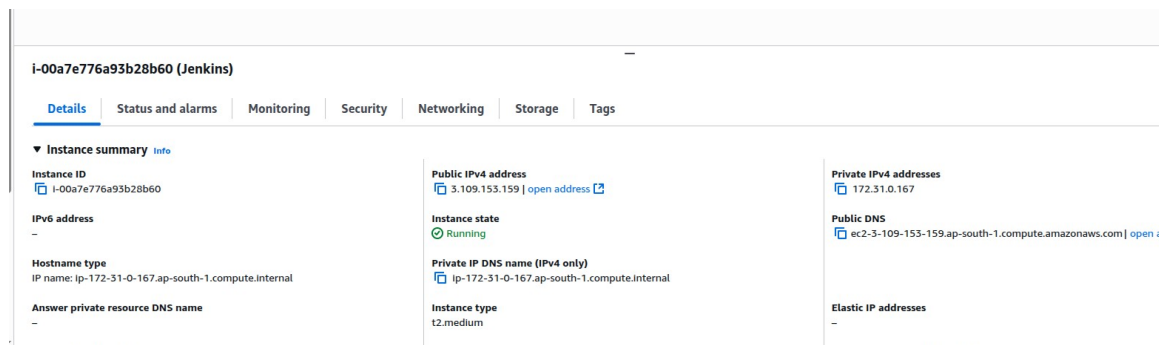


# Step-by-Step SOP: Launch Jenkins on AWS and Explore Project & User Creation

---

## 1. Launch EC2 Instance for Jenkins

- 1.1. Login to AWS Management Console.
- 1.2. Navigate to EC2 Dashboard -> Launch Instance.
- 1.3. Select Amazon Linux 2 or Ubuntu 22.04 LTS as the AMI.
- 1.4. Choose t2.medium instance type (recommended for Jenkins).
- 1.5. Create or choose an existing key pair.
- 1.6. Allow HTTP, HTTPS, and SSH in the security group.
- 1.7. Launch the instance.



## 2. Install Jenkins on EC2

- 2.1. Connect to your EC2 using SSH:  
`ssh -i your-key.pem ubuntu@<EC2-Public-IP>`
- 2.2. Update system:  
`sudo apt update && sudo apt upgrade -y`

### 2.3. Install Java (required by Jenkins):

```
sudo apt install openjdk-17-jdk -y
```

### 2.4. Add Jenkins repo and install Jenkins:

```
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
```

```
sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ >  
/etc/apt/sources.list.d/jenkins.list'
```

```
sudo apt update
```

```
sudo apt install jenkins -y
```

### 2.5. Start and enable Jenkins:

```
sudo systemctl start jenkins
```

```
sudo systemctl enable jenkins
```

```
ubuntu@ip-172-31-0-167:~$ 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 Sun 2025-06-22 18:47:43 UTC; 10s ago
     Main PID: 2849 (java)
        Tasks: 52 (limit: 4670)
       Memory: 870.7M
          CPU: 17.203s
      CGroup: /system.slice/jenkins.service
              └─2849 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --web

Jun 22 18:47:38 ip-172-31-0-167 jenkins[2849]: 2c3d559d4ae34d60a60dc003a44a66fe
Jun 22 18:47:38 ip-172-31-0-167 jenkins[2849]: This may also be found at: /var/lib/jenkins/secre
Jun 22 18:47:38 ip-172-31-0-167 jenkins[2849]: *****
Jun 22 18:47:38 ip-172-31-0-167 jenkins[2849]: *****
Jun 22 18:47:43 ip-172-31-0-167 jenkins[2849]: 2025-06-22 18:47:43.542+0000 [id=31]          INFO
Jun 22 18:47:43 ip-172-31-0-167 jenkins[2849]: 2025-06-22 18:47:43.556+0000 [id=23]          INFO
Jun 22 18:47:43 ip-172-31-0-167 systemd[1]: Started Jenkins Continuous Integration Server.
Jun 22 18:47:45 ip-172-31-0-167 jenkins[2849]: 2025-06-22 18:47:45.050+0000 [id=49]          INFO
Jun 22 18:47:45 ip-172-31-0-167 jenkins[2849]: 2025-06-22 18:47:45.051+0000 [id=49]          INFO
ubuntu@ip-172-31-0-167:~$
```

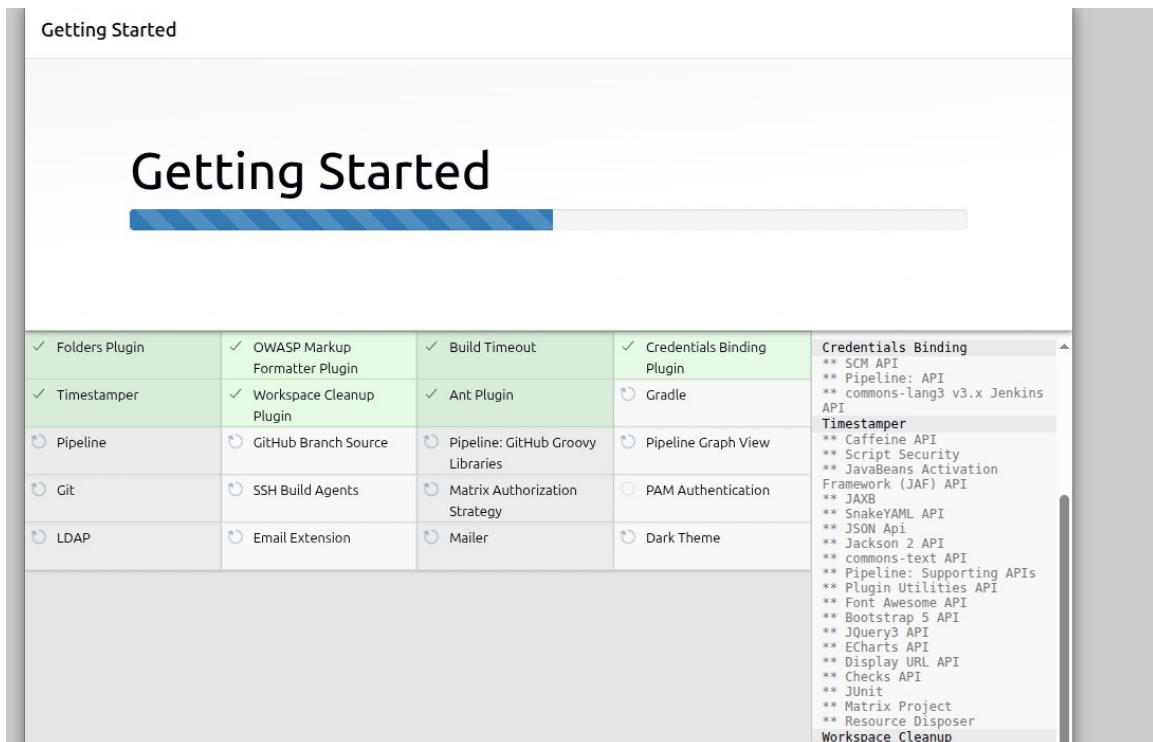
### 2.6. Allow port 8080 in security group (if not done earlier).

### 2.7. Open Jenkins in browser: <http://<EC2-Public-IP>:8080>



2.8. Retrieve initial admin password:

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



### 3. Setup Jenkins and Create Users

3.1. Paste the admin password in Jenkins UI and install suggested plugins.

3.2. Create admin user during setup.

3.3. Go to 'Manage Jenkins' -> 'Manage Users' -> 'Create User'.

3.4. Provide username, password, full name, and email.

Getting Started

## Create First Admin User

Username

Password

Confirm password

Full name

E-mail address

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.

## 4. Create a Jenkins Project

4.1. From Jenkins dashboard, click on 'New Item'.



4.2. Enter item name and select 'Freestyle project'. Click OK.

### New Item

Enter an item name

surya1

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.



#### Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.



#### Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.


OK

4.3. In the configuration page:

- Add a description.
- Under Source Code Management, connect to GitHub or other repo.
- Under Build Triggers, choose trigger (e.g., GitHub webhook).
- Under Build, add build steps (e.g., shell commands).

4.4. Save the job.

4.5. Click 'Build Now' to test.

 **Jenkins**

Dashboard > surya1 >

Status

</> Changes

Workspace

Build Now

Configure

Delete Project

GitHub

Rename

✓ surya1

**Permalinks**

- Last build (#1), 34 sec ago
- Last stable build (#1), 34 sec ago
- Last successful build (#1), 34 sec ago
- Last completed build (#1), 34 sec ago

Builds

Filter

Today

✓ #1 19:09

## 5. Post-Installation Notes

- Ensure your EC2 instance has sufficient disk space.
- Use IAM roles for better security with Git or AWS integrations.
- Setup backup for Jenkins home (/var/lib/jenkins).
- For production, consider placing Jenkins behind a reverse proxy with SSL.