SCALABLE & SECURE E-COMMERCE PLATFORM ON AWS

FINAL REPORT FOR MIDTERM PROJECT

GROUP - 6

Team Members :

Neonidh Singh - 116009102

Goutham Patchipulusu - 120112669

Muskan Sagar - 120152764

Venkata Umesh Chandra Pothugunta 120427121

TABLE OF CONTENTS

OVERVIEW

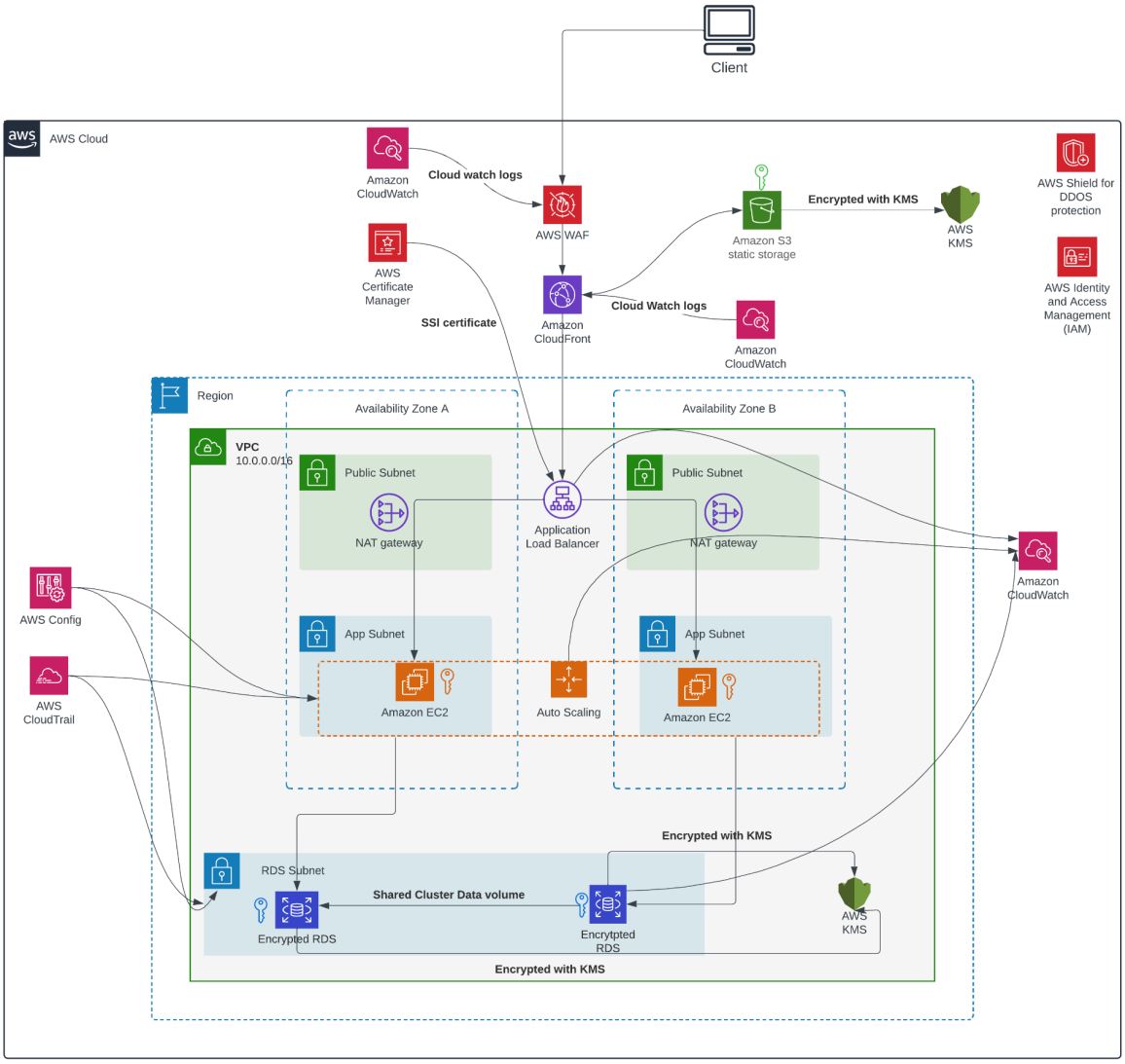
Our group will collaborate on designing, developing, and deploying a scalable e-commerce platform on AWS, focusing on creating a system that can handle fluctuating user traffic, secure customer data, and efficiently deliver content worldwide. To achieve this, we'll use key AWS services, starting with EC2 instances configured with Auto Scaling. This will allow our infrastructure to automatically adjust based on demand, ensuring a responsive platform that doesn't waste resources. Additionally, we'll set up Amazon RDS to securely store customer and product information in a database, ensuring that our data management is both efficient and protected.

Security is a critical component of our project. We'll integrate AWS WAF to defend our application against common cyber threats like SQL injection and cross-site scripting, building a strong layer of protection against potential vulnerabilities. Our team will also configure CloudFront CDN to enhance the speed and efficiency of content delivery, allowing us to cache and serve static assets globally. This setup will ensure that users, no matter their location, experience faster load times and seamless access to the platform, contributing to a better overall experience.

The outcome of our project will be a fully functional, scalable, and secure e-commerce platform hosted on AWS. By the end, we'll have created an infrastructure capable of automatically adjusting to traffic changes, securely storing sensitive data, and delivering fast content globally. Our platform will not only meet current e-commerce standards but will also provide a reliable, user-friendly experience that can scale as needed. This result demonstrates how cloud services can effectively support the growth and security needs of an e-commerce business, equipping us with practical experience in deploying secure, high-performing applications on AWS.

INFRASTRUCTURE SETUP - PHASE 1

Architecture Diagram:



Explanation:

When a user accesses our application, their request first passes through AWS WAF, which acts as the primary layer of defense. The WAF filters out potential threats, such as DDoS attacks or SQL injection attempts. Once the request is validated by WAF, it reaches AWS CloudFront, our Content Delivery Network (CDN) located at edge locations. CloudFront is linked to an S3 bucket that stores images and static files, allowing for faster load times on the Ecommerce platform. The request then flows into the VPC, where the Application Load Balancer (ALB)

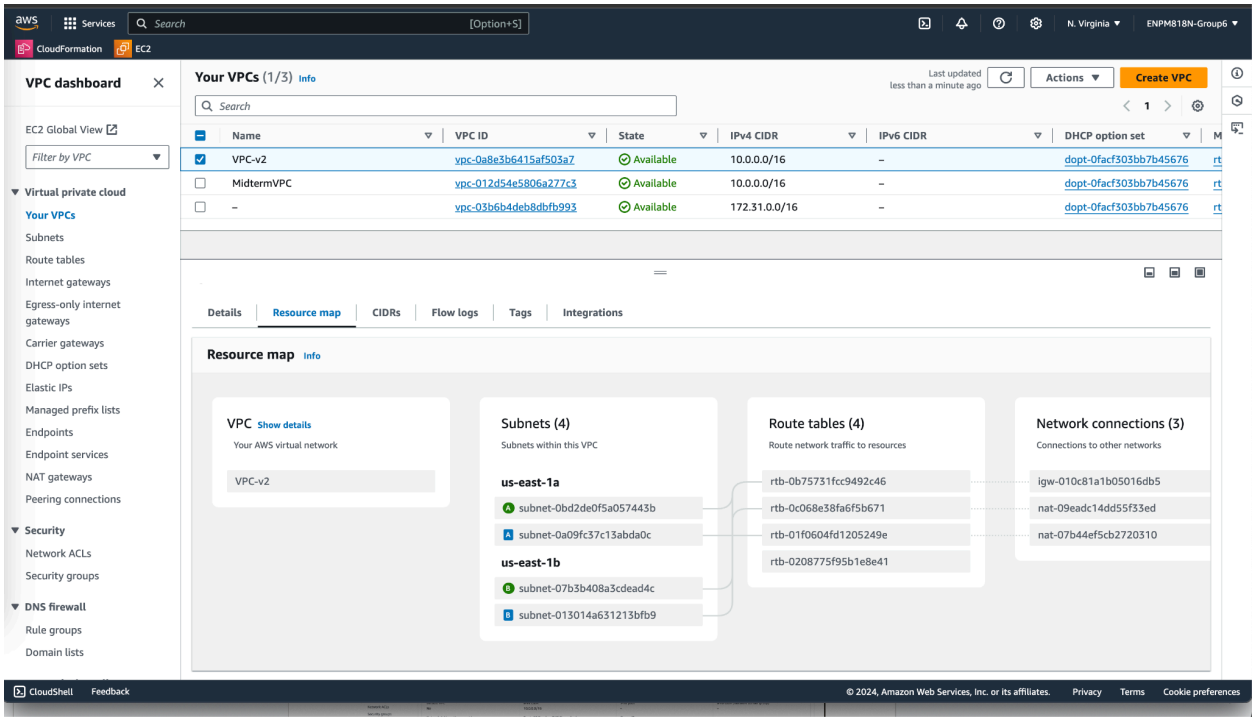
determines the optimal EC2 instance within the private subnets across two availability zones, ensuring traffic is directed based on health and availability checks.

To enable internet connectivity, the public subnets use NAT Gateways, allowing EC2 instances in private subnets to access necessary external resources. Auto Scaling is configured to manage fluctuations in demand automatically, scaling EC2 instances up or down within private subnets as needed. On the backend, the EC2 instances connect to an RDS database housed within a private subnet, which holds the ECommerce platform's data tables. Both database storage and network traffic are encrypted to secure all data exchanges.

Network Components

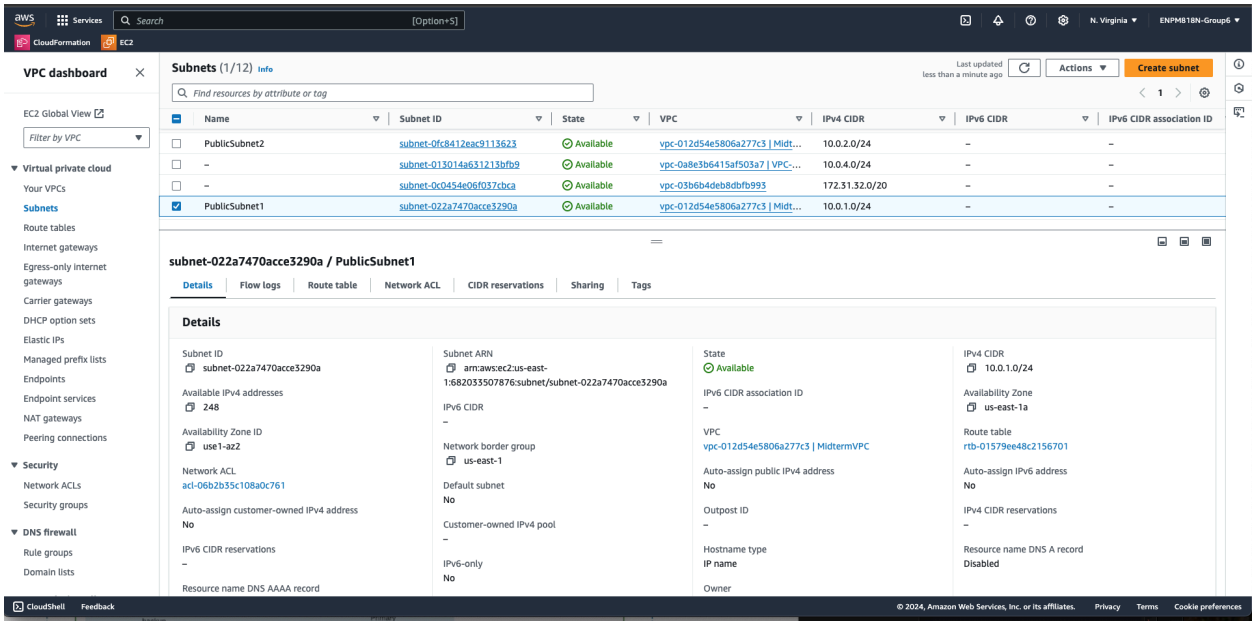
AWS VPC (Virtual Private Cloud):

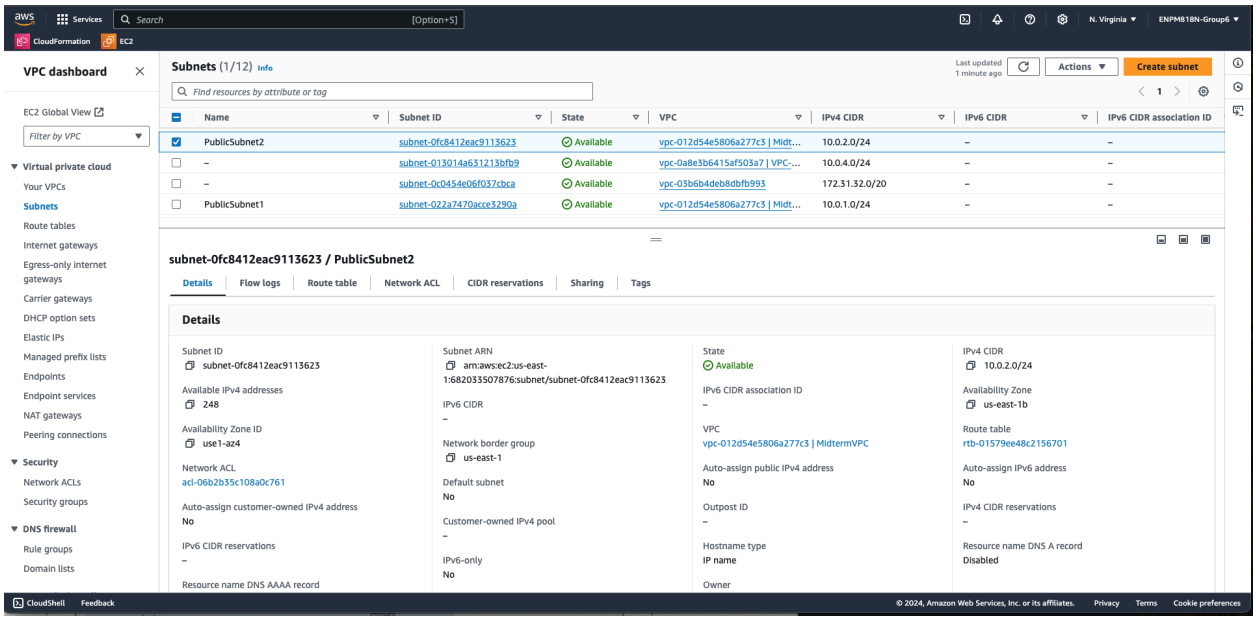
- **Interaction:**
    - This tool creates a private, isolated segment inside the AWS Cloud to allow resources to be securely deployed within a virtual network.
    - Offers flexibility to create IP address ranges, configure subnets, and control traffic access via route tables and gateways.
    - Facilitates the deployment of AWS resources such as Amazon EC2 instances and Amazon RDS databases within a secure, isolated network environment.
- **Our Approach:**
    - We created the VPC. In this template, we defined the CIDR block to be 10.0.0.0/16.



Subnets:

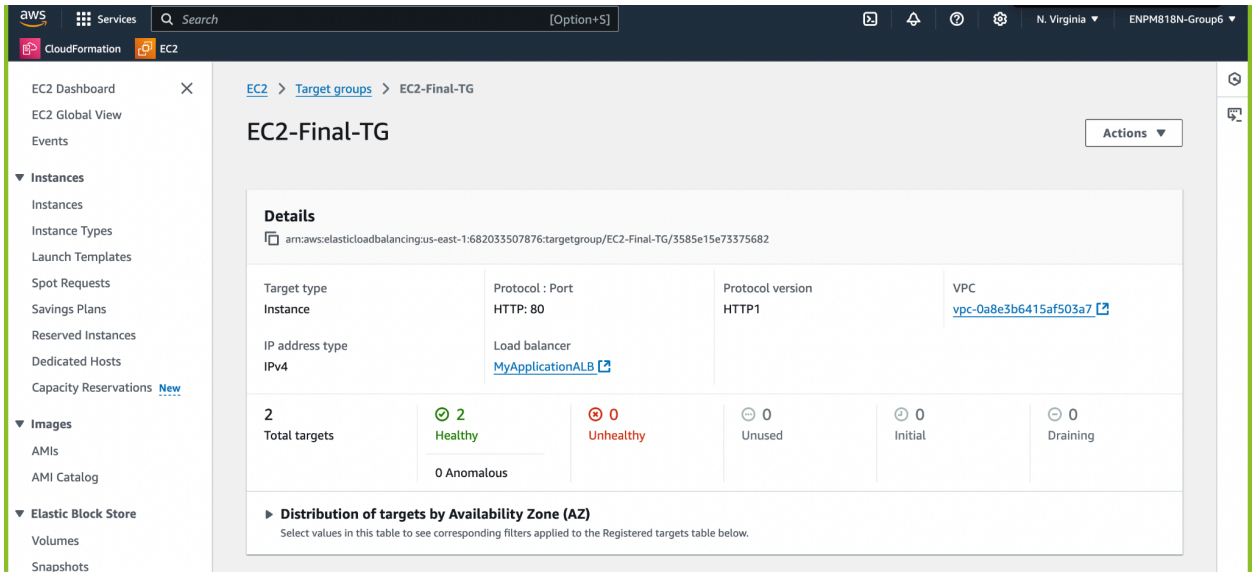
- **Function:**
  - Divides a VPC into several network segments to control and isolate network traffic effectively.
  - Enables the placement of both publicly accessible resources, such as for our web servers, and privately accessible resources, like our RDS databases.
- **Our Approach:**
  - We created the Public and Private Subnets. We created each with 2 availability zones (US-East-1a and 1b) to make PublicSubnetA and PublicSubnetB, and then PrivateSubnetA and PrivateSubnetB.
  - Each Subnet had a CIDR block of /24.
  - The Public Subnet housed the NAT gateway which allowed the internal EC2 instances to talk to the internet for tasks such as report downloading. The Private Subnet contained the EC2 instances which housed our ECommerce Platform, along with the auto scaling technique that we set up. The Private Subnet also housed our RDS Database.
  - The load balancer allowed the talking of all of the instances to each other since they were housed in private subnets


Services
Search
[Option+S]
CloudFormation
EC2
N. Virginia
ENPM818N-Group6
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
Network ACLs
Security groups
DNS firewall
Rule groups
Domain lists
Subnets (1/12) Info
Last updated less than a minute ago
Actions
Create subnet
Find resources by attribute or tag
Name
Subnet ID
State
VPC
IPv4 CIDR
IPv6 CIDR
IPv6 CIDR association ID
PublicSubnet2 subnet-0fc8412eac9113623 Available vpc-012d54e5806a277c3 | Midt... 10.0.2.0/24
– subnet-013014a631213bfb9 Available vpc-0a8e3b6415af503a7 | VPC-... 10.0.4.0/24
– subnet-0c0454e06f037cbca Available vpc-03b6b4deb8dbfb993 172.31.32.0/20
PublicSubnet1 subnet-022a7470acce3290a Available vpc-012d54e5806a277c3 | Midt... 10.0.1.0/24
subnet-022a7470acce3290a / PublicSubnet1
Details
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags
Details
Subnet ID subnet-022a7470acce3290a
Subnet ARN arn:aws:ec2:us-east-1:682033507876:subnet/subnet-022a7470acce3290a
State Available
IPv4 CIDR 10.0.1.0/24
Available IPv4 addresses 248
IPv6 CIDR –
IPv6 CIDR association ID –
Availability Zone us-east-1a
Availability Zone ID use1-az2
Network border group us-east-1
VPC vpc-012d54e5806a277c3 | MidtermVPC
Route table rtb-01579ee48c2156701
Network ACL acl-06b2b35c108a0c761
Default subnet No
Auto-assign public IPv4 address No
Auto-assign IPv6 address No
Auto-assign customer-owned IPv4 address No
Customer-owned IPv4 pool –
Outpost ID –
IPv4 CIDR reservations –
IPv6 CIDR reservations –
IPv6-only No
Hostname type IP name
Resource name DNS A record Disabled
Resource name DNS AAAA record
Owner
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Option+S]
CloudFormation
EC2
N. Virginia
ENPM818N-Group6
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
Network ACLs
Security groups
DNS firewall
Rule groups
Domain lists
Subnets (1/12) Info
Last updated 1 minute ago
Actions
Create subnet
Find resources by attribute or tag
Name
Subnet ID
State
VPC
IPv4 CIDR
IPv6 CIDR
IPv6 CIDR association ID
PublicSubnet2 subnet-0fc8412eac9113623 Available vpc-012d54e5806a277c3 | Midt... 10.0.2.0/24
– subnet-013014a631213bfb9 Available vpc-0a8e3b6415af503a7 | VPC-... 10.0.4.0/24
– subnet-0c0454e06f037cbca Available vpc-03b6b4deb8dbfb993 172.31.32.0/20
PublicSubnet1 subnet-022a7470acce3290a Available vpc-012d54e5806a277c3 | Midt... 10.0.1.0/24
subnet-0fc8412eac9113623 / PublicSubnet2
Details
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags
Details
Subnet ID subnet-0fc8412eac9113623
Subnet ARN arn:aws:ec2:us-east-1:682033507876:subnet/subnet-0fc8412eac9113623
State Available
IPv4 CIDR 10.0.2.0/24
Available IPv4 addresses 248
IPv6 CIDR –
IPv6 CIDR association ID –
Availability Zone us-east-1b
Availability Zone ID use1-az4
Network border group us-east-1
VPC vpc-012d54e5806a277c3 | MidtermVPC
Route table rtb-01579ee48c2156701
Network ACL acl-06b2b35c108a0c761
Default subnet No
Auto-assign public IPv4 address No
Auto-assign IPv6 address No
Auto-assign customer-owned IPv4 address No
Customer-owned IPv4 pool –
Outpost ID –
IPv4 CIDR reservations –
IPv6 CIDR reservations –
IPv6-only No
Hostname type IP name
Resource name DNS A record Disabled
Resource name DNS AAAA record
Owner
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Target group:

- **Function:**
  - Target groups specify where to send traffic within your architecture
  - Target groups monitor the health of their targets by running periodic health checks
- **Our Approach:**
  - We created a target group and attached it to our load balancer so when a user accesses the application, the load balancer distributes incoming requests to the EC2 instances in private subnets within the target group.



Compute, Load Balancing, and Data Storage

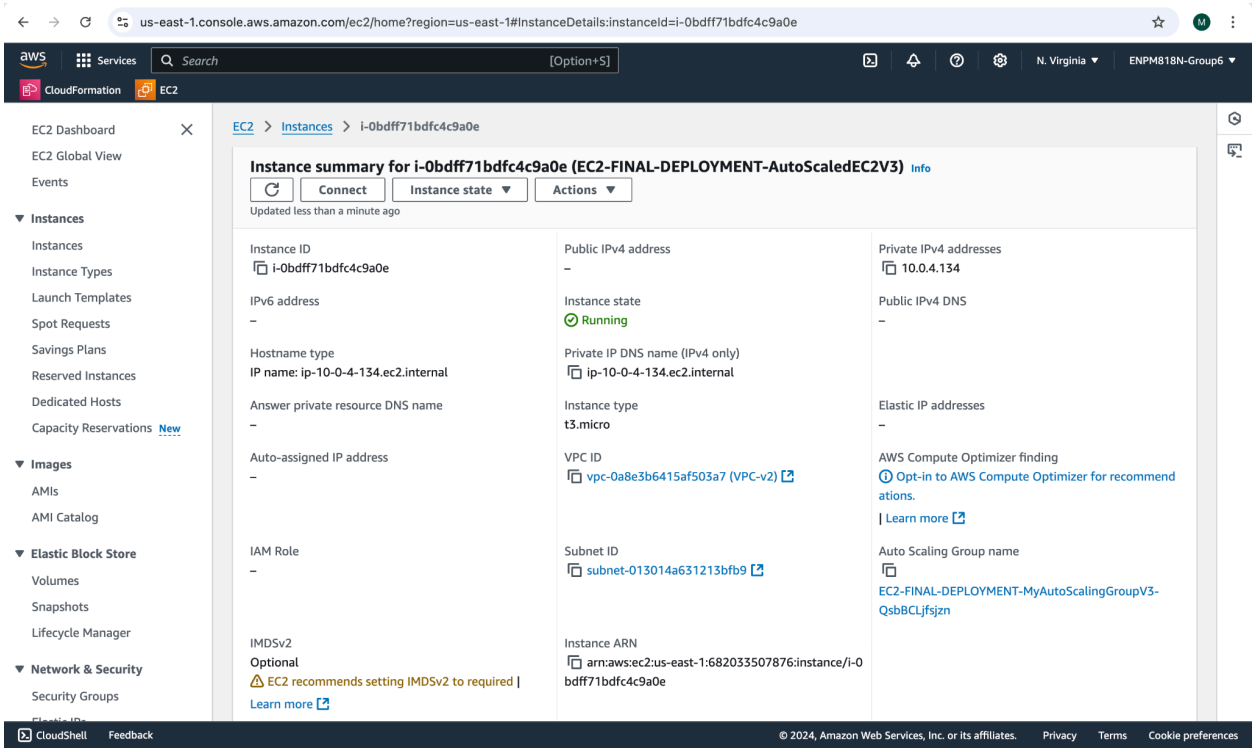
Amazon EC2 (Elastic Compute Cloud):

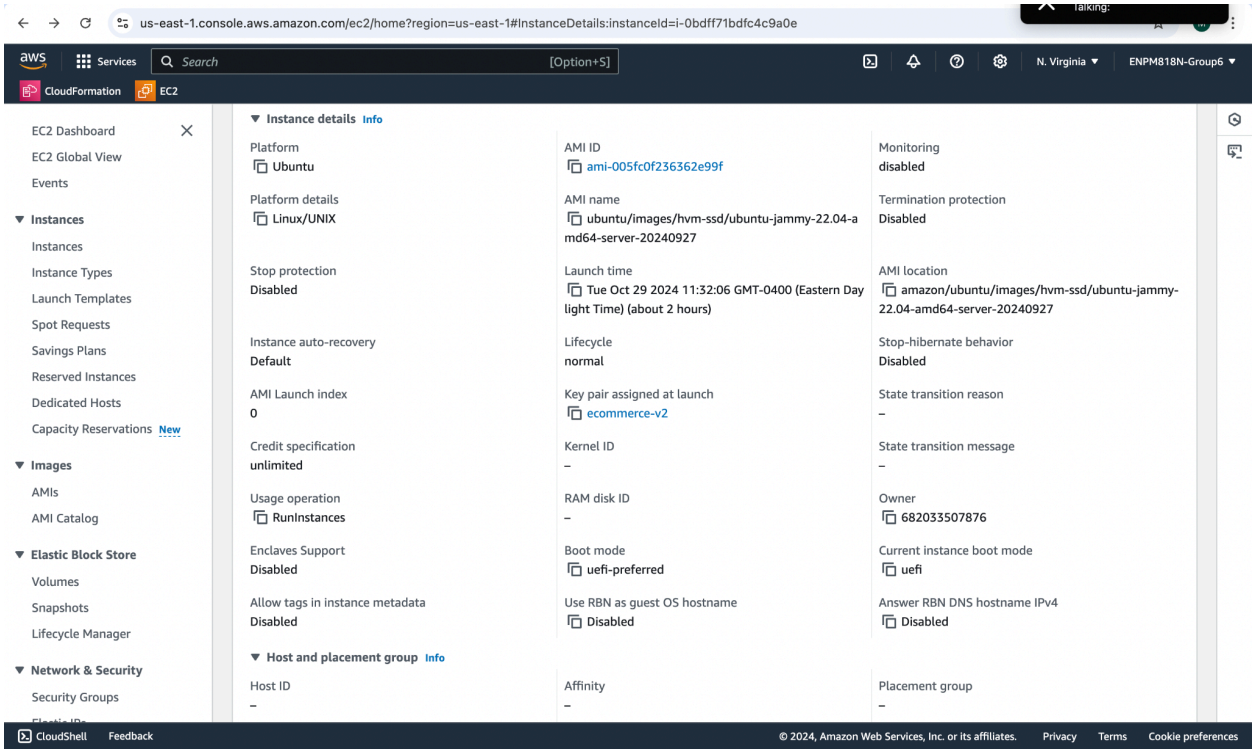
- **Interaction:**
  - EC2 instances to host our e-commerce application, ensuring scalability and flexibility.
  - EC2 was central to enabling the application's ability to handle varying traffic loads efficiently.
- **Our Approach:**
  - **Launch Template**: Configured EC2 instances with Amazon Machine Image(`ami-005fc0f236362e99f`), instance type (`t3.micro`), and SSH key (`ecommerce-v2`), as well as a custom `UserData` script.
  - **UserData Script**: The script, executed on instance launch, performs several key tasks:

- Updates the system, adds a repository for PHP 8.3, and installs Nginx, PHP, and Git with retries for resilience.
- Clones the e-commerce app from GitHub, transfers the app files to `/var/www/html`, and sets appropriate permissions for Nginx.
- Modifies the app's `connect.php` file to connect to a specified RDS database.
- Sets up a default Nginx server block to serve PHP files and restarts the service to apply configurations.

- **Security Group**: Allows inbound HTTP (port 80) and SSH (port 22) access from any IP address.

Deployed EC2 instance -




us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetails:instanceId=i-0bdff71bdfc4c9a0e
aws
Services
Search
[Option+S]
N. Virginia
ENPM818N-Group6
CloudFormation
EC2
EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Instance details Info
Platform
Ubuntu
AMI ID
ami-005fc0f236362e99f
Monitoring
disabled
Platform details
Linux/UNIX
AMI name
ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20240927
Termination protection
Disabled
Stop protection
Disabled
Launch time
Tue Oct 29 2024 11:32:06 GMT-0400 (Eastern Daylight Time) (about 2 hours)
AMI location
amazon/ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20240927
Instance auto-recovery
Default
Lifecycle
normal
Stop-hibernate behavior
Disabled
AMI Launch index
0
Key pair assigned at launch
ecommerce-v2
State transition reason
–
Credit specification
unlimited
Kernel ID
–
State transition message
–
Usage operation
RunInstances
RAM disk ID
–
Owner
682033507876
Enclaves Support
Disabled
Boot mode
uefi-preferred
Current instance boot mode
uefi
Allow tags in instance metadata
Disabled
Use RBN as guest OS hostname
Disabled
Answer RBN DNS hostname IPv4
Disabled
Host and placement group Info
Host ID
–
Affinity
–
Placement group
–
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

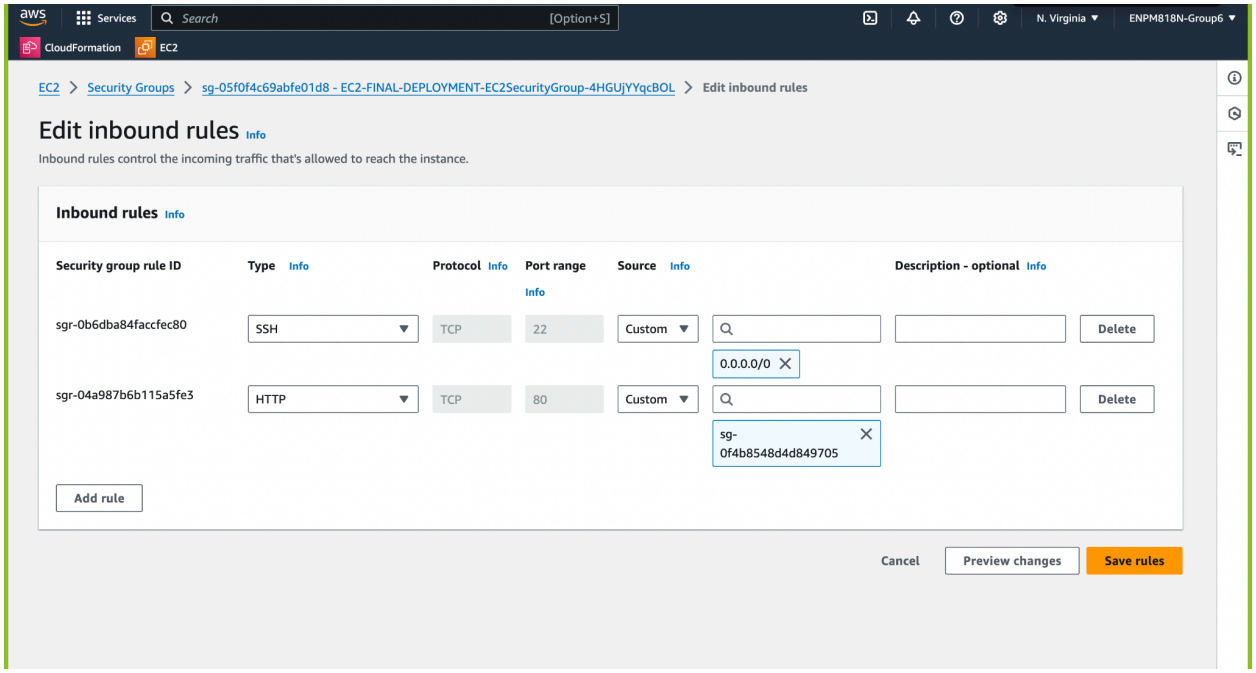
Connection to database from EC2 -

```
ubuntu@ip-10-0-1-100:~$ ssh -i "ecommerce-v2.pem" ubuntu@10.0.4.134
The authenticity of host '10.0.4.134 (10.0.4.134)' can't be established.
ED25519 key fingerprint is SHA256:hm8xDdm/cqhIiWuHXtSajdWqWuvahJcuj/A3S9fofAo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.4.134' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1015-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Tue Oct 29 17:24:55 UTC 2024

  System load:  0.0               Processes:             112
  Usage of /:   25.4% of 7.57GB   Users logged in:       0
  Memory usage: 27%               IPv4 address for ens5: 10.0.4.134
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

25 updates can be applied immediately.
18 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

2 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-4-134:~$ ls
818N-E_Commerce_Application
```

```
ubuntu@ip-10-0-4-134:~$ mysql -h rds-multiaz-myrdsinstance-qeruumzj9v4d.ctiowscucz1b.us-east-1.rds.amazonaws.com -u dbadmin -p ecommerce_1
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1237
Server version: 8.0.39 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Security group configuration -

(only ALB can communicate with EC2)

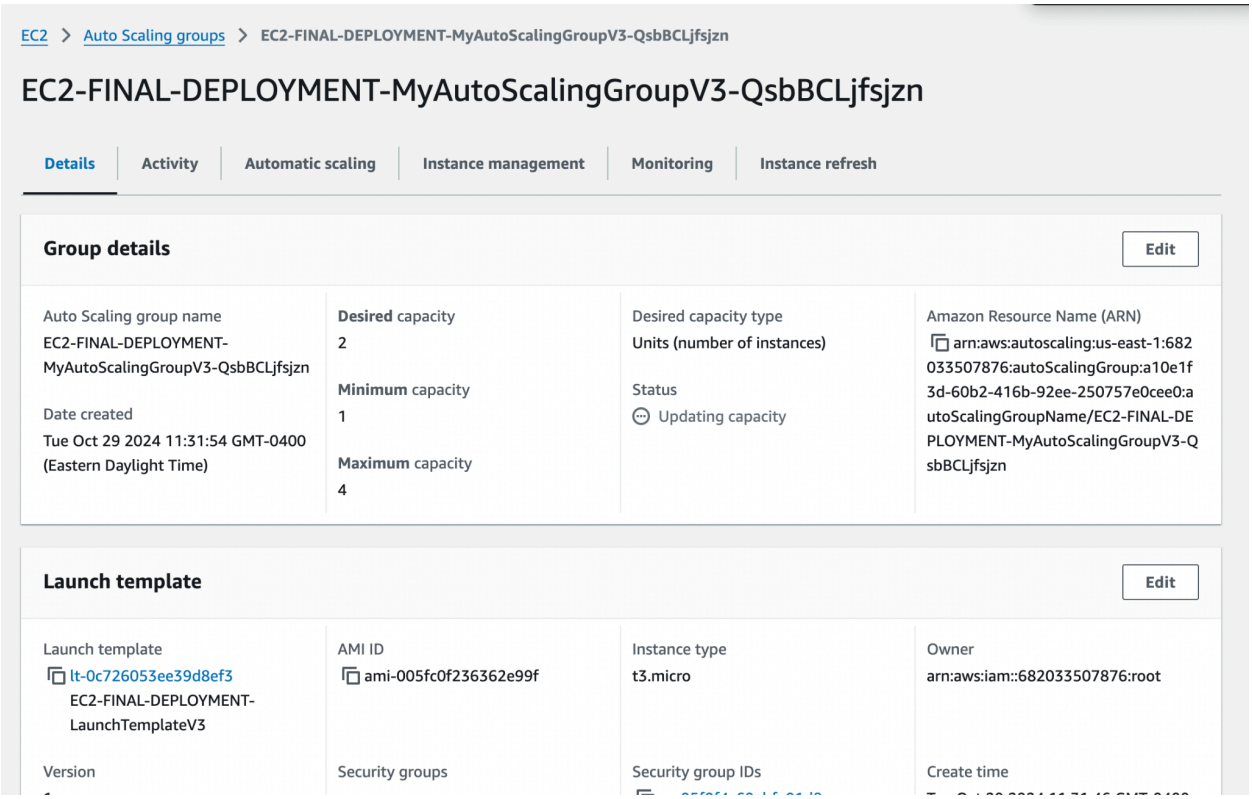


Auto Scaling:

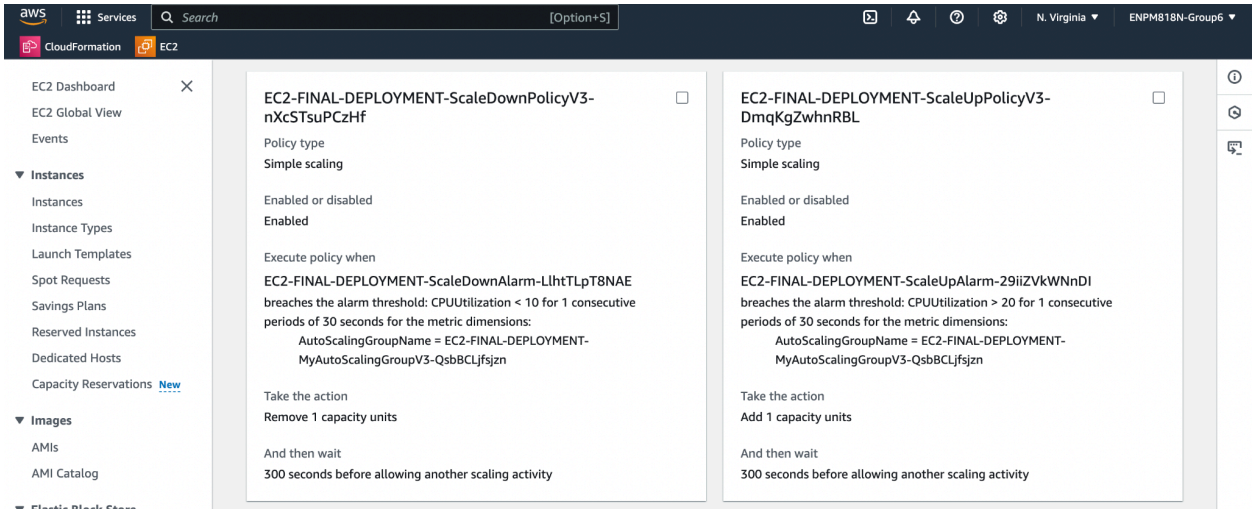
- **Interaction:**
  - To support high availability and manage traffic surges, we placed the EC2 instances in an Auto Scaling Group.
  - The ASG allowed for dynamic scaling, automatically adjusting the number of instances based on CPU usage thresholds, and optimizing cost-effectiveness.
- **Our Approach:**
  - Defined the ASG to deploy EC2 instances in specified private subnets.
  - We have set the MinSize to 1, MaxSize to 4, and DesiredCapacity to 2, providing a balance between resource availability and cost efficiency.

- Configured EC2-based health checks with a 5-minute grace period, which allows new instances to stabilize before they are monitored for health.

Auto-scaling group -

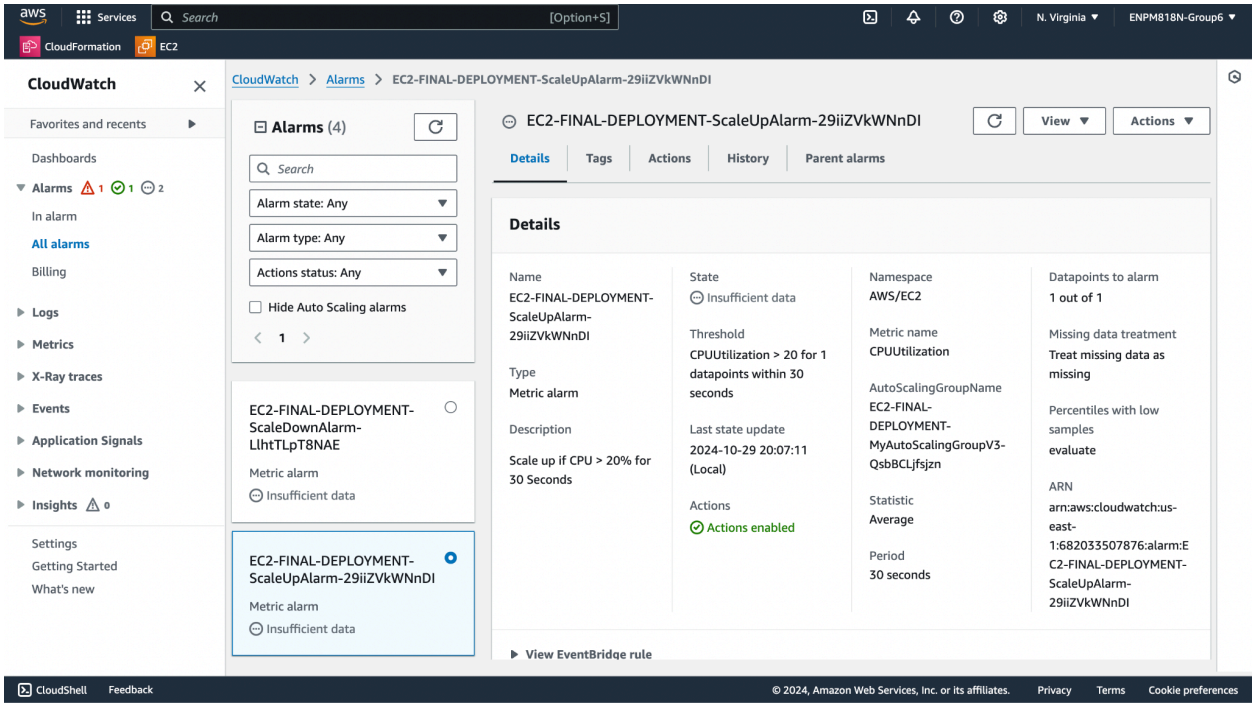


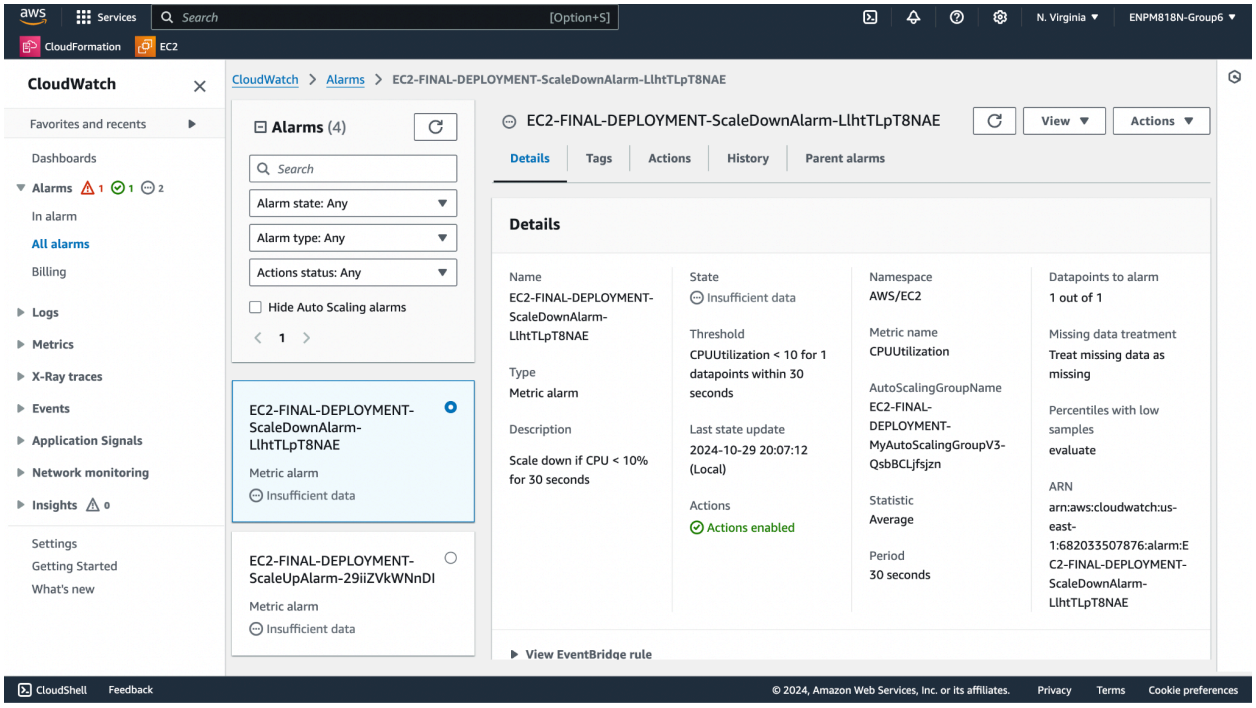
Auto-scaling policies -



Cloud Watch Alarms:

- **Interaction:**
  - CloudWatch metrics were monitored to track CPU utilization, memory usage, and network activity, helping us fine-tune the scaling policies for optimal performance.
- **Our Approach:**
  - **Scale-Up Policy**: Increase the ASG's instance count by one if triggered.
  - **Scale-Down Policy**: Decrease the ASG's instance count by one if triggered.
  - **ScaleUpAlarm**: Monitors CPU utilization of instances in the ASG and triggers the **Scale-Up Policy** if average CPU usage exceeds 20% for 30 seconds.
  - **ScaleDownAlarm**: Triggers the **Scale-Down Policy** if CPU usage drops below 10% for 30 seconds, allowing the ASG to reduce instance count during low usage.

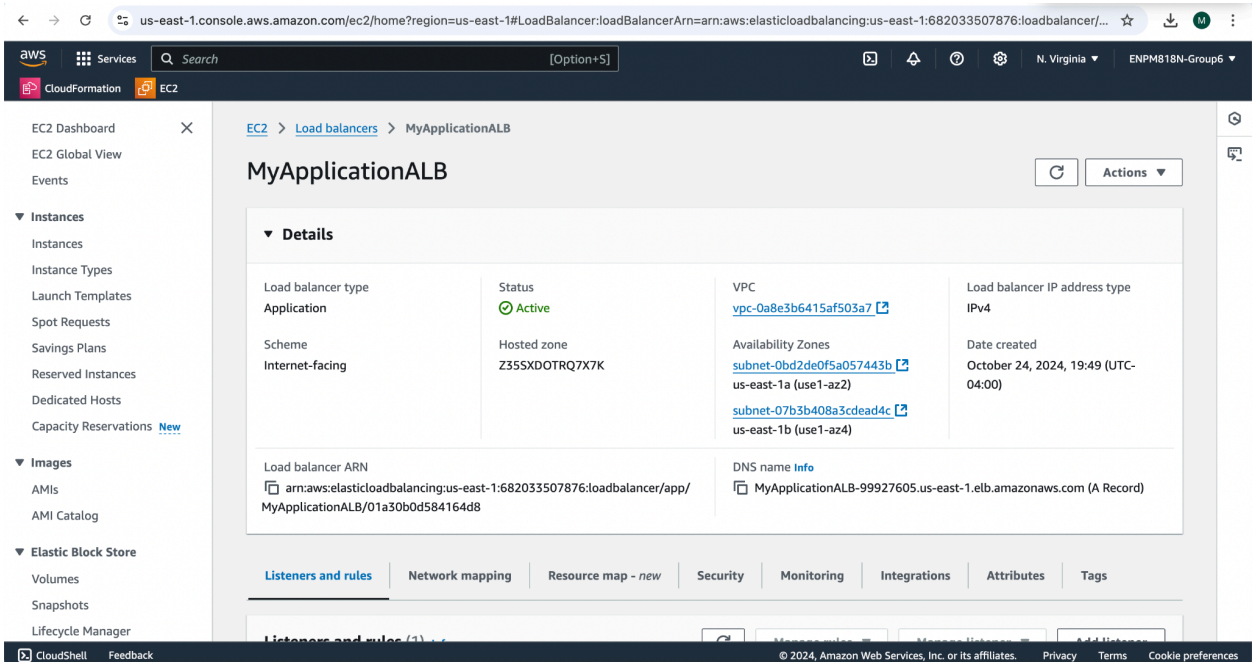

aws
Services
Search
[Option+S]
N. Virginia
ENPM818N-Group6
CloudFormation
EC2
CloudWatch
CloudWatch > Alarms > EC2-FINAL-DEPLOYMENT-ScaleUpAlarm-29iiZVkWNnDI
Favorites and recents
Dashboards
Alarms 1 1 2
In alarm
All alarms
Billing
Logs
Metrics
X-Ray traces
Events
Application Signals
Network monitoring
Insights 0
Settings
Getting Started
What's new
Alarms (4)
Search
Alarm state: Any
Alarm type: Any
Actions status: Any
Hide Auto Scaling alarms
1
EC2-FINAL-DEPLOYMENT-ScaleDownAlarm-LlhtTLpT8NAE
Metric alarm
Insufficient data
EC2-FINAL-DEPLOYMENT-ScaleUpAlarm-29iiZVkWNnDI
Metric alarm
Insufficient data
EC2-FINAL-DEPLOYMENT-ScaleUpAlarm-29iiZVkWNnDI
View
Actions
Details
Tags
Actions
History
Parent alarms
Details
Name
EC2-FINAL-DEPLOYMENT-ScaleUpAlarm-29iiZVkWNnDI
Type
Metric alarm
Description
Scale up if CPU > 20% for 30 Seconds
State
Insufficient data
Threshold
CPUUtilization > 20 for 1 datapoints within 30 seconds
Last state update
2024-10-29 20:07:11 (Local)
Actions
Actions enabled
Namespace
AWS/EC2
Metric name
CPUUtilization
AutoScalingGroupName
EC2-FINAL-DEPLOYMENT-MyAutoScalingGroupV3-QsbBCLjfsjzn
Statistic
Average
Period
30 seconds
Datapoints to alarm
1 out of 1
Missing data treatment
Treat missing data as missing
Percentiles with low samples
evaluate
ARN
arn:aws:cloudwatch:us-east-1:682033507876:alarm:EC2-FINAL-DEPLOYMENT-ScaleUpAlarm-29iiZVkWNnDI
View EventBridge rule
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

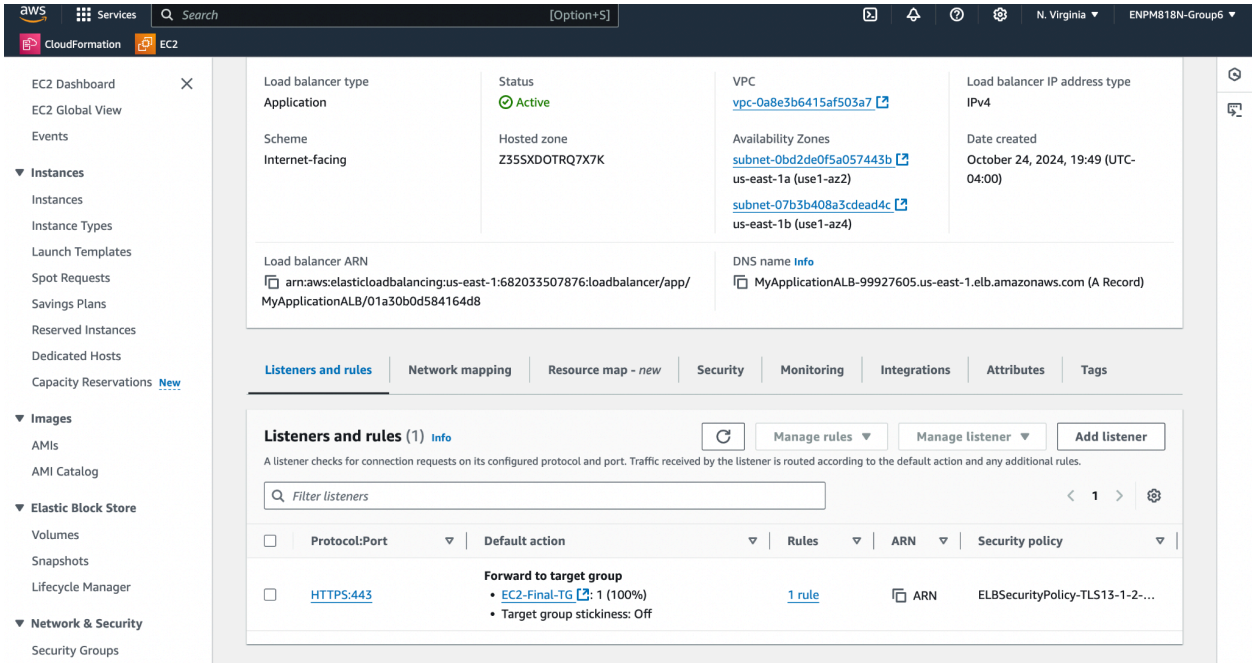

aws
Services
Search
[Option+S]
N. Virginia
ENPM818N-Group6
CloudFormation
EC2
CloudWatch
CloudWatch > Alarms > EC2-FINAL-DEPLOYMENT-ScaleDownAlarm-LlhtTLpT8NAE
Favorites and recents
Dashboards
Alarms 1 1 2
In alarm
All alarms
Billing
Logs
Metrics
X-Ray traces
Events
Application Signals
Network monitoring
Insights 0
Settings
Getting Started
What's new
Alarms (4)
Search
Alarm state: Any
Alarm type: Any
Actions status: Any
Hide Auto Scaling alarms
1
EC2-FINAL-DEPLOYMENT-ScaleDownAlarm-LlhtTLpT8NAE
Metric alarm
Insufficient data
EC2-FINAL-DEPLOYMENT-ScaleUpAlarm-29iiZVkWNnDI
Metric alarm
Insufficient data
EC2-FINAL-DEPLOYMENT-ScaleDownAlarm-LlhtTLpT8NAE
View
Actions
Details
Tags
Actions
History
Parent alarms
Details
Name
EC2-FINAL-DEPLOYMENT-ScaleDownAlarm-LlhtTLpT8NAE
Type
Metric alarm
Description
Scale down if CPU < 10% for 30 seconds
State
Insufficient data
Threshold
CPUUtilization < 10 for 1 datapoints within 30 seconds
Last state update
2024-10-29 20:07:12 (Local)
Actions
Actions enabled
Namespace
AWS/EC2
Metric name
CPUUtilization
AutoScalingGroupName
EC2-FINAL-DEPLOYMENT-MyAutoScalingGroupV3-QsbBCLjfsjzn
Statistic
Average
Period
30 seconds
Datapoints to alarm
1 out of 1
Missing data treatment
Treat missing data as missing
Percentiles with low samples
evaluate
ARN
arn:aws:cloudwatch:us-east-1:682033507876:alarm:EC2-FINAL-DEPLOYMENT-ScaleDownAlarm-LlhtTLpT8NAE
View EventBridge rule
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Load Balancer:

Application Load Balancer:

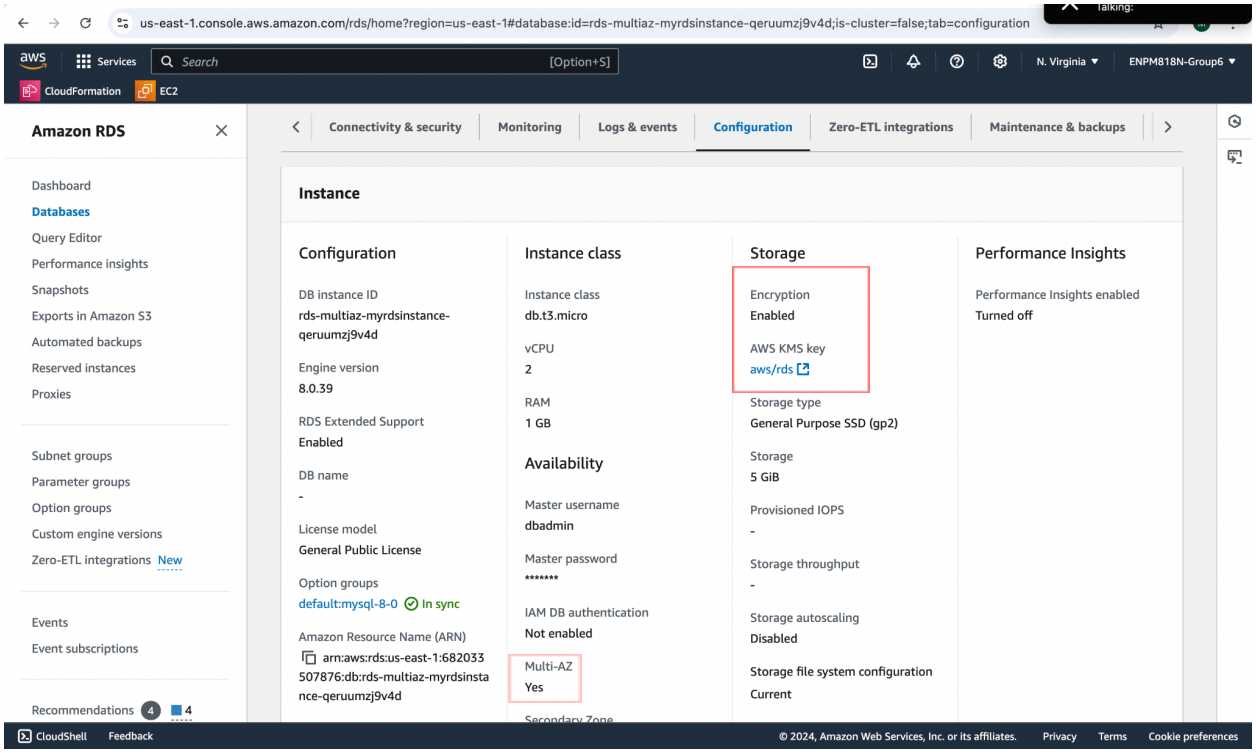
- **Interaction:**
  - Able to distribute incoming HTTPS traffic across our EC2 instances and balance the load accordingly
  - This ALB helps maintain availability for our platform and help with our zero downtime value
  - Can check which instances are healthy and route traffic to the ones that are healthier to keep a balanced vision
- **Our Approach:**
  - We set up a ALB to help forward the traffic for our ECommerce platform
  - We made a security group for the ALB, and allowed access from port 443 for HTTPS inbound traffic
  - We made a listener for the ALB on port 443 and secured the traffic using a SSL certificate for secure traffic



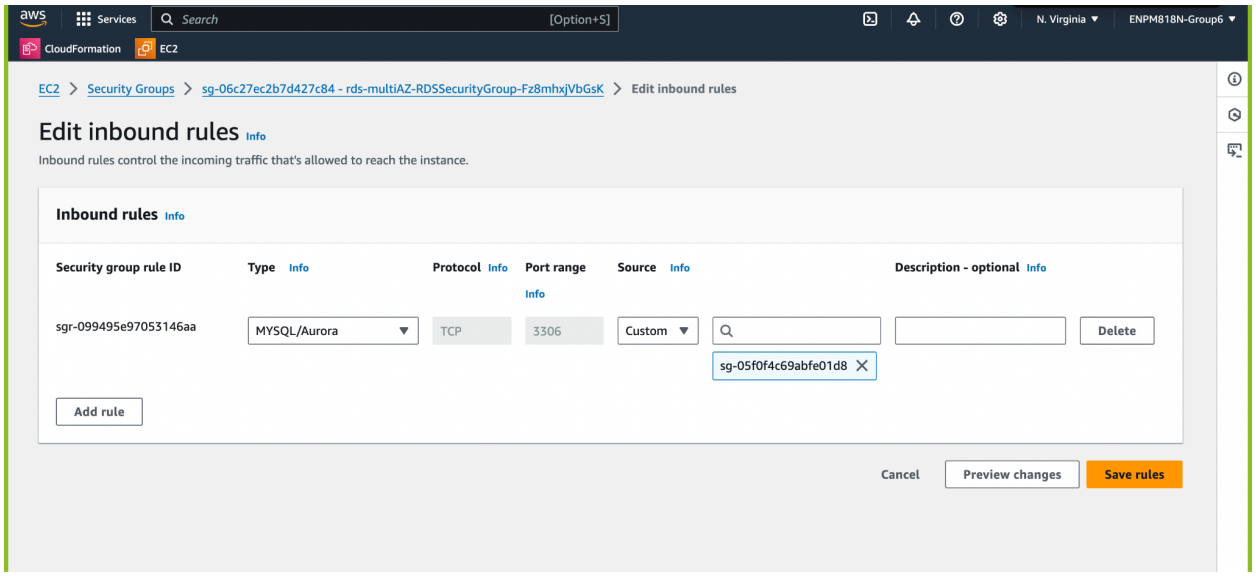

aws
Services
Search
[Option+S]
N. Virginia
ENPM818N-Group6
CloudFormation
EC2
EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Load balancer type
Application
Status
Active
VPC
vpc-0a8e3b6415af503a7
Load balancer IP address type
IPv4
Scheme
Internet-facing
Hosted zone
Z35SXDOTRQ7X7K
Availability Zones
subnet-0bd2de0f5a057443b
us-east-1a (use1-az2)
subnet-07b3b408a3cdead4c
us-east-1b (use1-az4)
Date created
October 24, 2024, 19:49 (UTC-04:00)
Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:682033507876:loadbalancer/app/MyApplicationALB/01a30b0d584164d8
DNS name Info
MyApplicationALB-99927605.us-east-1.elb.amazonaws.com (A Record)
Listeners and rules
Network mapping
Resource map - new
Security
Monitoring
Integrations
Attributes
Tags
Listeners and rules (1) Info
Manage rules
Manage listener
Add listener
A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.
Filter listeners
1
Protocol:Port
Default action
Rules
ARN
Security policy
HTTPS:443
Forward to target group
• EC2-Final-TG: 1 (100%)
• Target group stickiness: Off
1 rule
ARN
ELBSecurityPolicy-TLS13-1-2-...

Amazon RDS (Relational Database Service):

- **Function:**
  - Streamlines the configuration, management, and scaling of a relational database, making it easier to integrate into applications.
  - Delivers cost-effective and scalable capacity while handling tedious database administration tasks, allowing you to prioritize applications and business objectives.
- **Interaction:**
  - Connects seamlessly with AWS Identity and Access Management (IAM) to regulate access to RDS instances, enabling precise control over management privileges.
  - Engages with Amazon EC2 instances to furnish database services for applications operating within EC2, guaranteeing streamlined data retrieval and storage processes.
- **Approach:**
  - We selected MySQL as the database engine due to its reliability, performance, and support for a wide range of applications.
  - Multi-AZ deployment was enabled to enhance the availability and fault tolerance of the database.
  - Amazon Key Management Service (KMS) was used to encrypt the database at rest.
  - A security group was configured to control access to the RDS instance, allowing only trusted EC2 instances to communicate with the database. The subnet configuration was defined to ensure the RDS instance operates within a secure Virtual Private Cloud (VPC), providing network isolation and enhanced security.



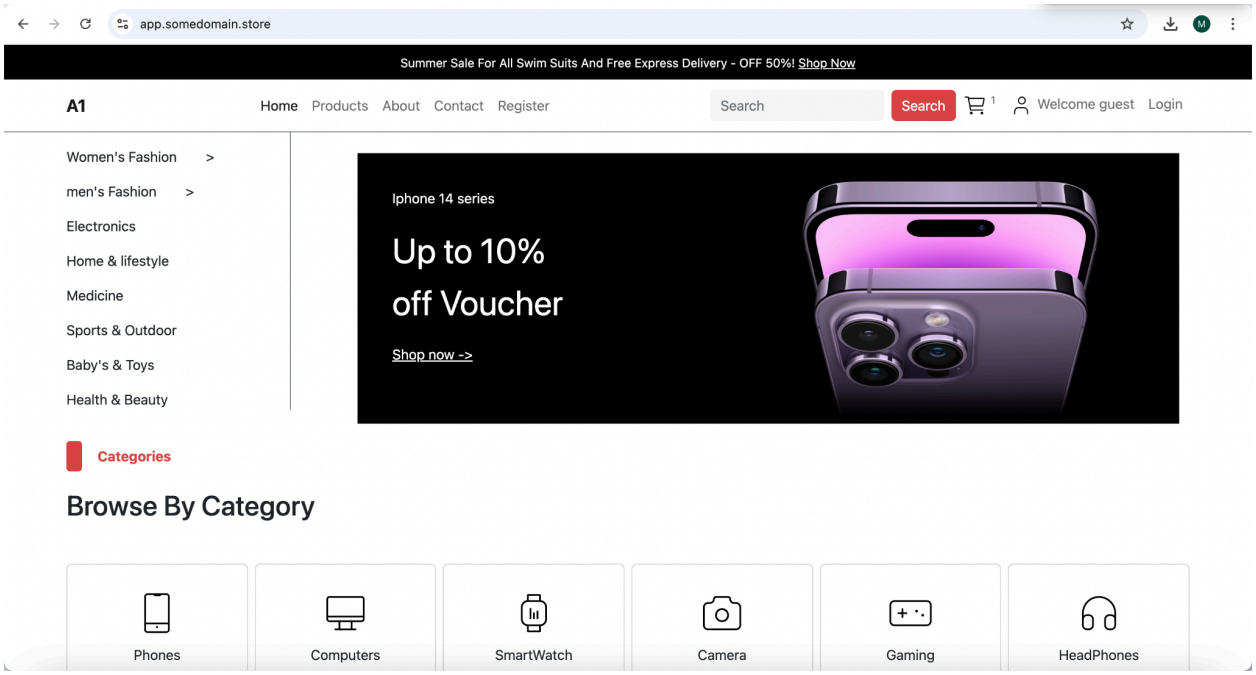
Only EC2 can communicate with RDS -



Encrypted password in database -

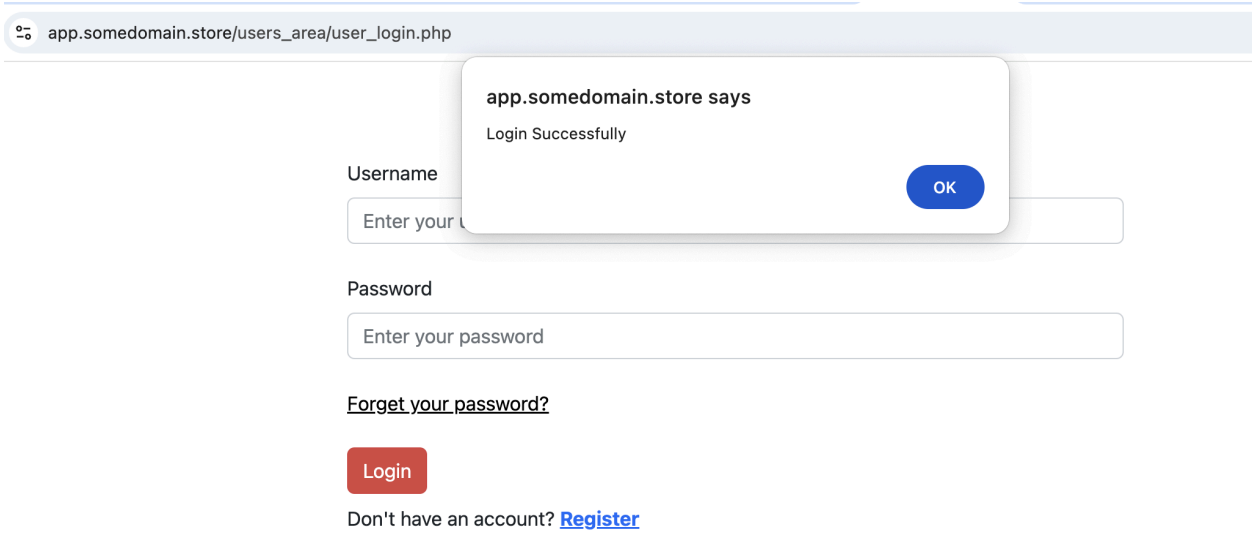
```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-10-0-4-134:~$ mysql -h rds-multiaz-myrdsinstance-qeruumzj9v4d.ctiowscucz1b.us-east-1.rds.amazonaws.com -u dbadmin -p ecommerce_1
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1237
Server version: 8.0.39 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use ecommerce_1;
Database changed
mysql> Show tables;
+-----------------------+
| Tables_in_ecommerce_1 |
+-----------------------+
| admin_table           |
| brands                |
| card_details          |
| categories            |
| orders_pending        |
| products              |
| user_orders           |
| user_payments         |
| user_table            |
+-----------------------+
9 rows in set (0.00 sec)

mysql> SELECT * FROM admin_table;
+----------+------------+----------------+--------------------+--------------------------------------------------------------+
| admin_id | admin_name | admin_email    | admin_image        | admin_password                                               |
+----------+------------+----------------+--------------------+--------------------------------------------------------------+
|        1 | abdo       | abdo@gmail.com | logo after 3d_2.png | $2y$10$M/A/r5j/GSeJrAZxI8NtRu9eG5yNltfgTrfQVoClfSIF/pzNUXa2W |
+----------+------------+----------------+--------------------+--------------------------------------------------------------+
1 row in set (0.00 sec)
```

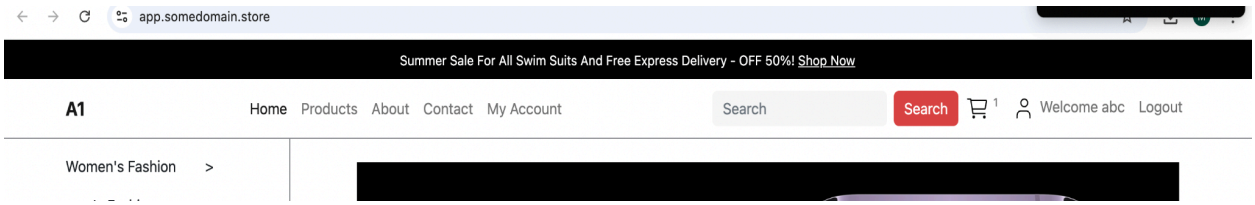
DEPLOYED APPLICATION



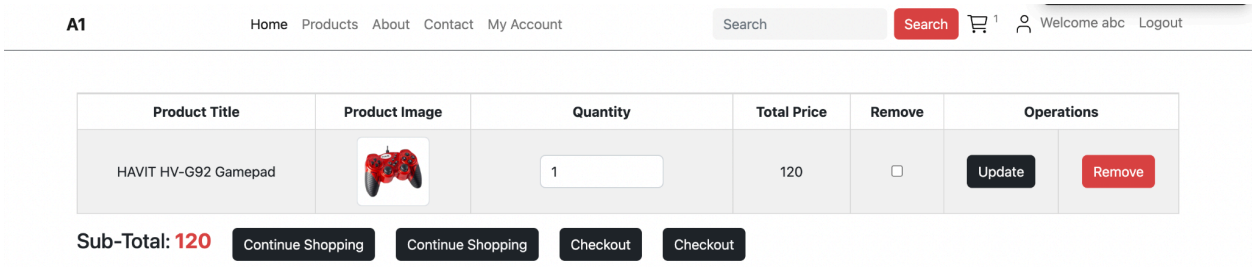
CRUD OPERATIONS VIA E-COMMERCE PLATFORM

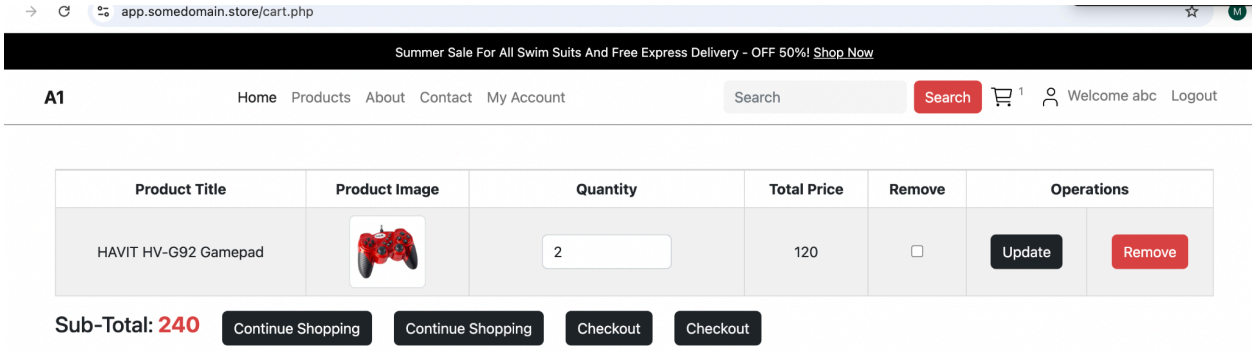
LOGIN



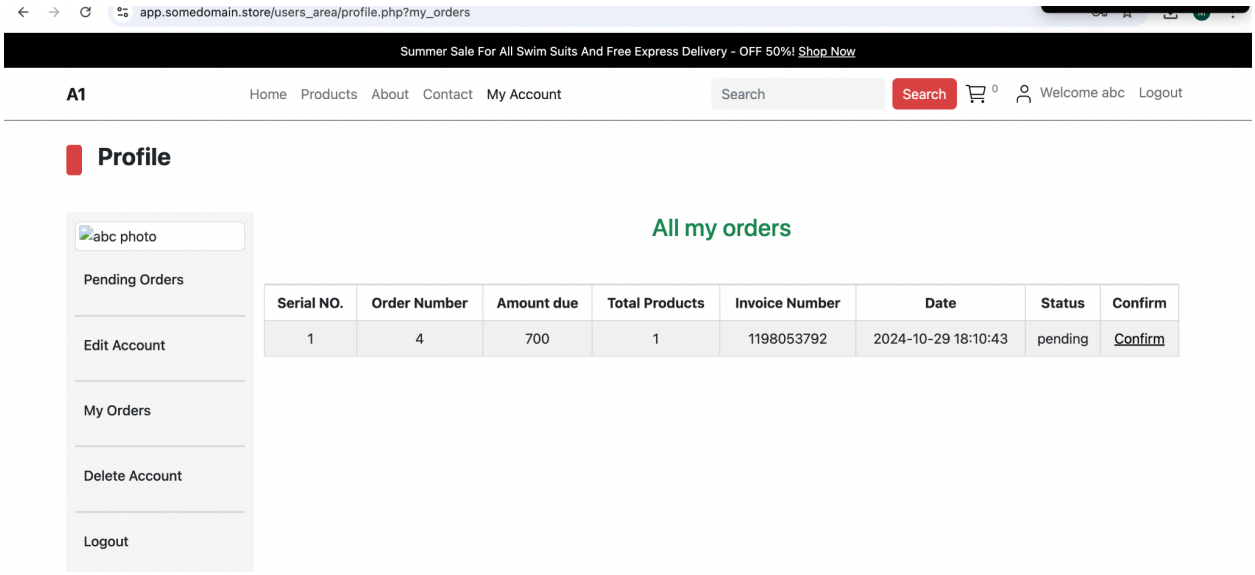


ADDING/UPDATING TO CART -





ORDER PLACED



DATABASE AFTER THE ABOVE OPERATIONS

```
mysql> use ecommerce_1;
Database changed
mysql> show tables;
+-------------------------+
| Tables_in_ecommerce_1 |
+-------------------------+
| admin_table           |
| brands                |
| card_details          |
| categories            |
| orders_pending        |
| products              |
| user_orders           |
| user_payments         |
| user_table            |
+-------------------------+
9 rows in set (0.00 sec)

mysql> SELECT * FROM admin_table;
+----------+------------+---------------+-------------------+--------------------------------------------------------------+
| admin_id | admin_name | admin_email   | admin_image       | admin_password                                               |
+----------+------------+---------------+-------------------+--------------------------------------------------------------+
|        1 | abdo       | abdo@gmail.com | logo after 3d_2.png | $2y$10$M/A/r5j/GSeJrAZxI8NtRu9eG5yNltfgTrfQVoClfSIF/pzNUXa2W |
+----------+------------+---------------+-------------------+--------------------------------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM user_table;
+---------+----------+---------------+--------------------------------------------------------------+-------------------+---------------+-----------------------+-------------+
| user_id | username | user_email    | user_password                                                | user_image        | user_ip       | user_address          | user_mobile |
+---------+----------+---------------+--------------------------------------------------------------+-------------------+---------------+-----------------------+-------------+
|       1 | abdo     | abdo@gmail.com | $2y$10$5ynby9fq7wf2ZmHlkvehu.JGbK6r7zZLtLzuJz9Jt5FP03rGZ9Mj. | new logo after Edit1920.png | ::1           | Cairo                 | 123456789   |
|       2 | abc      | abc@gmail.com  | $2y$10$xqwYuOxwwQQTk.RrKEqQveQI/VZxgA5tTyz7DfKRes8pgL5FZaV2u | apple.png         | 69.137.236.162 | street 123, apt 303   | 123456789   |
+---------+----------+---------------+--------------------------------------------------------------+-------------------+---------------+-----------------------+-------------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM user_orders;
+----------+---------+------------+----------------+----------------+---------------------+--------------+
| order_id | user_id | amount_due | invoice_number | total_products | order_date          | order_status |
+----------+---------+------------+----------------+----------------+---------------------+--------------+
|        1 |       1 |       1160 |      312346784 |              3 | 2023-10-22 15:31:20 | paid         |
|        2 |       1 |        760 |     1918753782 |              1 | 2023-10-24 00:25:10 | pending      |
|        3 |       1 |        240 |      351837813 |              1 | 2023-10-24 18:41:02 | pending      |
|        4 |       2 |        700 |     1198053792 |              1 | 2024-10-29 18:10:43 | pending      |
+----------+---------+------------+----------------+----------------+---------------------+--------------+
4 rows in set (0.01 sec)
```

SECURING THE APPLICATION - PHASE 2

Introduction:

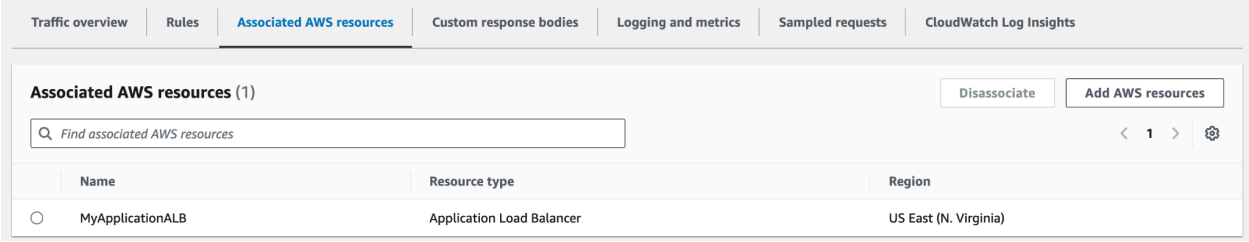
- In building an e-commerce application, one of the main concerns is to ensure the application is secure from any kind of security breaches.
- A security threat not only risks exposing sensitive information but also results in significant reputational damage and regulatory problems.
- To help mitigate this trouble we have to secure the application's infrastructure through AWS services which focus on application security and data protection.

In this phase we specially target **two critical security areas**:

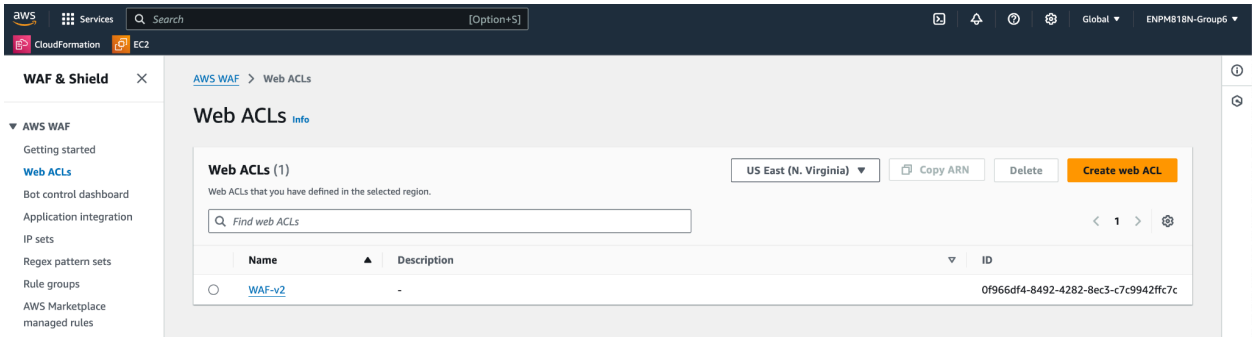
- Protecting the application from web-based attacks using Web Application Firewall (WAF)
- Securing the data in transit between clients, the application servers, and the backend database

AWS Web Application Firewall (WAF):

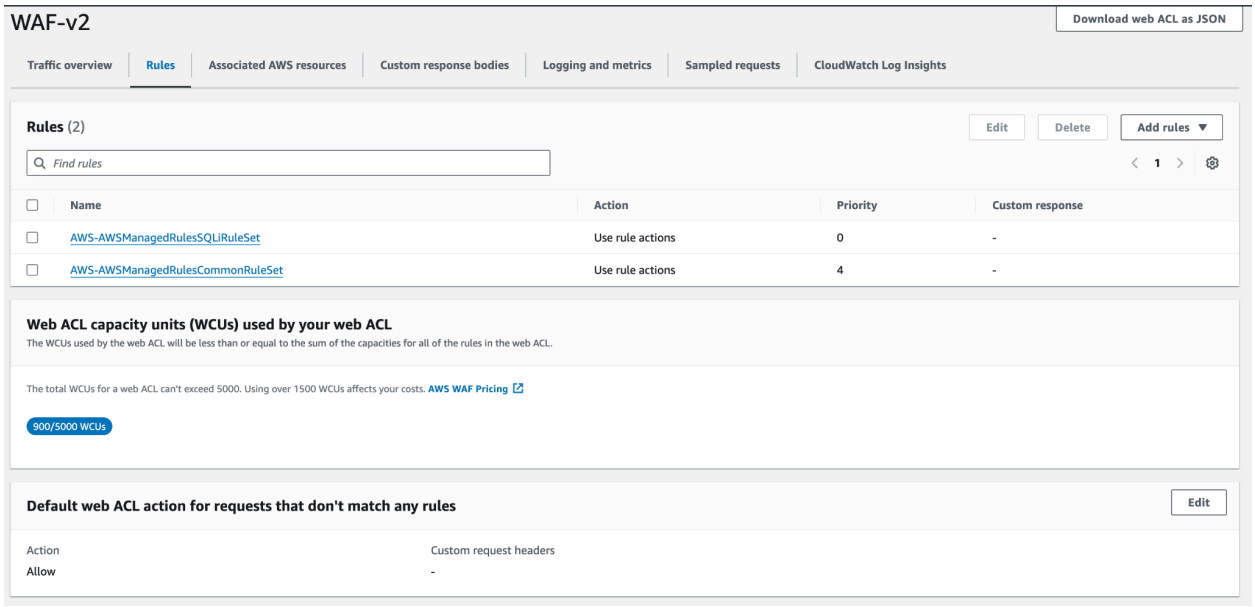
- **Interaction:**
  - AWS WAF blocks known attack patterns and web-based exploits such as SQL Injection and Cross-Site Scripting (XSS) by using the defined rules in WAF
  - AWS WAF inspects incoming traffic to the ALB, blocking threats before they reach application servers
  - Real-time WAF metrics and alarms in CloudWatch allow proactive monitoring and security responses



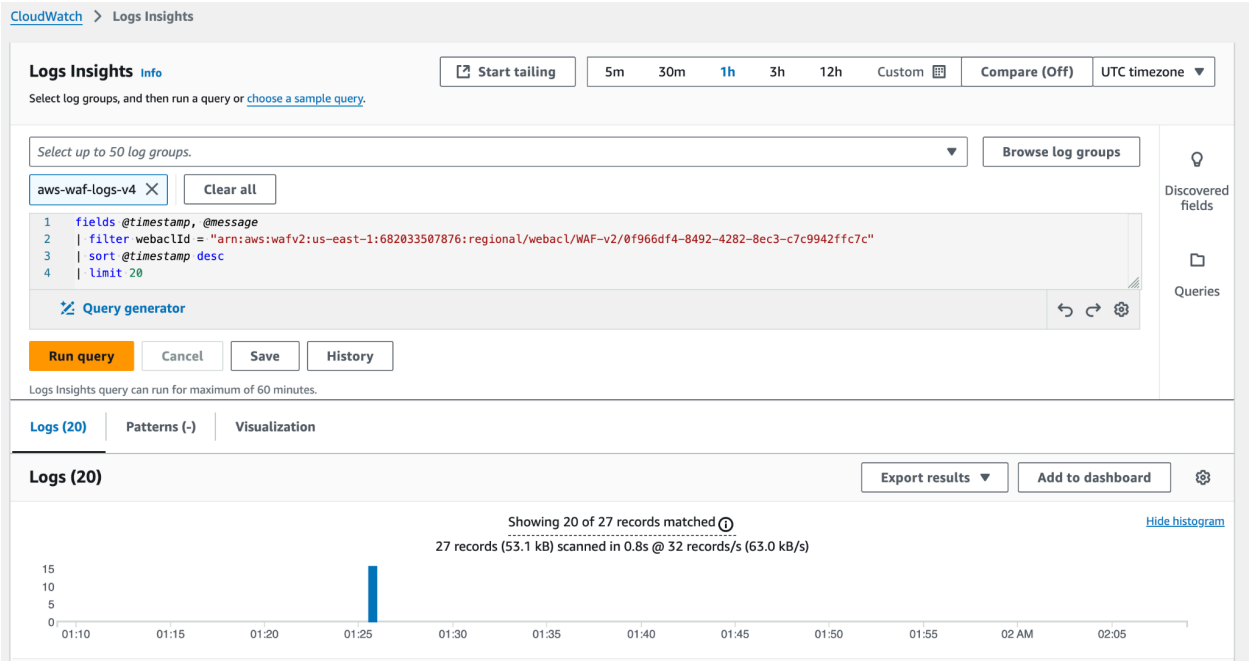
- **Our Approach:**
  - We have set up WAF by creating a Web ACL (WAF-v2) to secure the platform



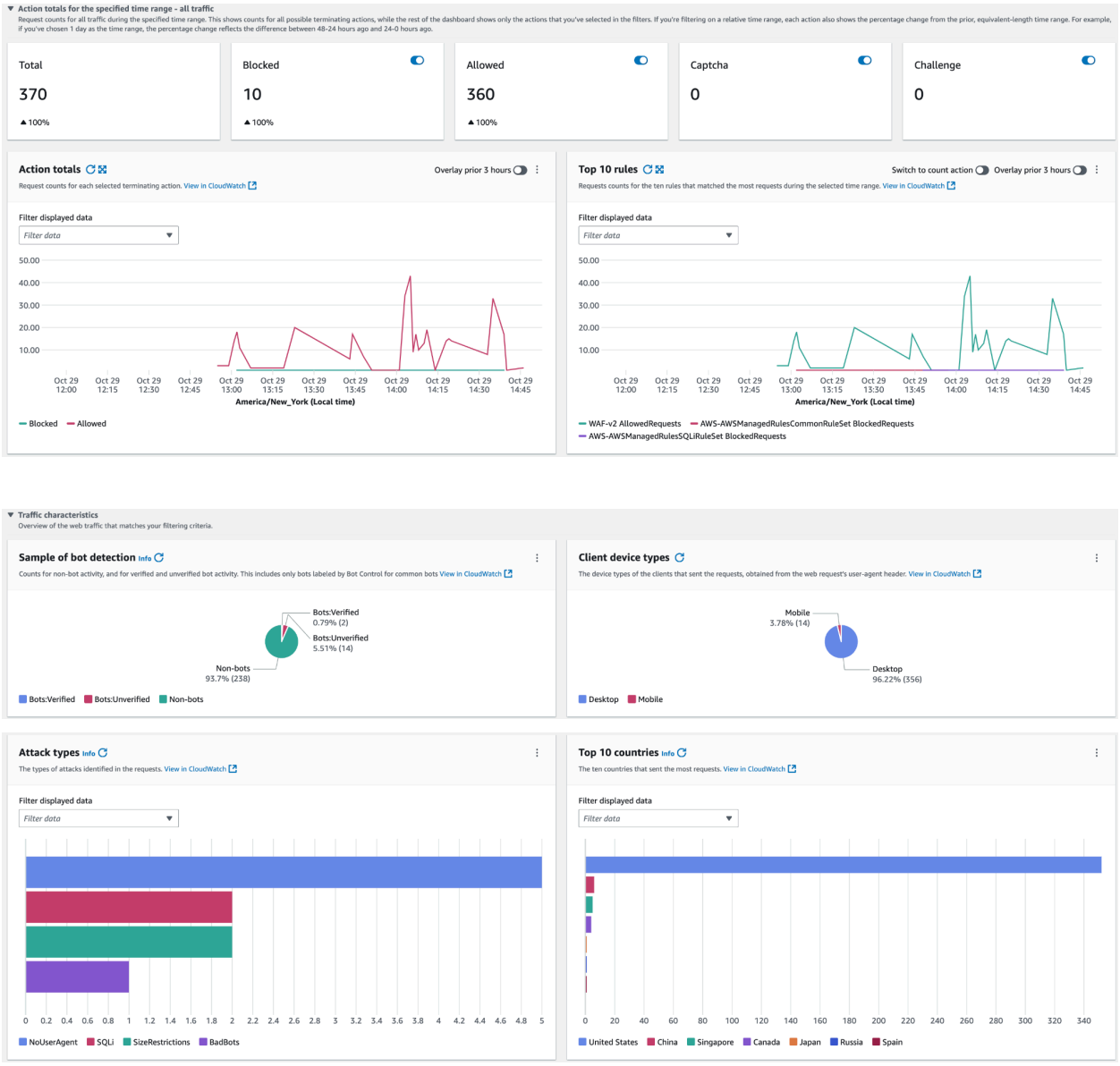
- The WAF rules used for this project were selected to cover the most common and impactful web-based threats. They are:
  - **AWS-AWSManagedRulesSQLiRuleSet** : To detect and block the SQL injection attacks, which could allow unauthorized access or any unallowed CRUD operations to the database.
  - **AWS-AWSManagedRulesCommonRuleSet** : Adds an additional layer of security to the application and provides general protection against common vulnerabilities and other known attack patterns.



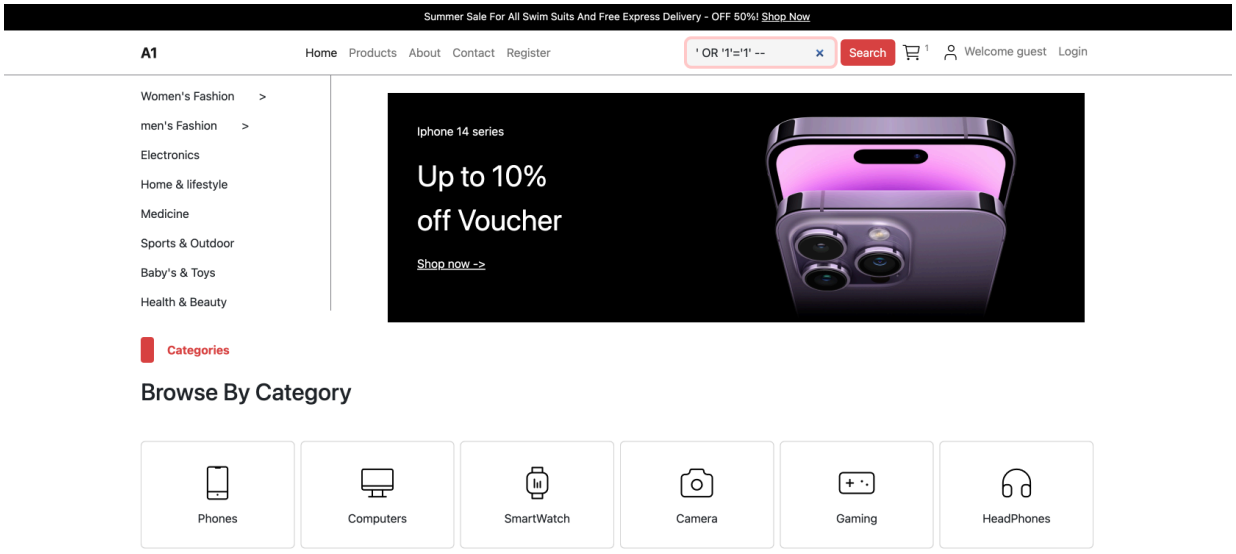
- Amazon CloudWatch is integrated to monitor the Web ACL's performance and effectiveness. By enabling CloudWatch logging, insights into blocked requests, allowed requests, and frequently matched rules are analyzed carefully.



#	@timestamp	@message
▶ 1	2024-10-29T01:37:29.9…	{"timestamp":1730165849973,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 2	2024-10-29T01:26:44.8…	{"timestamp":1730165204812,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 3	2024-10-29T01:26:26.6…	{"timestamp":1730165186663,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 4	2024-10-29T01:26:03.1…	{"timestamp":1730165163189,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 5	2024-10-29T01:26:03.1…	{"timestamp":1730165163160,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 6	2024-10-29T01:26:01.4…	{"timestamp":1730165161482,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 7	2024-10-29T01:26:01.4…	{"timestamp":1730165161438,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 8	2024-10-29T01:25:39.9…	{"timestamp":1730165139905,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 9	2024-10-29T01:25:36.9…	{"timestamp":1730165136958,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 10	2024-10-29T01:25:36.9…	{"timestamp":1730165136954,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 11	2024-10-29T01:25:36.9…	{"timestamp":1730165136901,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 12	2024-10-29T01:25:31.0…	{"timestamp":1730165131044,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 13	2024-10-29T01:25:30.6…	{"timestamp":1730165130627,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 14	2024-10-29T01:25:30.5…	{"timestamp":1730165130515,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 15	2024-10-29T01:25:30.5…	{"timestamp":1730165130515,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 16	2024-10-29T01:25:30.5…	{"timestamp":1730165130514,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 17	2024-10-29T01:25:30.5…	{"timestamp":1730165130514,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 18	2024-10-29T01:25:30.5…	{"timestamp":1730165130513,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 19	2024-10-29T01:25:30.5…	{"timestamp":1730165130507,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId
▶ 20	2024-10-29T01:25:30.4…	{"timestamp":1730165130458,"formatVersion":1,"webaclId":"arn:aws:wafv2:us-east-1:682033507876:regional/webacl/WAF-v2/0f966df4-8492-4282-8ec3-c7c9942ffc7c","terminatingRuleId

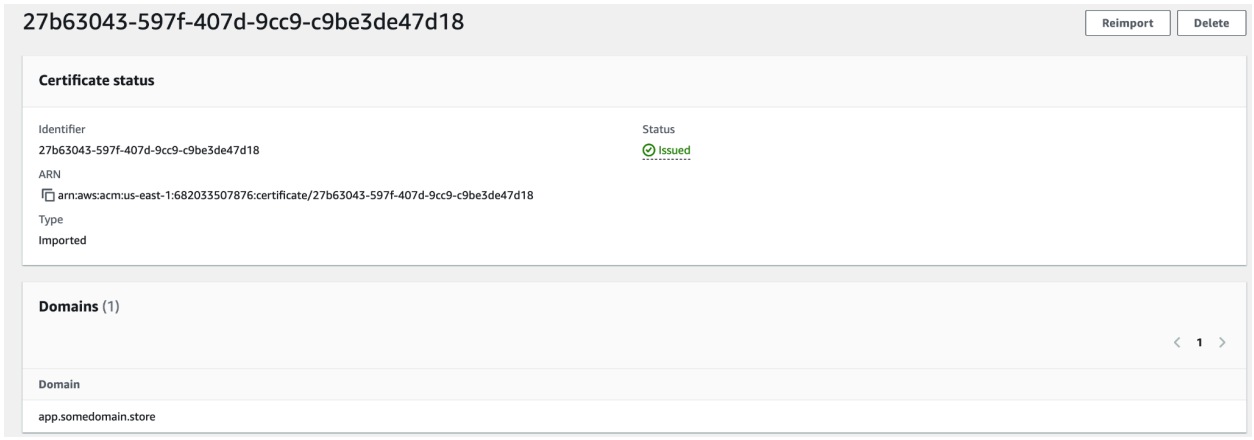


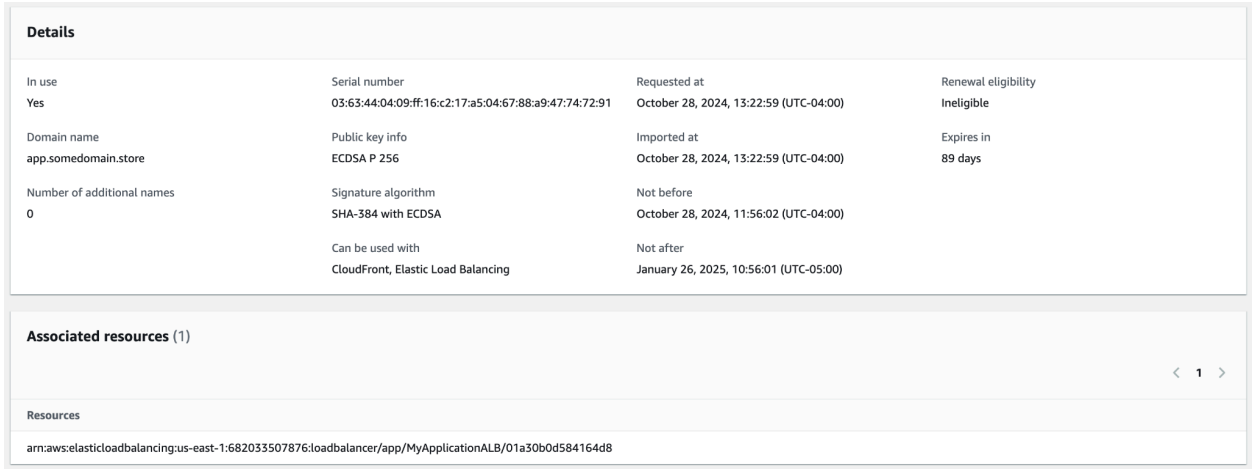
- **Test:** We conducted a proxy SQL injection test on the platform firewall using the command ' OR '1'='1' --. The attempt was successfully blocked by the AWSManagedRulesSQLiRuleSet in WAF.

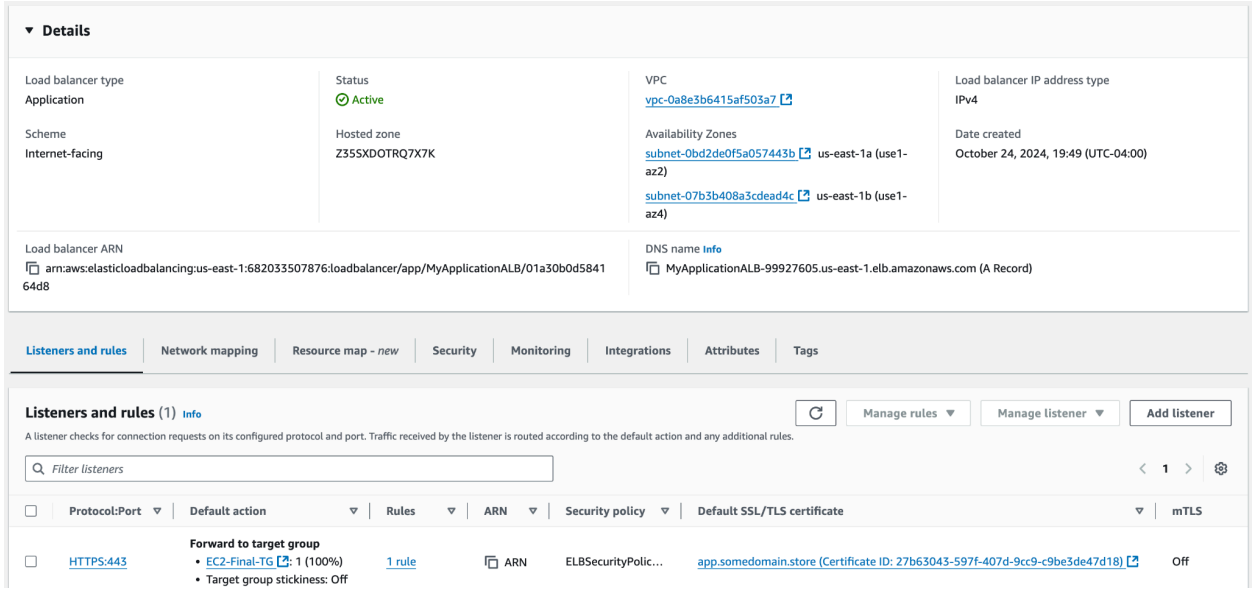


Ensuring a secure data transit through AWS Certificate Manager:

- **Interaction:**
  - ACM provides SSL certificates to the ALB, enabling HTTPS and encrypting data in transit between clients and the application.
  - ACM certificate is applied to the ALB to continue to provide SSL/TLS encryption, even as EC2 instances are scaled in and out.
- **Our Approach:**
  - A certificate was issued to our domain name (app.somedomain.store) and by AWS Certificate Manager.
  - Using ACM, a HTTPS listener was added to our load balancer to establish a secure entry point for all incoming traffic, which is crucial in protecting data and providing a trustworthy user experience.


27b63043-597f-407d-9cc9-c9be3de47d18
Reimport
Delete
Certificate status
Identifier
27b63043-597f-407d-9cc9-c9be3de47d18
Status
Issued
ARN
arn:aws:acm:us-east-1:682033507876:certificate/27b63043-597f-407d-9cc9-c9be3de47d18
Type
Imported
Domains (1)
1
Domain
app.somedomain.store


Details
In use
Yes
Serial number
03:63:44:04:09:ff:16:c2:17:a5:04:67:88:a9:47:74:72:91
Requested at
October 28, 2024, 13:22:59 (UTC-04:00)
Renewal eligibility
Ineligible
Domain name
app.somedomain.store
Public key info
ECDSA P 256
Imported at
October 28, 2024, 13:22:59 (UTC-04:00)
Expires in
89 days
Number of additional names
0
Signature algorithm
SHA-384 with ECDSA
Not before
October 28, 2024, 11:56:02 (UTC-04:00)
Can be used with
CloudFront, Elastic Load Balancing
Not after
January 26, 2025, 10:56:01 (UTC-05:00)
Associated resources (1)
1
Resources
arn:aws:elasticloadbalancing:us-east-1:682033507876:loadbalancer/app/MyApplicationALB/01a30b0d584164d8


Details
Load balancer type
Application
Status
Active
VPC
vpc-0a8e3b6415af503a7
Load balancer IP address type
IPv4
Scheme
Internet-facing
Hosted zone
Z35SXDOTRQ7X7K
Availability Zones
subnet-0bd2de0f5a057443b us-east-1a (use1-az2)
subnet-07b3b408a3cdead4c us-east-1b (use1-az4)
Date created
October 24, 2024, 19:49 (UTC-04:00)
Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:682033507876:loadbalancer/app/MyApplicationALB/01a30b0d584164d8
DNS name Info
MyApplicationALB-99927605.us-east-1.elb.amazonaws.com (A Record)
Listeners and rules
Network mapping
Resource map - new
Security
Monitoring
Integrations
Attributes
Tags
Listeners and rules (1) Info
A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.
Manage rules
Manage listener
Add listener
Filter listeners
1
Protocol:Port
Default action
Rules
ARN
Security policy
Default SSL/TLS certificate
mTLS
HTTPS:443
Forward to target group
EC2-Final-TG: 1 (100%)
Target group stickiness: Off
1 rule
ARN
ELBSecurityPolic...
app.somedomain.store (Certificate ID: 27b63043-597f-407d-9cc9-c9be3de47d18)
Off

Ensuring a secure connection between RDS and EC2 instance through SSL

- **Interaction:**
  - Safeguards data exchanged between EC2 and RDS
  - Protects logins and payment processes
- **Our Approach:**
  - Obtained the Amazon RDS root certificate and saved it to the EC2 instance.
  - Connected to the RDS instance using the MySQL client with the SSL certificate.
  - Configured the MySQL Client to Use SSL

SSL enabled

```
ubuntu@ip-10-0-3-150:~$ mysql -h rds-multiaz-myrdsinstance-qeruumzj9v4d.ctiowscucz1b.us-east-1.rds.amazonaws.com -u dbadmin -p --ssl-ca=rds-ca-cert.pem --ssl-mode=VERIFY_CA
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 999088
Server version: 8.0.39 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW STATUS LIKE 'Ssl_cipher';
+---------------+------------------------+
| Variable_name | Value                  |
+---------------+------------------------+
| Ssl_cipher    | TLS_AES_256_GCM_SHA384 |
+---------------+------------------------+
1 row in set (0.03 sec)

mysql> SHOW VARIABLES LIKE '%ssl%';
+---------------------------------------+---------------------------------------+
| Variable_name                         | Value                                 |
+---------------------------------------+---------------------------------------+
| admin_ssl_ca                          |                                       |
| admin_ssl_capath                      |                                       |
| admin_ssl_cert                        |                                       |
| admin_ssl_cipher                      |                                       |
| admin_ssl_crl                         |                                       |
| admin_ssl_crlpath                     |                                       |
| admin_ssl_key                         |                                       |
| have_openssl                          | YES                                   |
| have_ssl                              | YES                                   |
| performance_schema_show_processlist | OFF                                   |
| ssl_ca                                | /rdsdbdata/rds-metadata/ca-cert.pem   |
| ssl_capath                            |                                       |
| ssl_cert                              | /rdsdbdata/rds-metadata/server-cert.pem |
| ssl_cipher                            |                                       |
| ssl_crl                               |                                       |
| ssl_crlpath                           |                                       |
| ssl_fips_mode                         | OFF                                   |
| ssl_key                               | /rdsdbdata/rds-metadata/server-key.pem |
| ssl_session_cache_mode                | ON                                    |
| ssl_session_cache_timeout             | 300                                   |
+---------------------------------------+---------------------------------------+
20 rows in set (0.01 sec)
```

Cannot access DB without SSL

```
ubuntu@ip-10-0-3-150:~$ mysql -h rds-multiaz-myrdsinstance-qeruumzj9v4d.ctiowscucz1b.us-east-1.rds.amazonaws.com -u dbadmin -p --ssl-mode=DISABLED
Enter password:
ERROR 3159 (HY000): Connections using insecure transport are prohibited while --require_secure_transport=ON.
ubuntu@ip-10-0-3-150:~$
```

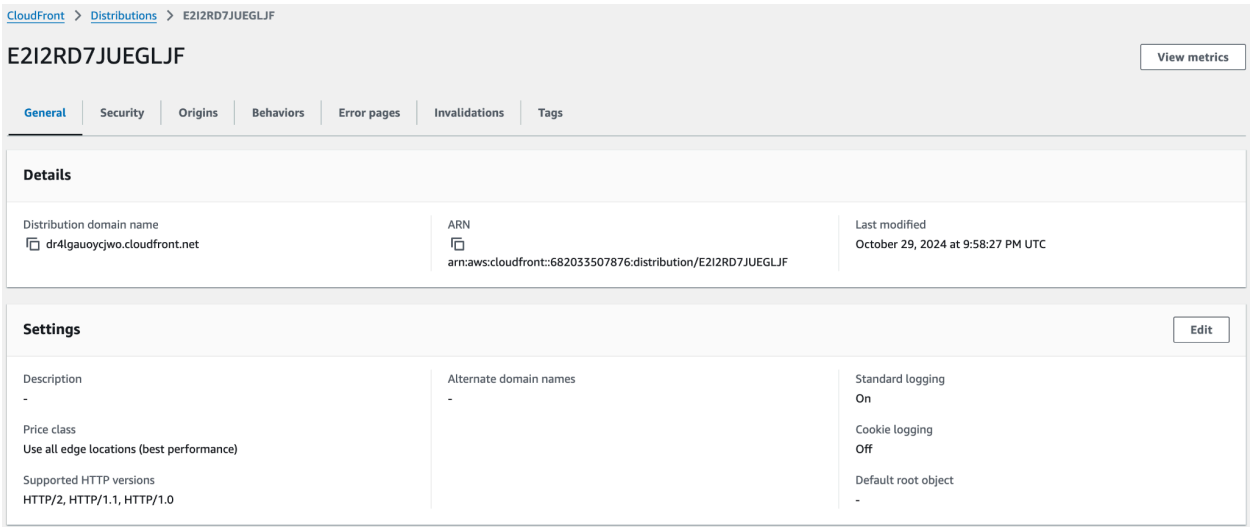
CONTENT DELIVERY AND PERFORMANCE OPTIMIZATION - PHASE 3

Introduction:

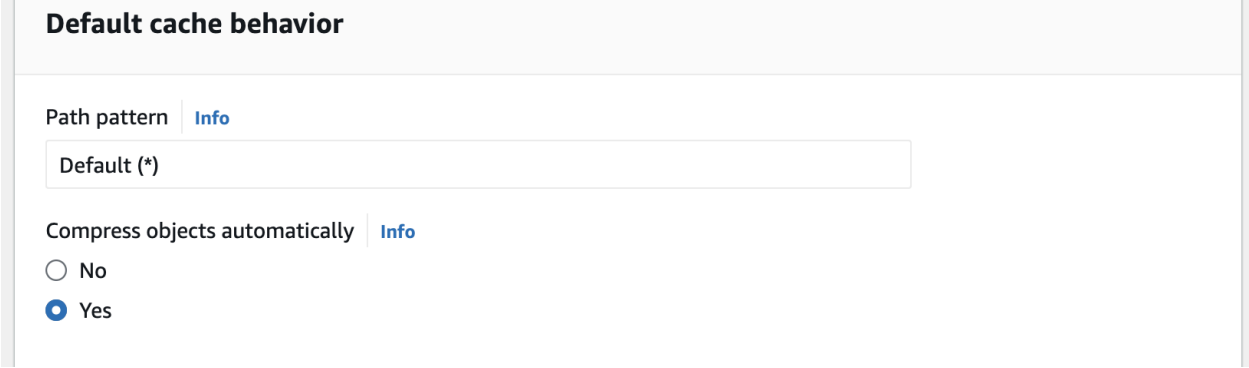
- The main objective of this phase is to improve the user experience of the website by reducing latency and increasing content delivery globally.
- Amazon Cloudfront and S3 are the key resources leveraged here to optimize performance and improve content delivery quickly.
- We will also be compressing objects and cache images to improve the load time and reduce latency.

Amazon CloudFront:

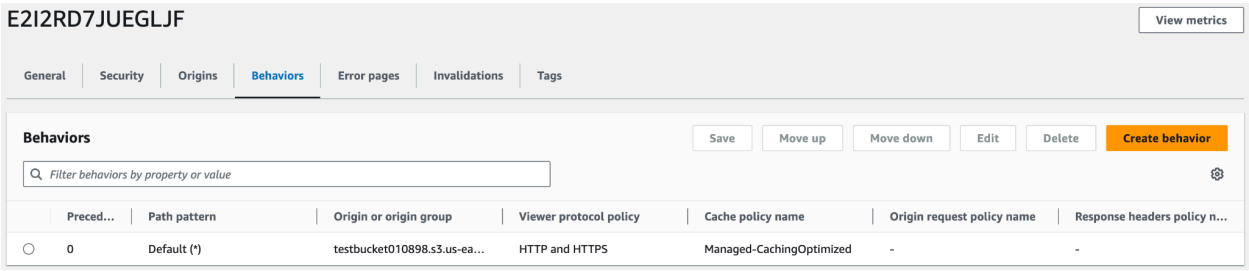
- **Interaction:**
  - As a content delivery network (CDN), AWS CloudFront distributes our applications' dynamic content such as images, and files that are used on the website, at a cached location on the edge, making it widely available for my users.
  - Engages with AWS services like Amazon EC2, Amazon S3, and Elastic Load Balancing to dynamically load website content for user traffic.
- **Our Approach:**
  - Implemented Amazon CloudFront to serve static content (like images), reducing latency by caching these assets closer to users around the world.



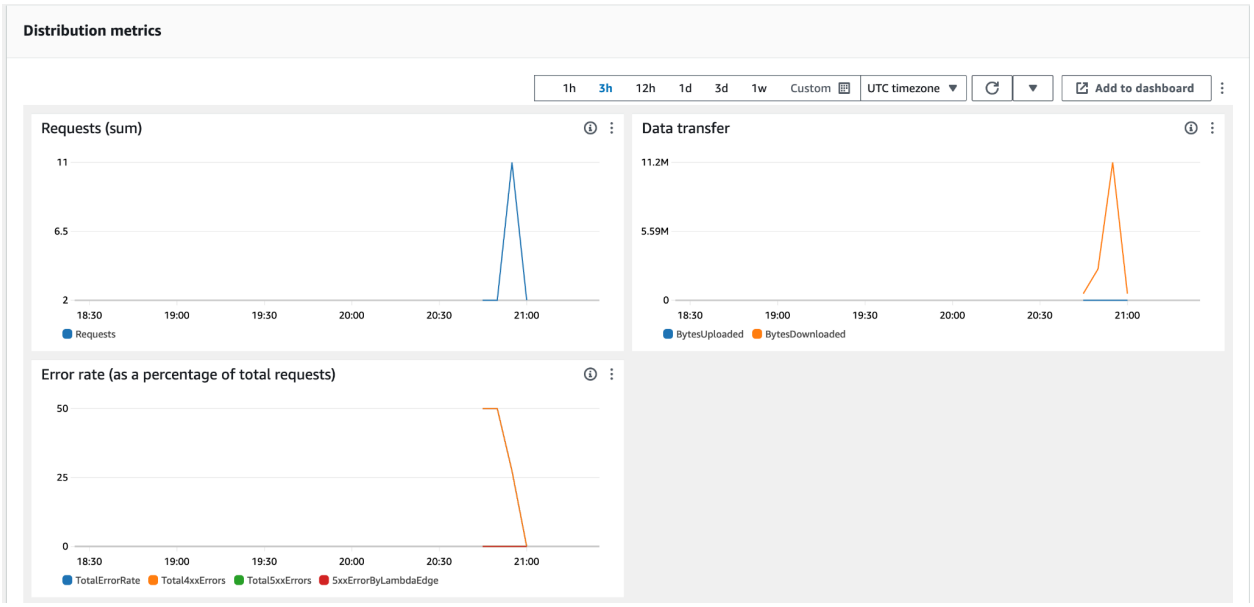
- Enabled gzip compression to reduce the size of images transferred over the network, thereby reducing the load times and costs for data transfer.



- Added the policy “Managed-CachingOptimized” in the behavior which caches all objects to maintain efficient performance and reduce the EC2 load by providing the frequently accessed images directly from the cache.

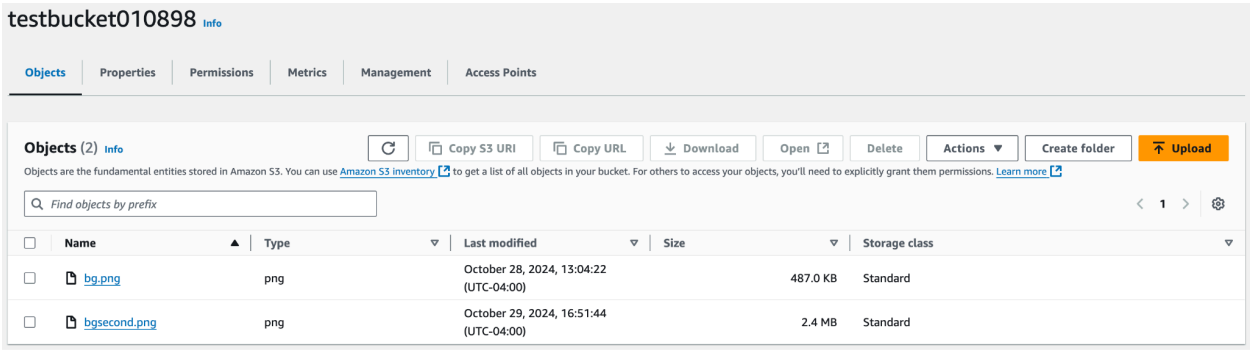


- Amazon Cloudwatch is integrated here as well to track the key performance metrics and adjust the configuration based on the performance



Amazon S3 (Simple Storage Service):

- **Function:**
  - Supplies object storage designed to efficiently store and retrieve vast amounts of data from various sources such as websites and mobile applications.
- **Our Approach:**
  - We set up a S3 Bucket to hold the static assets (like images) from our platform. The S3 was then connected to the cloud front to distribute and cache the images.



Static content delivery through cloudfront & S3:

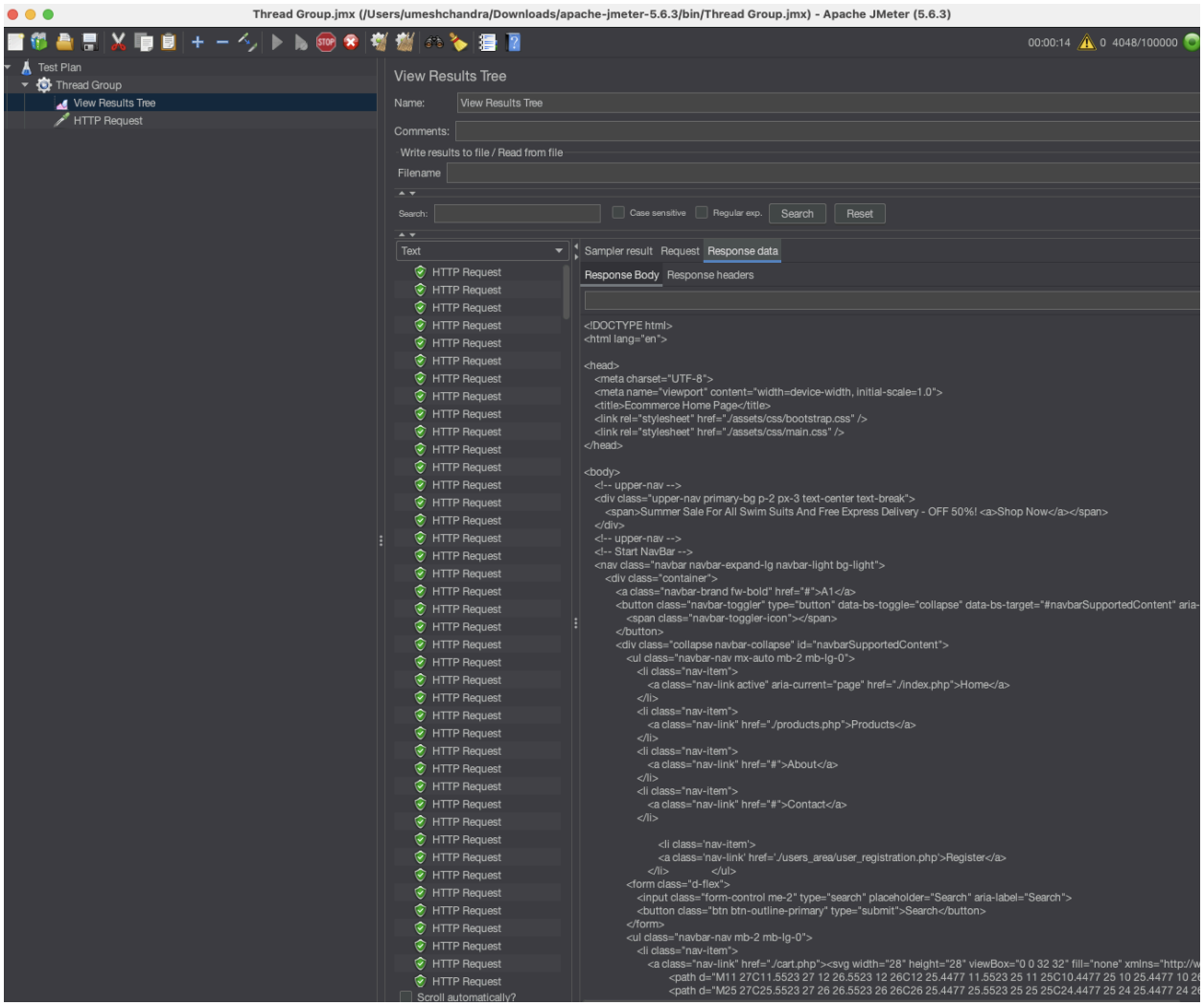
```
/* End NavBar */
/*  ********************
    **** Index.php ****
    ********************
*/
/* Start Landing Section  */
.landing .container .row .cover {
    background-image: url("https://dr4lgauoycjwo.cloudfront.net/bg.png");
    background-size: cover;
    height: 100%;
    display: flex;
    flex-direction: column;
    gap: 18px;
    align-items: flex-start;
    padding: 40px;
    color: white;
}
```

```
/* End Category Section  */
/* Start Advertise Section */
.adver {
    padding-top: 20px;
    padding-bottom: 40px;
    margin-bottom: 20px;
}

.adver .container .cover {
    background-image: url("https://dr4lgauoycjwo.cloudfront.net/bgsecond.png");
    background-position: center;
    background-size: cover;
    height: 60vh;
    display: flex;
    flex-direction: column;
    gap: 25px;
    align-items: flex-start;
    justify-content: center;
    color: white;
    padding: 70px;
}
```

TESTING AND MONITORING - PHASE 4

Stress Testing

- **Intention:**
    - To validate the application's resilience under heavy user load and assess the effectiveness of our Auto Scaling and RDS configurations.
    - To ensure the infrastructure can dynamically handle traffic spikes without degradation in performance.
- **Our Approach:**
    - We employed Apache JMeter to try simulating the real-world load conditions, and aimed for approximately 1 million requests to the target page.
    - Once the requests were sent key performance metrics, such as CPU utilization, memory usage, number requests were monitored to gauge the system's performance in handling load.
    - CloudWatch alarms were also setup to actively monitor the CPU utilization


Thread Group.jmx (/Users/umeshchandra/Downloads/apache-jmeter-5.6.3/bin/Thread Group.jmx) - Apache JMeter (5.6.3)
00:00:14 0 4048/100000
Test Plan
Thread Group
View Results Tree
HTTP Request
View Results Tree
Name: View Results Tree
Comments:
Write results to file / Read from file
Filename
Search: Case sensitive Regular exp. Search Reset
Text
Sampler result Request Response data
Response Body Response headers
HTTP Request
Scroll automatically?
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Ecommerce Home Page</title>
<link rel="stylesheet" href="./assets/css/bootstrap.css" />
<link rel="stylesheet" href="./assets/css/main.css" />
</head>
<body>
<!-- upper-nav -->

Summer Sale For All Swim Suits And Free Express Delivery - OFF 50%! <a>Shop Now

<!-- upper-nav -->
<!-- Start NavBar -->
<nav class="navbar navbar-expand-lg navbar-light bg-light">

A1
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-

</button>

<ul class="navbar-nav mx-auto mb-2 mb-lg-0">
<li class="nav-item">
Home

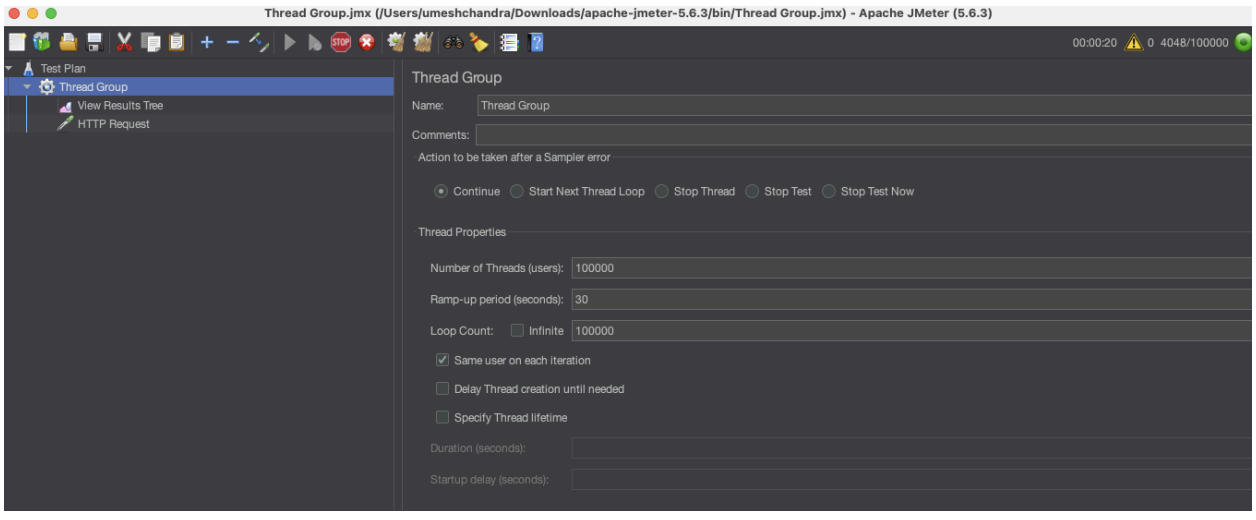
<li class="nav-item">
Products

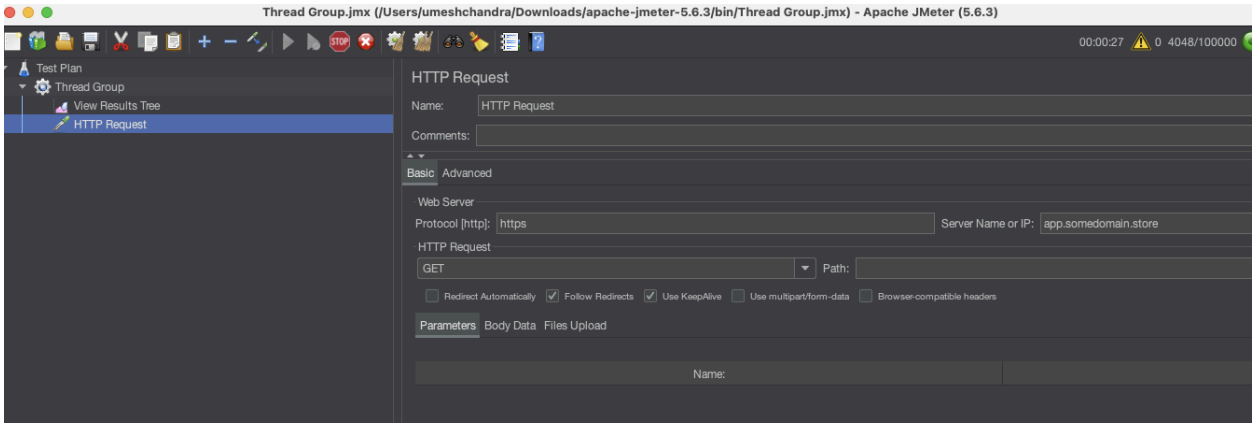
<li class="nav-item">
About

<li class="nav-item">
Contact

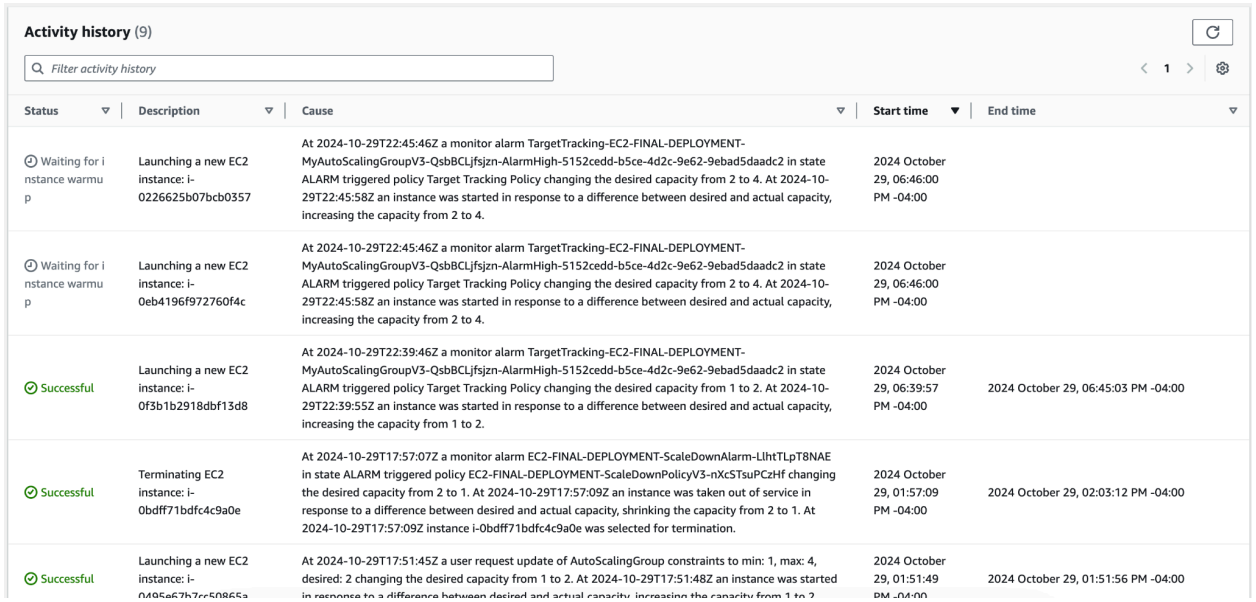
<li class='nav-item'>
Register

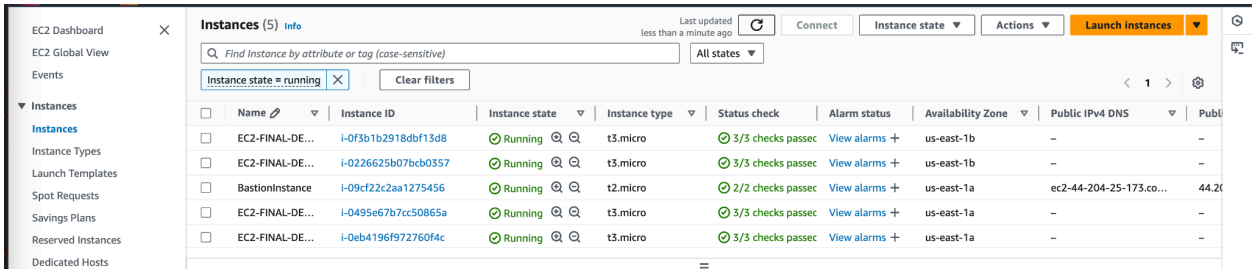
<form class="d-flex">
<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
<button class="btn btn-outline-primary" type="submit">Search</button>
</form>
<ul class="navbar-nav mb-2 mb-lg-0">
<li class="nav-item">
<a class="nav-link" href="./cart.php"><svg width="28" height="28" viewBox="0 0 32 32" fill="none" xmlns="http://w
<path d="M11 27C11.5523 27 12 26.5523 12 26C12 25.4477 11.5523 25 11 25C10.4477 25 10 25.4477 10 26
<path d="M25 27C25.5523 27 26 26.5523 26 26C26 25.4477 25.5523 25 25 25C24.4477 25 24 25.4477 24 26

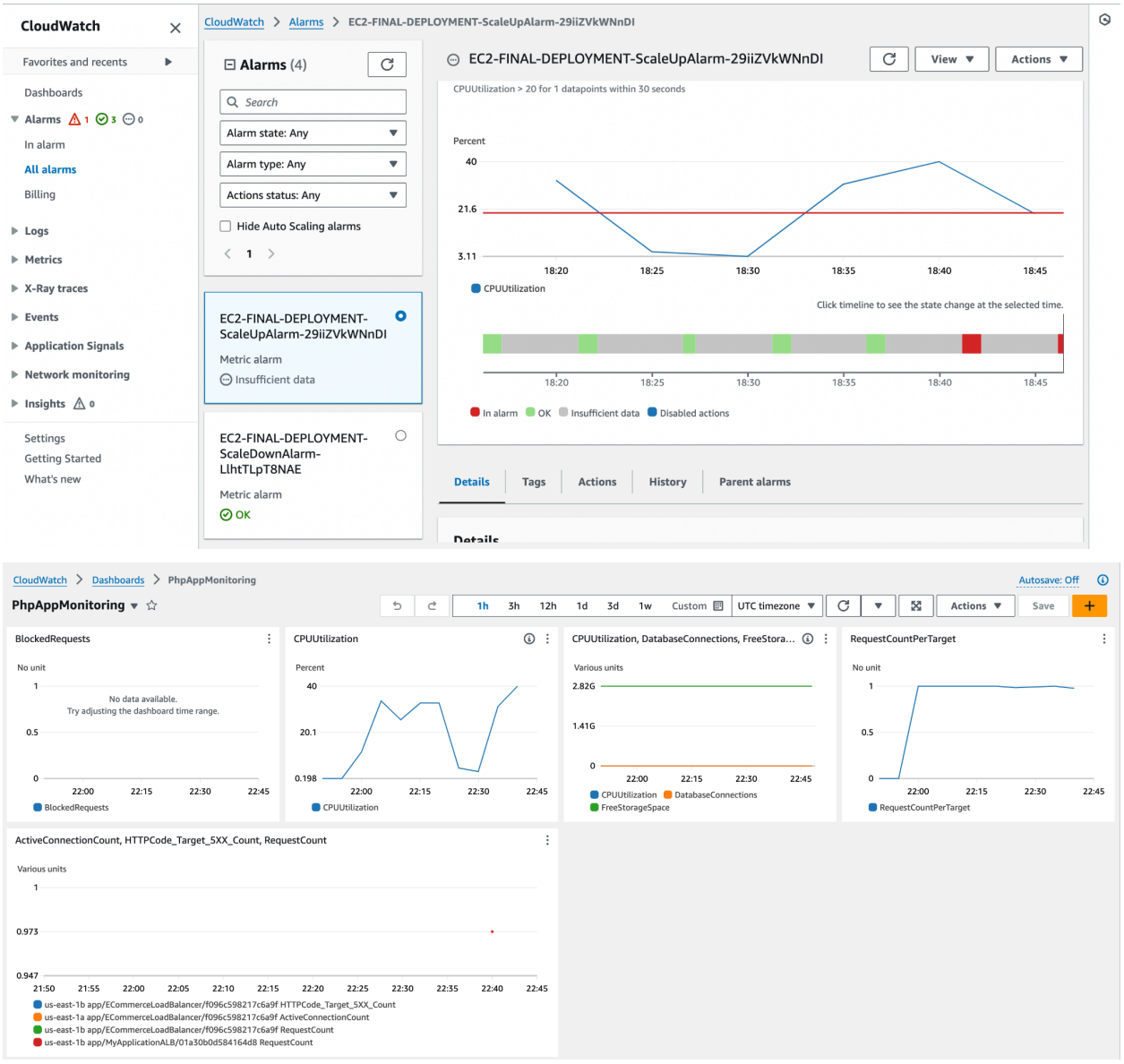

Thread Group.jmx (/Users/umeshchandra/Downloads/apache-jmeter-5.6.3/bin/Thread Group.jmx) - Apache JMeter (5.6.3)
00:00:20 0 4048/100000
Test Plan
Thread Group
View Results Tree
HTTP Request
Thread Group
Name: Thread Group
Comments:
Action to be taken after a Sampler error
Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now
Thread Properties
Number of Threads (users): 100000
Ramp-up period (seconds): 30
Loop Count: Infinite 100000
Same user on each iteration
Delay Thread creation until needed
Specify Thread lifetime
Duration (seconds):
Startup delay (seconds):



- During the bombardment of requests when the CPU utilization has crossed 20% the number of EC2 instances has scaled up and once the CPU utilization has gone less than 10% the number EC2 instances has also scaled down.
- Screenshot showing Autoscaled Ec2 instances when subjected to more load.





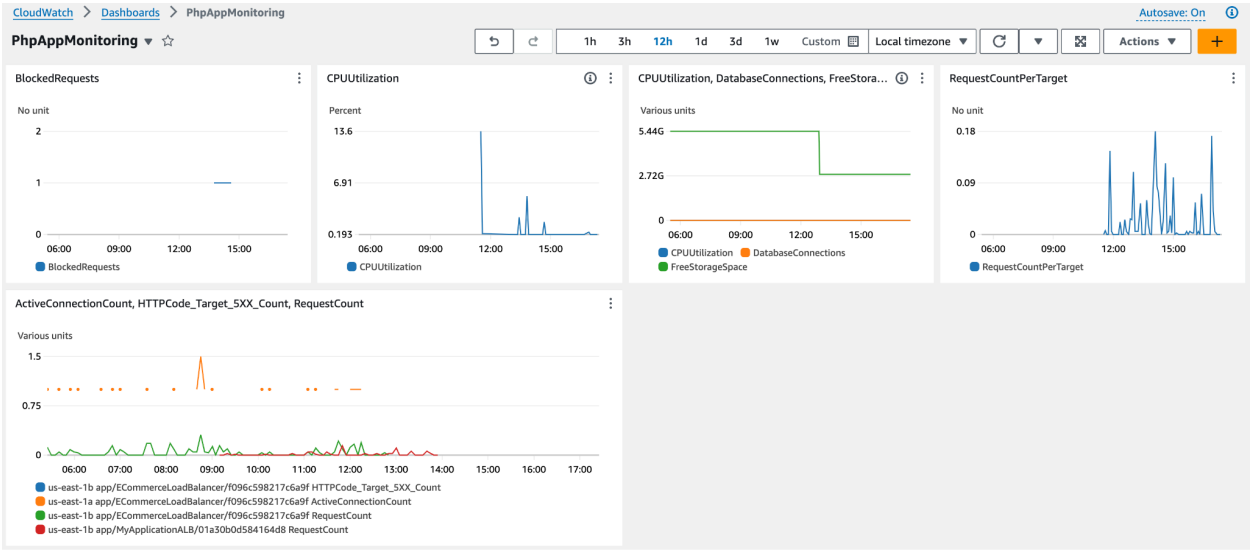


Monitoring Setup

AWS CloudWatch

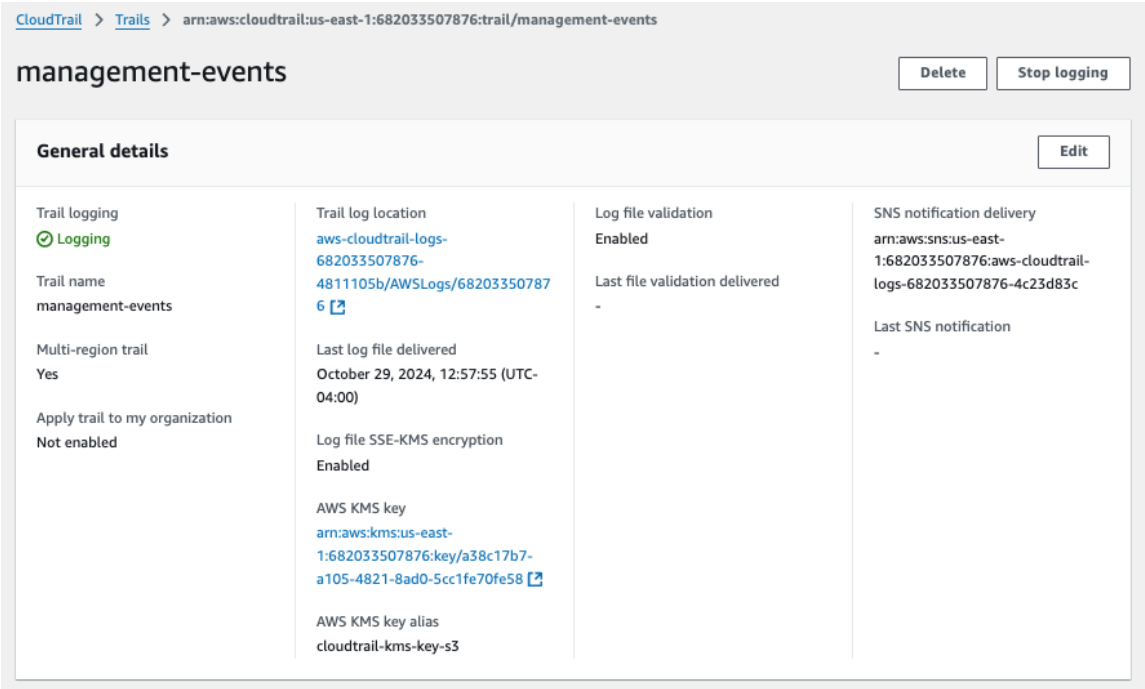
- **Intention:**
  - Collects the monitoring metrics such as disk space, and health of the app
  - CloudWatch collects and stores log files for errors, debugging, and other types of logging
  - Can set up alarms for triggering if a metric crosses a certain threshold
- **Our Approach:**

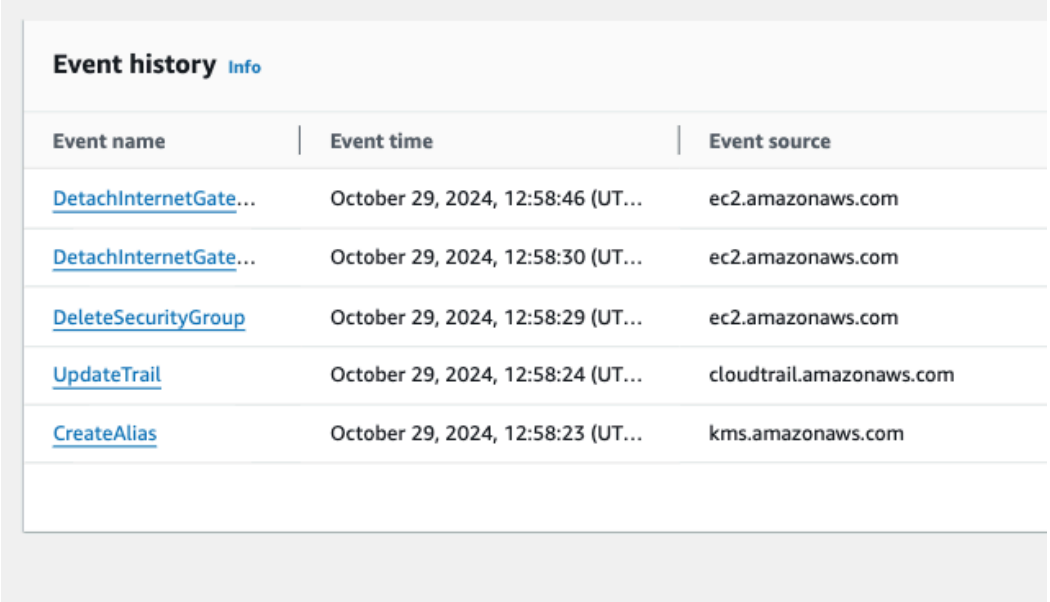
- We set up the CloudWatch dashboard to see the CPU utilization, amount of Requests Per Instance, any blocked requests, and how many active connections there are per availability zone

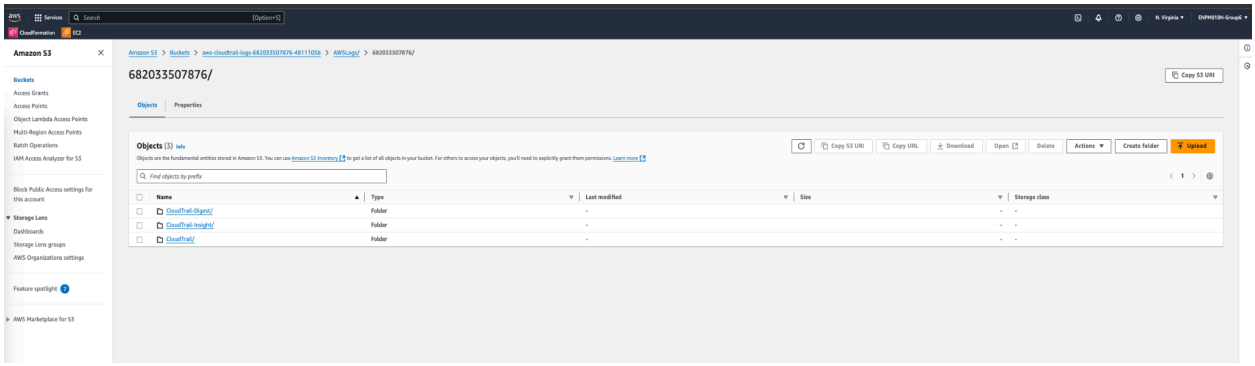


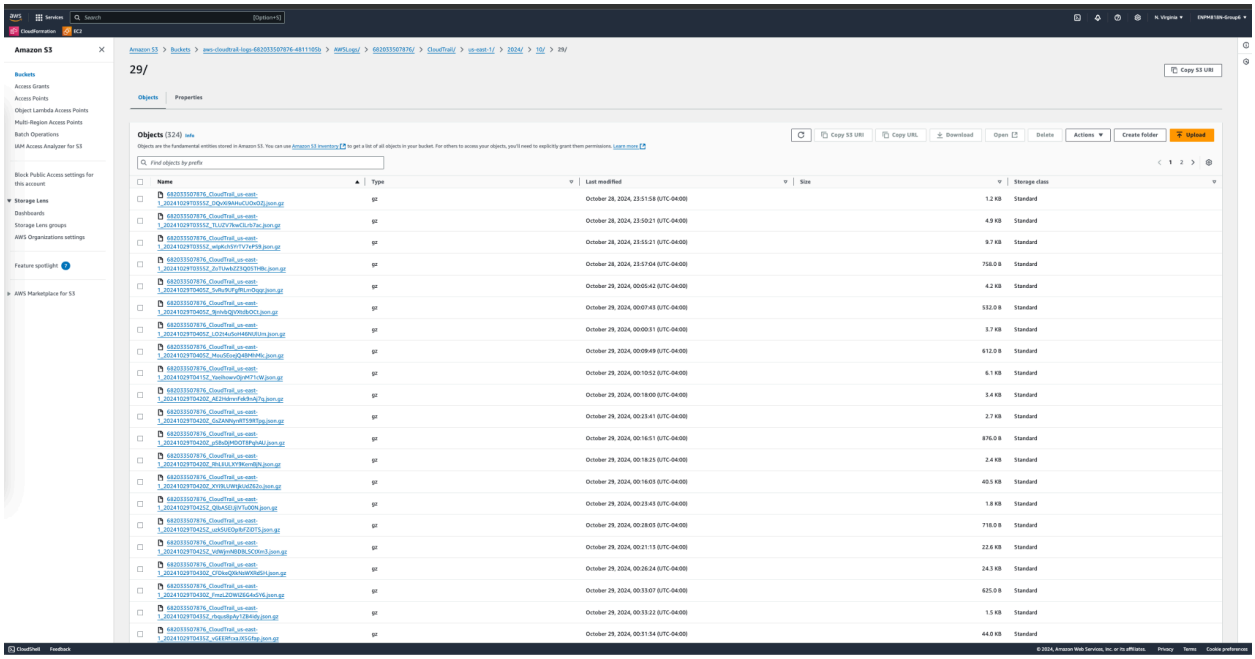
CloudTrail

- **Intention:**
  - CloudTrail allows for the event logging of account activity by users in the form of an event history log file
  - These are conducted through API calls within the past 90 days of the console
- **Our Approach:**
  - We set up CloudTrail to monitor the logs within the console and maintain visibility into the actions taken within our infrastructure
  - We set up CloudTrail Insights to alert us on any unusual activity that is detected and enabled KMS encryption for added security
  - The logs are stored in a new S3 bucket that we created and are set to alert us via SNS email delivery for each log file
  - We enabled Log File validation to ensure SHA-256 is used for each log file delivery


CloudTrail > Trails > arn:aws:cloudtrail:us-east-1:682033507876:trail/management-events
management-events
Delete
Stop logging
General details
Edit
Trail logging
Logging
Trail name
management-events
Multi-region trail
Yes
Apply trail to my organization
Not enabled
Trail log location
aws-cloudtrail-logs-682033507876-4811105b/AWSLogs/682033507876
Last log file delivered
October 29, 2024, 12:57:55 (UTC-04:00)
Log file SSE-KMS encryption
Enabled
AWS KMS key
arn:aws:kms:us-east-1:682033507876:key/a38c17b7-a105-4821-8ad0-5cc1fe70fe58
AWS KMS key alias
cloudtrail-kms-key-s3
Log file validation
Enabled
Last file validation delivered
-
SNS notification delivery
arn:aws:sns:us-east-1:682033507876:aws-cloudtrail-logs-682033507876-4c23d83c
Last SNS notification
-

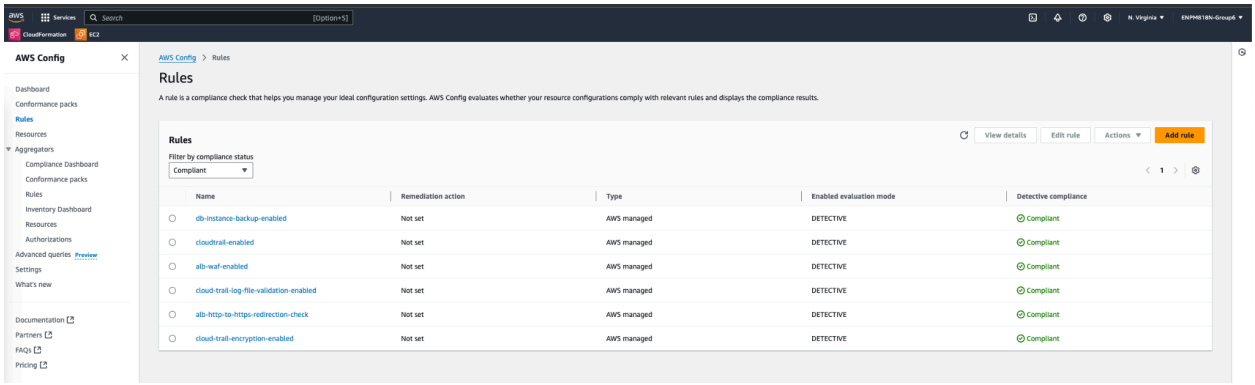

Event history Info
Event name
Event time
Event source
DetachInternetGate...
October 29, 2024, 12:58:46 (UT...
ec2.amazonaws.com
DetachInternetGate...
October 29, 2024, 12:58:30 (UT...
ec2.amazonaws.com
DeleteSecurityGroup
October 29, 2024, 12:58:29 (UT...
ec2.amazonaws.com
UpdateTrail
October 29, 2024, 12:58:24 (UT...
cloudtrail.amazonaws.com
CreateAlias
October 29, 2024, 12:58:23 (UT...
kms.amazonaws.com

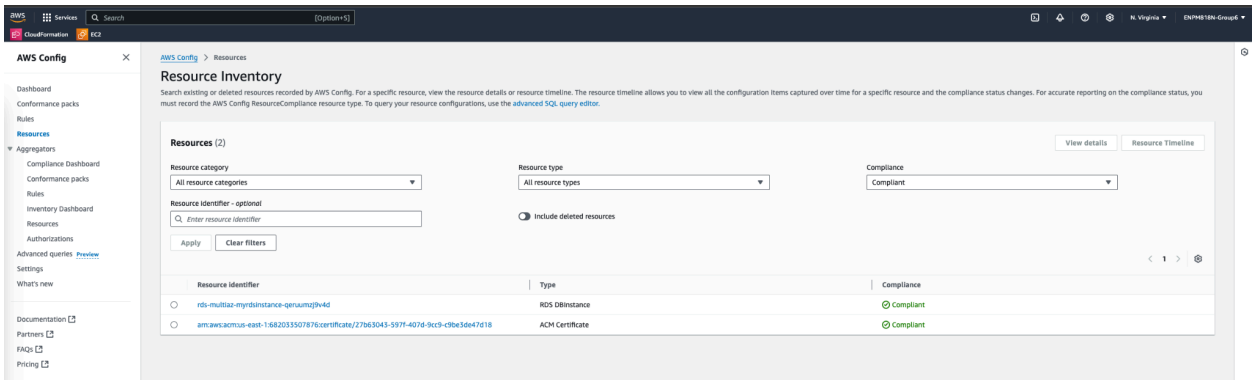


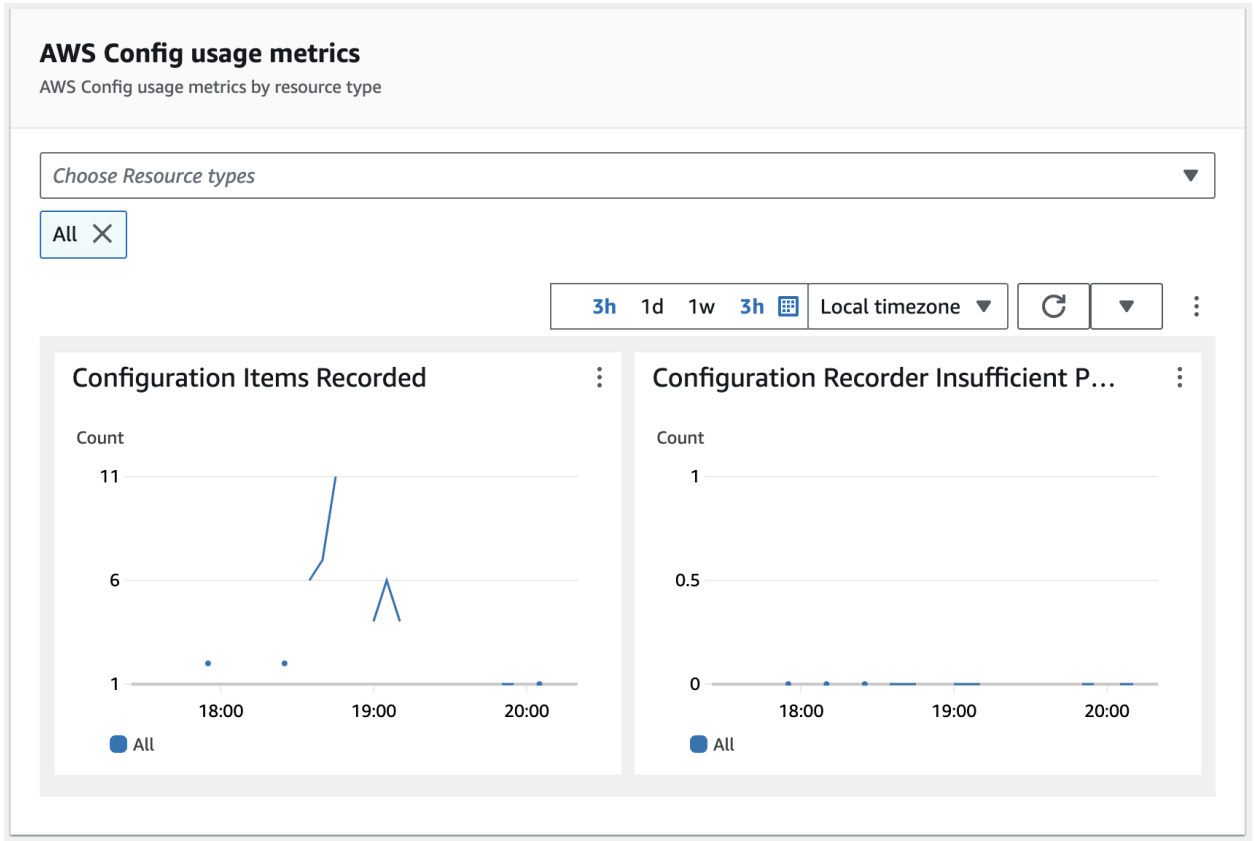


AWS Config

- **Intention:**
  - AWS Config tracks and records changes in settings for AWS resources such as EC2 instances and RDS Databases
  - It also conducts compliance monitoring to define rules and then check against those rules to see if the resources comply with the predefined rules
- **Our Approach:**
  - We set up AWS Config with predefined rules to test against our set up and see if we have everything set up correctly
  - Some of the rules include: Checking if WAF is enabled in our ALB, checking if our DB has backups enabled, checking cloudtrail enablement, and making sure the SSL certificate is not expired.

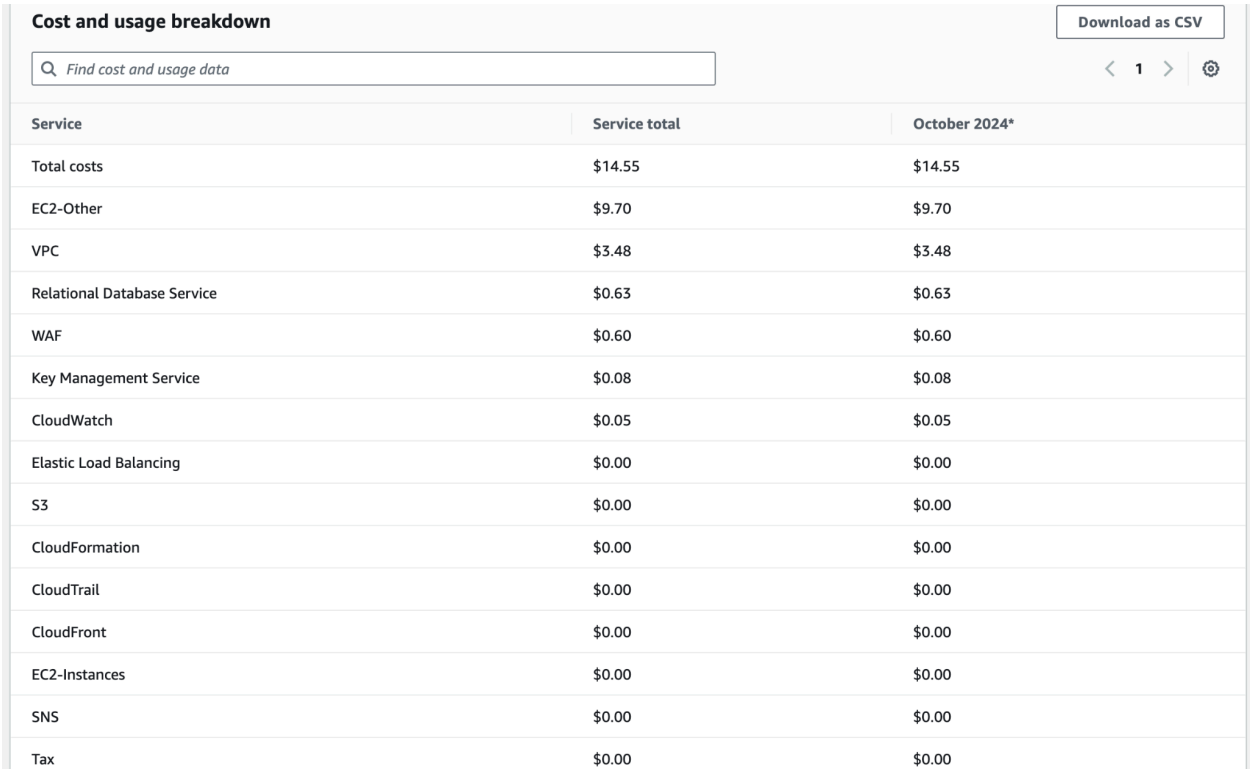

AWS Config
Rules
A rule is a compliance check that helps you manage your ideal configuration settings. AWS Config evaluates whether your resource configurations comply with relevant rules and displays the compliance results.
Add rule
db-instance-backup-enabled
cloudtrail-enabled
alb-waf-enabled
cloud-trail-log-file-validation-enabled
alb-http-to-https-redirection-check
cloud-trail-encryption-enabled


AWS Config
Resource Inventory
Resources (2)
Resource category
Resource type
Compliance
Resource identifier - optional
Include deleted resources
Apply
Clear filters


AWS Config usage metrics
AWS Config usage metrics by resource type
Choose Resource types
All
3h
1d
1w
3h
Local timezone
Configuration Items Recorded
Count
11
6
1
18:00
19:00
20:00
All
Configuration Recorder Insufficient P...
Count
1
0.5
0
18:00
19:00
20:00
All

Cost Optimization Analysis:

- **Intention:**
  - To analyze resource usage and explore options to reduce long-term costs, ensuring efficient allocation and potential savings.
- **Our Approach:**
  - Leveraged AWS Cost Explorer to assess current spending trends and identify opportunities for cost reductions.



Cost and usage breakdown

Download as CSV

Find cost and usage data

1

Service	Service total	October 2024*
Total costs	$14.55	$14.55
EC2-Other	$9.70	$9.70
VPC	$3.48	$3.48
Relational Database Service	$0.63	$0.63
WAF	$0.60	$0.60
Key Management Service	$0.08	$0.08
CloudWatch	$0.05	$0.05
Elastic Load Balancing	$0.00	$0.00
S3	$0.00	$0.00
CloudFormation	$0.00	$0.00
CloudTrail	$0.00	$0.00
CloudFront	$0.00	$0.00
EC2-Instances	$0.00	$0.00
SNS	$0.00	$0.00
Tax	$0.00	$0.00

Recommendations:

Reserved Instances (RIs):

- RIs are suitable for workloads when we know the specific instance types and regions that the project will consistently require over long periods of time. RIs are useful for predictable costs and stable workloads as they come with substantial appointments.
- For the stable resources like RDS Instance, standard RIs can be used because of the targeted discounts, thereby offering the best cost savings.

Saving Plans:

- They are more flexible alternatives than RI, and focus on reducing costs on various instances. Savings plan is more adaptable for dynamic environments where we may need to switch between regions.
- The Compute Savings Plan is likely the better choice for the dynamic EC2 components that run within Auto Scaling groups, as it provides flexibility and significant cost savings.

Lessons Learned:

Working on the Scalable and Secure E-commerce Platform project on AWS gave our team some valuable hands-on experience in building a secure, flexible, and efficient cloud infrastructure. Setting up Auto Scaling for EC2 instances was an eye-opener, teaching us how to handle traffic spikes without overspending on resources. Tackling the RDS setup together helped us navigate database management challenges, especially around encryption for data protection. Implementing the Web Application Firewall (WAF) showed us practical ways to safeguard against common web attacks, while configuring CloudFront highlighted how global content delivery can improve user experience through caching and faster load times. Beyond the technical skills we gained with core AWS services, this project underscored how essential teamwork is when developing secure, scalable cloud solutions.

Potential improvements for production environments:

To make our e-commerce platform even more robust and secure for a production environment, we could fine-tune configurations and incorporate key AWS services. First, we could implement Multi-Regional Deployment, which would add a secondary region for disaster recovery. This setup would replicate critical data and services, ensuring high availability and quick failover in case of regional issues. It would also reduce latency for users on the West Coast for example, by offering a region closer to them.

Since we use S3 buckets for functions like storing images on CloudFront, logging with CloudTrail, and AWS Config, we can enable automatic backups for these buckets to prevent data loss. This would also allow us to implement versioning for log files and move older files to lower-cost storage, helping manage expenses. Although we avoided this in our sample project to keep costs down, it would be essential in a real-world deployment.

Additionally, we could add Network Access Control Lists (NACLs) at the subnet level to secure our inbound and outbound traffic further. Finally, implementing AWS ElastiCache would introduce a caching layer for frequently accessed data, significantly improving database load times.

REFERENCES

Core Infrastructure Concepts

- Global Infrastructure - Overview
- AWS Resource Management Documentation
- AWS Well-Architected Framework
- Organizing Your AWS Environment Using Multiple Accounts
- Design Isolated Resource Environments
- AWS Identity and Access Management (IAM)
- AWS Systems Manager
- AWS Service Catalog

Infrastructure as Code (IaC)

- Choosing an Infrastructure as Code Tool
- Reviewing IaC Tools for the AWS Cloud
- AWS Cloud Development Kit (AWS CDK)
- AWS Serverless Application Model (AWS SAM)
- Using the AWS Infrastructure Composer Console
- What is AWS Infrastructure Composer?
- Infrastructure Composer Enhanced Component Cards

Infrastructure Management Tools

- AWS Config
- AWS Trusted Advisor
- AWS Organizations
- AWS Cost Management
- AWS CloudTrail
- AWS CloudWatch

- AWS Config Rules

Security and Compliance

- Best Practices for Tagging AWS Resources
- AWS Key Management Service (KMS)
- AWS Security Hub
- AWS Shield and WAF
- AWS Secrets Manager
- AWS Identity and Access Management (IAM) Best Practices
- Security and Compliance Documentation
- AWS Compliance

Application and Deployment Services

- Amazon EC2 Auto Scaling
- AWS CodePipeline
- AWS CodeBuild
- AWS Step Functions
- Amazon ECS
- AWS Elastic Load Balancing (ELB)
- Amazon VPC