**Python/Jupyter/Git/ML Basics**

**Introductory Lab Guide:**

**Getting Started with  
open source tools**

AI Center of Competency

[aicoc@us.ibm.com](mailto:aicoc@us.ibm.com)   
IBM Systems Client Experience Center  
May 8th , 2020

**Table of Contents**

[1. Introduction 3](#_Toc39570518)

[2. Prerequisites 3](#_Toc39570519)

[3. Important Notes 3](#_Toc39570520)

[4. Quick Start Guide to Setup Environment 3](#_Toc39570521)

[LOGIN 3](#_Toc39570522)

[Install Jupyter and JupyterLab 5](#_Toc39570523)

[Appendix A: Requesting an WML-CE Server from the CECC 7](#_Toc39570524)

**IBMers & BPs:** See Appendix A for instructions on requesting an IBM H2O.ai Driverless AI server.

# Introduction

This lab is designed to demonstrate some of the basics concepts you need to know when using WML-CE, Python, Jupyter Notebooks and Git. These are essential tools used by many in the data science community and a working knowledge of what these tools do and how to use them is critical to understand the workflow of a typical data scientist or developer. This lab will use resources from our Worldwide Client Experience Center Cloud or CECC for short. A detailed instruction is in the appendix for you to deploy your own instances. Let’s get started.

# Prerequisites

1. Access to a WML-CE instance (version 1.6.2, or later)
   * **IBMers and Business Partners can request access to an instance of WML-CE.** See Appendix A for instructions on how to request access.
2. PC or laptop with connectivity to the Internet to download datasets or interest.

This is not needed if datasets have been pre-loaded.

1. Connectivity between your PC or laptop to the WML-CE AI instance.
   * A VPN may be necessary on your device if the server resides behind a firewall. This is not needed if server is publicly accessible.

# Important Notes

* This lab was most recently tested with WML-CE 1.7 CECC instance. Using these instructions in other environments may require additional setup.

* **Do not use any hostnames, IP addresses, usernames, or passwords mentioned in this guide. They are for example purposes only.** They will not work for you. Use what has been provided to you by instructors or what you have provisioned yourself.
* Anything in **BOLD typeface** below is intended to mean that you should type that at the system prompt

# Quick Start Guide to Setup Environment

To start the lab, follow the instructions below. You can keep track of your progress by checking off each step when completed.

## LOGIN

Open up a terminal on your computer and ssh to your server (credentials to be provided by instructor)

ssh cecuser@129.40.94.249

## **Anaconda Basics**

All the WMLCE tools are installed in the Anaconda 3 python distribution. Anaconda is one of the most popular distributions and comes with a package manager tool called conda. The conda cheat sheet is located here

<https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf>

There are a number of commands that you can run to explore and manage you conda environment. The one most frequently used is **conda install your-favorite-package.** Another handy command you can use is **conda list**. This command shows you all the packages currently installed in your distribution.

Conda also has a concept of ‘virtual environments’. Virtual environments are awesome, and you should get used to using them if you end up doing your own projects. There are many reasons to use virtual environments, but the one we find most useful is that you can isolate projects from each other. This is good because many times different projects require different levels of python packages, and virtual environments allow you to have multiple distinct configurations.

Clone Conda Environment

conda create --name wmlce\_37 --clone wmlce\_env3

Activate Conda Virtual Environment

conda activate wmlce\_37

Note : There is a README file in the home directory with more details regarding recommended setup.

## **Github Basics**

Git Clone our sample project.

Git is a full fledged version control system, and there are a lot of commands to master to get the full use. For our purposes, we will use the git clone command, but a deeper understanding of git is recommended if you want to use it for version control. See this link for more information.

https://www.atlassian.com/git/tutorials/what-is-version-control

Here we will be using a sample repository for the python/ML training part of the lab. To access this repository

A screenshot of a cell phone

Description automatically generated

For any public Git project, you can always download the source code to your computer or server. To do this, browse to the git url (in this example <https://github.com/dustinvanstee/aicoc-python-basics>) and click the green ‘clone of download’ button and copy the url. Then in your terminal run the following command in your home directory.

git clone https://github.com/dustinvanstee/aicoc-python-basics

This will download the source contents to your directory.

Note: here we used the ‘git’ command line interface. This is a very complete tool that allows you to manage your code repositories. Git has a number of conceptual details we will review in the lab demonstration, but it is worthwhile to watch a Youtube video or 2 on the fundamentals of git to get the hang of it. The main commands are **git push, git pull, git clone, git add, git commit**. With just this small subset of commands you can accomplish quite a few tasks.

Advanced and Optional: Fork a copy of this project to your own github repo

Many times over the course of doing projects you will come across another persons’ implementation on Github that you might want to copy and then modify for your purposes. In Github parlance this is called forking. By forking a project, the code is copied to your repository, but it also points back to the originator. This allows you to make changes, and then if you feel like you want to contribute, you can ask the originator to accept your changes. This is called a pull request.

To fork a project, simply login to github and go to the project you are interested in and click the fork button …



Instead of cloning the project from the originator (in this case dustinvanstee) you can now clone from your own repo!

**Hint:** you’ll need to create a login ID for the github website if this is your first visit. The website is a popular resource for data scientists. It hosts a number of interesting code repositories.

## **Install Jupyter and JupyterLab**

Jupyter notebooks are used all the time in the data science and developer communities, here we will install the software and use it to load the notebooks we downloaded in our git repository.

A screenshot of a social media post

Description automatically generated

Install Jupyter tools

In your conda **wmlce\_37** environment run the following commands.

conda install -y jupyter

pip install jupyterlab

pip install seaborn

This will install the required packages needed to run Jupyter/Jupyterlab.

Startup Jupyter notebook

Now that we have installed Jupyter, let’s start it up and load our notebook! There are many ways to configure the startup, but the simplest is to start it on a port of your choosing with the following arguments. We are showing here how-to startup in non- password protected mode for ease of login, but if you prefer remove the last 2 arguments and use the token provide to you in the terminal.

jupyter notebook --ip=0.0.0.0 --port=5050 --no-browser --NotebookApp.token='' --NotebookApp.password=''

Browse to URL for the notebook

After invoking the Jupyter notebook, you can now use your browser to access the github repo that we installed. Simply open your browser and enter the URL. Here are some samples:

[http://your\_ip\_addr:5050/lab](http://129.40.94.249:3761/lab)

[http://your\_ip\_addr:5050/tree](http://129.40.94.249:3761/tree)

Notice there are two URL’s above. The /**lab** endpoint is the new more “modern” interface for the Jupyter notebook environment, while the /**tree** endpoint is the more traditional style format.

Here is a link to a Jupyter notebook cheatsheet. During our lab the instructors will demo some of the capabilities.

<https://www.edureka.co/blog/wp-content/uploads/2018/10/Jupyter_Notebook_CheatSheet_Edureka.pdf>

Once you have launched your jupyter notebook, start with the notebook starting with 00.\*ipynb and launch it.

00.\*ipynb - Python basics

01.\*ipynb - Telco Example and Data Exploration

02.\*ipynb - Deep Dive into Telco example basic ML

.

**Congratulations! You are now ready to proceed to the Jupyter Notebook TBD for further instruction and training on Python/Pandas and ML Basics.**

# Appendix A: Requesting an WML-CE Server from the CECC

People outside of IBM wanting to try WMLCE can request access to a trial environment by making a request to aicoc@us.ibm.com. At one point you will be given the opportunity to sign up for access.

IBMers and Business Partners can request access to an WMLCE environment through the Client Experience Center Cloud (CECC). The steps for requesting a server and accessing the provisioned server are included below. (note there is access to other SW packages including H20 DriverlessAI and IBM Visual Insights).

**Provision WMLCE Instance through the CECC Portal**

1. Here are the steps to Follow – click on this link: <http://ibm.biz/cecc-portal>.
2. Login with your IBM id and review / accept the usage agreement.
3. From the left sidebar, select the **WMCLE** checkbox.

A screenshot of a cell phone

Description automatically generated

Select on of the **WMLCE**  tiles by clicking the **Add to cart** link within it. You can select POWER8 if availability is an issue or you don’t need high performance levels (light demo for e.g.). Note that there is also are also tiles with CECC to request a bare metal environment, but this is typically reserved for Proofs of Concept and client opportunities, and not for education purposes.

A screenshot of a cell phone

Description automatically generated

Leave all the default options and click the **Add** button.

A screenshot of a cell phone

Description automatically generated

Click the **Checkout** button.

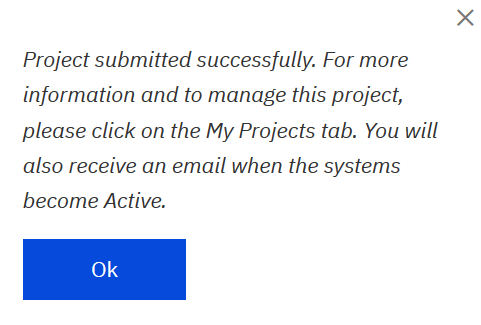
Fill out the project details and click the **Create Project** button. A few things to note:

* + Be sure to set a descriptive name and specify **“Education / Training”** in the **Used for** section.
  + If you are requesting immediate access (versus a time in the future) then provisioning may take upwards of two or more hours.
  + Please consider specifying a realistic timeframe that is courteous to other users of the CECC. For instance, if you’re only planning to go through a lab then you may only need it for a couple of days.

A screenshot of a cell phone

Description automatically generated

Once submitted successfully, the following message should be displayed. Click the **Ok** button to dismiss.



**Connecting to the Provisioned Visual Insights Server**

Navigate to the **My Projects** tab in the top navigation bar.

A screenshot of a cell phone

Description automatically generated

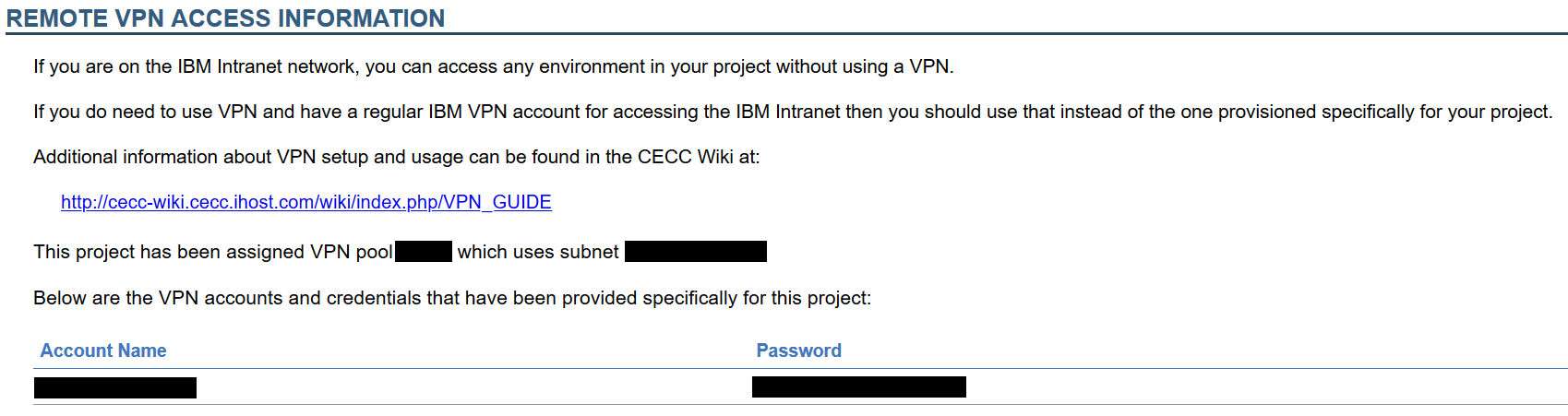
Click the **Project Kit URL** link associated with your request.

The Project Kit will look like this:

A screenshot of a cell phone

Description automatically generated

If you are not on the IBM Intranet then you need to VPN in to be able to access your Visual Insights instance. Scroll down to the bottom of the Project Kit and follow the instructions on connecting to the VPN.



Go to the **Reservation Information** section and click the link provided and review the instructions.

A screenshot of a social media post

Description automatically generated

**Deleting Your WMLCE Instance**

The environment will be active until the end date specified for the project, at which point it will be automatically deleted. However, if you are finished with it prior to this date then please be courteous and manually delete it so that the resources can be used by others. You can do this by choosing **End** from the **Action** list for your project.

A screenshot of a cell phone

Description automatically generated

*(Credit: Thomas Famularo)*

# Appendix B: Use ssh tunnel to connect Jupyter localhost on remote system

**Mac window 1 :**

(base) [cecuser@p1320-kvm1 aicoc-python-basics]$ conda activate wmlce\_lab

(wmlce\_lab) [cecuser@p1320-kvm1 aicoc-python-basics]$ jupyter notebook

[I 13:48:20.063 NotebookApp] JupyterLab extension loaded from /home/cecuser/anaconda3/envs/wmlce\_lab/lib/python3.6/site-packages/jupyterlab

[I 13:48:20.063 NotebookApp] JupyterLab application directory is /home/cecuser/anaconda3/envs/wmlce\_lab/share/jupyter/lab

[I 13:48:20.066 NotebookApp] Serving notebooks from local directory: /home/cecuser/aicoc-python-basics

[I 13:48:20.067 NotebookApp] The Jupyter Notebook is running at:

[I 13:48:20.067 NotebookApp] <http://localhost:8888/?token=15a299914d99553ae37185df925200bc540943549a223087>

[I 13:48:20.067 NotebookApp] or <http://127.0.0.1:8888/?token=15a299914d99553ae37185df925200bc540943549a223087>

[I 13:48:20.067 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

[W 13:48:20.073 NotebookApp] No web browser found: could not locate runnable browser.

[C 13:48:20.073 NotebookApp]

To access the notebook, open this file in a browser:

<file:///home/cecuser/.local/share/jupyter/runtime/nbserver-14050-open.html>

Or copy and paste one of these URLs:

<http://localhost:8888/?token=15a299914d99553ae37185df925200bc540943549a223087>

or <http://127.0.0.1:8888/?token=15a299914d99553ae37185df925200bc540943549a223087>

**Mac window 2: (do not close this window once log in to WML-CE server)**

**Note: use any port that is not used on Mac, in my case, 8889 works fine**

Jamess-MacBook-Pro-2:~ [jcwang@us.ibm.com$](mailto:jcwang@us.ibm.com$) ssh -L 8889:localhost:8888 [cecuser@p1320-kvm1.cecc.ihost.com](mailto:cecuser@p1320-kvm1.cecc.ihost.com)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Welcome to the Client Experience Center Cloud

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IBM's internal systems must only be used for conducting IBM's business

or for purposes authorized by IBM management. Use is subject to audit

at any time by IBM management.

Unauthorized access will be investigated and penalties will be pursued

in conformance with applicable laws and regulations. If you are not an

authorized user disconnect now.

[cecuser@p1320-kvm1.cecc.ihost.com's](mailto:cecuser@p1320-kvm1.cecc.ihost.com's) password: <enter cecuser password>

Last failed login: Thu May 7 13:54:37 EDT 2020 from 9.85.171.169 on ssh:notty

There were 3 failed login attempts since the last successful login.

Last login: Thu May 7 13:46:52 2020 from 9.85.171.169

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Welcome to the Client Experience Center Cloud

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IBM's internal systems must only be used for conducting IBM's business

or for purposes authorized by IBM management. Use is subject to audit

at any time by IBM management.

Please refer to the Client Experience Center Cloud Wiki which you will

find at <http://cecc-wiki.cecc.ihost.com> for information about the IaaS,

PaaS and other provisioned environments including FAQs and information

on performing common tasks.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PaaS Red Hat Enterprise Linux 7.6 with Watson Machine Learning Community Edition

KVM Guest, POWER8, 32 vCPU, 64GiB Memory

1 Tesla P100 SXM2 16GB GPU Available

(base) [cecuser@p1320-kvm1 ~]$

**From Mac Browser (only Firefox works for me):**

<http://localhost:8889/tree>

*(Credit: James Wang)*