This article on AWS VPC covers the following topics:

* Amazon VPC, it variations, and components
* Public, private, and elastic IP addresses
* Public and private subnets
* Internet gateways
* Route tables and when are they used
* NAT gateways
* Security groups and their importance
* Network ACLs and how they are used in Amazon VPC
* Amazon VPC best practices
* Costs associated with running a VPC in the Amazon Cloud

Let us now begin by understanding more on AWS VPC and subnets.

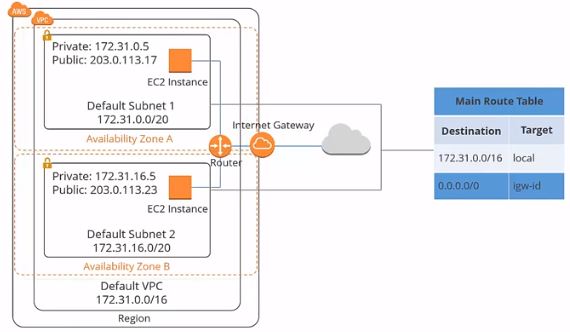
Amazon VPC and Subnets

Amazon VPC enables you to connect your on-premises resources to AWS infrastructure through a virtual private network. This virtual network closely resembles a traditional network that you'd operate in your data center but enables you to leverage the scalable infrastructure in AWS.

Each VPC that you create is logically isolated from other virtual networks in the AWS cloud and is fully customizable. You can select the IP address range, create subnets, configure root tables, set up network gateways, define security settings using security groups, and network access control lists.

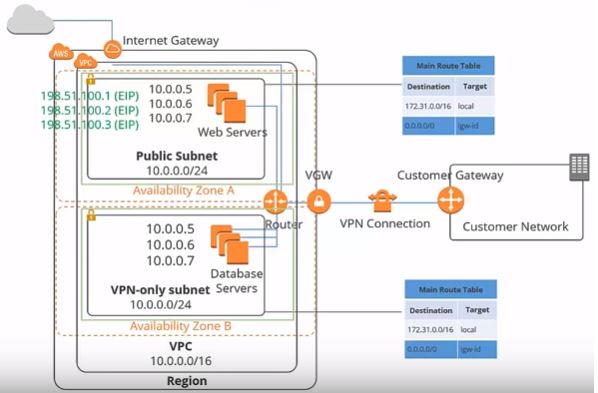
Default Amazon VPC

Each Amazon account comes with a default VPC that is pre-configured for you to start using immediately. A VPC can span multiple availability zones in a region. This is the diagram of a default VPC:



In the first section, there is a default Amazon VPC. The CIDR block for the default VPC is always a 16 subnet mask; in this example, it's 172.31.0.0/16. It means this VPC can provide up to 65,536 IP addresses.

Custom Amazon VPC

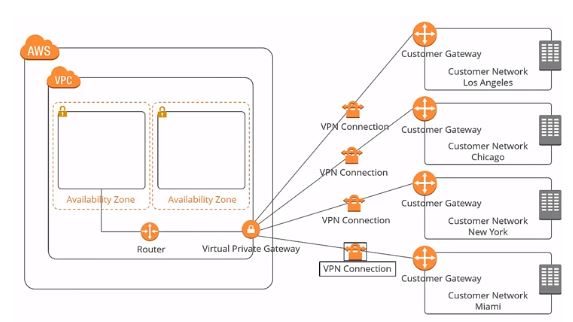


The default VPC is suitable for launching new instances when you're testing AWS, but creating a custom VPC allows you to:

* Make things more secure
* Customize your virtual network, as you can define your own our IP address range
* Create your subnets that are both private and public
* Tighten security settings

Hardware VPN Access

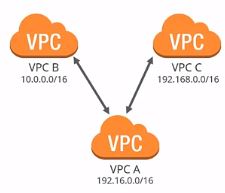
By default, instances that you launch into an Amazon VPC can't communicate with your network. You can connect your VPCs to your existing data center using hardware VPN access. By doing so, you can effectively extend your data center into the cloud and create a hybrid environment. To do this, you will need to set up a virtual private gateway.



There is a VPN concentrator on the Amazon side of the VPN connection. For your data center, you need a customer gateway, which is either a physical device or a software application that sits on the customer’s side of the VPN connection. When you create a VPN connection, a VPN tunnel comes up when traffic is generated from the customer's side of the connection.

VPC Peering

A peering connection can be made between your own VPCs or with a VPC in another AWS account, as long as it is in the same region.



If you have instances in VPC A, they wouldn't be able to communicate with instances in VPC B or C unless you set up a peering connection. Peering is a one-to-one relationship; a VPC can have multiple peering connections to other VPCs, but transitive peering is not supported. In other words, VPC A can connect to B and C in the above diagram, but C cannot communicate with B unless directly paired.

Additionally, VPCs with overlapping CIDRs cannot be paired. In the diagram, all VPCs have different IP ranges. If they have the same IP ranges, they wouldn't be able to pair.

Default VPC Deletion

In the event that the default VPC gets deleted, it is advised to reach out to AWS support for restoration. Therefore, you’ll only want to delete the default VPC only if you have a good reason.

After going through what AWS VPC is, let us next learn the IP addresses.

Private, Public, and Elastic IP Addresses

Private IP addresses

IP addresses not reachable over the internet are defined as private. Private IPs enable communication between instances in the same network. When you launch a new instance, a private IP address is assigned, and an internal DNS hostname allocated to resolves to the private IP address of the instance. If you want to connect to this from the internet, it will not work. You would need a public IP address for that.

Public IP addresses

Public IP addresses are used for communication between other instances on the internet and yours. Each instance with a public IP address is assigned an external DNS hostname too. Public IP addresses linked to your instances are from Amazon's list of public IPs. On stopping or terminating your instance, the public IP address gets released, and a new one is linked to the instance when it restarts. For retention of this public IP address even after stoppage or termination, an elastic IP address needs to be used.

Elastic IP Addresses

Elastic IP addresses are static or persistent public IPs that come with your account. If any of your software or instances fail, they can be remapped to another instance quickly with the elastic IP address. An elastic IP address remains in your account until you choose to release it. A charge is associated with an Elastic IP address if it is in your account, but not allocated to an instance.

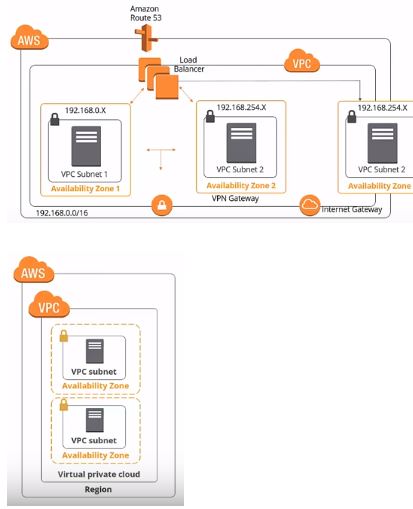
As a part of our AWS VPC tutorial, let us learn about subnets.

Subnet

AWS defines a subnet as a range of IP addresses in your VPC. You can launch AWS resources into a selected subnet. A public subnet can be used for resources connected to the internet and a private subnet for resources not connected to the internet.

The netmask for a default subnet in your VPC is always 20, which provides up to 4,096 addresses per subnet, with few of them reserved for AWS use. The VPC can span multiple availability zones, but the subnet is always mapped to a single availability zone.

The following is a basic diagram of a subnet:



There is a virtual private cloud consisting of availability zones. A subnet is created inside each availability zone, and you cannot launch any instances unless there are subnets in your VPC.

Public and Private Subnets

There are two types of subnets: public and private. A public subnet is used for resources that must be connected to the internet; web servers are an example. A public subnet is made public because the main route table sends the subnets traffic that is destined for the internet to the internet gateway.

Private subnets are for resources that don't need an internet connection, or that you want to protect from the internet; database instances are an example.

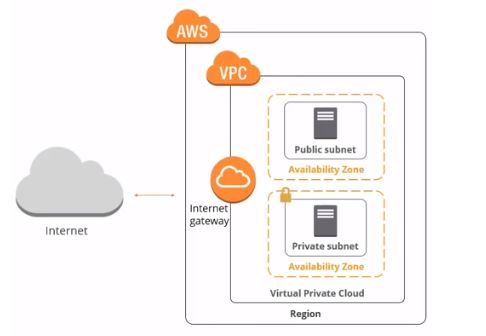
Let us extend our AWS VPC tutorial by looking into networking.

Networking

Internet Gateway

An internet gateway is a redundant, horizontally scaled, and is a highly available VPC component. It enables communication between instances in your VPC and the internet. Therefore, it imposes no availability risks or bandwidth constraints on your network traffic.

To give your VPC the ability to connect to the internet, you need to attach an internet gateway. Only one internet gateway can be attached per VPC. Attaching an internet gateway is the first stage in permitting internet access to instances in your VPC.



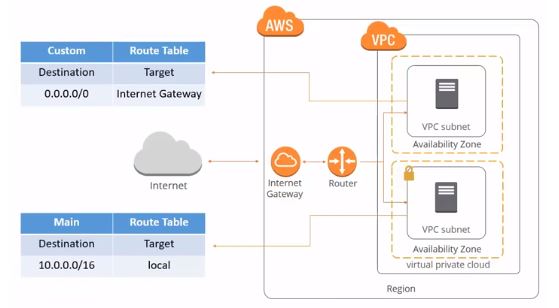
In this diagram, we've added the internet gateway that is providing the connection to the internet to your VPC. For an EC2 instance to be internet-connected, you have to adhere to the following rules:

1. Attach an Internet gateway to your VPC
2. Ensure that your instances have either a public IP address or an elastic IP address
3. Point your subnet’s route table to the internet gateway
4. Make sure that your security group and network access control rules allow relevant traffic to flow in and out of your instance

Route Table

Amazon defines a route table as a set of rules, called routes, which are used to determine where network traffic is directed.

Each subnet has to be linked to a route table, and a subnet can only be linked to one route table. On the other hand, one route table can have associations with multiple subnets. Every VPC has a default route table, and it is a good practice to leave it in its original state and create a new route table to customize the network traffic routes associated with your VPC. The route table diagram is as shown:



In this example, we've added two route tables: the main route table and the custom route table. The new route table or the custom route table informs the internet gateway to direct

internet traffic to the public subnet. However, the private subnet is still associated with the default route table, the main route table that does not allow internet traffic. All traffic inside the private subnet remains local.

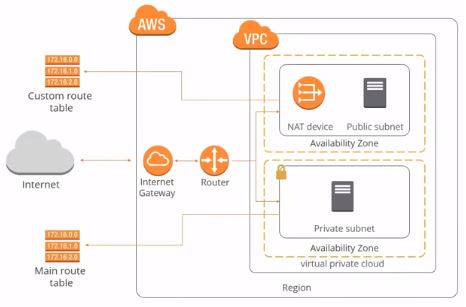
NAT Devices

A Network Address Translation (NAT) device can be used to enable instances in a private subnet to connect to the internet or the AWS services, but this prevents the internet from initiating connections with the instances in a private subnet.

As mentioned earlier, public and private subnets protect your assets from being directly connected to the internet. For example, your web server would sit in the public subnet and database in the private subnet, which has no internet connectivity. However, your private subnet database instance might still need internet access or the ability to connect to other AWS resources. You can use a NAT device to do so.

The NAT device directs traffic from your private subnet to either the internet or other AWS services. It then sends the response back to your instances. When traffic is directed to the internet, the source IP address of your instance is replaced with the NAT device address, and when the internet traffic returns, the NAT device translates the address to your instance’s private IP address.

NAT device diagram:



In the diagram, you can see a NAT device is added to the public subnet to get internet connectivity.

NAT Gateway vs. NAT Device

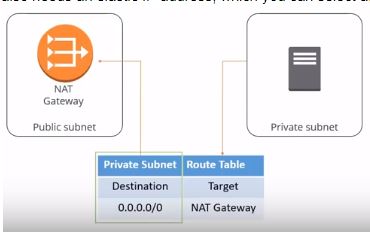
AWS provides two kinds of NAT devices:

* NAT gateway
* NAT instance

AWS recommends the NAT gateway because it is a managed service that provides better bandwidth and availability compared to NAT instances. Every NAT gateway is created in a specific availability zone and with redundancy in that zone. A NAT Amazon Machine Image (AMI) is used to launch a NAT instance, and it subsequently runs as an instance in your VPC.

NAT Gateway

A NAT gateway must be launched in a public subnet because it needs internet connectivity. It also requires an elastic IP address, which you can select at the time of launch.



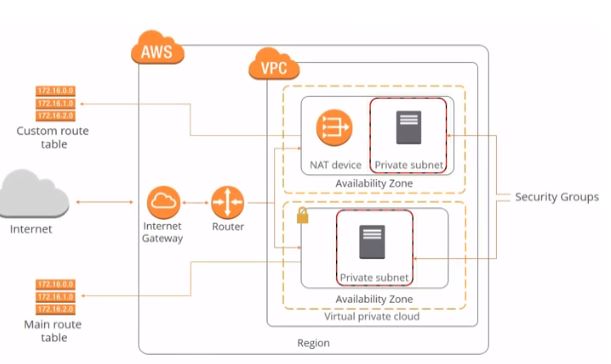
Once created, you need to update the route table associated with your private subnet to point internet-bound traffic to the NAT gateway. This way, the instances in your private subnet can communicate with the internet.

As a part of our learning about the AWS VPS, security groups and network ACLs takes an important role. So, let's dive in.

Security Groups and Network ACLs

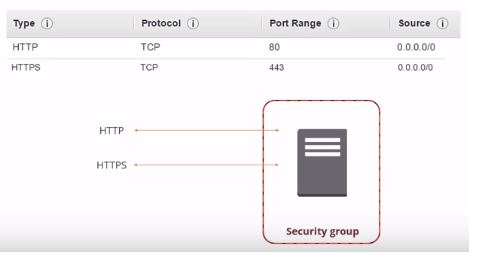
Amazon defines a security group as a virtual firewall that controls the traffic for one or more instances. Rules are added to each security group, which allows traffic to or from its associated instances. Basically, a security group controls inbound and outbound traffic for one or more EC2 instances. It can be found on both the EC2 and VPC dashboards in the AWS web management console.

Security group diagram:



Security Groups for Web Servers

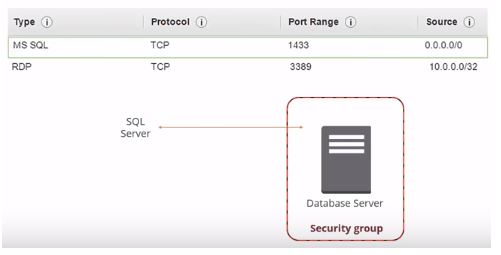
A web server needs HTTP and HTTPS traffic at the least to access it. The following is an example of the security group table:



Here, we allow HTTP and HTTPS, the ports associated with them, and the sources from the internet. All traffic is allowed to those ports, so any other traffic that arrives on different ports would be unable to reach the security group and the instances inside.

Security Groups for Database Servers

If we consider a SQL server database, then you need to open the SQL server port to access it.



We've allowed the source to come from the internet. Because it's a Windows machine, you may need RDP access to log on and do some administration. We've also added RDP access to the security group. You could leave it open to the internet, but that would mean anyone could try and hack their way into your box. In this example, we've added a source IP address of 10.0.0.0, so the only IP ranges from that address can RDP to the instance.

Security Groups Rules

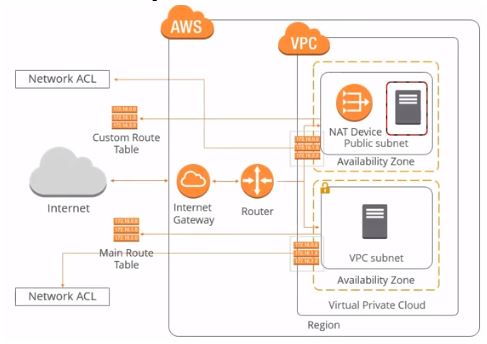
There are a few rules associated with security groups, such as:

* Security groups allow all outbound traffic by default. If you want to tighten your security, this can be done in a similar way as you define the inbound traffic
* Security group rules are always permissive. You can't create rules that deny access
* Security groups are stateful. If a request is sent from your instance, the response traffic for that request is allowed to flow in, regardless of the inbound security group rules
* The rules of a security group can be modified at any time, and apply immediately

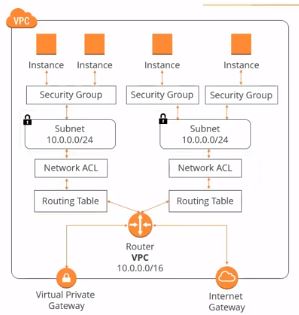
Network ACL

The Network Access Control List (ACL) is an optional security layer for your VPC. It acts as a firewall for controlling traffic flow o and from one or more subnets. Network ACLs can be set up with rules similar to your security groups

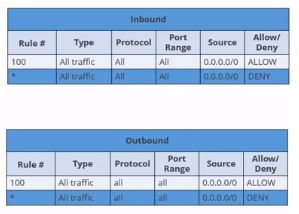
Here is the network diagram:



You can find the Network ACLs located somewhere between the root tables and the subnets. Here is a simplified diagram:



You can see an example of the default network ACL, which is configured to allow all traffic to flow in and out of the subnet to which it is associated.



Each network ACL includes a rule an \* (asterisk) as the rule number. The rule makes sure that if a packet is identified as not matching any of the other numbered rules, traffic is denied. You can't modify or remove this rule.

For traffic coming on the inbound:

* Rule 100 would allow traffic from all sources
* Rule \* would deny traffic from all sources

Network ACL Rules

* Every subnet in your VPC must be associated with an ACL, failing which the subnet gets automatically associated with your default ACL.
* One subnet can only be linked with one ACL. On the other hand, an ACL can be linked to multiple subnets.
* An ACL has a list of numbered rules that are evaluated in order, starting with the lowest. As soon as a rule matches, traffic is supplied regardless of any higher-numbered rules that may contradict it. AWS recommends incrementing your rules by a factor of 100. This allows for plenty of room to implement new rules at a later date.
* Unlike security groups, ACLs are stateless; responses to allow inbound traffic is subject to the rules for outbound traffic.

Let us next look into the AWS VPC best practices.

Amazon VPC Best Practices and Costs

Public and private subnets

You should use private subnets to secure resources that don't need to be available to the internet, such as database services. It enables the flexibility to launch a service in the subnets.

Provide NAT to private subnets

To provide secure internet access to instances that reside in your private subnets, you should leverage a NAT device.

Use NAT gateways

When using NAT devices, you should use the NAT gateway over NAT instances because they are managed services and require less administration. It also provides secure internet access to your private subnets.

CIDR blocks

You should choose CIDR blocks carefully; Amazon VPC can contain anywhere from 16 to 65,536 IP addresses. You can select your CIDR block according to the number of instances needed.

Create different VPCs for different environments

You should also create a separate Amazon VPC for development, staging, and test and production environments. Another option is to create an Amazon VPC with separate subnets with a subnet for each production, development, staging, and tests.

Understand Amazon VPC limits

There are various limitations to the VPC components. For example, you're allowed:

* Five VPCs per region
* 200 subnets per VPC
* 200 route tables per VPC
* 500 security groups per VPC
* 50 inbound and outbound rules per VPC

However, some of these limits can be increased by submitting a ticket to AWS support.

Use security groups and network ACLs

You should use security groups and network ACLs to secure the traffic coming in and out of your VPC. Amazon advises using security groups for whitelisting traffic and network ACLs for blacklisting traffic.

Tier security groups

Amazon recommends tiering your security groups. You should create different security groups for different tiers of your infrastructure architecture inside VPC. If you have web server tiers and database tiers, you should create different security groups for each of them. Creating tier wise security groups increases the infrastructure security inside the Amazon VPC. So, if you launch all your web servers in the web server security group, that means they'll automatically have HTTP and HTTPS open. Conversely, the database security group will have SQL server ports already open.

Standardize security group naming conventions

Following a security group, the naming convention allows Amazon VPC operation and management for large scale deployments to become much easier.

Span Amazon VPC

Make sure to span your Amazon VPC across multiple subnets across multiple availability zones within a region. This helps in architecting high availability inside your VPC.

Amazon VPC costs

If you opt to create a hardware VPN connection associated with your VPC using Virtual Private Gateway, you will have to pay for each VPN connection hour that your VPN connection is provisioned and available. Each partial VPN connection hour consumed is billed as a full hour. You'll also incur standard AWS data transfer charges for all data transferred via the VPN connection.

If you create a NAT gateway in your VPC, Charges are levied for each NAT gateway hour that your NAT gateway is provisioned and available for. Data processing charges apply for each gigabyte processed through a NAT gateway. Each partial NAT gateway hour consumed is billed as a full hour.