Jupyter on Azure (Work in Progress)

Sébastien Boisgérault, Mines ParisTech, under CC BY-NC-SA 4.0

July 27, 2017

Contents

Getting Started **1**

The Azure Platform **1**

Data Management **5**

Software Packages **6**

Networking **7**

Appendix – Conda Packages **7**

Getting Started

1. Create a Microsoft account if you don't already have one.
2. Go to the Microsoft Azure Notebooks web site and sign in.
3. Select "Libraries" in the navigation bar (libraries are groups of related notebooks) and create a new library named "Sandbox".
4. Create a new notebook named "My First Notebook.ipynb" in the Sandbox library, or upload an existing one. For this article, I will use a new Python 2.7 notebook.
5. Start the notebook.

The Azure Platform

To explore the Azure plaform hosting the Jupyter notebook, we will issue some shell commands; the simplest way to do that if from within a Python notebook

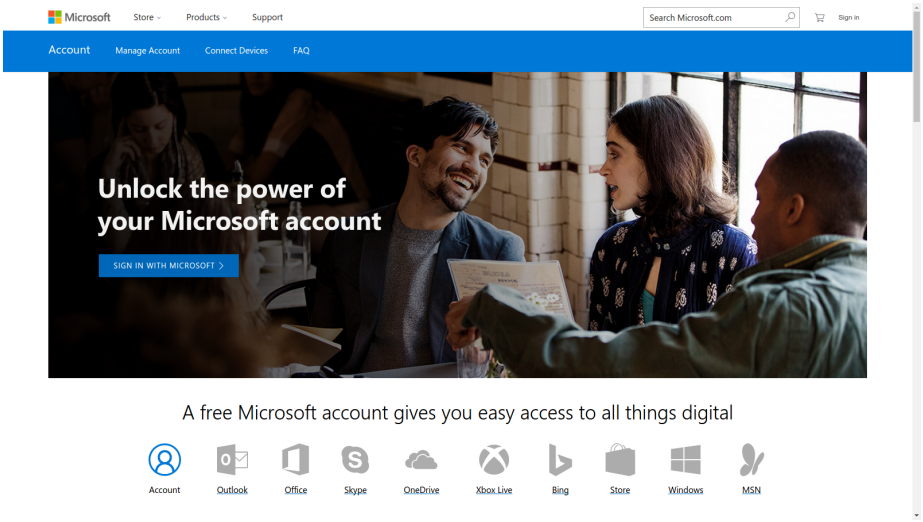

Figure 1: `https://account.microsoft.com/`

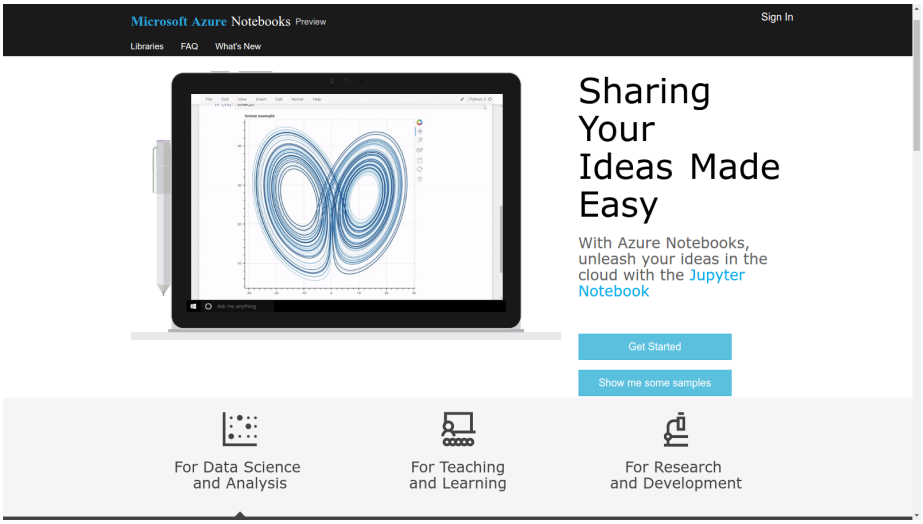

Figure 2: `https://notebooks.azure.com/`

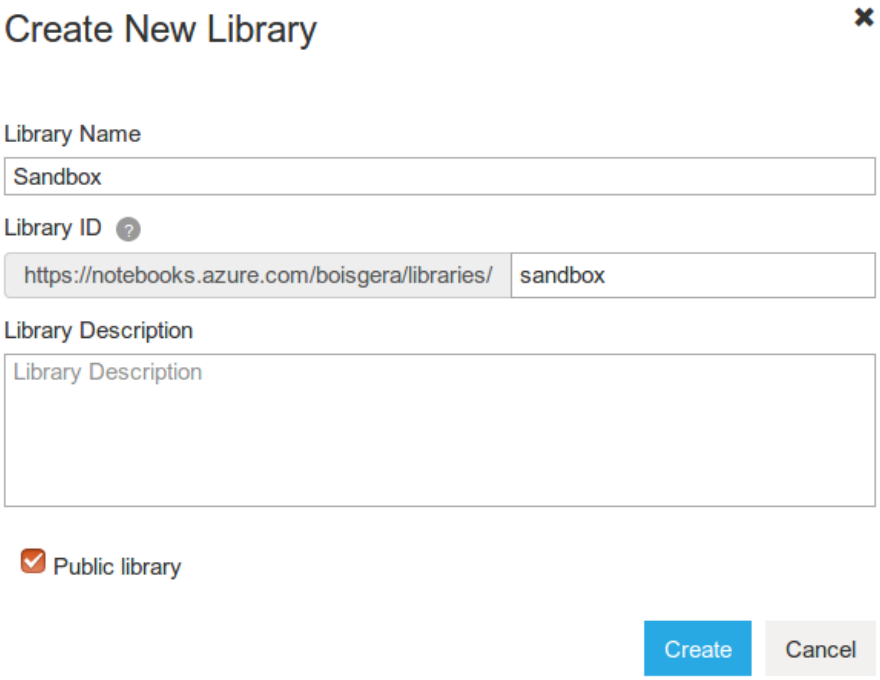

Figure 3:

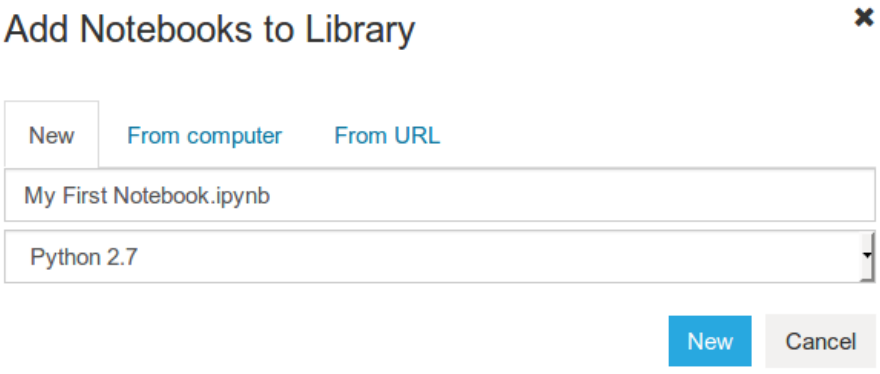

Figure 4:

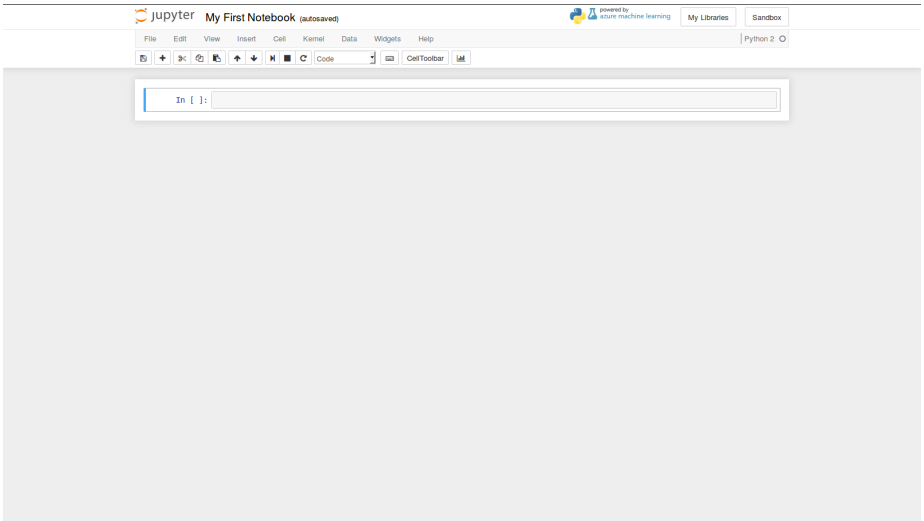

Figure 5:

is to type the command in a cell, prefixed with an exclamation point[1].

First of all, Azure notebooks are hosted on Linux (Debian-based) machine:

```
>>> import platform
>>> platform.system()
'Linux'
>>> platform.platform()
'Linux-4.4.0-81-generic-x86_64-with-debian-stretch-sid'
```

The distribution used is the latest LTS version of Ubuntu: Xenial Xerus.

```
>>> !cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04.2 LTS"
```

The notebook is actually running within a Docker container:

```
>>> import os.path
>>> os.path.isfile("/.dockerenv")
True
```

What it means concretely is that when you start – or restart – a notebook, you are likely to wait for a couple of seconds while Azure is provisioning a new

[1]Alternatively, you can open a full-fledged terminal. First you need to access the Jupyter dashboard (click on the Jupyter logo in the top-left corner of the notebook), then open the "New" drop down menu and select "Terminal".

container. As far as I can tell, container instances are shared between notebooks in the same library, but not across libraries.

The processor specs are:

```
>> !lscpu | grep "Model name"
Model name: Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz
```

PassMark gives this CPU a mark of 16.904; this is far better than the laptop I am working on, so I guess that performance is uniquely to be an issue.

Data Management

If you use the Jupyter Notebook App, which executes the notebooks on your computer, or if you have deployed a JupyterHub server, the input data that you use for your notebooks, the ouput data that they may produce and the notebooks themselves are in the same filesystem, probably organized into directories, one for each projects.

Things are different for Azure notebooks, where notebooks and data are handled separately. Notebook files (with the `.ipynb` extension) are stored permanently and associated to your Microsoft account; they can be managed from the Libraries view and/or from the dashboard. You can also download/upload them if you loke but this is not mandatory. However, you will not find them in the filesystem accessible in the notebook [2].

Your data files on the other hand – anything that is in the notebook filesystem – is ephemeral and will be lost when your library container is shut down. Consequently:

- If your notebook requires some input data, you need to check that it's available before you execute the notebook, or "rehydrate" your filesystem, upload this data again.
- If your notebook produces some output data that you want to keep, you need to download it.

For both steps there are several options available.

- You may upload/download your data manually from/to your computer: use the "Data" menu in the notebook navigation panel. Data may also be uploaded from a Dropbox account.
- To compress the files in the current directory into a `data.zip` archive – but anaconda Python distributions and hidden files – and upload the archive to `transfer.sh`, type in a cell:

[2] Actually there is a hidden `.library` directory, where *sometimes* you can find your notebook file, but not consistently AFAICT.

```
>>> !zip -q -r data.zip . -x "anaconda*" ".*"
>>> !curl --upload-file data.zip https://transfer.sh/data.zip
```

The last command prints an address, something like:

```
https://transfer.sh/yKwUl/data.zip
```

This is where your archive is located (and will be for 14 days). To download and unzip this archive, type:

```
>>> !curl https://transfer.sh/yKwUl/data.zip -o data.zip
>>> !unzip -o data.zip
```

Note that these commands work in Azure notebooks because `https://transfer.sh` is explicitly whitelisted (see Networking), so use specifically this service; other file sharing sites probably won't work.

Software Packages

By default, the Azure notebook platform comes with a large set of pre-installed software packages provided by Anaconda, a Python distribution popular in numerical analysis and data science circles. Actually, three different versions of the Anaconda distributions are installed:

```
>>> !ls
anaconda2_410  anaconda3_410  anaconda3_431
```

Each distribution supports a different version of Python (at the time of writing: Python 2.7.11, 3.5.1 and 3.6.0).

To see for yourself the list of installed packaged, type:

```
>>> !conda list
# packages in environment at /home/nbcommon/anaconda2_410:
#
_nb_ext_conf              0.2.0                    py27_0
adal                      0.4.6                     <pip>
alabaster                 0.7.8                    py27_0
altair                    1.2.0                     <pip>
anaconda                  custom                   py27_0
anaconda-client           1.4.0                    py27_0
anaconda-navigator        1.2.1                    py27_0
...
```

The full list is rather large; refer to the appendix if you are interested. The list is also compared with the default set of packages in the Anaconda distribution. There are generally more packages in the Azure notebook platform; some of them are obviously Azure-specific. Additionally, a package that would be

missing from the Azure platform – for example wrapt – can easily be installed, either with

```
>>> !conda install -y wrapt
```

or – as long as it's available on PyPI – with

```
>>> !pip install wrapt
```

Note that these installations are performed as a user, not at the system level: you are merely `nbuser` and you don't have administrator rights in the Azure container. In particular, you won't be able to `apt-get install` your way out of missing software.

TODO: document Fortran, C/C++ & other stuff. Binaries from sources, packaged via conda (ex: curl, etc.)

Networking

[**TODO**]

Appendix – Conda Packages

Package	Anaconda (default)	Azure Notebooks
_license		
_nb_ext_conf		
adal		
alabaster		
altair		
anaconda		
anaconda-client		
anaconda-navigator		
anaconda-project		
applicationinsights		
argcomplete		
asn1crypto		
astroid		
astropy		
attrs		
Automat		
azure-batch		
azure-cli		
azure-cli-acr		
azure-cli-acs		

Package	Anaconda (default)	Azure Notebooks
azure-cli-appservice		
azure-cli-batch		
azure-cli-billing		
azure-cli-cdn		
azure-cli-cloud		
azure-cli-cognitiveservices		
azure-cli-command-modules-nspkg		
azure-cli-component		
azure-cli-configure		
azure-cli-consumption		
azure-cli-core		
azure-cli-cosmosdb		
azure-cli-dla		
azure-cli-dls		
azure-cli-feedback		
azure-cli-find		
azure-cli-interactive		
azure-cli-iot		
azure-cli-keyvault		
azure-cli-lab		
azure-cli-monitor		
azure-cli-network		
azure-cli-nspkg		
azure-cli-profile		
azure-cli-rdbms		
azure-cli-redis		
azure-cli-resource		
azure-cli-role		
azure-cli-sf		
azure-cli-sql		
azure-cli-storage		
azure-cli-vm		
azure-common		
azure-datalake-store		
azure-graphrbac		
azure-keyvault		
azure-mgmt-authorization		
azure-mgmt-batch		
azure-mgmt-billing		
azure-mgmt-cdn		
azure-mgmt-cognitiveservices		
azure-mgmt-compute		
azure-mgmt-consumption		
azure-mgmt-containerregistry		

Package	Anaconda (default)	Azure Notebooks
azure-mgmt-datalake-analytics		
azure-mgmt-datalake-nspkg		
azure-mgmt-datalake-store		
azure-mgmt-devtestlabs		
azure-mgmt-dns		
azure-mgmt-documentdb		
azure-mgmt-iothub		
azure-mgmt-keyvault		
azure-mgmt-monitor		
azure-mgmt-network		
azure-mgmt-nspkg		
azure-mgmt-rdbms		
azure-mgmt-redis		
azure-mgmt-resource		
azure-mgmt-sql		
azure-mgmt-storage		
azure-mgmt-trafficmanager		
azure-mgmt-web		
azure-monitor		
azure-multiapi-storage		
azure-nspkg		
azure-servicefabric		
azureml		
babel		
backports		
backports.shutil_get_terminal_size		
backports.ssl-match-hostname		
backports.weakref		
backports_abc		
bcrypt		
beautifulsoup4		
bitarray		
bkcharts		
blaze		
bleach		
bleach-whitelist		
bokeh		
boto		
boto3		
botocore		
bottleneck		
bqplot		
brewer2mpl		
bz2file		

Package	Anaconda (default)	Azure Notebooks
cachecontrol		
cairo		
cdecimal		
certifi		
cffi		
chardet		
chest		
click		
cloudpickle		
clyent		
cntk		
colorama		
conda		
conda-build		
conda-env		
configobj		
configparser		
constantly		
contextlib2		
cryptography		
curl		
cycler		
cython		
cytoolz		
dask		
datashape		
dbus		
decorator		
dill		
distributed		
docker-py		
docker-pycreds		
docutils		
dynd-python		
edward		
elasticsearch		
entrypoints		
enum34		
et_xmlfile		
expat		
fastcache		
fastlmm		
feedparser		
flask		

Package	Anaconda (default)	Azure Notebooks
flask-cors		
fontconfig		
freetype		
funcsigs		
functools32		
future		
futures		
gdal		
geos		
geotiff		
get_terminal_size		
gevent		
ggplot		
glib		
graphviz		
greenlet		
grin		
grpcio		
gst-plugins-base		
gstreamer		
h5py		
harfbuzz		
hdf4		
hdf5		
heapdict		
holoviews		
html5lib		
humanfriendly		
hyperlink		
icu		
idna		
imagesize		
incremental		
ipaddress		
ipykernel		
ipython		
ipython_genutils		
ipywidgets		
isodate		
isort		
itsdangerous		
jbig		
jdcal		
jedi		

Package	Anaconda (default)	Azure Notebooks
jinja2		
jmespath		
joblib		
jpeg		
jsonschema		
jupyter		
jupyter_client		
jupyter_console		
jupyter_core		
kafka-python		
kazoo		
kealib		
keras		
keyring		
klein		
lazy-object-proxy		
libdynd		
libffi		
libgcc		
libgdal		
libgfortran		
libgpuarray		
libiconv		
libnetcdf		
libpng		
libpq		
libprotobuf		
libsodium		
libtiff		
libtool		
libxcb		
libxml2		
libxslt		
line-profiler		
llvmlite		
locket		
lockfile		
luigi		
lxml		
mako		
Markdown		
markupsafe		
matplotlib		
memory-profiler		

Package	Anaconda (default)	Azure Notebooks
mistune		
mkl		
mkl-service		
mock		
monotonic		
mpmath		
msgpack-python		
msrest		
msrestazure		
multipledispatch		
natsort		
navigator-updater		
nb_anacondacloud		
nb_conda		
nb_conda_kernels		
nbconvert		
nbformat		
nbpresent		
networkx		
nltk		
nose		
notebook		
numba		
numexpr		
numpy		
numpydoc		
oauthlib		
odo		
olefile		
opencv		
openfst		
openpyxl		
openssl		
packaging		
pandas		
pandasql		
pandocfilters		
pango		
param		
paramiko		
partd		
patchelf		
path.py		
pathlib2		

Package	Anaconda (default)	Azure Notebooks
patsy		
pbr		
pcre		
pep8		
pexpect		
pickleshare		
pillow		
pip		
pixman		
plotly		
ply		
proj4		
prompt-toolkit		
prompt_toolkit		
protobuf		
psutil		
psycopg2		
ptyprocess		
py		
pyang		
pyasn1		
pyasn1-modules		
pycairo		
pycosat		
pycparser		
pycrypto		
pycurl		
pydocumentdb		
pydot		
pyflakes		
PyGithub		
pygments		
pygpu		
PyJWT		
pykafka		
pylint		
pymc		
pymc3		
pymongo		
Pympler		
pymssql		
pymysql		
PyNaCl		
pyodbc		

Package	Anaconda (default)	Azure Notebooks
pyopenssl		
pypachy		
pyparsing		
pyprof2calltree		
pyqt		
pysnptools		
pytables		
pytest		
python		
python-daemon		
python-dateutil		
pytz		
PyWavelets		
pywavelets		
pywget		
pyyaml		
pyzmq		
qt		
qtawesome		
qtconsole		
qtpy		
readline		
redis		
redis-py		
requests		
requests-oauthlib		
rope		
rpy2		
ruamel_yaml		
s3transfer		
scandir		
scikit-bio		
scikit-image		
scikit-learn		
scipy		
scp		
seaborn		
SecretStorage		
service-identity		
setuptools		
simplegeneric		
singledispatch		
sip		
six		

Package	Anaconda (default)	Azure Notebooks
snakeviz		
snowballstemmer		
sockjs-tornado		
sortedcollections		
sortedcontainers		
sphinx		
sphinx_rtd_theme		
spyder		
sqlalchemy		
sqlite		
sshtunnel		
ssl_match_hostname		
statsmodels		
subprocess32		
sympy		
tabulate		
tblib		
tensorflow		
terminado		
testpath		
theano		
Theano		
tk		
toolz		
tornado		
tqdm		
traitlets		
traittypes		
treq		
Twisted		
unicodecsv		
unixodbc		
urllib3		
vega		
vsts-cd-manager		
wcwidth		
websocket-client		
werkzeug		
wheel		
Whoosh		
widgetsnbextension		
word2vec		
wrapt		
xerces-c		

Package	Anaconda (default)	Azure Notebooks
xlrd		
xlsxwriter		
xlutils		
xlwt		
xmltodict		
xz		
yaml		
zeromq		
zict		
zlib		
zope.interface		