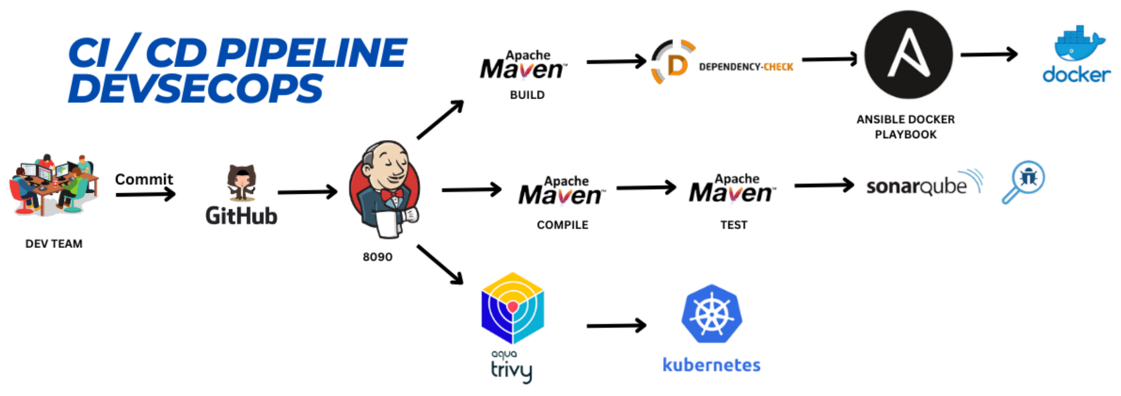
Ansible – DevSecOps Petshop project | Jenkins CI/CD



Hello friends, we will be deploying a Petshop Java Based Application. This is an everyday use case scenario used by several organizations. We will be using Jenkins as a CICD tool and deploying our application on a Docker container and Kubernetes cluster. Hope this detailed blog is useful.

We will be deploying our application in two ways, one using Docker Container and the other using K8S cluster.

Project Repo: github.com/Aj7Ay/jpetstore-6.git

STEPS:

Step 1 -- Create an Ubuntu(22.04) T2 Large Instance

Step 2 -- Install Jenkins, Docker and Trivy

Step 3 -- Install Plugins like JDK, Sonarqube Scanner, Maven, OWASP Dependency Check

Step 4 -- Configure Sonar Server in Manage Jenkins

Step 5 -- Install OWASP Dependency Check Plugins

Step 6 -- Docker plugin and credential Setup

Step 7 -- Adding Ansible Repository in Ubuntu and install Ansible

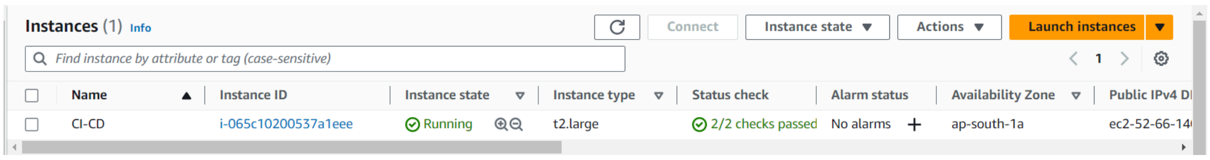
Step 8 -- Kuberenetes Setup

Step 9 -- Master-slave Setup for Ansible and Kubernetes

Now, let's get started and dig deeper into each of these steps:-

STEP1:Create an Ubuntu(22.04) T2 Large Instance

Launch an AWS T2 Large Instance. Use the image as Ubuntu. You can create a new key pair or use an existing one. Enable HTTP and HTTPS settings in the Security Group and open all ports (not best case to open all ports but just for learning purposes it's okay).



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DI
CI-CD	i-065c10200537a1eee	Running	t2.large	2/2 checks passed	No alarms	ap-south-1a	ec2-52-66-14

Step 2 — Install Jenkins, Docker and Trivy

2A — To Install Jenkins

Connect to your console, and enter these commands to Install Jenkins

```
vi jenkins.sh

#!/bin/bash

sudo apt update -y

#sudo apt upgrade -y

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

sudo apt update -y

sudo apt install temurin-17-jdk -y

/usr/bin/java --version

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \

            /usr/share/keyrings/jenkins-keyring.asc > /dev/null

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \

            https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

                    /etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt-get update -y

sudo apt-get install jenkins -y

sudo systemctl start jenkins

sudo systemctl status jenkins

sudo chmod 777 jenkins.sh

./jenkins.sh
```

Once Jenkins is installed, you will need to go to your AWS EC2 Security Group and open Inbound Port 8080 and 8090, 9000 for sonar, since Jenkins works on Port 8080.

But for this Application case, we are running Jenkins on another port. so change the port to 8090 using the below commands.

sudo systemctl stop jenkins

sudo systemctl status jenkins

cd /etc/default

sudo vi jenkins   #chnage port HTTP_PORT=8090 and save and exit

cd /lib/systemd/system

sudo vi jenkins.service #change Environments="Jenkins_port=8090" save and exit

sudo systemctl daemon-reload

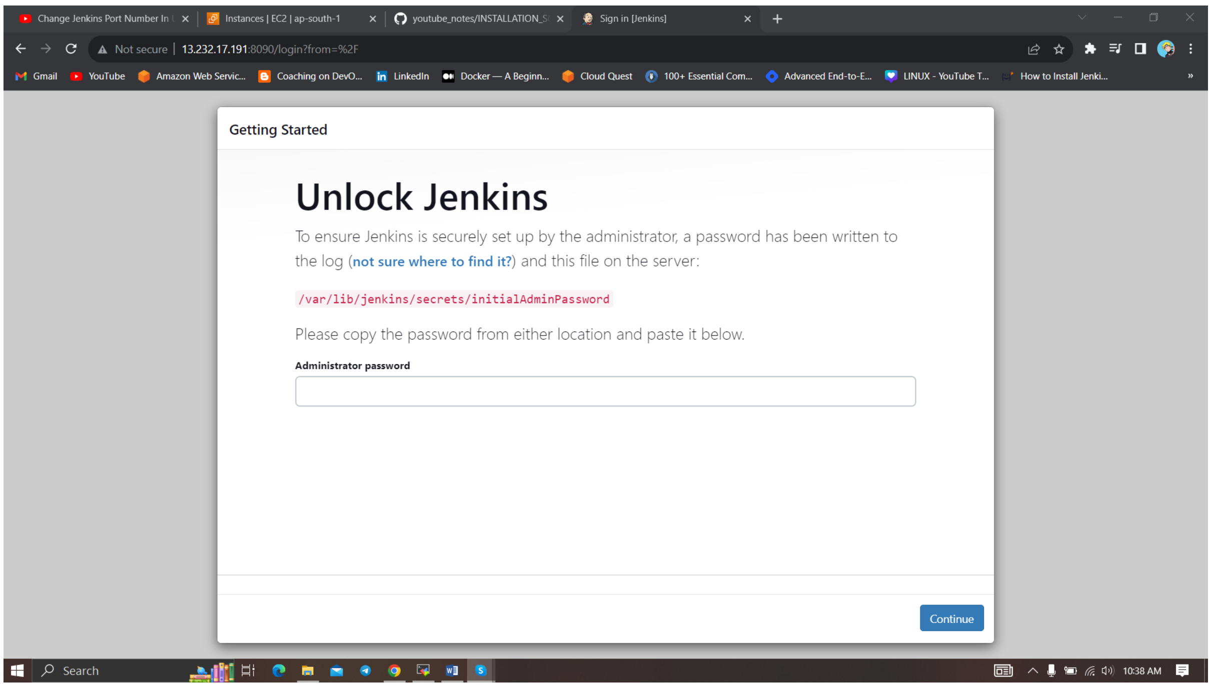
sudo systemctl restart jenkins

sudo systemctl status jenkins

Now, grab your Public IP Address

EC2 Public IP Address:8090

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



Getting Started

Unlock Jenkins

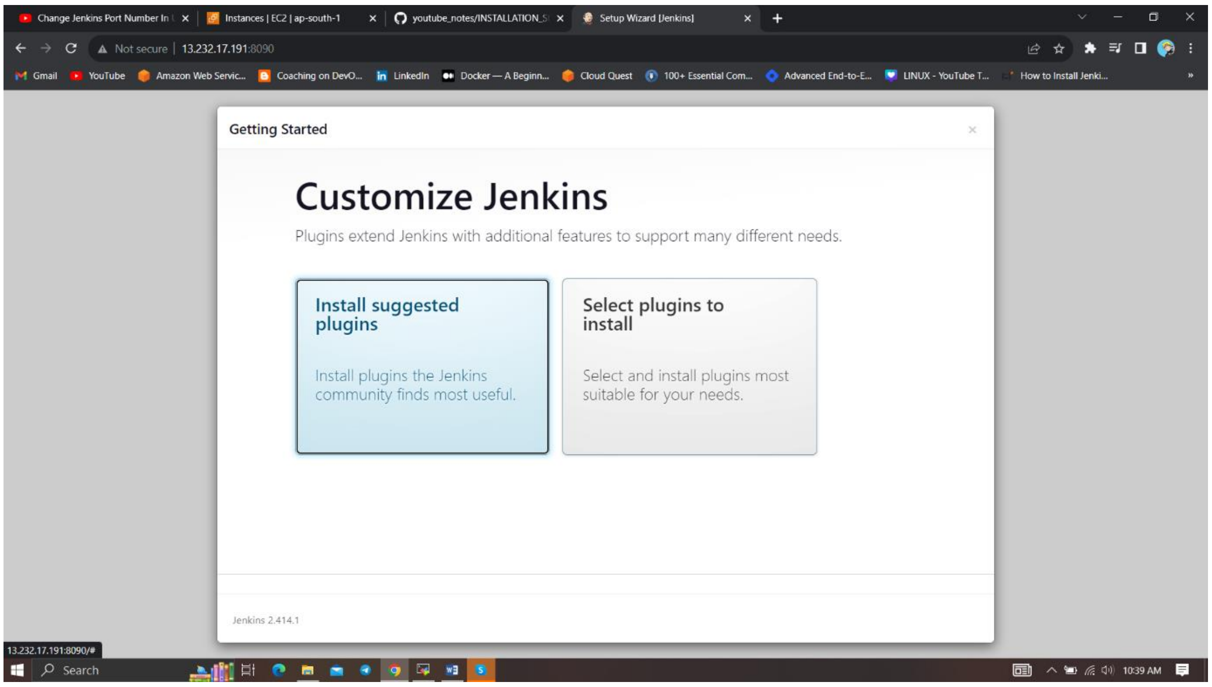
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

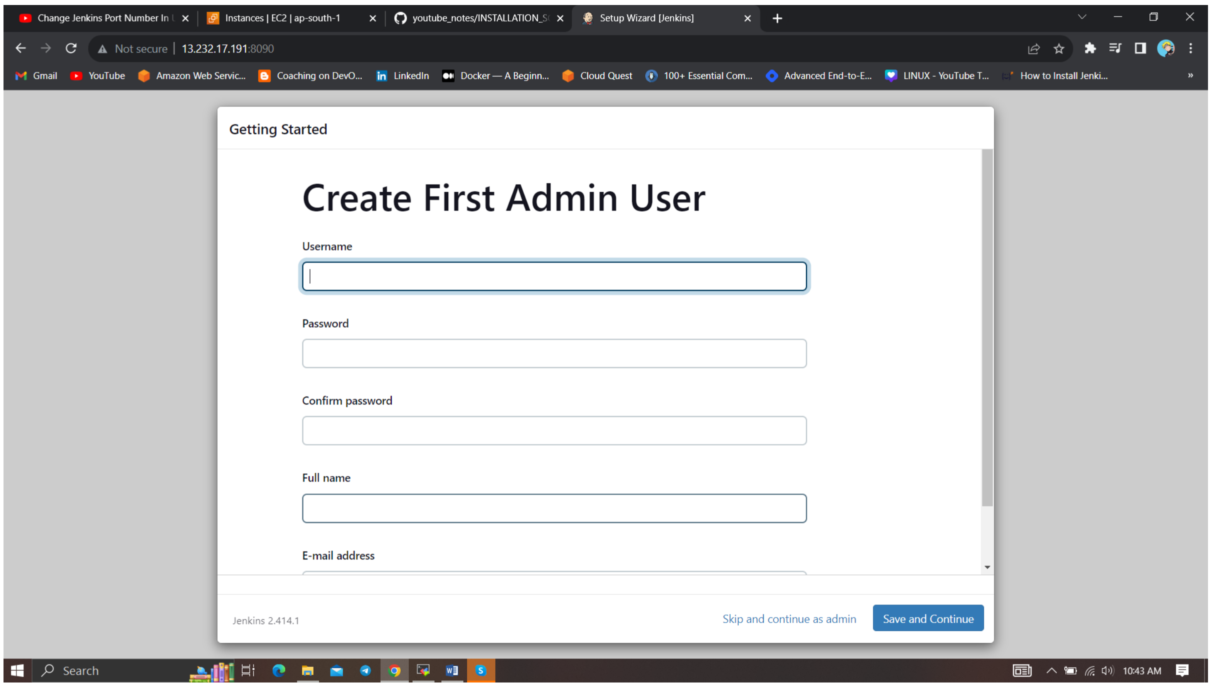
Please copy the password from either location and paste it below.

Administrator password

Unlock Jenkins using an administrative password and install the suggested plugins.

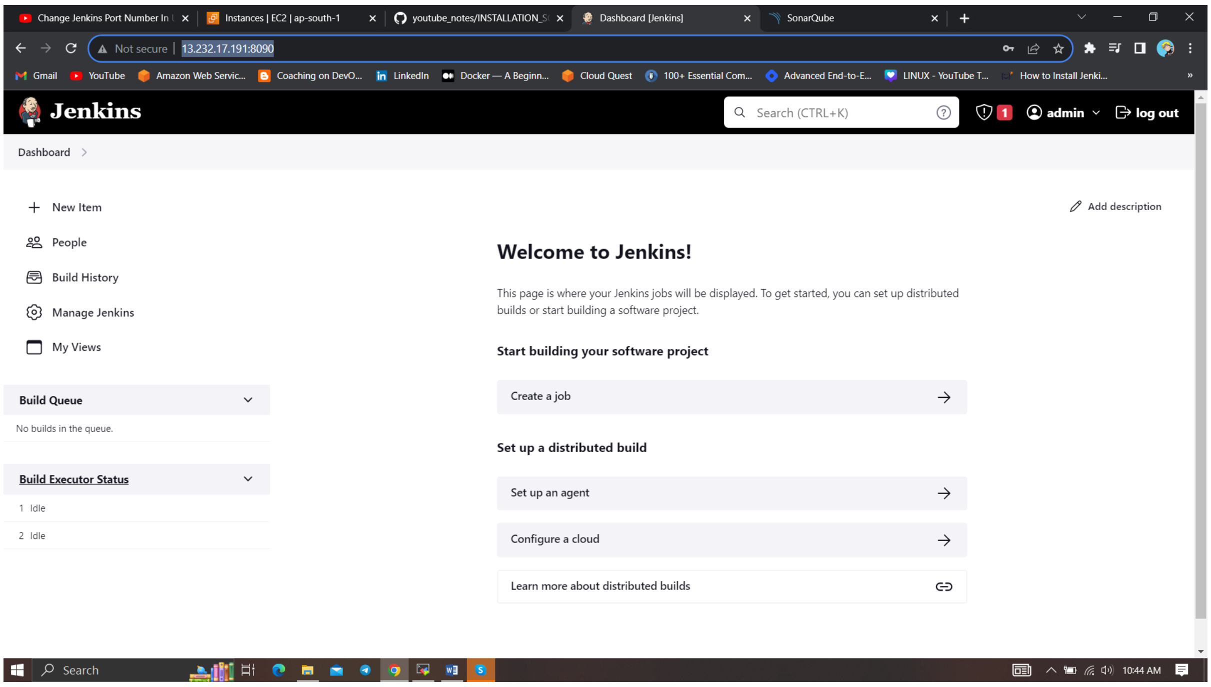

Getting Started
Customize Jenkins
Plugins extend Jenkins with additional features to support many different needs.
Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.
Jenkins 2.414.1

Jenkins will now get installed and install all the libraries.



Create a user click on save and continue.

Jenkins Getting Started Screen.



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

2B — Install Docker

sudo apt-get update

sudo apt-get install docker.io -y

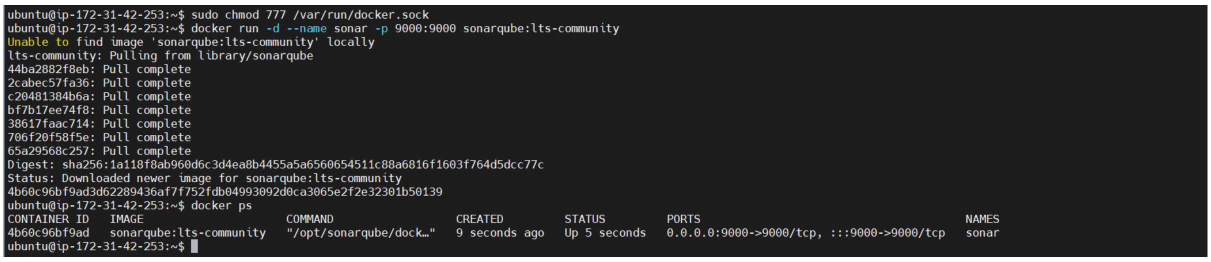
sudo usermod -aG docker $USER

newgrp docker

sudo chmod 777 /var/run/docker.sock

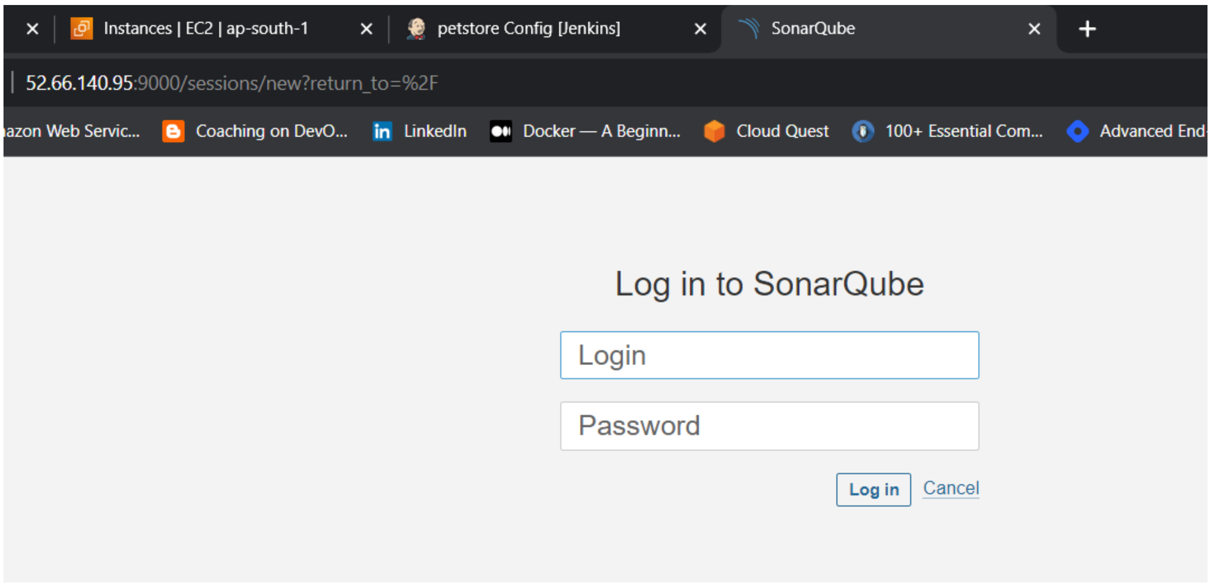
After the docker installation, we create a sonarqube container (Remember added 9000 ports in the security group).

docker run -d --name sonar -p 9000:9000 sonarqube:lts-community



```
ubuntu@ip-172-31-42-253:~$ sudo chmod 777 /var/run/docker.sock
ubuntu@ip-172-31-42-253:~$ docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
44ba2882f8eb: Pull complete
2cabec57fa36: Pull complete
c20481384b6a: Pull complete
bf7b17ee74f8: Pull complete
38617faac714: Pull complete
706f20f58f5e: Pull complete
65a29568c257: Pull complete
Digest: sha256:1a118f8ab960d6c3d4ea8b4455a5a6560654511c88a6816f1603f764d5dcc77c
Status: Downloaded newer image for sonarqube:lts-community
4b60c96bf9ad3d62289436af7f752fdb04993092d0ca3065e2f2e32301b50139
ubuntu@ip-172-31-42-253:~$ docker ps
CONTAINER ID   IMAGE                     COMMAND                  CREATED         STATUS         PORTS                                       NAMES
4b60c96bf9ad   sonarqube:lts-community   "/opt/sonarqube/dock…"   9 seconds ago   Up 5 seconds   0.0.0.0:9000->9000/tcp, :::9000->9000/tcp   sonar
ubuntu@ip-172-31-42-253:~$
```

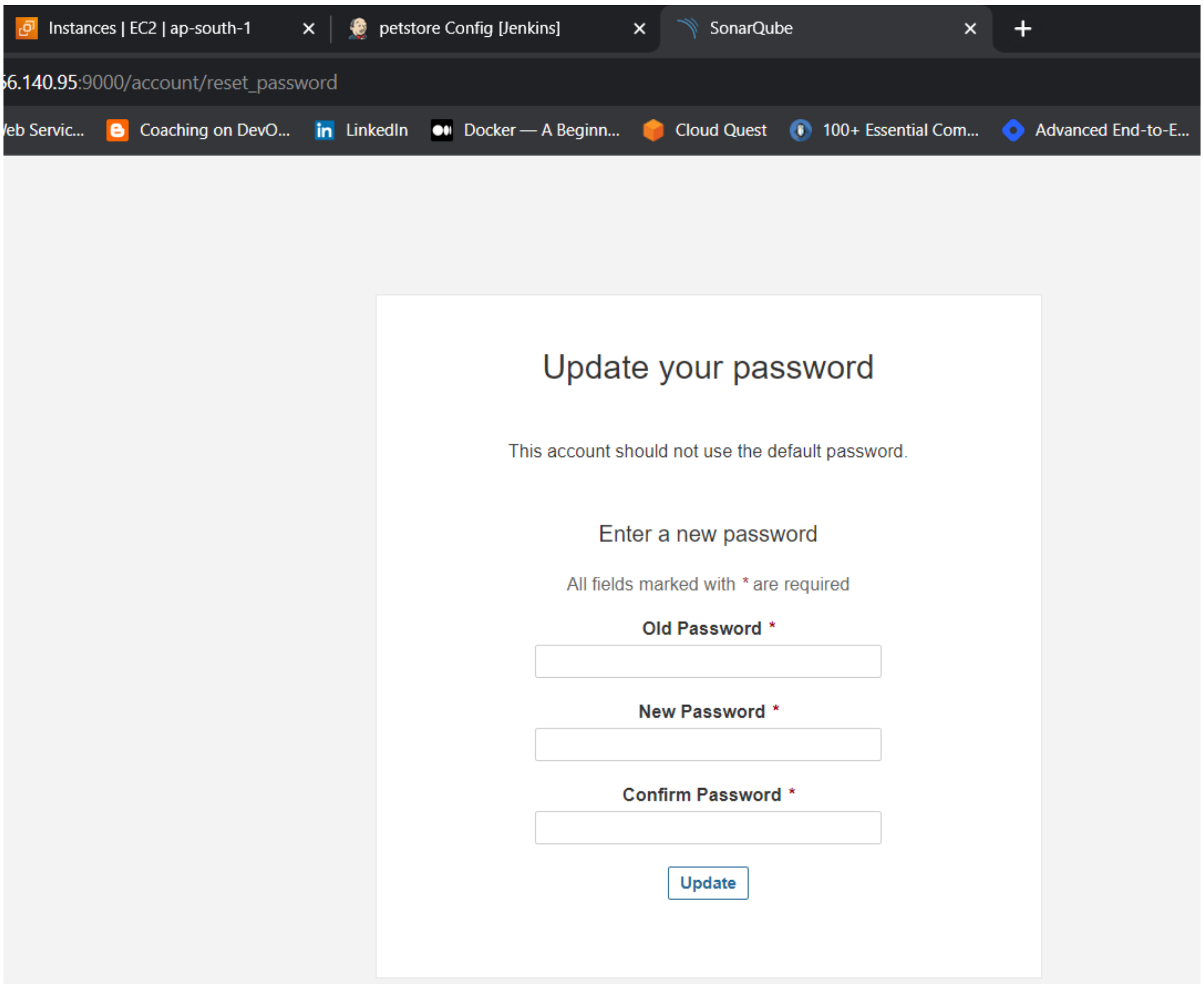
Now our sonarqube is up and running



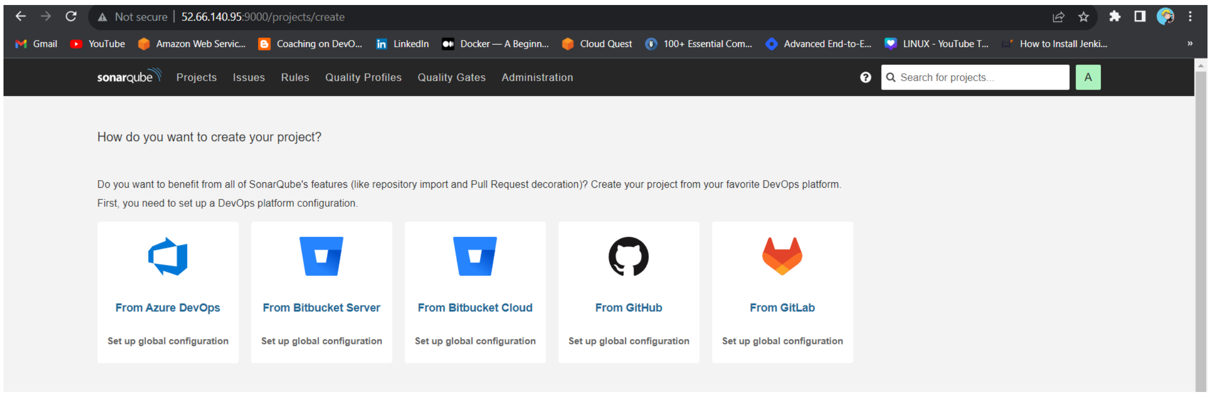
Enter username and password, click on login and change password

username admin

password admin



Update New password, This is Sonar Dashboard.


Not secure | 52.66.140.95:9000/projects/create
sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration
Search for projects...

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

2C — Install Trivy

vi trivy.sh

sudo apt-get install wget apt-transport-https gnupg lsb-release -y

wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null

echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.list

sudo apt-get update

sudo apt-get install trivy -y

Next, we will log in to Jenkins and start to configure our Pipeline in Jenkins

Step 3 — Install Plugins like JDK, Sonarqube Scanner, Maven, OWASP Dependency Check

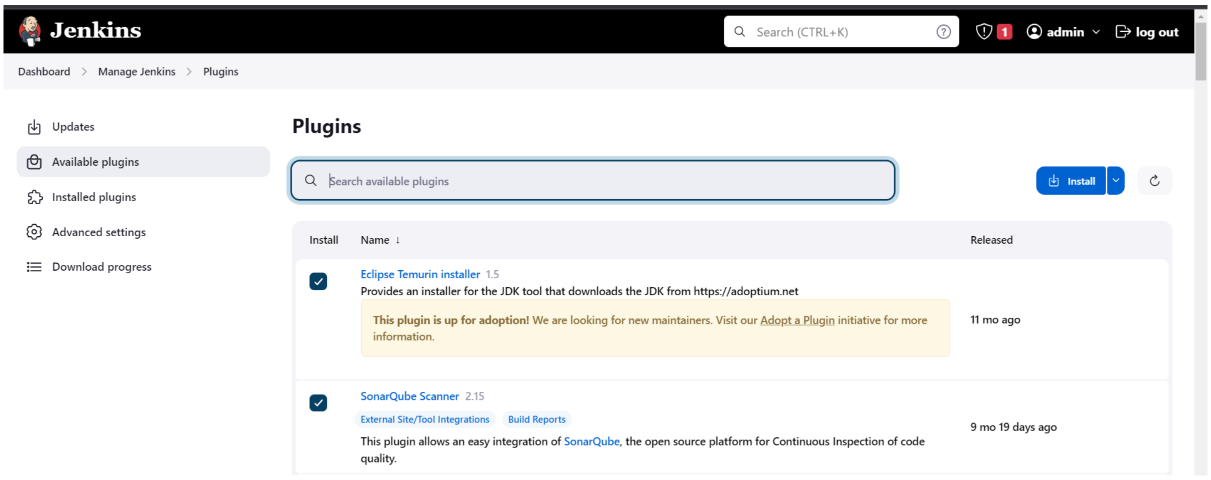
3A — Install Plugin

Goto Manage Jenkins →Plugins → Available Plugins →

Install below plugins

1 → Eclipse Temurin Installer (Install without restart)

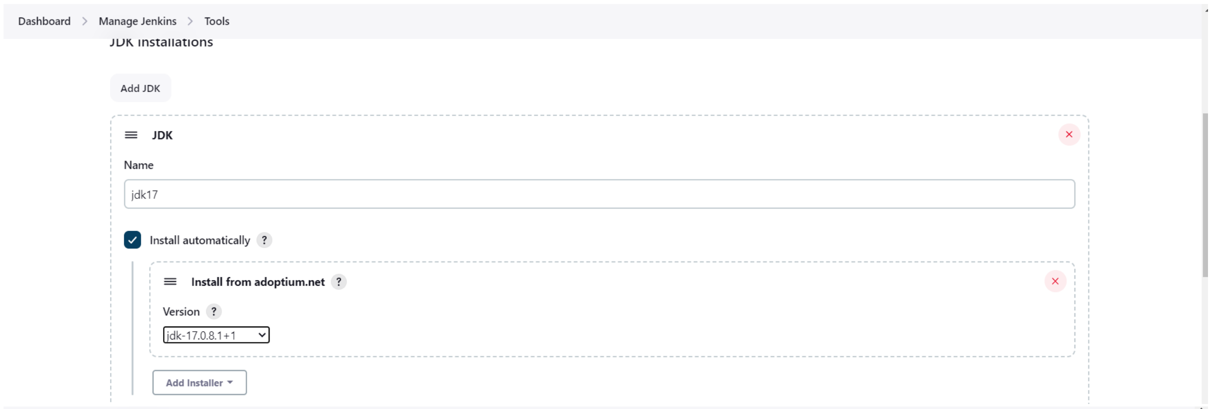
2 → SonarQube Scanner (Install without restart)



Install	Name ↓	Released
✓	Eclipse Temurin installer 1.5 Provides an installer for the JDK tool that downloads the JDK from https://adoptium.net This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	11 mo ago
✓	SonarQube Scanner 2.15 External Site/Tool Integrations Build Reports This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.	9 mo 19 days ago

3B — Configure Java and Maven in Global Tool Configuration

Goto Manage Jenkins → Tools → Install JDK(17) and Maven3(3.6.0) → Click on Apply and Save


Dashboard > Manage Jenkins > Tools

JDK installations

Add JDK

JDK

Name

jdk17

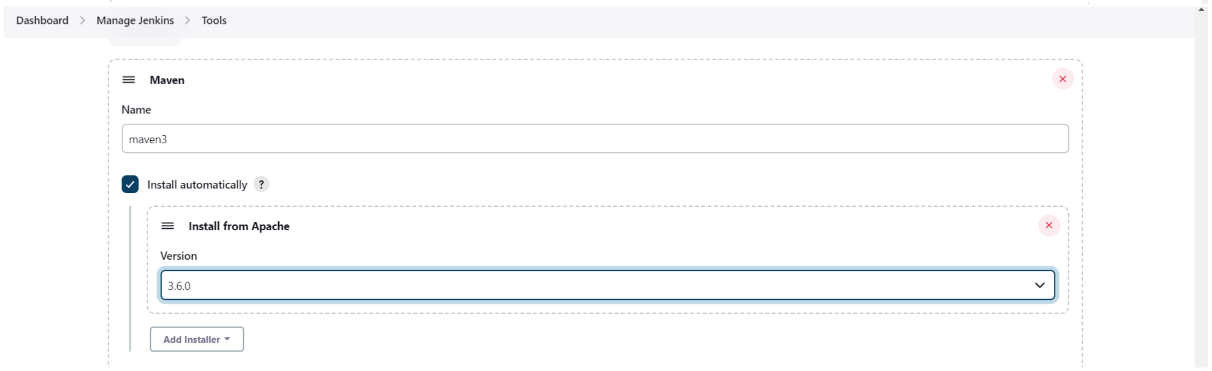
Install automatically ?

Install from adoptium.net ?

Version ?

jdk-17.0.8.1+1

Add Installer


Dashboard > Manage Jenkins > Tools

Maven

Name

maven3

Install automatically ?

Install from Apache

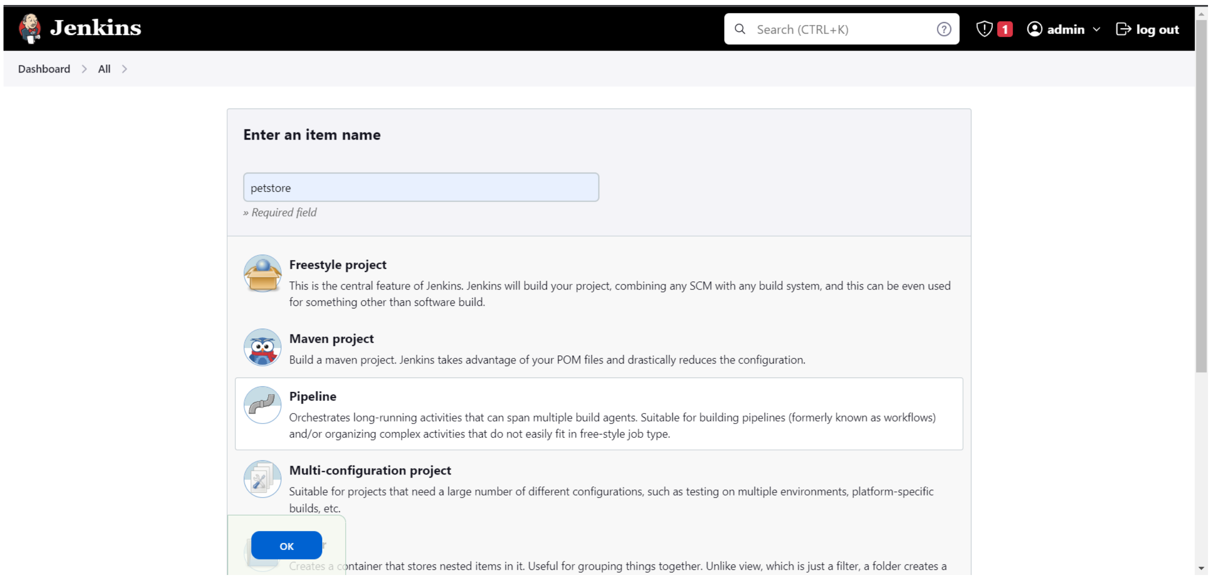
Version

3.6.0

Add Installer

3C — Create a Job

Label it as PETSHOP, click on Pipeline and OK.



Enter an item name

petstore

» 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.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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.

OK

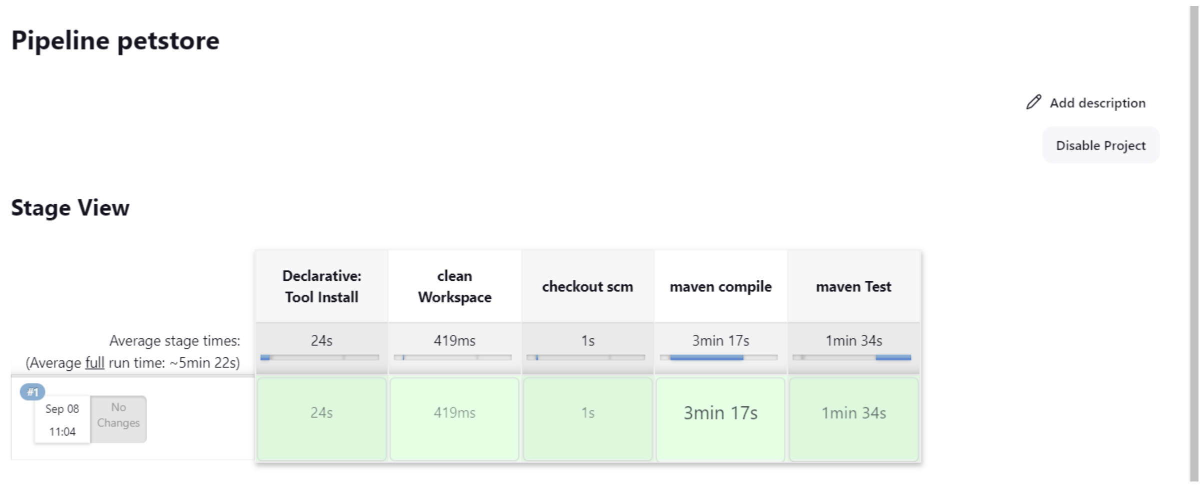
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a

Enter this in Pipeline Script,

```
pipeline{
  agent any
  tools {
    jdk 'jdk17'
```

```
        maven 'maven3'
    }
    stages{
        stage ('clean Workspace'){
            steps{
                cleanWs()
            }
        }
        stage ('checkout scm') {
            steps {
                git 'https://github.com/Aj7Ay/jpetstore-6.git'
            }
        }
        stage ('maven compile') {
            steps {
                sh 'mvn clean compile'
            }
        }
        stage ('maven Test') {
            steps {
                sh 'mvn test'
            }
        }
   }
}
```

The stage view would look like this,



Pipeline petstore

Add description

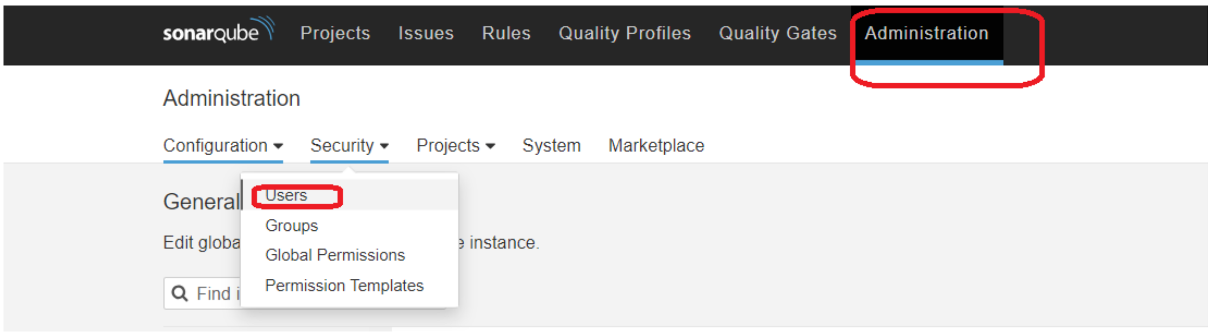
Disable Project

Stage View

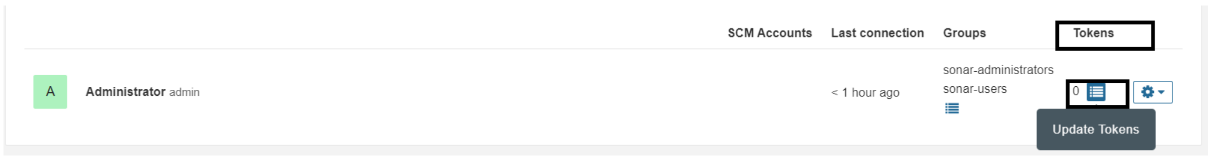
	Declarative: Tool Install	clean Workspace	checkout scm	maven compile	maven Test
Average stage times: (Average full run time: ~5min 22s)	24s	419ms	1s	3min 17s	1min 34s
#1 Sep 08 11:04 No Changes	24s	419ms	1s	3min 17s	1min 34s

Step 4 — Configure Sonar Server in Manage Jenkins

Grab the Public IP Address of your EC2 Instance, Sonarqube works on Port 9000, so <Public IP>:9000. Goto your Sonarqube Server. Click on Administration → Security → Users → Click on Tokens and Update Token → Give it a name → and click on Generate Token

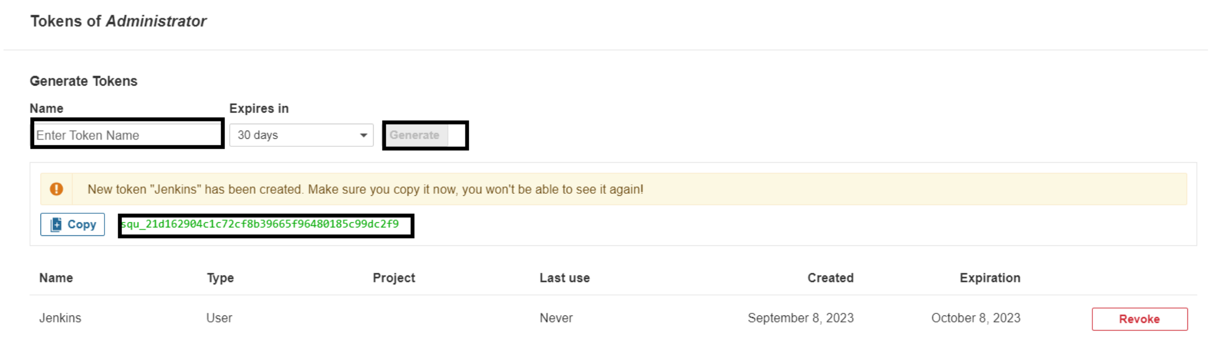


click on update Token



		SCM Accounts	Last connection	Groups	Tokens
A	Administrator admin		< 1 hour ago	sonar-administrators sonar-users	0

Create a token with a name and generate



Tokens of *Administrator*

Generate Tokens

Name: Enter Token Name

Expires in: 30 days

Generate

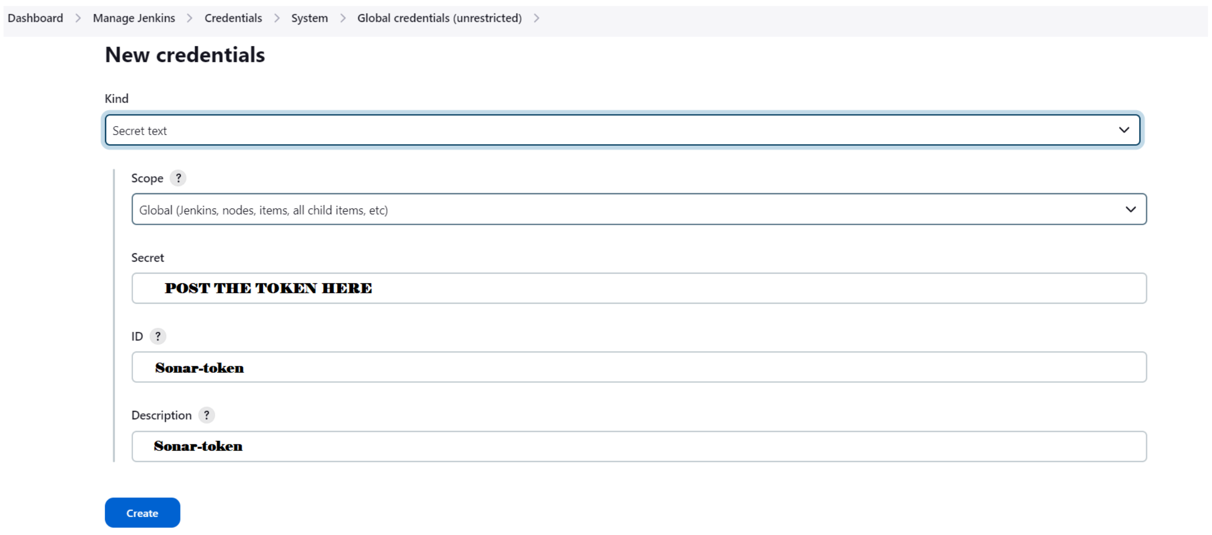
New token "Jenkins" has been created. Make sure you copy it now, you won't be able to see it again!

Copy `squ_21d162904c1c72cf8b39665f96480185c99dc2f9`

Name	Type	Project	Last use	Created	Expiration	
Jenkins	User		Never	September 8, 2023	October 8, 2023	Revoke

copy Token

Goto Jenkins Dashboard → Manage Jenkins → Credentials → Add Secret Text. It should look like this


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

New credentials

Kind: Secret text

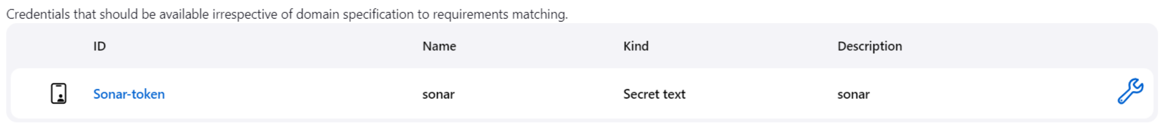
Scope: Global (Jenkins, nodes, items, all child items, etc)

Secret: POST THE TOKEN HERE

ID: Sonar-token

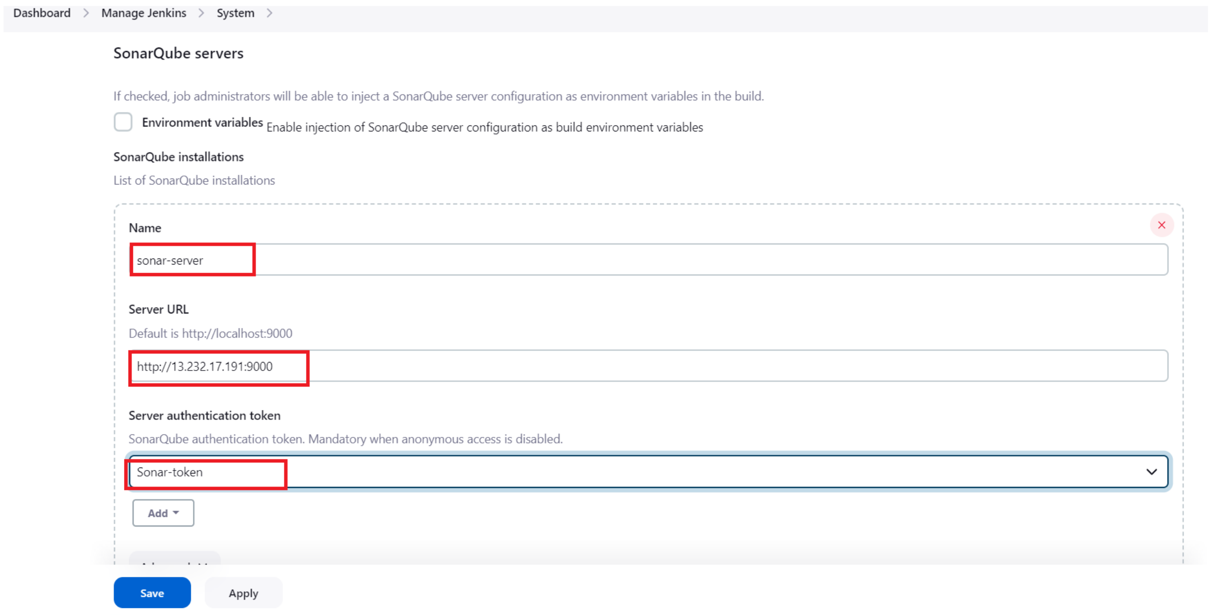
Description: Sonar-token

You will this page once you click on create


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

ID	Name	Kind	Description
Sonar-token	sonar	Secret text	sonar

Now, go to Dashboard → Manage Jenkins → System and Add like the below image.


Dashboard > Manage Jenkins > System >

SonarQube servers

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

Environment variables Enable injection of SonarQube server configuration as build environment variables

SonarQube installations

List of SonarQube installations

Name: sonar-server

Server URL

Default is http://localhost:9000

http://13.232.17.191:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

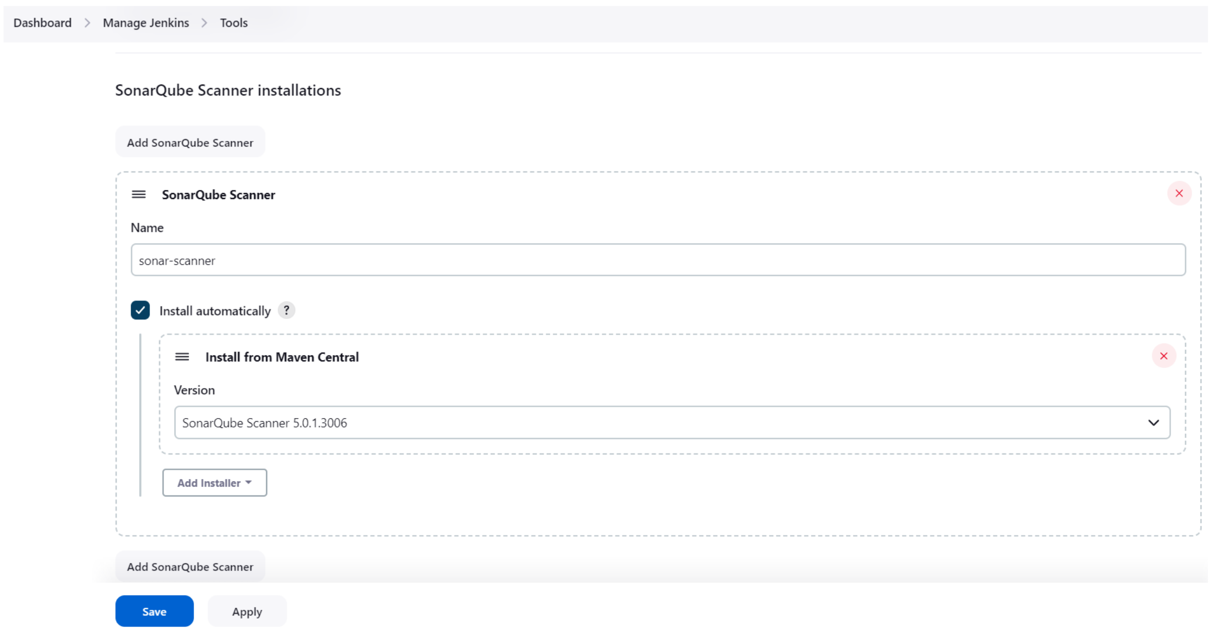
Sonar-token

Click on Apply and Save

The Configure System option is used in Jenkins to configure different server

Global Tool Configuration is used to configure different tools that we install using Plugins

We will install a sonar scanner in the tools.



SonarQube Scanner installations

Add SonarQube Scanner

SonarQube Scanner

Name: sonar-scanner

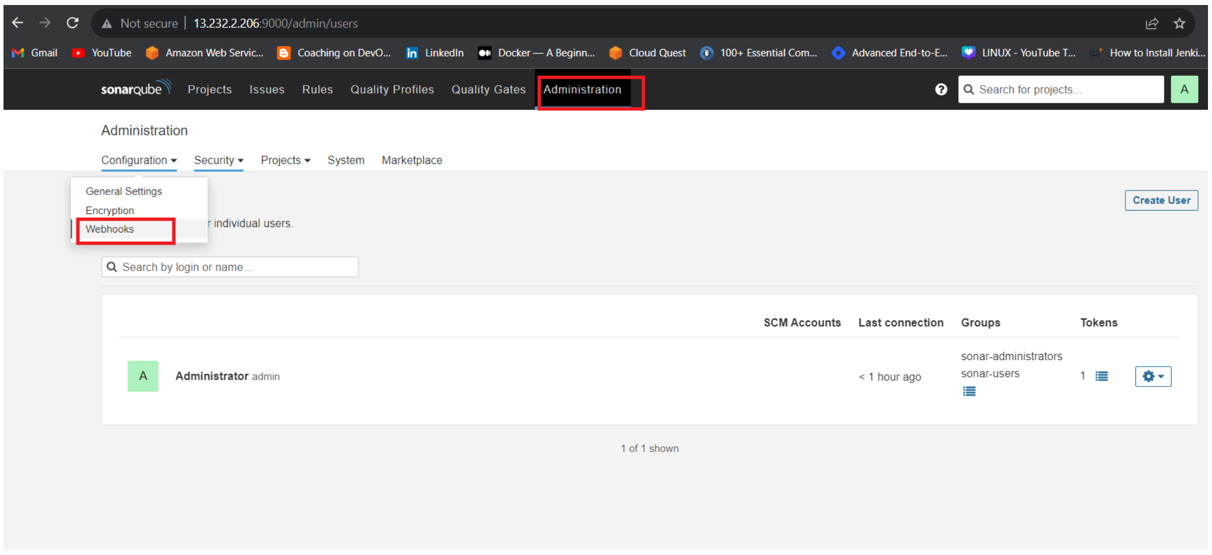
[✓] Install automatically

Install from Maven Central

Version: SonarQube Scanner 5.0.1.3006

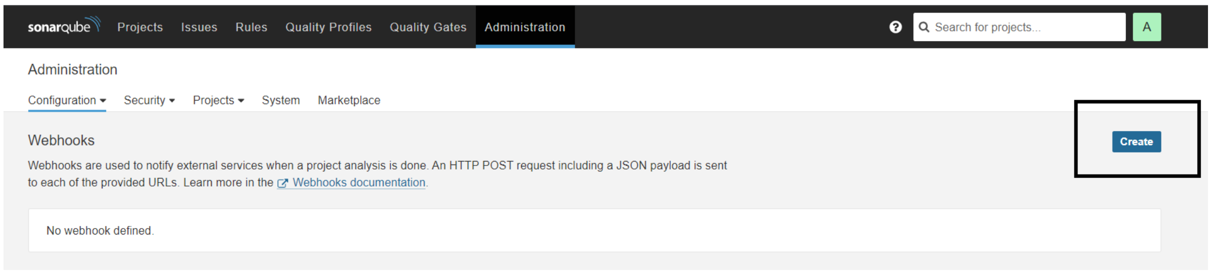
In the Sonarqube Dashboard add a quality gate also

Administration–> Configuration–>Webhooks



	SCM Accounts	Last connection	Groups	Tokens
A Administrator admin		< 1 hour ago	sonar-administrators sonar-users	1

Click on Create



Webhooks

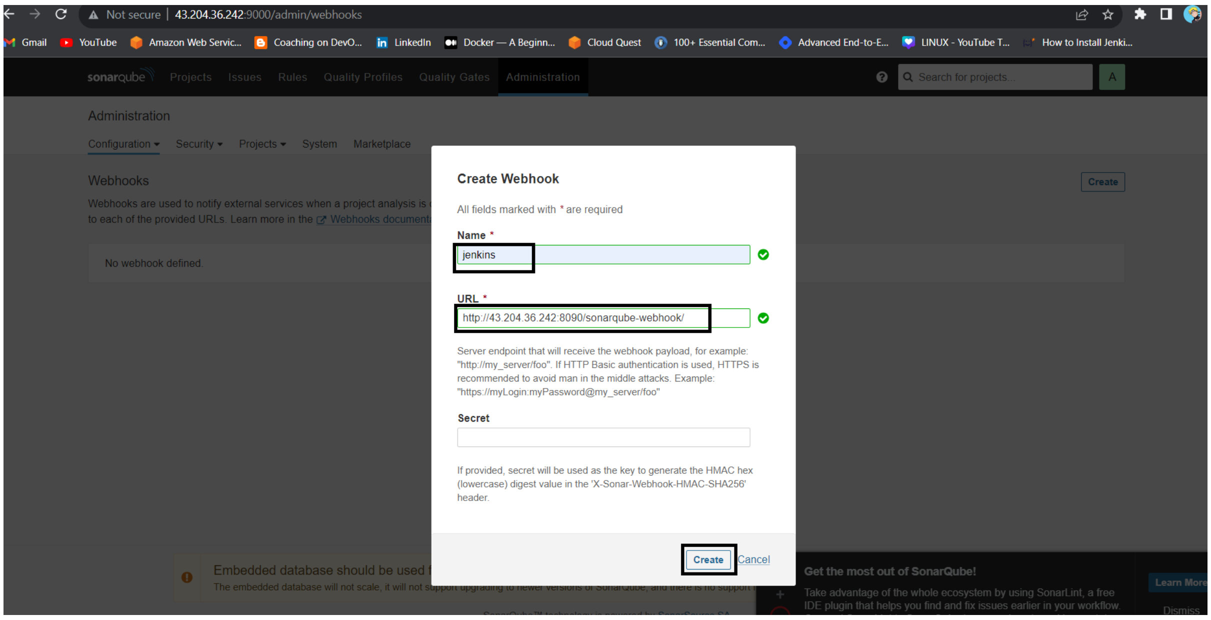
Webhooks are used to notify external services when a project analysis is done. An HTTP POST request including a JSON payload is sent to each of the provided URLs. Learn more in the Webhooks documentation.

No webhook defined.

Add details

#in url section of quality gate

http://jenkins-public-ip:8090/sonarqube-webhook/


Create Webhook

All fields marked with * are required

Name *
jenkins

URL *
http://43.204.36.242:8090/sonarqube-webhook/

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

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.

Let’s go to our Pipeline and add Sonarqube Stage in our Pipeline Script.

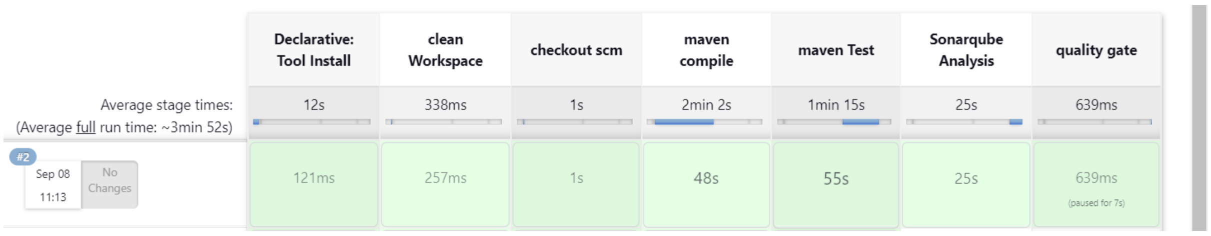
#under tools section add this environment

```
environment {
        SCANNER_HOME=tool 'sonar-scanner'
    }
```

# in stages add this

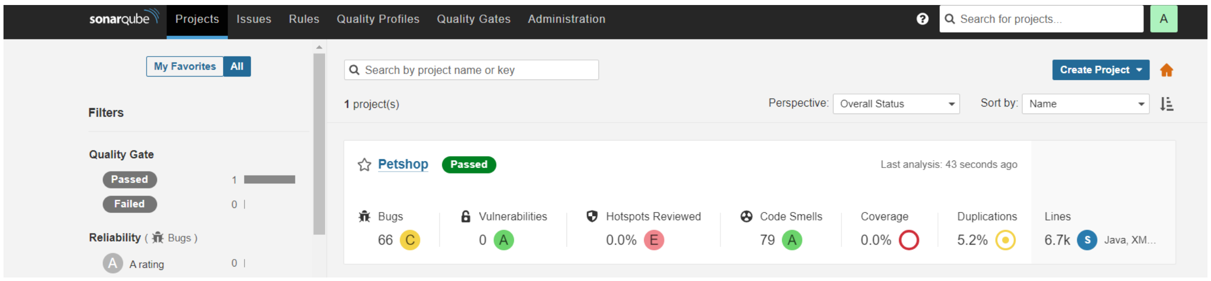
```
stage("Sonarqube Analysis "){
            steps{
                withSonarQubeEnv('sonar-server') {
                    sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Petshop \
                    -Dsonar.java.binaries=. \
                    -Dsonar.projectKey=Petshop '''
                }
            }
        }
        stage("quality gate"){
            steps {
                script {
                  waitForQualityGate abortPipeline: false, credentialsId: 'Sonar-token'
                }
        }
    }
```

Click on Build now, you will see the stage view like this



	Declarative: Tool Install	clean Workspace	checkout scm	maven compile	maven Test	Sonarqube Analysis	quality gate
Average stage times: (Average full run time: ~3min 52s)	12s	338ms	1s	2min 2s	1min 15s	25s	639ms
#2 Sep 08 11:13 No Changes	121ms	257ms	1s	48s	55s	25s	639ms (paused for 7s)

To see the report, you can go to Sonarqube Server and go to Projects.



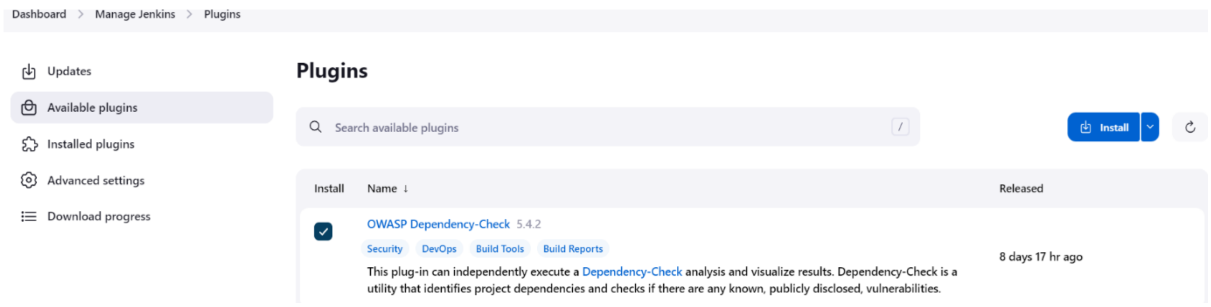
Petshop Passed — Last analysis: 43 seconds ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
66 C	0 A	0.0% E	79 A	0.0%	5.2%	6.7k S Java, XM...

You can see the report has been generated and the status shows as passed. You can see that there are 6.7k lines. To see a detailed report, you can go to issues.

Step 5 — Install OWASP Dependency Check Plugins

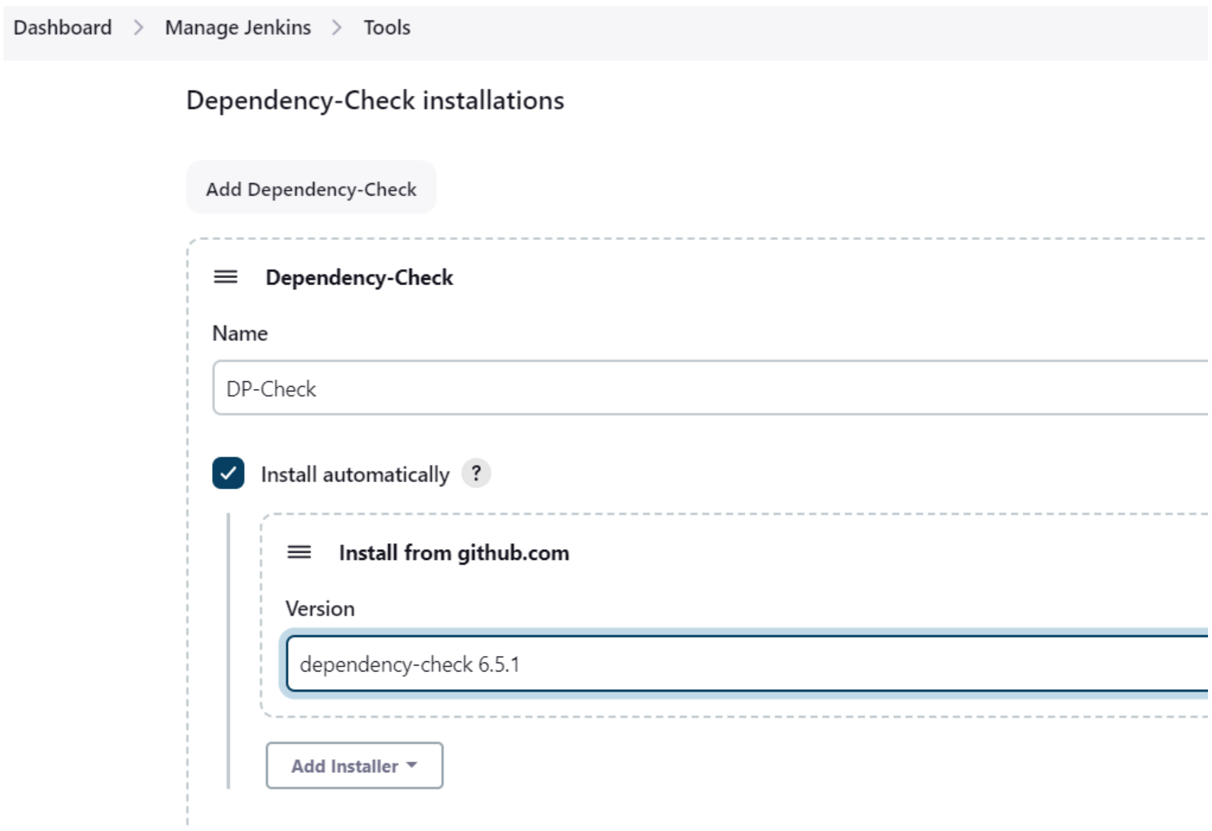
GotoDashboard → Manage Jenkins → Plugins → OWASP Dependency-Check. Click on it and install it without restart.



Install	Name ↓	Released
✓	OWASP Dependency-Check 5.4.2 Security DevOps Build Tools Build Reports This plug-in can independently execute a Dependency-Check analysis and visualize results. Dependency-Check is a utility that identifies project dependencies and checks if there are any known, publicly disclosed, vulnerabilities.	8 days 17 hr ago

First, we configured the Plugin and next, we had to configure the Tool

Goto Dashboard → Manage Jenkins → Tools →


Dashboard > Manage Jenkins > Tools

Dependency-Check installations

Dependency-Check

Name: DP-Check

Install automatically: ✓

Install from github.com

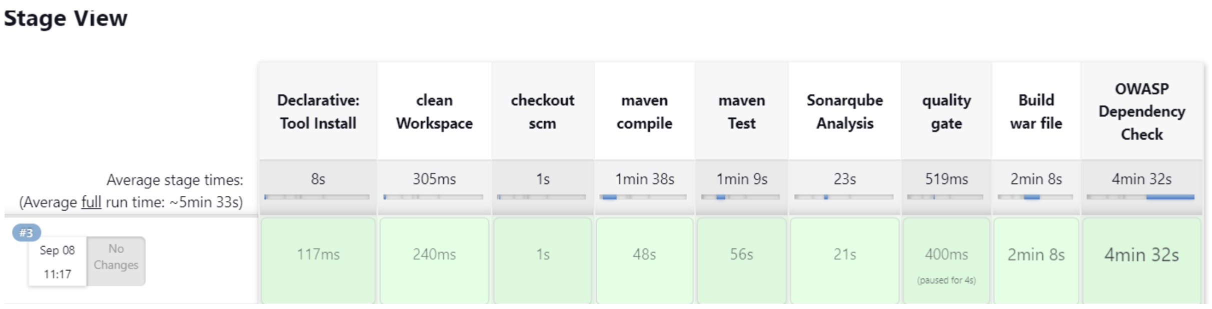
Version: dependency-check 6.5.1

Click on Apply and Save here.

Now go configure → Pipeline and add this stage to your pipeline and build.

```
stage ('Build war file'){
       steps{
         sh 'mvn clean install -DskipTests=true'
       }
    }
    stage("OWASP Dependency Check"){
       steps{
         dependencyCheck additionalArguments: '--scan ./ --format XML ', odcInstallation: 'DP-
Check'
         dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
       }
    }
```

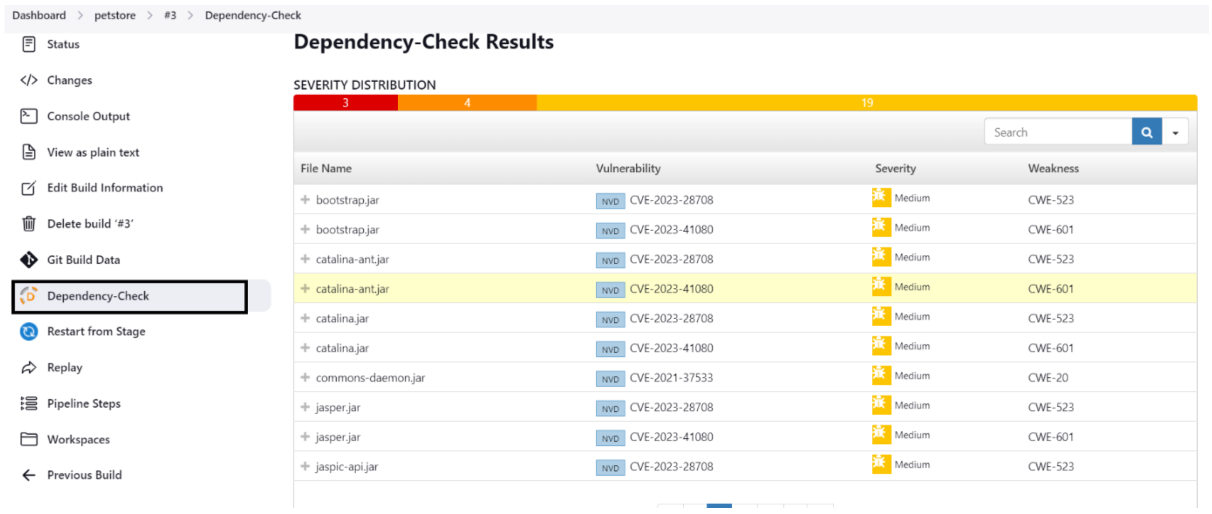
The stage view would look like this,



Stage View

	Declarative: Tool Install	clean Workspace	checkout scm	maven compile	maven Test	Sonarqube Analysis	quality gate	Build war file	OWASP Dependency Check
Average stage times: (Average full run time: ~5min 33s)	8s	305ms	1s	1min 38s	1min 9s	23s	519ms	2min 8s	4min 32s
#3 Sep 08 11:17 No Changes	117ms	240ms	1s	48s	56s	21s	400ms (paused for 4s)	2min 8s	4min 32s

You will see that in status, a graph will also be generated and Vulnerabilities.



Dependency-Check Results

SEVERITY DISTRIBUTION

File Name	Vulnerability	Severity	Weakness
bootstrap.jar	NVD CVE-2023-28708	Medium	CWE-523
bootstrap.jar	NVD CVE-2023-41080	Medium	CWE-601
catalina-ant.jar	NVD CVE-2023-28708	Medium	CWE-523
catalina-ant.jar	NVD CVE-2023-41080	Medium	CWE-601
catalina.jar	NVD CVE-2023-28708	Medium	CWE-523
catalina.jar	NVD CVE-2023-41080	Medium	CWE-601
commons-daemon.jar	NVD CVE-2021-37533	Medium	CWE-20
jasper.jar	NVD CVE-2023-28708	Medium	CWE-523
jasper.jar	NVD CVE-2023-41080	Medium	CWE-601
jaspic-api.jar	NVD CVE-2023-28708	Medium	CWE-523

Step 6 — Docker plugin and credential Setup

We need to install the Docker tool in our system, Goto Dashboard → Manage Plugins → Available plugins → Search for Docker and install these plugins

Docker

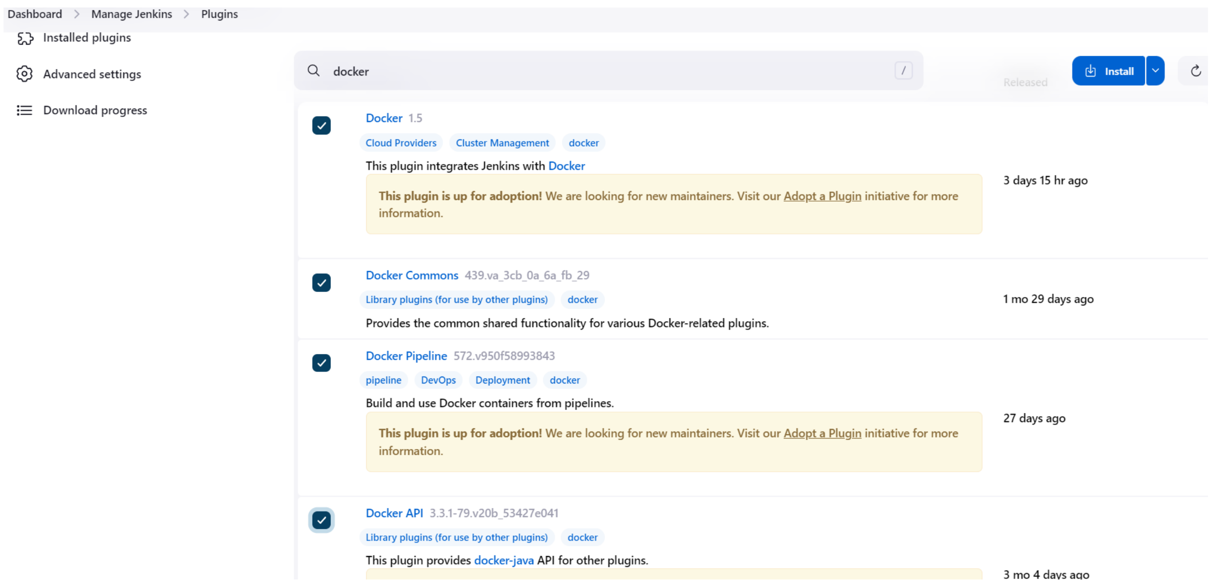
Docker Commons

Docker Pipeline

Docker API

docker-build-step

and click on install without restart


Dashboard > Manage Jenkins > Plugins

Installed plugins
Advanced settings
Download progress

docker

Install

Released

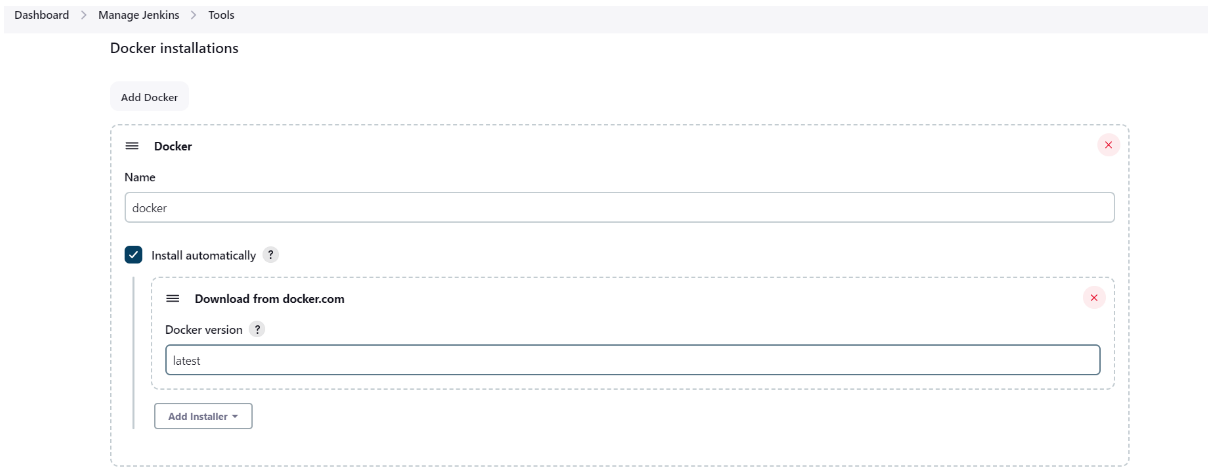
✓ Docker 1.5
Cloud Providers · Cluster Management · docker
This plugin integrates Jenkins with Docker
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.
3 days 15 hr ago

✓ Docker Commons 439.va_3cb_0a_6a_fb_29
Library plugins (for use by other plugins) · docker
Provides the common shared functionality for various Docker-related plugins.
1 mo 29 days ago

✓ Docker Pipeline 572.v950f58993843
pipeline · DevOps · Deployment · docker
Build and use Docker containers from pipelines.
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.
27 days ago

✓ Docker API 3.3.1-79.v20b_53427e041
Library plugins (for use by other plugins) · docker
This plugin provides docker-java API for other plugins.
3 mo 4 days ago

Now, goto Dashboard → Manage Jenkins → Tools →


Dashboard > Manage Jenkins > Tools

Docker installations

Add Docker

Docker

Name: docker

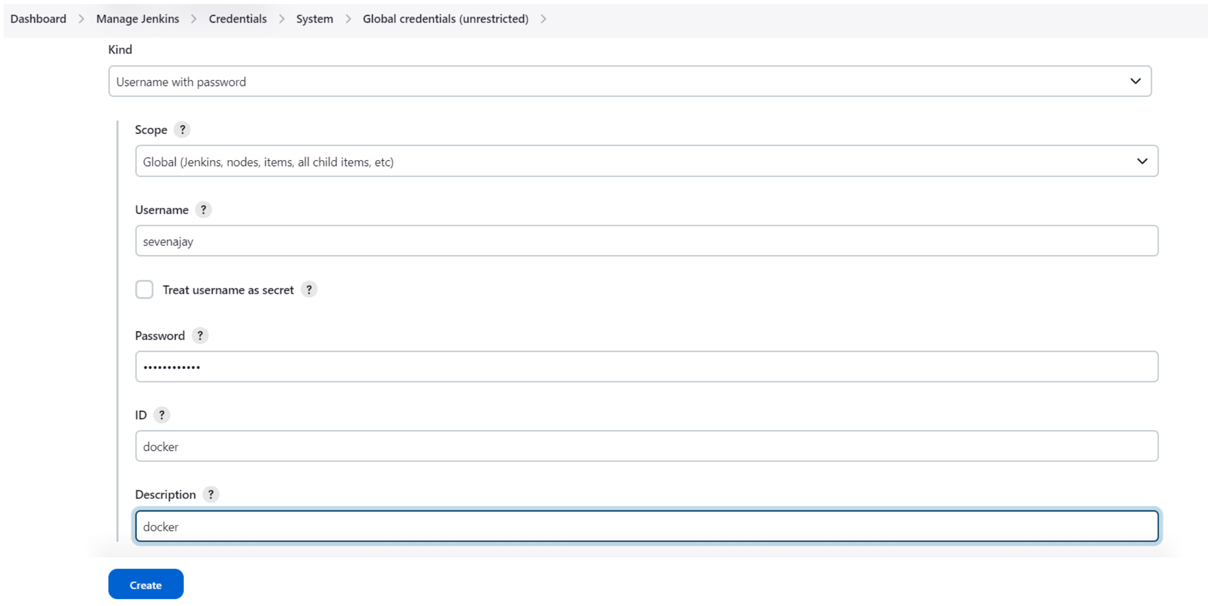
✓ Install automatically

Download from docker.com

Docker version: latest

Add Installer

Add DockerHub Username and Password under Global Credentials


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

Kind: Username with password

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

Username: sevenajay

☐ Treat username as secret

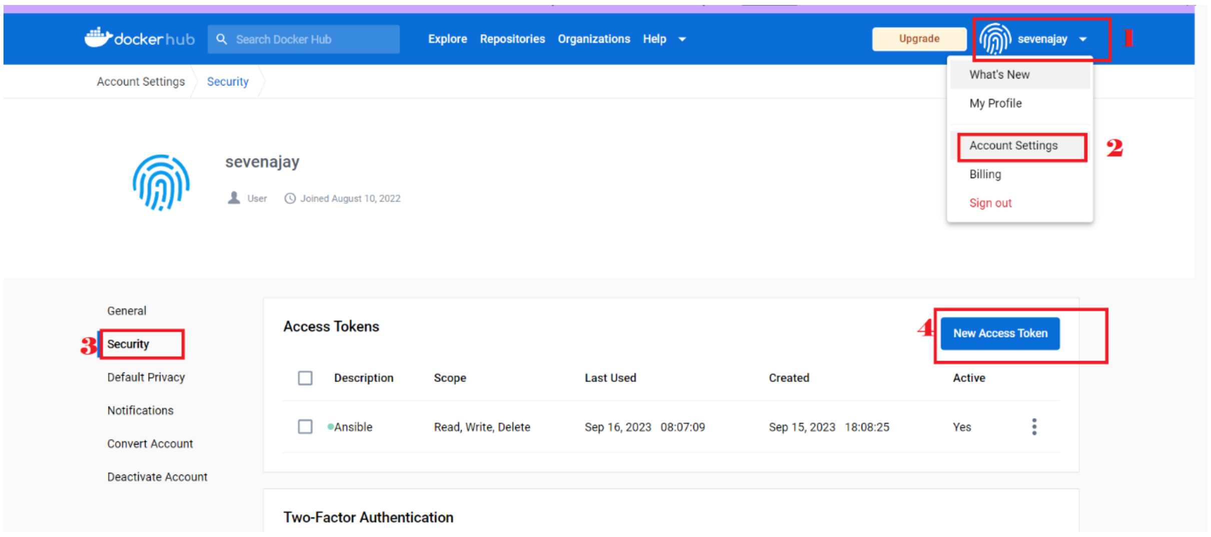
Password: ••••••••••••

ID: docker

Description: docker

Create

create a personal Access token from the docker hub which is used for ansible-playbook



sevenajay

User Joined August 10, 2022

Access Tokens

Description	Scope	Last Used	Created	Active
Ansible	Read, Write, Delete	Sep 16, 2023 08:07:09	Sep 15, 2023 18:08:25	Yes

copy it and save for later.

Let's install Ansible on the Jenkins server

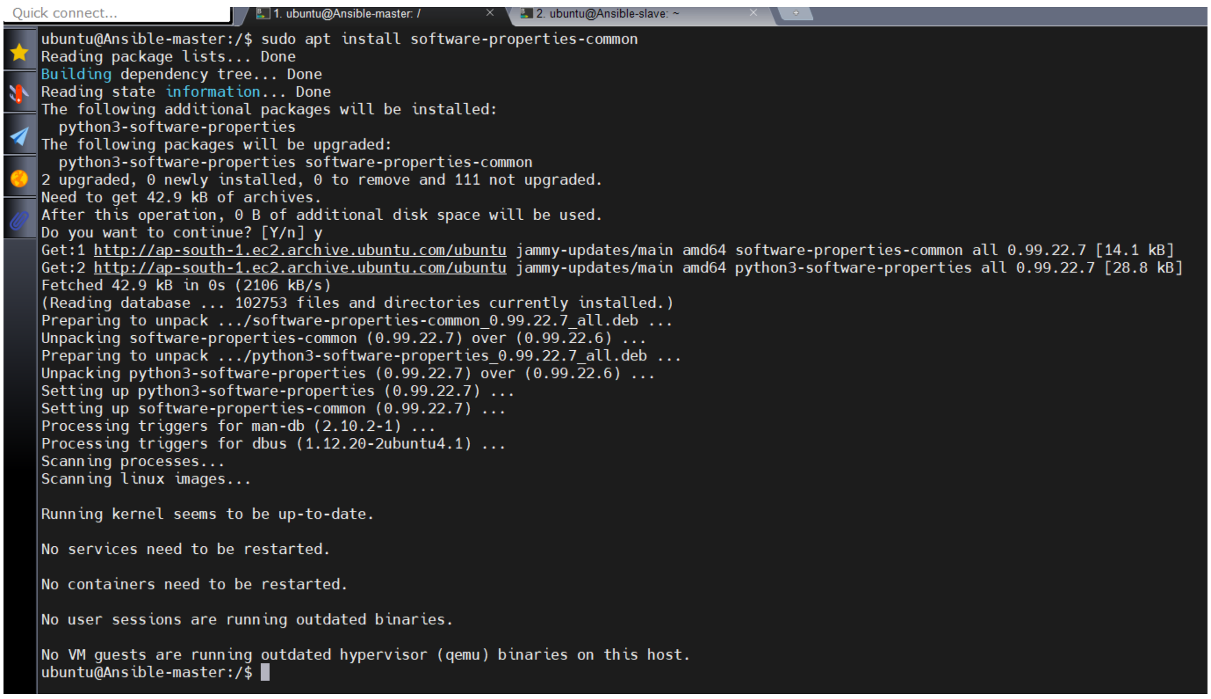
STEP 7 -Adding Ansible Repository in Ubuntu

Step1:Update your system packages:

sudo apt-get update

Step 2: First Install Required packages to install Ansible.

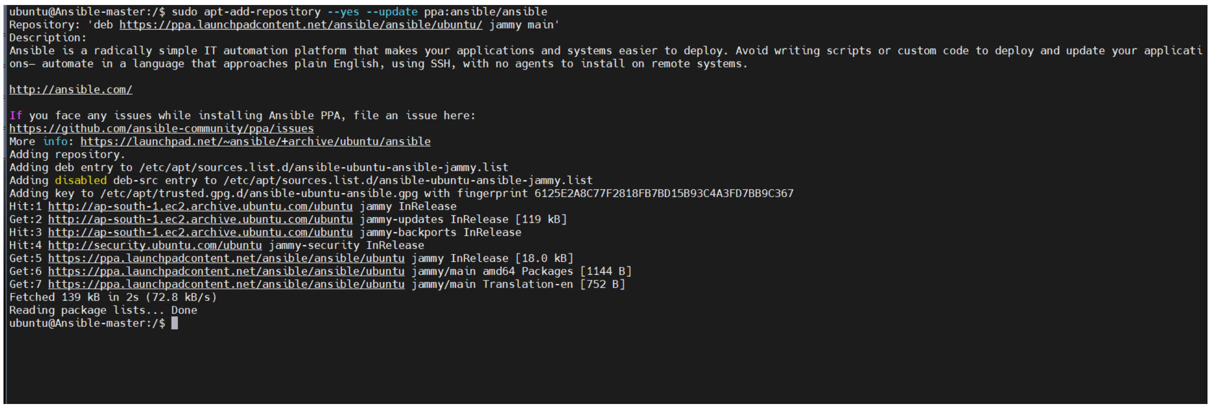
sudo apt install software-properties-common



```
ubuntu@Ansible-master:/$ sudo apt install software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-software-properties
The following packages will be upgraded:
  python3-software-properties software-properties-common
2 upgraded, 0 newly installed, 0 to remove and 111 not upgraded.
Need to get 42.9 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 software-properties-common all 0.99.22.7 [14.1 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-software-properties all 0.99.22.7 [28.8 kB]
Fetched 42.9 kB in 0s (2106 kB/s)
(Reading database ... 102753 files and directories currently installed.)
Preparing to unpack .../software-properties-common_0.99.22.7_all.deb ...
Unpacking software-properties-common (0.99.22.7) over (0.99.22.6) ...
Preparing to unpack .../python3-software-properties_0.99.22.7_all.deb ...
Unpacking python3-software-properties (0.99.22.7) over (0.99.22.6) ...
Setting up python3-software-properties (0.99.22.7) ...
Setting up software-properties-common (0.99.22.7) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@Ansible-master:/$
```

Step3: Add the ansible repository via PPA

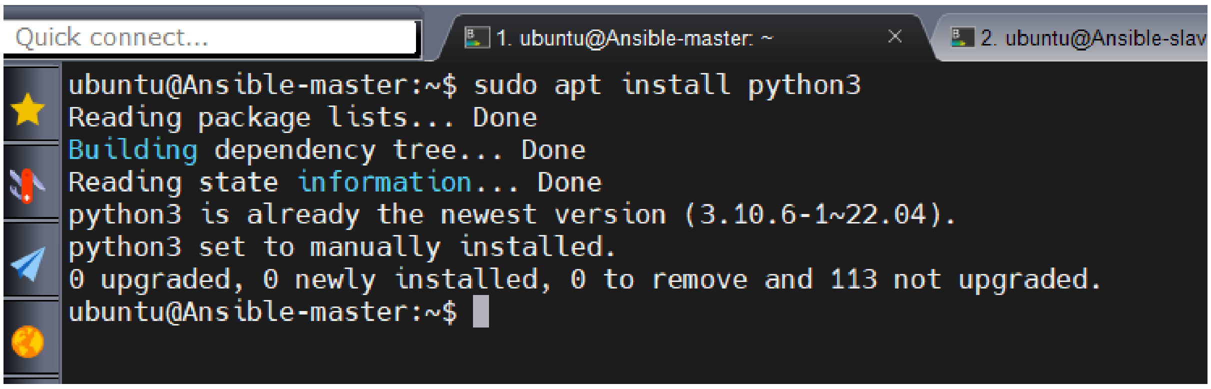
sudo add-apt-repository --yes --update ppa:ansible/ansible



```
ubuntu@Ansible-master:/$ sudo apt-add-repository --yes --update ppa:ansible/ansible
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applicati
ons— automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/

If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Adding deb entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding key to /etc/apt/trusted.gpg.d/ansible-ubuntu-ansible.gpg with fingerprint 6125E2A8C77F2818FB7BD15B93C4A3FD7BB9C367
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease [18.0 kB]
Get:6 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 Packages [1144 B]
Get:7 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main Translation-en [752 B]
Fetched 139 kB in 2s (72.8 kB/s)
Reading package lists... Done
ubuntu@Ansible-master:/$
```

Install Python3 on the Ansible master

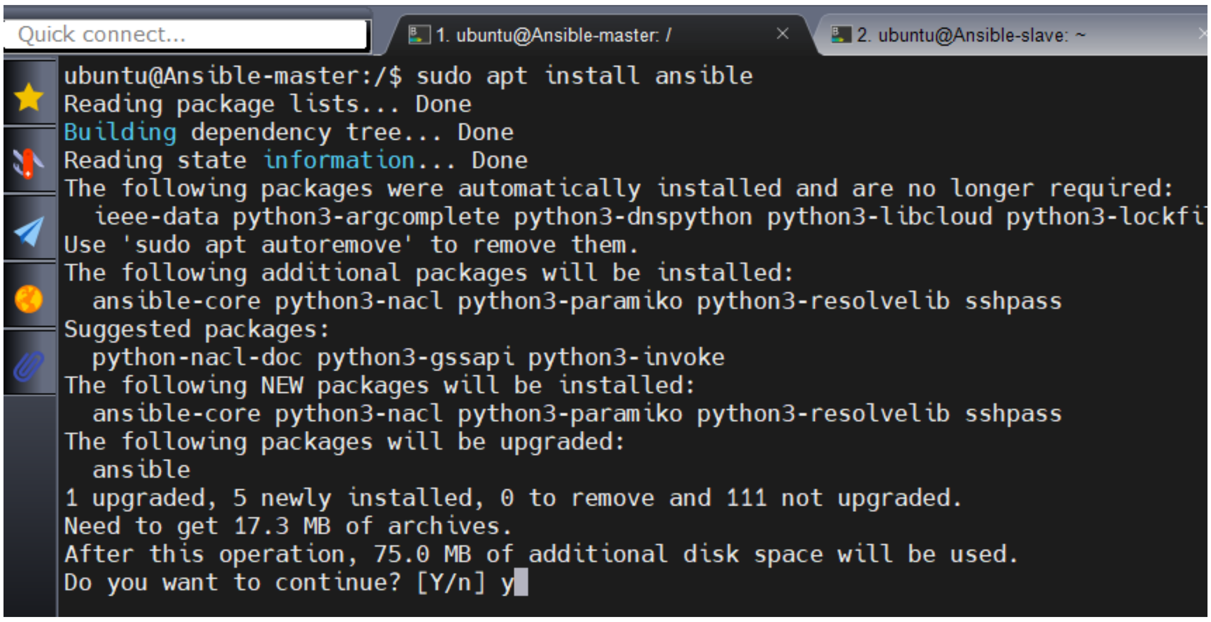
sudo apt install python3



```
ubuntu@Ansible-master:~$ sudo apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 113 not upgraded.
ubuntu@Ansible-master:~$
```

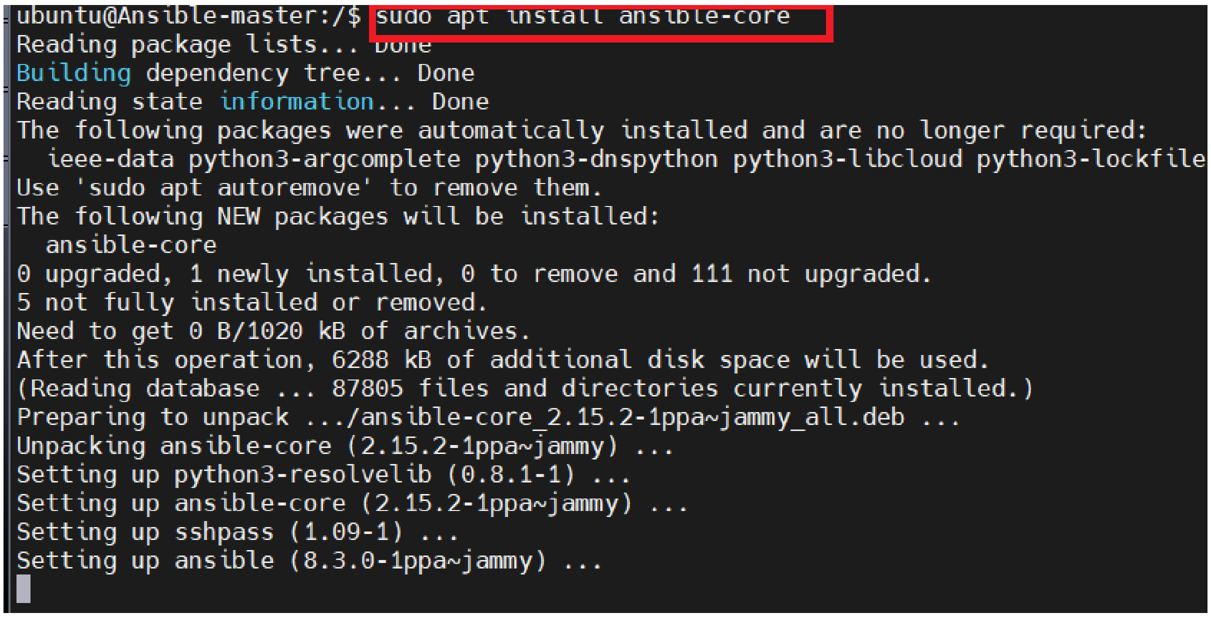
Step1: Install Ansible on Ubuntu 22.04 LTS

sudo apt install ansible -y



```
ubuntu@Ansible-master:/$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ieee-data python3-argcomplete python3-dnspython python3-libcloud python3-lockfi
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ansible-core python3-nacl python3-paramiko python3-resolvelib sshpass
Suggested packages:
  python-nacl-doc python3-gssapi python3-invoke
The following NEW packages will be installed:
  ansible-core python3-nacl python3-paramiko python3-resolvelib sshpass
The following packages will be upgraded:
  ansible
1 upgraded, 5 newly installed, 0 to remove and 111 not upgraded.
Need to get 17.3 MB of archives.
After this operation, 75.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

sudo apt install ansible-core -y



```
ubuntu@Ansible-master:/$ sudo apt install ansible-core
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ieee-data python3-argcomplete python3-dnspython python3-libcloud python3-lockfile
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  ansible-core
0 upgraded, 1 newly installed, 0 to remove and 111 not upgraded.
5 not fully installed or removed.
Need to get 0 B/1020 kB of archives.
After this operation, 6288 kB of additional disk space will be used.
(Reading database ... 87805 files and directories currently installed.)
Preparing to unpack .../ansible-core_2.15.2-1ppa~jammy_all.deb ...
Unpacking ansible-core (2.15.2-1ppa~jammy) ...
Setting up python3-resolvelib (0.8.1-1) ...
Setting up ansible-core (2.15.2-1ppa~jammy) ...
Setting up sshpass (1.09-1) ...
Setting up ansible (8.3.0-1ppa~jammy) ...
```

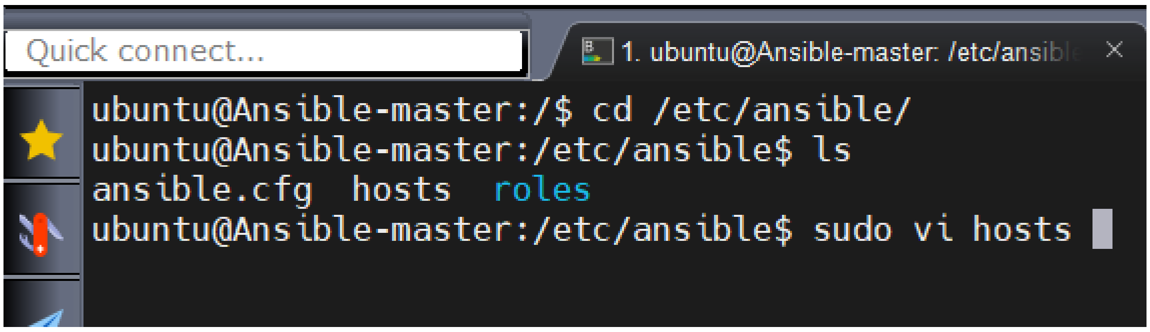
Step2: To check version :

ansible --version

cd /etc/ansible

sudo vi hosts

Now go to the host file inside the Ansible server and paste the public IP of the Jenkins

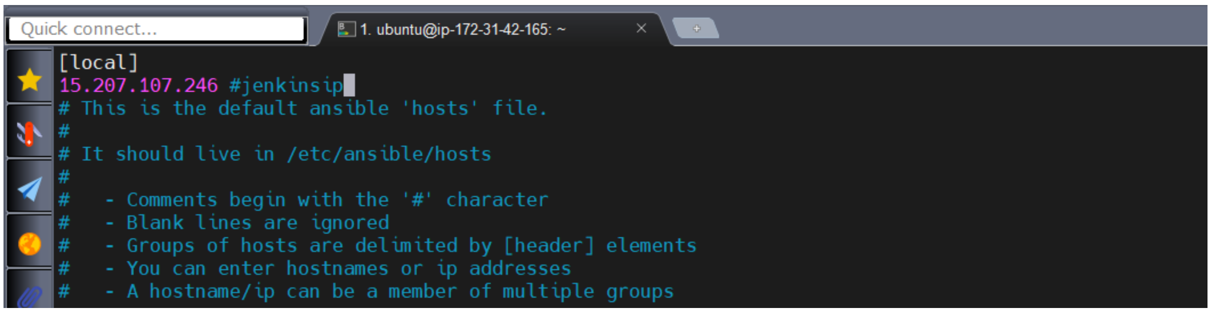


```
ubuntu@Ansible-master:/$ cd /etc/ansible/
ubuntu@Ansible-master:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@Ansible-master:/etc/ansible$ sudo vi hosts
```

You can create a group and paste ip address below:

[local]#any name you want

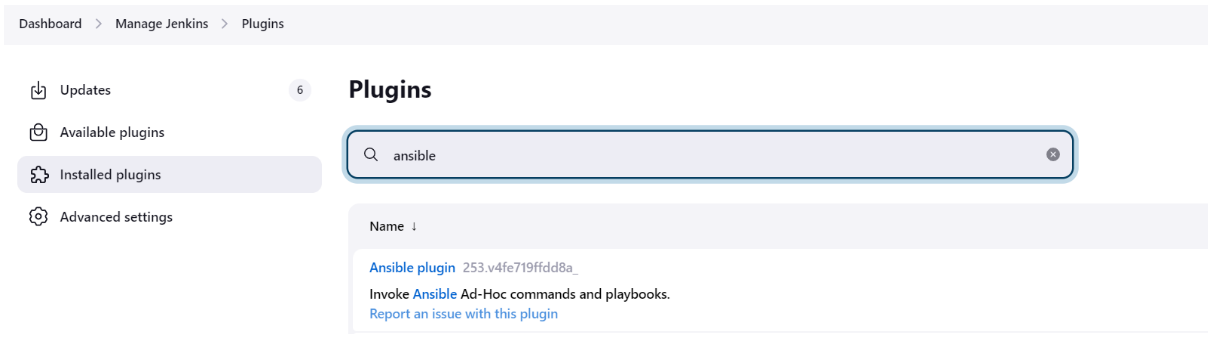
Ip of Jenkins



```
[local]
15.207.107.246 #jenkinsip
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups
```

save and exit from the file.

Let's install The Ansible plugin to integrate with Jenkins.


Dashboard > Manage Jenkins > Plugins

Updates 6
Available plugins
Installed plugins
Advanced settings

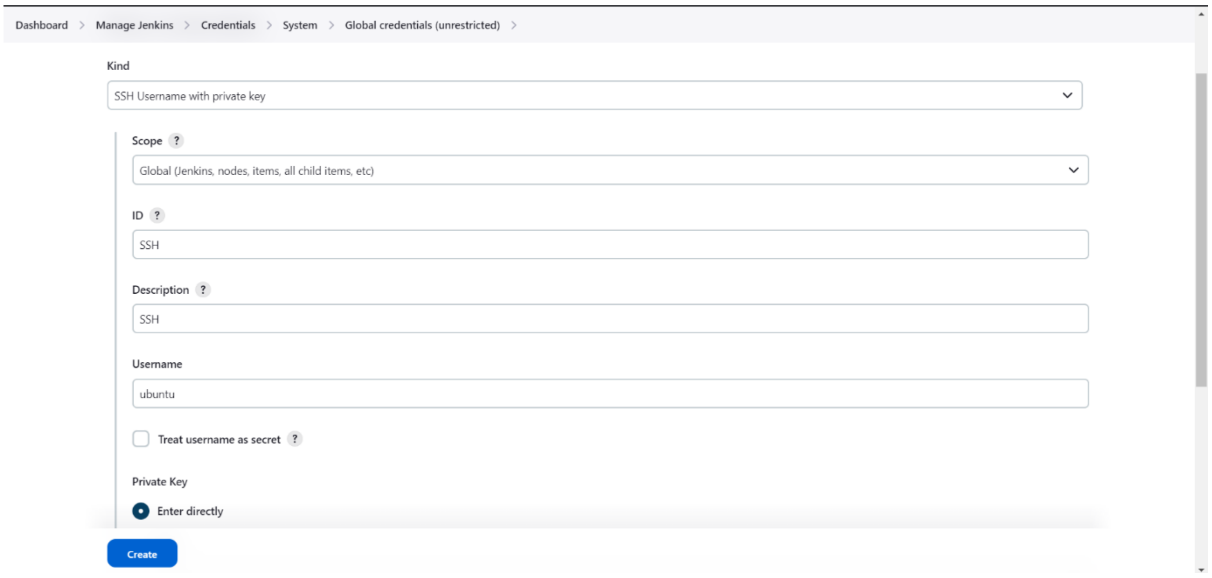
Plugins

ansible

Name ↓

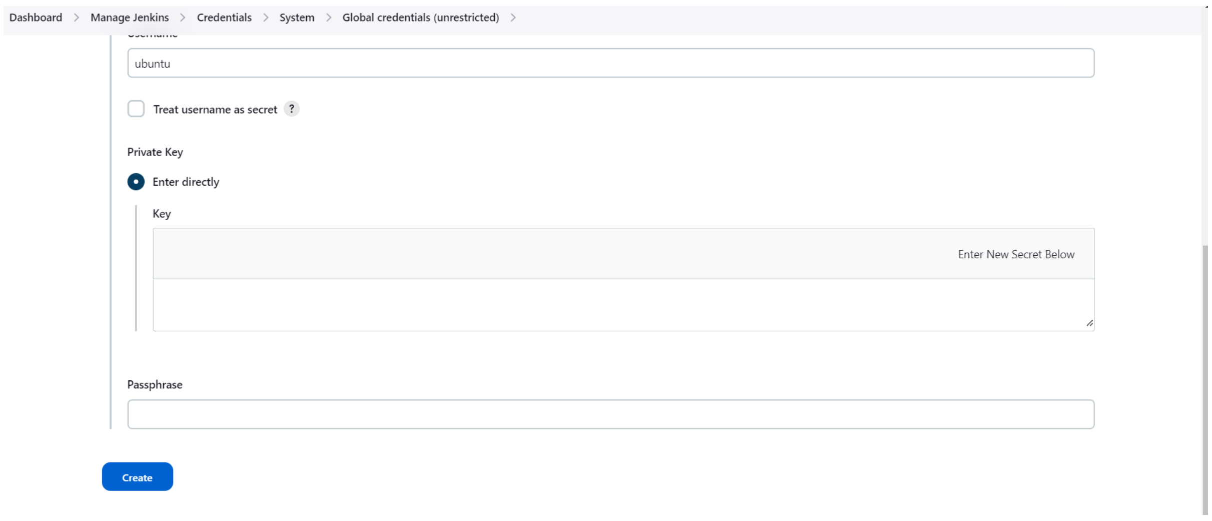
Ansible plugin 253.v4fe719ffdd8a_
Invoke Ansible Ad-Hoc commands and playbooks.
Report an issue with this plugin

Now add Credentials to invoke Ansible with Jenkins.


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

Kind: SSH Username with private key
Scope: Global (Jenkins, nodes, items, all child items, etc)
ID: SSH
Description: SSH
Username: ubuntu
Treat username as secret
Private Key
Enter directly
Create

In the private key section, Select Enter directly and add your Pem file for the key.

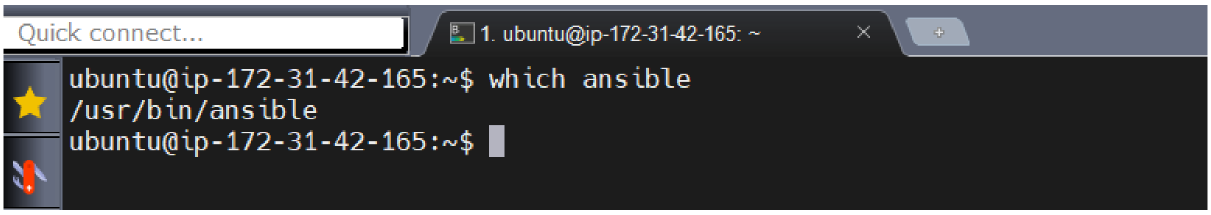

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

ubuntu
Treat username as secret
Private Key
Enter directly
Key
Enter New Secret Below
Passphrase
Create

and finally, click on Create.

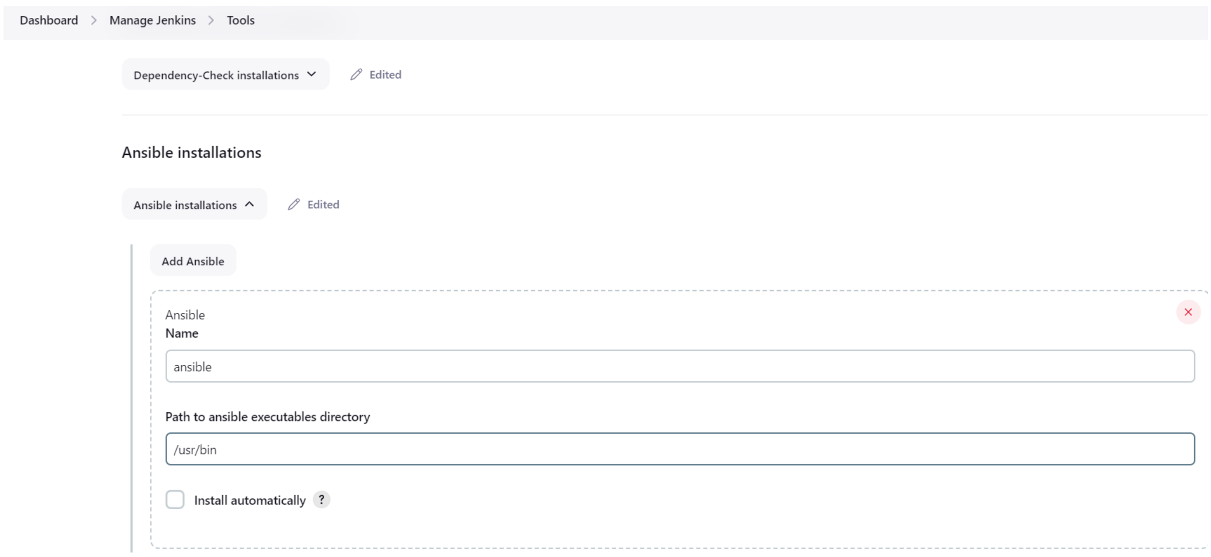
Give this command in your Jenkins machine to find the path of your ansible which is used in the tool section of Jenkins.

which ansible



```
ubuntu@ip-172-31-42-165:~$ which ansible
/usr/bin/ansible
ubuntu@ip-172-31-42-165:~$
```

Copy that path and add it to the tools section of Jenkins at ansible installations.


Dashboard > Manage Jenkins > Tools
Dependency-Check installations
Edited
Add Ansible
Install automatically

Ansible installations

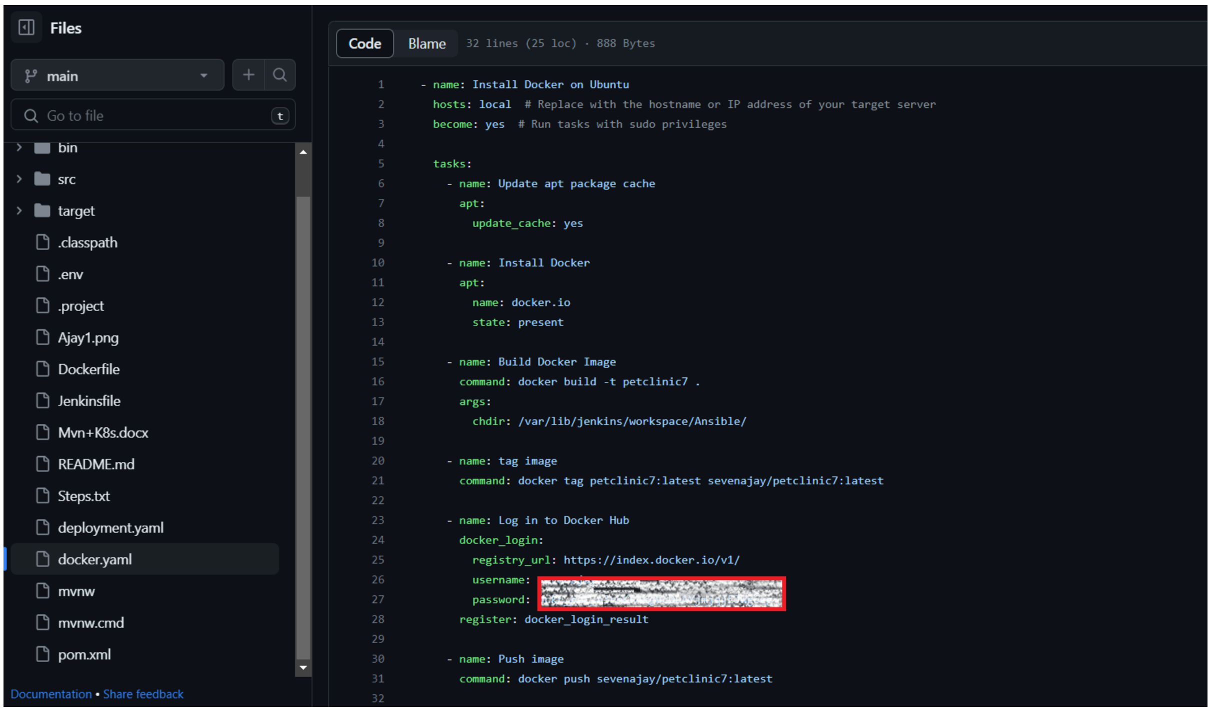
Ansible installations — Edited

Ansible
Name: ansible

Path to ansible executables directory: /usr/bin

Now write an Ansible playbook to create a docker image, tag it and push it to the docker hub, and finally, we will deploy it on a container using Ansible.

Just sample code.



```
- name: Install Docker on Ubuntu
  hosts: local  # Replace with the hostname or IP address of your target server
  become: yes  # Run tasks with sudo privileges

  tasks:
    - name: Update apt package cache
      apt:
        update_cache: yes

    - name: Install Docker
      apt:
        name: docker.io
        state: present

    - name: Build Docker Image
      command: docker build -t petclinic7 .
      args:
        chdir: /var/lib/jenkins/workspace/Ansible/

    - name: tag image
      command: docker tag petclinic7:latest sevenajay/petclinic7:latest

    - name: Log in to Docker Hub
      docker_login:
        registry_url: https://index.docker.io/v1/
        username: [redacted]
        password:
      register: docker_login_result

    - name: Push image
      command: docker push sevenajay/petclinic7:latest
```

- name: docker build and push

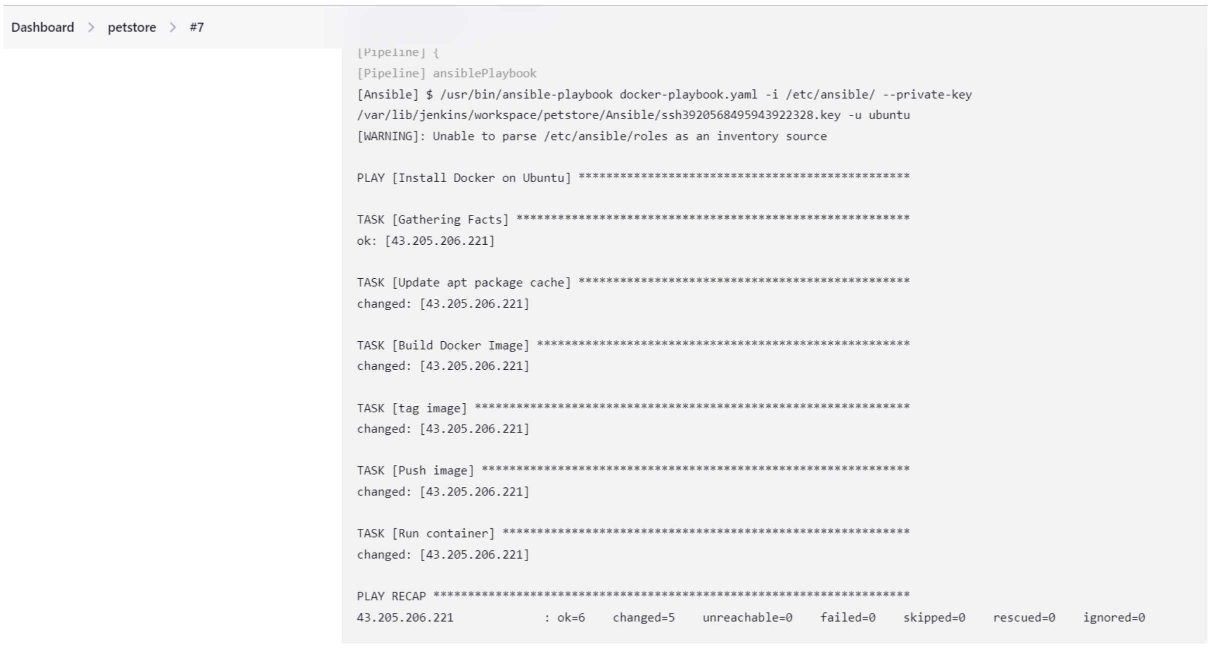
  hosts: docker # Replace with the hostname or IP address of your target server

```
  become: yes  # Run tasks with sudo privileges
  tasks:
   - name: Update apt package cache
     apt:
       update_cache: yes
   - name: Build Docker Image
     command: docker build -t petstore .
     args:
       chdir: /var/lib/jenkins/workspace/petstore
   - name: tag image
     command: docker tag petstore:latest sevenajay/petstore:latest
   - name: Log in to Docker Hub
     community.docker.docker_login:
       registry_url: https://index.docker.io/v1/
       username: sevenajay
       password: <docker pat>
   - name: Push image
     command: docker push sevenajay/petstore:latest
   - name: Run container
     command: docker run -d --name pet1 -p 8081:8080 sevenajay/petstore:latest
```

Add this stage to the pipeline to build and push it to the docker hub, and run the container.

```
stage('Install Docker') {
            steps {
                dir('Ansible'){
                  script {
                        ansiblePlaybook credentialsId: 'ssh', disableHostKeyChecking: true, installation:
'ansible', inventory: '/etc/ansible/', playbook: 'docker-playbook.yaml'
                    }
                }
            }
        }
```

Output of pipeline

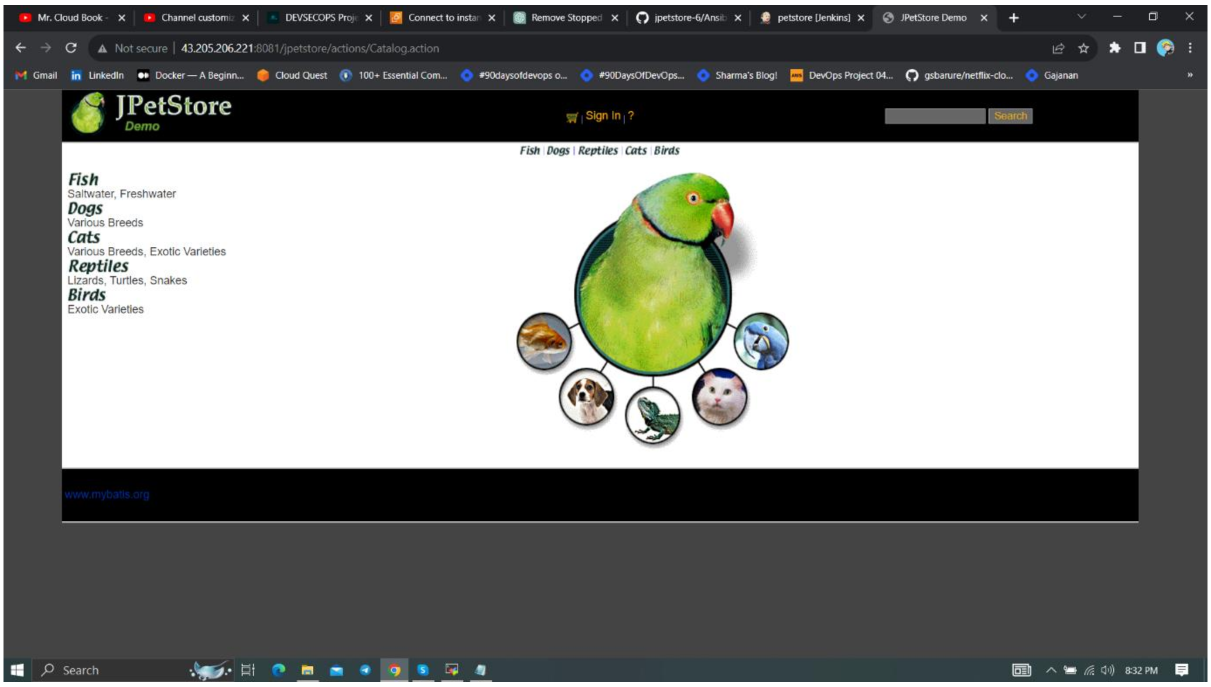


Dashboard > petstore > #7

```
[Pipeline] {
[Pipeline] ansiblePlaybook
[Ansible] $ /usr/bin/ansible-playbook docker-playbook.yaml -i /etc/ansible/ --private-key
/var/lib/jenkins/workspace/petstore/Ansible/ssh3920568495943922328.key -u ubuntu
[WARNING]: Unable to parse /etc/ansible/roles as an inventory source

PLAY [Install Docker on Ubuntu] ************************************************

TASK [Gathering Facts] *********************************************************
ok: [43.205.206.221]

TASK [Update apt package cache] ************************************************
changed: [43.205.206.221]

TASK [Build Docker Image] ******************************************************
changed: [43.205.206.221]

TASK [tag image] ***************************************************************
changed: [43.205.206.221]

TASK [Push image] **************************************************************
changed: [43.205.206.221]

TASK [Run container] ***********************************************************
changed: [43.205.206.221]

PLAY RECAP *********************************************************************
43.205.206.221             : ok=6    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Now copy the IP address of Jenkins and paste it into the browser

jenkins-ip:8081/jpetstore



Step 8 — Kuberenetes Setup

Connect your machines to Putty or Mobaxtreme

Take-Two Ubuntu 20.04 instances one for k8s master and the other one for worker.

Install Kubectl on Jenkins machine also.

Kubectl on Jenkins to be installed

```
sudo apt update

sudo apt install curl

curl -LO https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl

sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

kubectl version --client
```

Part 1 ———-Master Node————

```
sudo su

hostname master

bash

clear
```

———-Worker Node————

```
sudo su

hostname worker

bash

clear
```

Part 2 ————Both Master & Node ————

```
sudo apt-get update

sudo apt-get install -y docker.io

sudo usermod –aG docker Ubuntu

newgrp docker

sudo chmod 777 /var/run/docker.sock

sudo curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

sudo tee /etc/apt/sources.list.d/kubernetes.list <<EOF

deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF

sudo apt-get update

sudo apt-get install -y kubelet kubeadm kubectl

sudo snap install kube-apiserver
```

Part 3 ————— Master —————

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
```

# in case your in root exit from it and run below commands

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

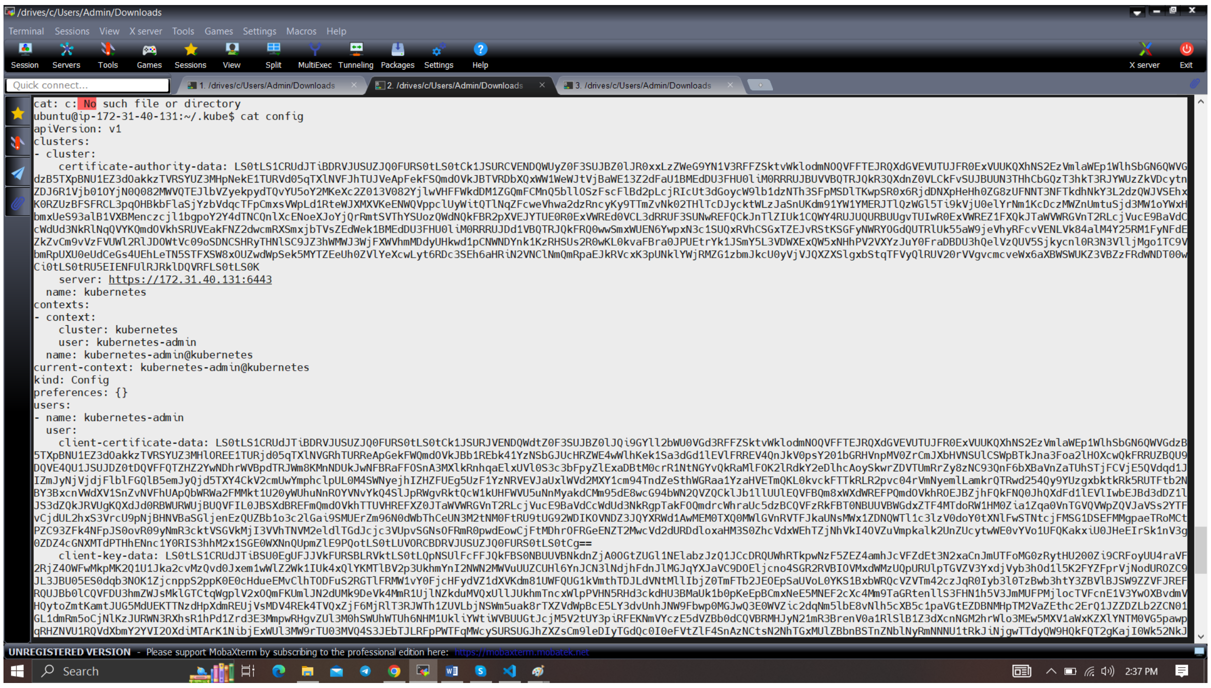
sudo chown $(id -u):$(id -g) $HOME/.kube/config

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

———-Worker Node————

sudo kubeadm join <master-node-ip>:<master-node-port> --token <token> --discovery-token-ca-cert-hash <hash>

Copy the config file to Jenkins master or the local file manager and save it


/drives/c/Users/Admin/Downloads
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
1. /drives/c/Users/Admin/Downloads
2. /drives/c/Users/Admin/Downloads
3. /drives/c/Users/Admin/Downloads
Search
2:37 PM

```
cat: c: No such file or directory
ubuntu@ip-172-31-40-131:~/.kube$ cat config
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJR0xxLzZWeG9YN1V3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWVG
dzB5TXpBNU1EZ3dOakkzTVRSYUZ3MHpNekE1TURVd05qTXlNVFJhTUJVeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUUN3THhCbGQzT3hkT3RJYWUzZkVDcytn
ZDJ6R1Vjb010YjN0Q082MWVQTEJlbVZyekpydTQvYU5oY2MKeXc2Z013V082YjlwVHFFWkdDM1ZGQmFCMnQ5bllOSzFscFlBd2pLcjRIcUt3dGoycW9lb1dzNTh3SFpMSDlTKwpSR0x6RjdDNXpHeHh0ZG8zUFNNT3NFTkdhNkY3L2dzQWJVSEhx
K0RZUzBFSFRCL3pqOHBkbFlaSjYzbVdqcTFpCmxsVWpLd1RteWJXMXVKeENWQVppclUyWitQTlNqZFcweVhwa2dzRncyKy9TTmZvNk02THlTcDJycktWLzJaSnUKdm91YW1YMERJTlQzWGl5Ti9kVjU0elYrNm1KcDczMWZnUmtuSjd3MW1oYWxH
bmxUeS93alB1VXBMenczcjl1bgpoY2Y4dTNCQnlXcENoeXJoYjQrRmtSVThYSUozQWdNQkFBR2pXVEJYTUE0R0ExVWREd0VCL3dRRUF3SUNwREFQQkJnTlZIUk1CQWY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJTaWVWRGVnT2RLcjVucE9BaVdC
cWdUd3NkRlNqQVYKQmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRQ0wwSmxWUEN6YwpxN3c1SUQxRVhCSGxTZEJvRStKSGFyNWRYOGdQUTRlUk55aW9jeVhyRFcvVENLVk84alM4Y25RM1FyNFdE
ZkZvCm9vVzFVUWl2RlJDOWtVc09oSDNCSHRyTHNlSC9JZ3hWMWJ3WjFXWVhmMDdyUHkwd1pCNWNDYnk1KzRHSUs2R0wKL0kvaFBra0JPUEtrYk1JSmY5L3VDWXExQW5xNHhPV2VXYzJuY0FraDBDU3hQelVzQUV5Sjkycnl0R3N3VlljMgo1TC9V
bmRpUXU0eUdCeGs4UEhLeTN5STFXSW8xOUZwdWpSek5MYTZEeUh0ZVlYeXcwLyt6RDc3SEh6aHRiN2VNClNmQmRpaEJkRVcxK3pUNklYWjRMZG1zbmJkcU0yVjVJQXZXSlgxbStqTFVyQlRUV20rVVgvcmcveWx6aXBWSWUKZ3VBZzFRdWNDT00w
Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K
    server: https://172.31.40.131:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURJVENDQWdtZ0F3SUJBZ0lJQi9GYll2bWU0VGd3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWVGdzB
5TXpBNU1EZ3dOakkzTVRSYUZ3MHl0REE1TURjd05qTXlNVGRhTURReApGekFWQmdOVkJBb1REbk41YzNSbGJUcHRZWE4wWlhKek1Sa3dGd1lEVlFRREV4QnJkV0psY201bGRHVnpMV0ZrCmJXbHVNSUlCSWpBTkJna3Foa2lHOXcwQkFRRUZBQU9
DQVE4QU1JSUJDZ0tDQVFFQTZHZ2YwNDhrWVBpdTRJWm8KMnNDUkJwNFBRaFFOSnA3MXlkRnhqaElxUVl0S3c3bFpyZlExaDBtM0crR1NtNGYvQkRaMlFOK2lRdkY2eDlhcAoySkwrZDVTUmRrZy8zNC93QnF6bXBaVnZaTUhSTjFCVjE5QVdqd1J
IZmJyNjVjdjFlblFGQlB5emJyQjd5TXY4CkV2cmUwYmphclpUL0M4SWNyejhIZHZFUEg5UzF1YzNRVEVJaUxlWVd2MXY1cm94TndZeSthWGRaa1YzaHVETmQKL0kvckFTTkRLR2pvc04rVmNyemlLamkrQTRwd254Qy9YUzgxbktkRk5RUTFtb2N
BY3BxcnVWdXV1SnZvNVFhUApQbWRWa2FMMkt1U20yWUhuNnROYVNvYkQ4SlJpRWgvRktQcW1kUHFWVU5uNnMyakdCMm95dE8wcG94bWN2QVZQCklJb1llUUlEQVFBQm8xWXdWREFPQmdOVkhROEJBZjhFQkFNQ0JhQXdFd1lEVlIwbEJBd3dDZ1l
JS3dZQkJRVUgKQXdJd0RBWURWUjBUQVFIL0JBSXdBREFmQmdOVkhTTUVHREFXZ0JTaWVWRGVnT2RLcjVucE9BaVdCcWdUd3NkRgpTakFOQmdrcWhraUc5dzBCQVFzRkFBT0NBUUVBWGdxZTF4MTdoRW1HM0Zia1Zqa0VnTGVQVWpZQVJaVSs2YTF
vCjdUL2hxS3VrcU9pNjBHNVBaSGljenEzQUZBb1o3c2lGai9SMUErZm96N0dWbThCeUN3M2tNM0FtRU9tUG92WDIKOVNDZ3JQYXRWd1AwMEM0TXQ0MWlGVnRVTFJkaUNsMWx1ZDNQWTl1c3lzV0doY0tXNlFwSTNtcjFMSG1DSEFMMgpaeTRoMCt
PZC93ZFk4NFpJS0ovR09yNmR3cktVSGVkMjI3VVhTNVM2eldlTGdJcjc3VUpvSGNsOFRmR0pwdEowCjFtMDhrOFRGeENZT2MwcVd2dURDdloxaHM3S0ZhcVdxWEhTZjNhVkI4OVZuVmpkalk2UnZUcytwWE0vYVo1UFQKakxiU0JHeEIrSk1nV3g
0ZDZ4cGNXMTdPTHhENnc1Y0RIS3hhM2x1SGE0WXNnQUpmZlE9PQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
    client-key-data: LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFcFFJQkFBS0NBUUVBNkdnZjA0OGtZUGl1NElabzJzQ1JCcDRQUWhRTkpwNzF5ZEZ4amhJcVFZdEt3N2xaCnJmUTFoMG0zRytHU200Zi9CRFoyUU4raVF
2RjZ4OWFwMkpMK2Q1U1Jka2cvMzQvd0Jxem1wWlZ2Wk1IUk4xQlYKMTlBV2p3UkhmYnI2NWN2MWVuUUZCUHl6YnJCN3lNdjhFdnJlMGJqYXJaVC9DOEljcno4SGR2RVBIOVMxdWMzUQpURUlpTGVZV3YxdjVyb3hOd1l5K2FYZFprVjNodURORC9
JL3JBU05ES0dqb3NOK1ZjcnppS2ppK0E0cHdueEMvClhTODFuS2RGTlFRMW1vY0FjcHFydVZ1dXVKdm81UWFQUG1kVmthTDJLdVNtMllIbjZ0TmFTb2JEOEpSaUVoL0YKS1BxbWRQcVZVTm42czJqR0Iyb3l0TzBwb3htY3ZBVlBJSW9ZZVFJREF
RQUJBb0lCQVFDU3hmZWJsMklGTCtqWgplV2x0QmFKUmlJN2dUMk9DeVk4MmR1UjlNZkduMVQxUllJUkhmTncxWlpPVHN5RHd3ckdHU3BMaUk1b0pKeEpBCmxNeE5MNEF2cXc4Mm9TaGRtenllS3FHN1h5V3JmMUFPMjlocTVFcnE1V3YwOXBvdmV
HQytoZmtKamtJUG5MdUEKTTNzdHpXdmREUjVsMDV4REk4TVQxZjF6MjRlT3RJWTh1ZUVLbjNSWm5uak8rTXZVdWpBcE5LY3dvUnhJNW9Fbwp0MGJwQ3E0WVZic2dqNm5lbE8vNlh5cXB5c1paVGtEZDBNMHpTM2VaZEthc2ErQ1JZZDZLb2ZCN01
GL1dmRm5oCjNlKzJURWN3RXhsR1hPd1Zrd3E3MmpwRHgvZUl3M0hSWUhWTUh6NHM1UkliYWtiWVBUUGtJcjM5V2tUY3piRFEKNmVYczE5dVZBb0dCQVBRMHJyN21mR3BrenV0a1RlSlB1Z3dXcnNGM2hrWlo3MEw5MXV1aWxKZXlYNTM0VG5pawp
qRHZNVU1RQVdXbmY2YVI2OXdiMTArK1NibjExWUl3MW9rTU03MVQ4S3JEbTJLRFpPWTFqMWcySURSUGJhZXZsCm9leDIyTGdQc0I0eFVtZlF4SnAzNCtsN2NhTGxMUlZBbnBSTnZNblNyRmNNNU1tRkJiNjgwTTdyQW9HQkFQT2gKajI0Wk52NkJ
```

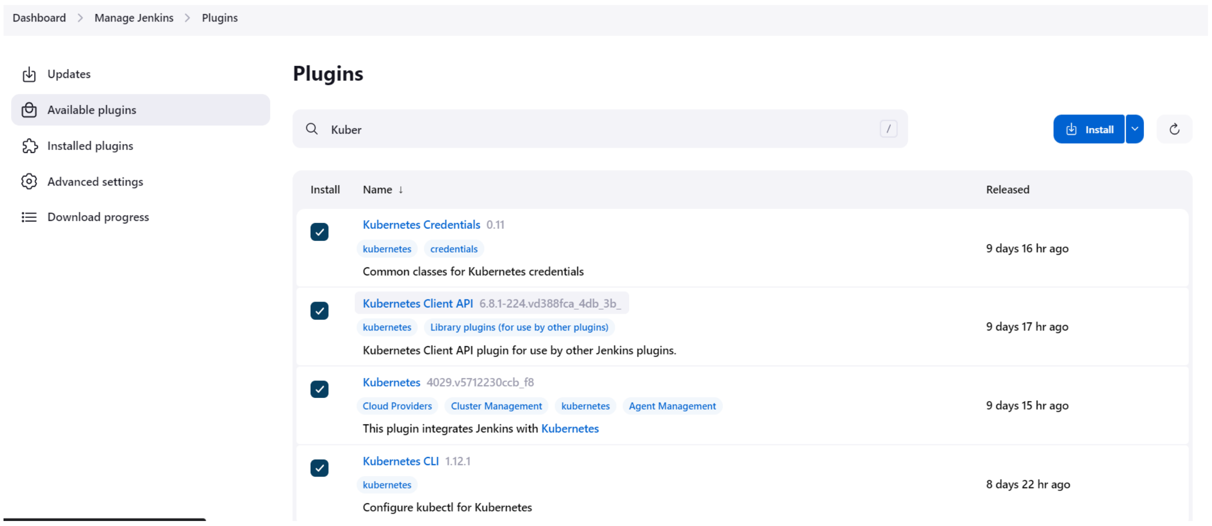
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

copy it and save it in documents or another folder save it as secret-file.txt

NOTE:

create a new textfile for the config file as secret-file.txt,
store the copied above complete config details and add it in the
credentials section.

Install Kubernetes Plugin, Once it's installed successfully


Dashboard > Manage Jenkins > Plugins

Updates
Available plugins
Installed plugins
Advanced settings
Download progress

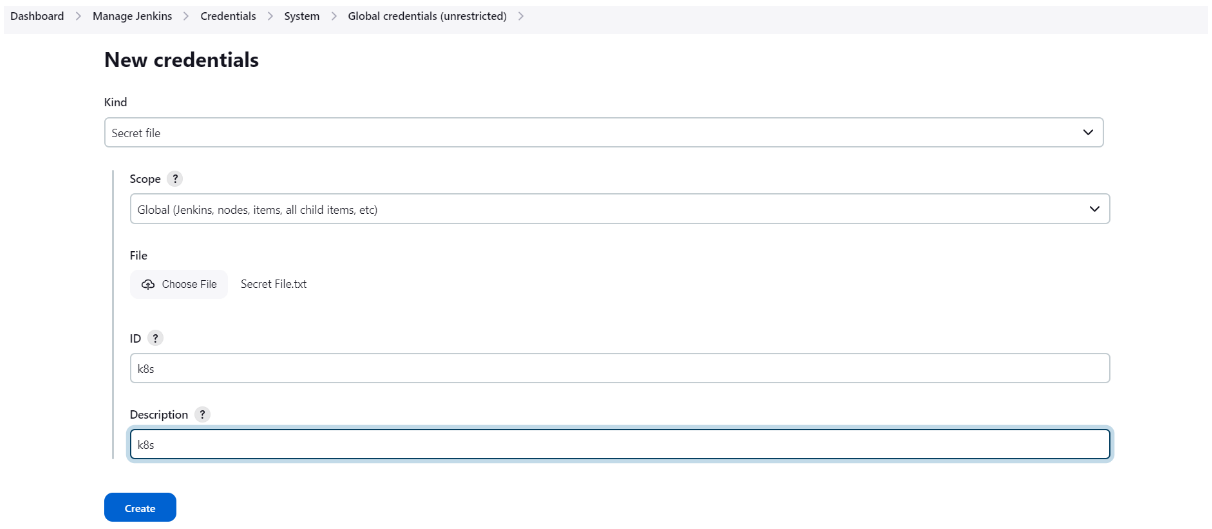
Plugins

Kuber

Install

Install	Name ↓	Released
✓	Kubernetes Credentials 0.11 kubernetes credentials Common classes for Kubernetes credentials	9 days 16 hr ago
✓	Kubernetes Client API 6.8.1-224.vd388fca_4db_3b_ kubernetes Library plugins (for use by other plugins) Kubernetes Client API plugin for use by other Jenkins plugins.	9 days 17 hr ago
✓	Kubernetes 4029.v5712230ccb_f8 Cloud Providers Cluster Management kubernetes Agent Management This plugin integrates Jenkins with Kubernetes	9 days 15 hr ago
✓	Kubernetes CLI 1.12.1 kubernetes Configure kubectl for Kubernetes	8 days 22 hr ago

go to manage Jenkins –> manage credentials –> Click on Jenkins global –> add credentials


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

New credentials

Kind: Secret file

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

File: Choose File Secret File.txt

ID: k8s

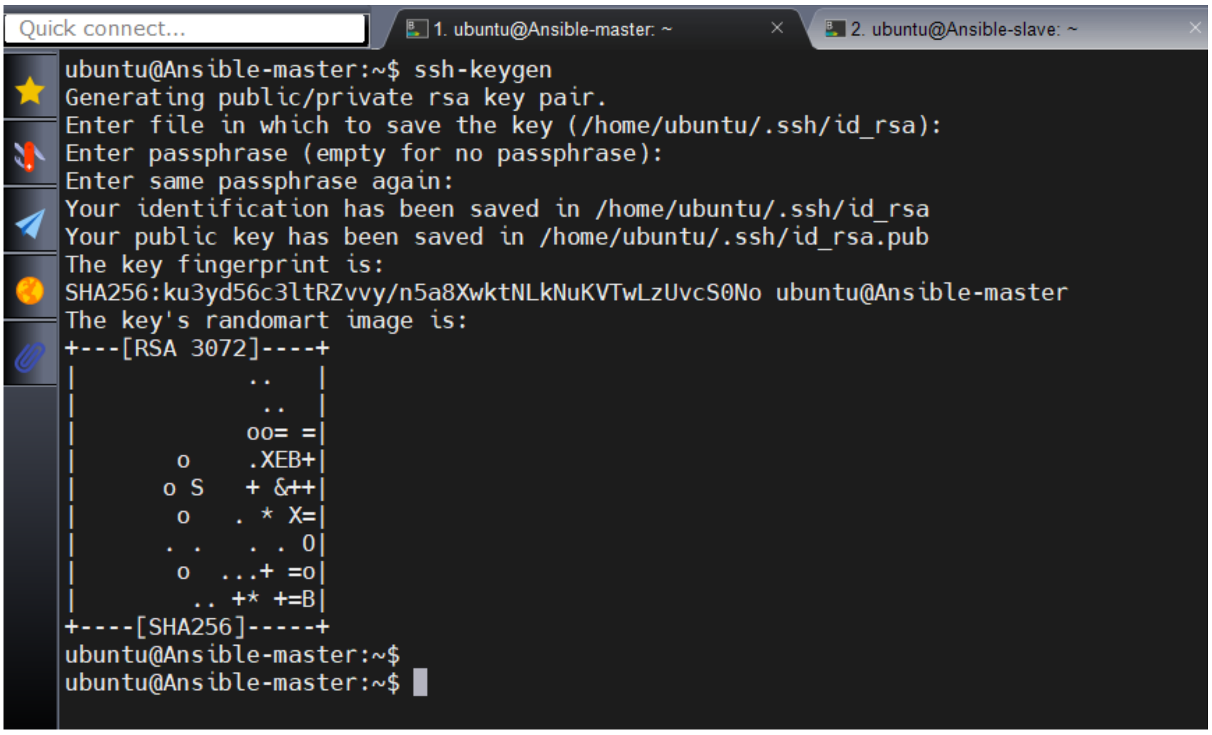
Description: k8s

Create

STEP 9 — Master-slave Setup for Ansible and Kubernetes

To communicate with the Kubernetes clients we have to generate an SSH key on the ansible master node and exchange it with K8s Master System.

ssh-keygen

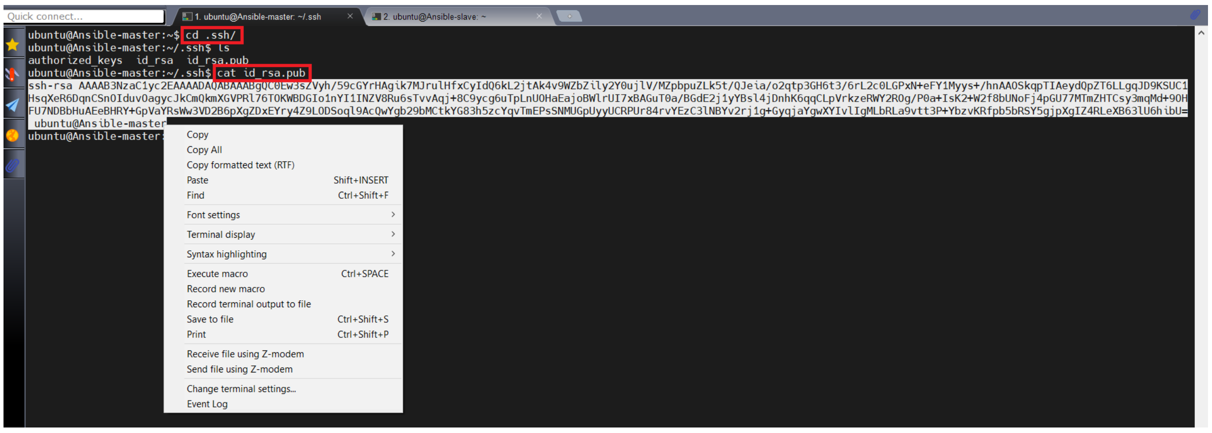


```
ubuntu@Ansible-master:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ku3yd56c3ltRZvvy/n5a8XwktNLkNuKVTwLzUvcS0No ubuntu@Ansible-master
The key's randomart image is:
+---[RSA 3072]----+
|            ..   |
|             ..  |
|            oo= =|
|      o     .XEB+|
|     o S    + &++|
|      o    . * X=|
|     . .    . . O|
|      o    ...+ =o|
|         .. +* +=B|
+----[SHA256]-----+
ubuntu@Ansible-master:~$
ubuntu@Ansible-master:~$
```

Change the directory to .ssh and copy the public key (id_rsa.pub)

cd .ssh

cat id_rsa.pub #copy this public key

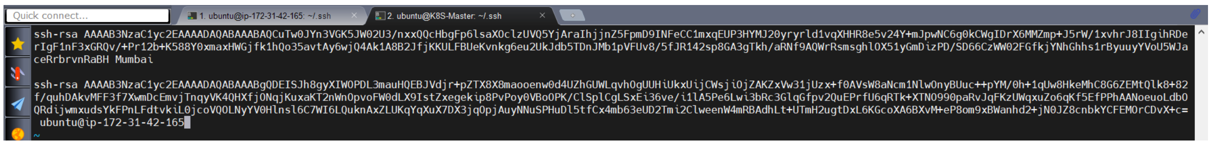


```
ubuntu@Ansible-master:~$ cd .ssh/
ubuntu@Ansible-master:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@Ansible-master:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC0Ew3sZVyh/59cGYrHAgik7MJrulHfxCyIdQ6kL2jtAk4v9WZbZily2Y0ujlV/MZpbpuZLk5t/QJeia/o2qtp3GH6t3/6rL2c0LGPxN+eFY1Myys+/hnAA0SkqpTIAeydQpZT6LLgqJD9KSUC1
HsqXeR6DqnCSn0Iduv0agycJkCmQkmXGVPRl76T0KWBDGIo1nYI1INZV8Ru6sTvvAqj+8C9ycg6uTpLnU0HaEajoBWlrUI7xBAGuT0a/BGdE2j1yYBsl4jDnhK6qqCLpVrkzeRWY2ROg/P0a+IsK2+W2f8bUNoFj4pGU77MTmZHTCsy3mqMd+9OH
FU7NDBbHuAEeBHRY+GpVaYRsWw3VD2B6pXgZDxEYry4Z9L0DSoql9AcQwYgb29bMCtkYG83h5zcYqvTmEPsSNMUGpUyyUCRPUr84rvYEzC3lNBYv2rj1g+GyqjaYgwXYIvlIgMLbRLa9vtt3P+YbzvKRfpb5bRSY5gjpXgIZ4RLeXB63lU6hibU=
 ubuntu@Ansible-master
ubuntu@Ansible-master:
```

Once you copy the public key from the Ansible master, move to the Kubernetes machine change the directory to .ssh and paste the copied public key under authorized_keys.

cd .ssh #on k8s master

sudo vi authorized_keys

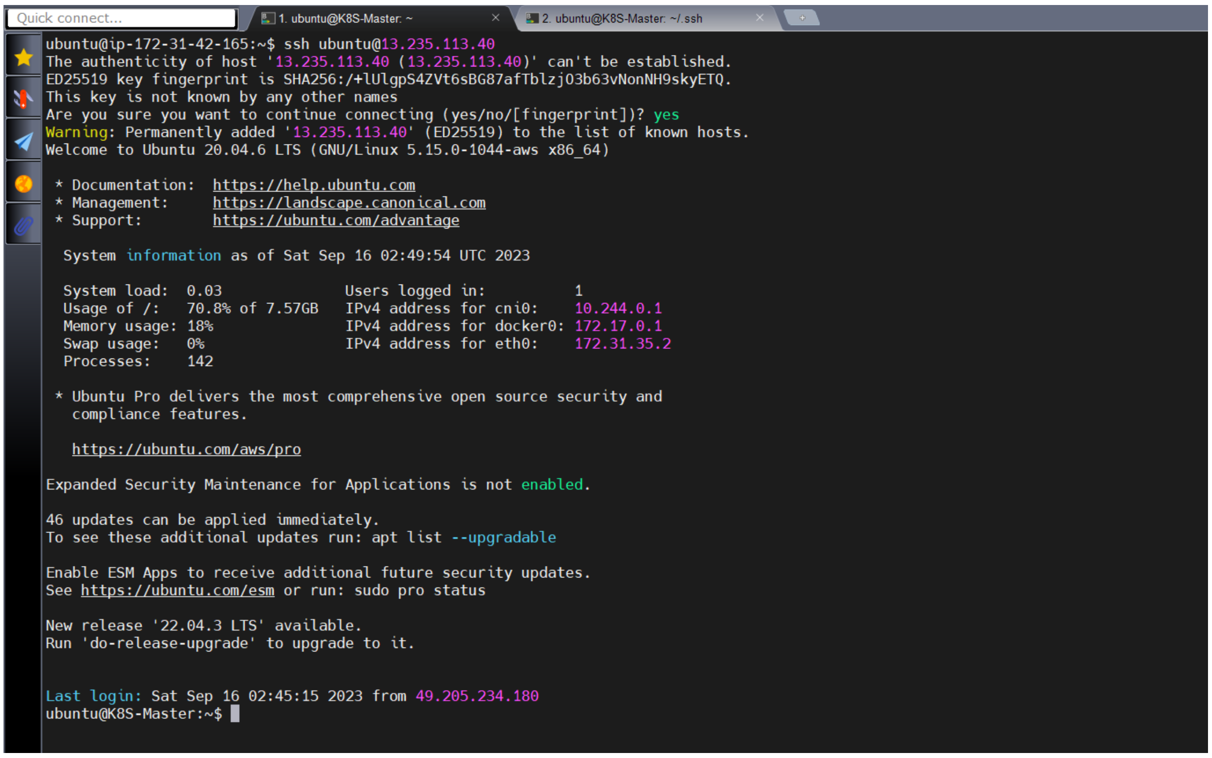

Quick connect...
1. ubuntu@ip-172-31-42-165: ~/.ssh
2. ubuntu@K8S-Master: ~/.ssh

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCuTw0JYn3VGK5JW02U3/nxxQQcHbgFp6lsaXOclzUVQ5YjAraIhjjnZ5FpmD9INFeCC1mxqEUP3HYMJ20yryrld1vqXHHR8e5v24Y+mJpwNC6g0kCWgIDrX6MMZmp+J5rW/1xvhrJ8IIgihRDe
rIgF1nF3xGRQv/+Pr12b+K588Y0xmaxHWGjfk1hQo35avtAy6wjQ4Ak1A8B2JfjKKULFBUeKvnkg6eu2UkJdb5TDnJMb1pVFUv8/5fJR142sp8GA3gTkh/aRNf9AQWrRsmsghl0X51yGmDizPD/SD66CzWW02FGfkjYNhGhhs1rByuuyYVoU5WJa
ceRrbrvnRaBH Mumbai

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDEISJh8gyXIWOPDL3mauHQEBJVdjr+pZTX8X8maooenw0d4UZhGUWLqvhOgUUHiUkxUijCWsji0jZAKZxVw31jUzx+f0AVsW8aNcm1NlwOnyBUuc++pYM/0h+1qUw8HkeMhC8G6ZEMtQlk8+82
f/quhDAkvMFF3f7XwmDcEmvjTnqyVK4QHXfj0NqjKuxaKT2nWn0pvoFW0dLX9IstZxegekip8PvPoy0VBoOPK/ClSplCgLSxEi36ve/i1lA5Pe6Lwi3bRc3GlqGfpv2QuEPrfU6qRTk+XTN0990paRvJqFKzUWqxuZo6qKf5EfPPhAANoeuoLdbO
QRdijwmxudsYkFPnLFdtvkiL0jcoVQOLNyYV0Hlnsl6C7WI6LQuknAxZLUKqYqXuX7DX3jq0pjAuyNNuSPHuDl5tfCx4mb63eUD2Tmi2ClweenW4mRBAdhLt+UTmH2ugtDxL6KGcoXA6BXvM+eP8om9xBWanhd2+jN0JZ8cnbkYCFEM0rCDvX+c=
 ubuntu@ip-172-31-42-165
~
```

Note: Now, insert or paste the copied public key into the new line. make sure don’t delete any existing keys from the authorized_keys file then save and exit.

By adding a public key from the master to the k8s machine we have now configured keyless access. To verify you can try to access the k8s master and use the command as mentioned in the below format.

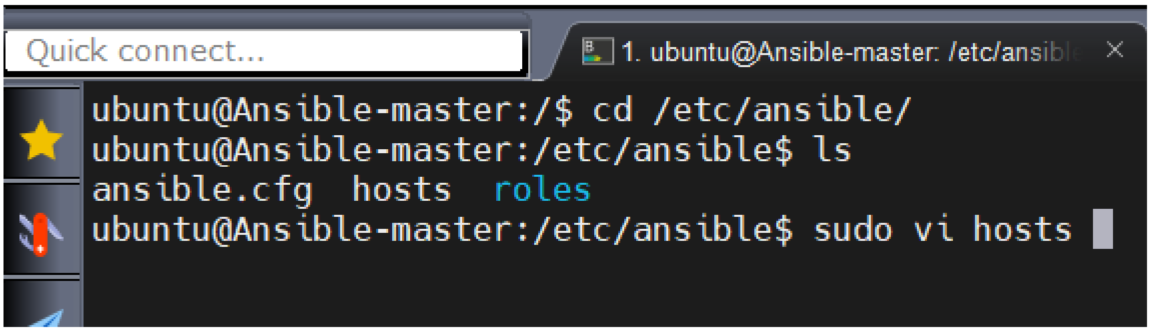
ssh ubuntu@<public-ip-k8s-master>



```
ubuntu@ip-172-31-42-165:~$ ssh ubuntu@13.235.113.40
The authenticity of host '13.235.113.40 (13.235.113.40)' can't be established.
ED25519 key fingerprint is SHA256:/+lUlgpS4ZVt6sBG87afTblzj03b63vNonNH9skyETQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.235.113.40' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1044-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Sep 16 02:49:54 UTC 2023

  System load:  0.03               Users logged in:       1
  Usage of /:   70.8% of 7.57GB    IPv4 address for cni0:    10.244.0.1
  Memory usage: 18%                IPv4 address for docker0: 172.17.0.1
  Swap usage:   0%                 IPv4 address for eth0:    172.31.35.2
  Processes:    142

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

46 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Sat Sep 16 02:45:15 2023 from 49.205.234.180
ubuntu@K8S-Master:~$
```

Verifying the above SSH connection from the master to the Kubernetes we have configured our prerequisites.

Now go to the host file inside the Ansible server and paste the public IP of the k8s master.

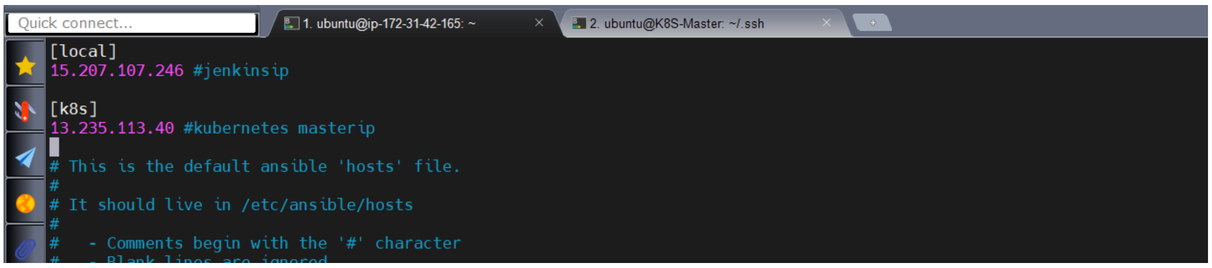


```
ubuntu@Ansible-master:/$ cd /etc/ansible/
ubuntu@Ansible-master:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@Ansible-master:/etc/ansible$ sudo vi hosts
```

You can create a group and paste ip address below:

[k8s]#any name you want

public ip of k8s-master



```
[local]
15.207.107.246 #jenkinsip

[k8s]
13.235.113.40 #kubernetes masterip

# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
```

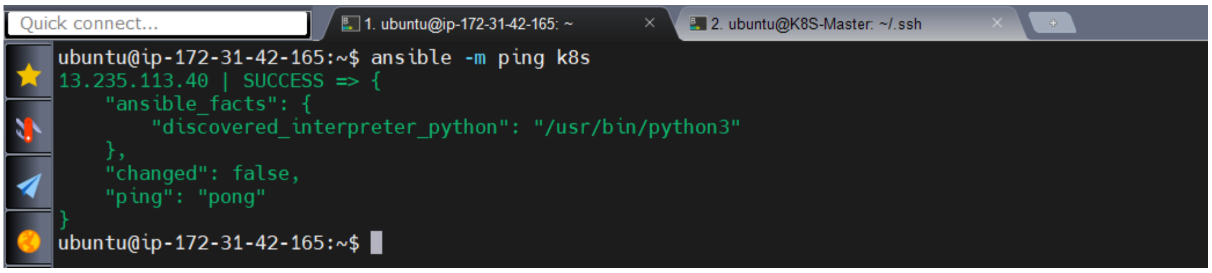
Test Ansible Master Slave Connection

Use the below command to check Ansible master-slave connections.

ansible -m ping k8s

ansible -m ping all#use this one

If all configuration is correct then you would get below output.



```
ubuntu@ip-172-31-42-165:~$ ansible -m ping k8s
13.235.113.40 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-42-165:~$
```

let’s create a simple ansible playbook for Kubernetes deployment.

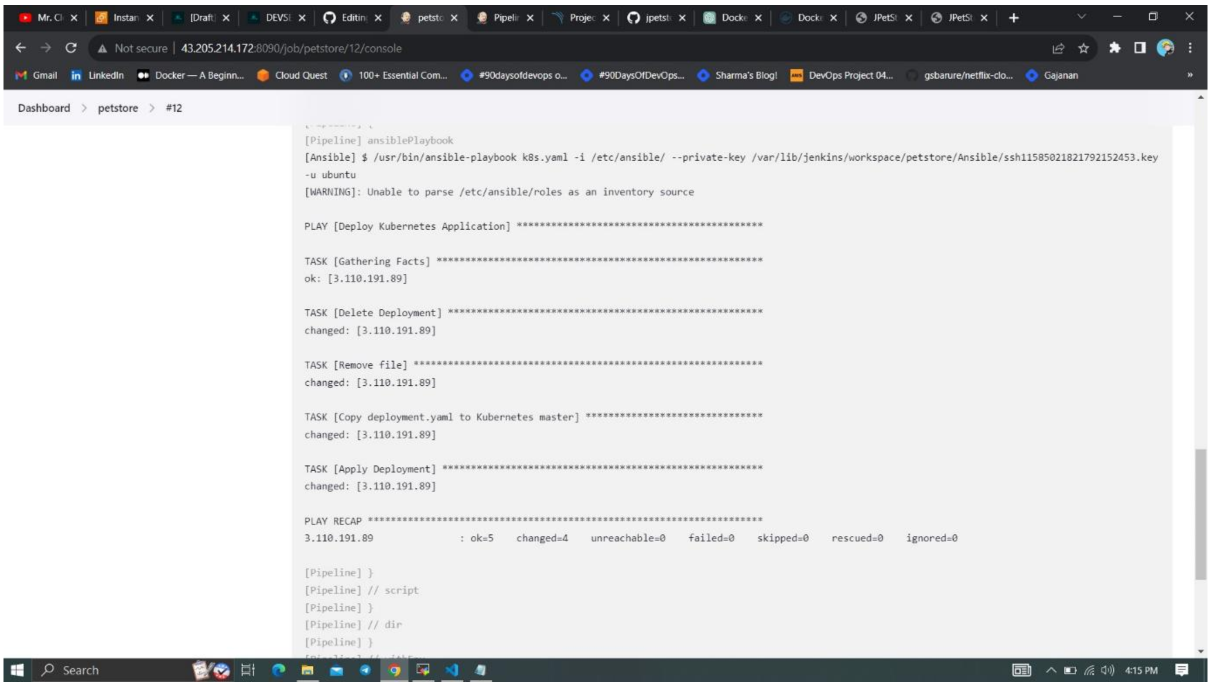
```
---
- name: Deploy Kubernetes Application
  hosts: k8s  # Replace with your target Kubernetes master host or group
  gather_facts: yes  # Gather facts about the target host
  tasks:
   - name:  deployment.yaml to Kubernetes master
     copy:
       src: /var/lib/jenkins/workspace/petstore/deployment.yaml  # Assuming Jenkins workspace variable
       dest: /home/ubuntu/
     become: yes  # Use sudo for copying if required
     become_user: root  # Use a privileged user for copying if required
   - name: Apply Deployment
     command: kubectl apply -f /home/ubuntu/deployment.yaml
```

Now add the below stage to your pipeline.

stage('k8s using ansible'){

```
          steps{
            dir('Ansible') {
              script{
                ansiblePlaybook credentialsId: 'ssh', disableHostKeyChecking: true, installation:
'ansible', inventory: '/etc/ansible/', playbook: 'kube.yaml'
              }
            }
          }
        }
```

output



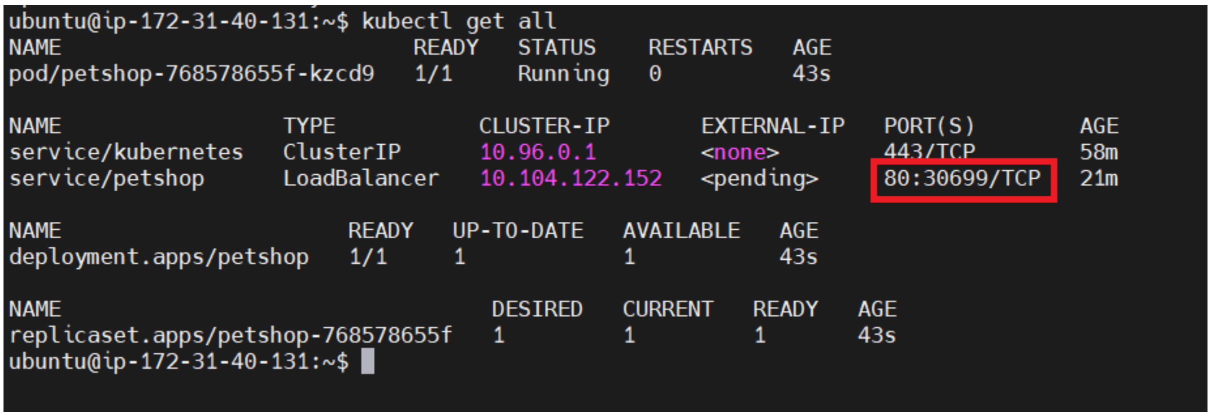
Dashboard > petstore > #12

```
[Pipeline] ansiblePlaybook
[Ansible] $ /usr/bin/ansible-playbook k8s.yaml -i /etc/ansible/ --private-key /var/lib/jenkins/workspace/petstore/Ansible/ssh11585021821792152453.key
-u ubuntu
[WARNING]: Unable to parse /etc/ansible/roles as an inventory source

PLAY [Deploy Kubernetes Application] *******************************************

TASK [Gathering Facts] *********************************************************
ok: [3.110.191.89]

TASK [Delete Deployment] *******************************************************
changed: [3.110.191.89]

TASK [Remove file] *************************************************************
changed: [3.110.191.89]

TASK [Copy deployment.yaml to Kubernetes master] *******************************
changed: [3.110.191.89]

TASK [Apply Deployment] ********************************************************
changed: [3.110.191.89]

PLAY RECAP *********************************************************************
3.110.191.89               : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
```

In the Kubernetes cluster give this command

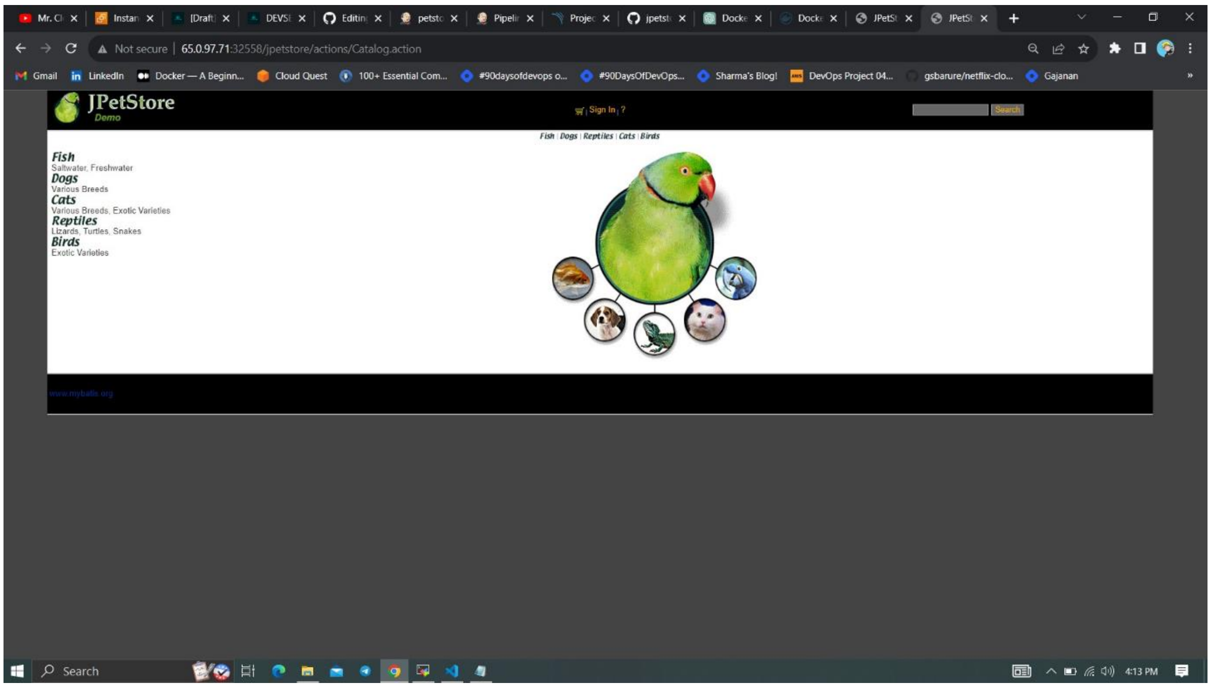
kubectl get all

kubectl get svc



```
ubuntu@ip-172-31-40-131:~$ kubectl get all
NAME                           READY   STATUS    RESTARTS   AGE
pod/petshop-768578655f-kzcd9   1/1     Running   0          43s

NAME                   TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
service/kubernetes     ClusterIP      10.96.0.1        <none>        443/TCP        58m
service/petshop        LoadBalancer   10.104.122.152   <pending>     80:30699/TCP   21m

NAME                      READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/petshop   1/1     1            1           43s

NAME                                 DESIRED   CURRENT   READY   AGE
replicaset.apps/petshop-768578655f   1         1         1       43s
ubuntu@ip-172-31-40-131:~$
```

slave-ip:serviceport(30699);/jpetstore



```
Not secure | 65.0.97.71:32558/jpetstore/actions/Catalog.action
```

complete Pipeline

```
pipeline{
    agent any
    tools {
        jdk 'jdk17'
        maven 'maven3'
    }
    environment {
        SCANNER_HOME=tool 'sonar-scanner'
    }
    stages{
```

```
stage ('clean Workspace'){
   steps{
      cleanWs()
   }
}
stage ('checkout scm') {
   steps {
      git 'https://github.com/Aj7Ay/jpetstore-6.git'
   }
}
stage ('maven compile') {
   steps {
      sh 'mvn clean compile'
   }
}
stage ('maven Test') {
   steps {
      sh 'mvn test'
   }
}
stage("Sonarqube Analysis "){
   steps{
      withSonarQubeEnv('sonar-server') {
         sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Petstore \
         -Dsonar.java.binaries=. \
         -Dsonar.projectKey=Petstore '''
      }
   }
}
stage("quality gate"){
   steps {
```

```
            script {
              waitForQualityGate abortPipeline: false, credentialsId: 'Sonar-token'
            }
        }
      }
      stage ('Build war file'){
         steps{
            sh 'mvn clean install -DskipTests=true'
         }
      }
      stage("OWASP Dependency Check"){
         steps{
            dependencyCheck additionalArguments: '--scan ./ --format XML ', odcInstallation: 'DP-
Check'
            dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
         }
      }
      stage('Ansible docker Docker') {
         steps {
            dir('Ansible'){
              script {
                   ansiblePlaybook credentialsId: 'ssh', disableHostKeyChecking: true, installation:
'ansible', inventory: '/etc/ansible/', playbook: 'docker.yaml'
               }
            }
         }
      }
      stage('k8s using ansible'){
         steps{
            dir('Ansible') {
               script{
                    ansiblePlaybook credentialsId: 'ssh', disableHostKeyChecking: true, installation: 
'ansible', inventory: '/etc/ansible/', playbook: 'kube.yaml'
                 }
              }
           }
        }
    }
}
```