PROJECT -1

BUILDING A CI/CD PIPELINE FOR A RETAIL COMPANY

Akanksha Sukre

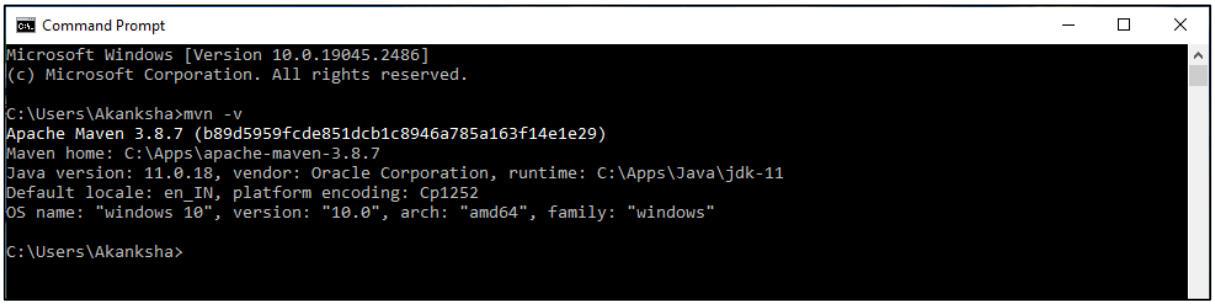
Date: 16/02/2023

Contents

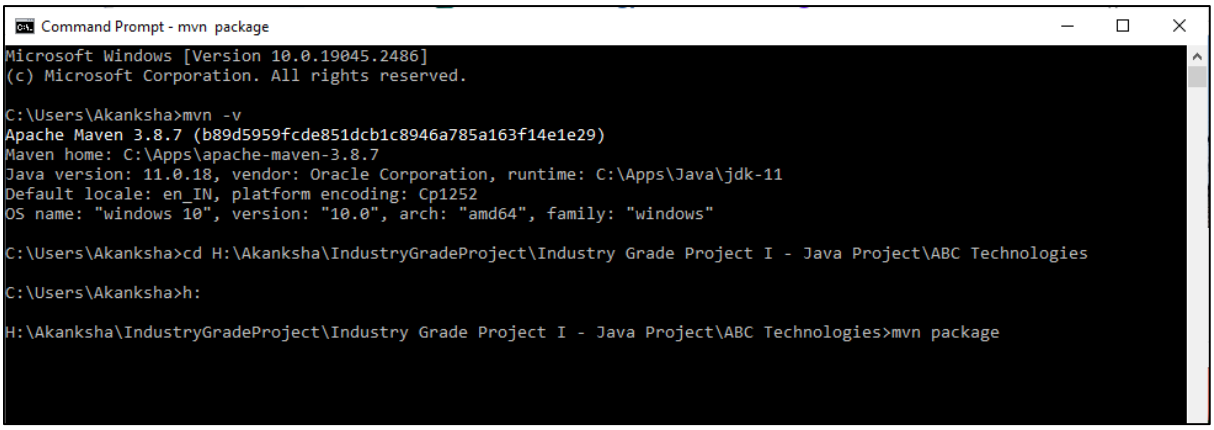
Task 1: Build the code using Maven commands

1) Install Maven and build code

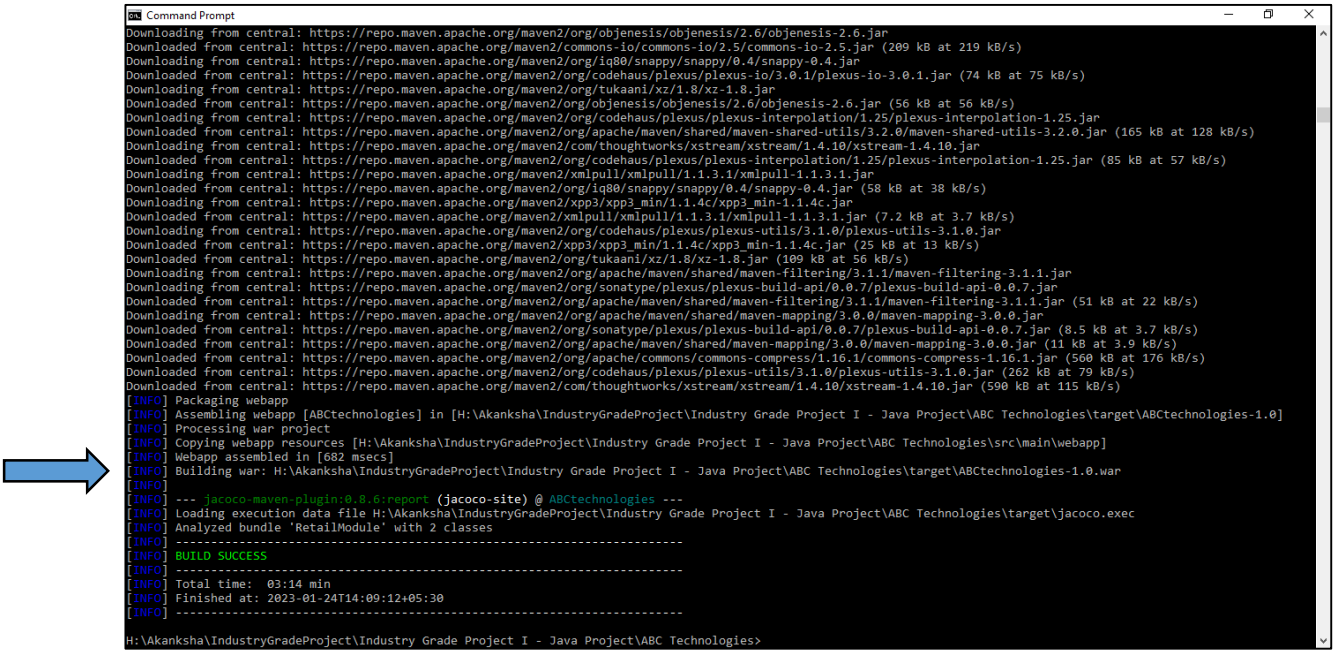
i. Install maven and check version

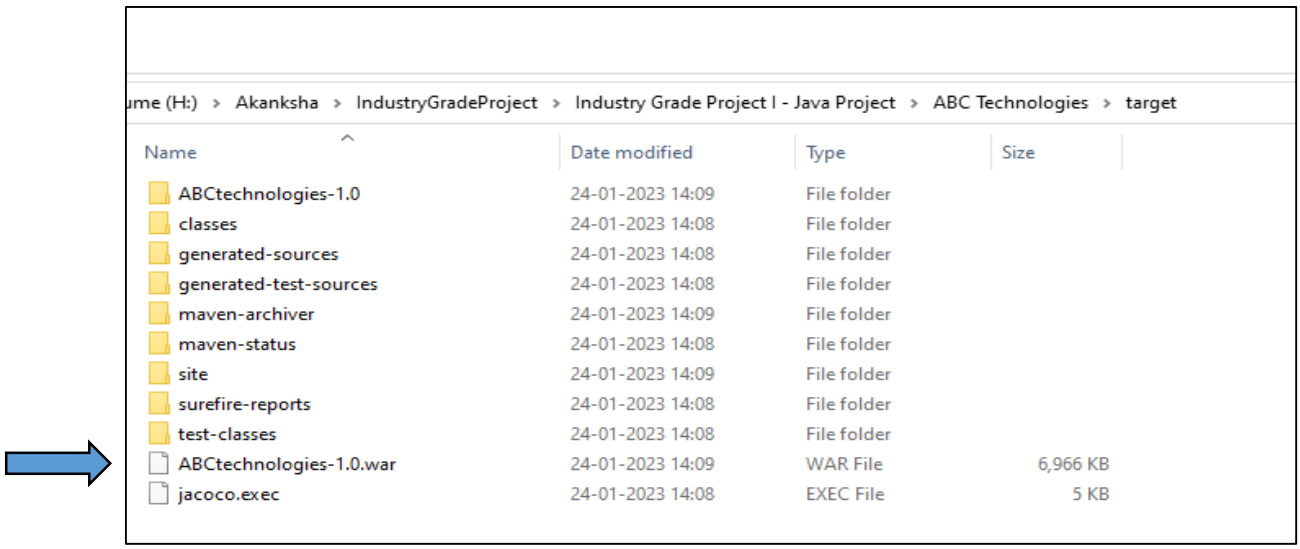


ii. Run command: mvn package



iii. Build successful and war file generated

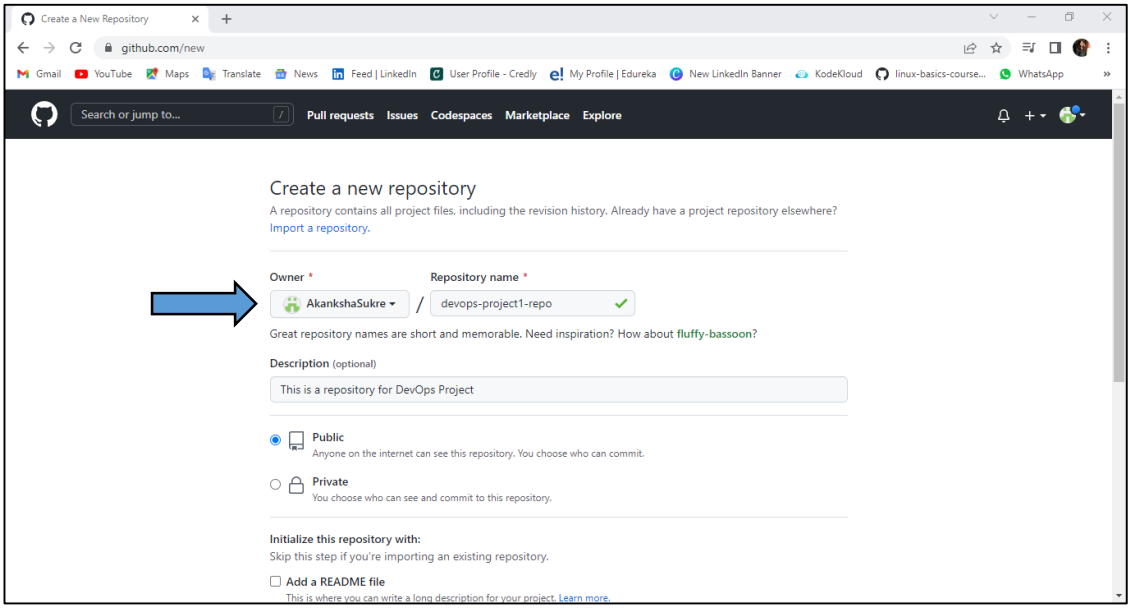




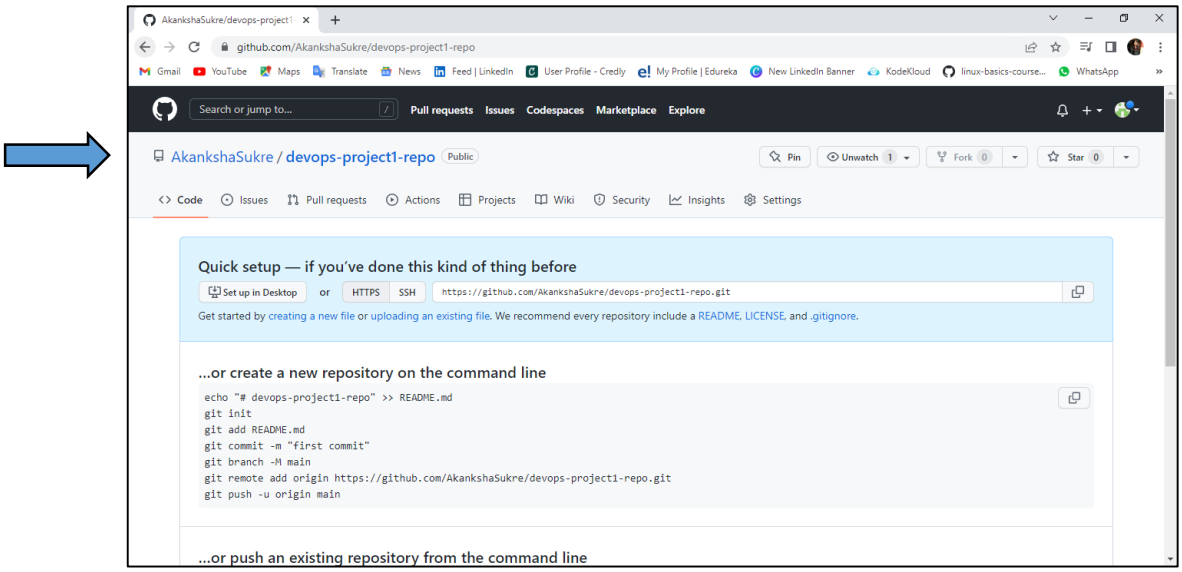
Task 2: Set up a GIT repository and push the source code

1) GitHub Repo Configuration

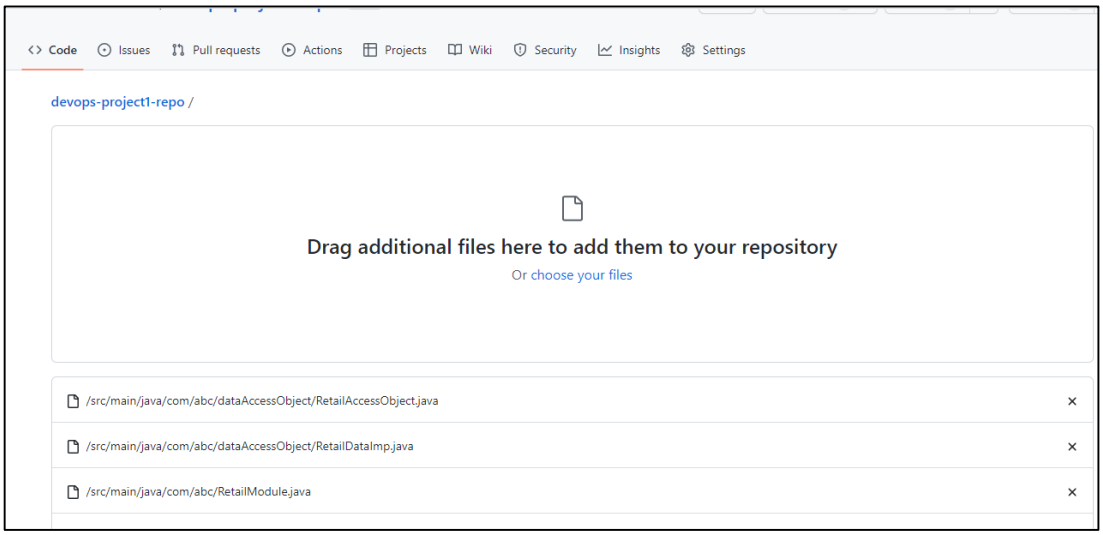
i. Create a new repository on GitHub

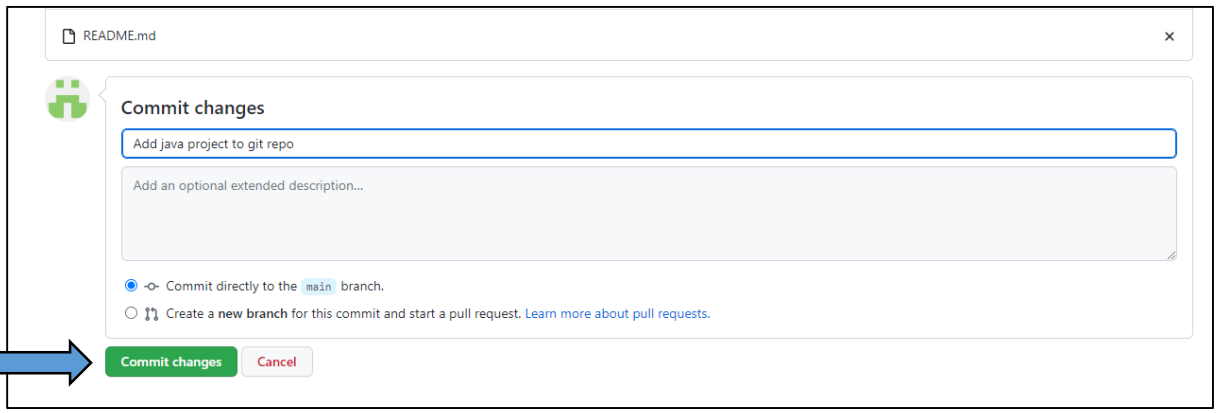


ii. New GIT repo created: **devops-project1-repo**

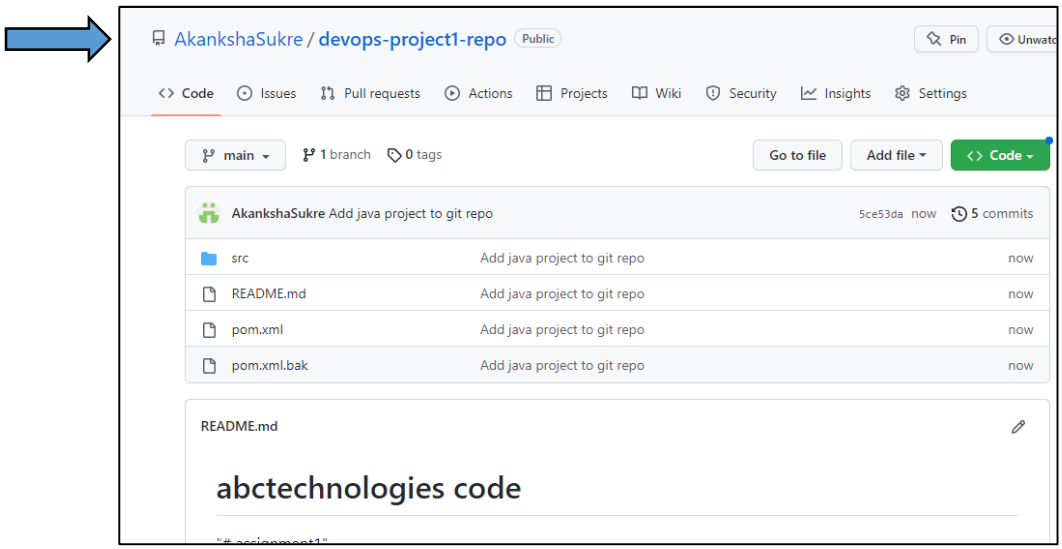


iii. Add project files to newly created repo and commit changes



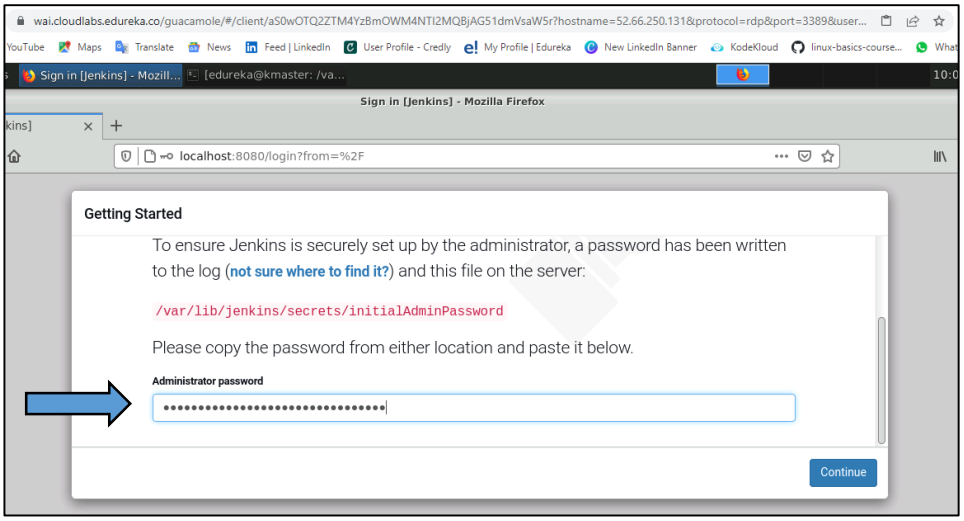


iv. Project successfully pushed to GIT repo

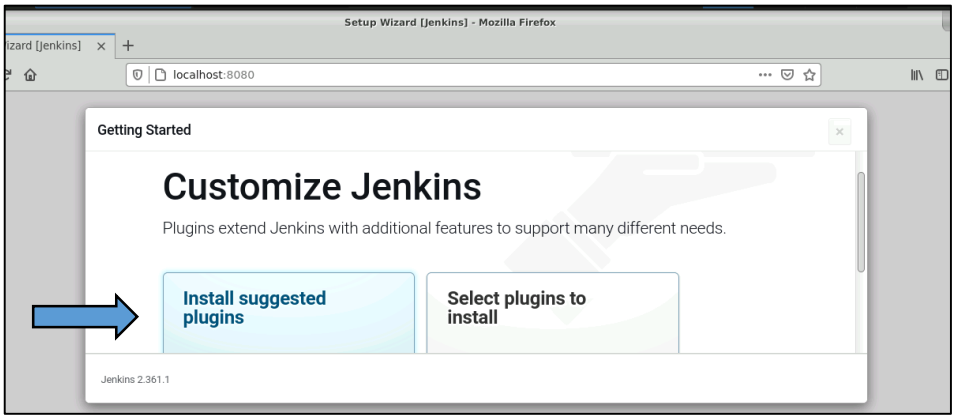


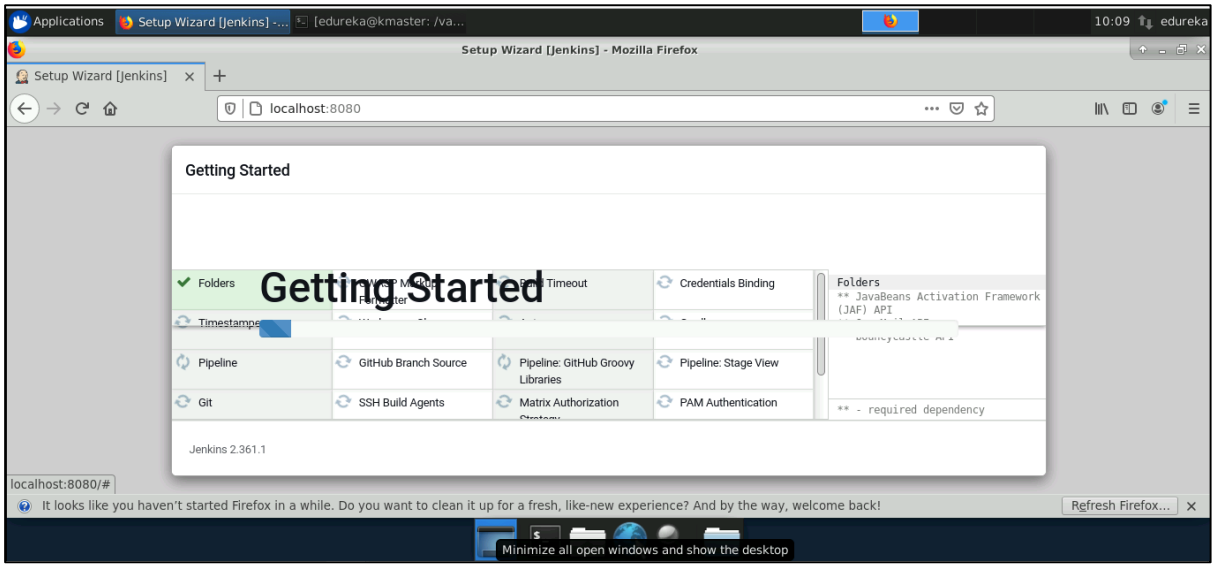
2) Jenkins Configuration on Master Node

i. Set up Jenkins on Master Node

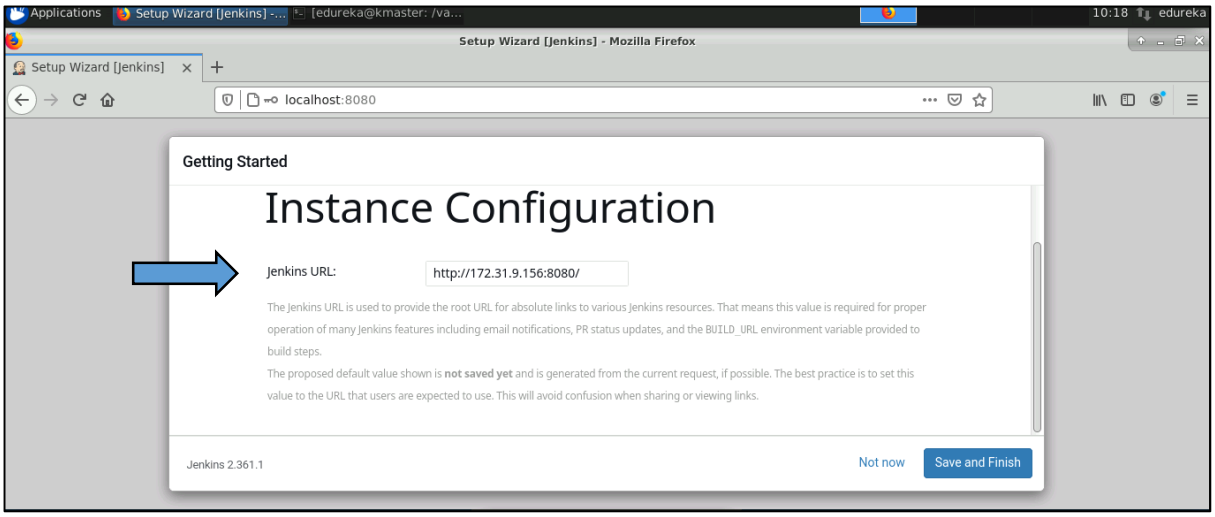


ii. Install required plugins in Jenkins

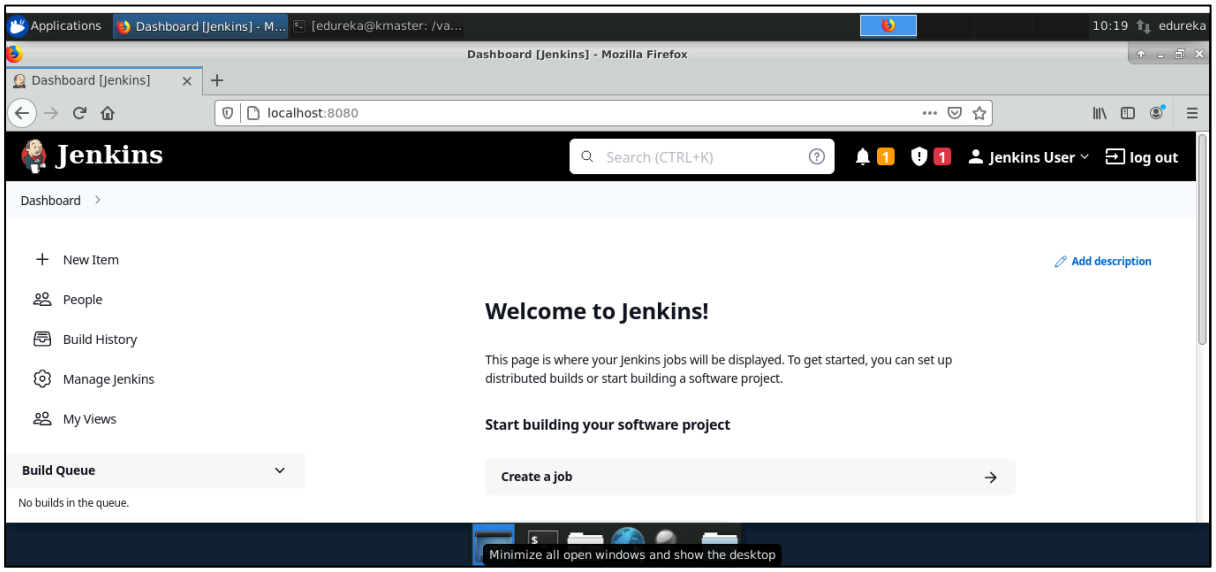




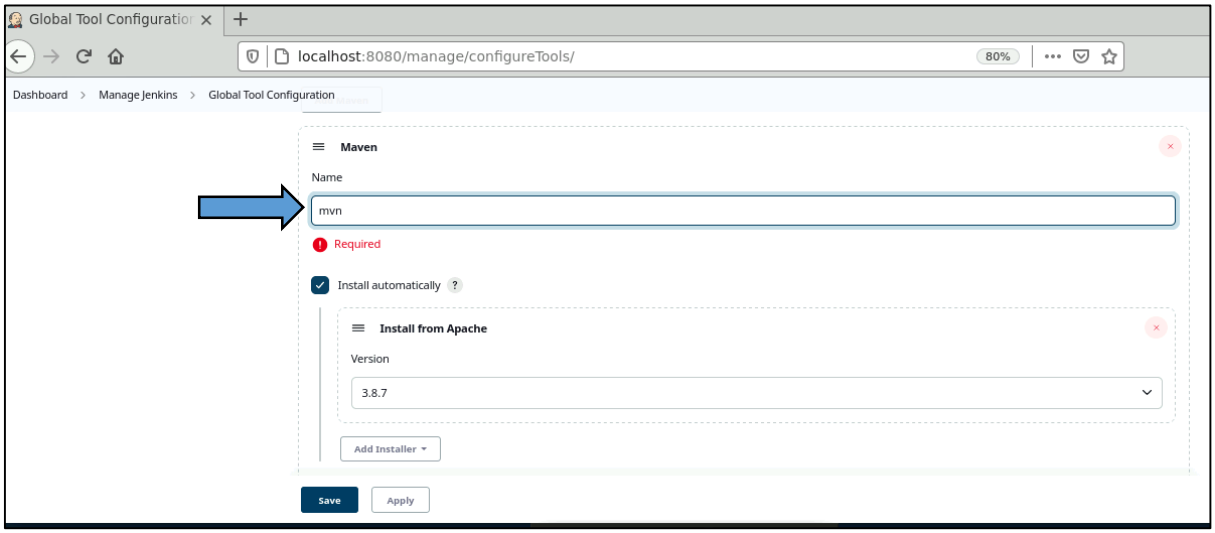
iii. Set up Jenkins URL: **http://172.31.9.156:8080**



iv. Jenkins is ready to use

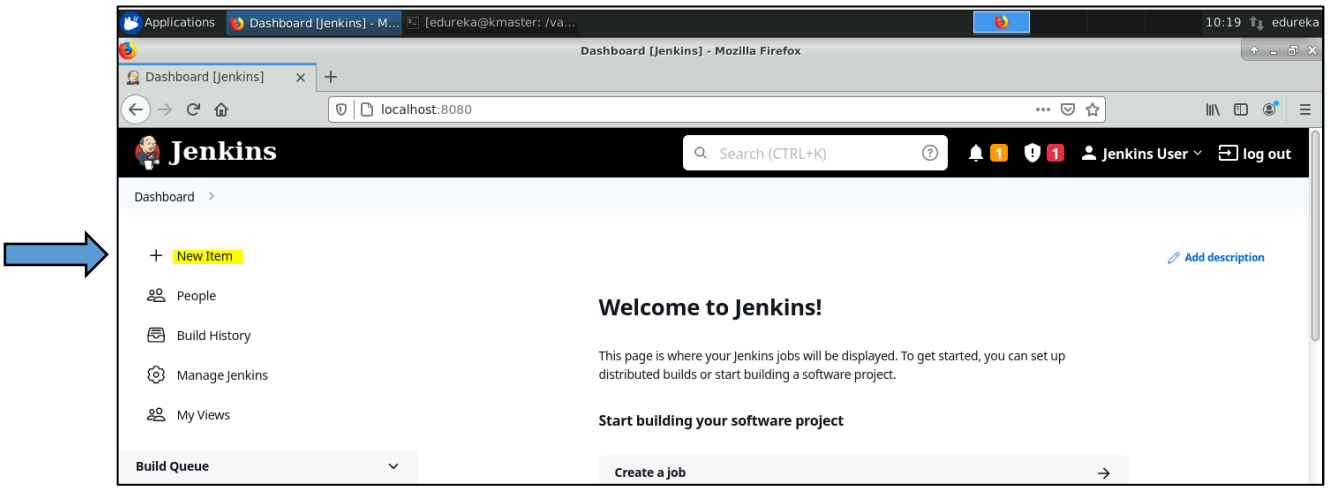


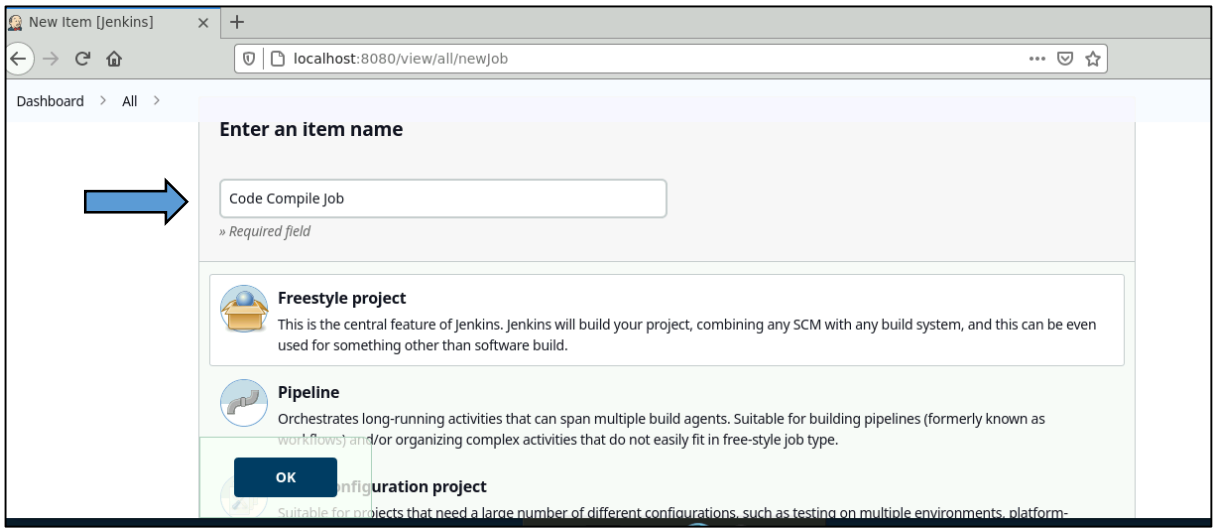
v. Add Maven to Global Tool Configuration



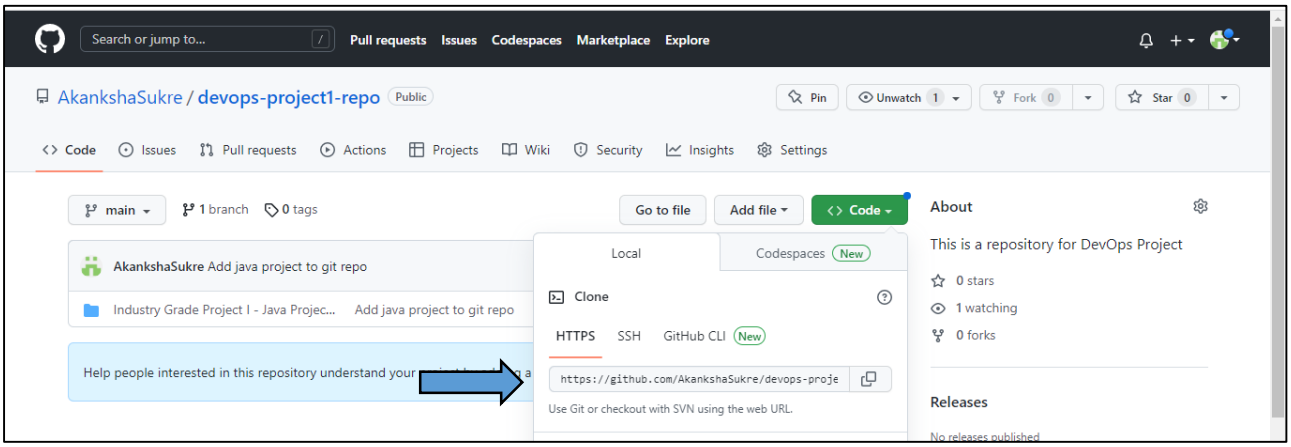
3) Create Jobs in Jenkins

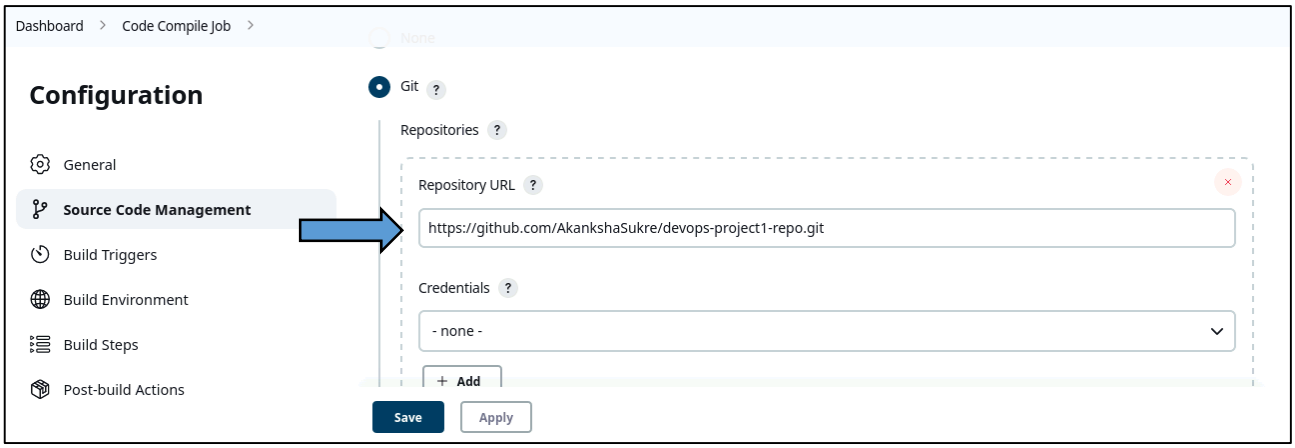
i. Create Compile Job



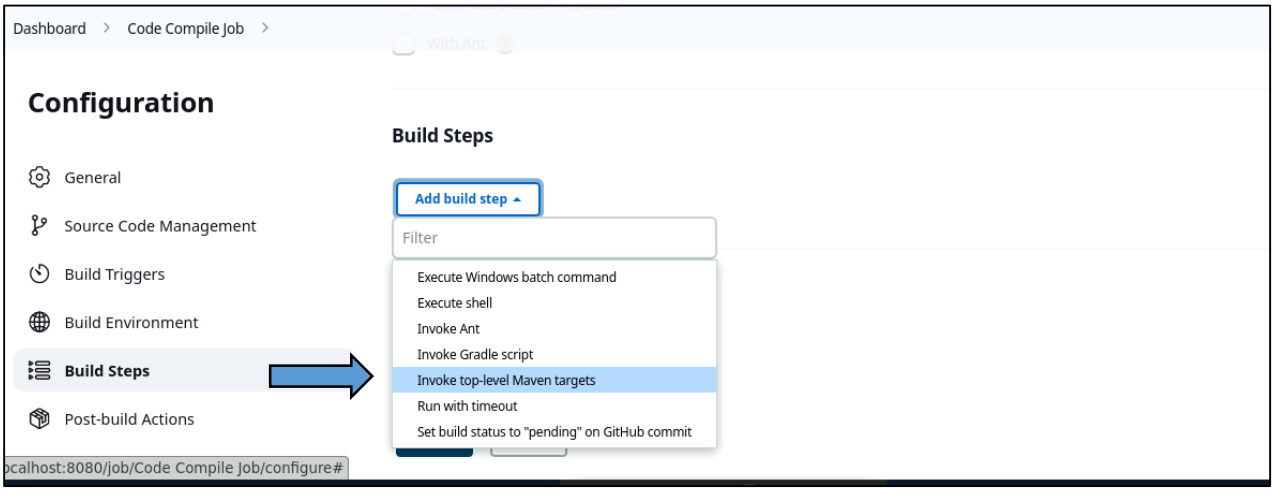


ii. Copy Git Repo URL for Configuring SCM in Compile Job

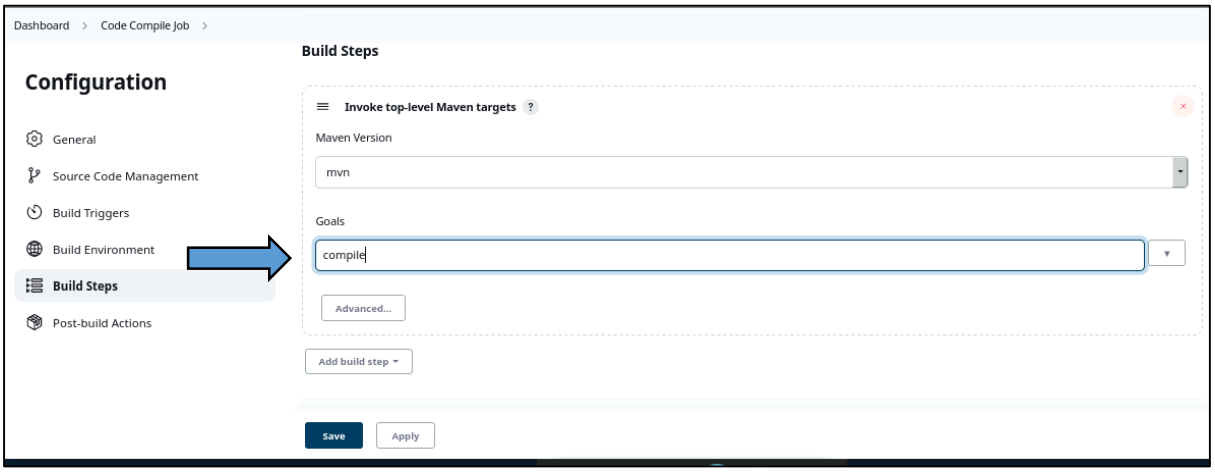




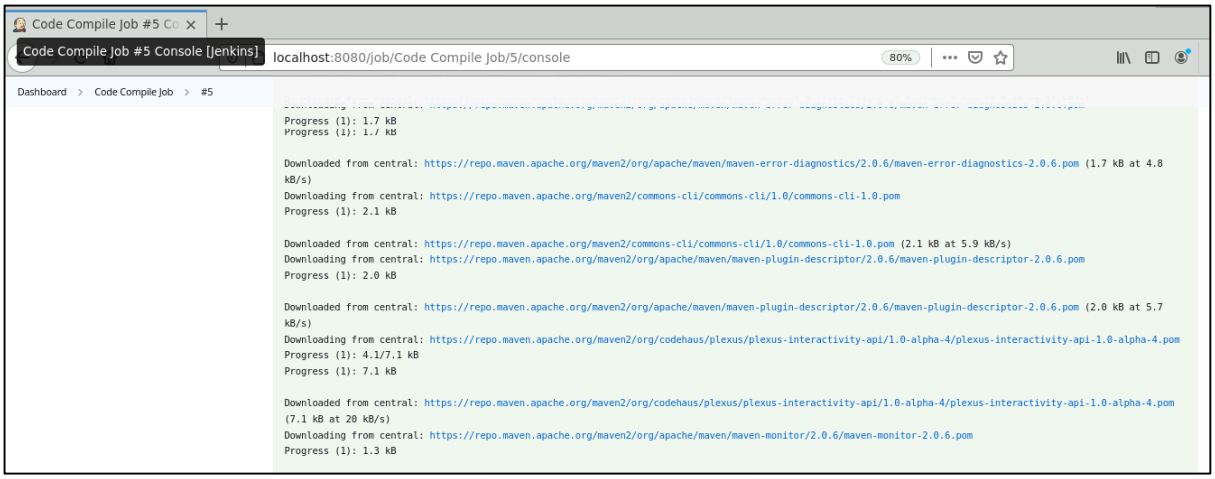
iii. Add Build step to invoke top-level Maven targets



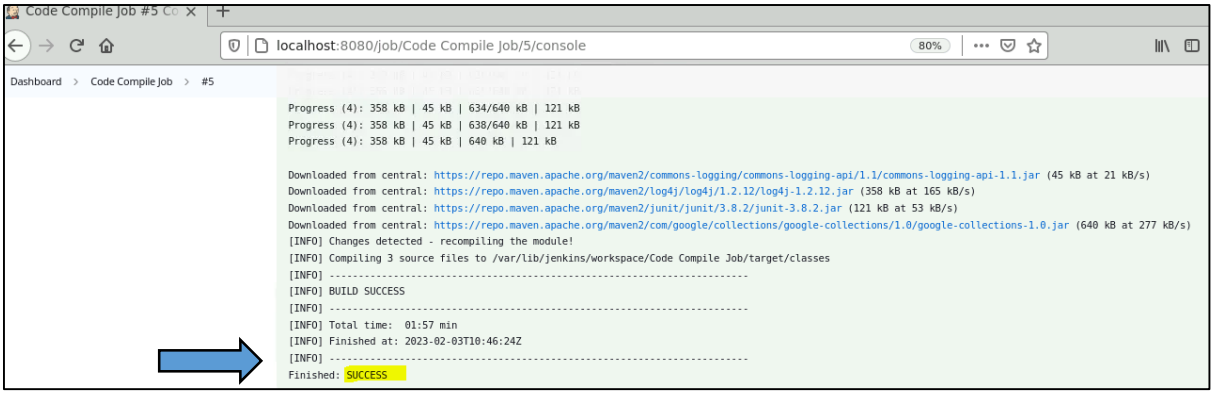
iv. Select Maven version and add goal : **compile**



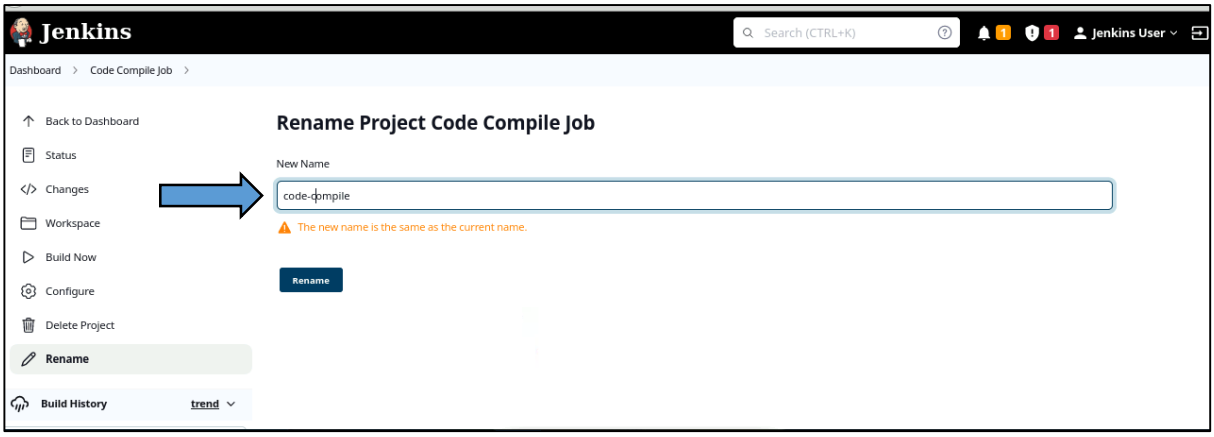
v. Build the compile job



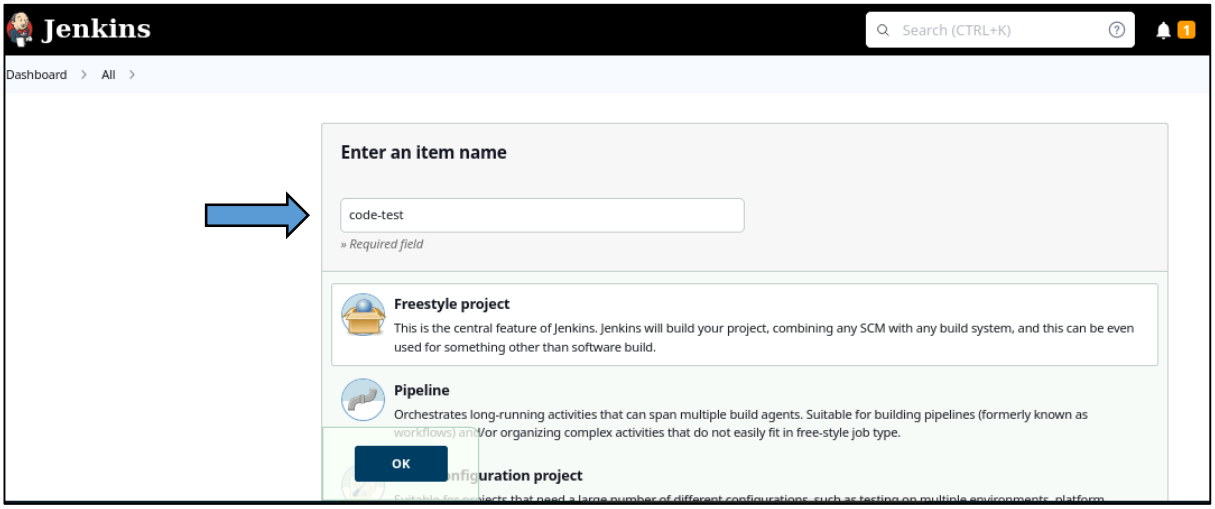
vi. Build successful



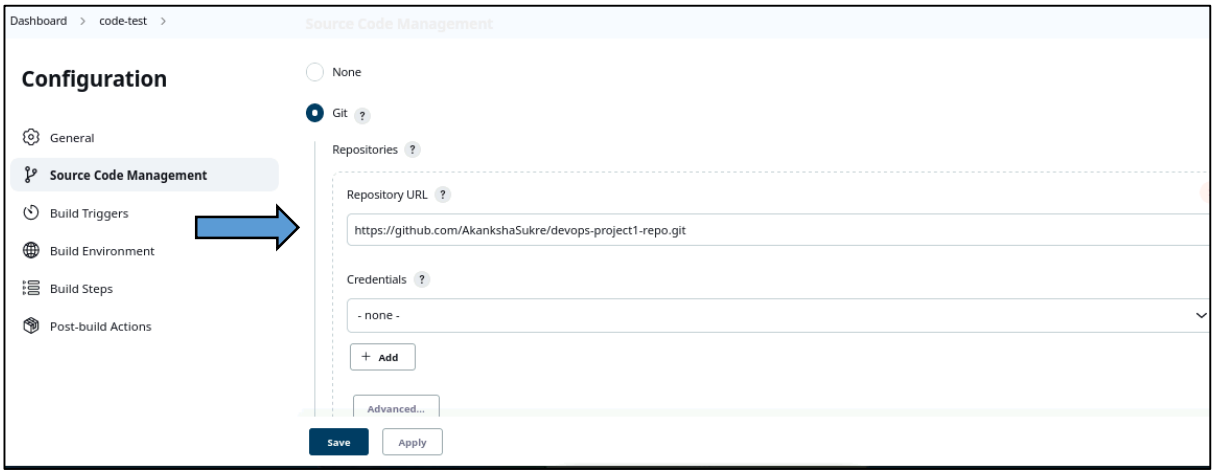
vii. Renamed project for maintaining consistency throughout



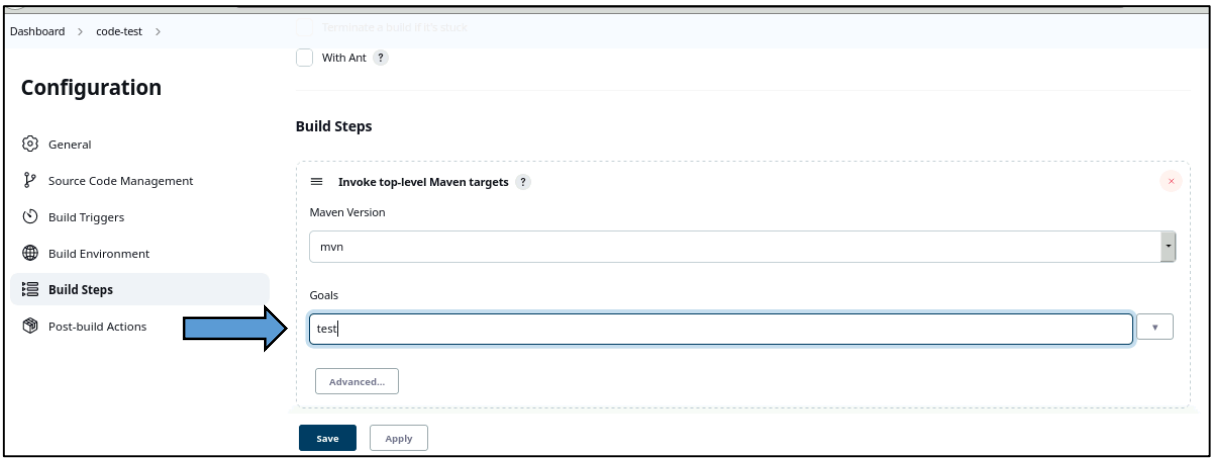
viii. Create test job



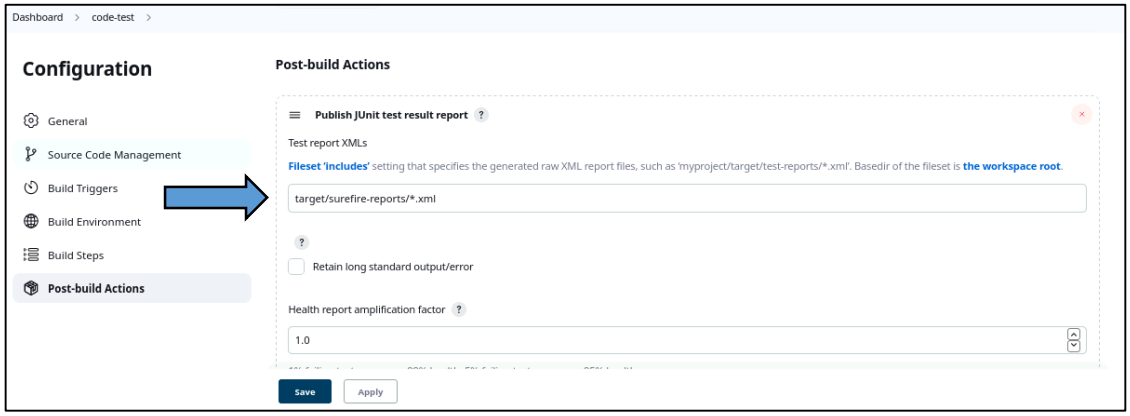
ix. Configure SCM with GIT Repo URL



x. Add build step as top-level maven targets with goal : **test**

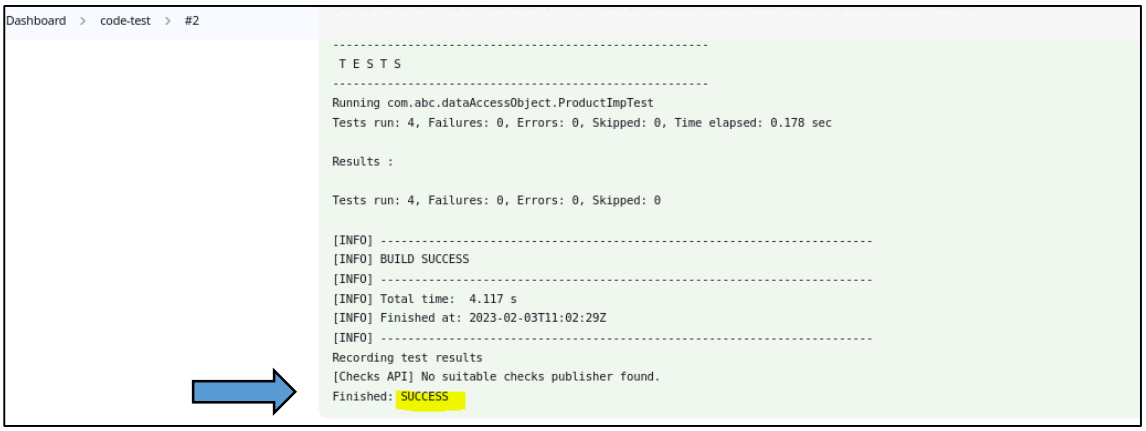


xi. Add Post build action to publish JUnit test reports

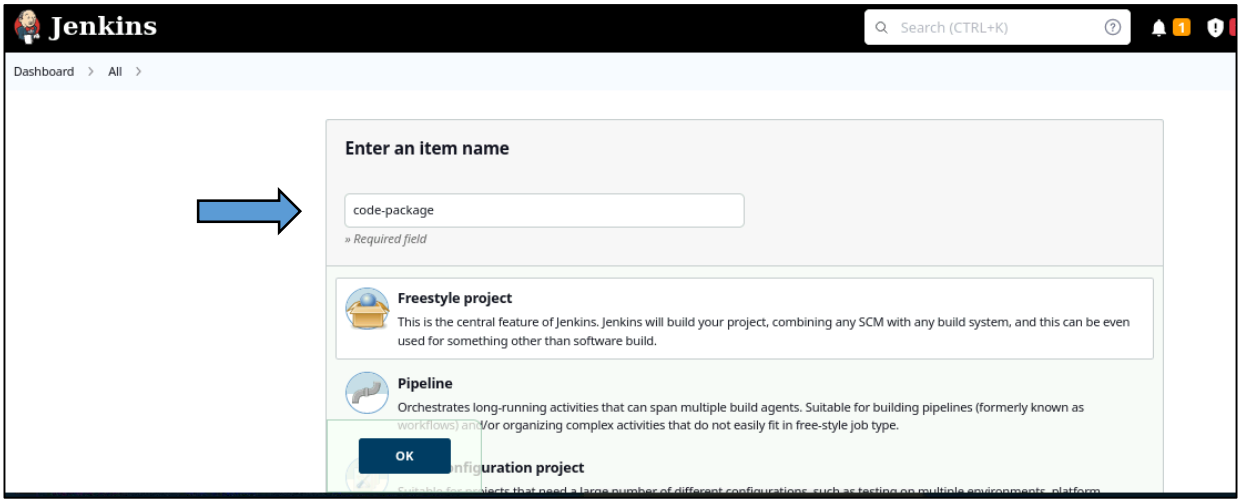


xii. Build the test Job

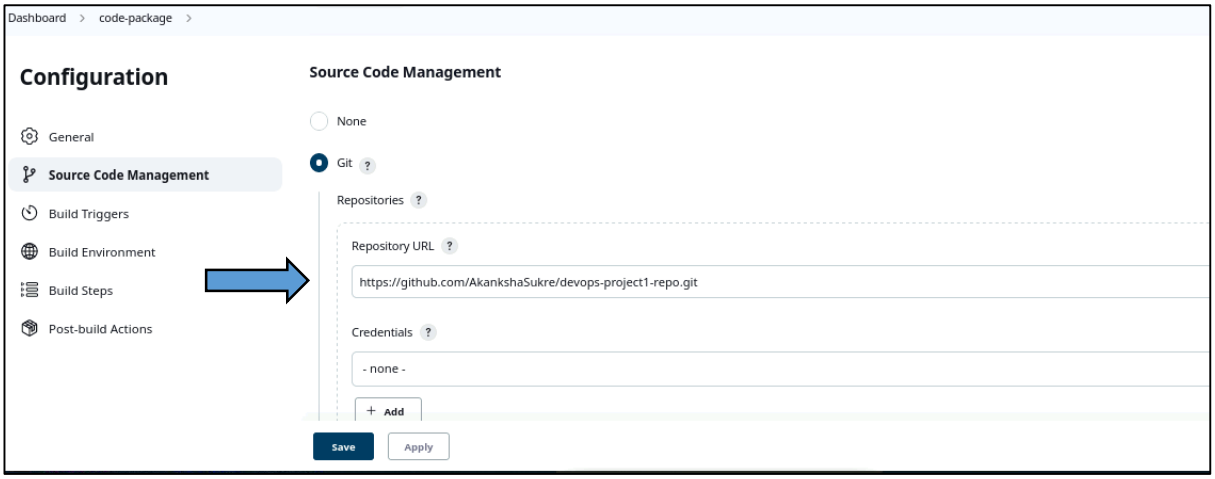
xiii. Build successful



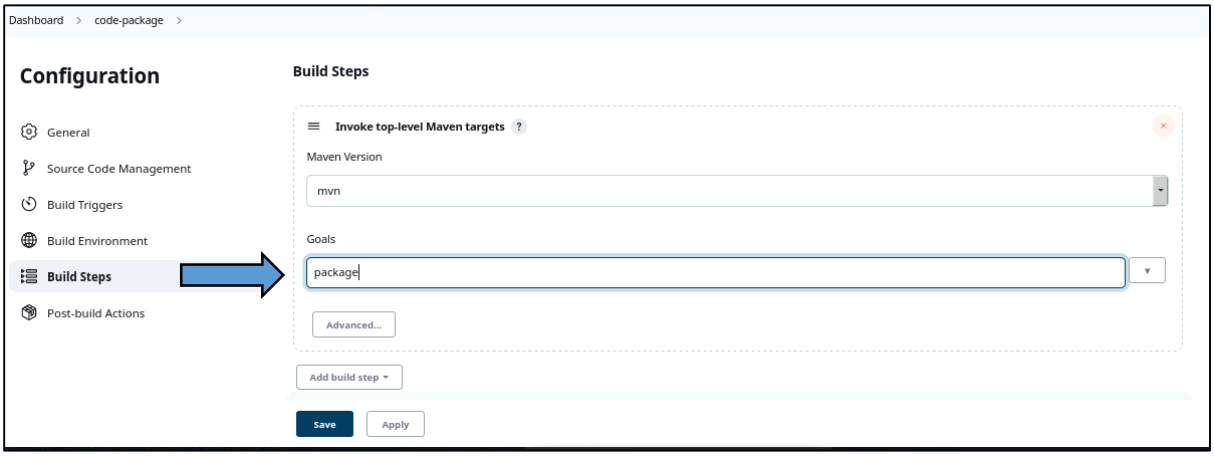
xiv. Create package job



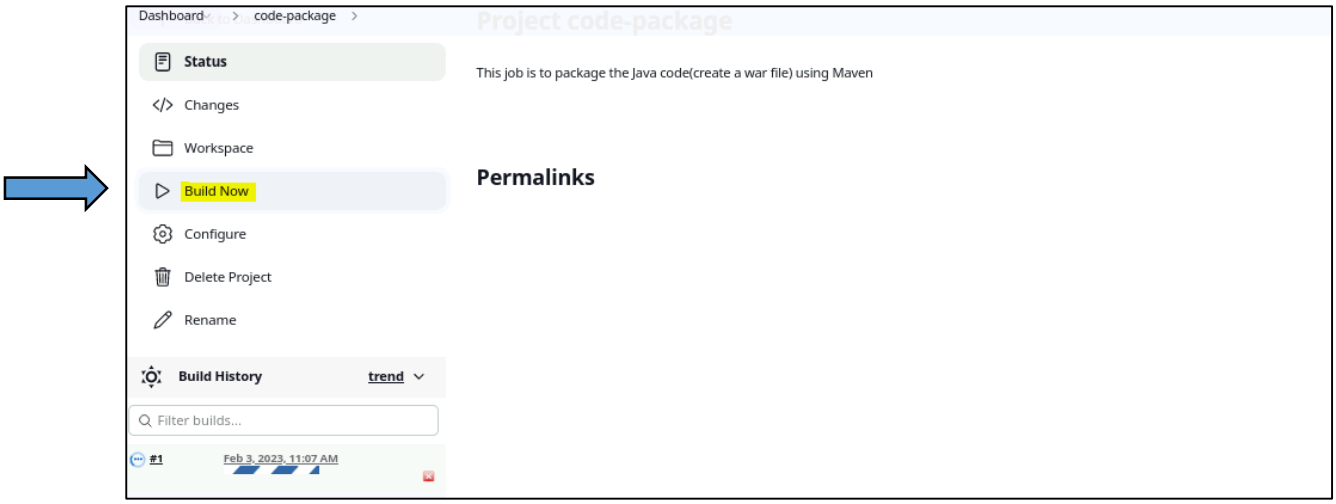
xv. Configure SCM with GIT Repo URL



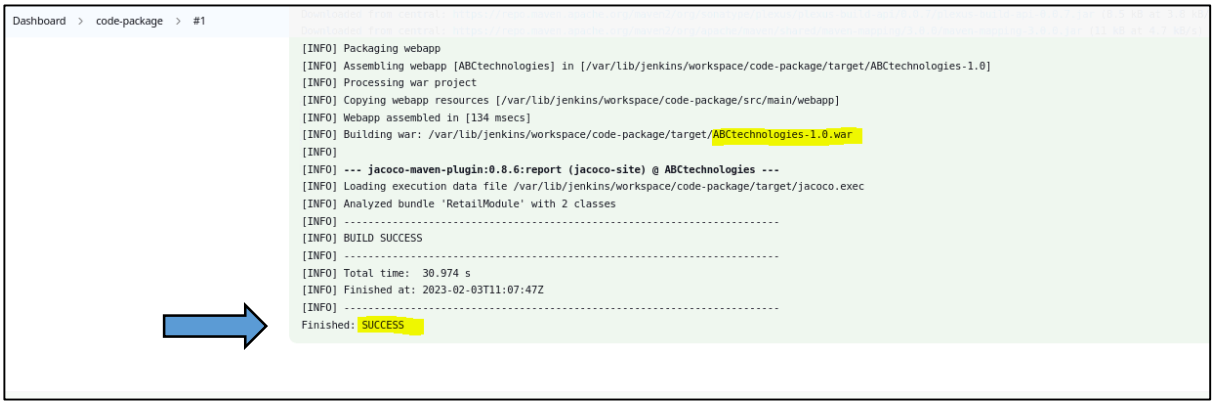
xvi. Add build step as top-level maven target with goal: **package**



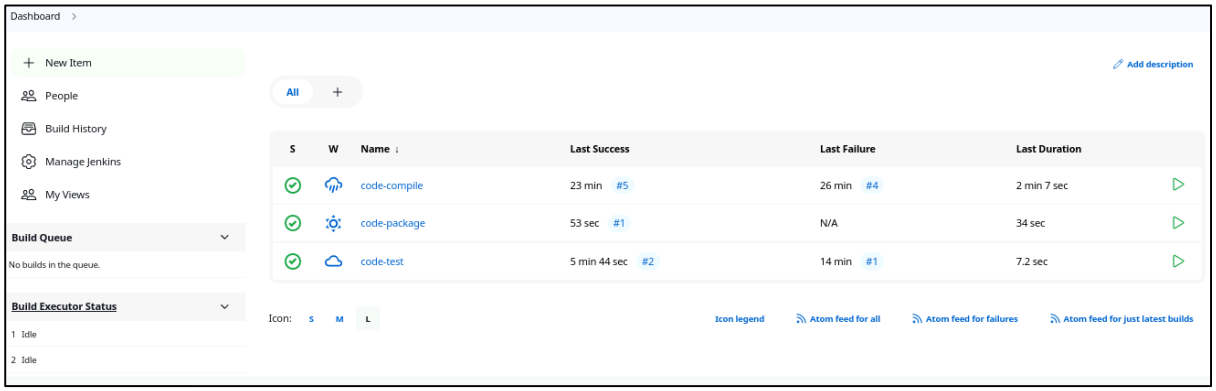
xvii. Build package Job



xviii. Build successful and war file generated



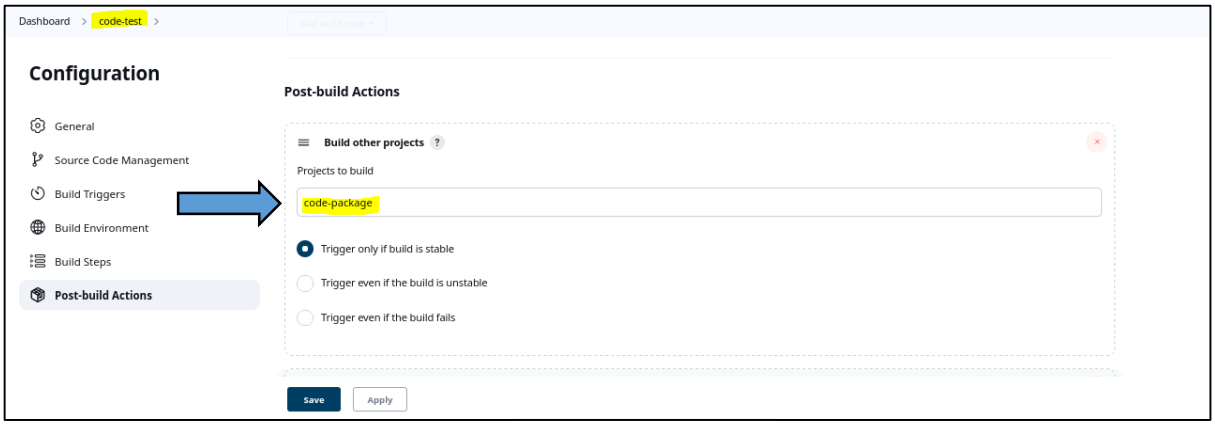
xix. Three jobs ready : compile,test and package



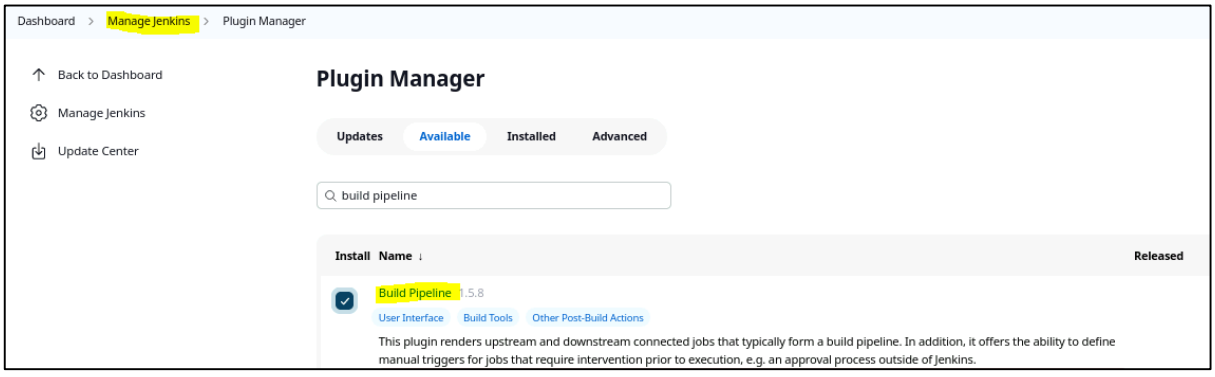
4) Create build pipeline with above three jobs

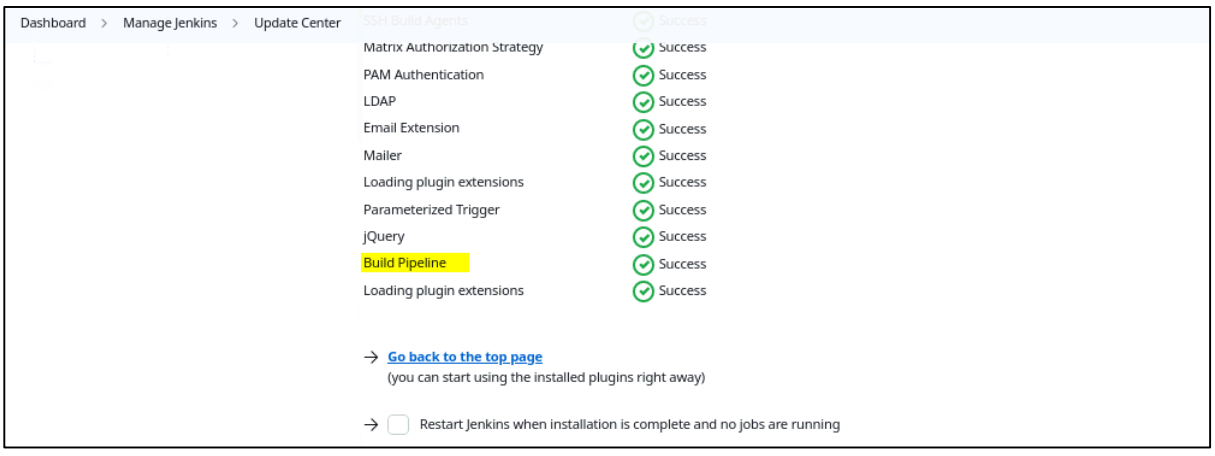
i. Add post-build action in Compile job to run Test job

ii. Add post-build action in Test job to build Package job

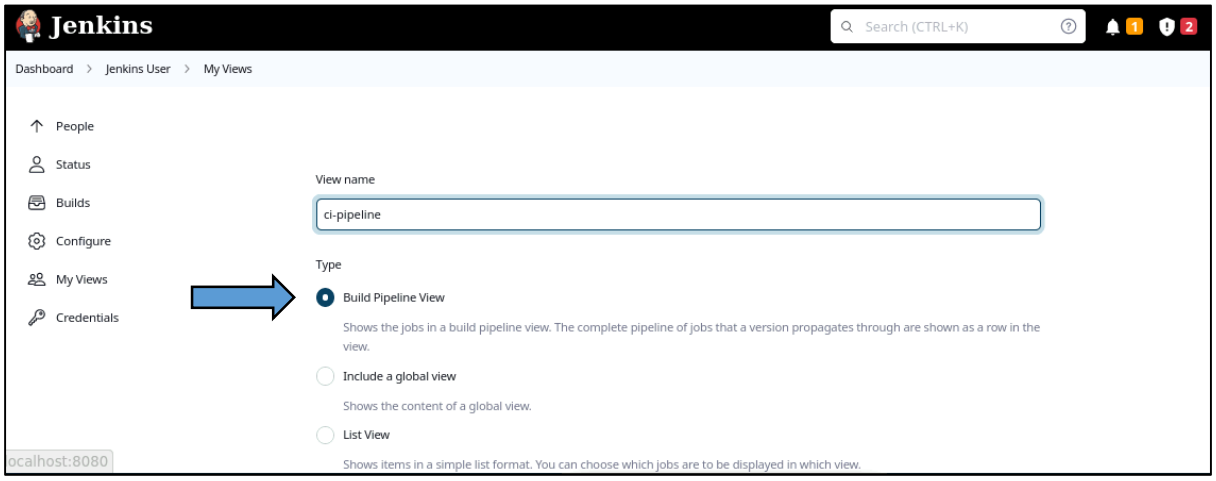


iii. Install Build pipeline plugin

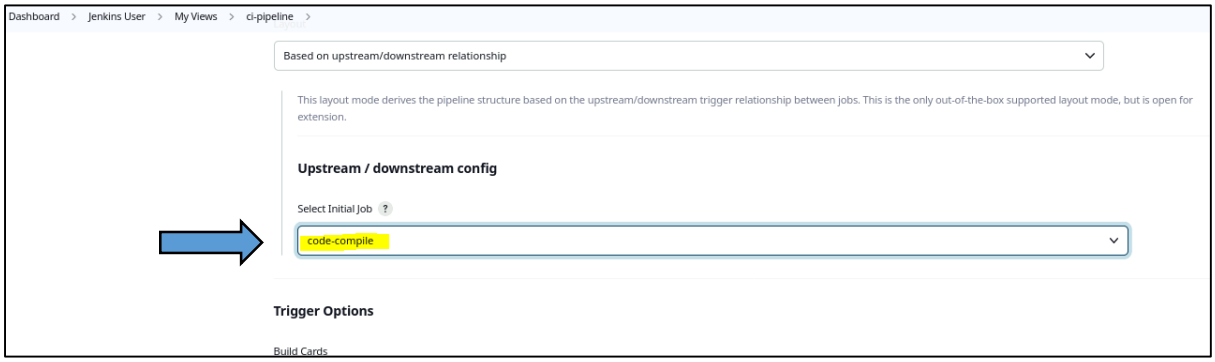




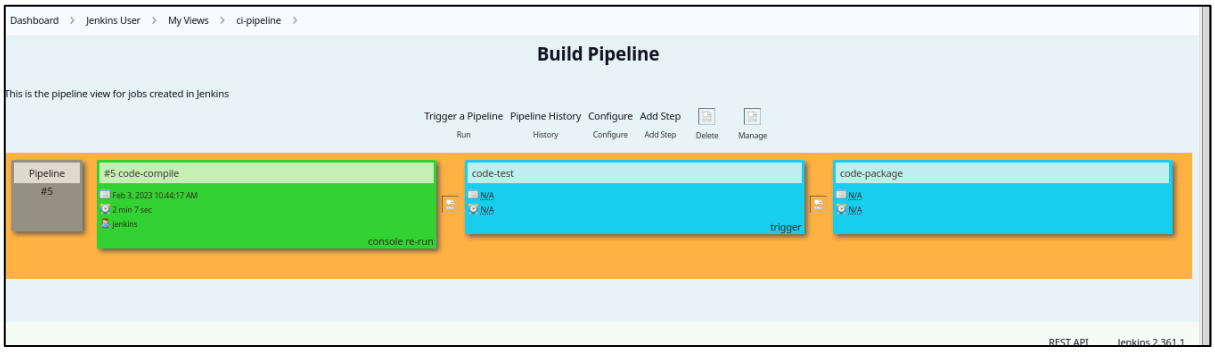
iv. Create a new view of type: **Build Pipeline View**

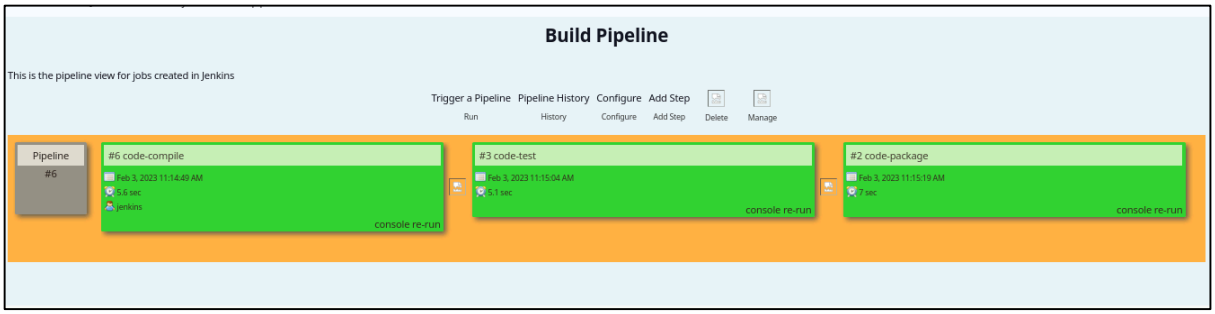


v. Set up initial job to run as Compile Job



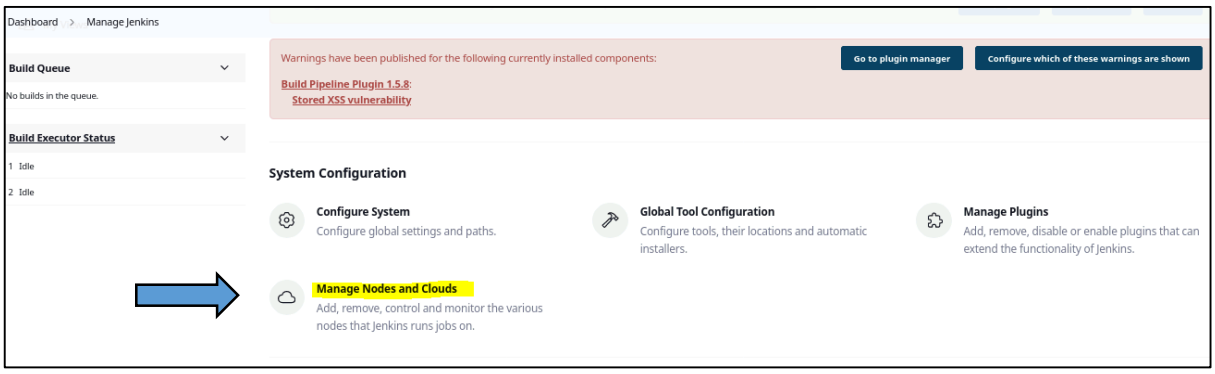
vi. Build pipeline is ready



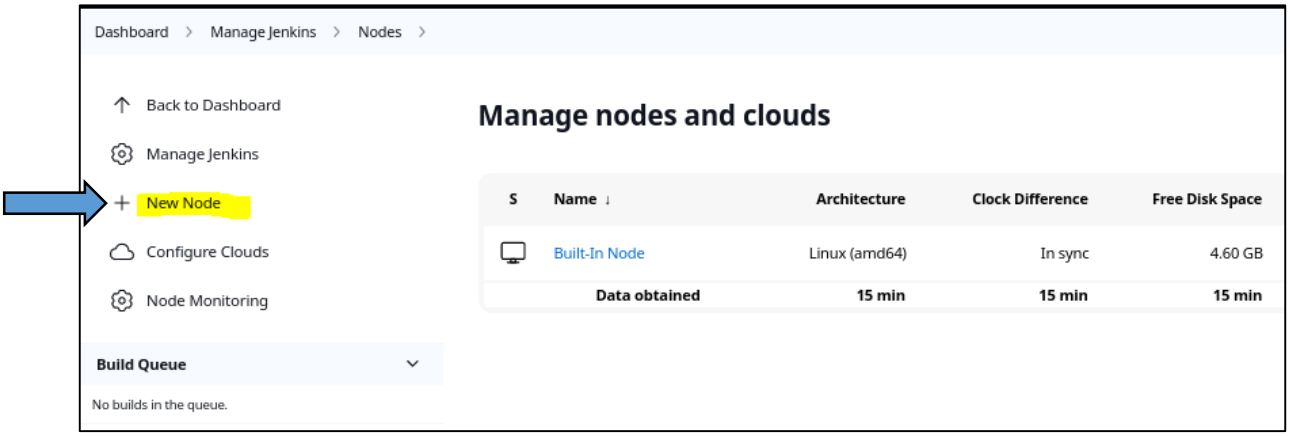


5) Set up master-slave nodes to distribute tasks in the pipeline

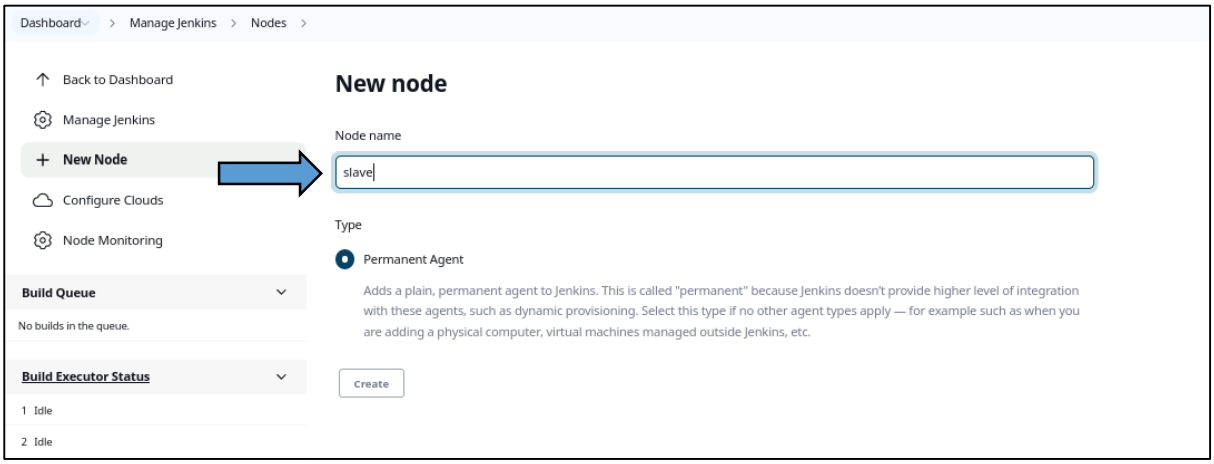
i. Go to “Manage Jenkins” and Click on “Manage Nodes and Clouds”



ii. Click on “New Node”

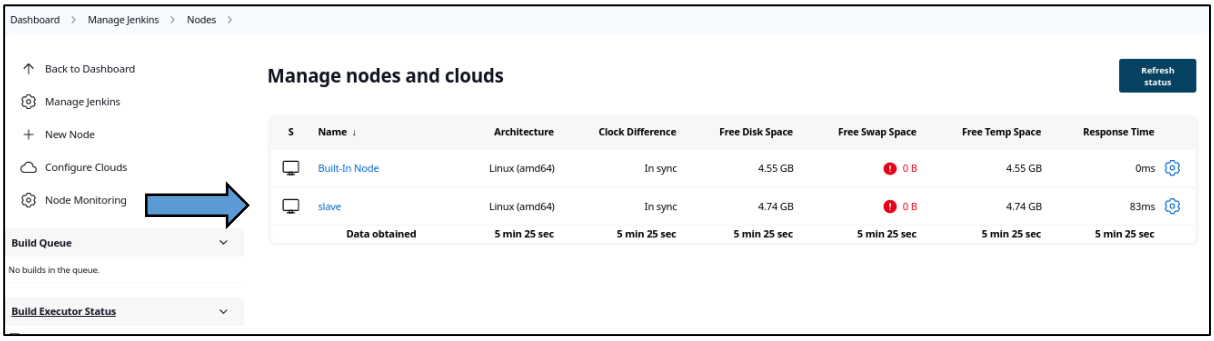


iii. Create a node named “slave”

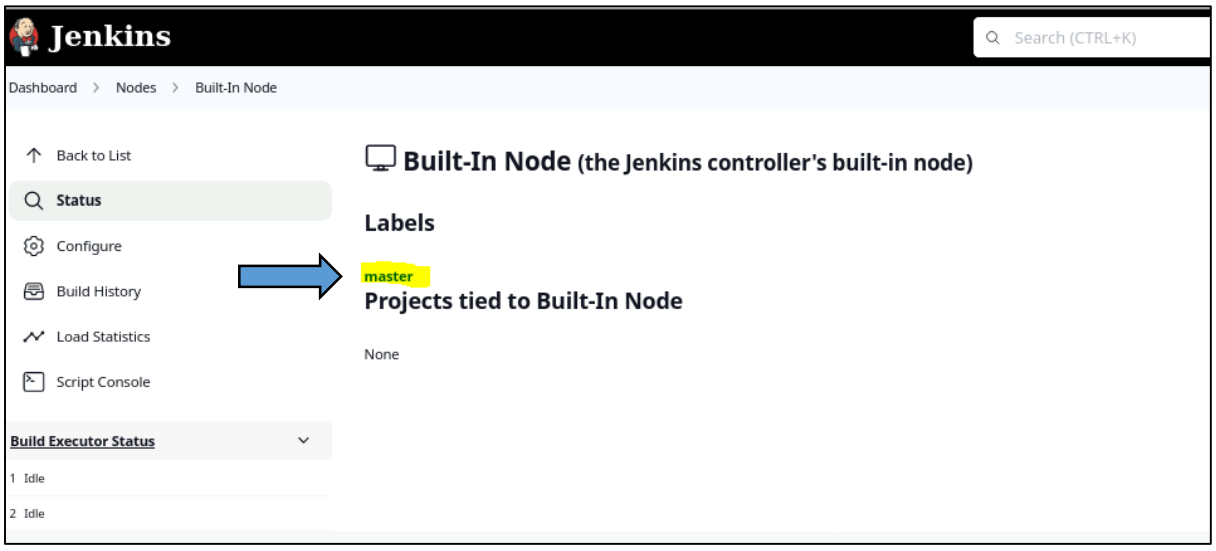


iv. Add slave node details as shown below

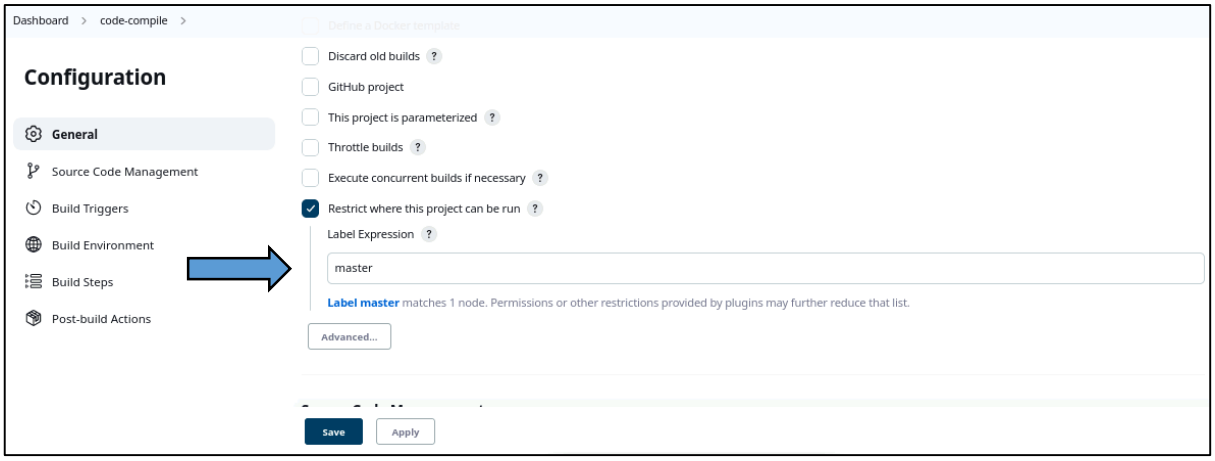
v. Node is added successfully



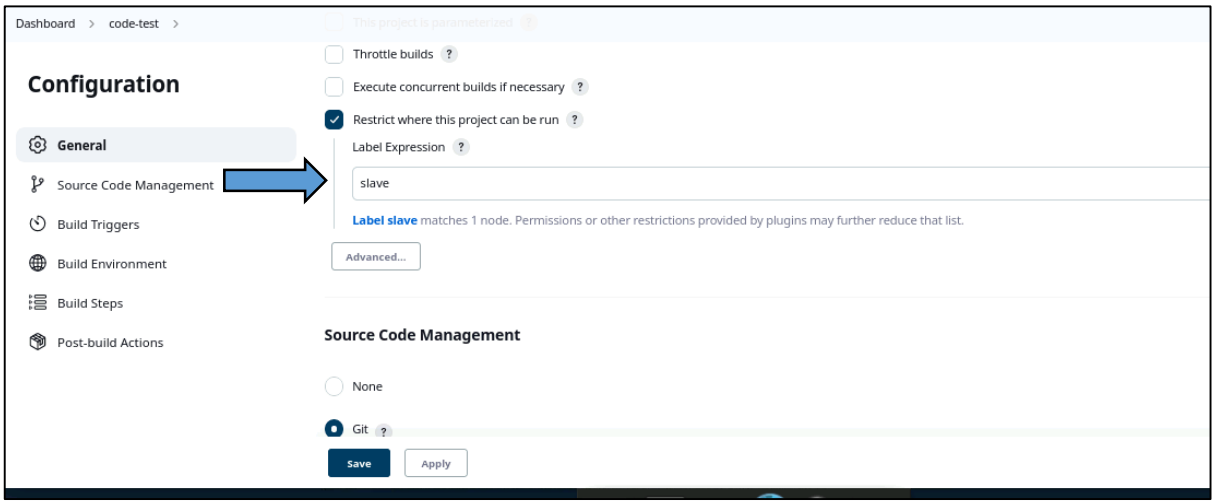
vi. Add label to Built-In Node as “master”

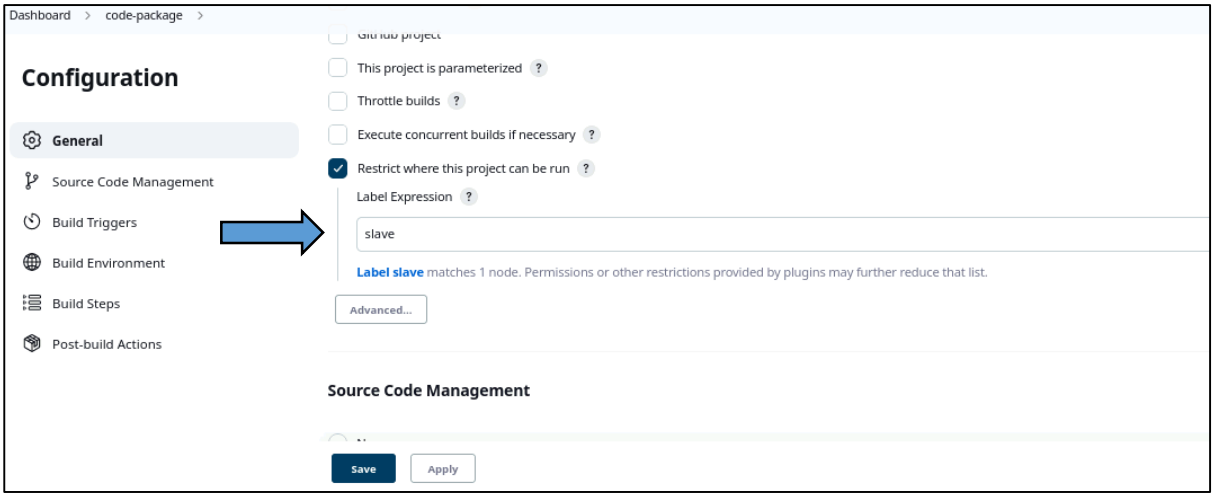


vii. Configure “compile job” to run on master node



viii. Configure “test job” and “package” job to run on slave node

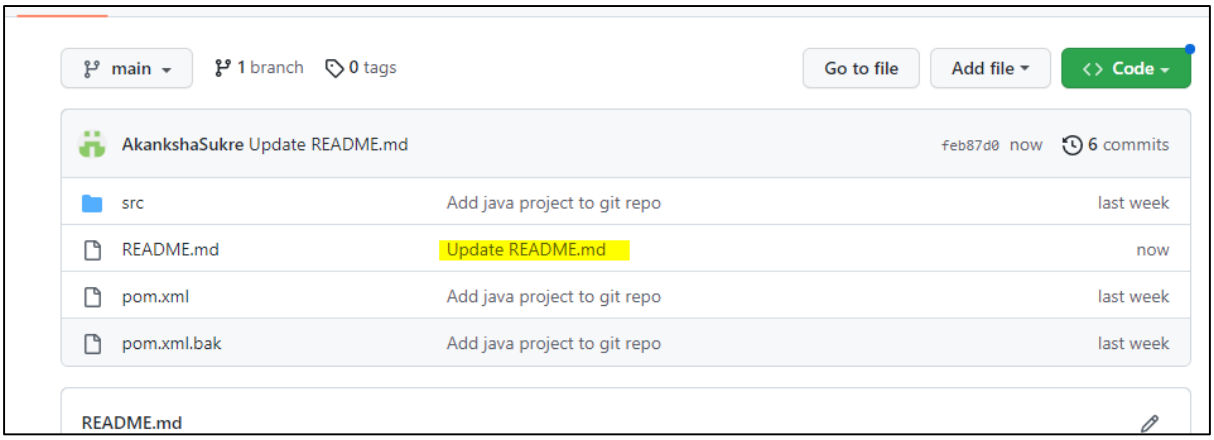




6) GITScm Polling for CI

i. Add build trigger in compile job to Poll SCM. Configure a schedule to check for changes in repo every minute

ii. Test by updating Readme file in Repo



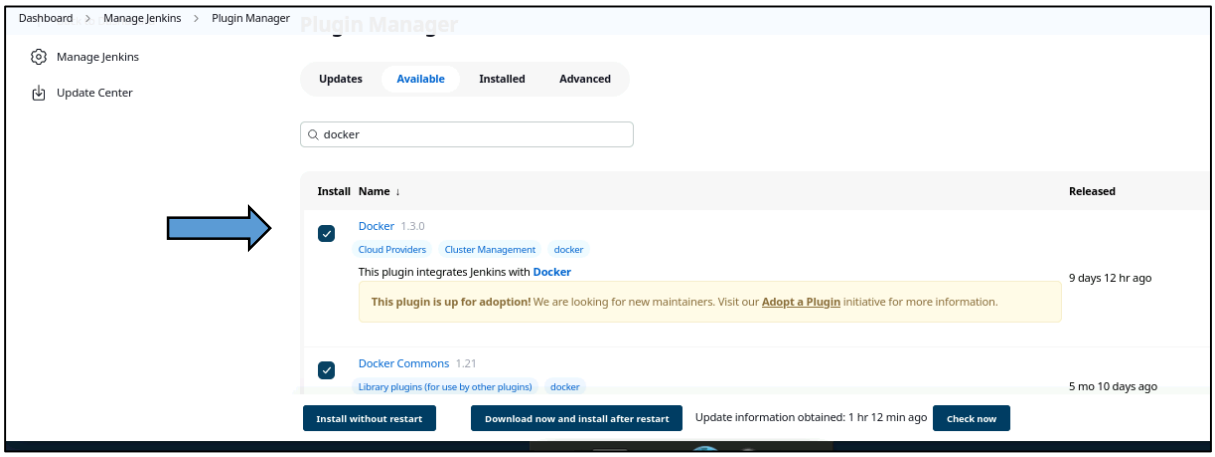
iii. Check Git Polling Log

iv. Pipeline executed successfully

Task 3: Create CI/CD pipeline in Jenkins to build and deploy on Docker container

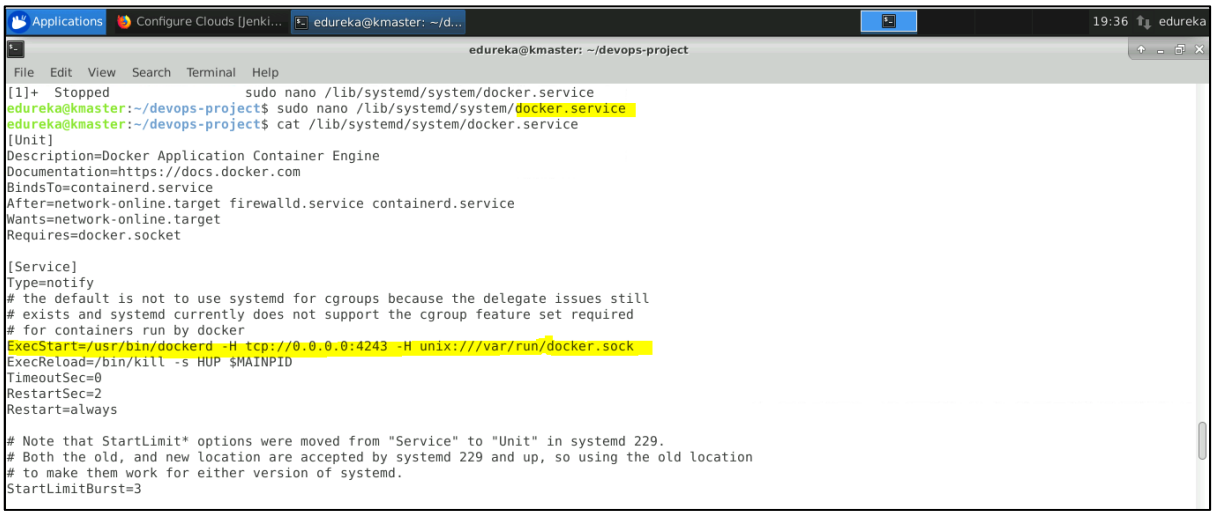
1) Integrate Docker with Jenkins

i. Install Docker plugin in Jenkins



ii. Go to Manage Jenkins → Configure Clouds → Add a new cloud **Docker**

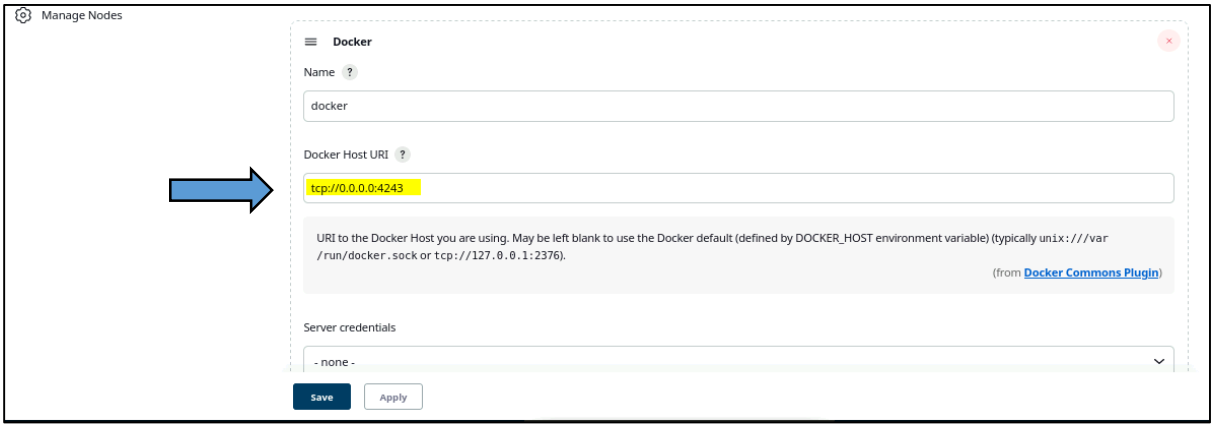
iii. Make the highlighted changes in docker service file on Docker host(Jenkins master node)



iv. Restart the docker service

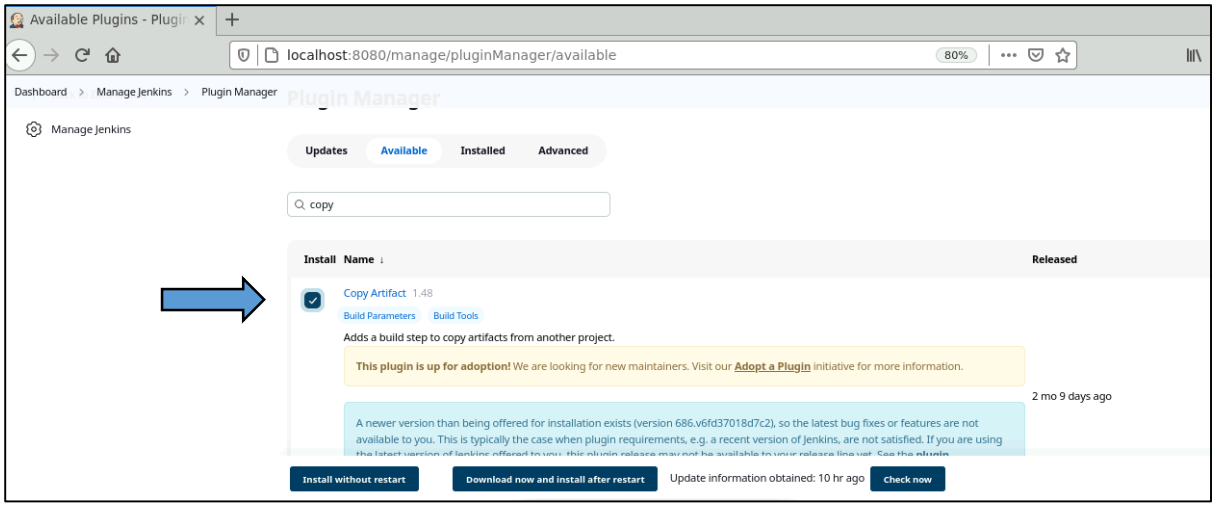
```
edureka@kmaster:~/devops-project$ sudo systemctl daemon-reload
edureka@kmaster:~/devops-project$ sudo service docker restart
edureka@kmaster:~/devops-project$ curl http://localhost:4243/version
{"Platform":{"Name":""},"Components":[{"Name":"Engine","Version":"19.03.6","Details":{"ApiVersion":"1.40","Arch":"amd64","BuildTime":"2020-02-19T01:
6:16.000000000+00:00","Experimental":"false","GitCommit":"369ce74a3c","GoVersion":"go1.12.17","KernelVersion":"4.15.0-1021-aws","MinAPIVersion":"1.1
","Os":"linux"}},{"Name":"containerd","Version":"1.3.3-0ubuntu1~18.04.2","Details":{"GitCommit":""}},{"Name":"runc","Version":"spec: 1.0.1-dev","Det
ils":{"GitCommit":""}},{"Name":"docker-init","Version":"0.18.0","Details":{"GitCommit":""}}],"Version":"19.03.6","ApiVersion":"1.40","MinAPIVersion"
"1.12","GitCommit":"369ce74a3c","GoVersion":"go1.12.17","Os":"linux","Arch":"amd64","KernelVersion":"4.15.0-1021-aws","BuildTime":"2020-02-19T01:06:
6.000000000+00:00"}
edureka@kmaster:~/devops-project$
```

v. Add the Docker Host URI in Docker cloud properties in Jenkins and save the configuration

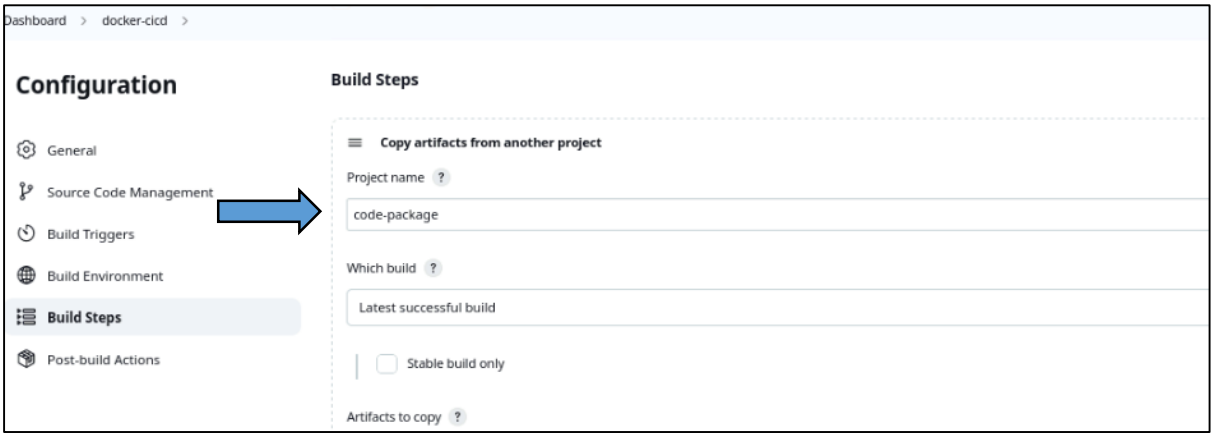


2) CI/CD job on Jenkins to build and deploy on Docker container

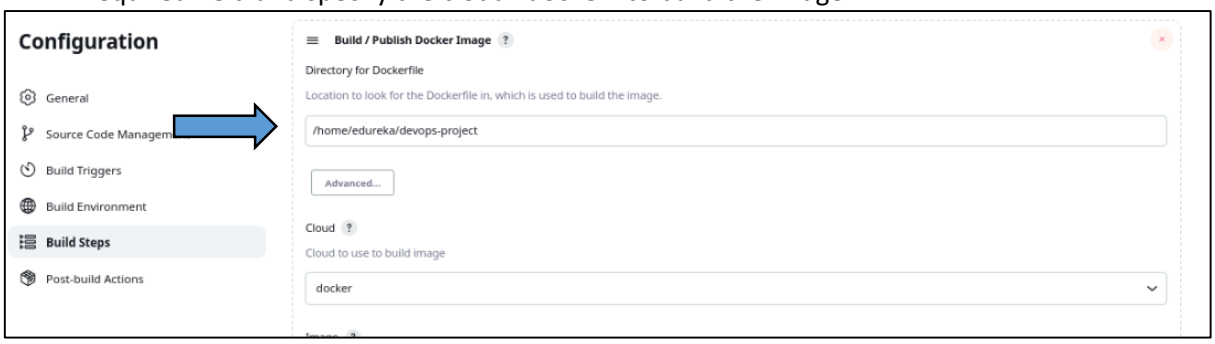
i. Install the Copy Artifact plugin in Jenkins



ii. Create a new Jenkins job: **docker-cicd** and add copy artifact as the first build step

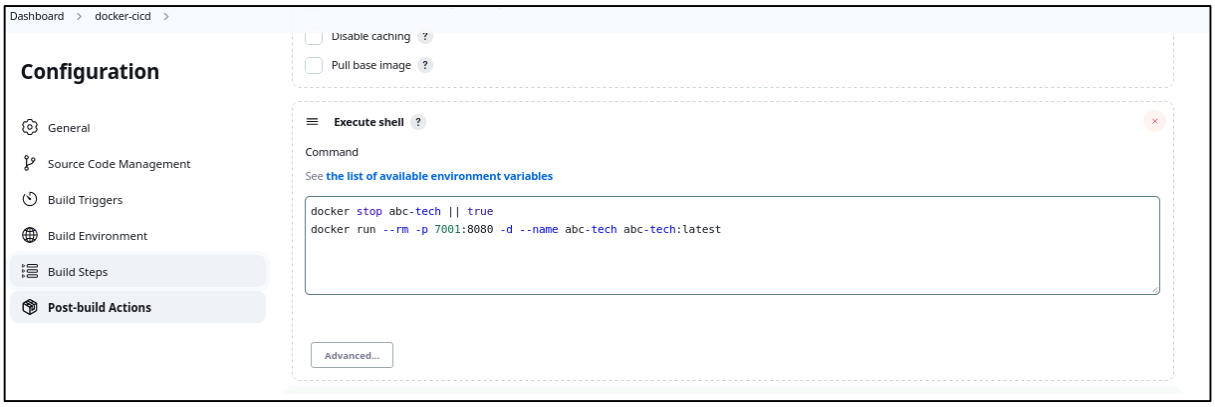


iii. Add second build step as Build/Publish Docker Image. Give Dockerfile path in the required field and specify the cloud "docker" to build the image

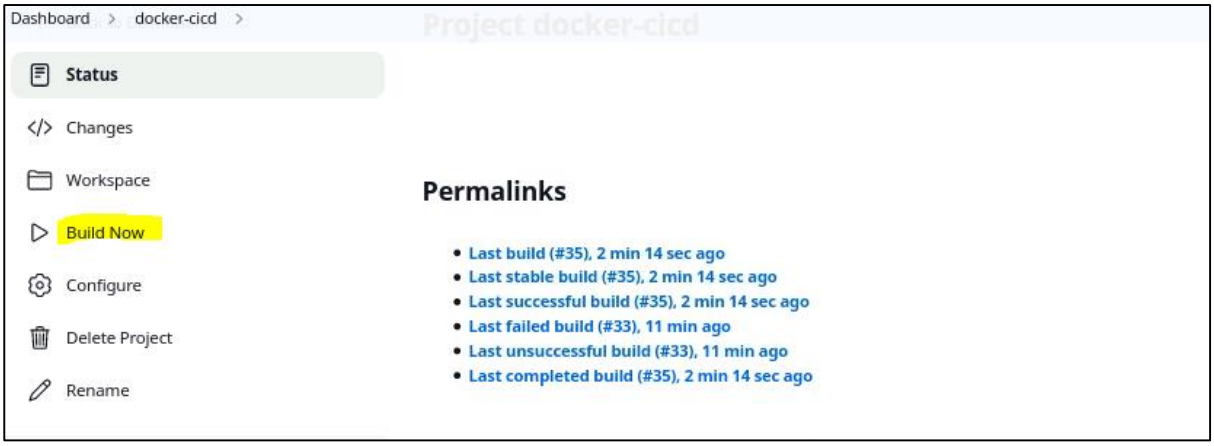


iv. Add docker hub credentials and push latest image to registry

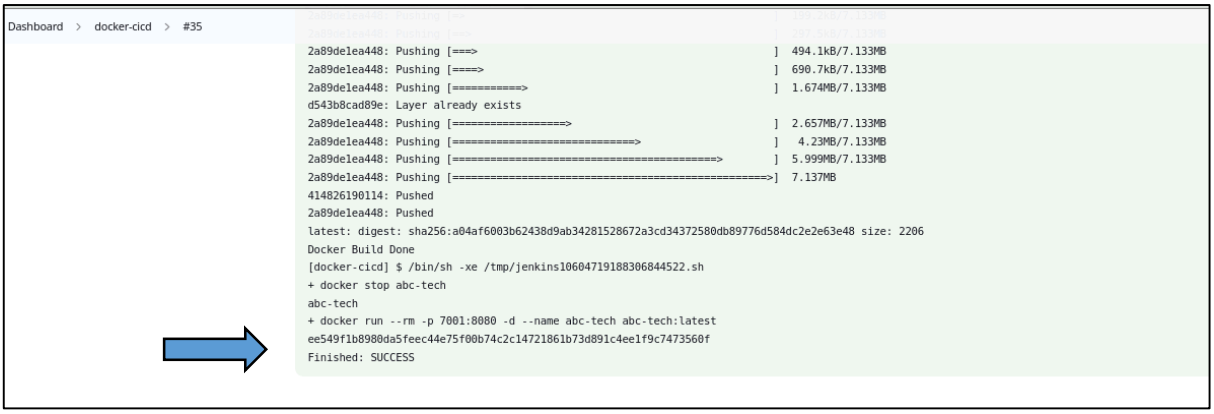
v. Add third build step to execute Shell command to create container



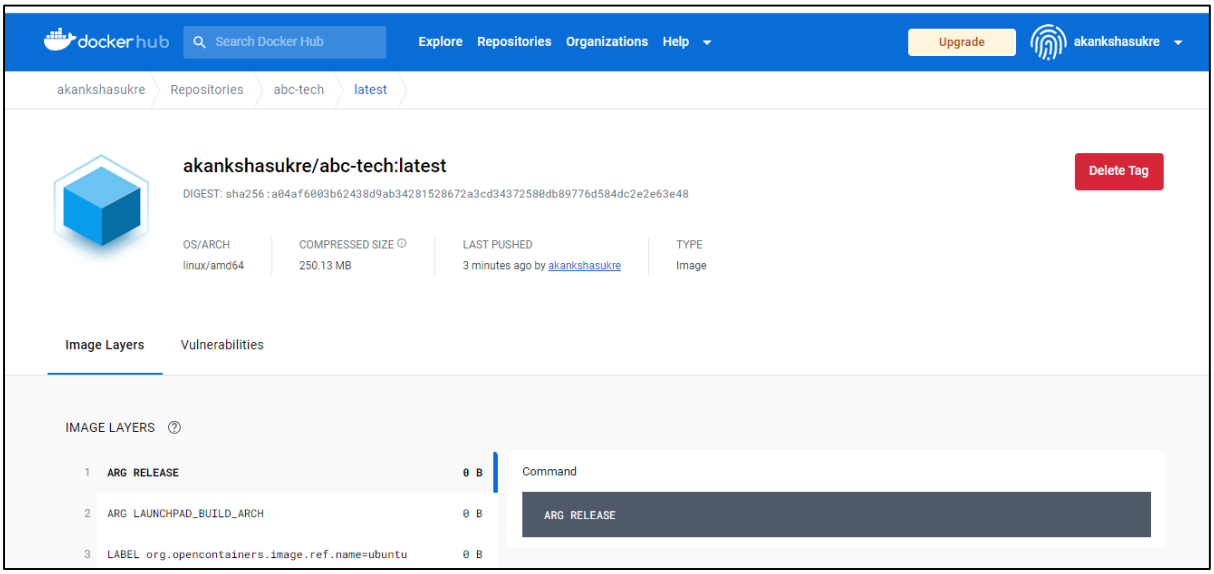
vi. Build the job docker-cicd



vii. Build successful



viii. Latest image built and published to docker hub as seen below



Note: Dockerfile is created and tested on Docker Host before creating this CI/CD pipeline in Jenkins.

Dockerfile is as below:

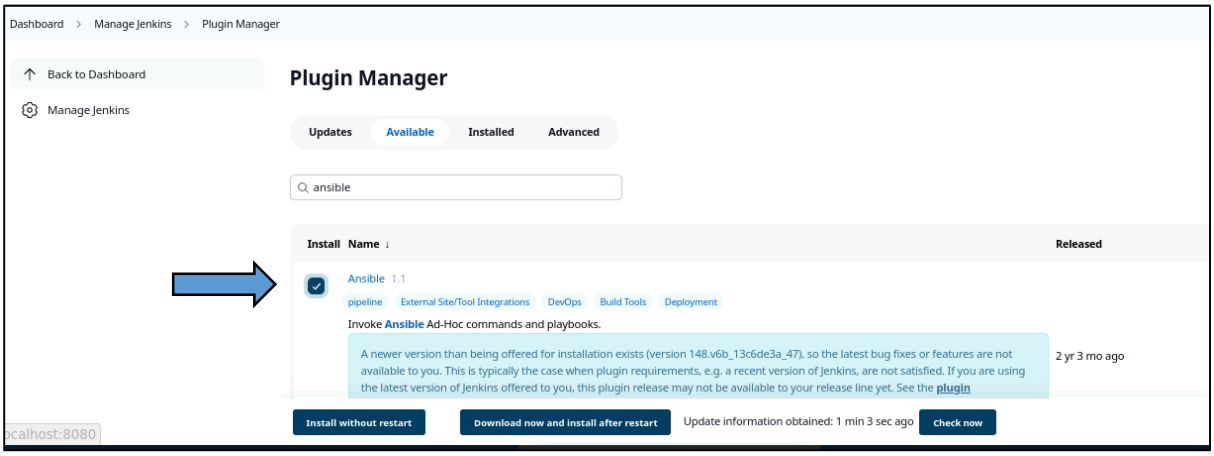
```
FROM tomcat:8.5-jdk11-temurin-focal

LABEL maintainer="Akanksha K Sukre"

WORKDIR /home/edureka/devops-project

ADD /target/ABCtechnologies-1.0.war /usr/local/tomcat/webapps/

EXPOSE 8080

CMD ["catalina.sh", "run"]
```

ix. Container created on Docker host and App is running on the tomcat server at **http://localhost:7001/ABCtechnologies-1.0**

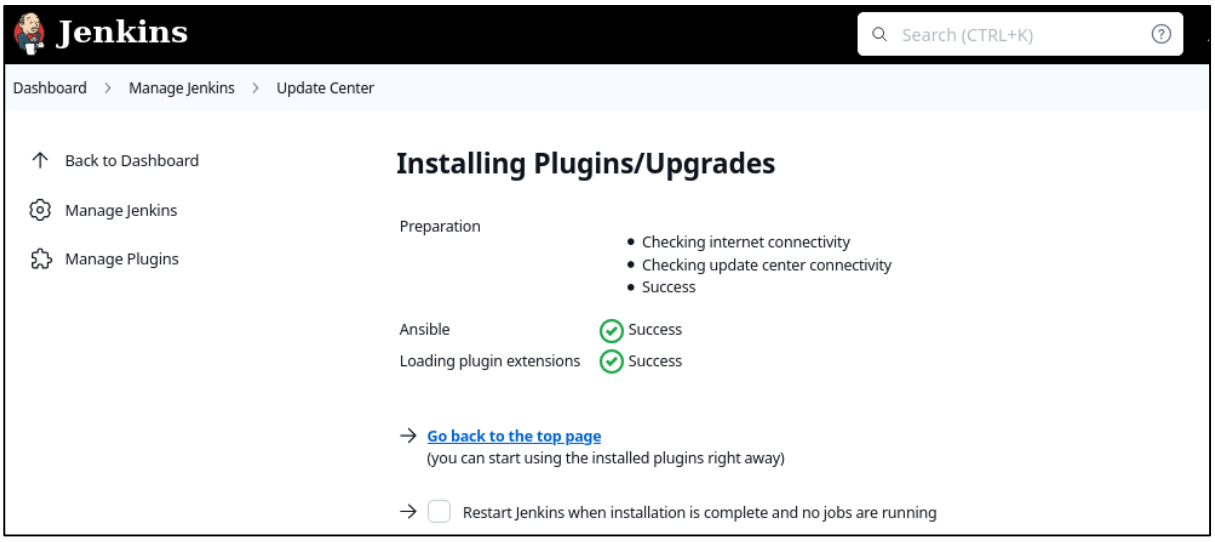
Task 4: Integrate Docker and Kubernetes with Ansible and create CI/CD pipeline in Jenkins to deploy the application on Kubernetes

1) Integrate Ansible with Jenkins

i. Install Ansible plugin in Jenkins



ii. Ansible plugin installed successfully

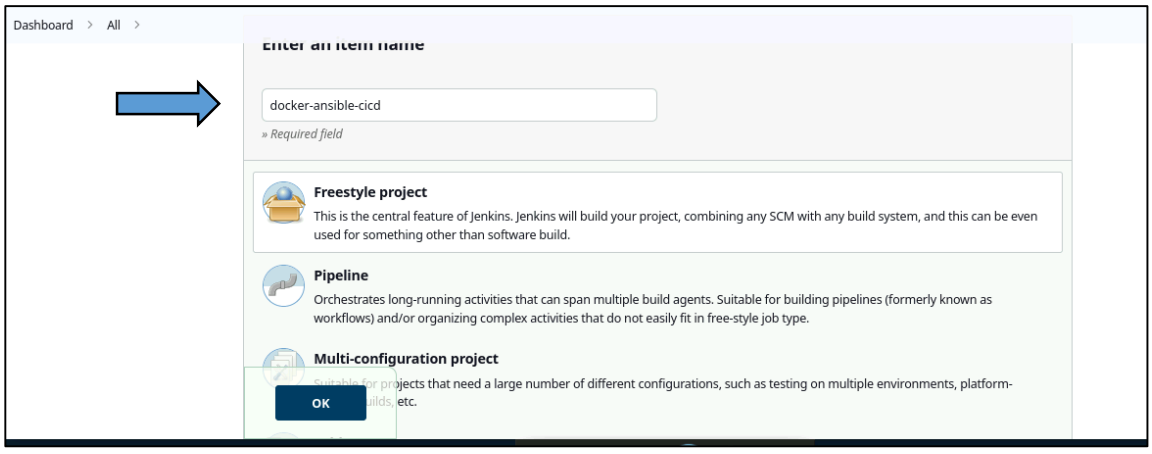


2) Create CI/CD pipeline to create docker image and container using Ansible playbook

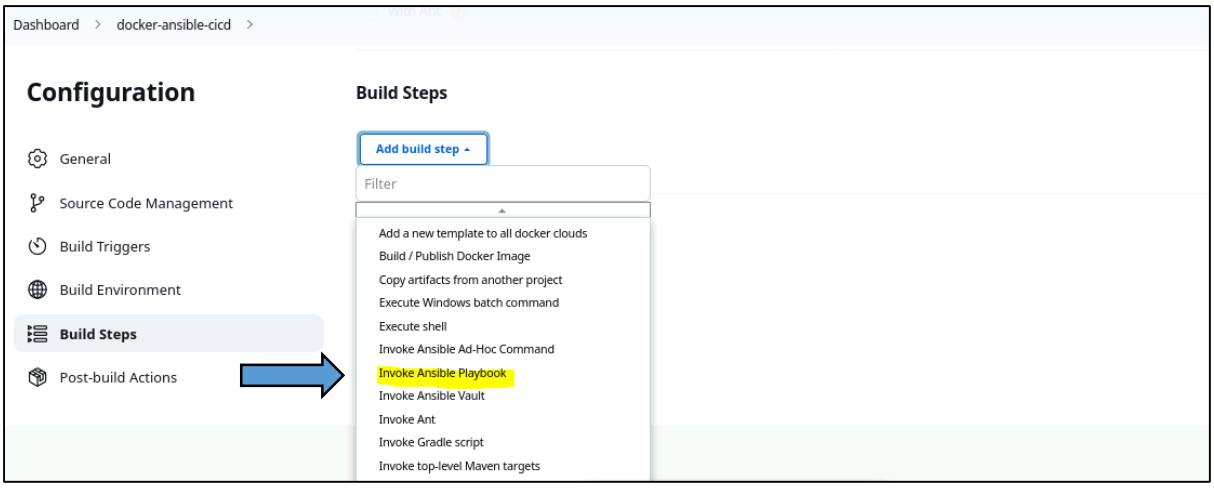
i. Write Ansible playbook to create docker image and container. Playbook contents are as follows:

```
- hosts: localhost
  vars:
      ansible_python_interpreter: /usr/bin/python
      docker_pwd: !vault |
                  $ANSIBLE_VAULT;1.1;AES256
                  6336626166616661623231643363623764326433616363333737653138653531656666631306563330
                  3239343738386138643738366132393232336330646431380a666634373166306237356138666334
                  3861363435376166343532666323864613461636635643632373965646230353830373064636130
                  6263346333306438330a336436626462303166396137306136353732643839353065386564386237
                  3838
  become_method: sudo
  tasks:
    - name: Log into DockerHub
      docker_login:
        username: akankshasukre
        password: '{{ docker_pwd }}'
    - name: Build Docker image from Dockerfile
      docker_image:
        name: abc-tech
        state: present
        source: build
        build:
          path: .
        push: yes
        tag: latest
        repository: akankshasukre/abc-tech
    - name: Running the container
      docker_container:
        image: abc-tech:latest
        name: abc-tech
        recreate: yes
        state: started
        ports:
          - "7001:8080"
```

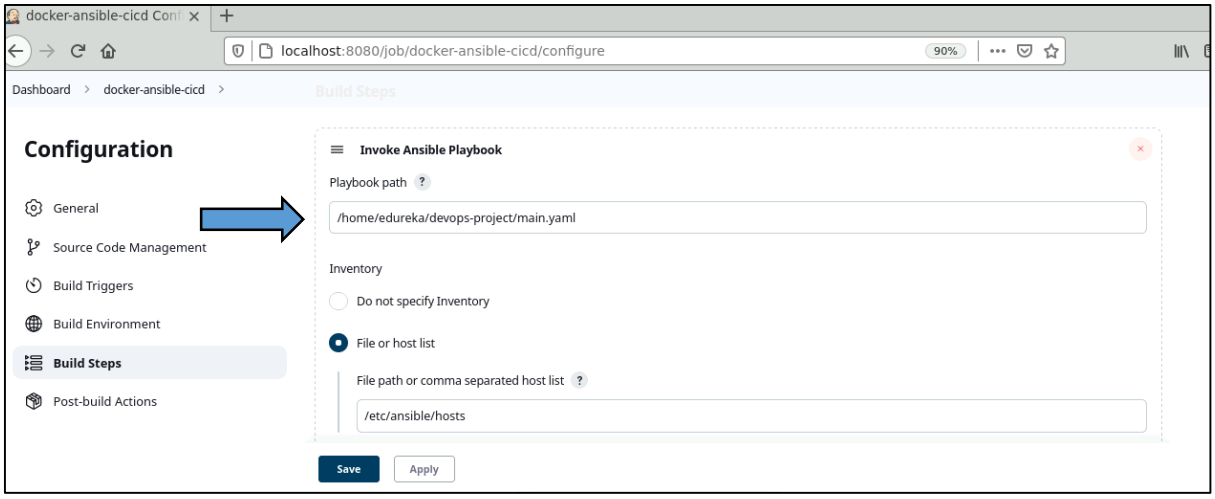
ii. Create a new job: **docker-ansible-cicd**



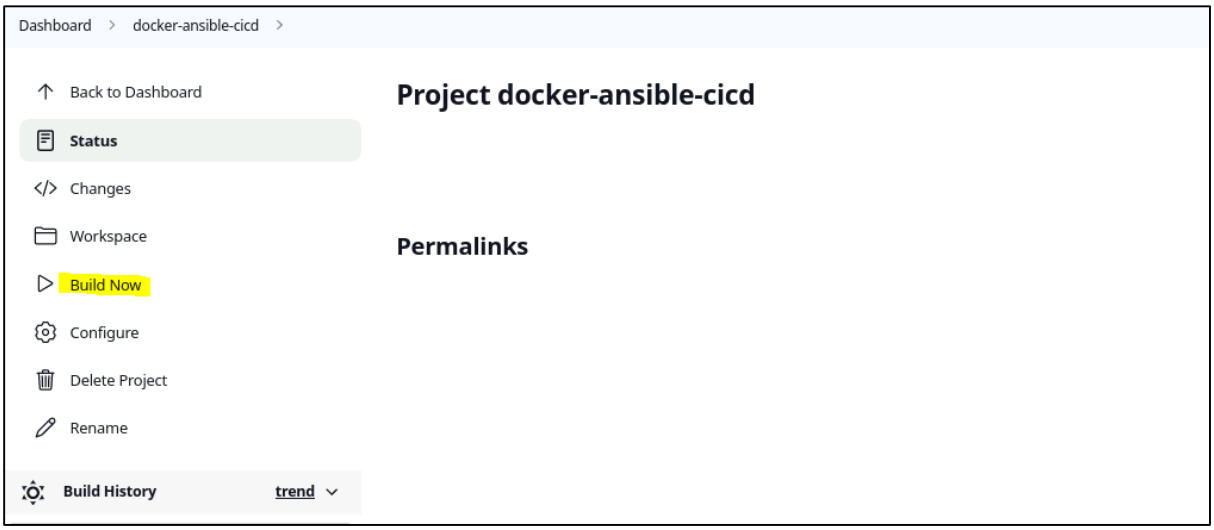
iii. Add a build step to invoke Ansible playbook



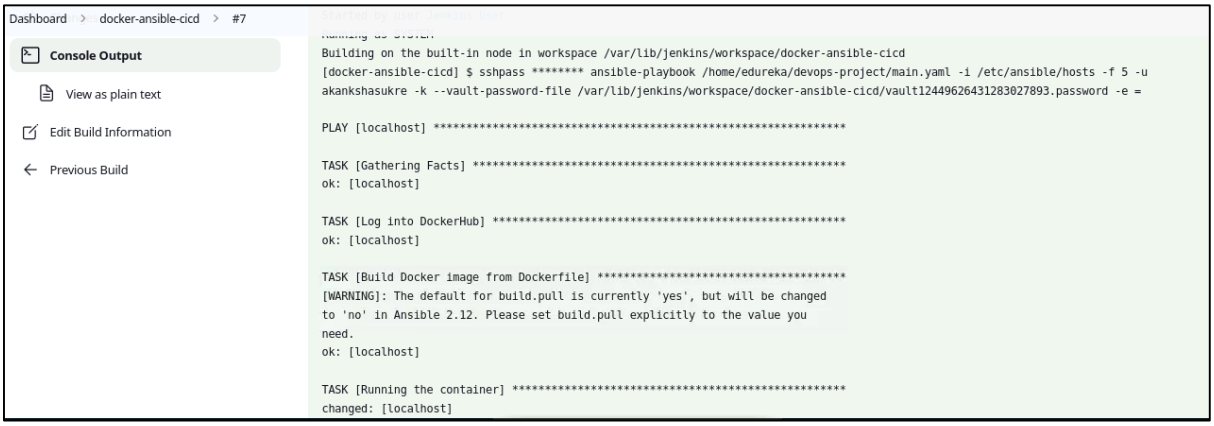
iv. Add path to playbook and hosts file in the required field as shown below:



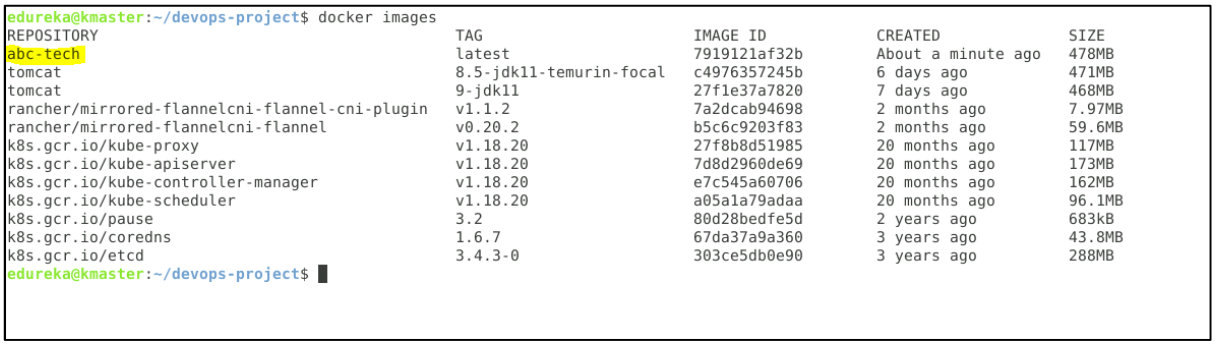
v. Build the job



vi. Build is successful



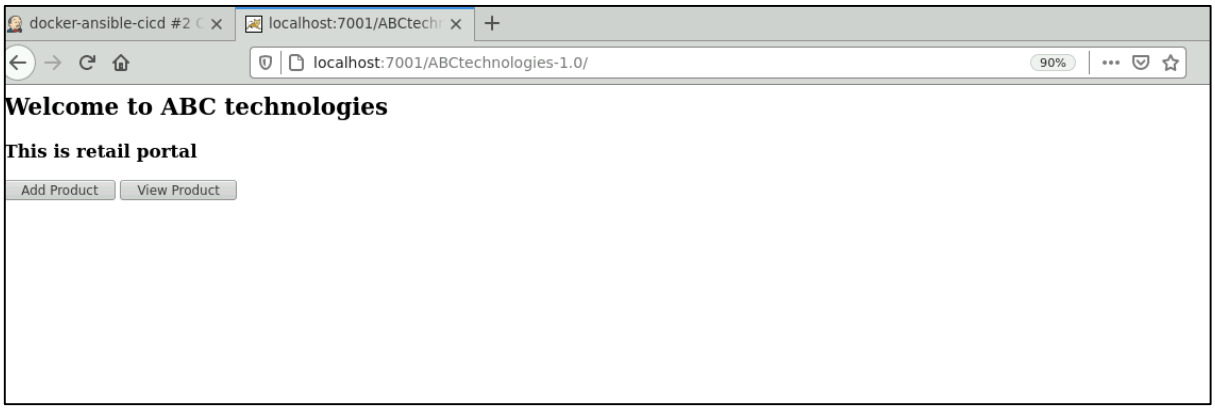
vii. View the image and container created on the docker host



```
edureka@kmaster:~/devops-project$ docker images
REPOSITORY                                   TAG                 IMAGE ID            CREATED             SIZE
abc-tech                                     latest              7919121af32b        About a minute ago  478MB
tomcat                                       8.5-jdk11-temurin-focal   c4976357245b        6 days ago          471MB
tomcat                                       9-jdk11             27f1e37a7820        7 days ago          468MB
rancher/mirrored-flannelcni-flannel-cni-plugin   v1.1.2              7a2dcab94698        2 months ago        7.97MB
rancher/mirrored-flannelcni-flannel          v0.20.2             b5c6c9203f83        2 months ago        59.6MB
k8s.gcr.io/kube-proxy                        v1.18.20            27f8b8d51985        20 months ago       117MB
k8s.gcr.io/kube-apiserver                    v1.18.20            7d8d2960de69        20 months ago       173MB
k8s.gcr.io/kube-controller-manager           v1.18.20            e7c545a60706        20 months ago       162MB
k8s.gcr.io/kube-scheduler                    v1.18.20            a05a1a79adaa        20 months ago       96.1MB
k8s.gcr.io/pause                             3.2                 80d28bedfe5d        2 years ago         683kB
k8s.gcr.io/coredns                           1.6.7               67da37a9a360        3 years ago         43.8MB
k8s.gcr.io/etcd                              3.4.3-0             303ce5db0e90        3 years ago         288MB
edureka@kmaster:~/devops-project$
```

```
edureka@kmaster:~/devops-project$ docker ps
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS              PORTS                    NAMES
8dd364b38c3d        abc-tech:latest        "catalina.sh run"        2 minutes ago       Up 2 minutes        0.0.0.0:7001->8080/tcp   abc-tech
2d378d6973c3        67da37a9a360           "/coredns -conf /etc…"   22 minutes ago      Up 22 minutes                                k8s_coredns_core
dns-66bff467f8-wkqhx_kube-system_b642060f-2dcc-4211-bc8c-25122e24b7f7_9
eb4419046603        k8s.gcr.io/pause:3.2   "/pause"                 22 minutes ago      Up 22 minutes                                k8s_POD_coredns-
66bff467f8-wkqhx_kube-system_b642060f-2dcc-4211-bc8c-25122e24b7f7_49
97b6fbbf409e        67da37a9a360           "/coredns -conf /etc…"   22 minutes ago      Up 22 minutes                                k8s_coredns_core
dns-66bff467f8-zvpsr_kube-system_ea24494f-9bee-4deb-a262-dacace7e6356_10
```

viii. The app is running successfully as seen below



3) Deploy artifacts to Kubernetes

i. Create pod-definition.yaml file. The file is as below:

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: abc-tech
  labels:
    type: webapp
spec:
  containers:
  - name: abc-tech-app
    image: akankshasukre/abc-tech:latest
    ports:
      - containerPort: 8080
```

ii. Create pod using this file and verify using command: kubectl get pods and describe pod

```
edureka@kmaster:~/devops-project/kubernetes$ kubectl create -f pod-definition.yaml
pod/abc-tech created
edureka@kmaster:~/devops-project/kubernetes$ kubectl get pods
NAME       READY   STATUS    RESTARTS   AGE
abc-tech   1/1     Running   0          6s
```

```
edureka@kmaster:~/devops-project/kubernetes$ kubectl describe pod abc-tech
Name:         abc-tech
Namespace:    default
Priority:     0
Node:         kslave1/172.31.18.87
Start Time:   Sat, 11 Feb 2023 14:13:10 +0000
Labels:       type=webapp
Annotations:  <none>
Status:       Running
IP:           10.244.1.3
IPs:
  IP:  10.244.1.3
Containers:
  abc-tech-app:
    Container ID:   docker://29d152ce36015eaf1c0bb8effee9455d96b5c950a2ecbe94c91b3b61b96e3e54
    Image:          akankshasukre/abc-tech:latest
    Image ID:       docker-pullable://akankshasukre/abc-tech@sha256:743d8b5a03e4790efdf5096c1aa46dbd549d7bc78dd772d8eb7ec1
```

iii. Create deployment-definition.yaml file to create new deployment

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: abc-tech-deployment
  labels:
    type: webapp
spec:
  replicas: 3
  selector:
    matchLabels:
      type: webapp
  template:
    metadata:
      labels:
        name: abc-tech
        type: webapp
    spec:
      containers:
      - name: abc-tech-app
        image: akankshasukre/abc-tech:latest
        ports:
        - containerPort: 8080
        imagePullPolicy: Always
```

iv. Run below commands to create new deployment using this file and later view the created deployment.

```
edureka@kmaster:~/devops-project/kubernetes$ kubectl delete pod abc-tech
pod "abc-tech" deleted
edureka@kmaster:~/devops-project/kubernetes$ vi deployment-definition.yaml
edureka@kmaster:~/devops-project/kubernetes$ kubectl create -f deployment-definition.yaml
deployment.apps/abc-tech-deployment created
edureka@kmaster:~/devops-project/kubernetes$ kubectl get deployments
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
abc-tech-deployment   3/3     3            3           12s
```

```
edureka@kmaster:~/devops-project/kubernetes$ kubectl get all
NAME                                      READY   STATUS    RESTARTS   AGE
pod/abc-tech-deployment-7bc6c66448-24dhl  1/1     Running   0          35s
pod/abc-tech-deployment-7bc6c66448-dzt5l  1/1     Running   0          35s
pod/abc-tech-deployment-7bc6c66448-qhprp  1/1     Running   0          35s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   27d

NAME                                 READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/abc-tech-deployment  3/3     3            3           35s

NAME                                            DESIRED   CURRENT   READY   AGE
replicaset.apps/abc-tech-deployment-7bc6c66448  3         3         3       35s
edureka@kmaster:~/devops-project/kubernetes$
```

v. Create service-definition.yaml file to create a service in Kubernetes cluster.

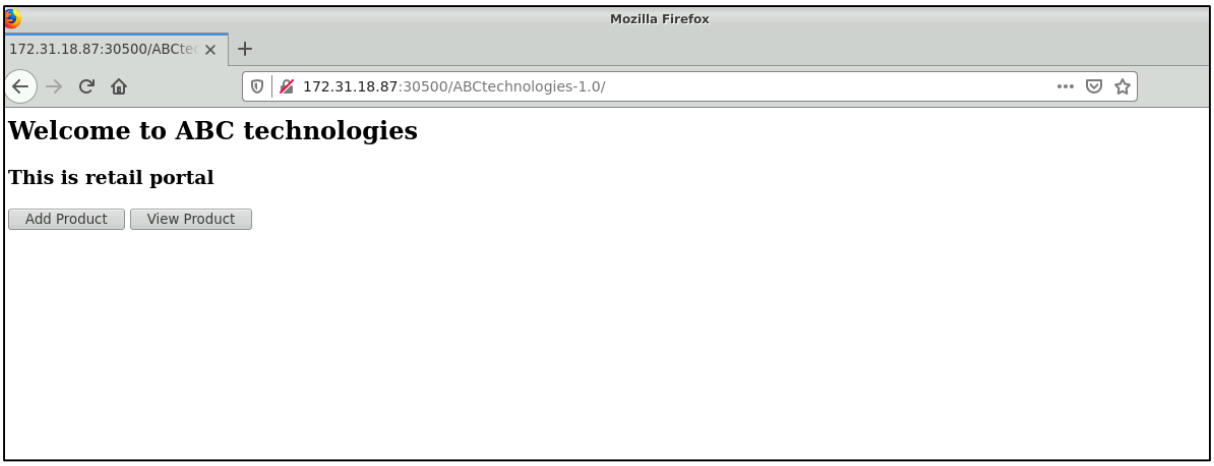
```yaml
apiVersion: v1
kind: Service
metadata:
  name: abc-tech-svc
  labels:
    type: webapp
spec:
  ports:
  - nodePort: 30500
    port: 80
    protocol: TCP
    targetPort: 8080
  selector:
    type: webapp
  type: NodePort
```

vi. Run kubectl command to create service and get service.

```
edureka@kmaster:~/devops-project/kubernetes$ kubectl create -f service-definition.yaml
service/abc-tech-svc created
edureka@kmaster:~/devops-project/kubernetes$ kubectl get svc
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
abc-tech-svc   NodePort    10.100.51.55    <none>        80:30500/TCP   4s
kubernetes     ClusterIP   10.96.0.1       <none>        443/TCP        27d
edureka@kmaster:~/devops-project/kubernetes$
```

```
edureka@kmaster:~/devops-project/kubernetes$ kubectl describe service
Name:                     abc-tech-svc
Namespace:                default
Labels:                   type=webapp
Annotations:              <none>
Selector:                 type=webapp
Type:                     NodePort
IP:                       10.100.51.55
Port:                     <unset>  80/TCP
TargetPort:               8080/TCP
NodePort:                 <unset>  30500/TCP
Endpoints:                10.244.1.4:8080,10.244.1.5:8080,10.244.1.6:8080
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>


Name:              kubernetes
Namespace:         default
Labels:            component=apiserver
                   provider=kubernetes
Annotations:       <none>
Selector:          <none>
Type:              ClusterIP
IP:                10.96.0.1
Port:              https  443/TCP
TargetPort:        6443/TCP
```

vii. Verify the deployment by hitting the URL as seen below:
http://172.31.18.87:30500/ABCtechnologies-1.0



4) Integrate Ansible with Kubernetes. Create CI/CD Pipeline to deploy artifacts on Kubernetes using Ansible playbook.

i. Create hosts file in Ansible with Kubernetes master and worker node details.

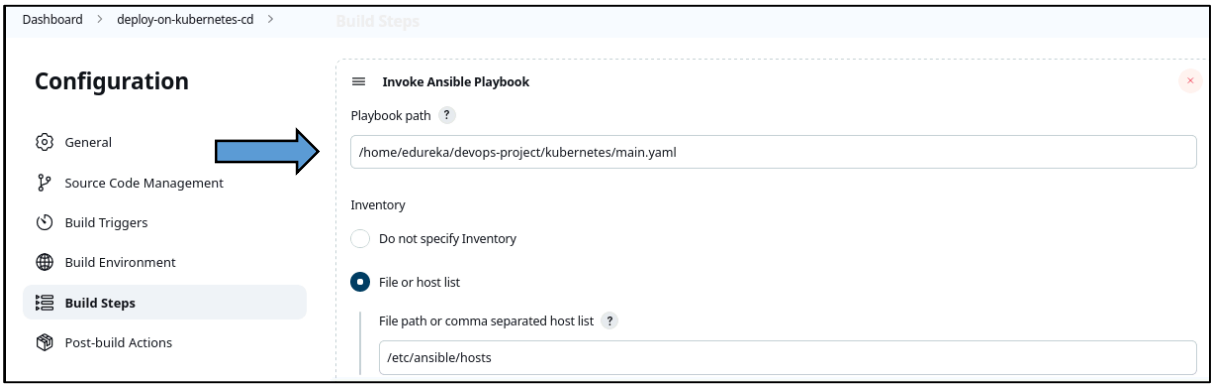
```
[masters]
master ansible_host=172.31.9.156 ansible_user=edureka ansible_connection=ssh ansible_python_interpreter=/usr/bin/python
[workers]
worker1 ansible_host=172.31.18.87 ansible_ssh_user=edureka ansible_connection=ssh ansible_python_interpreter=/usr/bin/python
```

ii. Create Ansible playbook main.yaml to create deployment and service in Kubernetes as below:

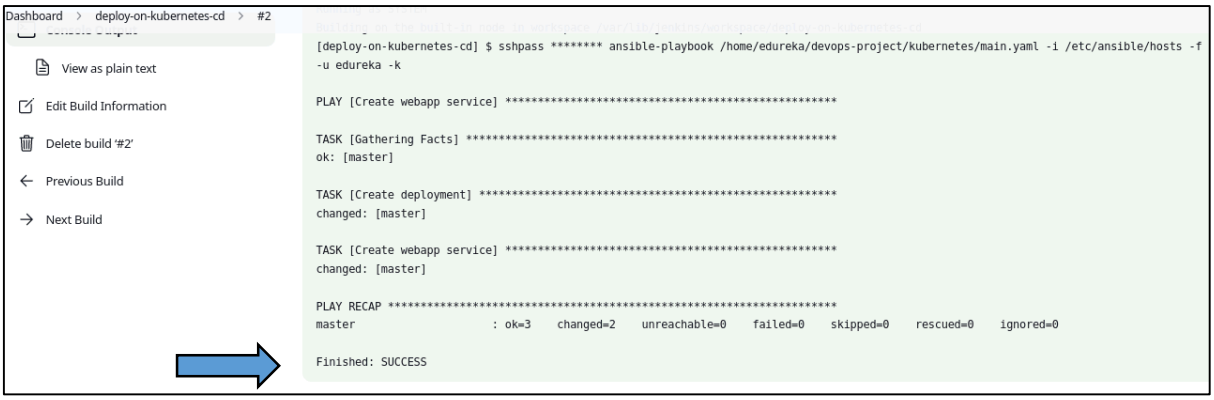
```yaml
- name: Create deployment and service in kubernetes
  hosts: master
  become: true
  user: edureka
  tasks:
    - name: Create deployment
      command: kubectl apply -f deployment-definition.yaml --kubeconfig=/home/edureka/.kube/admin.conf
      args:
        chdir: /home/edureka/devops-project/kubernetes

    - name: Update deployment with new pods if docker hub image is updated
      command: kubectl rollout restart deployment abc-tech-deployment --kubeconfig=/home/edureka/.kube/admin.conf

    - name: Create webapp service
      command: kubectl apply -f service-definition.yaml --kubeconfig=/home/edureka/.kube/admin.conf
      args:
        chdir: /home/edureka/devops-project/kubernetes
```

```
edureka@kmaster:~/devops-project/kubernetes$ ls -l
total 16
-rw-r--r-- 1 edureka edureka 399 Feb 11 14:19 deployment-definition.yaml
-rw-r--r-- 1 root    root    494 Feb 13 12:23 main.yaml
-rw-r--r-- 1 edureka edureka 201 Feb 11 14:11 pod-definition.yaml
-rw-r--r-- 1 edureka edureka 221 Feb 11 14:21 service-definition.yaml
edureka@kmaster:~/devops-project/kubernetes$
```

iii. Create a new Jenkins job: **deploy-on-kubernetes-cd**. Add build step to invoke Ansible playbook and give the path to main.yaml file in Jenkins master.



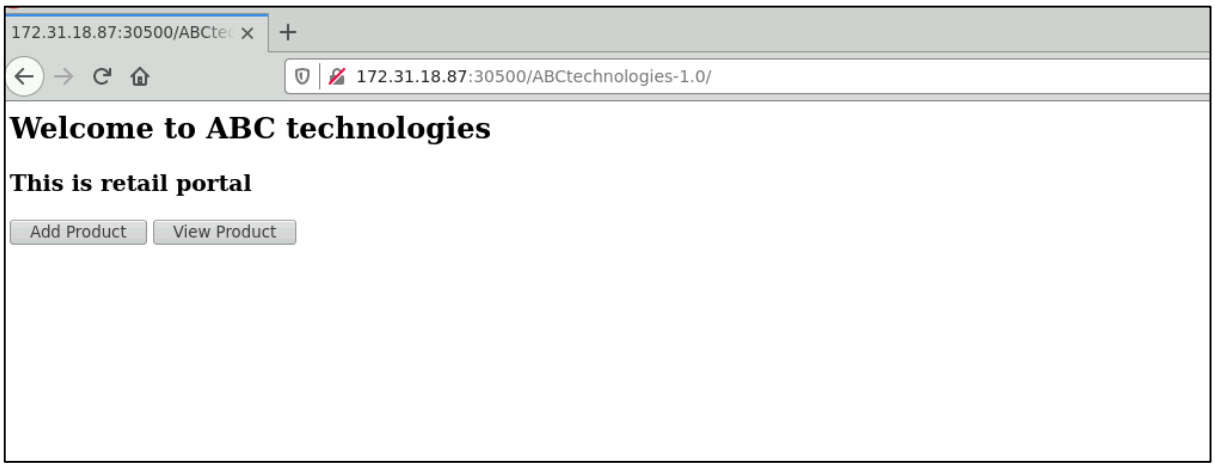
iv. Run the job and build is successful as seen below.

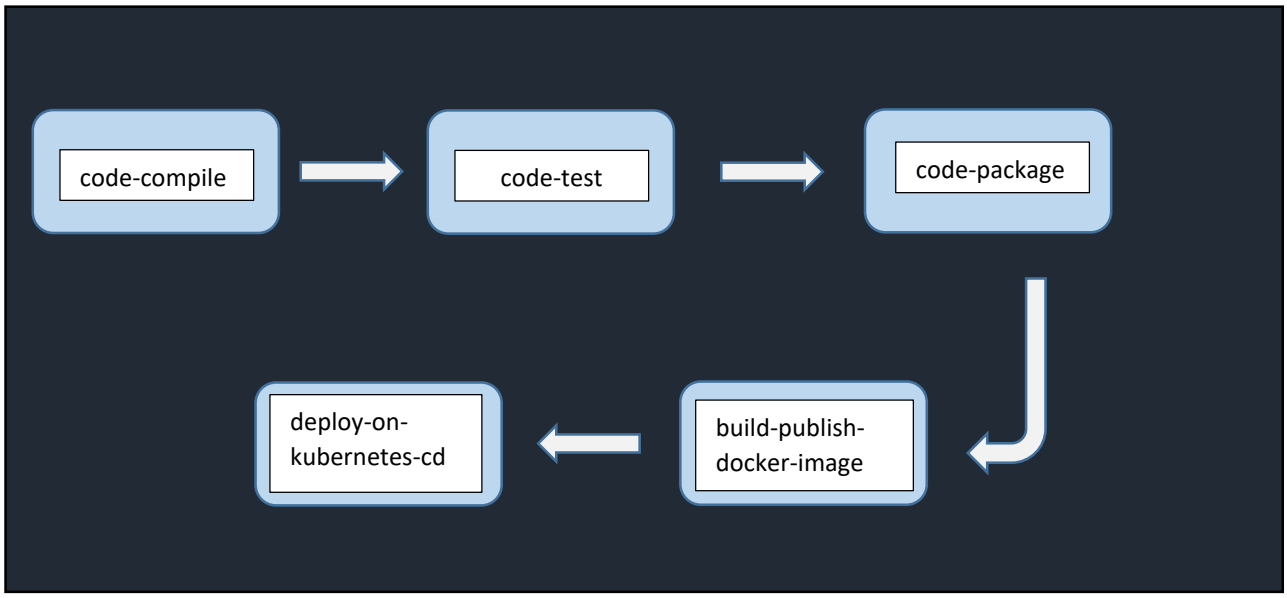


v. Verify the deployment using kubectl command as shown below:

```
edureka@kmaster:~/devops-project/kubernetes$ kubectl get all
NAME                                        READY   STATUS    RESTARTS   AGE
pod/abc-tech-deployment-7bc6c66448-6967w    1/1     Running   0          5m18s
pod/abc-tech-deployment-7bc6c66448-jkpvk    1/1     Running   0          5m18s
pod/abc-tech-deployment-7bc6c66448-n5znj    1/1     Running   0          5m18s

NAME                     TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
service/abc-tech-svc     NodePort    10.102.138.180   <none>        80:30500/TCP   5m17s
service/kubernetes       ClusterIP   10.96.0.1        <none>        443/TCP        29d

NAME                                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/abc-tech-deployment     3/3     3            3           5m18s

NAME                                              DESIRED   CURRENT   READY   AGE
replicaset.apps/abc-tech-deployment-7bc6c66448    3         3         3       5m18s
edureka@kmaster:~/devops-project/kubernetes$
```

vi. Verify that the application is running on desired port.





END TO END PIPELINE FLOW

5) Create end to end pipeline: Build and Publish docker image & Deploy to Kubernetes using Ansible

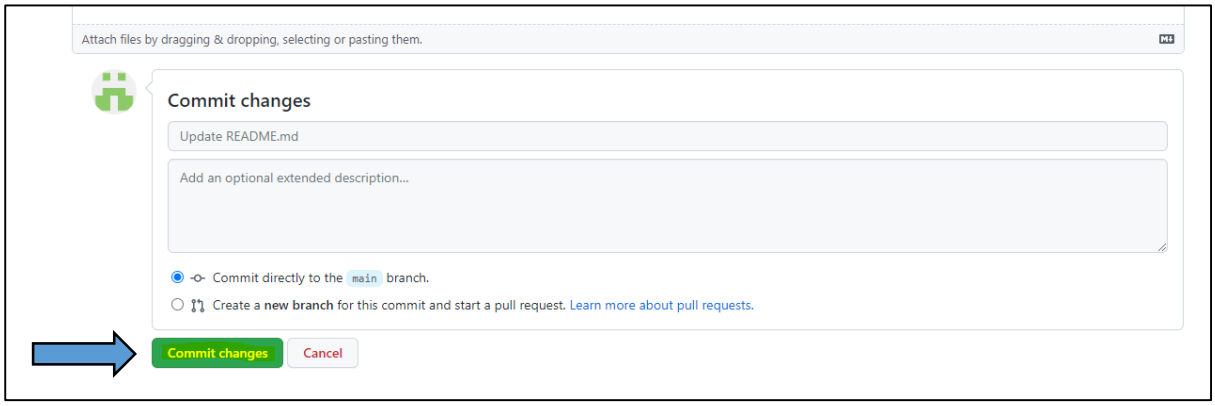
i. Add a post-build action in Package job to run the job: build-publish-docker-image

ii. Create a new job: **build-publish-docker-image** in Jenkins and add build step to copy artifacts from Package job and then to invoke following Ansible playbook:

```
- hosts: localhost
  vars:
      ansible_python_interpreter: /usr/bin/python
      docker_pwd: !vault |
                  $ANSIBLE_VAULT;1.1;AES256
                  633662616661666162323164336362376432643361636333737653138653531656666313065633
                  323934373838613864373836613239323233633064643138 0a666634373166306237356138666334
                  386136343537616634353266633238646134616366356436323739656462303538303730646361 30
                  626334633330643833 0a336436626462303166396137306136353732643839353065386564386237
                  3838
  become_method: sudo
  tasks:
    - name: Log into DockerHub
      docker_login:
        username: akankshasukre
        password: '{{ docker_pwd }}'
    - name: Build Docker image from Dockerfile
      docker_image:
        name: abc-tech
        state: present
        source: build
        build:
          path: .
        push: yes
        tag: latest
        repository: akankshasukre/abc-tech
```

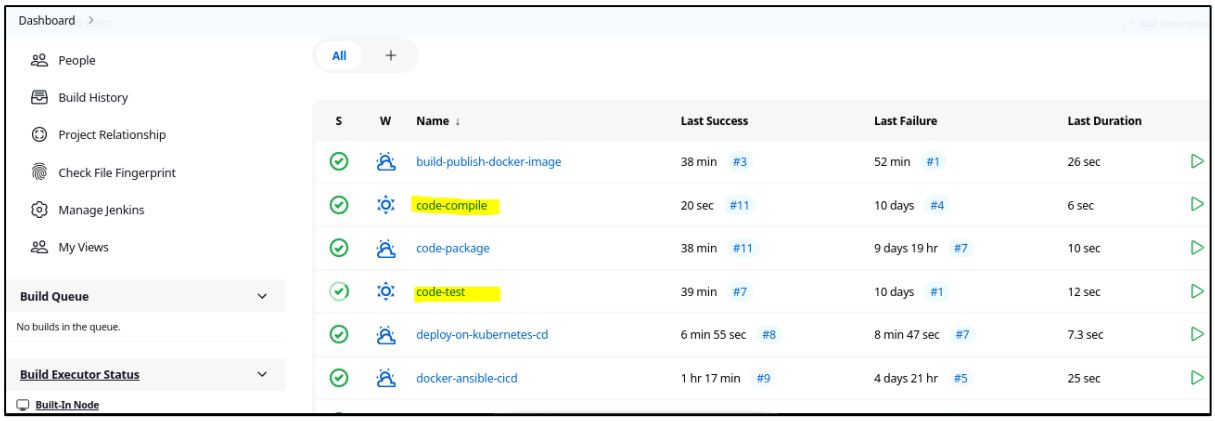
iii. Add a post-build action in this job to build the kubernetes job i.e. **deploy-on-kubernetes-cd**

iv. Test the final pipeline by committing a new change in GIT repo

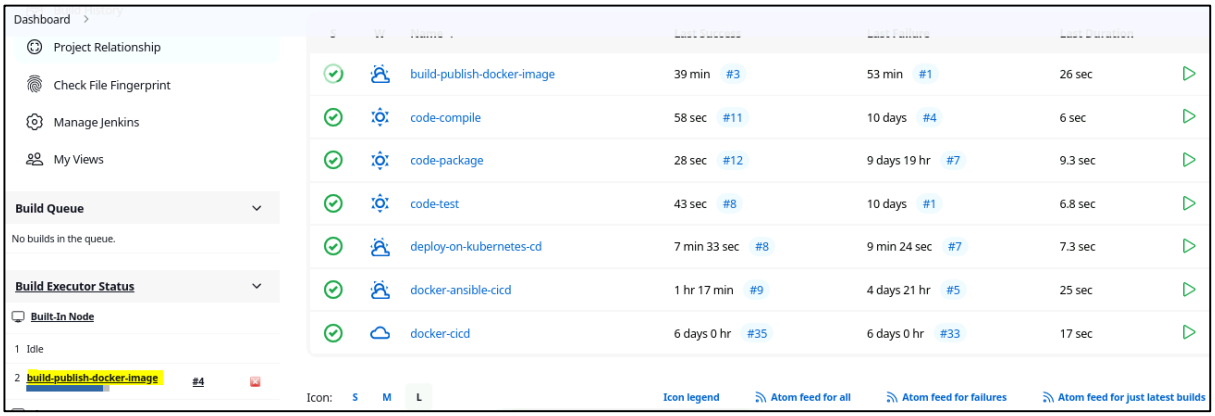


v. Check the Git Polling Log of Compile job

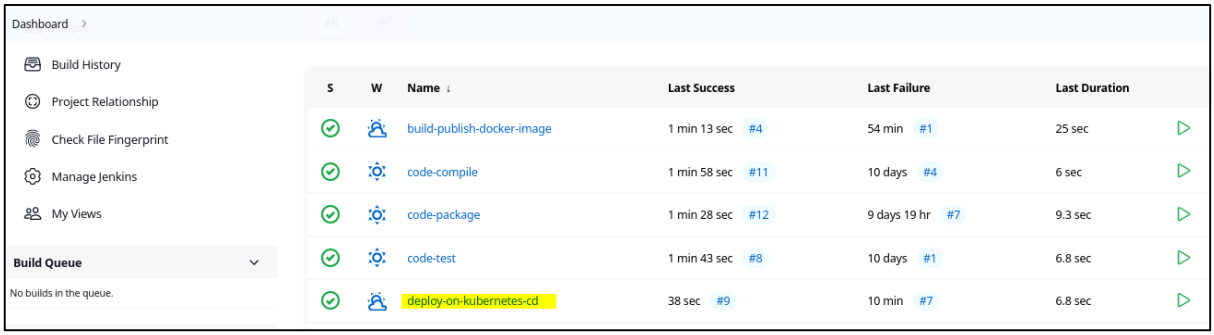
vi. Compile job is triggered which further triggers the test job

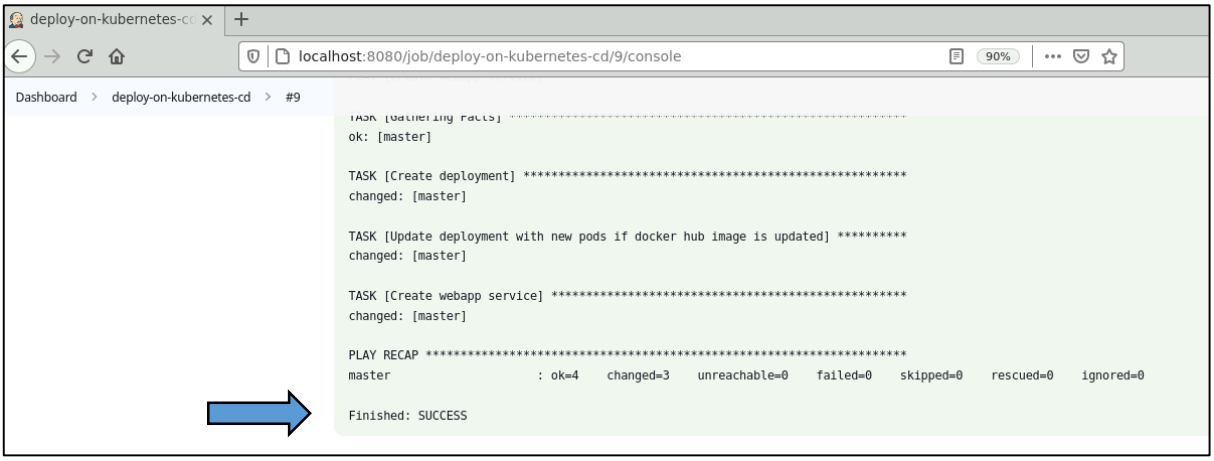


vii. After the test job, package job runs and then triggers the build-publish-docker-image job.

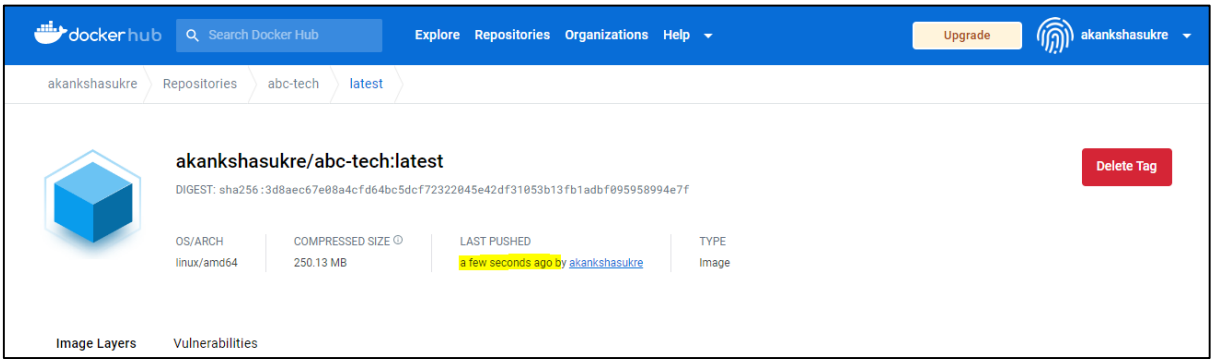


viii. Finally the last job(deploy-on-kubernetes-cd) is triggered which completes the pipeline





ix. Verify the deployment. Latest image is pushed to docker hub repository.



x. The application is running successfully on the given port as expected. Thus, the deployment is complete.

```
edureka@kmaster:~/devops-project/kubernetes$ kubectl get all
NAME                                          READY   STATUS    RESTARTS   AGE
pod/abc-tech-deployment-75bcb44865-84fv9      1/1     Running   0          95s
pod/abc-tech-deployment-75bcb44865-87csc      1/1     Running   0          100s
pod/abc-tech-deployment-75bcb44865-d8rzj      1/1     Running   0          107s

NAME                     TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
service/abc-tech-svc     NodePort    10.102.138.180   <none>        80:30500/TCP   3h4m
service/kubernetes       ClusterIP   10.96.0.1        <none>        443/TCP        29d

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/abc-tech-deployment      3/3     3            3           3h4m

NAME                                                DESIRED   CURRENT   READY   AGE
replicaset.apps/abc-tech-deployment-75bcb44865      3         3         3       107s
replicaset.apps/abc-tech-deployment-7bc6c66448      0         0         0       3h4m
replicaset.apps/abc-tech-deployment-95d8d6d8b       0         0         0       9m42s
```

Task 5: Monitoring the Kubernetes Cluster using Prometheus and creating dashboard in Grafana to view important metrics.

1) Install Node Exporter on master and slave node

i. Download node exporter on master node

```
edureka@kmaster:~$ wget https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_exporter-1.3.1.linux-amd64.tar.gz
--2023-02-13 15:54:22--  https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_exporter-1.3.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/7c60f6f9-7b41-446c-be81-a6c24a9d0383?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20230213%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20230213T155227Z&X-Amz-Expires=300&X-Amz-Signature=4637b6fe8490bed10bc4acc8ce42db71d71fea20642a04d030d0fd881eec6432&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=9524057&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.3.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2023-02-13 15:54:22--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/7c60f6f9-7b41-446c-be81-a6c24a9d0383?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20230213%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20230213T155227Z&X-Amz-Expires=300&X-Amz-Signature=4637b6fe8490bed10bc4acc8ce42db71d71fea20642a04d030d0fd881eec6432&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=9524057&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.3.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.111.133, 185.199.110.133, 185.199.109.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9033415 (8.6M) [application/octet-stream]
Saving to: 'node_exporter-1.3.1.linux-amd64.tar.gz'

node_exporter-1.3.1.linux-amd64.tar.g 100%[=========================================================================>]   8.61M  --.-KB/s    in 0.05s

2023-02-13 15:54:22 (168 MB/s) - 'node_exporter-1.3.1.linux-amd64.tar.gz' saved [9033415/9033415]
```

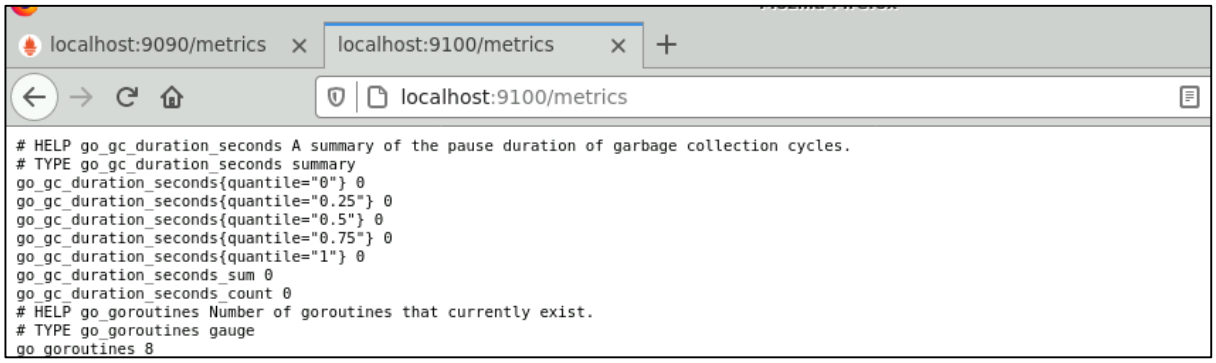
ii. Extract the node-exporter zip

```
edureka@kmaster:~$ tar xvfz node_exporter-*.*-amd64.tar.gz
node_exporter-1.3.1.linux-amd64/
node_exporter-1.3.1.linux-amd64/LICENSE
node_exporter-1.3.1.linux-amd64/NOTICE
node_exporter-1.3.1.linux-amd64/node_exporter
edureka@kmaster:~$
```

iii. Run node-exporter on master and verify functioning

```
edureka@kmaster:~$ cd node_exporter-1.3.1.linux-amd64/
edureka@kmaster:~/node_exporter-1.3.1.linux-amd64$ ls -l
total 17820
-rw-r--r-- 1 edureka edureka    11357 Dec  5  2021 LICENSE
-rw-r--r-- 1 edureka edureka      463 Dec  5  2021 NOTICE
-rwxr-xr-x 1 edureka edureka 18228926 Dec  5  2021 node_exporter
edureka@kmaster:~/node_exporter-1.3.1.linux-amd64$ ./node_exporter
ts=2023-02-13T15:55:28.967Z caller=node_exporter.go:182 level=info msg="Starting node_exporter" version="(version=1.3.1, branch=HEAD, revision=a2321e7b940ddcff26873612bccdf7cd4c42b6b6)"
ts=2023-02-13T15:55:28.967Z caller=node_exporter.go:183 level=info msg="Build context" build_context="(go=go1.17.3, user=root@243aafa5525c, date=20211205-11:09:49)"
ts=2023-02-13T15:55:28.969Z caller=filesystem_common.go:111 level=info collector=filesystem msg="Parsed flag --collector.filesystem.mount-points-exclude" flag=^/(dev|proc|run/credentials/.+|sys|var/lib/docker/.+)($|/)
ts=2023-02-13T15:55:28.969Z caller=filesystem_common.go:113 level=info collector=filesystem msg="Parsed flag --collector.filesystem.fs-types-exclude" flag=^(autofs|binfmt_misc|bpf|cgroup2?|configfs|debugfs|devpts|devtmpfs|fusectl|hugetlbfs|iso9660|mqueue|nsfs|overlay|proc|procfs|pstore|rpc_pipefs|securityfs|selinuxfs|squashfs|sysfs|tracefs)$
ts=2023-02-13T15:55:28.969Z caller=node_exporter.go:108 level=info msg="Enabled collectors"
```

iv. Hit the URL to view node-exporter metrics



v. Install node-exporter on slave node as well

```
edureka@kslave1:~$ wget https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_exporter-1.3.1.linux-amd64.tar.gz
--2023-02-13 16:30:26--  https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_exporter-1.3.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/7c60f6f9-7b41-446c-be81-a6c24a9d0383?X-Amz-A
WS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20230213%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20230213T163027Z&X-Amz-Expires=
Signature=50365835b4600b502d4dce407623bc294bb8cfe69bc4c6676590fd17e519399b&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=9524057&res
ent-disposition=attachment%3B%20filename%3Dnode_exporter-1.3.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [followi
--2023-02-13 16:30:26--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/7c60f6f9-7b41-446c-be81-a6c24
Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20230213%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20230213T163027Z
res=300&X-Amz-Signature=50365835b4600b502d4dce407623bc294bb8cfe69bc4c6676590fd17e519399b&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo
7&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.3.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.111.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9033415 (8.6M) [application/octet-stream]
Saving to: 'node_exporter-1.3.1.linux-amd64.tar.gz'

node_exporter-1.3.1 100%[===================>]   8.61M  5.36MB/s    in 1.6s

2023-02-13 16:30:28 (5.36 MB/s) - 'node_exporter-1.3.1.linux-amd64.tar.gz' saved [9033415/9033415]

edureka@kslave1:~$
```

```
edureka@kslave1:~$ tar xvfz node_exporter-*.*-amd64.tar.gz
node_exporter-1.3.1.linux-amd64/
node_exporter-1.3.1.linux-amd64/LICENSE
node_exporter-1.3.1.linux-amd64/NOTICE
node_exporter-1.3.1.linux-amd64/node_exporter
edureka@kslave1:~$
edureka@kslave1:~$ cd node_exporter-1.3.1.linux-amd64/
edureka@kslave1:~/node_exporter-1.3.1.linux-amd64$ ./node_exporter
ts=2023-02-13T16:30:51.793Z caller=node_exporter.go:182 level=info msg="Starting node_exporter" version="(version=1.3.1, branch=
7b940ddcff26873612bccdf7cd4c42b6b6)"
ts=2023-02-13T16:30:51.793Z caller=node_exporter.go:183 level=info msg="Build context" build_context="(go=go1.17.3, user=root@2
1205-11:09:49)"
ts=2023-02-13T16:30:51.793Z caller=filesystem_common.go:111 level=info collector=filesystem msg="Parsed flag --collector.filesy
ude" flag=^/(dev|proc|run/credentials/.+|sys|var/lib/docker/.+)($|/)
ts=2023-02-13T16:30:51.794Z caller=filesystem_common.go:113 level=info collector=filesystem msg="Parsed flag --collector.filesy
 flag=^(autofs|binfmt_misc|bpf|cgroup2?|configfs|debugfs|devpts|devtmpfs|fusectl|hugetlbfs|iso9660|mqueue|nsfs|overlay|proc|pro
securityfs|selinuxfs|squashfs|sysfs|tracefs)$
```

vi. Install node exporter as a service on master and slave nodes

```
-rwxr-xr-x 1 edureka edureka 18228926 Dec  5  2021 node_exporter
edureka@kmaster:~/node_exporter-1.3.1.linux-amd64$ sudo cp node_exporter /usr/local/bin
edureka@kmaster:~/node_exporter-1.3.1.linux-amd64$ sudo useradd --no-create-home --shell /bin/false node_exporter
edureka@kmaster:~/node_exporter-1.3.1.linux-amd64$ sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter
edureka@kmaster:~/node_exporter-1.3.1.linux-amd64$ sudo nano /etc/systemd/system/node_exporter.service
edureka@kmaster:~/node_exporter-1.3.1.linux-amd64$ sudo systemctl daemon-reload
edureka@kmaster:~/node_exporter-1.3.1.linux-amd64$ sudo systemctl enable node_exporter
Created symlink /etc/systemd/system/multi-user.target.wants/node_exporter.service → /etc/systemd/system/node_exporter.service.
edureka@kmaster:~/node_exporter-1.3.1.linux-amd64$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; vendor preset: enabled)
   Active: inactive (dead)
edureka@kmaster:~/node_exporter-1.3.1.linux-amd64$ sudo systemctl start node_exporter.service
edureka@kmaster:~/node_exporter-1.3.1.linux-amd64$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-02-14 17:08:33 UTC; 4s ago
 Main PID: 9449 (node_exporter)
    Tasks: 4 (limit: 4915)
   CGroup: /system.slice/node_exporter.service
           └─9449 /usr/local/bin/node_exporter

Feb 14 17:08:33 kmaster node_exporter[9449]: ts=2023-02-14T17:08:33.692Z caller=node_exporter.go:115 level=info collector=thermal_zone
Feb 14 17:08:33 kmaster node_exporter[9449]: ts=2023-02-14T17:08:33.692Z caller=node_exporter.go:115 level=info collector=time
Feb 14 17:08:33 kmaster node_exporter[9449]: ts=2023-02-14T17:08:33.692Z caller=node_exporter.go:115 level=info collector=timex
Feb 14 17:08:33 kmaster node_exporter[9449]: ts=2023-02-14T17:08:33.692Z caller=node_exporter.go:115 level=info collector=udp_queues
Feb 14 17:08:33 kmaster node_exporter[9449]: ts=2023-02-14T17:08:33.692Z caller=node_exporter.go:115 level=info collector=uname
```

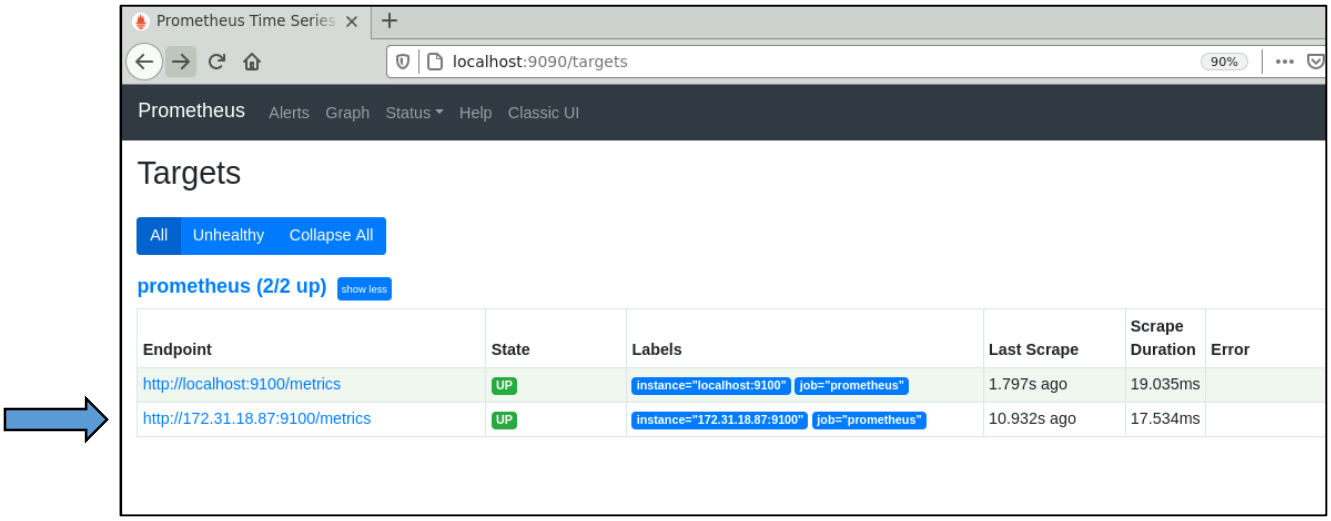
```
edureka@kslave1:~/node_exporter-1.3.1.linux-amd64$ sudo cp node_exporter /usr/local/bin
[sudo] password for edureka:
edureka@kslave1:~/node_exporter-1.3.1.linux-amd64$ sudo cp node_exporter /usr/local/bin
edureka@kslave1:~/node_exporter-1.3.1.linux-amd64$ sudo useradd --no-create-home --shell /bin/false node_exporter
edureka@kslave1:~/node_exporter-1.3.1.linux-amd64$ sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter
edureka@kslave1:~/node_exporter-1.3.1.linux-amd64$ sudo nano /etc/systemd/system/node_exporter.service
edureka@kslave1:~/node_exporter-1.3.1.linux-amd64$ sudo systemctl daemon-reload
edureka@kslave1:~/node_exporter-1.3.1.linux-amd64$ sudo systemctl enable node_exporter
Created symlink /etc/systemd/system/multi-user.target.wants/node_exporter.service → /etc/systemd/system/node_exporter.service.
edureka@kslave1:~/node_exporter-1.3.1.linux-amd64$ sudo systemctl start node_exporter
edureka@kslave1:~/node_exporter-1.3.1.linux-amd64$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; vendor pr
   Active: active (running) since Tue 2023-02-14 17:10:53 UTC; 9s ago
 Main PID: 7715 (node_exporter)
    Tasks: 4 (limit: 4627)
   CGroup: /system.slice/node_exporter.service
           └─7715 /usr/local/bin/node_exporter
```

2) Add Node URL to target in prometheus.yml

i. Add slave node URL to Prometheus targets

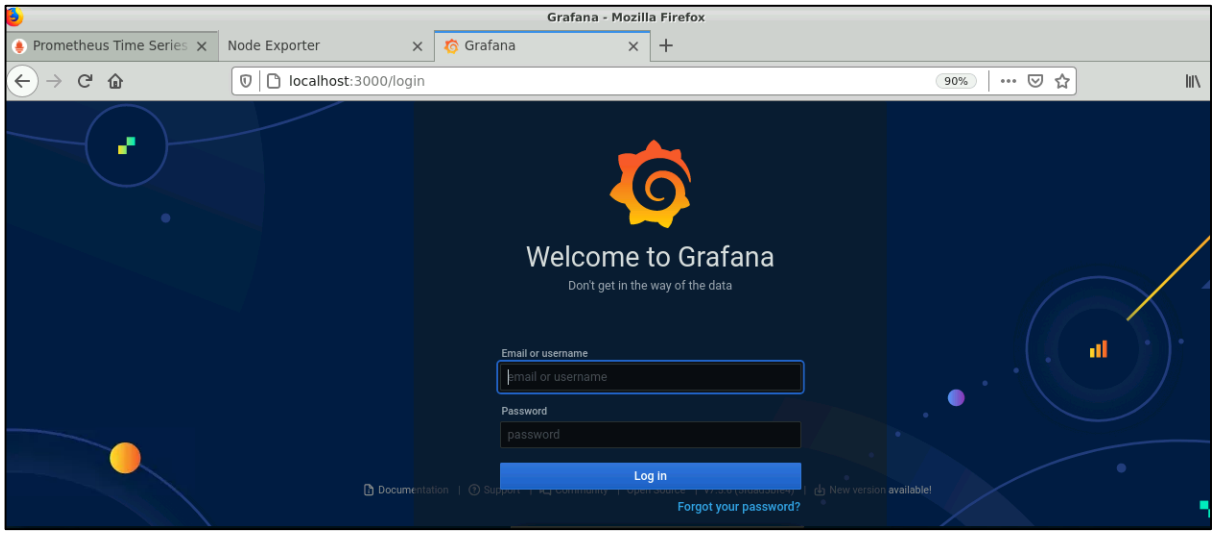
```
# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: 'prometheus'

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

    static_configs:
            - targets: ['localhost:9100','172.31.18.87:9100']
```

ii. Restart the Prometheus service and validate its working fine.

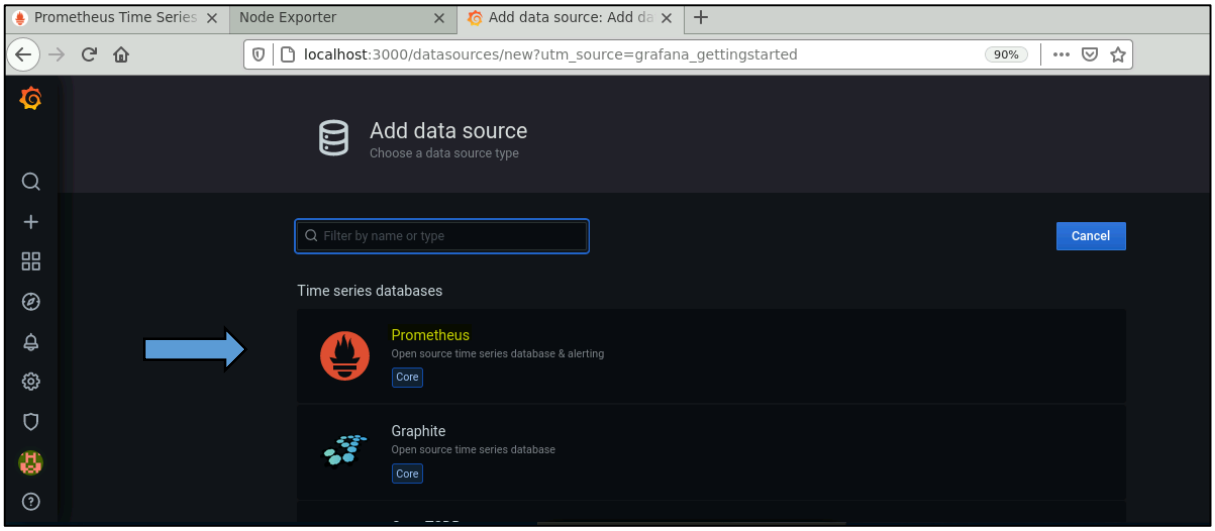


3) Create Dashboard in Grafana

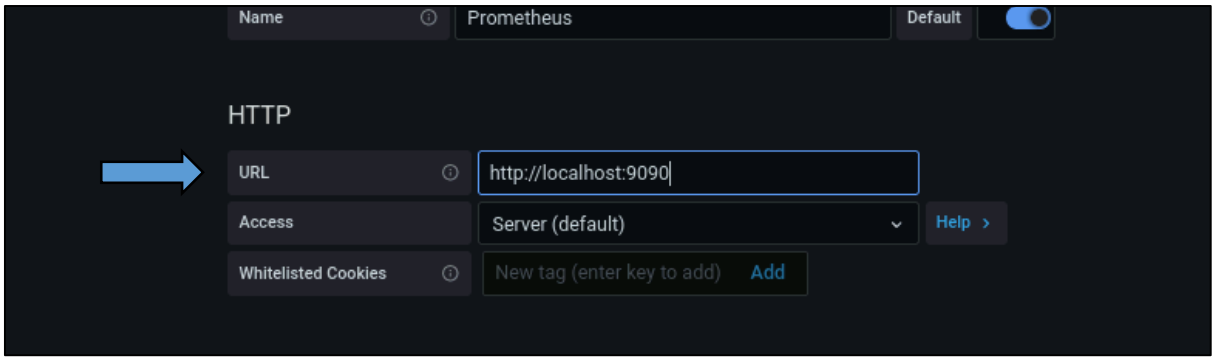
i. Access the Grafana URL on master node: **http://localhost:3000/login**



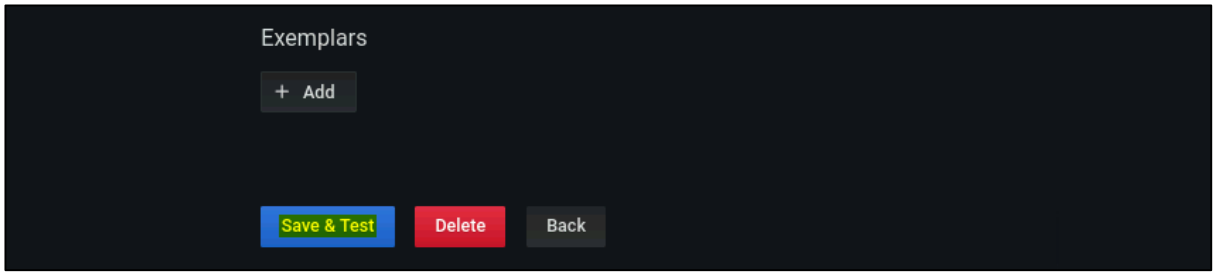
ii. Add a data source as Prometheus

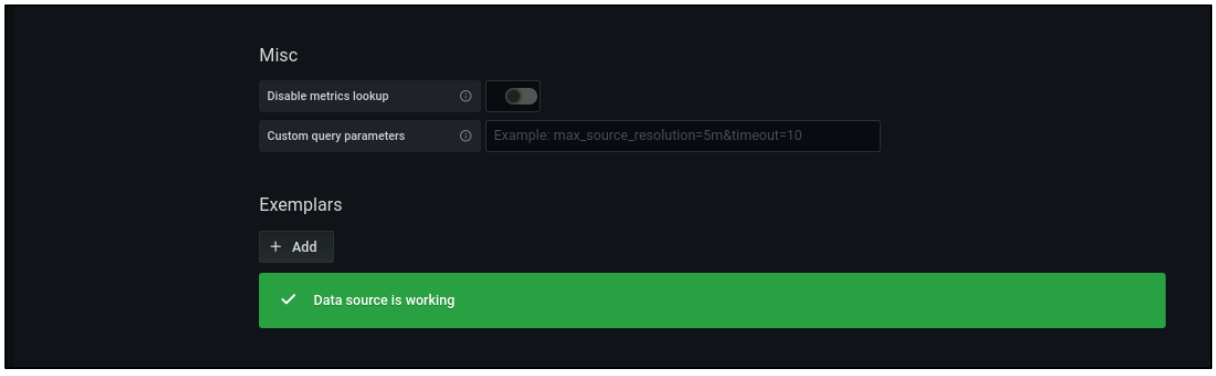


iii. Add Prometheus URL in data source configuration

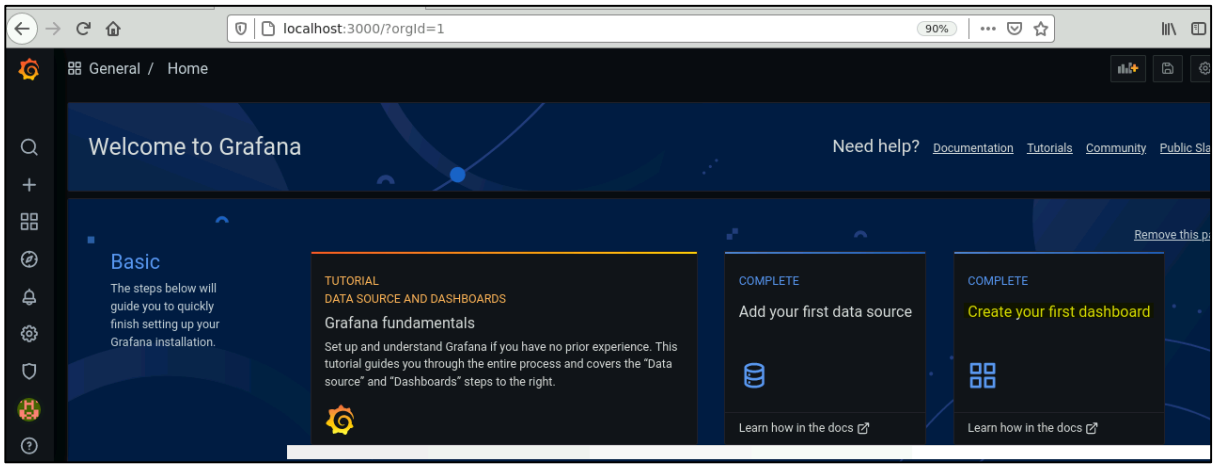


iv. Click on Save & Test

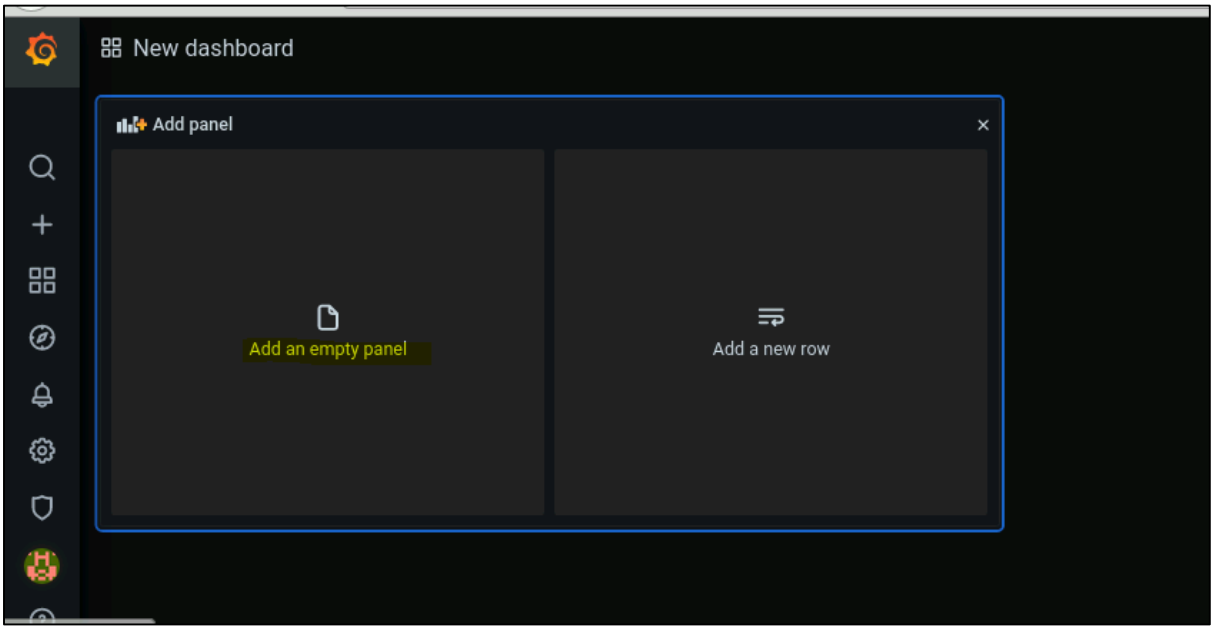




v. Create a new dashboard

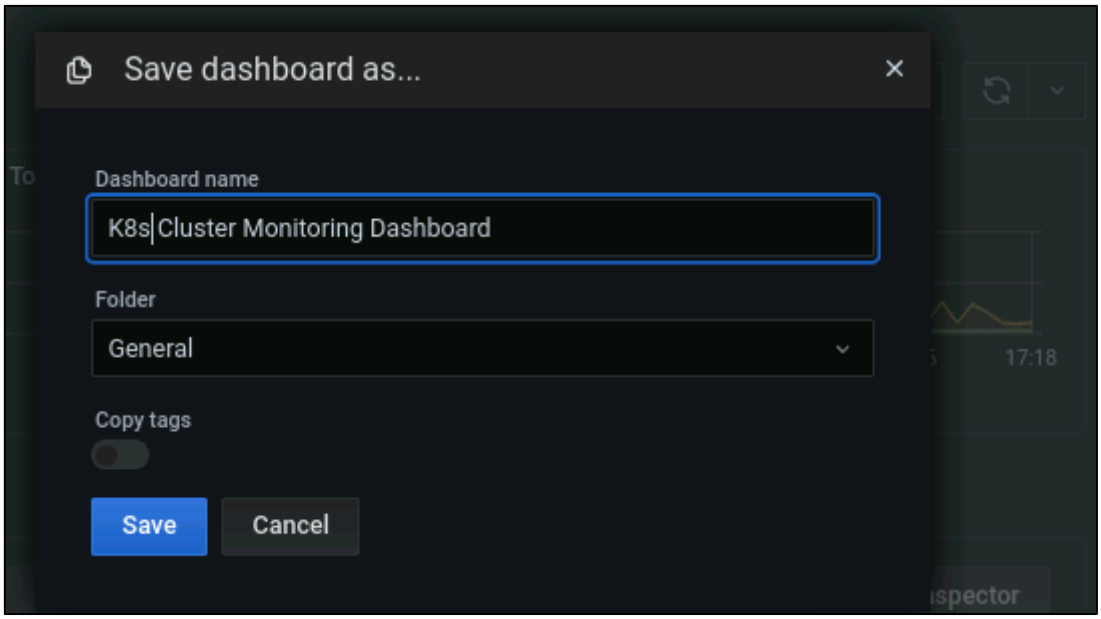


vi. Click on Add an empty panel

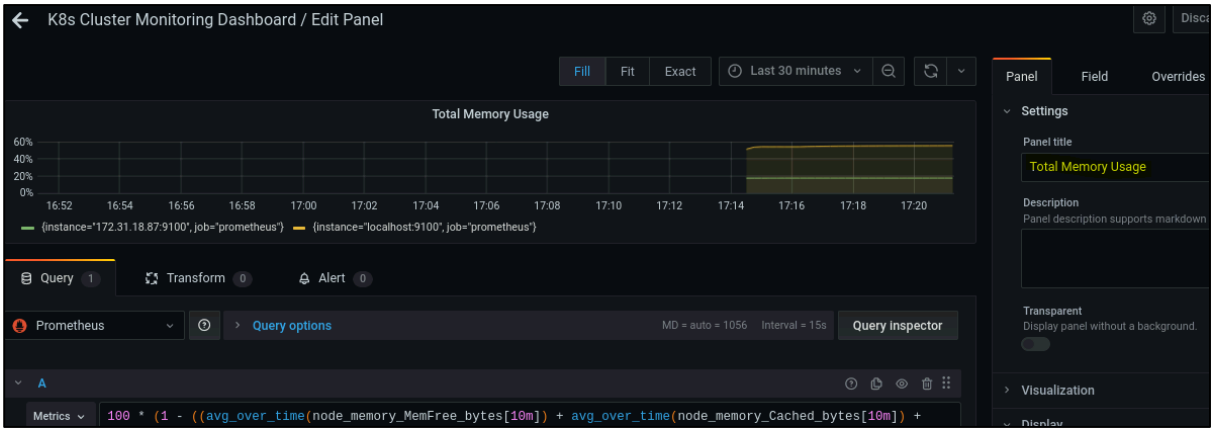


vii. Add metric for CPU utilization

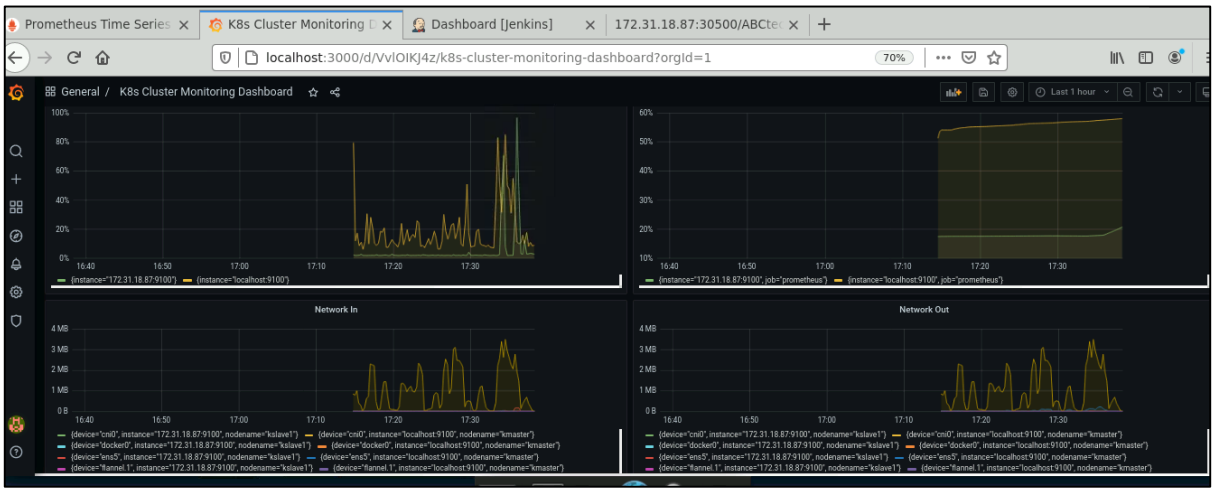
viii. Save the dashboard by giving it a name



ix. Similarly, add more panels and respective metrics to complete the dashboard.



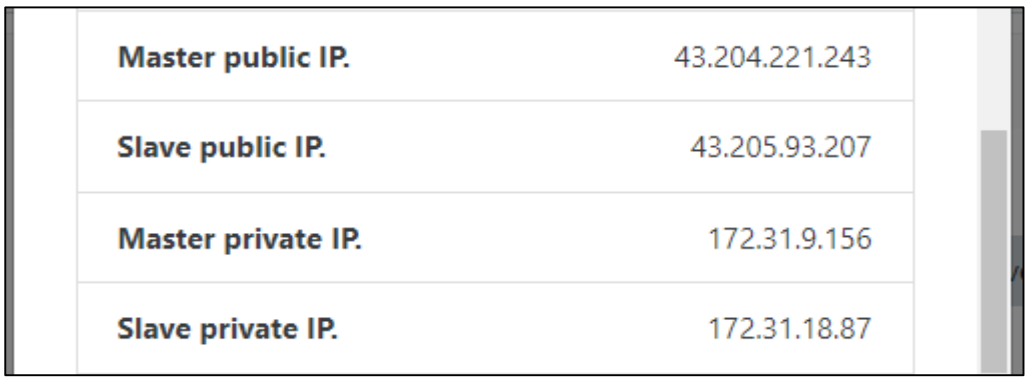
x. Final Grafana Dashboard



Important URLS:

1) GitHub Repo: AkankshaSukre/devops-project1-repo
2) Jenkins: http://localhost:8080/ jenkins/jenkins
3) Dockerhub Repo: akankshasukre/abc-tech
4) Prometheus: http://localhost:9090
5) Node Exporter: http://localhost:9100
6) Grafana: http://localhost:3000
7) URL to access the deployed application: http://172.31.18.87:30500/ABCtechnologies-1.0/

Cloud Lab Details:



Master public IP.	43.204.221.243
Slave public IP.	43.205.93.207
Master private IP.	172.31.9.156
Slave private IP.	172.31.18.87

Also, submitting the files mentioned in this solution document which are named as follows:

1) Dockerfile
2) main.yaml (docker)
3) pod-definition.yaml
4) deployment-definition.yaml
5) service-definition.yaml
6) build-publish.yaml
7) main.yaml (kubernetes)