**VPC**

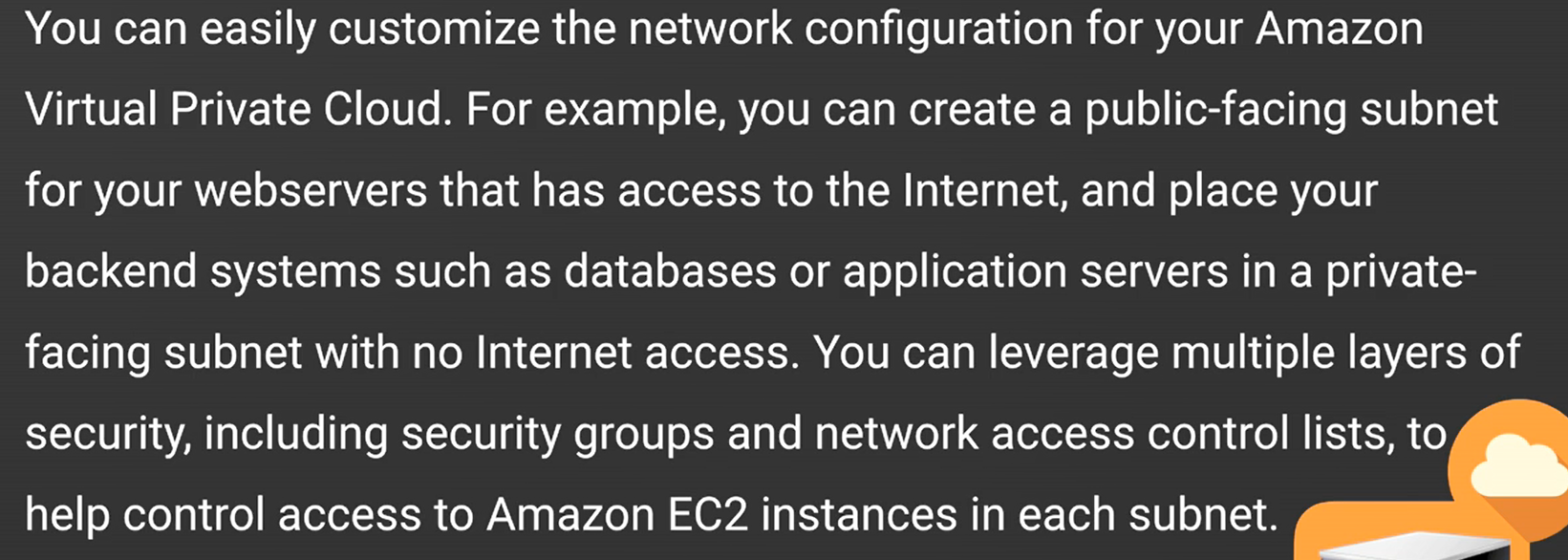
**What is VPC ?**

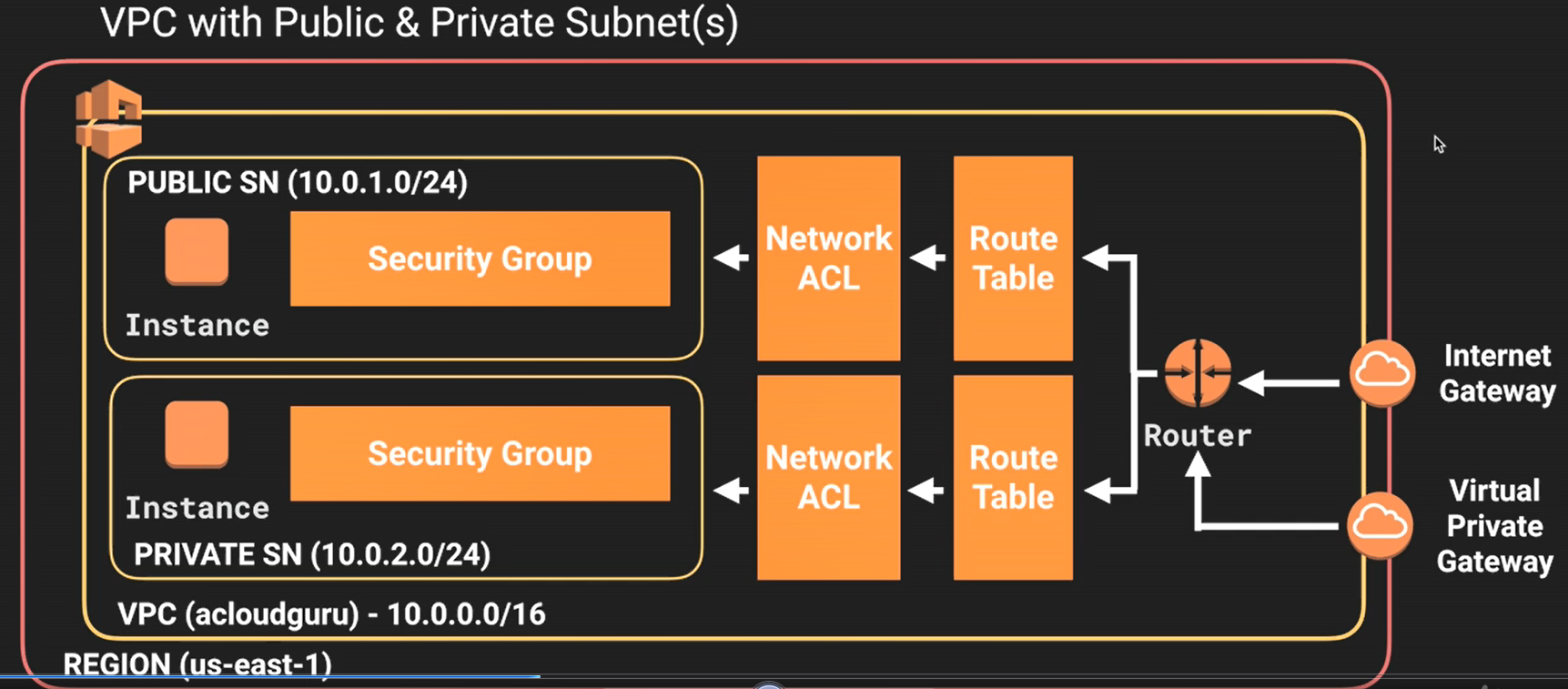
Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC. You can specify an IP address range for the VPC, add subnets, associate security groups, and configure route tables.

Additionally you can create a hardware VPN(Virtual private network) connection between your data center and your VPC and leverage the aws cloud as an extension of your corporate data center.

**What is subnet ?**

A *subnet* is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet, and a private subnet for resources that won't be connected to the internet.





**Relation between region, VPC and subnet and az ?**

We can have more than one VPC in a region. One VPC can have multiple subnet, some of them are private some of them are public. Each subnet be in different availability zone,(1 az = 1 subnet) means 10.0.0.0/24 can not be in any other availability zone. But one availability zone can have more than one subnet.

**How many internet gateway is possible per VPC ?**

There can be only one internet gateway for one VPC.

**Default VPC vs custom VPC**



**What is VPC peering ?**

One instance of VPC A can communicate with another instance of VPC B. This kind of VPC connection is not transitive.

**Whenever we create a new VPC by default which components are getting created ?**

Default security group

Default network access control list.

Default route table.

It won’t create subnet, internet gateway.



**Difference between security groups for EC2 classic and and EC2 VPc ?**

You can't use the security groups that you've created for use with EC2-Classic with instances in your VPC. You must create security groups specifically for use with instances in your VPC. The rules you create for use with a security group for a VPC can't reference a security group for EC2-Classic, and vice versa.

**How to assign a security group to a instance which is in our custom VPC ?**

Create a custom security group, attach the public subnet address of the VPC to the security group. Go to the instance which is in VPC and attach this security group to the instance.

**How many ip address we can not use of an ipaddress range of a subnet ?**

Total ipaddress - 5

First four and the last address we cannot use

1st is a network address. 2nd resrved for VPC router. 3rd and 4th for aws internal use. last is broadcast address.

**What is routing table default action for a VPC ?**

If we do not create any routing table for a VPC , aws always create a new default routing table for each VPC. All subnets are in this route table exposed to internet by default. To solve this we need to create a new route table assign it to a VPC and assign a internet gateway with the routing table, and assign the subnets which we want to make public.

**Is Aws auto assign public ip with the instance in a custom vpc ?**

No, by default auto Ip assigned is not selected. We need to select the option of auto ip assign for the subnet.

**What is NAT instance ?**

You can use a network address translation (NAT) instance in a public subnet in your VPC to enable instances in the private subnet to initiate outbound IPv4 traffic to the Internet or other AWS services, but prevent the instances from receiving inbound traffic initiated by someone on the Internet.

**What is disabling source and destination check ?**

Each EC2 instance performs source/destination checks by default. This means that the instance must be the source or destination of any traffic it sends or receives. However, a NAT instance must be able to send and receive traffic when the source or destination is not itself. Therefore, you must disable source/destination checks on the NAT instance.

**How to attach our VPC with a NAT instance ?**

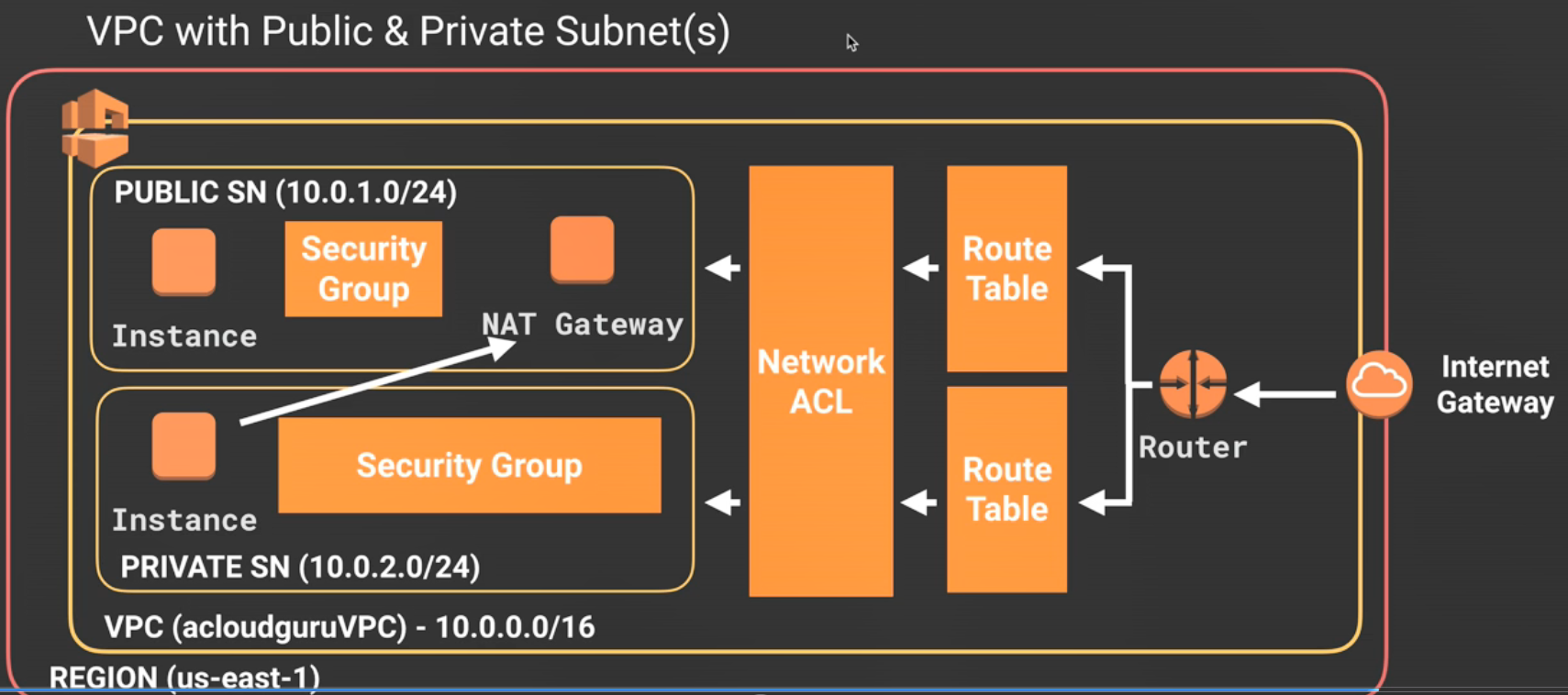
Go to Internet route table of our VPC and connect with NAT instance instead of Internet gateway. Nat instance is nothing but a EC2 instance of NAT (Network address translation type).

**What is the disadvantage of NAT instance ?**

We know by connecting with NAT instance a private subnet can access internet, but internet cannot access the private subnet. But we are relying on only one instance, if the NAT instance goes down, our subnets cannot access internet. For that aws introduces nat gateway.

**Advantages of NAT gateway over NAT instance**

* We do not need to maintain or install antivirus, or software patch in nat gateway unlike nat instance.
* Nat gateway is clustered, so no need to scare when the instance is down.
* We cannot configure Bastion host in NAT gateway.
* In the below picture private instance send outbound traffic through NAT gateway which is after security group. But incase of NAT instance it seats behind the security group, because it is an instance.
* NAT gateway is not associated with security group like NAT instance.



**Is Ipv6 is allowed in NAT gateway ?**

No, for Ipv6 gateway we need Egress only internet gateways.

**Can I connect one subnet with multiple NACL ?**

No, one subnet can connect only one NACL. But one NACL can be accessed by multiple subnets.

**Can a security group spread across multiple subnet ?**

Yes. one security group can be accessed by multiple subnet in one VPC.

It cannot span VPC

**What is NACL?**

Network access control list is like a firewall to our VPC, where we can set some rules to allow or deny inbound and outbound traffic. One default NACL is always created when we create our own VPC.

**What is the default traffic rule in NACL when we create our own NACL ?**

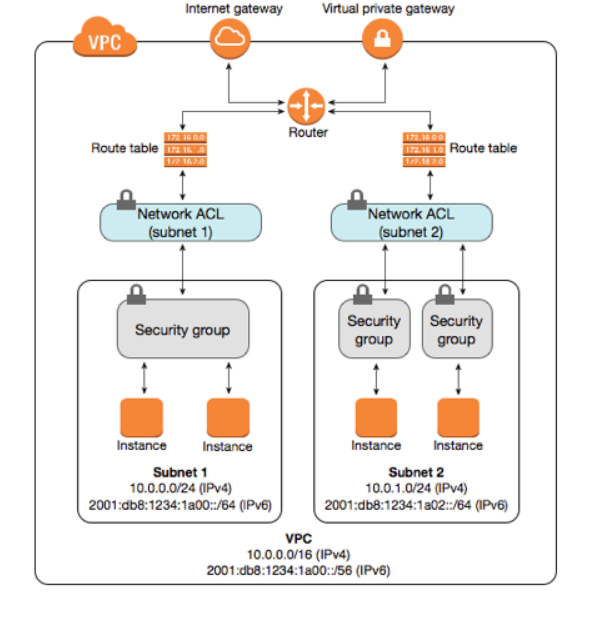
By default all the traffics are denied. default traffic rule starts with 100.

**Rule number technique in NACL ?**

If we add two rules for http one at 100, another at 101, 100 is for allow and 101 is for deny.

Then NACL will allow the http traffic. If we change the order then it will deny.

**Difference between security group and NACL ?**



|  |  |
| --- | --- |
| **Security Group** | **Network ACL** |
| **Supports Allow rules only {**by default all rules are denied **}**  You **cannot deny** a certain IP address from establishing a connection | **Supports Allow and Deny rules**  By Deny rules we mean, you could explicitly deny a certain IP address to establish a connection example: Block IP address 192.168.0.2 from establishing a connection to an EC2 Instance |
| **Stateful:**This means any changes applied to an incoming rule will be automatically applied to the outgoing rule.  Example: If you allow an incoming port 80, the outgoing port 80 will be automatically opened. | **Stateless**: This means any changes applied to an incoming rule will not be applied to the outgoing rule.  Example: If you allow an incoming port 80, you would also need to apply the rule for outgoing traffic. |
| **Security groups are tied to an instance**. | **Network ACL are tied to the subnet.** This means any instances within the subnet group gets the rule applied.  If you have many instances, managing the firewalls using Network ACL can be very useful. Otherwise, with Security group, you have to manually assign a security group to the instances. |
| **All rules in a security group are applied.** | **Rules are applied** **in their order (the rule with the lower number gets processed first)**  Example:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Rule # (Order number) | Type | Protocol | Port Range | Source | Allow/Deny | | 100 | HTTP (80) | TCP (6) | 80 | 0.0.0.0/0 | ALLOW | | 200 | HTTPS (443) | TCP (6) | 443 | 0.0.0.0/0 | ALLOW | |
| **First Layer of Defence** | **Second Layer of the defence** |
| **Is the Firewall of EC2 Instances** | **Is the Firewall of the Subnet** |
| Security groups are used for many cases, for example restricting inbound traffic of an EC2 instance to be from Load balancer only. |  |

**Relation with Elastic load balancer and VPC ?**

* An elastic load balancer in a VPC should connect with more than one subnet
* At least one subnet should be connected with internet gateway.

**Routing table and relation with other vpc components**

[**https://docs.aws.amazon.com/vpc/latest/userguide/VPC\_Route\_Tables.html**](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Route_Tables.html)

**What is VPC flow log ?**

VPC Flow Log is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data can be published to Amazon CloudWatch Logs and Amazon S3. After you've created a flow log, you can retrieve and view its data in the chosen destination.

Flow logs can help you with a number of tasks; for example, to troubleshoot why specific traffic is not reaching an instance, which in turn helps you diagnose overly restrictive security group rules. You can also use flow logs as a security tool to monitor the traffic that is reaching your instance.

**5-tuple of vpc flow log :**

Source address

source port

destination address

destination port

protocol

**Limitations of VPC flow log**

* You cannot enable flow logs for network interfaces that are in the EC2-Classic platform. This includes EC2-Classic instances that have been linked to a VPC through ClassicLink.
* You can't enable flow logs for VPCs that are peered with your VPC unless the peer VPC is in your account.
* You cannot tag a flow log.
* After you've created a flow log, you cannot change its configuration; for example, you can't associate a different IAM role with the flow log. Instead, you can delete the flow log and create a new one with the required configuration.
* If your network interface has multiple IPv4 addresses and traffic is sent to a secondary private IPv4 address, the flow log displays the primary private IPv4 address in the destination IP address field.
* If traffic is sent to an ENI and the destination is not any of the ENI IP addresses, the flow log displays the primary private IPv4 address in the destination IP address field.

Flow logs do not capture all IP traffic. The following types of traffic are not logged:

* Traffic generated by instances when they contact the Amazon DNS server. If you use your own DNS server, then all traffic to that DNS server is logged.
* Traffic generated by a Windows instance for Amazon Windows license activation.
* Traffic to and from 169.254.169.254 for instance metadata.
* Traffic to and from 169.254.169.123 for the Amazon Time Sync Service.
* DHCP traffic.
* Traffic to the reserved IP address for the default VPC router. For more information, see [VPC and Subnet Sizing](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Subnets.html#VPC_Sizing).
* Traffic between an endpoint network interface and a Network Load Balancer network interface. For more information, see [VPC Endpoint Services (AWS PrivateLink)](https://docs.aws.amazon.com/vpc/latest/userguide/endpoint-service.html).

**What is VPC peering ?**

VPC peering is a technology by which two vpc can be communicated even if they don’t have any internet gateway and act like they are in a single network. VPC peering can happen between two VPC of same region , of different region even of different aws accounts as well.

Traffic within VPC in same aws account stays in aws account and beyond account stays in aws backbone network, never exposes to internet.


            A VPC peering connection
        

**How to create a VPC peering ?**

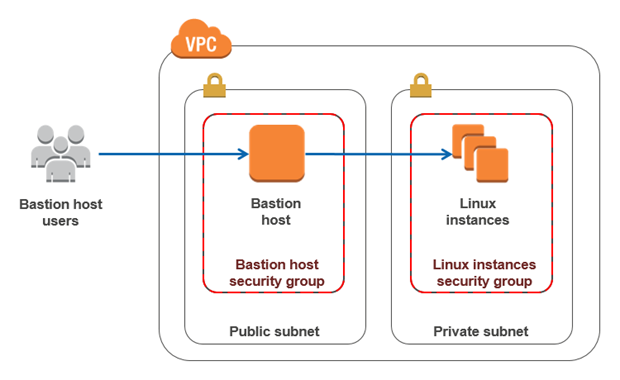
* 1. Create a vpc connecting network
  2. Add a requester
  3. Add a accepter if they are from different regions
  4. Modify routing table of each subnet set destination ip range and target as vpc peering connection like PCX-123456

**What is bastion host ?**

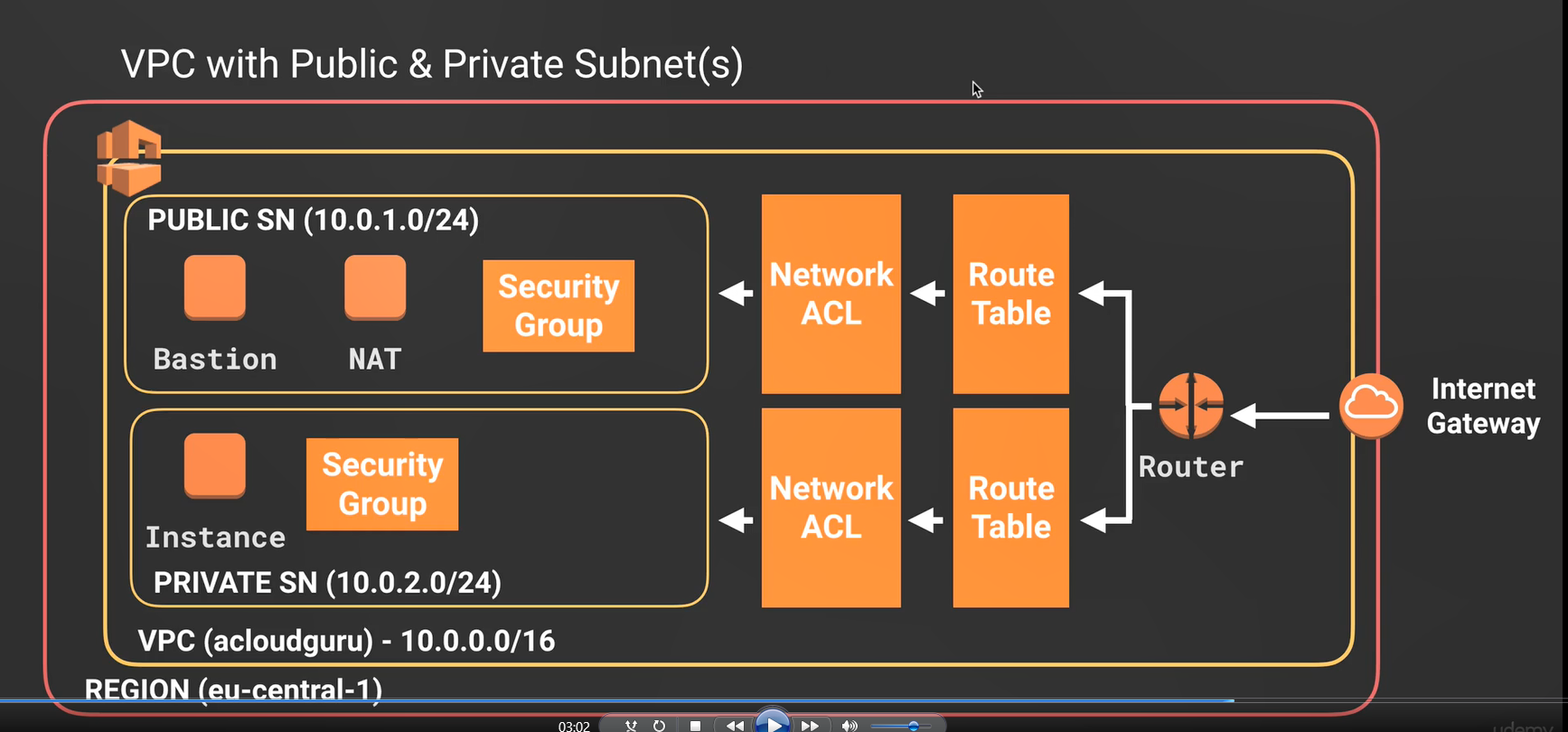
A bastion host is a specialized computer that is deliberately exposed on a public network. From a secured network perspective, it is the only node exposed to the outside world and is therefore very prone to attack. It is placed outside the firewall in single firewall systems or, if a system has two firewalls, it is often placed between the two firewalls or on the public side of a demilitarized zone (DMZ).

The most common examples of bastion hosts are mail, domain name system, Web and File Transfer Protocol (FTP) servers. Firewalls and routers can also become bastion hosts.

Amazon VPC enables you to launch AWS resources on a virtual private network that you have defined. The bastion host runs on an Amazon EC2 instance that is typically in a public subnet of your Amazon VPC. Linux instances are in a subnet that is not publicly accessible, and they are set up with a security group that allows SSH access from the security group attached to the underlying EC2 instance running the bastion host. Bastion host users connect to the bastion host to connect to the Linux instances, as illustrated in the following diagram.



**NATs and Bastion host**



By using NAT gateway or NAT instance private instance can access internet. In this picture Nat is an instance not gateway. By using bastion host we can connect the private instance from outside the VPC. So bastion host always resides behind security group where as NAT gateway always resides after security group. Bastion host is an instance but in aws we can apply auto-scaling group on bastion host to spread it over multiple availability zone for fault tolerance.

**What is VPC endpoint**

A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.

Example :

Suppose your ec2 server instance which is in a private subnet, want to access a aws instance but not the outside instance, then we can use VPC endpoints. To define a endpoint in a VPC we need to remove NAT gateway from the routing table and add a VPC endpoint in the routing table, internet gateway is not required in this case. To make specific acces, we can define IAM roles with the VPC endpoints.

**Some points to remember for VPC end points**

1. Our VPC can communicate through VPC endpoints to any aws service without using any internet.
2. Other aws service cannot connect to the vpc but our vpc can connect to other services like s3 or ec2 through endponts.
3. Add vpc endpoint id in routing table target and the destination is the aws service.

**What is virtual private gateway and customer gateway ?**

An Amazon VPC VPN connection links your data center (or network) to your Amazon Virtual Private Cloud (VPC). A customer gateway is the anchor on your side of that connection. It can be a physical or software appliance. The anchor on the AWS side of the VPN connection is called a virtual private gateway.

The following diagram shows your network, the customer gateway, the VPN connection that goes to the virtual private gateway, and the VPC. There are two lines between the customer gateway and virtual private gateway because the VPN connection consists of two tunnels to provide increased availability for the Amazon VPC service. If there's a device failure within AWS, your VPN connection automatically fails over to the second tunnel so that your access isn't interrupted.

You can create additional VPN connections to other VPCs using the same customer gateway device. You can reuse the same customer gateway IP address for each of those VPN connections.

When you create a VPN connection, the VPN tunnel comes up when traffic is generated from your side of the VPN connection. The virtual private gateway is not the initiator; your customer gateway must initiate the tunnels. AWS VPN endpoints support rekey and can start renegotiations when phase 1 is about to expire if the customer gateway hasn't sent any renegotiation traffic.


    Basic VPN diagram
   

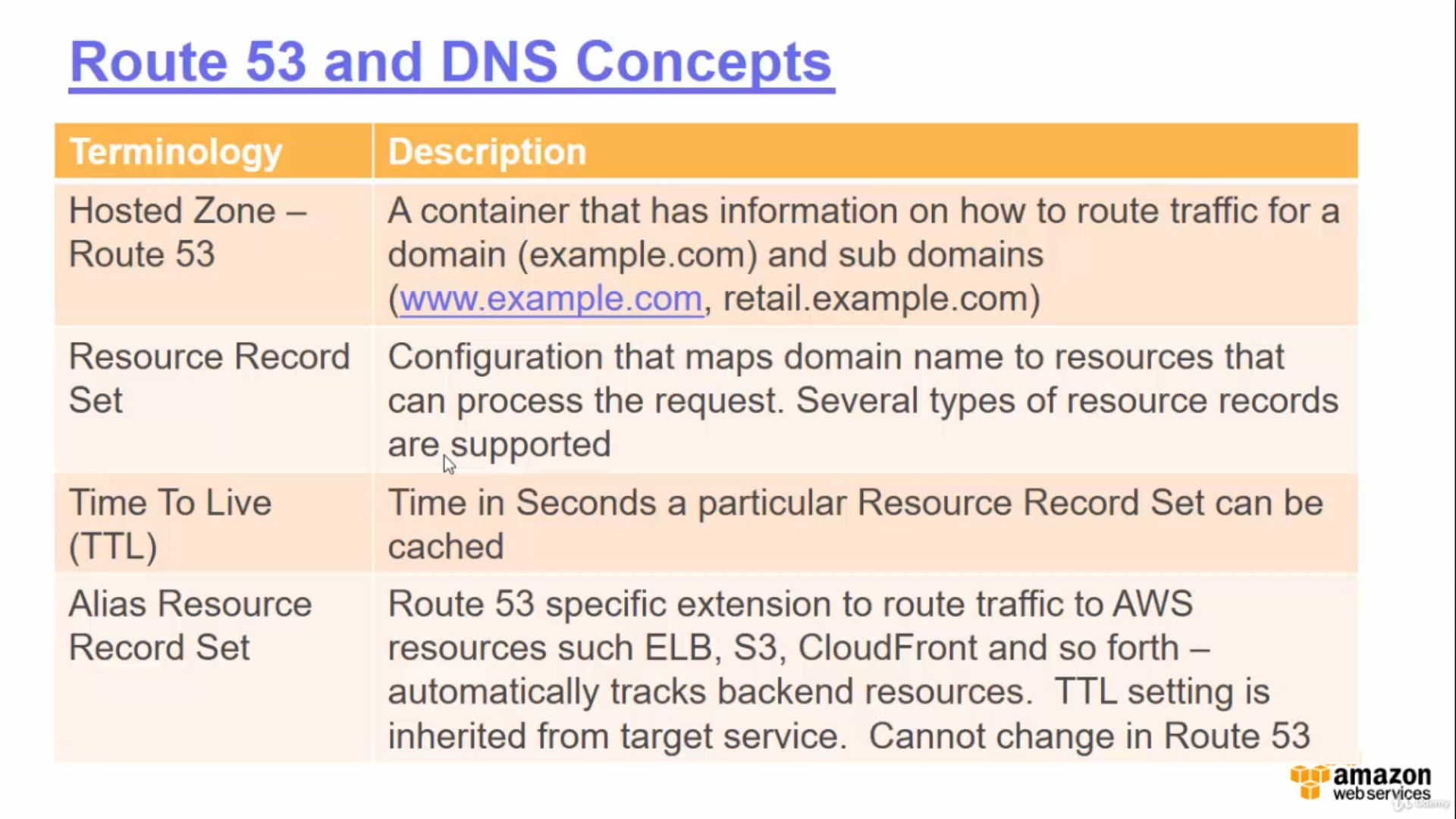
**Route 53**

**How DNS works ?**

* 1. Client hit the address yahoo.com, here the address can be splited into two part,

yahoo is the name and .com is the TLD or top level domain.

* 1. the request goes to the dns resolver first, when it receives the request first it will search its cache to find the ipaddress for yahoo.com, if it could not find then send request to the Root server, Root server knows where to redirect according to the TLD value, like .com or .org
  2. After receiving the request TLD server redirects the request to Name server relate to the value yahoo.
  3. Name server has the Ip address for yahoo.com it sends the ipaddress to the DNS resolver and DNS resolver sends it back to the client. Client then request the actual ipaddress.
  4. Here DNS resolver is our ISP or internet service provider, it always stores the ipaddress in its cache, so every time when a client requests , the entire flow is not going to happen.



**What is zone file ?**

Zone files are the way that name servers store information about the domains they know. The more zone files that a name server has, the more requests it will be able to answer authoritatively.

**What are the different record types ?**

DNS consists of a number of different record types, including but not limited to the following:

* A
* AAAA
* CNAME
* MX
* NS
* PTR
* SOA
* SPF
* TXT

**What is SOA records ?**

Start of authority record stores information about the name of the server that supplied the data for the zone.It supplies more information like

* The number of seconds a secondary name server should wait before checking for updates.
* The number of seconds secondary name server should wait before retrying a failed zone transfer.

**What is the default ttl time for dns record set in route 53 ?**

There is no default ttl ,the user has to mention the ttl.

**What is NS Records ?**

NS stands for the Name severe record and are used by TLD servers to direct traffic to the content DNS server which contains the authoritative DNS records.

**For a newly created hosted zone which records are created by default ?**

One NS record and one SOA record

**What is A record ?**

An “A” record is the fundamental type of DNS record and the “A” in A record stands for address. The A record is used by a computer to translate the name of the domain to ipaddress. For example <http://www.acloudguru.guru> might point to <http://123.10.10.80>

**What is hosted zone ?**

A *hosted zone* is a collection of resource record sets hosted by Amazon Route 53. Like a traditional DNS zone file, a hosted zone represents resource record sets that are managed together under a single domain name. Each hosted zone has its own metadata and configuration information.

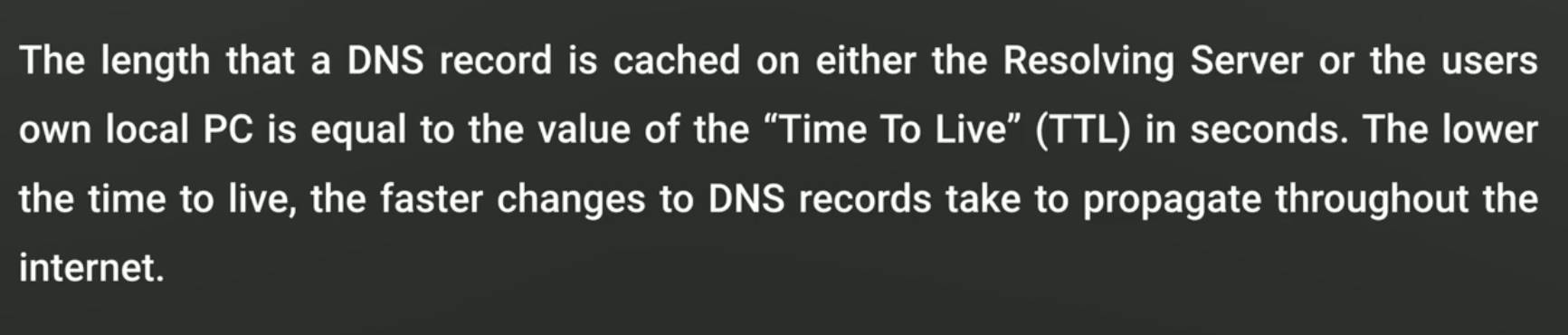
There are two types of hosted zones: private and public. A *private hosted zone* is a container that holds information about how you want to route traffic for a domain and its subdomains within one or more Amazon Virtual Private Clouds (Amazon VPCs). A *public hosted zone* is a container that holds information about how you want to route traffic on the Internet for a domain (for example, example.com) and its subdomains (for example, apex.example.com and acme.example.com).

The resource record sets contained in a hosted zone must share the same suffix. For example, the example.com hosted zone can contain resource record sets for the [www.example.com](http://www.example.com) and [www.aws.example.com](http://www.aws.example.com) subdomains, but it cannot contain resource record sets for a [www.example.ca](http://www.example.ca) subdomain.

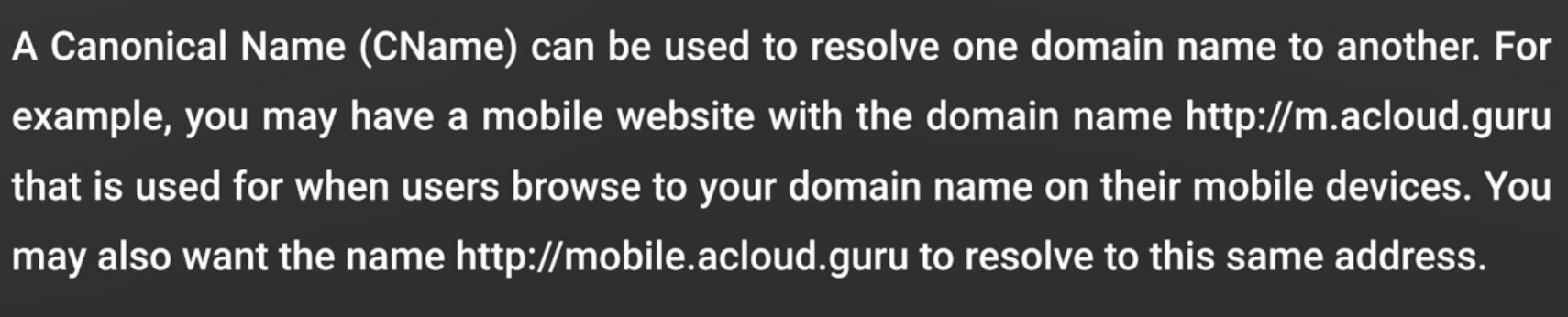
**Can we apply A record for load balancer ?**

Load balancer always recognized by a dns name not an ip address so load balancer we need to apply alias instead of A record.

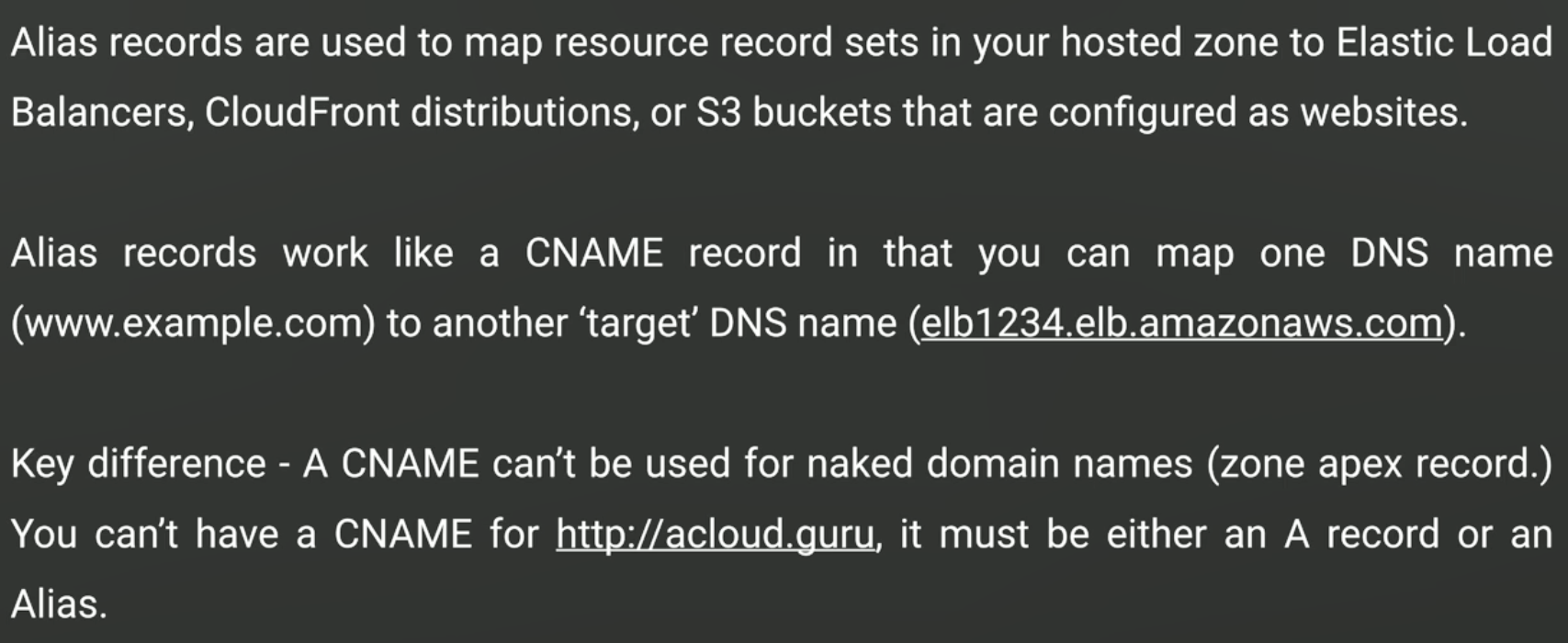
**What is TTL(Time to live)**



**What is CNAMES ?**



**What is alias record ?**



**When to use which record ?**

* Use an A record if you manage which IP addresses are assigned to a particular machine, or if the IP are fixed (this is the most common case).
* Use a CNAME record if you want to alias one name to another name, and you don’t need other records (such as MX records for emails) for the same name.
* Use an ALIAS record if you’re trying to alias the root domain (apex zone), or if you need other records for the same name.
* Use the URL record if you want the name to redirect (change address) instead of resolving to a destination.

**Different routing policy**

**Simple :** This is the default routing policy when you create a new resource. Use a simple routing policy when you have a single resource that performs a given function for your domain (for example, one web server that serves content for the example.com website)

**Weighted :** With weighted DNS, you can associate multiple resources (such as Amazon Elastic Compute Cloud [Amazon EC2] instances or Elastic Load Balancing load balancers) with a single DNS name.

Use the weighted routing policy when you have multiple resources that perform the same function (such as web servers that serve the same website), and you want Amazon Route 53 to route traffic to those resources in proportions that you specify. For example, you may use this for load balancing between different AWS regions or to test new versions of your website (you can send 10 percent of traffic to the test environment and 90 percent of traffic to the older version of your website).

**latency based :** *Latency-based routing* allows you to route your traffic based on the lowest network latency for your end user (for example, using the AWS region that will give them the fastest response time).

Use the latency routing policy when you have resources that perform the same function in multiple AWS Availability Zones or regions and you want Amazon Route 53 to respond to DNS queries using the resources that provide the best latency. For example, suppose you have Elastic Load Balancing load balancers in the U.S. West (Oregon) region and in the Asia Pacific (Singapore) region, and you created a latency resource record set in Amazon Route 53 for each load balancer. A user in London enters the name of your domain in a browser, and DNS routes the request to an Amazon Route 53 name server. Amazon Route 53 refers to its data on latency between London and the Singapore region and between London and the Oregon region. If latency is lower between London and the Oregon region, Amazon Route 53 responds to the user’s request with the IP address of your load balancer in Oregon. If latency is lower between London and the Singapore region, Amazon Route 53 responds with the IP address of your load balancer in Singapore

**Failover :** Use a failover routing policy to configure active-passive failover, in which one resource takes all the traffic when it’s available and the other resource takes all the traffic when the first resource isn’t available. Note that you can’t create failover resource record sets for private hosted zones.

For example, you might want your primary resource record set to be in U.S. West (N. California) and your secondary, Disaster Recovery (DR), resource(s) to be in U.S. East (N. Virginia). Amazon Route 53 will monitor the health of your primary resource endpoints using a health check.

A health check tells Amazon Route 53 how to send requests to the endpoint whose health you want to check: which protocol to use (HTTP, HTTPS, or TCP), which IP address and port to use, and, for HTTP/HTTPS health checks, a domain name and path.

After you have configured a health check, Amazon will monitor the health of your selected DNS endpoint. If your health check fails, then failover routing policies will be applied and your DNS will fail over to your DR site.

the application’s subdomain (for example, [www.domain.com](http://www.domain.com)) has an Amazon Route 53 alias record that points to prod.domain.com (as primary target) and fail.domain .com (as secondary target) using a failover routing policy. This ensures [www.domain.com](http://www.domain.com) routes to the production load balancers if at least one of them is healthy or the “fail whale” if all of them appear to be unhealthy.

**Geolocation :** *Geolocation routing* lets you choose where Amazon Route 53 will send your traffic based on the geographic location of your users (the location from which DNS queries originate). For example, you might want all queries from Europe to be routed to a fleet of Amazon EC2 instances that are specifically configured for your European customers, with local languages and pricing in Euros.

You can also use geolocation routing to restrict distribution of content to only the locations in which you have distribution rights. Another possible use is for balancing load across endpoints in a predictable, easy-to-manage way so that each user location is consistently routed to the same endpoint.

**Some objectives questions for Route53 ?**

* Which type of record is commonly used to route traffic to an IPv6 address?

An AAAA record

* Where do you register a domain name?

With InterNIC directly

* Which DNS record should you use to configure the transmission of email to your intended mail server ?

MX records

* Which DNS records are commonly used to stop email spoofing and spam ?

SPF record

* Which DNS record must all zones have by default

SOA

* Which type of DNS record should you use to resolve a domain name to another domain name?

A CNAME record

* Which port number is used to serve requests by DNS?

53

* Which protocol is primarily used by DNS to serve requests?

UDP

* Which protocol is used by DNS when response data size exceeds 512 bytes?

TCP

**With regards to VPC, what is the default maximum number of virtual private gateways allowed per region**

**5**

**What is cross stack reference in cloud formation ?**

Common resources can be managed using a separate stack. Other stacks can simply refer to the existing resources using cross-stack references. This allows independent teams to be responsible for their resources. When creating a template, you can indicate what resources are available for cross stack references by exporting those values (Export output field). Other stacks can use Fn::ImportValue function to import the value. Nested Stacks are used for common templates for creating the same type of resources across multiple stacks. For example: elastic load balancer required by each application.