Lab 2 - Working with Notebook VM

One of the most useful aspects of an Azure ML studio is its ability to do logging of measures of interest which will be covered in the next lab. Through this, for example, it is possible to log how a model's performance metrics develop iteration by iteration.

Before taking advantage of this capability, however, we will set up a Notebook VM to run our python code in. Even though all the features of Azure ML could be worked with from a local python runtime on your machine, for the purpose of this series of labs we will work with the Notebook VM to ensure all participant run the same version of python and the libraries required.

Setting up a Notebook VM

When interacting with the Azure ML workspace, data scientists enjoy freedom of choice when it comes to the development environment they wish to work with. The following options can be chosen from.

Environment	Pros	Cons
Cloud-based notebook VM	Easiest way to get started. The entire SDK is already installed in your workspace VM, and notebook tutorials are pre-cloned and ready to run.	Lack of control over your development environment and dependencies. Additional cost incurred for Linux VM (VM can be stopped when not in use to avoid charges). See pricing details.
Local environment	Full control of your development environment and dependencies. Run with any build tool, environment, or IDE of your choice.	Takes longer to get started. Necessary SDK packages must be installed, and an environment must also be installed if you don't already have one.
Azure Databricks	Ideal for running large-scale intensive machine learning workflows on the scalable Apache Spark platform.	Overkill for experimental machine learning, or smaller-scale experiments and workflows. Additional cost incurred for Azure Databricks. See pricing details.
The Data Science Virtual Machine (DSVM)	Similar to the cloud-based notebook VM (Python and the SDK are pre-installed), but with additional popular data science and machine learning tools pre-installed. Easy to scale and combine with other custom tools and workflows.	A slower getting started experience compared to the cloud-based notebook VM.
Azure Notebooks	Free and lightweight getting started experience, with Python and the SDK pre-installed.	Less-powerful VMs available compared to cloud-based notebook VM. Isolated from workspace and other resources.

For the purpose of this lab, we will work with the **Cloud-based notebook VM** which offers a simple setup, integration with Active Directory for single sign-on (SSO) and the azureml python libraries already pre-installed.

To set up the Notebook VM, click "Compute" in the "Manage" section of the Azure ML studio.
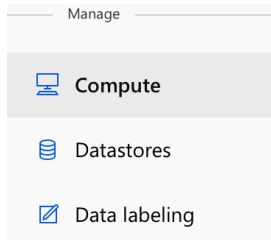
Ensure you are in the "Notebook VMs" tab. Then, click the **New** button to add a new Notebook VM to your workspace.

Specify a name for your VM and choose a VM size that is suitable for the volume and complexity of your dataset as well as your models. **Note that the name of the Notebook VM must be unique within an Azure region.** Hence, you may not use the same names shown in this document or that is used by your neighbor.

Click the **VM type** drop-down to browse the list of all available VM sizes and select a VM size different from the default. In general, consider the data volume available for training and compare it to the amount of memory (RAM) the VM has available as python will attempt to hold the entire dataset in memory to speed up computations. Also, consider using VM types that include GPUs when expecting to train large deep neural networks to speed up training times.

Note: Bigger VMs with more memory and CPUs will result in higher charges to your Azure subscription. More information on VM pricing can be found here. The default VM size STANDARD_D3_V2 with 4 vCPUs and 14 GB of RAM will suffice for our lab.

Virtual machine size

Search by name…				
Name ↑	vCPUs	GPUs	RAM	Resource disk
STANDARD_D3	4		14 GB	200 GB
STANDARD_D32S_V3	32		128 GB	256 GB
STANDARD_D32_V3	32		128 GB	800 GB
✓ STANDARD_D3_V2	4		14 GB	200 GB

Once an appropriate VM type has been selected, click the blue **Create** button to start the deployment process for the Notebook VM.

Create

When the status has changed from **Creating** to **Running**, you may proceed to the next paragraph. If for whatever reason the status shows as **Stopped**, you can start the Notebook VM by selecting it to the left of the name column and then clicking **Start** from the menu bar above.

Notebook VMs   Training Clusters   Inferenc

╋ New   ↻ Refresh   ▶ Start   ◻ Stop

Name	Status
✓ new2	🟥 Stopped

Lastly, when the status of the Notebook VM has changed to **Running**, click the **Jupyter** hyperlink that has now appeared. This will launch a separate browser tab or window that will bring you directly to the Jupyter Notebook environment of your Notebook VM.

Notebook VMs   Training Clusters   Inference Clusters   Attached Compute

╋ New   ↻ Refresh   ▶ Start   ◻ Stop   ↻ Restart   🗑 Delete   ⬤ Show created by me only

Name	Status	Application URI	Virtual Machine size	Created on ↓
nbvm21012020	▶ Running	JupyterLab Jupyter R-Studio	STANDARD_DS3_V2	January 17, 2020 1:56 PM

This should bring you to the Jupyter Notebook home page. Note that the Notebook VM also features an instance of R Studio. For this series of labs the focus will remain on python.

Uploading Notebooks to your Notebook VM

From the Jupyter Notebook home page, you can create new python based notebooks or upload existing ones by clicking the **Upload** button.
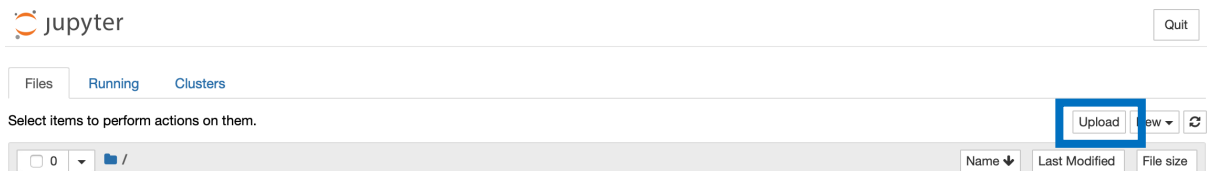
Go ahead and upload the following notebooks which contain the code for the rest of the Azure ML labs.

- `3_Training_and_Logging.ipynb`
- `4_Automated_ML.ipynb`
- `5_Remote_Compute.ipynb`
- `6_Deploy_Model.ipynb`

Conclusion

In this lab you have provisioned your first Notebook VM in your Azure ML workspace. We will use this Notebook VM to run the python code in the following labs.

Next Steps

From your Jupyter Notebook home page, please open the notebook entitled **`3_Training_and_Logging.ipynb`** and follow the instructions provided therein to build your first python based ML model of the day.

Please go through all the notebooks cell by cell and don't run everything at once – your learning experience will be much greater.

If this is your first time working with Jupyter Notebooks, note that you can run the code within an individual cell and advance the cursor to the next cell with **Shift + Enter** on your keyboard. A cell can contain multiple lines of code, all of which will be executed when Shift + Enter are pressed. Alternatively, you can press **Ctrl + Enter** to also run a cell but not advance the cursor.