DevOps Intern Assignment

Objective

The objective of this assignment is to assess your ability to work with Linux/Ubuntu systems and AWS services. You will set up a small cloud-based environment, automate basic tasks, and demonstrate your understanding of system monitoring and cloud integration.

Note: You may use your own AWS account for this task. If you do not already have one, please create a free-tier AWS account at https://aws.amazon.com/free before starting. Ensure that all resources used remain within the AWS Free Tier to avoid charges.

Part 1: Environment Setup

- Launch a t2.micro(Free Tier eligible). Ubuntu EC2 instance on AWS.
- SSH into it and create a new user named devops_intern.
- Grant this user sudo privileges (without a password prompt).
- Change the server's hostname to include your name (e.g., xyz-devops).

Deliverables:

Screenshot showing the hostname, the new user in /etc/passwd, and output of 'sudo whoami' run by devops_intern.

Part 2: Simple Web Service Setup

- Install Nginx or Apache on the EC2 instance.
- Create a simple HTML page at /var/www/html/index.html displaying:
- a Your name
  b The instance ID fetched from AWS metadata
  c The server uptime

Deliverables:

Screenshot of the webpage accessed through the instance's public IP.

Part 3: Monitoring Script

- Create a bash script /usr/local/bin/system_report.sh that prints the following metrics:
- a Current date and time
  b Uptime
  c CPU usage (%)
  d Memory usage (%)
  e Disk usage (%)
  f Top 3 processes by CPU usage
- Configure a cron job to run this script every 5 minutes and append the results to /var/log/system_report.log.

Deliverables:

1 Contents of the cron configuration file used.

2. Screenshot of log file entries after at least two runs.

Part 4: AWS Integration

- Create a CloudWatch log group named /devops/intern-metrics.
- Use the AWS CLI to push the system_report.log file to this CloudWatch log group.

Deliverables:

1. The AWS CLI command(s) used for the upload.
2. Screenshot showing the log data in CloudWatch Logs.

Part 5: Documentation & Cleanup

- Write a README.md file explaining your setup steps, scripts, and how to reproduce the environment.
- Terminate your EC2 instance after completing the task.

Deliverables:

1. README.md file.
2. All scripts and configuration files (e.g., system_report.sh, cron file).

Bonus (Optional)

- Use a systemd service instead of cron for the report script.
- Add an email alert (using AWS SES or mail command) when disk usage > 80%.

Final Submission

Duration: 2 days to submit

Submit a GitHub repository link or a ZIP file containing the following items:

- README.md
- All scripts and configuration files
- All screenshots are clearly labeled by task part

Evaluation Criteria

Area	Weight	What We're Looking For
Linux Fundamentals	25%	User management, cron, permissions, file system, commands
AWS Usage	25%	EC2 setup, metadata, CloudWatch integration
Automation	25%	Scripting, cron scheduling, CLI usage
Documentation	15%	Clear, reproducible instructions
Creativity / Bonus	10%	Error handling, extensions, and good coding practices