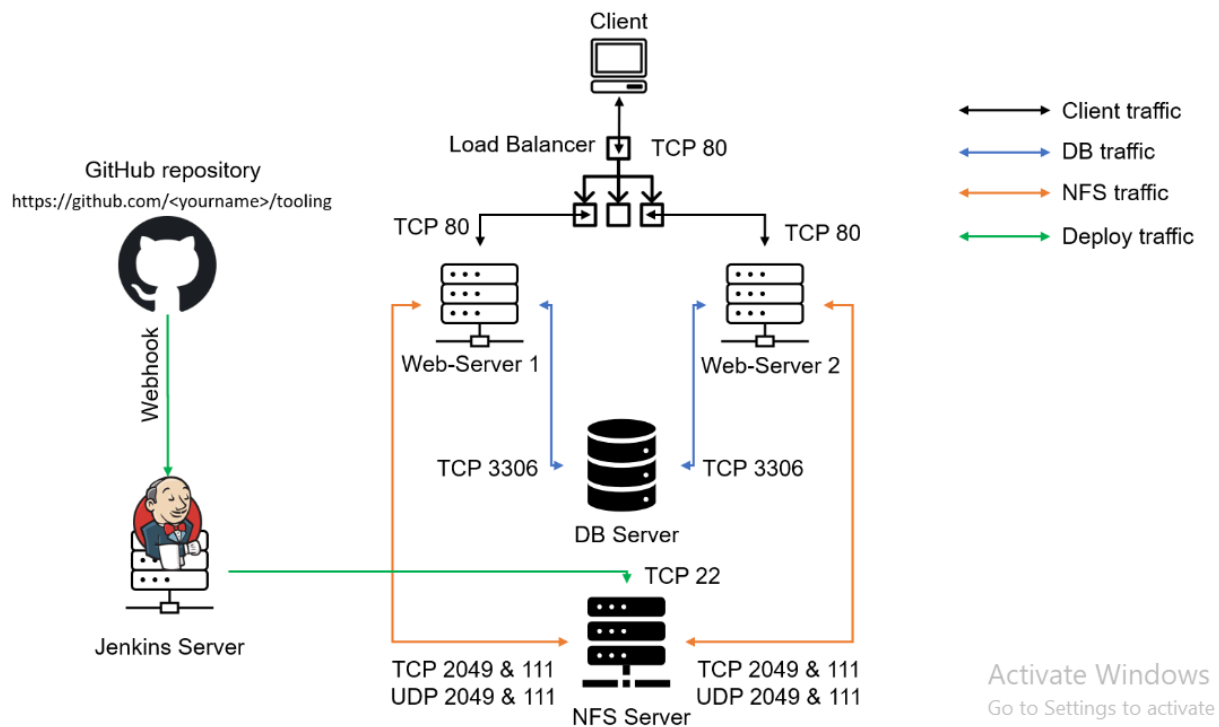


PROJECT 9



Continuous Integration Pipeline For Tooling Website

INSTALL AND CONFIGURE JENKINS SERVER

Step1: Install Jenkins server

- e an AWS EC2 server based on Ubuntu Server 20.04 LTS and name it "Jenkins"

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public
<input type="checkbox"/>	Pro7Webserver2	i-063d1bcac2917d135	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-16
<input type="checkbox"/>	Pro7WebServer1	i-0f880dbd41afd8b21	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-13
<input type="checkbox"/>	Pro7NFS-Server	i-00ba3bad502f4418f	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-13
<input type="checkbox"/>	Pro7dbServer	i-07aebc42bf18319b2	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-16
<input type="checkbox"/>	Pro8Apache-LB	i-06cb7df56ffeb35d3	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-13
<input checked="" type="checkbox"/>	Jenkins	i-0c71950a405073860	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	ec2-16

- Install **JDK** (since Jenkins is a Java-based application)

```
sudo apt update
```

```
sudo apt install default-jdk-headless
```

```
ubuntu@ip-172-31-38-178:~$ sudo apt install default-jdk-headless
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java default-jre-headless fontconfig-config fonts-dejavu-core java-common libavahi-client3 libavahi-common-data libavahi-common3 libcups2
  libfontconfig1 libgraphite2-3 libharfbuzz0b libjpeg-turbo8 libjpeg8 liblcms2-2 libpcsclite1 openjdk-11-jdk-headless openjdk-11-jre-headless
Suggested packages:
  default-jre cups-common liblcms2-utils pcscd openjdk-11-demo openjdk-11-source libnss-mdns fonts-dejavu-extra fonts-ipafont-gothic fonts-ipafont-mincho
  fonts-wqy-microhei | fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  ca-certificates-java default-jdk-headless default-jre-headless fontconfig-config fonts-dejavu-core java-common libavahi-client3 libavahi-common-data
  libavahi-common3 libcups2 libfontconfig1 libgraphite2-3 libharfbuzz0b libjpeg-turbo8 libjpeg8 liblcms2-2 libpcsclite1 openjdk-11-jdk-headless
  openjdk-11-jre-headless
```

- Install Jenkins

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key |
sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install -y jenkins
```

```
ubuntu@ip-172-31-38-178:~$ sudo systemctl status jenkins
* jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-07-17 17:54:14 UTC; 49s ago
     Main PID: 5289 (java)
       Tasks: 42 (limit: 1111)
      Memory: 304.6M
    CGroup: /system.slice/jenkins.service
            └─5289 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Jul 17 17:53:34 ip-172-31-38-178 jenkins[5289]: 5469278a32464534892d38c853336790
Jul 17 17:53:34 ip-172-31-38-178 jenkins[5289]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Jul 17 17:53:34 ip-172-31-38-178 jenkins[5289]: *****
Jul 17 17:53:34 ip-172-31-38-178 jenkins[5289]: *****
Jul 17 17:53:34 ip-172-31-38-178 jenkins[5289]: *****
Jul 17 17:54:14 ip-172-31-38-178 jenkins[5289]: 2023-07-17 17:54:14.914+0000 [id=30] INFO jenkins.InitReactorRunner$1onAttained: Completed initialization
Jul 17 17:54:14 ip-172-31-38-178 jenkins[5289]: 2023-07-17 17:54:14.935+0000 [id=22] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
Jul 17 17:54:14 ip-172-31-38-178 system[1]: Started Jenkins Continuous Integration Server.
Jul 17 17:54:14 ip-172-31-38-178 jenkins[5289]: 2023-07-17 17:54:15.331+0000 [id=46] INFO h.m.DownloadService$Downloadable$StatusObserver.onUpdate: Windows updated
Jul 17 17:54:15 ip-172-31-38-178 jenkins[5289]: 2023-07-17 17:54:15.331+0000 [id=46] INFO hudson.util.Retrier#start: Performed the action check updates
```

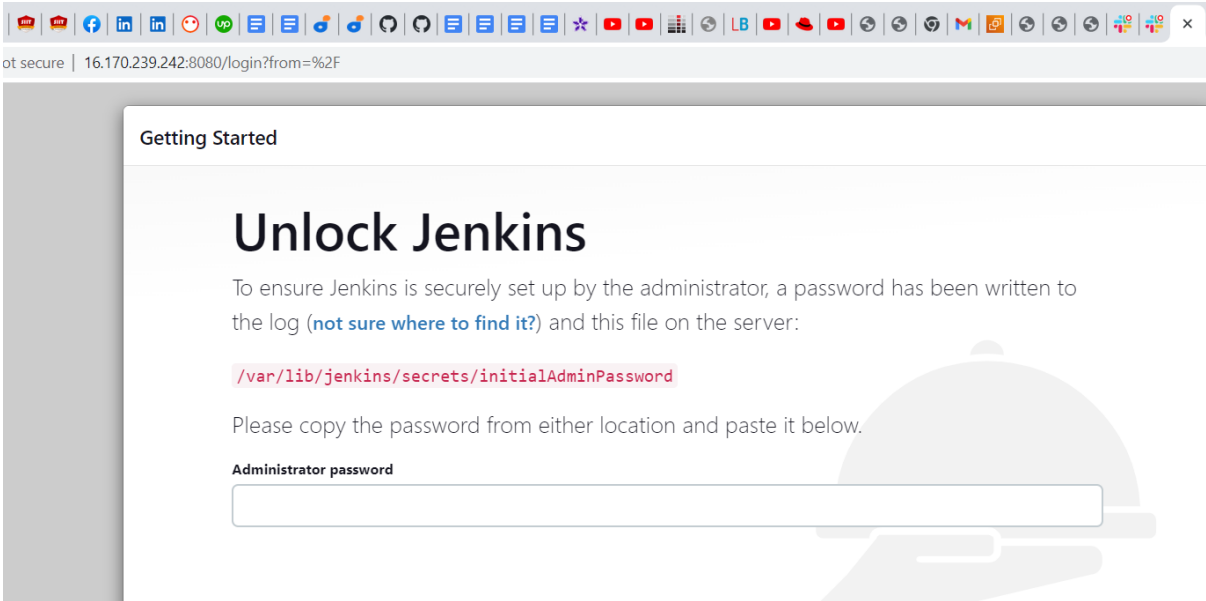
- By default Jenkins server uses TCP port 8080 – open it by creating a new Inbound Rule in your EC2 Security Group

Inbound rules (2)							
<input type="text" value="Filter security group rules"/> < 1 >							
<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	
<input type="checkbox"/>	-	sgr-0e606f7954ab7d904	IPv4	SSH	TCP	22	
<input type="checkbox"/>	-	sgr-0ea74efa301d58e1f	IPv4	Custom TCP	TCP	8080	

- Initial Jenkins setup.

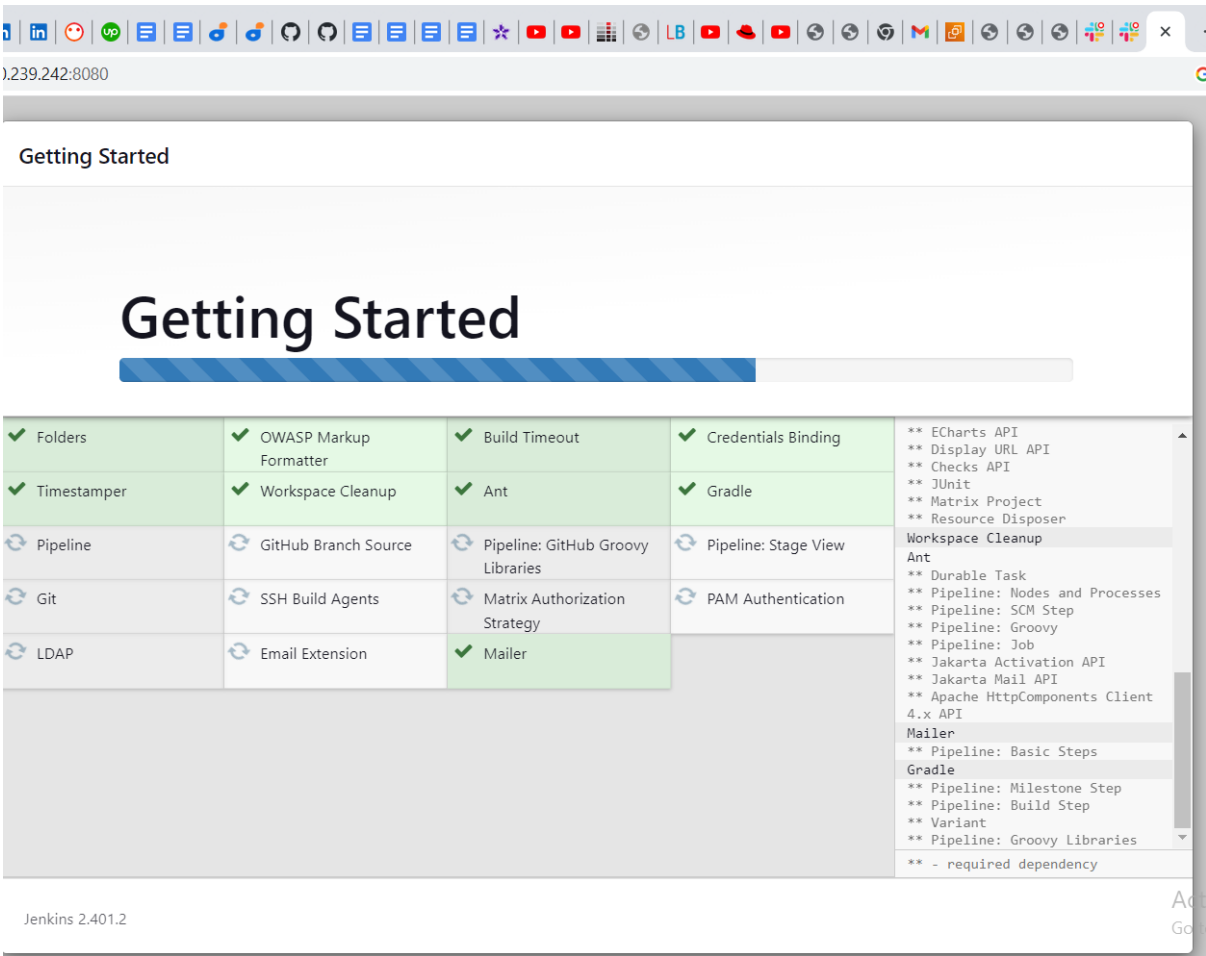
From your browser access

<http://<Jenkins-Server-Public-IP-Address-or-Public-DNS-Name>:8080>

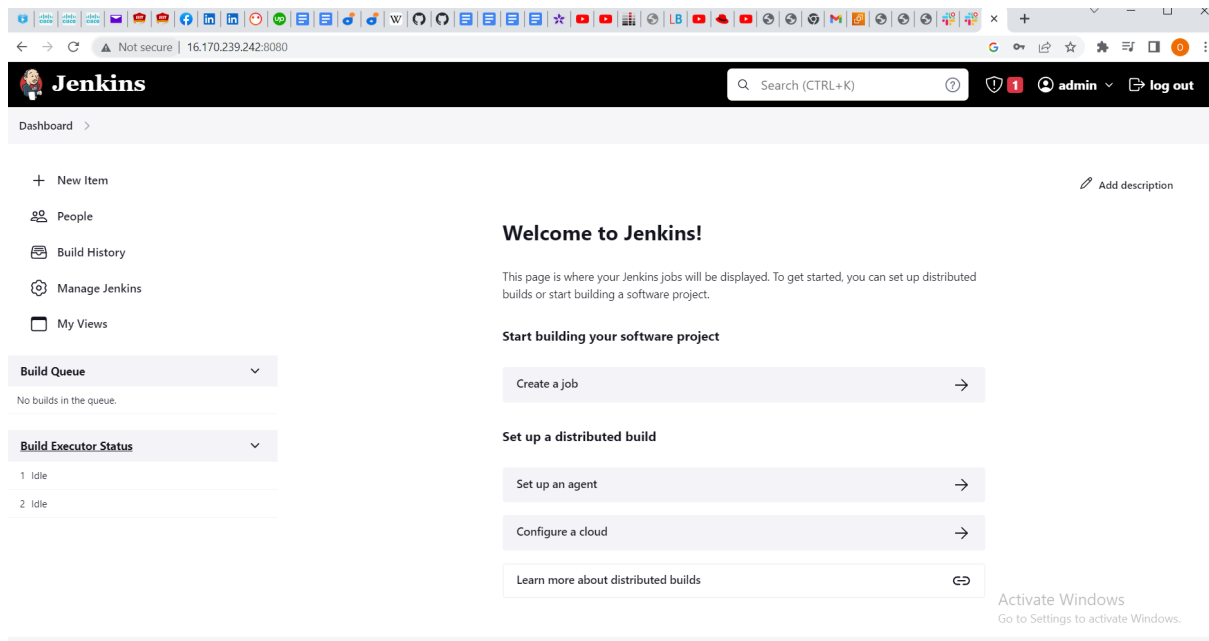


Retrieve it from your server: `sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

```
ubuntu@ip-172-31-38-178:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
5469278a32464534892d30c853336790
ubuntu@ip-172-31-38-178:~$
```



Click on get started;



- Configure Jenkins to retrieve source codes from GitHub using Webhooks

NB: We will configure a simple Jenkins job/project (these two terms can be used interchangeably). This job will be triggered by GitHub [webhooks](#) and will execute a 'build' task to retrieve codes from GitHub and store it locally on Jenkins server.

1. Enable webhooks in your GitHub repository settings

To create webhook, go to the settings tab on the github repo and click on webhooks. Webhook should look like this

<public_ip_of_jenkins_server>:8080/github-webhook/

github.com/ovaga/Project-9/settings/hooks/new

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules Beta

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub App

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

http://16.170.239.242:8080/Github-webhook/

Content type

application/json

Secret

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

Go to Jenkins web console, click “New Item” and create a “Freestyle project”

Enter an item name

project_9

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

branch Pipeline

Act
Go to

To connect your GitHub repository, you will need to provide its URL, you can copy from the repository itself. Provide the credentials (user/password) so Jenkins could access files in the repository.

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

https://github.com/ovaga/project_9.git

Credentials ?

Ovaga/******

Add ▾

Advanced ▾

Add Repository

Save

Apply

Activate Window
Go to Settings to activate

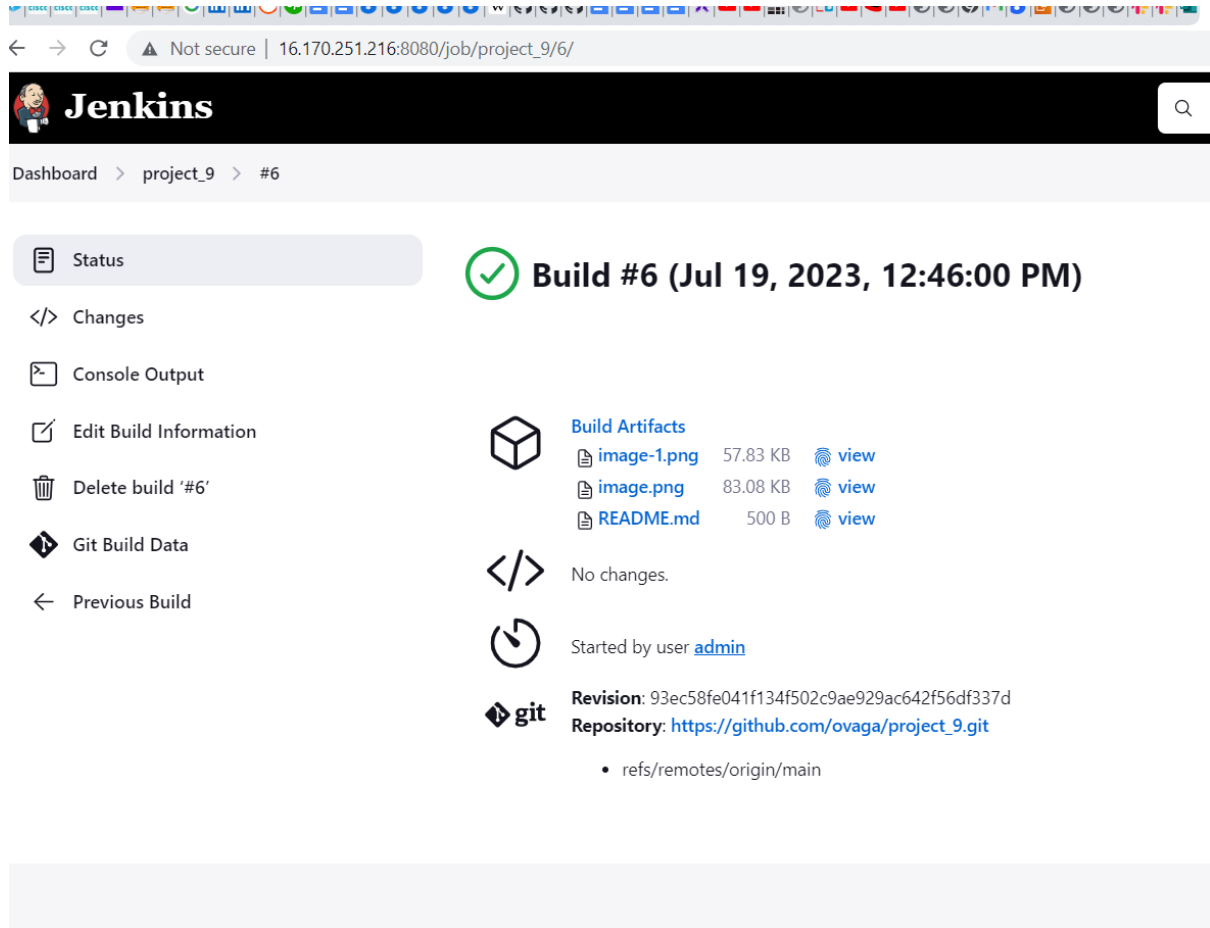
Save the configuration and try to run the build. For now we can only do it manually.

The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and the user 'admin'. The breadcrumb trail is 'Dashboard > project_9 > #3 > Console Output'. On the left sidebar, the 'Console Output' tab is selected. The main area displays the console output for build #3, which is a successful build. The output text is as follows:

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/project_9
The recommended git tool is: NONE
using credential e73634da-465c-4961-8182-86a8f1aacb33
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/project_9/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/ovaga/project_9.git # timeout=10
Fetching upstream changes from https://github.com/ovaga/project_9.git
> git --version # timeout=10
> git --version # 'git version 2.25.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/ovaga/project_9.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 401fcb2d46420942d0f91fc0eb385702eda5814 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 401fcb2d46420942d0f91fc0eb385702eda5814 # timeout=10
Commit message: "Initial commit"
First time build. Skipping changelog.
Finished: SUCCESS
```

At the bottom right of the console output area, there is a small 'Activate Window' watermark.

NB: The build is successful as shown in the output. However, it runs only when we trigger it manually. Let us fix it



The screenshot shows the Jenkins web interface in a browser. The address bar indicates the URL is `16.170.251.216:8080/job/project_9/6/`. The Jenkins logo and name are at the top. Below the navigation bar, the breadcrumb path is `Dashboard > project_9 > #6`. On the left sidebar, the 'Status' tab is selected. The main content area shows a green checkmark icon and the text **Build #6 (Jul 19, 2023, 12:46:00 PM)**. Below this, there are links for 'Changes', 'Console Output', 'Edit Build Information', 'Delete build '#6'', 'Git Build Data', and 'Previous Build'. To the right, under 'Build Artifacts', there is a table listing three files: `image-1.png` (57.83 KB), `image.png` (83.08 KB), and `README.md` (500 B), each with a 'view' link. Below the artifacts, it says 'No changes.' and 'Started by user admin'. At the bottom, the 'git' icon is shown with the 'Revision' `93ec58fe041f134f502c9ae929ac642f56df337d` and the 'Repository' `https://github.com/ovaga/project_9.git`, with a branch selection of `refs/remotes/origin/main`.

Configuring Jenkins To Copy Files(Artifact) to NFS Server

To achieve this, we install the `Publish Via SSH` plugin on Jenkins. The plugin allows one to send newly created packages to a remote server and install them, start and stop services that the build may depend on and many other use cases.

On main dashboard select "Manage Jenkins" and choose "Manage Plugins" menu item.

On "Available" tab search for "Publish Over SSH" plugin and install it

Plugins

Install	Name ↓	Released
<input checked="" type="checkbox"/>	Publish Over SSH 1.25 Artifact Uploaders Build Tools Send build artifacts over SSH	13 days ago

Install without restart

Download now and install after restart

Update information obtained: 1 hr 29 min ago

Check now

Configure the job to copy artifacts over to NFS server. On main dashboard select "Manage Jenkins" and choose "Configure System" menu item.

Scroll down to Publish over SSH plugin configuration section and configure it to be able to connect to the NFS server:

Provide a private key (content of .pem file that you use to connect to NFS server via SSH/Putty)

Hostname – can be private IP address of NFS server

Username – ec2-user (since NFS server is based on EC2 with RHEL 8)

Remote directory – /mnt/apps since our Web Servers use it as a mounting point to retrieve files from the NFS server

Test the configuration and make sure the connection returns Success. Remember, that TCP port 22 on NFS server must be open to receive SSH connections.

Dashboard > Manage Jenkins > System >

Publish over SSH

Jenkins SSH Key ?

Passphrase ?

The passphrase for the private key. Leave blank if the key is not encrypted. (from [Publish Over SSH](#))

Path to key ?

Key ?

```
H8oxuoD+jMdyC86pMthx2UwA1LXBZFEh90Zu3kNcVYfRnkzaMMemKnB89lc3Kb
PHHXPLUY7yCOeYVG147rcD/3/g3/QDzm4FRlgl3oHv/mSbk59D/yO1YLTl78ew
2EHPOwcCgYEAH4H/jK6v5bTZcsT1CjWw14bLVp7GnKF4gebNbEpHaYZ5R4dqs
nFV5NY1upcUn8DyotVZwWmLe7tpqbyCR8KMn4qjgh1gVNiWUngi8hT/8iq8dbuXg
oGt1PDYPgngMdhokmJOGbjcKqe+OkYElK26yTRFv13VwHRHqVAdVIQQ=
-----END RSA PRIVATE KEY-----
```

[Save](#) [Apply](#)

Skype
Vezeti Business in "ICNL NC
that is why they need a joint

> Manage Jenkins > System >

172.31.47.23

Username ?
ec2-user

Remote Directory ?
/mnt/apps

☐ Avoid sending files that have not changed ?

Advanced ▾

Success

Test Configuration

Add

Webhook will trigger a new job and in the “Console Output” of the job you will find something like this:

```
SSH: Transferred 25 file(s)
```

```
Finished: SUCCESS
```

Console Output

View as plain text

Edit Build Information

Delete build '#8'

Git Build Data

Previous Build

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/project_9
The recommended git tool is: NONE
using credential e73634da-465c-4961-8182-86a8f1aacb33
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/project_9/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/ovaga/project_9.git # timeout=10
Fetching upstream changes from https://github.com/ovaga/project_9.git
> git --version # timeout=10
> git --version # 'git version 2.25.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/ovaga/project_9.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 93ec58fe041f134f502c9ae929ac642f56df337d (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 93ec58fe041f134f502c9ae929ac642f56df337d # timeout=10
Commit message: "Project9 first commit"
> git rev-list --no-walk 93ec58fe041f134f502c9ae929ac642f56df337d # timeout=10
Archiving artifacts
SSH: Connecting from host [ip-172-31-38-178]
SSH: Connecting with configuration [NFS] ...
SSH: Disconnecting configuration [NFS] ...
SSH: Transferred 3 file(s)
Finished: SUCCESS
```

o make sure that the files in

`/mnt/apps`

have been updated – connect via SSH/Putty to your NFS server and check README.MD file

```
cat /mnt/apps/README.md
```

If you see the changes you had previously made in your GitHub – the job works as expected

```
[ec2-user@ip-172-31-47-23 ~]$ cat /mnt/apps/README.md
PROJECT 8 Latest
LOAD BALANCER SOLUTION WITH APACHE

![[Alt text](image.png)]

The concept in this project is to introduce a Load Balancer to distribute website traffic between the web servers and allow users to access the web site using a single URL.

Configure Apache As A Load Balancer
![[Alt text](image-1.png)]
Open TCP port 80 on Project-8-apache-lb by creating an Inbound Rule in Security Group.

[ec2-user@ip-172-31-47-23 ~]$
```

← → ↻ Not secure | 16.170.239.242:8080/job/project_9/configure

Dashboard > project_9 > 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) ?
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published build scans
- ☐ Terminate a build if it's stuck
- ☐ With Ant ?

Save **Apply**

- Configure “Post-build Actions” to archive all the files – files resulted from a build are called “artifacts”

← → ↻ Not secure | 16.170.239.242:8080/job/project_9/4/console

Jenkins Search (CTRL+K) admin

Dashboard > project_9 > #4 > Console Output

- Status
- Changes
- Console Output**
 - View as plain text
- Edit Build Information
- Delete build '#4'
- Polling Log
- Git Build Data
- Previous Build

Console Output

```
Started by GitHub push by ovaga
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/project_9
The recommended git tool is: NONE
using credential e73634da-465c-4961-8182-86a8f1aacb33
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/project_9/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/ovaga/project_9.git # timeout=10
Fetching upstream changes from https://github.com/ovaga/project_9.git
> git --version # timeout=10
> git --version # 'git version 2.25.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/ovaga/project_9.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 552fd410b090b91bee72782d2608fffd0349b9 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 552fd410b090b91bee72782d2608fffd0349b9 # timeout=10
Commit message: "Update README.md"
> git rev-list --no-walk 401fcbe2d46420942d0f91fc0eb385702eda5814 # timeout=10
Archiving artifacts
Finished: SUCCESS
```

Activate Window
Go to Settings to activate

