

EJ Lilagan, Kyrstn Hall

Prof. Gutta

CS446 - Cloud Computing

03 March 2023

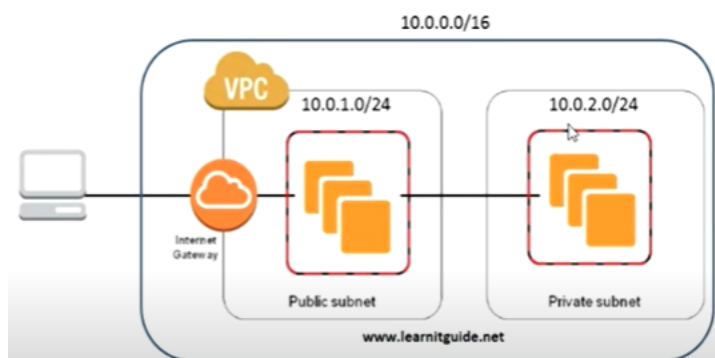
## AWS Virtual Private Cloud (VPC)

VPC will be successfully launched from the youtube video provided on canvas. To expand on the VPC, photos and summarizations are applied in this document. Before we show our demonstration of launching the VPC, here are its background of what VPC is/what it does.

Further details that point up about VPC include the following:

- Virtual network that can be created in a public cloud given an AWS account with sufficient access
- Isolated from other virtual networks in AWS cloud based on its requirements with what has been created.
- It can specify an IP address range for the VPC by adding subnets and associating security groups and configure the route tables, in order to launch database resources, such as Amazon EC2 instances within the VPC.

Given this photo we are trying to accomplish the following within the code



## Checklist:

1. One VPC instance with IP of 10.0.0.0/16
2. Create two subnets inside the VPC
  - a. Private with IP of 10.0.2.0/24 where its resource is 10.0.2
  - b. Public with IP of 10.0.1.0/24 where its resource is 10.0.1
3. Public subnet must be attached to an internet gateway, which is where its resource will be obtained via the internet access.
4. On the other hand, private subnet will not have any access to the external world as its gateway is accessed from public subnet.

The following process starts below:

### Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

#### VPC settings

Resources to create [Info](#)  
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only

☐ VPC and more

Name tag - *optional*  
Creates a tag with a key of 'Name' and a value that you specify.

my\_demo\_vpc

IPv4 CIDR block [Info](#)

☒ IPv4 CIDR manual input

☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/16

IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block

☐ IPAM-allocated IPv6 CIDR block

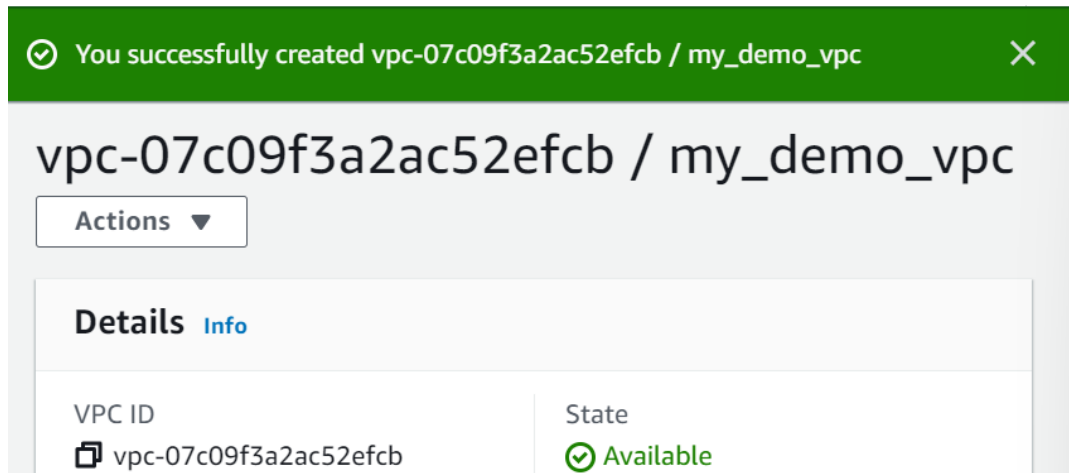
☐ Amazon-provided IPv6 CIDR block

☐ IPv6 CIDR owned by me

Tenancy [Info](#)

Default

Creating a VPC given the following instructions



Pop up message that VPC has been created

VPC > Internet gateways > Create internet gateway

## Create internet gateway [Info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

### Internet gateway settings

**Name tag**  
Creates a tag with a key of 'Name' and a value that you specify.

### Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	
<input type="text" value="Name"/>	<input type="text" value="my_demo_igw"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

Creating an internet gateway for the public subnet

The following internet gateway was created: igw-04e6e5b278a0293bd - my\_demo\_igw. You can now attach to a VPC to enable the VPC to communicate with the internet.




×

Attach to a VPC

igw-04e6e5b278a0293bd / my\_demo\_igw


Actions ▾

Details [Info](#)


Internet gateway ID	State
 igw-04e6e5b278a0293bd	 Detached
VPC ID	Owner
-	 668444490235

Tags 

Manage tags

 Search tags

< 1 >



Key	Value
Name	my_demo_igw

Showing the details that the internet gateway has been made without being attached to any subnets.

## Attach to VPC (igw-04e6e5b278a0293bd) [Info](#)

### VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

#### Available VPCs

Attach the internet gateway to this VPC.

Q vpc-07c09f3a2ac52efcb



► AWS Command Line Interface command

Cancel

Attach internet gateway

Selecting the VPC to attach with the internet gateway.

VPC > ... > igw-04e6e5b278a0293bd



Internet gateway igw-04e6e5b278a0293bd successfully attached to vpc-07c09f3a2ac52efcb



igw-04e6e5b278a0293bd /  
my\_demo\_igw

Actions ▼

### Details [Info](#)

Internet gateway ID

igw-04e6e5b278a0293bd

State

Attached

VPC ID

vpc-07c09f3a2ac52efcb |  
my\_demo\_vpc

Owner

668444490235

# Create subnet [Info](#)

## VPC

### VPC ID

Create subnets in this VPC.

vpc-07c09f3a2ac52efcb (my\_demo\_vpc) ▼

### Associated VPC CIDRs

#### IPv4 CIDRs

10.0.0.0/16

## Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

### Subnet 1 of 1

#### Subnet name

Create a tag with a key of 'Name' and a value that you specify.

Public Subnet

The name can be up to 256 characters long.

#### Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference ▼

#### IPv4 CIDR block [Info](#)

🔍 10.0.1.0/24



# Create subnet [Info](#)

## VPC

### VPC ID

Create subnets in this VPC.

vpc-07c09f3a2ac52efcb (my\_demo\_vpc) ▼

### Associated VPC CIDRs

#### IPv4 CIDRs

10.0.0.0/16

## Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

### Subnet 1 of 1

#### Subnet name

Create a tag with a key of 'Name' and a value that you specify.

Private Subnet

The name can be up to 256 characters long.

#### Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference ▼

#### IPv4 CIDR block [Info](#)

🔍 10.0.2.0/24



✓ You have successfully created 1 subnet: subnet-0e5d2717a772df3c4

## Subnets (2) [Info](#)



Actions ▼

Create subnet

🔍 *Filter subnets*

< 1 > ⚙️

<input type="checkbox"/>	Name ▼	Subnet ID ▼	State
<input type="checkbox"/>	Public Subnet	subnet-0bb8ac60b79bc7597	✓ Available
<input type="checkbox"/>	Private Subnet	subnet-0e5d2717a772df3c4	✓ Available

Creating both public and private subnets that are created within the VPC.

– Making route tables for both subnets –

## Manage tags for rtb-0f5a1b215868fb842 [Info](#)

### Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key

Value - optional

🔍 Name

🔍 Public RT



Remove

Add new tag

You can add 49 more tags.



## Create route table [Info](#)

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

### Route table settings

#### Name - *optional*

Create a tag with a key of 'Name' and a value that you specify.

#### VPC

The VPC to use for this route table.

✓ Route table rtb-0c87bca621e206380 | Private RT was created successfully.



### Route tables (2) [Info](#)



Actions ▼

Create route table

🔍 Filter route tables

< 1 > ⚙️

<input type="checkbox"/>	Name ▼	Route table ID ▼	Explicit subne
<input type="checkbox"/>	Private RT	<a href="#">rtb-0c87bca621e206380</a>	–
<input type="checkbox"/>	Public RT	<a href="#">rtb-0f5a1b215868fb842</a>	–

## Edit routes

### Edit routes

Destination

10.0.0.0/16

Target

Q local



Status

✔ Active

Propagated

No

### Edit routes

Destination

Q 0.0.0.0/0



Target

Q igw-04e6e5b278a0293



Status

-

Propagated

No

Remove

Add route

Cancel

Preview

Save changes

Searched target by clicking on “Internet Gateway” instead of typing “igw”

✔ Updated routes for rtb-0f5a1b215868fb842 / Public RT successfully

► Details

## Edit subnet associations

Change which subnets are associated with this route table.

**Available subnets (1/2)**

< 1 >

⚙

	Name ▾	Subnet ID ▾	IPv4 CIDR ▾	IPv6 CIDR
<input checked="" type="checkbox"/>	Public Subnet	subnet-0bb8ac60b79bc...	10.0.1.0/24	–
<input type="checkbox"/>	Private Subnet	subnet-0e5d2717a772d...	10.0.2.0/24	–

**Selected subnets**

subnet-0bb8ac60b79bc7597 / Public Subnet ✕

Connecting public subnet to the Public Route Table

## Edit subnet associations

Change which subnets are associated with this route table.

**Available subnets (1/2)**

< 1 >

⚙


	Name ▾	Subnet ID ▾	IPv4 CIDR ▾	IPv6 CIDR
<input type="checkbox"/>	Public Subnet	subnet-0bb8ac60b79bc...	10.0.1.0/24	–
<input checked="" type="checkbox"/>	Private Subnet	subnet-0e5d2717a772d...	10.0.2.0/24	–

**Selected subnets**


subnet-0e5d2717a772df3c4 / Private Subnet ✕

Same process again but for associating Private Subnet with Private Route Table

## Security Groups (1/1) [Info](#)

 **Actions** ▼ **Export security groups to CSV** ▼

Create security group

< 1 > 

<input checked="" type="checkbox"/>	Name ▼	Security group ID ▼	Security group n
<input checked="" type="checkbox"/>	my_demo_sg	sg-00e9b243a16cd3b51	default

Create security group to protect resources from accessing the internet

### Inbound rule 2

[Delete](#)

Security group rule ID

–

Type [Info](#)

All traffic ▼

Protocol [Info](#)

All

Port range [Info](#)

All

Source type [Info](#)

My IP ▼

Source [Info](#)

99.150.249.82/32 ✕

Description - optional [Info](#)

My Lab IP

Creating an inbound rule

Inbound rule

Rule number

Info

100

Type

Info

All traffic

Protocol

Info

All

Port range

Info

All

Source

Info

99.150.249.82/32

Allow/Deny

Info

Allow

Remove

Edit inbound rule for the Network ACL's

–Transition to EC2 instance–

Amazon Linux

aws

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

Search icon

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

Free tier eligible

ami-060d3509162bcc386 (64-bit (x86)) / ami-0001e277752486de5 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20230221.0 x86\_64 HVM gp2

Architecture

AMI ID

64-bit (x86)

ami-060d3509162bcc386

Verified provider

Setting AMI to Amazon Linux 2 (free tier)

## ▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-07c09f3a2ac52efcb (my\_demo\_vpc)  
10.0.0.0/16



Subnet [Info](#)

subnet-0bb8ac60b79bc7597      Public Subnet  
VPC: vpc-07c09f3a2ac52efcb    Owner: 668444490235  
Availability Zone: us-west-1a    IP addresses available: 251  
CIDR: 10.0.1.0/24



[Create new subnet](#)

Auto-assign public IP [Info](#)

Enable

Network settings included both network and subnet and enabling the public IP

## Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group

☒ Select existing security group

Common security groups [Info](#)

Select security groups

default sg-00e9b243a16cd3b51 ✕  
VPC: vpc-07c09f3a2ac52efcb



[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Select security group to existing, and select common security group to be the default

## Create key pair



Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Key pair name

my-demo-key

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA

RSA encrypted private and public key pair

☐ ED25519

ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

☒ .pem

For use with OpenSSH

☐ .ppk

For use with PuTTY

Creating a new key pair for using VPC

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

my-demo-key ▼

[Create new key pair](#)

▼ Network settings Info

VPC - required Info

vpc-07c09f3a2ac52efcb (my demo vpc)

[Feedback](#)
[Language](#)
[Privacy](#)
[Terms](#)
[Cookie preferences](#)

© 2023, Amazon Web Services, Inc. or its affiliates.

my-demo-key.pem
 

^

[Show all](#)

×

Use the download for this demo

Instances (1/1) Info

Connect

Instance state ▼

Actions ▼

Launch instances ▼

Find instance by attribute or tag (case-sensitive)

<

1

>

<input checked="" type="checkbox"/>	Name ▼	Instance ID	Instance state ▼	
<input checked="" type="checkbox"/>	ec2-public	i-026e78b877039880c	Running	

Once launched, check if on running state and attach instance to public subnet

–Same process again but for private subnet–



## ▼ Network settings [Info](#)

VPC - *required* [Info](#)

vpc-07c09f3a2ac52efcb (my\_demo\_vpc)  
10.0.0.0/16 ▼



Subnet [Info](#)

subnet-0e5d2717a772df3c4 Private Subnet  
VPC: vpc-07c09f3a2ac52efcb Owner: 668444490235  
Availability Zone: us-west-1a IP addresses available: 251  
CIDR: 10.0.2.0/24 ▼



[Create new subnet](#)

Auto-assign public IP [Info](#)

Enable ▼

Do the same but change the subnet for the private subnet.

## Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group

☒ Select existing security group

Common security groups [Info](#)

Select security groups ▼

default sg-00e9b243a16cd3b51 ✕  
VPC: vpc-07c09f3a2ac52efcb



[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Same as to public subnet security configuration

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

my-demo-key ▼

↺

Create new key pair

Use existing key pair that was created from public subnet instance

Instances (2) Info

↺

Connect

Instance state ▼

Actions ▼

Launch instances ▼

🔍 Find instance by attribute or tag (case-sensitive)







< 1 >

⚙️

<input type="checkbox"/>	Name ▼	Instance ID	Instance state ▼
<input type="checkbox"/>	ec2-public <a href="#">🔗</a>	i-026e78b877039880c	✔ Running <a href="#">🔍</a> <a href="#">🔊</a>
<input type="checkbox"/>	ec2-private <a href="#">🔗</a>	i-093ce6831d6966b98	✔ Running <a href="#">🔍</a> <a href="#">🔊</a>

Proof to show public/private were successfully launched

Key differences between both public and private EC2 include the Public IPv4 and private IP

Public IPv4 address  3.101.67.230   <a href="#">open address</a> 	Public IPv4 address  3.101.191.122   <a href="#">open address</a> 
IPv6 address –	IPv6 address –
Public IPv4 DNS –	Public IPv4 DNS –
Private IP DNS name (IPv4 only)  ip-10-0-2-118.us-west-1.compute.internal	Private IP DNS name (IPv4 only)  ip-10-0-1-211.us-west-1.compute.internal

```
C:\Windows\System32>ping 3.101.191.122

Pinging 3.101.191.122 with 32 bytes of data:
Reply from 3.101.191.122: bytes=32 time=114ms TTL=235
Reply from 3.101.191.122: bytes=32 time=38ms TTL=235
Reply from 3.101.191.122: bytes=32 time=37ms TTL=235
Reply from 3.101.191.122: bytes=32 time=39ms TTL=235

Ping statistics for 3.101.191.122:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 37ms, Maximum = 114ms, Average = 57ms
```

Open command prompt and type PING.EXE to see if it is located. Had to go into my windows directory to get the ping command to work.

```
C:\Windows\System32>ping 3.101.191.122

Pinging 3.101.191.122 with 32 bytes of data:
Reply from 3.101.191.122: bytes=32 time=114ms TTL=235
Reply from 3.101.191.122: bytes=32 time=38ms TTL=235
Reply from 3.101.191.122: bytes=32 time=37ms TTL=235
Reply from 3.101.191.122: bytes=32 time=39ms TTL=235

Ping statistics for 3.101.191.122:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 37ms, Maximum = 114ms, Average = 57ms
```

This photo pings the EC2 from the public EC2

```
C:\Windows\System32>ping 3.101.67.230

Pinging 3.101.67.230 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 3.101.67.230:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

This ping has been timed out due to the private subnet not being accessed to the internet gateway.

Challenges/Thoughts: Me and Kyrstn ended up having difficulties starting the demo when following the instructions on the YouTube video. Although the formats were different from current AWS and the video, it was a challenge from launching certain number of subnets/route tables along the way. What has happened was when a subnet was launched the first time, it would save but then the second time the subnet was created, it would not display the first subnet due to filters being activated, which led me to have around 4 subnets in total, and was a hassle to

know which subnet to use when attaching it to the internet gateway, thus led the both of us to restart the process from the very start of the video. A lesson that it both taught us was to not watch the video without volume that ended up being costly, and to pay attention to every detail. Another challenge was that the command ping would be difficult to run on the command prompt until we had to call PING.EXE in one of the windows 32 directory in order to run the IP address that would run the EC2 instance from the public subnet since it is connected to the internet gateway and gave results whereas the private subnet would run but not including any replies due to its resources not directly be allocated to the internet gateway.