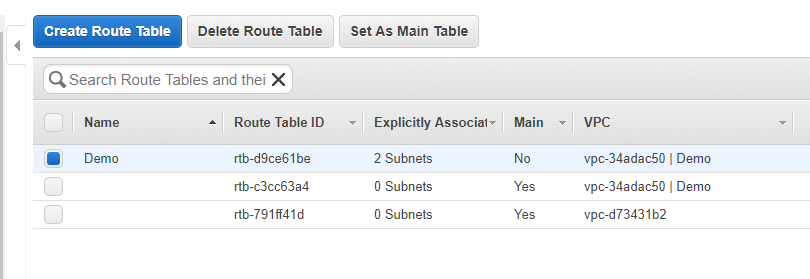
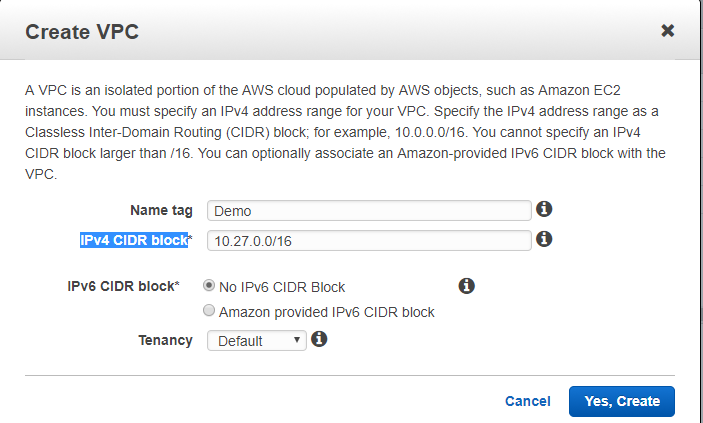
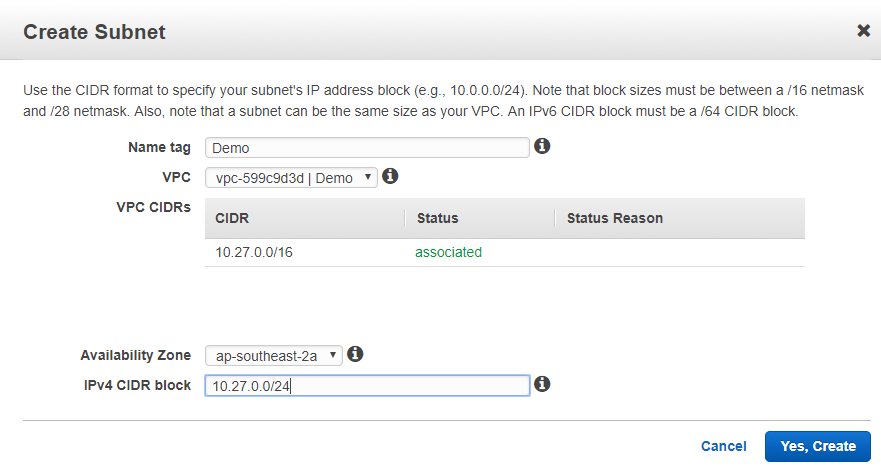
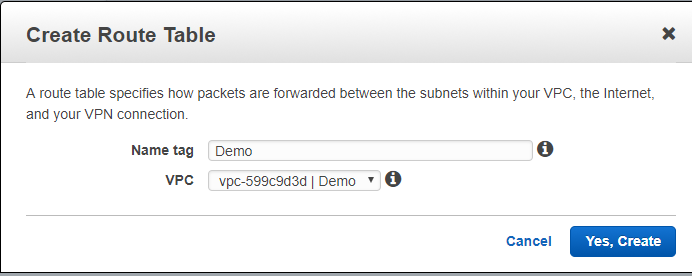
How to create an VPC.

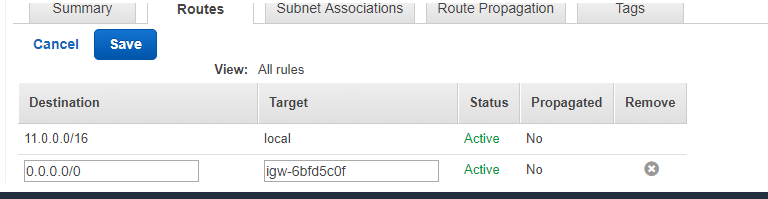
* Click on VPC service and select your VPC.
* Next click on create VPC



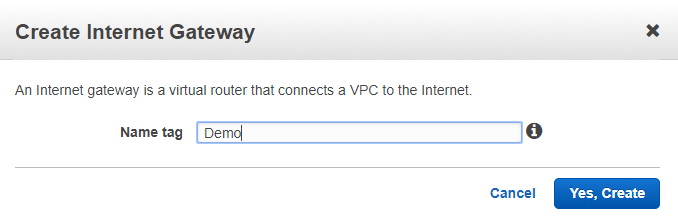
* Enter the Name tag and IPv4 CIDR block and click on create

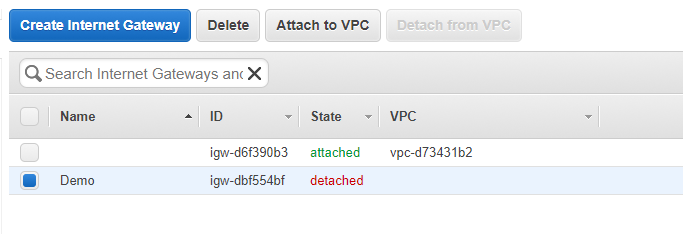


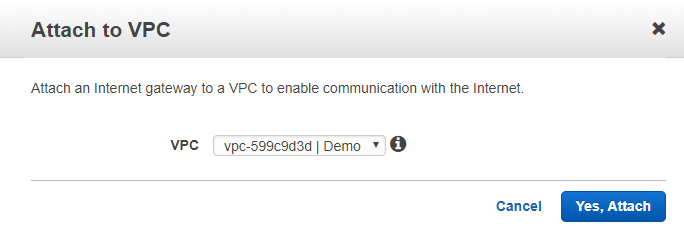
* Next click on subnets and create an subnet.
* Next create Route tables and click on create Route tables.
* And Select your VPC shown in route table and click on create.
* In Route table click on click on Routes and select an edit option.
* Add another role and select the target and click on save.



* And in subnet Associations click on edit and attach the subnet and save.
* Next create an Internet Gateway shown in your VPC.



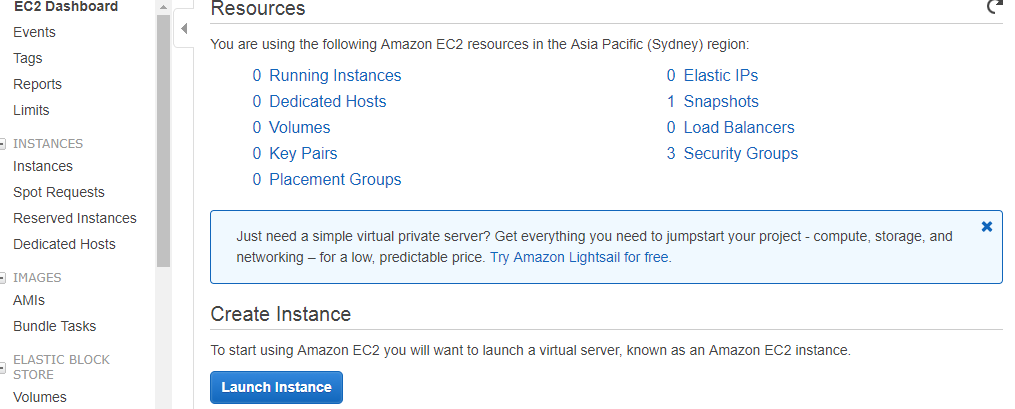
* And Attach to VPC shown above.

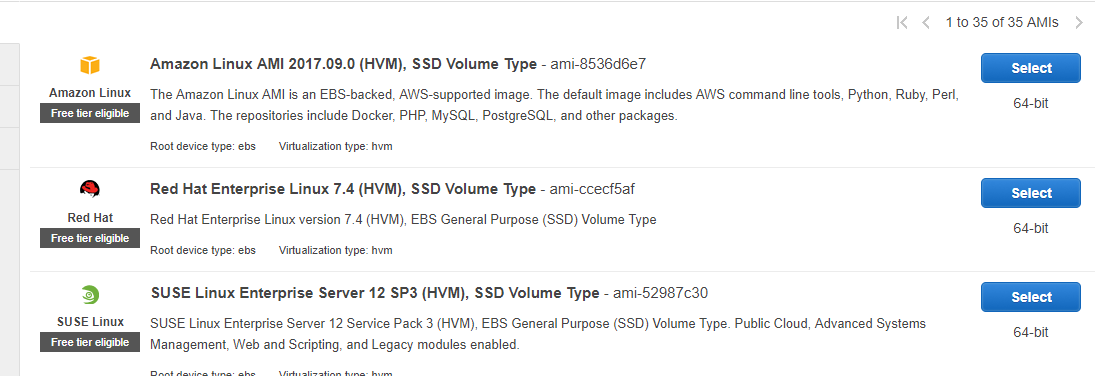
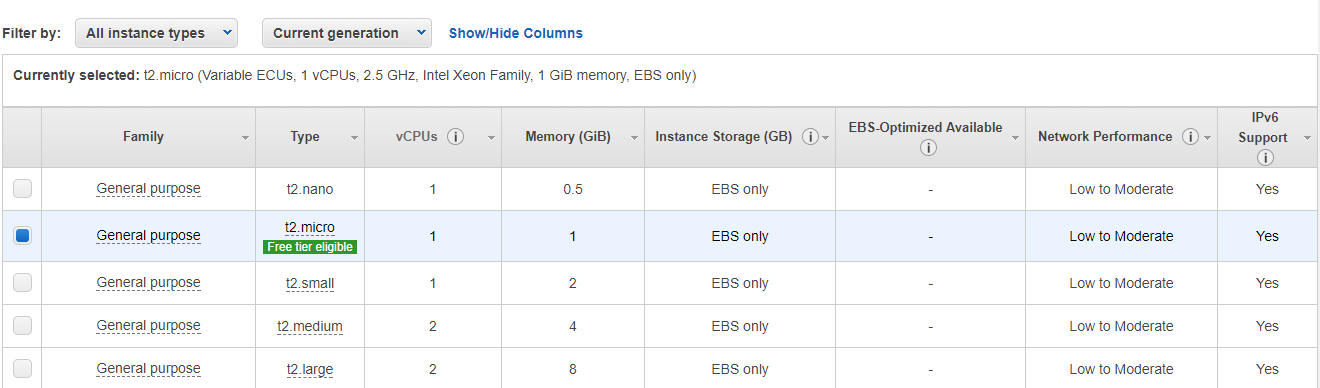


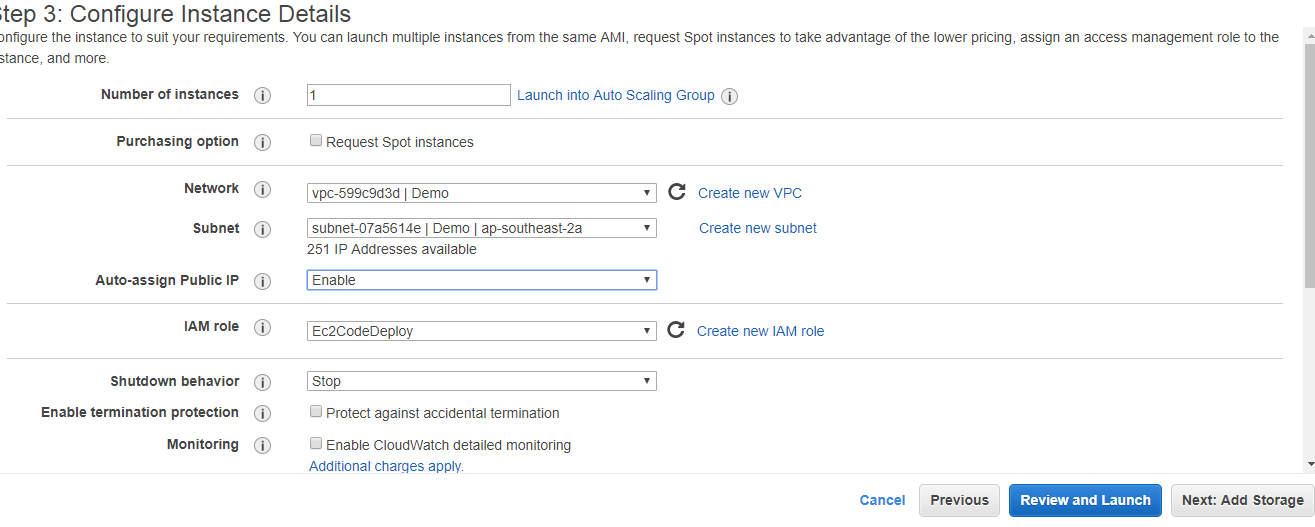
* Now you have created your VPC.

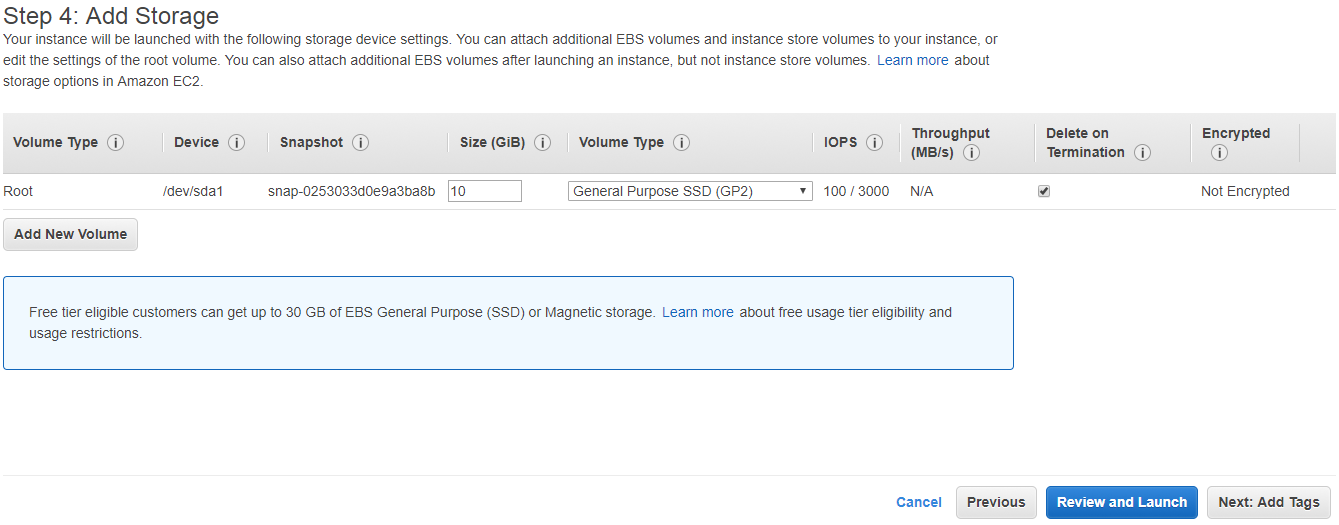
How to create an EC2 instance

* Click on EC2 dash board click on launch instance.

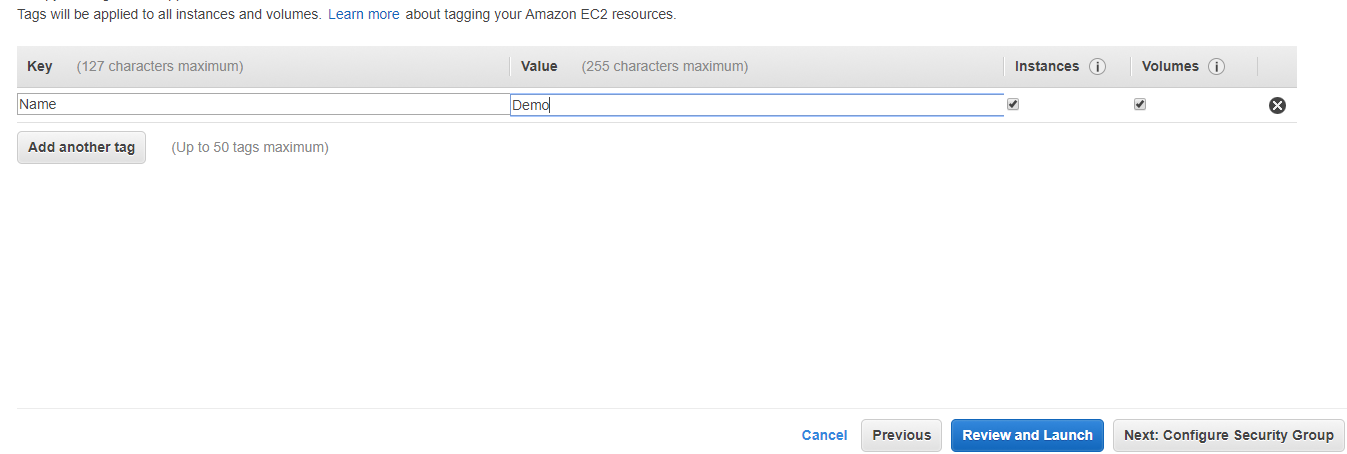


* Now select operating system, application server, required to launch your instance.
* Now choose an instance type as required and click on configure.
* And select the VPC network which was created in VPC as ‘Demo’ and subnet.
* Choose an IAM role as EC2codedeploy and click on add storage.

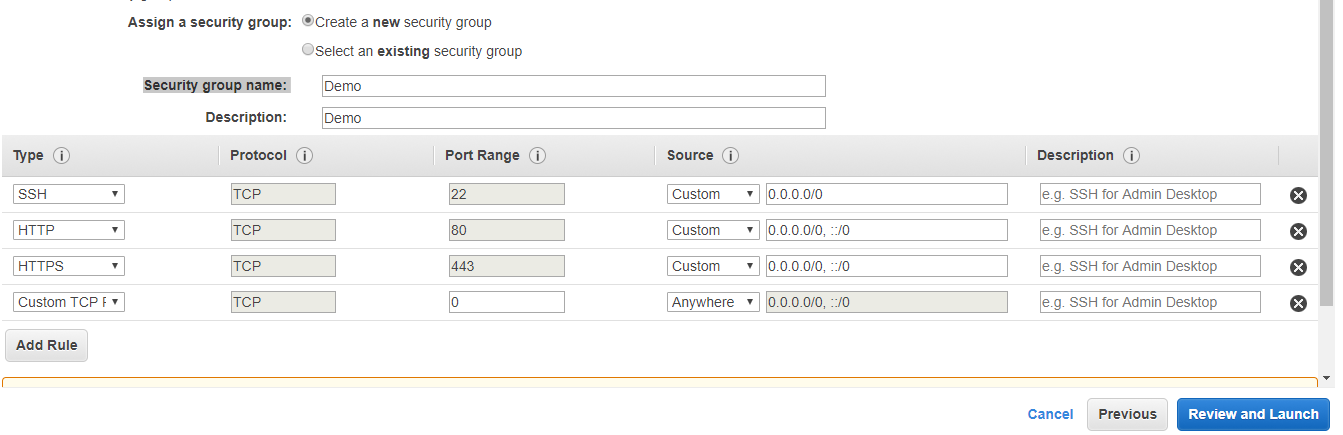




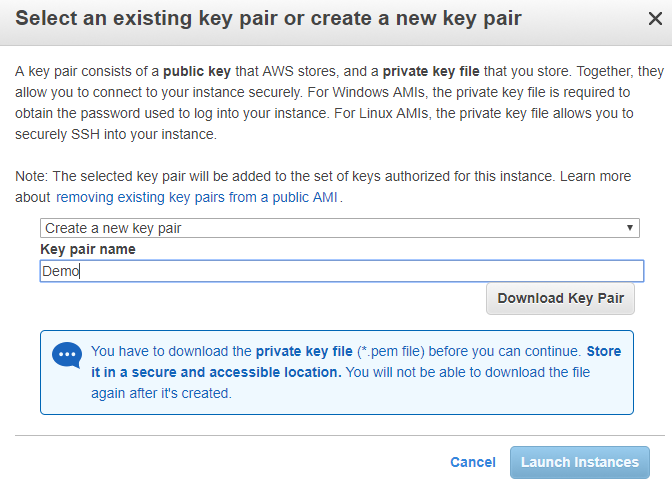
* Add tags and give the name ‘Demo’ and click on configure security group.



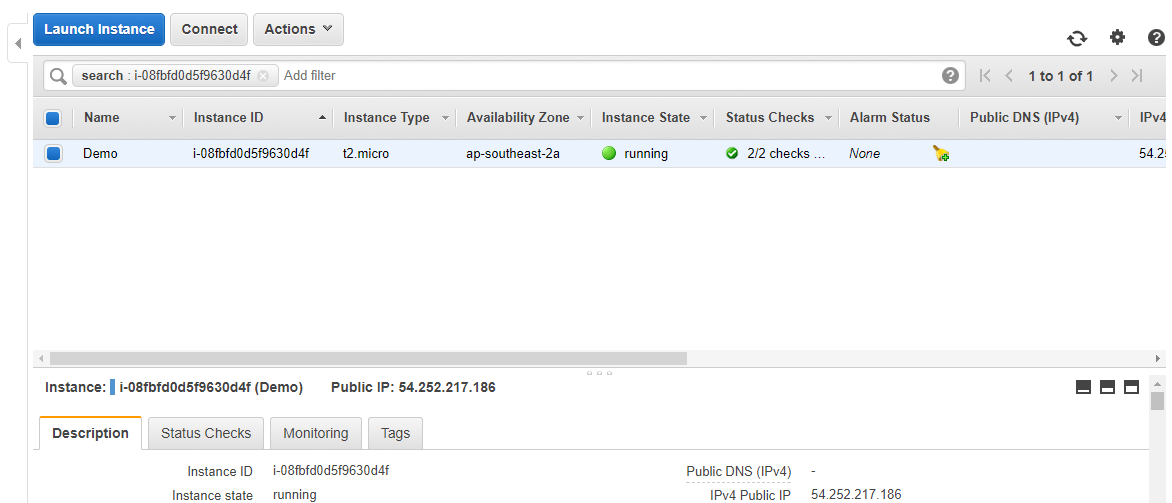
* Assign a security group as required.
* And give the security name as ‘Demo’ and launch.



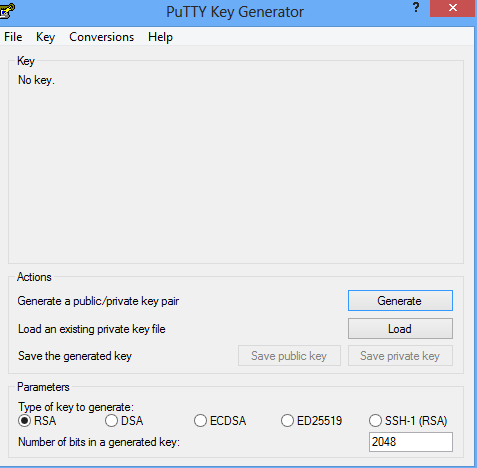
* Select an key pair as you require and give key name ‘Demo’ and download the key pair.



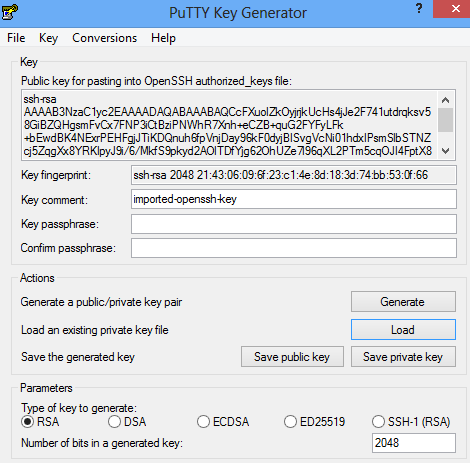
* Click on launch instance and now you can see your instance.

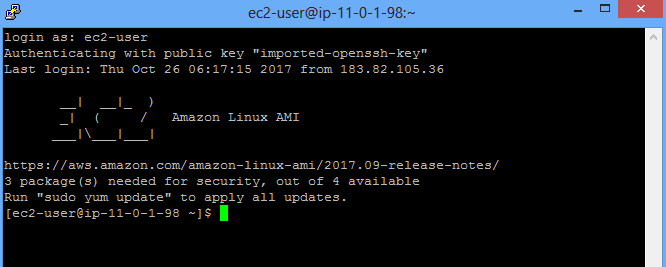


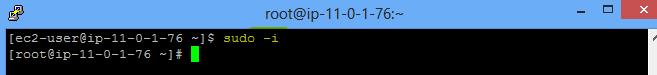
* The key pair is down loaded as Demo.pem .
* Next we have to convert that pem to ppk using puttygen.



* Load the pem into the putty key generator and save as private key



* And next choose putty and give the public IP from your instance.
* And click on SSH and Auth and browse the ppk file where you have saved click on open.
* Now you can see your login.
* After login to server we need to sudo privileges for configurations.



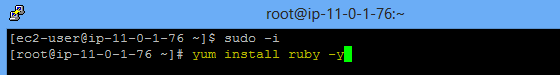
* Now we are in root user.
* Now we can configure applications what ever we need. For Code Pipeline and code Deploy we need to install code deploy agent.

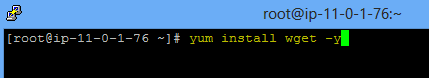
Steps to install Code Deploy agent:

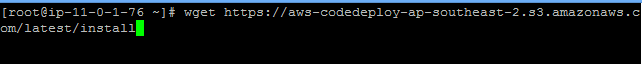
* Based on operating system the commands are different. For example if we take redhat linux the repositories are yum, for Ubuntu apt etc,
* Now we are configuring code deploy in redhat linux the following steps to be followed

Prerequisites:

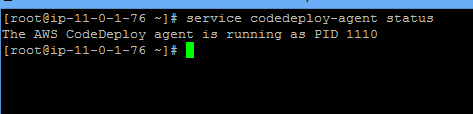
* sudo yum install ruby
* sudo yum install wget





* Now we need to download code deploy packages.

Note: Here the packages are region specific. So, we need to pass region name(aws-codedeploy-region name)

* After that you need to enter the following commands
  + - chmod +x ./install
    - sudo ./install auto
* To Check Code Deploy Agent Status
* Now we need to configure Java to run tomcat application.
* For Tomcat



* Now we need to extract the tar file.

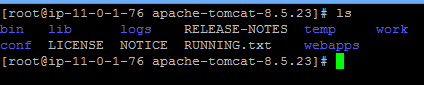


* Now you can observe apache-tomcat-8.5.23 folder.

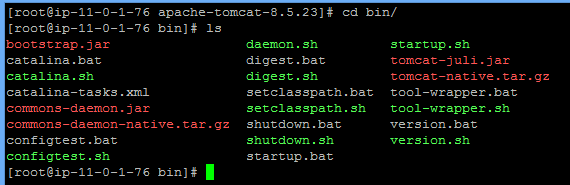
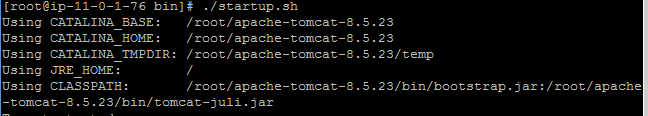


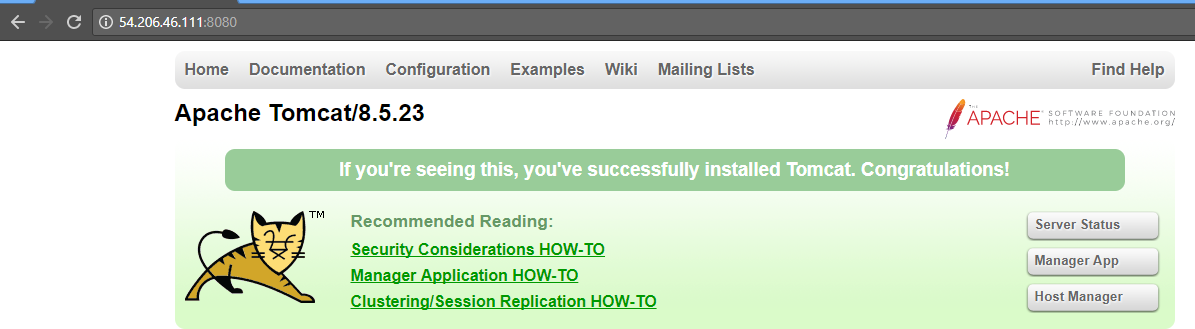
* Navigate to apache folder

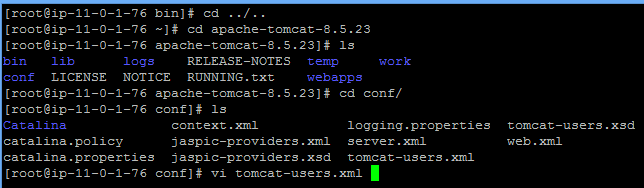




* To start Tomcat you need navigate bin directory and run the below command.

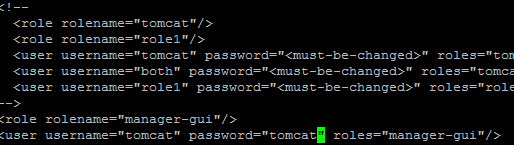


* Now you can check tomcat web page in your browser. Make sure the 8080 port should be enable.

* To access admin level tomcat we need to configure some files. The following screen shots describes where to change settings.
* Just you need to add the following lines under following screen

<role rolename="manager-gui"/>

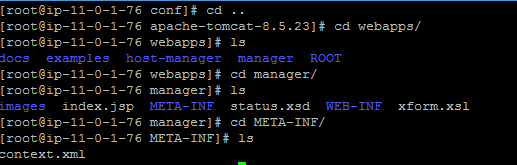
<user username="tomcat" password="s3cret" roles="manager-gui"/>



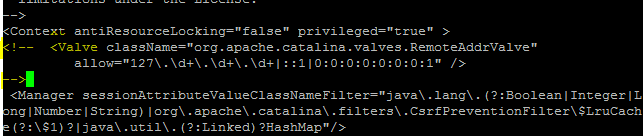
* After adding the two lines you need to save the file, before that you must press escape button. After that the following command.

Cmd: :wq

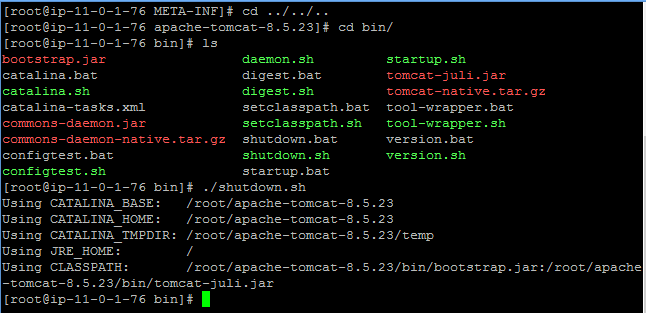
* After that you can comment one of the line to view admin access tomcat. You can follow the below screen shot.

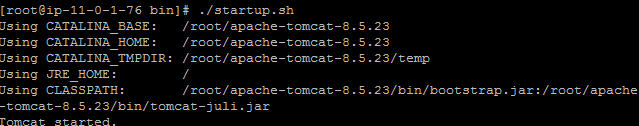
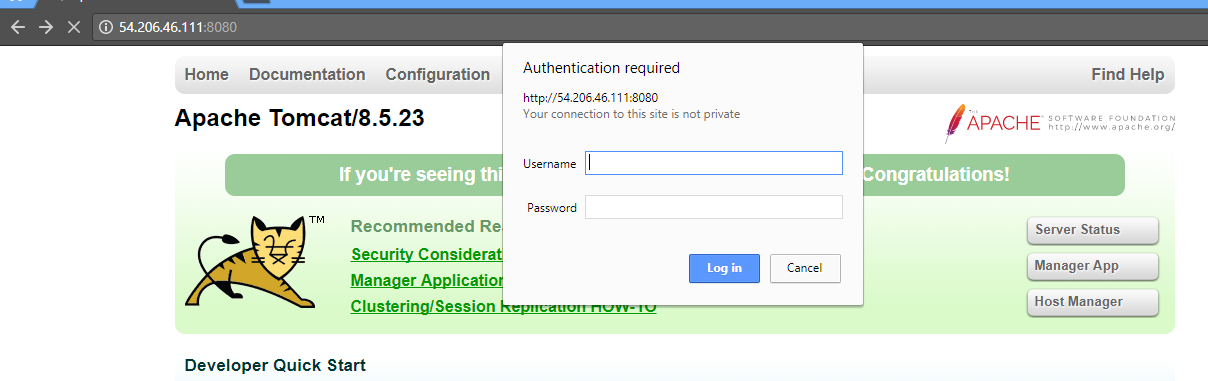


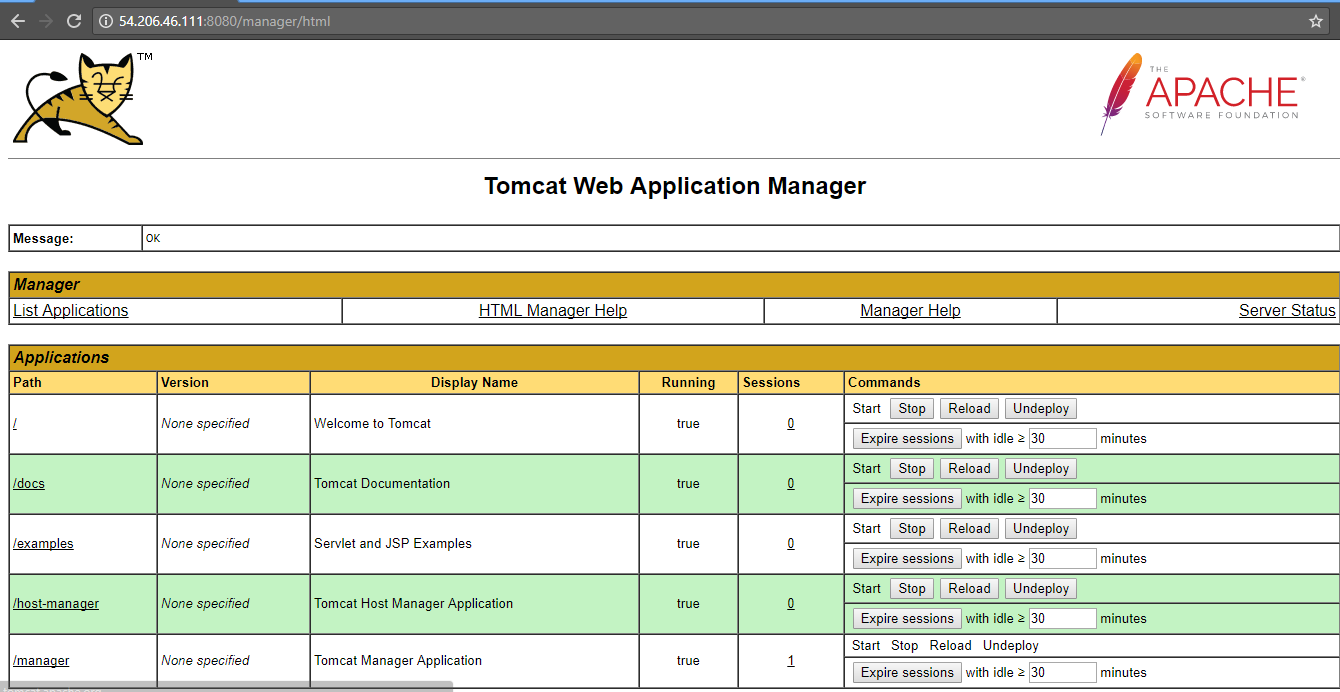




* Now we need to restart Tomcat.

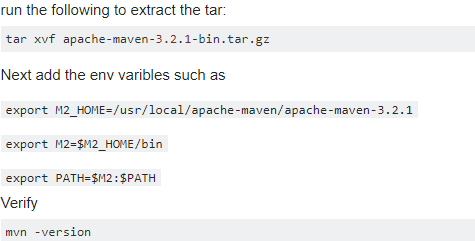






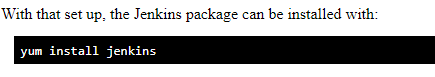
**Maven Installation:-**



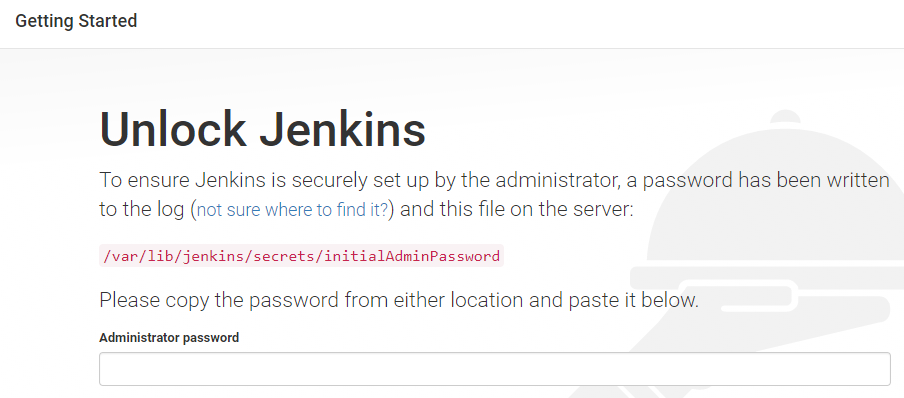


**For Jenkins:-**

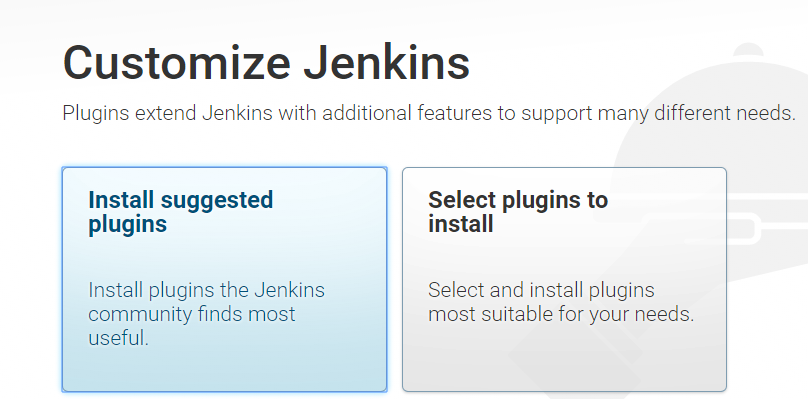


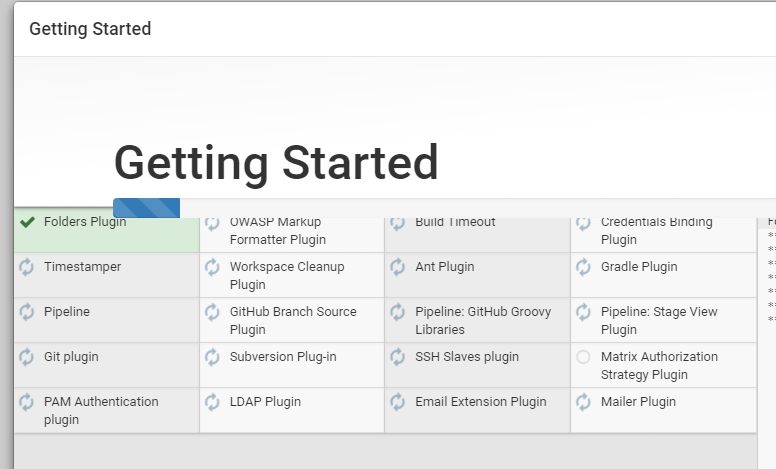


* Enter Cmd :- service Jenkins start

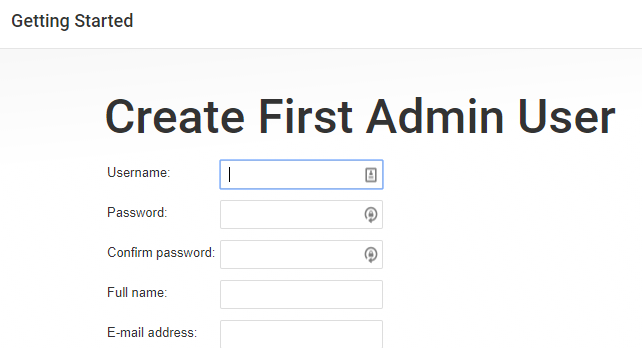


* Now enter the password shown on terminal.
* Now you can see the plugins and select the suggested plugins

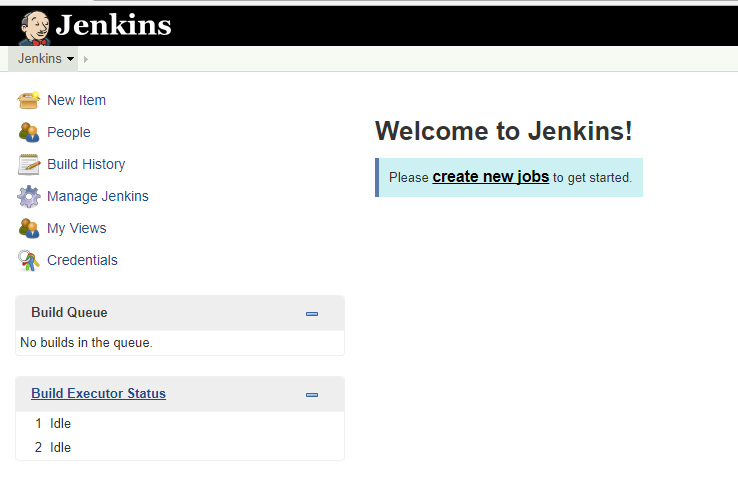




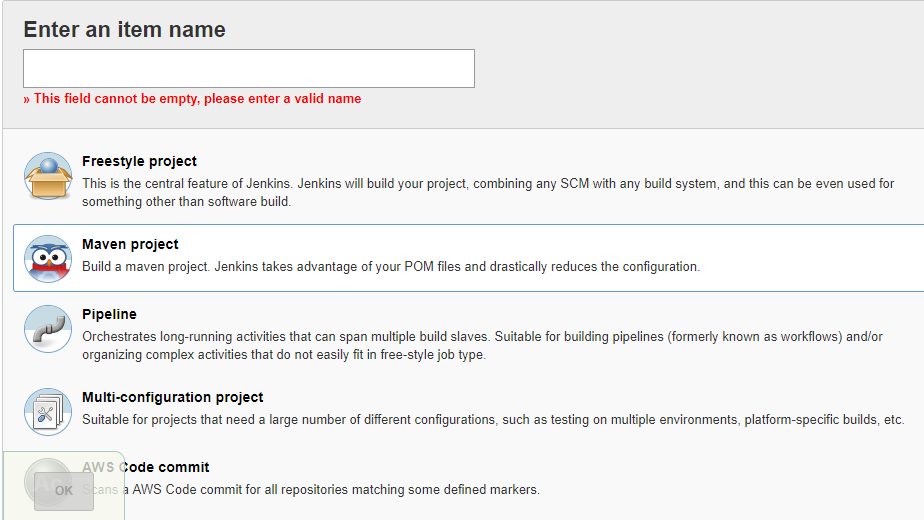
* Now create your own user name and password



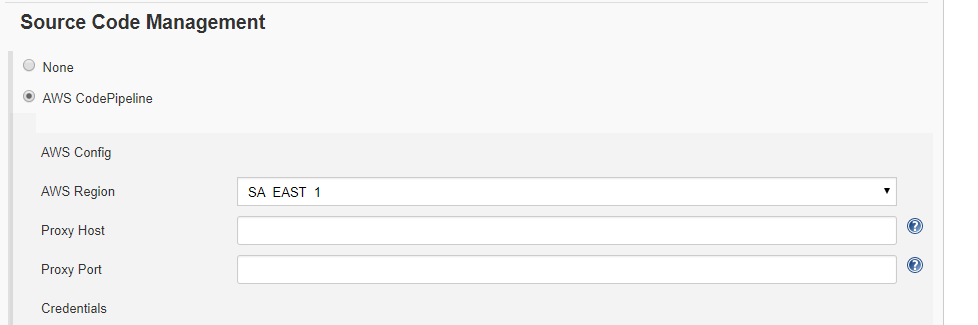
* Now you can see the Jenkins dash board.



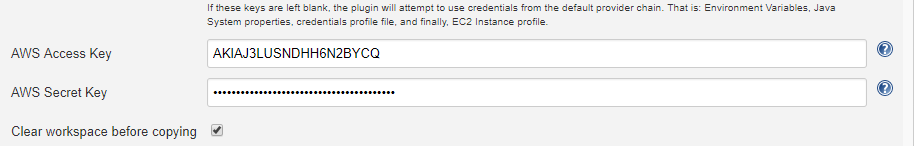
* Go to manage Jenkins and then manage plugins.
* Next click on available plugins and select codecommit, codebuild, codedeplot, codepipeline, maven plugins
* And install without restart Jenkins.
* Now go jenkis , select “New item”.
* Enter project name and select ‘maven project’.
* And click ok.

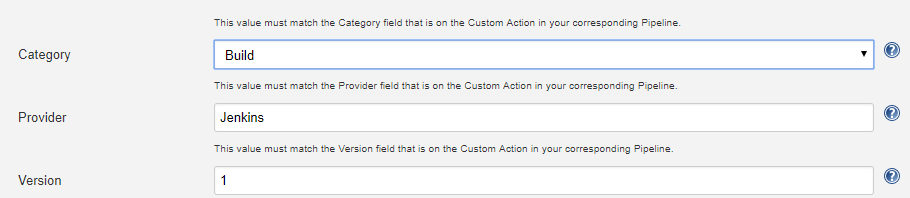


* Click on the source code management and select codepipeline and specified region.

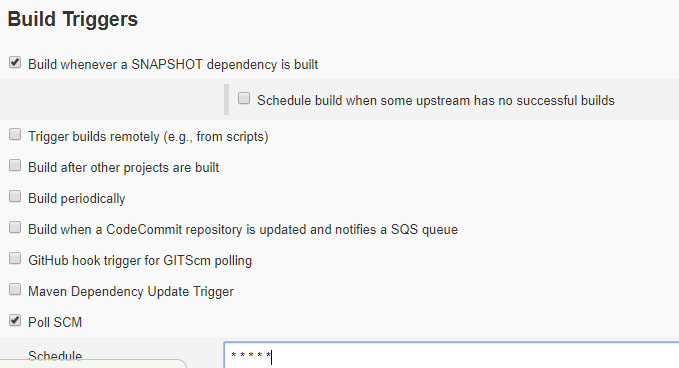


* In order to integrate with AWS CodePipeline, you must authorize access to the pipeline

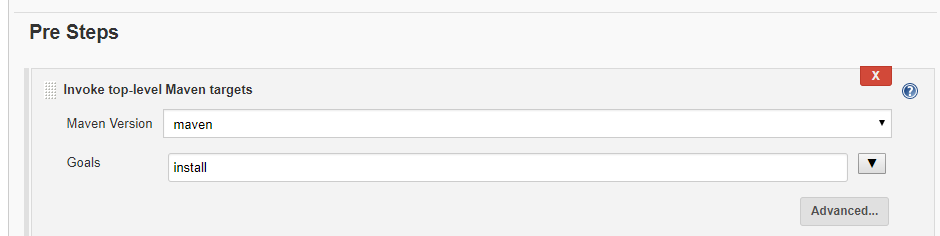


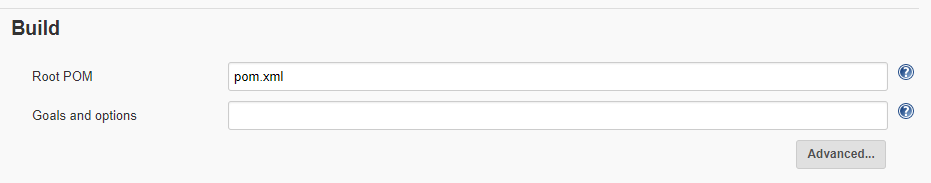


* Click on build trigger and select poll scm .

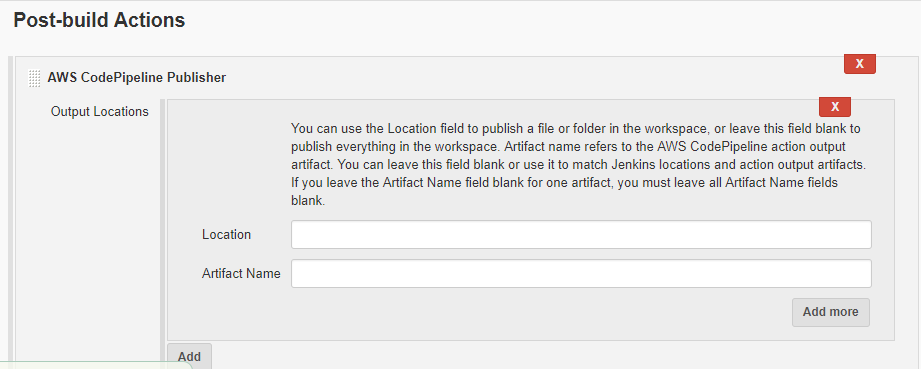


* Now click add pre build steps

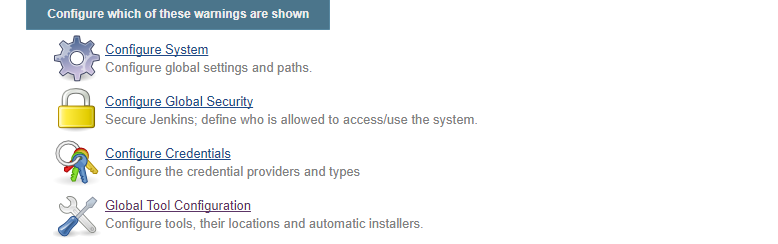




* Now select on post build step and click on add and leave it blank.
* And click on save.



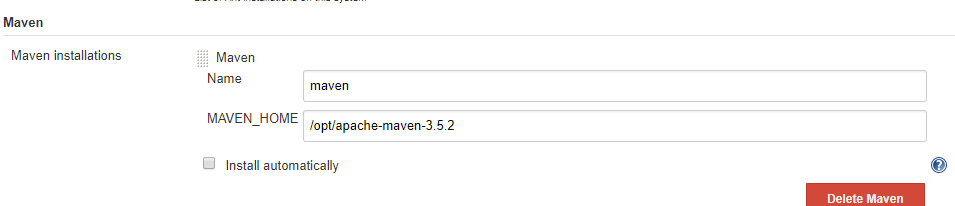
* Next go to the jenkins dash board and select the manage jenkins and select global tools configuration.



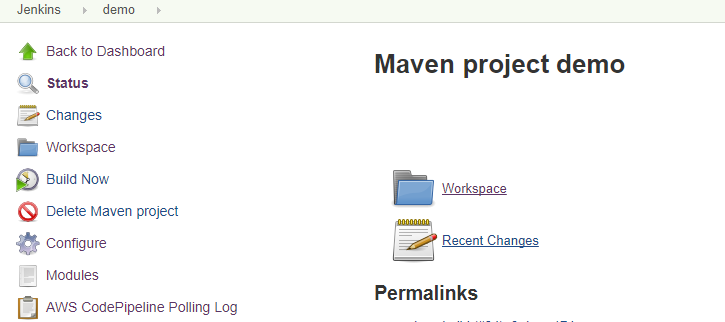
* And select jdk and enter jdk 8 in the blank table.



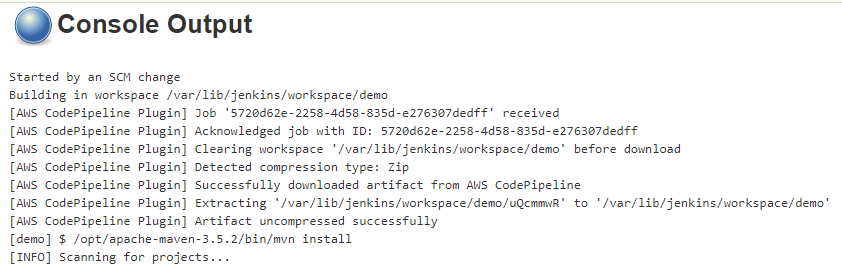
* And select the maven and enter the path shown in the maven installations.

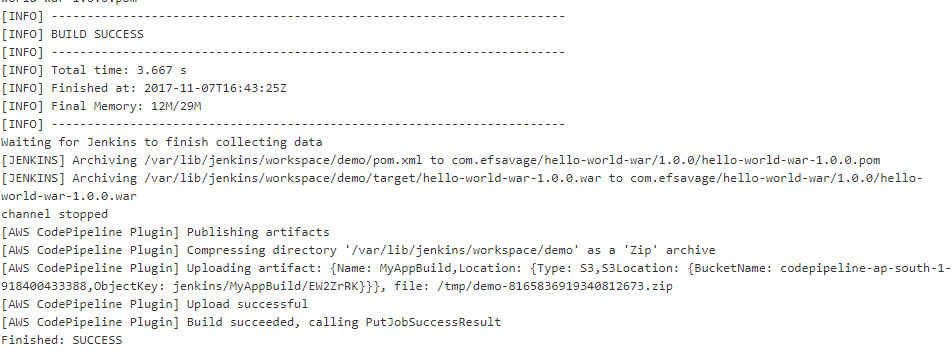


* And click on save and go to jenkins dash board.

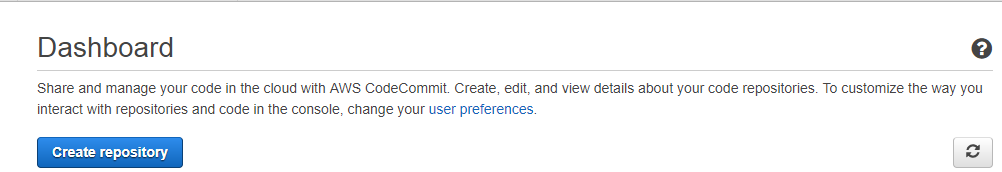


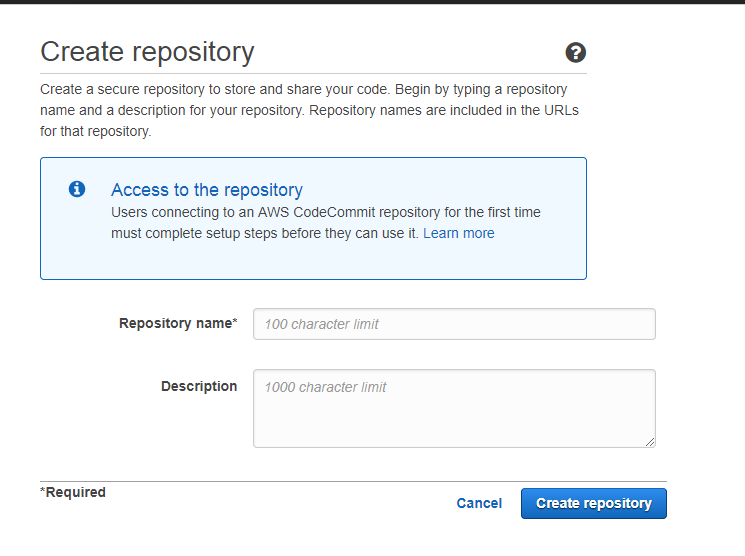
* Click on ‘Build Now’ shown in the jenkins dash board.
* Then you can see build success in the console output.





**Code Commit:-**

* First login to aws console and click code commit service.
* Next create a Repository.



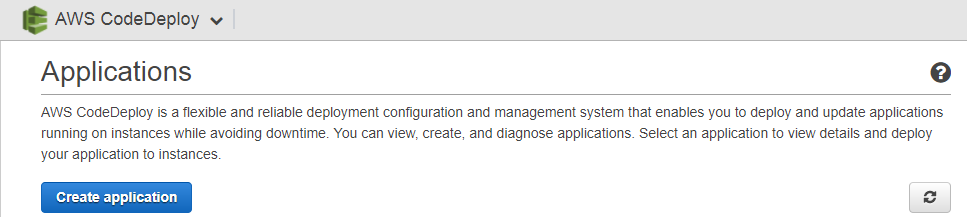
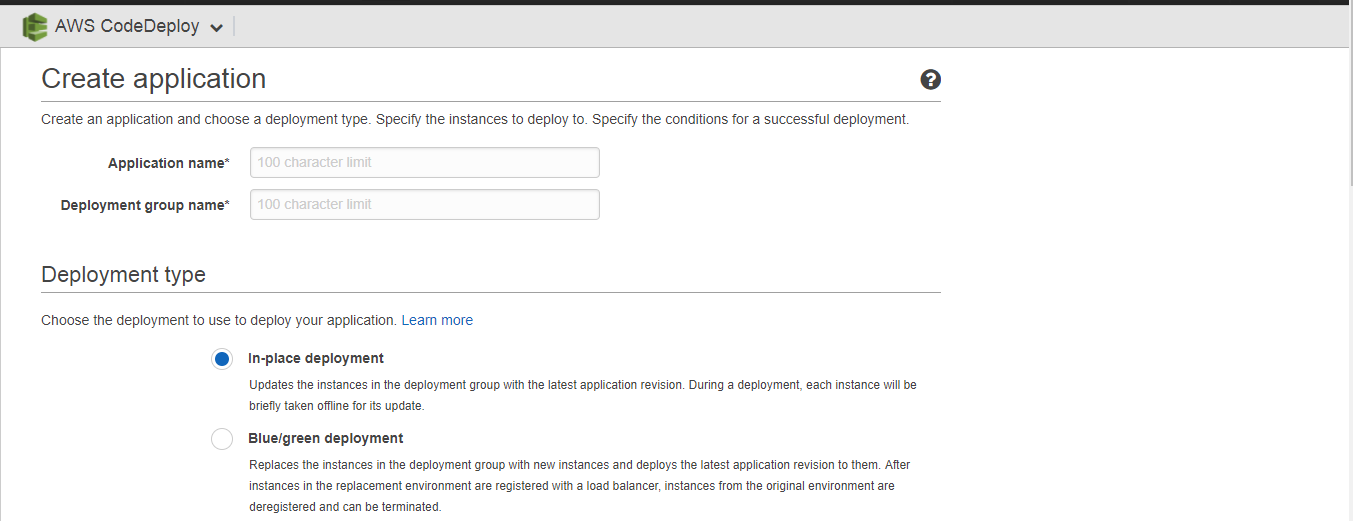
* Give the Repository name as u wish and click create repository.
* For example I given repository name as helloworld.
* After creating the repository click connect.
* It show ssh and https, click ssh and copy the URL and paste into your server by using following command.

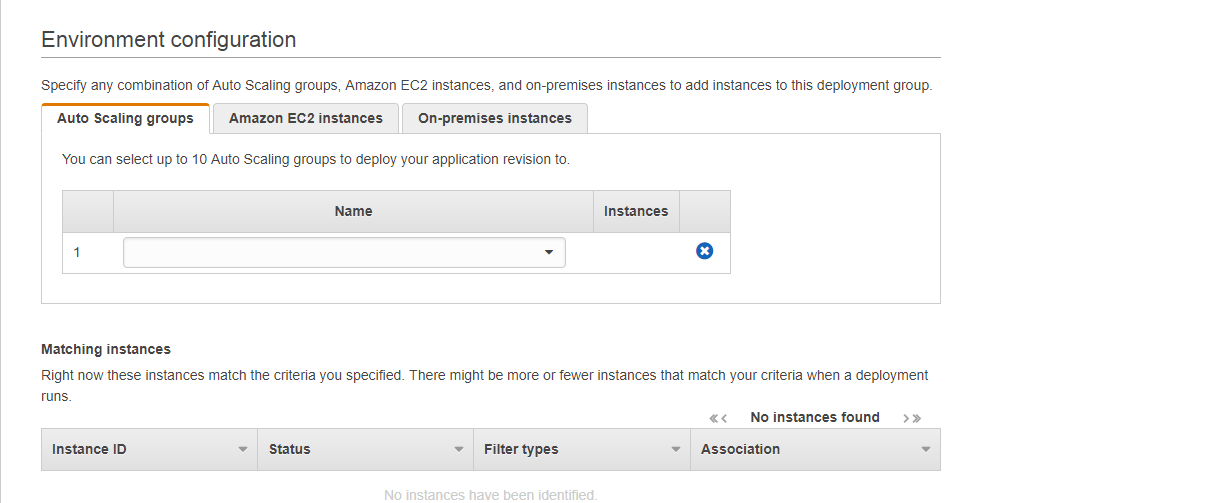
Cmd : git clone git clone ssh://git-codecommit.eu-west-1.amazonaws.com/v1/repos/ helloworld

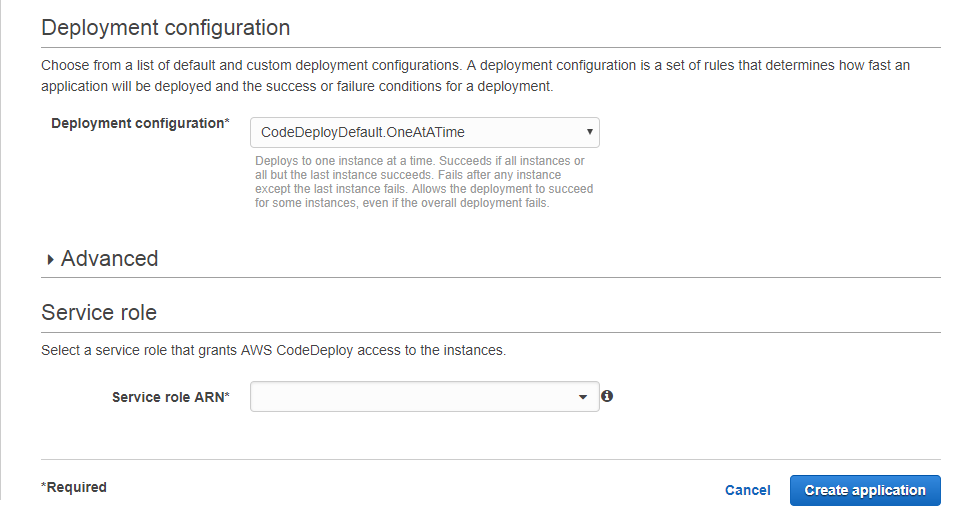
* Now you can see your repository in your server.
* Navigate into your repository. Now you can push your code into your repository by using the git commands.

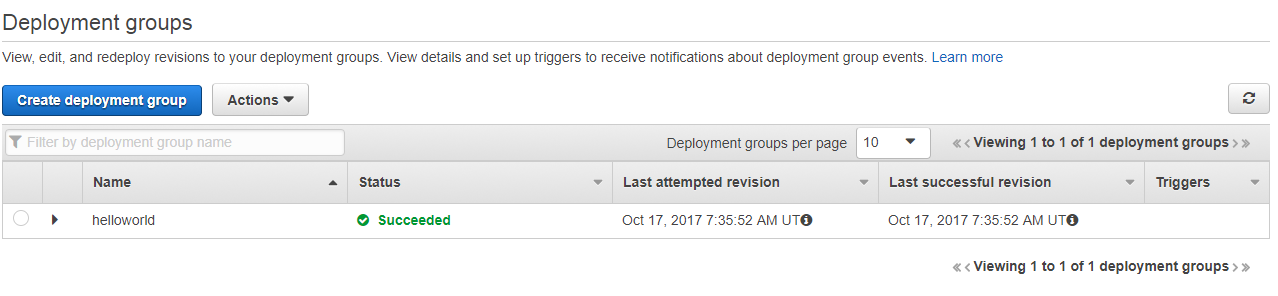


**Code Deploy:-**

* Click on Code deploy in your aws services.
* Click Create application and give required info.

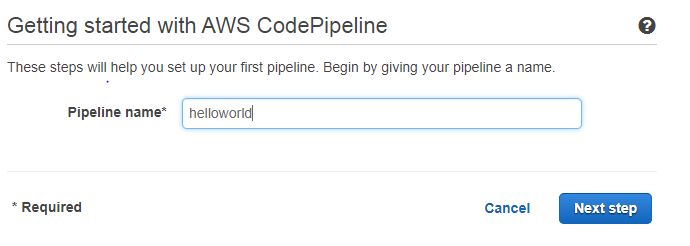


* If you have auto scaling you can mention the auto scaling group name , Ec2 instances. For that only the deployment will be done those instances.
* Give Service role Arn and finally click create application.
* For Example my application name is helloworld and finally you see the following screen.

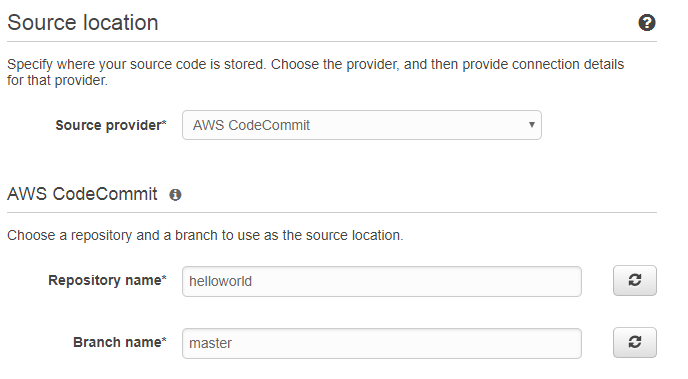


**Code Pipeline:-**

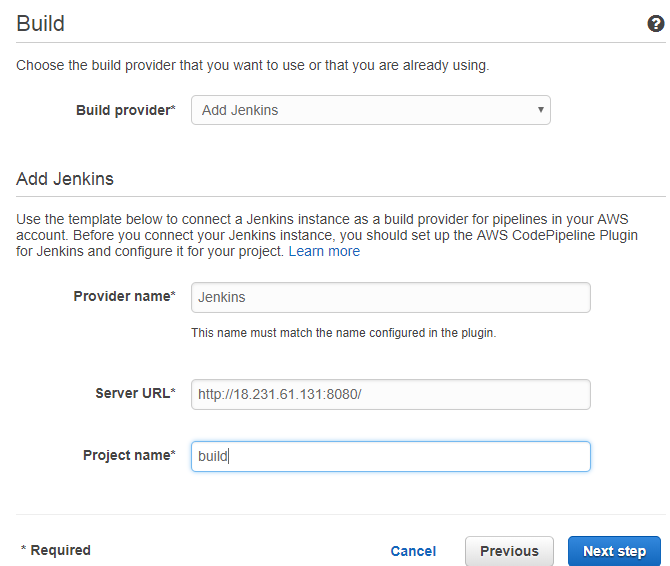
* By using code pipeline we can automate the code commit and code deploy. For that we need to create a pipeline.



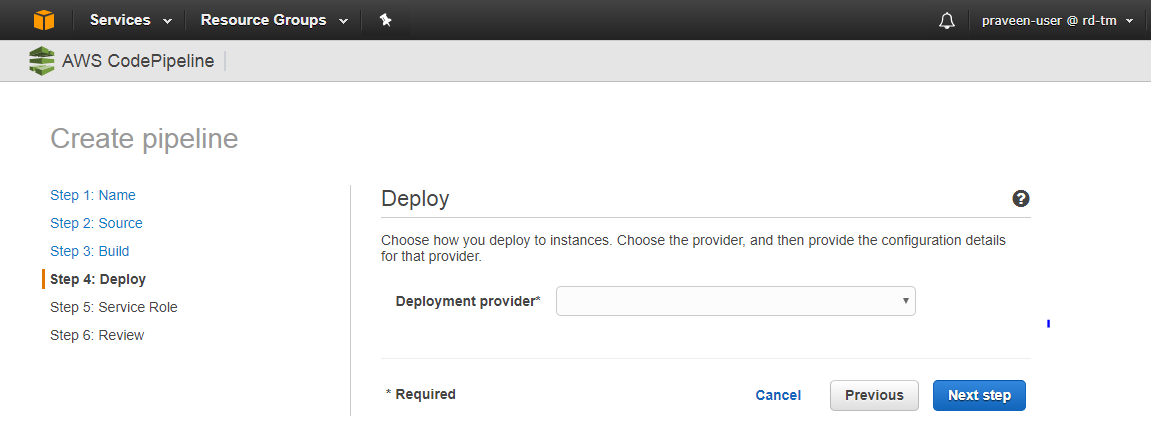
* Give the Pipe line name and click next step.

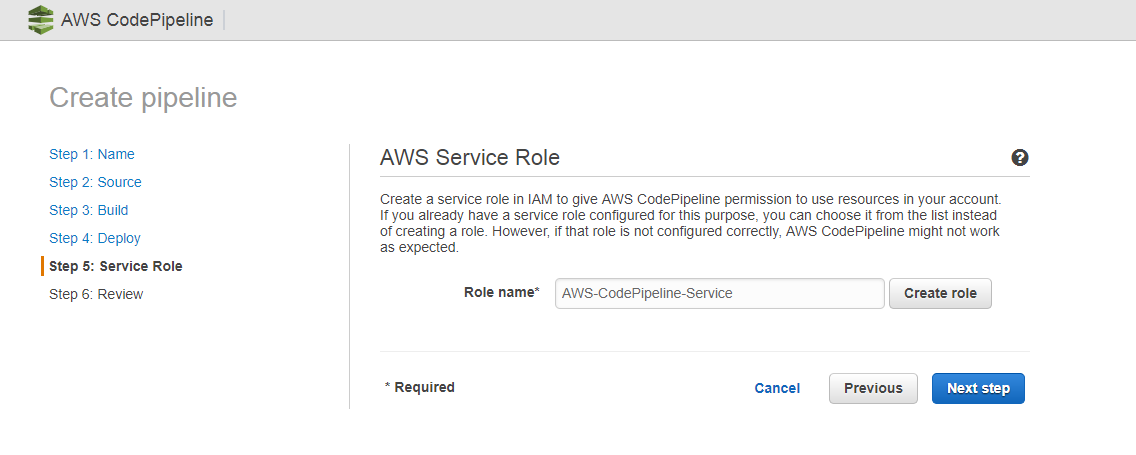


* Select aws code commit as a source provider and click next.
* Choose the Repository name and Branch name as “**Master**”.
* Choose the build provider that you want to use.



* Configure your project and give the project name



* For Deployment provider select code deploy from drop down list.
* Finally click review.

