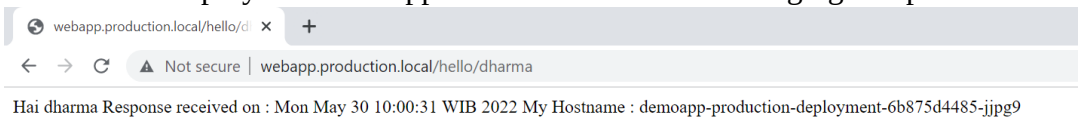
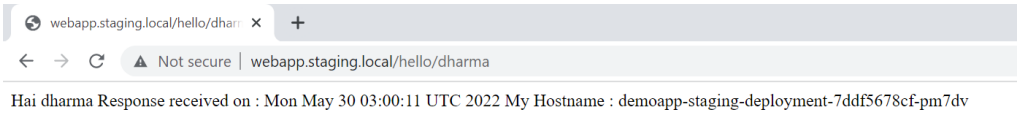
Technical Assesment Devops

A. How to use CI/CD Web with Jenkins

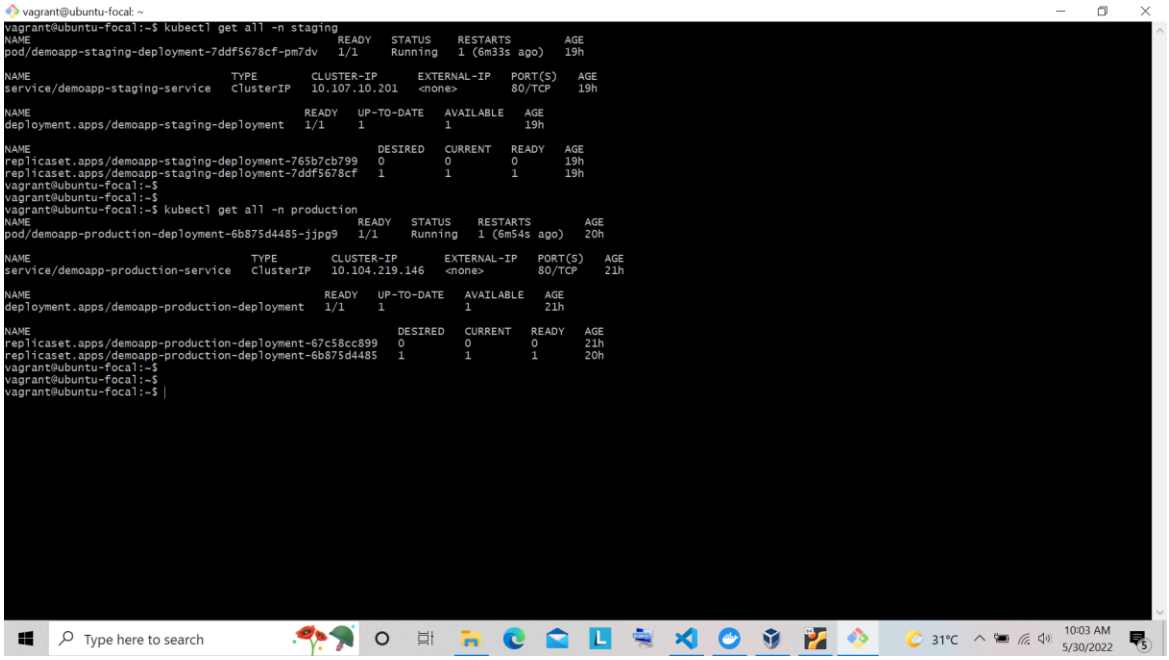
- **Check status before deployment**

  - First check status deployment web application on environment staging and production





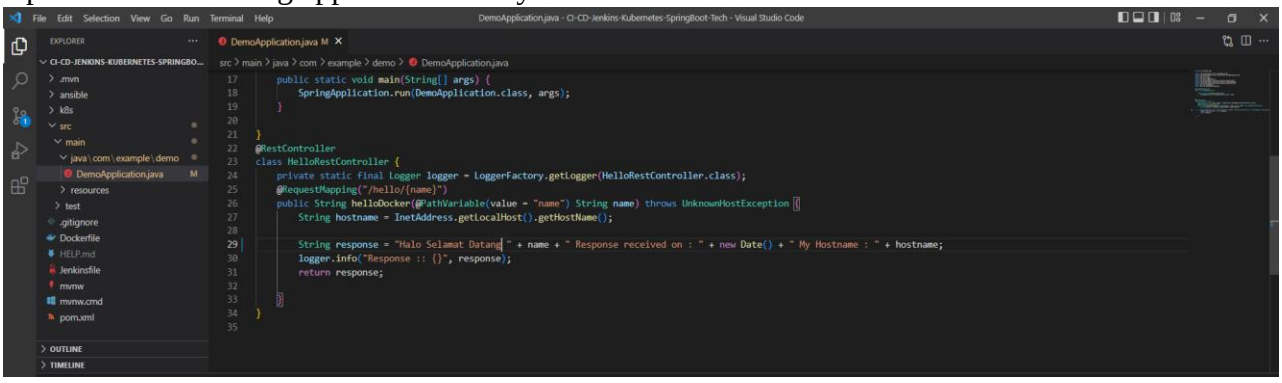
  - Check all service on Kubernetes cluster and check version image deployment file



```
vagrant@ubuntu-focal: ~/ci-cd-jenkins-kubernetes-springboot-tech/k8s
vagrant@ubuntu-focal:~/ci-cd-jenkins-kubernetes-springboot-tech/k8s$ cat prod/deployment.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    kubernetes.io/object: deployment
    app: demoapp
    env: production
  name: demoapp-production-deployment
  namespace: production
spec:
  replicas: 1
  selector:
    matchLabels:
      app: demoapp
      env: production
  template:
    metadata:
      labels:
        app: demoapp
        env: production
    spec:
      imagePullSecrets:
        - name: dockerhubpull
      containers:
      - image: dharmatkj/demoapp:v1.0.0-production
        imagePullPolicy: "Always"
        name: demoapp
        ports:
          - containerPort: 8080vagrant@ubuntu-focal:~/ci-cd-jenkins-kubernetes-springboot-tech/k8s$
```

```
vagrant@ubuntu-focal:~/ci-cd-jenkins-kubernetes-springboot-tech/k8s$ cat staging/deployment.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    kubernetes.io/object: deployment
    app: demoapp
    env: staging
  name: demoapp-staging-deployment
  namespace: staging
spec:
  replicas: 1
  selector:
    matchLabels:
      app: demoapp
      env: staging
  template:
    metadata:
      labels:
        app: demoapp
        env: staging
    spec:
      imagePullSecrets:
        - name: dockerhubpull
      containers:
      - image: dharmatkj/demoapp:v1.0.0-staging
        imagePullPolicy: "Always"
        name: demoapp
        ports:
          - containerPort: 8080vagrant@ubuntu-focal:~/ci-cd-jenkins-kubernetes-springboot-tech/k8s$
```

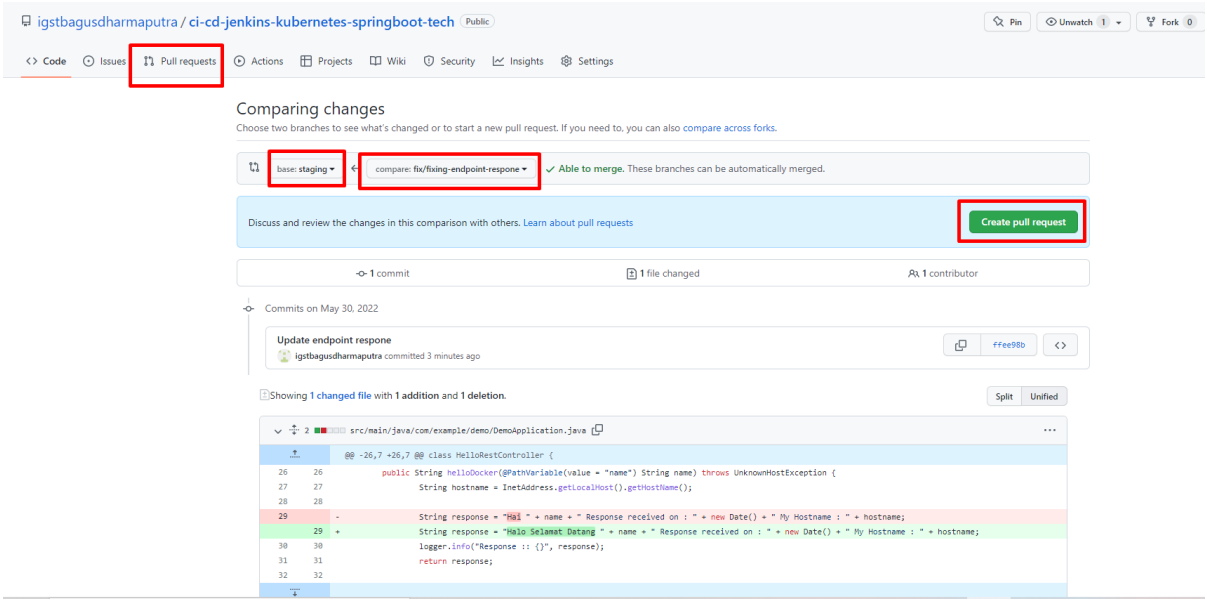
- **Developer create branch to update feature**
  - Create new branch for develop application for example name branch **fix/fixing-endpoint-response**.
  - Update code for fixing application with your code editor.



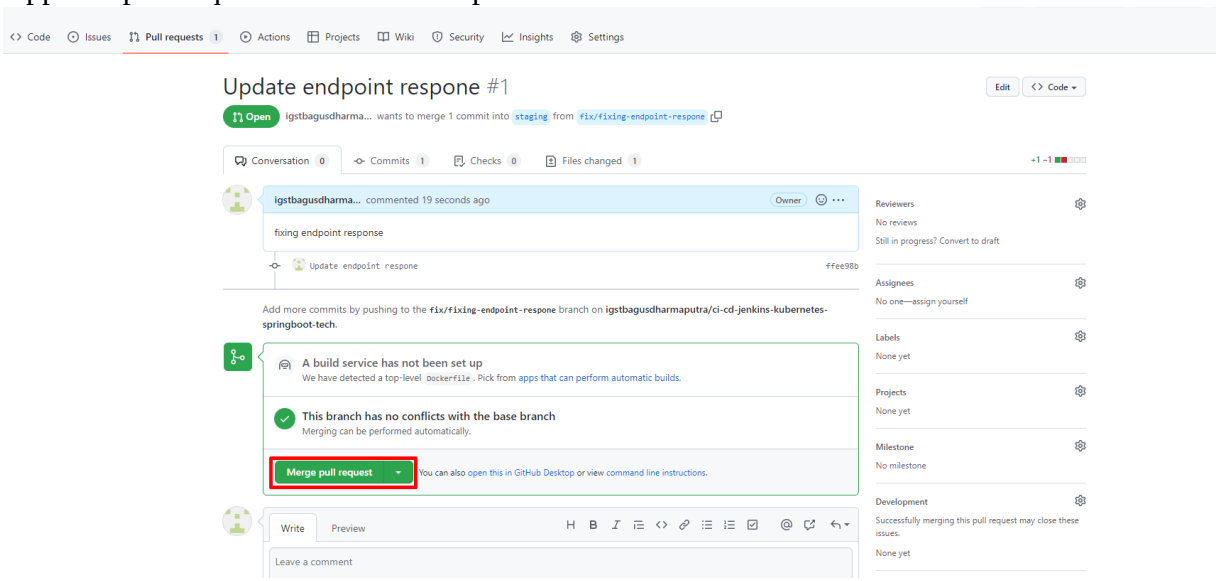
  - Push branch to repository.

```
Klik Digital Sinergi@LAPTOP-6OSCBK75 MINGW64 /d/Test Mandiri/app/CI-CD-Jenkins-Kubernetes-SpringBoot-Tech (fix/fixing-endpoint-respone)
$ git status
On branch fix/fixing-endpoint-respone
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/main/java/com/example/demo/DemoApplication.java

no changes added to commit (use "git add" and/or "git commit -a")

Klik Digital Sinergi@LAPTOP-6OSCBK75 MINGW64 /d/Test Mandiri/app/CI-CD-Jenkins-Kubernetes-SpringBoot-Tech (fix/fixing-endpoint-respone)
$ git add .

Klik Digital Sinergi@LAPTOP-6OSCBK75 MINGW64 /d/Test Mandiri/app/CI-CD-Jenkins-Kubernetes-SpringBoot-Tech (fix/fixing-endpoint-respone)
$ git commit -m "Update endpoint respone"
[fix/fixing-endpoint-respone ffee98b] Update endpoint respone
 1 file changed, 1 insertion(+), 1 deletion(-)

Klik Digital Sinergi@LAPTOP-6OSCBK75 MINGW64 /d/Test Mandiri/app/CI-CD-Jenkins-Kubernetes-SpringBoot-Tech (fix/fixing-endpoint-respone)
$ git push origin fix/fixing-endpoint-respone
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 644 bytes | 214.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'fix/fixing-endpoint-respone' on GitHub by visiting:
remote:      https://github.com/igstbagusdharmaputra/ci-cd-jenkins-kubernetes-springboot-tech/pull/new/fix/fixing-endpoint-respone
remote:
To https://github.com/igstbagusdharmaputra/ci-cd-jenkins-kubernetes-springboot-tech.git
 * [new branch]      fix/fixing-endpoint-respone -> fix/fixing-endpoint-respone

Klik Digital Sinergi@LAPTOP-6OSCBK75 MINGW64 /d/Test Mandiri/app/CI-CD-Jenkins-Kubernetes-SpringBoot-Tech (fix/fixing-endpoint-respone)
$
```

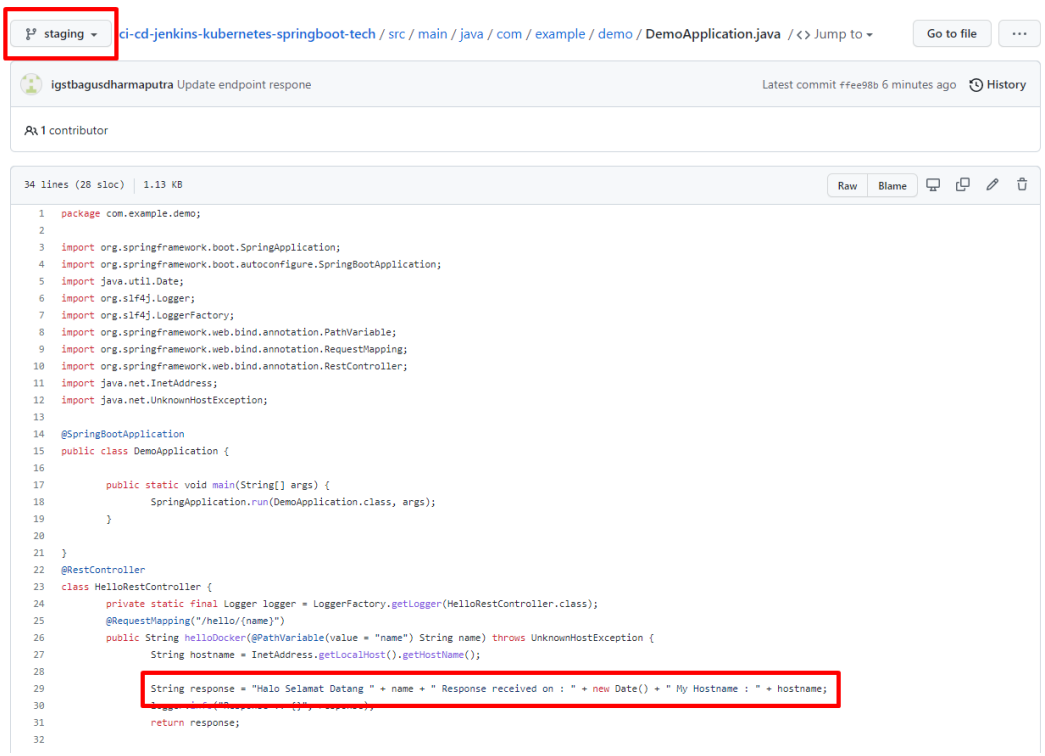
- Create pull request to merge from new source code fixing to branch environment **staging**



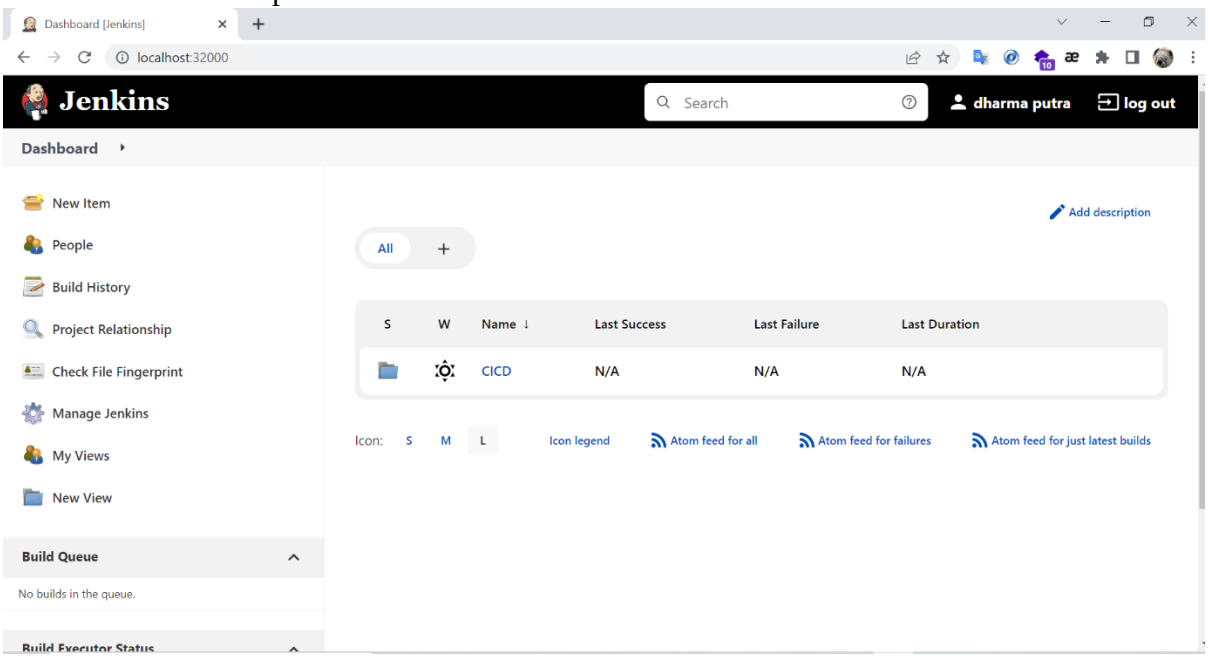
- Approve pull request for the next step.



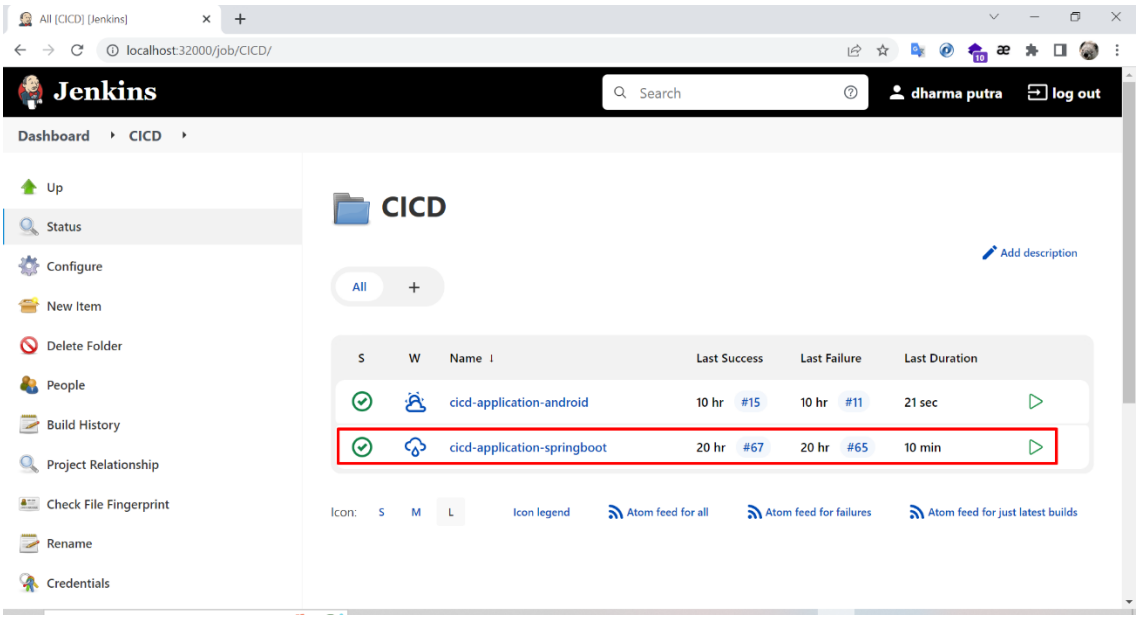
- Check changes source on branch staging.



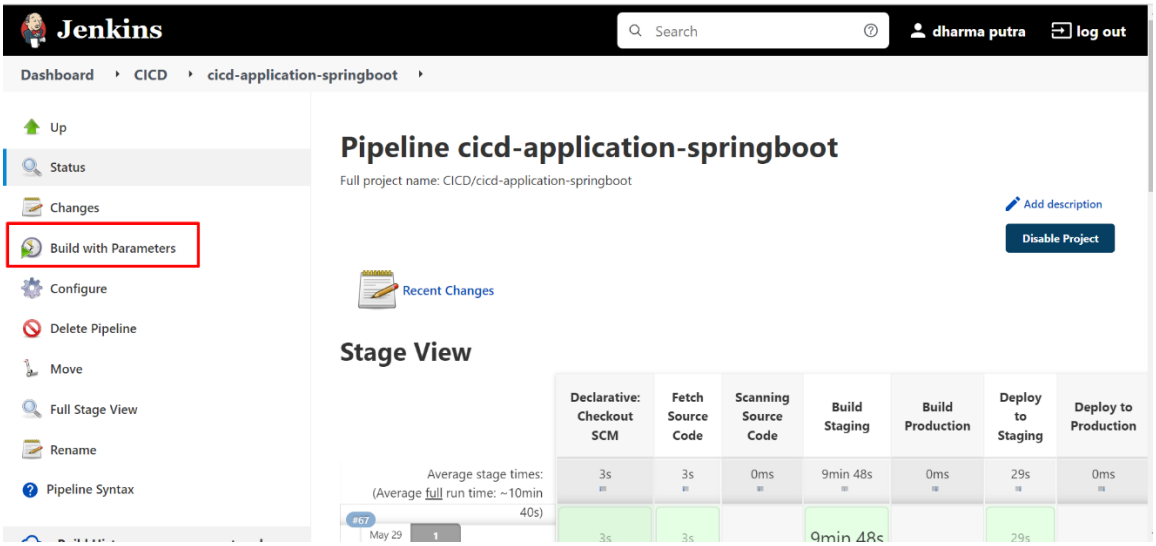
- **Run Jenkins to Deploy Web Applications Staging Environment**
  - First Access Jenkins platform.



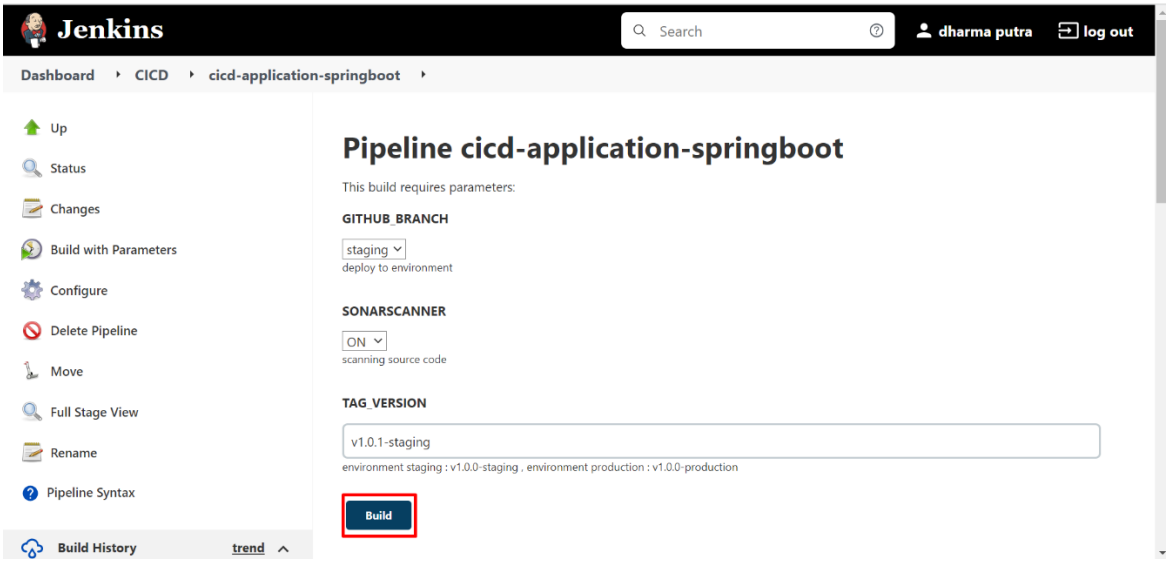
  - Select job Jenkins to use deployment.



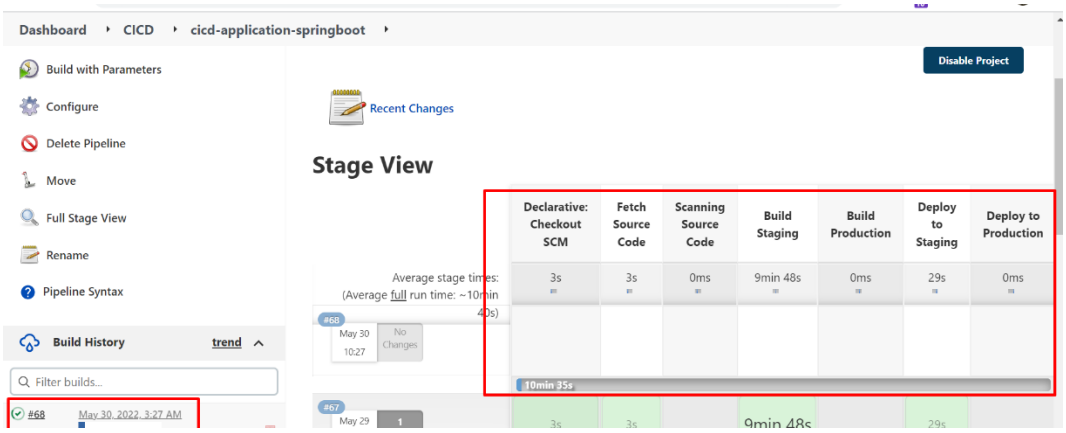
- Choice menu **“Build with parameters”**.



- Please input fields according to the environment and click build to next process.



- Pipeline runs according to cicd flow design.

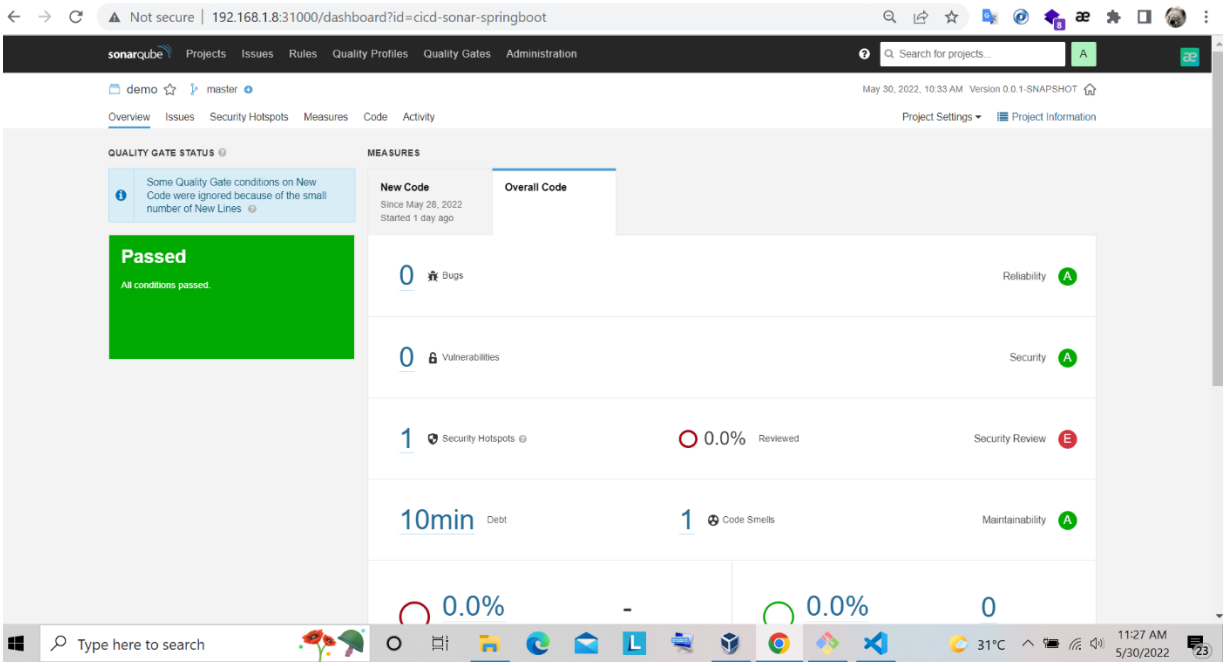


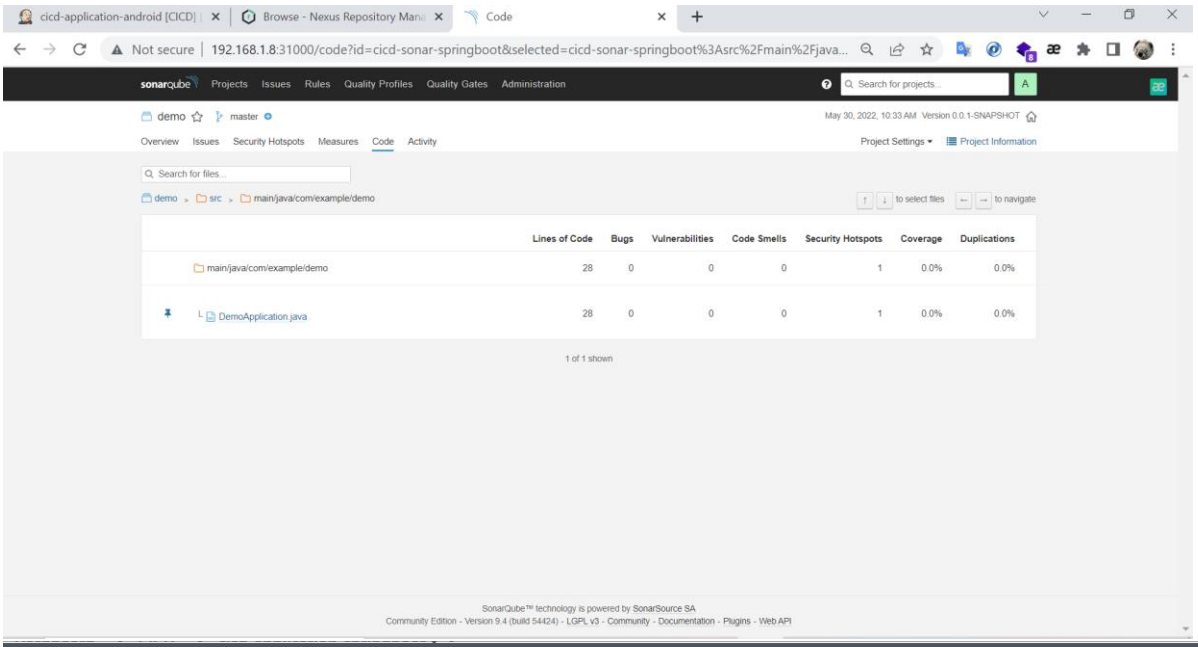
- Check job log in pipeline process (Scanning Source Code, Build, and Deploy), code review can check on sonarqube

Stage Logs (Scanning Source Code)

- Print Message -- Scanning (self time 189ms)
- Checks if running on a Unix-like node (self time 110ms)
- Shell Script -- docker inspect -f . "$JD_TO_RUN" (self time 1s)
- Shell Script -- pwd (self time 1s)
- Shell Script -- mvn -v (self time 1s)
- Shell Script -- mvn clean verify sonar:sonar -Dmaven.test.skip=true -Dsonar.projectKey=${SONAR_PROJECT_KEY} -Dsonar.host.url=${SONAR_URL} -Dsonar.login=${SONAR_TOKEN} (self time 8min 5s)

```
[INFO] Sensor Java CPD Block Indexer (done) | time=51ms
[INFO] SCM Publisher SCM provider for this project is: git
[INFO] SCM Publisher 1 source file to be analyzed
[INFO] SCM Publisher 1/1 source file have been analyzed (done) | time=7979ms
[INFO] CPD Executor Calculating CPD for 1 file
[INFO] CPD Executor CPD calculation finished (done) | time=20ms
[INFO] Analysis report generated in 298ms, dir size=123.0 kB
[INFO] Analysis report compressed in 1149ms, zip size=19.3 kB
[INFO] Analysis report uploaded in 131ms
[INFO] ANALYSIS SUCCESSFUL, you can find the results at: http://192.168.1.8:31000/dashboard?id=****
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] More about the report processing at http://192.168.1.8:31000/api/ce/task?id=AYETBvTWVJ6hguOpLkoT
[INFO] Analysis total time: 30.840 s
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  05:50 min
```





Stage Logs (Build Staging)

Shell Script -- docker login -u dharmatkj -p ${password} docker.io (self time 4s)

Shell Script -- docker build -t dharmatkj/demoapp:v1.0.1-staging . (self time 5min 52s)

```
Progress (1): 2.6/2.6 MB
Progress (1): 2.6/2.6 MB
Progress (1): 2.6 MB

Downloaded from central: https://repo.maven.apache.org/maven2/com/google/guava/guava/28.2-android/guava-28.2-android.jar (2.6 MB at 331 kB/s)
#12 246.1 [INFO] Replacing main artifact with repackaged archive
#12 246.1 [INFO] ------------------------------------------------------------------------
#12 246.1 [INFO] BUILD SUCCESS
#12 246.1 [INFO] ------------------------------------------------------------------------
#12 246.2 [INFO] Total time:  04:02 min
#12 246.2 [INFO] Finished at: 2022-05-30T03:40:40Z
#12 246.2 [INFO] ------------------------------------------------------------------------
#12 DONE 246.3s

#7 [stage-1 1/2] FROM docker.io/library/openjdk:11-jdk-slim-buster@sha256:b4089a6ece62068b2a5111bbb3e98f76498e7a0d813405557d6795357dfa01a8
#7 sha256:ac19e05dc84348f7867cec9cc6a314ccf96b11a594153e6a7050a49986418b01
#7 CACHED

#13 [stage-1 2/2] COPY --from=build /app/target/*.jar app.jar
#13 sha256:8956ea5c8ff41c746487cf62482714c41cadd23e9771dc5c8104df3e29ea51fc
```

Shell Script -- docker push dharmatkj/demoapp:v1.0.1-staging (self time 1min 50s)

Shell Script -- docker rmi dharmatkj/demoapp:v1.0.1-staging (self time 757ms)

Stage Logs (Deploy to Staging)

Print Message -- Deploy to staging (self time 92ms)

Checks if running on a Unix-like node (self time 119ms)

Shell Script -- docker inspect -f . "$JD_TO_RUN" (self time 988ms)

Shell Script -- cp ${myprivatekey} /root/id_rsa && chmod 400 /root/id_rsa (self time 1s)

Shell Script -- sed -i "s/{(VERSION)}/$TAG_VERSION/g" ./k8s/staging/deployment.yml (self time 1s)

Shell Script -- ansible-playbook -i ansible/hosts ansible/playbook-staging.yml --extra-vars "SOURCE_PATH=${WORKSPACE}/k8s/staging" (self time 19s)

```
+ ansible-playbook -i ansible/hosts ansible/playbook-staging.yml --extra-vars 'SOURCE_PATH=/var/jenkins_home/workspace/CICD/cicd-application-springboot/k8s/staging'

PLAY [vagrant] *****************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.1.99]

TASK [Copy Deployment File] ****************************************************
changed: [192.168.1.99]

TASK [Apply Kubectl] ***********************************************************
changed: [192.168.1.99]

PLAY RECAP *********************************************************************
192.168.1.99               : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

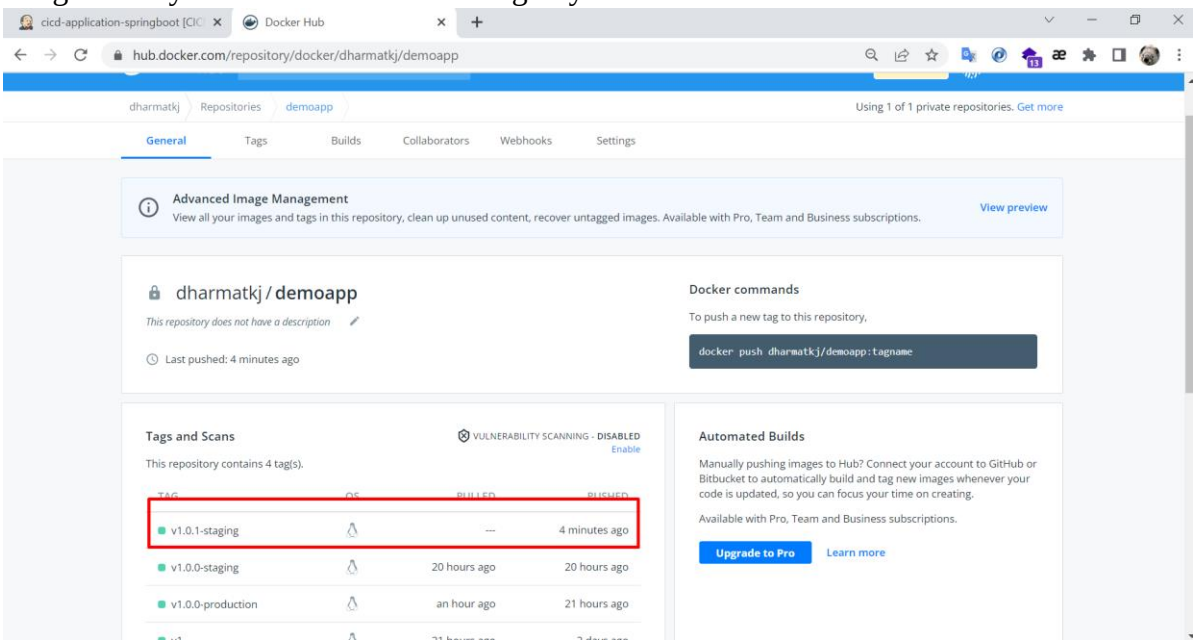
- Check status update service application on Kubernetes after deploy

```
vagrant@ubuntu-focal:~/ci-cd-jenkins-kubernetes-springboot-tech/k8s$ kubectl get all -n staging
NAME                                             READY   STATUS    RESTARTS   AGE
pod/demoapp-staging-deployment-6944697df8-g8prc   1/1     Running   0          85s

NAME                               TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
service/demoapp-staging-service    ClusterIP   10.107.10.201   <none>        80/TCP    20h

NAME                                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/demoapp-staging-deployment   1/1     1            1           20h

NAME                                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/demoapp-staging-deployment-6944697df8   1         1         1       85s
replicaset.apps/demoapp-staging-deployment-765b7cb799   0         0         0       20h
replicaset.apps/demoapp-staging-deployment-7ddf5678cf   0         0         0       20h
vagrant@ubuntu-focal:~/ci-cd-jenkins-kubernetes-springboot-tech/k8s$
```

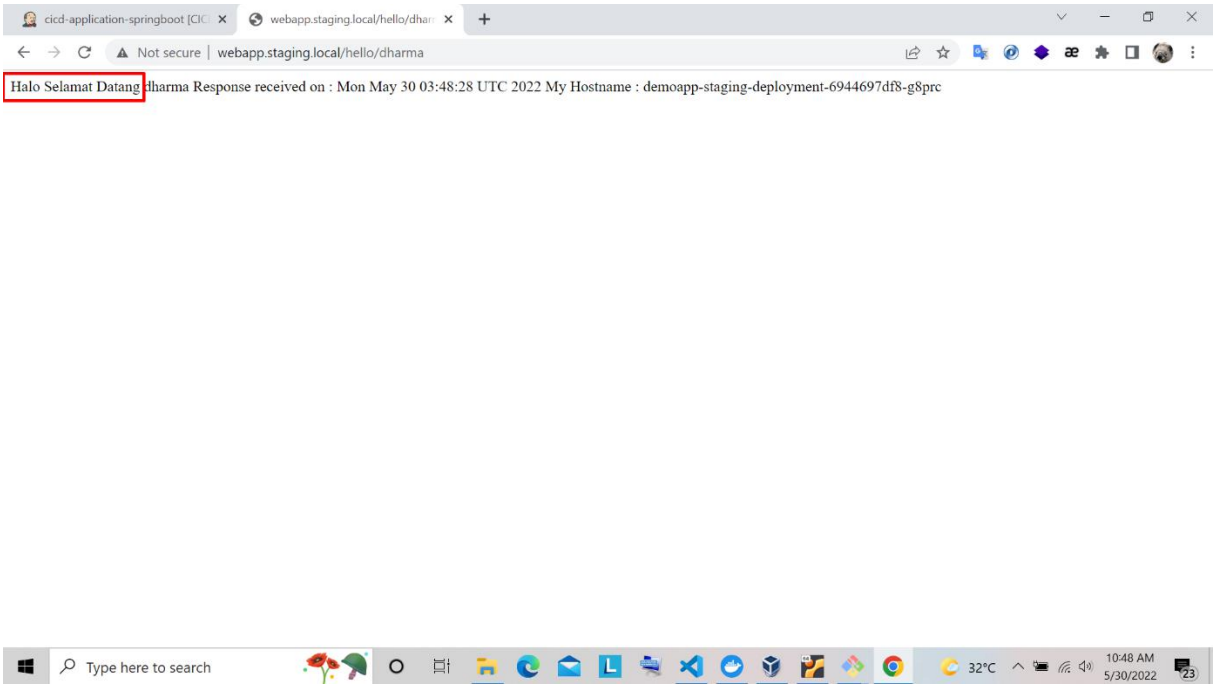
- Check deployment file application to make sure tag version image updated

```
vagrant@ubuntu-focal:~/ci-cd-jenkins-kubernetes-springboot-tech/k8s$ cat staging/deployment.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    kubernetes.io/object: deployment
    app: demoapp
    env: staging
  name: demoapp-staging-deployment
  namespace: staging
spec:
  replicas: 1
  selector:
    matchLabels:
      app: demoapp
      env: staging
  template:
    metadata:
      labels:
        app: demoapp
        env: staging
    spec:
      imagePullSecrets:
        - name: dockerhubpull
      containers:
      - image: dharmatkj/demoapp:v1.0.1-staging
        imagePullPolicy: "Always"
        name: demoapp
        ports:
          - containerPort: 8080vagrant@ubuntu-focal:~/ci-cd-jenkins-kubernetes-springboot-tech/k8s$
```

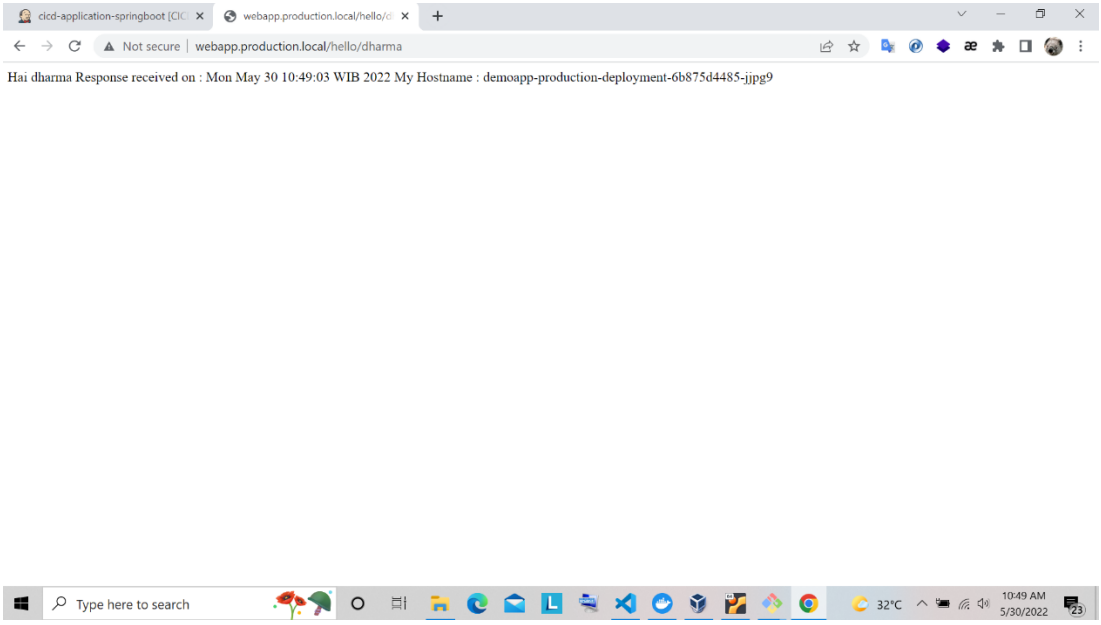
- Image already to store on docker hub registry



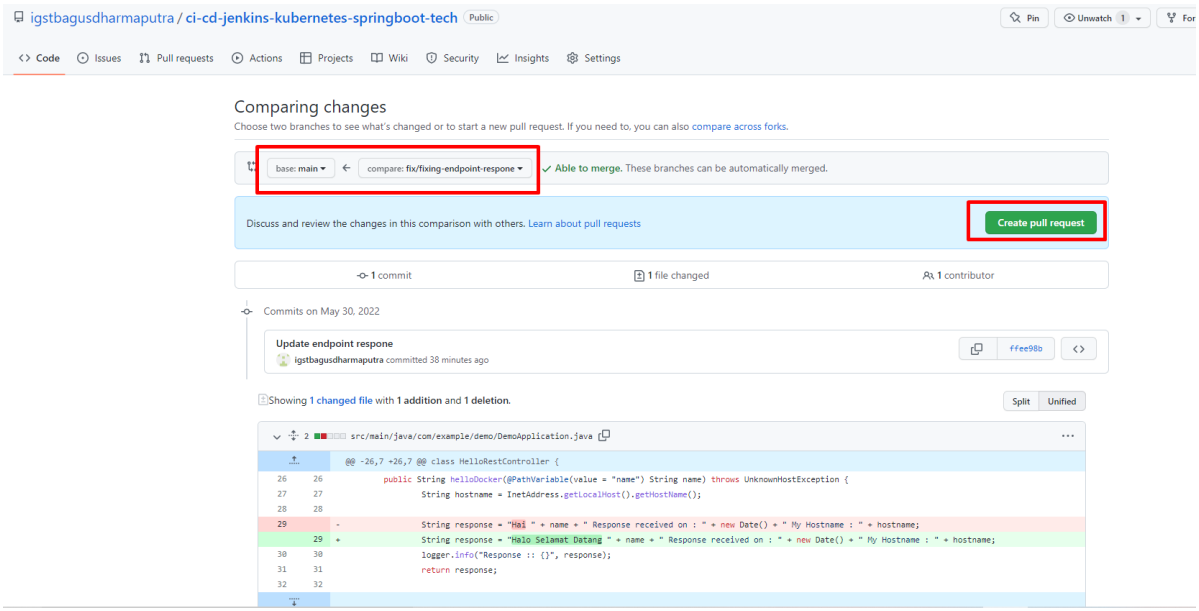
- Try access endpoint web application after deploy.



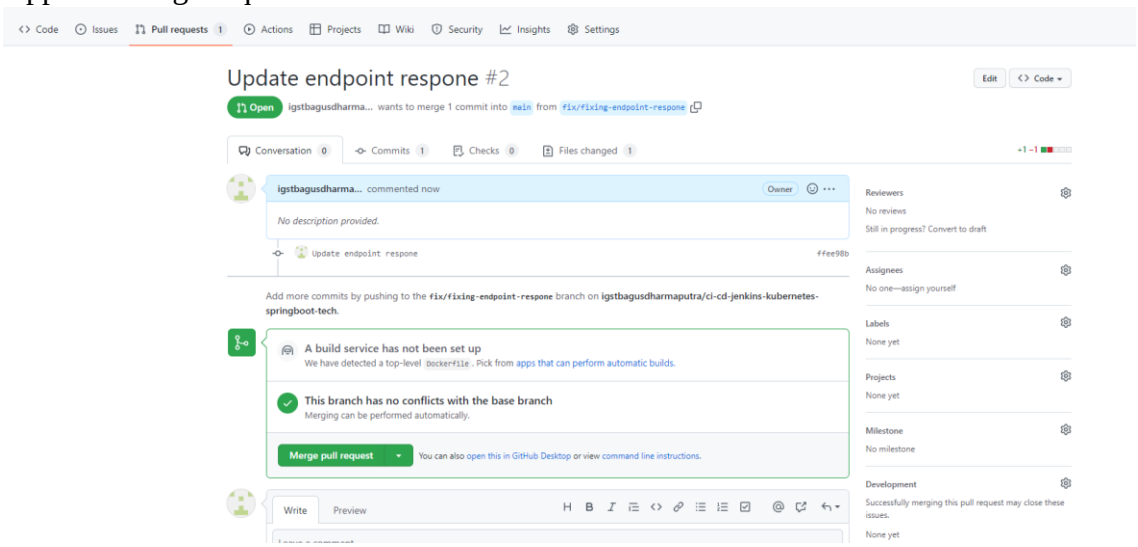
- **Run Jenkins to Deploy Web Applications Production Environment**
  - After staging already update, next to deploy environment production. Check endpoint url make sure still not yet update.



  - Create merge request with branch after test on staging environment.



- Approve merge request.



- Check file changes on branch **main** for environment production.

main ci-cd-jenkins-kubernetes-springboot-tech / src / main / java / com / example / demo / DemoApplication.java Jump to Go to file

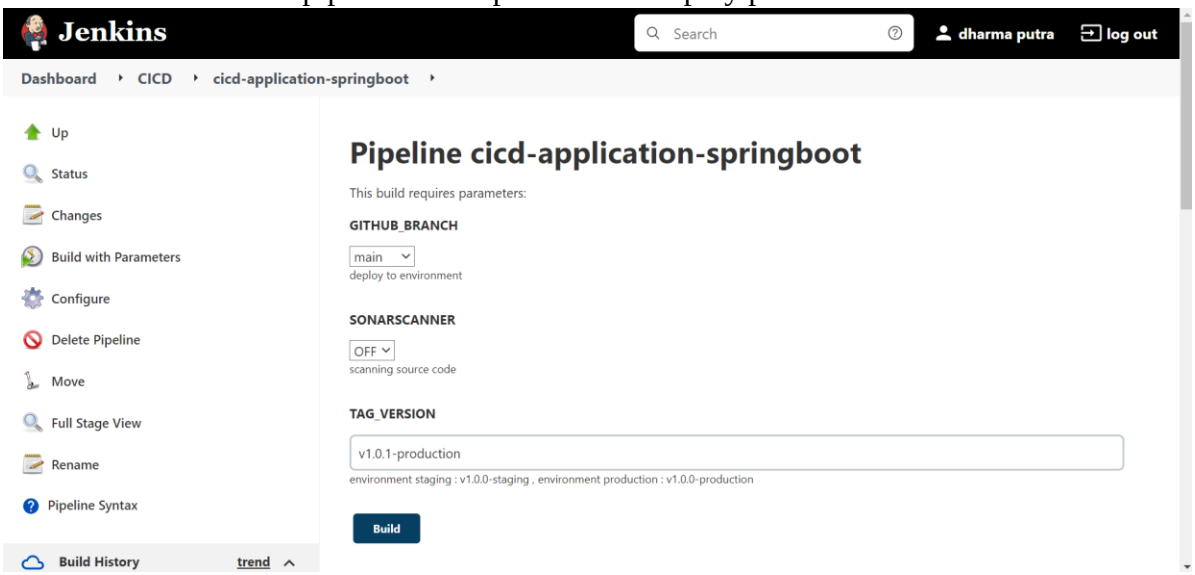
igstbagusdharmaputra Update endpoint respone — Latest commit ffee98b 39 minutes ago History

1 contributor

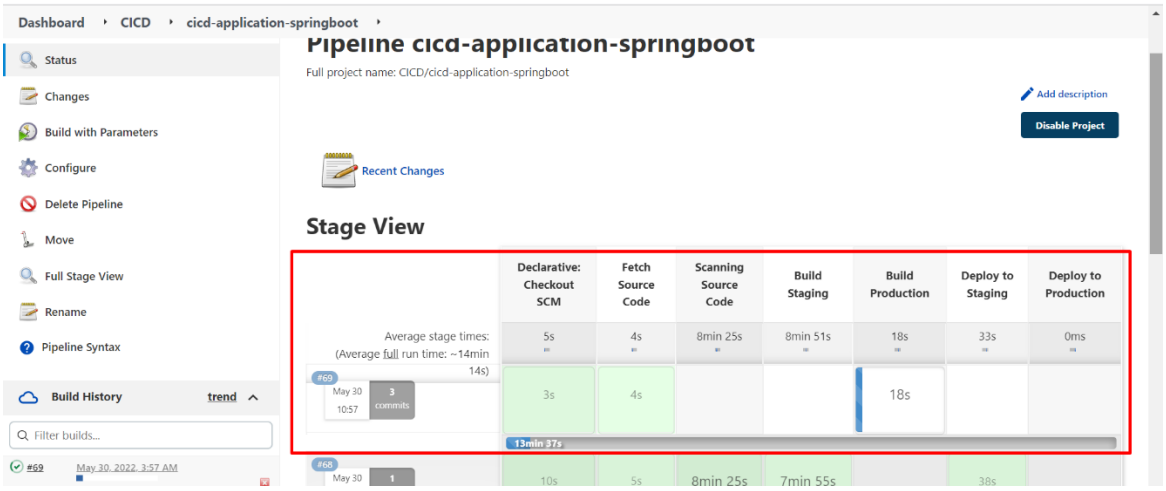
34 lines (28 sloc) 1.13 KB Raw Blame

```
1  package com.example.demo;
2
3  import org.springframework.boot.SpringApplication;
4  import org.springframework.boot.autoconfigure.SpringBootApplication;
5  import java.util.Date;
6  import org.slf4j.Logger;
7  import org.slf4j.LoggerFactory;
8  import org.springframework.web.bind.annotation.PathVariable;
9  import org.springframework.web.bind.annotation.RequestMapping;
10 import org.springframework.web.bind.annotation.RestController;
11 import java.net.InetAddress;
12 import java.net.UnknownHostException;
13
14 @SpringBootApplication
15 public class DemoApplication {
16
17         public static void main(String[] args) {
18                 SpringApplication.run(DemoApplication.class, args);
19         }
20
21 }
22 @RestController
23 class HelloRestController {
24         private static final Logger logger = LoggerFactory.getLogger(HelloRestController.class);
25         @RequestMapping("/hello/{name}")
26         public String helloDocker(@PathVariable(value = "name") String name) throws UnknownHostException {
27                 String hostname = InetAddress.getLocalHost().getHostName();
28
29                 String response = "Halo Selamat Datang " + name + " Response received on : " + new Date() + " My Hostname : " + hostname;
30                 logger.info("Response :: {}", response);
31                 return response;
32
```

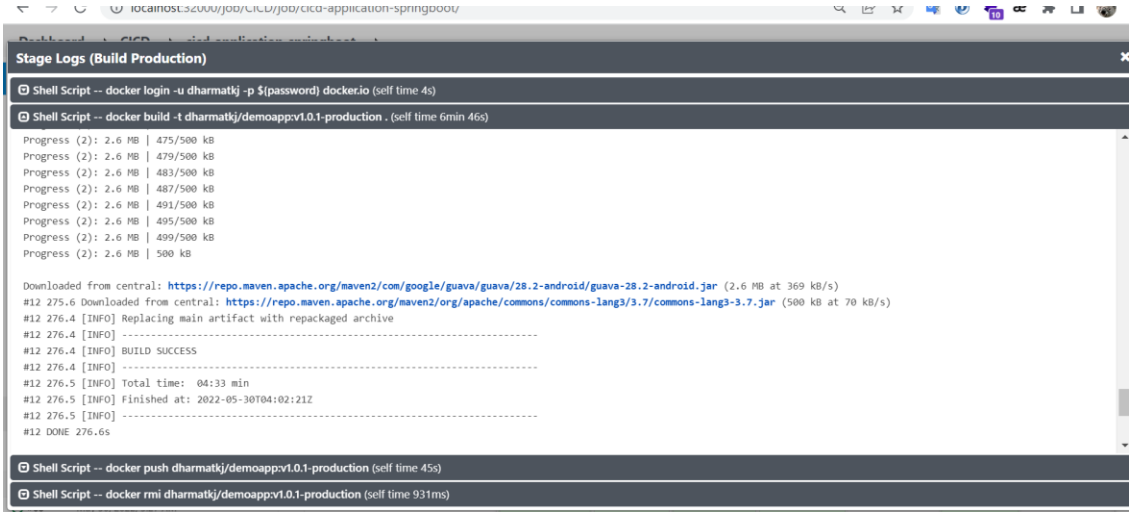
- Go to Jenkins with run pipeline and input fields to deploy production.



- Pipeline already to process jobs to update image applications and deploy.

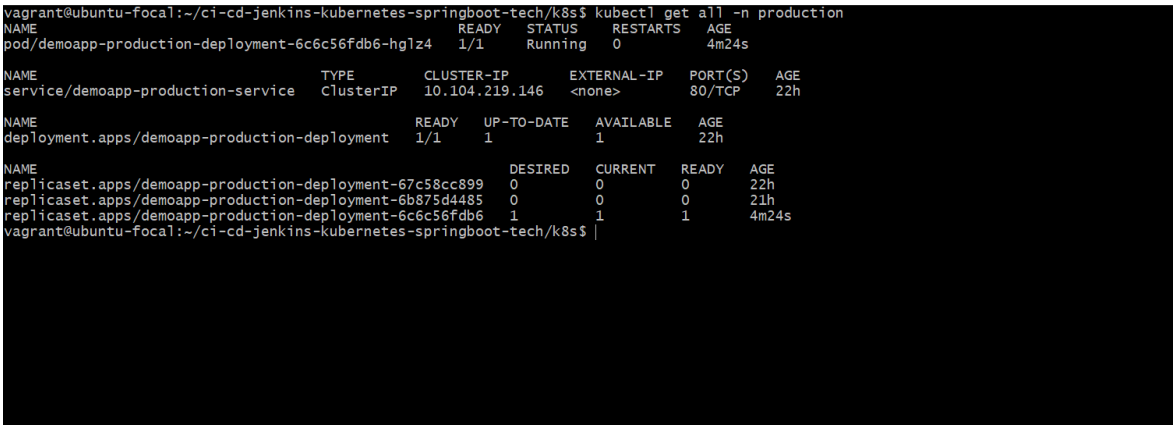


- Show log on job build



- Show logs deploy to make sure job success.

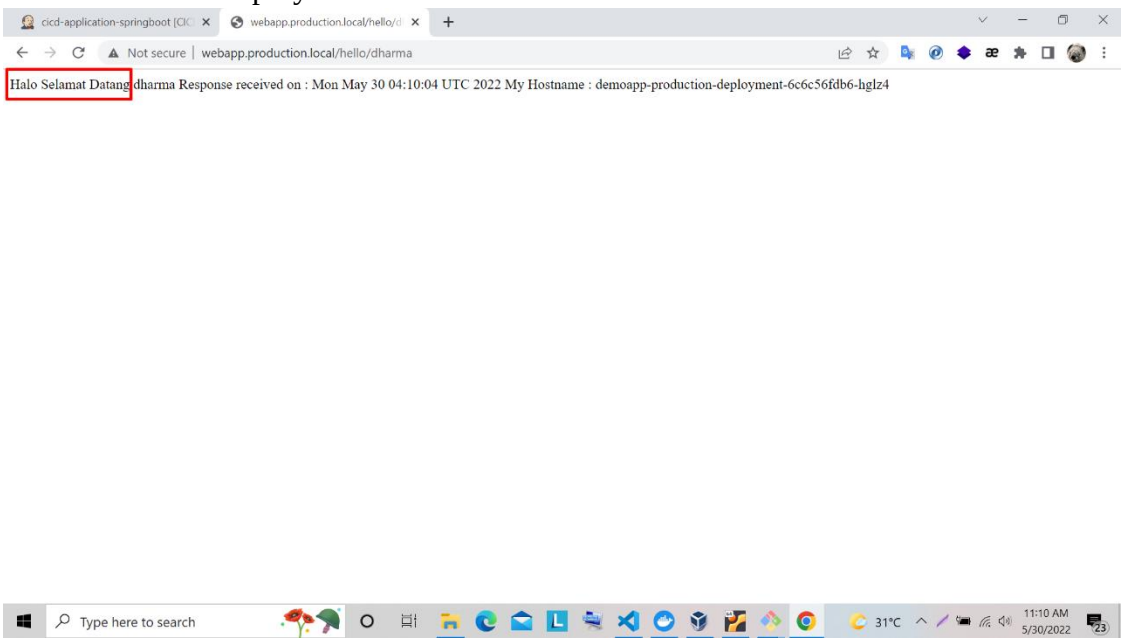
- Check service on Kubernetes after deployment update.



- Make sure to changes tag version image on deployment file.

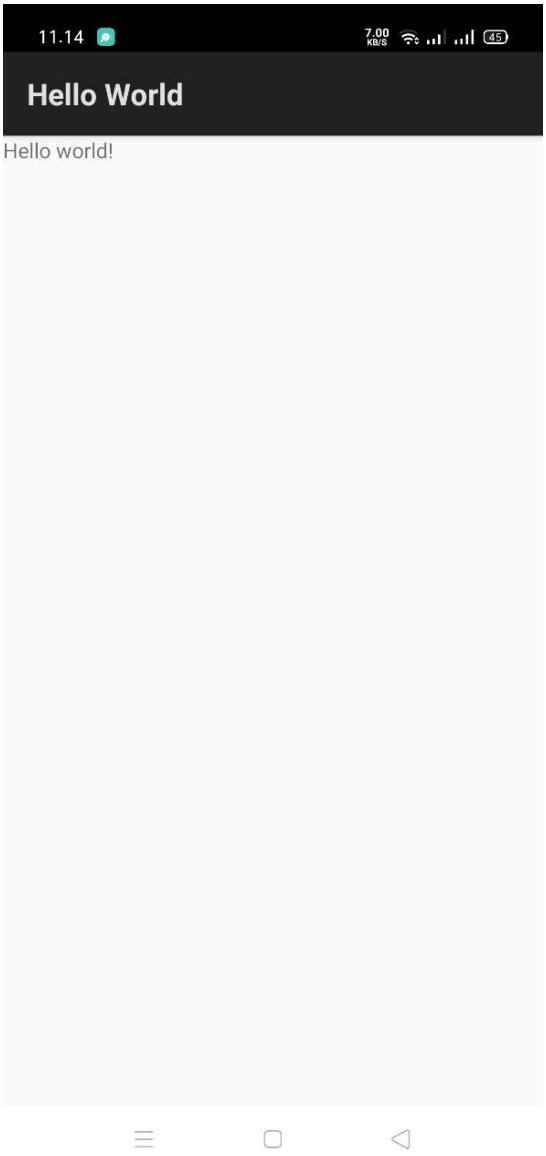
```
vagrant@ubuntu-focal:~/ci-cd-jenkins-kubernetes-springboot-tech/k8s$ cat prod/deployment.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    kubernetes.io/object: deployment
    app: demoapp
    env: production
  name: demoapp-production-deployment
  namespace: production
spec:
  replicas: 1
  selector:
    matchLabels:
      app: demoapp
      env: production
  template:
    metadata:
      labels:
        app: demoapp
        env: production
    spec:
      imagePullSecrets:
        - name: dockerhubpull
      containers:
      - image: dharmatkj/demoapp:v1.0.1-production
        imagePullPolicy: "Always"
        name: demoapp
        ports:
          - containerPort: 8080vagrant@ubuntu-focal:~/ci-cd-jenkins-kubernetes-springboot-tech/k8s$
```

- Final result after deployment.



B. How to use CI/CD Android with Jenkins

- **Check status before update deployment application android**
  - Please open application on your mobile phone.



- **Developer create branch and update page mobile**
  - Go to your code editor and update source on new branch for deploy

README.md | strings.xml | secret.yml M

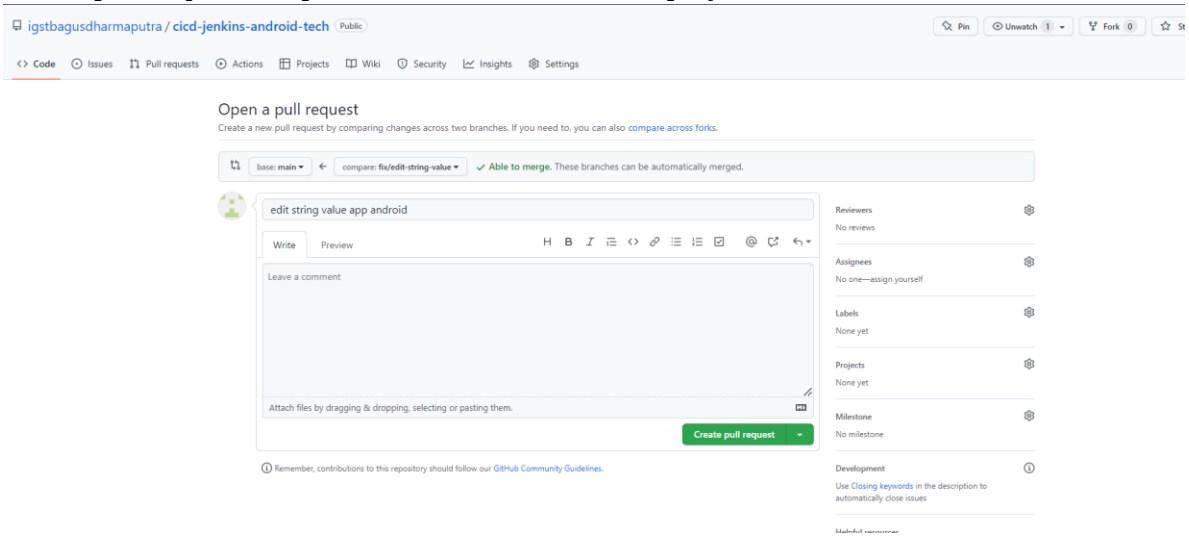
app > cicd-jenkins-android-tech > app > src > main > res > values > strings.xml

```
1  <resources>
2    <string name="app_name">Android App</string>
3    <string name="hello_world">Selamat Datang Dharma!</string>
4  </resources>
```

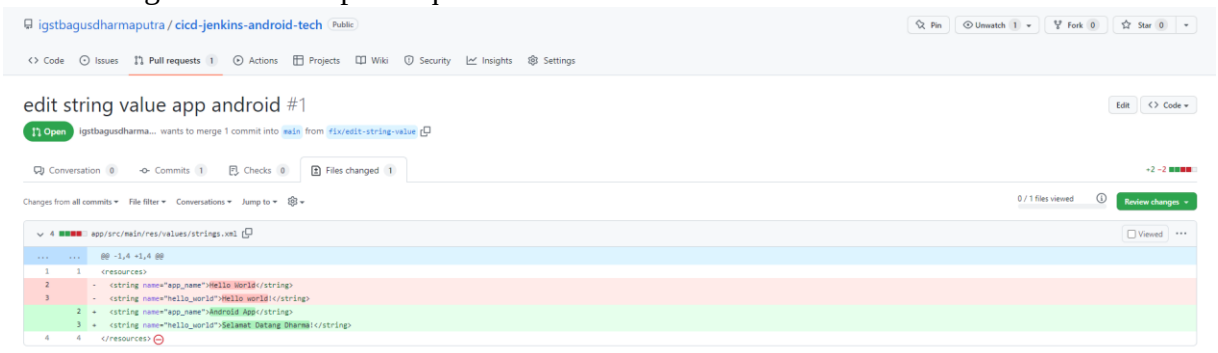
  - Push branch to update on repository.

```
Klik Digital Sinergi@LAPTOP-6OSCBK75 MINGW64 /d/Test Mandiri/app/cicd-jenkins-android-tech (main)
$ git branch
* main

Klik Digital Sinergi@LAPTOP-6OSCBK75 MINGW64 /d/Test Mandiri/app/cicd-jenkins-android-tech (main)
$ git checkout -b fix/edit-string-value
Switched to a new branch 'fix/edit-string-value'

Klik Digital Sinergi@LAPTOP-6OSCBK75 MINGW64 /d/Test Mandiri/app/cicd-jenkins-android-tech (fix/edit-string-value)
$ git add .

Klik Digital Sinergi@LAPTOP-6OSCBK75 MINGW64 /d/Test Mandiri/app/cicd-jenkins-android-tech (fix/edit-string-value)
$ git status
On branch fix/edit-string-value
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   app/src/main/res/values/strings.xml


Klik Digital Sinergi@LAPTOP-6OSCBK75 MINGW64 /d/Test Mandiri/app/cicd-jenkins-android-tech (fix/edit-string-value)
$ git commit -m "edit string value app android"
[fix/edit-string-value 92cb8c7] edit string value app android
 1 file changed, 2 insertions(+), 2 deletions(-)

Klik Digital Sinergi@LAPTOP-6OSCBK75 MINGW64 /d/Test Mandiri/app/cicd-jenkins-android-tech (fix/edit-string-value)
$ git push origin fix/edit-string-value
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 668 bytes | 668.00 KiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'fix/edit-string-value' on GitHub by visiting:
remote:      https://github.com/igstbagusdharmaputra/cicd-jenkins-android-tech/pull/new/fix/edit-string-value
remote:
To https://github.com/igstbagusdharmaputra/cicd-jenkins-android-tech
 * [new branch]      fix/edit-string-value -> fix/edit-string-value
```

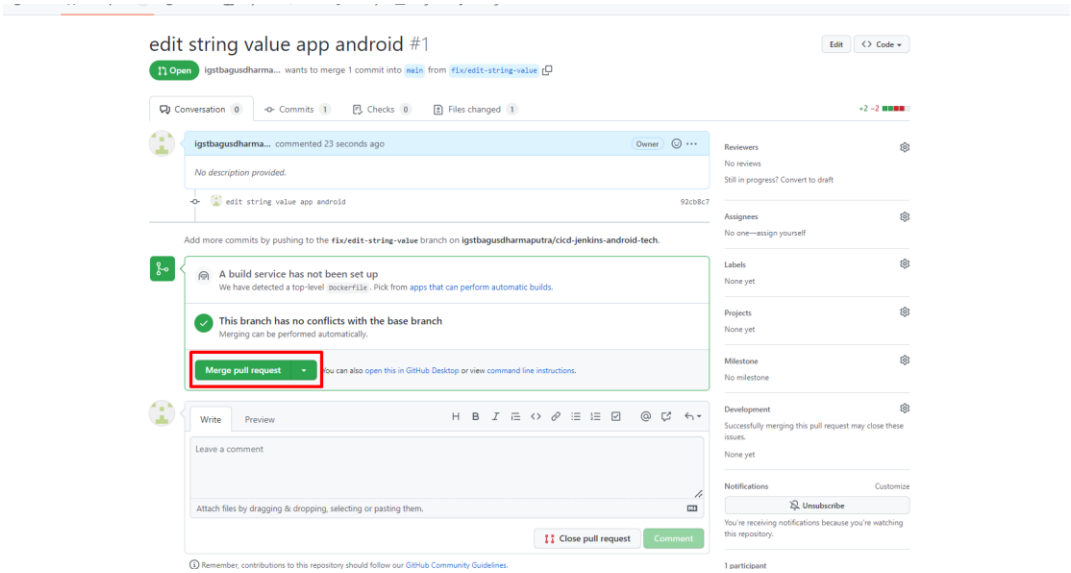
- Create pull request to update on branch **main** for deployment.



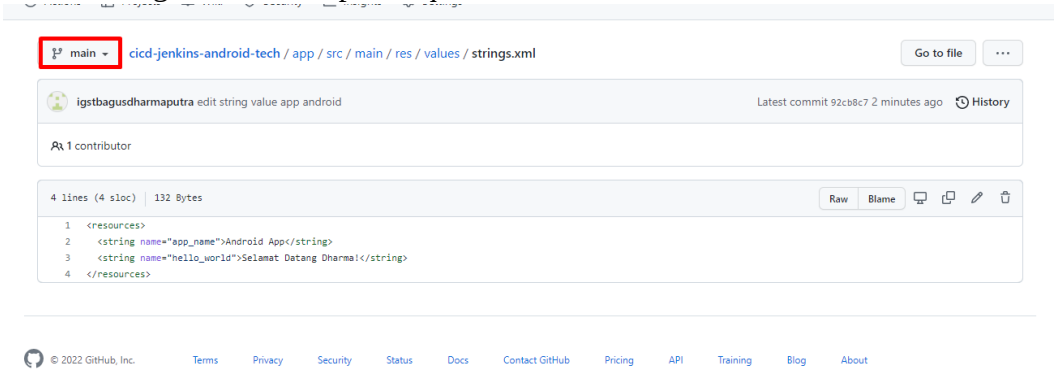
- Show changes code when pull request.



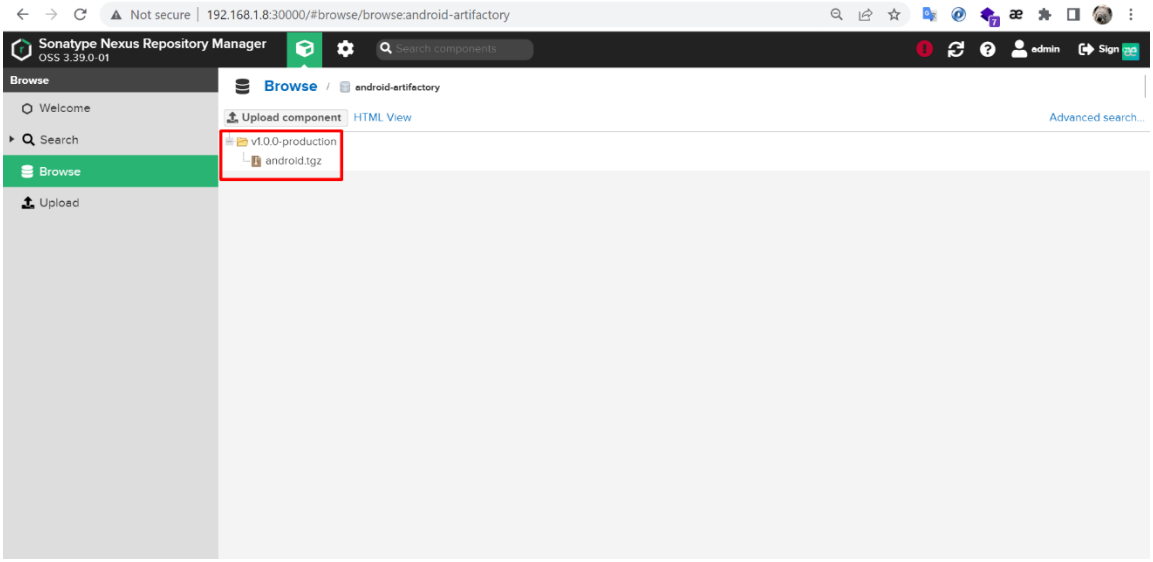
- Approve pull request to update on branch **main**.



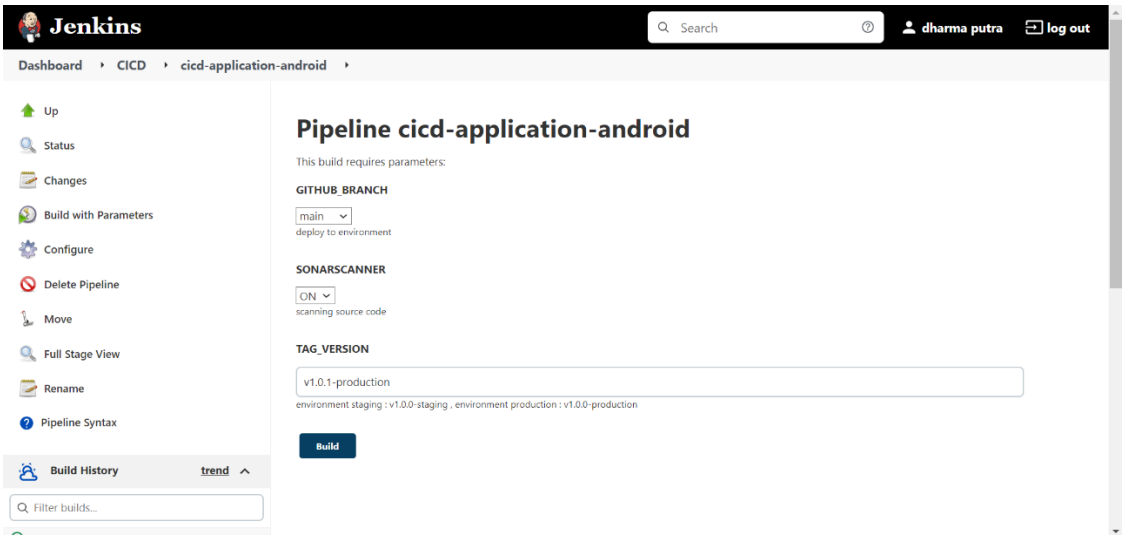
- Check changes code after pull request on branch **main**.



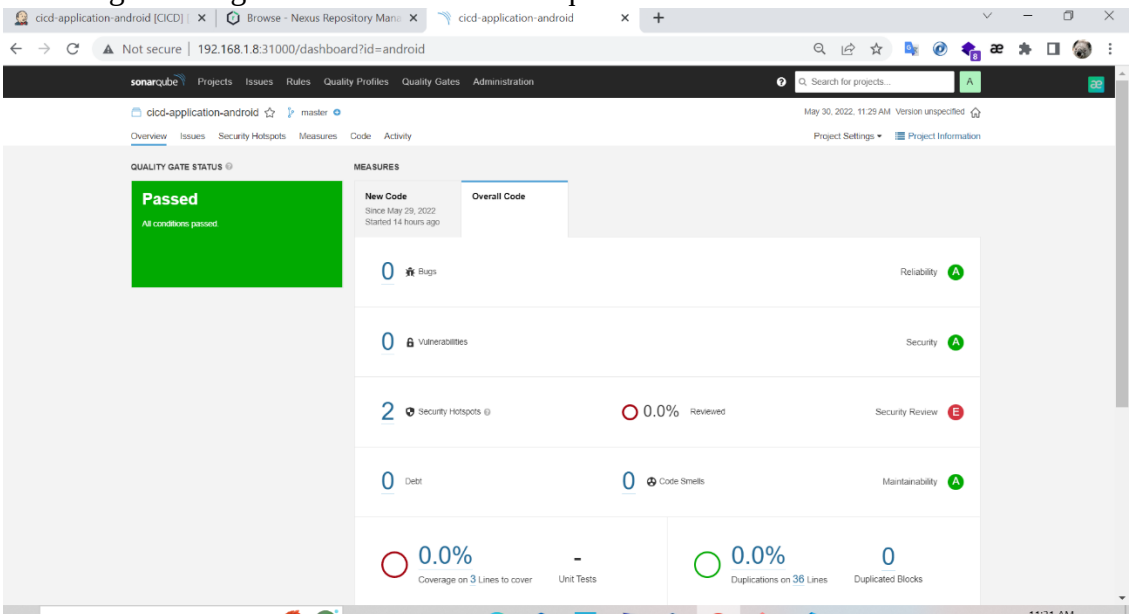
- Show artifactory application android before deploy.

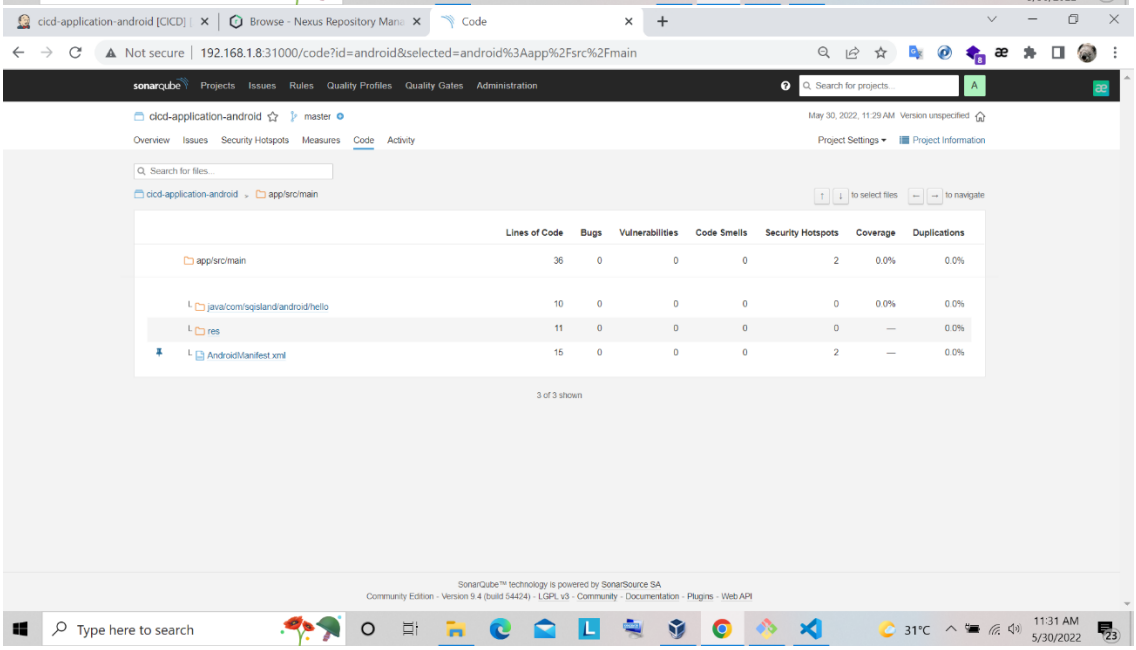


- **Run Jenkins to Deploy Application Mobile**
  - Go to Jenkins and select job to use deployment application android.
  - Input fields for running pipeline process.

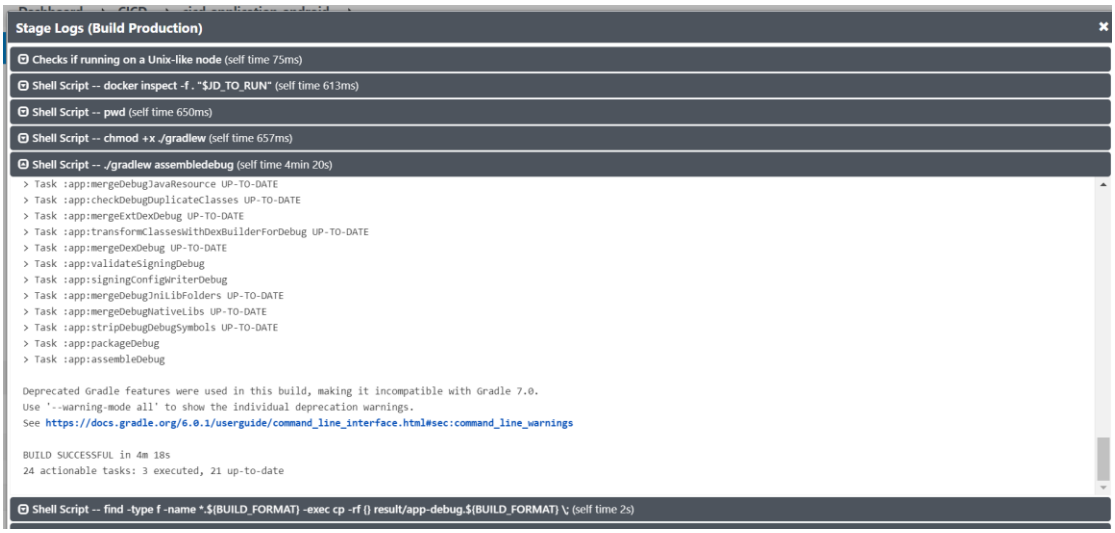


- Show log scanning for check code on sonarqube.

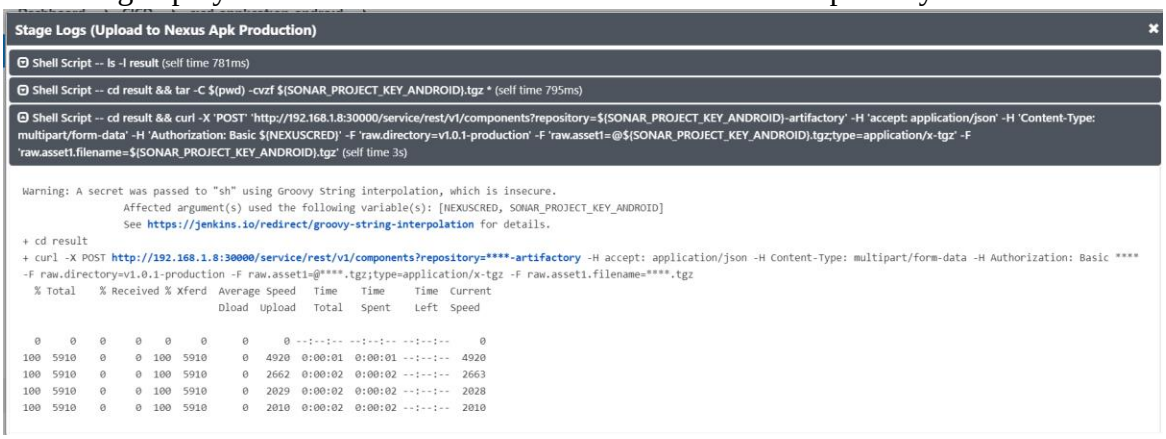




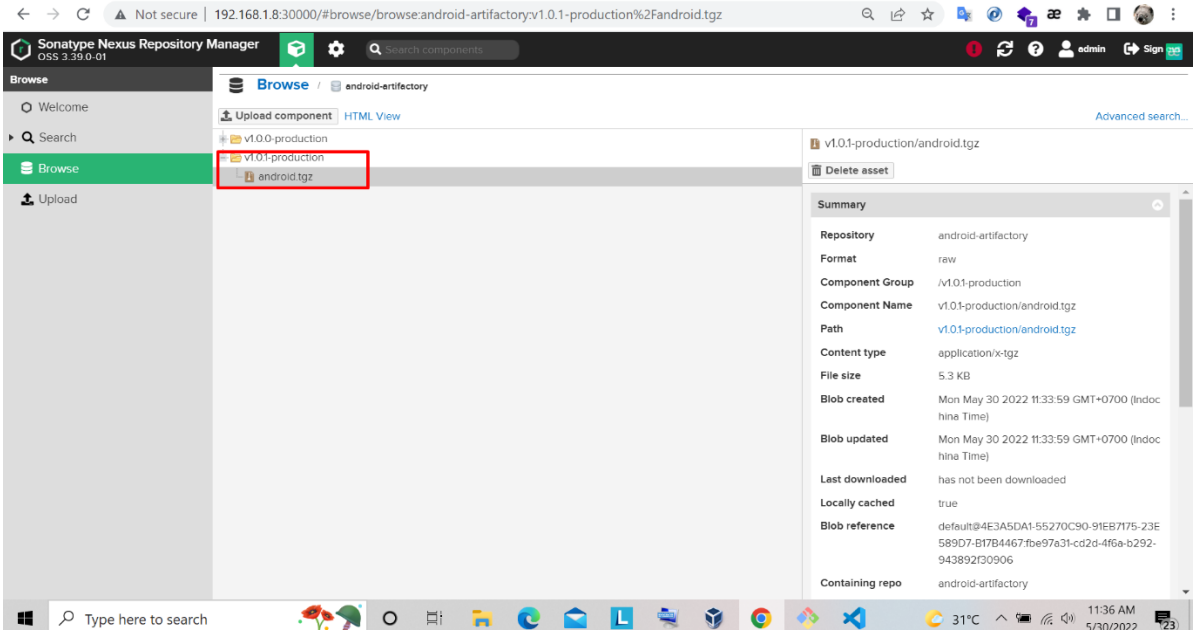
- Show log build to check process build success.



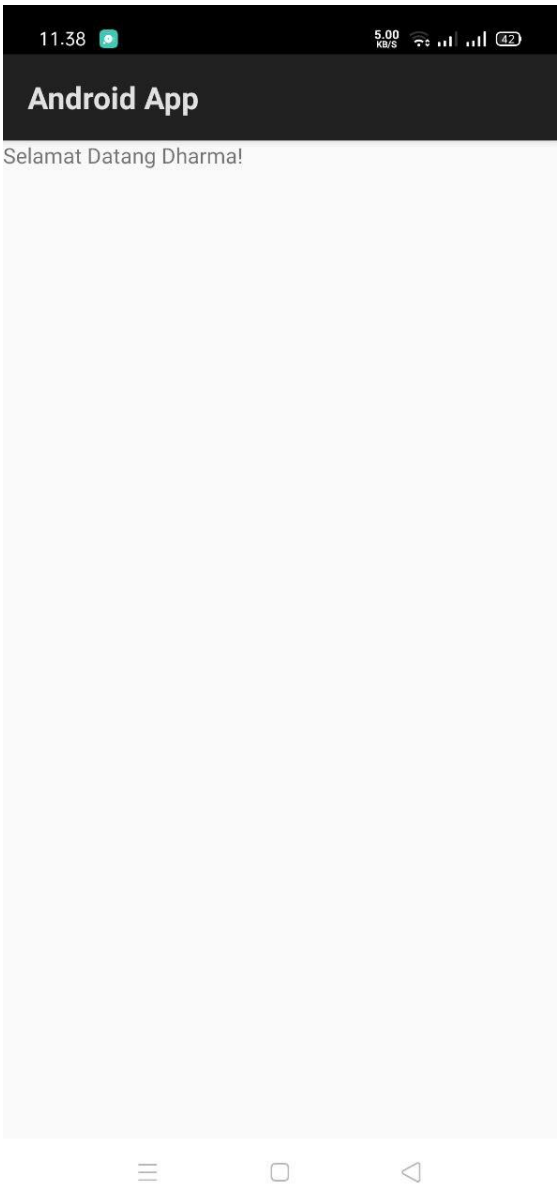
- Check log deploy to make sure artifact android store on nexus repository.



- Download artifact on nexus repository and install on your mobile phone.



- Final result after install new mobile application


11.38
5.00
KB/S
42
Android App
Selamat Datang Dharma!