Whirr Instructions:

The following steps are to be run on your local Linux machine.

Install the SUN JDK for Linux if you don't have it installed already.

Create a directory within your Linux home directory called `cloud`

```
mkdir cloud; cd cloud
```

Download and Unpack Apache Hadoop

```
wget http://mirrors.ibiblio.org/apache//hadoop/common/hadoop-0.20.2/hadoop-0.20.2.tar.gz
tar xvzf hadoop-0.20.2.tar.gz ; rm hadoop-0.20.2.tar.gz
```

Download and Unpack Apache Whirr

```
wget http://mirrors.ibiblio.org/apache//whirr/whirr-0.7.0/whirr-0.7.0.tar.gz
tar xvzf whirr-0.7.0.tar.gz ; rm whirr-0.7.0.tar.gz
```

Setup all the required environment variables. Please obtain your Amazon Web Services Key ID and Secret Access Key from http://aws-portal.amazon.com/gp/aws/developer/account/index.html?action=access-key if required. Edit your ~/`.bashrc` file and add the following lines to the end of the file:

```
export AWS_ACCESS_KEY_ID=<<YOUR AWS ACCESS KEY ID>>
export AWS_SECRET_ACCESS_KEY=<<YOUR AWS SECRET ACCESS KEY>>
export JAVA_HOME=/usr/java/jdk*
export HADOOP_HOME=$HOME/cloud/hadoop-0.20.2
export WHIRR_HOME=$HOME/cloud/whirr-0.7.0
```

Let us also add some of the directories to the PATH variable to make access to commands easier. Edit your `~/.bash_profile` file to the following:

```
export PATH=$PATH:$HOME/bin:$JAVA_HOME/bin:$HADOOP_HOME/bin:$WHIRR_HOME/bin
```

Whirr requires you to create an RSA public/private key pair that will be used to ensure that you (and only you) will be able to access the VMs that you are about the provision on the cloud. To do this, you need to generate the public/private key pair using the following command (ensure you let it use the default options and leave the passphrase as blank):

```
ssh-keygen –t rsa
```

Once you have completed these steps, you may have to reboot your linux VM or start a new terminal session for the environment variable changes to take effect.

Let's configure Whirr using a template file:

```
cd $WHIRR_HOME/recipes
wget http://www.qatar.cmu.edu/~msakr/15319-s12/projects/project1/files/hadoop-ec2.properties.
```

Now you are ready to launch a Hadoop cluster in the Cloud! To launch a hadoop cluster with 4 Hadoop slave nodes use the following command:

```
whirr launch-cluster --config $WHIRR_HOME/recipes/hadoop-ec2.properties --instance-templates='1 hadoop-namenode+hadoop-jobtracker,4 hadoop-datanode+hadoop-tasktracker'
```

Now keep an eye on the output and make sure the cluster was created successfully. **From this point on, the clock is ticking and your cluster is chargeable by the hour**!

Once your cluster has been provisioned, lets configure Hadoop to use it. Type the following Command to set the Hadoop configuration to use the cluster provisioned by Whirr.

```
export HADOOP_CONF_DIR=~/.whirr/hadoop
```

As a security measure, many of the ports on your Hadoop cluster in EC2 are closed and Whirr automatically configures a proxy server which you can now run to access your Hadoop cluster. Run the following command (in a new terminal window):

```
sh ~/.whirr/hadoop/hadoop-proxy.sh
```

Your Hadoop cluster on Amazon EC2 is ready for use! Lets run a Pi estimator program with 8 mappers and 1 million samples per map to test it out:

```
cd $HADOOP_HOME

hadoop jar hadoop-0.20.2-examples.jar pi 8 1000000
```

When you are done with the cluster, **do not forget to terminate your instances!**

```
whirr destroy-cluster --config recipes/hadoop-ec2.properties
```

Go to https://console.aws.amazon.com/ec2/ to ensure that your cluster has in fact, been terminated.