EXPERIMENT-09

AIM: Automate the process of running containerized application developed in exercise 7 using Kubernetes

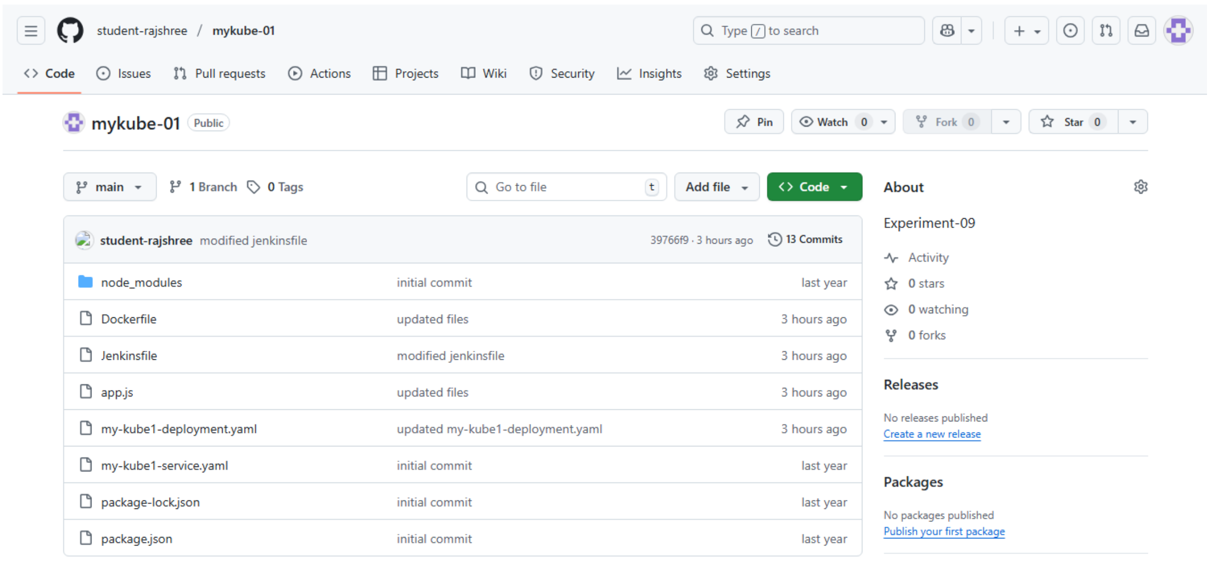
Objective: To automate the process of building, running, and managing containerized applications using Docker and scripting techniques.

Software Requirements

- Docker installed (https://www.docker.com/products/docker-desktop/)
- Kubernetes installed (Minikube or Docker Desktop with Kubernetes enabled)
- kubectl command-line tool
- Code editor (VS Code or any)
- Sample containerized application

Step1:

1. Clone this repository to your local repository https://github.com/shiv4j/kube
2. Make sure that cloned repository consist of "my-kube1-deployment.yaml", "my-kube1-service.yaml" files in folder.
3. Push this local repository to github (or) you can fork that repository from https://github.com/shiv4j/kube
4. After completion of pushing or forking of kube1 folder into your github repository you should able to see like this that contains all the files.



student-rajshree / mykube-01

mykube-01 Public

main · 1 Branch · 0 Tags

student-rajshree modified jenkinsfile · 39766f9 · 3 hours ago · 13 Commits

node_modules	initial commit	last year
Dockerfile	updated files	3 hours ago
Jenkinsfile	modified jenkinsfile	3 hours ago
app.js	updated files	3 hours ago
my-kube1-deployment.yaml	updated my-kube1-deployment.yaml	3 hours ago
my-kube1-service.yaml	initial commit	last year
package-lock.json	initial commit	last year
package.json	initial commit	last year

About

Experiment-09

Activity · 0 stars · 0 watching · 0 forks

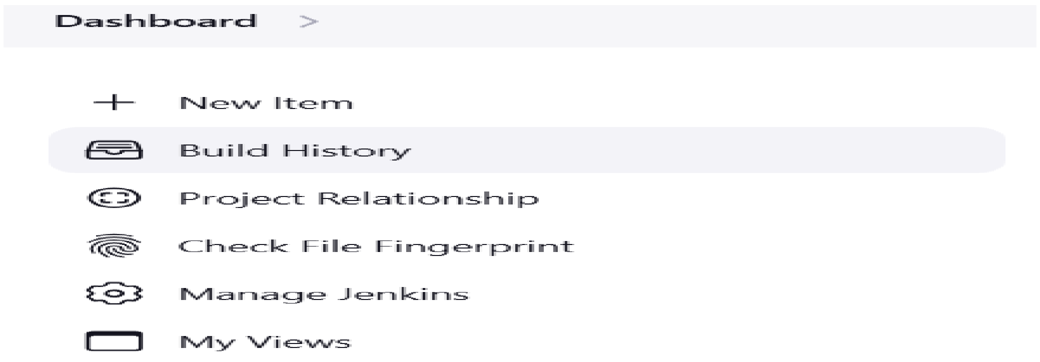
Releases: No releases published · Create a new release

Packages: No packages published · Publish your first package

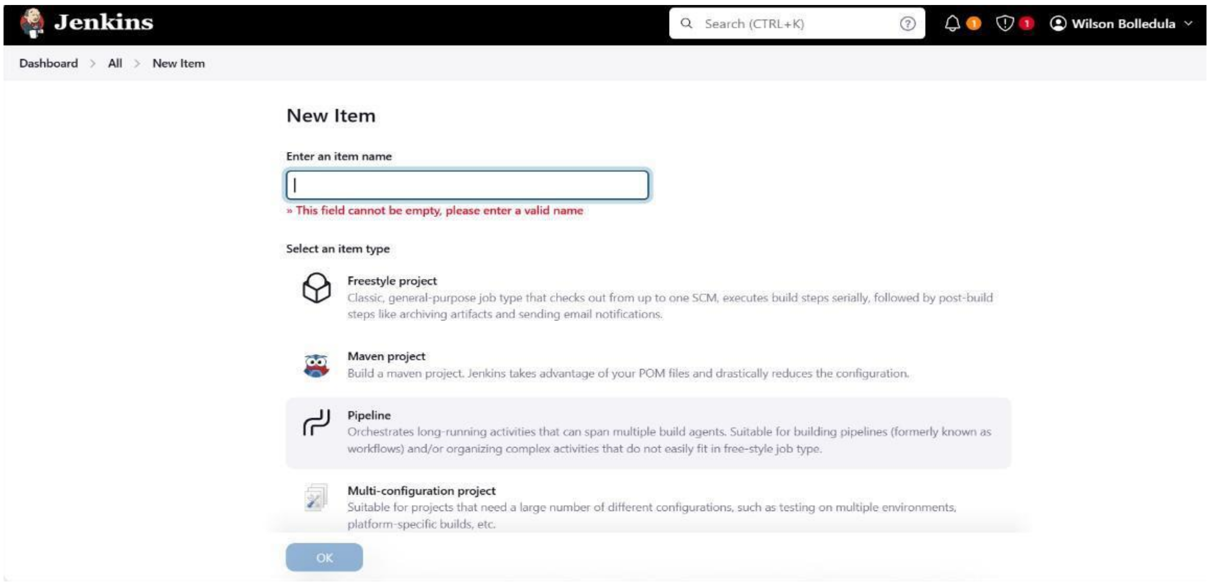
Step2:

Push your github repository to the Jenkins.

Click on New Item



Enter a name and select Pipeline project and click on Ok.



New Item

Enter an item name

» This field cannot be empty, please enter a valid name

Select an item type

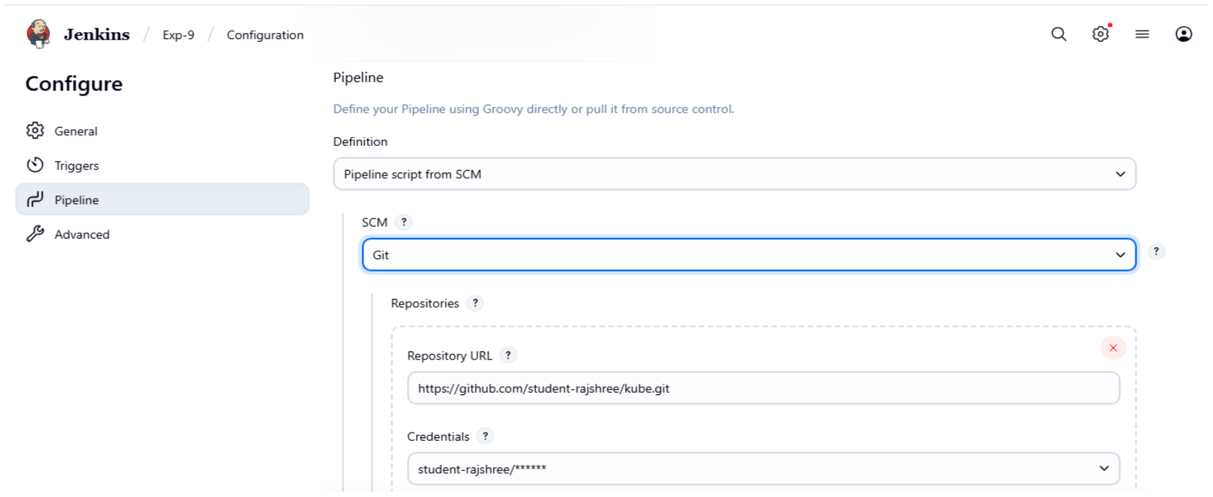
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

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

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

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

- In the configure, go to pipeline tab-> Select Definition as **Pipeline script from SCM-> Select SCM as Git** and paste your repository url
- Specify your branch whether it is main or master based on your github repository.
- Click on Apply and Save.


Jenkins / Exp-9 / Configuration

Configure

- General
- Triggers
- Pipeline
- Advanced

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

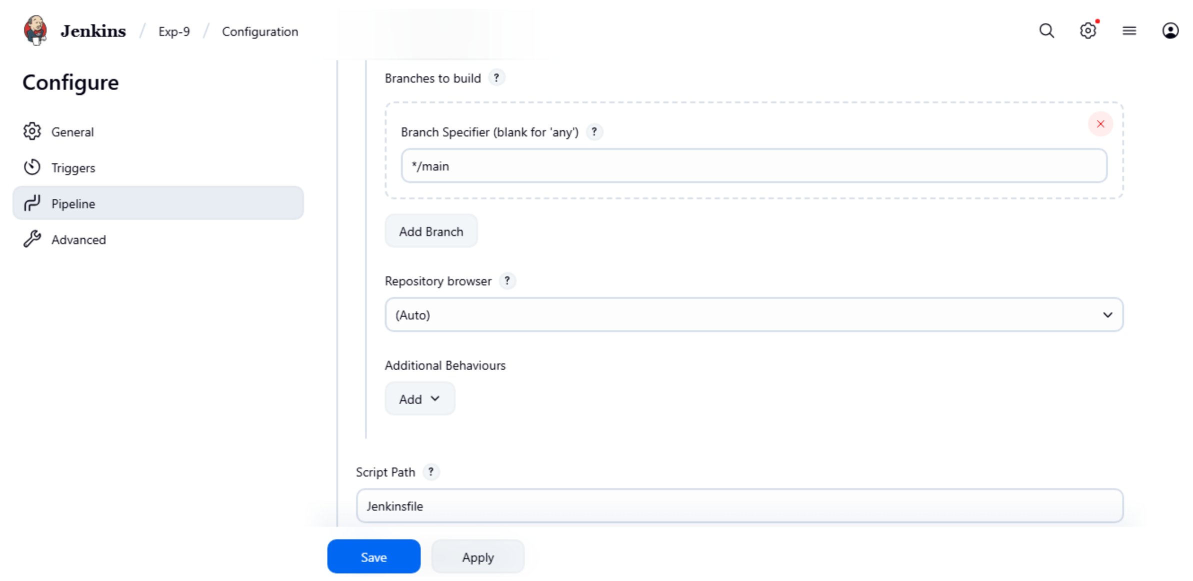
Definition: Pipeline script from SCM

SCM: Git

Repositories

Repository URL: https://github.com/student-rajshree/kube.git

Credentials: student-rajshree/******


Jenkins / Exp-9 / Configuration

Configure

- General
- Triggers
- Pipeline
- Advanced

Branches to build

Branch Specifier (blank for 'any'): */main

Add Branch

Repository browser: (Auto)

Additional Behaviours

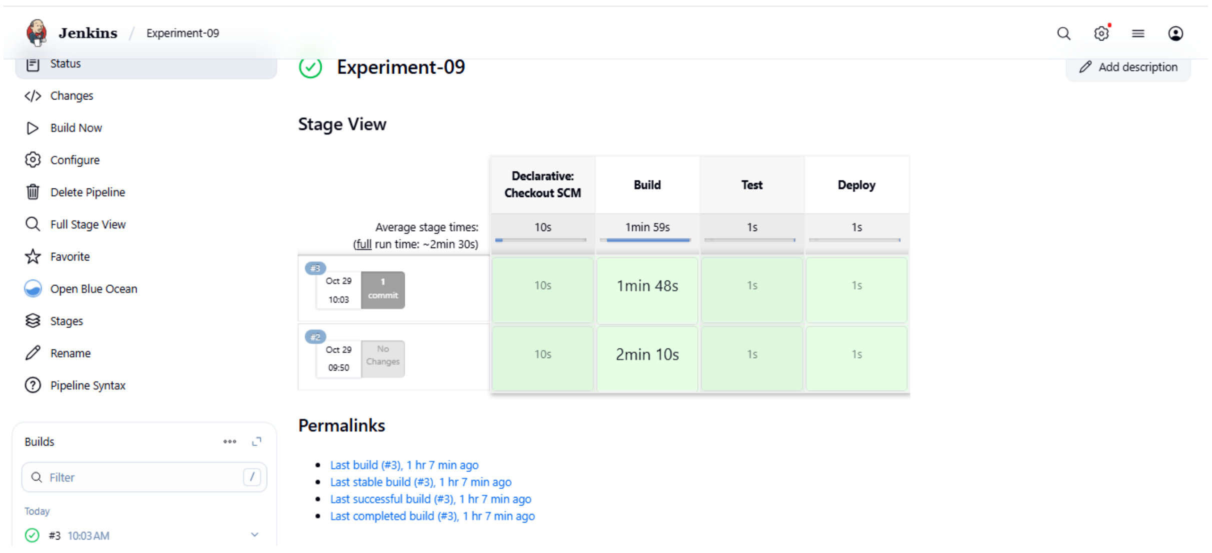
Add

Script Path: Jenkinsfile

Save Apply

- After creation of your Jenkins project build it-> Click on **Build Now**

The build should be shown in green tick mark.


Jenkins / Experiment-09

- Status
- Changes
- Build Now
- Configure
- Delete Pipeline
- Full Stage View
- Favorite
- Open Blue Ocean
- Stages
- Rename
- Pipeline Syntax

Experiment-09

Add description

Stage View

	Declarative: Checkout SCM	Build	Test	Deploy
Average stage times: (full run time: ~2min 30s)	10s	1min 59s	1s	1s
#3 Oct 29 10:03 1 commit	10s	1min 48s	1s	1s
#2 Oct 29 09:50 No Changes	10s	2min 10s	1s	1s

Permalinks

- Last build (#3), 1 hr 7 min ago
- Last stable build (#3), 1 hr 7 min ago
- Last successful build (#3), 1 hr 7 min ago
- Last completed build (#3), 1 hr 7 min ago

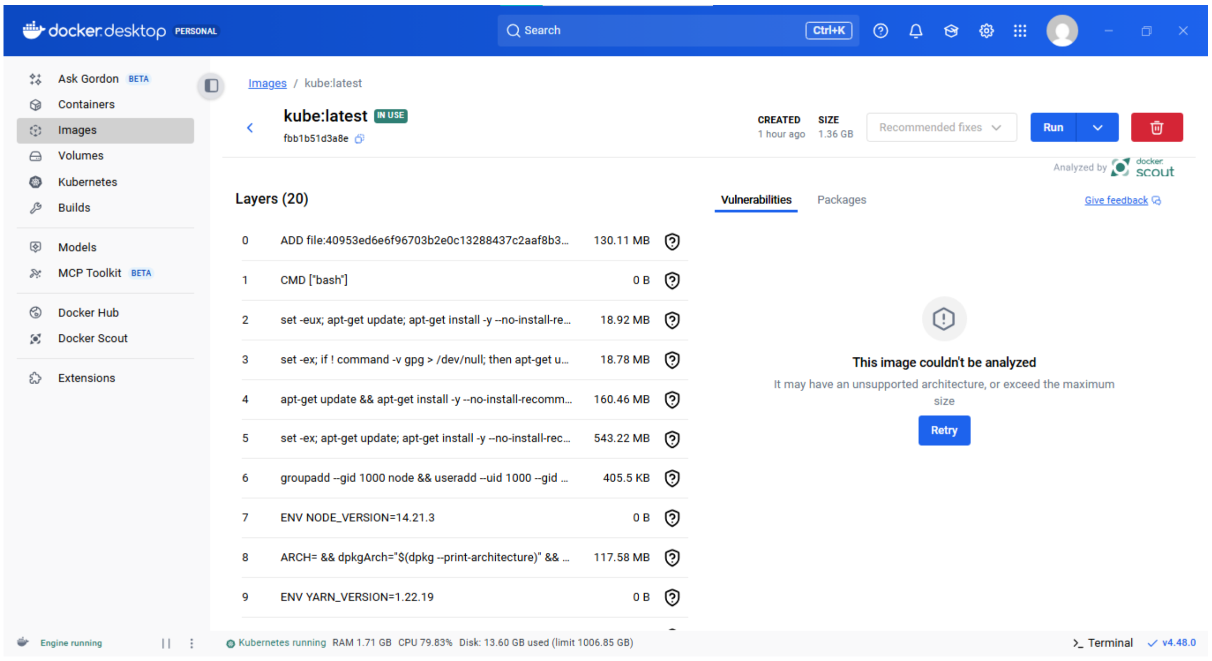
Builds

Filter

Today

#3 10:03 AM

You will get docker image for this project, like showed in the below



kube:latest IN USE
fbb1b51d3a8e

CREATED 1 hour ago SIZE 1.36 GB

Layers (20)

0	ADD file:40953ed6e6f96703b2e0c13288437c2aaf8b3...	130.11 MB
1	CMD ["bash"]	0 B
2	set -eux; apt-get update; apt-get install -y --no-install-re...	18.92 MB
3	set -ex; if ! command -v gpg > /dev/null; then apt-get u...	18.78 MB
4	apt-get update && apt-get install -y --no-install-recomm...	160.46 MB
5	set -ex; apt-get update; apt-get install -y --no-install-rec...	543.22 MB
6	groupadd --gid 1000 node && useradd --uid 1000 --gid ...	405.5 KB
7	ENV NODE_VERSION=14.21.3	0 B
8	ARCH= && dpkgArch="$(dpkg --print-architecture)" && ...	117.58 MB
9	ENV YARN_VERSION=1.22.19	0 B

This image couldn't be analyzed
It may have an unsupported architecture, or exceed the maximum size

Kubernetes running RAM 1.71 GB CPU 79.83% Disk: 13.60 GB used (limit 1006.85 GB)

Step3:

- Push the docker image into dockerhub
- open command prompt and run the command “docker login”
- Tag your iamge using this syntax

docker tag <local-image-name>:<tag> yourusername/image-name:<tag>

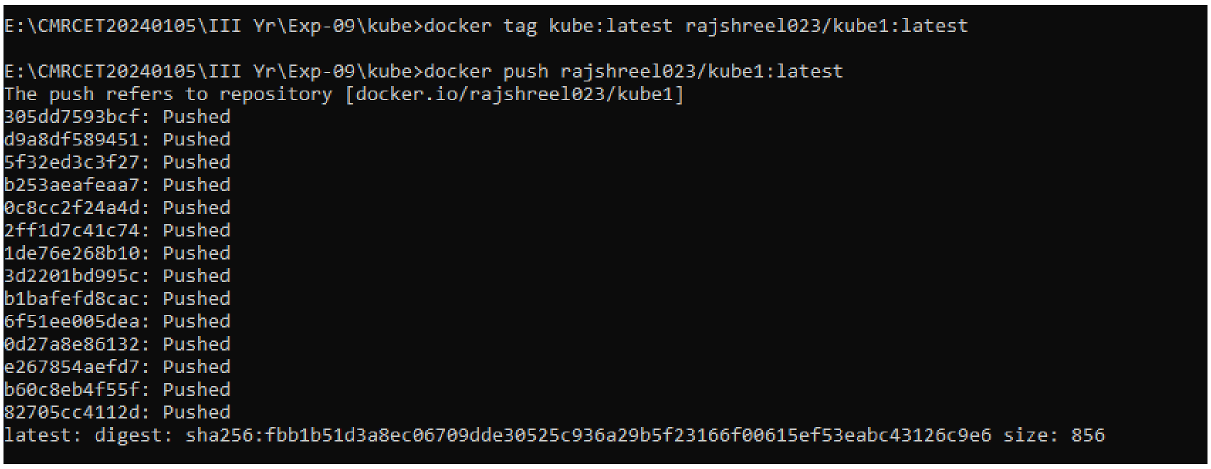
Ex: docker tag kube:latest rajshreel023/kube1:latest

Here username is your dockerhub account username

- Push the image to dockerhub

docker push yourusername/image-name:<tag>

Ex: docker push rajshreel023/kube1:latest



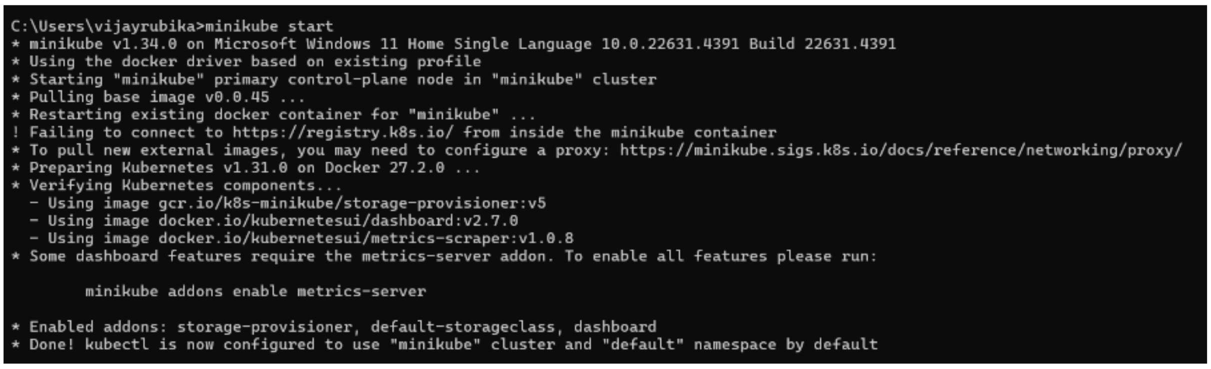
```
E:\CMRCET20240105\III Yr\Exp-09\kube>docker tag kube:latest rajshreel023/kube1:latest

E:\CMRCET20240105\III Yr\Exp-09\kube>docker push rajshreel023/kube1:latest
The push refers to repository [docker.io/rajshreel023/kube1]
305dd7593bcf: Pushed
d9a8df589451: Pushed
5f32ed3c3f27: Pushed
b253aeafeaa7: Pushed
0c8cc2f24a4d: Pushed
2ff1d7c41c74: Pushed
1de76e268b10: Pushed
3d2201bd995c: Pushed
b1bafefd8cac: Pushed
6f51ee005dea: Pushed
0d27a8e86132: Pushed
e267854aefd7: Pushed
b60c8eb4f55f: Pushed
82705cc4112d: Pushed
latest: digest: sha256:fbb1b51d3a8ec06709dde30525c936a29b5f23166f00615ef53eabc43126c9e6 size: 856
```

Step4:

Start the Kubernetes

Syntax: minikube start



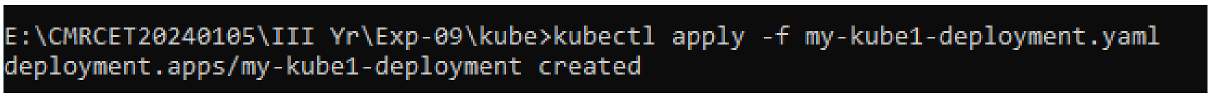
```
C:\Users\vijayrubika>minikube start
* minikube v1.34.0 on Microsoft Windows 11 Home Single Language 10.0.22631.4391 Build 22631.4391
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* Restarting existing docker container for "minikube" ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
  - Using image docker.io/kubernetesui/dashboard:v2.7.0
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
* Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server

* Enabled addons: storage-provisioner, default-storageclass, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

- Apply my-kube1-deployment.yaml file

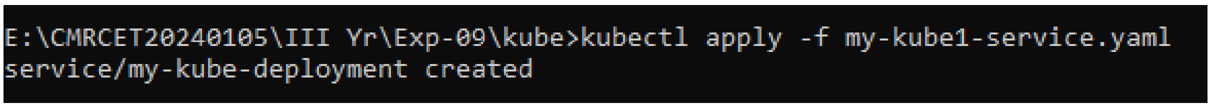
kubectl apply -f my-kube1-deployment.yaml



```
E:\CMRCET20240105\III Yr\Exp-09\kube>kubectl apply -f my-kube1-deployment.yaml
deployment.apps/my-kube1-deployment created
```

- Apply my-kube1-service.yaml file

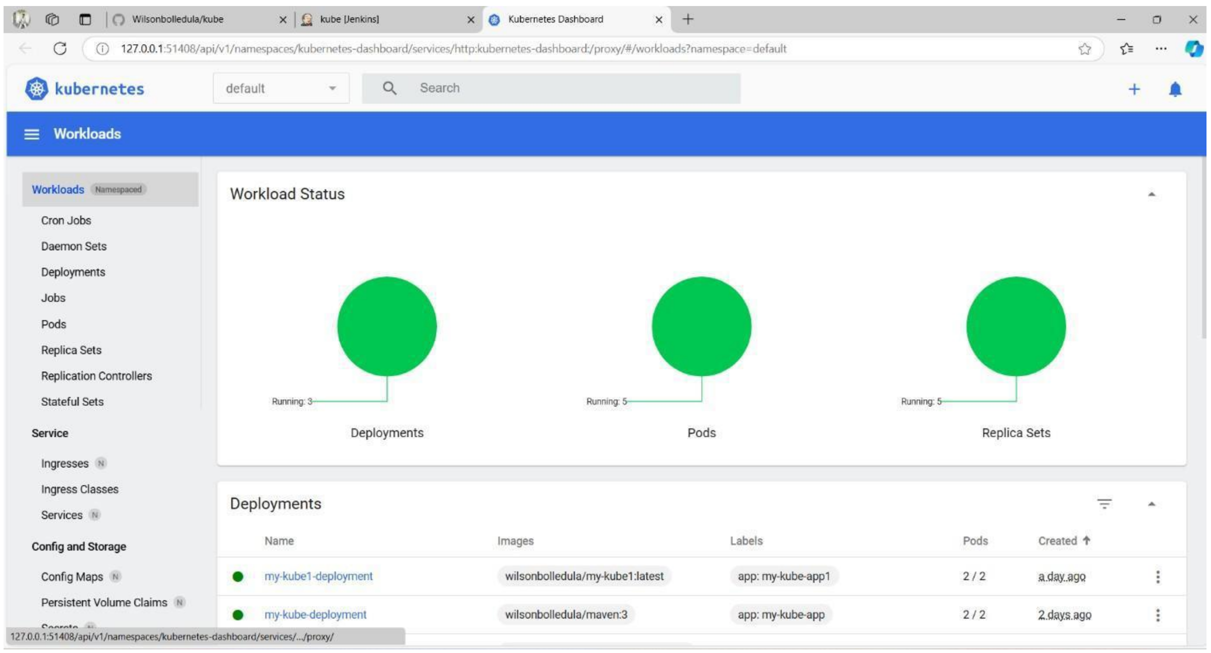
Syntax: **kubectl apply -f my-kube1-service.yaml**



```
E:\CMRCET20240105\III Yr\Exp-09\kube>kubectl apply -f my-kube1-service.yaml
service/my-kube-deployment created
```

That will apply to the deployments and services

- To check that type command “**kubectl get pods**” and “**kubectl get service**”
- To open the Kubernetes dashboard type = “minikube dashboard”
  That will open Kubernetes dashboard automatically on your default primary browser



	Name	Images	Labels	Pods	Created ↑
●	my-kube1-deployment	wilsonbolledula/my-kube1:latest	app: my-kube-app1	2 / 2	a day ago
●	my-kube-deployment	wilsonbolledula/maven:3	app: my-kube-app	2 / 2	2 days ago

127.0.0.1:51408/api/v1/namespaces/kubernetes-dashboard/services/.../proxy/

- Finally you can checkout your deployments and pods here.