Azure Deployment Documentation

e-Commerce Microservices

**Note:** All resources referenced in this document have been deleted. Names used are **project-related placeholders**. This documentation is **reference only** to demonstrate your deployment process.

## Login and Resource Setup

**Login via Azure CLI**

az login --tenant <TENANT\_ID>

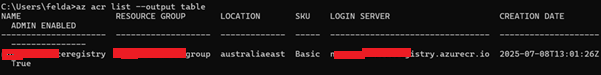
**Create Azure Resource Group**

az group create --name <xx-group> --location australiaeast

Create Azure Container Registry (ACR)

az acr create --resource-group <xx-group> --name <registry\_name> --sku Basic --location australiaeast --admin-enabled true

**Screenshots:**



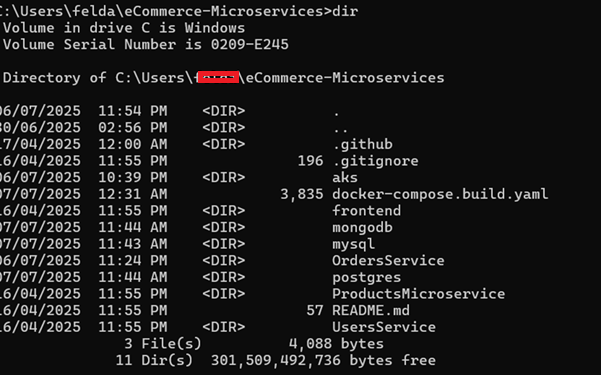
## Build Docker Images Locally

**Navigate to Microservices Root Folder**

cd /path/to/eCommerce-Microservices

**Check Docker Build File**

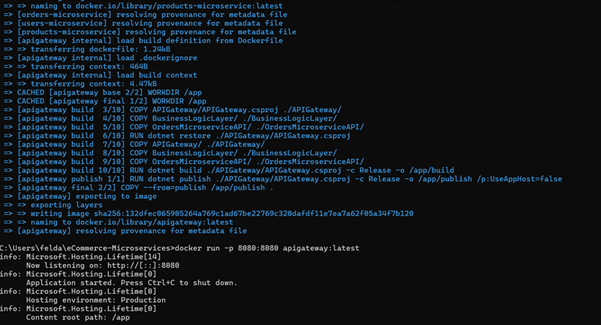
Ensure docker-compose.build.yaml exists in the root folder:



**Build and Run Docker Compose**

# Build images

docker-compose -f docker-compose.build.yaml build



# Run containers

docker-compose -f docker-compose.build.yaml up --build

# Remove old containers if needed

docker-compose -f docker-compose.build.yaml down

* **Microservices built:**
  + orders-microservice
  + products-microservice
  + users-microservice
  + apigateway
  + ecommerce-mysql
  + ecommerce-postgres
  + ecommerce-mongodb

**Screenshot:**

A screenshot of a computer program

AI-generated content may be incorrect.

## Tag and Push Images to ACR (Reference)

**Login to ACR**

az acr login --name <registry\_name>

**Tag Docker Images**

docker tag orders-microservice:latest <registry\_name>.azurecr.io/orders-microservice:latest

docker tag products-microservice:latest <registry\_name>.azurecr.io/products-microservice:latest

docker tag users-microservice:latest <registry\_name>.azurecr.io/users-microservice:latest

docker tag apigateway:latest <registry\_name>.azurecr.io/apigateway:latest

docker tag ecommerce-mysql:latest <registry\_name>.azurecr.io/ecommerce-mysql:latest

docker tag ecommerce-postgres:latest <registry\_name>.azurecr.io/ecommerce-postgres:latest

docker tag ecommerce-mongodb:latest <registry\_name>.azurecr.io/ecommerce-mongodb:latest

A black screen with white text

AI-generated content may be incorrect.

**Push Docker Images**

docker push xx-registry.azurecr.io/orders-microservice:latest

docker push xx-registry.azurecr.io/products-microservice:latest

docker push <registry\_name>..azurecr.io/users-microservice:latest

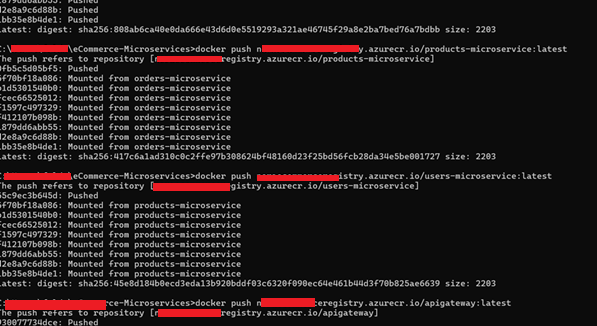
docker push <registry\_name>..azurecr.io/apigateway:latest

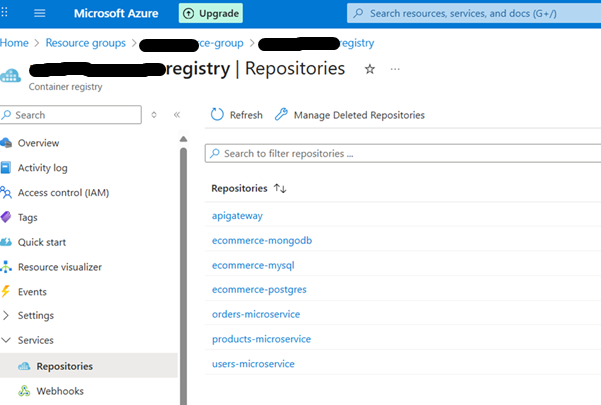
docker push <registry\_name>..azurecr.io/ecommerce-mysql:latest

docker push <registry\_name>..azurecr.io/ecommerce-postgres:latest

docker push <registry\_name>..azurecr.io/ecommerce-mongodb:latest

**Screenshot**:





## Create and Configure AKS Cluster (Reference)

**Create AKS Cluster**

az aks create \

--resource-group xx-group \

--name demo-aks-cluster \

--node-count 1 \

--node-vm-size Standard\_B2s \

--generate-ssh-keys

**Register Missing Providers (if needed)**

az provider register --namespace Microsoft.OperationalInsights

az provider show --namespace Microsoft.Insights --query "registrationState"

**Attach ACR to AKS**

az aks update \

--resource-group xx-group \

--name demo-aks-cluster \

--attach-acr xx-registry

**A screen shot of a computer

AI-generated content may be incorrect.**

**Configure kubectl**

az aks get-credentials --resource-group xx-group --name demo-aks-cluster

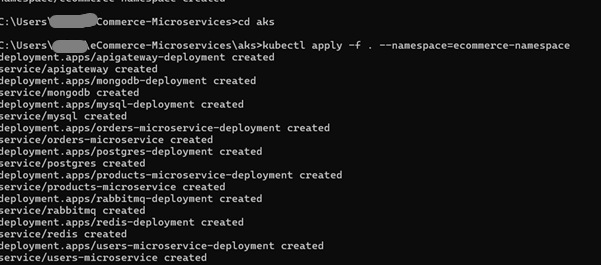
kubectl get nodes

kubectl create namespace ecommerce-namespace

## Deploy Microservices to AKS

**Apply Deployment and Services**

kubectl apply -f . --namespace=ecommerce-namespace

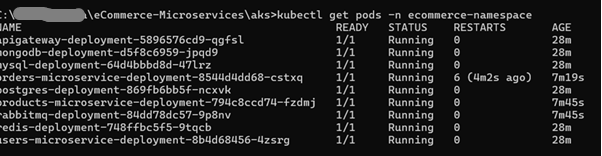


A screenshot of a computer

AI-generated content may be incorrect.

**Check Pods and Services**

kubectl get pods -n ecommerce-namespace



kubectl get svc -n ecommerce-namespace



## Troubleshooting Pods

* **502 Gateway Error**:
  + Check ocelot.json routing (upstream/downstream paths).
  + Rebuild affected Docker images:
  + docker-compose -f docker-compose.build.yaml build
  + docker push <affected\_image>
  + Trigger rolling update in AKS:
  + kubectl rollout restart deployment/<deployment\_name> -n ecommerce-namespace
  + kubectl rollout status deployment/<deployment\_name> -n ecommerce-namespace
* **Fixing Service Issues:**
  + Update code in the microservice if required.
  + Rebuild, tag, push, and rollout update as shown above.

## Test from Postman (Reference)

* Test API endpoints using the external IP from kubectl get svc.
* Replace [EXTERNAL\_IP] with the actual IP if the cluster is live.
* Example endpoints:
* [EXTERNAL\_IP]/gateway/products
* A screenshot of a computer

  AI-generated content may be incorrect.
* [EXTERNAL\_IP]/gateway/orders

A screenshot of a computer

AI-generated content may be incorrect.

Screenshots to demonstrate successful API calls and responses.

## Cleanup (Reference)

kubectl delete all --all -n ecommerce-namespace

az aks delete --resource-group xx-group --name demo-aks-cluster

All resources in the project have been deleted; these steps are for documentation purposes.