PW SKILLS | DevOps and Cloud Computing

Introducing to Cloud Computing & AWS

Objective

- Understand cloud computing services and architectures for compute, storage, and networking.
- Compare AWS, Azure, and GCP in terms of features, pricing and best use cases.
- Learn security, compliance, and cost optimization strategie in cloud environments.
- Explore availability and scalability concepts like multiple regions and Availability Zones (AZs).
- Identify how logs, metrics, and user traffic monitoring support production DevOps workflows.

PW SKILLS

Explaining the key cloud service models

Let's see

- Infrastructure as a Service (IaaS): Provides virtualized compute, storage, and networking resources. You manage the OS and applications.
- Platform as a Service (PaaS): Offers managed development environments for building, testing, and deploying apps without handling infrastructure.
- Software as a Service (SaaS): Delivers ready-to-use applications to end users over the internet; everything is managed by the provider.

PW SKILLS

Introducing compute options (EC2, Lambda, Containers, Kubernetes) and when to use them.

Compute options

1. **EC2 (Elastic Compute Cloud)**

What it is:

Virtual servers in the cloud with full control over the OS and environment.

When to use:

- You need custom configurations or full control over the OS.
- You’re running legacy apps or long-running processes.
- You require persistent workloads and manual scaling.

Compute options

2. AWS Lambda (Serverless Functions)

What it is:

Event-driven compute that runs code in response to triggers—no server management.

When to use:

- For short-lived, event-based tasks (e.g., image processing, API backends).
- You want auto-scaling and pay-per-use.
- Ideal for microservices or automation scripts.

Compute options

3. Containers (e.g., Docker on ECS or Fargate)

What it is:

Lightweight, portable application environments with all dependencies bundled.

When to use:

- For modern, microservice architectures.
- You need portability, isolation, and faster deployment.
- Better resource utilization than VMs (like EC2).

Compute options

4. Kubernetes (EKS on AWS)

What it is:

An orchestration system for managing containers at scale.

When to use:

- You’re deploying many containers across clusters.
- Need automated scaling, self-healing, and rolling updates.
- Best for complex, distributed systems with high availability needs.

PW SKILLS

Discussing storage types (S3, EBS, RDS, DynamoDB) and their use cases.

Let’s discuss

- **S3 (Simple Storage Service):**

Object storage for unstructured data like backups, media files, and logs.

Use for: Static websites, data lakes, archival storage.

- **EBS (Elastic Block Store):**

Block storage for EC2 instances, like virtual hard drives.

Use for: Databases, file systems, boot volumes.

Let’s discuss

- **RDS (Relational Database Service):**

Managed SQL database (e.g., MySQL, PostgreSQL).

Use for: Structured data with complex queries and transactions.

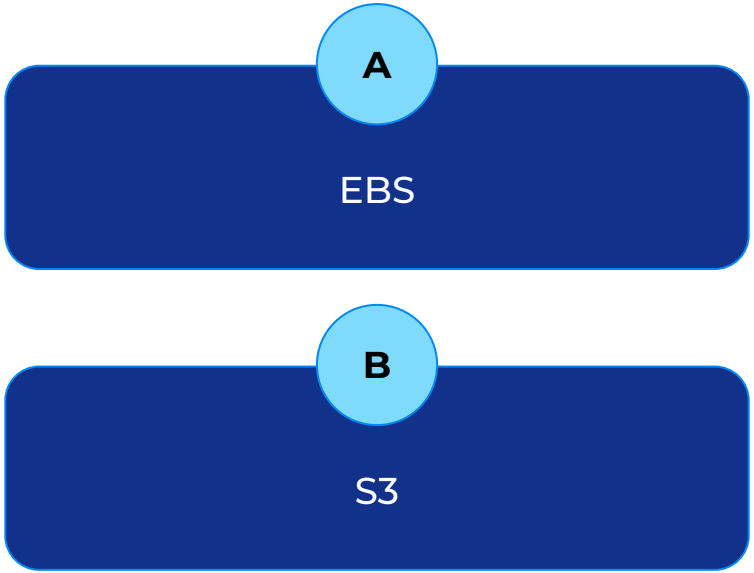
- **DynamoDB:**

Fully managed NoSQL database for low-latency key-value or document storage.

Use for: High-speed apps, IoT, gaming, real-time analytics.

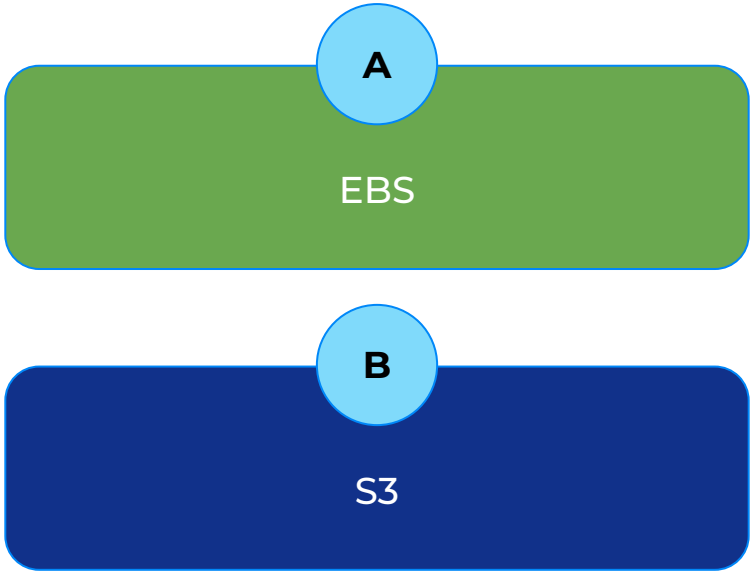
Pop Quiz

Q. Which service provides block-level storage that can be attached to EC2 instances?



Pop Quiz

Q. Which service provides block-level storage that can be attached to EC2 instances?



PW SKILLS

Explaining networking concepts (VPCs, subnets, security groups, load balancers).

Let’s see

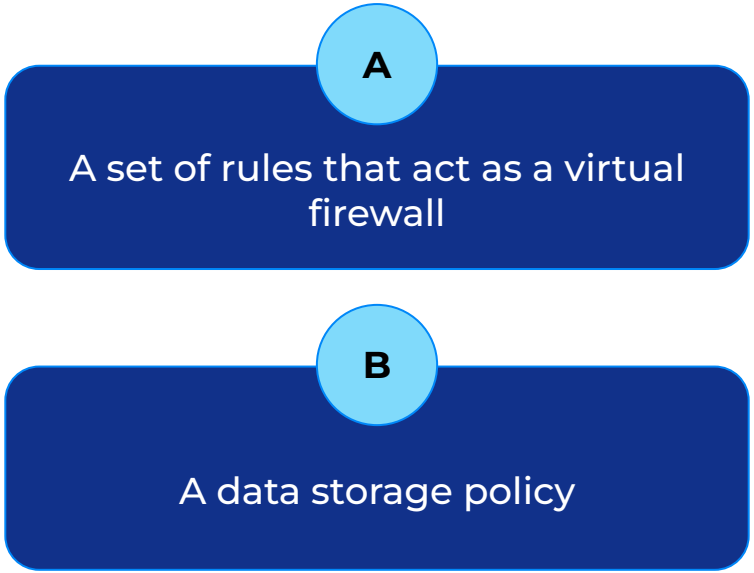
1. **VPC (Virtual Private Cloud):**
    - A private, isolated network in the cloud where you launch resources like EC2 instances.
    - Think of it as your own data center in the cloud.
2. **Subnets:**
    - Subdivisions of a VPC used to group resources.
    - Public subnets: Accessible from the internet (e.g., web servers).
    - Private subnets: Internal-only access (e.g., databases).

Let's see

3. **Security Groups:**
    - Virtual firewalls that control inbound and outbound traffic for cloud resources.
    - You define rules to allow/deny specific IPs, ports, and protocols.
4. **Load Balancers:**
    - Distribute traffic across multiple servers to ensure high availability and fault tolerance.
    - Common types:
    - Application Load Balancer (ALB): For HTTP/HTTPS traffic.
    - Network Load Balancer (NLB): For high-performance TCP traffic.

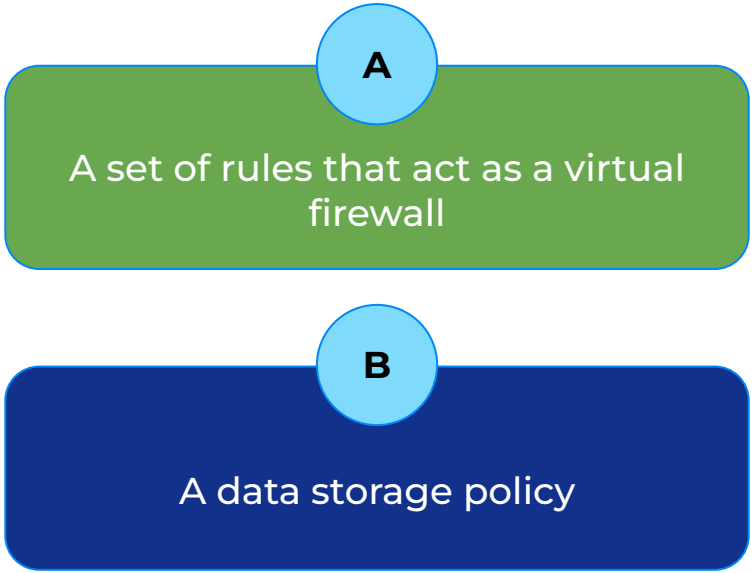
Pop Quiz

Q. Which of the following best describes a Security Group?



Pop Quiz

Q. Which of the following best describes a Security Group?



PW SKILLS

Discussing cloud security principles (IAM, encryption, compliance standards).

Let’s discuss

IAM (Identity and Access Management):

- Controls who can access what in the cloud using users, roles, and policies.
- Use least privilege principle.

Encryption:

- Protects data in transit and at rest using encryption keys (e.g., KMS in AWS).
- Ensures confidentiality and integrity.

Compliance Standards:

- Cloud providers support standards like ISO 27001, HIPAA, GDPR to meet legal and regulatory requirements.
- Helps build trust and meet governance needs.

PW SKILLS

Explaining cost optimization strategies

Let’s see

1. **Right-Sizing Resources**
    - **Reserved Instances:** Commit to long-term use (1–3 years) for lower rates.
    - **Auto-Scaling:** Automatically adjusts compute capacity based on demand—saves money during low usage.

    Goal: Avoid overprovisioning.

2. **Serverless and Spot Instances**
    - **Serverless (e.g., AWS Lambda):** Pay only when your code runs—no idle costs.
    - **Spot Instances:** Use spare capacity at up to 90% discount—best for non-critical, interruptible workloads.

Let’s see

3. **Storage Lifecycle Policies**

    - **S3 Intelligent Tiering:** Automatically moves data between storage tiers based on usage.
    - **Cold Storage (e.g., S3 Glacier):** Store infrequently accessed data at very low cost.

PW SKILLS

Introducing cloud migration strategies

Cloud migration strategies

1. **Rehost (Lift & Shift):**

   Move existing applications to the cloud without major changes.
   Fastest option; good for quick migrations with minimal risk.

2. **Refactor (Cloud-Native Rebuild):**

   Redesign applications to take advantage of cloud features like auto-scaling, serverless, and managed services.
   Best for long-term efficiency, scalability, and performance.

3. **Hybrid/Multi-Cloud:**

   Use a mix of on-premises and cloud infrastructure (hybrid) or multiple cloud providers (multi-cloud).
   Useful for compliance, redundancy, or avoiding vendor lock-in.

PW SKILLS

Comparing AWS, Azure, and GCP in terms of

Let’s compare

1. **Compute & Pricing Models**

AWS: EC2, Lambda, spot/reserved instances; flexible but complex pricing.

Azure: VMs, Functions, reserved savings; strong enterprise support.

GCP: Compute Engine, Cloud Functions; simple pricing with sustained-use discounts.

Let’s compare

2. **Storage & Databases**

AWS: S3, EBS, RDS, DynamoDB; wide database variety.

Azure: Blob Storage, SQL Database, Cosmos DB; great for global data.

GCP: Cloud Storage, Cloud SQL, Bigtable; strong in analytics and scalability.

Let’s compare

3. Networking & Security

AWS: VPC, IAM, KMS; mature and customizable.

Azure: Virtual Network, Active Directory, Key Vault; tight Microsoft integration.

GCP: Global VPC, IAM, Cloud KMS; strong data security and privacy.

PW SKILLS

Discussing why companies choose different cloud providers based on business needs.

Let’s discuss

Companies choose cloud providers based on their specific business needs:

- **AWS:** Chosen for its wide range of services, flexibility, and global reach, making it ideal for businesses needing scalability and control.
- **Azure:** Preferred by enterprises with a Microsoft-centric environment, or those needing hybrid cloud solutions and strong compliance features.
- **GCP:** Ideal for data-driven companies focused on big data, machine learning, and analytics, with competitive pricing and strong support for open-source technologies.

Take A 5-Minute Break!

- Stretch and relax
- Hydrate
- Clear your mind
- Be back in 5 minutes

PW SKILLS

Explaining Availability Zones (AZs) and Regions and how they ensure uptime.

Let’s see

Regions are geographic areas that contain multiple Availability Zones (AZs), which are isolated data centers within each region.

- Regions are independent of each other, ensuring that failures in one region don’t affect others.
- Availability Zones (AZs) are designed to be fault-tolerant, with separate power, cooling, and networking. If one AZ fails, traffic can be routed to another AZ within the same region for high availability.

Pop Quiz

Q. Which of the following best describes the relationship between Regions and AZs?

A

Each Region contains multiple isolated AZs

B

Each Region has one AZ only

Pop Quiz

Q. Which of the following best describes the relationship between Regions and AZs?

A

Each Region contains multiple isolated AZs

B

Each Region has one AZ only

PW SKILLS

Discussing scalability strategies

Let’s discuss

- **Auto Scaling Groups (ASG):** Automatically adjusts the number of compute instances (e.g., EC2) based on traffic demand, ensuring elasticity and cost-efficiency.
- **Load Balancing (ALB, NLB):** Application Load Balancer (ALB) handles HTTP/HTTPS traffic, while Network Load Balancer (NLB) handles high-throughput, low-latency traffic, ensuring even traffic distribution across instances.
- **Multi-region Deployments:** Distribute workloads across multiple regions for global availability, ensuring low latency and high fault tolerance in case of regional failures.

PW SKILLS

Discussing the advantages of multi-region deployment

Let’s discuss

1. **Reduced Latency**
    - By deploying applications in regions closer to users, data travels a shorter distance, resulting in faster response times and a better user experience globally.
2. **Improved Disaster Recovery & Failover**
    - If one region experiences a failure, traffic can automatically shift to another, ensuring high availability, business continuity, and minimal downtime.

PW SKILLS

Explaining challenges and solutions in multi-region deployment

Let’s see

1. **Data Replication & Consistency Issues**
    - **Challenge:** Keeping data in sync across regions can lead to delays or conflicts, especially in real-time applications.
    - **Solution:** Use globally distributed databases (e.g., DynamoDB Global Tables, Cloud Spanner) and design for eventual consistency where strong consistency isn’t critical.

Let’s see

2. **Network Latency & Cost Considerations**

   - **Challenge:** Cross-region data transfer increases latency and can lead to high costs.
   - **Solution:** Minimize inter-region communication, use caching (e.g., CloudFront), and route users to the nearest region with geo-based DNS (e.g., Route 53).

PW SKILLS

Demonstrating how to map AWS services to application needs

Let’s do it

Compute

- **EC2:** Best for full control over servers and custom environments.
- **Lambda:** Ideal for event-driven, short-duration tasks (serverless).
- **Kubernetes (EKS):** Use for containerized apps needing orchestration and scaling.

Let’s do it

Storage

- **S3:** For object storage, backups, and static content (e.g., images, logs).
- **EFS:** For shared file systems across multiple EC2 instances.
- **RDS:** For relational databases like MySQL, PostgreSQL, with managed scaling and backups.

Let’s do it

Networking

- **VPC:** Core for isolated networking, subnets, and security control.
- **VPN:** Connects on-premises networks to AWS over the internet.
- **Direct Connect:** For dedicated, high-bandwidth, low-latency links to AWS.

PW SKILLS

Explaining how different cloud services impact cost and how to optimize pricing.

Let’s see

Cost Impact by Service

- **Compute (EC2, Lambda):** Costs depend on instance size, runtime, and hours used.
- **Storage (S3, EBS, RDS):** Charged by GB stored, IOPS, and data retrieval.
- **Networking:** Data transfer between regions or to the internet adds cost.

Let’s see

Optimization Strategies

- **Right-size resources:** Use appropriate instance types and scale with Auto Scaling.
- **Use pricing models:** Choose Reserved Instances, Savings Plans, or Spot Instances where applicable.
- **Storage tiers:** Use S3 Intelligent-Tiering or Glacier for infrequently accessed data.
- **Monitor usage:** Use Cost Explorer, Budgets, and Trusted Advisor to track and reduce waste.

PW SKILLS

Explaining how production logs, monitoring tools, and alerts help DevOps teams.

Let’s see

Production logs, monitoring tools, and alerts help DevOps teams by:

- Detecting issues early through real-time metrics and log data.
- Diagnosing problems quickly using detailed logs for root cause analysis.
- Responding faster with alerts that notify teams of critical events.
- Ensuring system health by tracking performance and availability.

PW SKILLS

Discussing observability strategies

Let’s do it

Observability strategies help ensure application reliability by:

- **Monitoring traffic & performance:** Track user behavior, latency, error rates, and resource usage.
- **Using tools like CloudWatch, Prometheus, Datadog:** Gain real-time insights into system health and performance.
- **Automating alerts:** Detect anomalies (e.g., high error rates, latency spikes, failures) and notify teams for quick response.

PW SKILLS

Time for case study!

Important

- Complete the post-class assessment
- Complete assignments (if any)
- Practice the concepts and techniques taught in this sessio
- Review your lecture notes
- Note down questions and queries regarding this session ar
  consult the teaching assistants

Thanks !