

Learn

# AWS Basics

# aWS



# What is AWS?



**Amazon Web Services (AWS)** is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully-featured services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.

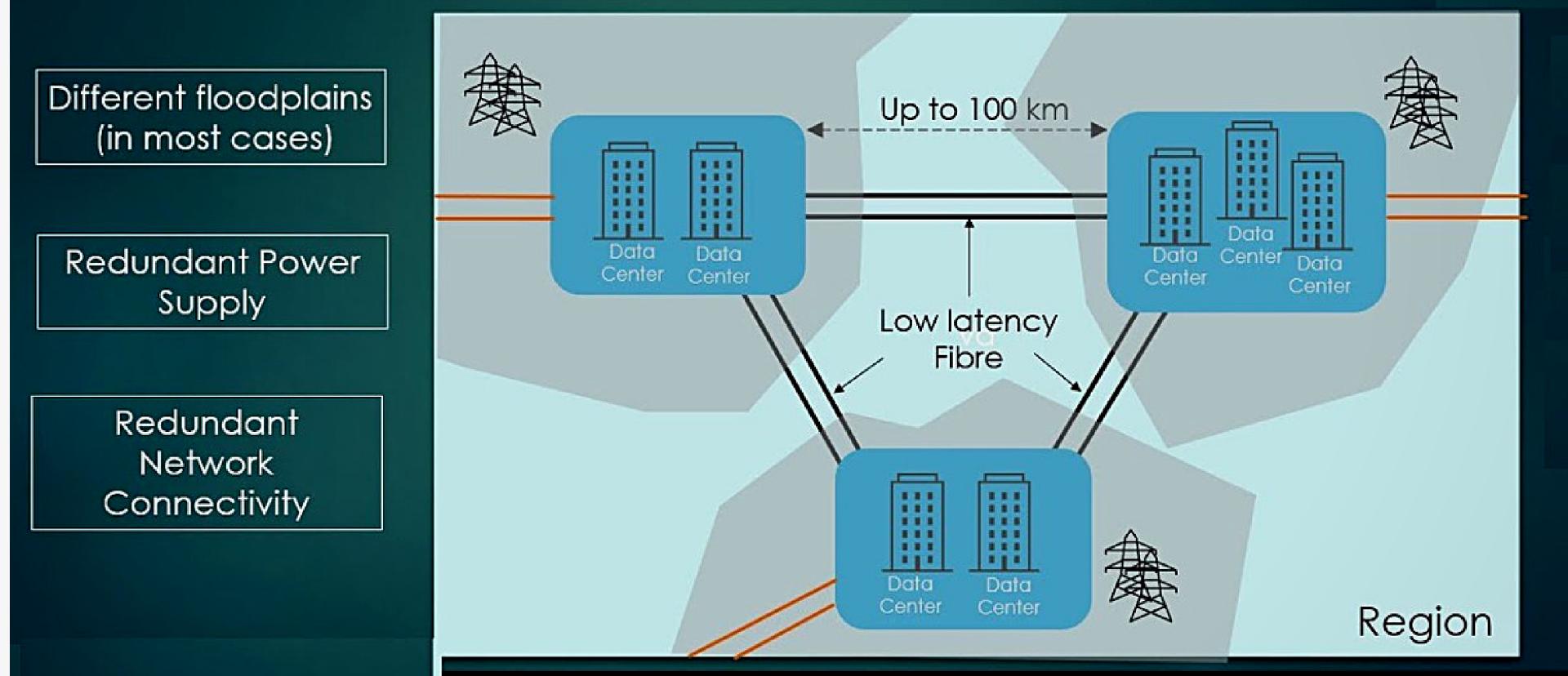
# AWS Global Infrastructure

The Most Secure, Extensive, and Reliable Global Cloud Infrastructure, for all your applications



- Regions
- Coming Soon

1 Region = Multiple AZs (Min 3)  
1 AZ = Cluster of Data centres



**Tip**

**Edge location > Availability Zones > Regions**

# What is Region & Availability Zones?

AWS has the concept of a **Region**, which is a physical location around the world where they cluster data centers.

They call each group of logical data centers an **Availability Zone**.

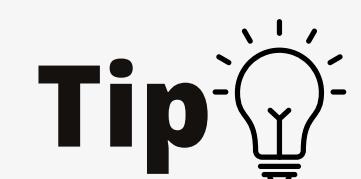
Each AWS Region consists of multiple, isolated, and physically separate AZs within a geographic area

To know more [click here](#)

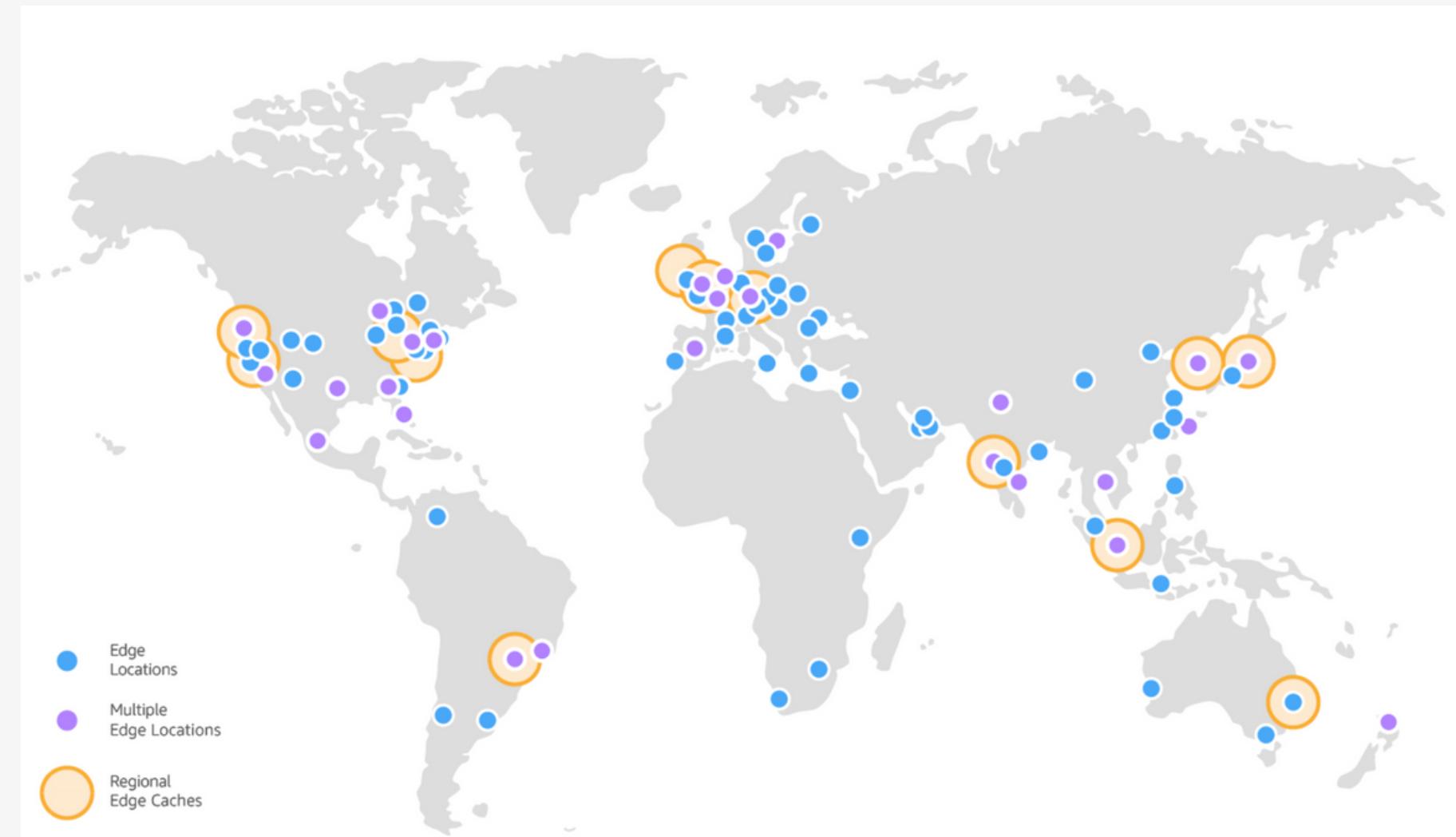
# What is Edge Locations?

An edge location is where end-users access services located at AWS, the cloud computing division of US-headquartered Amazon. They are located in most of the major cities around the world and are specifically used by CloudFront (CDN) to distribute content to end-users to reduce latency.

To learn more [Click Here](#)



**Edge location > Availability Zones > Regions**



# AWS Core Services

**List of the services you MUST definitely know**

**AWS IAM**

---

**AWS EC2**

---

**AWS EBS**

---

**AWS VPC**

---

**AWS RDS**

---

**AWS S3**

---

**AWS ECS**

---

**AWS Lambda**

---

**AWS CloudFront**

---

**AWS CloudWatch**

---

**AWS SES**

---

**AWS SNS**

---

**AWS SQS**

---

**AWS DevOps tools (CodePipeline + Code Commit + CodeBuild + CodeDeploy)**

---

# AWS Identity and Access Management (IAM)

Securely manage access to AWS services and resources

AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

## Points to focus

**Basic Components** Learn about basic components of IAM: Users, Groups, Permissions, MFA, Roles, Policies

**Don't use Root and never share root access** It is recommended to not use root for everything, instead, create IAM users, and utilize them and never share root access to anyone, must enable Multi-Factor Authentication for root (and for all other IAM users)

**Use roles as much as possible** Instead of creating IAM users and credentials, it's recommended, as well as a better + secure way is to use IAM role if IAM role can be used. e.g. IAM instance role, IAM service role

**Use Temporary Credentials** Utilize Identity Federation when applicable, e.g Enterprise identity federation, Web-based Federation, Cross-account Access Role etc

**Utilize IAM Credentials** Utilize Access Keys and password for AWS CLI & API Access, use IAM user for CodeCommit access

# AWS EC2

**Secure and resizable compute capacity to support virtually any workload Get started with Amazon EC2**

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment.

## Points to focus

**Instance Types Selection:** Based on need one should select the instance type and size such as General Purpose – (T2, M4, M3), Compute Optimized – (C5, C4, C3), Memory Optimized – (X1, R4, R3), Accelerated Computing (P3, P2, G3, F1), Storage optimized-(I3), Dense-storage Instances – (D2)

**Storage Selection:** Select and use storage as per need, allocate as per application need, and don't overestimate, if need huge storage to store some data then store in AWS S3 instead of EBS Volume, and if need multiple instances need to use a common storage location then use AWS EFS

**Network Selection & Configuration:** Configure secure VPC by configuring Public/Private subnets, assign Elastic IPs properly, and use only if needed else release the same, make sure configure security groups properly and allow port that needed and to the targeted users, allow ssh to particular IP addresses, only necessary persons should have .pem access or even better use AWS SSM

**Cost Optimization** Use **Spot Instances, Reserve Instances, Saving Plans** and create a strategy to reduce the cost, use **auto-scaling** to scale in when resources not needed and scale up and when needed, use auto turn off instances if instances not in use, use **serverless** tech (**Lambda**) if the application is intermittent or event-based

# Amazon Elastic Block Store (EBS)

Easy to use, high performance block storage at any scale

Amazon Elastic Block Store (EBS) is an easy-to-use, high-performance, block-storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction-intensive workloads at any scale. A broad range of workloads, such as relational and non-relational databases, enterprise applications, containerized applications, big data analytics engines, file systems, and media workflows are widely deployed on Amazon EBS.

## Points to focus

- Select** Select EBS Type based on the application need, there are multiple types of EBS volume available such as
- EBS Type:** **SSD Based:** gp2, gp3, io1, io2, io2 Block Express. "1" category is the highest performance and the cost is higher and "g" category is the general-purpose and cost is relatively less
- HDD Based:** st1, st2 . Throughput Optimized HDD (st1) for frequently accessed, throughput intensive workloads and the lowest cost Cold HDD (sc1) for less frequently accessed data.
- Termination** By default termination protection is turned off, so if instances get terminated the attached root EBS volume also
- Protection:** gets terminated, if you want to keep the storage volume for future use then to avoid this have to enable the termination protection option.
- Resize:** Volumes can be easily resized i.e. increase/decrease as per need
- Encrypt:** It fully supports encryption, using AWS KMS you can easily encrypt volume, and you can enable default while creating volume
- Snapshots** To create back-ups, use snapshots. Snapshots get stored in S3 (internally by AWS), you can copy snapshots to any region

# Amazon Virtual Private Cloud (VPC)

Build on a logically isolated virtual network in the AWS cloud

Amazon Virtual Private Cloud (Amazon VPC) is a service that lets you launch AWS resources in a logically isolated virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 for most resources in your virtual private cloud, helping to ensure secure and easy access to resources and applications.

## Points to focus

**Conceptualize & Understand** very well these concepts: CIDR, Subnet, Internet Gateway, NAT Gateway & NAT instances, Virtual Private Gateway, Customer Gateway, Router, VPC Endpoints, Egress-only Internet Gateway, Security Groups, **key components** Networks ACL, AWS direct connect, VPC Flow Logs, VPC endpoints, VPC peering,

**Pricing** Always remember to check the pricing estimate before selecting VPC/Network-specific service, for example NAT gateway charges, VPC endpoint charges, VPN connection per hour charges

**Understand ENI** Understand what is ENI and how can you utilize it, it basically gives you a unique MAC address which can be used for licensing purpose

**Tenancy Options:** Tenancy options are either shared or dedicated, inside VPC. while creating/assigning resources, you have the option to select tenancy, either shared which is virtually assigned to a common pool of resources, and dedicated in which you get dedicated hardware for a pool of resource allocation and it's not shared with anyone, it often required for regulatory purposes, but cost is higher than shared tendency.

# Amazon Relational Database Service (RDS)

Set up, operate, and scale a relational database in the cloud with just a few clicks.

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

## Points to focus

**Supportability** It supports: **Aurora**, MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server

**Use case based instance type selection** Select DB cluster/instance size based on use case, such as for development only use lowest size resources and for Production instance use DB instances with higher size but remember the cost is greater factor here, so based on budget select Production DB instance Size

**High Availability & Fault Tolerance** Fault tolerance and high availability can be achieved by Multi-AZ deployment strategy and multiple read/write nodes (selected DB engine support it)

**Auto Scaling** Utilize RDS storage autoscaling. Amazon RDS for MariaDB, Amazon RDS for MySQL, Amazon RDS for PostgreSQL, Amazon RDS for SQL Server, and Amazon RDS for Oracle support RDS Storage Auto Scaling.

**Pricing & Monitoring** Choose DB engine based on the use case and Instances size based on use case + budget, use creation screen and cost calculator to estimate the cost, enable the monitoring to observe free space, CPU, Network, Memory usage, set alerts as needed, use reserved instances and savings plan to save cost

**Use Aurora Serverless** use Aurora serverless, if your application is a Relational DB, usage is intermittent and unpredictable or of running serverless applications, it saves a lot of cost!

**Backup & Restore** One of the great features is Backup and Restore, use automated backups and set retention policy as needed or as per business requirement

# Amazon Simple Storage Service (Amazon S3)

Set up, operate, and scale a relational database in the cloud with just a few clicks.

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.

## Points to focus

- Understand** S3 STANDARD for general-purpose storage of frequently accessed data
- storage Classes** S3 STANDARD\_IA for long-lived, but less frequently accessed data. It stores the object data redundantly across multiple geographically separated AZs.  
S3 ONEZONE\_IA stores the object data in only one AZ. Less expensive than STANDARD\_IA, but data is not resilient to the physical loss of the AZ  
Amazon S3 Intelligent Tiering is a storage class designed for customers who want to optimize storage costs automatically when data access patterns change, without performance impact or operational overhead  
GLACIER is used for long-term archive
- Understand and utilize** key-value data, metadata architecture, bucket naming as a domain naming, data consistency model (get, put, for delete request), understand buck policies, transfer accelerations using edge locations, life cycle policies, cross-origin resource sharing (CORS), events notification, and listening/processing to such events using AWS Lambda, s3 static hosting, encryption methods in S3, cross-region replications, use S3 programmatically using AWS SDK or via AWS Cli
- these**

# Amazon Elastic Container Service (Amazon ECS)

**Highly secure, reliable, and scalable way to run containers**

Amazon Elastic Container Service (Amazon ECS) is a fully managed container orchestration service that helps you easily deploy, manage, and scale containerized applications. It deeply integrates with the rest of the AWS platform to provide a secure and easy-to-use solution for running container workloads in the cloud and now on your infrastructure with Amazon ECS Anywhere.

## Points to focus

**Understand Key Concepts** Understand well core components such as Containerization, Docker, Clusters, Task definition, Tasks, Task LifeCycle, Service, Task/service Autoscaling, Cluster Autoscaling, Capacity Providers, Ec2 based vs Fargate vs ECS anywhere

**Deployment Methods** There are two deployment methods, rolling update and blue/green deployment, choose the deployment method as per use case

**Load Balancing** Learn Service Autoscaling, learn how to use Application Load Balancer with Services and develop micro services

**Dynamic Port Mapping** One of the excellent features of using AWS ECS is Dynamic port mapping, This allows to run multiple container app in single host on different ports automatically and ECS will automatically do the traffic management via load balancer, learn how to achieve it using Task definition + Services, make sure to configure security group for the same properly

**AWS FARGATE** If would like to run the containerized application without managing Ec2 or if want to run containerized application serverless way, Use AWS Fargate, but remember the cost part, use AWS cost calculator before starting task/service using AWS Fargate

**ECS On-premise** Use ECS Anywhere to run and manage the containerized application on-premise or anywhere

**Auto Schedule** You can auto-schedule running of ECS tasks using AWS console , CLI or AWS SDK

**Automate** Automation can be achieved using AWS CLI, AWS SDK, CDK, Copilot etc

# AWS Lambda

**Run code without thinking about servers or clusters. Only pay for what you use.**

AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers, creating workload-aware cluster scaling logic, maintaining event integrations, or managing runtimes. With Lambda, you can run code for virtually any type of application or backend service - all with zero administration.

Just upload your code as a ZIP file or container image, and Lambda automatically and precisely allocates compute execution power and runs your code based on the incoming request or event, for any scale of traffic.

You can set up your code to automatically trigger from over 200 AWS services and SaaS applications or call it directly from any web or mobile app. You can write Lambda functions in your favorite language (Node.js, Python, Go, Java, and more) and use both serverless and container tools, such as AWS SAM or Docker CLI, to build, test, and deploy your functions.

## Points to focus

**Native Language** Node.js, Java, C#, Go, Python, Ruby, PowerShell (You can provide your custom runtime as well)

**Support**

**Cost parameter** Cost determined on the amount of memory used + CPU power

**Key components** Functions, Runtimes, Layers, Event Source, downstream resources, log streams, env variables,  
**to learn**

**Lambda Edge** Let's you run Lambda functions to customize content that CloudFront delivers, executing the functions in AWS locations closer to the viewer. The functions run in response to CloudFront events, without provisioning or managing servers.

**Serverless** Use serverless framework with your preferred choice of language , it makes life a lot easier while developing Serverless  
**Framework** applications

# Amazon CloudFront

**Fast, highly secure and programmable content delivery network (CDN)**

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.

CloudFront offers the most advanced security capabilities, including field level encryption and HTTPS support, seamlessly integrated with AWS Shield, AWS Web Application Firewall and Amazon Route 53 to protect against multiple types of attacks including network and application layer DDoS attacks. These services co-reside at edge networking locations – globally scaled and connected via the AWS network backbone – providing a more secure, performant, and available experience for your users.

CloudFront works seamlessly with any AWS origin, such as Amazon S3, Amazon EC2, Elastic Load Balancing, or with any custom HTTP origin. You can customize your content delivery through CloudFront using the secure and programmable edge computing features CloudFront Functions and AWS Lambda@Edge.

## Points to focus

**Use case** It's a Content Delivery Network (CDN) service

It delivers your content through a worldwide network of data centers called edge locations. When a user requests content that you're serving with CloudFront, the user is routed to the edge location that provides the lowest latency, so that content is delivered with the best possible performance. If the content is already in the edge location with the lowest latency, CloudFront delivers it immediately. If the content is not in that edge location, CloudFront retrieves it from an origin that you've defined

**Protocols and Methods Support** supports the WebSocket protocol as well as the HTTP protocol, and HTTP methods: GET, HEAD, POST, PUT, DELETE, OPTIONS, PATCH.

**Understand the concepts** Origin, object caching, distribution, ACM integration for HTTPS, cache behavior, Price class, origin access identity, field-level encryption, CloudFront integration with AWS Shield, AWS WAF, and Route 53

# Amazon CloudWatch

## Observability of your AWS resources and applications on AWS and on-premises

Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers. You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly.

## Points to focus

**Key components** Namespaces, Metrics, Dimensions, Statistics, Percentiles, Alarms, CloudWatch Dashboard, Events, Logs, log insights, Cloudwatch Agent,

**Pricing** charged for the number of metrics you have per month, charged per 1000 metrics requested using CloudWatch API calls, charged per dashboard per month, charged per alarm metric, charged per GB of collected, archived and analyzed log data, , charged per million custom events and per million cross-account events , Logs Insights is priced per query and charges based on the amount of ingested log data scanned by the query

**Cloud Watch** Deliver near real-time stream of system events that describe changes in AWS resources. Concepts needed to be noted

**Events** here: Events – indicates a change in your AWS environment, Targets – processes events, Rules – matches incoming events and routes them to targets for processing

**Works Great With** Cloud watch works great with other services, especially easy integration between other services can be achieved , AWS

**other Services** Lambda integration makes it so powerful.

# Amazon Simple Email Service

**High-scale inbound and outbound cloud email service**

Amazon Simple Email Service (SES) is a cost-effective, flexible, and scalable email service that enables developers to send mail from within any application. You can configure Amazon SES quickly to support several email use cases, including transactional, marketing, or mass email communications. Amazon SES's flexible IP deployment and email authentication options help drive higher deliverability and protect sender reputation, while sending analytics measure the impact of each email. With Amazon SES, you can send email securely, globally, and at scale.

## Points to focus

**Key parts need to be covered** Domain verification using DNS records, SES can be used for both sending and receiving emails, emails can be sent via SES Console, SMTP protocol & SES API using AWS SDK or AWS CLI

**Key Concepts needed** SES SMTP, Email deliverability, Reputation, Bounce, Global suppression list, Deliveries, Opens, Clicks, Configuration sets, Dedicated IP Pools

**Cost** 62,000 emails free, the cost is after that for every 1,000 emails and when receiving 1,000 emails you receive are free and then cost per email.

**Note** By default your application will be in sandbox mode, have to apply for limit increase/production access to out of sandbox and send email to anyone else

# Amazon Simple Notification Service

Fully managed pub/sub messaging, SMS, email, and mobile push notifications

Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication.

The A2A pub/sub functionality provides topics for high-throughput, push-based, many-to-many messaging between distributed systems, microservices, and event-driven serverless applications. Using Amazon SNS topics, your publisher systems can fanout messages to a large number of subscriber systems including Amazon SQS queues, AWS Lambda functions and HTTPS endpoints, for parallel processing, and Amazon Kinesis Data Firehose. The A2P functionality enables you to send messages to users at scale via SMS, mobile push, and email.

## Points to focus

**Key parts need to be covered** Event-driven computing model, Message filtering, SNS mobile notifications, Push email and text messaging using SNS

**Integration** SNS can be easily integrated with other services, especially AWS Lambda

**Use cases** Use case might be, sending OTP text messages or promotion messages, android/ios push notifications, system to system or service to service messaging via AWS Lambda, sending emails.

**Monitoring** Message delivered can be tracked via AWS Cloudwatch logs for both failure and successful deliveries and easily searchable using AWS Cloudwatch log insights

**Cost** Cost is based on number of messages, notifications, SMS sent

**Note** To send important SMS such as OTP you have to enable transaction mode with cost limit and have to apply via support

# Amazon Simple Queue Service

Fully managed message queues for microservices, distributed systems, and serverless applications

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications.

SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work. Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available. Get started with SQS in minutes using the AWS console, Command Line Interface or SDK of your choice, and three simple commands.

SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

## Points to focus

**Key parts need to be covered** standard and FIFO queues for polling, message encryption, message attributes, Message timers, Long polling vs short polling, visibility timeout, dead-letter queues, Delay queues, cost allocation tags, batches,

**Integration & Use cases** SNS can be easily integrated with other services, especially AWS Ec2, Lambda, and other service to active even greater scalability and loose coupling infra goals

**Monitoring** SQS queues using CloudWatch, SQS API Calls Using AWS CloudTrail, Automate notifications from AWS Services to SQS using CloudWatch Events

**Cost** charged per 1 million SQS requests. Price depends on the type and data transfer cost for egress traffic after 1 GB/month

# AWS DevOps Tools

## **CodeCommit**

AWS CodeCommit is a secure, highly scalable, managed source control service that hosts private Git repositories. It makes it easy for teams to securely collaborate on code with contributions encrypted in transit and at rest. CodeCommit eliminates the need for you to manage your own source control system or worry about scaling its infrastructure. You can use CodeCommit to store anything from code to binaries. It supports the standard functionality of Git, so it works seamlessly with your existing Git-based tools.

## **CodeBuild**

AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers. CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue. You can get started quickly by using prepackaged build environments, or you can create custom build environments that use your own build tools. With CodeBuild, you are charged by the minute for the compute resources you use.

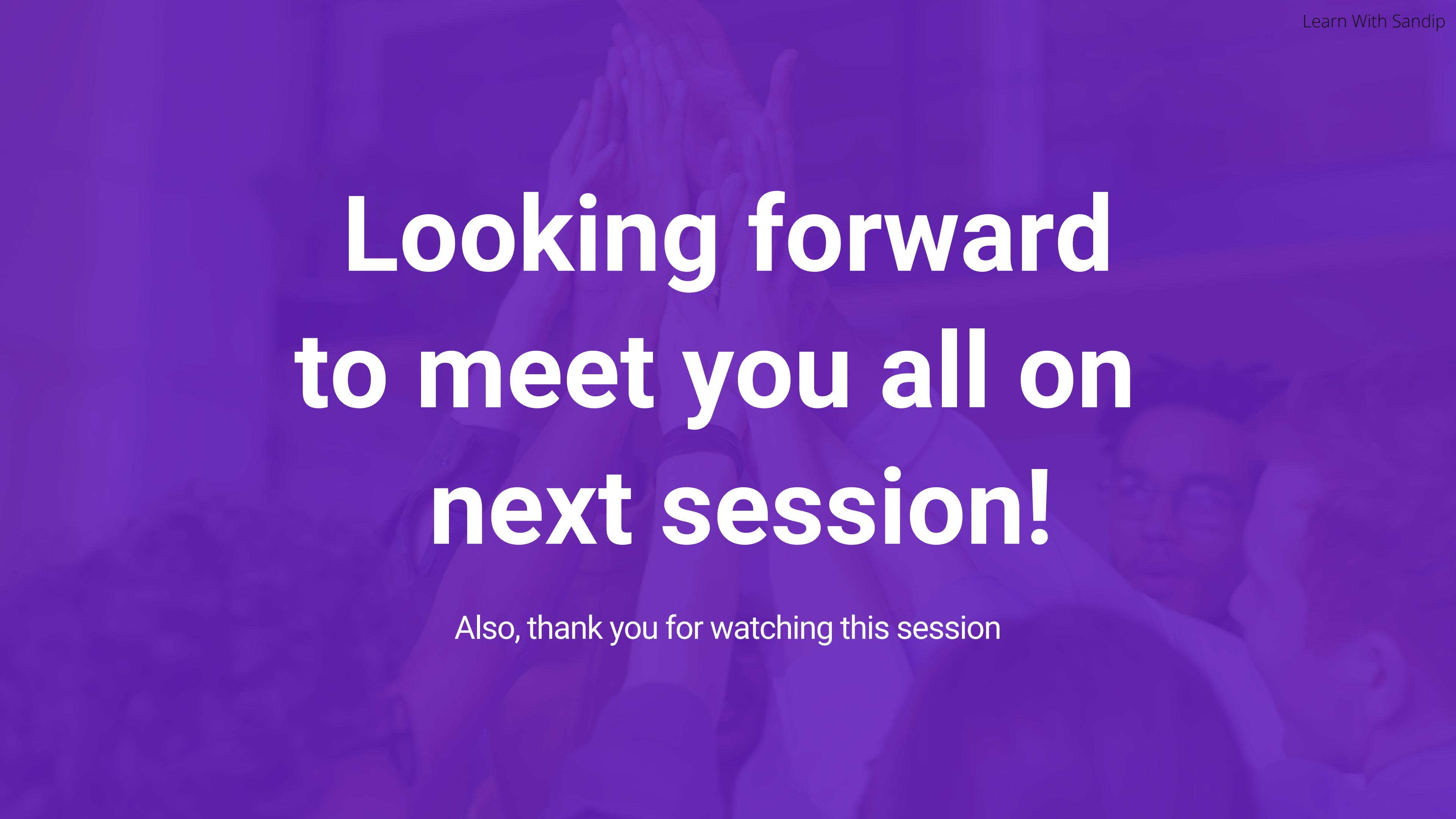
## **CodeDeploy**

AWS CodeDeploy is a fully managed deployment service that automates software deployments to a variety of compute services such as Amazon EC2, AWS Fargate, AWS Lambda, and your on-premises servers. AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications. You can use AWS CodeDeploy to automate software deployments, eliminating the need for error-prone manual operations. The service scales to match your deployment needs.

## **CodeCommit**

AWS CodePipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates. CodePipeline automates the build, test, and deploy phases of your release process every time there is a code change, based on the release model you define. This enables you to rapidly and reliably deliver features and updates. You can easily integrate AWS CodePipeline with third-party services such as GitHub or with your own custom plugin. With AWS CodePipeline, you only pay for what you use. There are no upfront fees or long-term commitments.

[\*\*Click Here to Learn More\*\*](#)



A faint background image shows a person's hands clasped together in a clapping motion, suggesting a positive or celebratory atmosphere.

**Looking forward  
to meet you all on  
next session!**

Also, thank you for watching this session