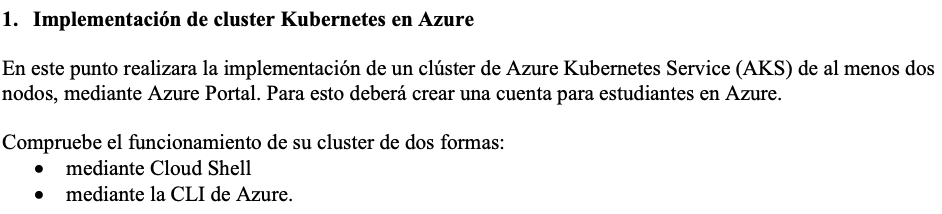
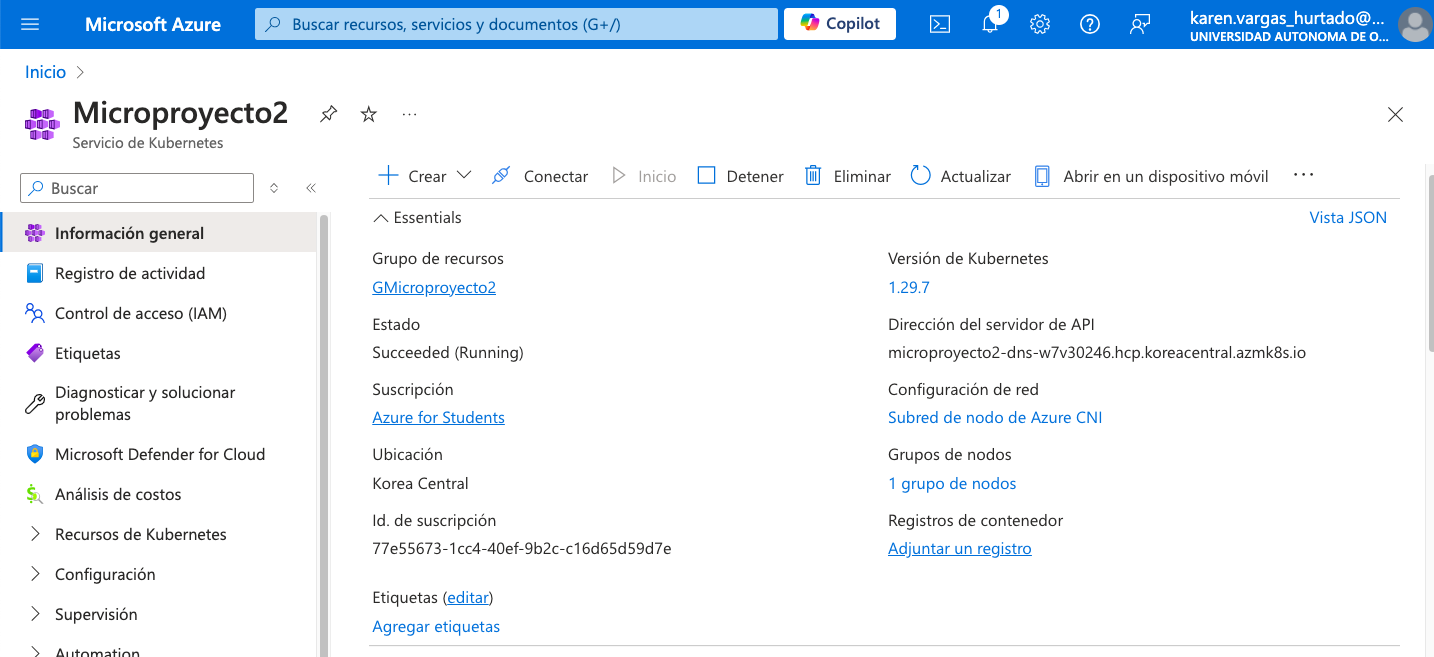
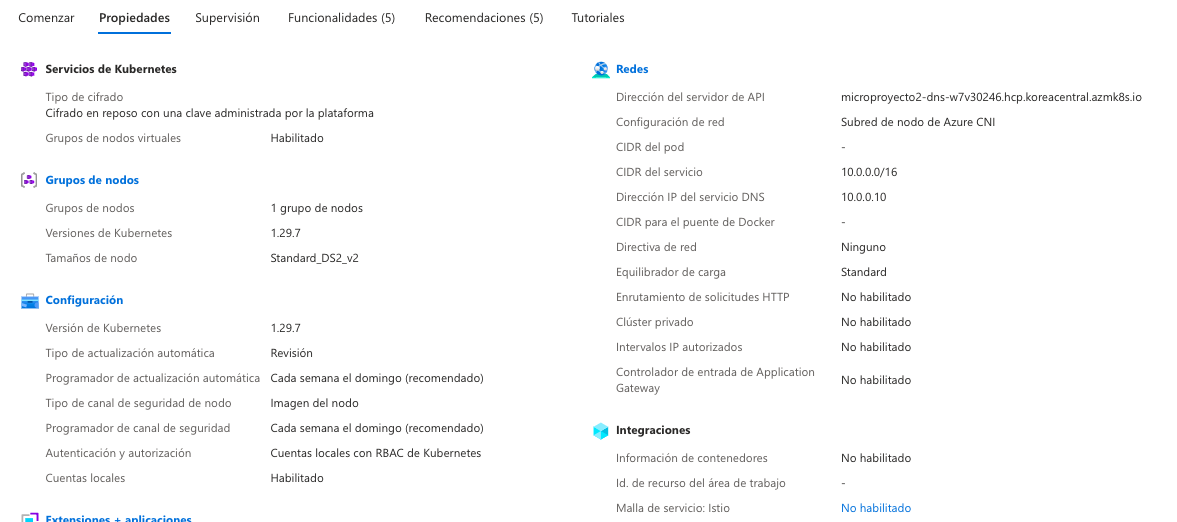
Microproyecto 2

Por: Prof. Oscar Mondragón

Presenta: Johan Camilo , Diego Moreno, Karen Vargas







karen [ ~ ]$ az account set --subscription 77e55673-1cc4-40ef-9b2c-c16d65d59d7e

karen [ ~ ]$ az aks get-credentials --resource-group GMicroproyecto2 --name Microproyecto2 --overwrite-existing

Merged "Microproyecto2" as current context in /home/karen/.kube/config

karen [ ~ ]$ export RANDOM\_ID="$(openssl rand -hex 3)"

export MY\_RESOURCE\_GROUP\_NAME="myAKSResourceGroup$RANDOM\_ID"

export REGION="westeurope"

export MY\_AKS\_CLUSTER\_NAME="myAKSCluster$RANDOM\_ID"

export MY\_DNS\_LABEL="mydnslabel$RANDOM\_ID"

karen [ ~ ]$ az group create --name Microproyecto2 --location karen

karen [ ~ ]$ az group create --name Microproyecto2 --location koreacentral

{

"id": "/subscriptions/77e55673-1cc4-40ef-9b2c-c16d65d59d7e/resourceGroups/Microproyecto2",

"location": "koreacentral",

"managedBy": null,

"name": "Microproyecto2",

"properties": {

"provisioningState": "Succeeded"

},

"tags": null,

"type": "Microsoft.Resources/resourceGroups"

}

karen [ ~ ]$ az aks get-credentials --resource-group GMicroproyecto2 --name Microproyecto2 --overwrite-existing

Merged "Microproyecto2" as current context in /home/karen/.kube/config

karen [ ~ ]$ kubectl get nodes

NAME STATUS ROLES AGE VERSION

aks-agentpool-34541555-vmss000002 Ready <none> 151m v1.29.7

aks-agentpool-34541555-vmss000003 Ready <none> 151m v1.29.7

virtual-node-aci-linux Ready agent 29h v1.25.0-vk-azure-aci-1.6.1

karen [ ~ ]$ mkdirs aks-store-quickstart.yaml

bash: mkdirs: command not found

karen [ ~ ]$ mkdir aks-store-quickstart.yaml

karen [ ~ ]$ cd

karen [ ~ ]$ ls

aks-store-quickstart.yaml aks-store-quickstart.yaml.

karen [ ~ ]$ apiVersion: apps/v1

karen [ ~ ]$ vim aks-store-quickstart.yaml

karen [ ~ ]$ vim aks-store-quickstart.yaml

karen [ ~ ]$ vim aks-store-quickstart.yaml

karen [ ~ ]$ vim aks-store-quickstart.yaml

karen [ ~ ]$ kubectl apply -f aks-store-quickstart.yaml

error: error reading [aks-store-quickstart.yaml]: recognized file extensions are [.json .yaml .yml]

karen [ ~ ]$ kubectl apply -f aks-store-quickstart.yaml

error: error reading [aks-store-quickstart.yaml]: recognized file extensions are [.json .yaml .yml]

karen [ ~ ]$ runtime="5 minutes"

endtime=$(date -ud "$runtime" +%s)

while [[ $(date -u +%s) -le $endtime ]]

do

STATUS=$(kubectl get pods -l app=store-front -o 'jsonpath={..status.conditions[?(@.type=="Ready")].status}')

echo $STATUS

if [ "$STATUS" == 'True' ]

then

export IP\_ADDRESS=$(kubectl get service store-front --output 'jsonpath={..status.loadBalancer.ingress[0].ip}')

echo "Service IP Address: $IP\_ADDRESS"

break

else

sleep 10

donei

^C

karen [ ~ ]$

karen [ ~ ]$ while [[ $(date -u +%s) -le $endtime ]]; do STATUS=$(kubectl get pods -l app=store-front -o 'jsonpath={..status.conditions[?(@.type=="Ready")].status}'); echo $STATUS; if [ "$STATUS" == 'True' ]; then export IP\_ADDRESS=$(kubectl get service store-front --output 'jsonpath={..status.loadBalancer.ingress[0].ip}'); echo "Service IP Address: $IP\_ADDRESS"; break; else sleep 10; fi; done

^C

karen [ ~ ]$ curl 10.0.0.10

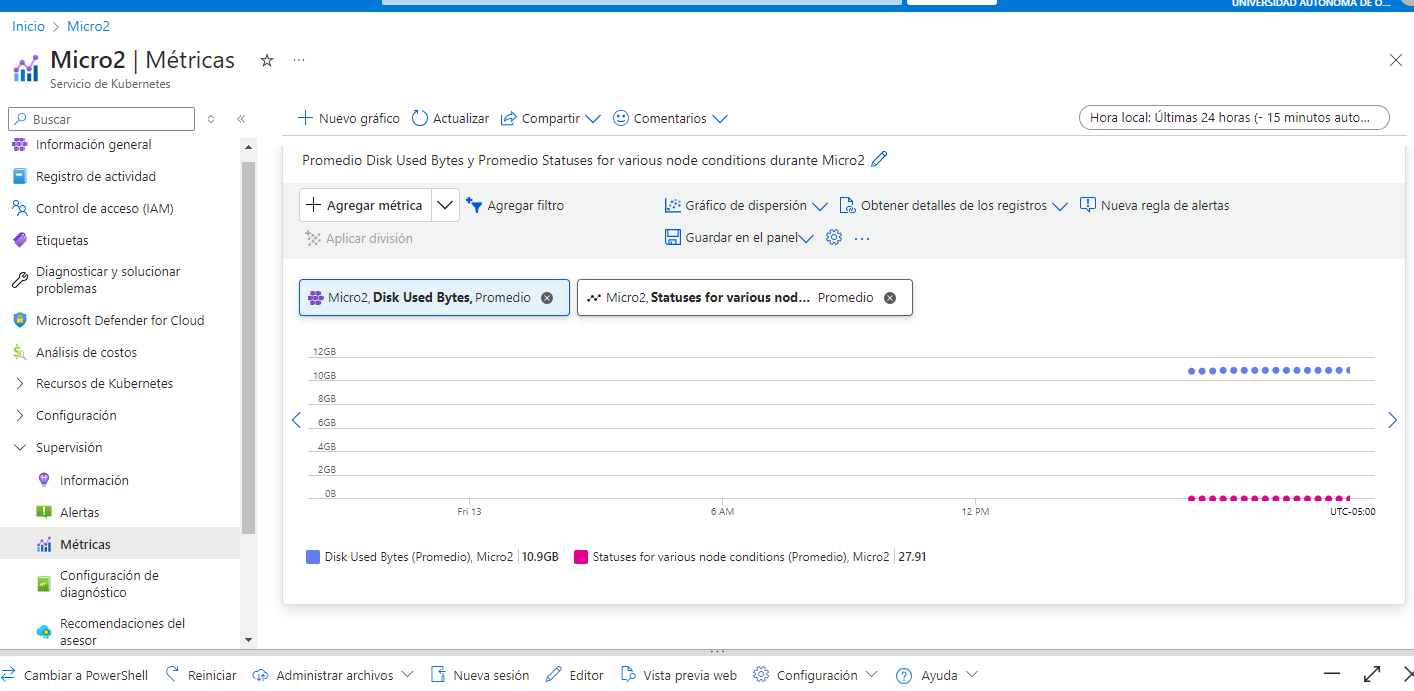
curl: (28) Failed to connect to 10.0.0.10 port 80 after 130649 ms: Couldn't connect to server

karen [ ~ ]$ curl 10.0.0.10:800

curl: (28) Failed to connect to 10.0.0.10 port 800 after 130442 ms: Couldn't connect to server

karen [ ~ ]$ curl 10.0.0.10/16

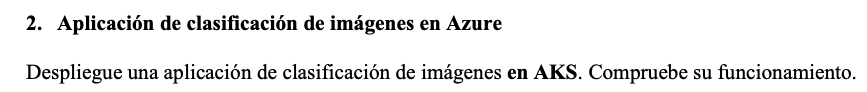
curl: (28) Failed to connect to 10.0.0.10 port 80 after 130673 ms: Couldn't connect to server



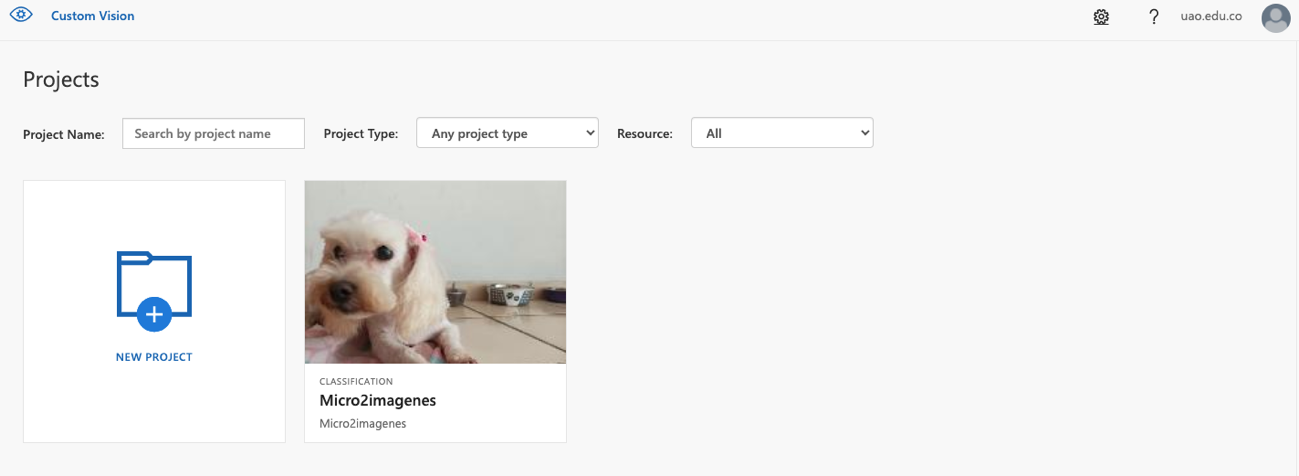
CLI

Texto

Descripción generada automáticamente



Intento de implementación aplicación de clasificación en Custom Vision.



Codigo:

import tensorflow as tf

import os

import numpy as np

# Descarga y extracción del set de datos

print("Descargando ZIP de datos")

url = 'https://storage.googleapis.com/mledu-datasets/cats\_and\_dogs\_filtered.zip'

carpeta\_zip = tf.keras.utils.get\_file('cats\_and\_dogs\_filterted.zip', origin=url, extract=True)

# Variables para rutas en disco

carpeta\_base = os.path.join(os.path.dirname(carpeta\_zip), 'cats\_and\_dogs\_filtered')

carpeta\_entrenamiento = os.path.join(carpeta\_base, 'train')

carpeta\_validacion = os.path.join(carpeta\_base, 'validation')

TAMANO\_LOTE = 100

TAMANO\_IMG = (150, 150)

# Crear datasets desde directorios

print("Creando datasets de entrenamiento y validación")

data\_gen\_entrenamiento = tf.keras.preprocessing.image\_dataset\_from\_directory(

    carpeta\_entrenamiento,

    batch\_size=TAMANO\_LOTE,

    image\_size=TAMANO\_IMG,

    shuffle=True

)

data\_gen\_validacion = tf.keras.preprocessing.image\_dataset\_from\_directory(

    carpeta\_validacion,

    batch\_size=TAMANO\_LOTE,

    image\_size=TAMANO\_IMG,

    shuffle=True

)

# Aumento de datos utilizando capas de preprocesamiento

print("Aplicando aumento de datos")

data\_augmentation = tf.keras.Sequential([

    tf.keras.layers.Rescaling(1./255),

    tf.keras.layers.RandomFlip('horizontal'),

    tf.keras.layers.RandomRotation(0.1),

    tf.keras.layers.RandomZoom(0.1),

])

AUTOTUNE = tf.data.AUTOTUNE

data\_gen\_entrenamiento = data\_gen\_entrenamiento.map(

    lambda x, y: (data\_augmentation(x, training=True), y),

    num\_parallel\_calls=AUTOTUNE

).repeat().prefetch(buffer\_size=AUTOTUNE)

data\_gen\_validacion = data\_gen\_validacion.map(

    lambda x, y: (x / 255.0, y),

    num\_parallel\_calls=AUTOTUNE

).repeat().prefetch(buffer\_size=AUTOTUNE)

# Modelo

modelo = tf.keras.models.Sequential([

    tf.keras.layers.InputLayer(input\_shape=(TAMANO\_IMG[0], TAMANO\_IMG[1], 3)),

    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),

    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),

    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),

    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),

    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Dropout(0.5),

    tf.keras.layers.Flatten(),

    tf.keras.layers.Dense(512, activation='relu'),

    tf.keras.layers.Dense(1, activation='sigmoid')  # Cambiamos a una única unidad con activación 'sigmoid' para clasificación binaria

])

# Compilación

modelo.compile(optimizer='adam',

               loss=tf.keras.losses.BinaryCrossentropy(from\_logits=False),  # Usamos 'binary\_crossentropy' para clasificación binaria

               metrics=['accuracy'])

# Calcular steps\_per\_epoch y validation\_steps

steps\_per\_epoch = np.ceil(2000 / TAMANO\_LOTE).astype(int)

validation\_steps = np.ceil(1000 / TAMANO\_LOTE).astype(int)

# Entrenar la red

print("Entrenando modelo...")

epocas = 60

history = modelo.fit(

    data\_gen\_entrenamiento,

    steps\_per\_epoch=steps\_per\_epoch,

    epochs=epocas,

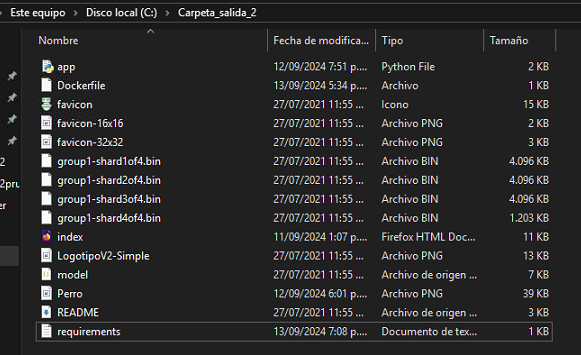
    validation\_data=data\_gen\_validacion,

    validation\_steps=validation\_steps

)

print("¡Modelo entrenado!")

Carpeta compartida windows – vagrant





App.py

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

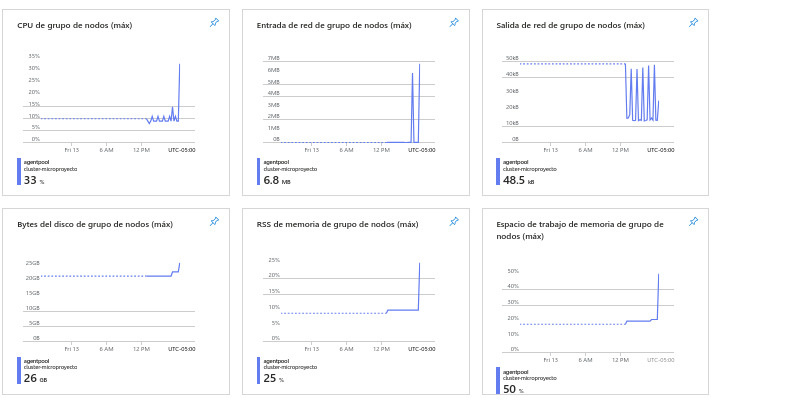
Pantalla de un celular con la imagen de un gato

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media





Links de consulta:   
<https://learn.microsoft.com/en-us/cli/azure/get-started-with-azure-cli#code-try-0>

<https://learn.microsoft.com/es-es/azure/aks/learn/quick-kubernetes-deploy-cli?source=recommendations>

<https://www.youtube.com/watch?v=G5O2Ft-SZh8>

<https://learn.microsoft.com/es-es/azure/ai-services/custom-vision-service/getting-started-build-a-classifier>