Andrew Dass Deployment #2 Guide

Steps:

1. Create a Amazon EC2 with the following steps:
    a. Make an AWS account, log in
    b. Go to AWS Management Console
    c. Click on Launch virtual machine
    d. Click on Amazon AMI (HVM), SSD Volume Type. Under the blue select button, choose the right architecture that matches your computer (I chose 64-bit (x86))
    e. Select an instance type (choose t2.micro for no charges)
    f. For configuring security groups, add the following 3 types with these modifications:

Type	Protocol	Port Range	Source	Description (Optional)
SSH	TCP	22	My IP	
Custom TCP Rule	TCP	8080	Custom	
HTTP	TCP	80	Custom	

    g. Press the Blue Launch button
    h. Create a new key pair, name and download the key pair
    i. Launch Instances
    j. See if the instance is running,
    k. *Optional* - hover over it to rename it if you like
2. Open up Linux Ubuntu's Terminal
3. cd to where you kept the downloaded .pem file for the instance you created. If you haven't moved your .pem file, it should be in Downloads. Its name is the name you gave it before downloading the file. If you can't find your pem file, delete this instance and make a new instance, since you cannot redownload pem files.
4. Once you found your pem file and cd into the directory where it is, enter the following into the terminal:
    a. sudo ssh -i "FileName".pem ec2-user@publicIPv4address, then press enter

* File name = The actual name of your pem file
* Public IPv4 address is found when you click on your instance on the AWS EC2 website

5. The following message should appear:

Warning: Identity file "FileName".pem not accessible: No such file or directory.
The authenticity of host 'Public IPv4 address (Public IPv4 address)' can't be established.
ECDSA key fingerprint is ----------.
Are you sure you want to continue connecting (yes/no/[fingerprint])?

6. Type in yes
7. After pressing yes, you should receive this message showing you are in the EC2 terminal:

__| __|_)
        _| (/   Amazon Linux 2 AMI
        ___|___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-”Private IPv4 address” ~]$

8. In the Amazon EC2, we must download Jenkins. To do so, the following steps are required and to be runned in this order:
    a. sudo yum update -y
    b. sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
    c. sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
    d. sudo yum upgrade
    e. sudo yum install jenkins java-1.8.0-openjdk-devel -y
    f. sudo systemctl daemon-reload
    g. sudo systemctl start jenkins
9. After the In your internet browser, (Chrome or Mozilla is preferred), paste the following into the url bar:
        PublicIPv4address:8080

* Remember the Public IPv4 address is available under the Amazon AWS EC2 instance information

10. You should see you are redirected to a new website where it says “Unlock Jenkins”
11. In the terminal, type the following:
    a. sudo cat /var/lib/jenkins/secrets/initialAdminPassword
    b. You should receive a password from the terminal, copy and paste that into the box on the jenkins website allowing you access
12. Click on “Install Suggested Plugins”
13. Create a username, password, and register with an email you use
14. Click save and finish
15. Click start using Jenkins
16. Click on new item - select pipeline and name it for your reference
17. Under ‘Build Triggers’ section, check mark ‘Build periodically’
18. In the box that appears, type */10 * * * *. This executes the script every 10 minutes as long as the EC2 is on and running. If it is stopped, the script won’t run every 10 minutes.
19. Under pipeline, select pipeline script
20. A pipeline script that appears should say:

```
pipeline {
        agent any

        stages {
        stage('Hello') {
        steps {
```

```
                echo 'Hello World'
        }
        }
        }
}
```

Add the following line under agent any:

```
triggers{
cron('*/10 * * * *')
}
```

The new code should like this:

```
pipeline {
        agent any
        triggers {
        cron('*/10 * * * *')
        }
        stages {
        stage('Hello') {
        steps {
                echo 'Hello World'
        }
        }
        }
}
```

21. Save changes
22. To stop Jenkins, use the following command:

```
sudo systemctl stop jenkins.service
```

23. While logged in into your AWS EC2, run the command:

```
sudo poweroff
```

*This exits out of your aws account and also stops the ec2 from running

24. To shut down the ec2 at a certain time, type :one of the commands into the terminal:

```
sudo shutdown +..
```

where .. = number. This specifies the number of minutes that will elapse and then the ec2 will shut down. Can manually calculate when it will be 9:00 pm to shut it down

or

```
sudo shutdown 0 21 * * *
```

The example above shows it will shutdown the ec2 by 9:00 pm. Use 00:00 - 23:59 to shutdown the ec2 when desired

Do not do sudo shutdown 21:00 because the EC2 or terminal will shut down on the next day and there is no way of changing this setting to shut it down aside from the next day