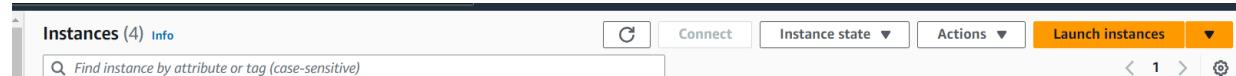


Day 11. Jenkins and Ansible Introduction with project

DEVOPS PROJECT DEPLOYMENT

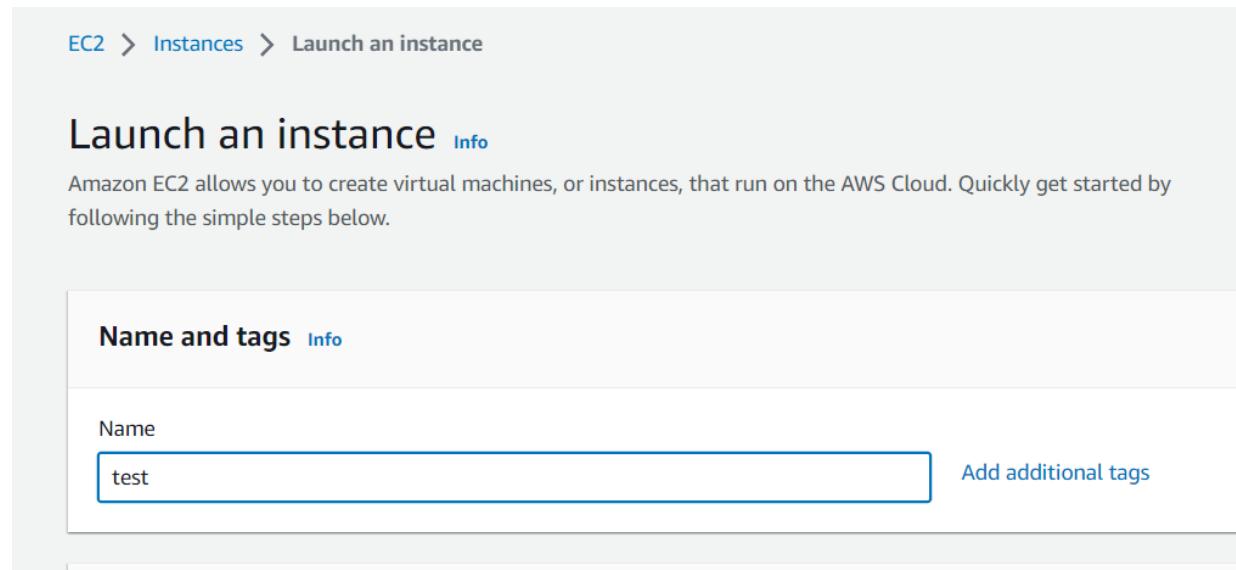
- CREATE 4 INSTANCES

- ❖ Click on launch instances



A screenshot of the AWS EC2 Instances page. At the top, there's a header with 'Instances (4) Info' and various buttons like 'Connect', 'Actions', and 'Launch instances'. Below the header is a search bar with placeholder text 'Find instance by attribute or tag (case-sensitive)'. The main area shows a table with four rows, each representing an instance.

- ❖ Write name of the instances



A screenshot of the 'Launch an instance' wizard. The breadcrumb navigation shows 'EC2 > Instances > Launch an instance'. The main title is 'Launch an instance Info'. Below it, a sub-instruction reads: 'Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.' The first step is titled 'Name and tags Info'. It has a 'Name' field containing 'test' and a 'Add additional tags' link.

- ❖ Select instance OS . here i have selected the amazon linux 2 OS

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat S >

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type ami-09988af04120b3591 (64-bit (x86)) / ami-013e77ebd63dc2197 (64-bit (Arm)) Virtualization: hvm ENA enabled: true Root device type: ebs	Free tier eligible ▾
---	----------------------

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20230530.0 x86_64 HVM gp2

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

❖ Selecting key pair login

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before launching the instance.

Key pair name - *required*

Select Create new

|

Proceed without a key pair (Not recommended) Default value

devops-masters
Type: rsa

❖ Changing inbound security groups rules of network to all traffic , anywhere

Inbound security groups rules

▼ Security group rule 1 (All, All, 0.0.0.0/0) [Remove](#)

Type Info All traffic	Protocol Info All	Port range Info All
Source type Info Anywhere	Source Info <input type="text"/> Add CIDR, prefix list or security	Description - optional Info e.g. SSH for admin desktop <input type="text"/> 0.0.0.0/0 X

❖ Configure storage

The screenshot shows the 'Configure storage' section of an AWS EC2 instance configuration. It displays a single volume configuration: 1x 8 GiB gp3. A note indicates that free-tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. There is also an 'Add new volume' button and a link to edit file systems.

1x	8	GiB	gp3	Root volume (Not encrypted)
----	---	-----	-----	-----------------------------

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage X

Add new volume

0 x File systems Edit

❖ Write the number of instances . In this case i have selected 4 instances and click on launch instances.

The screenshot shows the 'Summary' section of the EC2 instance configuration. It lists the following details:

- Number of instances: 4
- Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more ami-09988af04120b3591
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

A note about the free tier is displayed: "Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is available)." The 'Launch instance' button is highlighted in orange.

Number of instances Info

4

When launching more than 1 instance, consider EC2 Auto Scaling.

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-09988af04120b3591

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is available) X

Cancel Launch instance

Review commands

❖ A success message will appear

EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instances (i-08a17664634d617db, i-0b8f3f37b1ba9f2c3, i-0345adfb6149978db, i-044b2e518135ff17f)

▶ Launch log

- - ❖ Rename the instances

Instances (4) Info			
	Name	Instance ID	Instance state
<input type="checkbox"/>	Developer 	i-0345adfb6149978db	Running t2.micro
<input type="checkbox"/>	Jenkins 	i-044b2e518135ff17f	Running t2.micro
<input type="checkbox"/>	Ansible 	i-08a17664634d617db	Running t2.micro
<input type="checkbox"/>	Web 	i-0b8f3f37b1ba9f2c3	Running t2.micro

- Now login to github and create new repository

github.com

Search or jump to...

Pull requests Issues Codespaces Marketplace Explore

Top Repositories [New](#)

For you [Beta](#) Following

Welcome to the new feed!

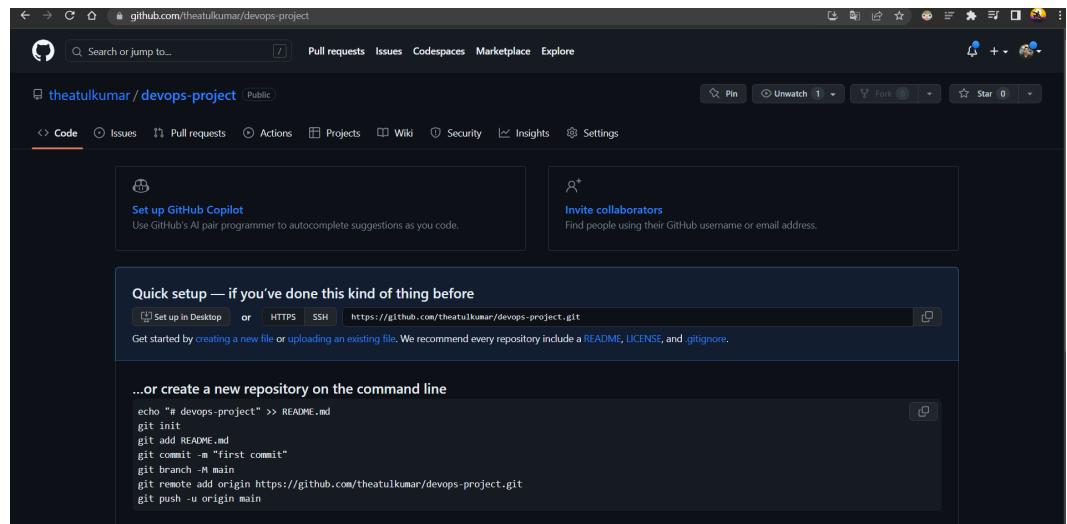
- ❖ Write repository name

Create a new repository	
A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.	
<i>Required fields are marked with an asterisk (*).</i>	
Owner *	Repository name *
theatulkumar 	/ devopsproject
<input checked="" type="checkbox"/> devopsproject is available.	

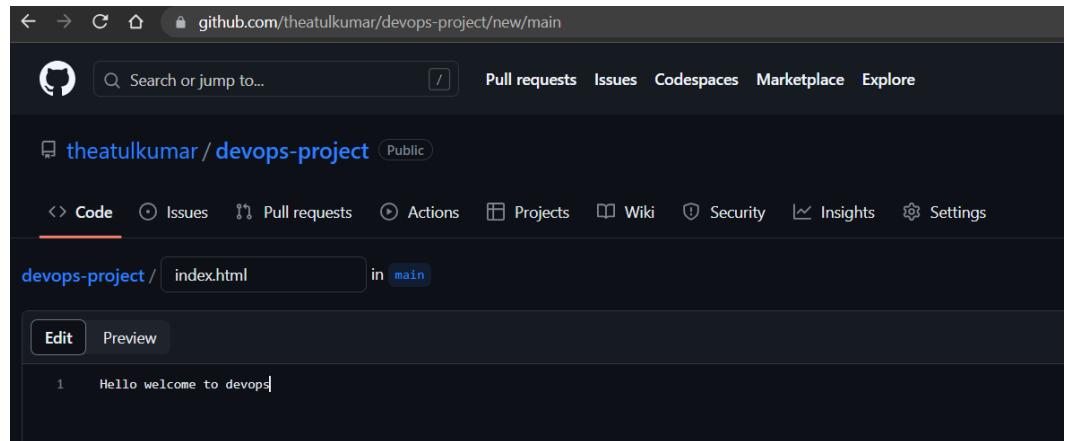
- ❖ Click on create repository

ⓘ You are creating a public repository in your personal account.
Create repository

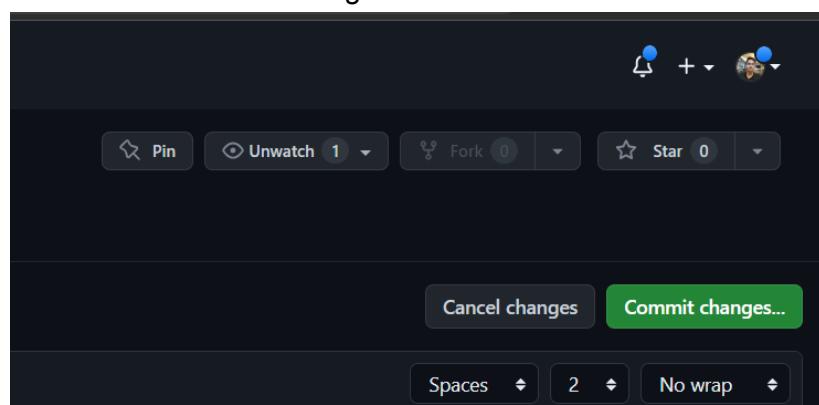
- ❖ This will appear after creating a repository. Now click on create a file(index.html).

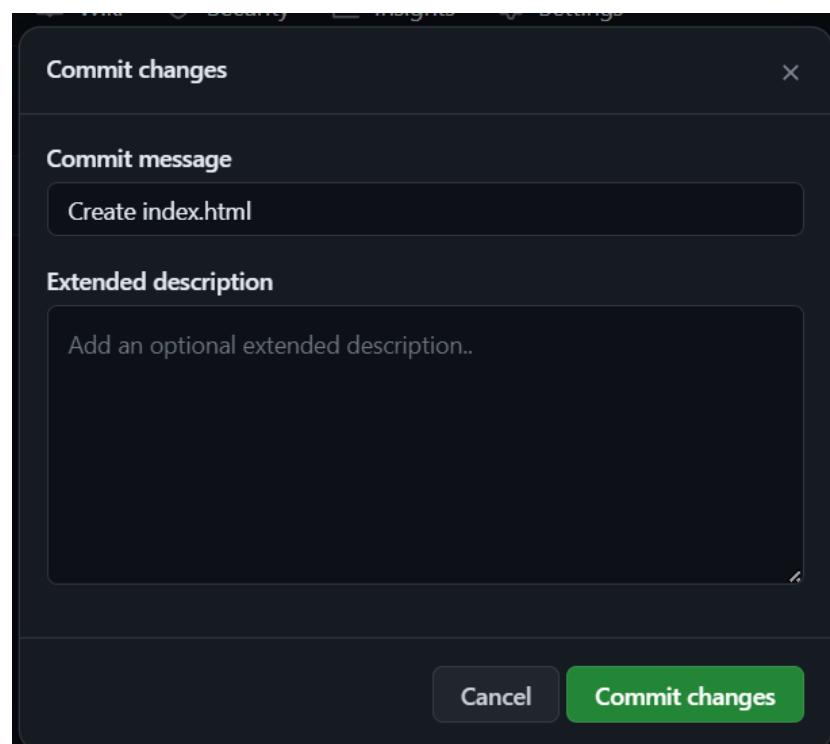


❖ Writing something in file like “Hello welcome to devops”



❖ Now click on commit changes





- Now open Xshell and connect all the instances.
Copy the command and paste accordingly one by one to connect all the 4 instances

AWS Services Search [Alt+S]

EC2 > Instances > i-0345adfb6149978db > Connect to instance

Connect to instance Info

Connect to your instance i-0345adfb6149978db (Developer) using any of these options

EC2 Instance Connect **Session Manager** **SSH client** **EC2 serial console**

Instance ID
i-0345adfb6149978db (Developer)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is devops-masters.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 devops-masters.pem
4. Connect to your instance using its Public DNS:
ec2-34-201-110-254.compute-1.amazonaws.com

Command copied

ssh -i "devops-masters.pem" ec2-user@ec2-34-201-110-254.compute-1.amazonaws.com

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Developer - root@ip-172-31-86-180:~ - Xshell 7 (Free for Home/School)

File Edit View Tools Tab Window Help

ssh://ec2-user@ec2-34-201-110-254.compute-1.amazonaws.com:22

To add the current session, click on the left arrow button.

Session Manager 1 Developer 2 Local Shell 3 Local Shell 4 Local Shell +

Xshell 7 (Build 0128)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserved.

Type `help` to learn how to use Xshell prompt.
[C:\~]\$ ssh -i "devops-masters.pem" ec2-user@ec2-34-201-110-254.compute-1.amazonaws.com

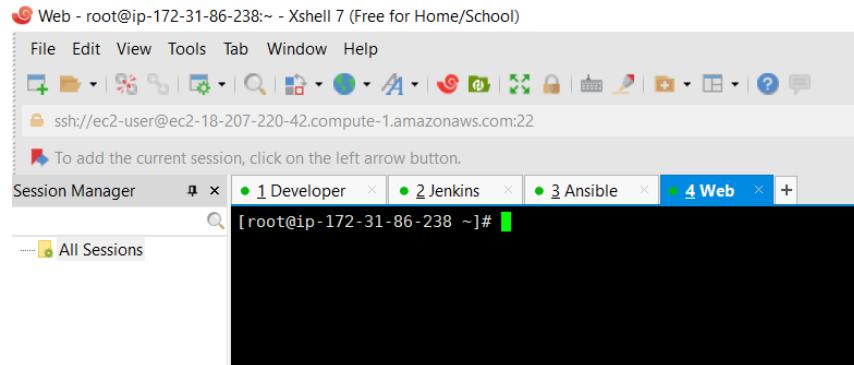
Host 'ec2-34-201-110-254.compute-1.amazonaws.com' resolved to 34.201.110.254.
Connecting to 34.201.110.254:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.
WARNING! The remote SSH server rejected X11 forwarding request.

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
7 package(s) needed for security, out of 13 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-86-180 ~]\$ sudo su
[root@ip-172-31-86-180 ec2-user]# cd
[root@ip-172-31-86-180 ~]#

- ❖ Write a command after connecting to the server.
- ❖ sudo su (superuser do)

- ❖ cd (change directory)
- ❖ We can use Ctrl + L to clear the screen of Xshell.
- ❖ Connected each instances one by one



❖

- Now we need to set up password less authentication from developer machine to github .
- We cannot give the id and password of github to developer that is why we need to do password less authentication.
- So we need to create a public key.
- Write the command “ssh-keygen”

```
[root@ip-172-31-86-180 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:5KYYesR8RzwAxw0BDUohHReKiX4IyMQBhpEE1v7es root@ip-172-31-86-180.ec2.internal
The key's randomart image is:
+---[RSA 2048]---+
|//o.=o          |
|%o=o 0o.        |
|@*..* =         |
|=+=o= *         |
| . * o S       |
|   . +          |
|     o .         |
|       . .        |
|         E        |
+---[SHA256]---+
[root@ip-172-31-86-180 ~]#
```

- Now use cat command to check the public key where it is stored

```
[root@ip-172-31-86-180 ~]# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAMBA0DE/6xMdtz4gnIfr1rs0+eNYv4EBfNMzGCKAh0MvbLasj5k7BZkqcsZzUfEW5SDczvzItJRhELAc/rk/FRrf1iosS3Xv0xd5cDz0X2k1sBj7RUdZg6JX0KNmNdj3ZFgGstcc4tjDNYF9w8BmrRknLCf6i16j02cp0wuaUm79pwfOnUPjXQgKUA0tZE60Q1CPfJY7qH0NuX1QEKLQ280D0H9z7omy9HJ3nw1WrdMuazX0lq1161Et2sb0Wvxt3gwrvUxJ7TDuo5Wj4xm30payV0DrCrXpN8nLda+eK6zHgHakSeKxc8khRC19 root@ip-172-31-86-180.ec2.internal
[root@ip-172-31-86-180 ~]#
```

- Now we need to copy the content of the file.
- Then go to github click on settings then click on deploy keys. Add new

The screenshot shows the GitHub Settings interface for a repository named 'devops-project'. The 'Deploy keys / Add new' section is active. A new key titled 'Atul' is being added. The key content is a long SSH RSA public key:

```

ssh-rsa
AAAAAADAQABAAAQDf/GxMdtz4gnJfrfrQ+eNy4EBIMMzGCKAh0MVblasjk7BZkqcsZzUEW5Sdc
zvzlUfRheLA/rk/FRfrfiosS3Xw0xd5cDZ0X2k1sB7RUUzGK0NmNnNdj3Zfgstcc4jDNTF9w8RmRlCf4f6QZopOwiua
Um79wpfOnUfPxQgKUAoFZEQO1CPHYtqHONuX1QEKLQ28ODD0I9ztvDRH91qJ7Dmy9HJ3nw1WrGMuazX0lq116Etzs
b0Wwxt3gWVlxjTTDU65Wlj4kmXQenSBAlp43DpayyW6DrCrxvNBnL0lda+eKzGmgThaKsekXclh1RC9 root@ip-172-
31-86-180.ec2.internal

```

The 'Allow write access' checkbox is checked.

The screenshot shows the GitHub Settings interface for a repository named 'devops-project'. The 'Deploy keys / Add new' section is active. A new key titled 'Atul' is being added. The key content is a long SSH RSA public key:

```

ssh-rsa
AAAAAADAQABAAAQDf/GxMdtz4gnJfrfrQ+eNy4EBIMMzGCKAh0MVblasjk7BZkqcsZzUEW5Sdc
zvzlUfRheLA/rk/FRfrfiosS3Xw0xd5cDZ0X2k1sB7RUUzGK0NmNnNdj3Zfgstcc4jDNTF9w8RmRlCf4f6QZopOwiua
Um79wpfOnUfPxQgKUAoFZEQO1CPHYtqHONuX1QEKLQ28ODD0I9ztvDRH91qJ7Dmy9HJ3nw1WrGMuazX0lq116Etzs
b0Wwxt3gWVlxjTTDU65Wlj4kmXQenSBAlp43DpayyW6DrCrxvNBnL0lda+eKzGmgThaKsekXclh1RC9 root@ip-172-
31-86-180.ec2.internal

```

The 'Allow write access' checkbox is checked.

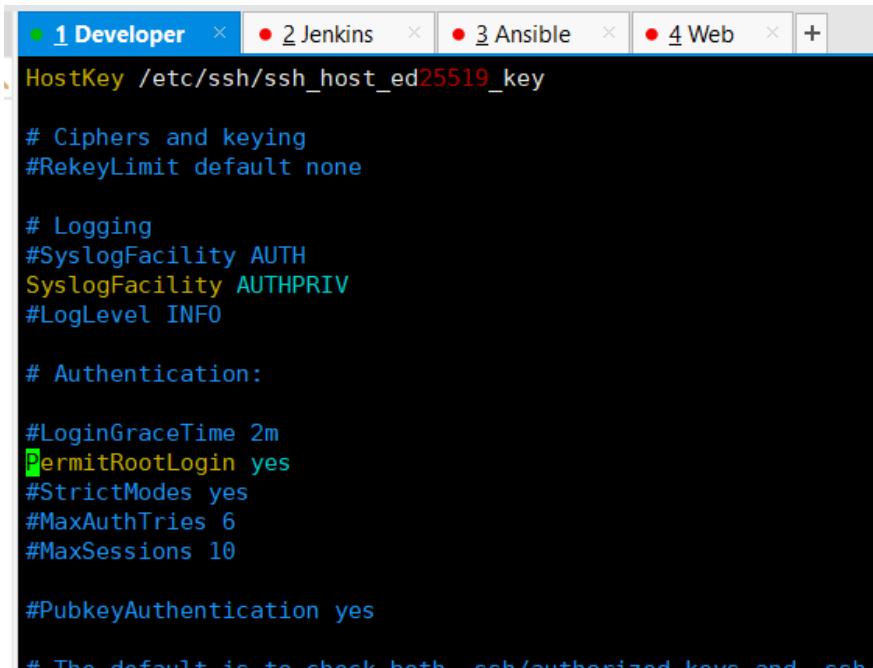
The screenshot shows the GitHub Settings interface for a repository named 'devops-project'. The 'Deploy keys' section is active. A key titled 'Atul' is listed. The key details are:

- SHA256:** 5kYye5R8RzwAwo0BDUohMlReK1x41yNQ8hpEE1v7es
- Added on:** Jun 14, 2023 by @theatulkumar
- Never used — Read/write**

- Now change the root password by writing command “`passwd root`”
- I gave my password as `root123` or `AKpass23`
- Then we need to enable password less authentication
- Write command “`vi /etc/ssh/sshd_config`”

```
[root@ip-172-31-86-180 ~]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-86-180 ~]#
```

- PermitRootLogin yes



```
HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/
```

- Now set PasswordAuthentication yes

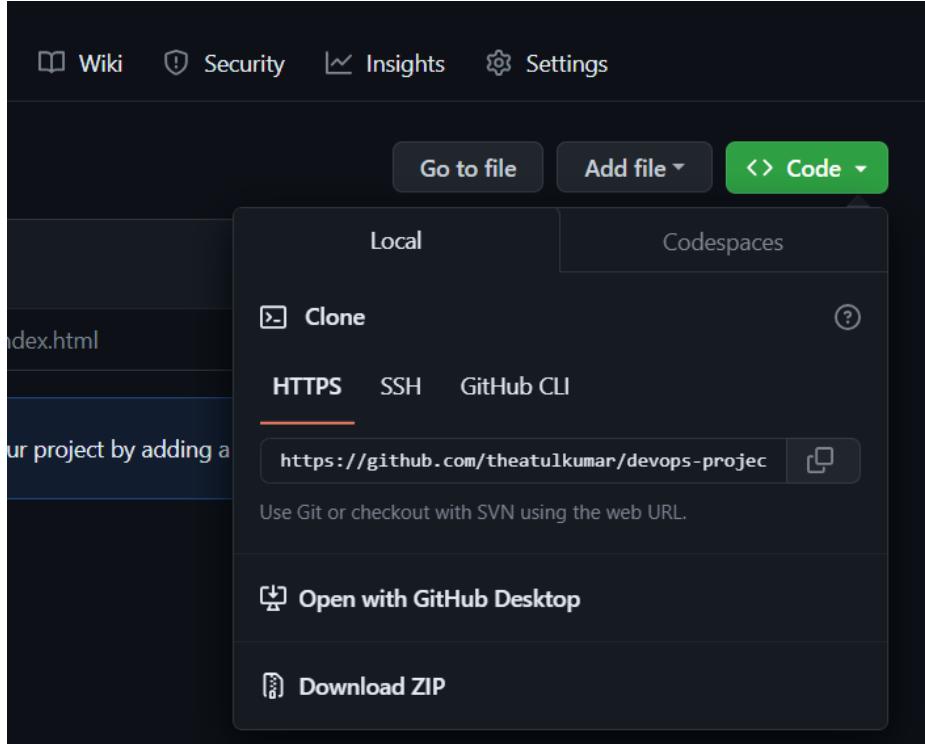
```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
```

- Now save and exit

```
# If you just want the PAM account and session checks to run without
:wq
```

- Now write command “systemctl restart sshd”
- Now we need to create a folder “mkdir /test”
- Then change directory to test “cd /test”
- Now we need to initialize git but before that we need to install git package.
- To install git “yum install git”

- “git init”
- Now we will clone the repository by writing command “git clone <https://github.com/theatulkumar/devops-project.git>”
- ***update instead of using https link use ssh link



- Once we are done do “ls”

```
[root@ip-172-31-86-180 ~]# ls
devops-project
[root@ip-172-31-86-180 ~]#
```

- “cd devops-project”
- “ls”

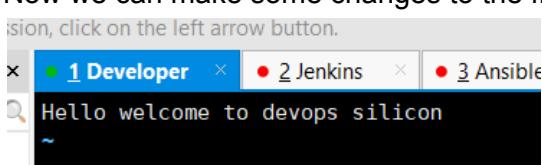
```
[root@ip-172-31-86-180 ~]# cd devops-project/
[root@ip-172-31-86-180 devops-project]# ls
index.html
[root@ip-172-31-86-180 devops-project]#
```

- “cat index.html”

```
[root@ip-172-31-86-180 devops-project]# cat index.html
Hello welcome to devops
[root@ip-172-31-86-180 devops-project]#
```

- “vi index.html”

- Now we can make some changes to the file. Here i added “silicon” in the file



- Now we need to push this to github.
- This is screenshot from class recording.

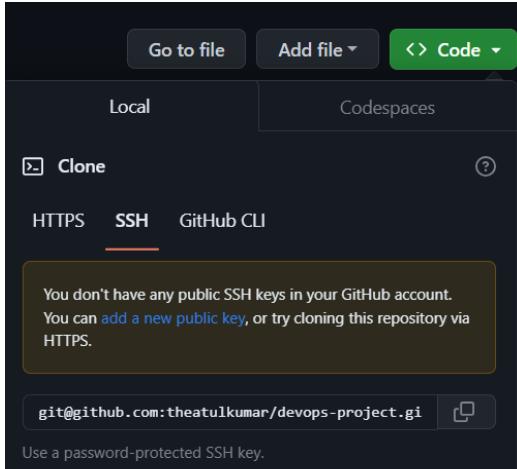
```
+++++
Developer->Github->Jenkins->Ansible->Webserver
+++++  
  
-----  
Github Setup  
-----  
  
echo "# devops-sit" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git branch -M main  
git remote add origin git@github.com:sumitsah9263/devops-sit.git  
git push -u origin main
```

- Now write command to add file to github “git add index.html”
- “git commit -m “test commit” index.html”

```
[root@ip-172-31-86-180 ~]# ls  
devops-project  
[root@ip-172-31-86-180 ~]# cd devops-project/  
[root@ip-172-31-86-180 devops-project]# ls  
index.html  
[root@ip-172-31-86-180 devops-project]# cat index.html  
Hello welcome to devops  
[root@ip-172-31-86-180 devops-project]# vi index.html  
[root@ip-172-31-86-180 devops-project]# git add index.html  
[root@ip-172-31-86-180 devops-project]# git commit -m "test-commit" index.html  
[main bb8e21e] test-commit  
Committer: root <root@ip-172-31-86-180.ec2.internal>  
Your name and email address were configured automatically based  
on your username and hostname. Please check that they are accurate.  
You can suppress this message by setting them explicitly. Run the  
following command and follow the instructions in your editor to edit  
your configuration file:  
  
git config --global --edit  
  
After doing this, you may fix the identity used for this commit with:  
  
git commit --amend --reset-author  
  
1 file changed, 1 insertion(+), 1 deletion(-)  
[root@ip-172-31-86-180 devops-project]#
```

- “git branch -M main”
- “git remote add origin [git@github.com:theatulkumar/devops-project.git](https://github.com/theatulkumar/devops-project.git)”

```
1 file changed, 1 insertion(+), 1 deletion(-)  
[root@ip-172-31-86-180 devops-project]# git branch -M main  
[root@ip-172-31-86-180 devops-project]# git remote add origin git@github.com:theatulkumar/devops-project.git  
error: remote origin already exists.  
[root@ip-172-31-86-180 devops-project]# git push -u origin main  
Username for 'https://github.com':  
Password for 'https://github.com':  
remote: No anonymous write access.  
fatal: Authentication failed for 'https://github.com/theatulkumar/devops-project.git/'  
[root@ip-172-31-86-180 devops-project]# mkdir /project  
[root@ip-172-31-86-180 devops-project]# cd /project  
[root@ip-172-31-86-180 project]#
```



- “git push -u origin main”

```
1 file changed, 1 insertion(+), 1 deletion(-)
[root@ip-172-31-86-180 devops-project]# git branch -M main
[root@ip-172-31-86-180 devops-project]# git remote add origin git@github.com:theatulkumar/devops-project.git
error: remote origin already exists.
[root@ip-172-31-86-180 devops-project]# git push -u origin main
Username for 'https://github.com':
Password for 'https://github.com':
remote: No anonymous write access.
fatal: Authentication failed for 'https://github.com/theatulkumar/devops-project.git'
[root@ip-172-31-86-180 devops-project]# mkdir /project
[root@ip-172-31-86-180 devops-project]# cd /project
[root@ip-172-31-86-180 project]#
```

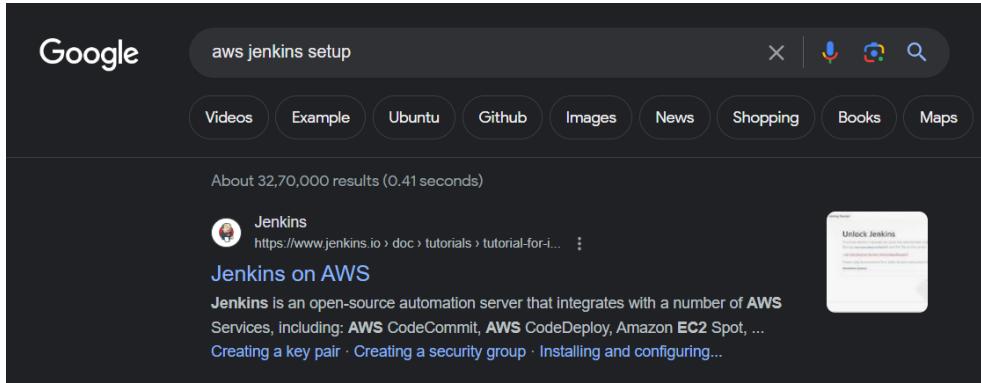
- We can see that authentication has failed. So now let's try again
- Creating new folder named ‘project’ - “mkdir /project”
- changing directory to project “cd /project”
- “git init”
- “git clone <https://github.com/theatulkumar/devops-project.git>”

```
[root@ip-172-31-86-180 devops-project]# cd /project
[root@ip-172-31-86-180 project]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /project/.git/
[root@ip-172-31-86-180 project]# git clone https://github.com/theatulkumar/devops-project.git
Cloning into 'devops-project'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
[root@ip-172-31-86-180 project]# cd
[root@ip-172-31-86-180 ~]# ls
devops-project
[root@ip-172-31-86-180 ~]# cd devops-project/
[root@ip-172-31-86-180 devops-project]#
```

- “cd”
- “ls”
- “cd devops-project”
- “ls”
- “cat index.html”
- “git add index.html”
- “git commit -m “test commit” index.html”
- “git branch -M main”
- “git remote add origin”
- “git push -u origin main”

Now we need to setup jenkins

- Open google and search “aws jenkins setup”. Click on the link



- After opening the link follow the given commands
 - ❖ `sudo yum update -y`
 - ❖ `sudo wget -O /etc/yum.repos.d/jenkins.repo \ https://pkg.jenkins.io/redhat-stable/jenkins.repo`
 - ❖ `sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key`
 - ❖ `sudo yum upgrade`
 - ❖ `sudo amazon-linux-extras install java-openjdk11 -y`
 - ❖ `sudo yum install jenkins -y`
 - ❖ `sudo systemctl enable jenkins`
 - ❖ `sudo systemctl start jenkins`
 - ❖ `sudo systemctl start jenkins`
 - ❖ `sudo systemctl status jenkins`

Downloading and installing Jenkins

Completing the previous steps enables you to download and install Jenkins on AWS. To download and install Jenkins:

1. Ensure that your software packages are up to date on your instance by using the following command to perform a quick software update:

```
[ec2-user ~]$ sudo yum update -y
```

2. Add the Jenkins repo using the following command:

```
[ec2-user ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

3. Import a key file from Jenkins-CI to enable installation from the package:

```
[ec2-user ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

```
[ec2-user ~]$ sudo yum upgrade
```

4. Install Java (Amazon Linux 2):

```
[ec2-user ~]$ sudo amazon-linux-extras install java-openjdk11 -y
```

5. Install Java (Amazon Linux 2023):

```
[ec2-user ~]$ sudo dnf install java-11-amazon-corretto -y
```

6. Install Jenkins:

```
[ec2-user ~]$ sudo yum install jenkins -y
```

7. Enable the Jenkins service to start at boot:

```
[ec2-user ~]$ sudo systemctl enable jenkins
```

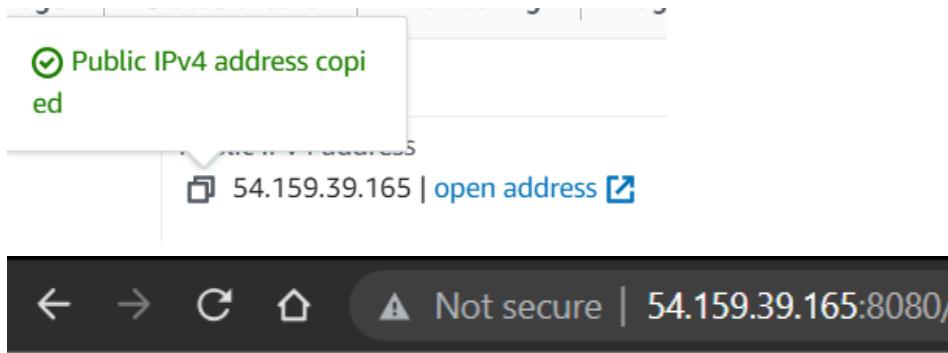
8. Start Jenkins as a service:

```
[ec2-user ~]$ sudo systemctl start jenkins
```

You can check the status of the Jenkins service using the command:

```
[ec2-user ~]$ sudo systemctl status jenkins
```

- After performing these commands copy the public ip of the jenkins server and paste in browser and jenkins runs on port 8080 so add 8080 after public ip address



- added :8080 (port number)

Now we can see the page like this, where the initial password is located. So “copy the location.”

A screenshot of a browser window displaying the Jenkins "Unlock Jenkins" setup page. The title is "Getting Started" and the main heading is "Unlock Jenkins". A note states: "To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server: /var/lib/jenkins/secrets/initialAdminPassword". Below this, instructions say: "Please copy the password from either location and paste it below." There is a text input field labeled "Administrator password" with a placeholder icon of a hand holding a key. A "Continue" button is at the bottom right.

Use cat command to check the password

```
Hint: Some lines were ellipsized, use -e to show in full.  
[root@ip-172-31-18-37 ~]# cat /var/lib/jenkins/secrets/initialAdminPassword  
65a19d0c64e649cdb016462b7f12c3f0  
[root@ip-172-31-18-37 ~]#
```

Now paste the password and click on continue

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

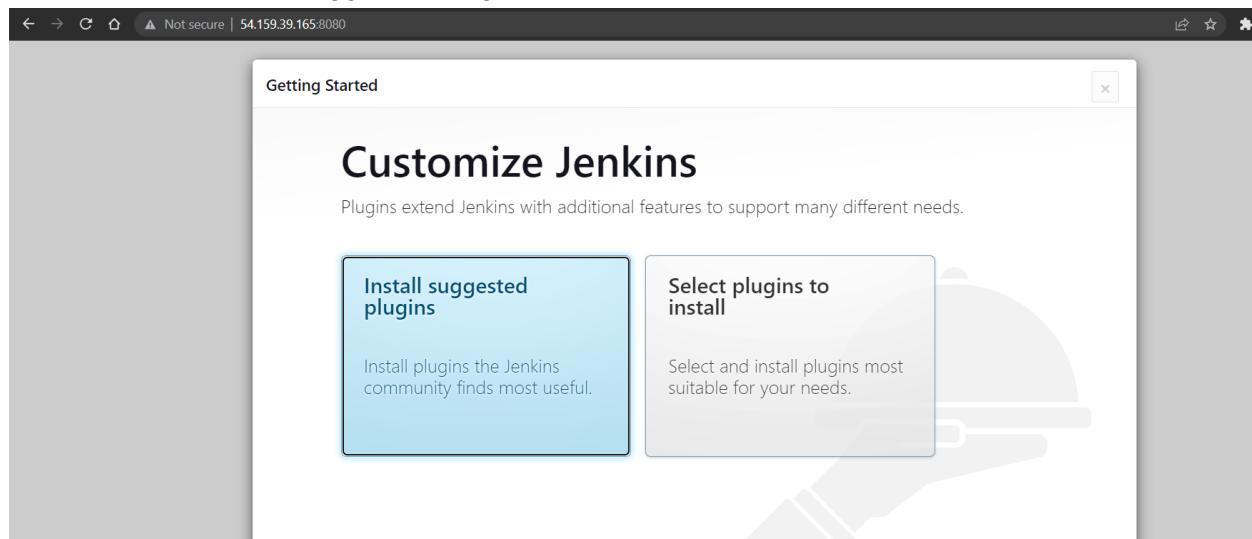
Please copy the password from either location and paste it below.

Administrator password

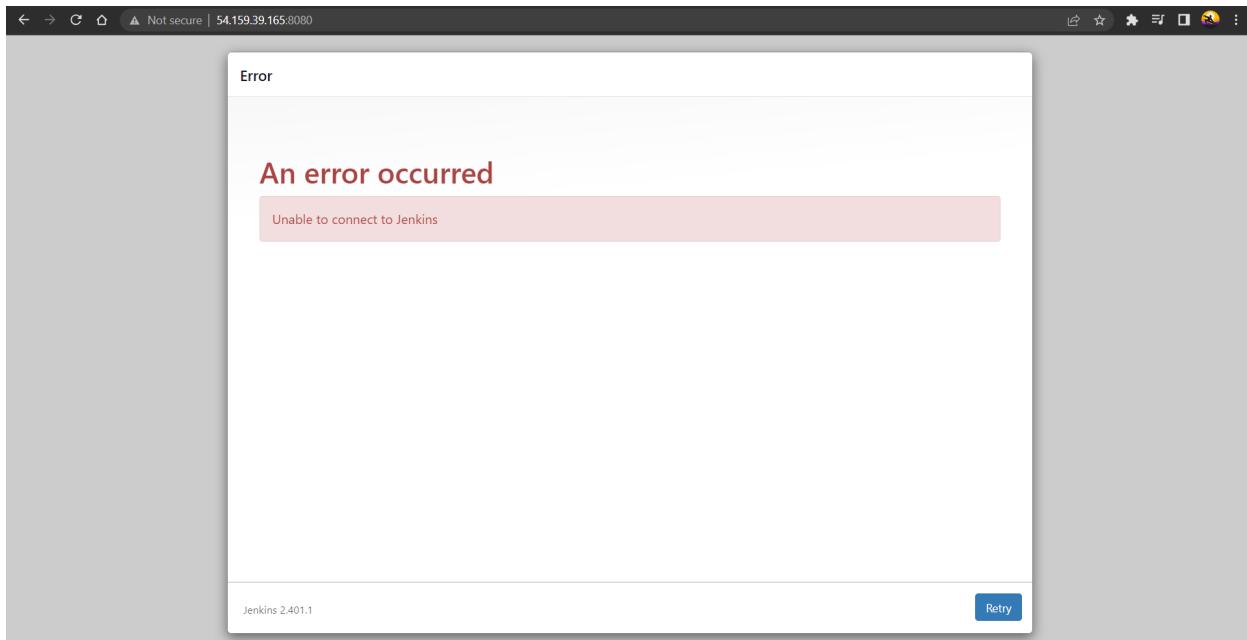
.....|

Continue

- Click on install suggested plugins.



In my case some error has occurred so I will try to fix the error.



- After clicking on Install suggested plugins this will appear.

A screenshot of the Jenkins 'Getting Started' page. The title 'Getting Started' is at the top. Below it is a large green button. The main content is a grid of plugin recommendations. The grid has four columns and five rows. The first column contains: 'Folders' (with a green checkmark icon), 'Timestamper', 'Pipeline', 'Git', and 'LDAP'. The second column contains: 'OWASP Markup Formatter', 'Workspace Cleanup', 'GitHub Branch Source', 'SSH Build Agents', and 'Email Extension'. The third column contains: 'Build Timeout', 'Ant', 'Pipeline: GitHub Groovy Libraries', 'Matrix Authorization Strategy', and 'Mailer'. The fourth column contains: 'Credentials Binding', 'Gradle', 'Pipeline: Stage View', 'PAM Authentication', and 'Mailer'. There are two notes on the right side: '** Ionicons API Folders' and '** bouncycastle API'. At the bottom right, it says '** - required dependency'. At the very bottom, it says 'Jenkins 2.401.1'.

Now fill in the details then save and continue.

Getting Started

Create First Admin User

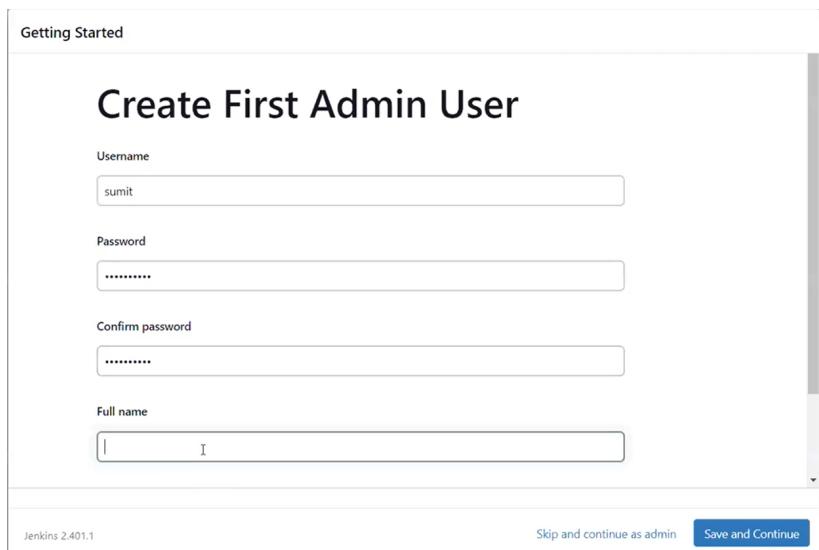
Username

Password

Confirm password

Full name

Jenkins 2.401.1 Skip and continue as admin Save and Continue



- Now click on save and finish.

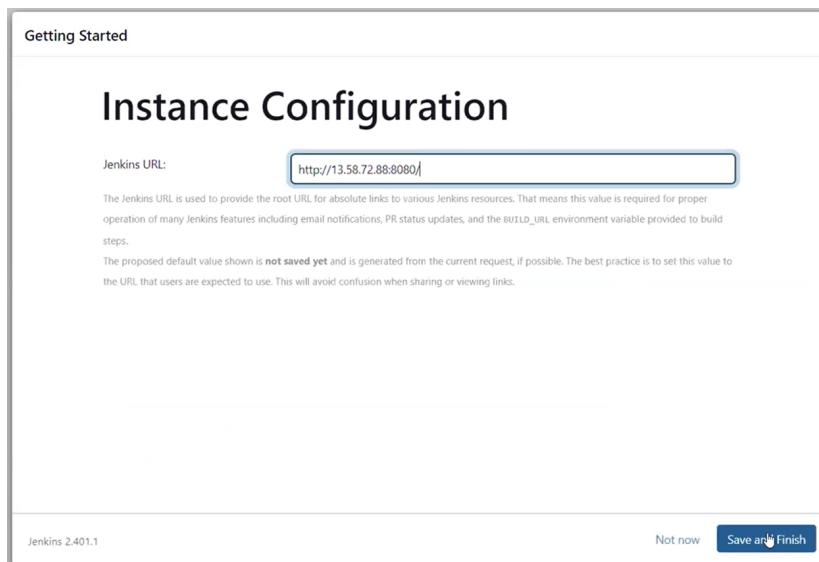
Getting Started

Instance Configuration

Jenkins URL:

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 steps.
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 2.401.1 Not now Save and Finish



- Now click on start using Jenkins

Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

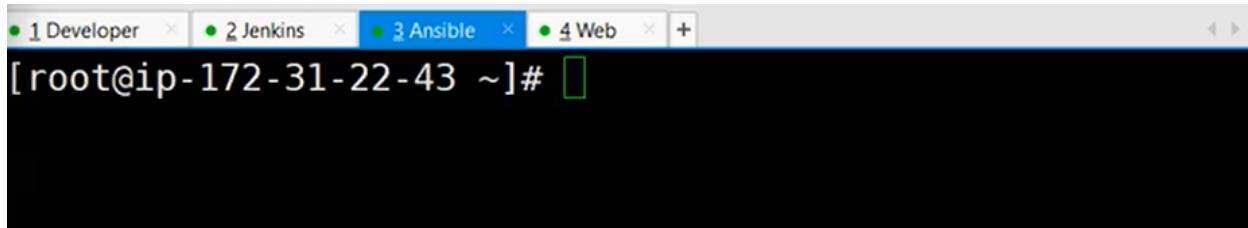
[Start using Jenkins](#)



This is the page of jenkins

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'My Views'. Below that is a 'Build Queue' section with a message 'No builds in the queue.' To the right, under 'Welcome to Jenkins!', it says 'This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.' It features three main call-to-action buttons: 'Create a job', 'Set up a distributed build', and 'Learn more about distributed builds'.

- Now we need to setup ansible



- For setting up ansible we have command : “**sudo amazon-linux-extras install ansible2**”

A terminal window showing the output of the command 'sudo amazon-linux-extras install ansible2'. The output includes:

```
[root@ip-172-31-22-43 ~]# sudo amazon-linux-extras install ansible2
Installing ansible
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-ansible2 amzn2extra-docker amzn2extra-kernel-5.10
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-ansible2
amzn2extra-docker
amzn2extra-kernel-5.10
(1/9): amzn2-core/2/x86_64/group_gz | 3.7 kB 00:00:00
(2/9): amzn2-core/2/x86_64/updateinfo | 3.0 kB 00:00:00
(3/9): amzn2extra-docker/2/x86_64/primary_db | 3.0 kB 00:00:00
(4/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo | 3.0 kB 00:00:00
(5/9): amzn2extra-ansible2/2/x86_64/updateinfo | 2.5 kB 00:00:00
(6/9): amzn2extra-docker/2/x86_64/updateinfo | 629 kB 00:00:00
(7/9): amzn2extra-ansible2/2/x86_64/primary_db | 106 kB 00:00:00
(8/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo | 30 kB 00:00:00
(9/9): amzn2extra-ansible2/2/x86_64/primary_db | 76 B 00:00:00
[...]
```

After installation of ansible we need to add ansible with the web server.

- For adding ansible with the web server we need to change the configuration file.
“vi /etc/ansible/hosts”

```
[root@ip-172-31-22-43 ~]# vi /etc/ansible/hosts
```

Here we can add any number of web servers by providing their private IP address. If we want to add 50 number of web servers we can add them here.

- Copy the private IP id

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Developer	i-08359f06419fd9da4	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2b	ec2-52-15-68-23
Jenkins	i-0b080b35a3b1c3679	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2b	ec2-13-58-72-88
Ansible	i-0eabab9b4036a46c	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2b	ec2-52-15-105-2
Web	i-001d7f594526ac6a6	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2b	ec2-18-216-208-

Instance: i-001d7f594526ac6a6 (Web)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Instance ID: i-001d7f594526ac6a6 (Web) Public IPv4 address: 18.216.208.246 [open address] Private IPv4 addresses: 172.31.16.30

IPV6 address: - Instance state: Running Public IPv4 DNS: ec2-18-216-208-246.us-east-2.compute.amazonaws.com [open address]

- Paste the IP address here like this :

[web]

172.31.16.30

```
# 
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

[web]
172.31.16.30

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
-- INSERT --
```

Then save and exit

Now ansible is configured with web server

- Now installing apache in the web server: “**yum install httpd**”

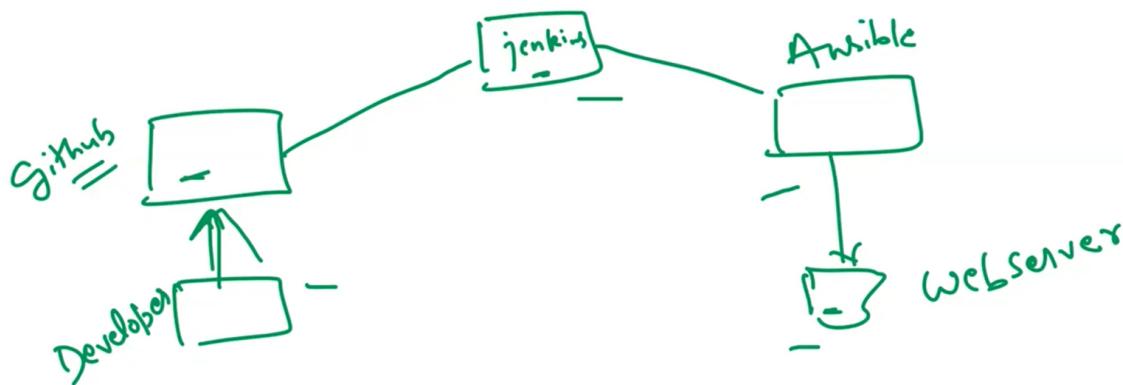
```
[root@ip-172-31-16-30 ~]# yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.57-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.57-1.amzn2 for package: httpd-2.4.57-1.amzn2.x86_64
| 3.7 kB 00:00:00
```

- “**systemctl start httpd**”
- “**systemctl enable httpd**”

```
[root@ip-172-31-16-30 ~]# systemctl start httpd
[root@ip-172-31-16-30 ~]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-16-30 ~]#
```

As we set up password less authentication for developer server on github.

We will set up password less authentication for ansible server to web server.



- Now change the password of the root “**passwd root**”

```
[root@ip-172-31-16-30 ~]# passwd root
Changing password for user root.
New password:
Retype new password:
```

We tend to forget the password or confuse it with the other password so we should write it somewhere.

Now we need to go to the sshd configuration file : “**vi /etc/ssh/sshd_config**”

```
[root@ip-172-31-16-30 ~]# vi /etc/ssh/sshd_config
```

- PermitRootLogin yes

```
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

- PasswordAuthentication yes

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
```

- After that we will do : “**systemctl restart sshd**”

```
[root@ip-172-31-16-30 ~]# systemctl restart sshd
```

Now in Ansible server

- “ssh-keygen”

The screenshot shows a terminal window with several tabs at the top: 1 Developer, 2 Jenkins, 3 Ansible (which is active), and 4 Web. The terminal content is as follows:

```
52 tomcat9           available      [ =stable ]
53 unbound1.13        available      [ =stable ]
54 mariadb10.5         available      [ =stable ]
55 kernel-5.10=latest  enabled       [ =stable ]
56 redis6             available      [ =stable ]
57 ruby3.0            available      [ =stable ]
58 postgresql12        available      [ =stable ]
59 postgresql13        available      [ =stable ]
60 mock2              available      [ =stable ]
61 dnsmasq2.85        available      [ =stable ]
62 kernel-5.15          available      [ =stable ]
63 postgresql14        available      [ =stable ]
64 firefox             available      [ =stable ]
65 lustre              available      [ =stable ]
66 php8.1              available      [ =stable ]
67 awsclil            available      [ =stable ]
68 php8.2              available      [ =stable ]
69 dnsmasq            available      [ =stable ]
70 unbound1.17          available      [ =stable ]
71 golang1.19          available      [ =stable ]
[root@ip-172-31-22-43 ~]# vi /etc/ansible/hosts
[root@ip-172-31-22-43 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): I
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:qU/myV92OrCytMqrztz/7PzpDkiE1YkMhzJd46tZTgHg root@ip-172-31-22-43.us-east-2.compute.internal
The key's randomart image is:
+---[RSA 2048]---+
| +.++o |
| . E.0 o |
| . = = .. |
| . . .0 + |
| . .Soooo o |
| o.0.o+.+ |
| .. 0.0oo |
| B.0..0 . |
| .oX+++=o=.. |
+---[SHA256]---+
[root@ip-172-31-22-43 ~]#
[root@ip-172-31-22-43 ~]#
[root@ip-172-31-22-43 ~]#
[root@ip-172-31-22-43 ~]# ssh-copy-id root@172.31.16.30
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.16.30 (172.31.16.30)' can't be established.
ECDSA key fingerprint is SHA256:P12jJLzMfh+vk0x3PeRk5ec/XSVWODCmdXK8wvaqdGs.
ECDSA key fingerprint is MD5:23:1d:f5:36:94:05:b7:ac:ea:c9:ca:7c:c0:55:b1:f5.
Are you sure you want to continue connecting (yes/no)? I
ssh://ec2-user@ec2-52-15-105-229.us-east-2.compute.amazonaws.com:22
SSH2 xterm 118x24 2456 4 sessions CAP NUM
```

The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Limits, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store. The main content area displays a table of instances with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. Five instances are listed: Developer (i-08359f06419fd9da4), Jenkins (i-0b080b35a3b1c5679), Ansible (i-0eaba9b84036a4a6c), and Web (i-001d7f594526ac6a6). The 'Web' instance is selected. The details pane for the 'Web' instance shows the instance summary with Public IPv4 address 18.216.208.246 and Private IPv4 address 172.31.16.30.

We put the private ip address of the web server to the ansible

- **ssh-copy-id root@172.31.16.30**

```
[root@ip-172-31-22-43 ~]# ssh-copy-id root@172.31.16.30
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.16.30 (172.31.16.30)' can't be established.
ECDSA key fingerprint is SHA256:P12jJLzFmH+vk0x3PeRk5ec/XSVWQDCmdXK8wvaqdGs.
ECDSA key fingerprint is MD5:23:1d:f5:36:94:05:b7:ac:ea:c9:ca:7c:c0:55:b1:f5.
Are you sure you want to continue connecting (yes/no)? yes
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.16.30's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@172.31.16.30'"
and check to make sure that only the key(s) you wanted were added.

[root@ip-172-31-22-43 ~]#
```

Now our ansible server will be able to communicate with the web server.

Now in the same way we need to setup password less authentication for jenkins to ansible.

Following the same steps

passwd root

The screenshot shows a terminal window with four tabs at the top: 'Developer', 'Jenkins', 'Ansible' (which is active), and 'Web'. The terminal content shows the user running 'ssh-copy-id' to add a public key from '/root/.ssh/id_rsa.pub' to the remote host '172.31.16.30'. It prompts for confirmation and asks for the password of the target machine. The message 'Number of key(s) added: 1' is displayed, indicating success. Below this, instructions are given to log in with 'ssh root@172.31.16.30' and verify the key was added.

```
B.o...o . |
| .oX=++o.. |
+---[SHA256]----+
[root@ip-172-31-22-43 ~]#
[root@ip-172-31-22-43 ~]#
[root@ip-172-31-22-43 ~]#
[root@ip-172-31-22-43 ~]# ssh-copy-id root@172.31.16.30
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '172.31.16.30 (172.31.16.30)' can't be established.
ECDSA key fingerprint is SHA256:P12jJLzFmH+vk0x3PeRk5ec/XSVWQDCmdXK8wvaqdGs.
ECDSA key fingerprint is MD5:23:1d:f5:36:94:05:b7:ac:ea:c9:ca:7c:c0:55:b1:f5.
Are you sure you want to continue connecting (yes/no)? yes
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.16.30's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.16.30'"
and check to make sure that only the key(s) you wanted were added.

[root@ip-172-31-22-43 ~]# passwd root
Changing password for user root.
New password: [REDACTED]
```

Now changing the configuration file “vi /etc/ssh/sshd_config”

```
[root@ip-172-31-22-43 ~]# vi /etc/ssh/sshd_config [REDACTED]
```

PermitRootLogin yes

```
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

PasswordAuthentication yes

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
```

- “**systemctl restart sshd**”

```
[root@ip-172-31-22-43 ~]# systemctl restart sshd
[root@ip-172-31-22-43 ~]#
```

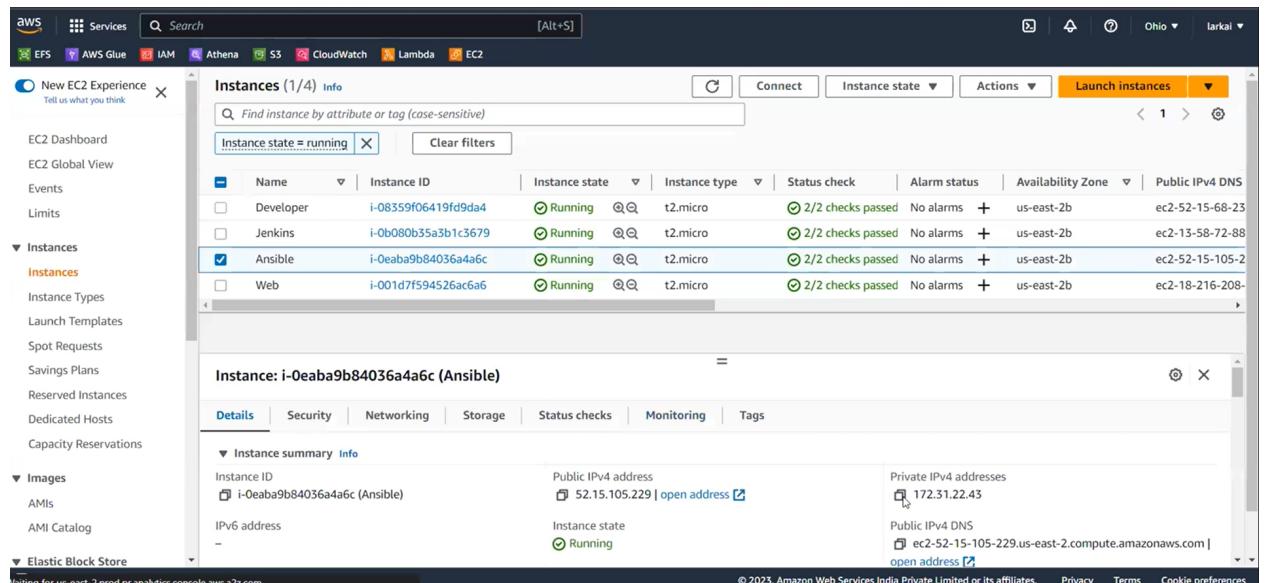
Now come to the jenkins server

- **Generate key pair**

“**ssh-keygen**”

```
4178c1ad0ce4428aa1a9a20eaee4260e
[root@ip-172-31-18-120 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:0a7SQH3KbWBBFiuU64n/eg6kNtxPHLog68kQLbfX64L0 root@ip-172-31-18-120.us-east-2.compute.internal
The key's randomart image is:
+---[RSA 2048]---+
| *o.
| + +
| o + = o
| . o + + B
| . + . o S.+
| + . o 0.00
| o 0.+.=*
| ....+.B=oo
| ooo. Eo+o..
+---[SHA256]---+
[root@ip-172-31-18-120 ~]#
[root@ip-172-31-18-120 ~]# ssh-copy-id root@172.31.22.43
```

- “**ssh-copy-id root@172.31.22.43**“ where 172.31.22.43 is the private ip address of the Ansible server.



Now give the password that we set in the Ansible server (see the ip address that whose password is being asked)

```
The key fingerprint is:  
SHA256:0a7SQH3KbWBBFiU64n/eg6kNtxPHLog68kQLbfX64L0 root@ip-172-31-18-120.us-east-2.compute.internal  
The key's randomart image is:  
+---[RSA 2048]---+  
| .*o.  
| + +  
| o += o  
| . o ++ B  
| . + . o S.+  
| + . o o.oo  
| o o.+.*  
| ....+.B=oo  
| ooo. Eo+o..  
+---[SHA256]---+  
[root@ip-172-31-18-120 ~]#  
[root@ip-172-31-18-120 ~]# ssh-copy-id root@172.31.22.43  
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"  
The authenticity of host '172.31.22.43' (172.31.22.43)' can't be established.  
ECDSA key fingerprint is SHA256:CD5055ydwV489Dk/zQzkvleuiBv+Ls+Uxf02+PNHH/8.  
ECDSA key fingerprint is MD5:57:f1:74:93:lc:06:61:57:c0:b1:65:6b:ef:72:fc:c0.  
Are you sure you want to continue connecting (yes/no)? yes  
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed  
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys  
root@172.31.22.43's password: █ I
```

Now we have established a connection between jenkins to ansible.

**Now changing the password of jenkins server “passwd root”
Then configuring the file by “vi /etc/ssh/sshd_config”**

```
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys  
root@172.31.22.43's password:  
Permission denied, please try again.  
root@172.31.22.43's password:  
Permission denied, please try again.  
root@172.31.22.43's password:  
Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).  
[root@ip-172-31-18-120 ~]# ssh-copy-id root@172.31.22.43  
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"  
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed  
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys  
root@172.31.22.43's password:  
  
Number of key(s) added: 1  
  
Now try logging into the machine, with: "ssh 'root@172.31.22.43'"  
and check to make sure that only the key(s) you wanted were added.  
  
[root@ip-172-31-18-120 ~]# passwd root  
Changing password for user root.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@ip-172-31-18-120 ~]# vi /etc/ssh/sshd_config █
```

PermitRootLogin yes

```
#LoginGraceTime 2m  
PermitRootLogin yes  
#StrictModes yes  
#MaxAuthTries 6  
#MaxSessions 10
```

PasswordAuthentication yes

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
```

```
systemctl restart sshd
```

```
[root@ip-172-31-18-120 ~]# vi /etc/ssh/sshd_config
[root@ip-172-31-18-120 ~]# systemctl restart sshd
```

Now we need to make a pipeline in a jenkins

- Click on the new item.

The screenshot shows the Jenkins dashboard. At the top left is the Jenkins logo and the word "Jenkins". To its right is a search bar with the placeholder "Search (CTRL+K)". On the far right are two small icons: a question mark and a shield. Below the header is a navigation bar with links: "Dashboard" (which is the active page, indicated by a grey background), "New Item" (highlighted with a light grey background), "People", "Build History", "Manage Jenkins", and "My Views". A dropdown menu titled "Build Queue" is open, showing the message "No builds in the queue.". To the right of the dashboard area, there's a "Welcome to Jenkins!" message with a subtext about starting a project. Below this are three buttons: "Start building your software project", "Create a job", and "Set up a distributed build". Further down are buttons for "Set up an agent" and "Configure a cloud".

**Enter an item name
Then select freestyle project and click ok**

The screenshot shows the Jenkins dashboard with a modal dialog titled "Enter an item name". Inside the dialog, the text "test-job" is entered into a text input field. A message below the field says "» Required field". Below the dialog, there are two items: "Freestyle project" with a brief description and "Pipeline".

**Now Jenkins needs to control ansible, web server like stuff.
So we need to connect jenkins with ansible and web server.**

So for that

Go to manage jenkins and click on Plugins

The screenshot shows the "Manage Jenkins" page. In the left sidebar, "Manage Jenkins" is selected. The main area displays various configuration sections: "System Configuration" (with a note about building on a built-in node), "Tools", "Nodes and Clouds", "Security", "Credentials", and "Credential Providers". The "Plugins" section is highlighted, showing a button to "Add, remove, disable or enable plugins that can extend the functionality of Jenkins".

Come to available plugins and then search for ssh

The screenshot shows the Jenkins Plugins page. A search bar at the top contains the text 'ssh'. Below the search bar, a table lists several Jenkins plugins:

Install	Name	Released
<input type="checkbox"/>	Command Agent Launcher 100.v2f6722292ee8 Agent Management	1 mo 16 days ago
<input type="checkbox"/>	Oracle Java SE Development Kit Installer 66.vd8fa_64ee9fb_d Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.	1 mo 24 days ago
<input type="checkbox"/>	JSch dependency 0.2.8-65.v052c39de79b_2 Library plugins (for use by other plugins) Miscellaneous	1 mo 9 days ago
<input type="checkbox"/>	JavaScript GUI Lib: ACE Editor bundle 1.1 JavaScript GUI lib: ACE Editor bundle plugin	

At the bottom of the table are three buttons: 'Install without restart' (highlighted in blue), 'Download now and install after restart', and 'Check now'.

- Select **Publish over SSH** and then click on **install without restart**

The screenshot shows the Jenkins Plugins page with the search bar containing 'ssh'. The 'JSch dependency' plugin is selected. Below it, the 'Publish Over SSH' plugin is highlighted with a checked checkbox. The table lists the following plugins:

Install	Name	Released
<input type="checkbox"/>	JSch dependency 0.2.8-65.v052c39de79b_2 Library plugins (for use by other plugins) Miscellaneous	1 mo 9 days ago
<input type="checkbox"/>	SSH Agent 333.v878b_53c8951 This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins.	1 mo 15 days ago
<input checked="" type="checkbox"/>	Publish Over SSH 1.24 Artifact Uploader Build Tools	1 yr 3 mo ago
<input type="checkbox"/>	SSH Pipeline Steps 2.0.65.vd26b_5b_9b_de4d pipeline	4 mo 21 days ago
<input type="checkbox"/>	SSH2 Easy 1.4 This plugin allows you to ssh2 remote server to execute linux commands , shell , sftp upload, download etc	7 yr 0 mo ago

At the bottom of the table are three buttons: 'Install without restart' (highlighted in blue), 'Download now and install after restart', and 'Check now'.

Then there will be option to restartselect restart

The screenshot shows the Jenkins 'Manage Jenkins > Plugins' page. It lists various Jenkins plugins with their status: Pipeline: KESI API (Success), Pipeline: Stage View (Success), Git (Success), SSH Build Agents (Success), Matrix Authorization Strategy (Success), PAM Authentication (Success), LDAP (Success), Email Extension (Success), Mailer (Success), Loading plugin extensions (Success), Oracle Java SE Development Kit Installer (Pending), Command Agent Launcher (Pending), Infrastructure plugin for Publish Over X (Pending), JSch dependency (Pending), Publish Over SSH (Pending), and Loading plugin extensions (Pending). Below the list are two links: '→ Go back to the top page' and '→ Restart Jenkins when installation is complete and no jobs are running'.

**Now go to manage jenkins and then system
Now we will be able to add servers to the jenkins**

The screenshot shows the Jenkins 'Manage Jenkins' page under the 'System Configuration' section. It includes links for 'System' (Configure global settings and paths), 'Tools' (Configure tools, their locations and automatic installers), and 'Plugins' (Add, remove, disable or enable plugins that can extend the functionality of Jenkins). A note at the top states: 'Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).'. There are also buttons for 'Set up agent', 'Set up cloud', and 'Dismiss'.

Path to key ?

Key ?

Disable exec ?

SSH Servers

≡ SSH Server

Name ?

Jenkins

① Required. Cannot contain < & ' " \

Hostname ?

Save

Apply

Dashboard > Manage Jenkins > System >

SSH Server

Name ? Jenkins

Hostname ? 172.31.18.120

Username ? root Required

Remote Directory ?

Advanced

Test Configuration

Save **Apply**

AWS Services Search [Alt+S] Ohio larkai

EFS AWS Glue IAM Athena S3 CloudWatch Lambda EC2

New EC2 Experience Tell us what you think

EC2 Dashboard EC2 Global View Events Limits Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations Images AMIs AMI Catalog Elastic Block Store

Instances (1/4) Info

Find instance by attribute or tag (case-sensitive)

Instance state = running Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Developer	i-08359f06419fd9da4	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2b	ec2-52-15-68-23
Jenkins	i-0b080b35a3b1c3679	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2b	ec2-13-58-72-88
Ansible	i-0eaba9b84036a4a6c	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2b	ec2-52-15-105-2
Web	i-001d7f594526ac6a6	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2b	ec2-18-216-208-

Instance: i-0b080b35a3b1c3679 (Jenkins)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID: i-0b080b35a3b1c3679 (Jenkins) Public IPv4 address: 13.58.72.88 | open address

IPv6 address: - Instance state: Running

Private IPv4 address copied: 172.31.18.120

Public IPv4 DNS: ec2-13-58-72-88.us-east-2.compute.amazonaws.com | open address

Click on advance and put password of jenkins server

The screenshot shows the Jenkins 'System' configuration page under 'Manage Jenkins'. It includes fields for 'Username' (root), 'Remote Directory', and an 'Advanced' section. In the 'Advanced' section, there is a checked checkbox for 'Use password authentication, or use a different key'. Below it is a 'Passphrase / Password' field containing '.....'. There are also fields for 'Path to key' and 'Key'. At the bottom are 'Save' and 'Apply' buttons.

**Now once we did that click on the test configuration
We can see the success message.**

The screenshot shows the same Jenkins 'System' configuration page after saving the changes. A 'Success' message is displayed above the configuration fields. To the right of the 'Proxy password' field, there is a 'Test Configuration' button. At the bottom of the page are 'Add', 'Advanced', 'Save', and 'Apply' buttons.

Similarly adding one ansible server

The screenshot shows two windows side-by-side. The left window is a Jenkins configuration page for an SSH Server. It has fields for 'Name' (Ansible) and 'Hostname' (left empty), both with validation errors. The right window is an AWS EC2 Instances list showing four instances: Developer, Jenkins, Ansible, and Web. The Ansible instance is selected, and its details are shown in a modal below.

Jenkins System Configuration (Left Window):

- Proxy password: [redacted]
- Success: Success
- Test Configuration: Test Configuration
- SSH Server:**
 - Name: Ansible
 - Hostname: [empty]
- Save: Save
- Apply: Apply

AWS EC2 Instances List (Right Window):

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Developer	i-08359f06419fd9da4	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2b	ec2-52-15-68-23
Jenkins	i-0b080b35a3b1c3679	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2b	ec2-13-58-72-88
Ansible	i-0eaba9b84036a4a6c	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2b	ec2-52-15-105-2
Web	i-001d7f594526ac6a6	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-2b	ec2-18-216-208-

Ansible Instance Details (Modal):

Details	Security	Networking	Storage	Status checks	Monitoring	Tags
Instance summary	Info					
Instance ID	i-0eaba9b84036a4a6c (Ansible)	Public IPv4 address	52.15.105.229 open address			
IPv6 address	-	Instance state	Running			

Private IPv4 address copied: 172.31.22.43
Private IPv4 addresses: 172.31.22.43
Public IPv4 DNS: ec2-52-15-105-229.us-east-2.compute.amazonaws.com | open address

Dashboard > Manage Jenkins > System >

172.31.22.43

Username ?
root
Required

Remote Directory ?
[empty field]

Advanced ▾

Test Configuration

Add

Advanced ▾

Save **Apply**

Dashboard > Manage Jenkins > System >

172.31.22.43

Username ?
root

Remote Directory ?
[empty field]

Advanced ^ Edited

Use password authentication, or use a different key ?
Passphrase / Password ?
.....

Path to key ?
[empty field]

Key ?
[empty field]

Save **Apply**

After test configuration apply and save

Now come to test-job

The screenshot shows the Jenkins dashboard. At the top, there's a search bar with placeholder text "Search (CTRL+K)" and a "log out" button. Below the header, a navigation bar includes links for "Dashboard", "New Item", "People", "Build History", "Manage Jenkins", and "My Views". A "Build Queue" section indicates "No builds in the queue". A "Build Executor Status" section shows 1 idle executor. The main content area displays a table for the "test-job" project, which has a status icon of a sun, a name of "test-job", and last build information: "Last Success" (N/A), "Last Failure" (N/A), and "Last Duration" (N/A). There are "Add description" and "Edit" buttons at the top right of the table. Below the table, there are links for "Atom feed for all", "Atom feed for failures", and "Atom feed for just latest builds".

Go to configure

The screenshot shows the "Project test-job" configuration page. At the top, there's a "Status" tab (highlighted in grey) and a "Changes" tab. Below the tabs, there are links for "Workspace", "Build Now", "Configure" (which is currently selected and highlighted in grey), "Delete Project", and "Rename". A "Permalinks" section is also present. The main content area is titled "Build History" and shows a "trend" dropdown set to "trend". It includes a "Filter builds..." input field and a "No builds" message. At the bottom, there are links for "Atom feed for all", "Atom feed for failures", and navigation icons (up, down, left, right).

The screenshot shows the Jenkins job configuration page for a job named "test-job". Under the "Source Code Management" section, the "Git" option is selected. The "Repository URL" field contains the value "https://github.com/sumitsah9263/devops-batch-2.git". A red error message below the field says "Please enter Git repository." There are "Save" and "Apply" buttons at the bottom.

The screenshot shows a GitHub repository page for "sumitsah9263 / devops-batch-2". In the "Code" tab, the "Clone" section displays the HTTPS URL "https://github.com/sumitsah9263/devops-batch-2". A tooltip indicates that this URL can be used with Git or SVN.

we see error because there is no git package installed in the jenkins server

Dashboard > test-job > Configuration

Configure

Source Code Management

- General
- Source Code Management**
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Source Code Management

Git

Repositories

Repository URL: `https://github.com/sumitsah9263/devops-batch-2.git`

Credentials: - none -

Advanced

Save Apply

```

root@172.31.22.43's password:
Permission denied, please try again.
root@172.31.22.43's password:
Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
[root@ip-172-31-18-120 ~]# ssh-copy-id root@172.31.22.43
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@172.31.22.43's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.22.43'"
and check to make sure that only the key(s) you wanted were added.

[root@ip-172-31-18-120 ~]# passwd root
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-18-120 ~]# vi /etc/ssh/sshd_config
[root@ip-172-31-18-120 ~]# systemctl restart sshd
[root@ip-172-31-18-120 ~]# yum install git
  
```

now

we can see that there is no error

Dashboard > test-job > Configuration

Configure

- General
- Source Code Management**
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Source Code Management

Git

Repositories

Repository URL: `https://github.com/sumitsah9263/devops-batch-2.git`

Credentials: - none -

Add

Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any') `*/master`

Save Apply

Change master to main

Dashboard > test-job > Configuration

Configure

- General
- Source Code Management**
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Repositories

Repository URL: https://github.com/sumitsah9263/devops-batch-2.git

Credentials: - none -

Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any'): */main

Save Apply

Dashboard > test-job > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment**
- Build Steps
- Post-build Actions

Use secret text(s) or file(s)

- Send files or execute commands over SSH before the build starts
- Send files or execute commands over SSH after the build runs
- Add timestamps to the Console Output
- Inspect build log for published build scans
- Terminate a build if it's stuck
- With Ant

Build Steps

Add build step ▾

Filter

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Send files or execute commands over SSH**
- Set build status to "pending" on GitHub commit

13.58.72.88:8080/job/test-job/configure#

Dashboard > test-job > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

Build Steps

SSH Publishers

SSH Server Name ? Jenkins

Advanced

Transfers

Transfer Set Source files ?

Either Source files, Exec command or both must be supplied

Remove prefix ?

Save Apply

This screenshot shows the Jenkins configuration interface for a job named 'test-job'. The 'Build Steps' section is selected. It contains two main sections: 'SSH Publishers' and 'Transfers'. In the 'SSH Publishers' section, there is a dropdown menu for 'SSH Server' with 'Jenkins' selected. Below it is an 'Advanced' button. In the 'Transfers' section, there is a 'Transfer Set' dropdown with 'Source files' selected. A warning message 'Either Source files, Exec command or both must be supplied' is displayed. There is also a 'Remove prefix' input field. At the bottom of the page, there are 'Save' and 'Apply' buttons.

Dashboard > test-job > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

Build Steps

Transfers

Transfer Set Source files ?

Either Source files, Exec command or both must be supplied

Remove prefix ?

Remote directory ?

Exec command ?

sync -avh /var/lib/jenkins/workspace/demo-project/ root@172.31.87.227:/opt/*

Either Source files, Exec command or both must be supplied

All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables

Save Apply

This screenshot shows the Jenkins configuration interface for a job named 'test-job'. The 'Build Steps' section is selected. It contains a 'Transfers' section. In the 'Transfers' section, there is a 'Transfer Set' dropdown with 'Source files' selected. A warning message 'Either Source files, Exec command or both must be supplied' is displayed. There is also a 'Remove prefix' input field and a 'Remote directory' input field. Below these, there is an 'Exec command' input field containing the command 'sync -avh /var/lib/jenkins/workspace/demo-project/ root@172.31.87.227:/opt/*'. Another warning message 'Either Source files, Exec command or both must be supplied' is displayed. At the bottom of the page, there are 'Save' and 'Apply' buttons.

Dashboard > test-job > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

Build Steps

Transfers

Transfer Set Source files ?

Either Source files, Exec command or both must be supplied

Remove prefix ?

Remote directory ?

Exec command ?

sync -avh /var/lib/jenkins/workspace/demo-project/ root@172.31.87.227:/opt/*

Either Source files, Exec command or both must be supplied

All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables

Save Apply

This screenshot shows the Jenkins configuration interface for a job named 'test-job'. The 'Build Steps' section is selected. It contains a 'Transfers' section. In the 'Transfers' section, there is a 'Transfer Set' dropdown with 'Source files' selected. A warning message 'Either Source files, Exec command or both must be supplied' is displayed. There is also a 'Remove prefix' input field and a 'Remote directory' input field. Below these, there is an 'Exec command' input field containing the command 'sync -avh /var/lib/jenkins/workspace/demo-project/ root@172.31.87.227:/opt/*'. Another warning message 'Either Source files, Exec command or both must be supplied' is displayed. At the bottom of the page, there are 'Save' and 'Apply' buttons.

Changing demo-project to test-job and writing index.html and changing the ip address(private) of the ansible server.

Exec command ?

```
rsync -avh /var/lib/jenkins/workspace/test-job/index.html root@172.31.22.43:/opt/index.html
```

The screenshot shows the AWS EC2 Instances page with four instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Developer	i-083590641969bd4d	Running	t2.micro	2/2 checks passed	No alarms	us-east-2b	ec2-52-15-68-25
Jenkins	i-0c08055a51c15e79	Running	t2.micro	2/2 checks passed	No alarms	us-east-2b	ec2-15-58-72-88
Ansible	i-0ca0b984036a4a6c	Running	t2.micro	2/2 checks passed	No alarms	us-east-2b	ec2-52-15-105-2
Web	i-0016f7594526a6af	Running	t2.micro	2/2 checks passed	No alarms	us-east-2b	ec2-16-216-208-

Humlog jenkins ke file ko ansible ke sath sync kar rhe aur jenkins ka location diya hua hai jahan pe index.html hai aur ansible ka location diya hua hai jahan pe file ko move kr rhe hai

We forgot to do one more configuration
So check the verbose output in console

The screenshot shows the Jenkins job configuration page under 'Build Steps'. In the 'SSH Publishers' section, the 'Advanced' tab is selected, and the 'Verbose output in console' checkbox is checked.

By enabling this we can see these stuffs

Jenkins

Dashboard > test-job > #1 > Console Output

Status Changes

Console Output

View as plain text

Edit Build Information

Delete build '#1'

Git Build Data

Console Output

```
Started by user sks
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/test-job
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/sumitsah9263/devops-batch-2.git
> git init /var/lib/jenkins/workspace/test-job # timeout=10
Fetching upstream changes from https://github.com/sumitsah9263/devops-batch-2.git
> git --version # timeout=10
> git --version # 'git version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/sumitsah9263/devops-batch-2.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/sumitsah9263/devops-batch-2.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 13b125e386308773e26af56ae3d5fd7118a8d30 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 13b125e386308773e26af56ae3d5fd7118a8d30 # timeout=10
Commit message: "test commit"
First time build. Skipping changelog.
SSH: Connecting from host [ip-172-31-18-120.us-east-2.compute.internal]
SSH: Connecting with configuration [Jenkins] ...
```

Ye iss name ka ek folder bana dega

Console Output

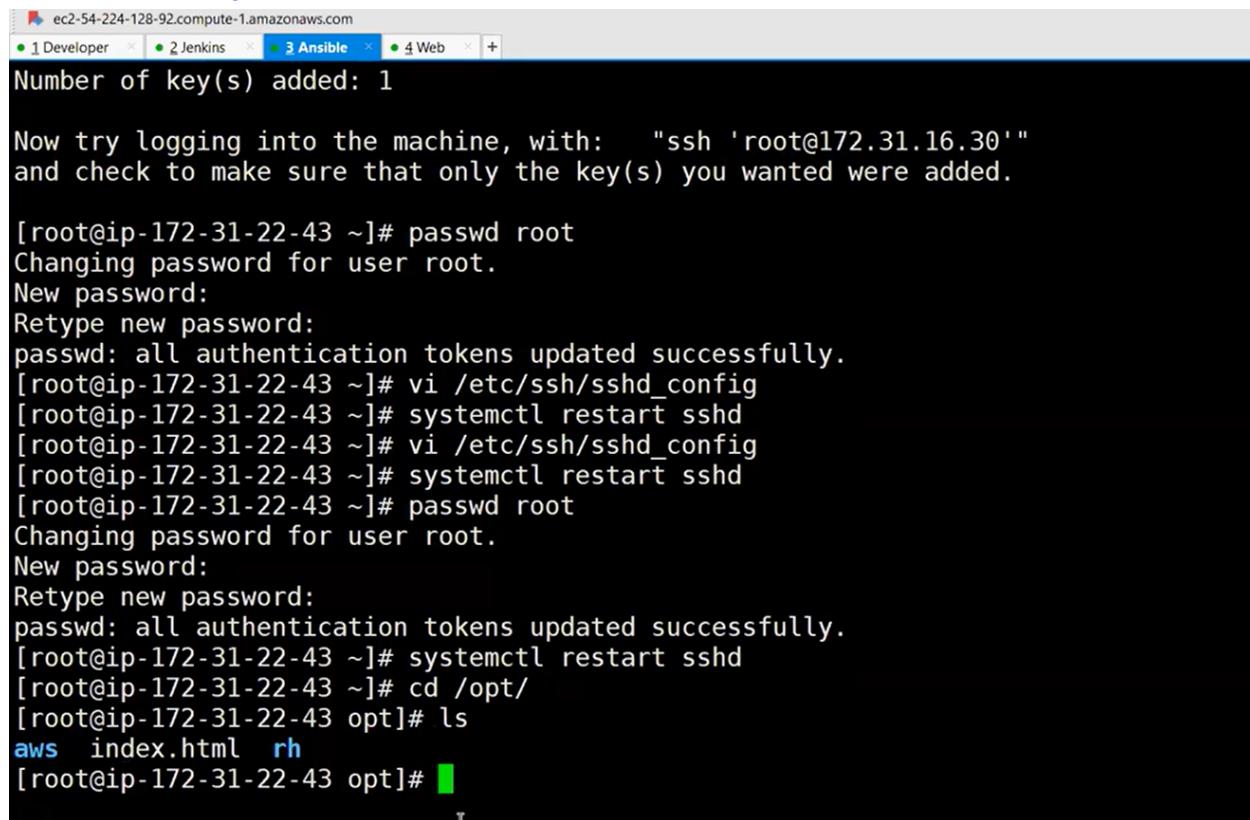
```
Started by user sks
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/test-job
The recommended git tool is: NONE
No credentials specified
```

If we check it in the path we can see the index.html file

```
• 1 Developer ✘ • 2 Jenkins ✘ • 3 Ansible ✘ • 4 Web ✘ +  
Installing : 1:perl-Error-0.17020-2.amzn2.noarch  
Installing : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64  
Installing : perl-Git-2.40.1-1.amzn2.0.1.noarch  
Installing : git-2.40.1-1.amzn2.0.1.x86_64  
Verifying  : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64  
Verifying  : git-core-2.40.1-1.amzn2.0.1.x86_64  
Verifying  : git-core-doc-2.40.1-1.amzn2.0.1.noarch  
Verifying  : perl-Git-2.40.1-1.amzn2.0.1.noarch  
Verifying  : 1:perl-Error-0.17020-2.amzn2.noarch  
Verifying  : git-2.40.1-1.amzn2.0.1.x86_64  
  
Installed:  
git.x86_64 0:2.40.1-1.amzn2.0.1  
  
Dependency Installed:  
git-core.x86_64 0:2.40.1-1.amzn2.0.1  
perl-Error.noarch 1:0.17020-2.amzn2  
perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2  
  
Complete!  
[root@ip-172-31-18-120 ~]# cd /var/lib/jenkins/workspace/test-job  
[root@ip-172-31-18-120 test-job]# ls  
index.html  
[root@ip-172-31-18-120 test-job]# █
```

```
SSH: Opening exec channel ...
SSH: EXEC: channel open
SSH: EXEC: STDOUT/STDERR from command [rsync -avh /var/lib/jenkins/workspace/test-job/index.html root@172.31.22.43:/opt/index.html] ...
SSH: EXEC: connected
```

We have moved jenkins file to ansible so we can see the files in ansible server also



The screenshot shows a terminal window titled 'ec2-54-224-128-92.compute-1.amazonaws.com' with four tabs: '1 Developer', '2 Jenkins', '3 Ansible', and '4 Web'. The '3 Ansible' tab is active, displaying the following text:

```
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@172.31.16.30'"
and check to make sure that only the key(s) you wanted were added.

[root@ip-172-31-22-43 ~]# passwd root
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-22-43 ~]# vi /etc/ssh/sshd_config
[root@ip-172-31-22-43 ~]# systemctl restart sshd
[root@ip-172-31-22-43 ~]# vi /etc/ssh/sshd_config
[root@ip-172-31-22-43 ~]# systemctl restart sshd
[root@ip-172-31-22-43 ~]# passwd root
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-22-43 ~]# systemctl restart sshd
[root@ip-172-31-22-43 ~]# cd /opt/
[root@ip-172-31-22-43 opt]# ls
aws  index.html  rh
[root@ip-172-31-22-43 opt]#
```

Ab ansible me ek play book banayenge

Play book ek file hota hai

File banane ke liye following steps

```
[root@ip-172-31-22-43 opt]# mkdir /srccode
[root@ip-172-31-22-43 opt]# cd /srccode/
[root@ip-172-31-22-43 srccode]# vi playbook.yml
```

```
● 1 Developer ✘ ● 2 Jenkins ✘ ● 3 Ansible ✘ ● 4 Web ✘ +  
- hosts: all  
  tasks:  
    - copy:  
        src: /opt/index.html  
        dest: /var/www/html
```

Now on configure Post build actions
Send build artifacts over ssh

The screenshot shows the Jenkins job configuration interface for a job named "test-job". The left sidebar lists various configuration sections: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The "Post-build Actions" section is currently selected and highlighted in grey. A dropdown menu is open under "Post-build Actions", showing a list of available actions. The action "Send build artifacts over SSH" is highlighted with a blue selection bar and has a mouse cursor pointing at it. Other actions listed include: Aggregate downstream test results, Archive the artifacts, Build other projects, Publish JUnit test result report, Record fingerprints of files to track usage, Git Publisher, E-mail Notification, Editable Email Notification, Set GitHub commit status (universal), Set build status on GitHub commit [deprecated], and Delete workspace when build is done. At the bottom of the dropdown menu is a button labeled "Add post-build action ▾". Below the dropdown, there are "Save" and "Apply" buttons. The footer of the page displays the URL "13.58.72.88:8080/job/test-job/configure#" and the Jenkins version "Jenkins 2.401.1".

We select the ansible server

The screenshot shows the Jenkins configuration interface for a job named "test-job". The left sidebar lists various configuration sections: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The "Post-build Actions" section is currently selected. At the top of this section is a button labeled "Add build step ▾". Below it, there is a heading "Post-build Actions" followed by a section titled "Send build artifacts over SSH ?". This section contains fields for "SSH Server Name" (set to "Ansible") and "Transfer Set Source files". A red "X" icon is visible in the top right corner of this section. At the bottom of the "Post-build Actions" section are "Save" and "Apply" buttons.

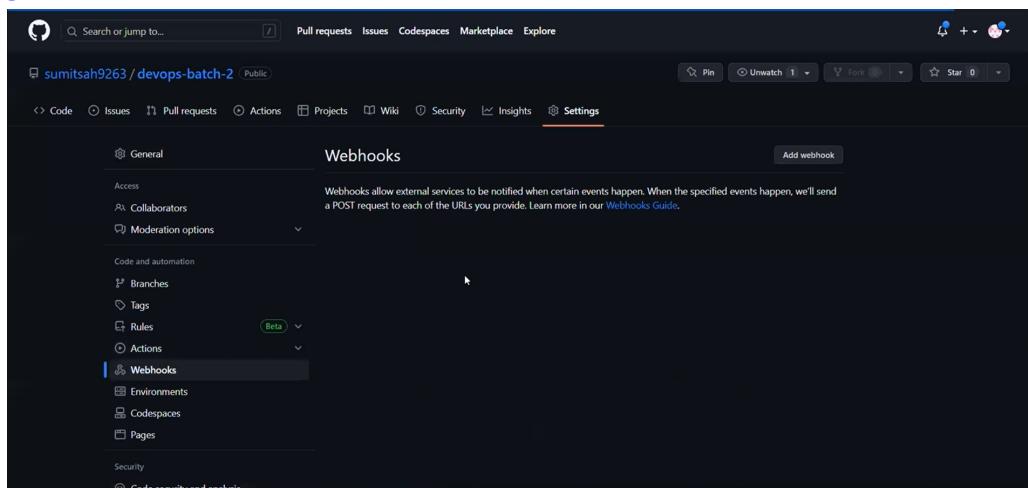
Copy the code

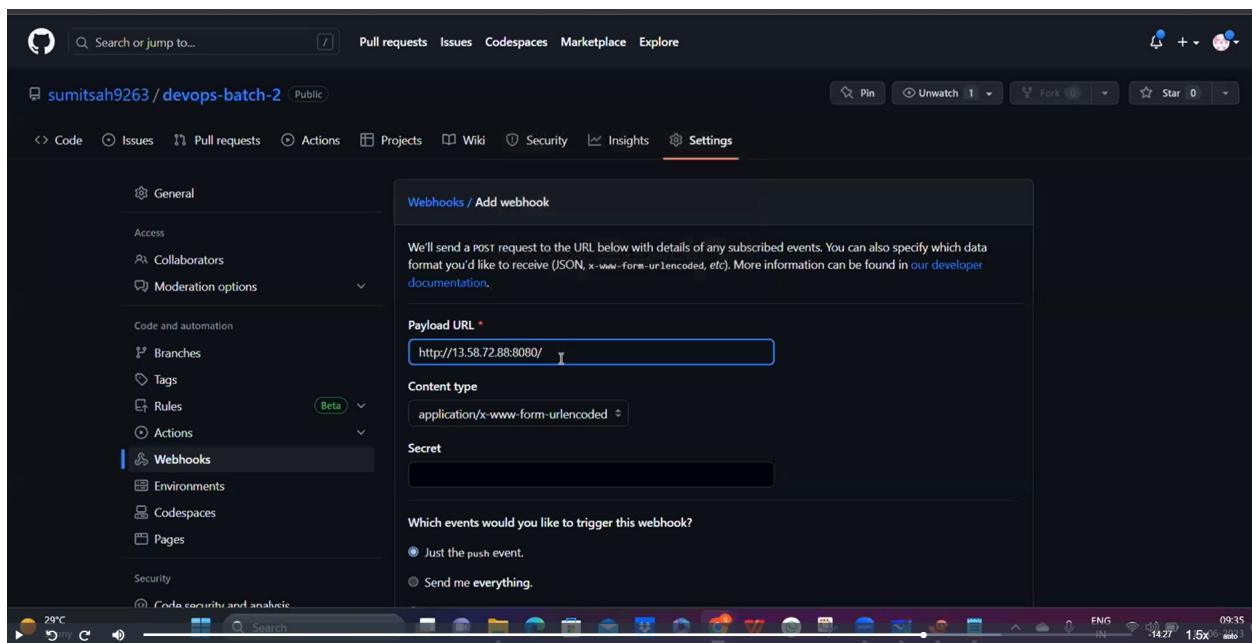
This screenshot shows the same Jenkins configuration interface for the "test-job" job. The "Post-build Actions" section is selected. In the "Exec command" field, the value "ansible-playbook /second/ansible/playbook.yml" has been entered. A red error message "Either Source files, Exec command or both must be supplied" is displayed above the "Remote directory" field. Below the "Exec command" field, another red message "Either Source files, Exec command or both must be supplied" is shown. A note at the bottom states "All of the transfer fields (except for Exec timeout) support substitution of Jenkins environment variables". At the bottom of the "Post-build Actions" section are "Save" and "Apply" buttons.

```
• 1 Developer • 2 Jenkins • 3 Ansible • 4 Web • +  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@ip-172-31-22-43 ~]# vi /etc/ssh/sshd_config  
[root@ip-172-31-22-43 ~]# systemctl restart sshd  
[root@ip-172-31-22-43 ~]# vi /etc/ssh/sshd_config  
[root@ip-172-31-22-43 ~]# systemctl restart sshd  
[root@ip-172-31-22-43 ~]# passwd root  
Changing password for user root.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@ip-172-31-22-43 ~]# systemctl restart sshd  
[root@ip-172-31-22-43 ~]# cd /opt/  
[root@ip-172-31-22-43 opt]# ls  
aws index.html rh  
[root@ip-172-31-22-43 opt]# cat index.html  
Hello welcome to ITW silicon  
test second  
[root@ip-172-31-22-43 opt]# mkdir /srccode  
[root@ip-172-31-22-43 opt]# cd /srccode/  
[root@ip-172-31-22-43 srccode]# vi playbook.yml  
[root@ip-172-31-22-43 srccode]# pwd  
/srccode  
[root@ip-172-31-22-43 srccode]# █
```

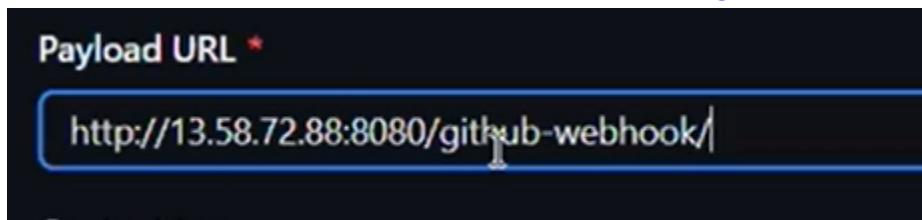
Now if we hit the public id of the web server we can see the content

So now we need to add webhooks so that any updates get automatically pushed to the gituhub and shown in the web server. Add webhooks





Jenkins ka url daalna hai aur ek chiz add krna hai "github-webhook/"



A screenshot of the Jenkins job configuration for 'test-job'. The 'General' tab is selected. In the 'Build Triggers' section, a context menu is open over the 'GitHub project' checkbox. The menu includes options like Cut, Copy, Paste, Select all, Undo, and Delete. Below the menu, there are additional build trigger options: 'Discard old builds', 'GitHub project', 'This project is parameterised', 'Throttle builds', and 'Execute concurrent builds if necessary'. At the bottom of the screen, there are 'Save' and 'Apply' buttons.

Now we need to give secret key , so for that go to the jenkins configure
Click generate

The screenshot shows the Jenkins 'Configure' screen for a user named 'sks'. On the left sidebar, 'Configure' is selected. The main area has two sections: 'Full Name' (sks) and 'Description' (empty). Below this is the 'API Token' section, which displays a message: 'There are no registered tokens for this user.' It includes a 'Default name' input field, a 'Generate' button, and 'Save' and 'Apply' buttons.

Then we have to just copy this and apply

API Token

Current token(s) ?

Token created on 2023-06-13T04:06:12.7

119edbf62f7d342eed4de626c29b38298



⚠ Copy this token now, because it cannot be recovered in the future.

Save

Apply

Payload URL *

http://13.58.72.88:8080/github-webhook/

Content type

application/json



Secret

1119edbf62f7d342eed4de626c29b38298

Which events would you like to trigger this webhook?

- Just the push event.
- Send me **everything**.
- Let me select individual events.

Active

We will deliver event details when this hook is triggered.

Add webhook

Dashboard > test-job > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build Triggers

Trigger builds remotely (e.g., from scripts) ?

Build after other projects are built ?

Build periodically ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Build Environment

Delete workspace before build starts

Use secret text(s) or file(s) ?

Send files or execute commands over SSH before the build starts ?

Send files or execute commands over SSH after the build runs ?

Add timestamps to the Console Output

Save **Apply**

Now everything is automated. If we push anything in the github then that will automatically be kept track by jenkins and reflected to the web server.