CLOUDWATCH ALARM — SIMPLE STEP-BY-STEP (UNIVERSAL)

Architecture (remember this)

```
Metric → Alarm → SNS → Notification
```

PART A — Create an alarm (generic steps)

Step 1 — Decide WHAT you want to alert on

Examples:

- Root login
- CPU high
- Any CloudTrail activity
- EC2 stopped
- S3 deleted

👉 This decides **which metric** you use.

Step 2 — Make sure a metric exists

Service: **Amazon CloudWatch**

Metrics can come from:

- Built-in AWS metrics (CPU, memory*)
- Log metric filters (CloudTrail, app logs)
- Custom metrics

❗ No metric = no alarm

Step 3 — Create the alarm

In CloudWatch:

1. Go to **Alarms → Create alarm**
2. Select the **metric**
3. Set condition, for example:

```
>= 1
> 80
< threshold
```

4. Set evaluation period (e.g. 5 minutes)

Meaning

When this condition is true → alarm fires.

Step 4 — Choose notification (SNS)

Service: **Amazon SNS**

- Select an SNS topic
- Make sure email subscription is **Confirmed**

Meaning

This is how the alert is delivered.

Step 5 — Create alarm

- Actions enabled ✅
- State will be:
  - `Insufficient data` (normal initially)

🎯 That's it — alarm is live

PART B — Log-based alarms (CloudTrail, app logs)

Use this when the question mentions **logs**.

Log-based alarm flow

```
Log event → Metric filter → Alarm → SNS
```

Step 1 — Logs must exist

Examples:

- CloudTrail → CloudWatch Logs
- App logs → CloudWatch Logs

Step 2 — Create metric filter

Service: **Amazon CloudWatch**

Examples:

Use case	Filter pattern
Any event	`{}`
Root activity	`{ $.userIdentity.type = "Root" }`
Write actions	`{ $.readOnly = false }`

Set:

- Metric value = 1

Meaning

Each matching log = +1 metric

Step 3 — Create alarm on that metric

Same as Part A:

- Condition: >= 1
- Period: 5 minutes
- SNS notification

PART C — General alarm patterns (memorize)

🔐 Security alarms

- Root login
- IAM policy change
- Security group deleted

Metric source: CloudTrail logs

Condition: >= 1

💻 Performance alarms

- CPU > 80%
- Disk full
- Memory high*

Metric source: EC2 metrics

Condition: `> threshold for N minutes`

💰 Cost alarms

- Billing > X dollars

Metric source: Billing metric

Condition: `> budget`

🚨 Availability alarms

- Instance stopped
- Health check failed

Metric source: Status checks

Condition: `!= healthy`

INTERVIEW ONE-LINERS (VERY IMPORTANT)

- **What is a CloudWatch alarm?**

It monitors a metric and triggers an action when a condition is met.

- **How do logs trigger alarms?**

Logs are converted to metrics using metric filters.

- **Why SNS?**

SNS delivers notifications like email or SMS.

- **Why Insufficient data?**

No metric datapoints yet.

MASTER TEMPLATE (say this in interviews)

“I select a metric, define a threshold in CloudWatch, and attach an SNS topic to notify when the alarm condition is met.”

One-line memory hook 🧠

Metric → Alarm → SNS → Alert