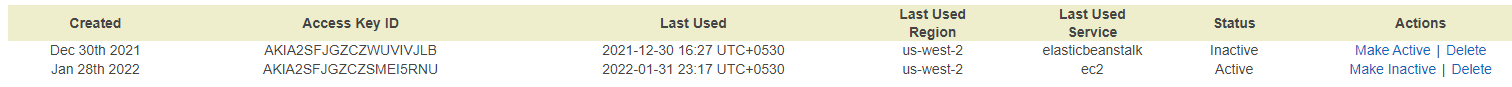
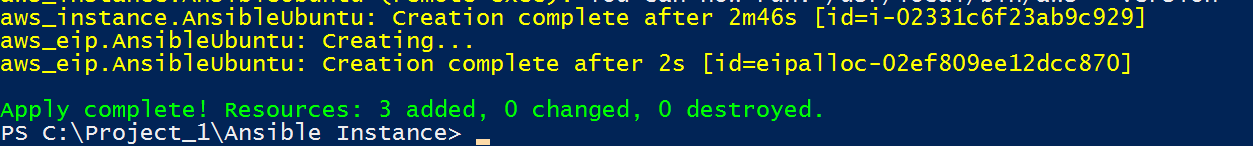
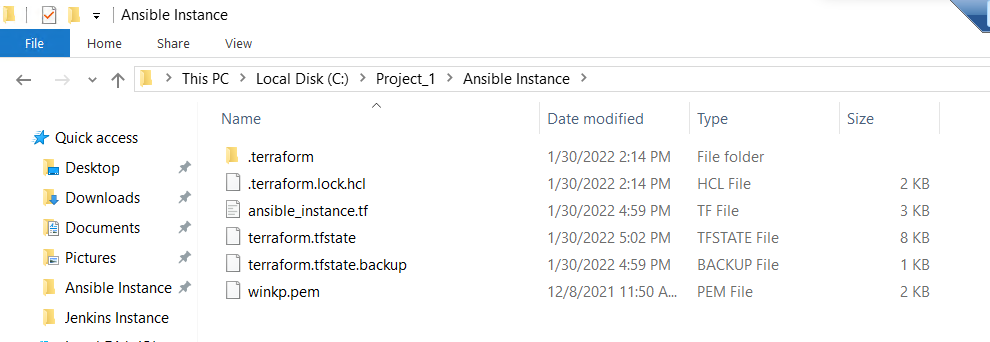
1. Created access key:



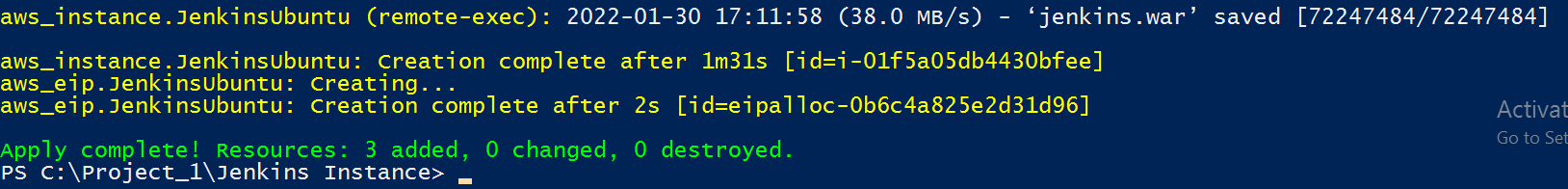
1. Created two terraform files
2. Ansible\_instance.tf: To create Ansible instance by terraform
3. Jenkins\_instance.tf: To create Jenkins instance by terraform
4. Executed Ansible instance terraform file and verified instance presence with status, instance configuration and folder structure shown as below.

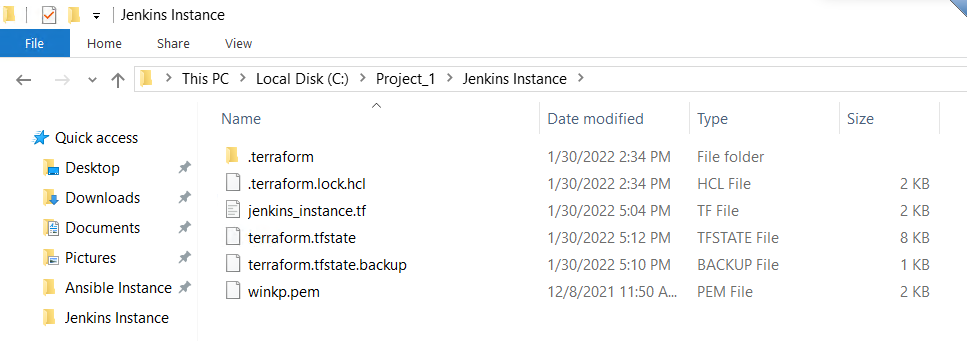






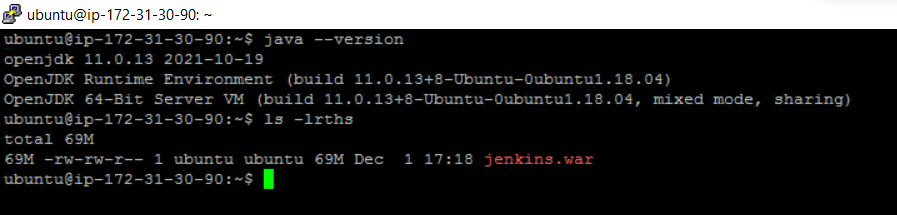
1. Executed Jenkins instance terraform file and verified instance presence with status, instance configuration and folder structure shown as below.



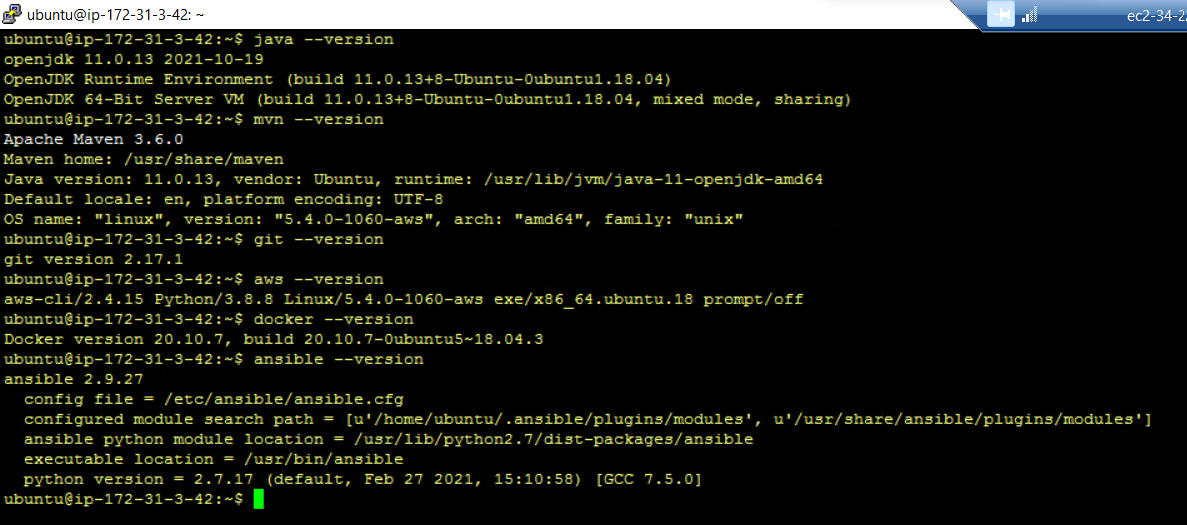




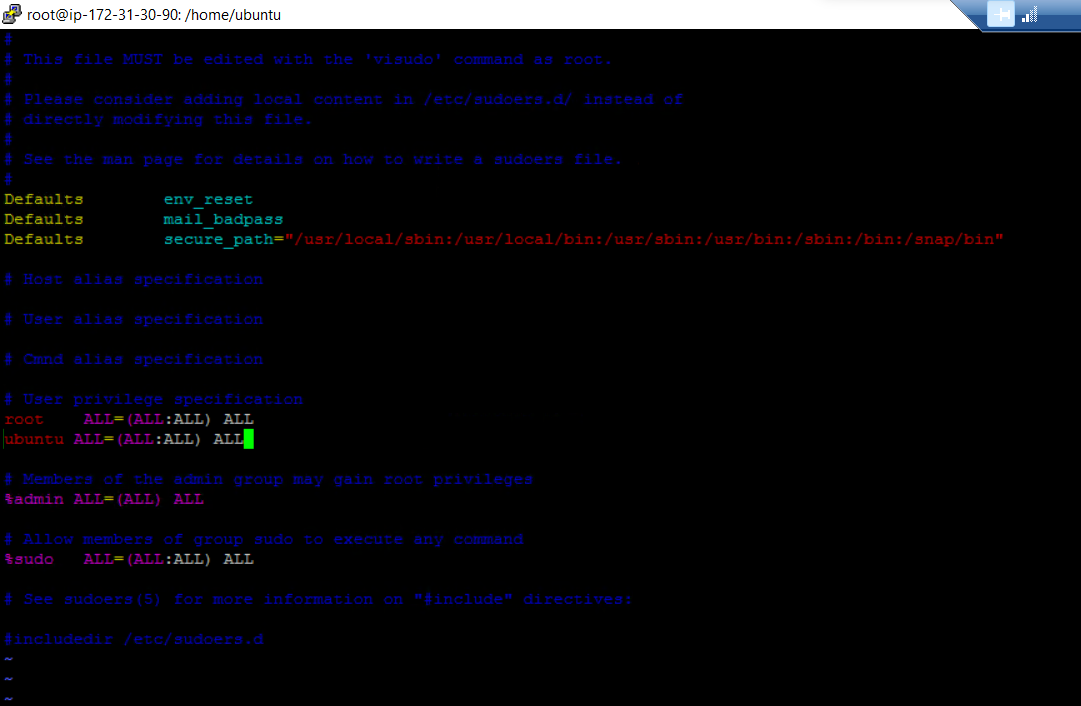
1. Logged in to Jenkins’s server and verified java and Jenkins are installed.
2. Java –version
3. Ls -lrths

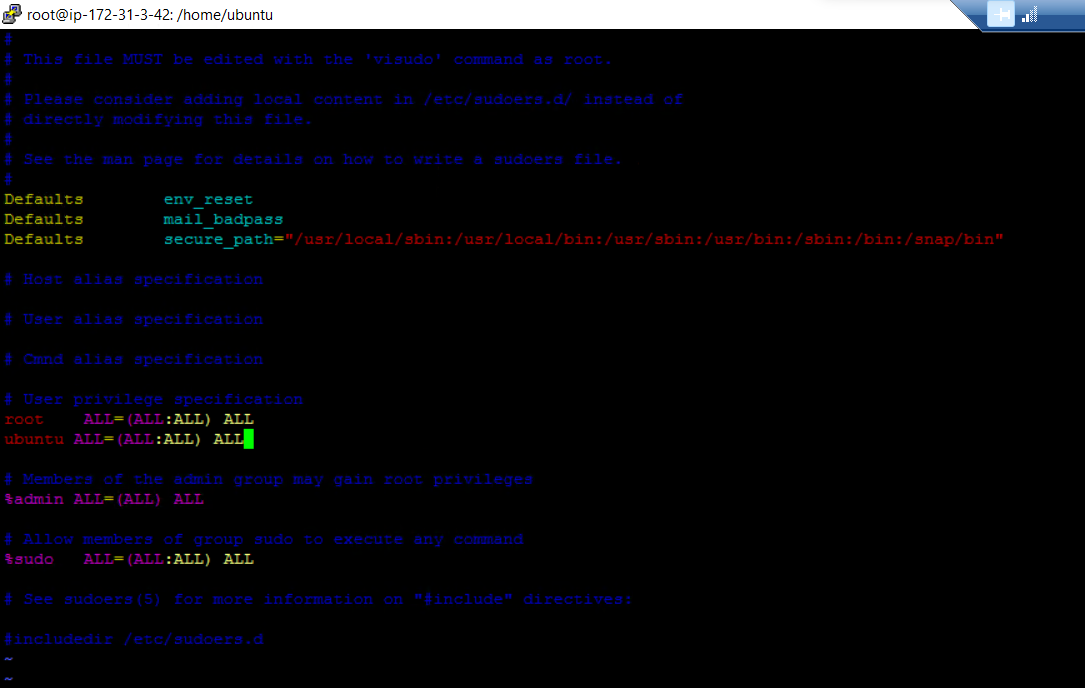


1. Logged in to Ansible server and verified java, maven, ansible, docker, AWS CLI and GIT are installed.
2. Java –version
3. Mvn –version
4. Git –version
5. Aws –version
6. Docker –version
7. Ansible –version

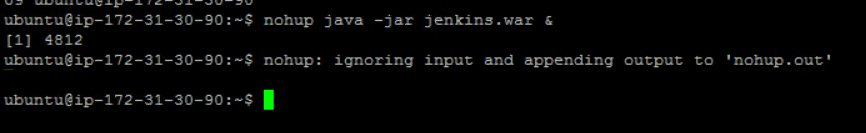


1. Ran aws configure to set up aws console authentication on ansible instance
2. Generated public and private key on ansible and Jenkins server using below command
3. Ssh-keygen
4. Added on both ansible and Jenkins server “ubuntu ALL=(ALL:ALL) ALL” line to /etc/sudoers file.



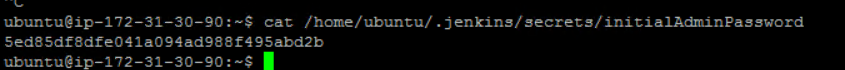


1. Added public key of Jenkins and ansible ubuntu user to ansible server “/home/ubuntu/.ssh/authorized\_keys”file
2. On Jenkins server manually started jenkins

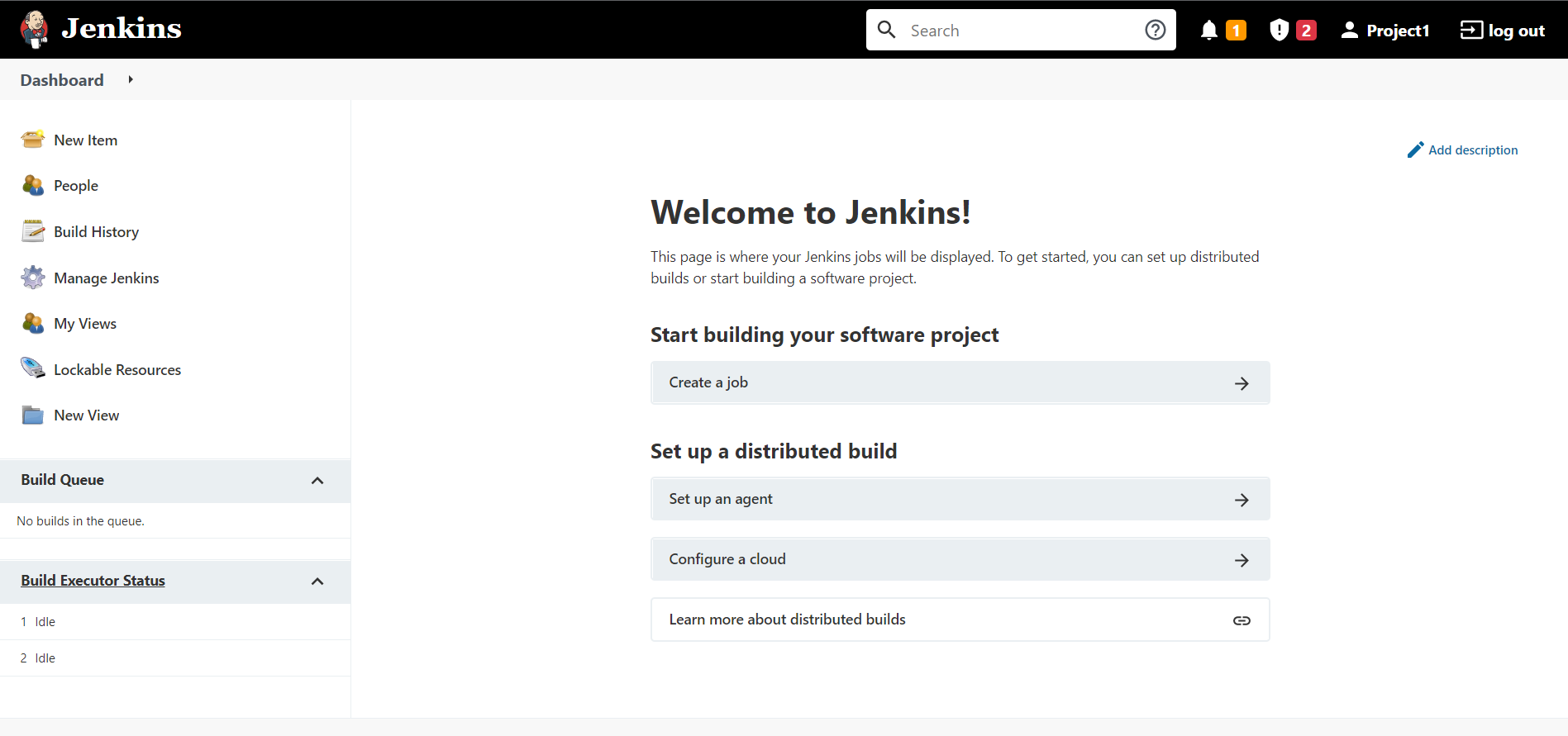


1. Collected admin password for initial set up of jenkin gui from nohup.out

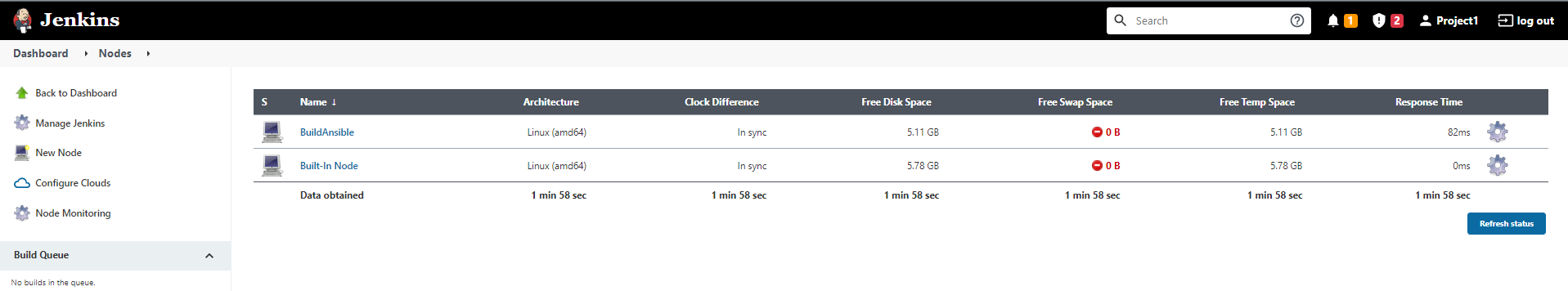
5ed85df8dfe041a094ad988f495abd2b



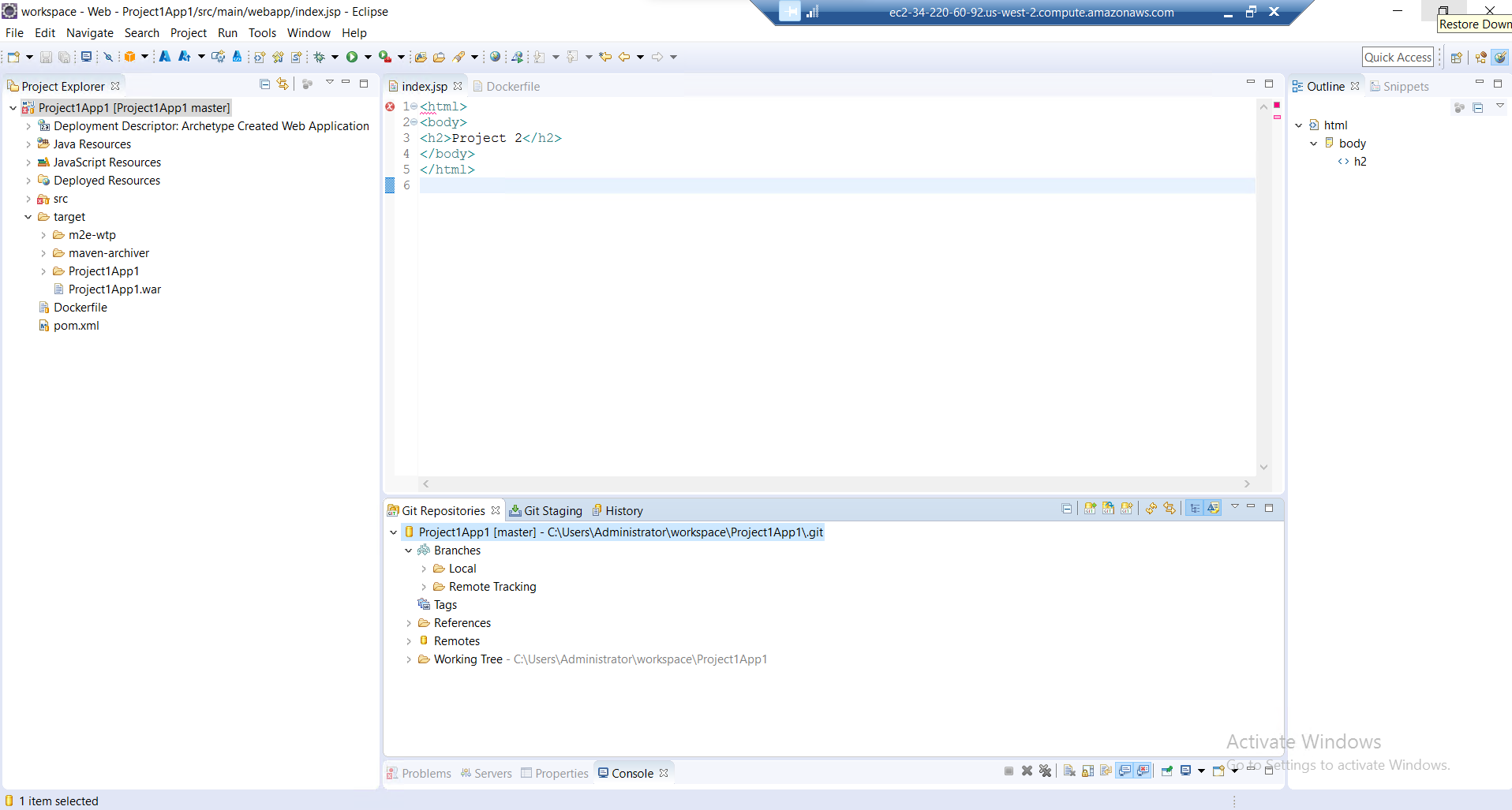
1. Installed suggested plugins, entered credentials and logged in to Jenkins GUI (http://35.83.130.222:8080/computer/new). After that installed few add-on plugins as below,
2. Ansible
3. Terraform
4. AWS Plugin
5. Docker
6. Maven Integration
7. Git Hub Integration



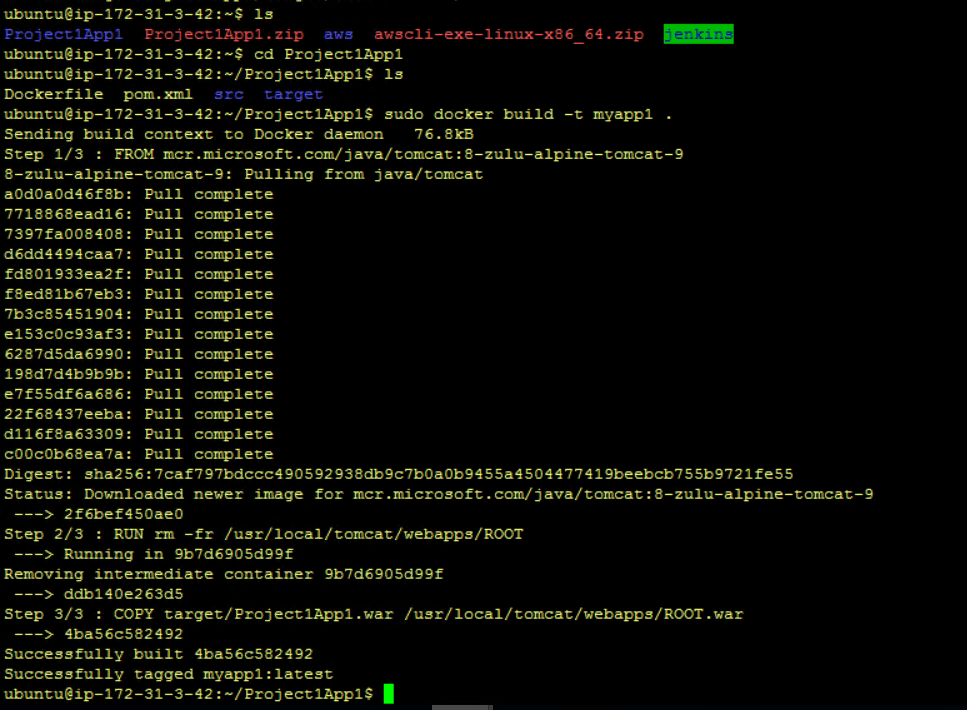
1. Created new node for pipeline execution.



1. Open Elips,
2. Created new maven project with web application arch-type
3. Added docker file with tomcat script
4. Right click on Project1app1 🡪 Runas 🡪 Maven install to generate Project1app1.war file in target folder

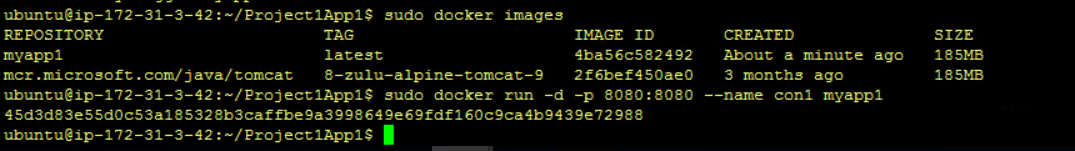


1. To Test the docker file,
2. right click on Project1app1 project 🡪 show in 🡪 system explorer
3. Compress the Project1app1 folder to zip,
4. Using WINSCP move that zip file to ansible server
5. Unzip it
6. Cd Project1app1
7. Execute command sudo docker build -t myapp1

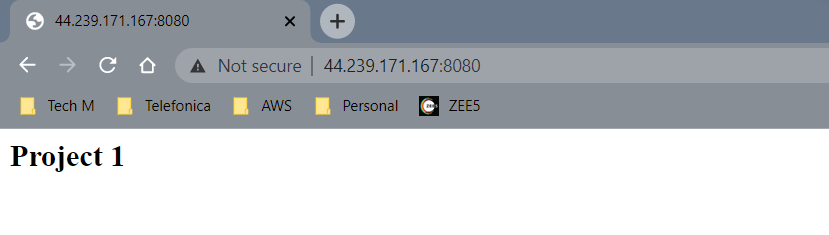


Verified images with command sudo docker images

Verified container sudo docker run -d -p 8080:8080 –name con1 myapp1

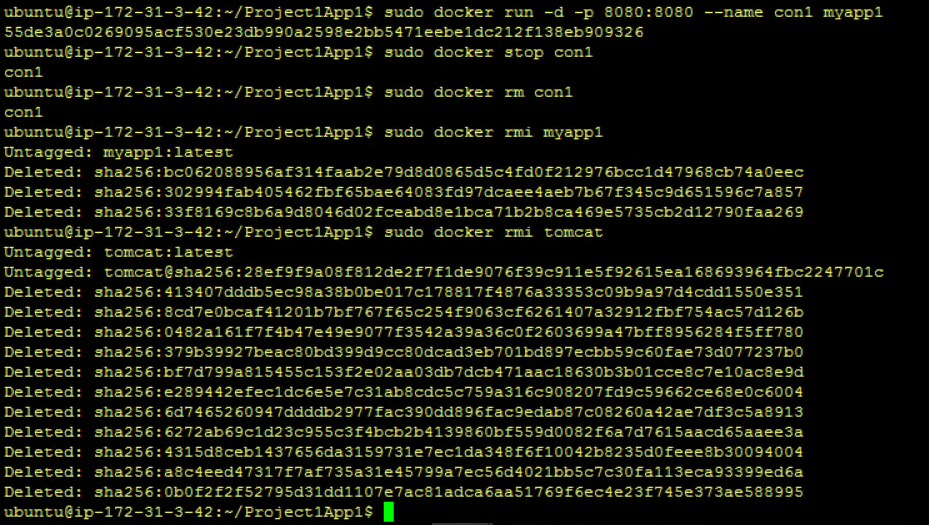


Verified with ansible local instance id, active or not

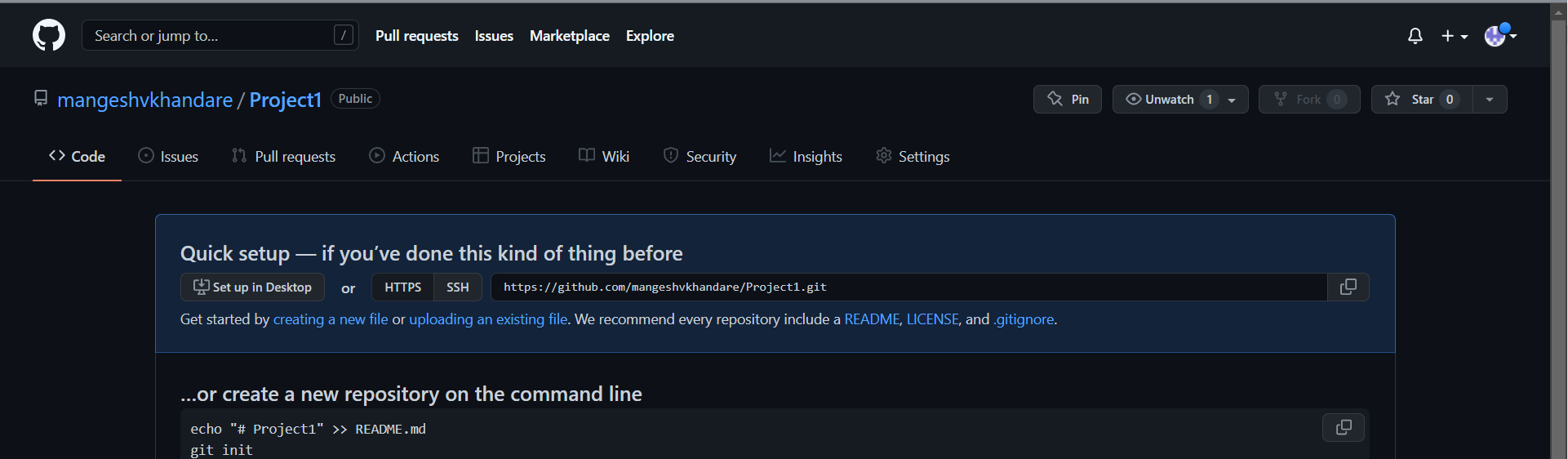


Once the test is successful, remove the container and image using below commands,

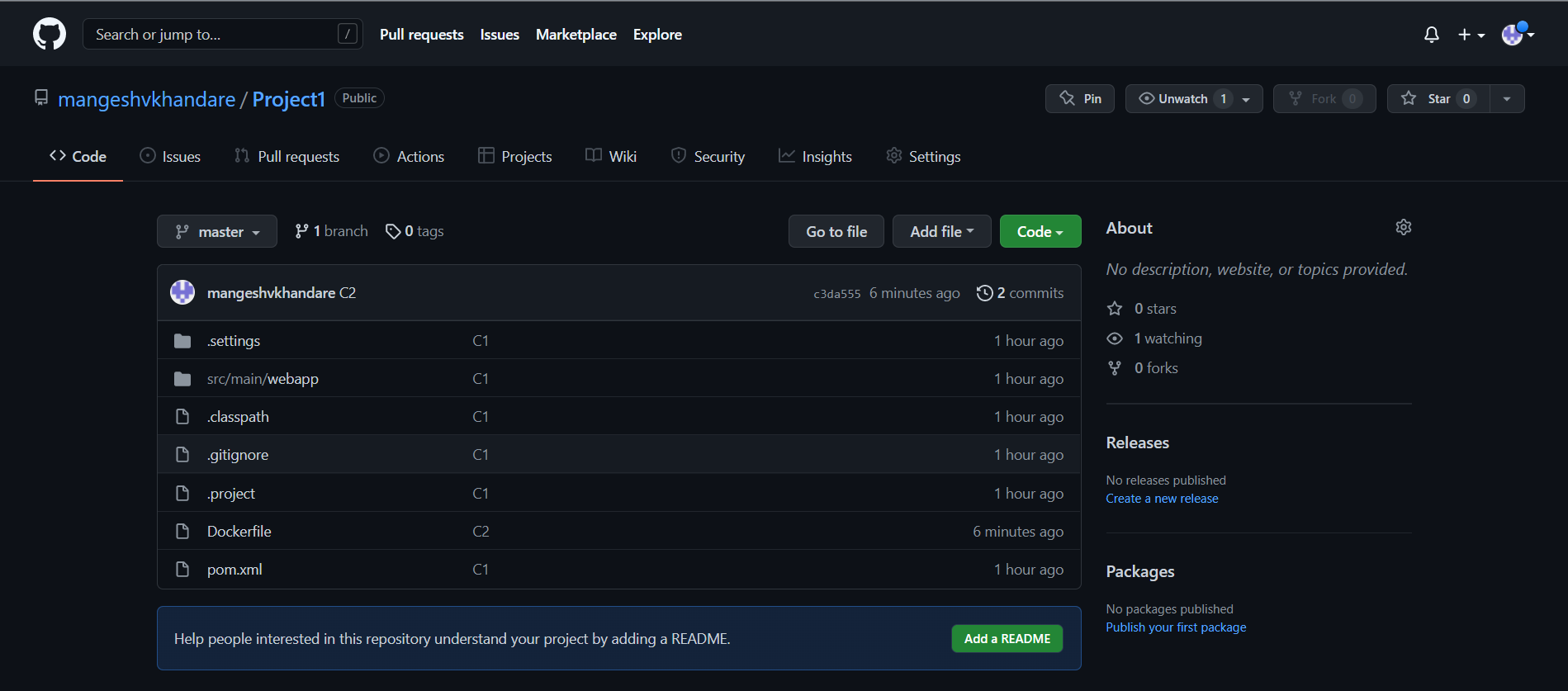
1. Sudo docker stop con1
2. Sudo docker rm con1
3. Sudo docker rmi myapp1
4. Sudo docker rmi tomcat



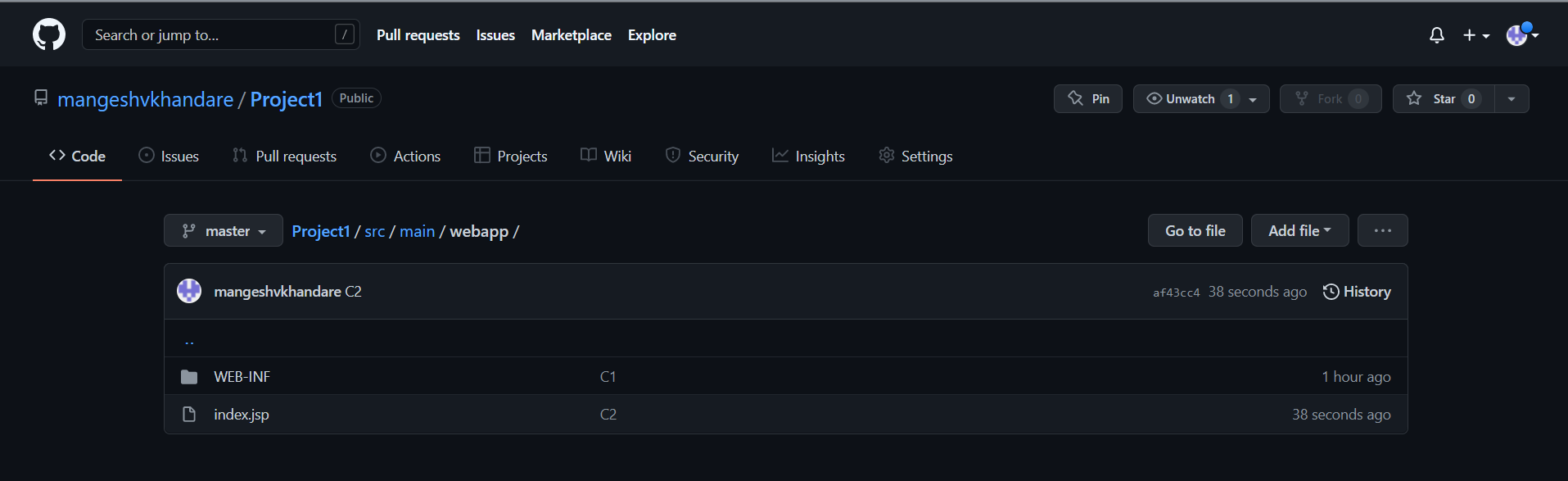
1. Created git repository,



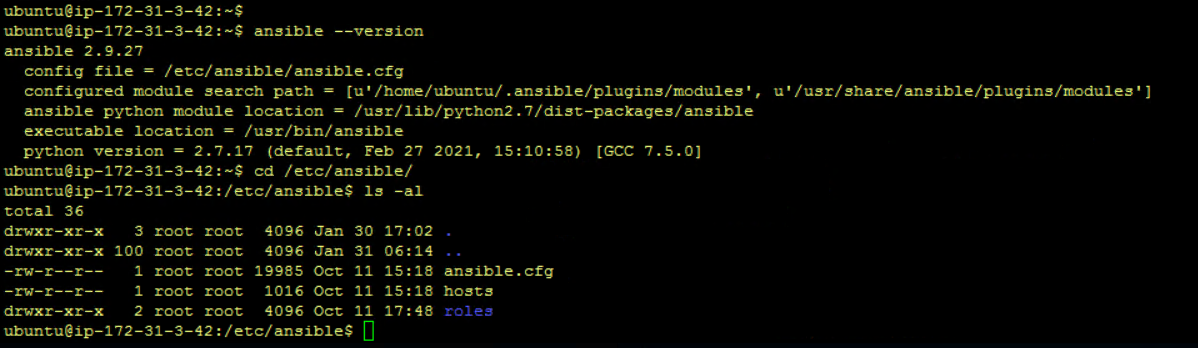
1. After commit and push, can see folder structure in git repository.



1. Verified pipeline is working by making changes in eclips index file.

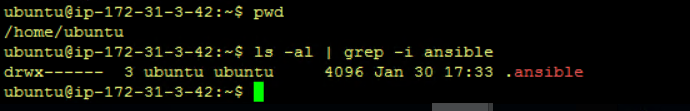


1. Ansible already installed using terraform earlier. verified using ansible –version command.

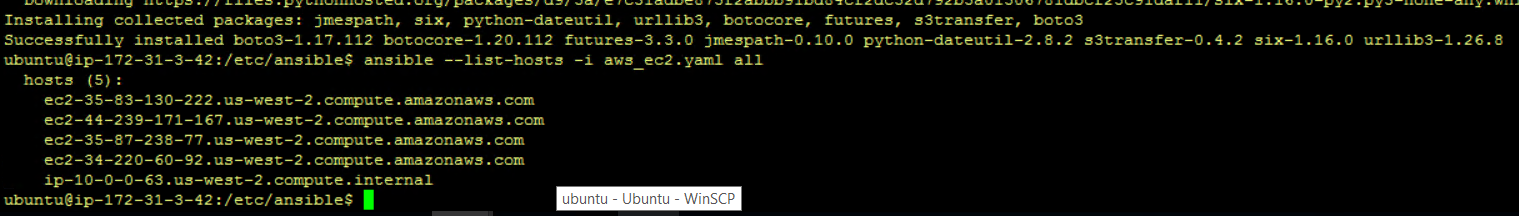


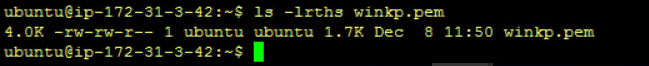
1. changed the owner ship of ansible folder from root to ubuntu

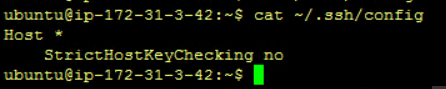




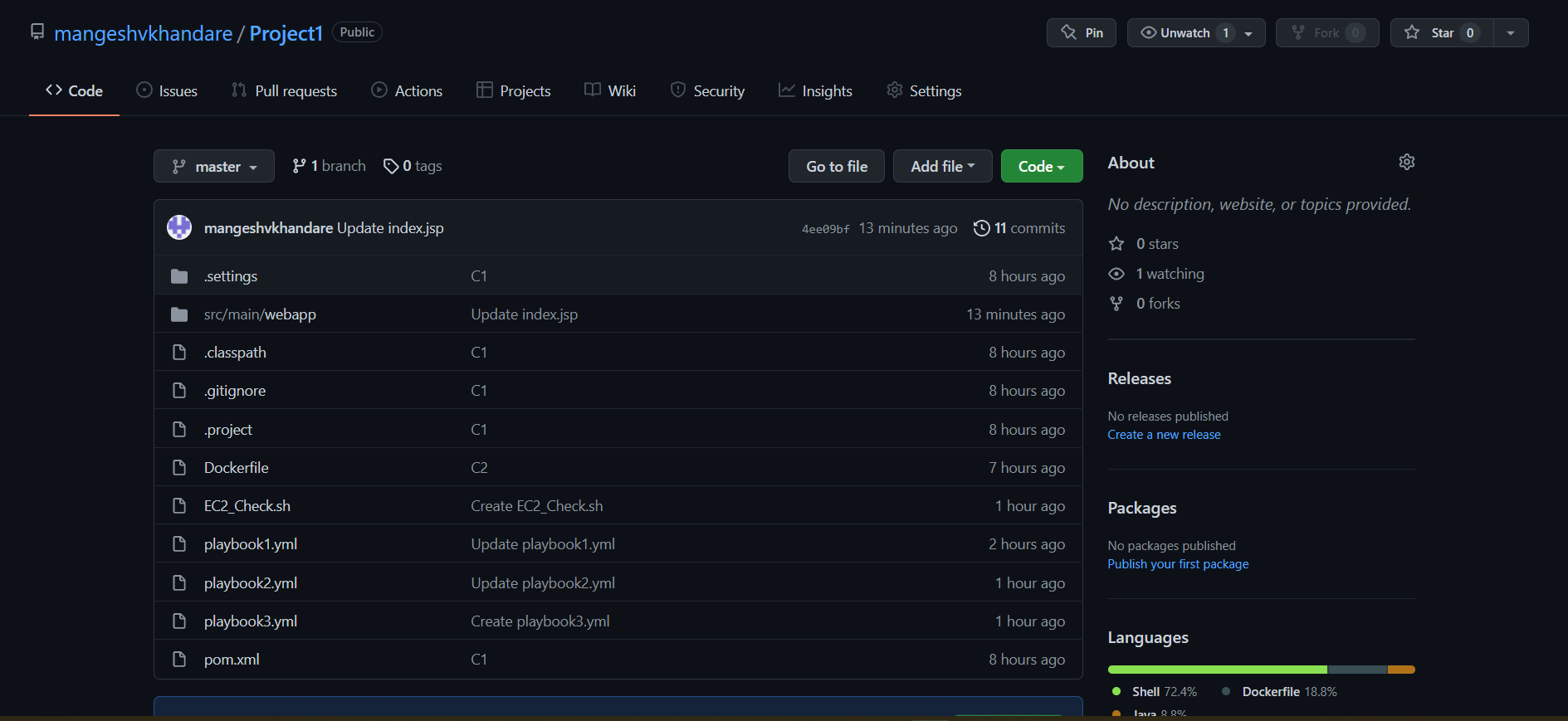
1. created aws\_ec2.yaml file to check getting list of instances of EC2 or not







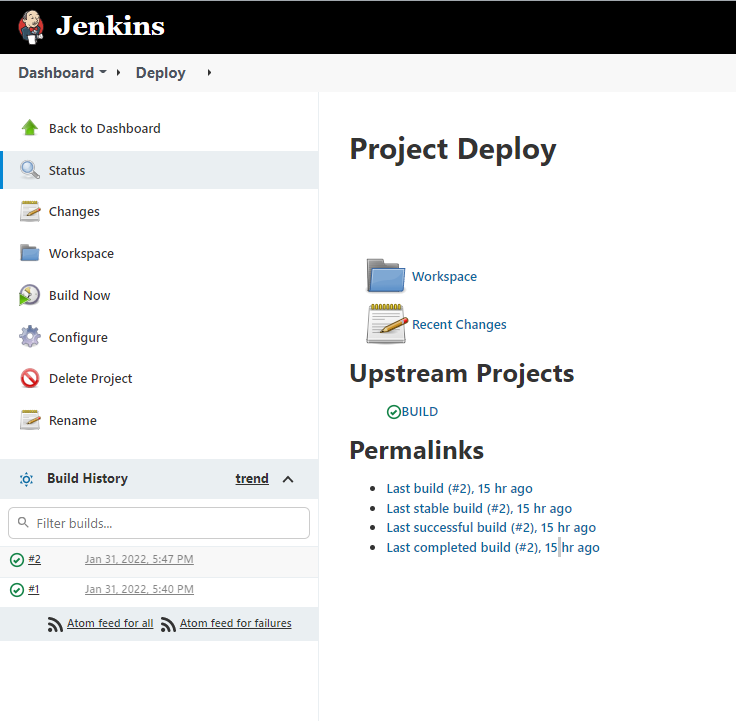
1. uploaded playbooks and scripts to git hub repository



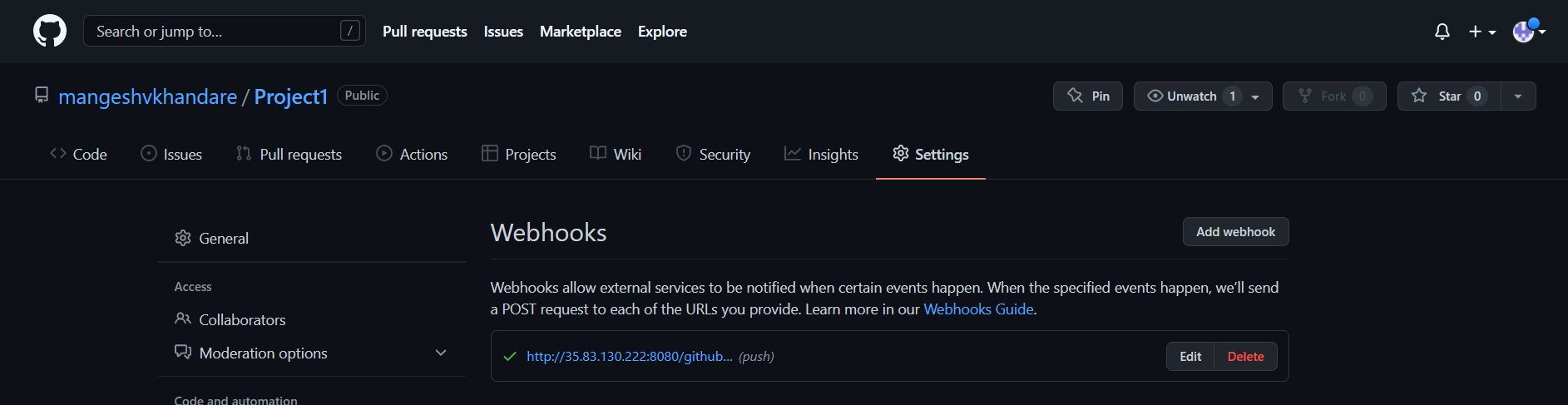
1. Created build pipeline



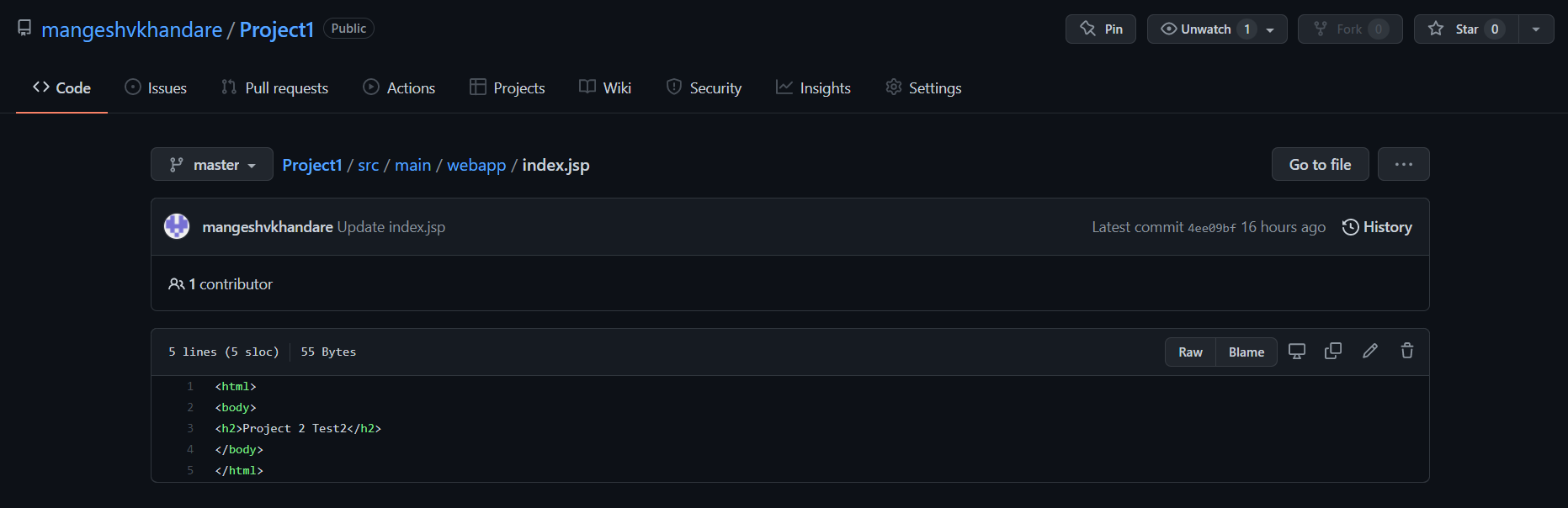
1. Created deploy pipeline



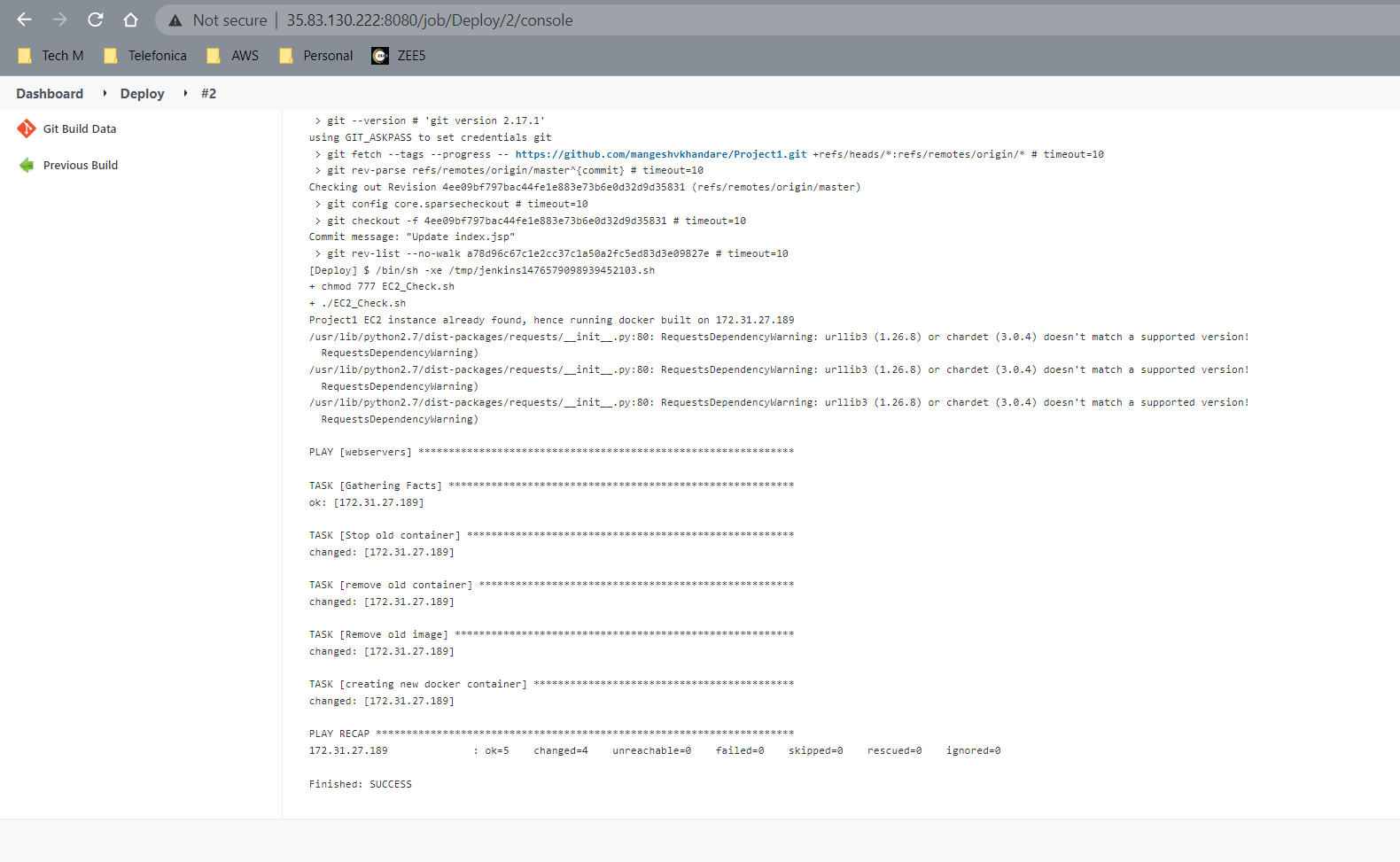
1. Added web hook for build and deploy pipeline execution,



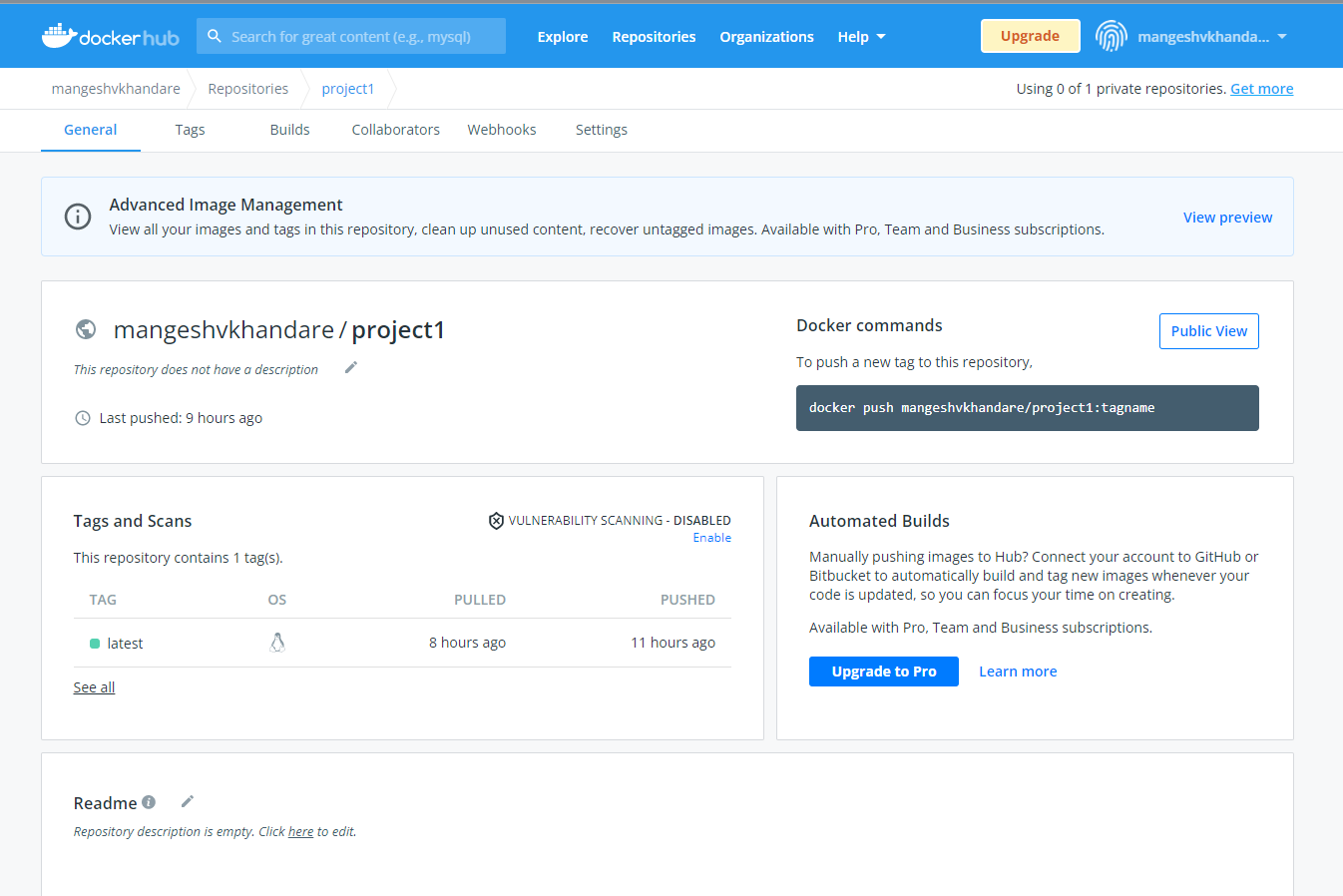
1. To test complete pipeline, made changes in index.jsp file and commited changes,



1. As clicked on commit changes button it stats to execute and it got successfully executed



1. Image got pushed to docker,



1. Same has been verified with Jenkins public id web page

