<https://aws.amazon.com/certification/our-certifications/>

<https://linuxacademy.com/cp/modules/view/id/44>

**AWS Architecture and Terminology**

* Each region consists of set of independently separated data centers known as “availability zones”.
* A resource available within the “available zone”, is available to other “availability zones” with in the same region.
* Availability zones work together and share resources with in a region to make a up a collection of AWS resources.
* Properly designed applications will use multiple availability zones for “Fault Tolerance Fail Over”.
* When designing our architecture as Fault tolerance and high availability we need to make use of availability zones. These availability zones have direct low latency connection between each availability zone with in a region, but each *AZ* in a region is isolated to ensure fault tolerance.
* Some AWS services work globally and not within a specific region. For example, users created within IAM will work across regions.
* **Edge Location:** It is a AWS data center which doesn’t contain AWS services. It is used to deliver contents to parts of the world. An example is CLOUD FRONT which is CDN. It is used to reduce space/time/latency for a request by an end user from part of the world that doesn’t have any region. When a request is made form that part of the world, it routes to nearest edge location where data is already cached to serve.
* **Scalability:** Ability of a system to scale up (expand) and scale down (contract) per work load demands.
  + Resilient
  + Operationally Efficient
  + Cost effective as service grows
* **Fault Tolerant:** Ability of a system to operate without interruption in the event of service failures.
  + Auto Scaling
  + Route 53
  + Availability Zones
  + Multiple Regions
* **Elasticity:** Ability for an infrastructure to adapt up and adapt down automatically to given load. This is the fundamental property of cloud.
  + EC2, AMI, RDS, Route 53, Auto Scaling, Bootstrapping
  + Proactive Cycle Scaling: scale out based off “known” periods.
  + Proactive Event Based Scaling: scale out in anticipation of increase demand.
  + Auto scaling based on demand: Scale out based on metrics such as CPU Utilization, network utilization etc. This is considered horizontal scaling vs vertical scaling.
* **AWS Services:** Services are grouped as below categories.
  + Compute and Networking
  + Storage and Content Delivery
  + Database Services
  + Analytics
  + App Services
  + Deployment Services
  + Management Services

A properly architected application will make use of multiple types of services. A focus for automation, fault tolerance, disaster recovery and high availability should always be able to use of multiple AWS services.

## Compute and Networking Services

## EC2 (Elastic Compute Cloud): Features of EC2

## Auto Scaling

## Elastic Load Balancer

## EBS Volumes

## Virtual Private Cloud

## Amazon Route 53

## EC2:

## Provides scalable virtual servers in the cloud. These servers can run on different operating systems.

## An EC2 virtual server is known as an “instance” and can be made up of different instance types and sizes.

## Pricing Models:

## Reserved Instances

## On-Demand Instances

## Spot Instances

## Auto Scaling:

## A service in AWS to increase the number of instances on-demand based on certain metrics. If your application demand increased un-expectantly auto scaling can scale up to meet the demand and then stop instances as soon as the demand decreases. This is known as “elasticity” in the AWS environment.

## Elastic Load Balancer:

## Load balancing is a common method for distributing traffic among servers in the IT environment.

## The Elastic Load Balancer is another service by AWS EC2 that allows you to add instances to the elastic load balancer and distribute traffic among those instances.

## The elastic load balancer can send traffic to different instances in different availability zones and should often be used with auto scaling and designing for fault tolerance.

## ELB uses Round Robin DNS for routing.

## Route 53:

## Route 53 is a domain management service by AWS.

## Route 53 will host the internal and external DNS for your application environment It is used commonly with ELB to direct traffic from the domain to the Elb.

## Amazon Machine Image(AMI):

## AMI is a template that contains a pre-built software configuration.

## Amazon Machine Images are used with Auto Scaling and Disaster recovery.

## Anytime if a EC2 is started it fires up from AMI. AMI has below storage types.

## Instance Store-backed Instances (Ephemeral Storage):

## Block level temporary storage over the life of an instance.

## Lives for as long as your instance is NOT turned off/shutdown. The temporary storage will not be lost if it is rebooted, but most importantly the temporary storage will be lost when the instance has been turned off or shutdown.

## EBS Backed Instance (Elastic Block Store)

## Can be attached to an EC2, it is attached over the network through network cables. It is network attached connection, it is not a direct connection.

## EBS volume needs to live in same availability zone as the EC2 instance we want to attach with.

## Easy to backup with snapshots stored on Amazon S3.

## Can be as small as 1GiB and 16,384GiB (16Tib) in size.

## Cannot be attached to instances in a different availability zone.

## Can only be attached to one instance at a time.

## Allows for point in time snapshots.

## Up to the customer to manage the software level for security on instances.

## Security groups.

## Firewalls in Linux (IP tables, Firewalld, etc).

## EBS encryption provided by AWS.

## Snapshots can also use EBS encryption.

## AWS EBS encryption utilizes AWS Key Management Service.

## Additional encryption can be to encrypt the entire file system using an encrypted file system.

## EBS encryption is only available on larger instance types and it is suggested to use an encrypted file system on EBS if using an instance size smaller than M3.

## So, instances smaller than M3, should not use EBS encryption.

## Apply SSL Cert to the ELB (Elastic Load Balancer). For example, when hosting a web application, instead of signing certificate to all EC2 instances we can sign it to an ECB (Elastic Load Balancer) and it gets applied to all of our instances.

## AWS Manages the hypervisor and physical layer of security for EC2.

## DDOS protection.

## Port scanning protection (not allowed even in your own environment without permission from AWS).

## Ingress network filtering.

## VPC (Virtual Private Cloud) is one of the core components of AWS and no application should be designed without it.

## VPC allows for the isolation of AWS resources in the cloud. Resources fired up in a single VPC will be part of the same network and can communicate internally. However, if multiple VPCs are used to provision resources then resources in one VPC are completely isolated from the other VPC by default. Resource sharing between VPCs in the same region can be allowed with VPC peering.

## VPC does not cost only the resources within the VPC are what costs. So, setting up an VPC is free of cost but firing up an EC2 instances with in VPC will cost.

## EC2-Classic classic is a deprecated service by AWS. Some accounts that have been around for long periods of time are still using the service. However, EC2 classic instances do not belong to a VPC (can be a security issue) and have certain limitations.

## Route 53:

## Route 53 is a DNS hosting solution provided by AWS. You can not only host the DNS for domains but can now also register and transfer domains to AWS as the domain authority.

## Route 53 manages external DNS for domain routing www.domain.com to the proper AWS resources such as a CloudFront distribution, ELB, EC2 instance, or RDS server. (Not a comprehensive list).

## Route 53 can also be used to manage internal DNS for custom internal hostnames within a VPC as long as the VPC is configured for it i.e., we can define a hostname for a specific instance.

## Latency, GEO, basic, and failover routing policies allow for region to region fault tolerant and architecture design.

## Failover to S3 or CloudFront (if website bucket hosting is enabled)

## Storage and Content Delivery Services

## Amazon S3:

## Infinitely scalable and highly available static hosting solution used to store file or back up software for enterprise applications.

## It can act as static web site hosting application. We can fail over to basic html if for some reasons our highly available web application goes down.

## Simple Storage Service (S3) is an object storage service from AWS.

## It can not only serve objects through a CDN to CloudFront, manage access to specific objects, enable versioning, and lifecycle policies, but it can also serve static HTML files with Route 53.

## It is a simple key-value store designed for unlimited object storage.

## All the objects stored inside a S3 synchronized in all of the available zones within a designated region.

## Designed for “11 nines” (99.999999999%) durability and 99.99% “availability”.

## Charges based off of per gig storage as well as data sent out of the region.

## Data transfer from S3 to an EC2 instance within the same region is free.

## Cost decreases as it scales.

## S3 objects can be encrypted using the S3 encryption option as well as data sent to and from end points are encrypted using the HTTPS protocol.

## Most commonly used for file storage (is also a hybrid solution when used with AWS storage gateway), delivering static content, backups and archiving with Amazon Glacier.

## Bucket names are unique across the entire S3 design (all regions included) i.e., regardless of region only one bucket can exist with same name. (it is like registered domain name across AWS)

## RRS (Reduced Redundancy Storage)

## Lifecycle policies and object versioning

## Pay for each version of the object.

## Unlimited versions.

## Needs to be enabled

## Versioning and lifecycle polices can work together for an automated backup and archiving solution.

## Lifecycle policies with Amazon Glacier

## Amazon Glacier:

## It is an archival storage type.

## Used for data not used frequently.

## Integrates with Amazon S3 lifecycle polices for easy archiving.

## .01/gig per month.

## Amazon Storage Gateway:

## Connects local data center software appliances to cloud based storage such as Amazon S3.

## Gateway-Cached Volumes

## Create storage volumes and mount them as iSCSI devices on the on-premise servers.

## The gateway will store the data written to this volume in Amazon S3 and will cache frequently access data on-premise in the storage device.

## Gateway-Stored Volumes.

## Store all the data locally in storage volumes

## Gateway will periodically take snapshots of the data as incremental backups and stores them on Amazon S3.

## Amazon Import/Export:

## AWS Import/Export gives the ability to take on-premise data and physically snail mail it to AWS.

## AWS will import the data to either S3, EBS, or Glacier within one business day of the physical device arriving at AWS.

## Benefits:

## Off-site backup policy.

## Quickly migrate LARGE amounts of data to the cloud.

## Disaster recovery (AWS will even take s3 data and ship it back to you)

## Amazon Relational Database Service (RDS)

## Relational databases are databases that organize stored data into tables.

## The associated tables have defined relationships between them.

## Amazon RDS is a fully managed database service for relational databases. This means that access to the underlying operating system is not allowed and software patches and management are handled by AWS.

## Databases Supported By RDS: MySQL, PostgreSQL, Oracle, SQL (MS SQL Server) , Aurora.(from AWS)

## What is Aurora?

## Home grown Relational Database forked, and fully compatible with MySQL. It has five times better performance then MySQL and a lower price point than commercial databases.

## Amazon ElastiCache:

## ElasticCache is a fully managed, in-memory cache engine.

## Available engines that power ElastiCache are Memcached and Redis. It is used to improve performance by caching results of queries, managing web sesssions, and caching dynamically generated data.

## Stores this high queries inside of an elastic cache cluster.

## Generally, the application needs to be built to work with either Redis or Memcached.

## Amazon DynamoDB:

## DynamoDB is a NoSQL fully managed database service provided by AWS. It is similar to MongoDB but is a home-grown AWS solution.

## Fully managed NoSQL service

## Service manages all provisioning and scaling of underlying hardware i.e., we can manage and increase our input output writing.

## Fully distributed and scales automatically

## Built as a fault tolerant highly available service, so in backend it synchronizes the data across all of the availability zones within the region that dynamo db has created in.

## Specify required throughput capacity and DynamoDB does the rest.

## Easily integrates with other services such as Elastic MapReduce, i.e., can be easily moved over to Hadoop Map reduce cluster from dynamo db. (Data Analytics)

## Amazon Redshift, Amazon Redshift is a petabyte-scale data warehousing service.

## Analytics Services

## Amazon Elastic MapReduce (EMR):

## Elastic MapReduce is a Hadoop clustering tool that makes it easy to manage and integrate with Hadoop clusters. Hadoop is used for big-data analytics and through Elastic MapReduce. It can integrate easily with other services such as Redshift and DynamoDB for data analytics.

## EMR is a service that spins up EC2 instances which allows the user full access to the underlying operating system unlike RDS, DynamoDB and ElastiCache.

## App Services

## Amazon Simple Work Flow Service (SWF):

## Long term processing workflow solution.

## Track work flow executions.

## AWS control panel ability to monitor task work flow.

## Consistent execution i.e., no duplicates and guaranteed to go through workflow process.

## Scalable parallel EC2 processing.

## Service can be used with on premise servers.

## Guarantees execution of work flow

## Amazon Simple Queue Service (SQS):

## Decouple infrastructure systems.

## Auto scale based off queue size.

## Guarantees delivery of “at least” 1 message but does not guarantee no duplicates.

## Scalable and highly available by design.

## Does not guarantee message order but does attempt “best effort” order delivery.

## Amazon Simple Notification Service (SNS):

## Coordinates and manages the delivery or sending of messages to specific end points. This service can be used for publishing IOS/Android app notifications, gluing together automation based off of sent notifications.

## End points can be as SQS, Email, Email-json, SMS, HTTPS, HTTP, Application.

## Deployment Services

## Amazon Elastic Beanstalk:

## Easily deploy complete application environments automatically. Integrates with the Elastic Load Balancer, Auto Scaling, EC2, and additional AWS

## Key benefit for developers who are building applications but do not have the technical knowledge for building application environments.

## Includes basic configurations such as web applications and worker instances.

## Support for Docker containers.

## Easily deploy “dev/test/qa/production” environments with the EB command line tool which integrates with gitrepositories.

## Amazon CloudFormation:

## Everything in AWS is an API accessible through an SDK, Command Line Tools, or the console.

## CloudFormation is a tool that allows you to “code” your infrastructure and deploy resources based off of a pre-build template. This gives the advantage of easy backup and disaster recovery and even version controlling your AWS infrastructure!

## For example, with a template and the template code you can build EC2 instances that belong to an Elastic Load Balancer and a Route 53 entry to your elastic load balancer all with code!

## Management Services

## Amazon Management Services:

## IAM (Identity Access Management)

## Web service that allows managing permissions to AWS resources.

## Can define resource level permissions and API call permissions based off user/group/resource.

## API keys and roles are used to manage access to AWS API to integrate with the SDKs.

## CouldTrail

## CloudTrail is an API logging service that logs ALL api calls made to AWS. It does not matter if the API calls from the command line, SDK, or console. This helps when addressing security concerns and even seeing what users on AWS performed certain accessed in your environment.

## CloudWatch

## Used to monitor AWS services such as EC2.

## Integrates very well with EC2 and helps provide centralized logging and performance metrics into instances such as CPU usage, Network Usage and more.

## Auto Scaling is heavily used with CloudWatch. For example, auto scale more worker EC2 instances if a queue size becomes too large.

## Directory Services

## Allows the ability to easily connect on premise Microsoft Active Directory with and AD connector. Also, has the ability to setup and operate new directories within the AWS cloud using simple AD.