HƯỚNG DẪN DEPLOY AZURE FUNCTIONS (.NET 8 – ISOLATED) LÊN AZURE CLOUD

Tài liệu này hướng dẫn triển khai backend Battle Game (Azure Functions .NET 8 – isolated) lên Azure Cloud, bao gồm cấu hình CORS, App Settings (MySQL), kiểm thử sau deploy và gợi ý CI/CD.

# 0) CHUẨN BỊ

- .NET 8 SDK, Azure Functions Core Tools v4

- VS Code + Azure Functions extension (tuỳ chọn) hoặc Azure CLI

- Đăng nhập Azure: VS Code → Azure: Sign In hoặc dùng CLI: az login

- Backend chạy OK local (func start)

- CSDL: khuyến nghị Azure Database for MySQL (Flexible Server) khi lên cloud

# 1) THÀNH PHẦN HẠ TẦNG

- Function App (Consumption Plan): tiết kiệm chi phí, auto-scale

- Storage Account: bắt buộc cho metadata/trigger

- Application Insights: logging/monitoring

- CORS: cho phép origin của FE

- App Settings: chứa chuỗi kết nối MySQL và các cấu hình khác

# 2) TRIỂN KHAI BẰNG VS CODE (GUI)

2.1 Tạo Function App

- Azure panel → Functions → Create Function App in Azure

- Runtime: .NET (Isolated), OS: Windows hoặc Linux, Region: Southeast Asia

- Resource Group + Storage Account + Plan: Consumption

2.2 Cấu hình App Settings & CORS

- Portal → Function App → Configuration → Application settings → New

• MySqlConnection=Server=<mysql-host>;Database=battlegame;User Id=<user>;Password=<pwd>;

• Nếu Azure MySQL: thêm SslMode=Required (tùy cấu hình)

- Portal → CORS: thêm origin FE (ví dụ https://<your-fe>.azurestaticapps.net)

2.3 Deploy từ VS Code

- Chuột phải thư mục chứa .csproj → Deploy to Function App → chọn Function App

2.4 Kiểm tra sau deploy

- Portal → Functions: thấy 4 function (registerplayer, createasset, assignasset, getassetsbyplayer)

- Có thể Test/Run ngay trong portal hoặc dùng curl

# 3) TRIỂN KHAI BẰNG AZURE CLI (SCRIPT)

# Login & chọn subscription

az login

az account set --subscription "<SUBSCRIPTION\_ID\_OR\_NAME>"

# Biến môi trường

RG="rg-battlegame"

LOC="southeastasia"

ST="stbattlegame$RANDOM"

FUNCAPP="func-battlegame-$RANDOM"

RUNTIME="dotnet-isolated"

OS="Windows" # hoặc Linux

# Tạo RG + Storage + Function App (Consumption)

az group create -n $RG -l $LOC

az storage account create -n $ST -g $RG -l $LOC --sku Standard\_LRS

az functionapp create --resource-group $RG --consumption-plan-location $LOC --name $FUNCAPP --storage-account $ST --functions-version 4 --runtime $RUNTIME --os-type $OS

# App Settings (MySQL)

az functionapp config appsettings set -g $RG -n $FUNCAPP --settings "MySqlConnection=Server=<mysql-host>;Database=battlegame;User Id=<user>;Password=<pwd>;"

# CORS

az functionapp cors add -g $RG -n $FUNCAPP --allowed-origins "https://<your-fe>.azurestaticapps.net"

# Zip deploy (sau khi publish)

dotnet publish -c Release

Compress-Archive -Path .\bin\Release\net8.0\publish\\* -DestinationPath .\deploy.zip

az functionapp deployment source config-zip -g $RG -n $FUNCAPP --src ./deploy.zip

# 4) DI TRÚ CẤU HÌNH TỪ LOCAL LÊN CLOUD

- local.settings.json chỉ dùng local, KHÔNG upload lên Azure

- Đưa cấu hình (MySqlConnection, v.v.) vào Application settings (Portal/CLI)

- Phân tách môi trường dev/test/prod bằng RG/App khác nhau hoặc Deployment Slots (Premium)

# 5) BẢO MẬT & QUYỀN TRUY CẬP

- AuthorizationLevel:

• Dev local: Anonymous (tạm thời)

• Prod: Function + dùng Function Key

- Lấy Function Key: Portal → Function → Function keys (hoặc az rest)

- Application Insights: theo dõi logs, requests, exceptions

# 6) KIỂM THỬ SAU TRIỂN KHAI

# GET report

curl -i "https://<your-function>.azurewebsites.net/api/getassetsbyplayer"

# POST create player (nếu dùng Function key)

curl -i -X POST "https://<your-function>.azurewebsites.net/api/registerplayer?code=<FUNCTION\_KEY>" -H "Content-Type: application/json" -d "{"PlayerName":"Player Cloud","FullName":"Demo Cloud","Age":21,"Level":5}"

# FE: REACT\_APP\_API\_BASE = https://<your-function>.azurewebsites.net

# 7) LOG & CHẨN ĐOÁN

- VS Code: Start Streaming Logs

- Azure CLI: az webapp log tail -g <rg> -n <functionapp>

- Application Insights: Logs (KQL), Live Metrics

# 8) CI/CD (GITHUB ACTIONS MẪU)

name: Deploy Azure Functions (dotnet-isolated)

on:

push:

branches: [ "main" ]

jobs:

build-and-deploy:

runs-on: ubuntu-latest

permissions:

id-token: write

contents: read

steps:

- uses: actions/checkout@v4

- name: Setup .NET

uses: actions/setup-dotnet@v4

with:

dotnet-version: '8.0.x'

- name: Restore & Publish

working-directory: ./backend

run: |

dotnet restore

dotnet publish -c Release -o publish

- name: 'Azure Login'

uses: azure/login@v2

with:

client-id: ${{ secrets.AZURE\_CLIENT\_ID }}

tenant-id: ${{ secrets.AZURE\_TENANT\_ID }}

subscription-id: ${{ secrets.AZURE\_SUBSCRIPTION\_ID }}

- name: Deploy to Function App

uses: azure/functions-action@v1

with:

app-name: "<your-function-app-name>"

package: "./backend/publish

# 9) LỖI THƯỜNG GẶP & CÁCH XỬ LÝ

- CORS blocked: thêm FE origin vào CORS của Function App, gọi đúng URL api

- MySQL connect fail: kiểm tra firewall Azure MySQL, SSL mode, user/pass

- 404: sai route hoặc deploy sai thư mục (phải là folder publish)

- 500: xem log streaming / App Insights (exceptions, traces)

# 10) ROLLBACK NHANH

- Zip deploy: giữ bản zip ổn → deploy lại zip cũ

- GitHub Actions: checkout tag/commit cũ rồi push lại để redeploy