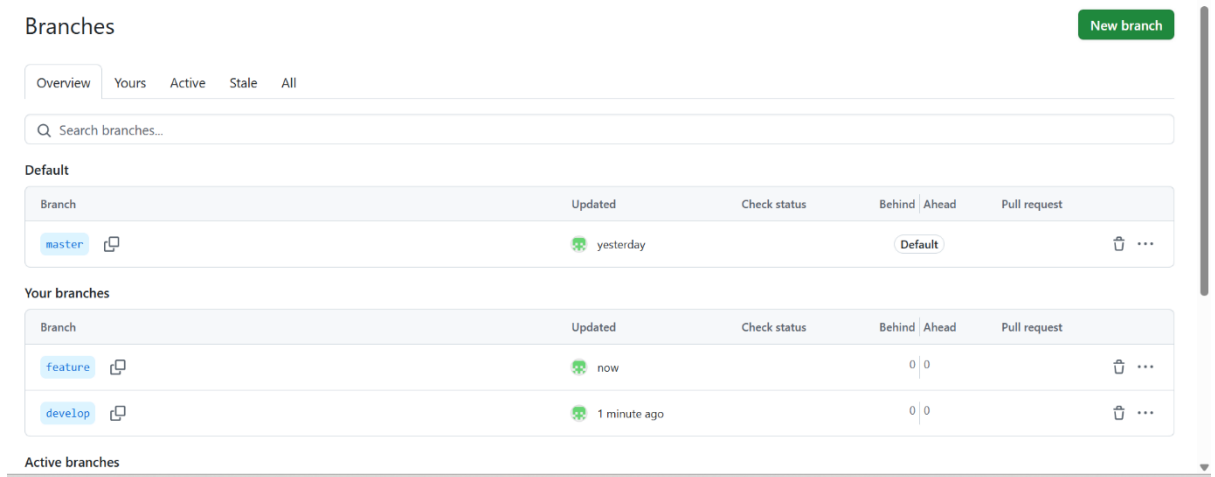
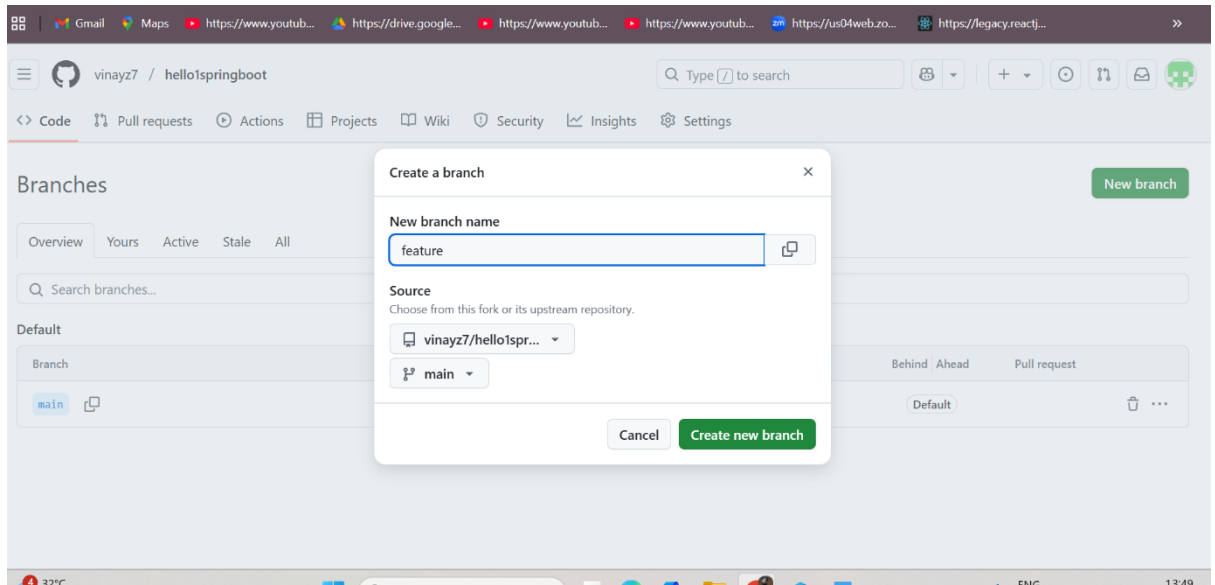


I. Branching Strategy and Execution Rules



II. Webhook Integration

a. Configure a GitHub webhook that triggers Jenkins when:

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code Security

Deploy keys

Secrets and variables

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

`https://18.232.80.97/github-webhook`

Content type *

application/x-www-form-urlencoded

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

☒ Enable SSL verification ☐ Disable (not recommended)

Which events would you like to trigger this webhook?

☐ Just the push event.

☒ Send me everything.

Codespaces

Pages

Security

Code Security

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

SSL verification

By default, we verify SSL certificates when delivering payloads.

☒ Enable SSL verification ☐ Disable (not recommended)

Which events would you like to trigger this webhook?

☐ Just the push event.

☒ Send me everything.

☐ Let me select individual events.

☒ **Active**

We will deliver event details when this hook is triggered.

Add webhook

Okay, that hook was successfully created. We sent a ping payload to test it out! Read more about it at <https://docs.github.com/webhooks/#ping-event>.

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

https://18.232.80.97/github-webho... (all events)

EditDelete

This hook has never been triggered.

III. Building Java Code with Maven

- Compile the application using Maven.**
- Generate .jar file as the build artifact.**

[illegible]

```
[INFO] Results:
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- maven-jar-plugin:3.4.2:jar (default-jar) @ demo ---
[INFO] Building jar: /home/ubuntu/hello!springboot/target/demo-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:3.4.4:repackage (repackage) @ demo ---
[INFO] Replacing main artifact /home/ubuntu/hello!springboot/target/demo-0.0.1-SNAPSHOT.jar with repackaged archive, adding nested dependencies in BOOT-INF/.
[INFO] The original artifact has been renamed to /home/ubuntu/hello!springboot/target/demo-0.0.1-SNAPSHOT.jar.original
[INFO]
[INFO] --- maven-install-plugin:3.1.4:install (default-install) @ demo ---
[INFO] Installing /home/ubuntu/hello!springboot/pom.xml to /root/.m2/repository/com/example/demo/0.0.1-SNAPSHOT/demo-0.0.1-SNAPSHOT.pom
[INFO] Installing /home/ubuntu/hello!springboot/target/demo-0.0.1-SNAPSHOT.jar to /root/.m2/repository/com/example/demo/0.0.1-SNAPSHOT/demo-0.0.1-SNAPSHOT.jar
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 10.008 s
[INFO] Finished at: 2025-04-10T08:47:51Z
[INFO]
root@kmaster:/home/ubuntu/hello!springboot#

Dockerfile HELP.md Jenkinsfile deploy.yaml mvnw mvnw.cmd pom.xml service.yaml src target
root@kmaster:/home/ubuntu/hello!springboot# cd target/
root@kmaster:/home/ubuntu/hello!springboot/target# ls
classes demo-0.0.1-SNAPSHOT.jar.original generated-test-sources maven-status test-classes
demo-0.0.1-SNAPSHOT.jar generated-sources maven-archiver surefire-reports
root@kmaster:/home/ubuntu/hello!springboot/target#
```

- a. Write a Dockerfile with the following specifications:
 - i. Base Image: `openjdk:11-jre-slim`
 - ii. Copies the application JAR from the build output.
 - iii. Exposes port 8080 (the application port).
 - iv. Uses `ENTRYPOINT ["java", "-jar", "/app.jar"]`.

```
GNU nano 7.2 Dockerfile *
# Use an official OpenJDK runtime as a parent image
FROM openjdk:11-jdk-slim

# Set the working directory in the container
WORKDIR /app

# Copy the project's JAR file into the container
COPY target/demo-0.0.1-SNAPSHOT.jar app.jar

# Expose the port the application runs on
EXPOSE 8080

# Run the JAR file
ENTRYPOINT ["java", "-jar", "app.jar"]

Read 14 lines
^G Help      ^O Write Out  ^W Where Is   ^R Cut        ^N Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-B Redo
```

Command : **docker build -t javaimage .**

```
=> extracting sha256:69e15dcd787ba2cfe67f6abf5970ed88a5e019efbb499e499da3ab20b85fcc7 3.7
=> [internal] load build context 0.4
=> transferring context: 20.75MB 0.3
=> [2/3] WORKDIR /app 2.5
=> [3/3] COPY target/demo-0.0.1-SNAPSHOT.jar app.jar 0.2
=> exporting to image 0.2
=> exporting layers 0.2
=> writing image sha256:f3fe57e9d45f817943f6d499b88faf7e55b49b044f0c1fb0792a8ad94aca234b 0.0
=> naming to docker.io/library/javaimage 0.0

root@kmaster:/home/ubuntu/hello1springboot# docker images
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
javaimage            latest       f3fe57e9d45f  13 seconds ago 444MB
<none>               <none>       86f04ddb2bca  8 minutes ago 811MB
vinayz7/java-web-app-cicd latest       a31fa9f1aeac  8 minutes ago 811MB
vinayz7/java-web-app-cicd <none>       eadb2e34c745  14 minutes ago 811MB
helloworld           latest       1f05d93baa49  23 hours ago  136MB
```

```
root@kmaster:/home/ubuntu/hello1springboot# docker run -d -it javaimage bash
97836936e3b62c7295e66e3cbe73705be6742834f807139e88d5f8ddclecc716
root@kmaster:/home/ubuntu/hello1springboot# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
root@kmaster:/home/ubuntu/hello1springboot# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
97836936e3b6  javaimage     "java -jar app.jar b..." 8 seconds ago  Exited (1) 7 seconds ago  trusting_sutherland
e192a2ac049c  javaimage     "java -jar app.jar b..." 29 seconds ago  Exited (1) 28 seconds ago  jovial_snyder
eecd4aecf49   javaimage     "java -jar app.jar"        45 seconds ago  Exited (1) 44 seconds ago  condescending_chaplygin
dc9dfca710d3  helloworld    "python app.py"           23 hours ago   Exited (137) 20 hours ago  flask-container
root@kmaster:/home/ubuntu/hello1springboot#
```

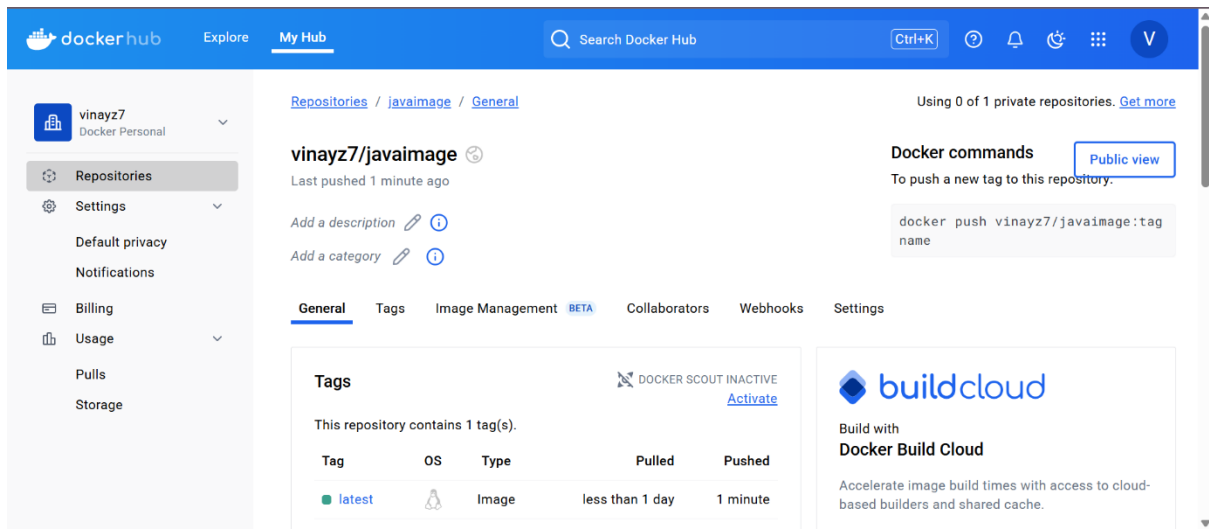
Docker push:

Dochub link: <https://hub.docker.com/repository/docker/vinayz7/javaimage/general>

```
root@kmaster:/home/ubuntu/hello1springboot# docker login
Authenticating with existing credentials... [Username: vinayz7]

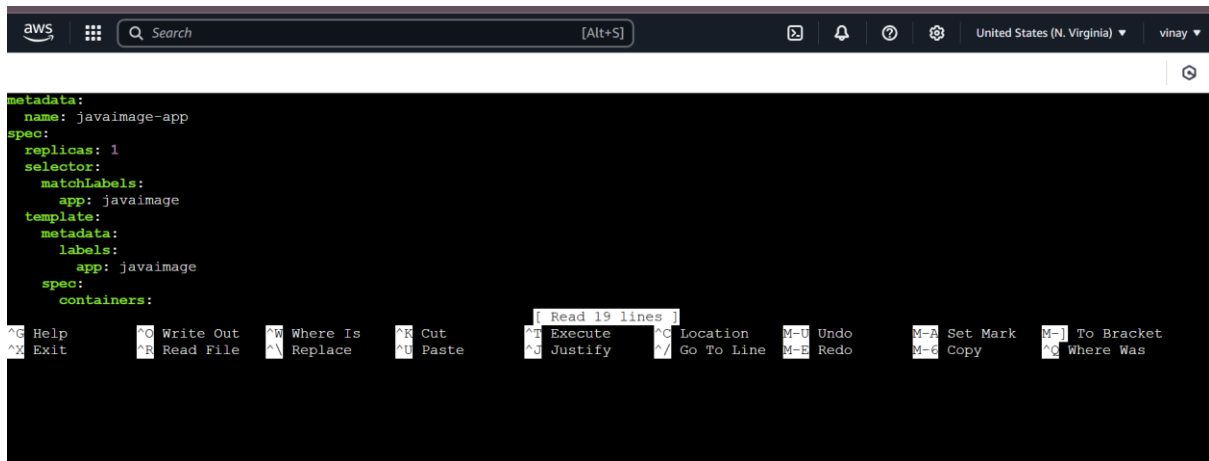
Info → To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
root@kmaster:/home/ubuntu/hello1springboot# docker tag javaimage:latest vinayz7/javaimage:latest
root@kmaster:/home/ubuntu/hello1springboot# docker push vinayz7/javaimage:latest
The push refers to repository [docker.io/vinayz7/javaimage]
ebc3b70d57e9: Pushed
2365b390f092: Pushed
eb6ee5b9581f: Mounted from library/openjdk
e3abdc2e9252: Mounted from library/openjdk
eafe6e032dbd: Mounted from library/openjdk
92a4e8a3140f: Mounted from library/openjdk
latest: digest: sha256:24f5528e9f085c9e06e168f8591d9c25bca43f030ebaa995cc7edc0a14f848a4 size: 1578
root@kmaster:/home/ubuntu/hello1springboot#
```

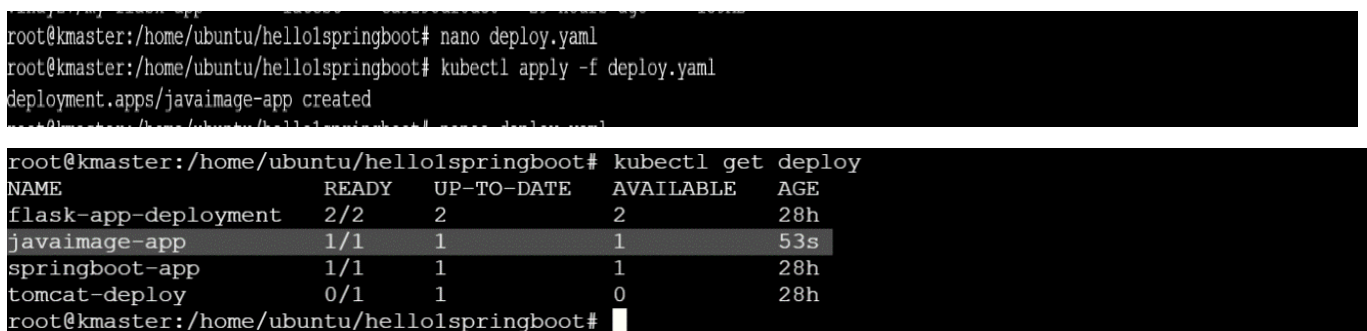


VI. Kubernetes Deployment

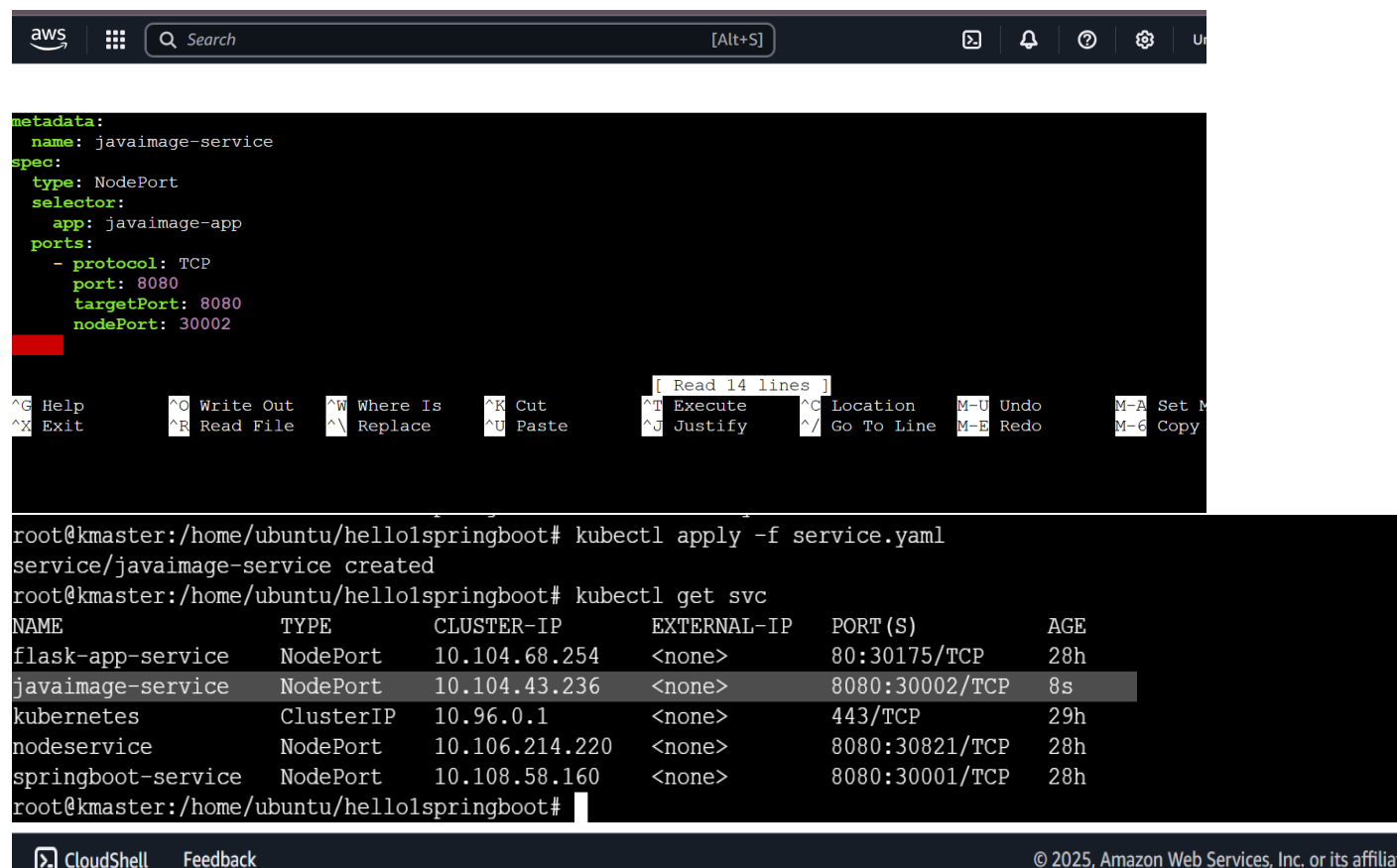
- a. Write Kubernetes manifest files that include:
 - i. deployment.yaml (Defines the app deployment and replicas).
 - ii. service.yaml (Exposes the application internally).
- b. Add Kubernetes apply command to Jenkinsfile to enable deployment.



Kubectl apply -f deploy.yaml



service.yaml (Exposes the application internally).



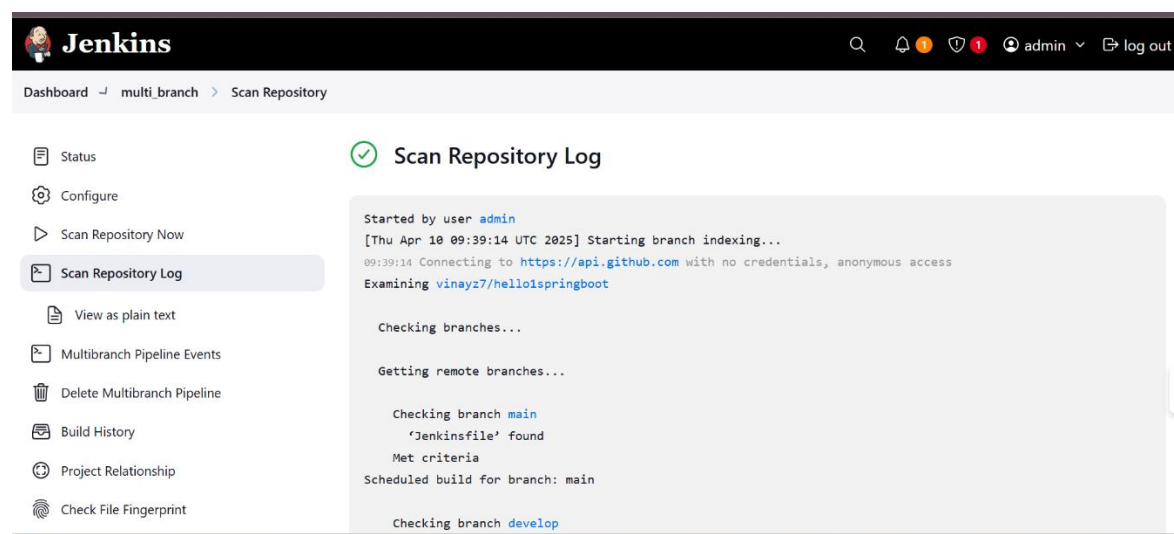
The screenshot shows the AWS CloudShell interface. At the top, there's a search bar and navigation icons. Below, a terminal window displays the content of a file named `service.yaml`. The file defines a `NodePort` service named `javaimage-service` for the `javaimage-app`, exposing port `8080` on the node port `30002`. Below the file content, a menu bar shows various keyboard shortcuts like `^G Help`, `^O Write Out`, etc. The terminal output shows the command `kubectl apply -f service.yaml` being executed, resulting in `service/javaimage-service created`. Then, `kubectl get svc` is run, displaying a table of services. The `javaimage-service` is highlighted in the table. At the bottom, the CloudShell footer shows the `CloudShell` logo, a `Feedback` link, and a copyright notice for Amazon Web Services.

```
metadata:
  name: javaimage-service
spec:
  type: NodePort
  selector:
    app: javaimage-app
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
      nodePort: 30002

root@kmaster:/home/ubuntu/hello1springboot# kubectl apply -f service.yaml
service/javaimage-service created
root@kmaster:/home/ubuntu/hello1springboot# kubectl get svc
NAME                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
flask-app-service   NodePort      10.104.68.254    <none>           80:30175/TCP     28h
javaimage-service    NodePort      10.104.43.236    <none>           8080:30002/TCP   8s
kubernetes           ClusterIP     10.96.0.1        <none>           443/TCP          29h
nodeservice         NodePort      10.106.214.220   <none>           8080:30821/TCP   28h
springboot-service  NodePort      10.108.58.160    <none>           8080:30001/TCP   28h
root@kmaster:/home/ubuntu/hello1springboot#
```

1. Implement a CI/CD pipeline using Jenkins multi-branch pipelines , Docker for containerization, GitHub webhooks and Kubernetes for deployment.

Github repo: <https://github.com/vinayz7/hello1springboot>



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search icon, notification icons, and user information (admin). The main content area shows the 'Scan Repository Log' for a multi-branch pipeline. The log indicates that the scan was started by user 'admin' on Thursday, April 18, 2025, at 09:39:14 UTC. It shows the process of connecting to the GitHub API with anonymous access to examine the repository 'vinayz7/hello1springboot'. The log details the steps: 'Checking branches...', 'Getting remote branches...', 'Checking branch main', finding the 'Jenkinsfile', meeting criteria, and scheduling a build for the 'main' branch. It also shows the start of checking the 'develop' branch.

```
Started by user admin
[Thu Apr 18 09:39:14 UTC 2025] Starting branch indexing...
09:39:14 Connecting to https://api.github.com with no credentials, anonymous access
Examining vinayz7/hello1springboot

Checking branches...

Getting remote branches...

Checking branch main
'Jenkinsfile' found
Met criteria
Scheduled build for branch: main

Checking branch develop
```

Build Queue

No builds in the queue.

Build Executor Status

0/2

multi_branch » main #1

multi_branch » develop #2

multi_branch » feature #2

'Jenkinsfile' found

Met criteria

Changes detected: feature (3165daae672591e9de453e6e573e8815e992836a → 565df184ef6c8a92ef5647b3bc18e73df410a952)

Scheduled build for branch: feature

3 branches were processed

Checking pull-requests...

Getting remote pull requests...

0 pull requests were processed

Finished examining vinayz7/hello1springboot

[Thu Apr 10 09:39:15 UTC 2025] Finished branch indexing. Indexing took 0.72 sec

Evaluating orphaned items in multi_branch

Will remove master

Finished: SUCCESS

18.232.80.97/job/multi_branch/credentials

Jenkins

admin

log out

Dashboard

multi_branch

Status

Configure

Scan Repository Now

Scan Repository Log

Multibranch Pipeline Events

Delete Multibranch Pipeline

Build History

Project Relationship

Check File Fingerprint

GitHub

multi_branch

Branches (3)Pull Requests (0)

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☁	develop	17 min #2	1 hr 26 min #1	10 min ▶
✓	☁	feature	17 min #2	1 hr 26 min #1	11 min ▶
✓	☀	main	17 min #1	N/A	11 min ▶

Iron: S M L

Dashboard

multi_branch

Configuration

Configuration

General

Branch Sources

Build Configuration

Scan Repository Triggers

Orphaned Item Strategy

Appearance

Health metrics

Properties

Branch Sources

GitHub

Credentials ?

vinayz7/*****

+ Add

Repository HTTPS URL

Repository HTTPS URL ?

https://github.com/vinayz7/hello1springboot

Validate

Save

Apply

Jenkins pipe line script :

```
pipeline {
  agent any
  environment {
    DOCKER_IMAGE = 'vinayz7/javaimage:latest'
  }
  stages {
    stage('Clone Repository') {
      steps {
        git url: 'https://github.com/vinayz7/hello1springboot', branch: 'main'
      }
    }
    stage('build mvn for jar') {
      steps {
        sh 'mvn clean install'
      }
    }
    stage('Build Docker Image') {
      steps {
        sh 'docker build -t $DOCKER_IMAGE .'
      }
    }
    stage('Push Docker Image') {
      steps {
        withCredentials([usernamePassword(credentialsId: 'docker-hub-credentials',
usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')]) {

          sh """
            docker login -u $DOCKER_USER -p $DOCKER_PASS
            """


          sh 'docker push $DOCKER_IMAGE'
        }
      }
    }
  }
}
```



```

    }
}
stage('Deploy to Kubernetes') {
    steps {
        sh 'kubectl apply -f deploy.yaml'
    }
}
stage('Expose Service') {
    steps {
        sh 'kubectl apply -f service.yaml'
    }
}
}
}
}

```


hello1springboot
Public

[forked from vilasvarghese/hello1springboot](#)

Pin
Watch 0
Fork 0
Star 0

main
3 Branches
0 Tags

Add file
Code

This branch is **12 commits ahead of** [vilasvarghese/hello1springboot:main](#)
Contribute
Sync fork

Commit	Message	Time
vinayz7	Update service.yaml	565df18 · 2 days ago
	init	5 days ago
	Updated Dockerfile	3 days ago
	init	5 days ago
	Update Jenkinsfile	3 days ago
	Rename deployment.yaml to deploy.yaml	3 days ago
	init	5 days ago
	init	5 days ago

About

No description, website, or topics provided.

Activity

0 stars

0 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

31°C
Search
ENG
15:19

Webhook-after any changes commints in repo it will automatically trigger

