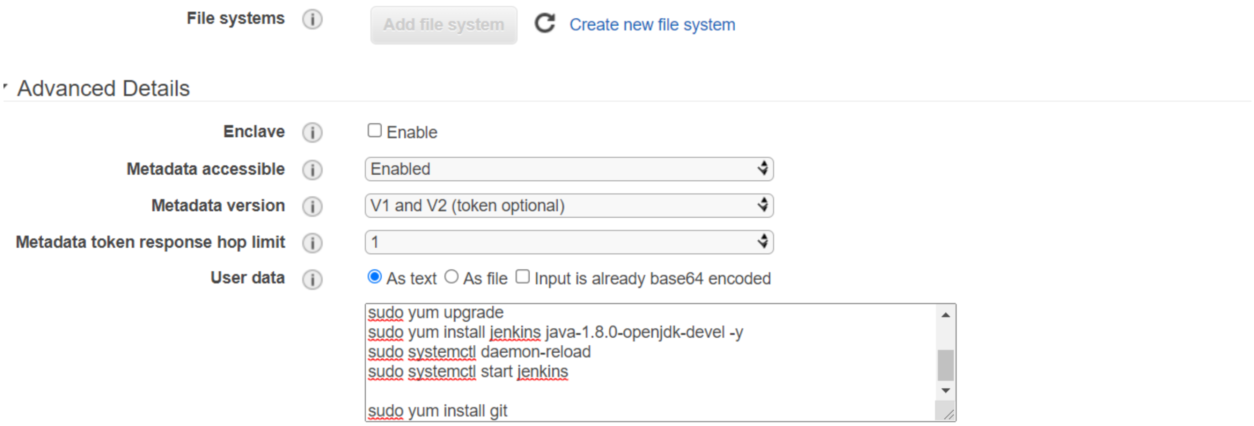
1. Create a new EC2 Amazon Linux instance using the following bootstrap script as shown below.

```
#!/bin/bash
sudo yum update -y
sudo wget -O /etc/yum.repos.d/jenkins.repo \https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
sudo yum upgrade
sudo yum install jenkins java-1.8.0-openjdk-devel -y
sudo systemctl daemon-reload
sudo systemctl start Jenkins

sudo yum install git
```



Advanced Details

Enclave □ Enable

Metadata accessible: Enabled

Metadata version: V1 and V2 (token optional)

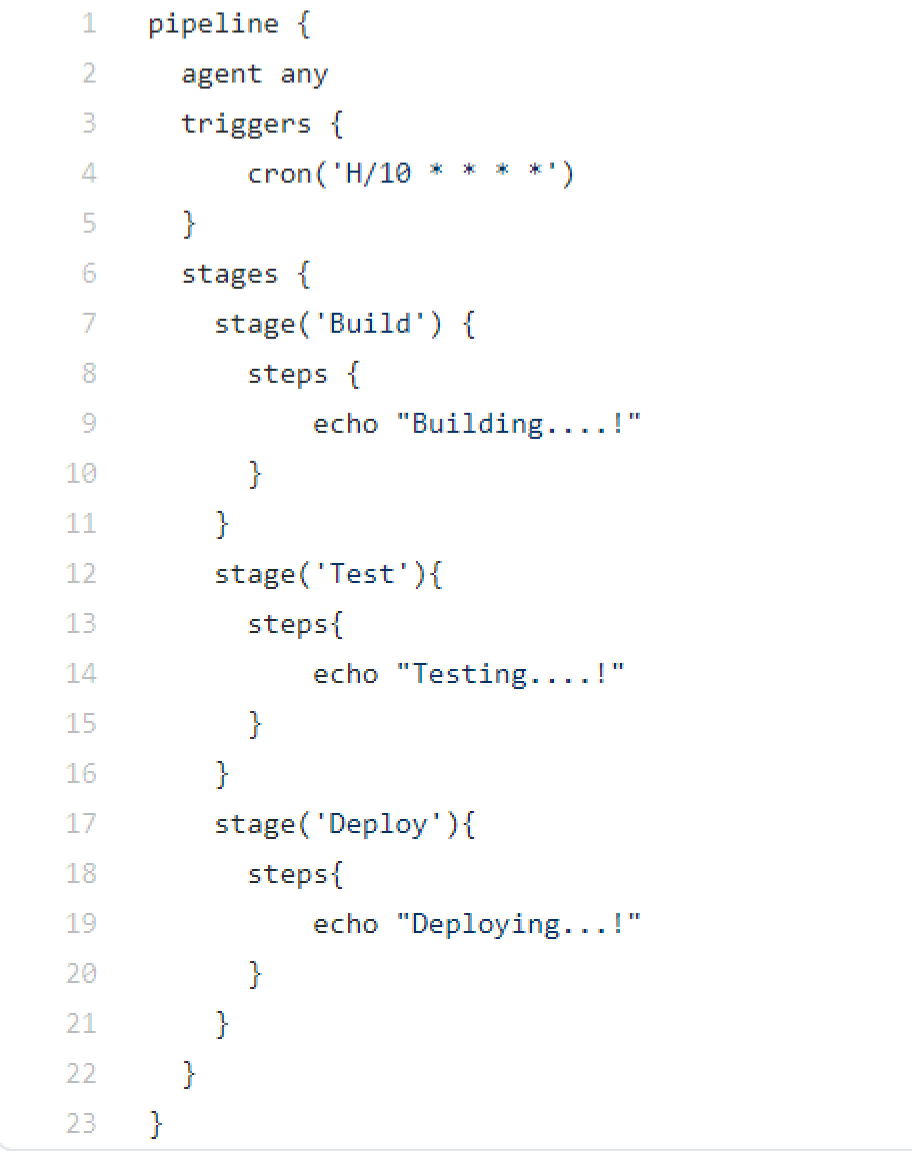
Metadata token response hop limit: 1

User data ● As text ○ As file □ Input is already base64 encoded

```
sudo yum upgrade
sudo yum install jenkins java-1.8.0-openjdk-devel -y
sudo systemctl daemon-reload
sudo systemctl start jenkins

sudo yum install git
```

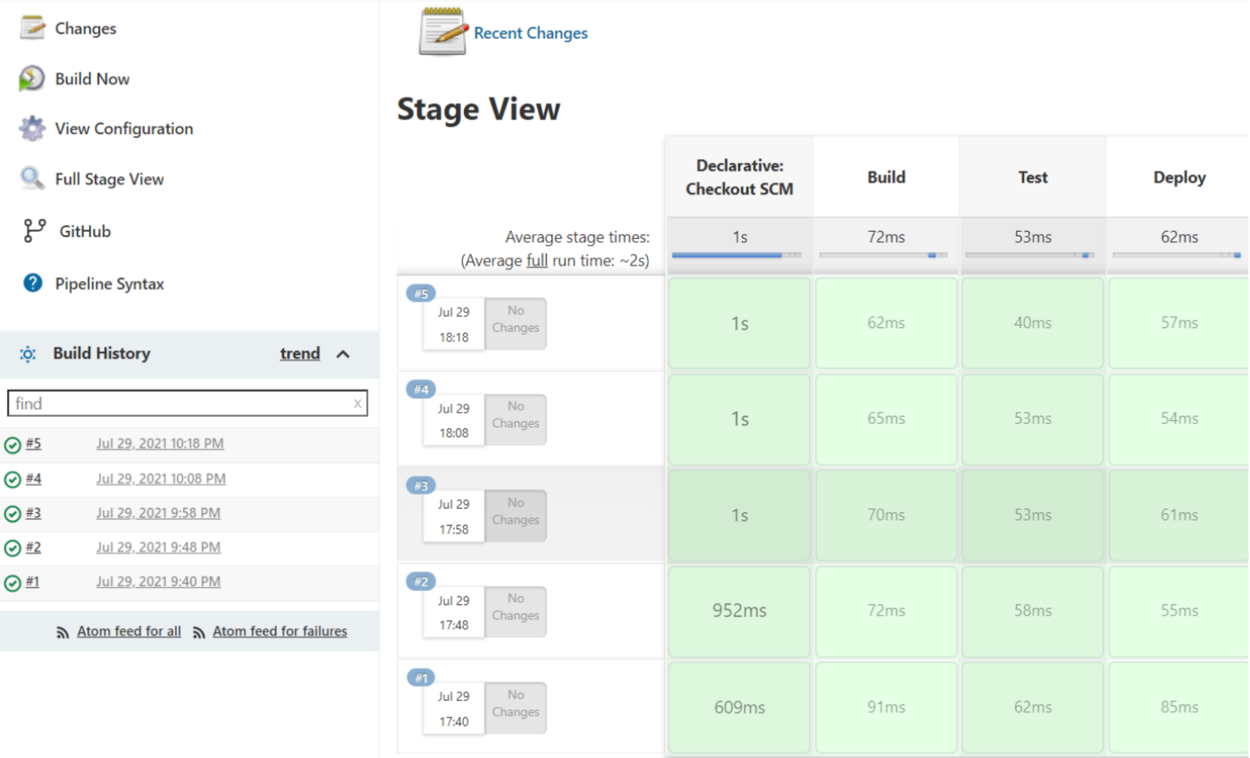
2. For security groups settings, set SSH port of 22 (My ip), a Custom TCP Rule of 8080 and HTTP port of 80.
3. Then launch your EC2 instance
4. After launching ssh into your instance, go to your terminal and enter the directory which contains your SSH keys and do the following:
   cd (directory where your key is located)
   ssh -i (EC2 keys) ec2-user@(public IPV4 address of your instance)
5. Check that Jenkins is running by using the command sudo systemctl status Jenkins.
6. Once Jenkins is running, enter your browser and enter the public IPV4 address of your instance with port 8080. Example 3.54.67.256:8080
7. Then after Jenkins loads, use the command sudo cat /var/lib/jenkins/secrets/initialAdminPassword to get that password you'll need to enter.
8. Paste the password that you got from previous step.
9. Now install suggested plugins for Jenkins
10. Setup up an admin account with your username, password and email.
11. Fork the DEPLOY02_CRON_JOB repository and edit your Jenkins file with the following code to create a pipeline with a build, test and deploy stage as shown below.



```
pipeline {
  agent any
  triggers {
      cron('H/10 * * * *')
  }
  stages {
    stage('Build') {
      steps {
          echo "Building....!"
      }
    }
    stage('Test'){
      steps{
          echo "Testing....!"
      }
    }
    stage('Deploy'){
      steps{
          echo "Deploying...!"
      }
    }
  }
}
```

12. Create a webhook to your github repo for Jenkins to read your Jenkins file.
13. Go to your Github account settings and generate a personal access token, copy the value and save it somewhere safe.
14. Go to your Jenkins instance and head to the dashboard to make a new item.
15. Give it a name then select multibranch pipeline.
16. Then add the branch source as Github then click on add and after click on Jenkins.
17. Add your Github username into the username field, and then enter your personal access token into the password field. For ID field enter Jenkins-webhook-id.
18. Then set credentials next to the add button.
19. Then switch to Repository Scan, there adding your Github account user name as owner and selecting the repository needed.
20. Then go to configure to find Scan Repository triggers. There you will click Periodic to set it for 10 minute intervals.

21. Then after building check to make sure build was done successfully as shown below.



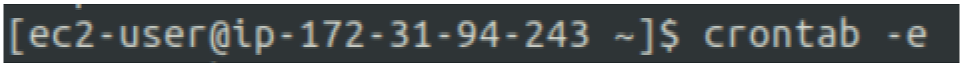
Build History

- #5 Jul 29, 2021 10:18 PM
- #4 Jul 29, 2021 10:08 PM
- #3 Jul 29, 2021 9:58 PM
- #2 Jul 29, 2021 9:48 PM
- #1 Jul 29, 2021 9:40 PM

Stage View

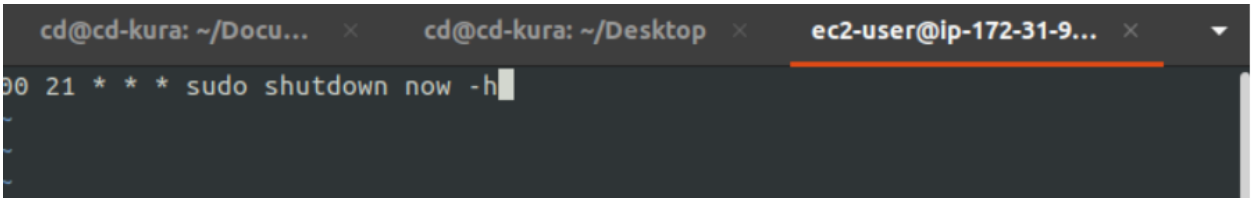
	Declarative: Checkout SCM	Build	Test	Deploy
Average stage times: (Average full run time: ~2s)	1s	72ms	53ms	62ms
#5 Jul 29 18:18 No Changes	1s	62ms	40ms	57ms
#4 Jul 29 18:08 No Changes	1s	65ms	53ms	54ms
#3 Jul 29 17:58 No Changes	1s	70ms	53ms	61ms
#2 Jul 29 17:48 No Changes	952ms	72ms	58ms	55ms
#1 Jul 29 17:40 No Changes	609ms	91ms	62ms	85ms

22. Then to shut down the instance by the end of class I used the command crontab -e. This allowed me to enter the vim editor to update the cron job.



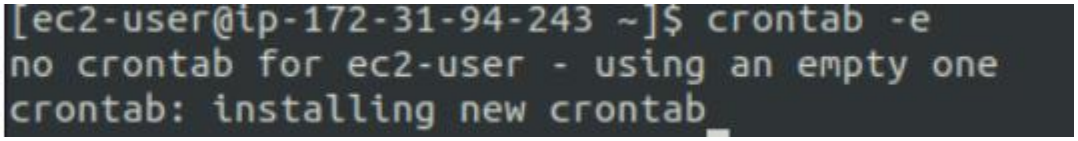
```
[ec2-user@ip-172-31-94-243 ~]$ crontab -e
```

23. Then entered 00 21 * * * sudo shutdown now -h in the vim text editor as shown in the figure below. This allowed me to shut the instance down when it’s 9 pm which is the time class ends.



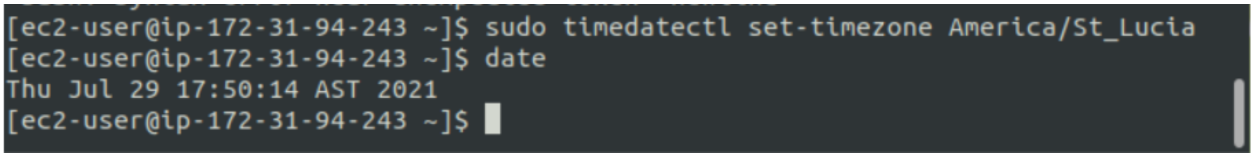
```
00 21 * * * sudo shutdown now -h
~
~
~
```

24. To exit the vim terminal press esc twice then “:wq” to write and quit.
25. As shown below you’ll see that a new crontab was installed.



```
[ec2-user@ip-172-31-94-243 ~]$ crontab -e
no crontab for ec2-user - using an empty one
crontab: installing new crontab
```

26. Then use the command sudo timedatectl set-timezone (your_time_zone) to set the time and the required time zone for you.
27. After check the time and date by using the date command as shown below



```
[ec2-user@ip-172-31-94-243 ~]$ sudo timedatectl set-timezone America/St_Lucia
[ec2-user@ip-172-31-94-243 ~]$ date
Thu Jul 29 17:50:14 AST 2021
[ec2-user@ip-172-31-94-243 ~]$ 
```