

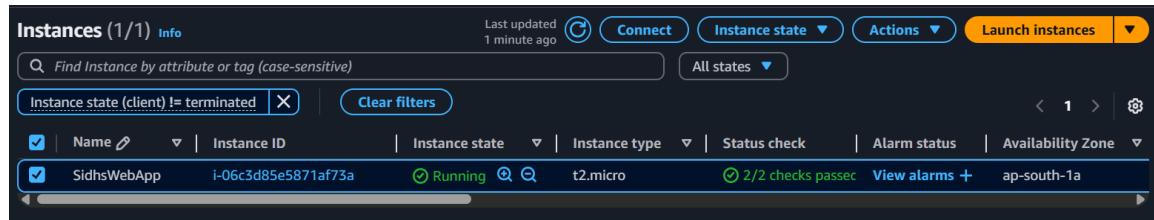
AWS Cloud Monitoring & Alerting Project Documentation

❖ Project Overview

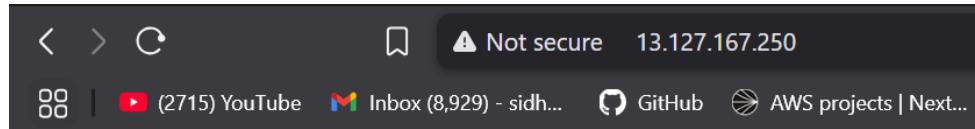
This project demonstrates the setup of a cloud-based web application hosted on an AWS EC2 instance, with monitoring and alerting configured using Amazon CloudWatch.

1. EC2 Web Server Setup

⌚ Screenshot 1: EC2 Instance Running



⌚ Screenshot 2: Web Page Live



Hello from the Cloud!

Screenshot 3: Apache Installed

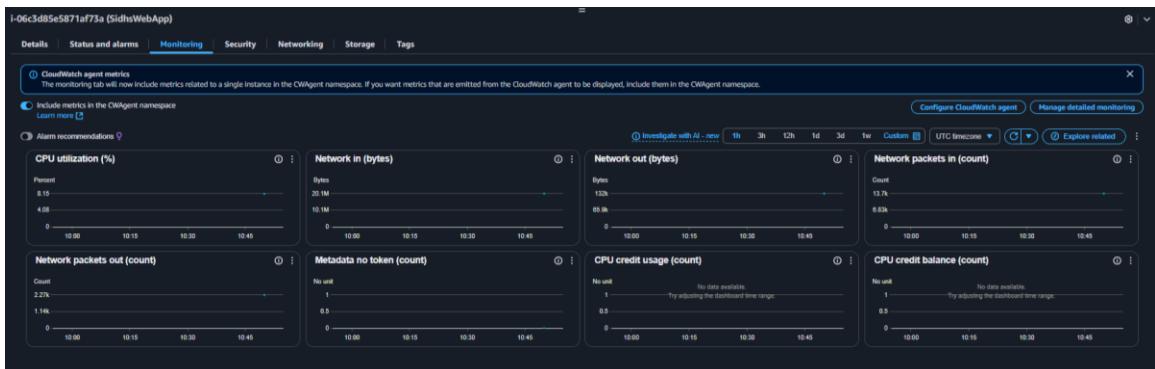
```
[ec2-user@ip-10-0-65-103 ~]$ sudo yum install -y httpd
Last metadata expiration check: 0:00:31 ago on Tue Apr 22 10:54:09 2025.
Dependencies resolved.

=====
| Package           | Architecture | Version      | Repository | Size |
| =====            | ======       | ======       | ======     | ===== |
| Installing:      |             |             |             |        |
| httpd            | x86_64      | 2.4.62-1.amzn2023 | amazonlinux | 48 k  |

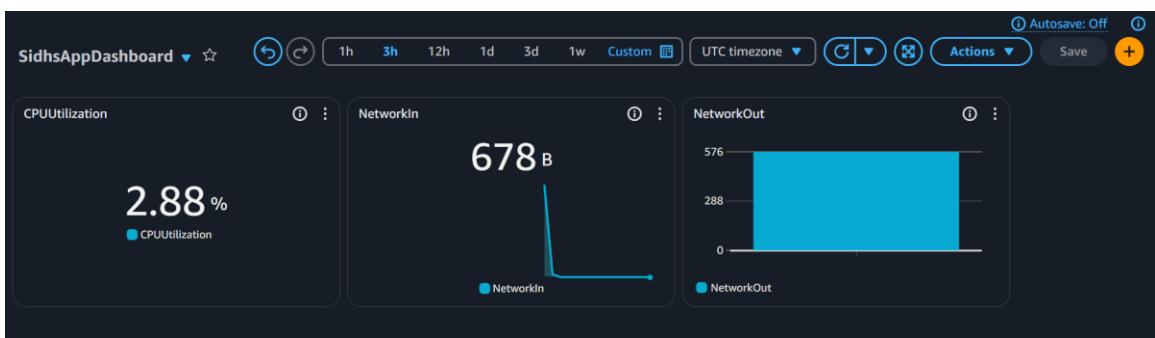
=====
Installed:
  apr-1.7.5-1.amzn2023.0.4.x86_64          apr-util-1.6.3-1.amzn2023.0.1.x86_64    apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-htpd-18.0.0-12.amzn2023.0.3.noarch  httpd-2.4.62-1.amzn2023.x86_64        httpd-core-2.4.62-1.amzn2023.x86_64
  httpd-filesystem-2.4.62-1.amzn2023.noarch   httpd-tools-2.4.62-1.amzn2023.x86_64    libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch       mod_http2-2.0.27-1.amzn2023.0.3.x86_64  mod_lua-2.4.62-1.amzn2023.x86_64
```

2. CloudWatch Monitoring Setup

Screenshot 4: EC2 Monitoring Tab

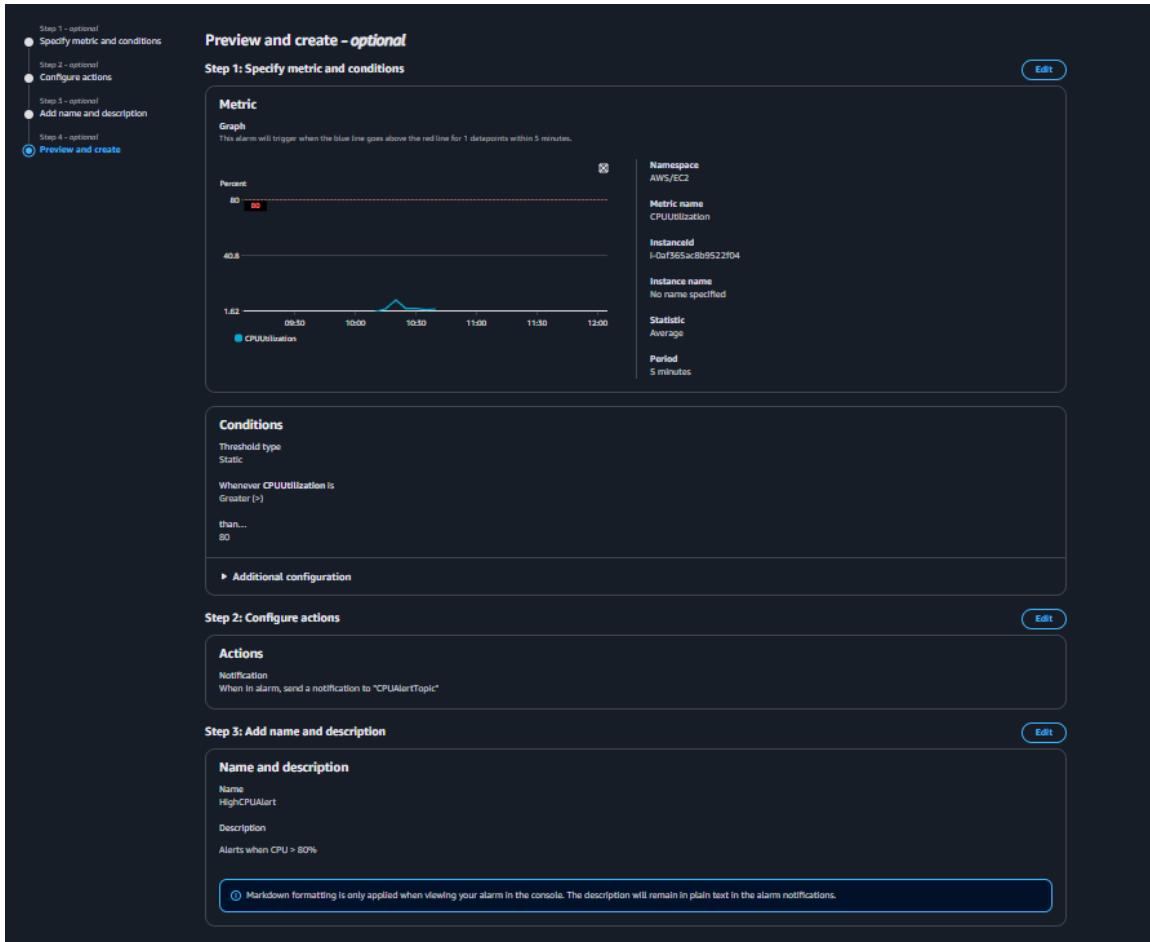


Screenshot 5: CloudWatch Dashboard



3. ⚡ CloudWatch Alarm Configuration

⌚ Screenshot 6: Alarm Setup Summary

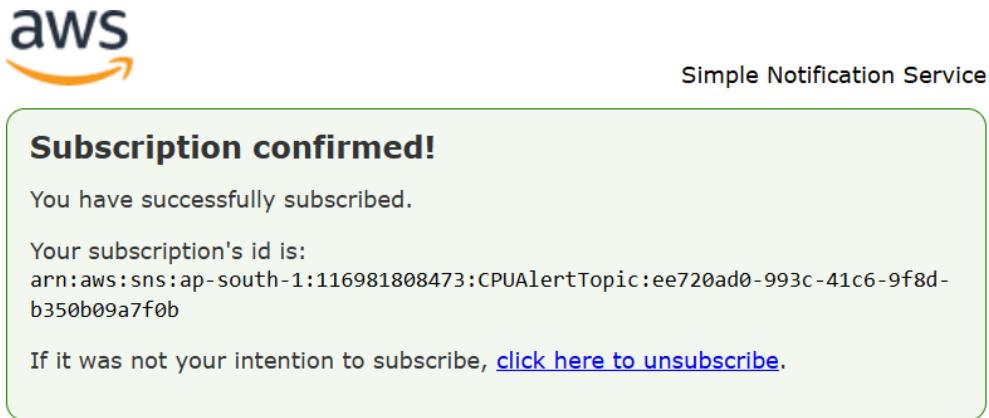


⌚ Screenshot 7: Alarm in Alarm List

Alarms (1/1)		<input type="checkbox"/> Hide Auto Scaling alarms		<input type="checkbox"/> Clear selection	<input type="checkbox"/> Create composite alarm	Actions	<input type="button" value="Create alarm"/>
		Search	Alarm state: Any	Alarm type: Any	Actions status: Any	< 1 >	
<input checked="" type="checkbox"/>	Name	State	Last state update (UTC)	Conditions	Actions		
<input checked="" type="checkbox"/>	HighCPUAlert		2025-04-22 11:57:24	CPUUtilization > 80 for 1 datapoints within 5 minutes			

4. 📩 Notification Setup

📸 Screenshot 8: SNS Email Confirmation



5. 🌐 Networking Setup (Bonus)

📸 Screenshot 9: Security Group Rules

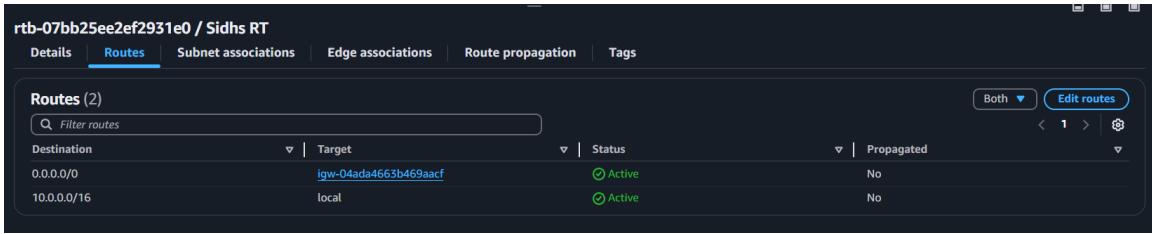
The screenshot displays the Inbound rules section of an AWS security group. It lists two rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-02079afb7cb7e4327	IPv4	SSH	TCP	22	0.0.0.0/0	-
-	sgr-03b7ea1f20a25dcfc	IPv4	HTTP	TCP	80	0.0.0.0/0	-

The screenshot displays the Outbound rules section of an AWS security group. It lists one rule:

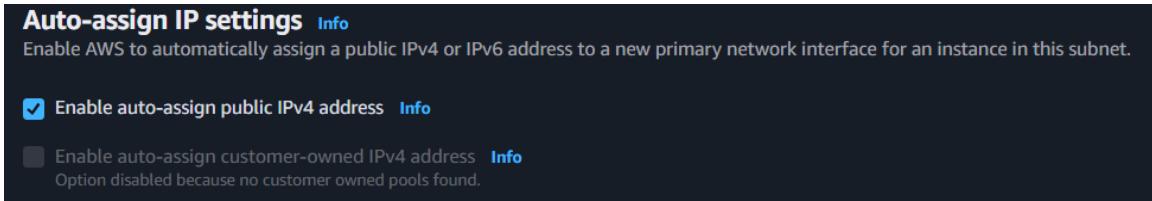
Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
-	sgr-0af654aea3002b5bf	IPv4	All traffic	All	All	0.0.0.0/0	-

Screenshot 10: Route Table with IGW



Routes (2)			
Destination	Target	Status	Propagated
0.0.0.0/0	igw-04ada4663b469aacf	Active	No
10.0.0.0/16	local	Active	No

Screenshot 11: Subnet Public IP Enabled



Auto-assign IP settings [Info](#)
Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.

Enable auto-assign public IPv4 address [Info](#)

Enable auto-assign customer-owned IPv4 address [Info](#)
Option disabled because no customer owned pools found.

Conclusion

This project demonstrates successful deployment and monitoring of a web application on AWS using EC2 and CloudWatch. Real-time monitoring, custom dashboards, and alarm-based notifications were implemented to meet cloud observability requirements.