Table of Contents

[General Design Principles 2](#_Toc33309796)

[Identity access management 3](#_Toc33309797)

[S3 - Simple Storage Service 4](#_Toc33309798)

[EC2 6](#_Toc33309799)

[EBS 6](#_Toc33309800)

[CloudWatch 8](#_Toc33309801)

# 

## General Design Principles

The Well-Architected Framework identifies a set of general design principles to facilitate good design in the cloud:

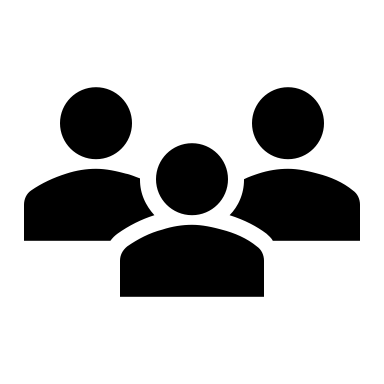
* Stop guessing your capacity needs: Eliminate guessing about your infrastructure capacity needs. When you make a capacity decision before you deploy a system, you might end up sitting on expensive idle resources or dealing with the performance implications of limited capacity. With cloud computing, these problems can go away. You can use as much or as little capacity as you need and scale up and down automatically.
* Test systems at production scale: In the cloud, you can create a production-scale test environment on demand, complete your testing, and then decommission the resources. Because you only pay for the test environment when it's running, you can simulate your live environment for a fraction of the cost of testing on premises.
* Automate to make architectural experimentation easier: Automation allows you to create and replicate your systems at low cost and avoid the expense of manual effort. You can track changes to your automation, audit the impact, and revert to previous parameters when necessary.
* Allow for evolutionary architectures: Allow for evolutionary architectures. In a traditional environment, architectural decisions are often implemented as static, one-time events, with a few major versions of a system during its lifetime. As a business and its context continue to change, these initial decisions might hinder the system's ability to deliver changing business requirements. In the cloud, the capability to automate and test on demand lowers the risk of impact from design changes. This allows systems to evolve over time so that businesses can take advantage of innovations as a standard practice.
* Drive architectures using data: In the cloud you can collect data on how your architectural choices affect the behavior of your workload. This lets you make fact-based decisions on how to improve your workload. Your cloud infrastructure is code, so you can use that data to inform your architecture choices and improvements over time.
* Improve through game days: Test how your architecture and processes perform by regularly scheduling game days to simulate events in production. This will help you understand where improvements can be made and can help develop organizational experience in dealing with events.

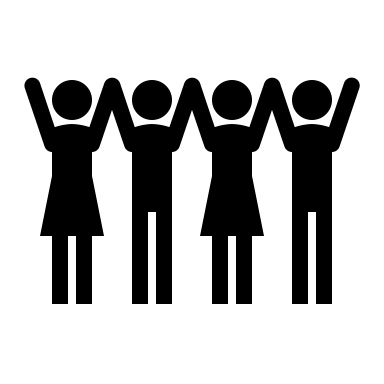
Identity Access Management

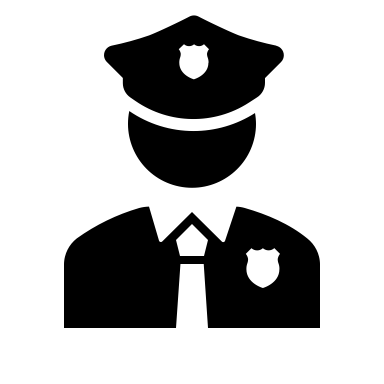
IAM allows us to manage users and their level of access to the AWS console. IAM offers the following features:

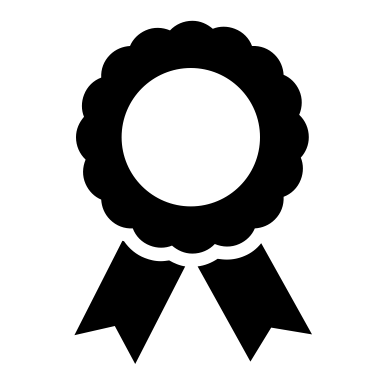
* Offers fine-grained access control to AWS resources.
* Shared access to AWS account
* Multifactor authentication
* Identity Federation - integrates with existing active directory account allowing single sign-on.
* Offers centralized control of our AWS account.
* Power User Access allows access to all AWS services except the management of groups and users within IAM.
* Provide temporary access for users/devices and services
* Allows to setup password rotation policy
* PCI DSS compliance

Key IAM terminologies:

**Users**: end users such as people, employees of an organization

**Groups**: a collection of users. Each user in the group will inherit the permission of the group.

**Polices**: Documents that are in JSON format that give permissions as to what a user/group/Role is able to do.

**Roles**: IAM entity that has specific permissions. Roles are created and assigned to AWS resources.

* Once Users and Policy Documents are created, this will be applied globally.
* newly created IAM User is granted no access to any AWS services.
* Newly created user will be able to interact with AWS using their access key ID and secret access key using the API, CLI, or the AWS SDKs.

S3 - Simple Storage Service

Key things:

* Read after Write consistency for PUTs of new objects
* Eventual Consistency for overwrite PUTs and DELETEs (can take some time to propagate)

|  |  |  |
| --- | --- | --- |
| **S3 Standard** | **S3 – IA** | **S3 one Zone - IA** |
| 99.99% availability, 99.999999999% durability, stored redundantly across multiple devices in multiple facilities, and is designed to sustain the loss of 2 facilities concurrently. | For data that is accessed less frequently but requires rapid access when needed. Lower fee than S3, but you are charged a retrieval fee. | For where you want a lower-cost option for infrequently accessed data, but do not require the multiple AZ data resilience. |
| **S3 – intelligent Tiering** | **S3 Glacier** | **S3 Glacier Deep Archive** |
| Uses ML and designed to optimize costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead. | Is a secure, durable and low-cost storage class for data archiving. Retrieval times configurable from minutes to hours. | Is S3’s lowest cost storage class where a retrieval time of 12 hour is acceptable. |

* Security and Encryption
  + Buckets are private by default
  + We can setup access control to our buckets using
    - Bucket policies – bucket wise
    - Access control list – individual level file or objects
  + Encryption is transit is achieved by
    - SSL/TLS
  + Encryption at Rest (server side) is achieved by:
    - S3 Managed keys – SSE-S3
    - AWS Key Management Service, Managed keys – SSE – KMS
    - Server-Side Encryption with Customer Provided Keys – SSE-C
  + Client-Side Encryption
  + Secure access to private files located on S3
    - Signed URLs and Signed Cookies are different ways to ensure that users attempting access to files in an S3 bucket can be authorized. One method generates URLs and the other generates special cookies but they both require the creation of an application and policy to generate and control these items. An Origin Access Identity on the other hand, is a virtual user identity that is used to give the CloudFront distribution permission to fetch a private object from an S3 bucket.
* Version control
  + Stores all versions of an object (including all writes and even if you delete an object)
  + Great backup tool
  + Once enabled, versioning cannot be disabled, only suspended
  + Integrates with lifecycle rules
  + Versioning’s MFA delete capability, which uses multi-factor authentication, can be used to provide an additional layer of security
* Lifecycle management and Glacier
  + Automates moving your object between the different storage tiers
  + Can be used in conjunction with versioning
  + Can be applied to current versions and previous versions
* Cross Region Replication
  + Versioning should be enabled both on the **source** and **destination** bucket
  + Region must be **unique**
  + File in an existing bucket are not replicated automatically
  + All subsequent updated files will be replicated automatically
  + **Delete markers are not replicated**
  + Deleting individual versions or delete markers will **not be replicated**
* Transfer Acceleration
  + Utilizes the CloudFront Edge Network to accelerate uploads to S3. Instead of uploading directly to S3 bucket, we can use a distinct URL to upload directly to edge location which will transfer that file to S3.
* CloudFront
  + A content delivery network (CDN) is a system of distributed servers (networks) that deliver webpages and other web content to a user based on the geographical locations of the user, the origin of the webpage, and a content delivery server.
    - Edge location – this is the location where content will be cached. This is separate from AWS Region/AZ.
      * Edge location are **not just READ only** – we can also write PUT to them too.
      * Objects are cached for life of the TTL (time to live)
      * Cached objects can be cleared, but additional cost will be charged
    - Origin – this is the origin of all the files that the CDN will distribute. This can be an EC2, S3, Elastic Load Balancer or Route53.
    - Distribution – this is the name given the CDN which consists of collection of edge locations.
  + Amazon’s CloudFront can be used to deliver an entire website, including dynamic, static, streaming, and interactive content using a global network of edge locations. Requests for content are automatically routed to the nearest edge location, so content is delivered with the best possible performance.
  + Two types for distribution type
    - * Web distribution – typically used for websites
      * RTPM – used for media streaming
* Storage Gateway
  + Is a service that connects an on-premise software appliance with cloud-based storage to provide seamless and secure integration between an organization’s on-premises IT environment and AWS’s storage infrastructure.
  + The service enables you to securely store data to the AWS cloud for scalable and cost-effective storage.
  + The software is available for download as a VM image that can be install on a host in a datacenter.
  + Three different types of storage Gateway:
    - File Gateway (NFS) – for flat files, stored directly on s3. Files are stored as objects in S3 buckets, accessed through a network File System (NFS) mount point.
    - Volume Gateway (iSCSI) – The volume interface presents applications with disk volumes using the iSCSI block protocol.
      * Data is saved in the cloud as Amazon EBS snapshots.
      * Snapshots are incremental backups that capture only changed clocks.
      * Two kind:
        + Stored volumes: Entire dataset is stored on site and is asynchronously backed up to S3. It lets us store primary data locally, while asynchronously backing up that data to AWS. This provides our on-premises applications with low-latency access to an entire dataset, while providing durable, off-site backups.
        + Cached volumes: Entire dataset is stored on S3 and the most frequently accessed data is cached on site. It lets us use Amazon S3 as a primary data storage while retaining frequently accessed data locally in our storage gateway.

Cached volumes minimize the need to scale our on-premises storage infrastructure, while still providing us application with low-latency access to frequently accessed data.

* + - Tape Gateway (VTL) Virtual Tape Library
      * Offer a durable, cost-effective solution to archive your data in the AWS cloud. The VTL interface it provides lets us leverage our existing tape-based backup application infrastructure to store data on virtual tape cartridges that we can create on our tape gateway.

EC2

* Network and security
  + By default, all inbound traffic is blocked.
  + Security Groups rule takes effects immediately.
  + Security Groups are STATEFUL. Anything we set on Inbound rule and Outbound rule is created automatically. If HTTP In is allowed HTTP Out is automatically allowed.
  + But network access control list is stateless. We must create both the inbound and outbound rule.
  + Multiple Security Groups can be attached to EC2 instances.
  + We may have any number of EC2 instances within a Security Group.
  + We cannot block specific IP addresses using Security Groups, instead use Network Access Control List.
  + We can specify allow rules, but not deny rules. By default, they deny everything, but we can’t go in set specific deny rules.

EBS

* + Virtual hard disk in the cloud.
  + Provides persistent block storage volumes for use with EC2 instances in the AWS cloud.
  + Each EBS volume is automatically replicated within its AZ to protect you from component failure, offering high availability and durability.
  + Five different Types of EBS storage:
    - **General Purpose (SSD)**
    - **Provisioned IOPS (SSD)**
    - **Throughput Optimized Hard Disk Drive**
    - **Cold Hard Disk Drive**
    - **Magnetic**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Solid-State Drives | | | | Hard Disk Drives | | | |
| Volume Type | General Purpose | Provisioned IOPS | Throughput Optimized | | Cold HDD | Magnetic |
| Description | Volume that balances price and performance for wide variety of transactional workloads. | Highest-performance SSD volume designed for mission-critical applications | Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads | | Lowest cost HDD volume designed for less frequently accessed workloads | Previous generation HDD |
| Use cases | Most Work Loads | Databases | Bigdata and Data Warehouses | | File Servers | Workloads where data is infrequently accessed. |
| API Name | gp2 | io1 | st1 | | sc1 | standard |
| Volume Size | 1 Gib – 16 Tib | 4 Gib – 16 Tib | 500 Gib – 16 Tib | | 500 GIB – 16 TIB | 1 GIB – 1 TIB |
| Max. IOPS/volume | 16,000 | 64,000 | 500 | | 250 | 40 - 200 |

* + EBS volume will be same AZ as the EC2 instance.
  + Volumes exist on EBS.
  + Snapshots exist on S3. Think of snapshots as a photograph of the disk.
  + Snapshots are points in time copies of volume.
  + Snapshots are incremental – this means that only the blocks that have changed since our last snapshot are moved to S3.
  + AMI’s can be created for both Volumes and Snapshots.
  + EBS volume size can be changed on the fly, including changing the and storage type.
  + Volumes will ALWAYS be in the same AZ as the EX2 instance.
  + **To move an EC2 volume from one AZ to another**, take a snapshot of it, create an AMI from the snapshot and then use the AMI to launch the EC2 in a new AZ.
  + Likewise, **to move an EC2 volume from one region to another**, take a snapshot of it, create an AMI from the snapshot and then copy the AMI from one region to the other. Then use the copied AMI to launch the new EC2 instances in the new region.
* **AMI types (EBS and instance Store)** 
  + All AMIs are categorized as either backed by Amazon EBS or backed by instance store. Storage for the Root Device (Root Device Volume) can be:
    - **Instance Store - (Ephemeral Storage) –** 
      * **Ephemeral – short-lived**
      * **Cannot be stopped. If the underlying host fails, we will lose data.**
      * For **Instance-Store Volume**: the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.
    - **EBS Backed Volumes –** 
      * For **EBS Volumes**: the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot.
      * **Can be stopped. Data will not be lost if it’s stopped.**
      * **Both instances can be rebooted, we’ll not lose data.**
      * **By default, both ROOT volumes will be deleted on termination. However, with EBS volumes, we can tell AWS to keep the root device volume.**
    - Snapshots of encrypted volumes are encrypted automatically.
    - Volumes restored from encrypted snapshots are encrypted automatically.
    - To make the root device encrypted:
      * Create a snapshot of the unencrypted root device volume
      * Create a copy of the snapshot and select then encrypt option
      * Create an AMI from the encrypted Snapshot
      * Use that AMI to launch new encrypted instances

CloudWatch

* CloudWatch: Is a monitoring service monitoring our AWS resources, as well as the applications that run on AWS.
* CloudWatch monitors performance.
* Can monitor things like:
  + Compute:
    - EC2 instances
    - Autoscaling Groups
    - ELB
    - Route53 health checks
  + Storage and Content Delivery
    - EBS Volumes
    - Storage Gateways
    - CloudFront
* CloudWatch and EC2
  + Host level metrics consist of
    - CPU
    - Network
    - Disk
    - Status Check
* AWS CloudTrail (**Don’t Confuse AWS CloudTrail with CloudWatch**) **Auditing** vs **Performance** monitoring
  + AWS CloudTrail increases visibility into our users and resources activity by recording AWS Management Console actions and API calls. We can identify which users and accounts called AWS, the source IP and address from which calls were made, and when the calls occurred.
* **Standard** monitoring is **5min** and **Detailed** monitoring is 1min