Common Questions

	Iaas services - Amazon Elastic Cloud or EC2 is its top VM Iaas service, Amazon storage services like S3, Elastic store, AWS container services like AWS Fargate, ECS and EKS Paas services - AWS CodeDeploy, AWS cloud90, AWS amplify Amazon API gateway, Database services like Amazon Aurora, Amazon Redshift, etc. Saas - Amazon ML service Sagemaker, AWS Data exchange, etc. To know more about AWS, check out the below video from Wiculty.
Not region specific	iam , route53, CloudFront

Storage

Amazon S3	**Amazon S3 Storage classes:** **S3 Standard:** general-purpose storage of frequently accessed data; **S3 Standard-Infrequent:** **S3 One Zone-Infrequent:** for long-lived, but less frequently accessed data **S3 Intelligent-Tiering:** for data with unknown or changing access patterns **Amazon S3 Glacier** **Amazon S3 Glacier Deep Archive:** for long-term archive and digital preservation. **Encrypting data stored on Amazon S3:** **Server-side encryption:** SSE-S3, SSE-KMS, SSE- C (customer provided keys) SSE-C,

	SSE-KMS, or **Client-Side Encryption:** a client library: **Versioning:** Versioning allows you to preserve, retrieve, and restore every version of every object stored in an Amazon S3 bucket. **S3 Access Points:** Access Points provide a customized path into a bucket, with a unique hostname and access policy that enforces the specific permissions and network controls for any request made through the access point. Each S3 Access Point is configured with an access policy specific to a use case or application, and a bucket can have hundreds of access points. **Event Notification:** Amazon S3 event notifications can be sent in response to actions in Amazon S3 like PUTs, POSTs, COPYs, or DELETEs. Notification messages can be sent through either Amazon SNS, Amazon SQS, or directly to AWS Lambda. S3 Intelligent-Tiering Query in Place: S3 Select, Amazon Athena, Redshift Spectrum **S3 Transfer Acceleration:** Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and your Amazon S3 bucket. S3 Transfer Acceleration leverages Amazon CloudFront’s globally distributed AWS Edge Locations. As data arrives at an AWS Edge Location, data is routed to your Amazon S3 bucket over an optimized network path. **Storage Class Analysis:** With Storage Class Analysis, you can analyze storage access patterns and transition the right data to the right storage class. This new S3 feature automatically identifies infrequent access patterns to help you transition storage to S3 Standard-IA. S3 Inventory: S3 Batch Operations: S3 Object Lock: Retain Until Date two Modes. **Governance Mode, Compliance Mode** Legal Hold **Replication:** Amazon S3 Replication enables automatic, asynchronous copying of objects across Amazon S3 buckets. Buckets that are configured for object

	replication can be owned by the same AWS account or by different accounts. Amazon S3 Same-Region Replication:
Amazon Elastic Block Store	Amazon EBS volume types **EBS Provisioned IOPS SSD (io2)** **EBS Provisioned IOPS SSD (io1)** **EBS General Purpose SSD (gp2)*** **Throughput Optimized HDD (st1)** **Cold HDD (sc1)** **Provisioned IOPS SSD (io2 & io1) volumes:** Provisioned IOPS SSD volumes are designed to deliver up to a maximum of **64,000** IOPS and 1,000 MB/s of throughput per volume1. **Io2: 500 IOPS** for every provisioned GB **Io1: 50 IOPS** for every provisioned GB **General purpose SSD (gp2) volumes:** 3 IOPS/GB (minimum 100 IOPS) to a maximum of 16,000 IOPS, and provide up to 250 MB/s of throughput per volume GP2 volumes smaller than 1 TB can also burst up to 3,000 IOPS. HDD-backed volumes (MB/s-intensive) **Throughput optimized HDD (st1) volumes** ST1 is backed by hard disk drives (HDDs) and is ideal for frequently accessed, throughput intensive workloads with large datasets and large I/O sizes, such as MapReduce, Kafka, log processing, data warehouse, and ETL workloads. **Cold HDD (sc1) volumes** ideal for less frequently accessed workloads with large, cold datasets. provides the lowest cost per GB of all EBS volume a maximum throughput of 250 MB/s per volume. **Amazon data lifecycle manager** for EBS snapshots Data Lifecycle Manager for EBS snapshots provides a simple, automated way to back up data stored on EBS volumes by ensuring that EBS snapshots are created and deleted on a custom schedule. Amazon EBS-Optimized instances Amazon EBS Multi-Attach
Amazon Elastic File System	Amazon EFS offers two **storage classes:**

	the **Standard storage class**, and the **Infrequent Access storage class** (EFS IA). All EFS file systems, regardless of size, can burst to **100 MiB/s of throughput.** File systems with more than 1 TiB of Standard storage can burst to 100 MiB/s per TiB of data stored on EFS Standard. **Performance mode:** **"General Purpose"** performance mode is appropriate for most file systems, and is the mode selected by default when you create a file system. **"Max I/O"** performance mode is optimized for applications where tens, hundreds, or thousands of EC2 instances are accessing the file system. **Provisioned Throughput:** **Provisioned Throughput** enables Amazon EFS customers to provision their file system's throughput independent of the amount of data stored, optimizing their file system throughput performance to match their application's needs. When you select the **default Bursting Throughput mode**, the throughput of your file system scales linearly with the amount of data stored in the Amazon EFS Standard storage class. In the default Bursting Throughput mode, you get a baseline rate of **50 KB/s per GB** of throughput included with the price of Standard storage. Using the EFS console, you can apply common policies to your file system such as disabling root access, enforcing read-only access, or enforcing that all connections to your file system are encrypted. **Access Point:** EFS Access Points simplify providing applications access to shared data sets in an EFS file system. EFS Access Points work together with AWS IAM and enforce an operating system user and group, and a directory for every file system request made through the access point.
Amazon S3 Glacier	Amazon S3 Glacier & S3 Glacier Deep Archive: **Vault:** A vault is a way to group archives together in Amazon S3 Glacier. Vault access policy: specify who has access to the vault and what actions they can perform on it. Vault Lock: Vault Lock allows you to easily deploy and enforce compliance controls on individual S3 Glacier vaults via a lockable policy (Vault Lock policy). enforce the prescribed controls to help achieve your compliance objectives. **Data retrievals:**

	Expedited: 1-5 minutes Standard: 3 – 5 hours Bulk retrievals: 5 – 12 hours Provisioned capacity unit: Provisioned Capacity guarantees that your retrieval capacity for Expedited retrievals will be available when you need it. S3 Glacier Select:
AWS Storage Gateway	**File Gateway:** NFS, SMB **Tape Gateway:** virtual tapes stored in S3, Glacier or S3 Glacier Deep Archive **Volume Gateway:** iSCSI **cached mode** **stored mode** **The File Gateway** Enables you to store and retrieve objects in Amazon S3 using file protocols such as Network File System (NFS) and Server Message Block (SMB). Objects written through File Gateway can be directly accessed in S3. File Gateway presents a **file-based interface** to Amazon S3, which appears as a network file share. **It enables you to store and retrieve Amazon S3 objects through standard file storage protocols.** **The Volume Gateway** provides block storage to your on-premises applications using iSCSI connectivity. Data on the volumes is stored in Amazon S3 and you can take point in time copies of volumes which are stored in AWS as Amazon EBS snapshots. Volume Gateway provides an iSCSI target, which enables you to create block storage volumes and mount them as iSCSI devices from your on-premises or EC2 application servers. The Volume Gateway runs in either a cached or stored mode. In the **cached mode**, your primary data is written to S3, while retaining your frequently accessed data locally in a cache for low-latency access. In the **stored mode**, your primary data is stored locally and your entire dataset is available for low-latency access while asynchronously backed up to AWS. The Tape Gateway provides your backup application with an iSCSI virtual tape library (VTL) interface, consisting of a virtual media changer, virtual tape drives, and virtual tapes. Virtual tapes are stored in Amazon S3 and can be archived to Amazon S3 Glacier or

	Amazon S3 Glacier Deep Archive. **AWS Storage Gateway**: You can have two touchpoints to use the service: the AWS Management Console and a gateway that is available as a virtual machine (VM) or as a physical hardware appliance.
Compute	
EC2	Instance types General Purpose instances | **T2**,**M5, M4** Compute Optimized instances | **C5, C4 C6g** Memory Optimized instances | **X1e, X1, R4** Storage Optimized instances | **H1, I3, I3en, D2** Accelerated Computing instances | **P3, P2, G3, F1** High Memory instances | Previous Generation instances | Instance Types: ● On-Demand Instances ● Reserved Instances ● Spot Instances Elastic Fabric Adapter (EFA) ENI vs Elastic Network Adapters (ENAs) ENI: An ENA ENI provides traditional IP networking features necessary to support VPC networking. **EFAs provide all of the same traditional IP networking features as ENAs, and they also support OS-bypass capabilities.** **Enhanced networking:** If your applications benefit from **high packet-per-second performance and/or low latency networking**, Enhanced Networking will provide significantly improved performance. consistency of performance and scalability. enhanced networking capabilities using **SR-IOV** (Single Root I/O Virtualization) HVM AMI with the appropriate drivers Enhanced networking can be enabled using one of the following mechanisms: **Intel 82599 Virtual Function (VF) interface** **Elastic Network Adapter (ENA)** **Billing and purchase options:** **Savings Plans:** When you sign up for Savings Plans, you will be charged the

discounted Savings Plans price for your usage up to your commitment.

AWS offers two types of Savings Plans:

Compute Savings Plans
EC2 Instance Savings Plans

Convertible Reserved Instances: RI is associated with a specific region, which is fixed for the duration of the reservation's term.

On-Demand Capacity Reservation

On-Demand Capacity Reservation is an EC2 offering that lets you create and manage reserved capacity on Amazon EC2.

Savings Plans, EC2 RIs, and Capacity Reservations

Use Savings Plans or Regional RIs to reduce your bill while committing to a one- or three-year term. Savings Plans offer significant savings over On Demand, just like EC2 RIs, but automatically reduce customers' bills on compute usage across any AWS region, even as usage changes.

Use **Capacity Reservations** if you need the additional confidence in your ability to launch instances. Capacity Reservations can be created for any duration and can be managed independently of your Savings Plans or RIs.

A Zonal RI provides both a discount and a capacity reservation in a specific Availability Zone in return for a 1-to-3 year commitment.

Reserved Instances
Standard RIs
Convertible RIs
Zonal RIs: When a Standard or Convertible RI is scoped to a specific Availability Zone (AZ), instance capacity matching the exact RI configuration is reserved for your use.

Instance purchasing options:

- **On-Demand Instances** –
- **Savings Plans** –
- **Reserved Instances** –
  - **Standard:**
  - **Convertible:**
- **Scheduled Instances** –
- **Spot Instances**
- **Dedicated Hosts**
- **Dedicated Instances**
- **Capacity Reservations** –

• **Spot Fleet** – A set of Spot Instances that is launched based on criteria that you specify.

Monitoring Amazon EC2:

System status checks – Loss of network connectivity, Loss of system power, Software issues on the physical host, Hardware issues on the physical host that impact network reachability

Instance status checks - Misconfigured networking or startup configuration, Exhausted memory,Corrupted file system

The following graphs are available:

- Average **CPU** Utilization (Percent)
- Average **Disk** Reads (Bytes)
- Average Disk Writes (Bytes)
- Maximum **Network** In (Bytes)
- Maximum **Network** Out (Bytes)
- Summary Disk Read Operations (Count)
- Summary Disk Write Operations (Count)
- Summary Status (Any)
- Summary **Status Instance** (Count)
- Summary **Status System** (Count)

Placement groups (Cluster, Partition and Spread)

Cluster placement groups

A cluster placement group is a logical grouping of instances within a single Availability Zone. A cluster placement group can span peered VPCs in the same Region.

Partition placement groups

Partition placement groups help reduce the likelihood of correlated hardware failures for your application. When using partition placement groups, Amazon EC2 divides each group into logical segments called partitions. Amazon EC2 ensures that each partition within a placement group has its own set of racks.

	Spread placement groups
	A spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source.
	Amazon Data Lifecycle Manager
	You can create, retain, and delete snapshots manually, or you can use Amazon Data Lifecycle Manager to manage your snapshots for you. For more information, see Data Lifecycle Manager.
	Extending a Linux file system after resizing a volume
	`sudo growpart /dev/nvme0n1 1`
	`sudo xfs_growfs -d /data`
	`Or`
	`sudo resize2fs /dev/nvme1n1`
	Amazon EBS–optimized instances
	An Amazon EBS–optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

Amazon EC2 Auto Scaling	**Scaling options** ● Maintain current instance levels at all times ● Scale manually ● Scheduled Scaling ● Dynamic Scaling **Scaling policy types:** ● **Target tracking scaling** ● **Step scaling** ● **Simple scaling** **Differences between step scaling policies and simple scaling policies** The main difference between the policy types is the **step adjustments** that you get with step scaling policies. When step adjustments are applied, and they increase or decrease the current capacity of your Auto Scaling group, the adjustments vary based on the **size of the alarm breach.**

	The main issue with simple scaling is that after a scaling activity is started, the policy must wait for the scaling activity or health check replacement to complete and the cooldown period to expire before responding to additional alarms. In contrast, with step scaling the policy can continue to respond to additional alarms, even while a scaling activity or health check replacement is in progress. **Predictive Scaling** Predictive Scaling, a feature of AWS Auto Scaling uses machine learning to schedule the right number of EC2 instances in anticipation of approaching traffic changes. Auto Scaling group (ASG): An Amazon EC2 Auto Scaling group (ASG) contains a collection of EC2 instances that share similar characteristics and are treated as a logical grouping for the purposes of fleet management and dynamic scaling. **Instance warm-up:** Warm-up value for Instances allows you to control the time until a newly launched instance can contribute to the CloudWatch metrics, so when warm-up time has expired, an instance is considered to be a part Auto Scaling group and will receive traffic. **Launch configuration:** A launch configuration is a template that an EC2 Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances such as the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping. **Launch templates:** defining a launch template instead of a launch configuration allows you to have multiple versions of a template. **Auto Scaling groups:** An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a **logical grouping for the purposes of automatic scaling and management.** **Default termination policy:** Before Amazon EC2 Auto Scaling selects an instance to terminate, it first **determines which Availability Zones have the most instances** and at least one instance that is not protected from scale in. Within the selected Availability Zone, the default termination policy behavior is as follows:

	1. Determine which instances to terminate so as to align the remaining instances to the **allocation strategy** for the On-Demand or Spot Instance that is terminating. 2. Determine whether any of the instances use the **oldest launch template** or configuration: 3. After applying all of the above criteria, if there are multiple unprotected instances to terminate, determine which instances are closest to the **next billing hour**. **Lifecycle hooks:** Lifecycle hooks let you take action before an instance goes into service or before it gets terminated. You can temporarily suspend Amazon EC2 Auto Scaling health checks by using the SuspendProcesses API. **Single ASG to scale instances across different purchase options:** You can provision and automatically scale EC2 capacity across different EC2 instance types, Availability Zones, and On-Demand, RIs and Spot purchase options in a single Auto Scaling Group.
Amazon Elastic Container Service	Amazon Elastic Container Service (Amazon ECS) is a highly scalable, high-performance container orchestration service that supports Docker containers and allows you to easily run and scale containerized applications on AWS. **Tasks:** Docker encourages you to split your applications up into their individual components. **Service:** The service scheduler helps you maintain application availability and allows you to scale your containers up or down to meet your application's capacity requirements.
AWS Elastic Beanstalk (upload the code, aws manages the deployment)	You can simply upload your code, and AWS Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and auto scaling to application health monitoring. At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.
AWS Lambda	AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume—there is no charge when your code is not running. Just upload your code, and Lambda takes care of everything required to run and scale your code with high availability. **Provisioned Concurrency** **When enabled, Provisioned Concurrency keeps functions initialized and**

	hyper-ready to respond in double-digit milliseconds. Lambda@Edge Lambda@Edge allows you to run code across AWS locations globally without provisioning or managing servers. On failure, Lambda functions being invoked **synchronously** will respond with an exception. Lambda functions being invoked **asynchronously** are retried at least 3 times. Events from Amazon Kinesis streams and Amazon DynamoDB streams are retried until the Lambda function succeeds or the data expires. Kinesis and DynamoDB Streams retain data for a minimum of 24 hours. On exceeding the throttle limit, AWS Lambda functions being invoked synchronously will return a throttling error (429 error code). Lambda functions being invoked asynchronously can absorb reasonable bursts of traffic for approximately 15-30 minutes, after which incoming events will be rejected as throttled.

Networking and Content Delivery

Amazon VPC	Amazon VPC comprises a variety of objects: **Subnet** **Internet Gateway** **NAT Gateway** **Virtual private gateway** **Peering Connection** **VPC Endpoints** **Egress-only Internet Gateway** **VPC endpoints:** Enable you to privately connect your VPC to services hosted on AWS without requiring an Internet gateway. **Gateway type endpoints** are available only for AWS services including S3 and DynamoDB. **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. **Default VPCs** The default VPC CIDR is **172.31.0.0/16.** Default subnets use **/20 CIDRs** within the

	default VPC CIDR. **Peering Connections** **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. **PrivateLink** AWS PrivateLink enables customers to access services hosted on AWS in a highly available and scalable manner, while keeping all the network traffic within the AWS network. **Service users can use this to privately access services powered by PrivateLink from their Amazon Virtual Private Cloud (VPC) or their on-premises, without using public IPs, and without requiring the traffic to traverse across the Internet.** **As a service user, you will need to create interface type VPC endpoints for services that are powered by PrivateLink.** These service endpoints will appear as Elastic Network Interfaces (ENIs) with private IPs in your VPCs. Once these endpoints are created, any traffic destined to these IPs will get privately routed to the corresponding AWS services. **As a service owner**, you can onboard your service to AWS PrivateLink by establishing a **Network Load Balancer (NLB) to front your service and create a PrivateLink service to register with the NLB.**
Amazon CloudFront	Amazon CloudFront is a fast content delivery network (CDN) service. Field-Level Encryption Access Control **Signed URLs and Signed Cookies** **Origin Access Identity (OAI)** **Geo-restriction capability** **Enabling redundancy for origins** **CloudFront also allows you to set up multiple origins** to enable redundancy in your backend architecture. You can use CloudFront's native origin failover capability to automatically serve your content from a backup origin when your primary origin is unavailable. Edge behaviors: How CloudFront communicates with your origin, customize what headers and

metadata are forwarded to your origin, create content variants with flexible cache-key manipulation, support for various compression modes, and other customizations. With built-in device detection, CloudFront can detect the device type (Desktop, Tablet, Smart TV, or Mobile device) and pass that information in the form of new HTTP Headers to your application to easily adapt content variants or other responses.

Lambda@Edge helps web developers, mobile developers and Amazon CloudFront customers run their code closer to their users.
For every origin that you add to a CloudFront distribution, **you can assign a backup origin** that can be used to automatically serve your traffic if the primary origin is unavailable.

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.

By default, if no cache control header is set, each edge location checks for an updated version of your file whenever it receives a request more than 24 hours after the previous time it checked the origin for changes to that file.

Remove an item from Amazon CloudFront edge locations: simply delete the file or use the Invalidation API

HTTP/2:

WebSocket

By using a persistent open connection, the client and the server can send real-time data to each other without the client having to frequently reinitiate connections checking for new data to exchange.

Field-Level Encryption:

HTTP cookies:

Amazon CloudFront supports the delivery of dynamic content that is customized or personalized using HTTP cookies. To use this feature, you specify whether you want Amazon CloudFront to forward some or all of your cookies to your custom origin server. Amazon CloudFront then considers the forwarded cookie values when identifying a unique object in its cache. This way, your end users get both the benefit of content that is personalized just for them with a cookie and the performance benefits of Amazon CloudFront.

Logging and reporting

Standard logs and Real-time logs.

Amazon Route 53	DNS Routing Policies **Simple routing policy** **Multivalue answer routing policy** **Failover routing policy** active/passive **Weighted Round Robin (WRR):**sending a small portion of traffic to a server on which you've made a software change. **Latency Based Routing:** LBR (Latency Based Routing): will route end users to the AWS region that provides the lowest latency. **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 That And is the key. You can assign bias to a geo proximity policy and make someone who's closer to Ohio, route to North Virginia because of assigned bias. -------------- **Traffic Flow** 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. **A traffic policy** is the set of rules that you define to route end users' requests to one of your application's endpoints. **Private DNS** Private DNS is a Route 53 feature that lets you have authoritative DNS within your VPCs without exposing your DNS records. You can associate multiple VPCs with a single hosted zone. You can associate VPCs belonging to different accounts with a single hosted zone. To enable DNS Failover for an ELB endpoint, create an Alias record pointing to the ELB and set the **"Evaluate Target Health"** parameter to true. **Route 53 Resolver** Route 53 Resolver is a regional DNS service that provides recursive DNS lookups for names hosted in EC2 as well as public names on the internet.

AWS PrivateLink	AWS PrivateLink provides private connectivity between VPCs, AWS services, and on-premises applications, securely on the Amazon network.
AWS Global Accelerator	AWS Global Accelerator is a networking service that improves the availability and performance of the applications that you offer to your global users. Today, if you deliver applications to your global users over the public internet, your users might face inconsistent availability and performance as they **traverse through multiple public networks to reach your application. AWS Global Accelerator uses the highly available and congestion-free AWS** global network to direct internet traffic from your users to your applications on AWS, making your users' experience more consistent. **AWS Global Accelerator improves application availability by continuously monitoring the health of your application endpoints and routing traffic to the closest healthy endpoints.** With Global Accelerator, you are provided two global static customer facing IPs to simplify traffic management. On the back end, add or remove your AWS application origins, such as **Network Load Balancers, Application Load Balancers, Elastic IPs, and EC2 Instances**, without making user facing changes. To mitigate endpoint failure, Global Accelerator automatically reroutes your traffic to your nearest healthy available endpoint.
Amazon API Gateway	Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. Amazon API Gateway **RESTful API**s For workloads that require API proxy functionality and API management features in a single solution, such as usage plans and API keys, API Gateway offers REST APIs. **HTTP APIs** are the best way to build APIs that do not require API management features and only need API proxy functionality. **WebSocket APIs** To build real-time two-way communication applications, such as chat apps and streaming dashboards, use WebSocket APIs.
AWS Transit Gateway	AWS Transit Gateway is a service that enables customers to connect their Amazon Virtual Private Clouds (VPCs) and their on-premises networks to a single gateway. With AWS Transit Gateway, you only have to create and manage a single

	connection from the central gateway into each Amazon VPC, on-premises data center, or remote office across your network. **Transit Gateway acts as a hub that controls how traffic is routed among all the connected networks which act like spokes.** This hub and spoke model significantly simplifies management and reduces operational costs because each network only has to connect to the Transit Gateway and not to every other network. AWS Transit Gateway AWS Transit Gateway connects VPCs and on-premises networks through a central hub. This simplifies your network and puts an end to complex peering relationships. It acts as a cloud router – each new connection is only made once.
Elastic Load Balancing	Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses. **Application Load Balancer** is best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers. **Network Load Balancer** is best suited for load balancing of TCP traffic where extreme performance is required. **Classic Load Balancer** provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Cross-zone load balancing The nodes for your load balancer distribute requests from clients to registered targets. When cross-zone load balancing is enabled, each load balancer node distributes traffic across the registered targets in all enabled Availability Zones. When cross-zone load balancing is disabled, each load balancer node distributes traffic only across the registered targets in its Availability Zone. Routing algorithm: **Application Load Balancers, Classic Load Balancers**: round robin routing

Consider using **least outstanding requests** when the requests for your application vary in complexity or your targets vary in processing capability. Round robin is a good choice when the requests and targets are similar, or if you need to distribute requests equally among targets.

Network Load Balancers: flow hash

Application Load Balancer components:

A ***listener*** checks for connection requests from clients, using the protocol and port that you configure

The **rules** that you define for a listener determine how the load balancer routes requests to its registered targets. Each rule consists of a priority, one or more actions, and one or more conditions. When the conditions for a rule are met, then its actions are performed.

Each ***target group*** routes requests to one or more registered targets, such as EC2 instances, using the protocol and port number that you specify.

Using an Application Load Balancer instead of a Classic Load Balancer has the following benefits:

- **Support for path-based routing**
- **Support for host-based routing**
- **Support for routing based on fields in the request, such as standard and custom HTTP headers and methods, query parameters, and source IP addresses**
- Support for registering **targets by IP address, including targets outside the VPC for the load balancer.**
  - Support for registering Lambda functions as targets.
- Support for the load balancer to authenticate users of your applications through their corporate or social identities before routing requests.
- Support for containerized applications. Amazon Elastic Container Service (Amazon ECS)

Listener rules

Default rules

When you create a listener, you define actions for the default rule. Default rules can't have conditions

Rule priority

Rule actions

The following are the supported action types for a listener rule:

```
Authenticate-cognito
Authenticate-oidc
Fixed-response
Forward
Redirect
```

Rule conditions

```
Host-header
Http-header
http-request-method
Path-pattern
query-string
Source-ip
```

Fixed-response actions

Forward actions

Redirect actions

Deregistration delay

Elastic Load Balancing stops sending requests to targets that are deregistering. By default, **Elastic Load Balancing waits 300 seconds** before completing the deregistration process, which can help in-flight requests to the target to complete.

Slow start mode

By default, a target starts to receive its full share of requests as soon as it is registered with a target group and passes an initial health check. Using slow start mode gives targets time to warm up before the load balancer sends them a full share of requests

Sticky sessions

Sticky sessions are a mechanism to route requests to the same target in a target group.

Network Load Balancer

A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model.

When you create a target group, you specify its target type, which determines whether you register targets **by instance ID or IP address**. **If you register targets by instance ID, the source IP addresses of the clients are preserved** and provided to your applications. If you **register targets by IP address, the source IP addresses are the private IP addresses of the load balancer nodes.**

Client IP preservation

When you specify targets by instance ID, the client IP of all incoming traffic is preserved and provided to your applications.

When you specify targets by IP address, the following conditions apply:

- If the target group protocol is TCP or TLS, client IP preservation is disabled by default.
- If the target group protocol is UDP and TCP_UDP, client IP preservation is enabled by default.

	Target security groups When you register EC2 instances as targets, you must ensure that the security groups for these instances allow traffic on both the listener port and the health check port. **Limits** ● Network Load Balancers do not have associated security groups. Therefore, the security groups for your targets must use IP addresses to allow traffic from the load balancer. ● You cannot use the security groups for clients as a source in the security groups for the targets. Instead, use the client CIDR blocks as sources in the target security groups.

Add database section

Database

Amazon Relational Database Service (RDS)	Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration **tasks such as hardware provisioning, database setup, patching and backups.**

Amazon RDS supports **Amazon Aurora, MySQL, MariaDB, Oracle, SQL Server, and PostgreSQL** database engines.

Enhanced Monitoring: For production databases we encourage you to enable **Enhanced Monitoring, which provides access to over 50 CPU, memory, file system, and disk I/O metrics**.

DB Parameter groups: A database parameter group (DB Parameter Group) acts as a "container" for engine configuration values that can be applied to one or more DB Instances.

Multi-AZ Deployments

When you create or modify your DB instance to run as a **Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous "standby" replica** in a different Availability Zone. Updates to your DB Instance are synchronously replicated across Availability Zones to the standby in order to keep both in sync and protect your latest database updates against DB instance failure. During certain types of planned maintenance, or in the unlikely event of DB instance failure or Availability Zone failure, **Amazon RDS will automatically fail over to the standby** so that you can resume database writes and reads as soon as the standby is promoted.

When running my DB instance as a Multi-AZ deployment, you can not use the standby for read or write operations.

Read Replicas

Read replicas make it easy to take advantage of supported engines' built-in replication functionality to **elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads.**

Amazon RDS for MySQL, MariaDB, PostgreSQL, Oracle and SQL Server allow you to **create up to 5 read replicas** for a given source DB instance.

Amazon RDS Proxy

Amazon RDS Proxy is a fully managed, highly available, and easy-to-use database proxy feature of Amazon RDS that enables your applications to: **1) improve scalability by pooling and sharing database connections;** 2) improve availability by reducing database failover times by up to 66% and **preserving application connections during failovers;** and 3) improve security by optionally enforcing AWS

	IAM authentication to databases, and securely storing credentials in AWS Secrets Manager.
Amazon Aurora	To scale read capacity and performance, you can add up to 15 low latency read replicas across three Availability Zones. Amazon Aurora Replicas **share the same underlying storage** as the source instance, lowering costs and avoiding the need to perform writes at the replica nodes. Amazon Aurora is designed to offer greater than 99.99% availability, **replicating 6 copies of your data across 3 Availability Zones and backing up your data continuously to Amazon S3.** **Aurora supports cross-region read replicas**. Cross-region replicas provide fast local reads to your users, and each region can have an additional 15 Aurora replicas to further scale local reads. You can choose between Global Database, which provides the best replication performance, and traditional binlog-based replication. Serverless Configuration Amazon Aurora Serverless is an on-demand, auto-scaling configuration for Aurora where the **database will automatically start-up, shut down, and scale up or down capacity based on your application's needs**. Aurora Serverless enables you to run your database in the cloud without managing any database instances. Custom Database Endpoints **Custom endpoints allow you to distribute and load balance workloads across different sets of database instances.** For example, you may provision a set of Aurora Replicas to use an instance type with higher memory capacity in order to run an analytics workload. A custom endpoint can then help you route the analytics workload to these appropriately-configured instances, while keeping other instances isolated from this workload. Multi-AZ Deployments with Aurora Replicas **On instance failure, Amazon Aurora uses RDS Multi-AZ technology to automate failover to one of up to 15 Amazon Aurora Replicas you have created in any of three Availability Zones.** Global Database For globally distributed applications you can use Global Database, where a single Aurora database can span multiple AWS regions to enable fast local reads and quick disaster recovery. Global Database uses storage-based replication **to replicate a**

database across multiple AWS Regions, with typical latency of less than 1 second. You can use a secondary region as a backup option in case you need to recover quickly from a regional degradation or outage. A database in a secondary region can be promoted to full read/write capabilities in less than 1 minute.

Fault-Tolerant and Self-Healing Storage

Each 10GB chunk of your database volume is replicated six ways, across three Availability Zones. Amazon Aurora storage is fault-tolerant, transparently handling the loss of up to two copies of data without affecting database write availability and up to three copies without affecting read availability.

Automatic, Continuous, Incremental Backups and Point-in-Time Restore

Backtrack

Backtrack lets you quickly move a database to a prior point in time without needing to restore data from a backup.

RDS Proxy Support

Aurora can work in conjunction with **Amazon RDS Proxy**, a fully managed, highly available database proxy that makes applications more scalable, more resilient to database failures, and more secure.

Cross-region replicas with Amazon Aurora: You can set up cross-region Aurora replicas using either **physical or logical replication.**

Physical replication, called **Aurora Global Database**, uses dedicated infrastructure that leaves your databases entirely available to serve your application, and can replicate to up to five secondary regions with typical latency of under a second.

Aurora MySQL also offers an easy-to-use logical cross-region read replica feature that supports up to five secondary AWS regions. It is based on single threaded MySQL binlog replication,

Yes, you can promote your cross-region replica to be the new primary from the RDS console. For logical (binlog) replication, the promotion process typically takes a few minutes depending on your workload. The cross-region replication will stop once you initiate the promotion process.

With Aurora Global Database, you can promote a secondary region to take full read/write workloads in under a minute.

You can assign a promotion priority tier to each instance on your cluster. When the primary instance fails, Amazon RDS will promote the replica with the highest priority to primary.

	You can add Amazon Aurora Replicas. Aurora Replicas in the same AWS Region share the same underlying storage as the primary instance. Any Aurora Replica can be promoted to become primary without any data loss and therefore can be used for enhancing fault tolerance in the event of a primary DB Instance failure. You can use Aurora Global Database if you want your database to span multiple AWS Regions. This will replicate your data with no impact on database performance, and provide disaster recovery from region-wide outages. **Will Aurora automatically fail over to a secondary region of an Aurora Global Database?** **No. If your primary region becomes unavailable, you can manually remove a secondary region from an Aurora Global Database and promote it to take full reads and writes.** **Amazon Aurora Multi-Master is a new feature of the Aurora MySQL-compatible edition that adds the ability to scale out write performance across multiple Availability Zones**, allowing applications to direct read/write workloads to multiple instances in a database cluster and operate with higher availability. Amazon Auro7. dx5bvra Serverless Amazon Aurora Serverless is an on-demand, autoscaling configuration for the MySQL-compatible and PostgreSQL-compatible editions of Amazon Aurora. An Aurora Serverless DB cluster automatically starts up, shuts down, and scales capacity up or down based on your application's needs.
Amazon DynamoDB	Amazon DynamoDB is a **key-value and document database** that delivers single-digit **millisecond performance at any scale.** It's a fully managed, multiregion, multimaster, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than **20 million requests per second.** **DynamoDB Accelerator (DAX):** Microsecond latency with DynamoDB Accelerator **Capacity modes: on-demand and provisioned** **Point-in-time recovery (PITR)** **Eventually consistent reads / Strongly consistent reads** **Primary key: single-attribute partition key or a composite partition-sort key**

	DynamoDB Streams **Automated global replication with global tables** DynamoDB global tables replicate your data automatically across your choice of AWS Regions and automatically scale capacity to accommodate your workloads. With global tables, your globally distributed applications can access data locally in the selected regions to get single-digit millisecond read and write performance. Encryption at rest DynamoDB encrypts all customer data at rest by default Point-in-time recovery Point-in-time recovery (PITR) helps protect your DynamoDB tables from accidental write or delete operations.PITR provides continuous backups of your DynamoDB table data, and you can restore that table to any point in time up to the second during the preceding 35 days. On-demand backup and restore Eventually consistent reads (the default) Strongly consistent reads ACID transactions A primary key can be either a single-attribute partition key or a composite partition-sort key. DynamoDB indexes a composite partition-sort key as a partition key element and a sort key element. This multipart key maintains a hierarchy between the first and second element values. For example, a composite partition-sort key could be a combination of UserID (partition) and Timestamp (sort).

Add other services like aws config