

ECV-Compute-201

Launch an EC2 in LAMP

Environment

2018.02.21

Version 2.1

Agenda

About this lab	3
Scenario	3
AWS EC2 introduction	3
Prerequisites.....	3
Lab tutorial	3
1.1 Create Your VPC	3
1.2 Launch an instance	5
Connect to your instance (Windows)	6
Installing and start the LAMP server on EC2	9
Set file permissions of Apache httpd serves	10
Conclusion	12

About this lab

Scenario

The following procedures help you install an Apache web server with PHP and MySQL support on your Amazon Linux instance (sometimes called a LAMP web server or LAMP stack). You can use this server to host a static website or deploy a dynamic PHP application that reads and writes information to a database.

This lab introduces you to Amazon Elastic Compute Cloud (Amazon EC2) using the AWS Management Console.



AWS EC2 introduction

What is Amazon EC2?

Amazon EC2 presents a true virtual computing environment, allowing you to use web service interfaces to launch instances with a variety of operating systems, load them with your custom application environment, manage your network's access permissions, and run your image using as many or few systems as you desire.

The workshop's region will be in 'N.Virginia'

Prerequisites

-  Download Putty: IF you don't already have the PuTTY client installed on your machine, you can download and then launch it from here:
<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>
-  Download the **PuTTYgen** from [PuTTYgen download page](#).

Lab tutorial

1.1 Create Your VPC

- 1.1. In the **AWS Management Console**, on the **service** menu, click **VPC**.

- 1.2. In the navigation pane, click **Your VPCs**.
- 1.3. Click **Create VPC**, enter the following details:
 - **Name tag: My Lab VPC**
 - **IPv4 CIDR block: 10.0.0.0/16**
- 1.4. Click **Yes, Create**.
- 1.5. In the navigation pane, click **Internet Gateways**.
- 1.6. Click **Create Internet Gateway**, enter **Name tag: My Lab IGW**.
- 1.7. Choose **My Lab IGW**, click **Attach to VPC**, and then choose **My Lab VPC** you created, click **Yes, Attach**.
- 1.8. In the navigation pane, click **Subnets**.
- 1.9. Click **Create Subnet**, enter the following details:
 - **Name tag: Public Subnet 1**
 - **VPC: My Lab VPC**
 - **Availability Zone: us-east-1a**
 - **IPv4 CIDR block: 10.0.1.0/24**
- 1.10. Click **Yes, Create**.
- 1.11. Click **Create Subnet**, enter the following details:
 - **Name tag: Public Subnet 2**
 - **VPC: My Lab VPC**
 - **Availability Zone: us-east-1b**
 - **IPv4 CIDR block: 10.0.2.0/24**
- 1.12. Click **Yes, Create**.
- 1.13. In the navigation pane, click **Route Tables**.
- 1.14. Click **Create Route Table**, enter the following details:
 - **Name tag: Public Route Table**
 - **VPC: select My Lab VPC**
- 1.15. Click **Yes, Create**.
- 1.16. Select the **Public Route Table** you just created, and then click **Routes** tab in the lower pane of the console, click **Edit**, click **Add another route**, enter the following details:
 - **Destination: 0.0.0.0/0**

- **Target:** choose **igw-xxxxxxx** that you created earlier
- 1.17. Click **Save**.
- 1.18. Select the **Public Route Table** you just created, and then click **Subnet Associations** tab in the lower pane of the console, click **Edit**, choose **Public Subnet 1** and **Public Subnet 2** you just created to associate.
- 1.19. Click **Save**.

1.2 Launch an instance

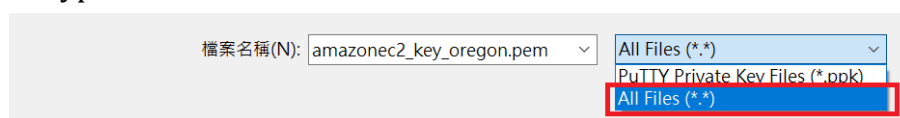
- 1.20. On the **service** menu, click **EC2**.
- 1.21. Click **Launch Instance**.
- 1.22. In the navigation pane, choose **Quick Start**, in the row for the first **Amazon Linux AMI**, click **Select**.
- 1.23. On **Step2: Choose a Instance Type** page, make sure **t2.micro** is selected and click **Next: Configure Instance Details**.
- 1.24. On **Step3: Configure Instance Details** page, enter the following and leave all other values with their default:
 - **Network: My Lab VPC**
 - **Subnet: Public Subnet 1**
 - Auto-assign Public IP: click **Enable**
- 1.25. Click **Next: Add Storage**, leave all values with their default.
- 1.26. Click **Next: Tag Instance**.
- 1.27. On **Step5: Tag Instance** page, enter the following information:
 - **Key: Name**
 - **Value: Web Server**
- 1.28. Click **Next: Configure Security Group**.
- 1.29. On **Setp6: Configure Security Group** page, click **create a new security group**, enter the following information:
 - **Security group name: WebServerSG**
 - **Description: Web Server Security Group**
- 1.30. Click **Add Rule**.
- 1.31. For Type, click **SSH (22), and HTTP (80)**.

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0, ::0	e.g. SSH for Admin Desktop

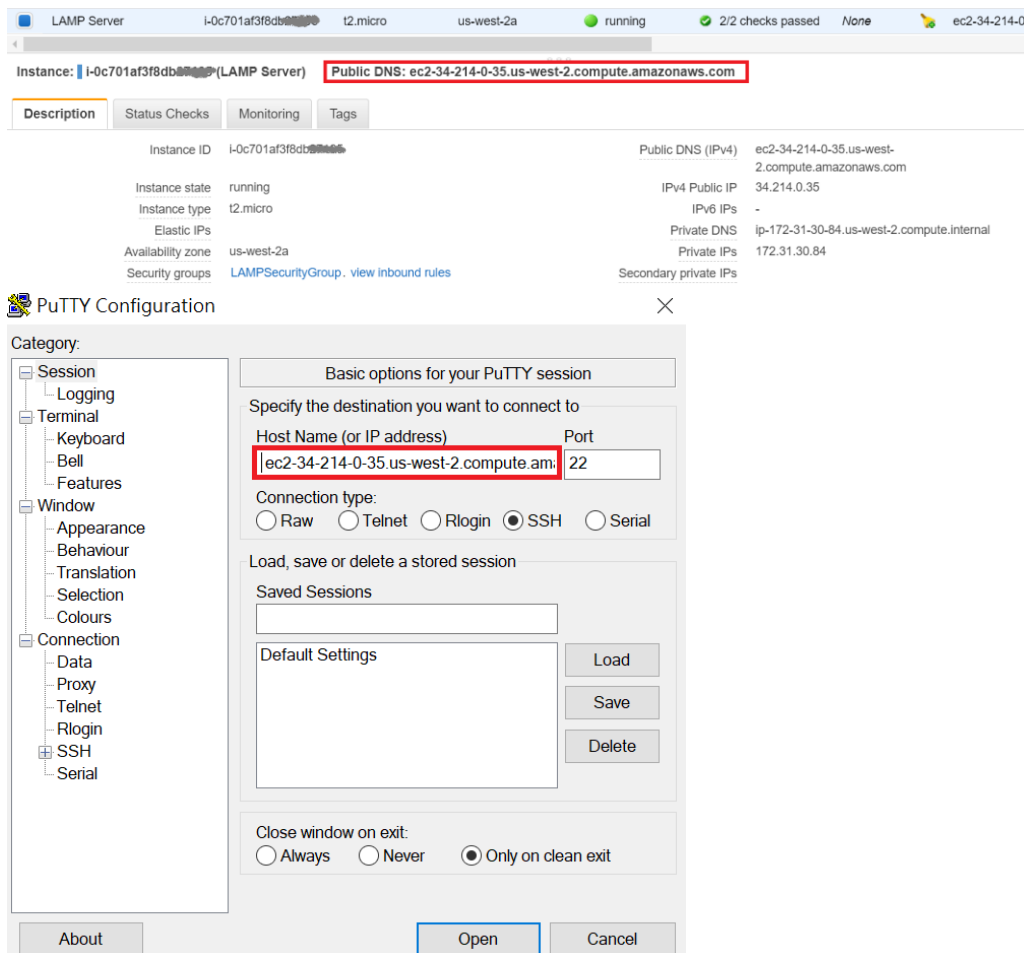
- 1.32. Click **Review and Launch**.
- 1.33. Review the instance information and click **Launch**.
- 1.34. Click **Create a new key pair**, enter the **Key pair name** (ex. **amazonec2_keypair_oregon**), click **Download Key Pair**.
- 1.35. Click **Launch Instances**.
- 1.36. Scroll down and click **View Instances**.
- 1.37. Wait for **Web Server** shows 2/2 checks passed in the **Status Checks** column.
This will take 3-5 minutes. Use the refresh icon at the top right to check for updates.
- 1.38. On the Description tab in the lower pane, note the **Public IP** for the instance.

1.3 Connect to your instance (Windows only)

- 1.39. Start PuTTYgen.exe, click **Load**. By default, PuTTYgen display only files with the extension *.ppk*. to locate your *.pem* file, select the option to display files of all types.



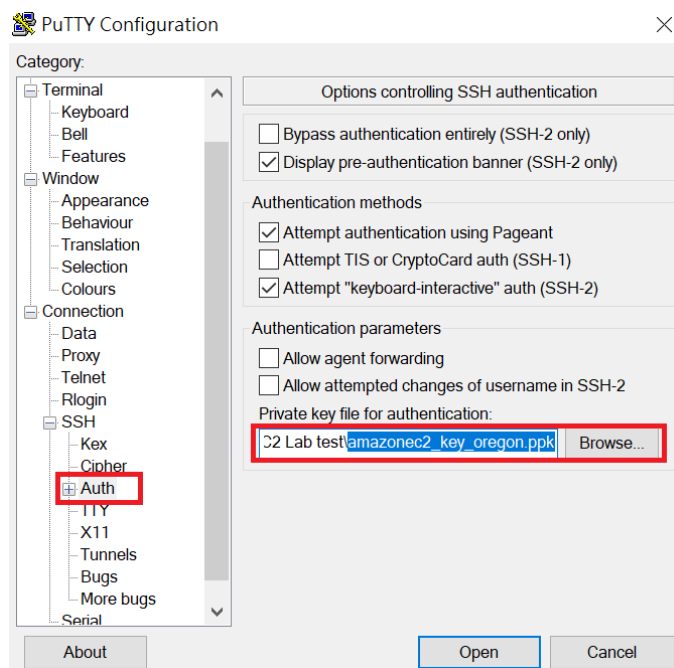
- 1.40. Select your *.pem* file (ex. **amazonec2_keypair_oregon.pem**), and then click **Open**. Click **OK** to dismiss the confirmation dialog box.
- 1.41. Click **Save private key** to save the key in the format that PuTTY can use.
PuTTYgen displays a warning about saving the key without a passphrase, click **Yes**.
- 1.42. Specify the same name for the key that you used for the key pair (ex. **amazonec2_keypair_oregon.ppk**). PuTTY automatically adds the *.ppk* extension.
- 1.43. Launch **PuTTY** by running the **PuTTY.exe** file you downloaded.
- 1.44. For **Host Name**, enter the Public IP address from your Web Server instance which you copied into a text editor in the lab.



The screenshot displays the AWS Management Console for an EC2 instance named "LAMP Server". The instance is in the "running" state, using the "t2.micro" instance type in the "us-west-2a" availability zone. The public DNS is highlighted as "ec2-34-214-0-35.us-west-2.compute.amazonaws.com". Below this, the "PuTTY Configuration" dialog is open, showing the "SSH" connection type selected. The "Host Name (or IP address)" field is set to "ec2-34-214-0-35.us-west-2.compute.am" and the "Port" is "22". The "Close window on exit" option is set to "Only on clean exit".

Instance ID	Instance state	Instance type	Elastic IPs	Availability zone	Security groups	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Private DNS	Private IPs	Secondary private IPs
i-0c701af3f8db...	running	t2.micro		us-west-2a	LAMPSecurityGroup	ec2-34-214-0-35.us-west-2.compute.amazonaws.com	34.214.0.35	-	ip-172-31-30-84.us-west-2.compute.internal	172.31.30.84	

1.45. In the Connection list, click **Connect>SSH>Auth**, click **Browse** to choose your key pair (ex. **amazonec2_keypair_oregon.ppk**), click **Open**.



- 1.46. Enter **ec2-user**, and press ENTER. You are now logged in to your **Web Server** instance.

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Thu Sep  7 03:49:53 2017 from 36-232-143-145.dynamic-ip.hinet.net

 _ _ | _ _ | _
|_| ( _ _ | _ |
 _ _ \ _ _ | _ |

Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2017.03-release-notes/
8 package(s) needed for security, out of 8 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-30-84 ~]$
```

1.4 Connect to your instance (Linux/OSX only)

Note: This section is for Linux and Max OSX users only. If you are running Windows but have not yet connected to your instance, go back to 1.3. If you have already connected to your instance, skip ahead to next step.

- 1.47. To connect to your EC2 instance, run the following commands in Terminal:

```
chmod 400 <path and name of pem>
ssh -i <path and name of pem> ec2-user@<public IP>
```

For **<path and name of pem>**, substitute the path/filename to the .pem file you downloaded.

For **<public IP>**, substitute the public IP address for your Web Server instance which you copied into a text editor earlier in the lab.

2.1 Installing and start the Web server on EC2

2.1. Update all your software package, enter:

```
[ec2-user ~]$ sudo yum update -y
```

2.2. Use yum install command to install multiple software packages and all related dependencies at the same time.

```
[ec2-user ~]$ sudo yum install -y httpd24 php70  
mysql56-server php70-mysqlnd
```

2.3. Start Apache web server.

```
[ec2-user ~]$ sudo service httpd start  
Starting httpd: [ OK ]
```

2.4. Use the **chkconfig** command to configure the Apache web server to start at each system boot.

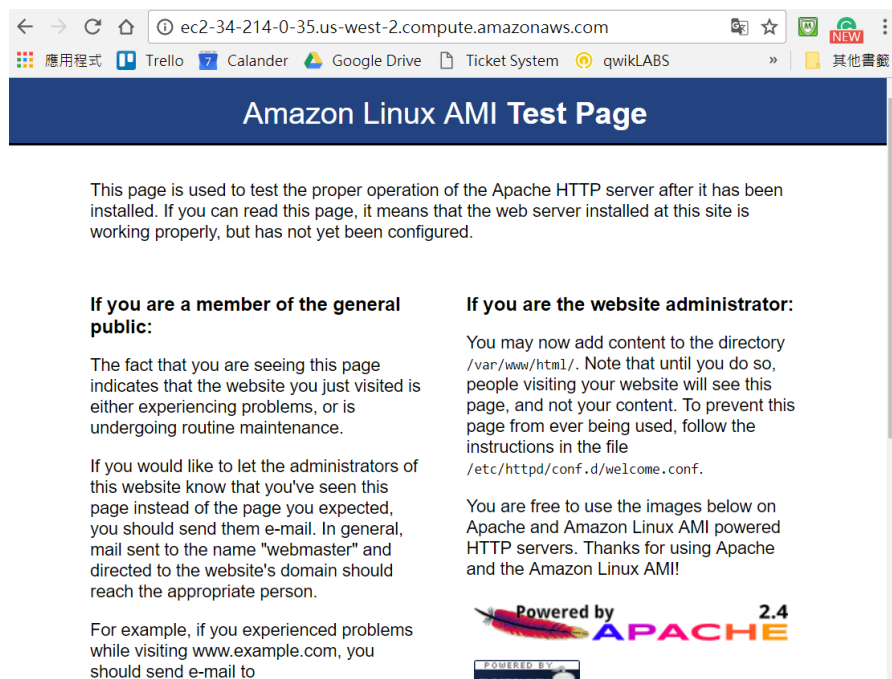
```
[ec2-user ~]$ sudo chkconfig httpd on
```

The **chkconfig** command does not provide any confirmation message when you successfully use it to enable a service. You can verify that **httpd** does on by running the following command:

```
[ec2-user ~]$ chkconfig --list httpd
```

```
ec2-user@ip-172-31-30-84:~  
mysql56-errmsg.x86_64 0:5.6.37-1.26.amzn1  
mysql56-libs.x86_64 0:5.6.37-1.26.amzn1  
perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.11.amzn1  
perl-Compress-Raw-Zlib.x86_64 1:2.061-4.1.amzn1  
perl-DBD-MySQL56.x86_64 0:4.023-5.21.amzn1  
perl-DBI.x86_64 0:1.627-4.8.amzn1  
perl-Data-Dumper.x86_64 0:2.145-3.5.amzn1  
perl-IO-Compress.noarch 0:2.061-2.12.amzn1  
perl-Net-Daemon.noarch 0:0.48-5.5.amzn1  
perl-PlRPC.noarch 0:0.2020-14.7.amzn1  
php70-cli.x86_64 0:7.0.21-1.23.amzn1  
php70-common.x86_64 0:7.0.21-1.23.amzn1  
php70-json.x86_64 0:7.0.21-1.23.amzn1  
php70-pdo.x86_64 0:7.0.21-1.23.amzn1  
php70-process.x86_64 0:7.0.21-1.23.amzn1  
php70-xml.x86_64 0:7.0.21-1.23.amzn1  
  
Complete!  
[ec2-user@ip-172-31-30-84 ~]$ sudo service httpd start  
Starting httpd: [ OK ]  
[ec2-user@ip-172-31-30-84 ~]$ sudo chkconfig httpd on  
[ec2-user@ip-172-31-30-84 ~]$ chkconfig --list httpd  
httpd 0:off 1:off 2:on 3:on 4:on 5:on 6:off  
[ec2-user@ip-172-31-30-84 ~]$
```

2.5. Test your web server. In a web browser, enter Public DNS address (or the public IP address) of your EC2; you should see the Apache test page.



Set file permissions of Apache httpd serves

2.6. Add your user (ex. **ec2-user**) to the **apache** group

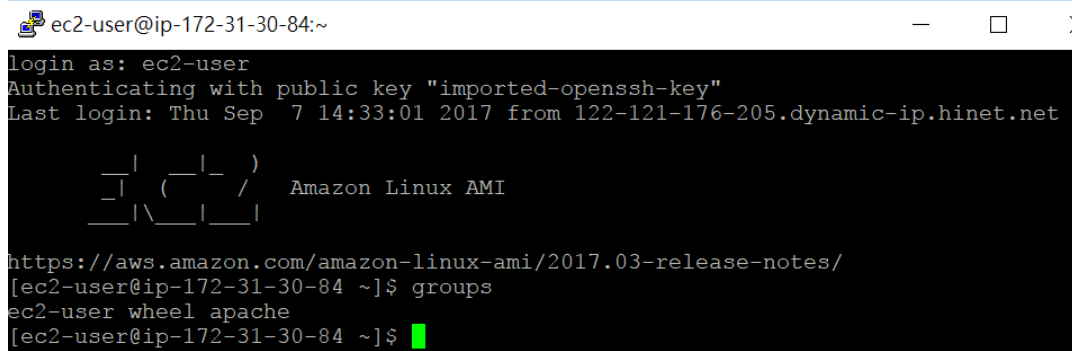
```
[ec2-user ~]$ sudo usermod -a -G apache ec2-user
```

2.7. Log out and log back in again, use **exit** command.

```
[ec2-user ~]$ exit
```

- 2.8. Reconnect to your instance, and then run the following command to verify your membership in the **apache** group.

```
[ec2-user ~]$ groups  
ec2-user wheel apache
```



A terminal window titled 'ec2-user@ip-172-31-30-84:~' showing the login process. It displays the login as 'ec2-user', authentication with a public key, and the last login time. Below this is the Amazon Linux AMI logo and a link to the release notes. The command 'groups' is run, showing the user is part of the 'ec2-user', 'wheel', and 'apache' groups.

- 2.9. Change the group ownership of **/var/www** and its contents to the **apache** group.

```
[ec2-user ~]$ sudo chown -R ec2-user:apache /var/www
```

- 2.10. Change the directory permissions of **/var/www** and its subdirectories to add group write permissions and to set the group ID on future subdirectories.

```
[ec2-user ~]$ sudo chmod 2775 /var/www  
[ec2-user ~]$ find /var/www -type d -exec sudo chmod  
2775 {} \;
```

- 2.11. Recursively change the file permissions of **/var/www** and its subdirectories to add group write permissions.

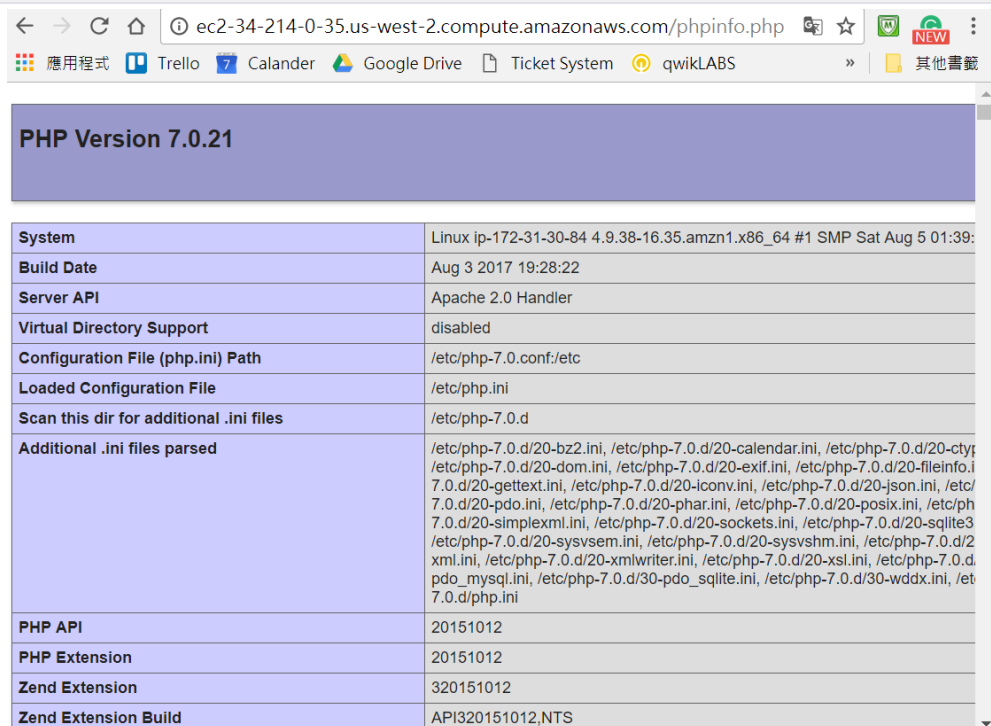
```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod  
0664 {} \;
```

- 2.12. Add a simple PHP file in the Apache document root to test LAMP web server.

```
[ec2-user ~]$ echo "<?php phpinfo(); ?>" >  
/var/www/html/phpinfo.php
```

2.13. In a web browser, enter the URL of the file you just created. This URL is the public DNS address of your EC2, for example:

`http://my.public.dns.amazonaws.com/phpinfo.php`



PHP Version 7.0.21	
System	Linux ip-172-31-30-84 4.9.38-16.35.amzn1.x86_64 #1 SMP Sat Aug 5 01:39:
Build Date	Aug 3 2017 19:28:22
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php-7.0.conf/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php-7.0.d
Additional .ini files parsed	/etc/php-7.0.d/20-bz2.ini, /etc/php-7.0.d/20-calendar.ini, /etc/php-7.0.d/20-ctype, /etc/php-7.0.d/20-dom.ini, /etc/php-7.0.d/20-exif.ini, /etc/php-7.0.d/20-fileinfo.i, /etc/php-7.0.d/20-gettext.ini, /etc/php-7.0.d/20-iconv.ini, /etc/php-7.0.d/20-json.ini, /etc/7.0.d/20-pdo.ini, /etc/php-7.0.d/20-phar.ini, /etc/php-7.0.d/20-posix.ini, /etc/ph7.0.d/20-simplexml.ini, /etc/php-7.0.d/20-sockets.ini, /etc/php-7.0.d/20-sqlite3, /etc/php-7.0.d/20-sysvsem.ini, /etc/php-7.0.d/20-sysvshm.ini, /etc/php-7.0.d/2xml.ini, /etc/php-7.0.d/20-xmlwriter.ini, /etc/php-7.0.d/20-xsl.ini, /etc/php-7.0.d/pdo_mysql.ini, /etc/php-7.0.d/30-pdo_sqlite.ini, /etc/php-7.0.d/30-wddx.ini, /et7.0.d/php.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS

2.14. You can review the item which just type-in via console

Conclusion

Congratulations! You now have learned how to:

- Logged into Amazon Management Console
- Create an Amazon Linux Instance from and Amazon Machine Image (AMI).
- Find your instance in the Amazon Management Console
- Logged into your instance