

1. AWS Cloud Computing ☁

Definition: On-demand IT resources (compute, storage, DB) over the Internet, pay-as-you-go.

5 Characteristics (NIST model):

- ▶ On-Demand Self-Service → provision instantly.
- ▶ Broad Network Access → access via devices/APIs.
- ▶ Resource Pooling → multi-tenant, shared.
- ▶ Rapid Elasticity → scale up/down.
- ▶ Measured Service → pay per use.

Benefits of AWS:

- ▶ Agility; Faster innovation, deploy in minutes.
- ▶ Cost savings: Handle spikes, avoid overprovisioning.
- ▶ Global reach: OPEX model, Reserved/Spot instances.
- ▶ Scalability: Deploy in multiple regions/AZs.
- ▶ Security: IAM, encryption, compliance standards.
- ▶ Innovation: ML, IoT, serverless, analytics.

Limitations + Mitigation:

Compliance:

Regulations (GDPR, HIPAA) → use AWS Artifact, regional data residency.

Latency:

Distance issues → use CloudFront, Edge Locations, Local Zones.

Vendor lock-in:

Hard to migrate → use hybrid/multi-cloud strategies.

2. AWS Networking & Security 🛡

VPC = isolated virtual network.

- ▶ **Subnets** → Public (Internet GW) / Private.
- ▶ **Route Tables** → traffic rules.
- ▶ **Security Groups (SGs)** → instance firewall (stateful, allow only).
- ▶ **NACLs** → subnet firewall (stateless, allow/deny).

Connectivity Options:

- ▶ VPN (encrypted tunnel)
- ▶ Direct Connect (dedicated private line)
- ▶ Hybrid Cloud = on-prem + AWS

6. Compute Services 💻

EC2	Resizable VMs
Lambda	Serverless compute, pay per request
Elastic Beanstalk	PaaS for app deployment
ECS/EKS	Containers

7. Storage Services ☁

Service	Type	Use Case
EBS	Block	Attach to EC2
S3	Object	General storage
EFS	File	Shared, scalable
Glacier	Archival	Backups, compliance

9. Cloud Architecture Design 🏗

- **Trusted Advisor** → recommends on cost, performance, security.
- **High Availability** → deploy across AZs & Regions.
- **Reliability** → backups, failover, replication.

3. AWS Architectures 🏗

Core Services:

- ▶ **EC2** = scalable VMs
- ▶ **S3** = object storage (11 9's durability)
- ▶ **Glacier** = archival storage

Optimisation:

- ▶ Right-size EC2, Auto Scaling
- ▶ Reserved Instances, Spot Instances
- ▶ Elastic Load Balancing

Well-Architected Framework → 5 pillars:

1. Operational Excellence
2. Security
3. Reliability
4. Performance Efficiency
5. Cost Optimisation

4. Automation & Serverless 🤖

- **CloudFormation** = Infrastructure as Code (IaC).
- **Lambda** = serverless, event-driven, stateless.
- **API Gateway** = expose Lambda as APIs.
- **Monitoring**: CloudWatch (metrics), X-Ray (tracing).

5. Identity & Access Management (IAM) 🔑

IAM Components:	Users, Groups, Roles, Policies.
Risks:	* : * permissions, long-lived keys.
Mitigation:	IAM Access Analyser, CloudTrail audits.
Best Practices:	MFA, Rotate keys, Avoid root account, Cross-account roles

Principle: Least Privilege (grant minimum required).

8. Database Services 🗄️

- **RDS** (SQL, managed) → backups, scaling.
 - **Aurora** → high-perf managed DB.
 - **DynamoDB** → NoSQL, serverless, key-value.
 - **Redshift** → analytics/data warehouse.
- SQL vs NoSQL:**
- **SQL** = structured, ACID, relational.
 - **NoSQL** = schema-less, horizontal scaling.

10. Scalability & Monitoring 🚦

- **Elastic Load Balancing (ELB)** → distributes traffic.
- **Auto Scaling** → adds/removes EC2 instances.
- **CloudWatch** → monitors metrics in real-time.



By **DiaVedu25**
cheatography.com/diavedu25/

Not published yet.
Last updated 8th September, 2025.
Page 2 of 2.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopad.com>