**Step 1:**

Before installing Docker, you first have to ensure that you have the right Linux kernel version running. Docker is only designed to run on Linux kernel version 3.8 and higher.

$ uname -a

Linux host 4.1.12-61.1.28.x86\_64 #2 SMP Thu Feb 23 19:55:12 PST 2017 x86\_64 x86\_64 x86\_64 GNU/Linux

From the output, we can see that the Linux kernel version is 4.1 which is higher than version 3.8, so we are good to go.

**Step 2 (Optional):**

If the system is under corporate proxy, then export proxy details

**Step 3:**

Prerequisite for Docker

$ cd /etc/yum.repos.d/

$ more docker.repo

[dockerrepo]

name=Docker Repository

baseurl=https://yum.dockerproject.org/repo/main

enabled=1

gpgcheck=1

gpgkey=https://yum.dockerproject.org/gpg

**Step 4:**

Install Docker and verify

$ yum install docker-engine

$ systemctl daemon-reload

$ service docker status

Note: can use start, stop, etc.

$ docker run hello-world

Hello from Docker!

This message shows that your installation appears to be working correctly.

…

$ docker version

Client:

Version: 17.03.1-ce

…

$ docker info

Containers: 1

Running: 0

Paused: 0

Stopped: 1

Images: 3

Server Version: 17.03.1-ce

**Step 5:**

Run Jenkins container in Docker

$ docker pull jenkins

Check docker images

$ docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

jenkins/jenkins lts a15183f6519b 2 weeks ago 700 MB

jenkins latest 07b4164f9789 2 weeks ago 696 MB

hello-world latest e38bc07ac18e 2 months ago 1.85 kB

jenkins/jenkins 2.73.1 c8a24e6775ea 9 months ago 814 MB

Create a directory for Jenkins and set to defined ownership

$ mkdir /var/jenkins\_home

$ cd /var/

$ ls –larth

…

drwxr-xr-x. 1 root root 0 Jun 26 13:22 jenkins\_home

…

$ chown 1000:1000 jenkins\_home

$ ls –larth

…

drwxr-xr-x. 1 sumit group 0 Jun 26 13:22 jenkins\_home

…

Run Jenkins container in background

$ docker run \

-d \

-p 8080:8080 \

-p 50000:50000 \

-v /var/jenkins\_home:/var/jenkins\_home \

--name sumit\_jenkins \

jenkins

…

cb5937ce08ab440916dad86ec58ab0502cc4d36d8032ab615f98f444847dbc7a

…

Check docker containers

$ docker ps –a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

cb5937ce08ab jenkins "/bin/tini -- /usr..." 2 minutes ago Up 2 minutes 0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp sumit\_jenkins

7a2a933c80c9 jenkins "/bin/tini -- /usr..." 23 hours ago Exited (143) 16 minutes ago jenkinslatest

2a9f96f3705d hello-world "/hello" 30 hours ago Exited (0) 30 hours ago priceless\_mestorf

Jenkins user credential

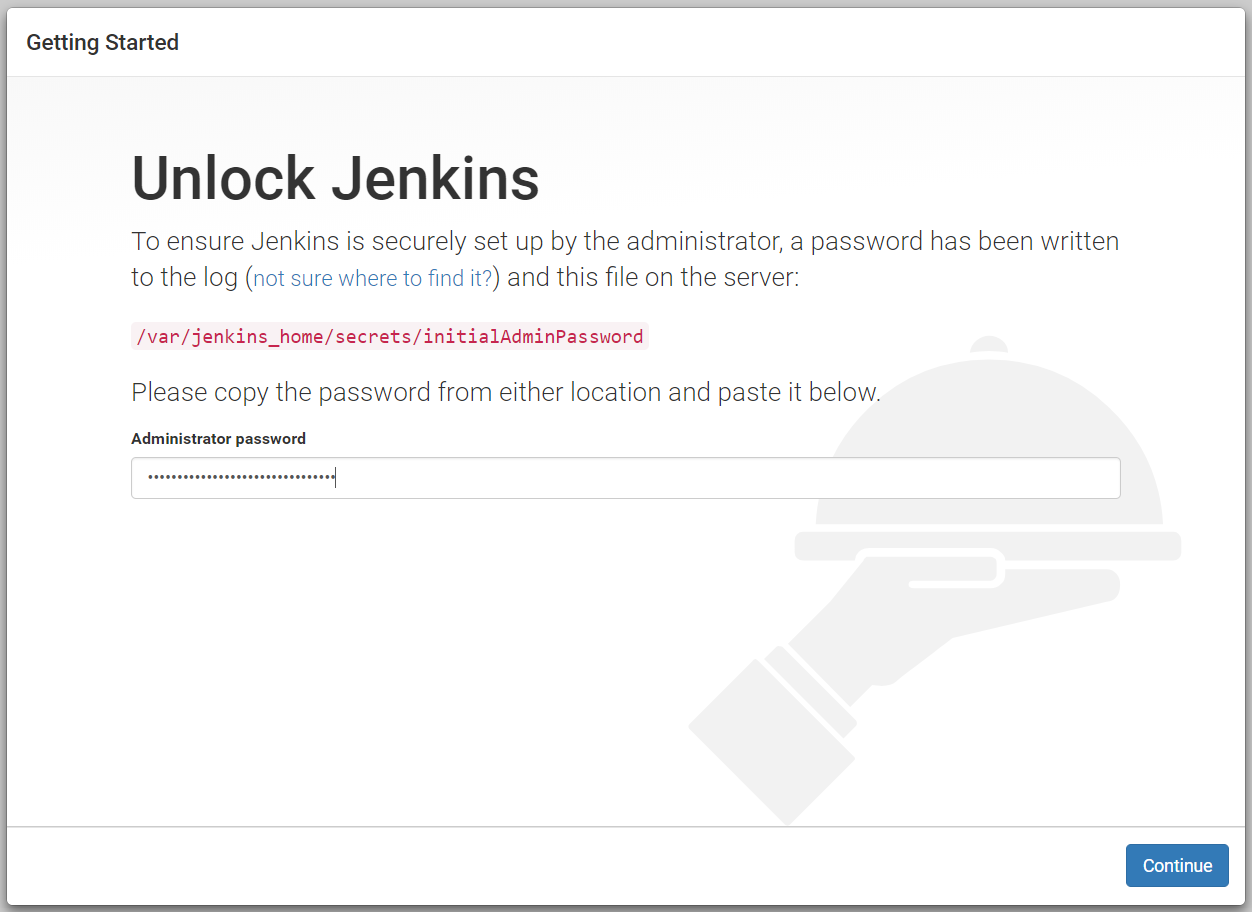
admin

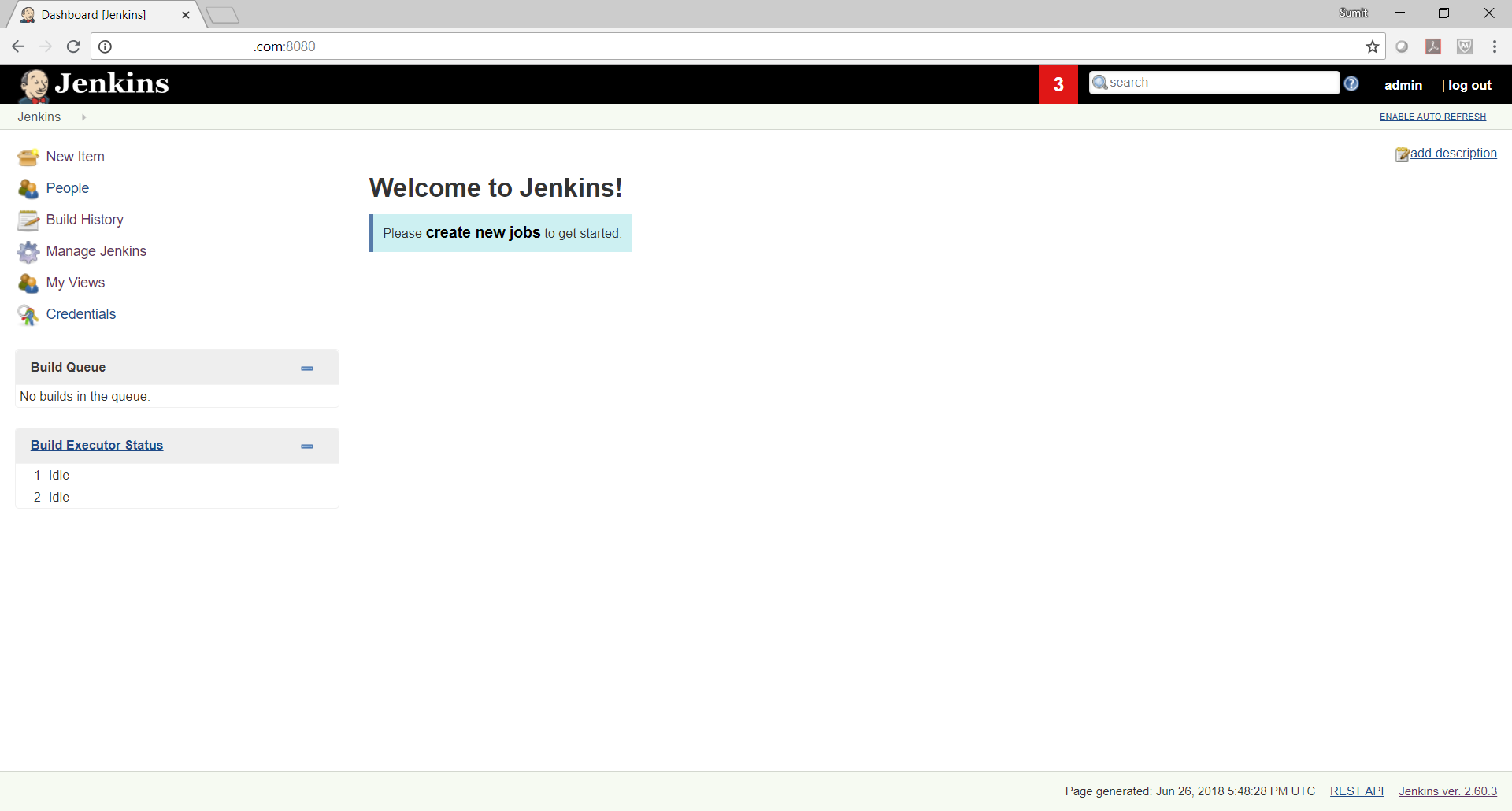
$ vi /var/jenkins\_home/secrets/initialAdminPassword

26398ce94a0547f5a8068238eb7f61ef

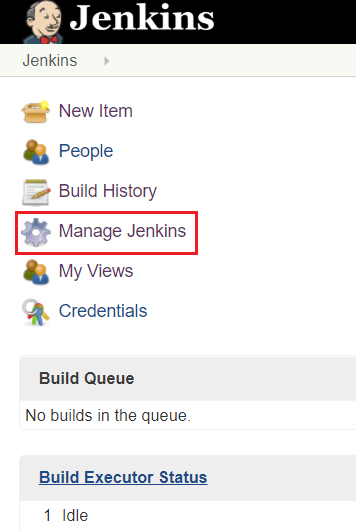
Access jenkins Web GUI

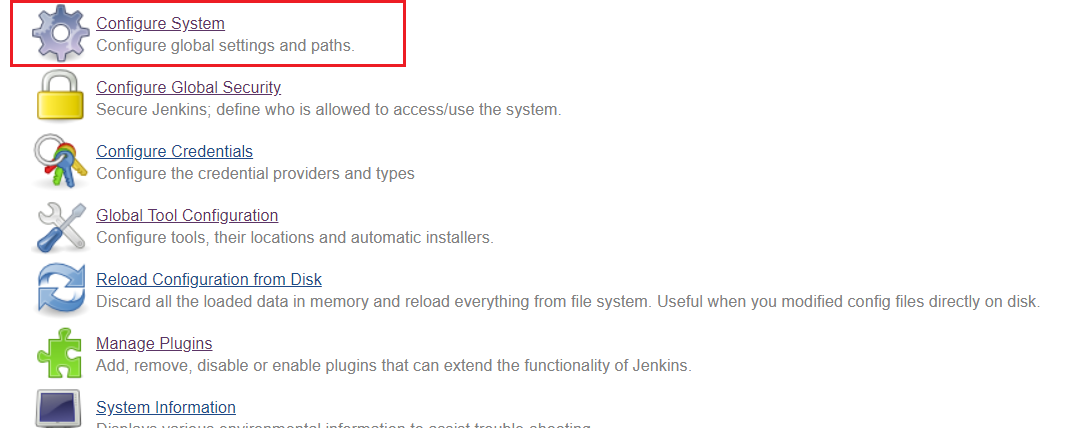
http://xxxx.com:8080/

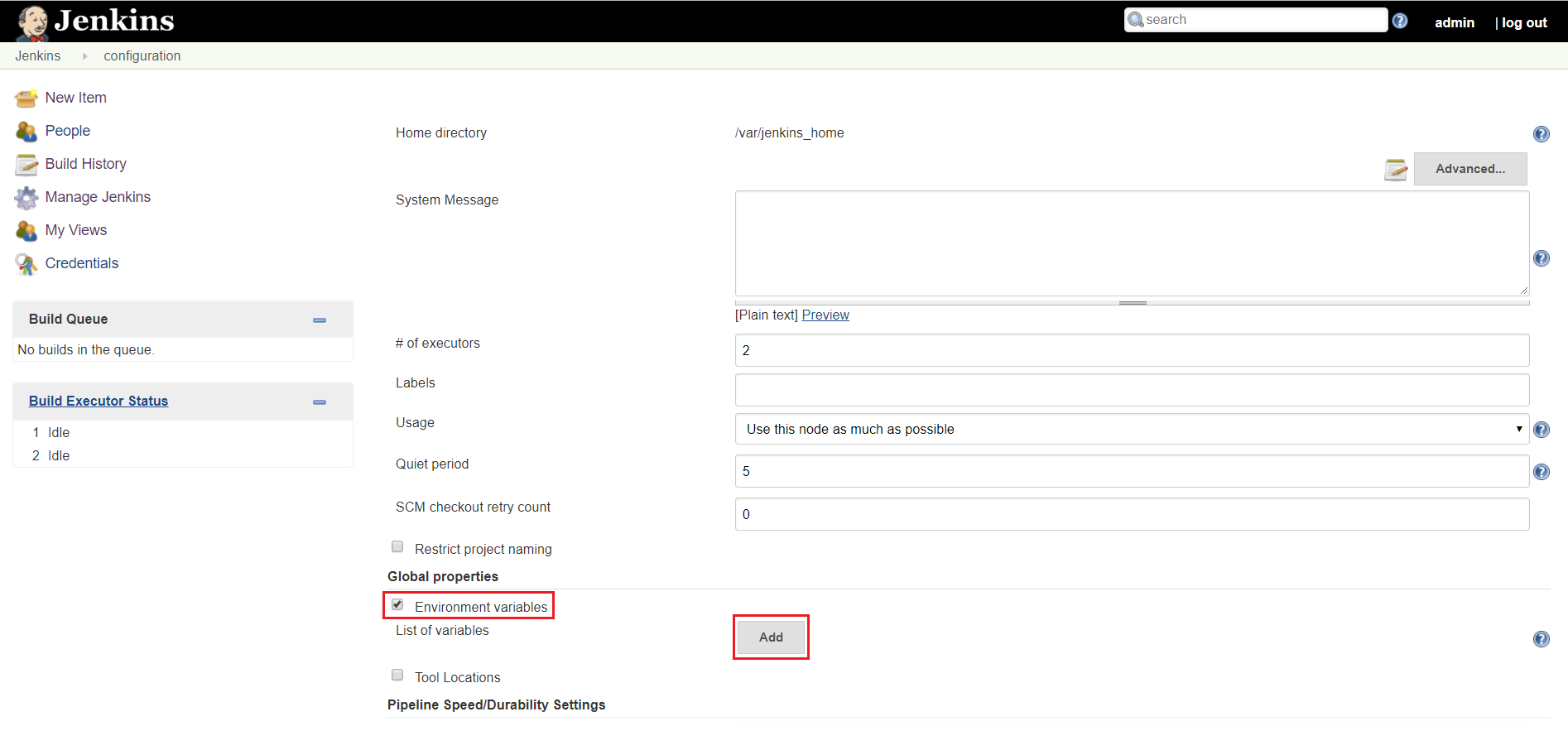


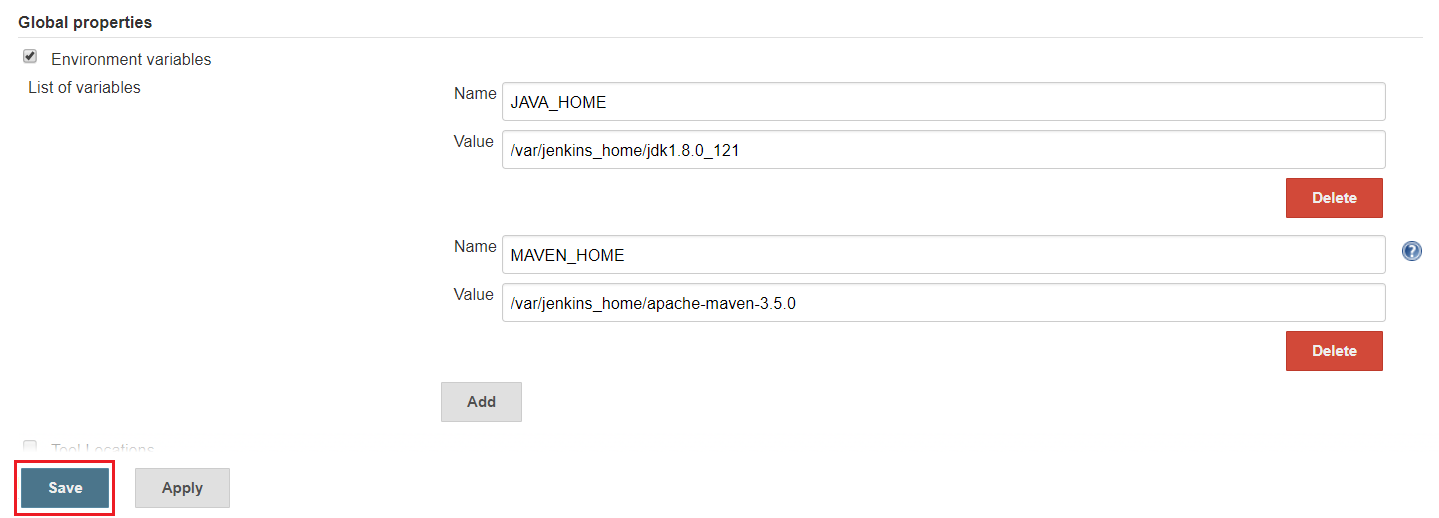


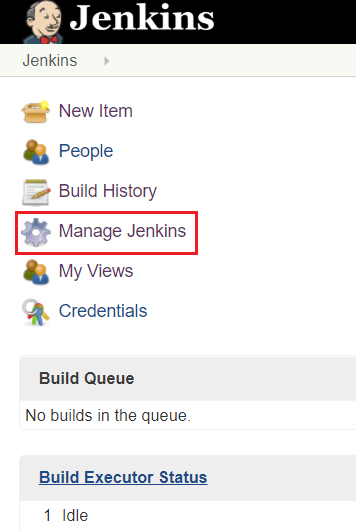
Setup Jenkins

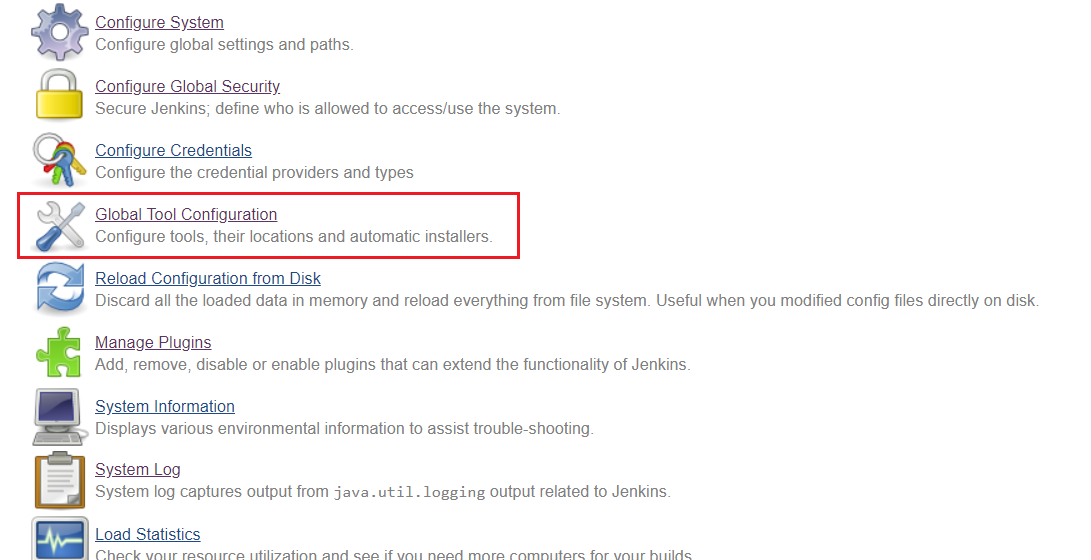


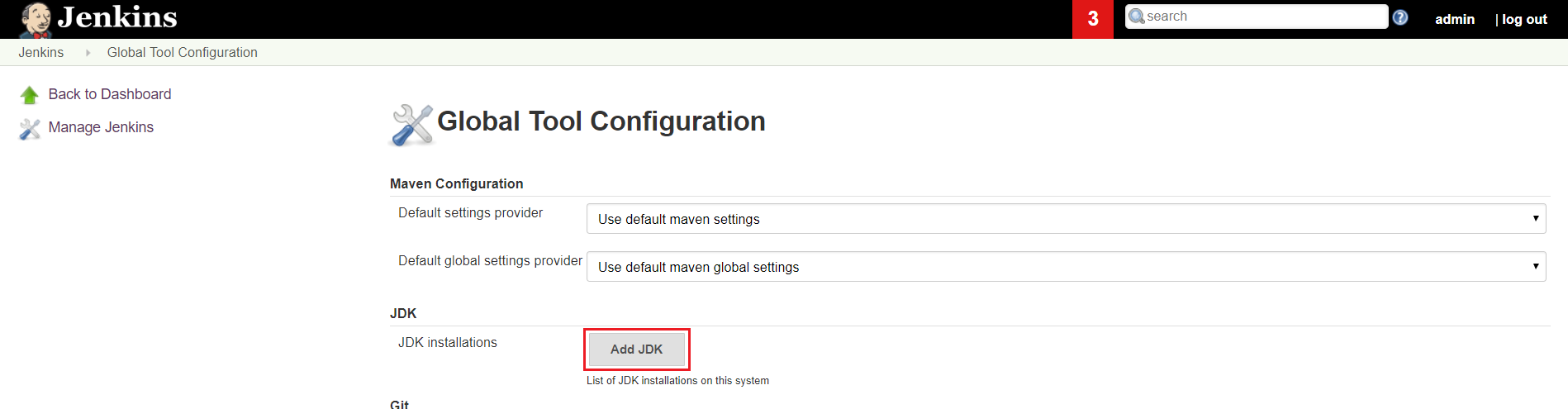


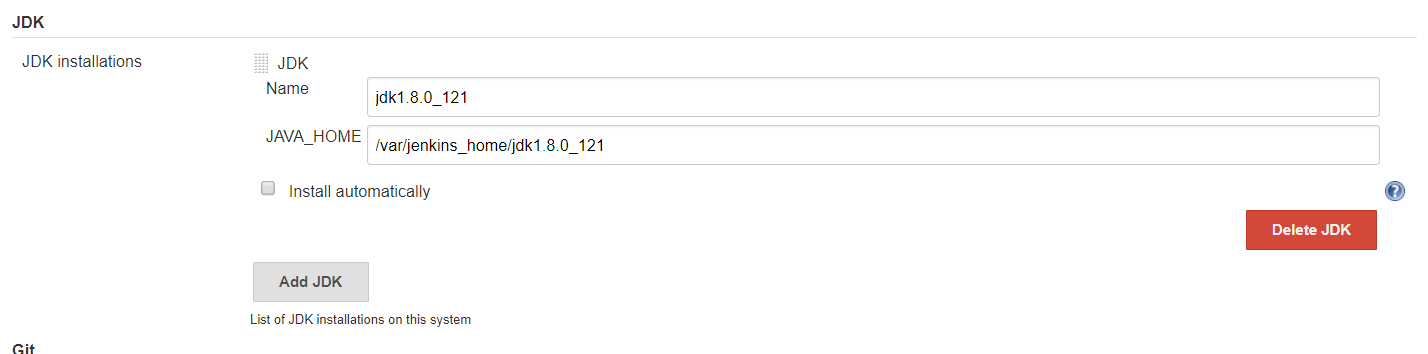


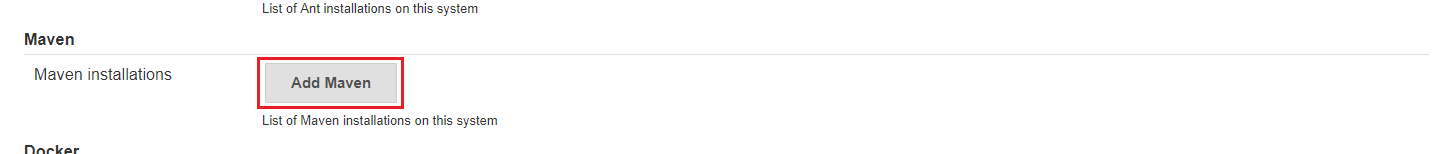


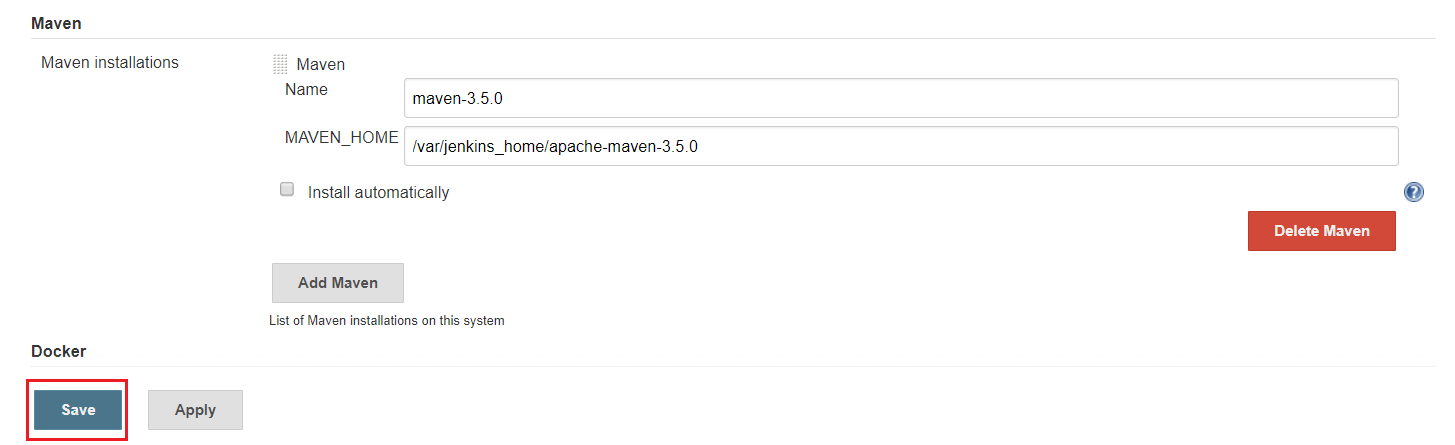






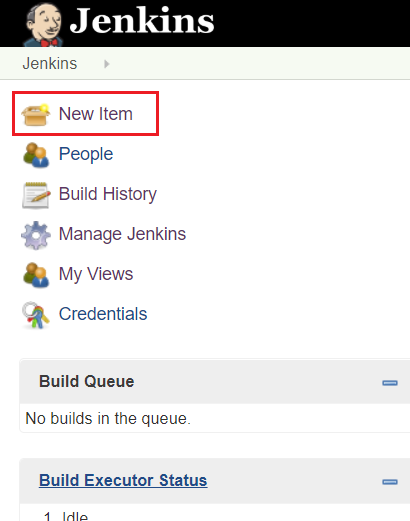


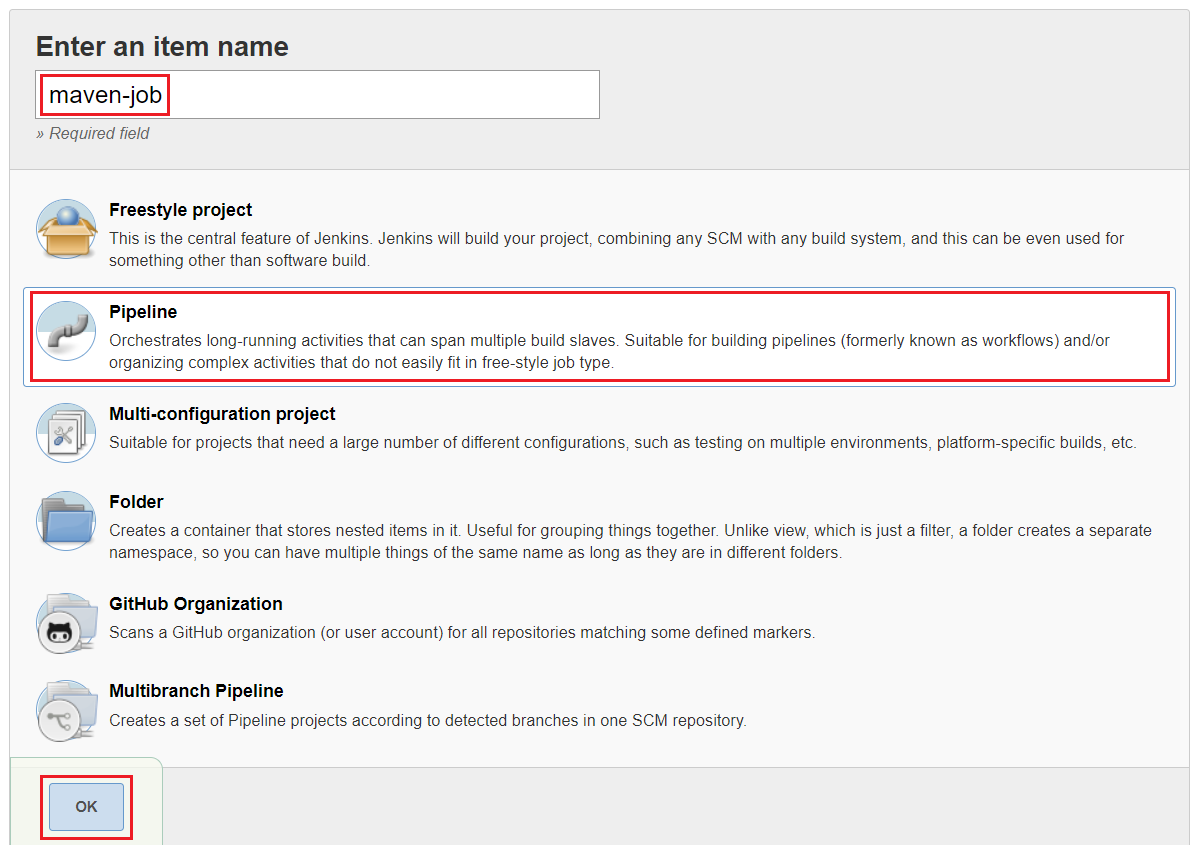


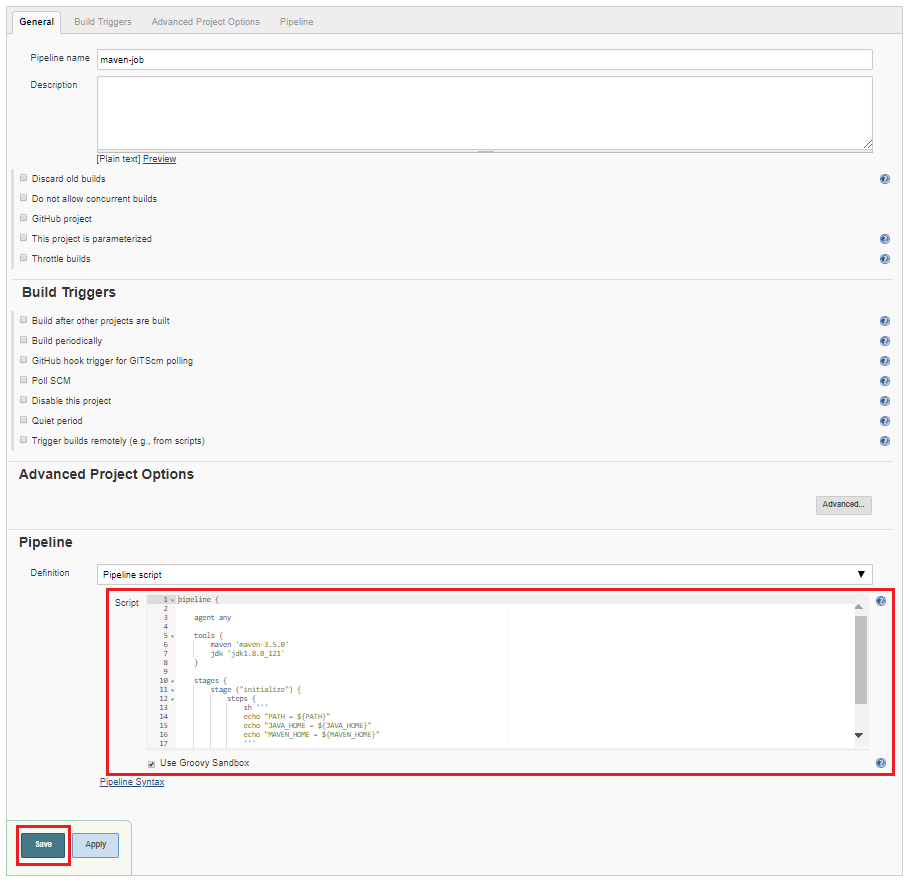


**Step 6:**

Create Jenkins job using pipeline and build







pipeline {

agent any

tools {

maven 'maven-3.5.0'

jdk 'jdk1.8.0\_121'

}

stages {

stage ("initialize") {

steps {

sh '''

echo "PATH = ${PATH}"

echo "JAVA\_HOME = ${JAVA\_HOME}"

echo "MAVEN\_HOME = ${MAVEN\_HOME}"

'''

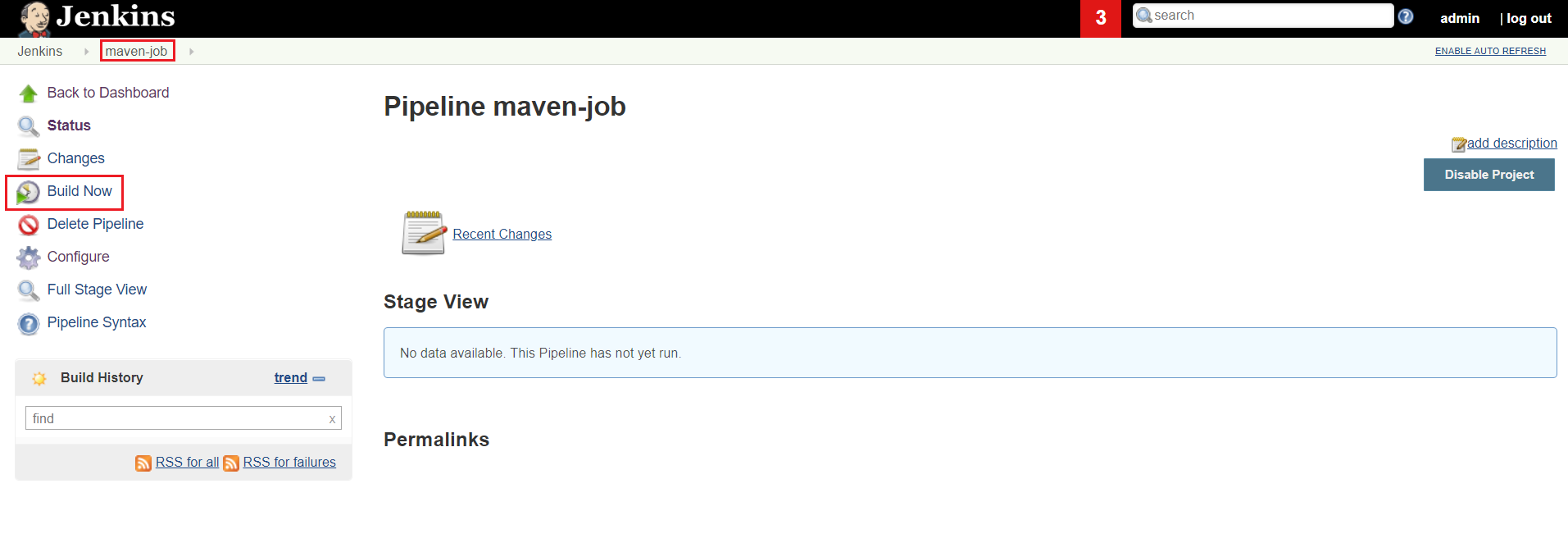
}

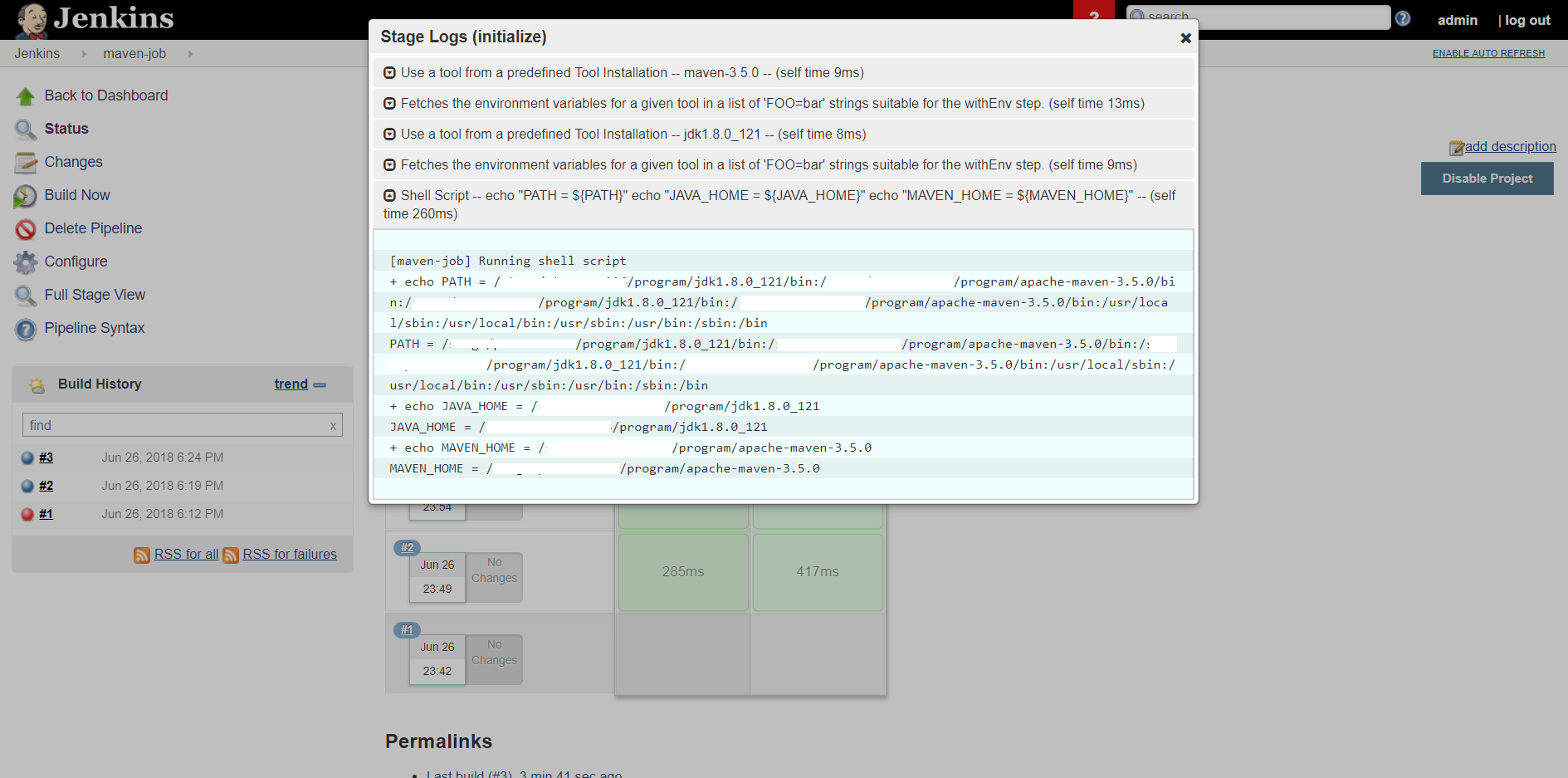
}

}

}

Build Jenkins job and view logs





**Step 7:**

Setup Maven

*Maven is a Java tool, so you must have Java installed in order to proceed.*

*It should print out your installed version of Maven*

$ mvn -v

Apache Maven 3.5.0 (ff8f5e7444045639af65f6095c62210b5713f426; 2017-04-03T15:39:06-04:00)

Maven home: /stage/xxxx/program/apache-maven-3.5.0

Java version: 1.8.0\_121, vendor: Java

Java home: /stage/xxxxx/program/jdk1.8.0\_121/jre

Default locale: en\_US, platform encoding: UTF-8

OS name: "linux", version: "4.1.12-61.1.28.el7uek.x86\_64", arch: "amd64", family: "unix"

*you may require extra configuration*

$ pwd

/home/sumit/.m2

$ vi settings.xml

…

<settings>

<proxies>

<proxy>

<id>xxxx-proxy</id>

<active>true</active>

<protocol>http</protocol>

<host>xxxx.com</host>

<port>80</port>

<nonProxyHosts>www.google.com|\*.example.com</nonProxyHosts>

</proxy>

</proxies>

</settings>

…

*You will need somewhere for your project to reside, create a directory somewhere and start a shell in that directory. On your command line, execute the following Maven goal*

$ pwd

/home/sumit/maven\_project

$ mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

…

[INFO] Parameter: basedir, Value: /home/sumit/maven\_project

[INFO] Parameter: package, Value: com.mycompany.app

[INFO] Parameter: groupId, Value: com.mycompany.app

[INFO] Parameter: artifactId, Value: my-app

[INFO] Parameter: packageName, Value: com.mycompany.app

[INFO] Parameter: version, Value: 1.0-SNAPSHOT

[INFO] project created from Old (1.x) Archetype in dir: /home/sumit/maven\_project/my-app

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 02:12 min

[INFO] Finished at: 2018-06-26T02:48:25-04:00

[INFO] Final Memory: 19M/316M

[INFO] ------------------------------------------------------------------------

…

*The src/main/java directory contains the project source code, the src/test/java directory contains the test source, and the pom.xml file is the project's Project Object Model, or POM*

$ pwd

/home/sumit/maven\_project/my-app

*Build the Project*

$ mvn package

…

-------------------------------------------------------

T E S T S

-------------------------------------------------------

Running com.mycompany.app.AppTest

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

…

[INFO] Building jar: /home/sumit/maven\_project/my-app/target/my-app-1.0-SNAPSHOT.jar

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 2.807 s

[INFO] Finished at: 2018-06-26T02:52:26-04:00

[INFO] Final Memory: 20M/310M

[INFO] ------------------------------------------------------------------------

…

*You may test the newly compiled and packaged JAR with the following command*

$ java -cp target/my-app-1.0-SNAPSHOT.jar com.mycompany.app.App

Hello World!

*To only test the maven project*

$ mvn test

…

[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ my-app ---

[INFO] Surefire report directory: /home/sumibisw/maven\_project/my-app/target/surefire-reports

-------------------------------------------------------

T E S T S

-------------------------------------------------------

Running com.mycompany.app.AppTest

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.006 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 1.059 s

[INFO] Finished at: 2018-06-26T03:13:18-04:00

[INFO] Final Memory: 11M/312M

[INFO] ------------------------------------------------------------------------

…

$ pwd

/home/sumit/maven\_project/my-app/target/surefire-reports

$ vi com.mycompany.app.AppTest.txt

…

-------------------------------------------------------------------------------

Test set: com.mycompany.app.AppTest

-------------------------------------------------------------------------------

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.006 sec

…

**Step 8:**

Use maven in Jenkins pipeline to build and test java project

$ cp -a ~/.m2/ /var/jenkins\_home/

$ cd ~/maven\_project/my-app/

$ ls -larth

-rw-r--r--. 1 sumit group 642 Jun 26 02:48 pom.xml

drwxr-xr-x. 1 sumit group 16 Jun 26 02:48 src

$ cp -a pom.xml /var/jenkins\_home/workspace/maven-job

$ cp -a src /var/jenkins\_home/workspace/maven-job

pipeline {

agent any

tools {

maven 'maven-3.5.0'

jdk 'jdk1.8.0\_121'

}

stages {

stage ("Initialize") {

steps {

sh '''

echo "PATH = ${PATH}"

echo "JAVA\_HOME = ${JAVA\_HOME}"

echo "MAVEN\_HOME = ${MAVEN\_HOME}"

'''

}

}

stage ("Build") {

steps {

sh "mvn -v"

sh "mvn -B -DskipTests clean package"

}

}

stage('Test') {

steps {

sh 'mvn test'

}

post {

always {

junit 'target/surefire-reports/\*.xml'

}

}

}

}

}

Run build project

$ pwd

/var/jenkins\_home/workspace/maven-job/target

$ /var/jenkins\_home/jdk1.8.0\_121/bin/java -cp my-app-1.0-SNAPSHOT.jar com.mycompany.app.App

Hello World!

$

**Step 9:**

Setup Github

$ mkdir git\_project

$ cd git\_project/

$ pwd

/root/git\_project

$ git init

Initialized empty Git repository in /root/git\_project/.git/

$ git config --global user.name "sumit"

$ git config --global user.email sumit@example.com

$ echo "# demo" >> README.md

$ git add README.md

$ git commit -m "first commit"

[master (root-commit) e87f098] first commit

1 file changed, 1 insertion(+)

create mode 100644 README.md

$ git remote add origin https://github.com/sumitbiswasgit/demo.git

$ git\_project]# git push -u origin master

Username for 'https://github.com': sumit@gmail.com

Password for 'https://sumit@gmail.com@github.com':

Counting objects: 3, done.

Writing objects: 100% (3/3), 211 bytes | 0 bytes/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To https://github.com/sumitbiswasgit/demo.git

\* [new branch] master -> master

Branch master set up to track remote branch master from origin.

$

