**Deploying a Python Web Application on Kubernetes Cluster for Warner Sisters Inc- Capstone Project**

**Creating Python Django Web Application:  
  
mywebapp/**

**├── manage.py**

**├── urls.py**

**│── views.py**

**├── mywebapp/**

**│ ├── \_\_init\_\_.py**

**│ ├── asgi.py**

**│ ├── settings.py**

**│ ├── urls.py**

**│ └── wsgi.py**

**pip install django**

**django-admin startproject mywebapp**

**Views.py**from django.http import HttpResponse

def hello(request):

return HttpResponse("Hello, World!")

**myapp/urls.py**from django.urls import path

from . import views

urlpatterns = [

path('hello/', views.hello, name='hello'),

]

**mywebapp/urls.py**

**f**rom django.contrib import admin

from django.urls import include, path

urlpatterns = [

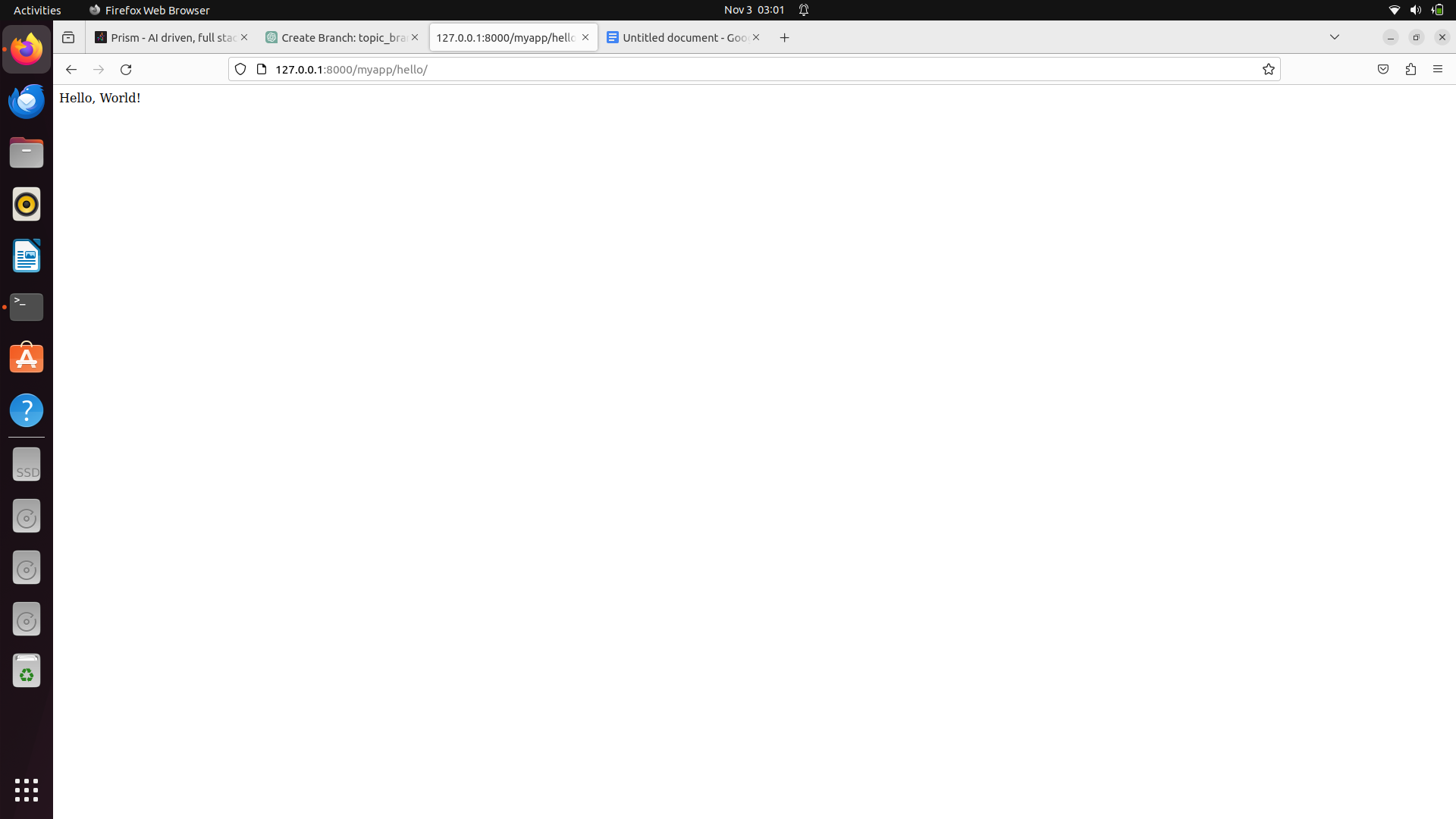
path('admin/', admin.site.urls),

path('myapp/', include('myapp.urls')), # Include the app's URLs

]

**python manage.py runserver**

**http://127.0.0.1:8000/myapp/hello/**

****

**Containerize Python Django Web Application  
  
Touch dockerfile**

# Use an official Python runtime as the base image

FROM python:3.9

# Set the working directory in the container

WORKDIR /app

# Copy the requirements.txt file into the container

COPY requirements.txt /app/

# Install the dependencies

RUN pip install -r requirements.txt

# Copy the entire project into the container

COPY . /app/

# Expose the port your application runs on

EXPOSE 8000

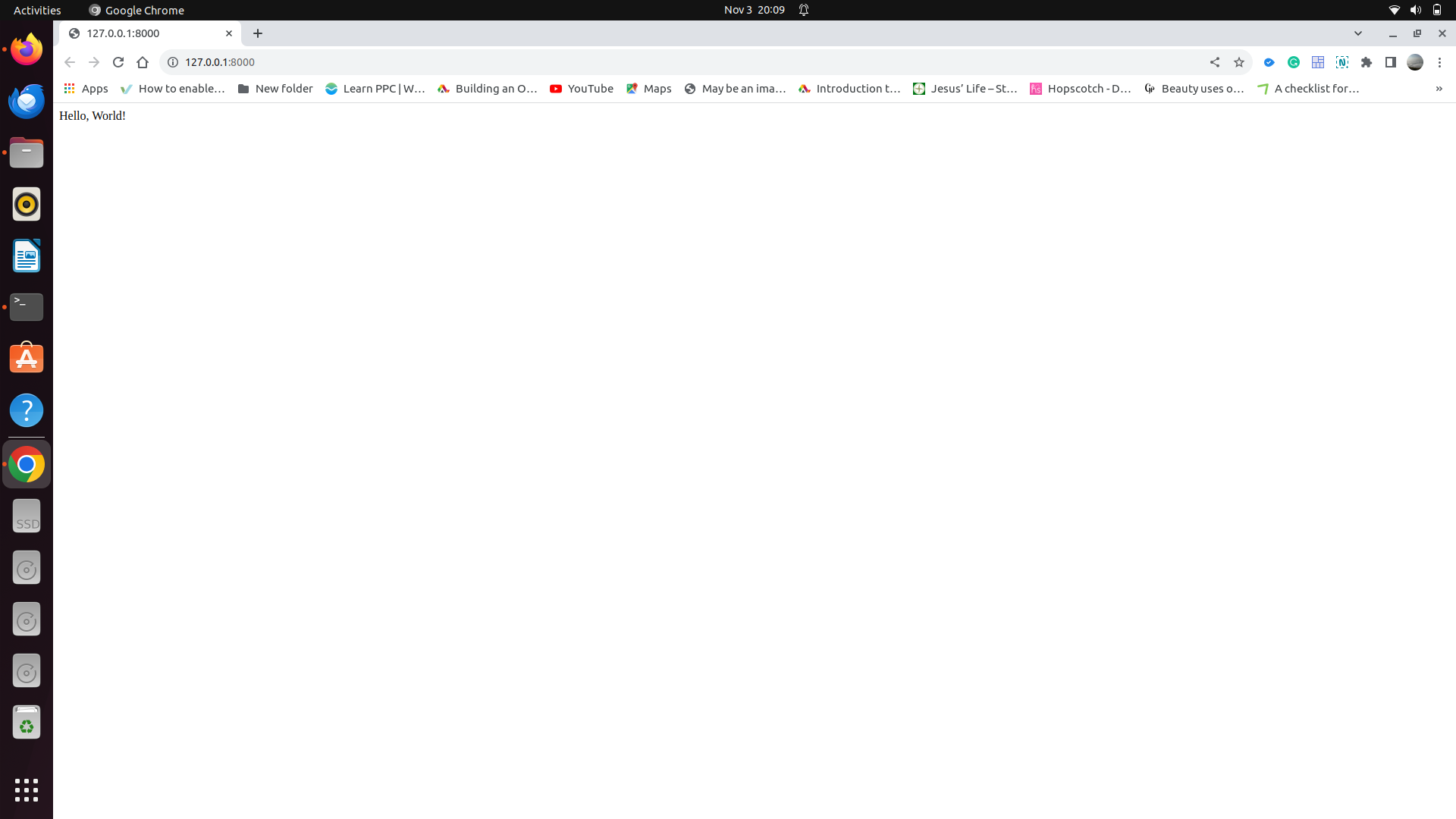
# Define the command to run your application

CMD ["gunicorn", "mywebapp.wsgi:application", "-b", "0.0.0.0:8000"]

**Touch requirements.txt**  
  
Django==3.2.5

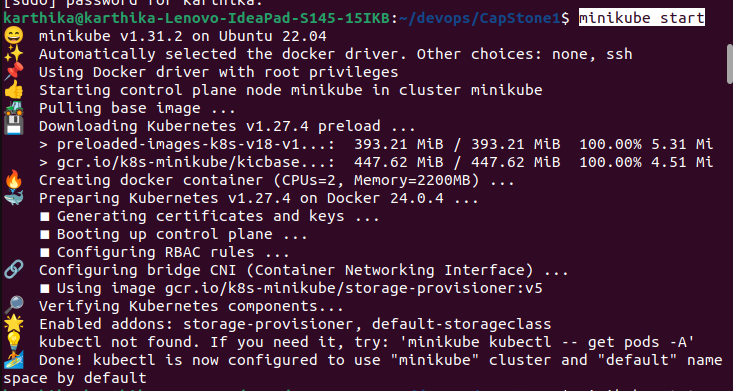
gunicorn==20.1.0

**python3 manage.py runserver**

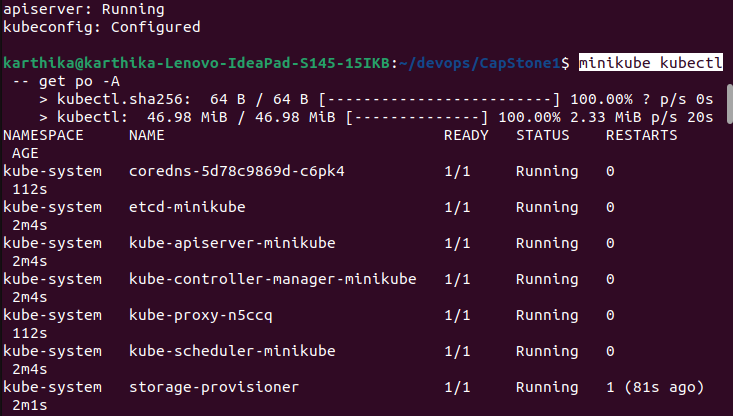


**Set Up Minikube Kubernetes Cluster:  
url -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd6464**

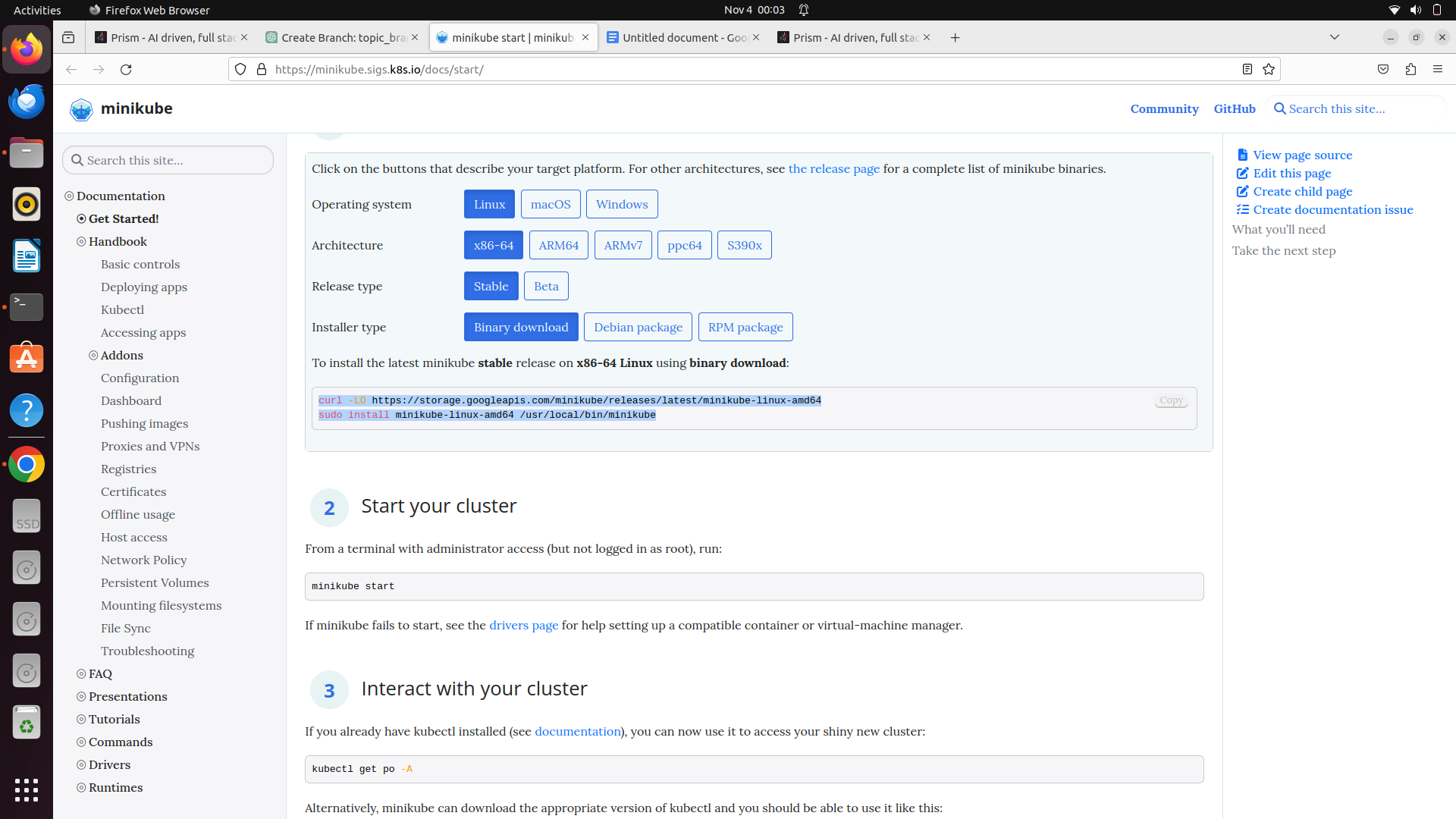
**sudo install minikube-linux-amd64 /usr/local/bin/minikube  
minikube start**

**  
minikube status**

**minikube kubectl**

****

**minikube dashboard**

****

**minikube start  
  
sudo snap install kubectl  
kubectl get nodes**

**mkdir kubernetes**

**cd kubernetes**

**touch myapp\_deployment.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: myapp-deployment

spec:

replicas: 3

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec:

containers:

- name: myapp-container

image: your-dockerhub-username/my-django-app:latest

ports:

- containerPort: 8000

**touch myapp\_service.yaml**apiVersion: v1

kind: Service

metadata:

name: myapp-service

spec:

selector:

app: myapp

ports:

- protocol: TCP

port: 80

targetPort: 8000

type: LoadBalancer

**touch myapp\_ingress.yaml**

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: myapp-ingress

spec:

rules:

- host: myapp.example.com

http:

paths:

- pathType: Prefix

path: /

backend:

service:

name: myapp-service

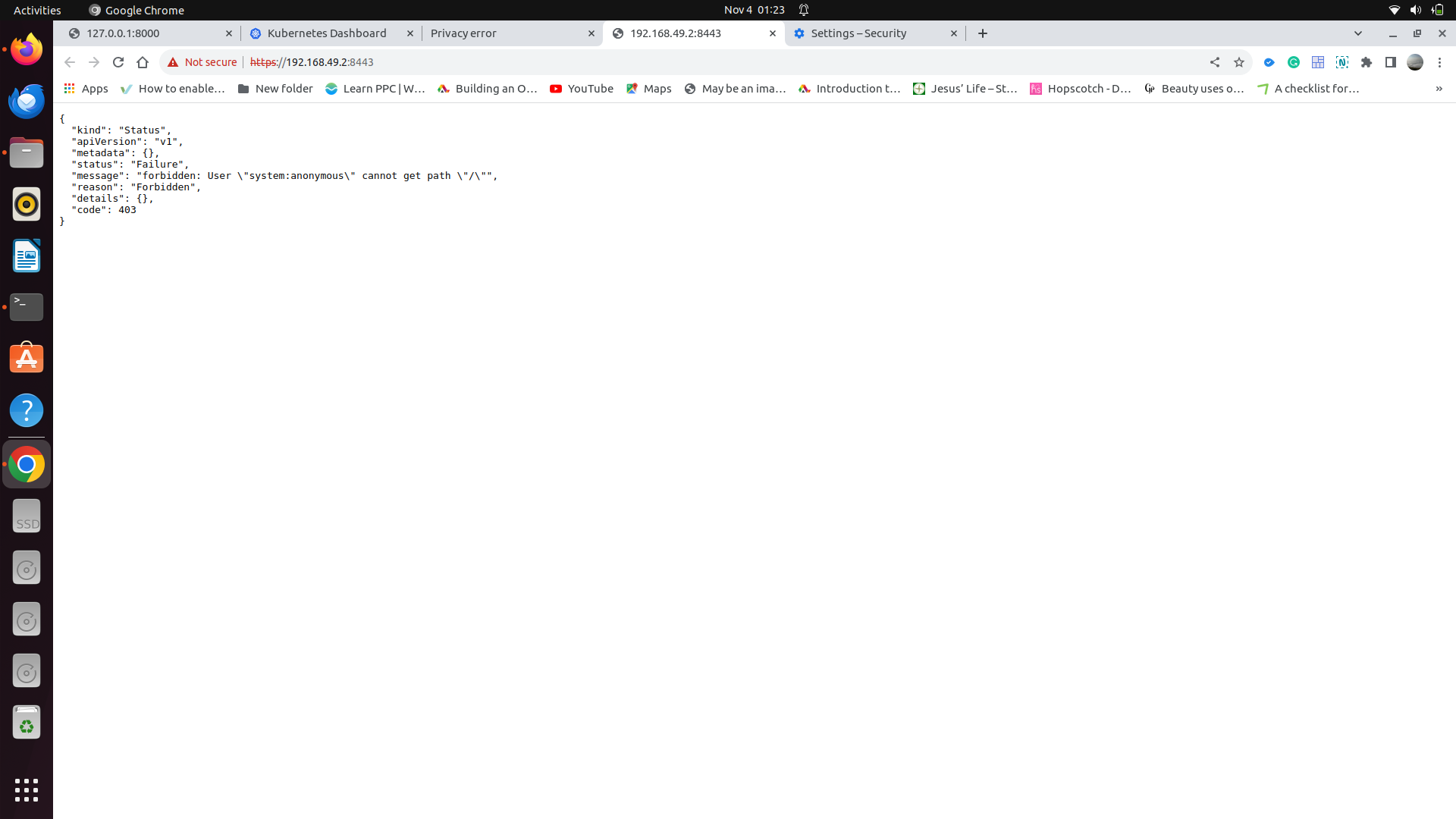
port:

number: 80

**touch db\_deployment.yaml**# Database Deployment YAML

**kubectl apply -f myapp\_deployment.yaml**

**kubectl cluster-info**

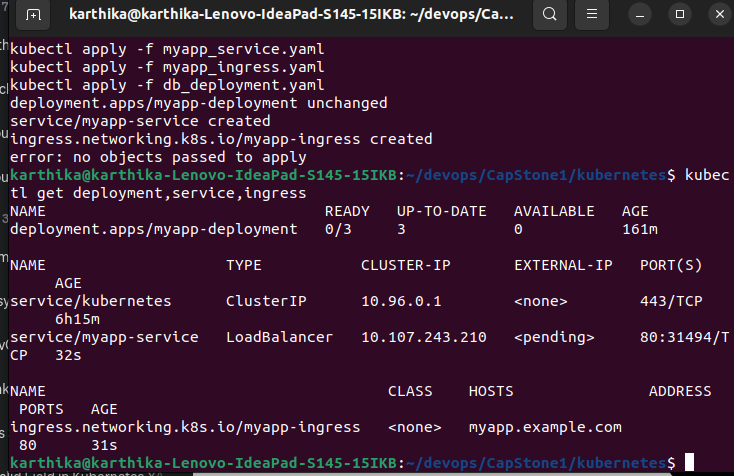
****

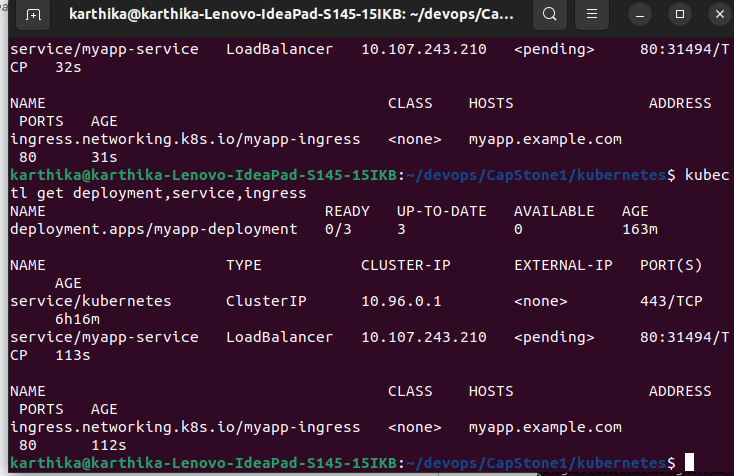
**kubectl apply -f myapp\_deployment.yaml**

**kubectl apply -f myapp\_service.yaml**

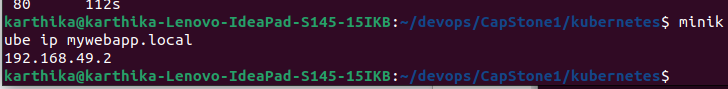
**kubectl apply -f myapp\_ingress.yaml**

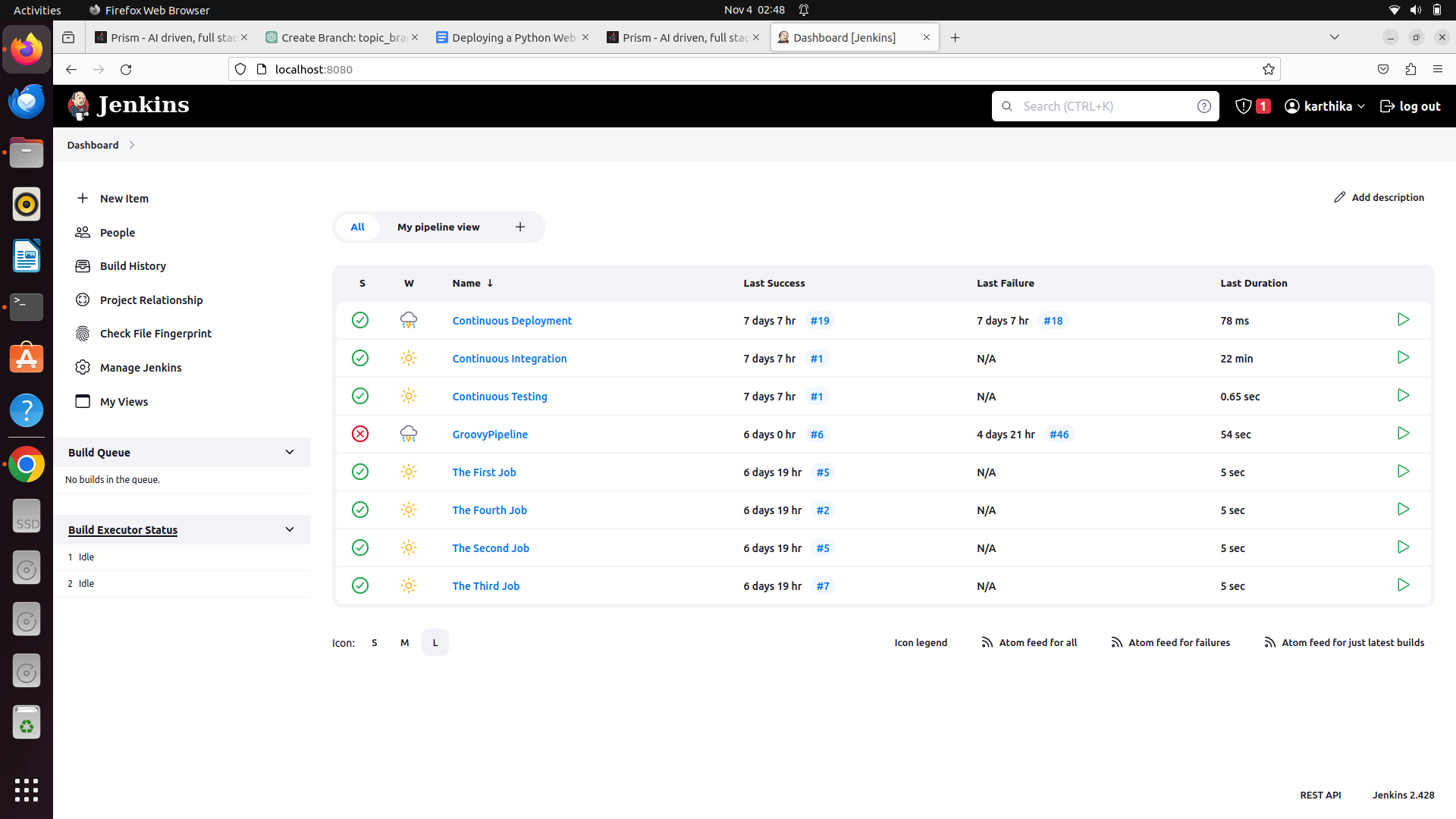
**kubectl apply -f db\_deployment.yaml**

****

**kubectl get deployment,service,ingress  
**

**minikube ip mywebapp.local**

****

**Implement Jenkins CI/CD Pipeline:  
**

**git remote set-url origin** [**https://karthika-pankaj:github\_pat\_11ASVPDAY0tu7LflIX7Up2\_v91F2IZc6bbkbnEzcoyB6Kg9HpGoJ1zLxrGQYYNwW0aGNCHMRJ4RLu3Z6Q5@github.com/savinderpuri/knowledgehut.git**](https://karthika-pankaj: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@github.com/savinderpuri/knowledgehut.git) **git remote set-url origin git@github.com:karthika-pankaj/Java\_maven.git**

**git push -u origin master**

**Enumerating objects: 32, done.**

**Counting objects: 100% (32/32), done.**

**Delta compression using up to 4 threads**

**Compressing objects: 100% (28/28), done.**

**Writing objects: 100% (32/32), 34.72 MiB | 296.00 KiB/s, done.**

**Total 32 (delta 4), reused 0 (delta 0), pack-reused 0**

**remote: Resolving deltas: 100% (4/4), done.**

**remote: warning: See https://gh.io/lfs for more information.**

**remote: warning: File minikube-linux-amd64 is 82.43 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB**

**remote: warning: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.**

**remote:**

**remote: Create a pull request for 'master' on GitHub by visiting:**

**remote: https://github.com/karthika-pankaj/Java\_maven/pull/new/master**

**remote:**

**To github.com:karthika-pankaj/Java\_maven.git**

**\* [new branch] master -> master**

**Branch 'master' set up to track remote branch 'master' from 'origin'.**

**minikube start  
minikube addons enable metrics-server  
kubectl create namespace monitoring**

**kubectl config current-context  
  
kubectl apply -f https://raw.githubusercontent.com/prometheus-operator/prometheus-operator/main/bundle.yaml -n monitoring  
kubectl apply -f https://raw.githubusercontent.com/prometheus-operator/grafana-operator/main/examples/default.yaml -n monitoring**