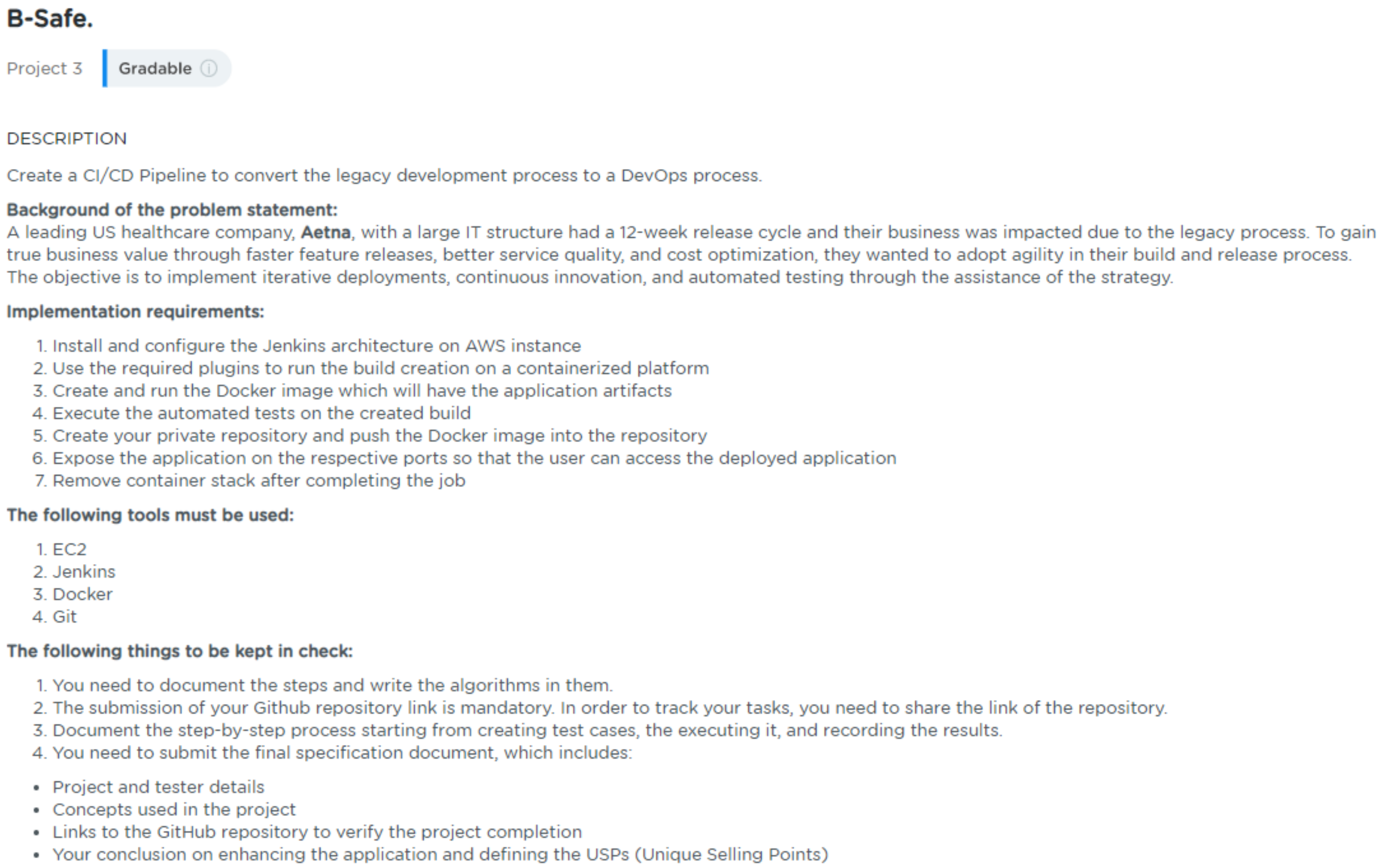
DevOps Capstone Project

By Tamzidul Matin

GitHub Repo: https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project



B-Safe.

Project 3 | **Gradable**

DESCRIPTION

Create a CI/CD Pipeline to convert the legacy development process to a DevOps process.

Background of the problem statement:
A leading US healthcare company, **Aetna**, with a large IT structure had a 12-week release cycle and their business was impacted due to the legacy process. To gain true business value through faster feature releases, better service quality, and cost optimization, they wanted to adopt agility in their build and release process. The objective is to implement iterative deployments, continuous innovation, and automated testing through the assistance of the strategy.

Implementation requirements:

1. Install and configure the Jenkins architecture on AWS instance
2. Use the required plugins to run the build creation on a containerized platform
3. Create and run the Docker image which will have the application artifacts
4. Execute the automated tests on the created build
5. Create your private repository and push the Docker image into the repository
6. Expose the application on the respective ports so that the user can access the deployed application
7. Remove container stack after completing the job

The following tools must be used:

1. EC2
2. Jenkins
3. Docker
4. Git

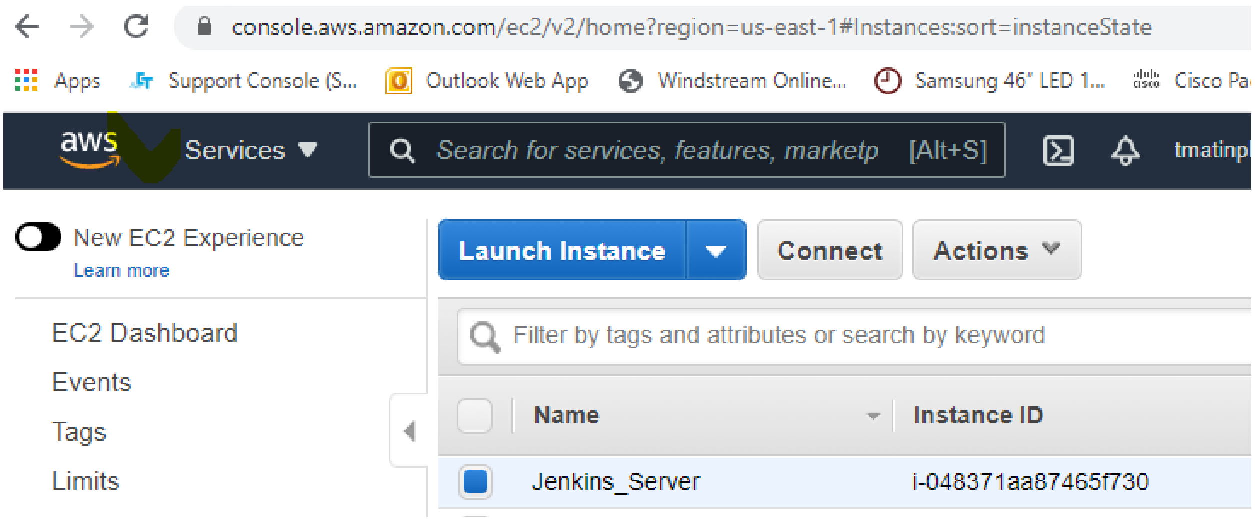
The following things to be kept in check:

1. You need to document the steps and write the algorithms in them.
2. The submission of your Github repository link is mandatory. In order to track your tasks, you need to share the link of the repository.
3. Document the step-by-step process starting from creating test cases, the executing it, and recording the results.
4. You need to submit the final specification document, which includes:

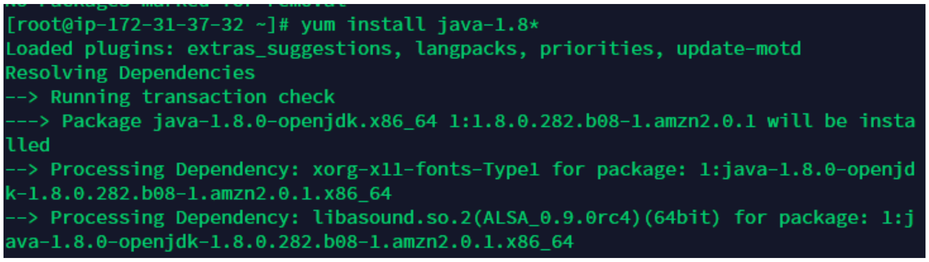
- Project and tester details
- Concepts used in the project
- Links to the GitHub repository to verify the project completion
- Your conclusion on enhancing the application and defining the USPs (Unique Selling Points)

Section I. Jenkins Setup

1. Launch ec2 instance



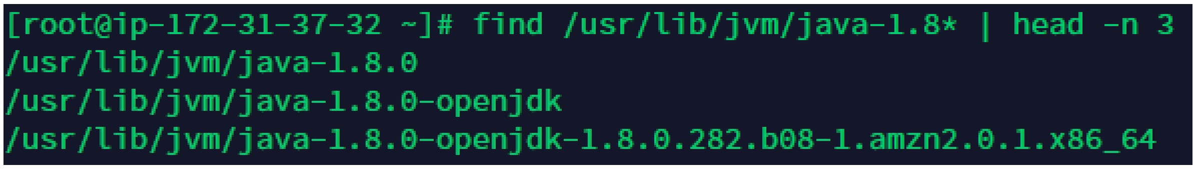
2. Install Java



```
[root@ip-172-31-37-32 ~]# yum install java-1.8*
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package java-1.8.0-openjdk.x86_64 1:1.8.0.282.b08-1.amzn2.0.1 will be insta
lled
--> Processing Dependency: xorg-x11-fonts-Type1 for package: 1:java-1.8.0-openjd
k-1.8.0.282.b08-1.amzn2.0.1.x86_64
--> Processing Dependency: libasound.so.2(ALSA_0.9.0rc4)(64bit) for package: 1:j
ava-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86_64
```

3. Set Java Home Path

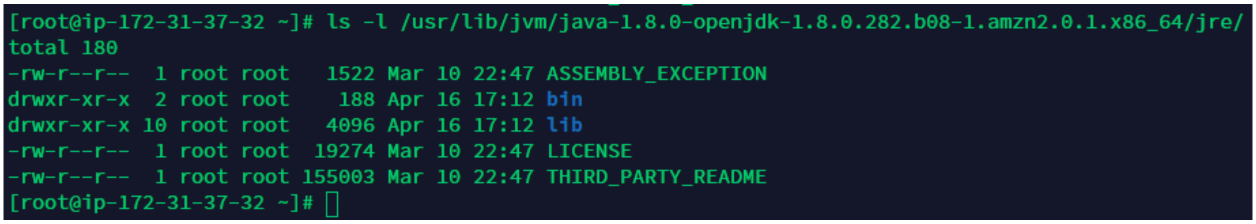
Use the find command: **find /usr/lib/jvm/java-1.8* | head -n 3**



```
[root@ip-172-31-37-32 ~]# find /usr/lib/jvm/java-1.8* | head -n 3
/usr/lib/jvm/java-1.8.0
/usr/lib/jvm/java-1.8.0-openjdk
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86_64
```

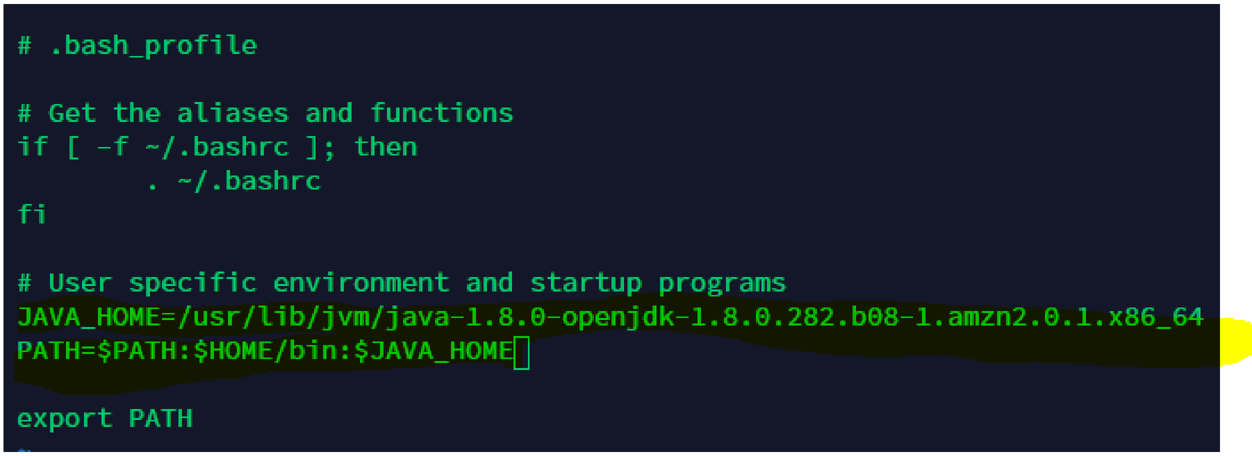
We can find the path by using /jre at the end of this that path

ls -l /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86_64/jre/



```
[root@ip-172-31-37-32 ~]# ls -l /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86_64/jre/
total 180
-rw-r--r--  1 root root   1522 Mar 10 22:47 ASSEMBLY_EXCEPTION
drwxr-xr-x  2 root root    188 Apr 16 17:12 bin
drwxr-xr-x 10 root root   4096 Apr 16 17:12 lib
-rw-r--r--  1 root root  19274 Mar 10 22:47 LICENSE
-rw-r--r--  1 root root 155003 Mar 10 22:47 THIRD_PARTY_README
[root@ip-172-31-37-32 ~]# 
```

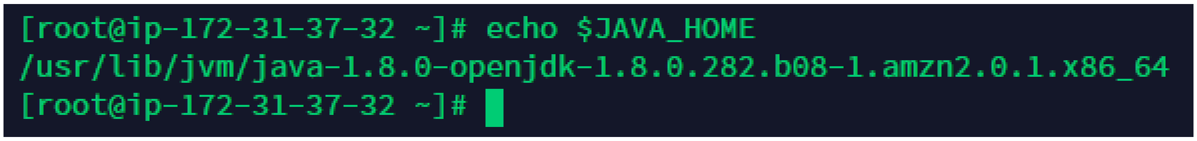
4. Set Java home path to this path by adding the following to the .bash_profile.
   cd ~
   vi .bash_profile



```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs
JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86_64
PATH=$PATH:$HOME/bin:$JAVA_HOME

export PATH
```

Save and exit.

Verify the path using: **echo $JAVA_HOME**



```
[root@ip-172-31-37-32 ~]# echo $JAVA_HOME
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86_64
[root@ip-172-31-37-32 ~]# 
```

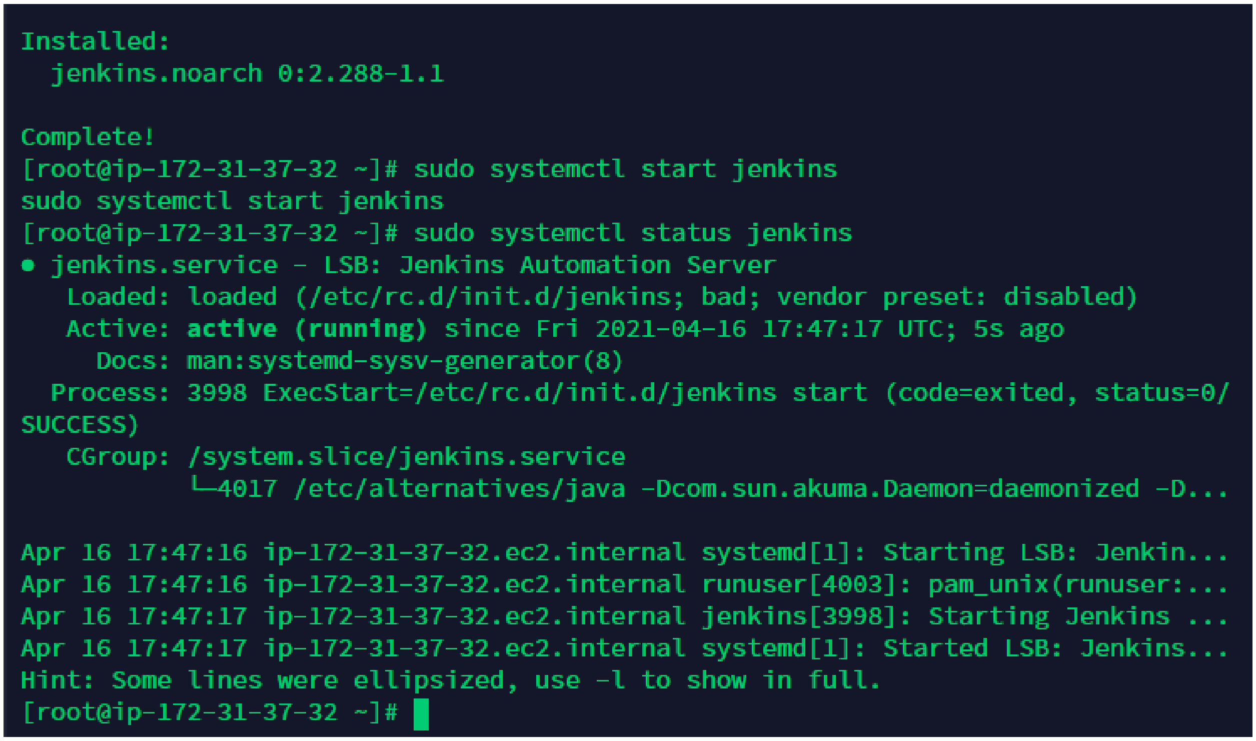
5. Set up jenkins repo and install Jenkins: https://www.jenkins.io/doc/book/installing/linux/

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
    https://pkg.jenkins.io/redhat/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io.key
sudo yum upgrade
sudo yum install jenkins java-1.8.0-openjdk-devel
sudo systemctl daemon-reload
```

Check installation status and start the jenkins service using:

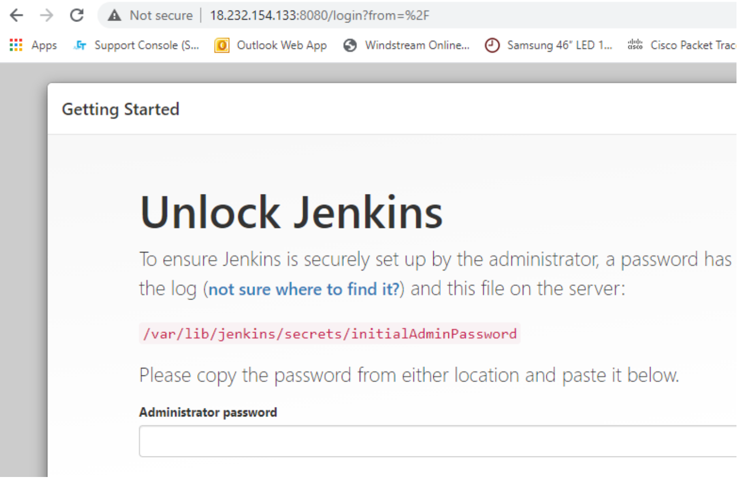
Service jenkins start

Service jenkins status



```
Installed:
  jenkins.noarch 0:2.288-1.1

Complete!
[root@ip-172-31-37-32 ~]# sudo systemctl start jenkins
sudo systemctl start jenkins
[root@ip-172-31-37-32 ~]# sudo systemctl status jenkins
● jenkins.service - LSB: Jenkins Automation Server
   Loaded: loaded (/etc/rc.d/init.d/jenkins; bad; vendor preset: disabled)
   Active: active (running) since Fri 2021-04-16 17:47:17 UTC; 5s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 3998 ExecStart=/etc/rc.d/init.d/jenkins start (code=exited, status=0/
SUCCESS)
   CGroup: /system.slice/jenkins.service
           └─4017 /etc/alternatives/java -Dcom.sun.akuma.Daemon=daemonized -D...

Apr 16 17:47:16 ip-172-31-37-32.ec2.internal systemd[1]: Starting LSB: Jenkin...
Apr 16 17:47:16 ip-172-31-37-32.ec2.internal runuser[4003]: pam_unix(runuser:...
Apr 16 17:47:17 ip-172-31-37-32.ec2.internal jenkins[3998]: Starting Jenkins ...
Apr 16 17:47:17 ip-172-31-37-32.ec2.internal systemd[1]: Started LSB: Jenkins...
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-37-32 ~]#
```

6. **Access the Jenkins server using the public ip and port 8080 of the ec2 instance and the initial login password at the path bellow. Skip plubin installation for now and go to admin configure on the top left corner and change your password by scrolling down to the password section.**



Getting Started

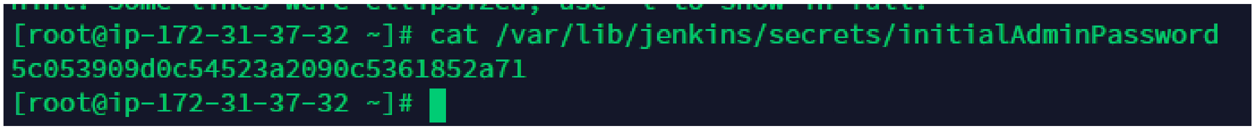
Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has the log (not sure where to find it?) and this file on the server:

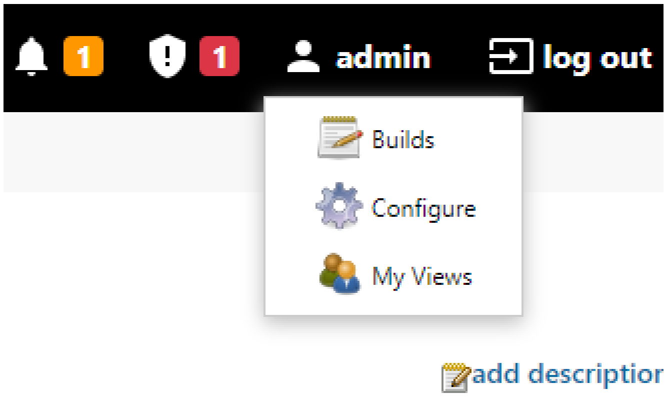
`/var/lib/jenkins/secrets/initialAdminPassword`

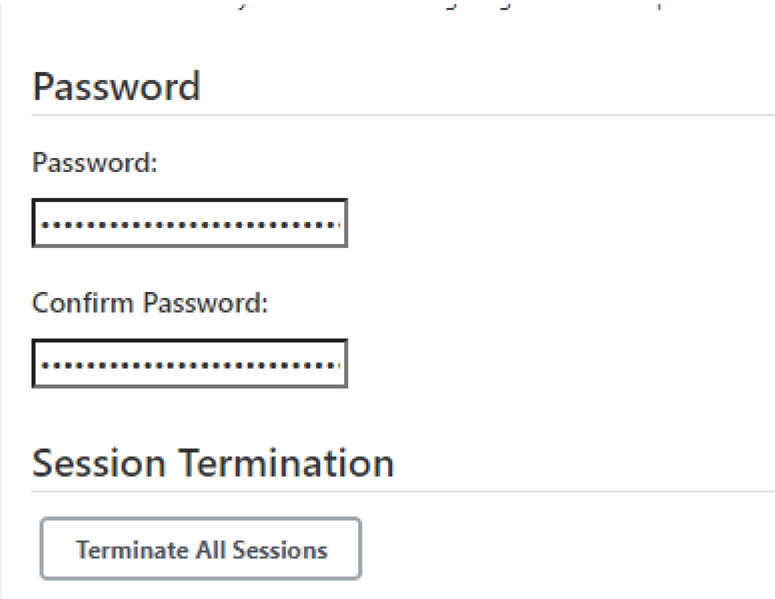
Please copy the password from either location and paste it below.

Administrator password

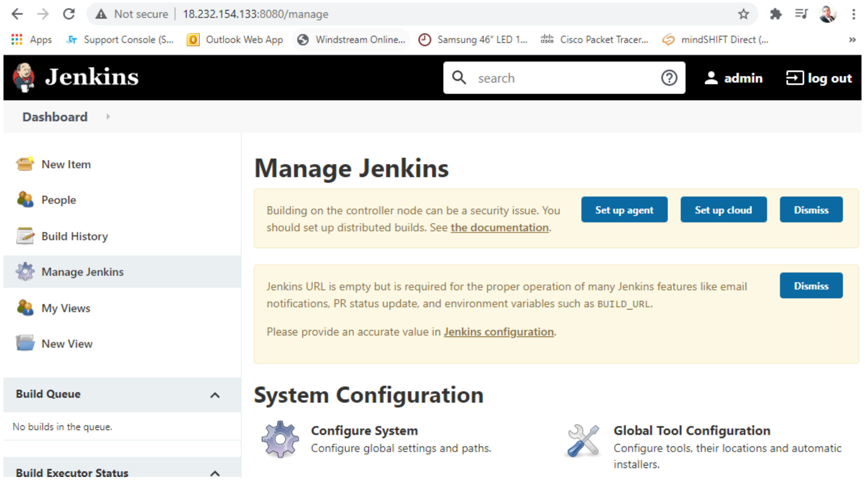


```
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-37-32 ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
5c053909d0c54523a2090c5361852a71
[root@ip-172-31-37-32 ~]#
```





7. Re login and go to **Global Tool Configuration** to set up Java home path.



No builds in the queue.

Manage Jenkins

Building on the controller node can be a security issue. You should set up distributed builds. See the documentation.

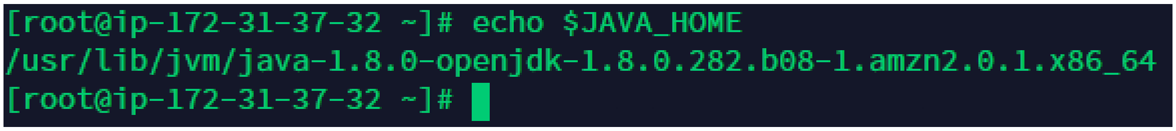
Jenkins URL is empty but is required for the proper operation of many Jenkins features like email notifications, PR status update, and environment variables such as `BUILD_URL`.

Please provide an accurate value in Jenkins configuration.

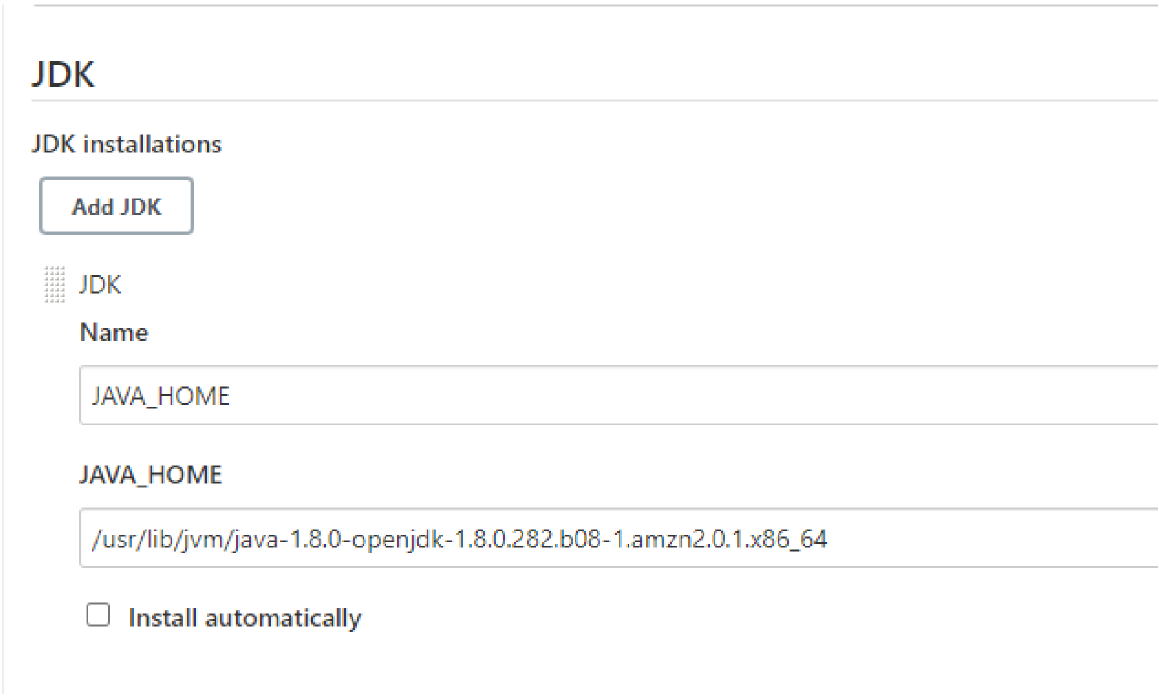
System Configuration

Configure System
Configure global settings and paths.

Global Tool Configuration
Configure tools, their locations and automatic installers.



```
[root@ip-172-31-37-32 ~]# echo $JAVA_HOME
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86_64
[root@ip-172-31-37-32 ~]#
```



JDK

JDK installations

JDK

Name

JAVA_HOME

JAVA_HOME

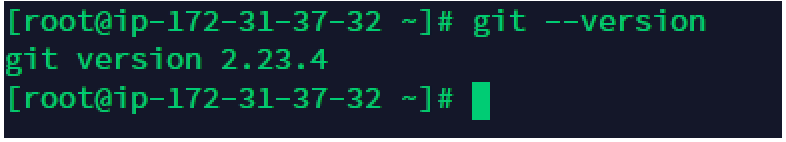
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86_64

☐ Install automatically

Click Save.

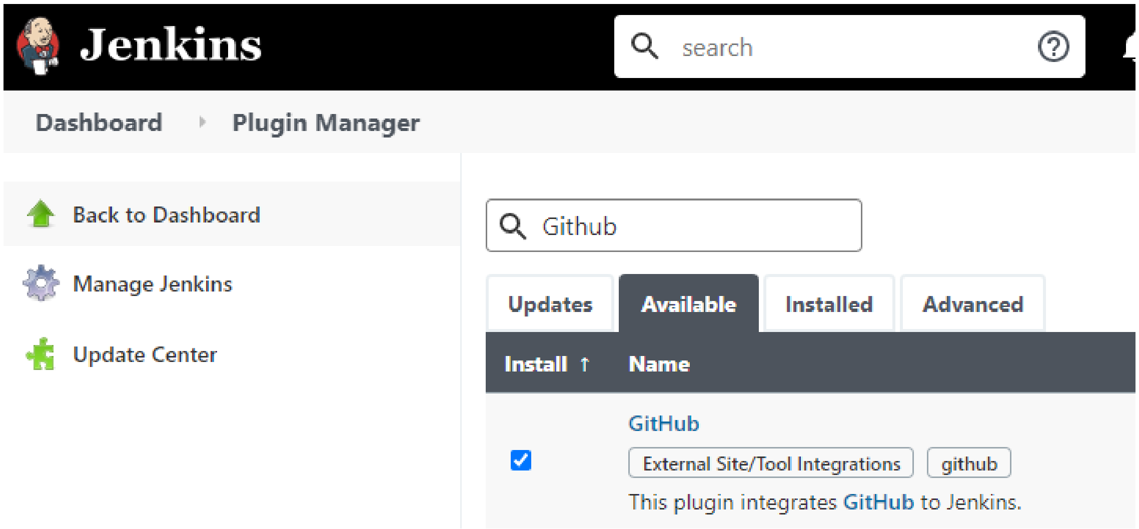
8. Install git on jenkins server

```
yum install git -y
git --version
```



```
[root@ip-172-31-37-32 ~]# git --version
git version 2.23.4
[root@ip-172-31-37-32 ~]#
```

9. Go to Jenkins Dashboard **Manage Jenkins> Plugin Manager** and install **GitHub plugin**

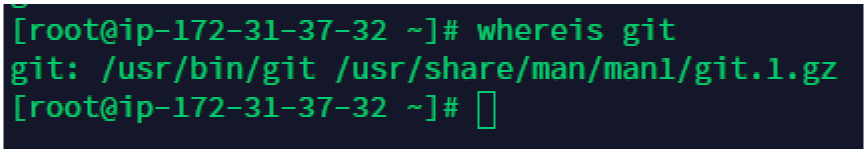


Github

Updates | Available | Installed | Advanced

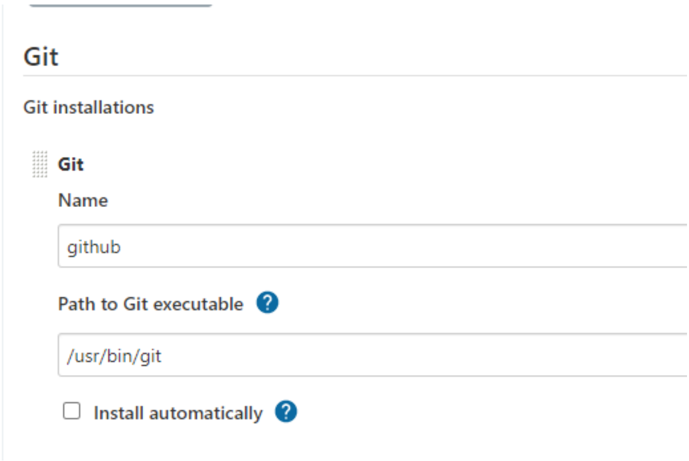
Install ↑	Name
☑	GitHub External Site/Tool Integrations github This plugin integrates GitHub to Jenkins.

10. Check Git path



```
[root@ip-172-31-37-32 ~]# whereis git
git: /usr/bin/git /usr/share/man/man1/git.1.gz
[root@ip-172-31-37-32 ~]# 
```

11. Go to Jenkins Global Tools configuration, configure GitHub and use the path above /usr/bin/git and save.



Git

Git installations

Git

Name

github

Path to Git executable

/usr/bin/git

☐ Install automatically

Section II. Setup Maven

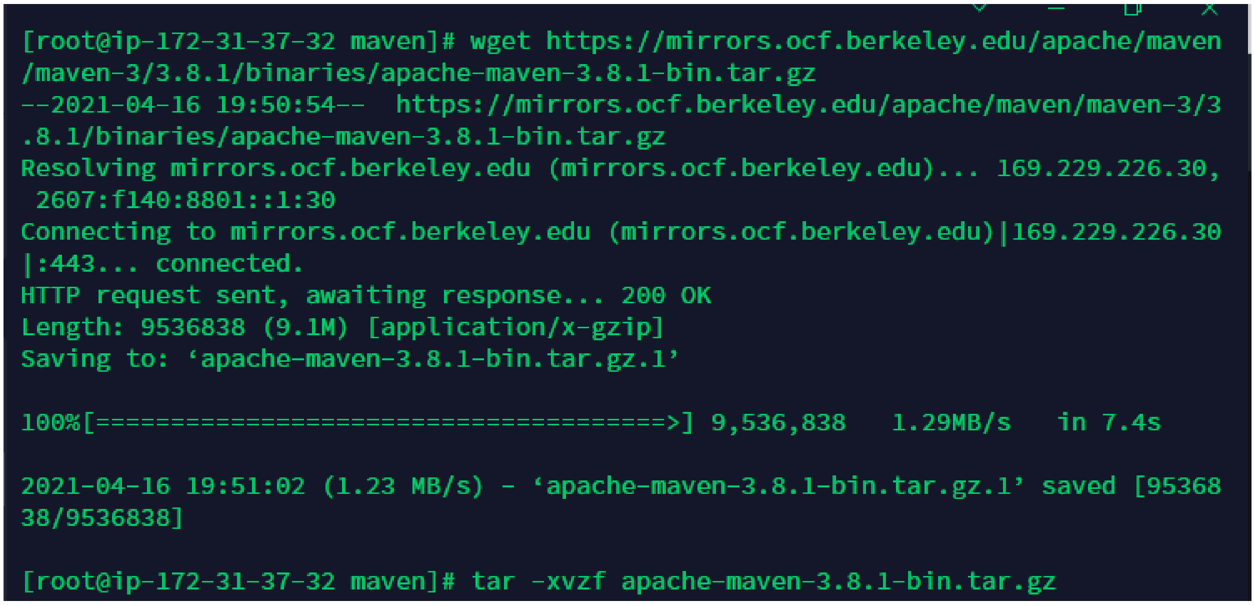
1. Install Maven on Jenkins
   Download maven packages from https://maven.apache.org/download.cgi onto Jenkins server by copying the link address. In this case, were using /opt/maven as the installation directory.

   a. Create a maven directory under /opt and change into it:
      mkdir /opt/maven
      cd /opt/maven

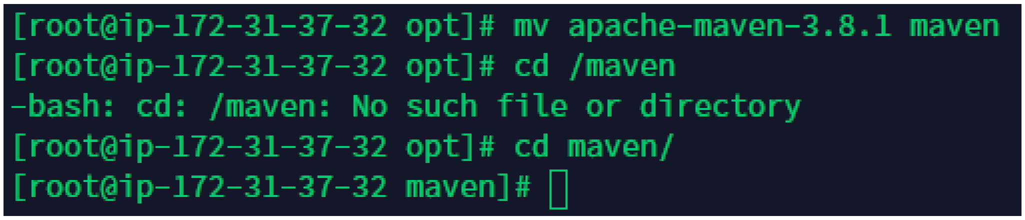
   b. Download maven version 3.6.0 and extract it using tar

   wget https://mirrors.ocf.berkeley.edu/apache/maven/maven-3/3.8.1/binaries/apache-maven-3.8.1-bin.tar.gz

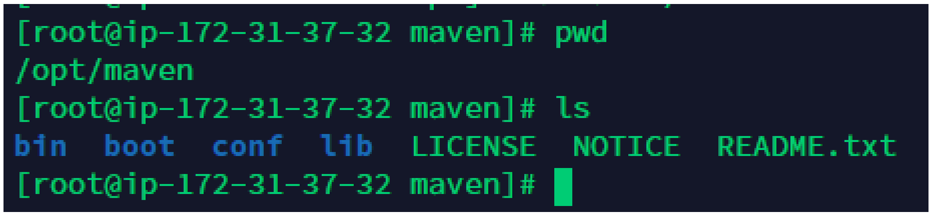
   tar -xvzf apache-maven-3.8.1-bin.tar.gz



```
[root@ip-172-31-37-32 maven]# wget https://mirrors.ocf.berkeley.edu/apache/maven
/maven-3/3.8.1/binaries/apache-maven-3.8.1-bin.tar.gz
--2021-04-16 19:50:54--  https://mirrors.ocf.berkeley.edu/apache/maven/maven-3/3
.8.1/binaries/apache-maven-3.8.1-bin.tar.gz
Resolving mirrors.ocf.berkeley.edu (mirrors.ocf.berkeley.edu)... 169.229.226.30,
 2607:f140:8801::1:30
Connecting to mirrors.ocf.berkeley.edu (mirrors.ocf.berkeley.edu)|169.229.226.30
|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9536838 (9.1M) [application/x-gzip]
Saving to: ‘apache-maven-3.8.1-bin.tar.gz.1’

100%[======================================>] 9,536,838   1.29MB/s   in 7.4s

2021-04-16 19:51:02 (1.23 MB/s) - ‘apache-maven-3.8.1-bin.tar.gz.1’ saved [95368
38/9536838]

[root@ip-172-31-37-32 maven]# tar -xvzf apache-maven-3.8.1-bin.tar.gz
```



```
[root@ip-172-31-37-32 opt]# mv apache-maven-3.8.1 maven
[root@ip-172-31-37-32 opt]# cd /maven
-bash: cd: /maven: No such file or directory
[root@ip-172-31-37-32 opt]# cd maven/
[root@ip-172-31-37-32 maven]# 
```

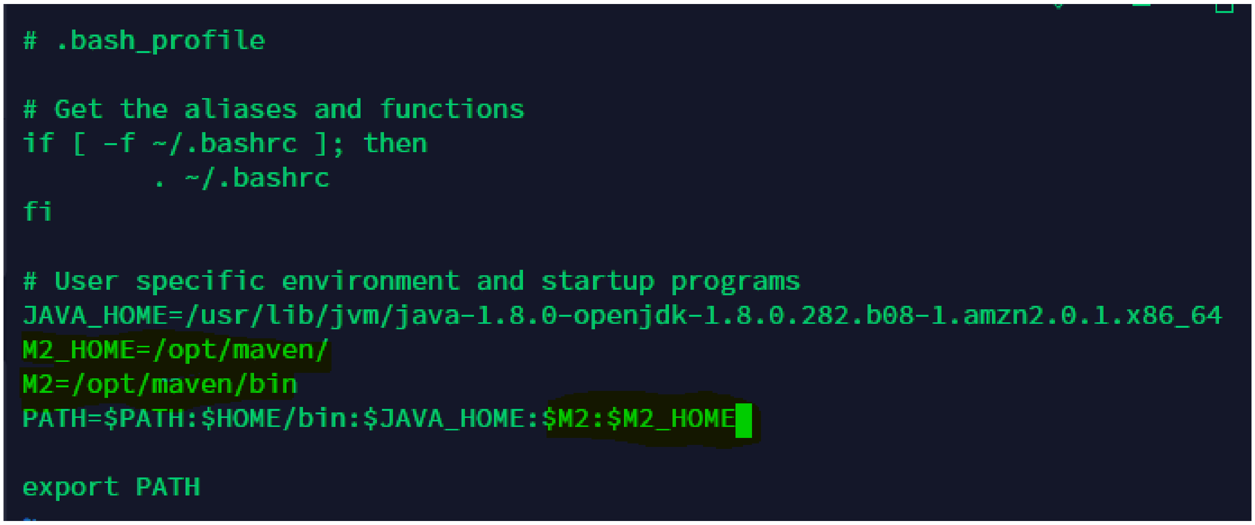


```
[root@ip-172-31-37-32 maven]# pwd
/opt/maven
[root@ip-172-31-37-32 maven]# ls
bin  boot  conf  lib  LICENSE  NOTICE  README.txt
[root@ip-172-31-37-32 maven]# 
```

c. Setup M2_HOME and M2 paths in .bash_profile of the user and add these to the path variable:

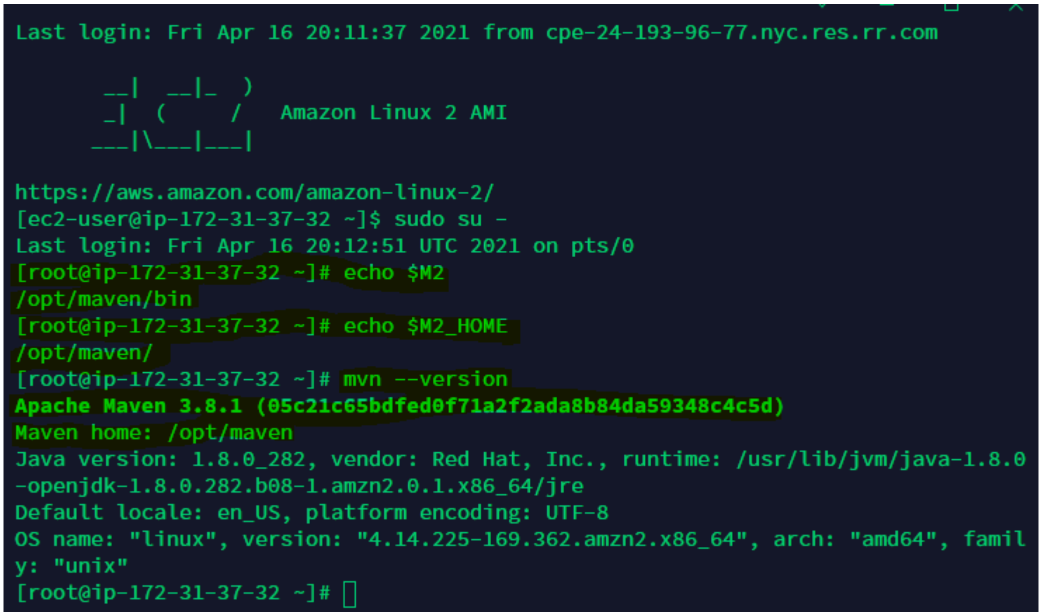
vi ~/.bash_profile

M2_HOME=/opt/maven/
M2=/opt/maven/bin
PATH=<Existing_PATH>:$M2_HOME:$M2



```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs
JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86_64
M2_HOME=/opt/maven/
M2=/opt/maven/bin
PATH=$PATH:$HOME/bin:$JAVA_HOME:$M2:$M2_HOME

export PATH
```

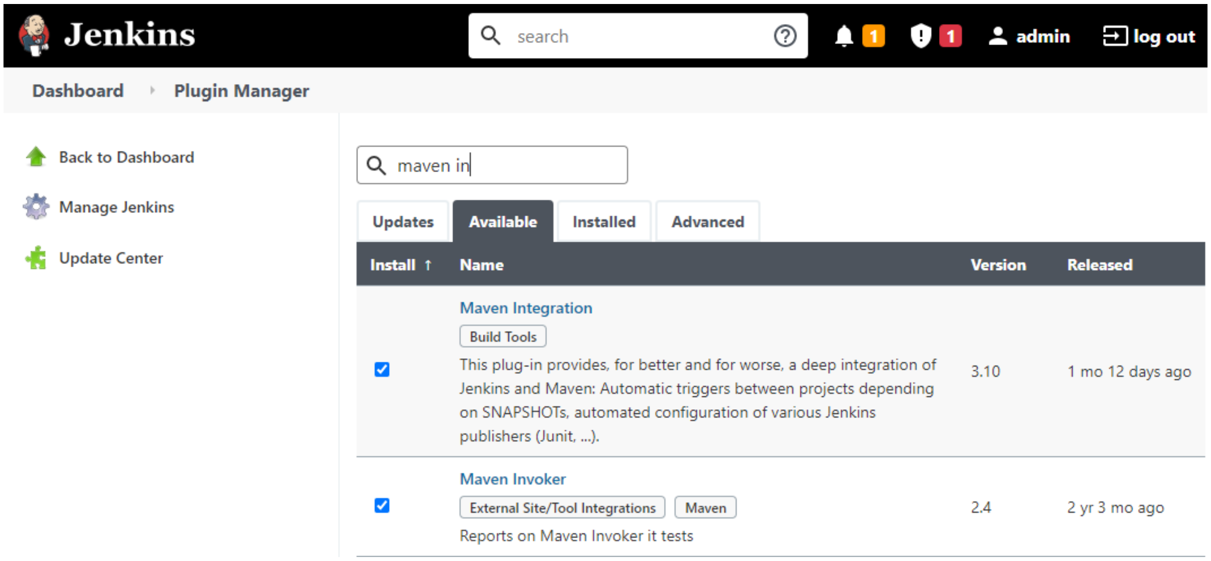
d. logoff and login to check the new path variable and maven version:
echo $M2
mvn –version



```
Last login: Fri Apr 16 20:11:37 2021 from cpe-24-193-96-77.nyc.res.rr.com

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-37-32 ~]$ sudo su -
Last login: Fri Apr 16 20:12:51 UTC 2021 on pts/0
[root@ip-172-31-37-32 ~]# echo $M2
/opt/maven/bin
[root@ip-172-31-37-32 ~]# echo $M2_HOME
/opt/maven/
[root@ip-172-31-37-32 ~]# mvn --version
Apache Maven 3.8.1 (05c21c65bdfed0f71a2f2ada8b84da59348c4c5d)
Maven home: /opt/maven
Java version: 1.8.0_282, vendor: Red Hat, Inc., runtime: /usr/lib/jvm/java-1.8.0
-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.14.225-169.362.amzn2.x86_64", arch: "amd64", famil
y: "unix"
[root@ip-172-31-37-32 ~]# 
```

e. Setup maven on Jenkins console, Install maven plugin without restart:

- **Manage Jenkins > Jenkins Plugins > available > Maven Invoker**
- **Manage Jenkins > Jenkins Plugins > available > Maven Integration**

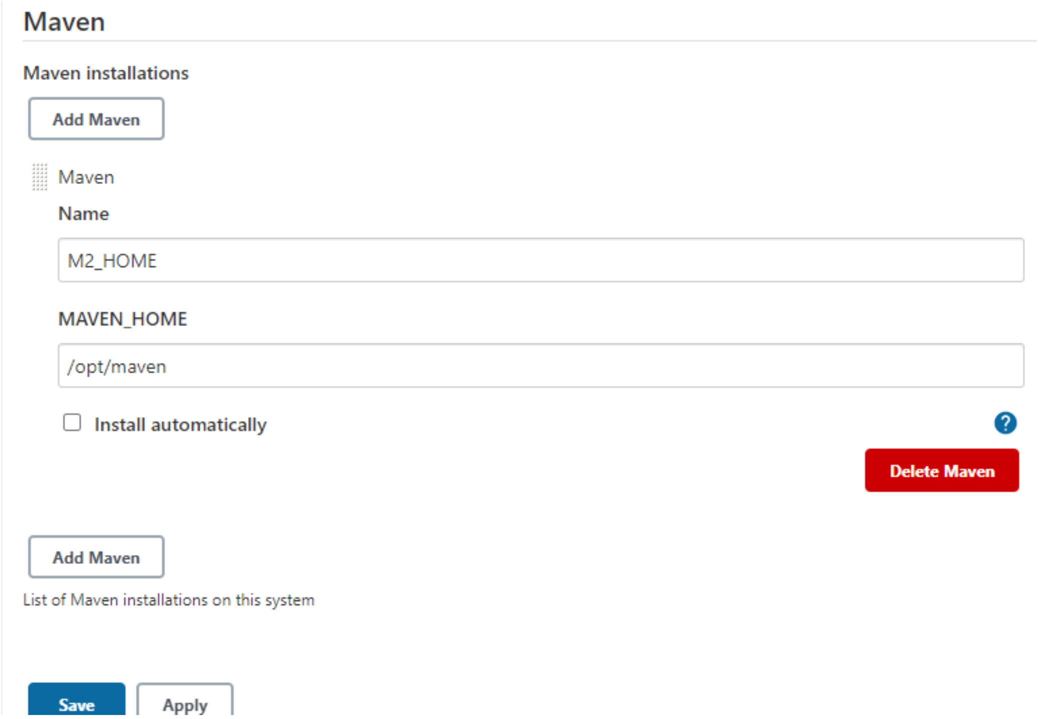


maven in

Install ↑	Name	Version	Released
☑	Maven Integration Build Tools This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTs, automated configuration of various Jenkins publishers (Junit, ...).	3.10	1 mo 12 days ago
☑	Maven Invoker External Site/Tool Integrations Maven Reports on Maven Invoker it tests	2.4	2 yr 3 mo ago

f. Configure maven path

- **Manage Jenkins > Global Tool Configuration > Maven**



Maven

Maven installations

Maven

Name

M2_HOME

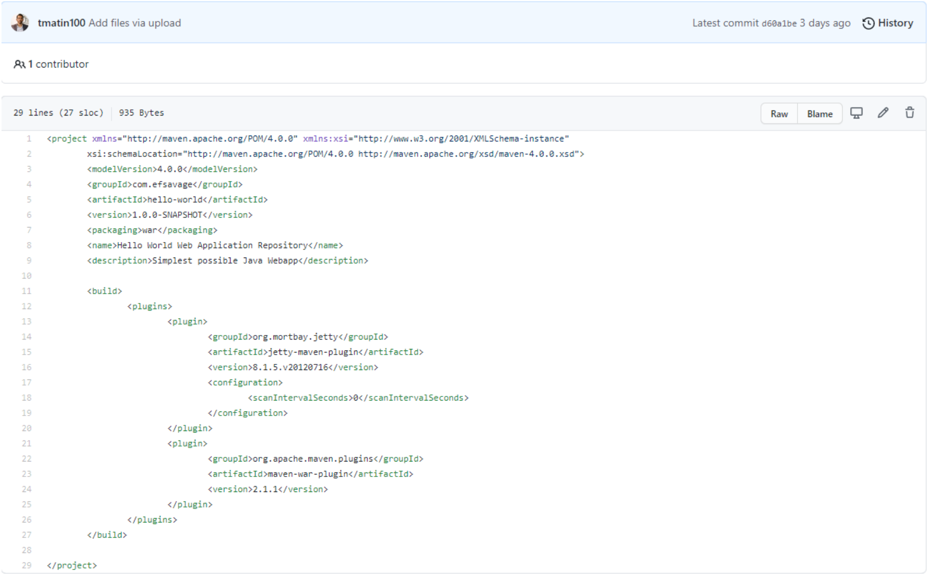
MAVEN_HOME

/opt/maven

□ Install automatically

List of Maven installations on this system

g. Finally, we are going to create a **POM.XML** file for the maven job in our GitHub repository. https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project/blob/master/pom.xml



tmatin100 Add files via upload Latest commit d60a1be 3 days ago History

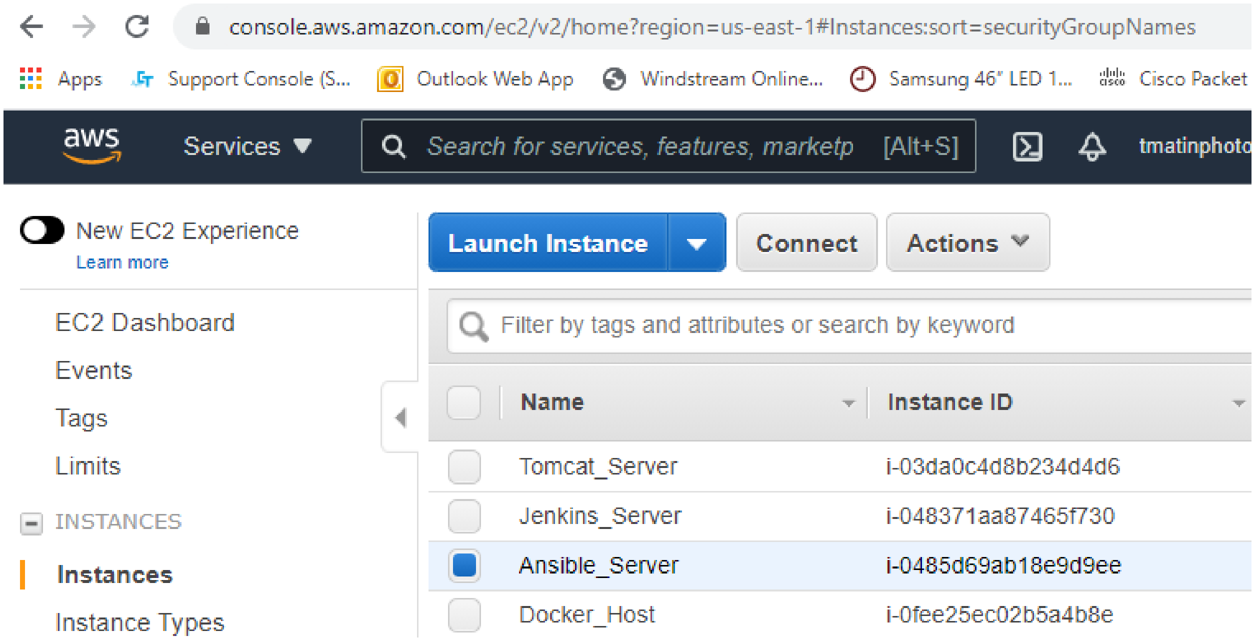
1 contributor

29 lines (27 sloc) | 935 Bytes

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <groupId>com.efsavage</groupId>
        <artifactId>hello-world</artifactId>
        <version>1.0.0-SNAPSHOT</version>
        <packaging>war</packaging>
        <name>Hello World Web Application Repository</name>
        <description>Simplest possible Java Webapp</description>

        <build>
                <plugins>
                        <plugin>
                                <groupId>org.mortbay.jetty</groupId>
                                <artifactId>jetty-maven-plugin</artifactId>
                                <version>8.1.5.v20120716</version>
                                <configuration>
                                        <scanIntervalSeconds>0</scanIntervalSeconds>
                                </configuration>
                        </plugin>
                        <plugin>
                                <groupId>org.apache.maven.plugins</groupId>
                                <artifactId>maven-war-plugin</artifactId>
                                <version>2.1.1</version>
                        </plugin>
                </plugins>
        </build>

</project>
```

Section III. Setup Ansible Host

1. Launch AWS EC2 Instance



console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=securityGroupNames

	Name	Instance ID
□	Tomcat_Server	i-03da0c4d8b234d4d6
□	Jenkins_Server	i-048371aa87465f730
☑	Ansible_Server	i-0485d69ab18e9d9ee
□	Docker_Host	i-0fee25ec02b5a4b8e

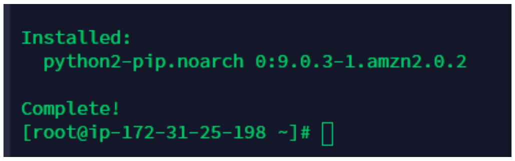
2. Install Ansible

a. Install python and python-pip:

yum install python

yum install python-pip

(sudo amazon-linux-extras install ansible2)

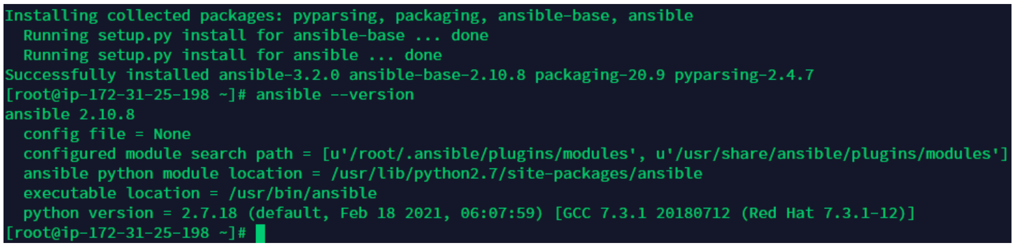


```
Installed:
  python2-pip.noarch 0:9.0.3-1.amzn2.0.2

Complete!
[root@ip-172-31-25-198 ~]# 
```

b. Install ansible using pip check for version:

pip install ansible

ansible –version

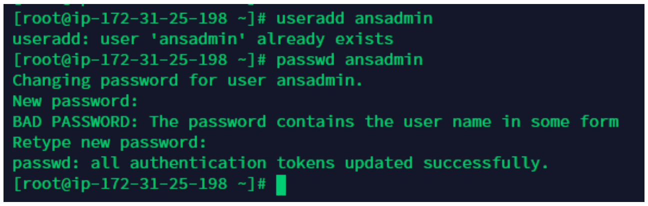


```
Installing collected packages: pyparsing, packaging, ansible-base, ansible
  Running setup.py install for ansible-base ... done
  Running setup.py install for ansible ... done
Successfully installed ansible-3.2.0 ansible-base-2.10.8 packaging-20.9 pyparsing-2.4.7
[root@ip-172-31-25-198 ~]# ansible --version
ansible 2.10.8
  config file = None
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.18 (default, Feb 18 2021, 06:07:59) [GCC 7.3.1 20180712 (Red Hat 7.3.1-12)]
[root@ip-172-31-25-198 ~]# 
```

c. Create a user called ansadmin (on Control node and Managed host):

useradd ansadmin

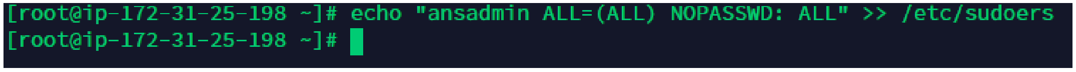
passwd ansadmin



```
[root@ip-172-31-25-198 ~]# useradd ansadmin
useradd: user 'ansadmin' already exists
[root@ip-172-31-25-198 ~]# passwd ansadmin
Changing password for user ansadmin.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-25-198 ~]#
```

d. Below command grant sudo access to ansadmin user. But we strongly recommended using "visudo" command if you are aware vi or nano editor. (on Control node and Managed host) :

echo "ansadmin ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

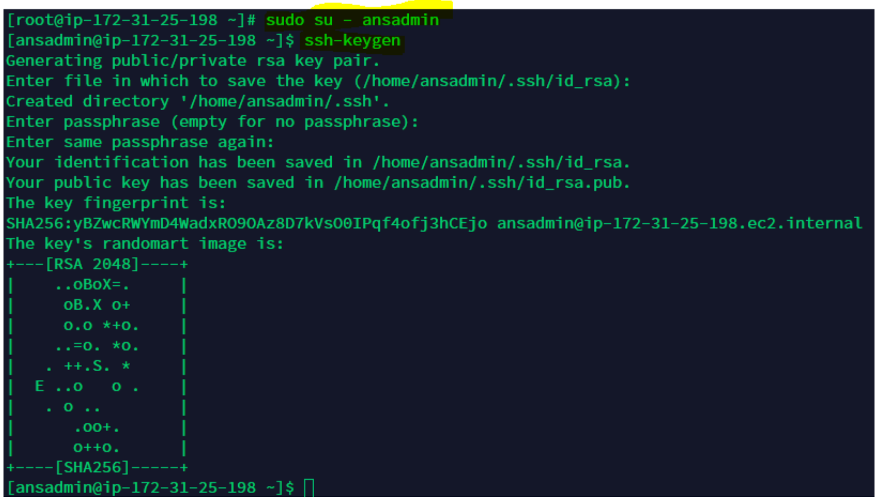


```
[root@ip-172-31-25-198 ~]# echo "ansadmin ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
[root@ip-172-31-25-198 ~]#
```

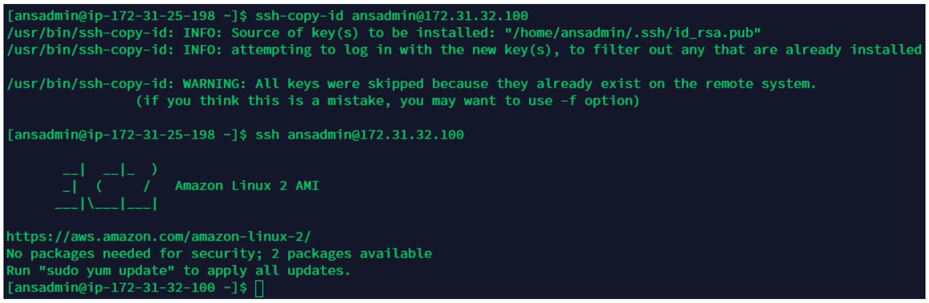
e. Log in as a ansadmin user on master and generate ssh key (on Control node):

sudo su - ansadmin

ssh-keygen



```
[root@ip-172-31-25-198 ~]# sudo su - ansadmin
[ansadmin@ip-172-31-25-198 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ansadmin/.ssh/id_rsa):
Created directory '/home/ansadmin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ansadmin/.ssh/id_rsa.
Your public key has been saved in /home/ansadmin/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:yBZwcRWYmD4WadxRO9OAz8D7kVsO0IPqf4ofj3hCEjo ansadmin@ip-172-31-25-198.ec2.internal
The key's randomart image is:
+---[RSA 2048]----+
|    ..oBoX=.     |
|     oB.X o+     |
|     o.o *+o.    |
|    ..=o. *o.    |
|   . ++.S. *     |
|  E ..o   o .    |
|   . o ..        |
|      .oo+.      |
|      o++o.      |
+----[SHA256]-----+
[ansadmin@ip-172-31-25-198 ~]$
```



```
[ansadmin@ip-172-31-25-198 ~]$ ssh-copy-id ansadmin@172.31.32.100
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote system.
                (if you think this is a mistake, you may want to use -f option)

[ansadmin@ip-172-31-25-198 ~]$ ssh ansadmin@172.31.32.100

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

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ansadmin@ip-172-31-32-100 ~]$
```

f. Ansible server used to create images and store on docker registry. Hence install docker, start docker services and add ansadmin to the docker group:

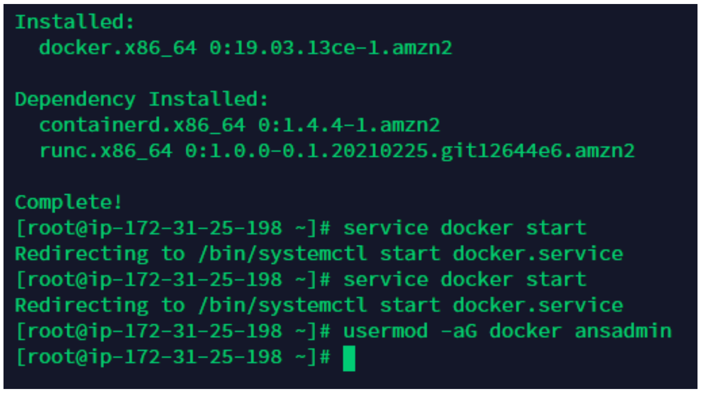
yum install docker

# start docker services

service docker start

# add user to docker group

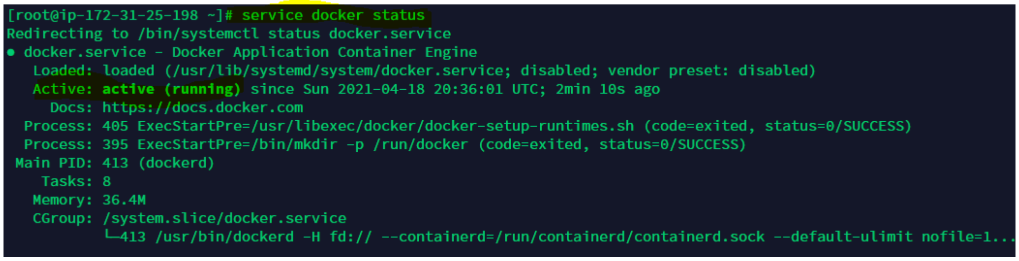
usermod -aG docker ansadmin



```
Installed:
  docker.x86_64 0:19.03.13ce-1.amzn2

Dependency Installed:
  containerd.x86_64 0:1.4.4-1.amzn2
  runc.x86_64 0:1.0.0-0.1.20210225.git12644e6.amzn2

Complete!
[root@ip-172-31-25-198 ~]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-31-25-198 ~]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-31-25-198 ~]# usermod -aG docker ansadmin
[root@ip-172-31-25-198 ~]# 
```

#check the docker service

service docker status

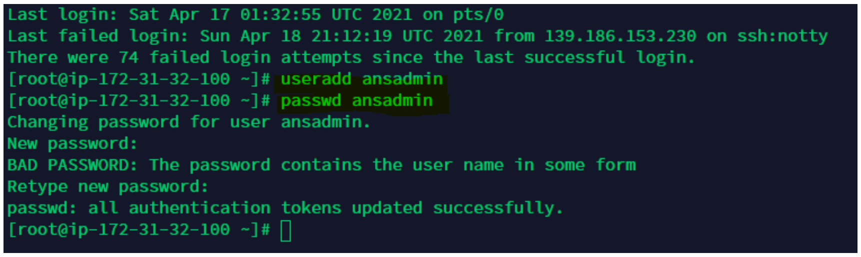


```
[root@ip-172-31-25-198 ~]# service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2021-04-18 20:36:01 UTC; 2min 10s ago
     Docs: https://docs.docker.com
  Process: 405 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
  Process: 395 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
 Main PID: 413 (dockerd)
    Tasks: 8
   Memory: 36.4M
   CGroup: /system.slice/docker.service
           └─413 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=1...
```

g. Login to the remote docker host server and create the ansadmin user :

useradd ansadmin

passwd ansadmin



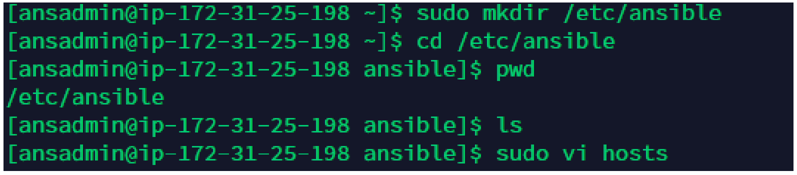
```
Last login: Sat Apr 17 01:32:55 UTC 2021 on pts/0
Last failed login: Sun Apr 18 21:12:19 UTC 2021 from 139.186.153.230 on ssh:notty
There were 74 failed login attempts since the last successful login.
[root@ip-172-31-32-100 ~]# useradd ansadmin
[root@ip-172-31-32-100 ~]# passwd ansadmin
Changing password for user ansadmin.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-32-100 ~]# 
```

# add ansadmin user to the sudoers file on both host and target sever, in this case the target is our docker-host

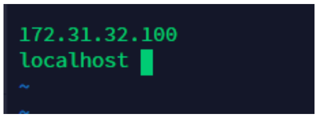
visudo

ansadmin ALL=(ALL)   NOPASSWD: ALL

h. Create a directory /etc/ansible and create an inventory file called "hosts" add control node and managed hosts IP addresses to it



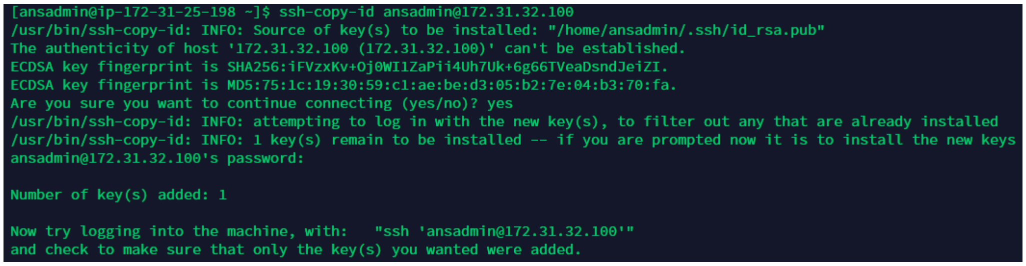
```
[ansadmin@ip-172-31-25-198 ~]$ sudo mkdir /etc/ansible
[ansadmin@ip-172-31-25-198 ~]$ cd /etc/ansible
[ansadmin@ip-172-31-25-198 ansible]$ pwd
/etc/ansible
[ansadmin@ip-172-31-25-198 ansible]$ ls
[ansadmin@ip-172-31-25-198 ansible]$ sudo vi hosts
```



```
172.31.32.100
localhost
~
~
```

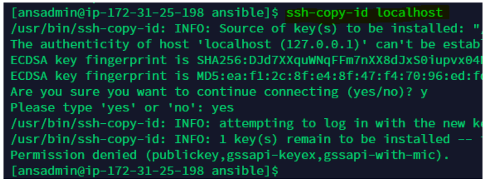
i. Copy keys onto all ansible managed hosts (on Control node), in this case we are going to copy into the docker host: ssh-copy-id ansadmin@<target-server>

ssh-copy-id ansadmin@172.31.32.100



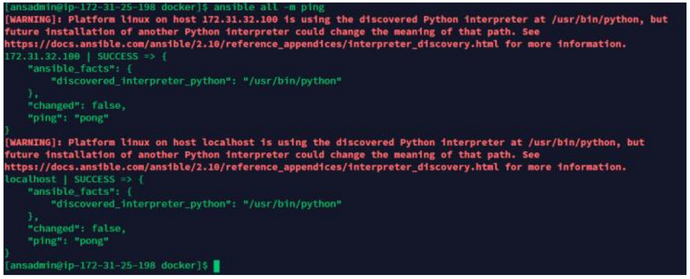
```
[ansadmin@ip-172-31-25-198 ~]$ ssh-copy-id ansadmin@172.31.32.100
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id_rsa.pub"
The authenticity of host '172.31.32.100 (172.31.32.100)' can't be established.
ECDSA key fingerprint is SHA256:iFVzxKv+Oj0WI1ZaPii4Uh7Uk+6g66TVeaDsndJeiZI.
ECDSA key fingerprint is MD5:75:1c:19:30:59:c1:ae:be:d3:05:b2:7e:04:b3:70:fa.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansadmin@172.31.32.100's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ansadmin@172.31.32.100'"
and check to make sure that only the key(s) you wanted were added.
```

#copy shh key to local host as well



```
[ansadmin@ip-172-31-25-198 ansible]$ ssh-copy-id localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "
The authenticity of host 'localhost (127.0.0.1)' can't be estab
ECDSA key fingerprint is SHA256:DJd7XXquWNqFFm7nXX8dJxS0iupvx04
ECDSA key fingerprint is MD5:ea:f1:2c:8f:e4:8f:47:f4:70:96:ed:f
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new k
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed --
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ansadmin@ip-172-31-25-198 ansible]$
```

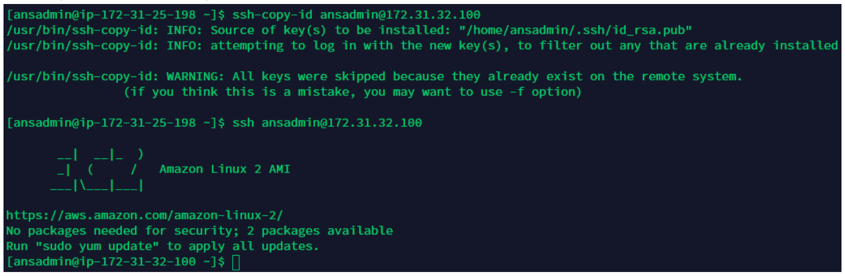
#varify connectivity by using **ansible all –m ping** command:



```
[ansadmin@ip-172-31-25-198 docker]$ ansible all -m ping
[WARNING]: Platform linux on host 172.31.32.100 is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible/2.10/reference_appendices/interpreter_discovery.html for more information.
172.31.32.100 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host localhost is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible/2.10/reference_appendices/interpreter_discovery.html for more information.
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[ansadmin@ip-172-31-25-198 docker]$
```

#Login to the remote to the remote docker host server.

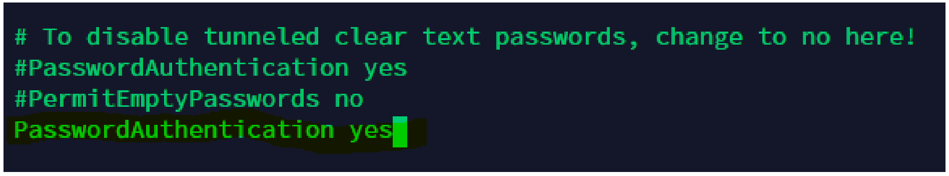
ssh @ansadmin@172.31.32.100



```
[ansadmin@ip-172-31-25-198 ~]$ ssh-copy-id ansadmin@172.31.32.100
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansadmin/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote system.
                (if you think this is a mistake, you may want to use -f option)

[ansadmin@ip-172-31-25-198 ~]$ ssh ansadmin@172.31.32.100

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

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ansadmin@ip-172-31-32-100 ~]$ 
```

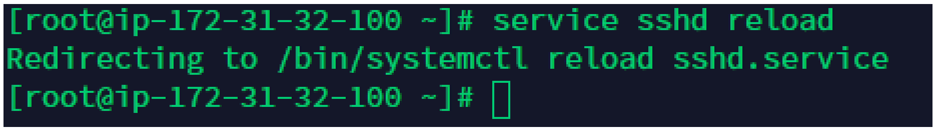
j. Go to your ansible host ec2 instance and edit the file /etc/ssh/sshd_config , change the Password Authentication option to yes, exit and reload sshd service.

vi /etc/ssh/sshd_config



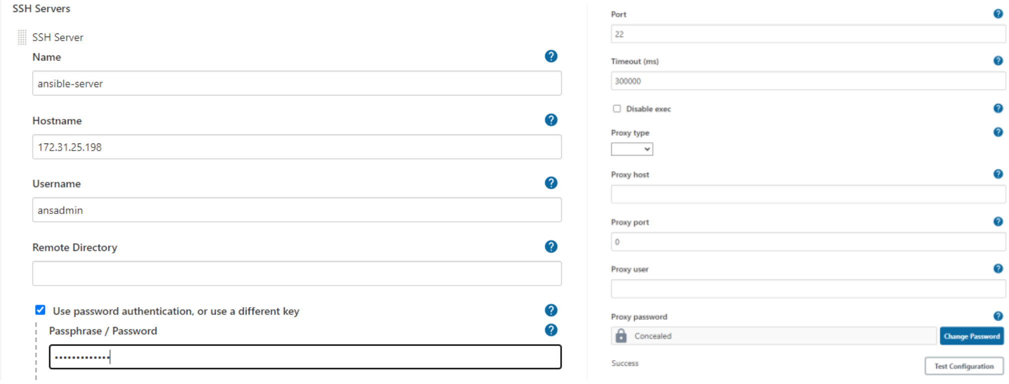
```
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes
```

service sshd reload



```
[root@ip-172-31-32-100 ~]# service sshd reload
Redirecting to /bin/systemctl reload sshd.service
[root@ip-172-31-32-100 ~]# 
```

k. Integrate ansible with Jenkins server. Go to manage Jenkins>configure system> publish over ssh and fill in the information, test connection. It should say success. Click apply and save.



SSH Servers

SSH Server

Name: ansible-server

Hostname: 172.31.25.198

Username: ansadmin

Remote Directory:

☑ Use password authentication, or use a different key

Passphrase / Password: •••••••••••••

Port: 22

Timeout (ms): 300000

☐ Disable exec

Proxy type:

Proxy host:

Proxy port: 0

Proxy user:

Proxy password: Concealed

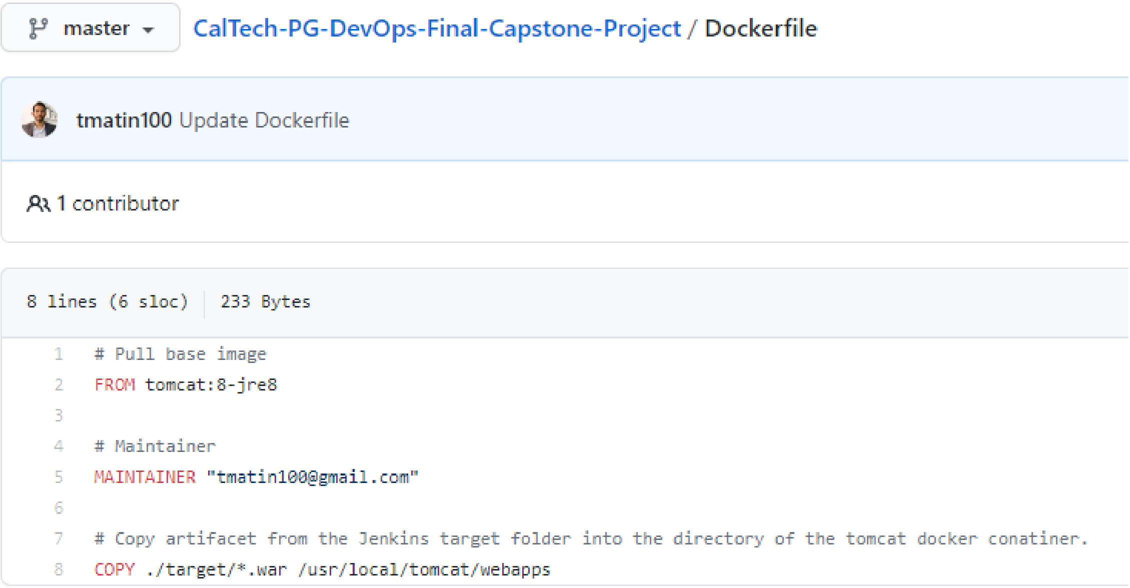
Success

IV. Create a Dockerfile

A **Dockerfile** is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build we can create an automated build that executes several command-line instructions in succession.

1. In this case we will pull the tomcat image from docker hub and use the COPY command to copy the Jenkins .war file into the directory of the tomcat container.

Dockerfile: https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project/blob/master/Dockerfile



tmatin100 Update Dockerfile

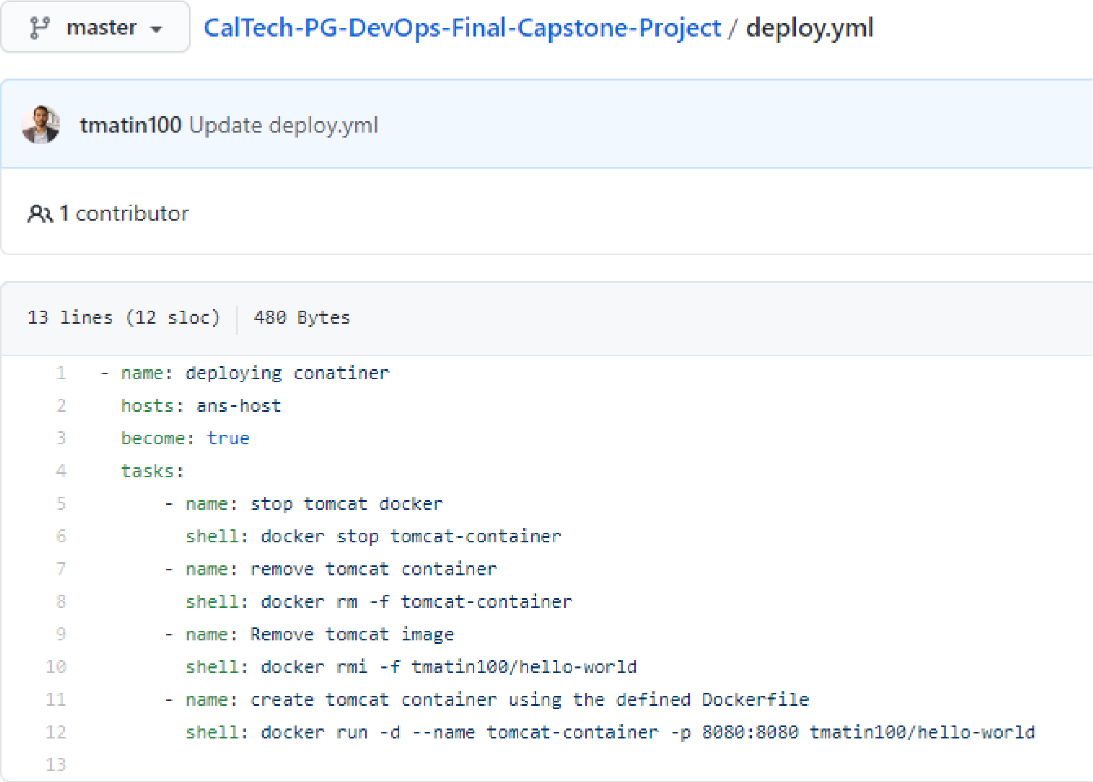
1 contributor

8 lines (6 sloc) | 233 Bytes

```
# Pull base image
FROM tomcat:8-jre8

# Maintainer
MAINTAINER "tmatin100@gmail.com"

# Copy artifacet from the Jenkins target folder into the directory of the tomcat docker conatiner.
COPY ./target/*.war /usr/local/tomcat/webapps
```

2.However, we do not want to do the build process manually, so let’s create an Ansible playbook named deploy.yml, that will automate our tasks once the code is checked into our repository, and when the Jenkins job is triggered. This play book will build the Docker image, copy the Jenkins artifact into the directory of the tomcat container, each time a new code is committed to the GitHub repository.

Ansible Playbook: https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project/blob/master/deploy.yml



master CalTech-PG-DevOps-Final-Capstone-Project / deploy.yml

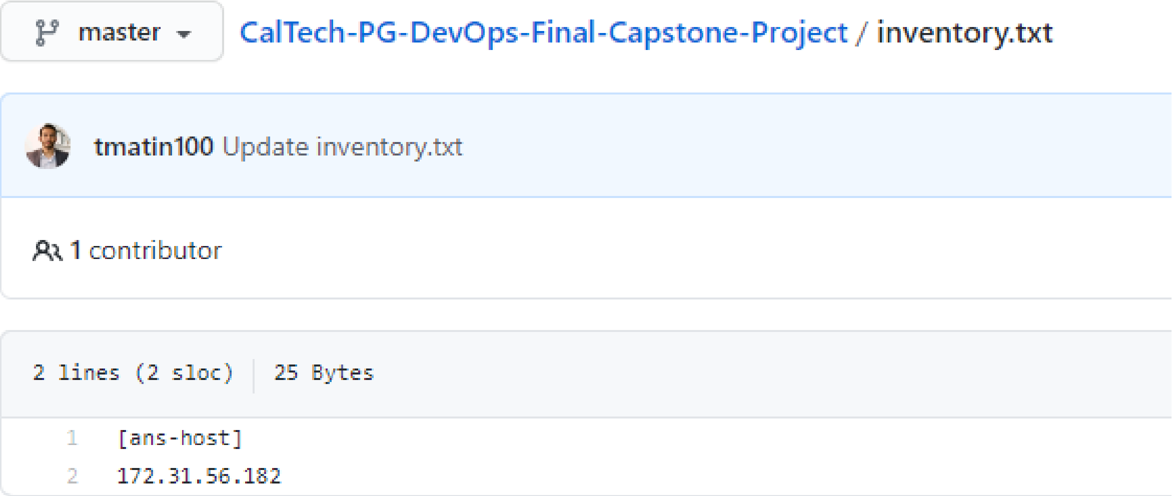
tmatin100 Update deploy.yml

1 contributor

13 lines (12 sloc) | 480 Bytes

```yaml
- name: deploying conatiner
  hosts: ans-host
  become: true
  tasks:
      - name: stop tomcat docker
        shell: docker stop tomcat-container
      - name: remove tomcat container
        shell: docker rm -f tomcat-container
      - name: Remove tomcat image
        shell: docker rmi -f tmatin100/hello-world
      - name: create tomcat container using the defined Dockerfile
        shell: docker run -d --name tomcat-container -p 8080:8080 tmatin100/hello-world

```

Ansible will use an inventory file, in order to know where to deploy this new container. In this case it is our Ansible host, ec2 instance.



master CalTech-PG-DevOps-Final-Capstone-Project / inventory.txt

tmatin100 Update inventory.txt

1 contributor

2 lines (2 sloc) | 25 Bytes

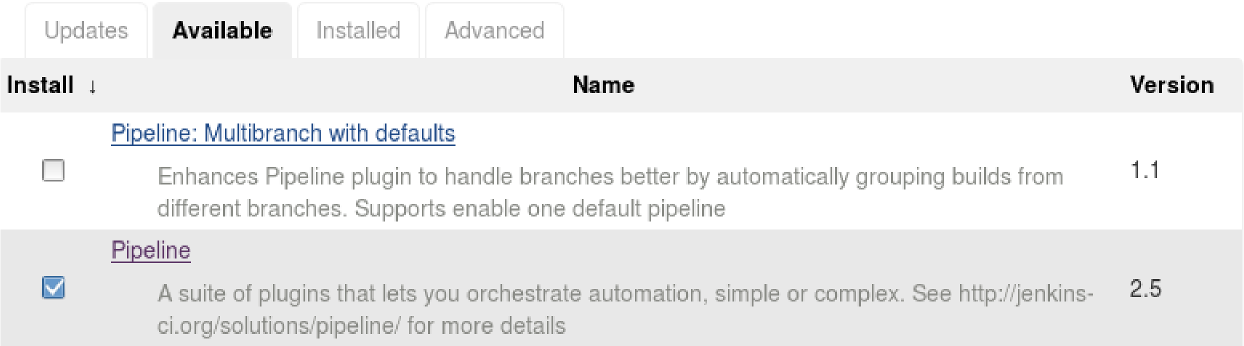
```
[ans-host]
172.31.56.182
```

V. Create a Jenkins Pipeline Project

A **Jenkins Pipeline** is a set of plugins which supports Continuous Integration (CI) and delivery.

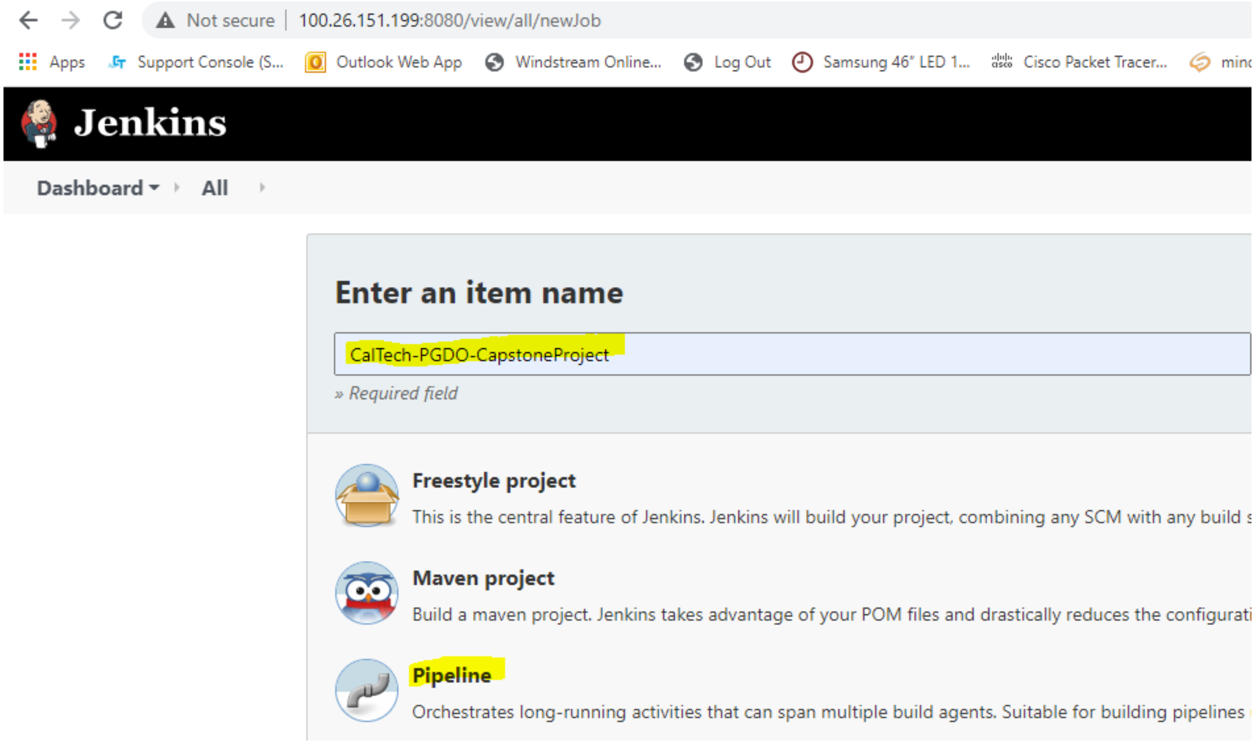
The first thing to get started Manage your Jenkins plugins is to add the pipeline plugin.

Go to **Mange Jenkins>Manage Plugins>** and search for the Pipeline plugin and add it.



Install ↓	Name	Version
☐	Pipeline: Multibranch with defaults Enhances Pipeline plugin to handle branches better by automatically grouping builds from different branches. Supports enable one default pipeline	1.1
☑	Pipeline A suite of plugins that lets you orchestrate automation, simple or complex. See http://jenkins-ci.org/solutions/pipeline/ for more details	2.5

In order to create a new Pipeline, go to the **Dashboard** select "**New Item**" and choose "**Pipeline**" give it a name and create project:



Enter an item name

CalTech-PGDO-CapstoneProject

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build s

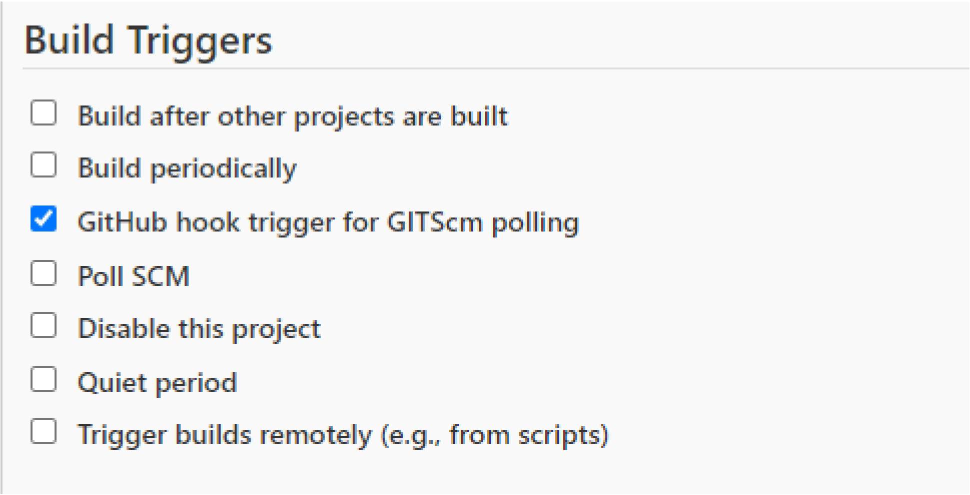
Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configurati

Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines

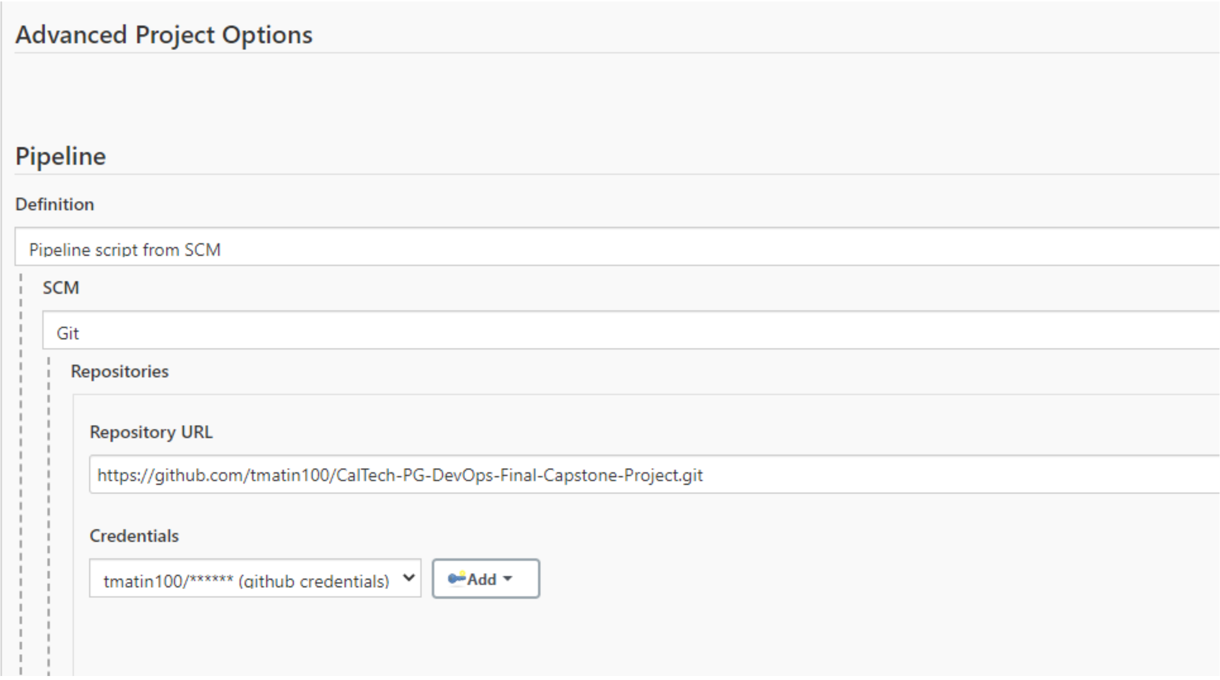
Scroll down to the “**Build Triggers**” section and choose GitHub hook trigger for **GITScm** Polling



Build Triggers

- ☐ Build after other projects are built
- ☐ Build periodically
- ☑ GitHub hook trigger for GITScm polling
- ☐ Poll SCM
- ☐ Disable this project
- ☐ Quiet period
- ☐ Trigger builds remotely (e.g., from scripts)

Scroll down to the “**Pipeline**” section and provide the “Repository URL”.



Advanced Project Options

Pipeline

Definition

Pipeline script from SCM

SCM

Git

Repositories

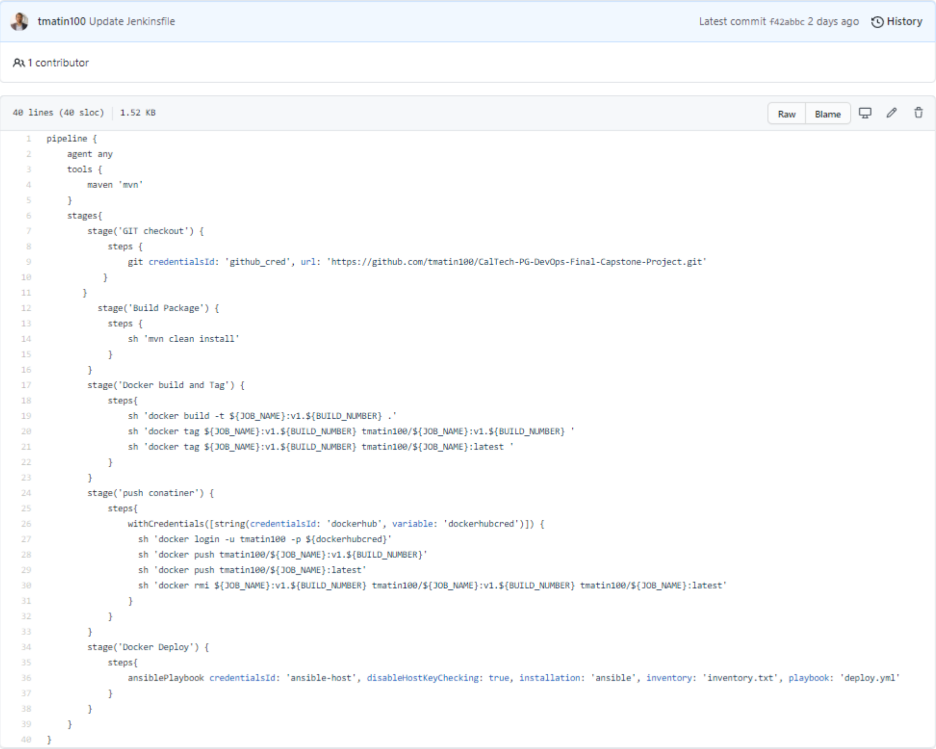
Repository URL

https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project.git

Credentials

tmatin100/****** (github credentials)

Jenkinsfile: https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project/blob/master/Jenkinsfile



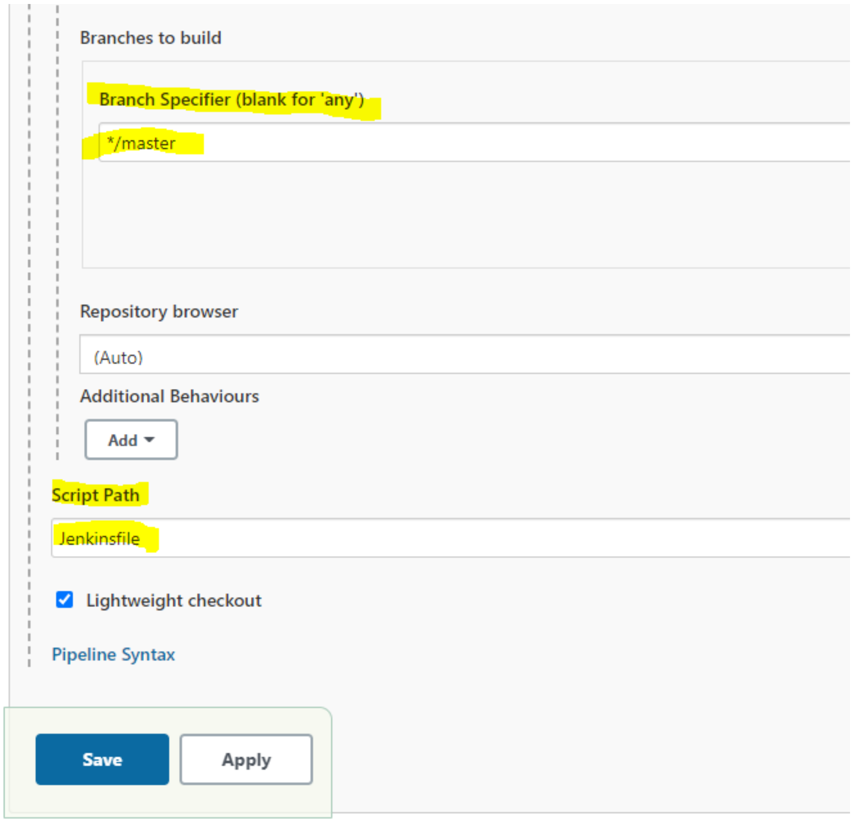
tmatin100 Update Jenkinsfile   Latest commit f42abbc 2 days ago   History

1 contributor

40 lines (40 sloc) | 1.52 KB

```
pipeline {
    agent any
    tools {
        maven 'mvn'
    }
    stages{
        stage('GIT checkout') {
            steps {
                git credentialsId: 'github_cred', url: 'https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project.git'
           }
       }
          stage('Build Package') {
            steps {
                sh 'mvn clean install'
            }
        }
        stage('Docker build and Tag') {
            steps{
                sh 'docker build -t ${JOB_NAME}:v1.${BUILD_NUMBER} .'
                sh 'docker tag ${JOB_NAME}:v1.${BUILD_NUMBER} tmatin100/${JOB_NAME}:v1.${BUILD_NUMBER} '
                sh 'docker tag ${JOB_NAME}:v1.${BUILD_NUMBER} tmatin100/${JOB_NAME}:latest '
            }
        }
        stage('push conatiner') {
            steps{
                withCredentials([string(credentialsId: 'dockerhub', variable: 'dockerhubcred')]) {
                  sh 'docker login -u tmatin100 -p ${dockerhubcred}'
                  sh 'docker push tmatin100/${JOB_NAME}:v1.${BUILD_NUMBER}'
                  sh 'docker push tmatin100/${JOB_NAME}:latest'
                  sh 'docker rmi ${JOB_NAME}:v1.${BUILD_NUMBER} tmatin100/${JOB_NAME}:v1.${BUILD_NUMBER} tmatin100/${JOB_NAME}:latest'
                }
            }
        }
        stage('Docker Deploy') {
            steps{
                ansiblePlaybook credentialsId: 'ansible-host', disableHostKeyChecking: true, installation: 'ansible', inventory: 'inventory.txt', playbook: 'deploy.yml'
            }
        }
    }
}
```

Scroll down to the **Branch Specifier** and **Script Path** and provide the following info bellow, save.



Branches to build

Branch Specifier (blank for 'any')

*/master

Repository browser

(Auto)

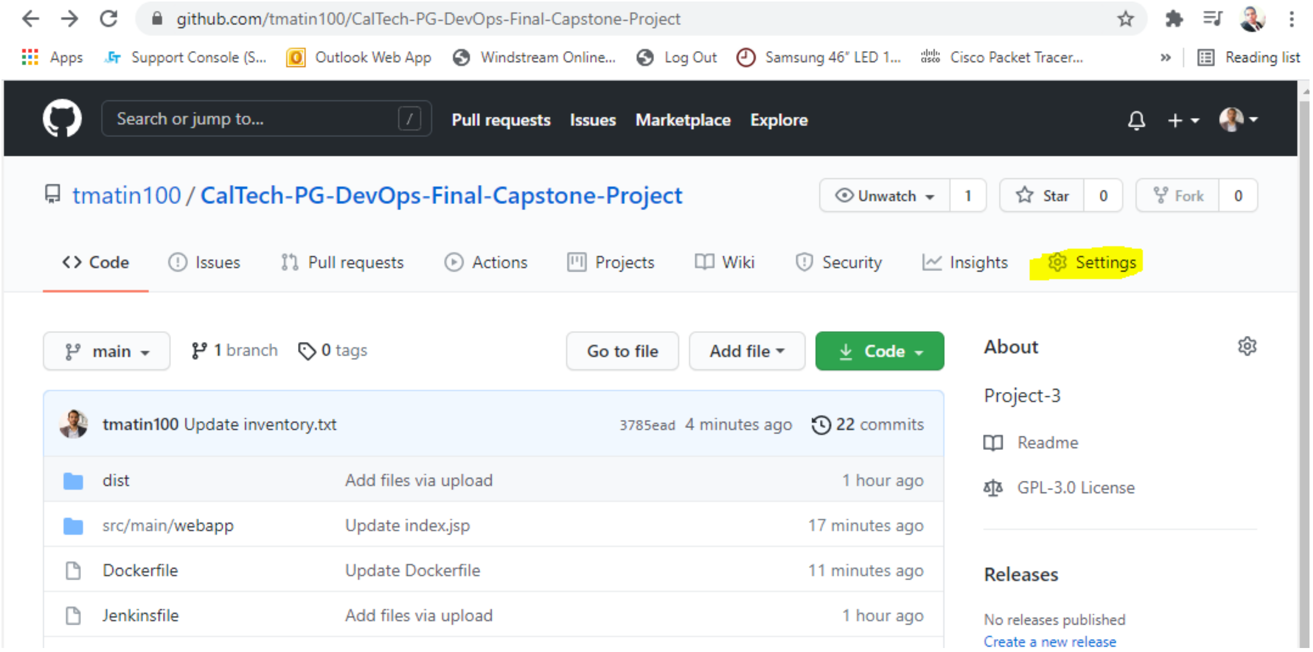
Additional Behaviours

Script Path

Jenkinsfile

☑ Lightweight checkout

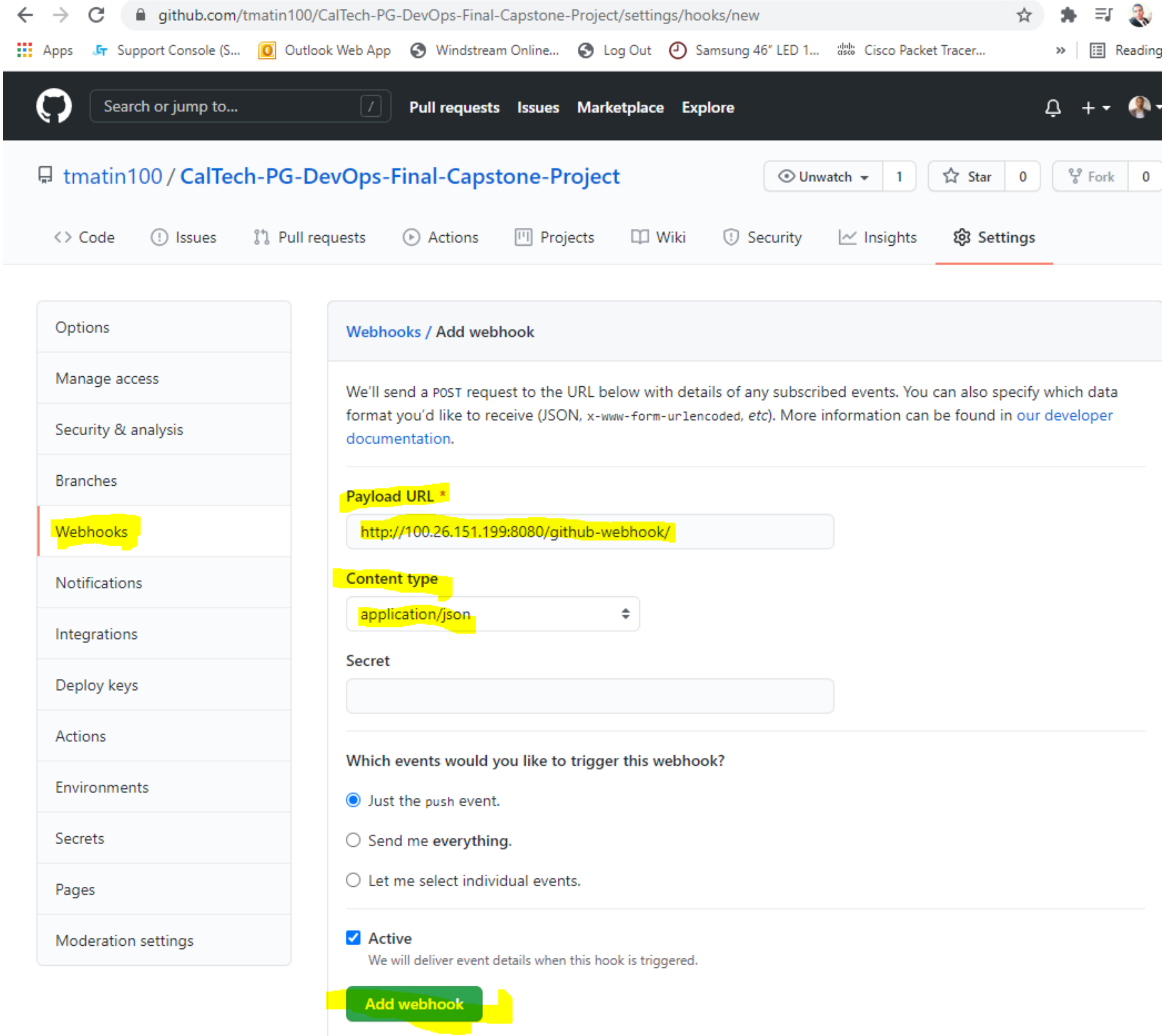
Setup a webhook on the GitHub repository so when it detects a change it can trigger the Jenkins job. Go to the **Settings** on top right corner of the repository.



tmatin100 Update inventory.txt 3785ead 4 minutes ago 22 commits

dist	Add files via upload	1 hour ago
src/main/webapp	Update index.jsp	17 minutes ago
Dockerfile	Update Dockerfile	11 minutes ago
Jenkinsfile	Add files via upload	1 hour ago

Provide the Jenkins Server's **public ip address** followed by **/github-webhook/,** and click the **Add webhook** button.



github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project/settings/hooks/new

tmatin100 / CalTech-PG-DevOps-Final-Capstone-Project

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, *etc*). More information can be found in our developer documentation.

Payload URL *

http://100.26.151.199:8080/github-webhook/

Content type

application/json

Secret

Which events would you like to trigger this webhook?

● Just the push event.

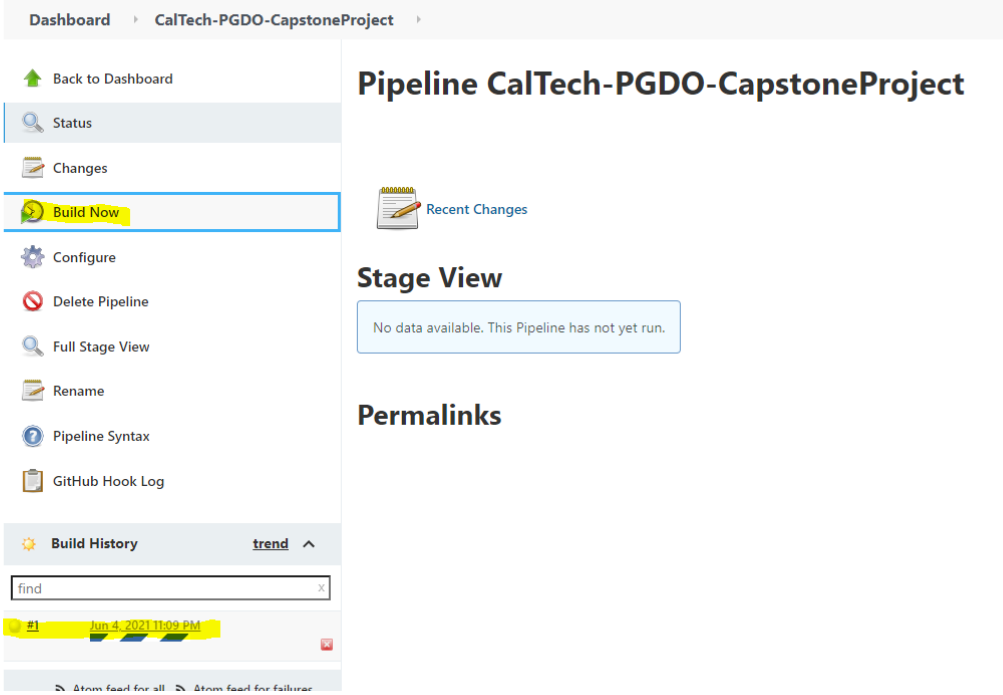
○ Send me **everything**.

○ Let me select individual events.

☑ Active

We will deliver event details when this hook is triggered.

Go to the Jenkins job and, click “**Build Now**” and wait for the build to start.



Pipeline CalTech-PGDO-CapstoneProject

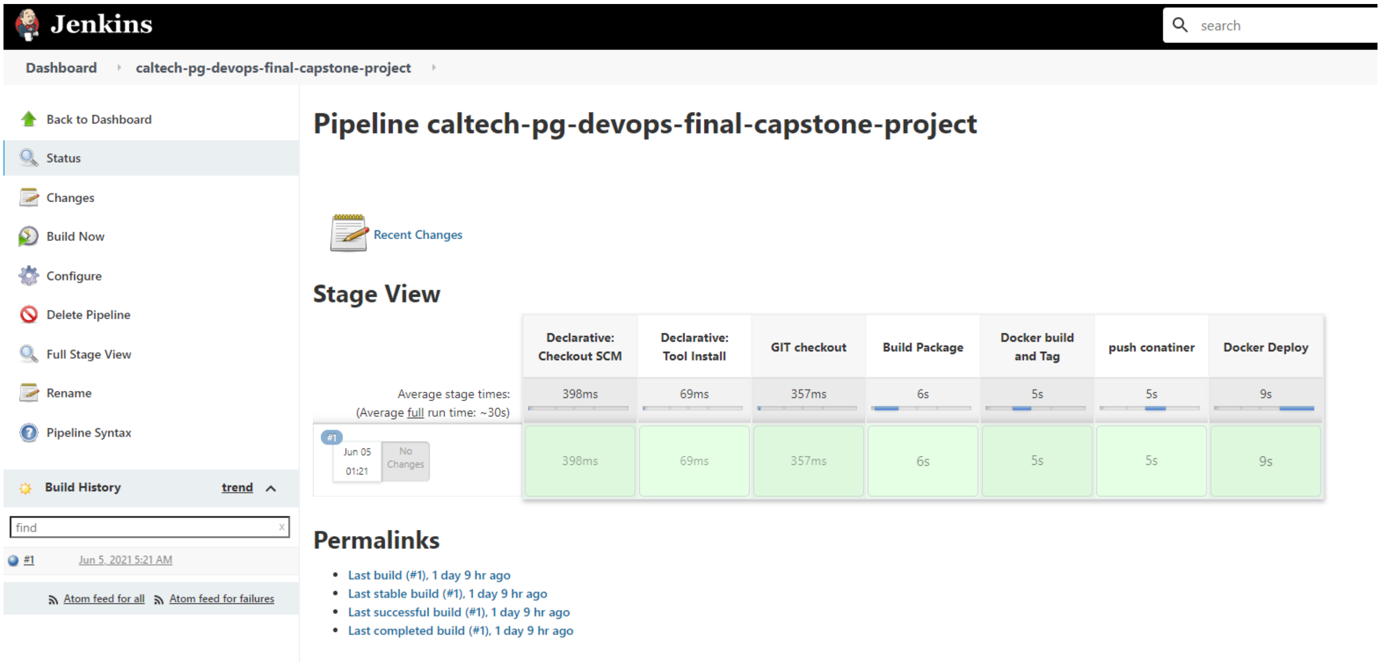
Recent Changes

Stage View

No data available. This Pipeline has not yet run.

Permalinks

You should be able to get a successful build and each of it’s stages defined in the Jenksinsfile.



Pipeline caltech-pg-devops-final-capstone-project

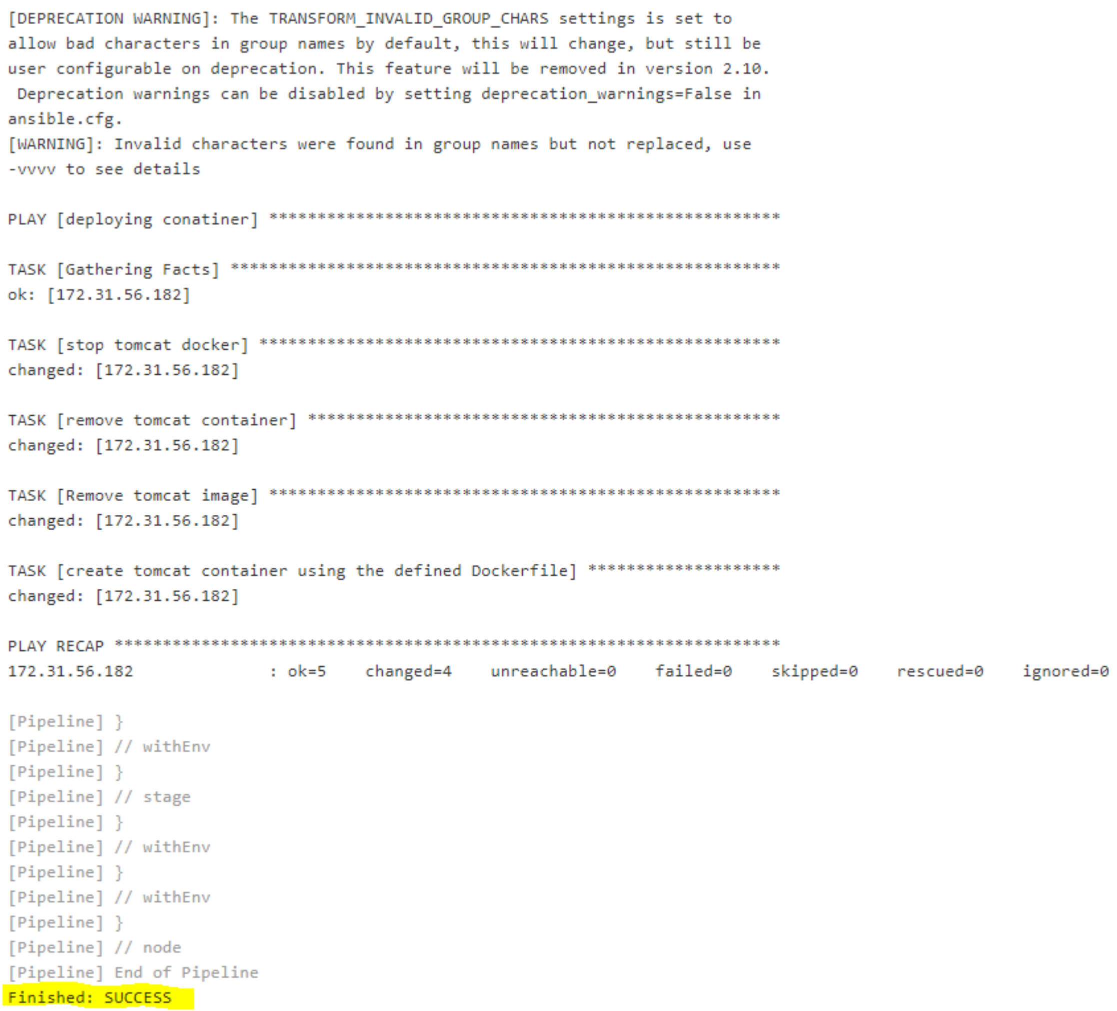
Recent Changes

Stage View

	Declarative: Checkout SCM	Declarative: Tool Install	GIT checkout	Build Package	Docker build and Tag	push conatiner	Docker Deploy
Average stage times: (Average full run time: ~30s)	398ms	69ms	357ms	6s	5s	5s	9s
#1 Jun 05 01:21 No Changes	398ms	69ms	357ms	6s	5s	5s	9s

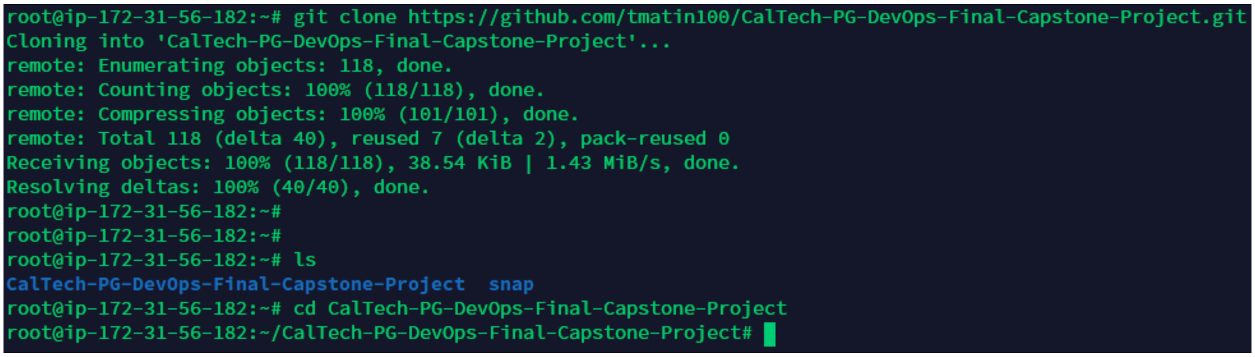
Permalinks

- Last build (#1), 1 day 9 hr ago
- Last stable build (#1), 1 day 9 hr ago
- Last successful build (#1), 1 day 9 hr ago
- Last completed build (#1), 1 day 9 hr ago



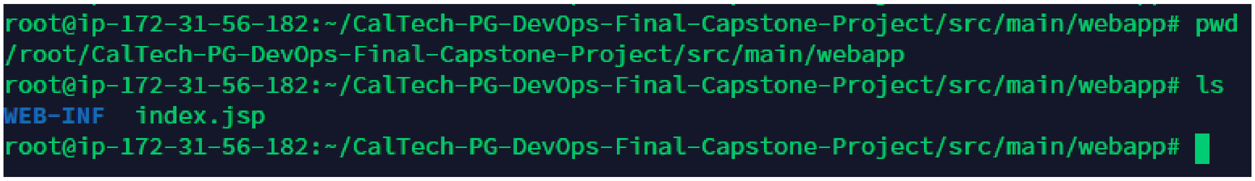
```
[DEPRECATION WARNING]: The TRANSFORM_INVALID_GROUP_CHARS settings is set to
allow bad characters in group names by default, this will change, but still be
user configurable on deprecation. This feature will be removed in version 2.10.
 Deprecation warnings can be disabled by setting deprecation_warnings=False in
ansible.cfg.
[WARNING]: Invalid characters were found in group names but not replaced, use
-vvvv to see details

PLAY [deploying conatiner] *****************************************************

TASK [Gathering Facts] *********************************************************
ok: [172.31.56.182]

TASK [stop tomcat docker] ******************************************************
changed: [172.31.56.182]

TASK [remove tomcat container] *************************************************
changed: [172.31.56.182]

TASK [Remove tomcat image] *****************************************************
changed: [172.31.56.182]

TASK [create tomcat container using the defined Dockerfile] ********************
changed: [172.31.56.182]

PLAY RECAP *********************************************************************
172.31.56.182              : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Let’s check test our automated CI/CD. First lets clone the repo into our local machine.
git clone https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project.git

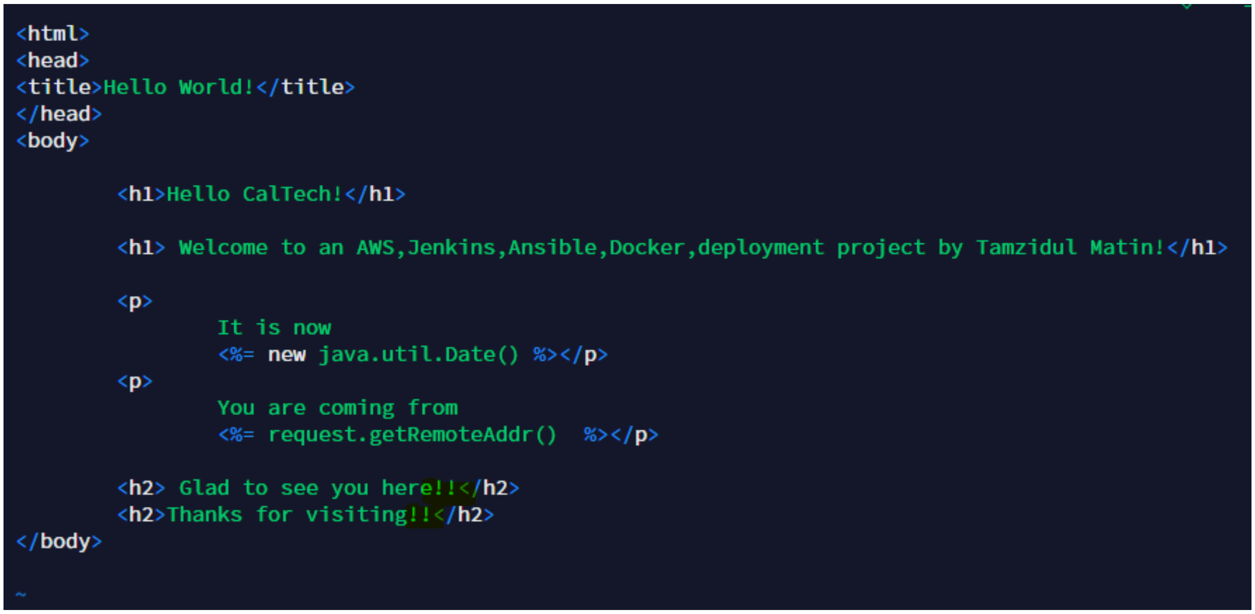


```
root@ip-172-31-56-182:~# git clone https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project.git
Cloning into 'CalTech-PG-DevOps-Final-Capstone-Project'...
remote: Enumerating objects: 118, done.
remote: Counting objects: 100% (118/118), done.
remote: Compressing objects: 100% (101/101), done.
remote: Total 118 (delta 40), reused 7 (delta 2), pack-reused 0
Receiving objects: 100% (118/118), 38.54 KiB | 1.43 MiB/s, done.
Resolving deltas: 100% (40/40), done.
root@ip-172-31-56-182:~#
root@ip-172-31-56-182:~#
root@ip-172-31-56-182:~# ls
CalTech-PG-DevOps-Final-Capstone-Project  snap
root@ip-172-31-56-182:~# cd CalTech-PG-DevOps-Final-Capstone-Project
root@ip-172-31-56-182:~/CalTech-PG-DevOps-Final-Capstone-Project#
```

Go into the repository, cd into this directory **/src/main/webapp,** make some changes to the index.jsp file and save.



```
root@ip-172-31-56-182:~/CalTech-PG-DevOps-Final-Capstone-Project/src/main/webapp# pwd
/root/CalTech-PG-DevOps-Final-Capstone-Project/src/main/webapp
root@ip-172-31-56-182:~/CalTech-PG-DevOps-Final-Capstone-Project/src/main/webapp# ls
WEB-INF  index.jsp
root@ip-172-31-56-182:~/CalTech-PG-DevOps-Final-Capstone-Project/src/main/webapp# 
```



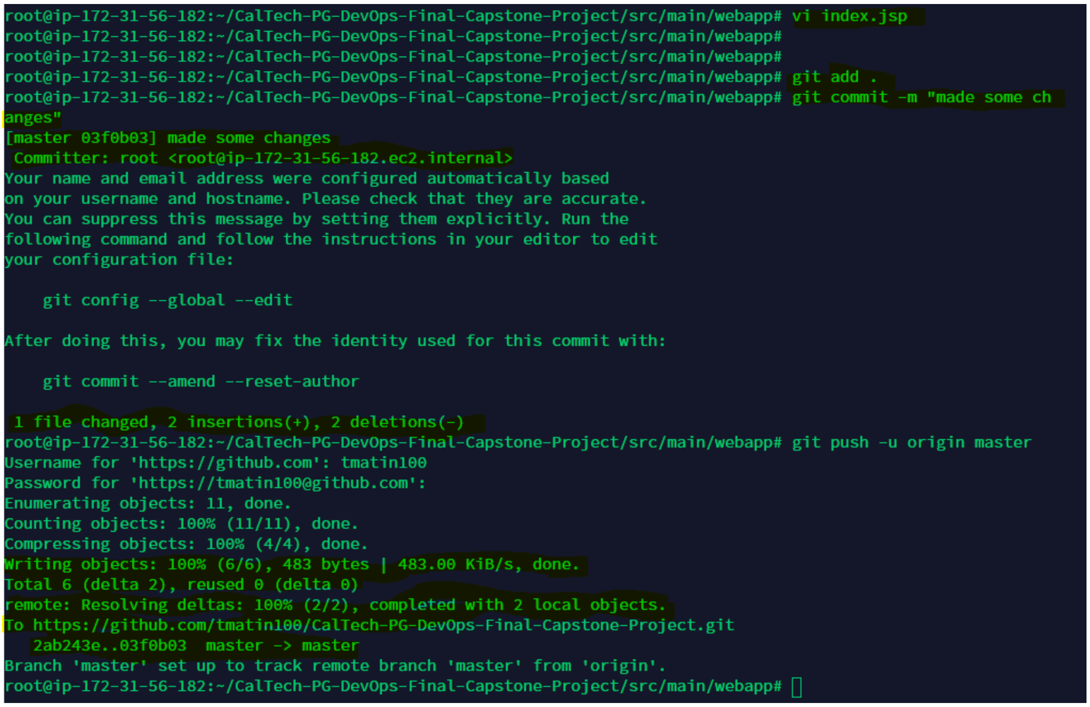
```
<html>
<head>
<title>Hello World!</title>
</head>
<body>

        <h1>Hello CalTech!</h1>

        <h1> Welcome to an AWS,Jenkins,Ansible,Docker,deployment project by Tamzidul Matin!</h1>

        <p>
                It is now
                <%= new java.util.Date() %></p>
        <p>
                You are coming from
                <%= request.getRemoteAddr()  %></p>

        <h2> Glad to see you here!!</h2>
        <h2>Thanks for visiting!!</h2>
</body>

~
```

Commit the changes and push the new code to the Github repsitory.

git add .

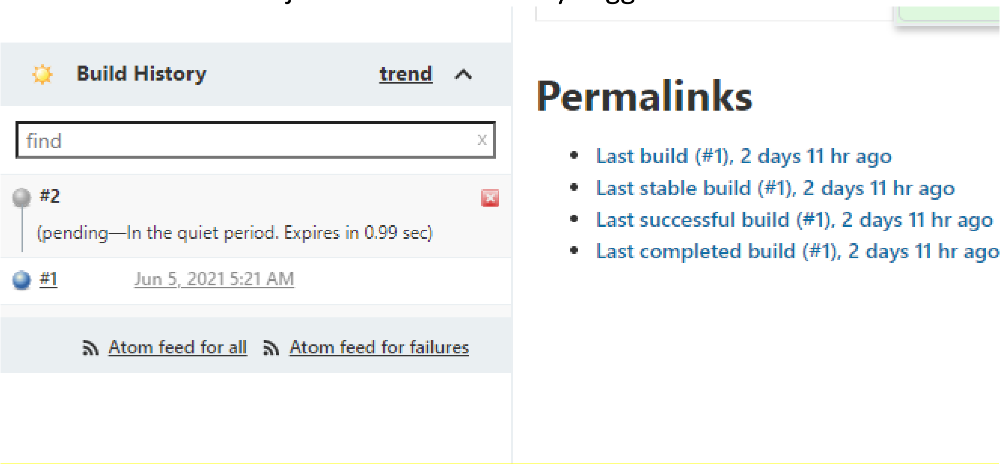
git commit -m “made some changes”

git push -u origin master



```
root@ip-172-31-56-182:~/CalTech-PG-DevOps-Final-Capstone-Project/src/main/webapp# vi index.jsp
root@ip-172-31-56-182:~/CalTech-PG-DevOps-Final-Capstone-Project/src/main/webapp#
root@ip-172-31-56-182:~/CalTech-PG-DevOps-Final-Capstone-Project/src/main/webapp#
root@ip-172-31-56-182:~/CalTech-PG-DevOps-Final-Capstone-Project/src/main/webapp# git add .
root@ip-172-31-56-182:~/CalTech-PG-DevOps-Final-Capstone-Project/src/main/webapp# git commit -m "made some ch
anges"
[master 03f0b03] made some changes
 Committer: root <root@ip-172-31-56-182.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 2 insertions(+), 2 deletions(-)
root@ip-172-31-56-182:~/CalTech-PG-DevOps-Final-Capstone-Project/src/main/webapp# git push -u origin master
Username for 'https://github.com': tmatin100
Password for 'https://tmatin100@github.com':
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 483 bytes | 483.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project.git
   2ab243e..03f0b03  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
root@ip-172-31-56-182:~/CalTech-PG-DevOps-Final-Capstone-Project/src/main/webapp# 
```

You can see a Jenkins job was automatically triggered.



Build History trend

find

#2

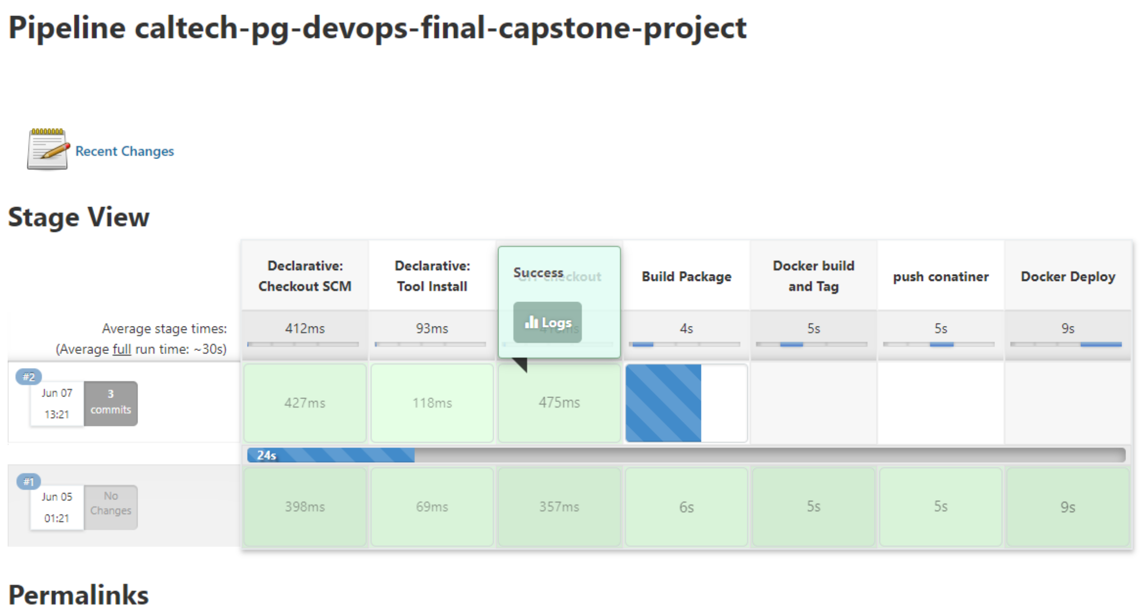
(pending—In the quiet period. Expires in 0.99 sec)

#1 Jun 5, 2021 5:21 AM

Atom feed for all Atom feed for failures

Permalinks

- Last build (#1), 2 days 11 hr ago
- Last stable build (#1), 2 days 11 hr ago
- Last successful build (#1), 2 days 11 hr ago
- Last completed build (#1), 2 days 11 hr ago

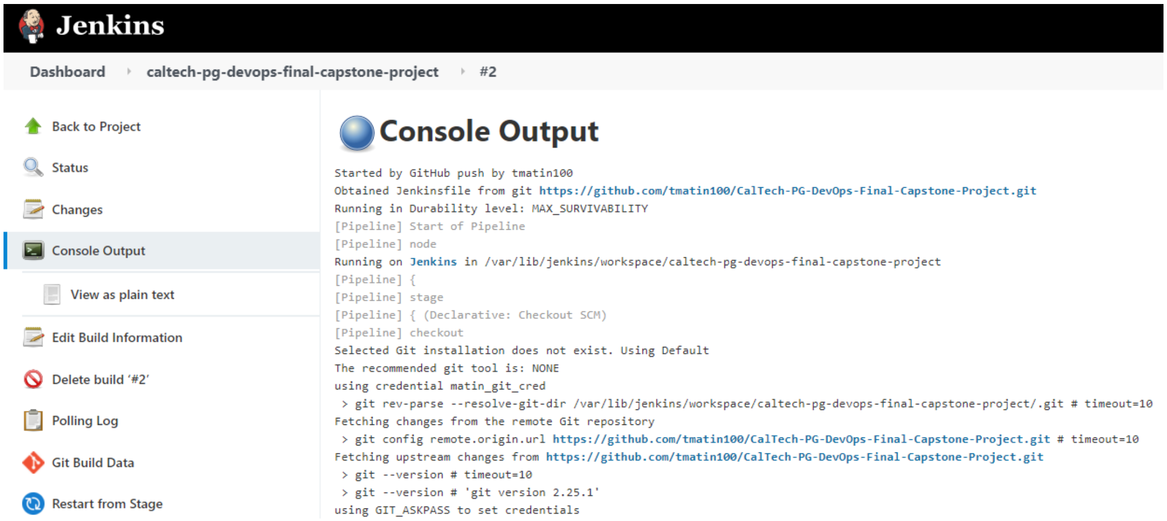


Pipeline caltech-pg-devops-final-capstone-project

Stage View

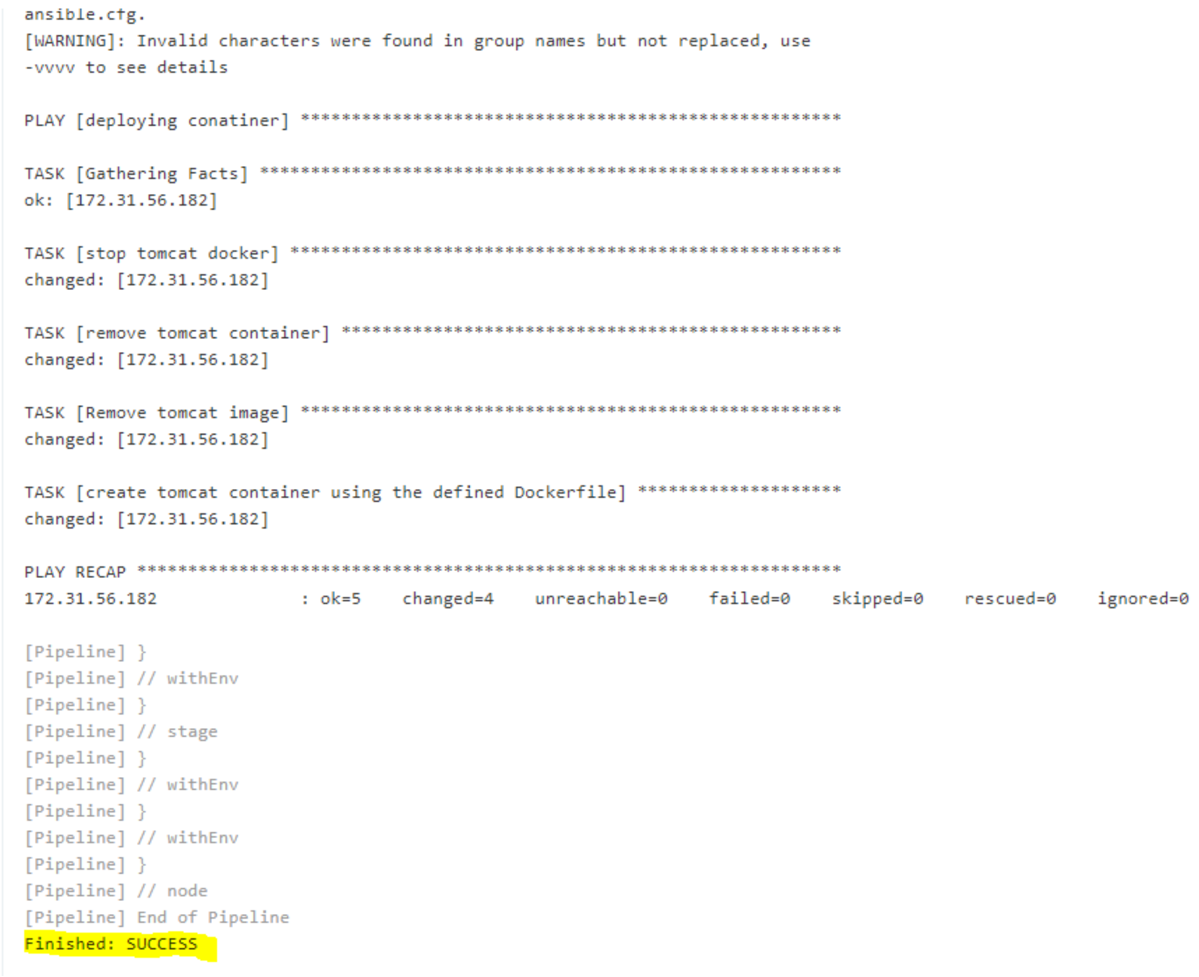
	Declarative: Checkout SCM	Declarative: Tool Install	Git Checkout	Build Package	Docker build and Tag	push conatiner	Docker Deploy
Average stage times: (Average full run time: ~30s)	412ms	93ms	[illegible]	4s	5s	5s	9s
#2 Jun 07 13:21 3 commits	427ms	118ms	475ms				
#1 Jun 05 01:21 No Changes	398ms	69ms	357ms	6s	5s	5s	9s

Permalinks



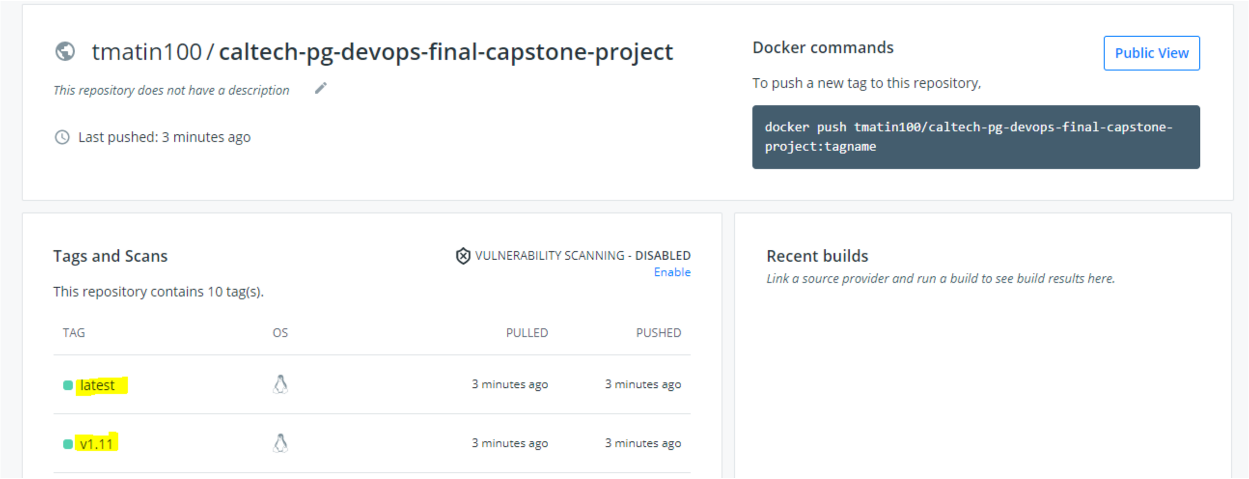
Console Output

```
Started by GitHub push by tmatin100
Obtained Jenkinsfile from git https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project.git
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/caltech-pg-devops-final-capstone-project
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential matin_git_cred
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/caltech-pg-devops-final-capstone-project/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project.git # timeout=10
Fetching upstream changes from https://github.com/tmatin100/CalTech-PG-DevOps-Final-Capstone-Project.git
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
using GIT_ASKPASS to set credentials
```



```
ansible.cfg.
[WARNING]: Invalid characters were found in group names but not replaced, use
-vvvv to see details

PLAY [deploying conatiner] *****************************************************

TASK [Gathering Facts] *********************************************************
ok: [172.31.56.182]

TASK [stop tomcat docker] ******************************************************
changed: [172.31.56.182]

TASK [remove tomcat container] *************************************************
changed: [172.31.56.182]

TASK [Remove tomcat image] *****************************************************
changed: [172.31.56.182]

TASK [create tomcat container using the defined Dockerfile] ********************
changed: [172.31.56.182]

PLAY RECAP *********************************************************************
172.31.56.182              : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

You can see that the newly built image was pushed to the Docker hub prior to deployment on the Ansible host.



tmatin100 / caltech-pg-devops-final-capstone-project

This repository does not have a description

Last pushed: 3 minutes ago

Docker commands

To push a new tag to this repository,

```
docker push tmatin100/caltech-pg-devops-final-capstone-project:tagname
```

Tags and Scans

VULNERABILITY SCANNING - DISABLED

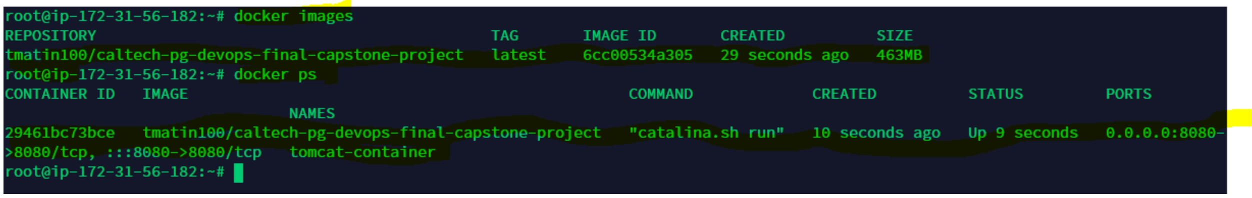
This repository contains 10 tag(s).

TAG	OS	PULLED	PUSHED
latest	🐧	3 minutes ago	3 minutes ago
v1.11	🐧	3 minutes ago	3 minutes ago

Recent builds

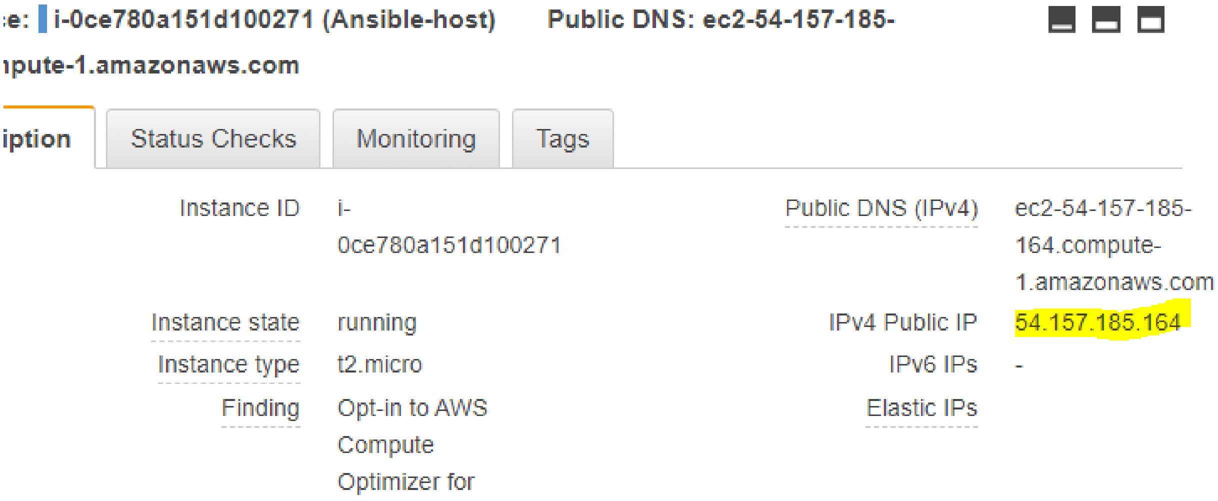
Link a source provider and run a build to see build results here.

Docker container was created and running on the defined ansible host machine, in the inventory file.



```
root@ip-172-31-56-182:~# docker images
REPOSITORY                                          TAG       IMAGE ID       CREATED          SIZE
tmatin100/caltech-pg-devops-final-capstone-project   latest    6cc00534a305   29 seconds ago   463MB
root@ip-172-31-56-182:~# docker ps
CONTAINER ID   IMAGE                                                COMMAND             CREATED          STATUS         PORTS
                              NAMES
29461bc73bce   tmatin100/caltech-pg-devops-final-capstone-project   "catalina.sh run"   10 seconds ago   Up 9 seconds   0.0.0.0:8080-
>8080/tcp, :::8080->8080/tcp   tomcat-container
root@ip-172-31-56-182:~# 
```

Check the public ip of the Ansible ec2 host in a browser to verify deployment is working.



e: i-0ce780a151d100271 (Ansible-host)   Public DNS: ec2-54-157-185-
ipute-1.amazonaws.com

Instance ID i-0ce780a151d100271

Public DNS (IPv4) ec2-54-157-185-164.compute-1.amazonaws.com

Instance state running

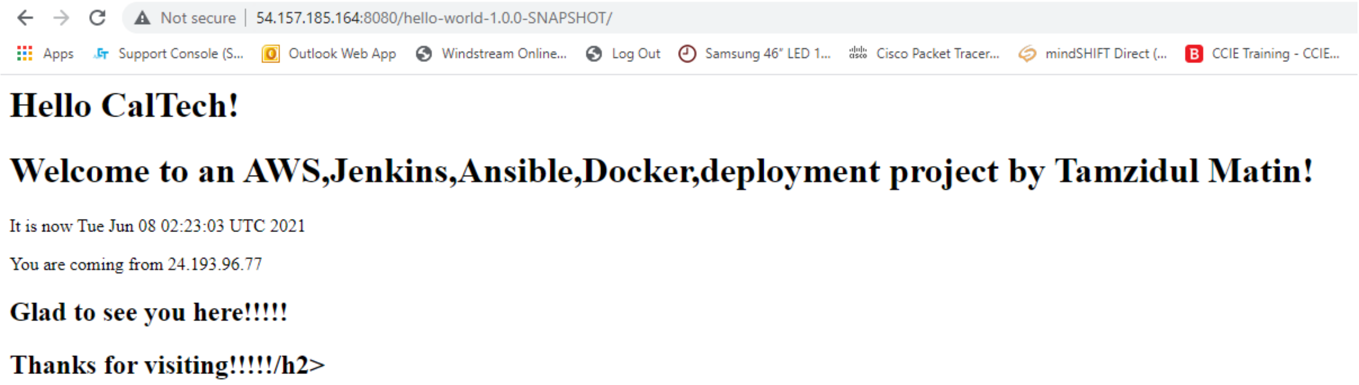
IPv4 Public IP 54.157.185.164

Instance type t2.micro

IPv6 IPs -

Finding Opt-in to AWS Compute Optimizer for

Elastic IPs



Not secure | 54.157.185.164:8080/hello-world-1.0.0-SNAPSHOT/

Hello CalTech!

Welcome to an AWS,Jenkins,Ansible,Docker,deployment project by Tamzidul Matin!

It is now Tue Jun 08 02:23:03 UTC 2021

You are coming from 24.193.96.77

Glad to see you here!!!!!

Thanks for visiting!!!!!/h2>