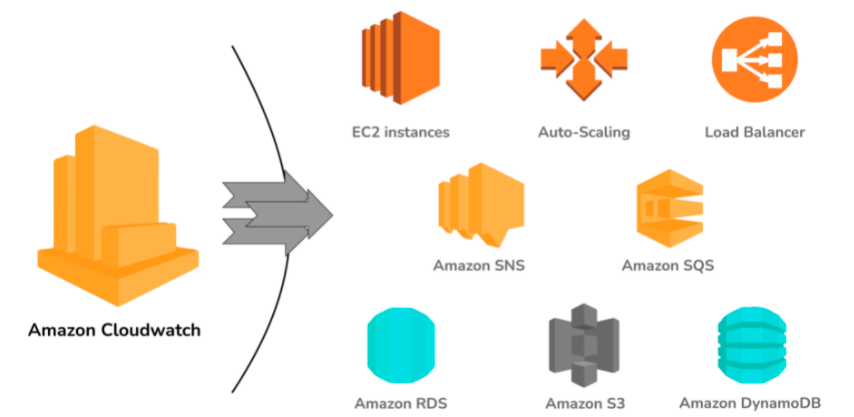
# **What is SnowBall?**

SnowBall is a small application that enables you to transfer terabytes of data inside and outside of the AWS environment.



# **What is CloudWatch?**

CloudWatch helps you to monitor AWS environments like EC2, RDS Instances, and CPU utilization. It also triggers alarms depending on various metrics.



# **What is Elastic Transcoder?**

Elastic Transcoder is an AWS Service Tool that helps you in changing a video’s format and resolution to support various devices like tablets, smartphones, and laptops of different resolutions.

# **What do you understand by VPC?**

VPC stands for Virtual Private Cloud. It allows you to customize your networking configuration. VPC is a network that is logically isolated from other networks in the cloud. It allows you to have your private IP Address range, internet gateways, subnets, and security groups.

# **DNS and Load Balancer Services come under which type of Cloud Service?**

DNS and Load Balancer are a part of IaaS-Storage Cloud Service.

# **What are the Storage Classes available in Amazon S3?**

Storage Classes available with Amazon S3 are:

* Amazon S3 Standard
* Amazon S3 Standard-Infrequent Access
* Amazon S3 Reduced Redundancy Storage
* Amazon Glacier

# **Explain what T2 instances are?**

T2 Instances are designed to provide moderate baseline performance and the capability to burst to higher performance as required by the workload.

# **What are Key-Pairs in AWS?**

Key-Pairs are secure login information for your Virtual Machines. To connect to the instances, you can use Key-Pairs which contain a Public Key and a Private Key.

# **How many Subnets can you have per VPC?**

You can have 200 Subnets per VPC.

# **List different types of Cloud Services.**

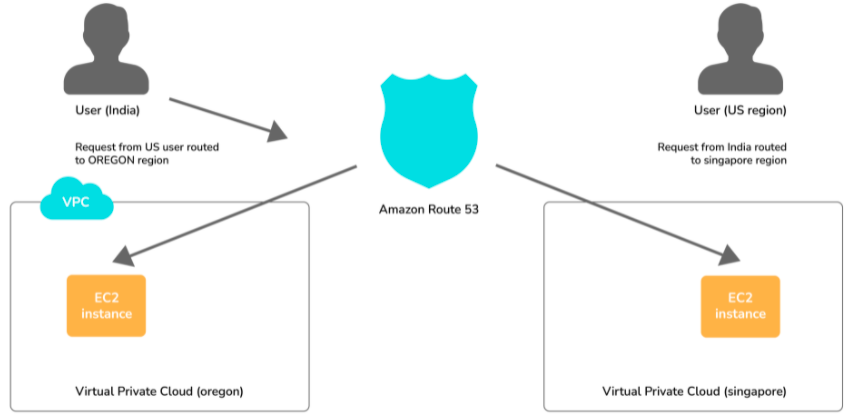
Different types of Cloud Services are:

* Software as a Service (SaaS)
* Data as a Service (DaaS)
* Platform as a Service (PaaS)
* Infrastructure as a Service (IaaS)

# **How does Amazon Route 53 provide high availability and low latency?**

Amazon Route 53 uses the following to provide high availability and low latency:

* **Globally Distributed Servers -** Amazon is a global service and consequently has DNS Servers globally. Any customer creating a query from any part of the world gets to reach a DNS Server local to them that provides low latency.
* **Dependency -** Route 53 provides a high level of dependability required by critical applications.
* **Optimal Locations -** Route 53 serves the requests from the nearest data center to the client sending the request. AWS has data-centers across the world. The data can be cached on different data-centers located in different regions of the world depending on the requirements and the configuration chosen. Route 53 enables any server in any data-center which has the required data to respond. This way, it enables the nearest server to serve the client request, thus reducing the time taken to serve.



# **What does AMI include?**

An AMI includes the following things:

* A template for the root volume for the instance.
* Launch permissions to decide which AWS accounts can avail the AMI to launch instances.
* A block device mapping that determines the volumes to attach to the instance when it is launched.

# **What are the different types of Instances?**

Following are the types of instances:

* Compute Optimized
* Memory-Optimized
* Storage Optimized
* Accelerated Computing
* General Purpose

# **What is the relation between the Availability Zone and Region?**

An AWS Availability Zone is a physical location where an Amazon data center is located. On the other hand, an AWS Region is a collection or group of Availability Zones or Data Centers.

This setup helps your services to be more available as you can place your VMs in different data centers within an AWS Region. If one of the data centers fails in a Region, the client requests still get served from the other data centers located in the same Region. This arrangement, thus, helps your service to be available even if a Data Center goes down.

# **How do you monitor Amazon VPC?**

You can monitor Amazon VPC using:

* CloudWatch
* VPC Flow Logs

# **What are the different types of EC2 instances based on their costs?**

The three types of EC2 instances based on the costs are:

**On-Demand Instance -** These instances are prepared as and when needed. Whenever you feel the need for a new EC2 instance, you can go ahead and create an on-demand instance. It is cheap for the short-time but not when taken for the long term.

**Spot Instance -** These types of instances can be bought through the bidding model. These are comparatively cheaper than On-Demand Instances. (this is a good option if you are flexible about when your applications can run and if your applications can be interrupted.)

**Reserved Instance -** On AWS, you can create instances that you can reserve for a year or so. These types of instances are especially useful when you know in advance that you will be needing an instance for the long term. In such cases, you can create a reserved instance and save heavily on costs.

# **What do you understand by stopping and terminating an EC2 Instance?**

Stopping an EC2 instance means to shut it down as you would normally do on your Personal Computer. This will not delete any volumes attached to the instance and the instance can be started again when needed.

On the other hand, terminating an instance is equivalent to deleting an instance. All the volumes attached to the instance get deleted and it is not possible to restart the instance if needed at a later point in time.

# **What are the consistency models for modern DBs offered by AWS?**

**Eventual Consistency -** It means that the data will be consistent eventually, but may not be immediate. This will serve the client requests faster, but chances are that some of the initial read requests may read the stale data. This type of consistency is preferred in systems where data need not be real-time. For example, if you don’t see the recent tweets on Twitter or recent posts on Facebook for a couple of seconds, it is acceptable.

**Strong Consistency -** It provides an immediate consistency where the data will be consistent across all the DB Servers immediately. Accordingly. This model may take some time to make the data consistent and subsequently start serving the requests again. However, in this model, it is guaranteed that all the responses will always have consistent data.

# **What is Geo-Targeting in CloudFront?**

Geo-Targeting enables the creation of customized content based on the geographic location of the user. This allows you to serve the content which is more relevant to a user. For example, using Geo-Targeting, you can show the news related to local body elections to a user sitting in India, which you may not want to show to a user sitting in the US. Similarly, the news related to Baseball Tournament can be more relevant to a user sitting in the US, and not so relevant for a user sitting in India.

# **What are the advantages of AWS IAM?**

AWS IAM enables an administrator to provide granular level access to different users and groups. Different users and user groups may need different levels of access to different resources created. With IAM, you can create roles with specific access-levels and assign the roles to the users.

It also allows you to provide access to the resources to users and applications without creating the IAM Roles, which is known as Federated Access.

# **What do you understand by a Security Group?**

When you create an instance in AWS, you may or may not want that instance to be accessible from the public network. Moreover, you may want that instance to be accessible from some networks and not from others.

Security Groups are a type of rule-based Virtual Firewall using which you can control access to your instances. You can create rules defining the Port Numbers, Networks, or protocols from which you want to allow access or deny access.

# **Explain Connection Draining.**

Connection Draining is a feature provided by AWS which enables your servers which are either going to be updated or removed, to serve the current requests.

If Connection Draining is enabled, the Load Balancer will allow an outgoing instance to complete the current requests for a specific period but will not send any new request to it. Without Connection Draining, an outgoing instance will immediately go off and the requests pending on that instance will error out.

# **What is a Stateful and a Stateless Firewall?**

A Stateful Firewall is the one that maintains the state of the rules defined. It requires you to define only inbound rules. Based on the inbound rules defined, it automatically allows the outbound traffic to flow.

On the other hand, a Stateless Firewall requires you to explicitly define rules for inbound as well as outbound traffic.

For example, if you allow inbound traffic from Port 80, a Stateful Firewall will allow outbound traffic to Port 80, but a Stateless Firewall will not do so.

# **What is a Power User Access in AWS?**

An Administrator User will be similar to the owner of the AWS Resources. He can create, delete, modify or view the resources and also grant permissions to other users for the AWS Resources.

A Power User Access provides Administrator Access without the capability to manage the users and permissions. In other words, a user with Power User Access can create, delete, modify or see the resources, but he cannot grant permissions to other users.

# **What are Recovery Time Objective and Recovery Point Objective in AWS?**

**Recovery Time Objective -** It is the maximum acceptable delay between the interruption of service and restoration of service. This translates to an acceptable time window when the service can be unavailable.

**Recover Point Objective -** It is the maximum acceptable amount of time since the last data restore point. It translates to the acceptable amount of data loss which lies between the last recovery point and the interruption of service.

# **What is the use of lifecycle hooks is Autoscaling?**

Lifecycle hooks are used for Auto-scaling to put an additional wait time to a scale-in or a scale-out event.

# **What are the policies that you can set for your user’s passwords?**

Following are the policies that can be set for user’s passwords:

* You can set a minimum length of the password.
* You can ask the users to add at least one number or special character to the password.
* Assigning the requirements of particular character types, including uppercase letters, lowercase letters, numbers, and non-alphanumeric characters.
* You can enforce automatic password expiration, prevent the reuse of old passwords, and request for a password reset upon their next AWS sign-in.
* You can have the AWS users contact an account administrator when the user has allowed the password to expire.

# **How can you secure the access to your S3 bucket?**

S3 bucket can be secured in two ways:

* **ACL (Access Control List)**  
  ACL is used to manage the access of resources to buckets and objects. An object of each bucket is associated with ACL. It defines which AWS accounts have granted access and the type of access. When a user sends the request for a resource, then its corresponding ACL will be checked to verify whether the user has granted access to the resource or not.  
  When you create a bucket, then Amazon S3 creates a default ACL which provides a full control over the AWS resources.
* **Bucket Policies**  
  Bucket policies are only applied to S3 bucket. Bucket policies define what actions are allowed or denied. Bucket policies are attached to the bucket not to an S3 object but the permissions define in the bucket policy are applied to all the objects in S3 bucket.

**The following are the main elements of Bucket policy:**

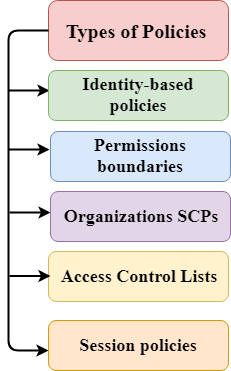
* **Sid**  
  A Sid determines what the policy will do. For example, if an action that needs to be performed is adding a new user to an Access Control List (ACL), then the Sid would be **AddCannedAcl**. If the policy is defined to evaluate IP addresses, then the Sid would be **IPAllow**.
* **Effect:** An effect defines an action after applying the policy. The action could be either to allow an action or to deny an action.
* **Principal**  
  A Principal is a string that determines to whom the policy is applied. If we set the principal string as '\*', then the policy is applied to everyone, but it is also possible that you can specify individual AWS account.
* **Action**  
  An Action is what happens when the policy is applied. For example, s3:Getobject is an action that allows to read object data.
* **Resource**  
  The Resource is a S3 bucket to which the statement is applied. You cannot enter a simply bucket name, you need to specify the bucket name in a specific format. For example, the bucket name is javatpoint-bucket, then the resource would be written as "arn:aws:s3""javatpoint-bucket/\*".

# **What are policies and what are the different types of policies**

Policy is an object which is associated with a resource that defines the permissions. AWS evaluate these policies when user makes a request. Permissions in the policy determine whether to allow or to deny an action. Policies are stored in the form of a JSON documents.

**AWS supports six types of policies:**

* Identity-based policies
* Resource-based policies
* Permissions boundaries
* Organizations SCPs
* Access Control Lists
* Session policies



* **Identity-based policies**  
  Identity-based policies are the permissions stored in the form of JSON format. This policy can be attached to an identity user, group of users or role. It determines the actions that the users can perform, on which resources, and under what conditions.  
  **Identity-based policies are further classified into two categories:**
  + **Managed Policies:** Managed Policies are the identity-based policies which can be attached to multiple users, groups or roles. There are two types of managed policies:
    - **AWS Managed Policies**  
      AWS Managed Policies are the policies created and managed by AWS. If you are using the policies first time, then we recommend you to use AWS Managed Policies.
    - **Custom Managed Policies**  
      Custom Managed Policies are the identity-based policies created by user. It provides more precise control over the policies than AWS Managed Policies.
  + **Inline Policies**  
    Inline Policies are the policies created and managed by user. These policies are encapsulated directly into a single user, group or a role.
* **Resource-Based Policies**  
  Resource-based policies are the policies which are attached to the resource such as S3 bucket. Resource-based policies define the actions that can be performed on the resource and under what condition, these policies can be applied.
* **Permissions boundaries**  
  Permissions boundaries are the maximum permissions that identity-based policy can grant to the entity.
* **Service Control Policies (SCPs)**  
  Service Control Policies are the policies defined in a JSON format that specify the maximum permissions for an organization. If you enable all the features in an Organization, then you can apply Service Control Policies to any or all of your AWS accounts. SCP can limit the permission on entities in member accounts as well as AWS root user account.
* **Access Control Lists (ACLs)**  
  ACL defines the control that which principals in another AWS account can access the resource. ACLs cannot be used to control the access of a principal in a different AWS account. It is the only policy type which does not have the JSON policy document format.

# **What is AWS Lambda?**

**AWS Lambda** is a serverless compute service provided by Amazon that runs the code in response to events and the compute resource is automatically managed.

# **What are the languages supported by AWS Lambda?**

**Java, Go, PowerShell, Node. js, C#, Python, and Ruby** code are the languages supported by AWS Lambda.

# **What is Auto-scaling in AWS Lambda?**

**Auto Scaling** is an application in AWS that helps in target tracking scaling policy that adjusts provisioned concurrency levels automatically.

# **What do you know about AMI?**

**AMI** stands for Amazon Machine Image that is used to provide the required information to launch an instance.

# **What is Serverless Architecture?**

**Serverless architecture** is a method where all the server management is done by AWS such that it doesn't need to manage an infrastructure to build or run any application.

# **List some limitations of AWS Lambda?**

Some of the limitations of AWS Lambda are:

* The disk space (ephemeral) is limited to 512 MB.
* The default deployment package size is 50 MB.
* The memory range is from 128 to 3008 MB.
* The maximum execution timeout for a function is 15 minutes

# **What are aws lambda path parameters?**

**AWS lambda path parameter** is used to test query string parameters in the AWS console.

# **What is Hybrid Cloud?**

A **hybrid cloud** is a service used to connect a private cloud with one or more public cloud services to enable communication between each distinct service.

# **What is an SQS dead-letter queue?**

An **SQS** dead letter queue is a queue that can be targeted by source queues for messages that can't be processed successfully.

# **What is an AWS Lambda application?**

An **AWS Lambda application** is a combination of Lambda functions, event sources, and other resources that work together to perform tasks.

# **What is Runtime interface emulator?**

**Runtime Interface** Emulator allows customers to locally test their Lambda function packaged as a container image. It acts as a proxy for Lambda's Runtime and Extensions APIs.

# **What is an external extension? List some Lambda runtimes external extensions?**

An **external extension** is an extension that runs as an independent process in the execution even after the invocation of function is fully processed. Some Lambda runtimes external extensions are:

* NET Core 3.1 (C#/PowerShell) ( dotnetcore3. 1 )
* Custom runtime ( provided )
* Custom runtime on Amazon Linux 2 ( provided. al2 )
* Java 11 (Corretto) ( java11 )
* Java 8 (Corretto) ( java8. al2 )

# **Explain AVX2 vectorization in Lambda?**

**AVX2** stands for Advanced Vector Extensions 2 which is a vectorization extension used for creating Lambda workloads or move existing AVX2-enabled workloads to Lambda at no additional cost

# **What events can trigger an AWS Lambda function?**

#### Lambda-based applications (also known as serverless applications) are composed of functions triggered by events. A standard serverless application consists of one or more functions triggered by events such as object uploads to Amazon S3, Amazon SNS notifications, or API actions.