

Cloud Computing-Curriculum

1. Introduction to Cloud Computing

1.1 Definition and basic concepts of cloud computing

- What is Cloud Computing
- Advantages and disadvantages of cloud computing

1.2 Cloud deployment models (public, private, hybrid)

1.3 Cloud Service Models

- Infrastructure as a Service (IaaS),
- Platform as a Service (PaaS),
- Software as a Service (SaaS).

1.4 Virtualization

1.5 Cloud Providers

2. Introduction to AWS

2.1 Overview of AWS services

2.2 Creating an AWS account and setting up billing

2.3 Basic AWS architecture and services (EC2, S3, RDS)

Compute:

- EC2, Lambda
- Elastic Beanstalk
- Batch

Storage:

- S3
- EBS
- Glacier
- Elastic File System

Database:

- RDS
- DynamoDB
- ElastiCache

Networking:

- VPC
- Route 53
- Direct Connect

Cloud Computing-Curriculum

Security:

- IAM
- WAF
- Shield

2.4 Deploying applications on AWS

Introduction to Azure

3.1 Overview of Azure services

3.2 Creating an Azure account and setting up billing

3.3 Basic Azure architecture and services (Virtual Machines, Blob Storage, SQL, Database)

Compute:

- Virtual Machines
- Azure Functions
- App Service
- Batch

Storage:

- Blob Storage
- File Storage
- Managed Disks

Database:

- Azure SQL Database
- Cosmos DB
- Redis Cache

Networking:

- Virtual Network
- Azure DNS
- ExpressRoute

Security:

- Azure Active Directory,
- Azure Firewall
- Security Center

3.4 Deploying applications on Azure

Cloud Computing-Curriculum

4. Introduction to GCP

4.1 Overview of GCP services

4.2 Creating a GCP account and setting up billing

4.3 Basic GCP architecture and services (Compute Engine, Cloud Storage, Cloud SQL)

Compute Services:

- Google Compute Engine (GCE)
- Google Kubernetes Engine (GKE)
- App Engine
- Cloud Functions

Storage Services:

- Google Cloud Storage
- Persistent Disk
- Cloud Filestore
- Cloud Storage for Firebase

Database Services:

- Google Cloud Spanner
- Google Cloud SQL
- Google Cloud Bigtable
- Google Cloud Firestore

Networking Services:

- Virtual Private Cloud (VPC)
- Cloud Load Balancing
- Cloud CDN
- Cloud DNS

Security and Identity Services:

- Identity and Access Management (IAM)
- Cloud Key Management Service
- Cloud Identity
- Cloud Armor

Management and Monitoring Services:

- Cloud Logging
- Cloud Monitoring
- Cloud Trace
- Cloud Profiler

Cloud Computing-Curriculum

BigQuery

- Cloud Dataflow
- Cloud Pub/Sub
- Cloud Dataproc

4.4 Deploying applications on GCP

5. Comparing AWS, Azure, and GCP

- 5.1 Key differences between AWS, Azure, and GCP
- 5.2 Selecting the right cloud service provider based on business requirements

6. Managing and Monitoring Cloud-based Solutions

- 6.1 Best practices for managing and monitoring cloud-based solutions
- 6.2 AWS, Azure, and GCP monitoring tools and services

7. Conclusion and Next Steps

- 7.1 Recap of the course content
- 7.2 Recommended next steps for further learning and certification

Cloud Computing-Curriculum

CAPSTONE PROJECTS

1 Host a Personal Website using Azure Virtual Machines

- This project involves hosting a personal website using an Azure Virtual Machine (VM) as the server infrastructure.
- The website is built with HTML, CSS, JavaScript, and Bootstrap to ensure a responsive and modern design.
- Azure VM provides a scalable and reliable cloud environment for deploying the static website.
- The process includes configuring the VM, setting up web server software, and deploying the site files.
- This approach leverages cloud computing to deliver a cost-effective and efficient hosting solution for personal or portfolio websites.

2 Host a Self Scaling Full stack application using AWS Elastic Beanstalk

- This project involves deploying a full-stack Laravel application on AWS using Elastic Beanstalk.
- The application is designed to automatically scale based on demand, leveraging AWS's managed infrastructure to handle server provisioning, load balancing, and scaling.
- Elastic Beanstalk simplifies the deployment, making it an ideal choice for hosting a dynamic, self-scaling web app.

3 Serverless Personal To-Do List Application.

- The Serverless Personal To-Do List Application is a scalable and cost-effective web-based solution built using AWS services
- It allows users to manage their tasks with features such as adding, viewing, updating, and deleting to-do items.
- The application leverages AWS Lambda for backend logic, Amazon API Gateway for API management, and DynamoDB for data storage, all within a serverless architecture.
- User authentication and authorization are handled by AWS Cognito, ensuring secure access to the application.
- Hosted on Amazon S3 and distributed via CloudFront, this project showcases the efficiency and scalability of serverless computing while introducing essential AWS services and best practices.

LIVE PROJECT

1 Develop a User Authentication System with AWS Cognito and LAMP Architecture.

- This project aims to develop a robust User Authentication System utilising AWS Cognito for secure and scalable identity management, integrated with a LAMP stack-based architecture.
- The system leverages AWS Cognito for handling user registration, login, and authentication.
- On the software front, the Laravel framework is used to build the frontend and backend, offering a seamless user experience with enhanced security and performance.
- The integration of AWS Cognito with Laravel simplifies the process of managing tokens and secure sessions.
- This project ensures compliance with modern security standards while maintaining scalability for handling high traffic and a growing user base.