

Repo: <https://github.com/devanarayanantm/hello1springboot>

1) Clone the repo

```
root@master:~# git clone git@github.com:devanarayanantm/hello1springboot.git
Cloning into 'hello1springboot'...
remote: Enumerating objects: 45, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 45 (delta 12), reused 13 (delta 6), pack-reused 20 (from 1)
Receiving objects: 100% (45/45), 13.51 KiB | 576.00 KiB/s, done.
Resolving deltas: 100% (12/12), done.
```

1) Created the docker file

```
root@master:~/hello1springboot# cat Dockerfile
FROM openjdk:17-jdk-slim

WORKDIR /app

COPY target/*.jar app.jar

EXPOSE 8080

CMD java -jar app.jar
```

2) Pushed Dockerfile

```
root@master:~/hello1springboot# nano Dockerfile
root@master:~/hello1springboot# git add .
root@master:~/hello1springboot# git commit -m "Docker file created and added"
[main b99f57d] Docker file created and added
1 file changed, 1 insertion(+)
root@master:~/hello1springboot# git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 312 bytes | 312.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:devanarayanantm/hello1springboot.git
d1f02cf..b99f57d main -> main
```

3) Created feature branch

Nb: Don't know about sonarqube so created a branch greeting and with simple message

```
root@master:~/hello1springboot# git checkout -b feature/greeting
Switched to a new branch 'feature/greeting'
```

4) Create a greeting txt filr

```
root@master:~/hello1springboot# cat greeting.txt
Hi! Welcome to my repo..
```

5) Then push it

```

root@master:~/hello1springboot# git add greeting.txt
root@master:~/hello1springboot# git commit -m "File created in greeting.txt of branch feature/greeting"
[feature/greeting 945bb8c] File created in greeting.txt of branch feature/greeting
1 file changed, 1 insertion(+)
create mode 100644 greeting.txt

```

```

root@master:~/hello1springboot# git push
fatal: The current branch feature/greeting has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feature/greeting

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetUpRemote' in 'git help config'.

root@master:~/hello1springboot# git push -u origin feature/greeting
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 326 bytes | 326.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature/greeting' on GitHub by visiting:
remote:   https://github.com/devanarayanantm/hello1springboot/pull/new/feature/greeting
remote:
To github.com:devanarayanantm/hello1springboot.git
 * [new branch]   feature/greeting -> feature/greeting
branch 'feature/greeting' set up to track 'origin/feature/greeting'.
root@master:~/hello1springboot#

```

Activate Windows
Go to Settings

6) Created deployment.yaml then it pushed it

```

root@master:~/hello1springboot/k8smanifest# k create deploy examspring --image=devanarayanantm/examspring --replicas=3 -o yaml > deployment.yaml
root@master:~/hello1springboot/k8smanifest#
root@master:~/hello1springboot/k8smanifest# nano deployment.yaml

```

Activate Windows

7) Create service and pushed

```

root@master:~/hello1springboot/k8smanifest# k expose deploy examspring --name=examspring --port=8000 --target-port=8080 --dry-run=client -o yaml > s
service.yaml
root@master:~/hello1springboot/k8smanifest#
root@master:~/hello1springboot/k8smanifest#
root@master:~/hello1springboot/k8smanifest# git add service.yaml
root@master:~/hello1springboot/k8smanifest# git commit -m "service.yaml created"
[main ae6c953] service.yaml created
1 file changed, 16 insertions(+)
create mode 100644 k8smanifest/service.yaml
root@master:~/hello1springboot/k8smanifest# git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 504 bytes | 504.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:devanarayanantm/hello1springboot.git

```

Activate Windows
Go to Settings to activate Windows

8) Created Webhook

The screenshot shows the GitHub repository settings for 'devanarayanantm / hello1springboot'. The 'Webhooks' tab is selected in the left sidebar. The main content area shows a list of webhooks with one entry: 'http://54.226.5.210:8080/github-we... (pull_request and push)'. The status is 'Last delivery was successful'. There are 'Edit' and 'Delete' buttons for this webhook. The top navigation bar includes links for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The top right has a search bar and icons for repository management.

9) Created Jenkinsfile

```

root@master:~/hello1springboot# cat Jenkinsfile
pipeline {
  agent any
  stages {
    stage('Build') {
      steps {
        sh 'mvn clean install'
      }
    }
    stage('Docker Image Build and Push') {
      when {
        branch 'main'
      }
      steps {
        withCredentials([usernamePassword(credentialsId: 'dockerhub-cred-id', usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')
      ] {
        sh '''
          docker login -u $DOCKER_USER -p $DOCKER_PASS
          docker build -t $DOCKER_USER/examspring .
          docker push $DOCKER_USER/examspring
        '''
      }
    }
  }
}

```

```

stage('Kubernetes Deploy') {
  when {
    branch 'main'
  }
  steps {
    sh '''
      kubectl delete -f k8smanifest/deployment.yaml --ignore-not-found
      kubectl apply -f k8smanifest/deployment.yaml

      kubectl delete -f k8smanifest/service.yaml --ignore-not-found
      kubectl apply -f k8smanifest/service.yaml
    '''
  }
}

```

Jenkins

The screenshot shows the Jenkins web interface in a browser. The top navigation bar includes the Jenkins logo, a search icon, and a user profile 'admin'. The main content area displays the 'exam' job status. On the left, there is a sidebar with various options: Status, Configure, Scan Repository Now, Scan Repository Log, Multibranch Pipeline Events, Delete Multibranch Pipeline, Build History, Project Relationship, Check File Fingerprint, and GitHub. The main area shows the 'exam' job with a 'Branches (1)' tab selected. Below this, a table lists the build history:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☁	main	21 min #15	54 min #11	18 sec

At the bottom of the interface, there is a search bar with the text 'action' and a list of search filters: Highlight All, Match Case, Match Diacritics, and Whole Words. The Windows taskbar is visible at the very bottom, showing the time as 17:53 on 10-04-2023.

Working successfully and creating deployments and service



```
[Pipeline] stage
[Pipeline] { (Kubernetes Deploy)
[Pipeline] sh
+ kubectl delete -f k8smanifest/deployment.yaml --ignore-not-found
deployment.apps "examspring" deleted
+ kubectl apply -f k8smanifest/deployment.yaml
deployment.apps/examspring created
+ kubectl delete -f k8smanifest/service.yaml --ignore-not-found
service "examspring" deleted
+ kubectl apply -f k8smanifest/service.yaml
service/examspring created
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline

GitHub has been notified of this commit's build result

Finished: SUCCESS
```

Activate Windows
Go to Settings to activate Windows.

Running every pods

```
root@master:~/hello1springboot# k get po
NAME                                READY    STATUS    RESTARTS   AGE
examspring-b974c65c8-k2p6v         1/1      Running   0           9m3s
examspring-b974c65c8-lrkcj         1/1      Running   0           9m3s
examspring-b974c65c8-mbjhj         1/1      Running   0           9m3s
root@master:~/hello1springboot#
```

Service also working

```
root@master:~/hello1springboot# k get svc examspring
NAME      TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
examspring ClusterIP   10.100.162.103 <none>       8000/TCP   14m
root@master:~/hello1springboot# curl 10.100.162.103:8000/hello
Hello Worldroot@master:~/hello1springboot#
```

“After curl it print Hello World”

All service pod deployment and replicaset

```
root@master:~/hello1springboot# k get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/examspring-b974c65c8-k2p6v     1/1      Running   0           7m42s
pod/examspring-b974c65c8-lrkcj     1/1      Running   0           7m42s
pod/examspring-b974c65c8-mbjhj     1/1      Running   0           7m42s

NAME                                TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
service/examspring                 ClusterIP   10.100.162.103 <none>       8000/TCP   7m42s
service/examspringnode              NodePort    10.107.11.29  <none>       8000:30570/TCP 28m
service/kubernetes                  ClusterIP   10.96.0.1     <none>       443/TCP    4h6m

NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/examspring          3/3      3              3            7m43s

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/examspring-b974c65c8 3          3          3        7m42s
root@master:~/hello1springboot#
```

Activate Windows
Go to Settings to activate Windows.