CIS 450/550 : Database and Information Systems

Setting up Amazon Web Services (AWS)

This document briefly summarizes how to get started in using Amazon's Elastic Compute Cloud and SimpleDB services. It is based on the Amazon "Getting Started" documentation but specialized to the needs of our class. SimpleDB provides a "key/value store" abstraction – think of it as a giant persistent hash table "in the cloud." Elastic Compute Cloud provides a large cluster of virtual Linux (or Windows) machines, with the ability to run Web servers and/or to run Hadoop MapReduce. RDS is a web service that makes it easy to set up, operate, and scale a relational database in the cloud and gives you access to the capabilities of a familiar MySQL, Oracle or Microsoft SQL Server database engine.

Initial Setup

1. Go to `aws.amazon.com`, click **Sign Up Now** and enter your user information and a credit card.

2. Then go to `aws.amazon.com/awscredits` and enter your AWS credit code (given out by email).

This should give you $100 in credits towards AWS resources. Beyond this your credit card will be charged. Note that only one code will work per account.

Elastic Compute Cloud (EC2)

Getting Started: Required Software for Your Local Machine

You will need the following:

- ssh client, which will let you remotely log into Amazon machines
  - Linux: ssh should be installed by default; if not you will have to run `yum install openssh` or the equivalent
  - Mac OS X: ssh is installed by default; if you have an old version you might need to run `fink update`
  - Windows:
    - Install cygwin (www.cygwin.com) and select **OpenSSH** in the **Networking** tab in the setup.
    - You may also want to install the PuTTY ssh terminal software, http://www.chiark.greenend.org.uk/~sgtatham/putty/, along with PuTTYGen
- Sun (not OpenJDK!) Java 6, http://java.sun.com/javase/6/
- Eclipse: Please install Eclipse 3.5 or higher

Getting Started: Creating AWS Account

1. Go to `aws.amazon.com`, click **Sign Up Now**

2. Find the link for **Amazon Elastic Compute Cloud** and click on it, then click on the button **Sign Up For Amazon EC2**

3. Find the menu **Account** and select **Security Credentials**

4. Click on the **X.509 Certificates** tab and create a new certificate; download it and rename to `cert.pem`.

5. Create an **Access Key**. Download the resulting private key file as `access.pem`.

6. Click on the **CloudFront Key Pairs** area. Then click on **Create Key Pair**. Give a name as `login.pem`.

7. Make an `.ec2` subdirectory under your home directory, and copy all downloaded certificates there.

8. Look up your **Account Number** under the same **Your Account | Security Credentials** area. This number, with dashes removed, is your account ID.

Getting Started: Client-Side Setup for AWS

Set up your ssh client to use the private key:

- **If you are using PuTTY**, you will need to run the accompanying PuTTYGen, then choose **Conversions | Import Key**, and select your `login.pem` file. Optionally choose a password to add, then save as a PuTTY `.ppk` file.

- **If you are using the regular ssh client**, you can copy the `login.pem` file to the new name and location `~/.ssh/id_rsa`, then later use `ssh` *{xyz}* where {xyz} is the Amazon virtual machine hostname you want to log into. Or you can use “`ssh -i ~/.ec2/login.pem`” to directly use the file wherever it is.

Next, download the Amazon AWS command-line tools from http://s3.amazonaws.com/ec2-downloads/ec2-api-tools.zip.

1. Unzip it to a particular path in your filesystem.

2. Set the EC2_HOME environment variable to point to the tools:

    a. Linux – bash: Add the following to `~/.profile`:
    `export EC2_HOME=`{path}

    b. Linux – csh: Add the following to `~/.cshrc`:
    `setenv EC2_HOME` {path}

    c. Windows XP: go into **Control Panel | System**, click on the **Advanced** tab and choose **Environment Variables**, then create a **User variable** `EC2_HOME` with the appropriate path.

   d. Windows Vista/7: go into **Control Panel | System and Maintenance | System | Advanced System Settings** and choose **Environment Variables**, then create a **User variable** `EC2_HOME` with the appropriate path.

3. Repeat the above procedure to set the variable `PATH` to point to the `bin` subdirectory of the tools you just installed.

4. If necessary, repeat the above to set JAVA_HOME to point to the base directory of your JDK, and the `PATH` to include the `bin` directory within the Java install.

5. Set the variable `EC2_CERT` to point to the path of the EC2 X.509 Certificate file.

6. Set the variable `EC2_PRIVATE_KEY` to point to the path of the EC2 Access Key private key certificate (not the keypair one used for ssh).

Getting Started: Configuring a Default Security Group

Go to **Resources | AWS Management Console** (under **Development Tools**) and sign in.

- Choose **Security Groups** under **Networking and Security**

- Select the **default** security group

- The default permissions allow for unfirewalled access among Amazon EC2 nodes, but no access from outside.

- We need to enable the HTTP protocol, which operates over TCP. Click on the drop-down and select HTTP. It should fill in “TCP”, “80”, “80”, “0.0.0/0”. Click **Save.**

- Repeat but select SSH. It should fill in “TCP”, “22”, “22”, “0.0.0/0”. Click **Save.**

Launching an AWS Instance

Go to **Resources | AWS Management Console** (under **Development Tools**) and sign in.

- Click on **Instances** in the sidebar.

- Select **Launch Instance**. Now you need to choose a type of virtual machine. You’ll probably want a **fedora** Linux install, probably the LAMP Web Starter (which has Apache Web server and PHP).

- Choose the number of machine instances you need, and the type of machine. You’ll probably want **Small**. Leave **Availability Zone** as it is.

- Leave the **Advanced Instance Options** as they are; just click **Continue**.

- Select your `login` keypair.

- From the **Configure Firewall** screen, go to **Choose one or more of your existing Security Groups**. Select your **default** group.

- In the review screen, choose **Launch**. Recall that you will be billed on an hourly basis, so don’t forget to turn it off later! Click **Close**.

- From the AWS Management Console, to the **Instances** area and wait for the main **My Instances** window to indicate the instance is ready.

- Note the **Public DNS** name for your new instance. You might have to increase the column width, or left-click on the entry and look at the details the bottom of the pane.

Connecting to an AWS Instance

You can connect to a Linux AWS instance using ssh. Here are the Amazon instructions for doing so.

1. In a command line shell, change directories to the location of the private key file that you created when you launched the instance. This is probably `.ec2` or `.ssh`.

2. Use the `chmod` command to make sure your private key file isn't publicly viewable. For example, if your file were the default of `login.pem`, you would enter:

   ```
   chmod 400 login.pem
   ```

3. Connect to your instance using the instance's public DNS name (which you should have recorded earlier). For example, if the key file is `login.pem` and the instance's DNS name is ec2-75-101-230-211.compute-1.amazonaws.com, use the following command.

   ```
   ssh -i ~/.ec2/login.pem root@ec2-75-101-230-211.compute-1.amazonaws.com
   ```

   You'll see a response like the following.

   ```
   The authenticity of host 'ec2-75-101-230-211.compute-1.amazonaws.com
   (75.101.230.211)' can't be established.
   RSA key fingerprint is fc:8d:0c:eb:0e:a6:4a:6a:61:50:00:c4:d2:51:78:66.
   Are you sure you want to continue connecting (yes/no)? yes
   ```

4. Enter `yes`.

   You'll see a response like the following.

   ```
   Warning: Permanently added 'ec2-75-101-230-211.compute-1.amazonaws.com' (RSA)
   to the list of known hosts.
   ```

You're now logged in as root and can work with the instance like you would any normal server. Just remember that you are being billed while the server is alive!

Log out using `exit` or `logout`.

Terminating an EC2 Instance

Please note that you will be billed for AWS instances as they are alive, so you will want to terminate them when they aren't in direct use. Here are the Amazon instructions.

1. In the AWS Management Console, locate the instance in your list of instances on the **Instances** page.

2. Right-click the instance, and then click **Terminate**.

3. Click **Yes, Terminate** when prompted for confirmation.

   Amazon EC2 begins terminating the instance. As soon as the instance status changes to `shutting down` or `terminated`, you stop incurring charges for that instance.

Relational Database Service (RDS)

Connecting to a MySQL database instance:

- You need to connect to a database on a MySQL DB instance using MySQL monitor commands. Use the MySQL Workbench (`dev.mysql.com/downloads/tools/workbench`) which is a GUI based interface to MySQL. MySQL help can be found on `dev.mysql.com/doc/`
- Type the following command at a command prompt on a client computer to connect to a database on a MySQL DB instance using the MySQL monitor.

```
mysql -h <endpoint> -P 3306 -u <mymasteruser> -p <password>
```

- You should now see a SQL prompt where you can enter your queries.

Connecting to an Oracle database instance:

- To connect to the sqlplus client, use ssh to log into eniac. Instructions are available here.
- At the command prompt enter the following command:

```
sqlplus
'cis550student@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=cis550hw1.c5cld00ma
sig.us-east-1.rds.amazonaws.com)(PORT=1521))(CONNECT_DATA=(SID=IMDB)))'
```

- You will be prompted to enter a password. The password for this database is `cis550hw1`
- You should now see a SQL prompt where you can enter your queries.

SimpleDB

1. On the link for **Amazon SimpleDB** (`aws.amazon.com/simpledb`)and click on it, then click on the button **Sign Up For Amazon SimpleDB**
2. Once you receive an email notifying you that you have signed up for SimpleDB, you need to download the SimpleDB Scratchpad. Download, save, and extract the .zip file from http://developer.amazonwebservices.com/connect/entry.jspa?externalID=1137&categoryID=189
3. Go into the extracted `webapp` directory and click on `index.html`
4. In another browser window, go to `aws.amazon.com` and choose **Account**, then **Security Credentials**.
5. Find the **Access Credentials** and make sure you have selected **Access Keys**. The AWS Access ID should show up in the **Access Key** column.
6. Click on the link in the box that leads you to the legacy security credentials page (`portal.aws.amazon.com/gp/aws/securityCredentials`). To view the Secret Access key, make sure you are on the Access Key tab and click **Show**.
7. Copy and paste the Access ID and Secret Access key from the AWS screen into the appropriate columns in the Scratchpad window.
8. Now we need to create a domain – think of this as a very simple database that can hold a single table. Choose **CreateDomain** from the **Explore API** list box. Fill in a name like `cis399` for the **Domain Name**. Click **Invoke Request**. You should get back a fragment of XML.

9. Select **ListDomains** from the Scratchpad **Explore API** list box. You should see an XML fragment with a `ListDomainsResult` with `DomainName` `cis399`.

10. To add some sample data:

    a. Choose **PutAttributes** in the Scratchpad's **Explore API** list box.

    b. Set **Domain Name** to `cis399`

    c. Create an **Item Name** for your first entry – this is the **key** and should be unique.

    d. Add an attribute **Name** and **value**. For each additional attribute, hit the green "+" key for another entry.

    e. Add a series of name-value pairs for the different attributes or properties of the item.

    f. Click **Invoke Request**

11. To see the data:

    a. Choose **Select** from the Scratchpad's **Explore API** list box.

    b. Set **Domain Name** to `cis399`

    c. Type in `select * from cis-399`