

AWS Cloud Responsibility Model Research & Documentation

1. Introduction to the Shared Responsibility Model

AWS (Amazon Web Services) follows a **Shared Responsibility Model** to define the boundary between **AWS's responsibilities** and the **customer's responsibilities** when using cloud services. This model varies based on the **type of cloud service** used—**Infrastructure as a Service (IaaS)**, **Platform as a Service (PaaS)**, or **Software as a Service (SaaS)**.

- **AWS is responsible for "Security of the cloud"** — the infrastructure that runs all the services offered in the AWS Cloud.
 - **Customers are responsible for "Security in the cloud"** — based on the service used, this may include data, identity, application configuration, etc.
-

2. Cloud Service Models in AWS

2.1 Infrastructure as a Service (IaaS)

IaaS provides basic building blocks such as virtual machines, storage, and networking. Customers manage:

- Operating systems
- Middleware
- Applications
- Security patches
- Runtime

AWS Examples:

- **Amazon EC2 (Elastic Compute Cloud)**
- **Amazon EBS (Elastic Block Store)**

- **Amazon VPC (Virtual Private Cloud)**

2.2 Platform as a Service (PaaS)

PaaS provides a platform allowing customers to develop, run, and manage applications without worrying about the underlying infrastructure. AWS manages:

- OS and platform patching
- Load balancing
- Database engine
- Scaling

AWS Examples:

- **Amazon RDS (Relational Database Service)**
- **AWS Lambda**
- **AWS Elastic Beanstalk**

2.3 Software as a Service (SaaS)

SaaS delivers fully functional applications over the internet. Users do not manage infrastructure or application configuration.

AWS Examples:

- **Amazon Chime**
- **Amazon WorkDocs**
- **AWS Marketplace SaaS products**

3. Classification of AWS Services

Below is a classification of selected AWS services under IaaS, PaaS, or SaaS with explanations:

AWS Service	Model	Justification
EC2	IaaS	EC2 provides raw virtual machines. Customers manage OS, middleware, and applications. It's a classic IaaS offering.

RDS	PaaS	RDS abstracts database engine management (MySQL, PostgreSQL, etc.). AWS handles backups, patching, and scaling.
S3	PaaS	Though S3 is a storage service, users don't manage the underlying storage infrastructure. AWS ensures durability, availability, and scaling.
DynamoDB	PaaS	AWS fully manages this NoSQL database. Users focus on table structure and queries. No need to provision or patch servers.
Redshift	PaaS	Amazon Redshift is a fully managed data warehouse service. AWS handles cluster provisioning, patching, and backups.

4. Explanation of Each Classification

4.1 Amazon EC2 – Infrastructure as a Service

- **Control level:** Maximum
- **Customer's responsibility:** Everything from OS up to applications
- **Use case:** Custom application hosting, container orchestration, virtual desktops
- **Why IaaS:** EC2 provides compute capacity with no pre-installed OS or software. Customers must configure everything.

4.2 Amazon RDS – Platform as a Service

- **Control level:** Medium
- **Customer's responsibility:** Schema design, query tuning, access control
- **Use case:** Relational database workloads (MySQL, PostgreSQL, Oracle)
- **Why PaaS:** RDS manages the database infrastructure. Users don't patch the engine or configure HA/failover.

4.3 Amazon S3 – Platform as a Service

- **Control level:** Limited

- **Customer's responsibility:** Data access, security policies, lifecycle rules
- **Use case:** Object storage for backups, logs, data lakes
- **Why PaaS:** AWS abstracts the entire storage architecture. Users get an interface to manage buckets and objects.

4.4 Amazon DynamoDB – Platform as a Service

- **Control level:** Minimal
- **Customer's responsibility:** Table design, item-level access, query performance
- **Use case:** Serverless applications, low-latency key-value stores
- **Why PaaS:** No infrastructure provisioning needed. AWS handles scaling, replication, and availability.

4.5 Amazon Redshift – Platform as a Service

- **Control level:** Moderate
- **Customer's responsibility:** Data loading, SQL queries, schema design
- **Use case:** Big data analytics, reporting
- **Why PaaS:** AWS manages clusters, performance tuning, and backups automatically. It's a fully managed data warehouse.