

# **ECV-STR-212**

## **Working with EC2/RDS/CW/SNS**

### **2017.12.17**

### **Version 2.1**

## Agenda

<b>About this lab .....</b>	<b>3</b>
Scenario.....	3
Amazon EC2 introduction .....	4
Amazon RDS introduction .....	4
Amazon CloudWatch introduction.....	4
Amazon Simple Notification Service introduction.....	5
Prerequisites .....	5
<b>Lab tutorial .....</b>	<b>6</b>
Create Your VPC.....	6
Launch an instance .....	7
Connect to your linux instance (Windows).....	8
Installing and start the LAMP server on EC2.....	11
Create a VPC security group for the RDS DB Instance.....	13
Create Private Subnets for Your Amazon RDS Instances.....	14
Create DB Subnet Group .....	14
To launch a MySQL DB instance.....	15
Use EC2 to connect with database.....	17
Create an alarm in CloudWatch and SNS service.....	18
<b>Conclusion .....</b>	<b>21</b>

## About this lab

### Scenario

The following procedures help you install an Apache web server with PHP and MySQL support on your Amazon Linux instance. 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.

Please referred to Figure 1 which is architecture of this lab.

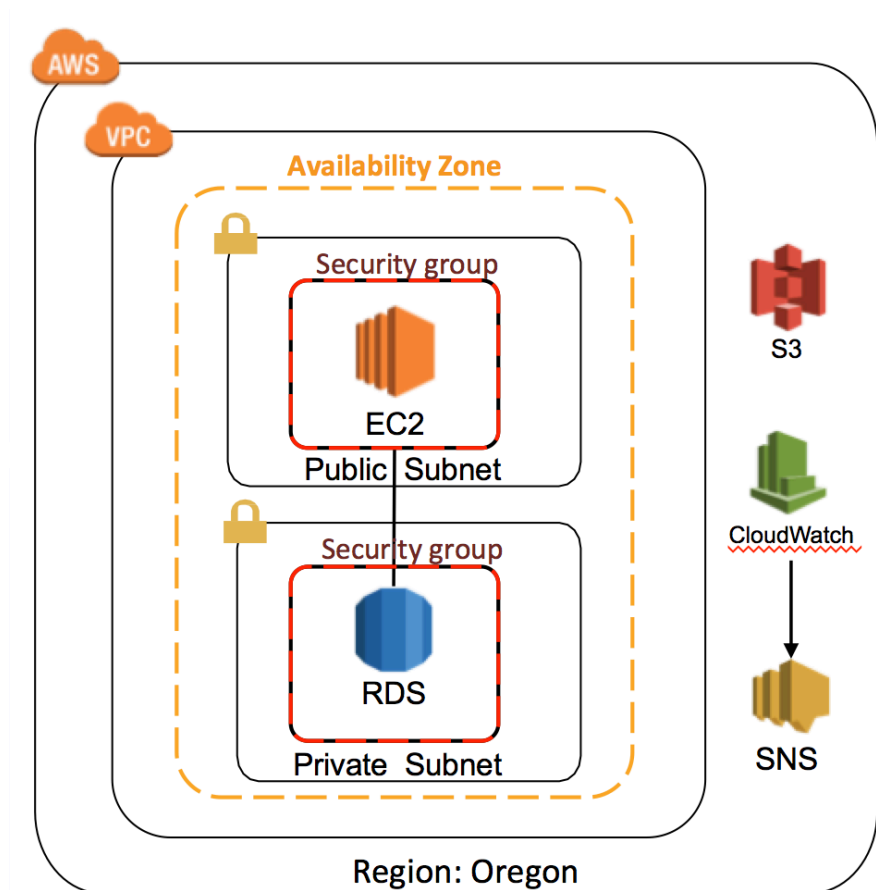


Figure 1: Architecture

## Amazon 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.

## Amazon RDS introduction

What is Amazon RDS?

Amazon Relation Database Service (Amazon RDS) makes it easy to setup, operate, and scale a relation database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

Amazon RDS is available on several database instance type-optimized for memory, performance or I/O-and provides you with six familiar database engines to choose from, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, and Microsoft SQL Server.

## Amazon CloudWatch introduction

What is Amazon CloudWatch?

Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources. Amazon CloudWatch can monitor AWS resources such as

Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate. You can use Amazon CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health.

## Amazon Simple Notification Service introduction

What is Amazon SNS?

Amazon Simple Notification Service (SNS) is a flexible, fully managed pub/sub messaging and mobile notifications service for coordinating the delivery of messages to subscribing endpoints and clients. With SNS you can fan-out messages to a large number of subscribers, including distributed systems and services, and mobile devices.

## The workshop's region will be in 'Oregon'

### Prerequisites

- Download Putty: IF you don't already have the **PuTTY client/PuTTYgen** installed on your machine, you can download and then launch it from here:  
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

## Lab tutorial

### 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:
  - i. Name tag: My Lab VPC
  - ii. 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. Click **Yes, Create**.
- 1.8. Choose **My Lab IGW**, click **Attach to VPC**, and then choose **My Lab VPC** you created, click **Yes, Attach**.
- 1.9. In the navigation pane, click **Subnets**.
- 1.10. Click **Create Subnet**, enter the following details:
  - i. Name tag: Public Subnet 1
  - ii. VPC: My Lab VPC
  - iii. IPv4 CIDR block: 10.0.1.0/24
  - iv. Availability Zone : 'us-west-2b'
- 1.11. Click **Yes, Create**.
- 1.12. Click **Create Subnet**, enter the following details:
  - i. Name tag: Public Subnet 2
  - ii. VPC: My Lab VPC
  - iii. IPv4 CIDR block: 10.0.2.0/24
  - iv. Availability Zone : 'us-west-2c'
- 1.13. Click **Yes, Create**.

- 1.14. Select **Public Subnet 1**, click **Route Table** in the lower pane, and then scroll down and click hyperlink of **Route Table** (ex.rtb-4e18d537) to Route Table pages, choose Route Table, in the **Route** tab, click **Edit**, click **Add another route**, enter **Destination: 0.0.0.0/0**, **Target: igw-xxxxxxx**. Click **Save**.
- 1.15. Check the route table for **Public Subnet 2**.

## Launch an instance

- 1.16. In the **AWS Management Console**, on the **service** menu, click **EC2**.
- 1.17. Click **Launch Instance**.
- 1.18. In the navigation pane, choose **Quick Start**, in the row for **Amazon Linux AMI**, click **Select**.

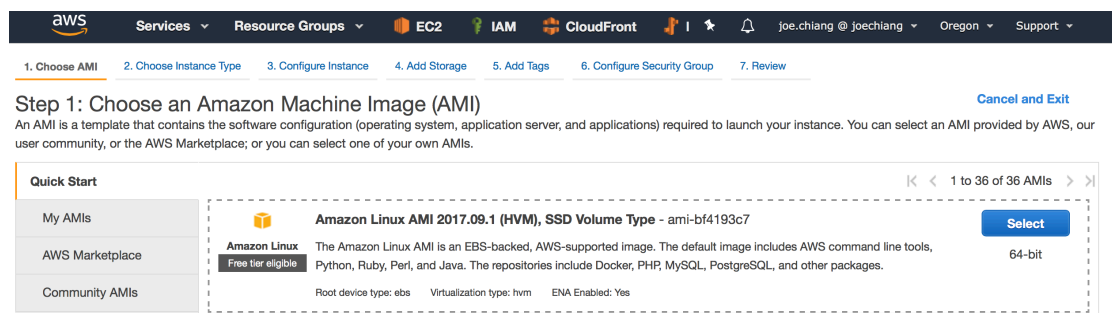


Figure 2: Amazon Linux AMI

- 1.19. On **Step2: Choose a Instance Type** page, make sure **t2.micro** is selected and click **Next: Configure Instance Details**.
- 1.20. On **Step3: Configure Instance Details** page, enter the following and leave all other values with their default:
  - i. Network: My Lab VPC
  - ii. Subnet: Public Subnet 1
  - iii. Auto-assign Public IP: click Enable
- 1.21. Click **Next: Add Storage**, leave all values with their default.
- 1.22. Click **Next: Add Tag**.

- 1.23. On **Step5: Tag Instance** page, enter the following information:
  - i. Key: Name
  - ii. Value: LAMP Server
- 1.24. Click **Next: Configure Security Group**.
- 1.25. On **Setp6: Configure Security Group** page, click **create a new security group**, enter the following information:
  - i. Security group name: LAMPSecurityGroup
  - ii. Description: Enable SSH, HTTP and HTTPS access
- 1.26. Click **Add Rule**.
- 1.27. For Type, click **SSH (22), HTTP (80) and HTTPs (443)**.

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 ✕
HTTPS ▾	TCP	443	Custom ▾ 0.0.0.0/0, :::0	e.g. SSH for Admin Desktop ✕

Figure 3: Rule in Security Group

- 1.28. Click **Review and Launch**.
- 1.29. Review the instance information and click **Launch**.
- 1.30. Click **Create a new key pair**, enter the **Key pair name** (ex. **amazonec2\_keypair\_oregon**), click **Download Key Pair**.
- 1.31. Click **Launch Instances**.
- 1.32. Scroll down and click **View Instances**.
- 1.33. Wait until **Lab 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.

## Connect to your linux instance (Windows)

- 1.34. 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.

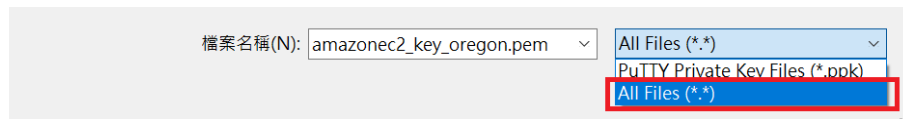


Figure 4: Puttygen

- 1.35. Select your **.pem** file (ex. **amazonec2\_keypair\_oregon.pem**), and then click **Open**. Click **OK** to dismiss the confirmation dialog box.
- 1.36. 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**.
- 1.37. 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.
- 1.38. Start **PuTTY.exe**, enter **Host Name**, find host name from AWS console. (select LAMP Server, and copy the **public IP** value.)

ELB-Demo-white	i-06af74aeb2c7d0a7b	t2.micro	us-west-2a	pending	Initializing	None	ec2-35-160-240-95.us-...	35.160.240.95
Internal Server	i-07e657c942d961d87	t2.micro	us-west-2a	stopped		None		-
KS-Demo	i-07ea20ae892d41d...	t2.micro	us-west-2a	stopped		None		-
Beanstalk-Demo	i-0a8714884cb0aadb5	t2.micro	us-west-2a	stopped		None		-
VPCFlowLogs-DemoPage	i-b46e2a70	t2.micro	us-west-2b	stopped		No Data		-

Instance: i-06af74aeb2c7d0a7b (ELB-Demo-white)		Public DNS: ec2-35-160-240-95.us-west-2.compute.amazonaws.com	
Description	Status Checks	Monitoring	Tags
Instance ID	i-06af74aeb2c7d0a7b		
Instance state	pending		
Instance type	t2.micro		
Elastic IPs			
		Public DNS (IPv4)	ec2-35-160-240-95.us-west-2.compute.amazonaws.com
		IPv4 Public IP	35.160.240.95
		IPv6 IPs	-
		Private DNS	ip-172-31-36-126.us-west-2.compute.internal

Figure 5: EC2 Public IP

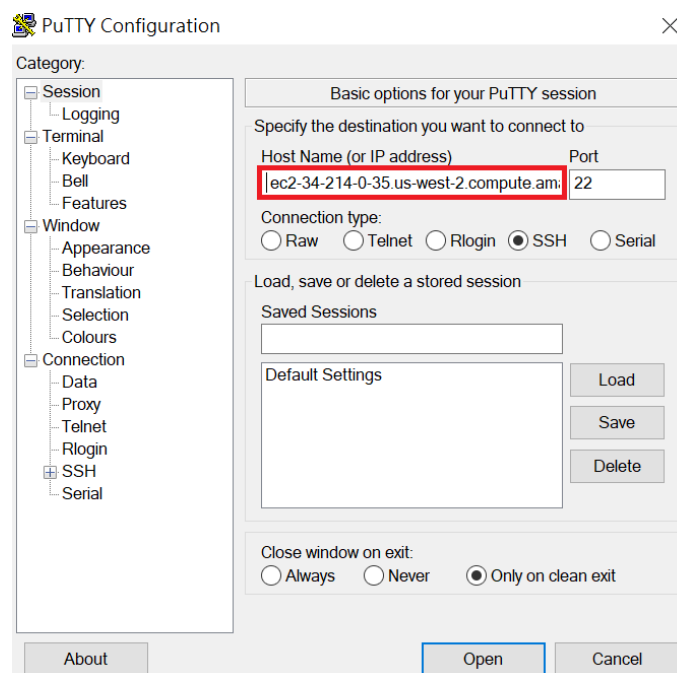


Figure 6 Paste EC2 public DNS into Putty

- 1.39. On the navigation pane, click **Connect>SSH>Auth**, click **Browse** to choose your key pair (ex. **amazonec2\_keypair\_oregon.ppk**), click **Open**.

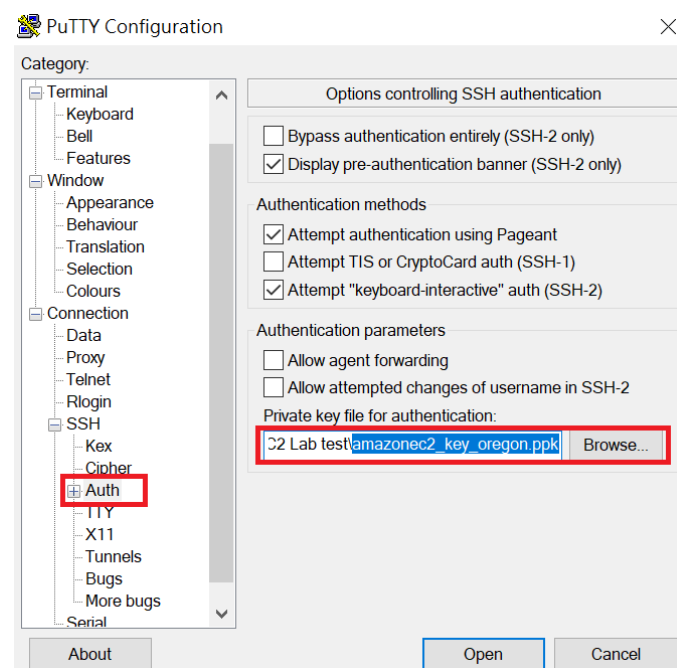


Figure 7: Interactive key pairs into putty

- 1.40. Enter **ec2-user**.

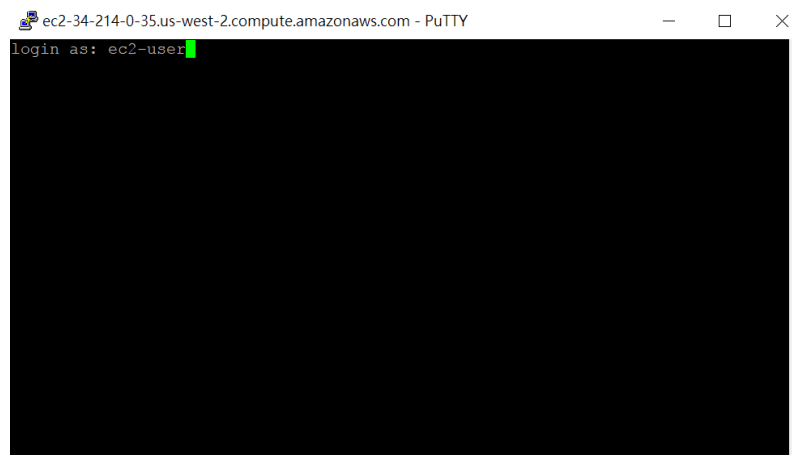


Figure 8: Connect with EC2

1.41. You are successfully connecting to EC2.

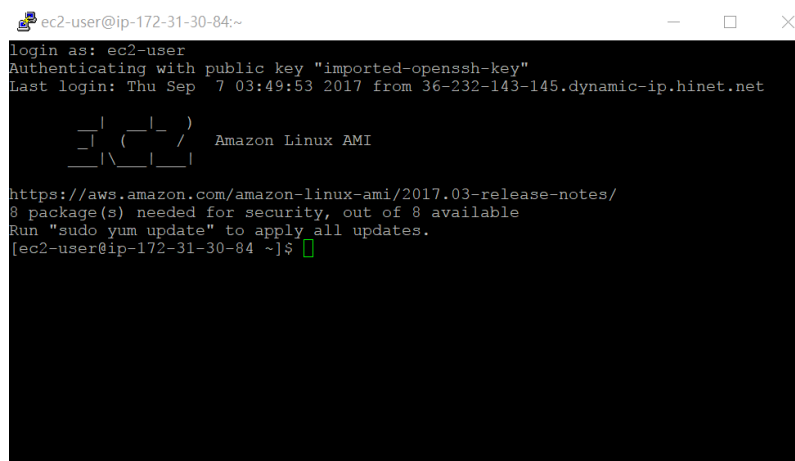


Figure 9: Successful interface

## Installing and start the LAMP 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-mysqld
```

2.3. Start Apache web server.

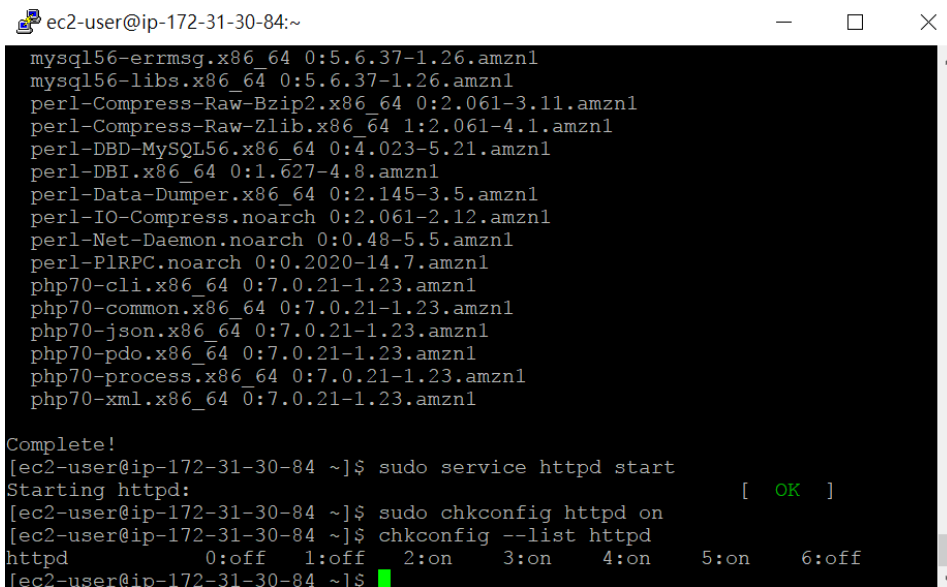
```
[ec2-user ~]$ sudo service httpd start  
  
Starting httpd: [ OK ]
```

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** dos on by running the following command:

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



```
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 ~]$
```

Figure 10: Verify https status

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.

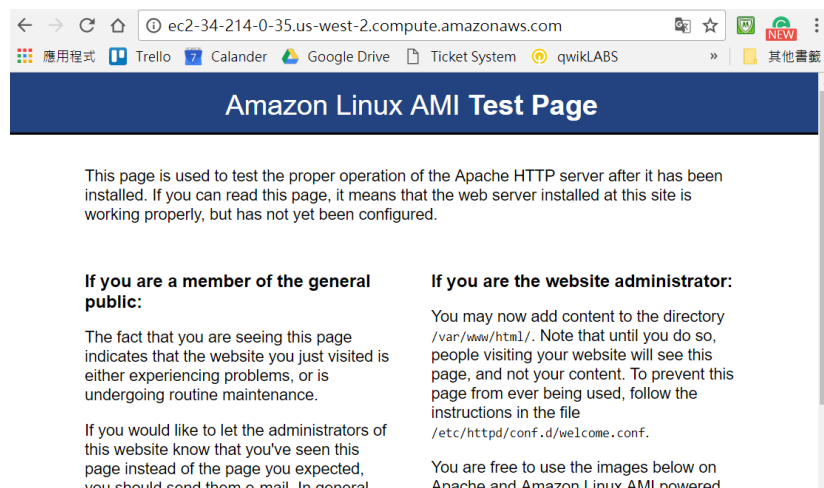


Figure 11 Install Successful Page

## Create a VPC security group for the RDS DB Instance

- 3.1. In the **AWS Management Console**, on the **service** menu, click **VPC**.
- 3.2. In the navigation, click Security Groups.
- 3.3. Click **Create Security Group**.
- 3.4. In the Create Security Group dialog box, enter the following details:
  - i. Name tag: DBSecurityGroup
  - ii. Group name: DBSecurityGroup
  - iii. Description: DB instance security group
  - iv. VPC: Click My Lab VPC
- 3.5. Click Yes, Create.
- 3.6. Select **DBSecurityGroup** you just created.
- 3.7. Click the **Inbound Rules** tab, and then click **Edit**.
- 3.8. Create an inbound rule with the following details:
  - i. Type: MySQL/Auora(3306)
  - ii. Protocol: TCP(6)
  - iii. Source: Paste Security Group ID of LAMPSecurityGroup

**\*\*Search LAMPSecurityGroup to find sg-xxxxxxx**

### 3.9. Click **Save**

## Create Private Subnets for Your Amazon RDS Instances

- 4.1. In the navigation pane, click **Subnets**.
- 4.2. Select **Public Subnet 1**, scroll down to the Summary tab in the lower pane. Take note of the **Availability Zone** for this subnet.
- 4.3. Select **Public Subnet 2**, scroll down to the Summary tab in the lower pane. Take note of the **Availability Zone** for this subnet.
- 4.4. Click Create Subnet dialog box, enter the following details:
  - i. Name tag : Private Subnet 3
  - ii. VPC : Select 'My Lab VPC'
  - iii. Availability Zone : 'us-west-2b'
  - iv. CIDR block : 10.0.5.0/24
- 4.5. Click **Yes, Create**.
- 4.6. Click **Create Subnet**.
- 4.7. In Create Subnet dialog box, enter the following details:
  - i. Name tag : Private Subnet 4
  - ii. VPC : Select 'My Lab VPC'
  - iii. Availability Zone : 'us-west-2c'
  - iv. CIDR block : 10.0.6.0/24
- 4.8. Click **Yes, Create**.

## Create DB Subnet Group

- 5.1. In the **AWS Management Console**, on the **service** menu, click **RDS**.
- 5.2. In the navigation pane, click **Subnet Groups**.
- 5.3. In the navigation pane, click **Create Subnet Groups**.

- 5.4. On the Create DB Subnet Group page, enter the following details:
  - i. Name: dbsubnetgroup
  - ii. Description: Lab DB Subnet Group
  - iii. VPC ID: Click My Lab VPC
- 5.5. For Add subnet, click Availability zone, choose us-west-2b, click subnet, choose **10.0.5.0/24**, then click **Add**.
- 5.1. Choose another **Availability Zone us-west-2c**, click the Availability Zone you selected for **Private Subnet 4**. For Subnet ID, click **10.0.6.0/24**, then click **Add**.
- 5.2. Click **Create**.
- 5.3. If you do not see your new subnet group, click the refresh icon in the upper-right corner of the console.

### To launch a MySQL DB instance

- 6.1. In the **AWS Management Console**, on the **service** menu, click **RDS**.
- 6.2. In the navigation pane, choose **Instances**.
- 6.3. Choose **Launch DB Instance** to start the **Launch DB Instance Wizard**. The wizard opens on the **Select Engine** page.

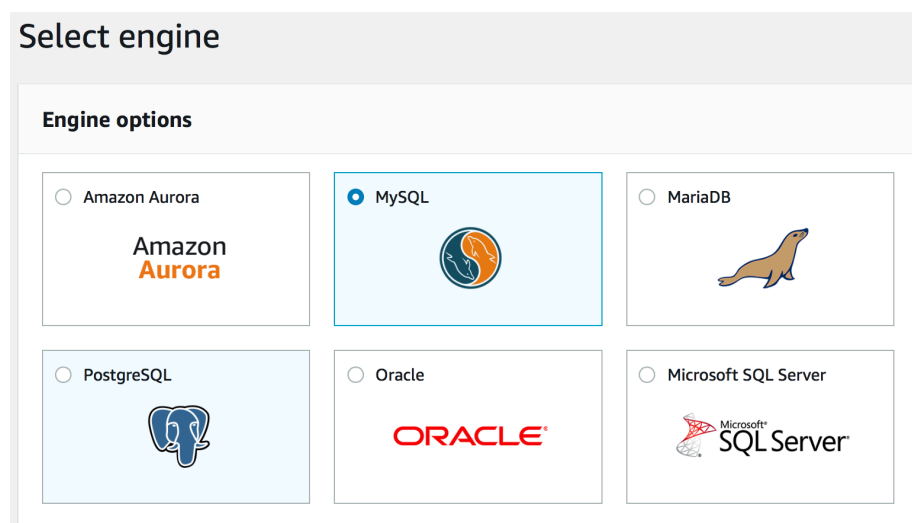


Figure 12: Launch DB Instance Wizard

- 6.4. In the **Select Engine** window, click the **Next** button for the MySQL DB engine.
- 6.5. Click **Use case** step. Click **Production - MySQL**. Click **Next**.
- 6.6. On the **Specify DB Details** page, enter the following details:
  - i. DB Instance Class: Choose db.t2.small—1 vCPU, 2GiB RAM.
  - ii. Multi-AZ Deployment: Click Yes.
  - iii. DB Instance Identifier: LabDBInstance
  - iv. Master Username: labuser
  - v. Master Password: labpassword.
  - vi. Confirm Password: labpassword
- 6.7. Click **Next**.
- 6.8. On the **Configure Advanced Settings** page, enter the following details and leave all other values with their default:
  - i. VPC: My Lab VPC
  - ii. Subnet Group: dbsubnetgroup
  - iii. Publicly Accessible: No
  - iv. VPC Security Group(s): Select existing VPC Security Group -> DBSecurityGroup
  - v. Database Name: sampledb
  - vi. Encryption: **Disable Encryption**.
  - vii. Backup: Backup retention period : **0 days**
  - viii. Monitoring: **Disable enhanced monitoring**
  - ix. Maintenance : **choose Disable auto minor version upgrade**
- 6.9. Click **Launch DB Instance**.
- 6.10. Click **View DB Instances details**.
- 6.11. Select **labdbinstance** and scroll down to **DETAILS** wait until **Endpoint** is available or modifying – this may take up to 10 minutes. Use the refresh icon in



the top right corner to check for updates.

Details <span>Modify</span>			
Configurations	Security and network	Instance and IOPS	Maintenance details
ARN arn:aws:rds:us-west-2:763064383464:db:labdbinstance	Availability zone us-west-2c  VPC <a href="#">My Lab VPC (vpc-045bac7d)</a>	Instance Class db.t2.small  Storage Type Provisioned IOPS (SSD)	Auto minor version upgrade <b>No</b>  Maintenance window sat:11:26-sat:11:56 UTC

Figure 13: RDS - Instance details screen

**6.12.** Copy and save the **Endpoint**, making sure to not copy the :3306 – your **Endpoint**. **Endpoint** should look similar to the following as below.:

example: db.choi5coyenv6.us-west-2.rds.amazonaws.com. You will change localhost to this endpoint later.

Username labuser	<a href="#">db.t2.small</a> ( active )	Multi AZ <b>Yes</b>	Encryption enabled <b>No</b>
Option Group default:mysql-5-6	Publicly accessible <b>No</b>	Secondary zone us-west-2b	
Parameter group default.mysql5.6 (in-sync)	Endpoint <a href="#">labdbinstance.ce1tnzyqpzqu.us-west-2.rds.amazonaws.com</a>	Automated backups <b>Disabled</b>	
Copy tags to snapshots <b>false</b>	Certificate authority rds-ca-2015 (Mar 5, 2020)	Latest restore time N/A	
Resource ID db-CLMU53POL2D7U76ML4DC2			

Figure 14: DB endpoint

## Use EC2 to connect with database

- 7.1. Log in to your **Lab Server** instance using SSH.
- 7.2. Install MySQL tool.

```
[ec2-user ~]$ sudo yum install mysql
```

- 7.3. After check state, please key **y** to start install.

```
Is this ok? [y/d/N]: y
```

- 7.4. After installed MySQL, to log-in RDS server, key “**mysql -h [Endpoint] -u [Username] -p**”, and press enter to key **[password]**.

```
[ec2-user ~]$ mysql -h db.choi5coyenv6.us-west-2.rds.amazonaws.com -u labuser -p
```

- 7.5. If you log-in to RDS, you can see the **mysql>**, and you can use the database.

```
mysql> select database();
+-----+
| database() |
+-----+
| NULL      |
```

- 7.6. If you want to leave the RDS, please press **Ctrl+c** to exit.

### Create an alarm in CloudWatch and SNS service

- 8.1. In the **AWS Management Console**, on **service** menu, click **CloudWatch**.
- 8.2. In the navigation pane, choose **Alarms, Create Alarm**
- 8.3. For the Select **Metric** step, you can set up the metric.
- 8.4. Choose a metric category - EC2 Metrics.
- 8.5. Select an instance (LAMP Server ID) and metric - **CPUUtilization**.

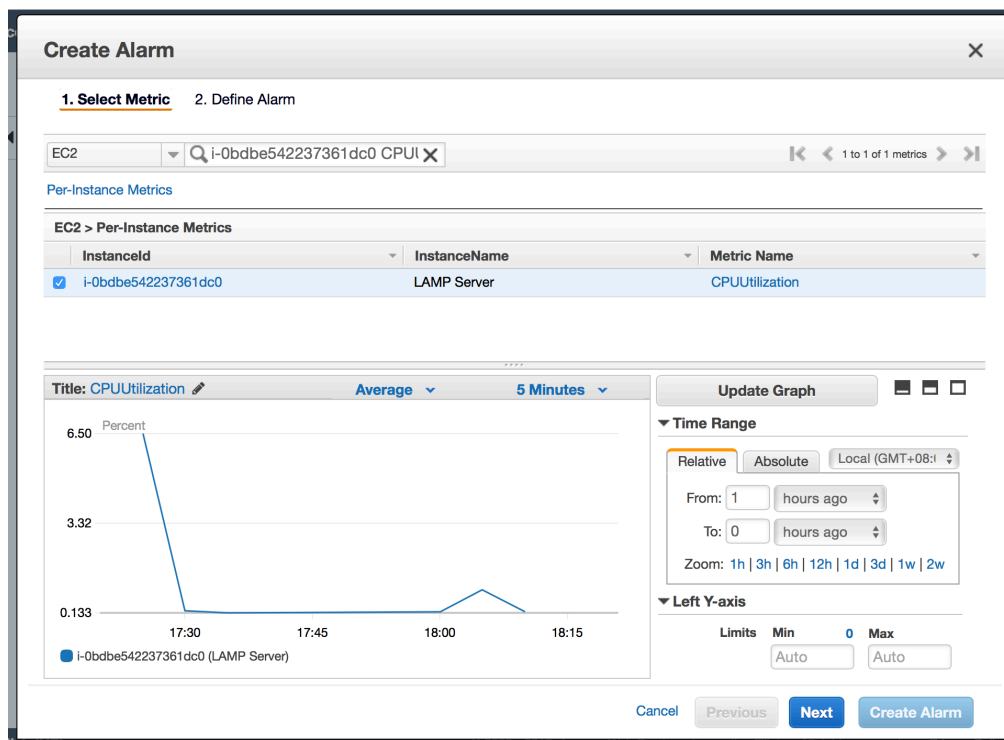


Figure 15: Select EC2 metric - CPUUtilization

- 8.6. For the Define Alarm step, you can set the alarm.
- 8.7. Under Alarm Threshold, type a unique name for the alarm (for example, **myHighCpuAlarm**) and a description of the alarm (for example, **CPU usage exceeds 30 percent**).
- 8.8. Under Whenever, for is, choose **>** and type **30**. For for, type **1**.

## Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to find an appropriate threshold.

**Name:**

**Description:**

---

**Whenever:** CPUUtilization

**is:**

**for:** 1  1  ⓘ

Figure 16: Alarm setting

- 8.9. Under Additional settings, for Treat missing data as, choose **bad (breaching threshold)**, as missing data points may indicate the instance is down
- 8.10. Under Actions, for Whenever this alarm, select State is ALARM. For Send notification to, Click **NEW List**.

Send notification to: **my-topic**

Sendemail : **Your email**

You must confirm the subscription before notifications can be sent.

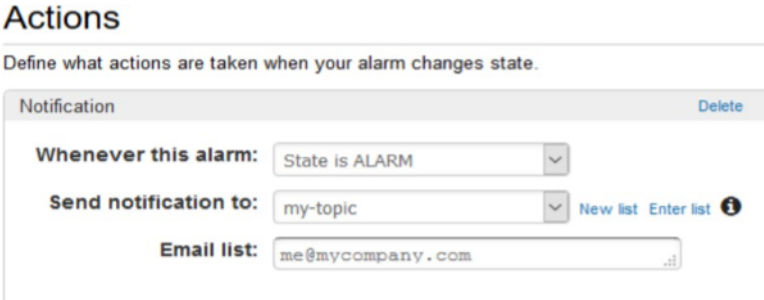


Figure 17: Notification setting

- 8.11. Choose Create Alarm.
- 8.12. When have alarm, the mailbox will get the alarm mail which sent from AWS
- 8.13. To test SNS function, Log in to your **Lab Server** instance using SSH.
- 8.14. You can test Alarm function. Try to increase CPU Utilization to 30%. you can Log in to your **LAMP Server** though SSH.

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

- 8.15. Burn up CPU in 120s.

```
[ec2-user ~]$ sudo stress --cpu 1 --timeout 120
```

- 8.16. If CPU alarm triggers by Cloudwatch, the email box will get the alarm mail which sent from AWS.

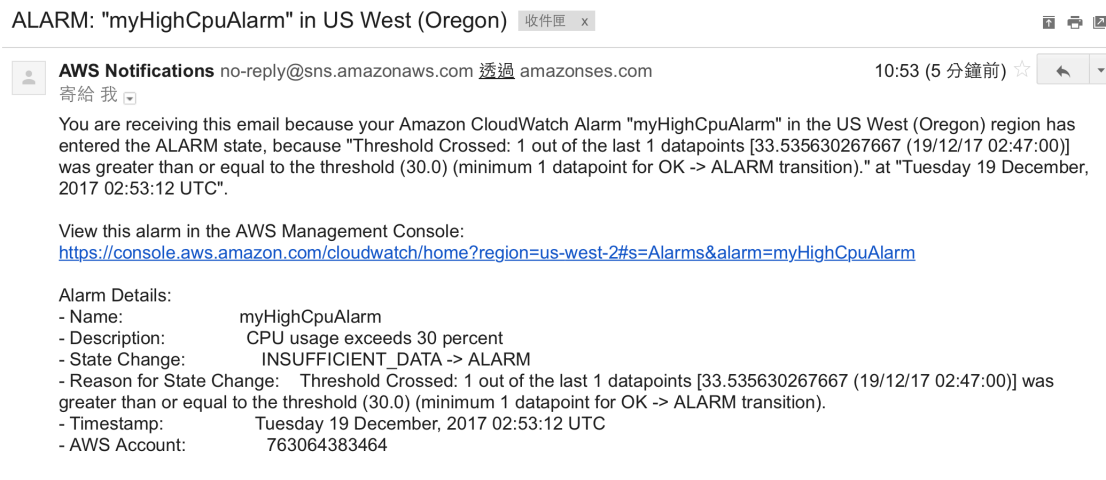


Figure 18: Alarm Email

## 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
- Setup network security included subnets and security group.
- Create an Amazon Relation Database (RDS).
- Logged into your DB instance.
- Monitor service states and send SNS when alarm.