1. Create S3 place holder for the new Amazon Machine Image

The following are executed on your **LOCAL** machine:

1. Download S3 command line tools from: [http://s3tools.org/s3cmd](http://s3tools.org/s3cmd)
2. Install the command line tools:
   - XXXX represents the version you have downloaded (currently it is 1.0.1).
   - Run the command:
     ```bash
tar -zxvf s3cmd-XXXX.tar.gz
     ```
3. Enter the extracted directory: `s3cmd-XXXX`
4. Configure the s3cmd tools.
   - Run the command:
     ```bash
     ./s3cmd --configure
     ```
   - When prompted enter the appropriate information. You will need your:
     AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY, later in the instructions referred to as `$ec2_access` and `$ec2_secret` If you don’t remember where to find them on the aws.amazon.com website, you can get them from your .bash_profile file that you have set up previously, or you can look at the info from the AWS Tutorial at: [http://www.cs.duke.edu/courses/fall11/cps216/TA_Material/03_AWS_inst.txt](http://www.cs.duke.edu/courses/fall11/cps216/TA_Material/03_AWS_inst.txt)
   - You should enter the path to the installation of GPG. This is usually `/usr/bin/gpg` on Linux systems and `/usr/local/bin/gpg` on Mac OS X.
   - For the rest of the information asked (encryption password etc.) press enter such that the default values will be used.
   - You will be asked if you want to test the connection with the info provided. You should type `Y` and get this message: “Success. Your access key and secret key worked fine : -)”. 
5. If there are no errors, you have configured the S3 tools correctly.
   - In your home directory you will have a `.s3cfg` file containing the setup information you entered (Do not delete it, otherwise you will have to reconfigure the S3 command line tools again).
   - If you want you can add the path to the s3cmd tools to your global `$PATH` using:
     ```bash
     export PATH=$PATH:/path/to/the/s3cmd-XXXX/folder/
     ```
6. Create a S3 bucket.
   - S3 storage is organized as follows, there are buckets that belong to different users and files are stored inside these buckets.
   - Buckets can contain only files, meaning no buckets nesting is supported.
   - When naming buckets use of the following naming convention is strongly encouraged: `edu_duke_$user_images` - where `$user` is your name or department login account.
   - Try to limit the special symbols in the bucket names, as the special symbols such as "/" or dots (“.”) can cause problems. Similarly, do not use capital letters.
   - In the following instructions your bucket (named by replacing `$user` appropriately in the `edu_duke_$user_images`) will be referred as `$s3_image_bucket`.
   - To create your bucket, run from the s3cmd-XXXX folder the command:
     ```bash
     ./s3cmd mb s3://$s3_image_bucket
     ```
7. List buckets.
   - To list all the buckets that you have in S3 you should run the command 
     `./s3cmd ls`
   - If you see the `$s3_image_bucket`, you have successfully created the bucket that later on will be used to store the newly created image.

2. AMI Creation

1. Install ec2-api-tools.
   - Download the tools from http://aws.amazon.com/developertools/Amazon-EC2/351
   - Run the command `unzip ec2-api-tools.zip`
   - Add the extract directory to your global `$PATH` (just like you did above for s3cmd).

2. Use the AWS Management Console or the ec2-api-tools to start an instance.
   - IMPORTANT: You should start with an S3 stored image, otherwise you will not be able to create the image with the steps below.
   - Instructions for the Management Console.
     - Log in to your EC2 Management Console https://console.aws.amazon.com/ec2/
     - Click Launch Instance and then go to the Community AMIs tab.
     - In the "Viewing:" drop box select "Instance-Store Images".
     - Select from the list or type in the available box the name of an image.
     - IMPORTANT: Use only images from sources you trust, otherwise there might be security issues. In the Section 5 of this document you will find a list of recommended AMIs (created by people from Duke) and with a description of what software they contain. In the TA session I have used "ami-73b97b1a".
     - In the next window you need to select the instance type. If you cannot choose m1.small or c1.medium it means that both the AMI you started from and the one you will create cannot be ran on the small/medium instances. This will result in more charges than necessary, so you should go back and change the starting AMI (the 64 bit images cannot be ran on small/medium).
     - In the next windows you should click Continue until you are asked to provide the EC2 Key Pair. You can then select the key pair from the drop down menu. For the rest of these instructions your EC2 private key pair will be referred as "$ec2".
     - In the next window select a security group or create a new one (the security group name is what you use in the harness as the cluster name). In the rest of the instructions this will be refered as the $group_name.

3. Connect to the instance.
   - In the management console look for the Public DNS of the instance you have started. It should look like XXXXXXXX.amazonaws.com. In the instructions below this will be refered to as $instance_public_ip.
   - By default, Amazon EC2 instances do not permit access on any ports. To access your instance with SSH, your instance must allow incoming traffic on port 22. To allow access from your computer you need your ip address (http://www.whatismyip.com/) and run from the ec2-api-tool-XXXX folder:
bin/ec2-authorize $group_name -p 22 -s your.ip.addr.ess/32

- To create the secure connection, run the command:
  
  
  ```
  ssh -i /path/to/$ec2.pem root@$instance_public_ip
  ```
  
  - We will be using the root account for these instructions. If you decide to change it, then do the appropriate replacement in the commands below and also make sure that the new account does not require a password to login.

4. Unless otherwise noted all of the following commands will be executed INSIDE the INSTANCE, NOT on your local machine.
5. You should now install the software that you need for your project.
   - Note that the already installed software of the starting AMI is found at /usr/local
   - If you need to do exports for the installation of your software make sure you add the export in the /root/.bash_profile — otherwise the export will not be already made when you start an instance with your AMI.
   - If you need to copy files from your computer to the instance you can use the command

     ```
     scp -i /path/to/$ec2.pem local_file root@$instance_public_ip:~
     ```

6. When all the modifications are complete proceed with the steps bellow.
7. Go to the local storage of the instance (/mnt) using the command cd /mnt

**IMPORTANT:** This directory will NOT be part of the image, so sensitive information should go there.
8. Create a directory for the image, later referred to as $image_dir.

   ```
   mkdir $image_dir
   ```
9. If ec2-ami-tools* are not installed, install them. Note: AMI tools not the API tools.*
   - Run the commands:

     ```
     mv /root/ec2-amigoools.zip /usr/local/ec2-ami-tools.zip
     unzip /usr/local/ec2-ami-tools.zip
     ```
   - If you do not find the zip, download from [http://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.zip](http://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.zip)
   - Export the home directory of the AMI Tools by running the command

     ```
     export EC2_AMITOOL_HOME=/usr/local/ec2-ami-tools/
     ```
10. From the LOCAL machine copy the following keys.
    - EC2_PRIVATE_KEY, refered below as $ec2_pk

      ```
      scp -i /path/to/$ec2.pem /path/to/$ec2_pk.pem root@$instance_public_ip:/mnt
      ```
    - EC2_CERT, refered below as $ec2_cert

      ```
      scp -i /path/to/$ec2.pem /path/to/$ec2_cert.pem root@$instance_public_ip:/mnt
      ```
    - **IMPORTANT:** Do not change the copy location, which is /mnt. If you copy your keys to directories which do not belong to the /mnt sub tree, then your keys will be stored in the image, thus presenting an opportunity for a security breach of your aws account!
11. Clean the private information from the instance.
    - To clean the shell history:

      ```
      rm /root/.bash_history
      ```
rm -rf /root/*.hist

- To clean the temporary files:
  
  rm -rf /tmp/*
  
  rm -rf /var/tmp/*

- To clean the logs. Note: If you have the folder /var/log/httpd make sure it does not get removed.
  
  rm -rf /var/log/*

- To clean login credentials:
  
  rm -rf ~/.ssh/authorized_keys

**IMPORTANT:** After this step you will **NOT** be able to login to the machine, so, do **NOT** close your ssh connection!

Delete all private keys and certificates (if you have copied anywhere else besides the /mnt directory). Do **NOT** delete anything from /mnt directory!!

12. To bundle the image you need to run the command

```bash
${EC2_AMITOOL_HOME}/bin/ec2-bundle-vol -d /mnt/$image_dir/ 
-k /mnt/$ec2_pk.pem -c /mnt/$ec2_cert.pem -u $account_number 
-r i386 -p $image_name
```

- `$ec2_pk` and `$ec2_cert` are the respective certificate private key and certificate public key that we uploaded in step 10.
- `$account` is your 12 digit Amazon account without the dashes ("-").
- `$image_name` is the name of your image.
- If you are creating a 64bit AMI you should replace i386 with x86_64.
- If the above command fails, just retry it again, till it succeeds.

13. To upload the bundle to S3 you have to run the command:

```bash
${EC2_AMITOOL_HOME}/bin/ec2-upload-bundle -b $s3_image_bucket -m 
/mnt/$image_dir/$image_name.manifest.xml -a $ec2_access 
-s $ec2_secret
```

- `$ec2_access` and `$ec2_secret` are your access key credentials, as discussed in section 1.4.
- If the above command fails, just retry it again, till it succeeds.

14. From the **LOCAL** machine you need to register the AMI. Go to the ec2-api-tool folder and run

```bash
bin/ec2-register $s3_image_bucket/$image_name.manifest.xml 
-K $ec2_pk -C $ec2_cert
```

- Save the image id that you will get from this command, we will refer to it as `$image_id`.

15. The image is ready and can be used. You should next follow the tutorial about downloading and using the new harness that allows you to specify the name of the AMI you want to use.

### 3. AMI Access

1. Following commands are executed on the **LOCAL** machine.
2. After you create your team’s AMI, you need to give access to every team member.
   
   - To do so, you need the Amazon account number of every user, no dashes, later refered to as `$other_account`.

   - From the ec2-api-tools folder you should run for each of the other members of your team, the following command:
bin/ec2-modify-image-attribute -l -a $other_account $image_id

3. The $other_user should be able to launch a cluster using the image.
   - If you replace $other_account with all you can make the AMI available to all Amazon users.

### 4. AMI Deletion

1. All commands are executed from the LOCAL machine.
2. De-register the AMI.
   
   `bin/ec2-deregister $image_id`

3. Delete from S3.
   
   `s3cmd del s3://$s3_image_bucket/*`

4. Image is gone completely.

### 5. List of Current AMIs stored on S3

<table>
<thead>
<tr>
<th>Image ID</th>
<th>Configuration</th>
<th>Owner</th>
<th>Image it is based on</th>
</tr>
</thead>
<tbody>
<tr>
<td>ami-73b97b1a</td>
<td>Same as ami-2817ff41 (the one you used for the previous assignments), but with Hadoop 0.20.203.0, Pig 0.9.0. and Ant</td>
<td>Fei</td>
<td>32-bit image</td>
</tr>
<tr>
<td>ami-81915de8</td>
<td>Based on ami-73b97b1a but with Hive 0.7.1</td>
<td>Rozemary</td>
<td>32-bit image</td>
</tr>
<tr>
<td>ami-a0ee12c9</td>
<td>Based on ami-58689831 but with Hadoop0.20.2, ant, make.</td>
<td>Fei</td>
<td>Image is based on ami-e291668b (basic 64-bit Fedora 14 image)</td>
</tr>
<tr>
<td>ami-e7bf7d8e</td>
<td>Based on ami-a0ee12c9 but with Hadoop0.20.203.0, Pig 0.9.0. and Ant</td>
<td>Fei</td>
<td>Image is based on ami-e291668b (basic 64-bit Fedora 14 image)</td>
</tr>
<tr>
<td>ami-41ea2628</td>
<td>Same as ami-e7bf7d8e but with Hive 0.7.1</td>
<td>Rozemary</td>
<td>Image is based on ami-e7bf7d8e</td>
</tr>
</tbody>
</table>
| ami-58689831 | Same as ami-f2df2f9b except that Hadoop 20 is patched to have ChainMapper and ChainReducer API. Notes:  
   - Make and lzo-devel library is installed  
   - Hadoop also contains the necessary lzo jar for lzo compression. | Harold | Image is based on ami-f2df2f9b |
| ami-e642bf8f | Same as ami-58689831 except that HBase 0.90 is installed.                   | Harold | Image is based on ami-e291668b (basic 64-bit Fedora 14 image) |
| ami-f2e9159b | Same as ami-58689831 except that HBase 0.89 (with Table Snapshot feature) is installed. | Harold | Image is based on ami-e291668b (basic 64-bit Fedora 14 image) |
| ami-7e0ff17  | Ubuntu 10.04 LTS  
   Java 1.6.0_22  
   Hadoop 0.21.0  
   Perl  
   Perl libraries - XML::Simple  
   Ganglia  
   BTrace | Harold | Image is based on ami-ac07f2c5 (Official image of 64-bit Ubuntu 10.04 LTS (Lucid Lynx) Server) |
<table>
<thead>
<tr>
<th>sysstat</th>
<th>gcc</th>
<th>unzip</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Note:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▪ You must login as ubuntu user instead of root (the official image is set this way as they want to prevent people for working as root)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>ami-84e412ed</strong></th>
<th>Same as <strong>ami-7e0ff17</strong> except that it has Hadoop 20 Warehouse <a href="https://github.com/facebook/hadoop-20-warehouse">https://github.com/facebook/hadoop-20-warehouse</a> is installed.</th>
<th>Harold</th>
<th>Image is based on <strong>ami-7e0ff17</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Notes:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▪ Use the scripts found in harness/hadoop_ec2_contrib_bin_ubuntu to launch a hadoop cluster</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>▪ 64-bit image</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Other useful information and resources:

- When you see something in the above instructions that is preceded by the sign ‘$’ it means that you should replace that something with your specific value. Do not write the ‘$’ character!
- If you have a Mac OS X system you should install GPG by following the tutorial from here: [http://macgpg.sourceforge.net/docs/howto-build-gpg-osx.txt.asc](http://macgpg.sourceforge.net/docs/howto-build-gpg-osx.txt.asc).
- If you have a Windows system you should use GPG4Win from here: [http://gpg4win.org/](http://gpg4win.org/).
- It is recommended to start from an AMI on 32 bit.
- If you start from one of the following 32 bit images: **ami-73b97b1a** or **ami-81915de8** you will have the AMI Tools installed. You can look in the `/usr/local/ec2/amitools` to see the aliases of the commands. As a consequence, instead of `$EC2_AMITOOL_HOME/bin/ec2-bundle-vol` you will write just `bundlevol` and instead of `$EC2_AMITOOL_HOME/bin/ec2-upload-bundle` you will write `uploadbundle` (you can use them no matter which is your pwd and, obviously, you would still pass all the parameters like –k, -c etc.).
- You can click on the S3 tab from the AWS Management Console and you can see your S3 buckets and what they contain. After uploading an AMI you should see the manifest file and all the parts the image is made of.
- Amazon has a pretty good and comprehensive documentation that you might find very useful: