Operating Azure Data Lake with Python

Updated on: 1/31/2017

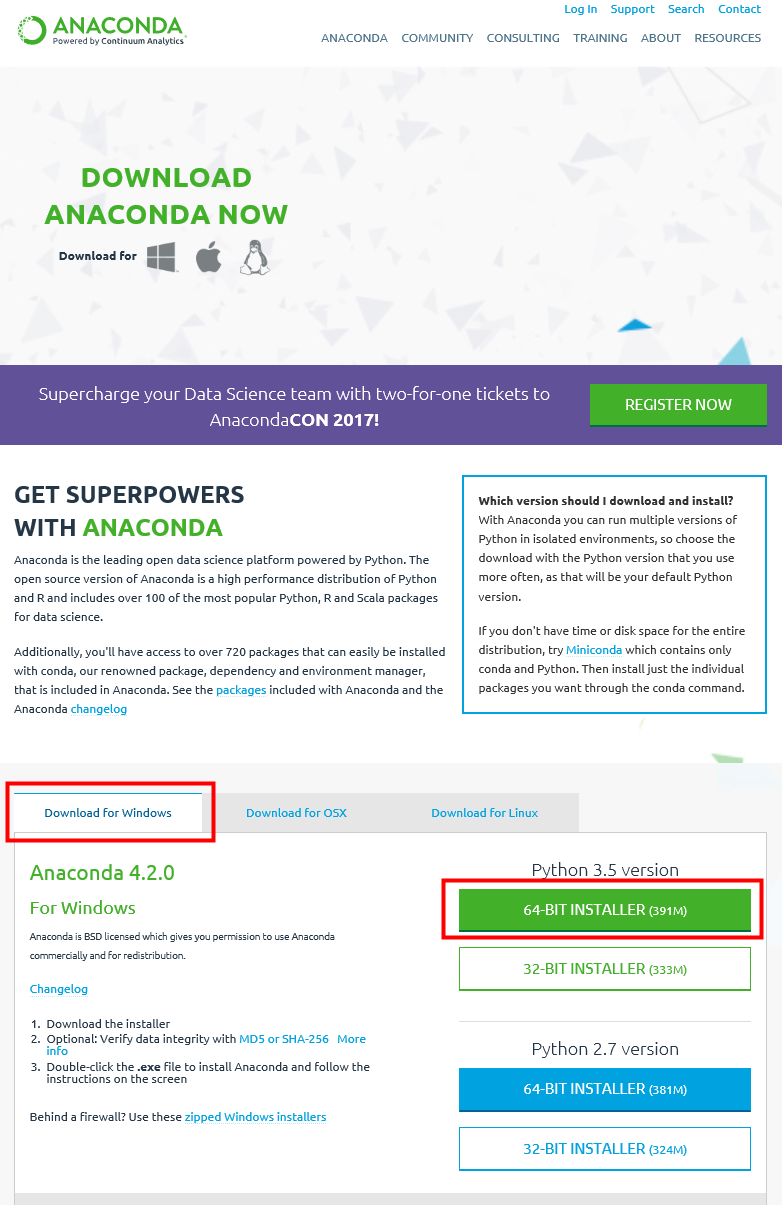
# Prerequisites

* You Need an ADLA Account
* You Need an ADLS Account

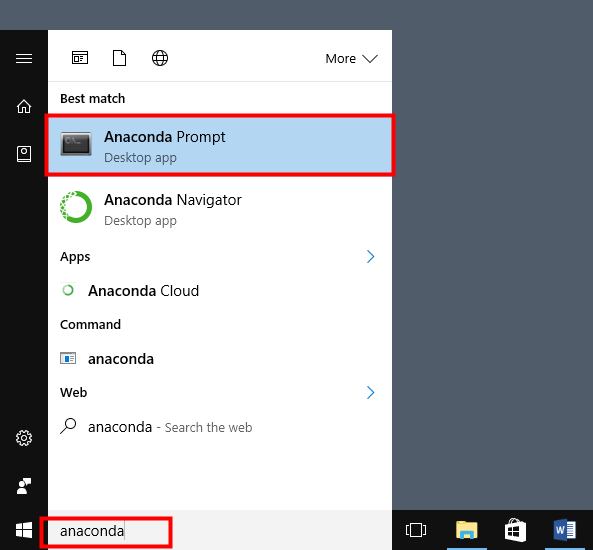
# Install the Anaconda Python distribution

Navigate here: <https://www.continuum.io/downloads>

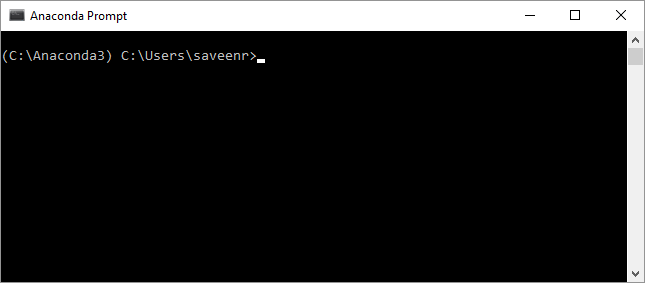
Under **Downloads for Windows** > **Python 3.5 version** click on **64-BIT INSTALLER**



Go to Start and search for “anaconda” and select **Anaconda Prompt**



This will launch the Anaconda prompt which is a normal CMD prompt



Close the Anaconda prompt.

# Introduction to pip

The pip command is how Python installs packages and plays the same role in the Python ecosystem as NuGet does for .NET developers.

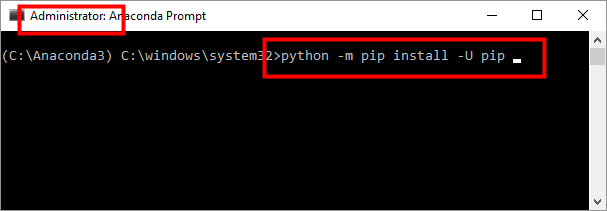
Pip is frequently updated so the first thing to do is upgrade it to the latest version.

# Upgrade pip to the latest Version

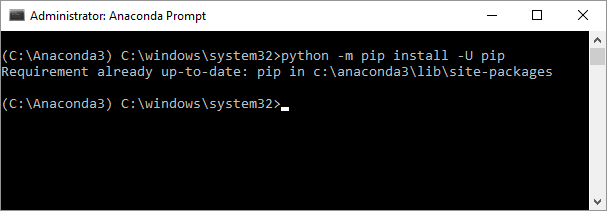
Launch the Anaconda Prompt **as Administrator**.

Run this command

python -m pip install -U pip



If everything is already up-to-date, you will see this:



# Install an Editor for Python Code

* You can use Visual Studio 2015
* Many people like using Visual Studio Code for editing python
* The Anaconda distribution also includes Spyder which is a Python IDE. If you installed Anaconda, Spyder is already installed on your machine.

# Setup a working folder for this Lab

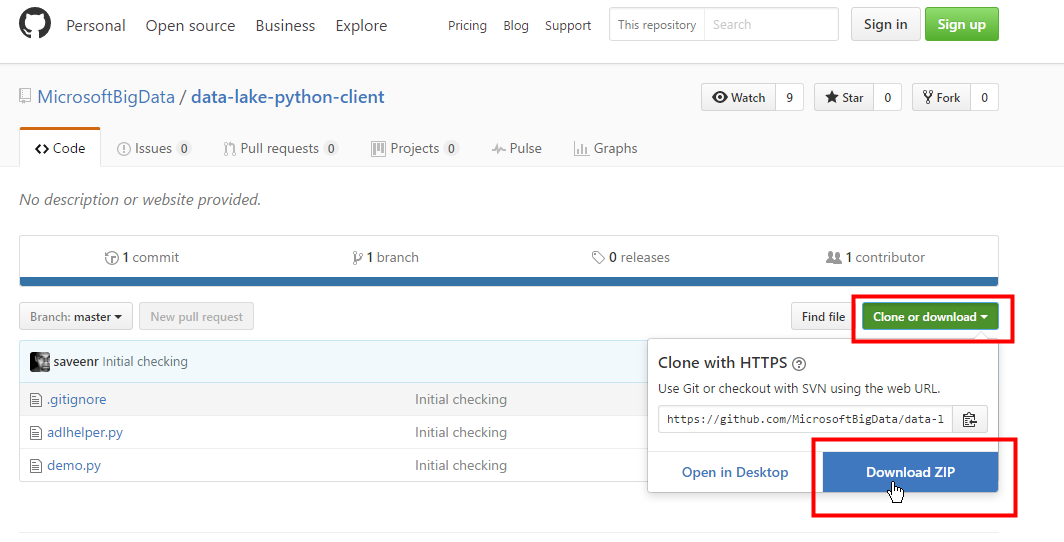
## Create a folder

* Create a folder to hold the source code we are going to be working on.

MKDIR c:\tr24ptyhon

## Download the helper code

* Launch a browser and go here: <https://github.com/Azure-Samples/data-lake-python-client>
* Click the **Clone or download** button
* Click **Download ZIP**



* Extract the ZIP
* The place the extracted files into c:\tr24python
  + So you should have:
    - c:\tr24python\adlhelper.py
    - c:\tr24python\demo.py
    - c:\tr24python\demo\_final.py
    - c:\tr24python\pip\requirements.txt
  + You can ignore the .gitignore file

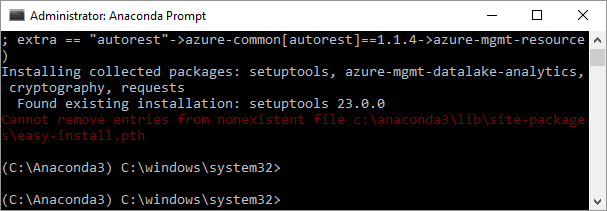
# Install/Upgrade Azure modules via pip

There are several modules we need to Azure Data Lake – and we always want the latest version.

This lab was developed with a specific set of packages. The ZIP included a requirements.txt file in the pip folder. It contains all a list of the modules and specific versions you’ll need. Run the command below to install them.

pip install -r  c:\tr24python\pip\requirements.txt

You might see a message that says “Cannot remove entries from nonexistent file c:\anaconda3\lib\site-packages\easy-install.pth”. Don’t worry. You can safely ignore this.



# Verify that the demo.py file runs

* Edit the demo.py file
* It will look like this

# Common Azure imports

import adal

import azure.mgmt.resource.resources

import msrestazure.azure\_active\_directory

# ADLS imports

import azure.mgmt.datalake.store

import azure.datalake.store

# ADLA imports

import azure.mgmt.datalake.analytics

# All other imports

import os

import sys

import itertools

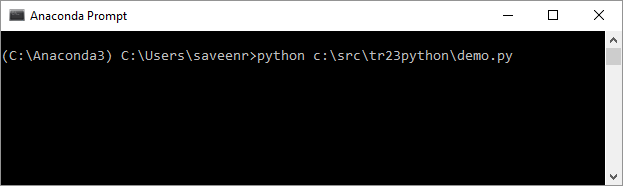
import adlhelper

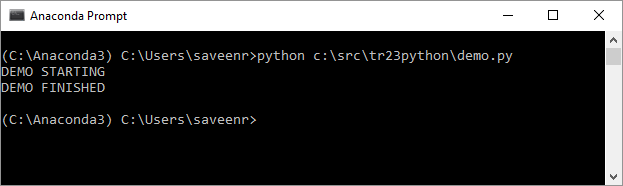
print("DEMO STARTING")

print("DEMO FINISHED")

* Launch the Anaconda Prompt
* Run the demo program using this command

python c:\src\tr23python\demo.py





* If you don’t see any errors it means that all the libraries you need to complete this lab are usable

## Customize for your environment

* Between the DEMO STARTING and DEMO ENDING print statements, add the following code

print("DEMO STARTING")

# define constants

tenant = "YOURTENANT"

subscription\_id = 'YOURSUBSCRIPTIONID'

adls\_account\_name = 'YOURADLSACCOUNTNAME'

adla\_account\_name = 'YOURADLAACCOUNTNAME '

# Handle Authentication

auth\_session = adlhelper.AuthenticatedSession( subscription\_id , tenant, r"C:\src\tr23python\adl\_demo\_tokencache.pickle")

token = auth\_session.Token

credentials = auth\_session.Credentials

# Client construction

resource\_clients = adlhelper.DataLakeResourceClients( auth\_session )

store\_clients = adlhelper.DataLakeStoreClients( auth\_session, adls\_account\_name)

analytics\_clients = adlhelper.DataLakeAnalyticsClients( auth\_session )

print("DEMO FINISHED")

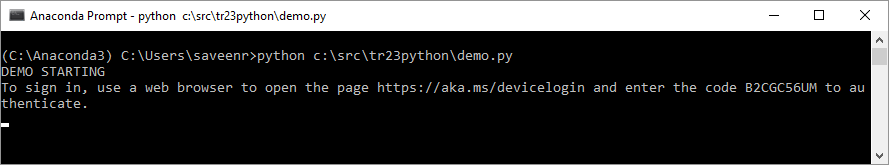
* In the section of the script called “define constants”, replace the values with what value you need for you subscription, tenant, and ADLA/AADLS accounts

## Verify that Authentication Works and that you can get Python clients

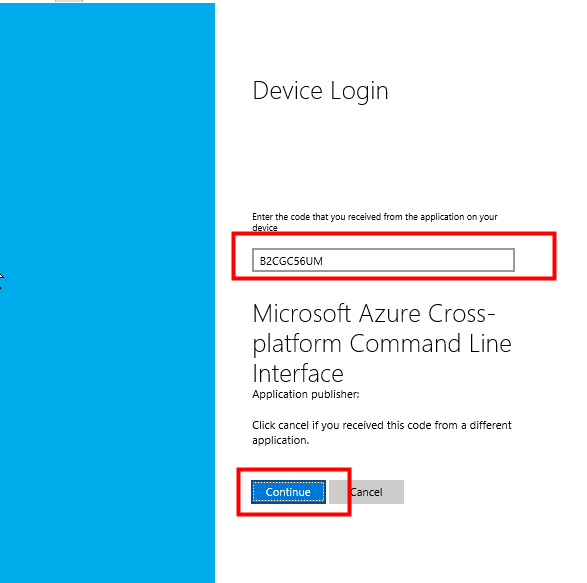
* Run the demo.py file by typing in the Anaconda Prompt

python c:\src\tr4\python.demo.py

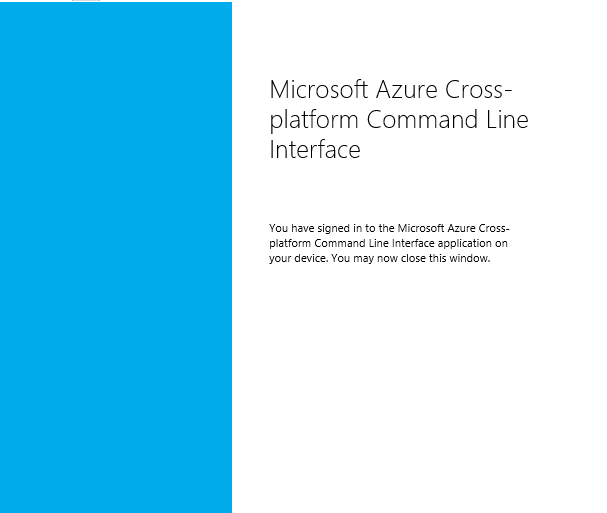
* The first time you run this demo with the code above you will see a message like this:



* Copy the code shown in your Command Prompt
* Go to <http://aka.ms/devicelogin>
* Paste the code in (make sure there is no leading or trailing whitespace)
* Press **Continue**



* You may be asked to sign-in to Azure. Do that.
* Once it is complete it will say this:



* This means the authentication has succeeded and you can close the browser window.
* Go back to the Anaconda Prompt
* You’ll see that the program should have completed.
* If you have gotten to this point without errors, you are ready to continue with the steps.

# Notes

In the demo program, notice that these objects have been created.

resource\_clients

store\_clients

analytics\_clients

Each one of these objects contains one or more REST clients objects that you can use to operate some facet of Azure Data Lake. For the rest of the lab you will call methods on those REST clients.

# Common Data Lake Analytics Tasks

## List ADLA Accounts

adla\_accounts = adlaAcctClient.account.list()

for a in adla\_accounts:

print("ADLA: " + a.name)

## List 10 Jobs in the Account

jobs = adlaJobClient.job.list( adla\_account\_name )

jobs = itertools.islice(jobs,10) # comment this out if you want all the jobs

for j in jobs:

print("---------------------------------")

print(j.name)

print(j.submit\_time)

print(j.submitter)

## Submit a Job

# Common Data Lake Store Tasks

## List ADLS Accounts

adls\_accounts = adlsAcctClient.account.list()

for a in adls\_accounts:

print("ADLS: " + a.name)

## List Files at the root of the account

files = adlsFileSystemClient.ls()

for f in files:

print(f)

## List Files in a folder

files = adlsFileSystemClient.ls( "/somefolder" )

for f in files:

print(f)