CMPE 281 - LAB #1 - Deploy VM (AWS & GCP)

In this Lab, you will be setting up your AWS account with SSH keys to access EC2 and then will launch a Amazon Linux "Free-Tier" AMI to install a LAMP Stack. You will then deploy some PHP code to remotely create "Load" on the Linux VM and observe the CPU utilization. You will then perform equivalent steps on Google Cloud Platform.

Part 1 - Amazon Web Services - Elastic Compute Cloud (EC2)

Lab Documents & Source Files:

- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html
- https://github.com/paulnguyen/cmpe281/tree/master/labs/lab1

Key Steps Are:

1. Setup Key Pair for EC2 and Download PEM file.
2. Create VPC: cmpe281 (Using Wizard).
3. Launch EC2 Instance.
4. Connect to EC2 Instance.
5. PHP Setup on EC2 Linux AMI.
6. PHP Test.
7. Create PHP AMI Image.

SETUP - CREATE EC2 KEY PAIR & ELASTIC IP

- In the EC2 Dashboard, select one of the following ***regions***:
  - ***US West (Oregon) - US West 2***
  - ***US East (N. Virgina) - US East 1***
  - ***REF:*** ***https://www.concurrencylabs.com/blog/choose-your-aws-region-wisely/***

- Create an EC2 Key Pair named (for your region):
  - **cmpe281-us-west-2**
  - **cmpe281-us-east-1**

- Download the Key Pair to your local machine and set it's permissions appropriately

- In the EC2 Dashboard, Elastic IP section, allocate a new Elastic IP in VPC Scope.

CREATE VPC

- Go to the VPC Dashboard and Start the ***VPC Wizard*** in your Region:
  - ***US West (Oregon) - US West 2 or US East (N. Virgina) - US East 1***
  - **VPC NAME:** CMPE281
  - **WIZARD OPTION:** Public with Private Subnets
  - **Elastic IP:** The Elastic IP you created in SETUP Step.

Creates: A /16 network with two /24 subnets.
Public subnet instances use Elastic IPs to
access the Internet.

Do not select this option:
Private subnet instances access the Internet
via Network Address Translation (NAT).
(Hourly charges for NAT devices apply.)

Instead, select this option (to avoid charges to your account):
In "Specify the details of your NAT gateway" Section,
Select "Use NAT Instance Instead".
NAT Instance Type: **t2.micro**
NAT Instance Keypair: **cmpe281-us-west-*2 (or cmpe281-us-east-1)***

CIDR block	10.0.0.0/16
IP range	10.0.0.0 - 10.0.255.255
Subnet Mask	255.255.0.0
IP Quantity	65536
Public Subnet:	10.0.0.0/24
Network =	10.0.0.0
Usable IPs =	10.0.0.1 to 10.0.0.254 for 254
Broadcast =	10.0.0.255
Netmask =	255.255.255.0
Wildcard Mask =	0.0.0.255
Private Subnet:	10.0.1.0/24
Network =	10.0.1.0
Usable IPs =	10.0.1.1 to 10.0.1.254 for 254
Broadcast =	10.0.1.255
Netmask =	255.255.255.0
Wildcard Mask =	0.0.0.255

LAUNCH A NEW EC2 INSTANCE INTO YOUR CMPE281 VPC AS FOLLOWS

Amazon Linux AMI

T2 Micro Instance

VPC: cmpe281

Public Subnet

Auto Assign Public IP

Security Group: cmpe281-dmz (create new)

    Open Ports: 22, 80, 443

Select Key Pair: cmpe281-us-west-2 (or cmpe281-us-east-1)

AWS Instance Name: aws-php

CONNECT TO EC2 INSTANCE

```
chmod 400 cmpe281-us-west-1.pem
```

Connect to your instance using its Public DNS (For Example):

```
ssh -i <pem file> <ec2-vm-external-host-name>
```

PHP SETUP ON EC2 LINUX AMI

1. Update Yum and Install LAMP Stack

```
sudo yum update -y
sudo yum install -y httpd24 php56 mysql55-server php56-mysqlnd
sudo service httpd start
sudo chkconfig httpd on
chkconfig --list httpd
```

2. Apache/PHP Web Root

```
/var/www/html
sudo groupadd www
sudo usermod -a -G www ec2-user

groups

sudo chown -R root:www /var/www
sudo chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
find /var/www -type f -exec sudo chmod 0664 {} \;
```

PHP TEST

1. Hello LAMP / PHP

```
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

2. Go to: http://<public dns or ip>/phpinfo.php

3. sudo yum install stress

```
stress [OPTION]

## Stress using CPU-bound task
stress -c 4

## Stress using IO-bound task
stress -i 2

## Example Test

stress -c 2 -i 1 -m 1 --vm-bytes 128M -t 10s
```

Where,

```
-c 2 : Spawn two workers spinning on sqrt()
-i 1 : Spawn one worker spinning on sync()
-m 1 : Spawn one worker spinning on malloc()/free()
--vm-bytes 128M : Malloc 128MB per vm worker (default is 256MB)
-t 10s : Timeout after ten seconds
-v : Be verbose
```

CREATE PHP AMI IMAGE

1. Create cmpe281-ami
2. From aws-php (EC2 instance)

Part 2 - Google Cloud Platform - Compute Engine

Lab Documents & Source Files:

- https://cloud.google.com/compute/docs
- https://cloud.google.com/vpc/docs
- https://cloud.google.com/compute/docs/instances/create-start-instance
- https://cloud.google.com/compute/docs/images/create-delete-deprecate-private-images
- https://github.com/paulnguyen/cmpe281/tree/master/labs/lab1

Key Steps Are:

1. Create Project: cmpe281
2. Create VPC Network
3. Delete Default Network
4. Create VM Instance
5. Connect to VM Instance
6. Install and Set Up PHP
7. Create PHP Image

CREATE NEW PROJECT: CMPE281

Note your Project ID (for example): cmpe281-263314

CREATE VPC NETWORK

1. Name: cmpe281
2. Subnet Creation Mode: automatic
3. Firewall rules: select all
4. Dynamic routing mode: regional
5. DNS server policy: none

DELETE THE DEFAULT VPC NETWORK

1. Select VPC Networks List
2. Select Default VPC
3. Select to "Delete VPC Network"

CREATE VM INSTANCE

1. Select your project: cmpe281

2. Select Compute Engine (on Left Nav)

3. Click on "Create" Button to Create a new VM instance

Instance Name:	php
Region:	us-west1 (or us-west2)
Zone:	us-west1-a (or us-west2-a)
Machine Family:	N1
Machine Type:	f1-micro
Boot Disk:	Debian GNU/Linux 9 / 10gb Disk (default)
Identity and API Access:	Leave Defaults
Firewall:	Select "Allow HTTP access"

CONNECT TO VM INSTANCE

1. In VM Instances list, note the "green" status of running php instance

2. Select SSH "pull down" and pick "Open in Browser Window"

PHP SETUP:

1. Update APT and Install PHP

```
sudo apt update
sudo apt install apache2
sudo apt install php libapache2-mod-php
```

2. Test Apache Service

Click on ling to default HTTP Port for your Public IP
Or, type in: http://<your public ip> on Browser

Note WWW Document Root: /var/www/html

PHP TEST

1. Hello LAMP / PHP

Create a file: /var/www/html/phpinfo.php
With Content: <?php phpinfo(); ?>
Note: using vi as sudo

2. Go to: http://<public ip>/phpinfo.php

INSTALL STRESS TEST TOOL

```
sudo apt-get install stress

stress [OPTION]

## Stress using CPU-bound task
stress -c 4

## Stress using IO-bound task
stress -i 2

## Example Test

stress -c 2 -i 1 -m 1 --vm-bytes 128M -t 300
```

Where,

```
-c 2 : Spawn two workers spinning on sqrt()
-i 1 : Spawn one worker spinning on sync()
-m 1 : Spawn one worker spinning on malloc()/free()
--vm-bytes 128M : Malloc 128MB per vm worker (default is 256MB)
-t 10s : Timeout after ten seconds
-v : Be verbose
```

In VM Instance "Info Panel", view the Monitoring Pane
and observer CPU Utilitzation Increase.

UPLOAD PHP SOURCE FILES TO WEB ROOT FOLDER

1. fibonacci.php
2. loadtest.php

In VM Instance SSH Web Shell, Select the "Gear" Icon
and select "Upload File" to upload the Lab files:
fibonacci.php and loadtest.php

Move php files to: /var/www/html

CREATE PHP IMAGE

1. Stop Running php Instance

2. Go to Compute Engine / Images / Create Image

Name:	php-image
Source:	Disk
Source Disk:	php
Location:	Multi-Region
Remaining Options:	Defaults