

# Deployment 6

Kenneth Tan

## Goal of Deployment:

- Have master ec2 orchestrate building and testing in different agents
- Testing would be done on the “testing” agent using cypress
- Have one successful test and one failed test

## Requirements for this Deployment

### EC2

- Master (jenkins)
  - Hosts jenkins to run. This will orchestrate which tasks are sent to which agent
- Agent (build)
  - I allowed for inbound ports 22 and 5000 traffic. Port 22 allows for me to SSH into the agent and port 5000 publishes the application on this port
  - To install the necessary packages I used the following bash script:

```
#!/bin/bash
Sudo apt-get updates
Sudo apt-get upgrade -y
Sudo apt-get install default-jre git nodejs npm -y
```
- Agent (test)
  - I allowed for inbound port 22 traffic
  - To install the necessary packages I used the following bash script:

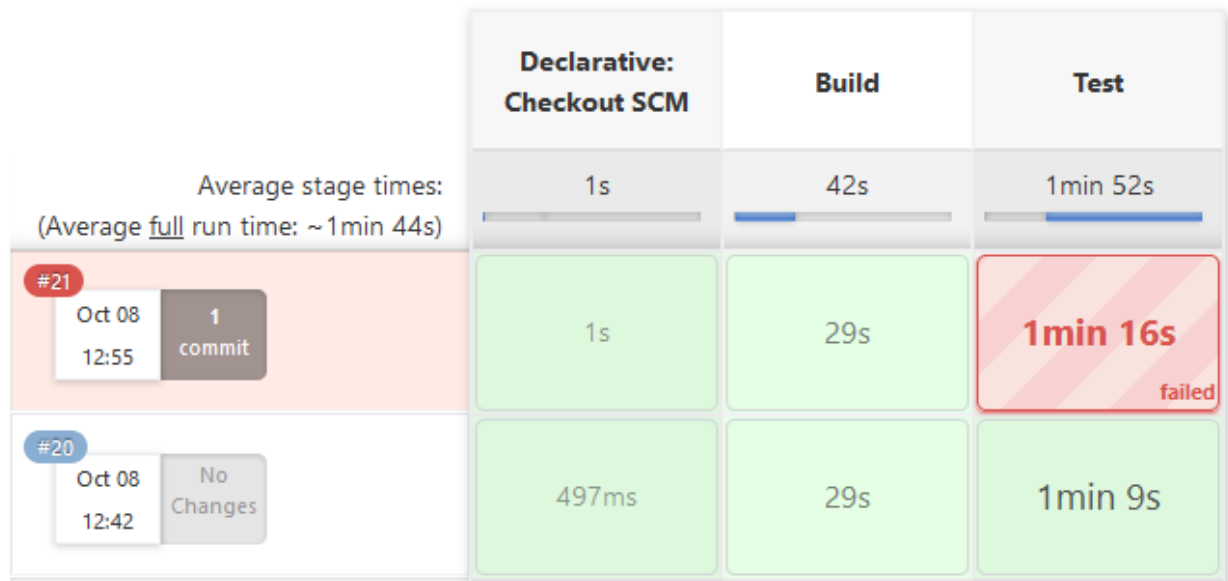
```
#!/bin/bash
Sudo apt-get updates
Sudo apt-get upgrade -y
Sudo apt-get install default-jre git nodejs npm maven libgtk2.0-0 libgtk-3-0
libgbm-dev libnotify-dev libgconf-2-4 libnss3 ,libxss1 libasound2 libxtst6
xauth xvfb -y
```

# Jenkins pipeline script

```
pipeline {
  agent {
    label 'build'
  }
  stages {
    stage ('Build') {
      steps {
        sh 'rm -rf ./cypress'
        sh '''
            npm install
            npm run build
            sudo npm install -g serve
            serve -s build &
        '''
      }
    }
    stage ('Test') {
      agent {
        label 'test'
      }
      steps {
        sh '''
            npm install cypress
            npm install mocha
            npx cypress run --spec ./cypress/integration/test.spec.js
        '''
      }
    }
    post {
      always {
        junit 'results/cypress-report.xml'
      }
    }
  }
}
```

## Screenshots

### Stage View



The passed test indicated that the test case that was set in the cypress test case matched the value or behavior of the part of the application that was tested. In this case it was for the header in example 1 to read "My Awesome Web Application"

The failed test indicates that the expected value for the part that cyprus was testing did not match up or behave as expected so it returned as failed. I had changed the test for example 1 to test for the header to read "My Totally Awesome Web Application!"

## Cypress files

There are video files that show the tests in cypress stored in the test agent directory `./cypress/integration/test.spec.js.mp4` . To extract these files, I moved this to my home directory. This is because that is where my `.pem` file is located to SSH in and out of the agent. I then use the following syntax:

```
scp -i ubuntu.pem <source/file to be moved> <target login>:<target directory>
scp -i ubuntu.pem test.spec.js.mp4 ubuntu@<private ipv4>: ./home
```

This command moves the `test.spec.js/mp4` video file into my master ec2. To extract this file from my master onto my local windows system, I used the follow command in my command prompt terminal:

```
scp -i ubuntu-agents.pem ec2-user@ec2-<private
ipv4>.compute-1.amazonaws.com:test.spec.js.mp4 ./Users/<UserName>/Desktop
```

This moves the file onto my local desktop. Both video files are available at [https://github.com/KennethT404/DEPLOY6\\_FE](https://github.com/KennethT404/DEPLOY6_FE)

## Challenges

The big challenge I had completing this deployment was the naming convention of the credential on Jenkins. Originally I had named the credentials to verify my access to the agent the same name as I had given my .pem file on my local machine. This gave me authorization errors as Jenkins did not recognize my key as the appropriate key to use. Despite my remaking of the agent on an EC2 level and recreation of the credentials in Jenkins it continued to give me errors. Soon I found out it was attempting to use credentials from a file named ubuntu so changing my credential username to ubuntu, I was able to successfully connect. From there I was able to both build and test on the separate agents.