# Assignment 1
# Cloud Computing Architecture and Deployment Models

**Dated: 17th August 2025**
**Last date of Submission: 25th September 2025**
**Total Marks: 20**

**Objective**
This assignment is designed to help students **apply cloud computing concepts** by designing, implementing, and deploying a real-world application using **AWS cloud services**. Students will learn to architect scalable, resilient, and secure solutions while using **at least 5 AWS services**.

**General Requirements for All Tracks**

- Use **at least 5 AWS services**.
- Provide **architecture diagram** showing components and flow.
- Include **deployment documentation** with AWS screenshots and video.
- Demonstrate **scalability or automation** where applicable.
- Include **security considerations** (IAM roles, policies, or security groups).
- Provide **repository link, presentation, documentation** and a **demo video (5–10 min)**.

**Evaluation Criteria (20 Marks)**

- Architecture & Service Usage – 7 marks
- Implementation & Deployment – 7 marks
- Scalability, Automation & Security – 3 marks
- Documentation & Presentation – 3 marks

## Assignment Tracks (Choose One)

### *Track A: SaaS Application Development*

**Goal:** Build a simple **multi-tenant SaaS application** (e.g., online notes, task manager, attendance portal).
**Tasks:**

- Design the application architecture.
- Implement frontend and backend.
- Deploy the application in AWS.

**Minimum AWS Services (5+):**

1. **Elastic Beanstalk / ECS / EKS/EC2** – for hosting the application
2. **RDS or DynamoDB** – for database storage
3. **S3** – for storing static files or media
4. **Cognito** – for user authentication
5. **CloudFront or Load Balancer** – for distribution and performance

**Deliverables:** Architecture diagram, AWS deployment screenshots/video, working application demo, code repository and final presentation and report

### *Track B: E-commerce Web Application*

**Goal:** Create a **small e-commerce platform** with catalog, cart, and order management.
**Tasks:**

- Design a scalable and highly available architecture.
- Implement backend, frontend, and database connectivity.
- Deploy the application in AWS.

**Minimum AWS Services (5+):**

1. **EC2** – for backend hosting
2. **RDS** – for relational database
3. **S3** – for storing product images
4. **Elastic Load Balancer (ELB)** – for distributing traffic
5. **Auto Scaling Group** – to handle load fluctuations

**Deliverables:** Architecture diagram, deployment screenshots, working demo, repository link.

### *Track C: Big Data Analytics Pipeline*

**Goal:** Build a **data processing pipeline** for tasks such as clickstream analysis, IoT logs, or customer insights.
**Tasks:**

- Design data ingestion, processing, and visualization workflow.
- Deploy at least one functional stage of the pipeline.

**Minimum AWS Services (5+):**

1. **S3** – for data lake/storage
2. **Kinesis / Data Streams** – for real-time data ingestion
3. **EMR / Glue / Lambda** – for processing or ETL
4. **Redshift** – for analytics storage
5. **QuickSight** – for visualization

**Deliverables:** Pipeline architecture diagram, deployment notes, screenshots, demo, code repo.

### *Track D: Serverless Application*

**Goal:** Build an **event-driven serverless application**, such as image processing or notification system.

**Tasks:**

- Design event-driven workflow.
- Implement backend logic using serverless compute.
- Deploy application on AWS.

**Minimum AWS Services (5+):**

1. **Lambda** – for compute
2. **API Gateway** – to expose API endpoints
3. **S3** – for data storage and event triggers
4. **DynamoDB** – for NoSQL data storage
5. **SNS / SQS** – for messaging and notifications

**Deliverables:** Workflow diagram, deployment screenshots, working demo, code repo.

### *Track E: AI/ML Model Deployment*

**Goal:** Train and deploy a **machine learning model** (e.g., text classification, image recognition, sales prediction).

**Tasks:**

- Train the model on AWS.
- Deploy the model as an API endpoint.

**Minimum AWS Services (5+):**

1. **SageMaker** – for model training and deployment
2. **S3** – for storing datasets
3. **Lambda** – for lightweight inference functions
4. **API Gateway** – to expose model as API
5. **CloudWatch** – for monitoring

**Deliverables:** ML architecture diagram, deployment screenshots, working inference demo, code repo.

### *Track F: IoT and Cloud Integration*

**Goal:** Build an **IoT system** (e.g., smart home sensors, health monitoring, vehicle tracking).

**Tasks:**

- Design data collection, processing, and visualization workflow.
- Deploy a functional demo of IoT data ingestion and processing.

**Minimum AWS Services (5+):**

1. **AWS IoT Core** – for device connectivity
2. **Greengrass** – for edge processing

3. **Kinesis / IoT Analytics** – for data ingestion
4. **DynamoDB / S3** – for storage
5. **QuickSight** – for dashboards

**Deliverables:** Architecture diagram, deployment screenshots, demo, code repo.

### *Track G: DevOps & CI/CD Implementation*

**Goal:** Build a **CI/CD pipeline** to deploy a sample application on AWS.

**Tasks:**

- Implement **automated build, test, and deployment workflow**.
- Use Infrastructure as Code (IaC) templates for automation.

**Minimum AWS Services (5+):**

1. **CodePipeline** – for workflow automation
2. **CodeBuild** – for building the code
3. **CodeDeploy** – for automated deployment
4. **CloudFormation / Terraform** – for IaC
5. **EC2 / ECS** – for application hosting

**Deliverables:** Pipeline diagram, deployment screenshots, automated demo, IaC files repo.