

## Deployment 6

### Master ec2 Configurations:

1. Create ec2 master with ports 22, 80, 8080
2. Download plugins:
  - a. Amazon EC2
  - b. Nodejs
  - c. Maven Integration
    - i. Had errors downloading it. Jenkins would freeze up so through the terminal, I installed "sudo yum install maven"

### Agent EC2:

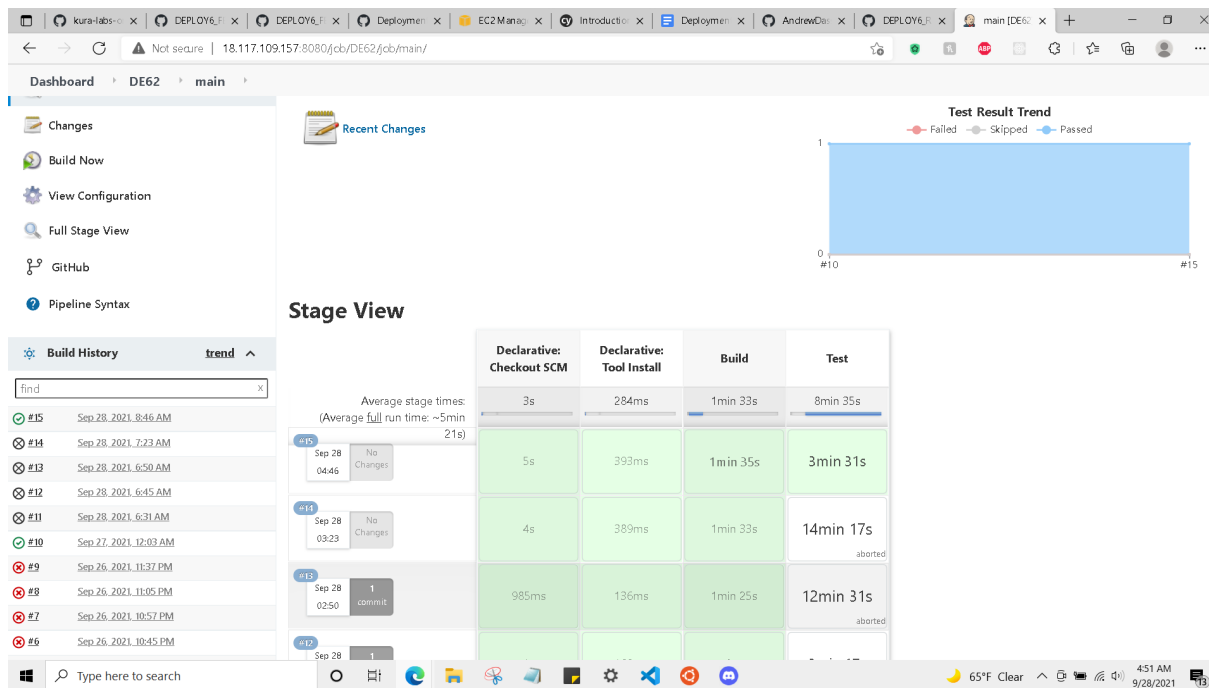
3. Auto Assign IP: Disable
4. Log in into both agents using this format: `ssh -i ".pem" ubuntu@Private IPv4 of agent`
5. Log into Jenkins through the master
6. Configure two agents
  - a. Build Executor Status
  - b. New Node
  - c. Name and select permanent agent
  - d. Name and description
  - e. 2 for executors
  - f. Put home directory / jenkins for both agents. You can use `pwd` to find the directory of the ubuntu agents it being:  
  
/home/ubuntu  
  
Then add jenkins to the end, so in the box enter: "/home/ubuntu/jenkins"
  - g. Write a label for each agent
  - h. Node as much as possible
  - i. Agent via ssh
  - j. Enter agent's ip address (update this every time)
  - k. Select ssh credentials
    - i. username: ubuntu
    - ii. key used to log in
  - l. \*OPTIONAL: If an agent needs to update java, then the following steps are needed:
    - i. `sudo apt-get update`
    - ii. `sudo apt-get install default-jre`
    - iii. `sudo apt-get install default-jdk`
    - iv. Check your java version with the command: "java -version" You should have and be using java 11
7. Create a brand new repository with the files that are in the "kura\_test\_repo" from the ["DEPLOY6\\_FE/kura\\_test\\_repo at main · kura-labs-org/DEPLOY6\\_FE \(github.com\)"](https://github.com/kura-labs-org/DEPLOY6_FE/tree/main/kura_test_repo) repository
8. Connect the repository onto a new multi pipeline Jenkins project
  - a. Select multi pipeline
  - b. Select Github

- c. In the Github Credentials section, add your Github account's username and the url of the Github repository
  - d. Under Build Configuration:
    - i. In the Mode box write "by Jenkinsfile"
    - ii. In the Script path box write "Jenkinsfile"
9. Edit the Jenkinsfile:
- a. Include the line: "tools {nodejs "node"}"
  - b. For agent { label: "insert agent 1 label name"}, enter your label name for your first agent
  - c. Add another stage to perform a test for the pipeline
  - d. Add another agent {label: "insert agent 2 label name"} and enter agent 2's label name
  - e. It can look like this down below:

```
39 lines (36 sloc) | 650 Bytes
Raw Blame

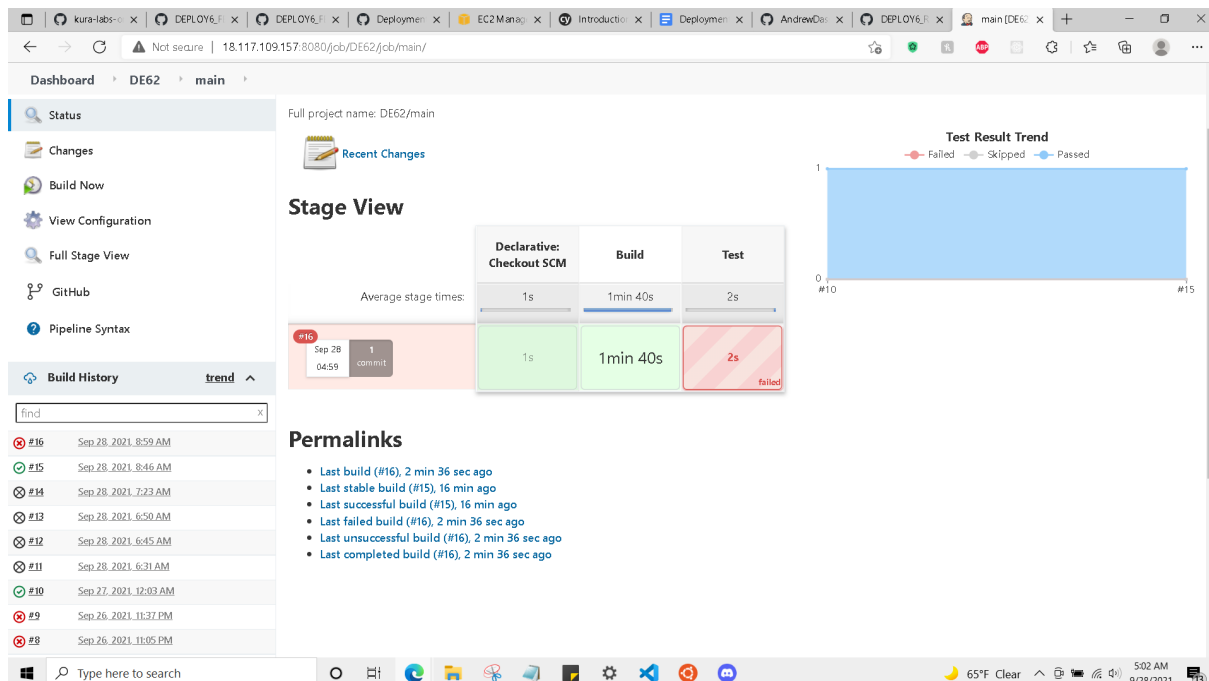
1 pipeline {
2
3   tools {nodejs "node"}
4
5   agent {
6     label 'a1'
7   }
8   stages {
9     stage ('Build') {
10      steps {
11        sh 'rm -rf ./kuna_test_repo/cypress2'
12        sh '''
13          npm install
14          npm run build
15          sudo npm install -g serve
16          serve -s build &
17          ...
18        '''
19      }
20    }
21    stage ('Test') {
22      agent {
23        label 'a2'
24      }
25      steps {
26        sh '''
27          npm install cypress
28          npm install mocha
29          npx cypress run --spec ./cypress/integration/test.spec.js
30          ...
31        '''
32      }
33      post {
34        always {
35          junit 'results/cypress-report.xml'
36        }
37      }
38    }
39  }
```

10. Edit the test.spec.js file located in the cypress/integration/test.spec.js directory.  
Replace the ip with your agent 1 private ip address.
11. Go to jenkins and run the the build



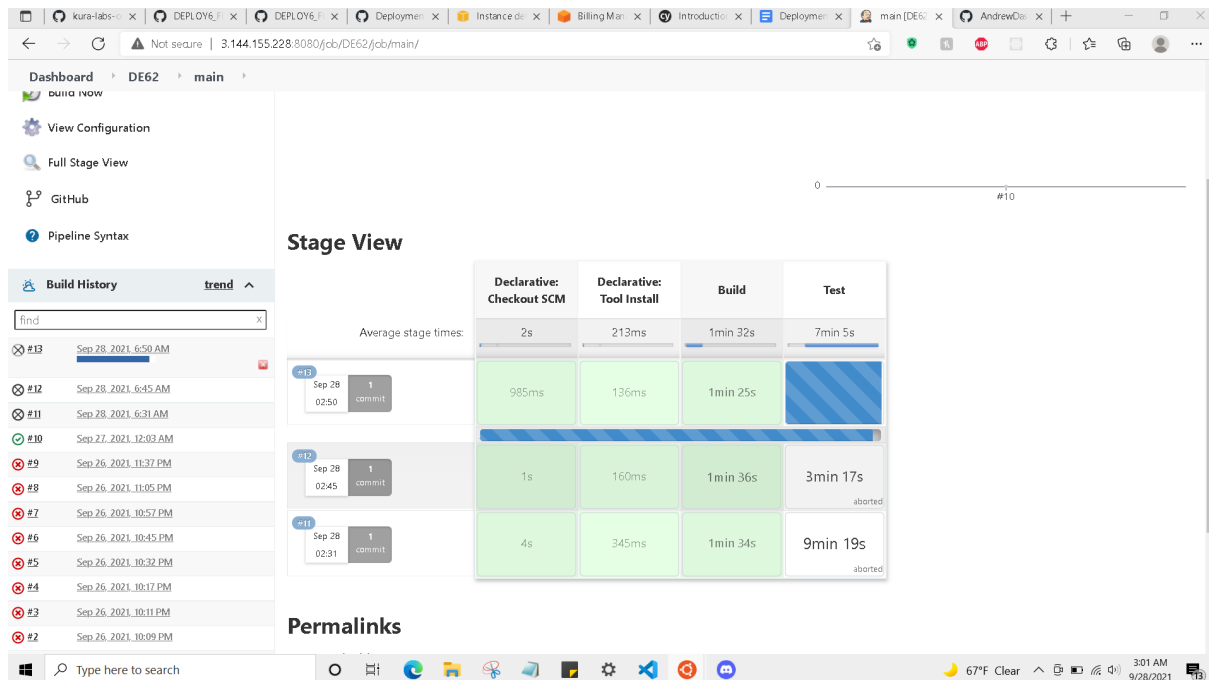
## Breaking the Build:

From modifying the “Jenkinsfile” in my Github repository that holds this project, I removed the line “tools {nodejs "node"}” from the script and it resulted in an error for the “Test” stage. Both agents need the node.js package to run the build. Only the test stage was not successful since without importing node.js, it cannot retrieve the file.



## Miscellaneous Issues, Agent 2 Having an Unsuccessful Connection:

When rerunning the build I came across the issue the Test stage would go infinitely:



After visiting the log page to see the issue, Agent 2 was not successfully connected online therefore I tried to troubleshoot and reconnect it back online, however it was always stuck on:

SSH Connection failed with IOException: "The kexTimeout (65000 ms) expired.", retrying in 15 seconds. There are 10 more retries left.

Once after waiting a long time, the SSH connection was successful and said the message "[SSH] Authentication successful." However, on the log page, it would still have the loading symbol even after 10 minutes of waiting and when going back to the "Manage Nodes and Clouds page", agent 2 would still have a red X since it is not connected to the master ec2.

Since I could not find a solution to this problem, I deleted and recreated the agent and installed the needed packages to make the build run again.