

## Install and Configure Jenkins on AWS EC2

### Prerequisites

- Launch an AWS EC2 instance to host Jenkins
- Use PuTTY to log on the EC2 via SSH

### Install Java 8

Check Java current version : `java -version`

If you could see the 1.7 version you can uninstall and install the 1.8 using following commands,

```
[ec2-user ~]$ sudo yum remove java-1.7.0-openjdk
```

```
[ec2-user ~]$ sudo yum install java-1.8.0
```

### Download and Install Jenkins

1. To ensure that your software packages are up to date on your instance, use the following command to perform a quick software update:

```
[ec2-user ~]$ sudo yum update -y
```

2. Add the Jenkins repo using the following command:

```
[ec2-user ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat/jenkins.repo
```

3. Import a key file from Jenkins-CI to enable installation from the package:

```
[ec2-user ~]$ sudo rpm --import http://pkg.jenkins-ci.org/redhat/jenkins-ci.org.key
```

4. Install Jenkins:

```
[ec2-user ~]$ sudo yum install jenkins -y
```

5. Start Jenkins as a service.

**Note : Before running the Jenkins, make sure your 8080 port is available or else you could run Jenkins on any other available port by simply changing the port inside the configuration file of CentOS rpm based linux i.e. `/etc/sysconfig/jenkins` file (The location in debian based linux is `/var/default/jenkins`), change the port as `JENKINS_PORT="8081"`**

```
[ec2-user@ip-172-31-25-182 sysconfig]$ sudo cat jenkins | grep JENKINS
```

```
JENKINS_HOME="/var/lib/jenkins"
```

```
JENKINS_JAVA_CMD=""
```

```
# permissions of $JENKINS_HOME and /var/log/jenkins.
```

```
JENKINS_USER="jenkins"
```

```
# $JENKINS_HOME location. Do not enable this, "true", unless
```

```
# you know what you're doing. See JENKINS-23273.
```

```
#JENKINS_INSTALL_SKIP_CHOWN="false"
```

```
JENKINS_JAVA_OPTIONS="-Djava.awt.headless=true"
```

```
JENKINS_PORT="8080"
```

```
JENKINS_LISTEN_ADDRESS=""
```

```
JENKINS_HTTPS_PORT=""
```

```
JENKINS_HTTPS_KEYSTORE=""
```

```
# Password to access the keystore defined in JENKINS_HTTPS_KEYSTORE.
```


```
JENKINS_HTTPS_KEYSTORE_PASSWORD=""  
JENKINS_HTTPS_LISTEN_ADDRESS=""  
JENKINS_DEBUG_LEVEL="5"  
JENKINS_ENABLE_ACCESS_LOG="no"  
JENKINS_HANDLER_MAX="100"  
JENKINS_HANDLER_IDLE="20"  
JENKINS_ARGS=""
```

```
[ec2-user ~]$ sudo service jenkins start
```

**To start the jenkins service at boot-up**, you can run

```
[ec2-user ~]$ sudo chkconfig jenkins on
```

#### 6. Modify EC2 Security Group to open Jenkins port (ex:8080)

Type 	Protocol 	Port Range 	Source 
HTTP	TCP	80	0.0.0.0/0
Custom TCP Rule	TCP	8080	0.0.0.0/0
Custom TCP Rule	TCP	8080	::/0

#### 7. Unlock Jenkins

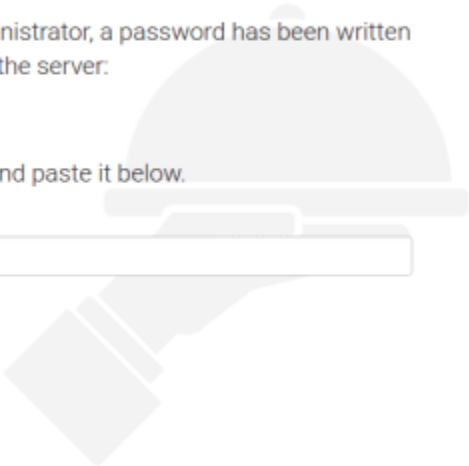
## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to 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.

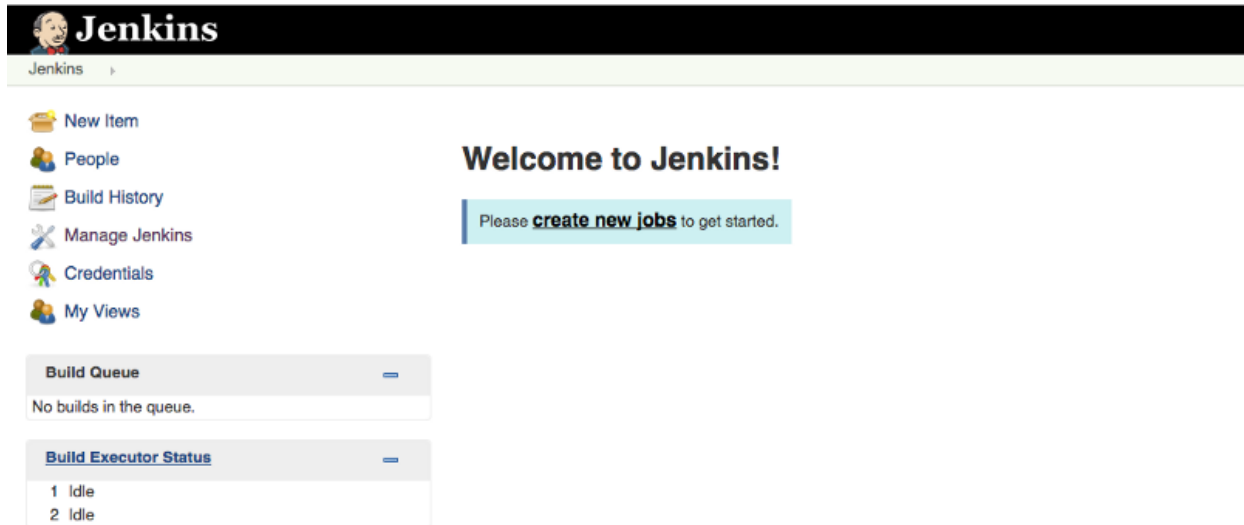
Administrator password



Continue

```
[ec2-user@ip-172-31-25-182 ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

## Jenkins Dashboard Home Page



The default workspace directory of Jenkins is “**/var/lib/jenkins**” and you can change it to any location you want on your server. You can cd into this directory and see all the **logs, jobs, plug-in's, configuration files, secret key, updates, libraries, etc.** are present there.

It's not a best practice to assign Jenkins user sudo permission. But, in some cases you may require to restart the server, build and run your script which needs sudo permission to access particular directory, etc. In that cases, you can assign the Jenkins user sudo permissions as following,

i) Open the sudoers configuration file in your favorite editor, in **/etc/sudoers**  
[ec2-user ~]\$ sudo vim /etc/sudoers

ii) Add/Modify Jenkins users. Modify the Jenkins users line as below if it's present or else you can simply add the following line as,  
jenkins ALL=(ALL) NOPASSWD: ALL

Finally save your file and you are done. Now you could be able to execute the build jobs which require sudo permissions.

## Install ChromeDriver, Xvfb and Git

```
cd /tmp
# download the Linux-based chromedriver
wget https://chromedriver.storage.googleapis.com/2.9/chromedriver\_linux64.zip
unzip chromedriver_linux64.zip
sudo mv chromedriver /usr/bin/chromedriver
chromedriver --version
```

```
cd
sudo yum install xorg-x11-server-Xvfb
sudo yum install git
```

## Jenkins Setup

### Install Plugins:

- Open the plugin manager by doing the following, navigate to: Jenkins > Manage Jenkins > Manage Plugins, then click the "Available" tab
- Select the following plugins by selecting from the alphabetized list, or by typing the plugin name in the "Filter" textbox, then click the associated checkbox and click the "Install without restart" button (repeat for each plugin)
  - GitHub Organization Folder
  - Pipeline
  - Xvfb
  - Safe Restart
  - Email Extension Plugin
  - Test Results Analyzer
  - TestNG Results

### Jenkins Global Tool Configuration:

Open the Jenkins Global Tool Configuration window by navigating to: Jenkins > Manage Jenkins > Global Tool Configuration

- Scroll to the "JDK" section and do the following:
  1. Click the "Add JDK" button
  2. For the name textbox, just enter: JDK 8 (this is whatever Java version you happen to be using, this could be 7, 8, 9, etc.)
  3. Click the "Install automatically" checkbox
  4. Select the "Install from java.sun.com" option, then select the appropriate version
  5. Click the "I agree..." checkbox. You may have to enter the Java/Oracle username credentials, which can be setup for free on their website
- Scroll to the "Git" section and do the following:
  1. Click the "Add Git" button
  2. Name the Git Installation Default or Git
  3. In the "Path to Git executable" textbox, enter: /usr/bin/git
- Scroll to the "Maven" section and do the following:
  1. Click the "Add Maven" button
  2. Just name the installation Maven or Maven <version number>
  3. Click the "Install automatically" checkbox
  4. Select the version installed (usually the latest version available) in the dropdown box under "Install form Apache"
  - 5.
- Scroll to the "Xvfb installation" section and do the following:
  1. Click the "Add Xvfb installation" button
  2. Name the installation simply, Xvfb
  3. Click the checkbox by "Install automatically" ( /usr/bin/Xvfb )

Click the "Save" button

### Jenkins System Configuration:

Open the Jenkins System Configuration by navigating to: Jenkins > Manage Jenkins > Configure System

Scroll to the "Global properties" section

Click the checkbox by "Environment variables"

Add a new environment variable by clicking the "Add" button

Name the variable: XDG\_RUNTIME\_DIR

For the value, enter: /run/usr/1001

Scroll to the "Jenkins Location" section

In the "Jenkins URL" textbox, enter: <http://34.219.163.216:8080/>

Click the "Save" button

### Create the Job for the Automation Project:

To create a new project for your automated tests, do the following:

1. From the main Jenkins vertical menu, select: "New Item"
2. Name the [project1](#) in the "Enter an item name" textbox
3. Select the "Freestyle project" option
4. Click the "OK" button

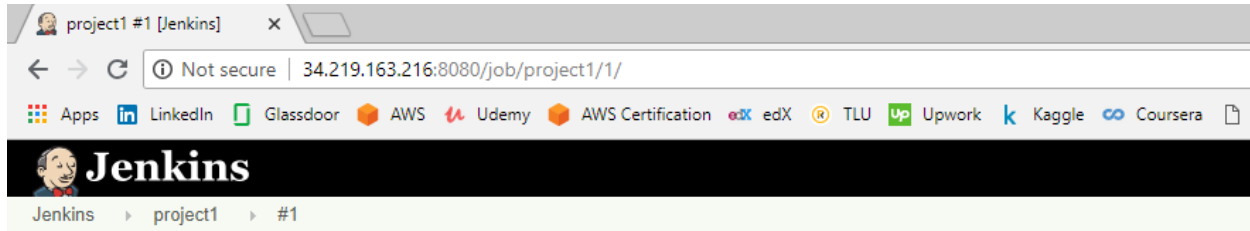
### Configure Jenkins Automated Test Project/Job:

Select the newly created project, then click "Configure" in the left vertical menu

- Scroll to the "Source Code Management" section
  1. Click the "Git" radio button
  2. Set the [repository URL](https://github.com/taislu/Maven_23July.git) in the associate textbox  
[https://github.com/taislu/Maven\\_23July.git](https://github.com/taislu/Maven_23July.git)
  3. If the GitHub repo is Private, rather than Public, add the necessary credentials
  4. Scroll to the "Branches to build" section
  5. In the "Branch Specifier" textbox enter: [\\*/master](#)
- Scroll to the "Build Environment" section
  1. Click the checkbox by the "Start Xvfb before the build, and shut it down after." Option
  2. Click the "Advanced..." button
  3. Select the Xvfb installation previously created from the drop down menu
- Scroll to the "Build" section
  1. Click the "Add build step" button
  2. Select the "Invoke top-level Maven targets" option
  3. Select the previously named Maven installation from the dropdown menu
  4. In the "Goals" textbox enter: [clean test](#)
  5. Click the "Advanced..." button
  6. In the "POM" textbox enter: [\\$workspace/pom.xml](#)
- Scroll to the "Post-build Actions" section
  1. Click the "Add post-build action" button
  2. Select the "Publish TestNG Results" option
  3. In the "TestNG XML report pattern" textbox enter: [\\*\\*/testng-results.xml](#)

Click the "Save" button

## Jenkins with Maven/Build and TestNG/Tests on AWS EC2



Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete Build

Git Build Data

No Tags

TestNG Results

**Build #1 (Sep 3, 2018 3:29:06 PM)**

No changes.

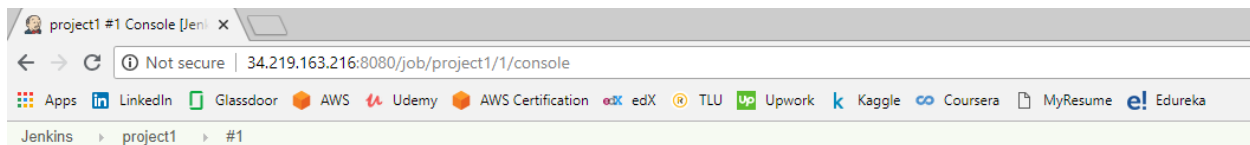
Started by user [Tai Lu](#)

Revision: 7486ec6d67fd8a78d9108b66e6111e5b699f9c7f

- refs/remotes/origin/master

[TestNG Results](#)

- Total Tests: 10 (+10)
- Failed Configurations: 0 (±0)
- Failed Tests: 0 (±0)
- Skipped Tests: 0 (±0)
- Skipped Configurations: 0 (±0)



```
-----
T E S T S
-----
Running HelloMaven.Test1_RestApiBasicForJson
Configuring TestNG with: org.apache.maven.surefire.testng.conf.TestNG652Configurator@3cd1f1c8
HTTP/1.1 200 200
Date: Mon, 03 Sep 2018 15:29:39 GMT
Server: Apache/2.4.25 (Debian)
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8

{
  "RestResponse": {
    "messages": [
      "Total [55] records found."
    ],
    "result": [
      {
        "id": 1,
        "country": "USA",

```

## GitHub Repository

**Maven\_23July** : [https://github.com/taislu/Maven\\_23July.git](https://github.com/taislu/Maven_23July.git)

GitHub, Inc. [US] | [https://github.com/taislu/Maven\\_23July](https://github.com/taislu/Maven_23July)

taislu / Maven\_23July

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. [Add topics](#) [Edit](#)

4 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

File	Commit	Time
.settings	First commit	17 days ago
src	2018-09-01	44 minutes ago
target	2018-09-01	44 minutes ago
test-output	2018-09-01	44 minutes ago
.classpath	First commit	17 days ago
.project	First commit	17 days ago
pom.xml	2018-09-01	44 minutes ago
pom1.xml	First commit	17 days ago
pom2.xml	2018-08-17 8:52am PDT	15 days ago

**pom.xml** : (extracted)

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-report-plugin</artifactId>
  <version>2.22.0</version>
  <configuration>
    <suiteXmlFiles>
      <suiteXmlFile>testng2.xml</suiteXmlFile>
    </suiteXmlFiles>
  </configuration>
</plugin>
```

**testing2.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">

  <test thread-count="5" name="Test">
    <classes>
      <class name="HelloMaven.Test1_RestApiBasicForJson"/>
    </classes>
  </test> <!-- Test -->
```

```
</suite> <!-- Suite -->
```

```
HelloMaven.Test1_RestApiBasicForJson
```

```
package HelloMaven;
```

```
import org.testng.annotations.Test;
```

```
import static io.restassured.RestAssured.*;
```

```
import static org.hamcrest.Matchers.equalTo;
```

```
import static org.hamcrest.Matchers.hasItems;
```

```
public class Test1_RestApiBasicForJson {
```

```
    // simply checking the status code
```

```
    @Test
```

```
    public void testStatusCode() {
```

```
        given().
```

```
            get("http://jsonplaceholder.typicode.com/posts/3").
```

```
        then().
```

```
            statusCode(200);
```

```
    }
```

```
    // verify status code and log response
```

```
    @Test
```

```
    public void testLogging() {
```

```
        given().
```

```
            get("http://services.groupkt.com/state/get/USA/all").
```

```
        then().
```

```
            statusCode(200).
```

```
            log().all();
```

```
    }
```

```
    // verify content using org.hamcrest.Matchers equalTo method
```

```
    @Test
```

```
    public void testEqualToMethod() {
```

```
        given().
```

```
            get("http://services.groupkt.com/state/get/USA/CA").
```

```
        then().
```

```
            body("RestResponse.result.name", equalTo("California"));
```

```
    }
```

```
    // verify multiple content using org.hamcrest.Matchers hasItems method
```

```
    @Test
```

```
    public void testHasItemsMethod() {
```

```
        given().
```



```

        get("http://services.groupkt.com/state/get/USA/all").
    then().
        body("RestResponse.result.name", hasItems("California",
"Massachusetts")));
    }

    // verify parameter can be set

@Test
public void testParameters() {

    given().
        param("text", "California").
    when().
        get("http://services.groupkt.com/state/search/USA").
    then().
        statusCode(200).
        log().all();
}

}

```

## Local Maven Build

### mvn clean install

```

[INFO] Building jar: c:\2018\edureka\selenium\Maven_23July\target\Maven_23July-0.0.1-SNAPSHOT-
sources.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ Maven_23July ---
[INFO] Installing c:\2018\edureka\selenium\Maven_23July\target\Maven_23July-0.0.1-SNAPSHOT.jar to
C:\Users\Tai\.m2\repository\Edureka\Maven_23July\0.0.1-SNAPSHOT\Maven_23July-0.0.1-
SNAPSHOT.jar
[INFO] Installing c:\2018\edureka\selenium\Maven_23July\pom.xml to
C:\Users\Tai\.m2\repository\Edureka\Maven_23July\0.0.1-SNAPSHOT\Maven_23July-0.0.1-
SNAPSHOT.pom
[INFO] Installing c:\2018\edureka\selenium\Maven_23July\target\Maven_23July-0.0.1-SNAPSHOT-
sources.jar to C:\Users\Tai\.m2\repository\Edureka\Maven_23July\0.0.1-SNAPSHOT\Maven_23July-
0.0.1-SNAPSHOT-sources.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 34.610 s
[INFO] Finished at: 2018-09-01T05:36:59-07:00
[INFO] -----

```

