Digital Egypt Pioneers

CLOUD DEVOPS PROJECT

Digital Egypt Pioneers

TRAINING COMPANY

ACICT - AST

GROUP

DEVOPS ENGINEER_SHR1_SWD1_G1E

Digital Egypt Pioneers

PRESENTED BY

ABDELAZIZ ABDELFATTAH ABDELAZIZ ABDELFATTAH

ABDULLAH AHMED SAAD ELSHARKAWI

ELHUSSEIN SHAABAN ABDELHAMIED

IBRAHIM GAMAL IBRAHIM ABDELSAMEEA

SALMA MOHAMED FATHY ELNAGGAR

INSTRUCTOR & SUPERVISOR

MAHMOUD MOHAMED ABDELWAHAB ELGHONEIMY

APPLICATION

```python
import os

from flask import Flask

app = Flask(__name__)


@app.route("/")
def hello_world():
    return "Hello From Depi!"


if __name__ == "__main__":
    app.run(debug=True, host="0.0.0.0", port=int(os.environ.get("PORT", 8080)))
```

APPLICATION

- A simple Application which should print “Hello from Depi!”.
- It’s running using Flask.
- It take a default port “8080”, which would be taken if no port is specified.

DOCKERFILE

Dockerfile > ...

```
...
FROM python:3-alpine

# Create app directory
WORKDIR /app

# Install app dependencies
COPY requirements.txt ./

RUN pip install -r requirements.txt

# Bundle app source
COPY . .

EXPOSE 5000

CMD ["flask", "run","--host","0.0.0.0","--port","5000"]
#CMD [ "python", "app.py" ]
```

DOCKERFILE

- Uses a lightweight Python 3 on Alpine Linux as the base image.
- Sets up the /app directory, installs dependencies from requirements.txt, and copies the app code.
- Exposes port 5000 for the Flask web application.
- Runs the app using Flask on 0.0.0.0:5000.

REQUIREMENTS

requirements.txt

App > requirements.txt

```
...
1
2   blinker==1.6.3
3   click==8.1.7
4   Flask==3.0.0
5   itsdangerous==2.1.2
6   Jinja2==3.1.2
7   MarkupSafe==2.1.3
8   Werkzeug==3.0.0
9
```

REQUIRED STEPS

- We need to push the image to docker registry.
- Centralized Storage: Docker registries provide a centralized repository for storing container images, making them accessible to any Kubernetes node.
- Accessibility: Kubernetes needs the image to be accessible from all nodes in the cluster. By using a registry, the nodes can pull the image regardless of their location.

KUBERNETES

Kubernetes > ! app-deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: flask-app
  template:
    metadata:
      labels:
        app: flask-app
    spec:
      containers:
        - name: app-container
          image: abdelaziz20598/depi-flask:last
          ports:
            - containerPort: 5000
          resources:
            limits:
              cpu: "500m"
              memory: "512Mi"
            requests:
              cpu: "250m"
              memory: "256Mi"
```

KUBERNETES

- Defines a Kubernetes Deployment: Specifies how to deploy the application as a set of replicas for scaling and high availability.
- Pod Configuration: Configures the pod template, including the container image, resource limits, environment variables, and ports.
- Replica Management: Sets the desired number of replicas for the deployment to maintain.
- Update Strategy: May include details for rolling updates to ensure zero downtime during updates.

KUBERNETES

Kubernetes > ! app-service.yaml

```
...
1
2   apiVersion: v1
3   kind: Service
4   metadata:
5     name: app-service
6   spec:
7     selector:
8       app: flask-app
9     ports:
10    - protocol: TCP
11      port: 80
12      targetPort: 5000
13    type: LoadBalancer
14
```

KUBERNETES

- Defines a Kubernetes Service: Exposes the application running in the pods to enable communication.
- Service Type: Specifies how the service is accessible (e.g., ClusterIP, NodePort, or LoadBalancer).
- Port Configuration: Maps an external port to the target port on the pods.
- Selector: Uses labels to match the service to the appropriate pods.

TERRAFORM

- Cluster.
- Network.
- Main.
- Provider.
- Terraform State.
- Terraform State Backup.

TERRAFORM

terraform > main.tf

```
module "network_module" {
  source = "./network"
  main = "main-vpc"

  # management-subnet-name = "management-subnet"
  # restricted-subnet-name = "restricted-subnet"
}

module "cluster_module" {
  source = "./cluster"
  private_us_east_1a_id = module.network_module.private_us_east_1a_id_output
  private_us_east_1b_id = module.network_module.private_us_east_1b_id_output
  public_us_east_1a_id = module.network_module.public_us_east_1a_id_output
  public_us_east_1b_id = module.network_module.public_us_east_1b_id_output
  vpc_id = module.network_module.vpc_id_output
}
```

TERRAFORM

- Defines Two Modules: Sets up a network_module for network resources and a cluster_module for cluster configuration.
- Network Configuration: Uses the network_module to create VPC and subnets (main VPC, management, and restricted subnets).
- Cluster Configuration: The cluster_module uses outputs from the network_module (subnet IDs and VPC ID) for setting up cluster resources.
- Module Source Paths: Both modules reference local directories (./network and ./cluster) for their configurations.

Digital Egypt Pioneers

THANK YOU