DAY 1 TRAINING OUTLINE

I. Introduction to AWS

- Definition of cloud computing and its benefits
- How AWS provides cloud computing services
- History of AWS
- Impact of AWS on the industry
- AWS Marketplace
- Benefits of AWS

II. AWS Services and Categories

- Compute Services
- Storage Services
- Database Services
- Networking Services
- Analytics Services
- Machine Learning Services
- Security Services
- Application Integration
- Customer Engagement

III. AWS Security and Compliance

- AWS Security Best Practices
- IAM (Identity and Access Management)
- Encryption
- Compliance

IV. AWS Deployment and Management

- CloudFormation
- Elastic Beanstalk
- OpsWorks
- CodeDeploy

V. AWS Monitoring and Automation

INTRODUCTION TO AWS

AWS stands for Amazon Web Services, which is a cloud computing platform provided by Amazon.com. AWS provides a wide range of cloud computing services to individuals, organizations, and governments, including compute, storage, databases, networking, analytics, machine learning, security, and more. Cloud computing refers to the delivery of computing resources and services over the internet, on a pay-as-you-go pricing model, without the need for on-premises infrastructure. AWS provides a scalable, flexible, and cost-effective way for businesses to run applications and services in the cloud, without the need to manage physical hardware or software.

AWS is the market leader in the cloud computing industry, with millions of active customers in over 190 countries, including startups, enterprises, and public sector organizations. Some of the popular companies and products that use AWS include Netflix, Airbnb, Samsung, and the US Department of Defense.

DEFINITION OF CLOUD COMPUTING AND IT BENEFITS

Cloud computing is the delivery of computing services, including servers, storage, databases, networking, software, analytics, and intelligence, over the internet on a pay-as-you-go pricing model. In other words, cloud computing enables users to access computing resources on demand, without the need for physical hardware or infrastructure.

The benefits of cloud computing include:

- Scalability: Cloud computing allows businesses to scale up or down their computing resources according to their needs, without the need for physical infrastructure. This means that businesses can handle sudden spikes in demand or seasonal fluctuations without incurring additional costs.
- Flexibility: Cloud computing provides businesses with the flexibility to choose the computing services they need, without being restricted by physical infrastructure or hardware. This allows businesses to experiment with new technologies or features without incurring upfront costs.
- Cost Savings: Cloud computing eliminates the need for businesses to invest in physical infrastructure, such as servers and data centers, reducing capital expenses. Additionally, businesses only pay for the resources they use, reducing operational expenses.
- Reliability: Cloud computing providers, such as AWS, offer high levels of uptime and availability, ensuring that businesses' applications and services are always accessible to their customers.
- Security: Cloud computing providers, such as AWS, invest heavily in security to protect their customers' data and applications from cyber threats, providing businesses with a level of security that is often difficult to achieve with physical infrastructure.

How AWS provides cloud computing services

AWS provides cloud computing services by offering a wide range of computing resources and services on a pay-as-you-go pricing model. These resources and services include compute, storage, databases, networking, analytics, machine learning, security, and more. AWS provides these resources and services from data centers located in different regions around the world. AWS allows users to access these resources and services through a web-based console, command line interface, or software development kits (SDKs). Users can create and manage virtual machines, storage systems, databases, and other resources through these interfaces.

AWS also provides several tools and services that help users to manage their applications and services on the cloud, such as Amazon CloudWatch, which provides monitoring and logging services, AWS Identity and Access Management (IAM), which provides access control and security management services, and AWS Elastic Beanstalk, which simplifies the deployment of web applications.

In addition to providing computing resources and services, AWS also offers a range of other services, such as training and certification programs, support services, and partner programs, to help customers get the most out of their cloud computing experience.

HISTORY OF AWS

AWS was officially launched by Amazon.com in 2006, but its origins date back to the late 1990s when Amazon.com started building its own infrastructure to support its growing e-commerce platform. Amazon.com realized that it had developed a lot of expertise in building and operating large-scale data centers, which it could leverage to offer cloud computing services to other businesses.

In 2002, Amazon.com started offering web services to developers, providing access to its infrastructure through APIs. These services included storage, computation, and messaging services, and were aimed at helping developers build and run their applications more easily and cost-effectively.

In 2006, Amazon.com launched AWS as a commercial cloud computing platform, offering a wide range of services to businesses and developers. AWS quickly gained popularity, as it provided a scalable, flexible, and cost-effective way for businesses to run their applications in the cloud.

Evolution of AWS services and features

Over the years, AWS has continued to expand its services and capabilities, introducing new services such as Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3), Amazon Relational Database Service (RDS), and Amazon Elastic Block Store (EBS). Today, AWS offers over 200 services to customers in over 190 countries, and is the market leader in the cloud computing industry.

Since its launch in 2006, AWS has evolved significantly, introducing new services and features to meet the changing needs of its customers. Here are some of the major milestones in the evolution of AWS services and features:

- Compute Services: AWS started with its Elastic Compute Cloud (EC2) service, which provided scalable compute capacity in the cloud. Over the years, AWS has added other compute services such as Elastic Container Service (ECS), Elastic Kubernetes Service (EKS), Lambda, and Fargate, to cater to different use cases and workloads.
- Storage Services: AWS launched Simple Storage Service (S3) in 2006, which provided scalable object storage in the cloud. Since then, AWS has introduced other storage services such as Elastic Block Store (EBS), Glacier, and Elastic File System (EFS), to cater to different storage needs and use cases.
- Database Services: AWS launched Amazon Relational Database Service (RDS) in 2009, which provided managed database services in the cloud. Since then, AWS has added other database services such as DynamoDB, Aurora, Redshift, and DocumentDB, to cater to different types of databases and workloads.
- Networking Services: AWS launched Amazon Virtual Private Cloud (VPC) in 2009, which provided isolated virtual networks in the cloud. Since then, AWS has added other networking services such as Elastic Load Balancer (ELB), AWS Direct Connect, and Route 53, to cater to different networking needs and use cases.
- Analytics Services: AWS launched Amazon EMR (Elastic MapReduce) in 2009, which provided managed Hadoop clusters in the cloud. Since then, AWS has added other analytics services such as Kinesis, Athena, QuickSight, and Glue, to cater to different analytics needs and use cases.
- Machine Learning Services: AWS launched Amazon Machine Learning (AML) in 2015, which provided a managed service for building and deploying machine learning models in the cloud. Since then, AWS has added other machine learning services such as SageMaker, Comprehend, Rekognition, and Polly, to cater to different machine learning needs and use cases.
- Security Services: AWS has always prioritized security and compliance, and has introduced several security services and features such as Identity and Access Management (IAM), Key Management Service (KMS), Certificate Manager, and GuardDuty, to provide a secure cloud computing environment for its customers.

AWS continues to innovate and introduce new services and features, to help its customers stay ahead of the curve and leverage the latest technologies in the cloud.

Impact of AWS in the industry

AWS has had a significant impact on the industry since its launch in 2006. Here are some of the ways in which AWS has influenced the industry:

- Cloud Computing Adoption: AWS has played a key role in the adoption of cloud computing by businesses of all sizes. AWS has made it easier and more cost-effective for businesses to access computing resources and services, without the need for upfront investments in hardware and infrastructure.

- Innovation and Agility: AWS has enabled businesses to innovate and move faster, by providing scalable and flexible computing resources that can be provisioned and de-provisioned quickly. This has helped businesses to experiment, test and launch new products and services more quickly, and respond faster to changing market demands.
- Cost Savings: AWS has helped businesses to save costs by reducing the need for upfront investments in hardware and infrastructure. By providing pay-as-you-go pricing and the ability to scale resources up or down as needed, AWS has enabled businesses to optimize their costs and only pay for what they use.
- Scalability and Resilience: AWS has enabled businesses to scale their applications and services quickly and easily, to accommodate sudden spikes in demand. AWS provides automatic scaling and load balancing capabilities, which help businesses to ensure that their applications and services are always available and performant.
- Global Reach: AWS has enabled businesses to reach customers and markets around the world, by providing a global infrastructure with data centers located in different regions. This has helped businesses to expand their reach and serve customers in new geographies.
- Technology Advancements: AWS has pushed the boundaries of cloud computing and introduced several new technologies and services, such as serverless computing, machine learning, and Internet of Things (IoT). These technologies have enabled businesses to leverage the latest advancements in computing and build innovative products and services.

In summary, AWS has had a significant impact on the industry, by enabling businesses to innovate, be more agile, and save costs. AWS has also pushed the boundaries of cloud computing and introduced several new technologies and services, which have enabled businesses to stay ahead of the curve and leverage the latest advancements in computing.

AWS Marketplace

AWS Marketplace is an online store that provides users with a selection of third-party software solutions that can be deployed on the AWS cloud platform. The AWS Marketplace makes it easy for customers to find, compare, and purchase software solutions that are designed to work with the AWS cloud platform.

Some of the benefits of using the AWS Marketplace include:

- Simplified Procurement: Customers can easily purchase software solutions directly from the AWS Marketplace, eliminating the need to work with multiple vendors.
- Flexible Pricing: Customers can choose from a variety of pricing models, including hourly, monthly, and annual pricing options.
- Integrated Billing: All charges for software solutions purchased through the AWS Marketplace are consolidated into a single AWS bill, simplifying the billing process.
- Trusted Vendors: All software solutions available on the AWS Marketplace have been vetted by AWS to ensure they meet security and compliance standards.

To use the AWS Marketplace, customers simply browse the available software solutions, select the ones they want to use, and deploy them on the AWS cloud platform. The AWS Marketplace provides a wide range of software solutions, including security tools, data management solutions, and application development tools, among others.

Benefits of AWS

There are several benefits of using AWS, including:

- Scalability: AWS provides elastic resources, which means that you can easily scale up or down depending on your business needs. This allows you to pay only for what you need, without having to worry about managing your own hardware.
- Cost-effective: AWS has a pay-as-you-go pricing model, which means that you only pay for what you use. This eliminates the need for upfront capital expenditure on hardware and software.
- Security: AWS provides a secure infrastructure, which includes network isolation, firewalls, and encryption. Additionally, AWS compliance programs help customers meet industry-specific regulatory requirements.
- Flexibility: AWS provides a wide range of services and tools, which allows customers to choose the services that best suit their needs. This provides flexibility in terms of application development and deployment.
- Global reach: AWS has a global network of regions and availability zones, which provides customers with low latency and high performance access to their applications.
- Reliability: AWS has a highly reliable infrastructure, which includes data replication across availability zones, automatic failover, and backup and restore capabilities.
- Innovation: AWS provides customers with access to new technologies and services, which enables them to innovate and respond to changing business needs.

AWS SERVICES

AWS provides a wide range of cloud computing services across different categories, including compute, storage, database, networking, analytics, machine learning, security, and more. Here are some of the key AWS services in each category:

Compute Services:

1. **Amazon EC2 (Elastic Compute Cloud):**
   Amazon EC2 (Elastic Compute Cloud) is a web service that provides scalable computing capacity in the cloud. With Amazon EC2, users can launch virtual machines (known as instances) on-demand, and pay only for the compute capacity that they actually use.
   Some of the key features of Amazon EC2 include:
   - Flexible Compute Options: Amazon EC2 provides a variety of instance types, each optimized for different use cases and workloads. These instances range from small, general-purpose instances to large, compute-optimized instances with specialized hardware.
   - Scalability and Elasticity: With Amazon EC2, users can easily scale their compute capacity up or down, based on changing demands. This means that users can add or remove instances as needed, without any upfront commitments.
   - Security and Compliance: Amazon EC2 provides a number of security features to help users secure their instances and data, including network security, encryption, and compliance certifications.
   - Integration with Other AWS Services: Amazon EC2 integrates with other AWS services, such as Amazon VPC (Virtual Private Cloud) and AWS Elastic Load Balancing, to provide a complete solution for running applications and services in the cloud.
   - Cost-effective: Amazon EC2 provides a variety of pricing options, including On-Demand, Reserved Instances, and Spot Instances. This enables users to optimize their costs based on their usage patterns and workloads.

   Overall, Amazon EC2 is a powerful and flexible service that enables users to launch and manage virtual machines in the cloud, and scale their compute capacity as needed.

   How to setup Amazon EC2 (Elastic Compute Cloud)

   To set up Amazon EC2, you will need an AWS account and some basic knowledge of AWS services. Here are the steps to follow:
   - Create an AWS account: If you don't have an AWS account already, go to the AWS homepage and sign up for an account.
   - Create an Amazon EC2 instance: Once you have an AWS account, go to the EC2 Dashboard and click on the "Launch Instance" button. Follow the wizard to select your instance type, Amazon Machine Image (AMI), storage options, security groups, and other configurations.
   - Connect to your instance: Once your instance is running, you can connect to it using SSH (for Linux instances) or RDP (for Windows instances). You can also use the AWS Management Console or the AWS CLI to manage your instance.

- Configure your instance: Once you are connected to your instance, you can configure it as needed, by installing software, configuring services, and setting up security features.
- Use your instance: Once your instance is set up and configured, you can use it to run your applications, services, and workloads in the cloud.
- Monitor and manage your instance: AWS provides a variety of tools and services for monitoring and managing your instances, including CloudWatch, AWS Systems Manager, and AWS CLI.

These are the basic steps to set up an Amazon EC2 instance. However, there are many more advanced features and options available, such as auto-scaling, load balancing, and advanced security options, that you can use to optimize your instance and application performance.

2. **Amazon ECS (Elastic Container Service):**

Amazon ECS (Elastic Container Service) is a fully-managed container orchestration service that enables users to run and scale containerized applications in the cloud. Amazon ECS supports Docker containers and can be integrated with other AWS services to provide a complete solution for deploying and managing containerized applications.

Some of the key features of Amazon ECS include:

- Container Management: Amazon ECS provides tools for managing containers, including scheduling, monitoring, and scaling. Users can easily deploy and manage containerized applications using the Amazon ECS console or the AWS CLI.
- Integration with Other AWS Services: Amazon ECS integrates with other AWS services, such as Amazon Elastic Load Balancing, AWS Identity and Access Management (IAM), and Amazon CloudWatch, to provide a complete solution for deploying and managing containerized applications.
- Auto-scaling: Amazon ECS enables users to automatically scale their containerized applications up or down, based on changing demands. This means that users can add or remove container instances as needed, without any upfront commitments.
- Cost-effective: Amazon ECS provides a variety of pricing options, including On-Demand, Reserved Instances, and Spot Instances. This enables users to optimize their costs based on their usage patterns and workloads.
- Flexible Networking: Amazon ECS enables users to customize their container networking using Amazon VPC (Virtual Private Cloud) and other networking features.

Overall, Amazon ECS is a powerful and flexible service that enables users to run and manage containerized applications in the cloud. With its integration with other AWS services and support for Docker containers, Amazon ECS provides a complete solution for deploying and managing modern, containerized applications.

How to set up Amazon ECS

Setting up Amazon ECS involves the following steps:

- Create an AWS account: If you don't have an AWS account already, go to the AWS homepage and sign up for an account.

- Create an Amazon VPC: Amazon ECS requires an Amazon VPC (Virtual Private Cloud) to run. If you don't have a VPC already, go to the Amazon VPC console and create one.
- Create a task definition: A task definition is a blueprint that describes how to run a containerized application. Go to the Amazon ECS console and create a task definition that specifies the container image, CPU and memory requirements, and networking options.
- Create a cluster: A cluster is a group of container instances that run together. Go to the Amazon ECS console and create a cluster that includes one or more container instances.
- Create a service: A service is a set of tasks that run together and are managed by Amazon ECS. Go to the Amazon ECS console and create a service that runs on the cluster you created in the previous step.
- Deploy the application: Once you have created the service, Amazon ECS will automatically deploy your application to the cluster. You can monitor the deployment using the Amazon ECS console or the AWS CLI.
- Scale the application: Amazon ECS enables you to scale your application up or down, based on changing demands. You can use the Amazon ECS console or the AWS CLI to adjust the number of tasks running in the service.

These are the basic steps to set up Amazon ECS. However, there are many more advanced features and options available, such as load balancing, auto-scaling, and advanced security options, that you can use to optimize your application performance and security.

Advance Features of Amazon ECS

Here are some of the advanced features of Amazon ECS:

- Load balancing: Amazon ECS can be integrated with Amazon Elastic Load Balancing (ELB) to distribute traffic across multiple containers running on different container instances. This helps to improve application availability and scalability.
  Steps to set up Load balancing for Amazon ECS:
  i. Create a task definition: Create a task definition that specifies the container image, CPU and memory requirements, and networking options for your application.
  ii. Create a service: Create a service that runs the task definition you created in the previous step. When you create the service, you can specify the number of tasks to run and the load balancer to use.
  iii. Create a load balancer: Create an Elastic Load Balancer (ELB) using the Amazon EC2 console or the AWS CLI. You can choose between Application Load Balancer (ALB) or Network Load Balancer (NLB) based on your application requirements.
  iv. Register the container instances: Register the container instances that will run your tasks with the load balancer. You can do this using the Amazon EC2 console or the AWS CLI.

v. Configure the target group: Configure the target group for the load balancer, which specifies the container instances that will receive traffic from the load balancer.
vi. Associate the service with the load balancer: Associate the service you created in step 2 with the load balancer you created in step 3. When you associate the service with the load balancer, Amazon ECS automatically registers the tasks in the service with the target group.
vii. Test the load balancer: Test the load balancer by accessing the DNS name or IP address of the load balancer. The load balancer will distribute traffic across the registered container instances based on the load balancing algorithm you specified.

These are the basic steps to set up load balancing for Amazon ECS. However, there are many more advanced features and options available, such as SSL termination, health checks, and cross-zone load balancing, that you can use to optimize your application performance and availability.

- Auto-scaling: Amazon ECS supports auto-scaling, which allows you to automatically adjust the number of container instances based on demand. You can use Amazon ECS to automatically scale up or down the number of container instances based on metrics such as CPU utilization or memory usage.

Here are the steps to set up auto-scaling for Amazon ECS:

i. Create an Amazon ECS cluster: Create an Amazon ECS cluster that will host your container instances and services.
ii. Create an Amazon ECS service: Create an Amazon ECS service that runs the tasks for your application. When creating the service, you can specify the minimum and maximum number of tasks to run.
iii. Configure Amazon CloudWatch alarms: Configure Amazon CloudWatch alarms to monitor the metrics that you want to use to trigger scaling events. For example, you can monitor CPU utilization, memory usage, or network traffic.
iv. Create an auto-scaling policy: Create an auto-scaling policy that defines the scaling behavior for your service. For example, you can create a policy that adds or removes tasks based on the number of CloudWatch alarms that are triggered.
v. Create an auto-scaling group: Create an auto-scaling group that launches new instances based on the scaling policy you created in the previous step. You can use Amazon EC2 Auto Scaling to automatically launch and terminate instances based on the demand for your application.
vi. Test the auto-scaling: Test the auto-scaling by generating traffic to your application and monitoring the number of tasks that are running in your service. As the demand for your application increases or decreases, Amazon

ECS will automatically scale up or down the number of tasks running in your service.

These are the basic steps to set up auto-scaling for Amazon ECS. However, there are many more advanced features and options available, such as step scaling, predictive scaling, and scheduled scaling, that you can use to fine-tune your auto-scaling behavior and optimize your application performance and cost.

- Service discovery and Task placement strategies: Amazon ECS supports service discovery, which enables your applications to discover and connect to other services running on the same cluster. You can use Amazon ECS to register services and their associated DNS names, and then use DNS queries to discover other services. Amazon ECS supports task placement strategies, which allow you to control how tasks are placed on container instances within a cluster. You can use placement strategies to optimize resource utilization, improve application performance, or ensure compliance with regulatory requirements.
  Here are the steps to set up service discovery and task placement strategies for Amazon ECS:
  i. Create a VPC: Create a Virtual Private Cloud (VPC) for your Amazon ECS cluster. This VPC will be used to isolate your container instances and services from other resources in your AWS account.
  ii. Create a namespace: Create a namespace in AWS Cloud Map that will be used to register and discover the services running in your Amazon ECS cluster. A namespace defines a domain name and a service name that you can use to identify your services.
  iii. Register services: Register the services running in your Amazon ECS cluster with AWS Cloud Map. When you register a service, you specify the service name, the DNS record type, and any custom attributes that you want to associate with the service.
  iv. Configure task placement strategies: Configure task placement strategies to control how Amazon ECS places tasks on your container instances. You can use task placement constraints to specify placement rules based on attributes such as instance type, availability zone, or custom labels.
  v. Enable service discovery: Enable service discovery for your Amazon ECS services by associating them with the AWS Cloud Map namespace you created in step 2. When you enable service discovery, Amazon ECS automatically registers your services with AWS Cloud Map and updates the DNS records to reflect the current state of your services.
  vi. Discover services: Discover the services running in your Amazon ECS cluster by querying the DNS records in AWS Cloud Map. You can use the service name and any custom attributes to filter the results and identify the specific instances that are running the service you are interested in.

vii. These are the basic steps to set up service discovery and task placement strategies for Amazon ECS. However, there are many more advanced features and options available, such as weighted routing policies, namespace sharing, and service discovery integrations, that you can use to optimize your application architecture and scalability.

- IAM integration: Amazon ECS integrates with AWS Identity and Access Management (IAM), which allows you to control access to your container instances and services using IAM roles and policies. To set up IAM integration for Amazon ECS, you need to create an IAM role that grants the necessary permissions to Amazon ECS to manage your resources on your behalf. You can use the AWS Management Console, AWS CLI, or AWS CloudFormation to create the IAM role.

  Here are the steps to set up IAM integration:

  i. Go to the AWS Management Console and navigate to the IAM dashboard.
  ii. Click on "Roles" and then "Create Role".
  iii. Select "AWS service" as the trusted entity and choose "ECS" as the service that will use this role.
  iv. Select the necessary permissions for your role, such as "AmazonEC2ContainerServiceFullAccess" and "AmazonEC2ReadOnlyAccess".
  v. Name your role and click "Create Role".

- Advanced networking options: Amazon ECS supports advanced networking options such as Network Load Balancer, which allows you to load balance traffic at the transport layer (Layer 4) for high-performance, low-latency applications. You can also use Amazon VPC networking features to configure advanced network topologies and security. Amazon ECS offers several advanced networking options that you can use to optimize the performance, security, and cost of your applications. These options include VPC networking, Elastic Network Interfaces (ENIs), and Service Load Balancers. To set up these options, you need to configure your Amazon ECS cluster and services using the AWS Management Console, AWS CLI, or AWS CloudFormation.

  Here are the steps to set up Advanced networking options:

  i. Create a VPC and subnets using the AWS Management Console or AWS CLI.
  ii. Configure your Amazon ECS cluster to use the VPC and subnets you created. You can do this using the AWS Management Console or AWS CLI.

iii. Configure ENIs and Service Load Balancers as needed for your application. You can do this using the AWS Management Console or AWS CLI.

- Fargate: Amazon ECS also supports a serverless option called Fargate, which allows you to run containers without managing the underlying infrastructure. With Fargate, you simply define your containerized application and the resources it requires, and Amazon ECS takes care of the rest. : Fargate is a serverless computing engine for Amazon ECS that allows you to run containers without managing the underlying infrastructure. To set up Fargate, you need to create an Amazon ECS task definition that specifies the container images, resources, and networking settings for your application. You can then create an Amazon ECS service that uses Fargate as the launch type to deploy your application.

Here's an overview of how to set up and use AWS Fargate:

i. Create your container images: Before using Fargate, you need to create container images using Docker or other containerization tools. You can then store your container images in a container registry, such as Amazon ECR or Docker Hub.
ii. Define your task definitions: Once you have container images, you need to define task definitions, which describe the resources and configuration required for your containerized applications. You can define various parameters, such as container images, memory, CPU, and networking settings.
iii. Create your services: After defining your task definitions, you can create services, which allow you to run and scale your containerized applications on Fargate. You can specify the number of tasks to run, the desired state, and the scaling policies.
iv. Monitor and troubleshoot: AWS provides various monitoring and troubleshooting tools, such as Amazon CloudWatch, AWS X-Ray, and AWS Fargate Insights, which help you monitor the performance and health of your containerized applications. You can also view task logs and metrics to troubleshoot issues.
v. Manage permissions and security: To use Fargate, you need to configure permissions and security settings, such as IAM roles and policies, VPCs, and security groups. You can also encrypt your data at rest and in transit using various encryption options.

These are the basic steps to set up IAM integration, advanced networking options, and Fargate for Amazon ECS. However, there are many more advanced features and options available that you can use to fine-tune your Amazon ECS environment and optimize your application performance and cost.

- Amazon EKS (Elastic Kubernetes Service)

  Amazon EKS (Elastic Kubernetes Service) is a fully managed Kubernetes service that makes it easy to deploy, manage, and scale containerized applications using Kubernetes on AWS.

  Here are the steps to set up Amazon EKS:

  i. Create an Amazon EKS cluster: To create an Amazon EKS cluster, you need to use the AWS Management Console, AWS CLI, or AWS CloudFormation. You can choose to create a new VPC or use an existing one. You will also need to specify the number of worker nodes and their instance types.
  ii. Set up worker nodes: Once you have created an Amazon EKS cluster, you need to set up worker nodes to run your containerized applications. You can do this by launching EC2 instances or using AWS Fargate, which is a serverless computing engine for containers.
  iii. Configure kubectl: kubectl is a command-line tool that you use to deploy and manage your applications on Kubernetes. To configure kubectl to communicate with your Amazon EKS cluster, you need to install the AWS CLI and configure it with your AWS credentials. You can then use the "aws eks update-kubeconfig" command to create a new kubeconfig file that points to your Amazon EKS cluster.
  iv. Deploy applications: Once you have set up your worker nodes and configured kubectl, you can deploy your containerized applications to your Amazon EKS cluster using Kubernetes manifests. Kubernetes manifests are YAML files that describe the desired state of your applications, including container images, resources, and networking settings.
  v. Scale your applications: Amazon EKS makes it easy to scale your applications horizontally or vertically to meet changing demand. You can use Kubernetes scaling features, such as Horizontal Pod Autoscaler (HPA) and Cluster Autoscaler, to automatically adjust the number of replicas or worker nodes based on resource utilization or other metrics.
  vi. Monitor and troubleshoot: To monitor and troubleshoot your Amazon EKS cluster and applications, you can use various AWS services and tools, such as Amazon CloudWatch, AWS X-Ray, and AWS Systems Manager. These services provide visibility into your application performance, logs, and metrics, and help you identify and fix issues quickly.

  These are the basic steps to set up and use Amazon EKS. However, there are many more advanced features and options available that you can use to optimize your Kubernetes environment and applications.

- AWS Lambda

  AWS Lambda is a serverless computing service provided by AWS, which allows you to run code without provisioning or managing servers. With Lambda, you can create functions that respond to events, such as changes to data in an Amazon S3 bucket, updates to a database, or user requests from an API.

  Here's an overview of how to set up and use AWS Lambda:

i. Create a Lambda function: To create a Lambda function, you can use the AWS Management Console, AWS CLI, or AWS CloudFormation. You will need to choose the programming language and runtime for your function, such as Node.js, Python, Java, or C#. You can also configure the memory and execution time limits for your function.
ii. Write your code: Once you have created a Lambda function, you need to write your code. Your code should be designed to handle the event triggers that you want to respond to, and it should be optimized for performance and scalability.
iii. Test your function: After writing your code, you can test your function using the AWS Management Console or the AWS CLI. You can simulate event triggers and check the response from your function.
iv. Deploy your function: Once you have tested your function, you can deploy it to AWS Lambda. You can also configure your function to trigger from various event sources, such as Amazon S3, Amazon Kinesis, or an API Gateway.
v. Monitor and troubleshoot: To monitor and troubleshoot your Lambda function, you can use various AWS services and tools, such as Amazon CloudWatch, AWS X-Ray, and AWS Lambda Insights. These services provide visibility into your function performance, logs, and metrics, and help you identify and fix issues quickly.
vi. Manage versions and permissions: AWS Lambda allows you to manage versions and aliases of your functions, which enables you to deploy new code versions without affecting the production environment. You can also manage permissions for your functions, such as IAM roles and policies.

These are the basic steps to set up and use AWS Lambda. However, there are many more advanced features and options available that you can use to optimize your serverless applications, such as configuring VPCs, using Lambda Layers, and integrating with other AWS services.

- AWS Batch

  AWS Batch is a managed service provided by AWS that allows you to run batch computing workloads on the AWS Cloud. With AWS Batch, you can run batch jobs of any scale, such as processing large amounts of data, running high-performance computing (HPC) applications, or executing containerized workloads.

  Here's an overview of how to set up and use AWS Batch:

  i. Define your compute environment: To use AWS Batch, you need to define a compute environment, which is a collection of Amazon EC2

instances that you can use to run your batch jobs. You can choose the type and size of the instances based on your workload requirements.

ii. Create your job definitions: Once you have defined your compute environment, you need to create job definitions, which describe the parameters and resources required for your batch jobs. You can define various parameters, such as container image, command, memory, and CPU requirements.

iii. Submit your jobs: After creating your job definitions, you can submit your batch jobs to AWS Batch. You can submit jobs using the AWS Management Console, AWS CLI, or SDKs. You can also configure job dependencies, job priorities, and job queues.

iv. Monitor and troubleshoot: AWS Batch provides various monitoring and troubleshooting tools, such as Amazon CloudWatch, AWS X-Ray, and AWS Batch Insights, which help you monitor the performance and health of your batch jobs. You can also view job logs and metrics to troubleshoot issues.

v. Manage permissions and security: To use AWS Batch, you need to configure permissions and security settings, such as IAM roles and policies, VPCs, and security groups. You can also encrypt your data at rest and in transit using various encryption options.

These are the basic steps to set up and use AWS Batch. However, there are many more advanced features and options available that you can use to optimize your batch workloads, such as using spot instances, managing job arrays, and integrating with other AWS services.

Storage Services:

Here is an overview of the AWS storage services and how to set them up:

- Amazon S3 (Simple Storage Service):

Amazon S3 is an object storage service that allows you to store and retrieve any amount of data from anywhere on the web.

Here's how to set up Amazon S3:

i. Create an S3 bucket: You can create a new S3 bucket using the AWS Management Console, AWS CLI, or AWS SDKs. You can specify various bucket properties, such as bucket name, region, access control settings, and encryption options.

ii. Upload and manage objects: Once you have created an S3 bucket, you can upload objects, such as files, images, and videos, and manage them using

the S3 console, CLI, or SDKs. You can also set object permissions, versioning, and lifecycle policies.

- Amazon EBS (Elastic Block Store):

Amazon EBS is a block-level storage service that provides persistent block storage volumes for EC2 instances.

Here's how to set up Amazon EBS:

i. Create an EBS volume: You can create a new EBS volume using the AWS Management Console, AWS CLI, or AWS SDKs. You can specify various volume properties, such as volume type, size, and encryption options.
ii. Attach and mount volumes: Once you have created an EBS volume, you can attach it to an EC2 instance and mount it as a block device. You can also detach and delete volumes when you no longer need them.

- Amazon Glacier:

Amazon Glacier is a low-cost storage service that provides secure and durable archival storage for data backups and long-term retention.

Here's how to set up Amazon Glacier:

i. Create a Glacier vault: You can create a new Glacier vault using the AWS Management Console, AWS CLI, or AWS SDKs. You can specify various vault properties, such as vault name, access control settings, and notification options.
ii. Upload and manage archives: Once you have created a Glacier vault, you can upload archives, such as files, backups, and data archives, and manage them using the Glacier console, CLI, or SDKs. You can also set archive retrieval options and policies.

- Amazon EFS (Elastic File System):

Amazon EFS is a fully managed file storage service that provides scalable and elastic NFS file storage for EC2 instances.

Here's how to set up Amazon EFS:

i. Create an EFS file system: You can create a new EFS file system using the AWS Management Console, AWS CLI, or AWS SDKs. You can specify various file system properties, such as file system name, performance mode, and encryption options.

ii. Mount file systems: Once you have created an EFS file system, you can mount it to EC2 instances using the NFS protocol. You can also set file system permissions, backup and restore options, and monitoring and logging settings.

- AWS Storage Gateway:

AWS Storage Gateway is a hybrid storage service that enables you to seamlessly connect on-premises applications with cloud storage services. Here's how to set up AWS Storage Gateway:

i. Install and activate a gateway: You can install a gateway as a virtual machine, a hardware appliance, or a container on your on-premises environment. You can then activate the gateway using the AWS Management Console, AWS CLI, or AWS SDKs.
ii. Create and manage storage volumes: Once you have activated a gateway, you can create storage volumes, such as file volumes, tape volumes, or volume gateways, and manage them using the Storage Gateway console, CLI, or SDKs.

Database Services:

- Amazon RDS (Relational Database Service)

  Here are some AWS Database services and how to set them up.

  i. Amazon RDS (Relational Database Service):

  Amazon RDS is a managed service that provides easy-to-scale and highly available relational databases in the cloud. You can use RDS to set up, operate, and scale popular database engines like MySQL, PostgreSQL, Oracle, and SQL Server.

  To set up Amazon RDS, follow these steps:

  i. Sign in to the AWS Management Console and navigate to the RDS dashboard.
  ii. Choose the database engine you want to use and click "Create database."
  iii. Configure the database instance settings, such as the DB engine version, DB instance class, and storage type.
  iv. Set up the database security group, which controls access to the database instance.
  v. Configure backup and maintenance settings for the database instance.
  vi. Click "Create database" to launch the RDS instance.

- Amazon DynamoDB:

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance, with seamless scalability.

To set up Amazon DynamoDB, follow these steps:

i. Sign in to the AWS Management Console and navigate to the DynamoDB dashboard.
ii. Click "Create table" and specify the table details, such as the table name and primary key.
iii. Configure the table's read and write capacity settings.
iv. Define any secondary indexes that you need.
v. Set up the table's data retention policy and backup settings.
vi. Click "Create table" to launch the DynamoDB table.

- Amazon Aurora:

Amazon Aurora is a MySQL and PostgreSQL compatible relational database engine that provides up to five times the performance of standard MySQL databases. Aurora is highly available, fault-tolerant, and self-healing.

To set up Amazon Aurora, follow these steps:

i. Sign in to the AWS Management Console and navigate to the RDS dashboard.
ii. Click "Create database" and choose "Amazon Aurora" as the engine type.
iii. Specify the Aurora database settings, such as the DB instance class and storage type.
iv. Configure the Aurora cluster settings, such as the number of instances and the replication settings.
v. Set up the Aurora cluster's security group.
vi. Configure backup and maintenance settings for the Aurora cluster.
vii. Click "Create database" to launch the Aurora cluster.

- Amazon Neptune:

Amazon Neptune is a fully managed graph database service that provides high performance, scalability, and security. Neptune is optimized for managing highly connected data, such as social media data, recommendations, and fraud detection.

To set up Amazon Neptune, follow these steps:

i. Sign in to the AWS Management Console and navigate to the Neptune dashboard.
ii. Click "Create database" and specify the database settings, such as the DB instance class and storage type.
iii. Configure the database's security group.
iv. Set up the database's backup and maintenance settings.
v. Click "Create database" to launch the Neptune instance.

- Amazon ElastiCache:

Amazon ElastiCache is a managed in-memory data store that supports both Memcached and Redis caching engines. ElastiCache provides high-performance and low-latency data access, and it can be used to improve the performance of web applications, gaming applications, and other data-intensive applications.

To set up Amazon ElastiCache, follow these steps:

i. Sign in to the AWS Management Console and navigate to the ElastiCache dashboard.
ii. Click "Create" and specify the cache cluster settings, such as the engine type and node type.
iii. Configure the cache cluster's security group.
iv. Set up the cache cluster's backup and maintenance settings.
v. Click "Create" to launch the ElastiCache cluster.

Networking Services:

Here's an overview of the Networking Services offered by AWS and how to set them up:

- Amazon VPC (Virtual Private Cloud):

Amazon VPC enables you to create a virtual network in the cloud that closely resembles a traditional network that you might operate in your own data center. You can define subnets, route tables, and network gateways to create the desired topology. You can also use network ACLs and security groups to control access to your resources.

Here are the steps to set up Amazon VPC:

i. Log in to the AWS Management Console and navigate to the Amazon VPC dashboard.
ii. Choose the "Start VPC Wizard" button.
iii. Choose the "VPC with Public and Private Subnets" option and then click "Select".
iv. Specify the IPv4 CIDR block for your VPC and click "Create VPC".
v. Specify the subnets for your VPC and click "Create".
vi. Configure route tables and network gateways to enable communication between subnets and to the internet.

- Amazon Route 53:

Amazon Route 53 is a highly scalable and reliable DNS (Domain Name System) service that translates domain names into IP addresses. You can use Route 53 to route traffic to your resources such as Amazon EC2 instances, Amazon S3 buckets, and other AWS services.

Here are the steps to set up Amazon Route 53:

i. Log in to the AWS Management Console and navigate to the Route 53 dashboard.
ii. Choose the "Create Hosted Zone" button.
iii. Enter a domain name for your hosted zone and click "Create".
iv. Create resource record sets to specify how to route traffic to your resources.
v. Update the nameservers for your domain with your registrar to start using Route 53.

There are Routing Policies in AWS

In Amazon Route 53, routing policies are used to determine how traffic is routed to different resources. There are several routing policies available in Route 53, each designed for different use cases.

i. Simple Routing Policy: This policy is used when there is only one resource that is serving the traffic. In this case, the DNS returns the IP address of the resource to the client.
ii. Weighted Routing Policy: This policy is used when there are multiple resources serving the same traffic. Traffic is routed to each resource based on a weight assigned to each resource. For example, if there are two resources A and B, with weights of 60 and 40 respectively, 60% of the traffic will be routed to A and 40% to B.
iii. Latency Routing Policy: This policy is used when the resources are located in different geographic locations. The policy routes traffic to the resource that has the lowest latency or response time.
iv. Failover Routing Policy: This policy is used to route traffic to a standby resource in case the primary resource becomes unavailable.
v. Geolocation Routing Policy: This policy is used to route traffic based on the geographic location of the client.
vi. To set up these routing policies in Route 53, you need to create a hosted zone and add resource record sets for the DNS name you want to route. In the resource record set, you can specify the routing policy and the details of the resources you want to route traffic to. The routing policy can be changed or updated at any time to meet the changing needs of your application.

- AWS Direct Connect:

AWS Direct Connect is a dedicated network connection between your on-premises data center and AWS. Direct Connect provides a more consistent and reliable network experience than internet-based connections.

Here are the steps to set up AWS Direct Connect:

i. Determine the appropriate Direct Connect location for your connection.
ii. Choose a Direct Connect partner that can provide the physical connection between your data center and the Direct Connect location.

iii. Create a virtual interface in the Direct Connect console to connect your VPC to your on-premises network.
iv. Configure your router to use the Direct Connect virtual interface.

- Elastic Load Balancing:

Elastic Load Balancing (ELB) distributes incoming traffic across multiple instances to improve availability and scalability of your application. You can use ELB with Amazon EC2 instances, containers, IP addresses, and Lambda functions.

Here are the steps to set up Elastic Load Balancing:

i. Log in to the AWS Management Console and navigate to the EC2 dashboard.
ii. Choose the "Load Balancers" option in the navigation pane.
iii. Click "Create Load Balancer" and choose the appropriate load balancer type.
iv. Configure the load balancer settings such as listener ports, security groups, and target groups.
v. Register the instances or resources with the load balancer.

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

  Steps to setup Amazon CloudFront

  i. Go to the CloudFront dashboard and click on "Create Distribution"
  ii. Select the type of distribution you want to create (Web, RTMP, or Custom)
  iii. Configure the distribution settings, such as the origin, cache behavior, and SSL certificate
  iv. Once the distribution is created, you can configure additional settings, such as geo-restriction and logging

- AWS PrivateLink:

AWS PrivateLink enables you to securely access AWS services over a private connection instead of over the internet. PrivateLink provides enhanced security and compliance for accessing AWS services from within your VPC or on-premises network.

Here are the steps to set up AWS PrivateLink:

i. Determine which AWS service you want to access using PrivateLink.
ii. Create a VPC endpoint for the AWS service in your VPC.
iii. Configure the security groups to control access to the endpoint.
iv. Test the connection to the AWS service using the endpoint.

These are high-level steps for setting up each networking service. The actual steps may vary based on the specific use case and requirements. It's important to follow the AWS documentation and best practices for each service to ensure a secure and reliable infrastructure.

Analytics Services:

Here's an explanation of each of these AWS analytics services along with instructions on how to set them up:

- Amazon Redshift:
  Amazon Redshift is a fully managed data warehouse service in the cloud that makes it simple and cost-effective to analyze data using standard SQL and existing business intelligence tools. It can handle petabyte-scale data warehousing and allows you to scale up or down as needed.
  To set up Amazon Redshift, you can follow the steps outlined in the AWS documentation: https://docs.aws.amazon.com/redshift/latest/gsg/getting-started.html

- Amazon EMR (Elastic MapReduce): Amazon EMR is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS. It automatically provisions and scales compute capacity, manages the installation and configuration of software, and monitors cluster health.
  To set up Amazon EMR, you can follow the steps outlined in the AWS documentation: https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-gs.html

- Amazon Athena: Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. You don't need to set up or manage any infrastructure, and you only pay for the queries you run.
  To set up Amazon Athena, you can follow the steps outlined in the AWS documentation: https://docs.aws.amazon.com/athena/latest/ug/getting-started.html

- Amazon Kinesis: Amazon Kinesis is a platform for streaming data on AWS, making it easy to collect, process, and analyze real-time, streaming data. It can handle gigabytes of data per second from hundreds of thousands of sources, allowing you to build applications that respond to data in real-time.
  To set up Amazon Kinesis, you can follow the steps outlined in the AWS documentation: https://docs.aws.amazon.com/streams/latest/dev/getting-started.html

- AWS Glue:
  AWS Glue is a fully managed ETL (extract, transform, load) service that makes it easy to move data between data stores, clean and transform data, and load it into data warehouses, data lakes, and other data stores. It can automatically discover and

categorize your data, generate ETL code to transform it, and run the code on fully managed Apache Spark clusters.
To set up AWS Glue, you can follow the steps outlined in the AWS documentation: https://docs.aws.amazon.com/glue/latest/dg/getting-started.html

Machine Learning Services:

Here are the Machine Learning Services of AWS and how to set them up.

- Amazon SageMaker:
  Amazon SageMaker is a fully managed service that provides developers and data scientists with the ability to build, train, and deploy machine learning models quickly and easily. It provides built-in algorithms, such as Linear Learner and XGBoost, and supports popular frameworks, such as TensorFlow and PyTorch.
  To set up Amazon SageMaker, you can follow these steps:
  i. Go to the Amazon SageMaker console.
  ii. Create a new notebook instance.
  iii. Choose an instance type and configure the instance settings.
  iv. Create a new notebook and start writing code.
  v. Train and deploy your machine learning model.

- Amazon Rekognition:
  Amazon Rekognition is a deep learning-based image and video analysis service that can identify objects, people, text, scenes, and activities in images and videos. It can also detect and recognize faces, emotions, and sentiments.
  To set up Amazon Rekognition, you can follow these steps:
  i. Go to the Amazon Rekognition console.
  ii. Create a new collection or use an existing one.
  iii. Upload images or videos to the collection.
  iv. Train the model and test it.
  v. Use the APIs to integrate the model into your applications.

- Amazon Comprehend:
  Amazon Comprehend is a natural language processing (NLP) service that uses machine learning to extract insights from text. It can analyze customer feedback, social media posts, and other text data to determine sentiment, topics, entities, and language.
  i. To set up Amazon Comprehend, you can follow these steps:
  ii. Go to the Amazon Comprehend console.
  iii. Create a new project and specify the input data format.
  iv. Train the model using the built-in algorithms.
  v. Analyze the output and adjust the settings as needed.
  vi. Use the APIs to extract insights from your text data.

- Amazon Polly:
  Amazon Polly is a text-to-speech service that uses advanced deep learning technologies to synthesize natural-sounding speech from text. It can be used to create voiceovers for videos, audio books, and other multimedia content.

  To set up Amazon Polly, you can follow these steps:

  i. Go to the Amazon Polly console.
  ii. Create a new voice or use an existing one.
  iii. Enter the text you want to synthesize or upload a text file.
  iv. Customize the settings, such as the language, voice, and speed.
  v. Download or stream the audio output.

- Amazon Transcribe:
  Amazon Transcribe is a speech recognition service that can convert audio and video files into text. It supports multiple languages and can identify different speakers in a conversation.

  To set up Amazon Transcribe, you can follow these steps:

  i. Go to the Amazon Transcribe console.
  ii. Create a new job and specify the input file location and format.
  iii. Customize the settings, such as the language, audio quality, and vocabulary.
  iv. Start the transcription job and monitor its progress.
  v. Download the output file or use the APIs to retrieve the results.

Security Services:

Here's an overview of each of the AWS Security Services and how to set them up:

- AWS Identity and Access Management (IAM):
  AWS IAM is a web service that helps you securely control access to AWS resources. You can use IAM to manage users, groups, and permissions for your AWS resources. To set up IAM, follow these steps:

i. Open the AWS Management Console and navigate to the IAM dashboard.
ii. Click on "Users" and then "Add User".
iii. Enter a user name and select the access type (programmatic access, AWS Management Console access, or both).
iv. Set permissions for the user by adding the user to a group, attaching policies to the user, or creating a custom policy.

- AWS Key Management Service (KMS):

AWS KMS is a managed service that makes it easy to create and control the encryption keys used to encrypt your data.

To set up KMS, follow these steps:

i. Open the AWS Management Console and navigate to the KMS dashboard.
ii. Create a customer master key (CMK) or use a default CMK provided by AWS.
iii. Define key policies and configure key rotation options.

- AWS Certificate Manager:

AWS Certificate Manager is a service that lets you easily provision, manage, and deploy public and private SSL/TLS certificates for use with AWS services and your internal connected resources.

To set up ACM, follow these steps:

i. Open the AWS Management Console and navigate to the ACM dashboard.
ii. Request a new SSL/TLS certificate by specifying the domain name and validating ownership of the domain.
iii. Choose the service you want to use the certificate with and complete the validation process.

- AWS WAF (Web Application Firewall):

AWS WAF is a web application firewall that helps protect your web applications from common web exploits.

To set up WAF, follow these steps:

i. Open the AWS Management Console and navigate to the WAF dashboard.
ii. Create a Web ACL and add one or more rules to the ACL.
iii. Associate the Web ACL with a CloudFront distribution, an Application Load Balancer, or an API Gateway.

- AWS Shield:

AWS Shield is a managed DDoS protection service that safeguards web applications running on AWS.

To set up AWS Shield, follow these steps:

i. Open the AWS Management Console and navigate to the Shield dashboard.
ii. Choose the AWS Shield Standard or AWS Shield Advanced plan.
iii. Configure your AWS resources to use AWS Shield by enabling AWS Shield protection on your CloudFront distributions, Elastic Load Balancers, and Amazon Route 53 resources.

Application Integration

AWS provides a range of application integration services that enable businesses to build and manage complex applications by connecting various components and services. Some of the key application integration services provided by AWS include:

- Amazon Simple Notification Service (SNS): This is a messaging service that enables the exchange of messages between applications, services, and devices. SNS allows for the creation of topics and subscriptions to these topics, which can be used to send messages to multiple subscribers.

  Steps to setup Amazon SNS (Simple Notification Service):

  i. Create a topic: A topic is a communication channel where messages are sent to subscribers. You can create a topic in the SNS console or using the AWS CLI.
  ii. Subscribe endpoints: You can subscribe endpoints such as email addresses, phone numbers, or AWS Lambda functions to receive messages from the topic.
  iii. Publish messages: You can publish messages to the topic using the AWS Management Console, AWS CLI, or AWS SDKs.

- Amazon Simple Queue Service (SQS): This is a fully-managed message queuing service that enables the decoupling of different components of a cloud application. SQS allows for reliable, scalable, and asynchronous communication between different services, without the need for any message broker or other middleware.

  Steps to setup Amazon SQS (Simple Queue Service):

  i. Create a queue: A queue is a buffer that stores messages sent by producers and consumed by consumers. You can create a queue in the SQS console or using the AWS CLI.
  ii. Send messages: You can send messages to the queue using the AWS Management Console, AWS CLI, or AWS SDKs.
  iii. Poll for messages: Consumers can poll the queue to retrieve messages using the ReceiveMessage API.

- Amazon Simple Workflow Service (SWF): This is a fully-managed workflow service that enables the coordination of tasks across different services and systems. SWF allows for the automation of complex workflows and enables businesses to build reliable, scalable, and auditable applications.

  Steps to Setting up Amazon SWF:

  i. Open the Amazon SWF console and select the region in which you want to create your domain.
  ii. Create a new domain, which is a container for your Amazon SWF resources.
  iii. Create a workflow type, which is a template for your workflow.
  iv. Define the activities that your workflow uses.
  v. Start a new workflow instance and track its progress using the console.

- Amazon AppSync: This is a fully-managed service that makes it easy to develop GraphQL APIs by handling the heavy-lifting of securely connecting to data sources like AWS DynamoDB, AWS Lambda, or any HTTP data source.

  Steps to setup Amazon AppSync:

  i. Define a schema: A schema defines the types and fields of the data that the API can return. You can define a schema in the AppSync console or using the AWS CLI.
  ii. Configure resolvers: Resolvers determine how the API resolves queries and mutations. You can configure resolvers in the AppSync console or using the AWS CLI.
  iii. Test the API: You can test the API using the AppSync console or tools such as GraphiQL.

- Amazon EventBridge:
  This is a serverless event bus that makes it easy to build event-driven applications. It enables the communication between different applications, services, and AWS resources by allowing them to react to events in real-time.

  Steps in Setting up Amazon EventBridge:

  i. Open the Amazon EventBridge console and select the region in which you want to create your event bus.
  ii. Create a new event bus, which is a named message bus that can receive and route events from various sources.
  iii. Define rules that match incoming events and route them to targets such as Amazon SNS, AWS Lambda, or Amazon Kinesis.
  iv. Test your rules by generating sample events and verifying that they are properly routed.

To set up these services, businesses can use the AWS Management Console, AWS Command Line Interface (CLI), AWS SDKs, or other automation tools like AWS CloudFormation and AWS Terraform. The exact setup process may vary depending on the service and the requirements of the application. AWS provides detailed documentation, tutorials, and best practices for each service to help businesses get started quickly and easily.

This is not an exhaustive list, as AWS provides a wide range of services and features to cater to different business needs and use cases. In addition to these services, AWS also provides tools and services for DevOps, management, migration, and more.

Customer Engagement:

Customer engagement refers to the interaction between a business and its customers across various channels, such as social media, email, chatbots, and mobile apps. It involves building a relationship with customers by providing them with valuable information, personalized experiences, and support throughout their buying journey. Effective customer engagement is critical for businesses to build brand loyalty, increase customer retention, and ultimately drive revenue growth.

To achieve effective customer engagement, businesses can leverage a variety of tools and services provided by AWS, including:

- Amazon Connect:
  A cloud-based contact center service that enables businesses to provide customer service at scale through voice and chat.
  Setting up steps for Amazon Connect:
  i. Sign in to the AWS Management Console
  ii. Click on Amazon Connect under the Contact Center category
  iii. Follow the prompts to create a new Amazon Connect instance, configure your contact flows, and add your users.

- Amazon Pinpoint:
  A fully managed service for creating targeted campaigns to engage with customers through email, SMS, push notifications, and voice messages.

  Setting up steps for Pinpoint:

  i. Sign in to the AWS Management Console
  ii. Click on Amazon Pinpoint under the Mobile Services category
  iii. Follow the prompts to create a new project, define your audiences, configure your channels, and send your messages.

- Amazon SES (Simple Email Service):
  A scalable and cost-effective email service for businesses to send transactional and marketing emails.

Setting up steps for Amazon SES:

i. Sign in to the AWS Management Console
ii. Click on Amazon SES under the Messaging & Targeting category
iii. Follow the prompts to verify your email address or domain, create an email sending identity, and configure your email sending.

- Amazon SNS (Simple Notification Service):
A messaging service that enables businesses to send push notifications, SMS, and email notifications to subscribers or mobile app users.

Setting up steps for Amazon SNS:

i. Go to the Amazon SNS dashboard.
ii. Click on the "Create Topic" button.
iii. Enter a topic name and display name for the new topic.
iv. Click on the "Create Topic" button.
v. After the topic is created, click on the topic ARN to view the topic details.
vi. Click on the "Create Subscription" button.
vii. Select the protocol for the subscription, such as email, SMS, or HTTP/S.
viii. Enter the endpoint for the selected protocol.
ix. Click on the "Create Subscription" button.
x. Verify the subscription by confirming the subscription request received by the endpoint.
xi. Publish a message to the topic to test the subscription.

Note that Amazon SNS has additional features and configurations beyond these basic steps, such as message filtering, message attributes, and encryption.

- Amazon Chime: A communications service that enables businesses to conduct online meetings, video conferencing, and chat.

Setting up steps for Amazon Chime:

i. Sign in to the AWS Management Console
ii. Click on Amazon Chime under the Business Productivity category
iii. Follow the prompts to create a new Amazon Chime account, invite your users, and schedule your meetings.

- Amazon Connect Voice ID: A service that uses machine learning to authenticate customers by analyzing their unique voice patterns.

Here are the steps to set up Amazon Connect Voice ID:

i. Set up an Amazon Connect instance: To use Amazon Connect Voice ID, you need to set up an Amazon Connect instance. If you haven't done so already, follow the instructions to set up an instance.
ii. Set up an Amazon Connect Voice ID domain: Go to the Amazon Connect Voice ID console and create a domain. You can create multiple domains if you need to use different sets of data for different purposes.
iii. Set up a Voice ID speaker enrollment job: You need to create a Voice ID speaker enrollment job to train the machine learning models that will be used to authenticate customers. Follow the instructions in the Amazon Connect Voice ID console to create a speaker enrollment job.
iv. Configure Amazon Connect: In the Amazon Connect console, you need to configure your contact flows to use Amazon Connect Voice ID for authentication. You can do this by adding the Voice ID authentication block to your contact flows.
v. Test your configuration: Once you have configured Amazon Connect Voice ID, test your configuration to make sure everything is working as expected. You can do this by making test calls to your Amazon Connect instance and verifying that the authentication is working correctly.
vi. Monitor and optimize: Once you have everything set up and running, you can monitor and optimize your configuration to ensure the best possible performance. Use the Amazon Connect Voice ID console to view metrics and logs and make any necessary adjustments.

AWS SECURITY AND COMPLIANCE

A. AWS Security Best Practices

AWS Security Best Practices are a set of guidelines and recommendations that help AWS users to secure their cloud environments and protect their data, applications, and infrastructure.

Here are some of the best practices:

Use Multi-Factor Authentication (MFA): MFA adds an extra layer of security to your AWS account by requiring users to provide two or more forms of authentication to access their account.

Use Identity and Access Management (IAM): IAM allows you to manage users and their access to AWS services and resources. You should follow the principle of least privilege, which means giving users only the minimum permissions they need to perform their job.

Encrypt Data: AWS provides several encryption services, such as Key Management Service (KMS) and Server-Side Encryption (SSE), which help you to encrypt your data at rest and in transit.

Use VPC and Security Groups: Virtual Private Cloud (VPC) allows you to create a private network within AWS and control traffic flow to and from your resources. Security Groups are a key component of VPC and allow you to control inbound and outbound traffic to your resources.

Use AWS WAF and Shield: AWS Web Application Firewall (WAF) protects your web applications from common web exploits and attacks, while AWS Shield protects your resources from Distributed Denial of Service (DDoS) attacks.

Monitor and Log AWS Activity: AWS provides several monitoring and logging tools, such as CloudTrail, CloudWatch, and GuardDuty, which help you to track and log AWS activity, detect and respond to security incidents, and comply with regulatory requirements.

Use AWS Trusted Advisor: AWS Trusted Advisor is a tool that provides real-time guidance to help you optimize your AWS resources, improve security, and reduce costs.

To implement these best practices, you can follow AWS security documentation and use AWS services such as AWS Config, AWS CloudFormation, and AWS Security Hub. Additionally, you can leverage third-party tools and services, such as AWS partner solutions, to enhance your security posture.

B. IAM (Identity and Access Management)

Users, Groups, and Roles:

IAM allows you to create and manage users, groups, and roles to control access to AWS resources. Users are individual accounts for people, groups are collections of users, and roles are permissions that you can grant to AWS resources. You can set up authentication for users by creating a username and password.

Policies and Permissions:

IAM provides policies and permissions that determine which AWS resources users and groups can access. Policies are written in JSON format and define the rules that govern access to AWS resources. You can attach policies to users, groups, and roles to grant permissions to access specific resources.

Here are the steps to set up IAM on AWS:

i. Sign in to the AWS Management Console.
ii. Open the IAM console.
iii. Click on the "Users" option in the navigation pane.
iv. Click on the "Add user" button to create a new user.
v. Enter a user name and select the access type (programmatic access, AWS Management Console access, or both).
vi. If you selected "Programmatic access", create an access key for the user by clicking on the "Create access key" button.

vii. If you selected "AWS Management Console access", create a password for the user and require the user to change it on first sign-in.
viii. Click on the "Next: Permissions" button.
ix. Select the permissions you want to grant to the user. You can either assign existing policies to the user, or create a custom policy.
x. Click on the "Next: Tags" button to add tags to the user (optional).
xi. Click on the "Next: Review" button to review the user information and permissions.
xii. Click on the "Create user" button to create the user.

Once you have created a user, you can assign them to a group with specific permissions, or attach policies directly to the user. It is important to follow the principle of least privilege when assigning permissions to users or groups. This means giving them only the permissions they need to perform their job, and nothing more.

C. Encryption

KMS (Key Management Service):

KMS is a fully managed service that allows you to create and control the encryption keys used to encrypt your data. KMS provides a secure and scalable solution for key management that allows you to create, rotate, and manage keys. You can use KMS to encrypt data at rest and in transit, including data stored in Amazon S3, EBS, RDS, and Redshift.

SSE (Server-Side Encryption):

Server-side encryption (SSE) is a data encryption feature provided by AWS for S3, EBS, and RDS. SSE encrypts your data at rest using industry-standard AES-256 encryption. You can choose to use SSE-S3, which uses keys managed by S3, or SSE-KMS, which uses KMS-managed keys for more control over key management.

How to Setup KMS

To set up AWS Key Management Service (KMS), follow these steps:

i. Log in to the AWS Management Console and navigate to the KMS dashboard.
ii. Create a new KMS key by clicking on "Create key".
iii. Choose either "Symmetric" or "Asymmetric" for your key type.
iv. Define your key usage permissions and select your key administrators and users.
v. Create key policies to define the actions that are allowed or denied on the key.
vi. Enable key rotation to automatically create new versions of the key.
vii. Generate data keys to protect your data in storage.
viii. Monitor your KMS usage and audit your key usage logs.

It is important to follow AWS security best practices when setting up KMS, such as defining strict key policies, rotating keys regularly, and monitoring key usage logs.

D. Compliance

HIPAA (Health Insurance Portability and Accountability Act):

HIPAA is a US federal law that sets standards for the security and privacy of protected health information (PHI). AWS provides HIPAA compliance resources and allows customers to build HIPAA-compliant applications on AWS. AWS has signed a Business Associate Agreement (BAA) with HIPAA-covered entities to ensure that AWS is compliant with HIPAA regulations.

PCI DSS (Payment Card Industry Data Security Standard):

PCI DSS is a set of security standards that ensure the protection of cardholder data for organizations that accept credit card payments. AWS provides a PCI DSS-compliant infrastructure that includes tools and resources to help customers meet PCI DSS requirements.

SOC (Service Organization Control):

SOC is a set of standards for auditing and reporting on service organizations' controls over data security and processing. AWS has multiple SOC reports that provide independent assurance of AWS's security and compliance controls. AWS customers can use the SOC reports to help them meet their own regulatory and compliance requirements.

To set up compliance on AWS, users should follow these steps:

i. Determine the regulatory requirements that apply to their organization.
ii. Understand the compliance frameworks and services provided by AWS that can help meet those requirements.
iii. Implement the necessary policies and procedures to meet the requirements of the chosen compliance framework.
iv. Configure the AWS services to align with the organization's compliance needs.
v. Regularly audit and monitor the environment to ensure continued compliance with the regulatory requirements.
vi. Engage with AWS's compliance support services as needed.

It is important to note that while AWS provides tools and services to help users meet compliance requirements, the responsibility of compliance ultimately lies with the user. Therefore, users should regularly review and assess their compliance posture and make necessary adjustments to ensure continued compliance.

AWS Deployment and Management

AWS Deployment and Management refers to the process of deploying and managing applications, infrastructure, and services on the AWS cloud platform. This involves using a range of AWS services and tools to manage the entire application lifecycle, from development and testing to deployment, scaling, and management.

There are several AWS services and tools available for deployment and management on the platform. These include:

A. CloudFormation:

AWS CloudFormation is an Infrastructure as Code (IaC) service that provides a way to model and set up AWS resources so that you can spend less time managing resources and more time focusing on your applications that run in AWS. It allows you to create and manage a collection of AWS resources and provision them in an orderly and predictable fashion.

To set up CloudFormation, you can follow these steps:

i. Open the CloudFormation console and choose "Create stack."
ii. Select a template to create your stack, either by specifying a template file or selecting a sample template.
iii. Configure your stack options, including the stack name, parameters, tags, and permissions.
iv. Review your stack configuration, and create your stack.

B. Elastic Beanstalk:

AWS Elastic Beanstalk is a fully-managed Platform as a Service (PaaS) that allows you to deploy and manage web applications, APIs, and services in several languages like Java, .NET, PHP, Python, Node.js, Ruby, and Go. It takes care of capacity provisioning, load balancing, scaling, and application health monitoring, allowing you to focus on your application code.

To set up Elastic Beanstalk, you can follow these steps:

i. Open the Elastic Beanstalk console and create a new application.
ii. Choose the platform that best suits your application.
iii. Configure your environment settings, including the environment name, application version, instance type, and security groups.
iv. Upload your application code, either through the console, the AWS Command Line Interface (CLI), or an integrated development environment (IDE).
v. Review your configuration, and launch your environment.

C. OpsWorks:

AWS OpsWorks is a configuration management service that uses Chef and Puppet to automate server configuration, deployment, and management on AWS. It provides a simple and flexible way to create and manage stacks and layers, which are groups of resources that share a common purpose.

To set up OpsWorks, you can follow these steps:

i. Open the OpsWorks console and create a new stack.
ii. Choose a stack type, either Chef 12 or Chef 11, and select a region.
iii. Configure your stack settings, including the stack name, VPC, and subnets.

iv. Create a new layer, which defines a set of resources, such as Amazon EC2 instances, that share a common purpose.
v. Add instances to your layer, and configure them with roles, recipes, and attributes.
vi. Deploy your configuration changes to your instances.

D. CodeDeploy:

AWS CodeDeploy is a fully-managed service that automates application deployments to Amazon EC2 instances, on-premises instances, and serverless Lambda functions. It allows you to deploy your application code, update your application configuration, and automate the deployment process, making it easier to manage and scale your applications.

To set up CodeDeploy, you can follow these steps:

i. Open the CodeDeploy console and create a new application.
ii. Create a new deployment group, which defines the instances that will receive the deployment.
iii. Choose a deployment type, either in-place or blue/green, and configure your deployment settings, including the deployment configuration, traffic routing, and rollback options.
iv. Upload your application revision, either through the console, the AWS CLI, or an integrated development environment (IDE).
v. Review your deployment configuration, and start your deployment.

AWS Monitoring and Automation

A. CloudWatch:

CloudWatch is a monitoring service provided by AWS that allows you to collect and track metrics, collect and monitor log files, and set alarms.

To set up CloudWatch, follow these steps:

i. Sign in to the AWS Management Console and navigate to the CloudWatch service.
ii. From the dashboard, you can create and manage alarms, monitor metrics, and view logs.
iii. To monitor metrics, you can create custom metrics, set up dashboards, and set alarms to notify you when metrics go beyond certain thresholds.
iv. To monitor logs, you can create log groups, define log streams, and create metric filters to extract metric data from log events.
v. CloudWatch also provides insights, which allow you to run queries on logs to identify patterns, troubleshoot issues, and monitor performance.

B. Auto Scaling:

Auto Scaling is a service provided by AWS that allows you to automatically scale your compute resources based on demand.

To set up Auto Scaling, follow these steps:

i. Sign in to the AWS Management Console and navigate to the Auto Scaling service.
ii. From the dashboard, you can create and manage Auto Scaling groups, define launch configurations, and set up scaling policies.
iii. To create an Auto Scaling group, you need to define the minimum and maximum number of instances to run and the desired capacity.
iv. To define launch configurations, you need to specify the AMI, instance type, and other launch parameters.
v. To set up scaling policies, you need to define the scaling behavior, such as scaling up or down based on CPU utilization, and set up alarm thresholds to trigger scaling actions.

C. AWS Config:

AWS Config is a service provided by AWS that allows you to manage and monitor your AWS resources' configurations.

To set up AWS Config, follow these steps:

i. Sign in to the AWS Management Console and navigate to the AWS Config service.
ii. From the dashboard, you can create and manage AWS Config rules, view compliance reports, and monitor resource inventory.
iii. To create AWS Config rules, you need to define the rule criteria, such as checking for security group rules that allow unrestricted inbound traffic.
iv. Once you create the rules, you can view compliance reports that show which resources are compliant and which are not.

AWS Config also provides a resource inventory, which allows you to view a detailed inventory of your AWS resources.

D. AWS Lambda:

AWS Lambda is a serverless compute service provided by AWS that allows you to run code without provisioning or managing servers.

To set up AWS Lambda, follow these steps:

i. Sign in to the AWS Management Console and navigate to the Lambda service.
ii. From the dashboard, you can create and manage Lambda functions, set up triggers, and monitor function invocations.

iii. To create a Lambda function, you need to define the function code, runtime, and other function parameters.
iv. To set up triggers, you can choose from a variety of event sources, such as S3, DynamoDB, or API Gateway.

Once you have set up your function and triggers, you can monitor function invocations, view logs, and troubleshoot issues.

AWS COST OPTIMIZATION

A. AWS Cost Management Tools:

AWS Cost Explorer: AWS Cost Explorer is a tool that helps you to visualize, understand and manage your AWS costs and usage over time. It provides a graphical representation of your AWS cost and usage data, which can be filtered and grouped by various dimensions such as service, region, tag, and more. You can also create custom reports and export data for further analysis.

To set up AWS Cost Explorer, you must have an AWS account and a billing and cost management role assigned to your account. You can access AWS Cost Explorer from the AWS Management Console.

AWS Budgets: AWS Budgets is a tool that enables you to set custom budgets and receive alerts when your actual or forecasted AWS usage exceeds your budget. With AWS Budgets, you can set up budget plans, create customized alerts, and forecast your costs.

To set up AWS Budgets, you need an AWS account and a billing and cost management role assigned to your account. You can access AWS Budgets from the AWS Management Console.

B. EC2 Cost Optimization:

Reserved Instances: Reserved Instances are a way to save money on your Amazon EC2 usage by committing to a certain amount of usage in advance. You can save up to 75% compared to on-demand pricing by purchasing Reserved Instances. There are three types of Reserved Instances: Standard, Convertible, and Scheduled.

To set up Reserved Instances, you must have an active AWS account and access to the Amazon EC2 console. You can purchase Reserved Instances through the console or the AWS Marketplace.

Spot Instances: Spot Instances are a way to save money on your Amazon EC2 usage by bidding on unused EC2 capacity. You can save up to 90% compared to on-demand pricing by using Spot Instances. However, since Spot Instances are not guaranteed, they may be terminated at any time if the price exceeds your bid.

To use Spot Instances, you need an active AWS account and access to the Amazon EC2 console. You can launch Spot Instances from the console or by using the AWS CLI or API.

C. S3 Cost Optimization:

Object Lifecycle Management: Object Lifecycle Management is a way to automate the transition of your S3 objects to different storage classes or to delete them. By using Object Lifecycle Management, you can reduce your storage costs and improve your data management.

To set up Object Lifecycle Management, you need an active AWS account and access to the Amazon S3 console. You can create lifecycle policies in the console or by using the AWS CLI or API.

S3 Intelligent-Tiering: S3 Intelligent-Tiering is a storage class that automatically moves your S3 objects between two access tiers based on changing access patterns and cost. With S3 Intelligent-Tiering, you can optimize costs for infrequently accessed data.

To use S3 Intelligent-Tiering, you need an active AWS account and access to the Amazon S3 console. You can create an S3 Intelligent-Tiering bucket or add the S3 Intelligent-Tiering class to an existing bucket through the console or by using the AWS CLI or API.

D. RDS Cost Optimization:

Reserved Instances: Reserved Instances are a way to save money on your Amazon RDS usage by committing to a certain amount of usage in advance. You can save up to 72% compared to on-demand pricing by purchasing Reserved Instances.

To set up Reserved Instances, you need an active AWS account and access to the Amazon RDS console. You can purchase Reserved Instances through the console or the AWS Marketplace.

Aurora Capacity Units: Aurora Capacity Units are a new way to purchase and consume Aurora database resources. With Aurora Capacity Units, you can purchase database capacity on an as-needed basis, rather than paying for a fixed amount of capacity.

Case Studies and Best Practices

A. Real-world scenarios

Web Applications: AWS is widely used for web applications and websites due to its scalability, reliability, and cost-effectiveness. Companies like Netflix, Airbnb, and LinkedIn use AWS for their web applications.

Big Data Analytics: AWS provides a range of services that can be used for big data analytics, including Amazon EMR, Amazon Redshift, and Amazon Kinesis. Companies like Nasdaq, Airbnb, and Yelp use AWS for their big data analytics needs.

Media and Entertainment: AWS offers a range of services that are ideal for media and entertainment companies, including Amazon S3 for storage, Amazon CloudFront for content delivery, and AWS Elemental Media Services for video processing. Companies like Netflix, BBC, and ESPN use AWS for their media and entertainment needs.

B. Best practices for AWS

Security: AWS offers a range of security features and services, but it is important to configure them properly to ensure maximum security. Best practices for AWS security include using multi-factor authentication, setting up network security groups, and using encryption.

Performance: To ensure optimal performance on AWS, it is important to properly size and configure resources, monitor performance metrics, and use autoscaling to ensure that resources are available as demand changes.

Cost Optimization: AWS offers a range of cost optimization tools and features, but it is important to use them properly to avoid unnecessary costs. Best practices for AWS cost optimization include using reserved instances, using spot instances for non-critical workloads, and using S3 lifecycle policies to move data to less expensive storage tiers.

Availability: Ensure that your applications and services are available to users at all times by leveraging multiple availability zones, implementing redundancy and failover mechanisms, and using auto scaling to handle traffic spikes.

Scalability: Plan and design your applications and infrastructure to be scalable and able to handle growing demand. Use services like Amazon Elastic Compute Cloud (EC2) and Amazon Elastic Container Service (ECS) to scale up or down based on traffic.

Monitoring and Alerting: Set up proactive monitoring and alerting to detect and respond to issues before they impact your application or service. Use AWS CloudWatch to monitor your resources and applications and set up alarms to notify you when issues arise.

Automation: Automate routine tasks and processes to increase efficiency and reduce errors. Use AWS CloudFormation to create and manage infrastructure as code and AWS Lambda to automate event-driven computing tasks.

Disaster Recovery: Plan and implement a disaster recovery strategy to minimize downtime and data loss in the event of a disaster. Use services like Amazon Elastic Block Store (EBS) to create backups and Amazon Route 53 to manage DNS failover.

Compliance: Ensure that your applications and infrastructure are compliant with relevant regulations and standards. Use AWS Config to audit and track changes to your AWS resources and use AWS Key Management Service (KMS) to encrypt and protect sensitive data.

Training and Education: Invest in training and education to ensure that your team has the knowledge and skills to effectively use AWS services and optimize your infrastructure. AWS offers a range of training and certification options to help you build your skills and expertise.

AWS GLOBAL Infrastructure

Sure, here's more information on AWS Global Infrastructure:

Regions and Availability Zones:

Regions are separate geographic areas that host at least two or more Availability Zones (AZs) with a low-latency network connection. Each region is isolated from other regions, which provides fault tolerance, redundancy, and compliance requirements.

Availability Zones (AZs) are data centers that are separated from other AZs in the same region, and are connected to each other by a high-bandwidth, low-latency network. Each AZ is designed to be independent and has its own power, cooling, and networking capabilities.

Regions and AZs are important for several reasons:

Latency: By having multiple regions, users can deploy their applications closer to their customers, reducing latency and providing better performance.

Compliance: Some regions are designed for specific compliance requirements, such as HIPAA compliance in the US-East region, which is important for healthcare providers and businesses.

Disaster recovery: By deploying applications across multiple AZs in a region or across multiple regions, users can have a disaster recovery plan in place that allows their applications to continue running in the event of a failure.

Edge Locations: Edge Locations are endpoints that are used to cache and deliver content to end users with low-latency performance. They are located in major cities around the world and are used to cache and deliver static and dynamic content, such as videos, images, and web pages.

Edge Locations are used for a variety of use cases, such as:

Content delivery: By caching content at edge locations, users can improve the performance of their applications and reduce the load on their origin servers.

Security: Edge locations are used for AWS security services, such as AWS Web Application Firewall (WAF) and AWS Shield, to protect applications against DDoS attacks and other web-based attacks.

IoT and mobile devices: Edge locations are used for AWS IoT services, such as AWS IoT Greengrass, which allows users to run IoT applications at the edge, closer to the devices.

Services by Region:

AWS services and features are available in different regions and AZs. Some services are available globally, while others are specific to a region or AZ. It is important for users to check the availability of the services and features they need before deploying their applications.

AWS offers an overview of global services and features that are available in all regions, as well as services and features that are specific to a region or AZ. Users can also use the AWS Region Table to check the availability of services and features by region.