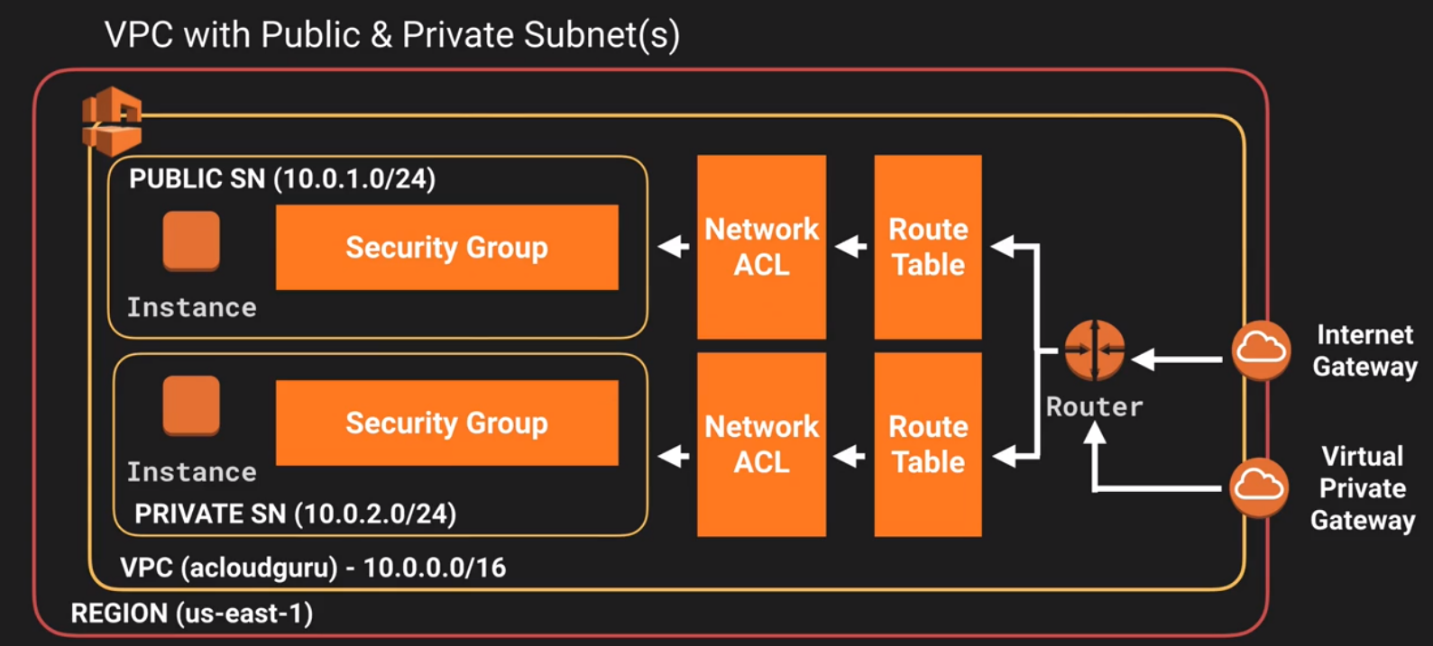
**[VPC (Virtual Private Cloud):](https://aws.amazon.com/vpc/)**

**Amazon Virtual Private Cloud (VPC)** allows the users to use AWS resources in a virtual network. The users can customize their virtual networking environment as they like, such as selecting own IP address range, creating subnets, and configuring route tables and network gateways.

* Virtual data center in the cloud
* Allowed up to 5 VPCs in each AWS region by default
* All subnets in default VPC have an Internet gateway attached
* Multiple IGW's can be created, but only a single IGW can be attached to a VPC
* Each EC2 instance has both a public and private IP address
* If you delete the default VPC, the only way to get it back is to submit a support ticket
* By default when you create a VPC, a default main routing table automatically gets created as well.
* Subnets are always mapped to a single AZ's
* Subnets cannot be mapped to multiple AZ's
* /16 is the largest CIDR block available when provisioning an IP space for a VPC
* Amazon uses 3 of the available IP addresses in a newly created subnet



* + x.x.x.0 - Always subnet network address and is never usable
  + x.x.x.1 - Reserved by AWS for the VPC router
  + x.x.x.2 - Reserved by AWS for subnet DNS
  + x.x.x.3 - Reserved by AWS for future use
  + x.x.x.255 - Always subnet broadcast address and is never usable.
* 169.254.169.253 - Amazon DNS
* By default all traffic between subnets is allowed
* By default not all subnets have access to the Internet. Either an Internet Gateway or NAT gateway is required for private subnets
* You can only have 1 Internet gateway per VPC
* A security group can stretch across different AZ's
* You can also create Hardware Virtual Private Network (VPN) connection between your corporate data center and your VPC and leverage the AWS cloud as an extension of your corporate data center
* Network Address Translation (NAT) Instances:
  + When creating a NAT instance, disable Source/Destination checks on the instance or you could encounter issues
  + NAT instances must be in a public subnet
  + There must be a route out of the private subnet to the NAT instance in order for it to work
  + The amount of traffic that NAT instances support depend on the size of the NAT instance
  + If you are experiencing any sort of bottleneck issues with a NAT instance, then increase the instance size
  + HA can be achieved by using Auto-scaling groups, or multiple subnets in different AZ's with a scripted fail-over procedure
  + NAT instances are always behind a security group
* Network Address Translation (NAT) Gateway:
  + NAT Gateways scale automatically up to 10Gbps
  + There is no need to patch NAT gateways as the AMI is handled by AWS
  + NAT gateways are automatically assigned a public IP address
  + When a new NAT gateway has been created, remember to update your route table
  + No need to assign a security group, NAT gateways are not associated with security groups
  + Preferred in the Enterprise
  + No need to disable Source/Destination checks
* Network Access Control Lists (NACLS):
  + Numbered list of rules that are evaluated in order starting at the lowest numbered rule first to determine what traffic is allowed in or out depending on what subnet is associated with the rule
  + The highest rule number is 32766
  + Start with rules starting at 100 so you can insert rules if needed
  + Default NACL will allow ALL traffic in and out by default
  + You must assign a NACL to each subnet, if a subnet is not associated with a NACL, it will allow no traffic in or out
  + NACL rules are stateless, established in does not create outbound rule automatically
  + You can only assign a single NACL to a single subnet
* VPC Peering:
  + Connection between two VPCs that enables you to route traffic between them using private IP addresses via a direct network route
  + Instances in either VPC can communicate with each other as if they are within the same network
  + You can create VPC peering connections between your own VPCs or with a VPC in another account within a SINGLE REGION
  + AWS uses existing infrastructure of a VPC to create a VPC peering connection. It is not a gateway nor a VPN, and does not rely on separate hardware
  + There is NO single point of failure for communication nor any bandwidth bottleneck
  + There is no transitive peering between VPC peers (Can't go through 1 VPC to get to another)
  + Hub and spoke configuration model (1 to 1)
  + Be mindful of IPs in each VPC, if multiple VPCs have the same IP blocks, they will not be able to communicate

**Layers of security**

For multiple layers of security, it’s recommended you use a VPC in addition to security groups and NACLs (Network Access Control Lists).

Security groups (first layer of defense) exist at the instance level.

NACLs (second layer of defense) exist at the subnet level.

It’s possible to implement a private cloud (i.e. a corporate data center) using VPCs.

**CIDR - Classless Inter-domain Routing**

**The first four IP addresses and the last IP address in each subnet CIDR block are not available for you to use, and cannot be assigned to an instance.**

i.e.

* 10.0.0.0: Network address.
* 10.0.0.1: Reserved by AWS for the VPC router.
* 10.0.0.2: Reserved by AWS for DNS.
* 10.0.0.3: Reserved by AWS for future use.
* 10.0.0.255: Network broadcast address. We do not support broadcast in a VPC, therefore we reserve this address.

Network masks:

* /16 - supports up to 65,536 IP addresses. Best for large networks.
* /24 - supports up to 256 IP addresses. Best for smaller networks.
* /27 - supports up to 32 IP addresses
* /28 - supports up to 16 IP addresses
* /32 - an absolute ip address - matches exactly one

t’s possible to split a CIDR block into two subnets:

* one subnet can use CIDR block 10.0.0.0/25 (for addresses 10.0.0.0 - 10.0.0.127)
* and then the other subnet can use the CIDR block 10.0.0.128/ 25 (for addresses 10.0.0.128 - 10.0.0.255)

The allowed CIDR block size in a VPC is between a /16 and /28 netmask.

To enable ping, you need to allow ICMP traffic.

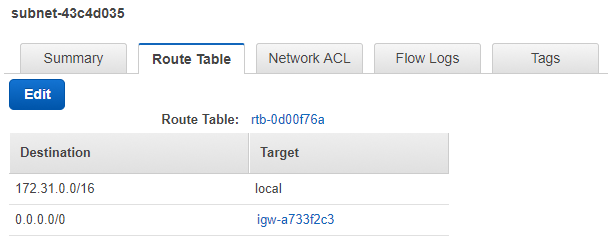
In order to ensure provisioned EC2 instances have a public IP address, enable “Auto-Assign Public IP” for the subnet.

### Subnets and route tables

### Route table – A route table contains a set of rules, called routes that are used to determine where network traffic is directed.

In order for instances in your subnet to be able to access the internet, ensure that there is appropriate an entry on the route table such as **0.0.0.0/0 -> Internet Gateway**

i.e.



## **Security groups vs NACLs**

|  |  |
| --- | --- |
| **Security Group** | **NACL** |
| Supports only Allow rules | Supports Allow and Deny rules |
| Stateful | Stateless |
| All rules are evaluated before allowing traffic | Evaluates rules in numerical order. As soon as a rule is found allowing the traffic, the traffic is allowed. |
| Rules apply to specific instances | Rules apply to ALL instances in the subnet |
| Unable to block IP addresses | Able to block IP addresses |
|  |  |
|  |  |

A NACL can be applied to multiple subnets. If you attempt to assign a NACL to a subnet which already has a NACL, the new NACL will replace the existing NACL for the subnet.

A subnet can have only one NACL assigned to it

Types of NACL:

* Default - allows all traffic by default
* Custom - denies all traffic by default

In order to allow outgoing traffic, enable outgoing and incoming ephemeral ports (1024-65535)

## **NAT gateways and NAT instances**

NAT is used for traffic routing

It’s best practice to always enable HTTP and HTTPs traffic.

Must be provisioned into a **public subnet**, and it **must be part of the private subnet’s route table** in order for your instances in the private subnet to communicate with the outside internet.

Instances within a private subnet cannot communicate with the outside internet by default. In order for your instances within the private subnet to communicate with the internet (i.e. to run “yum update”), you’ll need to add 0.0.0.0/0 with the target pointing to your NAT gateway/instance to the route table in the private subnet

|  |  |
| --- | --- |
| **NAT instance** | **NAT Gateway** |
| Custom EC2 instances provisioned via custom community AMIs | Provisioned and managed by AWS |
| The amount of traffic handled by a NAT instance depends on it ‘size – bigger instances handle more traffic. | Managed by AWS and scale automatically up to , and are preferred over NAT instances which are custom community AMIs. |
| NAT instances use security groups | . |
| Need to disable source/dest checks disabled via the Actions->Networking menu. | No need to disable source/dest check, no security group is needed; it is all AWS managed, rather than a custom EC2 instance which needs to be  Maintained. |
| Must be in the public subnet and must have a public ip address. | . |
| Performance depends on the instance size/td> | Scale automatically up to 10GBPS |
|  |  |

### NAT Gateways

When creating a NAT gateway you must specify the public subnet in which it should reside, and also specify the Elastic IP (EIP) address to associate with it.

After creating the NAT gateway, you then need to update the route tables associated with one or more private subnets to point internet traffic to the NAT gateway.

If the NAT gateway gets created with a failed status, the NAT gateway will be visible for short period of time (usually an hour), then will be automatically deleted.

You cannot route traffic to a NAT gateway through a VPC peering connection, a VPN connection, or AWS Direct Connect. A NAT gateway cannot be used by resources on the other side of these connections.

### Source/dest checks (applies to NAT instances only)

Source/dest checks are there to ensure that the EC2 instance by default must be the source or destination of any traffic it sends or received.

We want to route any outgoing traffic through our NAT instance, and route any traffic coming back in through our NAT instance. Because of this, we need to disable source/dest checks.

## **Stateful vs Stateless**

Security groups are **stateful**. This means that if you add an incoming HTTP rule, there will automatically be a corresponding outgoing one.

Subnet ACLs are **stateless** which means that if you add an incoming HTTP rule, you’ll need to add an outgoing one too, otherwise HTTP traffic won’t be able to get back out of your subnet.

## **Default VPC**

AWS provids a default VPC. This is intended to make it user friendly to deploy and test EC2 instances on a new account. All default VPCs have routes out to the internet.

If you delete the default VPC, you’ll need to raise an Amazon support ticket to get it back.