

INSTRUCTIONAL MATERIAL

DEPARTMENT OF INFORMATION TECHNOLOGY

DEVELOPING A CICD PIPELINE USING JENKINS

Ms. S. KRISHNAVENI

Assistant Professor

Department of Information Technology



TAGORE ENGINEERING COLLEGE

Rathinamangalam, Chennai-600 127

DEVELOPING A CICD PIPELINE USING JENKINS

Jenkins is a tool that is used for automation, and it is an open-source server that allows all the developers to build, test and deploy software. It works or runs on java as it is written in java. By using Jenkins we can make a continuous integration of projects(jobs) or end-to-endpoint automation.

What is Jenkins CI/CD Pipeline?

Jenkins CI/CD stands for Continuous Integration / Continuous Deployment first let us try to understand what is a pipeline. In computing, a pipeline is a set of stages or processes linked together to form a processing system. Each stage in the pipeline takes an input, processes it in accordance with a set of rules, and then sends the outputs to the stage that follows. Frequently, the pipeline's overall output is its final step's output. like the procedures outlined below

1. Test code
2. Build Application
3. Push Repository
4. Deploy to Server

All the steps mentioned above will perform in sequence one after the other if any step/stage get failed it will not move forward to another step/stage until the previous step got a success tested, and merged into a shared repository.

Benefits of Continuous Integration (CI)

- We can maintain the reports of the projects
- Deployments can be made within the given time
- Bugs can be found quickly.

What is Jenkins Continuous Deployment/Delivery (CD)?

Continuous Deployment

Continuous Deployment means automating the further stages of the pipeline automatically or manually deploying the application/code to different environments like Dev, Test, and Production. Automating the build is the main component of Continuous Integration and Continuous Deployment.

Continuous Delivery

Each and every build that passed all automated tests and was able to be fully automated and delivered into production only required one click of human intervention is called Continuous Delivery.

Jenkins Installation in Ubuntu

=====

1. `sudo apt update -y`
2. `sudo apt install openjdk-11-jre -y`
3. `curl -fsSL https://pkg.jenkins.io/debian/jenkins.io.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null`

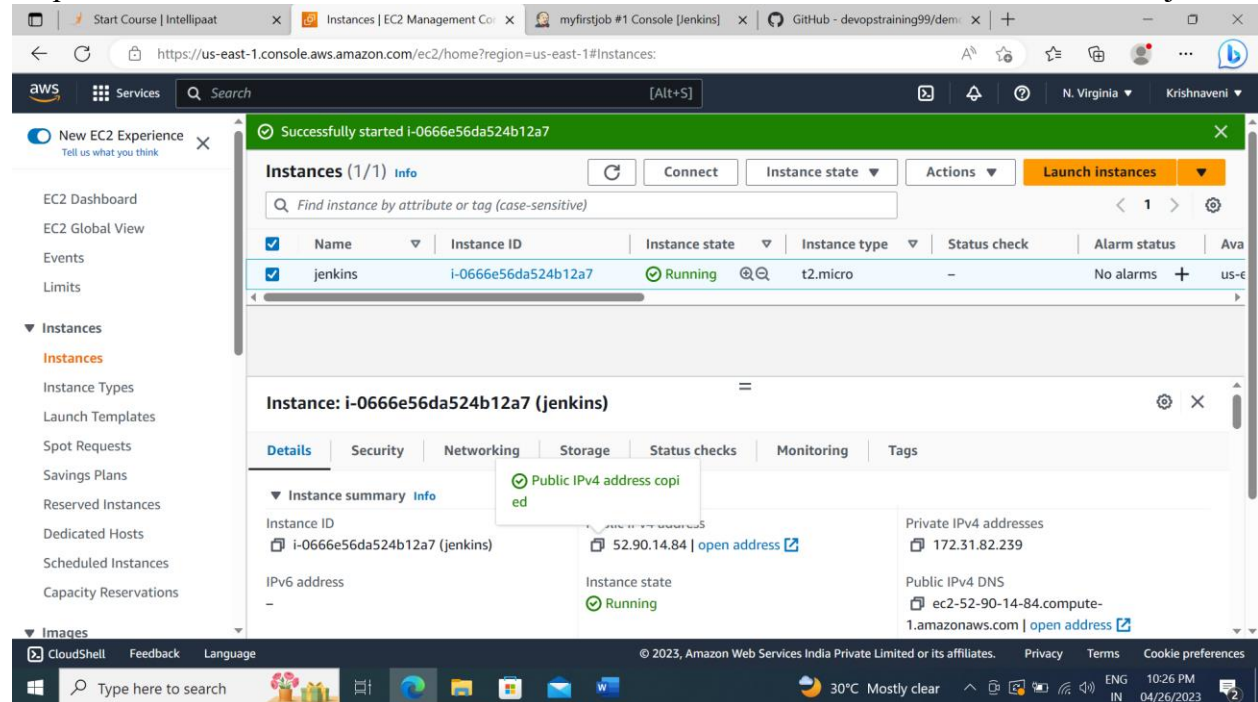
4. `echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null`
5. `sudo apt update -y`
6. `sudo apt install jenkins -y`
7. `sudo systemctl start jenkins`
8. `sudo systemctl enable jenkins`
9. `sudo systemctl status jenkins`

To access jenkins ui

http://server_ip:8080

`sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

Step:1 Create an EC2 instance and install jenkins



Step 2: Login into your Jenkins account as shown below.

Step 3. Once logged in, the user will be redirected to the Jenkins console, here's the reference for the same.



Welcome to Jenkins!

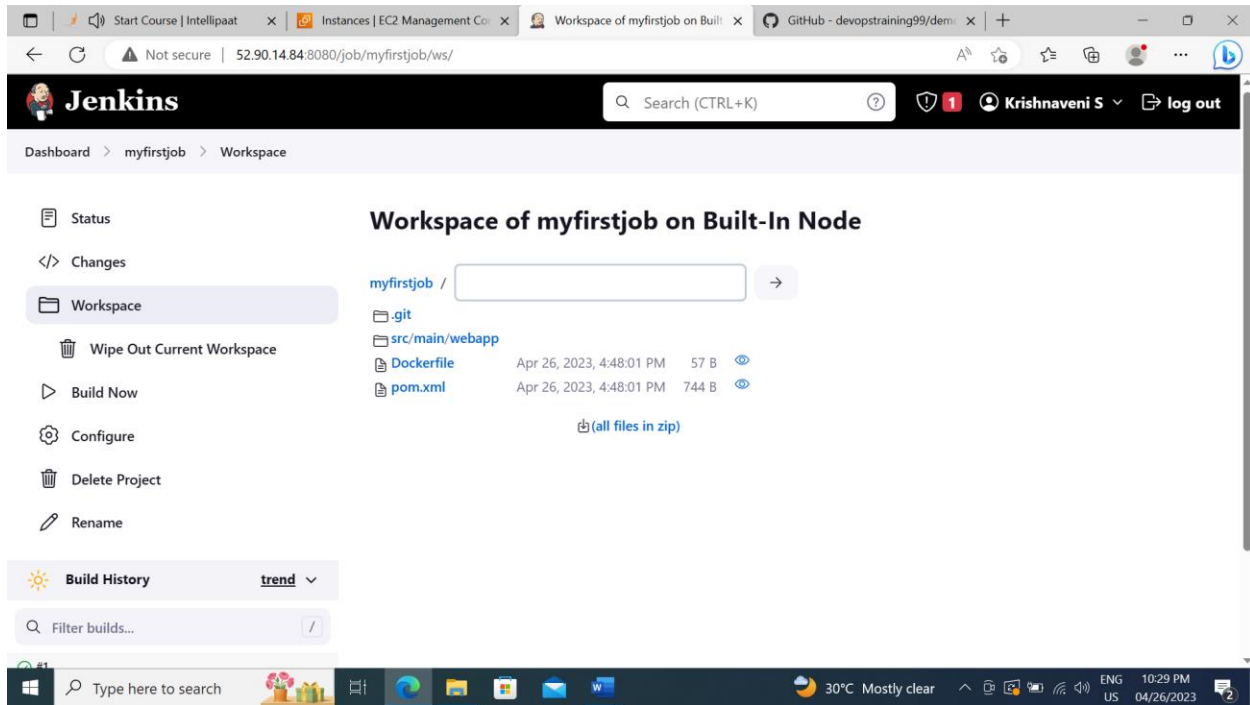
Sign in

☐ Keep me signed in

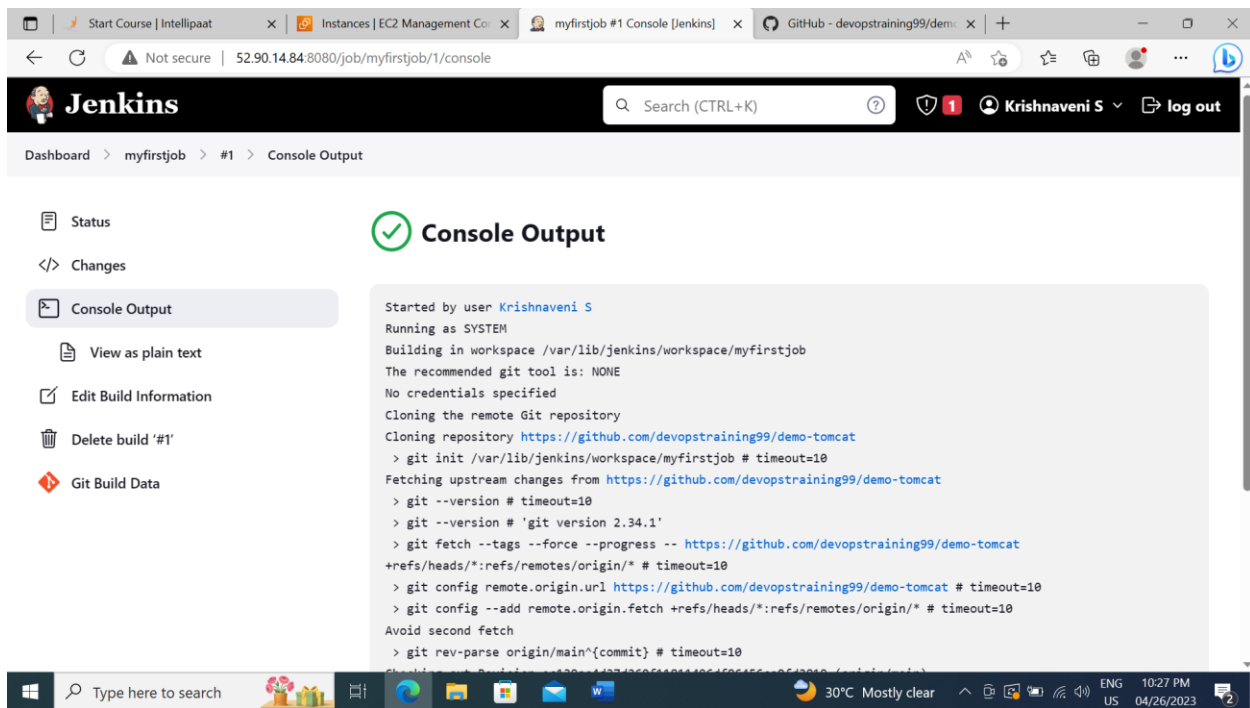
Step 4. To create a new project select the option available in the Dashboard which is “New Item” Refer to the image provided below:

The screenshot shows the Jenkins Dashboard interface. At the top, there's a navigation bar with the Jenkins logo, a search bar, and a user profile for 'Krishnaveni S' with a 'log out' button. Below the navigation bar, the 'Dashboard' section is visible. On the left sidebar, there are links for 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'My Views'. The main content area displays a table of build history for a project named 'myfirstjob'. The table has columns for 'S' (Success), 'W' (Warning), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The build history shows one successful build with a duration of 1.2 seconds. Below the table, there are links for 'Icon legend', 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. At the bottom, there's a 'Build Queue' section showing 'No builds in the queue.' and a 'Build Executor Status' section showing two idle executors.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	myfirstjob	12 min #1	N/A	1.2 sec



The screenshot shows the Jenkins 'Workspace of myfirstjob on Built-In Node' page. The left sidebar contains links for Status, Changes, Workspace (selected), Wipe Out Current Workspace, Build Now, Configure, Delete Project, and Rename. The main area displays the workspace contents: a .git directory, a src/main/webapp directory, a Dockerfile (57 B, updated Apr 26, 2023, 4:48:01 PM), and a pom.xml (744 B, updated Apr 26, 2023, 4:48:01 PM). A link to download all files as a zip is also present. The bottom of the page shows the 'Build History' section with a search filter and a table of builds.



The screenshot shows the Jenkins 'Console Output' page for build #1. The left sidebar contains links for Status, Changes, Console Output (selected), View as plain text, Edit Build Information, Delete build '#1', and Git Build Data. The main area displays the console output, which shows the build process starting by user Krishnaveni S, running as SYSTEM, and cloning the repository from https://github.com/devopstraining99/demo-tomcat. The output includes git commands for init, fetch, and rev-parse, along with their respective timeouts.

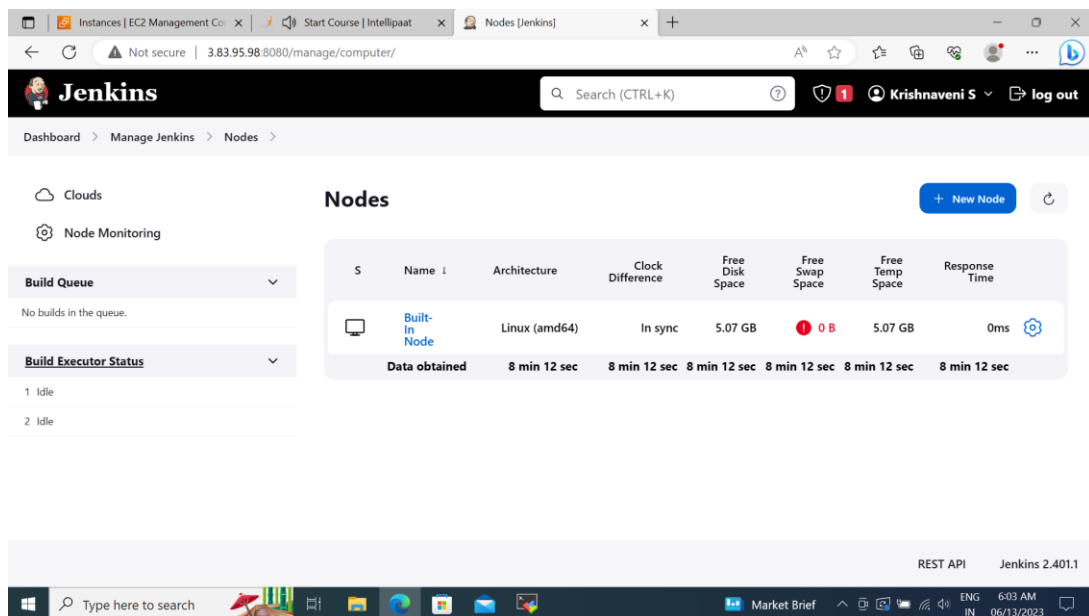
Jenkin Task1-

Tasks To Be Performed:

- 1.Trigger a pipeline using Git when push on develop branch

2. Pipeline should pull Git content to a folder

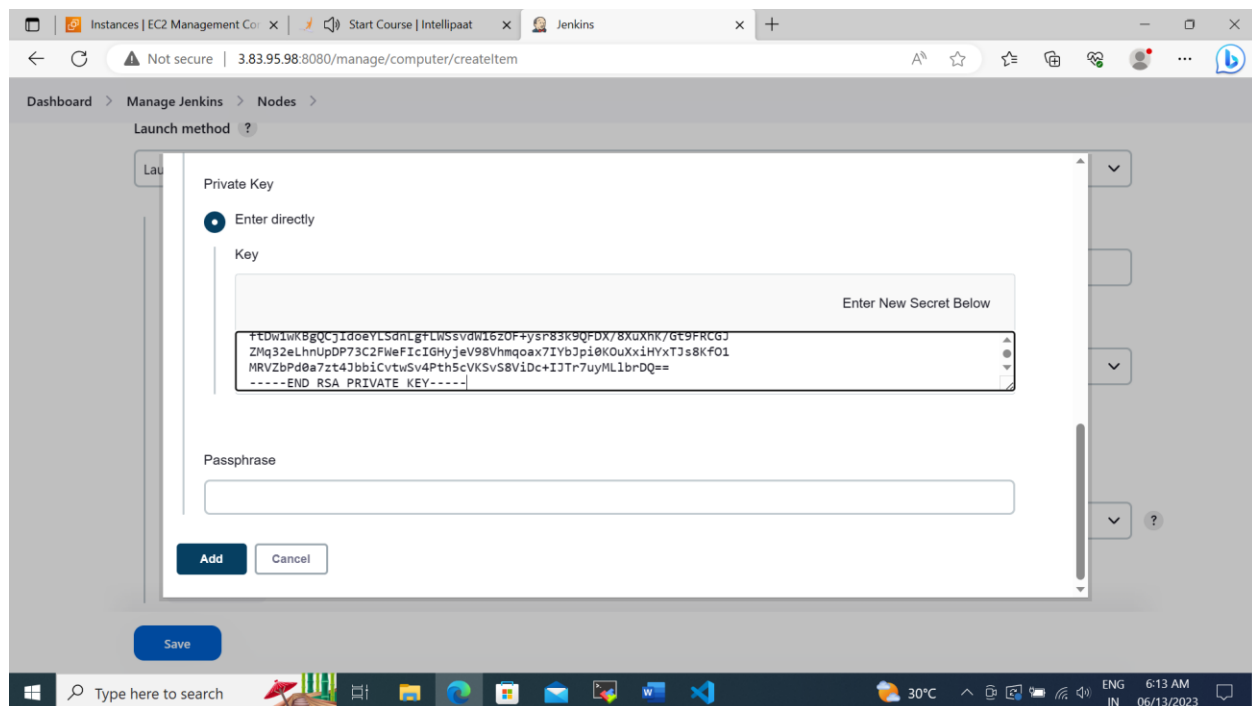
Step 1: Create a new node with the name test



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and a user profile for 'Krishnaveni S'. The breadcrumb trail is 'Dashboard > Manage Jenkins > Nodes'. On the left sidebar, there are links for 'Clouds' and 'Node Monitoring'. The main content area is titled 'Nodes' and features a '+ New Node' button. Below this is a table with columns: 'S', 'Name', 'Architecture', 'Clock Difference', 'Free Disk Space', 'Free Swap Space', 'Free Temp Space', and 'Response Time'. The table contains one entry for a 'Built-In Node' with architecture 'Linux (amd64)' and a clock difference of 'In sync'. Below the table, a 'Data obtained' row shows '8 min 12 sec' for several metrics. At the bottom, there is a 'REST API' link and the version 'Jenkins 2.401.1'.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	5.07 GB	0 B	5.07 GB	0ms
Data obtained		8 min 12 sec	8 min 12 sec	8 min 12 sec	8 min 12 sec	8 min 12 sec	8 min 12 sec

Step 2: created a test node



The screenshot shows the 'Add Private Key' dialog box in the Jenkins interface. The 'Launch method' is set to 'Enter directly'. The 'Key' field contains a long string of base64-encoded text, which is a private key. The 'Passphrase' field is empty. There are 'Add' and 'Cancel' buttons at the bottom of the dialog. The background shows the Jenkins 'Nodes' page with a 'Save' button visible.

Private Key

☒ Enter directly

Key

Enter New Secret Below

```
FTDw1wK8QCjIdoeYLSdnLgFLWSsVdW16zOF+ySf83K9QFDX/8XuXnk/Gt9FRGj  
ZMq32eLhnUpDP73C2FWeFicIGHyjeV98Vhmqaax7IYb3pi0KouXiHYxTJs8Kf01  
MRVZbPd0a7zt4JbbiCvtwSv4Pth5cVKsV8ViDc+IJTr7uyML1brDQ==  
-----END RSA PRIVATE KEY-----
```

Passphrase

Instances | EC2 Management Console | Start Course | Intellipaat | Jenkins

Not secure | 3.83.95.98:8080/manage/computer/createItem

Dashboard > Manage Jenkins > Nodes >

Launch method ?

Launch agents via SSH

Host ?

ip-172-31-93-46.ec2.internal

Credentials ?

ubuntu (credential for all my machines)

Add

Host Key Verification Strategy ?

Known hosts file Verification Strategy

Advanced

Save

Type here to search

Top... 6:13 AM 06/13/2023

Instances | EC2 Management Console | Start Course | Intellipaat | Nodes (Jenkins)

Not secure | 3.83.95.98:8080/manage/computer/

Jenkins

Search (CTRL+K)

Krishnaveni S log out

Dashboard > Manage Jenkins > Nodes >

Clouds

Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node

1 Idle

2 Idle

test

1 Idle

Nodes

19 min

+ New Node

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	5.07 GB	0 B	5.07 GB	0ms
	test		N/A	N/A	N/A	N/A	N/A
Data obtained		7 ms	19 min	19 min	19 min	19 min	19 min

REST API Jenkins 2.401.1

Type here to search

Top... 6:14 AM 06/13/2023

Step 3:master node:

master

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...

/home/ubuntu/

Name

..

.cache

.git

.local

.ssh

.bash_history

.bash_logout

.bashrc

.lessht

.profile

.sudo_as_admin_successful

.Xauthority

develop.txt

index.html

Remote monitoring

☐ Follow terminal folder

2 master

3 slave

```
ubuntu@ip-172-31-95-153:~$ history
1 sudo apt update
2 sudo apt install openjdk-11-jdk -y
3 curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.as
c > /dev/null
4 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
5 sudo apt-get update
6 sudo apt update
7 sudo apt-get install jenkins
8 cat /var/lib/jenkins/secrets/initialAdminPassword
9 sudo cat /var/lib/jenkins/secrets/initialAdminPassword
10 nano index.html
11 ls
12 git init
13 git status
14 git add index.html
15 git status
16 git commit -m "adding my first file"
17 ls
18 ssh-keygen
19 ls
20 ssh-keygen
21 cd .ssh
22 ls
23 cat id_rsa.pub
24 ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDgQ2fYEYA+bLg0wS iRXFK0G0jzM1qhxB9RnLWG6u3g0bxU6Kaep+WqJ8z5vpH51pLuQsH7PB9YcMFZ6
ujw8EmJZol3hnVn3Y/3Bz6439cuwR8CmJ9UC0h+G8L56WnZNx/v08Yib082AoIfM10FMcrU8KzCcrfDbdTQoS7eHy2Md7EF2Xi9TSqLCo7kS7vvyI1AtfP2epnbm9
ozEt/7oT9aw82XmcpwXWryJcT6GoFqn18L1svdUHZeY+qAknuKJvMMPiYAG24KSfR0ioRqE5ZEBngHeFmrKSjV8ErAVGqNdrjvbw+qGUCfLMA9Xb0gDcVz2JRqVQOV
xqd9P8ZNU0cTMv0Rm
25 cd
26 ls
27 git remote add origin git@github.com:krishnaveni1212/gittask.git
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Type here to search

Watchlist ideas

ENG US 2:01 PM 06/15/2023

master

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...

/home/ubuntu/

Name

..

.cache

.git

.local

.ssh

.bash_history

.bash_logout

.bashrc

.lessht

.profile

.sudo_as_admin_successful

.Xauthority

develop.txt

index.html

Remote monitoring

☐ Follow terminal folder

2 master

3 slave

```
11 ls
12 git init
13 git status
14 git add index.html
15 git status
16 git commit -m "adding my first file"
17 ls
18 ssh-keygen
19 ls
20 ssh-keygen
21 cd .ssh
22 ls
23 cat id_rsa.pub
24 ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDgQ2fYEYA+bLg0wS iRXFK0G0jzM1qhxB9RnLWG6u3g0bxU6Kaep+WqJ8z5vpH51pLuQsH7PB9YcMFZ6
ujw8EmJZol3hnVn3Y/3Bz6439cuwR8CmJ9UC0h+G8L56WnZNx/v08Yib082AoIfM10FMcrU8KzCcrfDbdTQoS7eHy2Md7EF2Xi9TSqLCo7kS7vvyI1AtfP2epnbm9
ozEt/7oT9aw82XmcpwXWryJcT6GoFqn18L1svdUHZeY+qAknuKJvMMPiYAG24KSfR0ioRqE5ZEBngHeFmrKSjV8ErAVGqNdrjvbw+qGUCfLMA9Xb0gDcVz2JRqVQOV
xqd9P8ZNU0cTMv0Rm
25 cd
26 ls
27 git remote add origin git@github.com:krishnaveni1212/gittask.git
28 git remote -v
29 git branch
30 git branch develop
31 git branch
32 git checkout develop
33 git branch
34 nano develop.txt
35 ls
36 gir status
37 git status
38 git add develop.txt
39 git commit -m "adding develop.txt is added to the develop branch"
40 ls
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Type here to search

Watchlist ideas

ENG US 2:01 PM 06/15/2023

Step4: created webhook

The image shows two screenshots. The top screenshot is a web browser view of the GitHub repository settings for 'krishnaveni1212 / gittask'. The 'Webhooks' tab is selected, showing a list of webhooks with one entry: 'http://44.211.46.241:8080/github-w... (push)'. The bottom screenshot is a terminal window running on a remote host (ip-172-31-95-153). It shows the process of setting up a local development environment with Git, including creating a 'develop' branch and pushing it to the remote repository. The terminal output includes commands like 'git init', 'git branch', 'git push', and 'git pull', along with their respective outputs and warnings.

GitHub Repository Settings (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](#).

✓ [http://44.211.46.241:8080/github-w... \(push\)](#) Edit Delete

Terminal Output

```
On branch develop
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .Xauthority
        .bash_history
        .bash_logout
        .bashrc
        .cache/
        .lessshst
        .profile
        .ssh/
        .sudo_as_admin_successful

nothing added to commit but untracked files present (use "git add" to track)
ubuntu@ip-172-31-95-153:~$ ls
develop.txt  index.html
ubuntu@ip-172-31-95-153:~$ git branch
* develop
master
ubuntu@ip-172-31-95-153:~$ git push origin develop
The authenticity of host 'github.com (140.82.112.3)' can't be established.
ED25519 key fingerprint is SHA256:+D1Y3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 663 bytes | 663.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:krishnaveni1212/gittask.git
 * [new branch]      develop -> develop
ubuntu@ip-172-31-95-153:~$
```

Instances | EC2 Management Co x Start Course | Intellipaat x Webhooks x assignment1 [Jenkins] x

Not secure | 44.211.46.241:8080/job/assignment1/

Dashboard > assignment1 >

running assignment1 task that pull all data from git to local test machine whenever develop branch is pushed

Workspace

Build Now

Configure

Delete Project

GitHub Hook Log

GitHub

Rename

Edit description

Disable Project

Permalinks

- Last build (#1), 1 min 28 sec ago
- Last stable build (#1), 1 min 28 sec ago
- Last successful build (#1), 1 min 28 sec ago
- Last completed build (#1), 1 min 28 sec ago

Build History trend

Filter builds...

#1 Jun 15, 2023, 8:38 AM

Atom feed for all Atom feed for failures

REST API Jenkins 2.401.1

Step 5: files are pushed by slave node from github

slave

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...

/home/ubuntu/

Name

- ..
- .cache
- .ssh
- jenkins
- .bash_history
- .bash_logout
- .bashrc
- .profile
- .sudo_as_admin_successful
- .Xauthority

Remote monitoring

Follow terminal folder

Usage of /: 30.2% of 7.57GB Users logged in: 0

Memory usage: 31% IPv4 address for eth0: 172.31.93.46

Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

21 updates can be applied immediately.

2 of these updates are standard security updates.

To see these additional updates run: `apt list --upgradable`

Enable ESM Apps to receive additional future security updates.

See <https://ubuntu.com/esm> or run: `sudo pro status`

*** System restart required ***

Last login: Thu Jun 15 08:29:39 2023 from 106.198.33.101

ubuntu@ip-172-31-93-46:~\$ ls

jenkins

ubuntu@ip-172-31-93-46:~\$ cd jenkins

ubuntu@ip-172-31-93-46:~/jenkins\$ ls

remoting remoting.jar workspace

ubuntu@ip-172-31-93-46:~/jenkins\$ cd workspace

ubuntu@ip-172-31-93-46:~/jenkins/workspace\$ ls

assignment1

ubuntu@ip-172-31-93-46:~/jenkins/workspace\$ cd assignment1

-bash: cd: assignment1: No such file or directory

ubuntu@ip-172-31-93-46:~/jenkins/workspace\$ cd assignment1

ubuntu@ip-172-31-93-46:~/jenkins/workspace/assignment1\$ ls

develop.txt index.html

ubuntu@ip-172-31-93-46:~/jenkins/workspace/assignment1\$ cat develop.txt

this is the develop file having all merged data from other feature branches

ubuntu@ip-172-31-93-46:~/jenkins/workspace/assignment1\$

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Jenkin Task2-

Tasks To Be Performed:

1. Add 2 nodes to Jenkins master
2. Create 2 jobs with the following jobs: a. Push to test b. Push to prod
3. Once a push is made to test branch, copy Git files to test server
4. Once a push is made to master branch, Copy gitfiles to prod server

Step 1: Create a three ec2 instances namely master, test and prod.

Step 2: created prod node and test node in jenkins

The screenshot shows the Jenkins 'Nodes' page. The left sidebar contains 'Build Queue' and 'Build Executor Status'. The main area displays a table of nodes:

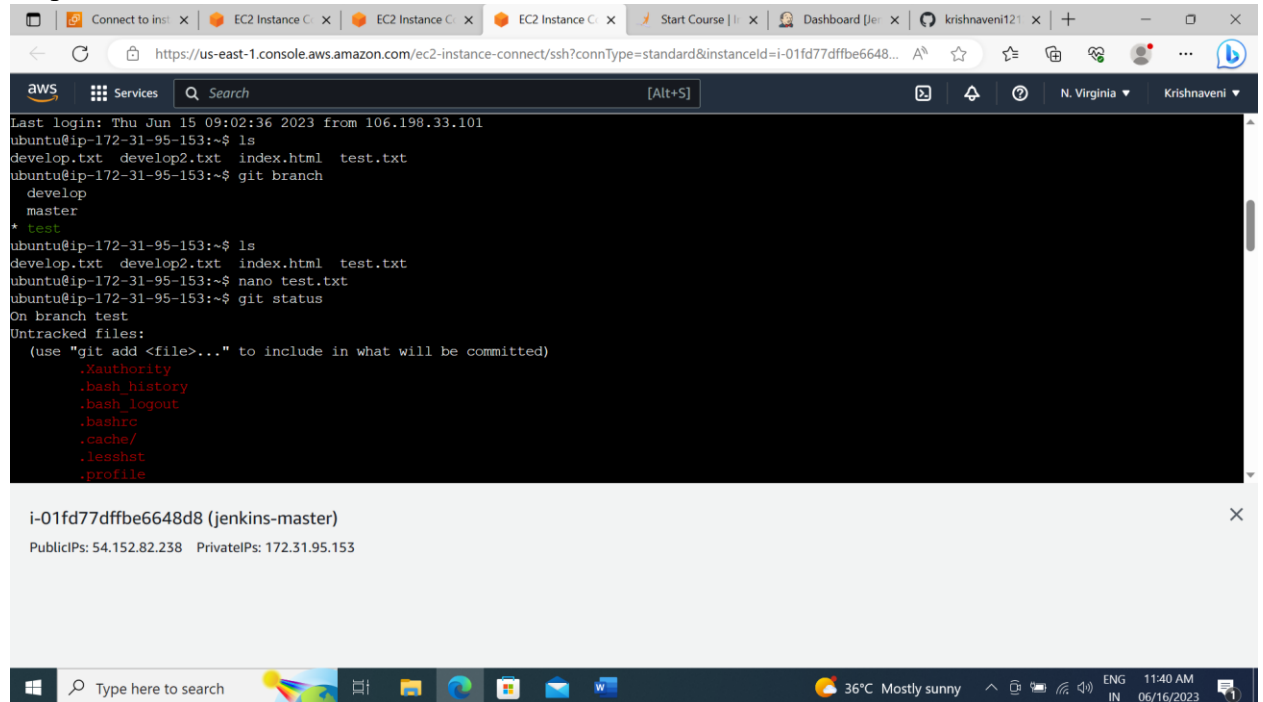
S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.69 GB	1.0 B	4.69 GB	0ms
	prod	Linux (amd64)	In sync	5.35 GB	1.0 B	5.35 GB	102ms
	test	Linux (amd64)	In sync	4.90 GB	1.0 B	4.90 GB	23ms
Data obtained			54 sec	54 sec	54 sec	54 sec	54 sec

Step 3: created two jobs push to prod and push to test in Jenkins

The screenshot shows the Jenkins 'Dashboard' page. The left sidebar contains 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main area displays a table of job history:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	assignment1	21 hr #2	N/A	0.58 sec
✓	☁	push to prod	1 min 12 sec #2	5 min 47 sec #1	3.6 sec
✓	☀	push to test	5 min 47 sec #1	N/A	4.7 sec

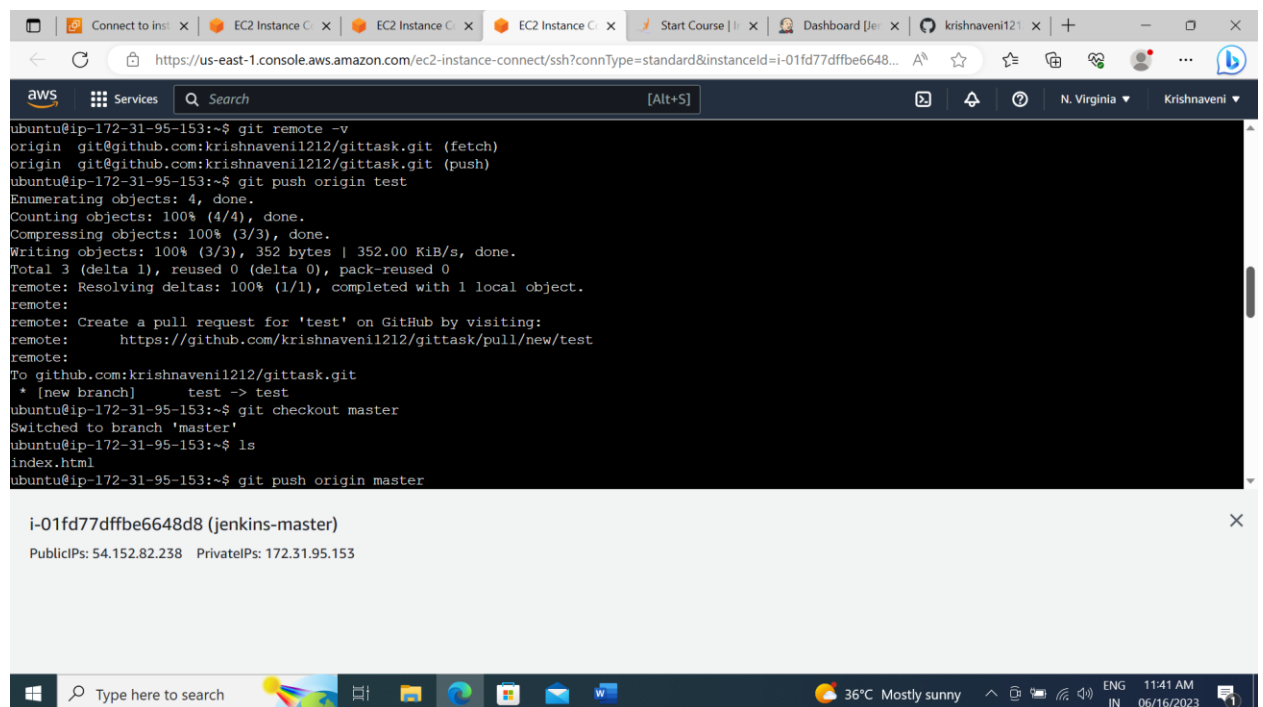
Step4: commands at master node



The screenshot shows the AWS Management Console terminal for an EC2 instance. The terminal output is as follows:

```
Last login: Thu Jun 15 09:02:36 2023 from 106.198.33.101
ubuntu@ip-172-31-95-153:~$ ls
develop.txt  develop2.txt  index.html  test.txt
ubuntu@ip-172-31-95-153:~$ git branch
  develop
* master
+ test
ubuntu@ip-172-31-95-153:~$ ls
develop.txt  develop2.txt  index.html  test.txt
ubuntu@ip-172-31-95-153:~$ nano test.txt
ubuntu@ip-172-31-95-153:~$ git status
On branch test
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .Xauthority
        .bash_history
        .bash_logout
        .bashrc
        .cache/
        .lessht
        .profile
```

Below the terminal, a metadata box for the instance `i-01fd77dffbe6648d8 (jenkins-master)` is visible, showing Public IPs: 54.152.82.238 and Private IPs: 172.31.95.153.

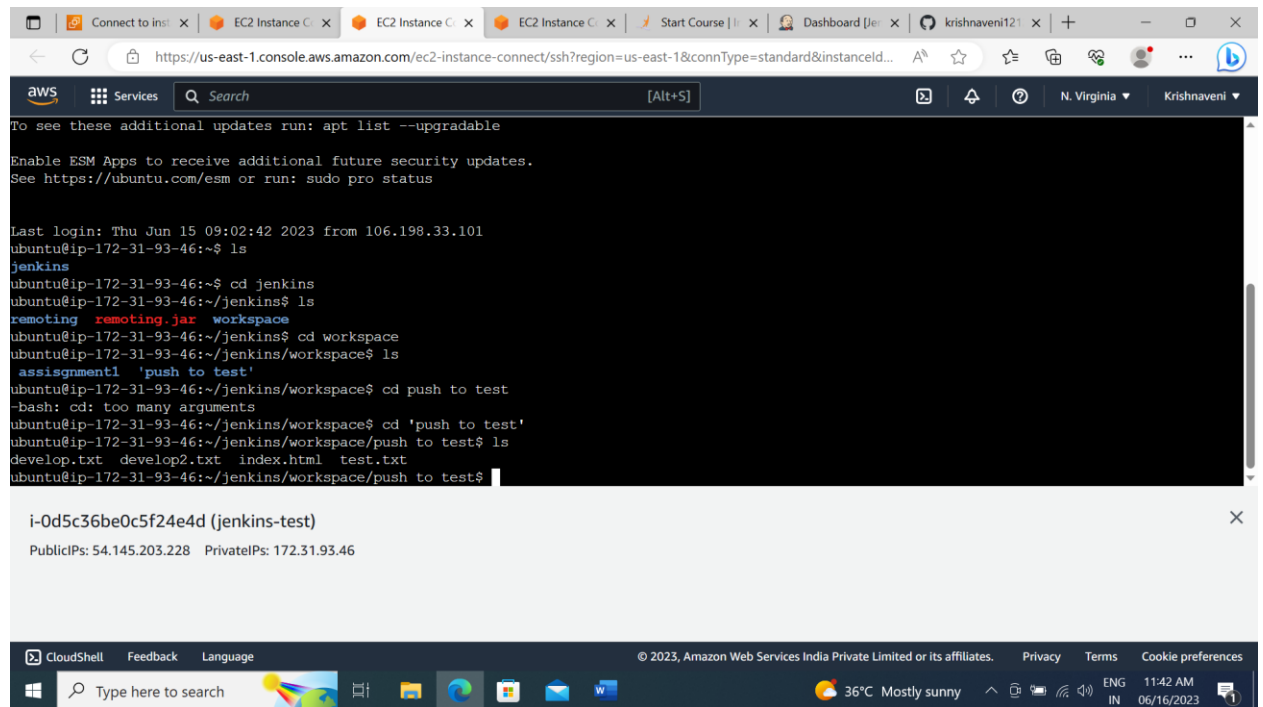


The screenshot shows the AWS Management Console terminal for the same EC2 instance. The terminal output is as follows:

```
ubuntu@ip-172-31-95-153:~$ git remote -v
origin  git@github.com:krishnaveni1212/gittask.git (fetch)
origin  git@github.com:krishnaveni1212/gittask.git (push)
ubuntu@ip-172-31-95-153:~$ git push origin test
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 352 bytes | 352.00 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 'test' on GitHub by visiting:
remote:   https://github.com/krishnaveni1212/gittask/pull/new/test
remote:
To github.com:krishnaveni1212/gittask.git
 * [new branch]      test -> test
ubuntu@ip-172-31-95-153:~$ git checkout master
Switched to branch 'master'
ubuntu@ip-172-31-95-153:~$ ls
index.html
ubuntu@ip-172-31-95-153:~$ git push origin master
```

Below the terminal, the same metadata box for the instance `i-01fd77dffbe6648d8 (jenkins-master)` is visible, showing Public IPs: 54.152.82.238 and Private IPs: 172.31.95.153.

Step 5: At test server



The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output includes instructions for updating apt, enabling ESM, and logging in. The user then navigates to the Jenkins workspace and runs a series of commands to set up the Jenkins environment, including installing Jenkins, creating a workspace, and pushing test files. The terminal output shows the successful execution of these commands and the creation of the Jenkins workspace.

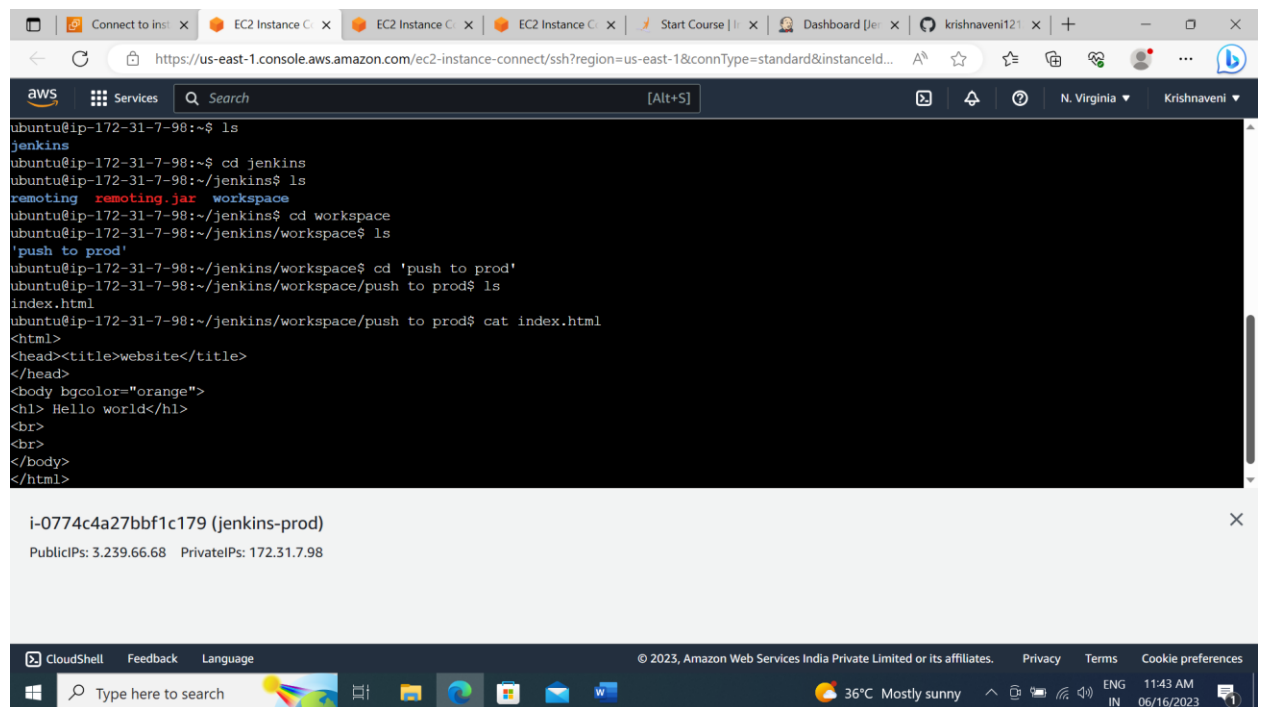
```
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Jun 15 09:02:42 2023 from 106.198.33.101
ubuntu@ip-172-31-93-46:~$ ls
jenkins
ubuntu@ip-172-31-93-46:~$ cd jenkins
ubuntu@ip-172-31-93-46:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-93-46:~/jenkins$ cd workspace
ubuntu@ip-172-31-93-46:~/jenkins/workspace$ ls
assignment1  'push to test'
ubuntu@ip-172-31-93-46:~/jenkins/workspace$ cd push to test
-bash: cd: too many arguments
ubuntu@ip-172-31-93-46:~/jenkins/workspace$ cd 'push to test'
ubuntu@ip-172-31-93-46:~/jenkins/workspace/push to test$ ls
develop.txt  develop2.txt  index.html  test.txt
ubuntu@ip-172-31-93-46:~/jenkins/workspace/push to test$
```

i-0d5c36be0c5f24e4d (jenkins-test)
PublicIPs: 54.145.203.228 PrivateIPs: 172.31.93.46

Step 6: at prod server



The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output includes instructions for updating apt, enabling ESM, and logging in. The user then navigates to the Jenkins workspace and runs a series of commands to set up the Jenkins environment, including installing Jenkins, creating a workspace, and pushing test files. The terminal output shows the successful execution of these commands and the creation of the Jenkins workspace.

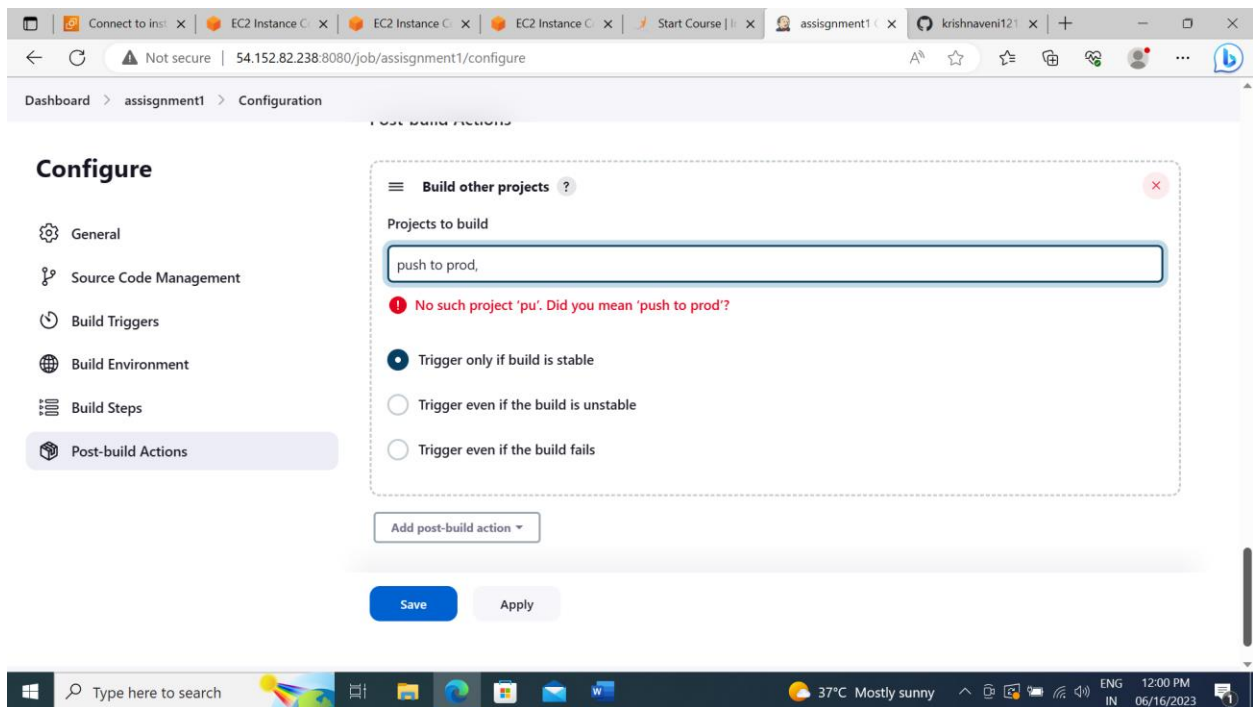
```
ubuntu@ip-172-31-7-98:~$ ls
jenkins
ubuntu@ip-172-31-7-98:~$ cd jenkins
ubuntu@ip-172-31-7-98:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-7-98:~/jenkins$ cd workspace
ubuntu@ip-172-31-7-98:~/jenkins/workspace$ ls
'push to prod'
ubuntu@ip-172-31-7-98:~/jenkins/workspace$ cd 'push to prod'
ubuntu@ip-172-31-7-98:~/jenkins/workspace/push to prod$ ls
index.html
ubuntu@ip-172-31-7-98:~/jenkins/workspace/push to prod$ cat index.html
<html>
<head><title>website</title>
</head>
<body bgcolor="orange">
<h1> Hello world</h1>
<br>
<br>
</body>
</html>
```

i-0774c4a27bbf1c179 (jenkins-prod)
PublicIPs: 3.239.66.68 PrivateIPs: 172.31.7.98

Jenkin Task3-

1. Create a pipeline in Jenkins
2. Once push is made to “develop” a branch in Git, trigger job “test”. This will copy Git files to test node
3. If test job is successful, then prod job should be triggered
4. Prod jobs should copy files to prod node

Step 1:configured in Task1 job



Step 2:

The screenshot shows the Jenkins web interface for a project named 'assignment1'. The browser tabs include 'Connect to ins...', 'EC2 Instance C...', 'Start Cour...', 'assignment1', and 'krishnaveni121'. The address bar shows '54.152.82.238:8080/job/assignment1/'. The left sidebar contains links: 'Build Now', 'Configure', 'Delete Project', 'GitHub Hook Log', 'GitHub', and 'Rename'. The main content area has a 'Downstream Projects' section with a 'push to prod' link. Below it is a 'Permalinks' section with a list of build links: 'Last build (#3), 4 sec ago', 'Last stable build (#3), 4 sec ago', 'Last successful build (#3), 4 sec ago', and 'Last completed build (#3), 4 sec ago'. A 'Build History' section is visible at the bottom, showing builds #1, #2, and #3 with their timestamps. The Windows taskbar at the bottom shows the search bar, task icons, and system tray with weather and time information.

Step

3:

The screenshot shows the Jenkins web interface for a project named 'push to prod'. The browser tabs include 'Connect to ins...', 'EC2 Instance C...', 'Start Cour...', 'push to prod', and 'krishnaveni121'. The address bar shows '54.152.82.238:8080/job/push%20to%20prod/'. The left sidebar contains links: 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', 'GitHub Hook Log', 'GitHub', and 'Rename'. The main content area has a 'Project push to prod' section with the description 'master will be pushed and data should be gathered in prod machine'. Below it is an 'Upstream Projects' section with an 'assignment1' link. A 'Permalinks' section follows with a list of build links: 'Last build (#3), 36 sec ago', 'Last stable build (#3), 36 sec ago', 'Last successful build (#3), 36 sec ago', 'Last failed build (#1), 32 min ago', 'Last unsuccessful build (#1), 32 min ago', and 'Last completed build (#3), 36 sec ago'. A 'Build History' section is visible at the bottom, showing builds #2 and #3 with their timestamps. The Windows taskbar at the bottom shows the search bar, task icons, and system tray with weather and time information.

Step 4: Installed build pipeline plugin and created a view

Dashboard > view > Configure

+ New Item

People

Build History

Edit View

Delete View

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

Build Executor Status

Built-In Node

Name: view

Description: assignment1 and push to prod

[Plain text] Preview

☐ Filter build queue

☐ Filter build executors

Build Pipeline View Title: jenkins-task3

OK Apply

Step 5.: created a pipeline view

Dashboard > view >

Build Pipeline: jenkins-task3

assignment1 and push to prod

Trigger a Pipeline Pipeline History Configure Add Step Delete Manage

Pipeline #3

#3 assignment1
run 16, 2023 6:34:52 AM
0.6 sec
console re-run

#3 push to prod
run 16, 2023 6:35:02 AM
0.74 sec
console re-run

REST API Jenkins 2.401.1