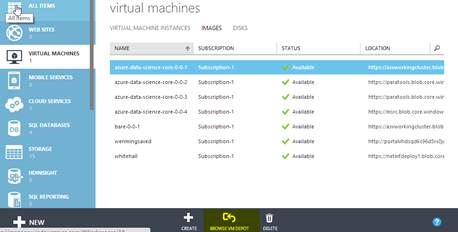
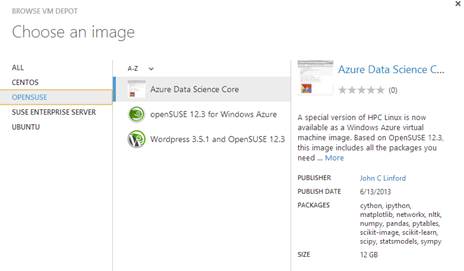
## Creating a VM from VM Depot

Log into your azure acct, go to Virtual machines and click on Images. Then click on browse VM depot.

cid:image001.jpg@01CE6879.9F9E0D30

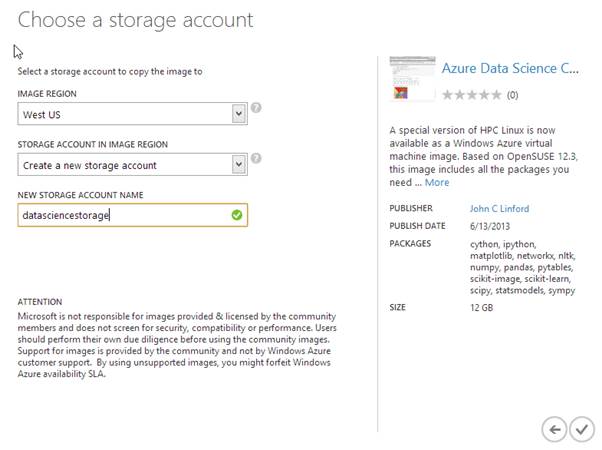


Pick the azure data science core VM under open SUSE. This VM was created using Suse studio, it is the easiest way for you to create, test a new customizable VM using their cloud based solution quickly.



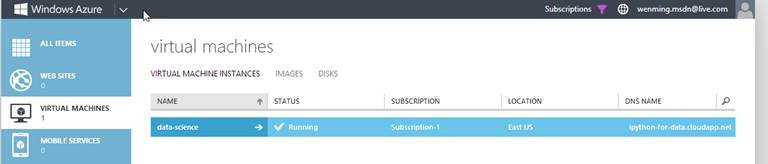
Click on Next (->)

Create a new storage acct, you should pick the same data center for both storage and VM creation.



Click on the check button, and that’s it.   Then you can ssh into the VM when it’s ready, it will say pending registration.  You will have to submit the URL of this VM running.  cid:image005.png@01CE6879.9F9E0D30

Click on Virtual machine instances again to see the actual VM running.



Click on the VM and the dash board to see the status and ssh information.



wenmingy@RPM-MSFT ~

$ ssh [ipython-for-data.cloudapp.net](http://ipython-for-data.cloudapp.net) -l azureuser

Password: Password you entered when the VM was created.

Last login: Wed Jun 12 19:32:30 2013 from ……..

Have a lot of fun...

azureuser@data-science:~> w

18:08:57 up 22:39,  1 user,  load average: 0.09, 0.04, 0.05

USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT

azureuse pts/0    c-67-164-167-117 18:08    1.00s  0.06s  0.00s w

azureuser@data-science:~>

azureuser@data-science:~>

azureuser@data-science:~>

azureuser@data-science:~> w

18:09:00 up 22:39,  1 user,  load average: 0.09, 0.04, 0.05

USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT

azureuse pts/0    c-67-164-167-117 18:08    0.00s  0.06s  0.00s w

azureuser@data-science:~>

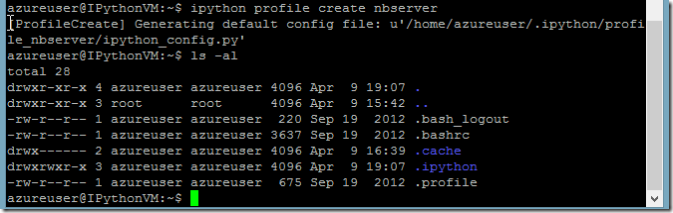
To run IPython, all the packages have been pre-installed, you can use my previous instructions from here:

## Configuring IPython notebook

Type: **ipython profile create nbserver**  to create a profile.  The command generates a default in your home directory under .ipython/profile\_nbserver/ipython\_config.py    Note that any directory starts with a “.” is a hidden directory in Linux. You must type **ls –al** to see it.

NOTE: You may also get ~/.config/ipython instead of ~/.ipython as the default dir.

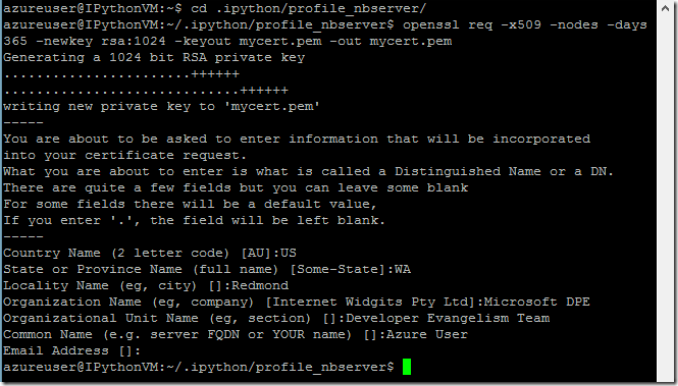
The .ipython directory is shown below in blue.

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-03-28-metablogapi/0815.image_5F00_7750D0CA.png)

Once we’ve created a profile, the next step is to create an SSL certificate and generate a password to protect the notebook webpage.

Type: **cd ~\.ipython\profile\_nbserver**  to switch into the profile we just created.

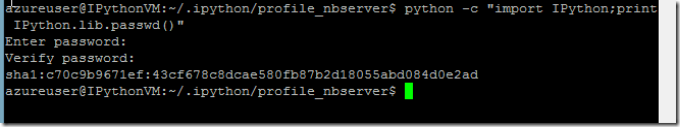
Then, type: **openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout mycert.pem -out mycert.pem** to create a certificate. Below is a sample session we used to create the certificate.

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-03-28-metablogapi/0827.image_5F00_1531E57D.png)

Since this is a self-signed certificate, the notebook your browser will give you a security warning. For long-term production use, you will want to use a properly signed certificate associated with your organization. Since certificate management is beyond the scope of this demo, we will stick to a self-signed certificate for now.

The next step is to create a password to protect your notebook.

Type: **python -c "import IPython;print IPython.lib.passwd()"   # password generation**

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-03-28-metablogapi/5125.image_5F00_3DD05184.png)

Next, we will edit the profile's configuration file, the ipython\_notebook\_config.py file in the profile directory you are in. This file has a number of fields and by default all are commented out. You can open this file with any text editor of your liking, and you should ensure that it has at least the following content, you may use either the Unix vi editor or nano which would be easier for beginners.

Make sure you make a copy of the sha1:c70c9b9671ef:43cf678c8dcae580fb87b2d18055abd084d0e2ad  string you got from the python password generator line above.

Type: **nano ipython\_config.py**

This will go into the editor, copy the appropriate line into your editor.  Note # is the comment sign for Python.

c = get\_config()

# This starts plotting support always with matplotlib

c.IPKernelApp.pylab = 'inline'

# You must give the path to the certificate file.

# If using a Linux VM:

c.NotebookApp.certfile = u'/home/azureuser/.ipython/profile\_nbserver/mycert.pem'

# Create your own password as indicated above

c.NotebookApp.password = u'sha1:c70c9b9671ef:43cf678c8dcae580fb87b2d18055abd084d0e2ad' **#use your own**

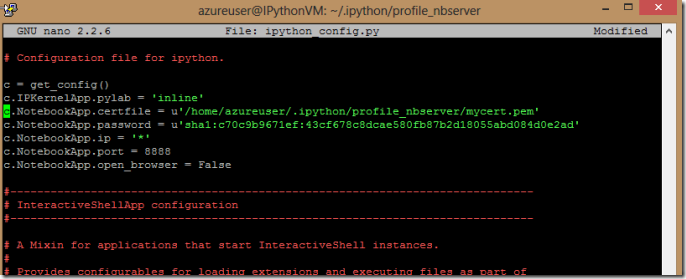
# Network and browser details. We use a fixed port (9999) so it matches

# our Windows Azure setup, where we've allowed traffic on that port

c.NotebookApp.ip = '\*'

c.NotebookApp.port = 8888

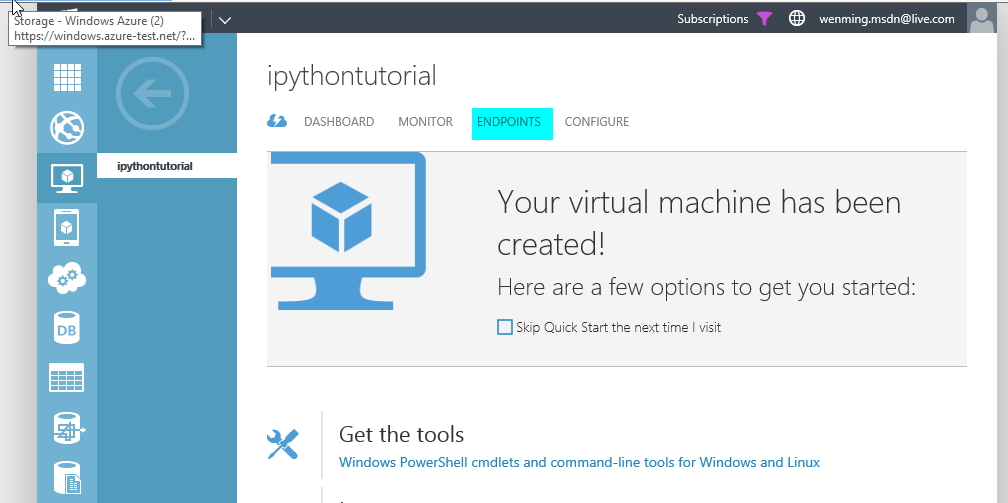
c.NotebookApp.open\_browser = False

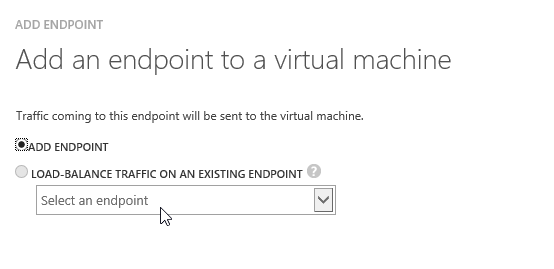
[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-03-28-metablogapi/2480.image_5F00_38ED9DC8.png)

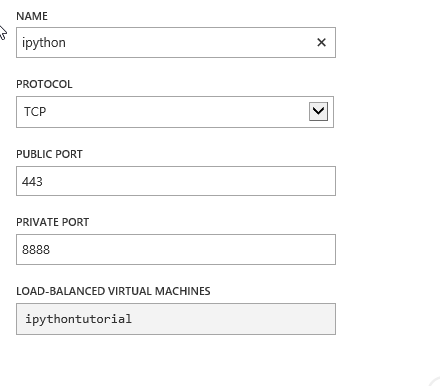
**Press control –X to exit nano and press Y to save the file.**

**Configure the Windows Azure Virtual Machines Firewall**

Go to Windows Azure portal => virtual machines => your VM => End points

****

****

****

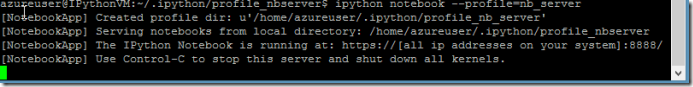
**Run the IPython Notebook**

At this point we are ready to start the IPython Notebook. To do this, navigate to the directory you want to store notebooks in and start the IPython Notebook Server:

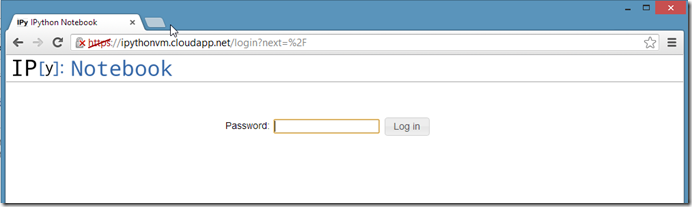
Type: **ipython notebook --profile=nbserver**

You should now be able to access your IPython Notebook at the address https://[Your Chosen Name Here].cloudapp.net.

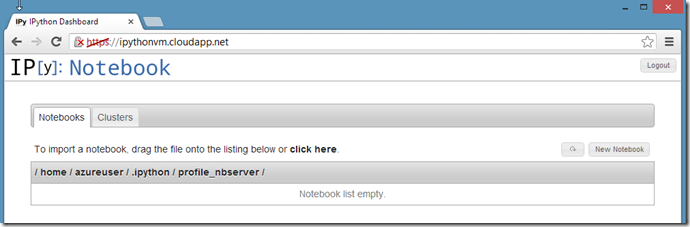
In our case it is:  [https://ipythonvm.cloudapp.net](https://ipythonvm.cloudapp.net/)

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-03-28-metablogapi/1488.image_5F00_2CEBAD94.png)

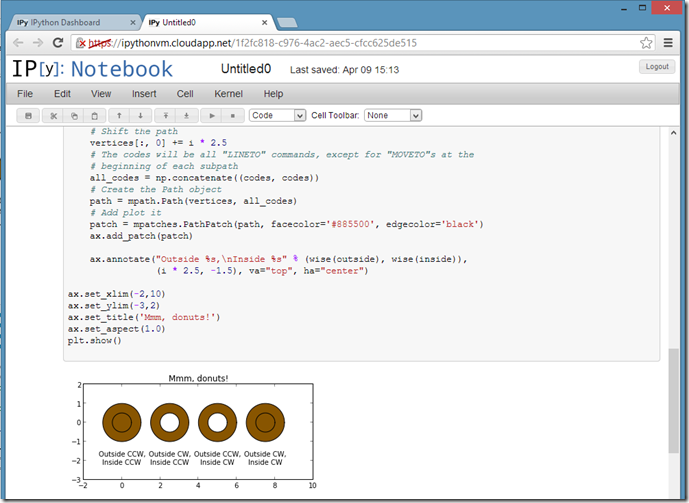
Type in the Password you set when you ran the **python -c "import IPython;print IPython.lib.passwd()"** command.

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-03-28-metablogapi/1401.image_5F00_03E39F49.png)

Once logged in, you should see an empty directory.  Click on **“New NoteBook”** to start.

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-03-28-metablogapi/6523.image_5F00_77E1AF14.png)

To reward your hard work, we’ll have IPython notebook plot a few donuts for us.  You can copy and paste the code from: <http://matplotlib.org/examples/api/donut_demo.html>  Please your cursor to the end of the last line, Press shift + Enter to run the code right after the last line.  If all goes well, you should see a set of 4 chocolate donuts almost instantly.

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-03-28-metablogapi/2500.image_5F00_47BA6451.png)

2. To download additional ipython tutorial materials:

<https://github.com/ipython/ipython-in-depth>.

Since we may still make updates to these materials between now and the tutorial, we recommend that you clone the git repo so that you can do a quick pull at the conference. But we will also bring copies of the entire materials on USB disks in case you don't download it in advance. Also note that if you don't use git regularly, it is NOT required for the tutorial at all, and you can download a plain zip file with the whole archive from this url:

<https://github.com/ipython/ipython-in-depth/archive/master.zip>

## Playing with Windows Azure Storage API

<http://www.windowsazure.com/en-us/develop/python/>

Detailed Installation instructions: <http://www.windowsazure.com/en-us/develop/python/common-tasks/install-python/>

Quick instruction: pip install azure

Installing the Windows Azure Client utility:

sudo npm install azure-cli -g

If you don’t have NODE.js installed, use the instruction here: <https://github.com/joyent/node/wiki/Installing-Node.js-via-package-manager>