



Amazon Redshift

- A fully managed, **petabyte-scale data warehouse** service.
- Redshift extends data warehouse queries to your data lake. You can run analytic queries against petabytes of data stored locally in Redshift, and directly against exabytes of data stored in S3.
- RedShift is an OLAP type of DB.
- Currently, Redshift only supports Single-AZ deployments.
- Features
 - Redshift uses **columnar storage**, data compression, and zone maps to reduce the amount of I/O needed to perform queries.
 - It uses a **massively parallel processing** data warehouse architecture to parallelize and distribute SQL operations.
 - Redshift uses machine learning to deliver high throughput based on your workloads.
 - Redshift uses **result caching** to deliver sub-second response times for repeat queries.
 - Redshift automatically and continuously backs up your data to S3. It can asynchronously replicate your snapshots to S3 in another region for disaster recovery.
- Security
 - By default, an Amazon Redshift cluster is only accessible to the AWS account that creates the cluster.
 - Use IAM to create user accounts and manage permissions for those accounts to control cluster operations.
 - If you are using the EC2-Classical platform for your Redshift cluster, you must use Redshift security groups.
 - If you are using the EC2-VPC platform for your Redshift cluster, you must use VPC security groups.
 - When you provision the cluster, you can optionally choose to encrypt the cluster for additional security. Encryption is an immutable property of the cluster.
 - Snapshots created from the encrypted cluster are also encrypted.
- Pricing
 - You pay a per-second billing rate based on the type and number of nodes in your cluster.
 - You pay for the number of bytes scanned by RedShift Spectrum
 - You can reserve instances by committing to using Redshift for a 1 or 3 year term and save costs.

Sources:

<https://docs.aws.amazon.com/redshift/latest/mgmt/>
<https://aws.amazon.com/redshift/features/>
<https://aws.amazon.com/redshift/pricing/>
<https://aws.amazon.com/redshift/faqs/>



NETWORKING AND CONTENT DELIVERY

Amazon API Gateway

- Enables developers to create, publish, maintain, monitor, and secure APIs at any scale.
- Allows creating, deploying, and managing a RESTful API to expose backend HTTP endpoints, Lambda functions, or other AWS services.
- Together with Lambda, API Gateway forms the app-facing part of the AWS serverless infrastructure.
- Features
 - API Gateway can execute Lambda code in your account, start Step Functions state machines, or make calls to Elastic Beanstalk, EC2, or web services outside of AWS with publicly accessible HTTP endpoints.
 - API Gateway helps you define plans that meter and restrict third-party developer access to your APIs.
 - API Gateway helps you manage traffic to your backend systems by allowing you to set throttling rules based on the number of requests per second for each HTTP method in your APIs.
 - You can set up a cache with customizable keys and time-to-live in seconds for your API data to avoid hitting your backend services for each request.
 - API Gateway lets you run multiple versions of the same API simultaneously with **API Lifecycle**.
 - After you build, test, and deploy your APIs, you can package them in an API Gateway usage plan and sell the plan as a Software as a Service (SaaS) product through AWS Marketplace.
 - API Gateway offers the ability to create, update, and delete documentation associated with each portion of your API, such as methods and resources.
 - Amazon API Gateway offers general availability of HTTP APIs, which gives you the ability to route requests to private ELBs AWS AppConfig, Amazon EventBridge, Amazon Kinesis Data Streams, Amazon SQS, AWS Step Functions and IP-based services registered in AWS CloudMap such as ECS tasks. Previously, HTTP APIs enabled customers to only build APIs for their serverless applications or to proxy requests to HTTP endpoints.
 - You can create data mapping definitions from an HTTP API's method request data (e.g. path parameters, query string, and headers) to the corresponding integration request parameters and from the integration response data (e.g. headers) to the HTTP API method response parameters.
 - Use wildcard custom domain names (*.example.com) to create multiple URLs that route to one API Gateway HTTP API.
 - You can configure your custom domain name to route requests to different APIs. Using multi-level base path mappings, you can implement path-based API versioning and migrate API traffic between APIs according to request paths with many segments.
- All of the APIs created expose **HTTPS endpoints only**. API Gateway does not support unencrypted (HTTP) endpoints.
- Monitoring



- API Gateway console is integrated with CloudWatch, so you get backend performance metrics such as API calls, latency, and error rates.
- You can set up custom alarms on API Gateway APIs.
- API Gateway can also log API execution errors to CloudWatch Logs.
- Pricing
 - You pay only for the API calls you receive and the amount of data transferred out.
 - API Gateway also provides optional data caching charged at an hourly rate that varies based on the cache size you select.

Sources:

<https://docs.aws.amazon.com/apigateway/latest/developerguide/>

<https://aws.amazon.com/api-gateway/features/>

<https://aws.amazon.com/api-gateway/pricing/>

<https://aws.amazon.com/api-gateway/faqs/>



Amazon CloudFront

- A web service that speeds up distribution of your static and dynamic web content to your users. A Content Delivery Network (CDN) service.
- It delivers your content through a worldwide network of data centers called **edge locations**. When a user requests content that you're serving with CloudFront, the user is routed to the edge location that provides the lowest latency, so that content is delivered with the best possible performance.
 - If the content is already in the edge location with the lowest latency, CloudFront delivers it immediately.
 - If the content is not in that edge location, CloudFront retrieves it from an origin that you've defined
- CloudFront also has **regional edge caches** that bring more of your content closer to your viewers, even when the content is not popular enough to stay at a CloudFront edge location, to help improve performance for that content.
- Different CloudFront Origins
 - **Using S3 buckets for your origin** - you place any objects that you want CloudFront to deliver in an S3 bucket.
 - **Using S3 buckets configured as website endpoints for your origin**
 - **Using a mediastore container or a media package channel for your origin** - you can set up an S3 bucket that is configured as a MediaStore container, or create a channel and endpoints with MediaPackage. Then you create and configure a distribution in CloudFront to stream the video.
 - **Using EC2 or other custom origins** - A custom origin is an HTTP server, for example, a web server.
 - **Using CloudFront Origin Groups for origin failover** - use origin failover to designate a primary origin for CloudFront plus a second origin that CloudFront automatically switches to when the primary origin returns specific HTTP status code failure responses.
- CloudFront Distributions
 - You create a **CloudFront distribution** to tell CloudFront where you want content to be delivered from, and the details about how to track and manage content delivery.
 - You create a distribution and choose the configuration settings you want:
 - Your content origin—that is, the Amazon S3 bucket, MediaPackage channel, or HTTP server from which CloudFront gets the files to distribute. You can specify any combination of up to 25 S3 buckets, channels, and/or HTTP servers as your origins.
 - Access—whether you want the files to be available to everyone or restrict access to some users.
 - Security—whether you want CloudFront to require users to use HTTPS to access your content.
- Price Class



- Choose the price class that corresponds with the maximum price that you want to pay for CloudFront service. By default, CloudFront serves your objects from edge locations in all CloudFront regions.
- Monitoring
 - CloudFront integrates with Amazon CloudWatch metrics so that you can monitor your website or application.
 - Capture API requests with AWS CloudTrail. CloudFront is a global service. To view CloudFront requests in CloudTrail logs, you must update an existing trail to include global services.
- Pricing
 - Charge for storage in an S3 bucket.
 - Charge for serving objects from edge locations.
 - Charge for submitting data to your origin.
 - Data Transfer Out
 - HTTP/HTTPS Requests
 - Invalidation Requests,
 - Dedicated IP Custom SSL certificates associated with a CloudFront distribution.
 - You also incur a surcharge for HTTPS requests, and an additional surcharge for requests that also have field-level encryption enabled.

Sources:

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide>

<https://aws.amazon.com/cloudfront/features/>

<https://aws.amazon.com/cloudfront/pricing/>

<https://aws.amazon.com/cloudfront/faqs/>



AWS Elastic Load Balancing

- Distributes incoming application or network traffic across multiple targets, such as **EC2 instances**, **containers (ECS)**, **Lambda functions**, and **IP addresses**, in multiple Availability Zones.

General features

- Accepts incoming traffic from clients and routes requests to its registered targets.
- Monitors the health of its registered targets and routes traffic only to healthy targets.
- Enable deletion protection to prevent your load balancer from being deleted accidentally. Disabled by default.
- Deleting ELB won't delete the instances registered to it.
- **Cross Zone Load Balancing** - when enabled, each load balancer node distributes traffic across the registered targets in all enabled AZs.
- Supports SSL Offloading which is a feature that allows the ELB to bypass the SSL termination by removing the SSL-based encryption from the incoming traffic.

Types of Load Balancers

- **Application Load Balancer**
 - Functions at the application layer, the **seventh layer** of the Open Systems Interconnection (OSI) model.
 - Allows HTTP and HTTPS.
 - At least 2 subnets must be specified when creating this type of load balancer.
 - Monitoring:
 - CloudWatch metrics - retrieve statistics about data points for your load balancers and targets as an ordered set of time-series data, known as *metrics*.
 - Access logs - capture detailed information about the requests made to your load balancer and store them as log files in S3.
 - CloudTrail logs - capture detailed information about the calls made to the Elastic Load Balancing API and store them as log files in S3.
- **Network Load Balancer**
 - Functions at the **fourth layer** of the Open Systems Interconnection (OSI) model. Uses TCP and UDP connections.
 - At least 1 subnet must be specified when creating this type of load balancer, but the recommended number is 2.
 - Monitoring:
 - CloudWatch metrics - retrieve statistics about data points for your load balancers and targets as an ordered set of time-series data, known as *metrics*.
 - VPC Flow Logs - capture detailed information about the traffic going to and from your Network Load Balancer.



- CloudTrail logs - capture detailed information about the calls made to the Elastic Load Balancing API and store them as log files in Amazon S3.
- **Gateway Load Balancer**
 - Enables you to deploy, scale, and manage virtual appliances, such as firewalls, intrusion detection and prevention systems, and deep packet inspection systems.
 - Operates at the third layer of the Open Systems Interconnection (OSI) model, the network layer. It listens for all IP packets across all ports and forwards traffic to the target group that's specified in the listener rule.
 - Gateway Load Balancers use Gateway Load Balancer endpoints to securely exchange traffic across VPC boundaries. A Gateway Load Balancer endpoint is a VPC endpoint that provides private connectivity between virtual appliances in the service provider VPC and application servers in the service consumer VPC.
 - Traffic to and from a Gateway Load Balancer endpoint is configured using route tables.
- **Classic Load Balancer**
 - Distributes incoming application traffic across multiple EC2 instances in multiple Availability Zones.
 - For use with EC2 classic only. Register instances with the load balancer. AWS recommends using Application or Network load balancers instead.
 - An **Internet-facing load balancer** has a publicly resolvable DNS name, so it can route requests from clients over the Internet to the EC2 instances that are registered with the load balancer. Classic load balancers are always Internet-facing.
 - Monitoring:
 - CloudWatch metrics - retrieve statistics about ELB-published data points as an ordered set of time-series data, known as *metrics*.
 - Access logs - capture detailed information for requests made to your load balancer and store them as log files in the S3 bucket that you specify.
 - CloudTrail logs - keep track of the calls made to the Elastic Load Balancing API by or on behalf of your AWS account.

Security, Authentication and Access Control

- Use IAM Policies to grant permissions
 - Resource-level permissions
 - Security groups that control the traffic allowed to and from your load balancer.
- Recommended rules for internet-facing load balancer:

Inbound	
Source	Port Range



0.0.0.0/0	<i>listener</i>
Outbound	
Destination	<i>Port Range</i>
<i>instance security group</i>	<i>instance listener</i>
<i>instance security group</i>	<i>health check</i>

For internal load balancer:

Inbound	
Source	Port Range
<i>VPC CIDR</i>	<i>listener</i>
Outbound	
Destination	Port Range
<i>instance security group</i>	<i>instance listener</i>
<i>instance security group</i>	<i>health check</i>

Summary of Features



Feature	Application Load Balancer	Network Load Balancer	Gateway Load Balancer
Protocols	HTTP, HTTPS, gRPC	TCP, UDP, TLS	IP
Platforms	VPC	VPC	VPC
Health checks	HTTP, HTTPS, gRPC	TCP, HTTP, HTTPS	TCP, HTTP, HTTPS
Cloudwatch Metrics	✓	✓	✓
Logging	✓	✓	✓
Zonal Failover	✓	✓	✓
Connection Draining (deregistration delay)	✓	✓	✓
Load Balancing to multiple ports on the same instance	✓	✓	✓
IP addresses as targets	✓	✓ (TCP, TLS)	✓
Load balancer deletion protection	✓	✓	✓
Configuration idle connection timeout	✓		
Cross-zone load balancing	✓	✓	✓
Sticky sessions	✓	✓	✓
Static IP		✓	
Elastic IP address		✓	
Preserve Source IP address	✓	✓	✓
Resource-based IAM permissions	✓	✓	✓
Tag-based IAM permissions	✓	✓	✓
Slow start	✓		
Web sockets	✓	✓	✓
PrivateLink Support		✓ (TCP, TLS)	✓ (GWLBE)
Source IP address CIDR-based routing	✓		



Feature	Application Load Balancer	Network Load Balancer	Gateway Load Balancer
Layer 7			
Path-based routing	✓		
Host-based routing	✓		
Native HTTP/2	✓		
Redirects	✓		
Fixed response	✓		
Lambda functions as targets	✓		
HTTP header-based routing	✓		
HTTP method-based routing	✓		
Query string parameter-based routing	✓		
Security			
SSL offloading	✓	✓	
Server Name Indication (SNI)	✓	✓	
Back-end server encryption	✓	✓	
User authentication	✓		
Session Resumption	✓	✓	
Terminates flow/proxy behavior	✓	✓	✓

Pricing

- You are charged for each hour or partial hour that an Application Load Balancer is running and the number of Load Balancer Capacity Units (LCU) used per hour.
- You are charged for each hour or partial hour that a Network Load Balancer is running and the number of Load Balancer Capacity Units (LCU) used by Network Load Balancer per hour.



- You are charged for each hour or partial hour that a Gateway Load Balancer is running and the number of Gateway Load Balancer Capacity Units (GLCU) used by Gateway Load Balancer per hour.
- You are charged for each hour or partial hour that a Classic Load Balancer is running and for each GB of data transferred through your load balancer.

Sources:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html>

<https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html>

<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/introduction.html>

<https://aws.amazon.com/elasticloadbalancing/features/>

<https://aws.amazon.com/elasticloadbalancing/pricing/?nc=sn&loc=3>



Amazon Route 53

- A highly available and scalable Domain Name System (DNS) web service used for domain registration, DNS routing, and health checking.

Key Features

- Resolver
- Traffic flow
- Latency based routing
- Geo DNS
- Private DNS for Amazon VPC
- DNS Failover
- Health Checks and Monitoring
- Domain Registration
- CloudFront and S3 Zone Apex Support
- Amazon ELB Integration

Domain Registration

- Choose a domain name and confirm that it's available, then register the domain name with Route 53. The service automatically makes itself the DNS service for the domain by doing the following:
 - Creates a hosted zone that has the same name as your domain.
 - Assigns a set of four name servers to the hosted zone. When someone uses a browser to access your website, such as `www.example.com`, these name servers tell the browser where to find your resources, such as a web server or an S3 bucket.
 - Gets the name servers from the hosted zone and adds them to the domain.
- If you already registered a domain name with another registrar, you can choose to transfer the domain registration to Route 53.

Routing Internet Traffic to your Website or Web Application

- Use the Route 53 console to register a domain name and configure Route 53 to route internet traffic to your website or web application.
- After you register your domain name, Route 53 automatically creates a **public hosted zone** that has the same name as the domain.
- To route traffic to your resources, you create **records**, also known as *resource record sets*, in your hosted zone.
- You can create special Route 53 records, called **alias records**, that route traffic to S3 buckets, CloudFront distributions, and other AWS resources.
- Each record includes information about how you want to route traffic for your domain, such as:



- Name - name of the record corresponds with the domain name or subdomain name that you want Route 53 to route traffic for.
- Type - determines the type of resource that you want traffic to be routed to.
- Value

Know the following Concepts

- Domain Registration Concepts - domain name, domain registrar, domain registry, domain reseller, top-level domain
- DNS Concepts
 - **Alias record** - a type of record that you can create to route traffic to AWS resources.
 - DNS query
 - DNS resolver
 - Domain Name System (DNS)
 - Private DNS
 - **Hosted zone** - a container for records, which includes information about how to route traffic for a domain and all of its subdomains.
 - **Name servers** - servers in the DNS that help to translate domain names into the IP addresses that computers use to communicate with one another.
 - **Record** (DNS record) - an object in a hosted zone that you use to define how you want to route traffic for the domain or a subdomain.
 - **Routing policy**
 - **Subdomain**
 - Time to live (TTL)

Records

- Alias Records
 - Route 53 **alias records** provide a Route 53–specific extension to DNS functionality. Alias records let you route traffic to selected AWS resources. They also let you route traffic from one record in a hosted zone to another record.
 - You can create an alias record at the top node of a DNS namespace, also known as the zone apex.
- CNAME Record
 - You cannot create an alias record at the top node of a DNS namespace using a CNAME record.
- Alias records vs CNAME records



CNAME Records	Alias Records
You can't create a CNAME record at the zone apex.	You can create an alias record at the zone apex. Alias records must have the same type as the record you're routing traffic to.
Route 53 charges for CNAME queries.	Route 53 doesn't charge for alias queries to AWS resources.
A CNAME record redirects queries for a domain name regardless of record type.	Route 53 responds to a DNS query only when the name and type of the alias record matches the name and type in the query.
A CNAME record can point to any DNS record that is hosted anywhere.	An alias record can only point to selected AWS resources or to another record in the hosted zone that you're creating the alias record in.
A CNAME record appears as a CNAME record in response to dig or Name Server (NS) lookup queries.	An alias record appears as the record type that you specified when you created the record, such as A or AAAA.

Route 53 Health Checks and DNS Failover

Step 1: Configure health check

Step 2: Get notified when health check fails

Configure health check

Route 53 health checks let you track the health status of your resources, such as web servers or mail servers, and take action when an outage occurs.

Name

What to monitor ☒ Endpoint ☐ Status of other health checks (calculated health check) ☐ State of CloudWatch alarm

Monitor an endpoint

Multiple Route 53 health checkers will try to establish a TCP connection with the following resource to determine whether it's healthy. [Learn more](#)

Specify endpoint by ☒ IP address ☐ Domain name

Protocol

IP address *

Host name

Port *

Path

Advanced configuration

URL

Health check type Basic - no additional options selected [View Pricing](#)

- Each health check that you create can monitor one of the following:
 - The health of a specified resource, such as a web server
 - The status of other health checks
 - The status of an Amazon CloudWatch alarm
- Two types of failover configurations
 - Active-Active Failover** - all the records that have the same name, the same type, and the same routing policy are active unless Route 53 considers them unhealthy. Use this failover configuration when you want all of your resources to be available the majority of the time.
 - Active-Passive Failover** - use this failover configuration when you want a primary resource or group of resources to be available the majority of the time and you want a secondary resource or group of resources to be on standby in case all the primary resources become unavailable. When responding to queries, Route 53 includes only the healthy primary resources.

Monitoring

- The Route 53 dashboard provides detailed information about the status of your domain registrations, including:



- Status of new domain registrations
- Status of domain transfers to Route 53
- List of domains that are approaching the expiration date
- You can use Amazon CloudWatch metrics to see the number of DNS queries served for each of your Route 53 public hosted zones. With these metrics, you can see at a glance the activity level of each hosted zone to monitor changes in traffic.
- You can monitor your resources by creating Route 53 health checks, which use CloudWatch to collect and process raw data into readable, near real-time metrics.
- Log API calls with CloudTrail

Pricing

- A hosted zone is charged at the time it's created and on the first day of each subsequent month. To allow testing, a hosted zone that is deleted within 12 hours of creation is not charged, however, any queries on that hosted zone will still incur charges.
- Billion queries / month
- Queries to Alias records are provided at no additional cost to current Route 53 customers when the records are mapped to the following AWS resource types:
 - Elastic Load Balancers
 - Amazon CloudFront distributions
 - AWS Elastic Beanstalk environments
 - Amazon S3 buckets that are configured as website endpoints
- Traffic flow policy record / month
- Pricing for domain names varies by Top Level Domain (TLD)

Sources:

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html>

<https://aws.amazon.com/route53/features/>

<https://aws.amazon.com/route53/pricing/>

Amazon VPC

- Create a virtual network in the cloud dedicated to your AWS account where you can launch AWS resources
- Amazon VPC is the networking layer of Amazon EC2
- A VPC spans all the Availability Zones in the region. After creating a VPC, you can add one or more subnets in each Availability Zone.

Key Concepts

- A **virtual private cloud** (VPC) allows you to specify an IP address range for the VPC, add subnets, associate security groups, and configure route tables.
- 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.
- To protect the AWS resources in each subnet, use **security groups** and **network access control lists (ACL)**.
- Expand your VPC by adding secondary IP ranges.

Default vs Non-Default VPC

Default	Non-Default VPC
If your account supports the EC2-VPC platform only, it comes with a default VPC that has a default subnet in each Availability Zone.	You can create your own non-default VPC, and configure it as you need. Subnets that you create in your non-default VPC and additional subnets that you create in your default VPC are called non-default subnets.
Your default VPC includes an internet gateway, which allows your instances to communicate with the internet, and each default subnet is a public subnet.	Instances can communicate with each other, but can't access the internet. You can enable internet access for an instance launched into a non-default subnet by attaching an internet gateway and associating an Elastic IP address with the instance.
Each instance that you launch into a default subnet has a private IPv4 address and a public IPv4 address.	By default, each instance that you launch into a non-default subnet has a private IPv4 address, but no public IPv4 address, unless you specifically assign one at launch, or you modify the subnet's public IP address attribute.
To allow an instance in your VPC to initiate outbound connections to the internet but prevent unsolicited inbound connections from the internet, you can use a network address translation (NAT) device for IPv4 traffic.	To allow an instance in your VPC to initiate outbound connections to the internet but prevent unsolicited inbound connections from the internet, you can use a network address translation (NAT) device for IPv4 traffic.
You can optionally associate an Amazon-provided IPv6 CIDR block with your VPC and assign IPv6 addresses to your instances. IPv6 traffic is separate from IPv4 traffic; your route tables must include separate routes for IPv6 traffic.	You can optionally associate an Amazon-provided IPv6 CIDR block with your VPC and assign IPv6 addresses to your instances. IPv6 traffic is separate from IPv4 traffic; your route tables must include separate routes for IPv6 traffic.



Accessing a Corporate or Home Network

- You can optionally connect your VPC to your own corporate data center using an **IPsec AWS managed VPN connection**, making the AWS Cloud an extension of your data center.
- A **VPN connection** consists of:
 - a **virtual private gateway** (which is the VPN concentrator on the Amazon side of the VPN connection) attached to your VPC.
 - a **customer gateway** (which is a physical device or software appliance on your side of the VPN connection) located in your data center.
 - A diagram of the connection

VPC Use Case Scenarios

- VPC with a Single Public Subnet
- VPC with Public and Private Subnets (NAT)
- VPC with Public and Private Subnets and AWS Managed VPN Access
- VPC with a Private Subnet Only and AWS Managed VPN Access

Subnets

- When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block (example: 10.0.0.0/16). This is the **primary CIDR block** for your VPC.
- You can add one or more subnets in each Availability Zone of your VPC's region.
- You specify the CIDR block for a subnet, which is a subset of the VPC CIDR block.
- A CIDR block must not overlap with any existing CIDR block that's associated with the VPC.
- Types of Subnets
 - Public Subnet - has an internet gateway
 - Private Subnet - doesn't have an internet gateway
 - VPN-only Subnet - has a virtual private gateway instead
- You cannot increase or decrease the size of an existing CIDR block.
- When you associate a CIDR block with your VPC, a route is automatically added to your VPC route tables to enable routing within the VPC (the destination is the CIDR block and the target is *local*).
- You have a limit on the number of CIDR blocks you can associate with a VPC and the number of routes you can add to a route table.

Subnet Routing


- Each subnet must be associated with a **route table**, which specifies the allowed routes for **outbound traffic** leaving the subnet.
- Every subnet that you create is automatically associated with the main route table for the VPC.
- You can change the association, and you can change the contents of the main route table.



- You can allow an instance in your VPC to initiate outbound connections to the internet over IPv4 but prevent unsolicited inbound connections from the internet using a **NAT gateway or NAT instance**.
- To initiate outbound-only communication to the internet over IPv6, you can use an egress-only internet gateway.

Subnet Security

- Security Groups — control inbound and outbound traffic for your instances
 - You can associate one or more (up to five) security groups to an instance in your VPC.
 - If you don't specify a security group, the instance automatically belongs to the default security group.
 - When you create a security group, it has no inbound rules. By default, it includes an outbound rule that allows all outbound traffic.
 - Security groups are associated with network interfaces.
- Network Access Control Lists — control inbound and outbound traffic for your subnets
 - Each subnet in your VPC must be associated with a network ACL. If none is associated, automatically associated with the default network ACL.
 - You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time.
 - A network ACL contains a numbered list of rules that is evaluated in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL.
 - The default network ACL is configured to **allow all traffic to flow in and out** of the subnets to which it is associated.
 - For custom ACLs, you need to add a rule for ephemeral ports, usually with the range of 32768-65535. If you have a NAT Gateway, ELB or a Lambda function in a VPC, you need to enable 1024-65535 port range.
- Flow logs — capture information about the IP traffic going to and from network interfaces in your VPC that is published to CloudWatch Logs.

SECURITY GROUP	NETWORK ACL
<p>Operates at the instance level</p> <p>Supports allow rules only</p> <p>Is stateful: Return traffic is automatically allowed, regardless of any rules</p> <p>We evaluate all rules before deciding whether to allow traffic</p> <p>Applies only to EC2 instances and similar services that use EC2 as a backend.</p> <p>Security group is specified when launching the instance, or is associated with the instance later on</p>	<p>Operates at the subnet level</p> <p>Supports allow rules and deny rules</p> <p>Is stateless: Return traffic must be explicitly allowed by rules</p> <p>We process rules in number order when deciding whether to allow traffic</p> <p>Automatically applies to all</p> <p>Instances in the subnets it's associated with</p> 


















- Diagram of security groups and NACLs in a VPC

VPC Networking Components

- Network Interfaces
 - A virtual network interface that can include:
 - a primary private IPv4 address
 - one or more secondary private IPv4 addresses
 - one Elastic IP address per private IPv4 address
 - one public IPv4 address, which can be auto-assigned to the network interface for eth0 when you launch an instance
 - one or more IPv6 addresses
 - one or more security groups
 - a MAC address
 - a source/destination check flag
 - a description
 - Network interfaces can be attached and detached from instances, however, you cannot detach a primary network interface.
- Route Tables
 - Contains a set of rules, called *routes*, that are used to determine where network traffic is directed.



- A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.
- You cannot delete the main route table, but you can replace the main route table with a custom table that you've created.
- You must update the route table for any subnet that uses gateways or connections.
- Internet Gateways
 - Allows communication between instances in your VPC and the internet.
 - Imposes no availability risks or bandwidth constraints on your network traffic.
- NAT
 - Enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating connections with the instances.
 - NAT Instance vs NAT Gateways

 Tutorials Dojo	NAT gateway	NAT instance
 Availability	Highly available. NAT gateways in each Availability Zone are implemented with redundancy. Create a NAT gateway in each Availability Zone to ensure zone-independent architecture.	Use a script to manage failover between instances
 Bandwidth	Can scale up to 45 Gbps.	Depends on the bandwidth of the instance type
 Maintenance	Manage by AWS	Manage by you.
 Performance	Software is optimized for handling NAT traffic	A generic Amazon Linux AMI that's configured to perform NAT
 Cost	Charged depending on the number of NAT gateways you use, duration of usage, and amount of data that you send through the NAT gateways.	Charged depending on the number of NAT instances that you use, duration of usage, and instance type and size.
 Type and size	Uniform offering; you don't need to decide on the type or size.	Choose a suitable instance type and size, according to your predicted workload.
 Public IP addresses	Choose the Elastic IP address to associate with a NAT gateway at creation.	Use an elastic IP address or a public IP address with a NAT instance. You can change the public IP address at any time by associating a new elastic IP address with the instance.
 Private IP addresses	Automatically selected from the subnet's IP address range when you create the gateway.	Assign a specific private IP address from the subnet's IP address range when you launch the instance.
 Security groups	Cannot be associated with a NAT gateway	Associate with your NAT instance and the resources behind your NAT instance to control inbound and outbound traffic.
 Network ACLs	Use a network ACL to control the traffic to and from the subnet in which your NAT gateway resides.	Use a network ACL to control the traffic to and from the subnet in which your NAT instance resides.
 Flow logs	Use flow logs to capture the traffic.	Use flow logs to capture the traffic.
 Port Forwarding	Not supported.	Manually customize the configuration to support port forwarding.
 Bastion Servers	Not supported.	Use as a bastion server.
 Traffic Metrics	Monitor your NAT gateway using cloudwatch.	View Cloudwatch metrics for the instance.
 Timeout Behavior	When a connection times out, a NAT gateway returns an RST packet to any resources behind the NAT gateway that attempt to continue the connection (it does not send a FIN packet).	When a connection times out, a NAT instance sends a FIN packet to resources behind the NAT instance to close the connection.
 IP Fragmentation	Supports forwarding of IP fragmented packets for the UDP protocol. Does not support fragmentation for the TCP and ICMP protocols. Fragmented packets for these protocols will get dropped.	Supports reassembly of IP fragmented packets for the UDP, TCP, and ICMP protocols.

- DNS
 - AWS provides instances launched in a default VPC with public and private DNS hostnames that correspond to the public IPv4 and private IPv4 addresses for the instance.
- Elastic IP Addresses
 - **A static, public IPv4 address.**
 - You can associate an Elastic IP address with any instance or network interface for any VPC in your account.



- You can mask the failure of an instance by rapidly remapping the address to another instance in your VPC.
- Your Elastic IP addresses remain associated with your AWS account until you explicitly release them.
- AWS imposes a small hourly charge when EIPs aren't associated with a running instance, or when they are associated with a stopped instance or an unattached network interface.
- You're limited to five Elastic IP addresses.

Pricing

- Charged for VPN Connection-hour
- Charged for each "NAT Gateway-hour" that your NAT gateway is provisioned and available.
- Data processing charges apply for each Gigabyte processed through the NAT gateway regardless of the traffic's source or destination.
- You also incur standard AWS data transfer charges for all data transferred via the NAT gateway.
- Charges for unused or inactive Elastic IPs.

Sources:

<https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html>

<https://aws.amazon.com/vpc/details/>

<https://aws.amazon.com/vpc/pricing/>

<https://aws.amazon.com/vpc/faqs/>



SECURITY AND IDENTITY

AWS Identity and Access Management (IAM)

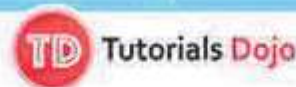
- Control who is authenticated (signed in) and authorized (has permissions) to use resources.
- AWS account **root user** is a single sign-in identity that has complete access to all AWS services and resources in the account.
- **Features**
 - You can grant other people permission to administer and use resources in your AWS account without having to share your password or access key.
 - You can grant different permissions to different people for different resources.
 - You can add two-factor authentication to your account and to individual users for extra security.
 - You receive AWS CloudTrail log records that include information about **IAM identities** who made requests for resources in your account.
 - You use an **access key** (an access key ID and secret access key) to make programmatic requests to AWS. An Access Key ID and Secret Access Key can only be uniquely generated once and must be regenerated if lost.
 - Your unique account sign-in page URL:
`https://My_AWS_Account_ID.signin.aws.amazon.com/console/`
 - You can use IAM tags to add custom attributes to an IAM user or role using a tag key–value pair.
 - You can generate and download a credential report that lists all users on your AWS account. The report also shows the status of passwords, access keys, and MFA devices.
- **Infrastructure Elements**
 - **Principal**
 - An entity that can make a request for an action or operation on an AWS resource. Users, roles, federated users, and applications are all AWS principals.
 - Your AWS account root user is your *first principal*.
 - **Request**
 - When a principal tries to use the AWS Management Console, the AWS API, or the AWS CLI, that principal sends a *request* to AWS.
 - Requests includes the following information:
 - **Actions or operations** – the actions or operations that the principal wants to perform.
 - **Resources** – the AWS resource object upon which the actions or operations are performed.
 - **Principal** – the user, role, federated user, or application that sent the request. Information about the principal includes the policies that are associated with that principal.



- **Environment data** – information about the IP address, user agent, SSL enabled status, or the time of day.
 - **Resource data** – data related to the resource that is being requested.
 - **Authentication**
 - To authenticate from the console as a user, you must sign in with your username and password.
 - To authenticate from the API or AWS CLI, you must provide your access key and secret key.
 - **Authorization**
 - To provide your users with permissions to access the AWS resources in their own account, you need **identity-based policies**.
 - **Resource-based policies** are for granting cross-account access.
 - Evaluation logic rules for policies:
 - By default, **all requests are denied**.
 - An *explicit allow* in a permissions policy overrides this default.
 - A *permissions boundary* overrides the allow. If there is a permissions boundary that applies, that boundary must allow the request. Otherwise, it is implicitly denied.
 - An explicit “deny” in any policy overrides any “allow”.
 - **Actions or Operations**
 - Operations are defined by a service, and include things that you can do to a resource, such as viewing, creating, editing, and deleting that resource.
 - **Resource**
 - An object that exists within a service. The service defines a set of actions that can be performed on each resource.
- **Users**
 - **IAM Users**
 - Instead of sharing your root user credentials with others, you can create individual **IAM users** within your account that correspond to users in your organization. IAM users are not separate accounts; they are users within your account.
 - Each user can have its own password for access to the AWS Management Console. You can also create an individual access key for each user so that the user can make programmatic requests to work with resources in your account.
 - By default, a brand new IAM user has **NO permissions** to do anything.
 - Users are global entities.
 - **Federated Users**
 - If the users in your organization already have a way to be authenticated, you can federate those user identities into AWS.
 - **IAM Groups**
 - An IAM group is a collection of IAM users.

- You can organize IAM users into IAM groups and attach access control policies to a group.
- A user can belong to multiple groups.
- Groups cannot belong to other groups.
- Groups do not have security credentials, and cannot access web services directly.
- **IAM Role**
 - A role does not have any credentials associated with it.
 - An IAM user can assume a role to temporarily take on different permissions for a specific task. A role can be assigned to a federated user who signs in by using an external identity provider instead of IAM.
 - **AWS service role** is a role that a service assumes to perform actions in your account on your behalf. This service role must include all the permissions required for the service to access the AWS resources that it needs.
- Users or groups can have multiple policies attached to them that grant different permissions.

When to Create IAM User	When to Create an IAM Role
You created an AWS account and you're the only person who works in your account.	You're creating an application that runs on an Amazon EC2 instance and that application makes requests to AWS.
Other people in your group need to work in your AWS account, and your group is using no other identity mechanism.	You're creating an app that runs on a mobile phone and that makes requests to AWS.
You want to use the command-line interface to work with AWS.	Users in your company are authenticated in your corporate network and want to be able to use AWS without having to sign in again (federate into AWS)



- **Policies**
 - Most permission policies are JSON policy documents.
 - To assign permissions to federated users, you can create an entity referred to as a **role** and define permissions for the **role**.