§ Networking & VPC

* VPC spans across a region ⟶ default VPC has CIDR ranges {172.16.0.0/12}
* subnet in is in a AZ

→ When defining the IP range for subnet 5 ip addresses are with AWS {first & last} {must belong to private IP ranges}

10.0.0.0 / 24

1. 10.0.0.0 → Network address
2. 10.0.0.1 → reserved by AWS for the VPC routes
3. 10.0.0.2 → — " — for mapping to amazon provided DNS
4. 10.0.0.3 → — " — for future use
5. 10.0.0.255 → Network Broadcast address, not supported in a VPC by AWS.

Briefing subnets & VPC ⟶ 🔗

Max CIDR in AWS is ⟹ 16

IP address & CIDRs

CIDRs → Classless Inter Domain Routing.

192.168.0.0/24 → subnet mask { represent the no. of bits in host part of ip address }

CIDR

→ The base ip or the open or the ip which represents the subnet ip range!

→ for the given CIDR the

* { mean assigned } first ip 192.168.0.0 → represents the base IP
* first assigned ip 192.168.0.1
* last-ip : 192.168.0.255 → used for broadcast purposes!
* /24 = 255.255.255.0 subnet mask!

Private IPs → for private networks (LAN)

IANA authorizes an IP range which can be used as private IPs.

* All other ips are public on internet.

(1) 10.0.0.0/8 => (10.0.0.0 → 10.255.255.255) { big networks } { 16,777,216 ips }

(2) 172.16.0.0/12 => { 172.16.0.0 - 172.31.255.255 } { AWS default VPC range }

(3) 192.168.0.0/16 => { 192.168.0.0 - 192.168.255.255 } { home network }

IPv6

→ successor to IPv4

separated by ':'

8 x 16 bit fields

hexadecimal bit fields

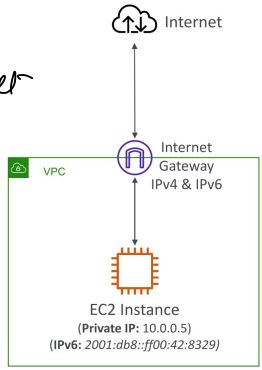
{ 0 bit segments are shown as :: }

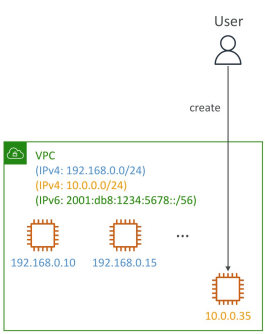
→ EC2 instance will get atleast 1 private IPv4 & a public IPv6 { as a dual-stack mode }

IAM can communicate using IPv4 or IPv6

→ Unlike IPv6, IPv4 cannot be disabled!

so if we cannot launch EC2 inst. in a subnet then we have exhausted IPv4 addr & never can it be IPv6.





→ if out of IPv4 create a new CIDR block.

- All new AWS accounts have a default VPC
- New EC2 instances are launched into the default VPC if no subnet is specified
- Default VPC has Internet connectivity and all EC2 instances inside it have public IPv4 addresses
- We also get a public and a private IPv4 DNS names

VPC in AWS – IPv4

- VPC = Virtual Private Cloud
- You can have multiple VPCs in an AWS region (max. 5 per region – soft limit)
- Max. CIDR per VPC is 5, for each CIDR:
  - Min. size is /28 (16 IP addresses)
  - Max. size is /16 (65536 IP addresses)
- Because VPC is private, only the Private IPv4 ranges are allowed:
  - 10.0.0.0 – 10.255.255.255 (10.0.0.0/8)
  - 172.16.0.0 – 172.31.255.255 (172.16.0.0/12)
  - 192.168.0.0 – 192.168.255.255 (192.168.0.0/16)

- Your VPC CIDR should NOT overlap with your other networks (e.g., corporate)

→ multiple VPC in region (5 max)
→ max 5 CIDR per VPC all private range belonging to

VPC – Subnet (IPv4)

- AWS reserves 5 IP addresses (first 4 & last 1) in each subnet
- These 5 IP addresses are not available for use and can't be assigned to an EC2 instance
- Example: if CIDR block 10.0.0.0/24, then reserved IP addresses are:
  - 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 mapping to Amazon-provided DNS
  - 10.0.0.3 – reserved by AWS for future use
  - 10.0.0.255 – Network Broadcast Address. AWS does not support broadcast in a VPC, therefore the address is reserved

- Exam Tip, if you need 29 IP addresses for EC2 instances:
  - You can't choose a subnet of size /27 (32 IP addresses, 32 – 5 = 27 < 29)
  - You need to choose a subnet of size /26 (64 IP addresses, 64 – 5 = 59 > 29)

Q How to provide internet access to VPC?

→ for public subnets!

1. Internet Gateway
   - → gives internet access to resource in a VPC
   - → created separately from VPC
   - → with high horizontal scalability &
   - → one IGW to one VPC
   - * Route table must be edited to allow IGW to allow internet access

→ For private subnets we have 2 ways!

1. Bastion Host - {aka jump server or jump host}
   - → To provide SSH connection to EC2 instances in the private subnet
   - → bastion is itself hosted on a server in the public subnet which is then connected to the private subnet.

this is a "jump server" meaning it has a singular purpose of just letting us connect through it.

Then we use "SSH tunnelling" or "SSH port forwarding".

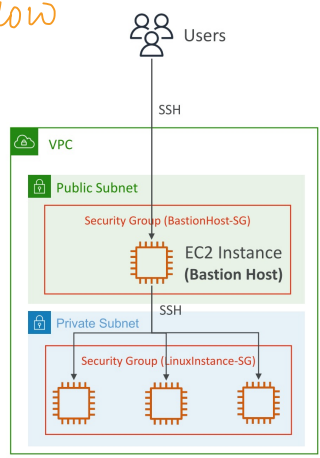
i.e. the user connects to the bastion host via SSH at port 22; port forwarding is established b/w bastion host & the internal private server; all the commands issued & responses recieved are to & from the private server through the bastion host via SSH tunnel!

* In Tech terms; → a jump server serves as an intermediary servers that allows secure access to other servers within a hosted environment.

Remember!

→ (1) Security group @ bastion host must allow inbound from internet @ port 22

→ (2) Security group @ EC2 instance must allow the security group of bastion host or the private IP of bastion host.



(2) NAT {Network Access Translation}

→ gives resources (EC2 instances) in private subnet an access to internet

→ hosted in public subnet

→ must have an Elastic IP attached to it

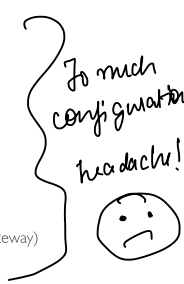
→ Route tables need to be configured to send all the packets from private instances to the NAT gateway.

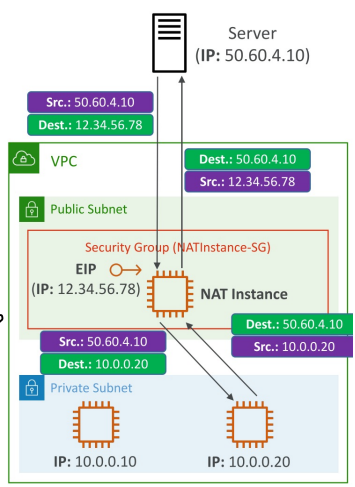
→ EC2 settings : Source / destination check must be disabled

basically are outdated !

NAT Instance – Comments

- Pre-configured Amazon Linux AMI is available
  - Reached the end of standard support on December 31, 2020
- Not highly available / resilient setup out of the box
  - You need to create an ASG in multi-AZ + resilient user-data script
- Internet traffic bandwidth depends on EC2 instance type
- You must manage Security Groups & rules:
  - Inbound:
    - Allow HTTP / HTTPS traffic coming from Private Subnets
    - Allow SSH from your home network (access is provided through Internet Gateway)
  - Outbound:
    - Allow HTTP / HTTPS traffic to the Internet





* Instead we should use

NAT Gateway

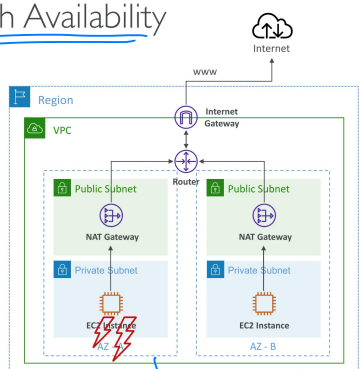
- AWS-managed NAT, higher bandwidth, high availability, no administration
- Pay per hour for usage and bandwidth
- NATGW is created in a specific Availability Zone, uses an Elastic IP
- Can't be used by EC2 instance in the same subnet (only from other subnets)
- Requires an IGW (Private Subnet => NATGW => IGW)
- 5 Gbps of bandwidth with automatic scaling up to 100 Gbps
- No Security Groups to manage / required

ezzz pzzz!

→ doesn't need managing yourself!

NAT Gateway with High Availability

- NAT Gateway is resilient within a single Availability Zone
- Must create multiple NAT Gateways in multiple AZs for fault-tolerance
- There is no cross-AZ failover needed because if an AZ goes down it doesn't need NAT



if this AZ fails then NAT is not needed anymore.

* Security groups	NACLs
→ control the inbound & outbound traffic in an EC2, ALB, ENIs	→ control the inbound & outbound traffic at subnet level.
→ instance level	→ subnet level
→ stateful {whatever is accepted in also accepted out & vice-versa}	→ stateless {outbound rules are evaluated & inbound rules —"— }
→ Many SGs to one EC2	→ one NACL to per subnet

→ All rule are evaluated.

→ lower the number in list higher the precidense

→ First rule match will be decisive

→ newly created NACL s will deny every thing.

dyault NACL

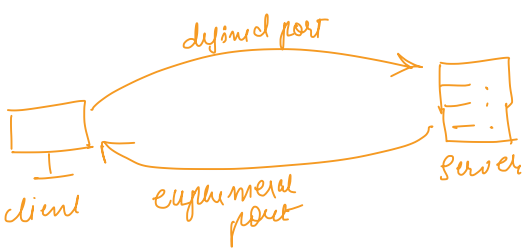
[→ accepts everything inbound / outbound !

if associated with a subnet!

but agar ऐसा नहीं करेंगे तो initially ~~दि करना~~ होगी ! contradicts

→ best practic ⇒ do NOT modify dyault NACL instead create a custom one.

Ephemeral Ports

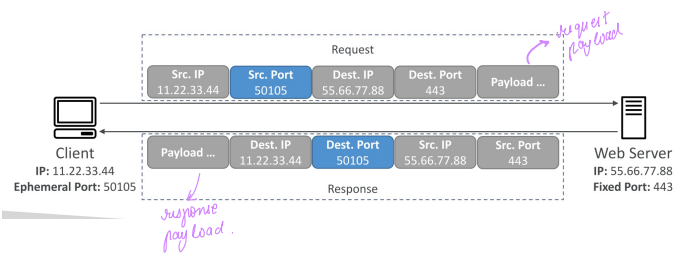


→ clients connect to a dyined port on server but if server want to send some requests, the client doesn't have any open ports by dyault.

so if a client is connecting to any server it will open a ephemeral port because it is alive till the connection is present.

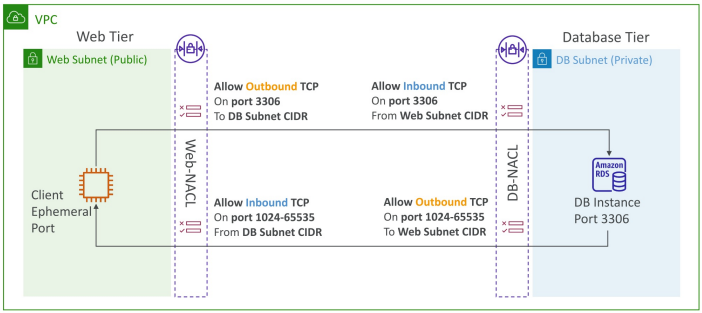
Different OS use different port ranges.

∴ It is a temp. port just for connection time.



NACL with Ephemeral Ports

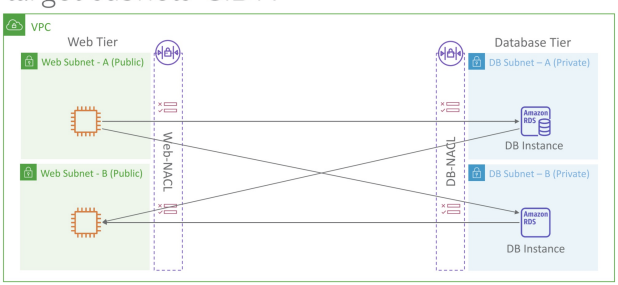
→ super imp. to config.



https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html#nacl-ephemeral-ports

↓ hence a combination of NACL rules for each subnets CIDR is necessary!

Create NACL rules for each target subnets CIDR



Q. Why associate ephemeral ports with NACLs and not with security groups?

→ NACLs are stateless packet filters
security grps are stateful connection tracking filters.

(1) NACLs are rarely used."

NACLs allow "explicit denies of traffic" which aren't allowed with SGs. Reiterating what other replies have said SGs are preffered for sure as they track connections & states rather than just direction ips & ports.
But when denying things or have ip range requirements then NACLs are good.

NACL is stateless

① It evaluate each packet independently without regard to previous packet.

② Incoming or outgoing each packet must be explicitly allowed or denied

③ Since NACLs do not remember the state of a conn hence the return traffic must be explicitely allowed / denied which mean for initial request ports के लिए लिए ephemeral ports used by the client's response!

Security grp is stateful

① Track the connection state

② If an inbound rule is set to allow on a specific port the response traffic is automatically allowed w/o the need of explicit ephemeral ports.

③ Its stateful nature simplifies the rule management.

VPC peering

→ privately connect 2 VPCs using AWS private network (in same as well as diff AWS acc.)

→ no overlapping CIDR

→ behave as if it is a single network.

→ NOT TRANSITIVE

→ Route tables must be updated in each VPC subnets to ensure EC2 can communicate

VPC Endpoint

→ allow us to connect to AWS services using AWS private network instead of public internet.

→ Route table & DNS resolution setting needs to be checked.

→ if we were to use internet gateway then we will be providing public internet access so there will be NAT devices or firewall to manage. so use VPC endpoints to leverage AWS private network.

endpoint?
→ a specific location for accessing a service using a particular protocol & data format.

→ interface endpoint is within the VPC which uses the private IPs to communicate.
while gateway endpoints are place within the scope of VPC through prefix-list in the VPC's route table.

(1) The first type of endpoint, a *Gateway Load Balancer* endpoint, allows you to intercept traffic and route it to a network or security service that you've configured using a Gateway Load Balancer. Gateway load balancers enable you to deploy, scale, and manage virtual appliances, such as firewalls, intrusion detection and prevention systems, and deep packet inspection systems. Our colleague Justin Davies has written an excellent blog post on supported architectural patterns using AWS Gateway Load Balancers.

(2) The second type of endpoint, a *Gateway* endpoint, allows you to provide access to Amazon Simple Storage Service (S3) and Amazon DynamoDB. You can configure resource policies on both the gateway endpoint and the AWS resource that the endpoint provides access to. A VPC endpoint policy is an AWS Identity and Access Management (AWS IAM) resource policy that you can attach to an endpoint. It is a separate policy for controlling access from the endpoint to the specified service. This enables granular access control and private network connectivity from within a VPC. For example, you could create a policy that restricts access to a specific DynamoDB table. This policy would only allow certain users or groups to access the table through a VPC endpoint.

no SG needed & you can use only S3 & DDB

Figure 1: Accessing Amazon S3 via a Gateway VPC endpoint

→ a policy to restrict access to specific dynamoDB table for security.

(3) The third type of endpoint, an *Interface* endpoint, allows you to connect to services powered by AWS PrivateLink. This includes a large number of AWS services. It also can also include services hosted by other AWS customers, and AWS Partner Network (APN) partners in their own VPCs. By using AWS partner services through AWS PrivateLink, you no longer have to rely on access to the public internet. Data transfer charges for traffic from Amazon EC2 to the internet vary based on volume. After the first 1 GB / month ($0.00 per GB), transfers are charged at a rate of $ 0.09/GB (for AWS US-East 1 Virginia). Like gateway endpoints, interface endpoints can be secured using resource policies on the endpoint itself, and the resource that the endpoint provides access to. Interface endpoints allow the use of security groups to restrict access to the endpoint.

→ need to provision ENI (for private link) & must attach security groups

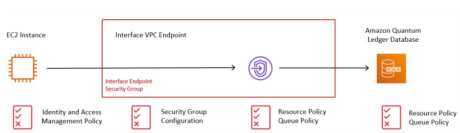
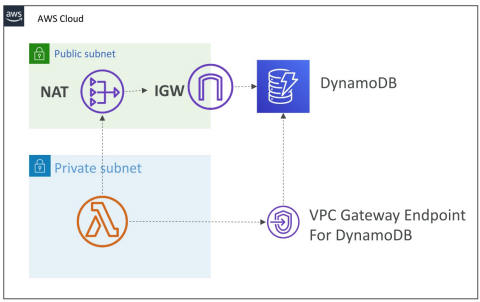


Figure 2: Accessing QLDB via an Interface VPC endpoint

→ interface endpoint are good for high performance & more security. but gateway endpoint is cost effective with basic connection needs.

- DynamoDB is a public service from AWS
- Option 1: Access from the public internet
  - Because Lambda is in a VPC, it needs a NAT Gateway in a public subnet and an internet gateway
- Option 2 (better & free): Access from the private VPC network
  - Deploy a VPC Gateway endpoint for DynamoDB
  - Change the Route Tables



VPC flow logs

→ captures information about ip traffic going through your interface.

into { . VPC flow logs
. subnet flow logs
. ENI flow logs

VPC Flow Logs

- Capture information about IP traffic going into your interfaces:
  - VPC Flow Logs
  - Subnet Flow Logs
  - Elastic Network Interface (ENI) Flow Logs
- Helps to monitor & troubleshoot connectivity issues
- Flow logs data can go to S3, CloudWatch Logs, and Kinesis Data Firehose
- Captures network information from AWS managed interfaces too: ELB, RDS, ElastiCache, Redshift, WorkSpaces, NATGW, Transit Gateway…

VPC Flow Logs Syntax

version interface-id dstaddr dstport packets start action

```
2 123456789010 eni-1235b8ca123456789 172.31.16.139 172.31.16.21 20641 22 6 20 4249 1418530010 1418530070 ACCEPT OK
2 123456789010 eni-1235b8ca123456789 172.31.9.69 172.31.9.12 49761 3389 6 20 4249 1418530010 1418530070 REJECT OK
```

account-id srcaddr srcport protocol bytes end log-status

- **srcaddr & dstaddr** – help identify problematic IP
- **srcport & dstport** – help identity problematic ports
- **Action** – success or failure of the request due to Security Group / NACL
- Can be used for analytics on usage patterns, or malicious behavior
- **Query VPC flow logs using Athena on S3 or CloudWatch Logs Insights**
- Flow Logs examples: https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs-records-examples.html

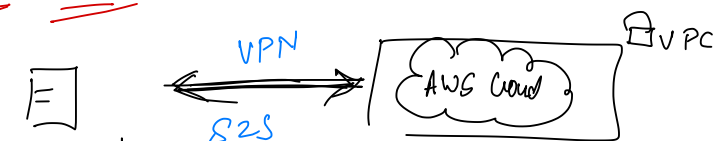
→ These logs can be used to trouble shoot network/ connection issues.

if the deny in inbound, reject ⇒ NACL or SG (any)
if inbound allow, outbound reject ⇒ NACL

if outbound reject ⇒ NACL or SG
outbound allow, inbound reject ⇒ NACL

किसकी गलती!

AWS S2S (Site-to-site VPN)



CGW → software or physical device on customer side

VGW → in VPC which needs to be accessed from.

VPN can also customize ASN (autonomous system number)

CGW → जो customer का ip → public (customer's gateway) → public ip

NAT / client (जो भी client हो)

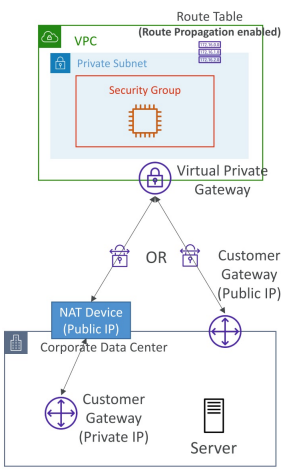
→ Route propogation for VGW in the route tables associated with your subnet must be ENABLED!

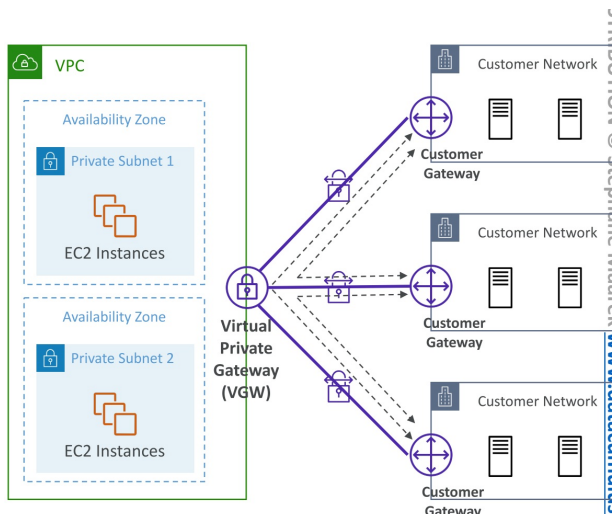
→ If multiple such VPN connections from different sites then use

AWS VPN Cloud Hub → works on "hub-and-spoke" model for primary or secondary connectivities between different locations.

→ VPN connection hereafter goes through public internet.

→ connect multiple VPN connections on the same VGW & configure the route tables & setup dynamic routing.



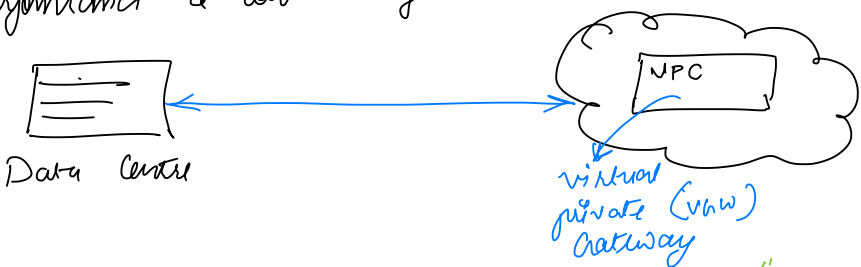


VPN Cloud hub

→ If EC2 needs to be pinged from on premise then security group must include ICMP protocol as an inbound rule.

AWS Direct Connect (DX)

→ To provide a private connection to AWS cloud for high performance & low latency networks. (Lower cost)

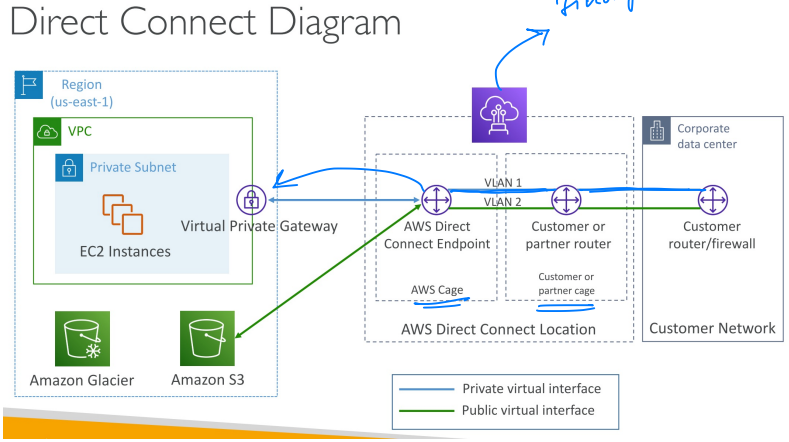


can access both "private" as well as "public" resources on same connection (EC2) (S3)

→ consistent network (real-time data feeds)
hybrid Environment-
Supports IPv4 & IPv6

Find physical location (available on web)

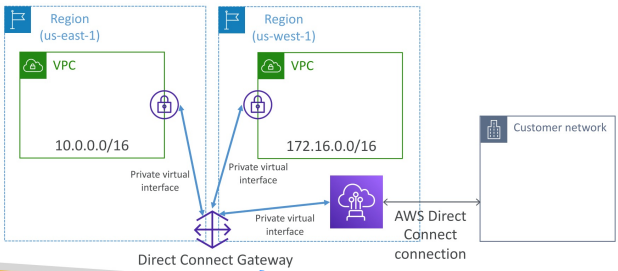
Direct Connect Diagram



Direct Connect Gateway

multiple VPCs & multiple regions.

- If you want to setup a Direct Connect to one or more VPC in many different regions (same account), you must use a Direct Connect Gateway



Connection types

(1) dedicated Connections { high throughput, bandwidth }

→ physical ethernet ports for a customer.

AWS direct connect partners do this! { 1Gbps – 10 Gbps }

(2) hosted Connections { less suitable }

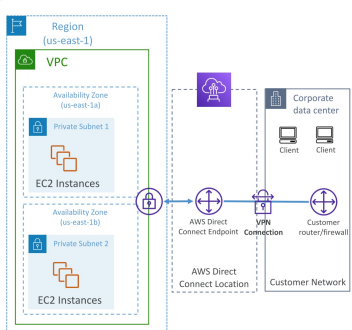
→ on demand capacity adding or removal. { 50 → Mbps }

Lead times to establish new connections are often longer than 1 month i.e. establishing takes time.

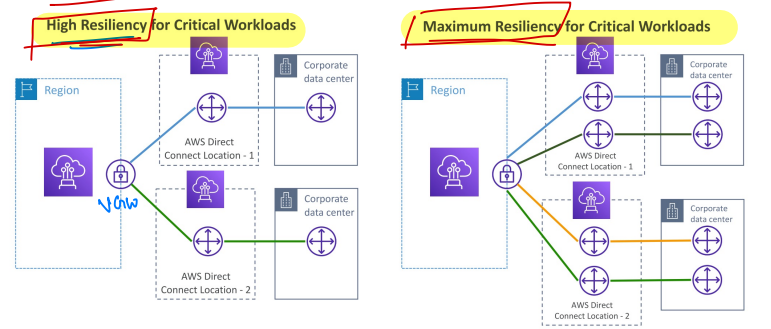
Direct Connect – Encryption

for encryption

- Data in transit is not encrypted but is private
- AWS Direct Connect + VPN provides an IPsec-encrypted private connection
- Good for an extra level of security, but slightly more complex to put in place

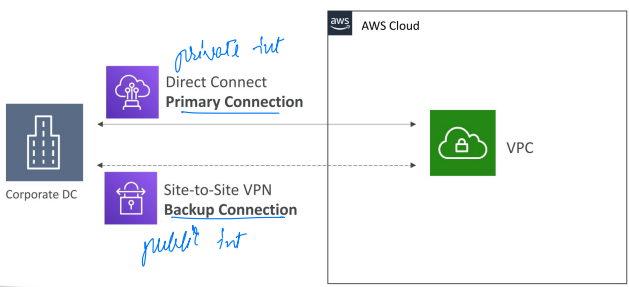


Direct Connect - Resiliency



Site-to-Site VPN connection as a backup

- In case Direct Connect fails, you can set up a backup Direct Connect connection (expensive), or a Site-to-Site VPN connection



→ Network topologies can become very complicated with multiple VPCs, gateways, VPN connection or direct connect or VPC peering

So use Transit Gateway

Transit gateway

→ for having transitive peering b/w 1000s of VPC, VPN & (Or) "on-premises", hub-and-spoke (star) connection

{ direct connect gateway
VPN conn

→ Regional resource (cross region)
→ cross-ac (using regional resource manager RAM)
→ Route tables (for which VPC to talk to who)

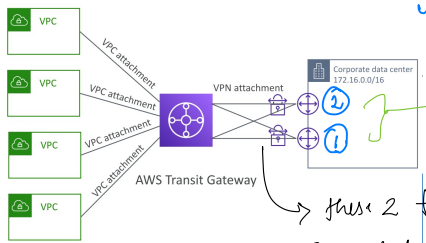
→ Only service which supports IP Multicast

Transit Gateway: Site-to-Site VPN ECMP

→ create multiple site to site VPN connection to increase bandwidth

- ECMP = Equal-cost multi-path routing
- Routing strategy to allow to forward a packet over multiple best path
- Use case: create multiple Site-to-Site VPN connections to increase the bandwidth of your connection to AWS

use case



4 tunnels! (1) throughput

these 2 tunnels is for data forwd & data received. Transit GW uses both these tunnel as part of the VPC connection.

Transit Gateway: throughput with ECMP

VPN to virtual private gateway

1x = 1x VPC

1x = 1.25 Gbps

VPN connection (2 tunnels)

1 of throughput

VPN to transit gateway

multiple VPCs

1x = 1x VPC VPC VPC VPC

[1.25 x 2] because of ECMP!

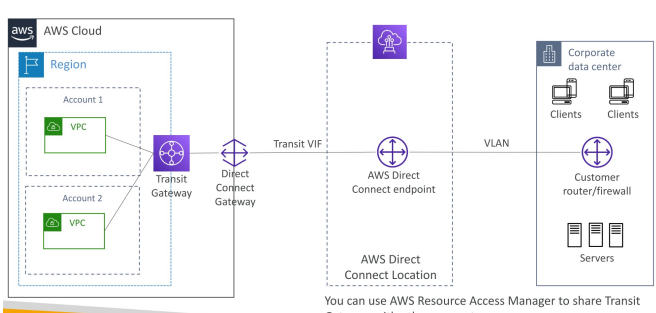
1x = 2.5 Gbps (ECMP) – 2 tunnels used

2x = 5.0 Gbps (ECMP) → 2 VPN conn

3x = 7.5 Gbps (ECMP) → 3 VPN conn

+$$ per GB of TGW processed data { each GB of data through TGW is payable }

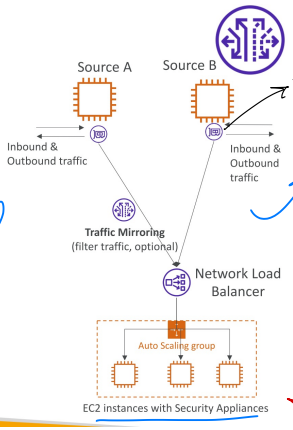
→ share direct connect connection with multiple accounts using transit gateway.



VPC traffic Mirroring

VPC – Traffic Mirroring

- Allows you to capture and inspect network traffic in your VPC
- Route the traffic to security appliances that you manage
- Capture the traffic
  - From (Source) – ENIs
  - To (Targets) – an ENI or a Network Load Balancer (NLB)
- Capture all packets or capture the packets of your interest (optionally, truncate packets)
- Source and Target can be in the same VPC or different VPCs (VPC Peering)
- Use cases: content inspection, threat monitoring, troubleshooting, …



capture the traffic of src & w/o disrupting the flow of src

→ copy network traffic from ENIs for further analysis.

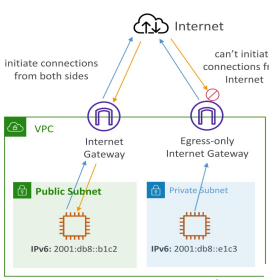
→ for VPC ip-traffic enable VPC flow log

Egress-only Internet gateway

→ only for IPv6 {similar to NAT but for IPv6}

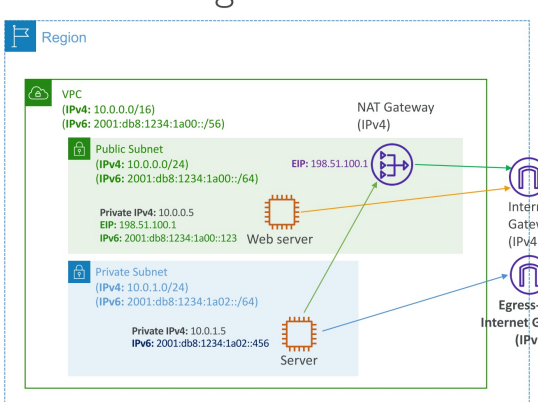
→ will be able to access internet over ipv6 but the internet will not be able to initiate conn from the → EC2.

only outbound conn!



Remember

IPv6 Routing



Route Table
(Public Subnet)

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	igw-id
::/0	igw-id

Route Table
(Private Subnet)

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	nat-gateway-id
::/0	eigw-id

AWS private link

→ to connect service privately from service VPC to customer VPC

→ Doesn't need public internet access

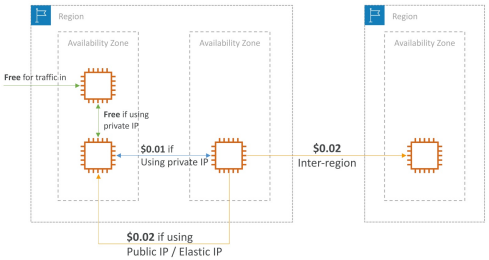
→ Must be used with NLB & ENIs

Classic link (deprecated)

Connect EC2-classic EC2 instances privately to your VPC

Networking Cost in AWS

① For EC2



- Use Private IP instead of Public IP for good savings and better network performance
- Use same AZ for maximum savings (at the cost of high availability)

→ traffic into EC2 is free

→ any traffic within the same AZ (subnet) using their private ip is free

→ if EC2 in diff AZ in same region.
- public (elastic ip) $ => $0.02 per GB (bc traffic has to leave AWS network to comm.)
- private ip $ => 0.01 (half) (using internal network)

→ traffic from 1 region to other is 0.02 $ per GB (expensive)

→ In in same region Use private ip as much as possible

→ max cost saving if being in same AZ (subtle tradeoff)
(but might fuck up availability during failures)

Example replicating RDS DB two same AZ will 'not' cost in terms of network

→ Minimize egress traffic

egress traffic → outbound (from AWS to outside)
Ingress traffic → inbound (from outside to AWS)

* Try to keep as much traffic within AWS to minimize cost.
* direct connect location is in same AWS region will result in lower egress cost.

(2) For S3

S3 ingress → free

S3 egress → $0.09 per GB

S3 transfer acceleration → + $(0.04 - 0.08) per GB

S3 to Cloudfront → free

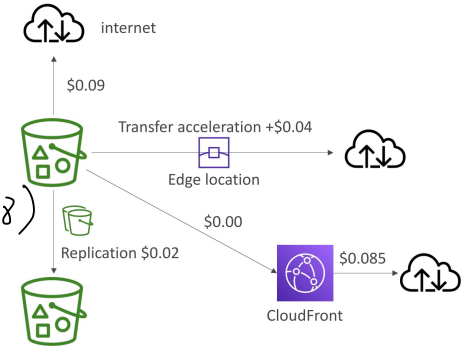
but

Cloudfront to internet → $0.085 per GB (slightly cheaper than S3)

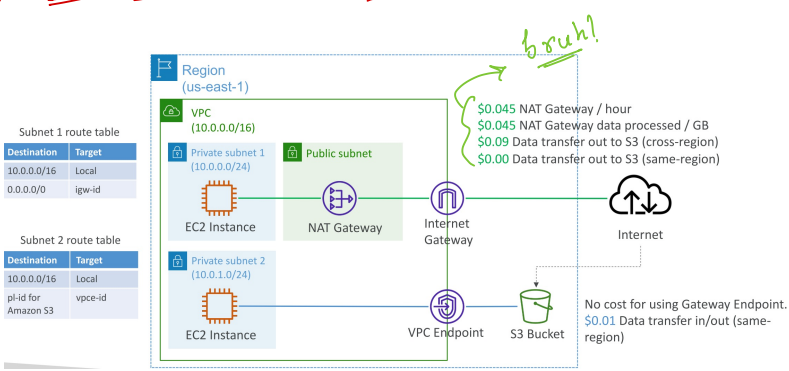
or

+ caching capability
+ Request to CF is cheaper than S3

cross region replic" → $0.02 per GB



(3) NAT GW v/s Gateway VPC Endpoint



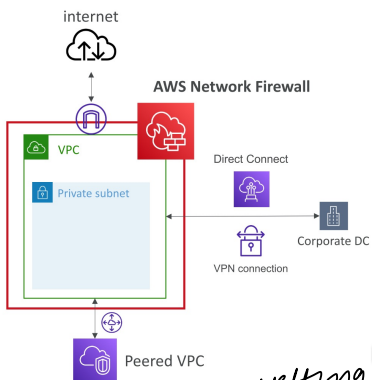
→ like other network protection services on AWS.
NACLs, WAF, Shield, Firewall Manager, VPC SGs
we can have firewall for entire VPC.

AWS network firewall

→ layer 3 to layer 7 protection

AWS Network Firewall

- Protect your entire Amazon VPC
- From Layer 3 to Layer 7 protection
- Any direction, you can inspect
  - VPC to VPC traffic
  - Outbound to internet
  - Inbound from internet
  - To / from Direct Connect & Site-to-Site VPN
- Internally, the AWS Network Firewall uses the AWS Gateway Load Balancer
- Rules can be centrally managed cross-account by AWS Firewall Manager to apply to many VPCs



instead of us setting third party appliance for security using the gateway load balancer we can use this managed AWS service! damn!

Network Firewall – Fine Grained Controls

- Supports 1000s of rules
  - IP & port - example: 10,000s of IPs filtering
  - Protocol – example: block the SMB protocol for outbound communications
  - Stateful domain list rule groups: only allow outbound traffic to *.mycorp.com or third-party software repo
  - General pattern matching using regex
- Traffic filtering: Allow, drop, or alert for the traffic that matches the rules
- Active flow inspection to protect against network threats with intrusion-prevention capabilities (like Gateway Load Balancer, but all managed by AWS)
- Send logs of rule matches to Amazon S3, CloudWatch Logs, Kinesis Data Firehose