

# Azure Machine Learning - (DL)

Mutaz Abu Ghazaleh | AI/ML Specialist  
26/04/2022



# Agenda (DL)

- Azure Portal
- Machine Learning Workspace
- Compute Instance – GPU
- Storage for saving images
- Notebooks and VS Code -> work with images
- Simple Cats/Dogs vision dataset

---

- Experiments and Tracking
- Saving model artifacts
- Jobs -> advanced (time permitting)

Get Azure

# Get Azure

## Azure Free Account

Popular  
services free  
for 12 months

40+ other  
services free  
always

+

**Start with USD200\* Azure credit**

You'll have **30 days** to use it—in addition to free services.

<https://azure.microsoft.com/en-au/free/>

## Azure for Students

Start with \$100  
Azure credit

No credit card  
required

+

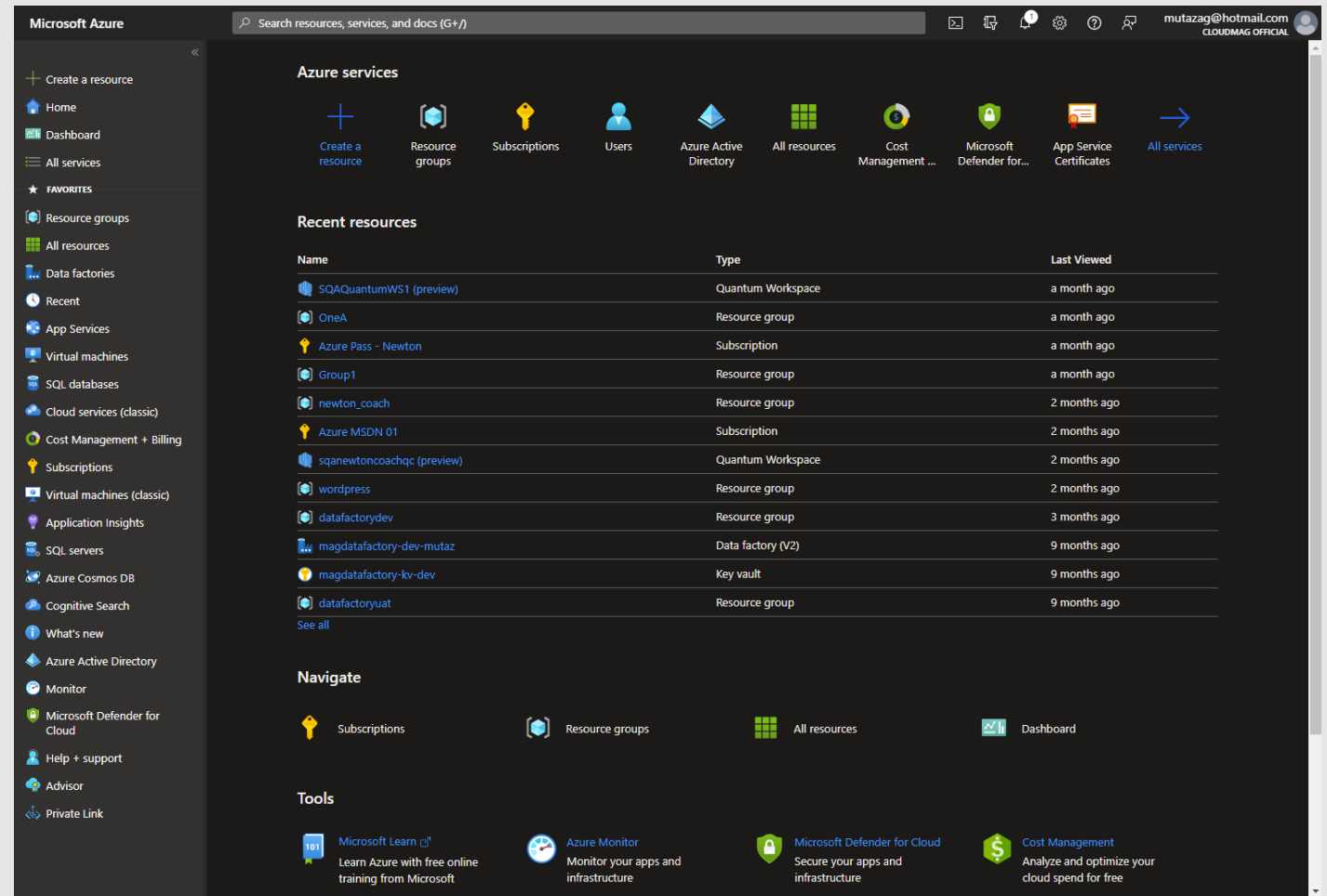
**Free services**

Get popular services free while you have your credit.

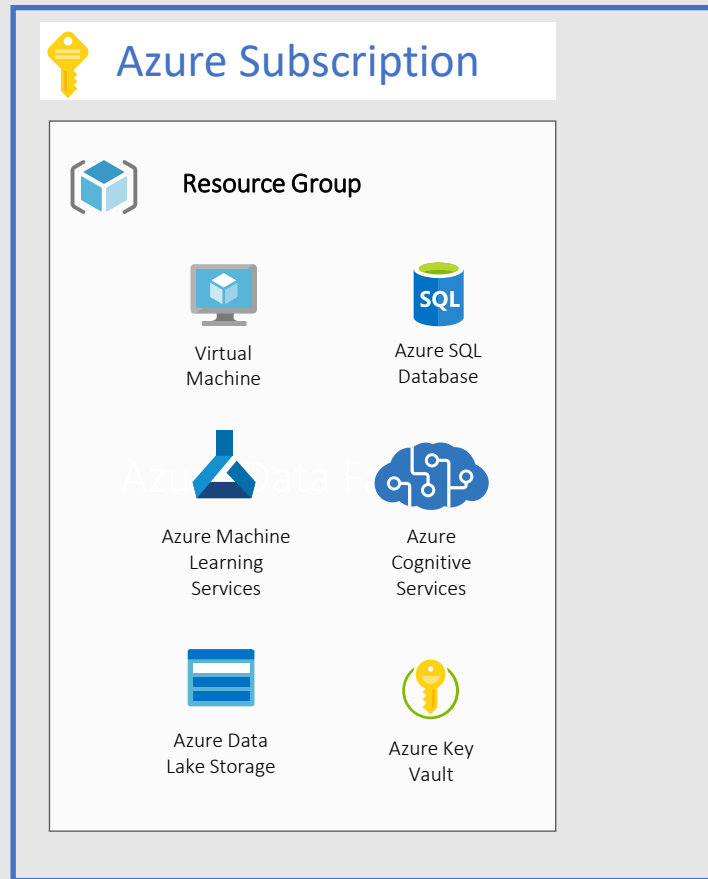
<https://azure.microsoft.com/en-us/free/students/>

# Azure Portal

- A single portal to access all applications in your Azure Subscription
- Build, manage and monitor Azure resources
- Login with an Azure account:  
<https://portal.azure.com>



# Hierarchy in Azure



**Azure account:** The email address that you provide when you create an Azure subscription is the Azure account for the subscription.

**Subscription:** A logical container for your resources. Each Azure resource is associated with only one subscription. Creating a subscription is the first step in adopting Azure.

**Resource groups:** Logical containers that you use to group related resources in a subscription. They're commonly used to represent a collection of assets that are required to support a workload, application, or specific function within a subscription.

**Resources:** An entity that's managed by Azure. Examples include Azure Virtual Machines, Machine Learning Services, SQL Database and storage accounts.

# Azure Products

## AI + Machine Learning

Analytics

Compute

Containers

Databases

Developer Tools

DevOps

Hybrid + multicloud

Identity

Integration

Internet of Things

Management and Governance

Media

Migration

Mixed Reality

Mobile

Networking

Security

Storage

Virtual desktop infrastructure

Web

## AI + Machine Learning

### Anomaly Detector

Easily add anomaly detection capabilities to your apps

### Azure Cognitive Search

AI-powered cloud search service for mobile and web app development

### Azure Machine Learning

Bring AI to everyone with an end-to-end, scalable, trusted platform with experimentation and model management

### Azure Video Analyzer for Media

Unlock video insights

### Custom Vision

Easily customize your own state-of-the-art computer vision models for your unique use case

### Form Recognizer

The AI-powered document extraction service that understands your forms

### Kinect DK

Build computer vision and speech models using a developer kit with advanced AI sensors

### Microsoft Genomics

Power genome sequencing & research insights

### QnA Maker

Distill information into conversational, easy-to-navigate answers

### Speech Translation

Easily integrate real-time speech translation to your app

### Translator

Easily conduct machine translation with a simple REST API call

### Azure Applied AI Services

Specialized services that enable organizations to accelerate time to value in applying AI to solve common scenarios

### Azure Cognitive Services

Add smart API capabilities to enable contextual interactions

### Azure Open Datasets

Cloud platform to host and share curated open datasets to accelerate development of machine learning models

### Computer Vision

Distill actionable information from images

### Data Science Virtual Machines

Rich pre-configured environment for AI development

### Health Bot

A managed service purpose-built for development of virtual healthcare assistants.

### Language Understanding

Teach your apps to understand commands from your users

### Personalizer

An AI service that delivers a personalized user experience

### Speaker Recognition (Preview)

Use speech to identify and verify individual speakers

### Text Analytics

Easily evaluate sentiment and topics to understand what users want

### Azure Bot Service

Intelligent, serverless bot service that scales on demand

### Azure Databricks

Fast, easy, and collaborative Apache Spark-based analytics platform

### Azure Video Analyzer (preview)

Build intelligent video-based applications using the AI of your choice

### Content Moderator

Automated image, text, and video moderation

### Face

Detect, identify, analyze, organize, and tag faces in photos

### Immersive Reader

Empower users of all ages and abilities to read and comprehend text

### Metrics Advisor

An AI service that monitors metrics and diagnoses issues

### Project Bonsai (Preview)

A machine teaching service for creating intelligent industrial control systems using simulations

### Speech to Text

Convert spoken audio to text for more natural interactions

### Text to Speech

Convert text to speech to create more natural, accessible interfaces

+ Create a resource

Home

Dashboard

All services

FAVORITES

Resource groups

All resources

Data factories

Recent

App Services

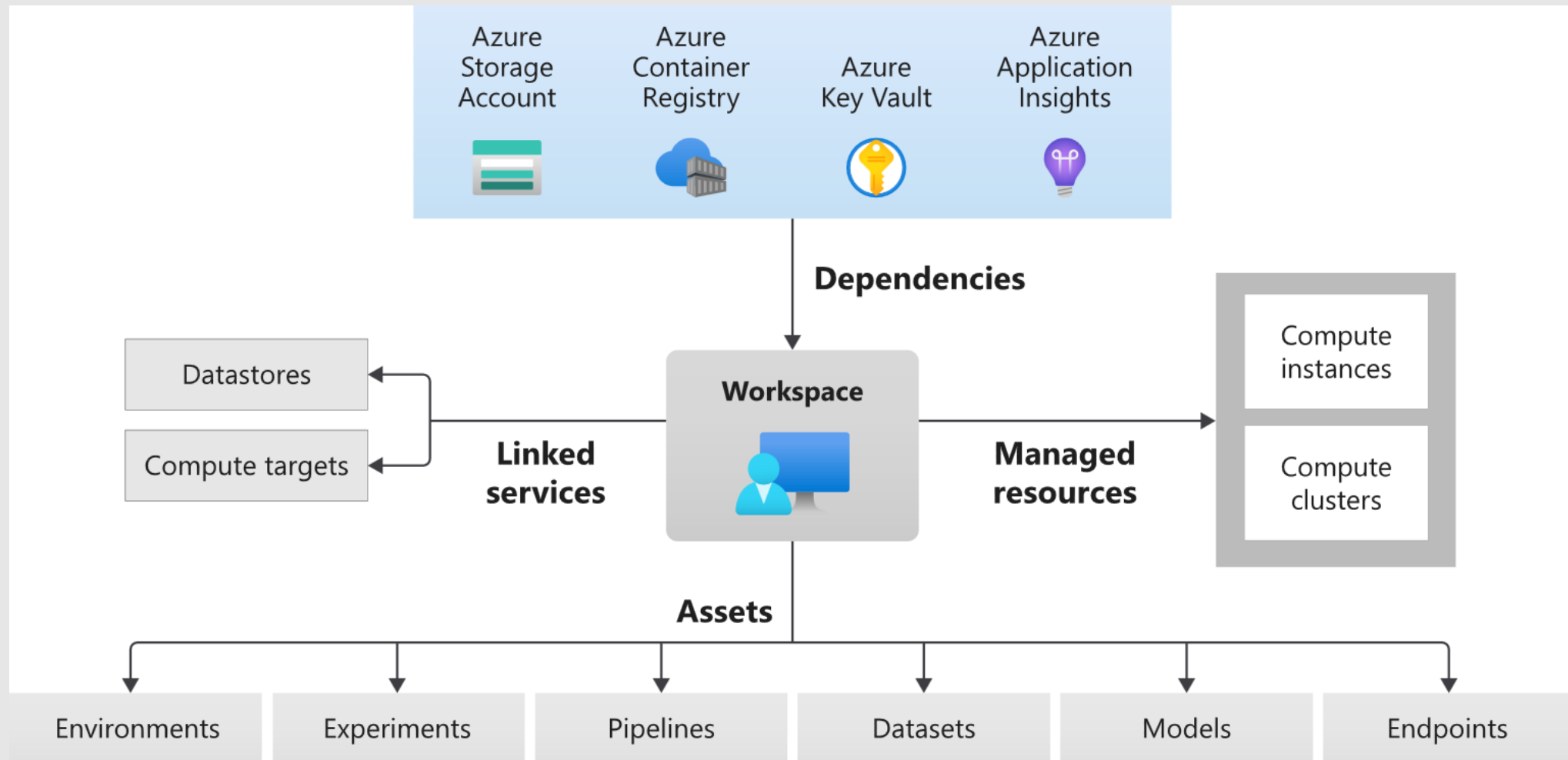
# Exercise: Create Subscription and Workspace

- Create an Azure Subscription:
  - Claim your student subscription or get a free azure subscription
- Create a Machine Learning Workspace

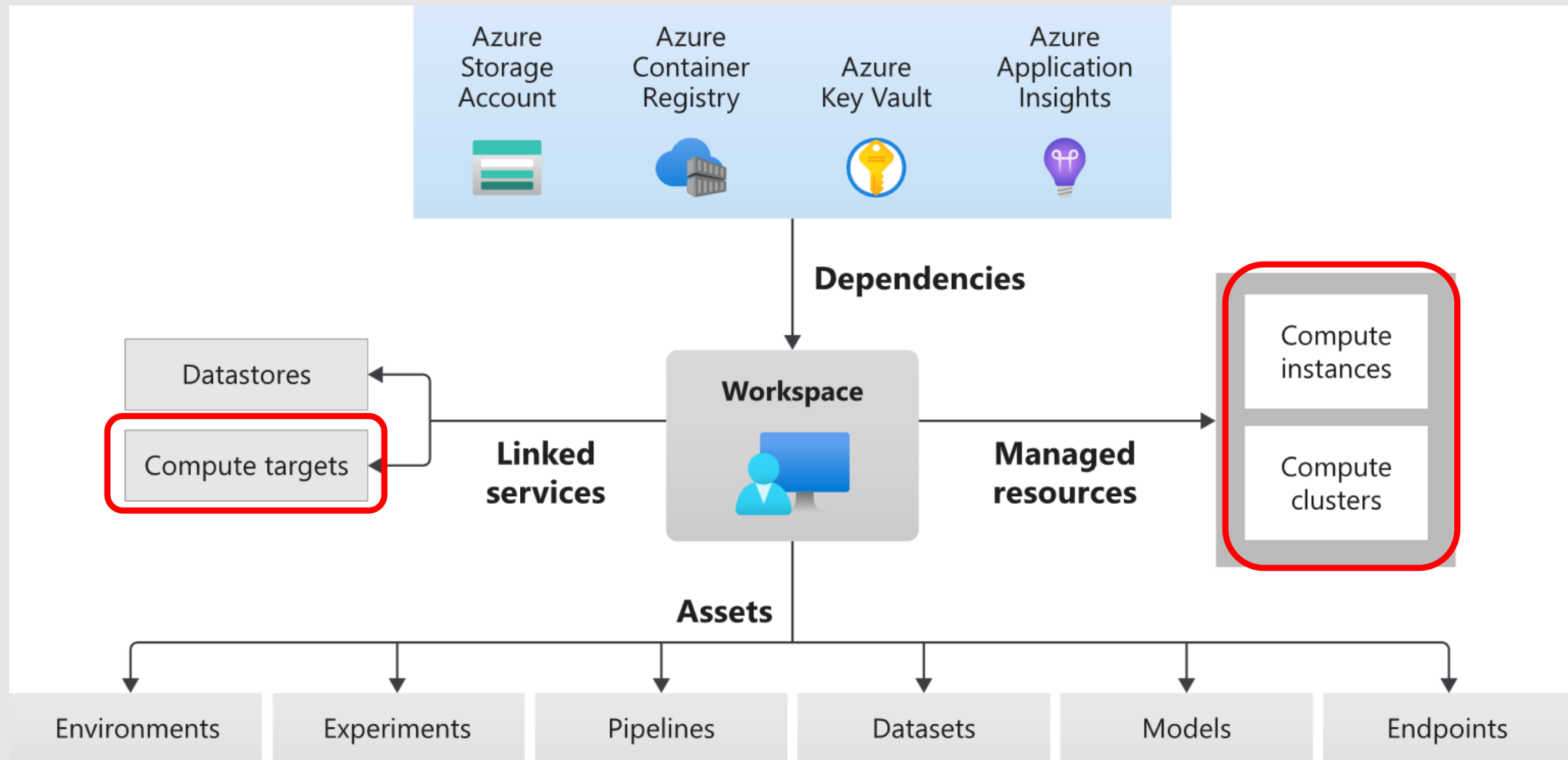


# Machine Learning Workspace

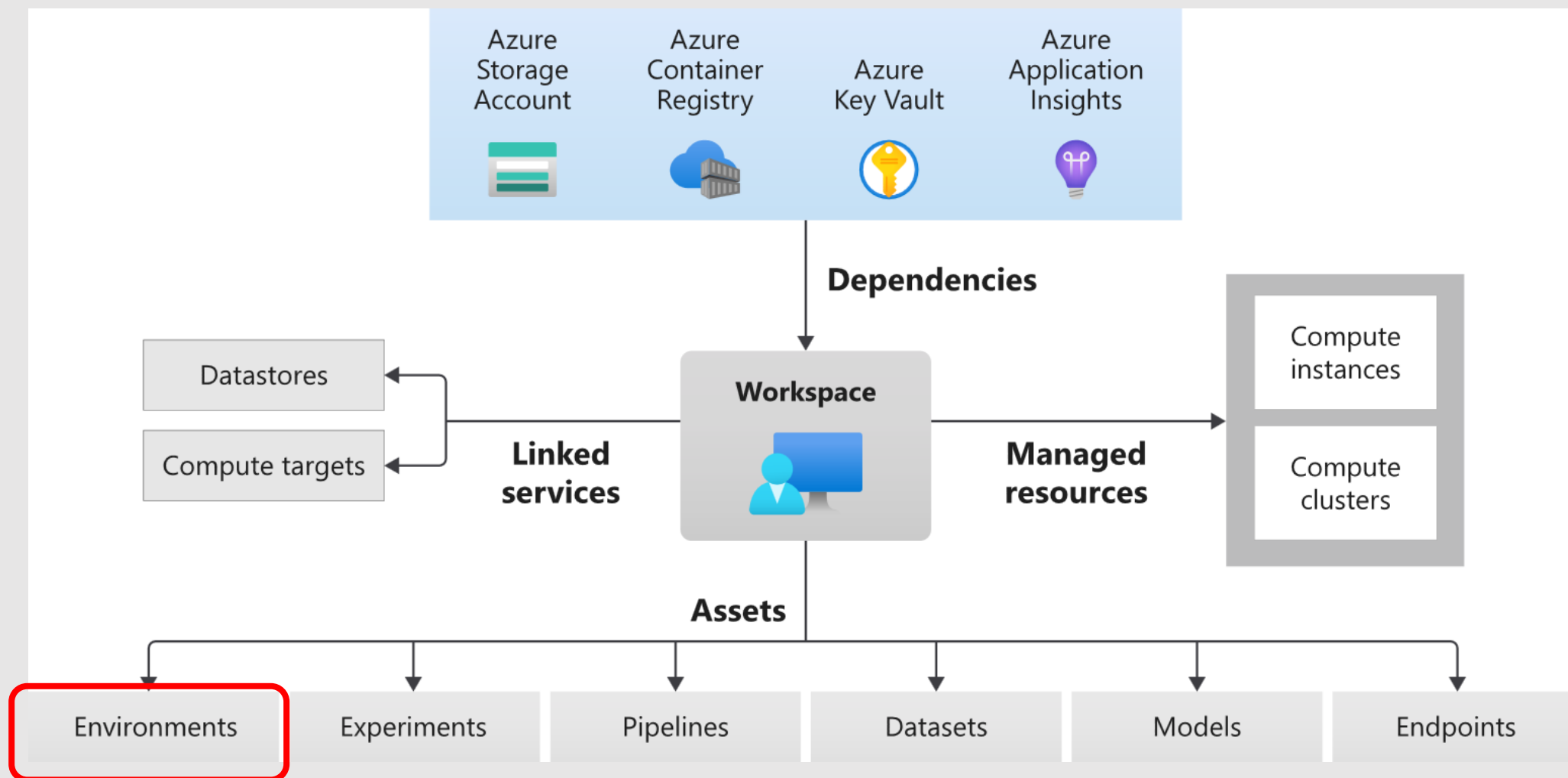
# Machine Learning Workspace



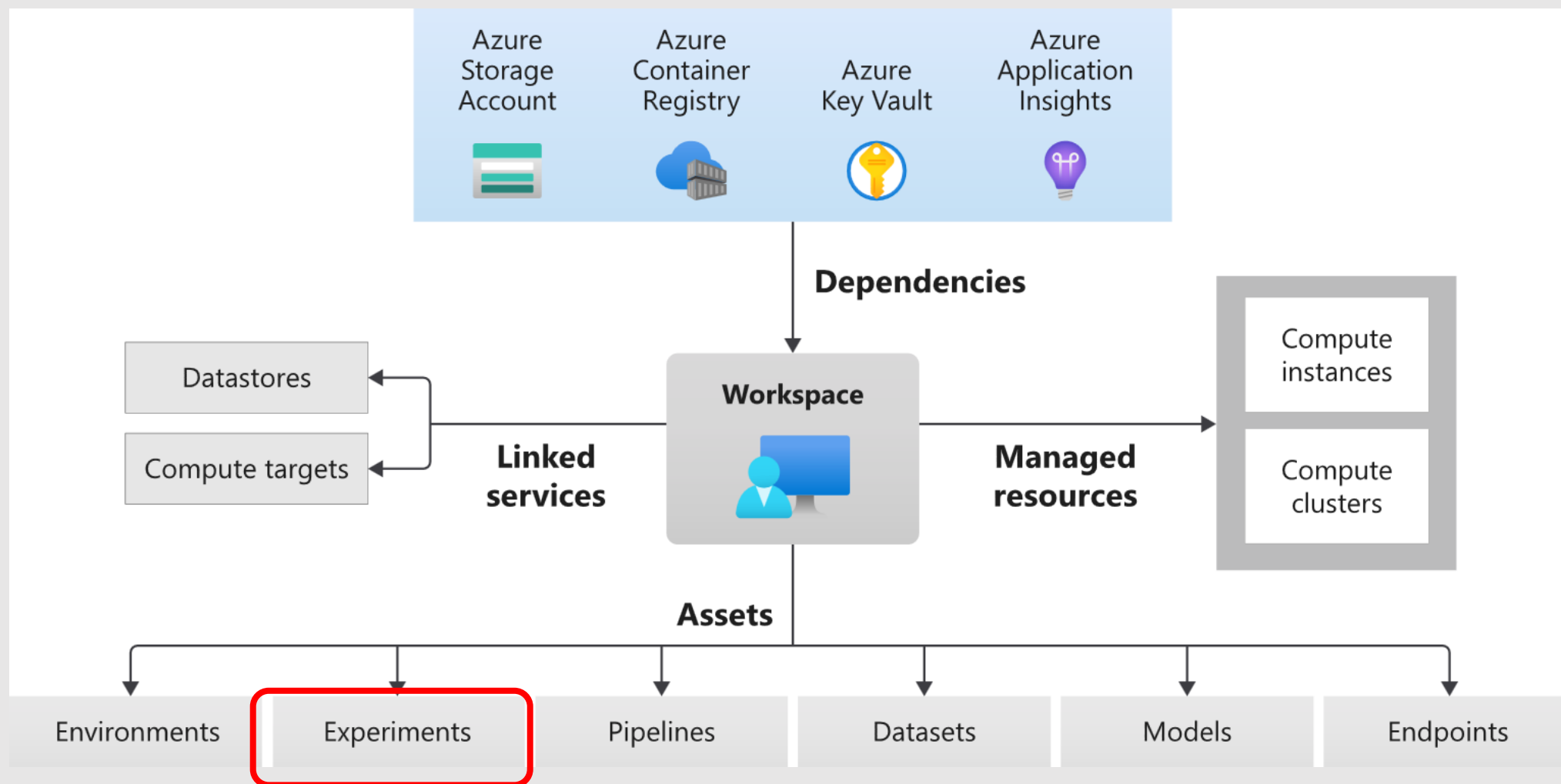
# Compute



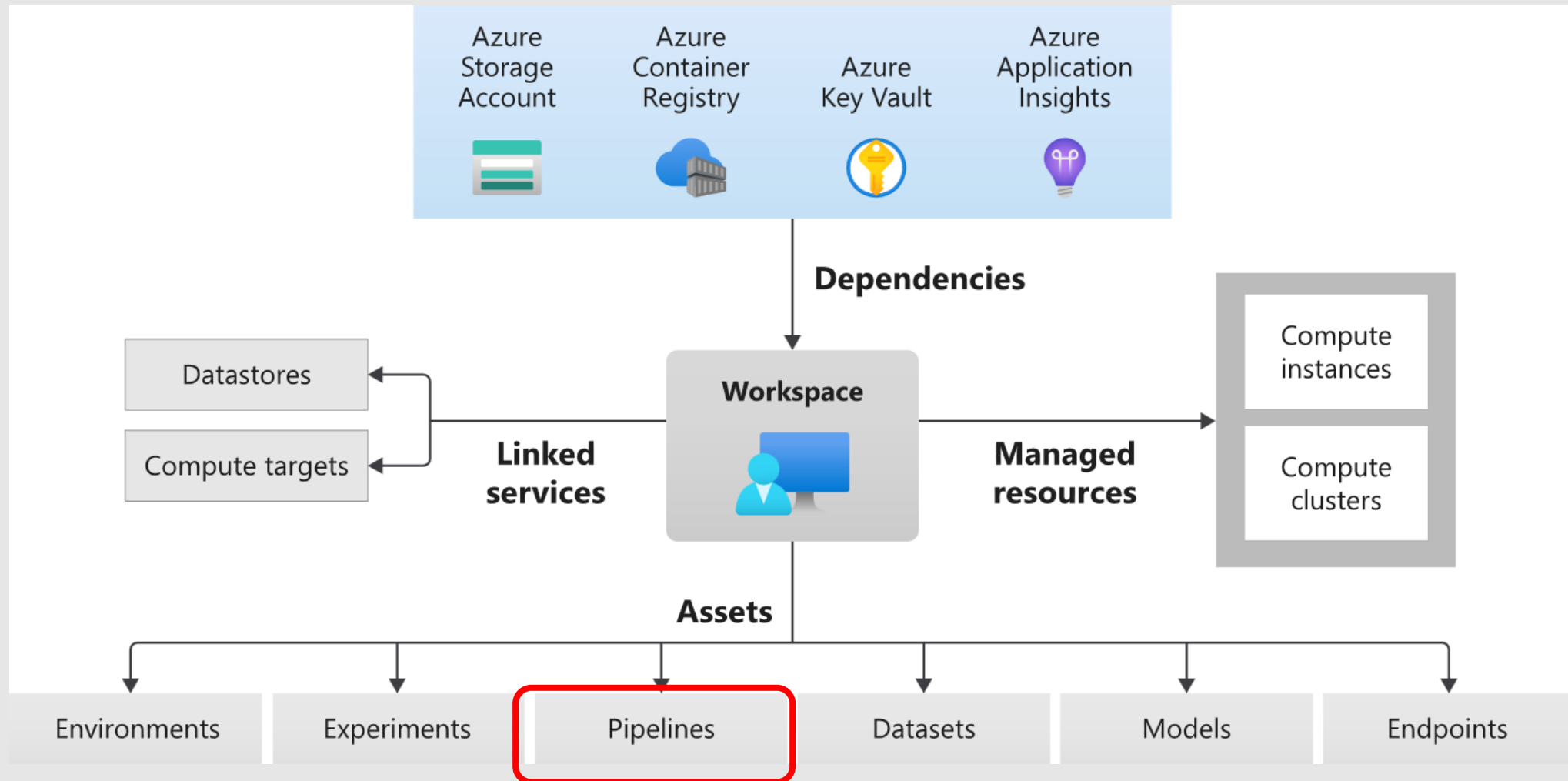
# Environments



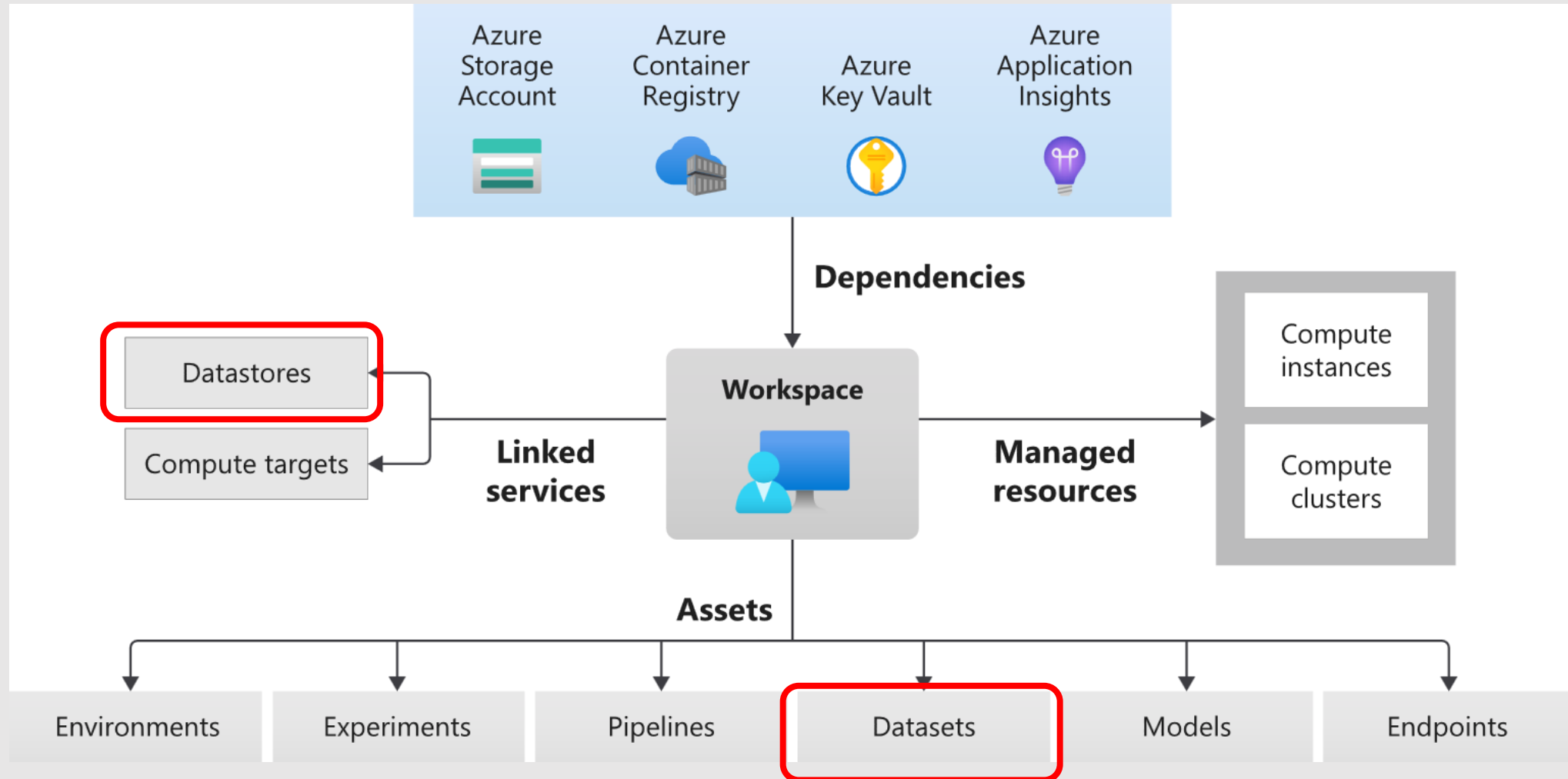
# Experiments



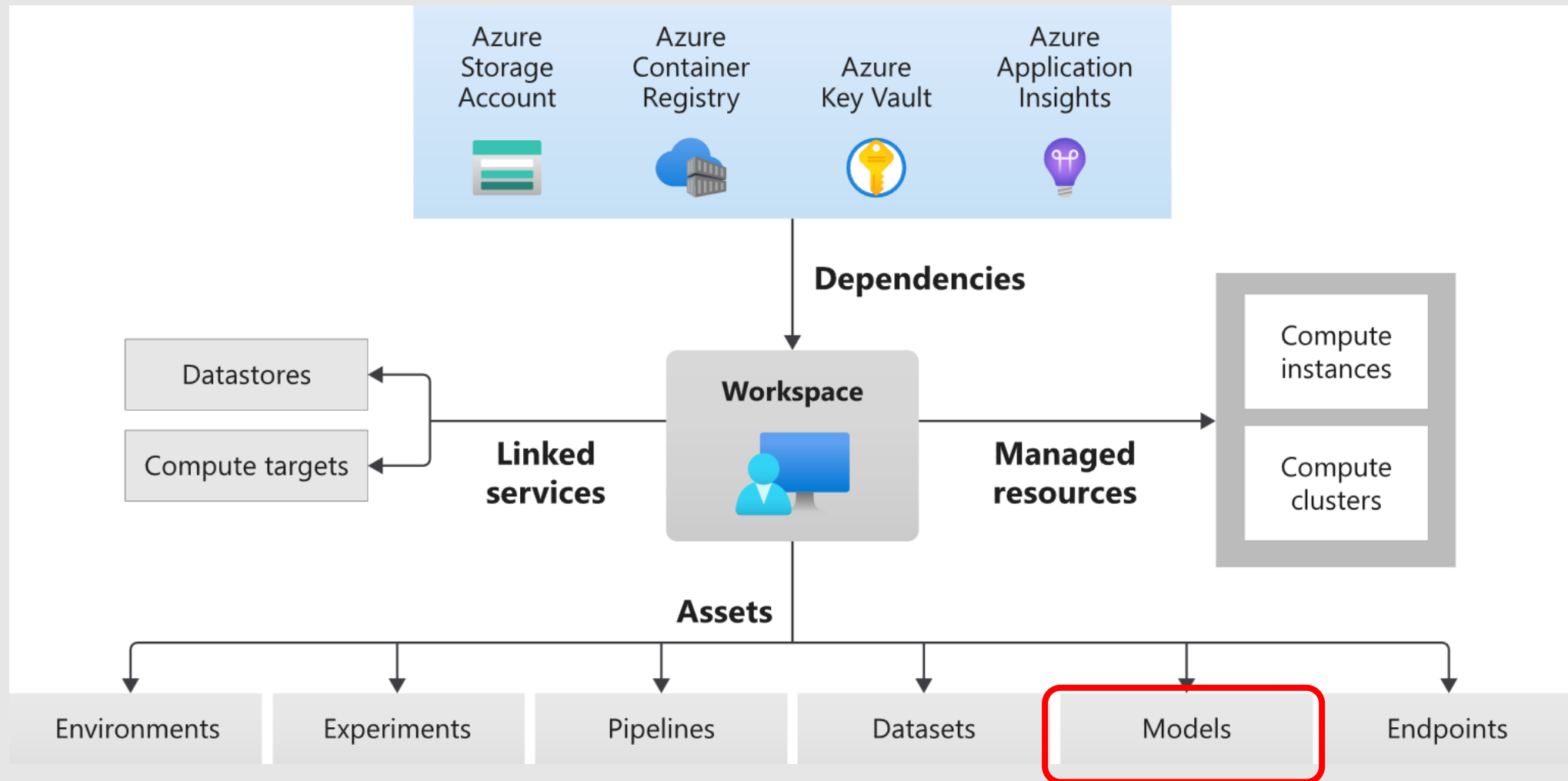
# Pipelines



# Datastores and Datasets

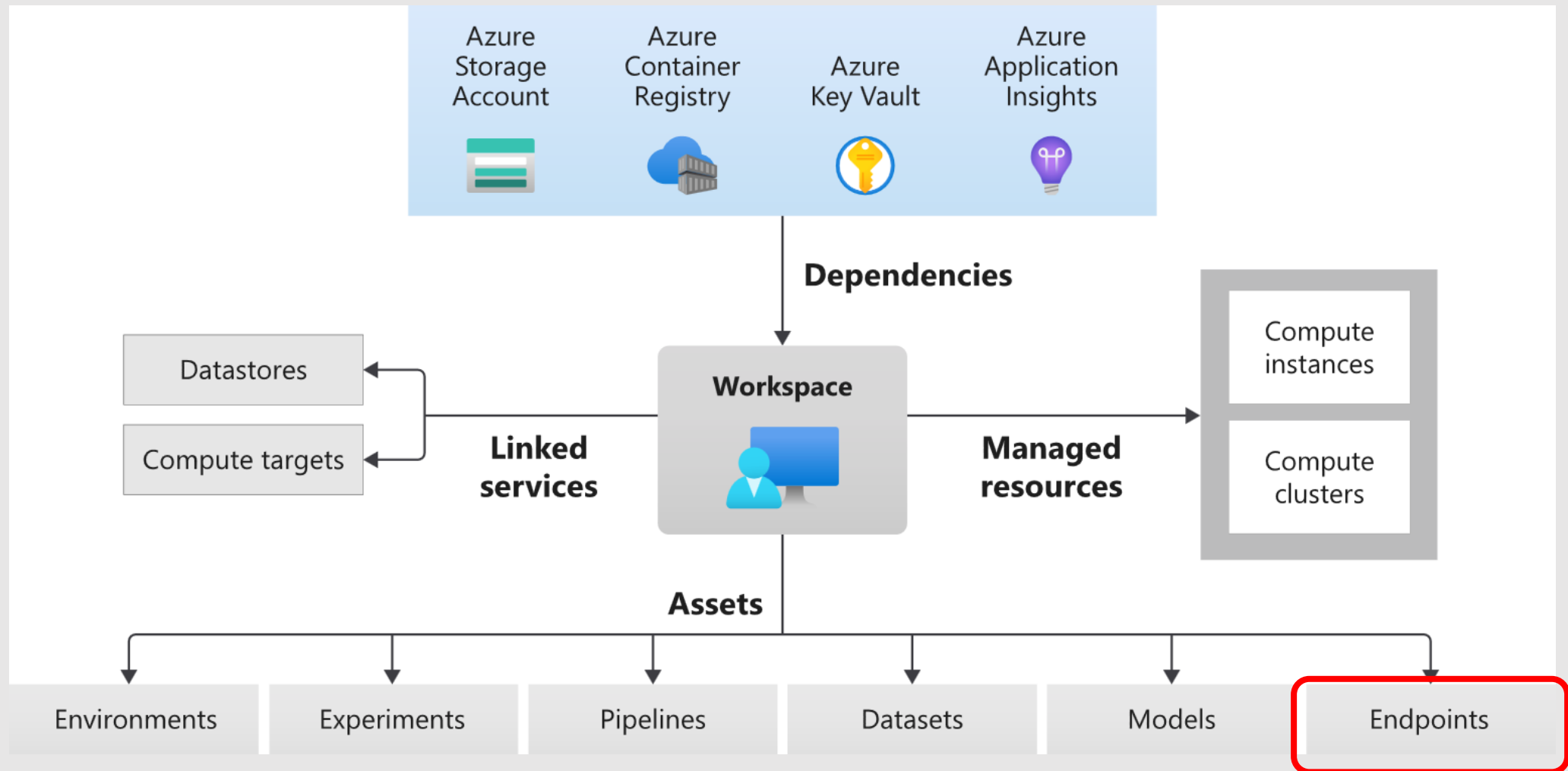


# Models

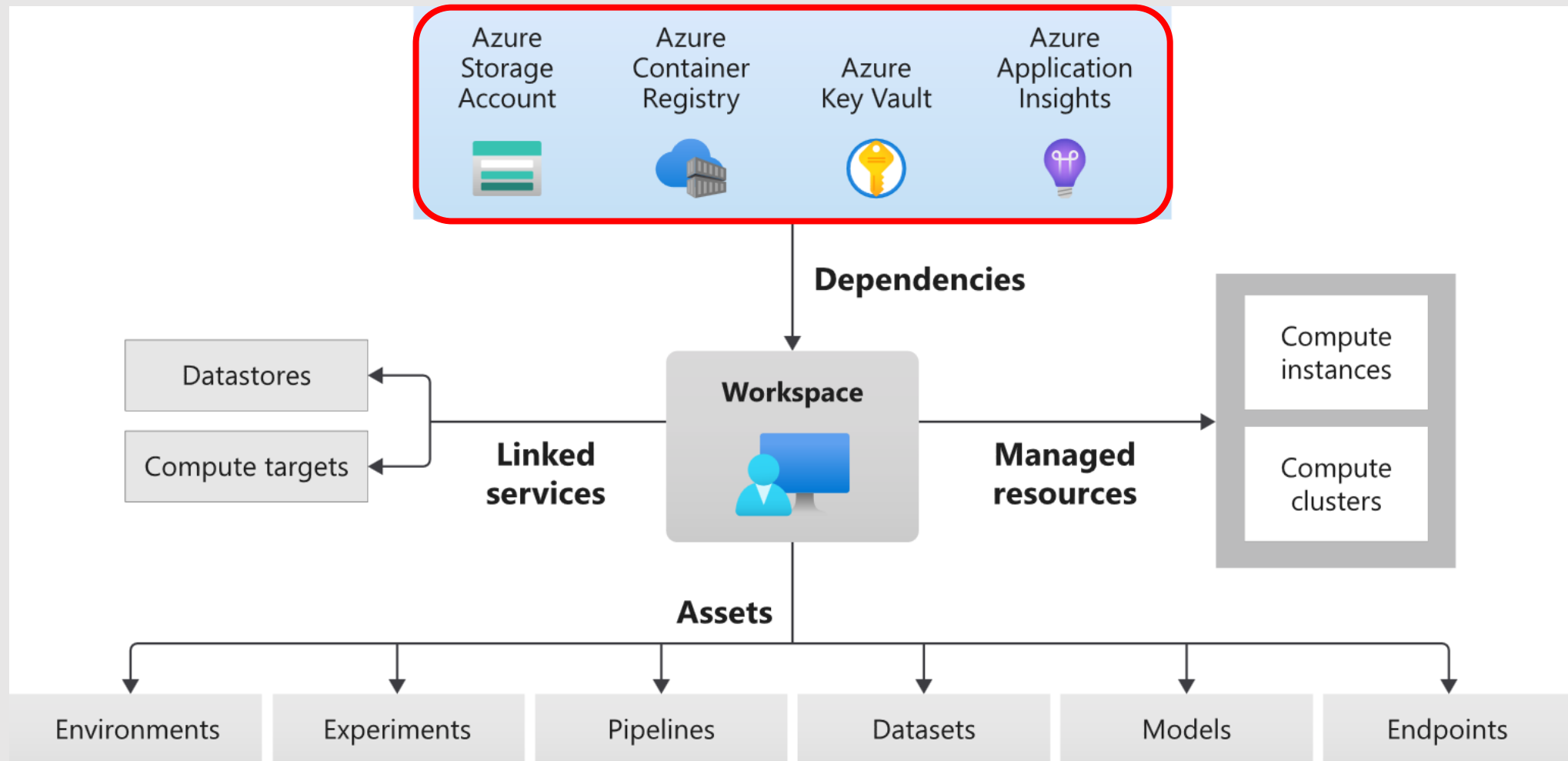




# Endpoints



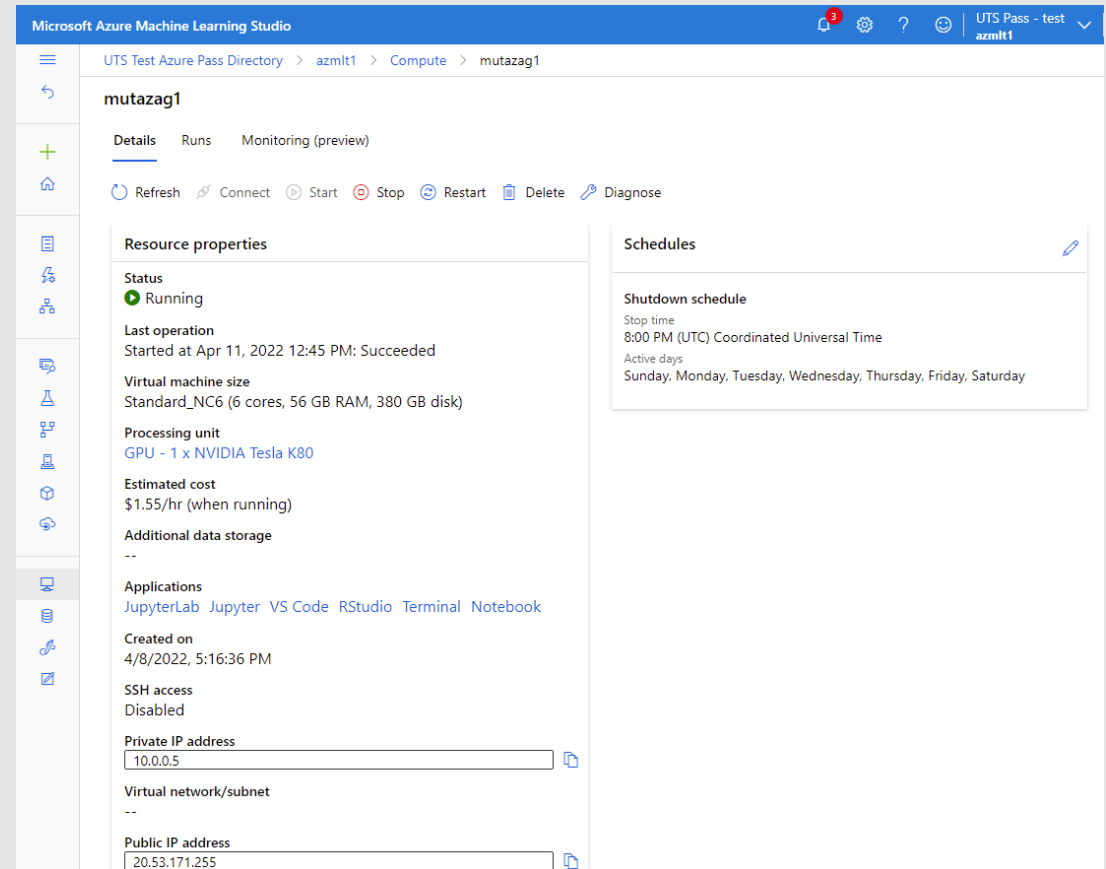
# Workspace Services



Compute

# Compute Instance

- Provision a compute instance
  - STANDARD\_DS3\_V2 (4 cores, 14 GB RAM, 28 GB disk)
  - GPU: Standard\_NC4as\_T4\_v3 or Standard\_NC6
  - Set shutdown schedule
- Use for interactive development in:
  - VSCode
  - Jupyter
  - Notebook
- Submit long running jobs using Azure Machine Learning SDK or CLI



The screenshot displays the Microsoft Azure Machine Learning Studio interface. The top navigation bar shows the path: UTS Test Azure Pass Directory > azmlt1 > Compute > mutazag1. The main content area is titled 'mutazag1' and has tabs for 'Details', 'Runs', and 'Monitoring (preview)'. The 'Details' tab is active, showing various properties and actions for the compute instance.

**Resource properties**

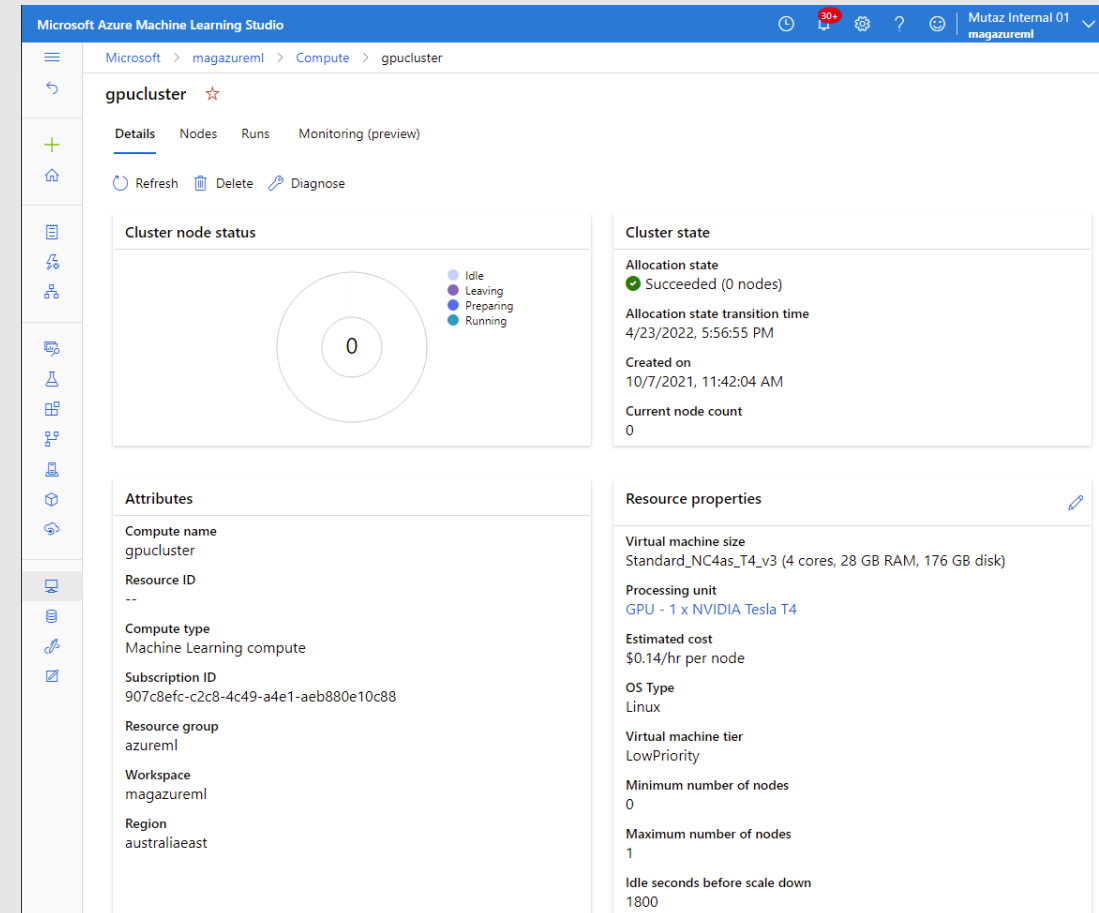
- Status: Running
- Last operation: Started at Apr 11, 2022 12:45 PM: Succeeded
- Virtual machine size: Standard\_NC6 (6 cores, 56 GB RAM, 380 GB disk)
- Processing unit: GPU - 1 x NVIDIA Tesla K80
- Estimated cost: \$1.55/hr (when running)
- Additional data storage: --
- Applications: JupyterLab, Jupyter, VS Code, RStudio, Terminal, Notebook
- Created on: 4/8/2022, 5:16:36 PM
- SSH access: Disabled
- Private IP address: 10.0.0.5
- Virtual network/subnet: --
- Public IP address: 20.53.171.255

**Schedules**

- Shutdown schedule: Stop time 8:00 PM (UTC) Coordinated Universal Time, Active days Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday

# Compute Cluster

- Provision with GPU options: Standard\_NC4as\_T4\_v3 or Standard\_NC6
- Specify min and max number of nodes in a cluster
- Use for running jobs or training pipelines
- Submit long running jobs using Azure Machine Learning SDK or CLI



The screenshot displays the Microsoft Azure Machine Learning Studio interface for a compute cluster named 'gpuccluster'. The interface is divided into several sections:

- Navigation Bar:** Shows the path 'Microsoft > magazureml > Compute > gpuccluster'.
- Cluster Name:** 'gpuccluster' with a star icon.
- Tabs:** 'Details' (selected), 'Nodes', 'Runs', and 'Monitoring (preview)'.
- Actions:** 'Refresh', 'Delete', and 'Diagnose'.
- Cluster node status:** A donut chart showing the status of nodes. The chart is currently empty, indicating 0 nodes. A legend shows four states: Idle (light blue), Leaving (dark blue), Preparing (purple), and Running (teal).
- Cluster state:**
  - Allocation state:** Succeeded (0 nodes)
  - Allocation state transition time:** 4/23/2022, 5:56:55 PM
  - Created on:** 10/7/2021, 11:42:04 AM
  - Current node count:** 0
- Attributes:**
  - Compute name:** gpuccluster
  - Resource ID:** --
  - Compute type:** Machine Learning compute
  - Subscription ID:** 907c8efc-c2c8-4c49-a4e1-aeb880e10c88
  - Resource group:** azureml
  - Workspace:** magazureml
  - Region:** australiaeast
- Resource properties:**
  - Virtual machine size:** Standard\_NC4as\_T4\_v3 (4 cores, 28 GB RAM, 176 GB disk)
  - Processing unit:** GPU - 1 x NVIDIA Tesla T4
  - Estimated cost:** \$0.14/hr per node
  - OS Type:** Linux
  - Virtual machine tier:** LowPriority
  - Minimum number of nodes:** 0
  - Maximum number of nodes:** 1
  - Idle seconds before scale down:** 1800

# Applications on Compute Instances

Microsoft Azure Machine Learning Studio

Microsoft > magazureml > Compute

## Compute

Compute instances    Compute clusters    Inference clusters    Attached computes

+ New    Refresh    Start    Stop    Restart    Delete    Schedule    Edit columns    Reset view    View quota

magazureml02

Show all instances    State    All filters    Clear all

Name	State	Applications	Size	Created on
magazureml02	Running	JupyterLab   Jupyter   VS Code   RStudio   Terminal   Notebook	STANDARD_DS3_V2	Oct 1, 2021 2:3:

# Exercise: Create Compute Instance

- Create Compute Instance, and optionally a GPU cluster (1 node)
- Launch Terminal on CI and run:

	Command	Result
Conda environments	<code>conda env list</code>	Tensorflow env is azureml_py38_PT_TF
Azure cli version	<code>az version -o table</code>	We want version >= 2.15.0
Azure ML cli version	<code>az extension list -o table</code>	Looking for cli v2
Git version	<code>git version</code>	
Configured defaults	<code>az configure</code>	[defaults] workspace = <workspace name> group = <resource group name>
Login to Azure	<code>az login --identity</code>	Cli context logged in to Azure
Get mlflow tracking URI	<code>az ml workspace show --query mlflow_tracking_uri</code>	"azureml://australiaeast.api.azureml.ms/mlflow/v1.0 ..."
Show Storage Account Key	<code>az ml workspace list-keys --query userStorageKey</code>	"ePf ..."

# Exercise: Get Started with a Git Repo

- Launch Terminal
- Clone a git repo – suggest to clone into ~/cloudfiles/code
- Explore project folder structure

```
cd ~/cloudfiles/code  
git clone https://github.com/mutazag/catsanddogs
```

Code Repo: <https://github.com/mutazag/catsanddogs>



# Datasets

# Exercise: Download Dataset

- Download the 'Cats and Dogs' dataset to compute instance
  - [https://storage.googleapis.com/mledu-datasets/cats\\_and\\_dogs\\_filtered.zip](https://storage.googleapis.com/mledu-datasets/cats_and_dogs_filtered.zip)
- Unzip the file
- Upload to Azure Storage Account
  - Use default storage account for the workspace
  - Create a storage container in storage account
  - Automate upload using cli for storage (storage-preview)
    - Requires: az extension add --name storage-preview
- Create and Register and Azure ML Dataset
  - Manually create the dataset
- Solution file: 01\_download\_files.sh:

# Script – set variables to your storage account

```
# script requires azure cli extension for storage (storage-preview)
# az extension add --name storage-preview

# set variables:
resource_group=sina
account_name=godzillasinastorage
container_name=datasets

# constants:
zip_filename=cats_and_dogs_filtered.zip
local_download_dir=~/.cloudfiles/data/
data_folder=cats_and_dogs_filtered
full_directory_name=$local_download_dir$data_folder
echo $full_directory_name

# download and unzip"
wget https://storage.googleapis.com/mledu-datasets/cats_and_dogs_filtered.zip -P $local_download_dir

unzip $local_download_dir$zip_filename -d $local_download_dir

chmod 777 $full_directory_name

# upload to azure storage
account_key=$(az storage account keys list -g $resource_group -n $account_name -o tsv --query "[0].{Value:value}")

az storage container create --name $container_name --auth-mode key --account-key $account_key --account-name $account_name -g $resource_group

az storage blob directory upload -c $container_name --auth-mode key --account-key $account_key --account-name $account_name -s $full_directory_name -d . --recursive
```

# Azure ML Dataset

Microsoft Azure Machine Learning Studio

Microsoft > magazureml > Datasets

## Datasets

Registered datasets Dataset monitors (preview)

✓ Unregister success: Successfully unregistered cats\_dogs\_file:

+ Create dataset ▾ Refresh Unregister

- From local files
- From datastore
- From web files
- From Open Datasets

bike-rentals-test

diabetes

Create dataset from datastore

Basic info

Name \* cats\_dogs \*

Dataset type File ▾

Description Dataset description

\* Choose "File" as dataset type

# Create Dataset

- Select the data store
  - This is the storage account where you previously uploaded the downloaded cats and dogs data set
  - Create a data store in Azure ML if you don't have one yet
- Select the path of the data set in the data store
  - Ensure that path points to the location of the uploaded files
  - Select to include files in subfolders

Create dataset from datastore

Basic info

Datastore selection

Confirm details

Datastore selection

Select or create a datastore \*

shared\_datastore

shared\_datastore  
Azure Blob Storage  
godzillasinastorage

azureml  
Azure Blob Storage  
magazureml9260696727

cer\_images  
Azure Blob Storage  
godzillasinastorage

test\_gen3  
Azure Data Lake Storage Gen2  
magazuremlgen2

ge2store2  
Azure Data Lake Storage Gen2  
magazuremlgen2

test\_gen2  
Azure Data Lake Storage Gen2  
magazuremlgen2

ge2store  
Azure Data Lake Storage Gen2  
magazuremlgen2

Path selection

Selected path: cats\_and\_dogs\_filtered/\*\*

Include files in subfolders

Filter (case-sensitive and prefix-only)...

Name	Created on	Modified on
train/	--	--
validation/	--	--
vectorize.py	Apr 13, 2022 2:22 PM	Apr 13, 2022 2:22 PM

Save Cancel

# Exercise: Consume Dataset

## File Datasets

- Use Dataset object to mount and start a mount point on a compute instance

```
ds_mount = ds.mount(mount_point)
ds_mount.start()
```

- This option presents the dataset as a mounted drive in Linux
- Solution file:  
02\_explore\_dataset.ipynb

## Tabular Datasets

- Tabular datasets are created from structure data files, e.g.: csv or parquet
- A workspace dataset can be consumed as a pandas dataframe in code

```
df = ds.to_pandas_dataframe()
df.head()
```
- Other options include spark or dask dataframes

Development Environment

# Development Environment Options

Environment	Pros	Cons
Local Environment	<ul style="list-style-type: none"><li>- Full control of your development environment and dependencies</li><li>- Run with any build tool, environment or IDE of your choice</li></ul>	<ul style="list-style-type: none"><li>- Takes longer to setup and get started</li><li>- Require installing SDKs and tools</li><li>- Compute and storage limits</li></ul>
Remote on Azure ML Compute Instance	<ul style="list-style-type: none"><li>- Easy to get started, compute instance preconfigured with tools and libraries</li><li>- AML SDK notebooks and tutorials preloaded</li><li>- Scale compute and storage</li></ul>	<ul style="list-style-type: none"><li>- Manage cost for compute instance</li><li>- More complex development environment setup</li></ul>



# Development with VS Code

- Use VS Code and Azure ML Extension
- Write and debug code locally or execute remotely on a compute instance
- Run Jupyter notebooks from within VS Code against a local or remote Jupyter server
- Access AML workspace artefacts



The image shows two screenshots. The top screenshot is the Visual Studio Code download page, which includes the title 'Download Visual Studio Code', a subtitle 'Free and built on open source. Integrated Git, debugging and extensions.', and three main sections for Windows, Linux, and Mac. Each section lists available download formats and architectures. The bottom screenshot shows the Azure Machine Learning extension interface in VS Code, displaying the extension's name, version (v0.8.1), a 'Preview' badge, the Microsoft logo, download count (409,132), and star rating (5 stars, 12 reviews). It also shows buttons for 'Disable', 'Uninstall', and a settings gear, along with the text 'This extension is enabled globally.'

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

**Windows**  
Windows 7, 8, 10, 11

↓ .deb  
Debian, Ubuntu

↓ .rpm  
Red Hat, Fedora, SUSE

↓ Mac  
macOS 10.11+

User Installer 64 bit 32 bit ARM  
System Installer 64 bit 32 bit ARM  
.zip 64 bit 32 bit ARM

.deb 64 bit ARM ARM 64  
.rpm 64 bit ARM ARM 64  
.tar.gz 64 bit ARM ARM 64  
Snap Store

.zip Universal Intel Chip Apple Silicon

**Azure Machine Learning** v0.8.1 Preview

Microsoft | 409,132 | ★★★★★ (12)

Visual Studio Code extension for Azure Machine Learning

Disable Uninstall ⚙️

This extension is enabled globally.

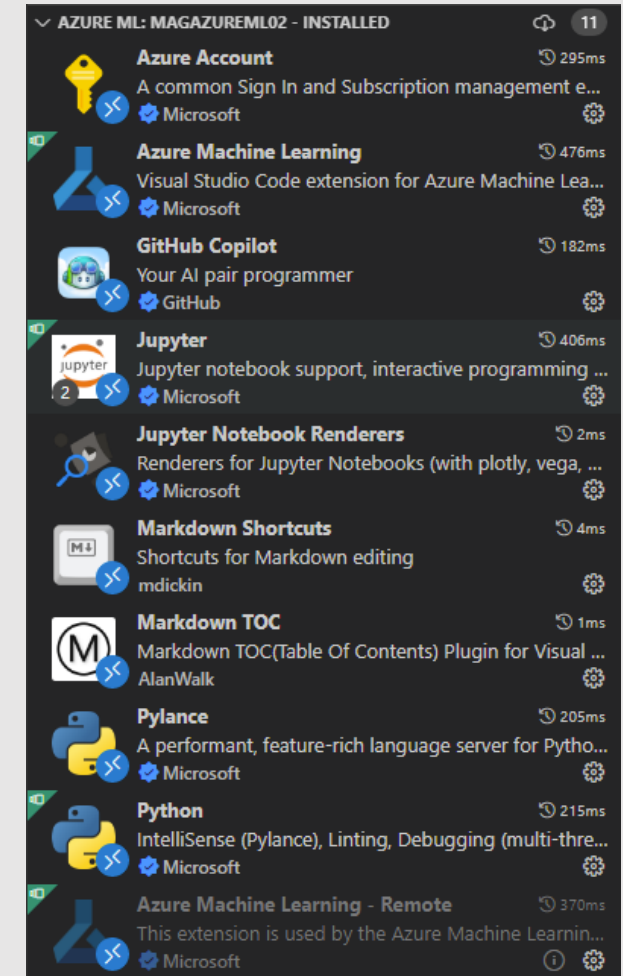
# Install VS Code

Access to AML workspace and compute instance

- Install VS Code: [Download Visual Studio Code - Mac, Linux, Windows](#)
- Python: [Download Python | Python.org](#)

VS Code Extensions:

- Python
- Azure Machine Learning
- Azure Account
- Jupyter



Overview

SETUP

GET STARTED

USER GUIDE

LANGUAGES

NODE.JS /  
JAVASCRIPT

TYPESCRIPT

PYTHON

JAVA

C++

CONTAINERS

DATA SCIENCE

Overview

Jupyter Notebooks

Data Science Tutorial

Python Interactive

PyTorch Support

Azure Machine  
Learning

REMOTE

# Azure Machine Learning in VS Code

Edit

Azure Machine Learning is a cloud-based environment you can use to train, deploy, automate, manage, and track machine learning models. For more information on Azure Machine Learning, see [What is Azure Machine Learning?](#)

The [Azure Machine Learning](#) VS Code extension lets you use the features you're used to in Visual Studio Code for developing your machine learning applications.

Extension: Azure Machine Learning - Visual Studio Code

Extension: Azure Machine Learning

Azure Machine Learning v0.6.27 Preview

Microsoft | 294,002 | ★★★★★ (12)

Visual Studio Code extension for Azure Machine Learning

Disable Uninstall

This extension is enabled globally.

Details Feature Contributions Changelog Dependencies Runtime Status

## Azure Machine Learning for Visual Studio Code

With the **Azure Machine Learning for Visual Studio Code** extension you can easily build, train, and deploy machine learning models to the cloud or the edge with [Azure Machine Learning service](#) from the Visual Studio Code interface. Earlier versions of this extension were released under the name *Visual Studio Code Tools for AI*.

With Azure Machine Learning service, you can:

- Build and train machine learning models faster, and easily deploy to the cloud or the edge.
- Use the latest open source technologies such as [TensorFlow](#), [PyTorch](#), or [Jupyter](#).
- Experiment locally and then quickly scale up or out with large GPU-enabled clusters in the cloud.
- Speed up data science with automated machine learning and hyper-parameter tuning.

With this extension installed, you can accomplish much of this workflow directly from Visual Studio Code.

### Supported Operating Systems

Currently this extension supports the following 64-bit operating systems:

## IN THIS ARTICLE

Connect to remote compute instances

Azure Machine Learning 2.0 CLI support (preview)

Train machine learning models

Manage resources

Remote Jupyter servers

Git integration

Next steps

Tweet this link

Subscribe

Ask questions

Follow @code

Request features

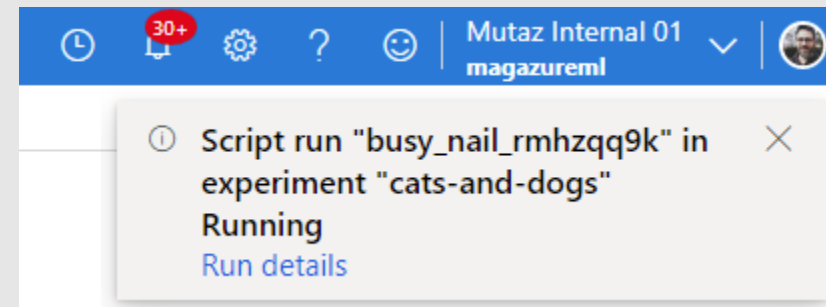
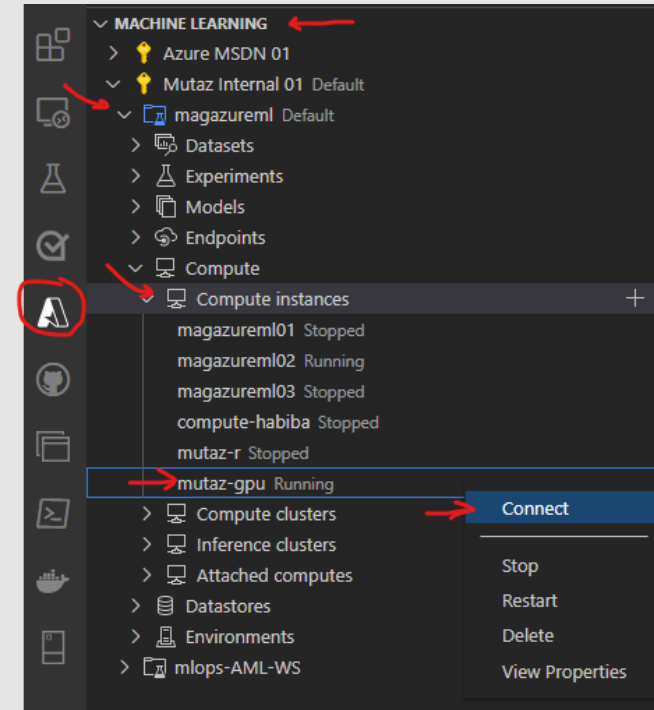
Report issues

Watch videos

<https://code.visualstudio.com/docs/datascience/azure-machine-learning>

# Get Started with Compute Instance

- Connect VS Code to compute instance
- Start a new terminal window in VS Code
- Open project folder  
(~/cloudfiles/code/catsanddogs)
- What is the name of your training experiment run in Azure ML Workspace?
- Run model training notebook:  
03\_train\_model.ipynb



# Experiments and Tracking

# MLFlow for Tracking Experiments

MLflow is an open-source platform for managing the end-to-end machine learning lifecycle.

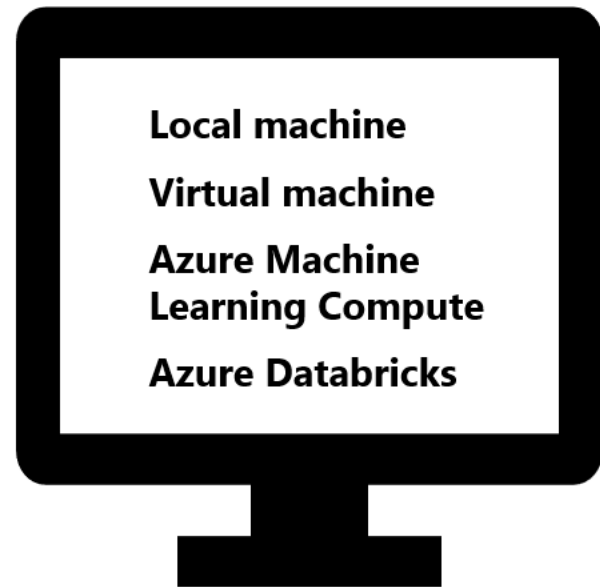
MLflow tackles key functions of ML workloads. MLFlow can be used to track experiments to record and compare parameters and results.

MLflow is library-agnostic. You can use it with any machine learning library, and in any programming language, since all functions are accessible through a REST API and CLI. For convenience, the project also includes a Python API, R API, and Java API.

# MLFlow and Azure ML

MLflow with Azure Machine Learning Experimentation

## Experiments



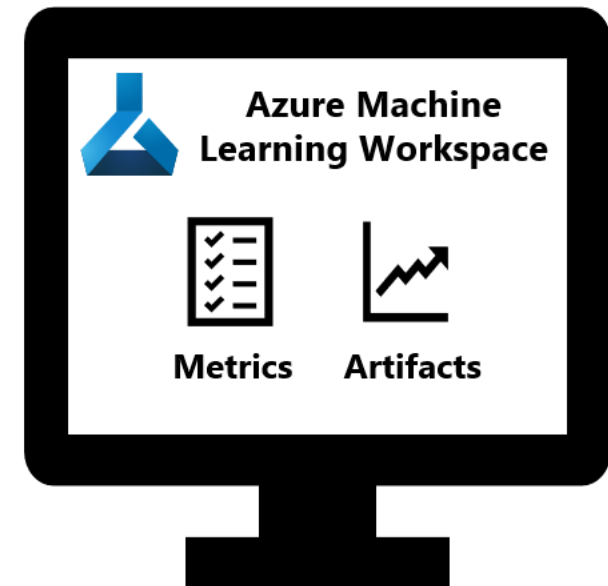
Experiments and Metrics Logging



**mlflow**

Logging API  
Tracking URI

## Experiments and Metrics Tracking



# Logging to Experiments

- Mlflow library is already installed when running in compute instances
- Automatic logging allows you to log metrics, parameters, and models without the need for explicit log statements.
- Call `mlflow.autolog()` before your training code. Enables auto logging for each supported library as soon as it is imported.
- [Automatic Logging](#) supports most popular ML libraries, including: TensorFlow and Keras, Scikit-learn, others.
- Use mlflow API for custom/explicit logging



# Auto logging for keras

- Metrics:
  - Training loss, validation loss, early stopping metrics
- Params:
  - fit() and fit\_generator() params
  - Learning rate, epsilon, optimizer name
- Artifacts
  - Model summary on training start
  - MLFlow model on training end
  - TensorBoard logs
- <https://mlflow.org/docs/latest/tracking.html#tensorflow-and-keras>

Tags	
input_dataset : /mnt/batch/tasks/shared/LS_root/jobs/magazureml/azureml/cats-and-dogs_1650697803_92a75656/wd/input_41e09c7c_41e09c7c-adaf-4040-89af-aaa3d2a3f33c	
logging : mlflow.autolog()	mlflow.source.name : train.py
mlflow.source.type : JOB	
Params	
baseline : None	batch_size : None
class_weight : None	epochs : 50
initial_epoch : 0	max_queue_size : 10
min_delta : 0	monitor : loss
opt_amsgrad : False	opt_beta_1 : 0.9
opt_beta_2 : 0.999	opt_decay : 0.0
opt_epsilon : 1e-07	opt_learning_rate : 1e-05
opt_name : Adam	patience : 5
restore_best_weights : False	sample_weight : None
shuffle : True	steps_per_epoch : 100
use_multiprocessing : False	validation_batch_size : None
validation_freq : 1	validation_split : 0.0
validation_steps : 50	workers : 1
Metrics	
accuracy	stopped_epoch
Min: 0.732, Max: 0.95, Last: 0.948	22
loss	val_accuracy
Min: 0.138, Max: 0.534, Last: 0.142	Min: 0.787, Max: 0.906, Last: 0.897
lr	val_loss
Min: 1.0000e-7, Max: 1.0000e-5, Last: 1.0000e-7	Min: 0.239, Max: 0.433, Last: 0.258

# Logging data to Runs using MLFlow

API	Description	Example
<code>mlflow.log_metric</code>  <code>mlflow.log_metrics</code>	Logs a metric for a run. Metrics are key-value pair that records a single float measure. A metric can be logged several times. The MLflow Backend keeps track of historical metric values.	<code>mlflow.log_metric("mse", 250.0)</code>  <code>metrics = {"mse": 250.0, "rmse": 50.0}</code> <code>mlflow.log_metrics(metrics)</code>
<code>mlflow.log_param</code>  <code>mlflow.log_params</code>	Logs a parameter for a run. Examples are params and hyperparams used for ML training. A param is a key-value pair. For a run, a single parameter is allowed to be logged only once.	<code>mlflow.log_param("learning_rate", 0.01)</code>  <code>params = {"learning_rate": 0.01, "n_estimators": 10}</code> <code>mlflow.log_params(params)</code>
<code>mlflow.set_tag</code>  <code>mlflow.set_tags</code>	Sets a tag on a run. Tags are run metadata that can be updated during a run and after a run completes.	<code>mlflow.set_tag("release.version", "2.2.0")</code>

MLflow logging functions: <https://mlflow.org/docs/latest/tracking.html#logging-functions>

# Loading Model Artefacts from Run

- Retrieve run details from Azure ML:

```
ws = Workspace.from_config()
exp = Experiment(ws, exp_name)
run = Run(exp, run_name)
```

- Access trained model:

```
run.download_files('model')
```

- Code file: `explore_run.ipynb`

- Access run metrics:

```
run.get_metrics().keys()
max(run.get_metrics()['val_accuracy'])
```

# Model Training

**Watching  
a model train**



**Watching  
a model train**



# Executing Model Training

- Build and debug code
- Use tools such as VS Code or Notebooks
- Local development or remote on Compute Instances

## Interactive Development



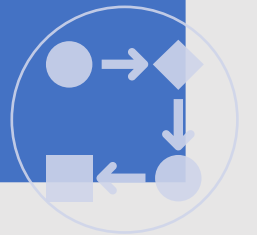
- Submit a python script as job for long running / unattended execution
- examine logs and results after completion
- Target execution to compute instances or clusters

## Submit Jobs



- Multiple steps, e.g.: data prep, feature engineering, modeling and validation
- Reusable and scalable when building MLOPS
- Execute on variety of compute options: CI, cluster, Spark

## ML Pipelines



You could be running on remote compute in any of the three model training approaches

# Training on Remote Compute

## Execution Environment

- Containerisation is utilised heavily to ensure environment requirements and dependencies are present for job execution
- Select from preconfigured environment, or create a custom env specification (including all python libraries or other programs)
  - AzureML-tensorflow-2.4-ubuntu18.04-py37-cuda11-gpu
- Training script and other code dependencies must be packaged and specified in the job configuration

## Data Access

- Data can be presented as mount points on the file system of the training docker container
- Training script must accept command line arguments, argparse is a great utility to work with python arguments
- When job is setup, file datasets are mounted and path to mounted file is passed in to the training script as a command line argument

# Exercise: Training Script

Refactor training script:

- Training script: refactor code to run in a training python script
- Utilise arguments, usually using argparse in python, to control parameters of a training job
  - File path to input data set is passed in as a command line argument
  - Other parameters, like training parameters could also be parameterised this way
- Solution file: `src/train.py`



# Exercise: Submit Training Job

## Job configuration process:

- Specify target Compute: compute instance or cluster
  - Execution environment: select from curated env with required dependencies
  - Data access: load dataset as mount point and pass the path in command line argument to the training script
  - Submit run: set experiment name and submit run configuration
- 
- Solution File: 05\_submit\_job.ipynb

yellow\_hat\_4hqswpm

 Refresh  Connect to compute  Resubmit  Cancel  Delete

[Home](#)

[Details](#)
[Metrics](#)
[Images](#)
[Child runs](#)
[Outputs + logs](#)
[Snapshot](#)
[Explanations \(preview\)](#)
[Fairness \(preview\)](#)
[Monitoring \(preview\)](#)

Properties

 **Status**  
Completed



**Created**  
Apr 23, 2022 5:10 PM

Started  
Apr 23, 2022 5:12 PM

Duration  
10m 55.58s

Compute duration  
10m 55.58s

Run ID
cats-and-dogs_1650697803_92a75656

	Script name
	train.py

Created by  
Mutaz Abu Ghazaleh

**Input datasets**  
Input name: input\_41e09c7c, Dataset: cats\_dogs:1

Status	Output datasets
Completed	None

	<b>Created</b> Apr 23, 2022 5:10 PM	<b>Environment</b> AzureML-tensorflow-2.4-ubuntu18.04-py37-cuda11-gpu:34
---	--	---

Started	Arguments
Apr 23, 2022 5:12 PM	--input_dataset
Duration	DatasetConsumptionConfig:input__41e09c7c

	10m 55.58s	
	Compute duration	Registered models None

10m 55.58s

Run ID [Git repository  
https://github.com/mutazag/azureml-samples](https://github.com/mutazag/azureml-samples)

cats-and-dogs_1650697803_92a75656	Git branch
Script name	main

train.py  
Created by [42f4b762492d8afe3788b2477974a325d87840b](#) **Git commit**

 Mutaz Abu Ghazaleh [See all properties](#)

Input name: input\_41e09c7c, Dataset: cats\_dogs:1

See [YAML Job definition](#)

 [Job YAML](#)

## Compute

	Target gpucluster
--	----------------------

Compute type  
amlcompute

Target	Instance count
gpucluster	1

**Properties**

Status	Output datasets
 Completed	None

Created	Environment	logging : mlflow.autolog()	mlflow.source.name : train.py	mlflow.source.type : JOB
Apr 23, 2022 5:10 PM	AzureML-tensorflow-2.4-ubuntu18.04-py37-cuda11-			

Apr 23, 2022 5:12 PM Arguments  
--input dataset Params

Duration 10m 55.58s DatasetConsumptionConfig:input\_\_41e09c7c baseline : None batch\_size : None class\_weight : None epochs : 50 initial\_epoch : 0 max\_queue\_size : 10

Compute duration	Registered models	min_delta : 0	monitor : loss	opt_amsgrad : False	opt_beta_1 : 0.9	opt_beta_2 : 0.999	opt_decay : 0.0
10 : 55.59	None						

10m 55.58s

Git repository

opt\_epsilon : 1e-07   opt\_learning\_rate : 1e-05   opt\_name : Adam   patience : 5   restore\_best\_weights : False

Run ID	<a href="https://github.com/mutazag/azureml-samples">https://github.com/mutazag/azureml-samples</a>	sample_weight : None	shuffle : True	steps_per_epoch : 100	use_multiprocessing : False	validation_batch_size : None
cats-and-dogs_1650697803_92a75656						

Script name	Git branch	validation_freq : 1	validation_split : 0.0	validation_steps : 50	workers : 1
main					

Created by	Git commit	Metrics
Mutaz Abu Ghazaleh	42f4bc762492d8afe3788b2477974a325d87840b	

Input datasets [See all properties](#)  
Input name: input\_41e09c7c; Dataset: [cats\\_dogs:1](#) [Raw JSON](#) [accuracy](#)  
Min: 0.732, Max: 0.95, Last: 0.948

loss	See YAML job definition	loss Min: 0.138, Max: 0.534, Last: 0.142
------	-------------------------	---

Job YAML

<b>Input datasets</b> Input name: input_41e09c7c, Dataset: <a href="#">cats_dogs:1</a>	<a href="#">See all properties</a>  <a href="#">Raw JSON</a>	<b>accuracy</b> Min: 0.732, Max: 0.95, Last: 0.948	<b>stopped_epoch</b> 22
---	--	---	----------------------------

See YAML job definition	loss Min: 0.138, Max: 0.534, Last: 0.142	val_accuracy Min: 0.787, Max: 0.906, Last: 0.897
-------------------------	---	---

Job YAML

lr  
Min: 1.0000e-7 Max: 1.0000e-5 Last: 1.0000e-7

val\_loss  
Min: 0.239 Max: 0.433 Last: 0.258

Target	Instance count	Description
--------	----------------	-------------

Cluster Name	Nodes	GPUs	Memory	Status	Created At	Updated At	Deleted At	Actions
gpucluster	1	1	16GB	Running	2023-10-27 10:10:10	2023-10-27 10:10:10		<div> <div></div> <div>Click edit icon to add a description</div> </div>