

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.

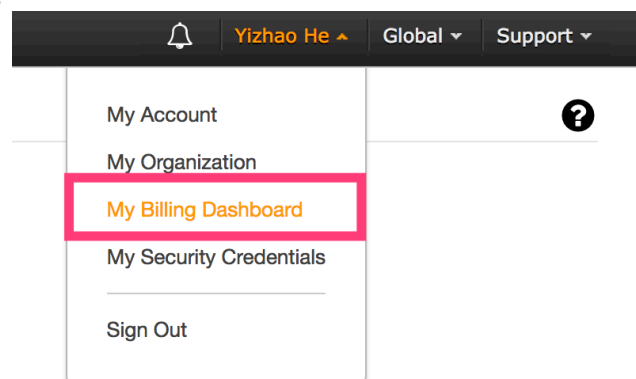
Then go to the URL:

<http://aws.amazon.com/awscredits/>

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.

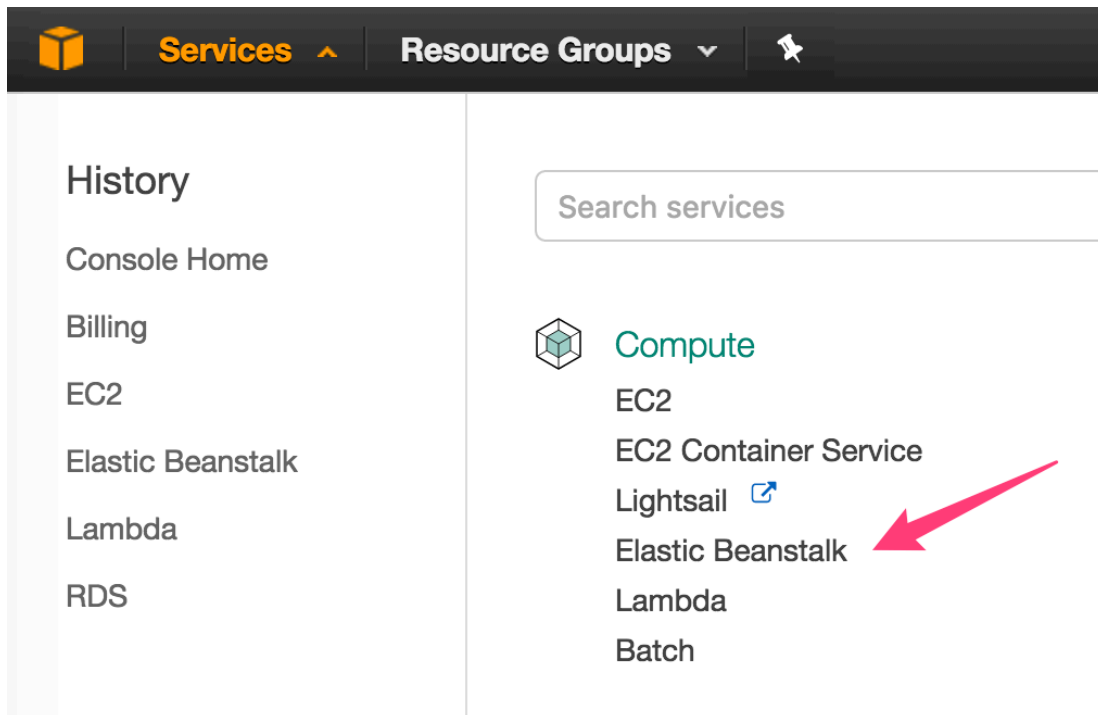


- In the left menu select **Payment Methods**
- Click on **Add a Card** button
- Provide your card information (Credit Card Number, Name of Cardholder, Expiration Date)

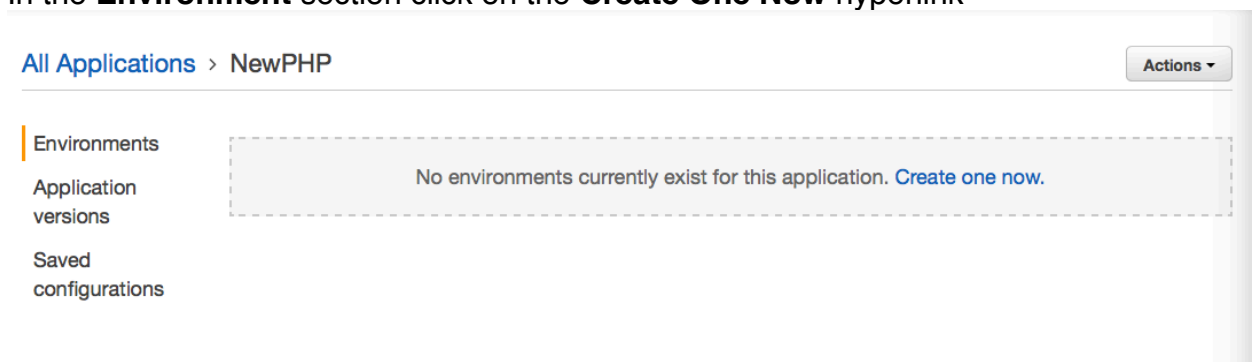
- Optionally Edit your billing address.

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**.



- 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



- In the **Choose an environment tier** dialog select **Web server environment** and click on **Select** button.



Select environment tier

AWS Elastic Beanstalk has two types of environment tiers to support different types of web applications. Web servers are standard applications that listen for and then process HTTP requests, typically over port 80. Workers are specialized applications that have a background processing task that listens for messages on an Amazon SQS queue. Worker applications post those messages to your application by using HTTP.

Web server environment ☒

Run a website, web application, or web API that serves HTTP requests.
[Learn more](#)

Worker environment ☐

Run a worker application that processes long-running workloads on demand or performs tasks on a schedule.
[Learn more](#)

[Cancel](#)

[Select](#)

- 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.

Create a new environment

Launch an environment with a sample application or your own code. By creating an environment, you allow AWS Elastic Beanstalk to manage AWS resources and permissions on your behalf. [Learn more](#)

Environment information

Choose the name, subdomain, and description for your environment. These cannot be changed later.

Application name NewPHP

Environment name

Domain

[Check availability](#)

csci571-php.us-east-1.elasticbeanstalk.com **is available.**

Description

- 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.**

Base configuration

Tier Web Server (Choose tier)

Platform ☒ Preconfigured platform
 Platforms published and maintained by AWS Elastic Beanstalk.

PHP

☐ Custom platform NEW
 Platforms created and owned by you. [Learn more](#)

-- Choose a custom platform --

Application code ☒ Sample application
 Get started right away with sample code.

☐ Existing version
 Application versions that you have uploaded for NewPHP.

-- Choose a version --

☐ Upload your code
 Upload a source bundle from your computer or copy one from Amazon S3.

ZIP or WAR

- Click **Create environment**.
- After a minute or so the “**Creating <environment-name>**” dialog appears, with the message “This will take a few minutes...”

All Applications > NewPHP > Newphp-env (Environment ID: e-ethjxs5uzy, URL: csc0571-php.us-east-1.elasticbeanstalk.com) Actions

Creating Newphp-env
 This will take a few minutes...

```

1:45pm Environment health has transitioned from Pending to Ok. Initialization completed 9 seconds ago and took 4 minutes.
1:43pm Added Instance [i-037c397c9487d2d3a] to your environment.
1:42pm Waiting for EC2 instances to launch. This may take a few minutes.
1:41pm Environment health has transitioned to Pending. Initialization in progress (running for 16 seconds). There are no instances.
1:41pm Created EIP: 54.83.201.175
1:40pm Created security group named:
       aws-eb-e-ethjxs5uzy-stack-AWSEBSecurityGroup-1FCZQNMU89H8M
1:40pm Using elasticbeanstalk-us-east-1-231003652681 as Amazon S3 storage bucket for environment data.
1:40pm createEnvironment is starting.

```

Learn More

[Get started using Elastic Beanstalk](#)
[Modify the code](#)
[Create and connect to a database](#)
[Add a custom domain](#)

Featured

[Create your own custom platform](#)

Command Line Interface (v3)

[Installing the AWS EB CLI](#)
[EB CLI Command Reference](#)

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.



Health

Ok

Causes

Running Version

Sample Application

Upload and Deploy



Configuration

64bit Amazon Linux 2017.03
v2.5.0 running PHP 7.1

Change

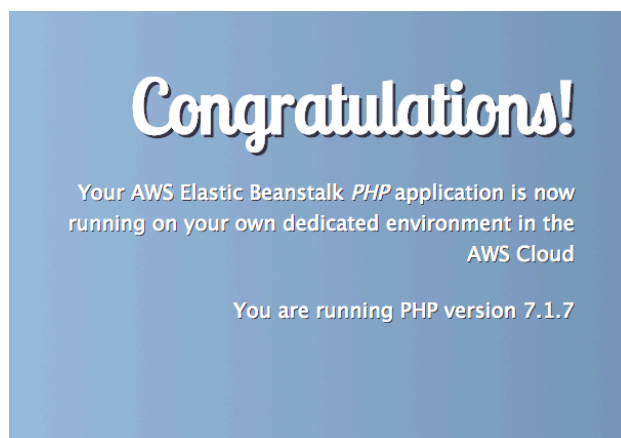
Recent Events

Show All

Time	Type	Details
2017-10-17 13:45:32 UTC-0700	INFO	Successfully launched environment: Newphp-env
2017-10-17 13:45:06 UTC-0700	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 9 seconds ago and took 4 minutes.
2017-10-17 13:43:06 UTC-0700	INFO	Added instance [i-037c397c9487d2d3a] to your environment.
2017-10-17 13:42:48 UTC-0700	INFO	Waiting for EC2 instances to launch. This may take a few minutes.
2017-10-17 13:41:06 UTC-0700	INFO	Environment health has transitioned to Pending. Initialization in progress (running for 16 seconds). There are no instances.

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.



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](#)

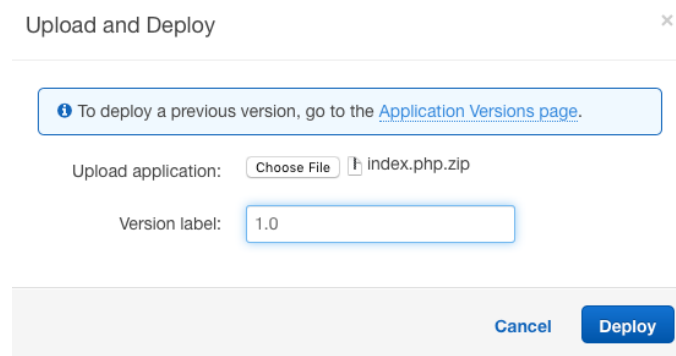
PHP Sample Application

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 phpinfo();?>
```

After you deploy your code, your website will look like this.

System	Linux ip-172-31-27-78 4.14.62-65.117.amzn1.x86_64 #1 SMP Fri Aug 10 20:03:52 UTC 2018 x86_64
Build Date	Jun 28 2018 17:24:12
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php-7.2.conf:/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php-7.2.d
Additional .ini files parsed	/etc/php-7.2.d/10-opcache.ini, /etc/php-7.2.d/20-bcmath.ini, /etc/php-7.2.d/20-bz2.ini, /etc/php-7.2.d/20-calendar.ini, /etc/php-7.2.d/20-ctype.ini, /etc/php-7.2.d/20-curl.ini, /etc/php-7.2.d/20-dom.ini, /etc/php-7.2.d/20-exif.ini, /etc/php-7.2.d/20-fileinfo.ini, /etc/php-7.2.d/20-ftp.ini, /etc/php-7.2.d/20-gd.ini, /etc/php-7.2.d/20-gettext.ini, /etc/php-7.2.d/20-iconv.ini, /etc/php-7.2.d/20-intl.ini, /etc/php-7.2.d/20-json.ini, /etc/php-7.2.d/20-mbstring.ini, /etc/php-7.2.d/20-mysqlnd.ini, /etc/php-7.2.d/20-odbc.ini, /etc/php-7.2.d/20-pdo.ini, /etc/php-7.2.d/20-pgsql.ini, /etc/php-7.2.d/20-phar.ini, /etc/php-7.2.d/20-posix.ini, /etc/php-7.2.d/20-shmop.ini, /etc/php-7.2.d/20-simplexml.ini, /etc/php-7.2.d/20-soap.ini, /etc/php-7.2.d/20-sockets.ini, /etc/php-7.2.d/20-sqlite3.ini, /etc/php-7.2.d/20-sysvmsg.ini, /etc/php-7.2.d/20-sysvsem.ini, /etc/php-7.2.d/20-sysvshm.ini, /etc/php-7.2.d/20-tokenizer.ini, /etc/php-7.2.d/20-xml.ini, /etc/php-7.2.d/20-xmlwriter.ini, /etc/php-7.2.d/20-xsl.ini, /etc/php-7.2.d/20-zip.ini, /etc/php-7.2.d/30-mcrypt.ini, /etc/php-7.2.d/30-mysql.ini, /etc/php-7.2.d/30-pdo_mysql.ini, /etc/php-7.2.d/30-pdo_odbc.ini, /etc/php-7.2.d/30-pdo_pgsql.ini, /etc/php-7.2.d/30-pdo_sqlite.ini, /etc/php-7.2.d/30-wddx.ini, /etc/php-7.2.d/30-xmlreader.ini, /etc/php-7.2.d/30-xmlrpc.ini, /etc/php-7.2.d/40-apcu.ini, /etc/php-7.2.d/40-igbinary.ini, /etc/php-7.2.d/40-imagick.ini, /etc/php-7.2.d/40-memcache.ini, /etc/php-7.2.d/40-oauth.ini, /etc/php-7.2.d/40-ssh2.ini, /etc/php-7.2.d/40-uuid.ini, /etc/php-7.2.d/50-memcached.ini, /etc/php-7.2.d/aws.ini, /etc/php-7.2.d/environment.ini, /etc/php-7.2.d/php.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar, zip, ssh2.shell, ssh2.exec, ssh2.tunnel, ssh2.scp, ssh2.sftp
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, bzip2.*, *

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			
The installer packages above will provide all of these (except PuTTYtel), but you can download them one by one by . (Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry .)			
putty.exe (the SSH and Telnet client itself)			
32-bit:	putty.exe	(or by FTP)	(signature)
64-bit:	putty.exe	(or by FTP)	(signature)
pscp.exe (an SCP client, i.e. command-line secure file copy)			
32-bit:	pscp.exe	(or by FTP)	(signature)
64-bit:	pscp.exe	(or by FTP)	(signature)
psftp.exe (an SFTP client, i.e. general file transfer sessions much like FTP)			
32-bit:	psftp.exe	(or by FTP)	(signature)
64-bit:	psftp.exe	(or by FTP)	(signature)
puttytel.exe (a Telnet-only client)			
32-bit:	puttytel.exe	(or by FTP)	(signature)
64-bit:	puttytel.exe	(or by FTP)	(signature)
plink.exe (a command-line interface to the PuTTY back ends)			
32-bit:	plink.exe	(or by FTP)	(signature)
64-bit:	plink.exe	(or by FTP)	(signature)
pageant.exe (an SSH authentication agent for PuTTY, PSCP, PSFTP, and Plink)			
32-bit:	pageant.exe	(or by FTP)	(signature)
64-bit:	pageant.exe	(or by FTP)	(signature)
puttygen.exe (a RSA and DSA key generation utility)			
32-bit:	puttygen.exe	(or by FTP)	(signature)
64-bit:	puttygen.exe	(or by FTP)	(signature)
putty.zip (a .ZIP archive of all the above)			
32-bit:	putty.zip	(or by FTP)	(signature)
64-bit:	putty.zip	(or by FTP)	(signature)

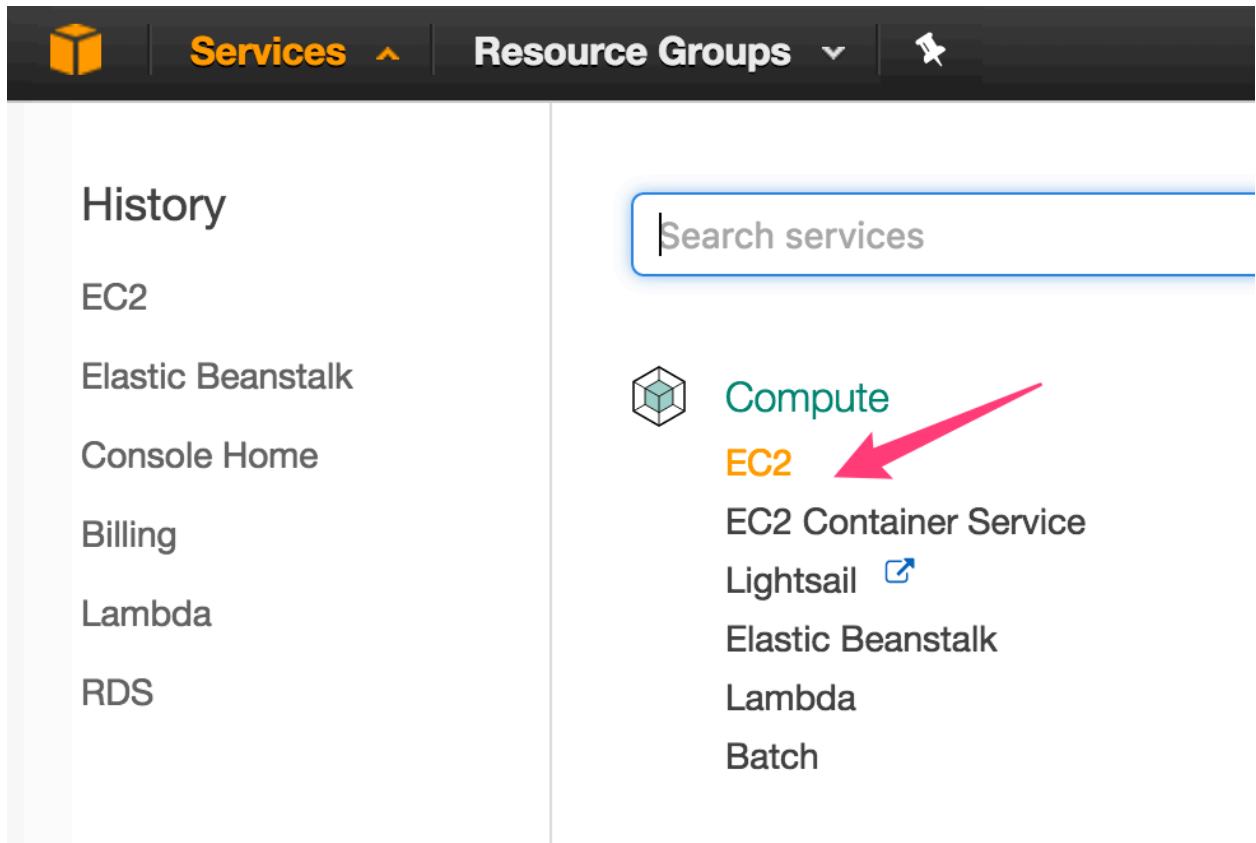
PuTTY needs additional setup as it needs to use a converted version of the private key. The instructions on how to perform such conversion are available here:

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>

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: [Refresh](#) 
Enables remote login to your instances.

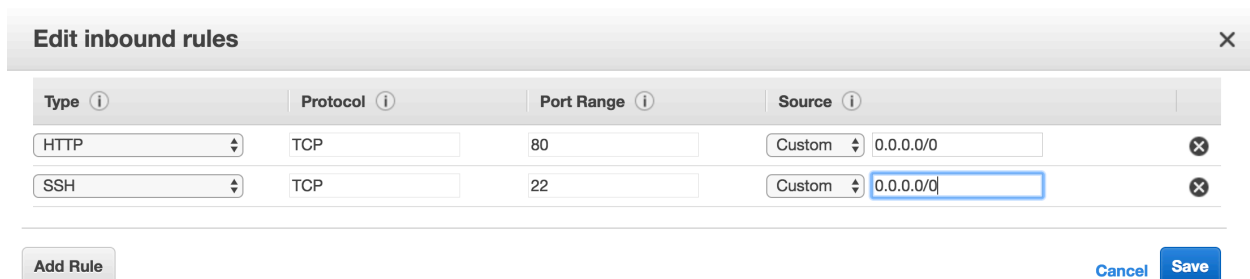
- Hit **Apply** and then **Save** and wait for a few minutes for the configuration changes to take place. You may get INFO, WARN and sometimes SEVERE messages during this time. Wait until the update of the environment has completed, and **Health** is back to **Ok**.
- Go back to your EC2 instance after some time and check under **Key Name**, you should now see your associated key pair.



8.3 Open port 22

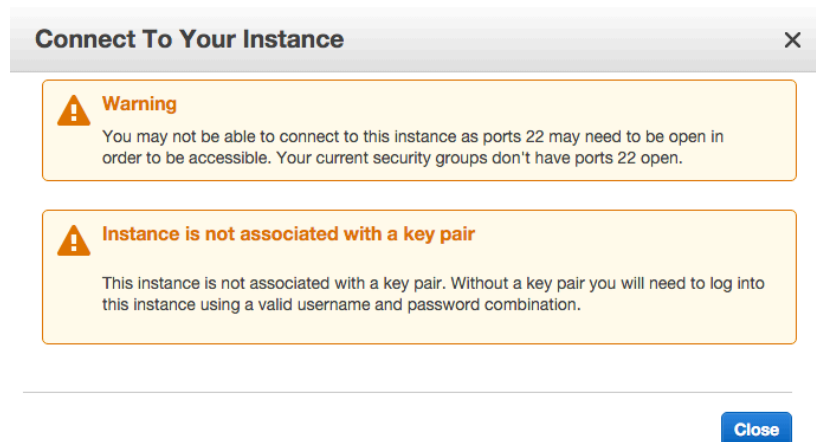
To open port 22, which is needed by SSH, follow these steps:

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**.



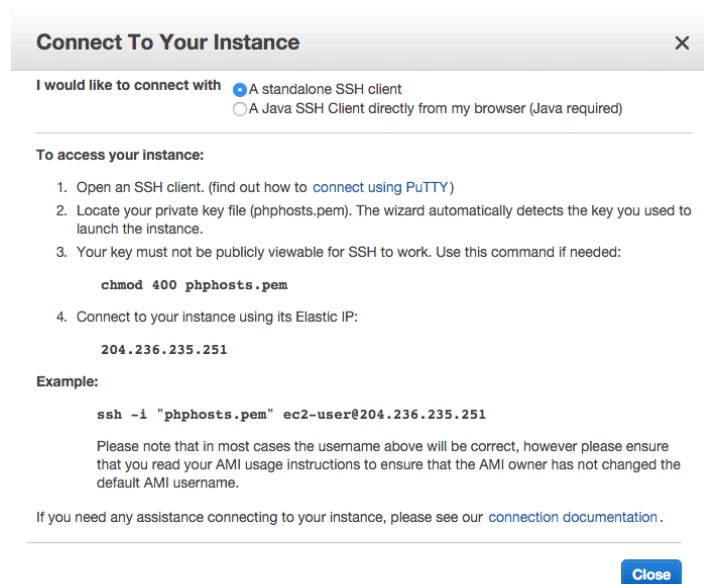
8.3.1 Errors when Connecting

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:

```
chmod 400 phphosts.pem
```

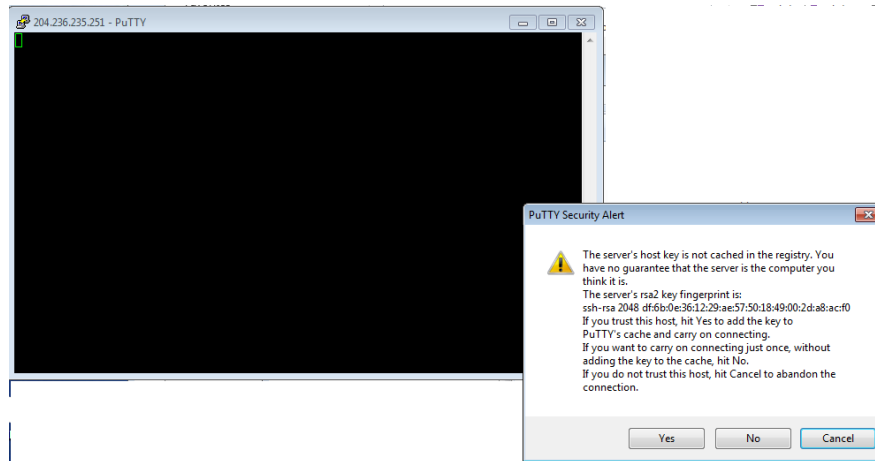

[illegible]

You can find more info here:

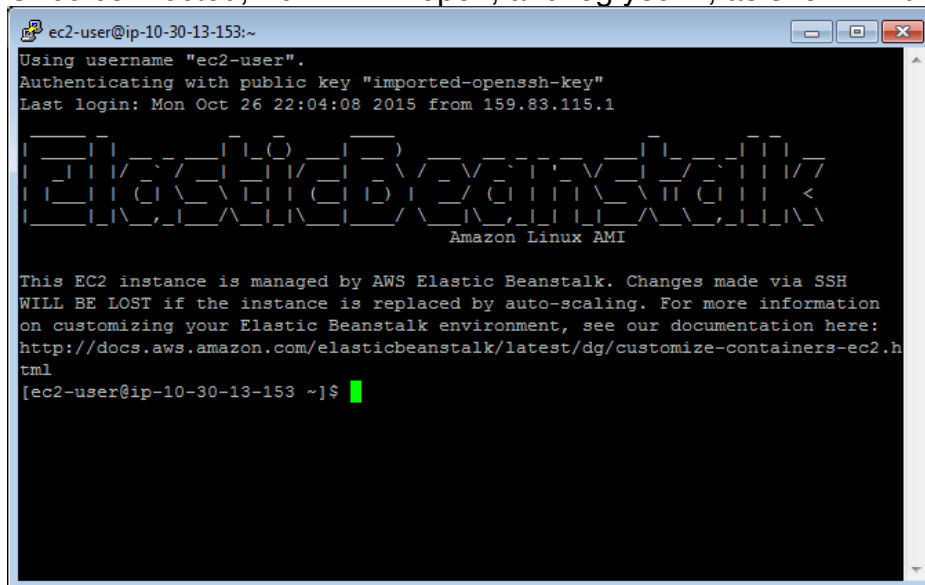
8.4.2 PC running Windows / PuTTY

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>

14



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/xvda1      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!!