CONTINUOUS INTEGRATION AND CONTINUOUS DEPLOYMENT LAB

Lab File
(2023-2024)

for

5th Semester

Submitted To

Dr. Hitesh Kumar Sharma
CI/CD Professor, Cluster Head
(Cybernetics)
School of Computer Science

Submitted By:

Arpit Goyal
B. Tech. CSE DevOps [5th Semester]
500094790
R2142210148
B-3

EXPERIMENT 1

Setting Up a Jenkins Job for Maven Build

Aim

Create a Jenkins job that builds a Maven project using Jenkins and triggers the build on changes in the version control repository.

Steps

1. Create a maven project
    a. Create the project

```
[INFO]
[INFO] --- surefire:3.1.2:test (default-test) @ cicd_Lab_1 ---
[INFO]
[INFO] --- jar:3.1.0:jar (default-jar) @ cicd_Lab_1 ---
[INFO]
[INFO] --- install:3.1.1:install (default-install) @ cicd_Lab_1 ---
[INFO] Installing C:\Users\hp\eclipse-workspace\cicd_Lab_1\pom.xml to C:\Users\hp\.m2\repository\DevOps_B3_Cicd\cicd_Lab_1\0.0.1-SNAPSHOT\cic
[INFO] Installing C:\Users\hp\eclipse-workspace\cicd_Lab_1\target\mycal.jar to C:\Users\hp\.m2\repository\DevOps_B3_Cicd\cicd_Lab_1\0.0.1-SNA
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.845 s
[INFO] Finished at: 2023-12-01T01:20:55+05:30
[INFO] ------------------------------------------------------------------------
```

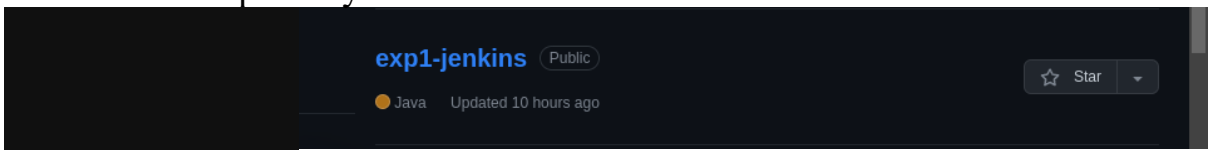
b. Create a package mypkg inside src/main/java/ and add myCalculator.java file

mycal.java × | cicd_Lab_1/pom.xml | cicd.lab3/pom.xml

```java
package cicd_Lab_1;

public class mycal {
    public int sum(int a,int b)
    {
        return (a+b);
    }
    public int diff(int a, int b)
    {
        return (a-b);
    }
    public int mul(int a, int b)
    {
        return(a*b);
    }
    public int div(int a,int b)
    {
        return(a/b);
    }

    public static void main(String[] args) {
        System.out.println("hello");
        mycal ob = new mycal();
        System.out.println(ob.sum(21,4));
        System.out.println(ob.diff(21,4));
        System.out.println(ob.mul(21,4));
        System.out.println(ob.div(21,4));

    }
```

2. Push the project on GitHub

a. Create a repository on GitHub



b. In the project run git init command

```
→  exp1-jenkins git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ayroid/Documents/Work/College/SEM5/Subjects/cicd/lab/HiteshSir/exp1/exp1-jenk
ins/.git/
→  exp1-jenkins git:(master) ✗
```

c. Link the local repository and the remote repository

```
→  exp1-jenkins git:(master) ✗ git remote add origin https://github.com/Ayroid/exp1-jenkins.git
→  exp1-jenkins git:(master) ✗
```

d. Add, Commit and Push the project

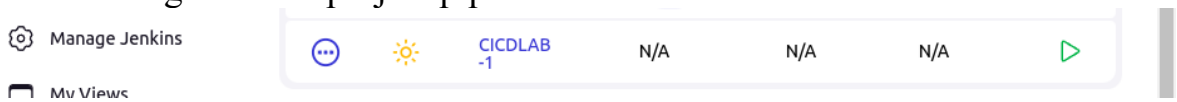
```
→  exp1-jenkins git:(master) ✗ git add .
```

3. Install and start Jenkins

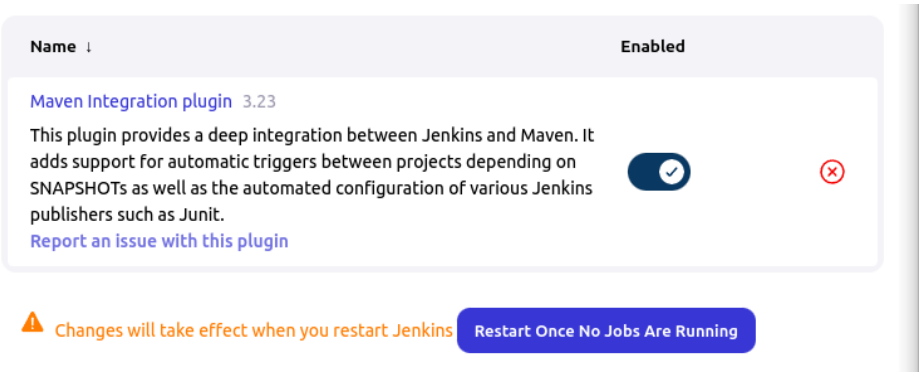
```
→  exp1-jenkins git:(master) jenkins --version
2.414.1
→  exp1-jenkins git:(master)
```

4. Create Jenkins job to clone GitHub project and make a build

a. Creating a maven project pipeline named CICDLAB-1



b. Installing maven integration plugin - Dashboard > Manage Jenkins > Plugins > Available plugins > Maven Integration plugin



c. Configuring the pipeline

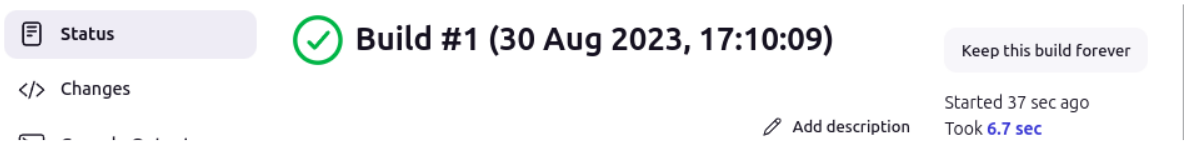
i. Provide GitHub repository url in Dashboard > CICDLAB-1 > Configuration > Source Code Management > Git > Repositories > Repository URL

ii. Specify the branch in Dashboard > CICDLAB-1 > Configuration > Source Code Management > Git > Branches to build > Branch Specifier

iii. Specify Build Goal in Dashboard > CICDLAB-1 > Configuration > Build > Goals and options

iv. Save the changes

5. Execute the build



6. Add Build Trigger for automated builds by scheduling them in Dashboard > CICDLAB-1 > Configuration > Build Triggers > Poll SCM > Schedule

7. Make changes to the project

```java
    public static int power(int a, int b) {
        int result = 1;
        for (int i = 0; i < b; i++) {
            result *= a;
        }
        return result;
    }
```

8. Commit and Push the changes

```
→  exp1-jenkins git:(master) ✗ git add .
→  exp1-jenkins git:(master) ✗ git commit -m "Added power calculation method"
[master 00ccf4d] Added power calculation method
 2 files changed, 9 insertions(+)
→  exp1-jenkins git:(master) git push
Username for 'https://github.com': Ayroid
Password for 'https://Ayroid@github.com':
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (11/11), 1.25 KiB | 1.25 MiB/s, done.
Total 11 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/Ayroid/exp1-jenkins.git
   165bb98..00ccf4d  master -> master
→  exp1-jenkins git:(master)
```

9. An automated build is triggered