How would you use SNS for alerting and monitoring in DevOps?

1. Why SNS for Alerting?
In DevOps, you need to be notified **immediately** when something goes wrong (e.g., high CPU usage, service down, build failed).
SNS is a great fit because it:

- Can **push alerts** to multiple people/systems at once.
- Supports **different protocols** (Email, SMS, HTTP, Lambda, SQS, Slack via webhook).
- Integrates easily with **CloudWatch, CloudTrail, CI/CD pipelines, and third-party tools**.

---

2. Example Use Cases
(a) CloudWatch Alarms → SNS → Email/SMS

- You set a **CloudWatch alarm** (e.g., EC2 CPU > 80% for 5 mins).
- The alarm triggers an **SNS topic**.
- SNS sends notifications to:
  - DevOps team (Email/SMS).
  - Incident management system (e.g., PagerDuty).

👉 This ensures engineers know about the issue **instantly**.

(b) Deployment Notifications in CI/CD

- After every **successful or failed deployment** (via CodePipeline or Jenkins), send a message to an SNS topic.
- Subscribers:
  - Slack webhook (team gets notified).
  - Email group (managers get notified).
  - Lambda function (auto-rollback trigger on failure).

(c) Centralized Monitoring Fan-out

- Suppose you want alerts to go to multiple channels (Ops team email, Security team Slack, Ticketing system like Jira).
- Instead of configuring alerts for each system separately,
  → Configure **one SNS topic** → Add multiple subscribers.
  → Publish alert once, everyone/system gets it.

✅ **Key takeaway:**
SNS is used in DevOps to **broadcast alerts quickly and reliably** to teams and tools → ensuring faster incident response and minimizing downtime.

Why is Amazon SNS a good fit for DevOps alerting and monitoring?
a) Because it stores historical logs of all alerts
b) Because it can push alerts instantly to multiple people and systems across different protocols
c) Because it automatically scales EC2 instances during high CPU load
d) Because it is only limited to sending emails
✅ **Answer:** b) Because it can push alerts instantly to multiple people and systems across different protocols

---

In a DevOps setup, how would you use SNS with CloudWatch alarms?
a) To archive metrics into S3 for analysis
b) To trigger EC2 instance scaling
c) To notify DevOps engineers via Email/SMS when alarms are triggered
d) To run SQL queries on CloudWatch data
✅ **Answer:** c) To notify DevOps engineers via Email/SMS when alarms are triggered

---

Which of the following is a valid use of SNS in a CI/CD pipeline?
a) Broadcasting deployment success/failure to Slack and Email
b) Running database migrations automatically
c) Building and packaging application artifacts
d) Storing container images
✅ **Answer:** a) Broadcasting deployment success/failure to Slack and Email

---

Suppose you want every alert to be sent to Ops email, Security Slack channel, and Jira.
What's the most efficient way to do it using SNS?
a) Create separate CloudWatch alarms for each system
b) Create one SNS topic with multiple subscribers (email, Slack, Jira)
c) Use SQS to queue all alerts and process one by one
d) Manually forward every email to each system
✅ **Answer:** b) Create one SNS topic with multiple subscribers (email, Slack, Jira)
