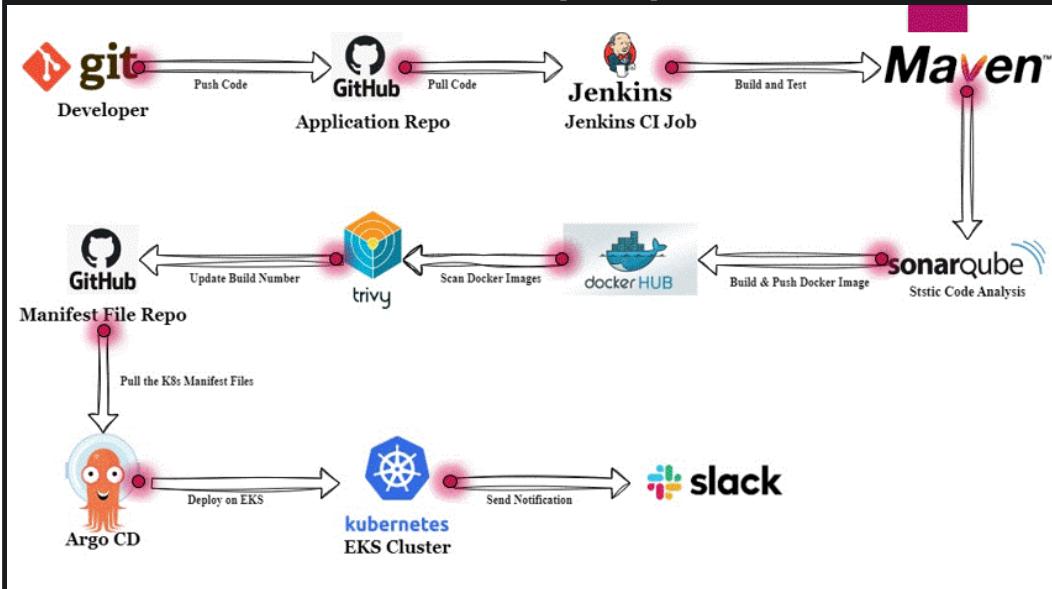


Real-Time DevOps Project | Deploy to Kubernetes Using Jenkins | End-to-End DevOps Project | CICD

Build a complete CI/CD pipeline using Jenkins, Maven, Docker, and the Elastic Kubernetes Services (EKS)



Project Description

Build a complete CI/CD pipeline using Jenkins, Maven, Docker, and the Elastic Kubernetes Services (EKS)

// In the CI/CD pipeline if the developer makes any changes to the ---> application repository on GitHub then it will trigger the CI job of the Jenkins ---> which will pull the code from the application repository and it will build and test the artifact using Maven ---> then we will do the static code analysis of the artifact using SonarQube ---> then we will build the Docker image and we will push that Docker image to the Docker hub, ---> After that we will scan the docker image using Trivy scan, here our CI job will end

** Once our CI job has been completed our CD job will automatically get triggered ---> and our CD job will first update the build number or the release number in the deployment yaml file on the git ops repository on the GitHub --->* and then ArgoCD will pull the manifest file, ---> updated manifest file from the GitHub repository and it will deploy the resources on the EKS cluster ---> and once our CD job is finished it will send the notification over the slack ---> or it can send the email also, so let's start building our project.//

YouTube Link For This Project

***** STAGES *****

1. Install and Configure the Jenkins-Master and Jenkins-Agent
2. Integrate Maven to Jenkins and Add GitHub Credentials to Jenkins
3. Create Pipeline Script(Jenkinsfile) for Build & Test Artifacts and Create CI Job on Jenkins
4. Install and Configure the SonarQube
5. Integrate SonarQube with Jenkins
6. Build and Push Docker Image using Pipeline Script
7. Setup Bootstrap Server for eksctl and Setup Kubernetes using eksctl
8. ArgoCD Installation on EKS Cluster and Add EKS Cluster to ArgoCD
9. Configure ArgoCD to Deploy Pods on EKS and Automate ArgoCD Deployment Job using GitOps GitHub Repository
- 10.DONE - Verify CI/CD Pipeline by Doing Test Commit on GitHub Application Repo

Install and Configure the Jenkins-Master and Jenkins-Agent

Create One EC2 instance t2. micro Ubuntu and 15GB Storage Configuration

The screenshot shows the AWS Management Console EC2 Instances page. It lists 7 instances in the N. Virginia region. The instances are:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
jenkins-agent	i-017b2726df03682a6	Running	t2.micro	2/2 checks passed	View alarms	us-east-1d	ec2-19-213-4-9f
sonarqube	i-0dd1f1e7f0925a39f	Running	t3.medium	2/2 checks passed	View alarms	us-east-1a	ec2-5a-242-174
eks-server	i-0951d8f244b1b991e	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-54-145-91-1
virtualtechbox...	i-0811086be87e5dd6	Running	t2.small	2/2 checks passed	View alarms	us-east-1a	ec2-19-127-22
jenkins master	i-09ac5f6b78027354e	Running	t2.micro	2/2 checks passed	View alarms	us-east-1c	ec2-18-234-227
virtualtechbox...	i-063c12a5454670bb3	Running	t2.small	2/2 checks passed	View alarms	us-east-1b	ec2-3-230-1-59
virtualtechbox...	i-03fc31d7fa9b7696e	Running	t2.small	2/2 checks passed	View alarms	us-east-1b	ec2-44-193-24-2

Copy the public IP of this instance and Connect via MobaXterm

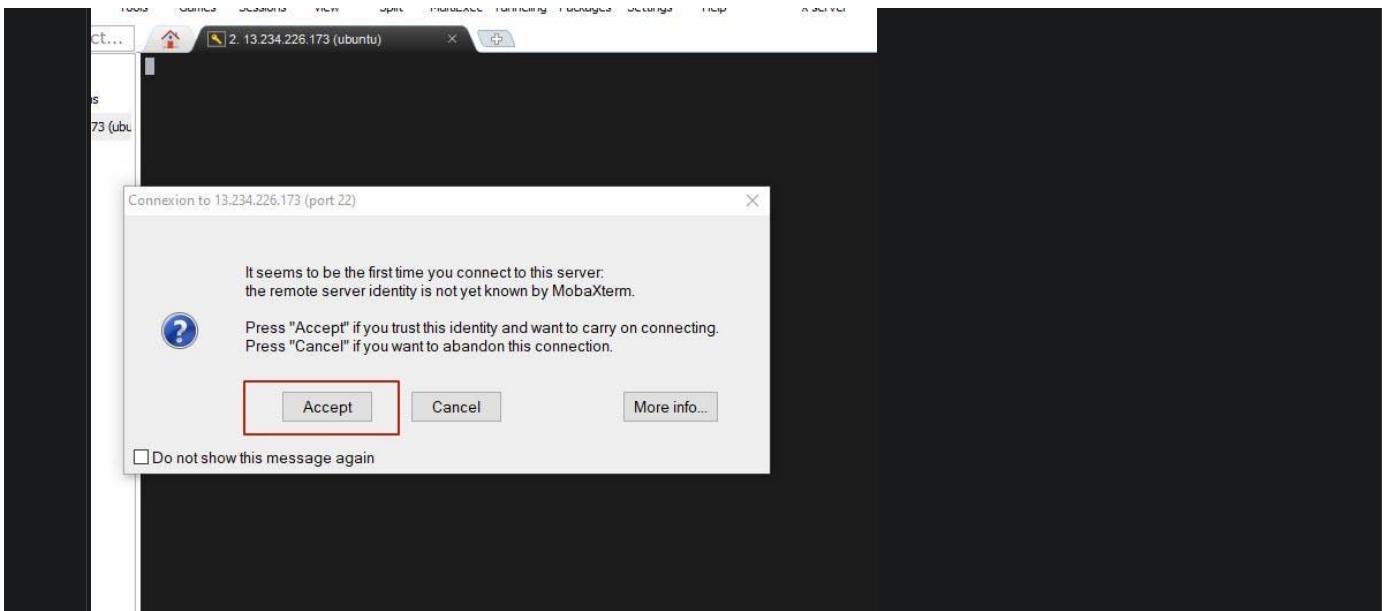
The screenshot shows the MobaXterm application window and the AWS EC2 Instances page.

MobaXterm Session Configuration:

- Basic SSH settings:** Remote host: 13.234.226.173, Specify username: ubuntu, Port: 22.
- Advanced SSH settings:** X11-Forwarding, Compression, Execute command: (empty), Do not exit after command ends, SSH-browser type: SFTP protocol, Use private key: C:\Users\SUNIL\Downloads\Linux- (selected), Expert SSH settings button.

AWS EC2 Instances Page:

- Instances (1/1) Info: Jenkins-Master (Running).
- Details tab: Public IPv4 address copied (13.234.226.173), Open address.
- Networking tab: Instance summary, Public IPv4 address 13.234.226.173, Public IPv4 DNS ec2-13-234-226-173.ap-south-1.compute.amazonaws.com.



After connecting update the machine and upgrade

```
sudo apt update -y
```

```
sudo apt upgrade -y
```

A screenshot of the MobaXterm terminal window. The title bar shows '2. 13.234.226.173 (ubuntu)'. The command 'sudo apt update -y' is typed into the terminal and is highlighted with a red selection bar.

A screenshot of the MobaXterm terminal window. The title bar shows '2. 13.234.226.173 (ubuntu)'. The command 'sudo apt upgrade -y' is typed into the terminal and is highlighted with a red selection bar.

Daemons using outdated libraries

Which services should be restarted?

- [*] chrony.service
- [*] cron.service
- [] dbus.service
- [] getty@tty1.service
- [] networkd-dispatcher.service
- [*] packagekit.service
- [*] polkit.service
- [*] serial-getty@ttyS0.service
- [*] snapd.service
- [*] ssh.service
- [] unattended-upgrades.service
- [] user@1000.service

<0k>

<Cancel>

Package configuration

Pending kernel upgrade

Newer kernel available

The currently running kernel version is 6.2.0-1012-aws which is not the expected kernel version 6.2.0-1015-aws.

Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.

<0k>

Press Enter

```
sudo vim /etc/hostname
```

Delete old name and Type: Jenkins-Master

```
ubuntu@ip-172-31-37-237:~$ sudo vim /etc/hostname
```

Open Hostname file
And change hostname:- Jenkins-Master

After that reboot system `sudo init 6`

```
ubuntu@ip-172-31-37-237:~$ sudo vim /etc/hostname  
ubuntu@ip-172-31-37-237:~$ hostname  
ip-172-31-37-237 Reboot system  
ubuntu@ip-172-31-37-237:~$ sudo init 6  
ubuntu@ip-172-31-37-237:~$ Remote side unexpectedly closed network connection
```

Session stopped
- Press **<Return>** to exit tab
- Press **R** to restart session
- Press **S** to save terminal output to file

```
ubuntu@Jenkins-Master:~$ sudo apt install openjdk-17-jre
```

**After Rebooting
Install java with above command**

```
ubuntu@Jenkins-Master:~$ java --version  
openjdk 17.0.8.1 2023-08-24  
OpenJDK Runtime Environment (build 17.0.8.1+1-Ubuntu-0ubuntu122.04)  
OpenJDK 64-Bit Server VM (build 17.0.8.1+1-Ubuntu-0ubuntu122.04, mixed mode, sharing)
```

```
sudo wget -O /usr/share/keyrings/jenkins-  
keyring.asc https://pkg.jenkins.io/debian/jenkins.io-2023.key | echo deb [signed-  
by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian binary/ |  
sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null sudo apt-get update  
sudo apt-get install jenkins
```

The screenshot shows the Jenkins User Handbook page with the 'Weekly release' section highlighted. It includes a note about weekly releases and a command-line example for installing Jenkins via apt.

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian/jenkins.io-2023.key
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
```

A terminal window titled 'ubuntu@Jenkins-Master:~\$' shows the execution of the above command. The output indicates the Java version (openjdk 17.0.8.1) and the successful download and installation of Jenkins.

Paste here copied code for installation of jenkins

You can enable the Jenkins service to start at boot with the command:

COPY
COPY

sudo systemctl enable jenkins

You can start the Jenkins service with the command:

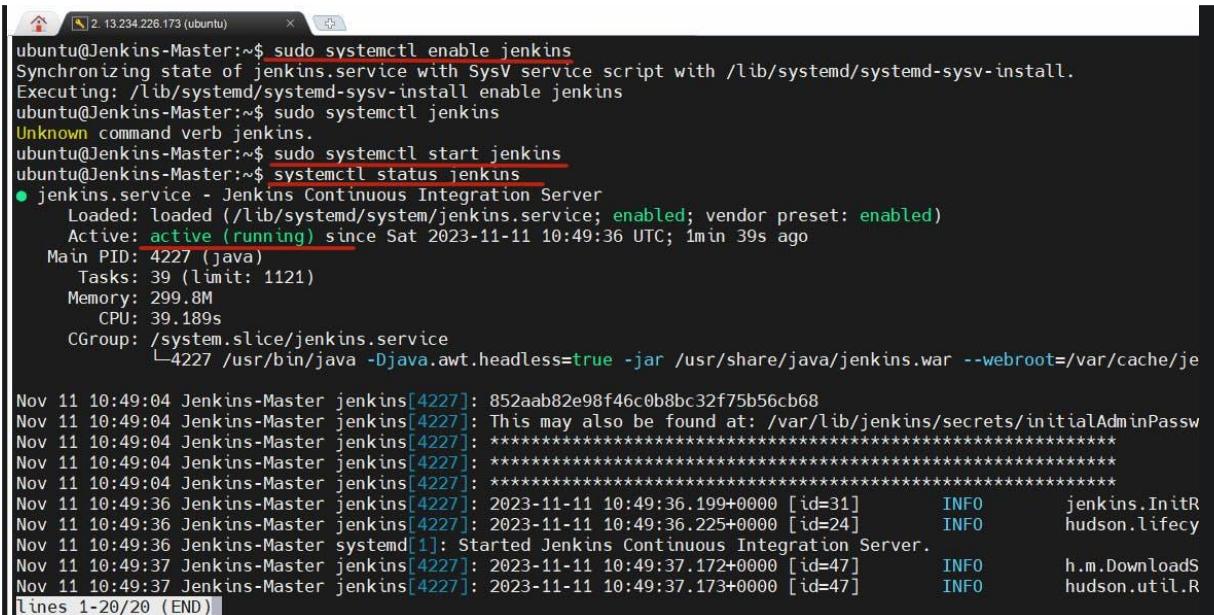
COPY
COPY

sudo systemctl start jenkins

You can check the status of the Jenkins service using the command:

COPY
COPY

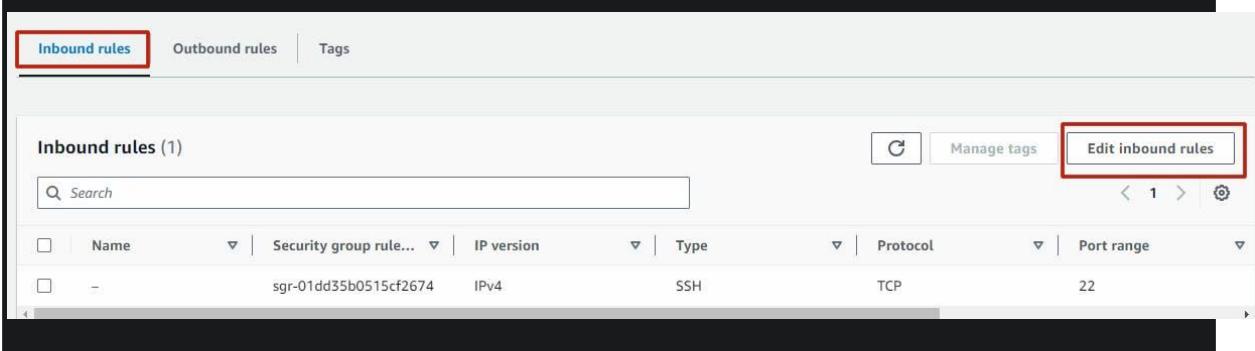
```
sudo systemctl status jenkins
```



```
ubuntu@Jenkins-Master:~$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
ubuntu@Jenkins-Master:~$ sudo systemctl jenkins
Unknown command verb jenkins.
ubuntu@Jenkins-Master:~$ sudo systemctl start jenkins
ubuntu@Jenkins-Master:~$ systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2023-11-11 10:49:36 UTC; 1min 39s ago
       Main PID: 4227 (java)
          Tasks: 39 (limit: 1121)
        Memory: 299.8M
         CPU: 39.189s
      CGroup: /system.slice/jenkins.service
              └─4227 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/instance/war

Nov 11 10:49:04 Jenkins-Master jenkins[4227]: 852aab82e98f46c0b8bc32f75b56cb68
Nov 11 10:49:04 Jenkins-Master jenkins[4227]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Nov 11 10:49:04 Jenkins-Master jenkins[4227]: ****
Nov 11 10:49:04 Jenkins-Master jenkins[4227]: ****
Nov 11 10:49:04 Jenkins-Master jenkins[4227]: ****
Nov 11 10:49:36 Jenkins-Master jenkins[4227]: 2023-11-11 10:49:36.199+0000 [id=31]      INFO      jenkins.InitR
Nov 11 10:49:36 Jenkins-Master jenkins[4227]: 2023-11-11 10:49:36.225+0000 [id=24]      INFO      hudson.lifecycle
Nov 11 10:49:36 Jenkins-Master systemd[1]: Started Jenkins Continuous Integration Server.
Nov 11 10:49:37 Jenkins-Master jenkins[4227]: 2023-11-11 10:49:37.172+0000 [id=47]      INFO      h.m.DownloadS
Nov 11 10:49:37 Jenkins-Master jenkins[4227]: 2023-11-11 10:49:37.173+0000 [id=47]      INFO      hudson.util.R
[lines 1-20/20 (END)]
```

Go to instance security group and allow port no. 8080 for jenkins



Inbound rules (1)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-01dd35b0515cf2674	IPv4	SSH	TCP	22

Manage tags

Edit inbound rules

Instances (1/1) Info

[Find Instance by attribute or tag \(case-sensitive\)](#)

Name	Instance ID	Instance state	Instance ...
<input checked="" type="checkbox"/> Jenkins-Master	i-054dbd4...	Running	t2.micro

Instance: i-054dbd4a7875e9a89 (Jenkins-Master)

Details **Security** Networking Storage Status checks Monitoring

▼ Security details

IAM Role	Owner ID
-	955232677187
Security groups	
<input checked="" type="checkbox"/> sg-0e3dc9f2379962ca5 (launch-wizard-7)	

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type info	Protocol info	Port range info	Source info	Description - optional <small>Info</small>
sgr-01dd35b0515cf2674	SSH	TCP	22	Custom	<input type="text"/> 0.0.0.0/0
-	Custom TCP	TCP	8080	Anyw...	<input type="text"/> 0.0.0.0/0

Add rule

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel **Save rules**

Now launch another instance with the name of Jenkins-Agent

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

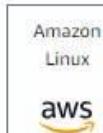
Name and tags Info

Name

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

[Recents](#)[Quick Start](#)

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-0287a05f0ef0e9d9a (64-bit (x86)) / ami-0b6581fde9e6e7779 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-09-19

Architecture

64-bit (x86)

AMI ID

ami-0287a05f0ef0e9d9a

Verified provider

▼ Instance type Info

Instance type

t2.micro
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0724 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour

Free tier eligible

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Linux-VM-key

[Create new key pair](#)

EBS volumes will be select 15GB

Storage (volumes) [Info](#)

EBS Volumes [Hide details](#)

Volume 1 (AMI Root) (Custom)

Storage type Info	Device name - required Info	Snapshot Info
EBS	/dev/sda1	snap-040e3cc70bbb7260f
Size (GiB) Info	Volume type Info	IOPS Info
15	gp2	100 / 3000
Delete on termination Info	Encrypted Info	KMS key Info
Yes	Not encrypted	Select
KMS keys are only applicable when encryption is set on this volume.		
<p>Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage X</p>		
Add new volume		
The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance		

Number of instances [Info](#)
1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0287a05f0ef0e9d9a

Virtual server type (instance type)
t2.micro

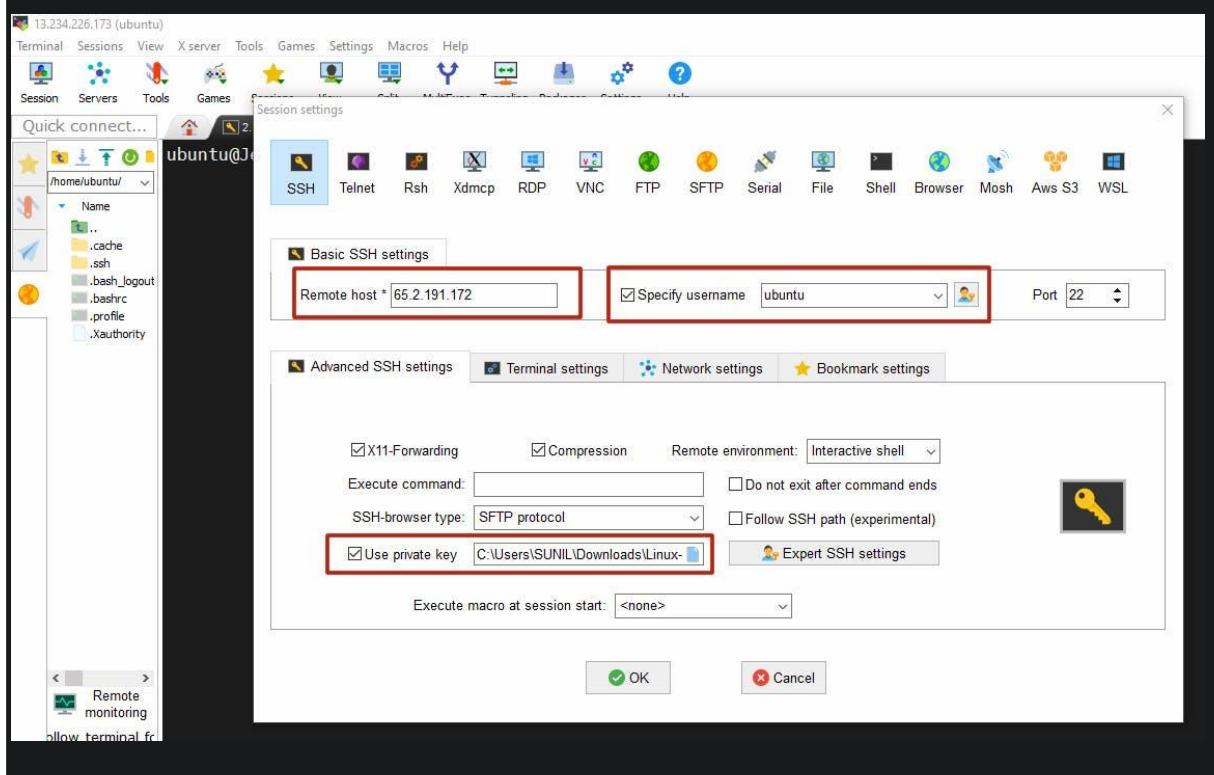
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 15 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS

[Cancel](#) [Launch instance](#) [Review commands](#)

Click on Launch Instance and Connect this instance with the help of mobaxterm



Instances (1/2) [Info](#)

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance ...	Status check
Jenkins-Master	i-054dbd4...	Running	Q Q	t2.micro
Jenkins-Agent	i-07ef1861...	Running	Q Q	t2.micro

Instance: i-07ef186135b0013d4 (Jenkins-Agent)

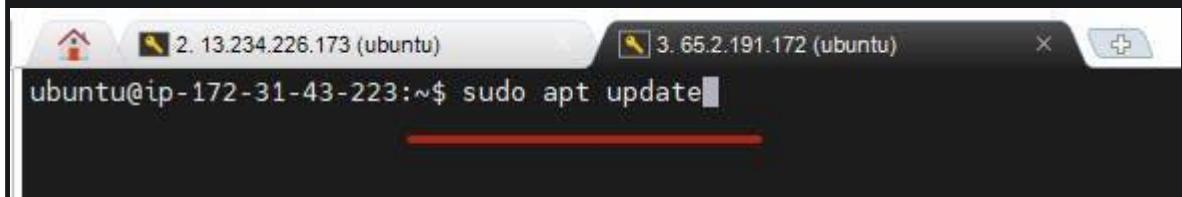
Details Security Networking Storage Status checks Monitoring Tags

Instance summary [Info](#)

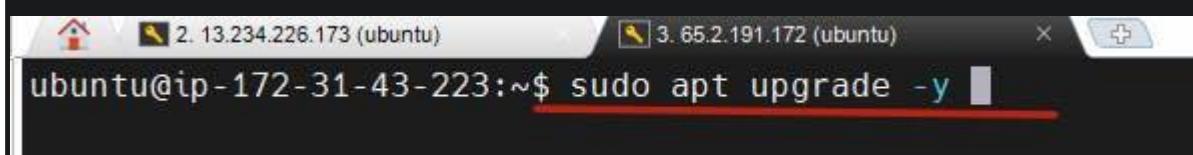
Instance ID	Public IPv4 address
i-07ef186135b0013d4 (Jenkins-Agent)	65.2.191.172 open address
IPv6 address	Instance state
-	Running

After connecting the server first we will update and upgrade the server with the help of the following command

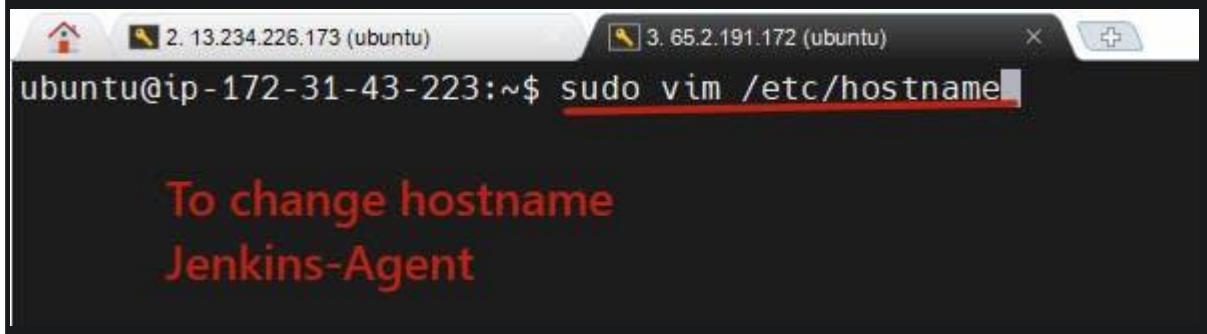
```
sudo apt update
```



```
sudo apt upgrade -y
```



To change hostname --> `sudo vim /etc/hostname`



```
ubuntu@ip-172-31-43-223:~$ sudo vim /etc/hostname
ubuntu@ip-172-31-43-223:~$ Hostname changed so reboot to apply changes
ubuntu@ip-172-31-43-223:~$ sudo init 6
ubuntu@ip-172-31-43-223:~$ Remote side unexpectedly closed network connection

Session stopped
- Press <Return> to exit tab
- Press R to restart session
- Press S to save terminal output to file
```

Now install java on this instance `sudo apt install openjdk-17-jre -y`



The screenshot shows a terminal window with the following content:

- Icons for Home, Help, and a connection labeled "2. 13.234.226.173 (ubuntu)".
- A tab labeled "3. 65.2.191.172 (ubuntu)".
- An "X" button to close the window.
- A plus sign icon to open a new tab.
- Text input field containing the command: `ubuntu@Jenkins-Agent:~$ sudo apt install openjdk-17-jre -y`.
- A large red text overlay at the bottom right of the terminal area: **Now install java with this command**.

Now install docker on this server --> `sudo apt-get install docker.io -y`

```
ubuntu@Jenkins-Agent:~$ java -version
openjdk version "17.0.8.1" 2023-08-24
To check Java installed or not
OpenJDK Runtime Environment (build 17.0.8.1+1-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 17.0.8.1+1-Ubuntu-0ubuntu122.04, mixed mode, sharing)
ubuntu@Jenkins-Agent:~$ sudo apt-get install docker.io -y
```

Now install docker with the help of above command

Give full rights of current user `sudo usermod -aG docker $USER`

```
ubuntu@Jenkins-Agent:~$ sudo usermod -aG docker $USER
ubuntu@Jenkins-Agent:~$ sudo init 6
```

Give full rights to current user on the docker with this command
and then reboot the system

`sudo vim /etc/ssh/sshd_config` ---- make some changes

```
ubuntu@Jenkins-Agent:~$ sudo vim /etc/ssh/sshd_config
```

Open this file and make some changes

Remove # from this line and save the file

```
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes Remove # from this line
Also from this line
# Expect .ssh/authorized_keys2 to be disregarded by default in future.
AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2
#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes
```

Do this settings same on Jenkins-Master server also

```
ubuntu@Jenkins-Master:~$ sudo vim /etc/ssh/sshd_config
```

Do same settings in Jenkins-Master server

```
PubkeyAuthentication yes  
# Expect .ssh/authorized_keys2 to be disregarded by default in future.  
AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2  
  
#AuthorizedPrincipalsFile none  
  
#AuthorizedKeysCommand none  
#AuthorizedKeysCommandUser nobody  
  
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts  
#HostbasedAuthentication no  
# Change to yes if you don't trust ~/.ssh/known_hosts for  
# HostbasedAuthentication  
#IgnoreUserKnownHosts no  
# Don't read the user's ~/.rhosts and ~/.shosts files  
#IgnoreRhosts yes  
  
# To disable tunneled clear text passwords, change to no here!  
PasswordAuthentication no  
#PermitEmptyPasswords no
```

And Reload sshd service ---> `sudo service sshd reload`

on both instance same command



```
ubuntu@Jenkins-Master:~$ sudo vim /etc/ssh/sshd_config  
ubuntu@Jenkins-Master:~$  
ubuntu@Jenkins-Master:~$  
ubuntu@Jenkins-Master:~$ sudo service sshd reload  
Reload sshd service with this command  
Same command copy to Jenkins-Agent server also
```

Now on Jenkins-Master server generate secrete key ---> `ssh-keygen -t ed25519`

```
ubuntu@Jenkins-Master:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:qaZJp8HszyinJPehQho7KQdCmuW4WpxqmyD13+T1NbI ubuntu@Jenkins-Master
The key's randomart image is:
+--[ED25519 256]--+
|                               |
|                               |
|                               |
|                               |
| ..      .|
| o=.      S|
| B+=      .|
| B** B + . . . o|
| @B+=.% + . . + .|
| ***++B.+ o E|
+---[SHA256]---+
ubuntu@Jenkins-Master:~$ cd .ssh/
ubuntu@Jenkins-Master:~/ssh$ ls
authorized_keys id_ed25519 id_ed25519.pub id_rsa id_rsa.pub
ubuntu@Jenkins-Master:~/ssh$
```

Go to cd .ssh/ path and it will generated public key and private key `id_ed25519` and `id_ed25519.pub`

Now copy the private key and copy this key on Jenkins-Agent authorized keys folder and save it.

The screenshot shows a desktop environment with a terminal window open. The terminal window title is "ubuntu@Jenkins-Agent:~\$". The log output is as follows:

```
ubuntu@Jenkins-Agent:~$ pwd
/home/ubuntu
ubuntu@Jenkins-Agent:~$ cd .ssh/
ubuntu@Jenkins-Agent:~/ssh$ ls
authorized_keys
```

MobaXterm

The file "authorized_keys" has been modified.
Do you want to replace the remote file on "65.2.191.172" with the new one?

```
cat authorized_keys
```

It will shows copied key.

```
ubuntu@Jenkins-Agent:~/ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQcmBhoy302boyLw5e8Y0heD6se3hjd27L+hy+NgHii20ed0PDeckLYMQHa6BkJfTcBR/LA8WkPVu60PxR0Gs6wosb30RvRX47VXSuwdsC35BLcGNkgYjMvCivaYectnGkf6HnVu0/XYmlSf9gypXkbBxo/Knset+xEH/0t0A55/mvaZ5aab08vuLXxyPqFbhxZmp5eedeQGmZh6UA8i02Wt2Sn/Cfl6XNhtnGLxjfV2hm5uj2WDJ0ttyMa04lyvGI3qnSY2xBw+Nildy5/VwGCB8x0ZJMr7y2td0km3iQ4x2Tqxw9nYwvq/7i9fx/K0jTEBobsze+x Linux-VM-key
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAInyvw0jyi5guR4rH4U0uQ1lp753x5u9vbbHVUtBZMLAo ubuntu@Jenkins-Master
ubuntu@Jenkins-Agent:~/ssh$
```

Now we need to access jenkins so copy the Jenkins-Master server public ip and paste it in browser <PublicIP>:8080

Instances (1/2) Info

Name	Instance ID	Instance state	Instance ...	Status check	Alarm s...
<input checked="" type="checkbox"/> Jenkins-Master	i-054dbd4...	Running	t2.micro	2/2 checks passed	No alarm
<input type="checkbox"/> Jenkins-Agent	i-07ef1861...	Running	t2.micro	-	No alarm

Instance: i-054dbd4a7875e9a89 (Jenkins-Master)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary

Instance ID	Public IPv4 address
i-054dbd4a7875e9a89 (Jenkins-Master)	13.234.226.173 [open address]
IPv6 address	Instance state
-	Running

It will show jenkins dashboard

Go to given path for password

The screenshot shows a web browser window with the URL `13.234.226.173:8080/login?from=%2F` in the address bar. The page title is "Getting Started". The main content is titled "Unlock Jenkins" with the sub-instruction: "To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:". Below this, a code block highlights the path `/var/lib/jenkins/secrets/initialAdminPassword`. A note below says "Please copy the password from either location and paste it below." There is a text input field labeled "Administrator password" with a placeholder "Paste your password here". A blue "Continue" button is at the bottom right.

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

It will show password copy it

The screenshot shows a terminal window with two tabs. The active tab shows the command `sudo cat /var/lib/jenkins/secrets/initialAdminPassword` being run, with the output being the password `852aab82e98f46c0b8bc32f75b56cb68`, which is highlighted with a red box.

And paste it on Administrator password and click on continue

The screenshot shows the Jenkins 'Unlock Jenkins' setup page. At the top, there are browser navigation icons and the URL '13.234.226.173:8080/login?from=%2F'. Below the header, the title 'Getting Started' is displayed. The main section features a large heading 'Unlock Jenkins'. A text block explains that a password has been written to the log (not sure where to find it?) and points to the file '/var/lib/jenkins/secrets/initialAdminPassword'. A text input field labeled 'Administrator password' contains several dots, indicating a password has been entered. A blue 'Continue' button is located at the bottom right of the form.

Click on Install Suggested plugins

The screenshot shows the Jenkins 'Customize Jenkins' setup page. At the top, there are browser navigation icons and the URL '13.234.226.173:8080'. Below the header, the title 'Getting Started' is displayed. The main section features a large heading 'Customize Jenkins'. A text block states that plugins extend Jenkins with additional features to support many different needs. Two options are presented in boxes: 'Install suggested plugins' (selected) and 'Select plugins to install'. The 'Install suggested plugins' box contains the text 'Install plugins the Jenkins community finds most useful.' The 'Select plugins to install' box contains the text 'Select and install plugins most suitable for your needs.' At the bottom left, a footer note says 'Jenkins 2.431'.

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	✓ Mailer	

Jenkins 2.431

- ** American Project
- ** Resource Disposer
- Workspace Cleanup
- Ant
- ** Durable Task
- ** Pipeline: Nodes and Processes
- ** Pipeline: SCM Step
- ** Pipeline: Groovy
- ** Pipeline: Job
- ** Jakarta Activation API
- ** Jakarta Mail API
- ** Apache HttpComponents Client 4.x API
- Mailer
- ** Pipeline: Basic Steps
- Gradle
- ** Pipeline: Milestone Step
- ** Pipeline: Build Step
- ** Variant
- ** - required dependency

Click on save and finish

Getting Started

Instance Configuration

Jenkins URL:

<http://13.234.226.173:8080/>

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features, including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is not saved yet and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.431

Not now

Save and Finish

create first admin user and click on save and continue

Getting Started

Create First Admin User

Username

admin

Password

Confirm password

Full name

Sunil Thorat

E-mail address

vistaemailservice@gmail.com

Jenkins 2.431

Skip and continue as admin

Save and Continue

Click on Start using Jenkins

Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

Start using Jenkins

Jenkins 2.431

This is Jenkins Dashboard.

Go to Manage Jenkins --> Nodes

Click Built-In Node

The screenshot shows the Jenkins 'Nodes' page. At the top left, there are navigation links: Dashboard > Manage Jenkins > Nodes. Below these are two tabs: 'Nodes' (selected) and 'Clouds'. On the left, there are dropdown menus for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). The main area is titled 'Nodes' and contains a table with one row. The table columns are 'S' (Status icon), 'Name' (Built-In Node, highlighted with a red box), 'Architecture' (Linux (amd64)), 'Data obtained' (Data obtained), and '57 min' (Last activity).

Configure ---> 0 and save

The screenshot shows the 'Configure' sub-page for the Built-In Node. The title bar says 'Jenkins' and 'Dashboard > Nodes > Built-In Node > Configure'. The main section is titled 'Make Zero and save' and contains a form with the following fields:

- 'Number of executors' input field set to '0' (highlighted with a red box)
- 'Labels' input field (empty)
- 'Usage' dropdown menu set to 'Use this node as much as possible'
- 'Node Properties' section with two checkboxes:
 - 'Disable deferred wipeout on this node' (unchecked)
 - 'Environment variables' (unchecked)
- A large blue 'Save' button at the bottom (highlighted with a red box)

Again go to Manage Jenkins ---> Nodes

Dashboard > Manage Jenkins

[+ New Item](#)

[People](#)

[Build History](#)

[Manage Jenkins](#)

[My Views](#)

Manage Jenkins

System Configuration

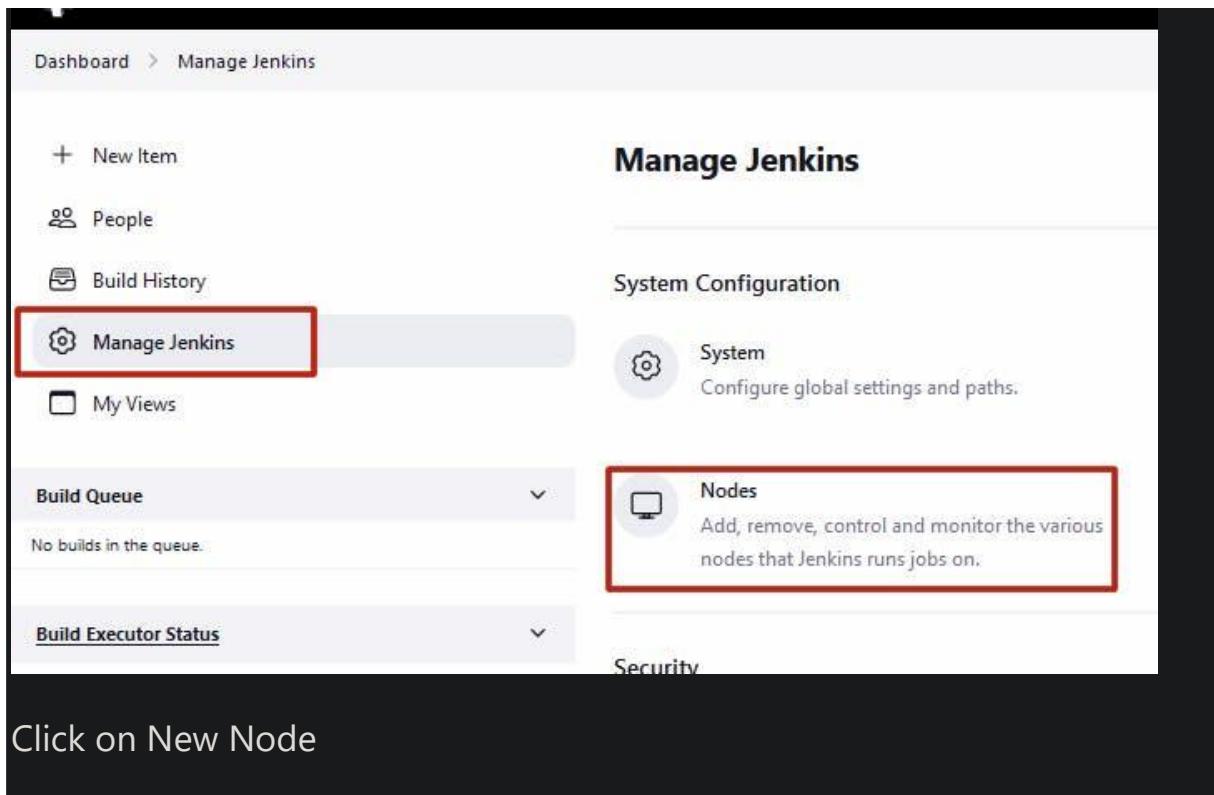
[System](#)
Configure global settings and paths.

Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Build Queue
No builds in the queue.

Build Executor Status

Security



Click on New Node

Nodes

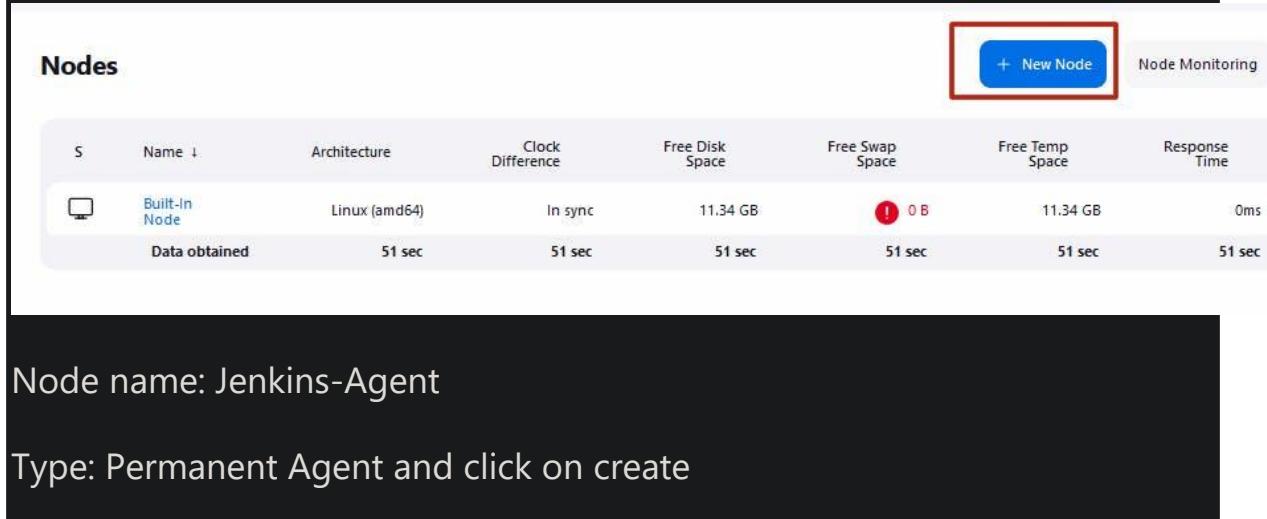
[+ New Node](#)

Node Monitoring

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node Data obtained	Linux (amd64)	In sync	11.34 GB	0 B	11.34 GB	0ms
		51 sec	51 sec	51 sec	51 sec	51 sec	51 sec

Node name: Jenkins-Agent

Type: Permanent Agent and click on create



New node

Node name

Jenkins-Agent

Type



Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Create

Number of executors: 2

Remote root directory: /home/ubuntu

Labels: Jenkins-Agent

Name ?

Jenkins-Agent

Description ?

Jenkins-Agent

Plain text [Preview](#)

Number of executors ?

2

Remote root directory ?

/home/ubuntu

Labels ?

Jenkins-Agent

Usage ?

[Save](#)

Host: Paste Private IP of Jenkins-Agent

Usage ?

Use this node as much as possible

Launch method ?

Launch agents via SSH

Host ?

Paste Private IP of Jenkins-Agent

172.31.43.223

Credentials ?

- none -

+ Add ▾

Jenkins Credentials cannot be found

Host Key Verification Strategy ?

Known hosts file Verification Strategy

Advanced ▾

Save

Click on Jenkins for adding credentials

Add Credentials

Domain

Global credentials (unrestricted)

Kind

SSH Username with private key

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

ID ?

Jenkins-Agent

Description ?

Jenkins-Agent

Username

Known hosts file Verification Strategy

Username

Treat username as secret ?

Private Key

Enter directly

Go to Jenkins-Master and copy private key and paste here

Key

No Stored Value

Add

Passphrase

Add Cancel

Copy This Private Key and Paste it in Key value

Quick connect...

ubuntu@Jenkins-Master:~/.ssh\$ cd
 ubuntu@Jenkins-Master:~\$ sudo cat authorized_keys
 852aab82e98f46c0b8bc32f75b56cb68
 ubuntu@Jenkins-Master:~\$ ls
 ubuntu@Jenkins-Master:~\$

Private Key

File Edit Format View Help

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZxktdjEAAAAABG5vbmuAAAAEBm9uZQAAAAAAAAAAAB1wAAAAdzc2gtcn
NhAAAAAAwEAAQAAAYEAufSbSoHAUy+o4dNiOOepDubaNSjtWUNTuqbVDA6Ynnlj5Rdch7T2
pyhHQx/lm7f4pSUQTG5mfaf8Z9YSVFeXkwaj7uvpTsosBNce1nFjp0+uODj0E114k1li1n6
Ts898PtuX3I79ptKP+GENPttcxQORRXOGYCSN+FdwRwk94v0M0wHg7gcPsQtcyQhauy
CJZ7iWaky01hgMzYPWGHT0YN6yshfIC+HShzzG/hbTk2t5+ehCOCISgjhkGCIVBQB8B6Qb
mgabucAMg+PebyUhM8cwQo0oVcapoeM7nyTong00567u1hVUV6j0XRKCVYyvP0a11r371Hi
bnjA/GUIJA3BA8His07FDb4oJL6o41iEiOHmgia7L0tQmbT/KA45fx7QyQvvvrMdM104U0vv
xPwBx88kHPSSad9n1uYRP5ue+H61xig8Mo12FVM6ccChvro00GYf00B6YMYKC8txv/sBNK3C
6HX2j6gEcue8MJ9R+yjXE4zLD9tLDEsx83LyBry1AAAFkAg0qsgINKrIAAAAB3NzaC1yc2
EAAAGBALn0m0qBwFMvq0HTYjjjnq7m2juiBCFDU7qm1QwOmJ55Y+UXXi09qcoR0Mf5Zu3
+KULExuZn2vGfWE1RX15MGo+7r6U7NLATXhtzsadPrjg49BJzeJNZytz+k7PPFd7b19y
O/absj/hhJz7bxF0DkUvzhmHEjfhxVka83PeL6DENMB404HD7EE3MkiamrsgiWe4lgJMg9
YRjM2D1hh09GdesrIXyAvh0oyC8xv4W05NrUvhI0qQaI0YI4ZBgiFQUAqekG5oGm7nADIPj
3m81ITPHMDqtKFXGqaHj058k6j4NN0eu7tYVVL+o9F0SnFwMDw2tda9+9R4m54wPx1CCQN
wQPB4rDuxQ2+KCS+qONyHijh5oIgOy9LUJm0/ygoEN12EL2L6zHTNDofNL78T8AcfPjBz0
mnfZ9VMkT+bnvh+tcYoPDKNdhVTOnAob66Djhmh9DgemDGcvLcb/7ATStwuhi9o+oBar
vDCfuFs01x0Myw/bSwxLMFnY8ga8tQAAAAMBAEAAAGANRvvSt8/HFECCdUSENvfvQGwZ
lQ9JU4LzXV5qm6Bq9Hyyv8NZ9CtCgq464U2TWU7OveMbtoVaOpso2us4Abo+8N4Wv5fdmA
/y3C6dxq6ny5jYn+5bndCsBug+WNu3r6f205a0MS21vKvbuZCsajFjSpz56/5CT2BY1L
usJ5v56VkoYgEqSv61QL7FP07MtWfj3y0LgwAGmljLvoX7XvV0HriCyCxMwSOAfGnKrywb
Ep8dnUGU/1BKnvXkw4TqRUFMR0dYgwEp6Y8RrB0M0+c5rNLmhC8o1400K0bwFkr4p0W+or
m/Axm70UB1x519Ef0wMqs30+mYXEZ0K9VJdtqlmEh2aVQc5RDqlf3bezyt06FtszVm38e
mfwpjjg9B0x8RQ04N3jPhHCi03SzFmjFkqKDPe0Wro9t8DjAn3p5IV3Icr6yMmfttRSAd
po687Wpq0RZLL+wt+yVpxpQ7WOpOppof4P7MI7ZJK3MUWgm3AGs18HG92x5fnKwhXAAAA
whdV2/5gdVsadi1htaAb3zqC8qa/ad3uDjeXhh3MooRuZpY4k3eyfkVviFjxK5nAswOX
Umplx45C1pNR1yYEwSpHCzghGjFlonBuWLIVyZLBhdzTpdg08zVJknLqTwDbmIKbMw+k
shwX4VVbhr0QoxmW6nIXWt1BwEXOYmzyFG2UoZTAxgTGcpDacMB16MsVRQ1zpwsVG2/r5
MDgsGGi81Tj7G9K0uqkakmadf6V3kw2D8Jp6ltxyKlwneK7FQAAAMEAzWF2+hLcUsQRfhNw
BGiybpYTAlk4Md9EwnoZYIxcdvtr8dbiGh4TMBd67wD+XYTgv0xvlqeF0169u24HHbj6bz
xc85T4T8RYQb8ABFRpDwtq77Lvvbj7VI4foINQ7PY6oXKuDuCnJmYDQht5Hf00w0zBKGz
We349N7C+b0eBZccBwneY4RtIltDXOKIor82oY70LtxrQC8iELn/ecR84zbHLarKc5399i
JVqeUdrX4YsaqEAhb8dx1iF4t/9pDAAAaWQDnyYDk++bHnP2pu9tc50BqhKVPfkMXZgil
QCUDinuzlyj7L9rjrbxv+hpsT1wCNYYYOoceUP/frvzdd4HfaXXtQ3Epn7eRYSdrRumDgL
b8RlUgrSSVMNH5h2HHBUZ3GTCqwd58PsavrBigi0mBAZKVDwbdjyi0AgX44dTVX0nKIE8d
dg0z7yNol/+Y14Ur1V21CMdk1xnBfWaD1GTeB4CDNQmAjN+7m6nwsRUEaYAX2K/k+SjY
QfwxypYxERCaAAAAVdWJ1bnR1QEplbmtpbnMtTWFzdGVyAQIDBAUG
-----END OPENSSH PRIVATE KEY-----
```

Treat username as secret ?

Private Key

Enter directly

Key

Paste here private key and click add Enter New Secret Below

```
-----BEGIN OPENSSH PRIVATE KEY-----  
b3B1bnNzaC1rZXKtdjEAAAABG5vbmuAAAAEbm9UZQAAAAAAAAAAQAB1wAAAAdzc2gtcn  
NzAAMAAAWEAA0AAAYEAufsb5oAUw+04dNiOoeDuBaNSJtwUNTugbVDA6Yn1j5Rdch7T2  
DvH0X/1m7f4psUQTG5mfAbZ39YSVFeXkwaj7uvptsesBNceinfJp0+uODj@El14k1l1n6  
-----END OPENSSH PRIVATE KEY-----
```

Passphrase

Add Cancel

Dashboard > Manage Jenkins > Nodes >

Host ?
172.31.43.223

Credentials ?
jenkins (Jenkins-Agent)
+ Add ▾

Host Key Verification Strategy ?
Non verifying Verification Strategy

Advanced ▾

Availability ?
Keep this agent online as much as possible

Node Properties

Disable deferred wipeout on this node ?
 Environment variables

Save

Nodes

[+ New Node](#)

Node Monitoring

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	Built-in Node	Linux (amd64)	In sync	11.34 GB	! 0 B	11.34 GB	0ms
2	Jenkins-Agent		N/A	N/A	N/A	N/A	N/A

Now go to Jenkins Dashboard and Click on Create a Job

Enter job name: Test--> Click on Pipeline and click on ok

Enter an item name

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

Go to Pipeline-->Pipeline Script-->Hello World

Dashboard > Test > Configuration

Configure

Pipeline

Definition

Pipeline script

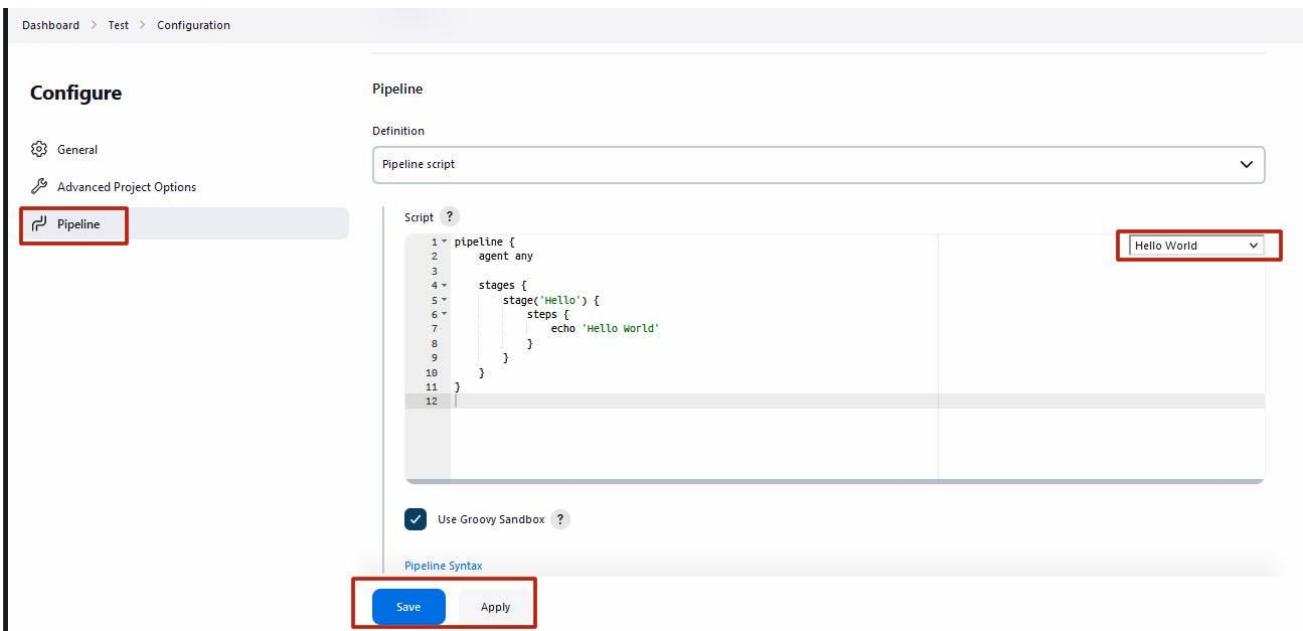
```
1 pipeline {  
2   agent any  
3  
4   stages {  
5     stage('Hello') {  
6       steps {  
7         echo 'Hello World'  
8       }  
9     }  
10  }  
11 }  
12 }
```

Hello World

Use Groovy Sandbox

Save Apply

Pipeline Syntax



Click on Build now and see test is successful.

Dashboard > Test >

Status Test

</> Changes

▷ Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

Build History trend

Filter builds... /

#2 Nov 11 21:24 No Changes

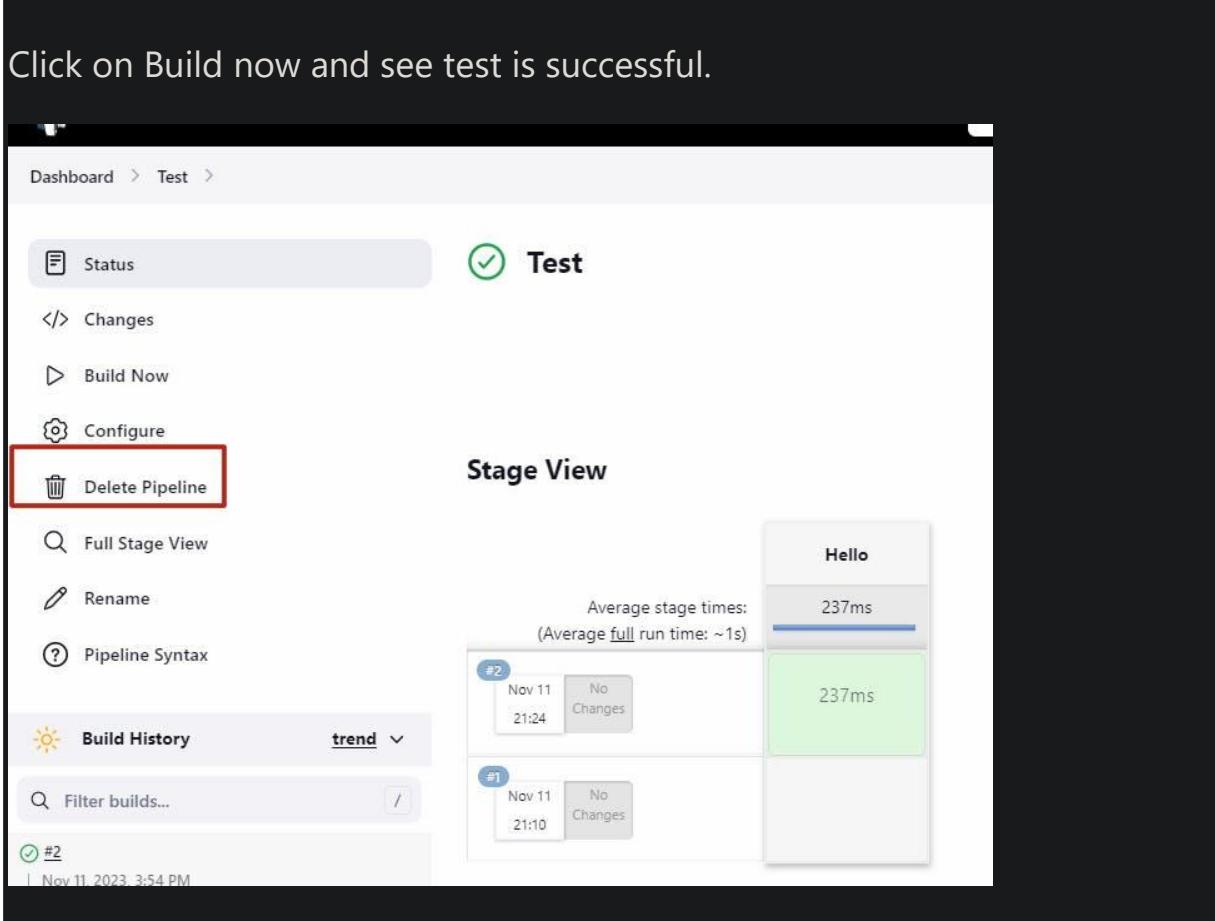
#1 Nov 11 21:10 No Changes

Average stage times:
(Average full run time: ~1s)

Stage View

Hello 237ms

237ms



The screenshot shows the Jenkins interface for a build named 'Test #2'. The 'Console Output' tab is selected, displaying the following log:

```
Started by user Sunil Thorat
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins-Agent in /home/ubuntu/workspace/Test
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Integrate Maven to Jenkins and Add GitHub Credentials to Jenkins

Go to Manage Jenkins-->Plugins

The screenshot shows the 'Manage Jenkins' section of the Jenkins dashboard under 'Manage Jenkins'. The 'Plugins' option is highlighted with a red box.

Manage Jenkins

System Configuration

- System: Configure global settings and paths.
- Tools: Configure tools, their locations and automatic installers.
- Nodes: Add, remove, control and monitor the various nodes that
- Clouds: Add, remove, and configure cloud instances to provision

Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Available Plugins>maven

Dashboard > Manage Jenkins > Plugins

Plugins

Updates

Available plugins (red box)

Installed plugins

Advanced settings

Download progress

Search: maven

Install Name ↴ Released

	Name	Version	Category	Description	Released
<input checked="" type="checkbox"/>	Maven Integration	3.23	Build Tools	This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.	3 mo 8 days ago
<input type="checkbox"/>	Config File Provider	959.vcff671a_4518b_	Groovy-related External Site/Tool Integrations Maven	Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace.	1 mo 24 days ago
<input type="checkbox"/>	Jira	3.11	External Site/Tool Integrations Maven jira	This plugin integrates Jenkins to Atlassian Jira.	2 mo 0 days ago
<input checked="" type="checkbox"/>	Pipeline Maven Integration	1362.vee39a_d4b_02b_1	pipeline Maven	This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path.	

File, Copy, Delete

Available Plugins>eclipse install them

Dashboard > Manage Jenkins > Plugins

Plugins

Updates

Available plugins (red box)

Installed plugins

Advanced settings

Download progress

Search: eclipse

Install Name ↴ Released

	Name	Version	Category	Description	Released
<input checked="" type="checkbox"/>	Maven Integration	3.23	Build Tools	This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.	3 mo 8 days ago
<input checked="" type="checkbox"/>	Pipeline Maven Integration	1362.vee39a_d4b_02b_1	pipeline Maven	This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path.	8 hr 24 min ago
<input type="checkbox"/>	Jersey 2 API	2.41-133.va_03323b_a_1396	Library plugins (for use by other plugins)	This plugin provides the JAX-RS 2.1 and Jersey 2 APIs for other plugins.	18 days ago
<input checked="" type="checkbox"/>	Eclipse Temurin installer	1.5		Provides an installer for the JDK tool that downloads the JDK from https://adoptium.net	1 yr 1 mo ago
<input type="checkbox"/>	Buckminster	1.1.1			

Manage Jenkins--> Tools

Dashboard > Manage Jenkins

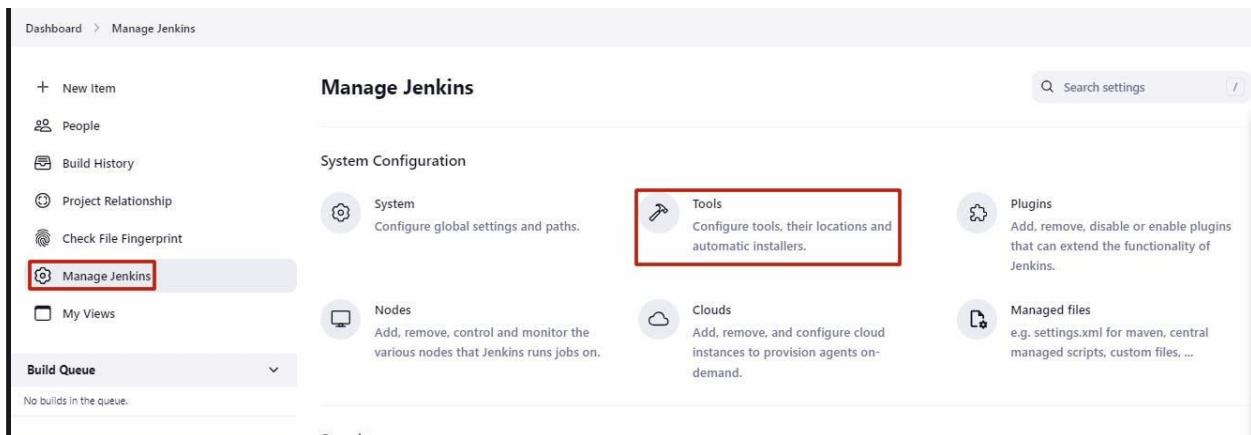
Manage Jenkins

System Configuration

- System: Configure global settings and paths.
- Tools**: Configure tools, their locations and automatic installers. **Tools** is highlighted with a red box.
- Nodes: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Clouds: Add, remove, and configure cloud instances to provision agents on-demand.

Build Queue: No builds in the queue.

Security:



Maven Installations-->Add Maven--> Maven3

Dashboard > Manage Jenkins > Tools

Add Ant

Maven installations

Add Maven

Maven

Name: **Maven3** (highlighted with a red box)

Install automatically ?

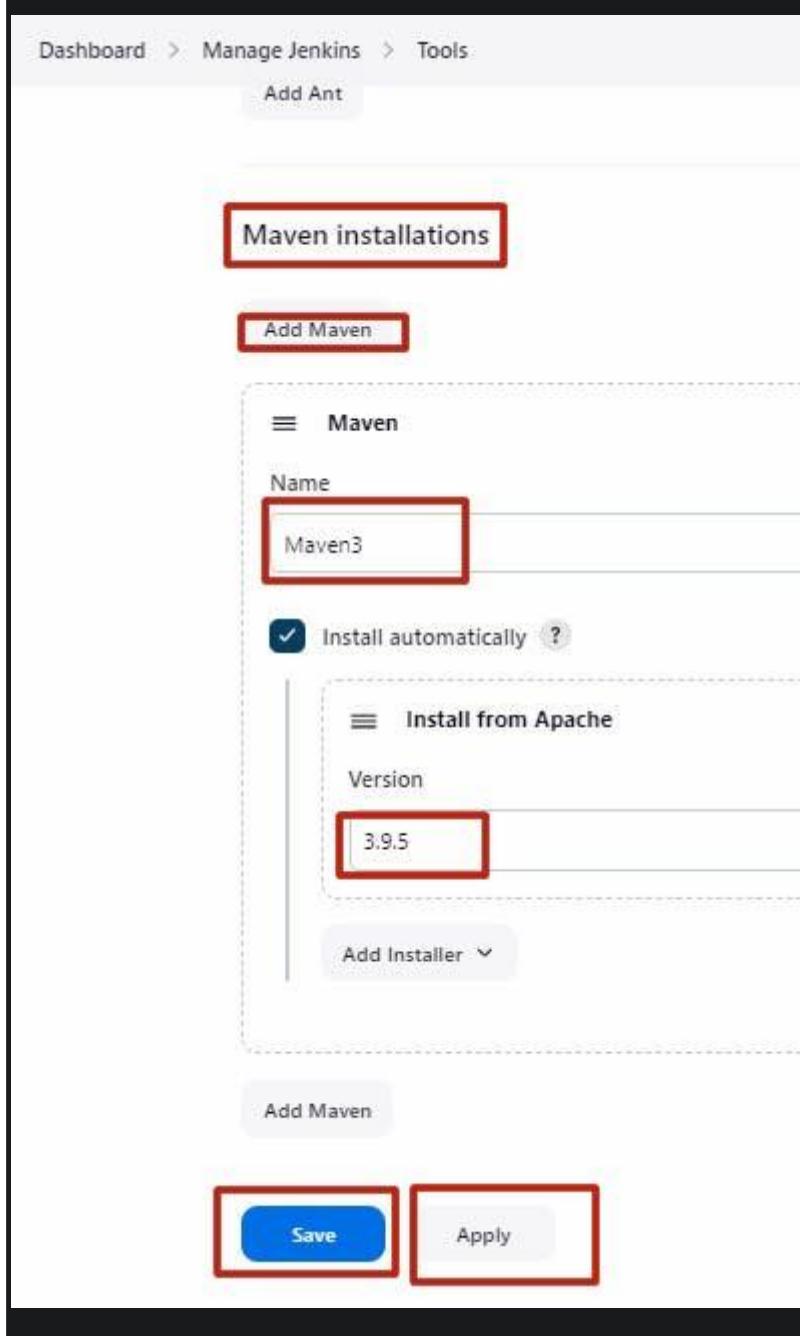
Install from Apache

Version: **3.9.5** (highlighted with a red box)

Add Installer ▾

Add Maven

Save **Apply**



JDK installations -->java17--> install from [adoptium.net](#)

The screenshot shows the Jenkins 'Tools' configuration page. At the top, there are two main sections: 'JDK installations' and 'Add JDK'. Below these, under the 'JDK' section, there is a form for defining a new JDK named 'Java17'. A checkbox labeled 'Install automatically' is checked and highlighted with a red box. A dropdown menu titled 'Add Installer' is open, showing several options: 'Extract *.zip/*.tar.gz', 'Install from adoptium.net' (which is also highlighted with a red box), 'Run Batch Command', and 'Run Shell Command'. Below this, there is another section titled 'Git' with a 'Name' field containing 'Default'.

Version -- jdk-17.0.5+8

JDK installations

Add JDK

JDK

Name

Java17

Install automatically ?

Install from adoptium.net ?

VERSION ?

jdk-17.0.5+8

Add Installer ▾

Add JDK

Git installations

Save

Apply



Manage Jenkins-->Credentials

The screenshot shows the Jenkins Manage Jenkins dashboard. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins' (which is highlighted with a red box), and 'My Views'. Below the sidebar are sections for 'Build Queue' (empty) and 'Build Executor Status' (Jenkins-Agent, 1 idle, 2 idle). The main area is titled 'System Configuration' and contains links for 'Tools', 'Nodes', 'Clouds', 'Plugins', 'Managed files', and 'Credential Providers'. Under 'Security', there are links for 'Security', 'Users', and 'Credentials' (which is also highlighted with a red box). A search bar at the top right says 'Search settings'.

Add Credentials

This screenshot shows the 'Add Credentials' page. It has a header 'Credentials' and a table with columns: T, P, Store, Domain, ID, and Name. One row is visible: a fingerprint icon, a person icon, 'System', '(global)', 'Jenkins-Agent', and 'ubuntu (Jenkins-Agent)'. Below this is a section titled 'Stores scoped to Jenkins' with a table showing 'Domains' and a 'Store' dropdown set to 'System'. At the bottom of this section is a red-bordered 'Add credentials' button.

Kind: Username with password

GitHub username: thoratsunil

For password we want to create token

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind Username with password

Scope

Username thoratsunil

Treat username as secret

Password

ID

Description

Create

Go to GitHub account setting

Developer settings

Personal access tokens --> Token (classic)

Settings / Developer Settings

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens Beta

Tokens (classic) Beta

GitHub Apps

Want to build something that integrates with GitHub? You can also read more about GitHub Apps.

Generate new token (classic)



It will ask password so give password and confirm

name for Note: gitHub and generate token

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

gitHub

What's this token for?

read:ssh_signing_key

Read public user SSH signing keys

Generate token

Cancel

Copy this token

And paste it in Password section and give id name gitHub

New credentials

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

thoratsunil

 Treat username as secret ?

Password ?

ID ?

gitHub

Description ?

gitHub

Create

so github credentials are created

+ Add Credentials

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description	
 Jenkins-Agent	ubuntu (Jenkins-Agent)	SSH Username with private key	Jenkins-Agent	
 gitHub	thoratsunil/***** (gitHub)	Username with password	gitHub	

Icons S M L

Create Pipeline Script(Jenkinsfile) for Build & Test Artifacts and Create CI Job on Jenkins

Go to GitHub repositories --> CI-CD_mark1

and create Jenkinsfile in master or main branch

Find a repository...

CI-CD_mark1 Public

its an registration app which contains end to end CICD process using various tools

Java Updated 24 minutes ago

CI-CD_mark1_manifest Public

this file contains the manifest file for argo cd

Updated 36 minutes ago

Sri Yeshwanth Ekavasalu
SriYeshwanthE · he/him

Start writing groovy syntax with the help of the Hello World script

Save file with commit changes

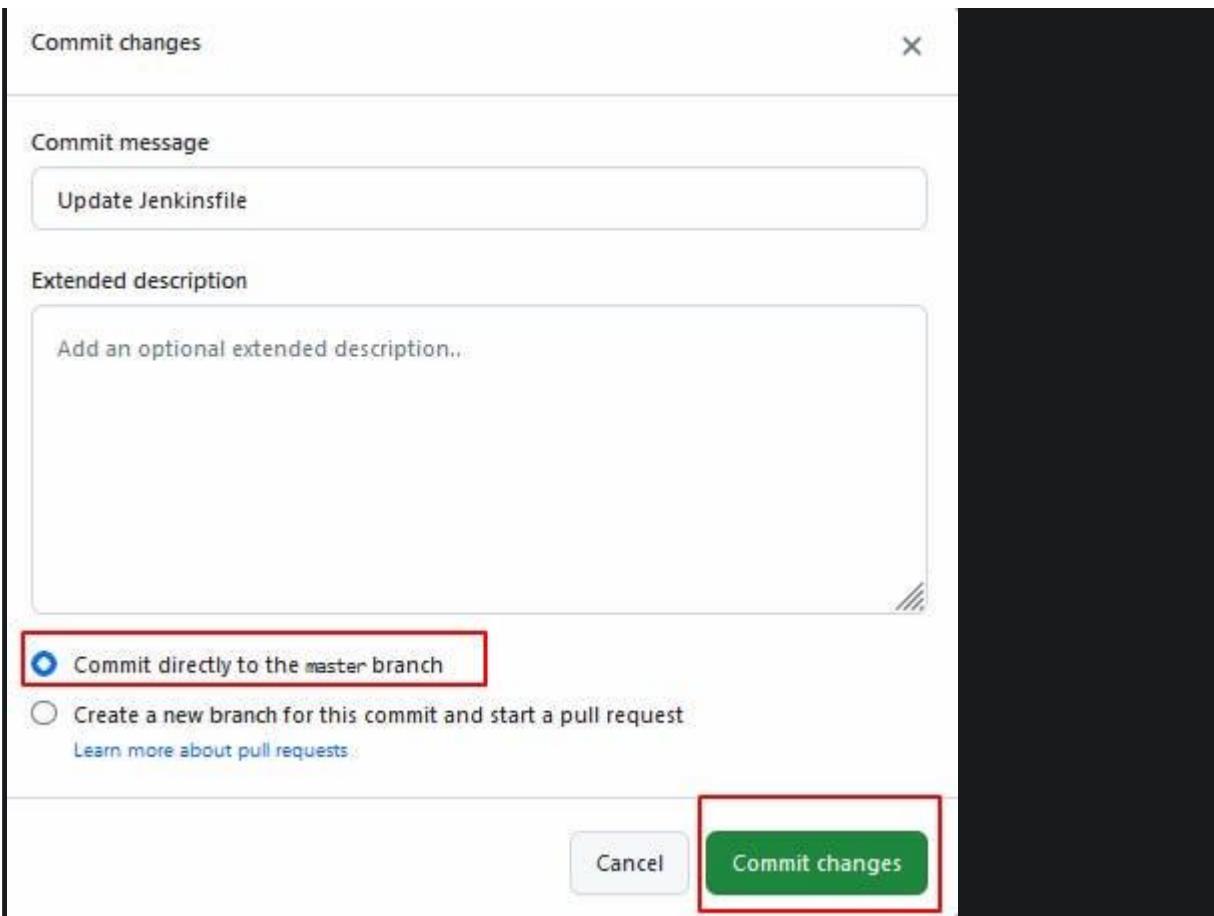
Commit changes X

Commit message
Update Jenkinsfile

Extended description
Add an optional extended description...

Commit directly to the master branch
 Create a new branch for this commit and start a pull request
[Learn more about pull requests](#)

Cancel Commit changes



Go to Jenkins Dashboard and create new job

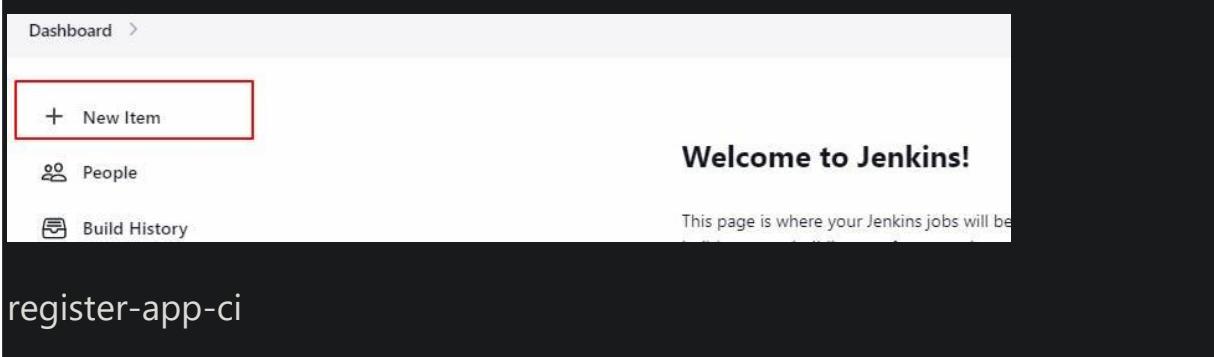
Dashboard >

+ New Item Welcome to Jenkins!

People This page is where your Jenkins jobs will be

Build History

register-app-ci



Enter an item name

register-app-ci

» Required field



Freestyle project

This is the central feature of Jenkins. Jenkins will build your project for something other than software build.



Maven project

Build a maven project. Jenkins takes advantage of your POM file to automatically detect dependencies and build your project.



Pipeline

Orchestrates long-running activities that can span multiple builds and/or organizing complex activities that do not easily fit in free-style projects.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as building multiple variants of the same codebase or multiple builds, etc.

OK

In the pipeline option select Pipeline script from SCM --> Git --> Paste Repository URL --> Select Credentials and click apply save.

Dashboard > register-app-ci > Configuration

Configure

Pipeline

Definition
Pipeline script from SCM

SCM ?
Git

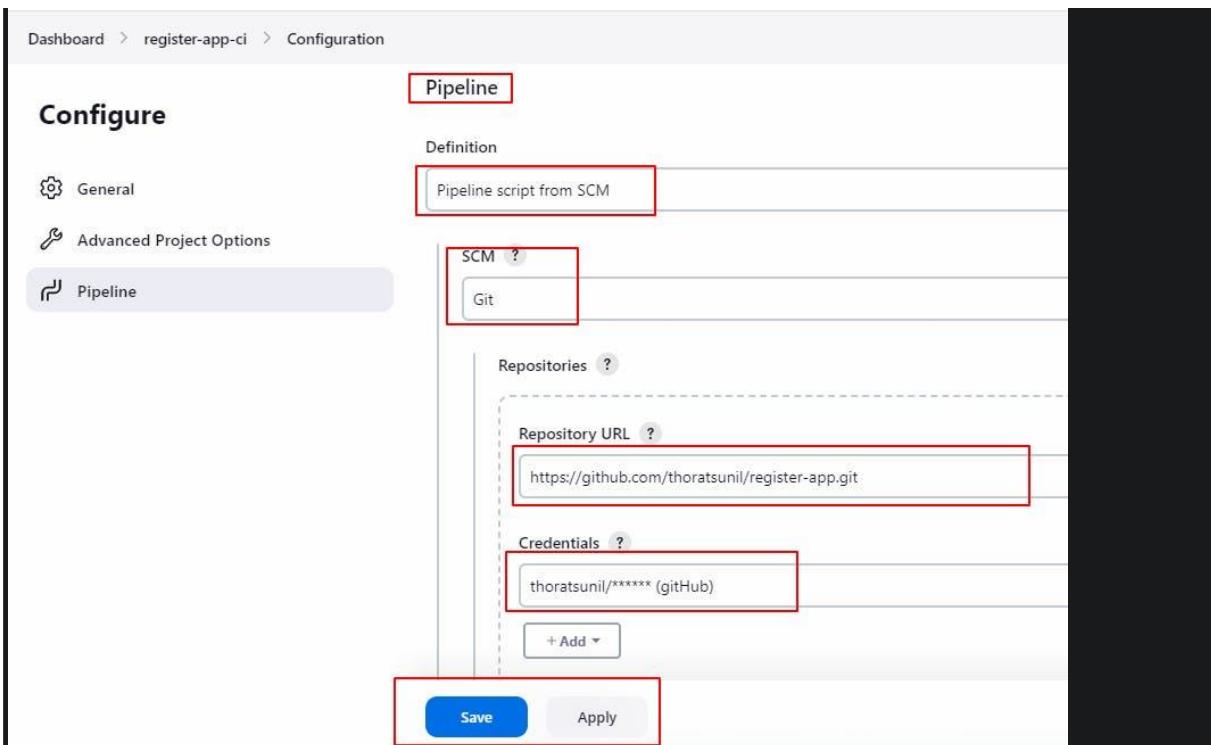
Repositories ?

Repository URL ?
`https://github.com/thoratsunil/register-app.git`

Credentials ?
`thoratsunil/******** (GitHub)`

+ Add ▾

Save **Apply**



This screenshot shows the Jenkins Pipeline configuration page. The 'Pipeline' tab is selected. Under 'Definition', 'Pipeline script from SCM' is chosen. The 'SCM' dropdown is set to 'Git'. In the 'Repositories' section, the 'Repository URL' is set to 'https://github.com/thoratsunil/register-app.git' and the 'Credentials' are set to 'thoratsunil/******** (GitHub)'. There is a '+ Add ▾' button for adding more repositories. At the bottom are 'Save' and 'Apply' buttons.

Select branch main or master

Script Path: Jenkinsfile

Dashboard > register-app-ci > Configuration

Configure

General

Advanced Project Options

Pipeline

Branches to build ?

Branch Specifier (blank for 'any') ?
`*/master`

Add Branch

Repository browser ?
(Auto)

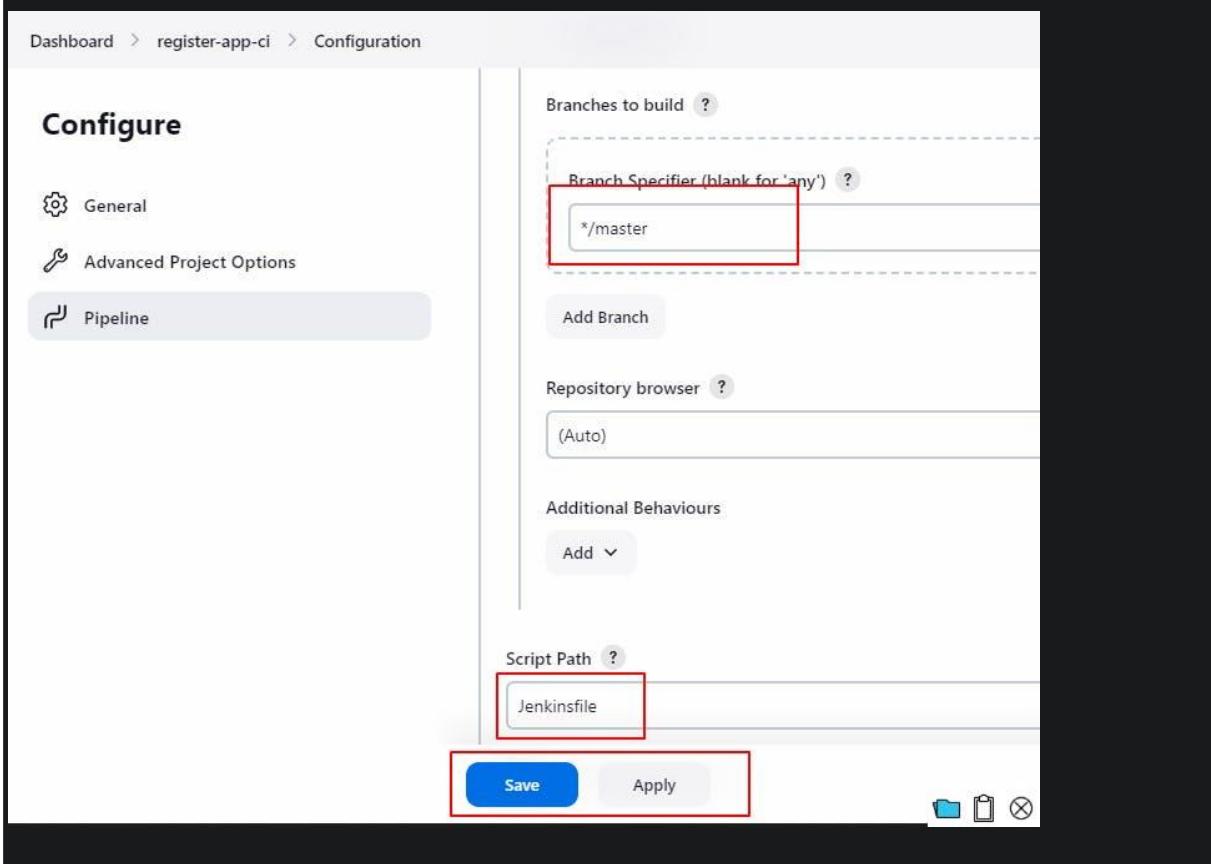
Additional Behaviours

Add ▾

Script Path ?
`Jenkinsfile`

Save **Apply**

File Edit Close



This screenshot shows the Jenkins Pipeline configuration page. The 'Pipeline' tab is selected. Under 'Branches to build', the 'Branch Specifier' is set to '*/master'. There is an 'Add Branch' button. Under 'Repository browser', it is set to '(Auto)'. Under 'Additional Behaviours', there is an 'Add' button. In the 'Script Path' section, the path is set to 'Jenkinsfile'. At the bottom are 'Save' and 'Apply' buttons, along with standard file operations icons.

Now click on Build Now



Jenkins

Dashboard >

+ New Item Build History All + Add description

Project Relationship Check File Fingerprint Manage Jenkins My Views

Build Queue: No builds in the queue. Icon: S M L

Build Executor Status: 0 idle, 0 used, 02 available

Pipeline: A horizontal flowchart showing the stages of a pipeline: Start, Checkout SCM, Tool Install, Cleanup Worksp..., Checkout from ..., Build Application, Test Application, SonarQube Anal..., Build & Push Do..., Trivy Scan, Cleanup Artifacts, Trigger CD Pipe..., Post Actions, End.

Dashboard > register-app-ci > #1

```
HTTP://.../job/register-app-ci/1/console
```

Downloading from central: <https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.11/surefire-junit4-2.11.jar>
Progress (1): 16/33 kB
Progress (1): 33/33 kB
Progress (1): 33 kB

Downloaded from central: <https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire-junit4/2.11/surefire-junit4-2.11.jar> (33 kB at 5.5 MB/s)

TESTS

Running com.example.TestGreeter
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.094 sec

Results :
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

```
[INFO]  
[INFO] --- jar:3.3.0:jar (default-jar) @ server ---  
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/file-management/3.1.0/file-management-3.1.0.pom  
Progress (1): 4.5 kB
```

```
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/file-management/3.1.0/file-management-3.1.0.pom (4.5 kB at 499 kB/s)
```

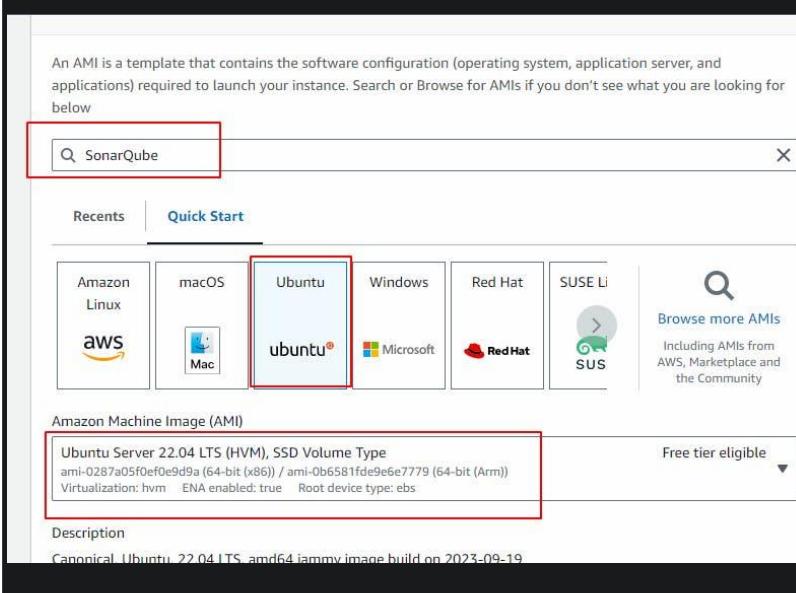
```

[INFO] -----
[INFO] Reactor Summary for Maven Project 1.0-SNAPSHOT:
[INFO]
[INFO] Maven Project ..... SUCCESS [ 0.004 s]
[INFO] Server ..... SUCCESS [ 2.096 s]
[INFO] Webapp ..... SUCCESS [ 0.341 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  2.728 s
[INFO] Finished at: 2023-11-11T19:52:46Z
[INFO] -----
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Install and Configure the SonarQube

For install SonarQube Create new EC2 instance : SonarQube -- Ubuntu --



Instance Type: t3.medium

▼ Instance type [Info](#)

Instance type

t3.medium

Family: t3 2 vCPU 4 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.0632 USD per Hour

On-Demand SUSE base pricing: 0.1011 USD per Hour

On-Demand RHEL base pricing: 0.1048 USD per Hour

On-Demand Linux base pricing: 0.0448 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Linux-VM-key

Create new key pair

Volume 15GiB and click on launch instance

▼ Configure storage [Info](#)

Advanced

1x GiB Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage [X](#)

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

0 x File systems [Edit](#)

► Advanced details [Info](#)

1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0287a05f0ef0e9d9a

Virtual server type (instance type)
t3.medium

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 15 GiB

Cancel [Launch instance](#) Review commands

Select SonarQube instance and copy public IP

The screenshot shows the AWS CloudWatch Instances interface. In the 'Instances (1/3)' list, the 'SonarQube' instance is selected and highlighted with a red box. The instance details page below shows the following information:

Name	Instance ID	Instance state	Instance type	Status
Jenkins-Master	i-01c22280...	Running	t2.micro	2/2
Jenkins-Agent	i-016dbbc8...	Running	t2.micro	2/2
SonarQube	i-0e22c3b6...	Running	t3.medium	2/2

Below the table, the 'Public IPv4 address' field is highlighted with a red box and contains the value 13.127.135.59. Other fields shown include 'IPv6 address' (empty), 'Instance state' (Running), and 'Status checks'.

Connect SonarQube with the help of Mobaxterm

The screenshot shows the 'Session settings' dialog of the Mobaxterm application. The 'SSH' tab is selected. The 'Basic SSH settings' section contains the following fields:

- Remote host: 13.127.135.59 (highlighted with a red box)
- Specify username: ubuntu (highlighted with a red box)
- Port: 22

Below this, the 'Advanced SSH settings' tab is selected. It includes the following options:

- X11-Forwarding (checked)
- Compression (checked)
- Remote environment: Interactive shell
- Execute command: (empty)
- Do not exit after command ends (unchecked)
- SSH-browser type: SFTP protocol
- Follow SSH path (experimental) (unchecked)
- Use private key (checked) - C:\Users\SUNIL\Downloads\Linux- (highlighted with a red box)
- Expert SSH settings (button)

At the bottom of the dialog are 'OK' and 'Cancel' buttons, with 'OK' highlighted with a red box.

Update the server `sudo apt update`

The screenshot shows a terminal window with three tabs at the top:

- 2. 15.207.84.111 (ubuntu)
- 3. 13.233.208.255 (ubuntu)
- 4. 13.127.135.59 (ubuntu)

The active tab (tab 4) shows the command `sudo apt update` being typed into the terminal. The output of the command is visible below the input line.

At the bottom of the terminal window, the command `sudo apt upgrade -y` is visible.

```
ubuntu@ip-172-31-33-125:~$ sudo apt upgrade -y
```

Now install PostgreSQL with the help of following commands

```
===== Install and Configure the SonarQube =====
## Update Package Repository and Upgrade Packages
$ sudo apt update
$ sudo apt upgrade
## Add PostgreSQL repository
$ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
$ wget -qO- https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo tee /etc/apt/trusted.gpg.d/pgdg.asc &>/dev/null
## Install PostgreSQL
$ sudo apt update
$ sudo apt-get -y install postgresql postgresql-contrib
$ sudo systemctl enable postgresql
## Create Database for SonarQube
$ sudo passwd postgres
$ su - postgres
$ createuser sonar
$ psql
$ ALTER USER sonar WITH ENCRYPTED password 'sonar';
$ CREATE DATABASE sonarqube OWNER sonar;
$ grant all privileges on DATABASE sonarqube to sonar;
$ \q
$ exit
## Add Adoptium repository
$ sudo bash
$ wget -O - https://packages.adoptium.net/artifactory/api/gpg/public | tee /etc/apt/keyrings/adoptium.asc
$ echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packages.adoptium.net/artifactory/deb $(awk -F= '/^VERSION_CODENAME/{print$2}' /etc/os-release) main" | tee
```

```
ubuntu@ip-172-31-33-125:~$ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
```

```
ubuntu@ip-172-31-33-125:~$ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
ubuntu@ip-172-31-33-125:~$ 
ubuntu@ip-172-31-33-125:~$ wget -qO- https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo tee /etc/apt/trusted.gpg.d/pgdg.asc &>/dev/null
ubuntu@ip-172-31-33-125:~$ 
ubuntu@ip-172-31-33-125:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://apt.postgresql.org/pub/repos/apt jammy-pgDG InRelease [123 kB]
Get:6 http://apt.postgresql.org/pub/repos/apt jammy-pgDG/main amd64 Packages [296 kB]
Fetched 529 kB in 2s (258 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
ubuntu@ip-172-31-33-125:~$ 
```

Add PostgresSQL repository

```
$ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list' $ wget -
```

```
q0- https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo tee  
/etc/apt/trusted.gpg.d/pgdg.asc &>/dev/null
```

```
2. 15.207.84.111 (ubuntu) 3. 13.233.208.255 (ubuntu) 4. 13.127.135.59 (ubuntu) × +  
ubuntu@ip-172-31-33-125:~$ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'  
ubuntu@ip-172-31-33-125:~$  
ubuntu@ip-172-31-33-125:~$ wget -qO- https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo tee /etc/apt/trusted.gpg.d/pgdg.asc &>/dev/null  
ubuntu@ip-172-31-33-125:~$  
ubuntu@ip-172-31-33-125:~$ sudo apt update  
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Get:5 http://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease [123 kB]  
Get:6 http://apt.postgresql.org/pub/repos/apt jammy-pgdg/main amd64 Packages [296 kB]  
Fetched 529 kB in 2s (258 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
All packages are up to date.  
ubuntu@ip-172-31-33-125:~$
```

Install PostgreSQL

```
$ sudo apt update  
  
$ sudo apt-get -y install postgresql postgresql-contrib  
  
$ sudo systemctl enable postgresql
```

```
2. 15.207.84.111 (ubuntu) 3. 13.233.208.255 (ubuntu) 4. 13.127.135.59 (ubuntu) × +  
ubuntu@ip-172-31-33-125:~$ sudo passwd postgres  
New password: _____  
Retype new password: _____  
passwd: password updated successfully  
ubuntu@ip-172-31-33-125:~$ su - postgres  
Password: _____  
postgres@ip-172-31-33-125:~$  
postgres@ip-172-31-33-125:~$ createuser sonar  
postgres@ip-172-31-33-125:~$
```

Create a Database for Sonarqube

```
$ sudo passwd postgres  
  
$ su - postgres  
  
$ createuser sonar  
  
$ psql
```

```
$ ALTER USER sonar WITH ENCRYPTED password 'sonar';

$ CREATE DATABASE sonarqube OWNER sonar;

$ grant all privileges on DATABASE sonarqube to sonar;

$ \q $ exit
```

```
ubuntu@ip-172-31-33-125:~$ sudo passwd postgres
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ip-172-31-33-125:~$ su - postgres
Password:
postgres@ip-172-31-33-125:~
postgres@ip-172-31-33-125:~$ createuser sonar
postgres@ip-172-31-33-125:~$ plsql
Command 'plsql' not found, did you mean:
  command 'psql' from deb postgresql-client-common (238)
Try: apt install <deb name>
postgres@ip-172-31-33-125:~$ psql
psql (16.1 (Ubuntu 16.1-1.pgdg22.04+1))
Type "help" for help.

postgres=# ALTER USER sonar WITH ENCRYPTED password 'sonar';
ALTER ROLE
postgres=# CREATE DATABASE sonarqube OWNER sonar;
CREATE DATABASE
postgres=# grant all privileges on DATABASE sonarqube to sonar;
GRANT
postgres=# \q
postgres@ip-172-31-33-125:~$ exit
logout
ubuntu@ip-172-31-33-125:~$
```

Add Adoptium repository

```
$ sudo bash $ wget -O
- https://packages.adoptium.net/artifactory/api/gpg/key/public | tee
/etc/apt/keyrings/adoptium.asc

$ echo "deb [signed-
by=/etc/apt/keyrings/adoptium.asc] https://packages.adoptium.net/artifactory/deb $(
awk -F= '/^VERSION_CODENAME/{print$2}' /etc/os-release) main" | tee
/etc/apt/sources.list.d/adoptium.list
```

```

ubuntu@ip-172-31-33-125:~$ sudo bash
root@ip-172-31-33-125:/home/ubuntu# wget -O - https://packages.adoptium.net/artifactory/api/gpg/key/public | tee /etc/apt/keyrings/adoptium.asc
--2023-11-11 20:29:06 -- https://packages.adoptium.net/artifactory/api/gpg/key/public
Resolving packages.adoptium.net (packages.adoptium.net)... 151.101.155.42
Connecting to packages.adoptium.net (packages.adoptium.net)|151.101.155.42|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1793 (1.8K) [text/plain]
Saving to: 'STDOUT'

[100%[=====] 1.75K --.-KB/s   in 0s]

-----BEGIN PGP PUBLIC KEY BLOCK-----
m0ENBGGT vT0BCAC6ey144n7CG8faofF6mwgIBN1fIm1ILZDuGS4tMr9/XI8pgJnT
QvspxZWEvtSm/bEM0bzOeZjcxWjbCj1l80u8k5kHMTI75gmZPs0KLFWIEpuRQ0
Pb0cws80ap0mLnNDUv0vDftuas5gna/fRw9y5mb0xBqvgrjFuIdGyW0vH5+a
90YlWD9n5V0gnMw-aclwzB/jzW3kHGSGzuMtlAheQah7Y8yom0n/UX8yqf
+11nP3+c87YcjkRqImRTtmKEDEtGAIXC6SY-aUEkbyE0fy0chkvtvWJ597fa
Epa4rnICU8zoJ6X5z3v1aM2WerhX9oq9x8PAABEAAg00Efkb3B0xVtIEdQryBL
ZKkgERFOi9sUE0g1UlnbmluZyBLZxpIDx0Zw1lcmuLwRldkBlY2xpcHNllm9y
Z26JA VIEEwEIADwWIQ078NdtyUNml00PzmEPElZfjwSvUCYZ09NAibAvULCQgH
AgM4agGFQoJCAsCBBYCAwEcHgcCF4AACgkQhDxipX48E4AggAj jZyWuKV3nG
7ngIngl8c/m9J0H7BmwgcQXYhyd5hVkmCmUx5JLexz2LMBUH/F2nD59ShgjMabk
kvib20x8lq9RsNdfc2hBcwU6ayHksIxq74boI2/XdyEzzMyzZWNGo/27c7Xmj
pWu31n0pdpqdwyD1kojbVvnxlCRY8as8Sm+1ufi709Kc14MuwhNs1lCSwb/fju
NkeHkrhLChKUUIEcmtSKRWrphYBzm1HYGbz4xpuElwfuUp7iehfoyBZlP6DRf
L5TY1fMcyHuvjNhrigVv7b0Tcf8y0Gy-TEUhzc4x0qcfr4ur9dopvsuPB0sv+
Klqf5KSZgrk80Rhk700AqAq14okly8cFrpyVneEOPb75AUZfkRpMd1R61xAj
SKch7as0f29AubuEBVpzsYt5sxePe14TdBf=m9F4y2Ec0LLa3ArLTj5H8FRl
UdgZB915mk4Gpt0zPM+ahMMu92vblZvju58Dv0i0Sp+cuM1Eq0MJS7e/4BM71z
E+OkavJc179pzhG3SK/IC/0lxxyETT66NSfJd7sW5R6Vr19am/uNU690W0CJ+q
V0efpmDm7LnfdrRIh+lle05+pvhXe1dGjox5cbG52vf8arIR/FgkfcFvqRMN1f
B+dVWueloUeVAnz2znoKnUEs7L90bJhYHggup47AU2wARAQAb10E2B8g8CAAq
F1EE0wTxU8kFDZpdND8shDxIpWx48EsFamGTVtOCGwAcgkQhdxipWx48EvXHqf/
00hZsGDXn2HTb0oje5dpk07WbjMIP3w1G0LVRp0UR8Tf0Pbz0oevZCNh38YgwF
yel3spvzD0rbXhgkzAGlucyg87Kha5Eb7m7DgMz37l1hYSZYCwd7aowfgy34
h0k3B67ffkJpIh7380a9Ctlwx09cytmBm01fbB0wm/9iHawhP0uydyIs40xWbj
0MGSP4fdntU7e4UjsNmhuhd0cy0lFaqdHIIIB9C/G4CzetRwHFOn3b4JwXMU7YU
6aAmXh3hggMC3wkT2HZ/Tquu0dNc02fypW0CD0Hz0alBBJNqoVUNFNq0U3tfj
wI4qf/KKq9BfyfucAs0ykA==

-----END PGP PUBLIC KEY BLOCK-----
2023-11-11 20:29:08 (18.5 MB/s) - written to stdout [1793/1793]

```

```

root@ip-172-31-33-125:/home/ubuntu# echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packages.adoptium.net/artifactory/_deb $(awk -F= '/^VERSION_CODENAME/{print$2}' /etc/os-release) main" | tee /etc/apt/sources.list.d/adoptium.list
deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packages.adoptium.net/artifactory/_deb jammy main
root@ip-172-31-33-125:/home/ubuntu#

```

Install Java 17

```

$ apt update

$ apt install temurin-17-jdk

$ update-alternatives --config java

$ /usr/bin/java --version $ exit

```

```

root@ip-172-31-33-125:/home/ubuntu# echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://pack/_deb $(awk -F= '/^VERSION_CODENAME/{print$2}' /etc/os-release) main" | tee /etc/apt/sources.list.d/ad deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packages.adoptium.net/artifactory/_deb jammy ma root@ip-172-31-33-125:/home/ubuntu# apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://packages.adoptium.net/artifactory/_deb jammy InRelease [6632 B]
Get:7 https://packages.adoptium.net/artifactory/_deb jammy/main amd64 Packages [6385 B]
Fetched 13.0 kB in 3s (4295 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@ip-172-31-33-125:/home/ubuntu# apt install temurin-17-jdk

```

```
2. 15.207.84.111 (ubuntu) 3. 13.233.208.255 (ubuntu) 4. 13.127.136.59 (ubuntu)
root@ip-172-31-33-125:/home/ubuntu# update-alternatives --config java
There is only one alternative in link group java (providing /usr/bin/java): /usr/lib/jvm/temurin-17-jdk-amd64/bin/java
Nothing to configure.
root@ip-172-31-33-125:/home/ubuntu#
```



```
2. 15.207.84.111 (ubuntu) 3. 13.233.208.255 (ubuntu) 4. 13.127.136.59 (ubuntu)
root@ip-172-31-33-125:/home/ubuntu# update-alternatives --config java
There is only one alternative in link group java (providing /usr/bin/java): /usr/lib/jvm/temurin-17-jdk-amd64/bin/java
Nothing to configure.
root@ip-172-31-33-125:/home/ubuntu# /usr/bin/java --version
openjdk 17.0.9 2023-10-17
OpenJDK Runtime Environment Temurin-17.0.9+9 (build 17.0.9+9)
OpenJDK 64-Bit Server VM Temurin-17.0.9+9 (build 17.0.9+9, mixed mode, sharing)
root@ip-172-31-33-125:/home/ubuntu# exit
exit
ubuntu@ip-172-31-33-125:~$
```

Linux Kernel Tuning

Increase Limits

```
$ sudo vim /etc/security/limits.conf //Paste the below values at the bottom of
the file sonarqube - nofile 65536
```

sonarqube - nproc 4096

```
root@ip-172-31-33-125:/home/ubuntu# /usr/bin/java --version
openjdk 17.0.9 2023-10-17
OpenJDK Runtime Environment Temurin-17.0.9+9 (build 17.0.9+9)
OpenJDK 64-Bit Server VM Temurin-17.0.9+9 (build 17.0.9+9, mixed mode, sharing)
root@ip-172-31-33-125:/home/ubuntu# exit
exit
ubuntu@ip-172-31-33-125:~$ sudo vim /etc/security/limits.conf
```

```
# End of file
sonarqube    -    nofile    65536
sonarqube    -    nproc     4096
-- INSERT --
```

Increase Mapped Memory Regions

```
sudo vim /etc/sysctl.conf //Paste the below values at the bottom of the file
vm.max_map_count = 262144
```

```
exit
ubuntu@ip-172-31-33-125:~$ sudo vim /etc/security/limits.conf
ubuntu@ip-172-31-33-125:~$ sudo vim /etc/sysctl.conf
```

```
#  
#####  
# Magic system request Key  
# 0=disable, 1=enable all, >1 bitmask of sysrq functions  
# See https://www.kernel.org/doc/html/latest/admin-guide/sysrq.html  
> # for what other values do  
#kernel.sysrq=438  
vm.max_map_count = 262144  
-- TNSFRT --
```

Restart the server --> `sudo init 6`

```
ubuntu@ip-172-31-33-125:~$ sudo vim /etc/security/limits.conf  
ubuntu@ip-172-31-33-125:~$ sudo vim /etc/sysctl.conf  
ubuntu@ip-172-31-33-125:~$ sudo init 6
```

Remote side unexpectedly closed network connection

```
Session stopped  
- Press <Return> to exit tab  
- Press R to restart session  
- Press S to save terminal output to file
```

Go to the SonarQube instance and allow port number 9000 because sonarqube will run on 9000 port

FIND INSTANCE BY ATTRIBUTE OR TAG (CASE-SENSITIVE)

Name	Instance ID	Instance state
Jenkins-Master	i-01c22280...	Running
Jenkins-Agent	i-016dbbc8...	Running
SonarQube	i-0e22c3b6...	Running

Instance: i-0e22c3b63dac00a87 (SonarQube)

Details | **Security** | Networking | Storage | Status checks

▼ Security details

IAM Role	Owner ID
-	955232677187
Security groups	
<input checked="" type="checkbox"/> sg-0a7f62ad6fe099324 (launch-wizard-12)	

Inbound rules | Outbound rules | Tags

Inbound rules (1)

Name	Security group rule...	IP version	Type	Protocol
-	sgr-097524e16f2c72fa9	IPv4	SSH	TCP

EC2 > Security Groups > sg-0a7f62ad6fe099324 - launch-wizard-12 > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules <small>Info</small>																		
<table border="1"> <thead> <tr> <th>Security group rule ID</th> <th>Type <small>Info</small></th> <th>Protocol <small>Info</small></th> <th>Port range <small>Info</small></th> <th>Source Info</th> <th>Description - optional <small>Info</small></th> </tr> </thead> <tbody> <tr> <td>sgr-097524e16f2c72fa9</td> <td>SSH</td> <td>TCP</td> <td>22</td> <td>Custom</td> <td><input type="text" value="0.0.0.0/0"/> <input type="button" value="X"/></td> </tr> <tr> <td>-</td> <td>Custom TCP</td> <td>TCP</td> <td>9000</td> <td>Anywhere</td> <td><input type="text" value="0.0.0.0/0"/> <input type="button" value="X"/></td> </tr> </tbody> </table> <p>Add rule</p> <p><small>⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.</small></p> <p><small>Cancel Preview changes Save rules</small></p>	Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source Info	Description - optional <small>Info</small>	sgr-097524e16f2c72fa9	SSH	TCP	22	Custom	<input type="text" value="0.0.0.0/0"/> <input type="button" value="X"/>	-	Custom TCP	TCP	9000	Anywhere	<input type="text" value="0.0.0.0/0"/> <input type="button" value="X"/>
Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source Info	Description - optional <small>Info</small>													
sgr-097524e16f2c72fa9	SSH	TCP	22	Custom	<input type="text" value="0.0.0.0/0"/> <input type="button" value="X"/>													
-	Custom TCP	TCP	9000	Anywhere	<input type="text" value="0.0.0.0/0"/> <input type="button" value="X"/>													

Now install SonarQube on This Instance



Sonarqube Installation

Download and Extract

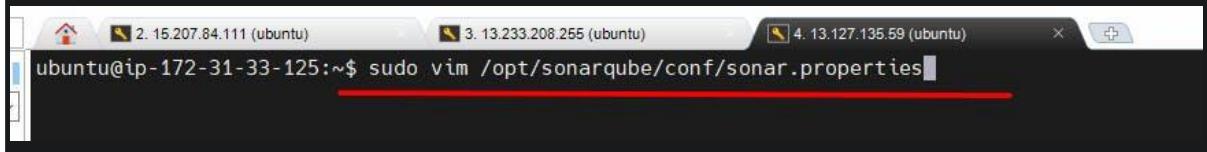
```
$ sudo wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.9.0.65466.zip
$ sudo apt install unzip
$ sudo unzip sonarqube-9.9.0.65466.zip -d /opt
$ sudo mv /opt/sonarqube-9.9.0.65466 /opt/sonarqube
[...]
  inflating: /opt/sonarqube-9.9.0.65466/lib/sonar-shutdowner-9.9.0.65466.jar
  creating: /opt/sonarqube-9.9.0.65466/elasticsearch/plugins/
> ubuntu@ip-172-31-33-125:~$ sudo mv /opt/sonarqube-9.9.0.65466 /opt/sonarqube
ubuntu@ip-172-31-33-125:~$ sudo groupadd sonar
ubuntu@ip-172-31-33-125:~$ sudo useradd -c "user to run SonarQube" -d /opt/sonarqube -g sonar sonar
ubuntu@ip-172-31-33-125:~$ sudo chown sonar:sonar /opt/sonarqube -R
```

Create a user and set permissions

```
$ sudo groupadd sonar
$ sudo useradd -c "user to run SonarQube" -d /opt/sonarqube -g sonar sonar
$ sudo chown sonar:sonar /opt/sonarqube -R
```

Update Sonarqube properties with DB credentials

```
$ sudo vim /opt/sonarqube/conf/sonar.properties //Find and replace the below values, you might need to add the sonar.jdbc.url sonar.jdbc.username=sonar sonar.jdbc.password=sonar  
sonar.jdbc.url=jdbc:postgresql://localhost:5432/sonarqube
```



```
# - The embedded H2 database is used by default. It is recommended for tests but not for production use. Supported databases are Oracle, PostgreSQL and Microsoft SQLServer.  
# - Changes to database connection URL (sonar.jdbc.url) can affect SonarSource licensed products.  
  
# User credentials.  
# Permissions to create tables, indices and triggers must be granted to JDBC user.  
# The schema must be created first.  
sonar.jdbc.username=sonar  
sonar.jdbc.password=sonar  
  
----- Embedded Database (default)  
# H2 embedded database server listening port, defaults to 9092  
#sonar.embeddedDatabase.port=9092  
  
----- Oracle 19c/21c  
# The Oracle JDBC driver must be copied into the directory extensions/jdbc-driver/oracle/.  
# Only the thin client is supported, and we recommend using the latest Oracle JDBC driver. See  
# https://jira.sonarsource.com/browse/SONAR-9758 for more details.  
# If you need to set the schema, please refer to http://jira.sonarsource.com/browse/SONAR-5000  
sonar.jdbc.url=jdbc:postgresql://localhost:5432/sonarqube  
  
----- PostgreSQL 11 or greater  
# By default the schema named "public" is used. It can be overridden with the parameter "currentSchema"  
#sonar.jdbc.url=jdbc:postgresql://localhost/sonarqube?currentSchema=my_schema
```

Create service for Sonarqube

```
$ sudo vim /etc/systemd/system/sonar.service //Paste the below into the file  
[Unit] Description=SonarQube service After=syslog.target network.target  
  
[Service] Type=forking  
  
ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start  
ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop  
  
User=sonar Group=sonar Restart=always  
  
LimitNOFILE=65536 LimitNPROC=4096  
  
[Install] WantedBy=multi-user.target
```

```
2. 15.207.84.111 (ubuntu) 3. 13.233.208.255 (ubuntu) 4. 13.127.135.59 (ubuntu)
ubuntu@ip-172-31-33-125:~$ sudo vim /opt/sonarqube/conf/sonar.properties
ubuntu@ip-172-31-33-125:~$ sudo vim /etc/systemd/system/sonar.service

[Unit]
Description=SonarQube service
After=syslog.target network.target

[Service]
Type=forking

ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start
ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop

User=sonar
Group=sonar
Restart=always

LimitNOFILE=65536
LimitNPROC=4096

[Install]
WantedBy=multi-user.target
~
```

Start Sonarqube and Enable service

```
$ sudo systemctl start sonar
$ sudo systemctl enable sonar
$ sudo systemctl status sonar
```

```

2. 15.207.84.111 (ubuntu) 3. 13.233.208.255 (ubuntu) 4. 13.127.135.59 (ubuntu)
ubuntu@ip-172-31-33-125:~$ sudo vim /opt/sonarqube/conf/sonar.properties
ubuntu@ip-172-31-33-125:~$ sudo vim /etc/systemd/system/sonar.service
ubuntu@ip-172-31-33-125:~$ sudo systemctl start sonar
ubuntu@ip-172-31-33-125:~$ sudo systemctl enable sonar
Created symlink /etc/systemd/system/multi-user.target.wants/sonar.service → /etc/systemd/system/sonar.service.
ubuntu@ip-172-31-33-125:~$ sudo systemctl status sonar
● sonar.service - SonarQube service
    Loaded: loaded (/etc/systemd/system/sonar.service; enabled; vendor preset: enabled)
      Active: active (running) since Sat 2023-11-11 20:57:43 UTC; 18s ago
        Main PID: 1183 (java)
       Tasks: 87 (limit: 4598)
      Memory: 1004.8M
         CPU: 34.366s
      CGroup: /system.slice/sonar.service
              └─1183 java -Xms8m -Xmx32m --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED -XX:+UseG1GC -Djava.io.tmpdir=/opt/sonarqube/temp -XX:ErrorFile=/opt/sonarqube/logs/sonar.log -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/opt/sonarqube/temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.nio=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/sun.management=ALL --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Dcom.redhat.fips=false -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[::1] -cp ./lib/sonar-application-9.9.0.65466.jar:/opt/sonarqube/lib/jdbc/tgresql-42.5.1.jar org.sonar.server.app.WebServer /opt/sonarqube/temp/sq-process14792958574606529031properties
Nov 11 20:57:43 ip-172-31-33-125 systemd[1]: Starting SonarQube service...
Nov 11 20:57:43 ip-172-31-33-125 sonar.sh[1160]: /usr/bin/java
Nov 11 20:57:43 ip-172-31-33-125 sonar.sh[1160]: Starting SonarQube...
Nov 11 20:57:43 ip-172-31-33-125 sonar.sh[1160]: Started SonarQube.
Nov 11 20:57:43 ip-172-31-33-125 systemd[1]: Started SonarQube service.
lines 1-17/17 (END)

```

Watch log files and monitor for startup

```
$ sudo tail -f /opt/sonarqube/logs/sonar.log
```

```

2. 15.207.84.111 (ubuntu) 3. 13.233.208.255 (ubuntu) 4. 13.127.135.59 (ubuntu)
ubuntu@ip-172-31-33-125:~$ sudo tail -f /opt/sonarqube/logs/sonar.log
2023.11.11 20:57:44 INFO app[][o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
2023.11.11 20:57:55 INFO app[][o.s.a.SchedulerImpl] Process[es] is up
2023.11.11 20:57:55 INFO app[][o.s.a.ProcessLauncherImpl] Launch process[WEB_SERVER] from [/opt/sonarqube]: /usr/lib/jdk-amd64/bin/java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/opt/sonarqube/temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.nio=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/sun.management=ALL --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Dcom.redhat.fips=false -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[::1] -cp ./lib/sonar-application-9.9.0.65466.jar:/opt/sonarqube/lib/jdbc/tgresql-42.5.1.jar org.sonar.server.app.WebServer /opt/sonarqube/temp/sq-process14792958574606529031properties
2023.11.11 20:58:00 INFO app[][o.s.a.SchedulerImpl] Process[web] is up
2023.11.11 20:58:50 INFO app[][o.s.a.ProcessLauncherImpl] Launch process[COMPUTE_ENGINE] from [/opt/sonarqube]: /usr/lib/jdk-amd64/bin/java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/opt/sonarqube/temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.lang=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/sun.management=ALL --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Dcom.redhat.fips=false -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[::1] -cp ./lib/sonar-application-9.9.0.65466.jar:/opt/sonarqube/lib/jdbc/postgresql/postgresql-42.5.1.jar org.sonar.ce.app.CeServer /opt/sonarqube/temp/sq-process14792958574606529031properties
2023.11.11 20:58:53 WARN app[] [startup] #####
2023.11.11 20:58:53 WARN app[] [startup] Default Administrator credentials are still being used. Make sure to change or deactivate the account.
2023.11.11 20:58:53 WARN app[] [startup] #####
2023.11.11 20:58:58 INFO app[] [o.s.a.SchedulerImpl] Process[ce] is up
2023.11.11 20:58:58 INFO app[] [o.s.a.SchedulerImpl] SonarQube is operational

```

Now go to sonarqube instance and copy PublicIP address for access sonarqube

The screenshot shows the AWS CloudWatch Instances dashboard. A table lists two instances: 'Jenkins-Agent' (t2.micro) and 'SonarQube' (t3.medium). The 'SonarQube' row is selected and highlighted with a red box. Below the table, a section titled 'Instance: i-0e22c3b63dac00a87 (SonarQube)' displays various tabs: Details (selected), Security, Networking, Storage, Status checks, Monitoring, and Tags. Under the 'Details' tab, the 'Instance summary' section shows the 'Public IPv4 address' as 13.127.135.59, which is also highlighted with a red box.

Go to Browser and paste publicip with port number 9000

Give admin admin username and password and login

The screenshot shows a browser window displaying the SonarQube login page. The URL in the address bar is 13.127.135.59:9000/sessions/new?return_to=%2F. The page title is 'Log in to SonarQube'. It features two input fields: 'Login' and 'Password', both of which are underlined with red. Below the fields are two buttons: 'Log in' and 'Cancel', with 'Log in' also underlined in red.

Update your passwoprd

Update your password

This account should not use the default password.

Enter a new password

All fields marked with * are required

Old Password *

New Password *

Confirm Password *

Update

This is the Sonar Qube dashboard

Not secure 13.127.135.59:9000/projects/create

There's an update available for your SonarQube instance. Please update to make sure you benefit from the latest security and bug fixes. [Learn More](#)

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration Search for projects... A

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform. First, you need to set up a DevOps platform configuration.

 From Azure DevOps Set up global configuration	 From Bitbucket Server Set up global configuration	 From Bitbucket Cloud Set up global configuration	 From GitHub Set up global configuration	 From GitLab Set up global configuration
--	--	---	--	--

Are you just testing or have an advanced use-case? Create a project manually.

 Manually

 Get the most out of SonarQube!
Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.

[Learn More](#) [Dismiss](#)

Integrate SonarQube with Jenkins

Integrate SonarQube with Jenkins

So go to Myaccount in that go to security

The screenshot shows the 'Create Project' section of the SonarQube interface. At the top, there's a navigation bar with links for Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. A search bar is also present. On the right, a dropdown menu for 'Administrator' is open, showing 'My Account' and 'Log out'. Below the navigation, a message asks how to create a project, mentioning Azure DevOps, Bitbucket, GitHub, and GitLab integration. Five cards provide links to set up global configuration for each of these platforms.

Click on generate token copy it

The screenshot shows the 'Tokens' page under the 'Security' tab. It includes a 'Generate Tokens' form with fields for Name, Type (set to 'Global'), and Expires in (set to '30 days'). A 'Generate' button is highlighted with a red box. Below the form, a success message states 'New token "jenkins-sonarqube-token" has been created. Make sure you copy it now, you won't be able to see it again!' A 'Copy' button next to the token value 'sqa_72b23a363aceb9d6127a312b6d8515362fa3fbfd' is also highlighted with a red box. A table at the bottom lists the generated token details: Name 'jenkins-sonarqube-token', Type 'Global', Last use 'Never', Created 'November 12, 2023', and a 'Revoke' button.

Go to Manage Jenkins --> Credentials

[+ New Item](#)

Manage Jenkins

[People](#)

[Build History](#)

[Project Relationship](#)

[Check File Fingerprint](#)

[Manage Jenkins](#)

[My Views](#)

Build Queue
No builds in the queue.

Build Executor Status
Jenkins-Agent
1 Idle

System Configuration

[System](#) Configure global settings and paths.

[Nodes](#) Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

[Tools](#) Configure tools, their locations and automatic installers.

[Clouds](#) Add, remove, and configure cloud instances to provision agents on-demand.

Security

[Security](#) Secure Jenkins; define who is allowed to access/use the system.

[Credentials](#) Configure credentials

Add Credentials

Dashboard > Manage Jenkins > Credentials

Credentials

T	P	Store	Domain	ID	Name
		System	(global)	Jenkins-Agent	ubuntu (Jenkins-Agent)
		System	(global)	gitHub	thoratsunil/******** (gitHub)

Stores scoped to Jenkins

P	Store	Domains
	System	(global)

[Add credentials](#)

Icon: S M L

Kind: Secret text

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind: Secret text

Scope: Global (Jenkins, nodes, items, all child items, etc)

Secret: Paste here sonar qube copied key

ID: jenkinsa-sonarqube-token

Description: jenkins-sonarqube-token

Create

jenkins-sonarqube-token created

Global credentials (unrestricted)

+ Add Credentials

ID	Name	Kind	Description
Jenkins-Agent	ubuntu (Jenkins-Agent)	SSH Username with private key	Jenkins-Agent
gitHub	thoratsunil/******** (gitHub)	Username with password	gitHub
jenkins-sonarqube-token	jenkins-sonarqube-token	Secret text	jenkins-sonarqube-token

Now go to Manage Jenkins --> Plugins

Dashboard > Manage Jenkins

Manage Jenkins

- + New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins**
- My Views
- Build Queue

System Configuration

- System: Configure global settings and paths.
- Tools: Configure tools, their locations and automatic installers.
- Clouds: Add, remove, and configure cloud instances to provision agents on-demand.
- Managed files: e.g. settings.xml for maven, central managed scripts, custom files, ...

Plugins: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

INInstall sonarqube plugins

The screenshot shows the Jenkins 'Plugins' page under 'Manage Jenkins'. A search bar at the top contains the query 'sonar'. Below it, a sidebar has 'Available plugins' selected. The main area lists three Sonar-related plugins:

- SonarQube Scanner**: Version 2.16.1, Released 1 mo 2 days ago. Description: This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
- Sonar Quality Gates**: Version 1.3.1, Released 5 yr 2 mo ago. Description: Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed"). Warning: This plugin version may not be safe to use. Please review the following security notices:
 - Credentials transmitted in plain text
- Quality Gates**: Version 2.5, Released 7 yr 5 mo ago. Description: Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed"). Warning: This plugin version may not be safe to use. Please review the following security notices:
 - Credentials transmitted in plain text

Download progress

Preparation

- Checking Internet connectivity
- Checking update center connectivity
- Success

Ionicons API		Success
Folders		Success
OWASP Markup Formatter		Success
Structs		Success
bouncycastle API		Success
Instance Identity		Success
JavaBeans Activation Framework (JAF) API		Success
JavaMail API		Success
Pipeline: Step API		Success
Token Macro		Success
Build Timeout		Success
Credentials		Success
Plain Credentials		Success
Trilead API		Success
SSH Credentials		Success
Credentials Binding		Success
SCM API		Success
Pipeline: API		Success
commons-lang3 v3.x Jenkins API		Success
Timestamper		Success
Caffeine API		Success

After Installation of plugins restart the Jenkins



Please wait while Jenkins is restarting ...

Your browser will reload automatically when Jenkins is ready.

Safe Restart

Builds on agents can usually continue.

After restart go to Manage Jenkins --> System Configure

Manage Jenkins

- + New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views

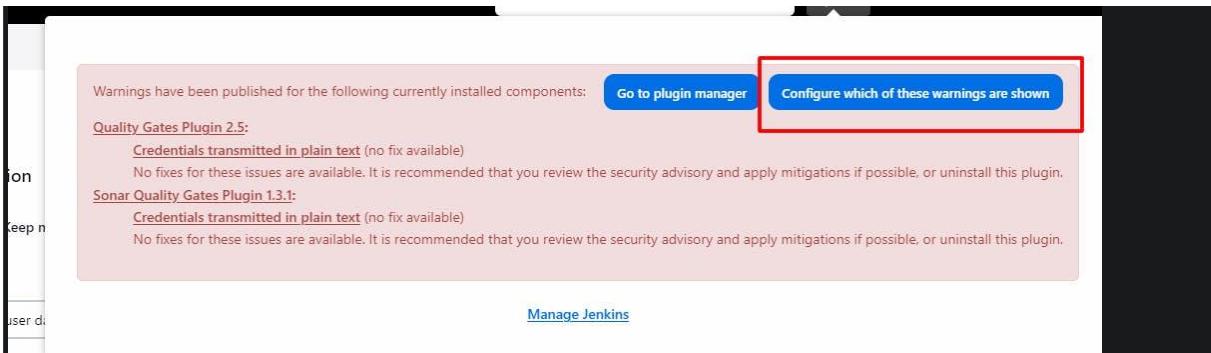
System Configuration

System
Configure global settings and paths.

Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Build Queue ▾
No builds in the queue.

To remove this popup Go to Configure which of these warnings are shown



Go to Hudden Security warnings and uncheck this two options

Dashboard > Security

Hidden security warnings

Security warnings ^ Edited

This section allows you to suppress warnings, applicable to your Jenkins configuration Warnings administrative monitor. Checked warnings are reported (the default), unchecked ones are suppressed.

Security warnings ?

Sonar Quality Gates Plugin: Credentials transmitted in plain text
<https://jenkins.io/security/advisory/2020-03-09/#SECURITY-1523>

Quality Gates Plugin: Credentials transmitted in plain text
<https://jenkins.io/security/advisory/2020-03-09/#SECURITY-1519>

API Token

Generate a legacy API token for each newly created user (Not recommended) ?

Allow users to manually create a legacy API token (Not recommended) ?

Save **Apply** (highlighted with a red box)

Now go to SonarQube Servers in the Manage Jenkins --> System

In that Paste Private Ip Address of SonarQube here

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables.

Environment variables

SonarQube installations

List of SonarQube installations

Name: sonarqube-server

Server URL: Default is http://localhost:9000
Paste here private IP of sonarqube: http://

Server authentication token: - none -

Save **Apply**

Instance: i-0e22c3b63dac00a87 (SonarQube)

Details **Security** **Networking** **Storage** **Status checks** **Monitoring** **Tags**

Instance summary **Info**

Instance ID: i-0e22c3b63dac00a87 (SonarQube)	Public IPv4 address: 13.127.135.59 Open address	Private IPv4 addresses: 172.31.33.125
IPv6 address: -	Instance state: Running	Public IPv4 DNS: ec2-13-127-135-59.ap-south-1.compute.amazonaws.com Open address

Dashboard > Manage Jenkins > System >

Environment variables

SonarQube installations

List of SonarQube installations

Name: sonarqube-server

This property is mandatory.

Server URL: Default is http://localhost:9000

http://172.31.33.125:9000

Server authentication token: SonarQube authentication token. Mandatory when using the SonarQube analysis feature.

jenkins-sonarqube-token

+ Add ▾

Save **Apply**

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue: No builds in the queue.

Build Executor Status: Jenkins-Agent

System Configuration

System: Configure global settings and paths.

Tools: Configure tools, their locations and automatic installers.

Nodes: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Clouds: Add, remove, and configure cloud instances to provision agents on-demand.

Security

The screenshot shows the Jenkins configuration interface for SonarQube Scanner installations. A red box highlights the title "SonarQube Scanner installations". Below it is a button "Add SonarQube Scanner". The main section is titled "SonarQube Scanner" with a "Name" field containing "sonarqube-scanner" and an "Install automatically" checkbox checked. A sub-section "Install from Maven Central" shows the version "SonarQube Scanner 5.0.1.3006". There is also an "Add Installer" dropdown. At the bottom, there is a "Save" button highlighted with a red box, an "Apply" button, and three icons: a folder, a clipboard, and a close button.

Go to jenkinsfile and add one more stage SonarQube Analysis

The screenshot shows the Jenkinsfile code editor. On the left, there are files: Jenkinsfile, README.md, and pom.xml. The Jenkinsfile content is as follows:

```
25          sh 'mvn test'
26      }
27  }
28  stage("SonarQube Analysis"){
29      steps {
30          script {
31              withSonarQubeEnv(credentialsId: 'Jenkins-sonarqube-token') {
32                  sh "mvn sonar:sonar"
33              }
34          }
35      }
}
```

A red box highlights the "SonarQube Analysis" stage block. At the bottom right of the code editor, there are save, apply, and other file management icons.

Commit changes

X

Commit message

Update Jenkinsfile

Extended description

Add an optional extended description..

- Commit directly to the `master` branch
- Create a **new branch** for this commit and start a pull request
[Learn more about pull requests](#)

Cancel

Commit changes

S	W	Name	Last Success	Last Failure	Last Duration	
		register-app-ci	10 hr #1	N/A	1 min 53 sec	

Status  register-app-ci

</> Changes

 Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

Average stage times:
(Average full run time: ~1min 53s)

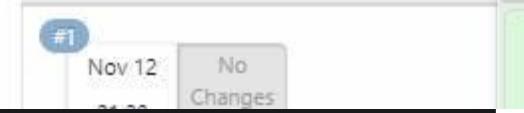
Declarative Checkout SCM

8s

8s

Stage View

Build History trend ▾



Now click on build now and check all stages

Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

Add descriptive...

Disable Proj...

Build History trend ▾

Build done : some spelling mistake issue but solved and build done

Stage View

Average stage times:
(Average full run time: ~1min 20s)

	Declarative: Checkout SCM	Declarative: Tool Install	Cleanup Workspace	Checkout from SCM	Build Application	Test Application	SonarQube Analysis
#1 Nov 12 11:36 1 commit	3s	18s	481ms	1s	11s	6s	9s
	995ms	179ms	350ms	1s	9s	6s	27s

Filter builds... 7 / 1 Nov 12

```

[INFO] Monday 13: TOTAL TIME: 19.000 s
[INFO] -----
[INFO] Reactor Summary for Maven Project 1.0-SNAPSHOT:
[INFO]
[INFO] Maven Project ..... SUCCESS [ 19.403 s]
[INFO] Server ..... SKIPPED
[INFO] Webapp ..... SKIPPED
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 22.898 s
[INFO] Finished at: 2023-11-12T06:07:22Z
[INFO] -----
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Go to the sonarqube dashboard and see all stages are green means no error

The screenshot shows the SonarQube interface. The top navigation bar has tabs for 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. The 'Projects' tab is highlighted with a red box. Below the navigation is a search bar and a 'Create Project' button. On the left, there are filters for 'My Favorites' and 'All' projects, and sections for 'Quality Gate' (Passed), 'Reliability' (A rating), and various code quality metrics like Bugs, Vulnerabilities, and Code Smells. A red box highlights the 'Maven Project' card, which shows a green 'Passed' status and details about the analysis: 0 bugs, 0 vulnerabilities, 2 code smells, 0.0% coverage, 0.0% duplications, and 311 lines of XML/JSP code.

Now add Jenkins in sonarqube pipeline

Go to Administration -- Configuration -- Webhooks

The screenshot shows the SonarQube administration interface. The top navigation bar has tabs for sonarqube, Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. The Administration tab is highlighted with a red box. Below the navigation is a secondary menu with Configuration, Security, Projects, System, and Marketplace. A dropdown menu under Configuration is open, showing General Settings, Encryption, and Webhooks, with Webhooks also highlighted by a red box. To the right of the dropdown is a note about enabling webhooks for this SonarQube instance. At the bottom of the screen are Analysis Scope and Duplications buttons.

Click on Create

The screenshot shows the SonarQube Webhooks configuration page. The top navigation bar includes sonarqube, Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. The Administration tab is selected. Below the navigation is a search bar and a 'Create' button, which is highlighted with a red box. The main content area is titled 'Webhooks' and contains a note about what webhooks do. It shows a message 'No webhook defined.' and a red horizontal line underneath it. There is also a link to the 'Webhooks documentation'.

Name sonarqube-webhook ---> paste Private Ip of Jenkins-server

Create Webhook

All fields marked with * are required

Name *
sonarqube-webhook

URL *
http://172.31.33.81:8080/sonarqube-webhook/

Private IP of Jenkins Server
Server endpoint that will receive the webhook payload, for example:
"http://my_server/foo". If HTTP Basic authentication is used, HTTPS is recommended to avoid man in the middle attacks. Example:
"https://myLogin:myPassword@my_server/foo"

Secret
[redacted]

If provided, secret will be used as the key to generate the HMAC hex (lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256' header

Create **Cancel**

sonarqube-webhook created

Administration				
Configuration Security Projects System Marketplace				
Create				
Webhooks				
Webhooks are used to notify external services when a project analysis is done. An HTTP POST request including a JSON payload is sent to each of the provided URLs. Learn more in the Webhooks documentation .				
Name	URL	Secret?	Last delivery	
sonarqube-webhook	http://172.31.33.81:8080/sonarqube-webhook/	No	Never	⚙️

Go to Jenkins file and add quality gate stage

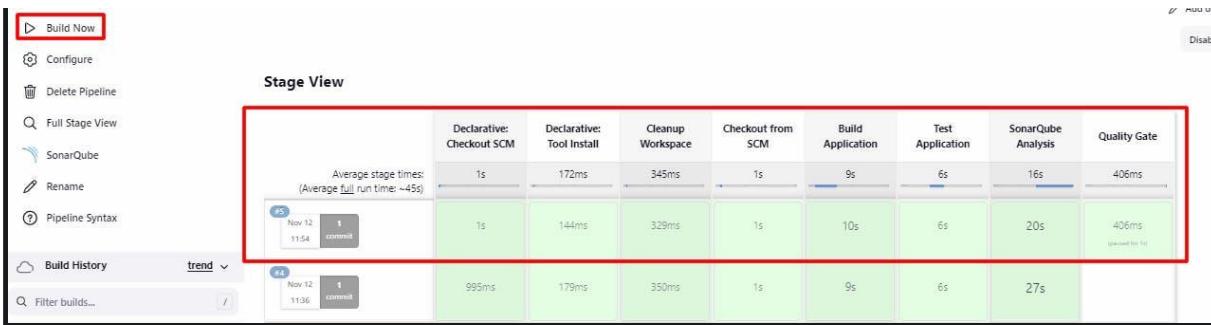
```

Jenkinsfile
README.md
pom.xml

32           sh "mvn sonar:sonar"
33
34         }
35     }
36   }
37   stage("Quality Gate"){
38     steps {
39       script {
40         waitForQualityGate abortPipeline: false, credentialsId: 'jenkins-sonarqube-token'
41       }
42     }
43   }

```

After that click on Build Now we will see all stages will be successfully done



Build and Push Docker Image using Pipeline Script

Now add one more stage for Build and Push Docker Image using Pipeline Script

so go to the Manage Jenkins --> Plugins

The screenshot shows the Jenkins System Configuration page. The sidebar on the left includes links for 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins' (which is highlighted with a red box), and 'My Views'. The main area has sections for 'System Configuration' (System, Tools, Clouds, Managed files), and a 'Plugins' section which is also highlighted with a red box. The 'Plugins' section contains a brief description: 'Add, remove, disable or enable plugins that can extend the functionality of Jenkins.'

Install docker plugins ---> 6 Plugins

The screenshot shows the Jenkins Plugins page. The top navigation bar shows 'Dashboard > Manage Jenkins > Plugins'. The sidebar on the left has tabs for 'Updates', 'Available plugins' (which is selected and highlighted with a red box), 'Installed plugins', and 'Advanced settings'. A search bar at the top right contains the text 'docker'. The main table lists several Docker-related plugins:

- Docker 1.5 (Cloud Providers, Cluster Management, docker) - Released 2 mo 8 days ago
- Docker Commons 439.va_3cb:0a_6a_fb_29 (Library plugins (for use by other plugins), docker) - Released 4 mo 4 days ago
- Docker Pipeline 572.v950f8993843 (pipeline, DevOps, Deployment, docker) - Released 3 mo 2 days ago
- Docker API 3.3.1-79.v20b_53427e041 (Library plugins (for use by other plugins), docker) - Released 5 mo 9 days ago
- docker-build-step 2.10 (Build Tools, docker) - Released 1 mo 8 days ago
- CloudBees Docker Build and Publish 1.4.0 (Build Tools, docker) - Released 1 yr 2 mo ago

A large red box highlights the 'Available plugins' tab and the 'Install' button for the Docker plugin.

Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Download progress

Preparation

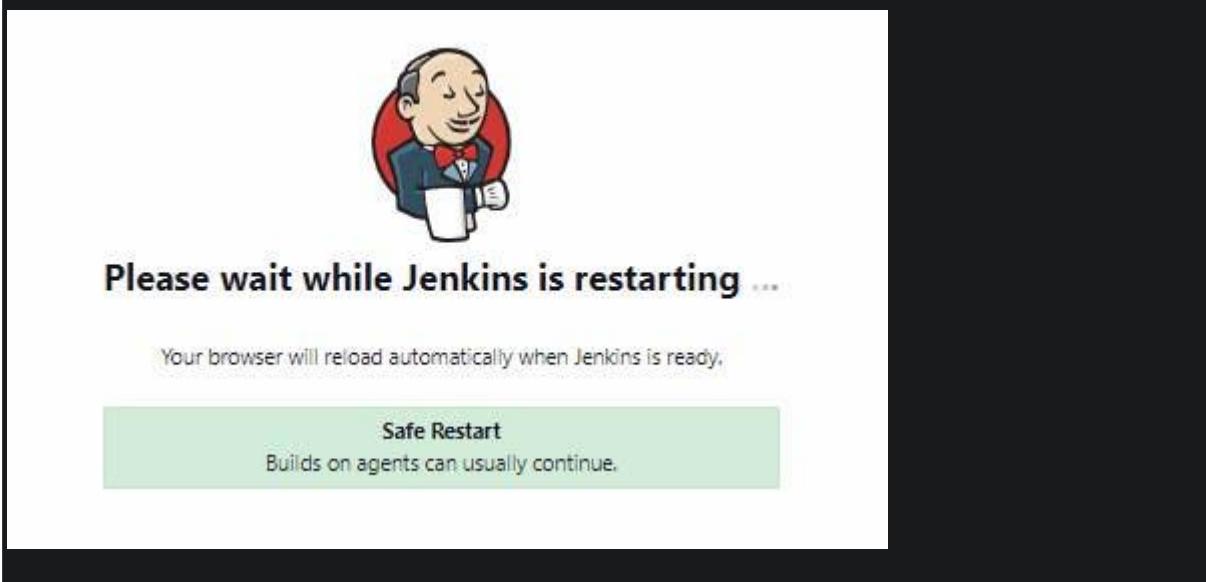
- Checking internet connectivity Success
- Checking update center connectivity Success
- Success

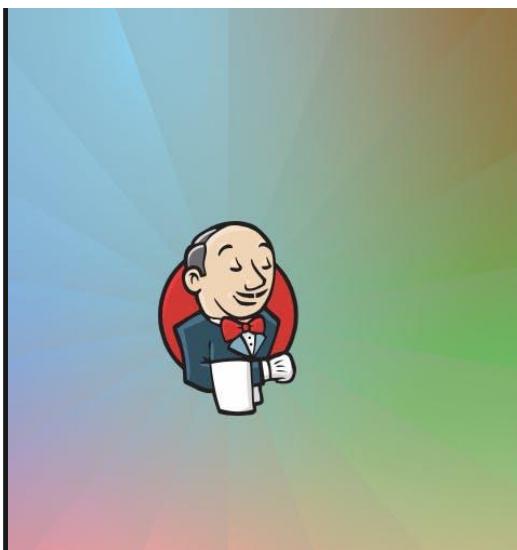
Cloud Statistics	Success
Authentication Tokens API	Success
Docker Commons	Success
Apache HttpComponents Client 5.x API	Success
Docker API	Success
Docker	Success
Docker Commons	Success
Docker Pipeline	Success
Docker API	Success
docker-build-step	Success
CloudBees Docker Build and Publish	Success
Loading plugin extensions	Success

→ Go back to the top page
(you can start using the installed plugins right away)

→ Restart Jenkins when installation is complete and no jobs are running

After installation restart the Jenkins





Sign in to Jenkins

Username

Password

Keep me signed in

Sign in

The 'Sign in' button is highlighted with a red box.

After login go to Manage Jenkins- Credentials

Manage Jenkins (highlighted with a red box)

My Views

Build Queue (No builds in the queue)

Build Executor Status (Jenkins-Agent, 1 Idle, 2 Idle)

System (Configure global settings and paths)

Clouds (Add, remove, and configure cloud instances to provision agents on-demand)

Tools (Configure tools, their locations and automatic installers)

Managed files (e.g. settings.xml for maven, central managed scripts, custom files, ...)

Security (Secure Jenkins; define who is allowed to access/use the system)

Credentials (highlighted with a red box) Configure credentials

add credentials

Stores scoped to Jenkins

P Store 1 Domains

Icon: S M L

System (highlighted with a red box)

(global) (highlighted with a red box)

Add credentials (highlighted with a red box)

Go to dockerhub account settings

0

sriyeshwanthe

What's new

My profile

Account settings

Billing

Security --> New Access Token

sriyeshwanthe

user Joined June 25, 2024

General

Security

Default Privacy

Notifications

Convert Account

Deactivate Account

	Description	Source	Scope	Last Used	Created	
<input type="checkbox"/>	jenkins token	MANUAL	Read, Write, Delete	Jun 26, 2024, 12:56:52	Jun 25, 2024, 14:50:24	<more></more>

New Access Token

Two-Factor Authentication

Two-factor authentication is not enabled yet.

Type anything in the description ; project5

New Access Token

A personal access token is similar to a password except you can have many tokens and revoke access to each one at any time. [Learn more](#)

Access Token Description *

project5

Access permissions

Read, Write, Delete

Read, Write, Delete tokens allow you to manage your repositories.

Cancel

Generate

Go to jenkins credentials and paste password section newly created token

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: dockerhub usedID
thoratsunil121

Treat username as secret

Password: Paste here docker Hub Token

ID: dockerhub

Description: dockerhub

Create

dockerhub credentials ready

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted)

Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
Jenkins-Agent	ubuntu (Jenkins-Agent)	SSH Username with private key	Jenkins-Agent
gitHub	thoratsunil/******/ (gitHub)	Username with password	gitHub
jenkins-sonarqube-token	jenkins-sonarqube-token	Secret text	jenkins-sonarqube-token
dockerhub	thoratsunil121/******/ (dockerhub)	Username with password	dockerhub

Icon: S M L

Now add environment variables in stages in groovy syntax

```
        }
    Add environment variables here for dockerhub
environment {
    APP_NAME = "register-app-pipeline"
    RELEASE = "1.0.0"
    DOCKER_USER ="thoratsunill21"
    DOCKER_PASS ="dockerhub"
    IMAGE_NAME = "${DOCKER_USER}" + "/" + "${APP_NAME}"
    IMAGE_TAG = "${RELEASE}-${BUILD_NUMBER}"
stages {
    stage("Cleanup Workspace"){
        steps{
            cleanWs()
```

```
}
```

```
environment {
    APP_NAME = "register-app-pipeline"
    RELEASE = "1.0.0"
    DOCKER_USER ="thoratsunill21"
    DOCKER_PASS ="dockerhub"
    IMAGE_NAME = "${DOCKER_USER}" + "/" + "${APP_NAME}"
    IMAGE_TAG = "${RELEASE}-${BUILD_NUMBER}"
}
```

```
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
```

```
        }
    stage("Build & Push Docker Image"){
        steps {
            script {
                docker.withRegistry(' ',DOCKER_PASS) {
                    docker_image = docker.build "${IMAGE_NAME}"
                }
                docker.withRegistry('',DOCKER_PASS) {
                    docker_image.push("${IMAGE_TAG}")
                    docker_image.push('latest')
                }
            }
        }
    }
```

there is no new repository on docker hub account

sriyeshwanthe

Search by repository name All Content

sriyeshwanthe / register-app-pipeline

Contains: Image • Last pushed: about 2 hours ago

0 59 Public Scout inactive

Click on Build Now

All stages passed successfully

Dashboard > register-app-ci >

Status register-app-ci

Changes

Build Now (highlighted with a red box)

Configure

Delete Pipeline

Full Stage View

SonarQube

Rename

Pipeline Syntax

Build History

trend

Filter builds...

Stage View

	Declarative: Checkout SCM	Declarative: Tool Install	Cleanup Workspace	Cleanup from SCM	Build Application	Test Application	SonarQube Analysis	Quality Gate	Build & Push Docker Image
Average stage times: (Average full run time: ~1min 5s)	2s	254ms	463ms	1s	9s	6s	20s	409ms	12s
#10 Nov 12, 2023 7:26 AM 12:55	972ms	164ms	322ms	1s	9s	6s	19s	327ms	45s
#9 Nov 12, 2023 7:25 AM 12:48	743ms	205ms	335ms	1s	9s	6s	20s	400ms	689ms
#8 Nov 12, 2023 7:18 AM 12:39	4s	309ms	719ms	1s	10s	6s	20s	433ms	2s
#7 Nov 12, 2023 7:09 AM 12:30	4s	371ms	611ms	1s	10s	6s	20s	479ms	2s
#6 Nov 12, 2023 7:09 AM 12:20	1s	144ms	329ms	1s	10s	6s	20s	406ms	

Refresh the docker hub page we will see the register-app-pipeline image be added here

Sort by Newest Filter Tags

TAG
latest
Last pushed 30 minutes ago by sriyeshwanthe

Digest fdd2b634de80 **OS/ARCH** linux/amd64 **Last pull** 15 minutes ago **Compressed Size** 210.94 MB

Copy

TAG
1.0.0-31
Last pushed 30 minutes ago by sriyeshwanthe

Digest fdd2b634de80 **OS/ARCH** linux/amd64 **Last pull** 15 minutes ago **Compressed Size** 210.94 MB

Copy

TAG
1.0.0-30
Last pushed 31 minutes ago by sriyeshwanthe

Digest 30d50d08f936 **OS/ARCH** linux/amd64 **Last pull** — **Compressed Size** 210.94 MB

Copy

Now one more stage add for trivy scan

```
63      }
64    }
65  }
66  } Add trivy scan stage for docker hub image scan
67  stage("Trivy Scan"){
68    steps {
69      script {
70        sh ('docker run -v /var/run/docker.sock:/var/run/docker.sock aquasec/trivy image thoratsunill121/register-a
71      }
72    }
73  }
74 }
75 }
```

Use `Control + Shift + m` to toggle the `tab` key moving focus. Alternatively, use `esc` then `tab` to move to the next interactive element on the page.

this stage will remove the previously create docker image

```
72      }
73  } this stage will remove the previously created docker image
74  stage("Cleanup Artifacts") {
75    steps {
76      script {
77        sh "docker rmi ${IMAGE_NAME}:${IMAGE_TAG}"
78        SH "docker rmi ${IMAGE_NAME}:latest"
79      }
80    }
81  }
82 }
```

Setup Bootstrap Server for eksctl and Setup Kubernetes using eksctl

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Recents Quick Start

Amazon Linux	macOS	Ubuntu ubuntu	Windows	Red Hat	SUSE I >
--------------	-------	----------------------	---------	---------	-----------

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Go to AWS and create one instance with the name of EKS-Bootstrap-Server

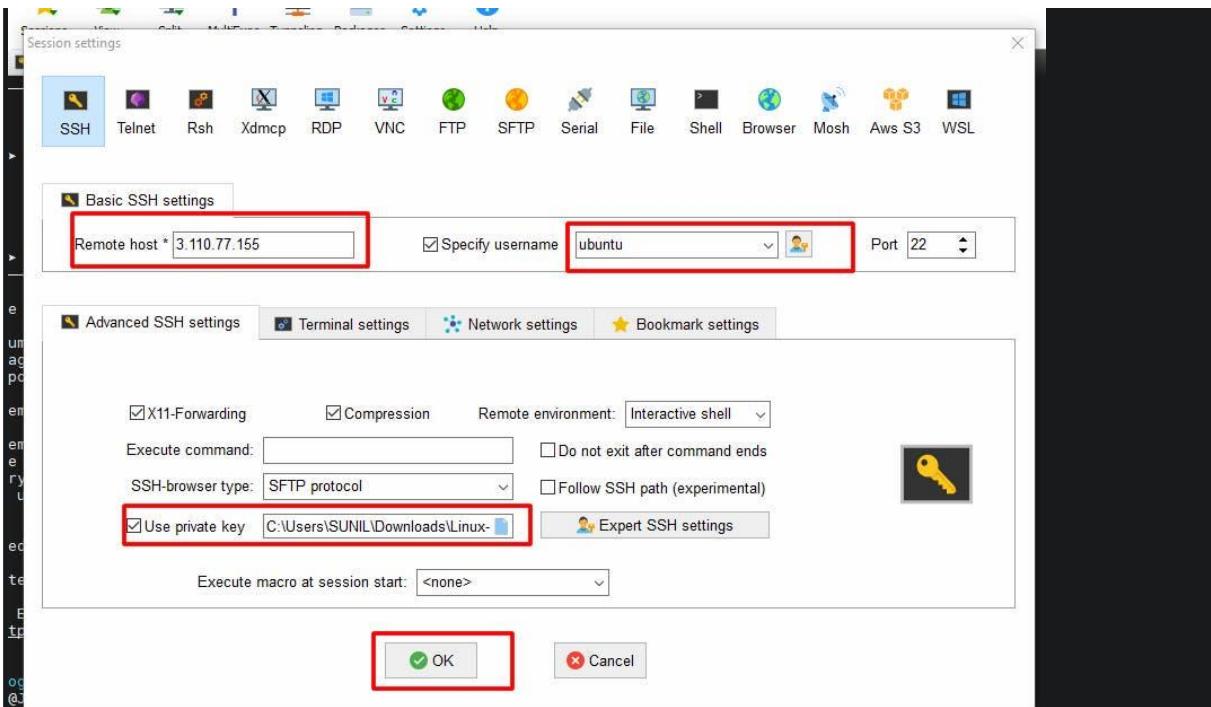
Ubuntu Image --t2.micro instance

The screenshot shows the AWS Lambda Create Function wizard. The first step, 'Function name', has 'HelloWorld' entered and is highlighted with a red box. The second step, 'Runtime', has 'Node.js 14.x' selected. The third step, 'Handler', has 'index.handler' selected. The fourth step, 'Memory size', has '128 MB' selected. The fifth step, 'Timeout', has '3 minutes' selected. The sixth step, 'Layers', is empty. The seventh step, 'Configure triggers', is also empty. The right side of the screen shows the 'Summary' section with the function name 'HelloWorld' and the ARN 'arn:aws:lambda:eu-central-1:123456789012:function>HelloWorld'. Below the summary is a large orange 'Create Function' button.

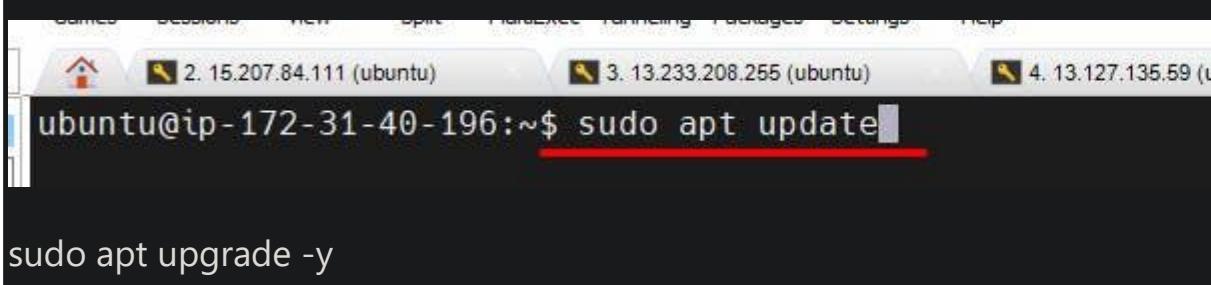
Copy Public IP of EKS-Bootstrap-Server and access it from local

The screenshot shows the AWS Lambda Instances page. It lists four instances: EKS-Bootstrap-Server, Jenkins-Agent, Jenkins-Master, and SonarQube. The EKS-Bootstrap-Server instance is selected and highlighted with a red box. Its details are shown in the main pane: Name (EKS-Bootstrap-Server), Instance ID (i-031b9823afb3fdc13), Instance state (Running), Instance type (t2.micro), and Status check (-). The Public IPv4 address is listed as 3.110.77.155. The right side of the screen shows the 'Details' tab selected for the EKS-Bootstrap-Server instance.

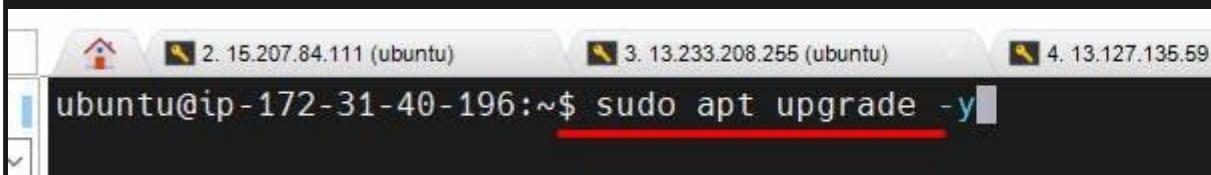
Open mobaxterm and connect instance



update instance: sudo apt update



sudo apt upgrade -y



To change hostname of this machine `sudo vim /etc/hostname`



Now reboot the instance : `sudo inti 6`

```
ubuntu@ip-172-31-40-196:~$ sudo vim /etc/hostname
ubuntu@ip-172-31-40-196:~$ sudo init 6
ubuntu@ip-172-31-40-196:~$ 
Remote side unexpectedly closed network connection

Session stopped
- Press <Return> to exit tab
- Press R to restart session
- Press S to save terminal output to file
```

After restarting install AWS cli on this server

Install AWS Cli on the above EC2

```
Refer--https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html
$ sudo su

$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"

$ apt install unzip,

$ unzip awscliv2.zip

$ sudo ./aws/install
```

```
ubuntu@EKS-Bootstrap-Server:~$ 
ubuntu@EKS-Bootstrap-Server:~$ sudo su
root@EKS-Bootstrap-Server:/home/ubuntu# cd ~
root@EKS-Bootstrap-Server:~# pwd
/root
root@EKS-Bootstrap-Server:~# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"

Now Install AWS CLI on this server with commands

root@EKS-Bootstrap-Server:~# aws --version
aws-cli/2.13.34 Python/3.11.6 Linux/6.2.0-1015-aws exe/x86_64/ubuntu.22 prompt/off
root@EKS-Bootstrap-Server:~#
```

Installing kubectl

```
Refer--https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html $  
sudo su
```

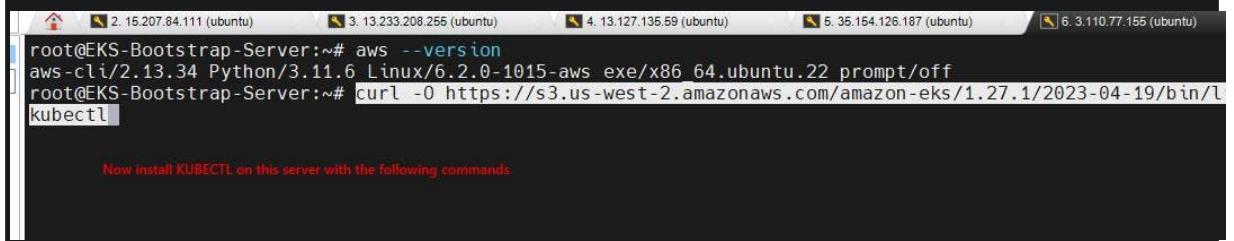
```
$ curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.1/2023-04-19/bin/linux/amd64/kubectl
```

```
$ ll ,
```

```
$ chmod +x ./kubectl //Gave executable permissions
```

```
$ mv kubectl /bin //Because all our executable files are in /bin
```

```
$ kubectl version --output=yaml
```



```
root@EKS-Bootstrap-Server:~# aws --version  
aws-cli/2.13.34 Python/3.11.6 Linux/6.2.0-1015-aws exe/x86_64.glibc.22 prompt/off  
root@EKS-Bootstrap-Server:~# curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.1/2023-04-19/bin/linux/amd64/kubectl  
  
Now install KUBECTL on this server with the following commands  
  
root@EKS-Bootstrap-Server:~# curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.1/2023-04-19/bin/linux/amd64/kubectl  
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current  
          Dload  Upload Total   Spent    Left  Speed  
100 46.9M  100 46.9M    0     0  5503k      0  0:00:08  0:00:08  ---:-- 7377k  
root@EKS-Bootstrap-Server:~# ll  
total 105924  
drwx----- 5 root root    4096 Nov 12 08:10 ./  
drwxr-xr-x 19 root root    4096 Nov 12 08:03 ../  
-rw-r--r--  1 root root   3106 Oct 15 2021 .bashrc  
-rw-r--r--  1 root root   161 Jul  9 2019 .profile  
drwx----- 2 root root    4096 Nov 12 07:54 .ssh/  
-rw-r--r--  1 root root      0 Nov 12 08:07 .sudo_as_admin_successful  
-rw-----  1 root root    751 Nov 12 08:02 .viminfo  
drwxr-xr-x  3 root root    4096 Nov 10 18:13 aws/  
-rw-r--r--  1 root root 59185475 Nov 12 08:06 awscliv2.zip  
-rw-r--r--  1 root root 49246208 Nov 12 08:10 kubectl  
drwx----- 4 root root    4096 Nov 12 07:54 snap/  
root@EKS-Bootstrap-Server:~#  
  
root@EKS-Bootstrap-Server:~# chmod +x ./kubectl  
root@EKS-Bootstrap-Server:~# mv kubectl /bin  
root@EKS-Bootstrap-Server:~# kubectl version --output=yaml  
clientVersion:  
  buildDate: "2023-04-14T20:43:13Z"  
  compiler: gc  
  gitCommit: abfec7d7e55d56346a5259c9379dea9f56ba2926  
  gitTreeState: clean  
  gitVersion: v1.27.1-eks-2f008fe  
  goVersion: go1.20.3  
  major: "1"  
  minor: 27+  
  platform: linux/amd64  
  kustomizeVersion: v5.0.1  
  
The connection to the server localhost:8080 was refused - did you specify the right host or port?  
root@EKS-Bootstrap-Server:~#
```

Installing eksctl

```
Refer--https://github.com/eksctl-io/eksctl/blob/main/README.md#installation $  
curl --silent --location
```

```
"[https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_](github.com/weaveworks/eksctl/releases/latest..)
```

```
[$(uname)](github.com/weaveworks/eksctl/releases/latest..) -s)_amd64.tar.gz" | tar xz  
-C /tmp
```

```
$ cd /tmp
```

```
$ ll
```

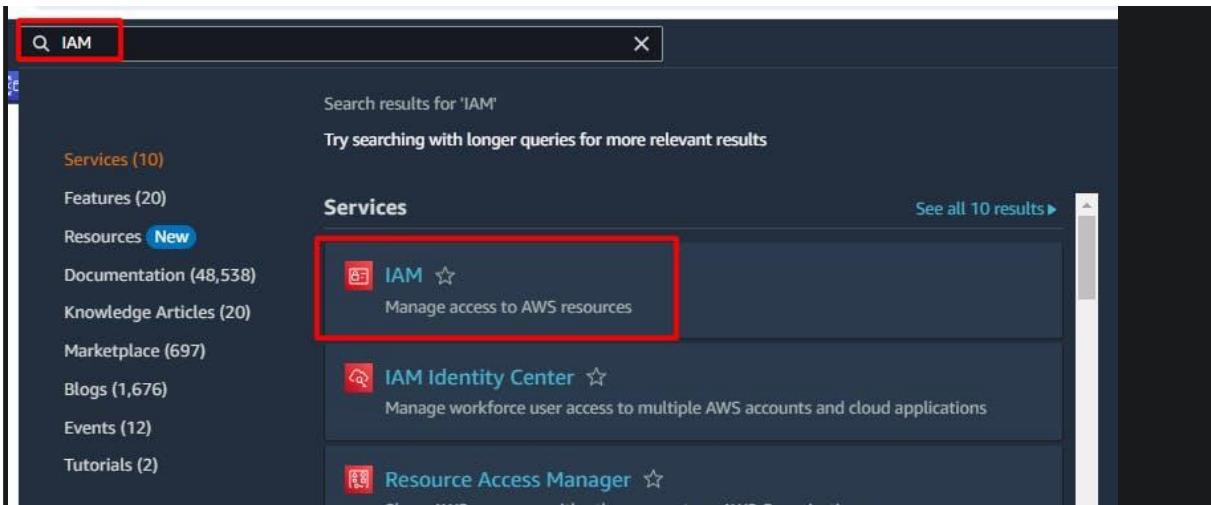
```
$ sudo mv /tmp/eksctl /bin
```

```
$ eksctl version
```

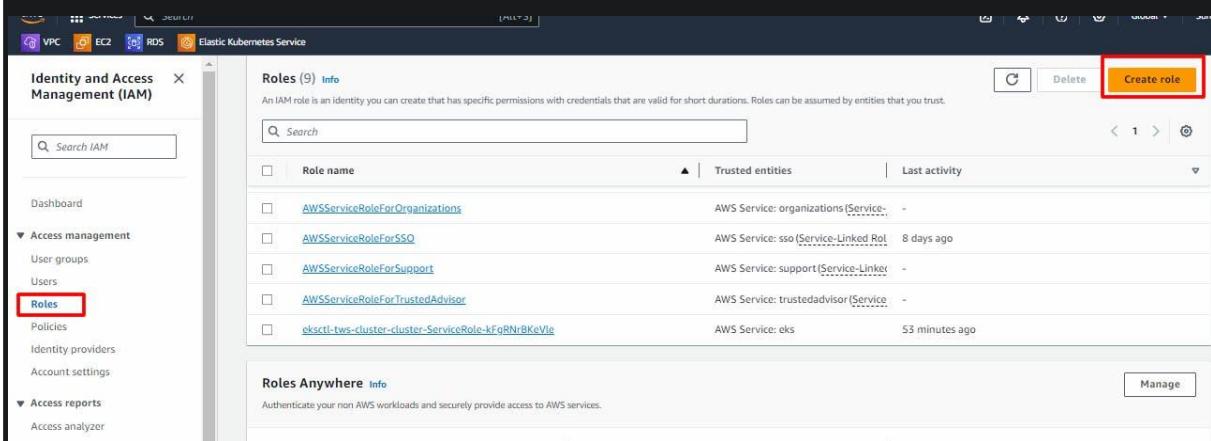
```
Now install EKSCTL on this server with following commands
```

```
root@EKS-Bootstrap-Server:~# curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp  
root@EKS-Bootstrap-Server:~# cd /tmp  
root@EKS-Bootstrap-Server:/tmp# ll  
total 141148  
drwxrwxrwt 11 root root 4096 Nov 12 08:15 ./  
drwxr-xr-x 19 root root 4096 Nov 12 08:03 ../  
drwxrwxrwt 2 root root 4096 Nov 12 08:03 .ICE-unix/  
drwxrwxrwt 2 root root 4096 Nov 12 08:03 .Test-unix/  
drwxrwxrwt 2 root root 4096 Nov 12 08:03 .X11-unix/  
drwxrwxrwt 2 root root 4096 Nov 12 08:03 .XIM-unix/  
drwxrwxrwt 2 root root 4096 Nov 12 08:03 .font-unix/  
-rwxr-xr-x 1 1001 127 144490496 Oct 27 08:39 eksctl*  
drwx----- 3 root root 4096 Nov 12 08:03 snap-private-tmp/  
drwx----- 3 root root 4096 Nov 12 08:03 systemd-private-9ff0b65d089a47238760ffeb9b6589b9-chrony.service-utbwKM  
/  
drwx----- 3 root root 4096 Nov 12 08:03 systemd-private-9ff0b65d089a47238760ffeb9b6589b9-systemd-logind.servic  
e-0UcJUX/  
drwx----- 3 root root 4096 Nov 12 08:03 systemd-private-9ff0b65d089a47238760ffeb9b6589b9-systemd-resolved.serv  
ice-DdgJX7/  
root@EKS-Bootstrap-Server:/tmp# sudo mv /tmp/eksctl /bin  
root@EKS-Bootstrap-Server:/tmp# eksctl version  
0.164.0  
root@EKS-Bootstrap-Server:/tmp#
```

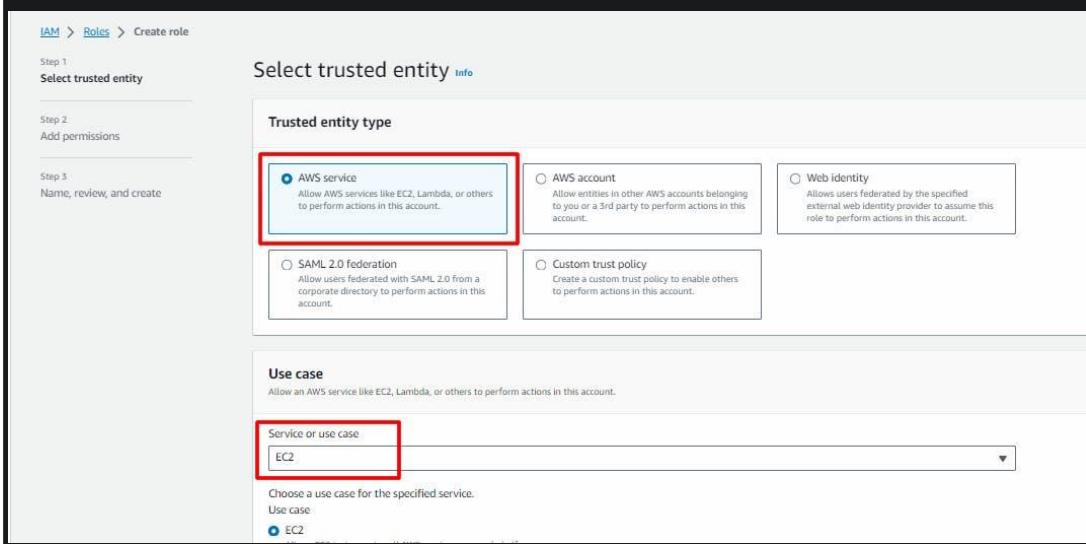
For cluster we want to create one IAM user so go to AWS and select IAM



Click on Role --> Create Role



AWS service --> EC2



Permissions Policies --> AdministratorAccess

Add permissions [Info](#)

Permissions policies (1/886) [Info](#)

Choose one or more policies to attach to your new role.

Filter by Type

Search All types

Policy name	Type
<input checked="" type="checkbox"/>  AdministratorAccess	AWS managed - job function
<input type="checkbox"/>  AdministratorAccess-Amplify	AWS managed

Roll Name: eksctl_role Click on Create role

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role. **After that click on Create Role**

eksctl_role

Maximum 64 characters. Use alphanumeric and '+,-,@,_' characters.

Description
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+,-,@,_' characters.

Go to Instance --Actions--Security--ModifyIAM user

EC2 Dashboard EC2 Global View Events Instances Instances Launch Templates Spot Requests Savings Plans Reserved Instances

Instances (1/4) [Info](#)

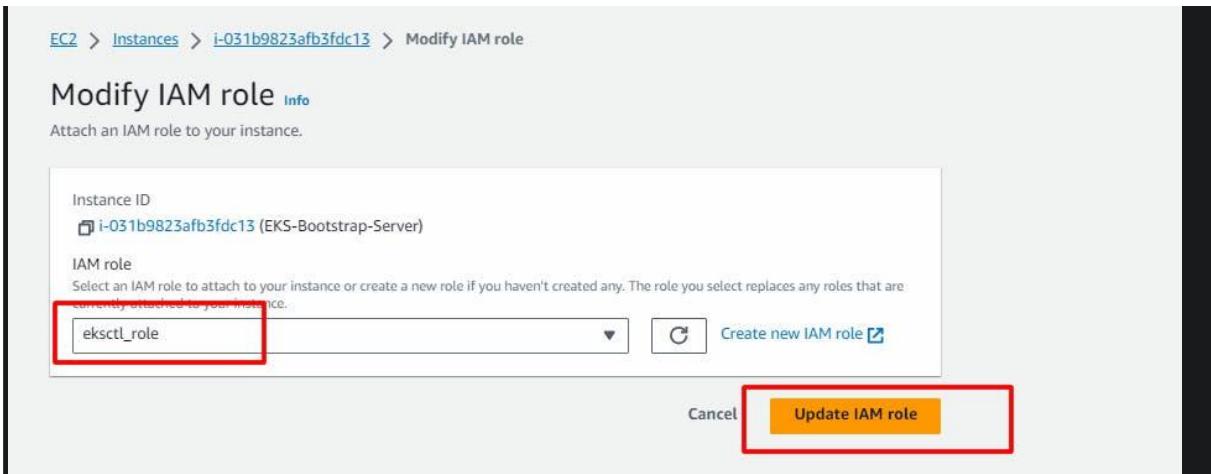
Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance ...	Status check	Alarm status
<input checked="" type="checkbox"/> EKS-Bootstrap-Server	i-031b982...	Running	t2.micro	2/2 checks passed	No alarms
<input type="checkbox"/> Jenkins-Agent	i-016dbc8...	Running	t2.micro	2/2 checks passed	No alarms
<input type="checkbox"/> Jenkins-Master	i-01c22280...	Running	t2.micro	2/2 checks passed	No alarms
<input type="checkbox"/> SonarQube	i-0e22c3b6...	Running	t3.medium	2/2 checks passed	No alarms

Actions ▾ Launch instances

Connect View details Manage instance state Instance settings Networking Security Get Windows password Modify IAM role

select eksctl_role and Update IAM role



Setup Kubernetes using eksctl

Refer--<https://github.com/aws-samples/eks-workshop/issues/734>

```
$ eksctl create cluster --name Sri1-cluster --region ap-south-1 --node-type t2.small --nodes 3 \
```

```
root@EKS-Bootstrap-Server:/# eksctl create cluster --name virtualtechbox-cluster \
> --region ap-south-1 \
> --node-type t2.small \
> --nodes 3 \
>
```

```
root@EKS-Bootstrap-Server:/# eksctl create cluster --name virtualtechbox-cluster \
> --region ap-south-1 \
> --node-type t2.small \
> --nodes 3 \
>
2023-11-12 08:36:27 [ ] eksctl version 0.164.0
2023-11-12 08:36:27 [ ] using region ap-south-1
2023-11-12 08:36:27 [ ] skipping ap-south-1c from selection because it doesn't support the following instance type(s): t2.1xlarge
2023-11-12 08:36:27 [ ] setting availability zones to [ap-south-1a ap-south-1b]
2023-11-12 08:36:27 [ ] subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.64.0/19
2023-11-12 08:36:27 [ ] subnets for ap-south-1b - public:192.168.32.0/19 private:192.168.96.0/19
2023-11-12 08:36:27 [ ] nodegroup "ng-c7f25a81" will use "" [AmazonLinux2/1.27]
2023-11-12 08:36:27 [ ] using Kubernetes version 1.27
2023-11-12 08:36:27 [ ] creating EKS cluster "virtualtechbox-cluster" in "ap-south-1" region with managed nodes
2023-11-12 08:36:27 [ ] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2023-11-12 08:36:27 [ ] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=virtualtechbox-cluster'
2023-11-12 08:36:27 [ ] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "virtualtechbox-cluster" in "ap-south-1"
2023-11-12 08:36:27 [ ] CloudWatch logging will not be enabled for cluster "virtualtechbox-cluster" in "ap-south-1"
2023-11-12 08:36:27 [ ] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-E.G. all} --region=ap-south-1 --cluster=virtualtechbox-cluster'
2023-11-12 08:36:27 [ ]
2 sequential tasks: { create cluster control plane "virtualtechbox-cluster",
  2 sequential sub-tasks: {
    wait for control plane to become ready,
    create managed nodegroup "ng-c7f25a81",
  }
}
2023-11-12 08:36:27 [ ] building cluster stack "eksctl-virtualtechbox-cluster-cluster"
2023-11-12 08:36:27 [ ] deploying stack "eksctl-virtualtechbox-cluster-cluster"
```

It will take 15 - 20 minutes

```

> --node-type t2.small \
> --nodes 3 \
>
2023-11-12 08:36:27 [::] eksctl version 0.164.0
2023-11-12 08:36:27 [::] using region ap-south-1
2023-11-12 08:36:27 [::] skipping ap-south-1c from selection because it doesn't support the following instance type(s): t2.sm...
2023-11-12 08:36:27 [::] setting availability zones to [ap-south-1a ap-south-1b]
2023-11-12 08:36:27 [::] subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.64.0/19
2023-11-12 08:36:27 [::] subnets for ap-south-1b - public:192.168.32.0/19 private:192.168.96.0/19
2023-11-12 08:36:27 [::] nodegroup "ng-c7f25a81" will use "" [AmazonLinux2/1.27]
2023-11-12 08:36:27 [::] using Kubernetes version 1.27
2023-11-12 08:36:27 [::] creating EKS cluster "virtualtechbox-cluster" in "ap-south-1" region with managed nodes
2023-11-12 08:36:27 [::] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2023-11-12 08:36:27 [::] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region ap-south-1 --cluster=virtualtechbox-cluster'
2023-11-12 08:36:27 [::] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "virtualtechbox-cluster" in "ap-south-1"
2023-11-12 08:36:27 [::] CloudWatch logging will not be enabled for cluster "virtualtechbox-cluster" in "ap-south-1"
2023-11-12 08:36:27 [::] you can enable it with 'eksctl utils update-cluster-logging --enable-types=[SPECIFY-YOUR-LOG-TYPES-Here e.g. all] --region=ap-south-1 --cluster=virtualtechbox-cluster'
2023-11-12 08:36:27 [::] 2 sequential tasks: { create cluster control plane "virtualtechbox-cluster",
2 sequential sub-tasks: {
    wait for control plane to become ready, It will take 15-20 Minutes
    create managed nodegroup "ng-c7f25a81",
}
2023-11-12 08:36:27 [::] building cluster stack "eksctl-virtualtechbox-cluster-cluster"
2023-11-12 08:36:27 [::] deploying stack "eksctl-virtualtechbox-cluster-cluster"
2023-11-12 08:36:57 [::] waiting for CloudFormation stack "eksctl-virtualtechbox-cluster-cluster"
2023-11-12 08:37:27 [::] waiting for CloudFormation stack "eksctl-virtualtechbox-cluster-cluster"
■

```

After creating creating cluster

```
kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
ip-192-168-21-115.ap-south-1.compute.internal	Ready	<none>	11m	v1.27.7-eks-4f4795d
ip-192-168-31-45.ap-south-1.compute.internal	Ready	<none>	11m	v1.27.7-eks-4f4795d
ip-192-168-62-210.ap-south-1.compute.internal	Ready	<none>	11m	v1.27.7-eks-4f4795d

ArgoCD Installation on EKS Cluster and Add EKS Cluster to ArgoCD

First, create a namespace

```
$ kubectl create namespace argocd
```

```

root@EKS-Bootstrap-Server:/# kubectl create namespace argocd
namespace/argocd created
root@EKS-Bootstrap-Server:/#

```

Next, let's apply the yaml configuration files for ArgoCd

```
$ kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
```

```
root@EKS-Bootstrap-Server:/# kubectl create namespace argocd
namespace/argocd created
root@EKS-Bootstrap-Server:/# kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-stable/manifests/install.yaml
```

Now we can view the pods created in the ArgoCD namespace.

```
$ kubectl get pods -n argocd
```

```
root@EKS-Bootstrap-Server:/# kubectl get pods -n argocd
NAME                               READY   STATUS    RESTARTS   AGE
argocd-application-controller-0     1/1     Running   0          57s
argocd-applicationset-controller-9d5cd5576-58ghc 1/1     Running   0          58s
argocd-dex-server-7bb566cc5b-zmbvr   1/1     Running   0          58s
argocd-notifications-controller-879ffb549-qnvg7 1/1     Running   0          57s
argocd-redis-b5d6bf5f5-tv76x       1/1     Running   0          57s
argocd-repo-server-5b66d87f77-czvpr 1/1     Running   0          57s
argocd-server-6fcfd9c654-6rr8b     1/1     Running   0          57s
root@EKS-Bootstrap-Server:/#
```

To interact with the API Server we need to deploy the CLI:

```
$ curl --silent --location -o
/usr/local/bin/argocd https://github.com/argoproj/argo-
cd/releases/download/v2.4.7/argocd-linux-amd64
```

```
$ chmod +x /usr/local/bin/argocd
```

```
root@EKS-Bootstrap-Server:/# curl --silent --location -o /usr/local/bin/argocd https://github.com/argoproj/argo-
cd/releases/download/v2.4.7/argocd-linux-amd64
root@EKS-Bootstrap-Server:/# chmod +x /usr/local/bin/argocd
root@EKS-Bootstrap-Server:/#
```

Expose argocd-server

```
$ kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'
```

```
root@EKS-Bootstrap-Server:/# kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'
service/argocd-server patched
root@EKS-Bootstrap-Server:/#
```

Wait about 2 minutes for the LoadBalancer creation

```
$ kubectl get svc -n argocd
```

```

root@EKS-Bootstrap-Server:/# kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'
service/argocd-server patched
root@EKS-Bootstrap-Server:/# kubectl get svc -n argocd
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   AGE
argocd-applicationset-controller   ClusterIP  10.100.203.64 <none>        4m3s
argocd-dex-server     ClusterIP  10.100.210.225 <none>        4m3s
argocd-metrics      ClusterIP  10.100.235.77  <none>        4m3s
argocd-notifications-controller-metrics ClusterIP  10.100.172.209 <none>        4m3s
argocd-redis        ClusterIP  10.100.47.118  <none>        4m3s
argocd-repo-server   ClusterIP  10.100.34.124  <none>        4m3s
argocd-server        LoadBalancer 10.100.119.171 af4fed8eca00f4afe8aef58
055ec65c9-1880344808.ap-south-1.elb.amazonaws.com 80:32663/TCP,443:31698/TCP 4m3s
argocd-server-metrics   ClusterIP  10.100.124.161  <none>        4m3s
                                         8083/TCP
root@EKS-Bootstrap-Server:/#

```

Get password and decode it.

```

$ kubectl get secret argocd-initial-admin-secret -n argocd -o yaml
$ echo QVpTbXZ0QnM1LXdNxp4WA== | base64 --decode

```

```

root@EKS-Bootstrap-Server:/# kubectl get secret argocd-initial-admin-secret -n argocd -o yaml
apiVersion: v1
data:
  password: QVpTbXZ0QnM1LXdNxp4WA==
kind: Secret
metadata:
  creationTimestamp: "2023-11-12T09:12:40Z"
  name: argocd-initial-admin-secret
  namespace: argocd
  resourceVersion: "5561"
  uid: aaf17b69-d93c-4dfc-ba4a-122e9a4a92c8
type: Opaque
root@EKS-Bootstrap-Server:/#

```

```

root@EKS-Bootstrap-Server:/# kubectl get secret argocd-initial-admin-secret -n argocd -o yaml
apiVersion: v1
data:
  password: QVpTbXZ0QnM1LXdNxp4WA==
kind: Secret
metadata:
  creationTimestamp: "2023-11-12T09:12:40Z"
  name: argocd-initial-admin-secret
  namespace: argocd
  resourceVersion: "5561"
  uid: aaf17b69-d93c-4dfc-ba4a-122e9a4a92c8
type: Opaque
root@EKS-Bootstrap-Server:/# echo QVpTbXZ0QnM1LXdNxp4WA== | base64 --decode
AZSmvtBs5-yC5zxXroot@EKS-Bootstrap-Server:/#

```

Add EKS Cluster to ArgoCD

login to ArgoCD from CLI

```
$ argocd login a2255bb2bb33f438d9addf8840d294c5-785887595.ap-south-1.elb.amazonaws.com --username admin
```

Copy this password for Argocd login

```
type: Opaque  
root@EKS-Bootstrap-Server:/# echo QVpTbXZ0QnM1LXdDNxp4WA== | base64 --decode  
AZSmvtBs5-yC5zxXroot@EKS-Bootstrap-Server:/#
```

Copy this and copy DNS name url

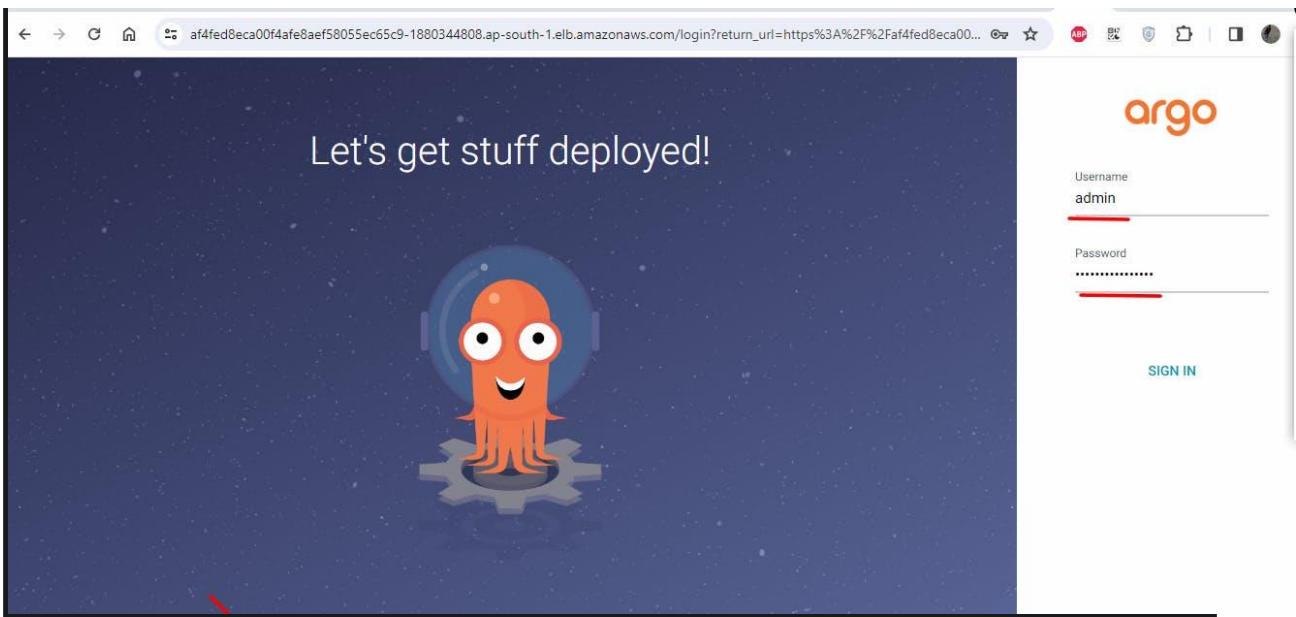
```
kubectl get svc -n argocd
```

copy argocd server url and paste it in browser for login argocd

```
type: Opaque  
root@EKS-Bootstrap-Server:/# echo QVpTbXZ0QnM1LXdDNxp4WA== | base64 --decode  
root@EKS-Bootstrap-Server:/# kubectl get svc -n argocd  
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP  
PORT(S)        AGE  
argocd-applicationset-controller   ClusterIP   10.100.203.64   <none>  
7000/TCP,8080/TCP    12m  
argocd-dex-server       ClusterIP   10.100.210.225  <none>  
5556/TCP,5557/TCP,5558/TCP    12m  
argocd-metrics         ClusterIP   10.100.235.77   <none>  
8082/TCP          12m  
argocd-notifications-controller-metrics ClusterIP   10.100.172.209  <none>  
9001/TCP          12m  
argocd-redis          ClusterIP   10.100.47.118   <none>  
6379/TCP          12m  
argocd-repo-server     ClusterIP   10.100.34.124   <none>  
8081/TCP,8084/TCP    12m  
argocd-server        LoadBalancer  10.100.119.171  af4fed8eca00f4afe8aef58055ec65c9-1880344808.ap-south-1.elb.amazonaws.com  
8083/TCP          12m  
argocd-server-metrics   ClusterIP   10.100.124.161  <none>  
root@EKS-Bootstrap-Server:/#
```

This is the login dashboard of argocd

Type username: admin and Password -> will be generate above



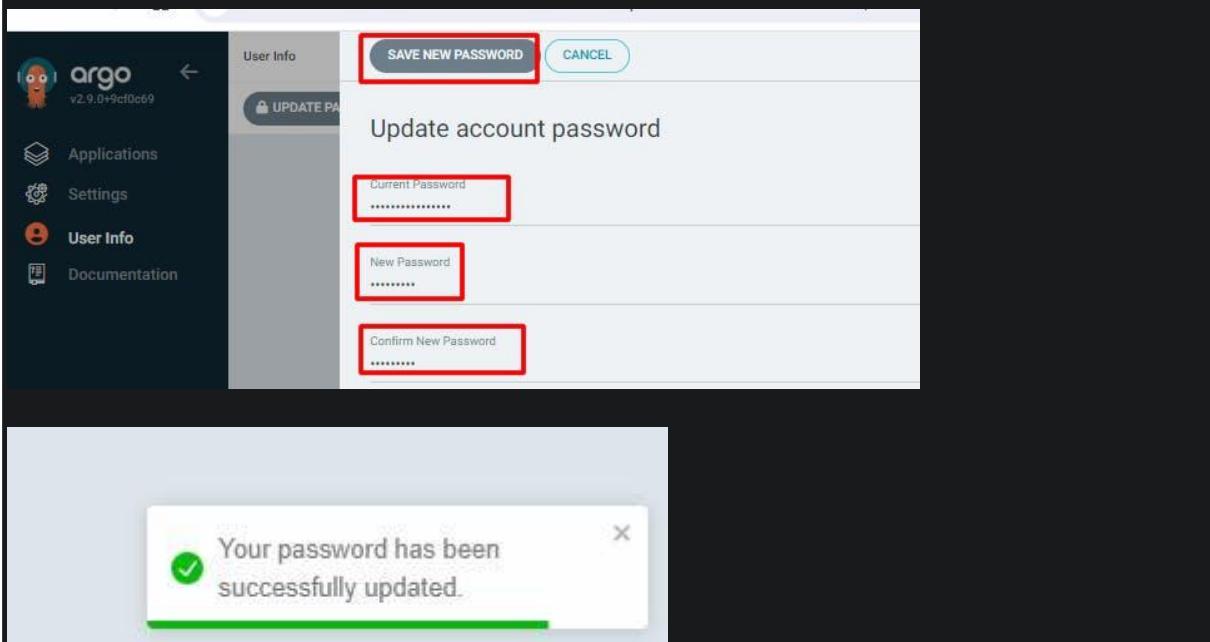
This is the Argocd dashboard

A screenshot of the Argo dashboard. The URL is af4fed8eca00f4afe8aef58055ec65c9-1880344808.ap-south-1.elb.amazonaws.com/applications. The left sidebar shows navigation links: Applications (selected), Settings, User Info (highlighted with a red box), and Documentation. The main content area has a large circular icon with three stacked layers. Below it, the text "No applications available to you just yet" and "Create new application to start managing resources in your cluster" is displayed. A "CREATE APPLICATION" button is at the bottom. The top right corner shows "APPLICATIONS TIL".

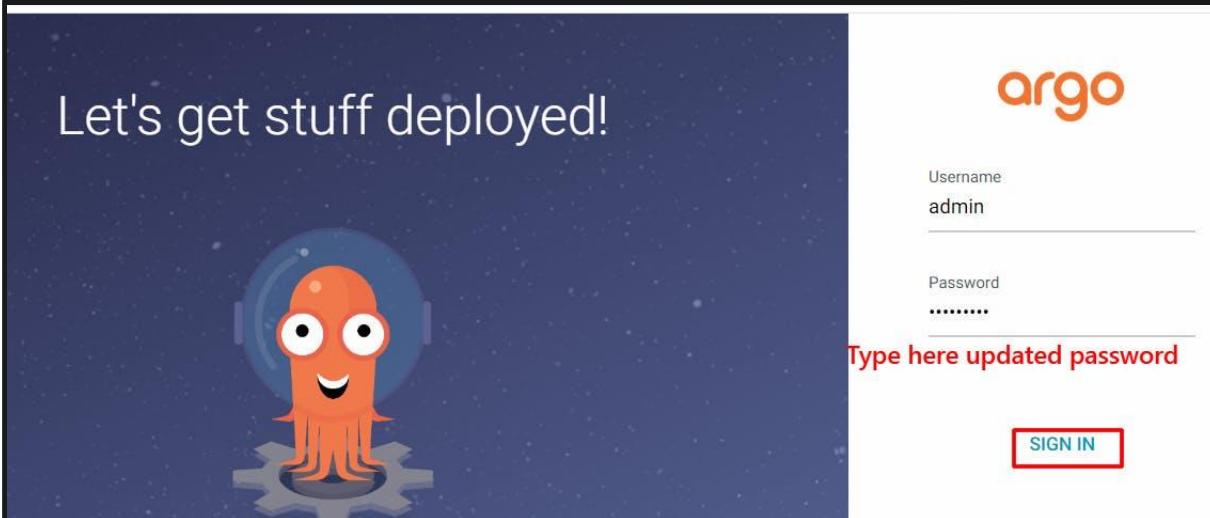
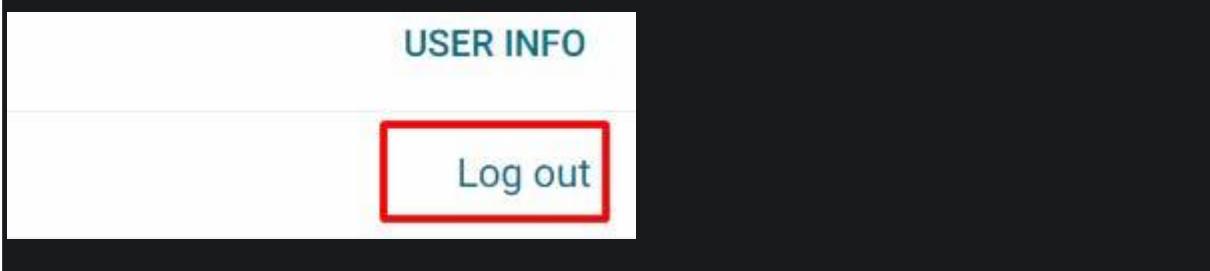
Go to User Info --> UPDATE PASSWORD

A screenshot of the User Info page. The URL is af4fed8eca00f4afe8aef58055ec65c9-1880344808.ap-south-1.elb.amazonaws.com/userinfo. The left sidebar shows "User Info" selected (highlighted with a red box). The main content area shows "User Info" and a prominent "UPDATE PASSWORD" button, which is also highlighted with a red box. To the right, a box displays "Username: admin" and "Issuer: argocd". The bottom of the sidebar also has "User Info" highlighted.

SAVE NEW PASSWORD



Logout and Login with new set password



On server type argocd login and paster url and --username admin

```

root@EKS-Bootstrap-Server:/# kubectl get svc -n argocd
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP
PORT(S)        AGE
argocd-applicationset-controller   ClusterIP   10.100.203.64 <none>
7000/TCP,8080/TCP    21m
argocd-dex-server   ClusterIP   10.100.210.225 <none>
5556/TCP,5557/TCP,5558/TCP  21m
argocd-metrics     ClusterIP   10.100.235.77 <none>
8082/TCP         21m
argocd-notifications-controller-metrics ClusterIP   10.100.172.209 <none>
9001/TCP         21m
argocd-redis       ClusterIP   10.100.47.118 <none>
6379/TCP          21m
argocd-repo-server ClusterIP   10.100.34.124 <none>
8081/TCP          21m
argocd-server      LoadBalancer 10.100.119.171 af4fed8eca00f4afe8aef58055ec65c9-1880344808.ap-south-1.elb.amazonaws.com
80:32663/TCP,443:31698/TCP  21m
argocd-server-metrics ClusterIP   10.100.124.161 <none>
8083/TCP          21m
root@EKS-Bootstrap-Server:/# argocd login af4fed8eca00f4afe8aef58055ec65c9-1880344808.ap-south-1.elb.amazonaws.com --username admin

```

Type password --> Login successfully

```

argocd-server-metrics   ClusterIP   10.100.124.161 <none>
8083/TCP              21m
root@EKS-Bootstrap-Server:/# argocd login af4fed8eca00f4afe8aef58055ec65c9-1880344808.ap-south-1.elb.amazonaws.com --username admin
WARNING: server certificate had error: x509: certificate is valid for localhost, argocd-server, argocd-server.argocd, argocd-server.argocd.svc, argocd-server.argocd.svc.cluster.local, not af4fed8eca00f4afe8aef58055ec65c9-1880344808.ap-south-1.elb.amazonaws.com. Proceed insecurely (y/n)? y
Password:
'admin:login' logged in successfully
Context 'af4fed8eca00f4afe8aef58055ec65c9-1880344808.ap-south-1.elb.amazonaws.com' updated
root@EKS-Bootstrap-Server:/#

```

Go to argocd Dashboard -- Settings -- Clusters

The screenshot shows the Argo CD web interface. On the left sidebar, the 'Settings' option is highlighted with a red box. The main content area is titled 'Settings'. Under the 'Clusters' section, which is also highlighted with a red box, there is a sub-section titled 'Configure connected Kubernetes clusters'. This section contains a table with one row, where the entire row is also highlighted with a red box. The table columns are 'NAME', 'URL', 'VERSION', and 'CONNECTION STATUS'. The single entry in the table is 'in-cluster' with URL 'https://kubernetes.default.svc' and connection status 'Unknown'.

See this is default cluster

The screenshot shows the Argo CD 'Clusters' page. The left sidebar has 'Settings' selected. The main content area is titled 'Clusters'. A table lists a single cluster entry, which is highlighted with a red box. The table columns are 'NAME', 'URL', 'VERSION', and 'CONNECTION STATUS'. The entry is 'in-cluster' with URL 'https://kubernetes.default.svc', version 'Unknown', and connection status 'Unknown'.

This will also show in terminal by typing following command

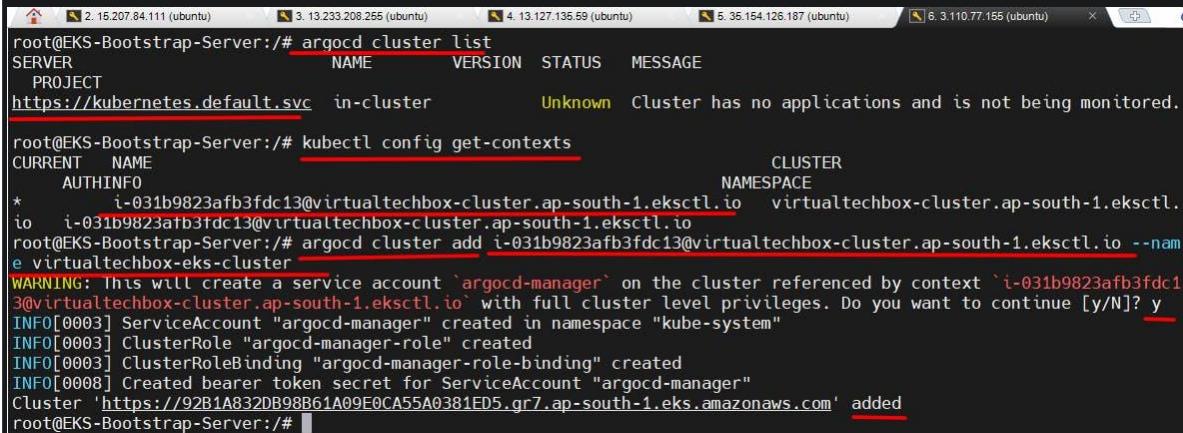
```
argocd cluster list
```

Below command will show the EKS cluster

```
$ kubectl config get-contexts
```

Add above EKS cluster to ArgoCD with below command

```
$ argocd cluster add i-08b9d0ff0409f48e7@virtualtechbox-cluster.ap-south-1.eksctl.io --name virtualtechbox-eks-cluster
```



A terminal window showing the process of adding an EKS cluster to ArgoCD. The user runs 'argocd cluster list' to check existing clusters, then 'kubectl config get-contexts' to identify the EKS context. Finally, they run 'argocd cluster add' with the context name and cluster name, followed by a confirmation prompt and success message.

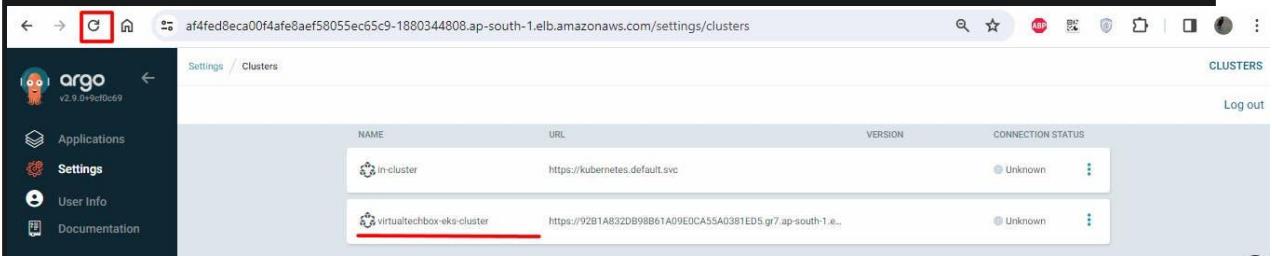
```
root@EKS-Bootstrap-Server:/# argocd cluster list
SERVER          NAME      VERSION STATUS MESSAGE
PROJECT
https://kubernetes.default.svc  in-cluster      Unknown Cluster has no applications and is not being monitored.

root@EKS-Bootstrap-Server:/# kubectl config get-contexts
CURRENT NAME                                     CLUSTER
AUTHINFO
*   i-031b9823afb3fdc13@virtualtechbox-cluster.ap-south-1.eksctl.io  virtualtechbox-cluster.ap-south-1.eksctl.io
root@EKS-Bootstrap-Server:/# argocd cluster add i-031b9823afb3fdc13@virtualtechbox-cluster.ap-south-1.eksctl.io --name virtualtechbox-eks-cluster
WARNING: This will create a service account `argocd-manager` on the cluster referenced by context `i-031b9823afb3fdc13@virtualtechbox-cluster.ap-south-1.eksctl.io` with full cluster level privileges. Do you want to continue [y/N]? y
INFO[0003] ServiceAccount "argocd-manager" created in namespace "kube-system"
INFO[0003] ClusterRole "argocd-manager-role" created
INFO[0003] ClusterRoleBinding "argocd-manager-role-binding" created
INFO[0008] Created bearer token secret for ServiceAccount "argocd-manager"
Cluster 'https://92B1A832DB98B61A09E0CA55A0381ED5.gr7.ap-south-1.eks.amazonaws.com' added
root@EKS-Bootstrap-Server:/#
```

```
$ kubectl get svc
```

```
root@EKS-Bootstrap-Server:/# argocd cluster list
SERVER          NAME      VERSION STATUS
S  MESSAGE          PROJECT
https://92B1A832DB98B61A09E0CA55A0381ED5.gr7.ap-south-1.eks.amazonaws.com  virtualtechbox-eks-cluster      Unknown
wn Cluster has no applications and is not being monitored.
https://kubernetes.default.svc          in-cluster      Unknown
wn Cluster has no applications and is not being monitored.
root@EKS-Bootstrap-Server:/#
```

Refresh the argocd page we will see the virtualtechbox eks cluster created



The ArgoCD web interface shows the 'Clusters' section. It lists two clusters: 'in-cluster' and 'virtualtechbox-eks-cluster'. Both clusters are marked as 'Unknown' in terms of connection status.

NAME	URL	VERSION	CONNECTION STATUS
in-cluster	https://kubernetes.default.svc		Unknown
virtualtechbox-eks-cluster	https://92B1A832DB98B61A09E0CA55A0381ED5.gr7.ap-south-1.e...		Unknown

Configure ArgoCD to Deploy Pods on EKS and Automate ArgoCD

Deployment Job using GitOps GitHub Repository

Go to github create CI-CD_mark1_manifest repository

SriYeshwanthE / CI-CD_mark1_manifest

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

CI-CD_mark1_manifest Public

main 1 Branch 0 Tags Go to file Add file Code

SriYeshwanthE Update deployment.yaml 24c4e32 · 1 hour ago 6 Commits

Jenkinsfile Update Jenkinsfile 2 hours ago

README.md manifest file for CI-CD_mark1 project 17 hours ago

deployment.yaml Update deployment.yaml 1 hour ago

service.yaml manifest file for CI-CD_mark1 project 17 hours ago

and connect this repository to argo cd

so go to argo cd settings -- Repositories

argo v2.9.0+9cf0c69

Applications Settings User Info Documentation

Settings

Repositories Configure connected repositories

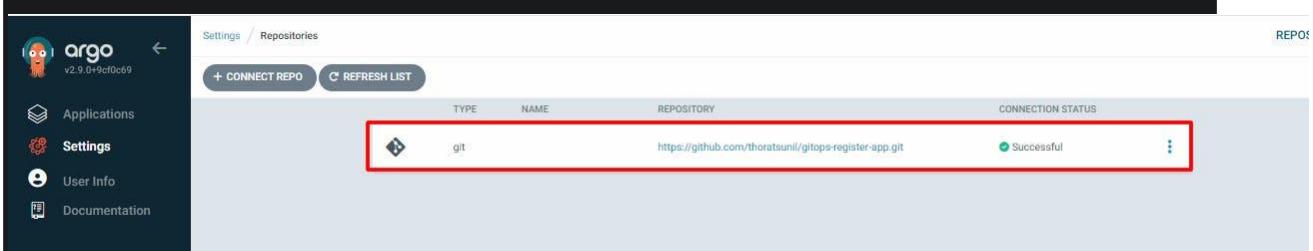
Repository certificates and known hosts Configure repository certificates and known hosts for connecting Git repositories

click --> + CONNECT REPO



Paste here url of github

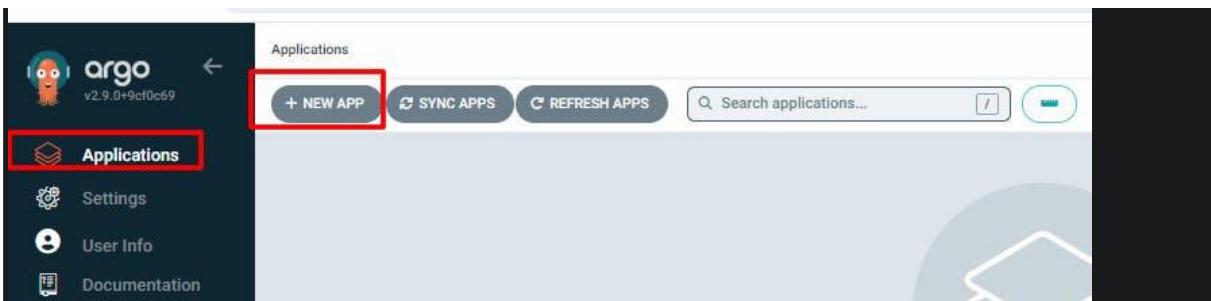
connected successfully



in the deployment.yaml file image tag should be same

The screenshot shows a code editor with 'deployment.yaml' open. The code defines a deployment named 'register-app-deployment' with 2 replicas, selecting pods labeled 'app: register-app'. It uses a template with metadata and a spec section containing a container named 'register-app' with an image tag 'thoratsunil121/register-app-pipeline:1.0.0-10'. To the right, a Docker Hub interface shows the repository 'thoratsunil121/register-app-pipeline'. It lists two tags: 'latest' (pushed 3 hours ago) and '1.0.0-10' (pushed 3 hours ago). A red arrow points from the '1.0.0-10' tag in the deployment.yaml code to the '1.0.0-10' tag in the Docker Hub interface.

Now go to argo cd Applications --> +NEW APPS



General --register-app --> project name:default

GENERAL

Application Name: register-app

Project Name: default

SYNC POLICY

Automatic

PRUNE RESOURCES

SELF HEAL

SET DELETION FINALIZER

paste here gitops-register-app git url here

SOURCE

Repository URL
<https://github.com/thoratsunil/gitops-register-app.git>

Revision
HEAD

Path
`/`

Branches ▾

DESTINATION

Cluster URL
<https://kubernetes.default.svc>

Namespace
default

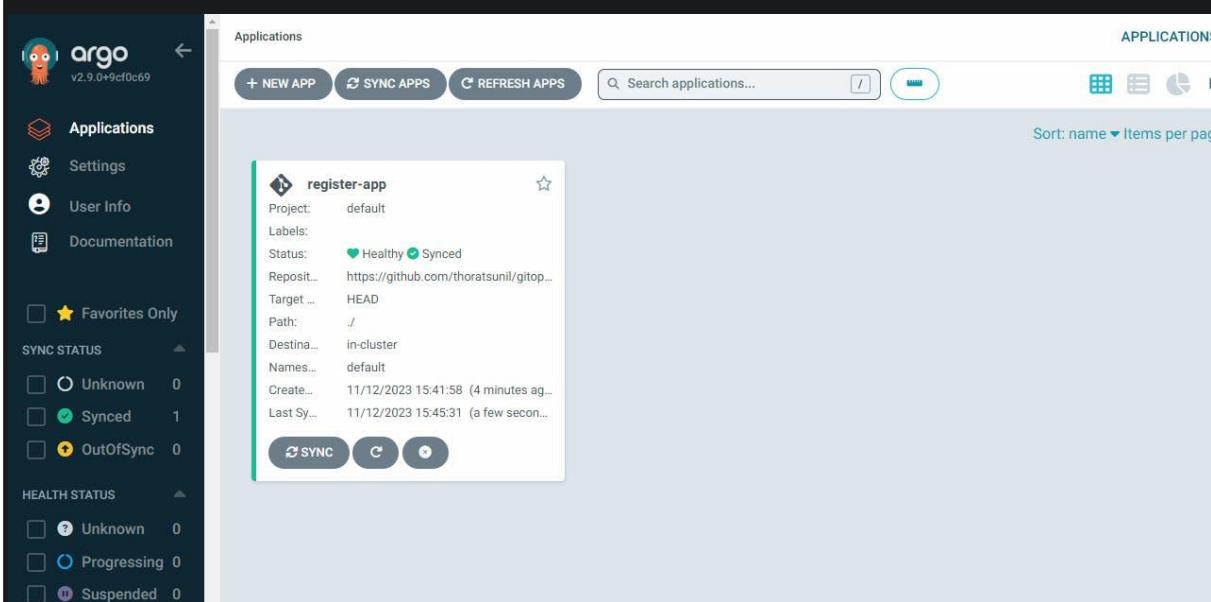
URL ▾

click on create



The screenshot shows the Argo UI interface. On the left, there's a sidebar with the Argo logo and version (v2.9.0+9cf0c69). The main area has tabs for 'Applications' (selected), 'Settings', 'User Info', and 'Documentation'. Below these are sections for 'SYNC STATUS' and 'HEALTH STATUS'. In the center, there's a 'CREATE' button highlighted with a red box, and a 'CANCEL' button next to it. To the right, there's a 'SOURCE' section.

Application are created in argocd



The screenshot shows the Argo UI Applications Tile. On the left, there's a sidebar with the Argo logo and version (v2.9.0+9cf0c69). The main area has tabs for 'Applications' (selected), 'SYNC APPS', and 'REFRESH APPS'. There's a search bar and a sort/filter button. Below these are sections for 'SYNC STATUS' and 'HEALTH STATUS'. In the center, there's a list of applications. One application, 'register-app', is highlighted with a green border. Its details are shown in a card: Project: default, Labels: Healthy Synced, Status: healthy, Repository: https://github.com/thoratsunil/gitop..., Target: HEAD, Path: /, Destination: in-cluster, Namespace: default, Create: 11/12/2023 15:41:58, Last Sync: 11/12/2023 15:45:31. Below the card are buttons for SYNC, CANCEL, and RELOAD.

Go to terminal and type `kubectl get pods`

It will show running pods

```
2. 15.207.84.111 (ubuntu) 3. 13.233.208.255 (ubuntu) 4. 13.127.135.59 (ubuntu) 5. 35.154.126.187 (ubuntu) 6. 3.110.77.155 (ubuntu)
root@EKS-Bootstrap-Server:/# kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
register-app-deployment-5c99f4c8f9-kqc2q   1/1     Running   0          62s
register-app-deployment-5c99f4c8f9-tfffm   1/1     Running   0          2m1s
root@EKS-Bootstrap-Server:/#
```

```
kubectl get svc
```

copy the url and paste it on browser for access the application

```
2. 15.207.84.111 (ubuntu) 3. 13.233.208.255 (ubuntu) 4. 13.127.135.59 (ubuntu) 5. 35.154.126.187 (ubuntu) 6. 3.110.77.155 (ubuntu)
root@EKS-Bootstrap-Server:/# kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
register-app-deployment-5c99f4c8f9-kqc2q   1/1     Running   0          62s
register-app-deployment-5c99f4c8f9-tfffm   1/1     Running   0          2m1s
root@EKS-Bootstrap-Server:/#
root@EKS-Bootstrap-Server:/# kubectl get svc
NAME              TYPE        CLUSTER-IP      EXTERNAL-IP
kubernetes        ClusterIP   <none>          10.100.0.1
                  443/TCP    95m
register-app-service   LoadBalancer  10.100.105.91  a3502ae2aa4d64acf6476c4183fe4d-1263707653.ap-south-1.elb.amazonaws.com:8080
onaws.com        8080:30905/TCP  5m4s
root@EKS-Bootstrap-Server:/#
```

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Developer Quick Start

[Tomcat Setup](#) [Realms & AAA](#) [Examples](#) [Servlet Specifications](#)
[First Web Application](#) [JDBC DataSources](#) [Tomcat Versions](#)

Managing Tomcat
For security, access to the `manager` webapp is restricted. Users are defined in:
`$CATALINA_HOME/conf/tomcat-users.xml`
In Tomcat 10.1 access to the manager application is split between different users.
[Read more...](#)

Documentation
[Tomcat 10.1 Documentation](#)
[Tomcat 10.1 Configuration](#)
[Tomcat Wiki](#)
Find additional important configuration information in:

Getting Help
[FAQ and Mailing Lists](#)
The following mailing lists are available:
[tomcat-announce](#)
Important announcements, releases, security vulnerability notifications. (Low volume).
[tomcat-users](#)

```
<curl>:8080/webapp/
```

<https://a3502ae2aa4d64acf8e6476c4183fe4d-1263707653.ap-south-1.elb.amazonaws.com:8080/webapp/>

New user Register for DevOps Learning at Youtube

Please fill in this form to create an account.

Enter Name	Enter Full Name
Enter mobile	Enter mobile number
Enter Email	Enter Email
Password	Enter Password
Repeat Password	Repeat Password

By creating an account you agree to our [Terms & Privacy](#).

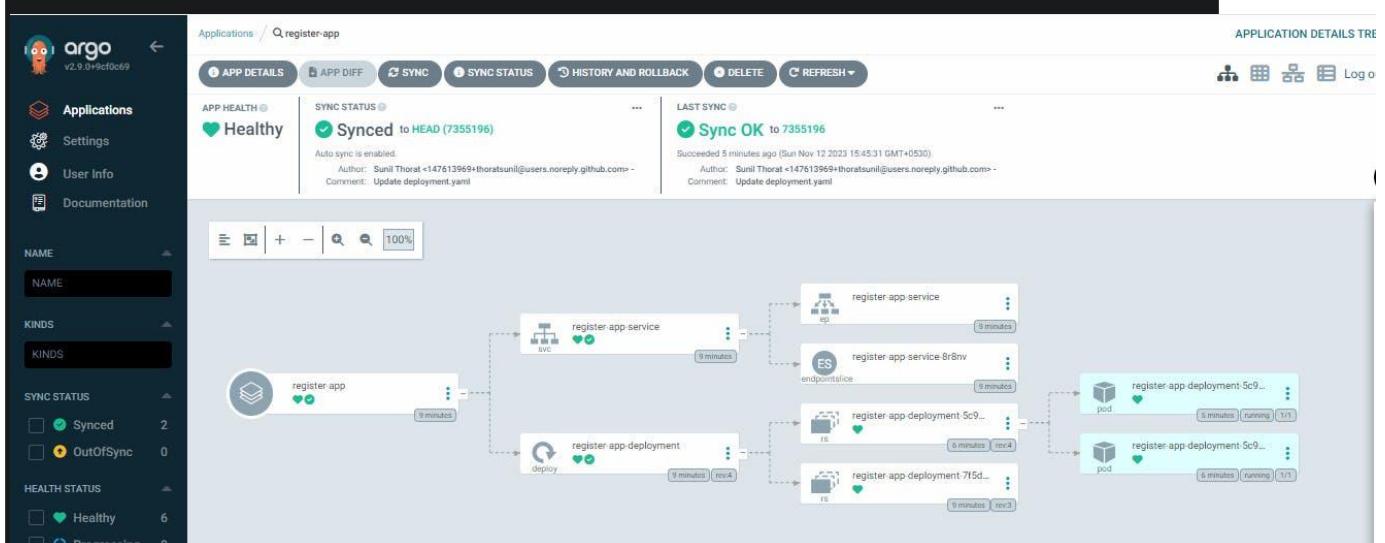
[Register](#)

Already have an account? [Sign in](#).

Thank You

Happy Learning. See You Again.

Argocd job will healthy



Now go to Jenkins and create new job "gitips-register-app-cd"

The image shows two screenshots of the Jenkins interface. The top screenshot is the Jenkins Dashboard, which includes a header with the Jenkins logo and the word "Jenkins". Below the header is a navigation bar with "Dashboard" and a right-pointing arrow. A red box highlights the "New Item" button, which is located in a white box with a plus sign and the text "New Item". To the right of the "New Item" button is a search bar with the word "All" highlighted in blue. Below the search bar are three links: "People", "Build History", and "Project Relationship". The bottom screenshot shows a modal dialog titled "Enter an item name". Inside the dialog, there is a text input field containing the text "gitops-register-app-cd", which is also highlighted with a red box. Below the input field is a note "» Required field". The dialog lists four project types: "Freestyle project", "Maven project", "Pipeline", and "Configuration project". The "Pipeline" option is highlighted with a red box. At the bottom of the dialog is a blue "OK" button, which is also highlighted with a red box.

Dashboard >

+ New Item

All

People

Build History

Project Relationship

Dashboard > All >

Enter an item name

gitops-register-app-cd

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files

Pipeline
Orchestrates long-running activities that can span multiple build and/or organizing complex activities that do not easily fit in free-

Configuration project
Builds configuration management projects that need a large number of different config

OK

General

Description

Plain text [Preview](#)

Discard old builds [?](#)

Strategy

Log Rotation

Days to keep builds
if not empty, build records are only kept up to this number of

Max # of builds to keep
if not empty, only up to this number of build records are kept

Advanced ▾

Do not allow concurrent builds



Preserve stashes from completed builds ?

This project is parameterized ?

Add Parameter ^

Filter

Boolean Parameter

Build Choice Parameter

Credentials Parameter

File Parameter

Multi-line String Parameter

Password Parameter

Run Parameter

String Parameter

Poll SCM ?

This project is parameterized ?

String Parameter ?

Name ?

IMAGE_TAG

Quiet period ?

Trigger builds remotely (e.g., from scripts) ?

Authentication Token

gitops-token

Use the following URL to trigger build remotely: JENKINS_URL/job/gitops-register-app-cd/build?token=TOKEN_NAME or /buildWithParam
Optional append &cause=Cause+Text to provide text that will be included in the recorded build cause.

Pipeline

Definition

SCM ?

Git

Repositories ?

Repository URI ?

<https://github.com/thoratsunil/gitops-register-app.git>

Credentials ?

thoratsunil/******** (GitHub)

+ Add ▾

Advanced ▾

S	W	Name	Last Success	Last Failure	Last Duration
...	...	gitops-register-app-cd	N/A	N/A	N/A
✓	...	register-app-ci	3 hr 2 min #10	3 hr 3 min #9	1 min 26 sec

Go to the AWS and copy PublicIP DNS of Jenkins-Master

Jenkins Agent	IP Address	Status	Region	Public IP
<input checked="" type="checkbox"/> Jenkins-Master	i-01c22280...	Running	t2.micro	2/2 checks passed No alarms + ap-south-1a ec2-35-154
<input type="checkbox"/> SonarQube	i-0e22c3b6...	Running	t3.medium	2/2 checks passed No alarms + ap-south-1a ec2-13-127

Instance: i-01c2228040fe71b50 (Jenkins-Master)

IPv6 address -

Instance state Running

Hostname type IP name: ip-172-31-33-81.ap-south-1.compute.internal

Private IP DNS name (IPv4 only) ip-172-31-33-81.ap-south-1.compute.internal

Public IPv4 DNS ec2-35-154-126-187.ap-south-1.compute.amazonaws.com [open address]

As well as create new stage "Trigger CD pipeline"

```

80
81      }
82      stage("Trigger CD Pipeline") {
83          steps {
84              script {
85                  sh "curl -v -k --user admin:${JENKINS_API_TOKEN} -X POST -H 'cache-control: no-cache' -H 'content-type: application/x-www-form-urlencoded' --data 'IMAG
86              }
87          }
88      }
89  
```

Go to github username-Configure

The screenshot shows the Jenkins user interface for a user named Sunil Thorat. The top navigation bar includes links for 'Builds', 'Configure', 'My Views', and 'Credentials'. The 'Configure' link is highlighted with a red box.

Create API Token

Full Name: Sunil Thorat

Description: (empty)

Plain text: [Preview](#)

API Token

Current token(s): There are no registered tokens for this user.

JENKINS_API_TOKEN	Generate	X
-------------------	----------	---

Add new Token

Credentials

Credentials are only available to the user they belong to

[Save](#) [Apply](#)

API Token

Current token(s):

JENKINS_API_TOKEN	11a93ae71b1f5ded26c4c9a69359a8eab1		
-------------------	------------------------------------	--	--

⚠ Copy this token now, because it cannot be recovered in the future.

Add new Token

Copy is only supported with a secure (HTTPS) connection

Current token(s) ?

JENKINS_API_TOKEN 11a93ae71b1f5ded26c4c9a69359a8eab1

⚠ Copy this token now, because it cannot be recovered in the future.

Copy is only supported with a secure (HTTPS) connection

□ ⚡ ⚡

Current token(s) ?

JENKINS_API_TOKEN 11a93ae71b1f5ded26c4c9a69359a8eab1

⚠ Copy this token now, because it cannot be recovered in the future.

Add new Token

Credentials

Credentials are only available to the user they belong to

Save

Apply



And this credentials in Jenkins Credentials

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

Jenkins-Agent

1 Idle

2 Idle

Clouds

Add, remove, and configure cloud instances to provision agents on-demand.

Security

Security

In-process Script Approval

Credentials

Configure global settings and paths.

Configure tools, their locations and automatic installers.

Managed files

e.g. settings.xml for maven, central managed scripts, custom files, ...

Stores scoped to Jenkins

P Store I Domains

System (global)

Add credentials

Icon: S M L

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind
Secret text

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

Secret
.....

ID ?
JENKINS_API_TOKEN

Description ?
JENKINS_API_TOKEN

Create

JENKINS_API_TOKEN credentials created

Global credentials (unrestricted) + Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching,

ID	Name	Kind	Description	
SSH	Jenkins-Agent	SSH Username with private key	Jenkins-Agent	
GitHub	thoratsunil/***** (GitHub)	Username with password	GitHub	
GitHub	jenkins-sonarqube-token	Secret text	jenkins-sonarqube-token	
GitHub	dockerhub	Username with password	dockerhub	
GitHub	<u>JENKINS_API_TOKEN</u>	Secret text	JENKINS_API_TOKEN	

Now add these credentials in environment variables of jenkins file on github

SriYeshwanthE / CI-CD_mark1

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

CI-CD_mark1 (Public)

main 1 Branch 0 Tags Go to file Add file <> Code

SriYeshwanthE Update and rename README.md to README.txt d391ebf · 52 minutes ago 18 Commits

gitops-register-app added manifest file 18 hours ago

server mark1 2 days ago

webapp Update index.jsp 1 hour ago

Dockerfile added manifest file 18 hours ago

Jenkinsfile Update Jenkinsfile 1 hour ago

README.txt Update and rename README.md to README.txt 52 minutes ago

pom.xml added manifest file 18 hours ago

```
}

environment {
    APP_NAME = "register-app-pipeline"
    RELEASE = "1.0.0"
    DOCKER_USER ="thoratsunil121"
    DOCKER_PASS ="dockerhub"
    IMAGE_NAME = "${DOCKER_USER}" + "/" + "${APP_NAME}"
    IMAGE_TAG = "${RELEASE}-${BUILD_NUMBER}"
    JENKINS_API_TOKEN = credentials("JENKINS_API_TOKEN")
}

stages {
    stage("Cleanup Workspace"){


```

SriYeshwanthE / CI-CD_mark1_manifest

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

CI-CD_mark1_manifest (Public)

main 1 Branch 0 Tags Go to file Add file <> Code

SriYeshwanthE Update deployment.yaml 24c4e32 · 1 hour ago 6 Commits

Jenkinsfile Update Jenkinsfile 2 hours ago

README.md manifest file for CI-CD_mark1 project 17 hours ago

deployment.yaml Update deployment.yaml 1 hour ago

service.yaml manifest file for CI-CD_mark1 project 17 hours ago

```

1 Pipeline {
2   agent { label "Jenkins-Agent" }
3   environment {
4     APP_NAME = "register-app-pipeline"
5   }
6
7   stages {
8     stage("Cleanup workspace") {
9       steps {
10         cleanWs()
11     }
12   }
13
14   stage("Checkout from SCM") {
15     steps {
16       git branch: 'master', credentialsId: 'github', url: 'https://github.com/choratsunil/gitops-register-app'
17     }
18   }
19
20   stage("Update the Deployment Tags") {
21     steps {
22       sh """
23         cat deployment.yaml
24         sed -i 's/${APP_NAME}/${DEPLOY_TAG}/g' deployment.yaml
25         cat deployment.yaml
26         """
27     }
28   }
29
30   stage("Push the changed deployment file to Git") {
31     steps {
32       sh """
33         git config --global user.name "choratsunil"
34         git config --global user.email "vistaneelserice@gmail.com"
35         git add deployment.yaml
36         git commit -m "Updated Deployment Manifest"
37         ...
38         withCredentials([gitUsernamePassword(credentialsId: 'github', gitToolName: 'Default')]) {
39           sh "git push https://github.com/choratsunil/gitops-register-app main"
40         }
41       """
42     }
43   }
44 }

```

Documentation • Share feedback

Verify CI/CD Pipeline by Doing Test Commit on GitHub Application Repo

Now go to register-app-ci ---> configure

S	W	Name	Last Success	Last Failure	Last Duration
		gitops-register-app-ci	9 min 44 sec #5	17 min #4	5.2 sec
		register-app-ci	10 hr #10	49 min #14	1 min 26 sec

Icon: S M L Icon legend Atom feed for all Atom feed for failures Atom feed for just late

Status
 register-app-ci

</> Changes:
▷ Build Now

Configure

Delete Pipeline

Stage View

Build Triggers --> Poll SCM --> schedule

Dashboard > register-app-ci > Configuration

Throttle builds ?

Configure

Build Triggers

General

Advanced Project Options

Pipeline

Build after other projects are built ?

Build periodically ?

Build whenever a SNAPSHOT dependency is built ?

GitHub hook trigger for GITScm polling ?

Monitor Docker Hub/Registry for image changes ?

Poll SCM ?

Schedule ?

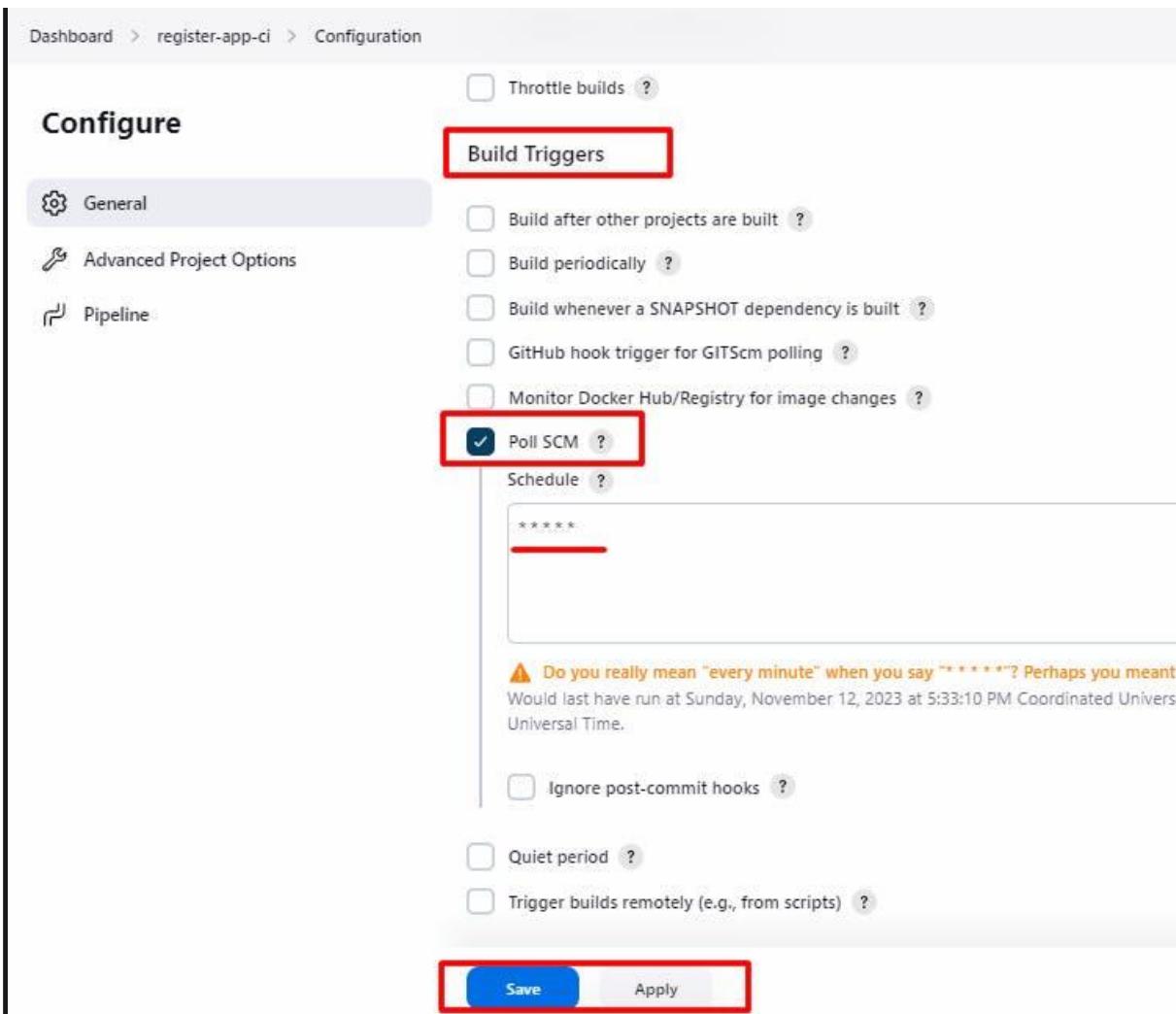
⚠ Do you really mean "every minute" when you say "***"? Perhaps you meant**
Would last have run at Sunday, November 12, 2023 at 5:33:10 PM Coordinated Universal Time.

Ignore post-commit hooks ?

Quiet period ?

Trigger builds remotely (e.g., from scripts) ?

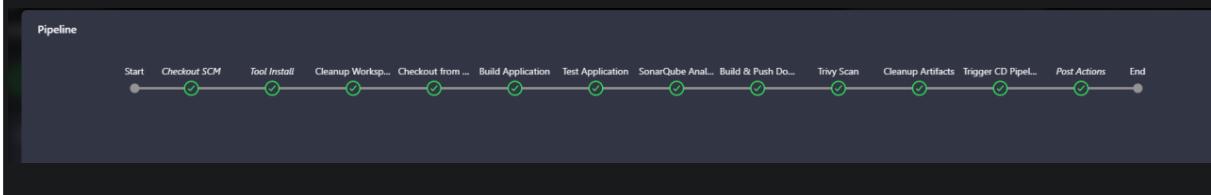
Save **Apply**



Now click on Build Now all stages will be completed (More Issues will occur with this stage search on the internet for a solution)

If I make some changes in the GitHub repository it will automatically trigger the CI job and after completing CI Job CD Job will be Trigger successfully and our changes will be made to our application.

CI Job will run successfully



And CD Job also ran successfully

Application Changed with the CI CD Pipeline

New user Register Test project

Please fill in this form to create an account

Enter Name [Enter Full Name]
Enter mobile [Enter mobile number]
Enter Email [Enter Email]
Password [Enter Password]
Repeat Password [Repeat Password]

By creating an account you agree to our [Terms & Privacy](#).

[Register](#)

Already have an account? [Sign in](#).

Thank You

hope we get selected .

Argocd job also healthy

register-app

APP DETAILS APP DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH

APP HEALTH Healthy

SYNC STATUS Synced to HEAD (762ffc3)

Last Sync Sync OK to 762ffc3

Sync OK to 762ffc3

Synced to HEAD (762ffc3)

Auto sync is enabled.

Author: Sunil Thorat <147613969+thoratsunil@users.noreply.github.com>

Comment: Update deployment.yaml

Successed a few seconds ago (Mon Nov 13 2023 02:12:42 GMT+0530)

Author: thoratsunil <sunilmallihub@gmail.com>

Comment: Updated Deployment Manifest

NAME NAME

KINDS KINDS

SYNC STATUS Synced 0 OutOfSync 0

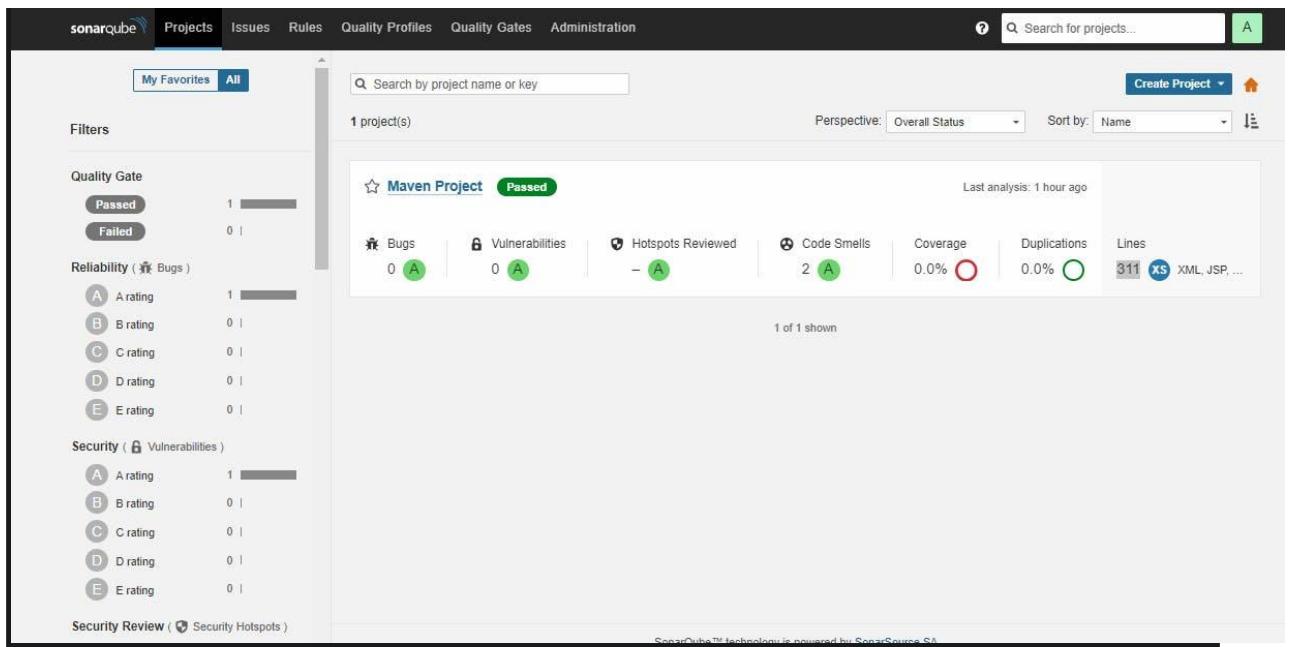
HEALTH STATUS Healthy 10 Progressing 0 Degraded 0 Suspended 0 Missing 0 Unknown 0

register app-service

register app-deployment

register app-deployment-pod

SonarQube Scanning also green



Done Seccessfully ...!!!