DEVOPS - DAY 2

1. Scaling Types in Cloud Infrastructure

- **Horizontal Scaling** (Scaling Out/In)
  - What it is: Adding more machines/instances to handle increased traffic.
  - Example: If one server can't handle 1 million users, you add more servers (like 2, 3, 4…) to balance the load.
  - Used when: You need high availability and scalability.
  - Cloud Benefit: Easy to do using services like AWS EC2 + Auto Scaling + Load Balancer.
- **Vertical Scaling (Scaling Up/Down)**
  - What it is: Increasing the resources (CPU, RAM, Storage) of a single server.
  - Example: Upgrading a t2.medium EC2 instance to a t2.large (more CPU/RAM).
  - Used when: You want more power in one machine without changing architecture.
  - Limitation: There's always a hardware limit to how much you can scale up.

2. Load Balancer (LB)

A load balancer distributes incoming traffic across multiple servers (also called targets) to ensure no single server gets overloaded.

- Why it's useful:
  - Ensures high availability and fault tolerance.
  - If one server goes down, LB reroutes traffic to healthy servers.
  - Can perform health checks to monitor server status.
- In AWS (example):
  - ELB (Elastic Load Balancer) has types:
    - Application Load Balancer (ALB) – for HTTP/HTTPS traffic.
    - Network Load Balancer (NLB) – for TCP/UDP traffic.
    - Gateway Load Balancer (GLB) – used for third-party virtual appliances.

3. Auto Scaling :

Auto Scaling automatically increases or decreases the number of running servers (EC2 instances) based on demand and policies.

Why it's useful:

- Saves money by shutting down unused instances.
- Ensures performance during traffic spikes (e.g., flash sales).

🔹 **Key Features:**

- Works with Load Balancer to distribute new traffic to scaled-up instances.
- Based on **policies**: CPU usage, network traffic, memory, or custom metrics.

4. CloudWatch Alarms

Amazon CloudWatch monitors your cloud infrastructure and **CloudWatch Alarms** notify you when certain conditions are met.

🔹 **Example:**

- If **CPU utilization > 80%**, an alarm triggers.
- Alarm can take actions like:
  - Trigger **Auto Scaling**.
  - Send alerts via **SNS**.
  - Stop, terminate, or reboot EC2 instances.

🔹 **Common Metrics Tracked:**

- CPU utilization, memory, disk usage.
- Custom application-level metrics.

5. SNS (Simple Notification Service)

SNS is a fully managed messaging service used to send **notifications** from AWS services to users or systems.

🔹 **How it works:**

- Set up a **topic** (e.g., CPU_Alerts).
- Subscribe endpoints to the topic (email, SMS, Lambda, HTTP).
- When CloudWatch triggers an alarm, it **publishes a message to the topic**.
- All subscribers get notified instantly.

🔹 **Use Cases:**

- Sending **email alerts** on EC2 instance failures.
- Notifying admins on high traffic events.
- Triggering automated Lambda functions.

How They Work Together

1. **CloudWatch** monitors server metrics.
2. If a metric (like CPU usage) crosses the threshold, it triggers a **CloudWatch Alarm**.
3. The alarm sends a notification via **SNS**.
4. It can also trigger **Auto Scaling** to add/remove EC2 instances.
5. **Load Balancer** manages the traffic to distribute it across scaled instances.

What is Amazon SNS?

Amazon **SNS (Simple Notification Service)** is a **fully managed pub/sub (publish/subscribe)** messaging service that allows systems and users to **send and receive notifications** instantly.

How SNS Works – Step-by-Step

1. Create a Topic

- A **topic** is a logical access point (like a channel) for publishing messages.
- Example: You create a topic called HighCPUAlert.

2. Subscribe to the Topic

- You **subscribe endpoints** to this topic. These endpoints can be:

  Email

  SMS

  AWS Lambda

  HTTP/HTTPS endpoint

  SQS (for queues)
- Example: You subscribe your email kushi@example.com to the topic.

3. Publish a Message to the Topic

- Any service (like **CloudWatch Alarm**) or user **publishes a message** to the topic.
- Example: A CloudWatch alarm detects CPU > 80% and publishes:
  "Alert: CPU usage crossed 80% on instance i-1234567890" to the HighCPUAlert topic.

4. SNS Delivers the Message

- SNS **automatically pushes** the message to all **subscribed endpoints**.

- Example: You instantly get an **email notification** with the alert message.

Example Flow:

CloudWatch Alarm ---> SNS Topic (HighCPUAlert) ---> Your Email / SMS / Lambda

SNS Topic Types

- **Standard Topics**: Best for most cases. Messages are delivered at least once, possibly out of order.
- **FIFO Topics (First-In-First-Out)**: Used when message order and exactly-once delivery are important.

Security in SNS

- You can use **access policies** to control who can publish or subscribe.
- **Encryption** (using AWS KMS) is supported for secure message delivery.

Real-life Use Cases

Use Case	How SNS Helps
Server Monitoring	Sends alerts via email/SMS when a server fails
Application Workflow	Triggers Lambda functions when an event occurs
User Notifications	Sends updates to users about order status, etc.
Distributed Systems	Keeps different services informed via messages

Example SNS Email Alert Setup with CloudWatch:

1. Create SNS topic: CPU_Alerts
2. Subscribe your email to it.
3. In CloudWatch:
    - Create alarm → choose metric (like CPU Utilization)
    - Set threshold → ex: > 80%
    - Set action: **Send message to CPU_Alerts SNS Topic**
4. Confirm the subscription from your email.

5. Get alerts when your server gets overloaded!