

Semana 4

DevOps (ISY2201)

Guía paso a paso para trabajar imágenes de contenedores en ACR y despliegue de la imagen en Docker en Máquina Virtual a través de Azure DevOps.

Profesor: Ignacio Pastenet	Autor: Ignacio Pastenet
Asignatura: DevOps	Carrera: Ingeniería en Desarrollo de Software
Versión: 1.0	Fecha: 19-08-2024

Contenido

Descripción de la actividad	3
Instrucciones específicas	3
Requisitos técnicos	4
Evidencias y ejecución paso a paso	5
Evidencias Paso 1: Validación de máquina virtual creada y grupo de recurso en Azure Portal.....	5
Evidencias Paso 2: Creación de imagen Docker y subir a Azure Container Registry (ACR)6	
Evidencias Paso 3: Preparar ambiente y ejecución del Pipeline para despliegue de imagen Docker en máquina virtual.....	13
Actividad Finalizada.....	20

Descripción de la actividad

Siguiendo con el aprendizaje anterior de creación de máquina virtual y su entorno de red virtual de forma automatizada a través de Azure DevOps, ocuparemos la misma máquina virtual, por lo que la actividad contempla la configuración de contenedores en una máquina virtual para lograr la segmentación de aplicaciones, lo que resultará en la independencia de códigos y tiempos de ejecución. Además, deberás utilizar repositorios de contenedores con el propósito de centralizar imágenes, optimizando así el manejo de recursos. Esta guía incluye el paso a paso que te ayudara a crear una imagen de contenedor y subir a tu ACR para posteriormente configurar y ejecutar en Docker la imagen en una máquina virtual Linux.

Instrucciones específicas

Deberán realizar lo siguiente:

***PASO1: Validación previa**

1. Validar que se este ejecutando la máquina virtual en Azure Portal
2. Si no esta en ejecución se debe desplegar nuevamente a través del pipeline de despliegue de infraestructura realizado en la experiencia n°1.
3. Confirmar la IP pública de la máquina virtual.
4. Nota: Si esta en ejecución el apache deben bajarlo con (sudo systemctl stop apache2), esto para que no choque el puerto del nginx en puerto 80, o si bien cambiar el en la imagen y ejecución del Docker.
5. Contar con un grupo de recursos creado, ideal mismo donde reside la maquina virtual.

***PASO 2: Creación de imagen Docker y subir a Azure Container Registry (ACR)**

1. Creación de Registro de contenedor (ACR).
2. Creación de imagen Docker (Dockerfile) “ej. Imagen nginx” y sus archivos de configuración default.conf y index.html.
3. Creación de código pipeline en yaml para subir imagen Dockerfile a ACR.
4. Subir archivos a GitHub: Dockerfile, default.conf, index.html en una estructura de directorio.
5. Subir archivo código pipeline a GitHub para subir imagen Dockerfile hacia ACR.
6. Crear Service Connections Docker Registry de tipo Azure Container Registry.
7. Creación y Ejecución de Azure Pipeline con código YAML desde repositorio GitHub para subir imagen Dockerfile.

8. Evidencias de la ejecución del pipeline en Azure Portal sección Registros de contenedor y validar que este la imagen subida.

***PASO 3: Preparar ambiente y ejecución del Pipeline para despliegue de imagen Docker en maquina virtual.**

1. Crear Service Connections con Azure Resource Manager y el grupo de recurso.
2. Crear un entorno en Azure pipeline para registrar la máquina virtual y dar permisos al repositorio pipeline.
3. Creación de código pipeline en yaml para instalar Docker, azure cli y hacer pull y run de imagen en ACR.
4. Subir archivo código pipeline a GitHub para posterior ejecución en Azure Pipeline.
5. Creación y Ejecución de Azure Pipeline con código YAML desde repositorio GitHub para instalar y ejecutar Docker con la imagen nginx y página de bienvenida.
6. Revisión de la ejecución del job del pipeline.
7. Ejecutar IP publica de la máquina virtual en navegador web.
8. Evidencias de la ejecución del pipeline desde máquina virtual como ejecución de Docker ps.

Requisitos técnicos

Contar con las siguientes plataformas y herramientas:

1. Cuenta duoc registrada en Azure Portal.
2. Cuenta duoc registrada en Azure DevOps
3. Cuenta duoc o personal en GitHub (Repositorio de código) ó usar Azure Repos
4. Contar con la herramienta Putty para acceso a la máquina Linux.
5. Contar con la instalación de la herramienta Azure Cli en la laptop para validaciones extras.
6. Contar con la instalación de Docker desktop en la laptop para validaciones extras.

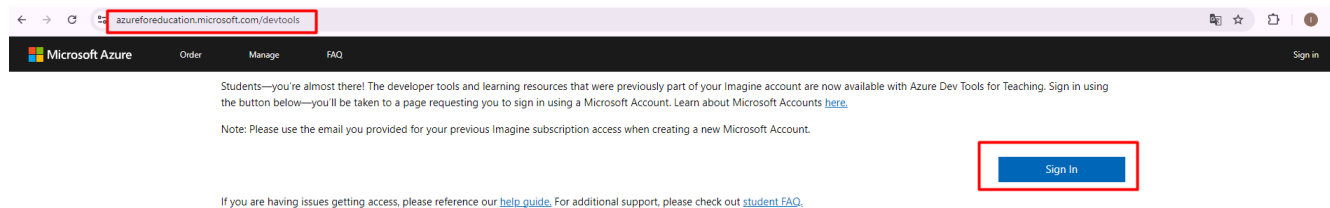
Nota: La región seleccionada debe ser EAST US 2 para todos los casos de despliegue de servicios en Azure Portal.

Evidencias y ejecución paso a paso

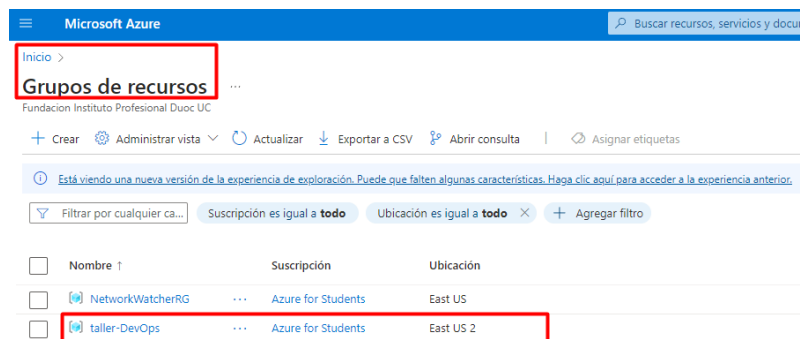
Nota: La región que se debe seleccionar es la **EAST US 2**, para todo el aprovisionamiento de recursos cloud en **Azure Portal**.

Evidencias Paso 1: Validación de máquina virtual creada y grupo de recurso en Azure Portal

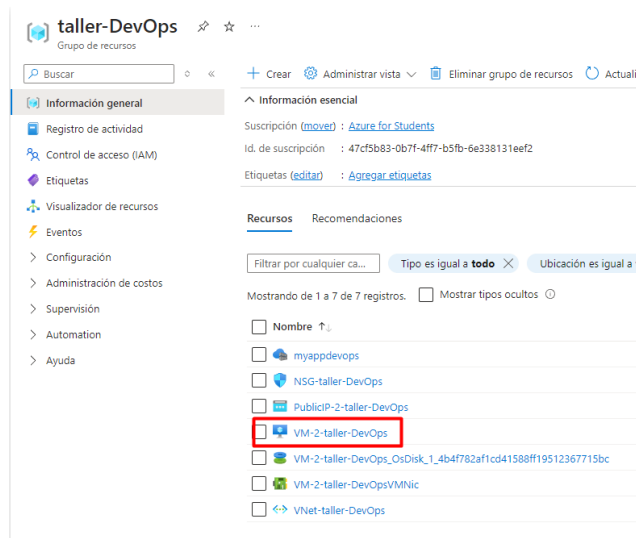
- Login en azure portal
- URL: <http://aka.ms/devtoolsforteaching>
- Ingresar con su cuenta DUOC



- Validación grupo de recurso creado



- Validación máquina virtual creada en grupo de recurso asignado.



- Validación IP Pública

Microsoft Azure

Inicio > Grupos de recursos > taller-DevOps > VM-2-taller-DevOps

VM-2-taller-DevOps | Conectar ☆ ...

Máquina virtual

Buscar

Actualizar Solucionar problemas Más opciones Comentarios

Conectándose mediante

Dirección IP pública | 40.123.37.134

Nombre de usuario del administrador : azureuser

Puerto (cambiar) : 22 Comprobación de acceso

Directiva Just-In-Time : No compatible con el plan

- Validar ejecución de apache2 y bajada del servicio.

```

azureuser@VM-2-taller-DevOps: ~
Last login: Sat Aug 31 08:18:49 2024 from 181.42.132.63
azureuser@VM-2-taller-DevOps:~$ ps -fea |grep apache
root        617      1   0  08:07 ?        00:00:02 /usr/sbin/apache2 -k start
www-data    619     617   0  08:07 ?        00:00:00 /usr/sbin/apache2 -k start
www-data    620     617   0  08:07 ?        00:00:00 /usr/sbin/apache2 -k start
azureus+   18899  17741   0  22:06 pts/0    00:00:00 grep --color=auto apache
azureuser@VM-2-taller-DevOps:~$ sudo systemctl stop apache2

```

Evidencias Paso 2: Creación de imagen Docker y subir a Azure Container Registry (ACR)

- Creación de Registro de contenedor (ACR).

Microsoft Azure

Buscar recursos, servicios y documentos (G+)

Inicio >

Registros de contenedor

Fundacion Instituto Profesional Duoc UC (duoc.cl)

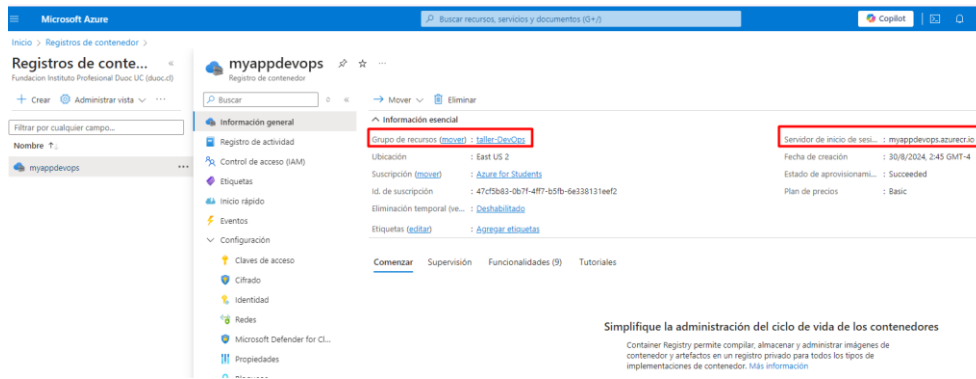
+ Crear Administrar vista Actualizar Exportar a CSV Abrir consulta Asignar etiquetas

Filtrar por cualquier ca... Suscripción es igual a todo Grupo de recursos es igual a todo Ubicación es igual a todo Agregar filtro

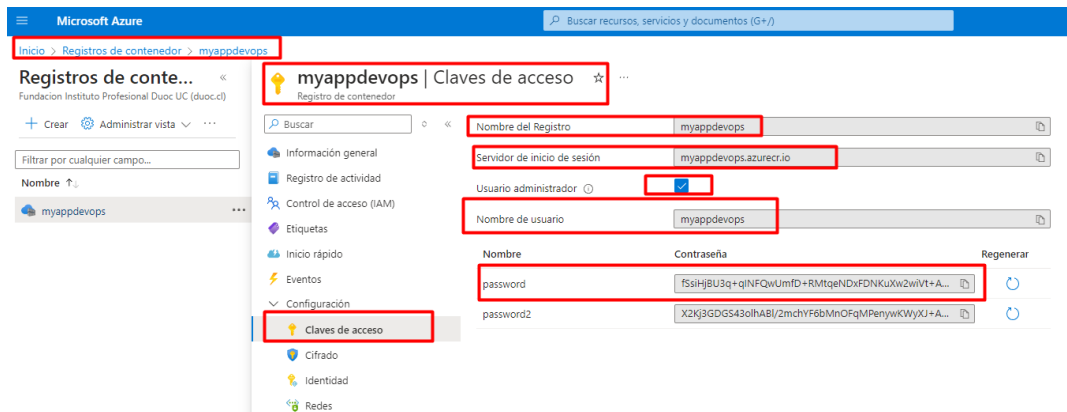
Mostrando de 1 a 1 de 1 registros.

Nombre	Tipo	Grupo de recursos	Ubicación	Suscripción
myappdevops	Registro de contenedor	taller-DevOps	East US 2	Azure for Students

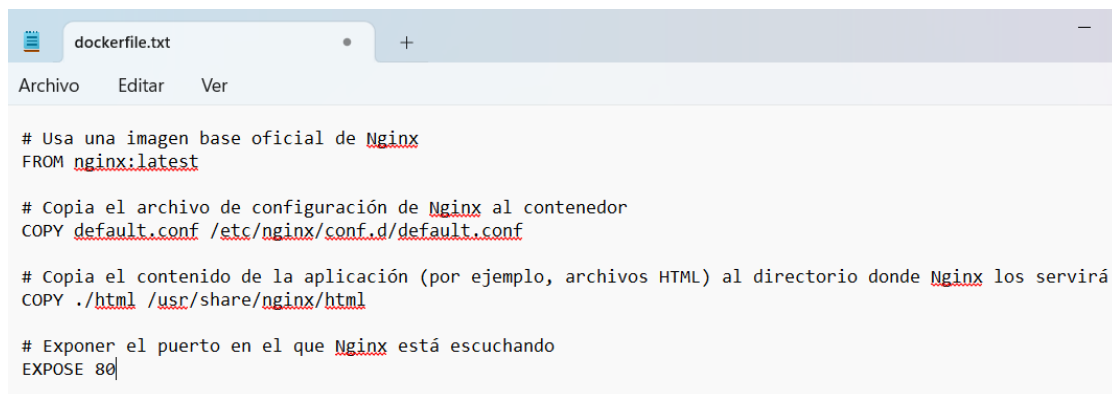
- Detalles de la creación del registro del contenedor.



- Habilitar claves de acceso necesarias para la configuración por pipeline en código de despliegue.



- Creación de imagen Docker (Dockerfile) “ej. Imagen nginx” y sus archivos de configuración default.conf y index.html.
- Archivo Dockerfile



Dockerfile

- Archivo default.conf

```

server {
    listen 80;
    server_name localhost;

    root /usr/share/nginx/html;
    index index.html index.htm;

    location / {
        try_files $uri $uri/ =404;
    }

    error_page 404 /404.html;
    location = /404.html {
        internal;
    }
}

```



default.conf

- Archivo index.html

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mi Aplicación Web</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      text-align: center;
      padding: 50px;
    }
    h1 {
      color: #333;
    }
    p {
      color: #666;
    }
  </style>
</head>
<body>
  <h1>¡Bienvenido a Mi Aplicación Web!</h1>
  <p>Esta es una página de ejemplo servida por Nginx dentro de un contenedor Docker.</p>
  <p>Todo está funcionando correctamente si ves esta página.</p>
</body>
</html>

```



index.html

- Creación de código pipeline en yaml para subir imagen Dockerfile a ACR.

```

trigger:
  branches:
    include:
      - main

variables:
  # Variables de entorno
  ACR_NAME: 'myappdevops' # Colocar el nombre de tu ACR
  IMAGE_NAME: 'myapp-nginx-s4' # Colocar el Nombre de Imagen
  SERVICE_CONNECTION_NAME: 'myapps-devops-s4' # Colocar nombre del Service Connection ubicado en Docker Registry--
  Azure Container Registry

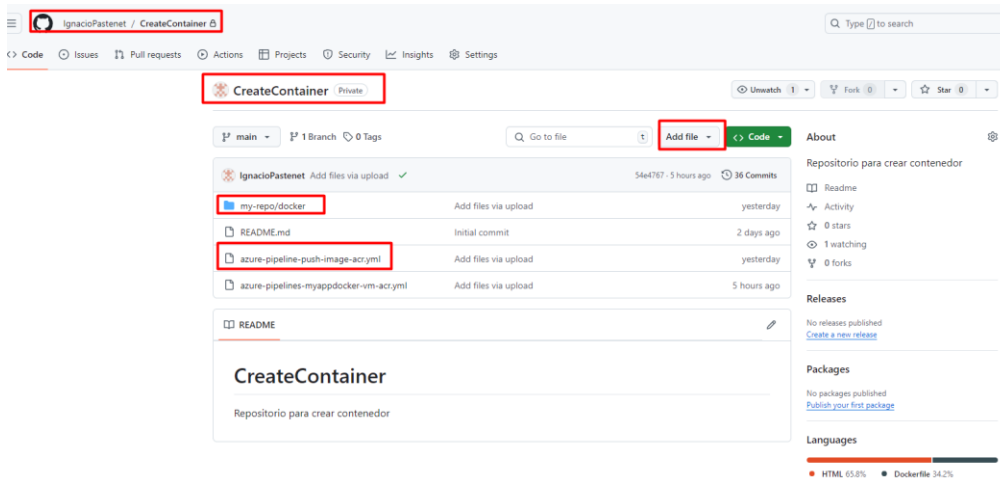
jobs:
  - job: BuildAndPush
    pool:
      vmImage: 'ubuntu-latest'
    steps:
      - task: Docker@2
        inputs:
          command: 'buildAndPush'
          containerRegistry: '$(SERVICE_CONNECTION_NAME)'
          repository: '$(ACR_NAME).azurecr.io/$(IMAGE_NAME)'
          dockerfile: '**/Dockerfile'
          tags: |
            latest
          displayName: 'Build and Push Docker Image'

```

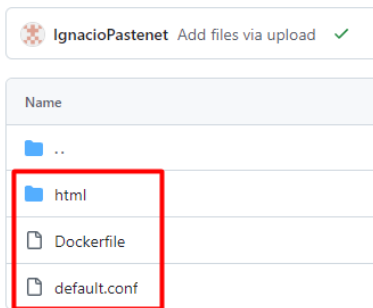


azure-pipeline-push-image-acr.yml

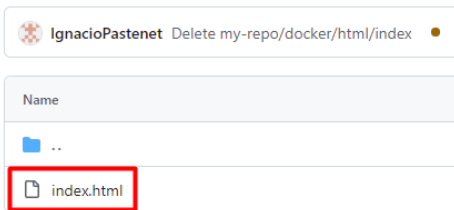
- Crear repositorio y subir archivos a GitHub: Dockerfile, default.conf, index.html en una estructura de directorio.



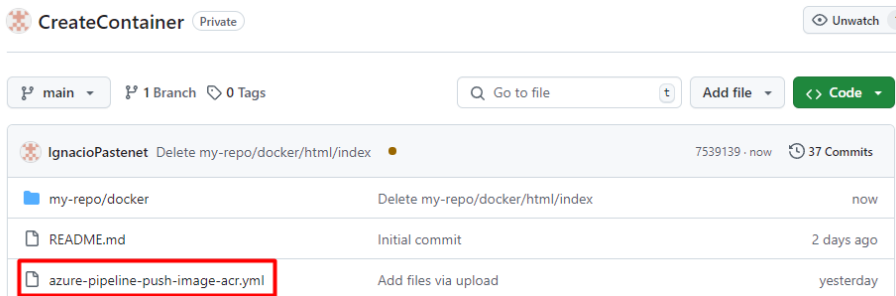
CreateContainer / my-repo / docker /



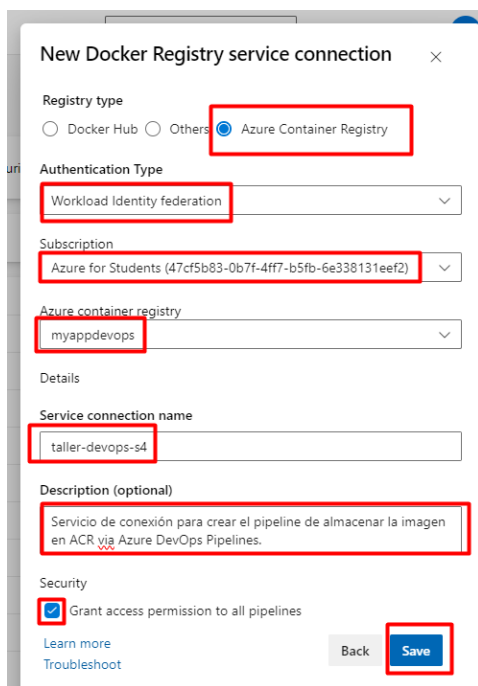
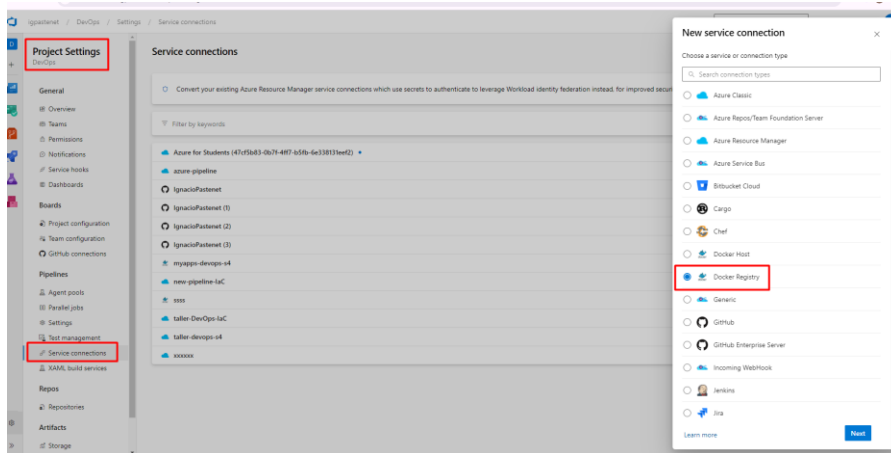
CreateContainer / my-repo / docker / html /



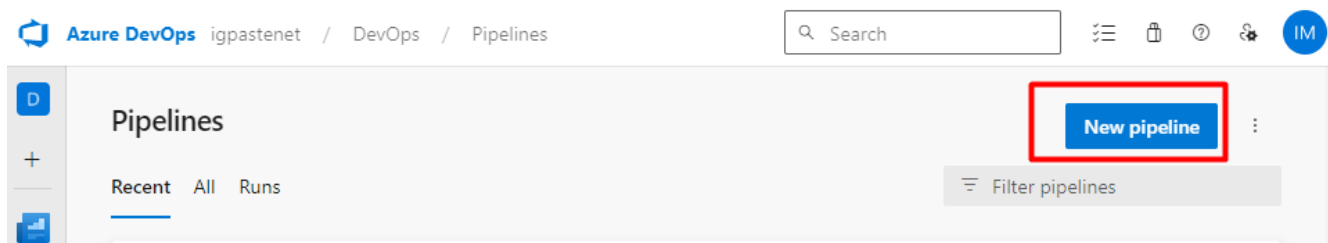
- Subir archivo código pipeline a GitHub para subir imagen Dockerfile hacia ACR.



- Crear Service Connections Docker Registry de tipo Azure Container Registry.



- Creación y Ejecución de Azure Pipeline con código YAML desde repositorio GitHub para subir imagen Dockerfile a ACR.



DevOps

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Library

Test Plans

Artifacts

Connect

Select

Configure

Review

New pipeline

Where is your code?

Azure Repos Git

YAML

Free private Git repositories, pull requests, and code search

Bitbucket Cloud

YAML

Hosted by Atlassian

GitHub

YAML

Home to the world's largest community of developers

GitHub Enterprise Server

YAML

The self-hosted version of GitHub Enterprise

DevOps

Overview

Boards

Repos

Pipelines

Pipelines

Environments

Library

Test Plans

Artifacts

✓ Connect

✓ Select

Configure

Review

New pipeline

Configure your pipeline

Docker

Build a Docker image

Docker

Build and push an image to Azure Container Registry

Deploy to Azure Kubernetes Service

Build and push image to Azure Container Registry; Deploy to Azure Kubernetes Service

Starter pipeline

Start with a minimal pipeline that you can customize to build and deploy your code.

Existing Azure Pipelines YAML file

Select an Azure Pipelines YAML file in any branch of the repository.

Show more

Select an existing YAML file

Select an Azure Pipelines YAML file in any branch of the repository.

Branch

main

Path

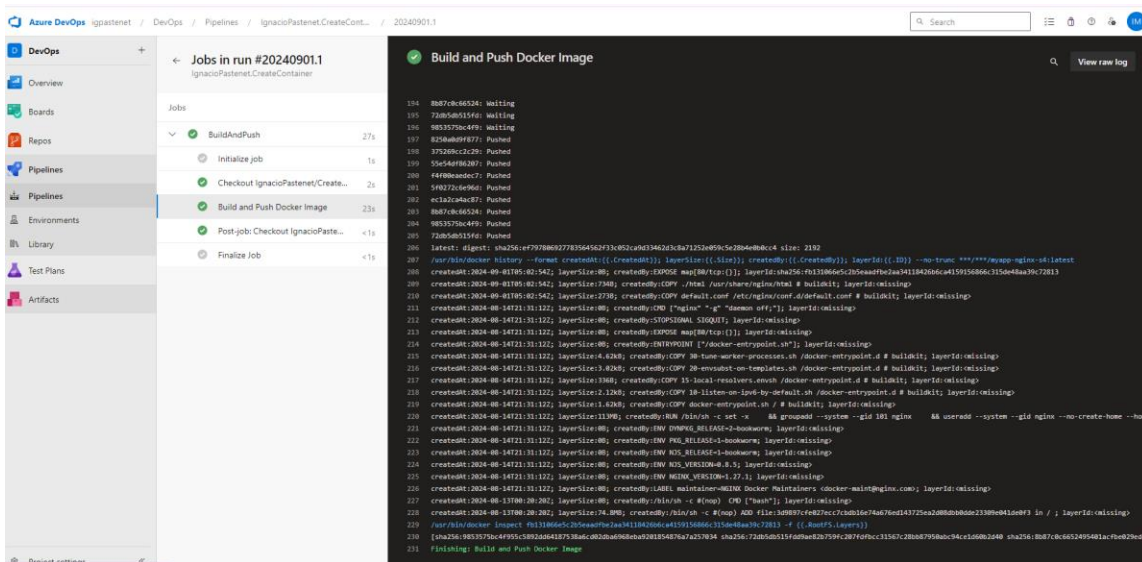
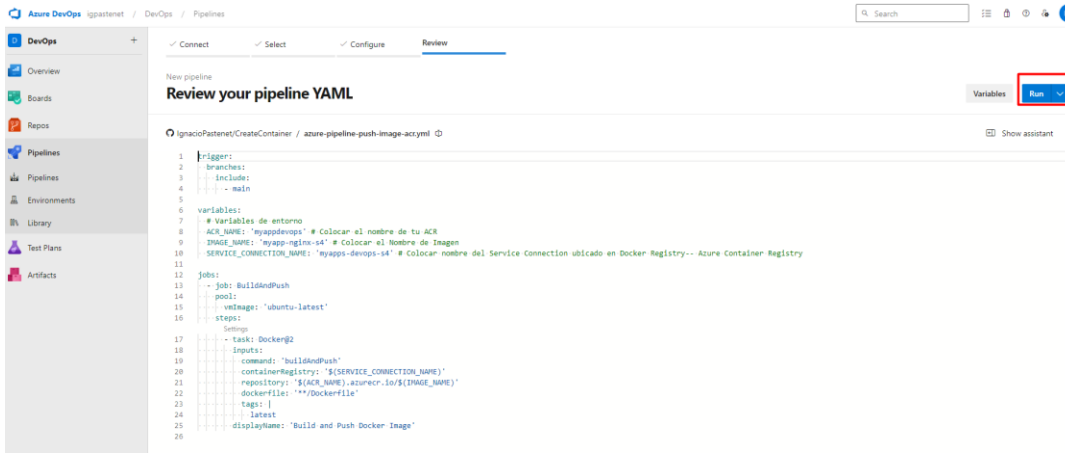
/azure-pipeline-push-image-acr.yml

Select a file from the dropdown or type in the path to your file

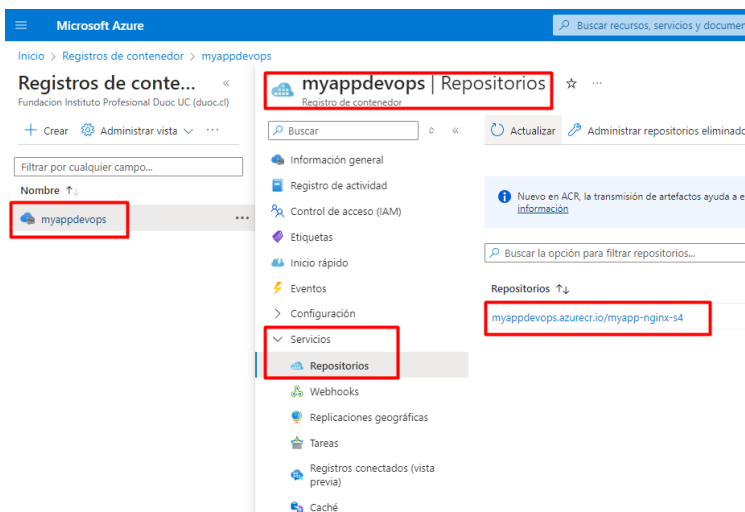
igpastenet/CreateContainer

Cancel

Continue

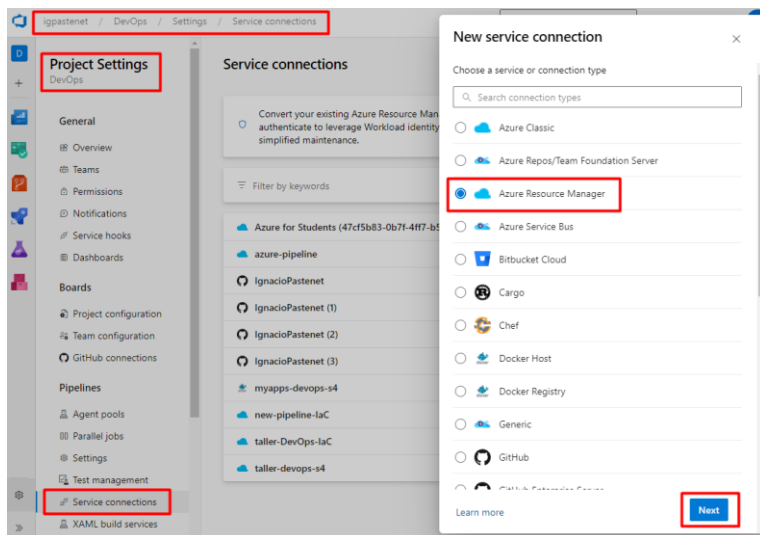


- Evidencias de la ejecución del pipeline en Azure Portal sección Registros de contenedor y validar que este la imagen subida.



Evidencias Paso 3: Preparar ambiente y ejecución del Pipeline para despliegue de imagen Docker en máquina virtual.

- Crear Service Connections con Azure Resource Manager y el grupo de recurso.



New Azure service connection

Azure Resource Manager

Authentication method

☒ Workload Identity federation (automatic) Recommended

☐ Workload Identity federation (manual)

☐ Service principal (automatic)

☐ Service principal (manual)

☐ Managed identity

☐ Publish Profile

[Need help choosing a connection type?](#) [Back](#) [Next](#)

New Azure service connection

Azure Resource Manager using Workload Identity federation with OpenID Connect (automatic)

Scope level

☒ Subscription

☐ Management Group

☐ Machine Learning Workspace

Subscription

Azure for Students (47cf5b83-0b7f-4ff7-b5fb-6e338131ee72)

Resource group

taller-DevOps

Details

Service connection name

taller-devops-s4

Description (optional)

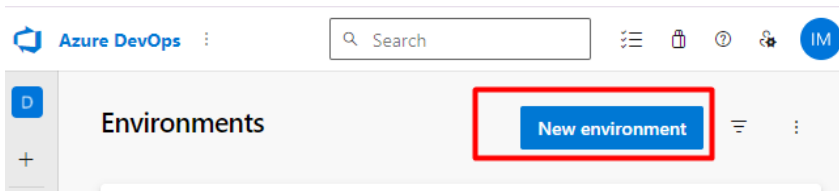
Servicio de conexión para el despliegue del Azure Pipeline e instalar docker y ejecutar la imagen desde ACR en la MV.

Security

☒ Grant access permission to all pipelines

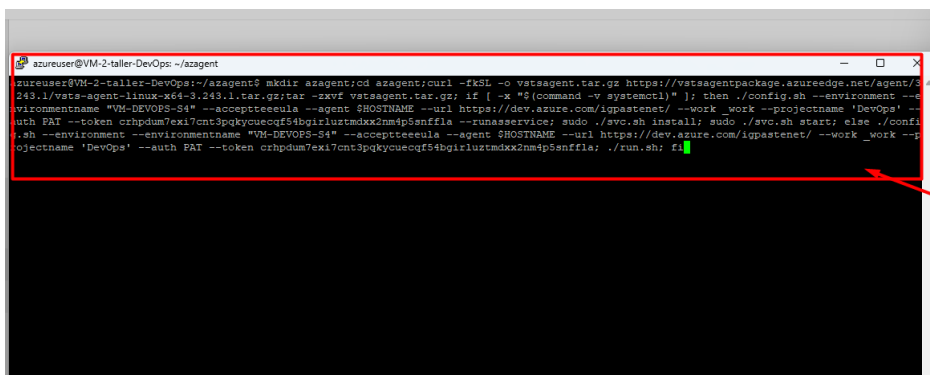
[Learn more](#) [Troubleshoot](#) [Back](#) [Save](#)

- Crear un entorno en Azure pipeline para registrar la máquina virtual y dar permisos al repositorio pipeline.

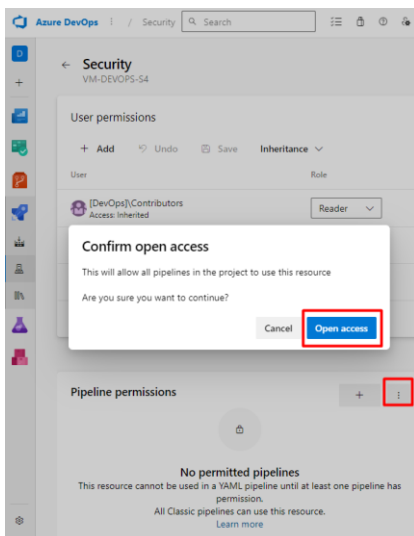
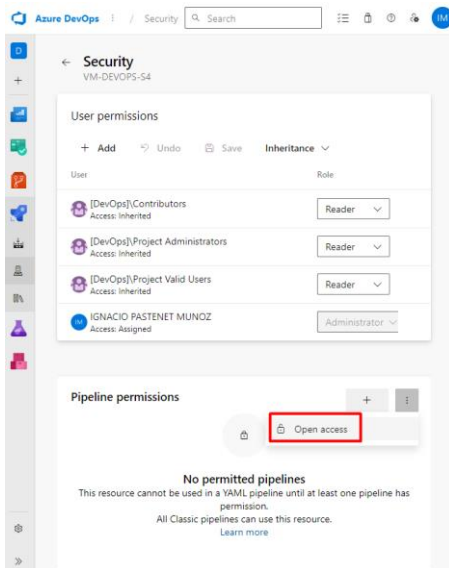
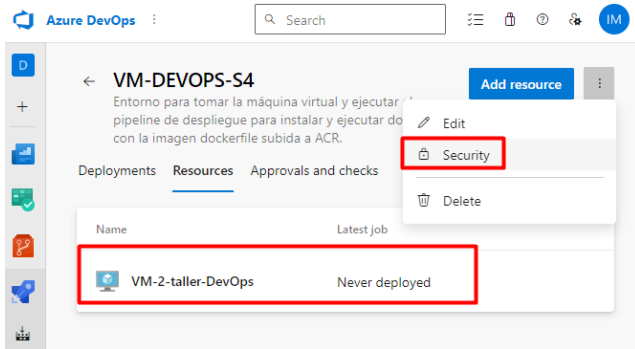
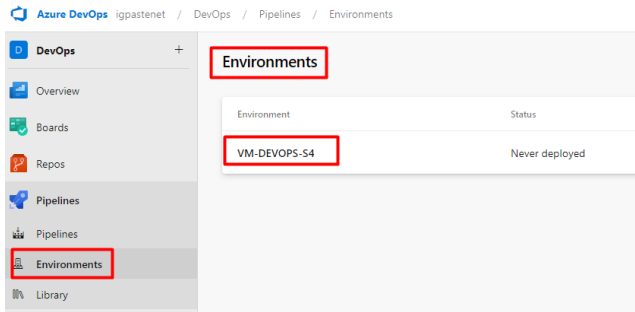


The 'New environment' form is shown. The 'Name' field contains 'VM-DEVOPS-S4'. The 'Description' field contains 'Entorno para tomar la máquina virtual y ejecutar el pipeline de despliegue para instalar y ejecutar docker con la imagen dockerfile subida a ACR.'. The 'Resource' section has 'Virtual machines' selected. The 'Next' button at the bottom right is highlighted with a red box.

The 'Virtual machine resource' section of the form is shown. The 'Provider' is set to 'Generic provider' and the 'Operating system' is set to 'Linux'. The 'Registration script' section shows a command to run 'azagent' and a warning about the Personal Access Token.



The 'Virtual machine resource' section of the form is shown. The 'Provider' is set to 'Generic provider' and the 'Operating system' is set to 'Linux'. The 'Registration script' section shows a command to run 'azagent' and a warning about the Personal Access Token.



- Creación de código pipeline en yaml para instalar Docker, azure cli y hacer pull y run de imagen en ACR.

```

trigger:
  branches:
    include:
      - main

variables:
  VM_IP_PUBLICA: '40.123.37.134' # Se debe colocar la IP pública de tu máquina virtual
  IMAGE_NAME: 'myapp-nginx-s4' # Se debe colocar el nombre de la imagen que será descargada desde ACR
  ACR_NAME: 'myappdevops' # Se debe colocar el nombre de tu Azure Container Registry (ACR)
  ACR_USERNAME: 'myappdevops' # Se debe colocar el nombre de usuario del ACR
  ACR_PASSWORD: 'f5s1HJB3q1q1NFQuuFDHRTqEhDfDKuKwZw1VtACR0m8trs' # Se debe colocar la contraseña del ACR

stages:
- stage: Deploy
  jobs:
  - deployment: DeployToVM
    displayName: 'Deploy to VM'
    environment:
      name: 'VM-DEVOPS-S4' # Nombre del entorno configurado en Azure DevOps
      resourceType: VirtualMachine # Especifica que se conecta a una máquina virtual
    strategy:
      runOnce:
        deploy:
          steps:
            - script: |
                # Verificar e instalar Azure CLI en la máquina virtual
                if ! command -v az && /dev/null; then
                  echo "Azure CLI no está instalado. Instalando Azure CLI..."
                  sudo apt-get update && \
                  sudo apt-get install -y curl && \
                  curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash && \
                  az --version
                else
                  echo "Azure CLI ya está instalado."
                fi
                displayName: 'Verificar e instalar Azure CLI'
                failOnStderr: true
            - script: |

```



azure-pipelines-myap
pdocker-vm-acr.yml

Registros de conte...
Fundación Instituto Profesional Duoc UC (duoc.cl)

myappdevops | Claves de acceso

Nombre de usuario: myappdevops

Contraseña: f5s1HJB3q1q1NFQuuFDHRTqEhDfDKuKwZw1VtACR0m8trs

- Subir archivo código pipeline a GitHub para posterior ejecución en Azure Pipeline.

IgnacioPastenet / CreateContainer

Code Issues Pull requests Actions Projects Security Insights Settings

CreateContainer Private

Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file Add file Code

IgnacioPastenet Add files via upload 1250893 · 22 minutes ago 39 Commits

File	Commit Message	Time
my-repo/docker	Delete my-repo/docker/html/index	51 minutes ago
README.md	Initial commit	2 days ago
azure-pipeline-push-image-acr.yml	Add files via upload	23 minutes ago
azure-pipelines-myappdocker-vm-acr.yml	Add files via upload	22 minutes ago

README

CreateContainer

Repositorio para crear contenedor

Repositorio para crear contenedor

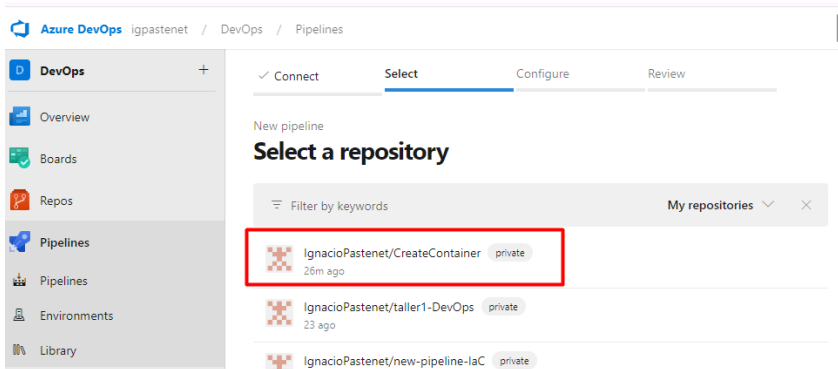
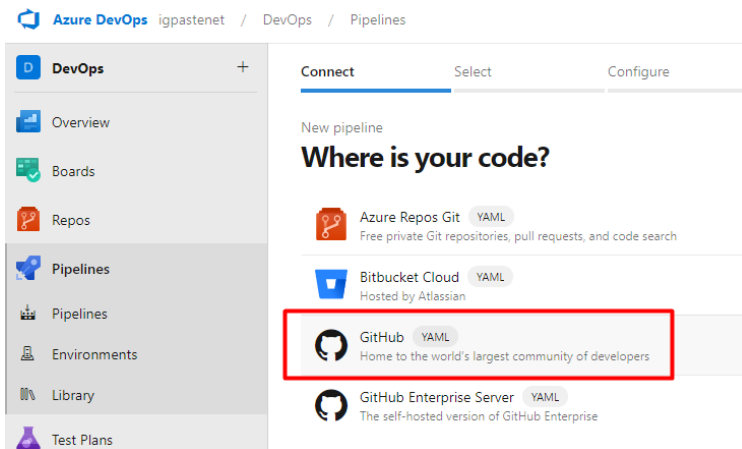
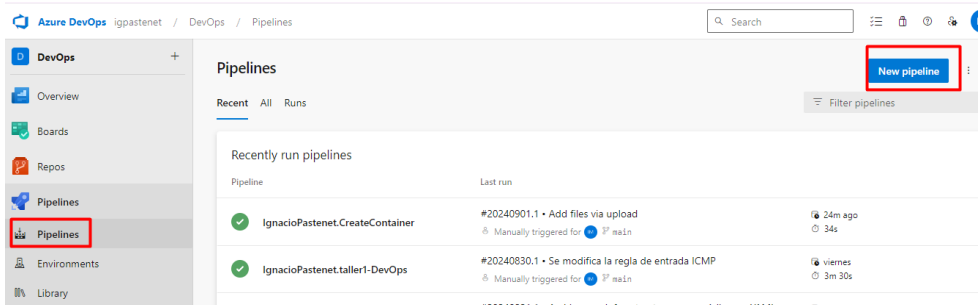
No releases published
[Create a new release](#)

No packages published
[Publish your first package](#)

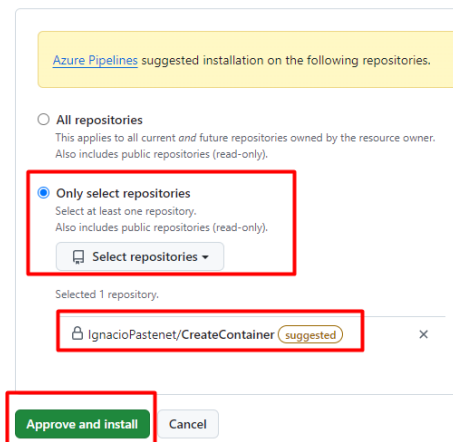
Languages

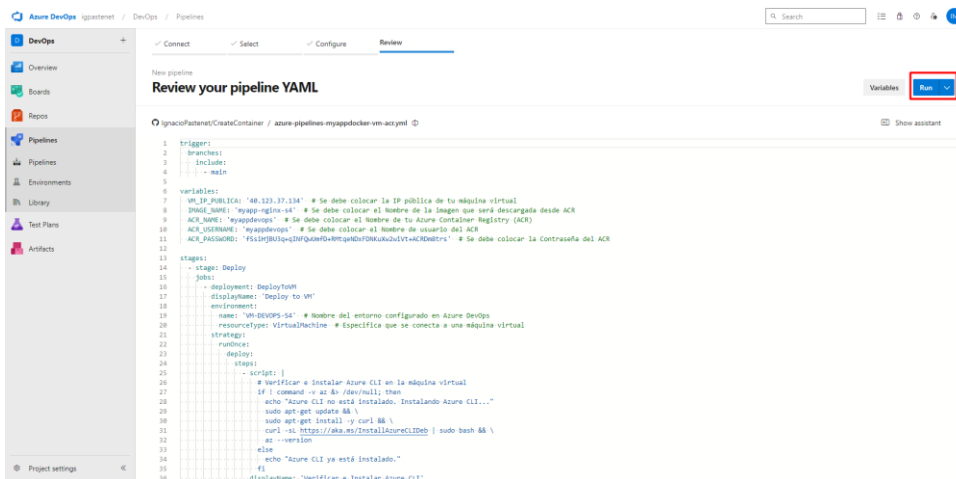
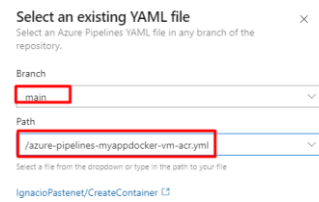
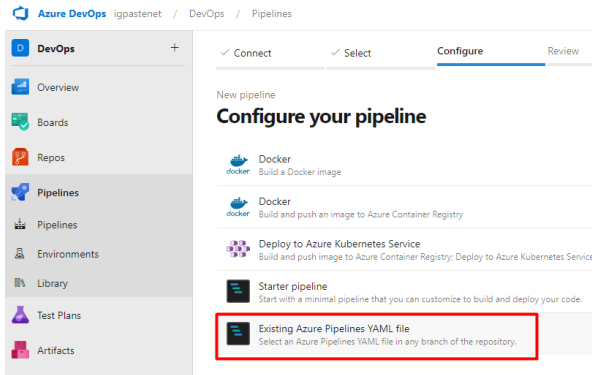
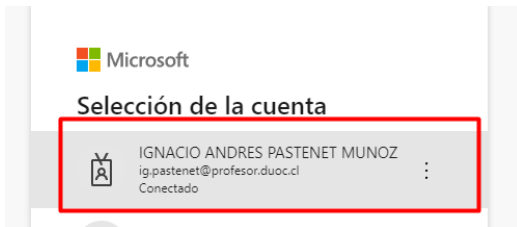
HTML 65.8% Dockerfile 34.2%

- Creación y Ejecución de Azure Pipeline con código YAML desde repositorio GitHub para instalar y ejecutar Docker con la imagen nginx y página de bienvenida.

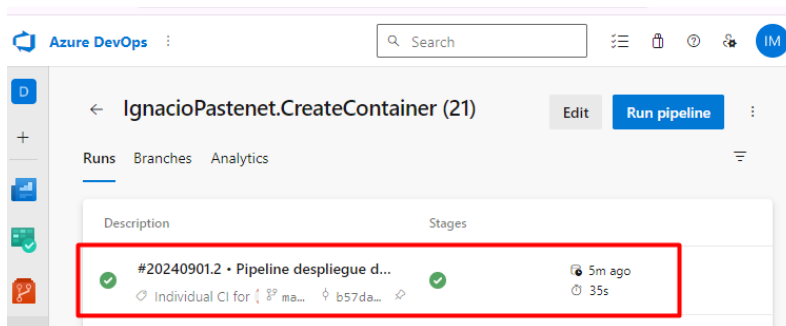
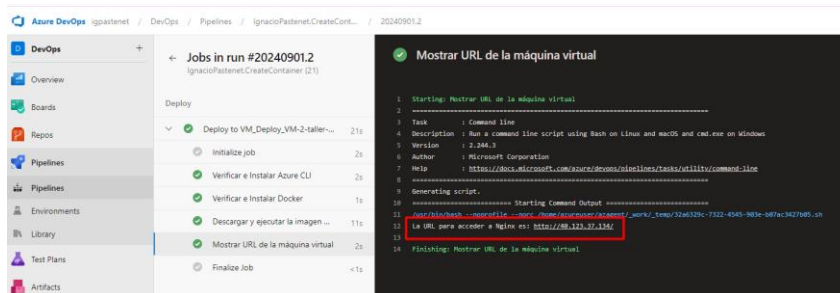
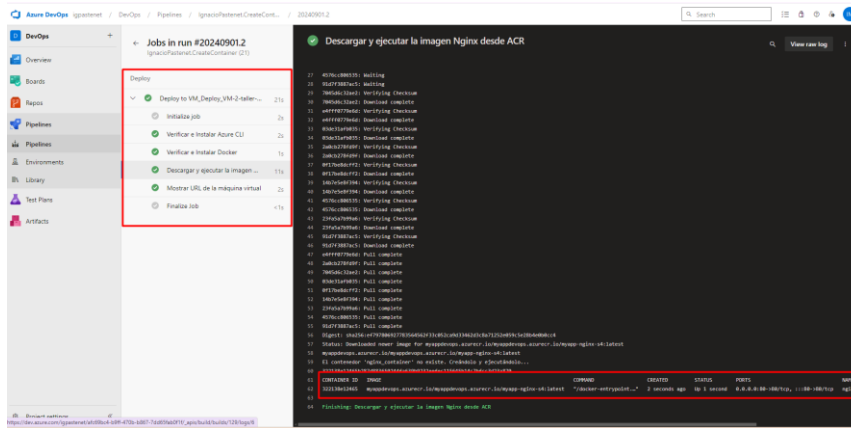


Repository access

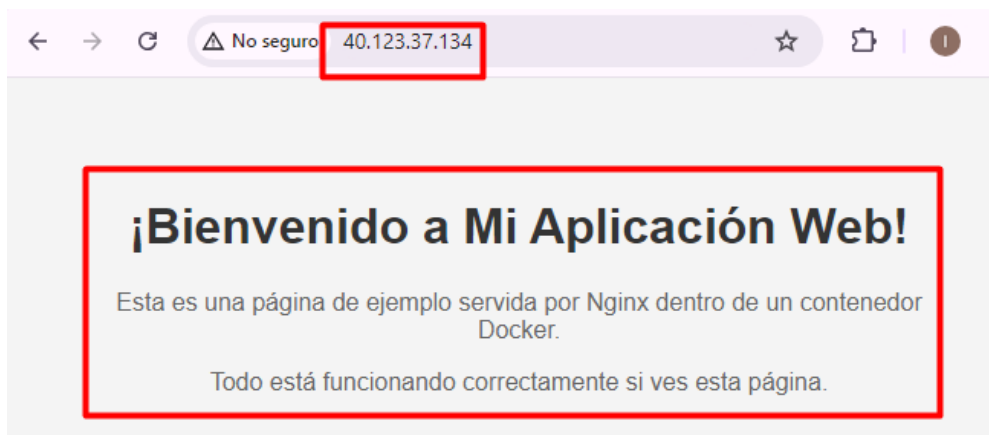




- Revisión de la ejecución del job del pipeline.



- Ejecutar IP publica de la máquina virtual en navegador web.



- Evidencias de la ejecución del pipeline desde máquina virtual como ejecución de Docker ps.

```
azureuser@VM-2-taller-DevOps: ~/azagent
azureuser@VM-2-taller-DevOps:~/azagent$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                    NAMES
322138e12465   myappdevops.azurecr.io/myappdevops.azurecr.io/myapp-nginx-s4:latest "/docker-entrypoint..." 6 minutes ago  Up 5 minutes  0.0.0.0:80->80/tcp, :::80->80/tcp nginx_container
```

Actividad Finalizada.

Con este paso a paso se evidencia la correcta ejecución de la creación de una imagen Docker y subirla al repositorio de imágenes en Azure Container Registry (ACR) de manera automatizada con Azure DevOps Pipelines.

También se evidencia paso a paso el despliegue de la instalación de Docker y la descarga de la imagen en ACR para luego ejecutar el Docker con la imagen, donde posteriormente se evidencia la ejecución del contenedor Docker y la evidencia de la página web del contenedor en el navegador web con la ip publica de la máquina virtual y su puerto asociado. Todo esto se creó de forma automatizada mediante código YAML a través de Azure DevOps Pipelines.

Con esto el alumno logra diferenciar la forma de automatizar el despliegue de contenedores e imágenes en Azure DevOps.



Reservados todos los derechos Fundación Instituto Profesional Duoc UC. No se permite copiar, reproducir, reeditar, descargar, publicar, emitir, difundir, de forma total o parcial la presente obra, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio (electrónico, mecánico, fotocopia, grabación u otros) sin autorización previa y por escrito de Fundación Instituto Profesional Duoc UC. La infracción de dichos derechos puede constituir un delito contra la propiedad intelectual.