Try it out objective

Use this hands-on to get started with private networking on the cloud also known as Virtual Private Cloud (VPC).

The goal

The following are the goals of this hands-on:

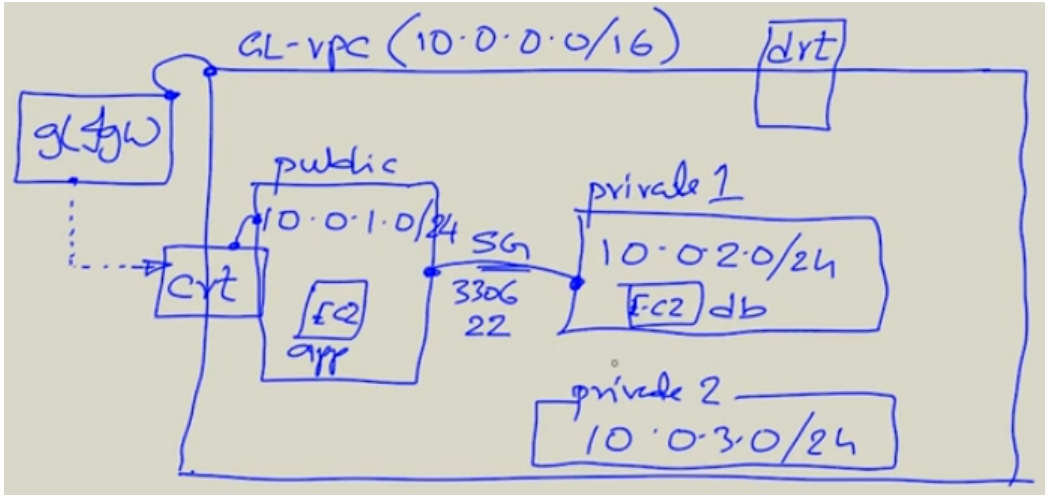
1.Create a VPC

2.Create a public and one private subnet

3.Create Internet gateway and NAT gateway

4.Create a route table, add entries to it

5.Launch EC2 instances in public and private subnets



Module 17 - Creating and using VPC (6:02)

Please note if a field (short for text field/text area/checkbox/radio/dropdown/list or any other UI element) is not specified in the following steps, it means the default value of the field set by AWS needs to be used. No change is needed for those fields as part of this hands-on.

A. Hands-on: Create a VPC

1.Go to the VPC management console at https://console.aws.amazon.com/vpc/ (you will be required to sign in)

2.Ensure the region is **N Virginia**

3.In the left navigation, under **Virtual Private Cloud**, choose **Your VPCs**

4.Click on **Create VPC** button on the right top of the screen

5.Select the option **VPC Only** in the **Resources to create** section.

6.Under the **VPC settings** card, use the following -

a) Paste the below text for the **Name tag** -

```
gl-vpc
```

b) Paste the below text for the **IPv4 CIDR block** field -

```
10.0.0.0/16
```

c) Click on **Create VPC** button

B. Hands-on: Create the subnets

1.In the left navigation, under **Virtual Private Cloud**, choose **Subnets**

2.Click on **Create subnet** button on the right top of the screen

3.In the **VPC** card for the **VPC ID dropdown** select the entry that has **gl-vpc**

4.In the **Subnet settings**, for the **Subnet 1 of 1** card use the following settings -

a) Paste the below value for the **Subnet name** field

```
public
```

b) For the **Availability Zone** dropdown select **us-east-1a**

c) Paste the below text for the **IPv4 CIDR block** field -

```
10.0.1.0/24
```

d) Click on the **Add new subnet** button at the bottom of the card (notice the first entry changes from “1 of 1” to “1 of 2”)

5.In the **Subnet settings**, for the **Subnet 2 of 2** card use the following settings -

a) Paste the below value for the **Subnet name** field

```
private
```

b) For the **Availability Zone** dropdown select **us-east-1b**

c) Paste the below text for the **IPv4 CIDR block** field -

```
10.0.2.0/24
```

6.Click on **Create subnet** button at the bottom of the page

7.In the left navigation, under **Virtual Private Cloud**, choose **Subnets** “again”

8.Click on the **checkbox** to the left of the **public** subnet

9.Click on **Actions** dropdown to the right top of the screen and select **Modify auto-assign IP settings** option

10.Click on the **checkbox** to the left of **Enable auto-assign public IPv4 address**

11.Click on the Save button

C. Hands-on: Create and attach the Internet Gateway

1.In the left navigation, under **Virtual Private Cloud**, choose **Internet gateways**

2.Click on **Create internet gateway** button on the right top of the screen

3.For the **Name tag** paste the below value

```
gl-igw
```

4.Click on the **Create internet gateway** button on the bottom of the page

5.In the left navigation, under **Virtual Private Cloud**, choose **Internet gateways** “again”

6.Click on the **checkbox to the left** of the internet gateway by the name **gl-igw**

7.Click on the **Actions** dropdown to the right top of the screen and select **Attach to VPC** option

8.Select **gl-vpc** in the dropdown for **Available VPCs**

9.Click on **Attach internet gateway** button

D. Hands-on: Create and use a custom route table

1.In the left navigation, under **Virtual Private Cloud**, choose **Route Tables**

2.Click on **Create route table** button on the right top of the screen

3.Under the **Route table settings** card and for the **Name** field paste the below value -

```
public-crt
```

4.Select **gl-vpc** for the **VPC dropdown**

5.Click on the **Create route table** button at the bottom of the page

6.In the left navigation, under **Virtual Private Cloud**, choose **Route Tables** “again”

7.In the listing of the route tables click on the **checkbox** to the left of the entry **public-crt**

8.In the **bottom half of the page** the details of the route table is visible and the default **tab** is **Routes**

9.Click on the **Edit routes** button on the right side near the bottom of the page

10.Click on **Add route** button

11.Paste the below text for the **Destination** field -

```
0.0.0.0/0
```

12.Click in the **Target** field and from the dropdown pick **Internet gateway** and select **gl-igw**

13.Click on the **Save changes** button

14.In the bottom part of the screen locate and click the **Subnet associations** tab

15.In the **Explicit subnet associations** card locate the **Edit subnet associations** button on the right side of the screen in the card

16.Click on the **checkbox** to the left of the **public** subnet

17.Click on **Save associations** button

E. Hands-On: Launch an instance in the public subnet

1.Open the EC2 management console at https://console.aws.amazon.com/ec2/

2.Change the region to **N Virginia** (if it is not already selected)

3.Start the process of launching an EC2 instance using the following values

a) AMI: **Amazon Linux 2**

b) Instance type: **t2.micro**

c) VPC: **gl-vpc**

d) Subnet: **public**

e) Paste the below script for the **User data**

Important note - please copy the complete script properly. A typical mistake is to not select the first and the last few characters.

```
#!/bin/bash
yum update -y
yum install httpd -y
service httpd start
chkconfig httpd on
IP_ADDR=$(curl http://169.254.169.254/latest/meta-data/public-ipv4)
echo "Instance in the public subnet with IP $IP_ADDR" > /var/www/html/index.html
```

f) Proceed to **Step 4: Add Storage**, there is **no change** in this step

g) Proceed to **Step 5: Add Tags**, there is **no change** in this step

h) Proceed to **Step 6: Configure Security Group**, create a security group that **opens port 22 and 80**

i) Proceed to **launch the instance** using the **existing PEM** file

j) Visit the **EC2 instance listing** page, copy the **public IP** and paste in a **new browser tab**, observe the html page

F. Hands-On: Launch an instance in the private subnet

1.Open the EC2 management console at https://console.aws.amazon.com/ec2/

2.Change the region to **N Virginia** (if it is not already selected)

3.Start the process of launching an EC2 instance using the following values

a) AMI: **Amazon Linux 2**

b) Instance type: **t2.micro**

c) VPC: **gl-vpc**

d) Subnet: **private**

e) Proceed to **Step 4: Add Storage**, there is **no change** in this step

f) Proceed to **Step 5: Add Tags**, there is **no change** in this step

k) Proceed to **Step 6: Configure Security Group**, create a security group using the following values -

i. Ensure the radio button **Create a new security group** is selected (should be already by default)

ii. Paste the value of the **Security Group Name** field from below (remove any existing value from this field)

```
private-sg
```

iii. Paste the value of the **Description** field from below (remove any existing value from this field)

```
Opens security groups for ssh and mysql only from the public subnet
```

iv. A rule for SSH is already added, change the **Source** dropdown to **Custom** and paste the following value in the field to the right under source

```
10.0.1.0/24
```

v. Click on the **Add Rule** button to add the second rule for this security group

vi. Click the **Type** dropdown of this row (not the earlier SSH row) and select **All ICMP - IPv4**, change the **Source** dropdown to **Custom** and paste the following value in the field to the right under source (MySQL will not be installed, port for ping is being used as a substitute)

```
10.0.1.0/24
```

vii. Click on **Review and Launch** button on the right side bottom of the page using the existing PEM file

4.Visit the **EC2 instance listing** page, notice there is no public IP address

G. Hands-On: SSH and accessing the private EC2 instance

This step of the hands-on is only for learners with administrative skills or are from technical background with understanding of Linux commands. This is an optional step and can be skipped. Proceed to the next step if you intend to skip it.

1.Go to the browser tab EC2 management console

2.Open a terminal window in your laptop/desktop

3.Use the scp command to **copy the PEM file** from the **laptop**/desktop to the **public instance**

Important note - please replace the bold areas of the command below with your own values. Do copy the complete command properly. A typical mistake is to not select the first and the last few characters

```
scp -i YOUR.pem ./YOUR.pem ec2-user@PUBLIC_IP:/home/ec2-user/YOUR.pem
```

4.Use the PEM file to SSH to the public instance

5.Once the SSH to the public instance succeeds **SSH** using the **same PEM** file to the **private instance**

Important note - please replace the bold areas of the command below with your own values. Do copy the complete command properly. A typical mistake is to not select the first and the last few characters

```
ssh -i YOUR.pem ec2-user@PRIVATE_IP_OF_THE_PRIVATE_INSTANCE
```

6.Execute any yum command and it will not work (sudo yum update)

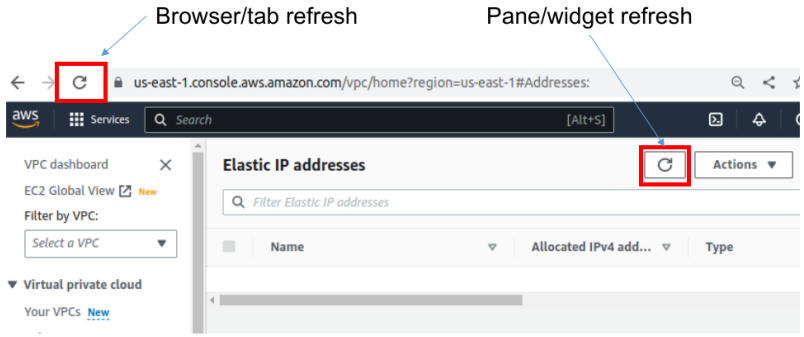
7.**Create a NAT** gateway, use the **public subnet**, add it to the **default route table** (destination of 0.0.0.0/0)

8.The **yum commands** in the private EC2 instance **will now work**

H. Hands-On: Cleaning up!

1.From the VPC management console **Delete the NAT** gateway. It takes a few minutes for the resources and the network interface to get deleted. Please wait for about a minute.

2.Using the left navigation visit the **Elastic IPs** option and **release the Elastic IP** address that was assigned at the time of creating the NAT gateway. This step is required if you see any IPs listed in **Elastic IP addresses** table. If you get an error then refresh the page using the **browser refresh** icon and try again.



3.Go to the browser tab EC2 management console

4.Terminate all EC2 instances

5.Go to the browser tab VPC management console and **delete the gl-vpc only** (care should be take to not delete the default VPC)