Day 4 – Automation & Monitoring

Introduction

On Day 4, the focus is on automating cloud tasks and monitoring system performance using AWS CloudWatch and AWS Command Line Interface (CLI). These tools help administrators manage cloud resources efficiently, monitor applications in real time, and automate routine tasks.

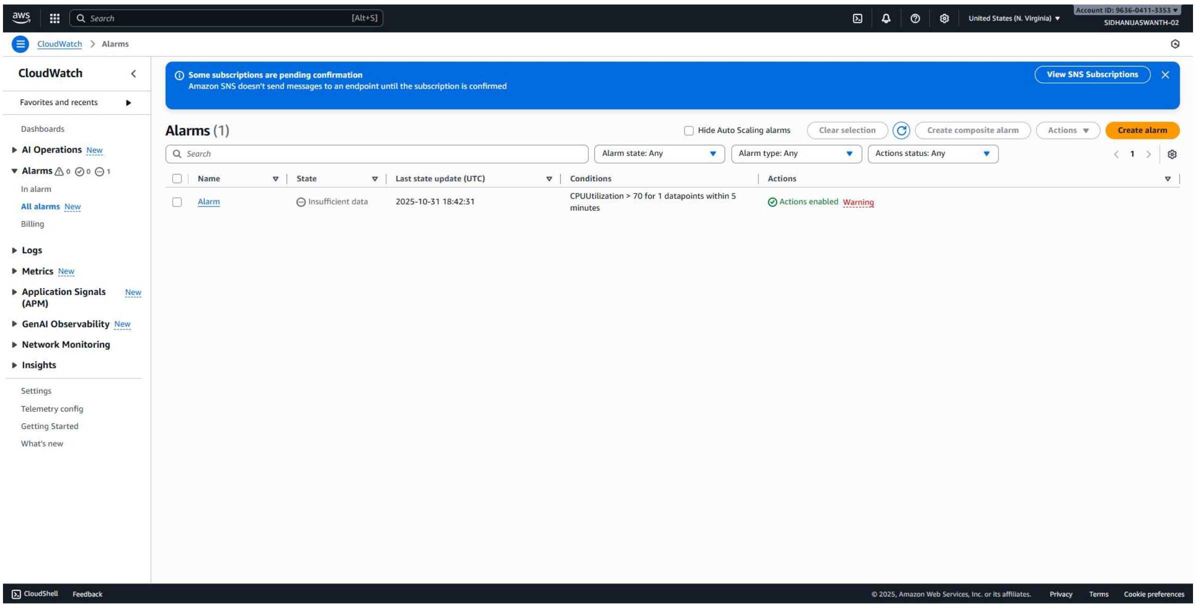
1. AWS CloudWatch (Monitoring)

Amazon CloudWatch is a monitoring and observability service for AWS cloud resources. It collects metrics, logs, and events, providing visibility into the performance and operational health of applications. CloudWatch can trigger alarms, automate responses, and help ensure that applications run smoothly.

Hands-On Steps:

- 1. Open the AWS Management Console and go to CloudWatch.
- 2. From the navigation pane, select 'Alarms' and click 'Create Alarm'.
- 3. Choose 'EC2' as the service and select the metric 'CPUUtilization'.
- 4. Set a threshold (e.g., alarm triggers when CPU exceeds 70%).
- 5. Configure actions such as email notification via SNS (Simple Notification Service).
- 6. Review and create the alarm, then monitor it in the dashboard.

Screenshot



Some subscriptions are pending confirmation
Amazon SNS doesn't send messages to an endpoint until the subscription is confirmed

Alarms (1)

□	Name	State	Last state update (UTC)	Conditions	Actions
□	Alarm	⊖ Insufficient data	2025-10-31 18:42:31	CPUUtilization > 70 for 1 datapoints within 5 minutes	✓ Actions enabled Warning

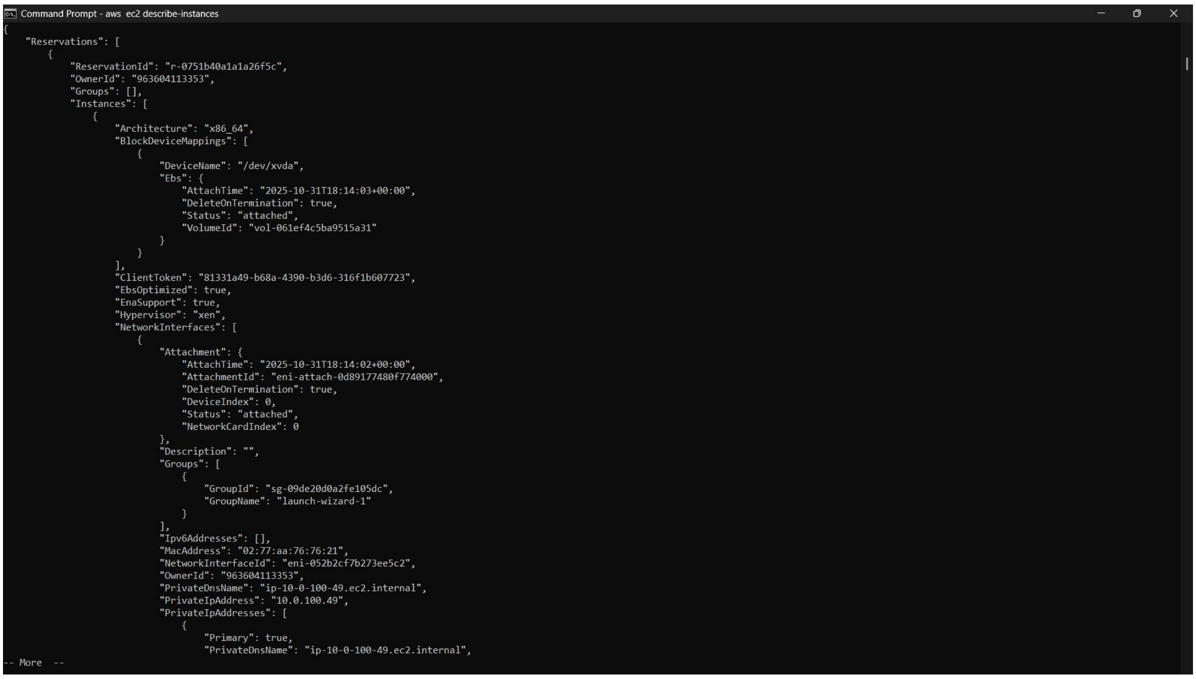
2. AWS Command Line Interface (CLI)

The AWS CLI is a unified command-line tool that allows users to manage AWS services using commands. It enables automation and simplifies repetitive tasks like launching instances, uploading files, and retrieving data without using the AWS Management Console.

Hands-On Steps:

- 1. Install the AWS CLI on your system (Windows, macOS, or Linux).
- 2. Open a terminal or command prompt and run 'aws configure'.
- 3. Enter your AWS Access Key, Secret Key, default region, and output format.
- 4. Run 'aws ec2 describe-instances' to list EC2 instances.
- 5. Run 'aws s3 ls' to list all S3 buckets in your account.
- 6. Verify outputs and ensure correct setup.

Screenshot



```
{
    "Reservations": [
        {
            "ReservationId": "r-0751b40a1a1a26f5c",
            "OwnerId": "963604113353",
            "Groups": [],
            "Instances": [
                {
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/xvda",
                            "Ebs": {
                                "AttachTime": "2025-10-31T18:14:03+00:00",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-061ef4c5ba9515a31"
                            }
                        }
                    ],
                    "ClientToken": "81331a49-b68a-4390-b3d6-316f1b607723",
                    "EbsOptimized": true,
                    "EnaSupport": true,
                    "Hypervisor": "xen",
                    "NetworkInterfaces": [
                        {
                            "Attachment": {
                                "AttachTime": "2025-10-31T18:14:02+00:00",
                                "AttachmentId": "eni-attach-0d89177480f774000",
                                "DeleteOnTermination": true,
                                "DeviceIndex": 0,
                                "Status": "attached",
                                "NetworkCardIndex": 0
                            },
                            "Description": "",
                            "Groups": [
                                {
                                    "GroupId": "sg-09de20d0a2fe105dc",
                                    "GroupName": "launch-wizard-1"
                                }
                            ],
                            "Ipv6Addresses": [],
                            "MacAddress": "02:77:aa:76:76:21",
                            "NetworkInterfaceId": "eni-052b2cf7b273ee5c2",
                            "OwnerId": "963604113353",
                            "PrivateDnsName": "ip-10-0-100-49.ec2.internal",
                            "PrivateIpAddress": "10.0.100.49",
                            "PrivateIpAddresses": [
                                {
                                    "Primary": true,
                                    "PrivateDnsName": "ip-10-0-100-49.ec2.internal",
-- More  --
```

Observations and Findings

Using CloudWatch and CLI together offers powerful control and insight into AWS resources. CloudWatch provides real-time data visualization, while the CLI enables quick automation of complex tasks. Combining both tools enhances operational efficiency, reduces human error, and improves resource management.

Reflection

Day 4 was very insightful as I learned how to automate and monitor AWS resources effectively. Setting up CloudWatch alarms showed how proactive monitoring prevents performance issues. Using the AWS CLI made me realize the power of automation in managing resources quickly. At first, configuring the CLI felt complex, but after setup, executing commands became straightforward. This experience strengthened my understanding of DevOps principles and cloud automation workflows.