SKILLS | DevOps and Cloud Computing

Cloud & AWS Fundamentals

Objective

- Understand cloud security and compliance in public, private, and hybrid environments.
- Learn about cloud deployment models (IaC, containers, serverless) and cloud service models (IaaS, PaaS, SaaS).
- Explore AWS global infrastructure, including regions and availability zones.
- Get an overview of AWS compute, storage, networking, and database services.
- Identify when to launch AWS services via console, CLI, or SDKs based on requirements.
- Explore AWS management tools for resource provisioning and monitoring.

PW SKILLS

Explaining cloud security principles

Let’s see

1. **Shared Responsibility Model**
    - AWS secures the cloud infrastructure (hardware, software, networking).
    - Customer secures what they put in the cloud (data, applications, access controls).
2. **Identity and Access Management (IAM)**
    - Controls who can access AWS resources and what they can do.
    - Use users, roles, groups, and policies to grant least-privilege access.
    - Enable MFA and avoid hardcoded credentials.
3. **Encryption**
    - Data at rest: Encrypt stored data using AWS services like S3, RDS, and KMS.
    - Data in transit: Use TLS/SSL (HTTPS) to encrypt data moving between systems.

PW SKILLS

Discussing compliance certifications (ISO, SOC, GDPR, HIPAA) for cloud security.

Let’s discuss

1. **ISO**
    - ISO/IEC 27001: Global standard for information security.
    - ISO 27017/27018: Focus on cloud security and data privacy.
    - Ensures strong security management practices.
2. **SOC**
    - SOC 1: Financial controls.
    - SOC 2: Security, availability, confidentiality, privacy.
    - SOC 3: Public summary of SOC 2.
    - Validates cloud provider's internal security controls.

Let’s discuss

3. **GDPR**
    - EU law for protecting personal data and privacy.
    - Requires user consent, data control, and breach notification.
    - Cloud providers must support data protection for EU citizens.
4. **HIPAA**
    - U.S. law protecting health information (PHI).
    - Requires strict data security and privacy controls.
    - Applies to healthcare data stored or processed in the cloud.

PW SKILLS

Explaining different cloud deployment models

Let’s see

1. Infrastructure as Code (IaC)

- Automates infrastructure setup using code.
- Tools: Terraform, AWS CloudFormation.
- Enables consistent, repeatable, and version-controlled deployments.

2. Containers

- Package apps and dependencies into lightweight, portable units.
- Tools: Docker, Amazon ECS, Amazon EKS (Kubernetes).
- Simplifies scaling, portability, and microservices architecture.

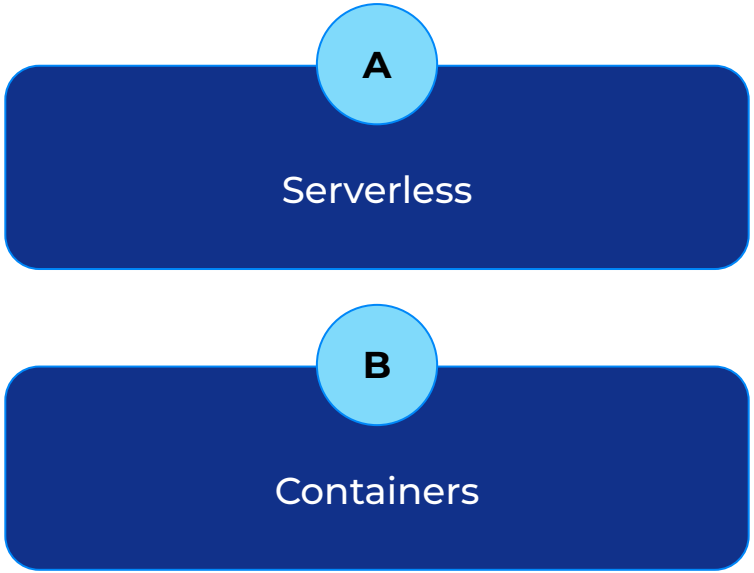
Let’s see

3. Serverless

- Run code without provisioning or managing servers.
- Tools: AWS Lambda, AWS Fargate.
- Auto-scales, pay-per-use, great for event-driven workloads.

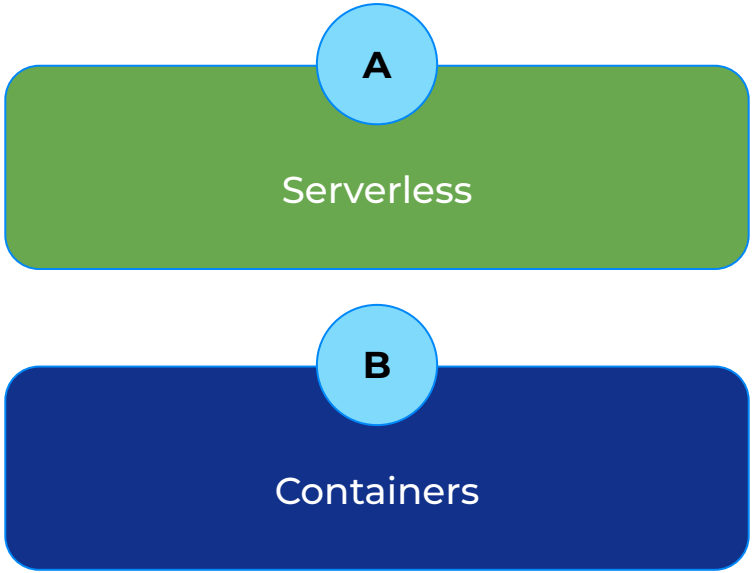
Pop Quiz

Q. AWS Lambda and AWS Fargate are examples of which cloud deployment model?



Pop Quiz

Q. AWS Lambda and AWS Fargate are examples of which cloud deployment model?



PW SKILLS

Discussing use cases and when to use each deployment model.

Let’s discuss

1. Infrastructure as Code (IaC)

- Use Cases: Automate cloud setup, manage environments, CI/CD pipelines.
- When to Use: For consistent, repeatable, scalable infrastructure.
- Tools: Terraform, AWS CloudFormation

2. Containers

- Use Cases: Deploy microservices, scalable apps, portable environments.
- When to Use: When you need flexibility, fast scaling, or app isolation.
- Tools: Docker, Amazon ECS, EKS (Kubernetes)

Let’s discuss

3. Serverless

- Use Cases: Event-driven tasks, APIs, automation, lightweight backends.
- When to Use: For minimal ops, auto-scaling, and cost-effective workloads.
- Tools: AWS Lambda, AWS Fargate

PW SKILLS

Explaining the differences between IaaS, PaaS, and SaaS

Let’s see

1. IaaS (Infrastructure as a Service):

- Provides virtual servers, storage, and networking.
- You manage the OS and apps.
- Example: Amazon EC2, Azure VMs
- Best for full control over infrastructure.

Let’s see

2. PaaS (Platform as a Service):

- Offers a managed environment for app development.
- You focus on code; the provider manages infrastructure.
- Example: AWS Elastic Beanstalk, Google App Engine
- Best for fast app development without managing servers.

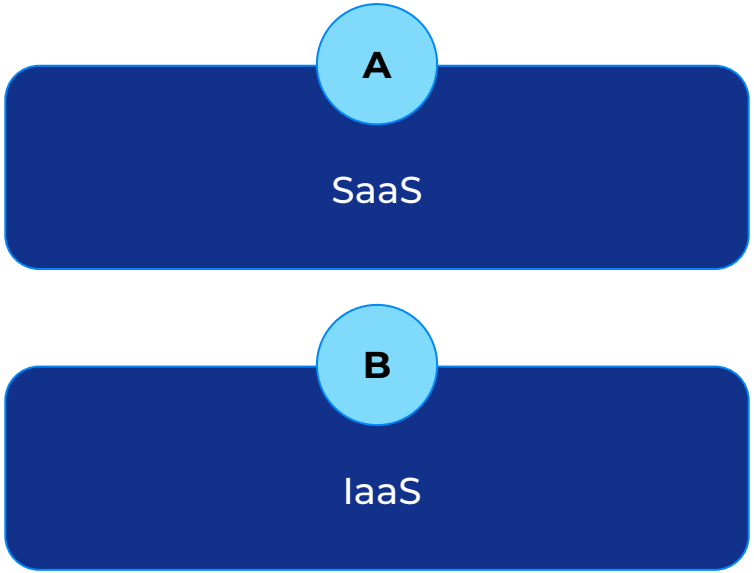
Let’s see

3. SaaS (Software as a Service)

- Fully managed, ready-to-use software over the internet.
- No infrastructure or app management needed.
- Example: Google Workspace, Salesforce
- Best for end-users needing immediate access to apps.

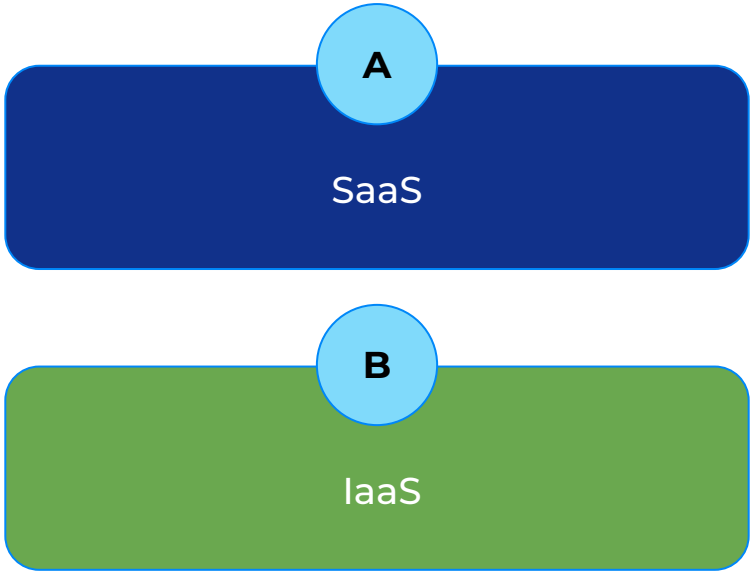
Pop Quiz

Q. Which cloud model provides virtual machines, storage, and networking resources?



Pop Quiz

Q. Which cloud model provides virtual machines, storage, and networking resources?



PW SKILLS

Discussing real-world examples and benefits of each model

Let’s discuss

1. IaaS (Infrastructure as a Service)

Examples: AWS EC2, Google Compute Engine, Microsoft Azure VMs

Benefits:

- Full control over servers and OS
- Flexible, scalable infrastructure
- Pay-as-you-go pricing reduces upfront costs

Let’s discuss

2. PaaS (Platform as a Service)

Examples: AWS Elastic Beanstalk, Google App Engine, Heroku

Benefits:

- Speeds up app development by handling infrastructure
- Automatic scaling and load balancing
- Simplifies management of runtime and middleware

Let’s discuss

3. SaaS (Software as a Service)

Examples: Gmail, Salesforce, Dropbox

Benefits:

- Accessible anywhere with internet
- No installation or maintenance required
- Subscription model offers predictable costs

PW SKILLS

Explaining the differences between public, private, and hybrid clouds

Let’s see

Public Cloud

- Owned and operated by third-party providers (e.g., AWS, Azure).
- Resources shared among multiple users.
- Cost-effective and highly scalable.

Private Cloud

- Dedicated infrastructure for one organization.
- Greater control, security, and customization.
- Often hosted on-premises or by a third party.

Let’s see

Hybrid Cloud

- Combination of public and private clouds.
- Enables data and apps to move between both.
- Balances flexibility, cost, and security.

Pop Quiz

Q. Which of the following is a real-world example of using hybrid cloud?

A

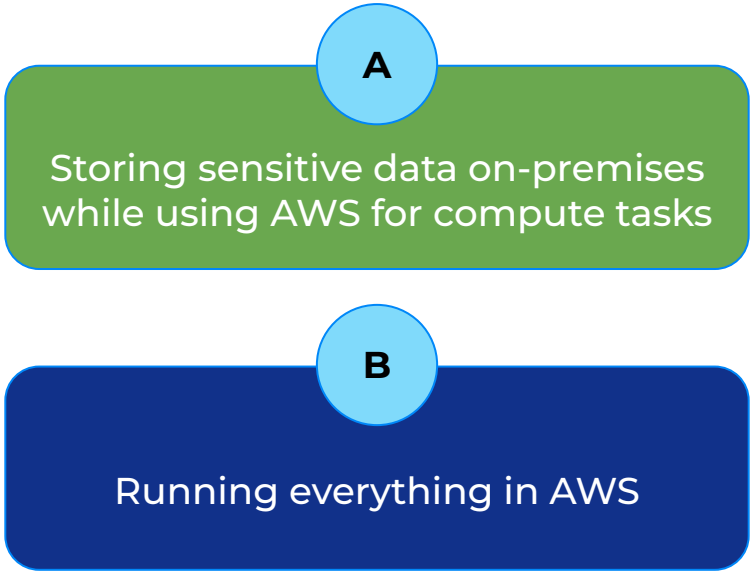
Storing sensitive data on-premises while using AWS for compute tasks

B

Running everything in AWS

Pop Quiz

Q. Which of the following is a real-world example of using hybrid cloud?



PW SKILLS

Discussing when organizations choose hybrid cloud for compliance, security, or data residency needs.

Let’s discuss

Organizations choose hybrid cloud when they need to balance compliance, security, and data residency requirements. For example:

- Keep sensitive data on private cloud or on-premises for strict security and compliance.
- Use public cloud for less sensitive workloads to gain scalability and cost savings.
- Move data or apps between environments to meet regulatory rules or geographic data residency laws.

Take A 5-Minute Break!

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

PW SKILLS

Explaining AWS global infrastructure

Let’s see

Regions

- Geographical areas (e.g., US-East, EU-West).
- Help meet latency and compliance needs by placing resources closer to users or within legal boundaries.

Availability Zones (AZs)

- Isolated data centers within a region.
- Provide redundancy and fault tolerance by allowing apps to run across multiple AZs.

Edge Locations & CDNs

- Global network of sites (via Amazon CloudFront) to cache and deliver content.
- Improve performance and reduce latency for end users.

Pop Quiz

Q. Why is choosing the right AWS Region important?

A

It determines your billing account

B

It affects latency, compliance, and service availability

Pop Quiz

Q. Why is choosing the right AWS Region important?

A

It determines your billing account

B

It affects latency, compliance, and service availability

PW SKILLS

Discussing how to choose the right AWS region based on pricing and performance.

Let’s discuss

- **Performance:** Pick a region closer to your users to reduce latency.
- **Pricing:** Costs vary by region—some (like US-East-1) are cheaper than others.
- **Compliance:** Choose a region that meets data residency or regulatory requirements.
- **Service Availability:** Not all AWS services are available in every region.

PW SKILLS

Providing a high-level overview of core AWS services

AWS services

Compute

- EC2: Scalable virtual servers.
- Lambda: Serverless, event-driven compute.
- Elastic Beanstalk: Easy app deployment platform.
- ECS / EKS: Container orchestration (ECS = AWS native, EKS = Kubernetes).

Storage

- S3: Object storage for any data type.
- EBS: Block storage for EC2 instances.
- Glacier: Low-cost archival storage.
- EFS: Scalable, shared file storage.

AWS services

Networking

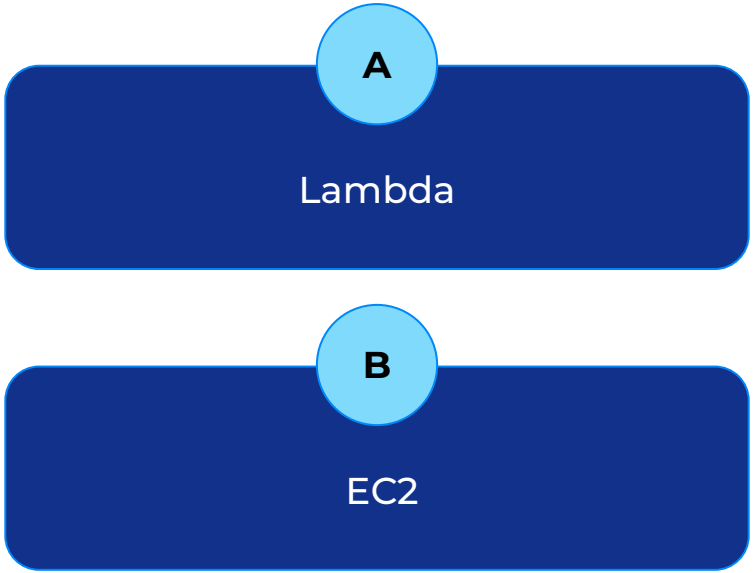
- VPC: Isolated cloud network.
- Route 53: Scalable DNS service.
- Load Balancers: Distribute traffic across resources.
- Direct Connect: Dedicated network link to AWS.

Databases

- RDS: Managed relational databases (e.g., MySQL, PostgreSQL).
- DynamoDB: NoSQL database for fast, scalable apps.
- Redshift: Data warehouse for analytics.

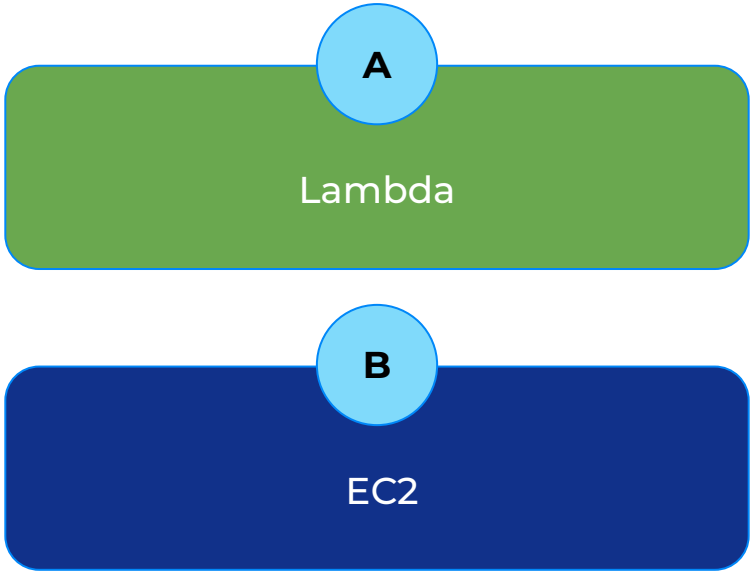
Pop Quiz

Q. Which service allows you to run code without provisioning or managing servers?



Pop Quiz

Q. Which service allows you to run code without provisioning or managing servers?



PW SKILLS

Discussing how different services fit together in cloud architecture.

Let’s discuss

In cloud architecture, AWS services work together to build scalable, secure, and efficient applications:

- Compute (EC2, Lambda) runs your application logic.
- Storage (S3, EBS) stores files, data, and backups.
- Databases (RDS, DynamoDB) handle structured and unstructured data.
- Networking (VPC, Load Balancers) connects and protects resources.
- Monitoring & Security (CloudWatch, IAM) ensure performance and control access.

PW SKILLS

Explaining different ways to deploy AWS services

Let’s see

AWS Management Console:

Web-based GUI for easy, manual setup and monitoring.

AWS CLI:

Command-line tool for automating tasks and scripting deployments.

AWS SDKs:

Programmatic access to AWS from apps using languages like Python, Java, or Node.js.

PW SKILLS

Discussing when to use console, CLI, or SDKs based on automation and scalability needs.

Let’s discuss

- **Console:**

Best for one-time tasks, learning, or manual setup.

✔️ Easy but not scalable.

- **CLI:**

Ideal for automation, scripts, and repeated tasks.

✔️ Scalable and efficient for DevOps.

- **SDKs:**

Use when applications need to interact with AWS programmatically.

✔️ Scalable, flexible, great for custom app logic.

PW SKILLS

Explaining AWS management tools for resource provisioning and monitoring:

Let’s see

1. AWS Management Console

- A web-based GUI that lets users visualize, configure, and manage AWS resources manually.
- Ideal for beginners, one-time setups, or monitoring dashboards.
- Offers point-and-click access to services like EC2, S3, CloudWatch, etc.

Let’s see

2. AWS CLI (Command Line Interface)

- A text-based interface to interact with AWS using commands.
- Useful for automation, scripting, and managing multiple resources efficiently.
- Supports repeatable tasks and integrates well with CI/CD pipelines.

Let's see

3. AWS SDKs (Software Development Kits)

- Enable programmatic access to AWS services from applications written in languages like Python, Java, or JavaScript.
- Ideal for developers to embed AWS functionality (e.g., uploading to S3, invoking Lambda) directly into applications.
- Useful for building scalable, cloud-integrated apps.

PW SKILLS

Discussing AWS CloudFormation and AWS Config for tracking and managing changes.

Let’s discuss

AWS CloudFormation

- Automates resource provisioning using code (Infrastructure as Code).
- Helps track and manage changes through versioned templates.
- Ensures consistent and repeatable deployments.

AWS Config

- Monitors and records AWS resource configurations over time.
- Detects changes, evaluates compliance, and helps with auditing.
- Useful for security, governance, and troubleshooting.

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 !