Jenkins Learning Journey – Day 1

What I Learned Today

Today, I learned about Jenkins, set it up, created my first job, and even explored declarative pipelines and agent configuration!

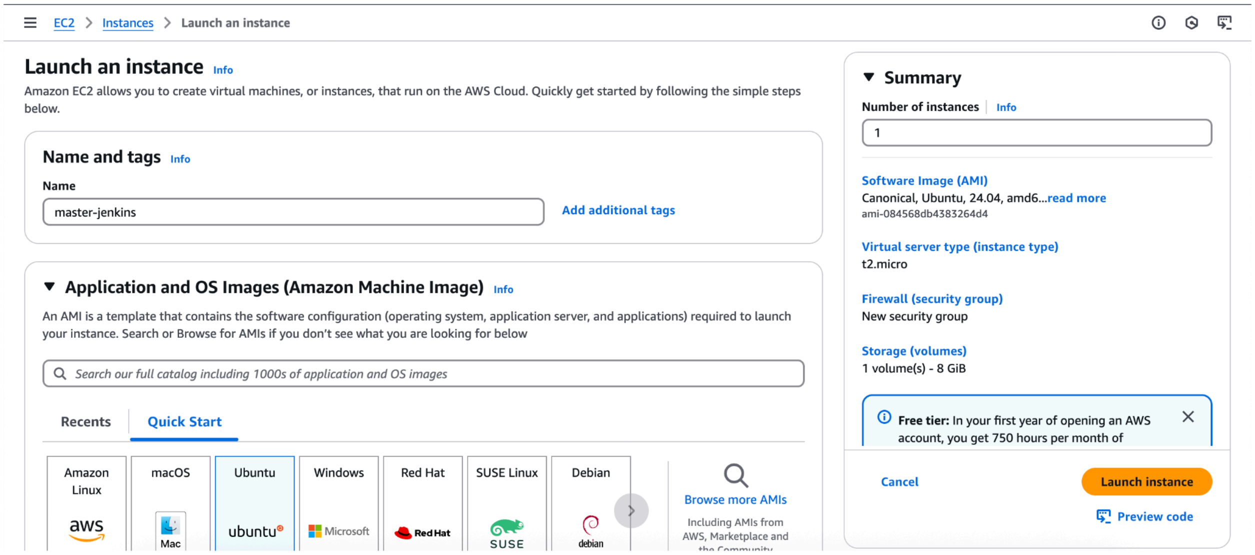
Setup Steps

1. What is Jenkins?

- Jenkins is an open-source automation server used mainly for building, testing, and deploying software automatically. It plays a key role in CI/CD pipelines.

2. Setting Up Jenkins on AWS EC2

- **Launch EC2 Instance:**
  - Launched an Amazon Linux 2 EC2 instance.



EC2 > Instances > Launch an instance

Launch an instance Info

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.

Name and tags Info

Name: master-jenkins

Add additional tags

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 | SUSE Linux | Debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Summary

Number of instances: 1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-084568db4383264d4

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 of opening an AWS account, you get 750 hours per month of

Install Java:

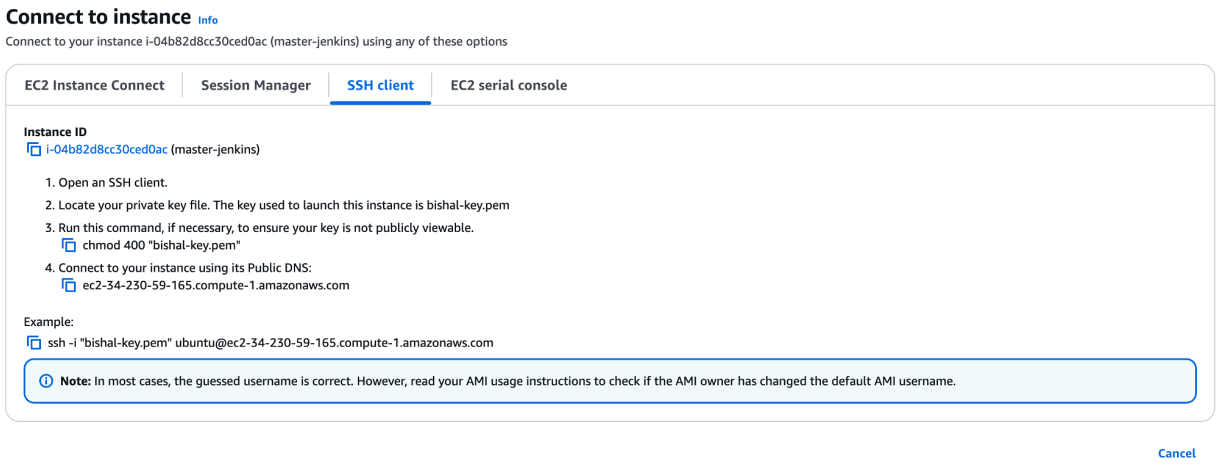
```
sudo apt update
sudo apt install fontconfig openjdk-21-jre
java -version
openjdk version "21.0.3" 2024-04-16
OpenJDK Runtime Environment (build 21.0.3+11-Debian-2)
OpenJDK 64-Bit Server VM (build 21.0.3+11-Debian-2, mixed mode,
sharing)
```

Install Jenkins:

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
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 jenkins
```

Configure Security Group:

Updated the EC2 security group to allow inbound TCP traffic on port **8080**.



Connect to instance

Connect to your instance i-04b82d8cc30ced0ac (master-jenkins) using any of these options

Instance ID

i-04b82d8cc30ced0ac (master-jenkins)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is bishal-key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "bishal-key.pem"
4. Connect to your instance using its Public DNS:
   ec2-34-230-59-165.compute-1.amazonaws.com

Example:

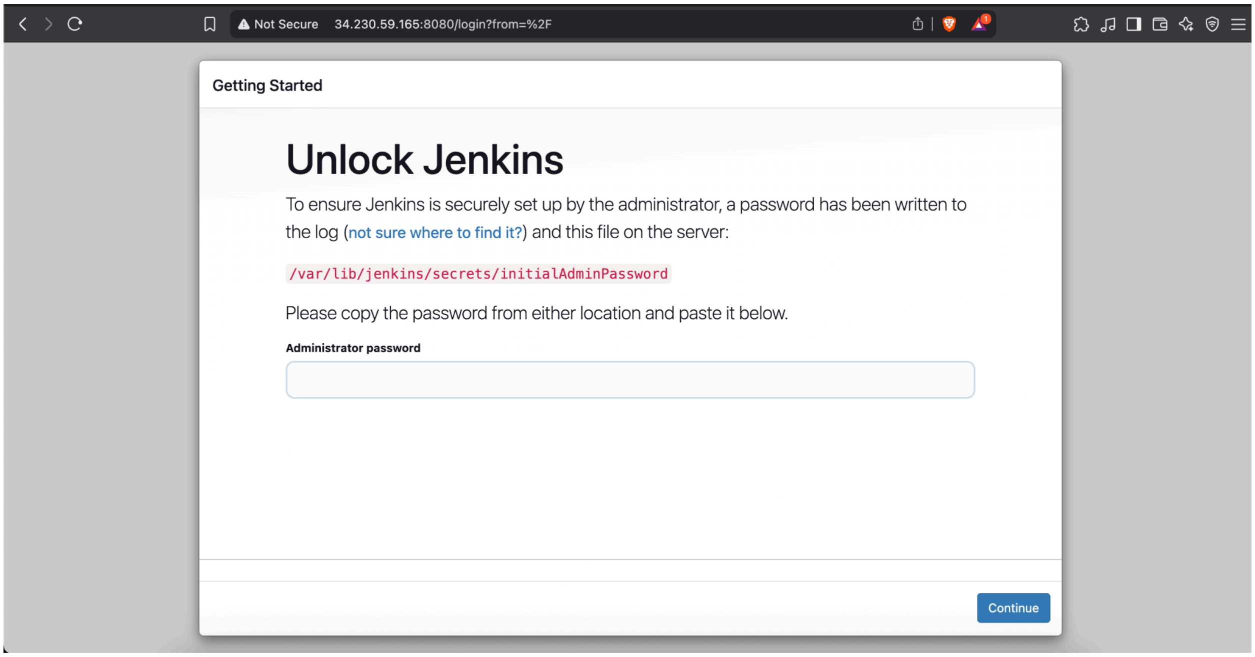
```
ssh -i "bishal-key.pem" ubuntu@ec2-34-230-59-165.compute-1.amazonaws.com
```

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

3. Access Jenkins

Accessed Jenkins in the browser using:

```
http://<EC2-Public-IP>:8080
```



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`

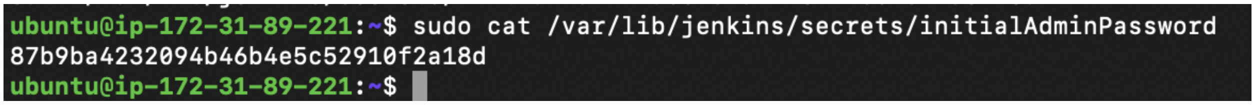
Please copy the password from either location and paste it below.

Administrator password

4. Unlock Jenkins

Retrieved the initial admin password:

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

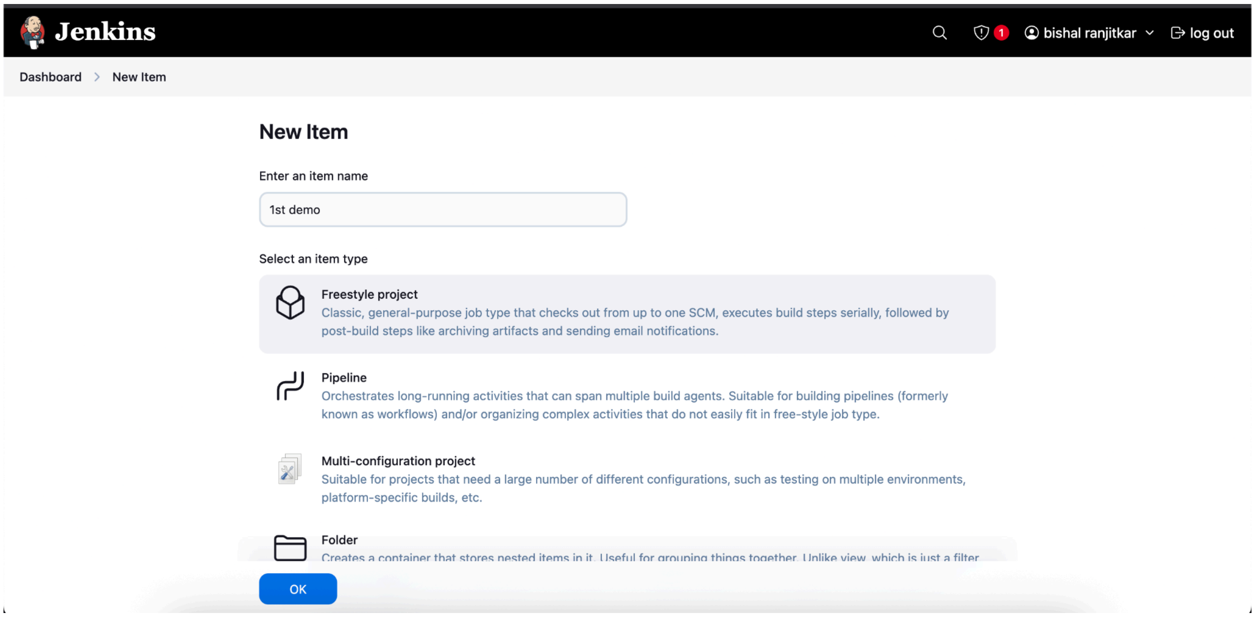


```
ubuntu@ip-172-31-89-221:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
87b9ba4232094b46b4e5c52910f2a18d
ubuntu@ip-172-31-89-221:~$
```

Completed Jenkins setup and installed the recommended plugins.

Task 1 – Create My First Job

- **Created a New Item** (`FirstJob`) → selected "Freestyle project."



New Item

Enter an item name

1st demo

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.

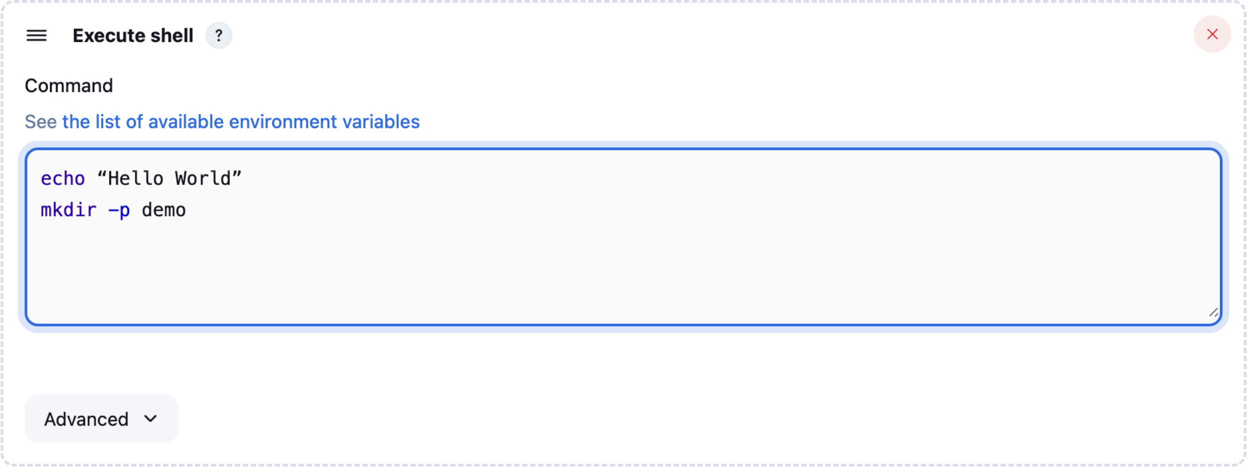
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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

Build Step: Chose "Execute Shell" and entered:

```
echo “Hello World”
mkdir -p demo
```



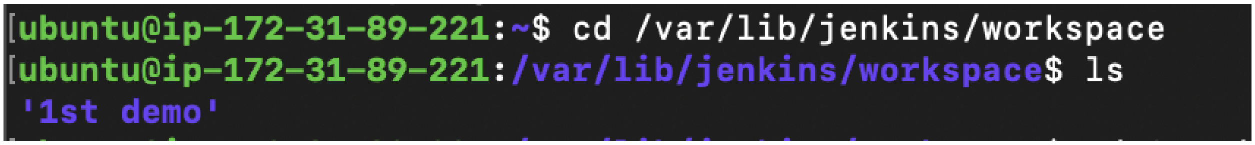
Command

See the list of available environment variables

```
echo “Hello World”
mkdir -p demo
```

Built the Job and **checked** workspace directory:

```
cd /var/lib/jenkins/workspace
```



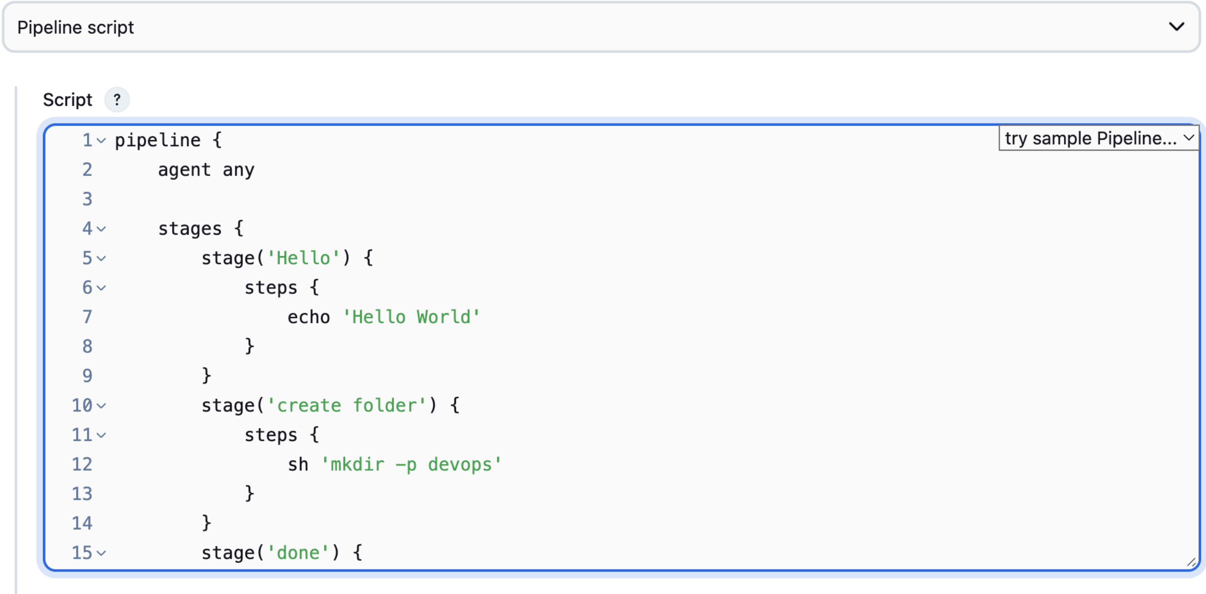
```
[ubuntu@ip-172-31-89-221:~$ cd /var/lib/jenkins/workspace
[ubuntu@ip-172-31-89-221:/var/lib/jenkins/workspace$ ls
 '1st demo'
```

Task 2 – Learning Declarative Pipeline

- **Created a New Pipeline Job:**
    - Inputted item name: `demo-multistage-pipeline`
    - Selected "Pipeline" project type.
    - Wrote the following **Declarative Pipeline** YAML script:

```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
            }
        }
        stage('create folder') {
            steps {
                sh 'mkdir -p devops'
            }
        }
        stage('done') {
            steps {
                echo 'folder made'
            }
        }
    }
}
```

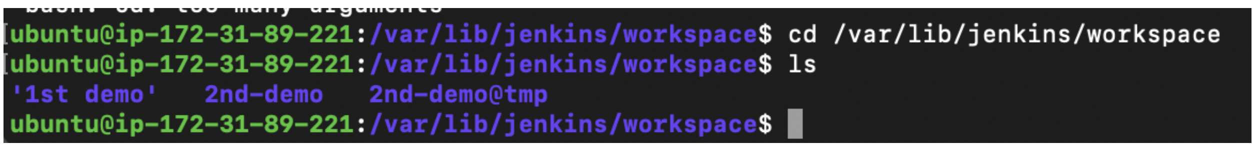


```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
            }
        }
        stage('create folder') {
            steps {
                sh 'mkdir -p devops'
            }
        }
        stage('done') {
```

Built the Pipeline Job and verified workspace:

```
cd /var/lib/jenkins/workspace
```

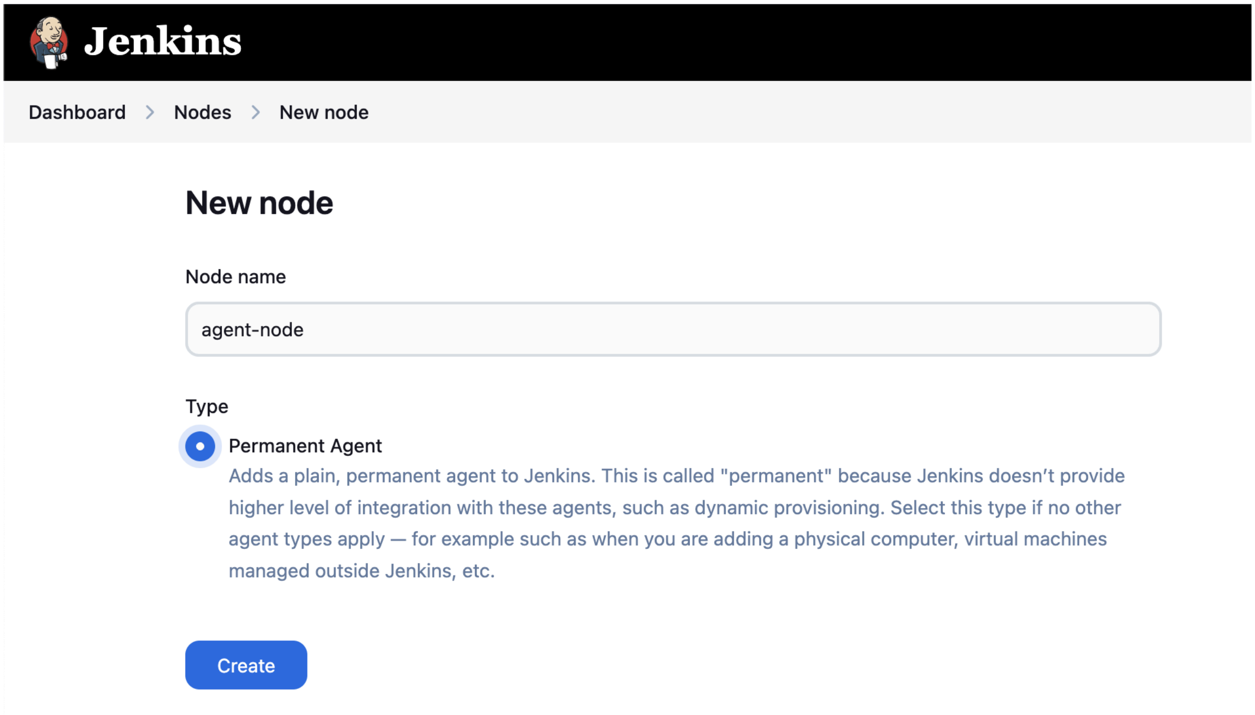


```
bash: cd: too many arguments
[ubuntu@ip-172-31-89-221:/var/lib/jenkins/workspace$ cd /var/lib/jenkins/workspace
[ubuntu@ip-172-31-89-221:/var/lib/jenkins/workspace$ ls
 '1st demo'   2nd-demo   2nd-demo@tmp
ubuntu@ip-172-31-89-221:/var/lib/jenkins/workspace$ 
```

Task 3 – Setting Up an Agent Node

- **Created a New EC2 Instance** (Agent):
    - Only **installed Java** (Jenkins installation not required on agent).

```
sudo apt update
sudo apt install fontconfig openjdk-21-jre
java -version
openjdk version "21.0.3" 2024-04-16
OpenJDK Runtime Environment (build 21.0.3+11-Debian-2)
OpenJDK 64-Bit Server VM (build 21.0.3+11-Debian-2, mixed mode,
sharing)
```



Jenkins

Dashboard > Nodes > New node

New node

Node name

agent-node

Type

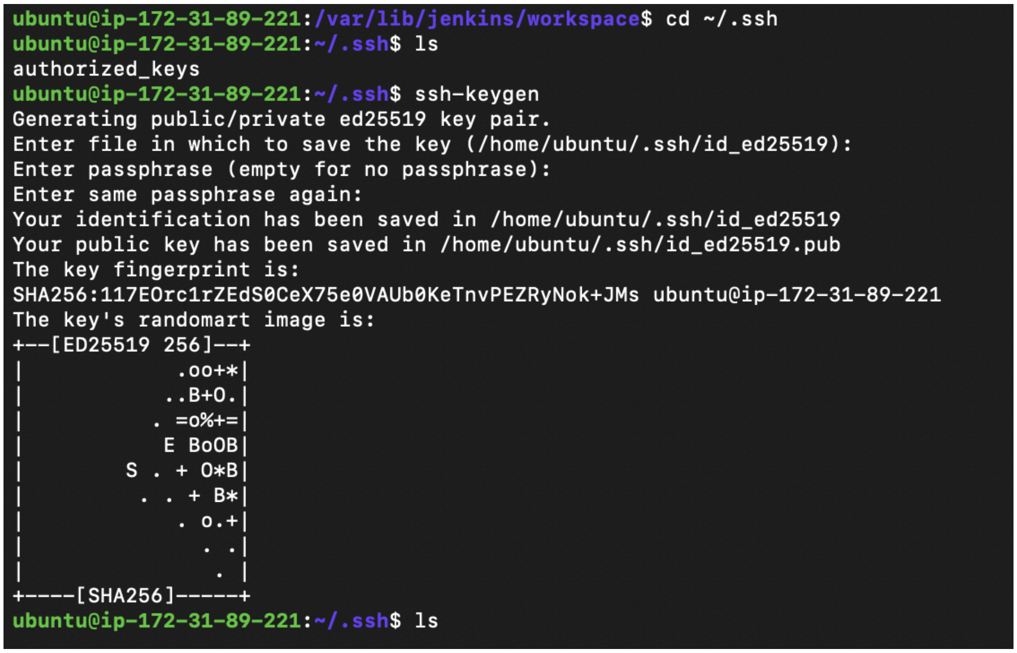
● Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Create

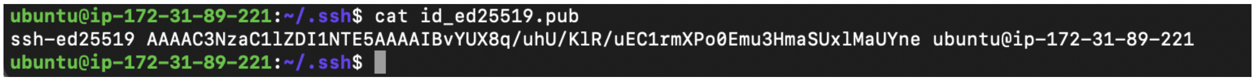
- **Configured SSH Authentication:**
  - On **Master EC2 (Jenkins Server):**

Created SSH key pair: `ssh-keygen`



```
ubuntu@ip-172-31-89-221:/var/lib/jenkins/workspace$ cd ~/.ssh
ubuntu@ip-172-31-89-221:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-89-221:~/.ssh$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:117EOrc1rZEdS0CeX75e0VAUb0KeTnvPEZRyNok+JMs ubuntu@ip-172-31-89-221
The key's randomart image is:
+--[ED25519 256]--+
|            .oo+*|
|           ..B+O.|
|          . =o%+=|
|           E BoOB|
|        S . + O*B|
|         . . + B*|
|            . o.+|
|              . .|
|               . |
+----[SHA256]-----+
ubuntu@ip-172-31-89-221:~/.ssh$ ls
```

Copied the contents of the public key (`~/.ssh/id_rsa.pub`).



```
ubuntu@ip-172-31-89-221:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBvYUX8q/uhU/KlR/uEC1rmXPo0Emu3HmaSUxlMaUYne ubuntu@ip-172-31-89-221
ubuntu@ip-172-31-89-221:~/.ssh$ 
```

On **Agent EC2:**
Pasted the public key into the `~/.ssh/authorized_keys` file.

✅ Now the Jenkins master can communicate with the agent securely via SSH!

📌 Key Takeaways

- Successfully installed and set up Jenkins on AWS.
- Created my first Freestyle Job.
- Learned basics of Declarative Pipeline in Jenkins.
- Understood multi-stage pipelines and pipeline syntax.
- Configured an EC2 instance as a Jenkins agent for future distributed builds.