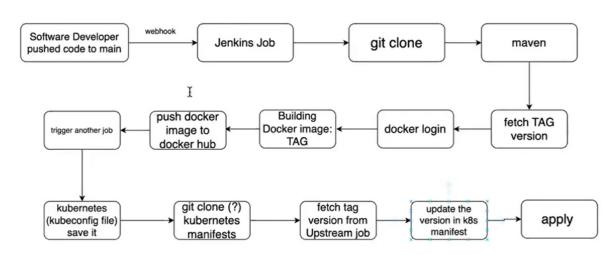
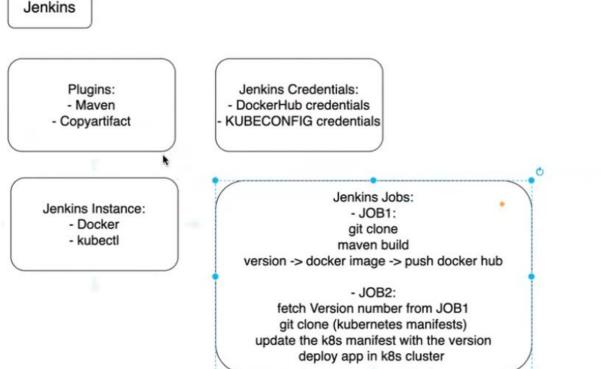
ALL INTEGRATION TOOLS



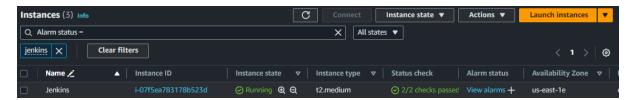


1. CI and CD





1. Launch Instances



2. Change Hostname

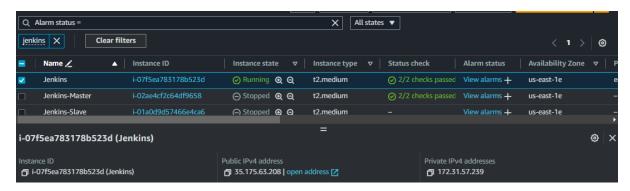
3. Install Jenkins

```
ubuntu@ip-172-31-57-239:~ X ubuntu@Jenkins-Master:~ X + V
ubuntu@ip-172-31-57-239:~$ nano jenkins_install.sh
```

4. Copy the things and Paste into the editor

```
ubuntu@Jenkins: ~
                            ubuntu@Jenkins-Master: ~
 GNU nano 7.2
                                                      jenkins_install.sh *
sudo apt update
sudo apt install openjdk-17-jre -y
java --version
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.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 -y
sudo systemctl daemon-reload
sudo systemctl enable jenkins
sudo systemctl start jenkins
sudo systemctl status jenkins
```

5. Give Execute Permission



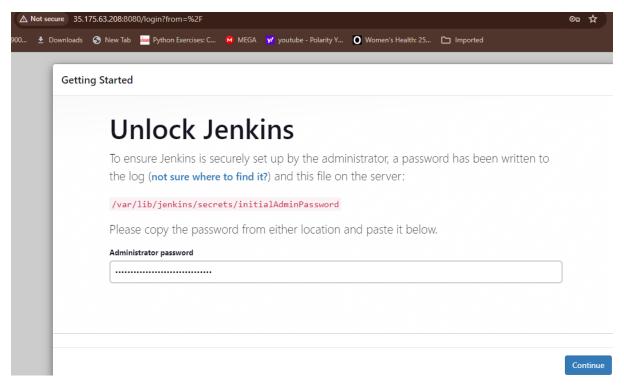
6. Jenkins Installed

```
jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
Active: active (running) since Fri 2024-06-07 15:36:53 UTC; 3s ago
  Main PID: 3648 (java)
    Tasks: 47 (limit: 4676)
   Memory: 1.2G (peak: 1.2G)
     CPU: 46.294s
   CGroup: /system.slice/jenkins.service
           -
-3648 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --web
Jun 07 15:36:35    Jenkins jenkins[3648]: eac7d2d2e272420ab231dd87d0ac9597
Jun 07 15:36:35 Jenkins jenkins[3648]: This may also be found at: /var/lib/jenkins/secrets/initia
Jun 07 15:36:53 Jenkins jenkins[3648]: 2024-06-07 15:36:53.754+0000 [id=31]
Jun 07 15:36:53 Jenkins jenkins[3648]: 2024-06-07 15:36:53.780+0000 [id=24]
                                                                  INFO
                                                                           hu
Jun 07 15:36:53 Jenkins jenkins[3648]: 2024-06-07 15:36:53.898+0000 [id=49]
                                                                           hu
lines 1-20/20 (END)
```

7. Go this Path and Copy it



8. Paste it



9. Install things

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

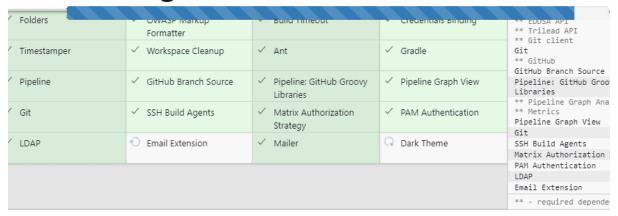
Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

10. Installed the things successfully

Getting Started



11. Create Admin User Credentials



Skip and continue as admin

Save and Continue

Instance Configuration

Jenkins URL:

http://35.175.63.208:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build stens.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.452.1

Not now

Save and Finish

13. Start using Jenkins

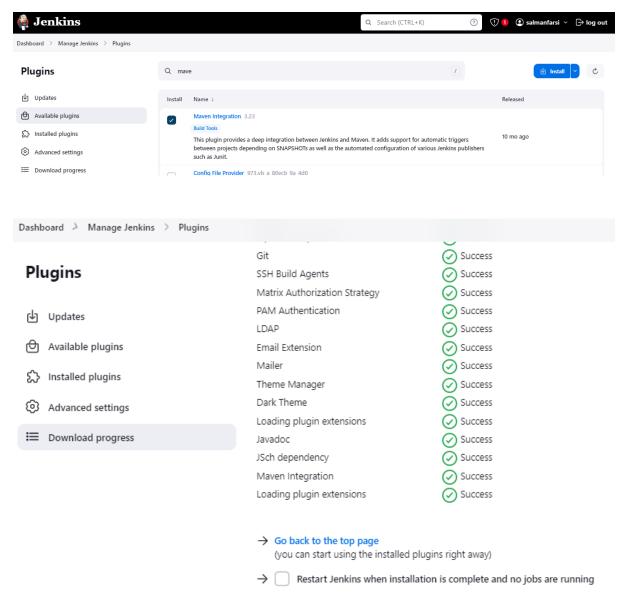
Getting Started

Jenkins is ready!

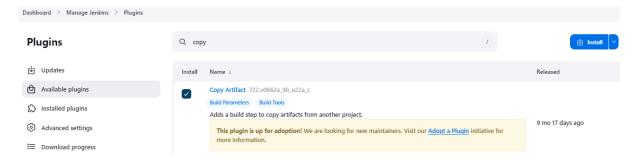
Your Jenkins setup is complete.

Start using Jenkins

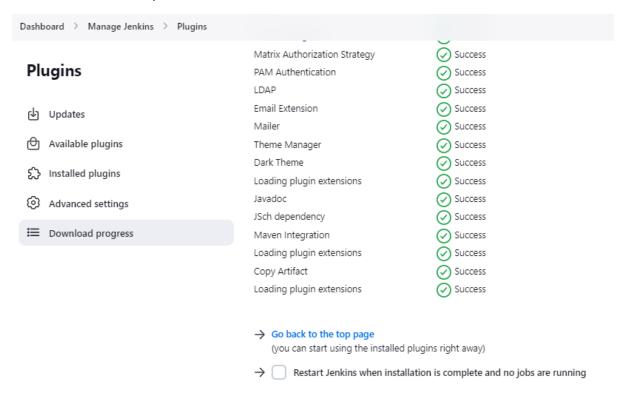
14. Go to Dashboard > Manage Jenkins > Plugins > in that Click Available Plugins search Maven Integration and Install it.



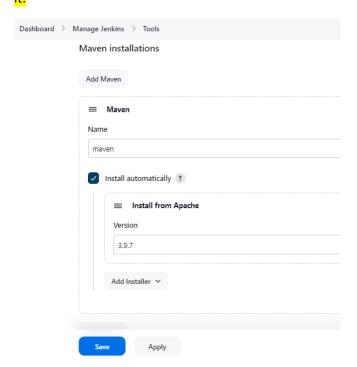
15. Same Like Install Copy Artifact



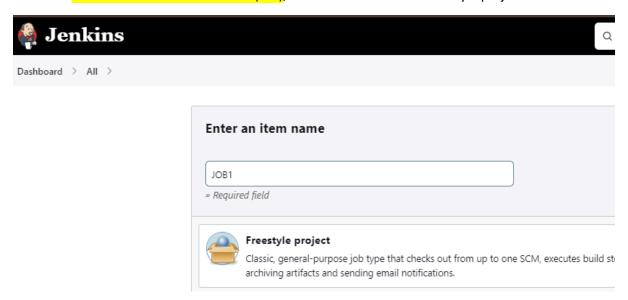
16. installed successfully



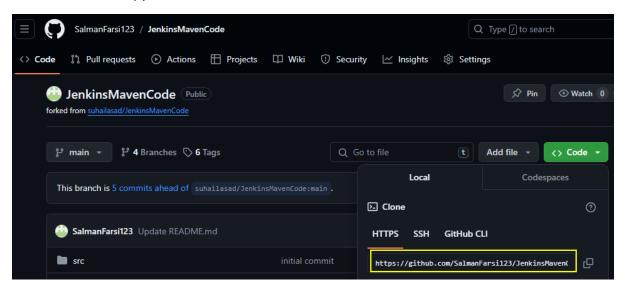
17. Go Dashboard > Manage Jenkins > Tools and Add Maven and Version which you want and Save it.



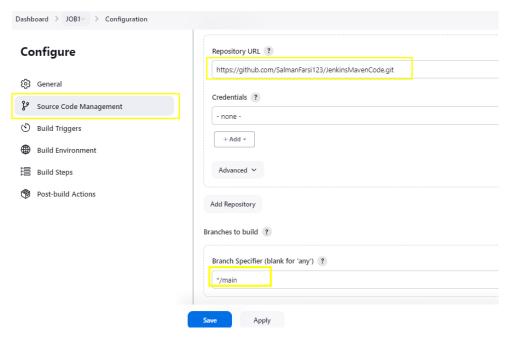
18. Go to Dashboard and Click New Item(Job), Create a New Job and Freestyle project and save it.



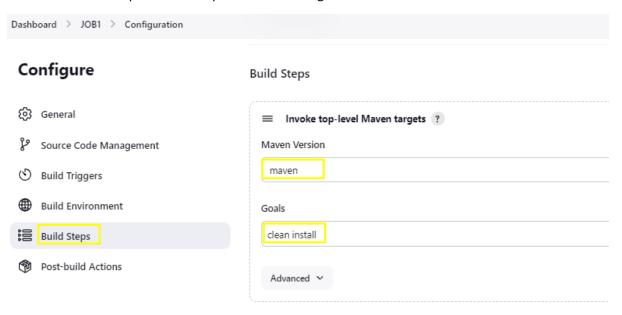
19. Go To GitHub Copy the Maven Code Path.



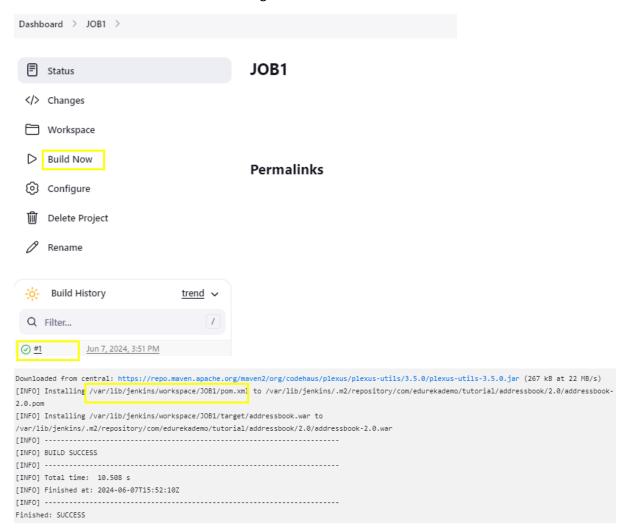
20. Paste in the source code management and Type main branch



21. And Go Build Steps > Invoke top-level Maven targets



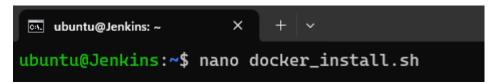
22. Click On Build Now and its ran Successfully and Now Maven successfully change that github mavencode into executable format through maven tool. And this is the JOB1



23. Go to Terminal here and Paste the path format, HERE MAVEN TOOL HAS DONE HIS JOB WHICH IS CREATED TARGET FILE

```
ubuntu@Jenkins: /var/lib/jenl ×
ubuntu@Jenkins:~$ cd /var/lib/jenkins/workspace/JOB1/
ubuntu@Jenkins:/var/lib/jenkins/workspace/JOB1$ ls
Dockerfile
                                             build.xml
                                                             pom.xml target
             build.properties
README.md
                                              context.xml
                                                                       tomcat-users.xml
ubuntu@Jenkins:/var/lib/jenkins/workspace/JOB1$ cd target
ubuntu@Jenkins:/var/lib/jenkins/workspace/JOB1/target$ ls
                                         generated-test-sources maven-status
                                                                                          test-classes
addressbook.war generated-sources maven-archiver
                                                                     surefire-reports
ubuntu@Jenkins:/var/lib/jenkins/workspace/JOB1/target$
```

24. Now install the Docker



25. Paste the installation stuff, which u can find in the Suhail drive

```
GNU nano 7.2

sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
echo \
"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker=
"$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" |
sudo apt-get update|
sudo apt-get update|
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
```

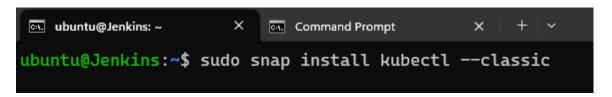
26. Give Permission and Run it.

27. And Give Permission to Ubuntu user and exit it and again login it.

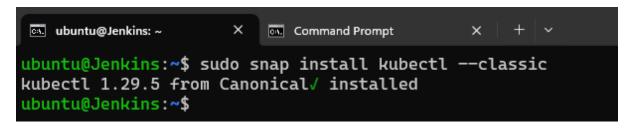
```
Command Prompt X Command Prompt X + V - X Ubuntu@Jenkins:~$ sudo usermod -aG docker ubuntu ubuntu@Jenkins:~$ exit logout Connection to ec2-35-175-63-208.compute-1.amazonaws.com closed.

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-35-175-63-208.com pute-1.amazonaws.com
```

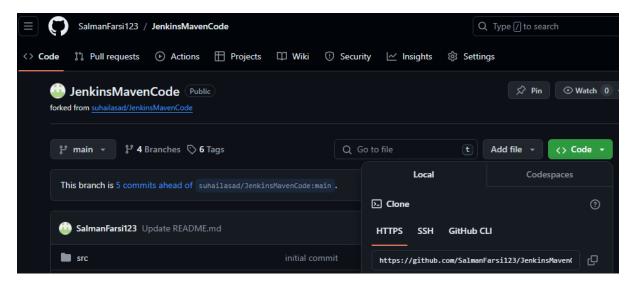
28. Install kubectl



29. Its installed successfully.



30. Go to Github and jenkinsMavenCode path copy it

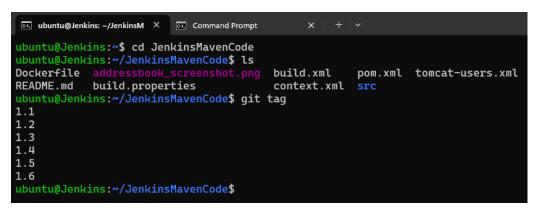


31. Do Git Clone

```
ubuntu@Jenkins:~/JenkinsM × Command Prompt × + + v

ubuntu@Jenkins:~$ git clone https://github.com/SalmanFarsi123/JenkinsMavenCode.git
Cloning into 'JenkinsMavenCode'...
remote: Enumerating objects: 133, done.
remote: Counting objects: 100% (50/50), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 133 (delta 39), reused 21 (delta 20), pack-reused 83
Receiving objects: 100% (133/133), 170.48 KiB | 15.50 MiB/s, done.
Resolving deltas: 100% (44/44), done.
```

32. Now check the git tag

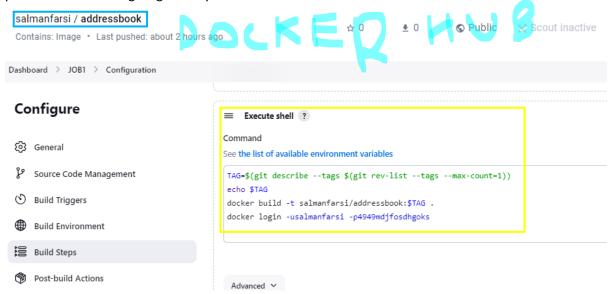


33. Now this git tag 1.6 and 76a1eac10cfd9b531e008c89c6b07ea496d2ad8 given name as TAG and the tag data stored in TAG with using linux commands

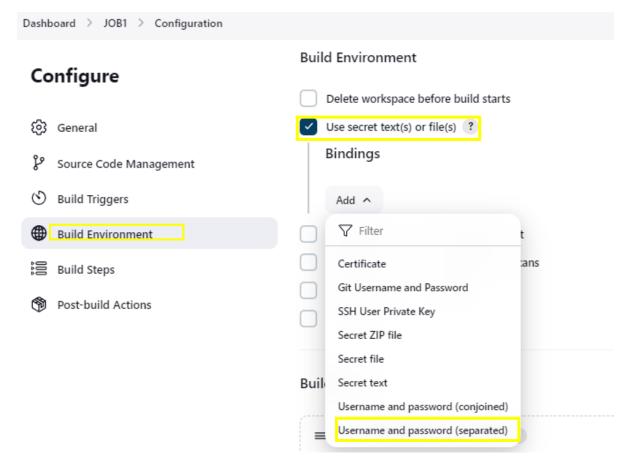
```
ubuntu@Jenkins:~/JenkinsM × Command Prompt × + | v

ubuntu@Jenkins:~/JenkinsMavenCode$ git rev-list --tags --max-count=1
76a1eac10cfd9b5361e008c89c6b07ea496d2ad8
ubuntu@Jenkins:~/JenkinsMavenCode$ git describe --tags 76a1eac10cfd9b5361e008c89c6b07ea496d2ad8
1.6
ubuntu@Jenkins:~/JenkinsMavenCode$ git describe --tags $(git rev-list --tags --max-count=1)
1.6
ubuntu@Jenkins:~/JenkinsMavenCode$ TAG=$(git describe --tags $(git rev-list --tags --max-count=1))
ubuntu@Jenkins:~/JenkinsMavenCode$ echo $TAG
1.6
ubuntu@Jenkins:~/JenkinsMavenCode$
```

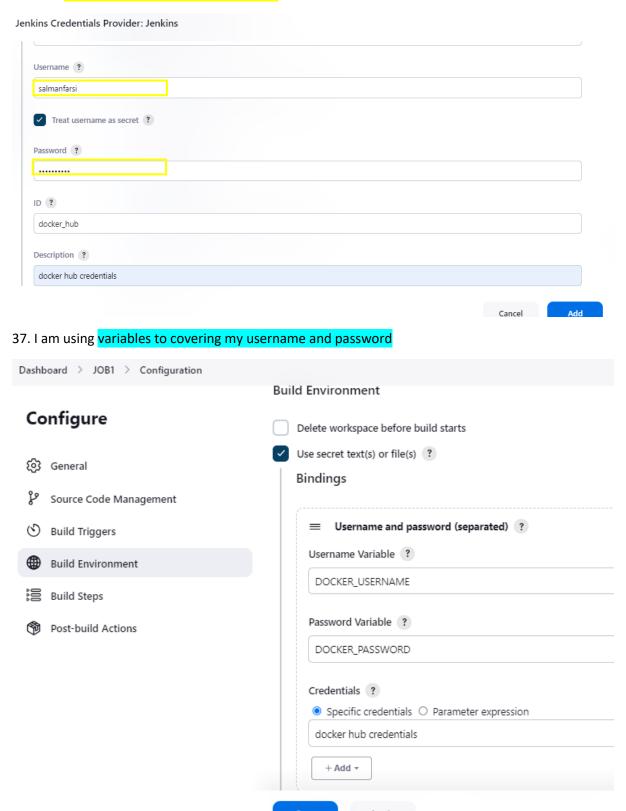
34. Go to Job1 > Configuration > Build Steps Paste the Commands first the TAG data will and then Tag data will be print and going to build docker and salmanfarsi/addressbook repository make sure uneed to have dockerhub account with have salmanfarsi/addressbook this name repository and paste docker login and paste dockhub username and password Below mentioned how to cover that password instead of giving directly.



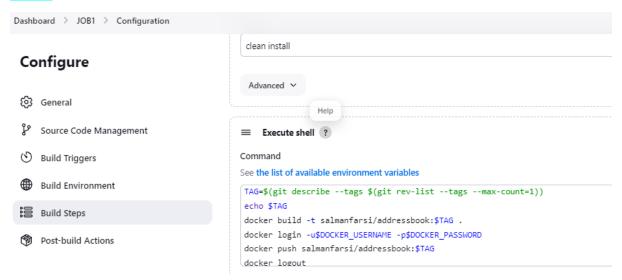
35. Go Build Environment > use secret file > username and password separated



36. Give Dockerhub username and password and ID and Description is your wish.

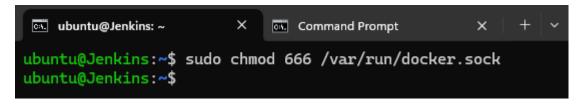


38. Now we can see username and password was covered. And pushing to dockerhub and docker logout.

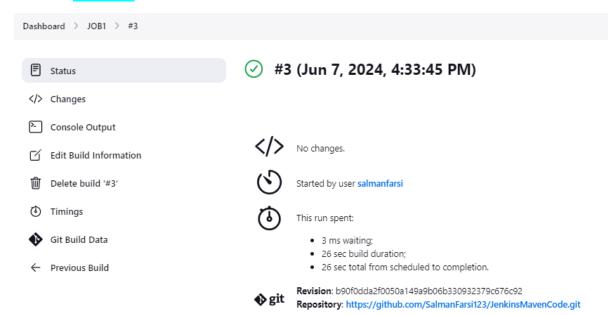


39. Click build and Now we can see Permision denied copy that /var/run/docker.sock

40. Sudo change permission to that file(Giving Permission to owner, Group, allusers = read and write permission)



41. Again Build Now its successfully ran.

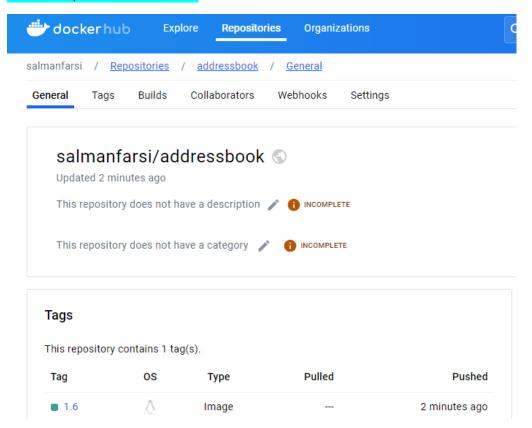


42. We can See the Success message.

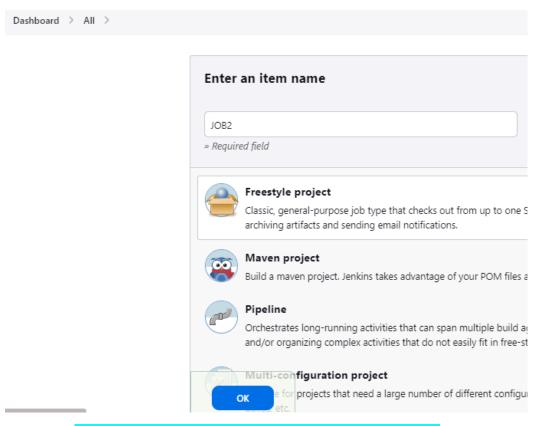
```
Dashboard > JOB1 > #3 > Console Output
                                                 f67287db387a: Waiting
                                                 1e85ed7274f1: Waiting
                                                 2f140462f3bc: Waiting
                                                 63c99163f472: Waiting
                                                 ccdbb80308cc: Waiting
                                                 1590c3df0957: Pushed
                                                 65fe9464d6bf: Pushed
                                                 dc3bac03618b: Pushed
                                                 225e381f3879: Mounted from library/tomcat
                                                 babf80fd23b7: Pushed
                                                 7c845aef85c9: Mounted from library/tomcat
                                                 d9210666eabd: Mounted from library/tomcat
                                                 f67287db387a: Mounted from library/tomcat
                                                 1e85ed7274f1: Mounted from library/tomcat
                                                 2f140462f3bc: Mounted from library/tomcat
                                                 63c99163f472: Mounted from library/tomcat
                                                 ccdbb80308cc: Mounted from library/tomcat
                                                 7a090d012ef5: Pushed
                                                 1.6: digest: sha256:6869e6129e14b3f453e4e75e1402fbf96e4a77ea1dbd18b68bc4d184cc4367dd size: 3041
                                                 Removing login credentials for https://index.docker.io/v1/
```

• refs/remotes/origin/main

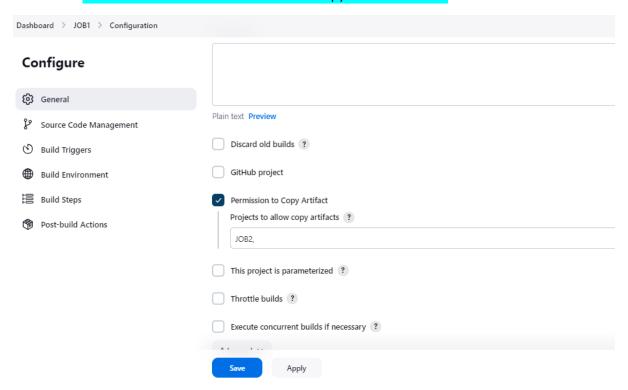
43. This is My Dockerhub Account we can see into repositories Tag 1.6 and Type Image was Pushed Successfully into the DockerHub



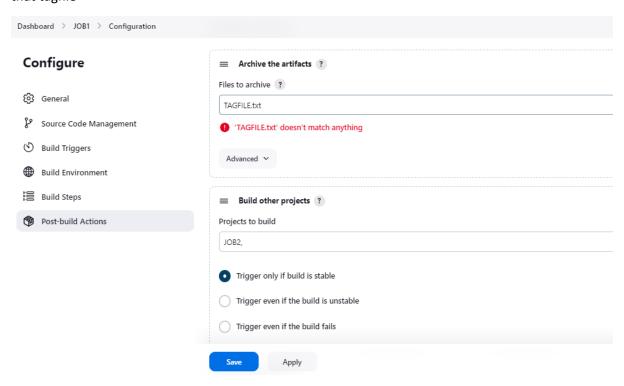
44. Create Job2 and Select Freestyle Project and Create it. As of Now don't Configure in Job2



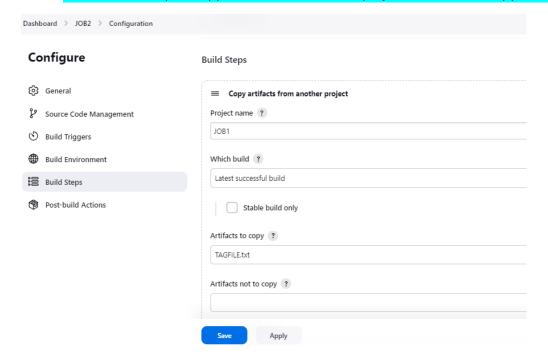
45. And Go to Job1 and In General > Permission to Copy Artifacts > select it



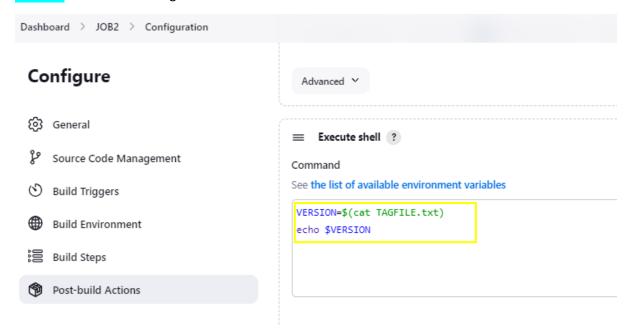
46. In Post-Build Actions > Archive the artifacts and Going to save TAGFILE.txt in JOB2 which is 1.6 in that tagfile



47. Go To Job2 Build Steps > Copy Artifacts from another project > Artifacts to copy TAGFILE.txt



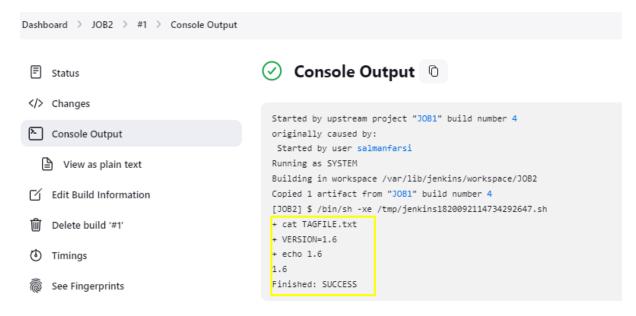
48. Post-Build Actions > Execute Shell Version =\$(cat TAGFILE.txt) and after that it will print what it will? 1.6 because in the tag file 1.6 data has been stored



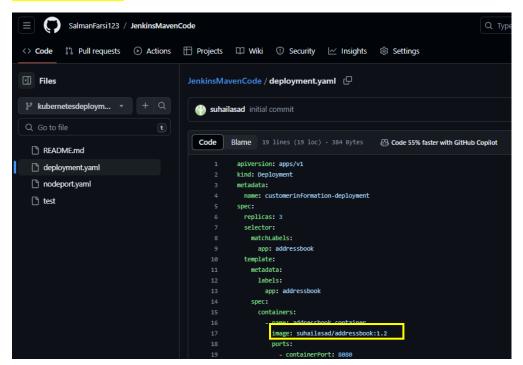
49. Ran the job successfully and triggering to JOB2

```
Dashboard > JOB1 > #4 > Console Output
                                                 ccdbb80308cc: Waiting
                                                 225e381f3879: Waiting
                                                 7c845aef85c9: Waiting
                                                 dc3bac03618b: Layer already exists
                                                 1590c3df0957: Layer already exists
                                                 65fe9464d6bf: Layer already exists
                                                 225e381f3879: Layer already exists
                                                 d9210666eabd: Laver already exists
                                                 7c845aef85c9: Layer already exists
                                                 babf80fd23b7: Layer already exists
                                                 f67287db387a: Layer already exists
                                                 1e85ed7274f1: Layer already exists
                                                 63c99163f472: Layer already exists
                                                 2f140462f3bc: Layer already exists
                                                 ccdbb80308cc: Layer already exists
                                                 729aec11dc22: Pushed
                                                 1.6: digest: sha256:27740186a0a593c377823f42ecfaa721bb6eeeb6afc14fee434756cb217ff912 size: 3041
                                                 + docker logout
                                                 Removing login credentials for https://index.docker.io/v1/
                                                 Archiving artifacts
                                                 Triggering a new build of JOB2
                                                Finished: SUCCESS
```

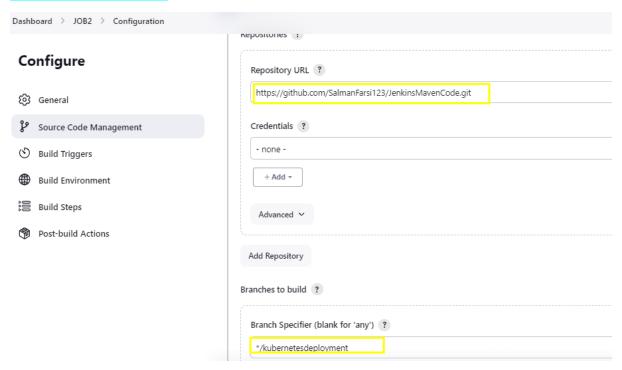
50. see the things, cat TAGFILE.txt in version=1.6 echo 1.6 = 1.6



51. deployment.yaml > in the bottom image suhailasad/addressbook:1.2 and we can edit but nothing to do here



52. In here Source Code Management > Repository URL > Our Github repository and Branch should be kubernetesdeployment



54. Go to Jenkins > JenkinsMavenCode > git branch, Is, git checkout kubernetesdeployment, Is

```
ubuntu@Jenkins: ~/JenkinsM ×
                             Command Prompt
ubuntu@Jenkins:~/JenkinsMavenCode$ git branch
ubuntu@Jenkins:~/JenkinsMavenCode$ ls
Dockerfile addressbook_screenshot.png
                                                build.xml
                                                                pom.xml tomcat-users.xml
              build.properties
README.md
                                                context.xml
                                                               src
ubuntu@Jenkins:~/JenkinsMavenCode$ git checkout kubernetesdeployment
branch 'kubernetesdeployment' set up to track 'origin/kubernetesdeployment'.
Switched to a new branch 'kubernetesdeployment' ubuntu@Jenkins:~/JenkinsMavenCode$ ls
README.md deployment.yaml nodeport.yaml test
ubuntu@Jenkins:~/JenkinsMavenCode$
ubuntu@Jenkins:~/JenkinsMavenCode$ ls
README.md deployment.yaml nodeport.yaml test
ubuntu@Jenkins:~/JenkinsMavenCode$
```

55. Cat deployment.yaml in the bottom we can see 1.2 version is there

```
ubuntu@Jenkins: ~/JenkinsM ×
                          Command Prompt
                                               ×
ubuntu@Jenkins:~/JenkinsMavenCode$ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: customerinformation-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: addressbook
  template:
    metadata:
      labels:
        app: addressbook
    spec:
      containers:
        - name: addressbook-container
          image: suhailasad/addressbook:1.2
          ports:
            - containerPort: 8080
ubuntu@Jenkins:~/JenkinsMavenCode$
```

56. In Kubernetesdeployment branch we are giving the VERSION=1.6 and and writtening the one command

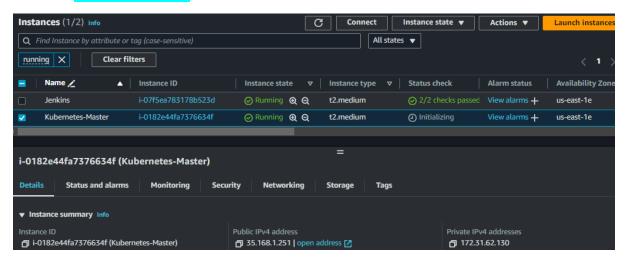
57. Now see again the image version 1.6 has been changed we can change from editor but using this command we have to integrate In Jenkins, I thought

```
ubuntu@Jenkins:~/JenkinsMavenCode$ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
netadata:
 name: customerinformation-deployment
spec:
 replicas: 3
 selector:
   matchLabels:
     app: addressbook
 template:
   metadata:
      labels:
       app: addressbook
   spec:
     containers:

    name: addressbook-container

          image: suhailasad/addressbook:1.6
          ports:
            - containerPort: 8080
ıbuntu@Jenkins:~/JenkinsMavenCode$
```

58. Select Kubernetes-Master



59. here we are connecting through with secret file not ANY SLAVE connecting remember it. And it will going to deploy in Kubernetes-manifest in your Kubernetes master node.

And copy it and save it in local machine

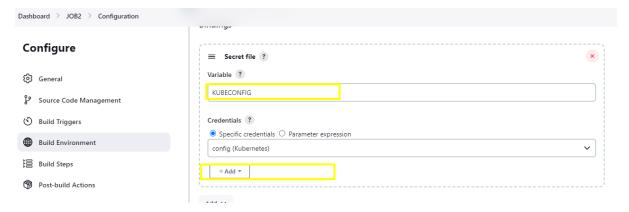
60. I used git bash, but we can also use notepad to save in text file it will works.



61. paste in git bash terminal and save it. It will save in windows machine dekstop or which location by default it will took, u can check with pwd

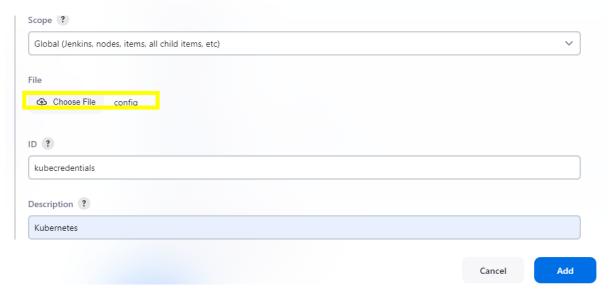
```
MINGW64:/c/Users/shaik
 GNU nano 8.0
                                                                       Modified
                                        config
apiVersion: v1
:lusters:
 cluster:
   certificate-authority-data: LSOtLS1CRUdJTiBDRVJUSUZJQ0FURSOtLSOtCk1JSURCVEN
   server: https://172.31.62.130:6443
 name: kubernetes
contexts:
 context:
   cluster: kubernetes
   user: kubernetes-admin
 name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
cind: Config
references: {}
users:
 name: kubernetes-admin
 user:
   client-certificate-data: LSOtLS1CRUdJTiBDRVJUSUZJQ0FURSOtLS0tCk1JSURJVENDQW
UklWQVRFIEtFWS0tLS0tCg==
```

62. Go Build Environment > Variable give KUBECONFIG and credentials I selected and follow the screenshot in 63 step

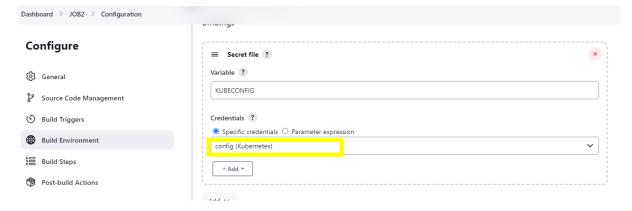


63. Choose file from Local Window Machine which u have saved. And ADD

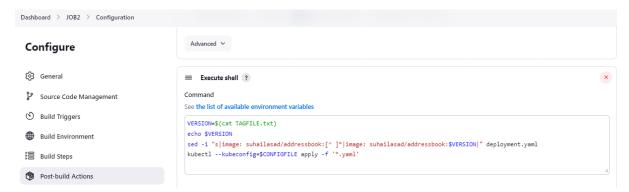
Jenkins Credentials Provider: Jenkins



64. And then Scroll



65. And Paste the 3rd and 4th command, One thing here is it will going to took the config data(Certificate) and going to apply and deploy the application in Kubernetes, In future we can understand how the commands actually works and why it use.



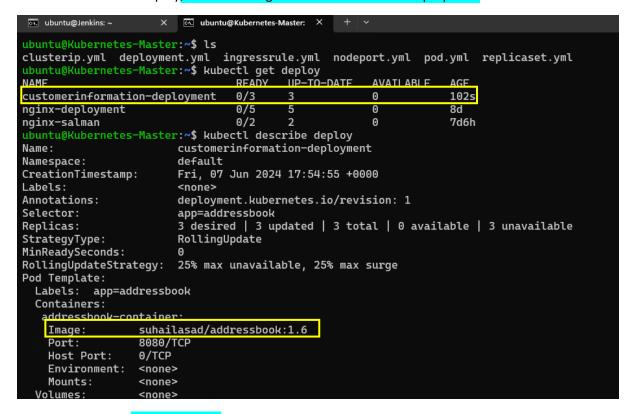
66. Run the Job1, it shows the success and Triggering the job2 because of copy artifacts setup.

```
Dashboard > JOB1 > #5 > Console Output
                                                 f67287db387a: Waiting
                                                 1e85ed7274f1: Waiting
                                                 2f140462f3bc: Waiting
                                                 1590c3df0957: Layer already exists
                                                 65fe9464d6bf: Laver already exists
                                                 dc3bac03618b: Layer already exists
                                                 babf80fd23b7: Laver already exists
                                                 225e381f3879: Layer already exists
                                                 d9210666eabd: Layer already exists
                                                 7c845aef85c9: Layer already exists
                                                 f67287db387a: Layer already exists
                                                 1e85ed7274f1: Layer already exists
                                                 2f140462f3bc: Layer already exists
                                                 63c99163f472: Layer already exists
                                                 ccdbb80308cc: Layer already exists
                                                 1c97d28afef6: Pushed
                                                 1.6: digest: sha256:e57e030f6b8299a8cfdd62ff174a8345e68bee19f4c9735e3768fd74547c53d3 size: 3041
                                                 Removing login credentials for https://index.docker.io/v1/
                                                 Archiving artifacts
                                                 Triggering a new build of JOB2
                                                 Finished: SUCCESS
```

67. We can in the bottom steps its shows deployment.apps/customerinformation-deployment created and in the nodeport service is publically it will available

```
Dashboard > JOB2 > #2 > Console Output
                                                            > git --version # 'git version 2.43.0'
                                                            > git fetch --tags --force --progress -- https://github.com/SalmanFarsi123/JenkinsMavenCode.git +refs/heads/*:refs/remotes/origin/* # timeout=10 > git config remote.origin.url https://github.com/SalmanFarsi123/JenkinsMavenCode.git # timeout=10
 See Fingerprints
                                                                                                                                                         Code.git # timeout=10
                                                             > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
                                                             > git rev-parse refs/remotes/origin/kubernetesdeployment^{commit} # timeout=10
                                                           Checking out Revision 302040ff7dde348a296d65c83f32017eff412a84 (refs/remotes/origin/kubernetesdeployment)
                                                            > git config core.sparsecheckout # timeout=10
                                                           > git checkout -f 302040ff7dde348a296d65c83f32017eff412a84 # timeout=10
Commit message: "Update test"
                                                           First time build. Skipping changelog.
Copied 1 artifact from "JOB1" build number
                                                           [JOB2] $ /bin/sh -xe /tmp/jenkins12616660909745864355.sh
                                                            + cat TAGFILE.txt
                                                           + VERSION=1.6
                                                           + sed -i s|image: suhailasad/addressbook:[^{n}]*|image: suhailasad/addressbook:1.6| deployment.yaml + kubectl --kubeconfig= apply -f *.yaml
                                                           deployment.apps/customerinformation-deployment created
                                                            service/customerinformation-nodeport created
                                                           Finished: SUCCESS
```

68. kubectl describe deploy, it will showing customerinformation-deployment



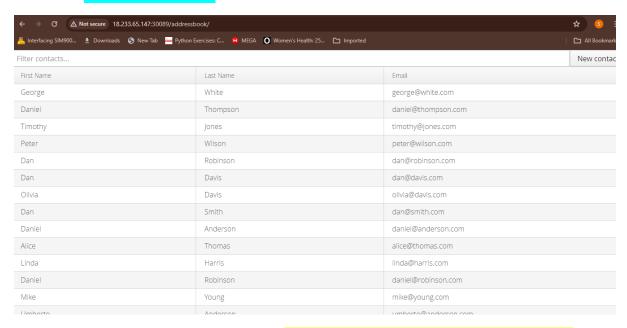
69. we can see the service is there

| □ ubuntu@Jenkins: ~ X □ ubuntu@Kubernetes-Master: X + ∨ | | | | | | | | | |
|---|-----------|----------------|---------------|--------------|-------|--|--|--|--|
| ubuntu@Kubernetes-Master:~\$ kubectl get svc | | | | | | | | | |
| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) | AGE | | | | |
| customerinformation-nodeport | NodePort | 10.104.80.255 | <none></none> | 80:30089/TCP | 3m20s | | | | |
| kubernetes | ClusterIP | 10.96.0.1 | <none></none> | 443/TCP | 9d | | | | |
| nginx-service-clusterip | ClusterIP | 10.109.225.246 | <none></none> | 8080/TCP | 8d | | | | |
| ubuntu@Kubernetes-Master:~\$ | | | | | | | | | |

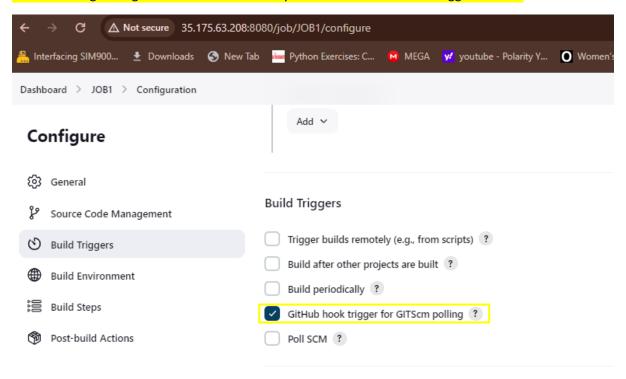
70. Observe the things Available 0, u cant able to see the data in browser because its unavailable, so need to Launch Kubernetes-slave which we already installed prerequites in both master and slave Kubernetes and we set-up it and then only u can see the available and then copy the PUBLIC IP OF KUBERNETES-MASTER and paste in the browser

| ubuntu@Kubernetes-Master:~\$ kubectl get deploy | | | | | | | |
|---|-------|------------|-----------|-----|--|--|--|
| NAME | READY | UP-TO-DATE | AVAILABLE | AGE | | | |
| customerinformation-deployment | 0/3 | 3 | Θ | 41h | | | |
| nginx-deployment | 1/5 | 5 | 1 | 10d | | | |
| nginx-salman | 0/2 | 2 | 0 | 9d | | | |
| ubuntu@Kubernetes-Master:~\$ kubectl get deploy | | | | | | | |
| NAME | READY | UP-TO-DATE | AVAILABLE | AGE | | | |
| customerinformation-deployment | 3/3 | 3 | 3 | 41h | | | |
| nginx-deployment | 5/5 | 5 | 5 | 10d | | | |
| nginx-salman | 2/2 | 2 | 2 | 9d | | | |
| ubuntu@Kubernetes-Master:~\$ | | | | | | | |

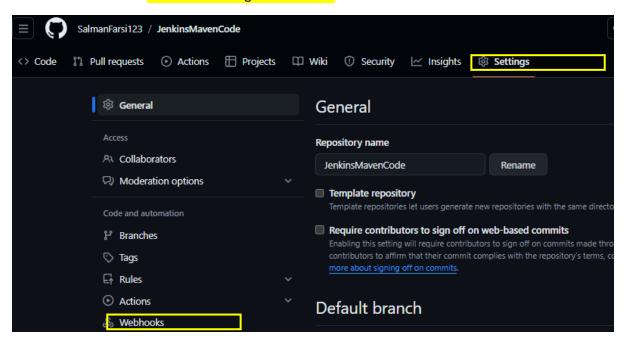
71. See the data is Available Now.



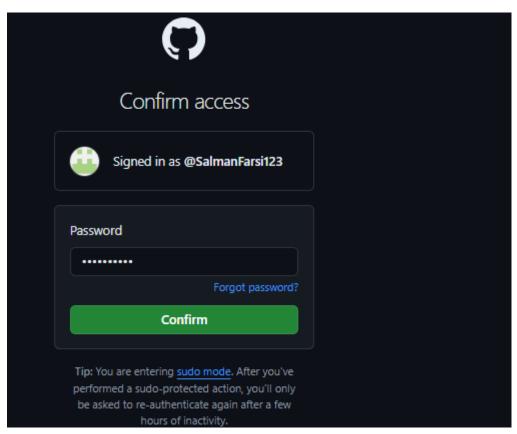
72. Enable GitHub trigger for GITScm polling, what it will do means If u made any changes and commit changes on github it will automatically run the JOB1 and it will trigger to JOB2



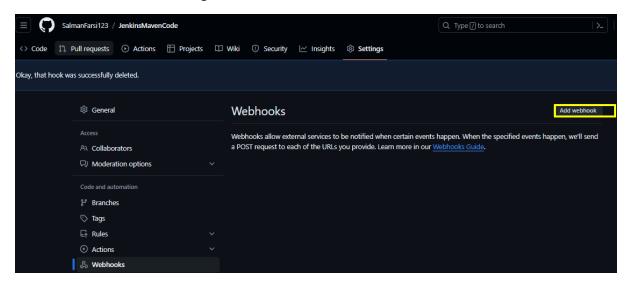
73. Go to Your GitHub Account > Settings > Webhooks



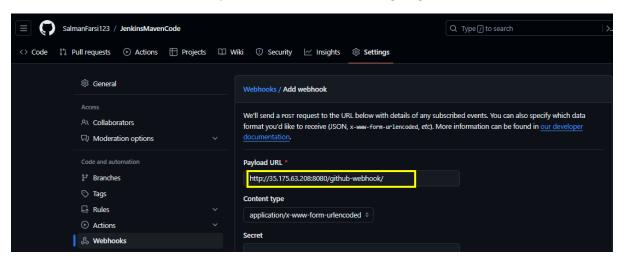
74. Sign in



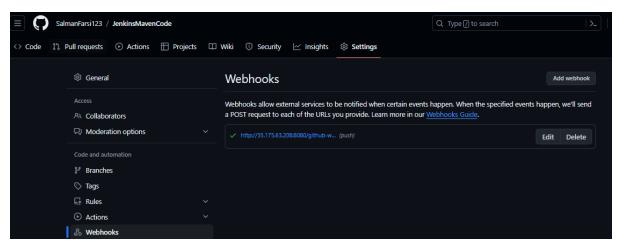
75. click Add Webhook in the right corner



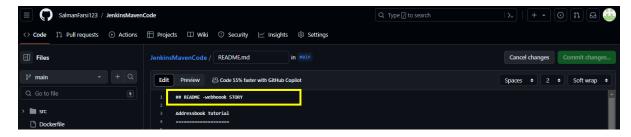
76. Paste the Jenkins Public-IP and port No of Jenkins 8080 and give github-webhook/



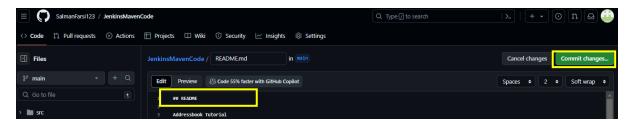
77. Add Successfully



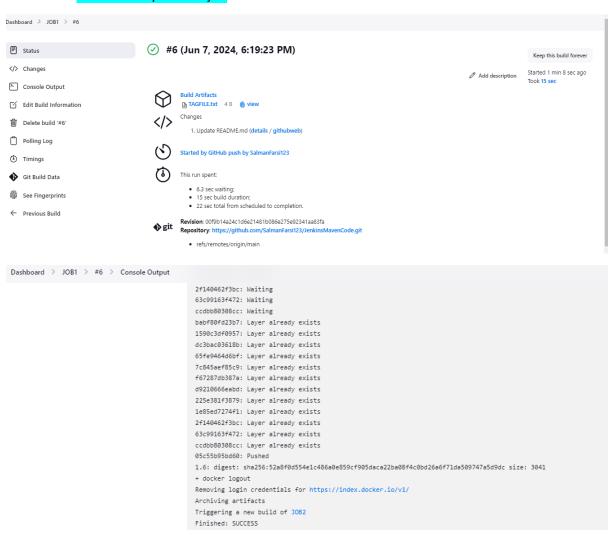
78. Now U can Make changes and I selected README.md



79. Now I removed webhook STORY and click commit changes



80. Now its automatically run the job



Dashboard > JOB1 > #6 > Console Output ✓ Console Output © ■ Status </>
Changes Started by GitHub push by SalmanFarsi123 Running as SYSTEM Console Output Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/3081
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/3081/.git # timeout=10
Fetching changes from the remote Git repository
> git toorfig remote-origin.url https://github.com/SalmanFarsi123/JenkinsMavenCode.git # timeout=10
Fetching upstream changes from https://github.com/SalmanFarsi123/JenkinsMavenCode.git # timeout=10
> git --version # timeout=10
> git --version # git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/SalmanFarsi123/JenkinsMavenCode.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/msin^(comst) # timeout=10
> git rev-parse refs/remotes/origin/msin^(comst) # timeout=10
> git rev-parse refs/remotes/origin/msin^(comst) # timeout=10
-- forcking out Revision 0e0POslan2e1de0c13808e0275e02340883Fa (refs/remotes/origin/msin) View as plain text ☑ Edit Build Information Delete build '#6' Polling Log (b) Timings / gat rev-parse reviewsormagnual (town) as timeout-ob Checking our Revision 00999b1a2a/cdo6e214818086c/27692341a883fa (refs/remotes/origin/main) > git config core.sparsecheckout # timeout-10 > git checkout -f 0099b1a2a/cdo6e21481086e275e92341a883fa # timeout-10 Git Build Data See Fingerprints Commit message: "Update README_md"

) git rev-list --no-walk b96F8dda2F805S0a140a9b86b338032379c678c92 # timeout=10

[J081] \$ /wan/jlb/jmkins/jcbols/hudson.tasks.Maven_MavenInstallation/maven/bin/mvn clean install

[INFO] Scanning for projects... ← Previous Build [MANNING] Some problems were encountered while building the effective model for com.edurekademo.tutorial:addressbook:war: [MANNING] Reporting configuration should be done in <reporting> section, not in maven-site-plugin <configuration> as reporting [WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.

Dashboard > JOB1 > #6 > Console Output

[MARNING] For this reason, future Maven versions might no longer support building such malformed projects. [INFO] [INFO] [INFO] -- clean:3.2.0:clean (default-clean) @ addressbook --- [INFO] Deleting /var/lib/jenkins/workspace/1081/target [INFO] [INFO] --- enforcer:1.0:enforce (enforce-versions) @ addressbook ---[INFO] --- resources:3.3.1:resources (default-resources) @ addressbook --[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/JOB1/src/main/resources [INFO] /var/lib/jenkins/workspace/JOB1/src/main/java/com/edurekademo/tutorial/addressbook/backend/ContactService.java:

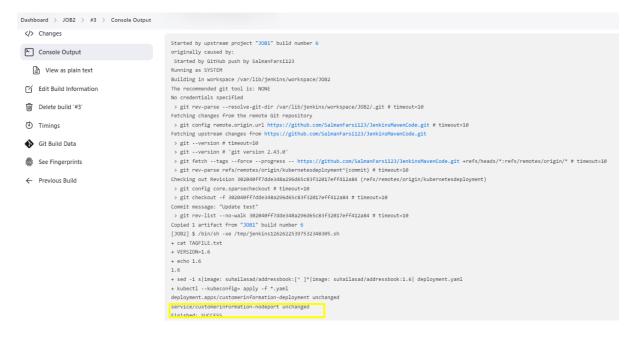
[INFO] /var/lib/jenkins/workspace/JOB1/src/main/java/com/edurekademo/tutorial/addressbook/backend/ContactService.java

[INFO] /var/lib/jenkin [INFO] --- resources:3.3.1:testResources (default-testResources) @ addressbook [INFO] Copying 1 resource from src/test/resources to target/test-classes [INFO] --- compiler:3.2:testCompile (default-testCompile) @ addressbook -[INFO] Changes detected - recompiling the module! [INFO] Compiling 5 source files to /var/lib/jenkins/workspace/JOB1/target/test-classes [INFO] /war/lib/jenkins/workspace/JOB1/src/test/jawa/com/edurekademo/utilities/TestGenericComparator.jawa:

/var/lib/jenkins/workspace/JOB1/src/test/jawa/com/edurekademo/utilities/TestGenericComparator.jawa uses or overrides a deprecated API that is marked to [INFO] /var/lib/jenkins/workspace/JOB1/src/test/jawa/com/edurekademo/utilities/TestGenericComparator.jawa uses or overrides a deprecated API that is marked to [INFO] /var/lib/jenkins/workspace/JOB1/src/test/jawa/com/edurekademo/utilities/TestGenericComparator.jawa: Recompile with -Xlint:removal for details.

```
TESTS
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found \ binding \ in \ [jar:file:/var/lib/jenkins/.m2/repository/org/slf4j/sif4j-simple/1.7.7/slf4j-simple-1.7.7.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4]: Found binding in [jar:file:/var/lib/jenkins/.m2/repository/org/slf4j/slf4j-log4j12/1.7.7/slf4j-log4j12-1.7.7.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.SimpleLoggerFactory]
Cause:at com.edurekademo.utilities.ExceptionThrower.getCounter(ExceptionThrower.java:13)
Cause:at com.edurekademo.utilities.TestLogger.testGetErrorMessage3(TestLogger.java:45)
[pool-1-thread-5] INFO com.edurekademo.utilities.TestLogger - java.lang.ArithmeticException: / by zero
Cause:at com.edurekademo.utilities.TestLogger.testGetErrorMessage1(TestLogger.java:14)
Cause:at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
[pool-1-thread-1] INFO com.edurekademo.utilities.TestLogger - java.lang.ArithmeticException: / by zero
Cause:at com.edurekademo.utilities.TestLogger.testGetErrorMessage2(TestLogger.java:30)
Cause:at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invokeθ(Native Method)
[pool-1-thread-4] ERROR com.edurekademo.utilities.ExceptionThrower - java.lang.Exception: DEAR
Cause:at com.edurekademo.utilities.ExceptionThrower.doYYY(ExceptionThrower.java:35)
Cause:at com.edurekademo.utilities.ExceptionThrower.doXXX(ExceptionThrower.java:23)
[pool-1-thread-4] INFO com.edurekademo.utilities.TestLogger - java.lang.Exception: TEST MESSAGE
{\tt Cause: at \ com.edurekademo.utilities. Exception Thrower. do Check (Exception Thrower. java: 43)}
Cause:at com.edurekademo.utilities.TestLogger.testGetErrorMessage5(TestLogger.java:75)
***** java.lang.Exception: TEST MESSAGE
Cause:at com.edurekademo.utilities.ExceptionThrower.doCheck(ExceptionThrower.java:43)
Cause:at com.edurekademo.utilities.TestLogger.testGetErrorMessage5(TestLogger.java:75)
[pool-1-thread-3] INFO com.edurekademo.utilities.TestLogger - java.io.IOException: TESTIOEXCEPTION
Cause:at com.edurekademo.utilities.ExceptionThrower.doNothing(ExceptionThrower.java:18)
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.025 sec - in com.edurekademo.utilities.TestLogger
74657374696e6720617363696920636f6e76657274696f6e20696e746f2068657861646563696d616c
testing ascii convertion into hexadecimal
```

81. and Job2 also run successfully.



If the Developer Change or Develop the Latest Version and Updating the Github Repository and it will automatically run the JOB1 and JOB2 and it will deploying the Application in Kubernetes Manifest.