Building a CI/CD Pipeline for a Retail Company (XYZ Technologies)

Post Graduate Program in DevOps

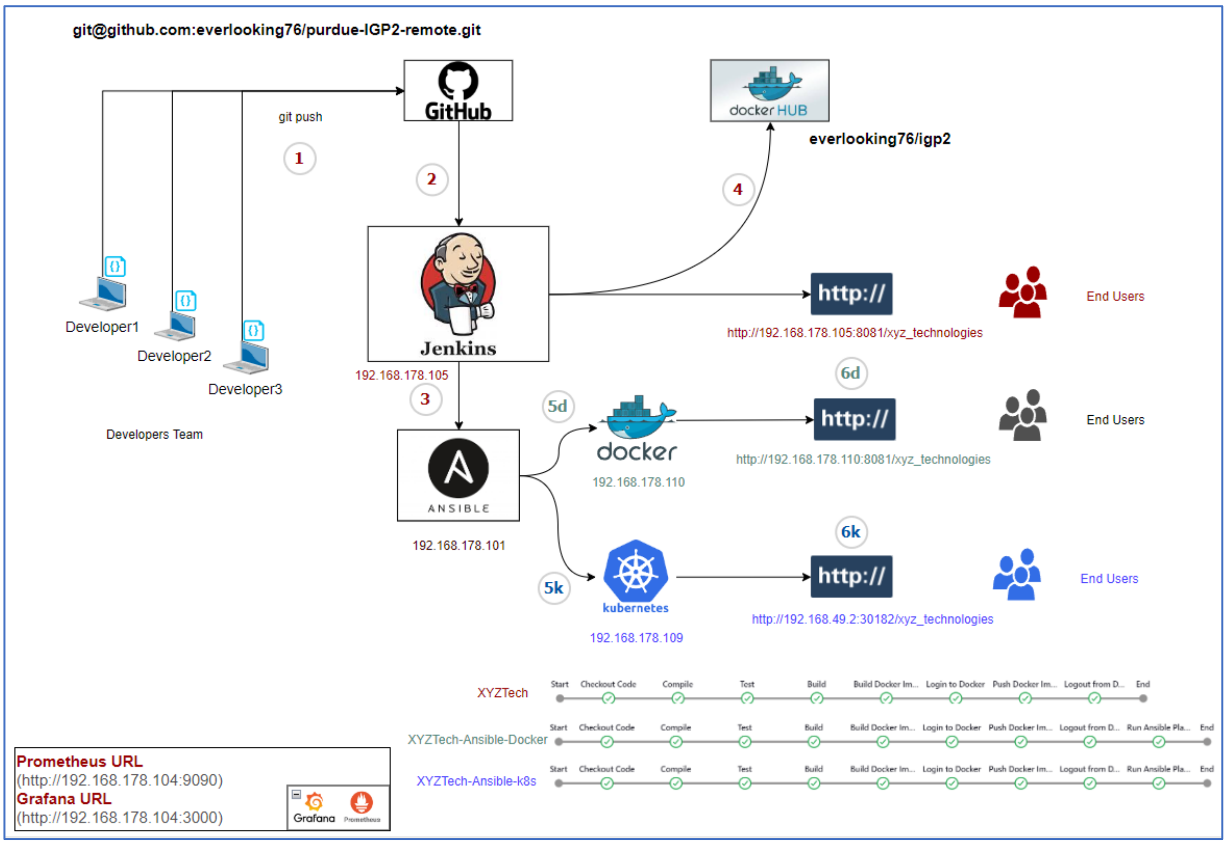
PROJECT II

Designed and implemented by:

Samer Mahayni

Network and Project Setup

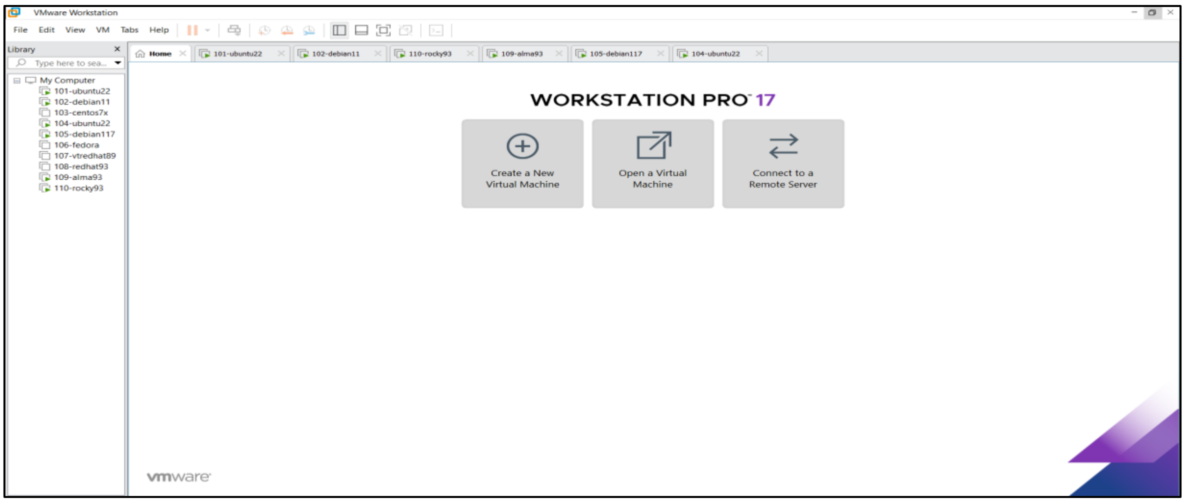
In order to demonstrate a solution for given Problem the following Network Setup is suggested:


git@github.com:everlooking76/purdue-IGP2-remote.git
git push
1
GitHub
2
docker HUB
everlooking76/igp2
4
Developer1
Developer2
Developer3
Developers Team
Jenkins
192.168.178.105
http://
End Users
http://192.168.178.105:8081/xyz_technologies
6d
3
5d
docker
192.168.178.110
http://
End Users
http://192.168.178.110:8081/xyz_technologies
ANSIBLE
192.168.178.101
6k
5k
kubernetes
192.168.178.109
http://
End Users
http://192.168.49.2:30182/xyz_technologies
XYZTech
Start, Checkout Code, Compile, Test, Build, Build Docker Im..., Login to Docker, Push Docker Im..., Logout from D..., End
XYZTech-Ansible-Docker
Start, Checkout Code, Compile, Test, Build, Build Docker Im..., Login to Docker, Push Docker Im..., Logout from D..., Run Ansible Pla..., End
XYZTech-Ansible-k8s
Start, Checkout Code, Compile, Test, Build, Build Docker Im..., Login to Docker, Push Docker Im..., Logout from D..., Run Ansible Pla..., End
Prometheus URL
(http://192.168.178.104:9090)
Grafana URL
(http://192.168.178.104:3000)
Grafana
Prometheus

All Linux Servers are hosted on VMWare Workstation 17 pro where:

196.168.178.1 **Router/DNS IP Address** **192.168.178.0/24**

IP Address	Server	Server OS	Major Software/Tools Installed
192.168.178.101	Ansible Server(master)	Ubuntu22	git – docker – Ansible – Prom.Node_Exporter
192.168.178.102	Linux Server	Debian11	git – docker – Prom.Node_Exporter
192.168.178.104	Prometheus/Grafana Server	Ubuntu22	git – docker – Prometheus – Grafana
192.168.178.105	Jenkins Server	Debian11	Jenkins – Maven – JDK – git – docker – Prom.Node_Exporter
192.168.178.109	Minikube Cluster Server	Almalinux	Minikube – kubectl – git – docker – Prom.Node_Exporter
192.168.178.110	Docker Server	Rockylinux	git – docker – Prom.Node_Exporter



Git-Hub Repository: git@github.com:everlooking76/purdue-IGP2-remote.git

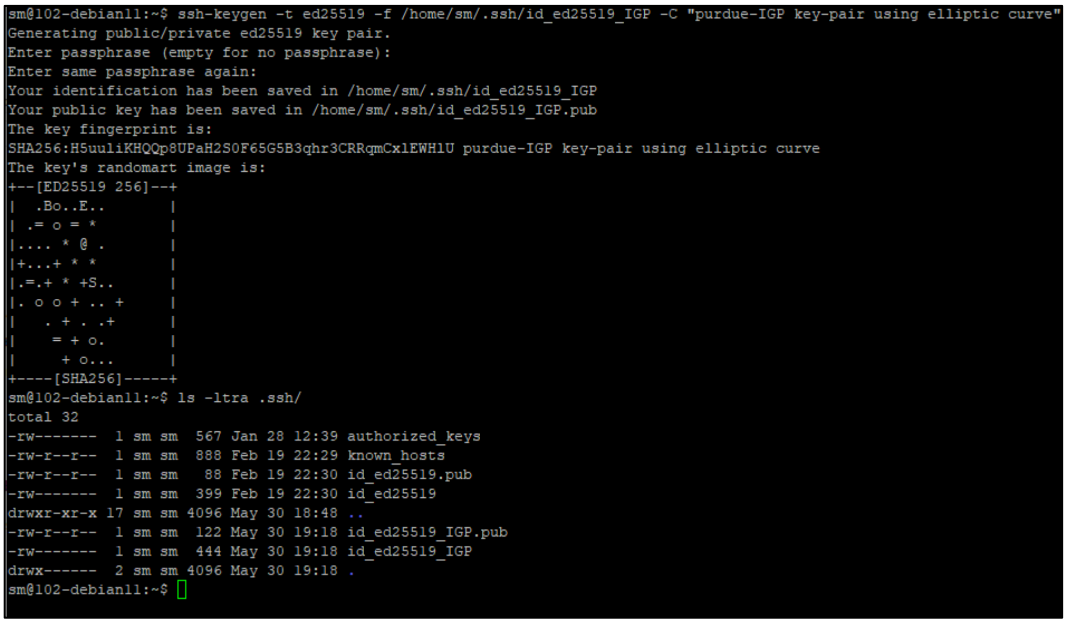
https://github.com/everlooking76/purdue-IGP2-remote.git

Docker-Hub Repository: everlooking76/igp2

1. Setting up git-Hub Repository and pushing up initial source code files

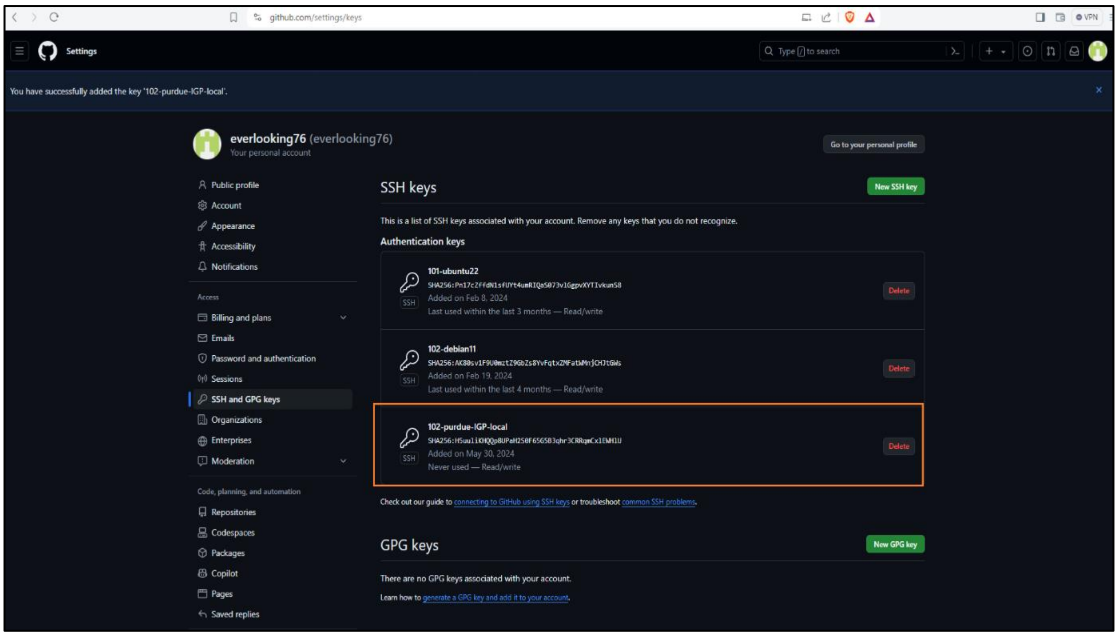
- On git-Hub a new public repository has been created git@github.com:everlooking76/purdue-IGP2-remote.git
- It is decided to use SSH Key-Pair to access this Repo following best-practice approach.

- Generating SSH Key-Pair using ed25519 encryption (**same** **Project I Procedures**)



```
sm@102-debian11:~$ ssh-keygen -t ed25519 -f /home/sm/.ssh/id_ed25519_IGP -C "purdue-IGP key-pair using elliptic curve"
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sm/.ssh/id_ed25519_IGP
Your public key has been saved in /home/sm/.ssh/id_ed25519_IGP.pub
The key fingerprint is:
SHA256:H5uuliKHQQp8UPaH2S0F65G5B3qhr3CRRqmCxlEWHlU purdue-IGP key-pair using elliptic curve
The key's randomart image is:
+--[ED25519 256]--+
|   .Bo..E..      |
|  .= o = *       |
|.... * @ .       |
|+...+ * *        |
|.=.+ * +S..      |
|. o o + .. +     |
|    . + . .+     |
|     = + o.      |
|      + o...     |
+----[SHA256]-----+
sm@102-debian11:~$ ls -ltra .ssh/
total 32
-rw-------  1 sm sm  567 Jan 28 12:39 authorized_keys
-rw-r--r--  1 sm sm  888 Feb 19 22:29 known_hosts
-rw-r--r--  1 sm sm   88 Feb 19 22:30 id_ed25519.pub
-rw-------  1 sm sm  399 Feb 19 22:30 id_ed25519
drwxr-xr-x 17 sm sm 4096 May 30 18:48 ..
-rw-r--r--  1 sm sm  122 May 30 19:18 id_ed25519_IGP.pub
-rw-------  1 sm sm  444 May 30 19:18 id_ed25519_IGP
drwx------  2 sm sm 4096 May 30 19:18 .
sm@102-debian11:~$ 
```

- Adding public key to git-Hub Repo.



github.com/settings/keys

You have successfully added the key '102-purdue-IGP-local'.

everlooking76 (everlooking76)
Your personal account

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

101-ubuntu22
SHA256:Pn17cZffdN1sfUYt4umRIQaS073vlGgpvXYTIvkunS8
Added on Feb 8, 2024
Last used within the last 3 months — Read/write

102-debian11
SHA256:AK80sv1F9U0mztZ9GbZs8YvFqtxZMFatWMnjCHJtGWs
Added on Feb 19, 2024
Last used within the last 4 months — Read/write

102-purdue-IGP-local
SHA256:H5uuliKHQQp8UPaH2S0F65G5B3qhr3CRRqmCx1EWH1U
Added on May 30, 2024
Never used — Read/write

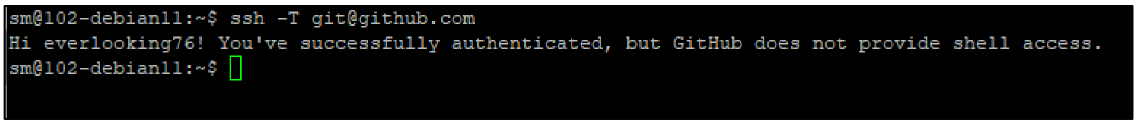
Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.

GPG keys

There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

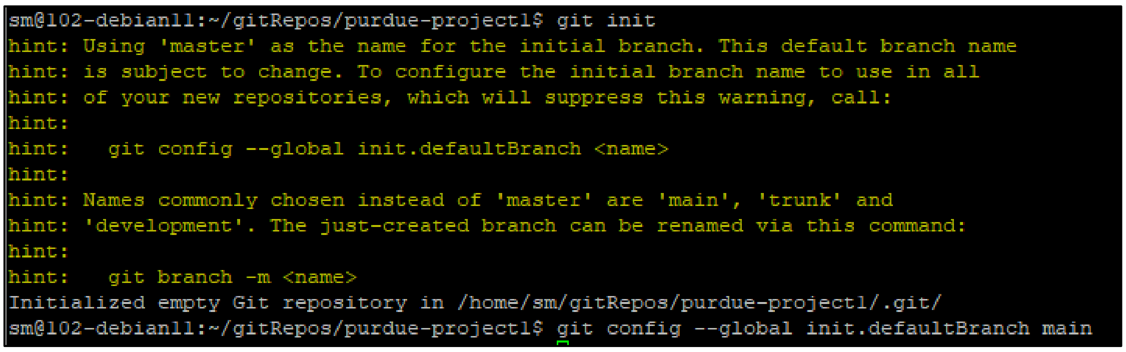
- Testing my SSH Connection with git-Hub



```
sm@102-debian11:~$ ssh -T git@github.com
Hi everlooking76! You've successfully authenticated, but GitHub does not provide shell access.
sm@102-debian11:~$
```

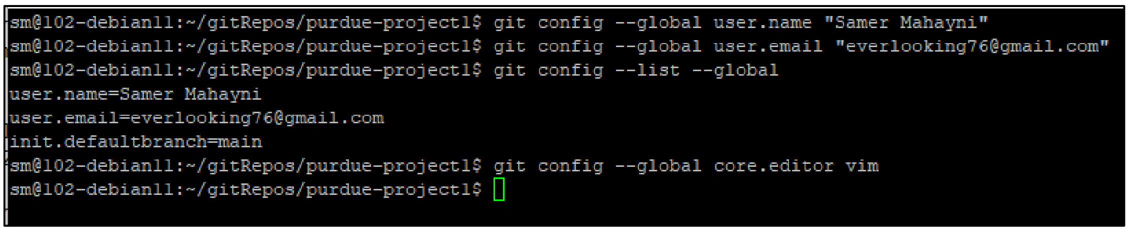
- The Source Code files have been copied to a local Linux Server(192.168.178.102) (see in NW diagram) via SFTP Client.

- On (`/home/sm/gitRepos/purdue-project2`) a local git Repository will be initialized, configured, and used to push up the SC files to the remote Repo. (**<u>same</u> <u>Project I Procedures</u>**)



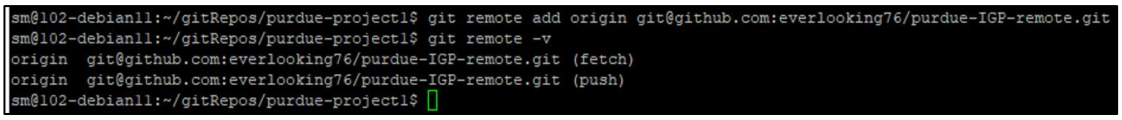
```
sm@102-debian11:~/gitRepos/purdue-project1$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/sm/gitRepos/purdue-project1/.git/
sm@102-debian11:~/gitRepos/purdue-project1$ git config --global init.defaultBranch main
```

- Configuring user.name and user.email



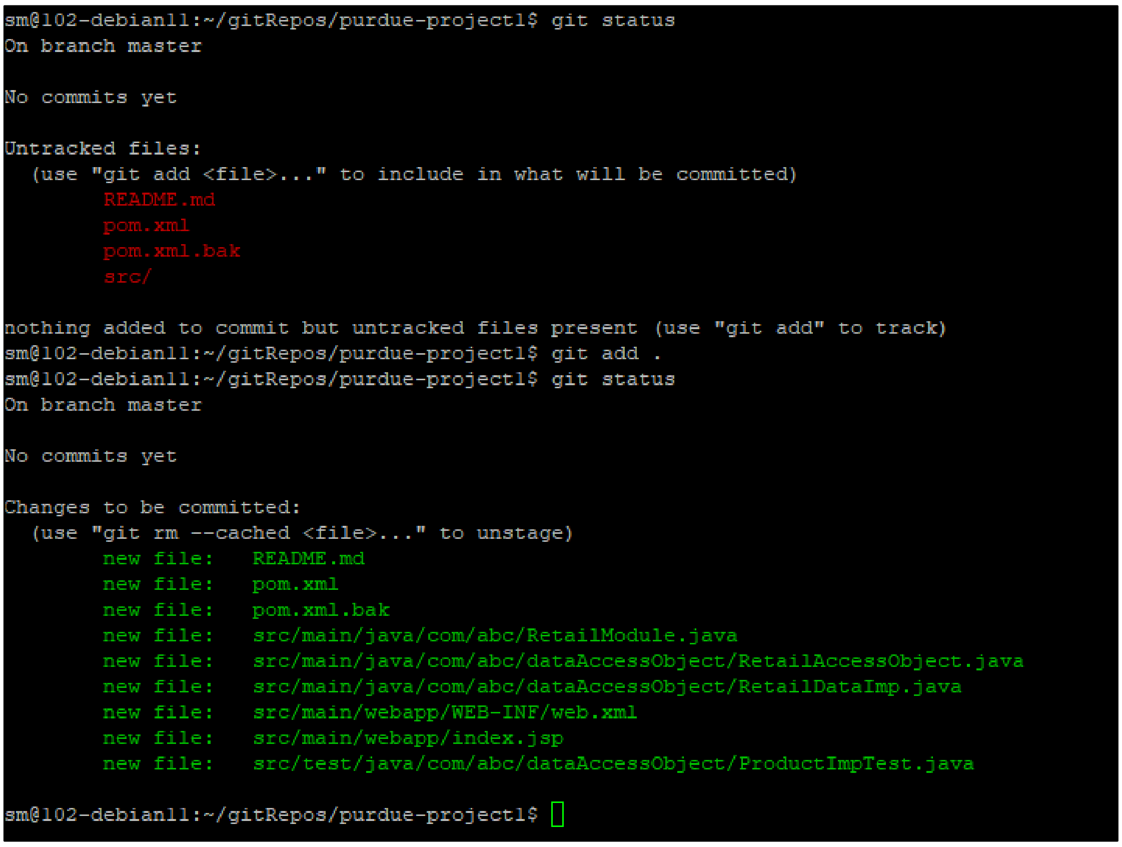
```
sm@102-debian11:~/gitRepos/purdue-project1$ git config --global user.name "Samer Mahayni"
sm@102-debian11:~/gitRepos/purdue-project1$ git config --global user.email "everlooking76@gmail.com"
sm@102-debian11:~/gitRepos/purdue-project1$ git config --list --global
user.name=Samer Mahayni
user.email=everlooking76@gmail.com
init.defaultbranch=main
sm@102-debian11:~/gitRepos/purdue-project1$ git config --global core.editor vim
sm@102-debian11:~/gitRepos/purdue-project1$
```

- Configuring git-Hub remote Repo



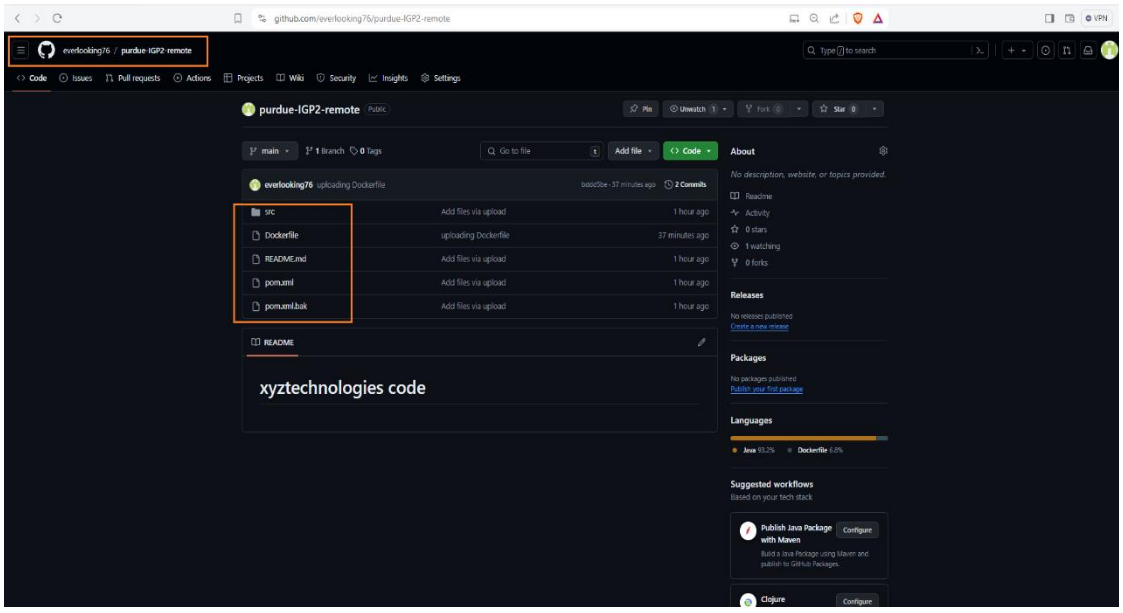
```
sm@102-debian11:~/gitRepos/purdue-project1$ git remote add origin git@github.com:everlooking76/purdue-IGP-remote.git
sm@102-debian11:~/gitRepos/purdue-project1$ git remote -v
origin  git@github.com:everlooking76/purdue-IGP-remote.git (fetch)
origin  git@github.com:everlooking76/purdue-IGP-remote.git (push)
sm@102-debian11:~/gitRepos/purdue-project1$
```

- Preparing local files for upload (Staging, Committing)



```
sm@102-debian11:~/gitRepos/purdue-project1$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md
        pom.xml
        pom.xml.bak
        src/

nothing added to commit but untracked files present (use "git add" to track)
sm@102-debian11:~/gitRepos/purdue-project1$ git add .
sm@102-debian11:~/gitRepos/purdue-project1$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
        new file:   pom.xml
        new file:   pom.xml.bak
        new file:   src/main/java/com/abc/RetailModule.java
        new file:   src/main/java/com/abc/dataAccessObject/RetailAccessObject.java
        new file:   src/main/java/com/abc/dataAccessObject/RetailDataImp.java
        new file:   src/main/webapp/WEB-INF/web.xml
        new file:   src/main/webapp/index.jsp
        new file:   src/test/java/com/abc/dataAccessObject/ProductImpTest.java

sm@102-debian11:~/gitRepos/purdue-project1$
```

- The files are uploaded to remote git-Hub Repo using this command (`git push -u origin`)
- On git-Hub website:



```
everlooking76 uploading Dockerfile          bddd5be · 37 minutes ago   2 Commits
src            Add files via upload          1 hour ago
Dockerfile     uploading Dockerfile          37 minutes ago
README.md      Add files via upload          1 hour ago
pom.xml        Add files via upload          1 hour ago
pom.xml.bak    Add files via upload          1 hour ago
```

README

xyztechnologies code

No description, website, or topics provided.

No releases published

No packages published

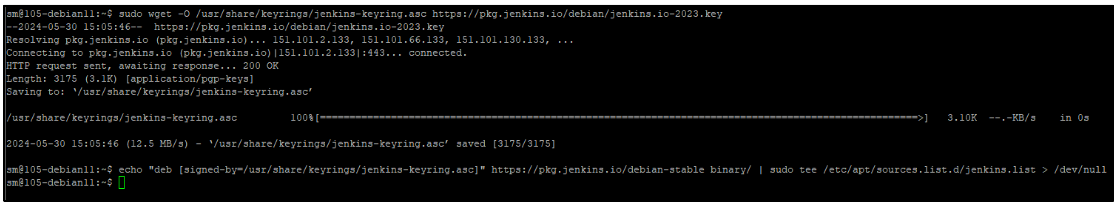
Build a Java Package using Maven and publish to GitHub Packages.

- The remote Repo (`git@github.com:everlooking76/purdue-IGP2-remote.git`) has been cloned using the command (`git clone git@github.com:everlooking76/purdue-IGP2-remote.git`) on the other servers of the network, and similar steps have been taken to authenticate and prepare SSH access.

2. Creating a CI pipeline using Jenkins to compile, test, and package the SC present in git-Hub

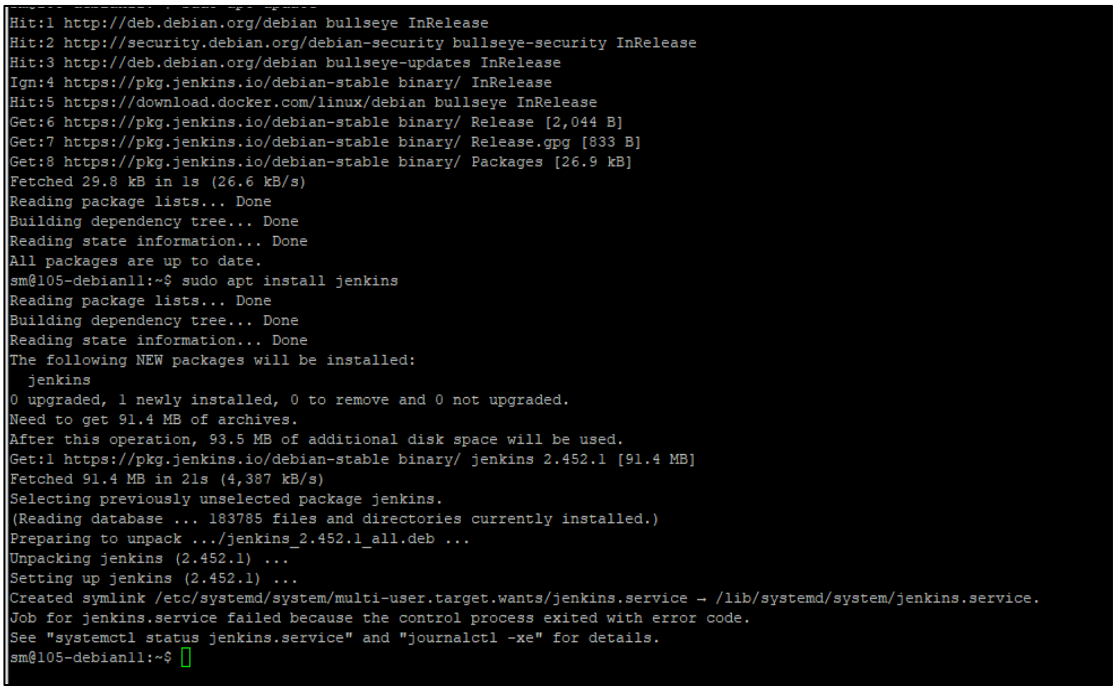
As prerequisite we need to install on the server (192.168.178.105) Jinkins, Maven, and JDK

- Installing and configuring Jenkins on (192.168.178.105)
  Installing keyring and adding Jenkins Repo.



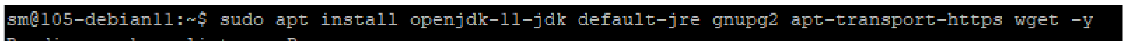
```
sm@105-debian11:~$ sudo wget -O /usr/share/keyrings/jenkins-keyring.asc https://pkg.jenkins.io/debian/jenkins.io-2023.key
--2024-05-30 15:05:46--  https://pkg.jenkins.io/debian/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.2.133, 151.101.66.133, 151.101.130.133, ...
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.2.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins-keyring.asc        100%[==========================================================================================>]   3.10K  --.-KB/s    in 0s

2024-05-30 15:05:46 (12.5 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]

sm@105-debian11:~$ 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
sm@105-debian11:~$
```

Installing Jenkins:



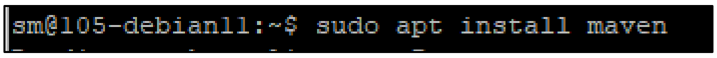
```
Hit:1 http://deb.debian.org/debian bullseye InRelease
Hit:2 http://security.debian.org/debian-security bullseye-security InRelease
Hit:3 http://deb.debian.org/debian bullseye-updates InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:5 https://download.docker.com/linux/debian bullseye InRelease
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release [2,044 B]
Get:7 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [26.9 kB]
Fetched 29.8 kB in 1s (26.6 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
sm@105-debian11:~$ sudo apt install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  jenkins
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 91.4 MB of archives.
After this operation, 93.5 MB of additional disk space will be used.
Get:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.452.1 [91.4 MB]
Fetched 91.4 MB in 21s (4,387 kB/s)
Selecting previously unselected package jenkins.
(Reading database ... 183785 files and directories currently installed.)
Preparing to unpack .../jenkins_2.452.1_all.deb ...
Unpacking jenkins (2.452.1) ...
Setting up jenkins (2.452.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Job for jenkins.service failed because the control process exited with error code.
See "systemctl status jenkins.service" and "journalctl -xe" for details.
sm@105-debian11:~$
```

Installing JDK



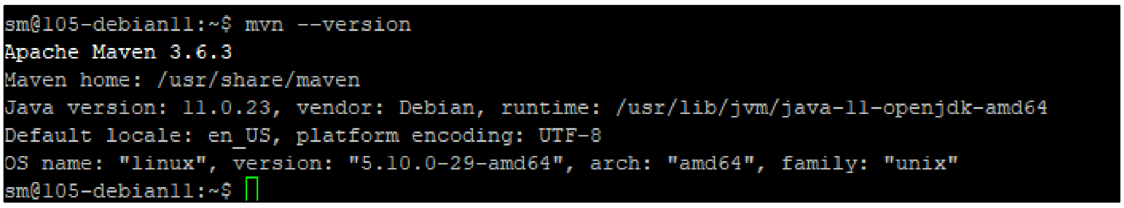
```
sm@105-debian11:~$ sudo apt install openjdk-11-jdk default-jre gnupg2 apt-transport-https wget -y
```

Installing maven



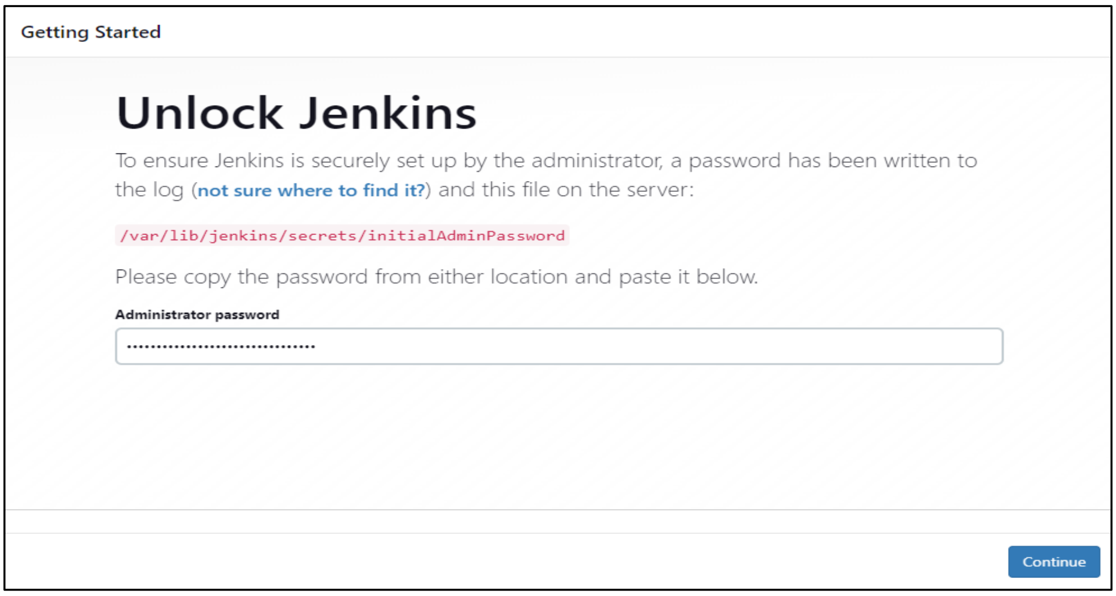
```
sm@105-debian11:~$ sudo apt install maven
```

To verify Maven is installed:



```
sm@105-debian11:~$ mvn --version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 11.0.23, vendor: Debian, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.10.0-29-amd64", arch: "amd64", family: "unix"
sm@105-debian11:~$
```

Configuring Jenkins:



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
```

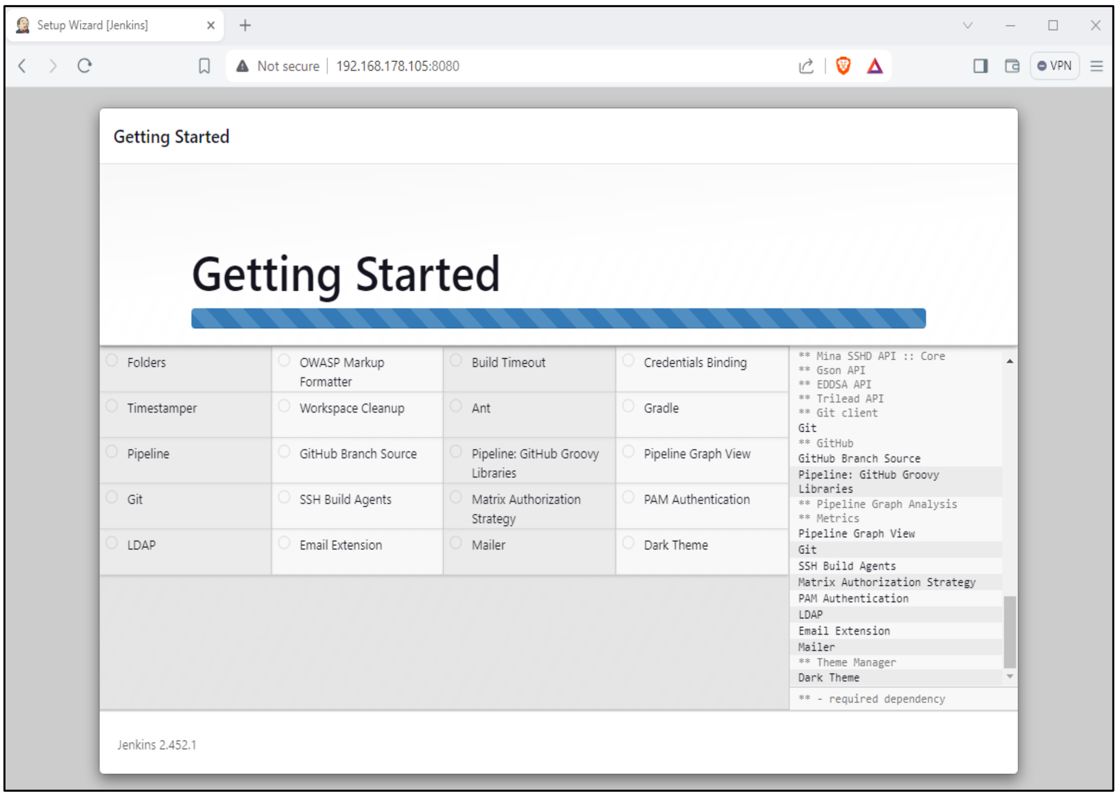
Please copy the password from either location and paste it below.

Administrator password

••••••••••••••••••••••••••••••••

Continue

Installing necessary Plug-ins and getting Started:



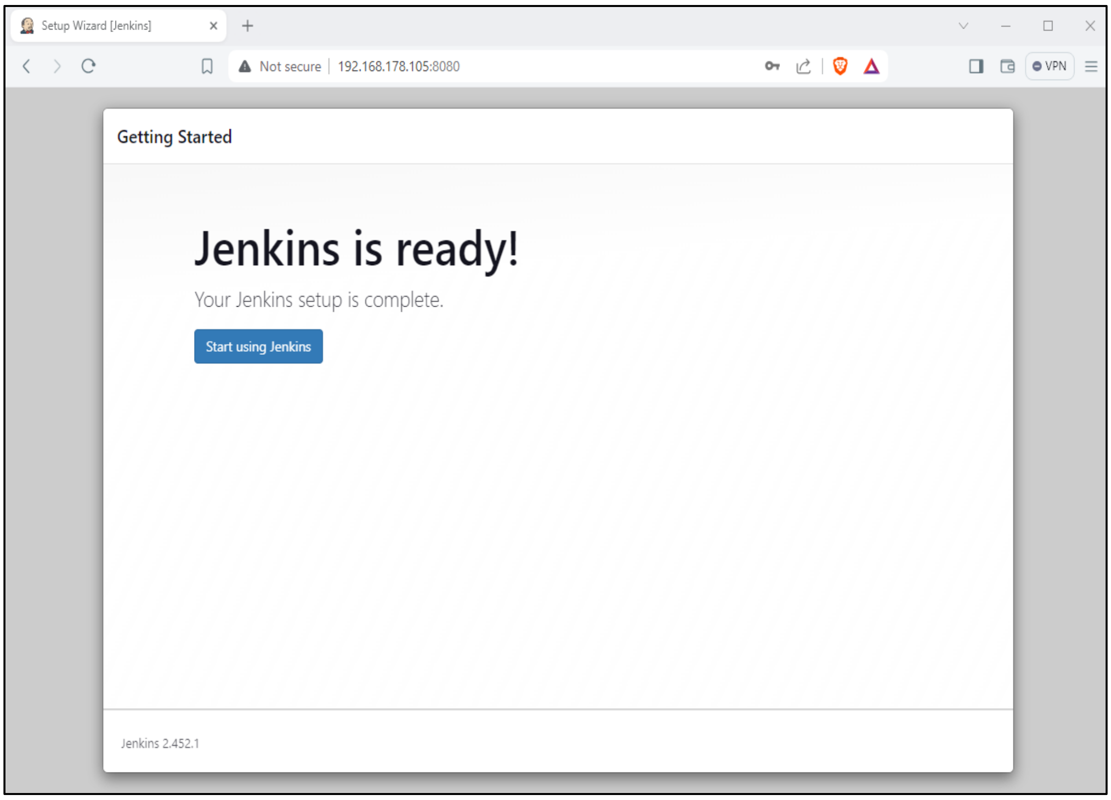
Getting Started

Getting Started

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

```
** Mina SSHD API :: Core
** Gson API
** EDDSA API
** Trilead API
** Git client
Git
** GitHub
GitHub Branch Source
Pipeline: GitHub Groovy
Libraries
** Pipeline Graph Analysis
** Metrics
Pipeline Graph View
Git
SSH Build Agents
Matrix Authorization Strategy
PAM Authentication
LDAP
Email Extension
Mailer
** Theme Manager
Dark Theme
** - required dependency
```

Jenkins 2.452.1



Getting Started

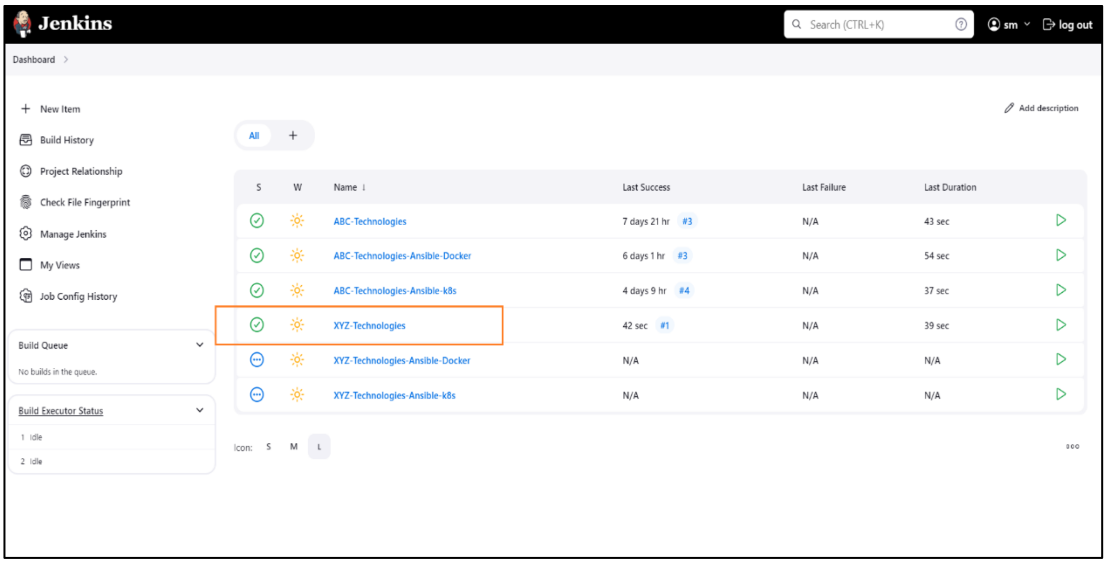
Jenkins is ready!

Your Jenkins setup is complete.

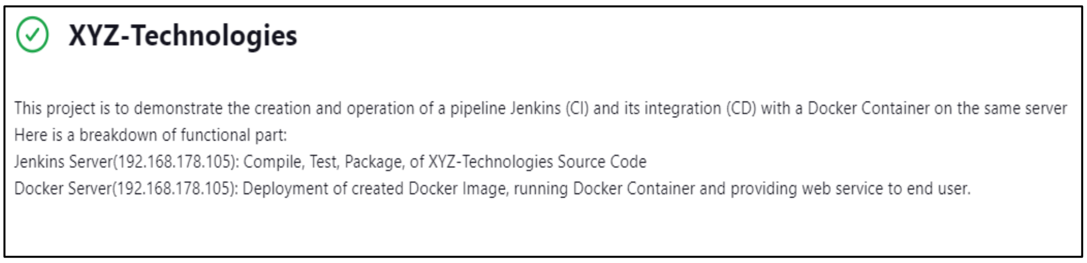
Start using Jenkins

Jenkins 2.452.1

- Creating the 1st pipeline: XYZ-Technologies



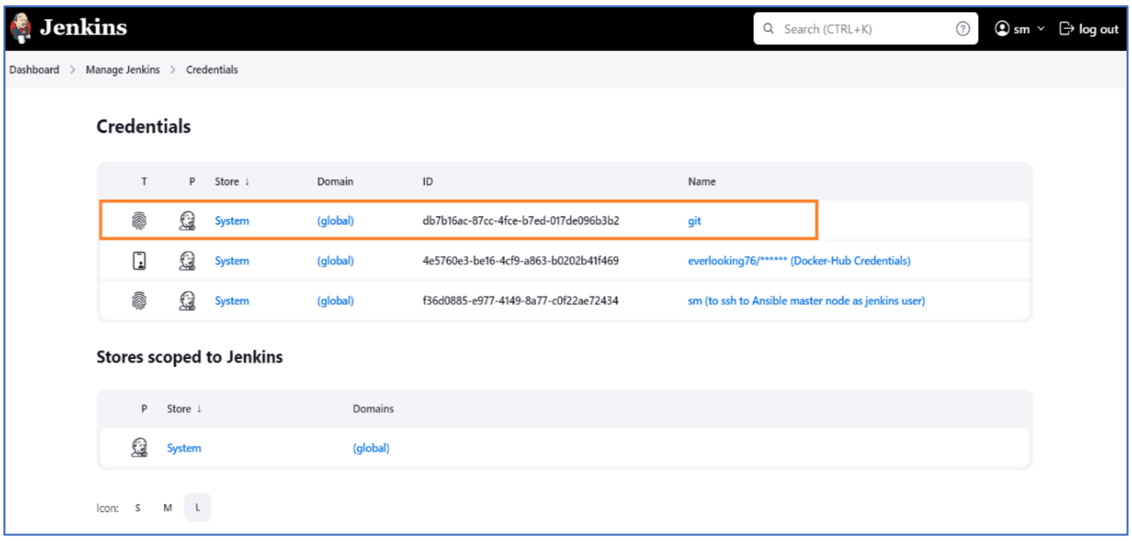
S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	ABC-Technologies	7 days 21 hr #3	N/A	43 sec
✓	☀	ABC-Technologies-Ansible-Docker	6 days 1 hr #3	N/A	54 sec
✓	☀	ABC-Technologies-Ansible-k8s	4 days 9 hr #4	N/A	37 sec
✓	☀	XYZ-Technologies	42 sec #1	N/A	39 sec
⋯	☀	XYZ-Technologies-Ansible-Docker	N/A	N/A	N/A
⋯	☀	XYZ-Technologies-Ansible-k8s	N/A	N/A	N/A



XYZ-Technologies

This project is to demonstrate the creation and operation of a pipeline Jenkins (CI) and its integration (CD) with a Docker Container on the same server
Here is a breakdown of functional part:
Jenkins Server(192.168.178.105): Compile, Test, Package, of XYZ-Technologies Source Code
Docker Server(192.168.178.105): Deployment of created Docker Image, running Docker Container and providing web service to end user.

- Git-Hub credentials are added



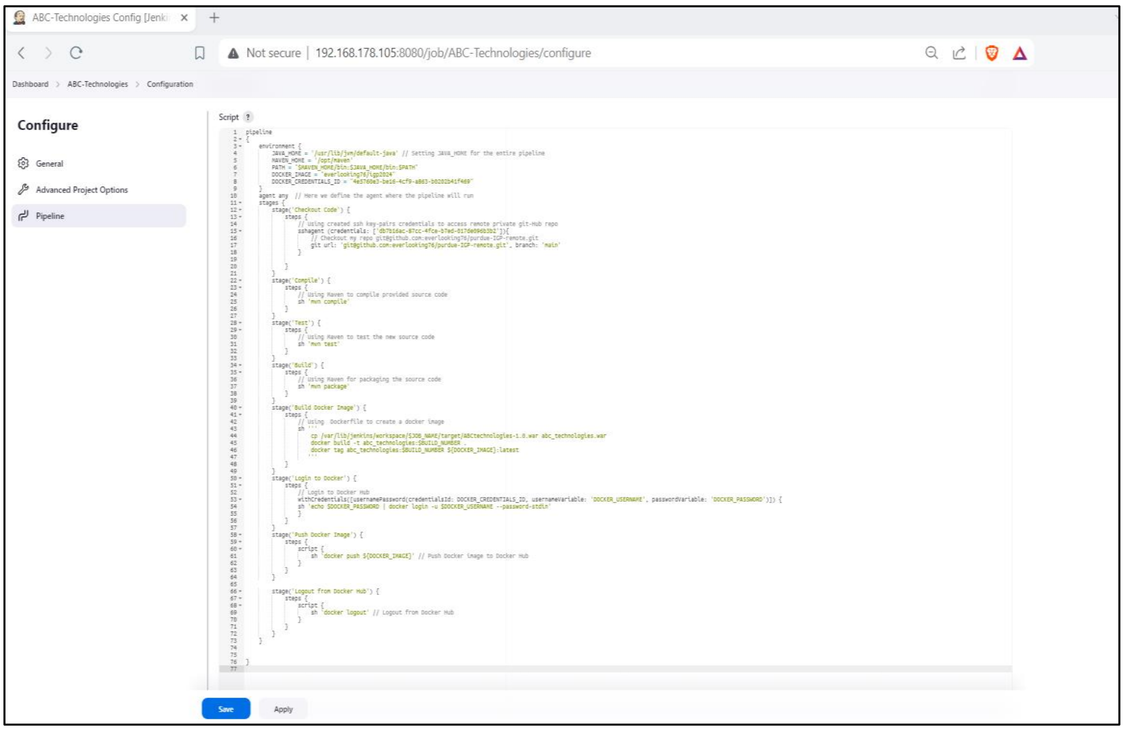
Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	db7b16ac-87cc-4fce-b7ed-017de096b3b2	git
		System	(global)	4e5760e3-be16-4cf9-a863-b0202b41f469	everlooking76/****** (Docker-Hub Credentials)
		System	(global)	f36d0885-e977-4149-8a77-c0f22ae72434	sm (to ssh to Ansible master node as jenkins user)

Stores scoped to Jenkins

P	Store ↓	Domains
	System	(global)

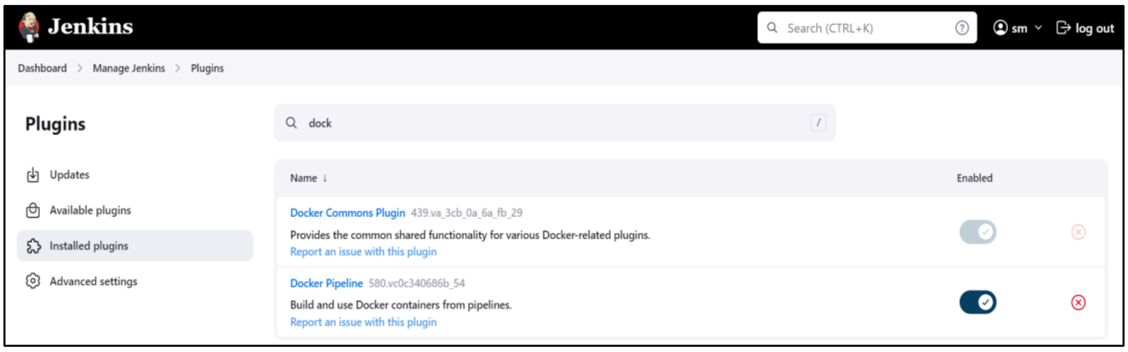
- Pipeline Code (Groovy Code): The code can be found in “jenkinsfile-XYZ_Technologies.txt” within project files. Here is a snapshot of the same pipeline code:



ABC-Technologies Config [Jenki
Not secure | 192.168.178.105:8080/job/ABC-Technologies/configure
Dashboard > ABC-Technologies > Configuration
Configure
General
Advanced Project Options
Pipeline
Script
Save
Apply

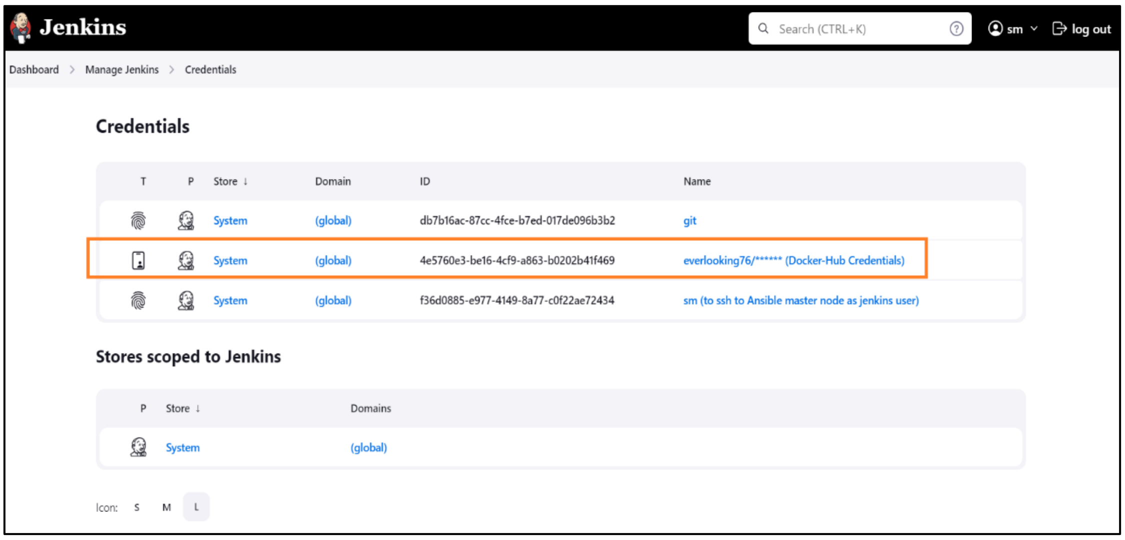
```
pipeline
{
    environment {
        JAVA_HOME = '/usr/lib/jvm/default-java' // Setting JAVA_HOME for the entire pipeline
        MAVEN_HOME = '/opt/maven'
        PATH = "$MAVEN_HOME/bin:$JAVA_HOME/bin:$PATH"
        DOCKER_IMAGE = "everlooking76/igp2024"
        DOCKER_CREDENTIALS_ID = "4e5760e3-be16-4cf9-a863-b0202b41f469"
    }
    agent any  // Here we define the agent where the pipeline will run
    stages {
        stage('Checkout Code') {
            steps {
                // Using created ssh key-pairs credentials to access remote private git-Hub repo
                sshagent (credentials: ['db7b16ac-87cc-4fce-b7ed-017de096b3b2']){
                    // Checkout my repo git@github.com:everlooking76/purdue-IGP-remote.git
                    git url: 'git@github.com:everlooking76/purdue-IGP-remote.git', branch: 'main'
                }

            }
        }
        stage('Compile') {
            steps {
                // Using Maven to compile provided source code
                sh 'mvn compile'
            }
        }
        stage('Test') {
            steps {
                // Using Maven to test the new source code
                sh 'mvn test'
            }
        }
        stage('Build') {
            steps {
                // Using Maven for packaging the source code
                sh 'mvn package'
            }
        }
        stage('Build Docker Image') {
            steps {
                // Using  Dockerfile to create a docker image
                sh '''
                    cp /var/lib/jenkins/workspace/$JOB_NAME/target/ABCtechnologies-1.0.war abc_technologies.war
                    docker build -t abc_technologies:$BUILD_NUMBER .
                    docker tag abc_technologies:$BUILD_NUMBER ${DOCKER_IMAGE}:latest
                   '''
            }
        }
        stage('Login to Docker') {
            steps {
                // Login to Docker Hub
                withCredentials([usernamePassword(credentialsId: DOCKER_CREDENTIALS_ID, usernameVariable: 'DOCKER_USERNAME', passwordVariable: 'DOCKER_PASSWORD')]) {
                sh 'echo $DOCKER_PASSWORD | docker login -u $DOCKER_USERNAME --password-stdin'
                }
            }
        }
        stage('Push Docker Image') {
            steps {
                script {
                    sh 'docker push ${DOCKER_IMAGE}' // Push Docker image to Docker Hub
                }
            }
        }

        stage('Logout from Docker Hub') {
            steps {
                script {
                    sh 'docker logout' // Logout from Docker Hub
                }
            }
        }
    }


}
```

- Docker plug-in is installed



Name ↓	Enabled
Docker Commons Plugin 439.va_3cb_0a_6a_fb_29 Provides the common shared functionality for various Docker-related plugins. Report an issue with this plugin	✓
Docker Pipeline 580.vc0c340686b_54 Build and use Docker containers from pipelines. Report an issue with this plugin	✓

- Docker-Hub credentials are added



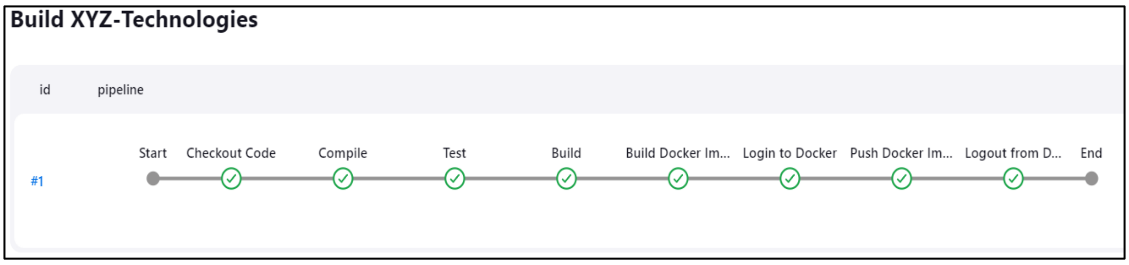
Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	db7b16ac-87cc-4fce-b7ed-017de096b3b2	git
		System	(global)	4e5760e3-be16-4cf9-a863-b0202b41f469	everlooking76/****** (Docker-Hub Credentials)
		System	(global)	f36d0885-e977-4149-8a77-c0f22ae72434	sm (to ssh to Ansible master node as jenkins user)

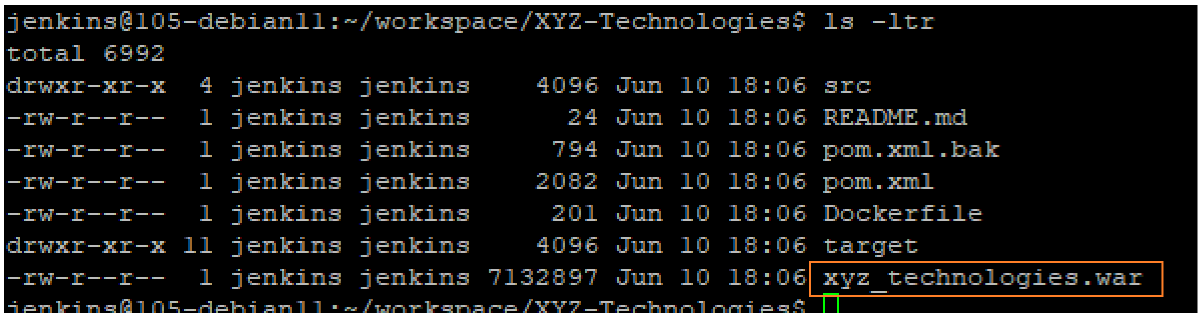
Stores scoped to Jenkins

P	Store ↓	Domains
	System	(global)

- The Build is successful!



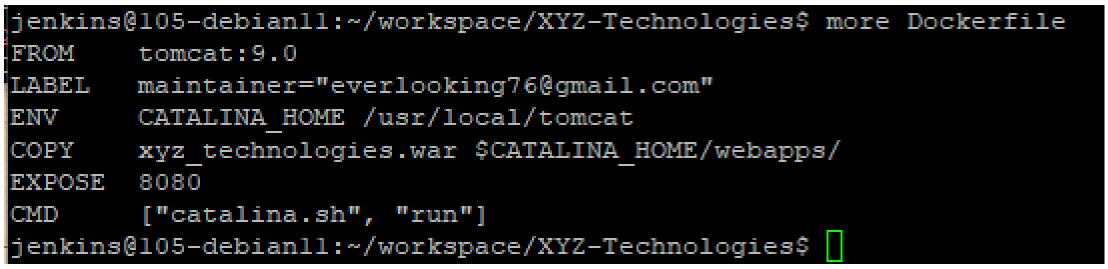
- The final artifact (xyz_technologies.war) has been created!



```
jenkins@105-debian11:~/workspace/XYZ-Technologies$ ls -ltr
total 6992
drwxr-xr-x  4 jenkins jenkins    4096 Jun 10 18:06 src
-rw-r--r--  1 jenkins jenkins      24 Jun 10 18:06 README.md
-rw-r--r--  1 jenkins jenkins     794 Jun 10 18:06 pom.xml.bak
-rw-r--r--  1 jenkins jenkins    2082 Jun 10 18:06 pom.xml
-rw-r--r--  1 jenkins jenkins     201 Jun 10 18:06 Dockerfile
drwxr-xr-x 11 jenkins jenkins    4096 Jun 10 18:06 target
-rw-r--r--  1 jenkins jenkins 7132897 Jun 10 18:06 xyz_technologies.war
jenkins@105-debian11:~/workspace/XYZ-Technologies$
```

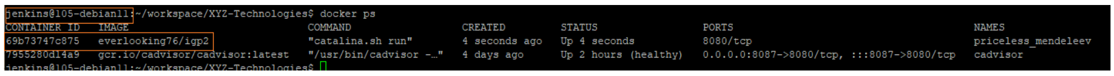
3. Writing Dockerfile to push the war file to tomcat server

- A dockerfile has been written (file is available in project files)



```
jenkins@105-debian11:~/workspace/XYZ-Technologies$ more Dockerfile
FROM     tomcat:9.0
LABEL    maintainer="everlooking76@gmail.com"
ENV      CATALINA_HOME /usr/local/tomcat
COPY     xyz_technologies.war $CATALINA_HOME/webapps/
EXPOSE   8080
CMD      ["catalina.sh", "run"]
jenkins@105-debian11:~/workspace/XYZ-Technologies$
```

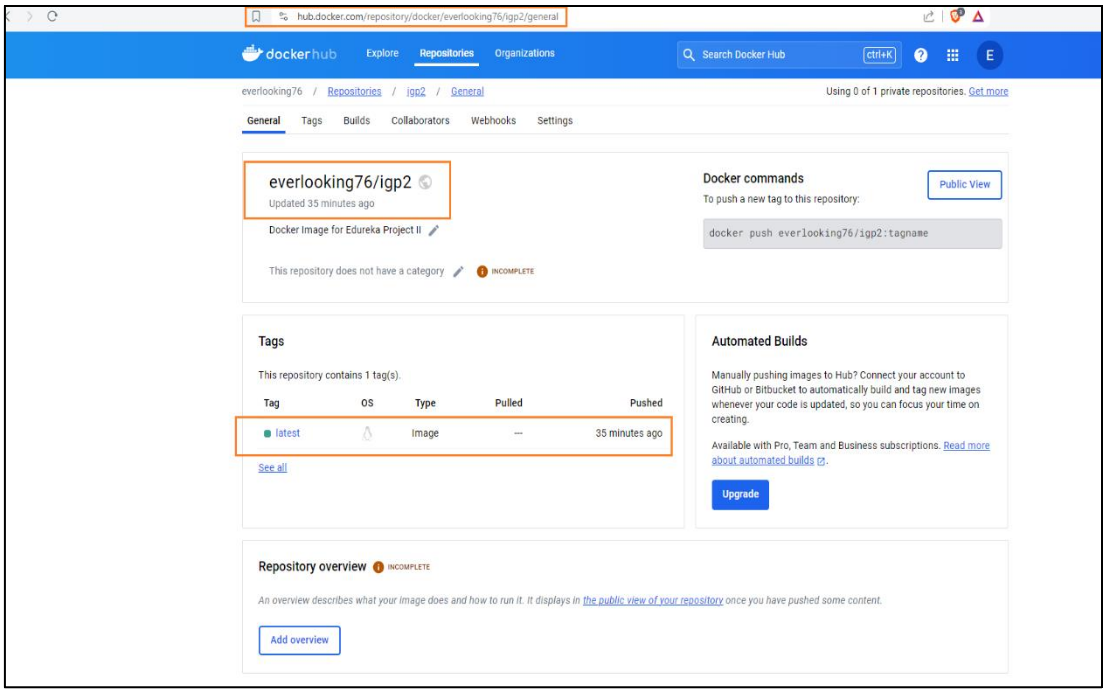
- Using our package "xyz_technologies.war" and dockerfile a docker image has been created.



```
jenkins@105-debian11:~/workspace/XYZ-Technologies$ docker ps
CONTAINER ID   IMAGE                              COMMAND                  CREATED         STATUS                 PORTS                                       NAMES
69b73747c875   everlooking76/igp2                 "catalina.sh run"        4 seconds ago   Up 4 seconds           8080/tcp                                    priceless_mendeleev
7955280d14a9   gcr.io/cadvisor/cadvisor:latest    "/usr/bin/cadvisor -…"   4 days ago      Up 2 hours (healthy)   0.0.0.0:8087->8080/tcp, :::8087->8080/tcp   cadvisor
jenkins@105-debian11:~/workspace/XYZ-Technologies$
```

- Created image to be tagged and pushed to docker-Hub within Jenkins pipeline

- On docker-Hub website:



hub.docker.com/repository/docker/everlooking76/igp2/general

everlooking76/igp2
Updated 35 minutes ago
Docker Image for Edureka Project II
This repository does not have a category INCOMPLETE

Docker commands
To push a new tag to this repository:
`docker push everlooking76/igp2:tagname`

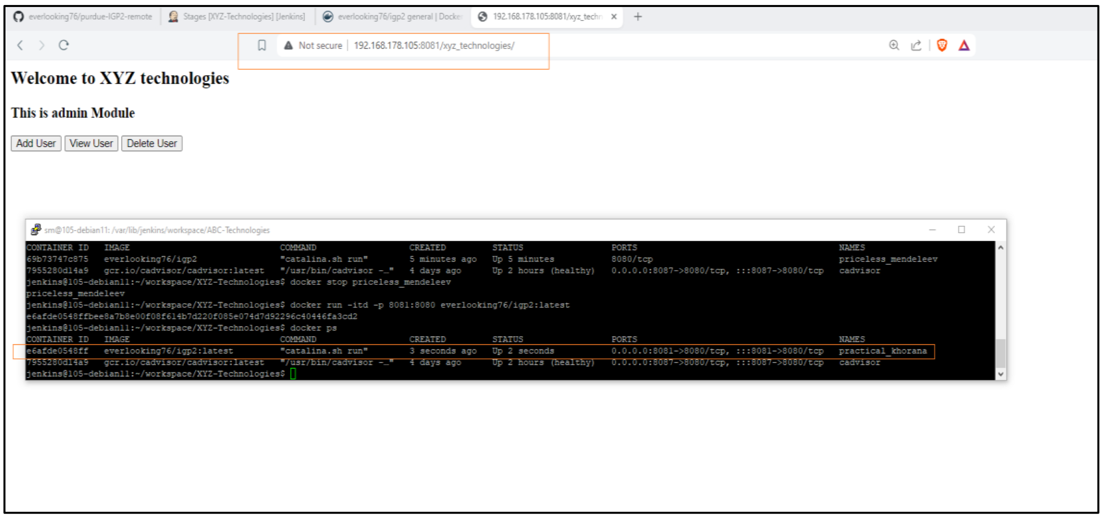
Tags
This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	---	35 minutes ago

Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions. Read more about automated builds.

Repository overview INCOMPLETE
An overview describes what your image does and how to run it. It displays in the public view of your repository once you have pushed some content.

- A Docker Image has been created! (everlooking76/igp2), when running this image, a container will be started and the service will be active and accessible on (http://192.168.178.105:8081/xyz_technologies)


everlooking76/purdue-IGP2-remote
Stages [XYZ-Technologies] [Jenkins]
everlooking76/igp2 general | Docker
192.168.178.105:8081/xyz_techn
Not secure
192.168.178.105:8081/xyz_technologies/
Add User
View User
Delete User
sm@105-debian11: /var/lib/jenkins/workspace/ABC-Technologies

Welcome to XYZ technologies

This is admin Module

```
CONTAINER ID   IMAGE                            COMMAND                  CREATED          STATUS                  PORTS                                       NAMES
69b73747c875   everlooking76/igp2               "catalina.sh run"        5 minutes ago    Up 5 minutes            8080/tcp                                    priceless_mendeleev
7955280d14a9   gcr.io/cadvisor/cadvisor:latest  "/usr/bin/cadvisor -…"   4 days ago       Up 2 hours (healthy)    0.0.0.0:8087->8080/tcp, :::8087->8080/tcp   cadvisor
jenkins@105-debian11:~/workspace/XYZ-Technologies$ docker stop priceless_mendeleev
priceless_mendeleev
jenkins@105-debian11:~/workspace/XYZ-Technologies$ docker run -itd -p 8081:8080 everlooking76/igp2:latest
e6afde0548ffbee8a7b8e00f08f614b7d220f085e074d7d92296c40446fa3cd2
jenkins@105-debian11:~/workspace/XYZ-Technologies$ docker ps
CONTAINER ID   IMAGE                            COMMAND                  CREATED          STATUS                  PORTS                                       NAMES
e6afde0548ff   everlooking76/igp2:latest        "catalina.sh run"        3 seconds ago    Up 2 seconds            0.0.0.0:8081->8080/tcp, :::8081->8080/tcp   practical_khorana
7955280d14a9   gcr.io/cadvisor/cadvisor:latest  "/usr/bin/cadvisor -…"   4 days ago       Up 2 hours (healthy)    0.0.0.0:8087->8080/tcp, :::8087->8080/tcp   cadvisor
jenkins@105-debian11:~/workspace/XYZ-Technologies$ 
```

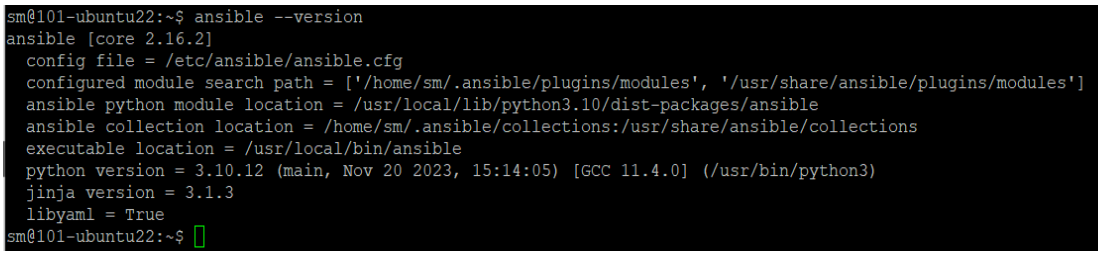
4. Integrating docker with Ansible using Jenkins pipeline

The pipeline used in step number 2 will be used and be modified to integrate Ansible(192.168.178.101) to deploy the artifact on dedicated Docker Server(192.168.178.110).

- Installing Ansible

The Ansible Server has been previous installed and configured on (192.168.178.101), here are the steps used to install Ansible:

```
# Updating installed packages
sudo apt update

# Upgrading packages if needed
sudo apt upgrade -y


# The package Repositories are added and updated
sudo apt install ansible


# After installation the ansible version can be checked
sudo ansible --version
```



```
sm@101-ubuntu22:~$ ansible --version
ansible [core 2.16.2]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/sm/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.10/dist-packages/ansible
  ansible collection location = /home/sm/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.1.3
  libyaml = True
sm@101-ubuntu22:~$ 
```

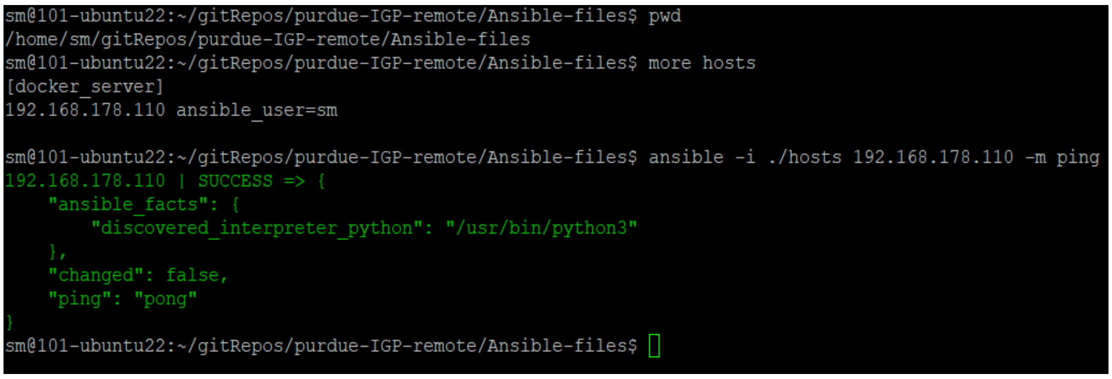
```
# The inventory file resides on /etc/ansible/hosts
# It contains the information of managed
nodes/servers/containers/entities
# This is the default location for this file
# A customized file can be used by using the switch "-i" when giving
# Ansible commands
# To Configure the /etc/ansible/hosts
sudo vi /etc/ansible/hosts
```

```
# For the sake of our IGP Project a custom inventory file will be
used in
# /home/sm/gitRepos/purdue-IGP2-remote/Ansible-files
# These files will be pushed to remote git-Hub for this project
```

- SSH Key-Pairs
  Ansible is an "Agentless" Configuration Management System, it uses SSH securely to communicate and execute management tasks on "slave" nodes defined in inventory file.
  Therefore, after installing Ansible, SSH Key-Pairs should be generated and Public Key should be copied to all slave nodes. The generation of SSH Key-Pairs is like 1st Step.

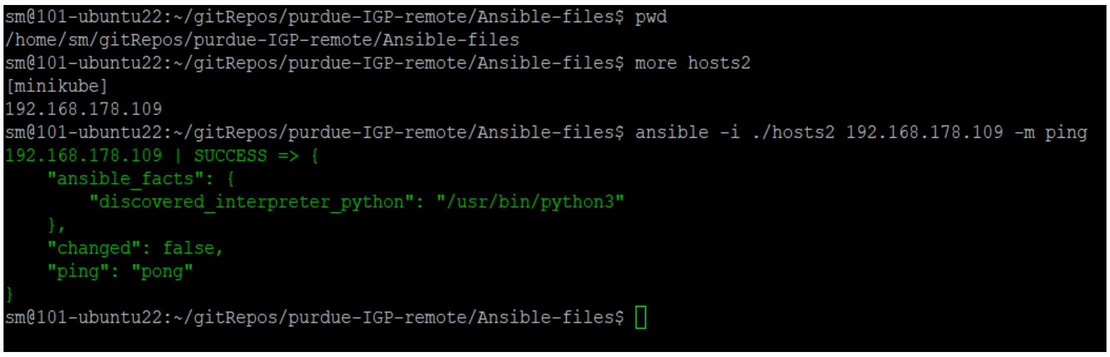
```
# Generating new SSH Key-Pair using ed_25519 encryption
ssh-keygen -t ed_25519 -C "Ansible – master"

# Public key should be copied to slave nodes
ssh-copy-id sm@<server_IP>
```

In our project, the master node is on (192.168.178.101), and slave nodes (192.168.178.109) and (192.168.178.110) so testing SSH access yields:



```
sm@101-ubuntu22:~/gitRepos/purdue-IGP-remote/Ansible-files$ pwd
/home/sm/gitRepos/purdue-IGP-remote/Ansible-files
sm@101-ubuntu22:~/gitRepos/purdue-IGP-remote/Ansible-files$ more hosts
[docker_server]
192.168.178.110 ansible_user=sm

sm@101-ubuntu22:~/gitRepos/purdue-IGP-remote/Ansible-files$ ansible -i ./hosts 192.168.178.110 -m ping
192.168.178.110 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
sm@101-ubuntu22:~/gitRepos/purdue-IGP-remote/Ansible-files$
```

Similarly on node (192.168.178.109):

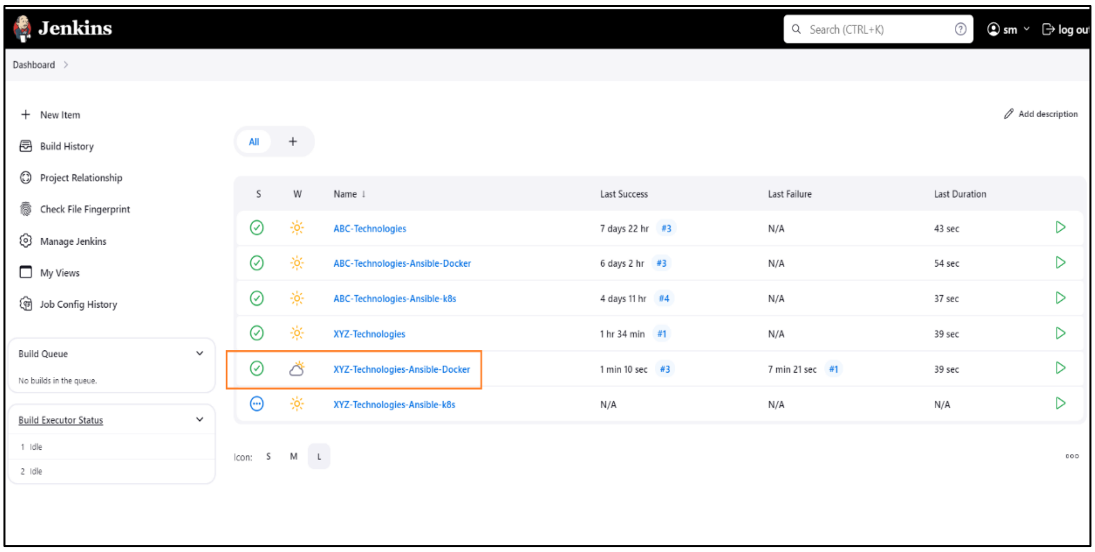


```
sm@101-ubuntu22:~/gitRepos/purdue-IGP-remote/Ansible-files$ pwd
/home/sm/gitRepos/purdue-IGP-remote/Ansible-files
sm@101-ubuntu22:~/gitRepos/purdue-IGP-remote/Ansible-files$ more hosts2
[minikube]
192.168.178.109
sm@101-ubuntu22:~/gitRepos/purdue-IGP-remote/Ansible-files$ ansible -i ./hosts2 192.168.178.109 -m ping
192.168.178.109 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
sm@101-ubuntu22:~/gitRepos/purdue-IGP-remote/Ansible-files$
```

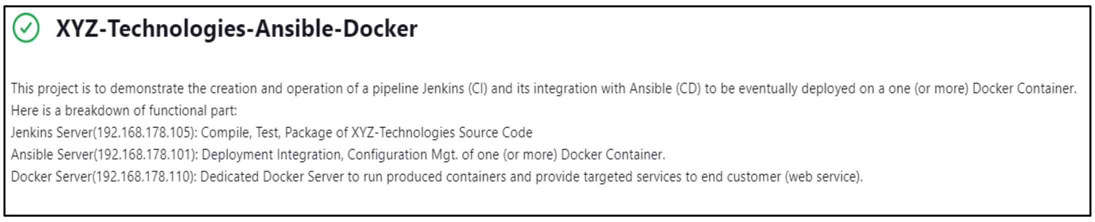
- Creating the 2nd pipeline: XYZ-Technologies_Ansible_Docker

  In this pipeline the creation of docker image using dockerfile, tagging the new image, and pushing it to docker-Hub are included within the pipeline staging, therefore the "Jinkins" user had to be added to Docker group in order to have the rights to process "Dockerfile", the command to do that (on Jenkins Server):

```
usermod -aG docker jenkins
```



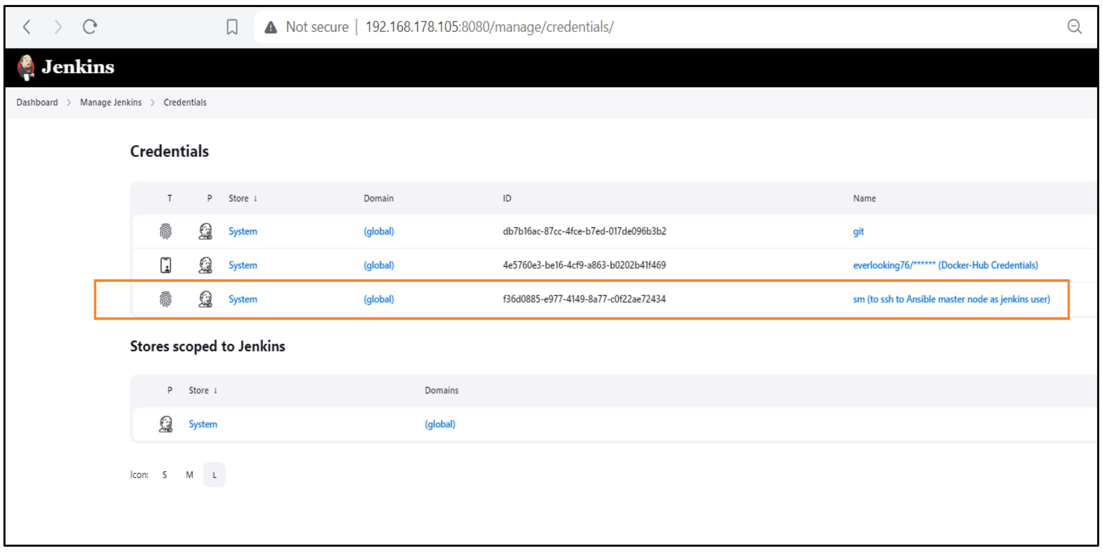
S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	ABC-Technologies	7 days 22 hr #3	N/A	43 sec
✓	☀	ABC-Technologies-Ansible-Docker	6 days 2 hr #3	N/A	54 sec
✓	☀	ABC-Technologies-Ansible-k8s	4 days 11 hr #4	N/A	37 sec
✓	☀	XYZ-Technologies	1 hr 34 min #1	N/A	39 sec
✓	⛅	XYZ-Technologies-Ansible-Docker	1 min 10 sec #3	7 min 21 sec #1	39 sec
⋯	☀	XYZ-Technologies-Ansible-k8s	N/A	N/A	N/A



XYZ-Technologies-Ansible-Docker

This project is to demonstrate the creation and operation of a pipeline Jenkins (CI) and its integration with Ansible (CD) to be eventually deployed on a one (or more) Docker Container.
Here is a breakdown of functional part:
Jenkins Server(192.168.178.105): Compile, Test, Package of XYZ-Technologies Source Code
Ansible Server(192.168.178.101): Deployment Integration, Configuration Mgt. of one (or more) Docker Container.
Docker Server(192.168.178.110): Dedicated Docker Server to run produced containers and provide targeted services to end customer (web service).

- For Jenkins to access Ansible Server (master) an SSH Key-Pairs have been prepared and credentials to access Ansible Server have been added.


Not secure | 192.168.178.105:8080/manage/credentials/
Jenkins
Dashboard > Manage Jenkins > Credentials
Icon: S M L

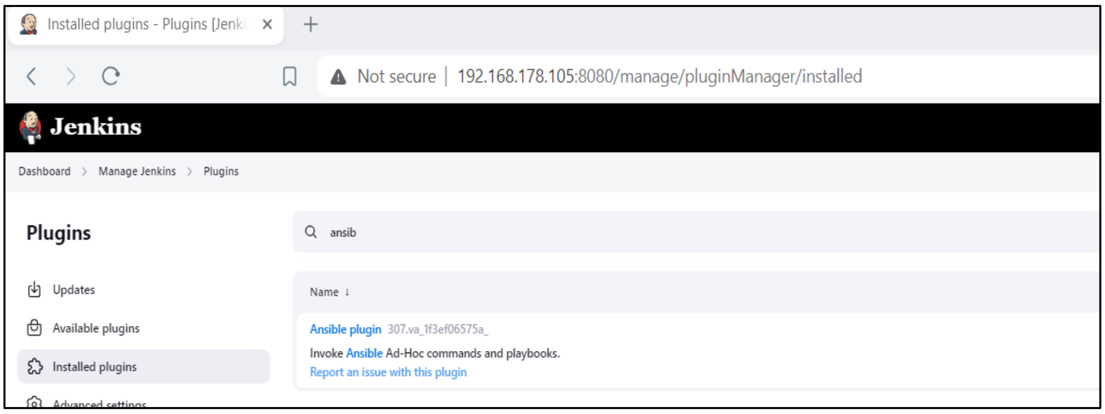
Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	db7b16ac-87cc-4fce-b7ed-017de096b3b2	git
		System	(global)	4e5760e3-be16-4cf9-a863-b0202b41f469	everlooking76/****** (Docker-Hub Credentials)
		System	(global)	f36d0885-e977-4149-8a77-c0f22ae72434	sm (to ssh to Ansible master node as jenkins user)

Stores scoped to Jenkins

P	Store ↓	Domains
	System	(global)

- Ansible plug-in is installed


Installed plugins - Plugins [Jenki

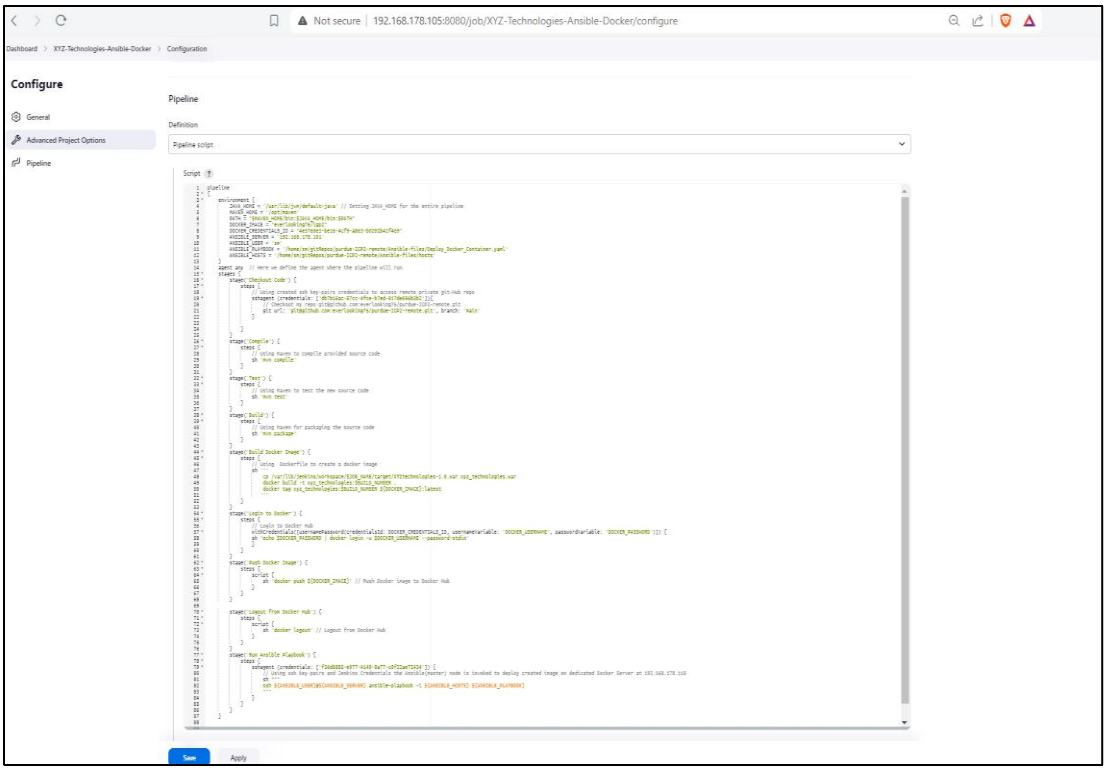
Not secure | 192.168.178.105:8080/manage/pluginManager/installed

Ansible plugin 307.va_1f3ef06575a_

Invoke Ansible Ad-Hoc commands and playbooks.

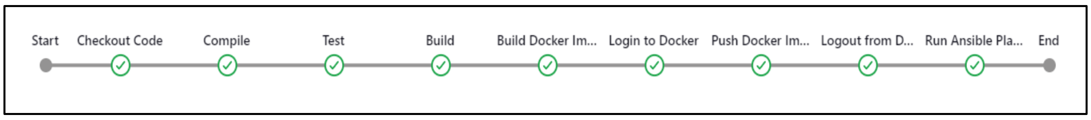
Report an issue with this plugin

- Pipeline Code (Groovy Code): The code can be found in "jenkinsfile-XYZ_Technologies_Ansible_Docker.txt" within project files. Here is a snapshot of the same pipeline code:

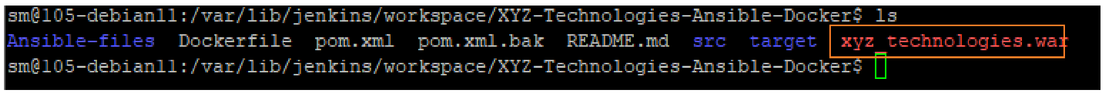

Not secure | 192.168.178.105:8080/job/XYZ-Technologies-Ansible-Docker/configure

```
pipeline
{
    environment {
        JAVA_HOME = '/usr/lib/jvm/default-java' // Setting JAVA_HOME for the entire pipeline
        MAVEN_HOME = '/opt/maven'
        PATH = "$MAVEN_HOME/bin:$JAVA_HOME/bin:$PATH"
        DOCKER_IMAGE = "everlooking76/igp2"
        DOCKER_CREDENTIALS_ID = "4e5760e3-be16-4cf9-a863-b0202b41f469"
        ANSIBLE_SERVER = '192.168.178.101'
        ANSIBLE_USER = 'sm'
        ANSIBLE_PLAYBOOK = '/home/sm/gitRepos/purdue-ICP2-remote/Ansible-files/Deploy_Docker_Container.yaml'
        ANSIBLE_HOSTS = '/home/sm/gitRepos/purdue-ICP2-remote/Ansible-files/hosts'
    }
    agent any  // Here we define the agent where the pipeline will run
    stages {
        stage('Checkout Code') {
            steps {
                // Using created ssh key-pairs credentials to access remote private git-Hub repo
                sshagent (credentials: ['db7b16ac-87cc-4fce-b7ed-817de896b3b2']){
                    // Checkout my repo git@github.com:everlooking76/purdue-ICP2-remote.git
                    git url: 'git@github.com:everlooking76/purdue-ICP2-remote.git', branch: 'main'
                }

            }
        }
        stage('Compile') {
            steps {
                // Using Maven to compile provided source code
                sh 'mvn compile'
            }
        }
        stage('Test') {
            steps {
                // Using Maven to test the new source code
                sh 'mvn test'
            }
        }
        stage('Build') {
            steps {
                // Using Maven for packaging the source code
                sh 'mvn package'
            }
        }
        stage('Build Docker Image') {
            steps {
                // Using  Dockerfile to create a docker image
                sh """
                    cp /var/lib/jenkins/workspace/$JOB_NAME/target/XYZtechnologies-1.0.war xyz_technologies.war
                    docker build -t xyz_technologies:$BUILD_NUMBER .
                    docker tag xyz_technologies:$BUILD_NUMBER ${DOCKER_IMAGE}:latest
                    """
            }
        }
        stage('Login to Docker') {
            steps {
                // Login to Docker Hub
                withCredentials([usernamePassword(credentialsId: DOCKER_CREDENTIALS_ID, usernameVariable: 'DOCKER_USERNAME', passwordVariable: 'DOCKER_PASSWORD')]) {
                sh 'echo $DOCKER_PASSWORD | docker login -u $DOCKER_USERNAME --password-stdin'
                }
            }
        }
        stage('Push Docker Image') {
            steps {
                script {
                    sh 'docker push ${DOCKER_IMAGE}' // Push Docker image to Docker Hub
                }
            }
        }

        stage('Logout from Docker Hub') {
            steps {
                script {
                    sh 'docker logout' // Logout from Docker Hub
                }
            }
        }
        stage('Run Ansible Playbook') {
            steps {
                sshagent (credentials: ['f36d0885-e977-4149-8a77-c8f22ae72434']) {
                    // Using ssh key-pairs and Jenkins Credentials the Ansible(master) node is invoked to deploy created image on dedicated Docker Server at 192.168.178.110
                    sh """
                    ssh ${ANSIBLE_USER}@${ANSIBLE_SERVER} ansible-playbook -i ${ANSIBLE_HOSTS} ${ANSIBLE_PLAYBOOK}
                    """
                }
            }
        }
    }
}
```

- The build is successful!



The created package(artifact) resides on:

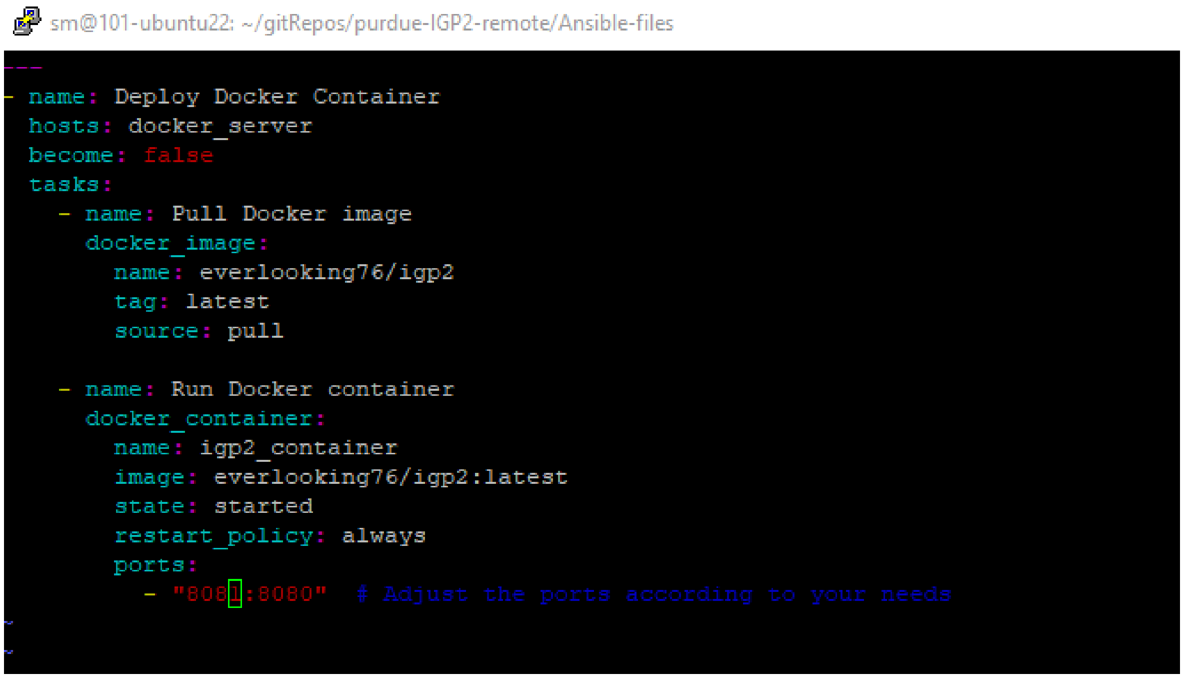


```
sm@105-debian11:/var/lib/jenkins/workspace/XYZ-Technologies-Ansible-Docker$ ls
Ansible-files  Dockerfile  pom.xml  pom.xml.bak  README.md  src  target  xyz_technologies.war
sm@105-debian11:/var/lib/jenkins/workspace/XYZ-Technologies-Ansible-Docker$
```

According to this pipeline, an Ansible playbook has been executed to do:

- Establish secure session with Docker Server using SSH
- Pull Created Docker Image from Docker-Hub (everlooking76/igp2) tagged as "latest"
- Run this Image to start tomcat container
- The Container will provide its Service using Port 8080 (intern) and can be reached externally over Port 8081

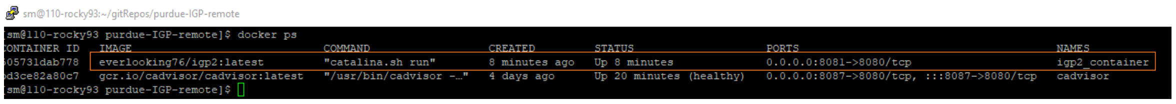
Here is the used playbook code: (Deploy_Docker_Container.yaml)



sm@101-ubuntu22: ~/gitRepos/purdue-IGP2-remote/Ansible-files

```
---
- name: Deploy Docker Container
  hosts: docker_server
  become: false
  tasks:
    - name: Pull Docker image
      docker_image:
        name: everlooking76/igp2
        tag: latest
        source: pull

    - name: Run Docker container
      docker_container:
        name: igp2_container
        image: everlooking76/igp2:latest
        state: started
        restart_policy: always
        ports:
          - "8081:8080"  # Adjust the ports according to your needs
~
~
```

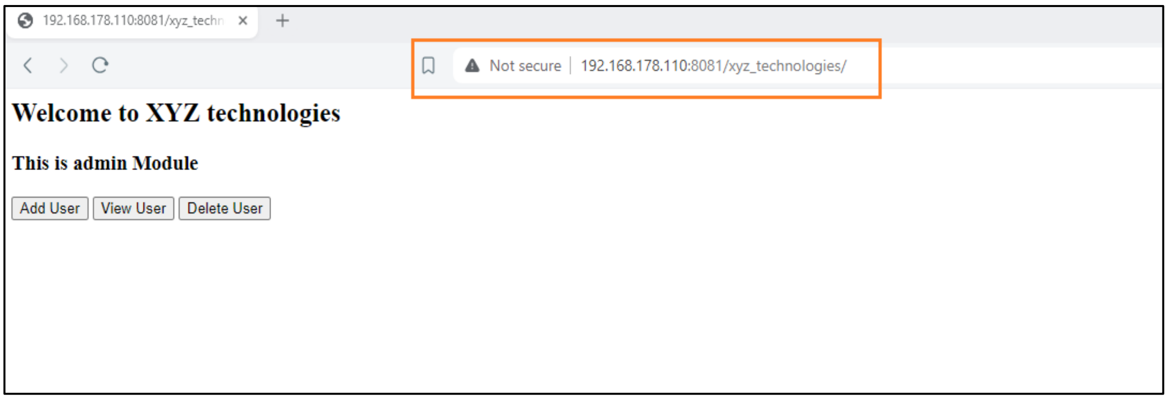
In Docker Server(192.168.178.110):



sm@110-rocky93:~/gitRepos/purdue-IGP-remote

```
[sm@110-rocky93 purdue-IGP-remote]$ docker ps
ONTAINER ID   IMAGE                             COMMAND                  CREATED          STATUS                     PORTS                                         NAMES
05731dab778   everlooking76/igp2:latest         "catalina.sh run"        8 minutes ago    Up 8 minutes               0.0.0.0:8081->8080/tcp                        igp2_container
d3ce82a80c7   gcr.io/cadvisor/cadvisor:latest   "/usr/bin/cadvisor -…"   4 days ago       Up 20 minutes (healthy)    0.0.0.0:8087->8080/tcp, :::8087->8080/tcp     cadvisor
[sm@110-rocky93 purdue-IGP-remote]$
```

On the browser:



192.168.178.110:8081/xyz_techn

Not secure | 192.168.178.110:8081/xyz_technologies/

Welcome to XYZ technologies

This is admin Module

5. Deploying artifacts to Kubernetes Cluster

Using Ansible Server as master configuration Server makes it possible for a setup to be scalable and easy to manage, because the setup can be replicated to “n” number of K8s Clusters providing a high level of automation, saving time and cost, and eliminating human error.

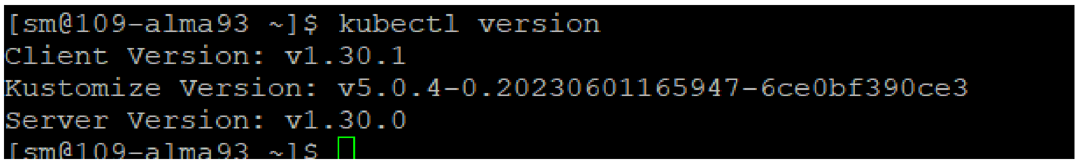
In our Demonstration, “Minikube” Cluster has been installed on an “Alma-Linux Server”(192.168.178.109), as well as “kubectl” and Docker engine.

A 3rd pipeline has been added to Jenkins Server, where it will process the previous steps then it will invoke a playbook yaml file, which will establish SSH session on “Minikube” Cluster and invoke a “Manifest” yaml file to deploy intended solution and create service pods which will manage and run our docker container.

- Setting up Minikube-Cluster

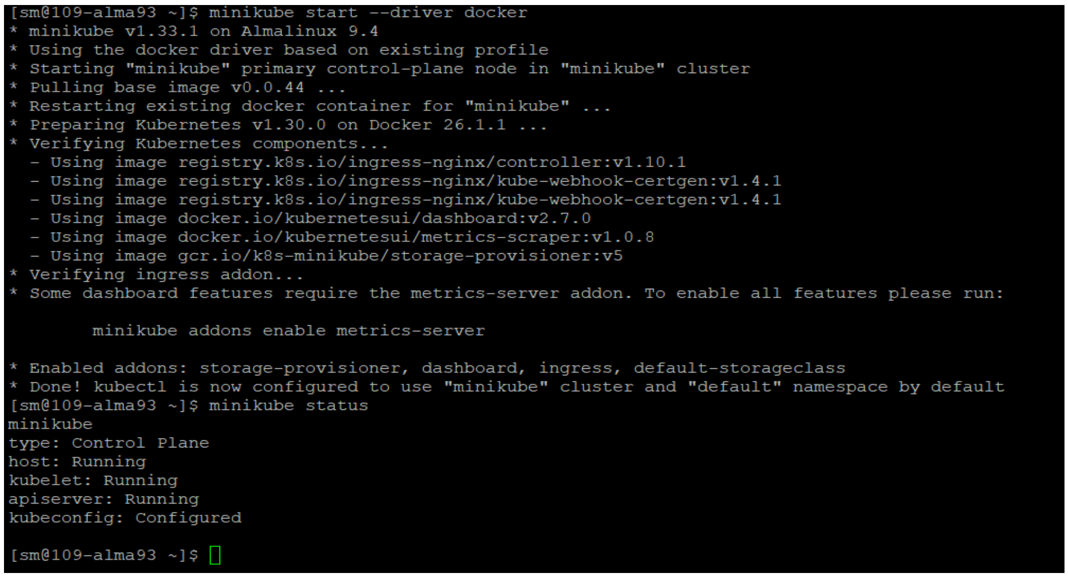
  After updating necessary Repositories on (192.168.178.109):

```
# Installing “kubectl” binary
 curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
sudo cp kubectl /usr/local/bin/ && sudo chmod +x
/usr/local/bin/kubectl
# Checking kubectl
```



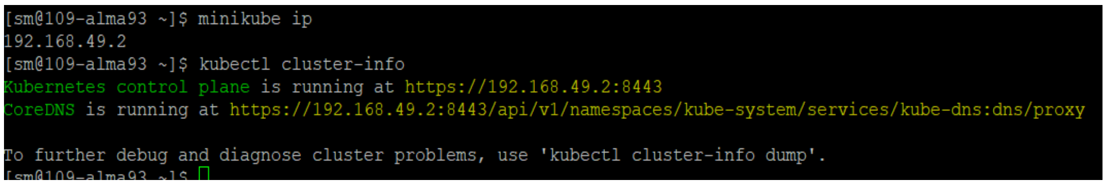
```
[sm@109-alma93 ~]$ kubectl version
Client Version: v1.30.1
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
Server Version: v1.30.0
[sm@109-alma93 ~]$
```

```
# Installing Minikube
curl -LO
https://storage.googleapis.com/minikube/releases/latest/minikube-
linux-amd64
sudo install minikube-linux-amd64 /usr/local/bin/minikube

# Starting and checking Minikube Cluster
```



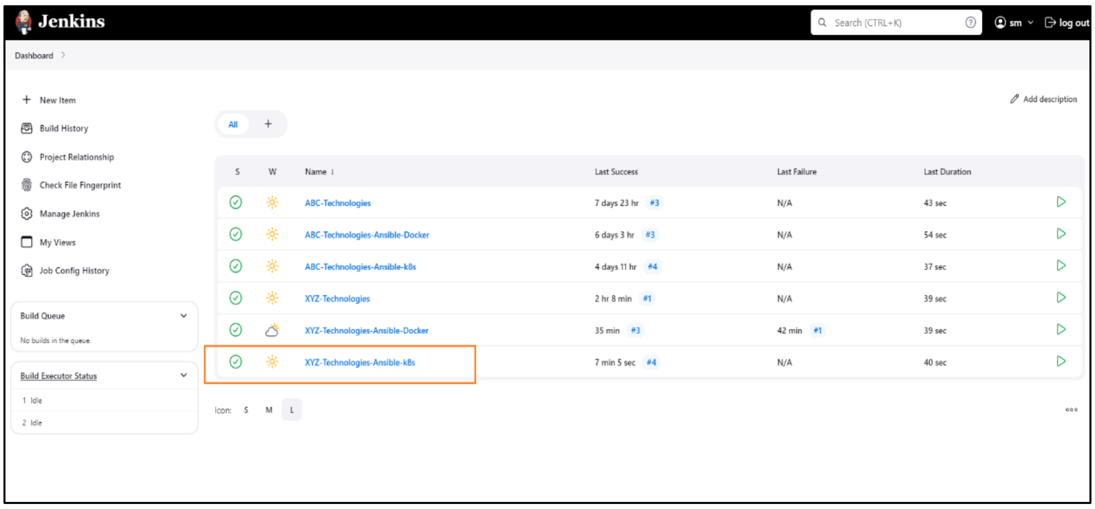
```
[sm@109-alma93 ~]$ minikube start --driver docker
* minikube v1.33.1 on Almalinux 9.4
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.44 ...
* Restarting existing docker container for "minikube" ...
* Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
* Verifying Kubernetes components...
  - Using image registry.k8s.io/ingress-nginx/controller:v1.10.1
  - Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
  - Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
  - Using image docker.io/kubernetesui/dashboard:v2.7.0
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Verifying ingress addon...
* Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server

* Enabled addons: storage-provisioner, dashboard, ingress, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
[sm@109-alma93 ~]$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

[sm@109-alma93 ~]$
```

Checking on Cluster Info

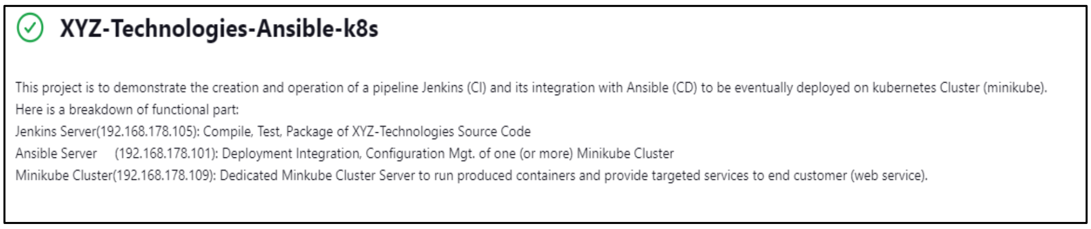


```
[sm@109-alma93 ~]$ minikube ip
192.168.49.2
[sm@109-alma93 ~]$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
[sm@109-alma93 ~]$
```

- Creating 3rd jenkins pipeline



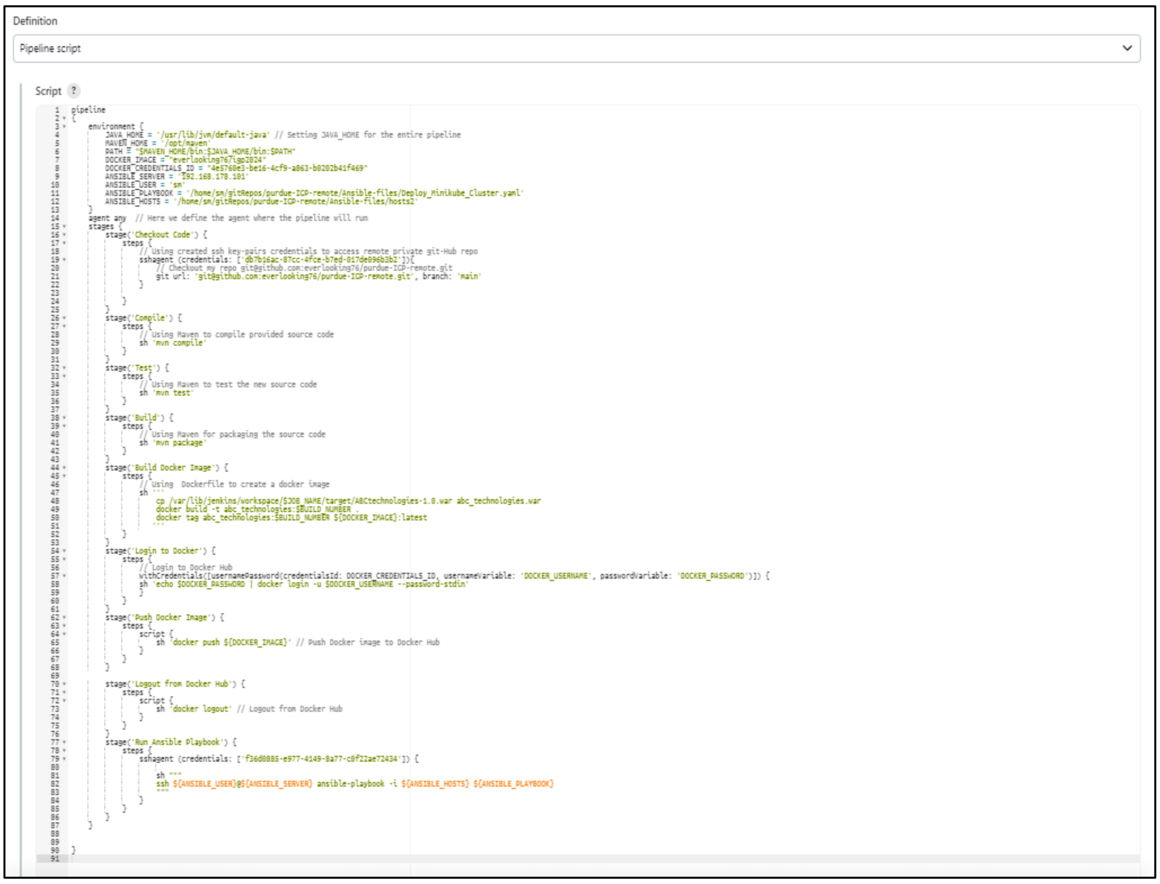
S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	ABC-Technologies	7 days 23 hr #3	N/A	43 sec
✓	☀	ABC-Technologies-Ansible-Docker	6 days 3 hr #3	N/A	54 sec
✓	☀	ABC-Technologies-Ansible-k8s	4 days 11 hr #4	N/A	37 sec
✓	☀	XYZ-Technologies	2 hr 8 min #1	N/A	39 sec
✓	⛅	XYZ-Technologies-Ansible-Docker	35 min #3	42 min #1	39 sec
✓	☀	XYZ-Technologies-Ansible-k8s	7 min 5 sec #4	N/A	40 sec



XYZ-Technologies-Ansible-k8s

This project is to demonstrate the creation and operation of a pipeline Jenkins (CI) and its integration with Ansible (CD) to be eventually deployed on kubernetes Cluster (minikube).
Here is a breakdown of functional part:
Jenkins Server(192.168.178.105): Compile, Test, Package of XYZ-Technologies Source Code
Ansible Server (192.168.178.101): Deployment Integration, Configuration Mgt. of one (or more) Minikube Cluster
Minikube Cluster(192.168.178.109): Dedicated Minkube Cluster Server to run produced containers and provide targeted services to end customer (web service).

- Pipeline (groovy) Code: (jenkinsfile -ABC_Technologies_Ansible_k8s.txt)



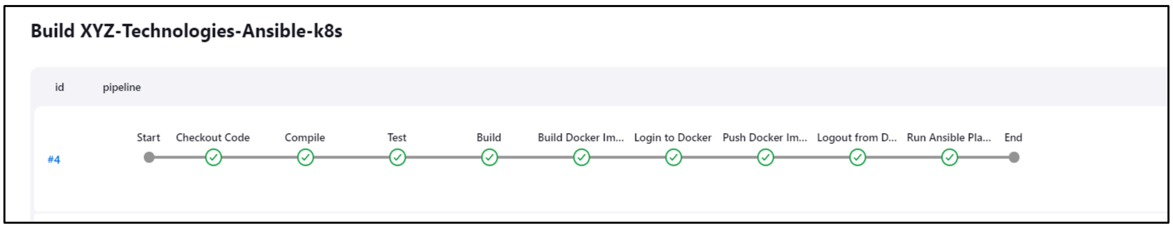
Definition

Pipeline script

Script

```groovy
pipeline
{
    environment {
        JAVA_HOME = '/usr/lib/jvm/default-java' // Setting JAVA_HOME for the entire pipeline
        MAVEN_HOME = '/opt/maven'
        PATH = "$MAVEN_HOME/bin:$JAVA_HOME/bin:$PATH"
        DOCKER_IMAGE = "everlooking76/igp2024"
        DOCKER_CREDENTIALS_ID = "4e5760e3-be16-4cf9-a863-b0202b41f469"
        ANSIBLE_SERVER = '192.168.178.101'
        ANSIBLE_USER = 'sm'
        ANSIBLE_PLAYBOOK = '/home/sm/gitRepos/purdue-IGP-remote/Ansible-files/Deploy_Minikube_Cluster.yaml'
        ANSIBLE_HOSTS = '/home/sm/gitRepos/purdue-IGP-remote/Ansible-files/hosts2'
    }
    agent any  // Here we define the agent where the pipeline will run
    stages {
        stage('Checkout Code') {
            steps {
                // Using created ssh key-pairs credentials to access remote private git-Hub repo
                sshagent (credentials: ['db7b16ac-87cc-4fce-b7ed-017de096b3b2']){
                    // Checkout my repo git@github.com:everlooking76/purdue-IGP-remote.git
                    git url: 'git@github.com:everlooking76/purdue-IGP-remote.git', branch: 'main'
                }

            }
        }
        stage('Compile') {
            steps {
                // Using Maven to compile provided source code
                sh 'mvn compile'
            }
        }
        stage('Test') {
            steps {
                // Using Maven to test the new source code
                sh 'mvn test'
            }
        }
        stage('Build') {
            steps {
                // Using Maven for packaging the source code
                sh 'mvn package'
            }
        }
        stage('Build Docker Image') {
            steps {
                // Using  Dockerfile to create a docker image
                sh '''
                    cp /var/lib/jenkins/workspace/$JOB_NAME/target/ABCtechnologies-1.0.war abc_technologies.war
                    docker build -t abc_technologies:$BUILD_NUMBER .
                    docker tag abc_technologies:$BUILD_NUMBER ${DOCKER_IMAGE}:latest
                   '''
            }
        }
        stage('Login to Docker') {
            steps {
                // Login to Docker Hub
                withCredentials([usernamePassword(credentialsId: DOCKER_CREDENTIALS_ID, usernameVariable: 'DOCKER_USERNAME', passwordVariable: 'DOCKER_PASSWORD')]) {
                sh 'echo $DOCKER_PASSWORD | docker login -u $DOCKER_USERNAME --password-stdin'
                }
            }
        }
        stage('Push Docker Image') {
            steps {
                script {
                    sh 'docker push ${DOCKER_IMAGE}' // Push Docker image to Docker Hub
                }
            }
        }

        stage('Logout from Docker Hub') {
            steps {
                script {
                    sh 'docker logout' // Logout from Docker Hub
                }
            }
        }
        stage('Run Ansible Playbook') {
            steps {
                sshagent (credentials: ['f36d8885-e977-4149-8a77-c0f22ae72434']) {

                    sh """
                    ssh ${ANSIBLE_USER}@${ANSIBLE_SERVER} ansible-playbook -i ${ANSIBLE_HOSTS} ${ANSIBLE_PLAYBOOK}
                    """
                }
            }
        }
    }


}
```

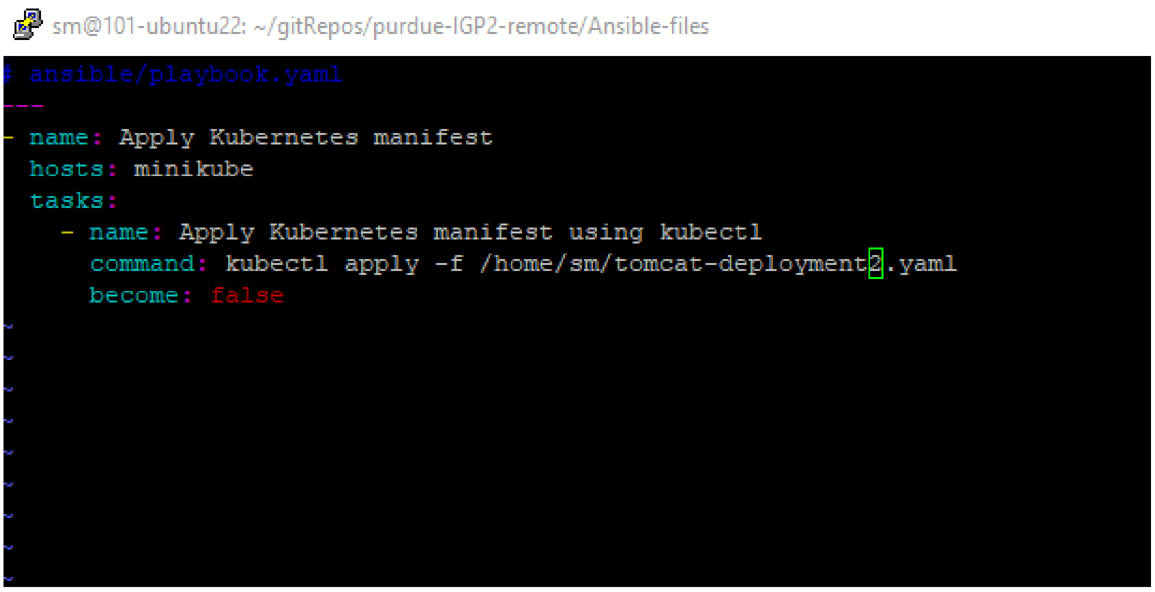
- Build is successful!



Build XYZ-Technologies-Ansible-k8s

id	pipeline											
	Start	Checkout Code	Compile	Test	Build	Build Docker Im...	Login to Docker	Push Docker Im...	Logout from D...	Run Ansible Pla...	End
#4	●	✓	✓	✓	✓	✓	✓	✓	✓	✓	●

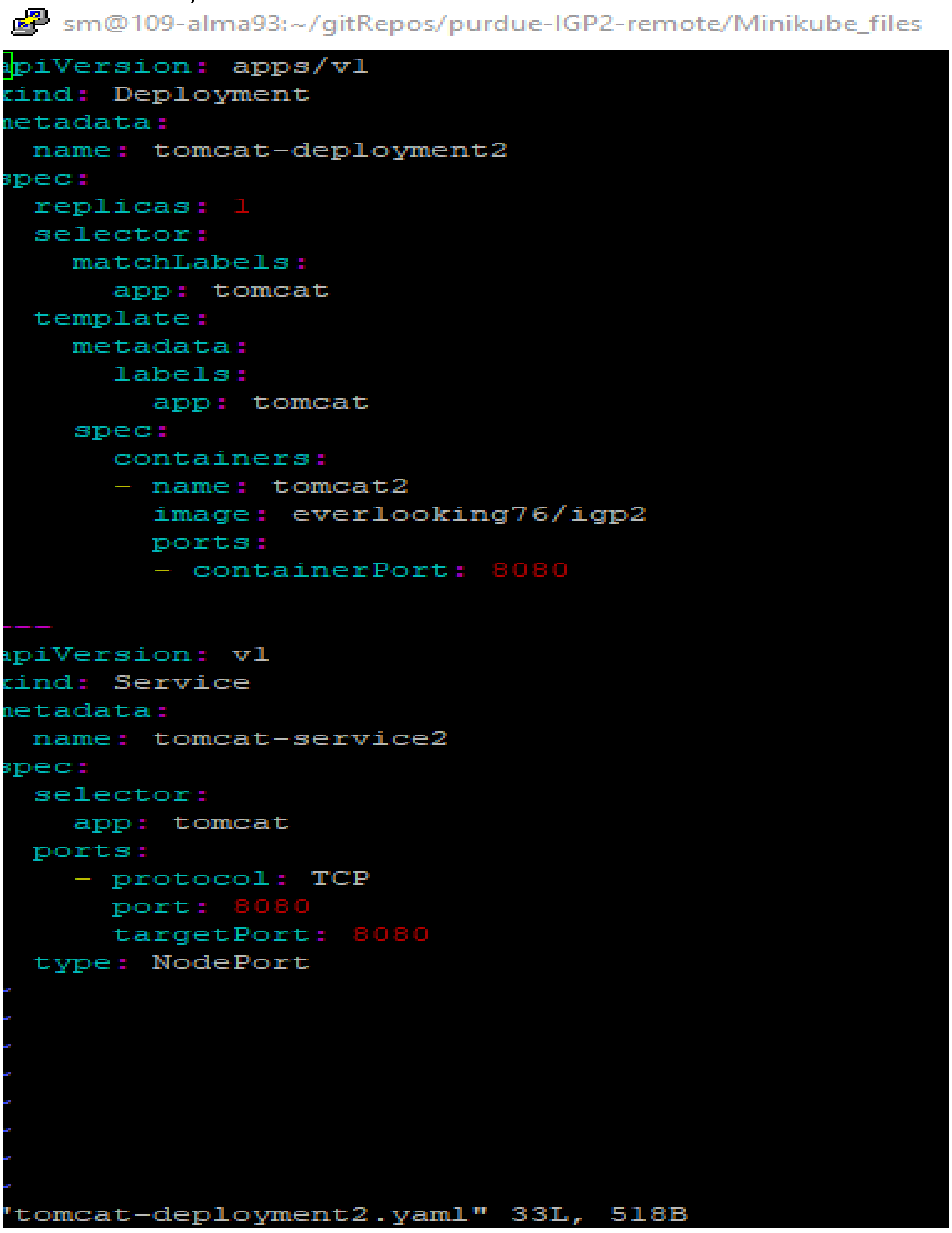
- Ansible yaml file: (Deploy_Minikube_Cluster.yaml)



sm@101-ubuntu22: ~/gitRepos/purdue-IGP2-remote/Ansible-files

```
# ansible/playbook.yaml
---
- name: Apply Kubernetes manifest
  hosts: minikube
  tasks:
    - name: Apply Kubernetes manifest using kubectl
      command: kubectl apply -f /home/sm/tomcat-deployment2.yaml
      become: false
~
~
~
~
~
~
~
~
~
```

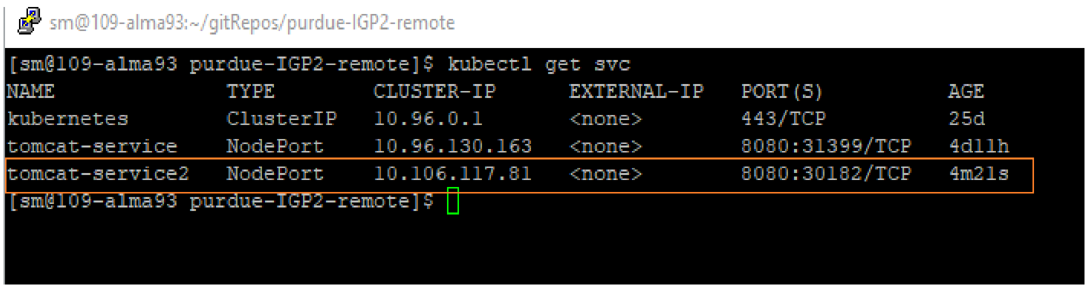
- Minikube Manifest yaml file



sm@109-alma93:~/gitRepos/purdue-IGP2-remote/Minikube_files

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: tomcat-deployment2
spec:
  replicas: 1
  selector:
    matchLabels:
      app: tomcat
  template:
    metadata:
      labels:
        app: tomcat
    spec:
      containers:
      - name: tomcat2
        image: everlooking76/igp2
        ports:
        - containerPort: 8080

---
apiVersion: v1
kind: Service
metadata:
  name: tomcat-service2
spec:
  selector:
    app: tomcat
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
  type: NodePort
~
~
~
~
~
~
~
~
"tomcat-deployment2.yaml" 33L, 518B
```

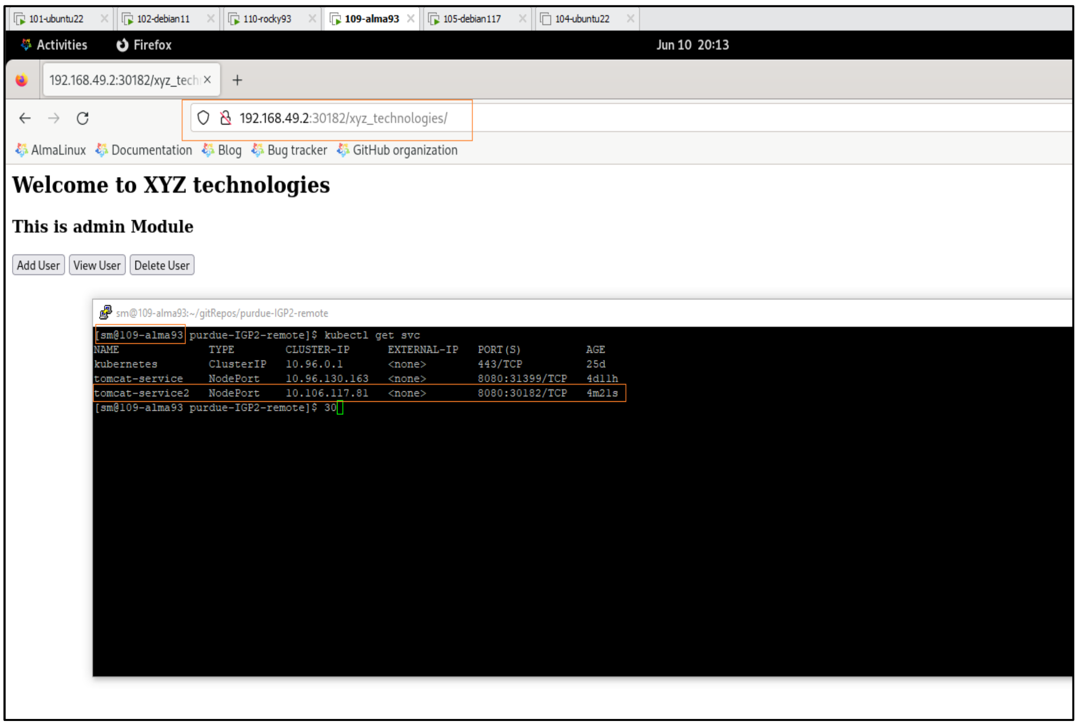
- Deployed artifact on (192.168.178.109 Almalinux Node using Minikube IP)



sm@109-alma93:~/gitRepos/purdue-IGP2-remote

```
[sm@109-alma93 purdue-IGP2-remote]$ kubectl get svc
NAME               TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes         ClusterIP   10.96.0.1        <none>        443/TCP          25d
tomcat-service     NodePort    10.96.130.163    <none>        8080:31399/TCP   4d11h
tomcat-service2    NodePort    10.106.117.81    <none>        8080:30182/TCP   4m21s
[sm@109-alma93 purdue-IGP2-remote]$
```

- The deployed Docker Container on Minikube Cluster can be accessed on this link: `http://192.168.42.2:30182/xyz_technologies`



Welcome to XYZ technologies

This is admin Module

```
[sm@109-alma93 purdue-IGP2-remote]$ kubectl get svc
NAME              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes        ClusterIP   10.96.0.1       <none>        443/TCP          25d
tomcat-service    NodePort    10.96.130.163   <none>        8080:31399/TCP   4d11h
tomcat-service2   NodePort    10.106.117.81   <none>        8080:30182/TCP   4m21s
[sm@109-alma93 purdue-IGP2-remote]$ 30
```

Where: Minikube Cluster IP: 192.168.49.2

NodePort : 30182

If the solution was implemented on Cloud using one of the major Cloud providers (AWS, GCP, Azure) then we would have used "LoadBalancer" in our Configurations instead of NodePort.

6. Monitoring Resources Using Prometheus/Grafana

A dedicated Server on (192.168.178.104) is used as Prometheus (master) server, Grafana will be installed on the same server.

Prometheus is an agent-based monitoring platform, meaning for prometheus master to monitor other nodes on the network an agent on these nodes must be installed “Node Exporter”. As the name implies, this agent will work to scrape the metrics on the node and send it back to master prometheus server.

To monitor docker containers on the network a special docker images must be pulled and run: (cadvisor, redis), but as a pre-requisite docker engine and docker-compose should be installed and configured on prometheus configurations to be commissioned.

The following servers are going to be included and monitored using our prometheus/grafana services:

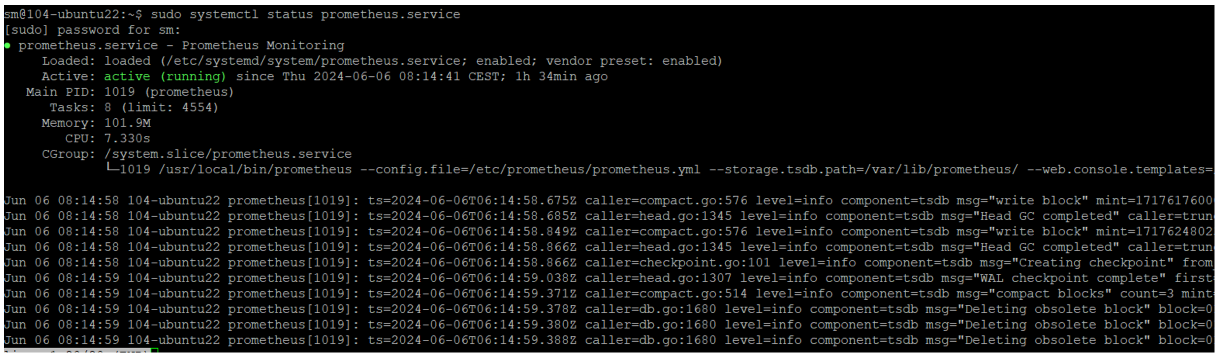
➢ Jenkins Server	(192.168.178.105)	(Node Exporter)
➢ Ansible Server	(192.168.178.101)	(Node Exporter)
➢ Docker Server	(192.168.178.110)	(Node Exporter)
➢ Minikube Cluster	(192.168.178.109)	(Node Exporter)
➢ Local Repo	(192.168.178.102)	(Node Exporter)
➢ Docker Container on (Jenkins Server, Docker Server, Minikube Cluster)		(cadvisor, redis)

- Installing Prometheus/Grafana (192.168.178.104)

The following commands have been used to install prometheus:

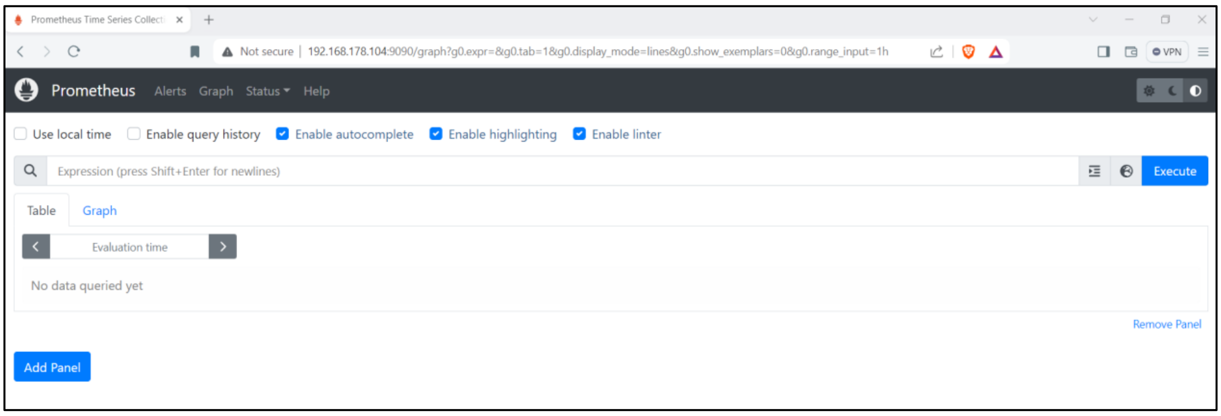
```
# Updating the System
sudo apt update && sudo apt upgrade -y
# Creating Promethues User
sudo useradd -no-create-home -shell /bin/false promtheus
# On Server go to /tmp and download prometheus using wget
wget
https://github.com/prometheus/prometheus/releases/download/v2.33.5/prome
theus-2.33.5.linux-amd64.tar.gz
# untar and extract the package
tar -xvfz prometheus-2.33.5.linux-amd64.tar.gz
# Moving Configuration files and binaries
sudo mv /tmp/prometheus-2.33.5.linux-amd64/prometheus /usr/local/bin
sudo mv /tmp/prometheus-2.33.5.linux-amd64/promtool /usr/local/bin
sudo mv /tmp/prometheus-2.33.5.linux-amd64/console_libraries
/etc/prometheus
sudo mv /tmp/prometheus-2.33.5.linux-amd64/prometheus.yml
/etc/prometheus
# Set the correct ownership for the files
sudo chown prometheus:prometheus /usr/local/bin/prometheus
sudo chown prometheus:prometheus /usr/local/bin/promtool
sudo chown -R prometheus:prometheus /etc/prometheus/consoles
sudo chown -R prometheus:prometheus /etc/prometheus/console_libraries
# Create the directory where Prometheus will store its data:
sudo mkdir /var/lib/prometheus
sudo chown prometheus:prometheus /var/lib/prometheus
# Creating a Prometheus Service
sudo vi /etc/systemd/system/prometheus.service
# Include the following configuration:
[Unit]
Description=Prometheus Monitoring
```

```
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
 --config.file=/etc/prometheus/prometheus.yml \
 --storage.tsdb.path=/var/lib/prometheus/ \
 --web.console.templates=/etc/prometheus/consoles \
 --web.console.libraries=/etc/prometheus/console_libraries

   [Install]
   WantedBy=multi-user.target

   # Reload the systemd daemon and start the Prometheus service:
   sudo systemctl daemon-reload
   sudo systemctl start prometheus
   sudo systemctl enable prometheus
   #  Verifying Prometheus Installation
   sudo systemctl status prometheus
```



```
sm@104-ubuntu22:~$ sudo systemctl status prometheus.service
[sudo] password for sm:
● prometheus.service - Prometheus Monitoring
     Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2024-06-06 08:14:41 CEST; 1h 34min ago
   Main PID: 1019 (prometheus)
      Tasks: 8 (limit: 4554)
     Memory: 101.9M
        CPU: 7.330s
     CGroup: /system.slice/prometheus.service
             └─1019 /usr/local/bin/prometheus --config.file=/etc/prometheus/prometheus.yml --storage.tsdb.path=/var/lib/prometheus/ --web.console.templates=

Jun 06 08:14:58 104-ubuntu22 prometheus[1019]: ts=2024-06-06T06:14:58.675Z caller=compact.go:576 level=info component=tsdb msg="write block" mint=1717617600
Jun 06 08:14:58 104-ubuntu22 prometheus[1019]: ts=2024-06-06T06:14:58.685Z caller=head.go:1345 level=info component=tsdb msg="Head GC completed" caller=trun
Jun 06 08:14:58 104-ubuntu22 prometheus[1019]: ts=2024-06-06T06:14:58.849Z caller=compact.go:576 level=info component=tsdb msg="write block" mint=1717624802
Jun 06 08:14:58 104-ubuntu22 prometheus[1019]: ts=2024-06-06T06:14:58.866Z caller=head.go:1345 level=info component=tsdb msg="Head GC completed" caller=trun
Jun 06 08:14:58 104-ubuntu22 prometheus[1019]: ts=2024-06-06T06:14:58.866Z caller=checkpoint.go:101 level=info component=tsdb msg="Creating checkpoint" from
Jun 06 08:14:59 104-ubuntu22 prometheus[1019]: ts=2024-06-06T06:14:59.038Z caller=head.go:1307 level=info component=tsdb msg="WAL checkpoint complete" first
Jun 06 08:14:59 104-ubuntu22 prometheus[1019]: ts=2024-06-06T06:14:59.371Z caller=compact.go:514 level=info component=tsdb msg="compact blocks" count=3 mint
Jun 06 08:14:59 104-ubuntu22 prometheus[1019]: ts=2024-06-06T06:14:59.378Z caller=db.go:1680 level=info component=tsdb msg="Deleting obsolete block" block=0
Jun 06 08:14:59 104-ubuntu22 prometheus[1019]: ts=2024-06-06T06:14:59.380Z caller=db.go:1680 level=info component=tsdb msg="Deleting obsolete block" block=0
Jun 06 08:14:59 104-ubuntu22 prometheus[1019]: ts=2024-06-06T06:14:59.388Z caller=db.go:1680 level=info component=tsdb msg="Deleting obsolete block" block=0
```

```
# Checking http://192.168.178.104:9090
```


Prometheus Time Series Collect
Not secure | 192.168.178.104:9090/graph?g0.expr=&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range_input=1h
VPN
Prometheus
Alerts
Graph
Status
Help
Execute
Table
Graph
Evaluation time
Remove Panel
Add Panel

Use local time [ ] · Enable query history [ ] · Enable autocomplete [✓] · Enable highlighting [✓] · Enable linter [✓]

Expression (press Shift+Enter for newlines)

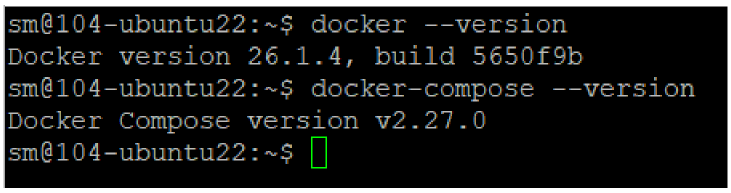
No data queried yet

- Monitoring Docker Containers

In order to use prometheus to monitor docker containers, we need to install docker engine, docker-compose, prepare docker-compose yaml file, and change the configuration of prometheus yaml file to run docker images (cadvisor, redis)

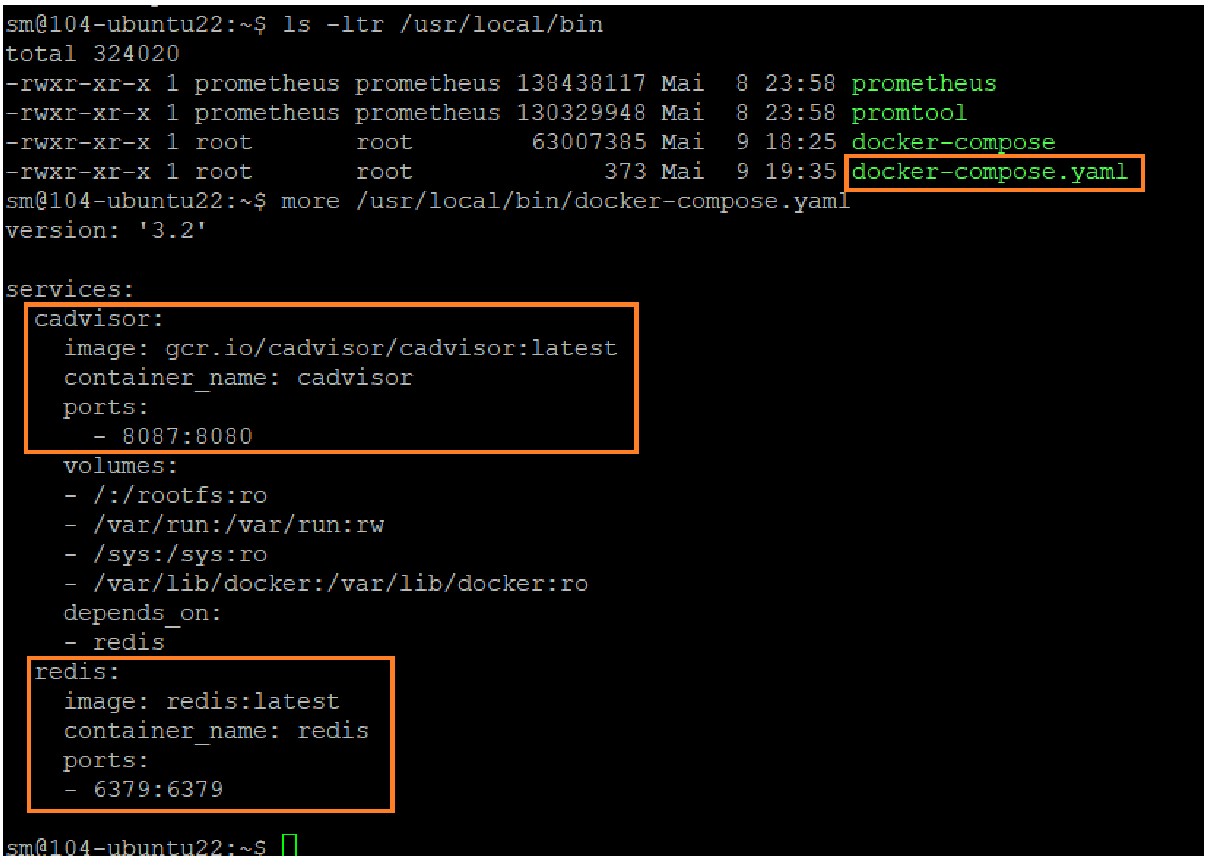
# Update package repositories for docker, adding keyrings (done in erlier stage)
# Installing latest version

sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin



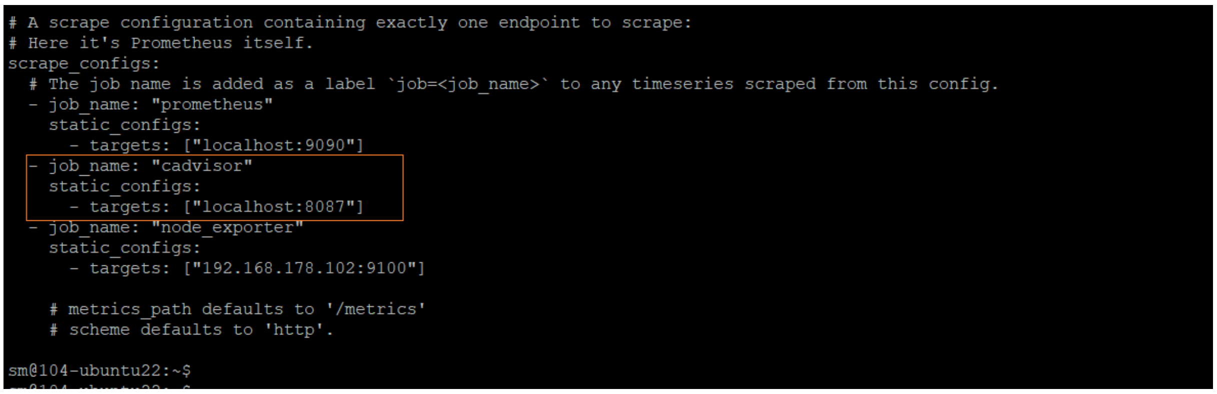
```
sm@104-ubuntu22:~$ docker --version
Docker version 26.1.4, build 5650f9b
sm@104-ubuntu22:~$ docker-compose --version
Docker Compose version v2.27.0
sm@104-ubuntu22:~$
```

# For monitoring work on prometheus server, two docker containers should be build with docker compose

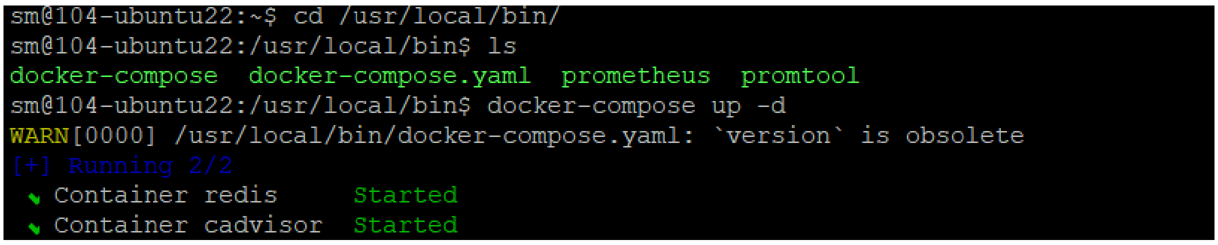


```
sm@104-ubuntu22:~$ ls -ltr /usr/local/bin
total 324020
-rwxr-xr-x 1 prometheus prometheus 138438117 Mai  8 23:58 prometheus
-rwxr-xr-x 1 prometheus prometheus 130329948 Mai  8 23:58 promtool
-rwxr-xr-x 1 root       root        63007385 Mai  9 18:25 docker-compose
-rwxr-xr-x 1 root       root             373 Mai  9 19:35 docker-compose.yaml
sm@104-ubuntu22:~$ more /usr/local/bin/docker-compose.yaml
version: '3.2'

services:
  cadvisor:
    image: gcr.io/cadvisor/cadvisor:latest
    container_name: cadvisor
    ports:
      - 8087:8080
    volumes:
    - /:/rootfs:ro
    - /var/run:/var/run:rw
    - /sys:/sys:ro
    - /var/lib/docker:/var/lib/docker:ro
    depends_on:
    - redis
  redis:
    image: redis:latest
    container_name: redis
    ports:
    - 6379:6379

sm@104-ubuntu22:~$
```

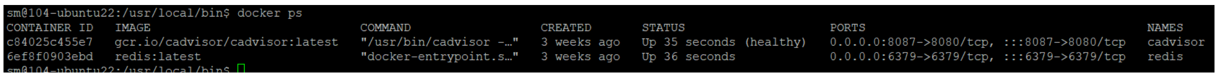
# Then prometheus Configuration yaml file in (/etc/prometheus/prometheus.yaml) should edited



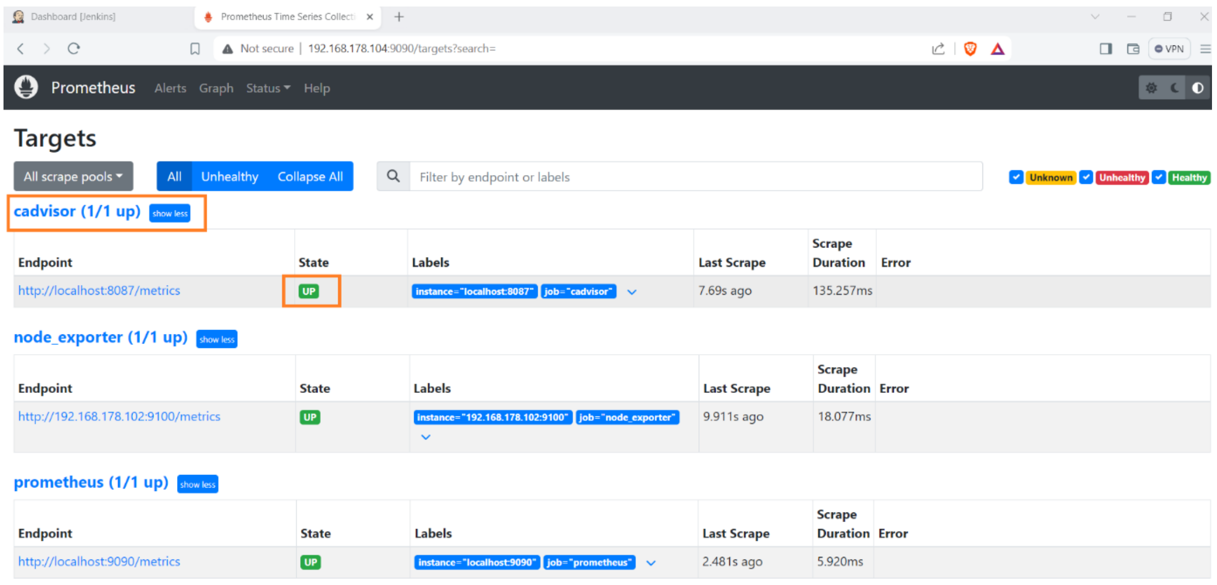
```
# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]
  - job_name: "cadvisor"
    static_configs:
      - targets: ["localhost:8087"]
  - job_name: "node_exporter"
    static_configs:
      - targets: ["192.168.178.102:9100"]

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

sm@104-ubuntu22:~$
```



```
sm@104-ubuntu22:~$ cd /usr/local/bin/
sm@104-ubuntu22:/usr/local/bin$ ls
docker-compose  docker-compose.yaml  prometheus  promtool
sm@104-ubuntu22:/usr/local/bin$ docker-compose up -d
WARN[0000] /usr/local/bin/docker-compose.yaml: `version` is obsolete
[+] Running 2/2
 ✔ Container redis      Started
 ✔ Container cadvisor   Started
```



```
sm@104-ubuntu22:/usr/local/bin$ docker ps
CONTAINER ID   IMAGE                             COMMAND                  CREATED       STATUS                    PORTS                                       NAMES
c84025c455e7   gcr.io/cadvisor/cadvisor:latest   "/usr/bin/cadvisor -…"   3 weeks ago   Up 35 seconds (healthy)   0.0.0.0:8087->8080/tcp, :::8087->8080/tcp   cadvisor
6ef8f0903ebd   redis:latest                      "docker-entrypoint.s…"   3 weeks ago   Up 36 seconds             0.0.0.0:6379->6379/tcp, :::6379->6379/tcp   redis
```



Targets

cadvisor (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:8087/metrics	UP	instance="localhost:8087" job="cadvisor"	7.69s ago	135.257ms	

node_exporter (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.178.102:9100/metrics	UP	instance="192.168.178.102:9100" job="node_exporter"	9.911s ago	18.077ms	

prometheus (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	2.481s ago	5.920ms	

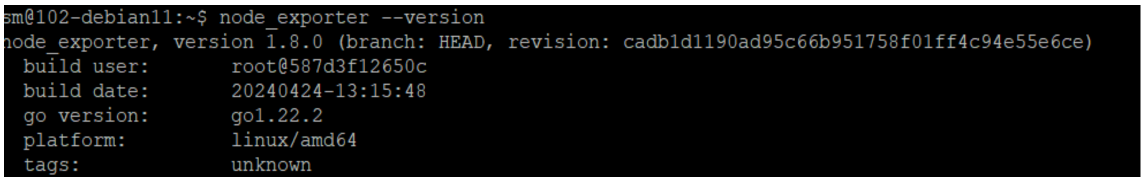
- Installing Node Exporter on (192.168.178.102)

# On 192.168.178.102 download Node Exporter package

curl -s https://api.github.com/repos/prometheus/node_exporter/releases/latest| grep browser_download_url|grep linux-amd64|cut -d '"' -f 4|wget -qi -

tar -xvf node_exporter*.tar.gz
sudo cp node_exporter /usr/local/bin
# Adding user and grou “prometheus:prometheus”

```
sudo groupadd –system prometheus
sudo useradd -s /sbin/nologin –system -g prometheus prometheus
# Change the ownership of the file node_exporter
sudo chown prometheus:prometheus /usr/local/bin/node_exporter
```

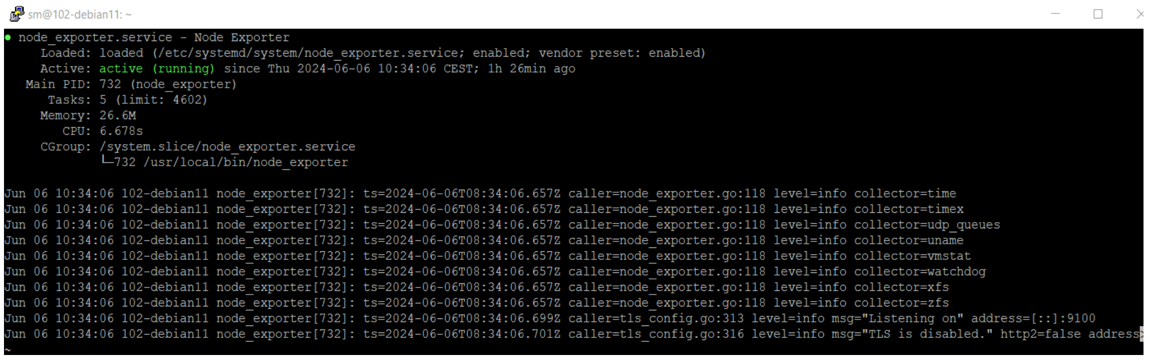


```
sm@102-debian11:~$ node_exporter --version
node_exporter, version 1.8.0 (branch: HEAD, revision: cadb1d1190ad95c66b951758f01ff4c94e55e6ce)
  build user:       root@587d3f12650c
  build date:       20240424-13:15:48
  go version:       go1.22.2
  platform:         linux/amd64
  tags:             unknown
```

```
# Create node_exporter service
sudo tee /etc/systemd/system/node_exporter.service <<EOF
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=default.target
EOF
```

```
# Reload systemd and start the service
sudo systemctl daemon-reload
sudo systemctl start node_exporter
sudo systemctl enable node_exporter
```

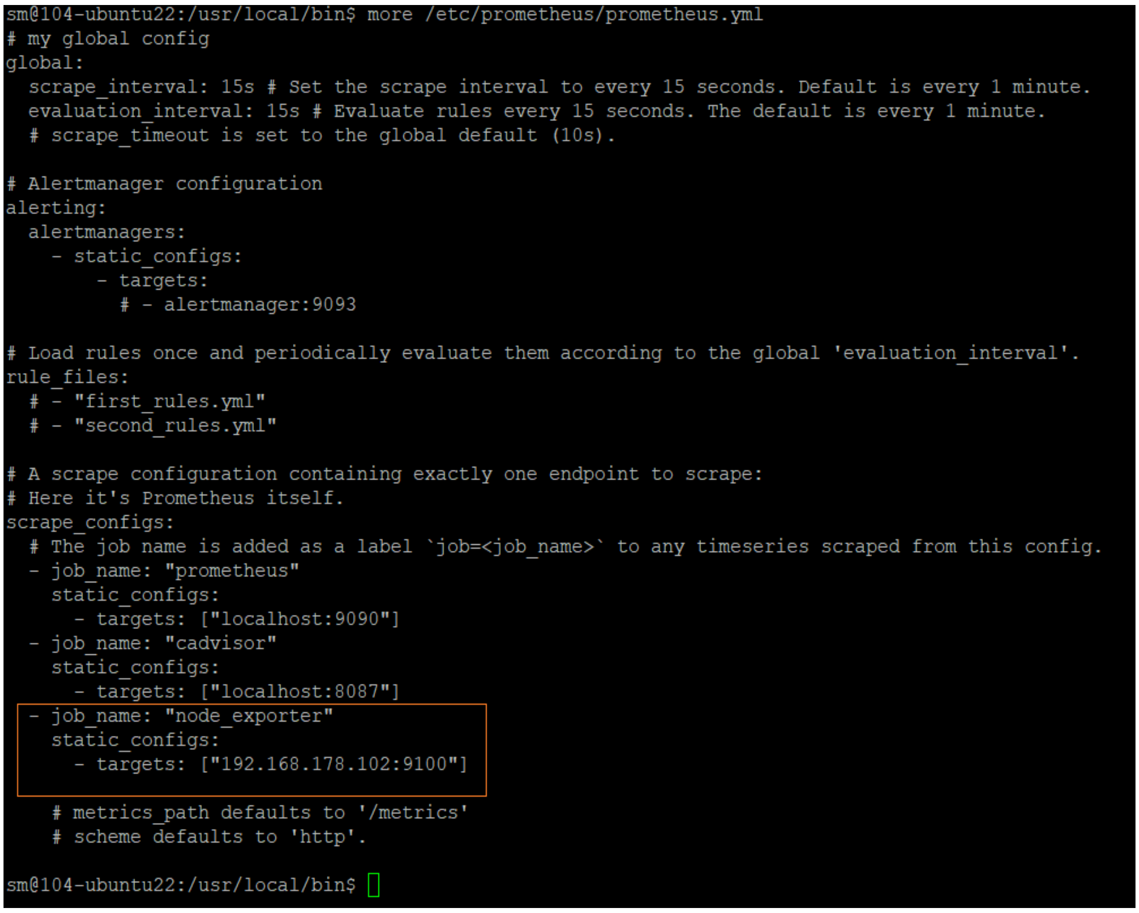
```
# Status Confirmation
```



sm@102-debian11: ~

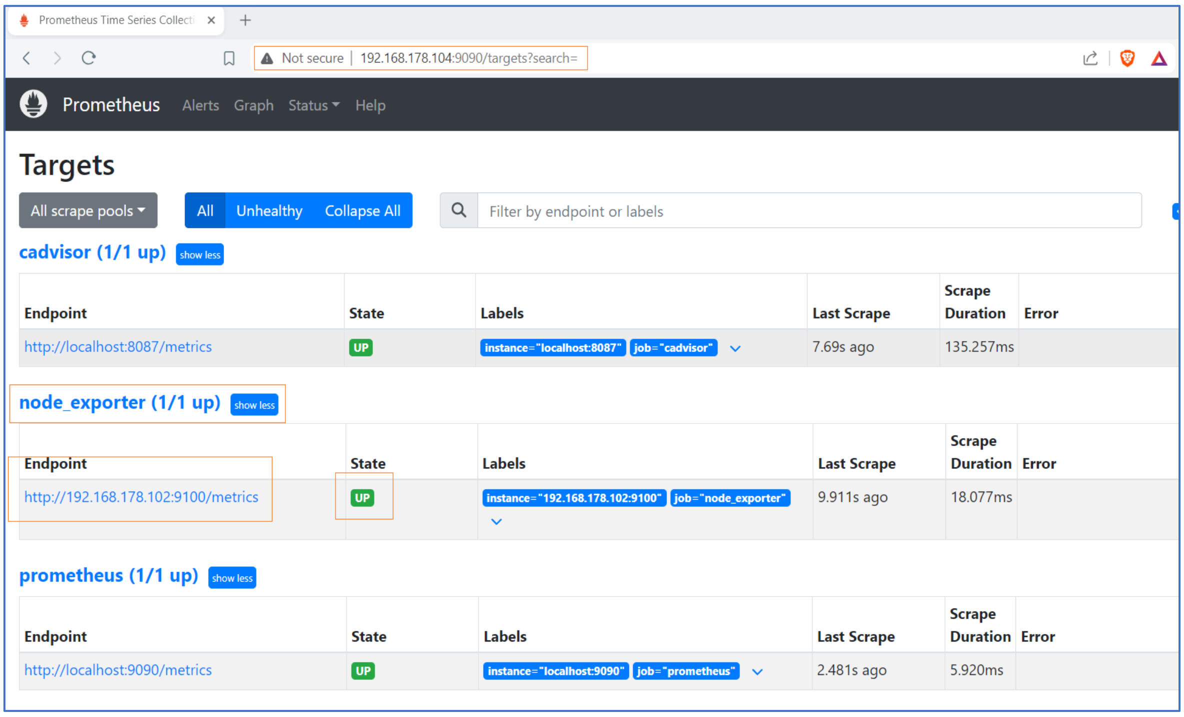
```
● node_exporter.service - Node Exporter
     Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2024-06-06 10:34:06 CEST; 1h 26min ago
   Main PID: 732 (node_exporter)
      Tasks: 5 (limit: 4602)
     Memory: 26.6M
        CPU: 6.678s
     CGroup: /system.slice/node_exporter.service
             └─732 /usr/local/bin/node_exporter

Jun 06 10:34:06 102-debian11 node_exporter[732]: ts=2024-06-06T08:34:06.657Z caller=node_exporter.go:118 level=info collector=time
Jun 06 10:34:06 102-debian11 node_exporter[732]: ts=2024-06-06T08:34:06.657Z caller=node_exporter.go:118 level=info collector=timex
Jun 06 10:34:06 102-debian11 node_exporter[732]: ts=2024-06-06T08:34:06.657Z caller=node_exporter.go:118 level=info collector=udp_queues
Jun 06 10:34:06 102-debian11 node_exporter[732]: ts=2024-06-06T08:34:06.657Z caller=node_exporter.go:118 level=info collector=uname
Jun 06 10:34:06 102-debian11 node_exporter[732]: ts=2024-06-06T08:34:06.657Z caller=node_exporter.go:118 level=info collector=vmstat
Jun 06 10:34:06 102-debian11 node_exporter[732]: ts=2024-06-06T08:34:06.657Z caller=node_exporter.go:118 level=info collector=watchdog
Jun 06 10:34:06 102-debian11 node_exporter[732]: ts=2024-06-06T08:34:06.657Z caller=node_exporter.go:118 level=info collector=xfs
Jun 06 10:34:06 102-debian11 node_exporter[732]: ts=2024-06-06T08:34:06.657Z caller=node_exporter.go:118 level=info collector=zfs
Jun 06 10:34:06 102-debian11 node_exporter[732]: ts=2024-06-06T08:34:06.699Z caller=tls_config.go:313 level=info msg="Listening on" address=[::]:9100
Jun 06 10:34:06 102-debian11 node_exporter[732]: ts=2024-06-06T08:34:06.701Z caller=tls_config.go:316 level=info msg="TLS is disabled." http2=false address>
~
```

```
# Back to Prometheus Server to add the following to prometheus.yaml file
```



```
sm@104-ubuntu22:/usr/local/bin$ more /etc/prometheus/prometheus.yml
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
          # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]
  - job_name: "cadvisor"
    static_configs:
      - targets: ["localhost:8087"]
  - job_name: "node_exporter"
    static_configs:
      - targets: ["192.168.178.102:9100"]

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

sm@104-ubuntu22:/usr/local/bin$ 
```

# Restart Prometheus service
sudo systemctl restart prometheus


Prometheus Time Series Collecti
Not secure | 192.168.178.104:9090/targets?search=
Prometheus
Alerts
Graph
Status
Help
All scrape pools
All
Unhealthy
Collapse All
Filter by endpoint or labels
show less

Targets

cadvisor (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:8087/metrics	UP	instance="localhost:8087" job="cadvisor"	7.69s ago	135.257ms	

node_exporter (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.178.102:9100/metrics	UP	instance="192.168.178.102:9100" job="node_exporter"	9.911s ago	18.077ms	

prometheus (1/1 up)

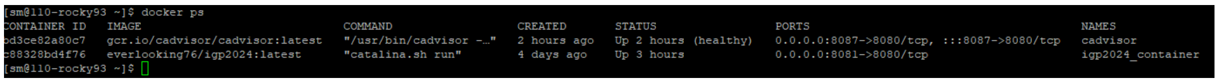
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	2.481s ago	5.920ms	

# Similarly, the Node Export will be implemented on other nodes

# The cAdvisor container is pulled and run on all other nodes so that the metrics of docker containers are included too
# The command to run the cAdvisor is

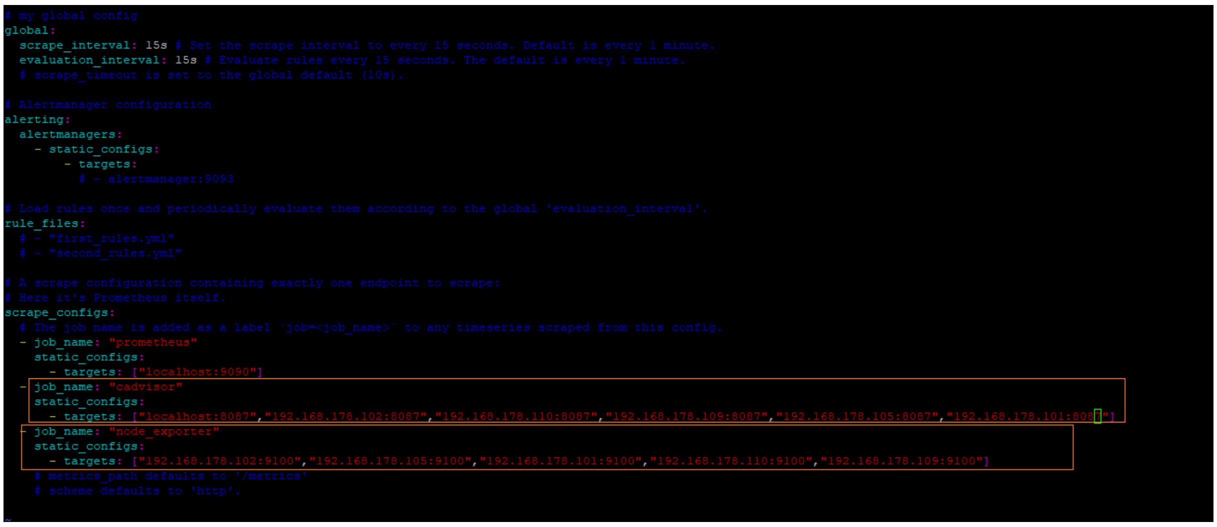
docker run -d --name=cadvisor --volume=/var/run/docker.sock:/var/run/docker.sock --volume=/sys:/sys --volume=/var/lib/docker/:/var/lib/docker:ro --publish=8087:8080 --detach=true --restart=always gcr.io/cadvisor/cadvisor:latest

# An Example of the running containers on 192.168.178.110



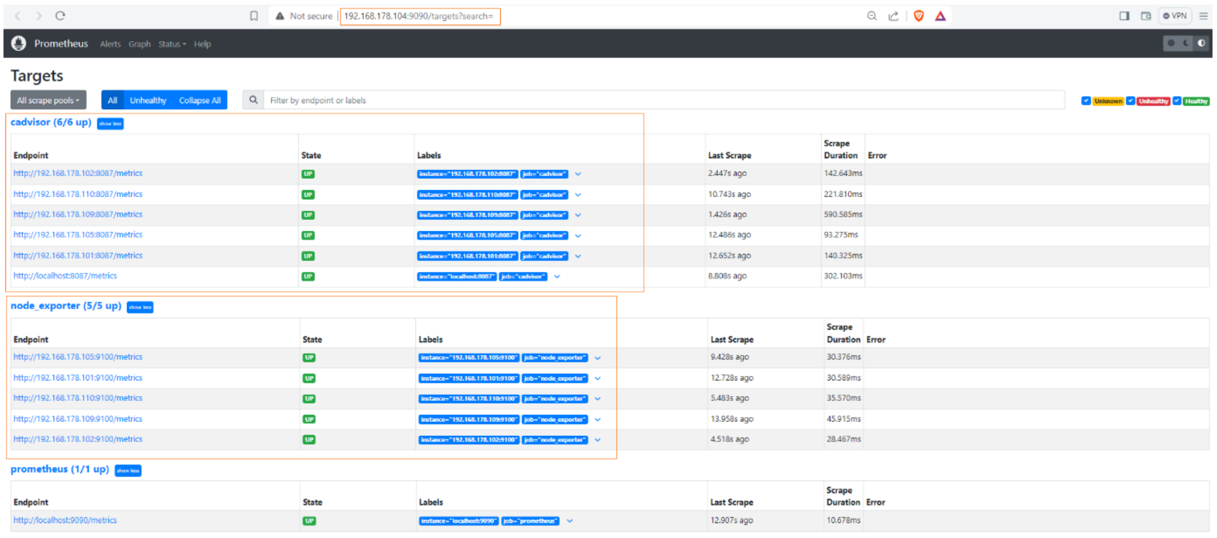
```
[sm@110-rocky93 ~]$ docker ps
CONTAINER ID   IMAGE                              COMMAND                  CREATED       STATUS                 PORTS                                       NAMES
bd3ce82a80c7   gcr.io/cadvisor/cadvisor:latest    "/usr/bin/cadvisor -…"   2 hours ago   Up 2 hours (healthy)   0.0.0.0:8087->8080/tcp, :::8087->8080/tcp   cadvisor
c88328bd4f76   everlooking76/igp2024:latest       "catalina.sh run"        4 days ago    Up 3 hours             0.0.0.0:8081->8080/tcp                      igp2024_container
[sm@110-rocky93 ~]$
```

# On Prometheus Server(master)



```
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
          # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]
  - job_name: "cadvisor"
    static_configs:
      - targets: ["localhost:8087","192.168.178.102:8087","192.168.178.110:8087","192.168.178.109:8087","192.168.178.105:8087","192.168.178.101:8087"]
  - job_name: "node_exporter"
    static_configs:
      - targets: ["192.168.178.102:9100","192.168.178.105:9100","192.168.178.101:9100","192.168.178.110:9100","192.168.178.109:9100"]
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
~
```

# Prometheus dashboard



Not secure | 192.168.178.104:9090/targets?search=

Prometheus Alerts Graph Status Help

Targets

All scrape pools | All | Unhealthy | Collapse All | Filter by endpoint or labels | Unknown | Unhealthy | Healthy

cadvisor (6/6 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.178.102:8087/metrics	UP	instance="192.168.178.102:8087" job="cadvisor"	2.447s ago	142.643ms	
http://192.168.178.110:8087/metrics	UP	instance="192.168.178.110:8087" job="cadvisor"	10.743s ago	221.810ms	
http://192.168.178.109:8087/metrics	UP	instance="192.168.178.109:8087" job="cadvisor"	1.426s ago	590.585ms	
http://192.168.178.105:8087/metrics	UP	instance="192.168.178.105:8087" job="cadvisor"	12.486s ago	93.275ms	
http://192.168.178.101:8087/metrics	UP	instance="192.168.178.101:8087" job="cadvisor"	12.652s ago	140.325ms	
http://localhost:8087/metrics	UP	instance="localhost:8087" job="cadvisor"	8.808s ago	302.103ms	

node_exporter (5/5 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.178.105:9100/metrics	UP	instance="192.168.178.105:9100" job="node_exporter"	9.428s ago	30.376ms	
http://192.168.178.101:9100/metrics	UP	instance="192.168.178.101:9100" job="node_exporter"	12.728s ago	30.589ms	
http://192.168.178.110:9100/metrics	UP	instance="192.168.178.110:9100" job="node_exporter"	5.483s ago	35.570ms	
http://192.168.178.109:9100/metrics	UP	instance="192.168.178.109:9100" job="node_exporter"	13.958s ago	45.915ms	
http://192.168.178.102:9100/metrics	UP	instance="192.168.178.102:9100" job="node_exporter"	4.518s ago	28.467ms	

prometheus (1/1 up) show less

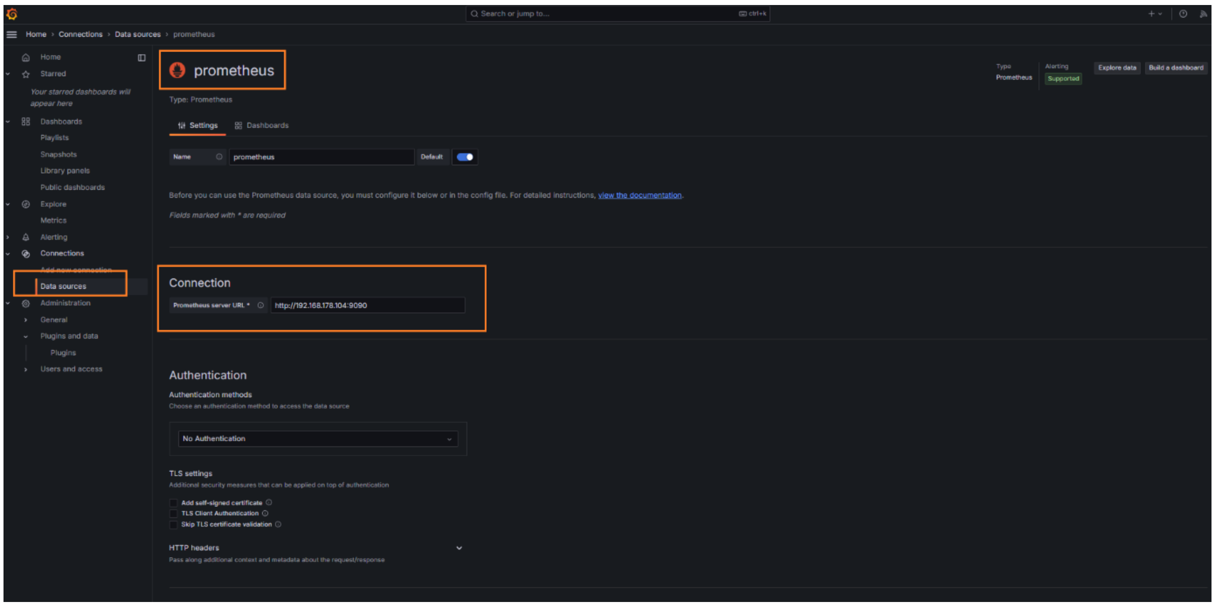
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	12.907s ago	10.678ms	

- Installing Grafana Dashboard

# Similarly, to previous steps, the package repos are update and to install Grafana we use this command

sudo apt install grafana

# Starting Grafana Service, using the URL http://192.168.178.104:3000 (admin/admin)
sudo systemctl start grafana-server


Home › Connections › Data sources › prometheus

prometheus

Type: Prometheus

Settings | Dashboards

Name: prometheus | Default: on

Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, view the documentation.

*Fields marked with * are required*

Connection

Prometheus server URL *: http://192.168.178.104:9090

Authentication

Authentication methods

Choose an authentication method to access the data source

No Authentication

TLS settings

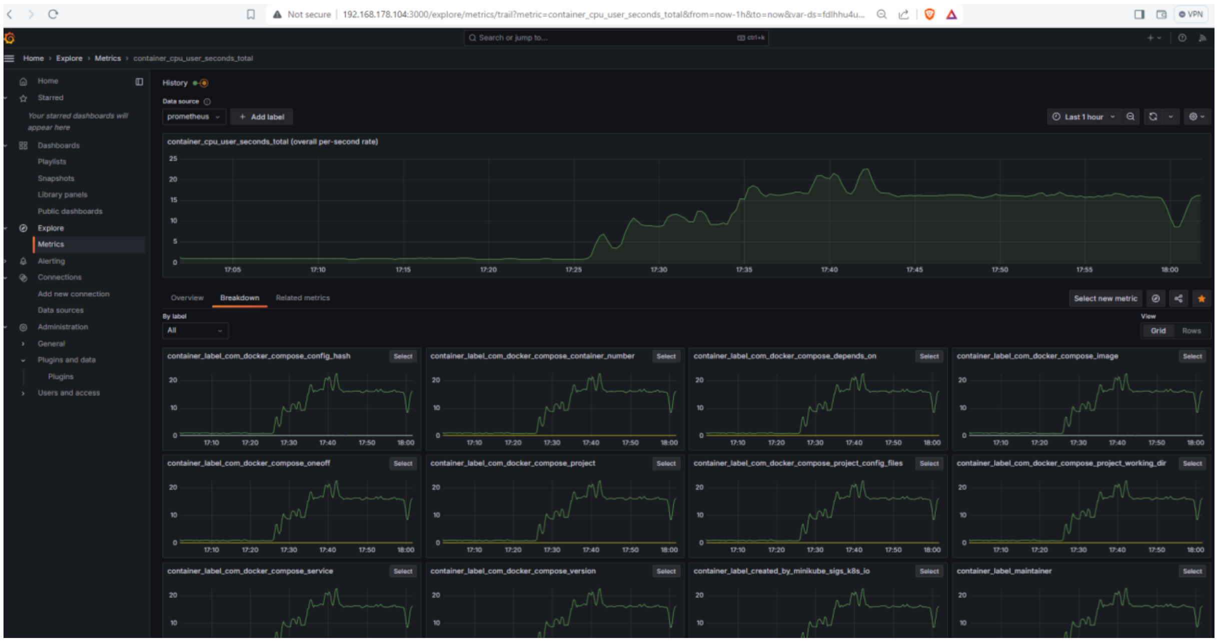
Additional security measures that can be applied on top of authentication

- Add self-signed certificate
- TLS Client Authentication
- Skip TLS certificate validation

HTTP headers

Pass along additional context and metadata about the request/response

Home
Starred
Dashboards
Explore
Alerting
Connections
Data sources
Administration
Type Prometheus
Alerting Supported
Explore data
Build a dashboard

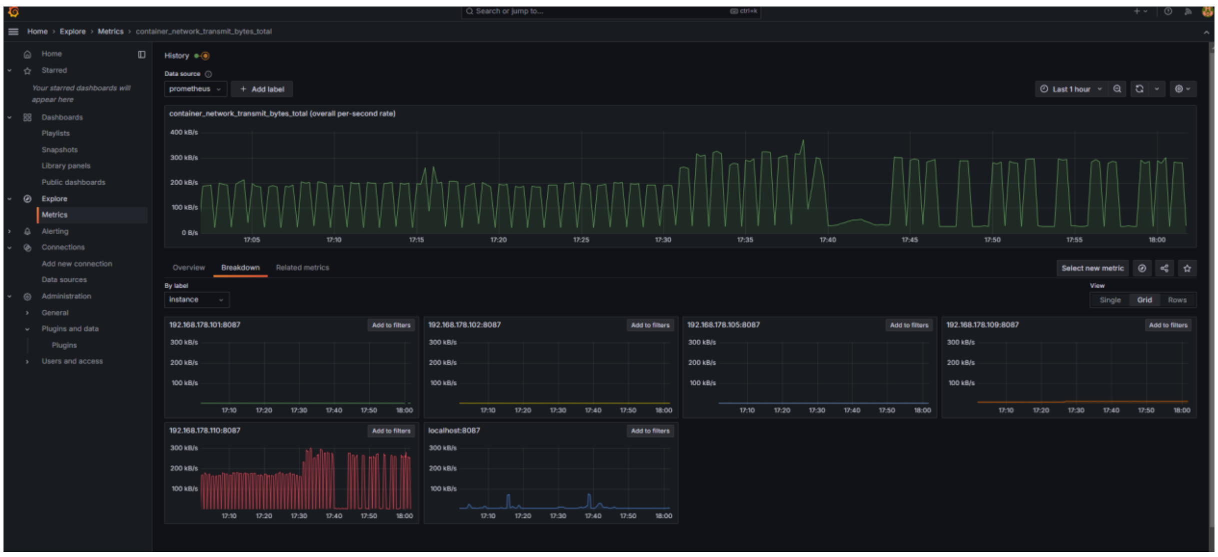

Not secure | 192.168.178.104:3000/explore/metrics/trail?metric=container_cpu_user_seconds_total&from=now-1h&to=now&var-ds=fdlhhu4u...

Home › Explore › Metrics › container_cpu_user_seconds_total

Data source: prometheus

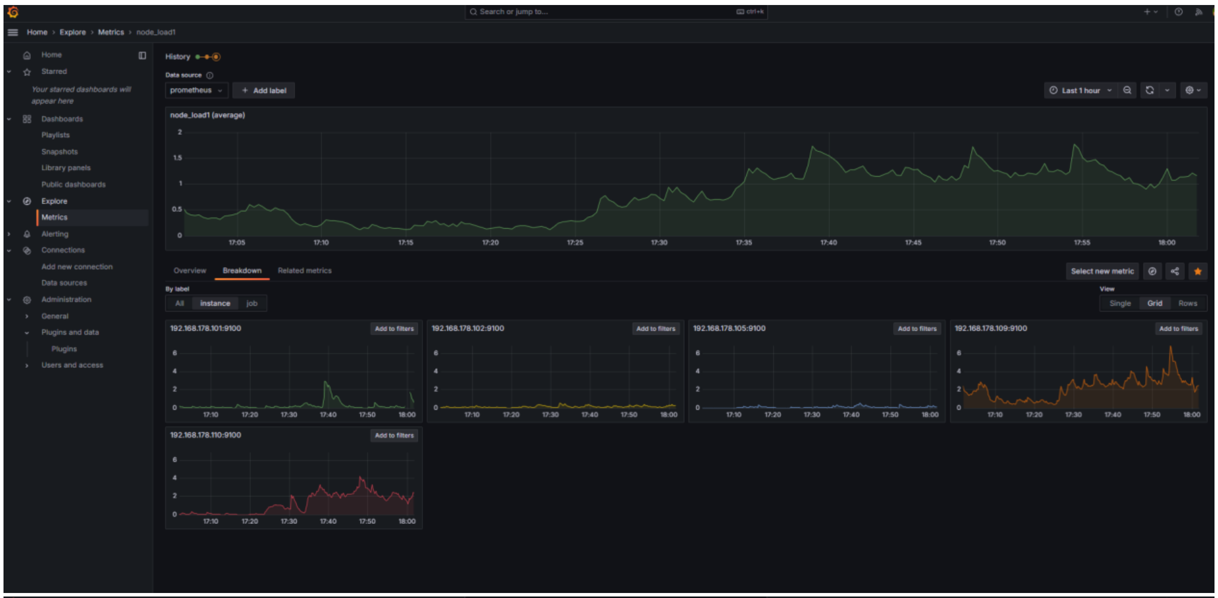
container_cpu_user_seconds_total (overall per-second rate)

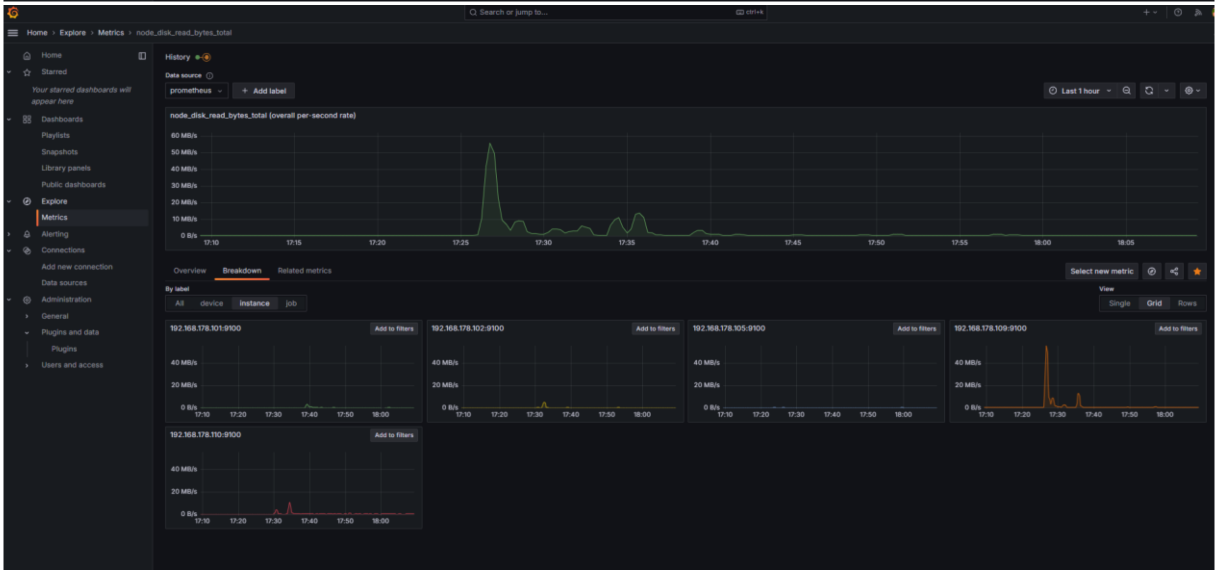
History
Add label
Last 1 hour
Overview
Breakdown
Related metrics
Select new metric
By label
All
View
Grid
Rows
container_label_com_docker_compose_config_hash
container_label_com_docker_compose_container_number
container_label_com_docker_compose_depends_on
container_label_com_docker_compose_image
container_label_com_docker_compose_oneoff
container_label_com_docker_compose_project
container_label_com_docker_compose_project_config_files
container_label_com_docker_compose_project_working_dir
container_label_com_docker_compose_service
container_label_com_docker_compose_version
container_label_created_by_minikube_sigs_k8s_io
container_label_maintainer
Select

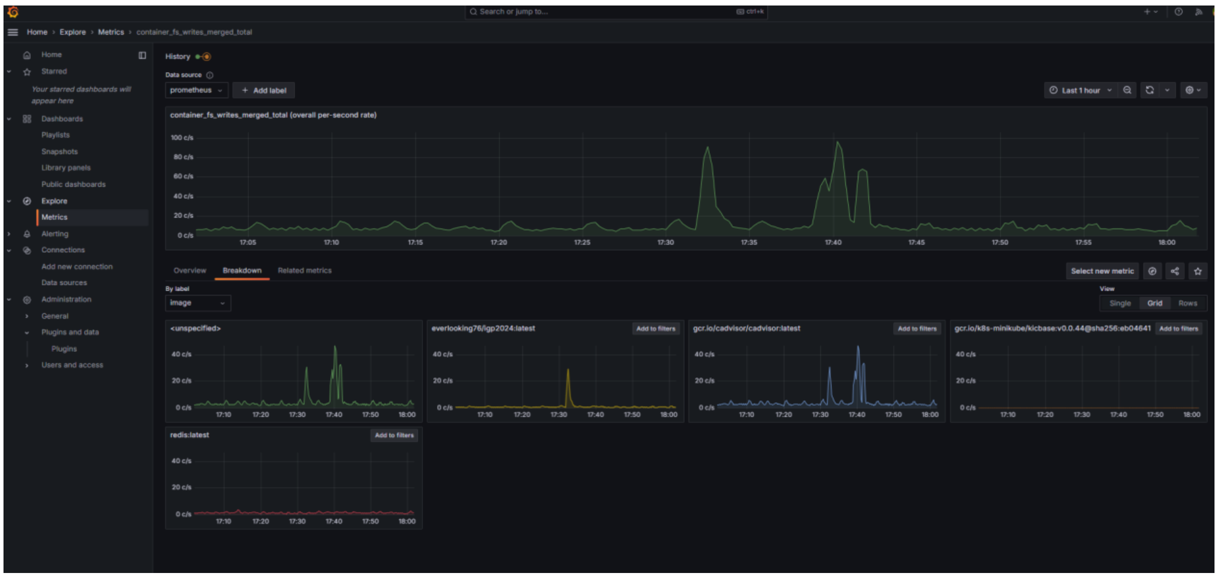

Home › Explore › Metrics › container_network_transmit_bytes_total

Data source: prometheus

container_network_transmit_bytes_total (overall per-second rate)


Home › Explore › Metrics › node_load1
History
Data source
prometheus
Add label
Last 1 hour
node_load1 (average)
Overview
Breakdown
Related metrics
Select new metric
By label
All
instance
job
View
Single
Grid
Rows
192.168.178.101:9100
192.168.178.102:9100
192.168.178.105:9100
192.168.178.109:9100
192.168.178.110:9100
Add to filters


Home › Explore › Metrics › node_disk_read_bytes_total
History
Data source
prometheus
Add label
Last 1 hour
node_disk_read_bytes_total (overall per-second rate)
Overview
Breakdown
Related metrics
Select new metric
By label
All
device
instance
job
View
Single
Grid
Rows
192.168.178.101:9100
192.168.178.102:9100
192.168.178.105:9100
192.168.178.109:9100
192.168.178.110:9100
Add to filters


Home › Explore › Metrics › container_fs_writes_merged_total
History
Data source
prometheus
Add label
Last 1 hour
container_fs_writes_merged_total (overall per-second rate)
Overview
Breakdown
Related metrics
Select new metric
By label
image
View
Single
Grid
Rows
<unspecified>
everlooking76/igp2024:latest
gcr.io/cadvisor/cadvisor:latest
gcr.io/k8s-minikube/kicbase:v0.0.44@sha256:eb04641
redis:latest
Add to filters