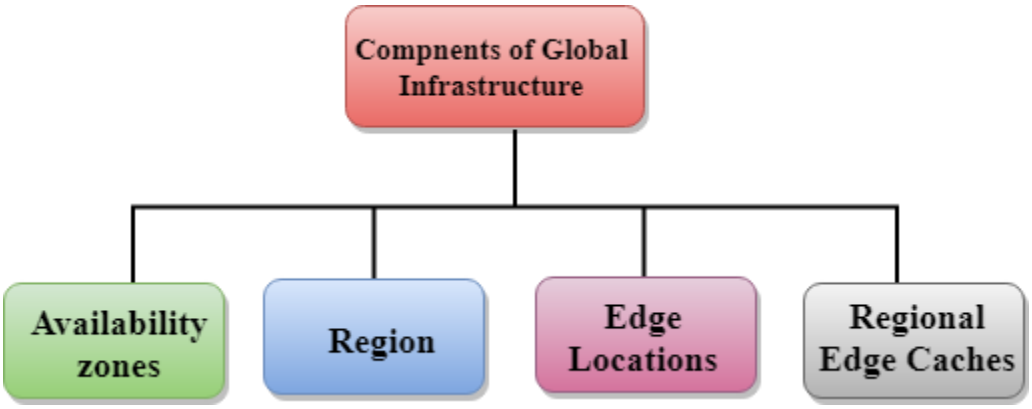
[1]AWS Global Infrastructure

- AWS is a cloud computing platform which is globally available.
- Global infrastructure is a region around the world in which AWS is based. Global infrastructure is a bunch of high-level IT services which is shown below:
- AWS is available in 19 regions, and 57 availability zones in December 2018 and 5 more regions 15 more availability zones for 2019.

The following are the components that make up the AWS infrastructure:

- Availability Zones
- Region
- Edge locations
- Regional Edge Caches



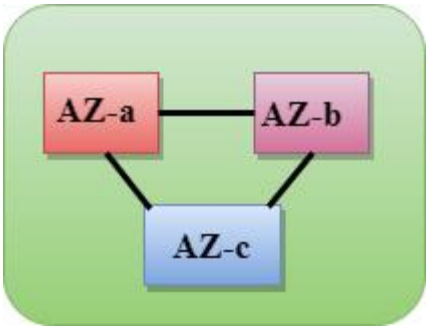
Availability zone as a Data Center

- An availability zone is a facility that can be somewhere in a country or in a city. Inside this facility, i.e., Data Centre, we can have multiple servers, switches, load balancing, firewalls. The things which interact with the cloud sits inside the data centers.
- An availability zone can be a several data centers, but if they are close together, they are counted as 1 availability zone.

1

Region

- A region is a geographical area. Each region consists of 2 more availability zones.
- A region is a collection of data centers which are completely isolated from other regions.
- A region consists of more than two availability zones connected to each other through links.



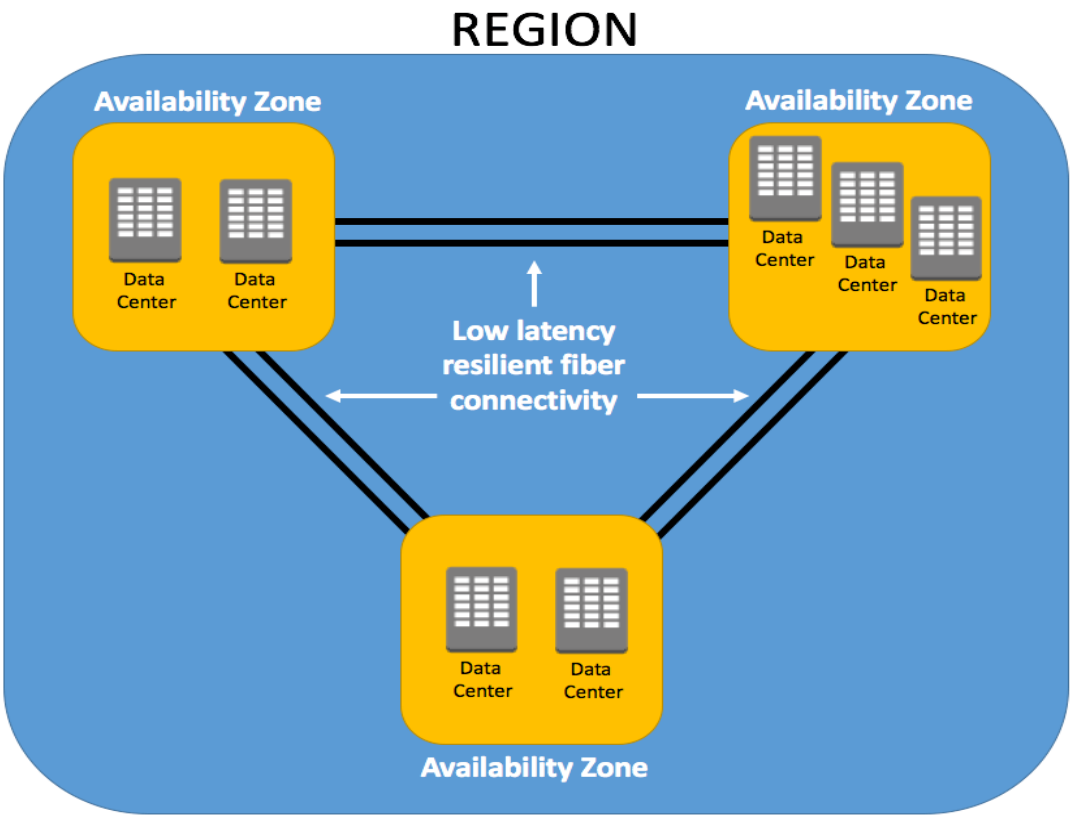
- Availability zones are connected through redundant and isolated metro fibers.

Edge Locations

- Edge locations are the endpoints for AWS used for caching content.
- Edge locations consist of CloudFront, Amazon's Content Delivery Network (CDN).
- Edge locations are more than regions. Currently, there are over 150 edge locations.
- Edge location is not a region but a small location that AWS have. It is used for caching the content.
- Edge locations are mainly located in most of the major cities to distribute the content to end users with reduced latency.
- For example, some user accesses your website from Singapore; then this request would be redirected to the edge location closest to Singapore where cached data can be read.

Regional Edge Cache

- AWS announced a new type of edge location in November 2016, known as a Regional Edge Cache.
- Regional Edge cache lies between CloudFront Origin servers and the edge locations.
- A regional edge cache has a large cache than an individual edge location.
- Data is removed from the cache at the edge location while the data is retained at the Regional Edge Caches.
- When the user requests the data, then data is no longer available at the edge location. Therefore, the edge location retrieves the cached data from the Regional edge cache instead of the Origin servers that have high latency.



What is a reservation?

A Reserved Instance is a reservation of resources and capacity, for either one or three years, for a particular Availability Zone within a region. Unlike on-demand, when you purchase a reservation, you commit to paying for *all* of the hours of the 1- or 3-year term; in exchange, the hourly rate is lowered significantly.

Furthermore, when you purchase a reservation, you're not just getting cost savings; you're also reserving the capacity that you need in that particular Availability Zone.

Amazon EC2 RI Types

With RIs, you can choose the type that best fits your applications needs.

- Standard RIs: These provide the most significant discount (up to 75% off On-Demand) and are best suited for steady-state usage.
- Convertible RIs: These provide a discount (up to 54% off On-Demand) and the capability to change the attributes of the RI as long as the exchange results in the creation of Reserved Instances of equal or greater value. Like Standard RIs, Convertible RIs are best suited for steady-state usage.
- Scheduled RIs: These are available to launch within the time windows you reserve. This option allows you to match your capacity reservation to a predictable recurring schedule that only requires a fraction of a day, a week, or a month.

Note:

When people talk about Reserved Instances, they tend to focus on EC2. However, Amazon also offers reservations for databases and for CDN; it's important to take the database side into account, as this is another easy source of cost savings.

Why make reservation?

- Savings

- The second reason primary reason for making a reservation is to reserve capacity in your chosen Availability Zone.
- Failsafe for outages

Dedicated Hosts

- You pay hourly for a single physical host based on instance type, which can then hold a certain number of instances.

Each Dedicated Host that is allocated costs the same regardless of the number of instances launched on the host

AWS Policies are of two kinds.

Identity based policies: The identity based policy is the one which can be attached directly with AWS identities like user, group or a role. IAM policy is an example of that. These policies can be AWS managed or a customer managed.

Resource based policies: Resource based policies are the ones which can be directly attached to the AWS resource like S3(called Amazon S3 bucket policy). Resource based policies are available only for certain services.

AM Policy Structure:

There are two ways you can create IAM policies from IAM web console . Visual Editor and a character based JSON policy editor. However, we focus on JSON policy which can give fine grained customised control over the resources. Once the policy is created it can be attached to user, group or role. The JSON policy document consist of following elements:

Effect –Allow or Deny access to the resource is decided by Effect (Allow/Deny)

Action — A set of service specific parameters (like “iam:CreateUser”).

Resource — Resource names (like “arn:aws:s3:::conf-* “)

Condition (Optional) — Grant conditions (like “aws:RequestedRegion”: “ap-south-1”)

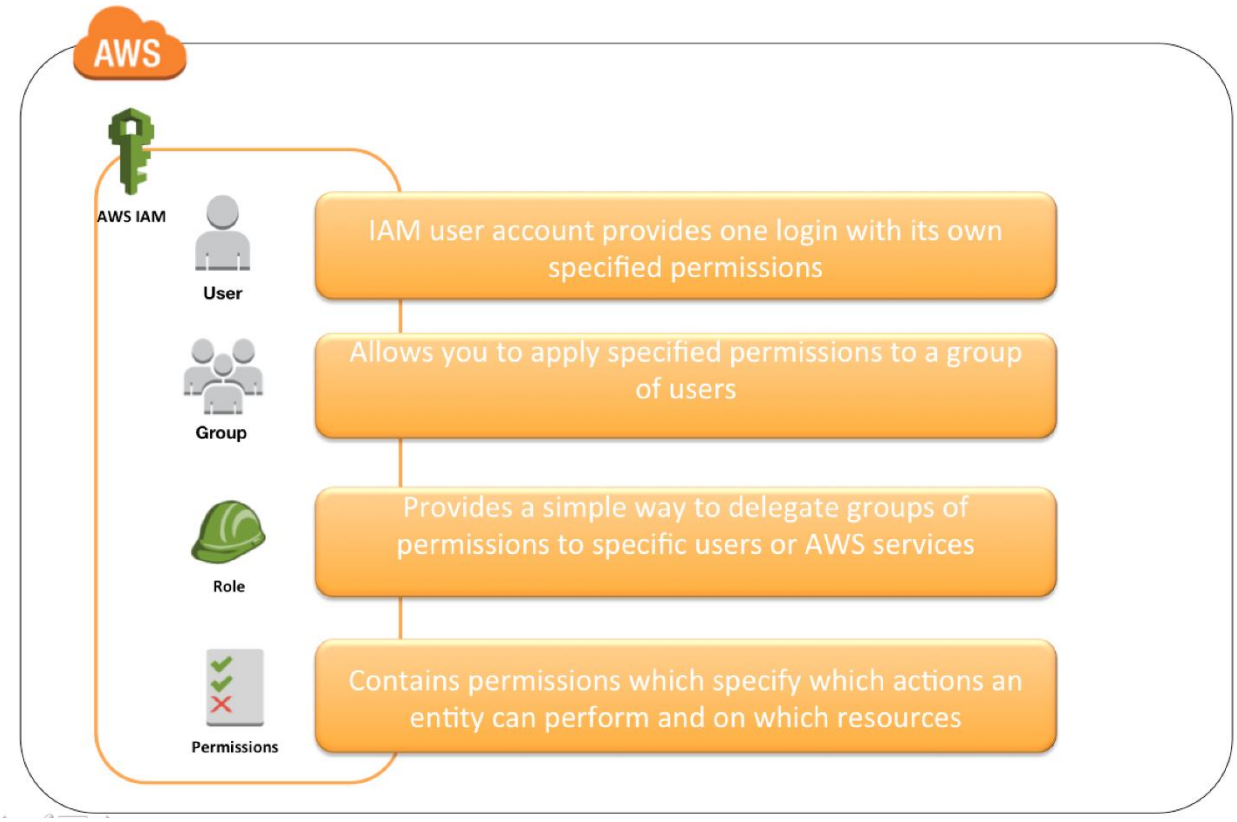
AWS Trusted Advisor

Trusted Advisor inspects the AWS environment to make recommendations for system performance, saving money, availability and closing security gaps

Trusted Advisor checks the following four categories

- **Cost Optimization**
  - Recommendations that can potentially save money by highlighting unused resources and opportunities to reduce your bill.
- **Security**
  - Identification of security settings and gaps, inline with the best practices, that could make the AWS solution less secure
- **Fault Tolerance**
  - Recommendations that help increase the resiliency and availability of the AWS solution by highlighting redundancy shortfalls, current service limits, and over-utilized resources.
- **Performance**
  - Recommendations that can help to improve the speed and responsiveness of the applications
- **Service Limits** (Circa Nov 2017 – Latest Addition)
  - Checks for service usage that is more than 80% of the service limit.
  - Values are based on a snapshot, so the current usage might differ.
  - Limit and usage data can take up to 24 hours to reflect any changes

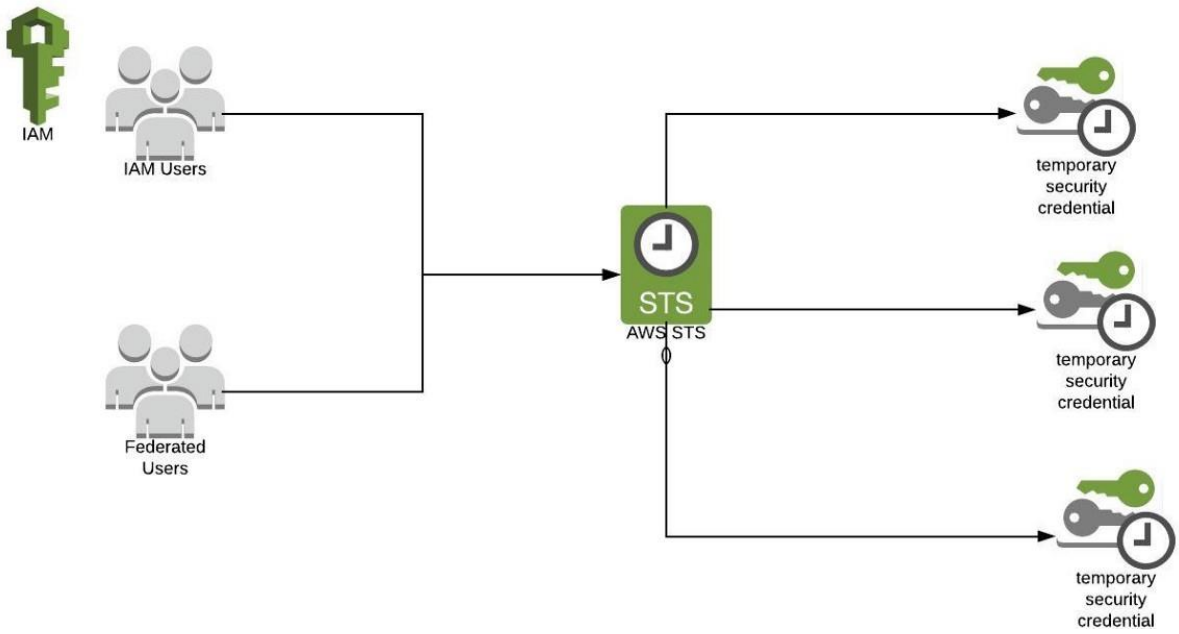
AWS IAM Identities



Credential Type	Use	Description
Passwords	AWS root account or IAM user account login to the AWS Management Console	A string of characters used to log into your AWS account or IAM account. AWS passwords must be a minimum of 6 characters and may be up to 128 characters.
Multi-Factor Authentication (MFA)	AWS root account or IAM user account login to the AWS Management Console	A six-digit single-use code that is required in addition to your password to log in to your AWS Account or IAM user account.
Access Keys	Digitally signed requests to AWS APIs (using the AWS SDK, CLI, or REST/Query APIs)	Includes an access key ID and a secret access key. You use access keys to digitally sign programmatic requests that you make to AWS.
Key Pairs	• SSH login to EC2 instances • CloudFront signed URLs	A key pair is required to connect to an EC2 instance launched from a public AMI. The keys that Amazon EC2 uses are 1024-bit SSH-2 RSA keys. You can have a key pair generated automatically for you when you launch the instance or you can upload your own.
X.509 Certificates	• Digitally signed SOAP requests to AWS APIs • SSL server certificates for HTTPS	X.509 certificates are only used to sign SOAP-based requests (currently used only with Amazon S3). You can have AWS create an X.509 certificate and private key that you can download, or you can upload your own certificate by using the Security Credentials page.

AWS Security Token Service(STS):

AWS Security Token Service(STS) that enables you to request temporary, limited privilege credentials for IAM Users or Federated Users).



Benefits

By Default STS is a global service.

- *No need to embed token in the code*
- *Limited Lifetime(15min — 1 and 1/2 day)*

Web-identity federation

Suppose you are creating a mobile app that accesses AWS resources such as a game that run on a mobile device, but the information is stored using Amazon S3 and DynamoDB.

When you create such an app, you need to make requests to the AWS services that must be signed with an AWS access key. However, it is recommended not to use long-term AWS credentials, not even in an encrypted form. An Application must request for the temporary security credentials which are dynamically created when needed by using web-identity federation. These temporary security credentials will map to a role that has the permissions needed for the app to perform a task.

With web-identity federation, users do not require any custom sign-in code or user identities. A User can log in using the external identity provider such as login with Amazon, Facebook, Google or another OpenID. After login, the user gets the authentication token, and they exchange the authentication token for receiving the temporary security credentials.

LIMITS:

Resource	Default Limit
Customer managed policies in an AWS account	1500
Groups in an AWS account	300
Roles in an AWS account	1000
Managed policies attached to an IAM role	10
Managed policies attached to an IAM user	10
Virtual MFA devices (assigned or unassigned) in an AWS account	Equal to the user quota for the account
Instance profiles in an AWS account	1000
Access keys assigned to an IAM user	2
Access keys assigned to the AWS account root user	2

Access keys assigned to the AWS account root user	2
Aliases for an AWS account	1
Groups an IAM user can be a member of	10
IAM users in a group	Equal to the user quota for the account
Users in an AWS account	5000
Managed policies attached to an IAM group	10
MFA devices in use by an IAM user	1
MFA devices in use by the AWS account root user	1
SSH public keys assigned to an IAM user	5
Tags that can be attached to an IAM role	50

A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. When you create a permissions policy to restrict access to a resource, you can choose an *identity-based policy* or a *resource-based policy*.

Identity-based policies are attached to an IAM user, group, or role. These policies let you specify what that identity can do (its permissions). For example, you can attach the policy to the IAM user named John, stating that he is allowed to perform the Amazon EC2 RunInstances action. The policy could further state that John is allowed to get items from an Amazon DynamoDB table

named MyCompany. You can also allow John to manage his own IAM security credentials. Identity-based policies can be managed or inline.

Resource-based policies are attached to a resource.

Permissions Boundaries for IAM Entities

AWS supports *permissions boundaries* for IAM entities (users or roles). A permissions boundary is an advanced feature in which you use a managed policy to set the maximum permissions that an identity-based policy can grant to an IAM entity. When you set a permissions boundary for an entity, the entity can perform only the actions that are allowed by both its identity-based policies and its permissions boundaries.

Note:

- Groups cannot be nested.
- Always Root administrator should grant permissions for modifying passwords to sub users.