IGP Project Submission – Name Vivek Gupta

I'm using 4 servers.

1 server's hostname as jump using for

    Docker, Jenkins, Git Maven

3 servers hostname master, worker-1 and worker-2 using for

    Kubernetes cluster.

Task 1

Install git

Download source code.

Initialization git

```
[vivek@jamp ABC_Technologies]$ 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 /source_code/ABC_Technologies/.git/
[vivek@jamp ABC_Technologies]$ git add
```

Code moves to staging area.

```
[vivek@jamp ABC_Technologies]$ git add .
[vivek@jamp ABC_Technologies]$
```

Git status

```
[vivek@jamp ABC_Technologies]$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .classpath
        new file:   .project
        new file:   .settings/org.eclipse.jdt.core.prefs
        new file:   .settings/org.eclipse.m2e.core.prefs
        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

[vivek@jamp ABC_Technologies]$ git commit -m "first code commit"
```

Commit the code.

```
[vivek@jamp ABC_Technologies]$ git commit -m "first code commit"
[master (root-commit) d12f241] first code commit
 13 files changed, 313 insertions(+)
 create mode 100644 .classpath
 create mode 100644 .project
 create mode 100644 .settings/org.eclipse.jdt.core.prefs
 create mode 100644 .settings/org.eclipse.m2e.core.prefs
 create mode 100644 README.md
 create mode 100644 pom.xml
 create mode 100644 pom.xml.bak
 create mode 100644 src/main/java/com/abc/RetailModule.java
 create mode 100644 src/main/java/com/abc/dataAccessObject/RetailAccessObject.java
 create mode 100644 src/main/java/com/abc/dataAccessObject/RetailDataImp.java
 create mode 100644 src/main/webapp/WEB-INF/web.xml
 create mode 100644 src/main/webapp/index.jsp
 create mode 100644 src/test/java/com/abc/dataAccessObject/ProductImpTest.java
[vivek@jamp ABC_Technologies]$
```

Code push to the github in master branch.

```
[vivek@jamp ABC_Technologies]$ git push -u origin master
Username for 'https://github.com': vgvivekgupt
Password for 'https://vgvivekgupt@github.com':
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Compressing objects: 100% (20/20), done.
Writing objects: 100% (29/29), 4.48 KiB | 1.49 MiB/s, done.
Total 29 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/vgvivekgupt/ABC_Technologies.git
 * [new branch]      master → master
branch 'master' set up to track 'origin/master'.
[vivek@jamp ABC_Technologies]$
```

Here is the reposistory.

Install Jenkins.
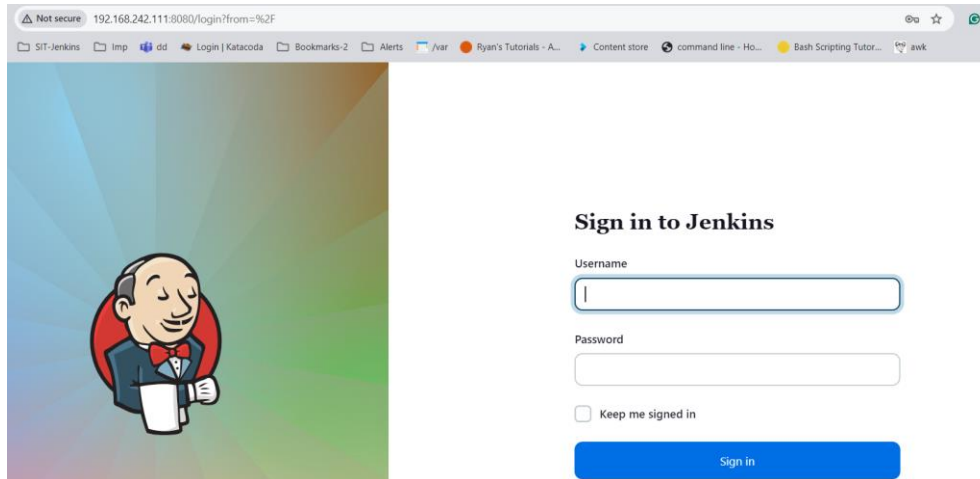
```
[vivek@jamp ABC_Technologies]$ sudo systemctl start jenkins
[vivek@jamp ABC_Technologies]$
[vivek@jamp ABC_Technologies]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled
   Active: active (running) since Wed 2024-10-02 12:48:56 EDT; 5min ago
 Main PID: 2073 (java)
    Tasks: 40 (limit: 17394)
   Memory: 619.0M
   CGroup: /system.slice/jenkins.service
           └─2073 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --w
```

```
[vivek@jamp ABC_Technologies]$ jenkins --version
2.462.1
```

Install maven.

Login to the Jenkins.

Creating pipeline folder for code.

I have attached the pipeline script in the github repo you can refer.

Run the pipeline to check the pipe on server using tomcat web-server.

	Declarative: Tool Install	Code-Compile	Code-Review	Code-Test	Code-Coverage	Code-Build	Code-Deploy	Declarative: Post Actions
Average stage times: (Average full run time: ~1min 10s)	381ms	10s	20s	11s	11s	10s	1s	186ms
#2 Oct 02 22:37 No Changes	345ms	7s	15s	7s	9s	9s	1s	286ms
#1 Oct 02 22:27 No Changes	417ms	13s	25s	15s	13s	10s	760ms	87ms

Pipeline working fine we can see the below snap.

Install Docker/

Login to the docker with user and password.

```
[vivek@jamp docker]$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker
eate one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizat
SO. Learn more at https://docs.docker.com/go/access-tokens/

Username: vgvivekgupt
Password:
WARNING! Your password will be stored unencrypted in /home/vivek/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```
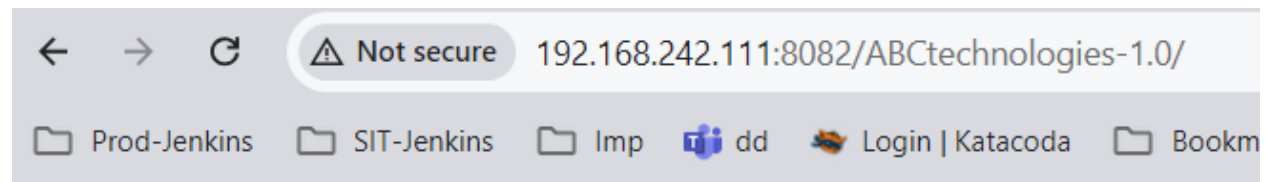
Building the image

```
[vivek@jamp docker]$ sudo docker build -t abctechnologies .
[+] Building 2.5s (8/8) FINISHED                                                                          docker:default
 ⇒ [internal] load build definition from Dockerfile                                                               0.0s
 ⇒ ⇒ transferring dockerfile: 393B                                                                                0.0s
 ⇒ [internal] load metadata for docker.io/library/tomcat:9                                                        2.4s
 ⇒ [internal] load .dockerignore                                                                                  0.0s
 ⇒ ⇒ transferring context: 2B                                                                                     0.0s
 ⇒ [internal] load build context                                                                                  0.0s
 ⇒ ⇒ transferring context: 47B                                                                                    0.0s
 ⇒ [1/3] FROM docker.io/library/tomcat:9@sha256:67f6b5f0d4061aea0cbd1e005d5987b13ccab5505aeecc75383203cf0650dfe3  0.0s
 ⇒ CACHED [2/3] COPY ABCtechnologies-1.0.war /usr/local/tomcat/webapps/                                           0.0s
 ⇒ CACHED [3/3] WORKDIR /usr/local/tomcat/bin                                                                     0.0s
 ⇒ exporting to image                                                                                             0.0s
 ⇒ ⇒ exporting layers                                                                                             0.0s
 ⇒ ⇒ writing image sha256:2f0b7c70e483c6644bb72dd3701591385c31c8d6c8f6b882839deb38a502a43c                        0.0s
 ⇒ ⇒ naming to docker.io/library/abctechnologies                                                                  0.0s
[vivek@jamp docker]$
```

Deploying the image with code to the container for testing.

```
CONTAINER ID   IMAGE    COMMAND            CREATED        STATUS        PORTS                                          NAMES
7c79f84d9478   app      "catalina.sh run"  5 minutes ago  Up 5 minutes  0.0.0.0:8082→8080/tcp, :::8082→8080/tcp        my-tomcat-container
```

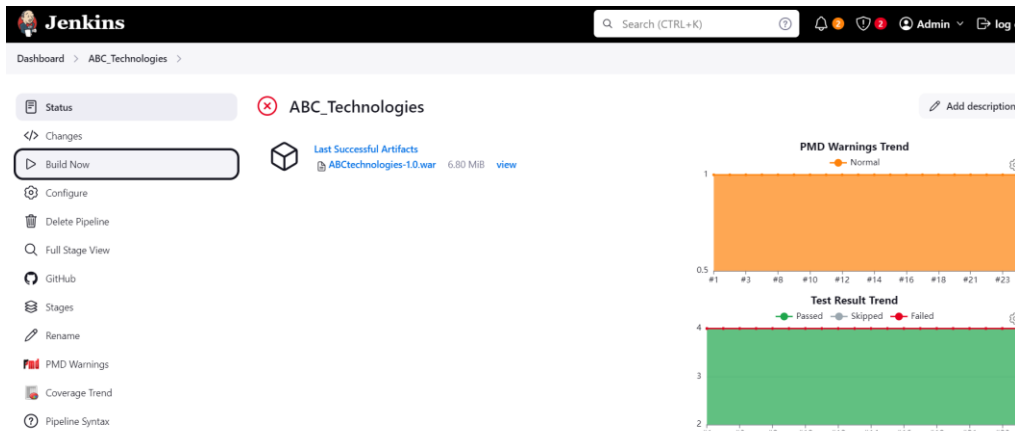Container working with we can check with IP with bind port,

← → C ⚠ Not secure 192.168.242.111:8082/ABCtechnologies-1.0/

📁 Prod-Jenkins 📁 SIT-Jenkins 📁 Imp 🔷 dd 🔶 Login | Katacoda 📁 Bookm

Welcome to ABC technologies

This is retail portal

[Add Product] [View Product]

Adding docker image build, running the container and push image to dockerhub.


Image has build after running pipeline.

```
[vivek@jamp ~]$ docker image ls -a
REPOSITORY                    TAG            IMAGE ID        CREATED         SIZE
abctechnologies               25             3903aa9472c0    12 hours ago    471MB
abctechnologies               26             3903aa9472c0    12 hours ago    471MB
abctechnologies               BUILD_NUMBER   3903aa9472c0    12 hours ago    471MB
vgvivekgupt/abctechnologies   latest         3903aa9472c0    12 hours ago    471MB
```

Container created after running pipeline.

```
[vivek@jamp ~]$ docker container ls -a
CONTAINER ID   IMAGE                                 COMMAND            CREATED            STATUS            PORTS                                              N
AMES
60ff3a31d3a4   vgvivekgupt/abctechnologies:latest    "catalina.sh run"  About a minute ago Up About a minute 0.0.0.0:8082→8080/tcp, :::8082→8080/tcp   a
bctechnologies
[vivek@jamp ~]$
```

Build image has push to the dockerhub.
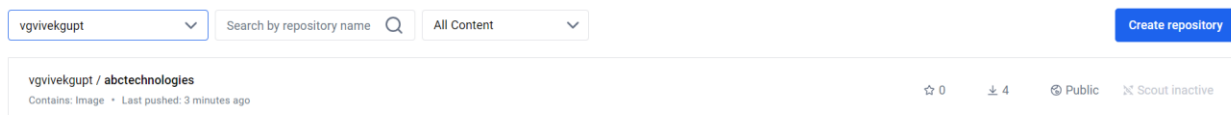
Pipeline working file we can see the bind port with IP of container.

Now deploying source code to the Kubernetes using 1 master 2 worker. We can see in below snap.

Writing the deployment and service file and creating pod.

```
[vivek@master ~]$ kubectl apply -f deployment.yml
deployment.apps/abctechnologies created
[vivek@master ~]$
[vivek@master ~]$
[vivek@master ~]$ kubectl apply -f service.yml
service/service-abctechnologies created
```

Here we can see the pods.

Created the secret file for dockerhub.

```
[vivek@master ~]$ kubectl get secrets -o wide
NAME                 TYPE                                 DATA    AGE
dockerhub-secret     kubernetes.io/dockerconfigjson       1       14m
[vivek@master ~]$
```

Here we can see pod is working with help of nodePort.

Doing setup for Kubernetes monitoring.

These are the repo have to install

1. helm repo add stable https://charts.helm.sh/stable
2. helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
3. helm search repo prometheus-community
4. kubectl create namespace Prometheus

Here we can see the pods of monitoring
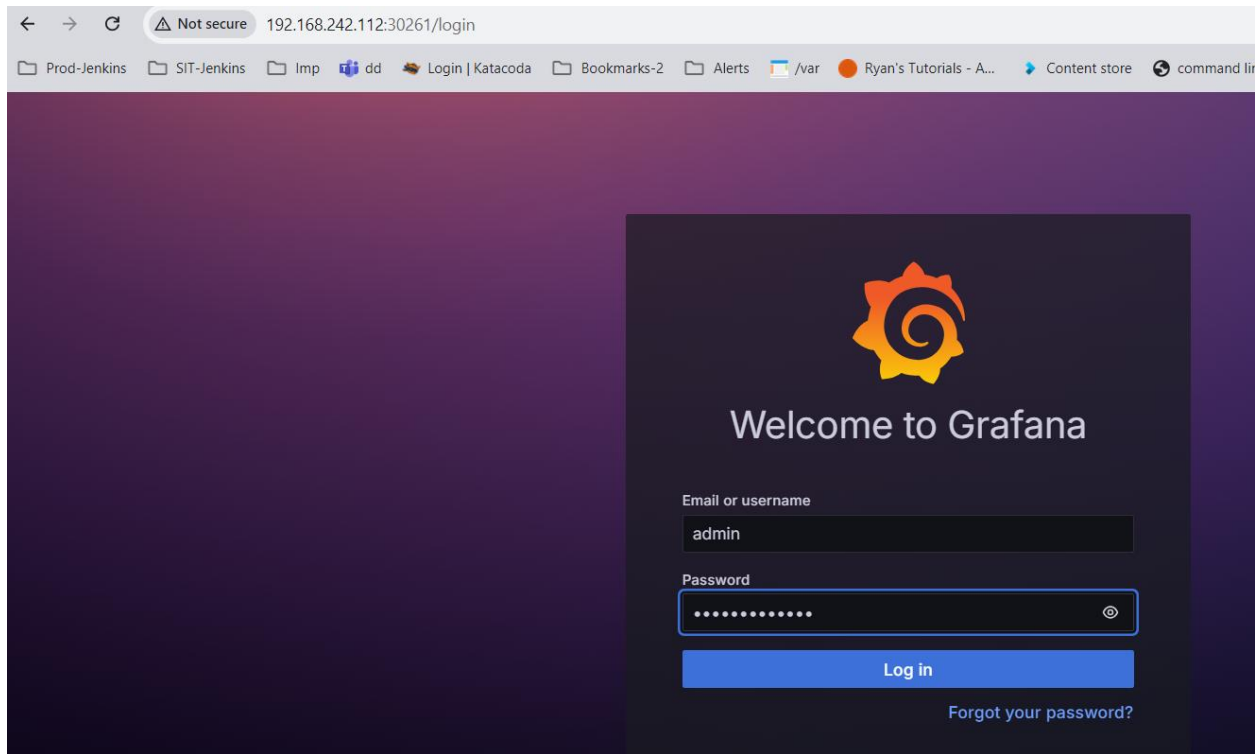
Here we can see the list of services

Login Grafana dashboard with help of NodePort and use and password
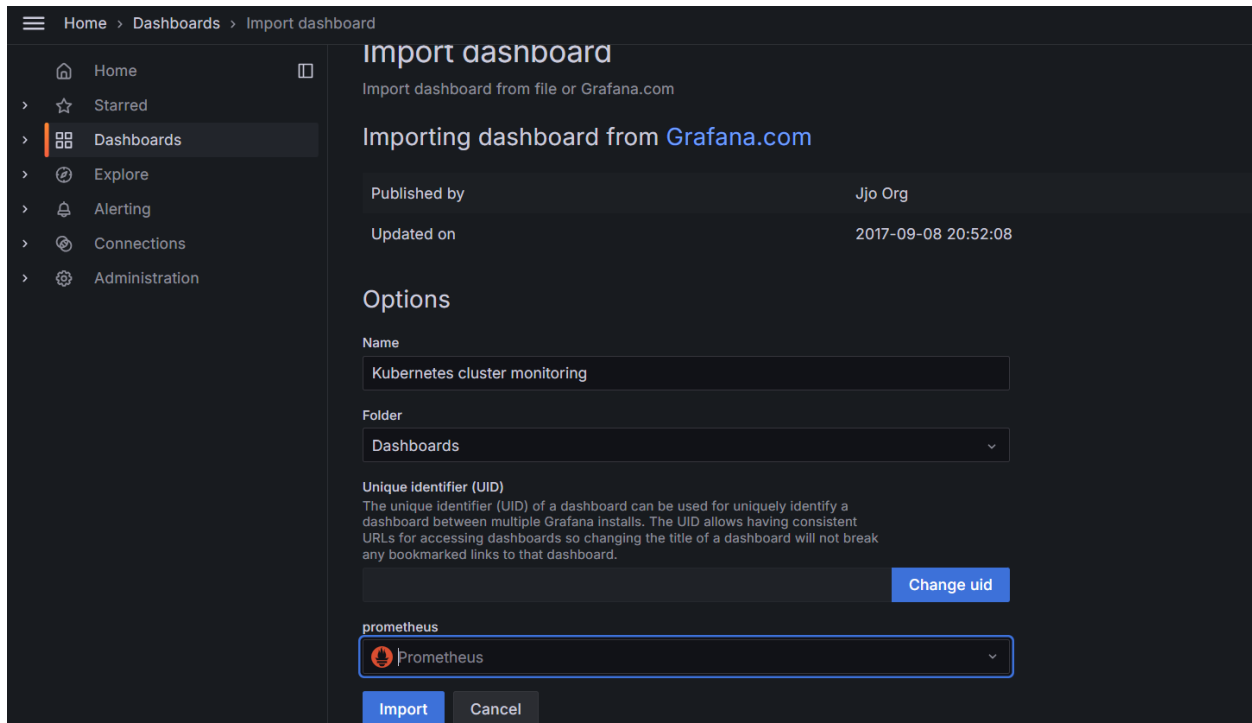
Importing the dashboard for monitoring.
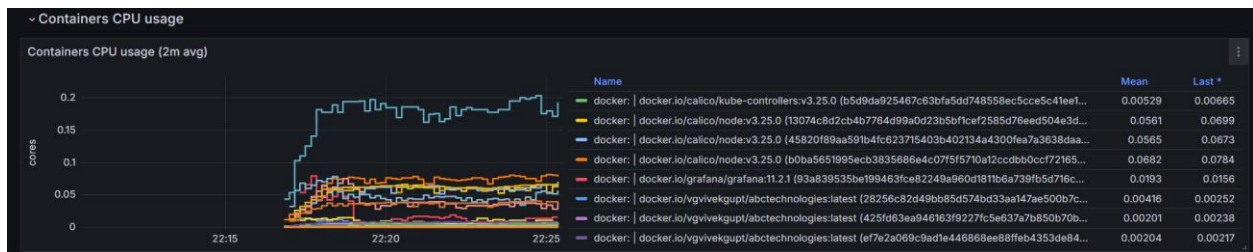
Here we can see the network input and output monitoring data.
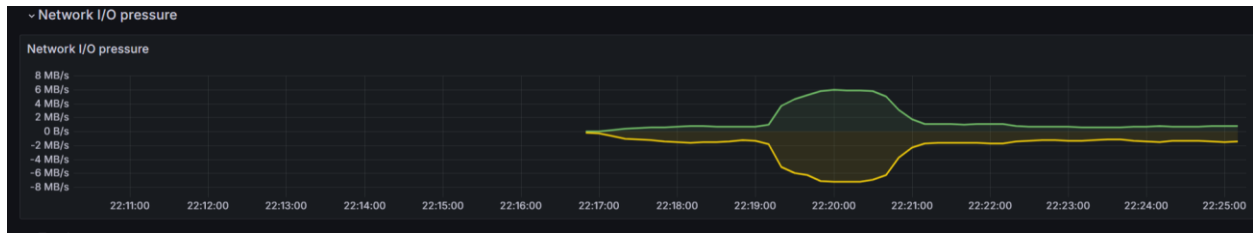
Here we can see the container CPU utilization monitoring data.

Here we can see the container memory utilization monitoring data.

Here is the Prometheus dashboard to get query to monitoring data fetch.