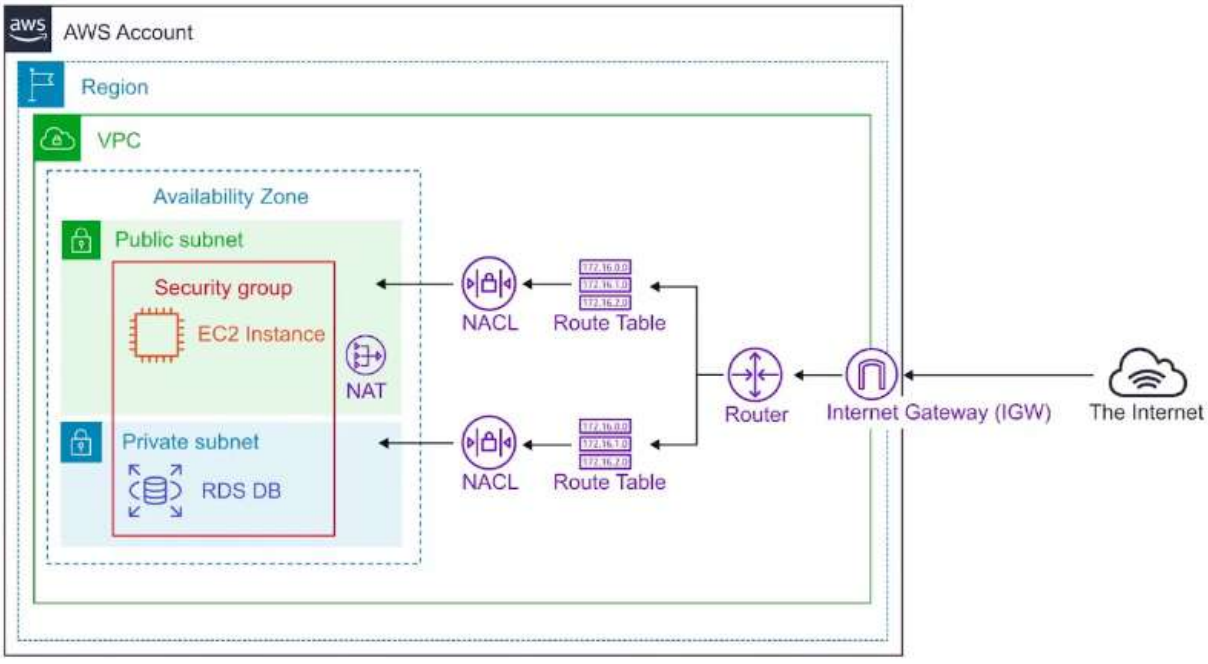
Think of a AWS VPC as your own **personal data centre.**

Gives you complete control over your virtual networking environment



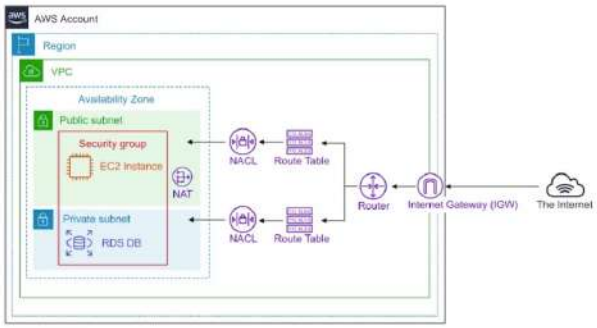
AWS Certified Solutions Architect - Note (2)

Virtual Private Cloud (VPC)

Provision a **logically isolated section** of the AWS Cloud where you can launch AWS resources in a **virtual network** that you define.

Think of a AWS VPC as your own **personal data centre.**

Gives you complete control over your virtual networking environment



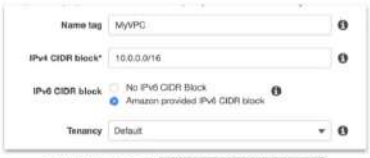
Core Components

Combining these components and services is what makes up your VPC.

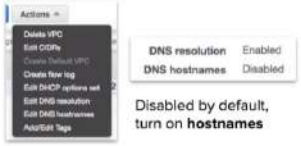
- Internet Gateway (IGW)
- Virtual Private Gateway (VPN Gateway)
- Routing Tables
- Network Access Control Lists (NACLs) - Stateless
- Security Groups (SG) Stateful
- Public Subnets
- Private Subnets
- Nat Gateway
- Customer Gateway
- VPC Endpoints
- VPC Peering

Key Features

- VPCs are **Region Specific** they do not span regions
- You can create upto **5 VPC** per region.
- Every region comes with a default VPC
- You can have **200 subnets** per VPC
- You can use **IPv4 Cidr Block** and in addition to a **IPv6 Cidr Blocks** (the address of the VPC)
- **Cost nothing:** VPC's, Route Tables, Nacls, Internet Gateways, Security Groups and Subnets, VPC Peering
- **Some things cost money:** eg. NAT Gateway, VPC Endpoints, VPN Gateway, Customer Gateway
- **DNS hostnames** (should your instance have domain name addresses)



IPv6 Cidr Block 2600:1f16:9e0:8d00::/56



Disabled by default, turn on **hostnames**

Default VPC

AWS has a default VPC in every region so you can **immediately** deploy instances.

- Create a VPC with a size /16 IPv4 CIDR block (172.31.0.0/16).
- Create a size /20 **default subnet in each Availability Zone**.
- Create an **Internet Gateway** and connect it to your default VPC.
- Create a **default security group** and associate it with your default VPC.
- Create a **default network access control list (NACL)** and associate it with your default VPC.
- Associate the **default DHCP** options set for your AWS account with your default VPC.
- *When you create a VPC, it automatically has a main route table

0.0.0.0/0

0.0.0.0/0 is also know as **default**

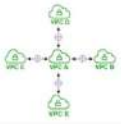
It represents **all possible IP addresses**

When we specify **0.0.0.0/0** in our route table for IGW we are allow internet access
When we specific **0.0.0.0/0** in our security groups inbound rules we are allowing all traffic from the internet access our public resources `

When you see **0.0.0.0/0**, just *think* of giving access from anywhere or the internet.

VPC Peering

VPC Peering allows you to connect one VPC with another over a **direct network route** using **private IP addresses.**



- Instances on peered VPCs **behave** just like they are on the **same network**
- Connect VPCs across **same** or **different AWS accounts** and **regions**
- Peering uses a **Star Configuration: 1 Central VPC - 4 other VPCs**
- **No Transitive Peering** (peering must take place directly between VPCs)
  - Needs a one to one connect to immediate VPC
- **No Overlapping CIDR Blocks**

Route Tables

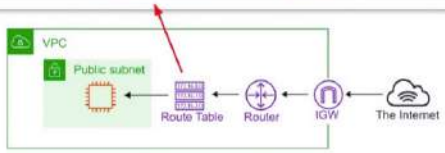
Route tables are used to determine where **network traffic is directed**

Each **subnet** in your VPC **must be associated** with a route table

Each record is called a "route"

A subnet can only be associated **with one route table at a time**, but you can associate multiple subnets with the same route table.

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-19e3a2e134fe086e2	active	No



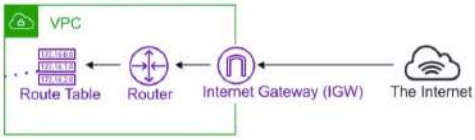
Internet Gateway (IGW)

The Internet Gateway allows **your VPC access to the internet.**

IGW does ✌ two things:

1. provide a target in your VPC route tables for internet-routable traffic
2. perform network address translation (NAT) for instances that have been assigned **public IPv4 addresses**.

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	**igw-id**



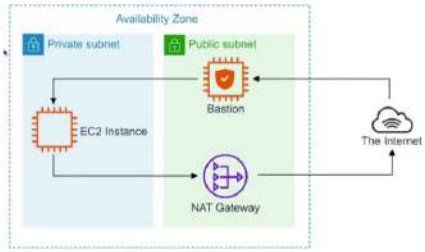
To route out to the internet you need to add in your route tables you need to add a route To the internet gateway and set the Destination to be **0.0.0.0/0**

Basiton/Jumpbox

Bastions are EC2 instances which are security harden. They are designed to help you gain access to your EC2 Instances via SSH or RCP That are in a **private subnet**.

They are also known as Jump boxes because you are jumping from one box to access another.

NAT Gateways/Instances are only intended for EC2 instances to gain outbound access to the internet for things such as security updates. NATs cannot/should not be used as Bastions

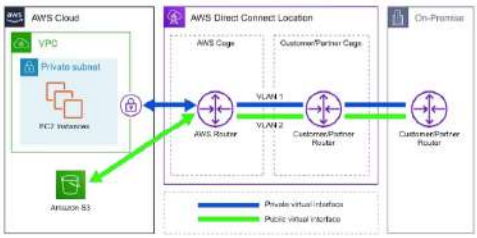


System Manager's **Sessions Manager** replaces the need for Bastions

Direct Connect

AWS Direct Connect is the AWS solution for establishing **dedicated network** connections from on-premises locations to AWS.

Very fast network Lower Bandwidth **50M-500M** or Higher Bandwidth **1GB** or **10GB**



Helps **reduce network costs** and **increase bandwidth throughput.** (great for high traffic networks)

Provides a **more consistent network experience** than a typical internet-based connection. (reliable and secure)

VPC Endpoints

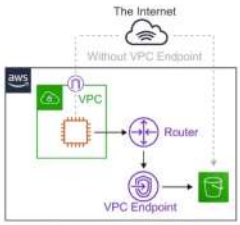
Think of a secret tunnel where you don't have to leave the AWS network

VPC Endpoints allow you to **privately connect your VPC to other AWS services**, and VPC endpoint services.

- **Eliminates** the need for an **Internet Gateway**, **NAT device**, **VPN connection**, or **AWS Direct Connect** connections.
- **Instances** in the VPC **do not require a public IP address** to communicate with service resources.
- **Traffic** between your VPC and other services **does not leave the AWS network**.
- **Horizontally scaled**, **redundant**, and **highly available** VPC component.
- Allows secure communication between instances and services - **without adding availability risks or bandwidth constraints** on your traffic.

There are ✌ **2 Types** of VPC Endpoints

1. Interface Endpoints
2. Gateway Endpoints



Interface Endpoints

Interface Endpoints are **Elastic Network Interfaces (ENI)** with a **private IP address**. They serve as an entry point for traffic going to a supported service.

Interface Endpoints are powered by **AWS PrivateLink**

Access services hosted on AWS easily and securely by keeping your network traffic within the AWS network

Pricing per VPC endpoint per AZ ($/hour)	**0.01**
Pricing per GB data processed ($)	**0.01**

~$7.5 / mo

Interface Endpoints support the following AWS Services...

- API Gateway
- CloudFormation
- CloudWatch
- Kinesis
- SageMaker
- Codebuild
- AWS Config
- EC2 API
- ELB API
- AWS KMS
- Secrets Manager
- Security Token Service
- Service Catalog
- SNS
- SQS
- Systems Manager
- Marketplace Partner Services
- Endpoint Services in other AWS accounts

VPC Gateway Endpoints

VPC Gateway Endpoints are **Free!**

A **Gateway Endpoint** is a gateway that is a target **for a specific route** in your **route table**, used for traffic destined for a supported AWS service.

To create a Gateway Endpoint, you must specify the VPC in which you want to create the endpoint, and the service to which you want to establish the connection.

AWS Gateway Endpoint currently only supports 2 services...

- Amazon S3
- DynamoDB

VPC Endpoint CheatSheet

- VPC Endpoints help keep traffic between AWS services within the AWS Network.
- There are two kinds of VPC Endpoints. Interface Endpoints and Gateway Endpoints.
- Interface Endpoints cost money, Gateway Endpoints are free.
- Interface Endpoints uses an Elastic Network Interface (ENI) with Private IP (powered by AWS PrivateLink).
- Gateway Endpoints is a target for a specific route in your route table.
- Interface Endpoints support many AWS services.
- Gateway Endpoint only supports DynamoDB and S3.

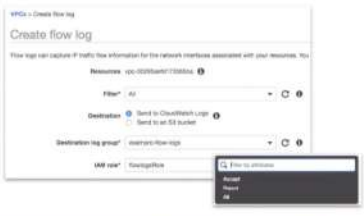
VPC Flow Logs

VPC Flow Logs allow you to capture **IP traffic information** in-and-out of Network Interfaces within your VPC.

Flow Logs can be created for.

1. **VPC**
2. **Subnets**
3. **Network Interface**

Look for this tab



All log data is **stored** using Amazon **CloudWatch Logs**.

After a Flow Log is created it can be viewed in detail within CloudWatch Logs

```
<version> <account-id> <interface-id> <srcaddr> <dstaddr> <srcport> <dstport> <protocol> <packets> <bytes> <start>
<end> <action> <log-status>

2 123456789010 eni-abc123de 172.31.16.139 172.31.16.21 20641 22 6 20 4249 1418530010 1418530070 ACCEPT OK
```

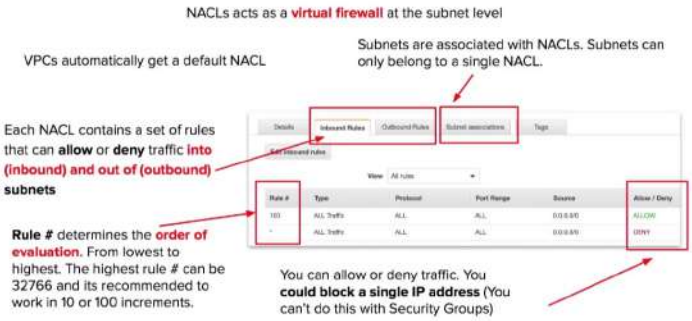
version	The VPC Flow Logs version.
account-id	The AWS account ID for the flow log.
interface-id	The ID of the network interface for which the traffic is recorded.
srcaddr	The source **IPv4 or IPv6 address**. The IPv4 address of the network interface is always its private IPv4 address.
dstaddr	The destination **IPv4 or IPv6 address**. The IPv4 address of the network interface is always its private IPv4 address.
srcport	The source port of the traffic.
dstport	The destination port of the traffic.
protocol	The IANA protocol number of the traffic. For more information, see Assigned Internet Protocol Numbers.
packets	The number of packets transferred during the capture window.
bytes	The number of bytes transferred during the capture window.
start	The time, in Unix seconds, of the start of the capture window.
end	The time, in Unix seconds, of the end of the capture window.
action	The action associated with the traffic: ACCEPT: The recorded traffic was permitted by the security groups or network ACLs. REJECT: The recorded traffic was not permitted by the security groups or network ACLs.
log-status	The logging status of the flow log: OK: Data is logging normally to the chosen destinations. NODATA: There was no network traffic to or from the network interface during the capture window. SKIPDATA: Some flow log records were skipped during the capture window. This may be because of an internal capacity constraint, or an internal error.

CheatSheet

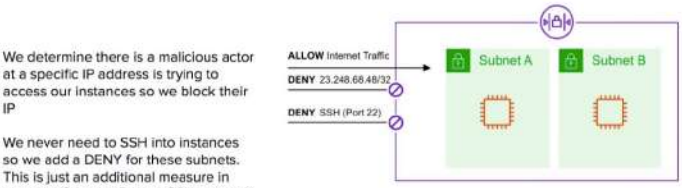
- VPC Flow Logs monitor the in-and-out traffic of your Network Interfaces within your VPC.
- You can turn on Flow Logs at the VPC, Subnet or Network Interface level.
- VPC Flow Logs **cannot be tagged** like other AWS resources.
- You **cannot change the configuration** of a flow log **after it's created**.
- You **cannot enable** flow logs for VPCs which are peered with your VPC **unless it is in the same account**.
- VPC Flow Logs can be delivered to an **S3** or **CloudWatch Logs**.
- VPC Flow Logs contains the source and destination **IP addresses** (not hostnames)
- Some instance traffic is **not monitored**:
  - Instance traffic generated by contacting the AWS DNS servers
  - Windows license activation traffic from instances
  - Traffic to and from the instance metadata address (169.254.169.254)
  - DHCP Traffic
  - Any traffic to the reserved IP address of the default VPC router

---

Network Access Control List (NACL)



NACLs Use Case

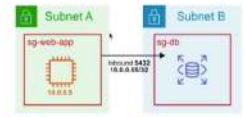


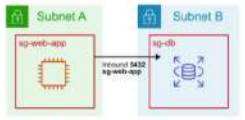
NACLs CheatSheet

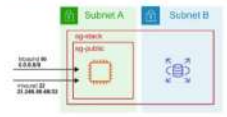
- Network Access Control List is commonly known as NACL.
- VPCs are automatically given a default NACL which allows **all** outbound and inbound traffic.
- Each subnet within a VPC must be associated with a NACL.
- Subnets can only be associated with 1 NACL at a time. Associating a subnet with a new NACL will remove the previous association.
- If a NACL is not explicitly associated with a subnet, the subnet will automatically be associated with the default NACL.
- NACL has inbound and outbound rules (just like Security Groups).
- Rule can either **allow** or **deny** traffic. (unlike Security Groups which can only allow)
- NACLs are **STATELESS** ~~(any allowed inbound traffic is also allowed outbound)~~
- When you create a NACLs it will deny all traffic by default.
- NACLs contain a numbered list of rules that get evaluated in order from lowest to highest.
- If you needed to block a single IP address you could via NACLs. (Security Groups cannot deny)

Security Groups

Security Group Use case





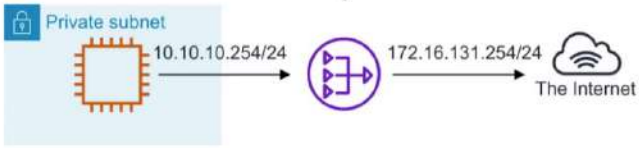


Security Group CheatSheet

- Security Groups act as a firewall at the instance level.
- Unless allowed specifically, all **inbound traffic** is **blocked by default**.
- All **Outbound traffic** from the instance is allowed by default.
- You can specific for the source to be either an IP range, single Ip Address or another security group.
- Security Groups are **STATEFUL** ~~(if traffic is allowed inbound it is also allowed outbound)~~. Note: stateful firewall tracks the operating state and characteristics of network connections traversing it, such as TCP stages.
- Any changes to a Security Group take effect immediately.
- EC2 Instances can belong to multiple security groups.
- Security groups can contain multiple EC2 Instances.
- You **cannot block specific IP addresses** with Security Groups (only allow rules), for this you would need a Network Access Control List (NACL) to deny IP addresses.
- You can have upto 10,000 Security Groups per Region. (default 2,5000)
- You can have 60 inbound and 60 outbound rules pre Security Group.
- You can have 16 Security Groups associated to an ENI (default is 5).

Network Address Translation (NAT)

Network Address Translation (NAT) is the method of **re-mapping** one IP address space into another.



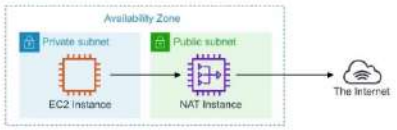
If you have a private network and you need to help gain outbound access to the internet you would need to use a NAT gateway to remap the Private IPs

If you have two networks which have conflicting network addresses you can use a NAT to make the addresses more agreeable

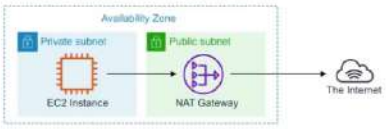
NAT Instances vs NAT Gateways

NATs have to run within a **Public Subnet**

NAT Instances (legacy) are individual EC2 instances. Community AMIs exist to launch NAT Instances.



NAT Gateways is a managed service which launches redundant instances within the selected AZ.



NAT Instance and NAT Gateway CheatSheet

NAT Instance:

- When creating a NAT instance you **must disable source and destination checks** on the instance.
- NAT instances **must exist in a public subnet**.
- You must have a **route out** of the private subnet to the NAT instance.
- The size of a NAT instance determines **how much traffic can be handled**.
- High availability can be achieved using **Autoscaling Groups**, multiple subnets in different AZs, and automate failover between them using a script.

NAT Gateway:

- NAT Gateways are **redundant inside an Availability Zone**. (can survive failure of EC2 instance)
- You can only have **1 NAT Gateway inside 1 Availability Zone**. (cannot span AZs)
- Starts at 5 Gbps and scales all the way up to 45 Gbps.
- NAT Gateways are the **preferred setup for enterprise systems**.
- There is no requirement to patch NAT Gateways, and there is no need to disable Source/Destination checks for the NAT Gateway. (unlike NAT Instances)
- NAT Gateways are **automatically assigned a public IP address**.
- Route Tables for the NAT Gateway MUST be updated.
- Resources in multiple AZs sharing a Gateway will lose internet access if the Gateway goes down, unless you create a **Gateway in each AZ** and configure **route tables** accordingly.