Azure ML

ESPECIALIZACIÓN EN ANALÍTICA – UDEA

DATA STREAMING Y SERVICIOS EN LA NUBE

REYSON DÍAZ

¿Qué es?

Servicio de Azure para crear, entrenar e implementar modelos de Machine Learning y fomentar la colaboración en equipo.

Permite llevar a cabo el ciclo MLOps (DevOps para analítica)

- Proyectos
- Entrenamiento
- Registro
- Despliegue

Workspace Azure ML

Grupo de recursos (resource group): Agrupador de los componentes que están involucrados en un espacio de trabanjo de Azure ML

Machine learning: Espacio de trabajo donde se llevan a cabo todas las actividades del flujo de ML

Application Insights: Ayuda a la generación de métricas acerca del desempeño del componente de azure ml

Key vault: Bodega para almacenamiento de credenciales

Storage account: Repositorio para almacenar los datasets y demás recursos que usamos en los modelos de ML

Container registry: Repositorio para almacenar las imágenes de los modelos

Configuración ambiente

Usar >= 3.5 Python < 3.9

conda create -n azure-ml python=3.8

conda activate azure-ml

Pip install azureml-core

Azure-ml-core

Librería Python para interactuar con los servicios de Azure

Se estructura en:

- Paquetes
- Módulos
- Clases

https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core?view=azure-ml-py

Azure-ml-core

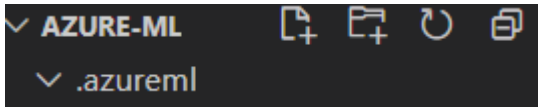
- Paquetes
  - Compute
  - Image
  - Webservice

Creación estructura del proyecto

mkdir azure-ml - Directorio raiz del proyecto

cd azure-ml

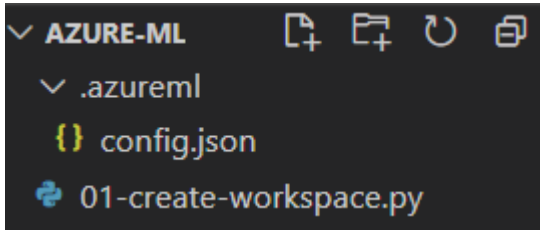
mkdir .azureml – Directorio oculto para almacenar la configuración del Proyecto



Configuración espacio trabajo (Workspace)

Un workspace es el recurso de alto de nivel para Azure Machine Learning, sirve para:

- Gestionar recursos tales como la computación
- Almacenar notebooks, ambientes, datasets, pipelines, modelos y endpoints
- Colaborar con otros miembros



Verificación Workspace

You have logged into Microsoft Azure ML Service!

You can close this window, or we will redirect you to the Azure ML documentation in 10 seconds.

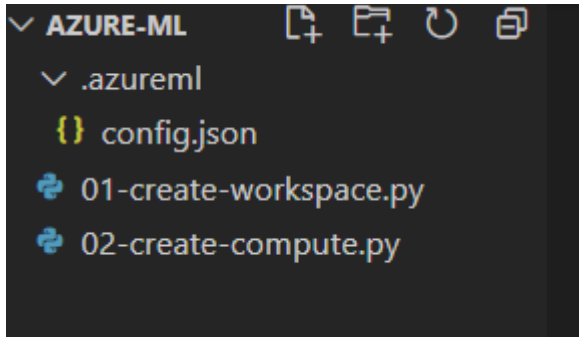
```
(azure-ml) PS C:\Users\Reyson\Documents\Personal\DOCENCIA\Machine Learning UdeA\azure-ml> python .\01-create-workspace.py
Deploying KeyVault with name azuremlkeyvaultb8e5fe99f.
Deploying AppInsights with name azuremlinsightsfdb555e8e.
Deployed AppInsights with name azuremlinsightsfdb555e8e. Took 5.78 seconds.
Deploying StorageAccount with name azuremlstoragef6b4fad3cc.
Deployed KeyVault with name azuremlkeyvaultb8e5fe99f. Took 21.76 seconds.
Deployed StorageAccount with name azuremlstoragef6b4fad3cc. Took 24.67 seconds.
Deploying Workspace with name azure-ml.
Deployed Workspace with name azure-ml. Took 68.29 seconds.
(azure-ml) PS C:\Users\Reyson\Documents\Personal\DOCENCIA\Machine Learning UdeA\azure-ml>
```

azure-ml	Machine learning	East US 2
azuremlinsightsfdb555e8e	Application Insights	East US 2
azuremlkeyvaultb8e5fe99f	Key vault	East US 2
azuremlstoragef6b4fad3cc	Storage account	East US 2

Creación computo

Permite la ejecución de tareas de entrenamiento de los modelos

> **Note**
>
> When the cluster is created, it will have 0 nodes provisioned. The cluster *does not* incur costs until you submit a job. This cluster will scale down when it has been idle for 2,400 seconds (40 minutes).



Verificación computo

```
(azure-ml) PS C:\Users\Reyson\Documents\Personal\DOCENCIA\Machine Learning UdeA\azure-ml> python .\02-create-compute.py
Creating.........
SucceededProvisioning operation finished, operation "Succeeded"
Succeeded
AmlCompute wait for completion finished

Minimum number of nodes requested have been provisioned
```

Compute

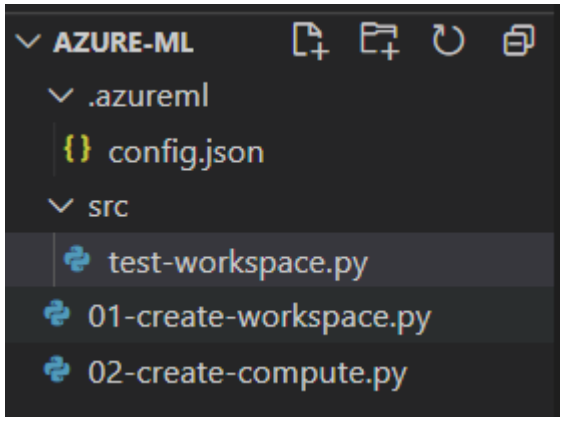
Compute instances | **Compute clusters** | Inference clusters | Attached computes

+ New | Delete | Edit columns | Refresh | Reset to default view | View quota

Add filter | Search computes

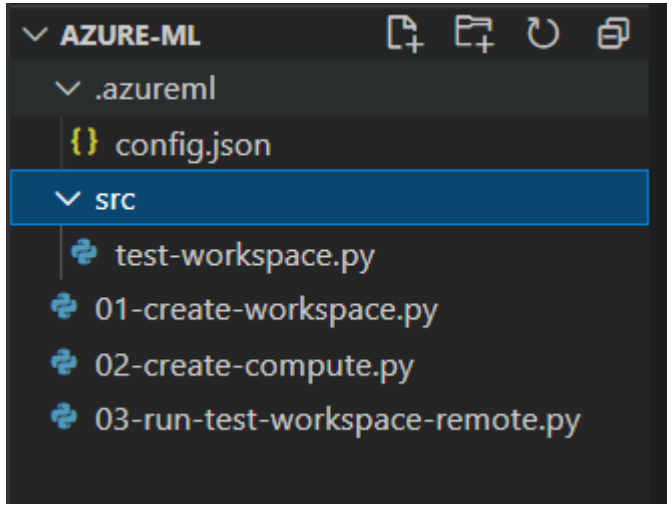
Name	State	Size	Created on ↓	Active runs	Idle nodes	Busy nodes	Unprovisioned nodes
cpu-cluster	Succeeded (0 nodes)	STANDARD_D2_V2	18 de abr. de 2021 16:05	0	0	0	4

Prueba workspace local




```
(azure-ml) PS C:\Users\Reyson\Documents\Personal\DOCENCIA\Machine Learning UdeA\azure-ml> python .\src\test-workspace.py
Testing Workspace...
```

Prueba Workspace Remota



03-run-test-workspace-remote.py

```python
# tutorial/03-run-test-workspace-remote.py
from azureml.core import Workspace, Experiment, Environment, ScriptRunConfig

ws = Workspace.from_config()
experiment = Experiment(workspace=ws, name='day1-experiment-testing-workspace')

config = ScriptRunConfig(source_directory='./src', script='test-workspace.py', compute_target='cpu-cluster')

run = experiment.submit(config)
aml_url = run.get_portal_url()
print(aml_url)
```

Prueba Workspace entendimiento

ws = Workspace.from_config(): Conecta con el espacio de trabajo de Azure Machine Learning

experiment = Experiment(...): Permite organizar multiples ejecuciones de un modelo bajo el mismo nombre, permitiendo por ejemplo comprara métricas de desempeño entre diferentes ejecuciones

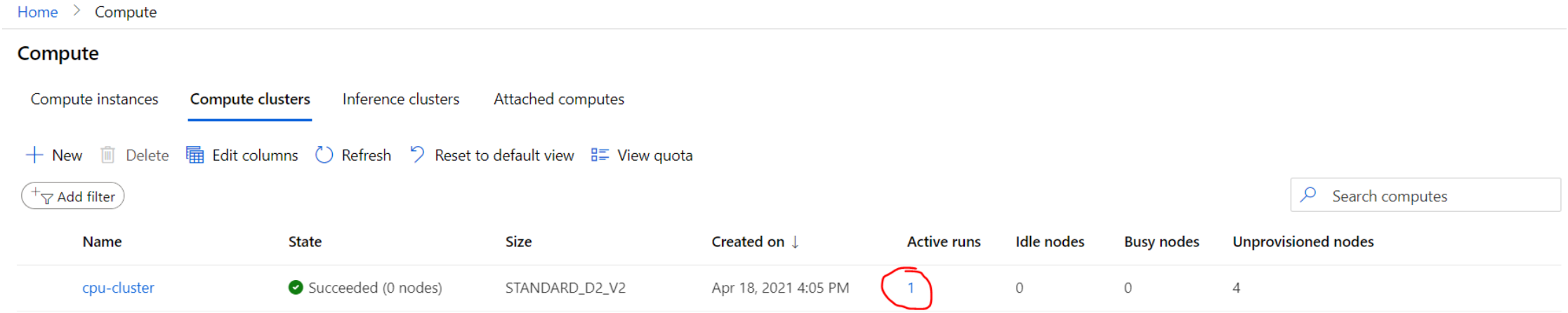
config = ScriptRunConfig(...): Encapsula el código y lo pasa al espacio de trabajo para su ejecución. Permite configurar la forma en como se va a ejecutar el código en el espacio de trabajo

run = experiment.submit(config): Envía el código para la ejecución. Cada run representa una única ejecución

aml_url = run.get_portal_url(): Cada objeto de ejecución tiene atributos para controlar la ejecución del código, permitiendo el monitoreo del progreso de la ejecución del código

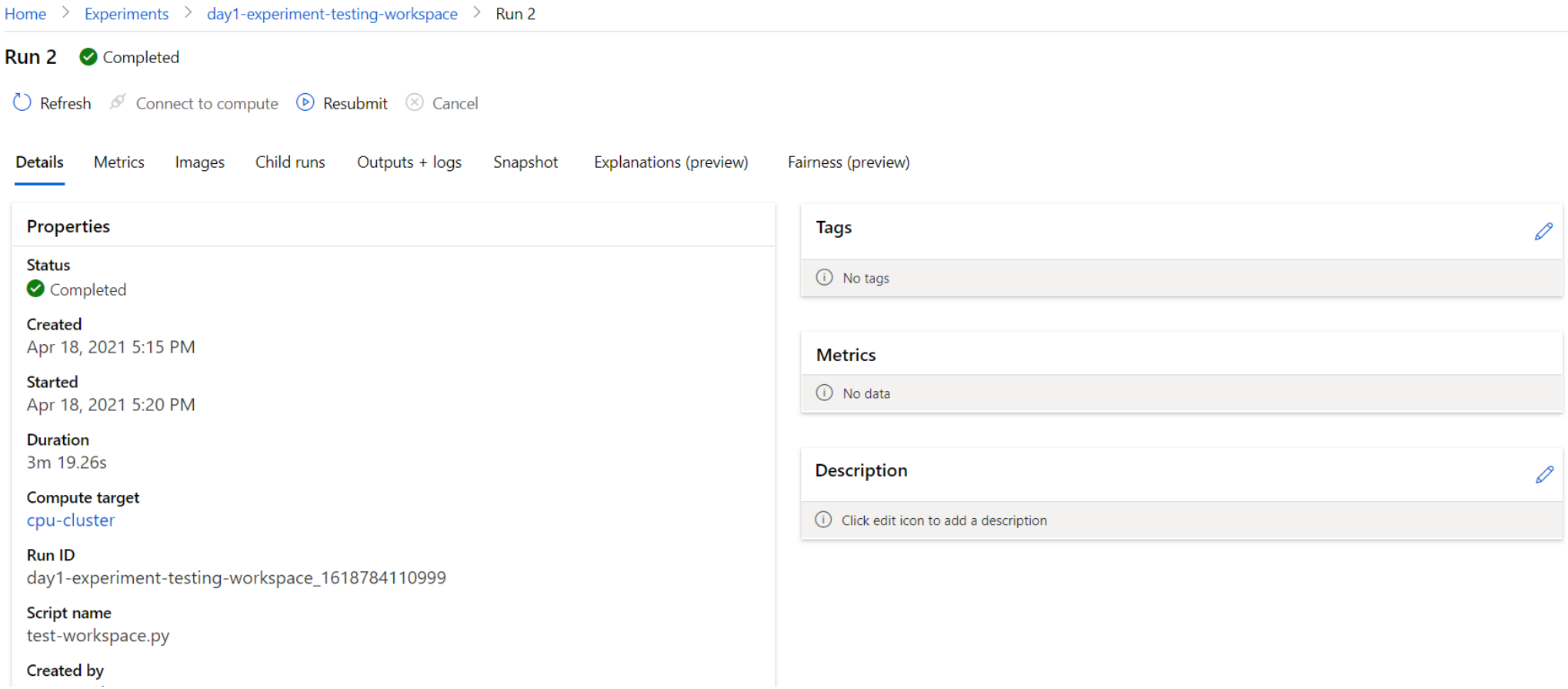
Prueba Workspace resultado

```
(azure-ml) PS C:\Users\Reyson\Documents\Personal\DOCENCIA\Machine Learning UdeA\azure-ml> python .\03-run-test-workspace-remote.py
https://ml.azure.com/runs/day1-experiment-testing-workspace_1618782296_ca4c2d75?wsid=/subscriptions/8e156a0e-3179-4b44-ba79-7f51ee70e15d/resourcegroups/BPT/workspaces/azure-ml&tid=aa018d6b-f5ba-4f79-84f5-87563b44392c
```



Prueba Workspace resultado

Eventualmente podría salir un error de imagen en docker mientras se crea en el cluster



Crear modelo

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = x.view(-1, 16 * 5 * 5)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

Entrenar modelo

train.py

```python
import torch
import torch.optim as optim
import torchvision
import torchvision.transforms as transforms

from model import Net

# download CIFAR 10 data
trainset = torchvision.datasets.CIFAR10(
    root="./data",
    train=True,
    download=True,
    transform=torchvision.transforms.ToTensor(),
)
trainloader = torch.utils.data.DataLoader(
    trainset, batch_size=4, shuffle=True, num_workers=2
)

if __name__ == "__main__":

    # define convolutional network
    net = Net()

    # set up pytorch loss /  optimizer
    criterion = torch.nn.CrossEntropyLoss()
    optimizer = optim.SGD(net.parameters(), lr=0.001, momentum=0.9)

    # train the network
    for epoch in range(2):

        running_loss = 0.0
        for i, data in enumerate(trainloader, 0):
            # unpack the data
            inputs, labels = data

            # zero the parameter gradients
            optimizer.zero_grad()
```

Crear entorno para ejecución para entrenamiento del modelo Local

```
conda deactivate # If you are still using the azure-ml environment, exit it

conda env create -f .azureml/pytorch-env.yml # create the new Conda environment

conda activate pytorch-env # activate new Conda environment

python src/train.py # train model
```

Entrenamiento Local

```
(pytorch-env) PS C:\Users\Reyson\Documents\Personal\DOCENCIA\Machine Learning UdeA\azure-ml> python .\src\train.py
Downloading https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz to ./data\cifar-10-python.tar.gz
64.5%Failed download. Trying https -> http instead. Downloading http://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz to ./data\cifar-10-python.tar.gz
100.0%Extracting ./data\cifar-10-python.tar.gz to ./data
Files already downloaded and verified
Files already downloaded and verified
epoch=1, batch= 2000: loss 2.29
epoch=1, batch= 4000: loss 2.11
epoch=1, batch= 6000: loss 1.98
epoch=1, batch= 8000: loss 1.85
epoch=1, batch=10000: loss 1.75
epoch=1, batch=12000: loss 1.64
Files already downloaded and verified
Files already downloaded and verified
epoch=2, batch= 2000: loss 1.58
epoch=2, batch= 4000: loss 1.53
epoch=2, batch= 6000: loss 1.47
epoch=2, batch= 8000: loss 1.45
epoch=2, batch=10000: loss 1.44
epoch=2, batch=12000: loss 1.42
Finished Training
```

Entrenamiento en Workspace

Conda deactivate

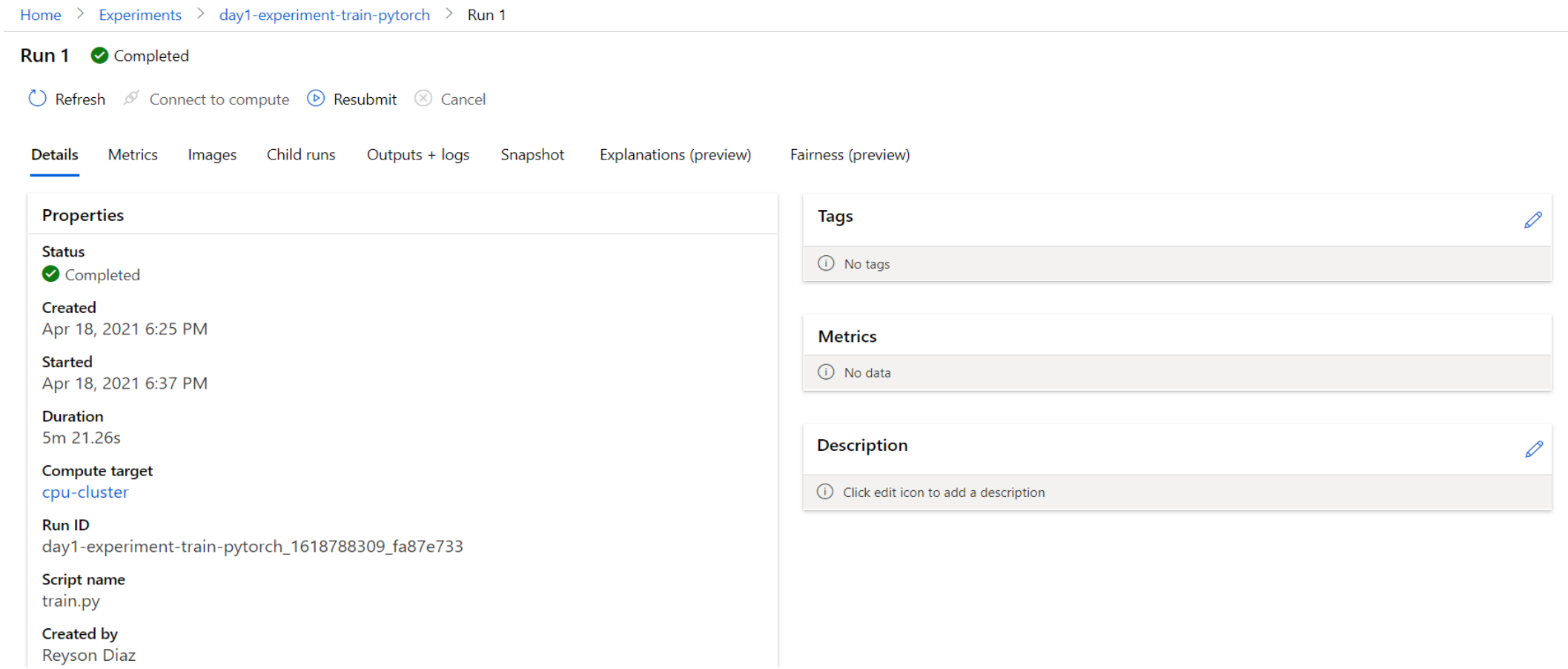
Conda actívate azure-ml (ya contiene las dependencias necesarias para ejecutat pytorch)

python 04-run-pytorch.py

> ⓘ Note
>
> The first time you run this script, Azure Machine Learning will build a new Docker image from your PyTorch environment. The whole run might take 5 to 10 minutes to complete.
>
> You can see the Docker build logs in the Azure Machine Learning studio. Follow the link to the studio, select the **Outputs + logs** tab, and then select `20_image_build_log.txt`.
>
> This image will be reused in future runs to make them run much quicker.

After your image is built, select `70_driver_log.txt` to see the output of your training script.

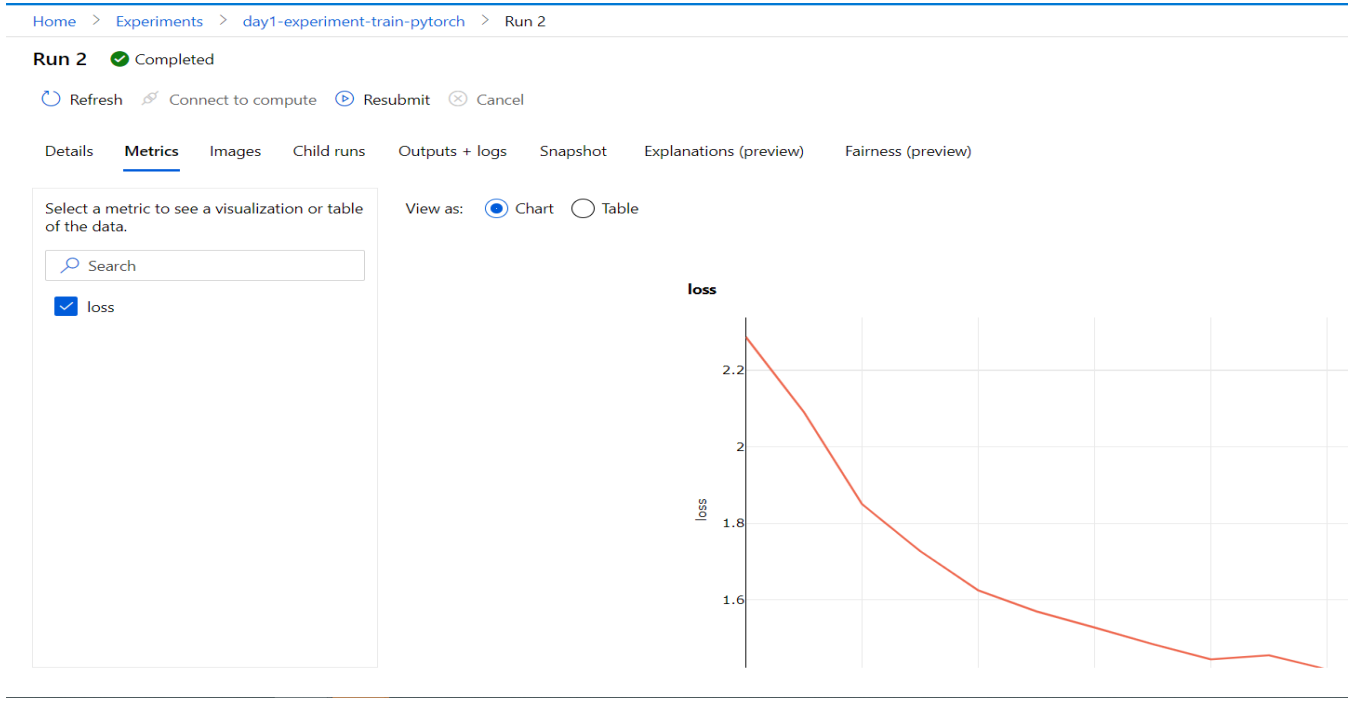
Entrenamiento Workspace resultado



Analizar métricas de las ejecuciones

```
# ADDITIONAL CODE: get AML run from the current context
run = Run.get_context()
```

```
# ADDITIONAL CODE: log loss metric to AML
run.log('loss', loss)
```



Cargar data en el Workspace

Se agregan parámetros para recibir el dataset y los parámetros del modelo como parámetros

```
parser = argparse.ArgumentParser()
parser.add_argument(
    '--data_path',
    type=str,
    help='Path to the training data'
)
parser.add_argument(
    '--learning_rate',
    type=float,
    default=0.001,
    help='Learning rate for SGD'
)
parser.add_argument(
    '--momentum',
    type=float,
    default=0.9,
    help='Momentum for SGD'
)

args = parser.parse_args()
```

Prueba Local

conda deactivate

conda env create -f .azureml/pytorch-env.yml # create the new conda environment with updated dependencias

conda activate pytorch-aml-env # activate new conda environment

python src/train.py --data_path ./data --learning_rate 0.003 --momentum 0.92

```
(pytorch-aml-env) PS C:\Users\Reyson\Documents\Personal\DOCENCIA\Machine Learning UdeA\azure-ml> python src/train.py --data_path ./data --learning_rate 0.003 --momentum
 0.92
===== DATA =====
DATA PATH: ./data
LIST FILES IN DATA PATH...
['cifar-10-batches-py', 'cifar-10-python.tar.gz']
===============
Attempted to log scalar metric loss:
2.0344588753581045
epoch=1, batch= 2000: loss 2.03
Attempted to log scalar metric loss:
1.7685605351626874
epoch=1, batch= 4000: loss 1.77
Attempted to log scalar metric loss:
1.6520960516929626
epoch=1, batch= 6000: loss 1.65
Attempted to log scalar metric loss:
1.6082215337157248
epoch=1, batch= 8000: loss 1.61
Attempted to log scalar metric loss:
1.565536759287119
epoch=1, batch=10000: loss 1.57
Attempted to log scalar metric loss:
1.5561724902763963
epoch=1, batch=12000: loss 1.56
Attempted to log scalar metric loss:
1.5000996266305446
epoch=2, batch= 2000: loss 1.50
Attempted to log scalar metric loss:
1.4908698590099811
epoch=2, batch= 4000: loss 1.49
Attempted to log scalar metric loss:
1.503056163161993
epoch=2, batch= 6000: loss 1.50
Attempted to log scalar metric loss:
1.4760852197669447
epoch=2, batch= 8000: loss 1.48
Attempted to log scalar metric loss:
1.458710665397346
epoch=2, batch=10000: loss 1.46
Attempted to log scalar metric loss:
1.4679711819589139
epoch=2, batch=12000: loss 1.47
Finished Training
```

Cargar dataset en Azure

05-upload-data.py

```python
# 05-upload-data.py
from azureml.core import Workspace
ws = Workspace.from_config()
datastore = ws.get_default_datastore()
datastore.upload(src_dir='./data',
                 target_path='datasets/cifar10',
                 overwrite=True)


```

Tip

While you're using Azure Machine Learning to upload the data, you can use **Azure Storage Explorer** to upload ad hoc files. If you need an ETL tool, you can use **Azure Data Factory** to ingest your data into Azure.

http://www.cs.toronto.edu/~kriz/cifar.html

Cargar dataset en Azure verificación

```
(pytorch-aml-env) PS C:\Users\Reyson\Documents\Personal\DOCENCIA\Machine Learning UdeA\azure-ml> python .\05-upload-data.py
Uploading an estimated of 9 files
Uploading ./data\cifar-10-batches-py\batches.meta
Uploaded ./data\cifar-10-batches-py\batches.meta, 1 files out of an estimated total of 9
Uploading ./data\cifar-10-batches-py\readme.html
Uploaded ./data\cifar-10-batches-py\readme.html, 2 files out of an estimated total of 9
Uploading ./data\cifar-10-batches-py\test_batch
Uploaded ./data\cifar-10-batches-py\test_batch, 3 files out of an estimated total of 9
Uploading ./data\cifar-10-batches-py\data_batch_5
Uploaded ./data\cifar-10-batches-py\data_batch_5, 4 files out of an estimated total of 9
Uploading ./data\cifar-10-batches-py\data_batch_2
Uploaded ./data\cifar-10-batches-py\data_batch_2, 5 files out of an estimated total of 9
Uploading ./data\cifar-10-batches-py\data_batch_1
Uploaded ./data\cifar-10-batches-py\data_batch_1, 6 files out of an estimated total of 9
Uploading ./data\cifar-10-batches-py\data_batch_3
Uploaded ./data\cifar-10-batches-py\data_batch_3, 7 files out of an estimated total of 9
Uploading ./data\cifar-10-batches-py\data_batch_4
Uploaded ./data\cifar-10-batches-py\data_batch_4, 8 files out of an estimated total of 9
Uploading ./data\cifar-10-python.tar.gz
Uploaded ./data\cifar-10-python.tar.gz, 9 files out of an estimated total of 9
Uploaded 9 files
(pytorch-aml-env) PS C:\Users\Reyson\Documents\Personal\DOCENCIA\Machine Learning UdeA\azure-ml>
```

Authentication method: Access key (Switch to Azure AD User Account)
Location: azureml-blobstore-e2374ea0-b328-4f72-b3eb-f5c8d34924cd / datasets / cifar10 / cifar-10-batches-py

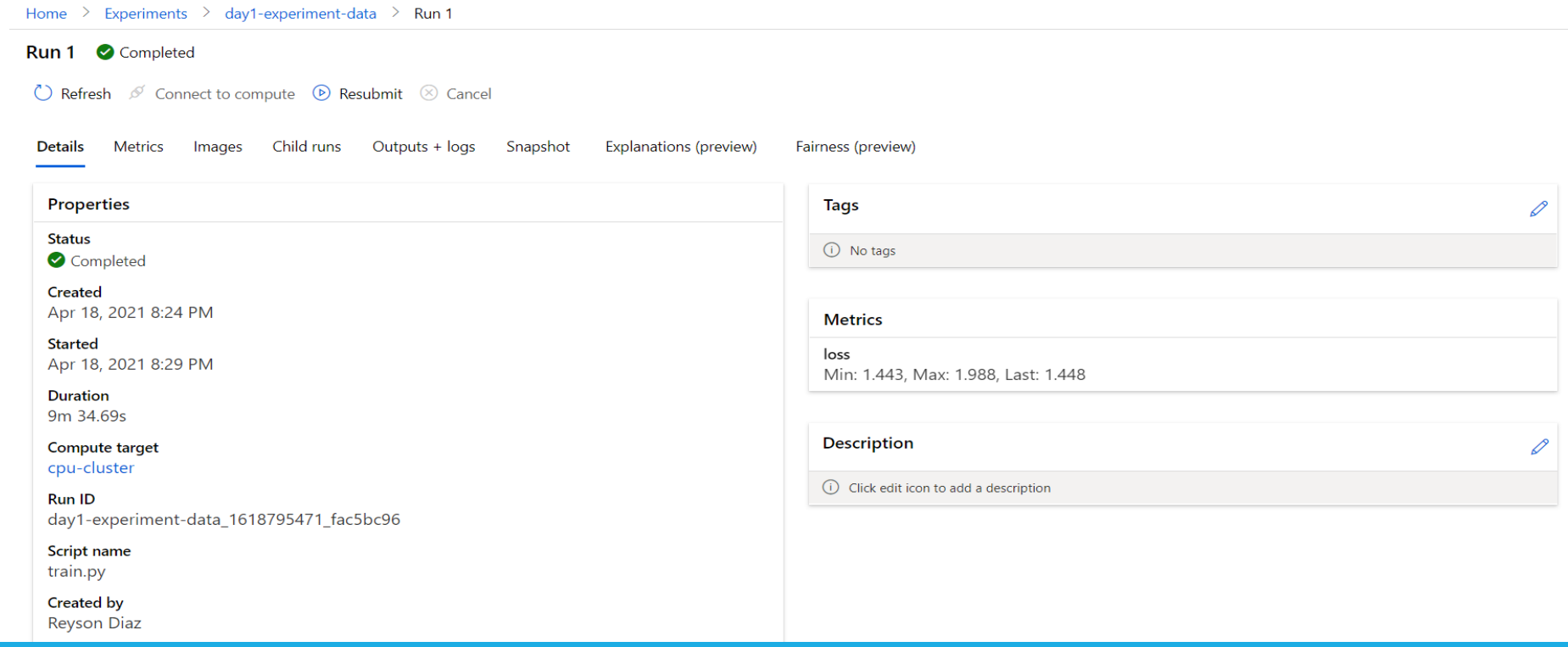
Search blobs by prefix (case-sensitive)

Name	Modified	Access tier
[..]		
batches.meta	4/18/2021, 8:10:56 PM	Hot (Inferred)
data_batch_1	4/18/2021, 8:14:37 PM	Hot (Inferred)
data_batch_2	4/18/2021, 8:14:36 PM	Hot (Inferred)
data_batch_3	4/18/2021, 8:14:44 PM	Hot (Inferred)
data_batch_4	4/18/2021, 8:14:44 PM	Hot (Inferred)
data_batch_5	4/18/2021, 8:14:35 PM	Hot (Inferred)
readme.html	4/18/2021, 8:10:56 PM	Hot (Inferred)
test_batch	4/18/2021, 8:14:25 PM	Hot (Inferred)

Ejecutar entrenamiento con dataset en Azure

```
(pytorch-aml-env) PS C:\Users\Reyson\Documents\Personal\DOCENCIA\Machine Learning UdeA\azure-ml> python .\06-run-pytorch-data.py
Submitted to compute cluster. Click link below

https://ml.azure.com/runs/day1-experiment-data_1618795471_fac5bc96?wsid=/subscriptions/8e156a0e-3179-4b44-ba79-7f51ee70e15d/resourcegroups/BPT/workspaces/azure-ml&tid=a
a018d6b-f5ba-4f79-84f5-87563b44392c
```



Deplegar Modelo

1. Registrar modelo
2. Preparar configuración de inferencia
3. Preparar script de entrada
4. Seleccionar computo de destino
5. Desplegar el modelo
6. Probar resultados en un Webservice

Registrar modelo Local

```
Pkl_Filename = "outputs/cifar_10_model.pkl"

with open(Pkl_Filename, 'wb') as file:
    pickle.dump(net, file)
```

```
# 07-model-registration-azure.py
from azureml.core import Workspace
from azureml.core import Model

if __name__ == "__main__":
    ws = Workspace.from_config()

    model = Model.register(model_name='cifar_10',
                           tags={'area': 'udea_training'},
                           model_path='outputs/cifar_10_model.pkl',
                           workspace = ws)
    print(model.name, model.id, model.version, sep='\t')
```

Fix LocalService

```
conda install -c anaconda pywin32
```

Docker python

https://docs.docker.com/language/python/build-images/