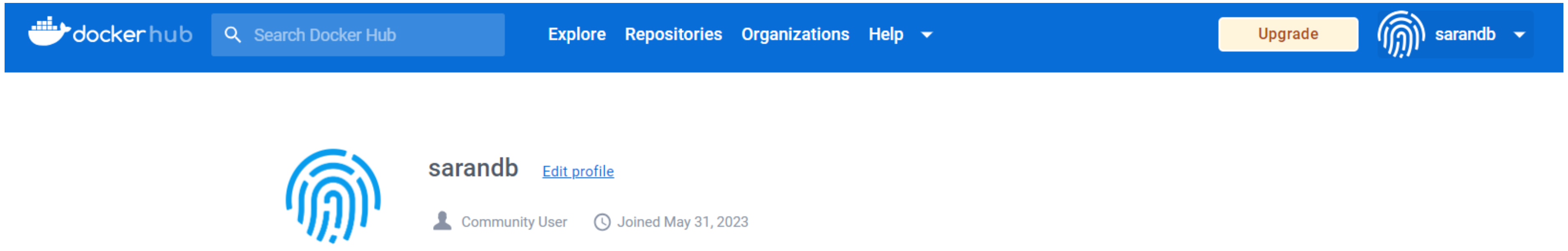
- Meet the system requirements

Checking the Docker version after the successful installation.

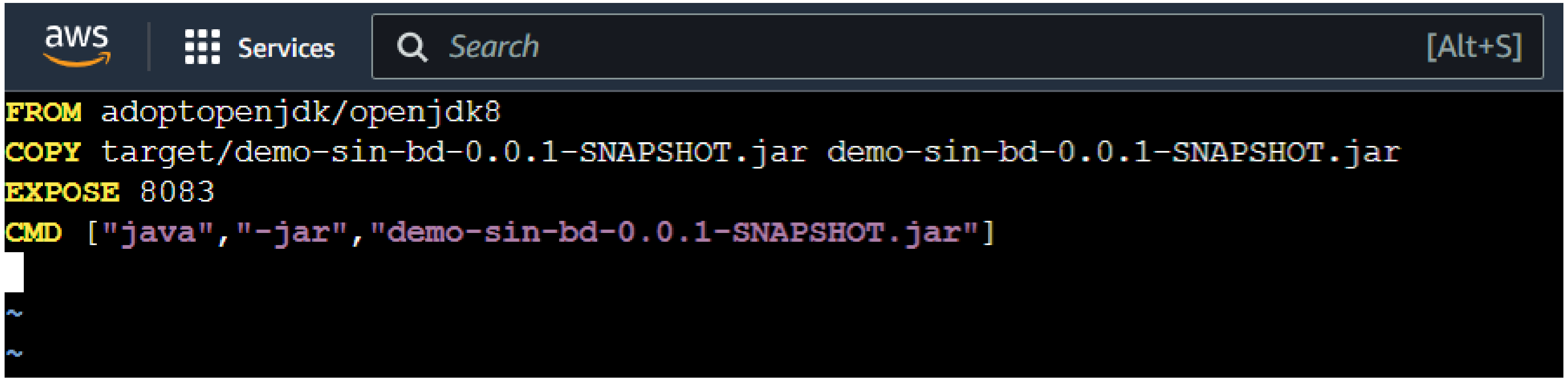


```
ubuntu@ip-172-31-92-253:~$ docker --version
Docker version 20.10.25, build 20.10.25-0ubuntu1~20.04.2
ubuntu@ip-172-31-92-253:~$ 
```

Creation of an Account in Docker Hub

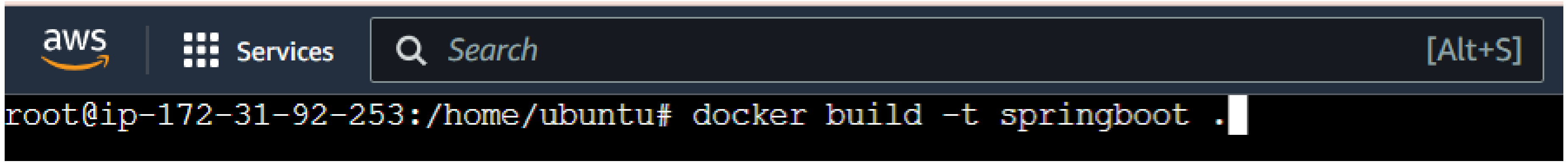


Next step is creating a Docker file

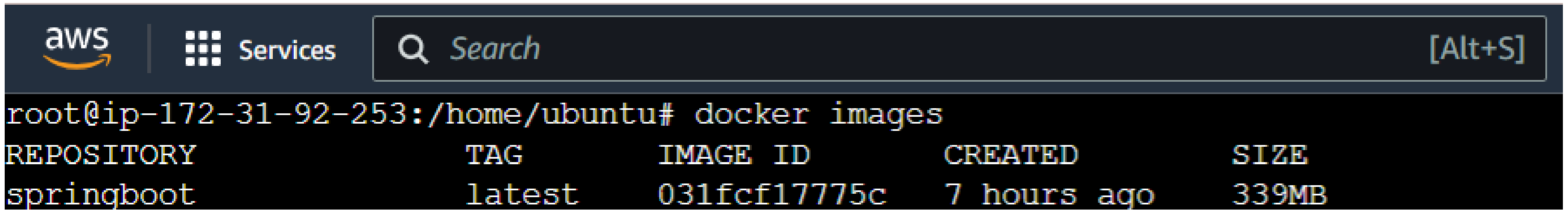


```
FROM adoptopenjdk/openjdk8
COPY target/demo-sin-bd-0.0.1-SNAPSHOT.jar demo-sin-bd-0.0.1-SNAPSHOT.jar
EXPOSE 8083
CMD ["java","-jar","demo-sin-bd-0.0.1-SNAPSHOT.jar"]

~
~
```

Building Docker Images

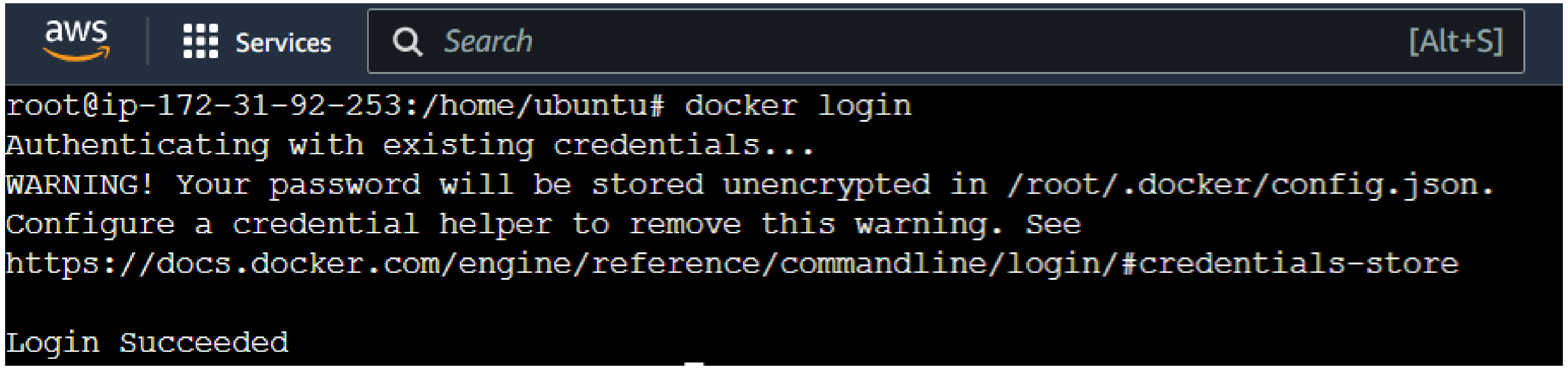


```
root@ip-172-31-92-253:/home/ubuntu# docker build -t springboot .
```



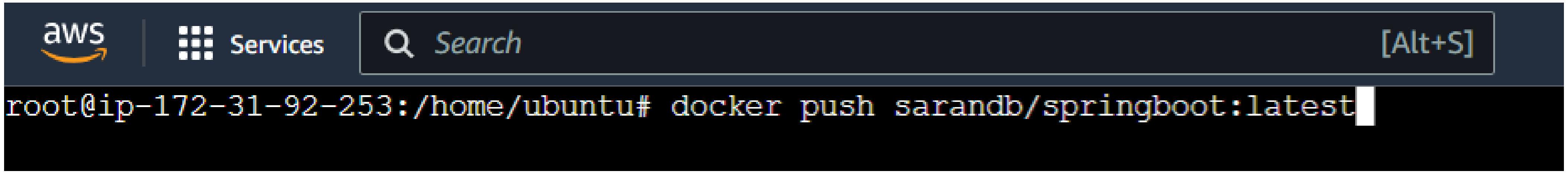
```
root@ip-172-31-92-253:/home/ubuntu# docker images
REPOSITORY                TAG         IMAGE ID       CREATED         SIZE
springboot                latest      031fcf17775c   7 hours ago     339MB
```

Docker Login



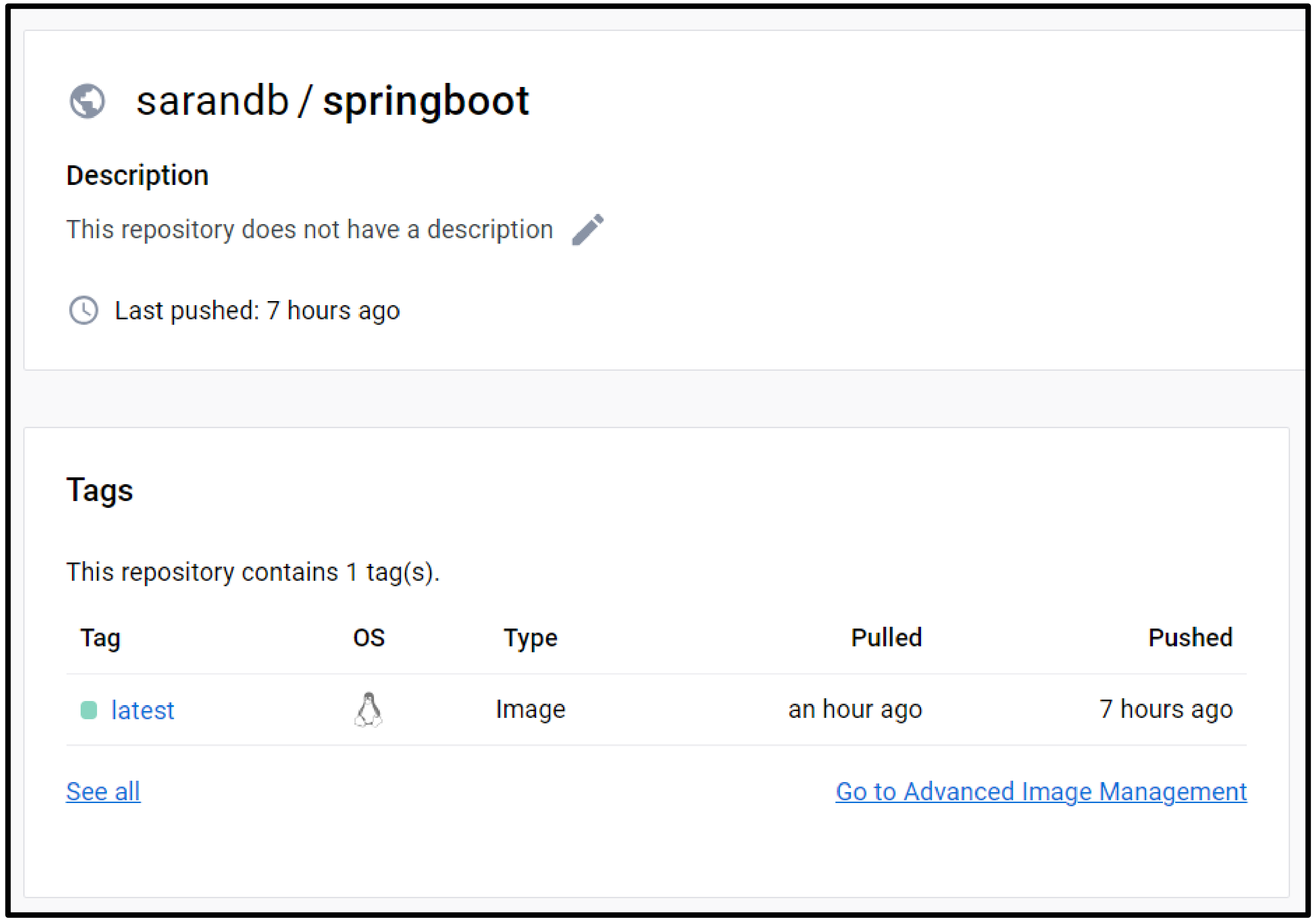
```
root@ip-172-31-92-253:/home/ubuntu# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Pushing the Docker Image to the Docker Registry



```
root@ip-172-31-92-253:/home/ubuntu# docker push sarandb/springboot:latest
```

Docker Image in GitHub



sarandb / **springboot**

Description

This repository does not have a description

Last pushed: 7 hours ago

Tags

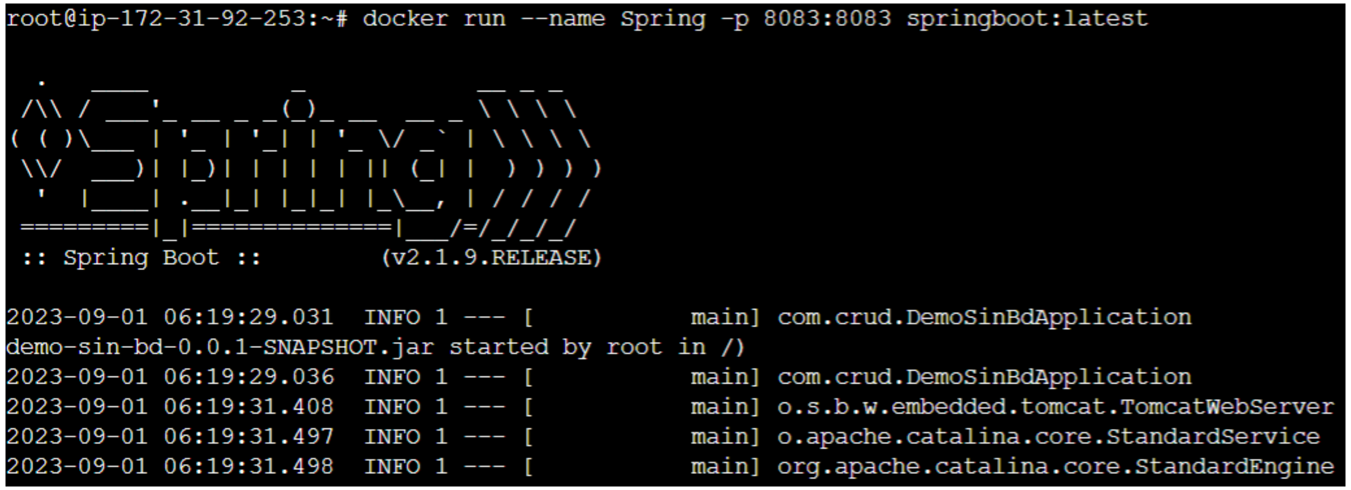
This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
🟩 latest		Image	an hour ago	7 hours ago

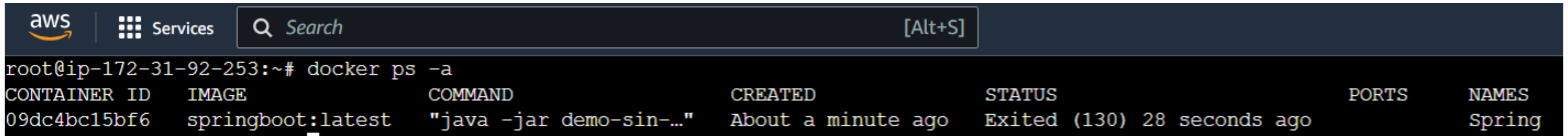
See all

Go to Advanced Image Management

Deploying Image in Container



```
root@ip-172-31-92-253:~# docker run --name Spring -p 8083:8083 springboot:latest

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.1.9.RELEASE)

2023-09-01 06:19:29.031  INFO 1 --- [           main] com.crud.DemoSinBdApplication
demo-sin-bd-0.0.1-SNAPSHOT.jar started by root in /)
2023-09-01 06:19:29.036  INFO 1 --- [           main] com.crud.DemoSinBdApplication
2023-09-01 06:19:31.408  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer
2023-09-01 06:19:31.497  INFO 1 --- [           main] o.apache.catalina.core.StandardService
2023-09-01 06:19:31.498  INFO 1 --- [           main] org.apache.catalina.core.StandardEngine
```

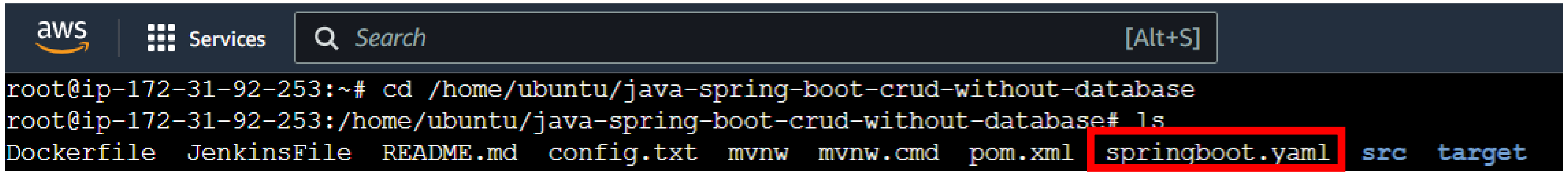


```
root@ip-172-31-92-253:~# docker ps -a
CONTAINER ID   IMAGE               COMMAND                  CREATED              STATUS                        PORTS     NAMES
09dc4bc15bf6   springboot:latest   "java -jar demo-sin-…"   About a minute ago   Exited (130) 28 seconds ago             Spring
```

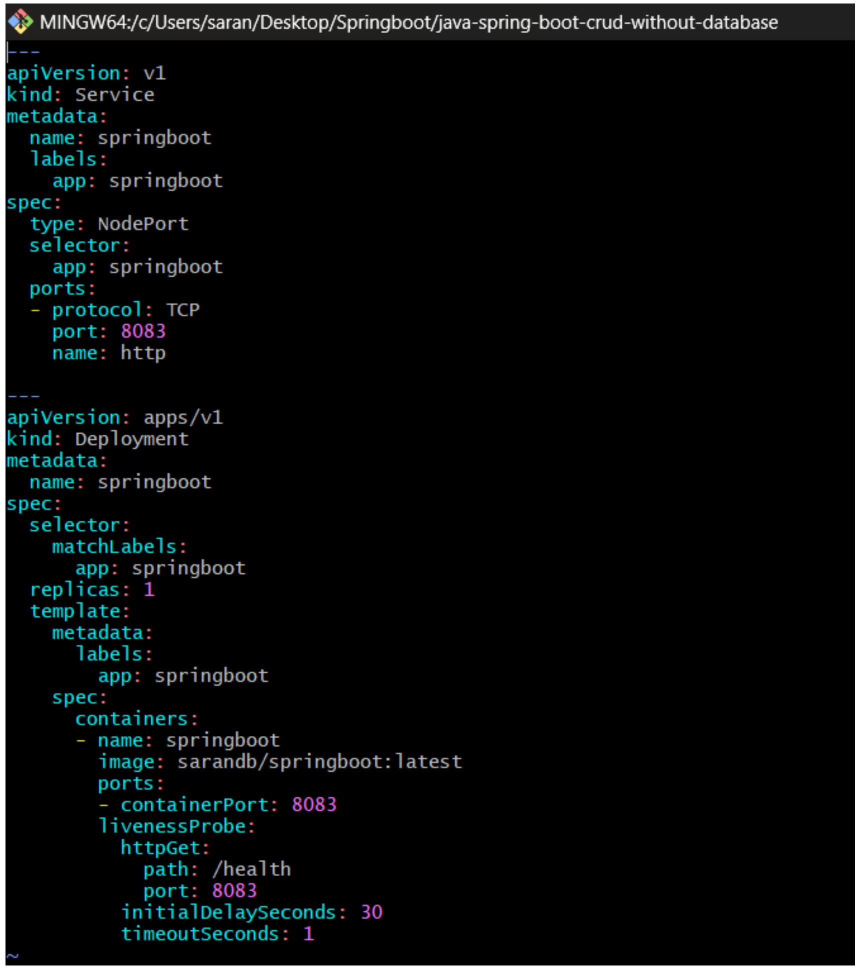
Kubernetes Installation

```
Kubernetes Installation:---
ubuntu20

Master

    1. apt update
    2  sudo mkdir -m 755 /etc/apt/keyrings
    3  apt install docker.io
    4  sudo apt-get install -y apt-transport-https ca-certificates curl
    5  curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.27/deb/Release.key | sudo
gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
    6  echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.27/deb/ /' | sudo tee
/etc/apt/sources.list.d/kubernetes.list
    7  apt update
    8  apt install -y kubeadm kubelet kubectl
    9  sudo kubeadm init --pod-network-cidr=192.168.0.0/16  &(copy the token)
   10  mkdir -p $HOME/.kube
   11  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
   12  sudo chown $(id -u):$(id -g) $HOME/.kube/config
   13  kubectl create -f
https://raw.githubusercontent.com/projectcalico/calico/v3.26.1/manifests/tigera-ope
rator.yaml
   14  kubectl create -f
https://raw.githubusercontent.com/projectcalico/calico/v3.26.1/manifests/custom-res
ources.yaml
   15  kubectl get nodes
   17  watch kubectl get pods -n calico-system
```

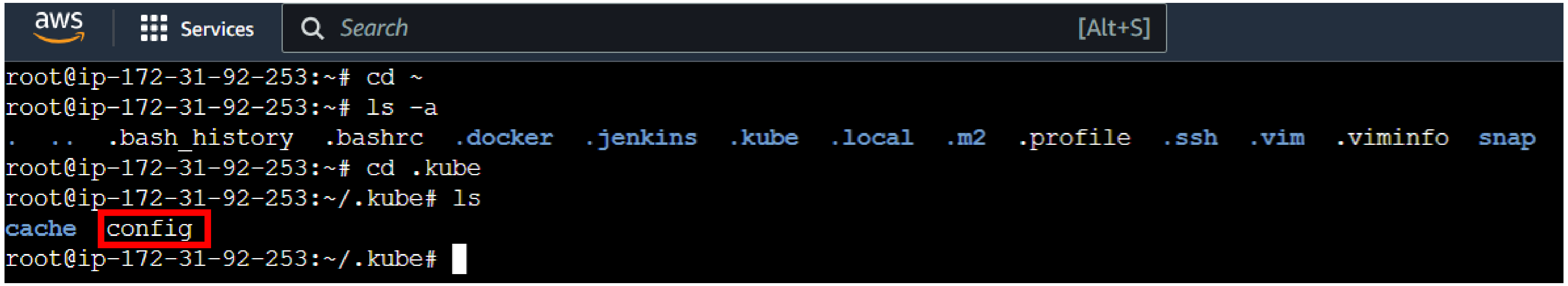
Creating Manifest File



```
root@ip-172-31-92-253:~# cd /home/ubuntu/java-spring-boot-crud-without-database
root@ip-172-31-92-253:/home/ubuntu/java-spring-boot-crud-without-database# ls
Dockerfile  JenkinsFile  README.md  config.txt  mvnw  mvnw.cmd  pom.xml  springboot.yaml  src  target
```

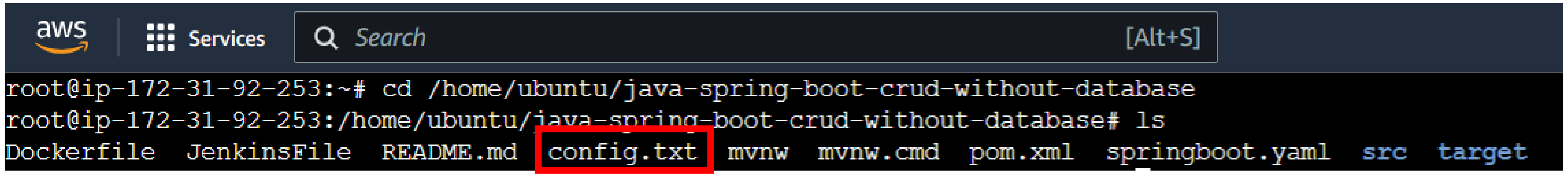

MINGW64:/c/Users/saran/Desktop/Springboot/java-spring-boot-crud-without-database

```
---
apiVersion: v1
kind: Service
metadata:
  name: springboot
  labels:
    app: springboot
spec:
  type: NodePort
  selector:
    app: springboot
  ports:
  - protocol: TCP
    port: 8083
    name: http

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: springboot
spec:
  selector:
    matchLabels:
      app: springboot
  replicas: 1
  template:
    metadata:
      labels:
        app: springboot
    spec:
      containers:
      - name: springboot
        image: sarandb/springboot:latest
        ports:
        - containerPort: 8083
        livenessProbe:
          httpGet:
            path: /health
            port: 8083
          initialDelaySeconds: 30
          timeoutSeconds: 1
~
```



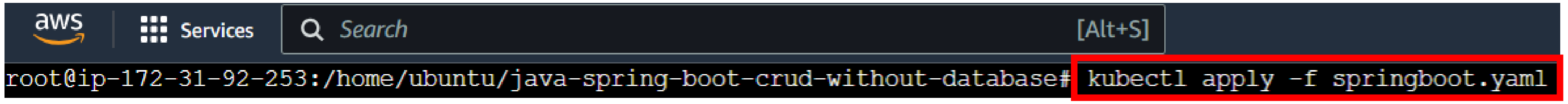
```
root@ip-172-31-92-253:~# cd ~
root@ip-172-31-92-253:~# ls -a
.  ..  .bash_history  .bashrc  .docker  .jenkins  .kube  .local  .m2  .profile  .ssh  .vim  .viminfo  snap
root@ip-172-31-92-253:~# cd .kube
root@ip-172-31-92-253:~/.kube# ls
cache  config
root@ip-172-31-92-253:~/.kube#
```

Copy all the configurations from config to config.txt in the local repository



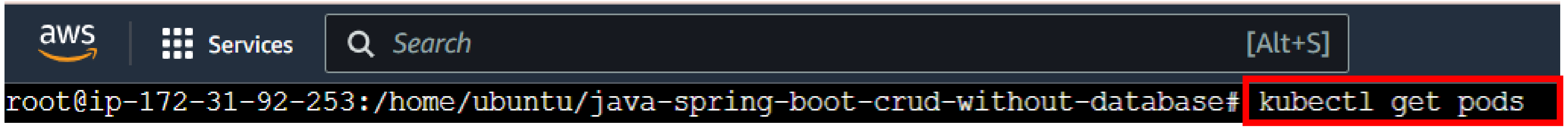
```
root@ip-172-31-92-253:~# cd /home/ubuntu/java-spring-boot-crud-without-database
root@ip-172-31-92-253:/home/ubuntu/java-spring-boot-crud-without-database# ls
Dockerfile  JenkinsFile  README.md  config.txt  mvnw  mvnw.cmd  pom.xml  springboot.yaml  src  target
```

Deploying the Manifest in the Kubernetes cluster



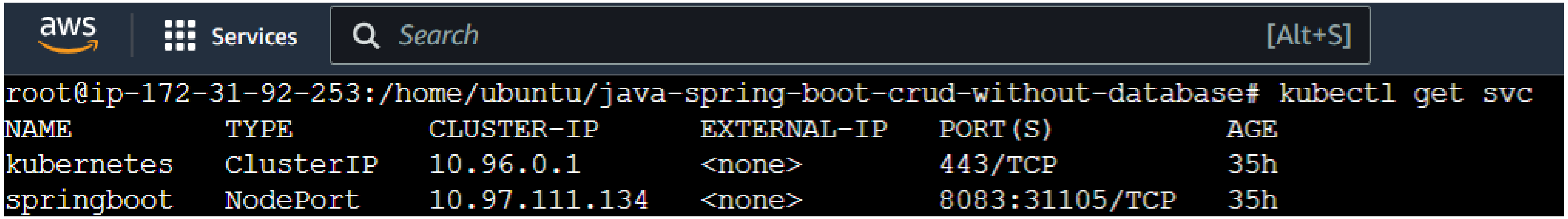
```
root@ip-172-31-92-253:/home/ubuntu/java-spring-boot-crud-without-database# kubectl apply -f springboot.yaml
```

Pods Running Check

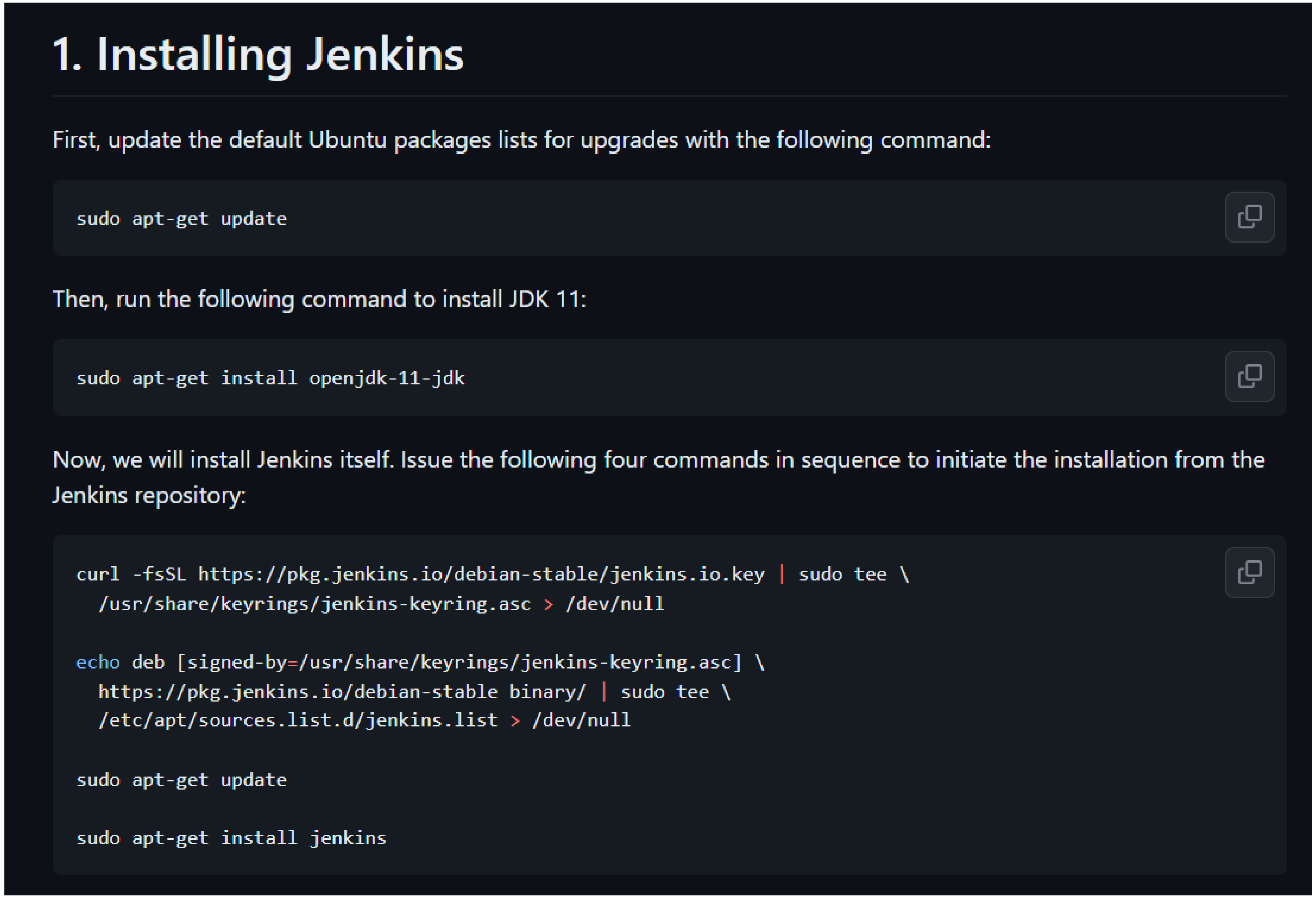


```
root@ip-172-31-92-253:/home/ubuntu/java-spring-boot-crud-without-database# kubectl get pods
```

Port range for accessing the application



```
root@ip-172-31-92-253:/home/ubuntu/java-spring-boot-crud-without-database# kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP          35h
springboot   NodePort    10.97.111.134   <none>        8083:31105/TCP   35h
```


1. Installing Jenkins

First, update the default Ubuntu packages lists for upgrades with the following command:

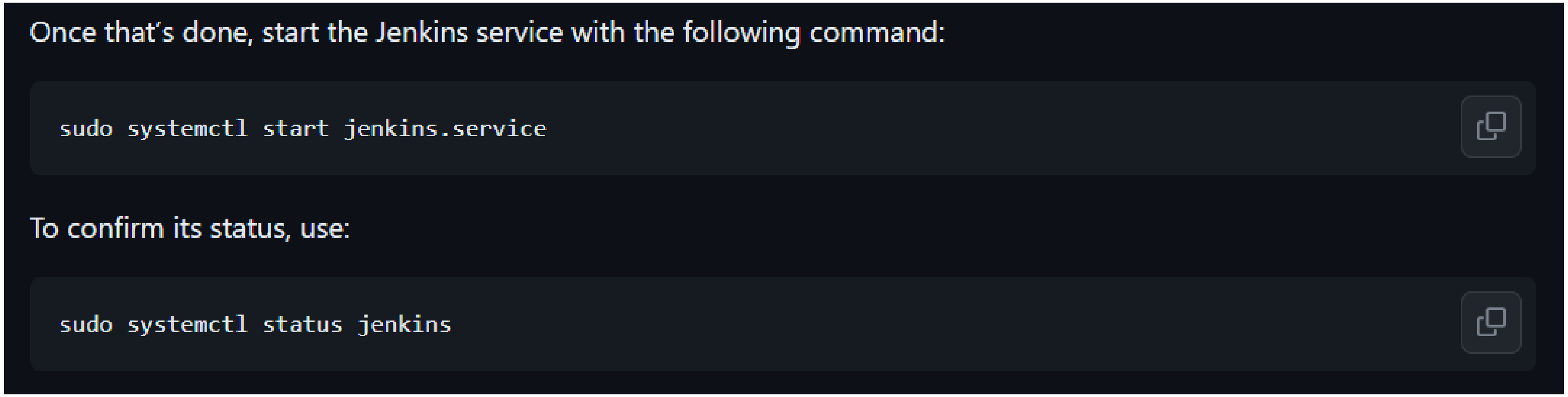
```
sudo apt-get update
```

Then, run the following command to install JDK 11:

```
sudo apt-get install openjdk-11-jdk
```

Now, we will install Jenkins itself. Issue the following four commands in sequence to initiate the installation from the Jenkins repository:

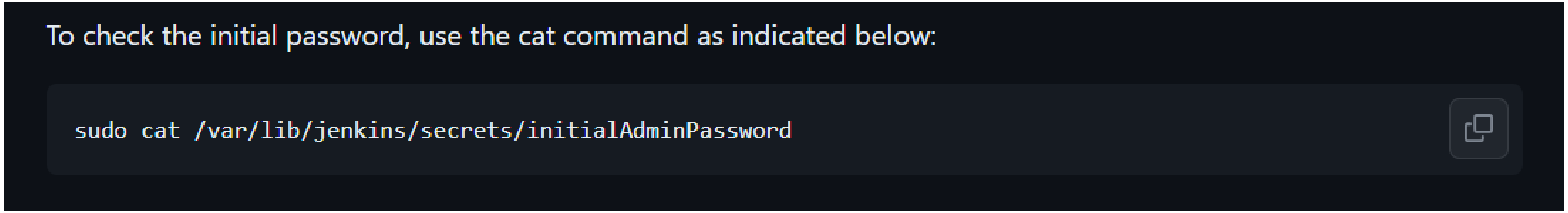
```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt-get update

sudo apt-get install jenkins
```


Once that’s done, start the Jenkins service with the following command:

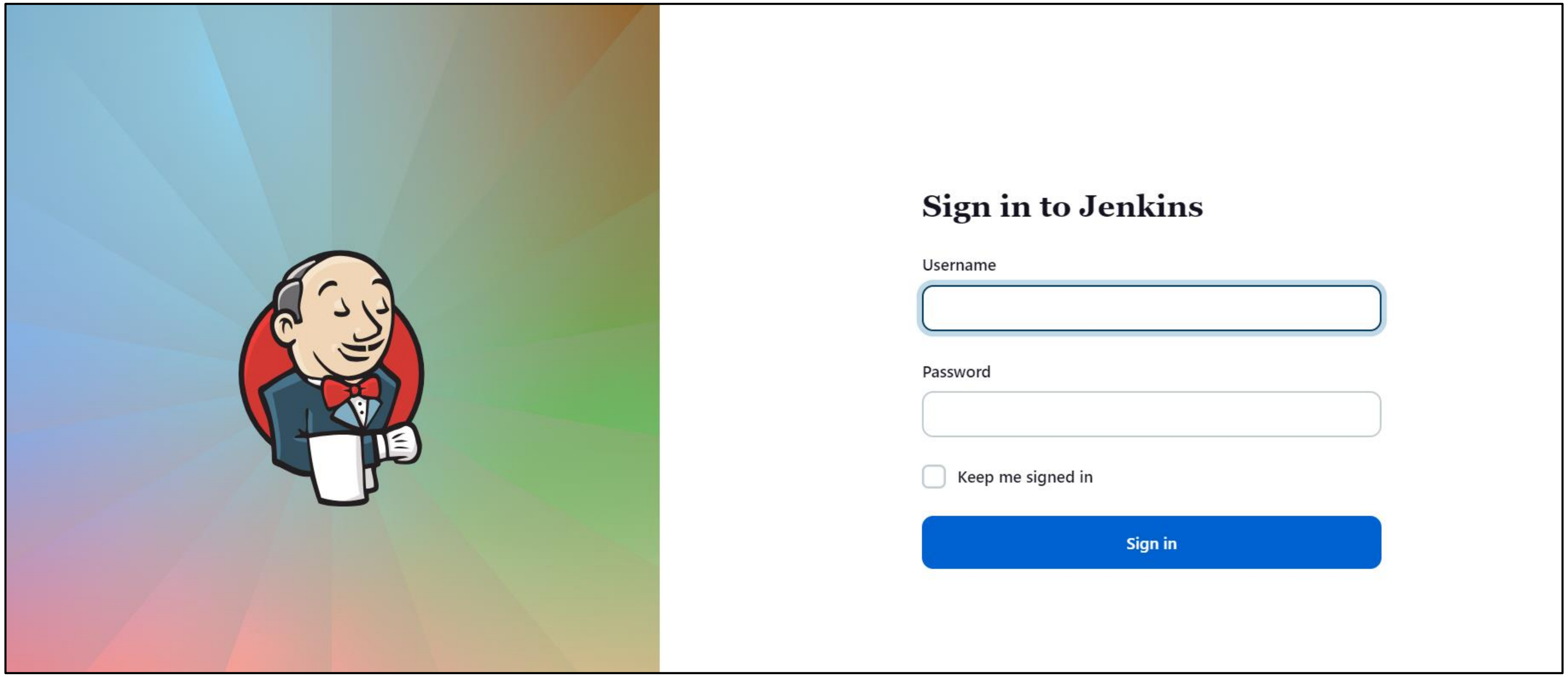
```
sudo systemctl start jenkins.service
```

To confirm its status, use:

```
sudo systemctl status jenkins
```


To check the initial password, use the cat command as indicated below:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```


Sign in to Jenkins

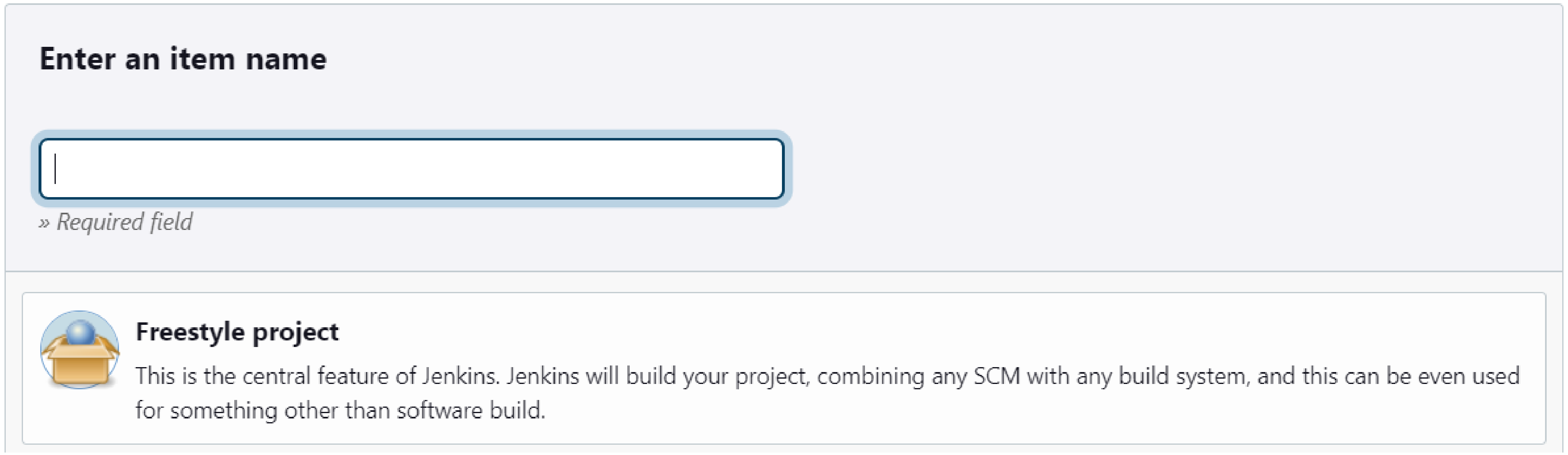
Username

Password

Keep me signed in

Sign in

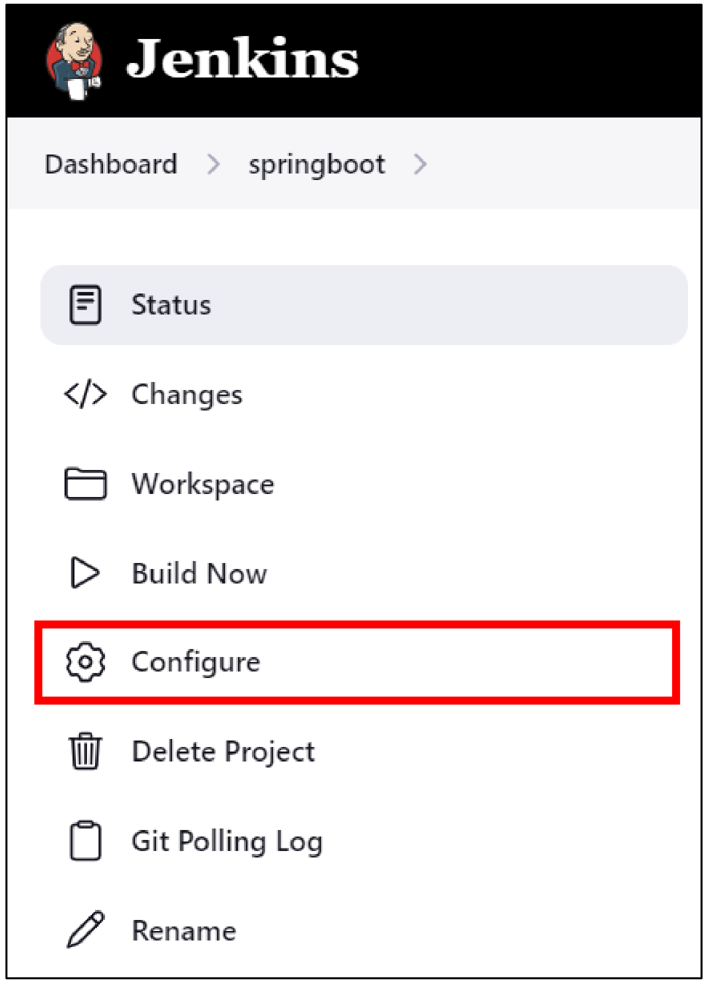
Creating Job

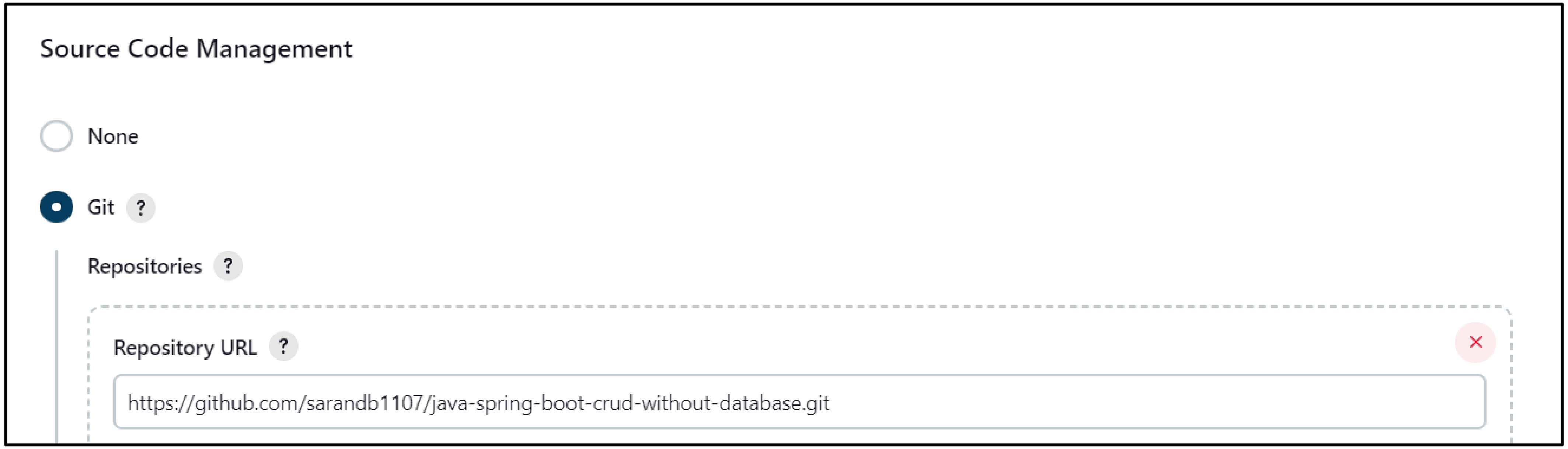

Enter an item name

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.


Jenkins
Dashboard > springboot >
Status
Changes
Workspace
Build Now
Configure
Delete Project
Git Polling Log
Rename


Source Code Management

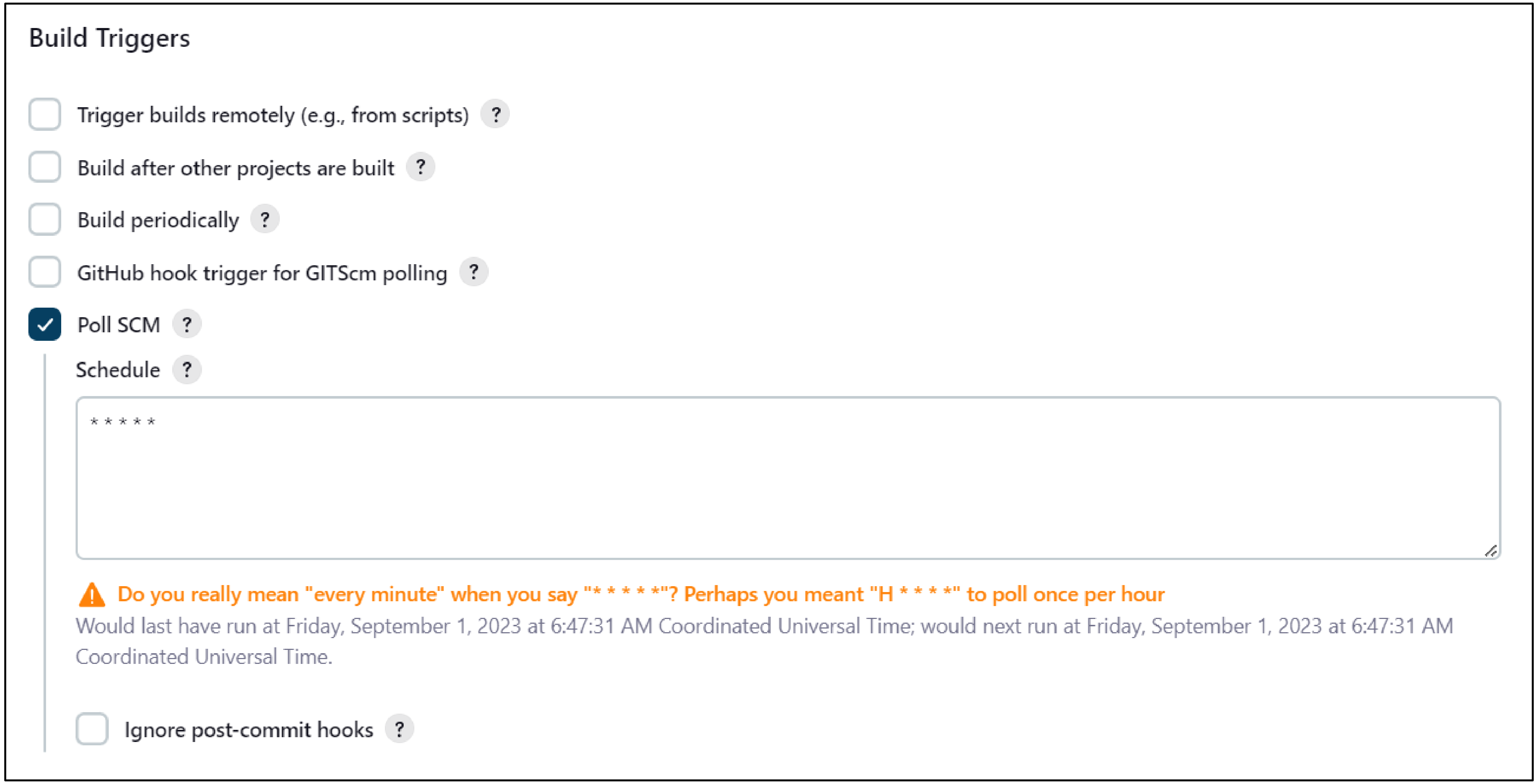
None

[✓ Git]

Repositories

Repository URL

https://github.com/sarandb1107/java-spring-boot-crud-without-database.git



Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts) ?
- [ ] Build after other projects are built ?
- [ ] Build periodically ?
- [ ] GitHub hook trigger for GITScm polling ?
- [✓] Poll SCM ?

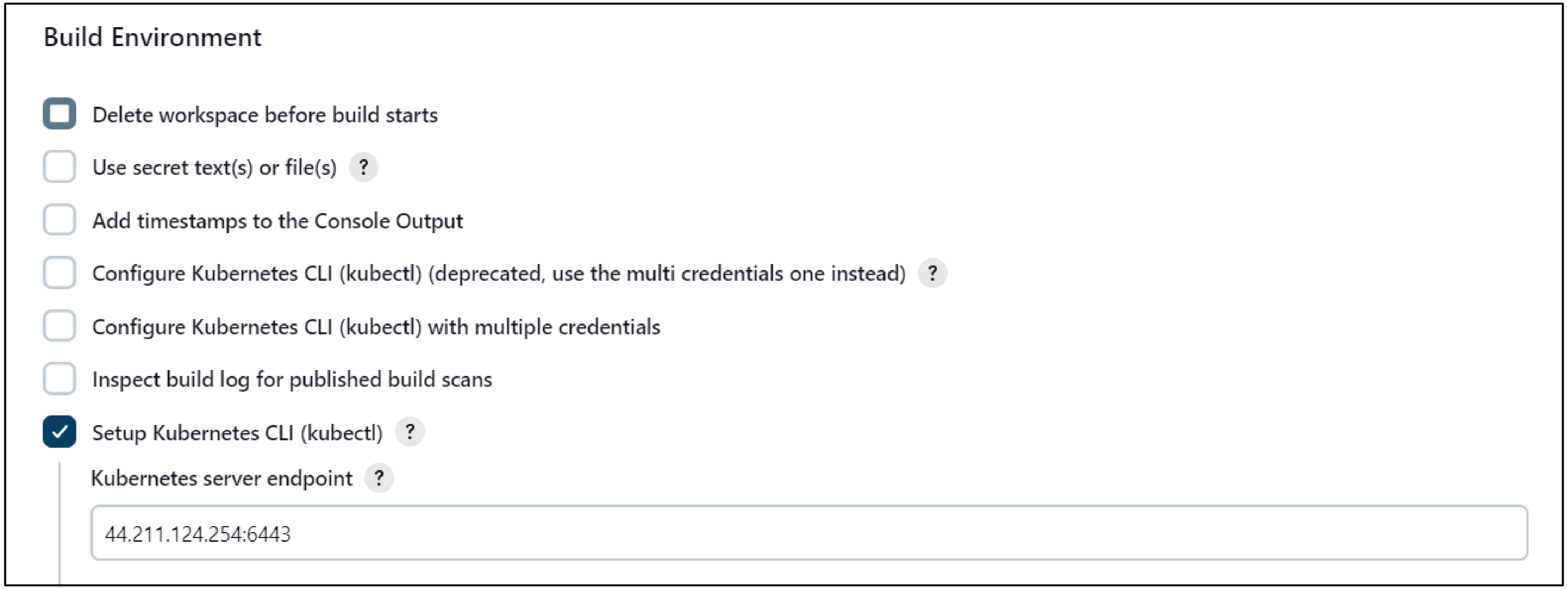
Schedule ?

```
* * * * *
```

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Friday, September 1, 2023 at 6:47:31 AM Coordinated Universal Time; would next run at Friday, September 1, 2023 at 6:47:31 AM Coordinated Universal Time.

- [ ] Ignore post-commit hooks ?

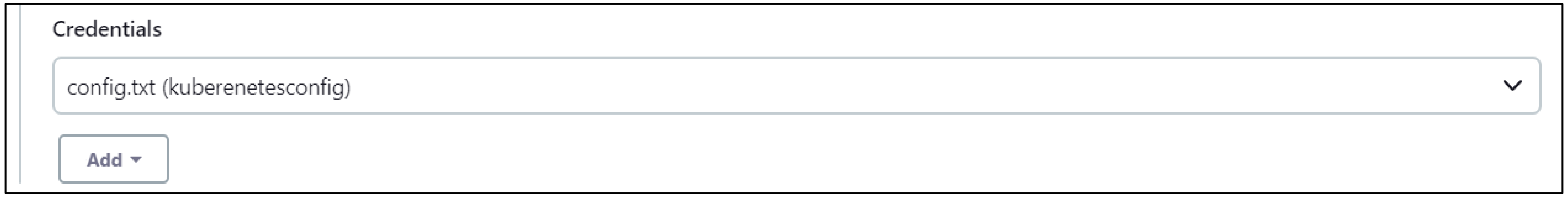


Build Environment

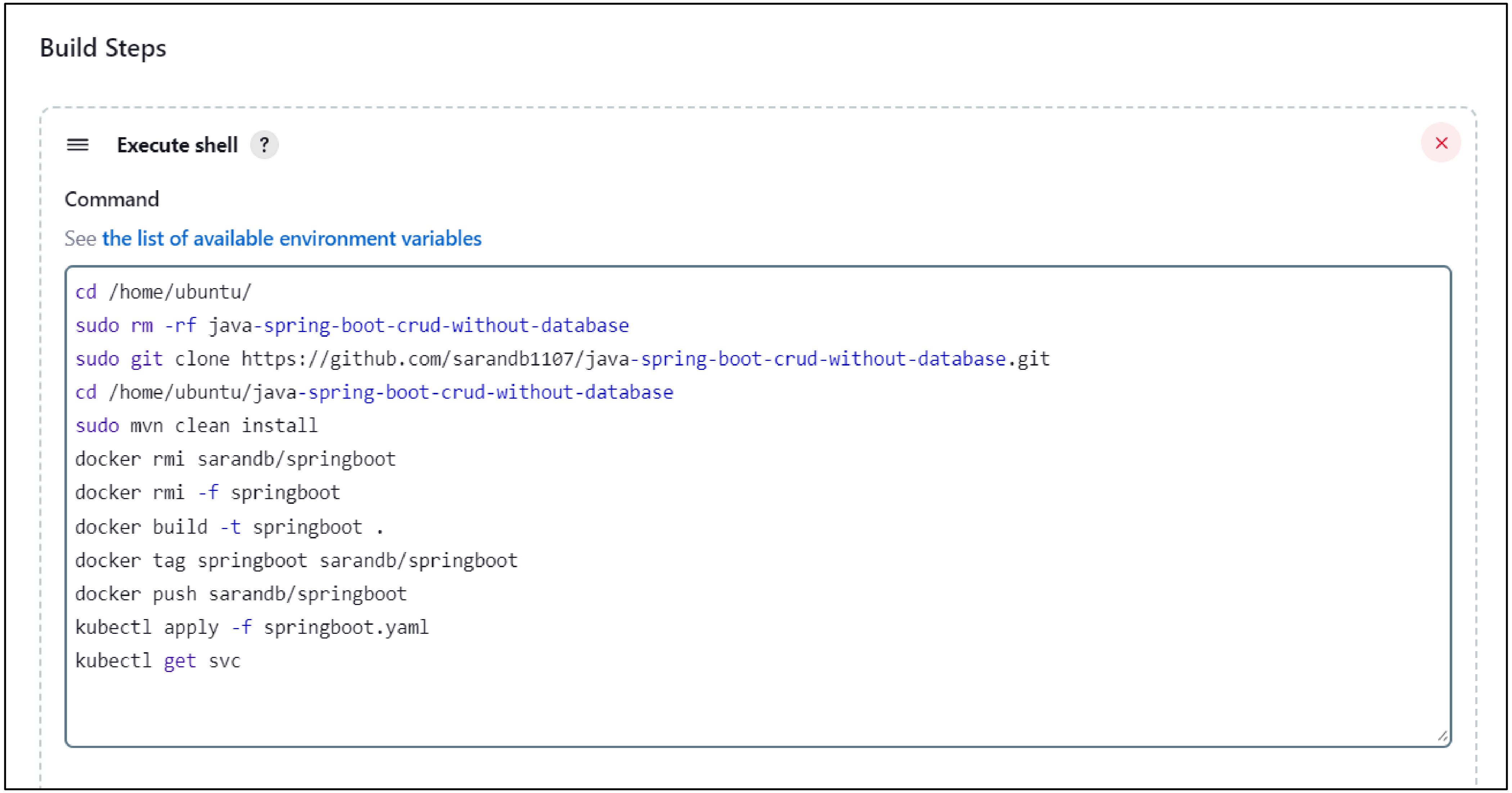
- [ ] Delete workspace before build starts
- [ ] Use secret text(s) or file(s) ?
- [ ] Add timestamps to the Console Output
- [ ] Configure Kubernetes CLI (kubectl) (deprecated, use the multi credentials one instead) ?
- [ ] Configure Kubernetes CLI (kubectl) with multiple credentials
- [ ] Inspect build log for published build scans
- [✓] Setup Kubernetes CLI (kubectl) ?

Kubernetes server endpoint ?

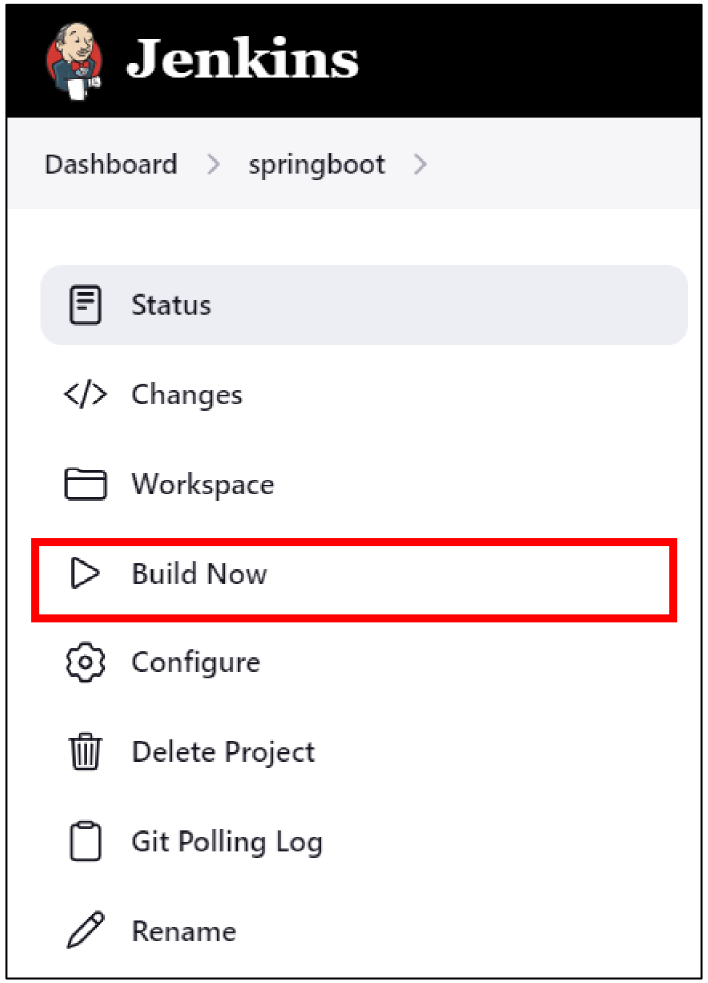
```
44.211.124.254:6443
```

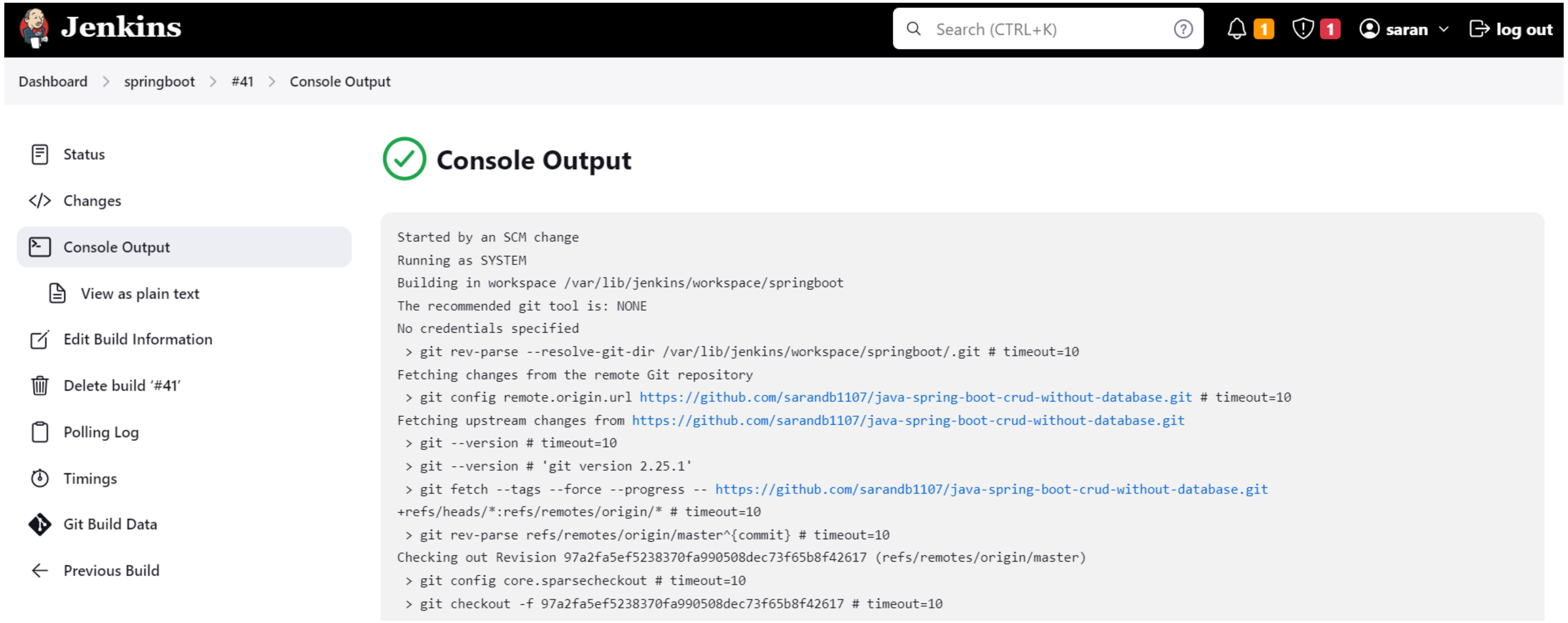


config.txt (kuberenetesconfig)



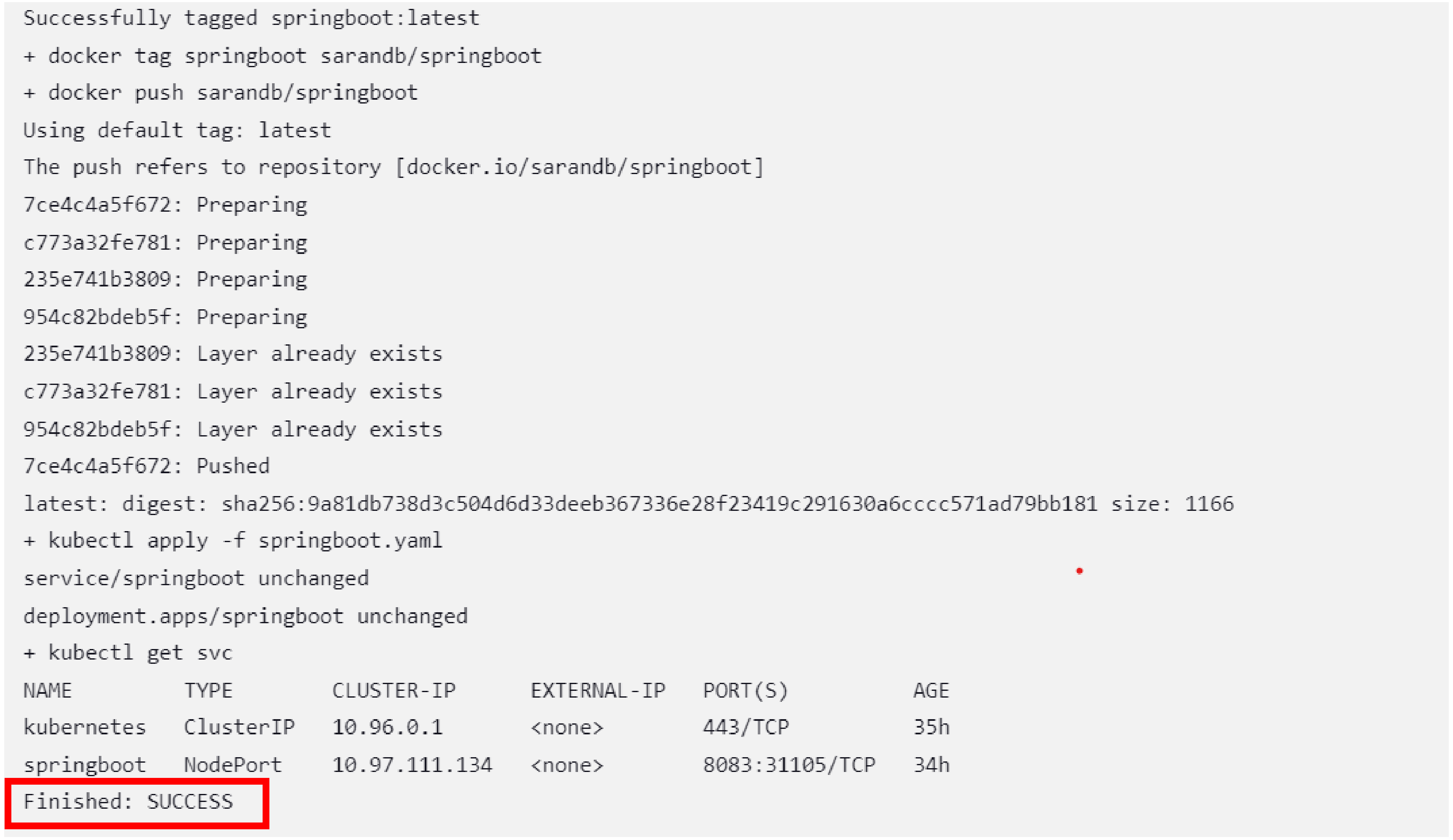
```
cd /home/ubuntu/
sudo rm -rf java-spring-boot-crud-without-database
sudo git clone https://github.com/sarandb1107/java-spring-boot-crud-without-database.git
cd /home/ubuntu/java-spring-boot-crud-without-database
sudo mvn clean install
docker rmi sarandb/springboot
docker rmi -f springboot
docker build -t springboot .
docker tag springboot sarandb/springboot
docker push sarandb/springboot
kubectl apply -f springboot.yaml
kubectl get svc
```


Jenkins
Dashboard > springboot >
Status
Changes
Workspace
Build Now
Configure
Delete Project
Git Polling Log
Rename

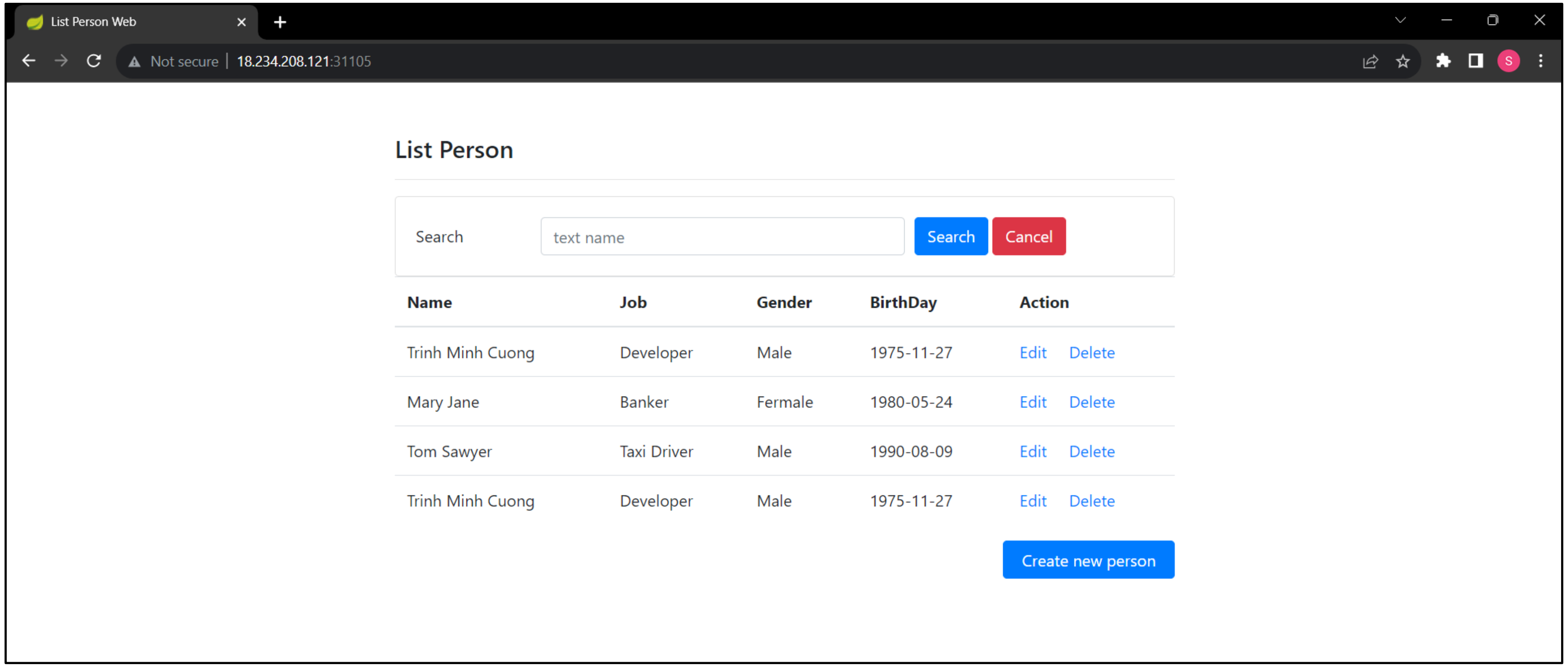


Console Output

```
Started by an SCM change
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/springboot
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/springboot/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/sarandb1107/java-spring-boot-crud-without-database.git # timeout=10
Fetching upstream changes from https://github.com/sarandb1107/java-spring-boot-crud-without-database.git
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
 > git fetch --tags --force --progress -- https://github.com/sarandb1107/java-spring-boot-crud-without-database.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 97a2fa5ef5238370fa990508dec73f65b8f42617 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 97a2fa5ef5238370fa990508dec73f65b8f42617 # timeout=10
```



```
Successfully tagged springboot:latest
+ docker tag springboot sarandb/springboot
+ docker push sarandb/springboot
Using default tag: latest
The push refers to repository [docker.io/sarandb/springboot]
7ce4c4a5f672: Preparing
c773a32fe781: Preparing
235e741b3809: Preparing
954c82bdeb5f: Preparing
235e741b3809: Layer already exists
c773a32fe781: Layer already exists
954c82bdeb5f: Layer already exists
7ce4c4a5f672: Pushed
latest: digest: sha256:9a81db738d3c504d6d33deeb367336e28f23419c291630a6cccc571ad79bb181 size: 1166
+ kubectl apply -f springboot.yaml
service/springboot unchanged
deployment.apps/springboot unchanged
+ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP          35h
springboot   NodePort    10.97.111.134   <none>        8083:31105/TCP   34h
Finished: SUCCESS
```



List Person Web

Not secure | 18.234.208.121:31105

List Person

Search [text name] Search Cancel

Name	Job	Gender	BirthDay	Action
Trinh Minh Cuong	Developer	Male	1975-11-27	Edit Delete
Mary Jane	Banker	Fermale	1980-05-24	Edit Delete
Tom Sawyer	Taxi Driver	Male	1990-08-09	Edit Delete
Trinh Minh Cuong	Developer	Male	1975-11-27	Edit Delete

Create new person