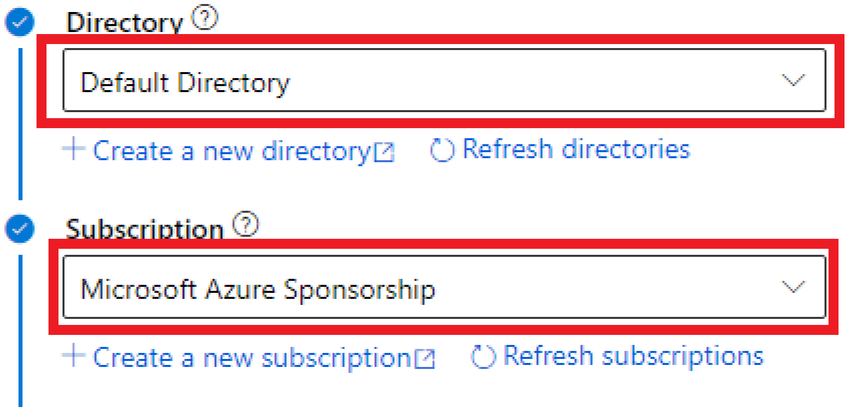
Lab 1A: Creating an Azure Machine Learning Workspace

Task 1: Create an Azure ML Workspace

As its name suggests, a workspace is a centralized place to manage all of the Azure ML assets you need to work on a machine learning project.

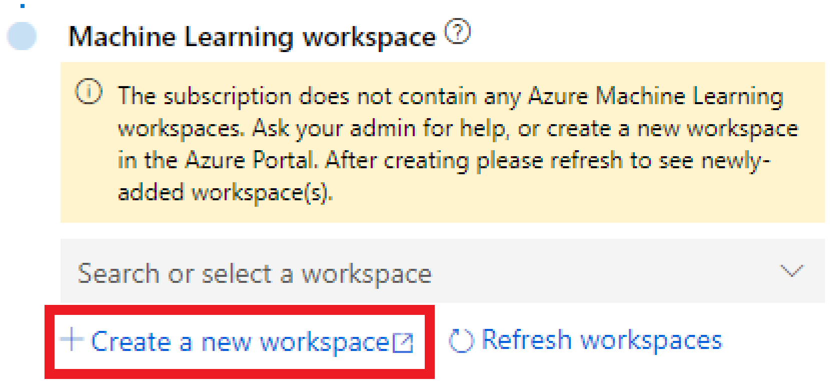
1. Go to https://ml.azure.com, select the **Directory** and **Subscription**



Directory: Default Directory

Subscription: Microsoft Azure Sponsorship

2. In **Machine Learning workspace**, click on **+ Create a New workspace**

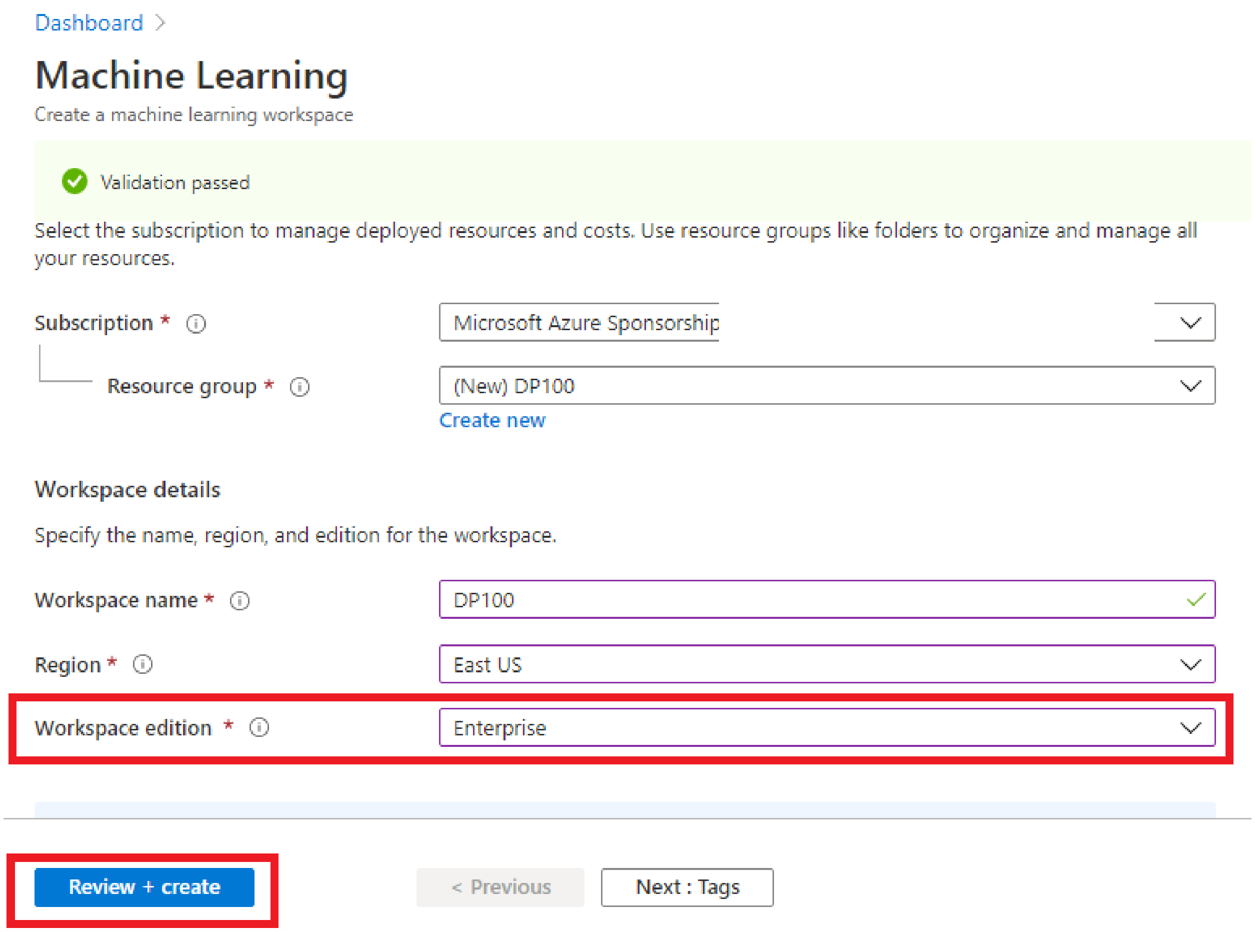


Machine Learning workspace

The subscription does not contain any Azure Machine Learning workspaces. Ask your admin for help, or create a new workspace in the Azure Portal. After creating please refresh to see newly-added workspace(s).

Search or select a workspace

3. In the Azure portal, specifying a unique workspace name and creating a new resource group in the region nearest your location. Select the **Enterprise** workspace edition. Click **Review + create** button.



Machine Learning

Create a machine learning workspace

Validation passed

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Microsoft Azure Sponsorship

Resource group * : (New) DP100

Create new

Workspace details

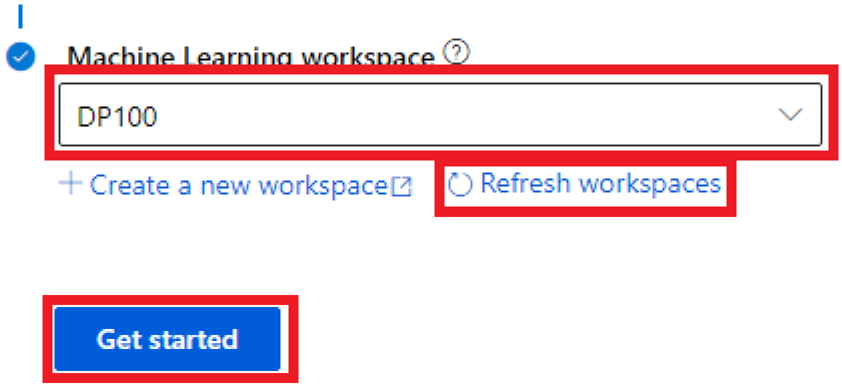
Specify the name, region, and edition for the workspace.

Workspace name * : DP100

Region * : East US

Workspace edition * : Enterprise

4. Click on **Create** button.
5. Back to **Step 1** page, click on Refresh workspaces. You will see the workspace and select that workspace that created just now. Click on Get **Started** button.

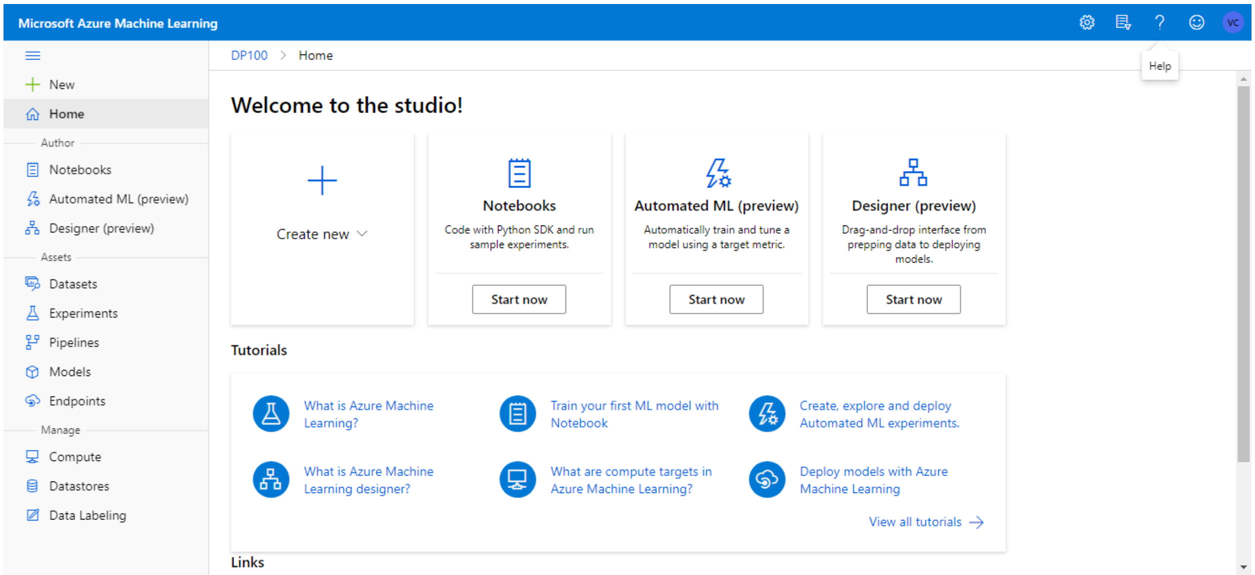


Machine Learning workspace : DP100

Task 2: Explore the Azure ML Studio Interface

You can manage workspace assets in the Azure portal, but for data scientists, this tool contains lots of irrelevant information and links that relate to managing general Azure resources. An alternative, Azure ML-specific web interface for managing workspaces is available.

1. View the Azure Machine Learning studio interface for your workspace - you can manage all of the assets in your workspace from here.



Welcome to the studio!

Notebooks: Code with Python SDK and run sample experiments.

Automated ML (preview): Automatically train and tune a model using a target metric.

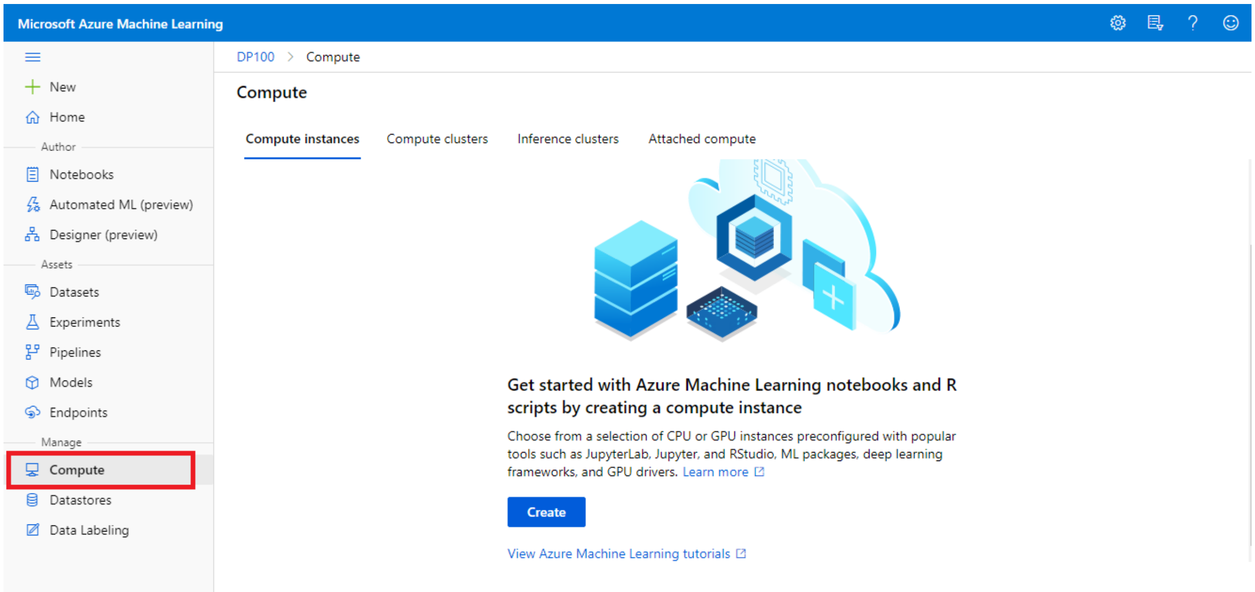
Designer (preview): Drag-and-drop interface from prepping data to deploying models.

Tutorials: What is Azure Machine Learning? · Train your first ML model with Notebook · Create, explore and deploy Automated ML experiments. · What is Azure Machine Learning designer? · What are compute targets in Azure Machine Learning? · Deploy models with Azure Machine Learning

Task 3: Create Compute Resources

One of the benefits of Azure Machine Learning is the ability to create cloud-based compute on which you can run experiments and training scripts at scale.

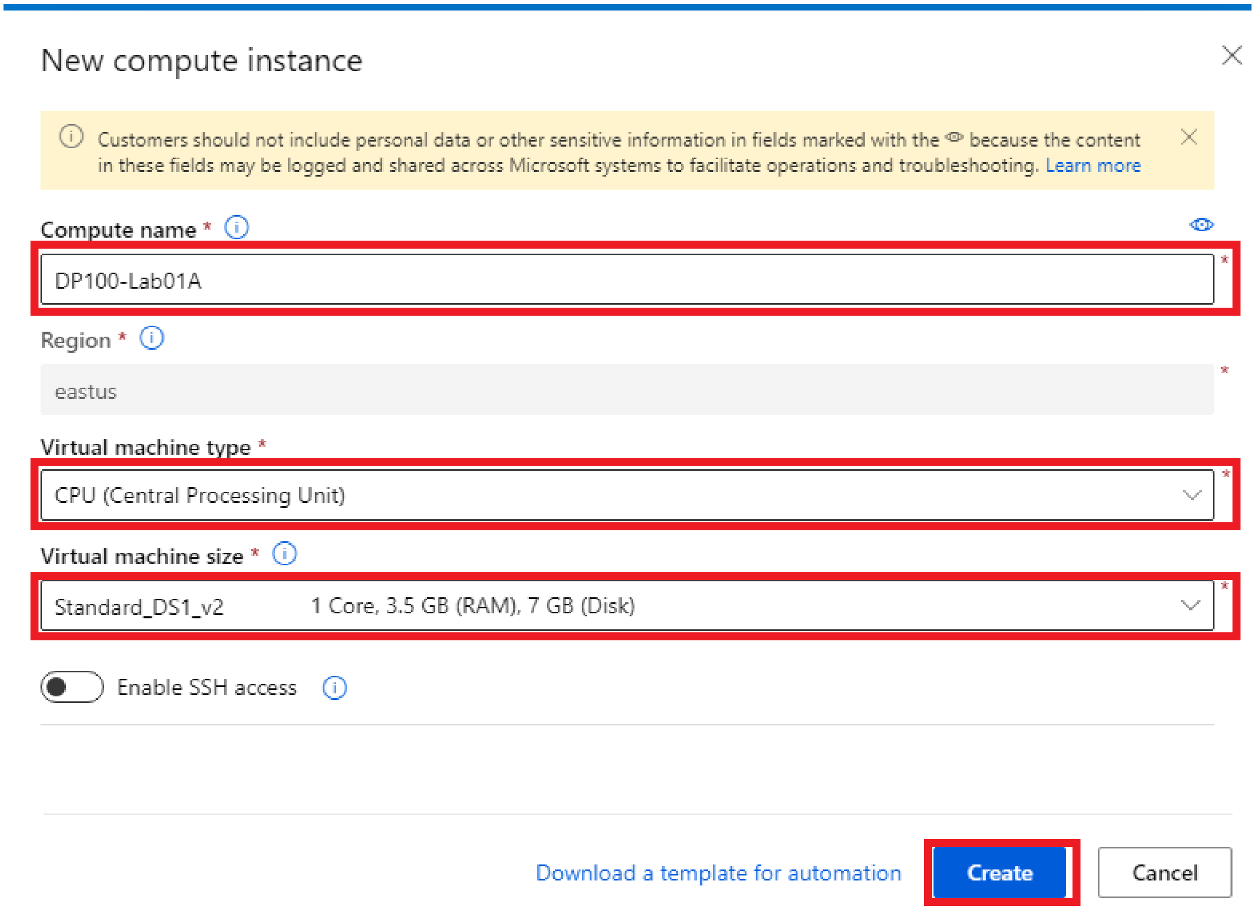
1. In the Azure Machine Learning studio web interface for your workspace, view the **Compute** page. This is where you'll manage all the compute targets for your data science activities.



Get started with Azure Machine Learning notebooks and R scripts by creating a compute instance

Choose from a selection of CPU or GPU instances preconfigured with popular tools such as JupyterLab, Jupyter, and RStudio, ML packages, deep learning frameworks, and GPU drivers. Learn more

2. On the **Compute instances** tab, click on **Create** button to add a new compute instance with the following settings. You'll use this as a workstation from which to test your model:
   - **Compute name**: enter a unique name
   - **Virtual Machine type**: CPU
   - **Virtual Machine size**: Standard_DS1_v2



New compute instance

Customers should not include personal data or other sensitive information in fields marked with the because the content in these fields may be logged and shared across Microsoft systems to facilitate operations and troubleshooting. Learn more

Compute name *: DP100-Lab01A

Region *: eastus

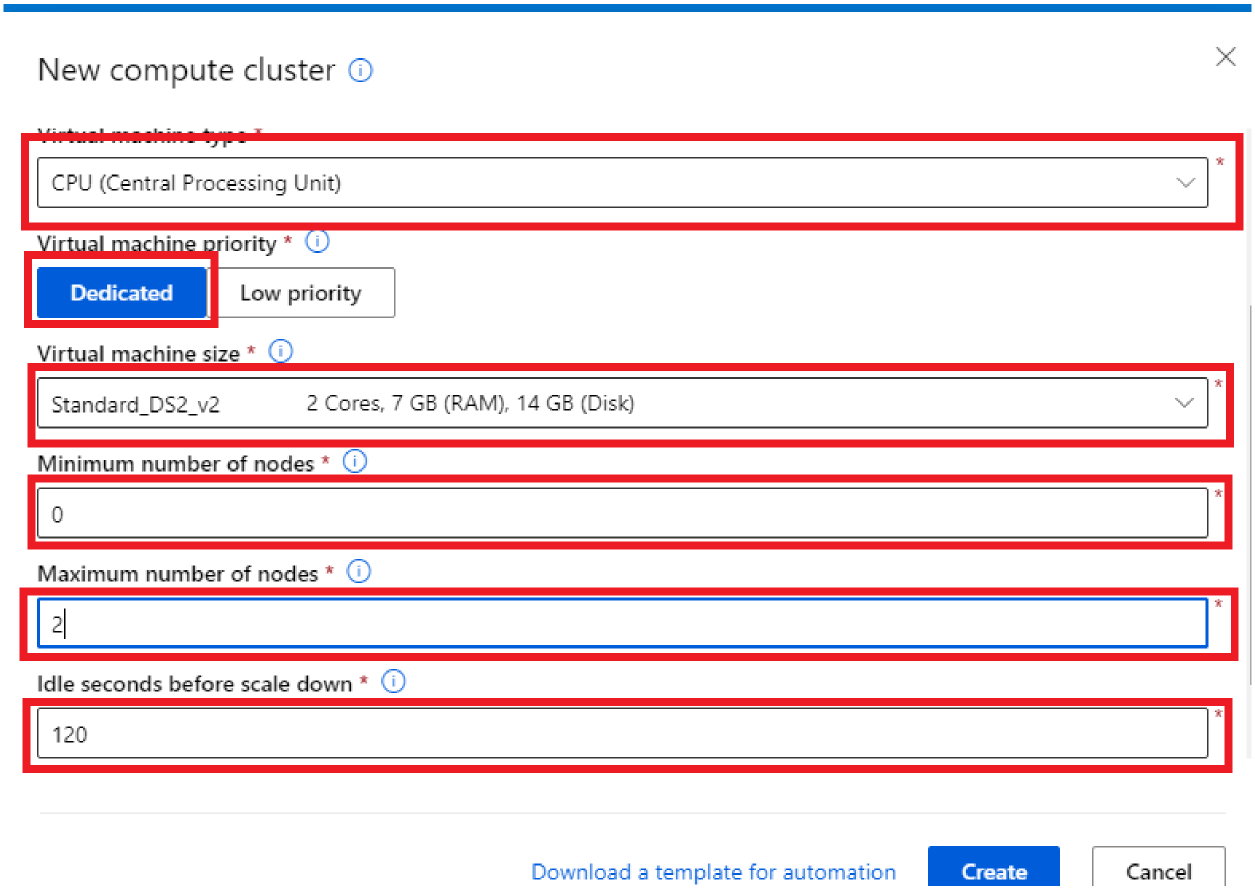
Virtual machine type *: CPU (Central Processing Unit)

Virtual machine size *: Standard_DS1_v2 1 Core, 3.5 GB (RAM), 7 GB (Disk)

Enable SSH access

3. While the compute instance is being created, switch to the **Compute Clusters** tab and add a new compute cluster with the following settings. You'll use this to train a machine learning model:
   - **Compute name**: enter a unique name
   - **Virtual Machine type**: CPU
   - **Virtual Machine priority**: Dedicated
   - **Virtual Machine size**: Standard_DS2_v2
   - **Minimum number of nodes**: 0
   - **Maximum number of nodes**: 2

- **Idle seconds before scale down**: 120



New compute cluster

Virtual machine type *

CPU (Central Processing Unit)

Virtual machine priority *

Dedicated | Low priority

Virtual machine size *

Standard_DS2_v2 2 Cores, 7 GB (RAM), 14 GB (Disk)

Minimum number of nodes *

0

Maximum number of nodes *

2

Idle seconds before scale down *

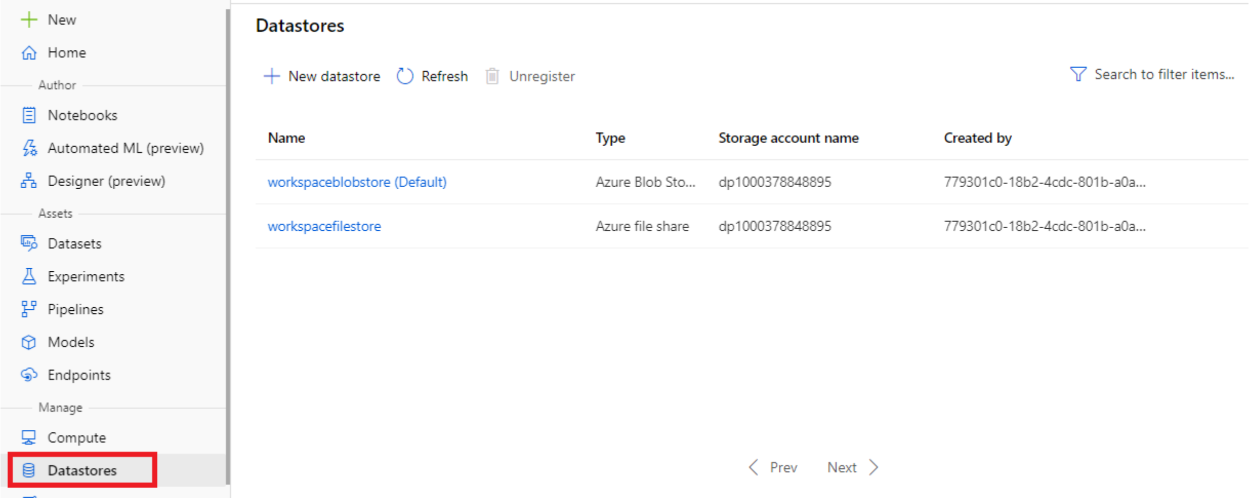
120

4. Note the **Inference Cluster** tab. This is where you can create and manage compute targets on which to deploy your trained models as web service for client applications to consume.
5. Note the **Attached Compute** tab. This is where you could attach a virtual machine or Databricks cluster that exists outside of your workspace.

Task 4: Create Data Resources

Now that you have some compute resources that you can use to process data, you'll need a way to store and ingest the data to be processed.

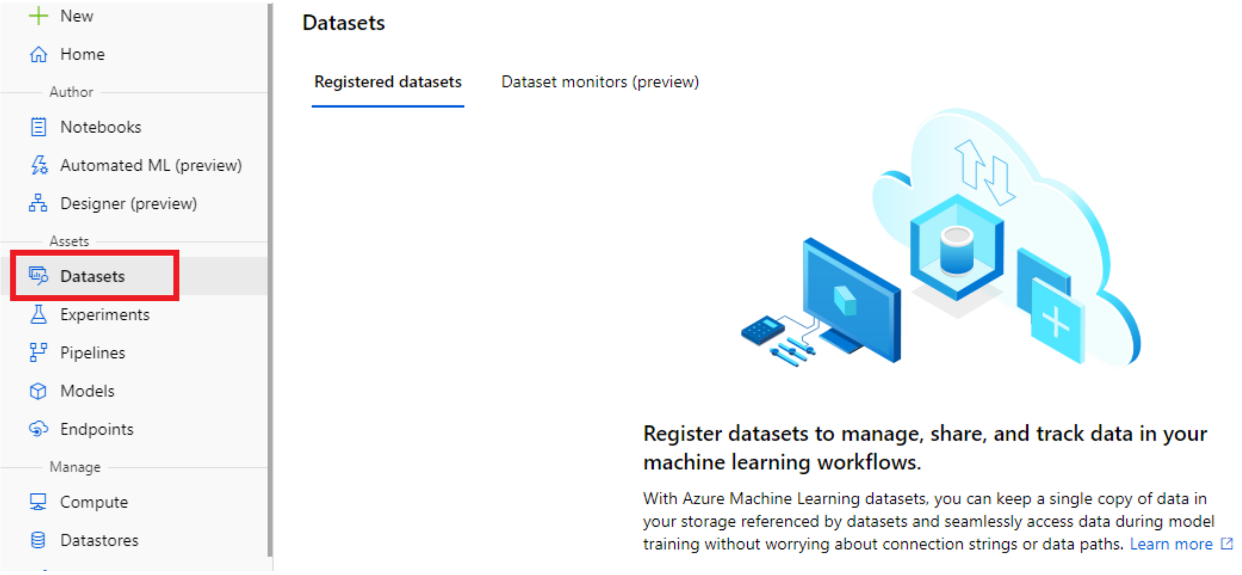
1. In the Studio interface, view the **Datastores** page. Your Azure ML workspace already includes two datastores based on the Azure Storage account that was created along with the workspace. These are used to store notebooks, configuration files, and data. Note: In the real-world environment, you'd likely add custom datastores that reference your business data stores - for example, Azure blob containers, Azure Data Lakes, Azure SQL Databases, and so on.



Datastores

Name	Type	Storage account name	Created by
workspaceblobstore (Default)	Azure Blob Sto...	dp1000378848895	779301c0-18b2-4cdc-801b-a0a...
workspacefilestore	Azure file share	dp1000378848895	779301c0-18b2-4cdc-801b-a0a...

2. In the Studio interface, view the **Datasets** page. Datasets represent specific data files or tables that you plan to work with in Azure ML.

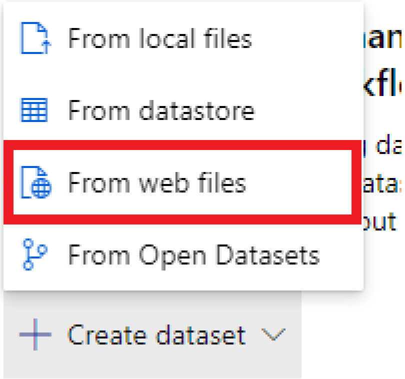


Datasets

Register datasets to manage, share, and track data in your machine learning workflows.

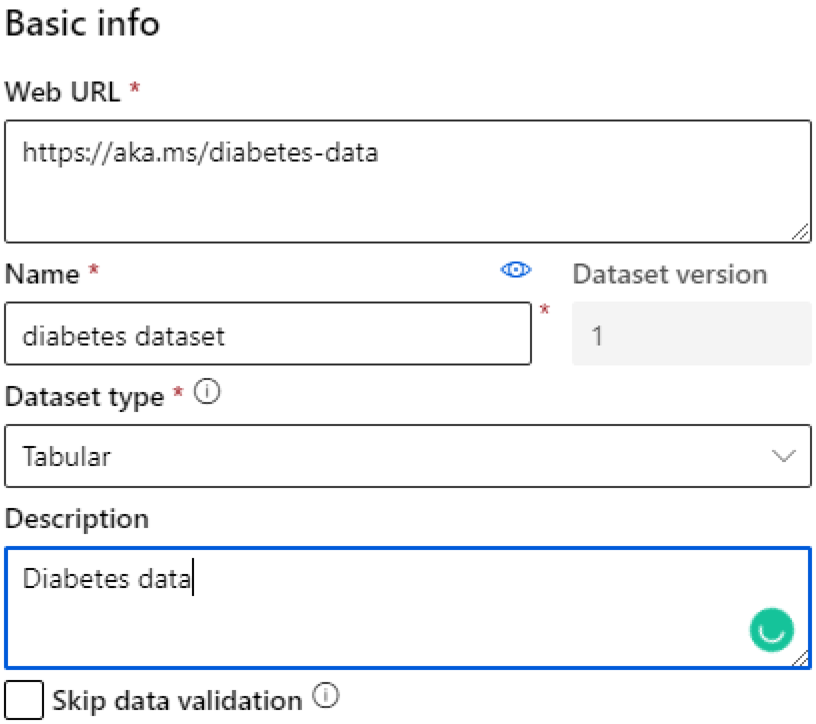
With Azure Machine Learning datasets, you can keep a single copy of data in your storage referenced by datasets and seamlessly access data during model training without worrying about connection strings or data paths. Learn more

3. Create a new dataset **from web files**, using the following settings:



Basic Info:

- **Web URL**: https://aka.ms/diabetes-data
- **Name**: diabetes dataset (be careful to match the case and spacing)
- **Dataset type**: Tabular
- **Description**: Diabetes data



Basic info

Web URL *

https://aka.ms/diabetes-data

Name *

diabetes dataset

Dataset version

1

Dataset type *

Tabular

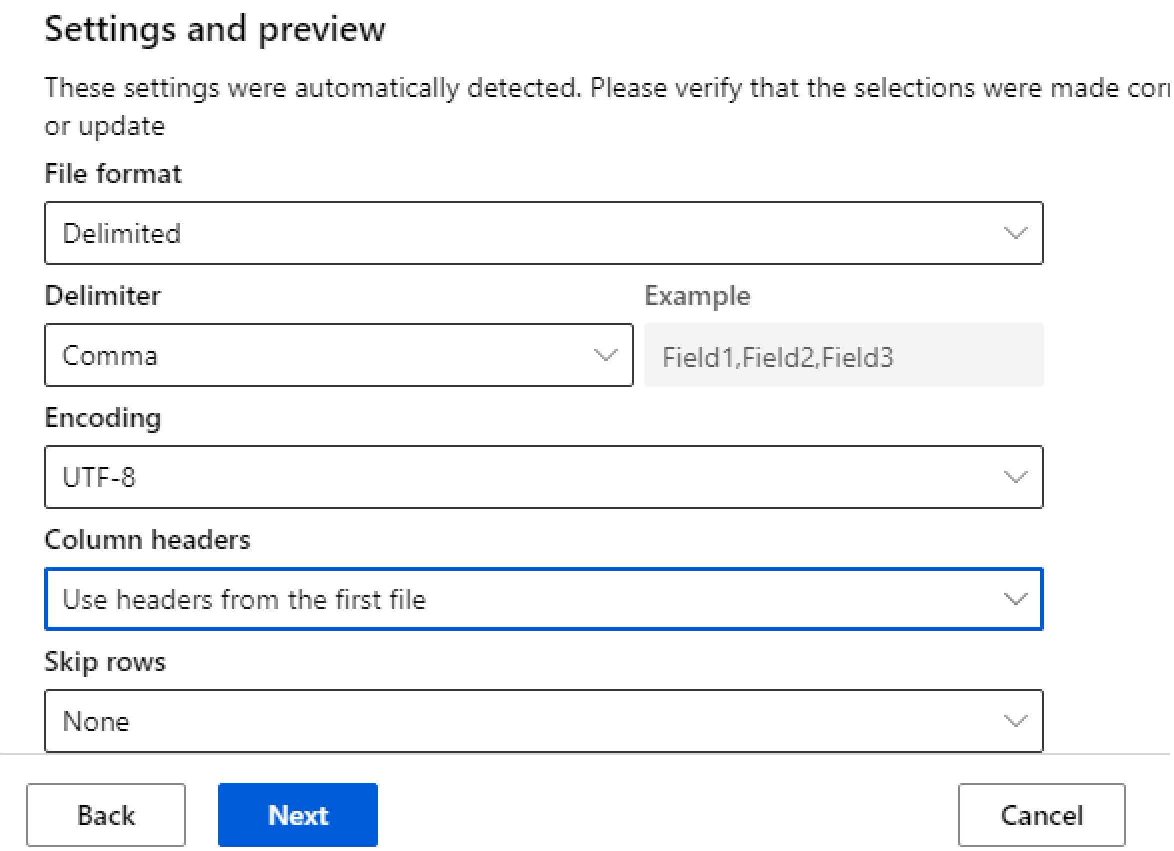
Description

Diabetes data

□ Skip data validation

Settings and preview:

- **File format**: Delimited
- **Encoding**: Comma
- **Column headers**: Use headers from first file
- **Skip rows**: None



Settings and preview

These settings were automatically detected. Please verify that the selections were made cor
or update

File format

Delimited

Delimiter

Comma

Example

Field1,Field2,Field3

Encoding

UTF-8

Column headers

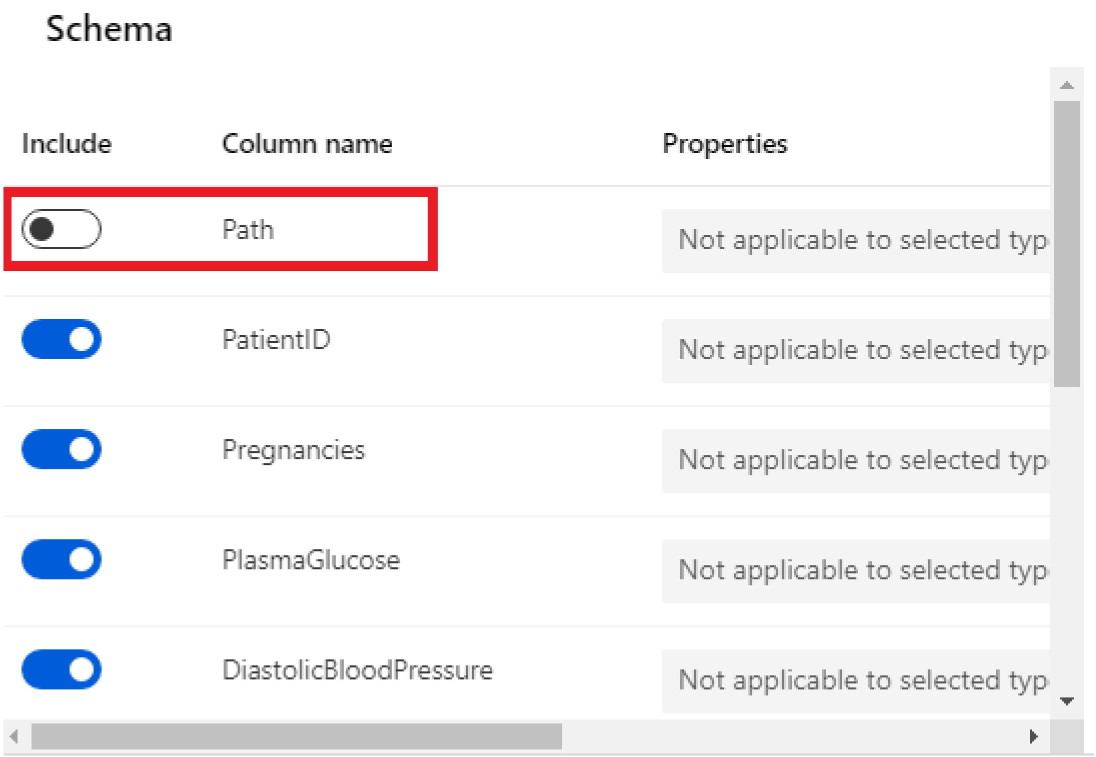
Use headers from the first file

Skip rows

None

Schema:

- Include all columns other than *Path*
- Review the automatically detected types



Schema

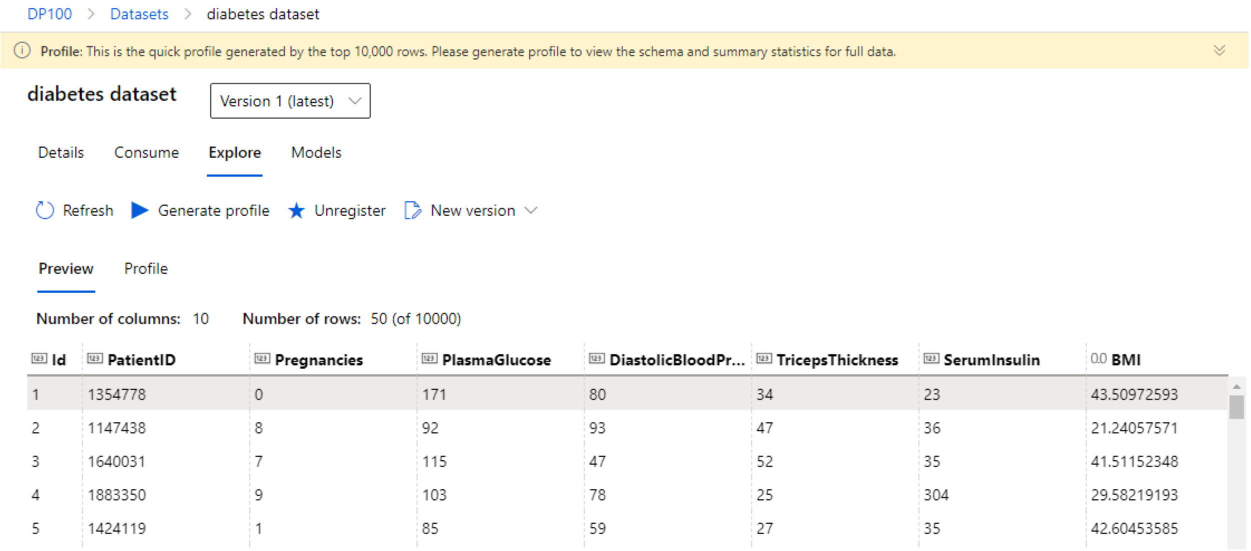
Include	Column name	Properties
○	Path	Not applicable to selected typ
●	PatientID	Not applicable to selected typ
●	Pregnancies	Not applicable to selected typ
●	PlasmaGlucose	Not applicable to selected typ
●	DiastolicBloodPressure	Not applicable to selected typ

Confirm details:

- Do not profile the datasets after creation

□ Profile this dataset after creation

4. After the dataset has been created, open it and view the **Explore** page to see a sample of the data. This data represents details from patients who have been tested for diabetes.



Profile: This is the quick profile generated by the top 10,000 rows. Please generate profile to view the schema and summary statistics for full data.

diabetes dataset

Number of columns: 10 Number of rows: 50 (of 10000)

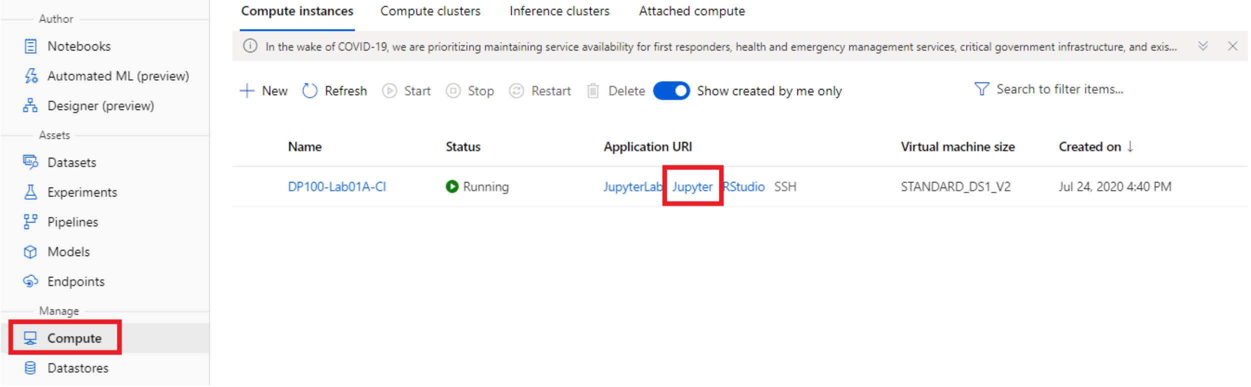
Id	PatientID	Pregnancies	PlasmaGlucose	DiastolicBloodPr...	TricepsThickness	SerumInsulin	BMI
1	1354778	0	171	80	34	23	43.50972593
2	1147438	8	92	93	47	36	21.24057571
3	1640031	7	115	47	52	35	41.51152348
4	1883350	9	103	78	25	304	29.58219193
5	1424119	1	85	59	27	35	42.60453585

Lab 1B: Working with Azure Machine Learning Tools

Task 1: Use the Azure ML SDK in a Compute Instance

You can perform most asset management tasks to set up your environment in the *Studio* interface, but it's also important to be able to script configuration tasks to make them easier to repeat and automate.

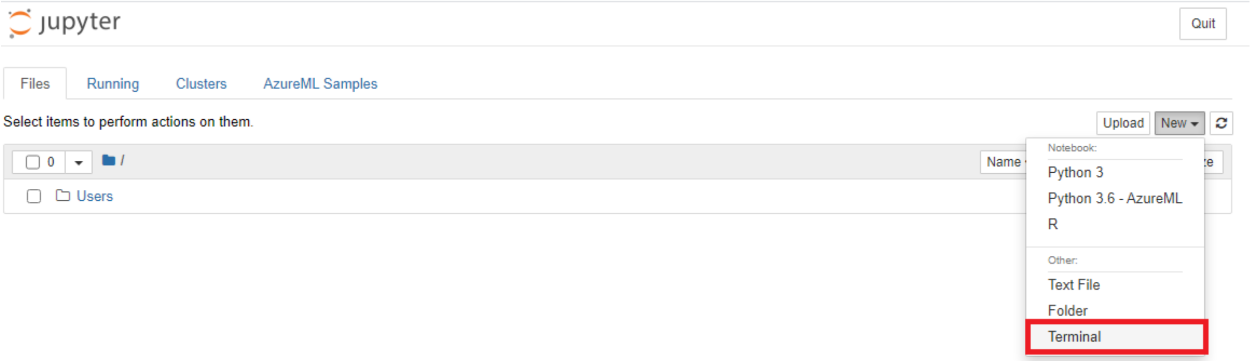
1. In Azure Machine Learning studio, on the **Compute** page for your workspace, view the **Compute Instances** tab, and if necessary, click **Refresh** periodically until the compute instance you created in the previous lab has started.
2. Refresh the Azure Machine Learning studio web page in your browser to ensure your authenticated session has not expired. Then click your compute instance's **Jupyter** link to open Jupyter Notebooks in a new tab. If prompted, sign in using the Microsoft account associated with your Azure Subscription.



In the wake of COVID-19, we are prioritizing maintaining service availability for first responders, health and emergency management services, critical government infrastructure, and exis...

Name	Status	Application URI	Virtual machine size	Created on ↓
DP100-Lab01A-CI	Running	JupyterLab Jupyter RStudio SSH	STANDARD_DS1_V2	Jul 24, 2020 4:40 PM

3. In the notebook environment, create a new **Terminal**. This will open a new tab with a command shell.

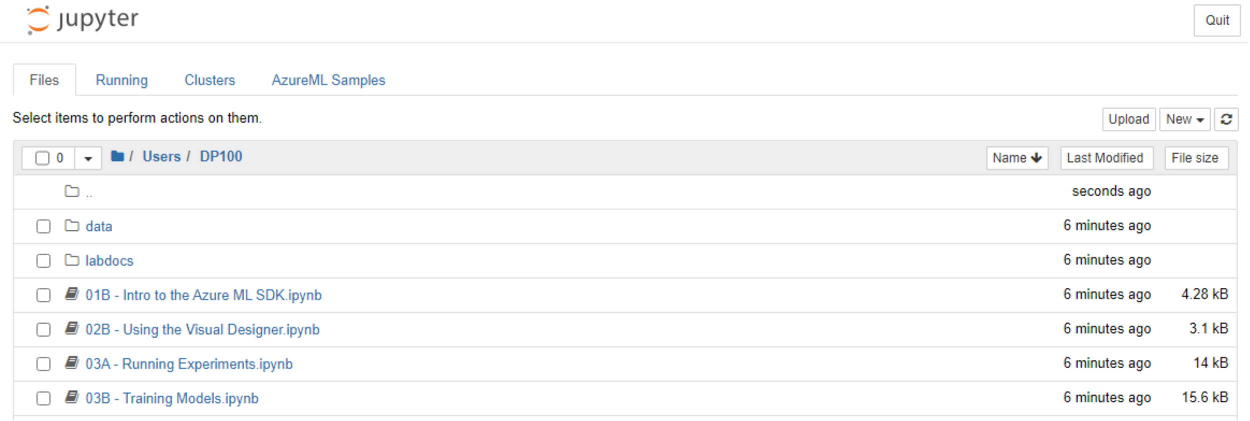


Select items to perform actions on them.

4. The Azure Machine Learning SDK is already installed in the compute instance image, but it's worth ensuring you have the latest version, with the optional packages you'll need in this course; so enter the following command to update the SDK packages:
5.
```
pip install --upgrade azureml-sdk[notebooks,automl,explain]
```

   You may see some warnings as the package dependencies are installed. You can ignore these.

6. Next, run the following commands to change the current directory to the **Users** directory, and retrieve the notebooks you will use in the labs for this course:
7. `cd Users`
8. `git clone https://github.com/MicrosoftLearning/DP100`
9. After the command has completed, close the terminal tab and view the home page in your Jupyter notebook file explorer. Then open the **Users** folder - it should contain an **DP100** folder, containing the files you will use in the rest of this lab.

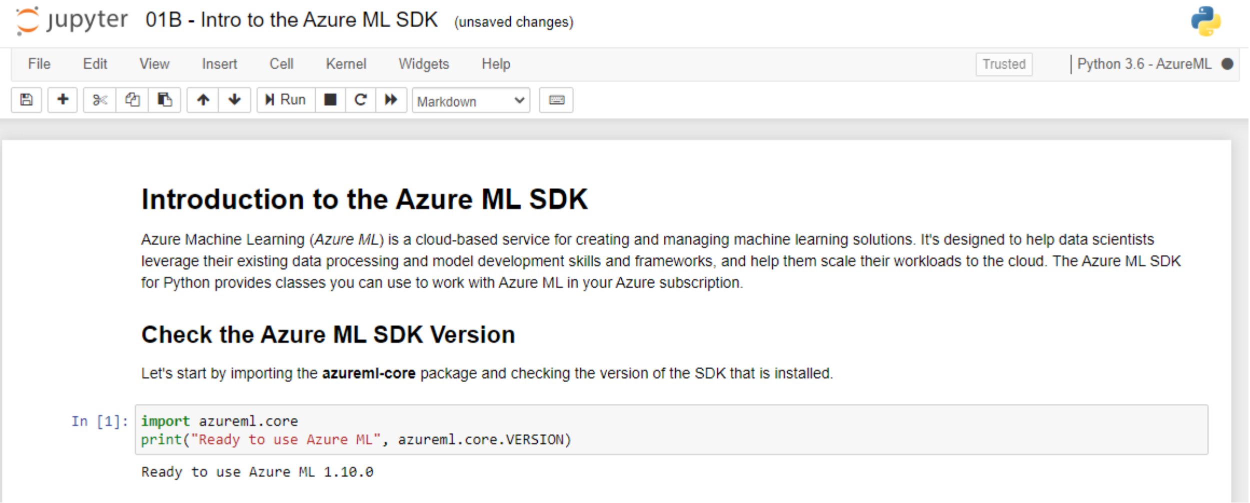


Select items to perform actions on them.

/ Users / DP100

Name	Last Modified	File size
..	seconds ago	
data	6 minutes ago	
labdocs	6 minutes ago	
01B - Intro to the Azure ML SDK.ipynb	6 minutes ago	4.28 kB
02B - Using the Visual Designer.ipynb	6 minutes ago	3.1 kB
03A - Running Experiments.ipynb	6 minutes ago	14 kB
03B - Training Models.ipynb	6 minutes ago	15.6 kB

10. In the **Users/DP100** folder, open the **01B - Intro to the Azure ML SDK.ipynb** notebook. Then read the notes in the notebook, running each code cell in turn.



Introduction to the Azure ML SDK

Azure Machine Learning (*Azure ML*) is a cloud-based service for creating and managing machine learning solutions. It's designed to help data scientists leverage their existing data processing and model development skills and frameworks, and help them scale their workloads to the cloud. The Azure ML SDK for Python provides classes you can use to work with Azure ML in your Azure subscription.

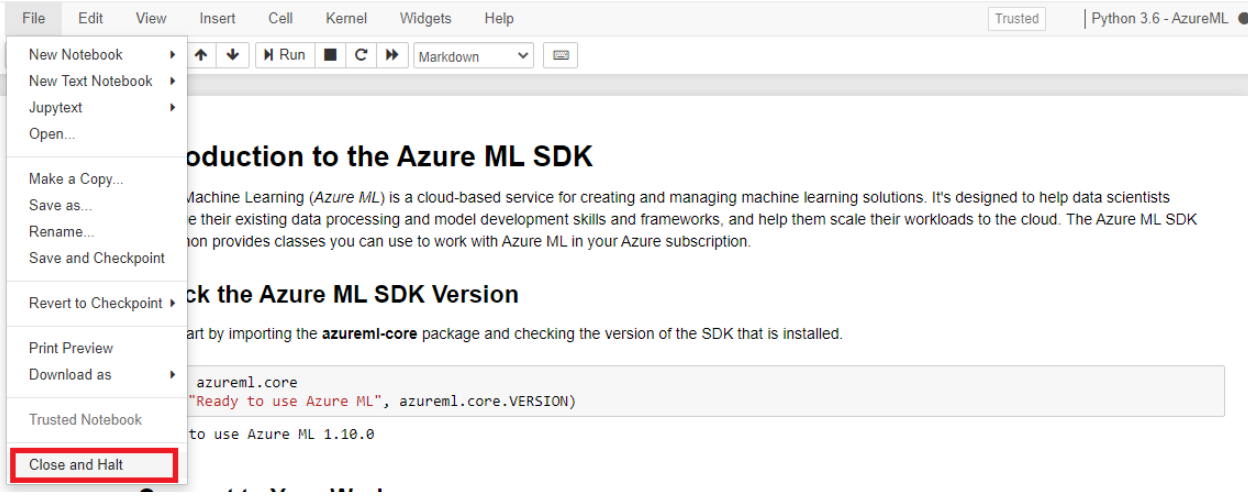
Check the Azure ML SDK Version

Let's start by importing the **azureml-core** package and checking the version of the SDK that is installed.

In [1]:
```
import azureml.core
print("Ready to use Azure ML", azureml.core.VERSION)
```

```
Ready to use Azure ML 1.10.0
```

11. When you have finished running the code in the notebook, on the **File** menu, click **Close and Halt** to close it and shut down its Python kernel. Then close all Jupyter browser tabs.



oduction to the Azure ML SDK

lachine Learning (*Azure ML*) is a cloud-based service for creating and managing machine learning solutions. It's designed to help data scientists e their existing data processing and model development skills and frameworks, and help them scale their workloads to the cloud. The Azure ML SDK on provides classes you can use to work with Azure ML in your Azure subscription.

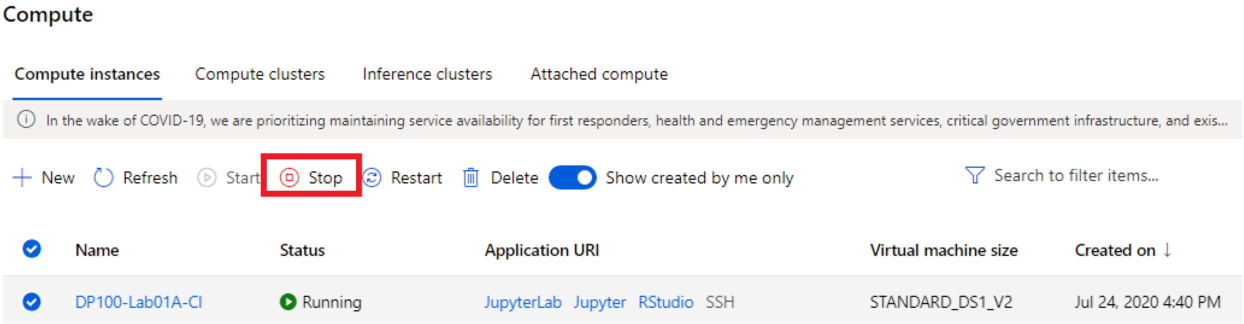
ck the Azure ML SDK Version

art by importing the **azureml-core** package and checking the version of the SDK that is installed.

```
azureml.core
"Ready to use Azure ML", azureml.core.VERSION)
```

```
to use Azure ML 1.10.0
```

12. If you're finished working with Azure Machine Learning for the day, in Azure Machine Learning studio, on the **Compute** page, select your compute instance and click **Stop** to shut it down. Otherwise, leave it running for the next lab.



In the wake of COVID-19, we are prioritizing maintaining service availability for first responders, health and emergency management services, critical government infrastructure, and exis...

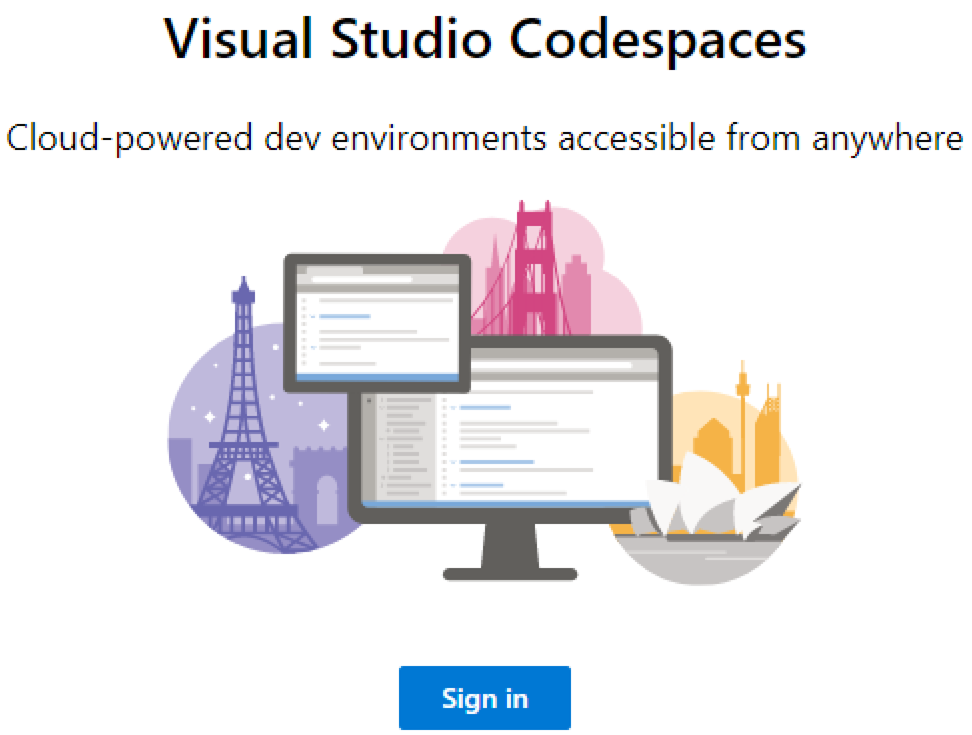
✓	Name	Status	Application URI	Virtual machine size	Created on ↓
✓	DP100-Lab01A-CI	● Running	JupyterLab Jupyter RStudio SSH	STANDARD_DS1_V2	Jul 24, 2020 4:40 PM

Task 2: Set Up a Visual Studio Codespace

Compute instances in Azure Machine Learning provide an easy to manage Python environment for working with Azure ML without the need to manage your own Python installation. However, sometimes you may want to use your own graphical Python development environment. In this course, we'll use a Visual Studio Codespace to simplify installation, but the principles of using the Azure Machine Learning SDK are the same in any Python environment.

Note: Visual Studio Codespaces is in *preview* at the time of writing. You may experience some unexpected error messages.

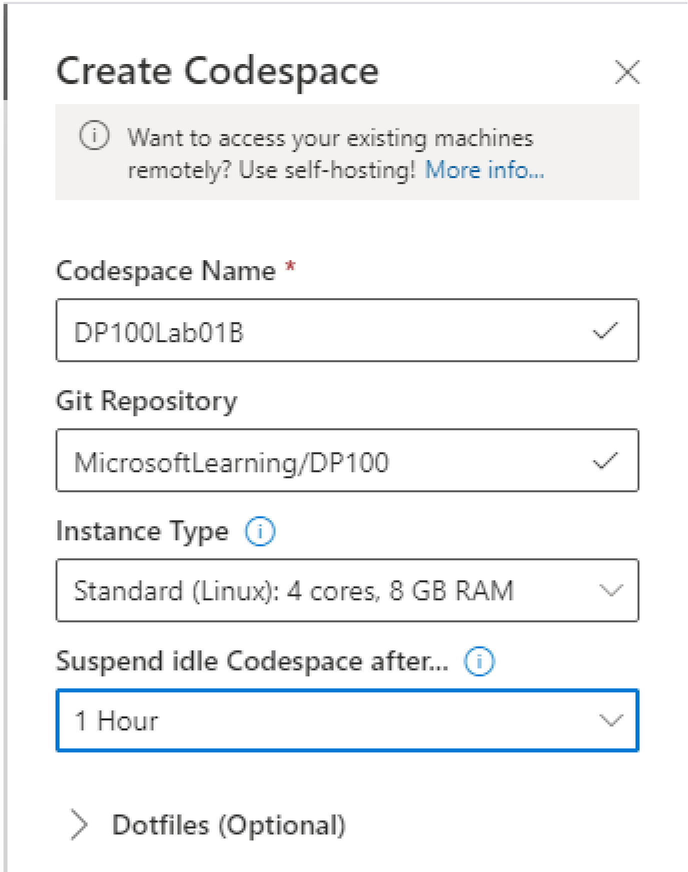
1. In a new browser tab, navigate to https://online.visualstudio.com. If prompted, sign into Visual Studio Codespaces using the same Microsoft credentials you used to sign into Azure.



Visual Studio Codespaces

Cloud-powered dev environments accessible from anywhere

2. Create a codespace with the following settings(if you don't already have a Visual Studio codespaces plan, create one when prompted - this is used to track resource utilization by your codespaces):
   - **Codespace Name**: *A unique name of your choice**
   - **Git Repository**: MicrosoftLearning/DP100
   - **Instance Type**: Standard(Linux)
   - **Suspend idle Codespace after**: 1 Hour



Create Codespace

Want to access your existing machines remotely? Use self-hosting! More info...

Codespace Name *

DP100Lab01B

Git Repository

MicrosoftLearning/DP100

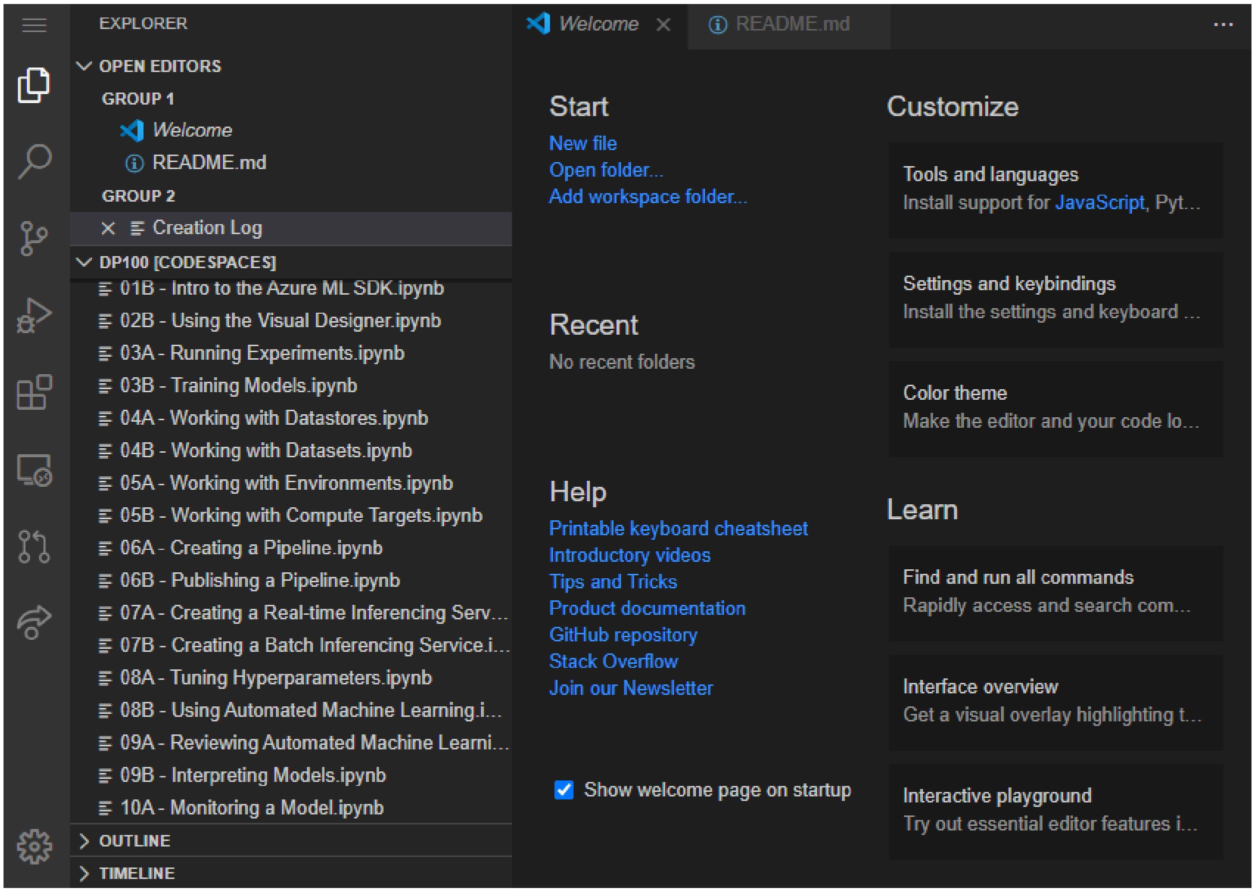
Instance Type

Standard (Linux): 4 cores, 8 GB RAM

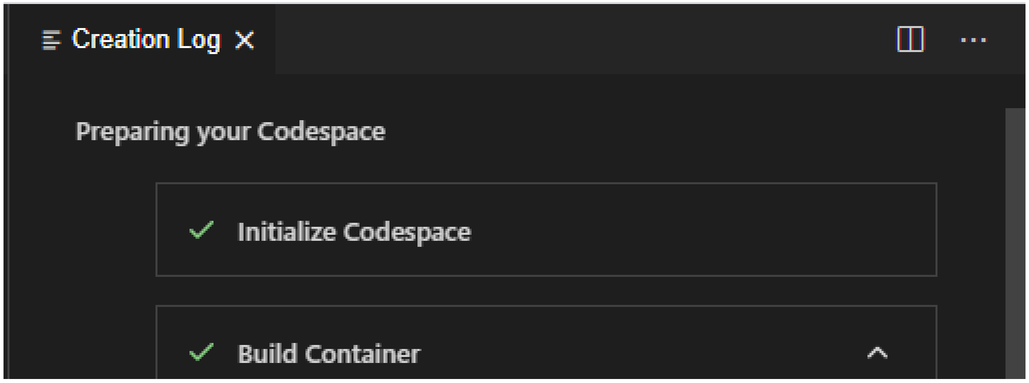
Suspend idle Codespace after...

1 Hour

3. Wait for the codespace to be created. This will open a browser-based instance of Visual Studio Code.
4. Wait for a minute or so while the environment is set up for you. It might look like nothing is happening, but in the background we are installing some extensions that you will use in the labs. You'll see the following things happen:
   - A script pane will open to show status as your codespace is prepared.
   - The Visual Studio Code interface will be loaded.
   - The file in this repo will appear in the pane on the left.


EXPLORER
OPEN EDITORS
GROUP 1
Welcome
README.md
GROUP 2
Creation Log
DP100 [CODESPACES]
01B - Intro to the Azure ML SDK.ipynb
02B - Using the Visual Designer.ipynb
03A - Running Experiments.ipynb
03B - Training Models.ipynb
04A - Working with Datastores.ipynb
04B - Working with Datasets.ipynb
05A - Working with Environments.ipynb
05B - Working with Compute Targets.ipynb
06A - Creating a Pipeline.ipynb
06B - Publishing a Pipeline.ipynb
07A - Creating a Real-time Inferencing Serv...
07B - Creating a Batch Inferencing Service.i...
08A - Tuning Hyperparameters.ipynb
08B - Using Automated Machine Learning.i...
09A - Reviewing Automated Machine Learni...
09B - Interpreting Models.ipynb
10A - Monitoring a Model.ipynb
OUTLINE
TIMELINE
Welcome
README.md
Start
New file
Open folder...
Add workspace folder...
Recent
No recent folders
Help
Printable keyboard cheatsheet
Introductory videos
Tips and Tricks
Product documentation
GitHub repository
Stack Overflow
Join our Newsletter
☑ Show welcome page on startup
Customize
Tools and languages
Install support for JavaScript, Pyt...
Settings and keybindings
Install the settings and keyboard ...
Color theme
Make the editor and your code lo...
Learn
Find and run all commands
Rapidly access and search com...
Interface overview
Get a visual overlay highlighting t...
Interactive playground
Try out essential editor features i...

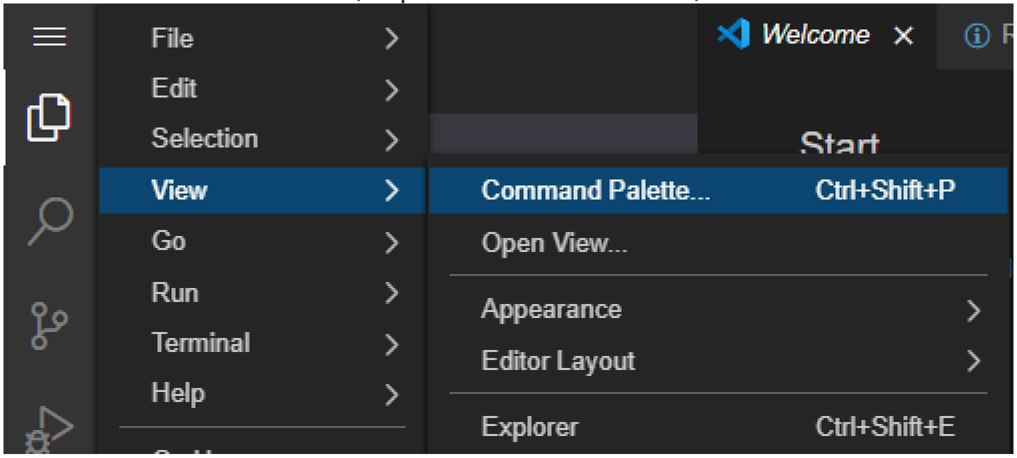
5. After setup has completed successfully, you can close the **Creation Log** pane.



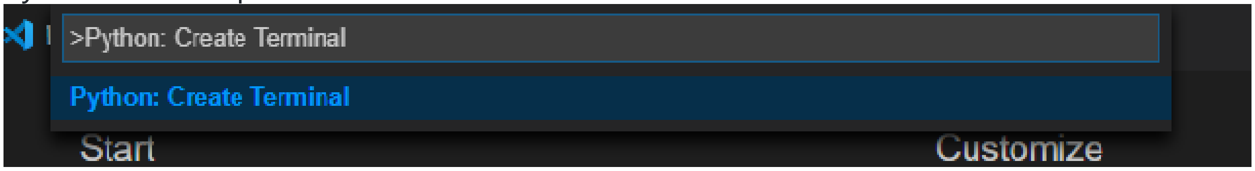
A Visual Studio Codespace is a hosted instance of Visual Studio Code that you can use in a web browser. Visual Studio Code is general code editing environment, with support for various programming languages through the installation of *extensions*. To work with Python, you'll need the Microsoft Python extension, which was installed for you along with some commonly used Python packages when you created this environment from the **DP100** repo.

The codespace includes an installation of Python(version 3.x), including common packages and support for Jupyter Notebooks within the Visual Studio Code interface. To run code that works with Azure Machine Learning, you just need to install the Azure ML SDK.

6. In the Visual Studio codespace, in the Application Menu (☰), on the **View** menu, click **Command Palette** (or press CTRL+SHIFT+P).



Then in the Palette, enter the command **Python: Create Terminal**. This opens a Python terminal pane at the bottom of the interface.



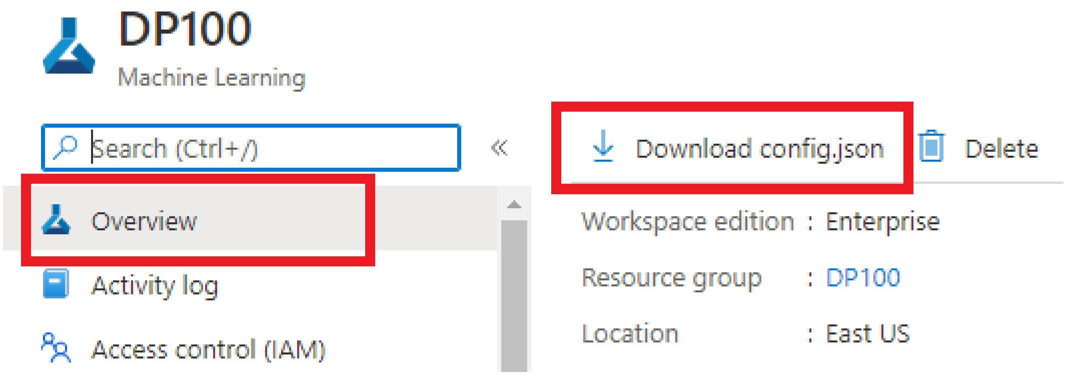
>Python: Create Terminal

7. In the terminal pane, enter the following command to install the Azure Machine Learning SDK (with the optional *notebooks* extra package) using this command:
8. `pip install azureml-sdk[notebooks]`
9. Close the Terminal pane.

Task 3: Use the Azure ML SDK in Visual Studio Codespaces

Now that you have a Python development environment, you can use the Azure Machine Learning SDK in it. First, you need to get the configuration information required to connect to your Azure Machine Learning workspace.

1. In a new browser tab, open the Azure portal at https://portal.azure.com, signing in if necessary.
2. Open the Azure Machine Learning workspace resource you created in the previous lab, and on its **Overview** page, click **Download config.json** and download the file to your local computer.



Workspace edition : Enterprise

Resource group : DP100

Location : East US

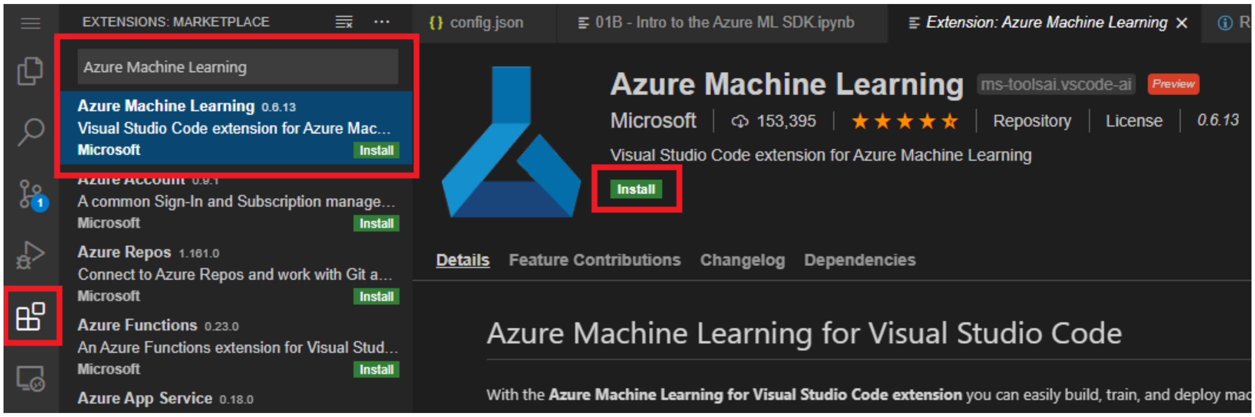
3. From your local computer, drag the downloaded **config.json** file into the Codespace in your browser, and drop it on the notebook files there. This uploads the config file and opens it in the Codespace editor.
4. Review the contents of the config.json file, and then close it.
5. In your codespace, open the **01B - Intro to the Azure ML SDK.ipynb** notebook - this will be loaded in the Jupyter Notebook interface. It may take a while to load the first time the Jupyter Notebooks interface is used, and you may briefly see two panes - one containing the JSON representation of the notebook, and the other containing the notebook visual interface.
6. When the notebook has loaded, read the notes it contains and run each code cell in turn, just as you did in the Azure Machine Learning Notebook VM Jupyter environment.

Task 4: Use the Visual Studio Code Azure Machine Learning Extension

If you plan to work with Azure Machine Learning in a Visual Studio codespace (or a local installation of Visual Studio Code), the Azure Machine Learning extension can help make it easier to work with resources in your workspace without needing to switch between your code development environment and the Azure Machine Learning studio web interface.

1. In the Visual Studio codespace interface, click the **Extensions** tab (⊞), and search for "Azure Machine Learning". Then install the **Azure Machine Learning** extension from Microsoft. After the extension has installed, click the **Reload Required** button to reload the environment with the

extension.



Visual Studio Code extension for Azure Mac...

A common Sign-In and Subscription manage...

Connect to Azure Repos and work with Git a...

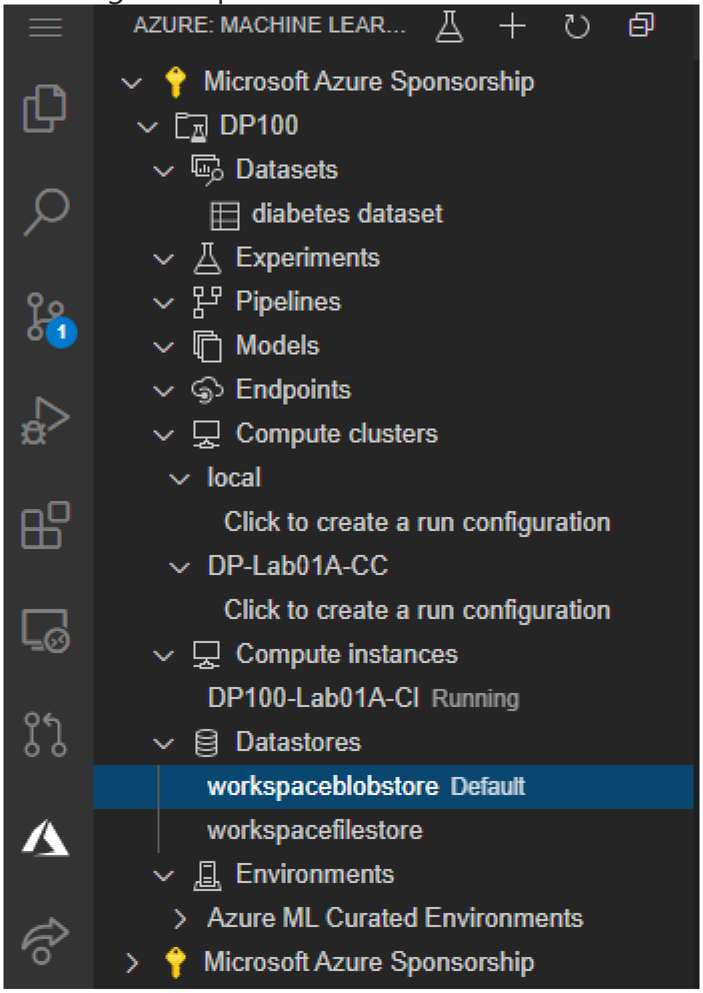
An Azure Functions extension for Visual Stud...

Visual Studio Code extension for Azure Machine Learning

Azure Machine Learning for Visual Studio Code

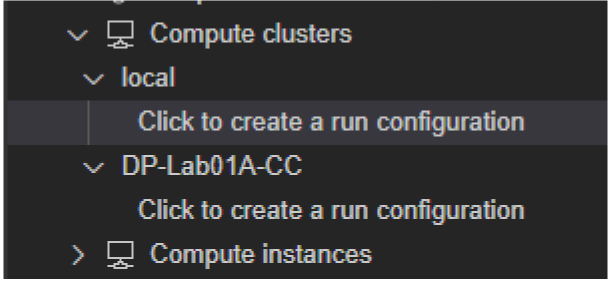
With the **Azure Machine Learning for Visual Studio Code extension** you can easily build, train, and deploy mac

2. In the Visual Studio codespace interface, click the **Azure** tab (***Δ***) and in the **Azure Machine Learning** section, expand your subscription and your Azure Machine Learning workspace.



3. Expand **Compute Clusters** and verify that the **aml-cluster** compute resource you created in your workspace is listed along with a **local** compute resource, which in this case represents the hosted codepace environment - you can run Azure

Machine Learning code experiments on local compute as well as on compute resources defined in the workspace.



- Compute clusters
  - local
    - Click to create a run configuration
  - DP-Lab01A-CC
    - Click to create a run configuration
- Compute instances

4. Close the Visual Studio codespace browser tab.

What is Azure Machine Learning?

Azure Machine Learning is a platform for operating machine learning workloads in the cloud.

Built on the Microsoft Azure cloud platform, Azure Machine Learning enables you to manage:

- Scalable on-demand compute for machine learning workloads.
- Data storage and connectivity to ingest data from a wide range sources.
- Machine learning workflow orchestration to automate model training, deployment, and management processes.
- Model registration and management, so you can track multiple versions of models and the data on which they were trained.
- Metrics and monitoring for training experiments, datasets, and published services.
- Model deployment for real-time and batch inferencing.

Azure Machine Learning Workspaces

A workspace is a context for the experiments, data, compute targets, and other assets associated with a machine learning workload.

A workspace defines the boundary for a set of related machine learning assets. You can use workspaces to group machine learning assets based on projects, deployment environments (for example, test and production), teams, or some other organizing principle. The assets in a workspace include:

- Compute targets for development, training, and deployment.
- Data for experimentation and model training.
- Notebooks containing shared code and documentation.
- Experiments, including run history with logged metrics and outputs.
- Pipelines that define orchestrated multi-step processes.
- Models that you have trained.

Workspaces are Azure resources, and as such they are defined within a resource group in an Azure subscription, along with other related Azure resources that are required to support the workspace.

The Azure resources created alongside a workspace include:

- A storage account - used to store files used by the workspace as well as data for experiments and model training.
- An Application Insights instance, used to monitor predictive services in the workspace.
- An Azure Key Vault instance, used to manage secrets such as authentication keys and credentials used by the workspace.

- Virtual Machines, and their associated virtual hardware resources, used to provide compute for notebook development in the workspace.
- A container registry, used to manage containers for deployed models.

Considerations for Creating a Workspace

When planning to create an Azure Machine Learning workspace, there are some options you should consider.

Region

Your workspace will be created in an Azure region, which determines the data center in which the workspace resources will be hosted. You should ensure that the region where you create the workspace will support the resources you will need. For example, you might need to create virtual machines in your workspace to support notebook development or model training. If you intend to leverage graphical processing unit (GPU) processing, which are commonly used to perform training of deep neural network (DNN) models, then you should ensure that the region where your workspace is hosted supports the appropriate series of virtual machines (for example, NC-series virtual machines support GPUs, but are not available in all regions).

Edition

Azure Machine Learning workspaces are available in two editions:

- Enterprise - includes all features.
- Basic - includes core features, but does not include Designer or graphical tools for automated machine learning or data drift monitoring.

Azure Machine Learning studio

You can manage the assets in your Azure Machine Learning workspace in the Azure portal, but as this is a general interface for managing all kinds of resources in Azure, data scientists and other users involved in machine learning operations may prefer to use a more focused, dedicated interface.

The Azure Machine Learning SDK for Python

While graphical interfaces like Azure Machine Learning studio make it easy to create and manage machine learning assets, it is often advantageous to use a code-based approach to managing resources. By writing scripts to create and manage resources, you can:

- Automate asset creation and configuration to make it repeatable.
- Ensure consistency for resources that must be replicated in multiple environments (for example, development, test, and production)
- Incorporate machine learning asset configuration into developer operations (DevOps) workflows, such as continuous integration / continuous deployment (CI/CD) pipelines. Azure

Machine Learning provides software development kits (SDKs) for Python and R, which you can use to create, manage, and use assets in an Azure Machine Learning workspace.

The Azure Machine Learning CLI Extension

The Azure command-line interface (CLI) is a cross-platform command-line tool for managing Azure resources. The Azure Machine Learning CLI extension is an additional package that provides commands for working with Azure Machine Learning.

Compute Instances

Azure Machine Learning includes the ability to create Compute Instances in a workspace to provide a development environment that is managed with all of the other assets in the workspace.

Compute Instances include Jupyter Notebook and JupyterLab installations that you can use to write and run code that uses the Azure Machine Learning SDK to work with assets in your workspace.

You can choose a VM image that provides the compute specification you need, from small CPU-only VMs to large GPU-enabled workstations. Because the VMs are hosted in Azure, you only pay for the compute resources when they are running; so you can create a VM to suit your needs, and stop it when your workload has completed to minimize costs.

You can store notebooks independently in workspace storage, and open them in any VM.

The Azure Machine Learning Extension for VS Code

Visual Studio Code (VS Code) is a lightweight code editing environment for Microsoft Windows, Apple macOS, and Linux. It provides a visual interface for many kinds of code, including Microsoft C#, Javascript, Python and others; as well as intellisense and syntax formatting for common data formats such as JSON and XML.

VS Code's flexibility is based on the ability to install modular extensions that add syntax checking, debugging, and visual management interfaces for specific workloads. For example, the Microsoft Python extension for VS Code adds support for writing and running Python code in scripts or notebooks within the VS Code interface.