High-Level Plan

1. Launch Ubuntu EC2 instance
2. Install required tools (awscli, kubectl, kops, terraform, docker)
3. Create infrastructure with Terraform (S3 bucket, DNS zone setup)
4. Use Kops to create the cluster
5. Build & push Docker image of a Python Flask app
6. Deploy Flask app to Kubernetes
7. Access your app via browser

🚀 Step-by-Step Instructions

1. 🔧 Launch an Ubuntu EC2 Instance

* AMI: Ubuntu 22.04 LTS
* Instance Type: t3.medium
* Add IAM role with AdministratorAccess
* Allow security group for:
  + SSH (port 22)
  + HTTP (port 80)
  + HTTPS (port 443)

2. 📦 Install Required Tools

SSH into your EC2 instance and run:

# Update system

sudo apt update && sudo apt upgrade -y

# Install AWS CLI

sudo apt install awscli -y

# Install kubectl

curl -LO "https://dl.k8s.io/release/$(curl -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

chmod +x kubectl

sudo mv kubectl /usr/local/bin/

# Install kops

curl -LO https://github.com/kubernetes/kops/releases/download/v1.28.0/kops-linux-amd64

chmod +x kops-linux-amd64

sudo mv kops-linux-amd64 /usr/local/bin/kops

# Install Terraform

sudo apt install unzip -y

wget https://releases.hashicorp.com/terraform/1.8.0/terraform\_1.8.0\_linux\_amd64.zip

unzip terraform\_1.8.0\_linux\_amd64.zip

sudo mv terraform /usr/local/bin/

Verify installations:

bash

CopyEdit

aws --version

kubectl version --client

kops version

terraform version

3. 🏗️ Terraform Project to Setup S3 for Kops

📁 Project Structure

css

CopyEdit

kops-cluster/

├── main.tf

├── variables.tf

├── outputs.tf

✏️ variables.tf

h

CopyEdit

variable "region" { default = "us-east-1" }

variable "bucket\_name" { default = "kops-state-miyasa-shop-123456" }

✏️ main.tf

hcl

CopyEdit

provider "aws" {

region = var.region

}

resource "aws\_s3\_bucket" "kops" {

bucket = var.bucket\_name

acl = "private"

}

resource "aws\_s3\_bucket\_versioning" "versioning" {

bucket = aws\_s3\_bucket.kops.id

versioning\_configuration {

status = "Enabled"

}

}

✏️ outputs.tf

hcl

CopyEdit

output "bucket\_name" {

value = aws\_s3\_bucket.kops.bucket

}

🔨 Run Terraform

bash

CopyEdit

terraform init

terraform apply -auto-approve

4. 🌐 Set Env Vars and Create Cluster with Kops

bash

CopyEdit

export KOPS\_CLUSTER\_NAME=cluster.miyasa.shop

export KOPS\_STATE\_STORE=s3://kops-state-miyasa-shop-123456

Create the cluster:

bash

CopyEdit

kops create cluster \

--cloud=aws \

--zones=us-east-1a,us-east-1b \

--name=${KOPS\_CLUSTER\_NAME} \

--state=${KOPS\_STATE\_STORE} \

--node-count=2 \

--node-size=t3.medium \

--master-size=t3.medium \

--dns-zone=miyasa.shop \

--yes

⚠️ Wait 5–10 mins. Then check:

bash

CopyEdit

kops validate cluster

5. 🐍 Python Flask App with Docker

📁 Flask App Structure

CopyEdit

flask-app/

├── app.py

├── Dockerfile

✏️ app.py

python

CopyEdit

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello():

return "Hello from Kubernetes!"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=80)

✏️ Dockerfile

Dockerfile

CopyEdit

FROM python:3.9-slim

WORKDIR /app

COPY app.py .

RUN pip install flask

EXPOSE 80

CMD ["python", "app.py"]

🐳 Build and Push Docker Image

bash

CopyEdit

# Log in to Docker

docker login

# Build and push image

docker build -t yourdockerhub/flask-k8s-app:latest .

docker push yourdockerhub/flask-k8s-app:latest

6. 🚀 Deploy Flask App on Kubernetes

📁 flask-k8s-deploy.yaml

apiVersion: v1

kind: Service

metadata:

name: flask-service

spec:

selector:

app: flask-app

ports:

- port: 80

targetPort: 80

type: LoadBalancer

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: flask-app

spec:

replicas: 2

selector:

matchLabels:

app: flask-app

template:

metadata:

labels:

app: flask-app

spec:

containers:

- name: flask-container

image: yourdockerhub/flask-k8s-app:latest

ports:

- containerPort: 80

🔧 Apply it

bash

CopyEdit

kubectl apply -f flask-k8s-deploy.yaml

🌍 Get LoadBalancer URL

bash

CopyEdit

kubectl get svc

Look for EXTERNAL-IP of flask-service. Open in your browser.

🎉 You're Done!

* Flask app is running in a scalable Kubernetes cluster
* DNS + Kops + EC2 + Terraform all integrated
* Highly available setup

Would you like a downloadable ZIP of the whole project with pre-filled values (miyasa.shop, DockerHub username, etc.)?