

How to deploy a .NET 7 Web API in Azure Container Instance

1. Create an Azure Container Registry

We create a Container Registry service

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo, a search bar, and various utility icons. The left sidebar contains a list of navigation options, with 'Create a resource' at the top. Below it, 'Home', 'Dashboard', and 'All services' are listed. The 'FAVORITES' section includes 'All resources', 'Resource groups', 'Quickstart Center', 'App Services', 'Function App', 'SQL databases', 'Azure Cosmos DB', 'Virtual machines', 'Load balancers', 'Storage accounts', 'Virtual networks', 'Microsoft Entra ID', 'Monitor', 'Advisor', 'Microsoft Defender for Cloud', and 'Cost Management + Billing'. The main content area is titled 'Create a resource' and features a search bar for 'Search services and marketplace'. Below the search bar, there are sections for 'Get Started', 'Recently created', and 'Categories'. The 'Categories' section lists various Azure services, with 'Containers' highlighted. The 'Popular Azure services' section lists several services, including 'Azure Kubernetes Service (AKS)', 'Web App for Containers', 'Batch Service', 'Kubernetes - Azure Arc', 'Container App', 'Container Instances', 'Container Registry' (which is highlighted with a red box), and 'Service Fabric Cluster'. The 'Popular Marketplace products' section lists various operating systems and container management tools, including 'Windows Server 2022 Core Datacenter Minimal OS', 'Basic', 'Hyper-V Server on Windows Server 2016', 'Docker Engine Community on Ubuntu 20.04 LTS', 'Docker Compose Server on Windows Server 2016', 'Pay-as-You-Go', 'Portainer 2.19 on Ubuntu 22.04', and 'Docker Compose Server on Ubuntu Server 20.04'.

[Home](#) > [Create a resource](#) >

Create container registry

Basics[Networking](#)[Encryption](#)[Tags](#)[Review + create](#)

Azure Container Registry allows you to build, store, and manage container images and artifacts in a private registry for all types of container deployments. Use Azure container registries with your existing container development and deployment pipelines. Use Azure Container Registry Tasks to build container images in Azure on-demand, or automate builds triggered by source code updates, updates to a container's base image, or timers. [Learn more](#)

Project details

Subscription *

Azure subscription 1



Resource group *

(New) myRG

[Create new](#)

Instance details

Registry name *


mywebapicontainer





.azurecr.io

Location *












West Europe

Use availability zones ☐

 Availability zones are activated on premium registries and in regions that support availability zones. [Learn more](#)

Pricing plan * 

Basic

 Create a resource Home Dashboard All services**FAVORITES** All resources Resource groups Quickstart Center App Services Function App SQL databases Azure Cosmos DB Virtual machines Load balancers Storage accounts Virtual networks Microsoft Entra ID Monitor Advisor Microsoft Defender for Cloud Cost Management +

Billing

Review + create

< Previous

Next: Networking >

[Home](#) > [Create a resource](#) >

Create container registry ...



Validation passed

[Basics](#) [Networking](#) [Encryption](#) [Tags](#) [Review + create](#)

Registry details

Basics

Registry name	mywebapicontainer
Subscription	Azure subscription 1
Resource Group	myRG
Location	West Europe
Availability zones	Disabled
Pricing plan	Basic

Networking

Public network access	Yes
-----------------------	-----

Encryption

Customer-Managed Key	Disabled
Identity	None
Key Vault	None
Encryption key	None

[+ Create a resource](#)[Home](#)[Dashboard](#)[All services](#)

★ FAVORITES

[All resources](#)[Resource groups](#)[Quickstart Center](#)[App Services](#)[Function App](#)[SQL databases](#)[Azure Cosmos DB](#)[Virtual machines](#)[Load balancers](#)[Storage accounts](#)[Virtual networks](#)[Microsoft Entra ID](#)[Monitor](#)[Advisor](#)[Microsoft Defender for](#)

[Create](#)[< Previous](#)[Next >](#)[Download a template for automation](#)

Microsoft Azure

[+ Create a resource](#)[Home](#)[Dashboard](#)[All services](#)[★ FAVORITES](#)[All resources](#)[Resource groups](#)[Quickstart Center](#)[App Services](#)[Function App](#)[SQL databases](#)[Azure Cosmos DB](#)[Virtual machines](#)[Load balancers](#)[Storage accounts](#)[Home >](#)

Microsoft.ContainerRegistry | Overview

Deployment



Delete



Cancel



Redeploy



Download



Refresh

[Overview](#)[Inputs](#)[Outputs](#)[Template](#)

Your deployment is complete



Deployment name : Microsoft.ContainerRegistry

Subscription : [Azure subscription 1](#)Resource group : [myRG](#)

Start time : 12/12/2023, 5:17:47 PM

Correlation ID : 2c07fd62-23fb-40f9-963c-156ccad22410

[Deployment details](#)[Next steps](#)[Go to resource](#)

Give feedback

[Tell us about your experience with deployment](#)

The screenshot displays the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo, a search bar, and user information for 'luiscoenriquez@hotmail.com'. The left sidebar lists various Azure services, with 'mywebapicontainer' selected under 'Container registry'. The main content area shows the 'Overview' page for 'mywebapicontainer'. It includes a search bar, 'Move' and 'Delete' buttons, and a table of 'Essentials' with details like Resource group (myRG), Location (West Europe), Subscription (Azure subscription 1), and Subscription ID (846901e6-da09-45c8-98ca-7cca2353ff0e). Below this, there are tabs for 'Get started', 'Monitoring', 'Capabilities (9)', and 'Tutorials'. A large section titled 'Simplify container lifecycle management' explains the purpose of container registry and provides three actionable links: 'Push a Container Image', 'Manage your access controls', and 'Deploy a container image'.

Home > Microsoft.ContainerRegistry | Overview >

mywebapicontainer
Container registry

Search

Move Delete

Overview

Activity log

Access control (IAM)

Tags

Quick start

Events

Settings

Access keys

Encryption

Identity

Networking

Microsoft Defender for Cloud

Properties

Locks

Services

Repositories

Webhooks

Geo-replications

Tasks

Essentials

Resource group (move) : [myRG](#)

Location : West Europe

Subscription (move) : [Azure subscription 1](#)

Subscription ID : 846901e6-da09-45c8-98ca-7cca2353ff0e

Soft delete (Preview) : [Disabled](#)

Tags (edit) : [Add tags](#)

Login server : mywebapicontainer.azurecr.io

Creation date : 12/12/2023, 5:17 PM GMT+1

Provisioning state : Succeeded

Pricing plan : Basic

JSON View

Get started Monitoring Capabilities (9) Tutorials

Simplify container lifecycle management

Container registry allows you to build, store, and manage container images and artifacts in a private registry for all types of container deployments. [Learn more](#)

Push a Container Image
Get instructions on how to store your container images in your container registry. [Learn more about container images](#)

Manage your access controls
Secure your container registry by configuring how users interact with your container registry

Deploy a container image
Explore the latest ways to control, secure, and streamline artifact deployments in Container Registry - or choose one and get started.

2. Create a .NET 7 Web API in Visual Studio 2022 Community Edition

Open Visual Studio 2022 Community Edition and we create a new .NET 7 API

— □ ×

Additional information

ASP.NET Core Web API

C# Linux macOS Windows API Cloud Service Web Web API

Framework ⓘ

.NET 7.0 (Standard Term Support) ▾

Authentication type ⓘ

None ▾

☐ Configure for HTTPS ⓘ

☐ Enable Docker ⓘ

Docker OS ⓘ

Linux ▾

☒ Enable OpenAPI support ⓘ

☐ Do not use top-level statements ⓘ

☒ Use controllers ⓘ

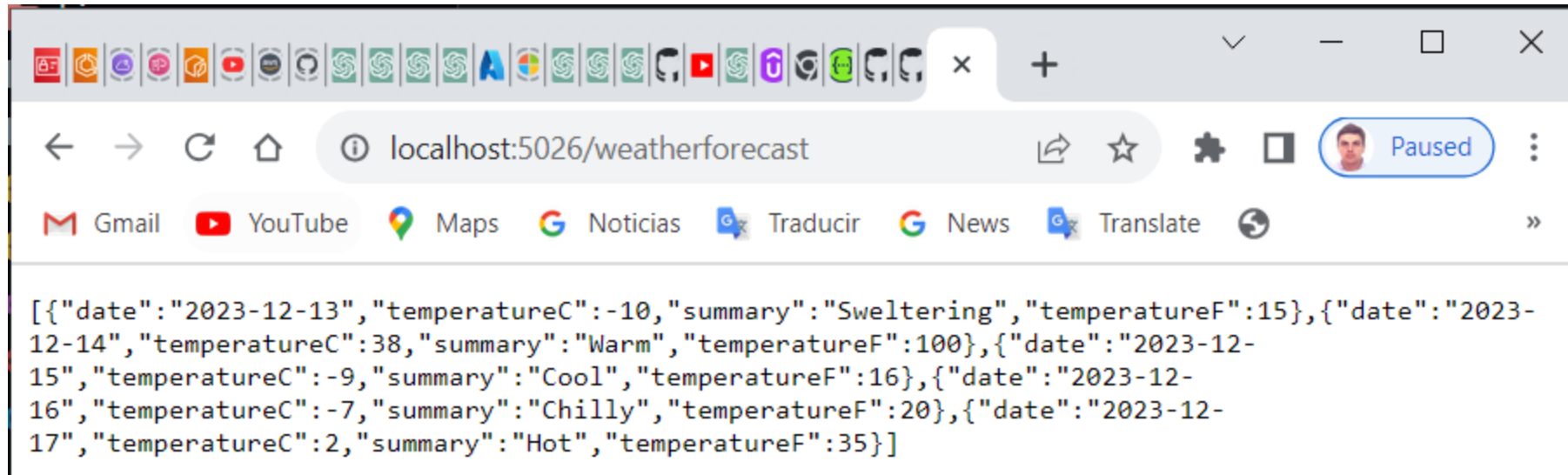
Back

Create

We build and run the application to verify it

We navigate to the Web API URL controller endpoint:

<http://localhost:5026/weatherforecast>



And also we visit the swagger API doc webpage:

localhost:5026/swagger/index.html

Gmail YouTube Maps Noticias Traducir News Translate Elastic Container Se... LattePanda 3 Delta... Turkey launches TO... ESP32 SIM800L GS... Basic-ESP32-Tutori... GPS Car Tracker Usi...

Swagger.
Supported by SMARTBEAR

Select a definition **Azure .NET 8 API v1**

Azure .NET 8 API ^{1.0} ^{OAS3}

<http://localhost:5026/swagger/v1/swagger.json>

Azure .NET 8 API

GET /weatherforecast

Parameters Cancel

No parameters

Execute **Clear**

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5026/weatherforecast' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:5026/weatherforecast
```

The screenshot shows a web browser at `localhost:5026/swagger/index.html`. The Swagger UI displays the following information:

- Responses:** A section for viewing API responses.
- Curl:** A code block showing the command:

```
curl -X 'GET' \
'http://localhost:5026/weatherforecast' \
-H 'accept: application/json'
```
- Request URL:** A text field containing `http://localhost:5026/weatherforecast`.
- Server response:** A table with two columns: **Code** and **Details**.
- Response 200:** The **Details** column shows the response body as a JSON array of five objects, each representing a day's weather forecast. The objects are:
 - `{ "date": "2023-12-13", "temperatureC": 38, "summary": "Warm", "temperatureF": 100 }`
 - `{ "date": "2023-12-14", "temperatureC": 6, "summary": "Hot", "temperatureF": 42 }`
 - `{ "date": "2023-12-15", "temperatureC": -4, "summary": "Sweltering", "temperatureF": 25 }`
 - `{ "date": "2023-12-16", "temperatureC": 47, "summary": "Bracing", "temperatureF": 116 }`
 - `{ "date": "2023-12-17", ... }` (partially visible)

At the bottom right of the response body, there are icons for copying and a **Download** button.

3. Create the docker image and we push in Azure ACR

IMPORTANT! We first have to run Docker Desktop

The screenshot shows the Docker Desktop application window. The top bar is blue with the 'Docker Desktop' logo, an 'Update to latest' button, a search bar, and user information 'luiscoco'. The left sidebar contains navigation links: 'Containers', 'Images', 'Volumes', 'Dev Environments' (marked BETA), 'Extensions', and 'Add Extensions'. The main content area is titled 'Containers' and includes a 'Give feedback' link. Below the title is a descriptive sentence about containers and a 'Learn more' link. A large blue 3D cube icon is centered. Below it is the heading 'Run a Sample Container'. A text prompt asks the user to try running a container by copying a command into a terminal. The command is displayed in a light gray box: `docker run -d -p 80:80 docker/getting-started`. Below the command box is a link to 'Explore more in the Docker Docs'. Further down is a 'Guides' section with two cards: 'Redis' and 'NGINX'. Each card has an icon, a description, and a 'Run' button. The Redis card describes it as an open-source in-memory key-value store. The NGINX card describes it as an open-source web server, reverse proxy, load balancer, and HTTP cache. The bottom status bar shows system metrics: 'RAM 3.59 GB', 'CPU 0.04%', and 'Connected to Hub'. On the right side of the status bar, it shows the version 'v4.17.1' and a Docker logo.

Docker Desktop

Update to latest

Search for local and remote images, containers, and more...

Ctrl+K

luiscoco

Containers

Give feedback

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

Run a Sample Container

Try running a container: Copy and paste this command into your terminal and then come back

```
docker run -d -p 80:80 docker/getting-started
```

[Explore more in the Docker Docs](#)

Guides

Redis

An open-source in-memory key-value store that functions as a data structure server.

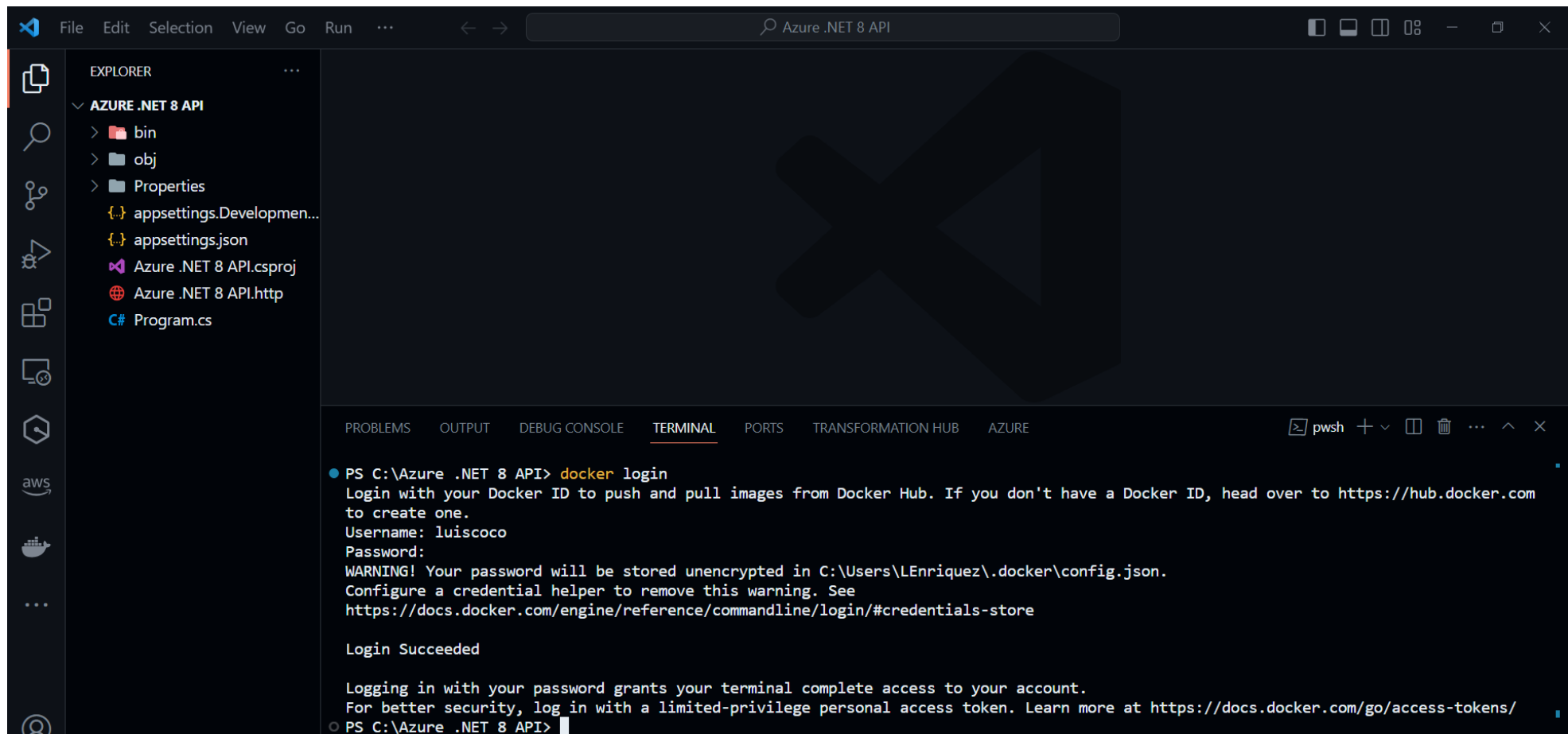
NGINX

An open-source web server, reverse proxy, load balancer and HTTP cache.

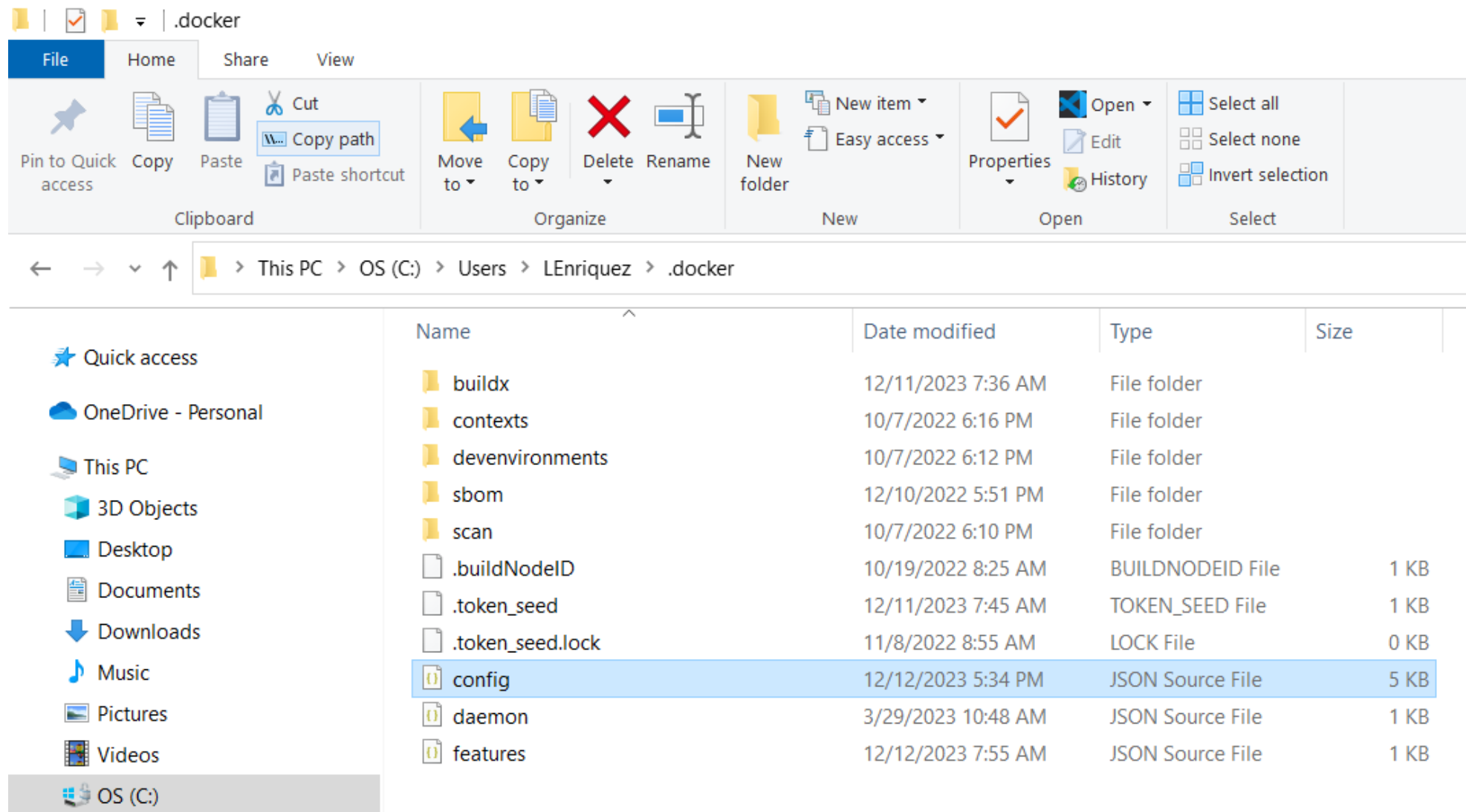
RAM 3.59 GB CPU 0.04% Connected to Hub

v4.17.1

Then we login in docker



If we want to reset the docker configuration, go to this path and delete the config.json file and then run the above command **docker login**



Now we login in Azure CLI

```
az login
```

We go to the Azure Portal and we activate the admin user

The screenshot shows the Azure portal interface for the 'mywebapicontainer' Container Registry. The 'Access keys' page is active, and the 'Admin user' checkbox is checked. The following table displays the generated passwords and their regeneration options:

Name	Password	Regenerate
password	1R8Ou4DmGX1TMQOWrFHSv3dC3vi968CSYipZy+595Q+A...	
password2	L+z/y/fauQayN9HwYOFQW/tmUnZB5SLb6vaDXACQ1+A...	

or we can activate the admin user with this command:

```
az acr update --name mywebapicontainer --resource-group myRG --admin-enabled true
```

Then we login in the Azure ACR and set the username and password in the admin user page

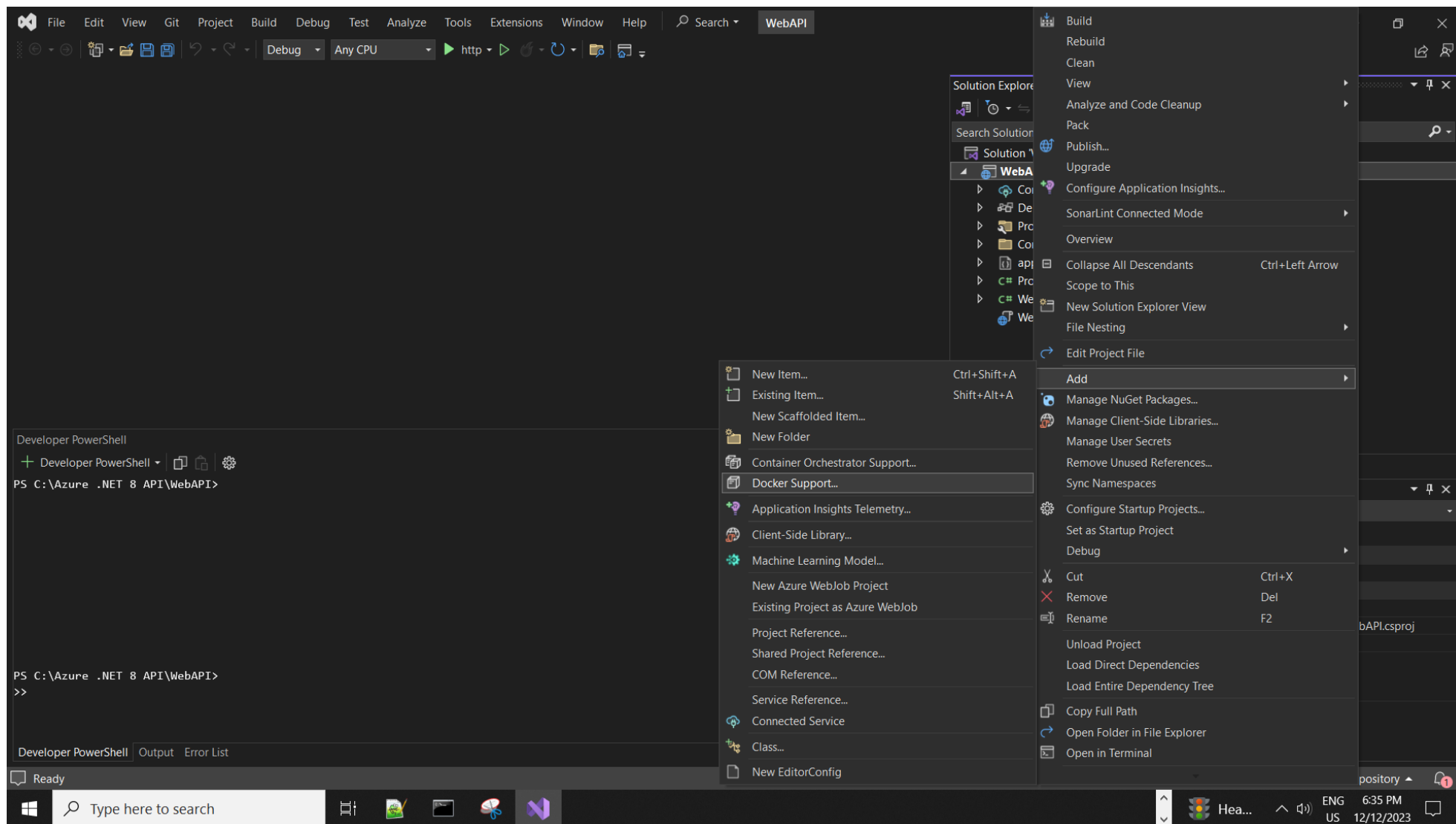
```
az acr login --name mywebapicontainer
```

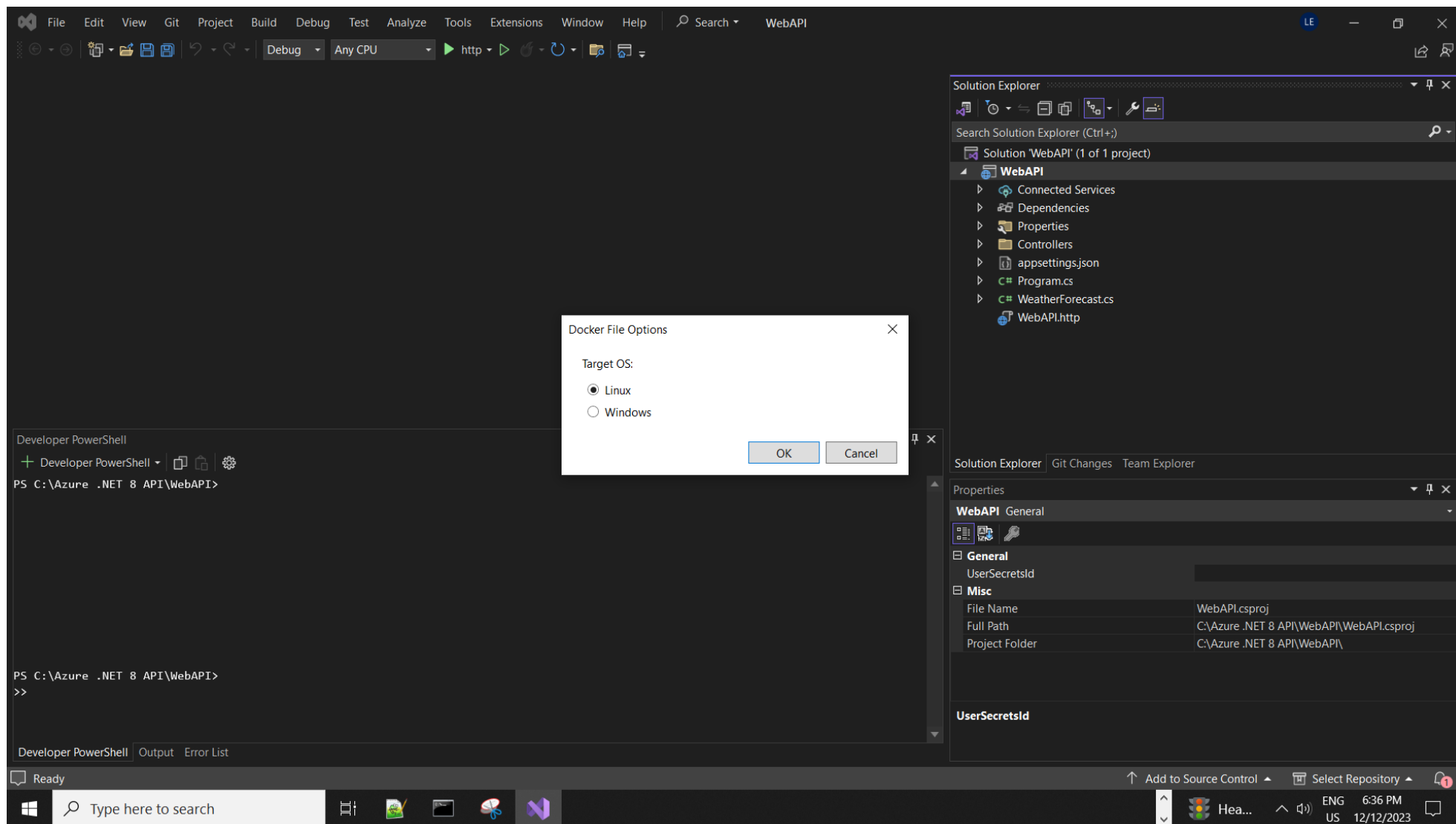
The screenshot shows the Microsoft Azure portal interface. On the left is a sidebar with navigation options like 'Create a resource', 'Home', 'Dashboard', 'All services', and 'FAVORITES'. The main area displays the 'mywebapicontainer' container registry's 'Access keys' page. It includes fields for 'Registry name' (mywebapicontainer), 'Login server' (mywebapicontainer.azurecr.io), and 'Admin user' (checked). Below these, the 'Username' is 'mywebapicontainer'. A table lists access keys with columns 'Name', 'Password', and 'Regenerate'. The first key, 'password', has a password starting with '1R8Ou4DmGX1TMQOWrFHSv3dC3vi968CSYipZy+595Q+A...' and is highlighted with a red box. The second key, 'password2', has a password starting with 'L+z/y/fauQayN9HwYOFQW/tmUnZB5SLb6vaDXACQ1+A...'.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS TRANSFORMATION HUB AZURE pwsh + - [ ] [X] ... ^ X

● PS C:\WebAPI> az acr login --name mywebapicontainer
Unable to get AAD authorization tokens with message: 2023-12-12 17:17:39.125509 An error occurred: CONNECTIVITY_REFRESH_TOKEN_ERROR
Access to registry 'mywebapicontainer.azurecr.io' was denied. Response code: 401. Please try running 'az login' again to refresh permissions.
Unable to get admin user credentials with message: The resource with name 'mywebapicontainer' and type 'Microsoft.ContainerRegistry/registries' could not be found in subscription 'Azure subscription 1 (846901e6-da09-45c8-98ca-7cca2353ff0e)'.
Username: mywebapicontainer
Password:
Login Succeeded
WARNING! Your password will be stored unencrypted in C:\Users\LEnriquez\.docker\config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
PS C:\WebAPI>
```

To automatically create the Dockerfile we add docker support to our application





This is the dockerfile generated by Visual Studio

#See <https://aka.ms/customizecontainer> to learn how to customize your debug container and how Visual Studio uses this Dockerfile

```
FROM mcr.microsoft.com/dotnet/aspnet:7.0 AS base
WORKDIR /app
EXPOSE 80
```

EXPOSE 443

```
FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build
ARG BUILD_CONFIGURATION=Release
WORKDIR /src
COPY ["WebAPIdotNet7.csproj", "."]
RUN dotnet restore ".././WebAPIdotNet7.csproj"
COPY . .
WORKDIR "/src/."
RUN dotnet build "../WebAPIdotNet7.csproj" -c $BUILD_CONFIGURATION -o /app/build

FROM build AS publish
ARG BUILD_CONFIGURATION=Release
RUN dotnet publish "../WebAPIdotNet7.csproj" -c $BUILD_CONFIGURATION -o /app/publish /p:UseAppHost=false

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "WebAPIdotNet7.dll"]
```

4. We build the Docker Image

```
docker build -t mywebapicontainer.azurecr.io/mywebapicontainer:v1 .
```

5. We push the Docker image to the Azure ACR

```
docker push mywebapicontainer.azurecr.io/mywebapicontainer:v1
```

5. We create the Azure Container Instance (ACI) and we deploy it

```
az container create --resource-group myRG --name mycontainerinstance --image mywebapicontainer.azurecr.io/mywebapicontainer:v1 --
```

6. In Azure Portal we navigate to Azure ACI

We navigate to the ACI container in Azure Portal

The screenshot shows the Microsoft Azure portal interface. The left sidebar contains navigation options like 'Create a resource', 'Home', 'Dashboard', 'All services', and 'FAVORITES'. The main area displays the 'myRG' resource group. The 'Resources' tab is active, showing a list of resources. The 'mycontainerinstance' resource is highlighted with a red box. The 'mywebapicontainer' resource is also visible. The 'Essentials' section shows subscription details and deployment status. The 'Resources' section shows a table of resources with columns for Name, Type, and Location.

Name	Type	Location
mycontainerinstance	Container instances	West Europe
mywebapicontainer	Container registry	West Europe

We copy the DNS and past it in the internet web browser: <http://mywebapidns.westeurope.azurecontainer.io/weatherforecast>

Microsoft Azure

Search resources, services, and docs (G+/)

Home > All resources >

Create a resource

Home

Dashboard

All services

FAVORITES

All resources

Resource groups

Quickstart Center

App Services

Function App

SQL databases

Azure Cosmos DB

Virtual machines

All resources

particular

Create Manage view

Filter for any field...

mycontainerinstance

mywebapicontainer

mycontainerinstance

Container instances

Search

Start Restart Stop Delete Refresh Give feedback

Overview

Activity log

Access control (IAM)

Tags

Settings

Containers

Identity

Properties

Locks

Essentials

Resource group (move) myRG

Status Running

Location West Europe

Subscription (move) Azure subscription 1

Subscription ID 846901e6-da09-45c8-98ca-7cca2353ff0e

Tags (edit) Add tags

SKU Standard

OS type Linux

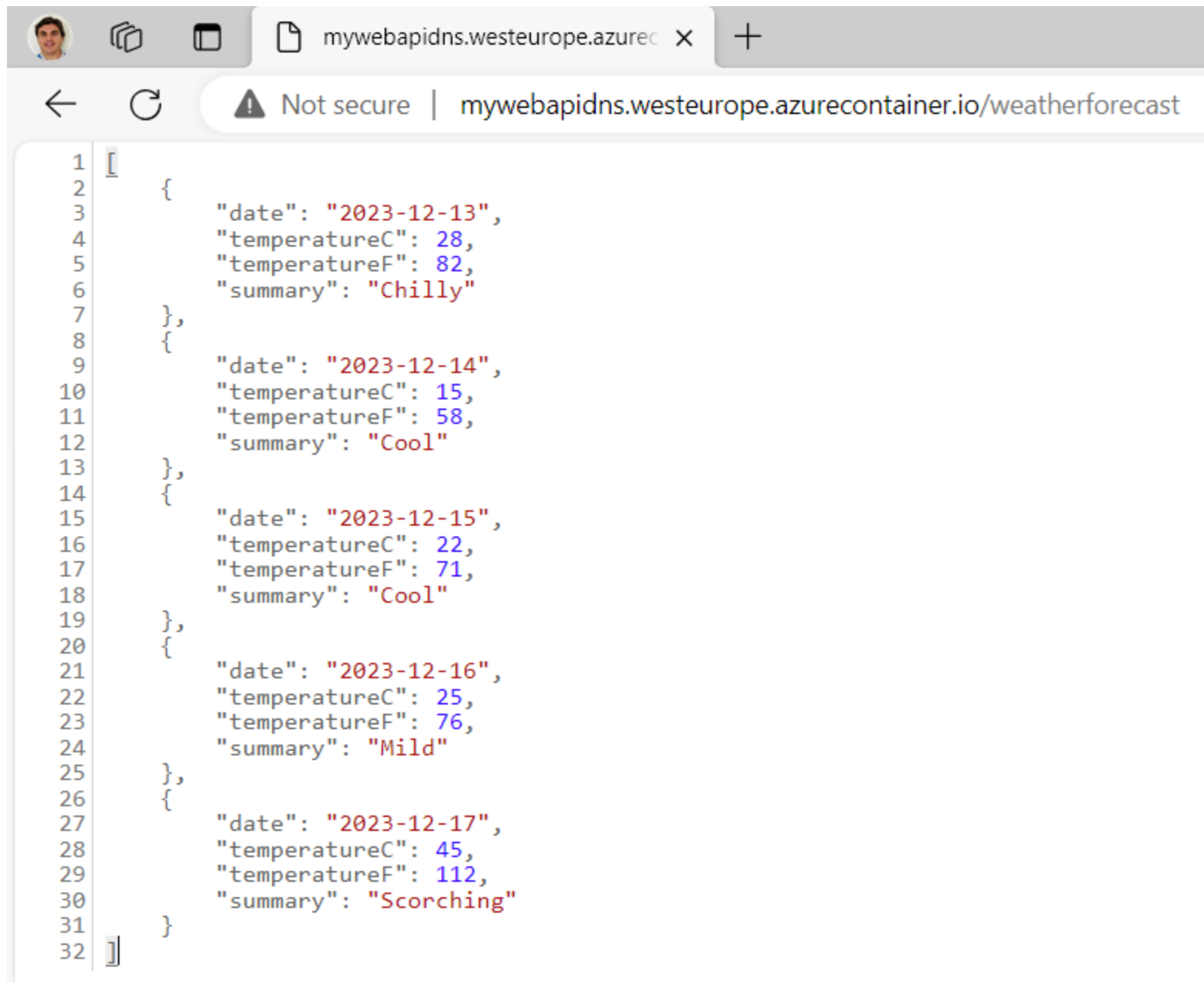
IP address (Public) 20.71.80.91

FQDN mywebapidns.westeurope.azurecontainer.io

Container count 1

JSON View

Copied



The screenshot shows a web browser window with a single tab titled "mywebapidns.westeurope.azurec...". The address bar displays "mywebapidns.westeurope.azurecontainer.io/weatherforecast" with a "Not secure" warning. The main content area shows a JSON array of five weather forecast objects, each representing a day from December 13th to December 17th, 2023. The JSON is syntax-highlighted with red for strings, blue for numbers, and black for punctuation. Line numbers 1 through 32 are visible on the left side of the code editor.

```
1 [
2   {
3     "date": "2023-12-13",
4     "temperatureC": 28,
5     "temperatureF": 82,
6     "summary": "Chilly"
7   },
8   {
9     "date": "2023-12-14",
10    "temperatureC": 15,
11    "temperatureF": 58,
12    "summary": "Cool"
13  },
14  {
15    "date": "2023-12-15",
16    "temperatureC": 22,
17    "temperatureF": 71,
18    "summary": "Cool"
19  },
20  {
21    "date": "2023-12-16",
22    "temperatureC": 25,
23    "temperatureF": 76,
24    "summary": "Mild"
25  },
26  {
27    "date": "2023-12-17",
28    "temperatureC": 45,
29    "temperatureF": 112,
30    "summary": "Scorching"
31  }
32 ]
```