**Azure Machine Learning (AML)**

Azure Machine Learning is a cloud service for accelerating and managing the machine learning project lifecycle. Machine learning professionals, data scientists, and engineers can use it in their day-to-day workflows: Train and deploy models, and manage MLOps.

You can create a model in Azure Machine Learning or use a model built from an open-source platform, such as Pytorch, TensorFlow, or scikit-learn. MLOps tools help you monitor, retrain, and redeploy models.

At this stage, you will not fully use the functionality of AML. You will only use it to create a virtual machine for each user (called a compute instance), access it and use it in the traditional way via *JupyterLab* or *VSCode*.

The ultimate goal is to gradually transition to using and leveraging all of its features, including storing and managing data and models, as well as preparing training code for production. and to track machine learning experiments using AML Pipelines.

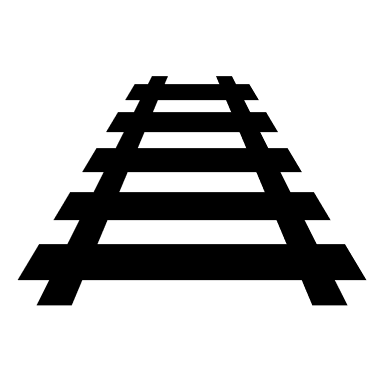
**Workspace**

The workspace is the top-level resource for Azure Machine Learning, providing a centralized place to work with all the artifacts you create while using Azure Machine Learning.

**Tools for workspace interaction**

You can interact with your workspace in the following ways:

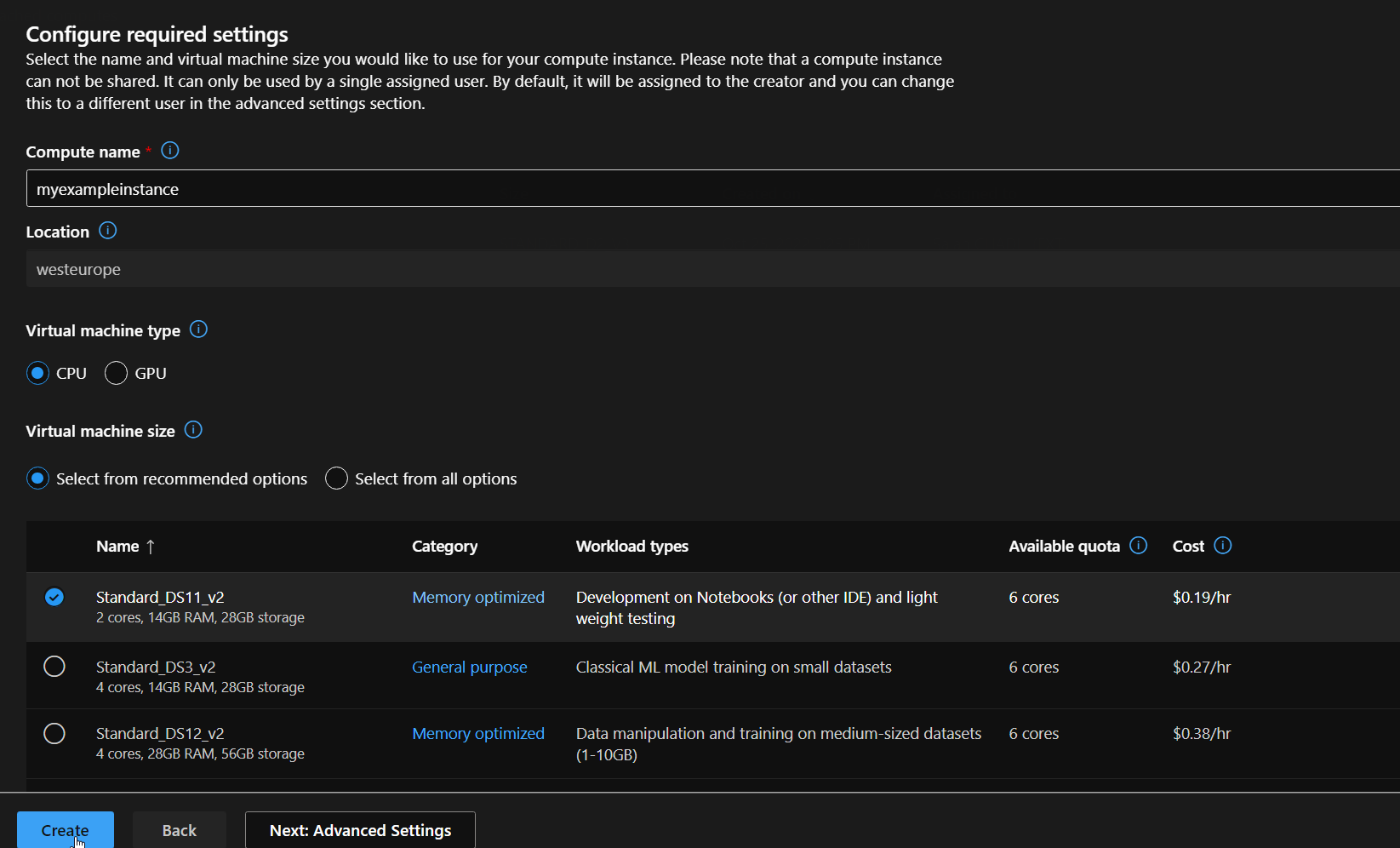
* On the web:
  + Azure Machine Learning studio
  + Azure Machine Learning designer
* In any Python environment with the Azure Machine Learning SDK for Python.
* On the command line using the Azure Machine Learning CLI extension
* Azure Machine Learning VS Code Extension



At this point, the only artifacts and components that will be discussed are related to managing data and compute instances. And the only tools of workspace interaction used will be JupyterLab and the VSCode Extension.

In the target, all artifacts related to the machine learning project workflow will also be managed here. The workspace will keep a history of all training runs, including logs, metrics, output, snapshot of your scripts. You will be able to use this information to determine which training run produces the best model and deploy it.

**Compute instance**



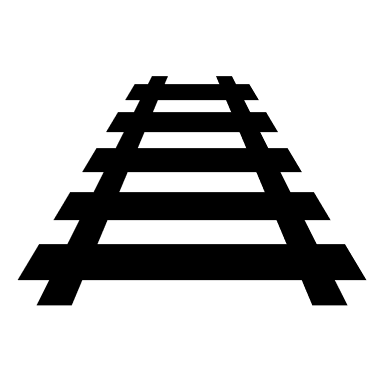
An Azure Machine Learning compute instance is a managed cloud-based workstation for data scientists.

Each compute instance has only one owner, although you can share files between multiple compute instances.

You can create a compute instance yourself, or an administrator can create a compute instance on your behalf.

To save on costs, create a schedule to automatically start and stop the compute instance.

|  |  |
| --- | --- |
| Key benefits | Description |
| Productivity | You can build and deploy models using integrated notebooks and the following tools in Azure Machine Learning studio: - Jupyter - JupyterLab - VS Code (preview) Compute instance is fully integrated with Azure Machine Learning workspace and studio.  You can share notebooks and data with other data scientists in the workspace. |
| Managed & secure | Reduce your security footprint and add compliance with enterprise security requirements.  Compute instances provide robust management policies and secure networking configurations. |
| Preconfigured for ML | Save time on setup tasks with pre-configured and up-to-date ML packages, deep learning frameworks, GPU drivers. |
| Fully customizable | Broad support for Azure VM types including GPUs and persisted low-level customization such as installing packages and drivers makes advanced scenarios a breeze. You can also use setup scripts to automate customization |



At this point, use a compute instance as your fully configured and managed development environment in the cloud for machine learning, as well for training and inferencing for development and testing purposes.

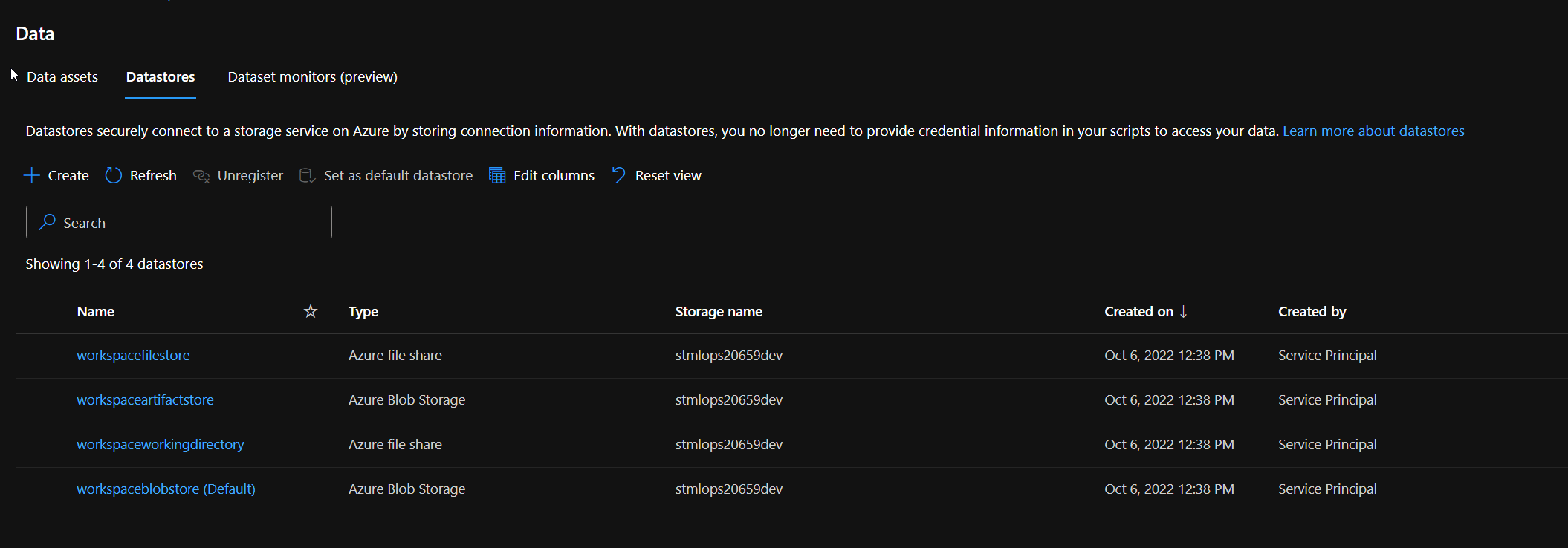
In the target, compute instances will only be used for development and testing purposed. Compute clusters, another types of computing components, will be used as a computing target.

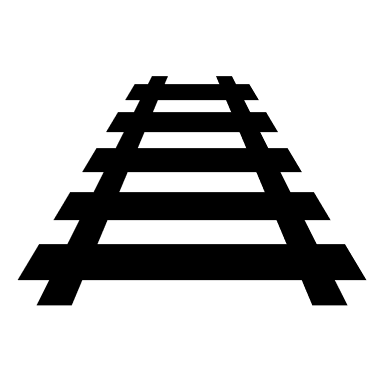
**Data**

Azure Machine Learning lets you bring data from a local machine or an existing cloud-based storage using **Datastores.**

**Datastore**

Azure Machine Learning Datastores securely keep the connection information (storage container name, credentials) to your data storage on Azure, so you don't have to code it in your scripts. You can use AzureML datastore uri and relative path to your data to point to your data. You can also register files/folders in your AzureML datastore into data assets.





At this point, you will only use datastores to access data in Azure.

In the target, other Azure features and components can be used to handle and manage data (see. [Data in Azure Machine Learning](https://learn.microsoft.com/en-us/azure/machine-learning/concept-data))

**Accessing files**

Notebooks and Python scripts are stored in the default storage account of your workspace in Azure file share. These files are located under your “User files” directory. This storage makes it easy to share notebooks between compute instances. The storage account also keeps your notebooks safely preserved when you stop or delete a compute instance.

The Azure file share account of your workspace is mounted as a drive on the compute instance. This drive is the default working directory for Jupyter, Jupyter Labs, and RStudio. This means that the notebooks and other files you create in Jupyter, JupyterLab, or RStudio are automatically stored on the file share and available to use in other compute instances as well.

The files in the file share are accessible from all compute instances in the same workspace. Any changes to these files on the compute instance will be reliably persisted back to the file share.

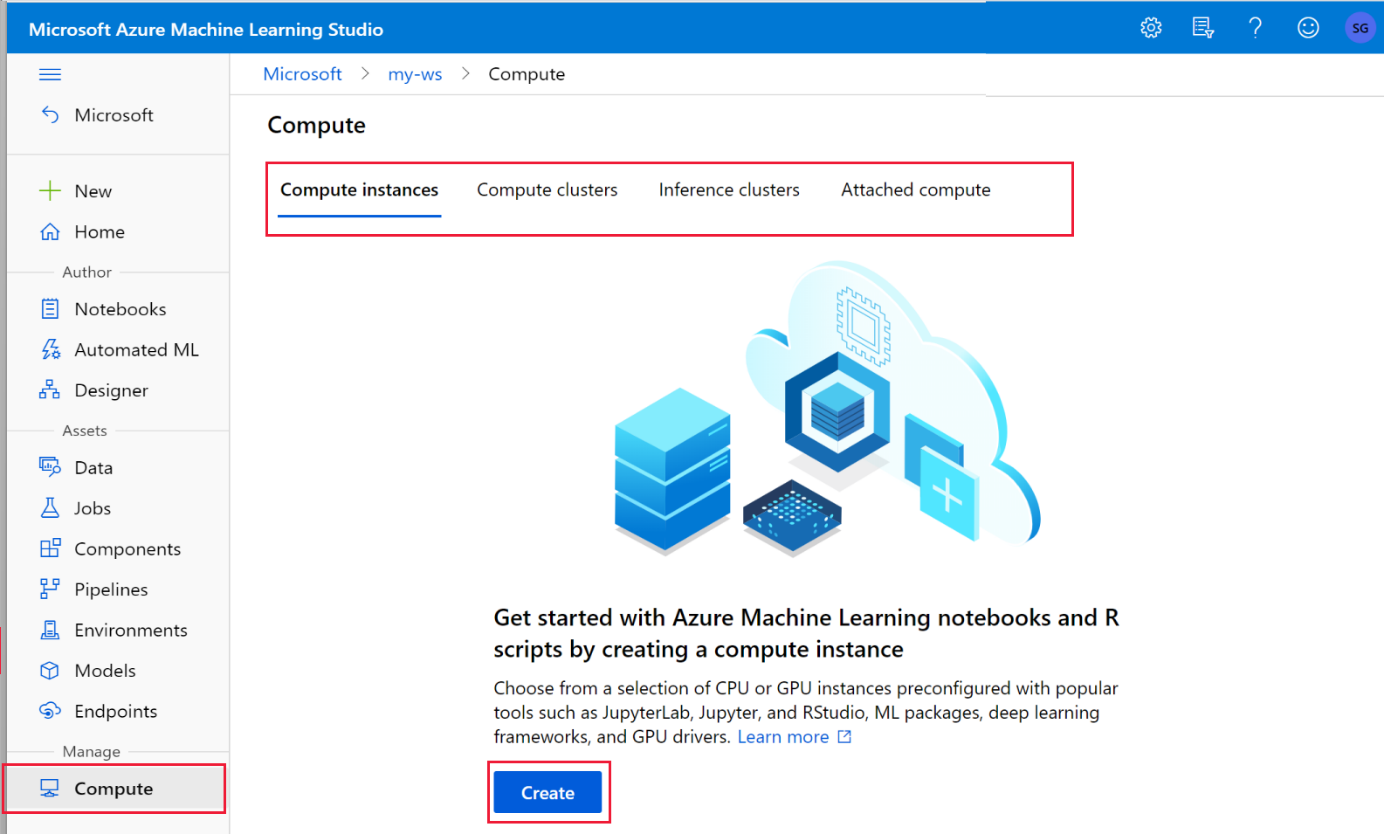
Writing small files can be slower on network drives than writing to the compute instance local disk itself. If you are writing many small files, try using a directory directly on the compute instance, such as a /tmp directory. Note these files will not be accessible from other compute instances.

Do not store training data on the notebooks file share. You can use the /tmp directory on the compute instance for your temporary data. However, do not write very large files of data on the OS disk of the compute instance. OS disk on compute instance has 128 GB capacity. You can also store temporary training data on temporary disk mounted on /mnt. Temporary disk size is configurable based on the VM size chosen and can store larger amounts of data if a higher size VM is chosen. You can also mount datastores and datasets.

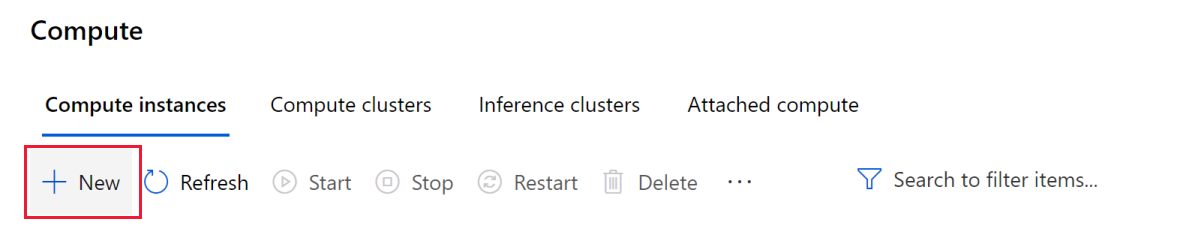
**Quickstart**

**Create a compute instance**

1. Navigate to [Azure Machine Learning studio](https://ml.azure.com/).
2. Under **Manage**, select **Compute**.
3. Select **Compute instance** at the top.
4. If you have no compute instances, select **Create** in the middle of the page.



1. If you see a list of compute resources, select **+New** above the list.



1. Fill out the form:

|  |  |
| --- | --- |
| Field | Description |
| Computer name | * Name is required and must be between 3 to 24 characters long. * Valid characters are upper and lower case letters, digits, and the - character. * Name must start with a letter * Name needs to be unique across all existing computes within an Azure region. You'll see an alert if the name you choose isn't unique * If - character is used, then it needs to be followed by at least one letter later in the name |
| Virtual machine type | Choose CPU or GPU. This type can't be changed after creation |
| Virtual machine size | Supported virtual machine sizes might be restricted in your region. Check the availability list. |

1. Select **Create** unless you want to configure advanced settings for the compute instance (you can add VPN configuration, schedule, etc.)
2. Wait for few minutes until the compute instance is create it, then access it using JupyterLab
3. Configure Git in your instance and clone this Github repository under your user directory in (/Users/$username)

https://github.com/chadlis/mlprojectexample

1. Follow the code structure given in the example to create your project (cf. README.md)
2. The instance should be stopped when not used. A schedule can be defined in the instance advances settings.