**DNN Workshop**

**Objectives**

In this hands-on lab, you will learn how to:

* Create a Linux Data Science Virtual Machine
* Connect to the DSVM via remote desktop
* Download the workshop content from GitHub
* Access JupyterHub remotely from your laptop

**Prerequisites**

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

* An active Microsoft Azure subscription. If you don't have one, [sign up for a free trial](http://aka.ms/WATK-FreeTrial).
* An [Xfce](https://xfce.org/) remote-desktop client such as [X2Go](https://wiki.x2go.org/doku.php/download:start)

**Exercises**

This hands-on lab includes the following exercises:

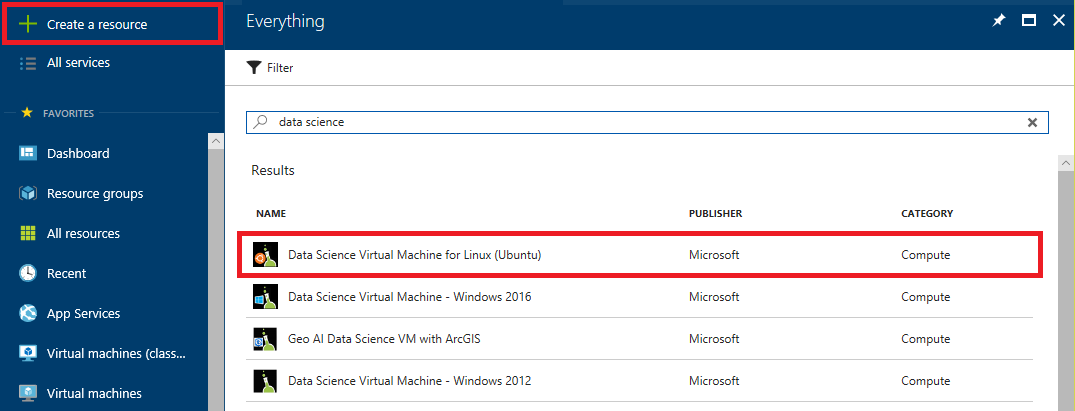
* [Exercise 1: Create an Ubuntu Data Science VM](#Excersize1)
* [Exercise 2: Connect to the Data Science VM via X2Go](#Excersize2)
* [Exercise 3: Install Azure CLI onto VM](#Excersize3)
* [Exercise 4: Clone (download) the workshop content from GitHub repo](#Excersize4)
* [Exercise 5: From your laptop connect to JupyterHub hosting the lab notebooks](#Excersize5)

Estimated time to complete this lab: **60** minutes.

**Exercise 1: Create an Ubuntu Data Science VM**

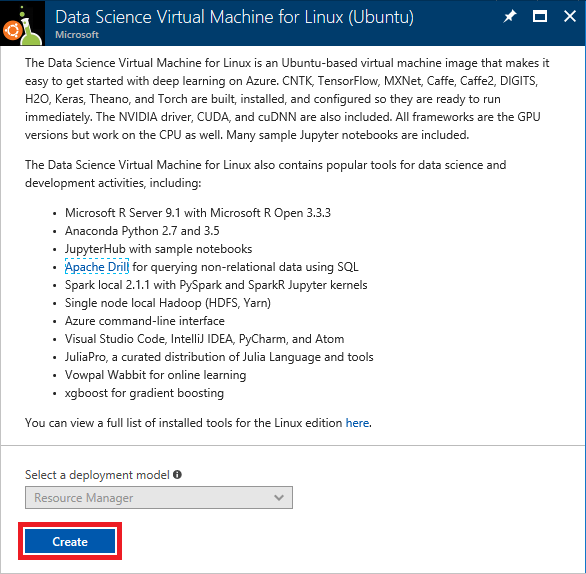
The Ubuntu Data Science Virtual Machine for Linux is a virtual-machine image that makes it easy to get started with data science. Multiple tools are already built, installed, and configured to get you up and running quickly. The NVIDIA GPU driver, [NVIDIA CUDA](https://developer.nvidia.com/cuda-downloads), and [NVIDIA CUDA Deep Neural Network](https://developer.nvidia.com/cudnn) library (cuDNN) are also included, as are [Jupyter](http://jupyter.org/) 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](https://portal.azure.com) 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.

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/new-data-science-vm-1.png)

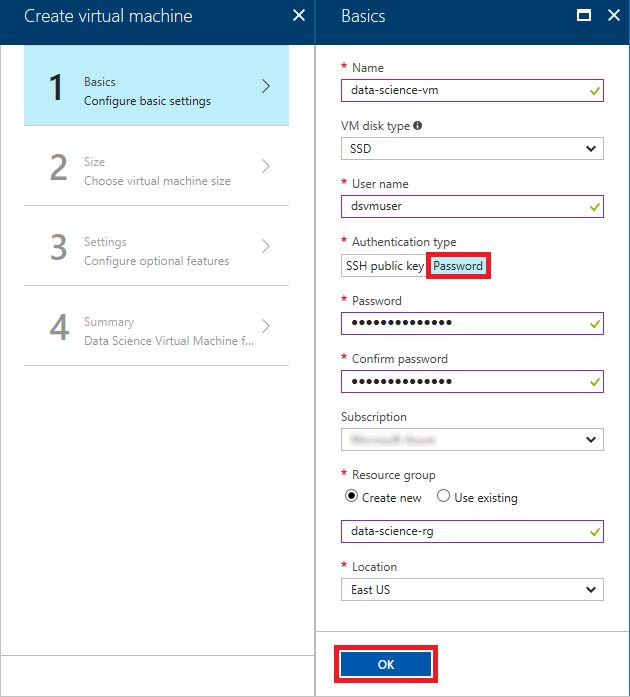
*Finding the Ubuntu Data Science VM*

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

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/new-data-science-vm-2.png)

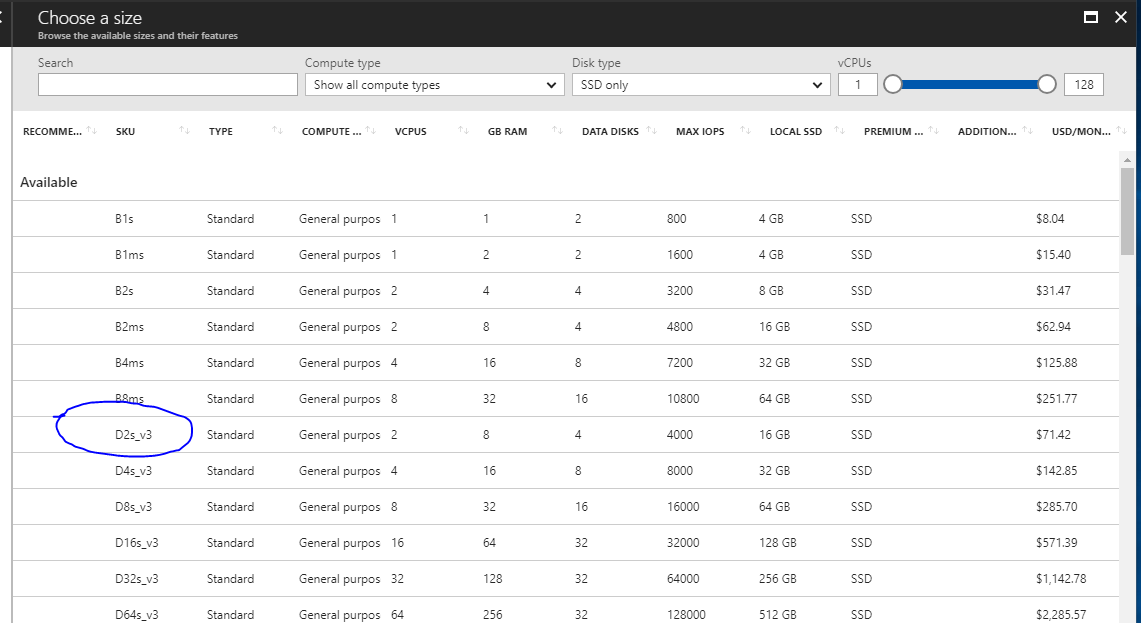
*Creating a Data Science VM*

1. 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**.

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/create-data-science-vm-1.png)

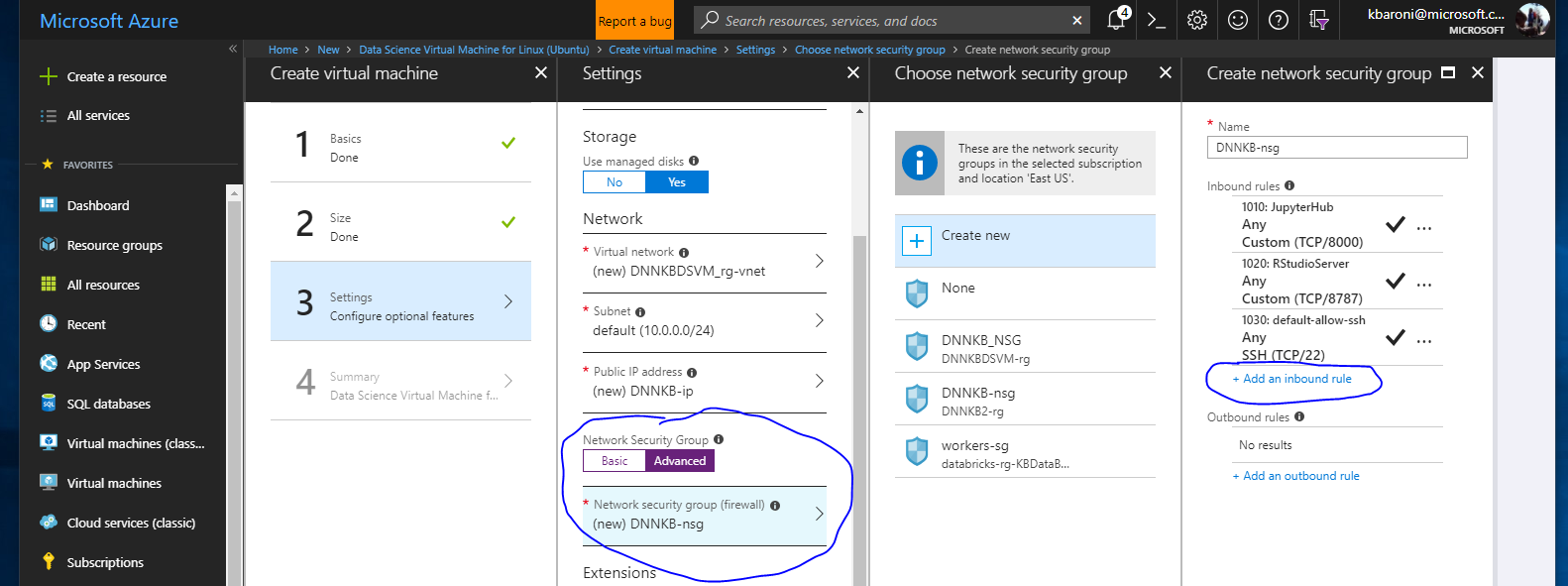
*Entering basic settings*

1. In the "Choose a size" blade, select **D2s\_v3**, which provides a low-cost way to experiment with Deep Learning VMs. Then click the **Select** button at the bottom of the blade.

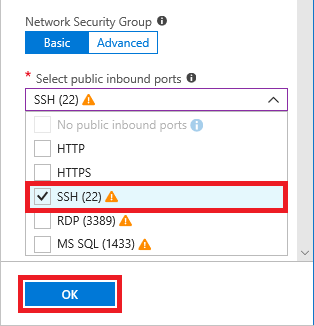


*Choosing a VM size*

1. In previous versions of DSVM, port 8000 was open by default when provisioning. It is no longer open by default, so you must explicitly create a firewall rule to allow access. You can do this during provisioning by: And create an inbound rule for port 8000 (will be required to access JupyterHub remotely) -> add the rule and continue provisioning the DSVM and submit for provisioning. Under Network Security Group, select ‘Advanced’ and add an inbound rule:

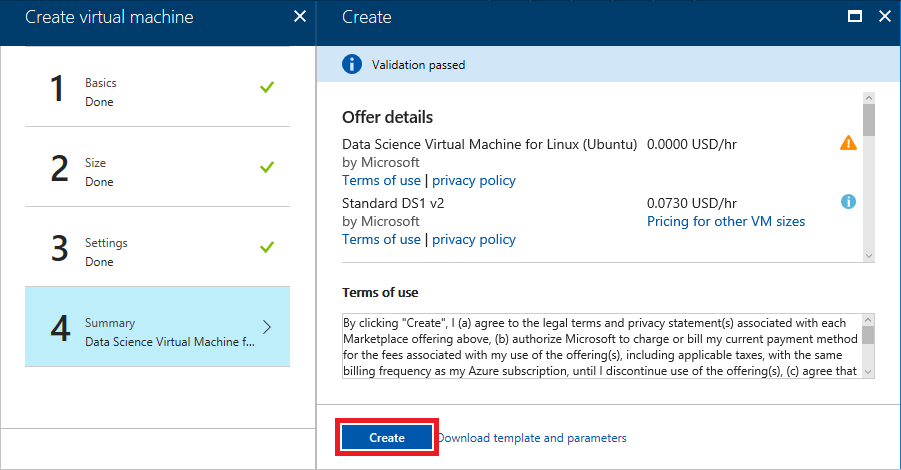


1. In the "Settings" blade, check **SSH (22)** in the list of inbound ports so clients can connect to the VM using the [Secure Shell](https://en.wikipedia.org/wiki/Secure_Shell) (SSH) protocol on port 22. Then click **OK**.

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/create-data-science-vm-3.png)

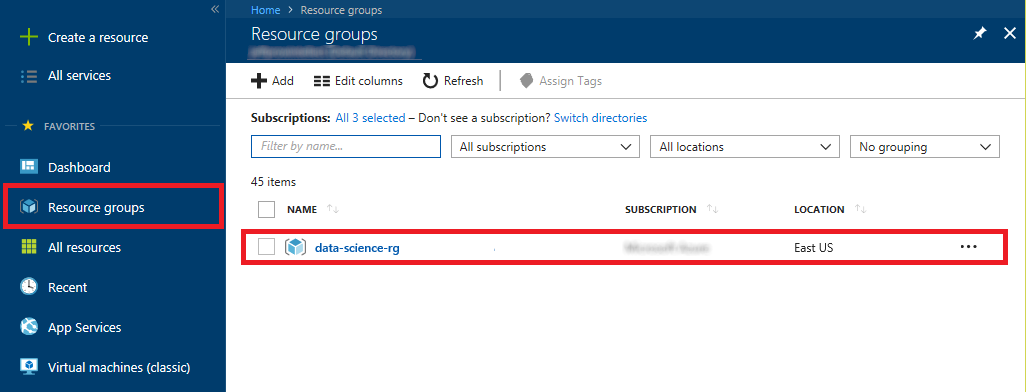
*Choose SSH*

1. In the "Create" blade, take a moment to review the options you selected for the VM, and click **Create** to start the VM creation process.

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/create-data-science-vm-4.png)

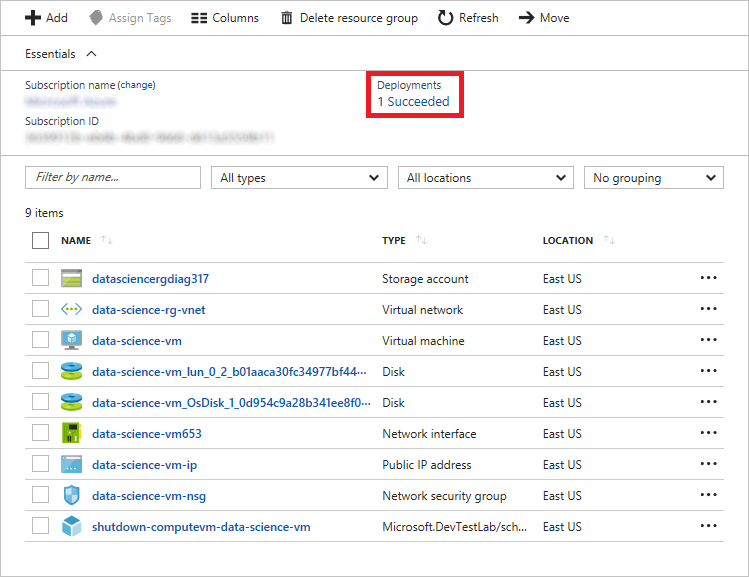
*Creating the VM*

1. 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.

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/open-resource-group.png)

*Opening the resource group*

1. 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.

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/deployment-succeeded.png)

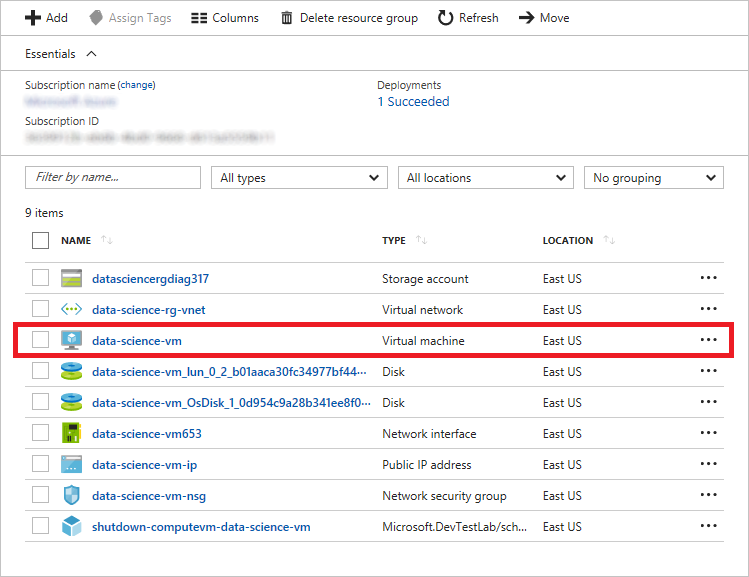
*Monitoring the deployment status*

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

**Exercise 2: Connect to the Data Science VM via X2Go**

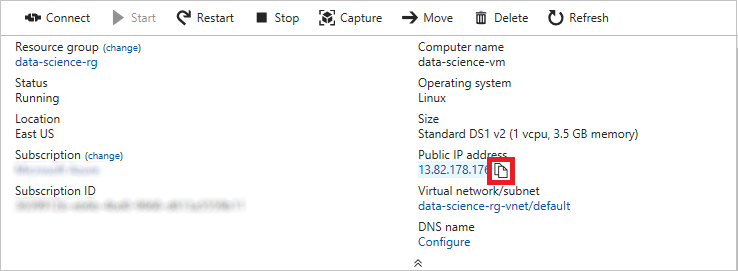
In this exercise, you will connect remotely to the Ubuntu desktop in the VM that you created in the previous exercise. To do so, you need a client that supports [Xfce](https://xfce.org/), which is a lightweight desktop environment for Linux.

1. If you don't already have an Xfce client installed, download the [X2Go client](https://wiki.x2go.org/doku.php/download:start) 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 if 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.

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/open-data-science-vm.png)

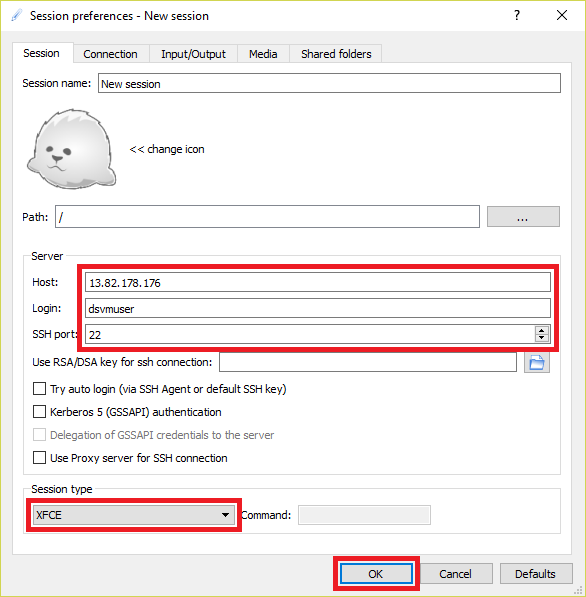
*Opening the Data Science VM*

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

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/copy-ip-address.png)

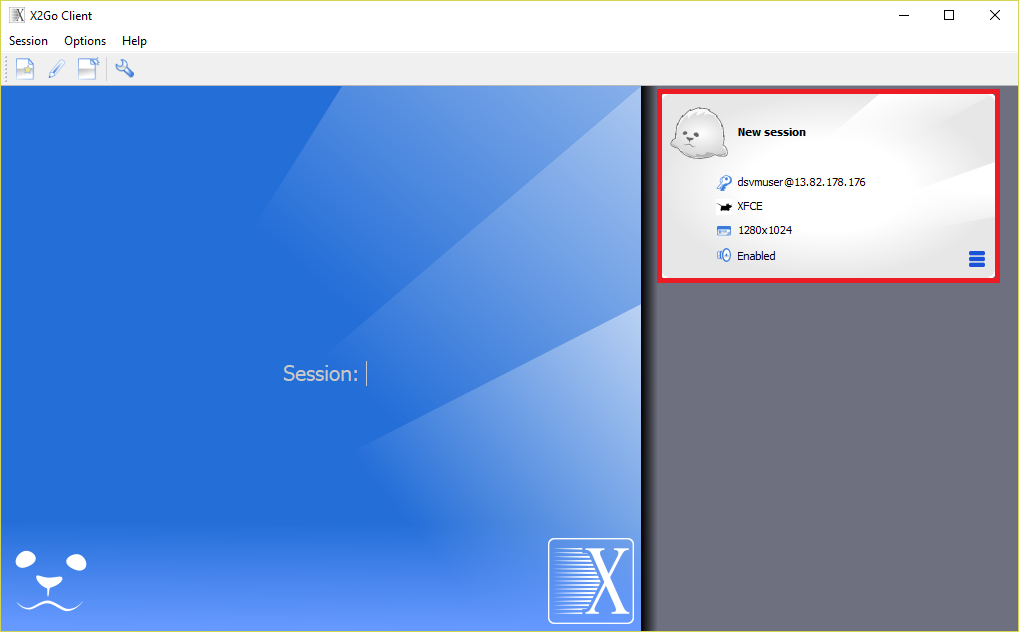
*Copying the VM's IP address*

1. 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.

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/new-session-1.png)

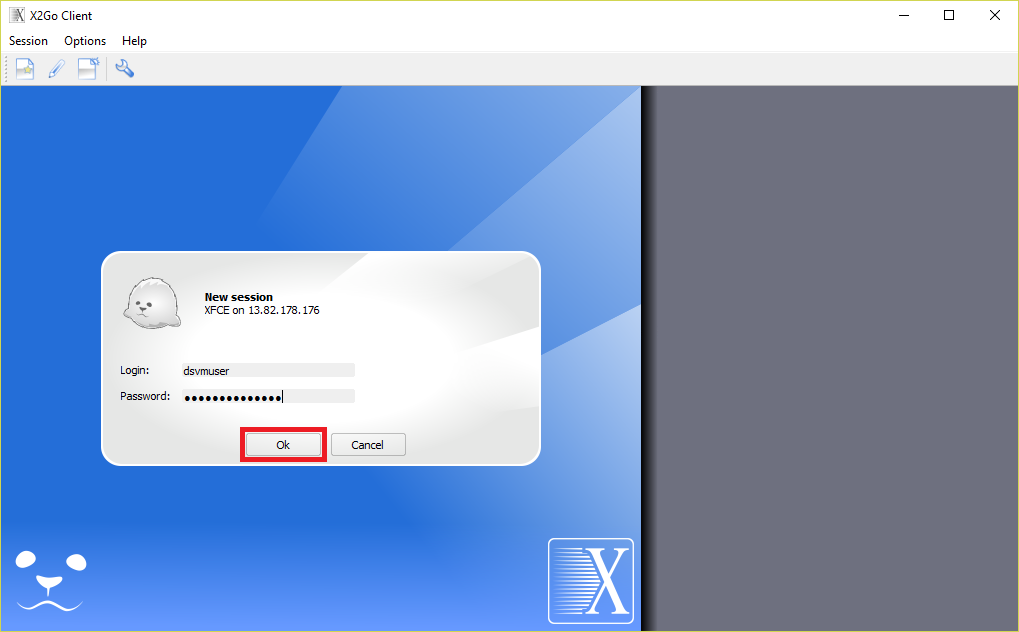
*Connecting with X2Go*

1. 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.

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/new-session-2.png)

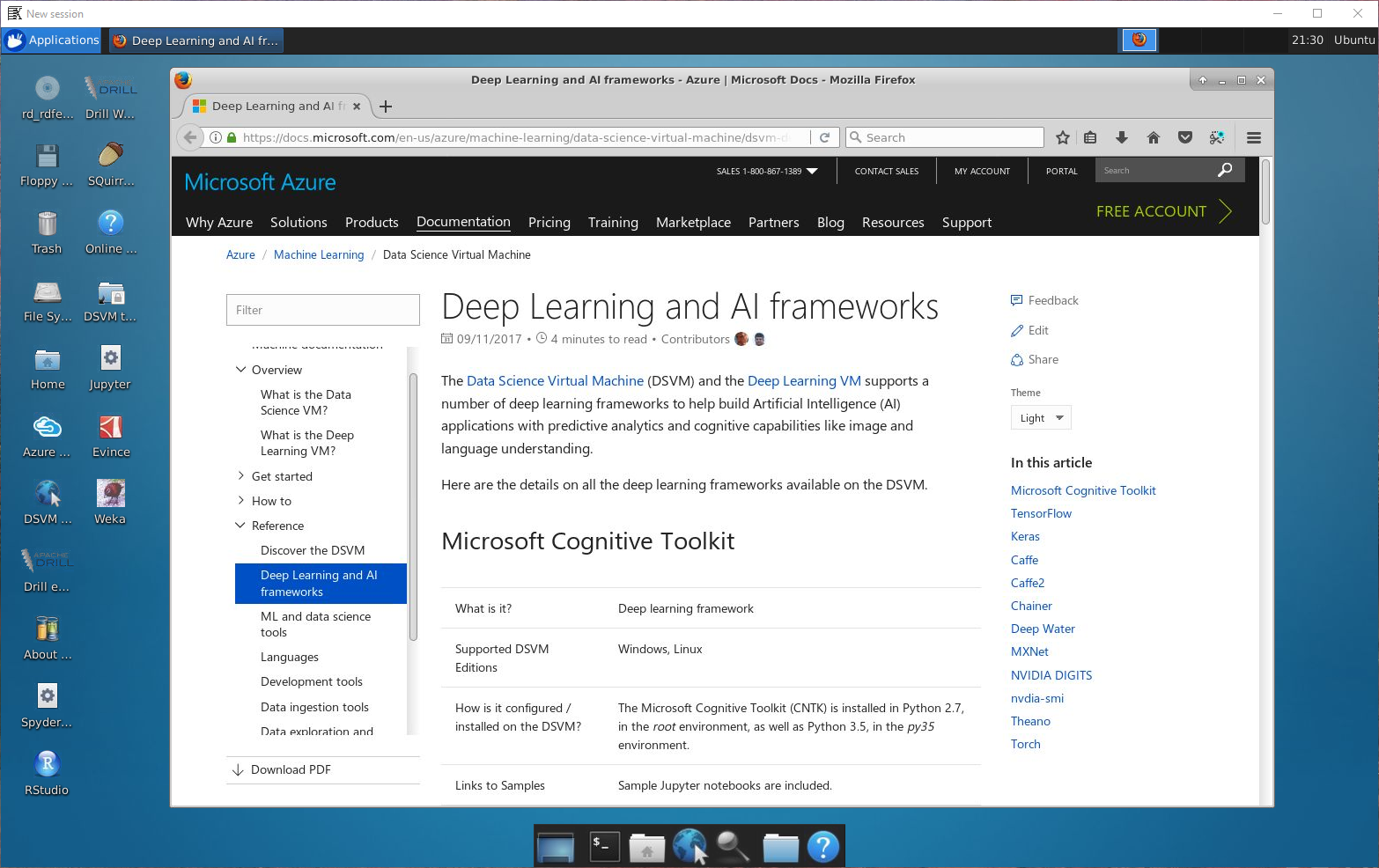
*Starting a new session*

1. Enter the password you specified in [Exercise 1](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/readme.md#Exercise1), 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."

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/new-session-3.png)

*Logging into the VM*

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

[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/ubuntu-desktop.png)

*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](http://jupyter.org/), [R Studio](https://www.rstudio.com/), and the [Microsoft Azure Storage Explorer](https://azure.microsoft.com/en-us/features/storage-explorer/), among others.

**Exercise 3: Install Azure CLI onto VM**

To download Azure CLI, follow the instructions here: <https://docs.microsoft.com/en-us/cli/azure/install-azure-cli> (I used apt for Ubuntu and followed the instructions here: <https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-apt?view=azure-cli-latest>.

Copy and paste the commands into the terminal emulator, then from the command line, run

az login

and you will be prompted to open a browser and activate your session:



Once the session is activated, verify by running

az account show -o table

az account list

**Exercise 4: Clone (download) the workshop content from GitHub rep**

In this exercise, you will download the content from the lab repo on GitHub and load a notebook into [JupyterHub](https://jupyterhub.readthedocs.io/en/stable/). JupyterHub is already installed in the VM and is accessible through your remote browser and is what we will use to serve Jupyter notebooks. 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.

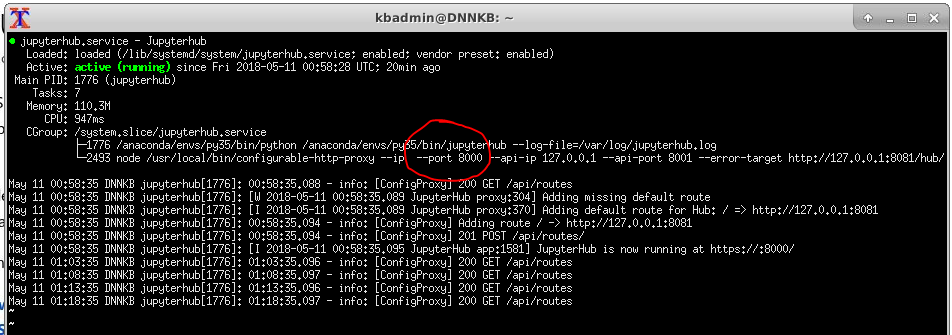
[](https://github.com/Microsoft/computerscience/blob/6fe13af1b7c684b47db404ce9fd06c4542bb9147/Labs/Deep%20Learning/200%20-%20Machine%20Learning%20in%20Python/1%20-%20Ingest/Images/open-terminal.png)

*Opening a terminal window*

1. Enter the following command in the terminal window to confirm JupyterHub is running:

Systemctl status jupyterhub

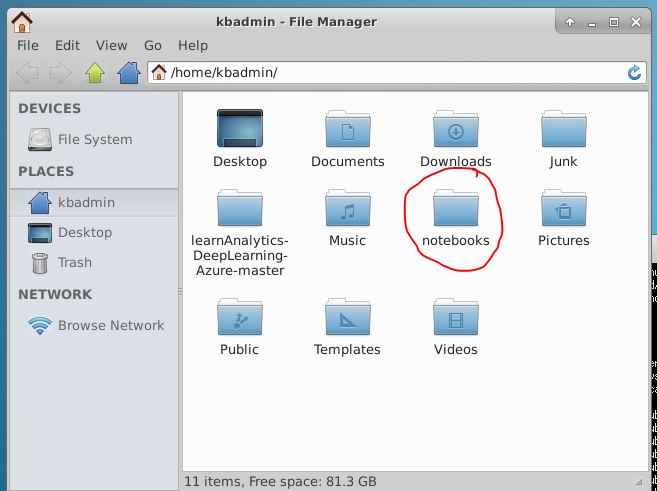
Notice the jupyterhub service is running on port 8000 and close the command window.



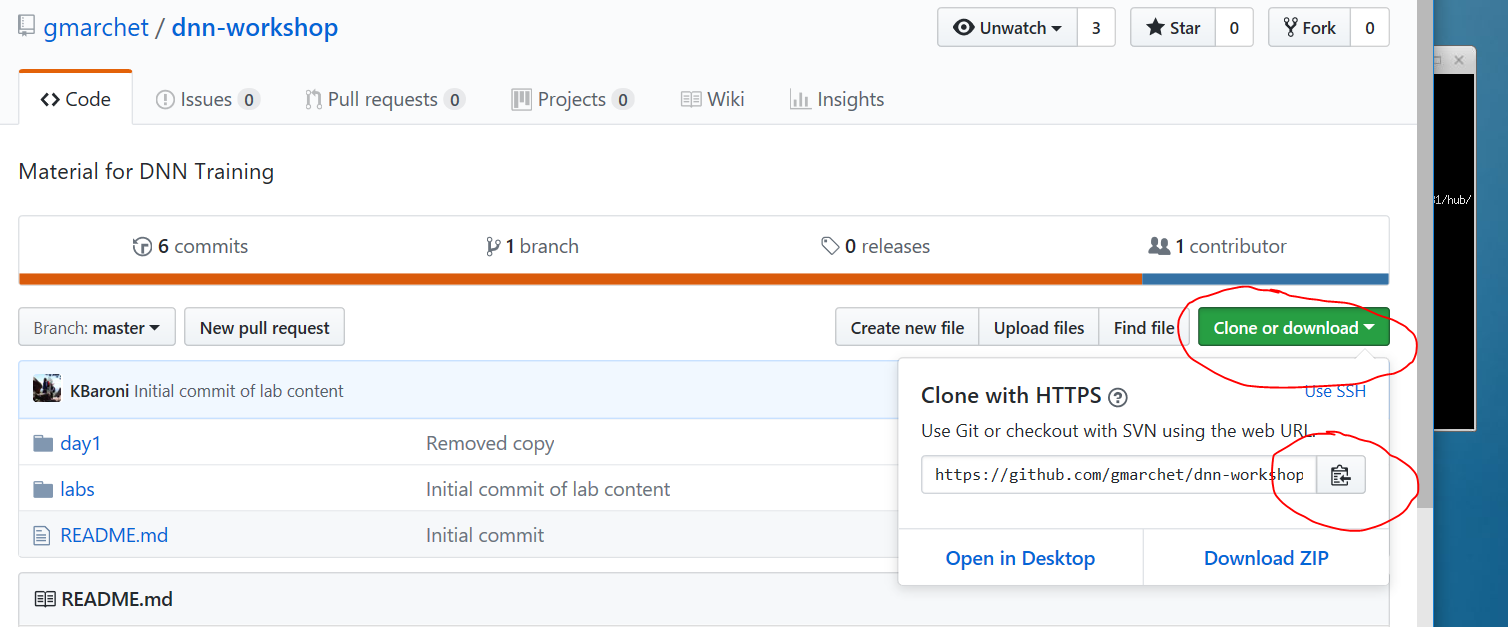
1. Click on the file manager icon at the bottom of the desktop to bring it up:



Select *Open Folder* and click on the **notebooks** folder. You will be downloading the git repo to the ***notebooks*** folder.



1. In this step we are going to clone the lab repository in GitHub. Bring up a browser and navigate to the lab’s git repository here: <https://github.com/gmarchet/dnn-workshop> and click on ‘Clone or download’. Select the icon to copy the web URL.

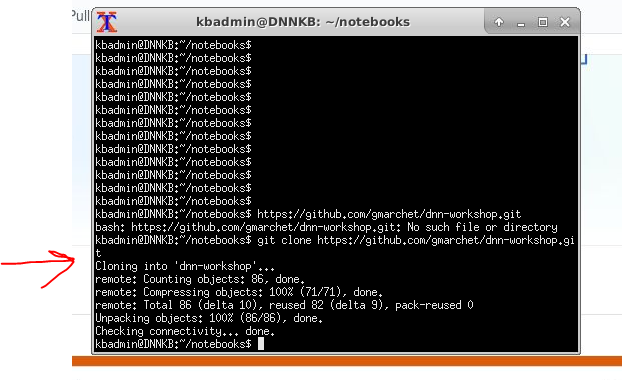


Bring up a new command window and type in

cd notebooks and then type in

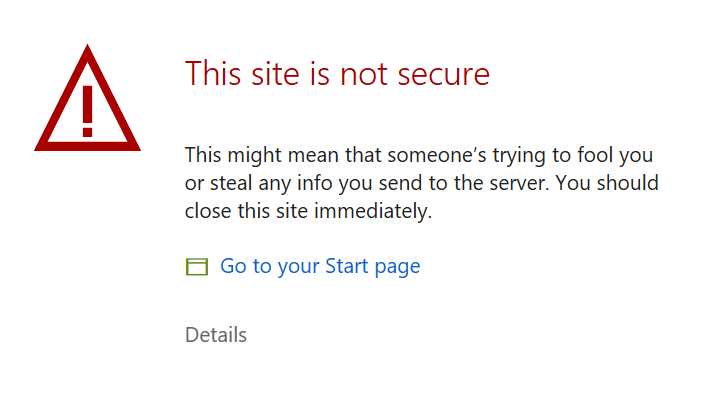
git clone <paste in the URL from github>

This will clone the repo called *dnn-workshop*:

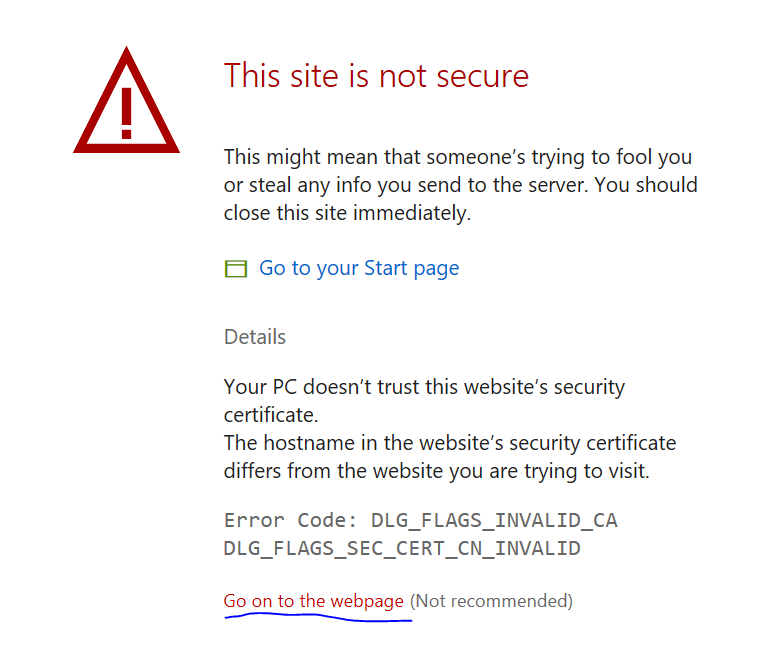


**Exercise 5: From your laptop connect to JupyterHub hosting the lab notebooks**

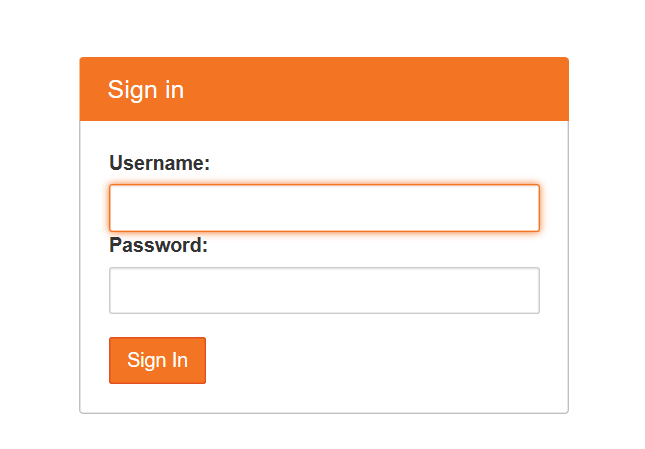
1. In this step we will access a notebook hosted on JupyterHub from a local laptop. To connect, browse on your laptop to <https://your-vm-ip:8000>. Expect to get a ‘certificate error’ looks something like this and click on ‘details’



then ‘Go on to the webpage’



1. Enter your VM credentials to access JupyterHub:



Successful connection will look like this:



1. Confirm you have access to the workshop notebooks. In Jupyter, find the *dnn-workshop* folder and browse to the labs folder. From here you will be running several of the workshop notebooks:

