Homework #5 Amazon Web Services (AWS) with PHP

This semester we are allowing all students to explore cloud computing as offered by Amazon’s Web Services. Using the instructions below one can establish a service at AWS. Once established, you will be able to move your PHP back-end program developed for Assignment #6 to your AWS instance and have it executed there.

1. Sign up for AWS

To sign up for AWS, you need a credit card. If you do not have one, buy a $25 American Express Gift card at Ralphs or other grocery store.

To sign up go to:

http://aws.amazon.com

and click on Sign In to the Console. Follow the instructions to create your AWS account using the "AWS Free Usage Tier".

After you are signed up, from the drop down next to your name, select My Account. In the Account Settings section, you will notice the Account Id. This is the AWS Account ID to be used next when signing up for AWS Educate.

Please note that many of the URLs listed from now on will only be available if you are signed up to AWS.

2. Sign up for AWS Educate

To sign up for AWS Educate and get a $100 credit (USC is a member of AWS Educate) go to:

http://aws.amazon.com/education/awseducate

Click on the Join AWS Educate Today button.

Click on the Apply for AWS Educate for Students button.

On Step 1, select I am a Student.

On Step 2, fill out the form appropriately. Leave Promocode field empty and go to the next step.
On Step 3, Fill out the AWS Account ID field, enter the ID from your initial AWS signup.
Click **Next** and finish the sign-up process.

After your application is reviewed and approved, you will receive a welcome e-mail from AWS Educate Support, which includes details about the $100 promotional credit, as shown below:

**AWS Promotional Credit**
It's our pleasure to issue you an Amazon Web Services (AWS) promotional credit code in the amount listed below.
Credit Amount: $100.00
Credit Code: PC26S89AM7I7EMY

3. **Apply the $100 credit**

Once you have completed signing up for both AWS and AWS Educate, login to your AWS account.

Then go to the URL:


Click **Redeem Credit**. Enter the Credit Code that was provided to you in the e-mail in the Promo Code field. Enter the displayed characters for the Security Check, and click the **Redeem** button. The Promo Code details (expiration date, credit used, credit remaining, etc.) will be displayed in the table at the bottom.

4. **Provide Credit Card or Gift Card Information**
   - In the top menu, click on your name and select **My Billing Dashboard** in the drop-down menu.

   ![Billing Dashboard Menu](image)

   - In the left menu select **Payment Methods**
   - Click on **Add a Card** button
   - Provide your card information (Credit Card Number, Name of Cardholder, Expiration Date)
• Optionally Edit your billing address.

5. Set up the Default Elastic Beanstalk Application
   • Click the top left menu Services
   • From the list of Amazon Web Services, select Elastic Beanstalk, under Compute.

   ![Compute Services Menu](image)

   • Select Create New Application in the top right, right underneath your account name, and follow the Wizard.
   • In the Application name field, enter a name for your application. Click Create.
   • In the Environment section click on the Create One Now hyperlink

   ![Environment Create Dialog](image)

   • In the Choose an environment tier dialog select Web server environment and click on Select button.
In the Environment Information section, select a Domain (use the default or check availability of your own subdomain of elasticbeanstalk.com). Click on “Check availability” button. Your URL should be green. Otherwise you should change the environment URL.

In the Base configuration section, choose the Preconfigured platform, and one of the following options in the drop-down list:
- Platform: PHP
- Environment Type: Single Instance.
• In the **Application Code** section, select **Sample application**.

  ![Application Code Screenshot](image)

• Click **Create environment**.

• After a minute or so the “Creating `<environment-name>`” dialog appears, with the message “This will take a few minutes…”

![Creating Environment Dialog](image)

You will need to wait for several minutes as your Linux + Apache + PHP instance is created and launched. You will see several messages appear as the instance is being created and deployed. a **rotating wheel** next to the **“Monitor” button**. Once creation and launch are completed, you will see the wheel turn into a green round circle with a check mark in the middle.
PHP Instance Dashboard

Beside "<YourEnvironment>" subtitle there is a [URL] such as YourAppName-env.elasticbeanstalk.com, click on it. You should see the "Congratulations" page. If you see it as shown below, your application and environment have been created properly. Then go back to Elastic Beanstalk console.

Congratulations!
Your AWS Elastic Beanstalk PHP application is now running on your own dedicated environment in the AWS Cloud
You are running PHP version 7.1.7

What's Next?
- AWS Elastic Beanstalk overview
- Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git
- Using Amazon RDS with PHP
- Customizing the Software on EC2 Instances
- Customizing Environment Resources

AWS SDK for PHP
- AWS SDK for PHP home
- PHP developer center
- AWS SDK for PHP on GitHub

PHP Sample Application
6. Upload your PHP application

Develop your PHP server application, and make sure that you name the file `index.php`. Compress the file with ZIP so that the resulting file is named `index.php.zip`. On a Mac, you can right click the file and choose compress. On Windows, there are several free programs, such as 7-Zip or FreeZip, etc. that you can use.

From the Elastic Beanstalk console, select your environment and click on the **Upload and Deploy** button.

The **Upload and Deploy** popup will display. Enter a Version **label** (e.g., version 1.0). Click on the **Choose File** button and select the `index.php.zip` file. Then click on **Deploy** button.

Again, wait several minutes for the *rotating wheel* to finish and the green circle with checkmark to appear. Click again on the link “*YourAppName-env.elasticbeanstalk.com*”. Check that your PHP app is running correctly.

- **Important Note:** in the future if you want to upload an updated version of `index.php`, you should enter a different version label. Otherwise, you will get an error. It is recommended that you use increasing version numbers (2.0, 2.1, 3.0, etc.)

To browse how your web service and application has been set up, go to the Services console (which is in the top menu of Elastic Beanstalk console) and select the **Compute EC2** Service. You should see that you have **1 Running Instance**, **1 Volume**, **1 Elastic IP**, zero or **1 Load Balancer** (managing port 80), and **1 (or 2) Security Groups**.

For example, you can type this code and save it as `index.php` and then compress it as `index.php.zip`:

```php
<?php phpinfo(); ?>
```

After you deploy your code, your website will look like this.
7. Set up Exploring Your Instance (Optional)

If you want to explore your Instance and create your own domain-based URL with SSH control, you can add the following steps.

7.1 Get and Setup SSH

Once the PHP app with SSH-enabled environment is running, you can get access using SSH. You can use SSH on a Mac running MacOS, or Putty when running on Windows.

On a Mac, SSH is built into MacOS and can be accessed through the Terminal app and there is no additional setup needed.

On a Windows PC, you will need to download the complete PuTTY distribution at:

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
You should download the file `putty.zip` that contains all the binaries, including PuTTYgen as see in this snapshot from the website above:

### Alternative binary files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>putty.exe</td>
<td>SSH and Telnet client itself</td>
</tr>
<tr>
<td>putty.exe [x64]</td>
<td>ssh and telnet client itself</td>
</tr>
<tr>
<td>putty.zip</td>
<td>ZIP archive of all the above</td>
</tr>
</tbody>
</table>

PuTTY needs additional setup as it needs to use a converted version of the private key. The instructions on how to perform such conversion are available here:


The major step is to use PuTTYgen to convert your private key format (.pem) generated by Amazon EC2 into the required PuTTY format (.ppk).

### 8.2 Create a Key Pair

- From the Services drop down, under the Compute section, select the EC2.
Under **NETWORK AND SECURITY** select **Key Pairs**.
Click on the button **Create Key Pair**.
Enter a name like *phphosts* (you must have your own random name!) and click on **Create**.
A download of your private key should start automatically. Save the key, like *phphosts.pem*, in an appropriate location.

**8.2.1 Associate your Instance to the Key Pair**

You now need to associate your Instance with the just created key pair.
Select the **Elastic Beanstalk** under **Services**.
Select your application environment, but clicking anywhere in the “green” rectangle.
Click on **Configuration** on the left menu.
Click on the settings wheel next to **Instances**.
Select the key pair you just created for the **EC2 key pair** field. Click **Refresh**.

![EC2 key pair](image)
• Hit **Apply** and then **Save** and wait for a few minutes for the configuration changes to take place. You may get INFO, WARN and sometimes SEVERE messages during this time. Wait until the update of the environment has completed, and **Health** is back to **Ok**.

• Go back to your EC2 instance after some time and check under **Key Name**, you should now see your associated key pair.

![](image)

8.3 Open port 22

To open port 22, which is needed by SSH, follow these steps:

1. In the EC2 Management Console, click on **Instances**.
2. Under **NETWORK & SECURITY**, click on **Security Groups**.
3. Select the security group (present as a link) configured for your instance.
4. For the security group, edit the "Inbound rules" (Inbound tab present on the bottom of the pane) by clicking the **Edit** button.
5. Add a new rule for **Type** = SSH, **Protocol** = TCP, **Port Range** = 22, **Source** = Custom 0.0.0.0/0. Click **Save**.

![](image)

8.3.1 Errors when Connecting

If you fail to either open port 22 or associate your instance to a key pair, you will get an error popup when you try to **Connect to Your Instance** using EC2 Dashboard >> INSTANCES >> Instances >> select instance >> Connect, as show in the picture below.
8.4 Access your Linux Instance with SSH

- To see how to launch your SSH client go to Services and select EC2.
- Under the INSTANCES section in the navigation pane on the left, select Instances.
- Select your instance in the table (the check box turns blue) and select the Connect button next to Launch Instance.
- The Connect To Your Instance popup will display. Select the radio button A Standalone SSH Client. Notice the hyperlink “connect using PuTTY” (see section 7.4.2). See the snapshot below, showing Elastic IP connection string.

8.4.1 Mac running OS X / ssh

Change the permission of phphosts.pem first:

```bash
chmod 400 phphosts.pem
```
On a Mac you will need to enter a command like this one (when using Public DNS):

```
ssh -i phphosts.pem ec2-user@ec2-54-235-60-138.compute-1.amazonaws.com
```

or this one (when using Elastic IP)

```
ssh -i phphosts.pem ec2-user@204.236.235.251
```

type `yes`, when asked. Make sure that you are executing the ssh command in the same folder that contains the key. You should see output like this one (using Public DNS):

```
$ ssh -i "phphosts.pem" ec2-user@ec2-204-236-235-251.compute-1.amazonaws.com
Last login: Tue Oct 27 16:22:06 2015 from 159.83.115.214

Amazon Linux AMI

This EC2 instance is managed by AWS Elastic Beanstalk. Changes made via SSH WILL BE LOST if the instance is replaced by auto-scaling. For more information on customizing your Elastic Beanstalk environment, see our documentation here:

http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/customize-containers-ec2.html

[ec2-user@ip-10-30-13-153 ~]$`

Or output like this (using Elastic IP):

```
marcopapa$ ssh -i "phphosts.pem" ec2-user@204.236.235.251
The authenticity of host '204.236.235.251 (204.236.235.251)' can't be established.
ECDSA key fingerprint is SHA256:j4I3q03uqRCH9uYP5EwZTIWkk8EtmeLEf6Ve0jnYctDY.
Are you sure you want to continue connecting (yes/no)? yes
```

Warning: Permanently added '204.236.235.251' (ECDSA) to the list
This EC2 instance is managed by AWS Elastic Beanstalk. Changes made via SSH WILL BE LOST if the instance is replaced by auto-scaling. For more information on customizing your Elastic Beanstalk environment, see our documentation here:
http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/customize-containers-ec2.html
[ec2-user@ip-10-30-13-153 ~]$ 

You can find more info here:


8.4.2 PC running Windows / PuTTY

In the popup windows titled Connect To Your Instance, click on Connect using PuTTY. You will be redirected to the URL.


Follow the steps under Starting a PuTTY Session to connect to your the Linux instance using PuTTY. The first time you connect by clicking Open to start the session, PuTTY displays a PuTTY Security Alert dialog box, as show in the following snapshot. Click the Yes button.
Once connected, PuTTY will open, and log you in, as shown in the next snapshot.

As with SSH, you can either use tout Public DNS or your Elastic IP to log in.

### 8.5 Explore

You can now explore your Instance. When you log in with SSH, your account home directory will be located at:

/home/ec2-user

That folder is empty and is not where your apache / PHP files are. Run `ps -ax`; and you should see several instances of `httpd`:

```
  1940 ?    S   0:00 /usr/sbin/httpd -D FOREGROUND
```
1941 ?  S  0:00  /usr/sbin/httpd -D FOREGROUND
1942 ?  S  0:00  /usr/sbin/httpd -D FOREGROUND
1944 ?  S  0:00  /usr/sbin/httpd -D FOREGROUND
1945 ?  S  0:00  /usr/sbin/httpd -D FOREGROUND

To see your mounted volumes, run 'df -h':

[ec2-user@ip-10-158-56-168 sbin]$ df -h
Filesystem   Size  Used  Avail  Use% Mounted on
/dev/xvdal      7.9G   1.3G   6.6G   17% /
tmpfs           298M     0  298M    0% /dev/shm
[ec2-user@ip-10-158-56-168 sbin]$  

To see your document root, run 'ls /var/www/html':

[ec2-user@ip-10-158-56-168 html]$ ls -l /var/www/html
lrwxrwxrwx 1 root root 16 Mar 22 20:38 /var/www/html -> /var/app/current
[ec2-user@ip-10-158-56-168 html]$  

To see your uploaded index.php file:

[ec2-user@ip-10-158-56-168 sbin]$ cd /var/www/html
[ec2-user@ip-10-158-56-168 html]$ ls -l
total 4
-rw-r--r-- 1 webapp webapp 3723 Feb  5 02:09 index.php
[ec2-user@ip-10-158-56-168 html]$  

To see your php.ini file, 'ls -l /etc/php.ini':

[ec2-user@ip-10-158-56-168 html]$ ls -l /etc/php.ini
-rw-r--r-- 1 root root 65782 Mar 22 20:38 /etc/php.ini
[ec2-user@ip-10-158-56-168 html]$  

Have fun exploring AWS!!