

Hands-on with AWS CloudWatch for Monitoring EC2, S3 & Logs

What is CloudWatch?

Amazon CloudWatch is a monitoring and observability service provided by AWS. It allows tracking of **metrics**, collection of **logs**, and setting of **alarms** for AWS resources such as:

- EC2 (Virtual servers)
- EBS (Block storage)
- S3 (Object storage)
- ELB (Load balancer)
- RDS (Databases)

CloudWatch helps maintain visibility into resource utilization, application performance, and operational health.

Note: CloudWatch is a **region-specific** service.

Key Concepts

Metric:

A **metric** is a fundamental unit of measurement (like CPU Utilization, Disk I/O, NetworkIn, etc.) used by CloudWatch to track resource activity.

Alarm States in CloudWatch:

- **OK** – Everything is within the threshold
- **Alarm** – Metric has breached the threshold
- **Insufficient Data** – No data available yet'

Monitoring EC2 Instance State Changes with CloudWatch Events

Steps:

1. Go to **CloudWatch** service from AWS Console.
2. In the left sidebar, click on **Events > Rules**.
3. Click on **Create Rule**.
4. For **Service Name**, choose EC2.
5. Under **Event Type**, select EC2 Instance State-change Notification.
6. Add **SNS Topic** to send notification.
7. Review and create the rule.

Screenshot 1: CloudWatch Rule Creation for EC2 Instance State Change

Monitoring S3 Bucket Events Using CloudWatch

Steps:

1. Navigate to the **S3** service.
2. Select the desired **S3 bucket**.
3. Go to the **Properties** tab.
4. Scroll to **Event Notifications** and click **Create Event Notification**.
5. Choose event types such as:
 - o s3:ObjectCreated:*
 - o s3:ObjectRemoved:*
6. Add an **SNS Topic** to receive notifications.

Screenshot 2: S3 Bucket Event Notification Configuration

Setting up a Web Server and Monitoring Access Logs

Install Apache HTTPD server on EC2:

```
sudo yum install -y httpd
```

```
sudo systemctl start httpd
```

```
sudo chkconfig httpd on
```

```
echo "Hello" > /var/www/html/index.html
```

To verify access logs:

```
cat /etc/httpd/logs/access_log
```

```
[root@ip-172-31-33-195 awslogs]# cat /etc/httpd/logs/access_log
223.185.40.232 - - [09/Jul/2025:12:52:14 +0000] "GET / HTTP/1.1" 200 13 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36"
223.185.40.232 - - [09/Jul/2025:12:52:14 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "http://13.39.83.70/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36"
```

Screenshot 3: HTTPD Installation and Web Page Setup on EC2

Pushing EC2 Logs to CloudWatch Logs

IAM Role for Log Access:

1. Go to **IAM > Roles**.
2. Create a role for EC2 and attach the **CloudWatchLogsFullAccess** policy.
3. Launch a new **EC2 instance** with this role.**Log Agent Setup on EC2:**

1. SSH into the EC2 instance.
2. Install required tools:
3. sudo yum install -y httpd awslogs
4. sudo systemctl start httpd
5. Create sample content:
6. echo "Hello" > /var/www/html/index.html
7. Check Apache logs:
8. cat /etc/httpd/logs/access_log
9. Navigate to:
10. cd /etc/awslogs
11. ls
12. Edit configurations:
 - **awscli.conf** – Add region
 - **awslogs.conf** – Specify the log file to
(e.g., /etc/httpd/logs/access_log)
13. Restart awslogs service:
14. sudo systemctl restart awslogsd
15. Go to **CloudWatch > Log Groups** and confirm log stream creation.

Screenshot 4: CloudWatch Log Stream Display for EC2 Instance Logs

Screenshot 5: CloudWatch Log Group Display Showing EC2 Access Logs

Types of Monitoring in CloudWatch

Monitoring Type	Description
Basic Monitoring	Default, 5-minute data intervals for some AWS resources.
Detailed Monitoring	1-minute data intervals, enabled manually for more insights (incurs additional cost).

Conclusion

In this hands-on implementation, the following key skills and services were applied:

- Configured **CloudWatch Events** for EC2 state-change notifications.
- Monitored **S3 bucket** activity via event notifications.
- Set up **Apache Web Server** and captured real-time logs.
- Configured **awslogs agent** to push server logs to **CloudWatch Logs**.
- Understood the **difference between Basic and Detailed Monitoring**.

This practical exposure improves understanding of AWS observability and operational monitoring using native services.