Set up and Access an Ubuntu Data Science VM and Creating a Jupyter Notebook

Prerequisites

The following are required to complete this hands-on lab:

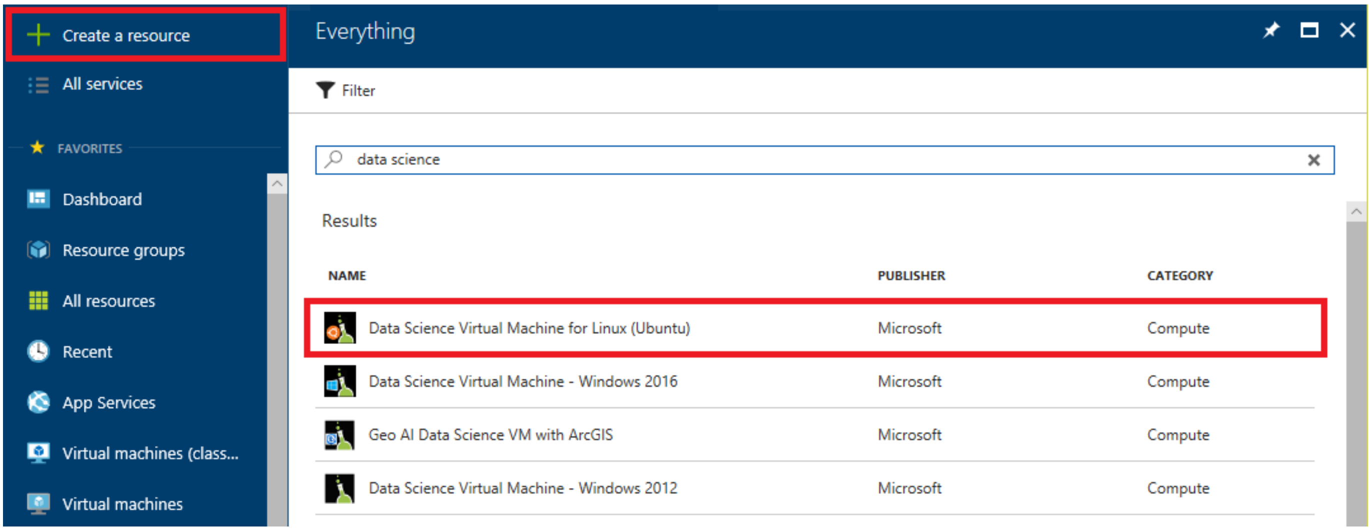
- An active Microsoft Azure subscription.
- An Xfce remote-desktop client such as X2Go

Exercise 1: Create an Ubuntu Data Science VM

The Ubuntu Data Science Virtual Machine for Linux is a virtual-machine image that makes it easy to get started with data science. Multiple tools are already built, installed, and configured in order to get you up and running quickly. The NVIDIA GPU driver, NVIDIA CUDA, and NVIDIA CUDA Deep Neural Network library (cuDNN) are also included, as are Jupyter and several sample Jupyter notebooks. All installed frameworks are GPU-enabled but work on CPUs as well. In this exercise, you will create an instance of the Data Science Virtual Machine for Linux in Azure.

1. Open the Azure Portal in your browser. If asked to log in, do so using your Microsoft account.

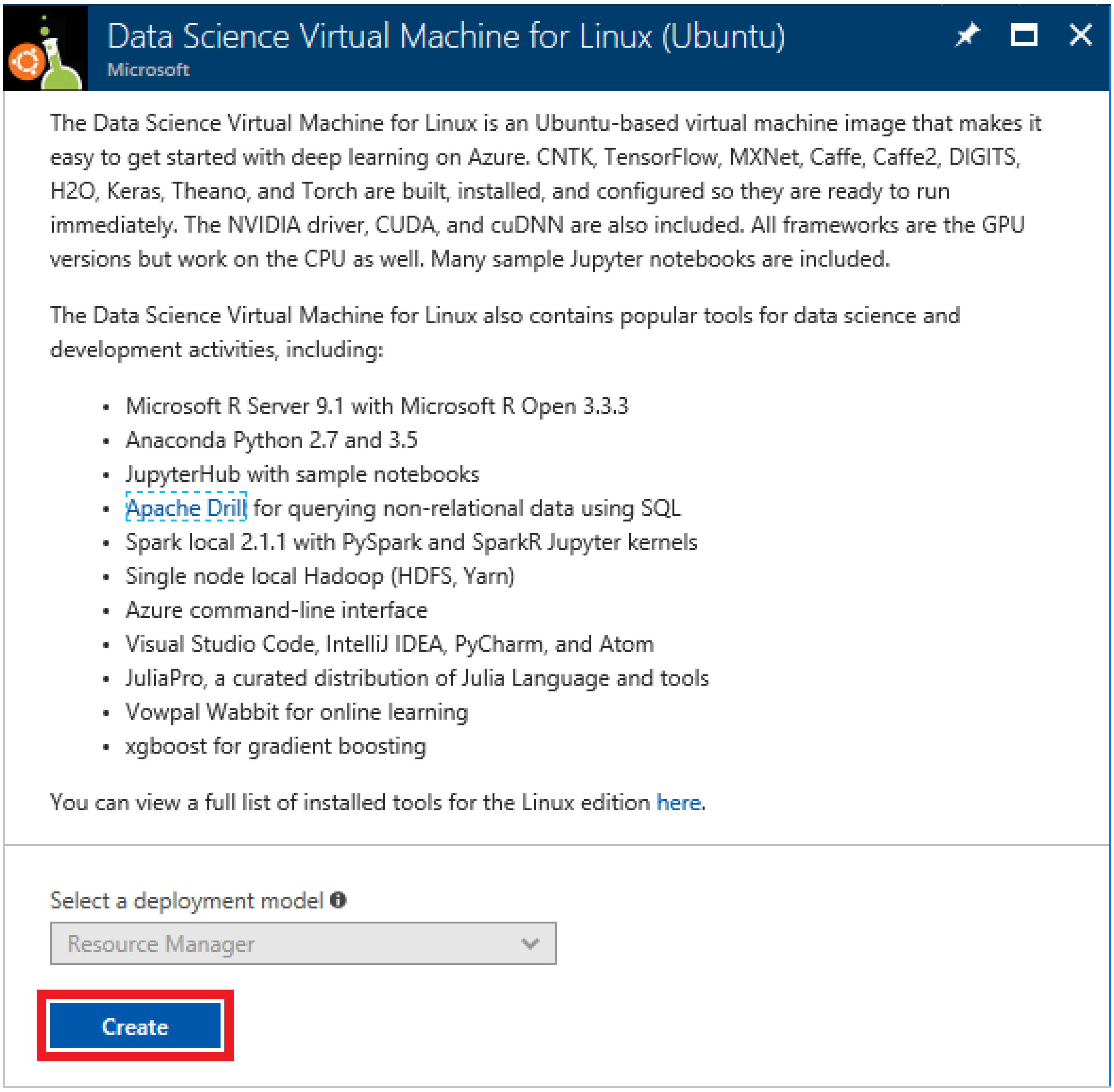
2. Click **+ Create a resource** in the menu on the left side of the portal, and then type "data science" (without quotation marks) into the search box. Select **Data Science Virtual Machine for Linux (Ubuntu)** from the results list.



NAME	PUBLISHER	CATEGORY
Data Science Virtual Machine for Linux (Ubuntu)	Microsoft	Compute
Data Science Virtual Machine - Windows 2016	Microsoft	Compute
Geo AI Data Science VM with ArcGIS	Microsoft	Compute
Data Science Virtual Machine - Windows 2012	Microsoft	Compute

Finding the Ubuntu Data Science VM

3. Take a moment to review the list of tools included in the VM. Then click **Create**.



Data Science Virtual Machine for Linux (Ubuntu)

Microsoft

The Data Science Virtual Machine for Linux is an Ubuntu-based virtual machine image that makes it easy to get started with deep learning on Azure. CNTK, TensorFlow, MXNet, Caffe, Caffe2, DIGITS, H2O, Keras, Theano, and Torch are built, installed, and configured so they are ready to run immediately. The NVIDIA driver, CUDA, and cuDNN are also included. All frameworks are the GPU versions but work on the CPU as well. Many sample Jupyter notebooks are included.

The Data Science Virtual Machine for Linux also contains popular tools for data science and development activities, including:

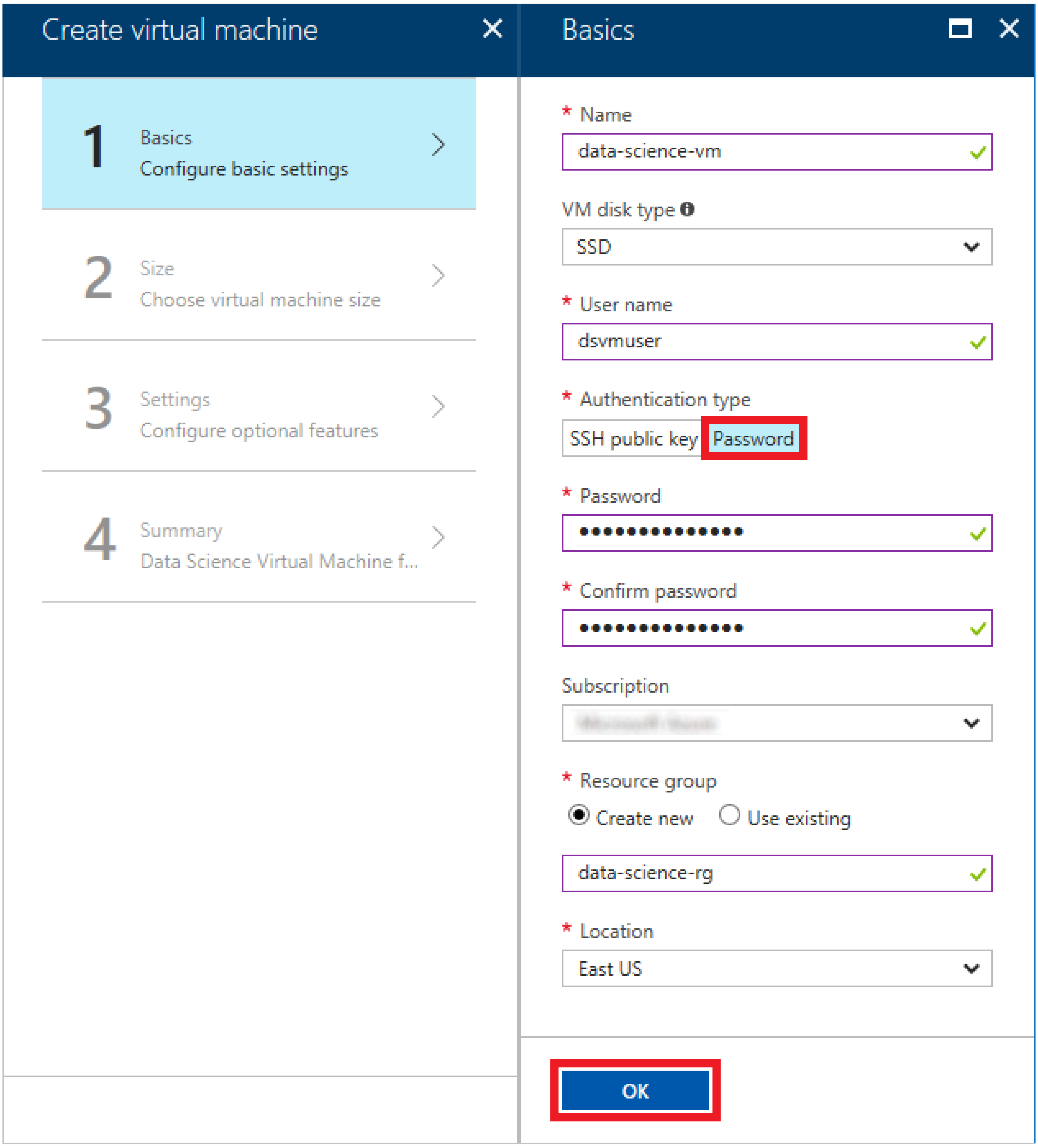
- Microsoft R Server 9.1 with Microsoft R Open 3.3.3
- Anaconda Python 2.7 and 3.5
- JupyterHub with sample notebooks
- Apache Drill for querying non-relational data using SQL
- Spark local 2.1.1 with PySpark and SparkR Jupyter kernels
- Single node local Hadoop (HDFS, Yarn)
- Azure command-line interface
- Visual Studio Code, IntelliJ IDEA, PyCharm, and Atom
- JuliaPro, a curated distribution of Julia Language and tools
- Vowpal Wabbit for online learning
- xgboost for gradient boosting

You can view a full list of installed tools for the Linux edition here.

Select a deployment model: Resource Manager

Creating a Data Science VM

4. Enter a name for the virtual machine and a user name for logging into it. Set **Authentication type** to **Password** and enter a password. *Be sure to remember the user name and password that you enter*, because you will need them to access the VM. Select **Create new** under **Resource group** and enter a resource-group name such as "data-science-rg." Select the **Location** nearest you, and then click **OK**..



Create virtual machine
1 Basics
Configure basic settings
2 Size
Choose virtual machine size
3 Settings
Configure optional features
4 Summary
Data Science Virtual Machine f...
Basics

* Name: data-science-vm

VM disk type: SSD

* User name: dsvmuser

* Authentication type: SSH public key [✓ Password]

* Password: ••••••••••••••

* Confirm password: ••••••••••••••

Subscription: [redacted]

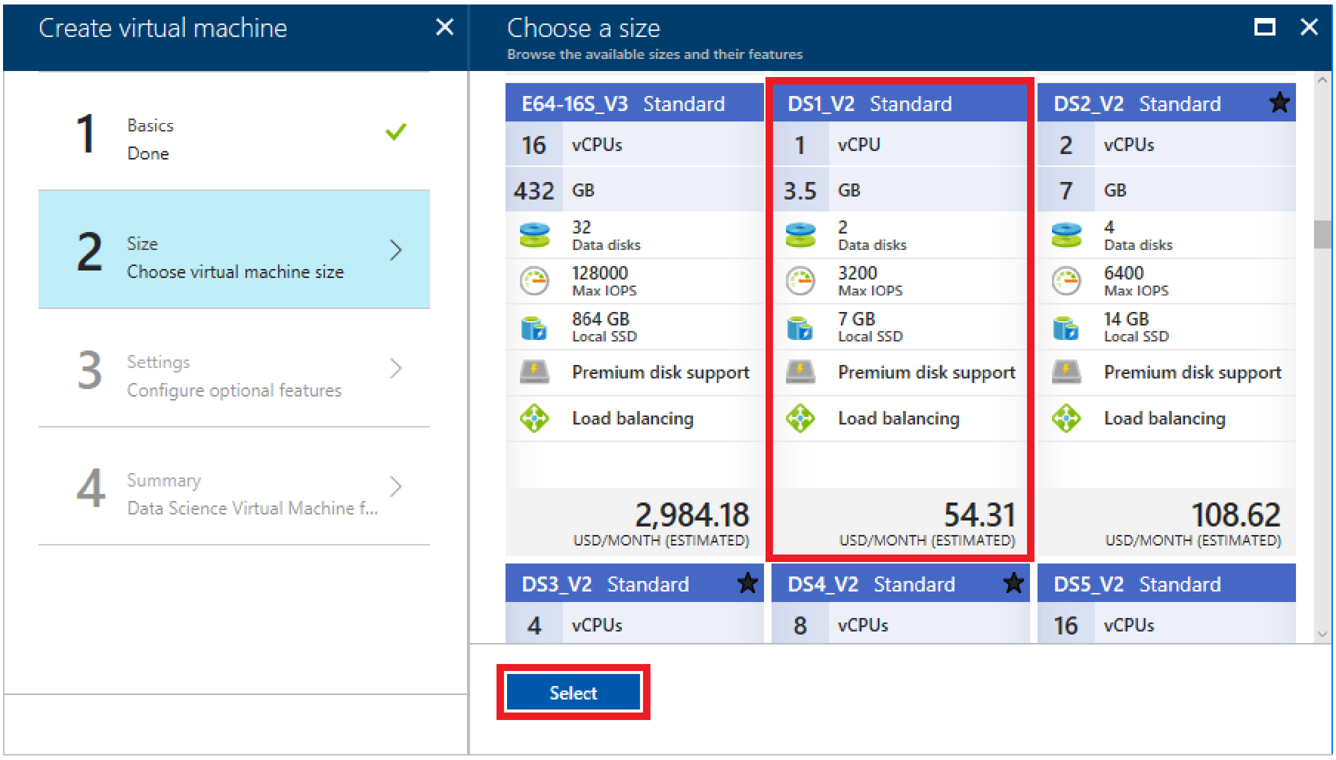
* Resource group: ● Create new ○ Use existing

data-science-rg

* Location: East US

Entering basic information about the VM

5. Next, choose a size for the VM. In order to show all size options available, click **View All**. Scroll down and select **DS1_V2 Standard**, which provides a low-cost way to experiment with Data Science VMs. Then click the **Select** button.



1 Basics
Done

2 Size
Choose virtual machine size

3 Settings
Configure optional features

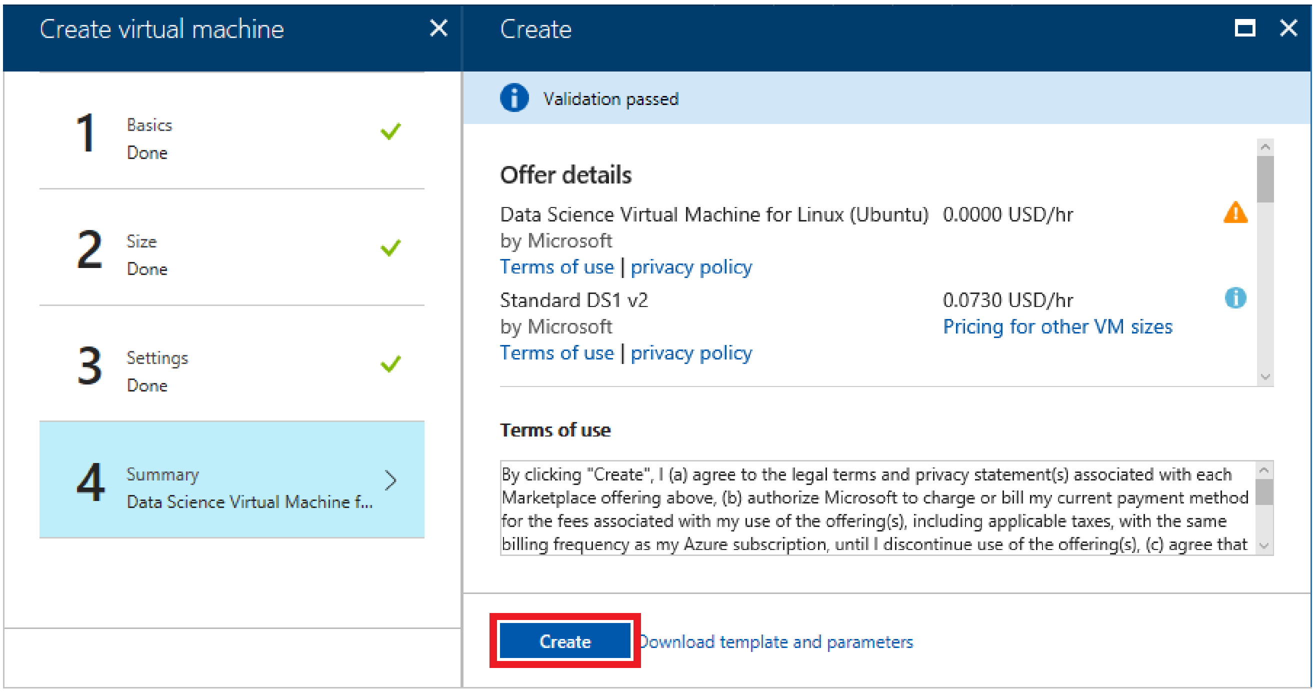
4 Summary
Data Science Virtual Machine f...

E64-16S_V3 Standard	DS1_V2 Standard	DS2_V2 Standard ★
16 vCPUs	1 vCPU	2 vCPUs
432 GB	3.5 GB	7 GB
32 Data disks	2 Data disks	4 Data disks
128000 Max IOPS	3200 Max IOPS	6400 Max IOPS
864 GB Local SSD	7 GB Local SSD	14 GB Local SSD
Premium disk support	Premium disk support	Premium disk support
Load balancing	Load balancing	Load balancing
2,984.18 USD/MONTH (ESTIMATED)	54.31 USD/MONTH (ESTIMATED)	108.62 USD/MONTH (ESTIMATED)

DS3_V2 Standard ★	DS4_V2 Standard ★	DS5_V2 Standard
4 vCPUs	8 vCPUs	16 vCPUs

Choosing a VM size

6. Click **OK** at the bottom of the "Settings" blade. Then take a moment to review the options you selected for the VM, and click **Create** to create it.



1 Basics
Done

2 Size
Done

3 Settings
Done

4 Summary
Data Science Virtual Machine f...

Validation passed

Offer details

Data Science Virtual Machine for Linux (Ubuntu) 0.0000 USD/hr
by Microsoft
Terms of use | privacy policy

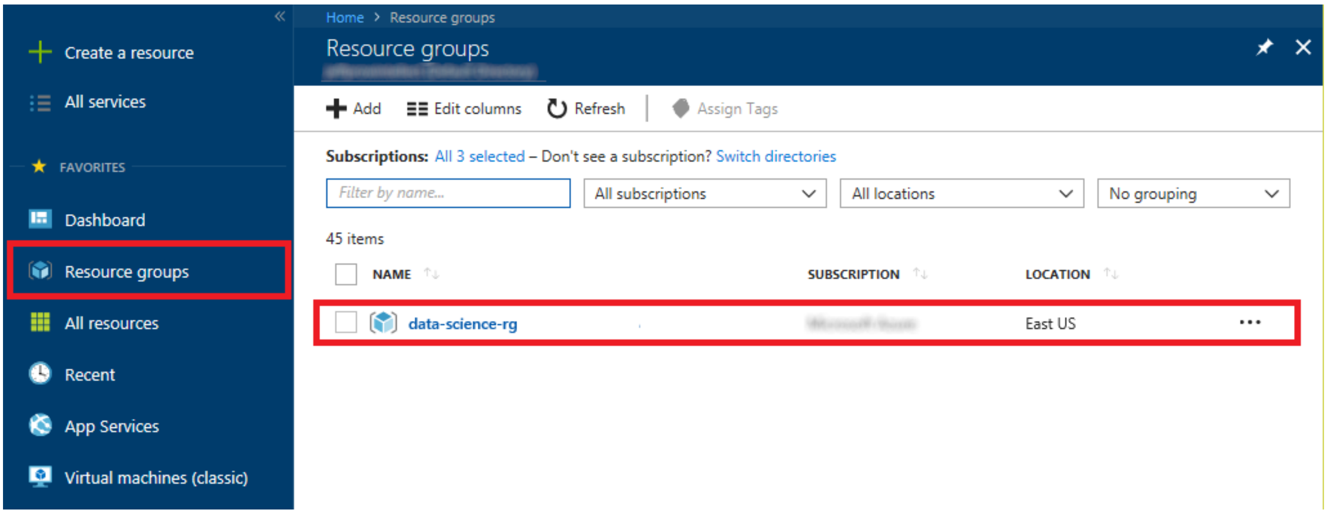
Standard DS1 v2 0.0730 USD/hr
by Microsoft
Terms of use | privacy policy
Pricing for other VM sizes

Terms of use

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with each Marketplace offering above, (b) authorize Microsoft to charge or bill my current payment method for the fees associated with my use of the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s), (c) agree that

Creating the VM

7. Click **Resource groups** in the menu on the left side of the portal. Then click the resource group whose name you specified in Step 4.



[redacted]

Subscriptions: All 3 selected – Don't see a subscription? Switch directories

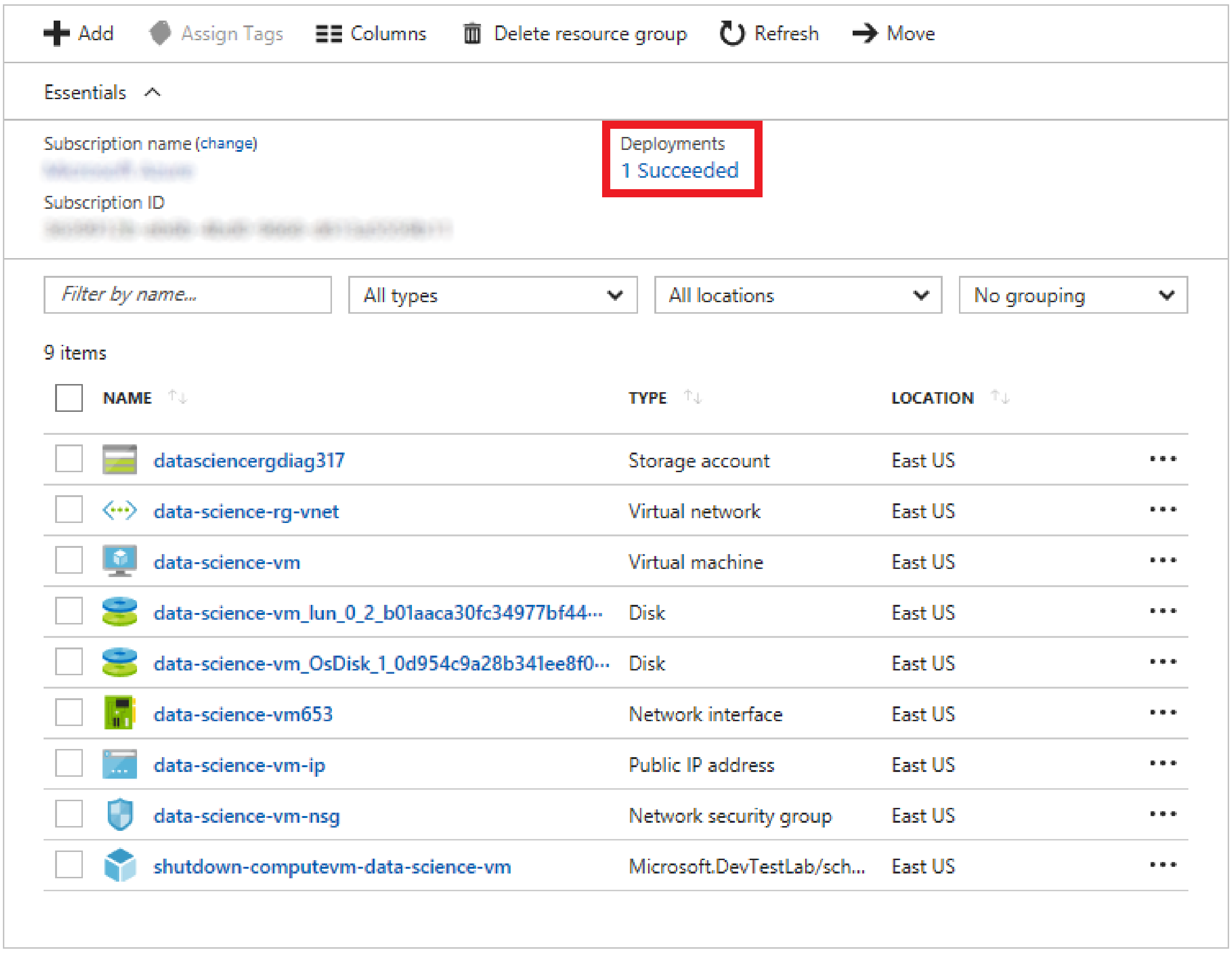
Filter by name... | All subscriptions | All locations | No grouping

45 items

□	NAME	SUBSCRIPTION	LOCATION	
□	data-science-rg	[redacted]	East US	•••

Opening the resource group

8. Wait until "Deploying" changes to "Succeeded" indicating that deployment has completed. Deployment typically takes 5 minutes or less. Periodically click **Refresh** at the top of the blade to refresh the deployment status.



Subscription name (change)
[redacted]

Subscription ID
[redacted]

Deployments
1 Succeeded

9 items

	NAME	TYPE	LOCATION	
□	datasciencergdiag317	Storage account	East US	•••
□	data-science-rg-vnet	Virtual network	East US	•••
□	data-science-vm	Virtual machine	East US	•••
□	data-science-vm_lun_0_2_b01aaca30fc34977bf44···	Disk	East US	•••
□	data-science-vm_OsDisk_1_0d954c9a28b341ee8f0···	Disk	East US	•••
□	data-science-vm653	Network interface	East US	•••
□	data-science-vm-ip	Public IP address	East US	•••
□	data-science-vm-nsg	Network security group	East US	•••
□	shutdown-computevm-data-science-vm	Microsoft.DevTestLab/sch...	East US	•••

Monitoring the deployment status

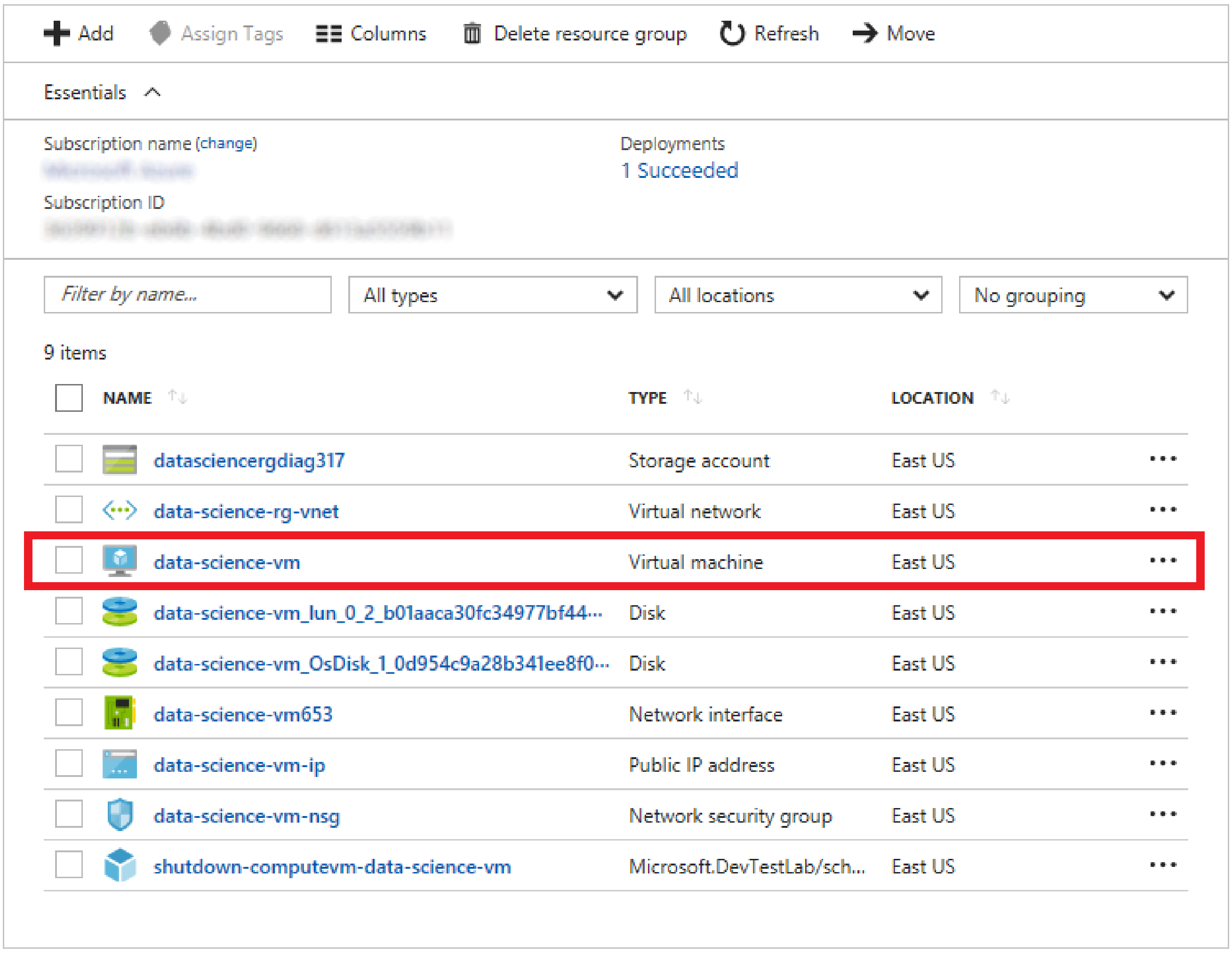
The VM has been created. The next step is to connect to it remotely so you can work with the VM's Ubuntu desktop.

Exercise 2: Connect to the Data Science VM

In this exercise, you will connect remotely to the Ubuntu desktop in the VM that you created in the previous exercise. To do so, you need a client that supports Xfce, which is a lightweight desktop environment for Linux.

1. If you don't already have an Xfce client installed, download the X2Go client and install it now. X2Go is a free and open-source Xfce solution that works on a variety of operating systems. The instructions in this exercise assume you are using X2Go, but you can use any client as long as it supports Xfce.

2. Return to the Azure Portal and the blade for the resource group containing the Data Science VM. Then click the VM.



Subscription name (change)
[redacted]

Subscription ID
[redacted]

Deployments
1 Succeeded

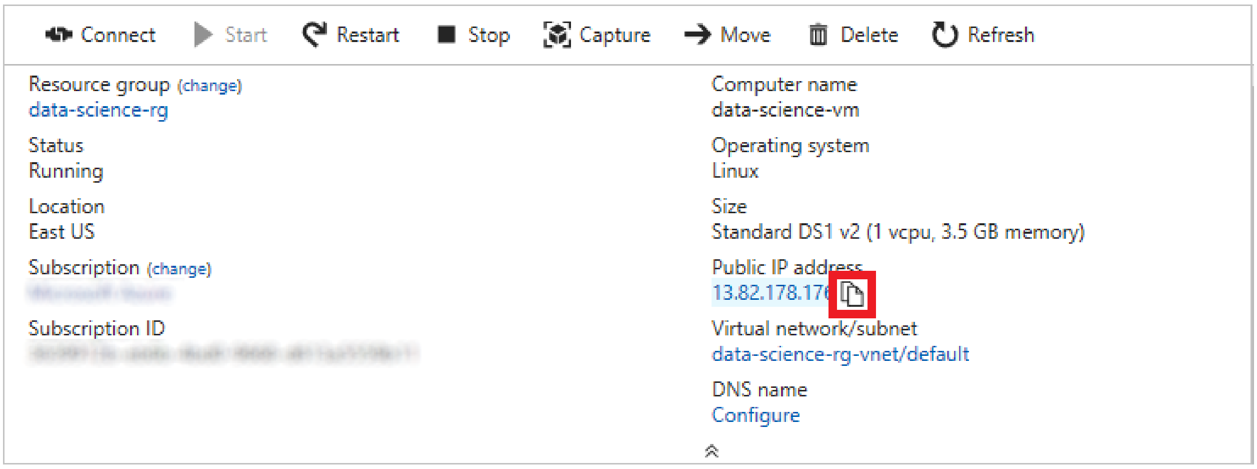
Filter by name...
All types
All locations
No grouping

9 items

NAME	TYPE	LOCATION
datasciencergdiag317	Storage account	East US
data-science-rg-vnet	Virtual network	East US
data-science-vm	Virtual machine	East US
data-science-vm_lun_0_2_b01aaca30fc34977bf44...	Disk	East US
data-science-vm_OsDisk_1_0d954c9a28b341ee8f0...	Disk	East US
data-science-vm653	Network interface	East US
data-science-vm-ip	Public IP address	East US
data-science-vm-nsg	Network security group	East US
shutdown-computevm-data-science-vm	Microsoft.DevTestLab/sch...	East US

Opening the Data Science VM

3. Hover over the IP address shown for the VM and click the **Copy** button that appears to copy the IP address to the clipboard.



Resource group (change)
data-science-rg

Status
Running

Location
East US

Subscription (change)
[redacted]

Subscription ID
[redacted]

Computer name
data-science-vm

Operating system
Linux

Size
Standard DS1 v2 (1 vcpu, 3.5 GB memory)

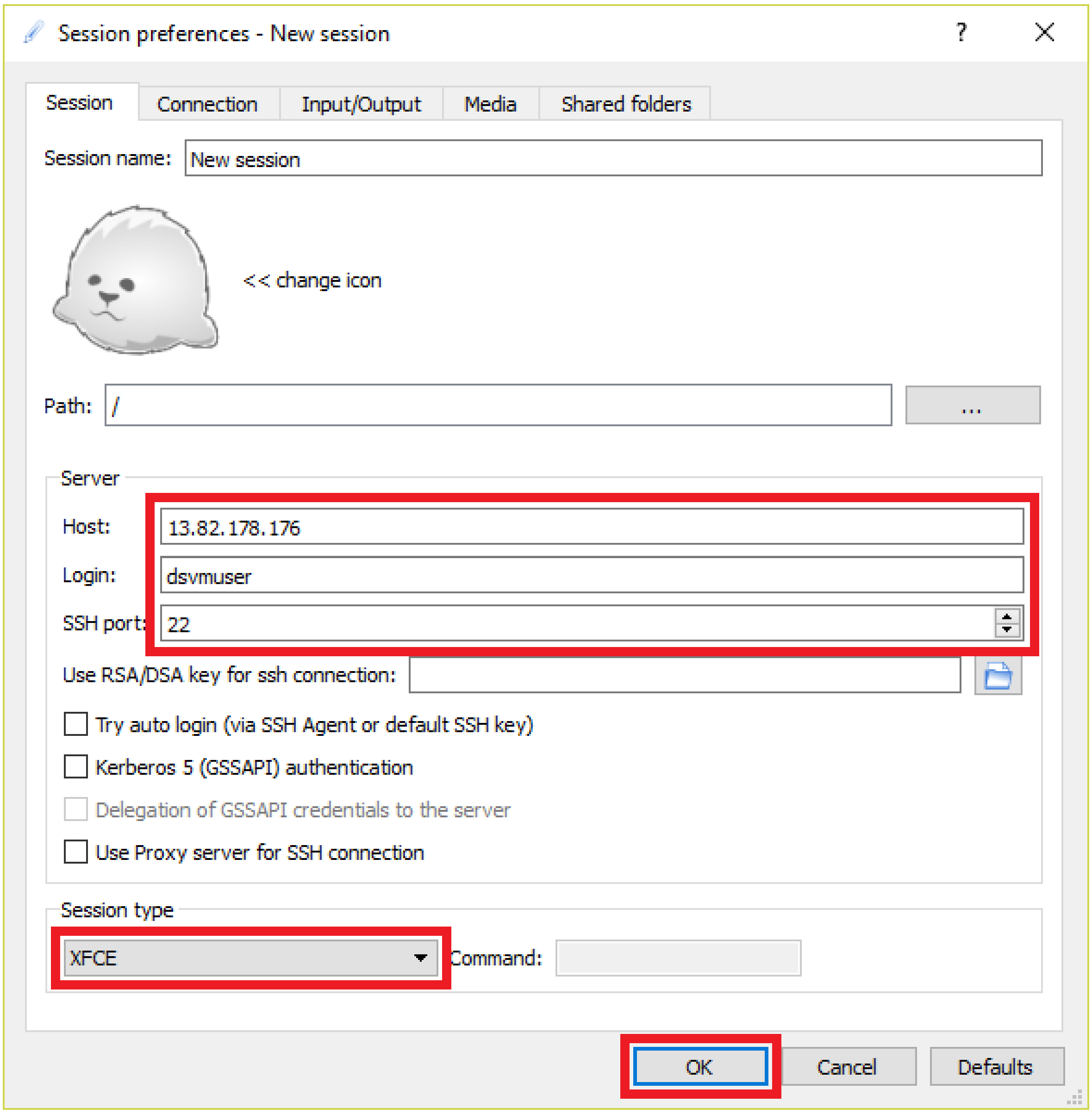
Public IP address
13.82.178.176

Virtual network/subnet
data-science-rg-vnet/default

DNS name
Configure

Copying the VM's IP address

4. Start the X2Go client and connect to the Data Science VM at the IP address that's on the clipboard using the user name you specified in the previous exercise. Connect via port **22** (the standard port used for SSH connections), and specify **XFCE** as the session type.



Session preferences - New session

Session name: New session

Path: /

Server

Host: 13.82.178.176

Login: dsvmuser

SSH port: 22

Use RSA/DSA key for ssh connection:

□ Try auto login (via SSH Agent or default SSH key)

□ Kerberos 5 (GSSAPI) authentication

□ Delegation of GSSAPI credentials to the server

□ Use Proxy server for SSH connection

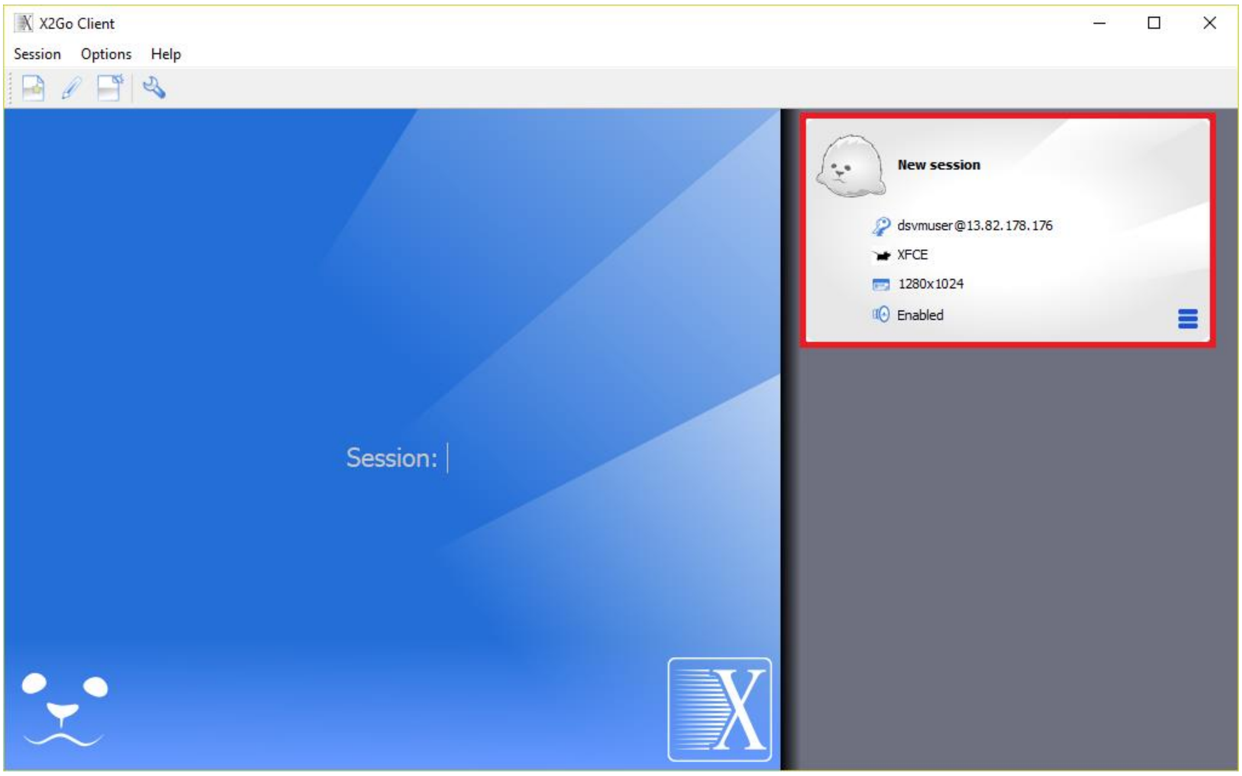
Session type

XFCE

Command:

Connecting with X2Go

5. In the **New session** panel on the right, select the resolution that you wish to use for the remote desktop. Then click the **New session** panel.



New session

dsvmuser@13.82.178.176

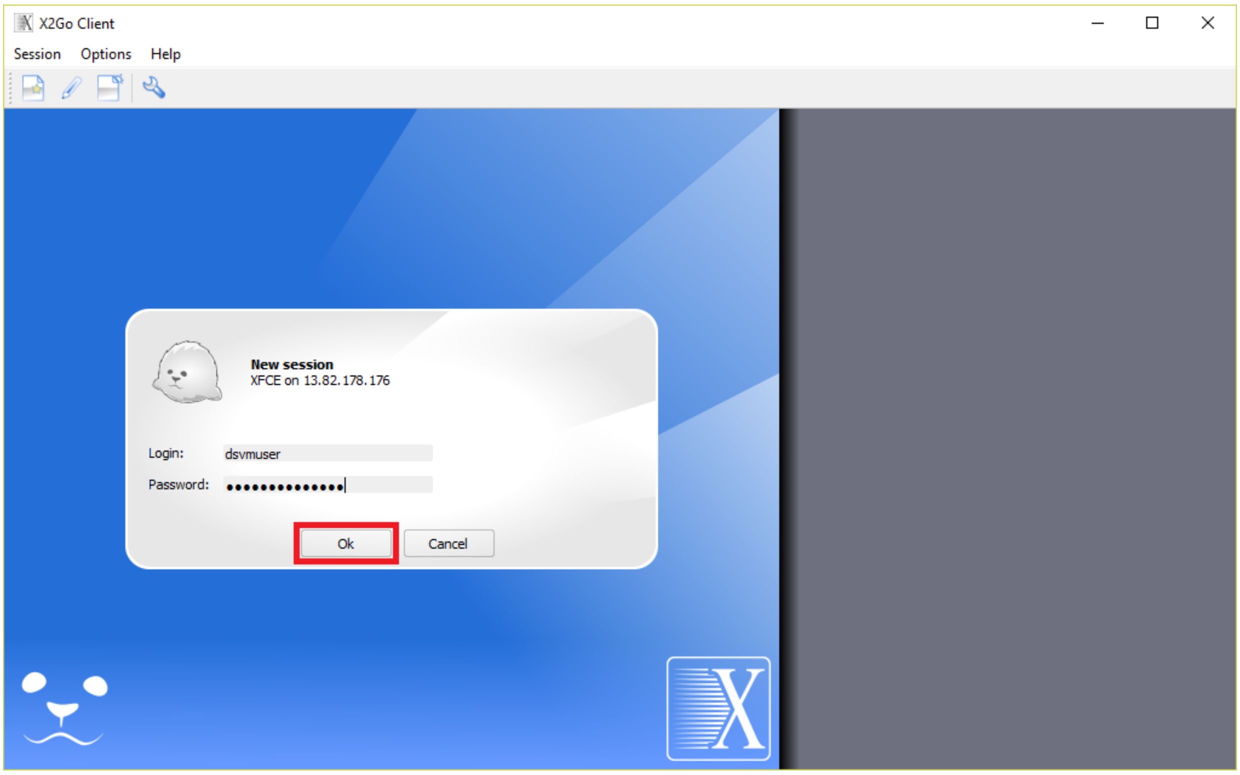
XFCE

1280x1024

Enabled

Starting a new session

6. Enter the password you specified in Exercise 1, and then click the **OK** button. If asked if you trust the host key, answer **Yes**. Also ignore any error messages saying the "SSH daemon could not be started."



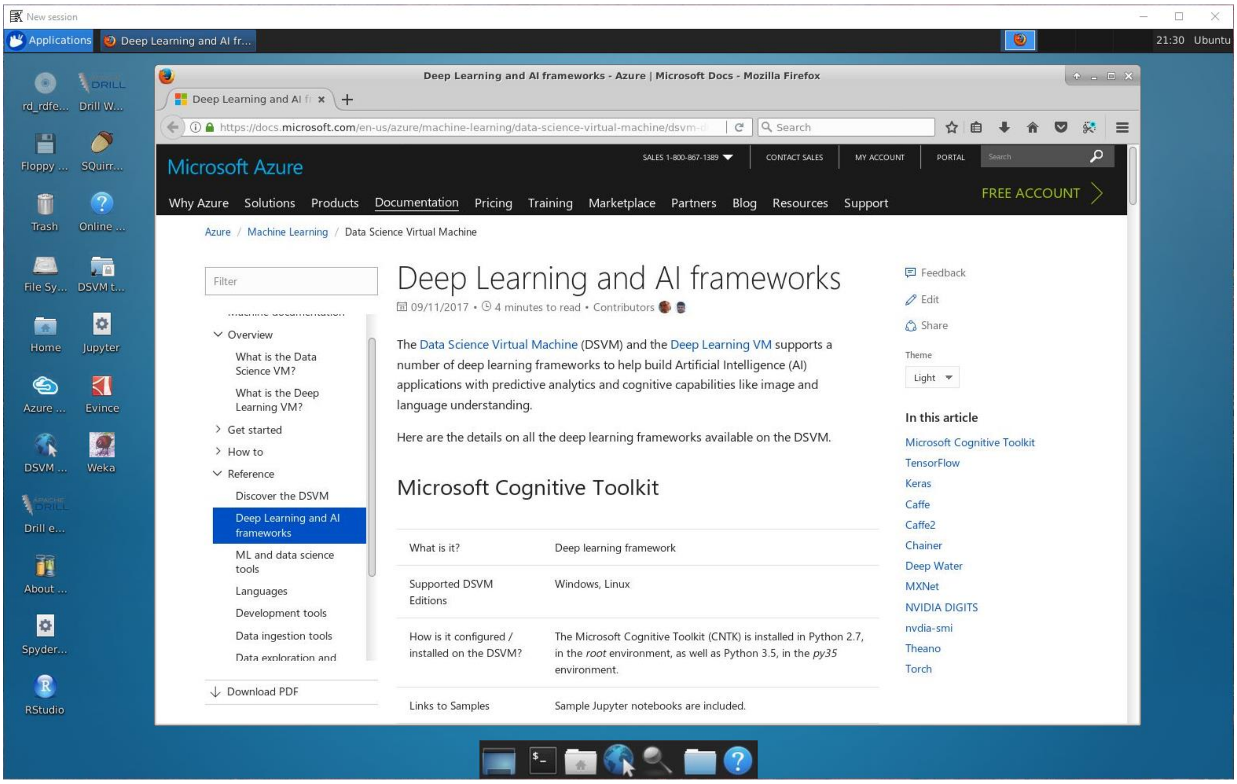
New session
XFCE on 13.82.178.176

Login: dsvmuser

Password: ••••••••••••••

Logging into the VM

7. Wait for the remote desktop to appear and confirm that it resembles the one below.



Deep Learning and AI frameworks

09/11/2017 • 4 minutes to read • Contributors

The Data Science Virtual Machine (DSVM) and the Deep Learning VM supports a number of deep learning frameworks to help build Artificial Intelligence (AI) applications with predictive analytics and cognitive capabilities like image and language understanding.

Here are the details on all the deep learning frameworks available on the DSVM.

Microsoft Cognitive Toolkit

What is it?	Deep learning framework
Supported DSVM Editions	Windows, Linux
How is it configured / installed on the DSVM?	The Microsoft Cognitive Toolkit (CNTK) is installed in Python 2.7, in the *root* environment, as well as Python 3.5, in the *py35* environment.
Links to Samples	Sample Jupyter notebooks are included.

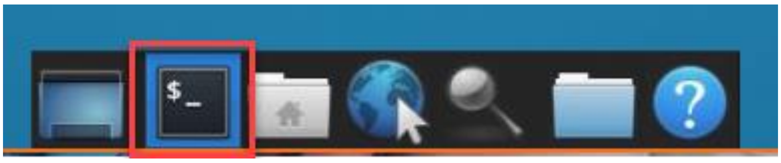
Connected!

Now that you are connected, take a moment to explore the shortcuts on the desktop. These are shortcuts to the numerous data-science tools preinstalled in the VM, which include Jupyter, R Studio, and the Microsoft Azure Storage Explorer, among others.

Exercise 3: Download a dataset and create a Jupyter notebook

In this exercise, you will import a dataset from Azure blob storage into the VM and load it into a Jupyter notebook. Jupyter is already installed in the VM and is accessible through the **Applications** menu or through the shortcut on the desktop. Jupyter notebooks are widely used in the data-science community to explore, transform, and visualize data. Notebooks are highly interactive, and since they can include executable code, they provide the perfect platform for manipulating data and building predictive models from it.

1. Click the Terminal icon at the bottom of the desktop to open a terminal window.

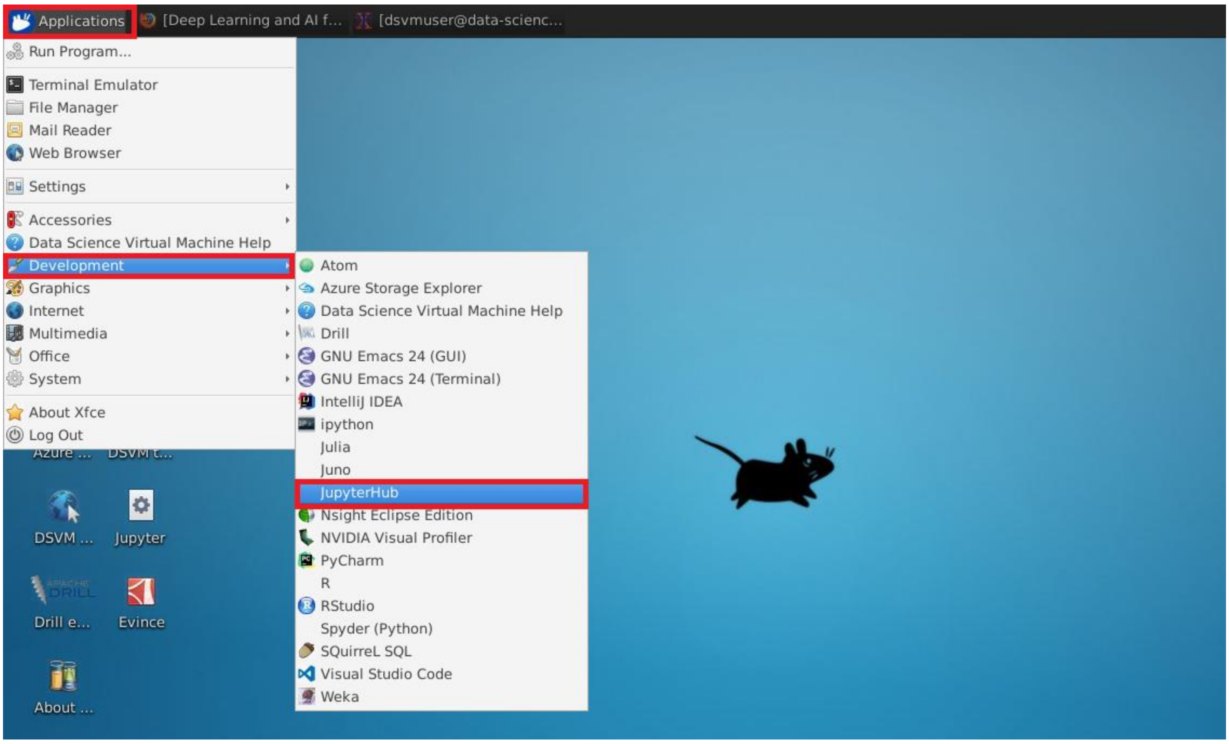


Opening a terminal window

2. Enter the following commands in the terminal window to create a "flights" subdirectory in the "notebooks" directory and download a dataset from Azure blob storage into the "flights" subdirectory:

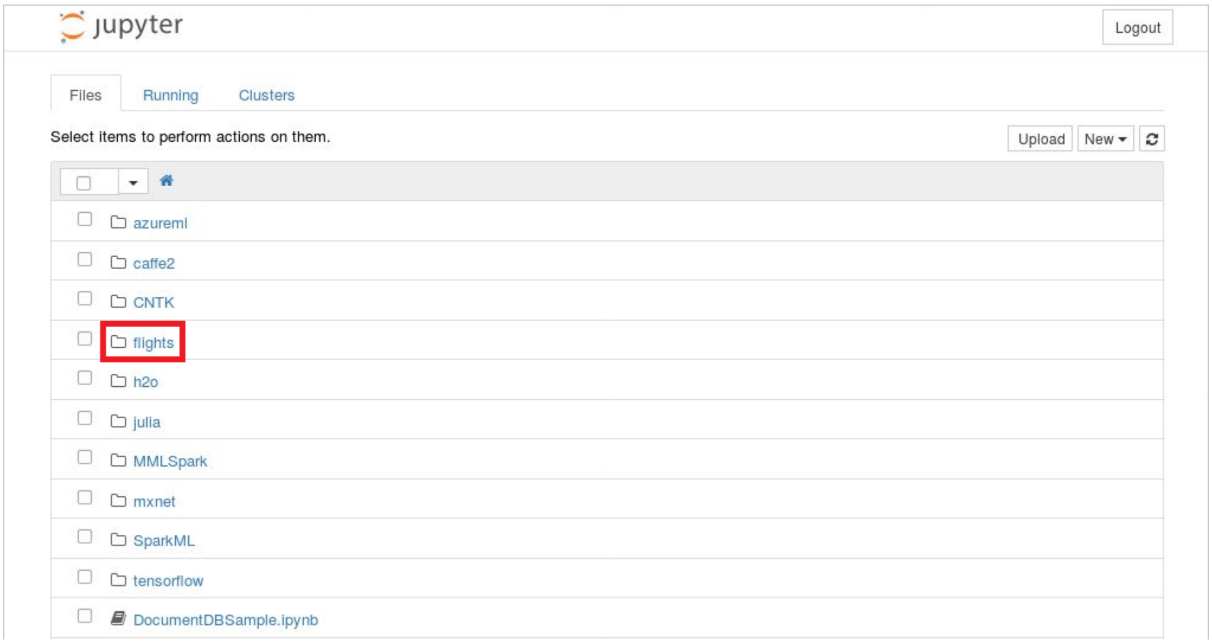
```
3. cd notebooks
4. mkdir flights
5. cd flights
   curl https://topcs.blob.core.windows.net/public/FlightData.csv --output
   flightdata.csv
```

6. Click **Applications** in the upper-left corner of the desktop. Then click **Development**, followed by **JupyterHub**.



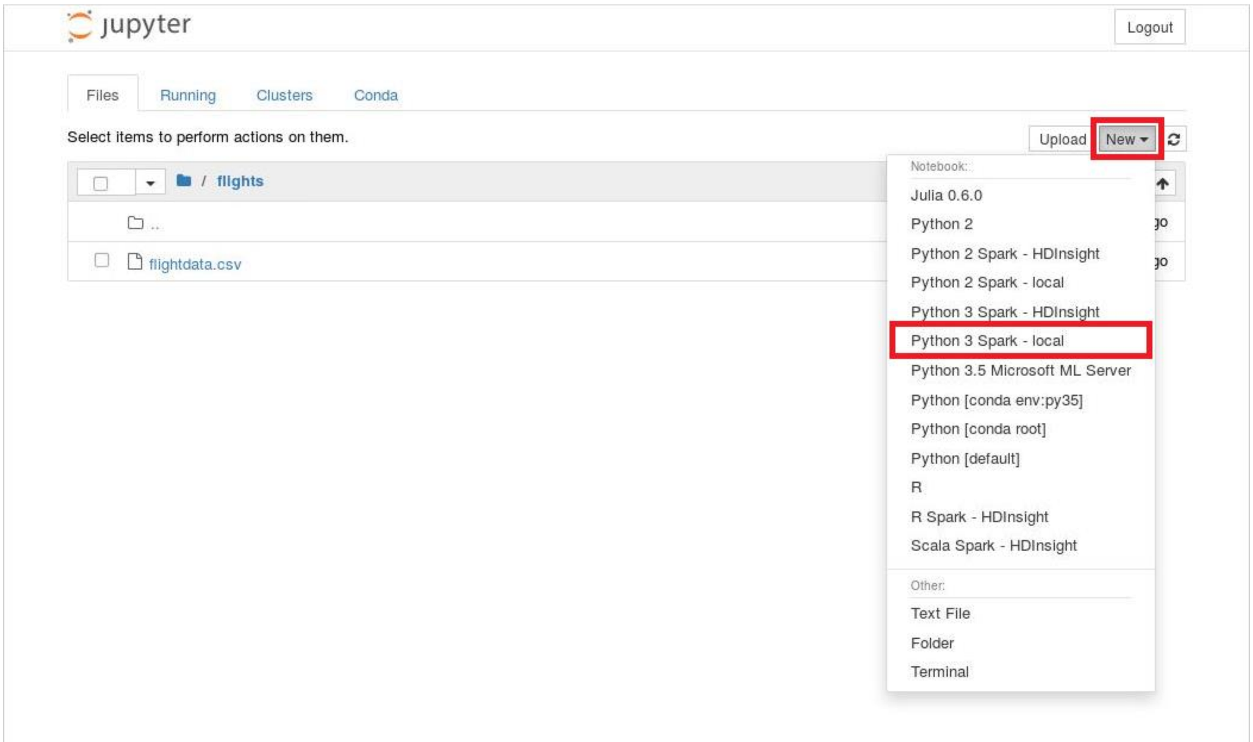
Launching JupyterHub

7. In the browser window that opens, click **flights** to open the "flights" directory.



Opening the "flights" directory

8. Confirm that **flightdata.csv** is present in the "flights" directory. Then click the **New** button and choose **Python 3 Spark - local** from the drop-down list to create a new Jupyter notebook with a Python 3 kernel.

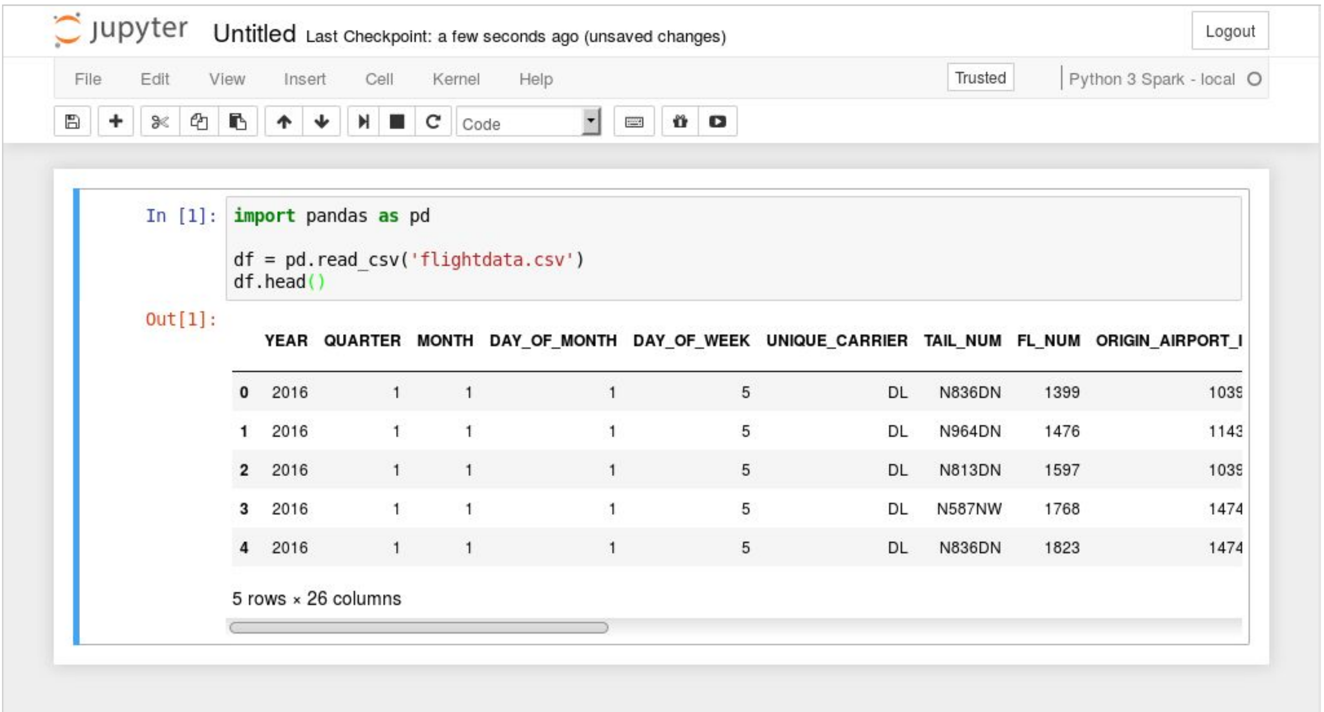


Creating a new Jupyter notebook

9. In the first cell of the notebook, enter the following Python code to load **flightdata.csv** and create a Pandas DataFramefrom it.

```
10. import pandas as pd
11.
12. df = pd.read_csv('flightdata.csv')
    df.head()
```

13. Select the **Run Cells** command from the **Cell** menu (or press **Ctrl+Enter**) to execute the Python code. Confirm that the output resembles the output below.



In [1]:
```
import pandas as pd

df = pd.read_csv('flightdata.csv')
df.head()
```

Out[1]:

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	UNIQUE_CARRIER	TAIL_NUM	FL_NUM	ORIGIN_AIRPORT_I
0	2016	1	1	1	5	DL	N836DN	1399	1039
1	2016	1	1	1	5	DL	N964DN	1476	1143
2	2016	1	1	1	5	DL	N813DN	1597	1039
3	2016	1	1	1	5	DL	N587NW	1768	1474
4	2016	1	1	1	5	DL	N836DN	1823	1474

5 rows × 26 columns

Loading the dataset

The DataFrame that you created contains on-time arrival information for a major U.S. airline. It has more than 11,000 rows and 26 columns. (The output says "5 rows" because DataFrame's head function only returns the first five rows.) Each row represents one flight and contains information such as the origin, the destination, the scheduled departure time, and whether the flight arrived on time or late. You will learn more about the data, including its content and structure, in the next lab.

14. Use the **File** -> **Save and Checkpoint** command to save the notebook.

15. Use the **File** -> **Rename...** command to name the notebook "FlightData."

If you check the "flights" directory, you should find that it now contains a file named **FlightData.ipynb** containing the Jupyter notebook you created. You will return to this notebook in the next lab and use it to prepare the data for use in a machine-learning model.