Introduction

In this course, you will learn how to write and debug Python Spark (pySpark) programs in four weekly lab exercises. We expect everyone to use the same software development environment. Our software development environment uses a Virtual Machine (VM) to simplify the installation process and provide all students with the same development environment. You will need to first install two free software packages (VirtualBox and Vagrant), and then download and install our VM image. In this segment, we will walk you through the process of downloading and installing the required software and VM image, and of running a test pySpark IPython notebook.

*Note: The total size of all the files you will need to download is less than 1 GB.*

Hardware and Software Prerequisites

You will need a machine that meets certain minimum hardware requirements and is running a supported operating system. *Note: Other configurations may work fine, but we will not be able to provide official support for them.*

MINIMUM HARDWARE REQUIREMENTS

You should use a machine with the following minimum hardware requirements:

* + Free disk space: 3.5 GB
  + RAM memory: 2 GB (4+ GB preferred)
  + Processor:  Any recent Intel or AMD multicore processor should be sufficient.

SUPPORTED OPERATING SYSTEMS

We support the following Operating Systems:

* + - 64-bit (preferred) Windows 7 or later
    - 64-bit (preferred) Mac OS X 10.9.5 or later
    - 64-bit (preferred) Linux (CentOS 6 or later, or Ubuntu 14.04 or later)
    - 32-bit Windows 7 or later
    - 32-bit Linux (CentOS 6 or later, or Ubuntu 14.04 or later)

*NOTE for Linux users: Based on our testing, please make sure you are using Vagrant 1.7 or newer with Linux. The default Vagrant version packaged with most Linux distributions does not seem to work properly with our course software.*

(Optional) For the best performance with VirtualBox, we recommend using a machine that supports hardware virtualization features: Intel VT-X or AMD AMD-V. *Note that on many systems, the hardware virtualization features first need to be enabled in the BIOS before VirtualBox can use them.*

# Installing the Required Software Packages

## REQUIRED SOFTWARE PACKAGES

There are two required free software packages:

* Oracle's [Virtual Box](https://www.virtualbox.org/)
* [Vagrant](https://www.vagrantup.com/) automatic VM configuration

Note: If you already have either software package installed, makes sure that the versions are VirtualBox 4.3.28 (or later) and Vagrant 1.7.2 (or later).

The following video segments walk you through the steps to download and install the required software packages on your machine, and to run your first Apache Spark notebook.

# Downloading and Installing VirtualBox

Go to the [Downloads](https://www.virtualbox.org/wiki/Downloads) page for VirtualBox and download the appropriate version for your operating system.

While you wait for your download to complete, watch the appropriate video (below) for your system to see the VirtualBox download and installation process.

## FAQS FOR WINDOWS

When attempting to start VirtualBox, you may encounter the following error:

VirtualBox - Critical Error  
Failed to create the VirtualBox COM object.  
The application will now terminate.  
Callee RC: E\_NOINTERFACE (0x80004002)

If you encounter this error, you may not have installed VirtualBox as Administrator. You should perform the following steps:

* + Uninstall VirtualBox
  + Reboot your machine
  + Reinstall VirtualBox by right-clicking on the installation file and selecting "Run as Administrator"
  + Once the installation completes, select "Start Menu->Oracle VM VirtualBox"
  + Right click on "Oracle VM VirtualBox" and select "Properties"
  + Navigate to the "Compatibility" tab
  + In the Compatibility mode box, click the checkbox for  "Run this program in compatibility mode for:" and select "Windows XP (Service Pack 3)"
  + In the Priviledge Level box, click on the check box for "Run this program as an administrator"
  + Click on "Change settings for all users" and make sure that both the checkboxes for "Run this program in compatibility mode for:" and "Run this program as an administrator" are set.

# Downloading and Installing Vagrant

Go to the [Downloads](https://www.vagrantup.com/downloads.html) page for Vagrant and download the appropriate version for your operating system.

While you wait for your download to complete, watch the appropriate video (below) for your system to see the Vagrant download and installation process.

*NOTE for Linux users: Based on our testing, please make sure you are using Vagrant 1.7 or newer with Linux. The default Vagrant version packaged with most Linux distributions does not seem to work properly with our course software.*

If you encounter the Windows error: “Installation Directory must be on a local hard drive”, this is a permissions error and you need to force the installer to run with administrator privileges.

To do so:

* 1. Locate the Vagrant MSI installer file that you downloaded.
  2. While holding down the Shift key on the keyboard, right-click on the Vagrant MSI, then choose **Copy As Path**.
  3. Go to Start > All Programs > Accessories.
  4. Right-click on **Command Prompt** and choose **Run As Administrator**. This should open a command prompt window, labeled "Administrator:".
  5. In the Command Prompt window, type **msiexec /i**(you need to enter a single space after "/i").
  6. Right-click in the Command Prompt window, then choose **Paste**. This should paste the path to the MSI file that you copied in Step 2 above.
  7. Press Enter to run the command.

# Downloading and installing the virtual machine

1. Create a custom directory (e.g., for windows users c:\users\marco\myvagrant or for Mac/Linux users/home/marco/myvagrant)
2. Download this [file](https://github.com/spark-mooc/mooc-setup/archive/master.zip) (https://github.com/spark-mooc/mooc-setup/archive/master.zip) to the custom directory and unzip it.
3. From the unzipped file, copy Vagrantfile to the custom directory you created in step #1 (NOTE: It must be named exactly "Vagrantfile" with no extension)
4. Open a DOS prompt (Windows) or Terminal (Mac/Linux), change to the custom directory, and issue the command "vagrant up"

NOTE: On Microsoft Windows you may get a Windows Firewall exception popup regarding "vboxheadless".  Adding an exception is optional and does not seem to impact the Virtual Machine once it is installed.

When running "vagrant up" again, you may receive the following error:

bsdtar: Error opening archive: Unrecognized archive format

The solution is to first use the command:

vagrant box remove sparkmooc/base

Then, you can use the "vagrant up" command again.

## FAQS FOR WINDOWS

When you type "vagrant up", if you encounter the following error:

Bringing machine 'sparkvm' up with 'virtualbox' provider...

There was an error while executing 'VBoxManage', a CLI used by Vagrant for controlling VirtualBox. The command and stderr is shown below.

Command: ["list", "hostonlyifs"]

Stderr: VBoxManage.exe: error: Failed to create the VirtualBox object!

VBoxManage.exe: error: Code E\_NOINTERFACE (0x80004002) - No such interface supported (extended info not available)

VBoxManage.exe: error: Most likely, the VirtualBox COM server is not running or failed to start.

First, make sure you installed VirtualBox using "Run as Administrator". If you did not, then you should follow the directions in the Installing Virtual Box section to uninstall and reinstall VirtualBox. Second, try starting command using "Run as Administrator" (right-click on command and select "Run as Administrtor").

On Linux, if you encounter the following error message:

SSH:

\* The following settings shouldn't exist: insert\_key

vm:

\* The box 'sparkmooc/base' could not be found.

It means that your Vagrant version is out of date. Remove the vagrant install (apt-get remove). Then you'll want to download the latest version from the vagrant site: <https://www.vagrantup.com/downloads.html>.

Install using the command "sudo dpkg -i vagrant.deb" and re-run "vagrant up" and it should install correctly.

# Basic Instructions for Using the Virtual Machine

1. To start the VM, from a DOS prompt (Windows) or Terminal (Mac/Linux), issue the command "vagrant up".
2. To stop the VM, from a DOS prompt (Windows) or Terminal (Mac/Linux), issue the command "vagrant halt".Note: You should always stop the VM before you log off, turn off, or reboot your computer.
3. At the end of the course, to erase or delete the VM, from a DOS prompt (Windows) or terminal (Mac/Linux), issue the command "vagrant destroy". **Warning: If you erase or delete the VM, you will lose any work you have done and data you have saved, and you will have download it again when you use the "vagrant up" command.**
4. Once the VM is running, to access the notebook, open a web browser to "<http://localhost:8001/>" (on Windows and Mac) or "<http://127.0.0.1:8001/>" (on Linux).

NOTE: In step #4 above, try "<http://localhost:8001/>" first, if that doesn't work, try "<http://127.0.0.1:8001/>", the proper choice depends on the configuration of your computer.  We pre-configuted the virtual machine to by default start the iPython notebook on port 8001.  If you are having problems connecting to the notebook (neither of the two links works), you should check the output of the "vagrant up" command. It may be the case that there was a conflict on your computer with a program already using port 8001 and a higher port number was automatically used. If this occurs, you should use that port number instead of port 8001, and all the rest of the instructions will be the same.

The following videos walk you through these basic steps.

# Running Your First Notebook

Running your first notebook will test your software setup and environment.

1. If it is not already running, start the Virtual Machine by issuing issue the command "vagrant up" from a DOS prompt (Windows) or Terminal (Mac/Linux).
2. You should have already downloaded and unzipped the master.zip file in the module "[Downloading and installing the virtual machine](https://courses.edx.org/courses/BerkeleyX/CS100.1x/1T2015/courseware/d1f293d0cb53466dbb5c0cd81f55b45b/920d3370060540c8b21d56f05c64bdda/)".  The zip file contains the file "lab0\_student.ipynb" that you will need in step #4.  You can view a read-only online version of the Spark iPython notebook [here](http://nbviewer.ipython.org/github/spark-mooc/mooc-setup/blob/master/lab0_student.ipynb).
3. Once the Virtual Machine is running, access the Jupyter web UI for running IPython notebooks by navigating your web browser to "[http://localhost:8001](http://localhost:8001/)" (or "<http://127.0.0.1:8001/>").
4. On the Jupyter web page, use the Upload button to upload the "lab0\_student.ipynb" Spark iPython notebook file that was mentioned in step #2.
5. Select the file and run each cell - verify that you do not encounter any errors.
6. At the end of the notebook, instructions are included on how to export the notebook as a Python (.py) file and to submit that file to the autograder.

The following videos walk you through the steps of running your first notebook.

## THE COURSE AUTOGRADER

Assignments in this course are automatically graded using an autograder, which runs in the cloud on Amazon EC2. The autograder uses automatic scaling to handle large increases in the number of submissions.

## SUBMISSION FORMAT

Because we are using an autograder, it is very important that your submissions comply with the following rules:

* Only use the following libraries: standard python libraries, numpy, pyspark, and test\_helper (the autograder library).
* Don't leave in extraneous code, as the autograder will timeout if a submission takes too long.
* Only change sections of code where you see <FILL IN>, as changing other parts of the code, including directory paths, may cause the code to fail the autograder's tests.

## AUTOGRADER RESPONSE TIME

When you submit an assignment to be graded, you can expect to see a graded response within a few minutes. Note that the scale-up process itself can take a few minutes when many users simultaneously submit their assignments, so you may see additional delays in grading. If you do not receive feedback from the autograder server within 1 hour, please resubmit your code.  If you not receive feedback on your resubmitted code within 30 minutes, please use the [Piazza discussion group](https://piazza.com/edx_berkeley/summer2015/cs1001x) to contact the TAs for support.

Note: The autograder server cluster checks the submission volume (numbers of submissions) every five minutes.  If necessary, the autograder launches additional instances to handle the volume.

## AUTOGRADER FEEDBACK

There are 3 types of feedback.

* All tests passed or some tests failed.  The result for each test will be in the feedback.
* Timeout.  The submitted code took too long to execute.  In this case, please optimize your code's running time.
* Abnormal results are detected.  In this case, please review your code carefully according to the guidelines found on this page. If you need help with your code, please use the [Piazza discussion group](https://piazza.com/edx_berkeley/summer2015/cs1001x) to contact the TAs for support.

## SUBMITTING THE TEST NOTEBOOK TO THE AUTOGRADER

(100 points possible)

Once you have successfully run the test notebook, you can submit the notebook to the course autograder. First, export the test notebook as an Python (.py) file. Next, use this file chooser to select your file and click "Check" to submit your code to the course autograder. The video in the "Running your first notebook" unit shows you this process.

The course autograder automatically scales as the number of pending submissions grows, but the time to receive your feedback will depend on how many other users are submitting at the same time.

Reminder: The previous module "Running Your First Notebook" guides your through how to export a notebook to a Python file and how to submit the file to the autograder.

Select your Python (.py) file: