

1. Install Jenkins and run Jenkins on port number 8081.

Go to ec2 and launch one instance and login into that instance.

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The top navigation bar shows 'EC2 > Instances > Launch an instance'. A blue header bar contains a message: 'It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices' with 'Take a walkthrough' and 'Do not show me this message again.' buttons. The main form has sections for 'Name and tags', 'Application and OS Images (Amazon Machine Image)', and a summary. In the 'Name and tags' section, 'Name' is set to 'Jenkins'. In the 'Application and OS Images (Amazon Machine Image)' section, a search bar is shown. The 'Summary' section on the right shows 'Number of instances' set to 1, 'Software Image (AMI)' as Amazon Linux 2023 AMI 2023.9.2..., 'Virtual server type (instance type)' as t3.micro, and 'Storage (volumes)' as 1 volume(s) - 8 GiB. At the bottom are 'Cancel' and 'Launch instance' buttons.

```
MUJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ ssh -i red.pem ec2-user@35.173.219.195
The authenticity of host '35.173.219.195 (35.173.219.195)' can't be established.
ED25519 key fingerprint is SHA256:k68aIeaPYXXQnAw1lk6YP9y1kLAGMBq87Y2ikZamiF8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '35.173.219.195' (ED25519) to the list of known hosts.

,
~\_
###_
Amazon Linux 2023
~~ \
#####
~~ \
###
~~ \
#/ \
https://aws.amazon.com/linux/amazon-linux-2023
~~ \
V~ \
-->
~~ \
/ \
~~ \
/ \
~~ \
/ \
/_m/ \
[ec2-user@ip-172-31-60-147 ~]$ sudo su -
[root@ip-172-31-60-147 ~]# |
```

Go to browser and search download jenins select redhat enterprises linux.

4. You may also want to verify the package you downloaded. [Learn more about verifying Jenkins downloads.](#)

Download Jenkins 2.528.1 LTS for:

- Generic Java package (.war)
SHA-256: d630dca265f75a8d581f127a9234f1679d4b0800a8f370d03ad4a154ceb7295b
- Docker
- Kubernetes
- Ubuntu/Debian
- Red Hat Enterprise Linux and derivatives**
- Fedora
- Windows
- openSUSE
- FreeBSD
- Gentoo

Download Jenkins 2.532 for:

- Generic Java package (.war)
SHA-256: 50e9b91184005736a6399dfdd26fece06ff26f9a720c
- Docker
- Ubuntu/Debian
- Red Hat Enterprise Linux and derivatives
- Fedora
- Windows
- openSUSE
- Arch Linux
- FreeBSD
- Gentoo

Download java

- sudo dnf install -y java-17-amazon-corretto-devel

Add Jenkins repository and packages.

- 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.key>

install Jenkins.

sudo install -y Jenkins

```
[root@ip-172-31-60-147 ~]# sudo dnf install -y jenkins
Last metadata expiration check: 0:22:11 ago on Thu Oct 16 09:28:04 2025.
Package jenkins-2.528.1-1.1.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

- systemctl start Jenkins.
- Systemctl status Jenkins

```
[root@ip-172-31-60-147 ~]# sudo systemctl daemon-reload
[root@ip-172-31-60-147 ~]# sudo systemctl enable --now jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[root@ip-172-31-60-147 ~]# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: disabled)
   Active: active (running) since Thu 2025-10-16 09:50:59 UTC; 29s ago
     Main PID: 26976 (java)
        Tasks: 43 (limit: 1053)
       Memory: 366.6M
          CPU: 20.175s
        CGroup: /system.slice/jenkins.service
                  └─26976 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war

Oct 16 09:50:54 ip-172-31-60-147.ec2.internal jenkins[26976]: [LF]>
Oct 16 09:50:54 ip-172-31-60-147.ec2.internal jenkins[26976]: [LF]> ****
Oct 16 09:50:54 ip-172-31-60-147.ec2.internal jenkins[26976]: [LF]> ****
Oct 16 09:50:54 ip-172-31-60-147.ec2.internal jenkins[26976]: [LF]> ****
Oct 16 09:50:59 ip-172-31-60-147.ec2.internal jenkins[26976]: 2025-10-16 09:50:59.258+0000 [id=31]      INFO      jenkins
Oct 16 09:50:59 ip-172-31-60-147.ec2.internal jenkins[26976]: 2025-10-16 09:50:59.274+0000 [id=23]      INFO      hudson
Oct 16 09:50:59 ip-172-31-60-147.ec2.internal systemd[1]: started jenkins.service - Jenkins Continuous Integration Server.
Oct 16 09:50:59 ip-172-31-60-147.ec2.internal jenkins[26976]: 2025-10-16 09:50:59.355+0000 [id=49]      INFO      h.m.r
Oct 16 09:50:59 ip-172-31-60-147.ec2.internal jenkins[26976]: 2025-10-16 09:50:59.357+0000 [id=49]      INFO      hudson
Oct 16 09:51:04 ip-172-31-60-147.ec2.internal jenkins[26976]: 2025-10-16 09:51:04.333+0000 [id=68]      WARNING    hudson
```

Stop the Jenkins by using systemctl stop Jenkins

Execute this command to change port number: sudo vi
`/usr/lib/systemd/system/jenkins.service`

There is 8080 port in the environment change to 8081.

- Systemctl daemon-reload
- Systemctl start Jenkins
- Go to browser and search with the public ip and port number 8081.

```

# Directory where Jenkins stores its configuration and workspaces
Environment="JENKINS_HOME=/var/lib/jenkins"
WorkingDirectory=/var/lib/jenkins

# Location of the Jenkins WAR
#Environment="JENKINS_WAR=/usr/share/java/jenkins.war"

# Location of the exploded WAR
Environment="JENKINS_WEBROOT=%C/jenkins/war"

# Location of the Jenkins log. By default, systemd-journald(8) is used.
#Environment="JENKINS_LOG=%L/jenkins/jenkins.log"

# The Java home directory. When left empty, JENKINS_JAVA_CMD and PATH are consulted.
#Environment="JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64"

# The Java executable. When left empty, JAVA_HOME and PATH are consulted.
#Environment="JENKINS_JAVA_CMD=/etc/alternatives/java"

# Arguments for the Jenkins JVM
Environment="JAVA_OPTS=-Djava.awt.headless=true"

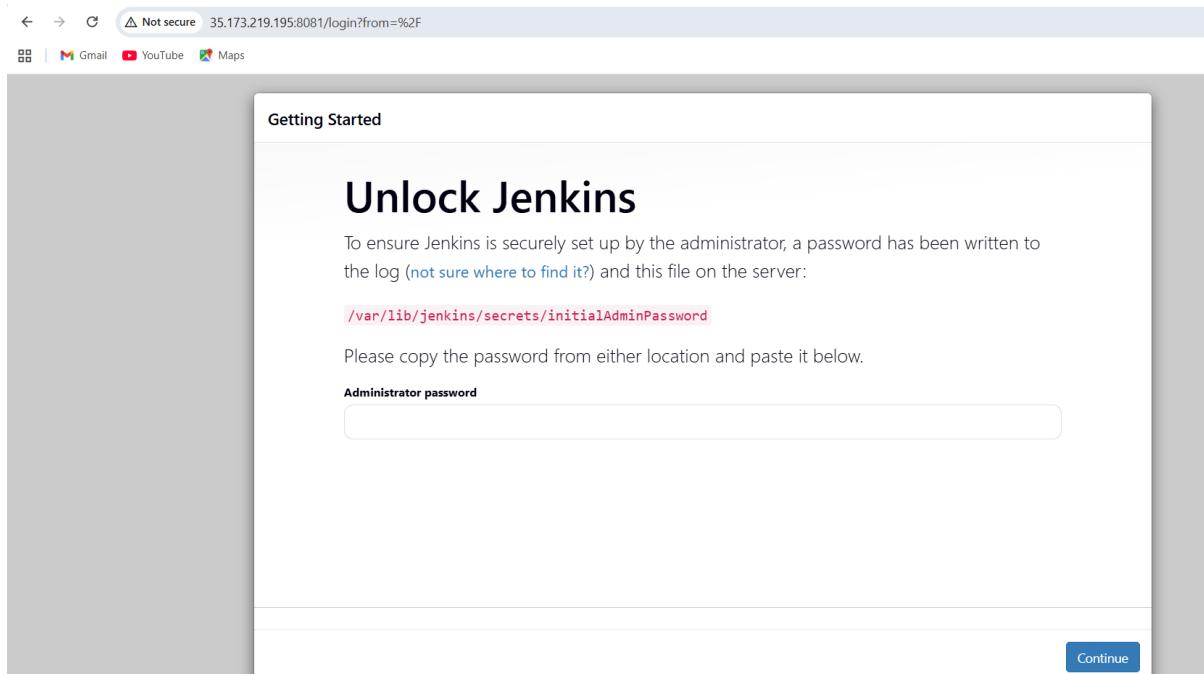
# Unix Domain Socket to listen on for local HTTP requests. Default is disabled.
#Environment="JENKINS_UNIX_DOMAIN_PATH=/run/jenkins/jenkins.socket"

# IP address to listen on for HTTP requests.
# The default is to listen on all interfaces (0.0.0.0).
#Environment="JENKINS_LISTEN_ADDRESS="

# Port to listen on for HTTP requests. Set to -1 to disable.
# To be able to listen on privileged ports (port numbers less than 1024),
# add the CAP_NET_BIND_SERVICE capability to the AmbientCapabilities
# directive below.
Environment="JENKINS_PORT=8081" ↓

# IP address to listen on for HTTPS requests. Default is disabled.
#Environment="JENKINS_HTTPS_LISTEN_ADDRESS="

```



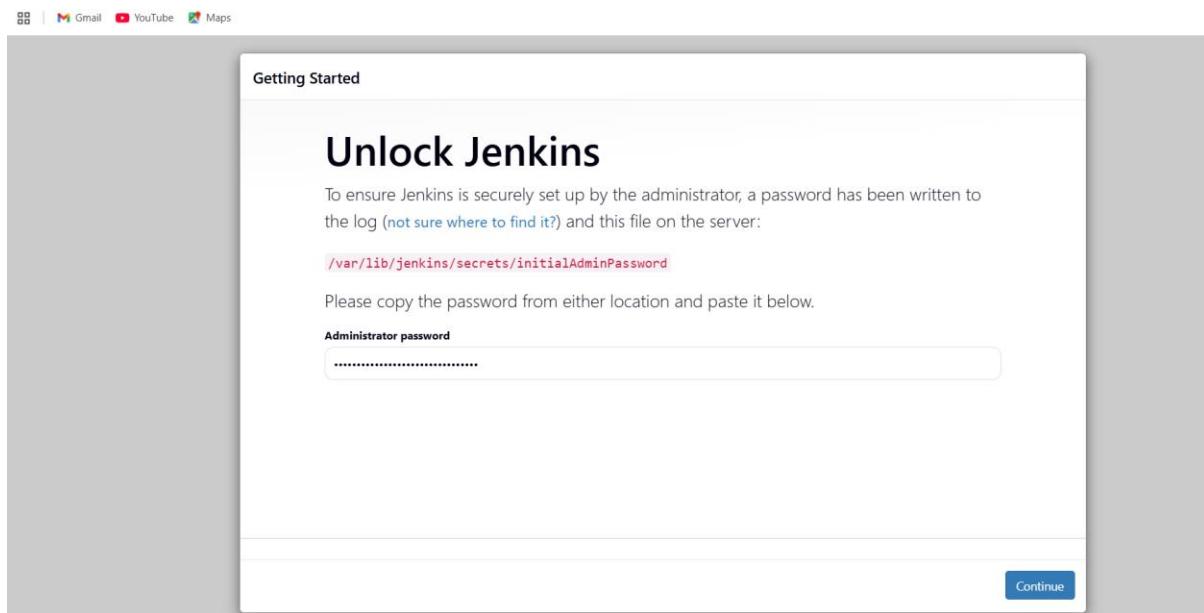
2. Secure Jenkins server

Go to ssh and cat

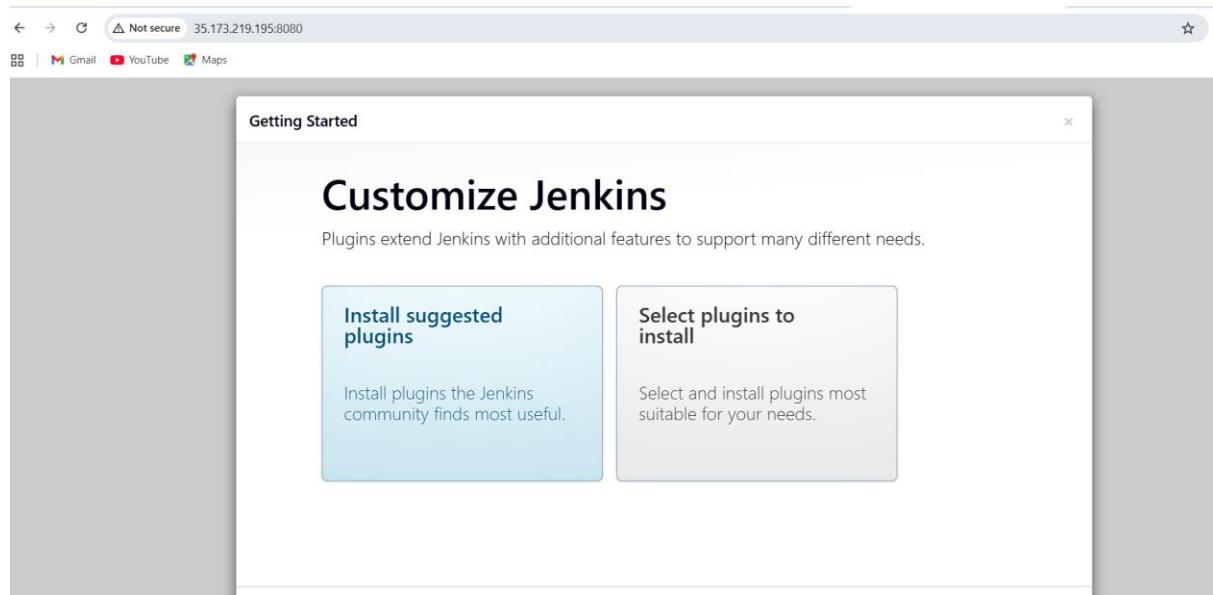
/var/lib/Jenkins/secrets/initialAdminPassword

Copy the password and paste in the website.

```
[root@ip-172-31-60-147 jenkins]# cat /var/lib/jenkins/secrets/initialAdminPassword  
ad517e0a42dc4ff5a2c81052f9dedeb9  
[root@ip-172-31-60-147 jenkins]# |
```



Click on install suggested plugins.



Give you details and create own password.

Getting Started

Username
mujaheed

Password
.....

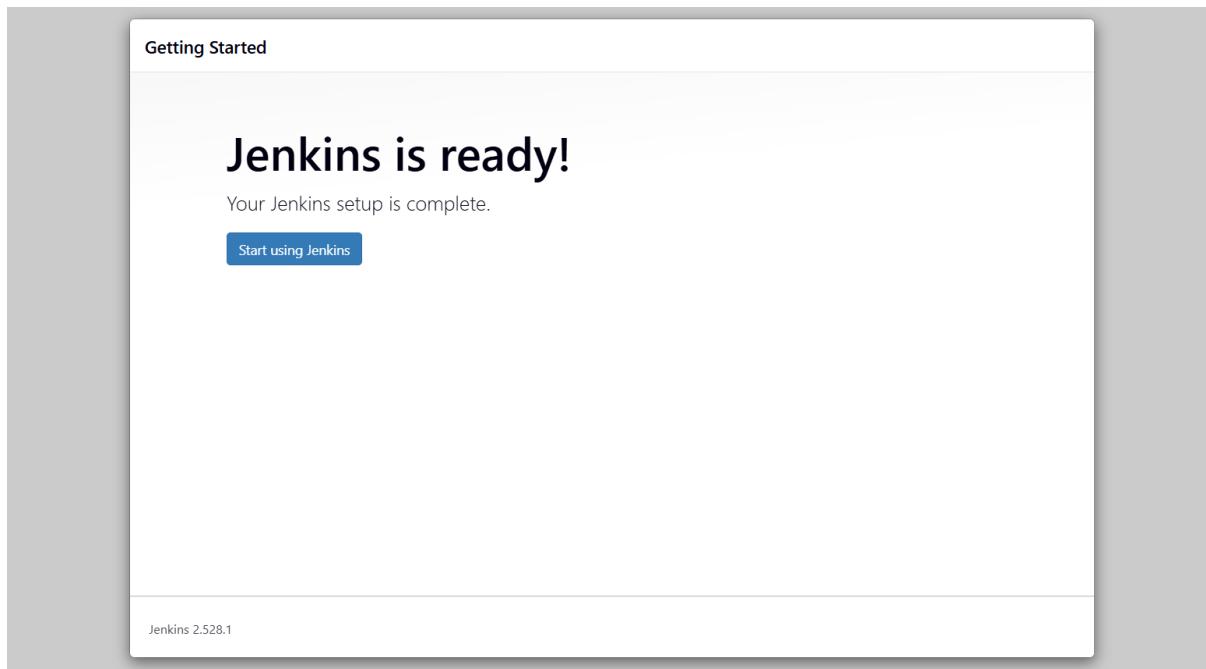
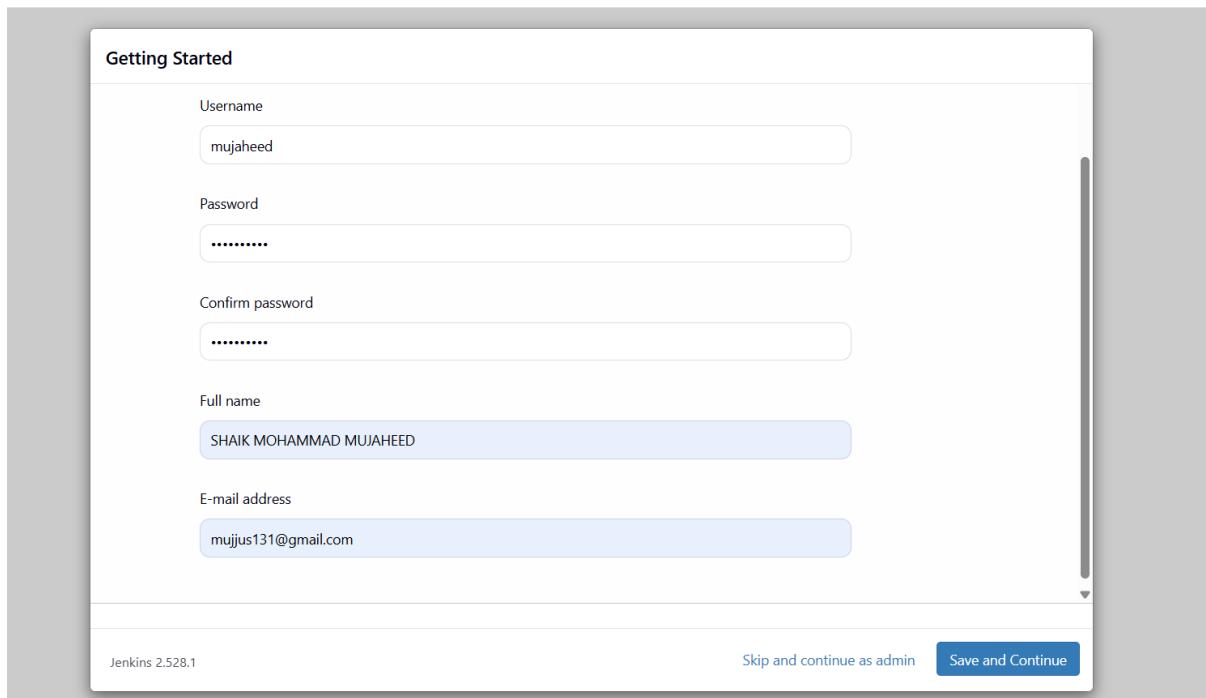
Confirm password
.....

Full name
SHAIK MOHAMMAD MUJAHEED

E-mail address
mujus131@gmail.com

Jenkins 2.528.1

Skip and continue as admin [Save and Continue](#)



3. Create users called DevOps, Testing in Jenkins with Limited access.

Go to the Jenkins account and select managed Jenkins.

Jenkins / Manage Jenkins

Nodes Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Clouds Add, remove, and configure cloud instances to provision agents on-demand.

functionality of Jenkins.

Appearance Configure the look and feel of Jenkins

Security

Security Secure Jenkins; define who is allowed to access/use the system.

Credentials Configure credentials

Credential Providers Configure the credential providers and types

Users Create/delete/modify users that can log in to this Jenkins.

Status Information

System Information Displays various environmental information to assist trouble-shooting.

System Log System log captures output from `java.util.logging` output related to Jenkins.

Load Statistics Check your resource utilization and see if you need more computers for your builds.

About Jenkins See the version and license information.

Click on create user

Jenkins / Manage Jenkins / Jenkins' own user database

Users 1

These users can log into Jenkins. This is a sub set of [this list](#), which also contains auto-created users who really just made some commits on some projects and have no direct Jenkins access.

User ID ↓	Name	⋮
mujahed	SHAIK MOHAMMAD MUJAHEED	⚙️

give the name called DevOps and give the password, email and create user.

← → ⚡ Not secure 35.173.219.195:8081/securityRealm/addUser

Gmail YouTube Maps

Jenkins / Jenkins' own user database / Create User

Create User

Username

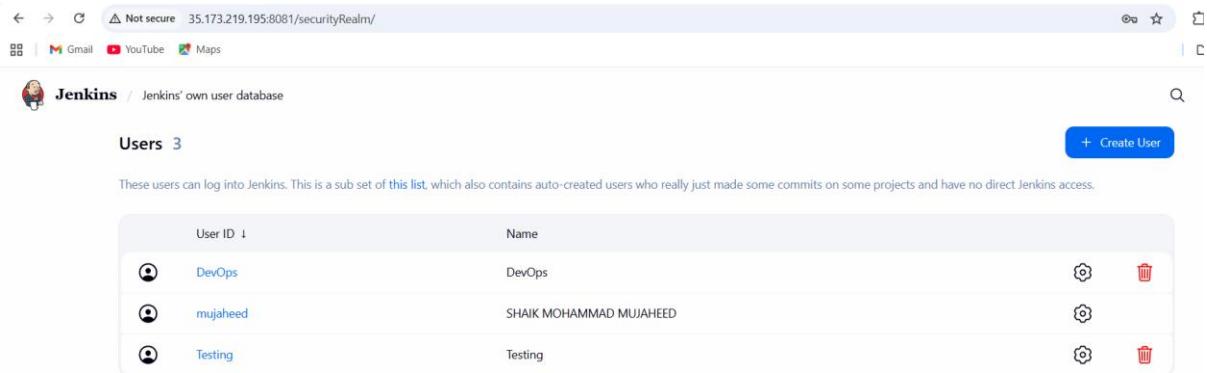
Password

Confirm password

Full name

E-mail address

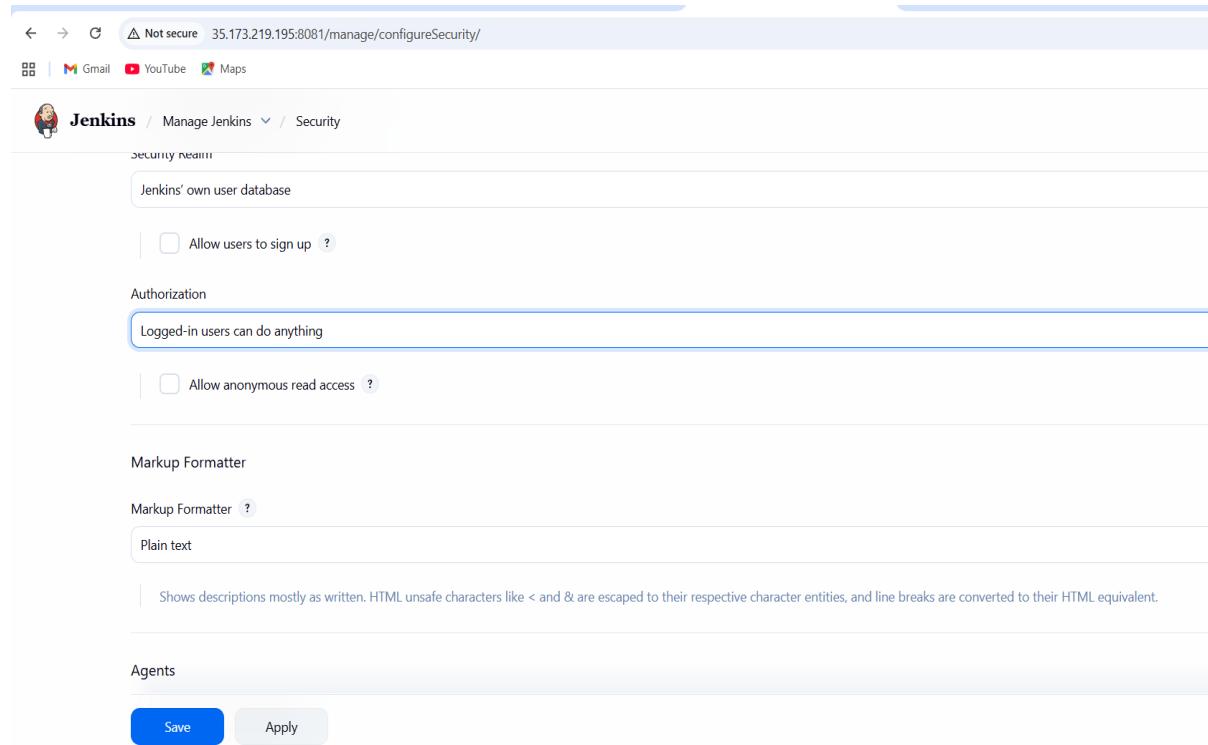
Create another user named as Testing and provide password, email and click on create user.



The screenshot shows the Jenkins user management interface. At the top, there's a header bar with a back arrow, forward arrow, refresh button, and a URL bar indicating 'Not secure' and the address '35.173.219.195:8081/securityRealm/'. Below the header is a navigation bar with links for 'Gmail', 'YouTube', and 'Maps'. The main content area has a title 'Jenkins / Jenkins' own user database' with a search icon. Underneath is a table titled 'Users 3'. The table has two columns: 'User ID ↓' and 'Name'. It lists three users: 'DevOps' (User ID 1), 'mujaheed' (User ID 2), and 'Testing' (User ID 3). Each user row includes a gear icon for settings and a trash can icon for deletion. A blue button at the bottom right of the table says '+ Create User'.

User ID ↓	Name		
1 DevOps	DevOps		
2 mujaheed	SHAIK MOHAMMAD MUJAHEED		
3 Testing	Testing		

Go to managed Jenkins and click on security and see the authorization in that select “project metrics-authorization strategy”.



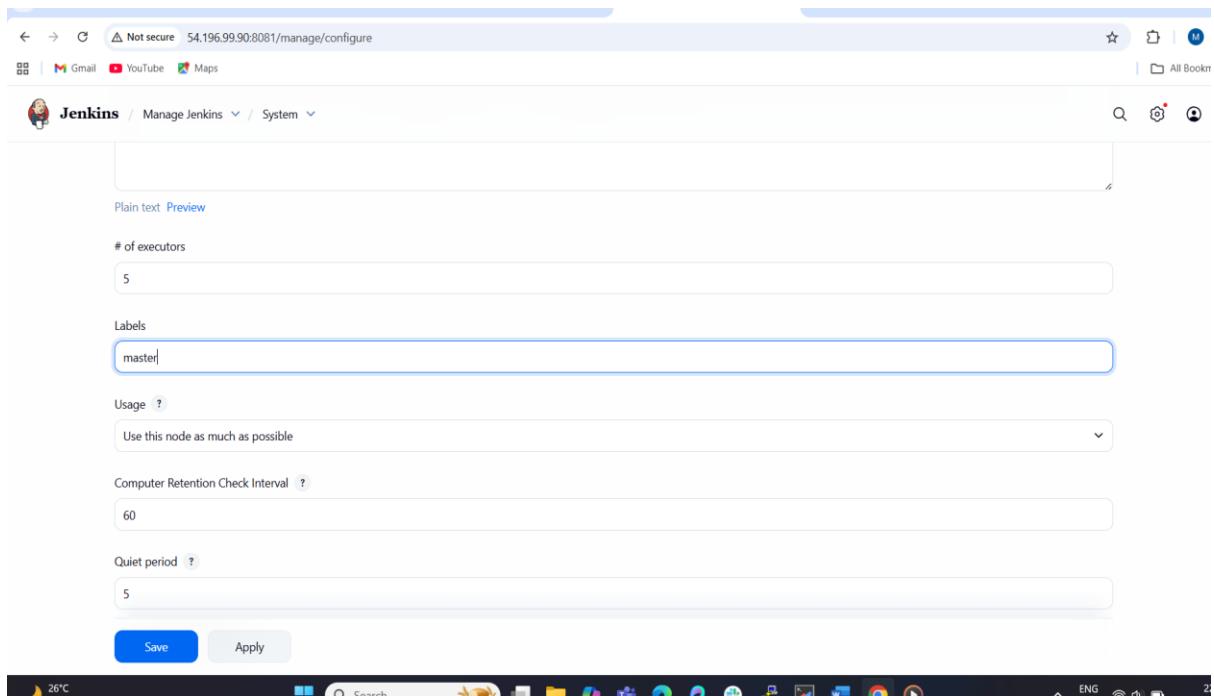
The screenshot shows the Jenkins security configuration page. At the top, there's a header bar with a back arrow, forward arrow, refresh button, and a URL bar indicating 'Not secure' and the address '35.173.219.195:8081/manage/configureSecurity/'. Below the header is a navigation bar with links for 'Gmail', 'YouTube', and 'Maps'. The main content area has a title 'Jenkins / Manage Jenkins / Security'. It includes sections for 'Security realm' (set to 'Jenkins' own user database'), 'Authorization' (set to 'Logged-in users can do anything'), 'Markup Formatter' (set to 'Plain text'), and 'Agents'. At the bottom are 'Save' and 'Apply' buttons.

Add the users and give limited access.

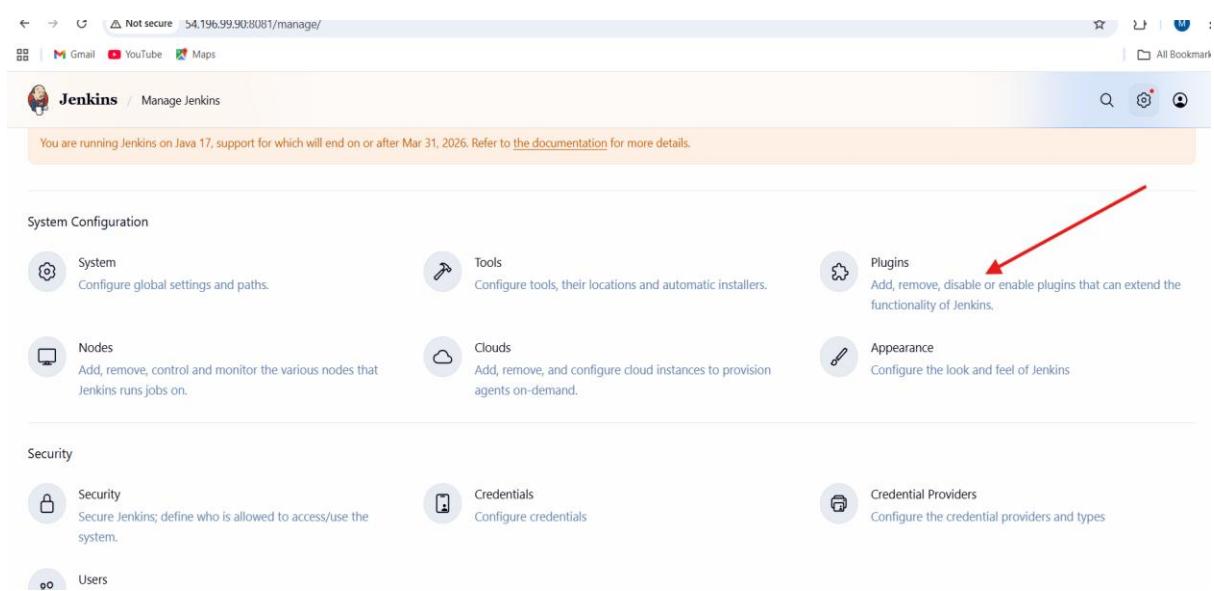
The screenshot shows the Jenkins security configuration interface. At the top, there's a navigation bar with links to Gmail, YouTube, and Maps, and a "All Bookmarks" link. Below that is the Jenkins logo and the "Manage Jenkins" and "Security" links. The main area is titled "Authorization" and shows the "Project-based Matrix Authorization Strategy". A legend indicates that checked boxes represent granted permissions and unchecked boxes represent denied permissions. The matrix includes columns for User/group, Overall, Credentials, Agent, Job, Run, View, SCM, Metrics, and various Jenkins-specific actions like ThreadDump, HealthCheck, and Tag. Rows include "Anonymous", "Authenticated Users", "DevOps", and "Testing". Buttons at the bottom allow adding users or groups and saving changes.

4. Configure labels and restrict the jobs to execute based on label only.

Go to managed Jenkins and go to systems and select label give the name as master.



Go to managed Jenkins and select plugins.



Go to available plugins an search for nodes and select node and label parameter.

A screenshot of a web browser displaying the Jenkins plugin manager at the URL <http://54.196.99.90:8081/manage/pluginManager/available>. The page title is "Not secure". The left sidebar shows navigation options: "Updates", "Available plugins" (which is selected), "Installed plugins", and "Advanced settings". A search bar at the top contains the query "node". Below the search bar, a list of plugins is displayed, each with a checkbox, a name, a brief description, a timestamp, and a green circular badge indicating the number of installations. A red arrow points to the "Node and Label parameter" plugin, which has 100 installations.

Plugin	Description	Last Updated	Installations
Build Wrappers	This plugin allows for throttling the number of concurrent builds of a project running per node or globally.	2 mo 9 days ago	100
built-on-column	Shows the actual node that a job was built on.	8 mo 14 days ago	94
Node and Label parameter	This plugin allows the node for a job to be selected dynamically.	7 mo 18 days ago	96
Node Iterator API	This plugin provides support for iterating through all the Node instances that are in use by Jenkins, even those Node instances that are not traditionally attached to Jenkins. The API exposed by this plugin can be used by cloud provider plugins to identify unused provisioned resource.	7 mo 16 days ago	96
Build Blocker	This plugin blocks a build if one of the given jobs is running. The blocking behaviour can be configured.	1 yr 4 mo ago	86

Install that node

A screenshot of a web browser displaying the Jenkins plugin manager at the URL <http://54.196.99.90:8081/manage/pluginManager/updates/>. The left sidebar shows navigation options: "Updates", "Available plugins" (selected), "Installed plugins", and "Advanced settings". A section titled "Download progress" is expanded, showing the status of various updates. A red arrow points to the "Node and Label parameter" update, which is marked as "Success".

Update	Status
Preparation	<ul style="list-style-type: none">Checking internet connectivityChecking update center connectivitySuccess
Node and Label parameter	Success
Loading plugin extensions	Success

→ Go back to the top page
(you can start using the installed plugins right away)

→ Restart Jenkins when installation is complete and no jobs are running

Select any job and go to configure and give restrict where the project can be run and select label. save the configurations.

The screenshot shows the Jenkins configuration interface for a job named 'third_job'. On the left, a sidebar lists sections: General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The 'General' section is selected and expanded. Under 'General', there are several checkboxes: 'Enable project-based security', 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Throttle builds', 'Execute concurrent builds if necessary', and 'Restrict where this project can be run'. The last checkbox, 'Restrict where this project can be run', is checked. Below it is a 'Label Expression' field containing 'master'. A note below the field states: 'Label master matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.' At the bottom of the configuration panel are 'Save' and 'Apply' buttons.

5. Create Three sample jobs using the below URL. https://github.com/betawins/Techie_horizon_Login_a_pp.git

Click on create job

The screenshot shows the Jenkins dashboard at the URL <http://100.29.11.131:8081>. The top navigation bar includes links for 'New Item', 'Build History', 'Add description', and a search icon. The main content area features a 'Welcome to Jenkins!' message: '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.' Below this is a 'Start building your software project' section with a 'Create a job' button and a '+' icon. There are also three buttons for 'Set up a distributed build': 'Set up an agent' (with a monitor icon), 'Configure a cloud' (with a cloud icon), and 'Learn more about distributed builds' (with a help icon). On the left, there are two collapsed sections: 'Build Queue' (which is currently empty) and 'Build Executor Status' (which shows '(0 of 2 executors busy)').

Give the name and select freestyle project.

Not secure 100.29.11.131:8081/new/job

Jenkins / New Item

New Item

Enter an item name

Select an item type

- Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- 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

Give the description and select git option provide the git URL and save it.

Not secure 100.29.11.131:8081/job/first-item/configure

Jenkins / first-item / Configuration

Configure	General	Enabled <input checked="" type="checkbox"/>
<input checked="" type="radio"/> General	Description <input type="text" value="first job"/>	
<input type="radio"/> Source Code Management		
<input type="radio"/> Triggers	<input type="checkbox"/> Enable project-based security	
<input type="radio"/> Environment	<input type="checkbox"/> Discard old builds ?	
<input type="radio"/> Build Steps	<input type="checkbox"/> GitHub project	
<input type="radio"/> Post-build Actions	<input type="checkbox"/> This project is parameterized ?	
	<input type="checkbox"/> Throttle builds ?	
	<input type="checkbox"/> Execute concurrent builds if necessary ?	

The screenshot shows the Jenkins configuration interface for a job named 'first-item'. The left sidebar lists configuration sections: General, Source Code Management (selected), Triggers, Environment, Build Steps, and Post-build Actions. Under 'Source Code Management', 'Git' is selected, and the 'Repository URL' is set to 'https://github.com/betawins/Techie_horizon_Login.app.git'. The 'Branches to build' section shows 'Branch Specifier (blank for 'any')' set to '*/*master'. At the bottom are 'Save' and 'Apply' buttons.

Click on build now to run the job.

The screenshot shows the Jenkins dashboard. The 'Build History' section displays a single build entry for 'first-item', which was successful ('S') and triggered ('W'). The build took 17 seconds and is labeled '#1'. The 'Build Queue' section shows 'No builds in the queue.' The 'Build Executor Status' section indicates '(0 of 2 executors busy)'. At the bottom, there are icons for S, M, and L.

Do the same process for another two jobs for creation.

The screenshot shows the Jenkins dashboard at 100.29.11.131:8081. At the top, there's a navigation bar with links to Gmail, YouTube, and Maps. Below it, the Jenkins logo is visible. A search bar and a 'New Item' button are on the left. In the center, there's a 'Build History' section with a table showing three builds: 'first-item', 'second_job', and 'third_job'. The table includes columns for Name, Last Success, Last Failure, and Last Duration. On the left, there are dropdown menus for 'Build Queue' (showing 'No builds in the queue') and 'Build Executor Status' (showing '0 of 5 executors busy'). At the bottom, there's a footer with icons for S, M, and L.

6. Create one Jenkins job using git hub Private repository.

Go to git hub and select your repository and go to settings and click on create tokens and create classic token.

The screenshot shows the GitHub developer settings page under 'Personal access tokens'. The 'Tokens (classic)' tab is selected. A new token named 'mujaheed_token' is being created with an expiration date of '30 days (Nov 18, 2025)'. The 'Select scopes' section lists various permissions: 'repo' (checkbox checked), 'repo:status' (checkbox checked), 'repo_deployment' (checkbox checked), 'public_repo' (checkbox checked), 'repo:invite' (checkbox checked), 'security_events' (checkbox checked), and 'workflow' (checkbox checked). Descriptions for each scope are provided on the right.

Go to managed Jenkins and click on credentials and select system create credentials and save it.

System Configuration

- System**: Configure global settings and paths.
- Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Tools**: Configure tools, their locations and automatic installers.
- Clouds**: Add, remove, and configure cloud instances to provision agents on-demand.
- Plugins**: Add, remove, disable or enable plugins that can extend functionality of Jenkins.
- Credentials**: Configure credentials (red arrow points here)
- Appearance**: Configure the look and feel of Jenkins
- Credential Providers**: Configure the credential providers and types

Credentials

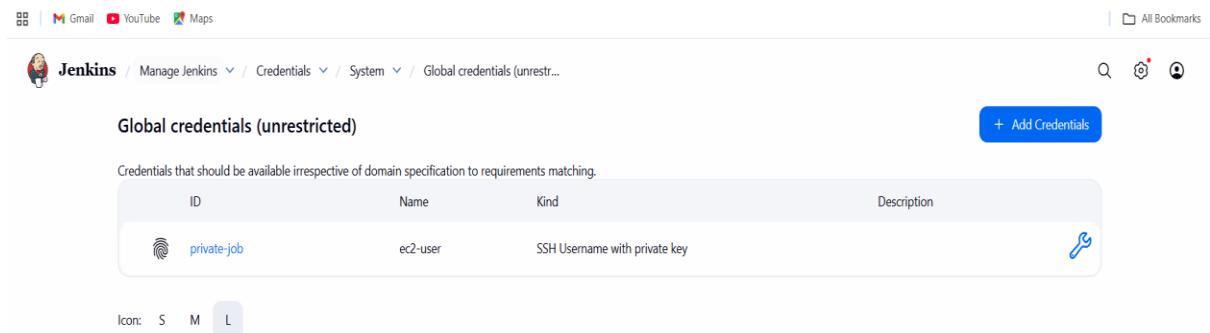
T	P	Store ↓	Domain	ID	Name
Stores scoped to Jenkins					
			Domains		
		System	(global)		
Icon:	S	M	L		

System

Domain ↓	Description
Global credentials (unrestricted)	Credentials that should be available irrespective of domain specification to requirements matching.

Select create with ssh.

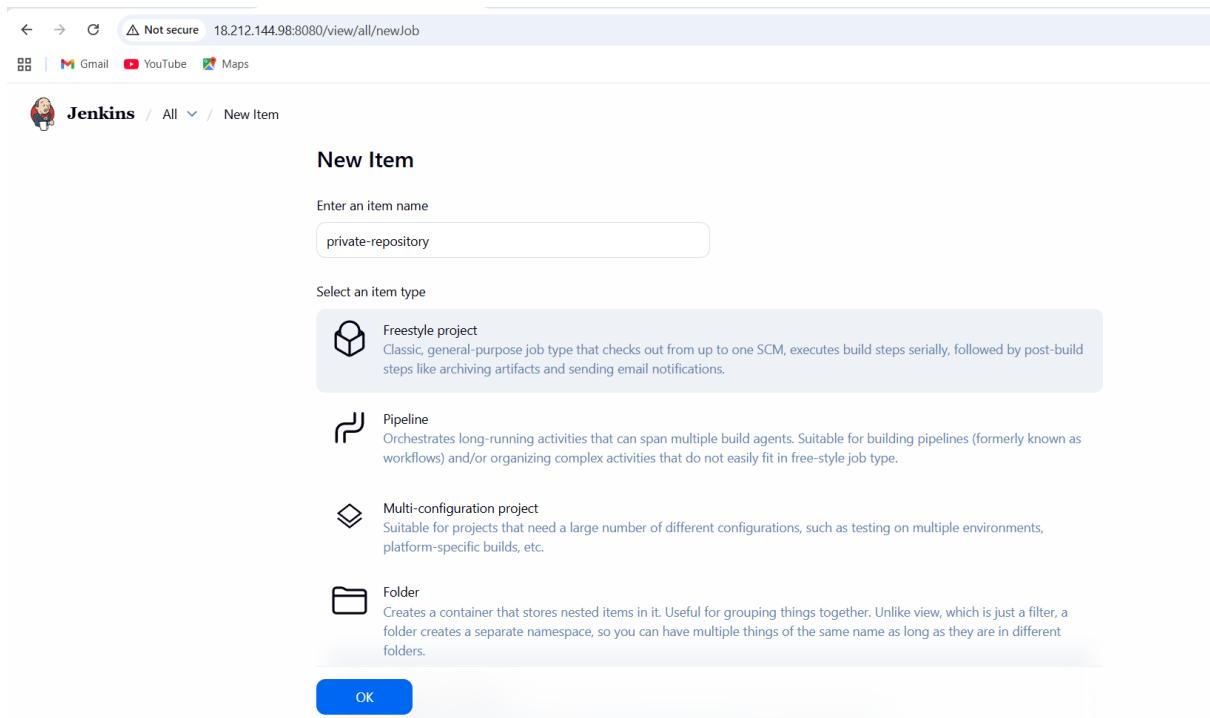
Give the id as private which is name. and user name as ec2-user and copy the sshkey and paste create it.



The screenshot shows the Jenkins Global credentials (unrestricted) page. It displays a table with one row. The columns are ID, Name, Kind, and Description. The ID is 'private-job', the Name is 'ec2-user', the Kind is 'SSH Username with private key', and the Description is empty. There is a blue 'Edit' icon next to the row. Below the table, there are icons for S, M, and L.

ID	Name	Kind	Description
private-job	ec2-user	SSH Username with private key	

Go to Jenkins and create a new job.



The screenshot shows the Jenkins New Item creation page. The item name is 'private-repository'. The item type is set to 'Freestyle project'. Other options shown include 'Pipeline', 'Multi-configuration project', and 'Folder'. A blue 'OK' button is at the bottom.

New Item

Enter an item name
private-repository

Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

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

The screenshot shows the Jenkins job configuration page for a job named "private-job". The "General" tab is selected. The "Description" field contains "private-job". Several checkboxes are available under the "Advanced" section, including "Discard old builds", "GitHub project", "This project is parameterized", "Throttle builds", and "Execute concurrent builds if necessary". Buttons for "Save" and "Apply" are at the bottom.

Give your private repository link and password.

The screenshot shows the Jenkins job configuration page for a job named "private-job". The "Source Code Management" tab is selected. It is configured to use "Git" with a repository URL of "https://github.com/mujaheed00/Techie-horizon-01.git". A dropdown menu for "Credentials" is set to "- none -". Buttons for "+ Add Repository" and "+ Add" are visible.

← → ⌛ Not secure 18.212.144.98:8080/job/private-repository/

Gmail YouTube Maps

 Jenkins / private-repository

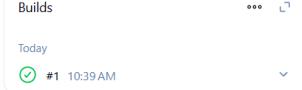
Status Changes Workspace Build Now Configure Delete Project Rename Credentials

private-repository

Permalinks

Builds

Today #1 10:39 AM



 Jenkins

+ New Item

Build History All +

Add description

Build Queue

Build Executor Status 0/2

S	W	Name ↓	Last Success	Last Failure	Last Duration
		first-item	1 hr 19 min #1	N/A	3.5 sec
		private-repository	1 min 49 sec #1	N/A	9 ms

Icon: S M L

