

# CloudWatch & SNS-

## What is CloudWatch?

Amazon CloudWatch is a monitoring and observability service designed to help track the performance and operational health of your AWS resources and applications. It automatically collects metrics such as CPU usage, memory, disk activity, network traffic, and application logs.

## Why Do We Use CloudWatch?

Feature	Description
Performance Monitoring	Monitor services like EC2, RDS, Lambda, etc., to ensure they run smoothly.
Alarms & Alerts	Set thresholds to trigger alerts when performance deviates from expected norms.
Dashboards	Visualize system metrics in real time through custom dashboards.
Logs	Collect, store, and analyze logs to identify and troubleshoot issues.
Automation	Respond automatically using Lambda, Auto Scaling, or other AWS services.

## Types of Monitoring in CloudWatch

Type	Description	Frequency
Basic Monitoring	Default, fewer metrics	Every 5 minutes
Detailed Monitoring	More metrics, better granularity	Every 1 minute (extra cost)

# What is Amazon SNS (Simple Notification Service)?

## Simple Definition:

Amazon SNS is a fully managed messaging service for sending notifications from one system to many subscribers (applications, users, or other services).

## How It Works:

- 1. **Publishers** send messages to an **SNS topic**.
- 2. **Subscribers** (like Email, SMS, Lambda, SQS, etc.) receive those messages.

## Why Do We Use SNS?

Feature	Description
Message Broadcasting	Send one message to multiple subscribers simultaneously.
Multiple Protocols	Supports delivery via Email, SMS, HTTP/S, AWS Lambda, or SQS.
System Decoupling	Allows services to communicate asynchronously without tight coupling.
Event-driven Design	Ideal for triggering actions in serverless or event-driven architectures.
Scalable	Handles millions of messages automatically without manual intervention.

## Types of SNS Topics

### 1. Standard Topic

#### Definition:

High-speed, best-effort messaging for applications where speed is more important than guaranteed order or duplication control.

Feature	Description
High Throughput	Designed for high-volume, fast delivery.

<b>Possible Duplicates</b>	Messages may be delivered more than once.
<b>No Guaranteed Order</b>	Messages might arrive out of sequence.
<b>Common Use Cases</b>	Push notifications, alerts, data processing.

## 2. FIFO Topic (First-In-First-Out)

### Definition:

Ensures messages are delivered exactly once and in the exact order they were sent. Suitable for high-integrity and ordered systems.

Feature	Description
<b>Ordered Delivery</b>	Maintains strict message order.
<b>No Duplicates</b>	Each message is delivered only once.
<b>Lower Throughput</b>	Slightly slower than Standard due to guarantees.
<b>Common Use Cases</b>	Banking, order systems, ticketing platforms.

## CloudWatch + SNS Integration: Real-World Example

### Scenario:

You are running an EC2 instance and want to be notified if CPU usage exceeds 80%.

### Steps:

1. **CloudWatch** monitors the CPU metric.
2. You create a **CloudWatch Alarm** with a threshold of 80%.
3. The alarm is configured to trigger an **SNS topic**.
4. SNS sends a notification via **Email**, **SMS**, or another supported protocol.

### Result:

You're alerted instantly when performance issues occur, enabling quick response and minimal downtime.

## Final Summary Table

Topic	Explanation
<b>Amazon CloudWatch</b>	Monitors AWS resources (CPU, memory, logs, metrics).
<b>Basic Monitoring</b>	Collects metrics every 5 minutes.
<b>Detailed Monitoring</b>	Collects metrics every 1 minute (requires extra cost).
<b>Amazon SNS</b>	Sends automated messages or alerts to subscribers.
<b>Standard Topic</b>	Fast, high-throughput, allows duplicates and unordered messages.
<b>FIFO Topic</b>	Ordered, exactly-once delivery, lower throughput.

## Full EC2 Monitoring & CloudWatch Alarm Setup with SNS Notification

### Step 1: Create an EC2 Instance

1. Go to **AWS Console > EC2**.
2. Click **Launch Instance**.
3. Enter a name like MyEC2.
4. Select **Amazon Linux 2 AMI**.
5. Choose **t2.micro** (Free Tier eligible).
6. Keep other settings default.
7. Select/create a key pair.
8. Click **Launch Instance**.

### Step 2: View Monitoring for the Instance

1. Go to **EC2 > Instances**.
2. Click on your **Instance ID**.

3. Scroll down to the **Monitoring tab** to see CPU, network, etc.

### **Step 3: Enable Detailed Monitoring (Optional but Recommended)**

1. On the instance page, click **Actions > Monitor and troubleshoot > Manage detailed monitoring**.
2. Enable it.

✅ **Why?** Basic monitoring gives data every **5 minutes**, but **detailed monitoring gives data every 1 minute**, which is better for real-time alerts.

### **Step 4: Go to CloudWatch Dashboard**

1. Go to **CloudWatch** from the AWS Console.
2. Click on **Alarms > All alarms**.
3. Click **Create alarm**.

### **Step 5: Select Metric – CPU Utilization**

1. Click **Select metric**.
2. Click **Browse > EC2 > Per-Instance Metrics**.
3. **Paste your instance ID** in the search bar.
4. Select the **CPUUtilization** checkbox for your instance.
5. Click **Select metric**.

### **Step 6: Configure Alarm**


- Set threshold: **CPUUtilization > 80%**
- Set evaluation period: 1 minute for 2 periods (this means 2 minutes of high usage will trigger alarm).
- You'll see:

- Instance Name
- ID
- Metric
- State
- Time Period
- Condition

## Step 7: Setup SNS Notification

1. Under **Notification**, click **Create new topic**.
2. Name it something like HighCPUALerts.
3. Add your **email address**.
4. Create topic and subscription.
5. Go to your email inbox and **confirm the subscription** by clicking the link.

## Step 8: Review and Create Alarm

1. Review everything.
2. Click **Create alarm**.
3.  Done! You now have a working alarm.

## Step 9: Stress the Server to Trigger the Alarm

1. **Connect to your EC2 instance** via SSH.
2. Run the following commands to install and use stress:

```
sudo yum install stress -y
stress --cpu 8 --timeout 60
```

This will use 8 CPUs for 60 seconds, causing high CPU usage.

## Step 10: Watch Alarm Trigger + Email Notification

- Go back to **CloudWatch > Alarms**.
- You'll see the state change from **OK** → **ALARM**.
- You will also get an **email alert** to the address you provided.

## Conclusion

Amazon CloudWatch and Amazon SNS are essential AWS services for building reliable, scalable, and event-driven architectures.

- CloudWatch ensures you have visibility into the health and performance of your infrastructure.
- SNS enables fast, automated communication with users and systems in response to events.

When combined, they form a powerful monitoring and notification pipeline — ideal for real-time alerting, automation, and system resilience.

Let me know if you want this turned into a PDF, cheat sheet, or diagram!