Cloud Computing

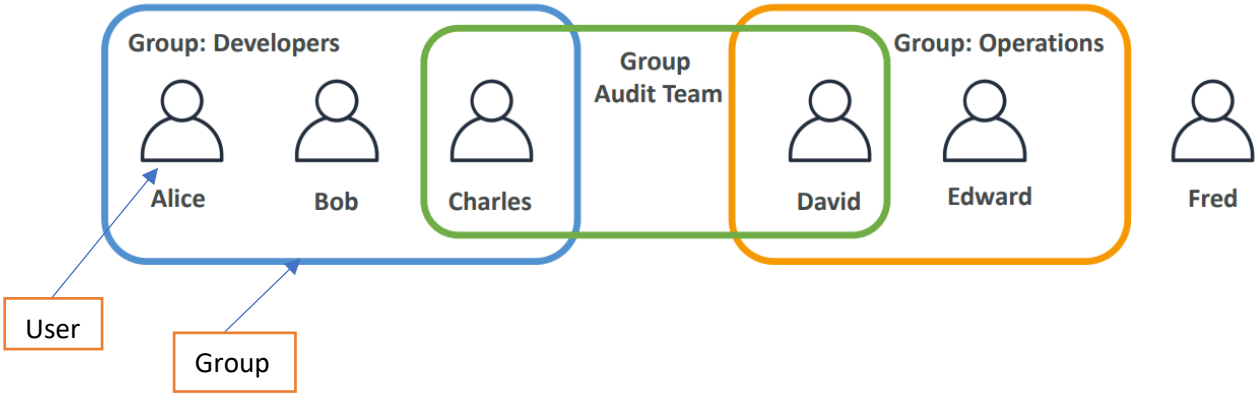
1. Users, groups, security policies, roles, permission boundaries.0

IAM: Identity Access Management

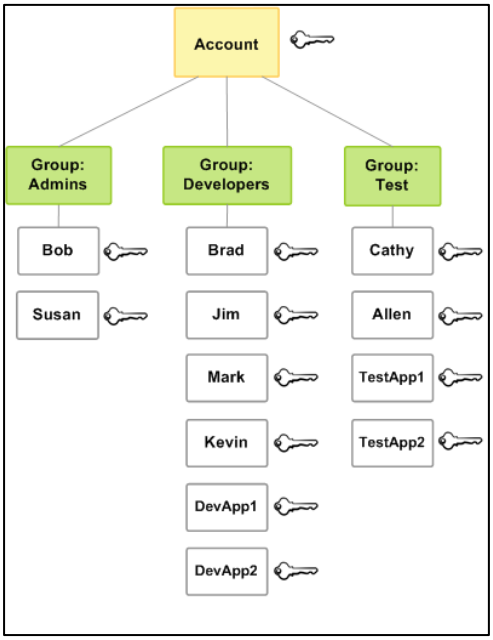
1) AWS Identity and Access Management (IAM) is a web service that helps you **securely control access** to AWS resources.
2) It is a global service
3) With IAM, you can centrally manage permissions that control which AWS resources users can access.
4) It contains Users, Group

Users:

1) An AWS Identity and Access Management (IAM) user is an **entity** that you create in AWS.
2) A user in AWS consists of a **name** and **credentials**.
3) A user is associated with a set of **security** credentials that enable them to access and perform actions on AWS resources.
4) IAM allows you to create and manage users with different levels of access to AWS resources, control their **permissions**, and monitor their activities.
5) Users can be created and managed using the AWS **Management Console**, AWS Command Line Interface (**CLI**), or AWS **SDKs**.
6) Users can be granted permissions to access AWS services and resources through policies attached to their IAM user accounts.



Groups:



1) An IAM user group is a **collection** of IAM **users**.
2) Groups can be created and managed using the AWS Identity and Access Management (**IAM**) service.
3) IAM allows you to create groups and attach **policies** that define the permissions the group members should have.
4) Groups only contain users, not other groups.
5) You can use groups to organize users based on their roles, departments, or projects.
6) IAM allows you to create groups and attach **policies** that define the permissions the group members should have.

Security Policies:

Roles:

1) An IAM role is an IAM identity that you can create in your account that has specific permissions.
2) Roles allow you to define a set of permissions for making AWS service requests.
3) Roles can be used to grant access to AWS resources across different AWS accounts.
4) IAM Roles can be used with AWS services: IAM Roles can be used with a variety of AWS services, such as EC2, Lambda, and Code Pipeline.
5) By using roles, you can improve the security of your AWS environment and simplify access management.
6) Overall, roles are a powerful tool for managing access to AWS resources and controlling permissions for service requests.

Policies and permissions in IAM:

Permission boundaries:

1) Users or Groups can be assigned JSON documents called policies.
2) These policies define the permissions of the users.
3) A permissions boundary is an advanced feature for using a managed policy to set the maximum permissions that an identity-based policy can grant to an IAM entity.

S3: Simple Storage Service

S3 Features:

1. Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance.
2. Storage classes:
3. Storage management:
   S3 Lifecycle
   S3 Object Lock
   S3 Replication
   S3 Batch Operations
4. Access management:
   S3 Block Public Access
   AWS Identity and Access Management (IAM)
   Bucket policies
   Amazon S3 access points
   Access control lists (ACLs)
   S3 Object Ownership
   Access Analyzer for S3
5. Data processing:
6. Storage logging and monitoring:
7. Analytics and insights
8. Strong consistency

Using versioning in S3 buckets:

1. Versioning is a feature in Amazon S3 that allows you to keep multiple versions of an object in the same bucket.
2. This includes all writes and deletes, so you can easily recover from unintended deletions and overwrites.

S3 Storage classes:

S3 provides multiple storage classes designed to offer different performance, durability, and cost options based on the access patterns of the data being stored.

Storage classes for frequently accessed objects:

1. S3 Standard: S3 Standard is the **default** storage class and is designed for **frequently accessed data** that requires **low latency and high throughput**.
2. S3 Intelligent-Tiering: S3 Intelligent-Tiering is designed for data with unknown or changing access patterns.
3. S3 Standard-Infrequent Access (S3 Standard-IA): S3 Standard-IA is designed for data that is accessed **less frequently** but requires rapid access when needed.
4. S3 One Zone-Infrequent Access (S3 One Zone-IA): S3 One Zone-IA is similar to S3 Standard-IA but stores data in a **single availability zone**, instead of replicating it across multiple zones.
5. S3 Glacier: S3 Glacier is designed for data archiving and **long-term storage**. It provides low-cost storage, with retrieval times ranging from minutes to hours depending on the retrieval option selected.
6. S3 Glacier Deep Archive: S3 Glacier Deep Archive is designed for data that is **rarely accessed** and needs to be stored for long periods of time.

Each S3 storage class has different pricing and availability characteristics, allowing customers to choose the most appropriate class based on their requirements and budget.

S3 Object Lifecycle policy:

Object Lifecycle policy is a feature in AWS that automates the process of moving objects between different S3 storage classes or deleting them when they are no longer needed, based on predefined rules.

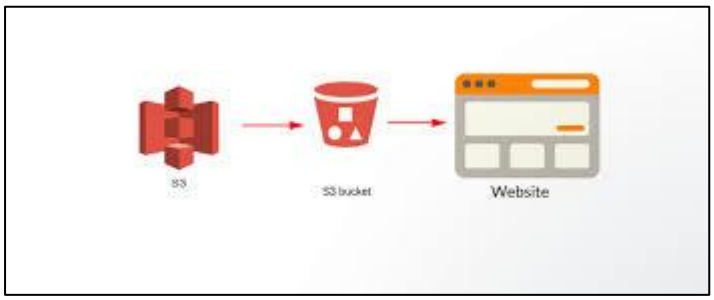
- With S3 Object Lifecycle policy, you can define rules to automatically transition objects from one storage class to another.
- To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their Amazon S3 Lifecycle.
- An S3 Lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects.
- There are two types of actions:
    a) Transition actions: These actions define when objects transition to another storage class.
    b) Expiration actions: These actions define when objects expire. Amazon S3 deletes expired objects on your behalf.

Using S3 Object Lifecycle policy can help you optimize storage costs and simplify the management of your S3 objects.

Replication policy:

Event notification:

Static website Hosting:



We can use Amazon S3 to host a static website. On a static website, individual webpages include static content.

Here are the steps to host a static website on S3:

1. Create an S3 bucket: First, create an S3 bucket with a unique name. This bucket will be used to store the static website files.
2. Configure the bucket for website hosting: Once the bucket is created, enable website hosting for the bucket by going to the Properties tab of the bucket, clicking on the Static website hosting card, and selecting the option to use this bucket to host a website.
3. Upload website files: Upload the static website files, including HTML, CSS, JavaScript, and images, to the S3 bucket.
4. Set permissions: Set permissions on the bucket and objects to allow public read access so that the website can be accessed by anyone on the internet.
5. Configure the DNS: Configure the DNS settings for your domain to point to the S3 bucket URL. You can use Amazon Route 53, AWS Certificate Manager, or other DNS providers to configure DNS settings.
6. Test the website: Test the website by accessing the domain name in a web browser.

By following these steps, you can host a static website on S3, with high availability and scalability, and low cost.

Bucket policy:

Bucket policy is a feature in AWS that allows you to define permissions for a specific S3 bucket, specifying who can access the bucket and how they can access it.

Bucket policy is a JSON-based document that can be attached to an S3 bucket, and it is used to grant or deny access to specific resources in the bucket.

Here are some key features of S3 bucket policy:

1. Access Control: S3 bucket policy allows you to control access to your bucket and objects within the bucket.
2. Resource-level permissions: With S3 bucket policy, you can specify permissions at the object level, enabling granular control over access to specific objects within the bucket.
3. Conditional statements: S3 bucket policy supports conditional statements that can be used to allow or deny access based on a set of conditions, such as the time of day, IP address range, or user agent.
4. Replication: S3 bucket policy can be used to control replication of objects between buckets, allowing you to specify which objects should be replicated, and to which bucket.
5. Cross-Account Access: S3 bucket policy allows you to grant access to other AWS accounts, enabling collaboration with other users.
6. Monitoring: S3 bucket policy can be used to monitor access to your bucket, providing detailed logging and auditing capabilities.

Overall, S3 bucket policy provides a flexible and powerful mechanism for controlling access to your S3 bucket, enabling you to secure your data and manage access at a granular level.

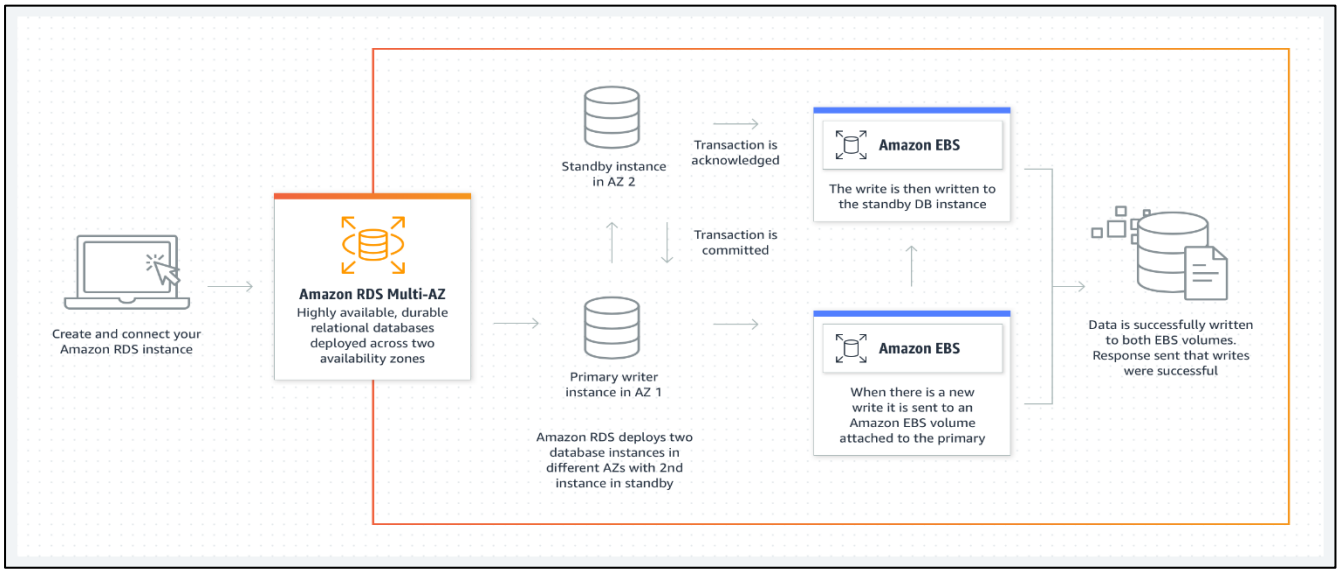
RDS: Relational Database Service

- RDS stands for Relational Database Service.
- It is a **managed** DB service for DB use SQL as a query language.
- Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a **relational database** in the AWS Cloud.
- It allows you to create databases in the cloud that are managed by AWS.
    - Postgres
    - MySQL
    - MariaDB
    - Oracle
    - Microsoft SQL Server
    - Aurora (AWS Proprietary database)

- Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud.

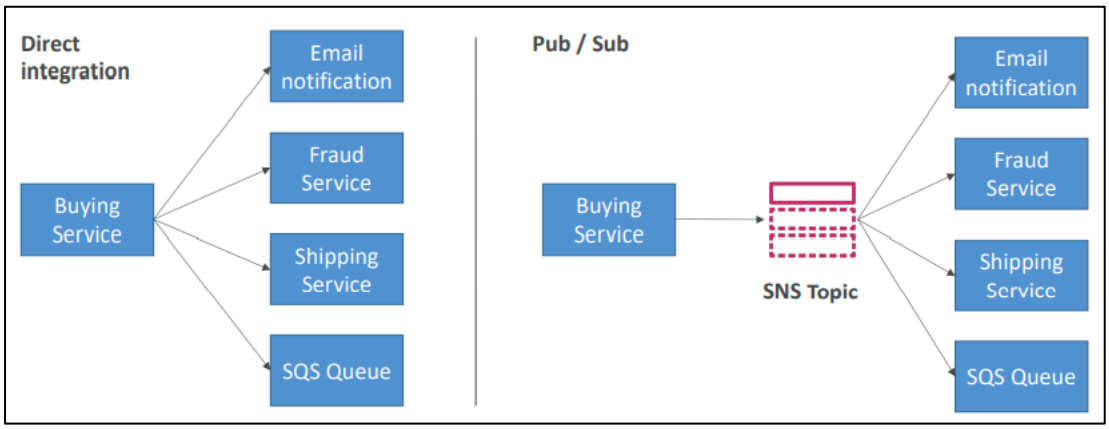
- It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

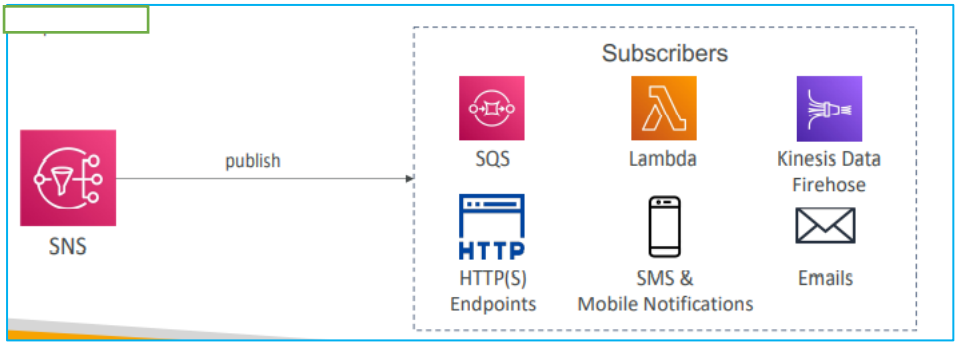
RDS MultiAZ deployment:



- Amazon Relational Database Service (RDS) Multi-AZ (Availability Zone) deployment is a high availability feature that automatically replicates a primary database instance to a secondary instance in a different Availability Zone within the same AWS Region.
- Multi-AZ deployments can have one standby or two standby DB instances.
- When the deployment has one standby DB instance, it is called a multi-AZ DB instance deployment.
- A Multi-AZ DB instance deployment has the following characteristics:
    - There is only one row for the DB instance.
    - The value of Role is Instance or Primary.
    - The value of Multi-AZ is Yes.

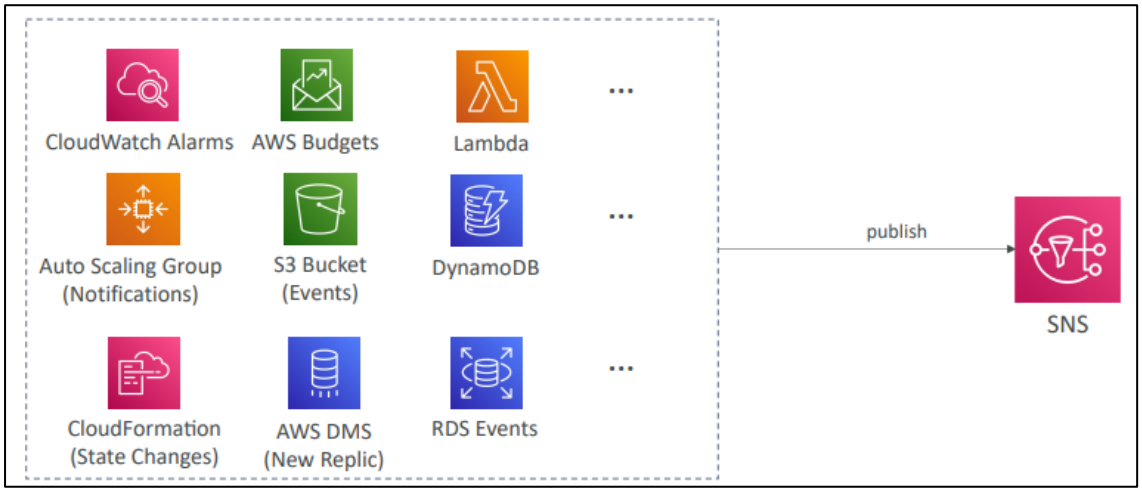
SNS: Simple Notification Service





- Amazon Simple Notification Service (Amazon SNS) is a managed service that provides message delivery from publishers to subscribers (also known as producers and consumers).
- SNS integrates with other AWS services, such as Amazon S3, AWS Lambda, and Amazon EC2, and provides access to AWS CloudFormation, AWS Management Console, and AWS SDKs.
- It also provides features for message filtering, message attributes, message encryption, and message retries.

Many AWS services can send data directly to SNS for notifications



SNS Features:

1. Application-to-application messaging:
2. Application-to-person notifications:
3. Standard and FIFO topics:
4. Message durability:
5. Message archiving and analytics:
6. Message attributes:
7. Message filtering:
8. Message security: