Jenkins

Objectives:

- Install Jenkins
- Forking a github repo
- Configuring a build job in Jenkins

Description:

In this assignment you will begin by installing jenkins. You will then utilize jenkins to deploy the java application from the last project that we did. Jenkins will allow us to explore the idea of Continuous Integration. You should use a different vm than the production vm you created in the last assignment, although you will want to redo all the stuff you did in that assignment on the new machine. Also, don’t forget to do the `wget` from the previous assignment to fix the java bug.

Requirements:

Please **READ** all of the requirements first **BEFORE** doing any configuration to your vm.

You will be following some of chapter2 of the text. You should complete all the steps up through ‘Configuring a build job’ *with a few exceptions*. Here is an example as to how I did the project (after installing the VM and configuring Jenkins).

**Here** is an example as to how to test your webhooks.

You will complete the assignment using an AWS t2.medium instance. If you use a t2.micro instance, you won’t have enough space and you will cry because you will have to start the project over on a medium instance. After you have passed off the assignment, I would shut it down (don’t delete it).

First, see below as to how to install Jenkins, the text instructions did not work. I notice that there are some tweaks that need to be made:

**To install Jenkins**

```bash
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
sudo apt update
sudo apt install jenkins
```

Immediately after installing, you should be able to visit this host port 8080 in the browser and continue as described in chapter 2 (Copy password from admin file). Install suggested plugins. (Don’t forget that you may need to open port 8080 in the security group)

**Errata**

- When configuring Java within Jenkins, I first did a `sudo update-alternatives --display java` to see what java versions I had available on my Jenkins machine. (remember you have to set this vm up like your initial production one). Mine showed that my current java version was `/usr/lib/jvm/java-11-openjdk-amd64/bin/java`. In the global tool configuration, I then added a JDK installation, gave it a name of `java-11` and listed the JAVA_HOME as `/usr/lib/jvm/java-11-openjdk-amd64`.

- When configuring Maven, you should click to Install automatically.

- Don’t fork their repository, fork mine (if you haven’t already). Any other screens that ask for a spring-clinic git repo, you should use yours.

- Under **Build Triggers** the book mentions something about building when a change is pushed to GitHub. In the new version of Jenkins, it is GitHub hook trigger for GITScm polling. Make sure to select it.

**Check off procedure:**
This will be passed off in class. You should have your browser open to jenkins and demonstrate you have a successful build. You should also demonstrate that after making a change and doing a push to your github account that another build starts.