```
#######################################
# Commands Used in Hands-on Exercises #
#######################################

=============================================================
<DIY1> Connect to HiPerGator, launch standalone Spark cluster
=============================================================

# connect to hipergator
ssh -Y <your_ID>@hpg.rc.ufl.edu

# copy hands-on exercise files to your home directory
cp /ufrc/spark_workshop/share/* ~/

# see the list of files
ls

# launch the standard-alone spark cluster
sbatch spark-local-cluster.sh

# check your job status
squeue –u <your ID>


=====================================================
<DIY2> Monitor the Spark cluster in your local browser
=====================================================

# get the IP and port of MasterWebUI
grep MasterWebUI spark_cluster.err

# open a new terminal on your laptop, in the new terminal, type the
following
# you need to replace "172.16.192.164:8080" with your MasterWebUI IP
# your_ID = your-user-name

ssh -L 10001:172.16.192.164:8080 your_ID@hpg.rc.ufl.edu

# on your laptop, open a web browser and enter the following web
address

 localhost:10001


================================
<DIY3> Spark interactive shells
================================

# Spark Interactive shells in Scala
```

```
SPARK_MASTER=$(grep "Starting Spark master" *.err | cut -d " " -f 9)

module load spark

spark-shell --master $SPARK_MASTER

# Spark Interactive shells in Python

SPARK_MASTER=$(grep "Starting Spark master" *.err | cut -d " " -f 9)

module load spark

pyspark --master $SPARK_MASTER

# Here is a sample python program for calculating pi value
# you can enter to the interactive shell

from operator import add
from random import random

partitions = 10
n = 100000 * partitions

def f(_):
   x = random() * 2 - 1
   y  = random() * 2 - 1
   return 1 if x ** 2 + y ** 2 <= 1 else 0

count = sc.parallelize(range(1, n + 1), partitions).map(f).reduce(add)
print("Pi is roughly %f" % (4.0 * count / n))


=====================================================
<DIY4> Running python script via pyspark command line
=====================================================

# see the python file for calculating pi
cat diy4.py

# load spark module if you haven't done so
module load spark

# gett the Spark Master IP and Port
SPARK_MASTER=$(grep "Starting Spark master" *.err | cut -d " " -f 9)

# run the python program via command line
PYTHONSTARTUP=diy4.py pyspark --master $SPARK_MASTER


================================================
```

```
<DIY5> Submit Spark batch job using spark-submit
================================================

module load spark

SPARK_MASTER=$(grep "Starting Spark master" *.err | cut -d " " -f 9)

spark-submit --master $SPARK_MASTER $SPARK_HOME/examples/src/main/
python/pi.py 10

# or

spark-submit --master $SPARK_MASTER $SPARK_HOME/examples/src/main/
python/pi.py 10 2> /dev/null

==========================================
<DIY6> Wordcount example using spark-submit
==========================================

module load spark

SPARK_MASTER=$(grep "Starting Spark master" *.err | cut -d " " -f 9)

spark-submit --master $SPARK_MASTER $SPARK_HOME/examples/src/main/
python/wordcount.py spark_cluster.err > wc.result

cat wc.result
```