

6. Set up a Jenkins and run few sample jobs on a different server for deployments (download sample code from git and need to deploy in remote server)

Prerequisites:

- Jenkins installed and configured on a server (e.g., Jenkins Server)
- Git repository set up (e.g., Git-hub)
- Remote server with SSH access (e.g., Deployment Server)

Step 1: Install and Configure Jenkins

- Install Jenkins on the Jenkins Server
- Configure Jenkins by setting up the administrator account and plugins

Step 2: Create a New Jenkins Job

- Log in to Jenkins and click on "New Item"
- Enter a job name (e.g., "Demo") and select "Freestyle project"
- Click "OK"

Step 3: Add Credentials and publish over ssh

- Dashboard-Manage Jenkins-Credentials-Click on global (Add details e.g.name,kind,username,Private key)
- Dashboard-Manage Jenkins-add Jenkins URL(e.g,<HTTP://65.1.85.248:8080/>)-Publish over SSH-add SSH server-Name(Ubuntu)-Host name(e.g.<65.1.85.248>)-username(**Ubuntu**)-Remote Directory(</var/www/myapp>)-click on Apply and Save.
- Dashboard-Manage Keys-Security-Git Host Key Verification Configuration-Manually Provided Keys-apply-save.

Step 4: Configure Git Repository

- In the job configuration page, scroll down to the "Source Code Management" section
- Select "Git" and enter the repository URL (e.g.,(<https://git-hub.com/Noorjahan153/github-ass1.git>))
- Adding webhook from github (<http://65.1.85.248:8080/github-webhook/>)
- Click "Save"
- Credentials-(e.g,Ubuntu)
- Build Triggers-(Git-hub hook trigger for GiTScm polling)
- Build Environment(Send files or execute command over SSH after the build runs)
- Add SSH Server(E.g,Name,Source Files,etc..)
- Build Steps-Execute Shell-Apply-Save.

Step 6: Run the Job

- Click on "Build Now" to run the job
- Jenkins will clone the Git repository and deploy the code to the remote server using SSH

Expected Outcome:

- Jenkins will successfully deploy the code to the remote server
- The deployment will be successful, and the log will show "Finished: SUCCESS"
- Test the Deployment(Open a web browser e.g,(<http://65.1.85.248/Jenkins.html>))

Commands

- Installing Jenkins

```
1 sudo apt update
2 sudo apt install fontconfig openjdk-17-jre
3 java -version
4 sudo wget -O /usr/share/keyrings/jenkins-keyring.asc
https://pkg.jenkins.io/debian/jenkins.io-2023.key
5 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
6 sudo apt-get update
7 sudo apt-get install jenkins
8 sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

- Create SSH directory for Jenkins

```
9 sudo mkdir -p /var/lib/jenkins/.ssh
```

- Add remote server to known_hosts to avoid SSH prompts

```
10 sudo ssh-keyscan -H 65.1.85.248 | sudo tee -a /var/lib/jenkins/.ssh/known_hosts
```

- Set permissions for Jenkins SSH directory and files

```
11 sudo chmod 700 /var/lib/jenkins/.ssh
sudo chmod 644 /var/lib/jenkins/.ssh/known_hosts
sudo chown -R jenkins:jenkins /var/lib/jenkins/.ssh
```

- Switch to Jenkins user and test SSH connection

```
12 sudo -u jenkins -i
ssh -i /var/lib/jenkins/.ssh/id_rsa ubuntu@65.1.85.248
```

- Install Apache on the remote server

```
13 sudo apt install apache2
```

- Move Jenkins.html to Apache's web directory

```
14 sudo mv /var/www/html/Jenkins.html /home/ubuntu/deployment/
sudo mv /home/ubuntu/deployment/Jenkins.html /var/www/html/
```

- Set permissions for the moved file

```
15 sudo chmod 644 /var/www/html/Jenkins.html
sudo chown www-data:www-data /var/www/html/Jenkins.html
```

- Restart Apache to apply changes

```
16 sudo systemctl restart apache2
```

- Copy Jenkins.html from Jenkins workspace to the remote server

```
17 scp -i /var/lib/jenkins/.ssh/id_rsa /var/lib/jenkins/workspace/RemoteServer/Jenkins.html
ubuntu@65.1.85.248/home/ubuntu/deployment/
```

The screenshot displays three separate browser windows showing the AWS CloudShell interface, the Jenkins sign-in screen, and the Jenkins instance configuration screen.

AWS CloudShell:

- The top window shows the AWS CloudShell interface with multiple tabs open, including "Instances | EC2 | ap-south-1", "EC2 Instance Connect", "Instances | EC2 | ap-south-1", "RemoteServer #21 Co...", "404 Not Found", "404 Not Found", "My Static Site", and "Mumbai".
- The sidebar on the left lists various AWS services: EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups), and CloudShell Feedback.
- The main content area shows the "Instances (1/2) Info" page for a single instance named "Jenkins Instance" (ID: i-061fd6fce712162cb). The instance is listed as "Running" (t2.micro), with 2/2 checks passed. It has a Public IPv4 DNS of "ec2-65-1-85-248.ap-south-1.amazonaws.com".

Jenkins Sign-in:

- The middle window shows the Jenkins sign-in screen titled "Sign in to Jenkins". It features a cartoon character holding a coffee cup on a colorful background.
- The form includes fields for "Username" (with placeholder "jenkins") and "Password".
- Checkboxes include "Keep me signed in" and "Remember me".
- A blue "Sign in" button is at the bottom.

Jenkins Instance Configuration:

- The bottom window shows the "Getting Started" screen for Jenkins instance configuration.
- The title is "Instance Configuration".
- The "Jenkins URL:" field contains "http://65.2.186.22:8080/".
- Text below the field explains that the Jenkins URL is used for root links to Jenkins resources and provides instructions for setting it correctly.
- At the bottom, there are buttons for "Not now" and "Save and Finish".

The screenshot shows the Jenkins dashboard at 65.2.186.228:8080. The main header includes tabs for 'Console Home', 'Launch an instance', 'Instances | EC2', 'EC2 Instance Connect', '(1) Day-19 | Jenkins', 'Jenkins-Zero-To...', 'Dashboard [jenkins]', 'New Tab', and 'log out'. The dashboard features a sidebar with links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. It also displays sections for 'Build Queue' (empty), 'Build Executor Status' (0/2), 'Create a job', 'Set up a distributed build', 'Start building your software project', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.

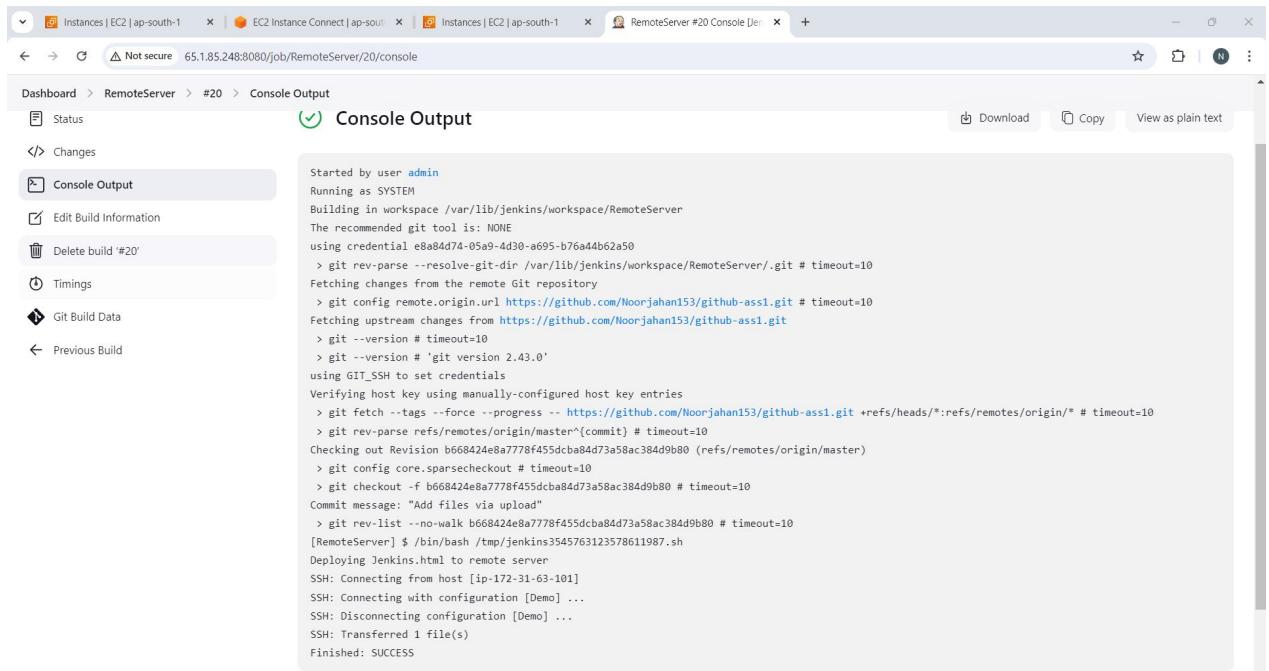
The screenshot shows the configuration page for a Jenkins job named 'RemoteServer'. The 'General' configuration tab is selected, showing the 'Enabled' status (checked). The 'Source Code Management' section is set to 'GitHub project' with the URL <https://github.com/Noorjahan153/github-ass1/blob/master/Jenkins.html>. Other options like 'Discard old builds', 'Advanced', 'Parameterized', and 'Throttle builds' are available but unchecked. At the bottom are 'Save' and 'Apply' buttons.

The screenshot shows the 'Build Steps' configuration tab for the 'RemoteServer' job. A single 'Execute shell' step is defined with the following command:

```
#!/bin/bash
# Define variables
WORKSPACE_PATH="/var/lib/jenkins/workspace/RemoteServer/"
REMOTE_USER="ubuntu"
REMOTE_SERVER="65.1.85.248"
DEPLOYMENT_DIR="/home/ubuntu/deployment/"

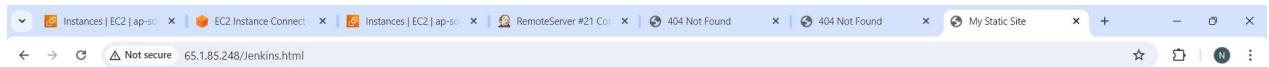
# Deploying files to remote server
echo "Deploying Jenkins.html to remote server"
scp -i ~/.ssh/id_rsa ${WORKSPACE_PATH}Jenkins.html ${REMOTE_USER}@${REMOTE_SERVER}:${DEPLOYMENT_DIR}
```

The 'Advanced' button is visible below the command. At the bottom of the page are 'Save' and 'Apply' buttons.



The screenshot shows the Jenkins console output for build #20. The build was started by user `admin` and is running as `SYSTEM`. It is building in workspace `/var/lib/jenkins/workspace/RemoteServer`. The recommended git tool is `NONE`, and it is using credential `e8a84d74-05a9-4d30-a695-b76a44b62a50`. The log shows the execution of several git commands to fetch from a GitHub repository and update Jenkins's configuration. It also shows the deployment of `Jenkins.html` to a remote server via SSH. The build has completed successfully.

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/RemoteServer
The recommended git tool is: NONE
using credential e8a84d74-05a9-4d30-a695-b76a44b62a50
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/RemoteServer/.git # timeout=10
> git config remote.origin.url https://github.com/Noorjahan153/github-ass1.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.fetch +refs/heads/*:refs/remotes/origin/*
Fetching upstream changes from https://github.com/Noorjahan153/github-ass1.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
using GIT_SSH to set credentials
Verifying host key using manually-configured host key entries
> git fetch --tags --force --progress -- https://github.com/Noorjahan153/github-ass1.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision b668424e8a7778f455dcba84d73a58ac384d9b80 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f b668424e8a7778f455dcba84d73a58ac384d9b80 # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk b668424e8a7778f455dcba84d73a58ac384d9b80 # timeout=10
[RemoteServer] $ /bin/bash /tmp/jenkins3545763123578611987.sh
Deploying Jenkins.html to remote server
SSH: Connecting from host [ip-172-31-63-101]
SSH: Connecting with configuration [Demo] ...
SSH: Disconnecting configuration [Demo] ...
SSH: Transferred 1 file(s)
Finished: SUCCESS
```



Welcome to My Static Site!

This is a sample static website deployed using Jenkins.