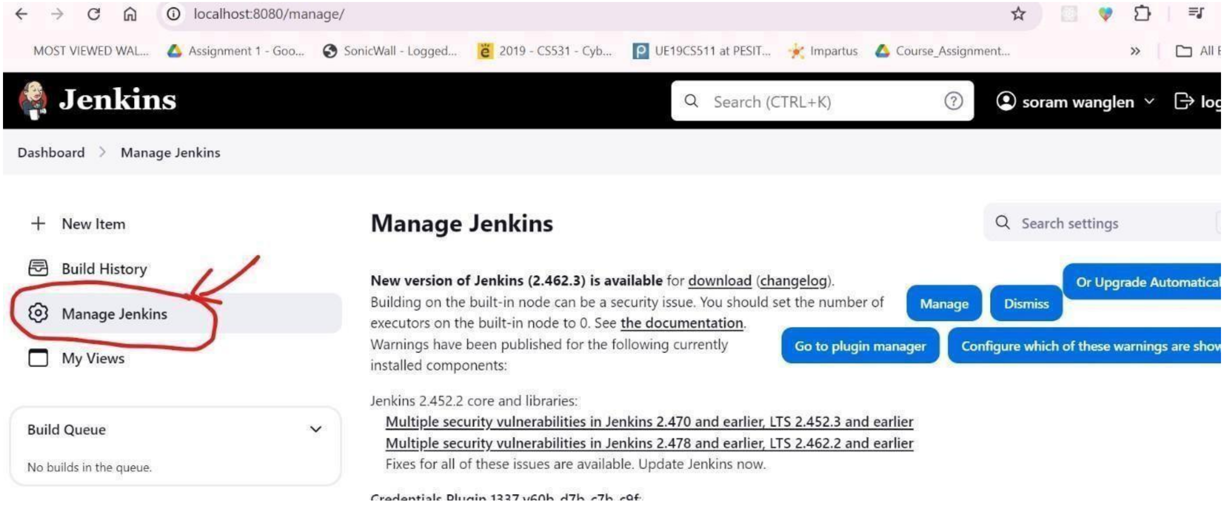
Practical 8

Titel : Use Jenkins to set up a distributed pipeline that will compile and test a Maven project on two different slave nodes respectively.

Step 1: Create two slave nodes in Jenkins.



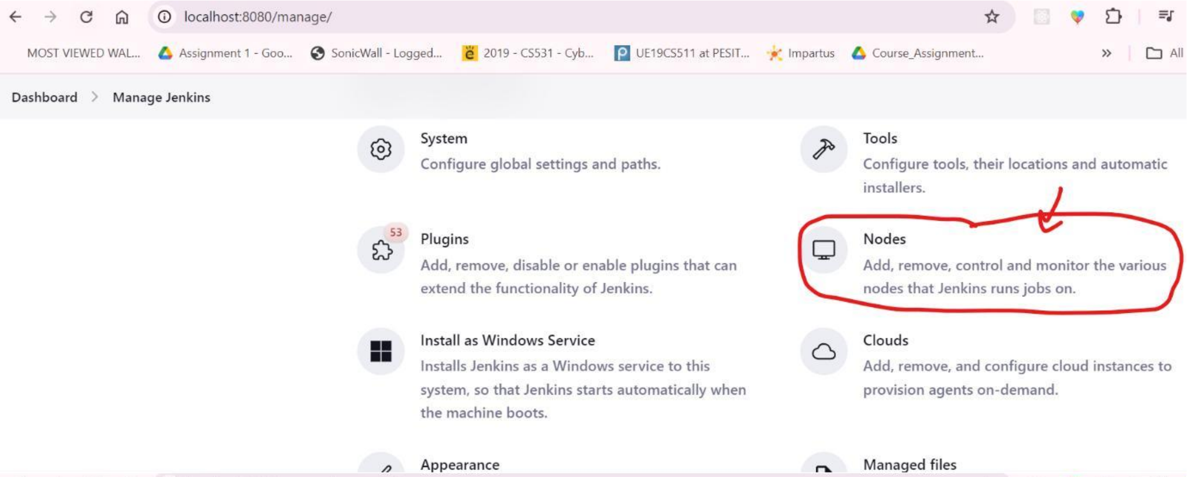
Manage Jenkins

New version of Jenkins (2.462.3) is available for download (changelog).

Building on the built-in node can be a security issue. You should set the number of executors on the built-in node to 0. See the documentation.

Warnings have been published for the following currently installed components:

Jenkins 2.452.2 core and libraries:
Multiple security vulnerabilities in Jenkins 2.470 and earlier, LTS 2.452.3 and earlier
Multiple security vulnerabilities in Jenkins 2.478 and earlier, LTS 2.462.2 and earlier
Fixes for all of these issues are available. Update Jenkins now.



System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

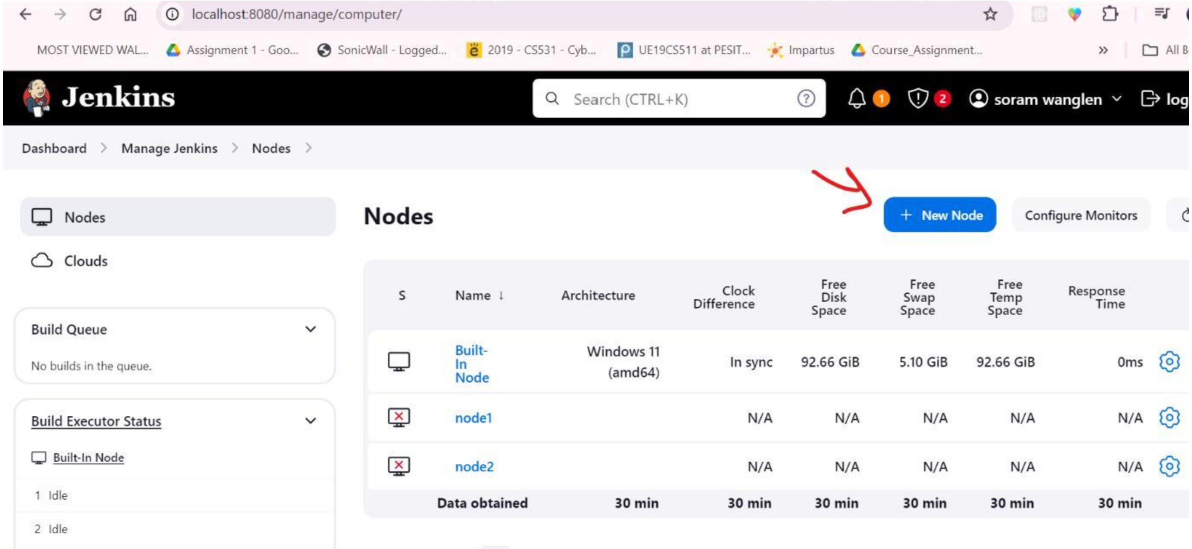
Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Install as Windows Service
Installs Jenkins as a Windows service to this system, so that Jenkins starts automatically when the machine boots.

Clouds
Add, remove, and configure cloud instances to provision agents on-demand.

Appearance

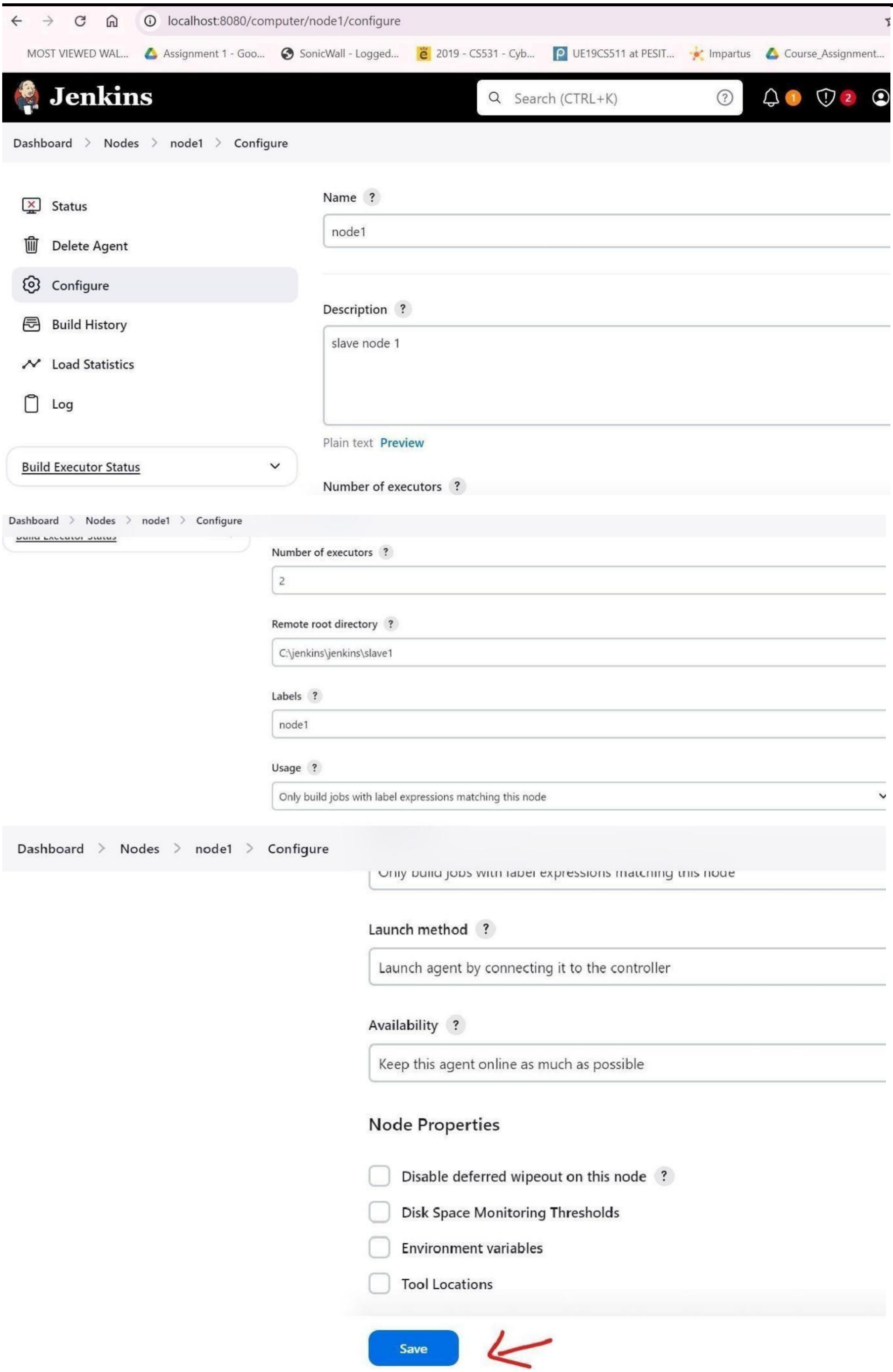
Managed files



Nodes

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Windows 11 (amd64)	In sync	92.66 GiB	5.10 GiB	92.66 GiB	0ms
	node1		N/A	N/A	N/A	N/A	N/A
	node2		N/A	N/A	N/A	N/A	N/A
	Data obtained	30 min	30 min	30 min	30 min	30 min	30 min

Step 2: Configure it.


localhost:8080/computer/node1/configure
MOST VIEWED WAL... Assignment 1 - Goo... SonicWall - Logged... 2019 - CS531 - Cyb... UE19CS511 at PESIT... Impartus Course_Assignment...
Jenkins
Search (CTRL+K)
Dashboard > Nodes > node1 > Configure
Status
Delete Agent
Configure
Build History
Load Statistics
Log
Build Executor Status
Plain text Preview
Node Properties
Save

Name: node1

Description: slave node 1

Number of executors: 2

Remote root directory: C:\jenkins\jenkins\slave1

Labels: node1

Usage: Only build jobs with label expressions matching this node

Launch method: Launch agent by connecting it to the controller

Availability: Keep this agent online as much as possible

Node Properties

□ Disable deferred wipeout on this node

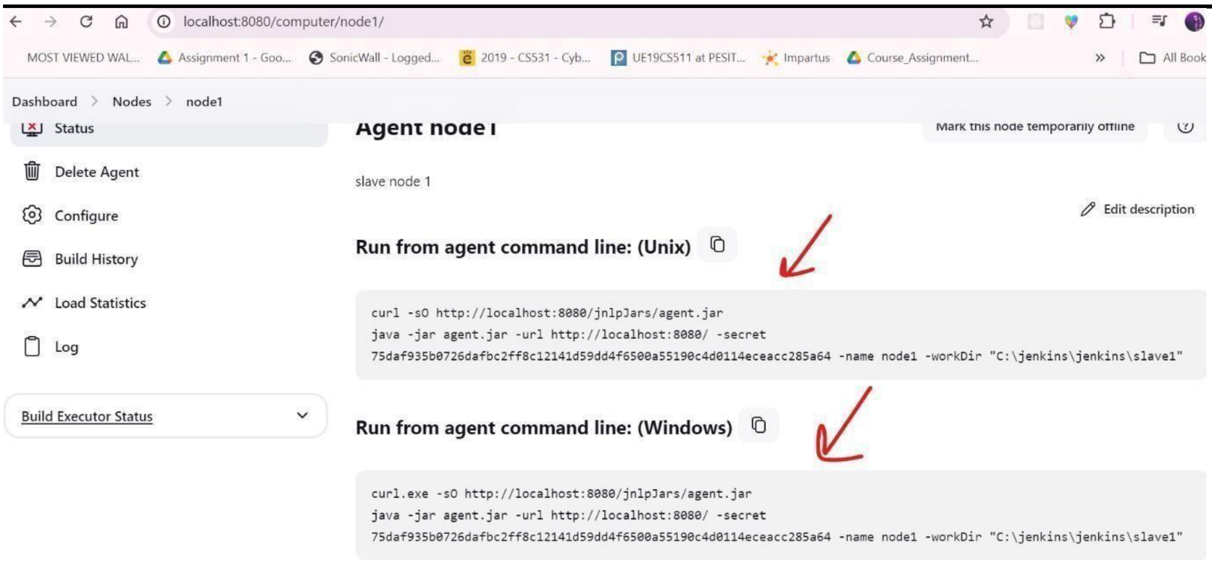
□ Disk Space Monitoring Thresholds

□ Environment variables

□ Tool Locations

Step 3: Similarly create another agent

Step 4: Start your Jenkins agent.



Agent node1

slave node 1

Run from agent command line: (Unix)

```
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://localhost:8080/ -secret
75daf935b0726dafbc2ff8c12141d59dd4f6500a55190c4d0114eceacc285a64 -name node1 -workDir "C:\jenkins\jenkins\slave1"
```

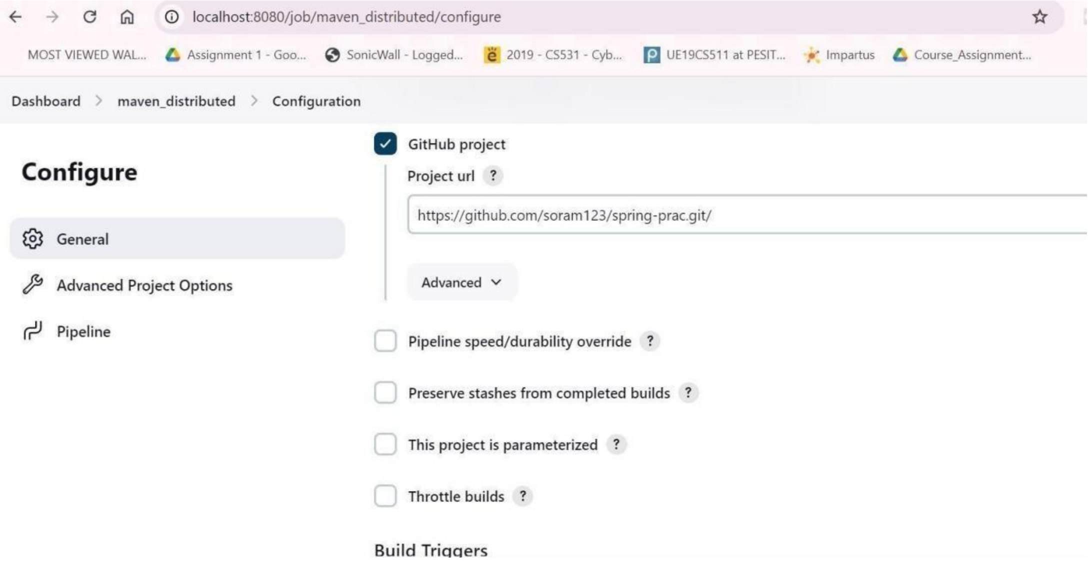
Run from agent command line: (Windows)

```
curl.exe -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://localhost:8080/ -secret
75daf935b0726dafbc2ff8c12141d59dd4f6500a55190c4d0114eceacc285a64 -name node1 -workDir "C:\jenkins\jenkins\slave1"
```

Run the respective commands in linux or windows accordingly.

Note : after starting the agents it should not be idle.

Step 5: Create a new pipeline job and configure it.



☑ GitHub project

Project url ?

https://github.com/soram123/spring-prac.git/

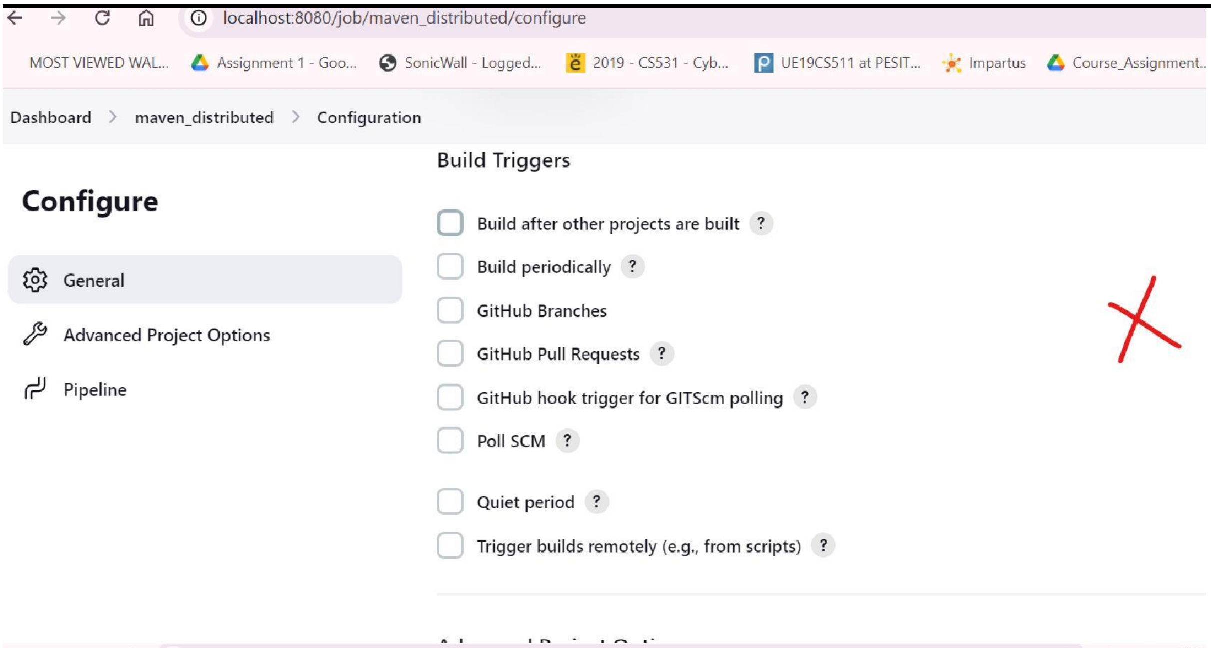
□ Pipeline speed/durability override ?

□ Preserve stashes from completed builds ?

□ This project is parameterized ?

□ Throttle builds ?

Build Triggers



Build Triggers

□ Build after other projects are built

□ Build periodically

□ GitHub Branches

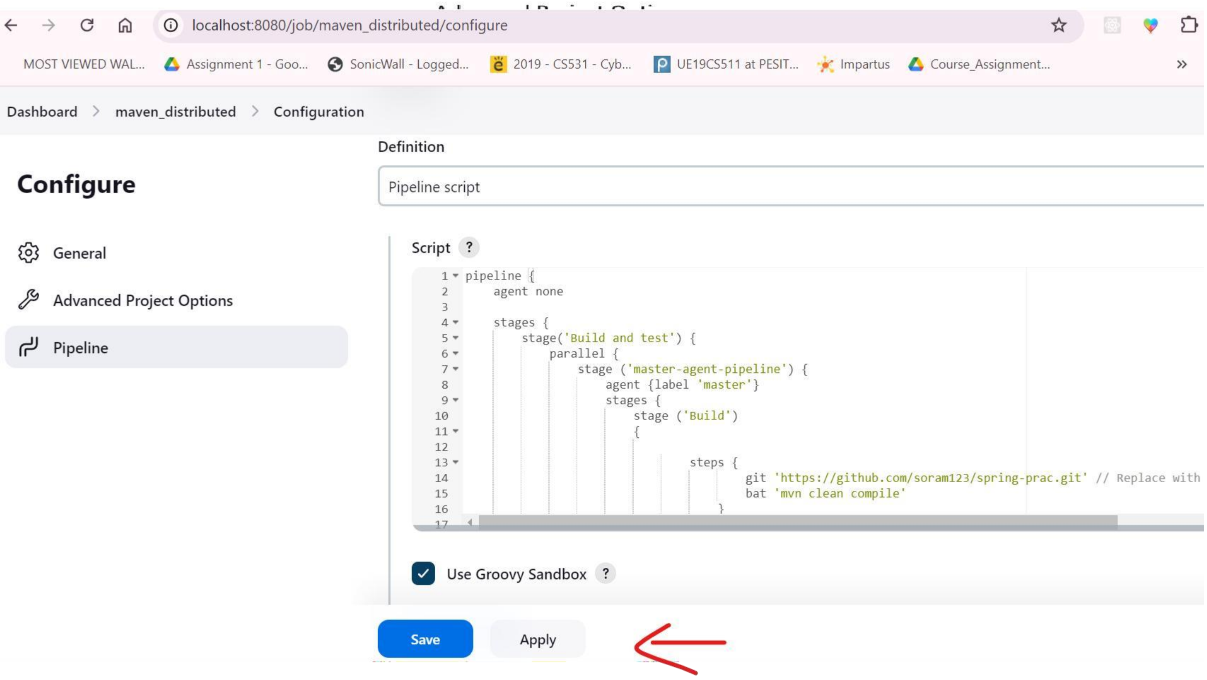
□ GitHub Pull Requests

□ GitHub hook trigger for GITScm polling

□ Poll SCM

□ Quiet period

□ Trigger builds remotely (e.g., from scripts)



Definition: Pipeline script

Script

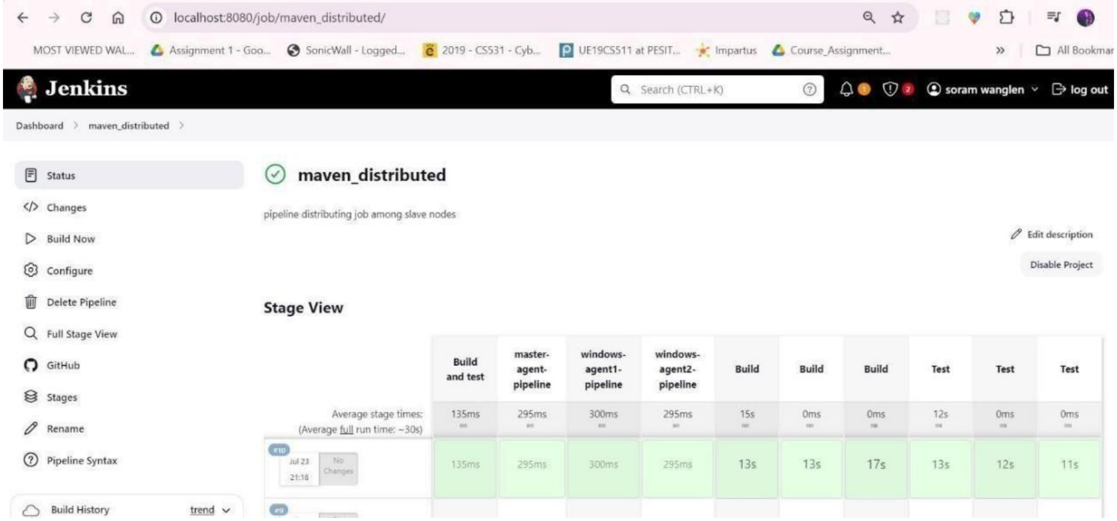
```
pipeline {
    agent none

    stages {
        stage('Build and test') {
            parallel {
                stage ('master-agent-pipeline') {
                    agent {label 'master'}
                    stages {
                        stage ('Build')
                        {

                                steps {
                                    git 'https://github.com/soram123/spring-prac.git' // Replace with
                                    bat 'mvn clean compile'
                                }
```

☑ Use Groovy Sandbox

Here is the whole pipeline script:

```
pipeline
  {agent
  none

  stages {
     stage('Build and test')
       {parallel {
          stage ('master-agent-pipeline')
            {agent {label 'master'}
            stages {
               stage ('Build')
               {
                           steps {git 'https://github.com/soram123/spring-prac.git' // Replace with your Git
                           repository
URL
                       bat 'mvn clean compile'
                    }

              }
              stage ('Test')
                { steps {
                    git 'https://github.com/soram123/spring-prac.git'
                     bat 'mvn test'
                }
              }
           }
        }
        stage('windows-agent1-
           pipeline'){ agent {label 'node1'}
           stages
              { stage('Build')
              {
                 steps {
URL                  git 'https://github.com/soram123/spring-prac.git' // Replace with your Git repository

                       bat 'mvn clean compile'
                 }
              }
              stage('Test'){
                  steps {
                    git 'https://github.com/soram123/spring-prac.git'
                     bat 'mvn test'
                 }
              }
           }

        }
         stage('windows-agent2-
           pipeline'){ agent {label 'node2'}
           stages
              { stage('Build')
              {
                 steps {
URL                  git 'https://github.com/soram123/spring-prac.git' // Replace with your Git repository

                       bat 'mvn clean compile'
                 }
              }
              stage('Test'){
                  steps {
                    git 'https://github.com/soram123/spring-prac.git'
                     bat 'mvn test'
                 }
              }
           }}
        }
    }
}}
```

Step 6: Now build it.



localhost:8080/job/maven_distributed/

maven_distributed

pipeline distributing job among slave nodes

Stage View

	Build and test	master-agent-pipeline	windows-agent1-pipeline	windows-agent2-pipeline	Build	Build	Build	Test	Test	Test
Average stage times: (Average full run time: ~30s)	135ms	295ms	300ms	295ms	15s	0ms	0ms	12s	0ms	0ms
#10 Jul 23 21:18 No Changes	135ms	295ms	300ms	295ms	13s	13s	17s	13s	12s	11s
#9										