

Lab 2: Simple Jenkins job automation with Groovy

Objective

- Use Groovy scripts to automate Jenkins jobs
 - Create jobs dynamically (DSL)
 - Trigger builds programmatically
 - Query build results & artifacts
 - Lay foundation for CI/CD automation
-

Step 1 — Running Groovy inside Jenkins

- Go to: **Manage Jenkins → Script Console**
 - Or use a **Jenkins pipeline with Groovy steps**
 - You can also install the **Job DSL Plugin** for declarative job creation
-

Step 2 — List All Jenkins Jobs

```
1 Jenkins.instance.items.each { job ->
2     println "Job: ${job.name}"
3 }
```

Step 3 — Create a Simple Freestyle Job (DSL)

```
1 import jenkins.model.Jenkins
2 import hudson.model.FreeStyleProject
3 import hudson.tasks.Shell
4
5 def jobName = "demo-job"
6 def jenkins = Jenkins.get()
7
8 if (jenkins.getItem(jobName) == null) {
9     def job = jenkins.createProject(FreeStyleProject, jobName)
10    job.setDescription("Created via Groovy Script")
11    println "Job '${jobName}' created."
12 } else {
13     println "Job '${jobName}' already exists."
14 }
```

Step 4 — Add a Simple Build Step

```
1 // Add or replace shell build step
2 job.buildersList.clear()
3 job.buildersList.add(new Shell("echo 'Hello from Groovy-created
4 job!'""))
5 job.save()
6 println "Shell step added to '${jobName}'."
```

Step 5 — Trigger a Build Programmatically

```
1 // Step 3: Trigger Build and Wait
```

```
2 def future = job.scheduleBuild2(0) // no delay
3 def build = future.get()           // wait until build finishes
4 println "Step 3: Build #${build.number} completed with status:
   ${build.result}"
```

Step 6 — Read Build Logs (Add separate build step “ Execute system Groovy script”)

```
1 import jenkins.model.Jenkins
2
3 def jobName = "demo-job"
4 def job = Jenkins.get().getItem(jobName)
5 def lastBuild = job.getLastBuild()
6
7 if (lastBuild != null) {
8     println "Build #${lastBuild.number} log:"
9     println lastBuild.getLog().join("\n")
10 } else {
11     println "No builds found for job '${jobName}'."
12 }
```

Validation

- Run the scripts in **Script Console** → check if jobs appear in Jenkins dashboard.
- Trigger jobs → ensure console logs show expected output.
- Export job config XML (`job.configFile.asString()`) for verification.