	Pimpri Chinchwad Education Trust's **Pimpri Chinchwad College of Engineering (PCCoE)** (An Autonomous Institute) Affiliated to Savitribai Phule Pune University(SPPU) ISO 21001:2018 Certified by TUV	

Course: DevOps Laboratory	**Code:** BIT26VS01
Name: Amar Vaijinath Chavan	**PRN:** 124B2F001
Assignment 4: Create a Freestyle Jenkins job: Configure a job to print "Hello, Jenkins!" in the console output and view logs.	

Aim: To create a basic Freestyle job in Jenkins, execute a shell command on the Master node, and Analyze the console output .

Objectives:

1. To understand the Jenkins Freestyle project workspace.
2. To learn the process of manual build triggering.
3. To study how Jenkins captures and displays logs on the Controller (Master) node.

Prerequisites:

- Jenkins Master node installed and running on AWS EC2.
- Admin access to the Jenkins Dashboard.

Theory:

1. Freestyle Project on Master

In a standard Jenkins setup, the Master node can act as its own execution agent. This is useful for simple administrative tasks or initial testing. A Freestyle job provides a menu-driven interface to define these tasks without needing complex Groovy scripts.

2. Workspace Management

When a job runs on the Master, Jenkins creates a specific directory called a **Workspace** (usually located at /var/lib/jenkins/workspace/). This is where Jenkins performs the work and stores any temporary files generated during the build.

3. Console Output

The Console Output is the "heart" of Jenkins debugging. It provides a transparent view of:

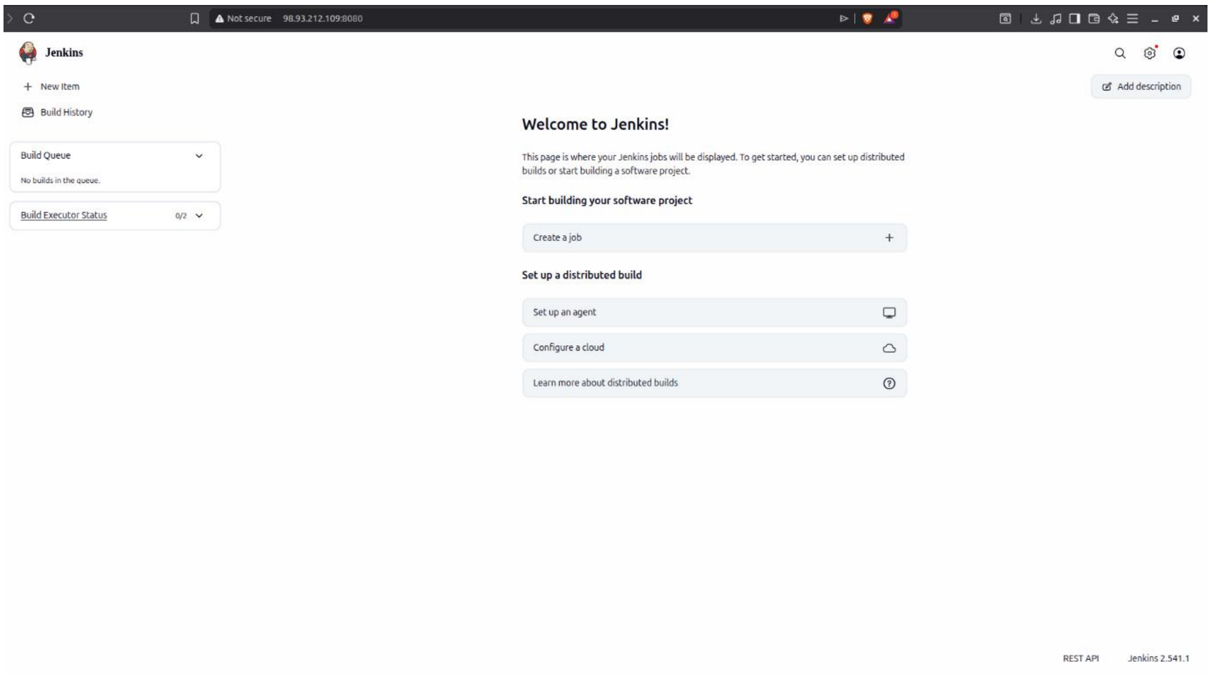
- **Who** triggered the build.
- **Where** the build is running (Master vs. Worker).

- **What** commands were executed.
- **The Result** (Success/Failure status).

Practical Procedure / Steps:

Step 1: Create the Job

1. Open the Jenkins Dashboard and click **New Item**.
2. Name the project: <First-Jenkins-Job>.
3. Select **Freestyle project** and click **OK**.


Not secure 98.93.212.109:8080

Jenkins

New Item

Build History

Build Queue

No builds in the queue.

Build Executor Status 0/2

Add description

Welcome to Jenkins!

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.

Start building your software project

Create a job

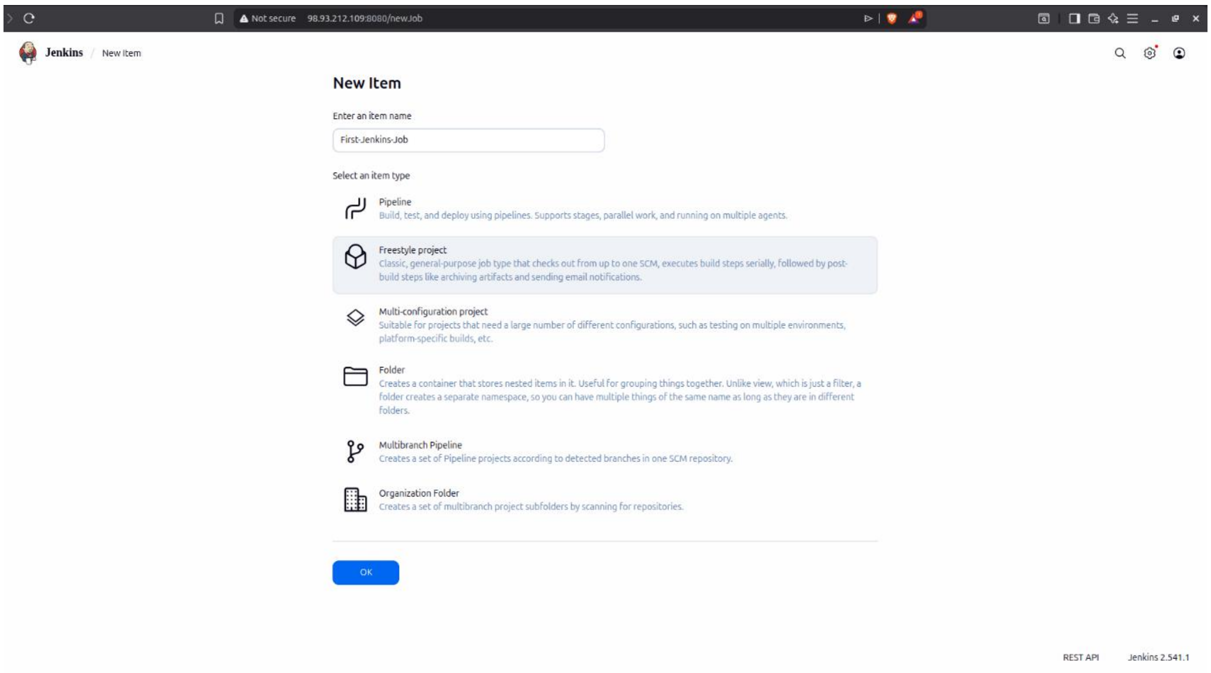
Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

REST API Jenkins 2.541.1


Not secure 98.93.212.109:8080/newJob

Jenkins / New Item

New Item

Enter an item name

First-Jenkins-Job

Select an item type

Pipeline
Build, test, and deploy using pipelines. Supports stages, parallel work, and running on multiple agents.

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.

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

REST API Jenkins 2.541.1

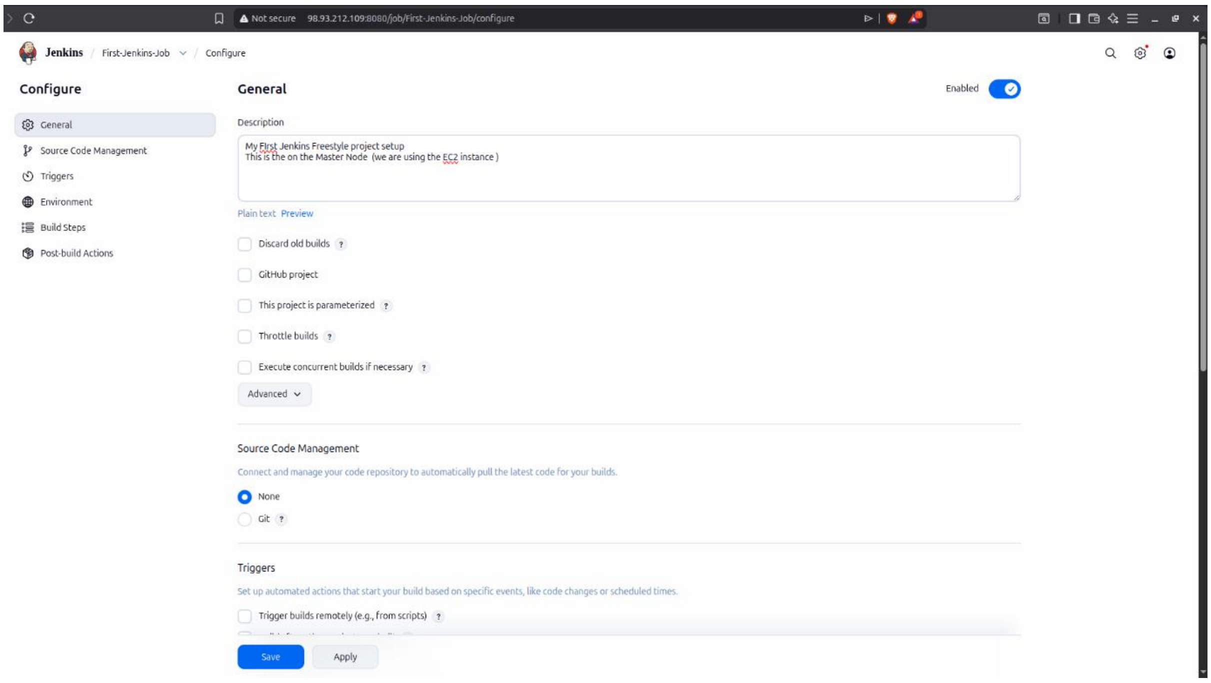
Step 2: Restricting to Master (Internal Logic)

Since we are not using the worker node for this task, we ensure it runs on the Master.

1. Under the **General** tab, ensure the option "Restrict where this project can be run" is **unchecked**, or specifically label it as built-in (the default label for the Master node).

Step 3: Add Build Step

1. Scroll down to the **Build Steps** section.
2. Click **Add build step** -> **Execute shell**.
3. Enter the command:


Not secure 98.93.212.109:8080/job/First-Jenkins-Job/configure

Jenkins / First-Jenkins-Job / Configure

Configure

General

Enabled

Description

My FIrst Jenkins Freestyle project setup
This is the on the Master Node (we are using the EC2 instance)

- [ ] Discard old builds
- [ ] GitHub project
- [ ] This project is parameterized
- [ ] Throttle builds
- [ ] Execute concurrent builds if necessary

Source Code Management

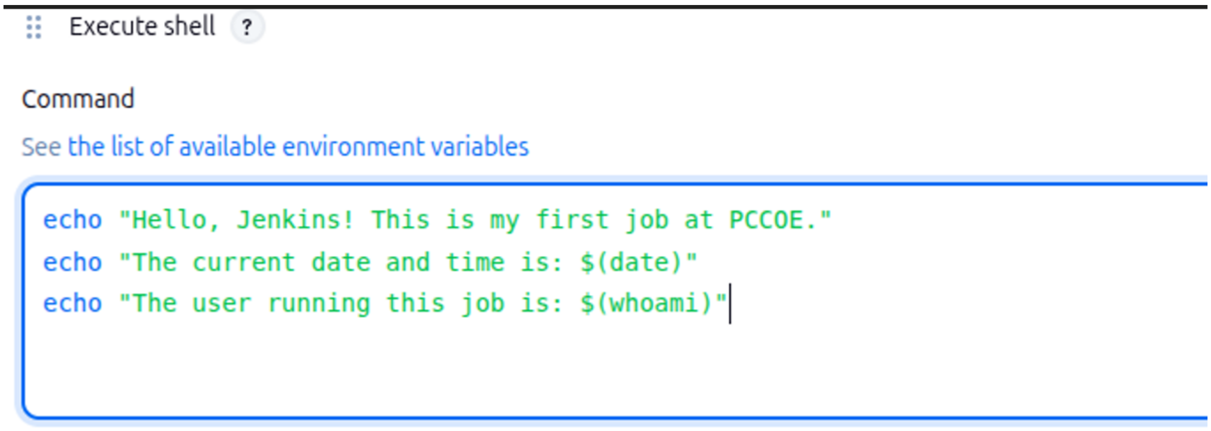
Connect and manage your code repository to automatically pull the latest code for your builds.

[✓ None] Git

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- [ ] Trigger builds remotely (e.g., from scripts)

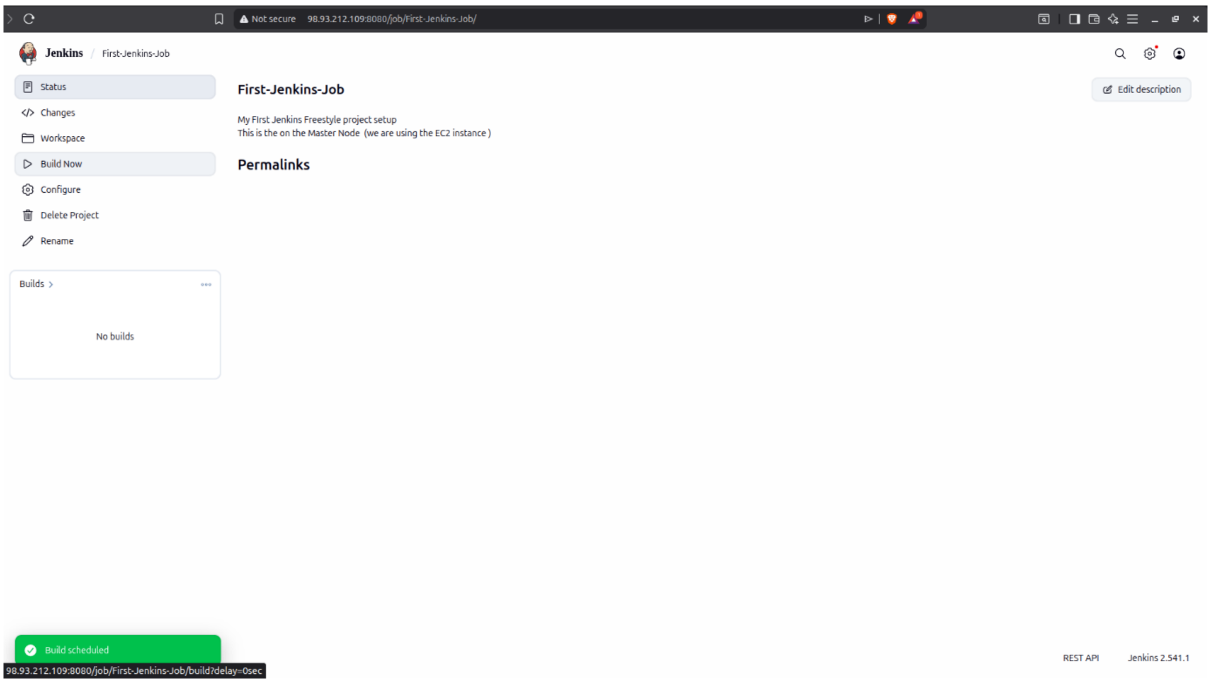

Execute shell

Command

```
echo "Hello, Jenkins! This is my first job at PCCOE."
echo "The current date and time is: $(date)"
echo "The user running this job is: $(whoami)"
```

Step 4: Execute and Verify

1. Click **Save** and then click **Build Now**.
2. Once the build appears in the **Build History** (Green circle), click on the build number.
3. Click **Console Output**.
4. Check in the EC2 Machine (Terminal) /var/lib/jenkins/workspace/ the project folder is created.

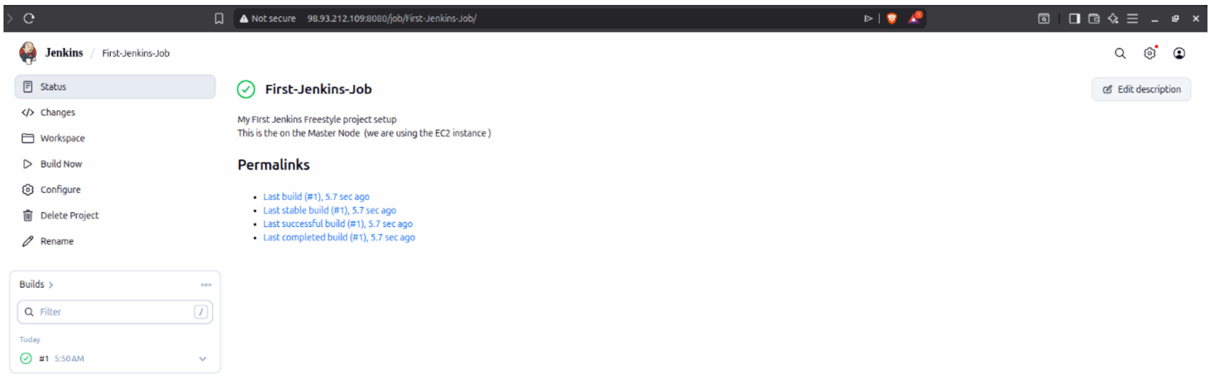


First-Jenkins-Job

My First Jenkins Freestyle project setup
This is the on the Master Node (we are using the EC2 instance)

Build scheduled

98.93.212.109:8080/job/First-Jenkins-Job/build?delay=0sec

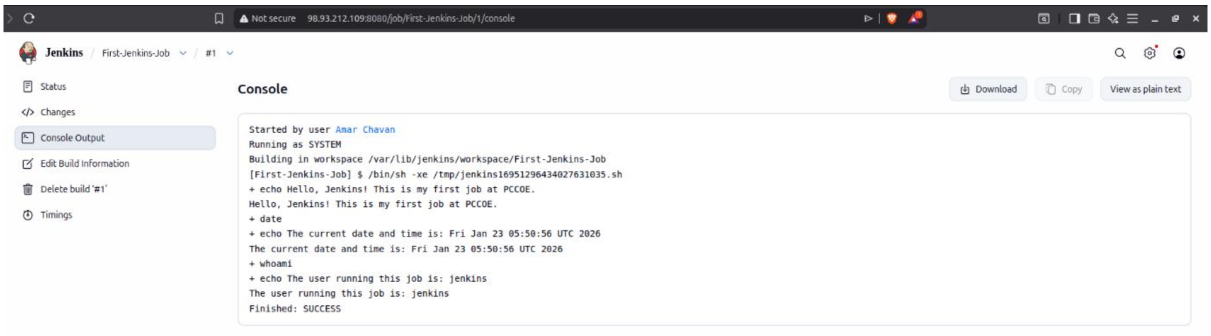


First-Jenkins-Job

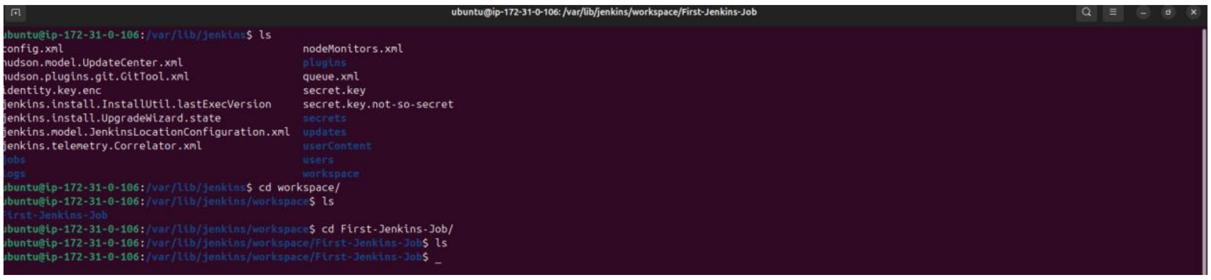
My First Jenkins Freestyle project setup
This is the on the Master Node (we are using the EC2 instance)

- Last build (#1), 5.7 sec ago
- Last stable build (#1), 5.7 sec ago
- Last successful build (#1), 5.7 sec ago
- Last completed build (#1), 5.7 sec ago

#1 5:50 AM



```
Started by user Amar Chavan
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/First-Jenkins-Job
[First-Jenkins-Job] $ /bin/sh -xe /tmp/jenkins16951296434027631035.sh
+ echo Hello, Jenkins! This is my first job at PCCOE.
Hello, Jenkins! This is my first job at PCCOE.
+ date
+ echo The current date and time is: Fri Jan 23 05:50:56 UTC 2026
The current date and time is: Fri Jan 23 05:50:56 UTC 2026
+ whoami
+ echo The user running this job is: jenkins
The user running this job is: jenkins
Finished: SUCCESS
```


ubuntu@ip-172-31-0-106: /var/lib/jenkins/workspace/First-Jenkins-Job

```
buntu@ip-172-31-0-106:/var/lib/jenkins$ ls
onfig.xml                                   nodeMonitors.xml
udson.model.UpdateCenter.xml                plugins
udson.plugins.git.GitTool.xml               queue.xml
dentity.key.enc                             secret.key
enkins.install.InstallUtil.lastExecVersion  secret.key.not-so-secret
enkins.install.UpgradeWizard.state          secrets
enkins.model.JenkinsLocationConfiguration.xml updates
enkins.telemetry.Correlator.xml             userContent
obs                                         users
ogs                                         workspace
buntu@ip-172-31-0-106:/var/lib/jenkins$ cd workspace/
buntu@ip-172-31-0-106:/var/lib/jenkins/workspace$ ls
irst-Jenkins-Job
buntu@ip-172-31-0-106:/var/lib/jenkins/workspace$ cd First-Jenkins-Job/
buntu@ip-172-31-0-106:/var/lib/jenkins/workspace/First-Jenkins-Job$ ls
buntu@ip-172-31-0-106:/var/lib/jenkins/workspace/First-Jenkins-Job$ _
```

Conclusion:

The successful execution of this assignment demonstrates the fundamental capability of Jenkins to automate simple command-line tasks through a Freestyle Project. By running the job directly on the Master (Controller) node, we observed how Jenkins manages its internal Workspace environment, automatically creating dedicated directories for each project to ensure organized file handling. The analysis of the Console Output highlighted the transparency of the Jenkins build process, providing a detailed audit trail of the user who triggered the event, the exact commands executed, and the final status of the operation. This process serves as the essential first step in understanding CI/CD automation, proving that the Jenkins environment is correctly configured to handle script execution. Furthermore, verifying the physical folder creation within the EC2 terminal (/var/lib/jenkins/workspace/) bridged the gap between the web-based UI and the underlying Linux filesystem. Ultimately, this assignment establishes a solid foundation for more complex automation tasks, such as source code integration and distributed building across multiple worker nodes.