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.

Logging into Jenkins:

- Learned how to access the Jenkins dashboard by navigating to the Jenkins URL and logging in with provided credentials.
-

Creating a New Jenkins Job:

- Understood the process of initiating a new job by selecting 'New Item' from the Jenkins dashboard.
- Learned the importance of naming conventions for jobs for easy identification and management.
- Explored different types of Jenkins jobs and specifically how to set up a 'Freestyle project'.
- Configuring Source Code Management (SCM):

Gained knowledge on how to integrate Jenkins with a version control system (VCS).

- Learned to select 'Git' in the SCM section and provide the URL of a public Git repository.
- Understood the significance of proper repository URL configuration for successful code retrieval.

Adding a Build Step:

- Discovered how to add a build step to a Jenkins job.
- Learned to select 'Execute shell' from the available build step options.
- Practiced writing a simple shell command (echo "Hello World") to be executed during the build process.

Saving and Running:

- Understood the importance of saving job configurations to preserve changes.
- Learned how to trigger the job manually using the 'Build Now' option.

Observing Build Output:

- Acquired skills to monitor build progress and access build history.
- Learned to navigate to the build’s detail page and check the 'Console Output' for debugging and validation.
- Verified that the build executed successfully by ensuring the output contained the expected Hello World message.

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' me

Creating or Selecting a Jenkins Job:

New Job Creation: Learned how to create a new Jenkins job by navigating to the Jenkins dashboard, clicking on "New Item", entering a job name, selecting "Freestyle project", and clicking "OK".

Selecting Existing Job: Understood how to select an existing job from the list on the Jenkins dashboard for modification.

Configuring Source Code Management (SCM) (Optional):

Integrating with Git: If the Maven project resides in a Git repository, learned to configure the SCM section by selecting "Git" and entering the repository URL to enable Jenkins to pull the project code.

Adding a Build Step for Maven:

Build Section: Scrolled to the "Build" section and clicked "Add build step".

Maven Invocation: Selected "Invoke top-level Maven targets" from the list of build steps, enabling Jenkins to run Maven commands as part of the build process.

Specifying Maven Goals:

Maven Goals Field: Entered clean install in the "Goals" field to ensure that Jenkins runs the corresponding Maven goals. This command cleans the project, removes the target directory, and then compiles and packages the project.

Saving the Job:

Save Configuration: Saved the job configuration by scrolling to the bottom and clicking the "Save" button, ensuring all changes were preserved.

Running the Job:

Manual Trigger: Triggered the job manually by clicking the "Build Now" link on the job's main page, initiating the build process.

Observing Build Output:

Build Progress: Monitored the build progress in the "Build History" section.

Console Output: Accessed the build’s detail page by clicking on the build number and then clicked on "Console Output" to view the logs.

Build Success Verification: Verified the build success by ensuring the console output contained a BUILD SUCCESS message, indicating that the Maven goals executed successfully.