**Jenkins**

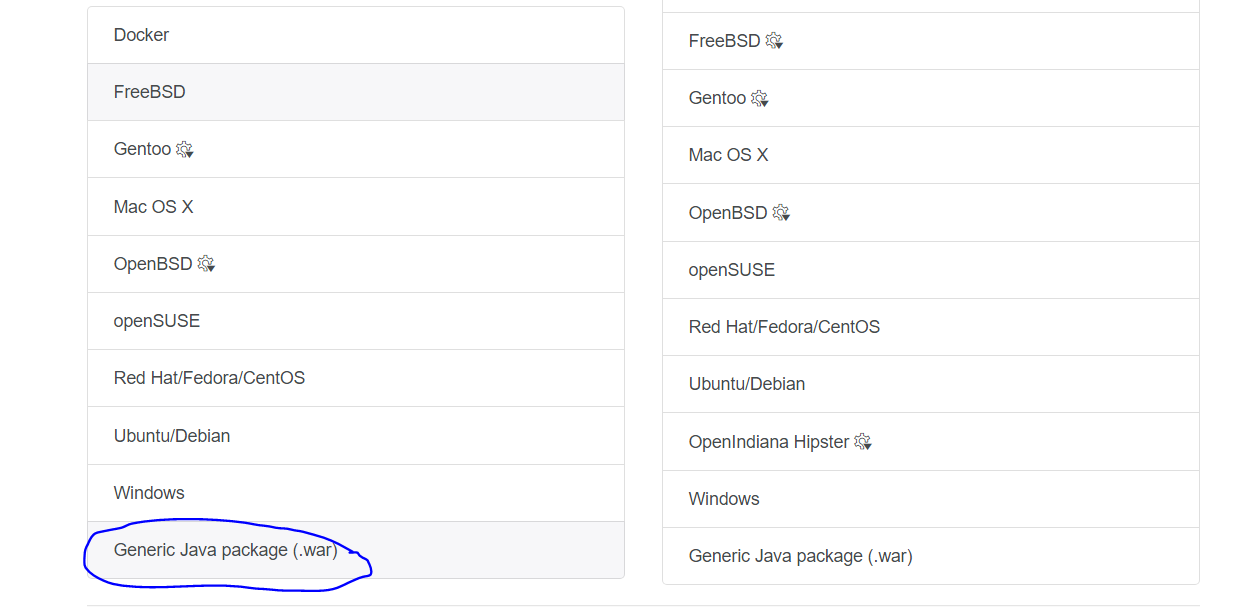
**Prerequisites:**

**JDK Installation, AWS CLI installation, AWS Account, Git Bash installation.**

**If you want to deploy your application on AWS elasticbeanstalk then you must have AWS Credentials for elastic beanstalk setup refer elasticbeanstalk documentation.**

# Step1: Jenkins Installation

**Step1.1:** Download Jenkins war from “[**https://jenkins.io/**](https://jenkins.io/)” and then click on “download” and then click on “Generic java package (.WAR)” to download it. You can click on windows to download jenkins.exe file but you need admin access to install it.



**Step 1.2:** Keep Jenkins.war in any folder.

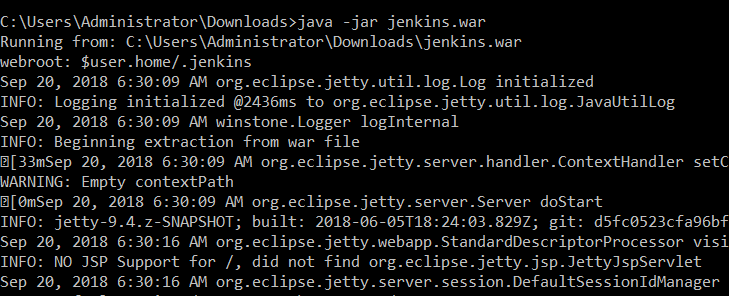
**Step 1.3**: Open CMD and go to folder where you have kept your Jenkins.war file

**Step 1.4:** run following command to run Jenkins

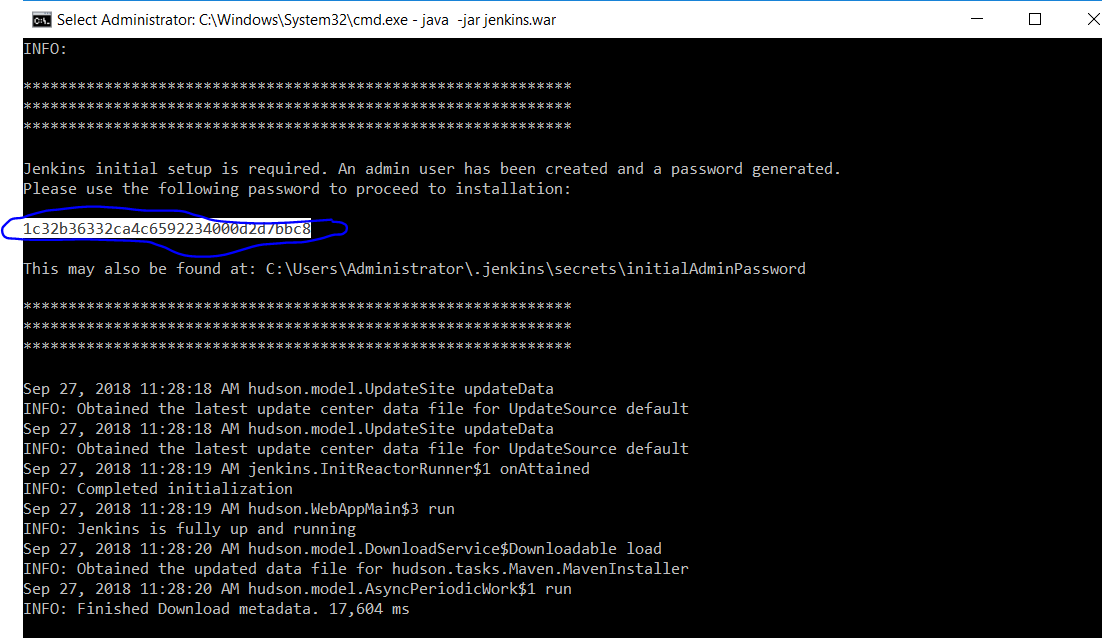
java -jar jenkins.war

By default Jenkins run on 8080 port. If you want to change port use following command.

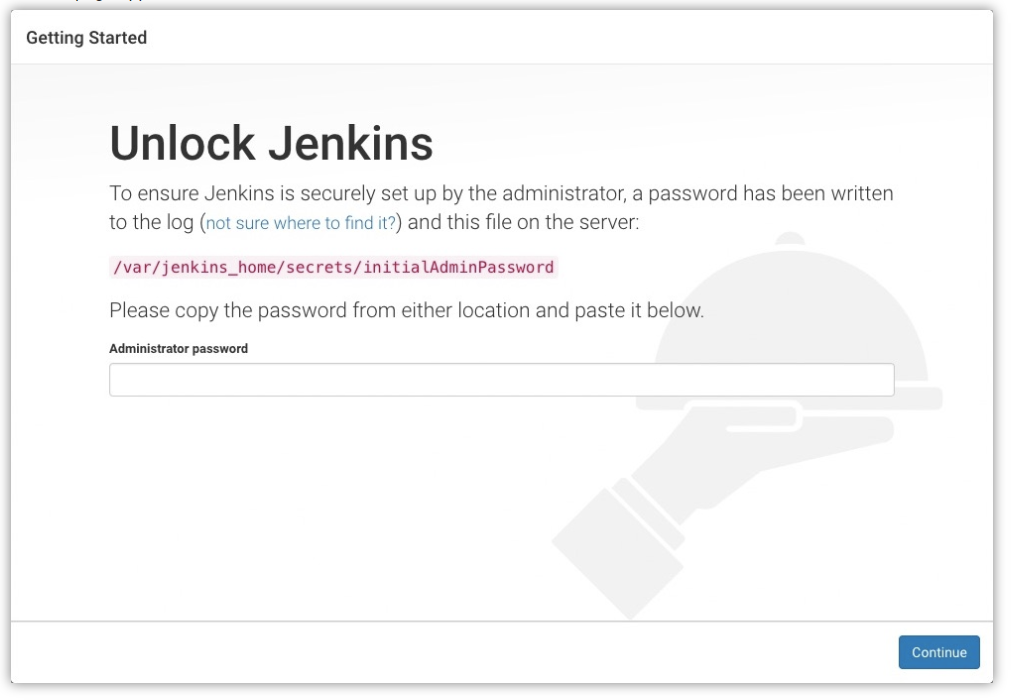
Java -jar jenkins.war httpPort=8181



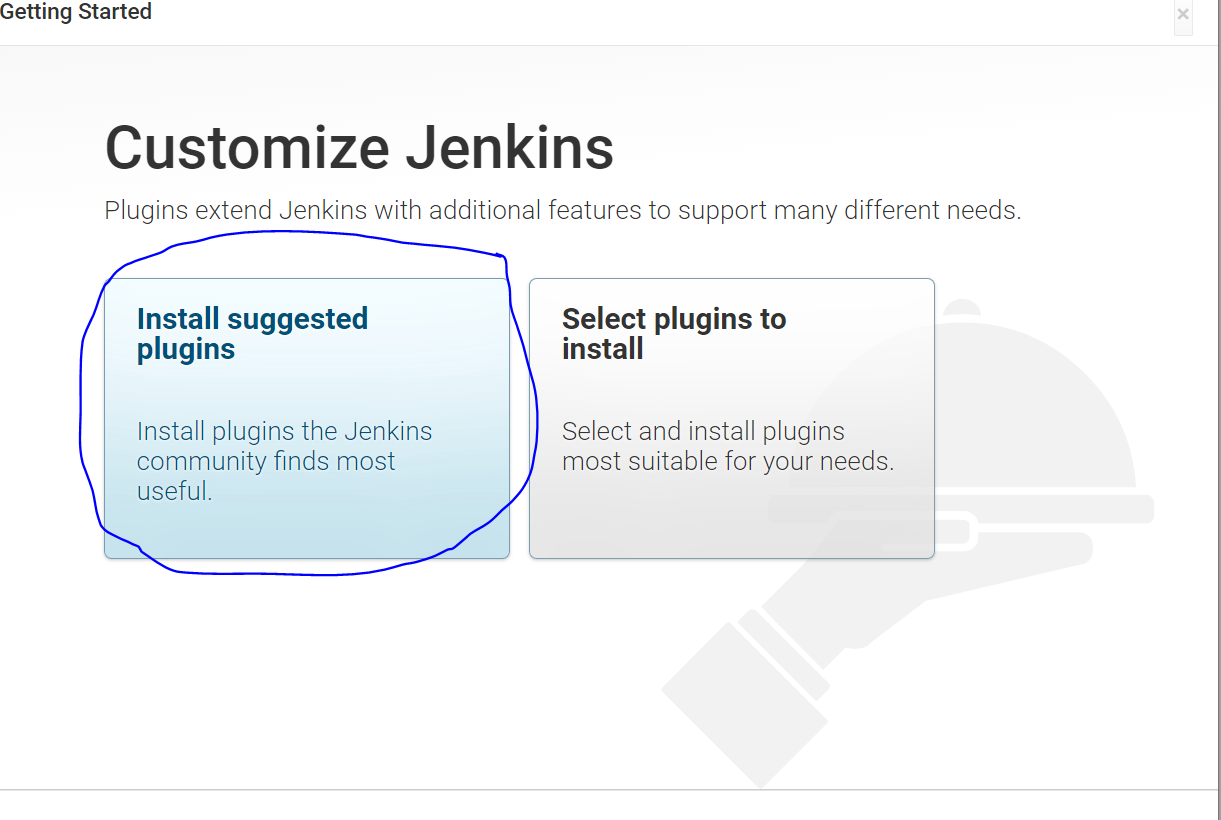
**Step 1.5:** Copy generated password and keep it in secure place.



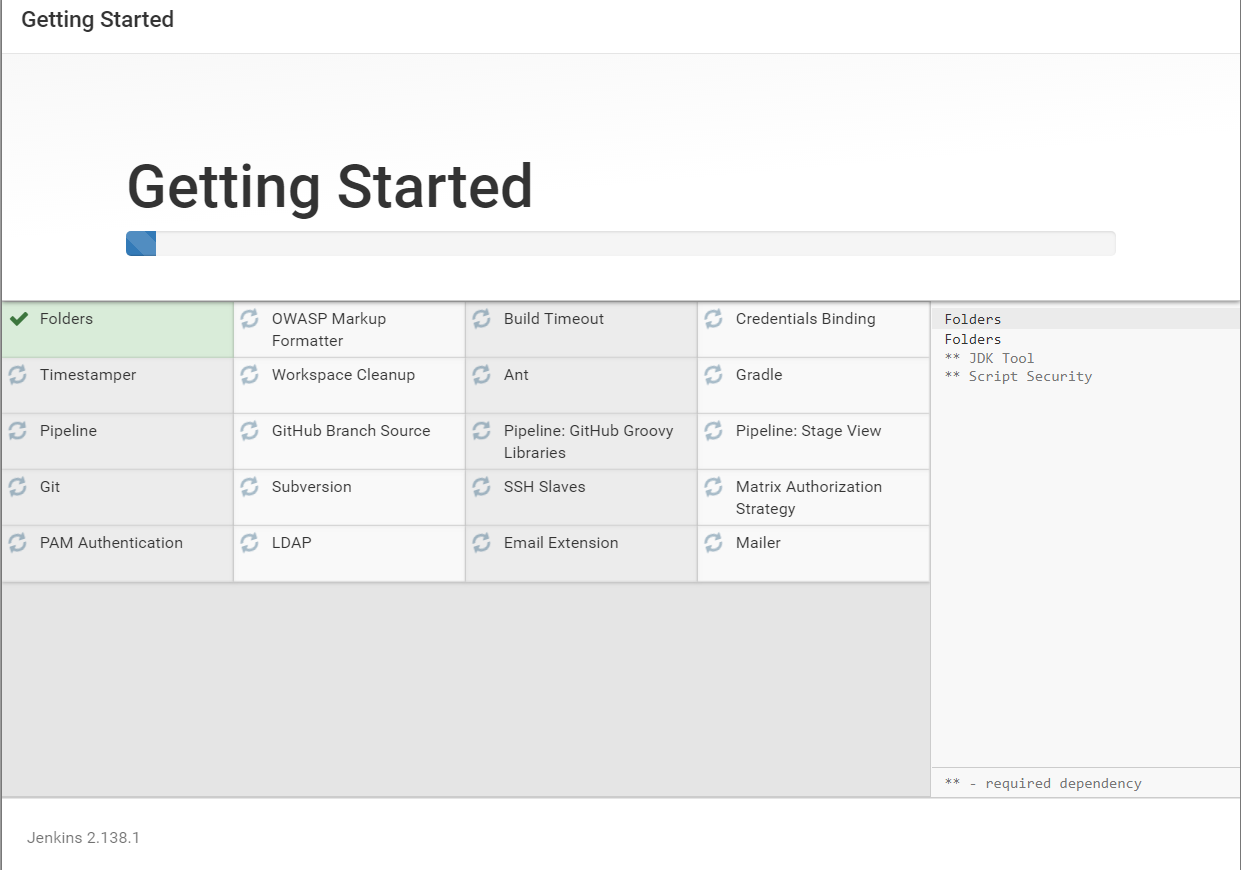
**Step 1.6:** Start Jenkins by giving [**localhost:8080**](https://jenkins.io/)in browser and paste password which you just copied.



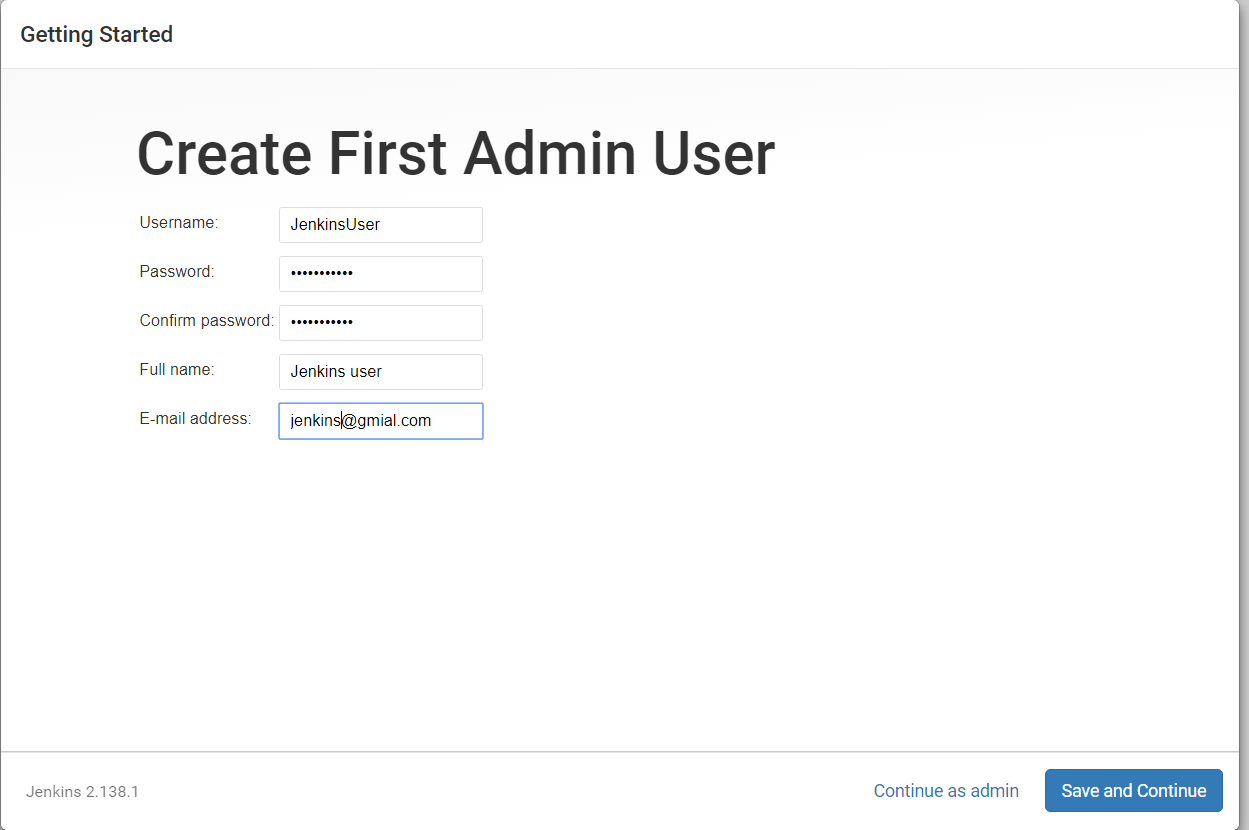
**Step 1.7:** Then click on install suggested plugins.

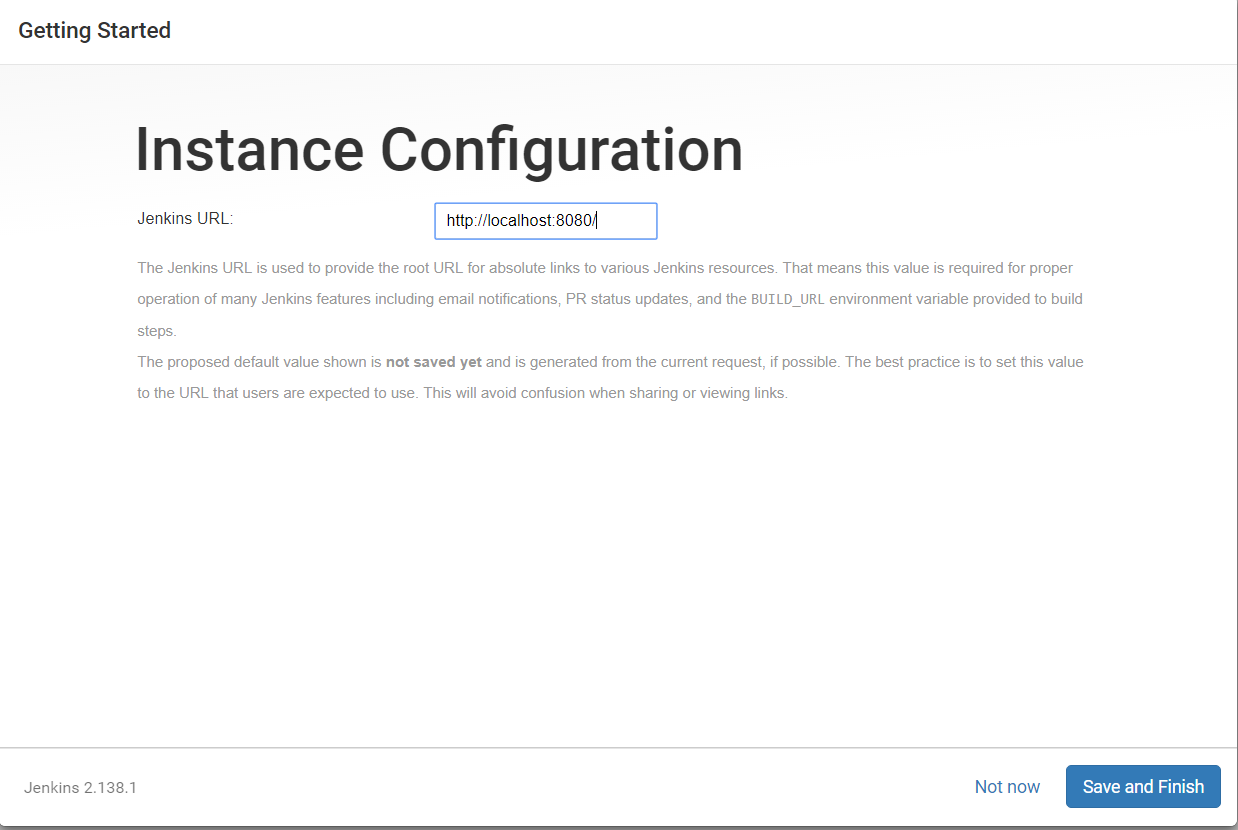


**Strp 1.8:** Then wait until installation of default plugins successfully completed(If your system connected to Capgemini network then you will not authorized to install all plugins.)

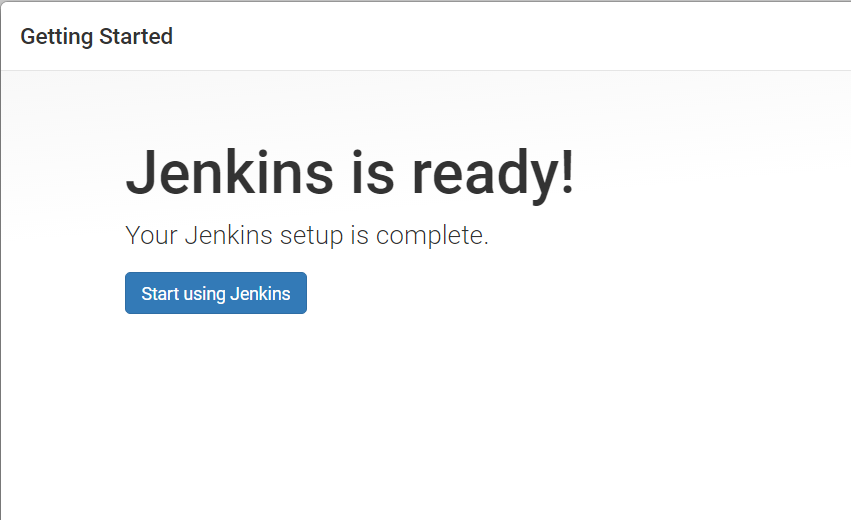


**Step 1.9:** Create User by giving username and password and click on save and continue.

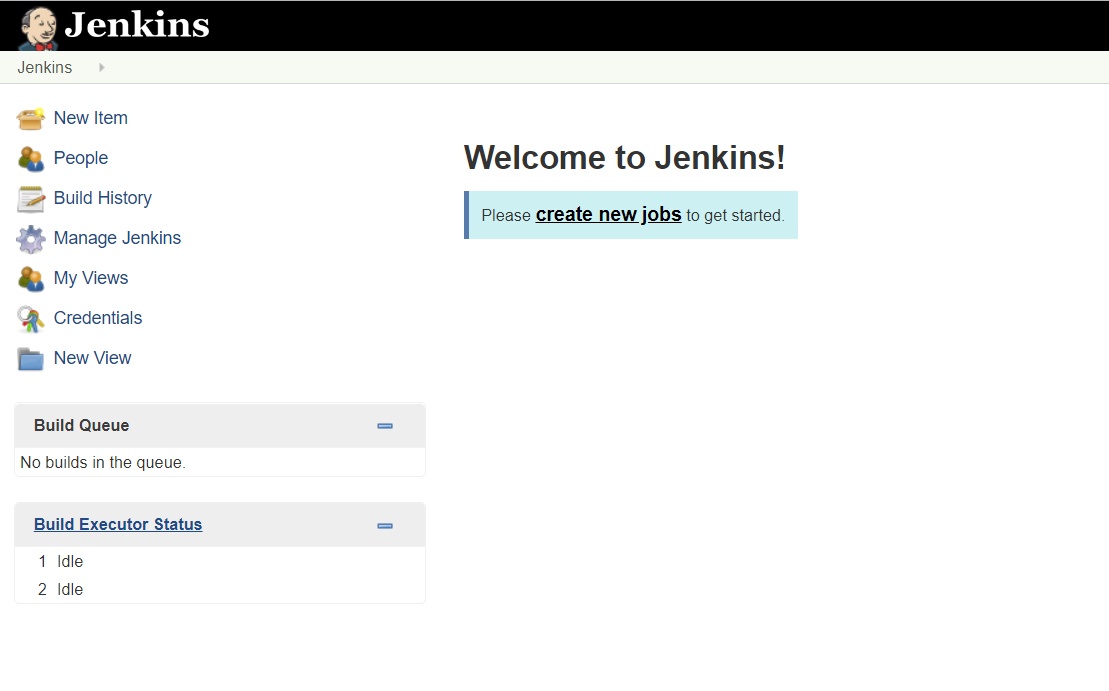


**Step 1.10:** Click on Save and finish

**Step 1.11:** Click on start using Jenkins and now you are ready with Jenkins installation.



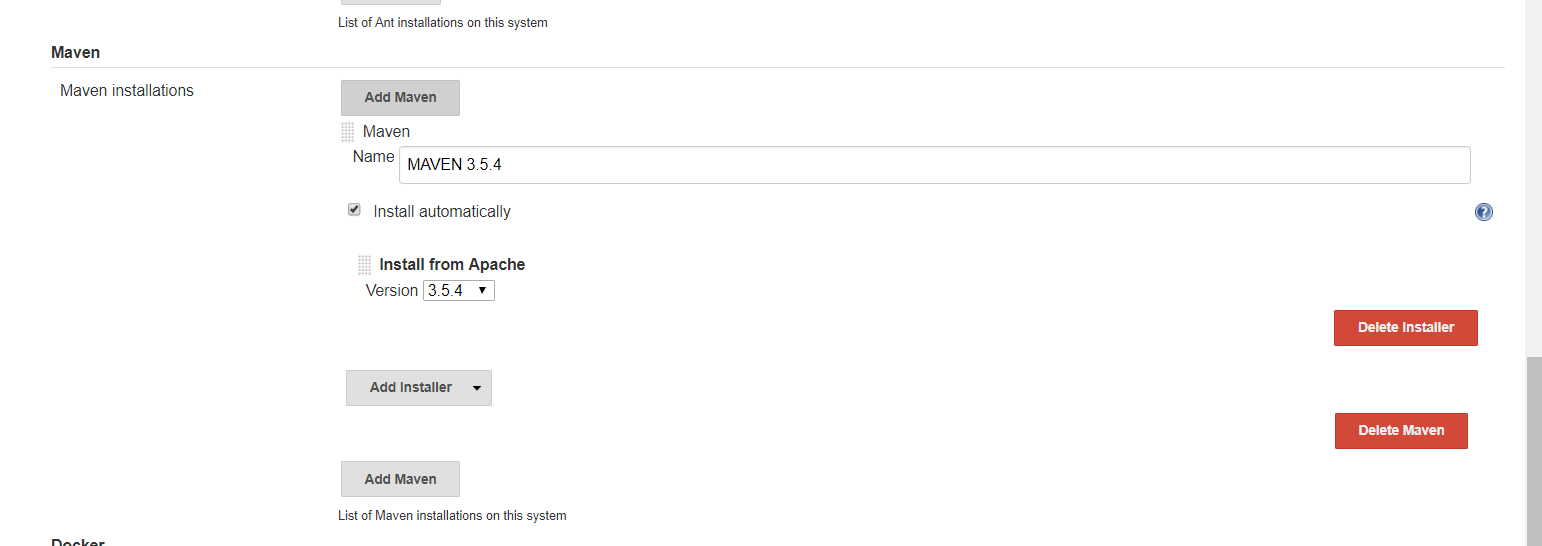
**Step 1.12**: Home page of Jenkins be like this**.**



# **Step 2:** Some Required Configuration

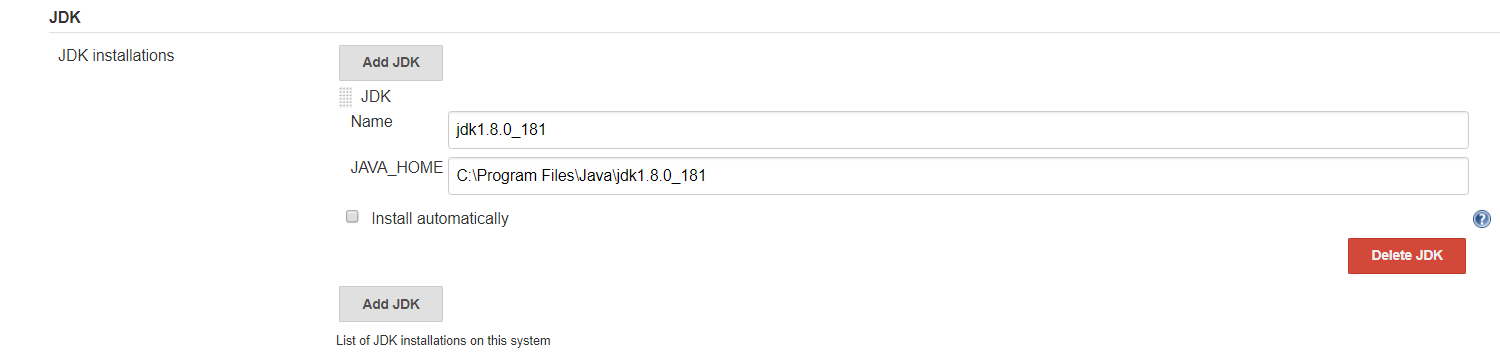
**Step 2.1:** To add Maven

* To Add Maven Application we need to first add maven plugins in Jenkins(Go to step 7)
* Go to manage Jenkins->Global tool configuration->Maven->Add Maven
* In maven name enter maven with version
* Click on install automatically
* Click on add maven.



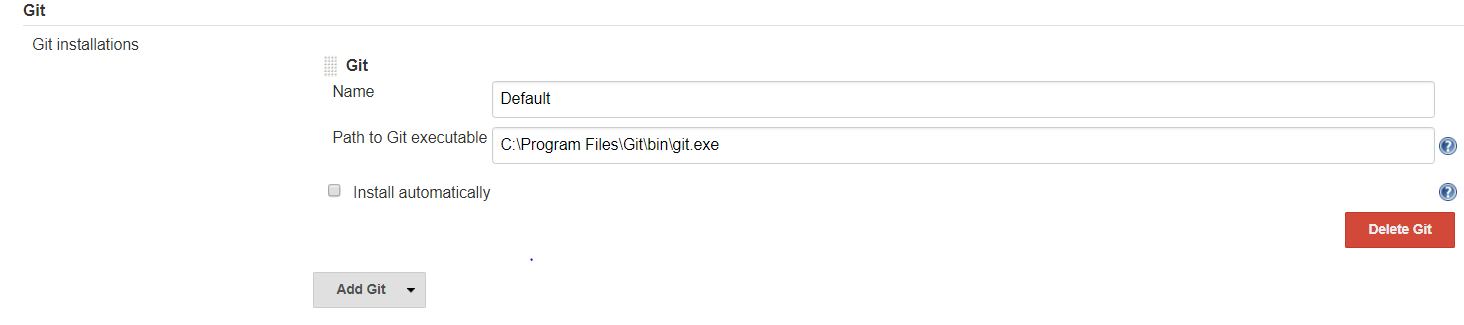
**Step 2.2:** To add JDK

* To add jdk , first add jdk plugins in Jenkins (go to step 7)
* Go to manage Jenkins->Global tool configuration->JDK->JDK installation
* In Name enter jdk with version and in JAVA\_HOME give absolute path of JDK
* Click on install automatically
* Click on Add JDK.



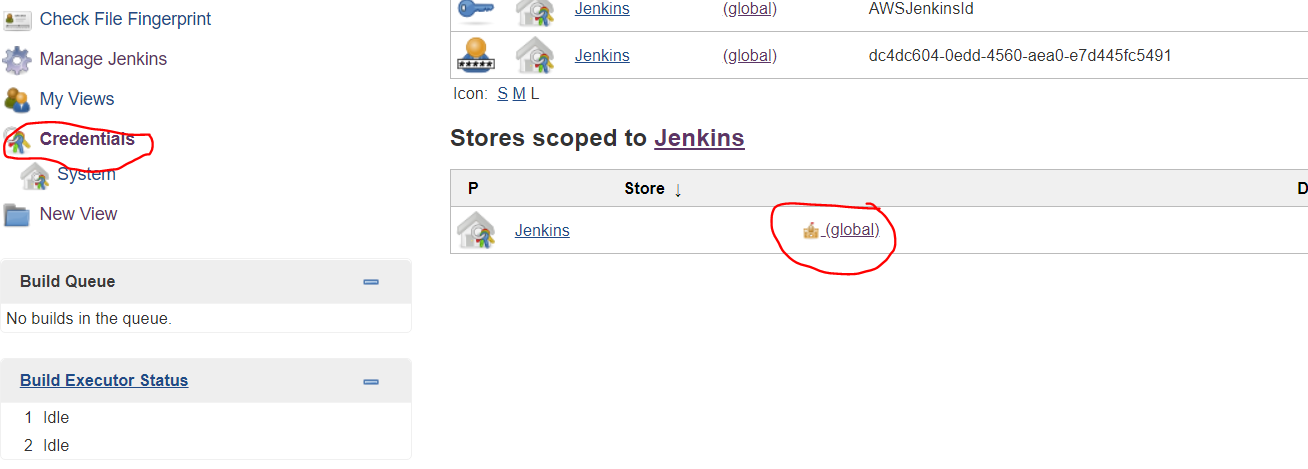
**Step 2.3:** To add Git

* To add Git , first add “Github integration plugins” in Jenkins (go to step 7)
* Go to manage Jenkins->Global tool configuration->Git->Git installations
* Click on install automatically
* OPTIONAL-(Give absolute path of git. Make sure you have install git on your system)
* Click on Add Add Git.



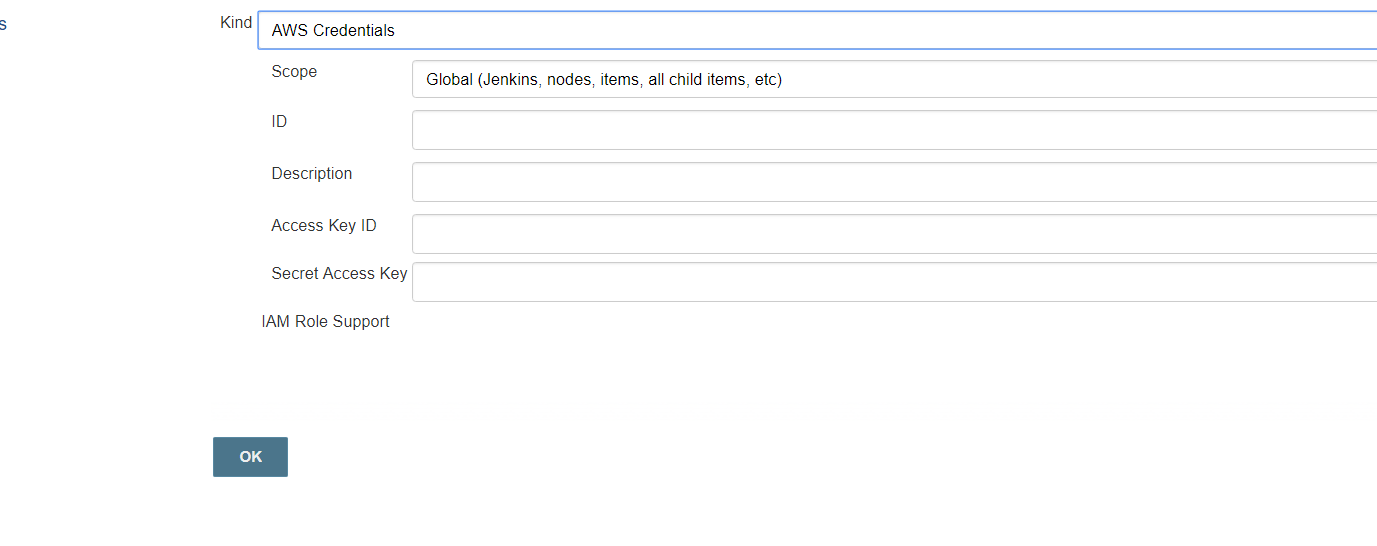
**Step 2.4** To add credentials

* Select Credentails
* Select dropdown near global and click on add credentials



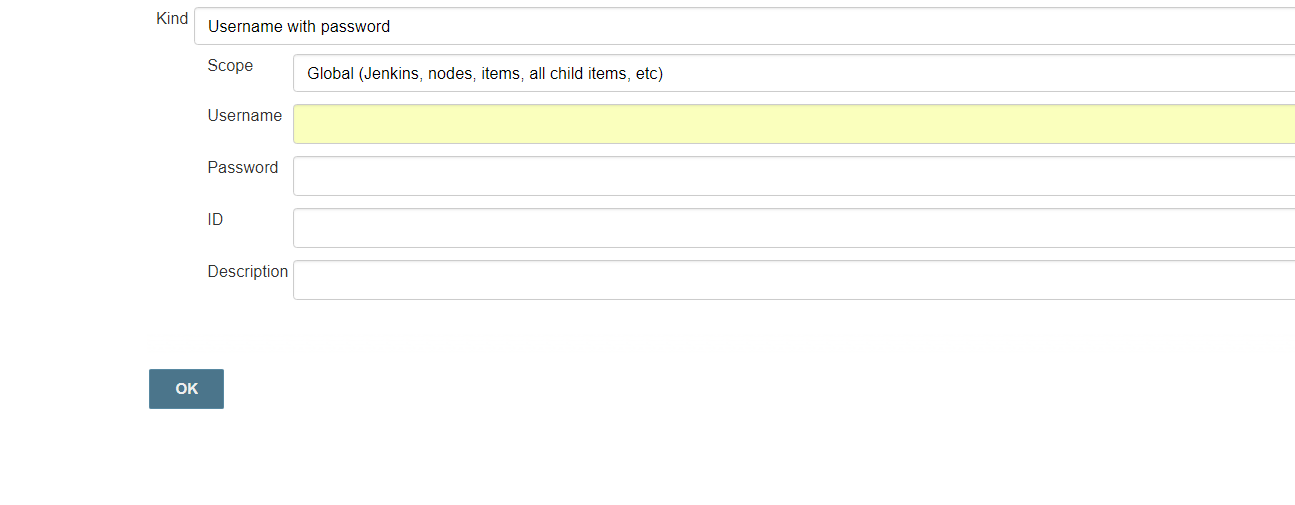
**Step 2.4.1**: To add aws credentials select dropdown at kind and select AWS credentials

* Enter Access key id or secret access key of your AWS credentials and click on OK.



**Step 2.4.2** To add Git credentials select dropdown at kind and select username and password

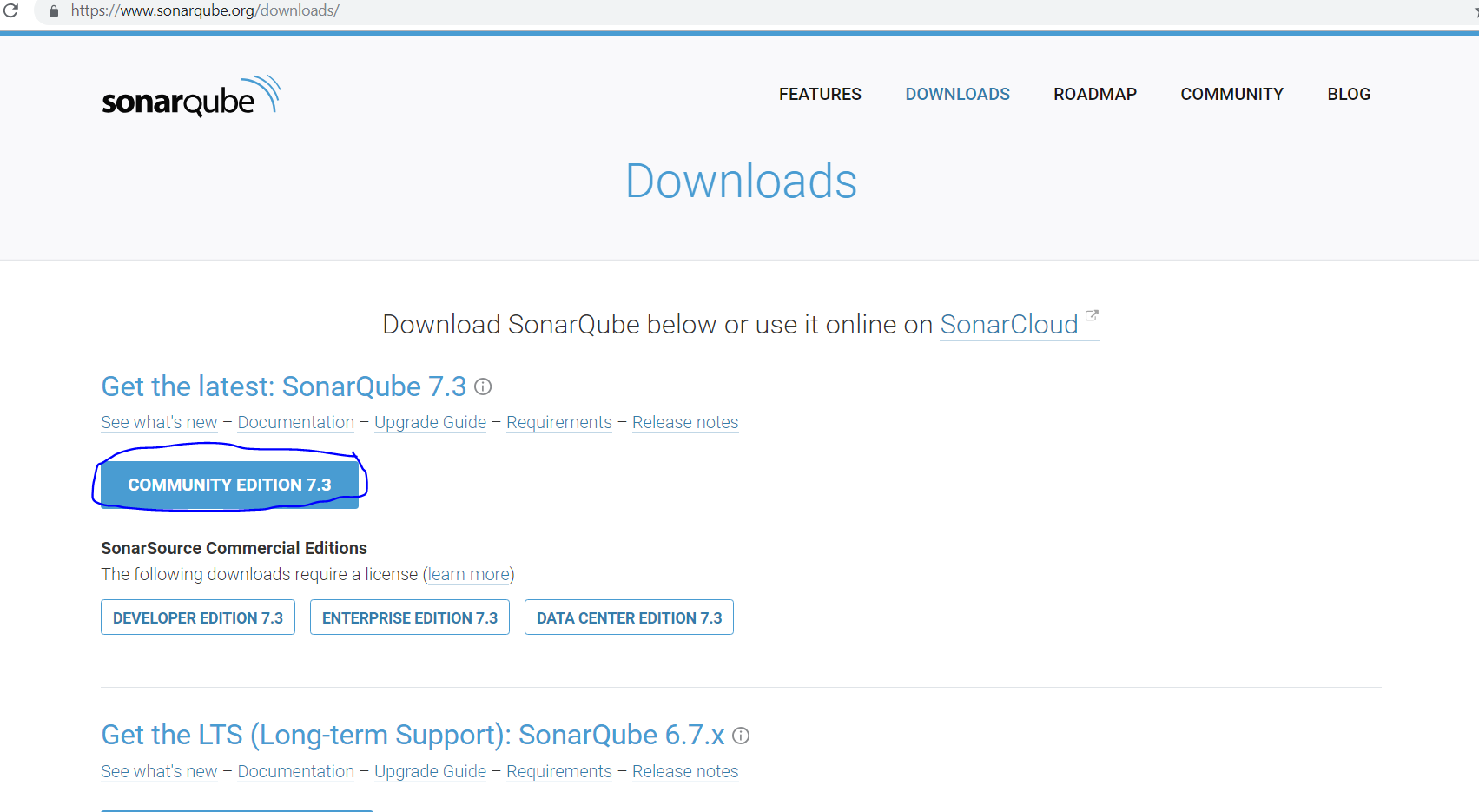
* Enter your git username and password and click on OK



**Step 2.5: Jenkins integration with SonarQube.**

* Go to manage Jenkins ->manage plugins->click on available and search for SonarQube scanner click on install without restart.
* Go to following link and click on download button then click on community edition 7.3 to download SonarQube and extract it.

<https://www.sonarqube.org/>



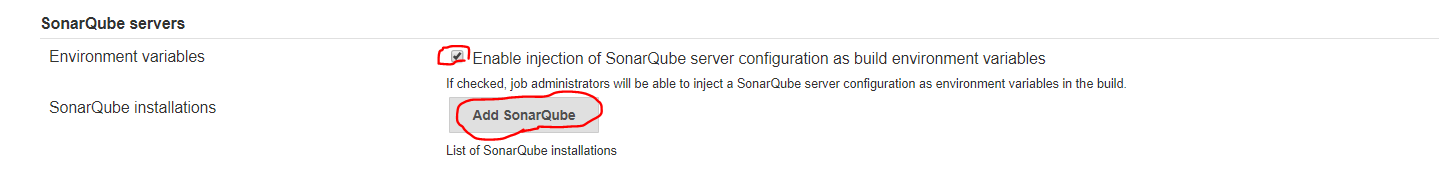
* After extract SonarQube, go to folder \sonarqube-7.3\sonarqube-7.3\bin\windows-x86-64 and then double click on StartSonar
* Go to browser and run localhost:9000(By default SonarQube run on 9000 port)
* Login with admin credentials

-Username= admin

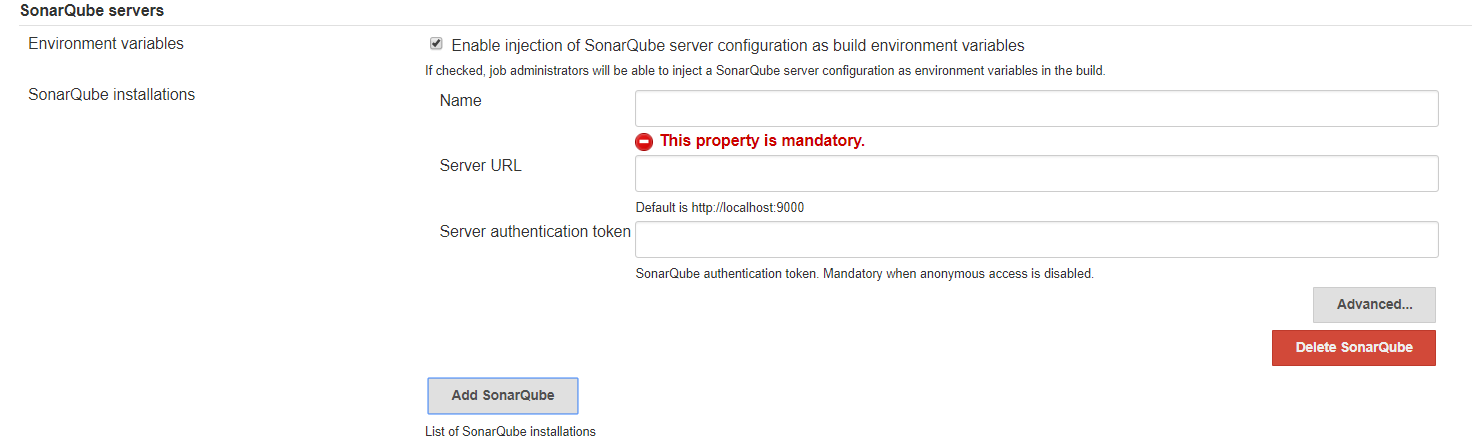
-Password= admin

Copy token generate on screen after login with admin credentials

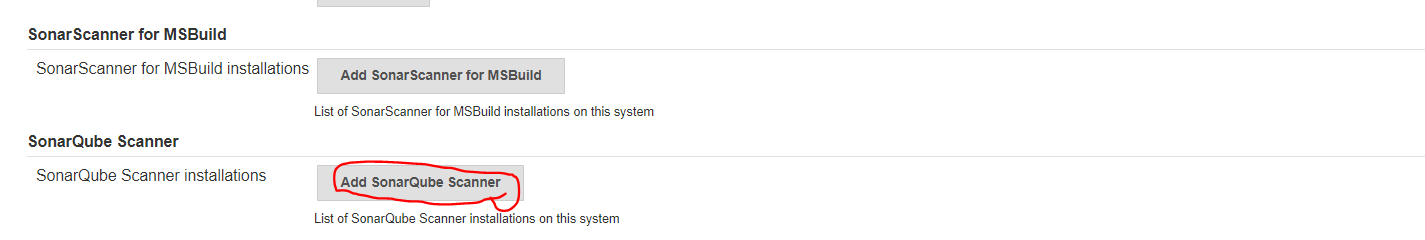
* Go to manage Jenkins->configure system->SonarQubeService->Check on Enable injection of SonarQube server configuration as build environment variables and click on Add SonarQube.



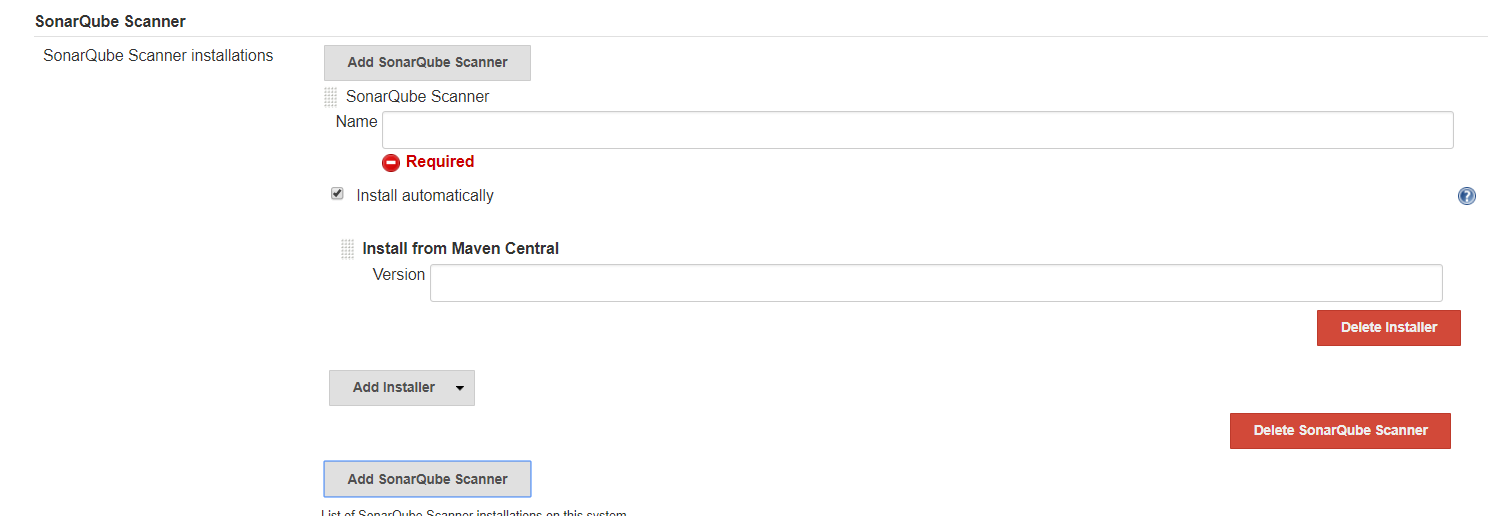
* Enter any name, Enter “http://localhost:9000” in Server URL and paste token which u obtained in last step.
* Click on Add SonarQube
* Click on save button



* Go to manage Jenkins ->Global tool configuration->SonarQube Scanner.

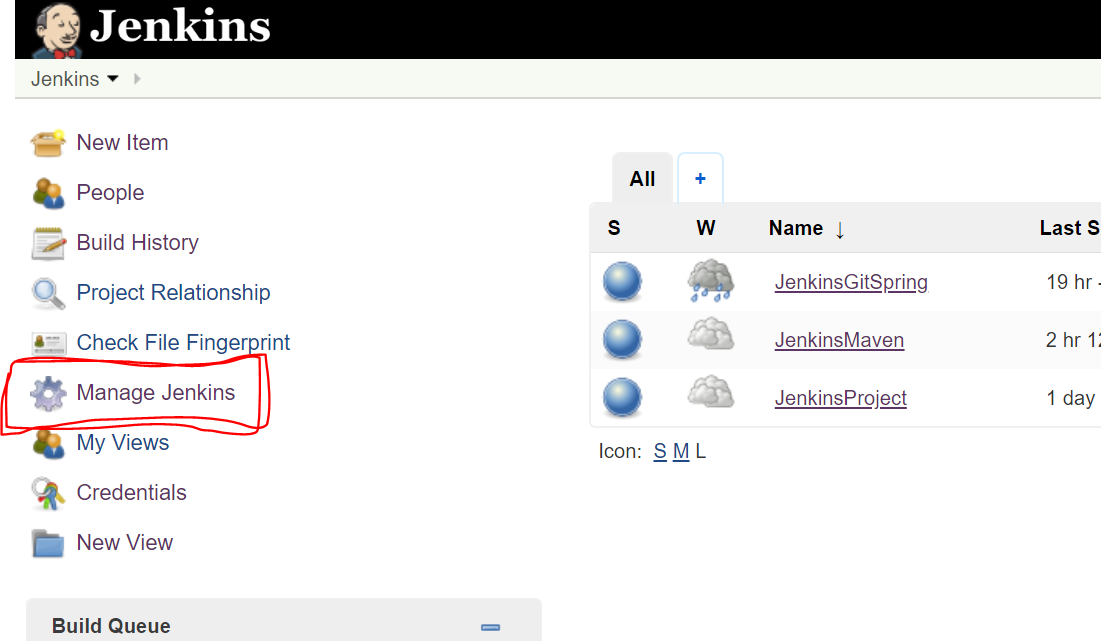


* Enter name and click on install automatically.
* Click on Add SonarQube Scanner.

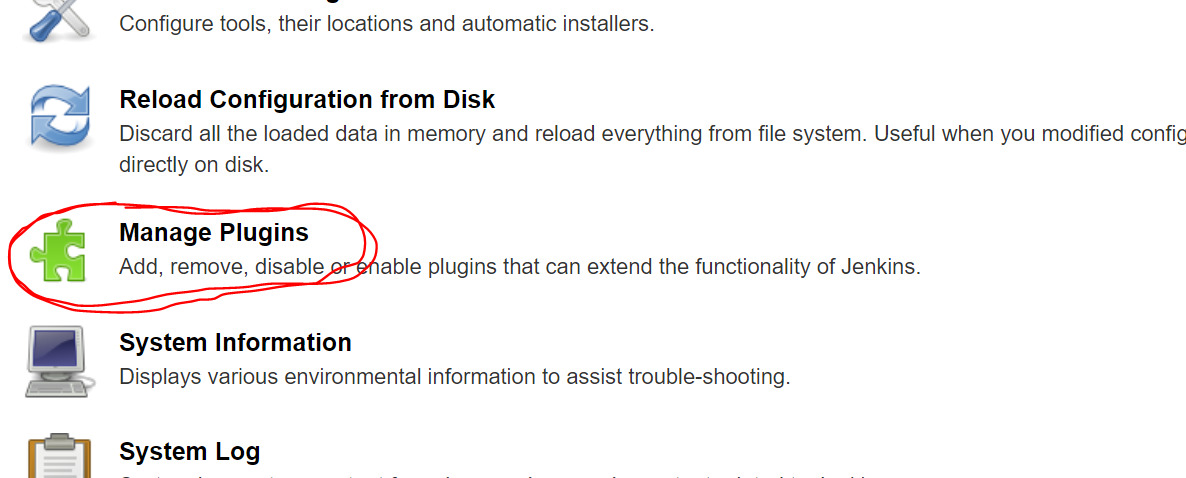


# **Step 3 :** To download plugins

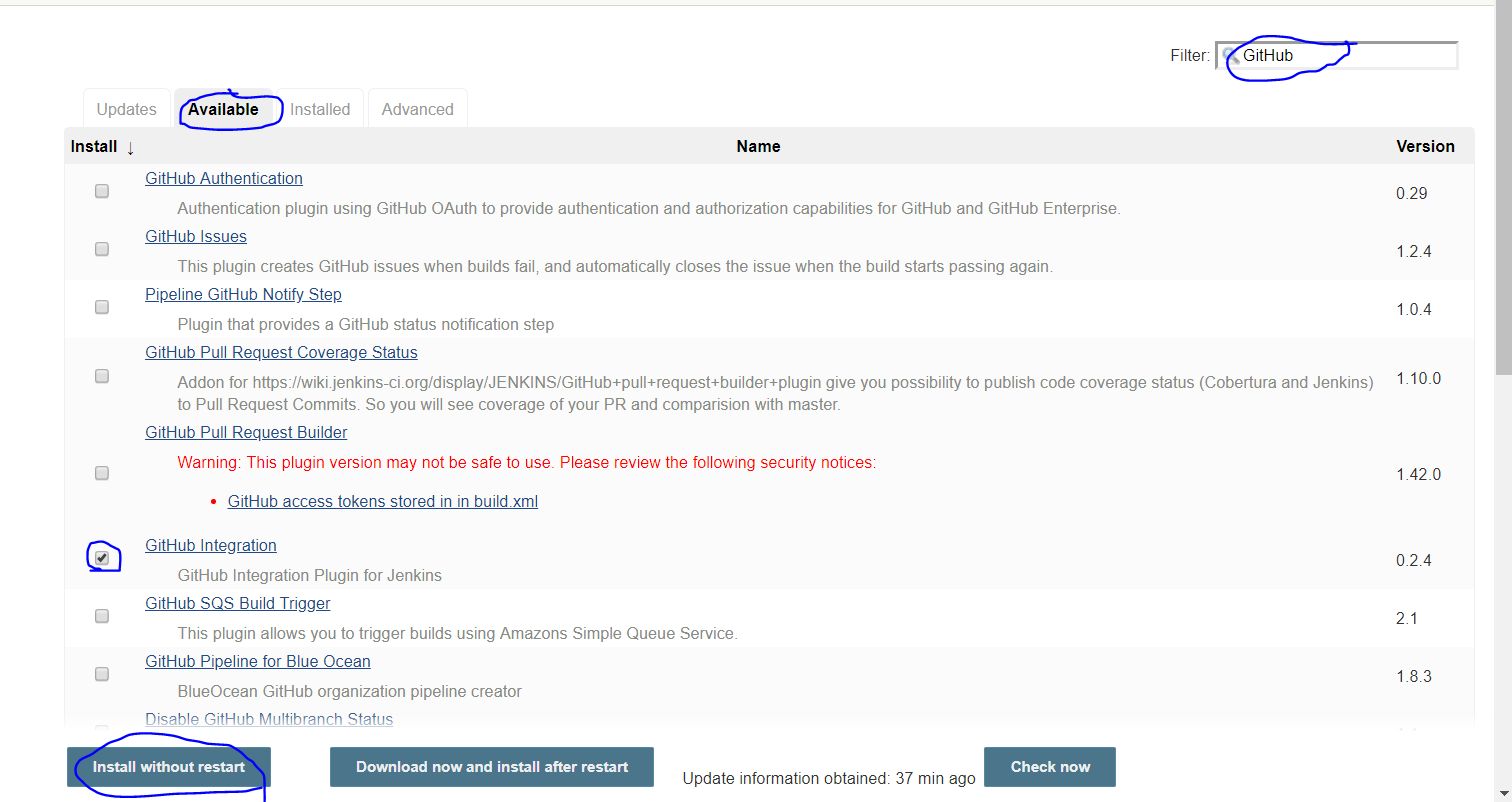
* Go to the **Manage Jenkins**



* Go to **Manage Plugins**



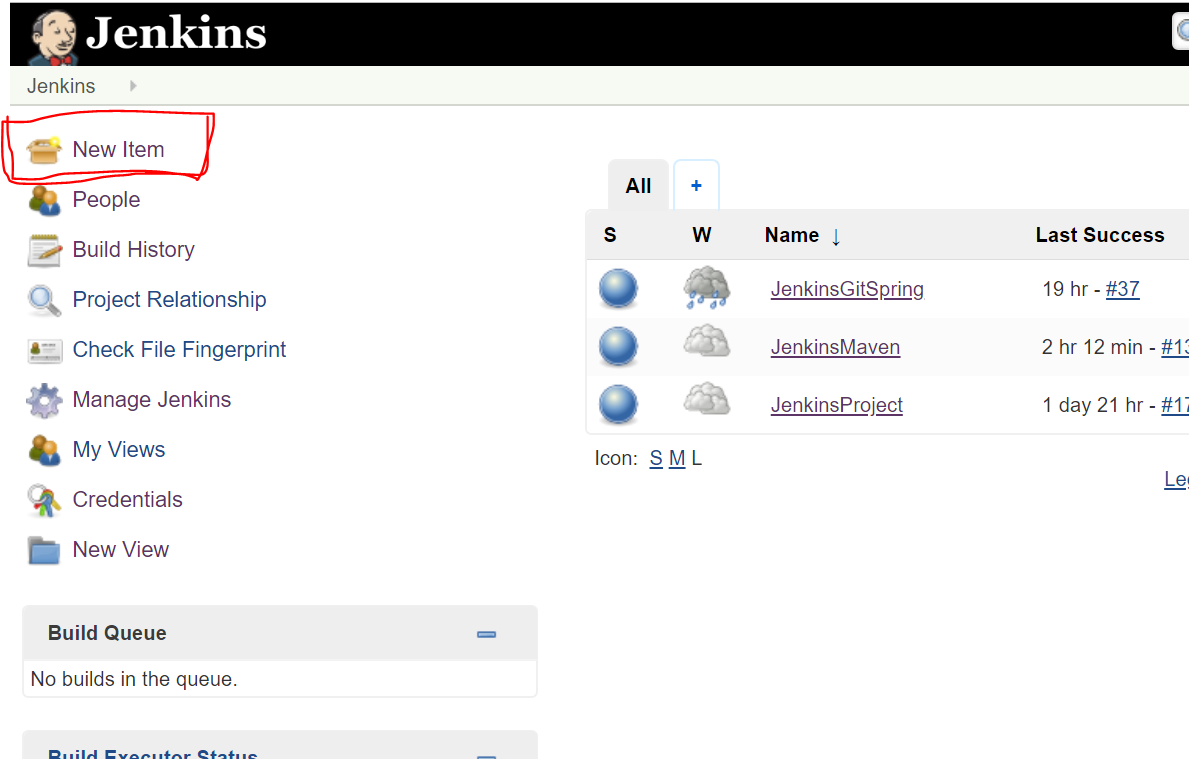
* Click on available(If there is check now button then click it)
* In filter type required plugin and search it
* Select required plugin and click on install without restart .



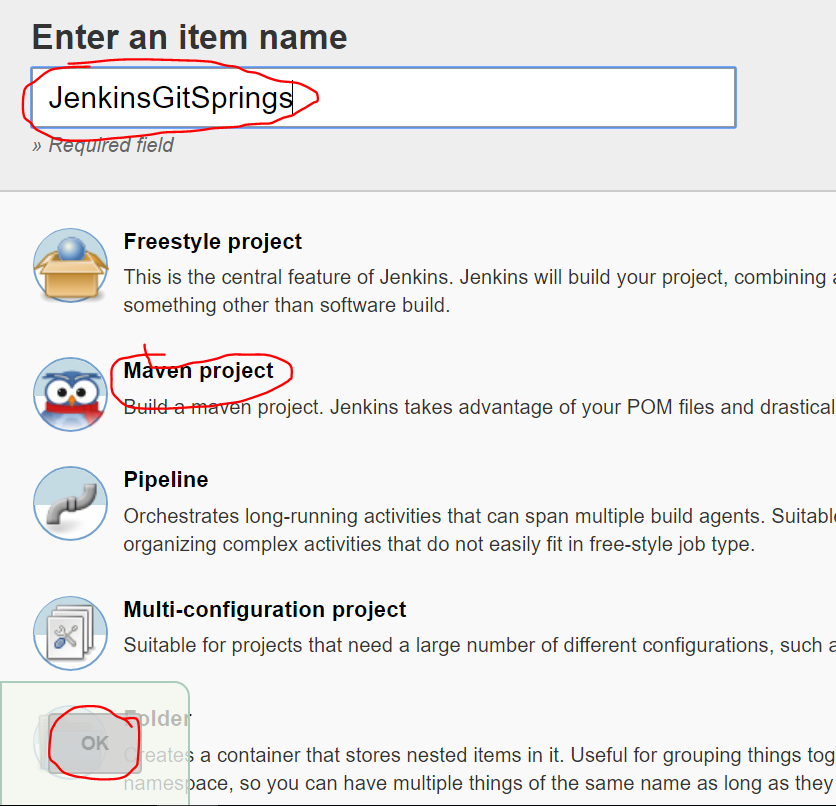
* Likewise download **Github Integration**,Github Authentication plugins, Maven Integration plugins, AWS Elastic Beanstalk Deployment,AWS elasticbeans publisher plugins, sonarQube scanner plugins .

# **Step 4**: Add new Application in jenkins

* Click on Add new item

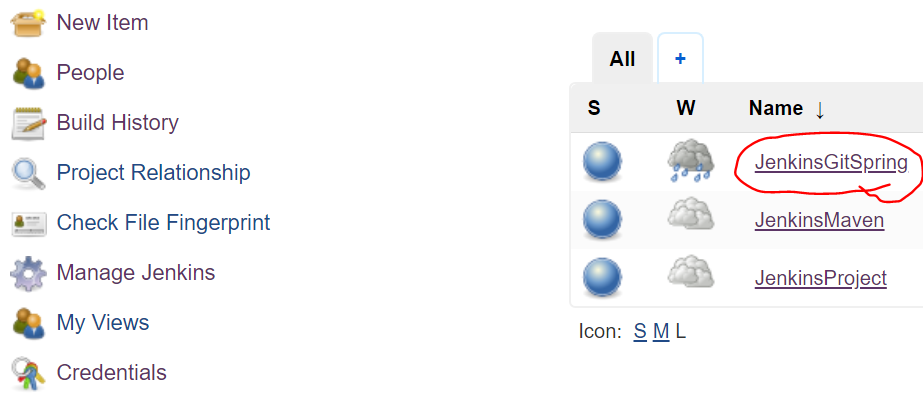


* Enter application name
* Select your application type. For example -maven project
* Click on OK

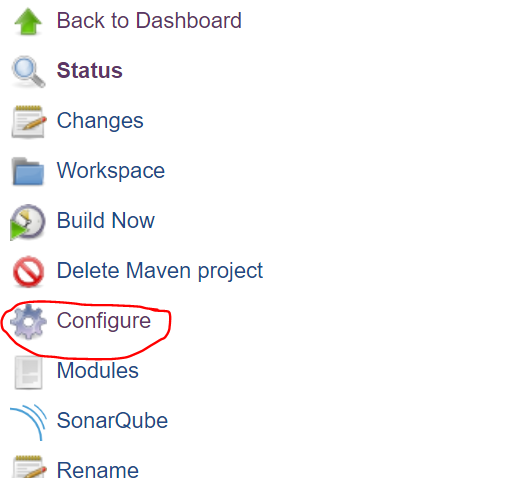


# **Step 5:** Configure Application

* Select your Application



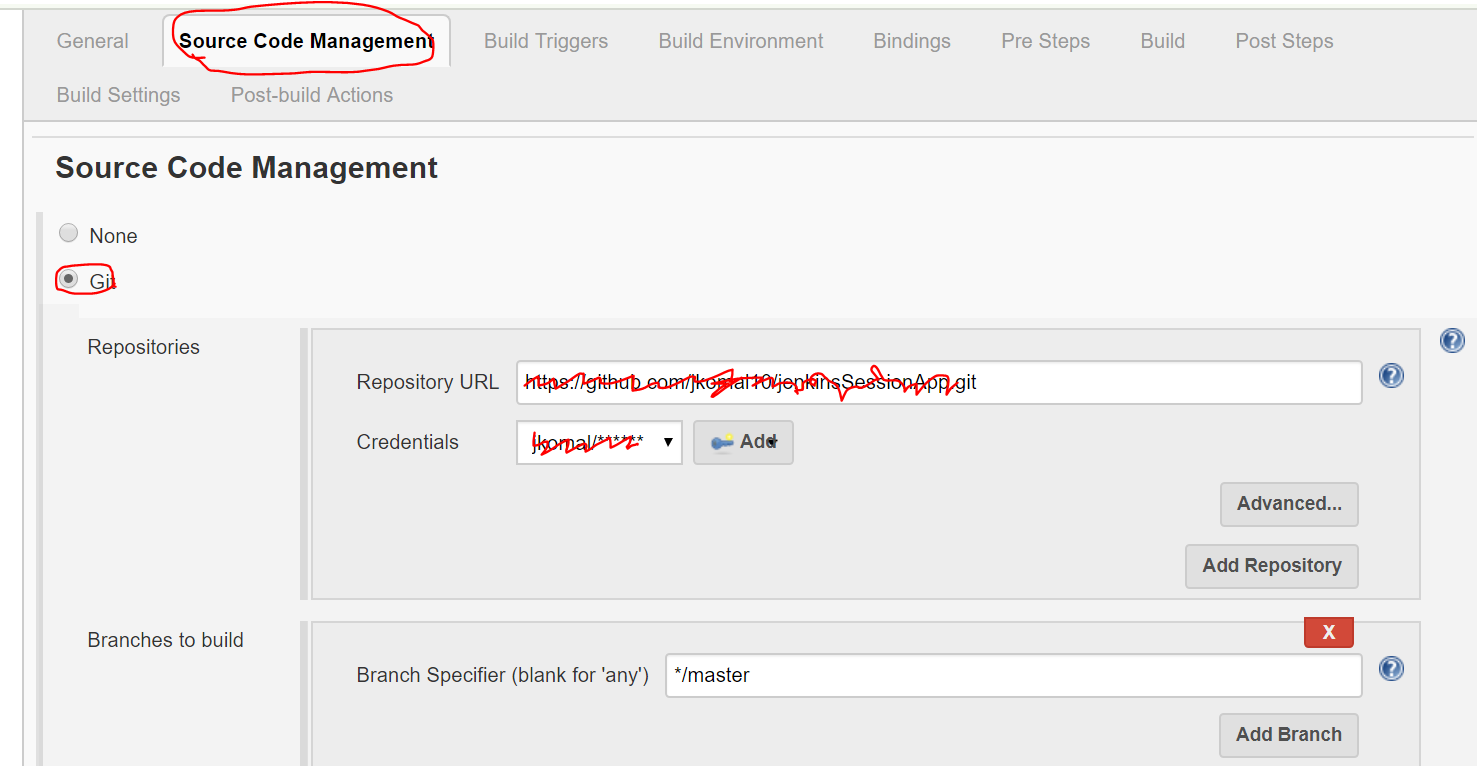
* Click on Configure



**Step 5.1** Click on Source code management->Git

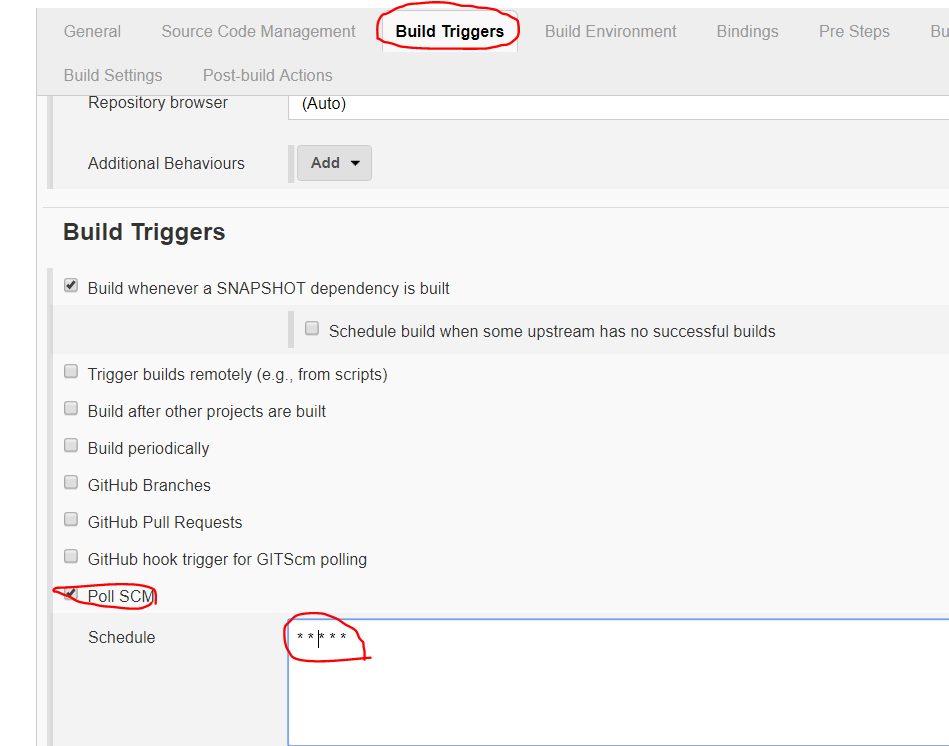
In repository URL give your git repository url and select your git credentials .

(To set credentials go to step 2.4.2 )

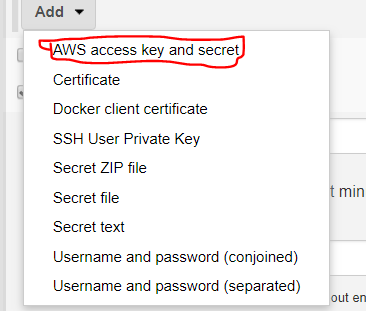
****

**Step 5.2** Click on Build trigger->Poll SCM->Enter \* \* \* \* \*(it means build application for every minute )

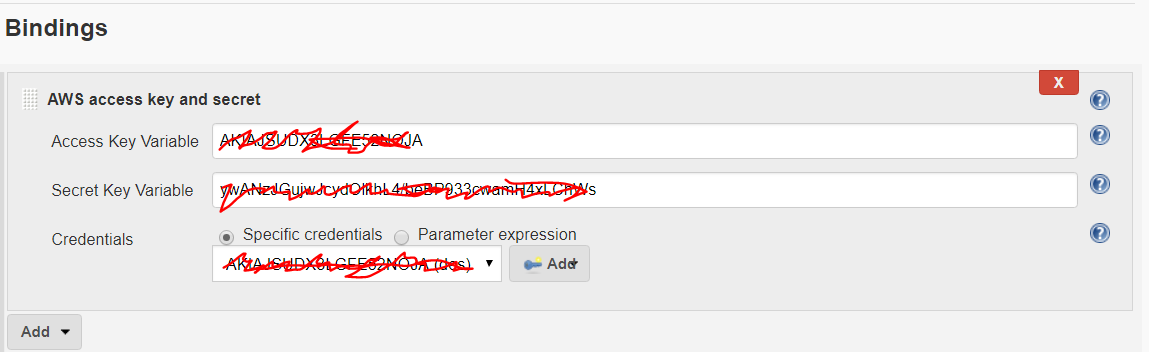
(Click on “?” icon near to description block to know more about how to set time to auto build)(Optional)



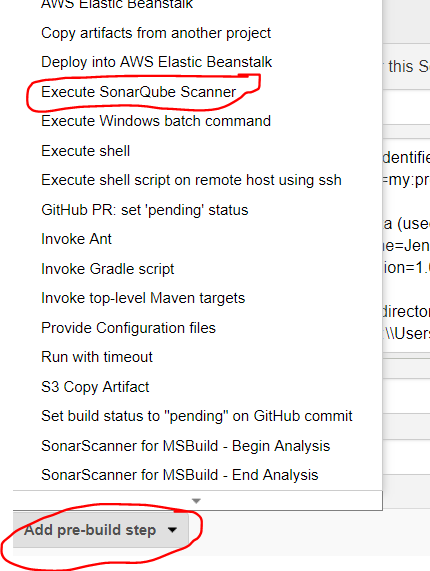
**Step 5.3** Select Building ->Add->AWS access key and secret access key



**Step 5.4** Enter your Access key variable and secrete key variable of your AWS account and select your credentials (Go to step 2.4.1)



**Step 5.5** Click on Pre-Steps->Add pre-build-steps->select execute SonarQube scanner(for SonarQube setup go to step 2.5)



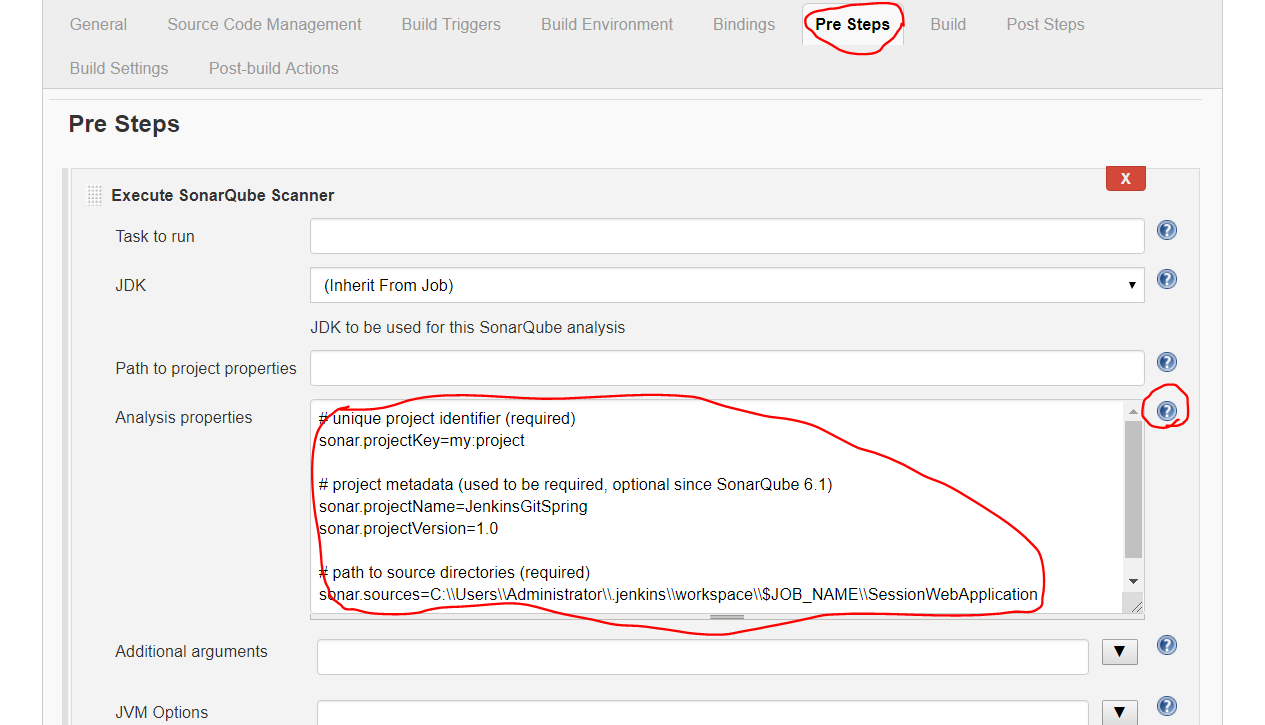
**Step 5.6** In Analysis properties Provide following information(for more details click on ? symbol)

Sonar.projectkey=(any key)

Sonar.projectName=your project name

Sonar.project.version=1.0

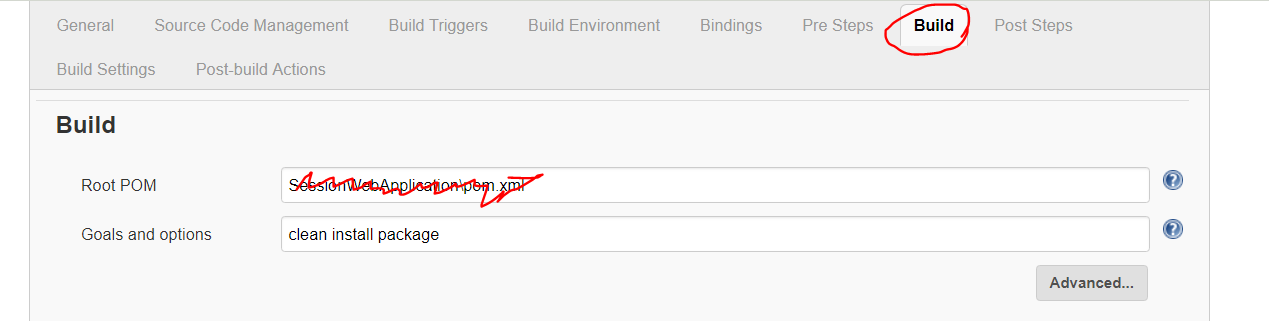
Sonar source=absolute path of your project from Jenkins workspace.



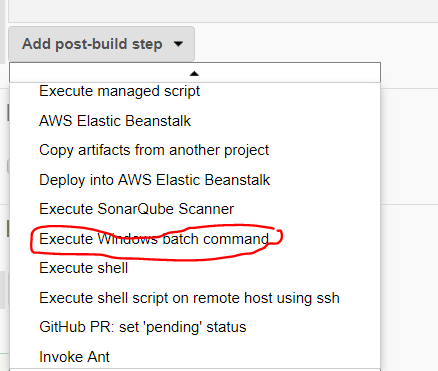
**Step 5.7** Click on Build

Enter your pom.xml directory in Root POM

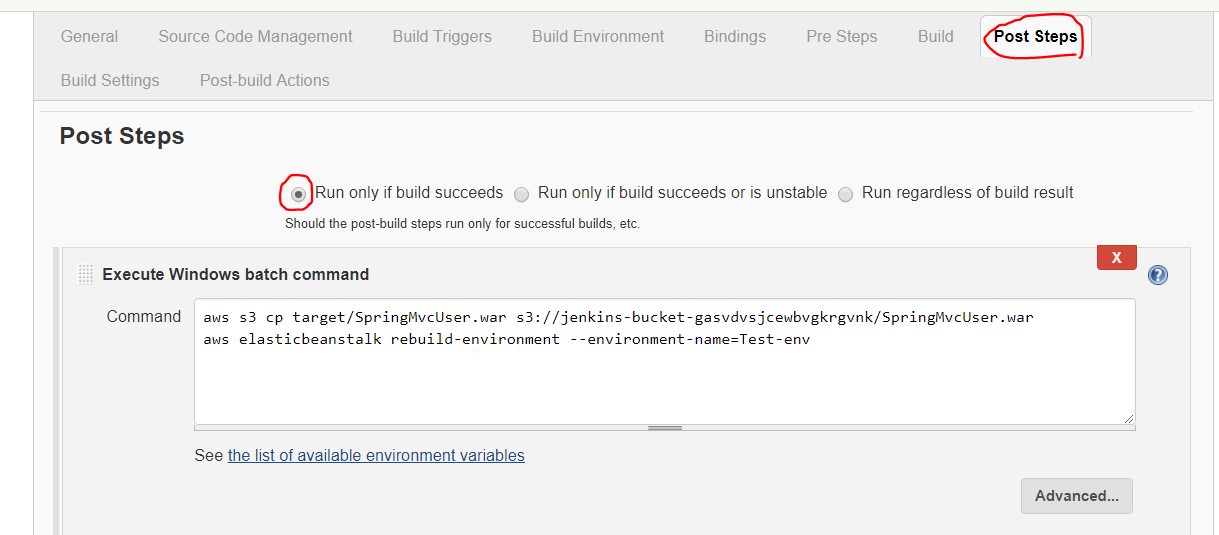
In goals and option enter clean compile package or other options as per your requirement.



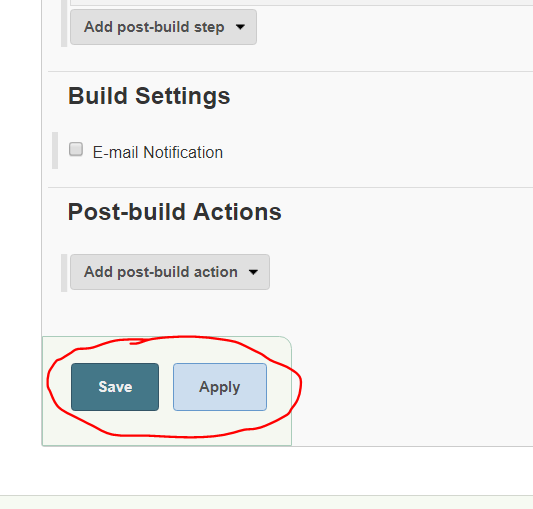
**Step 5.8** Click Post Steps->Add post-steps->select execute windows batch command



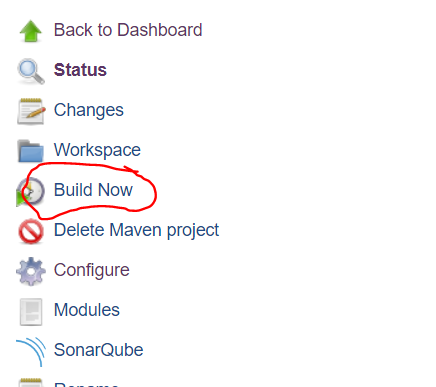
**Step 5.9** Enter your commands



# **Step 6:** Click on apply and save



# **Step 7:** Click on build now



**Step 8:**  If it build successfully it will be in blue color and on failure it will be in red as shown in following diagram. Right click and select show console output

