

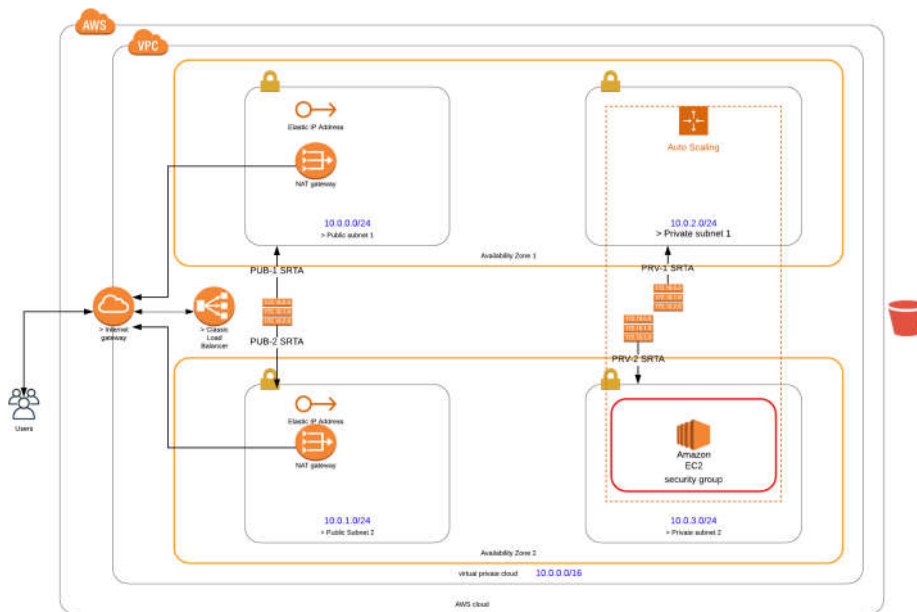
## 3. B. Architecture Diagram

- Architecture Diagram
  - Public vs Private Subnets
  - Internet Gateway
  - NAT (Network Address Translation)
  - Autoscaling Group
  - Load Balancing
  - Security Groups
  - Route Tables
    - S3 Bucket

### Architecture Diagram

Key components:

1. AWS Account
2. VPC (within a Region typically)
3. AZ - Availability Zones
  - AZ1
  - AZ2
4. Subnets:
  - Public (x2)
  - Private (x2)
5. IGW - Internet Gateway:
  - The IGW Internet Gateway
  - The IGW Attachment
6. NAT - Network Address Translation
  - Elastic IP (NatGW-EIP-1 & NatGW-EIP-2)
  - NAT Gateway (NatGW-1 & NatGW-2)
7. Public Subnet RouteTable:
  - The RouteTable
  - The Route
  - SRTA - SubnetRouteTableAssociation
    - Public Subnet-1
    - Public Subnet-2
8. Private Subnet-1 RouteTable:
  - The RouteTable
  - The Route
  - SRTA - SubnetRouteTableAssociation
9. Private Subnet-2 RouteTable:
  - The RouteTable
  - The Route
  - SRTA - SubnetRouteTableAssociation



## AWS Containers:

**AWS Cloud container**: contains resources within your AWS account

**AWS Region**: you'll be working based on a region most of the time. This is because your business would typically reside within a specific geographical region. However for a global company, multi-region deployment will be needed.

**Availability Zone**:

- think of this as a Data Centre, a physical building housing your computers and services. Obviously, located within the **region** you're working in.
- You need to design with High Availability requirements in mind. Therefore, use at least 2 AZs.
- You need to avoid SPOF (Single Point of Failure) in your diagram.
  - Design for HA: 2 or more of every resources.
  - Design for costs: only 1 AZ. (e.g. for test environment)
- **Virtual Private Cloud** and **subnet** follow the same rules as traditional networking.
- The main attribute of a VPC is to have a block of IP addresses a.k.a. address space or CIDR block: e.g. 10.0.0.0/16 give 65,000 IP addresses.

**Virtual Private Cloud (VPC)**:

- A virtual private cloud is a pool of networked cloud resources. *It can span more than one availability zone.*
- The equivalent of this would be a data center. However, thanks to availability zones, *VPCs can span more than one physical building.* This is an amazing feature that *protects against real world disasters* like electrical failures, fires and similar events.

**Subnets**:

- Create separation between resources
- Block or allow access to/from groups of resources
- Provide services to a specific set of resources

- a subset of the overall VPC network and only exists in a single AZ, unlike its parent network, the VPC.
- A subnet contains resources, and can be assigned access rights that apply to all resources within that subnet.

So, within 10.0.0.0/16 >> 65,000 addresses

can create subnets such as:

10.0.1.0/24 >> 255 addresses

10.0.2.0/24 >> 255 addresses

10.0.3.0/24 >> 255 addresses, and so on.

## AWS VPC and Subnets

### Classless Inter Domain Routing

Key points:

- VPCs provide you with private IP addresses for your networking resources
- Subnets are smaller subsets of your available IP address space
- The /00 at the end is the number of bits, from left to right that are fixed
- Subnets help with routing and services to specific groups of resources
- Create subnets and VPCs with future expansion in mind.

Notes - CloudFormation:

- When you create each subnet, you provide the VPC ID and IPv4 CIDR block for the subnet.
- After you create a subnet, you can't change its CIDR block.
- The size of the subnet's IPv4 CIDR block can be the same as a VPC's IPv4 CIDR block, or a subset of a VPC's IPv4 CIDR block.
- If you create more than one subnet in a VPC, the subnets' CIDR blocks must not overlap.
- The smallest IPv4 subnet (and VPC) you can create uses a /28 netmask (16 IPv4 addresses), and the largest uses a /16 netmask (65,536 IPv4 addresses).
- If you've associated an IPv6 CIDR block with your VPC, you can create a subnet with an IPv6 CIDR block that uses a /64 prefix length.

[Top](#)

## Public vs Private Subnets

The real goal here is to use the IP addresses in these subnets as our key for routing traffic. This is done via a routing table.

- I want this traffic to stay within my VPC, or
- this traffic to just go to this one subnet or to that subnet
- We use this as an element of security.
- We use routing rules/tables and security group to control access to subnets.

[Top](#)

## Internet Gateway

- Provides inbound and outbound traffic to your VPC
- An internet gateway allows external users access to communicate with parts of your VPC.
- If you create a private VPC for an application that is internal to your company, you will not need an internet gateway.
- If public access is not needed but instead only internal company access to the Cloud, you can use VPN connection or Direct Connect connection.
- Resource in private subnets sometime need to be able to access the 'outside world'. For example, downloading db server patches.
  - Also, if need to access S3, it will need to have access to it as S3 is actually a public service.
- So, even though we call it 'private', it still needs to have a way to have outbound internet access.

If I just created a VPC and I want to provide internet access to it, I should make sure to:

1. Create an IGW
2. Attach the IGW to VPC
3. Create a route to the IGW and associate it with your subnet(s)

All of the above steps are required.

### Software Defined Networking

What we have created here it's called Software Defined Networking. That is, using APIs and already-existing physical infrastructure to create our own networking layer on top, with our own privacy rules, our own routing and our own Private IP Space.

### VPN or Virtual Private Network

It is a type of encrypted connectivity that You can setup between your on-premise data center and your Virtual Private Cloud. This allows access in and out of your AWS VPC in a secure manner, across the internet and using internal, Private IP addresses.

### DirectConnect

It is a DirectConnect is a physical data line that you can purchase directly from AWS or through a telecommunication service provider to access your AWS Cloud without moving your data traffic across the public internet.

### Internet Gateway

## NAT (Network Address Translation)

- Provides outbound internet access to resources in Private subnets
- It "Translates" incoming public traffic into private traffic
- It needs public access itself, remember to place it in a PUBLIC subnet.

When using CloudFormation, 2 steps involved:

1. Create NAT
2. NAT attachment to EIP (Elastic IP)
  - If you need a fix IP that will never change (e.g. when there's maintenance on the system, a new dynamically assigned public-IP may be provided to your resource, which may impact the working of your resources), then make sure to do NAT attachment of the EIP.
  - Most of the time, you want to have 'stability' and have a fix public-IP assigned to your resource.

CloudFormation:

- If you need a specific logic for your CloudFormation to wait for the creation of another resource on the same script, use the property: `DependsOn`
- This is a good hint for CloudFormation on what to build first in your script and on which order to build resources.
- In NAT's case, we specify `DependsOn` on the `InternetGatewayAttachment`

[Top](#)

## Autoscaling Group

It is a coherent group of Virtual Machines (EC2 instances) that allows running the exact number of VMs that are required to meet the demand/specification.

The autoscaling group can automatically start or stop the servers (EC2 instances) according to the amount of incoming traffic. This behavior of the autoscaling group helps in two ways:

1. The consumer pays for the only duration of the servers when they were active.
2. The consumer doesn't have to worry about horizontal scaling of servers for a sudden peak in incoming traffic.

### Best Practice

- It is recommended that an autoscaling group spans more than one availability zone, for reliability.
- If we set the autoscaling group to run one resource, it will run that one resource in one of the availability zones.
- If there is a failure of that resource, the autoscaling group will shut it down in that availability zone and start that same resource in the other availability zone.


[Autoscaling Group](#)

[Top](#)

## Load Balancing

- It's a service designed to *distribute work requests meant for a target group*
- -S

- As requests come in, the Load Balancer will spread the requests evenly across its target group

 fdb28067.png

Example:

- A service to process an image: to resize, to generate metadata, to generate thumbnail
- A load balancer takes incoming traffic and distributes it to two or more resources. For example, it can take inbound user requests to access your website, and it can distribute the requests evenly among two or more servers.
- Without a load balancer, having public-facing servers in more than one AZ would mean that users would have to use a different URL to reach each of the AZs. This can be impractical compared to just a single URL.
- Good practice - Assume we have a set of web-servers in private subnet(s). Then, we must have a Load Balancer that would access our web-servers. These web-servers, in turn, would access the backend database.

### AWS - Elastic Load Balancing

We recommend you to read about three types of load balancing offered by AWS at different layers of networking protocol:

- Classic load balancer
- Application load balancer
- Network load balancer
- Load Balancer can not only be associated with servers, but also with a scaling group. When the servers within the group are scaling up, the LB will know where to direct the requests to.
- Another benefit of an LB is health-check. The LB will check your servers to make sure they're operational. If they're not, they will be taken out of service completely and traffic will be routed to the healthy instances.
- Any servers you deploy in the public sub-nets, they do have visibility into the private sub-nets because they're all in the same Cloud, in the same VPC container. So, in this container, its inbound and outbound traffic is unrestricted for everything within there because obviously it is trusted traffic and they'll be able to talk to one another without any issues.

Note:

- When specifying YAML, must include at least 2 AZs for LB entry.

[Top](#)

## Security Groups

- Security groups is a way to manage traffic at the server level (the resource level). Security Groups aren't for managing higher-level groups such as subnets, VPC, or user accounts.
- The same security group can be assigned to multiple resources that require the same security access settings defined by that security group.
- The default behaviour is all ports are locked down unless specifically opened.
- Once you create a security group, you get to assign it to any resource that you want e.g. a LB, an EC2, etc. You can use that same security group over and over.
- Security Group is a collection of networking rules for `inbound` and `outbound` traffic.
- A security group can be scoped to a single IP address.
  - You can be as specific as 1 IP address when giving access to yourself (for example) or as broad as the entire network (0.0.0.0/0)

[Security Groups](#)

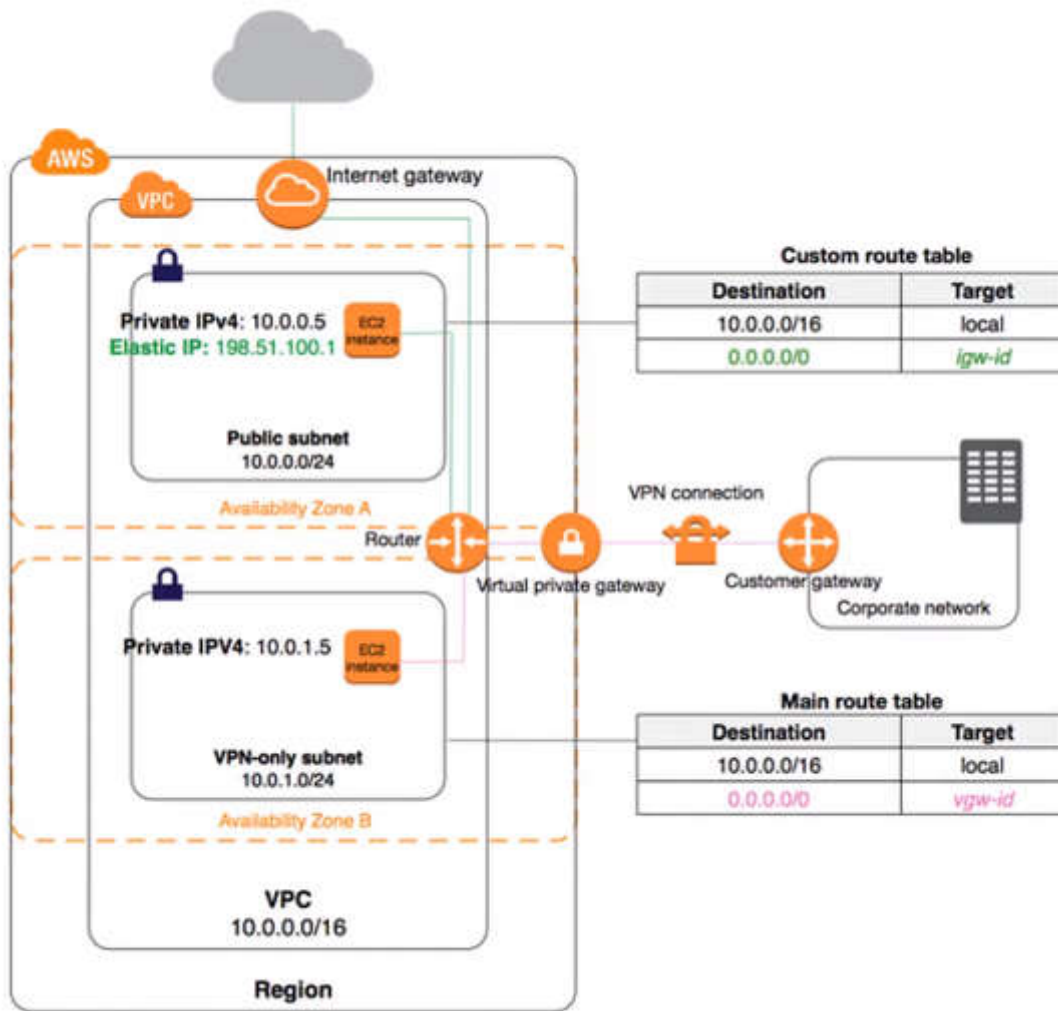
[Top](#)

## Route Tables

Routing in the Cloud is a very important component in your infrastructure. By using `route-tables` you can regulate network traffic to and from your network.

- It's a set of entries or rules *associated with one or more of your subnets* inside your VPC
- These rules *allow or deny traffic to/from the address ranges* that you specify
- Rules can be as open as the entire world or restricted to a single IP address.

Example:



- The VPC is connected to on-premise network as well as the Internet
- There is a `public-subnet` and a `private-subnet`
- Private subnet rule: (Main Routing Table)
  - `all-traffic` shall go to the `virtual-private-gateway` (vgw) and will send traffic through `vpn` connection to `corporate-network`
  - Traffic with destination to any IP address within range of `10.0.0.0/16`, which is the address range of the VPC, and the target is `local`.
    - This means, all servers within the Cloud (`VPC`) can talk to one another without restrictions.
    - However, when the `private-subnet` attempts to access anything to the outside world it can only do so through the `corporate-network`. It's not visible from the outside, it can only be accessed to and from the `corporate-network`.
- Public subnet rule: (Custom Routing Table)
  - It has the same target for local routes. It has access to and from as long as it's within the same VPC.
  - However, any other address that are not within this VPC will be routed to the `IGW`
  - Any machines that you put inside this public subnet will be visible to the outside world and will have Internet access.

`private-subnet` route table can be configured as follows:

- requirement: we want to make the `private-subnets` be able to reach the Internet.
- solution: we associate the 2 `private-subnets` to a routing table.



- Routing rules:
  - The default: all `inbound` and `outbound` traffic is allowed unrestricted provided it is within the isolated VPC Cloud.
  - If traffics are going to the Internet, we're going to allow outbound traffic only for the private subnets. So, we're going to route traffic from the `private-subnets`, from any server in there, going through the NATs.

`public-subnet` route table is different from that of the `private-subnet`'s one.:

- Rule: for any traffic send it to the IGW (Internet Gateway), there's no restrictions whether `inbound` or `outbound`

References:

[Route Table](#)

[Top](#)

## S3 Bucket

- An S3 bucket is a *public service* for users to upload or download files.
- Place the S3 service outside of your VPC.

Sanitized Logs: all sensitive information removed, e.g. credit card numbers, passwords, etc.

Examples:

[AWS Reference Architecture](#)

[WordPress Architecture](#)

[Top](#)