**JENKINS INSTALLATION STEPS FOR EC2 INSTANCE OF REDHAT**

**STEPS:**

**Install and Configure Jenkins on Amazon AMI**

1. **Step 1: Set up Prerequisites (make sure you have jdk 1.8 or more).**
2. **Launch an EC2 Instance.**

In this step you will launch a virtual server to host Jenkins. These virtual instances are called EC2 instances. This is also called as Amazon Machine Image(AMI).

**You will complete the following tasks:**

* Create a Security Group for Your Amazon EC2 Instance
* Launch your EC2 Instance

1. **Install and Configure Jenkins**

In this step you will deploy Jenkins on your EC2 instance by completing the following tasks:

* Connect to Your Linux Instance
* Download and Install Jenkins
* Configure Jenkins

|  |
| --- |
| **Step 1:** This Command will logged you as a ROOT User **[ec2-user@ip-172-31-22-53 ~]$ sudo su** |
| **Step 2:** Before we do anything, we’re going to check that our operating system is up to date and To ensure that your software packages are up to date on your instance, use the following command to perform a quick software update**: [root@ip-172-31-22-53 ec2-user]# sudo yum -y update** |
| **Step 3:** Set up Prerequisites (make sure you have jdk 1.8 or more).  Before doing the steps make sure you have java1.8 rather then default java1.7 ship with Amazon AMI. If you could see the 1.7 version you can uninstall and install the 1.8 using following commands:  **[root@ip-172-31-22-53 ec2-user]# sudo yum remove java-1.7.0-openjdk**  **[root@ip-172-31-22-53 ec2-user]# sudo yum install java-1.8.0** |
| **Step 4:** Download the latest Jenkins code package**. Or** Add the Jenkins repo using the following command.  We need to add the **Jenkins repository** so that **yum** knows **where to install Jenkins from**.  **# sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat/jenkins.repo** |
| **Step 5:** Import a key file from Jenkins-CI to enable installation from the package:  Next, we’re adding the Jenkins GPG key to our trusted keys so that we’re able to install Jenkins, verifying that the files are being sourced from a trusted location.  **#sudo rpm --import** [**http://pkg.jenkins-ci.org/redhat/jenkins-ci.org.key**](http://pkg.jenkins-ci.org/redhat/jenkins-ci.org.key) |
| **Step 6:** Great! We’ve prepared our environment with the required dependencies so we can now install Jenkins.  **# sudo yum install jenkins –y**  **If you want to check weather Jenkins is installed or not?**  **# cd /var/lib**  This lib Directory contains Jenkins   |  | | --- | | Go to ec2 redhat(Linux)instance 🡪 Security Group 🡪 inbound 🡪 edit 🡪 add rule 🡪 TCP ALL 🡪 Save | |
| **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”** |
| **Step 7:** Let’s Start & Enable the Jenkins service:  **# sudo service jenkins start / sudo systemctl start jenkins.service** Or  **# systemctl start jenkins.service**  **# systemctl enable jenkins.service** |
| If you want Jenkins to automatically start when your instance is started, we can use chkconfig to add Jenkins to our startup services.  **# sudo chkconfig --add jenkins** |
| **Step 8:** All done! You can now access your Jenkins server using the public DNS on port 8080.  **http://{ec2-public-dns}:8080**  i.e Configure Jenkins, now that it’s installed and running on your Amazon EC2 instance. Use its management interface at port 80 or 8080.  The first time you use the dashboard at **http://*your-server-address*:8080**, you will be prompted to unlock Jenkins: |
| Step 9: As noted on the user interface, you can find this password in  **/var/lib/jenkins/secrets/initialAdminPassword**  **# sudo cat /var/lib/jenkins/secrets/initialAdminPassword**  Copy the output from terminal & Paste the value into the password box, then choose**Continue.**  **unlock-jenkins.png**  Select **install suggested plugins 🡪 enter uname & etc.** Click on **Save & Finish.** |
| Here your **installation steps completed** and you can **start configuring and managing the Jenkins CI server** as per your requirement. **Automate the task for your workflow**. You can see the *“****Welcome to Jenkins!”* message on the home page** as shown below. |

The default workspace directory of jenkins in most of the OS is “**/var/lib/jenkins**”

**Let’s Configure Maven into Jenkins**

|  |
| --- |
| **Step 1:** First Install maven on EC2 redhat (Linux) instance 🡪 go to following URL  [**https://www.javahelps.com/2017/10/install-apache-maven-on-linux.html**](https://www.javahelps.com/2017/10/install-apache-maven-on-linux.html)  **on this webpage**  **Step 1:** Download *apache-maven-3.5.0-bin.tar.gz* binary archive from this official link: [Download Apache Maven](https://maven.apache.org/download.cgi). This article uses the Maven version 3.5.0. You need to replace the version number by whatever the version you are downloading.  Click on [Download Apache Maven](https://maven.apache.org/download.cgi) you will be redirected to 🡪 <https://maven.apache.org/download.cgi>  Go to files 🡪    Right click on 🡪 **apache-maven-3.5.4-bin.tar.gz 🡪** click on **copy link address** |
| **Step 2:** Open the Terminal and move to the /opt directory.  **Go to ec2 instance 🡪 cd /usr/local/share/ 🡪 create directory (mkdir maven) 🡪 execute a command below:** # cd maven  **# wget paste the copied link**  **# wget https://www-eu.apache.org/dist/maven/maven-3/3.5.4/binaries/apache-maven-3.5.4-bin.tar.gz** |
| **Step 3:** Extract the apache-maven archive into the maven directory**.**  **# sudo tar –xvf apache-maven-3.5.4-bin.tar.gz** |
| **Step 4:** Edit the /etc/environment file and add the following environment variable: # Cd /etc  #Vi /etc/profile   |  | | --- | | **M2\_HOME="/usr/local/share/maven/apache-maven-3.5.4"**  and append the bin directory to the PATH variable:  **PATH="/usr/local/share/maven/apache-maven-3.5.4/bin:$PATH** |   Again go to 🡪 **cd /usr/local/share/maven Directory** |
| **Step 5:** Update the mvn command:  # sudo update-alternatives --install "/usr/bin/mvn" "mvn" "/**opt**/apache-maven-3.5.0/bin/mvn" 0  **#** update-alternatives --install "/usr/bin/mvn" "mvn" **"/usr/local/share/maven**/apache-maven-3.5.0/bin/mvn" 0  # sudo update-alternatives --set mvn /opt/apache-maven-3.5.0/bin/mvn  # update-alternatives --set mvn /**usr/local/share/maven**/apache-maven-3.5.0/bin/mvn |
| **Step 6:** Add Bash completion to mvn so that you can complte complex Maven commands by hitting Tab multiple times.  sudo wget https://[raw.github.com/dimaj/maven-bash-completion/master/bash\_completion.bash](http://raw.github.com/dimaj/maven-bash-completion/master/bash_completion.bash) --output-document /etc/bash\_completion.d/mvn |
| **Step 7:** Logout and login to the computer and check the Maven version using the following command. **# mvn --version** |
| *If it works, hooray! you have successfully installed the latest Apache Maven on your computer.* |
| **Configure maven with jenkins** |
| [root@ip-172-31-22-53 maven]# **cd apache-maven-3.5.4**  [root@ip-172-31-22-53 apache-maven-3.5.4]# mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false  **# cd my-app**  **my-app**  **|-- pom.xml**  **`-- src**  **|-- main**  **| `-- java**  **| `-- com**  **| `-- mycompany**  **| `-- app**  **| `-- App.java**  **`-- test**  **`-- java**  **`-- com**  **`-- mycompany**  **`-- app**  **`-- AppTest.java** |
| **[root@ip-172-31-22-53 my-app]# vim pom.xml**  Sample POM  <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.mycompany.app</groupId>  <artifactId>my-app</artifactId>  <version>1.0-SNAPSHOT</version>  <packaging>jar</packaging>    <name>Maven Quick Start Archetype</name>  <url>http://maven.apache.org</url>    <dependencies>  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>4.8.2</version>  <scope>test</scope>  </dependency>  </dependencies>  </project> |
| Build the Project  [**root@ip-172-31-22-53 apache-maven-3.5.4]#** cd my-app/  **[root@ip-172-31-22-53 my-app]# mvn package**  After Build Success  **[root@ip-172-31-22-53 my-app]# ls**  pom.xml src target  After Build Success  **[root@ip-172-31-22-53 my-app]# java -cp target/my-app-1.0-SNAPSHOT.jar com.mycompany.app.App** |
| Output:  Hello World!  **[root@ip-172-31-22-53 my-app]# readlink /usr/local/share/maven/apache-maven-3.5.4/bin/**  **[root@ip-172-31-22-53 my-app]# which java**  /bin/java  **[root@ip-172-31-22-53 my-app]# readlink -f /bin/java/bin**  **[root@ip-172-31-22-53 my-app]# readlink -f /maven/apache-maven-3.5.4/bin/mvn**  **[root@ip-172-31-22-53 my-app]# which mvn**  /usr/local/share/maven/apache-maven-3.5.4/bin/mvn  **Mvn on Jenkins:**  Manage Jenkins 🡪 add maven  **Name: mvn 3.5.4**  **Maven Home:** /usr/local/share/maven/apache-maven-3.5.4  Add MVN. |
|  |
|  |
|  |
|  |