CANTON WEKE OTIENO

SECURITY ENGINEER TRACK

STUDENT TRACKING NUMBER; ADC-SE01-24010

CYBER SHUJAA

INSTRUCTOR: Dr. Paula

LAB 09B - IMPLEMENT AZURE CONTAINER INSTANCES

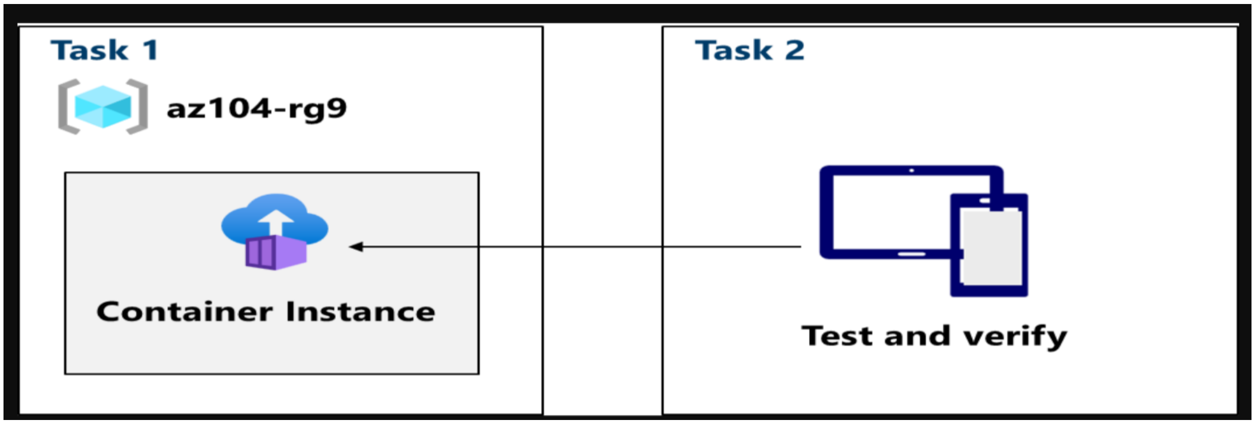
The completely managed service Azure Container Instances (ACI) is used to launch and operate containerized apps in Azure. Once a container is deployed, it will automatically distribute the resources to it based on the precise number of CPU cores and memory that one chooses**.**

Introduction

In this lab, involves practical procedure on how to implement and deploy Azure Container Instances.

The lab has two task and configuration. Task one involves deploying an azure container instance using a docker. And task two involves testing and verifying of azure container instance. Through these tasks, the Lab assignment aims to enhance understanding of containerization and cloud deployment, equipping individuals with the skills to efficiently deploy and manage containerized applications on Azure.

Architecture diagram



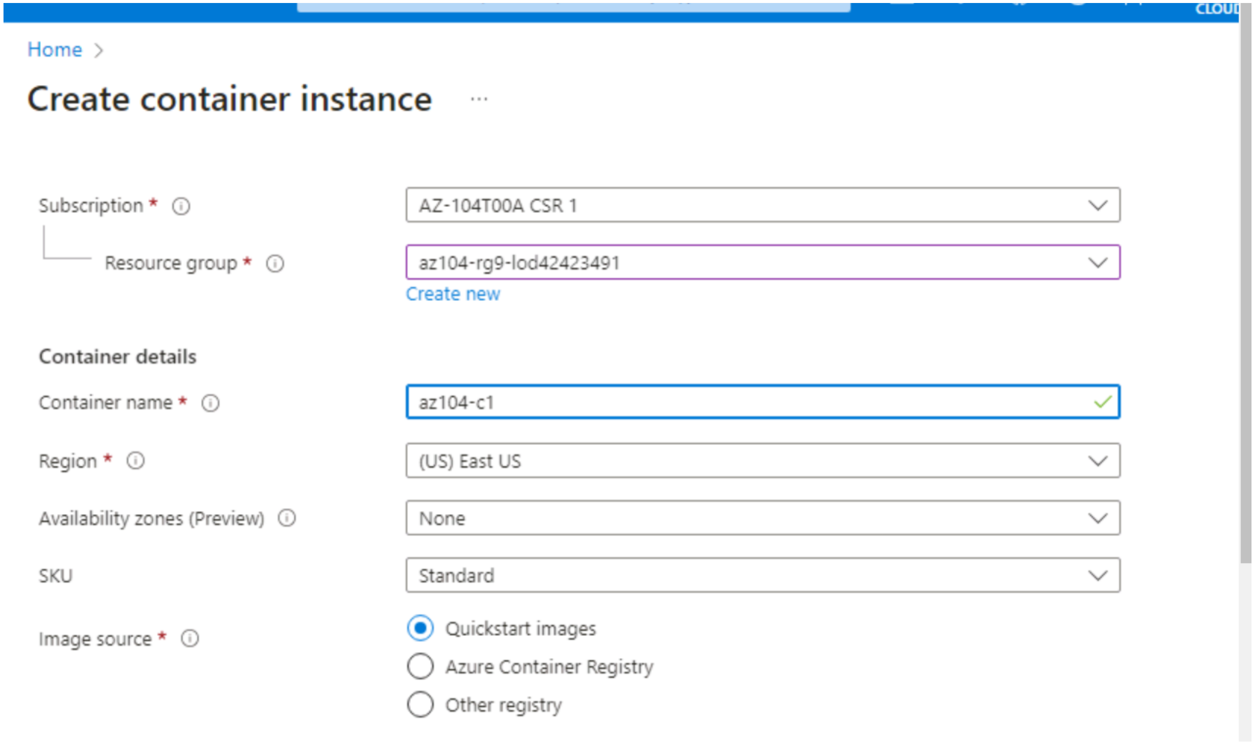
Task 1: Deploy an Azure Container Instance using a Docker image

In this task, it involves creating of a simple web application using a Docker image. Docker is a platform that provides the ability to package and run applications in isolated environments called containers. Azure Container Instances provides the compute environment for the container image.

Instructions

- Sign in to the Azure portal - https://portal.azure.com.
- In the Azure portal, search for and select Container instances and then, on the Container instances blade, click + Create.

- On the Basics tab of the Create container instance blade, specify the following settings (leave others with their default values):



Home >

Create container instance ...

Subscription * ⓘ AZ-104T00A CSR 1

Resource group * ⓘ az104-rg9-lod42423491

Create new

Container details

Container name * ⓘ az104-c1

Region * ⓘ (US) East US

Availability zones (Preview) ⓘ None

SKU Standard

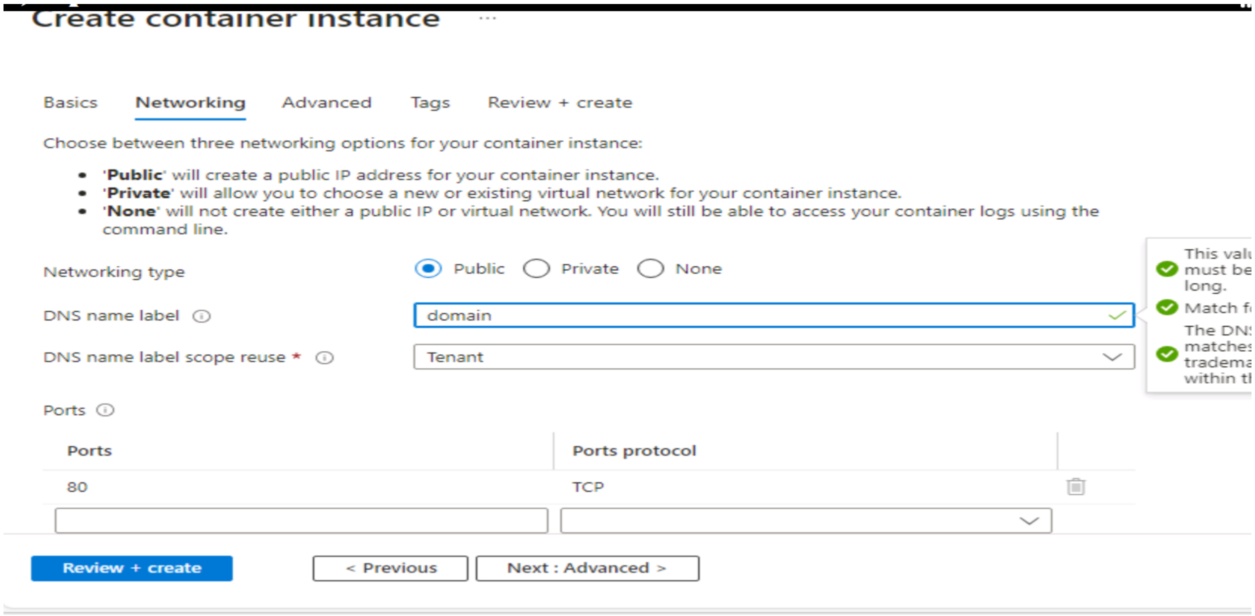
Image source * ⓘ

● Quickstart images

○ Azure Container Registry

○ Other registry

- Click Next: Networking > and specify the following settings (leave others with their default values):



Create container instance ...

Choose between three networking options for your container instance:

- '**Public**' will create a public IP address for your container instance.
- '**Private**' will allow you to choose a new or existing virtual network for your container instance.
- '**None**' will not create either a public IP or virtual network. You will still be able to access your container logs using the command line.

Networking type ● Public ○ Private ○ None

DNS name label ⓘ domain

DNS name label scope reuse * ⓘ Tenant

This valu must be long.

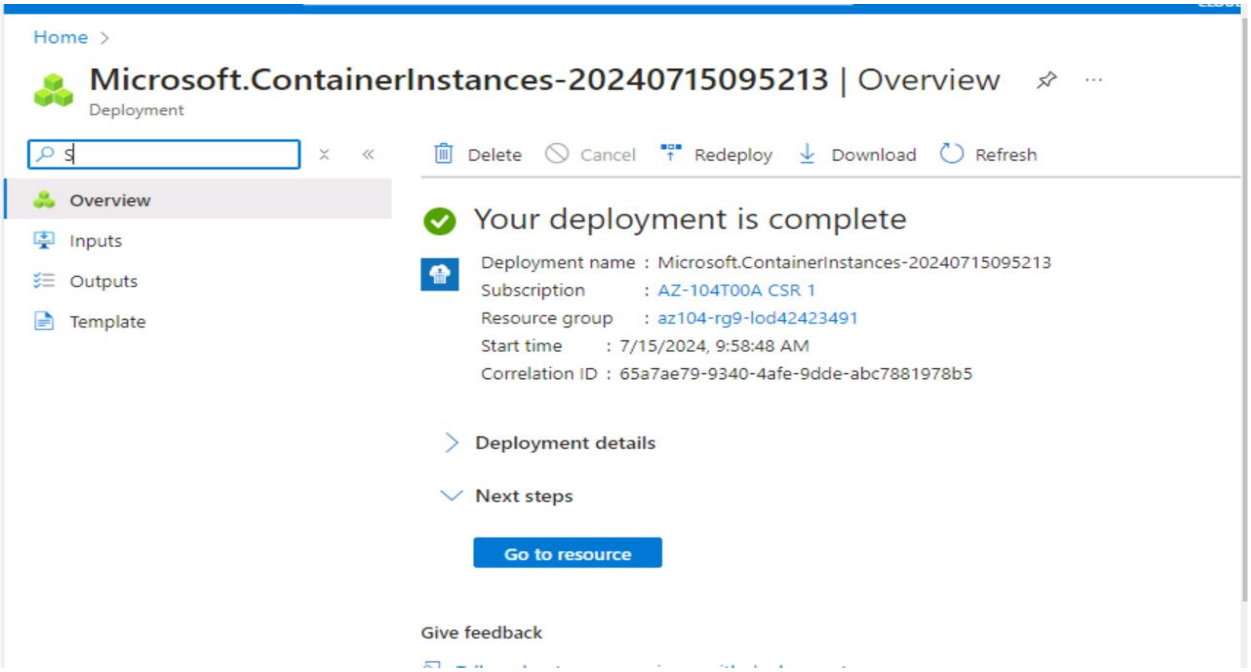
Match f

The DN: matches tradema within tl

Ports ⓘ

Ports	Ports protocol
80	TCP

- Setting Value.
- Click Next: Advanced >, review the settings without making any changes.
- Click Review + Create, ensure that the validation passed and then select Create.



Your deployment is complete

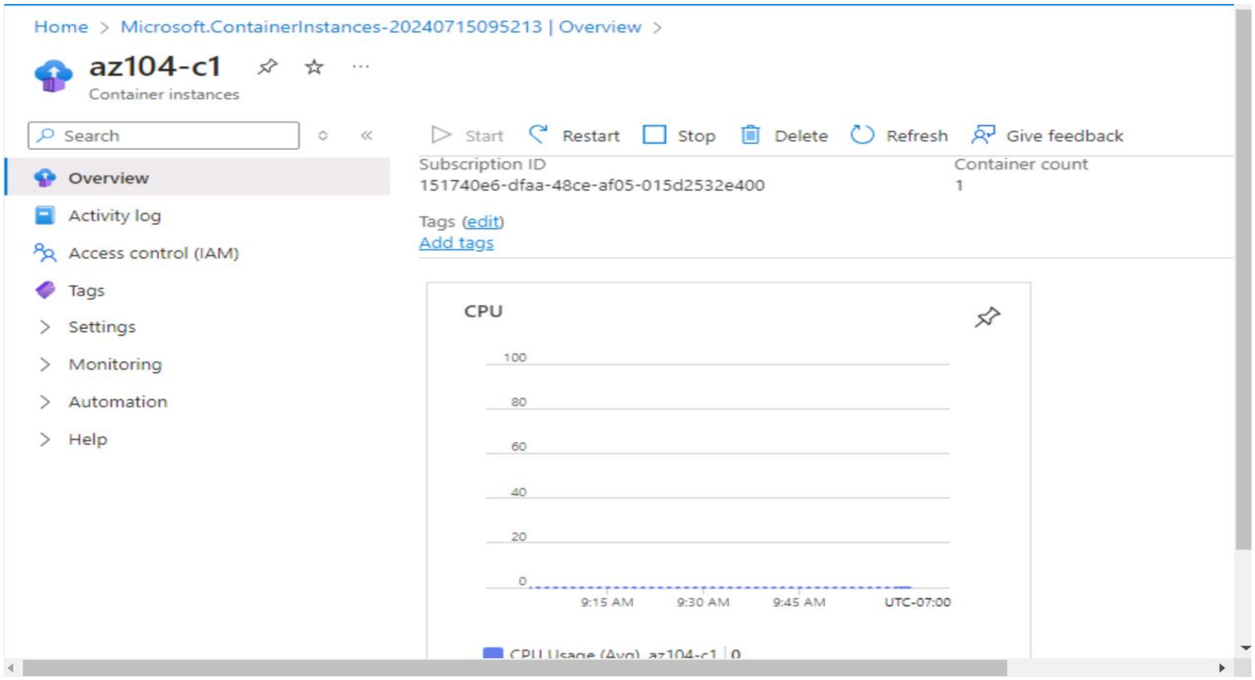
Deployment name : Microsoft.ContainerInstances-20240715095213
Subscription : AZ-104T00A CSR 1
Resource group : az104-rg9-lod42423491
Start time : 7/15/2024, 9:58:48 AM
Correlation ID : 65a7ae79-9340-4afe-9dde-abc7881978b5

Task 2: Test and verify deployment of an Azure Container Instance

This task involves reviewing of the deployment of the container instance. By default, the Azure Container Instance is accessible over port 80. After the instance has been deployed, you can navigate to the container using the DNS name that has been provided in the previous lab task.

Instruction on how to do the configuration

- On the deployment blade, click the Go to resource link.
- On the Overview blade of the container instance, verify that Status is reported as Running.

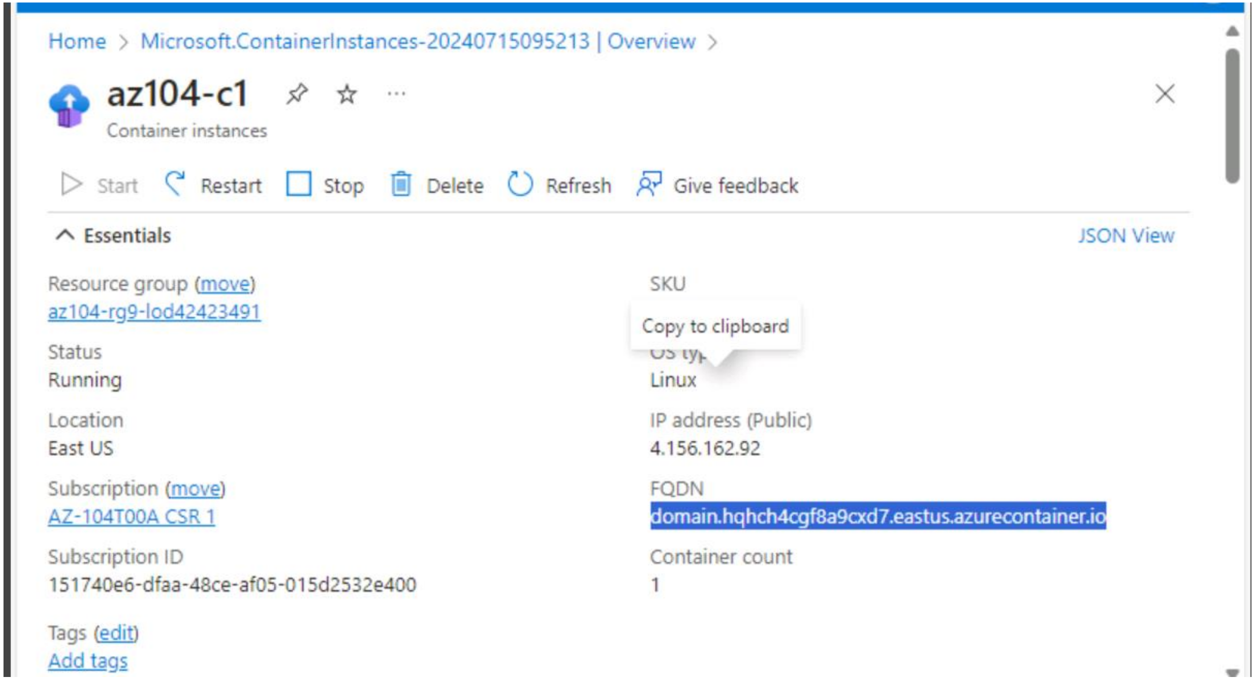


Subscription ID
151740e6-dfaa-48ce-af05-015d2532e400

Container count
1

Tags (edit)
Add tags

- Copy the value of the container instance FQDN, open a new browser tab, and navigate to the corresponding URL.



az104-c1

Container instances

Resource group (move)
az104-rg9-lod42423491

Status
Running

Location
East US

Subscription (move)
AZ-104T00A CSR 1

Subscription ID
151740e6-dfaa-48ce-af05-015d2532e400

Tags (edit)
Add tags

SKU

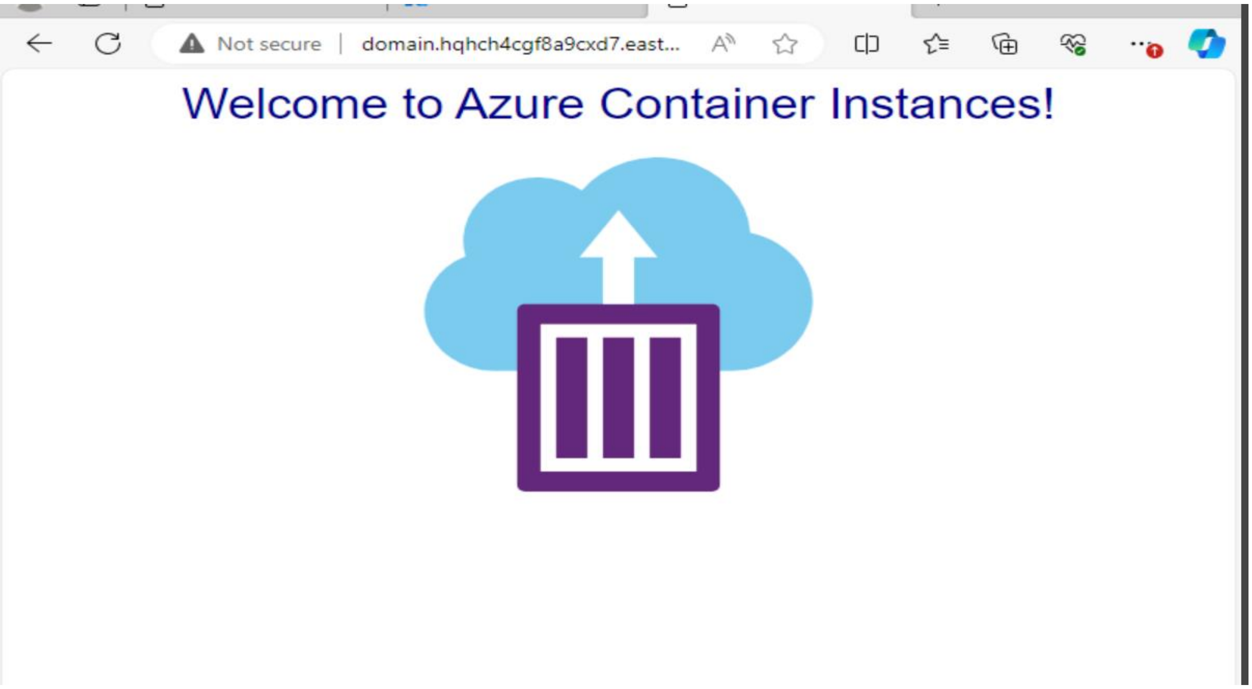
OS type
Linux

IP address (Public)
4.156.162.92

FQDN
domain.hqhch4cgf8a9cxd7.eastus.azurecontainer.io

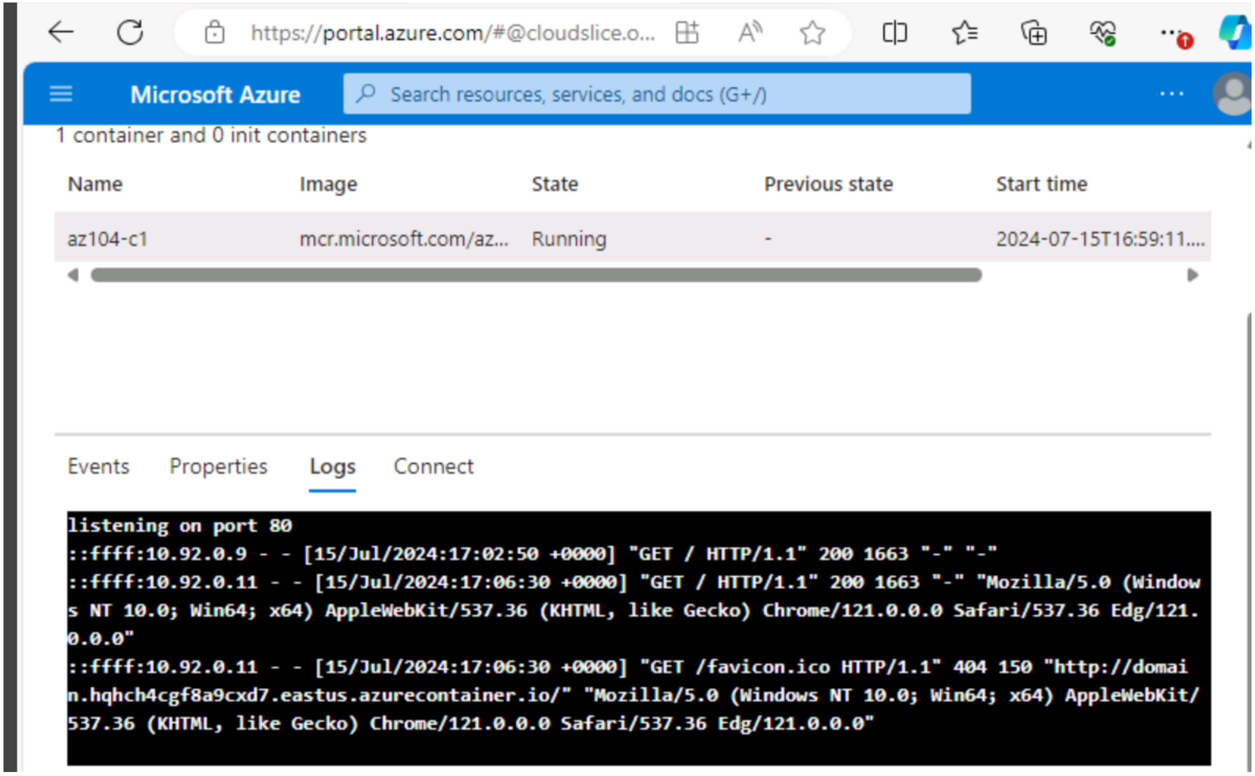
Container count
1

- Verify that the Welcome to Azure Container Instance page is displayed. Refresh the page several times to create some log entries then close the browser tab.



Welcome to Azure Container Instances!

- In the Settings section of the container instance blade, click Containers, and then click Logs.
- Verify that you see the log entries representing the HTTP GET request generated by displaying the application in the browser.


https://portal.azure.com/#@cloudslice.o...
Microsoft Azure
Search resources, services, and docs (G+/)

1 container and 0 init containers

Name	Image	State	Previous state	Start time
az104-c1	mcr.microsoft.com/az...	Running	-	2024-07-15T16:59:11....

Events
Properties
Logs
Connect

```
listening on port 80
::ffff:10.92.0.9 - - [15/Jul/2024:17:02:50 +0000] "GET / HTTP/1.1" 200 1663 "-" "-"
::ffff:10.92.0.11 - - [15/Jul/2024:17:06:30 +0000] "GET / HTTP/1.1" 200 1663 "-" "Mozilla/5.0 (Window
s NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36 Edg/121.
0.0.0"
::ffff:10.92.0.11 - - [15/Jul/2024:17:06:30 +0000] "GET /favicon.ico HTTP/1.1" 404 150 "http://domai
n.hqhch4cgf8a9cxd7.eastus.azurecontainer.io/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36 Edg/121.0.0.0"
```

Conclusion

Lab 09b has provided valuable insights into the deployment and management of containerized applications using Azure services. By deploying an Azure Container Instance (ACI) using a Docker image, I gained practical experience in container orchestration and cloud deployment, showcasing the ease and efficiency of using ACI for running isolated applications in the cloud. The subsequent task of testing and verifying the deployment ensured that the application was not only successfully deployed but also functioned as expected. This process underscored the importance of thorough verification and testing in the deployment cycle to maintain application integrity and performance**.**

Lab assignment has taught me several important concepts that one may use in real-world situations. First off, using a Docker image to launch an Azure Container Instance demonstrated how easy and efficient containerization is for delivering applications. Greater adaptability, scalability, and consistency across many situations are made possible by this method. Second, the process of testing and confirming the deployment brought to light how important it is to have thorough validation procedures in place to guarantee that applications are dependable and strong when put into use. This stage is essential for averting possible problems and guaranteeing that the application satisfies user needs and functional specifications.