Creating a "Hello World" Jenkins project is a great way to get started with Jenkins and understand its basic functionality. Here are a few simple project ideas that can help you learn Jenkins concepts and pipelines.

# 1. Hello World Console Output (Freestyle Project)

Objective: Create a simple Jenkins job that prints "Hello World" to the console.

Steps:

1. Create a New Freestyle Project:

- Go to Jenkins Dashboard.

- Click on `New Item`.

- Enter a name (e.g., `HelloWorldFreestyle`).

- Select `Freestyle project` and click `OK`.

2. Add a Build Step:

- Under the `Build` section, click `Add build step`.

- Choose `Execute shell`

3. Write the Script:

echo “Hello World”

4. Save and Build:

- Click `Save`.

- Click `Build Now` and check the console output to see the "Hello World" message.

# 2. Hello World Scripted Pipeline

Objective: Create a Jenkins Pipeline project that prints "Hello World" using a Jenkinsfile.

Steps:

1. Create a New Pipeline Project:

- Go to Jenkins Dashboard.

- Click on `New Item`.

- Enter a name (e.g., `HelloWorldPipeline`).

- Select `Pipeline` and click `OK`.

2. Write Pipeline Script:

- Scroll down to the `Pipeline` section.

- Ensure `Pipeline script` is selected and enter the following script:

pipeline {

agent any

stages {

stage('Hello') {

steps {

echo 'Hello'

}

}

stage('World') {

steps {

echo 'World'

}

}

}

}

3. Save and Build:

- Click `Save`.

- Click `Build Now` and check the console output.

# 3. Hello World from a Git Repository

Objective: Set up a Jenkins job that pulls a "Hello World" script from a Git repository and executes it.

Steps:

1. Set Up a Git Repository:

- Create a simple script that prints "Hello World" (e.g., `hello.sh` for Unix or `hello.bat` for Windows).

- Push this script to a Git repository (e.g., GitHub).

Note: for testing I have put sh 'ls -al' in the command so that files from the repo are displayed.

2. Create a Freestyle or Pipeline Project:

- Go to Jenkins Dashboard.

- Click on `New Item`.

- Enter a name (e.g., `HelloWorldFromGit`).

- Select either `Freestyle project` or `Pipeline` depending on your preference.

3. Configure Git Repository:

- If Freestyle:

- Under `Source Code Management`, select `Git`.

- Enter the repository URL.

- If Pipeline:

- Use a `Pipeline script from SCM` option.

- Choose `Git` and provide the repository URL.

4. Add Build Step:

- If Freestyle, add a build step to execute the script:

sh 'ls -al'

- If Pipeline, your `Jenkinsfile` in the repository could look like this:

pipeline {

agent any

stages {

stage('Checkout') {

steps {

git 'https://github.com/your-repo.git'

}

}

stage('Run Script') {

steps {

sh 'ls -al'

}

}

}

}

5. Save and Build:

- Click `Save`.

- Click `Build Now` and verify that the "Hello World" message is displayed.

# 4. Scheduled Hello World (Cron Job)

Objective: Create a Jenkins job that prints "Hello World" at regular intervals using a cron schedule.

Steps:

1. Create a Freestyle or Pipeline Project.

2. Add a Build Step:

- Use the same steps as in the first or second idea.

3. Set a Build Trigger:

- Under `Build Triggers`, check `Build periodically`.

- Enter a cron schedule (e.g., `H/5 \* \* \* \*` to run every 5 minutes).

4. Save:

- Jenkins will automatically build the project based on the schedule.

Note:

The smallest interval you can specify using Jenkins' cron syntax is one minute, command: \* \* \* \* \*

Write a pipeline for 5 sec:

pipeline {

agent any

stages {

stage('Repeat Every 5 Seconds') {

steps {

script {

for (int i = 0; i < 12; i++) {

echo "Running iteration ${i + 1}"

sh 'sleep 5'

}

}

}

}

}

}

5. Parameterize Hello World

Objective: Create a Jenkins job that takes a user input (e.g., a name) and prints "Hello [Name]" to the console.

Steps:

1. Create a Freestyle or Pipeline Project.

2. Add a String Parameter:

- Under `General`, check `This project is parameterized`.

- Add a `String Parameter` with the name `USERNAME` and a default value (e.g., `World`).

3. Modify Build Step:

- For Freestyle, update the script to:

```bash

echo "Hello $USERNAME"

```

- For Pipeline, update the script to:

```groovy

pipeline {

agent any

parameters {

string(name: 'USERNAME', defaultValue: 'World', description: '')

}

stages {

stage('Hello') {

steps {

echo "Hello ${params.USERNAME}"

}

}

}

}

```

4. Save and Build:

- When building, Jenkins will prompt for the `USERNAME` parameter.

These simple "Hello World" projects will help you get comfortable with Jenkins' basic functionalities and features, such as Freestyle projects, Pipelines, Git integration, scheduling, and parameterization.