

# JÓVENES BICENTENARIO 3.0 – CERTUS

Docente: EDWIN MARAVI PÉREZ

POR: CARRION HUACANI GEAN CARLO

## ACTIVIDAD ASINCRÓNICA N° 05

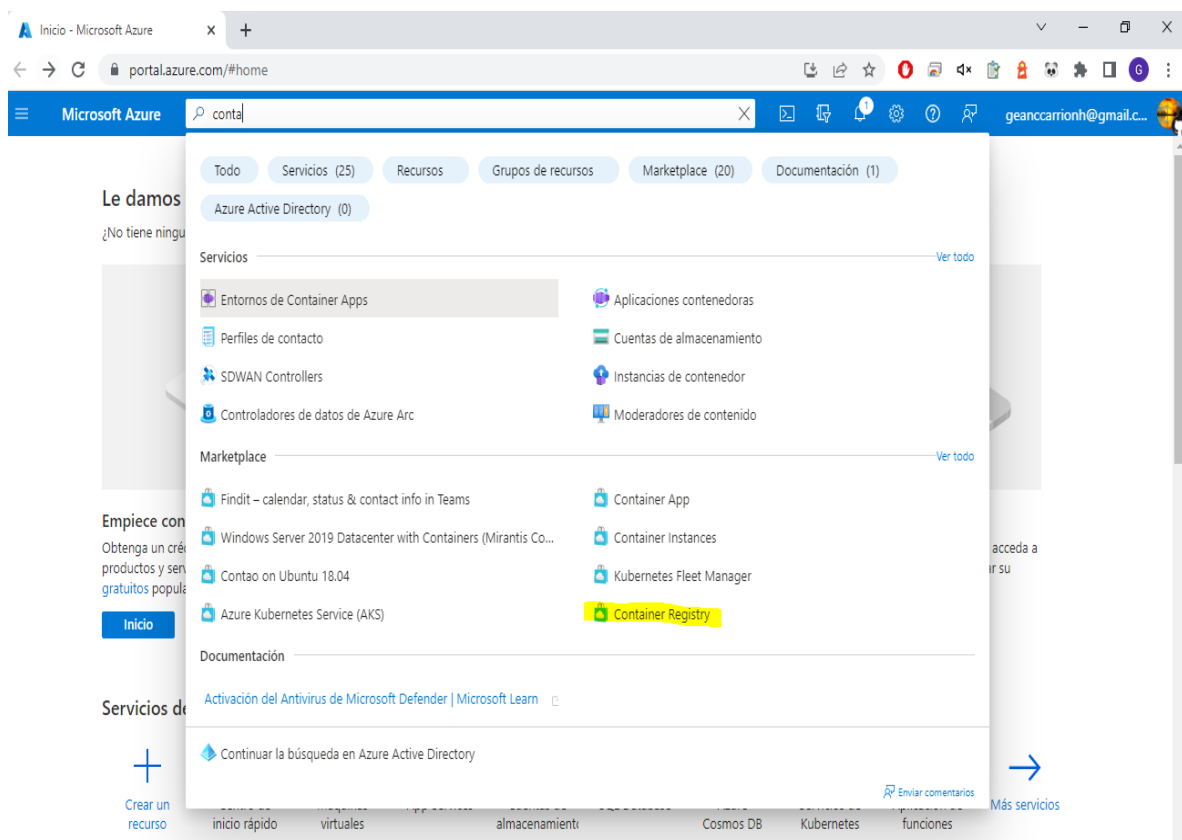
### Azure Registry

Procedimiento para subir una imagen a Azure Registry.

La primera vez que se comparte una imagen seguramente sea a través de los repositorios públicos de Docker Hub. Sin embargo, está claro que, en la mayoría de los escenarios, sobre todo los empresariales, se necesita algo privado, no compartido con el mundo entero. Hay diferentes opciones de registros privados, como el propio Docker Hub, crearte uno en local o usar alguna de las nubes.

#### Crear registro privado en Microsoft Azure

Para crear un registro privado en Microsoft Azure ingresar a <https://portal.azure.com/#home> y busca Container a través del botón Create a resource del menú. Selecciona el servicio Container Registry.



#### *Create a resource – Azure Container Registry*

Para este ejemplo he utilizado el SKU Basic, ya que es más que suficiente.

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 \*

Visual Studio Enterprise Subscription

Resource group \*

YuriISPlayground

Create new

Instance details

Registry name \*

YuriISProgrammersWeek2021CR

.azurecr.io

Location \*

East US

Availability zones

☐ Enabled
 

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

SKU \*

Standard

Review + create

< Previous

Next: Networking >

Create container registry

Validation passed

Basics

Networking

Encryption

Tags

Review + create

Registry details

Basics

Registry name

YuriISProgrammersWeek2021CR

Subscription

Visual Studio Enterprise Subscription

Resource Group

YuriISPlayground

Location

East US

Availability zones

Disabled

SKU

Standard

Networking

Allow public network access

Yes

Encryption

Customer-Managed Key

Disabled

Identity

None

Key Vault

None

Encryption key

None

Version

None

Create

< Previous

Next >

Download a template for automation

### Create container registry

Lo primero que se necesita es subir tu imagen de nodejs-webapp al registro, en este caso en Azure. A diferencia de Docker Hub, para poder conectarte a este nuevo registro necesitas tener instalado **Azure CLI** y ejecutar el siguiente comando:

```
az login
```

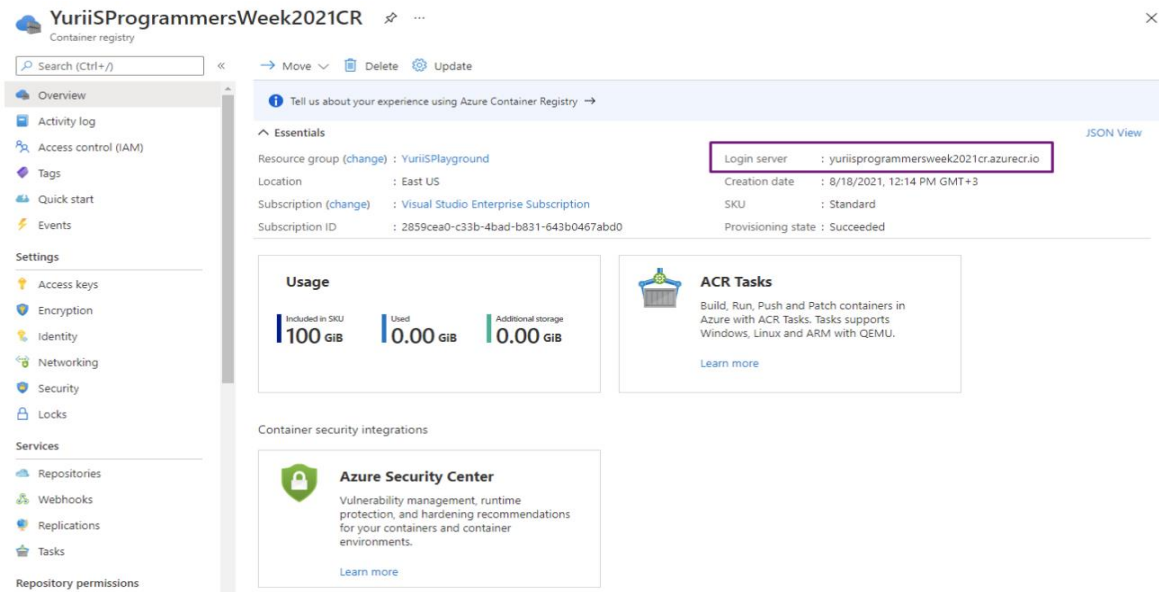
Si tienes más de una suscripción, selecciona aquella en la que has creado el servicio a través de este otro comando:

```
az account set -s SUBSCRIPTION_ID
```

Por último, haz login en tu Azure Container Registry, indicando como name el nombre que elegiste durante la creación:

```
az acr login --name returngis
```

Si el inicio de sesión ha sido satisfactorio aparecerá el mensaje de **Login Succeeded**, antes de subir tu imagen a cualquier registro es necesario utilizar una nomenclatura específica. En el caso de Docker Hub era tu\_nombre\_de\_usuario/repositorio:etiqueta. En el caso de Azure Container Registry es nombre\_del\_servidor\_de\_registro/repositorio:etiqueta. El nombre de tu servidor puedes encontrarlo aquí:



### Azure Container Registry – Login Server

El comando que tengo que lanzar para generar una imagen con la nomenclatura correcta sería:

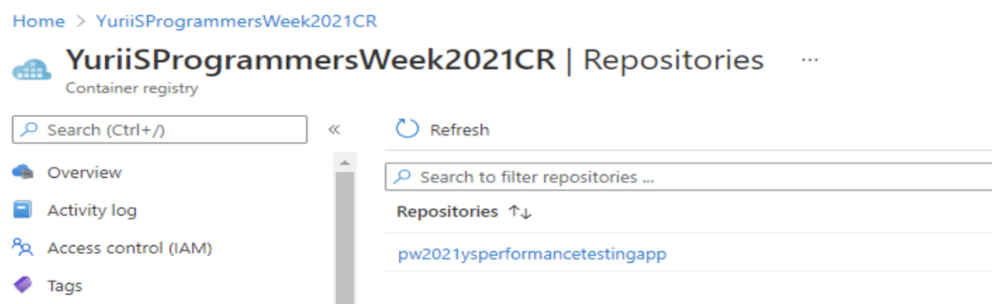
```
docker tag nodejs-webapp returngis.azurecr.io/nodejs-webapp:v1
```

Ahora ya puedo publicar mi imagen en Azure Container Registry, a través de docker push.

```
docker push returngis.azurecr.io/nodejs-webapp:v1
```

Cuando finalice la subida podrás ver la imagen en el apartado **Repositories**.

```
PS C:\Users\Yurii Skolozdra\Desktop\Azure VS Programmers Week\PA2021YSPerformanceTestingApp> docker tag pw2021ysperformentestingapp yuriiprogrammersweek2021cr.azurecr.io/pw2021ysperformentestingapp
PS C:\Users\Yurii Skolozdra\Desktop\Azure VS Programmers Week\PA2021YSPerformanceTestingApp> docker push yuriiprogrammersweek2021cr.azurecr.io/pw2021ysperformentestingapp
Using default tag: latest
The push refers to repository [yuriiprogrammersweek2021cr.azurecr.io/pw2021ysperformentestingapp]
adcc6722a22f: Pushed
55bb6a58772c: Pushed
7663e41a9c47: Pushed
ada5e91e138: Pushed
7e7e452bf676: Pushed
99744128ede5: Pushed
f68ef921efae: Pushed
latest: digest: sha256:28567fb4f33287d890b4bcd6b975a587fa8e80a5c2bb53edec2f8c93c30e4c98 size: 1793
PS C:\Users\Yurii Skolozdra\Desktop\Azure VS Programmers Week\PA2021YSPerformanceTestingApp>
```



### Imagen nodejs-webapp en Azure Container Registry

Por último, queda comprobar que es posible descargar e instanciar la imagen en un contenedor en local. Para hacer la prueba, vamos a parar todos los contenedores que tengamos ejecutándose:

```
docker kill $(docker ps -q)
```

Una vez parados los eliminamos:

```
docker rm $(docker ps -a -q)
```

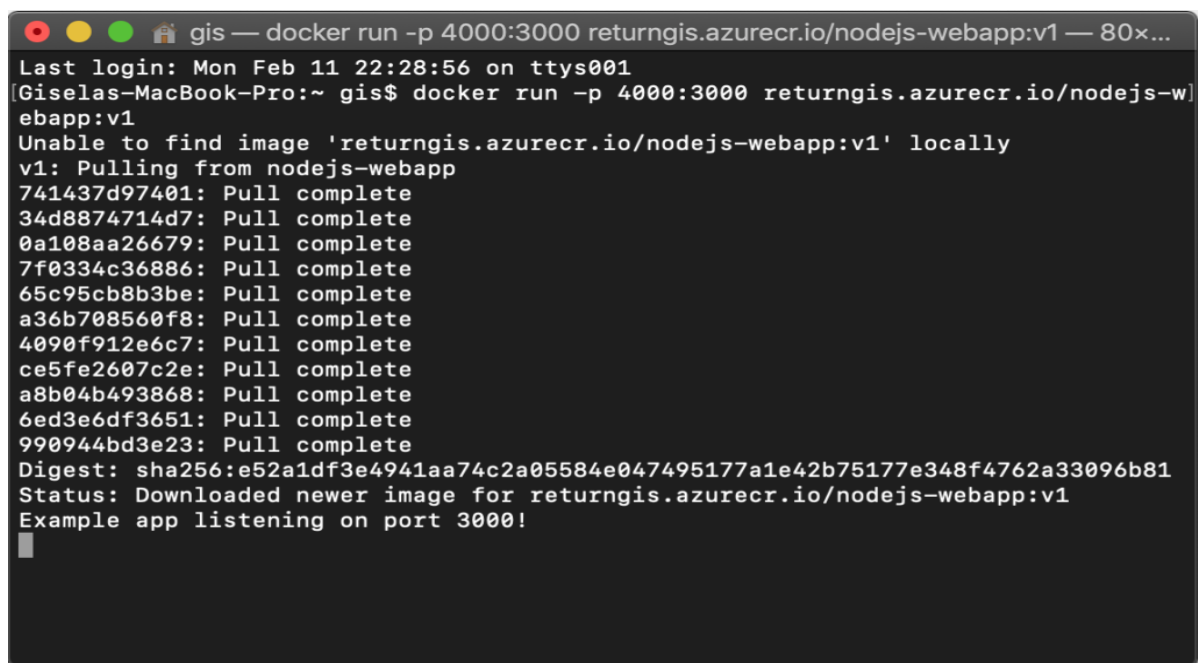
Y eliminamos además todas las imágenes.

```
docker rmi -f $(docker images -q)
```

Ahora, con todo limpiísimo, ejecuta el siguiente comando para descargar e instanciar tu imagen, que ya está en Azure Container Registry.

```
docker run -p 4000:3000 returngis.azurecr.io/nodejs-webapp:v1
```

El resultado será el siguiente:



```
gis — docker run -p 4000:3000 returngis.azurecr.io/nodejs-webapp:v1 — 80x...
Last login: Mon Feb 11 22:28:56 on ttys001
[Giselas-MacBook-Pro:~ gis$ docker run -p 4000:3000 returngis.azurecr.io/nodejs-w
ebapp:v1
Unable to find image 'returngis.azurecr.io/nodejs-webapp:v1' locally
v1: Pulling from nodejs-webapp
741437d97401: Pull complete
34d8874714d7: Pull complete
0a108aa26679: Pull complete
7f0334c36886: Pull complete
65c95cb8b3be: Pull complete
a36b708560f8: Pull complete
4090f912e6c7: Pull complete
ce5fe2607c2e: Pull complete
a8b04b493868: Pull complete
6ed3e6df3651: Pull complete
990944bd3e23: Pull complete
Digest: sha256:e52a1df3e4941aa74c2a05584e047495177a1e42b75177e348f4762a33096b81
Status: Downloaded newer image for returngis.azurecr.io/nodejs-webapp:v1
Example app listening on port 3000!
```

*docker run -p 4000:3000 returngis.azurecr.io/nodejs-webapp:v1*

Se puede comprobar que la aplicación es accesible desde <http://localhost:4000>

Por último, para comprobar que todas la imágenes se tiene disponible en el repositorio privado puedes hacerlo desde el siguiente comando:

```
az acr repository list -n returngis -o table
```