



Máster en Ingeniería MultiCloud, DevOps y Seguridad.

AZURE LAB #9a

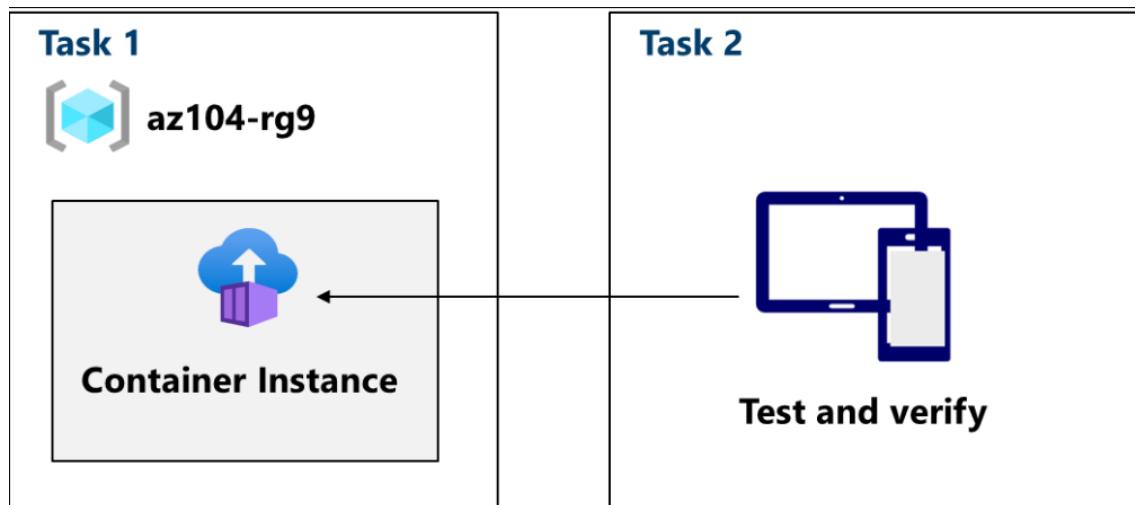
Implementación de containers en Azure



Contenido

Esquema del laboratorio	3
Desplegar un container utilizando una imagen de Docker.	4
Comprobando el despliegue del contenedor.....	6

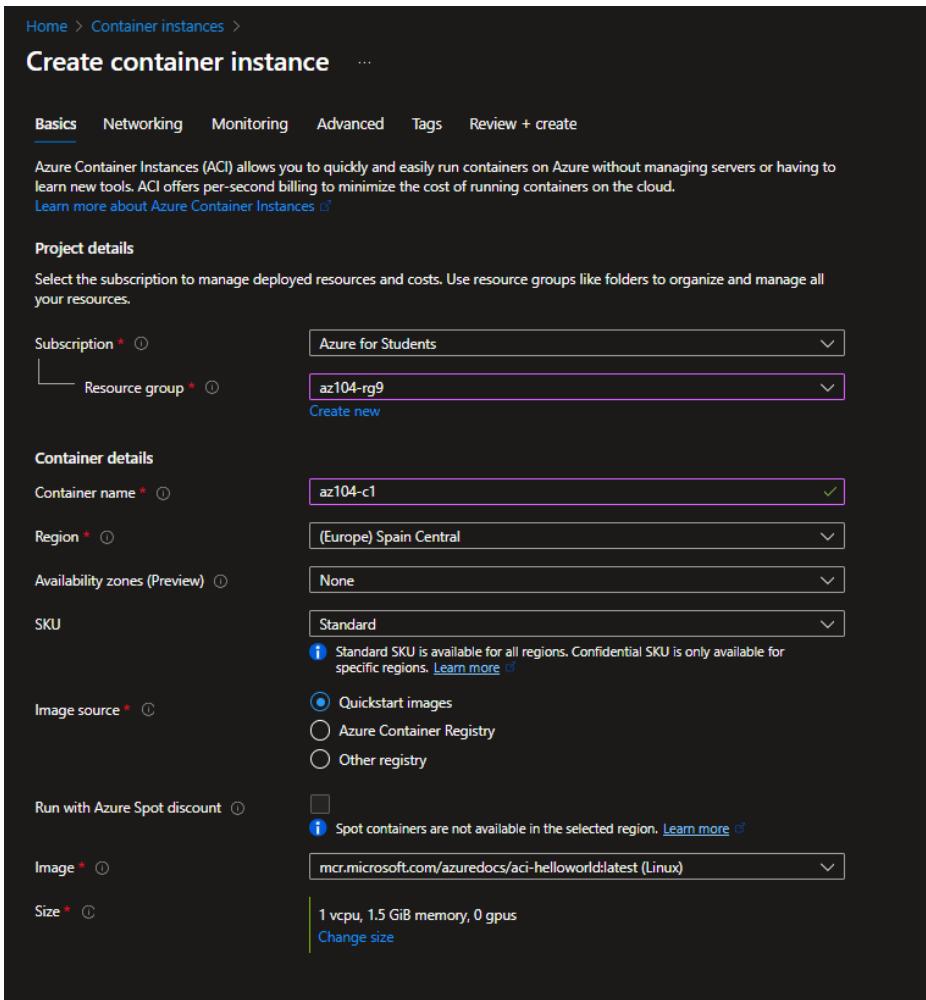
Esquema del laboratorio



 tajamar.	Máster en Ingeniería MultiCloud, DevOps y Seguridad.
AZURE LAB #9a	

Desplegar un container utilizando una imagen de Docker.

Azure Container Instances (ACI) es un servicio de contenedores sin servidor y a petición que te permite ejecutar contenedores de Docker de manera rápida y sencilla en Azure, sin la necesidad de administrar máquinas virtuales.



The screenshot shows the 'Create container instance' wizard in the Azure portal. The 'Basics' tab is selected. The 'Project details' section includes a subscription dropdown set to 'Azure for Students' and a resource group dropdown set to 'az104-rg9'. The 'Container details' section includes a container name 'az104-ci', region '(Europe) Spain Central', availability zones 'None', and SKU 'Standard'. The 'Image source' section has 'Quickstart images' selected. The 'Run with Azure Spot discount' checkbox is unchecked. The 'Image' dropdown is set to 'mcr.microsoft.com/azuredocs/aci-helloworld:latest (Linux)'. The 'Size' dropdown shows '1 vcpu, 1.5 GiB memory, 0 gpus' with a 'Change size' link.

Accedemos al servicio de container instances.

Vamos a crear un nuevo contenedor con una imagen de un Linux que nos despliega un servidor web con un bienvenido a container instances o algo parecido.



Create container instance

Basics Networking Monitoring Advanced Tags Review + create

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 Tenant

Ports

Ports	Ports protocol
80	TCP
<input type="text"/>	<input type="text"/>

Opciones de Tipo de Red

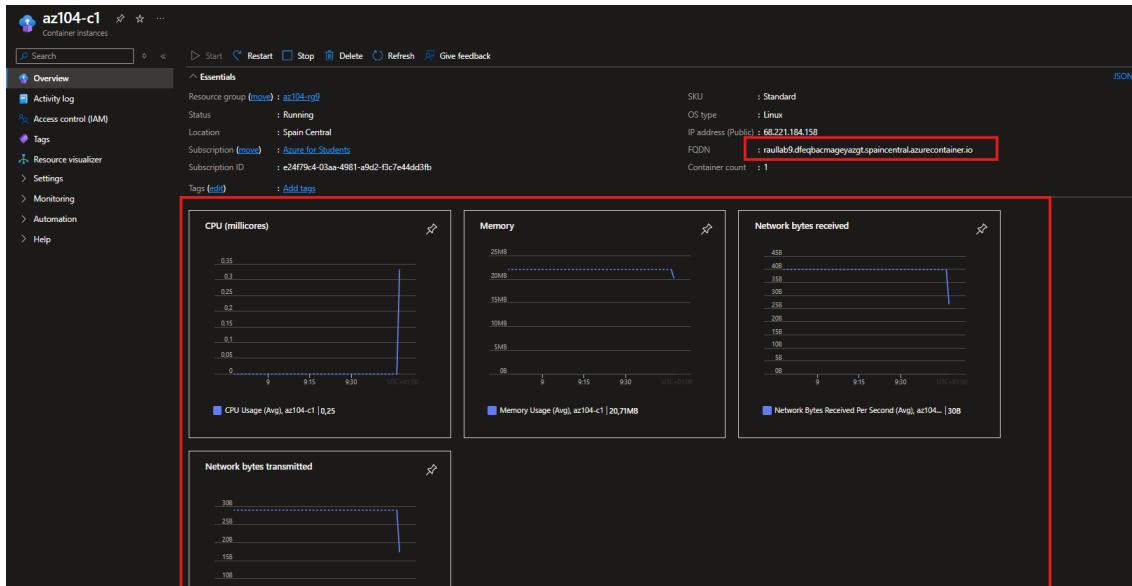
Pública (Public): Al seleccionar esta opción, Azure aprovisiona automáticamente una dirección IP pública para la instancia de contenedor. Este método hace que la aplicación o servicio dentro del contenedor sea accesible directamente desde Internet.

La configuración de la Etiqueta de Nombre DNS (DNS name label), solo es relevante cuando se selecciona el tipo de red Pública. Esta etiqueta es un prefijo que, al combinarse con la región de Azure, genera un Nombre de Dominio Completo (FQDN) único y accesible públicamente, con el formato <etiqueta-dns>.



AZURE LAB #9a

Comprobando el despliegue del contenedor.



Una vez desplegado accedemos al recurso, donde podemos ver unas métricas de CPU, memoria, tráfico de red...

Copio el FQDN para comprobar via web.

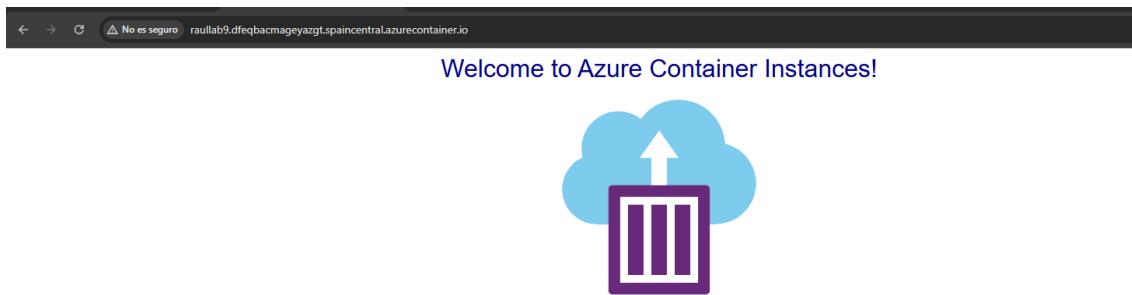


Imagen.



AZURE LAB #9a

The screenshot shows the Azure Container Instances blade for a container named 'az104-c1'. The 'Logs' tab is selected, with a red box highlighting it. The log output shows several GET requests from the IP address ::ffff:10.92.0.10 to port 80, indicating multiple reloads of the browser. The log entries are as follows:

```
listening on port 80
::ffff:10.92.0.9 - [11/Oct/2025:08:47:57 +0000] "GET / HTTP/1.1" 200 1696 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
::ffff:10.92.0.9 - [11/Oct/2025:08:47:58 +0000] "GET /favicon.ico HTTP/1.1" 404 158 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
::ffff:10.92.0.10 - [11/Oct/2025:08:49:17 +0000] "GET / HTTP/1.1" 200 1696 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
::ffff:10.92.0.10 - [11/Oct/2025:08:49:17 +0000] "GET /favicon.ico HTTP/1.1" 404 158 "http://caullana.diegobarangaygt.apicentral.azurecontainer.io/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
::ffff:10.92.0.10 - [11/Oct/2025:08:53:10 +0000] "GET / HTTP/1.1" 304 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
::ffff:10.92.0.10 - [11/Oct/2025:08:53:11 +0000] "GET / HTTP/1.1" 304 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
::ffff:10.92.0.10 - [11/Oct/2025:08:53:12 +0000] "GET / HTTP/1.1" 304 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
::ffff:10.92.0.10 - [11/Oct/2025:08:53:12 +0000] "GET / HTTP/1.1" 304 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
::ffff:10.92.0.10 - [11/Oct/2025:08:53:13 +0000] "GET / HTTP/1.1" 304 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36"
```

He hecho varias peticiones a la web recargando la página, dentro del recurso > settings > containers > logs se pueden apreciar las request GET que he hecho