

## Jenkins

**Task 1:** Create a Basic Jenkins Job

**Topic:** Jenkins Essentials - Job Configuration in Jenkins

**Description:** Set up a basic Jenkins job that pulls from a public Git repository and executes a simple shell command.

**Steps:**

Log into Jenkins with the provided credentials.

Navigate to the Jenkins dashboard and select 'New Item'.

Enter a name for the job, select 'Freestyle project', and click OK.

In the Source Code Management section, select Git and enter the URL of a public repository (e.g., a GitHub repo URL).

In the Build section, add a 'Execute shell' build step with a command like `echo "Hello World"`.

Save the job and click 'Build Now' to run it.

Observe the build output to ensure the command executed successfully.

### Step 1:

Log into Jenkins with your provided credentials.

### Step 2:

- Navigate to the Jenkins dashboard.
- Click on 'New Item' or 'Create a job'.
- Enter a name for your job (e.g., "BasicJob").
- Select 'Freestyle project'.
- Click 'OK'.

### Step 3:

#### Source Code Management:

- In the 'Source Code Management' section, select 'Git'.
- Enter the URL of a public Git repository (e.g., <https://github.com/user/repo.git>).

#### Build Section:

- Scroll down to the 'Build' section.
- Click on 'Add build step'.
- Select 'Execute shell'.

`echo "Hello World"`

#### **Step 4:**

- Click 'Save' to save the job configuration.
- On the job's main page, click 'Build Now' to run the job.

#### **Step 5:**

- Once the build is complete, click on the build number (e.g., #1) under 'Build History' on the left side.
- Click on 'Console Output' to view the build details.
- Ensure the command echo "Hello World" executed successfully and the output is displayed.

### **Task 2: Configure Jenkins to Run a Simple Maven Build**

**Topic:** Building with Jenkins - Automating Builds

**Description:** Configure Jenkins to build a Maven-based Java project.

#### **Steps:**

Create a new Jenkins job or select an existing job.

Under the Build section, add a build step for 'Invoke top-level Maven targets'.

In the Goals field, input clean install to run the corresponding Maven goals.

Save the job and click 'Build Now'.

After the build completes, check the console output for a 'BUILD SUCCESS' message.

#### **Step 1:**

- Log into Jenkins with your provided credentials.
- Navigate to the Jenkins dashboard.
- If creating a new job:
  1. Click on 'New Item'.
  2. Enter a name for your job (e.g., "MavenBuildJob").
  3. Select 'Freestyle project'.
  4. Click 'OK'.
  5. If using an existing job, navigate to that job and click 'Configure'.

**Step 2:**

- In the 'Source Code Management' section, select 'Git'.
- Enter the URL of your Maven-based Java project's Git repository (e.g., <https://github.com/user/repo.git>).

**Step 3:**

- Scroll down to the 'Build' section.
- Click on 'Add build step'.
- Select 'Invoke top-level Maven targets'.

**Step 4:**

- In the 'Goals' field, input clean install to run the corresponding Maven goals.

**Step 5:**

- Click 'Save' to save the job configuration.
- On the job's main page, click 'Build Now' to run the job.

**Step 6:**

- Once the build is complete, click on the build number (e.g., #1) under 'Build History' on the left side.
- Click on 'Console Output' to view the build details.
- Ensure the build completes with a 'BUILD SUCCESS' message.