**Networking and Content Delivery**

**Amazon VPC**

Q. What is Amazon Virtual Private Cloud?

Amazon VPC lets you provision a logically isolated section of the Amazon Web Services (AWS) cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address ranges, creation of subnets, and configuration of route tables and network gateways. You can also create a hardware Virtual Private Network (VPN) connection between your corporate datacenter and your VPC and leverage the AWS cloud as an extension of your corporate datacenter.

You can easily customize the network configuration for your Amazon VPC. For example, you can create a public-facing subnet for your web servers that have access to the Internet, and place your backend systems such as databases or application servers in a private-facing subnet with no Internet access. You can leverage multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet.

Q. What are the components of Amazon VPC?

Amazon VPC comprises a variety of objects that will be familiar to customers with existing networks:

* A Virtual Private Cloud: A logically isolated virtual network in the AWS cloud. You define a VPC’s IP address space from ranges you select.
* Subnet: A segment of a VPC’s IP address range where you can place groups of isolated resources.
* Internet Gateway: The Amazon VPC side of a connection to the public Internet.
* NAT Gateway: A highly available, managed Network Address Translation (NAT) service for your resources in a private subnet to access the Internet.
* Virtual private gateway: The Amazon VPC side of a VPN connection.
* Peering Connection: A peering connection enables you to route traffic via private IP addresses between two peered VPCs.
* VPC Endpoints: Enables private connectivity to services hosted in AWS, from within your VPC without using an Internet Gateway, VPN, Network Address Translation (NAT) devices, or firewall proxies.
* Egress-only Internet Gateway: A stateful gateway to provide egress only access for IPv6 traffic from the VPC to the Internet.

Q: Why should I use Amazon VPC?

Amazon VPC enables you to build a virtual network in the AWS cloud - no VPNs, hardware, or physical datacenters required. You can define your own network space, and control how your network and the Amazon EC2 resources inside your network are exposed to the Internet. You can also leverage the enhanced security options in Amazon VPC to provide more granular access to and from the Amazon EC2 instances in your virtual network.

Q. What are the different types of VPC endpoints available on Amazon VPC?

VPC endpoints enable you to privately connect your VPC to services hosted on AWS without requiring an Internet gateway, a NAT device, VPN, or firewall proxies. Endpoints are horizontally scalable and highly available virtual devices that allow communication between instances in your VPC and AWS services. Amazon VPC offers two different types of endpoints: gateway type endpoints and interface type endpoints.

Gateway type endpoints are available only for AWS services including S3 and DynamoDB. These endpoints will add an entry to your route table you selected and route the traffic to the supported services through Amazon’s private network.

Interface type endpoints provide private connectivity to services powered by PrivateLink, being AWS services, your own services or SaaS solutions, and supports connectivity over Direct Connect. More AWS and SaaS solutions will be supported by these endpoints in the future.

Q. What are the connectivity options for my Amazon VPC?

You may connect your Amazon VPC to:

* The internet (via an internet gateway)
* Your corporate data center using an AWS Site-to-Site VPN connection (via the virtual private gateway)
* Both the internet and your corporate data center (utilizing both an internet gateway and a virtual private gateway)
* Other AWS services (via internet gateway, NAT, virtual private gateway, or VPC endpoints)
* Other Amazon VPCs (via VPC peering connections)

Q. How do I connect my VPC to the Internet?

Amazon VPC supports the creation of an Internet gateway. This gateway enables Amazon EC2 instances in the VPC to directly access the Internet.

Q. Are there any bandwidth limitations for Internet gateways? Do I need to be concerned about its availability? Can it be a single point of failure?

No. An Internet gateway is horizontally-scaled, redundant, and highly available. It imposes no bandwidth constraints.

Q. How do instances in a VPC access the Internet?

You can use public IP addresses, including Elastic IP addresses (EIPs), to give instances in the VPC the ability to both directly communicate outbound to the Internet and to receive unsolicited inbound traffic from the Internet (e.g., web servers). You can also use the solutions in the next question.

Q. How do instances without public IP addresses access the Internet

Instances without public IP addresses can access the Internet in one of two ways:

1. Instances without public IP addresses can route their traffic through a NAT gateway or a NAT instance to access the Internet. These instances use the public IP address of the NAT gateway or NAT instance to traverse the Internet. The NAT gateway or NAT instance allows outbound communication but doesn’t allow machines on the Internet to initiate a connection to the privately addressed instances.
2. For VPCs with a hardware VPN connection or Direct Connect connection, instances can route their Internet traffic down the virtual private gateway to your existing datacenter. From there, it can access the Internet via your existing egress points and network security/monitoring devices.

Q. Does traffic go over the internet when two instances communicate using public IP addresses?

Traffic between two EC2 instances in the same AWS Region stays within the AWS network, even when it goes over public IP addresses.

Traffic between EC2 instances in different AWS Regions stays within the AWS network, if there is an Inter-Region VPC Peering connection between the VPCs where the two instances reside.

Traffic between EC2 instances in different AWS Regions where there is no Inter-Region VPC Peering connection between the VPCs where these instances reside, is not guaranteed to stay within the AWS network.

Q. How does an AWS Site-to-Site VPN connection work with Amazon VPC?

An AWS Site-to-Site VPN connection connects your VPC to your datacenter. Amazon supports Internet Protocol Security (IPSec) VPN connections. Data transferred between your VPC and datacenter routes over an encrypted VPN connection to help maintain the confidentiality and integrity of data in transit. An internet gateway is not required to establish an AWS Site-to-Site VPN connection.

**IP Addressing**

Q. What IP address ranges can I use within my Amazon VPC?

You can use any [IPv4](http://en.wikipedia.org/wiki/IPv4) address range, including [RFC 1918](https://tools.ietf.org/html/rfc1918) or publicly routable IP ranges, for the primary CIDR block. For the secondary CIDR blocks, certain [restrictions](http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html#add-cidr-block-restrictions) apply. Publicly routable IP blocks are only reachable via the Virtual Private Gateway and cannot be accessed over the Internet through the Internet gateway. AWS does not advertise customer-owned IP address blocks to the Internet. You can allocate an Amazon-provided IPv6 CIDR block to a VPC by calling the relevant API or via the AWS Management Console.

Q. How do I assign IP address ranges to Amazon VPCs?

You assign a single [Classless Internet Domain Routing (CIDR)](http://en.wikipedia.org/wiki/CIDR) IP address range as the primary CIDR block when you create a VPC and can add up to four (4) secondary CIDR blocks after creation of the VPC. Subnets within a VPC are addressed from these CIDR ranges by you. Please note that while you can create multiple VPCs with overlapping IP address ranges, doing so will prohibit you from connecting these VPCs to a common home network via the hardware VPN connection. For this reason we recommend using non-overlapping IP address ranges. You can allocate an Amazon-provided IPv6 CIDR block to your VPC.

Q. What IP address ranges are assigned to a default Amazon VPC?

Default VPCs are assigned a CIDR range of 172.31.0.0/16. Default subnets within a default VPC are assigned /20 netblocks within the VPC CIDR range.

Q. Can I advertise my VPC public IP address range to the internet and route the traffic through my datacenter, via the AWS Site-to-Site VPN, and to my Amazon VPC?

Yes, you can route traffic via the AWS Site-to-Site VPN connection and advertise the address range from your home network.

Q. Can I use my public IPv4 addresses in VPC and access them over the Internet?

Yes, you can bring your public IPv4 addresses into AWS VPC and statically allocate them to subnets and EC2 instances. To access these addresses over the Internet, you will have to advertise them to the Internet from your on-premises network. You will also have to route the traffic over these addresses between your VPC and on-premises network using AWS DX or AWS VPN connection. You can route the traffic from your VPC using the Virtual Private Gateway. Similarly, you can route the traffic from your on-premises network back to your VPC using your routers.

Q. How large of a VPC can I create?

Currently, Amazon VPC supports five (5) IP address ranges, one (1) primary and four (4) secondary for IPv4. Each of these ranges can be between /28 (in CIDR notation) and /16 in size. The IP address ranges of your VPC should not overlap with the IP address ranges of your existing network.

For IPv6, the VPC is a fixed size of /56 (in CIDR notation). A VPC can have both IPv4 and IPv6 CIDR blocks associated to it.

Q. Can I change the size of a VPC?

Yes. You can expand your existing VPC by adding four (4) secondary IPv4 IP ranges (CIDRs) to your VPC. You can shrink your VPC by deleting the secondary CIDR blocks you have added to your VPC. You cannot however change the size of the IPv6 address range of your VPC.

Q. How many subnets can I create per VPC?

Currently you can create 200 subnets per VPC. If you would like to create more, please [submit a case at the support center](https://aws.amazon.com/contact-us/vpc-request/).

Q. Is there a limit on how large or small a subnet can be?

The minimum size of a subnet is a /28 (or 14 IP addresses.) for IPv4. Subnets cannot be larger than the VPC in which they are created.

For IPv6, the subnet size is fixed to be a /64. Only one IPv6 CIDR block can be allocated to a subnet.

Q. Can I use all the IP addresses that I assign to a subnet?

No. Amazon reserves the first four (4) IP addresses and the last one (1) IP address of every subnet for IP networking purposes.

Q. How do I assign private IP addresses to Amazon EC2 instances within a VPC?

When you launch an Amazon EC2 instance within a VPC, you may optionally specify the primary private IP address for the instance. If you do not specify the primary private IP address, AWS automatically addresses it from the IP address range you assign to that subnet. You can assign secondary private IP addresses when you launch an instance, when you create an Elastic Network Interface, or any time after the instance has been launched or the interface has been created.

Q. Can I change the private IP addresses of an Amazon EC2 instance while it is running and/or stopped within a VPC?

Primary private IP addresses are retained for the instance's or interface's lifetime. Secondary private IP addresses can be assigned, unassigned, or moved between interfaces or instances at any time.

Q. If an Amazon EC2 instance is stopped within a VPC, can I launch another instance with the same IP address in the same VPC?

No. An IP address assigned to a running instance can only be used again by another instance once that original running instance is in a “terminated” state.

Q. Can I assign any IP address to an instance?

You can assign any IP address to your instance as long as it is:

* Part of the associated subnet's IP address range
* Not reserved by Amazon for IP networking purposes
* Not currently assigned to another interface

Q. Can I assign multiple IP addresses to an instance?

Yes. You can assign one or more secondary private IP addresses to an Elastic Network Interface or an EC2 instance in Amazon VPC. The number of secondary private IP addresses you can assign depends on the instance type.

Q. How do I secure Amazon EC2 instances running within my VPC?

Amazon EC2 security groups can be used to help secure instances within an Amazon VPC. Security groups in a VPC enable you to specify both inbound and outbound network traffic that is allowed to or from each Amazon EC2 instance. Traffic which is not explicitly allowed to or from an instance is automatically denied.

In addition to security groups, network traffic entering and exiting each subnet can be allowed or denied via network Access Control Lists (ACLs).

Q. What are the differences between security groups in a VPC and network ACLs in a VPC?

Security groups in a VPC specify which traffic is allowed to or from an Amazon EC2 instance. Network ACLs operate at the subnet level and evaluate traffic entering and exiting a subnet. Network ACLs can be used to set both Allow and Deny rules. Network ACLs do not filter traffic between instances in the same subnet. In addition, network ACLs perform stateless filtering while security groups perform stateful filtering.

Q. Can Amazon EC2 instances within a VPC communicate with Amazon EC2 instances not within a VPC?

Yes. If an Internet gateway has been configured, Amazon VPC traffic bound for Amazon EC2 instances not within a VPC traverses the Internet gateway and then enters the public AWS network to reach the EC2 instance. If an Internet gateway has not been configured, or if the instance is in a subnet configured to route through the virtual private gateway, the traffic traverses the VPN connection, egresses from your datacenter, and then re-enters the public AWS network.

Q. Can Amazon EC2 instances within a VPC communicate with Amazon S3?

Yes. There are multiple options for your resources within a VPC to communicate with Amazon S3. You can use VPC Endpoint for S3, which makes sure all traffic remains within Amazon's network and enables you to apply additional access policies to your Amazon S3 traffic. You can use an Internet gateway to enable Internet access from your VPC and instances in the VPC can communicate with Amazon S3. You can also make all traffic to Amazon S3 traverse the Direct Connect or VPN connection, egress from your datacenter, and then re-enter the public AWS network.

Q. Can I monitor the network traffic in my VPC?

Yes. You can use the Amazon VPC Flow Logs feature to monitor the network traffic in your VPC.

Q. Can a VPC span multiple Availability Zones?

Yes.

Q. Can a subnet span Availability Zones?

No. A subnet must reside within a single Availability Zone.

Q. How many Amazon EC2 instances can I use within a VPC?

You can run any number of Amazon EC2 instances within a VPC, so long as your VPC is appropriately sized to have an IP address assigned to each instance. You are initially limited to launching 20 Amazon EC2 instances at any one time and a maximum VPC size of /16 (65,536 IPs).

Q. What is a default VPC?

A default VPC is a logically isolated virtual network in the AWS cloud that is automatically created for your AWS account the first time you provision Amazon EC2 resources. When you launch an instance without specifying a subnet-ID, your instance will be launched in your default VPC.

Q. What are the benefits of a default VPC?

When you launch resources in a default VPC, you can benefit from the advanced networking functionalities of Amazon VPC (EC2-VPC) with the ease of use of Amazon EC2 (EC2-Classic). You can enjoy features such as changing security group membership on the fly, security group egress filtering, multiple IP addresses, and multiple network interfaces without having to explicitly create a VPC and launch instances in the VPC.

Q. What is ClassicLink?

Amazon Virtual Private Cloud (VPC) ClassicLink allows EC2 instances in the EC2-Classic platform to communicate with instances in a VPC using private IP addresses. To use ClassicLink, enable it for a VPC in your account, and associate a Security Group from that VPC with an instance in EC2-Classic. All the rules of your VPC Security Group will apply to communications between instances in EC2-Classic and instances in the VPC.

Q. What is the Bring Your Own IP feature?

Bring Your Own IP (BYOIP) enables customers to move all or part of their existing publicly routable IPv4 address space to AWS for use with their AWS resources. Customers will continue to own the IP range, however, AWS will take over its advertisement on the internet. Customers can create Elastic IPs from the IP space they bring to AWS and use them with EC2 instances, NAT Gateways, and Network Load Balancers. Customers will continue to have access to Amazon-supplied IPs and can choose to use BYOIP Elastic IPs, Amazon-supplied IPs, or both.

Q. Why should I use BYOIP?

You may want to bring your own IP addresses to AWS for the following reasons:

IP Reputation: Many customers consider the reputation of their IP addresses to be a strategic asset and want to use those IPs on AWS with their resources. For example, customers who maintain services such as outbound e-mail MTA and have high reputation IPs, can now bring over their IP space and successfully maintain their existing sending success rate.

Customer whitelisting: BYOIP also enables customers to move workloads that rely on IP address whitelisting to AWS without the need to re-establish the whitelists with new IP addresses.

Hardcoded dependencies: Several customers have IPs hardcoded in devices or have taken architectural dependencies on their IPs. BYOIP enables such customers hassle free migration to AWS.

Regulation and compliance: Many customers are required to use certain IPs because of regulation and compliance reasons. They too are unlocked by BYOIP.

**Amazon CloudFront**

Q. What is Amazon CloudFront?

Amazon CloudFront is a web service that gives businesses and web application developers an easy and cost effective way to distribute content with low latency and high data transfer speeds. Like other AWS services, Amazon CloudFront is a self-service, pay-per-use offering, requiring no long term commitments or minimum fees. With CloudFront, your files are delivered to end-users using a global network of edge locations.

Q. What can I do with Amazon CloudFront?

Amazon CloudFront provides a simple API that lets you:

* Distribute content with low latency and high data transfer rates by serving requests using a network of edge locations around the world.
* Get started without negotiating contracts and minimum commitments.

Q. What can I do with Amazon CloudFront?

Amazon CloudFront provides a simple API that lets you:

* Distribute content with low latency and high data transfer rates by serving requests using a network of edge locations around the world.
* Get started without negotiating contracts and minimum commitments.

Q. How do I use Amazon CloudFront?

To use Amazon CloudFront, you:

* For static files, store the definitive versions of your files in one or more origin servers. These could be Amazon S3 buckets. For your dynamically generated content that is personalized or customized, you can use Amazon EC2 – or any other web server – as the origin server. These origin servers will store or generate your content that will be distributed through Amazon CloudFront.
* Register your origin servers with Amazon CloudFront through a simple API call. This call will return a CloudFront.net domain name that you can use to distribute content from your origin servers via the Amazon CloudFront service. For instance, you can register the Amazon S3 bucket “bucketname.s3.amazonaws.com” as the origin for all your static content and an Amazon EC2 instance “dynamic.myoriginserver.com” for all your dynamic content. Then, using the API or the AWS Management Console, you can create an Amazon CloudFront distribution that might return “abc123.cloudfront.net” as the distribution domain name.
* Include the cloudfront.net domain name, or a CNAME alias that you create, in your web application, media player, or website. Each request made using the cloudfront.net domain name (or the CNAME you set-up) is routed to the edge location best suited to deliver the content with the highest performance. The edge location will attempt to serve the request with a local copy of the file. If a local copy is not available, Amazon CloudFront will get a copy from the origin. This copy is then available at that edge location for future requests.

Q. How does Amazon CloudFront provide higher performance?

Amazon CloudFront employs a global network of edge locations and regional edge caches that cache copies of your content close to your viewers. Amazon CloudFront ensures that end-user requests are served by the closest edge location. As a result, viewer requests travel a short distance, improving performance for your viewers. For files not cached at the edge locations and the regional edge caches, Amazon CloudFront keeps persistent connections with your origin servers so that those files can be fetched from the origin servers as quickly as possible. Finally, Amazon CloudFront uses additional optimizations – e.g. wider TCP initial congestion window – to provide higher performance while delivering your content to viewers.

Q. How does Amazon CloudFront speed up my entire website?

Amazon CloudFront uses standard cache control headers you set on your files to identify static and dynamic content. Delivering all your content using a single Amazon CloudFront distribution helps you make sure that performance optimizations are applied to your entire website or web application. When using AWS origins, you benefit from improved performance, reliability, and ease of use as a result of AWS’s ability to track and adjust origin routes, monitor system health, respond quickly when any issues occur, and the integration of Amazon CloudFront with other AWS services. You also benefit from using different origins for different types of content on a single site – e.g. Amazon S3 for static objects, Amazon EC2 for dynamic content, and custom origins for third-party content – paying only for what you use.

Q. How is Amazon CloudFront different from traditional content delivery solutions?

Amazon CloudFront lets you quickly obtain the benefits of high performance content delivery without negotiated contracts or high prices. Amazon CloudFront gives all developers access to inexpensive, pay-as-you-go pricing – with a self-service model. Developers also benefit from tight integration with other Amazon Web Services. The solution is simple to use with Amazon S3, Amazon EC2, and Elastic Load Balancing as origin servers, giving developers a powerful combination of durable storage and high performance delivery. Amazon CloudFront also integrates with Amazon Route 53 and AWS CloudFormation for further performance benefits and ease of configuration.

Q. What types of content does Amazon CloudFront support?

Amazon CloudFront supports content that can be sent using the HTTP or WebSocket protocols. This includes dynamic web pages and applications, such as HTML or PHP pages or WebSocket-based applications, and any popular static files that are a part of your web application, such as website images, audio, video, media files or software downloads. Amazon CloudFront also supports delivery of live or on-demand media streaming over HTTP.

Q. What is CloudFront Regional Edge Cache?

CloudFront delivers your content through a worldwide network of data centers called edge locations. The regional edge caches are located between your origin web server and the global edge locations that serve content directly to your viewers. This helps improve performance for your viewers while lowering the operational burden and cost of scaling your origin resources.

Q. How does regional edge cache work?

Amazon CloudFront has added several regional edge cache locations globally, at close proximity to your viewers. They are located between your origin webserver and the global edge locations that serve content directly to your viewers. As objects become less popular, individual edge locations may remove those objects to make room for more popular content. Regional Edge Caches have a larger cache width than any individual edge location, so objects remain in the cache longer at the nearest regional edge caches. This helps keep more of your content closer to your viewers, reducing the need for CloudFront to go back to your origin webserver and improving overall performance for viewers. For example, CloudFront edge locations in Europe now go to the regional edge cache in Frankfurt to fetch an object before going back to your origin webserver. Regional edge cache locations are currently used only for requests that need to go back to a custom origin; i.e. requests to S3 origins will skip regional edge cache locations.

Q. Can I choose to serve content (or not serve content) to specified countries?

Yes, the Geo Restriction feature lets you specify a list of countries in which your users can access your content. Alternatively, you can specify the countries in which your users cannot access your content. In both cases, CloudFront responds to a request from a viewer in a restricted country with an HTTP status code 403 (Forbidden).

Q. What is Field-Level Encryption?

Field-Level Encryption is a feature of CloudFront that allows you to securely upload user-submitted data such as credit card numbers to your origin servers. Using this functionality, you can further encrypt sensitive data in an HTTPS form using field-specific encryption keys (which you supply) before a PUT/ POST request is forwarded to your origin. This ensures that sensitive data can only be decrypted and viewed by certain components or services in your application stack.

Q. How can I safeguard my web applications delivered via CloudFront from DDoS attacks?

As an AWS customer, you get [AWS Shield Standard](https://aws.amazon.com/shield/) at no additional cost. AWS Shield is a managed service that provides protection against DDoS attacks for web applications running on AWS. AWS Shield Standard provides protection for all AWS customers against common and most frequently occurring Infrastructure (layer 3 and 4) attacks like SYN/UDP Floods, Reflection attacks, and others to support high availability of your applications on AWS.

[AWS Shield Advanced](https://aws.amazon.com/shield/) is an optional paid service available to AWS Business Support and AWS Enterprise Support customers. AWS Shield Advanced provides additional protections against larger and more sophisticated attacks for your applications running on Elastic Load Balancing (ELB), Amazon CloudFront and Route 53.

Q. How can I protect my web applications delivered via CloudFront?

You can integrate your CloudFront distribution with [AWS WAF](https://aws.amazon.com/waf/), a web application firewall that helps protect web applications from attacks by allowing you to configure rules based on IP addresses, HTTP headers, and custom URI strings. Using these rules, AWS WAF can block, allow, or monitor (count) web requests for your web application.

Q. Can I choose to only serve content from less expensive Amazon CloudFront regions?

Yes, "Price Classes" provides you an option to lower the prices you pay to deliver content out of Amazon CloudFront. By default, Amazon CloudFront minimizes end user latency by delivering content from its entire global network of edge locations. However, because we charge more where our costs are higher, this means that you pay more to deliver your content with low latency to end-users in some locations. Price Classes let you reduce your delivery prices by excluding Amazon CloudFront’s more expensive edge locations from your Amazon CloudFront distribution. In these cases, Amazon CloudFront will deliver your content from edge locations within the locations in the price class you selected and charge you the data transfer and request pricing from the actual location where the content was delivered.

**Amazon API Gateway**

Q: What is Amazon API Gateway?

Amazon API Gateway is a fully managed service that makes it easy for developers to publish, maintain, monitor, and secure APIs at any scale. With a few clicks in the AWS Management Console, you can create an API that acts as a “front door” for applications to access data, business logic, or functionality from your back-end services, such as applications running on Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Container Service (Amazon ECS) or AWS Elastic Beanstalk, code running on AWS Lambda, or any web application. Amazon API Gateway handles all of the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, authorization and access control, monitoring, and API version management. Amazon API Gateway has no minimum fees or startup costs. For REST APIs, you pay only for the API calls you receive and the amount of data transferred out. For WebSocket APIs, you pay only for messages sent and received and for the time a user/device is connected to the WebSocket API.

Q: Why use Amazon API Gateway?

Amazon API Gateway provides developers with a simple, flexible, fully managed, pay-as-you-go service that handles all aspects of creating and operating robust APIs for application back ends. With API Gateway, you can launch new services faster and with reduced investment so you can focus on building your core business services. API Gateway was built to help you with several aspects of creating and managing APIs:

1) Metering. API Gateway helps you define plans that meter and restrict third-party developer access to your APIs. You can define a set of plans, configure throttling, and quota limits on a per API key basis. API Gateway automatically meters traffic to your APIs and lets you extract utilization data for each API key.

2) Security. API Gateway provides you with multiple tools to authorize access to your APIs and control service operation access. API Gateway allows you to leverage AWS administration and security tools, such as AWS Identity and Access Management (IAM) and Amazon Cognito, to authorize access to your APIs. API Gateway can verify signed API calls on your behalf using the same methodology AWS uses for its own APIs. Using custom authorizers written as AWS Lambda functions, API Gateway can also help you verify incoming bearer tokens, removing authorization concerns from your backend code.

3) Resiliency. API Gateway helps you manage traffic with throttling so that backend operations can withstand traffic spikes. API Gateway also helps you improve the performance of your APIs and the latency your end users experience by caching the output of API calls to avoid calling your backend every time.

4) Operations Monitoring. After an API is published and in use, API Gateway provides you with a metrics dashboard to monitor calls to your services. The API Gateway dashboard, through integration with Amazon CloudWatch, provides you with backend performance metrics covering API calls, latency data and error rates. You can enable detailed metrics for each method in your APIs and also receive error, access or debug logs in CloudWatch Logs.

5) Lifecycle Management. After an API has been published, you often need to build and test new versions that enhance or add new functionality. API Gateway lets you operate multiple API versions and multiple stages for each version simultaneously so that existing applications can continue to call previous versions after new API versions are published.

6) Designed for Developers. API Gateway allows you to quickly create APIs and assign static content for their responses to reduce cross-team development effort and time-to-market for your applications. Teams who depend on your APIs can begin development while you build your backend processes.

7) Real-Time Two-Way Communication. Build real-time two-way communication applications such as chat apps, streaming dashboards, and notifications without having to run or manage any servers. API Gateway maintains a persistent connection between connected users and enables message transfer between them.

Q: What API types are supported by Amazon API Gateway?

Amazon API Gateway supports creating REST APIs and WebSocket APIs.

REST API: In Amazon API Gateway, a REST API is a group of resources and methods, or endpoints. REST APIs can be deployed to different stages and cloned to new versions.

WebSocket API: In Amazon API Gateway, a WebSocket API maintains a persistent connection between connected clients to enable real-time message communication. With WebSocket APIs in API Gateway, you can define backend integrations with AWS Lambda functions, Amazon Kinesis, or any HTTP endpoint to be invoked when messages are received from the connected clients.

Q: With what backends can Amazon API Gateway communicate?

Amazon API Gateway can execute AWS Lambda functions in your account, start AWS Step Functions state machines, or call HTTP endpoints hosted on AWS Elastic Beanstalk, Amazon EC2, and also non-AWS hosted HTTP based operations that are accessible via the public Internet.API Gateway also allows you to specify a mapping template to generate static content to be returned, helping you mock your APIs before the backend is ready. You can also integrate API Gateway with other AWS services directly – for example, you could expose an API method in API Gateway that sends data directly to Amazon Kinesis.

Q: Can I use AWS CloudTrail with Amazon API Gateway?

Yes. Amazon API Gateway is integrated with [AWS CloudTrail](https://aws.amazon.com/cloudtrail/) to give you a full auditable history of the changes to your REST APIs. All API calls made to the Amazon API Gateway APIs to create, modify, delete, or deploy REST APIs are logged to CloudTrail in your AWS account.

Q: How does Amazon API Gateway work with an Amazon Virtual Private Cloud (Amazon VPC)?

In Amazon API Gateway, you can proxy requests to backend HTTP/HTTPS resources running in your Amazon VPC by setting up [Private Integrations](https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-private-integration.html) using VPC Links. Client-side SSL certificates in Amazon API Gateway can be used to verify that requests to your backend systems were sent by API Gateway using the public key of the certificate. You can also create Private APIs in Amazon API Gateway which can only be accessible by resources within your Amazon VPC through Amazon VPC Endpoints.

Q: Can I restrict access to private APIs to a specific Amazon VPC or VPC endpoint?

Yes, you can apply a Resource Policy to an API to restrict access to a specific Amazon VPC or VPC endpoint. You can also give an Amazon VPC or VPC endpoint from a different account access to the Private API using a Resource Policy.

Q: How can I monitor my Amazon API Gateway APIs?

Amazon API Gateway logs API calls, latency, and error rates to Amazon CloudWatch in your AWS account. The metrics are also available through the Amazon API Gateway console in a REST API dashboard. API Gateway also meters utilization by third-party developers, the data is available in the API Gateway console and through the APIs.

Q: Can I set up alarms on the Amazon API Gateway metrics?

Yes, Amazon API Gateway sends logging information and metrics to Amazon CloudWatch. You can utilize the Amazon CloudWatch console to set up custom alarms.

Q: How does throttling help me?

Throttling ensures that API traffic is controlled to help your backend services maintain performance and availability.

Q: Does Amazon API Gateway provide API result caching?

Yes. You can add caching to API calls by provisioning an API Gateway cache and specifying its size in gigabytes. The cache is provisioned for a specific stage of your APIs. This improves performance and reduces the traffic sent to your back end. Cache settings allow you to control the way the cache key is built and the time-to-live (TTL) of the data stored for each method. API Gateway also exposes management APIs that help you invalidate the cache for each stage. Caching is available for REST APIs in API Gateway.

Q: What happens if a large number of end users try to invoke my API simultaneously?

If caching is not enabled and throttling limits have not been applied, then all requests will pass through to your backend service until the account level throttling limits are reached. If throttling limits are in place, then Amazon API Gateway will shed the necessary amount of requests and send only the defined limit to your back-end service. If a cache is configured, then Amazon API Gateway will return a cached response for duplicate requests for a customizable time, but only if under configured throttling limits. This balance between the backend and client ensures optimal performance of the APIs for the applications that it supports. Requests that are throttled will be automatically retried by the client-side SDKs generated by Amazon API Gateway. By default, Amazon API Gateway does not set any cache on your API methods.

Q: How do APIs scale?

Amazon API Gateway acts as a proxy to the backend operations that you have configured. Amazon API Gateway will automatically scale to handle the amount of traffic your API receives. Amazon API Gateway does not arbitrarily limit or throttle invocations to your backend operations and all requests that are not intercepted by throttling and caching settings in the Amazon API Gateway console are sent to your backend operations.

**Amazon Route 53**

Q. What is Amazon Route 53?

Amazon Route 53 provides highly available and scalable Domain Name System (DNS), domain name registration, and health-checking web services. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names like example.com into the numeric IP addresses, such as 192.0.2.1, that computers use to connect to each other. You can combine your DNS with health-checking services to route traffic to healthy endpoints or to independently monitor and/or alarm on endpoints. You can also purchase and manage domain names such as example.com and automatically configure DNS settings for your domains. Route 53 effectively connects user requests to infrastructure running in AWS – such as Amazon EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets – and can also be used to route users to infrastructure outside of AWS.

Q. What is a Domain Name System (DNS) Service?

[DNS](https://aws.amazon.com/route53/what-is-dns/) is a globally distributed service that translates human readable names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other. The Internet’s DNS system works much like a phone book by managing the mapping between names and numbers. For DNS, the names are domain names (www.example.com) that are easy for people to remember and the numbers are IP addresses (192.0.2.1) that specify the location of computers on the Internet. DNS servers translate requests for names into IP addresses, controlling which server an end user will reach when they type a domain name into their web browser. These requests are called "queries."

Q. What can I do with Amazon Route 53?

With Amazon Route 53, you can create and manage your public DNS records. Like a phone book, Route 53 lets you manage the IP addresses listed for your domain names in the Internet’s DNS phone book. Route 53 also answers requests to translate specific domain names like into their corresponding IP addresses like 192.0.2.1. You can use Route 53 to create DNS records for a new domain or transfer DNS records for an existing domain. The simple, standards-based REST API for Route 53 allows you to easily create, update and manage DNS records. Route 53 additionally offers health checks to monitor the health and performance of your application as well as your web servers and other resources. You can also register new domain names or transfer in existing domain names to be managed by Route 53.

Q. How does Amazon Route 53 provide high availability and low latency?

Route 53 is built using AWS’s highly available and reliable infrastructure. The globally distributed nature of our DNS servers helps ensure a consistent ability to route your end users to your application by circumventing any internet or network related issues. Route 53 is designed to provide the level of dependability required by important applications. Using a global anycast network of DNS servers around the world, Route 53 is designed to automatically answer queries from the optimal location depending on network conditions. As a result, the service offers low query latency for your end users.

To provide you with a highly available service, each Amazon Route 53 hosted zone is served by its own set of virtual DNS servers. The DNS server names for each hosted zone are thus assigned by the system when that hosted zone is created.

Q. What is the difference between a Domain and a Hosted Zone?

A domain is a general DNS concept. Domain names are easily recognizable names for numerically addressed Internet resources. For example, amazon.com is a domain. A hosted zone is an Amazon Route 53 concept. A hosted zone is analogous to a traditional DNS zone file; it represents a collection of records that can be managed together, belonging to a single parent domain name. All resource record sets within a hosted zone must have the hosted zone’s domain name as a suffix. For example, the amazon.com hosted zone may contain records named www.amazon.com, and www.aws.amazon.com, but not a record named www.amazon.ca. You can use the Route 53 Management Console or API to create, inspect, modify, and delete hosted zones. You can also use the Management Console or API to register new domain names and transfer existing domain names into Route 53’s management.

Q. What is the price of Amazon Route 53?

Amazon Route 53 charges are based on actual usage of the service for Hosted Zones, Queries, Health Checks, and Domain Names.

Q. I have subscribed for Amazon Route 53 but when I try to use the service it says "The AWS Access Key ID needs a subscription for the service.", Why ?

When you sign up for a new AWS service, it can take up to 24 hours in some cases to complete activation, during which time you cannot sign up for the service again. If you've been waiting longer than 24 hours without receiving an email confirming activation, this could indicate a problem with your account or the authorization of your payment details.

Q. Does Amazon Route 53 offer a Service Level Agreement (SLA)?

Yes. The Amazon Route 53 SLA provides for a service credit if a customer’s monthly uptime percentage is below our service commitment in any billing cycle.

Q. Does Amazon Route 53 provide query logging capability?

You can configure Amazon Route 53 to log information about the queries that Amazon Route 53 receives including date-time stamp, domain name, query type, location etc. When you configure query logging, Amazon Route 53 starts to send logs to CloudWatch Logs.

Q. Does Amazon Route 53 also provide website hosting?

No. Amazon Route 53 is an authoritative DNS service and does not provide [website hosting](https://aws.amazon.com/websites/). However, you can use Amazon Simple Storage Service (Amazon S3) to host a static website. To host a dynamic website or other web applications, you can use Amazon Elastic Compute Cloud (Amazon EC2), which provides flexibility, control, and significant cost savings over traditional [web hosting](https://aws.amazon.com/websites/) solutions. For both static and dynamic websites, you can provide low latency delivery to your global end users with Amazon CloudFront.

Q. How quickly will changes I make to my DNS settings on Amazon Route 53 propagate globally?

Amazon Route 53 is designed to propagate updates you make to your DNS records to its world-wide network of authoritative DNS servers within 60 seconds under normal conditions. A change is successfully propagated world-wide when the API call returns an INSYNC status listing.

Note that caching DNS resolvers are outside the control of the Amazon Route 53 service and will cache your resource record sets according to their time to live (TTL). The INSYNC or PENDING status of a change refers only to the state of Route 53’s authoritative DNS servers.

Q. Can I point my zone apex (example.com versus www.example.com) at my Elastic Load Balancer?

Yes. Amazon Route 53 offers a special type of record called an 'Alias' record that lets you map your zone apex (example.com) DNS name to the DNS name for your ELB load balancer (such as my-loadbalancer-1234567890.us-west-2.elb.amazonaws.com). IP addresses associated with load balancers can change at any time due to scaling up, scaling down, or software updates. Route 53 responds to each request for an Alias record with one or more IP addresses for the load balancer. Route 53 supports alias records for three types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. There is no additional charge for queries to Alias records that are mapped to AWS ELB load balancers. These queries are listed as “Intra-AWS-DNS-Queries” on the Amazon Route 53 usage report.

Q. Can I point my zone apex (example.com versus www.example.com) at my website hosted on Amazon S3?

Yes. Amazon Route 53 offers a special type of record called an ‘Alias’ record that lets you map your zone apex (example.com) DNS name to your Amazon S3 website bucket (i.e. example.com.s3-website-us-west-2.amazonaws.com). IP addresses associated with Amazon S3 website endpoints can change at any time due to scaling up, scaling down, or software updates. Route 53 responds to each request for an Alias record with one IP address for the bucket. Route 53 doesn't charge for queries to Alias records that are mapped to an S3 bucket that is configured as a website. These queries are listed as “Intra-AWS-DNS-Queries” on the Amazon Route 53 usage report.

Q. Can I point my zone apex (example.com versus www.example.com) at my Amazon CloudFront distribution?

Yes. Amazon Route 53 offers a special type of record called an ‘Alias’ record that lets you map your zone apex (example.com) DNS name to your Amazon CloudFront distribution (for example, d123.cloudfront.net). IP addresses associated with Amazon CloudFront endpoints vary based on your end user’s location (in order to direct the end user to the nearest CloudFront edge location) and can change at any time due to scaling up, scaling down, or software updates. Route 53 responds to each request for an Alias record with the IP address(es) for the distribution. Route 53 doesn't charge for queries to Alias records that are mapped to a CloudFront distribution. These queries are listed as “Intra-AWS-DNS-Queries” on the Amazon Route 53 usage report.

Q. Can I point my zone apex (example.com versus www.example.com) at my AWS Elastic Beanstalk environment?

Yes. Amazon Route 53 offers a special type of record called an ‘Alias’ record that lets you map your zone apex (example.com) DNS name to your AWS Elastic Beanstalk DNS name (i.e. example.elasticbeanstalk.com). IP addresses associated with AWS Elastic Beanstalk environments can change at any time due to scaling up, scaling down, or software updates. Route 53 responds to each request for an Alias record with one or more IP addresses for the environment. Queries to Alias records that are mapped to AWS Elastic Beanstalk environments are free. These queries are listed as “Intra-AWS-DNS-Queries” on the Amazon Route 53 usage report.

Q. Can I point my zone apex (example.com versus www.example.com) at my Amazon API Gateway?

Yes. Amazon Route 53 offers a special type of record called an ‘Alias’ record that lets you map your zone apex (example.com) DNS name to your Amazon API Gateway DNS name (i.e. api-id.execute-api.region.amazonaws.com/stage). IP addresses associated with Amazon API Gateway can change at any time due to scaling up, scaling down, or software updates. Route 53 responds to each request for an Alias record with one or more IP addresses for the API Gateway. There is no additional charge for queries to Alias records that are mapped to Amazon API Gateways. These queries are listed as “Intra-AWS-DNS-Queries” on the Route 53 usage report.

Q. Can I point my zone apex (example.com versus www.example.com) at my Amazon VPC endpoint?

Yes. Amazon Route 53 offers a special type of record called an ‘Alias’ record that lets you map your zone apex (example.com) DNS name to your Amazon VPC Endpoint DNS name (i.e. vpce-svc-03d5ebb7d9579a2b3.us-east-1.vpce.amazonaws.com). IP addresses associated with Amazon VPC Endpoints can change at any time due to scaling up, scaling down, or software updates. Route 53 responds to each request for an Alias record with one or more IP addresses for the VPC endpoint. There is no additional charge for queries to Alias records that are mapped to Amazon VPC endpoints. These queries are listed as “Intra-AWS-DNS-Queries” on the Amazon Route 53 usage report.

**DNS Routing Policies**

Q. What are different DNS Routing Policies?

When you create a record, you choose a routing policy, which determines how Amazon Route 53 responds to queries:

* **Simple routing policy** – Use for a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.
* **Failover routing policy** – Use when you want to configure active-passive failover.
* **Geolocation routing policy** – Use when you want to route traffic based on the location of your users.
* **Geoproximity routing policy** – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.
* **Latency routing policy** – Use when you have resources in multiple AWS Regions and you want to route traffic to the region that provides the best latency.
* **Multivalue answer routing policy** – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.
* **Weighted routing policy** – Use to route traffic to multiple resources in proportions that you specify.

Q. What is Simple Routing?

Simple routing lets you configure standard DNS records, with no special Route 53 routing such as weighted or latency. With simple routing, you typically route traffic to a single resource, for example, to a web server for your website.

If you choose the simple routing policy in the Route 53 console, you can't create multiple records that have the same name and type, but you can specify multiple values in the same record, such as multiple IP addresses. (If you choose the simple routing policy for an alias record, you can specify only one AWS resource or one record in the current hosted zone.) If you specify multiple values in a record, Route 53 returns all values to the recursive resolver in random order, and the resolver returns the values to the client (such as a web browser) that submitted the DNS query. The client then chooses a value and resubmits the query.

Q. What is Failover Routing?

Failover routing lets you route traffic to a resource when the resource is healthy or to a different resource when the first resource is unhealthy (active-passive failover). The primary and secondary records can route traffic to anything from an Amazon S3 bucket that is configured as a website to a complex tree of records.

Q. What is Multivalue answer routing?

Multivalue answer routing lets you configure Amazon Route 53 to return multiple values, such as IP addresses for your web servers, in response to DNS queries. You can specify multiple values for almost any record, but multivalue answer routing also lets you check the health of each resource, so Route 53 returns only values for healthy resources. It's not a substitute for a load balancer, but the ability to return multiple health-checkable IP addresses is a way to use DNS to improve availability and load balancing.

To route traffic approximately randomly to multiple resources, such as web servers, you create one multivalue answer record for each resource and, optionally, associate a Route 53 health check with each record. Route 53 responds to DNS queries with up to eight healthy records and gives different answers to different DNS resolvers. If a web server becomes unavailable after a resolver caches a response, client software can try another IP address in the response.

Q. Does Amazon Route 53 support Weighted Round Robin (WRR)?

Yes. Weighted Round Robin allows you to assign weights to resource record sets in order to specify the frequency with which different responses are served. You may want to use this capability to do A/B testing, sending a small portion of traffic to a server on which you’ve made a software change. For instance, suppose you have two record sets associated with one DNS name—one with weight 3 and one with weight 1. In this case, 75% of the time Route 53 will return the record set with weight 3 and 25% of the time Route 53 will return the record set with weight 1. Weights can be any number between 0 and 255.

Q. What is Amazon Route 53's Latency Based Routing (LBR) feature?

LBR (Latency Based Routing) is a new feature for Amazon Route 53 that helps you improve your application’s performance for a global audience. You can run applications in multiple AWS regions and Amazon Route 53, using dozens of edge locations worldwide, will route end users to the AWS region that provides the lowest latency.

Q. How do I get started using Amazon Route 53's Latency Based Routing (LBR) feature?

You can start using Amazon Route 53’s new LBR feature quickly and easily by using either the AWS Management Console or a simple API. You simply create a record set that includes the IP addresses or ELB names of various AWS endpoints and mark that record set as an LBR-enabled Record Set, much like you mark a record set as a Weighted Record Set. Amazon Route 53 takes care of the rest - determining the best endpoint for each request and routing end users accordingly, much like Amazon CloudFront, Amazon’s global content delivery service, does.

Q. What is Amazon Route 53's Geo DNS feature?

Route 53 Geo DNS lets you balance load by directing requests to specific endpoints based on the geographic location from which the request originates. Geo DNS makes it possible to customize localized content, such as presenting detail pages in the right language or restricting distribution of content to only the markets you have licensed. Geo DNS also lets you balance load across endpoints in a predictable, easy-to-manage way, ensuring that each end-user location is consistently routed to the same endpoint. Geo DNS provides three levels of geographic granularity: continent, country, and state, and Geo DNS also provides a global record which is served in cases where an end user’s location doesn’t match any of the specific Geo DNS records you have created. You can also combine Geo DNS with other routing types, such as Latency Based Routing and DNS Failover, to enable a variety of low-latency and fault-tolerant architectures.

Q. When using Geo DNS, do I need a "global" record? When would Route 53 return this record?

Yes, we strongly recommend that you configure a global record, to ensure that Route 53 can provide a response to DNS queries from all possible locations—even if you have created specific records for each continent, country, or state where you expect your end users will be located. Route 53 will return the value contained in your global record in the following cases:

The DNS query comes from an IP address not recognized by Route 53’s Geo IP database.

The DNS query comes from a location not included in any of the specific Geo DNS records you have created.

Q. What is the difference between Latency Based Routing and Geo DNS?

Geo DNS bases routing decisions on the geographic location of the requests. In some cases, geography is a good proxy for latency; but there are certainly situations where it is not. LatencyBased Routing utilizes latency measurements between viewer networks and AWS datacenters. These measurements are used to determine which endpoint to direct users toward.

If your goal is to minimize end-user latency, we recommend using Latency Based Routing. If you have compliance, localization requirements, or other use cases that require stable routing from a specific geography to a specific endpoint, we recommend using Geo DNS.

Q. What is Amazon Route 53 Traffic Flow?

Amazon Route 53 Traffic Flow is an easy-to-use and cost-effective global traffic management service. With Amazon Route 53 Traffic Flow, you can improve the performance and availability of your application for your end users by running multiple endpoints around the world, using Amazon Route 53 Traffic Flow to connect your users to the best endpoint based on latency, geography, and endpoint health. Amazon Route 53 Traffic Flow makes it easy for developers to create policies that route traffic based on the constraints they care most about, including latency, endpoint health, load, geoproximity and geography. Customers can customize these templates or build policies from scratch using a simple visual policy builder in the AWS Management Console.

Q. How does a traffic policy using geoproximity rule route DNS traffic?

When you create a traffic flow policy, you can specify either an AWS region (if you're using AWS resources) or the latitude and longitude for each endpoint. For example, suppose you have EC2 instances in the AWS US East (Ohio) region and in the US West (Oregon) region. When an user in Seattle visits your website, geoproximity routing will route the DNS query to the EC2 instances in the US West (Oregon) region because it's closer geographically.

Q. How does the geoproximity bias value of an endpoint affect DNS traffic routing to other endpoints?

Geoproximity routing lets Amazon Route 53 route traffic to your resources based on the geographic location of your users and your resources. You can also optionally choose to route more traffic or less to a given resource by specifying a value, known as a *bias*. A bias expands or shrinks the size of the geographic region from which traffic is routed to a resource.

To use geoproximity routing, you must use Route 53 [traffic flow](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/traffic-flow.html).

Changing the geoproximity bias value on an endpoint either expands or shrinks the area from which Route 53 routes traffic to a resource. The geoproximity bias can't accurately predict the load factor, though, because a small shift in the size of geographic areas might include or exclude major metropolitan areas that generate large numbers of queries.

Q. What is DNS Failover?

DNS Failover consists of two components: health checks and failover. Health checks are automated requests sent over the Internet to your application to verify that your application is reachable, available, and functional. You can configure the health checks to be similar to the typical requests made by your users, such as requesting a web page from a specific URL. With DNS failover, Route 53 only returns answers for resources that are healthy and reachable from the outside world, so that your end users are routed away from a failed or unhealthy part of your application.

Q. How can I measure the performance of my application’s endpoints using Amazon Route 53?

Amazon Route 53 health checks include an optional latency measurement feature which provides data on how long it takes your endpoint to respond to a request. When you enable the latency measurement feature, the Amazon Route 53 health check will generate additional Amazon CloudWatch metrics showing the time required for Amazon Route 53’s health checkers to establish a connection and to begin receiving data. Amazon Route 53 provides a separate set of latency metrics for each AWS region where Amazon Route 53 health checks are conducted.

Q. How can I be notified if one of my endpoints starts failing its health check?

Because each Route 53 health check publishes its results as a CloudWatch metric, you can configure the full range of CloudWatch notifications and automated actions which can be triggered when the health check value changes beyond a threshold that you specify. First, in either the Route 53 or CloudWatch console, configure a CloudWatch alarm on the health check metric. Then add a notification action and specify the email or SNS topic that you want to publish your notification to.

Q. How do I transfer my existing domain name registration to Amazon Route 53 without disrupting my existing web traffic?

First, you need to get a list of the DNS record data for your domain name, generally available in the form of a “zone file” that you can get from your existing DNS provider. With the DNS record data in hand, you can use Route 53’s Management Console or simple web-services interface to create a hosted zone that can store the DNS records for your domain name and follow its transfer process, which will include such steps as updating the name servers for your domain name to the ones associated with your hosted zone. To complete the domain name transfer process, contact the registrar with whom you registered your domain name and follow its transfer process, which will include steps such as updating the name servers for your domain name to the ones associated with your hosted zone. As soon as your registrar propagates the new name server delegations, the DNS queries from your end users will start to get answered by the Route 53 DNS servers.

**AWS Direct Connect**

Q. What is AWS Direct Connect?

AWS Direct Connect is a network service that provides an alternative to using the Internet to connect customer's on premise sites to AWS.

Q. What can I do with AWS Direct Connect?

Using AWS Direct Connect, data that would have previously been transported over the Internet can now be delivered through a private network connection between AWS and your datacenter or corporate network.

Q. What are the benefits of using AWS Direct Connect and private network connections?

In many circumstances, private network connections can reduce costs, increase bandwidth, and provide a more consistent network experience than Internet-based connections.

Q. If I’m using Amazon CloudFront and my origin is in my own data center, can I use AWS Direct Connect to transfer the objects stored in my own data center?

Yes. Amazon CloudFront supports custom origins including origins you run outside of AWS. The access to the CloudFront edge locations will be restricted to the geographically nearest AWS region, with the exception of the North America regions which currently allow access to all North American region's on-net CloudFront origins. With AWS Direct Connect, you will pay AWS Direct Connect data transfer rates for origin transfer.

Through Direct Connect, customer traffic will remain in Amazon's backbone network after it enters it. Therefore, prefixes of CloudFront locations that are not on the Amazon backbone network will not be advertised through Direct Connect.

Q. How does AWS Direct Connect differ from an IPSec VPN Connection?

A VPC VPN Connection utilizes IPSec to establish encrypted network connectivity between your intranet and Amazon VPC over the Internet. VPN Connections can be configured in minutes and are a good solution if you have an immediate need, have low to modest bandwidth requirements, and can tolerate the inherent variability in Internet-based connectivity. AWS Direct Connect does not involve the Internet; instead, it uses dedicated, private network connections between your intranet and Amazon VPC.