Hello,

Here will be a step by step introduction on what i used in order to create this pipeline as well as what technologies and Components.

Components

1) Microservices Architecture

2) Automated Build

Structure

Microservices:

1) User Service: Handles user registration and authentication

2) Inventory Service: Manages Product listing and details

3) Order Service: Processes orders and manages user transaction

Technologies:

1) Python for microservices

2) Docker for Containerization

3) Azure Kubernetes Service (AKS) for Deployment

4) Azure Container Regisrty (ACR) for image storage

CI/CD Pipeline

A screenshot of a computer

Description automatically generatedI created a Repository with the name "CI-CD-Pipeline" as well as the needed structure that I needed.

The structure if the Repository is the Following

* .github/workflows
* Inventory-Service
* K8s
* Order-Service
* User-srvice

For each directory files are created accordingly, for instance

* .github/workflows

Holds the ci-cd-pipeline.yml file. This is my workflow for everyhting I need to created aboyt the services and the authentication for Azure and Docker.



A screenshot of a computer program

Description automatically generated

For Invetory-Service, Order-Service and User-Service the folder structure is similar

* Dockerfile
* App.py
* Requirements.txt
* User-deployment.yml
* User-service.yml

A black background with horizontal lines

Description automatically generated

The Docker file is essential for defining how my application works , it contains instratctons on how to create a Docker Image for my application and it specifies:

1. The Base Image
2. The installation dependecies
3. The command to run my application

The App.py holds my logic for each of my microservices in Python. Depnding on the functionality of the microservice, the file contains the code for HTTP requests, interact woth database and process any buisness logic.

A computer screen with text and numbers

Description automatically generated

The Requirements.txt hold the version of flask tha I used

User-deployment and user-service.yml are the setup for deploying the User Service on Azure Kubernetes Service.

User-Service

A screenshot of a computer

Description automatically generated

User-Deployment

A screenshot of a computer

Description automatically generated

The k8s folder contains the followoing files:

* Deployment.yaml
* Service.yaml

Deploymnent as well as service yaml files are a Kubernetes manifest file used to define a Service and a Deployment resource in a Kuberneted cluster.

With the Depolymentl.yaml file we can:

1. Define Desired State
2. Automatic Scalling and Rollback
3. Manage Pods

A screenshot of a computer program

Description automatically generated

With the Service.yaml file we can:

1. Expose pods
2. Load Balancing
3. Service Discovery

A screenshot of a computer

Description automatically generated

Both of these files are crucial for managing our applications lifecycle efficientlyin Kubernetes, allowing for better scalabillity reliability and ease of updates.

Azure Stracture

For azure i created the Following:

* Resource Group
* Azure Kubernetes Service
* Azure Container Registry



Inside the Resource Group I created an AKS Cluster as well as an Azure Container Registry

A screenshot of a computer

Description automatically generated

An important step is to provide the Service Principle in github and a Docker Token in order to minimize the authentication error we may have.

So on the Azure Cloud shell I run the following command

**az ad sp create-for-rbac --name "<your-service-principal-name>" --role contributor --scopes /subscriptions/<subscription-id>**

and then I got the following jason output

(this is an example outpout)

A black background with green text

Description automatically generated

Then I copied to my github secrets

A black and white background with black lines

Description automatically generated with medium confidence

We can see that I have the docker token as well, wich helped me authenticate Docker Hub to push and pull images.

A screenshot of a computer

Description automatically generated

To conclude, after managin and resolving any error that occurred I started the Action on git hub

The Build and Deploy section was succsefull

A screenshot of a computer

Description automatically generated

Although the workflow ended succsefully, when I run kubectl get pods I got the below error

A yellow background with black text

Description automatically generated

After some troubleshooting I did find that I didn’t had any authorization

So I createtd a secrete wiith the admin user of ACR

A computer code on a black background

Description automatically generated

But unfortunatly I didn’t have any outcomes form that.

I’ll be pleased if we can discuss the solution to our call.