**Jenkins**

**Prerequisites:**

**We need JDK installation**

**If you want to deploy your application on AWS elastic beanstalk then you must have AWS Credentials**

**Jenkins Installation**

**Step1:** Download Jenkins war from following link

[**https://jenkins.io/**](https://jenkins.io/)

**Step 2:** Keep Jenkins.war in any folder.

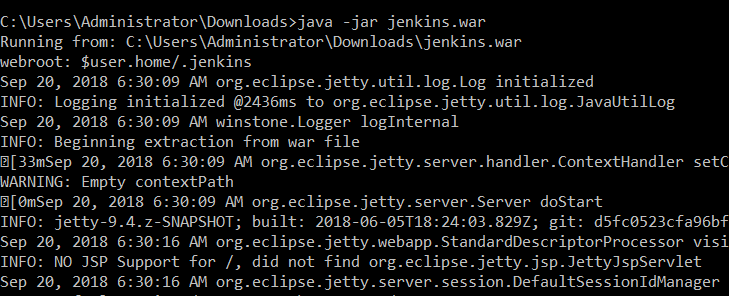
**Step 3**: Open CMD and go to folder where you have kept your Jenkins.war file

**Step 4:** run following command to run Jenkins

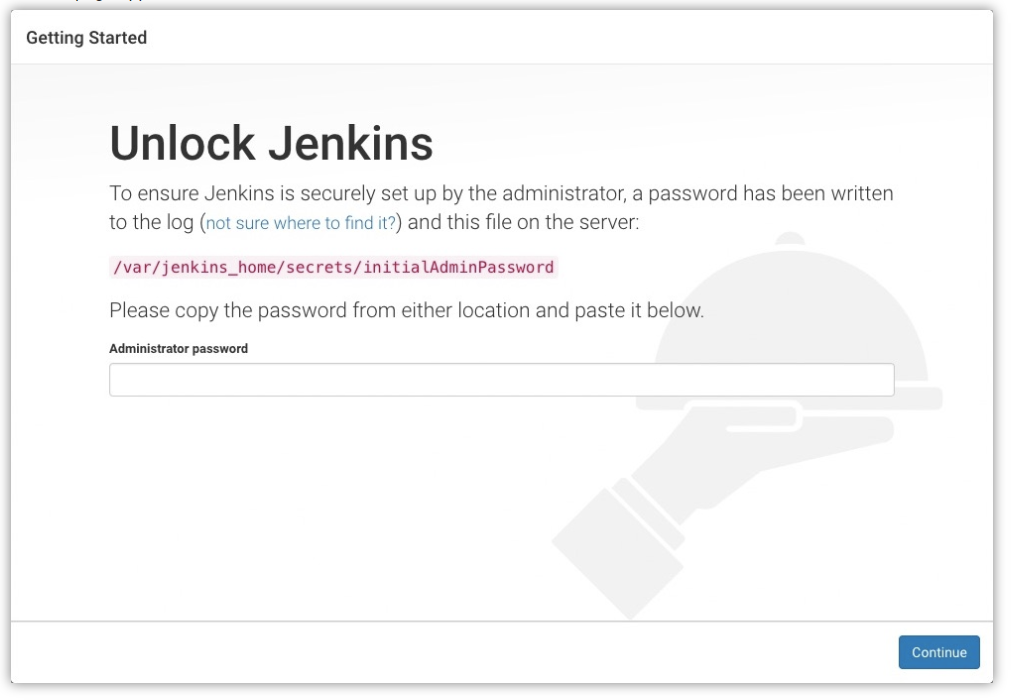
java -jar jenkins.war

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

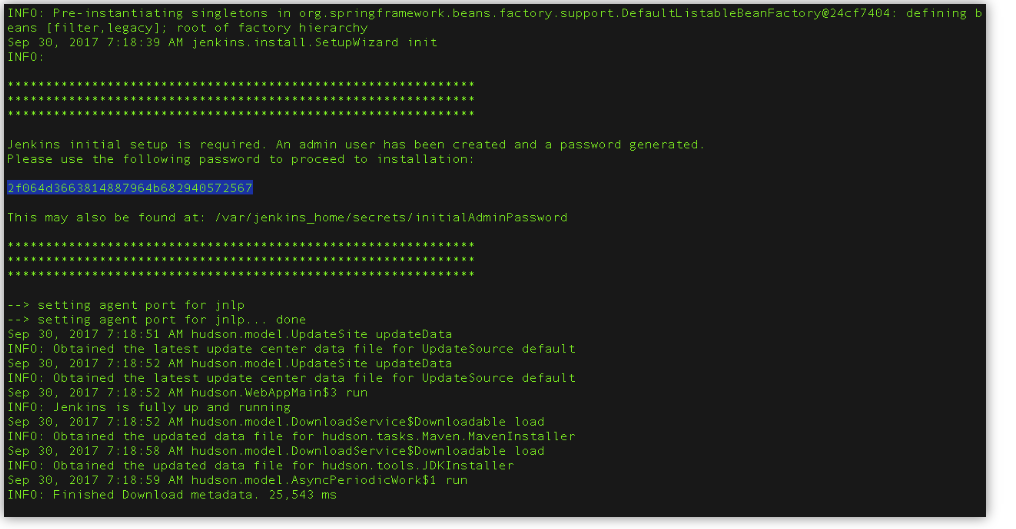
Java -jar jenkins.war httpPort=8181



**Step 5:** Start Jenkins by giving [**localhost:8080**](https://jenkins.io/)



**Enter admin password(it will get from cmd)**



**Step-6:** Create User by giving username and password and click on button create user.

Now you are ready with Jenkins installation.

**Step 6:** Some Required Configuration

**Step 6.1**

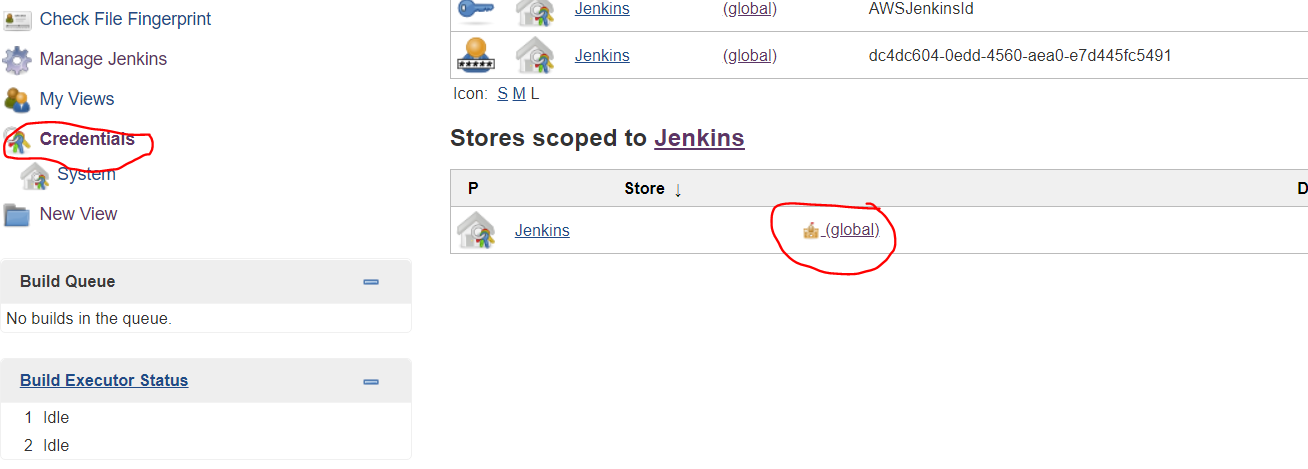
* To Add Maven Application we need to first add maven plugins in Jenkins(Go to step 7)
* We need to download maven and extract it some location
* Go to manage Jenkins->Global tool configuration->Maven->Add Maven
* In maven name give you maven with version and in maven home give your maven path
* If you have not download maven then simply click on install automatically
* Click on save.

**Step 6.2**

* To add jdk , first add jdk plugins in Jenkins (go to step 7)
* We need to download jdk and extract it some location
* Go to manage Jenkins->Global tool configuration->JDK->JDK installation
* In jdk name give your jdk with version and in java home give your jdk path
* If you have not download jdk then simply click on install automatically
* Click on save.

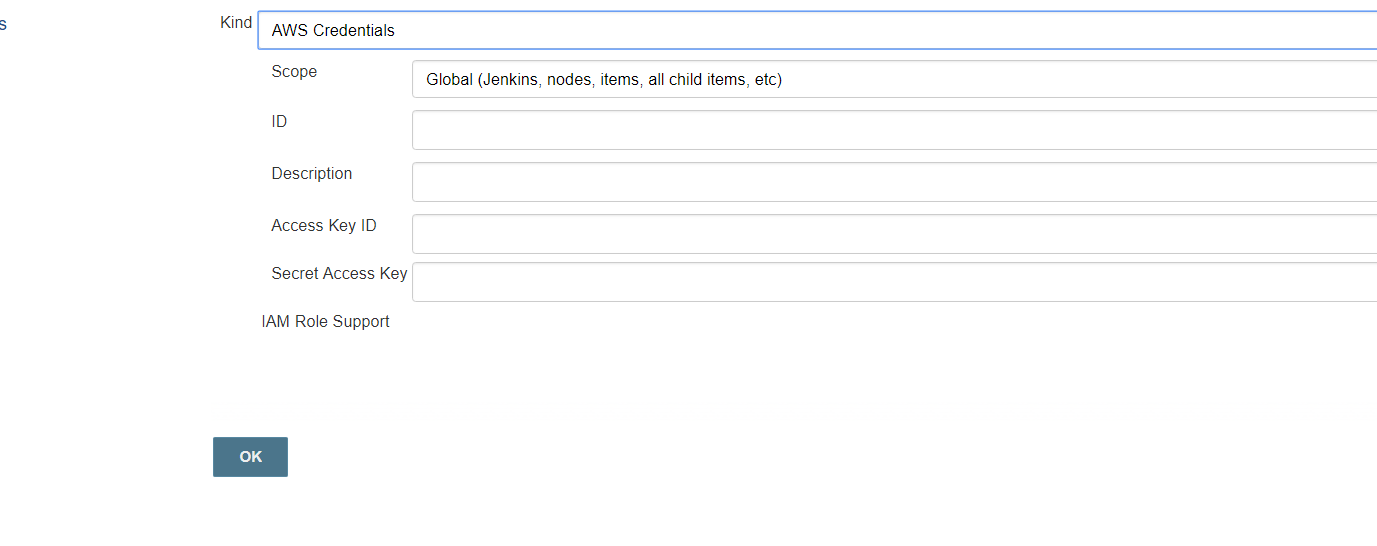
**Step 6.3** To add git and git credentials

* Go to manage Jenkins->manage plugins->search for git plugins and click on install without restart.
* Select Credentails
* Select dropdown near global and click on add credentials



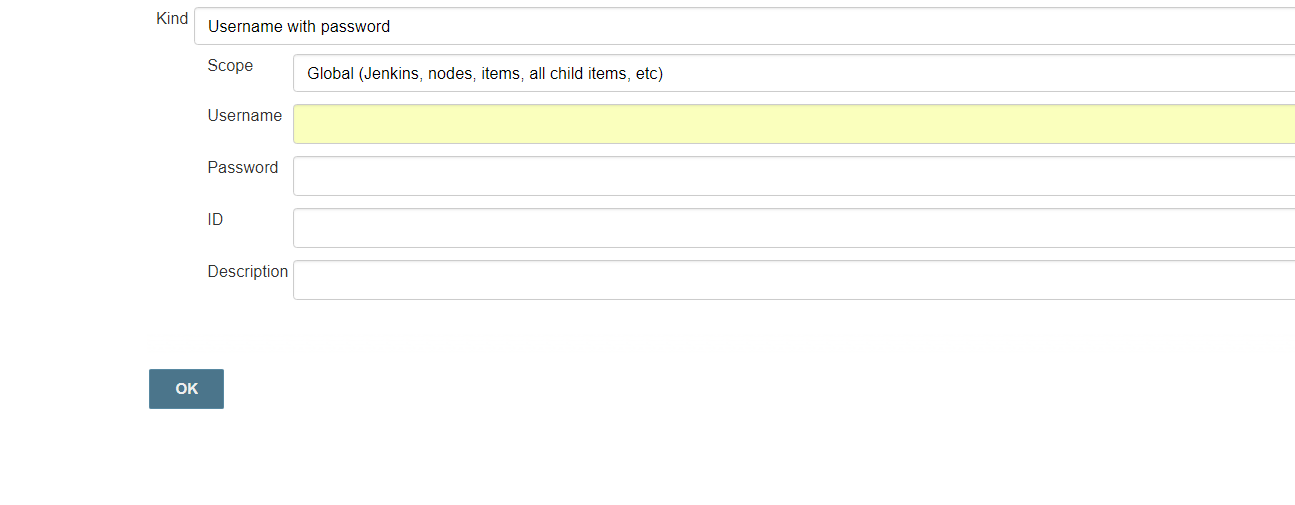
**Step 6.3.1** To add aws credentials select dropdown at kind and select AWS credentials

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



**Step 6.3.2** To add Git credentials select dropdown at kind and select username and password

* Enter your git username and password and click on OK



**Step 6.4: 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 download SonarQube and extract it

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

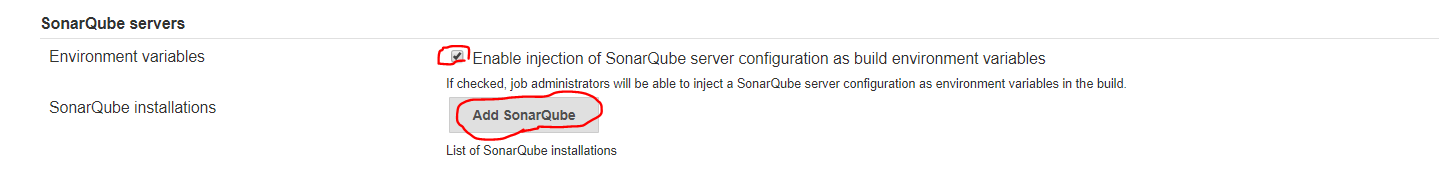
* After extract SonarQube run StartSonar.exe
* Go to browser and run localhost:9000(By default SonarQube run on 9000 port)
* Login with admin credentials

-Username= admin

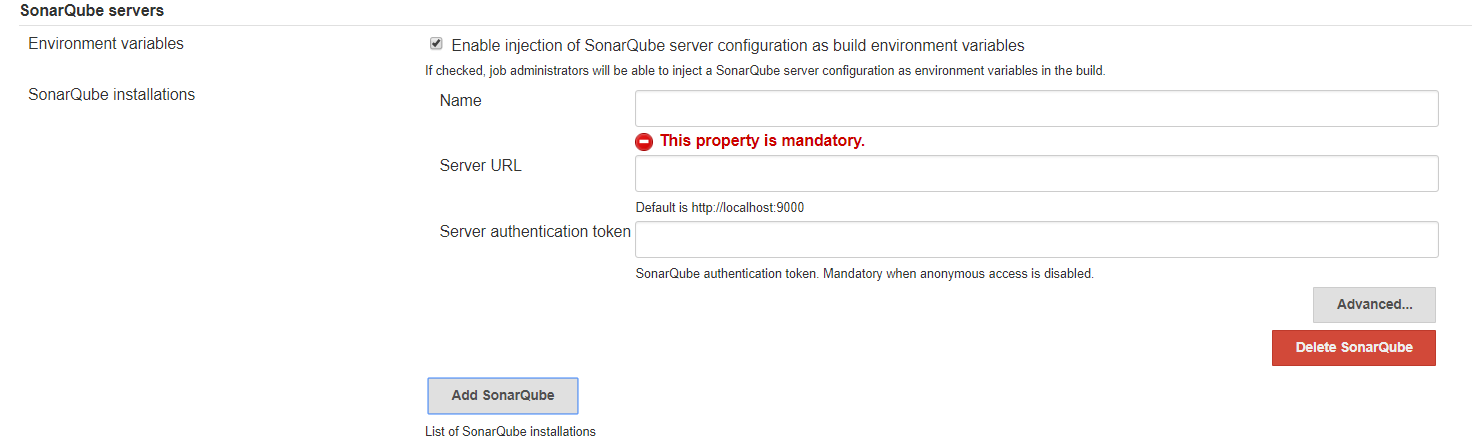
-Password= admin

Copy token generate on screen after login with admin credentials

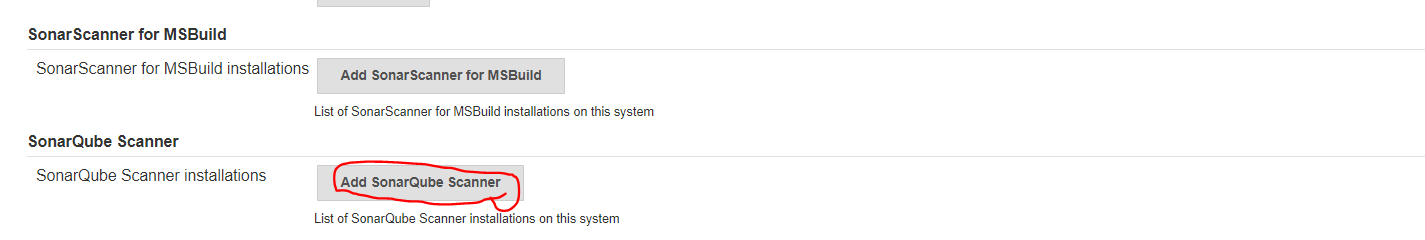
* Go to manage Jenkins->configure system->SonarQubeService->Check on Enable injection of SonarQube server configuration as build environment variables and click on Add SonarQube.



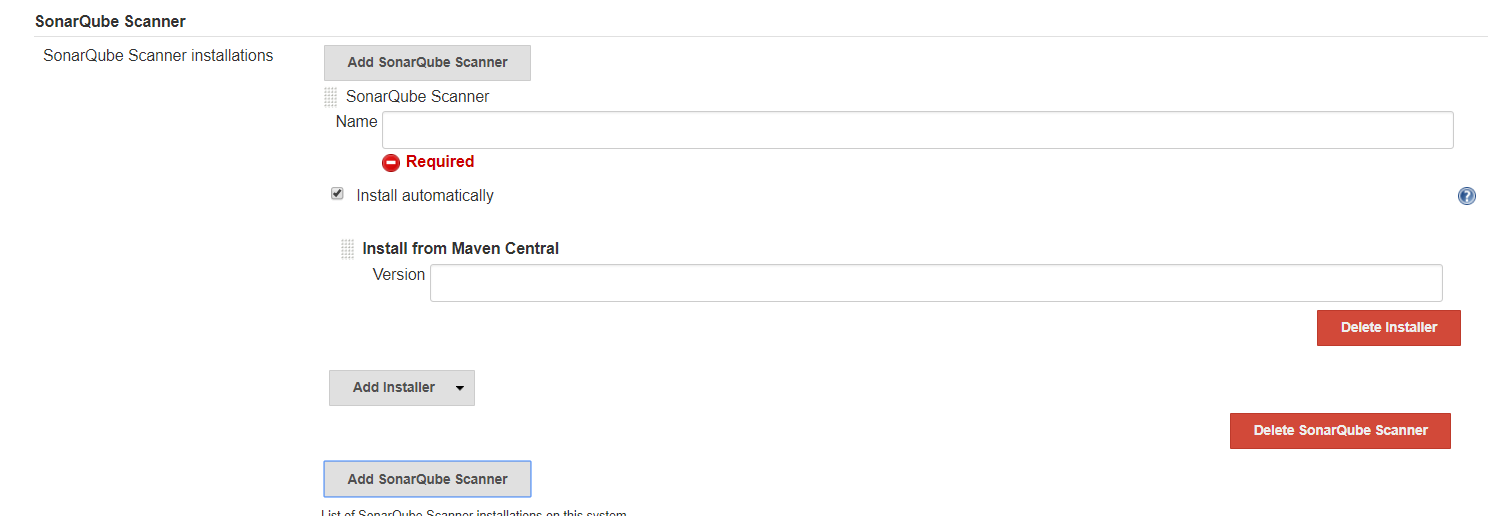
* Enter any name, Enter localhost:9000 in Server URL and paste token which u obtained in last step.
* Click on Add SonarQube
* Click on save button



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



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



**Step 6.5: To add shell**

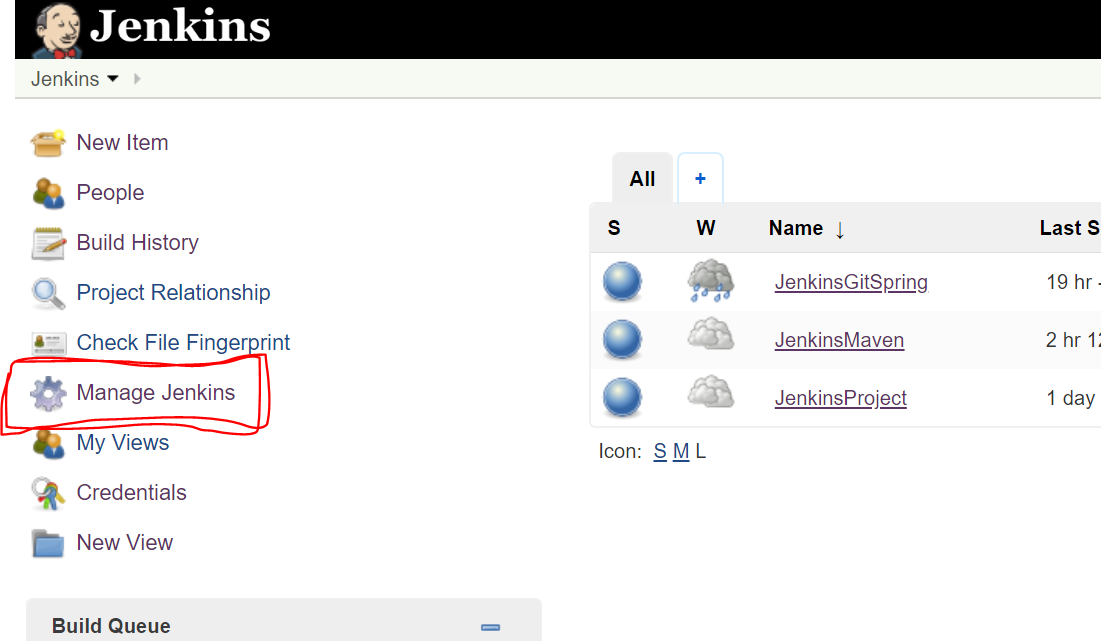
* Go to manage Jenkins->global tool configuration->shell->enter your shell or cmd path.\
* Go to cmd and write aws configure
* Then write id
* \*\*\*\*\*\*ZzJTAz5d
* \*\*\*\*\*\*IAJ3BTT

**Step 6.6: To add git**

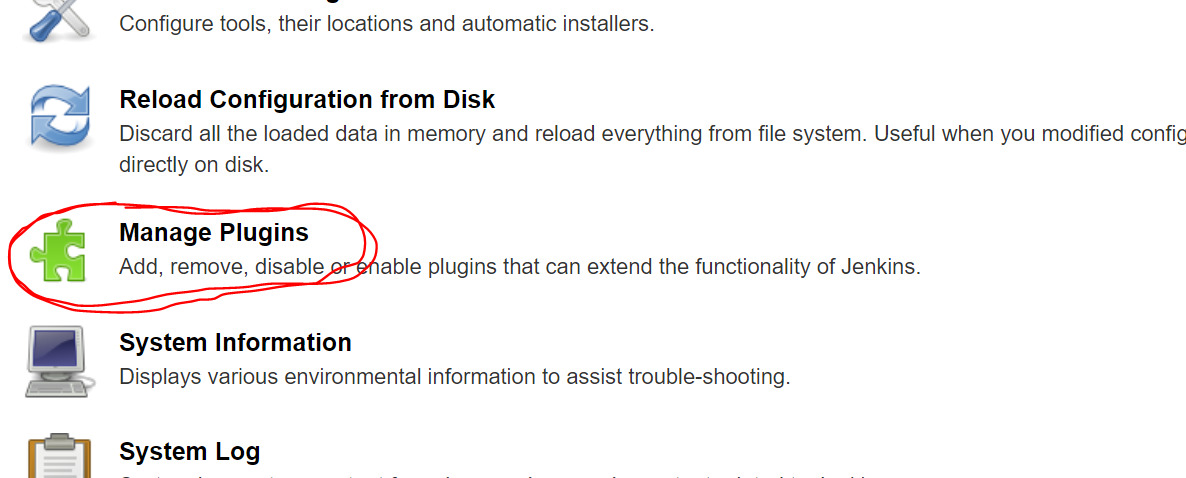
* Go to manage Jenkins->global tool configuration->Git ->in path enter your git path upto bin and after that concat it with \git.exe

**Step-7:** To download plugins

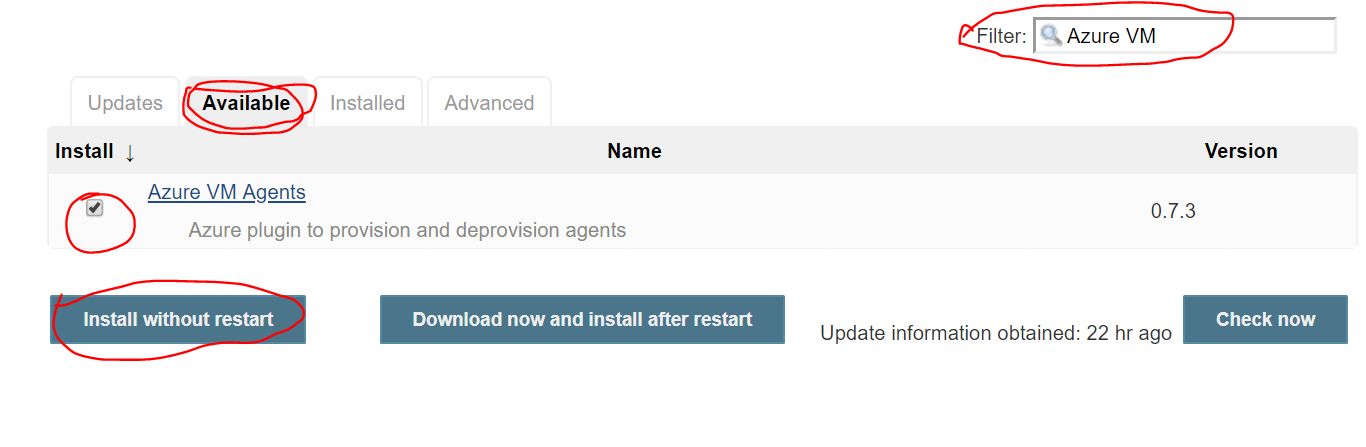
* Go to the **Manage Jenkins**



* Go to **Manage Plugins**



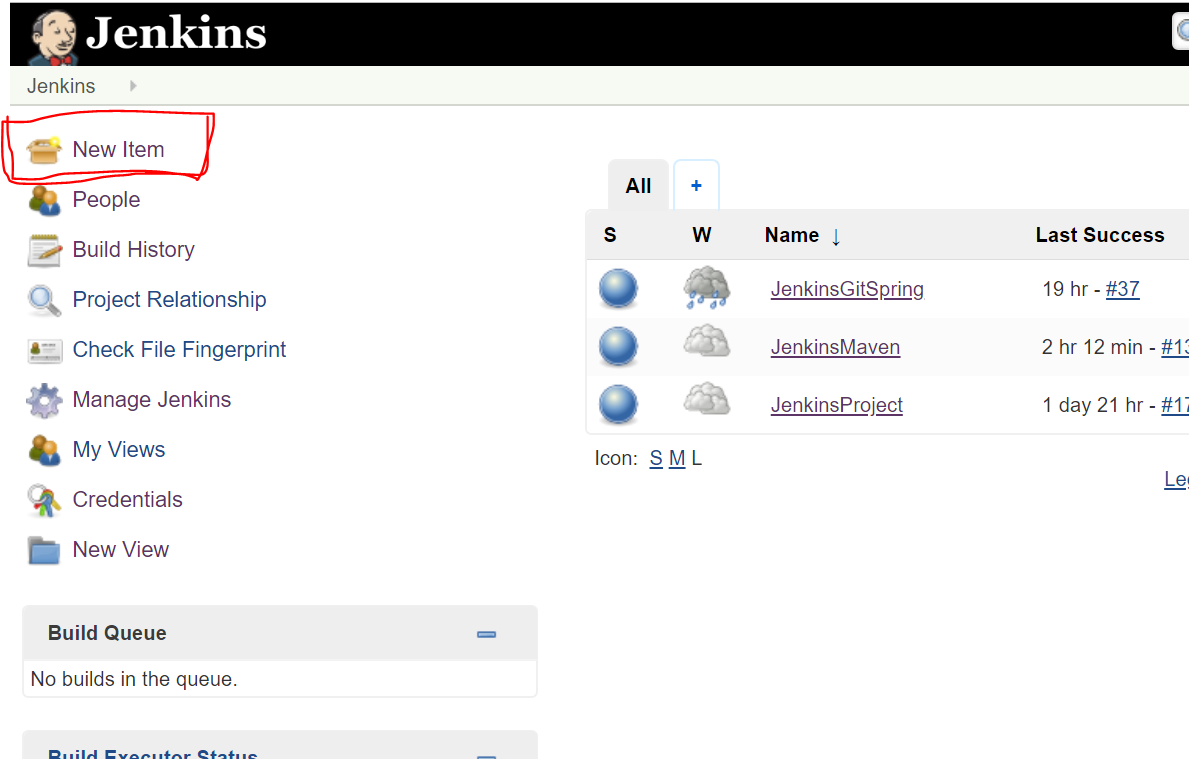
* Click on available
* In filter type required plugin and search it
* Select required plugin and click on install without restart or restart after download



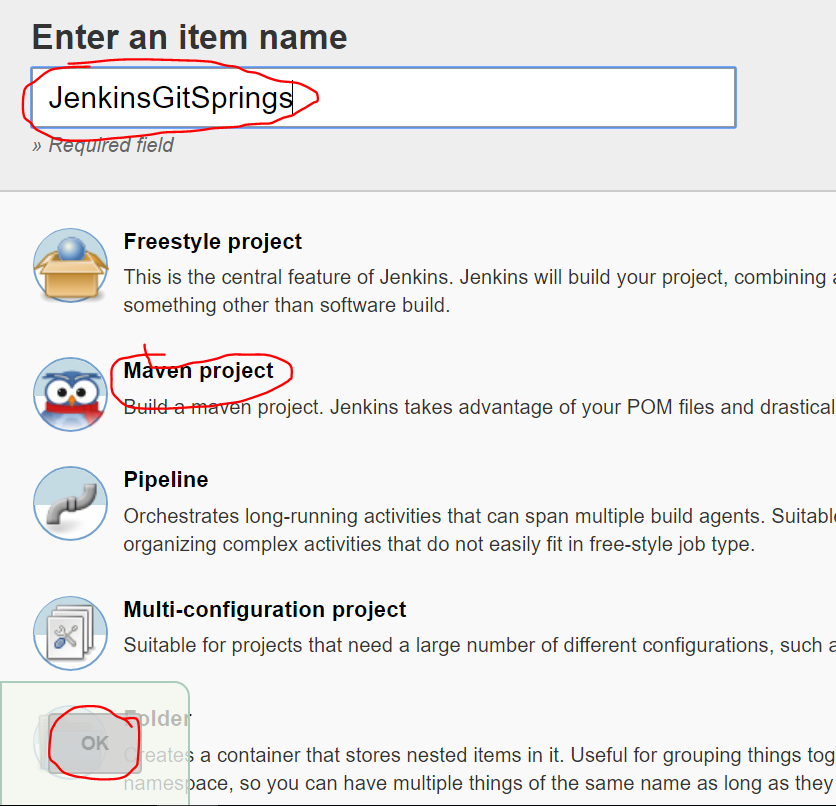
* Download **Github Integration**, **Maven Integration** , JDK AWS CLI & other default plugins.

**Step 8**: Add new Application in jenkins

* Click on Add new item

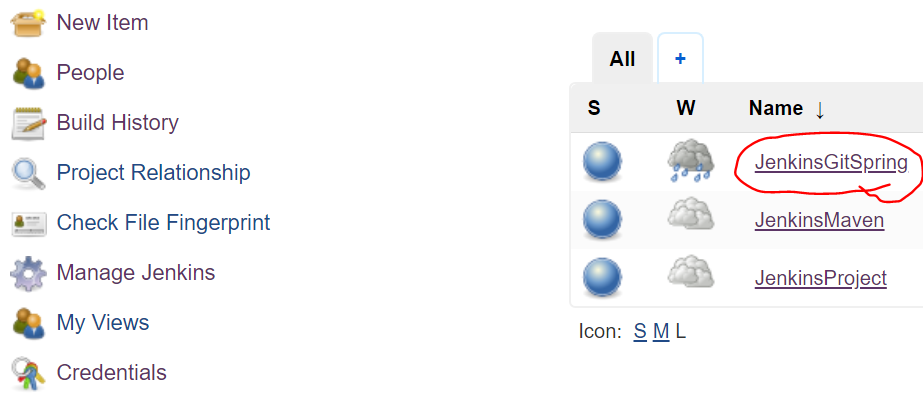


* Enter application name
* Select your application type. For example -maven project
* Click on OK

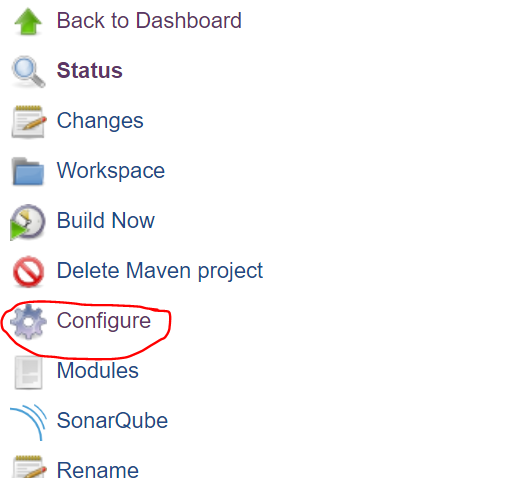


**Step 9:** Configure Application

* Select your Application



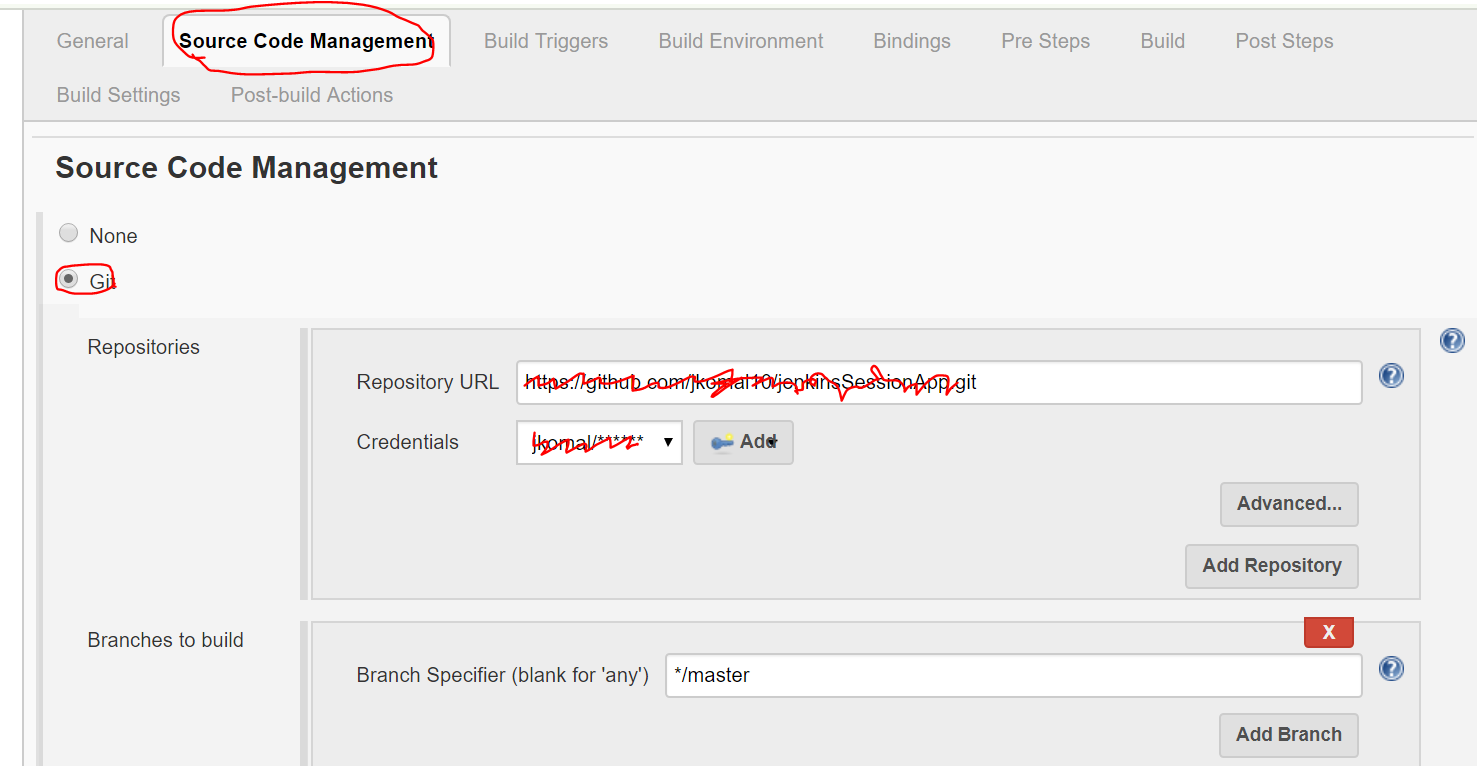
* Click on Configure



**Step 9.1** Click on Source code management->Git

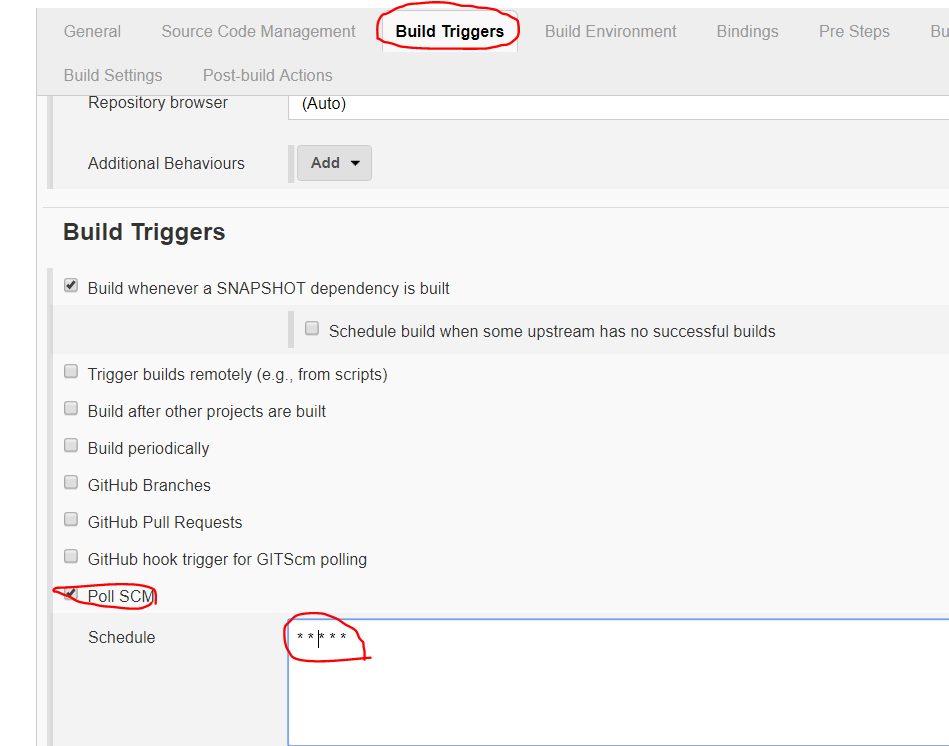
In repository URL give your git repository url and select your git credentials .

(To set credentials go to step 6.3 )

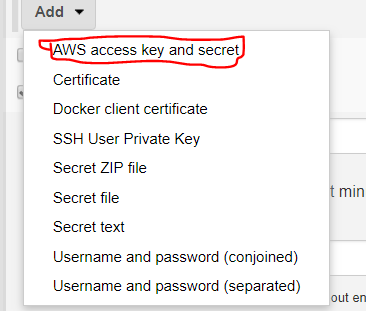
****

**Step 9.2** Click on Build trigger->Poll SCM->Enter \* \* \* \* \*(it means build application for very minute )

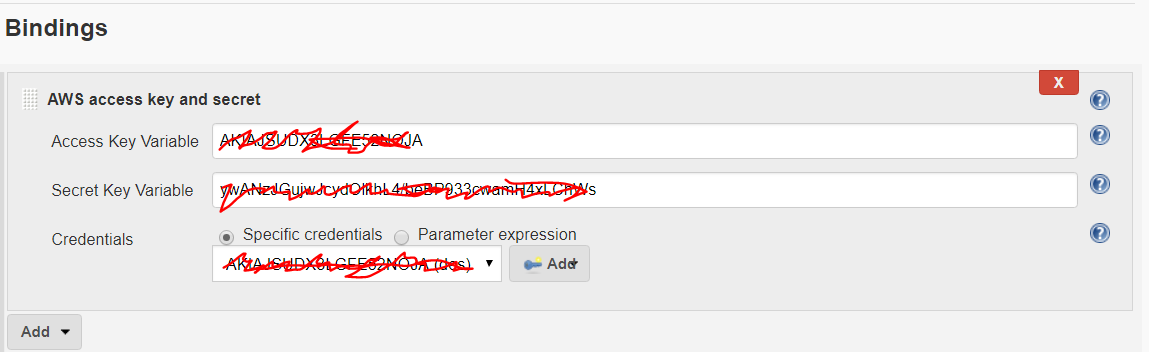
(Click on ? icon near to description block to know more about how to set time to auto build)



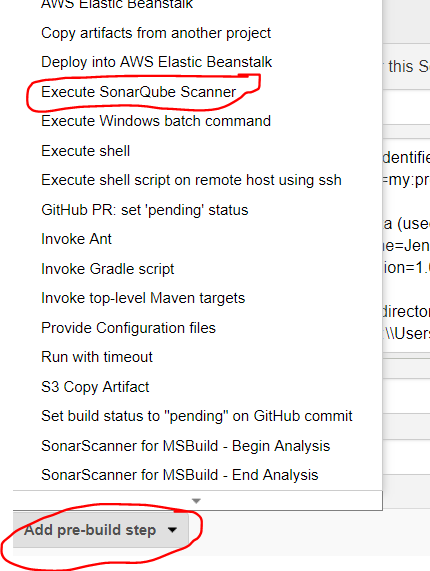
**Step 9.3** Select Building ->Add->AWS access key and secret



**Step 9.4** Enter your Access key variable and secrete key variable of your AWS account and select your credentials (Go to step 6.3.1)



**Step 9.5** Click on Pre-Steps->Add pre-build-steps->select execute SonarQube scanner(for SonarQube setup go to step 6.4)



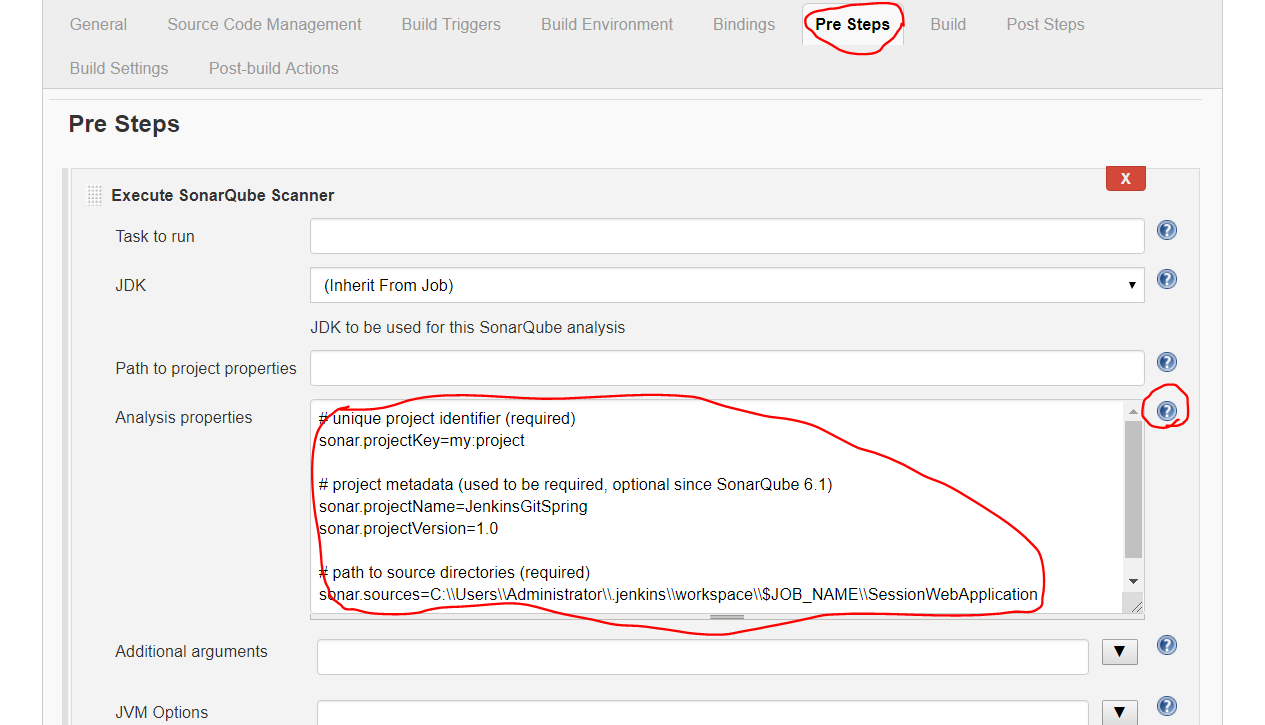
**Step 9.6** In Analysis properties Provide following information(for more details click on ? symbol)

Sonar.projectkey=(any key)

Sonar.projectName=your project name

Sonar.project.version=1.0

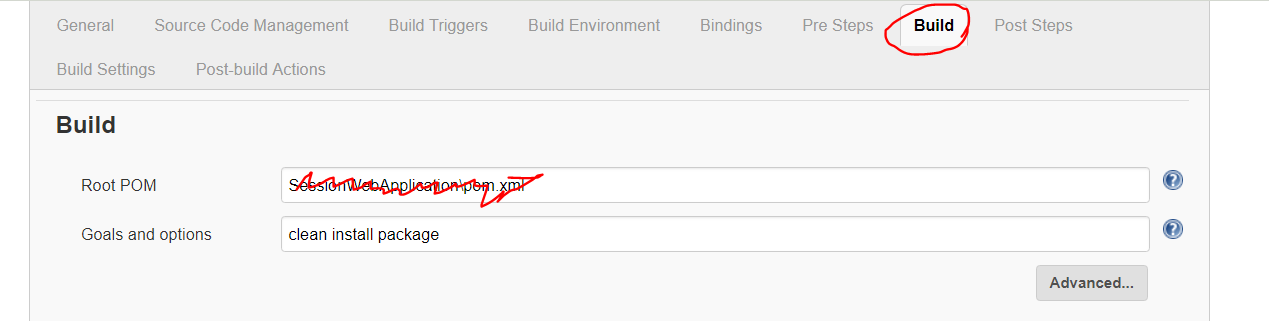
Sonar source=absolute path of your project from Jenkins workspace.



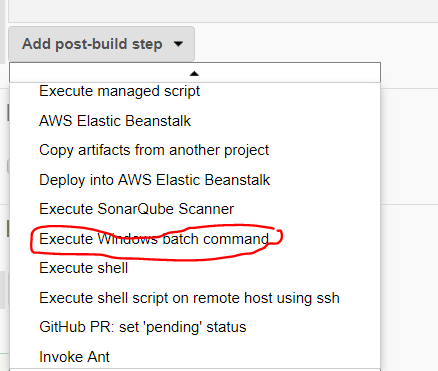
**Step 9.7** Click on Build

Enter your pom.xml directory in Root POM

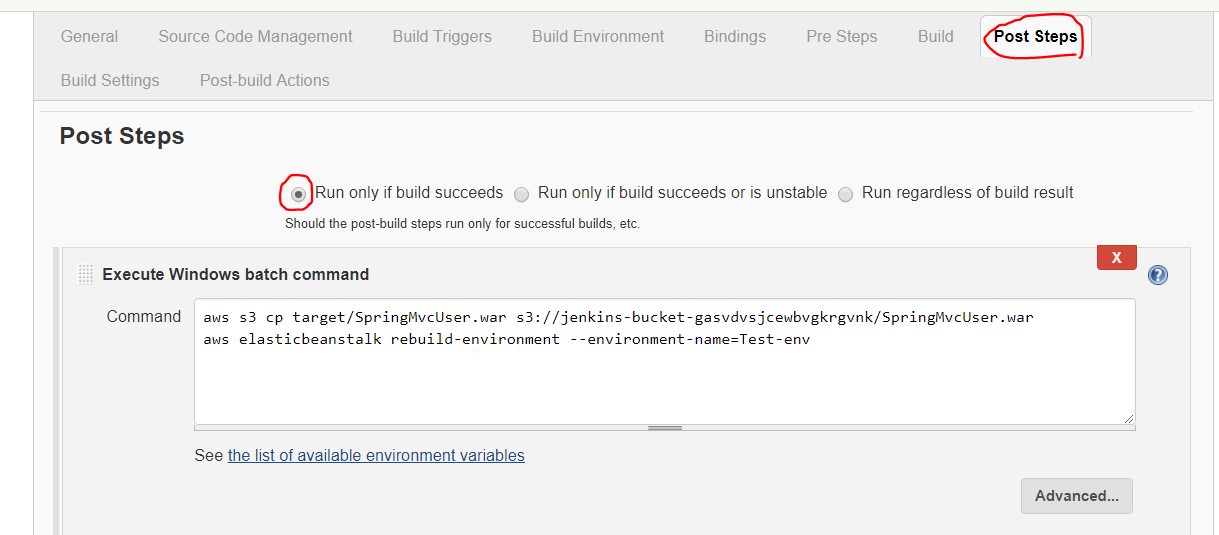
In goals and option enter clean compile package or other options as per your requirement.



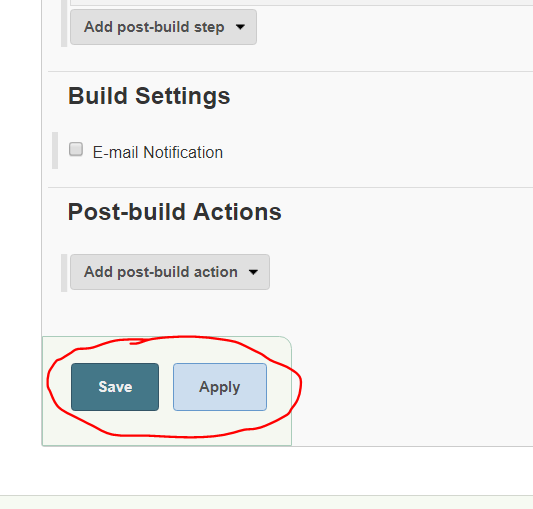
**Step 9.8** Click Post Steps->Add post-steps->select excute windows batch command



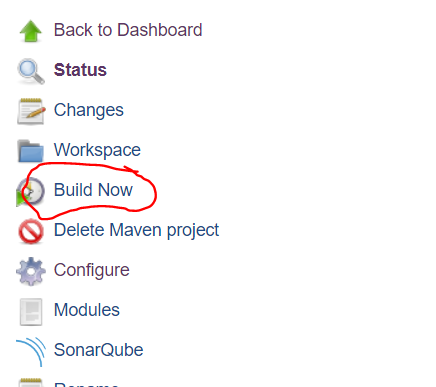
**Step 9.9** Enter your commands



**Step 10** Click on apply and save



**Step 11** Click on build now



**Step 12** 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

