AWS Certified

Cloud Practitioner

* Sony Shrestha

AWS allows you to share a single server with other AWS customers using virtualization.

Cloud computing services are delivered via the Internet and managed through web browser.

**Cloud Terminology**

1. Availability
2. Agility
3. Durability
4. Elasticity
5. Availability

* always up and running

1. Agility

* high speed

1. Durability

* data is not corrupted

1. Elasticity

* auto scaling, scale up and scale down

**6 advantages of Cloud Computing**

1. Go global in minutes
2. Stop guessing capacity
3. Stop spending money running and maintaining data centers
4. Increase speed and agility
5. Benefit from massive economic of scale
6. Trade capital expense for variable expense
7. Go global in minutes

* You can deploy your applications around the world at the click of a button.

1. Stop guessing capacity

* Start with what you want then scale-up or scale-down

1. Stop spending money running and maintaining data centers

* No need to manage servers
* Focus on your applications instead of managing hardware

1. Increase speed and agility

* High speed

1. Benefit from massive economic of scale

* Huge discount

1. Trade capital expense for variable expense

* Pay for what you use instead of making huge upfront investments

**Cloud Computing Models**

1. IaaS
2. PaaS
3. SaaS
4. IaaS

* Infrastructure as a Service
* It is the basic building block that you can rent like an EC2 instance.
* Eg: Web hosting

1. PaaS

* Platform as a Service
* Mostly used by developers to build application.
* Eg: Storefront website

1. SaaS

* Software as a Service
* It is complete product or application that you can rent.
* Eg: Email Provider

**Cloud Deployment Models**

1. Private Cloud
2. Public Cloud
3. Hybrid Cloud
4. Private Cloud

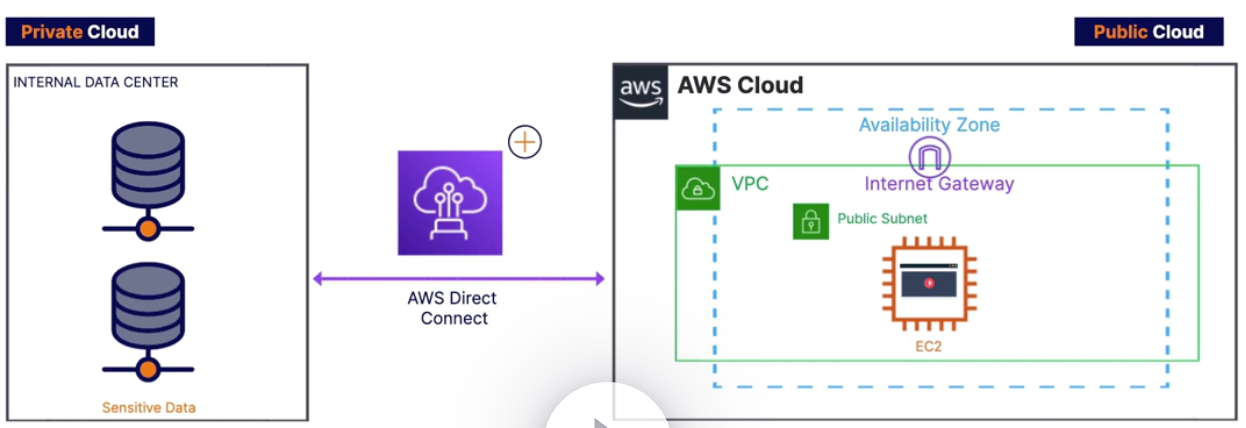
* Called on-premises
* Exists in your internal data center. Everything runs on internal data center
* Does not offer advantage of cloud computing
* There is increased level of security

1. Public Cloud

* Offered by AWS
* Offers advantages of cloud computing

1. Hybrid Cloud

* Combination of private and public cloud
* Highly sensitive data resides in private cloud, application that reads these data run in AWS and they communicate with each other using a service provided by AWS called AWS Direct Connect.



Note:

Hybrid deployments are supported by Direct Connect.

**AWS Global Infrastructure**

1. Availability Zone
2. Region
3. Edge Location

Note:

240 countries

80 availability zone

25 regions

Data Center

Collection of servers

1. Availability Zone

* Consists of one or more physically separated data centers, each with redundant power, networking, and connectivity, housed in separate facilities.
* Characteristics of Availability Zone
  + They are physically separated
  + They use separate power grids
  + They are connected among themselves within a region through low-latency links
  + Fault tolerance- If one Availability zone gets out of service, other should not get impact
  + Allows for high availability
* An availability zone is associated with a single region
* Availability Zone contains server you are renting, and it is where you deploy your applications

1. Region

* Physical location
* Collection of multiple availability zones
* When deploying your application, select region that is closest to users. It improves performance and availability
* Characteristics of Region
  + Fully independent and isolated- if one region gets impact, other will not be affected
  + Resource and service specific- resources are not automatically replicated across regions.

1. Edge Location

* Mini data center
* Cache content for faster delivery of content to users
* There are more edge locations than availability zones and regions
* It is not used to launch resources like EC2 instance. It is just used to cache content.
* Reduces latency and speeds up delivery of your application
* Related to Content Delivery Network (CDN) and Amazon CloudFront

**Ways to access AWS**

1. AWS Management Console
2. AWS Command Line Interface (CLI)
3. AWS Software Development Kit (SDK)
4. AWS Management Console

* Allows you to access your AWS account and manage resources using web browser
* Root user is created when we initially sign up for your account. He has unrestricted access to your account, and it cannot be restricted.
* For best practice, use root user for first time access. After that create separate users for day-to-day activities.
* For best practice, protect root user with MFA (Multi-Factor Authentication) This is where a code is sent to your phone that you then have to enter in order to officially sign into your accounts. MFA is sometimes called 2-factor authentication because you have to enter two things your password and code sent to your phone.
* Following tasks can only be performed by root user:
  + Deleting account
  + Changing email address
  + Changing service plan

1. AWS Command Line Interface (CLI)

* It is called programmatic way to access your AWS account
* Allows you to access AWS account through a terminal or command window on your local laptop
* Mostly used by developers
* Some new features are available via command line before console
* When you set up CLI locally, you will have to do some configurations like generating secret key and access key and then using that access keys locally.

1. AWS Software Development Kit (SDK)

* Access AWS account through application code
* Allows you to access AWS Services from popular programming languages like Java, Python, C# and many more

**Compute Services**

1. EC2 (Elastic Cloud Compute)
2. Lambda
3. AWS Fargate
4. Amazon Lightsail
5. Amazon Outposts
6. AWS Batch

**EC2 (Elastic Cloud Compute)**

**Introduction**

* Elastic Cloud Compute
* Virtual server in the cloud
* Allows you to rent and manage virtual servers in the cloud
* Has elastic compute power: It can grow and shrink based on load in your application
* Are not serverless
* You are able to provision an EC2 instance at the click of a button
* You can also use pre-configured template called AMI (Amazon Machine Image) to launch your resources.
* You will receive 750 compute hours per month on Free Tier Plan
* You can deploy your applications directly to EC2 instances.

**EC2 in real world**

1. Deploy a database

Deploy a database to EC2 gives you full control over the database.

1. Deploy a web application

Deploy web application to multiple availability zones to make web application highly available.

**Methods to access EC2 instance**

1. AWS Management Console
2. Secure Shell (SSH)
3. EC2 Instance Connect (EIC)
4. AWS Systems Manager
5. AWS Management Console

* You are able to configure and manage instances via a web browser.

1. Secure Shell (SSH)

* Allows you to connect to your instance from local laptop using SSH Client and keys
* Puttygen converts .pem(privacy enhanced mail) file to ..ppk (putty private key)

1. EC2 Instance Connect

* Allows you to connect to instance without using SSH client and keys and directly using terminal in web browser
* For this, you must grant IAM users permission to push public key to the instance.

1. AWS System Manager

* Allows you to manage EC2 instances via a web browser or CLI

Note

The must common way to connect to Linux EC2 instance is via Secure Shell (SSH)

For this key pair is generated.

A key pair, consists of a private key and a public key, which proves your identity when connecting to an EC2 instance.

When user connects to SSH Client laptop, he uses private key. When he connects to EC2 instance, he uses public key.

**EC2 Pricing Model**

1. On demand
2. Spot
3. Reserved Instance
4. Dedicated Hosts
5. Savings Plans
6. On demand

* Fixed price in which you are billed based on instance type used
* You pay for what you use
* No contract
* Use on-demand instances when
  + You cannot make upfront payment or long-term commitment
  + Your applications have unpredictable workloads that cannot be interrupted
  + Your applications are under-development
  + Your applications will not run for more than a year

1. Spot

* Spot instance lets you take advantage of unused EC2 capacity
* Your request is available only if capacity is available
* Use Spot instance when
  + Your application is not concerned about start and stop time of execution
  + Your workload can be interrupted
* You can save upto 90% off on-demand prices
* You pay for spot that is in effect at the beginning of each hour

1. Reserved Instance

* Allows you to commit to a specific instance type for 1 or 3 years
* Use reserved instance when
  + Your application has steady state usage and you can commit for 1 or 3 years
  + You can make upfront payment
  + Your application requires capacity reservation
* You can save upto 75% off on-demand prices
* You are required to sign a contract of 1 or 3 years
* You can pay all-upfront, partial upfront or no upfront. All upfront for maximum term earns the highest discount

1. Dedicated Hosts

* Allows you to pay for physical server that is dedicated to running your instances. No other AWS customers is going to have applications running on this server. The server basically belongs to you
* Use dedicated hosts when
  + You want to bring your own server-bound software license from vendors like Microsoft and Oracle
  + You have regulatory or corporate compliance requirements around tenancy model
* You can save upto 70% off on-demand prices
* You can bring your existing per-core, per-socket and per-VM software licenses.

1. Savings Plan

* You do not commit for specific instance type instead you commit to compute usage (measured per hour) for 1 to 3 years
* Use savings plan when
  + You want to lower your bill across multiple compute services
  + You want flexibility to change compute services, instance types, OS or regions
* You can save upto 72% off on-demand prices
* Savings can be shared across various compute services like EC2, Fargate and lambda

Features

1. Elastic Load Balancing
2. EC2 Autoscaling
3. Elastic Load Balancing

* Automatically distributes your incoming traffic across multiple EC2 instances
* Types
  + Classic Load Balancer
  + Application Load Balancer
  + Gateway Load Balancer
  + Network Load Balancer

1. EC2 Autoscaling

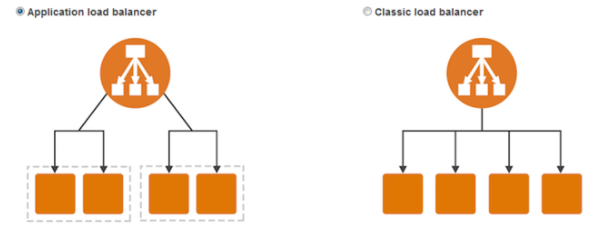
* Scale up or scale down based on workload in your application
* Reduces impact of system failure and improves availability of your applications
* You can also use autoscaling with Aurora and DynamoDB

Classic Load Balancer

* All incoming traffics will be equally distributed among number of servers we have

Application load Balancer

* Incoming traffics are distributed based on needs of customers
* It has some level of intelligence.
* It enables content-based routing and allows requests to be routed to different application/server behind single load balance.



Note

Horizontal Scaling vs Vertical Scaling

Horizontal Scaling (Scaling out)- adds or replaces instances

Vertical Scaling (Scaling up)- upgrades an existing instance

Tag

Allows you to track your instance by adding key/value pair

Eg: We can add a tag to specify “This instance is dedicated for Management purpose.”

AWS lambda

Introduction

* Serverless compute service that allows you to run your code without managing servers
* Allows developers to focus on core business logic for the application they are developing instead of worrying about managing servers
* Scales automatically
* Lambda functions can be written in popular programming languages like Java, Go, Node Js, C#, Python
* You can author your code using your favorite development environment locally and upload or you can write code via the console.
* Lambda can execute your code in response to some events
* Lambda functions have a 15-minute timeout. So, lambda is not for processes that run longer than 15 minutes.

Pricing Model

1. Compute time
2. Request
3. Always free
4. Compute time

* Time between start of execution and end of execution

1. Request

* A request is counted each time it starts execution
* Test invokes on the console is also considered as a request

1. Always free

* 1 million free requests per month (even after free usage tier expires)

Note

Duration: execution tine

Billed Duration: time you are actually billed for

AWS Fargate

Introduction

* Serverless compute service for containers
* Allows you to manager containers like Docker
* Works with Amazon Elastic Container Service (ECS)
* Scales automatically

AWS Lightsail

Introduction

* Quickly launch all the resources you need for small projects
* Deploy preconfigured applications like WebPress Website with the click of a button
* Includes Virtual Server, SSD based storage, static IP address, data transfer and DNS management
* Provides a low, predictable monthly fee, as low as $3.50
* Used to start small and scale as you grow

AWS Outspots

* AWS delivers and installs servers in your internal data center
* Allows you to run cloud services in your internal data center
* Supports hybrid deployment model

AWS Batch

* Allows you to process large workloads in smaller chunks (or batches)

Storage Services

1. Amazon S3

Amazon S3

Introduction

* Object Storage Service
* Objects are stored in buckets
* Unlimited storage- Millions of objects can be stored per bucket
* Objects can be private or public
* Objects can be uploaded via Management Console, CLI or from code using SDK
* Is a regional service, but bucket names must be globally unique
* You can set security at bucket level or individual object level using Access Control Lists (ACL), Bucket policies or Access Point policies
* You can enable versioning to create multiple versions of your file in order to protect against accidental deletion or roll back to previous version or file
* You can use S3 access logs to track access to your buckets and objects
* Data stored is replicated across multiple servers and so it gives high availability and durability in regional level.
* We can also set up Cross-Region Replication where your data is replicated across regions. But, S3 does not do that out of box. You have to set that up.

Two aspects of data accessibility

1. Durability
2. Availability
3. Durability

* Objects are never lost or compromised
* Amazon S3 Standard gives 99.999999999% (11 9’s) durability

1. Availability

* Access data quickly
* Amazon S3 Standard gives 99.99% availability

S3 Storage Class

There are different ways of storing your data

1. S3 Standard
2. S3 Intelligent Tiering
3. S3 Standard- Infrequent Access
4. S3 One Zone- Infrequent Access
5. S3 Glacier
6. S3 Glacier Deep Archive
7. S3 Outposts
8. S3 Standard

* General purpose storage
* Data is stored across multiple availability zones
* Recommended for
  + Data that are frequently accessed
* Durability of 99.999999999%
* Availability of 99.99%

1. S3 Intelligent Tiering

* Automatically moves your data to most cost-effective storage class
* Automatic cost savings
* Data is stored across multiple availability zones
* Recommended for
  + Data with unknown or changing access pattern like new app or data lake
* Durability of 99.999999999%
* Availability of 99.9%

1. S3 Standard- Infrequent Access

* Data is accessed less frequently but requires rapid access (millisecond access)
* Cheaper than S3 Standard
* Data is stored across multiple availability zones
* Recommended for
  + Data which is not frequently accessed but requires rapid access (millisecond access)
* Durability of 99.999999999%
* Availability of 99.9%

1. S3 One Zone- Infrequent Access

* Data is accessed less frequently but requires rapid access (millisecond access)
* Costs 20% less than S3 Standard Infrequent Access
* Data is stored in single availability zone
* Data can be lost
* Recommended for
  + Data which is not frequently accessed but requires rapid access (millisecond access)
  + Availability and Durability is not so essential
* Durability of 99.999999999%
* Availability of 99.5%

1. S3 Glacier

* Cold Storage for archival data
* Data Retrieval takes longer time
* 3 retrieval options: 1-5 min, 3-5 hours, 5-12 hours
* Data is stored in multiple availability zones
* Recommended for
  + Storage of archival data
* Durability of 99.999999999%

1. S3 Glacier Deep Archive

* Like S3 Glacier but longer access time
* 2 retrieval options: 12 hours, 48 hours
* Cheapest of all S3 options
* Data is stored in multiple availability zones
* Recommended for
  + Long-term archival data which is accessed once or twice a year
  + Retaining data for regulatory compliance requirement
* Durability of 99.999999999%

1. S3 Outposts

* Provides object storage on-premises
* Recommended for
  + Data that needs to be kept local (on-premises)

S3 in real world

1. Static Website

Deploy static websites to S3 and use CloudFront for global distribution

1. Data Archive

Archive data using Amazon Glacier as a storage option for Amazon S3

1. Analytics Services

Stores data in Amazon S3 for use with analytics services like Redshift and Athena

EBS (Elastic Block Storage)

Introduction

* Block storage service
* Like physical hard drive
* Accessible to a single EC2 instance at a time
* EBS instances can be either General Purpose SSD or Provisioned IOPs SSD
* Data persists even when instance is not running
* Recommended for
  + Quickly accessible data
  + Long-term data storage
  + Running a database on an instance

EC2 Instance Store

Introduction

* Local storage that is physically attached to host computer and cannot be removed
* Storage is temporary as data loss occurs when EC2 instance is stopped
* Faster with higher I/O speed
* Are ephemeral or temporary
* Recommended for
  + Temporary storage need
  + Data replicated across multiple instances

Amazon Elastic File System (EFS)

Introduction

* Hierarchical structure
* Can be mounted by multiple EC2 instances can be mounted in EC2 or on-premises servers
* Only supports the Linux file system
* More expensive than EBS
* Recommended for
  + Main directories for business-critical applications
  + Lift and shift existing enterprise apps

Note

In S3, API calls are made to access data

S3 can be accessed by other cloud services (unlike EBS and EFS)

Storage Gateway

Introduction

* Hybrid Storage Service
* Supports hybrid model

Content Delivery Services

Introduction

* Mechanism to deliver content quickly and efficiently based on your geographical location

1. Amazon CloudFront
2. Amazon Global Accelerator
3. Amazon S3 Transfer Acceleration

Amazon CloudFront

Introduction

* Allows global distribution of content
* Makes content available globally
* Delivers data and applications globally with low latency
* Speeds up delivery of static and dynamic web content
* Uses Edge Location to cache content
* If content is already in Edge Location, CloudFront delivers it immediately if not CloudFront retrieves the files from origin

CloudFront in real world

1. S3 static websites

CloudFront is often used with S3 to deploy content globally.

1. Prevent attacks

Prevent web attacks like DDoS.

1. IP address blocking

Geo restriction prevents users in certain countries from accessing content.

Amazon Global Accelerator

Introduction

* Networking Service that improves performance and availability of your application upto 60% using AWS Global Network Infrastructure
* When the Internet is congested, AWS Global Accelerator optimizes path to your application to keep packet loss, jitter and latency consistently low
* You are provided with two global static Ips that acts as fixed entry point to your application
* On the back end, add or remove your AWS application end points like Application Load Balancer, Network Load Balancer, EC2 instances and Elastic Ips without making user facing changes
* Automatically reroutes traffic to nearest healthy available endpoint to mitigate endpoint failure
* Performs health check of end point

Similarities between Amazon CloudFront and Amazon Global Accelerator

1. Both use Edge Location and Amazon Global Network Infrastructure
2. Both are integrated with Amazon Shield to protect DDoS attack

Amazon S3 Transfer Acceleration

Introduction

* Fast, easy and secure transfer of files over long distance between your client and S3 bucket
* Uses Edge Location
* Customers can upload files to central bucket

Networking Services

Networking connects computers together and allows for sharing of data and applications around the globe, in a secure manner using virtual routers, firewalls and network management services.

1. Amazon Virtual Private Cloud (VPC)
2. Route 53
3. Direct Connect
4. AWS VPN

Amazon Virtual Private Cloud

Introduction

* Allows you to create virtual private network in the AWS cloud where you launch your resources and deploy applications
* It is a way to isolate and protect resources
* You are able to setup a private virtual network and you specify various things like IP address ranges, security groups, subnet and you configure route tables
* You are able to launch your EC2 instances within your VPC

Components

1. Subnet
2. NACL
3. Router and Route Table
4. Internet Gateway
5. VPC Peering
6. Subnet

Allows you to split network inside the VPC.

1. NACL

Network Access Control List

Ensures proper traffic is allowed into subnet

Can block traffic to particular instance

1. Router and Route table

Defines where network traffic is routed

1. Internet Gateway

Allows communication between Internet and VPC

Allows traffic to internet from VPC

1. VPC Peering

Allows you to connect two VPC’s together

Note

To specify a range of IP addresses for VPC, we use CIDR

Route 53

Introduction

* Domain Name System (DNS) service that routes user to application by translating domain name to IP addresses
* Allows domain name registration
* Performs health checks on AWS resources
* Supports hybrid architecture

AWS Direct Connect

Introduction

* Supports hybrid model
* Sensitive Data remains in private cloud. Application that reads these data remains in public cloud and they communicate with each other through Direct Connect
* Data travels over private cloud

Direct Connect in real world

1. Transfer internal data directly to AWS, bypassing your Internet Service Provider

AWS VPN

Introduction

* Site-to-site VPN
* By default, instances that are launched into Amazon VPC cannot communicate with internal/ on-premises network
* We can enable access to internal network from your VPC by creating an AWS site-to-site VPN
* Supports hybrid environment
* Cheaper than Direct Connect
* Similar to Direct Connect but data travels over public internet

Databases

Different types of Databases

1. RDS
2. Aurora
3. DynamoDB
4. DocumentDB
5. ElastiCache
6. Neptune

RDS

* Supports popular database engines
  + MySQL
  + Microsoft SQL Server
  + Oracle
  + Amazon Aurora
  + PostgreSQL
  + MariaDB
* AWS manages database with automatic software patching, automatic backups, operating system management and more
* Launch read replicas across regions in order to provide enhanced performance and durability

Aurora

* Relational database compatible with MySQL and PostgreSQL
* Supports MySQL and PostgreSQL database engines
* 5x times faster than MySQL and 3x times faster than PostgreSQL
* Managed by RDS

DynamoDB

* Fully managed NoSQL database
* Serverless

DocumentDB

* Fully managed document database that supports MongoDB
* Serverless

ElastiCache

* Fully managed in memory datastore
* Compatible with Redis and Memcached engine
* Data can be lost

Neptune

* Fully managed graph database that supports highly connected datasets
* Supports highly connected datasets like social media networks
* Serverless

Usecases

1. Migrate an on-premises Oracle to Cloud- RDS
2. Migrate an on-premises PostgreSQL database to the cloud- RDS/ Aurora
3. Alleviate database load for data that is accessed often- ElastiCache
4. Process large sets of user profiles and social interactions- Neptune
5. NoSQL database fast enough to handle millions of requests per second- DynamoDB
6. Operate MongoDB workloads at scale- DocumentDB

Migration and Transfer Services

1. Database Migration Service (DMS)
2. Server Migration Service (SMS)
3. Snow Family
4. Datasync

Database Migration Service (DMS)

Introduction

* Migrate database to or within AWS
* Supports homogenous (oracle-oracle) and heterogenous (oracle-sql) migrations
* Virtually no downtime i.e. your source database remains operational

DMS in real world

1. Oracle to Aurora MySQL

Migrate on-premises oracle database to Aurora MySQL

1. Oracle to Oracle

Migrate an on-premises Oracle database to Oracle in EC2

1. RDS Oracle to Aurora MySQL

Migrate an RDS Oracle database to Aurora MySQL

Server Migration Service (SMS)

Introduction

* Allows you to migrate on-premises servers to AWS
* Server is saved as a new AMI (Amazon Machine Image) and you use that AMI to launch that server as EC2 instance

AWS Snow Family

Snow Family includes

1. Snowcone
2. Snowball and Snowball Edge
3. Snowmobile
4. Snowcone

* Allows you to transfer large amount of on-premises data to cloud using physical device
* Smallest member of data transport devices
* 8 terabytes of storage

1. Snowball and Snowball Edge

* Petabyte scale of data transport
* Cheaper than internet transfer
* It is not just data migration device but is also an edge computing device i.e., it allows local processing
* Snowball edge supports EC2 and lambda

1. Snowmobile

* Multi-petabyte or exabyte scale of data transport
* Data is transported using 45-foot long shipping container
* People use this when they are completely shipping their on-premises database and moving to AWS
* With these trucks, there is GPS tracking, alarm monitoring, 24X7 video surveillance, and as escort security vehicle during transport

Datasync

Introduction

* Allows for online data transfer from on-premises to AWS Storage Services like S3 or EFS
* Data transfer speeds are upto 10 times faster than open source tools
* Can be used to replicate data across cross region or cross account

Analytics Services

1. Redshift
2. Athena
3. Glue
4. Kinesis
5. EMR (Elastic Map Reduce)
6. Data Pipeline
7. Redshift

* Scalable data warehouse solution

1. Athena

* Analyze S3 data using SQL

1. Glue

* Prepares data for analytics
* ETL (Extract, Transform, Load)

1. Kinesis

* Analyze real time streaming data
* Analyze log data

1. EMR (Elastic Map Reduce)

* Process large amount of data
* Process big data
* Analyze data using Hadoop
* Works with big data frameworks like Apache Spark

1. Data Pipeline

* Helps you move data between compute or storage services running either on AWS or on-premises
* Moves data at specific interval
* Moves data based on conditions
* Sends notifications on success or failure
* Eg: Move data from S3 to Redshift

Analytics in Real World

1. Search data in S3- Athena
2. Log analytics- Kinesis

Machine Learning Services

1. Recognition
2. Comprehend
3. Polly
4. Sagemaker
5. Translate
6. Lex

Recognition

Introduction

* Allows you to automate image and video analysis

Real World Use cases

* Identify toppings on pizza to make sure that they are being made consistently
* Object detection
* Face recognition
* Identify text in image

Comprehend

Introduction

* Natural Language Processing service
* Analyze text

Real World Use case

* Review social media posts

Analyze comments in social media to find how many people are happy

Polly

Introduction

* Turns text into speech
* Can create a custom voice

SageMaker

Introduction

* Helps you build, train and deploy machine learning models quickly
* Prepare data for models
* Train and deploy models
* Provides Deep Learning AMIs

Real World Use case

* Recommendation Engine

Translate

Introduction

* Language translation

Lex

Introduction

* Build conversational interfaces like Chatbots

Real World Use case

* Integrate voice into a device

Developer Tools

1. Cloud9
2. Code Commit
3. Code Build
4. Code Deploy
5. Code Pipeline
6. X-Ray
7. Cloud9

Introduction

* IDE (Integrated Development Environment)
* Allows you to write and debug code from web browser

1. Code Commit

Introduction

* Similar to GitHub
* Create repositories to store code

Use case

* Manage versions of source code files for your applications

1. Code Build

Introduction

* Allows you to build and test your application source code
* Produces build artifacts that are ready to be deployed

Use case

* Run tests before deploying a new version of an application to production

1. Code Deploy

Introduction

* Manages deployment of code
* Maintains application uptime

Use case

* Maintains application uptime when rolling out a new version

1. Code Pipeline

Introduction

* Automate software release process
* Quickly deliver new features and updates
* Integrates with Code Commit to retrieve source code
* Integrates with Code Build to build and run application source code
* Integrates with Code Deploy to deploy your changes

Use case

* Automation of building, testing and deployment of your application

1. X-Ray

Introduction

* Analyze and debug production workloads
* Map application components
* View request flow end-to-end

Use case

* Trace calls to RDS Database

Deployment and Infrastructure Management Service

1. CloudFormation
2. Elastic Beanstalk
3. OpsWorks

Infrastructure as a code

It allows you to write code in JSON or YAML format to provision AWS resources

1. CloudFormation

Introduction

* Allows you to provision AWS resources using Infrastructure as a Code
* Works with most AWS services
* Create template for resources you want to provision

Use case

Automate Infrastructure-provisioning process for EC2 servers

1. Elastic Beanstalk

Introduction

* Serverless compute service that allows you to deploy your web application or web services on Cloud
* It is not used to deploy your application on-premises
* Monitors application health via a health dashboard

Use case

* Deploy a scalable Java-based web application to AWS

1. OpsWorks

Introduction

* Allows you to use Chef or Puppet to automate configuration of your servers and deploy code
* Can deploy applications on-premises

Use case

* Automate software configurations and infrastructure management for your application

Messaging and Integration Services

1. SQS (Simple Queue Service)
2. SNS (Simple Notification Service)
3. SES (Simple Email Service)
4. Simple Queue Service (SQS)

Introduction

* Message queuing service that allows you to build loosely coupled systems
* Allows component-to-component communication using messages
* Multiple components can add messages to the queue
* Messages are processed in an asynchronous manner

Use case

* Build money transfer app that performs well under heavy workload

1. Simple Notification Service (SNS)

Introduction

* Allows you to send plain text email or text messages from your application

Use case

* Send as email when CPU utilization of EC2 instance goes above 80%

1. Simple Email Service (SMS)

Introduction

* Allows you to send richly formatted email from your application
* Unlike SNS, it is used to send HTML emails
* Ideal choice for marketing campaigns or professional emails

Use case

* Send marketing emails

Exploring Auditing, Monitoring and Logging Services

1. CloudWatch
2. CloudTrail
3. CloudWatch

Introduction

* Collection of services like CloudWatch Alarms, CloudWatch Logs, CloudWatch Metrics, CloudWatch Events
* CloudWatch Alarm is used to set alarm
* CloudWatch Logs is used to monitor application logs
* CloudWatch Metrics is used to visualize metrics over time
* CloudWatch Events is used to trigger an event based on condition

Use case

* Provide real time monitoring on EC2 instance
* Receive a notification when root user activity is detected in your account

1. CloudTrail

Introduction

* Track user activities within your account that has been performed using console, CLI or SDK
* Track which user made changes
* Detect unusual activity in your account
* If you want to track activities of above 90 days, you have to create custom trail

Use case

* Track time when a particular event occurred in your account

Security and Compliance

Shared Responsibility Model

In the public cloud, there is shared responsibility between you and AWS

AWS responsibility- Security of the cloud

Your responsibility- Security in the cloud

Security of the cloud

1. AWS Global Infrastructure

AWS is responsible for global infrastructure elements like Region, Availability Zone and Edge Locations

1. Building Security

AWS controls access to its data centers where your data resides

1. Networking Components

AWS maintains networking components: Generators, Uninterruptable Power Supply, Computer room air conditioning units, fire suppression systems, etc.

1. Software

AWS is responsible for any managed services like RDS, S3, lambda.

Security in the cloud

1. Application Data

You are responsible for managing your application data which includes encryption options.

1. Security Configuration

You are responsible for securing your account like rotating credentials, restricting Internet access to VPC.

1. Patching

You are responsible for guest operating system which includes updates and security patches.

1. Identity and Access Management
2. Network Traffic

Firewall configurations

1. Installed Software

Who is responsible for what?

1. Firewall configuration- You
2. Data center security for physical building- AWS
3. Encryption of EBS volumes- You
4. Language versions of lambda- AWS
5. Taking DB backups in RDS- You
6. Updating firmware on underlying EC2 hosts- AWS
7. Ensuring data is encrypted at rest- You
8. Ensuring network infrastructure- AWS
9. Patching guest OS for EC2- You
10. Physically destroying storage media at the end of file- AWS

Which security responsibilities are shared?

1. Patch Management
2. Configuration Management
3. Awareness and Training
4. Patch Management

AWS- Patching infrastructure

You- Patching guest OS and applications

1. Configuration Management

AWS- Configuring infrastructure devices

You- Configuring applications and databases

1. Awareness and Training

AWS- AWS employees

You- Your employees

How do I report abuse of AWS resources?

1. Contact [abuse@amazonaws.com](mailto:abuse@amazonaws.com)
2. Contact AWS Trust and Safety Team using Report Amazon AWS Abuse Form

Well-Architectured Framework

5 pillars

1. Operational Excellence
2. Reliability
3. Security
4. Performance Efficiency
5. Cost Optimization
6. Operational Excellence
   * Build application that effectively supports production workload
7. Reliability
   * Build application that works consistently and recovers quickly
8. Security
   * Build application that protects system and data
9. Performance Efficiency
   * Build application that meets business and system requirements while removing bottlenecks
   * Use serverless architectures first
   * Use multi-regions deployments
10. Cost Optimization
    * Build application at the least cost to the user

Five pillars in real world

1. Operational Excellence- You can use AWS Code Commit
2. Security- CloudTrail
3. Reliability- Multi AZs deployment
4. Performance Efficiency- Use AWS lambda
5. Cost Optimization- S3 Intelligent-Tiering