What is Cloud?

In the past, people would run applications or programs from software downloaded on a physical computer or server in their building.

Traditional business applications have always been very complicated and expensive. The amount and variety of hardware and software required to run them are daunting. You need a whole team of experts to install, configure, test, run, secure, and update them.

When you multiply this effort across dozens or hundreds of apps, it's easy to see why the biggest companies with the best IT departments aren't getting the apps they need. Small and midsize businesses don't stand a chance.

In the simplest terms, **cloud** computing means storing and accessing data and programs over the Internet instead of your computer's hard drive. The **cloud** is just a metaphor for the Internet.

Cloud Computing is the use of hardware and software to deliver a service over a network (typically the Internet). Cloud computing is a kind of outsourcing of computer programs. Using cloud computing, users are able to access software and applications from wherever they are; the computer programs are being hosted by an outside party and reside in the cloud. This means that users do not have to worry about things such as storage and power, they can simply enjoy the end result.

What are the types of cloud Computing?

Based on a service the cloud model is offering, we are speaking of either:

- IaaS (Infrastructure-as-a-Service)
- PaaS (Platform-as-a-Service)
- SaaS (Software-as-a-Service)
- Storage, Database, Information, Process, Application, Integration, Security, Management, Testing-as-a-service.

IAAS (Infrastructure as a Service):

IAAS (Infrastructure as a Service) is the base layer and typically provide access to networking, virtual machines, storage. The client has the highest level of flexibility and management over resources compared to PAAS and SAAS.

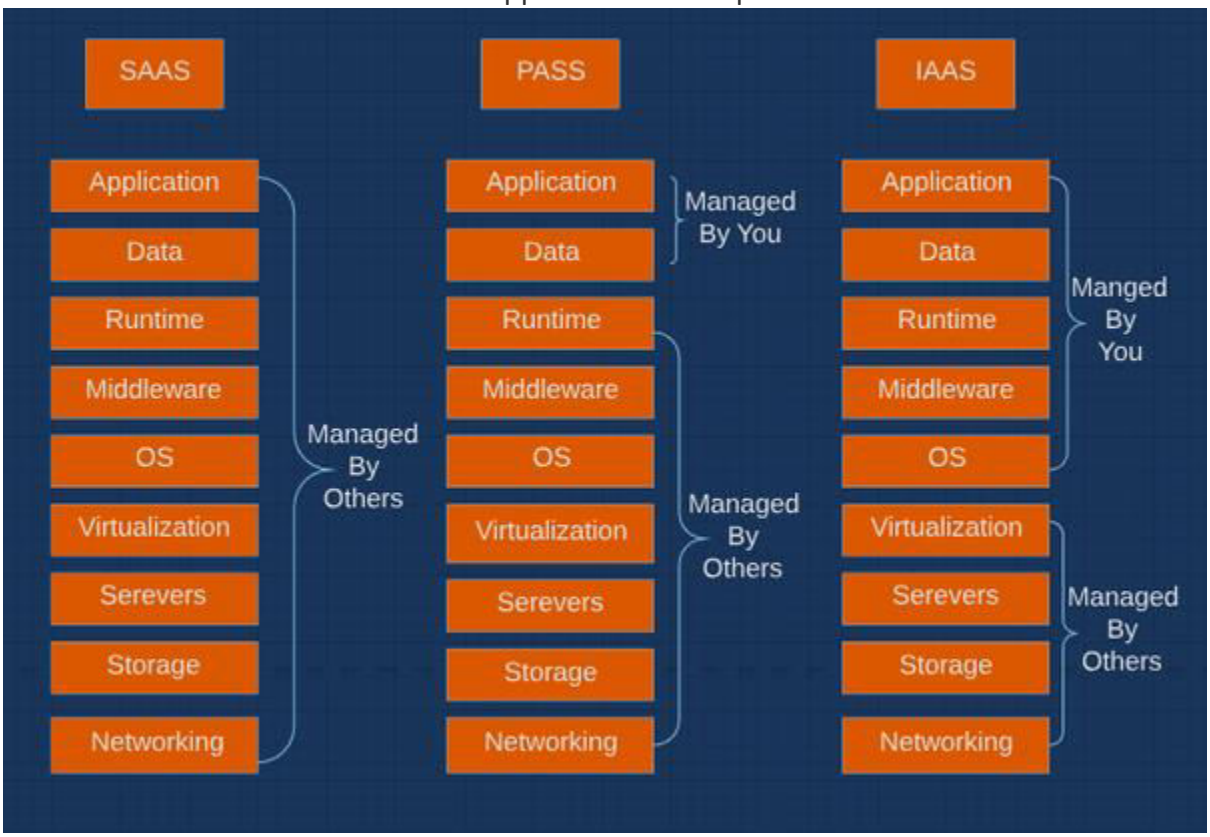
PAAS (Platform as a Service):

PAAS (Platform as a Service) provides flexibility to users that they no need to manage the underlying infrastructure like operating system, runtime, software maintenance, patching etc. PAAS is suitable for developers because of provider will manage OS, servers, storage, Networking. So they can only concentrate on the development of the application.

SAAS (Software as a Service):

SAAS (Software as a Service) is most popular and easy to use. Because service provider takes care of all the responsibilities like networking, storage, servers, firewalls, runtime,

data, Application. Whenever client chooses cloud type as SAAS, he/she don't have any control over the resources under the application and responsibilities.



Here are some other benefits of cloud computing.

Adaptable

Cloud computing allows for adaptable programs and applications that are customizable, while allowing owners control over the core code.

Multitenant

Cloud software provides the opportunity to provide personalized applications and portals to a number of customers or tenants.

Reliable

Because it is hosted by a third party, businesses and other users have greater assurance of reliability, and when there are problems, easy access to customer support.

Scalable

With the Internet of Things, it is essential that software functions across every device and integrates with other applications. Cloud applications can provide this.

Secure

Cloud computing can also guarantee a more secure environment, thanks to increased resources for security and centralization of data.

Advantages of using Cloud Storage. Cloud **Storage** is a service where data is remotely maintained, managed, and backed up. The service allows the users to store files online, so that they can access them from any location via the Internet.

Amazon Web Services (**AWS**) is a secure **cloud** services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow.

Introduction to AWS

What is AWS Technology?

Amazon Web Services (**AWS**) is a comprehensive, evolving cloud computing platform provided by Amazon. It provides a mix of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings. As of April 2018 there are **142 services** listed on the AWS platform that are categorized in 19 different types. As of 11 Apr, 2018, there are total **102** distinct services available. There are some more services added in IoT and **one service** for Game Development as well.

How AWS Works?

Cloud Products & Services. Amazon Web Services (**AWS**) offers a broad set of global compute, storage, database, analytics, application, and deployment services that help organizations move faster, lower IT costs, and scale applications.

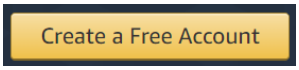
The 7 Most Popular AWS APIs

- Amazon EC2 API. Using the Amazon console is relatively easy.
- Amazon S3 API. Amazon Simple Storage Service (S3) gives you an infinite amount of storage.
- Amazon RDS API.
- Dynamo DB API.
- Amazon KMS API.
- Amazon SQS API.
- Amazon SNS API.

Having said that, EC2 is **IaaS** and is probably the only **AWS** offering that falls into the **IaaS** category. Using EC2, **AWS** users can provision Compute, networking and storage just by calling various APIs. ... **AWS** Elastic BeanStalk is **PaaS**. This diagram shows differences between **Iaas**, **PaaS** and **SaaS**.

Before going to use AWS service first you need to create AWS Account which came up with first one year of free Tier Access.

To getting started with AWS follow Below steps:

1. you need to go to https://aws.amazon.com/ click on a 

2. Enter details like email address, password and account name then click on **continue.**
3. Then choose account type as personnel, enter all the details prompted and check the checkbox to agree terms and conditions then click on **create Account and continue.**
4. Then enter debit or credit card details and click on secure submit.
5. Select support plan as Basic plan (Free).

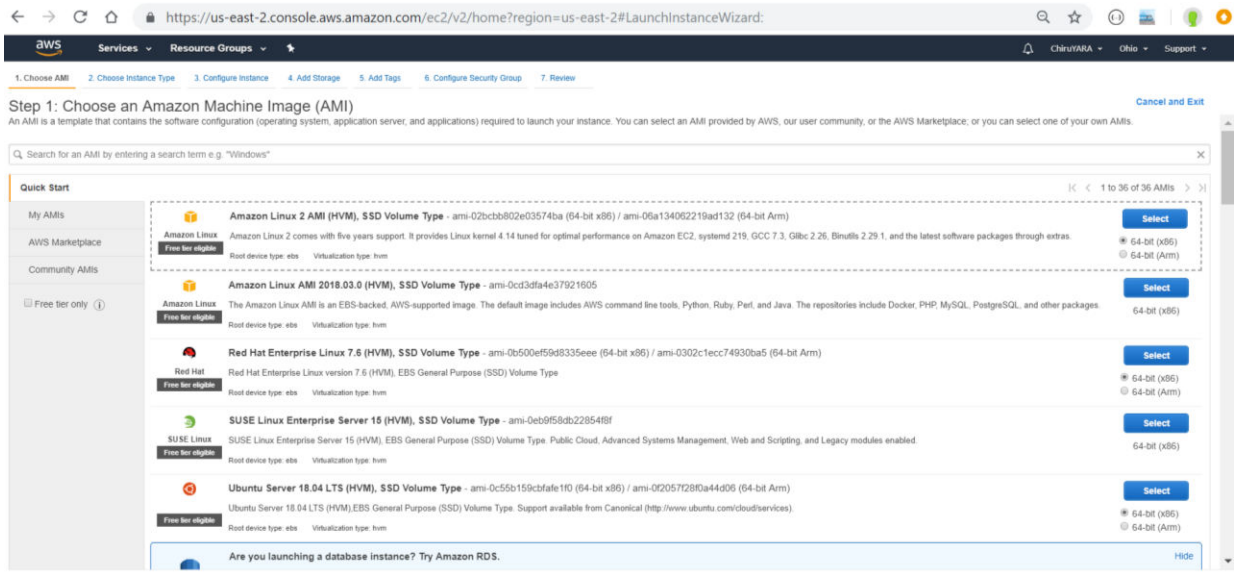
EC2

Amazon Elastic Compute **Cloud** (EC2) provides virtual servers -- called instances -- for compute capacity. ... The **Amazon** EC2 Container Service and EC2 Container Registry enable customers to **work** with Docker containers and images on the **AWS**platform.

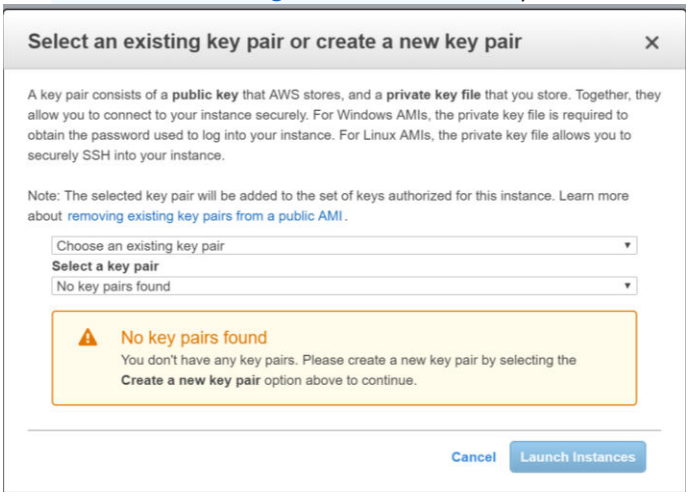
What is DevOps engineer role?
DevOps Engineer works with developers and the IT staff to oversee the code releases. They are either developers who get interested in deployment and network operations or sysadmins who have a passion for scripting and coding and move into the development side where they can improve the planning of test and deployment.

Creating EC2 Instance:

1. From the services select EC2 then click on launch new instance.
2. Choose your preferred configuration and select the window will looks like below image



3. Choose instance type as t2.micro (free tier eligible) then click on configure review and launch.
4. Then click on launch.
5. For the first use of ec2 instance you need to create key pair for that this wizard will be looks like below image.
   a. From the dropdown list choose create a new key pair
   b. Enter any name as key pair name.
   c. Then click on download key pair (You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.)



6. Then click on launch instances.
7. It will take time to create your instance.

Connecting to EC2 instance using putty.

1. To connect to ec2 instance first time you need to install Putty in your local machine.
2. Then you need to create .pem file as .ppk file using puttygen.

a. To generate .ppk file first open puttygen
b. Then click on load button and browse for the .pem file which you downloaded when creating ec2 instance
c. And then click on **save Private Key.**

3. Go to AWS service → EC2 → Running Instances.
4. Select the instance and in the description tab you will have public IP address/ public DNS copy that and open putty.
5. In the hostname text area enter the public IP / public DNS of that ec2 instance.
6. Then from the left side tab in putty connections→SSH → Auth
7. Then browse .ppk file which is you generated and click on open
8. Select yes from the alert box and in the putty it will prompt for **log in as** enter user name as **ec2-user**.
9. You can configure your virtual machine according to your requirements.

Setting up DevOps tools in your virtual machine.

How to install and Configure Jenkins on Linux7

Step: 1 Download java and Jenkins

Jenkins package is not available in the default RHEL repositories. So we need to add jenkins repository using the beneath commands.

```
[ec2-user@ip-172-31-45-142 opt]# sudo yum install java-1.8.0-openjdk-devel

[ec2-user@ip-172-31-45-142 opt]# curl --silent --location http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo | sudo tee /etc/yum.repos.d/jenkins.repo

[ec2-user@ip-172-31-45-142 opt]# sudo rpm --import https://jenkins-ci.org/redhat/jenkins-ci.org.key
```

Step:2 Install Jenkins and Java

Run the below yum command to install Jenkins and java.

```
[ec2-user@ip-172-31-45-142 opt]# sudo yum install jenkins
```

Step:3 Start and Enable Jenkins Service

Run the following systemctl commands to start and enable the jenkins service

```
[ec2-user@ip-172-31-45-142 opt]# sudo service Jenkins start

[ec2-user@ip-172-31-45-142 opt]# systemctl enable jenkins
```

You should see something similar to this.

```
[ec2-user@ip-172-31-45-142 opt]# systemctl enable Jenkins

● jenkins.service - LSB: Jenkins Automation Server

   Loaded: loaded (/etc/rc.d/init.d/jenkins; bad; vendor preset: disabled)  Active: active (running) since Sun 2019-
03-17 13:36:14 UTC; 23s ago

     Docs: man:systemd-sysv-generator(8)

  Process: 32754 ExecStart=/etc/rc.d/init.d/jenkins start (code=exited, status=0/SUCCESS)

   CGroup: /system.slice/jenkins.service

           └─305 /etc/alternatives/java -Dcom.sun.akuma.Daemon=daemonized -Djava.awt.headless=true -
DJENKINS_HOME=/var/lib/jenkins -jar /usr/lib/jenkins/j...

Mar 17 13:36:13 ip-172-31-45-142.us-east-2.compute.internal systemd[1]: Starting LSB: Jenkins Automation
Server...

Mar 17 13:36:13 ip-172-31-45-142.us-east-2.compute.internal runuser[32759]: pam_unix(runuser:session): session
opened for user jenkins by (uid=0)

Mar 17 13:36:14 ip-172-31-45-142.us-east-2.compute.internal jenkins[32754]: Starting Jenkins [  OK  ]

Mar 17 13:36:14 ip-172-31-45-142.us-east-2.compute.internal systemd[1]: Started LSB: Jenkins Automation Server.
```

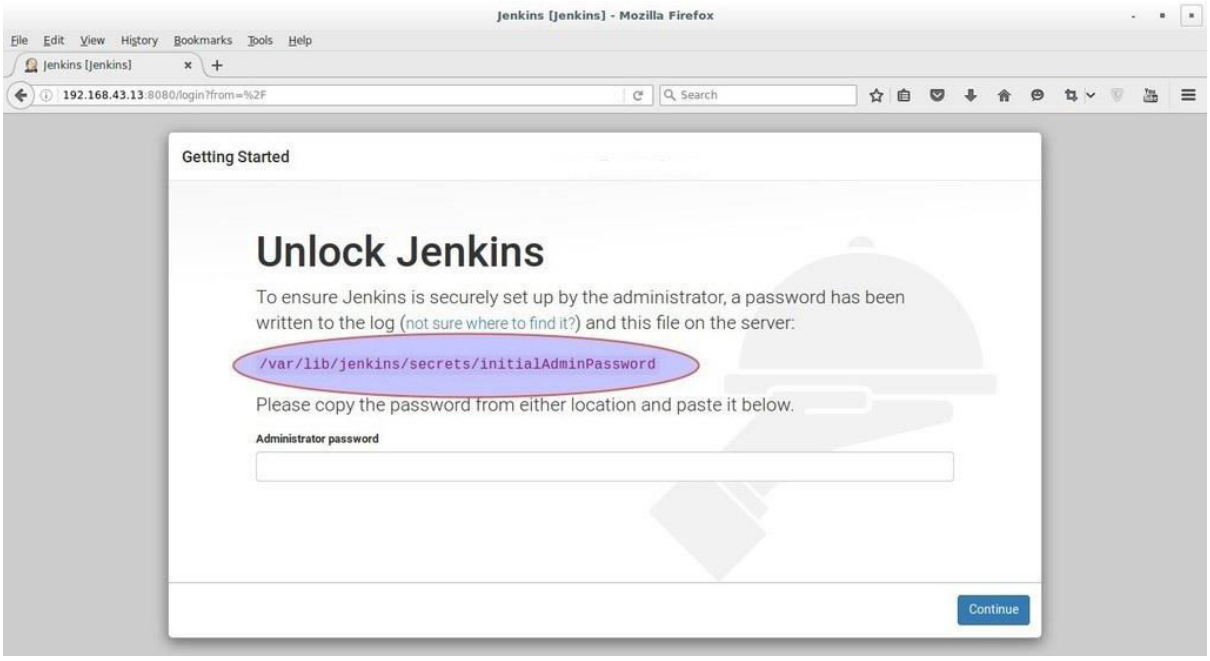
Step:4 Open the ports (80 and 8080) in OS firewall.

In case firewall is enabled on your Linux server then run the following commands to open jenkins related ports like 80 and 8080.

```
[ec2-user@ip-172-31-45-142 opt]# firewall-cmd --zone=public --add-port=8080/tcp –permanent success

[ec2-user@ip-172-31-45-142 opt]# firewall-cmd --zone=public --add-service=http –permanent success

[ec2-user@ip-172-31-45-142 opt]# firewall-cmd –reload

success
```

Step:5 Access the Jenkins Web portal

Access the URL : http://<Ip-Address-of-your-Server>:8080

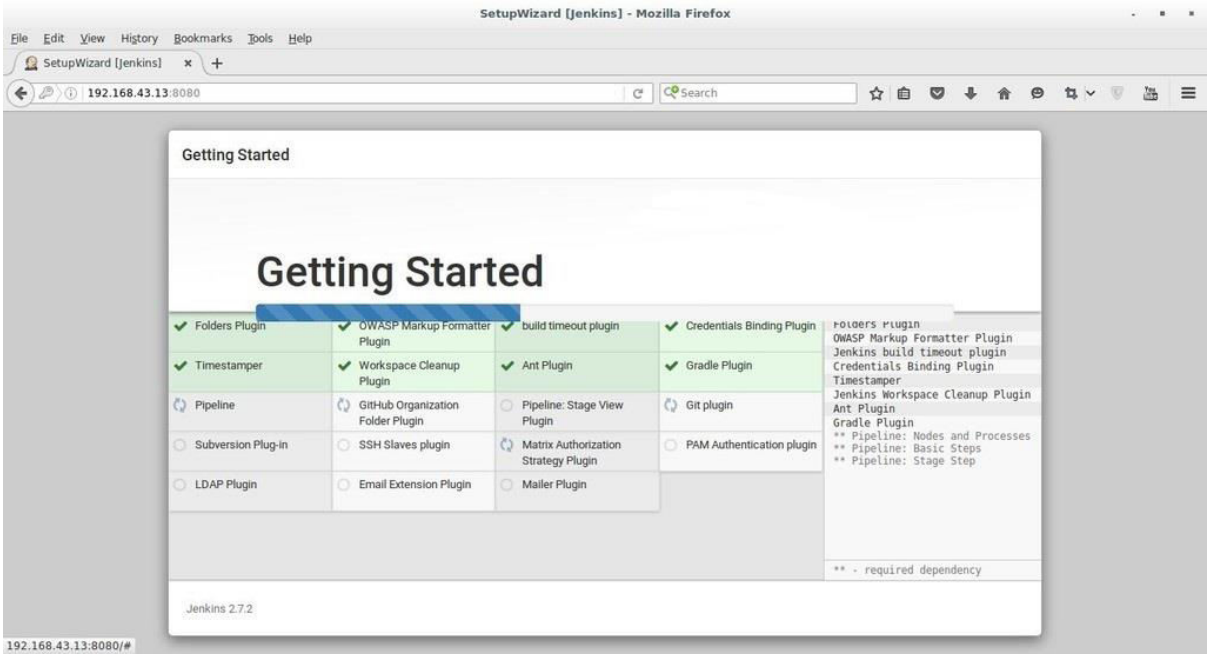


Admin password is created and stored in the log file "**/var/log/jenkins/jenkins.log**". Run the below command to get the password.

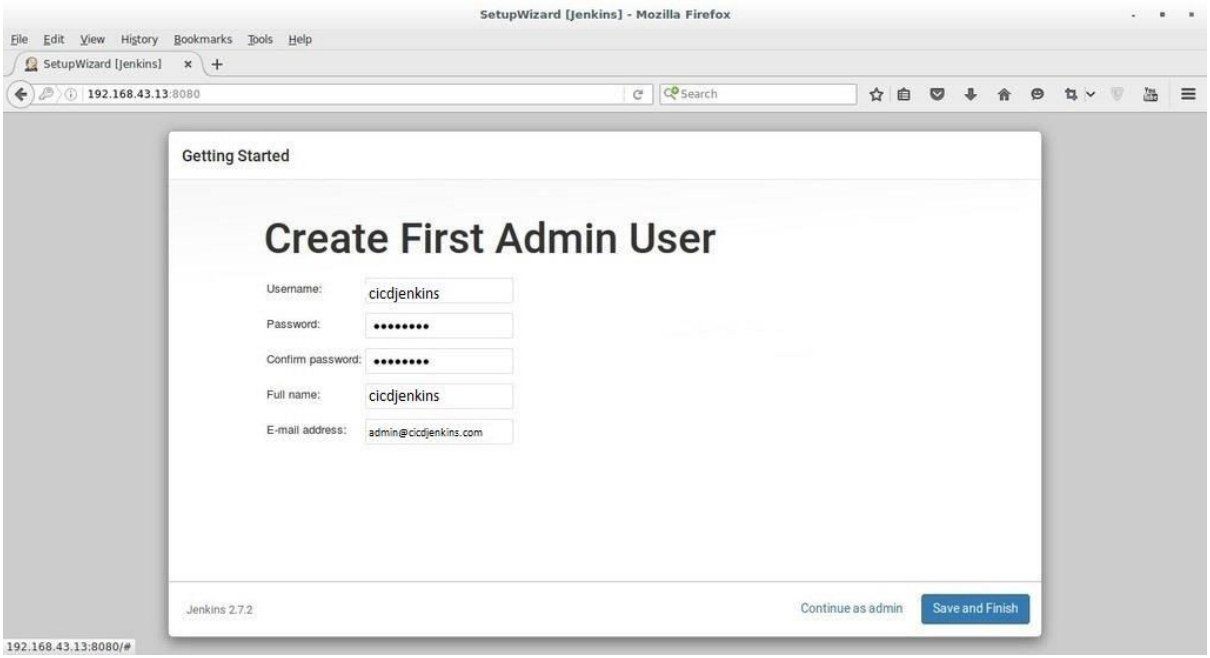
```
[ec2-user@ip-172-31-45-142 opt]# grep -A 5 password /var/log/jenkins/jenkins.log
```

Copy the password and paste it in above windows and click on Continue..

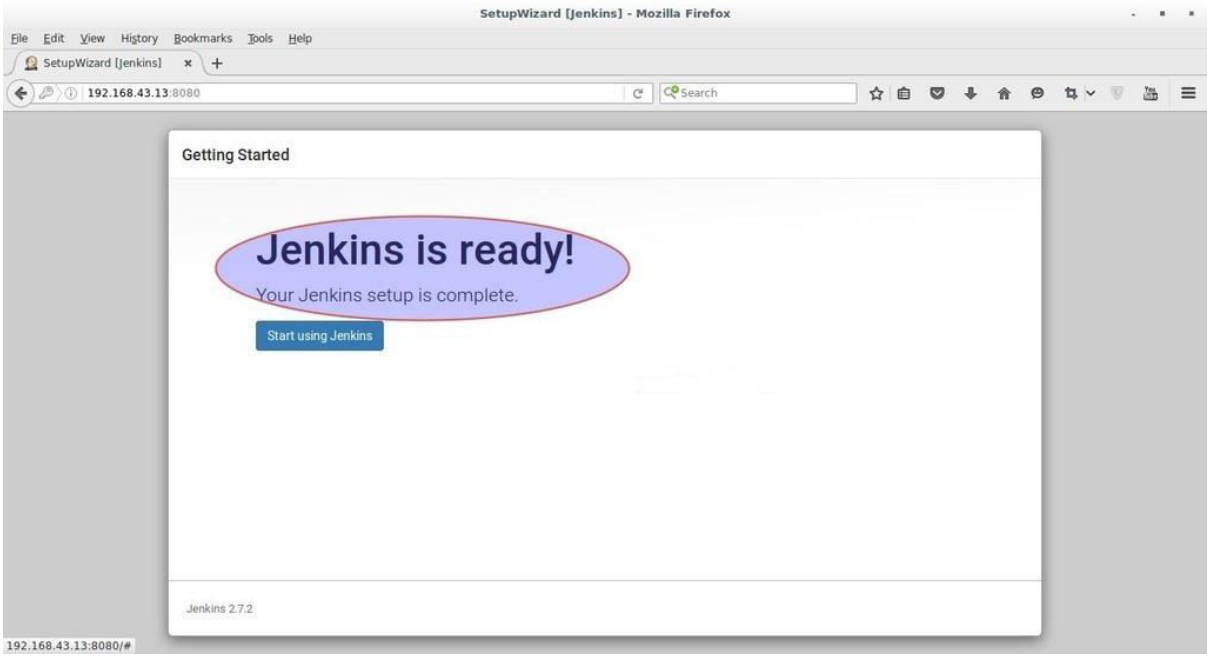
In the next windows Select the option : **Install suggested plugins**



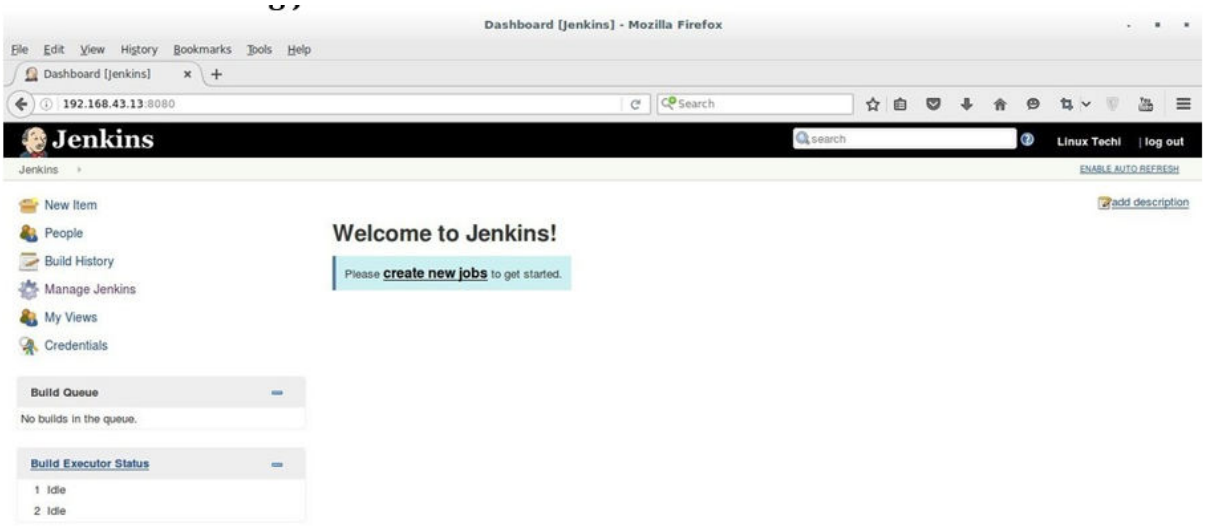
As we can see required plugin installation is in progress for Jenkins. Once it is done with plugin installation. It will ask to create Admin User



Click on Save and Finish

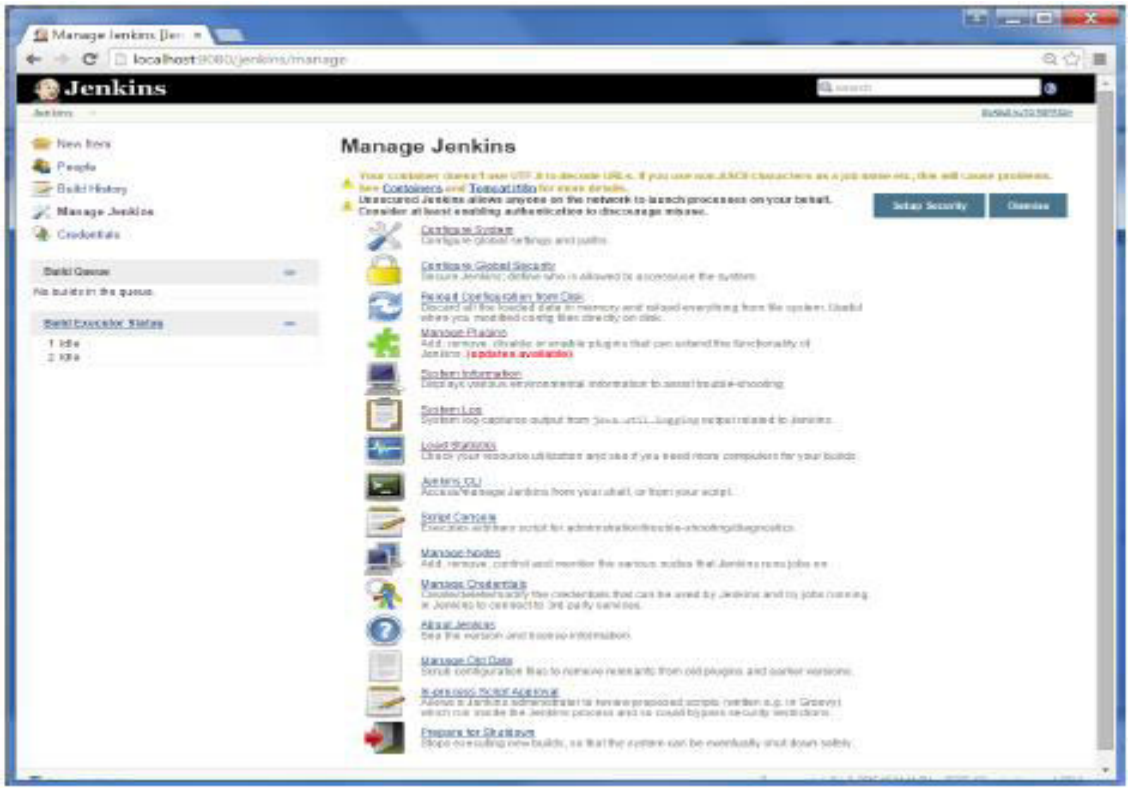


click on "**Start using Jenkins**"



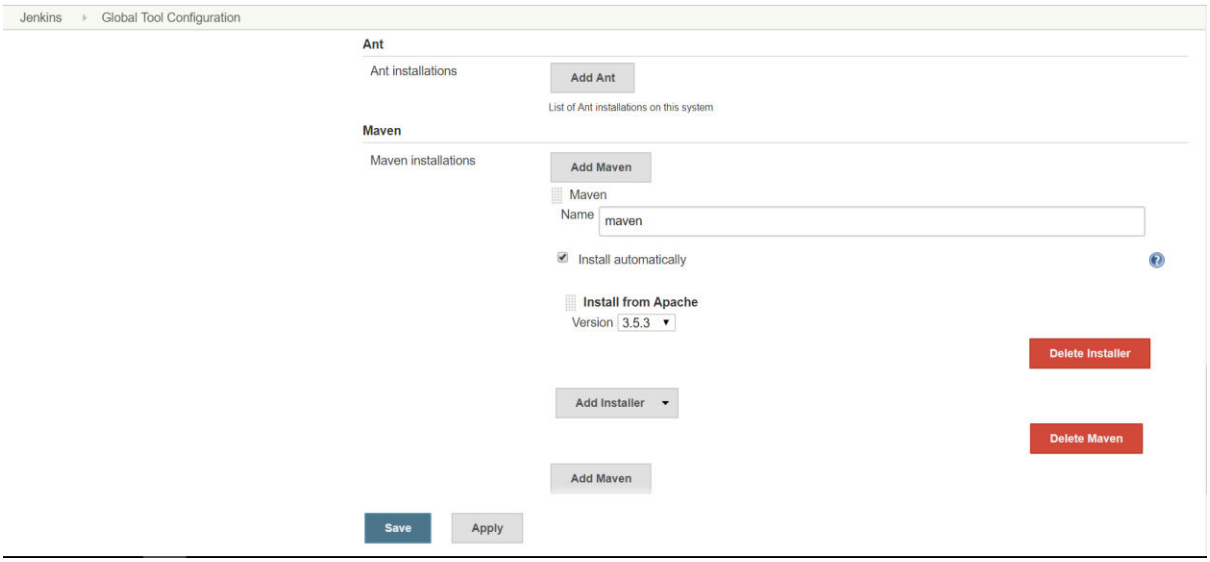
How to install and Configure Maven in Jenkins Server

1. Go to Jenkins Home page, On right side Click on Manage Jenkins



2. Click on Global Tool Configuration

In the Global Tool Configuration screen, scroll down till you see the Maven section and then click on the ‘Add Maven’ button.



Enter the name of Maven to identify, Click on Install Automatically check-box and Select type of version according to the project specification.

Then Click on Save button

3. To see the Maven Installation In Linux

```
[ec2-user@ip-172-31-45-142 opt]# cd /var/lib/jenkins/tools/Hudson.tasks.Maven_MavenInstallation/

Maven/
```

How to install and Configure Git in Linux

```
[ec2-user@ip-172-31-45-142 opt]# yum install git -y
```

To Configure Git Repository below command has to Run:

```
[ec2-user@ip-172-31-45-142 opt]# git clone https://repo/directory/git/repositoryname
```

How to install and Configure SonarQube on Linux7

Step:1 Download Sonarqube

Download the latest sonarqube installation file to /opt folder. You can get the latest download link from here. http://www.sonarqube.org/downloads/

Step:2 Install Sonarqube

```
[ec2-user@ip-172-31-45-142 opt]# wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-7.6.zip
```

Unzip package install & unzip the sonarqube

```
[ec2-user@ip-172-31-45-142 opt]# yum install unzip -y

[ec2-user@ip-172-31-45-142 opt]# unzip sonarqube-7.6.zip
```

Step:3 Start Sonarqube Service

Giving Permissions To User

```
[ec2-useradmin@cicd opt]$ sudo chown -R ec2-user:ec2-user sonarqube-7.6
```

Navigate to the start script directory

```
[ec2-useradmin@cicd opt]$ cd /opt/sonarqube-7.6/bin/linux-x86-64
```

Start the sonarqube service

```
[ec2-useradmin@cicd opt]$ sudo ./sonar.sh start
```

To check the status of sonarqube running

```
[ec2-useradmin@cicd opt]$ sudo ./sonar.sh status
```

Step:4 Setting up Sonarqube As A Service

1. Create a file /etc/init.d/sonar and copy the following content on to the file.

```
1  #!/bin/sh
2  #
3  # rc file for SonarQube
4  #
5  # chkconfig: 345 96 10
6  # description: SonarQube system (www.sonarsource.org)
7  #
8  ### BEGIN INIT INFO
9  # Provides: sonar
10 # Required-Start: $network
11 # Required-Stop: $network
12 # Default-Start: 3 4 5
13 # Default-Stop: 0 1 2 6
14 # Short-Description: SonarQube system (www.sonarsource.org)
15 # Description: SonarQube system (www.sonarsource.org)
16 ### END INIT INFO
17
18 /usr/bin/sonar $*
```

2. Now, create a symbolic link for /usr/bin/sonar with out sonarqube start scripts in the source file directory. i.e /opt/sonarqube/bin/linux-x86-64

```
[ec2-useradmin@cicd opt]$ sudo ln -s /opt/sonarqube/bin/linux-x86-64/sonar.sh /usr/bin/sonar
```

3. Change the file permissions and add sonar to the boot.

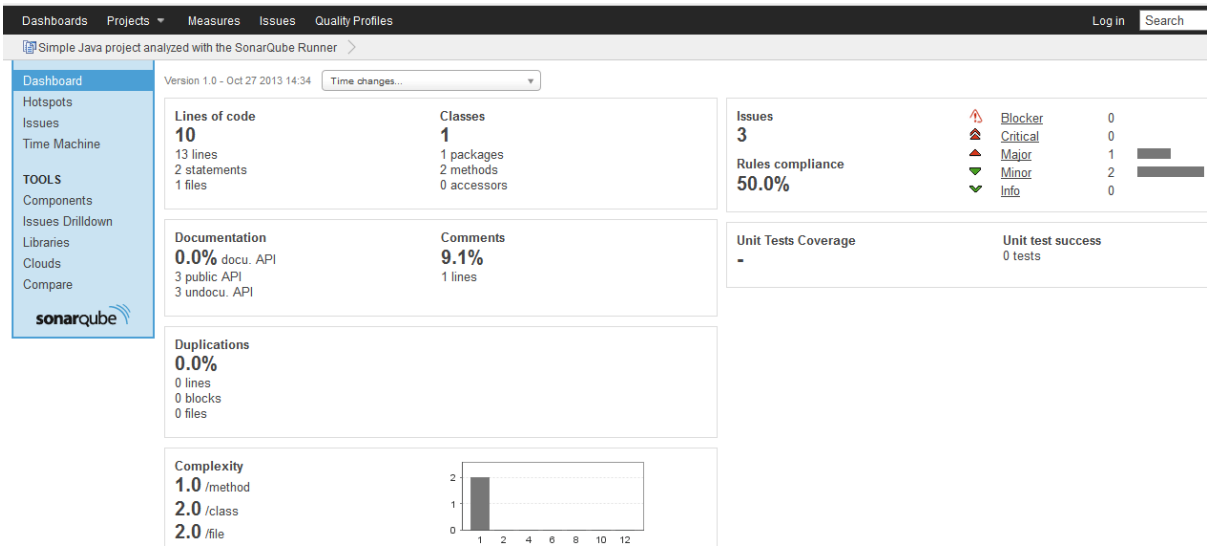
```
[ec2-useradmin@cicd opt]$ sudo chmod 755 /etc/init.d/sonar

[ec2-useradmin@cicd opt]$ sudo chkconfig –add sonar
```

4. Once you are done with all the above configurations, you can manage sonar using the following commands.

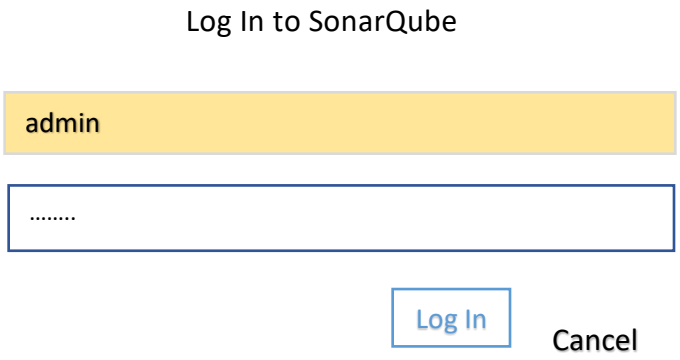
```
[ec2-useradmin@cicd opt]$ sudo service sonar start

[ec2-useradmin@cicd opt]$ sudo service sonar status

[ec2-useradmin@cicd opt]$ sudo service sonar stop

[ec2-useradmin@cicd opt]$ sudo service sonar restart
```

5. Now go to browser, <ipadress>:9000

Sonarqube page will open



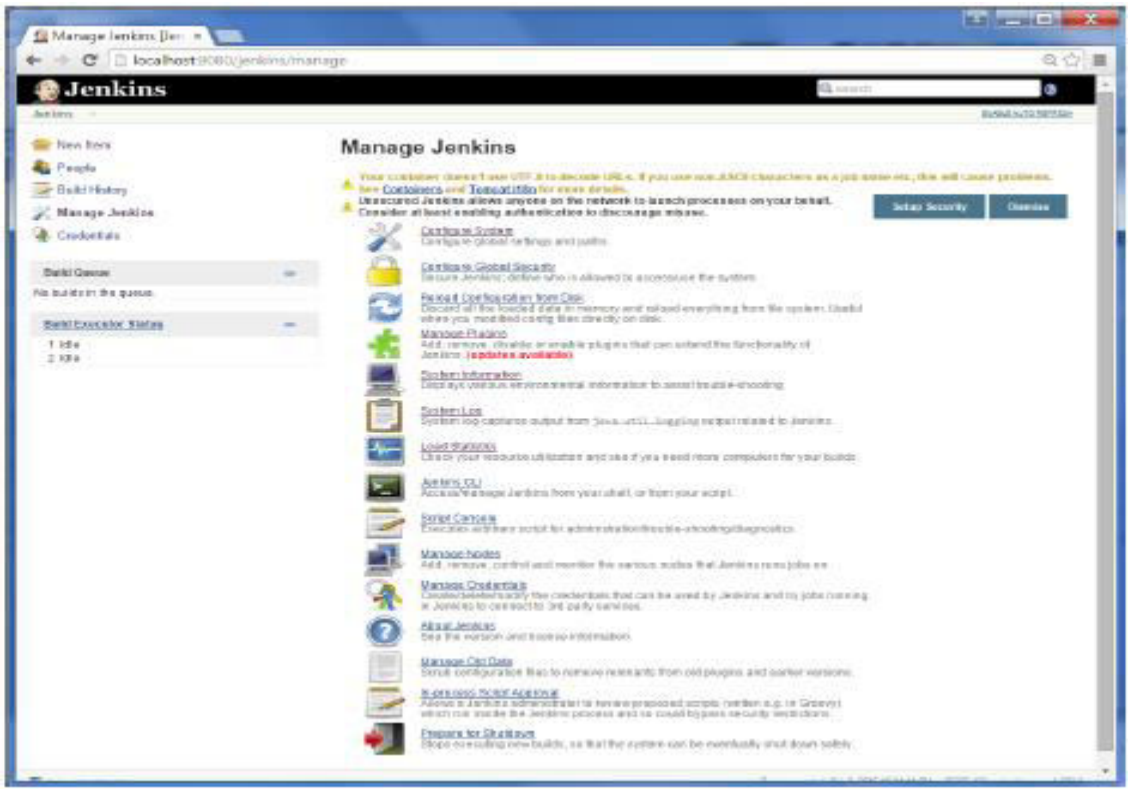
6. Click on signin button and login with default username and password



Or you can refer https://devopscube.com/setup-and-configure-sonarqube-on-linux/

Step:5 Cofiguring Sonarqube in Jenkins

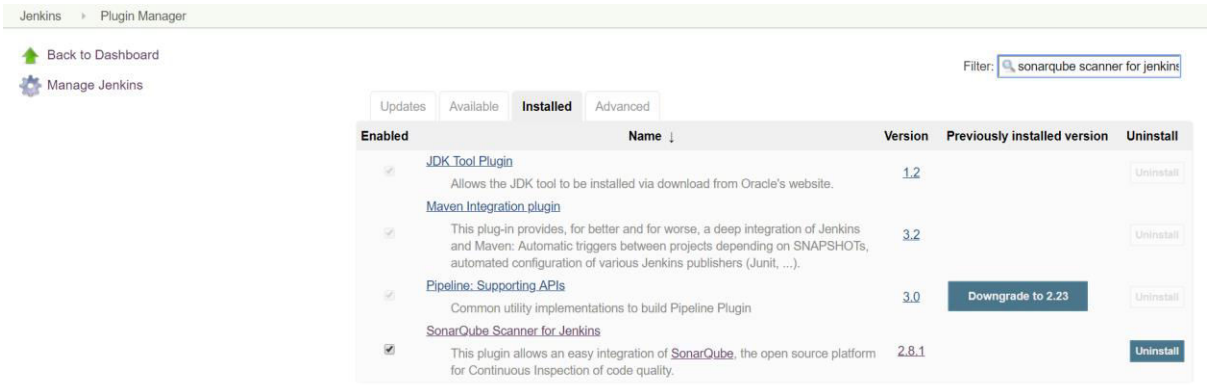
1. Go to Jenkins Home page, On right side Click on Manage Jenkins



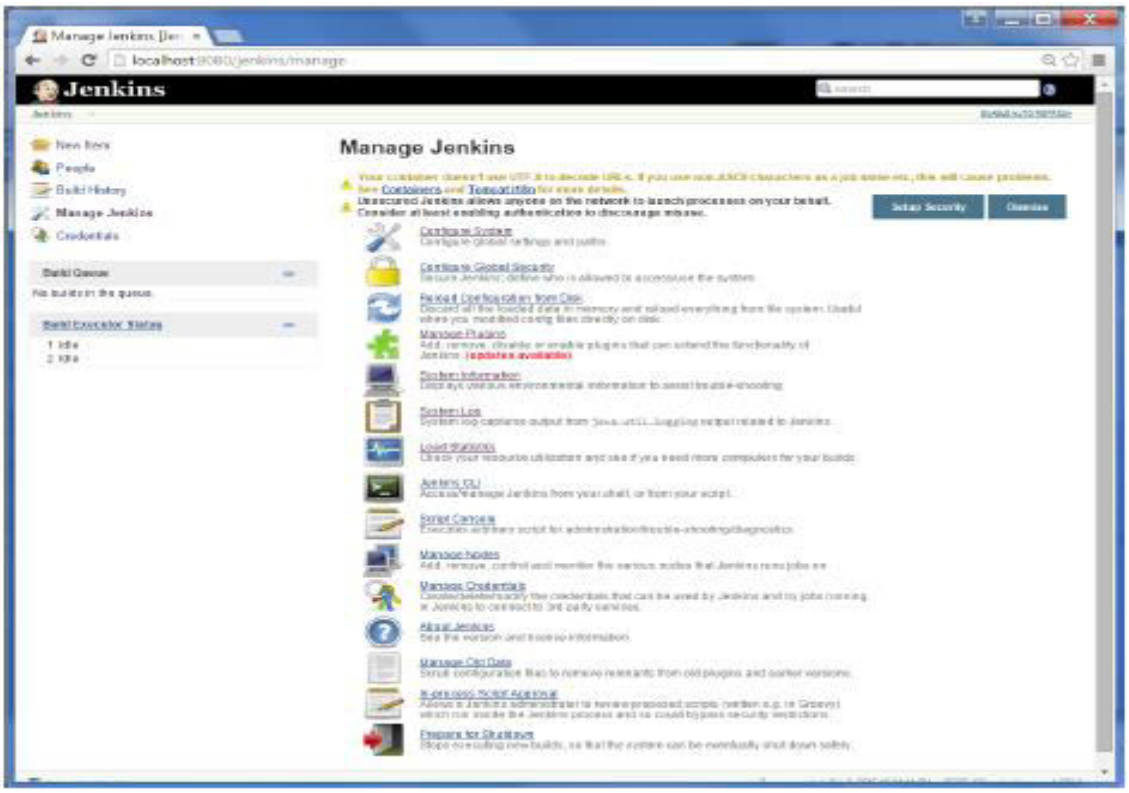
2. Now Click on Manage Plugins,

In the Plugin Manager screen, Click on Available Plugins and search for Sonarqube Scanner Plugin.

Now Click Install without restart check-box

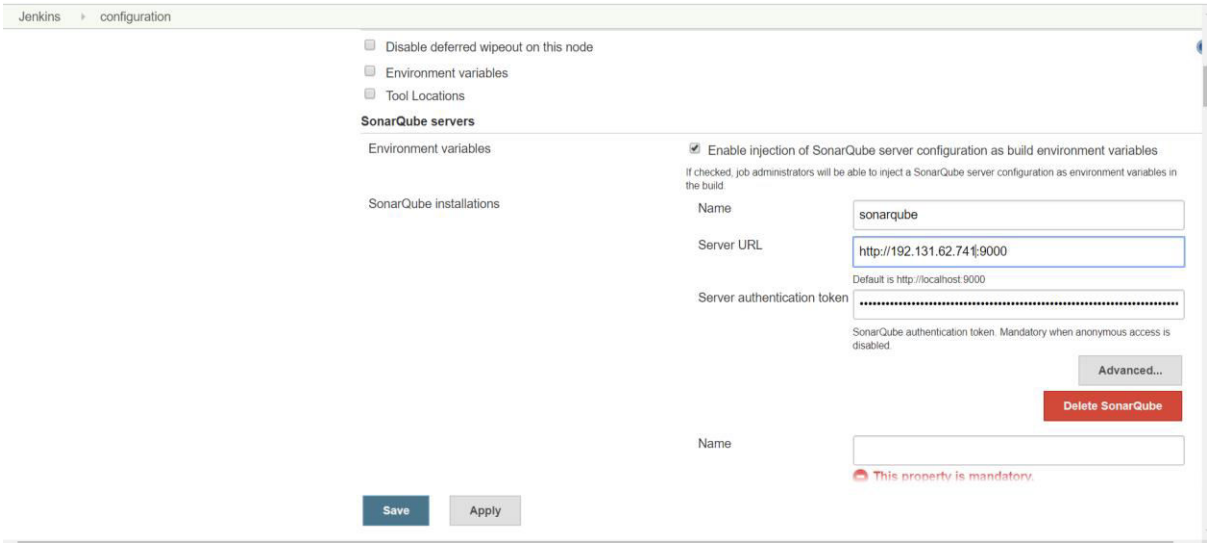


3. Go to Jenkins Home Page, Click on Manage Jenkins



4. Now Click on Cofigure system,

In the Configuration screen, scroll down till you see the SonarQube servers section and then click on the ‘Add SnonarQube’ button.



Click on Environment variables Check-box, Provide the details name, Server url and Security token of the Sonarqube, Then Click on Save button.

Now SonarQube was Configured with Jenkins Successfully.

How to install and Configure Nexus on Linux7

Step:1 Download Nexus

Download the latest Nexus installation file to /opt folder. You can get the latest download link from here. https://help.sonatype.com/repomanager3/download

Step:2 Install Nexus

```
[ec2-user@ip-172-31-45-142 opt]# wget https://download.sonatype.com/nexus/3/latest-unix.tar.gz
```

Untar the downloaded file

```
[ec2-user@ip-172-31-45-142 opt]# tar -xvf latest-unix.tar.gz
```

Rename the untared file to nexus

```
[ec2-user@ip-172-31-45-142 opt]# mv nexus-3.14.0-04 nexus
```

Step:3 Start Nexus Service

Giving Permissions To User

```
[ec2-useradmin@cicd opt]$sudo chown -R ec2-useradmin:ec2-useradmin nexus sonatype-work
```

Navigate to the start script directory

```
[ec2-useradmin@cicd opt]$ cd nexus/bin
```

Start the nexus service

```
[ec2-useradmin@cicd opt]$ sudo ./nexus start

[ec2-useradmin@cicd opt]$ sudo ./nexus status
```

Step:4 Setting up Nexus As A Service

1. Then create a `nexus` user with sufficient access rights to run the service. Change `NEXUS_HOME` to the absolute folder location in your `.bashrc` file, then save.

```
NEXUS_HOME=”/opt/nexus”
```

2. In `bin/nexus.rc` assign the user between the quotes in the line below:

```
Run_as_user=”nexus”
```

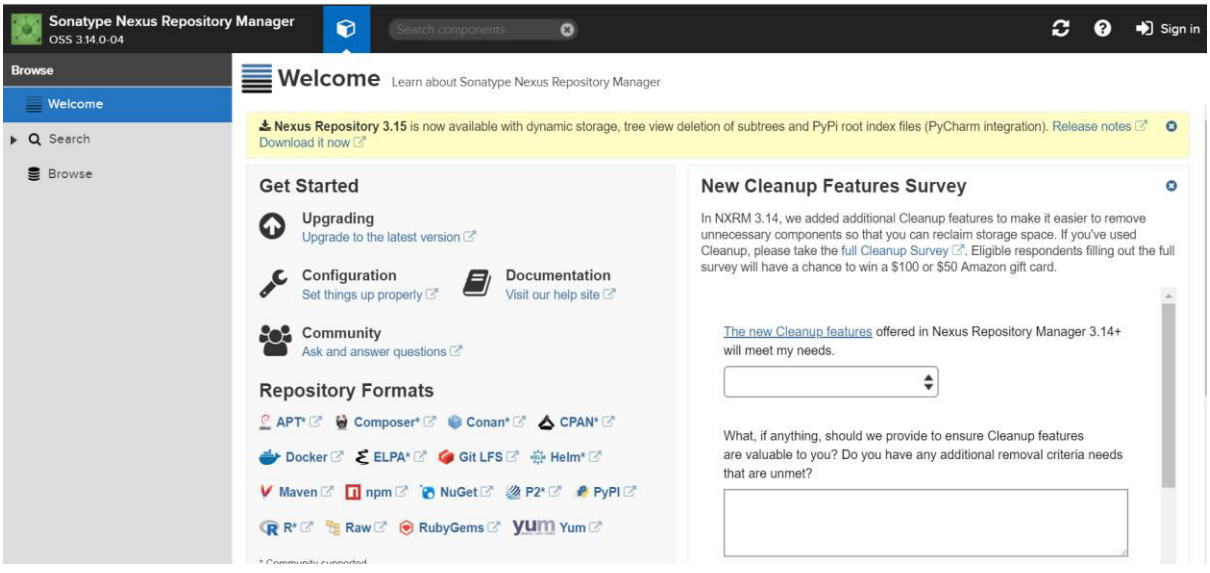
3. Now, create a symbolic link for /etc/init.d/nexus with out nexus start scripts in the source file directory. i.e NEXUS_HOME/bin/nexus

```
[ec2-useradmin@cicd opt]$ sudo ln -s $NEXUS_HOME/bin/nexus /etc/init.d/nexus
```

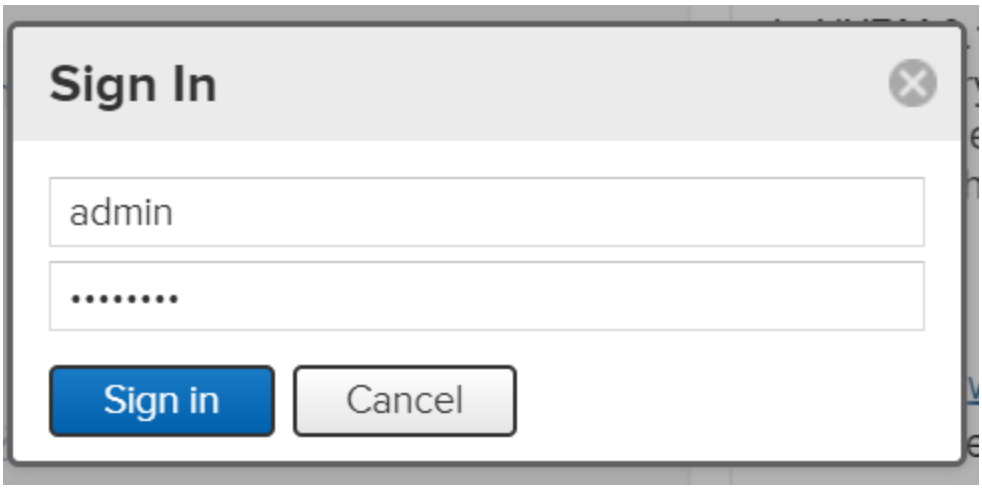
4. chkconfig, a tool that targets the initscripts in init.d to run the nexus service. Run these commands to activate the service:

```
[ec2-useradmin@cicd opt]$ cd /etc/init.d

[ec2-useradmin@cicd opt]$ sudo chkconfig –-add nexus

[ec2-useradmin@cicd opt]$ sudo chkconfig –-levels 345 nexus on

[ec2-useradmin@cicd opt]$ sudo service start
```

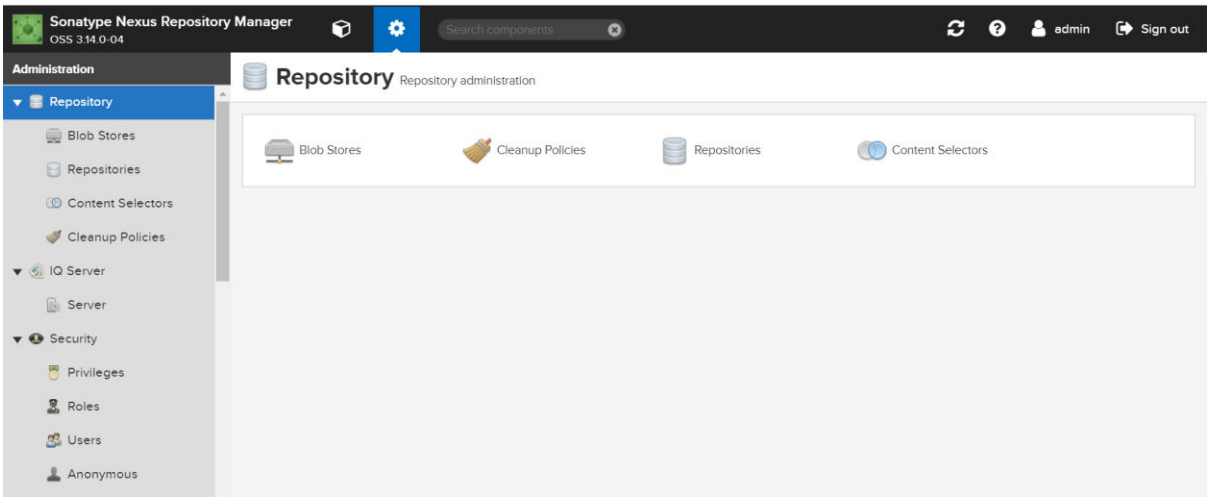
5. Go to browser <ipadress>:8081



6. Click on Signin button with default username and password



7. Nexus starts and page Will appear



Step:5 Configuring Nexus With Maven

1. Edit Settings.Xml in Maven Configuration file

```
[ec2-useradmin@cicd opt]$ sudo vi /var/lib/Jenkins/tools/Hudson.tasks.Maven_

MavenInstallation/maven/conf/settings.xml
```

2. Here Update Username, Password and Server details of Nexus in Servers path as mentioned below:

```
<server>

    <id>nexus</id>
```

```
    <username>username</username>

    <password>password</password>

</server>
```

Now Nexus was configured with Maven Successfully.