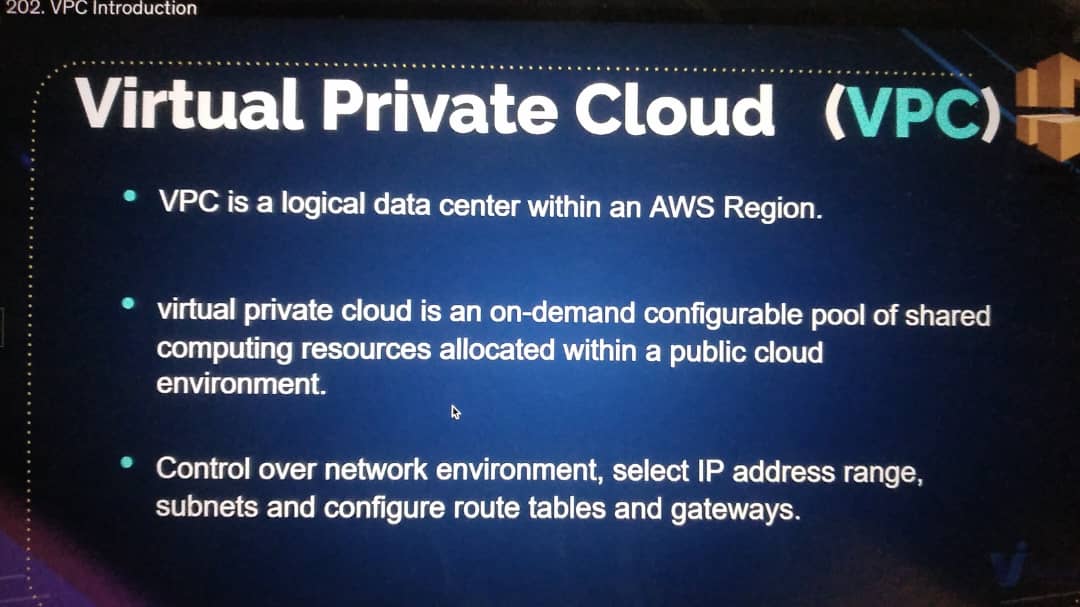
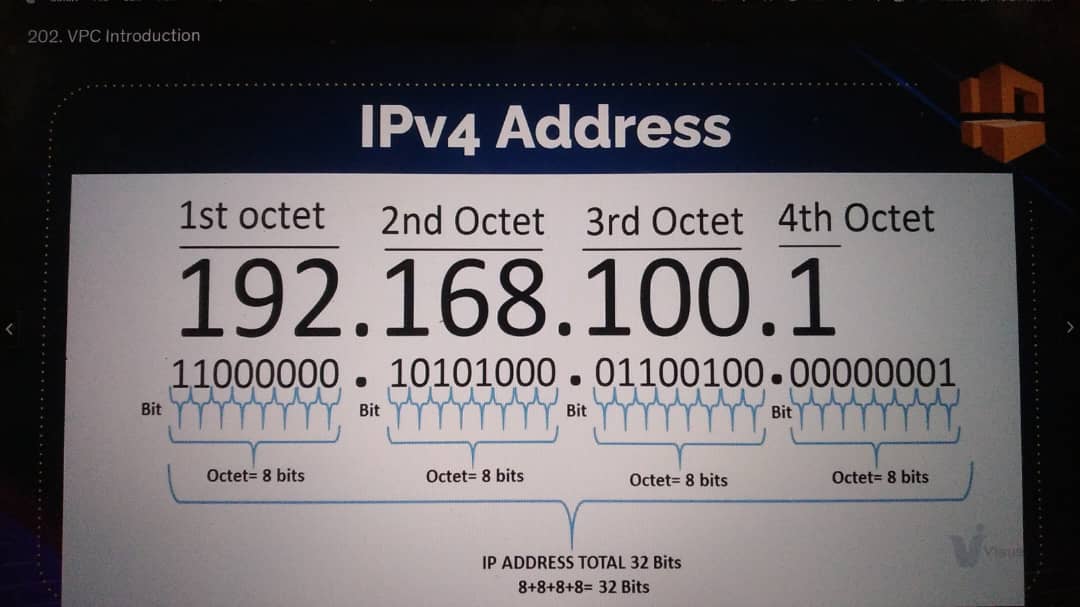
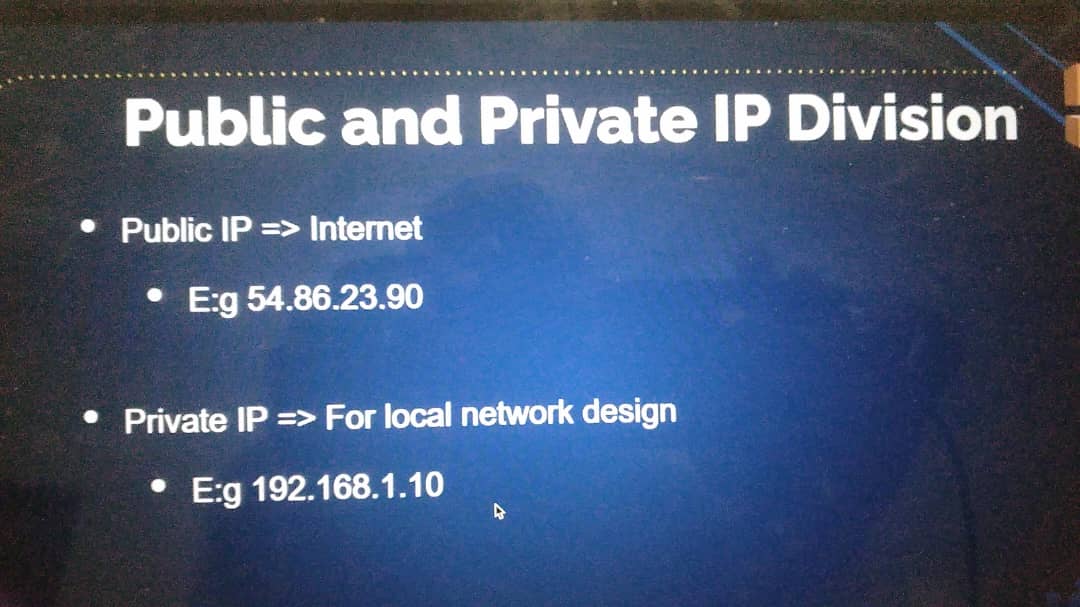
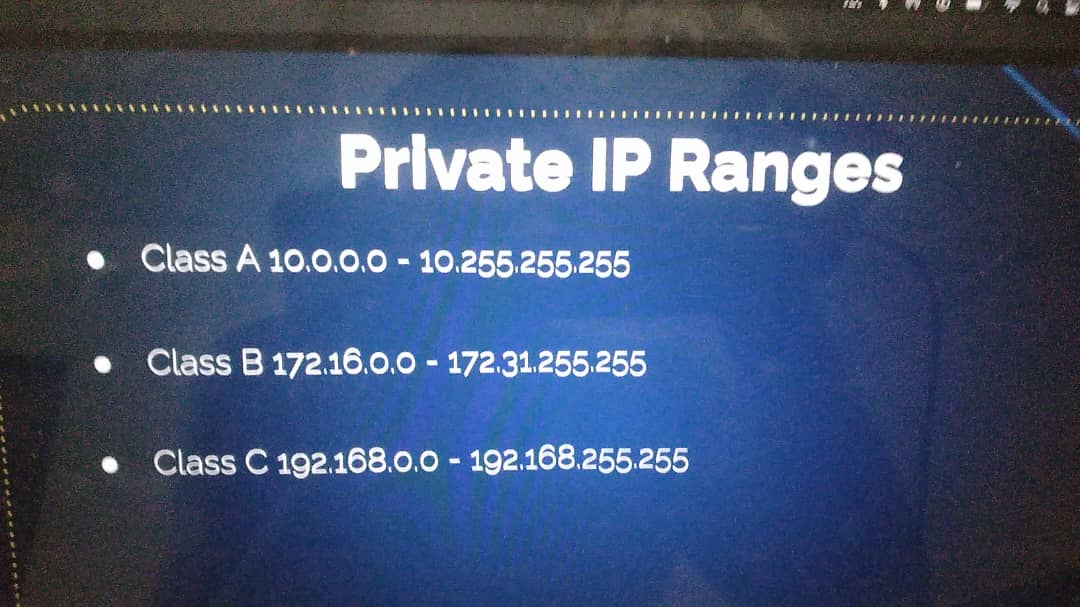
**VIRTUAL PRIVATE CLOUD**

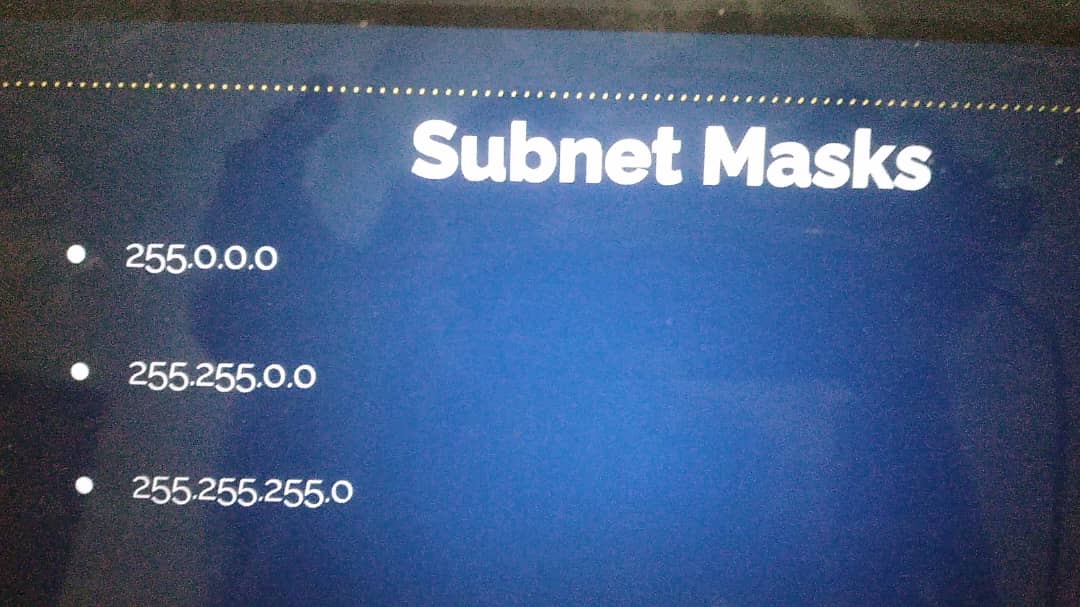
A virtual private cloud (VPC) is the division of a service provider's public cloud multi-tenant architecture to support private cloud computing. VPCs are, therefore, a private cloud hosted within a public cloud architecture.

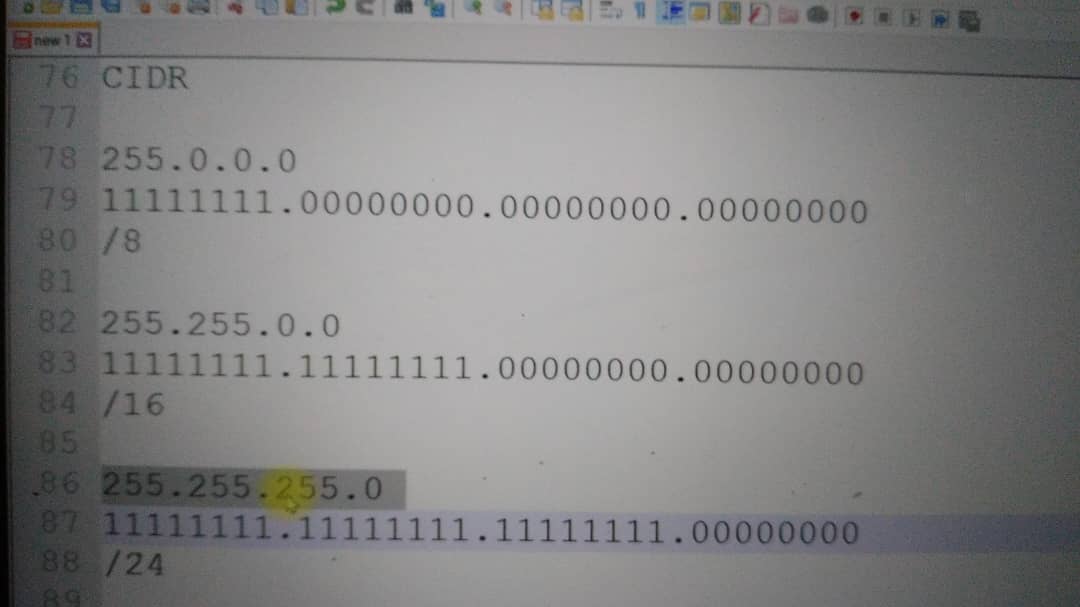






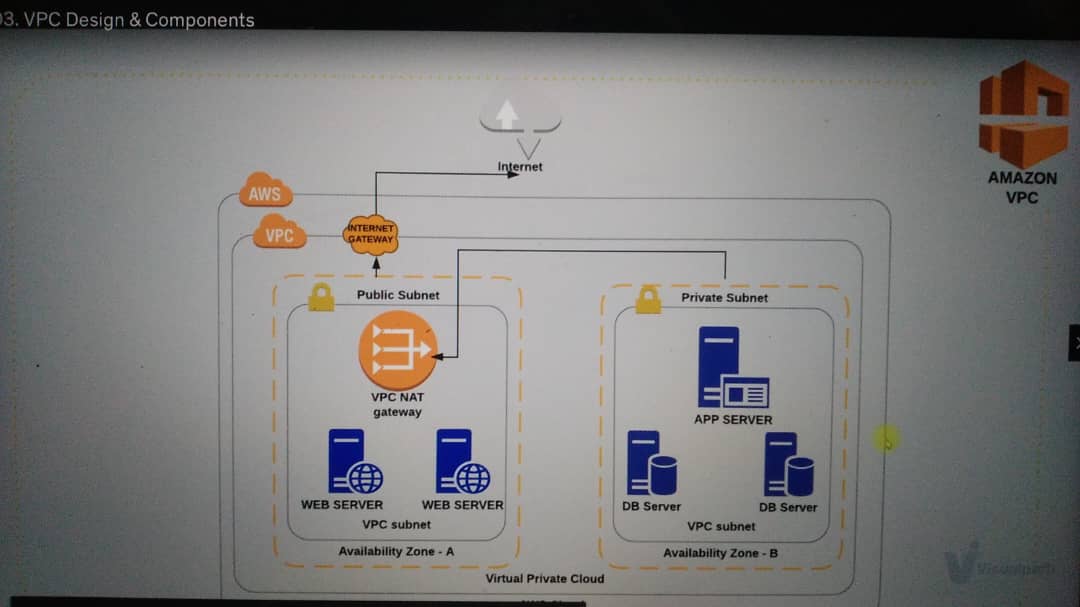


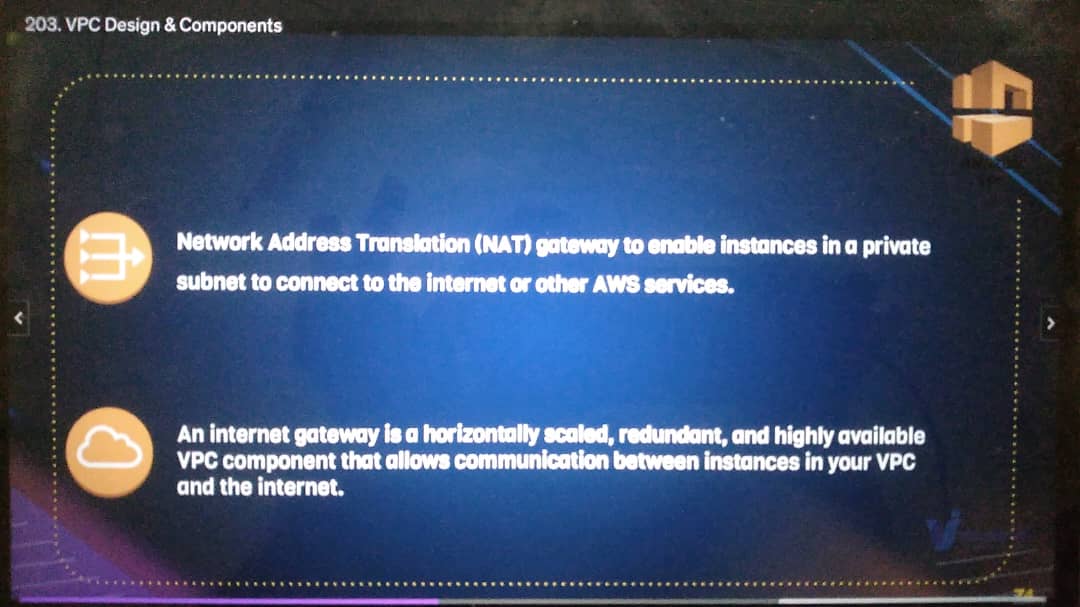


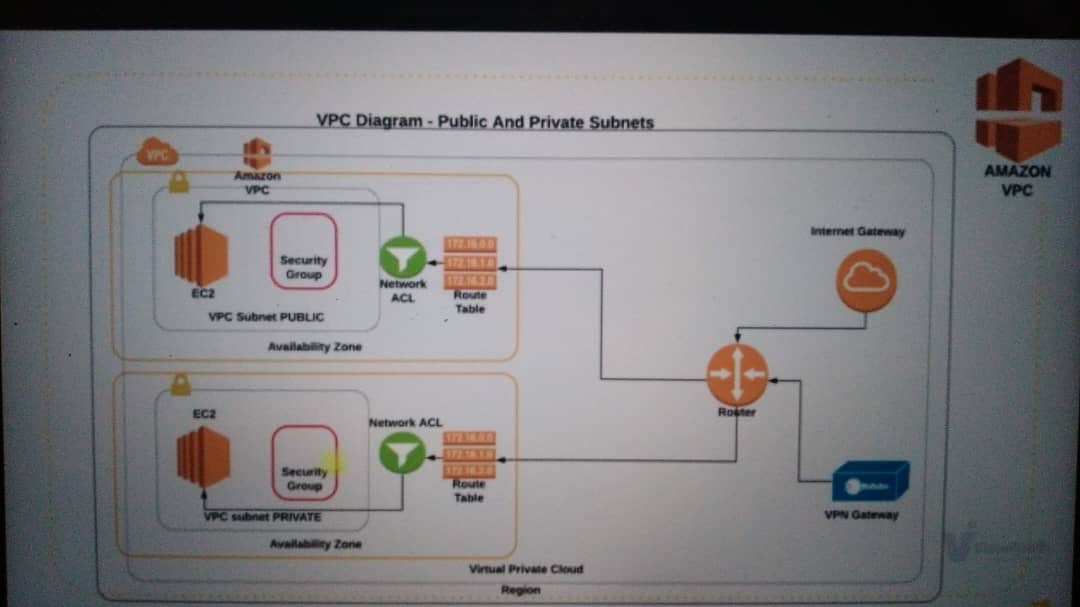


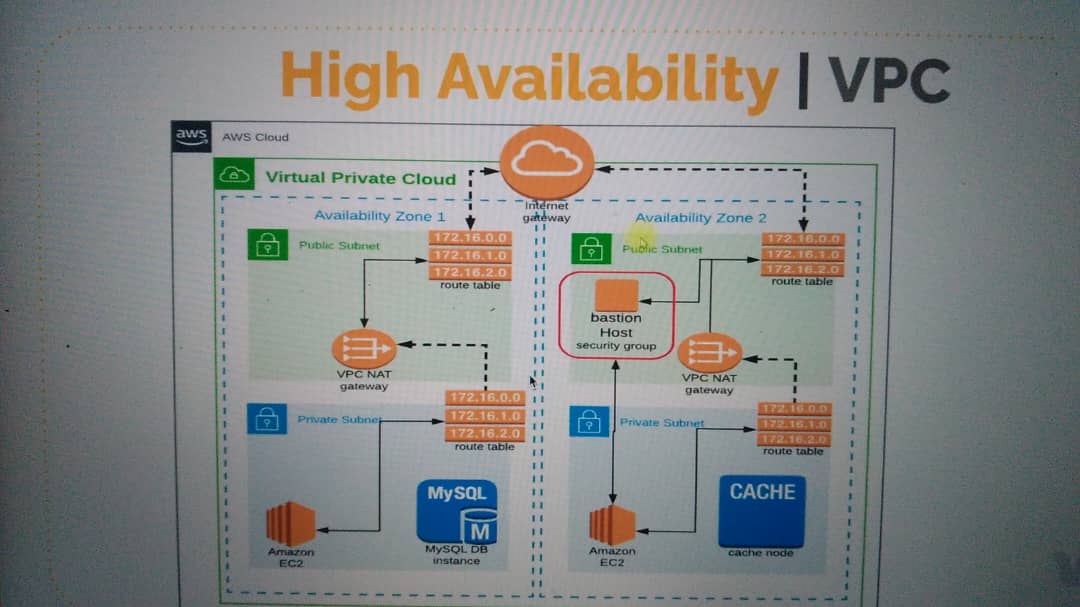
You can go on google, and type subnet calculators. Or: you can use these site: <https://jodies.de/ipcalc>

**VPC DESIGN & COMPONENTS**









**VPC SETUP**

Region: us-west-1

Assuming we have a VPC Range: 172.20.0.0/16

We can divide it, having: 4 Subnets (2 Public Subnets and 2 Private Subnets)

2 Zones: us-west-1a, us-west-1b

172.20.1.0/24 => pub-sub-1 [us-west-1a]

172.20.2.0/24 => pub-sub-2 [us-west-1b]

172.20.3.0/24 => priv-sub-1 [us-west-1a]

172.20.4.0/24 => priv-sub-2 [us-west-1b]

Requiements:

* 1 internet Gateway
* 2 NAT Gateway (For High Availability, but for these project we would use one for the private subnets only)
* 1 Elastic IP for the NAT Gateway
* 2 Route Tables: 1 public subnet, 1 private subnet
* 1 Baston host in the Pub Subnet

Network Access Control List (NACL) for public subnet: It acts as a firewall, it is like the security groups but gives more control. (Security Groups are for the instance, NACL is for the subnets)

* 1 more VPC (without subnets) => VPC Peering

**Steps to create VPC from the above resources:**

1. Login to your AWS account
2. Under the AWS console, click on Network & Content Delivery
3. Click on VPC, and then click on VPC
4. On the left pane, go to Virtual Private cloud section and click on “Your VPC’s”
5. click on create VPC on the right pane

* Name Tag = vprofile-VPC
* IPv4 CIDR block = Paste the VPC Range above here (172.20.0.0/16)
* Tenancy = Default
* key = Name
* Value = vprofile-VPC
* click on create VPC

1. The step above creates the Network Range for VPC and 1 default route table and NACL but we will not use the defaults
2. On the left pane, go to Virtual Private cloud section and click on Subnets
3. click on create subnets (Here, we will create 2 Public Subnets and 2 Private Subnets)

**Public Subnet 1:**

* VPC ID = select the VPC you created (vprofile-VPC)
* Associated VPC CIDRs: 172.20.0.0/16
* subnet name = vpro-pubsub-1
* Availability Zone = US-West-1a
* IPv4 CIDR block = Paste the IPv4 Range for us-1a above here (172.20.1.0/24)
* key = Name
* Value = vpro-pubsub-1
* click on create subnet

**Public Subnet 2:**

* VPC ID = select the VPC you created (vprofile-VPC)
* Associated VPC CIDRs: 172.20.0.0/16
* subnet name = vpro-pubsub-2
* Availability Zone = US-West-1b
* IPv4 CIDR block = Paste the IPv4 Range for us-1b above here (172.20.2.0/24)
* key = Name
* Value = vpro-pubsub-1
* click on create subnet

The Public Subnet is not yet public because Internet Gateway hasn’t been added

**Private Subnet 1:**

* VPC ID = select the VPC you created (vprofile-VPC)
* Associated VPC CIDRs: 172.20.0.0/16
* subnet name = vpro-privsub-1
* Availability Zone = US-West-1a
* IPv4 CIDR block = Paste the IPv4 Range for us-1a above here (172.20.3.0/24)
* key = Name
* Value = vpro-pubsub-1
* click on create subnet

**Private Subnet 2:**

* VPC ID = select the VPC you created (vprofile-VPC)
* Associated VPC CIDRs: 172.20.0.0/16
* subnet name = vpro-privsub-2
* Availability Zone = US-West-1b
* IPv4 CIDR block = Paste the IPv4 Range for us-1b above here (172.20.4.0/24)
* key = Name
* Value = vpro-pubsub-1
* click on create subnet

1. On the left pane, go to Virtual Private cloud section and click on Internet Gateways

* Name Tag = vpro-RT
* click on create internet gateway
* in the next page, click on Actions and click on attach to VPC
* select the VPC and click on Attach internet gateway

1. On the left pane, go to Virtual Private cloud section and click on Route Tables

* click on create a Route Table
* Name Tag = vpro-pub-RT
* VPC = select your VPC you created (vprofile-vpc)
* click on create
* In the next page, click to check the box to select the created Route Table
* click on Subnet Associations tab and click on Edit Subnet association
* select the 2 public subnets and click on save
* click on routes tab beside the subnet association tab, click on edit routes
* click on Add route and type:

Destination: 0.0.0.0/0

Target: Internet Gateway, select the created Route table (vpro-RT)

click on save route…. this is what makes the subnet public

1. On the left pane, go to Virtual Private cloud section and click on NAT Gateway (create for the private subnets)

* click on create NAT Gateway
* Name = vpro-NAT-GW
* Subnet = select the public subnet (vpro-pubsub-1)
* Elastic IP allocation = click on Allocate Elastic IP
* click on create NAT Gateway

1. On the left pane, go to Virtual Private cloud section and click on Route Tables

* click on create a Route Table
* Name Tag = vpro-priv-RT
* VPC = select your VPC you created (vprofile-vpc)
* click on create
* In the next page, click to check the box to select the created Route Table
* click on Subnet Associations tab and click on Edit Subnet association
* select the 2 private subnets and click on save
* click on routes tab beside the subnet association tab, click on edit routes
* click on Add route and type:

Destination: 0.0.0.0/0

Target: NAT Gateway, select the created Route table (vpro-NAT-GW)

click on save route…. this is what makes the private subnet connect to the public subnet

1. On the left pane, go to Virtual Private cloud section and click on Subnets

* select the first public subnet and click on actions button
* click on modify auto-assign IP settings
* check the box Enable auto-assign public IPv4 address
* click on save
* select the second public subnet and click on actions button
* click on modify auto-assign IP settings
* check the box Enable auto-assign public IPv4 address
* click on save

1. On the left pane, go to Virtual Private cloud section and click on Your VPCs

* select your VPC and click on Actions
* click on edit DNS hostnames
* click on enable and click save

**PROJECT:**

Using the created VPC, I will create:

A few instances

* place some of them in a private subnet
* and others in the public subnet
* Use load balancer

Steps:

1. Launch an Instance

* AMI = Centos7 (For Server)
* Configure Instance details:
* Number of Instance = 1
* Network = select the created VPC
* Subnet = private subnet 1
* Additional Info: Provision using the script in the folder
* Tags (Key = Name, Value = web01)
* Security Group
  + SG-Name = web-sg
  + Description = web-sg
  + SSH/TCP/22/myip (from myip, i will not be able to ssh becos it is in a private subnet. I need a Bastion Host instance)
  + create Keypair (Name = wave-key)
  + Go ahead and launch your instance

1. The instance created, if you check will not have a Public IP. We need to create a Bastion Host in the public subnet

* Launch an Instance
* AMI = Amazon Linux (For the Bastion Host)
* Configure Instance details:
* Number of Instance = 1
* Network = select the created VPC
* Subnet = public subnet 1
* Additional Info: Provision using the script in the folder
* Tags (Key = Name, Value = Bastion/JumpServer)
* Security Group
  + SG-Name = bastion-sg
  + Description = bastion-sg
  + SSH/TCP/22/myip
  + create Keypair (Name = Bastion-key)
  + Go ahead and launch your instance
* Connect the Bastion Host will the Web01 server
* copy the private Ip for the Bastion Host EC2 instance
* Go to the Security Tab for web01 EC2, click on Edit Inbound security rules
* SSH / TCP / 22 / (paste the private IP/32)
* click on save rule
* Testing the SSH
* Copy the Public IP of the Bastion Host and Open GitBash
* ssh –i Downloads/bastion-key.pem ec2-user@paste the Public IP
* If you are able to SSH it means the Public subnets are good
* logout
* scp –i Downloads/bastion-key.pem Downloads/wave-key.pem ec2-user@paste the PublicIP:/home/ec2-user/ (meaning: for Authentication it will use Downloads/bastion-key.pem, while Downloads/wave-key.pem is been copied into the path /home/ec2-user/)
* ssh –i Downloads/bastion-key.pem ec2-user@paste the Public IP
* ls –l
* chmod 400 wave-key.pem
* ssh –i wave-key.pem centos@paste the private IP of web01
* type yes and hit enter
* you will be logged into the web01 through the Bastion Host EC2
* sudo –i
* systemctl status httpd
* ls /var/www/html/
* check if the index.html is there

1. Create the Load Balancer in the Public Subnet

* Go to the left pane, scroll down to Load Balancing section
* click on create Load Balancer
* under Class Load Balancer, click on create
* Load Balancer Name = wave-elb
* create LB inside insde = vprofile-VPC
* click to Add vpro-pubsub-1 and vpro-pubsub-2
* HTTP/80 and HTTP/80
* click on next to set security group
* click on create security group
* SG Name = wave-elb-sg
* Description = wave-elb-sg
* customTCP / TCP / 80 / custom / 0.0.0.0/0
* Configure Health checks
* ping protocol = HTTP
* ping port = 80
* ping path = /index.html
* Response Timeout = 5 secs
* Interval = 30 secs
* Unhealthy Threshold = 2
* Healthy Threshold = 2
* In the next page, you will see your instances. select web01
* click next, until you see create. click on it
* Go to the security group of the web server web01 and allow connection from the load balancer (custom TCP / TCP / 80 / custom / load balancer
* click on save rule

1. Click on Description Tab under the Load balancer created, copy the DNS name and paste in the browser

**VPC PEERING**

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same network.

It can be done:

* Across regions
* Across different accounts

STEPS:

* Go to another region and create another VPC with network parameters only different from the VPC IP created above. (Scroll up for Guidelines)
* copy the VPC ID and store in a word
* Return back to the former region you left
* Go to the left pane and click on Peering connections
* click on create Peering connections
* Peering connection Name = region1-region2
* VPC requester = vprofile-VPC
* account = My Account
* Region = Another Region
* select the new region
* VPC ID (Accepter) = paste the copied VPC ID from word
* click on create
* Return back to the new region you created the new VPC in
* Go to the left pane and click on Peering connections
* click to check the request
* click on Actions Tab and click on Accept request.
* click Yes, Accept
* copy the network range of these new VPC (cidr)
* Go to the former region
* on the left pane, click on Route Table
* click to check the public RT (vpro-pub-RT)
* click on routes tab and click on edit routes
* click on Add Routes
* Destination = Paste the network range of these new VPC (cidr)

Target = Peering connection and select the connection name

click on save routes

* Go to the new region
* on the left pane, click on Route Table
* click to check the RT
* click on routes tab and click on edit routes
* click on Add Routes
* Destination = Paste the network range of the old VPC (cidr)

Target = Peering connection and select the connection name

click on save routes

* Go to the former region
* on the left pane, Under Security, click on Network ACLs
* NACL stands as the Security Group for subnets
* click on create Network ACL

Name tag = pub-sub-NACL

VPC = select the VPC you created

click on create

* click to check the pub-sub-NACL
* click on the subnet association Tab and click on Edit subnet associations
* select the two public Subnets and click on Edit
* click on the outbound rules tab and click on Edit outbound rules
* click on add rule:

rule = 100

type = custom TCP rules

protocol = TCP(6)

port range = 0

destination = 0.0.0.0/0

allow/deny = allow

* click on add rule:

rule = 200

type = custom TCP rules

protocol = TCP(6)

port range = 0

destination = 0.0.0.0/0

allow/deny = deny

* click on the inbound rules tab and click on Edit inbound rules
* click on add rule:

rule = 100

type = custom TCP rules

protocol = TCP(6)

port range = 0

destination = 172.21.0.0/16

allow/deny = deny

* click on add rule:

rule = 200

type = custom TCP rules

protocol = TCP(6)

port range = 0

destination = 0.0.0.0/0

allow/deny = allow

CLEAN UP TO PREVENT AWS BILLINGS

* Delete the Load Balancer
* Delete the instances (Bastion Host & Web01)
* Delete the NAT Gateway and release the Elastic IP
* Delete the VPC peering connection
* and Delete the VPCs (everything will be deleted now)