

AWS LAB Introduction

Sunday, 19 October 2025 10:01 am

The purpose of this project is to have hands on practice with AWS for better understanding and exposure to Cloud and AWS infrastructure and its inner workings.

In this LAB, the use case will be adopting AWS cloud services for a company's ecommerce website to allow for scalability as the business grows. Low latency and accessibility are also considered here in the choosing of AWS to host our webapp.

Below is a diagram of the cloud infrastructure

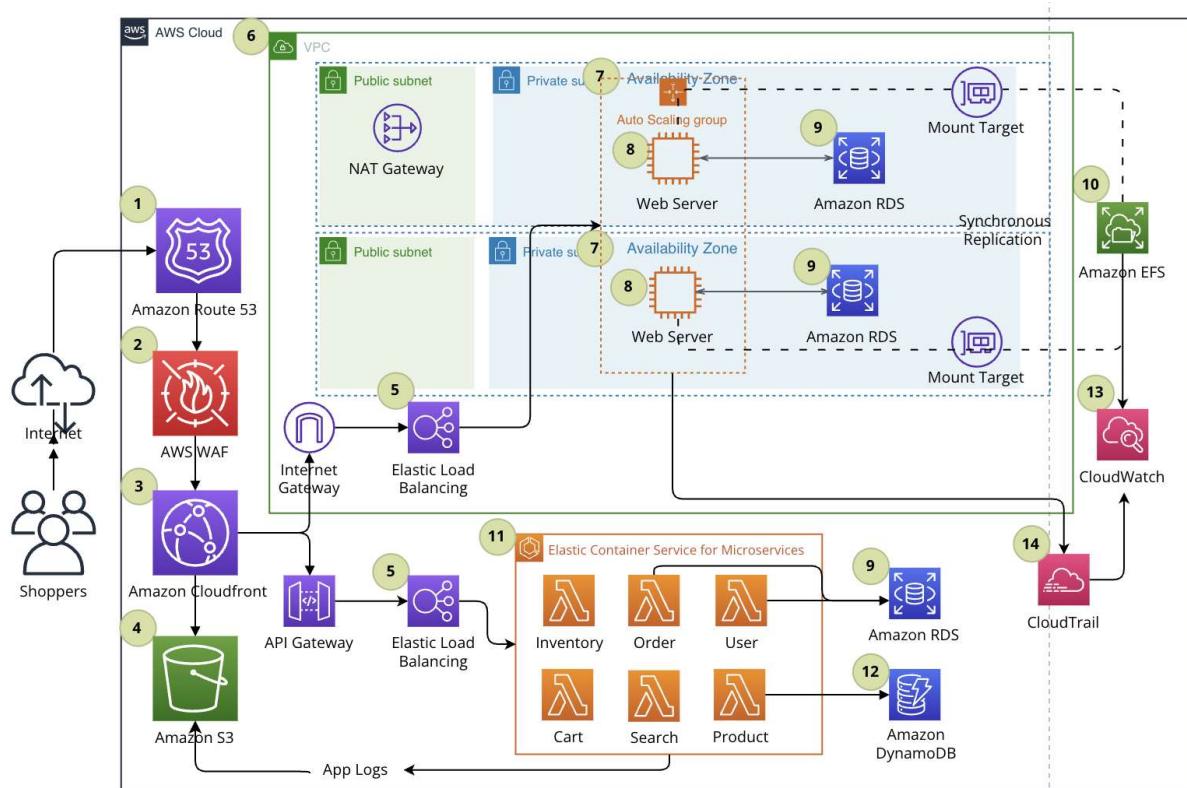


Figure 1: Cloud Infrastructure Diagram

1. Amazon Route 53: Routes shoppers DNS requests resolving domain name service (DNS) and provides global traffic management capabilities
2. AWS WAF: Web application firewall that protects the e-commerce website against common web exploits.
3. Amazon CloudFront: Provides a highly secure and programmable content delivery network (CDN) with edge locations around the globe. It can deliver dynamic content with low latency and high transfer speeds from locations close to the customer.
4. Amazon S3: Stores all static catalogue content, such as product images and videos, log files of the Microservices (#6) and clickstream information from Amazon CloudFront (#3).
5. Elastic Load Balancing (ELB): Distributes network traffic to improve the scalability and availability of web servers across multiple AZs.
6. Single AWS Region and single Amazon Virtual Private Cloud (Amazon VPC): Provide

- similar architecture as an on-premise data centre setup.
- 7. Multiple Availability Zones (AZs): Provide resilience and high availability for the production workload.
 - 8. Frontend application Web Server: The frontend application web server for the e-commerce website is deployed by AWS EC2 Auto Scaling, which automatically handles the details of capacity provisioning, load balancing, auto scaling, and application health monitoring. Elastic IP will be sitting on top of the EC2 instance which allows for IP failover. If the EC2 instance needs to be reconfigured, the Elastic IP and Route 53 will work together to ensure that the service is still available when the web server comes back up.
 - 9. Amazon RDS: To provide high availability, the user and orders databases are hosted redundantly on a multi-AZ (multi Availability Zone) deployment of Amazon Relational Database Service (Amazon RDS) within private subnets that are isolated from the public Internet.
 - 10. Amazon Elastic File System: Shared file storage mounted to instances in the Auto Scaling group.
 - 11. Microservices: Microservices hosted in Amazon Elastic Container Service that represent domain constructs such as products, carts, orders, inventory and users as well as search services.
 - 12. Amazon DynamoDB: Amazon DynamoDB is a fully-managed, high performance, NoSQL database service that is easy to set up, operate, and scale. It is used both as a session store for persistent session data, such as the shopping cart, and as the product database. Because DynamoDB does not have a schema, we have a great deal of flexibility in adding new product categories and attributes to the catalogue.
 - 13. Amazon CloudWatch: Used for application logging, monitoring, and alarms for dynamic scaling. When a performance threshold is breached, a CloudWatch alarm triggers an automatic scaling event that either scales out or scales in EC2 instances in the Auto Scaling group.
 - 14. Amazon CloudTrail: Used to track user activity and API usage i.e. logging the date, time, and identity of users accessing directory data. Logs from CloudTrail to be sent to CloudWatch for dynamic monitoring.

For demonstration purposes, I will be showing the setup and security hardening for cloud services 4, 6, 7, 8, 9, 13 and 14 labelled above.

VPC Set-Up

Sunday, 19 October 2025 10:06 am

The first step to cloud migration is setting up the Virtual Private Cloud (VPC) for the company. For demonstration, I will be showcasing the migration of our web server onto the cloud, with screenshots, as well as a link to the screen recording videos in 4.13.

Region : N. Virginia (us-east-1)

VPC Name: MRSH

IPv4 CIDR block: 192.168.254.0/24

The VPC settings are as follows with the creation of 2 public and 2 private subnets:

The screenshot shows the 'VPC settings' page for creating a new VPC. Key configurations include:

- Resources to create:** VPC and more
- Name tag auto-generation:** Auto-generate (selected), Name tag: MRSH
- IPv4 CIDR block:** 192.168.254.0/24 (256 IPs)
- IPv6 CIDR block:** No IPv6 CIDR block
- Tenancy:** Default
- Number of Availability Zones (AZs):** 2 (selected from 1, 2, 3)
- Number of public subnets:** 2 (selected from 0, 2, 4)
- Number of private subnets:** 2 (selected from 0, 2, 4)
- Public subnet CIDR blocks:** 192.168.254.0/26 (64 IPs), 192.168.254.64/26 (64 IPs)
- Private subnet CIDR blocks:** 192.168.254.128/26 (64 IPs), 192.168.254.192/26 (64 IPs)
- NAT gateways (\$):** In 1 AZ (selected from None, In 1 AZ, 1 per AZ)
- VPC endpoints:** S3 Gateway (selected from None, S3 Gateway)
- DNS options:** Enable DNS hostnames, Enable DNS resolution

Figure 2 & 3: VPC Creation Settings

Following the creation of the above subnets, we can check to see the route tables have been configured accordingly with Figures 2 & 3 showing the public route table MRSH-rtb-public containing a route to the target starting with "igw", which stands for Internet Gateway; thereby granting our public subnets direct

access to the internet.

rtb-0735820d408ed0300 / MRSH-rtb-public				
Details	Routes	Subnet associations	Edge associations	Route propagation
Routes (3)				
<input type="text"/> Filter routes				Both: <input type="button" value="Edit routes"/>
Destination	Target	Status	Propagated	< 1 > <input type="button"/>
0.0.0.0/0	igw-02easas3sef8200	● Active	No	

rtb-0735820d408ed0300 / MRSH-rtb-public				
Details	Routes	Subnet associations	Edge associations	Route propagation
Explicit subnet associations (2)				
<input type="text"/> Find subnet association				<input type="button" value="Edit subnet associations"/>
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	
MRSH-subnet-public1-us-east-1a	subnet-0d5c5eeb0277d0695	192.168.254.0/26	-	
MRSH-subnet-public2-us-east-1b	subnet-0e4af022b45c1543	192.168.254.64/26	-	

Figure 4 & 5: Public route table

Figures 6 & 7 show that both route tables MRSH-rtb-private1-us-east-1a and MRSH-rtb-private1-us-east-1b contain a route 0.0.0.0/0 connected to the NAT gateway, which grants our private subnets internet connection whilst keeping resources private.

rtb-0a6475a9a89692f1c / MRSH-rtb-private1-us-east-1a				
Details	Routes	Subnet associations	Edge associations	Route propagation
Routes (4)				
<input type="text"/> Filter routes				Both: <input type="button" value="Edit routes"/>
Destination	Target	Status	Propagated	< 1 > <input type="button"/>
pl-63a5400a	vpc-0051bc407b17e04dd	● Active	No	
0.0.0.0/0	nat-02695268ea3a4a1a9	● Active	No	

rtb-02c8e84885c4956f1 / MRSH-rtb-private2-us-east-1b				
Details	Routes	Subnet associations	Edge associations	Route propagation
Routes (4)				
<input type="text"/> Filter routes				Both: <input type="button" value="Edit routes"/>
Destination	Target	Status	Propagated	< 1 > <input type="button"/>
pl-63a5400a	vpc-0051bc407b17e04dd	● Active	No	
0.0.0.0/0	nat-02695268ea3a4a1a9	● Active	No	

Figures 6 & 7: Public route table

EC2 Instance - Web Server

Sunday, 19 October 2025 10:12 am

Web Server Security Group

Before launching the web server, I prepare a security group which acts as a virtual firewall for our EC2 Web Server instance, consisting of rules to monitor and control inbound and outbound traffic. In this case, as the VPC will contain the web server, it will need a security group to allow inbound and outbound HTTP and HTTPS access:

The screenshot shows the 'Create security group' wizard. The first step, 'Basic details', includes fields for the security group name ('Web Security Group'), a description ('Enable HTTP access'), and the VPC ('vpc-0fd39154c420c18a5 (MRSH-vpc)'). The second step, 'Inbound rules', contains one rule: 'Inbound rule 1' allowing HTTP (TCP port 80) from anywhere. A warning message states: 'Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' The third step, 'Outbound rules', contains one rule: 'Outbound rule 1' allowing all traffic to all destinations. A similar warning message is present. The final step, 'Tags - optional', shows no tags assigned. At the bottom right are 'Cancel' and 'Create security group' buttons.

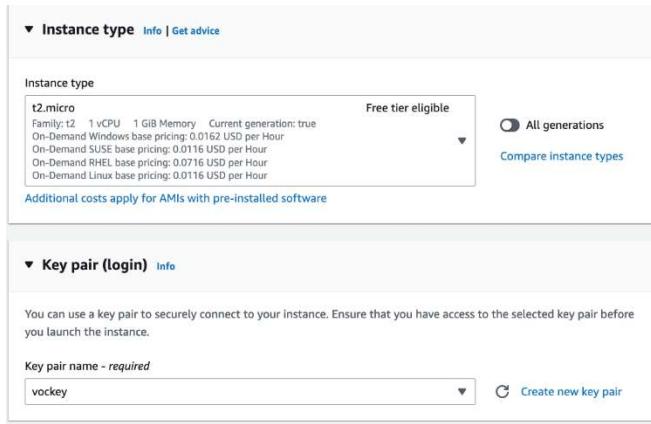
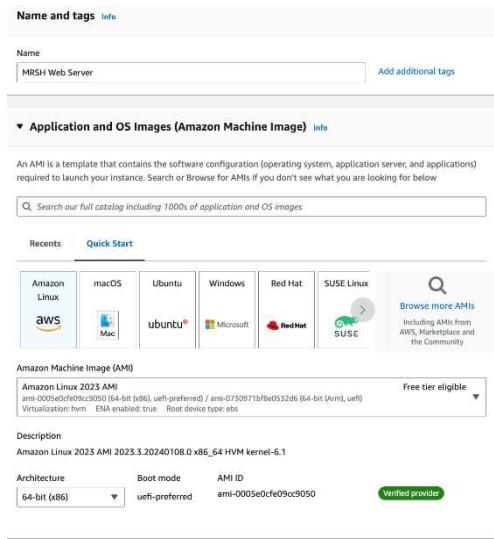
Figures 8 & 9: Web Server Security Group

Web Server Set-Up

In the next step, I launch an EC2 instance configured as a web server.

Name: MRSW Web Server

AMI: Amazon Linux 2023 AMI



Figures 10 & 11: Web Server Settings 1

In Figures 10 & 11, we connect our web server to our Web Security Group created in the VPC Set-Up section. And as shown in Figure 13, we insert the following script under the “User data” section:

```
#!/bin/bash
# Install Apache Web Server and PHP
dnf install -y httpd wget php mariadb105-server
# Download Lab files
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-ACCLFO-2/2-lab2-vpc/s3/lab-app.zip
unzip lab-app.zip -d /var/www/html/
# Turn on web server
chkconfig httpd on
service httpd start
```

The script above will run root user permissions on the instance’s guest OS automatically upon first launch. It will also install a web server (Apache), PHP libraries and PHP web application, and a database.

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0f039154c420c18a5 (MRSH-vpc)
192.168.254.0/24

Subnet [Info](#)

subnet-0e964928bd465ed0e MRSH-subnet-public2
VPC vpc-0f039154c420c18a5 Owner: 256232071561 Availability Zone: us-east-1b
IP addresses available: 57 CIDR: 192.168.254.64/26

Create new subnet [Info](#)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Common security groups [Info](#)

Select security groups [Info](#)

Web Security Group sg-04772095927399760 X
VPC vpc-0f039154c420c18a5

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

► Advanced network configuration

User data - optional [Info](#)

Upload a file with your user data or enter it in the field.

```
#!/bin/bash
# Install Apache Web Server and PHP
dnf install -y httpd wget php mariadb105-server
# Download Lab files
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-ACCLFO-2/2-lab2-vpc/s3/lab-app.zip
unzip lab-app.zip -d /var/www/html/
# Turn on web server
chkconfig httpd on
service httpd start
```

User data has already been base64 encoded

Figures 12 & 13: Web Server Settings 2

RDS Set-Up (DB)

Sunday, 19 October 2025 10:15 am

Amazon Relational Database Service (RDS)

DB Instance Set-Up

After launching our web server, we go onto the Amazon RDS console to create a database (DB) instance with the following settings shown above. The following table shows the comparison of Engine Types in Amazon RDS:

Engine Type	Use Cases	Strengths	Weaknesses	Typical Costs
Amazon Aurora MySQL/PostgreSQL	High-performance, scalable OLTP workloads	<ul style="list-style-type: none">• High throughput and low latency• Automatic scaling• Read replicas for high availability• Cost-effective	<ul style="list-style-type: none">• Not suitable for complex queries• Limited feature set compared to standard MySQL/PostgreSQL	Generally lower than standard MySQL/PostgreSQL on RDS due to automatic scaling capabilities
Standard MySQL/PostgreSQL	General-purpose workloads, web applications	<ul style="list-style-type: none">• Familiar engine for many developers• Wide range of features and tools• Flexible configuration options	<ul style="list-style-type: none">• May not be as scalable or performant as Aurora• Can be expensive for high-traffic applications	More expensive than Aurora, pricing varies depending on instance type and configuration
MariaDB	Open-source alternative to MySQL	<ul style="list-style-type: none">• High compatibility with MySQL• Lower cost than Amazon MySQL	<ul style="list-style-type: none">• Limited feature set compared to Amazon MySQL• Fewer tools and resources available	Generally slightly cheaper than Amazon MySQL on RDS
Oracle Database Enterprise Edition	Mission-critical enterprise applications	<ul style="list-style-type: none">• Robust security and compliance features• High availability and disaster recovery options• Powerful features for complex workloads	<ul style="list-style-type: none">• Higher cost than other options• Can be complex to manage	Most expensive option on RDS, pricing varies depending on edition and instance type
Microsoft SQL Server	Windows-based applications,	<ul style="list-style-type: none">• Familiar for Windows	<ul style="list-style-type: none">• Can be expensive for high-traffic	Typically more expensive than Amazon

	enterprise workloads	<p>developers</p> <ul style="list-style-type: none"> • High compatibility with other Microsoft products • Strong performance for certain workloads 	<p>applications</p> <ul style="list-style-type: none"> • Licensing can be complex 	MySQL/PostgreSQL, but cheaper than Oracle
Amazon DynamoDB	NoSQL database for high-throughput, low-latency applications	<ul style="list-style-type: none"> • Extremely scalable and performant • Pay-per-use billing model • Flexible data model 	<ul style="list-style-type: none"> • Not suitable for relational data • More complex to query than a relational database 	Generally lower cost than RDS options for high-throughput workloads

Table 1: Engine Types in Amazon RDS

I selected MariaDB as our Engine type as it is a modified version of MySQL and has more features, higher scalability and query speed compared to MySQL, while at the same time still supporting/retaining many features of MySQL (e.g. structure, naming conventions, connectors, connections, ports).

Create database

Choose a database creation method [Info](#)

Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)

Aurora (MySQL Compatible) 

Aurora (PostgreSQL Compatible) 

MySQL 

MariaDB 

PostgreSQL 

Oracle 

Figure 14: DB Instance Setting 1-1

Templates
Choose a sample template to meet your use case.

<input type="radio"/> Production Use defaults for high availability and fast, consistent performance.	<input type="radio"/> Dev/Test This instance is intended for development use outside of a production environment.
<input checked="" type="radio"/> Free tier Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info	

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. The first character must be a letter.

Manage master credentials in AWS Secrets Manager
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

① If you manage the master user credentials in Secrets Manager, some RDS features aren't supported. [Learn more](#)

Figure 15: DB Instance Setting 1-1

<input type="checkbox"/> Auto generate a password Amazon RDS can generate a password for you, or you can specify your own password.
Master password Info <input type="password"/>
Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote)', "(double quote)" and @ (at sign).
Confirm master password Info <input type="password"/>

Instance configuration
The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)
▼ Hide filters

Show instance classes that support Amazon RDS Optimized Writes [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

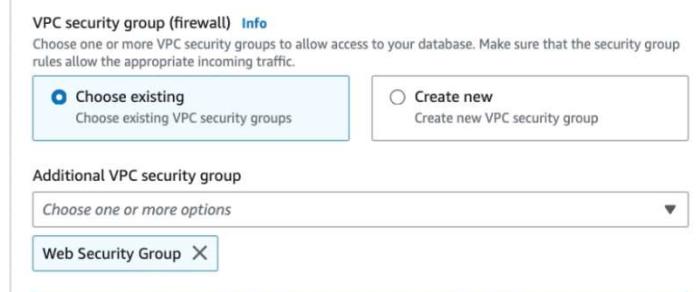
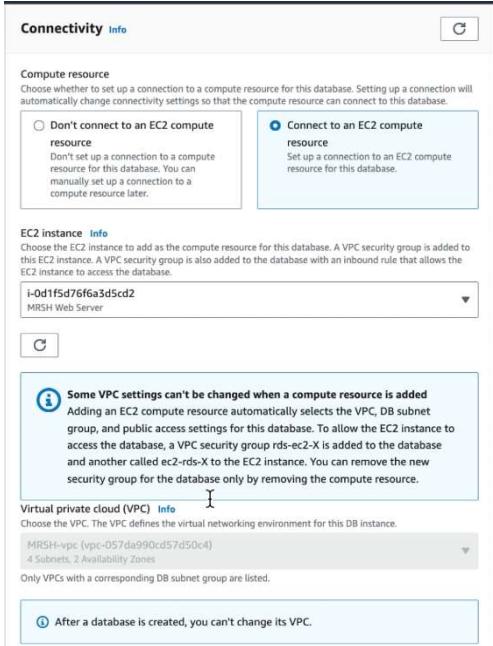
Include previous generation classes

Standard classes (includes m classes)
 Memory optimized classes (includes r and x classes)
 Burstable classes (includes t classes)

2 vCPUs 1 GiB RAM Network: 2,085 Mbps

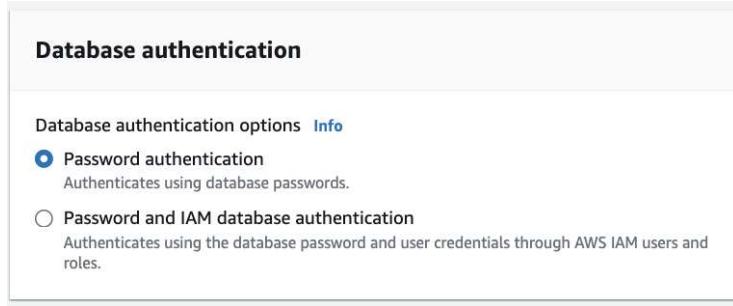
Figure 16: DB Instance Settings 1

Under Connectivity, I select the option to connect to our MRSIH Web Server compute resource.



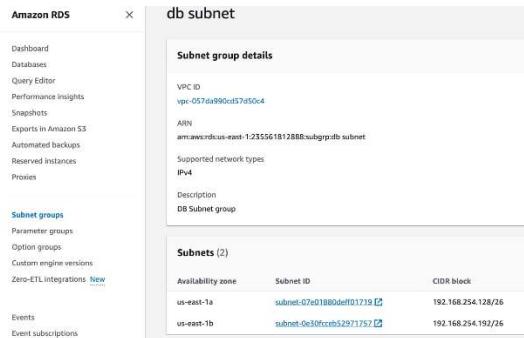
Figures 17 & 18: DB Instance Settings 2

In the figure below, I select Password authentication as it is suitable for my scenario of an SME. This enables easier administration especially in our case of a relatively smaller user community. Additionally, due to the AWS account restrictions, I am unable to utilise IAM capabilities although IAM is of a higher security standard.



Figures 19: DB Instance Settings 3

With the creation of the RDS, we must also ensure that we have 2 private subnets for the allocation of our RDS. For reference, we can see that the 2 subnets (192.168.254.128 & 192.168.254.192) assigned as db subnets are private subnets in our VPC.



Subnets (4/10) Info						
Actions Create subnet						
<input type="text"/> Find resources by attribute or tag						
Name	Subnet...	State	VPC	IPv4 CIDR	IPv6	IPv4
-	subnet-...	Available.	vpc-034...	172.31.80.0/20	-	-
-	subnet-...	Available.	vpc-034...	172.31.0.0/20	-	-
-	subnet-...	Available.	vpc-034...	172.31.48.0/20	-	-
<input checked="" type="checkbox"/> MRSH-subnet-private1-us-east-1a	subnet-...	Available.	vpc-057...	192.168.254.128/26	-	-
-	subnet-...	Available.	vpc-034...	172.31.32.0/20	-	-
<input checked="" type="checkbox"/> MRSH-subnet-public1-us-east-1a	subnet-...	Available.	vpc-057...	192.168.254.0/26	-	-
-	subnet-...	Available.	vpc-034...	172.31.64.0/20	-	-
-	subnet-...	Available.	vpc-034...	172.31.16.0/20	-	-
<input checked="" type="checkbox"/> MRSH-subnet-private2-us-east-1b	subnet-...	Available.	vpc-057...	192.168.254.192/26	-	-
<input checked="" type="checkbox"/> MRSH-subnet-public2-us-east-1b	subnet-...	Available.	vpc-057...	192.168.254.64/26	-	-

Figures 20: DB Instance private subnets allocation

Upon successful connection of the DB and EC2 web server instances, the following will pop up:

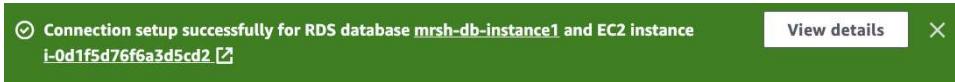


Figure 21: EC2-RDS successful connection

RDS-EC2 Web Server Security Group

Sunday, 19 October 2025 10:19 am

RDS-EC2 Web Server Security Group

The VPC, subnets, and security groups are required in order for the DB instance and Web Server to communicate. The VPC and subnets have already been configured in previous sections. Next, a security group to ensure connection is required for private RDS access where I create a new inbound rule allowing for MySQL traffic via port 3306 and the EC2 Web Server's private IP address as the Destination address. From this, we can connect our web server to the RDS instance for the purpose of retrieving or storing data between the web server and database.

Firstly, we create a new security group called WebServerConnect for the abovementioned purpose:

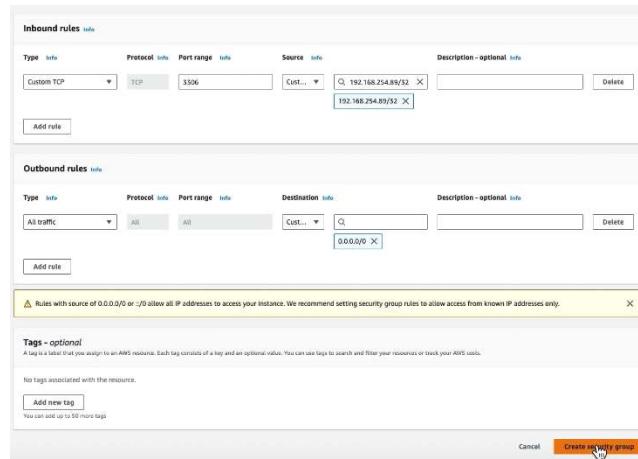
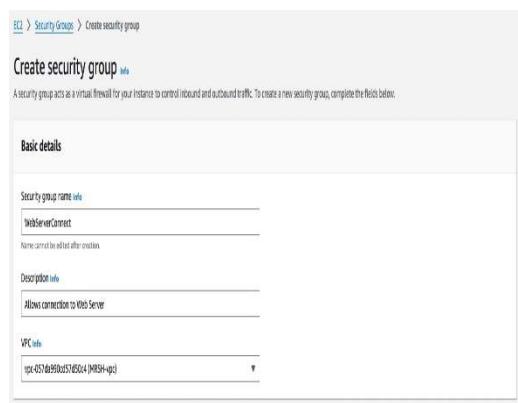


Figure 22: WebServerConnect Security Group

Thereafter, we return to the RDS console and add the WebServerConnect Security Group to our DB instance under "Modify".

Connectivity

Network type [Info](#)
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4
Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

DB subnet group
default-vpc-057da990cd57d50c4

Security group
List of DB security groups to associate with this DB instance.
[Choose security groups](#)

Web Security Group X **WebServerConnect** X

Certificate authority [Info](#)
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-2019 (default)
Expiry: Aug 23, 2024

► Additional configuration



Figure 23: Adding WebServerConnect to DB Instance

After modifying, the WebServerConnect security group should appear under VPC security groups as seen below with the “Active” symbol:

Amazon RDS

Databases **mrsh-db-instance1**

Summary

DB identifier	Status	Role	Engine	Recommendations
mrsh-db-instance1	Available	Instance	MariaDB	2 Informational
CPU	Class	Current activity	Region & AZ	
2.48%	db.t3.micro	0 Connections	us-east-1b	

Connectivity & security

Endpoint & port	Networking	Security
Endpoint mrsh-db- instance1.cz4g0wsupk.us-east- 1.rds.amazonaws.com	Availability Zone us-east-1b	VPC security groups Web Security Group (sg- 0x1f58331f1f1301d8)
Port 3306	VPC M8S14+VPC (vpc- 057da990cd57d50c4)	WebServerConnect (sg- 02190af538824aa896)
	Subnet group default-vpc-057da990cd57d50c4	Publicly accessible No
	Subnets subnet-07d1880e0d0f1718 subnet-04b55eef0277100995 subnet-0430fc8333971757 subnet-044aaaf022b45c1543	Certificate authority Info rds-ca-2019
	Network type IPv4	Certificate authority date August 23, 2024, 01:08 (UTC+ 08:00)
		DB instance certificate expiration date August 23, 2024, 01:08 (UTC+ 08:00)

Figure 24: WebServerConnect Security Group successfully added

S3 Bucket

Sunday, 19 October 2025 10:20 am

Bucket

S3 buckets can be created by moving to the S3 console and creating a bucket under “Buckets”. The basic settings for a bucket are as follows:

General configuration

AWS Region: US East (N. Virginia) us-east-1

Bucket type: General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory - New
Recommended for low-latency use cases. These buckets use only the S3 Express One-Zone storage class, which provides fast processing of data within a single Availability Zone.

Bucket name: mrsbucket1

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming [\[Learn more\]](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

Choose bucket
Format: s3://bucket/prefix

Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership: Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning: Disable
 Enable

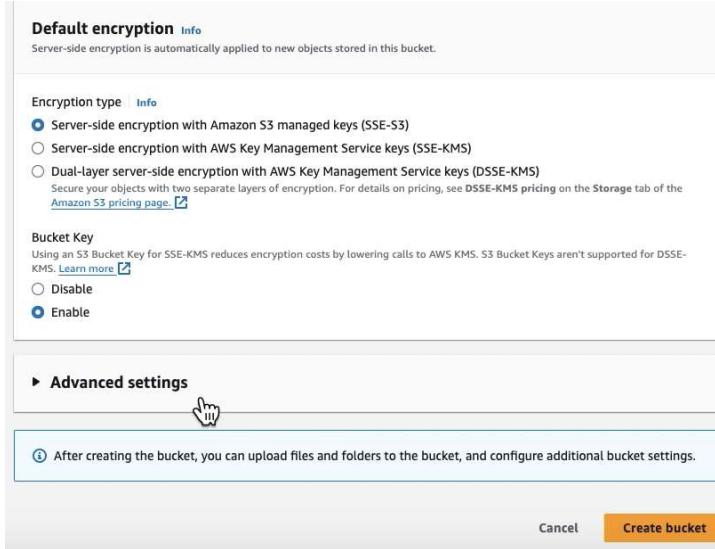
Tags - optional (0)

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

Add tag

Figures 25 & 26: S3 Bucket Set-Up 1



Figures 27: S3 Bucket Set-Up 2

Due to account restrictions, I am unable to connect the S3 bucket to the EC2 Web Server instance due to the unavailability of IAM functions. Below is a snapshot from the AWS website detailing the need of IAM to enable me to do so.

Short description

To connect to your S3 buckets from your EC2 instances, you must do the following:

1. Create an AWS Identity and Access Management (IAM) profile role that grants access to Amazon S3.
2. Attach the IAM instance profile to the instance.
3. Validate permissions on your S3 bucket.
4. Validate network connectivity from the EC2 instance to Amazon S3.
5. Validate access to S3 buckets.

Figure 28: AWS guide S3-EC2 connect

Access Control List (ACL)

Sunday, 19 October 2025 10:21 am

Network Access Control List (ACL)

We use Network ACLs to control inbound and outbound traffic at the subnet level for added security. To add ACLs, we go on the VPC console and click on Network ACLs. Then, we select our MRSH VPC:

The screenshot shows the AWS VPC Network ACLs page. On the left, there's a navigation sidebar with options like 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), and Network Firewall (Firewalls, Firewall policies, Network Firewall rule groups, TLS inspection configurations, Network Firewall resource groups). The main content area is titled "Network ACLs (1/2) Info". It displays two entries in a table:

Name	Network ACL ID	Associated with	Default	VPC ID
-	acl-06c55ac8e5dc77533	4 Subnets	Yes	vpc-057da990cd57d50c4 / MRSH-vpc
-	acl-05060b115b6f2f45d	6 Subnets	Yes	vpc-0349f53642659c0b5

The second row is highlighted with a mouse cursor. Below this, a detailed view for "acl-06c55ac8e5dc77533" is shown with tabs for Details, Inbound rules, Outbound rules, Subnet associations, and Tags. The Details tab shows the following information:

Network ACL ID acl-06c55ac8e5dc77533	Associated with 4 Subnets	Default Yes	VPC ID vpc-057da990cd57d50c4 / MRSH-vpc
Owner 235561812888			

Figure 29: Network ACLs page.

From this page, we can edit inbound and outbound rules under “Details” and add in the following rules. The rules below are a Custom network ACL for a IPV4 supported VPC. It allows for two-way traffic for HTTP, HTTPS (for internet connection), as well as rule 140 which covers ephemeral ports 32768-65535 (example). SSH and RDP is also enabled for remote access from a specific IP address. The use of Network ACLs adds an additional layer of security for our subnets as all traffic that are not specifically Allowed access from our ACLs will be directed to rule * which Denies all ineligible traffic.

acl-06c55ac8e5dc77533Details | **Inbound rules** | Outbound rules | Subnet associations | Tags**Inbound rules (6)**

Inbound rules (6)								
<input type="text"/> Filter inbound rules								
Rule number	Type	Protocol	Port range	Source	Allow/Deny			
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	<input checked="" type="radio"/> Allow			
110	HTTPS (443)	TCP (6)	443	0.0.0.0/0	<input checked="" type="radio"/> Allow			
120	SSH (22)	TCP (6)	22	42.0.0.0/8	<input checked="" type="radio"/> Allow			
130	RDP (3389)	TCP (6)	3389	42.0.0.0/8	<input checked="" type="radio"/> Allow			
140	Custom TCP	TCP (6)	32768 - 65535	0.0.0.0/0	<input checked="" type="radio"/> Allow			
*	All traffic	All	All	0.0.0.0/0	<input checked="" type="radio"/> Deny			

Figure 30: Network ACL Inbound rules

acl-06c55ac8e5dc77533Details | Inbound rules | **Outbound rules** | Subnet associations | Tags**Outbound rules (5)**

Outbound rules (5)								
<input type="text"/> Filter outbound rules								
Rule number	Type	Protocol	Port range	Destination	Allow/Deny			
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	<input checked="" type="radio"/> Allow			
110	HTTPS (443)	TCP (6)	443	0.0.0.0/0	<input checked="" type="radio"/> Allow			
120	SSH (22)	TCP (6)	22	42.0.0.0/8	<input checked="" type="radio"/> Allow			
140	Custom TCP	TCP (6)	32768 - 65535	0.0.0.0/0	<input checked="" type="radio"/> Allow			
*	All traffic	All	All	0.0.0.0/0	<input checked="" type="radio"/> Deny			

Figure 31: Network ACL Outbound rules

Security Groups

Sunday, 19 October 2025 10:22 am

Security Groups

Security Groups are yet another security feature that protects our instances from unauthorised traffic. For our EC2 Web Server instance, we can see in Figure 32 that it has been assigned 3 Security Groups;

SSH – This security group allows for traffic via port 22 (SSH) connected to a specific IP address; which means that only that IP address can connect to the EC2 instance via SSH. All other SSH attempts from other IP addresses are denied. This is illustrated in Figure 33.

Web Security Group – This group allows for HTTP and HTTPS traffic from all sources, as shown in Figure 34.

ec2-rds-1 – This security group was automatically created upon setting up the DB instance (RDS) and selecting the option to connect to this web server. Figure 35 shows the outbound rule for MYSQL via port 3306, enabling connectivity between the web server and the DB instance.

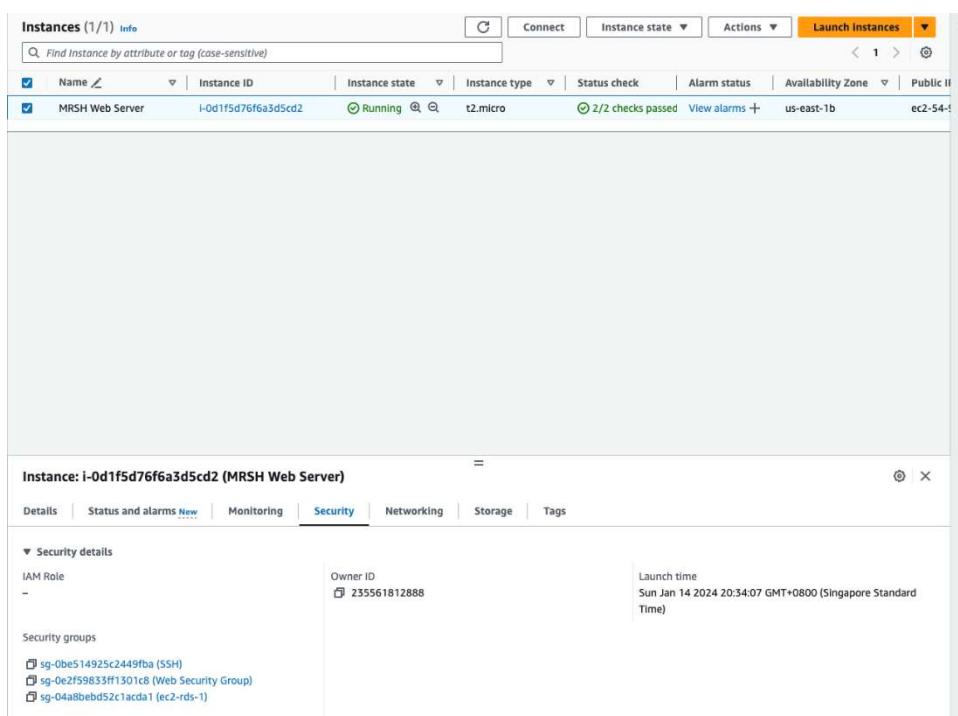


Figure 32: EC2 Security Groups

[EC2](#) > [Security Groups](#) > sg-0be514925c2449fba - SSH

sg-0be514925c2449fba - SSH

[Actions ▾](#)

Details			
Security group name SSH	Security group ID sg-0be514925c2449fba	Description Enable SSH Access	VPC ID vpc-057da990cd57d50c4
Owner 235561812888	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Tags

Inbound rules (1)						
<input type="text"/> Search C Manage tags Edit inbound rules 						
Security group rule...	IP version	Type	Protocol	Port range	Source	
sgr-04f0dfa6b364eac81	IPv4	SSH	TCP	22	42.60.61.87/32	

Figure 33: EC2 SSH Security Group

[EC2](#) > [Security Groups](#) > sg-0e2f59833ff1301c8 - Web Security Group

sg-0e2f59833ff1301c8 - Web Security Group

[Actions ▾](#)

Details			
Security group name Web Security Group	Security group ID sg-0e2f59833ff1301c8	Description Enable HTTP access	VPC ID vpc-057da990cd57d50c4
Owner 235561812888	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Tags

Inbound rules (2)						
<input type="text"/> Search C Manage tags Edit inbound rules 						
Security group rule...	IP version	Type	Protocol	Port range	Source	
sgr-0685310dea981c49f	IPv4	HTTPS	TCP	443	0.0.0.0/0	
sgr-00085b860662d4...	IPv4	HTTP	TCP	80	0.0.0.0/0	

Figure 34: EC2 Web Security Group

[EC2](#) > [Security Groups](#) > sg-04a8beb52c1acda1 - ec2-rds-1

sg-04a8beb52c1acda1 - ec2-rds-1

[Actions ▾](#)

Details			
Security group name ec2-rds-1	Security group ID sg-04a8beb52c1acda1	Description Security group attached to instances to securely connect to mrs-db-instance1. Modification could lead to connection loss.	VPC ID vpc-057da990cd57d50c4
Owner 235561812888	Inbound rules count 0 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules | [Outbound rules](#) | Tags

Outbound rules (1/1)						
<input type="text"/> Search C Manage tags Edit outbound rules 						
Security group rule...	IP version	Type	Protocol	Port r...	Destination	
sgr-09c22b3cce23cc11b	-	MySQL/Aurora	TCP	3306	sg-0fde710fe83c24fe...	

Figure 35: EC2 ec2-rds-1 Security Group

As for the DB Instance, the final security group configuration is depicted in Figure 36 under “VPC Security groups”, containing only the WebServerConnect Security Group for the DB instance to communicate with the web server. All other traffic is denied access for added security on top of our DB instance being placed in a private subnet.

The screenshot shows the AWS RDS console with the following details:

Summary

DB Identifier	Status	Role	Engine	Recommendations
mrsh-db-instance1	Available	Instance	MariaDB	2 Informational
CPU	Class	Current activity	Region & AZ	
2.26%	db.t3.micro	0 Connections	us-east-1b	

Connectivity & security

Endpoint & port	Networking	Security
Endpoint mrsh-db-instance1.cx4g0awsupkl.us-east-1.rds.amazonaws.com	Availability Zone us-east-1b	VPC security groups WebServerConnect (sg-02c900f538824aa96) Active
Port 3306	VPC MRSH-vpc (vpc-057da990cd57d50c4)	Publicly accessible No
	Subnet group default-vpc-057da990cd57d50c4	Certificate authority Info rds-ca-2019
	Subnets subnet-07e01880deff01719 subnet-0d5ceeb0277d0695 subnet-0e30fcceb52971757 subnet-0e4af022b45c1543	Certificate authority date August 23, 2024, 01:08 (UTC+08:00)
	Network type IPv4	DB instance certificate expiration date ⚠️ August 23, 2024, 01:08 (UTC+08:00)

Figure 36: RDS DB instance Security Group

CloudTrail Trail

Sunday, 19 October 2025 10:25 am

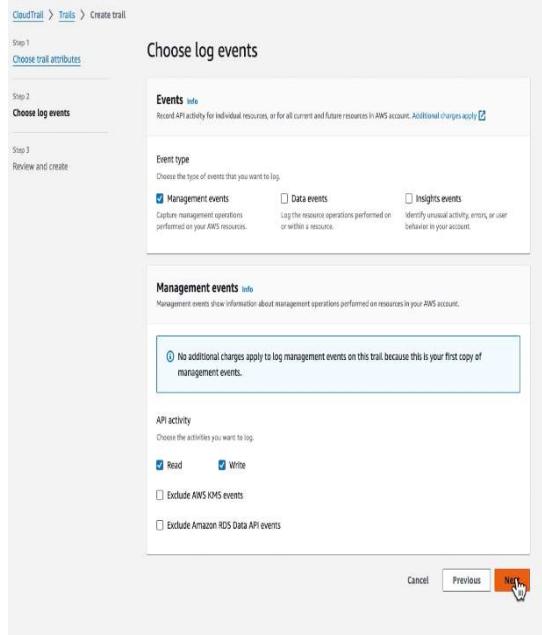
CloudTrail Trail

CloudTrail is an AWS function that lets us track user/role/service activities down to each action (AKA events). This helps to monitor our cloud environment and conduct audits and troubleshooting. I created a CloudTrail trail by entering the CloudTrail console and clicking on “Create trail” under Trails. I also enabled CloudWatch Logs in Figures 37 & 38 to centralise and monitor my trail logs.

The screenshot shows the 'Choose trail attributes' step of the CloudTrail trail creation wizard. It includes:

- General details:** A trial named 'MHSCloudTrail' is selected. It's noted that this is a multi-region trail.
- Storage location:** The 'Create new S3 bucket' option is selected, creating a new bucket named 'aws-cloudtrail-logs-235561812888-5f4ad7ac'.
- CloudWatch Logs - optional:** Log file SSE-KMS encryption is disabled. Log file validation is enabled. Log file validation role is 'Lambda'. Log group name is 'aws-cloudtrail-logs-235561812888-5f4ad7ac'.

Figures 37 & 38: CloudTrail trail Set-up 1



Review and create

Step 1: Choose trail attributes

General details		
Trail name	Trail log location	Log file validation
HRSICloudTrail	aws-cloudtrail-logs-235561812888	Enabled
Multi-region trail	674ad7ac(AWSLogs/235561812888)	SNS notification delivery
Yes	R	Disabled
Apply trail to my organization	Log file SSE-KMS encryption	
Not enabled	Not enabled	

CloudWatch Logs

Log group	IAM Role
aws-cloudtrail-logs-235561812888-8660f4d5	LabRole

Tags

Key	Value
No tags No tags associated with this trail	

Step 2: Choose log events

Management events

No additional charges apply to log management events on this trail because this is your first copy of management events.

API activity	Exclude AWS KMS events
All	No
	Exclude Amazon RDS Data API events
	No

Figures 39 & 40: CloudTrail trail Set-up 2

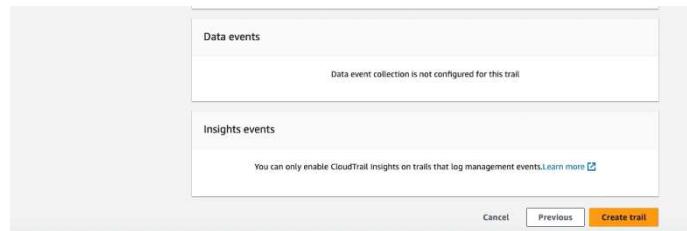


Figure 41: CloudTrail trail Set-up 3

After completing the trail set-up, the following pop-up emerges, which may be due to the security restrictions placed on AWS accounts that are used for labs:

⚠️ We were unable to create/update the role or policy.

Figure 42: Trail Creation Error

The new trail will appear under Trails:

Trails								
Name	Home region	Multi-region trail	Insights	Organization trail	S3 bucket	Log file prefix	CloudWatch Logs log group	Status
MRSHCloudTrail	US East (N. Virginia)	Yes	Disabled	No	aws-cloudtrail-logs-235561812885-674ad7ac	-	-	Logging

Figure 43: Trails

Simple Notification Service (SNS)

Sunday, 19 October 2025 10:26 am

Simple Notification Service

Amazon Simple Notification Service (SNS) is an AWS service that provides notifications from publishers to subscribers. We can utilise this service to send us notification messages whenever a certain event occurs. We create an SNS by going onto the SNS and clicking on “Create Topic” under Topics. The SNS Set-Up is as follows:

Amazon SNS > Topics > Create topic

Create topic

Details

Type [Info](#)
Topic type cannot be modified after topic is created.

FIFO (first-in, first-out)
• Strictly-preserved message ordering
• Exactly-once message delivery
• High throughput, up to 300 publishes/second
• Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Standard
• Best-effort message ordering
• At-least-once message delivery
• Higher throughput in publisher/second
• Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name
Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name [Info](#)
To use the topic in an SNS subscribe, enter a display name. Only the first 10 characters are displayed in an SNS message.

Maximum 100 characters.

Encryption - optional
Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

▼ Access policy - optional [Info](#)
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

Choose method
 Basic Use simple criteria to define a basic access policy
 Advanced Use a JSON object to define an advanced access policy.

Publishers Specify who can publish messages to the topic.
 Everyone Anybody can publish

Subscribers Specify who can subscribe to this topic.
 Everyone Any AWS account can subscribe to the topic

JSON preview

```
{ "Version": "2008-10-17", "Id": "..._default_policy_ID", "Statement": [ { "Sid": "..._default_statement_ID", "Effect": "Allow", "Principal": { "AWS": "*" }, "Action": [ "SNS:Publish", "SNS:RemovePermission", "SNS:DeleteTopic", "SNS:ListTopics", "SNS:Subscribe", "SNS:ListSubscriptions", "SNS:ListSubscriptionsByTopic", "SNS:GetSubscriptionAttributes", "SNS:ChangeTopicOwner", "SNS:DeleteSubscription", "SNS:ListTopics" ] } ] }
```

► Data protection policy - optional [Info](#)
This policy defines which sensitive data to monitor and to prevent from being exchanged via your topic.

► Delivery policy (HTTP/S) - optional [Info](#)
The policy defines how Amazon SNS retries failed deliveries to HTTP/S endpoints. To modify the default settings, expand this section.

► Delivery status logging - optional [Info](#)
These settings configure the logging of message delivery status to CloudWatch Logs.

► Tags - optional [Info](#)
A tag is a metadata label that you can assign to an Amazon SNS topic. Each tag consists of a key and an optional value. You can use tags to search and filter your topics and track your costs. [Learn more](#)

► Active tracing - optional [Info](#)
Use AWS X-Ray active tracing for this topic to view its traces and service map in Amazon CloudWatch. Additional costs apply.

[Cancel](#) [Create topic](#)

Figures 44 & 45: SNS Set-Up

After setting up the topic, we can create subscriptions to the topic for subscribers to receive notifications.

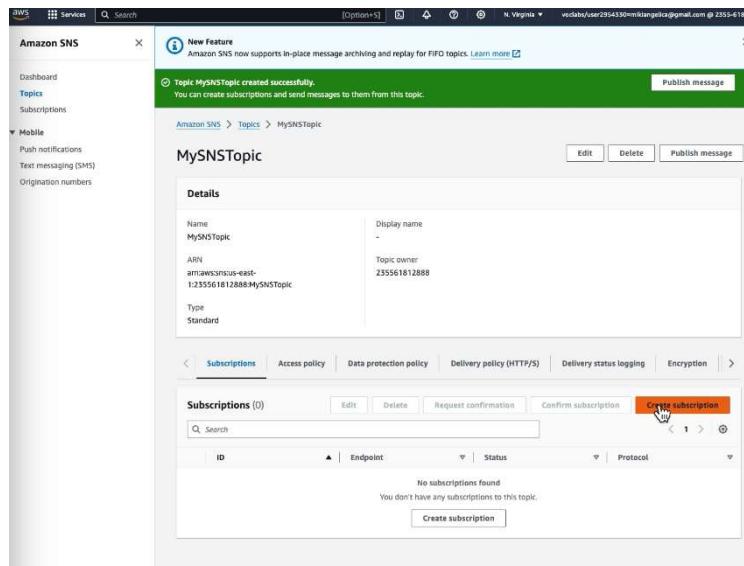


Figure 46: SNS successful creation

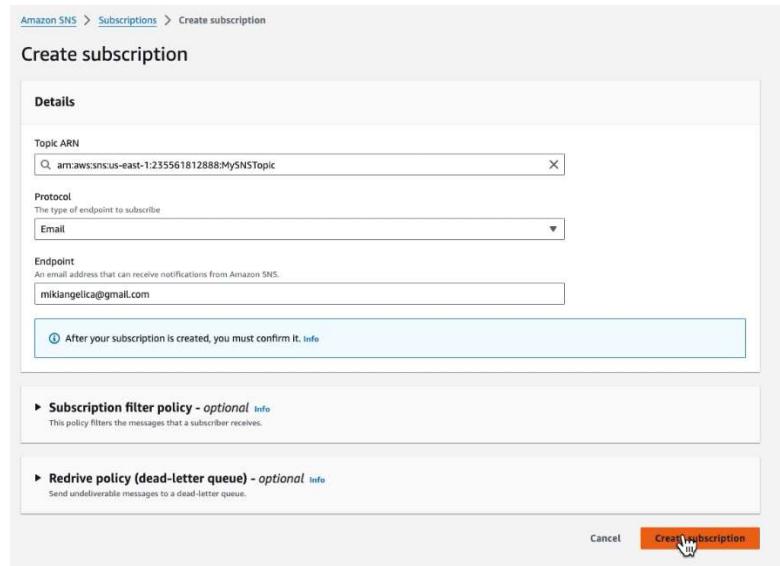


Figure 47: Create Subscription

Upon subscription, a message will be sent to the subscriber to confirm subscription:



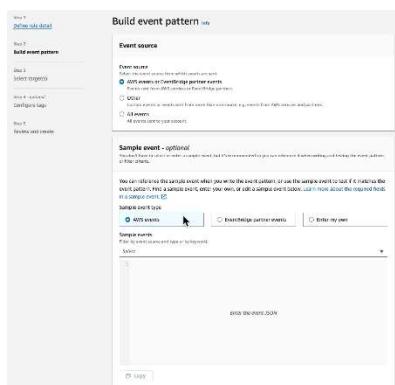
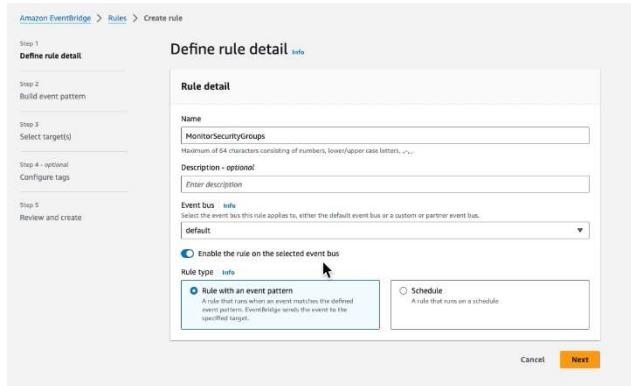
Figure 48: Subscription confirmed

EventBridge

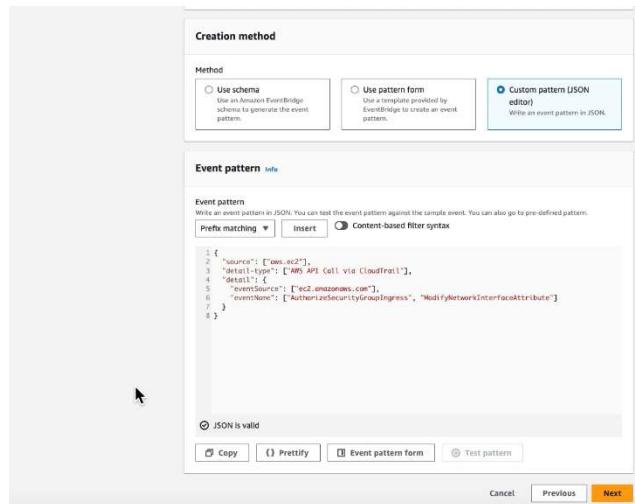
Sunday, 19 October 2025 10:27 am

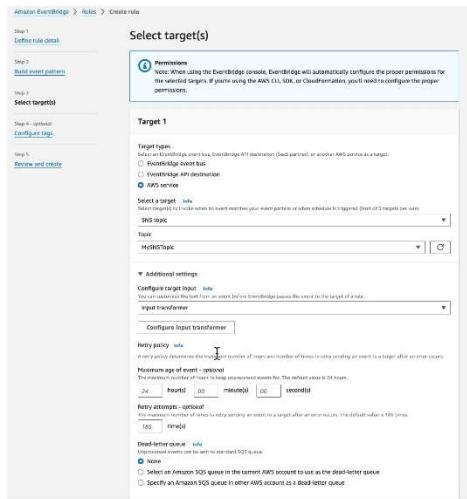
EventBridge Rule

In this section, I create an EventBridge Rule using AWS EventBridge which will detect the occurrence of a specified event and send it to our target (SNS). The set-up is as follows:



Figures 49 & 50: EventBridge Rule Set-Up 1





Figures 51 & 52: EventBridge Rule Set-Up 2

Click on Configure input transformer to paste the following code that configures the message:

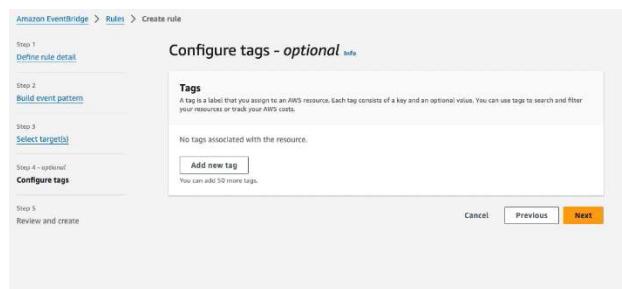
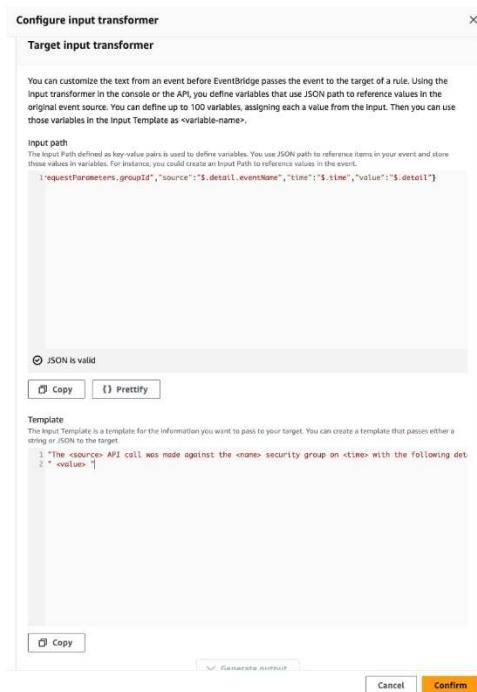


Figure 53 & 54: EventBridge Rule Set-Up 3

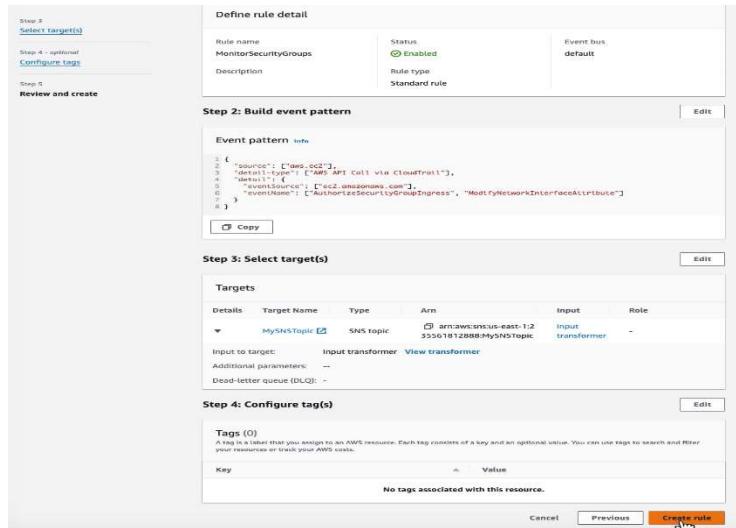


Figure 55: EventBridge Rule Set-Up 4

To test the EventBridge rule that we have set up above, we go back onto the EC2 console and select our MRSW Web Server instance > Security > click on Web Security Group link > Edit inbound rules > add SSH port:

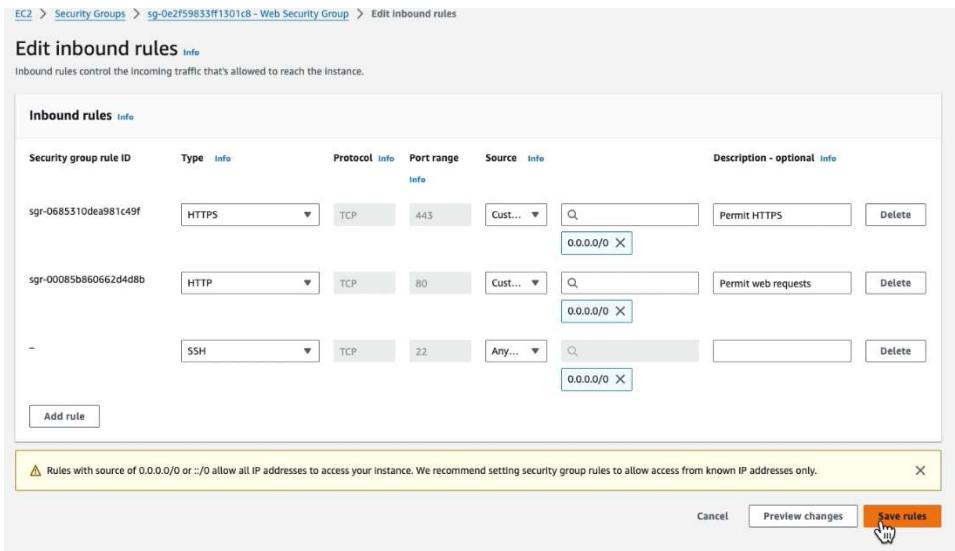


Figure 56: EventBridge Test: Add SSH port to EC2 Security Group

Then we return to CloudTrail > Event history. After a minute or two, the event "AuthorizeSecurityGroupIngress" should appear:

ListDelegatedAdministrators	January 14, 2024, 00:02:16 (UT...)	user2954330=miki...	organizations.amazonaws.com	-
ListNotificationHubs	January 14, 2024, 00:02:16 (UT...)	user2954330=miki...	notifications.amazonaws.com	-
DescribeEventAggregates	January 14, 2024, 00:02:15 (UT...)	user2954330=miki...	health.amazonaws.com	-
ListNotificationHubs	January 14, 2024, 00:02:15 (UT...)	user2954330=miki...	notifications.amazonaws.com	-
DescribeEventAggregates	January 14, 2024, 00:02:15 (UT...)	user2954330=miki...	health.amazonaws.com	-
ListNotificationHubs	January 14, 2024, 00:02:12 (UT...)	user2954330=miki...	notifications.amazonaws.com	-
ListNotificationHubs	January 14, 2024, 00:02:11 (UT...)	user2954330=miki...	notifications.amazonaws.com	-
DescribeEventAggregates	January 14, 2024, 00:02:11 (UT...)	user2954330=miki...	health.amazonaws.com	-
DescribeEventAggregates	January 14, 2024, 00:02:11 (UT...)	user2954330=miki...	health.amazonaws.com	-
GetTrailStatus	January 14, 2024, 00:02:10 (UT...)	user2954330=miki...	cloudtrail.amazonaws.com	AWS::CloudTrail
LookupEvents	January 14, 2024, 00:02:10 (UT...)	user2954330=miki...	cloudtrail.amazonaws.com	-
DescribeTrails	January 14, 2024, 00:02:10 (UT...)	user2954330=miki...	cloudtrail.amazonaws.com	-
ListDelegatedAdministrators	January 14, 2024, 00:02:09 (UT...)	user2954330=miki...	organizations.amazonaws.com	-
LookupEvents	January 14, 2024, 00:02:09 (UT...)	user2954330=miki...	cloudtrail.amazonaws.com	-
DescribeOrganization	January 14, 2024, 00:02:09 (UT...)	user2954330=miki...	organizations.amazonaws.com	-
ListBuckets	January 14, 2024, 00:02:08 (UT...)	user2954330=miki...	s3.amazonaws.com	-
DescribeTrails	January 14, 2024, 00:02:08 (UT...)	user2954330=miki...	cloudtrail.amazonaws.com	-
DescribeEventAggregates	January 14, 2024, 00:02:05 (UT...)	user2954330=miki...	health.amazonaws.com	-
DescribeEventAggregates	January 14, 2024, 00:02:05 (UT...)	user2954330=miki...	health.amazonaws.com	-
DescribeSecurityGroups	January 14, 2024, 00:01:54 (UT...)	user2954330=miki...	ec2.amazonaws.com	-
DescribeSecurityGroupRules	January 14, 2024, 00:01:54 (UT...)	user2954330=miki...	ec2.amazonaws.com	-
AuthorizeSecurityGroupIngress	January 14, 2024, 00:01:53 (UT...)	user2954330=miki...	ec2.amazonaws.com	AWS::EC2
DescribeSecurityGroups	January 14, 2024, 00:01:42 (UT...)	user2954330=miki...	ec2.amazonaws.com	-
DescribeSecurityGroupRules	January 14, 2024, 00:01:42 (UT...)	user2954330=miki...	ec2.amazonaws.com	-
DescribeSecurityGroupRules	January 14, 2024, 00:01:39 (UT...)	user2954330=miki...	ec2.amazonaws.com	-

Figure 57: EventBridge Test: Check CloudTrail Event History

Upon clicking onto the event “AuthorizeSecurityGroupIngress”, we will be able to view its event record containing important details such as userIdentity, eventTime, and awsRegion:

AWS:EC2:SecurityGroup		sp-0e2f59833ff1301cb	Enable AWS Config resource recording
Event record Info			
JSON view			
<pre>{ "eventVersion": "1.07", "userIdentity": { "type": "AssumedRole", "principalId": "AROAYTMSEUJUNQNGGCCXQDC:user2954330=mikiangelica@gmail.com", "arn": "arn:aws:sts::235561812888:assumed-role/voclafe/user2954330=mikiangelica@gmail.com", "accountId": "15161812888", "accessKeyId": "ASIAZM7DMDGJ", "sessionContext": { "sessionIssuer": { "type": "Role", "principalId": "AROAYTMSEUJUNQNGGCCXQDC", "arn": "arn:aws:iam::235561812888:role/voclafe", "accountId": "235561812888", "username": "voclafe" }, "attributes": { "creationDate": "2024-01-19T14:25:47Z", "mfaAuthenticated": "false" } } }, "eventTime": "2024-01-19T14:01:53Z", "eventSource": "ec2.amazonaws.com", "eventName": "AuthorizeSecurityGroupIngress", "version": "1", "sourceIPAddress": "42.69.41.87", "userAgent": "AWS Internal", "requestParameters": { "groupId": "sg-e02f59833ff1301cb", "ipPermissions": ["items": [{ "ipProtocol": "tcp", "fromPort": 22, "toPort": 22, "groups": [], "ipRanges": [{ "cidrIp": "0.0.0.0/0" }], "ipvRanges": [], "prefixListIds": [] }]] } }</pre>			

Figure 58: EventBridge Test: ‘AuthorizeSecurityGroupIngress’ Event Record

Subscribers to the relevant SNS topic for this EventBridge rule will receive the following email signalling test success:

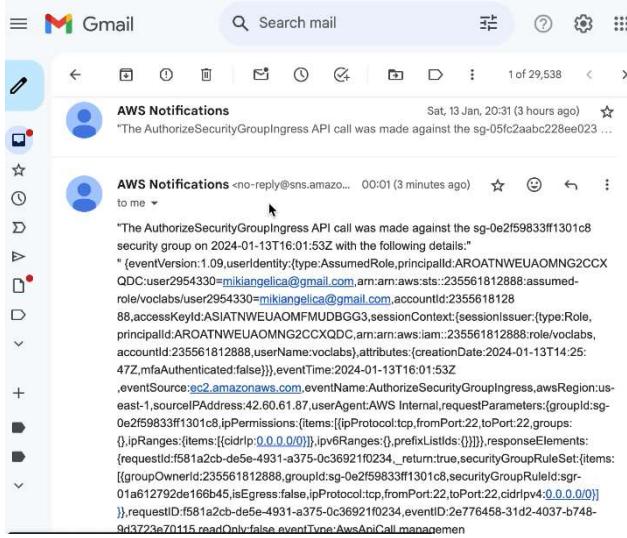


Figure 59: EventBridge Test: Email Notification

Cloudwatch Alarm

Sunday, 19 October 2025 10:30 am

Cloudwatch Alarm

Using CloudWatch, we can set alarms to monitor and detect CloudWatch metrics that we define. Notifications will also be sent if defined metric thresholds are met. CloudWatch Alarms can also be used to automatically stop, terminate, reboot, or recover our EC2 instances.

To create an alarm, we go to CloudWatch console > Navigation bar > Log groups > Select your Trail log group > Actions > Create metric filter:

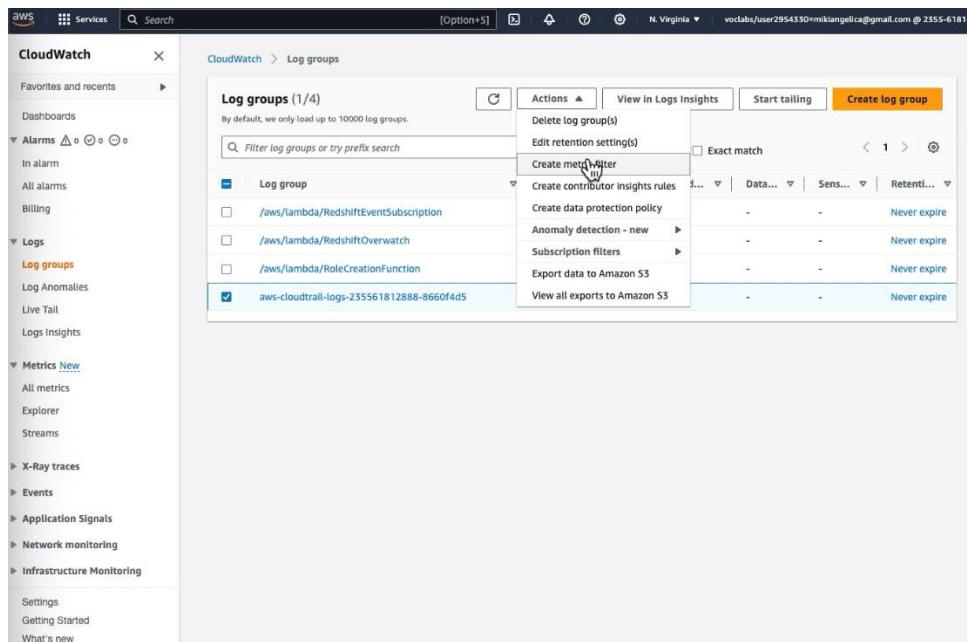
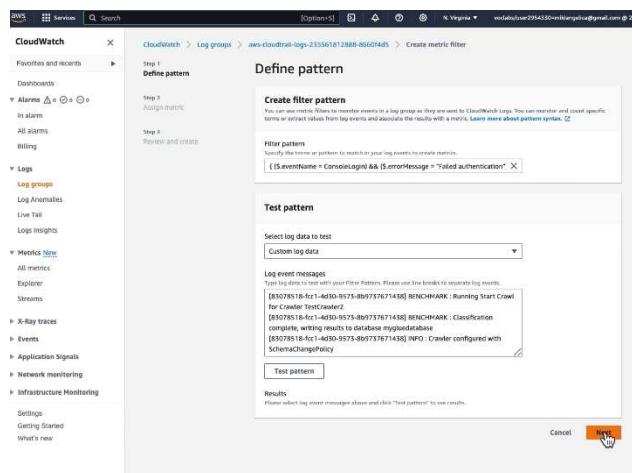


Figure 60: CloudWatch Log groups > Create metric filter



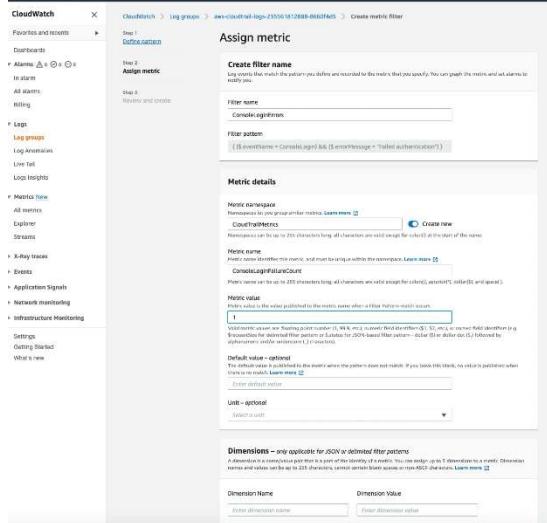


Figure 61 & 62: Metric Filter Set-up 1

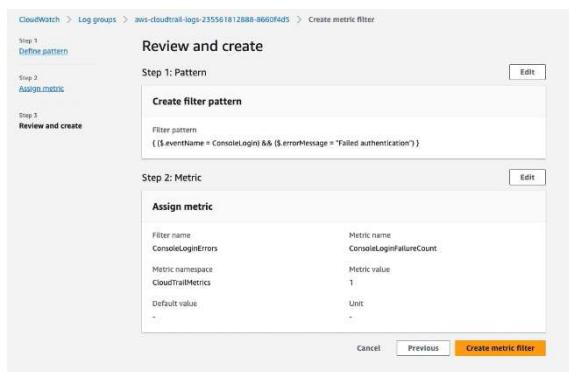


Figure 63: Metric Filter Set-up 2

After setting up the metric filter that detects and counts login failures, we create an alarm that detects when the Login Failure count reaches our defined threshold (3 or more), which will then send a notification to our topic subscribers.

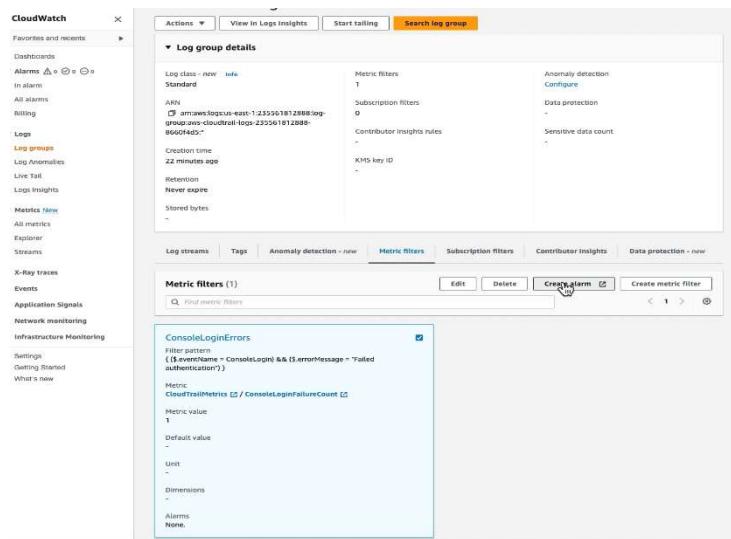


Figure 64: Metric Filters

CreateWatch > Alarms > Create alarm

Specify metric and conditions

Step 1
Configure actions

Step 2
Add name and description

Step 3
Preview and create

Metric

Graph
The alarm will trigger when the total logins for "ConsoleLogInFailureCount" datapoint is 5 minutes.

Key unit: Namespace:

Metric name:

Statistic: Sum

Period:

Conditions

Threshold type:
 Static Use a value as a threshold Anomaly detection Use a band as a threshold

Whenever ConsoleLogInFailureCount is...
 Greater than or equal to Greater/Equal to Less/Equal to Less than or equal to

Than...
 Exact value Percent of threshold value Above threshold Below threshold

Point to a location

Additional configuration

Cancel **Next**

CreateWatch > Alarms > Create alarm

Configure actions

Step 1
Specify metric and conditions

Step 2
Configure actions

Step 3
Add name and description

Step 4
Preview and create

Notification

Alarm state trigger
Define the alarm state that will trigger the notification.

An alarm triggers when the metric is outside of its defined thresholds. OK When the metric is at or above the threshold. Insufficient data When there is not enough data available to evaluate the alarm.

Select a notification to the following SNS topic
Select the SNS topic that will receive the notification.

Select an existing SNS topic Create new topic Use next step to notify other accounts

Send a notification to...

Email (Amazon SES) Loading Add notification

Lambda action

Auto Scaling action

EC2 action

Systems Manager action

This action is only available for EC2 for Instance Health.

Cancel **Next**

Figures 65 & 66: Alarm Set-up 1

CreateWatch > Alarms > Create alarm

Add name and description

Step 1
Specify metric and conditions

Step 2
Configure actions

Step 3
Add name and description

Step 4
Preview and create

Name and description

Alarm name:

Alarm description - optional | [View formatting guidelines](#)

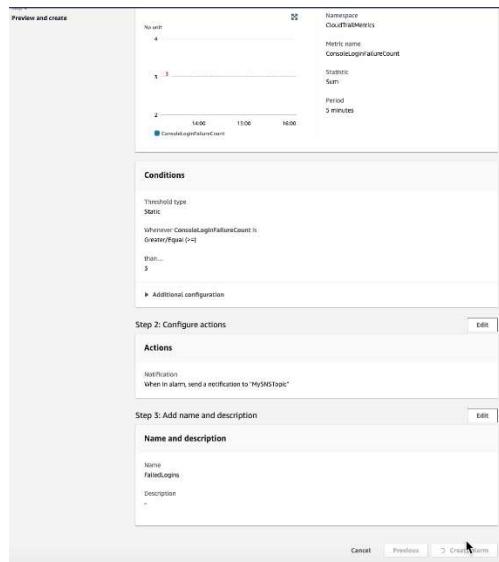
Edit **Preview**

This is an H1
double asterisks will produce strong character
This is [an example](https://example.com/) inline link.

Up to 1024 characters (0/1024)

Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

Cancel **Previous** **Next**



Figures 67 & 68: Alarm Set-up 2

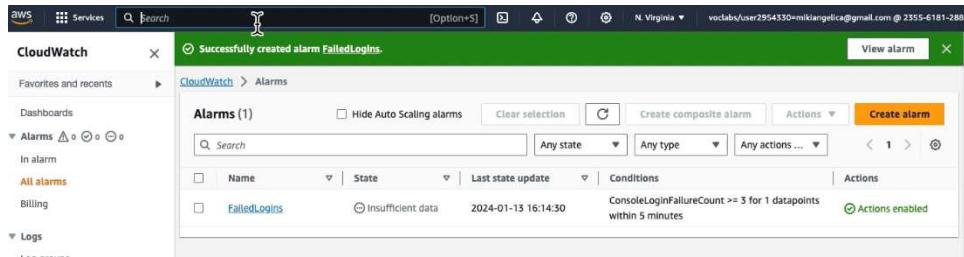


Figure 69: Alarms

Due to the inability to access IAM features, I am unable to test the alarms above from Failed Logins.

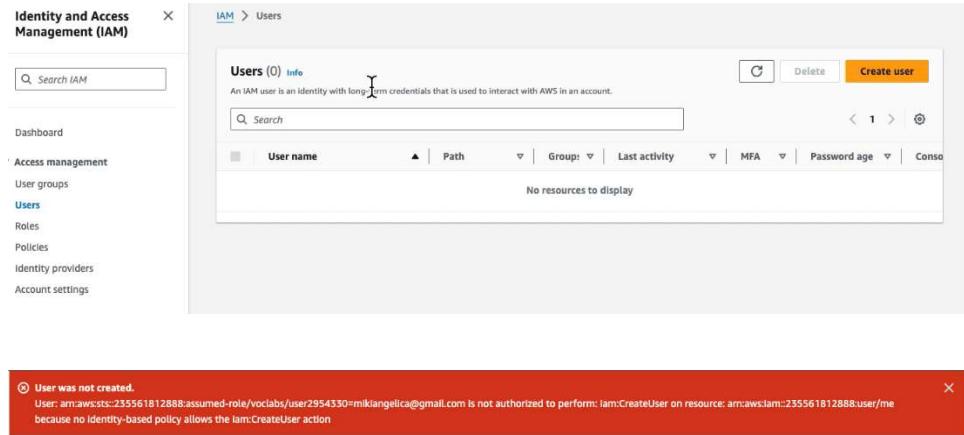


Figure 69: IAM console

Querying CloudTrail Logs

Sunday, 19 October 2025 10:33 am

Querying CloudTrail logs

To query CloudTrail logs, we go back to CloudWatch > Navigation Bar > Logs Insights > Select your log group > Paste code > Run query > A Histogram should appear below for our FailedLogins log but because we are unable to test, no data is shown.

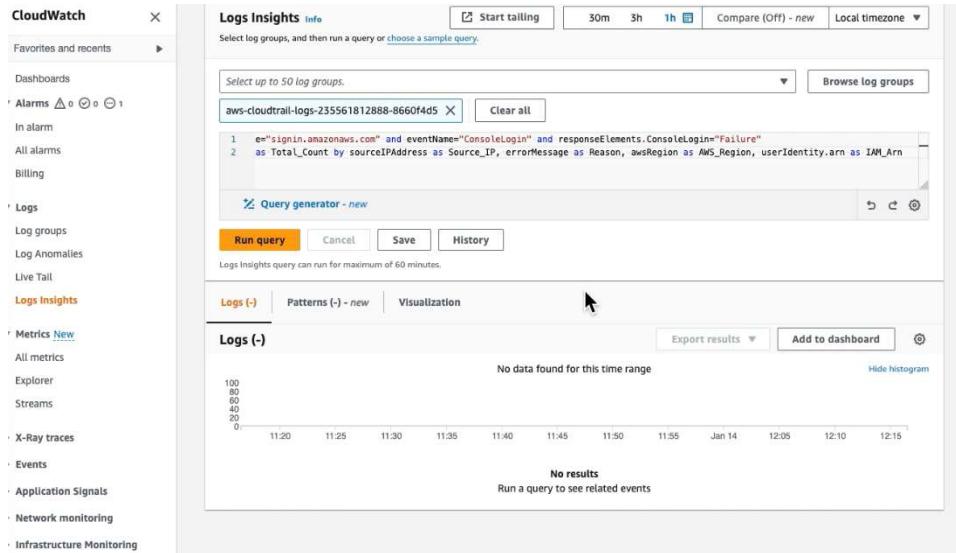


Figure 70: Logs Insights

Auto-scaling and Load Balancing

Sunday, 19 October 2025 10:33 am

Auto-Scaling and Load Balancing

Load-Balancing

We create an instance-type Target Group named “MRSHGroup1. Target groups define the destination to send incoming traffic from load balancers to.

Target group name
MRSHGroup1
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port
Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP 80 1-65535

IP address type
Only targets with the indicated IP address type can be registered to this target group.

IPv4
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

MRSH-vpc
vpc-057da990cd57d50c4
IPv4: 192.168.254.0/24

Protocol version
 HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
 HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
 gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Figure 71: Target Group MRSHGroup1

Then, we create a Load Balancer. Load balancing allows MRS to spread traffic across multiple instances and availability zones to accommodate high or low traffic periods. In this case, we will be making an Application Load Balancer so that we can send traffic to our instance-type target group. We create a Load Balancer mapped to our 2 public subnets for our MRSH-vpc. This Load Balancer will be assigned the Web Security Group.

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)
Scheme can't be changed after the load balancer is created.

Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)
Select the type of IP addresses that your subnets use.

IPv4
Recommended for internal load balancers.

Dualstack
Includes IPv4 and IPv6 addresses.

Network mapping [Info](#)
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)
Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

[vpc-057da990cd57d50c4](#) [IPv4: 192.168.254.0/24](#) [C](#)

Mappings [Info](#)
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

Figure 72: Load Balancer MRSH_ELB

We connect the Load Balancer to our MRSHGroup1 Target Group created prior:

Listeners and routing [Info](#)
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80 [Remove](#)

Protocol	Port	Default action	Info
HTTP	: 80 1-65535	Forward to	MRSHGroup1 Target type: Instance, IPv4
HTTP C			
Create target group			

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)
You can add up to 50 more tags.

[Add listener](#)

Figure 73: Connection to MRSHGroup1 Target Group

Auto-scaling

On the EC2 console, we create an image for our MRSH Web Server as in the figure below for autoscaling. This allows us to save boot disk contents to launch instances with identical initial contents.

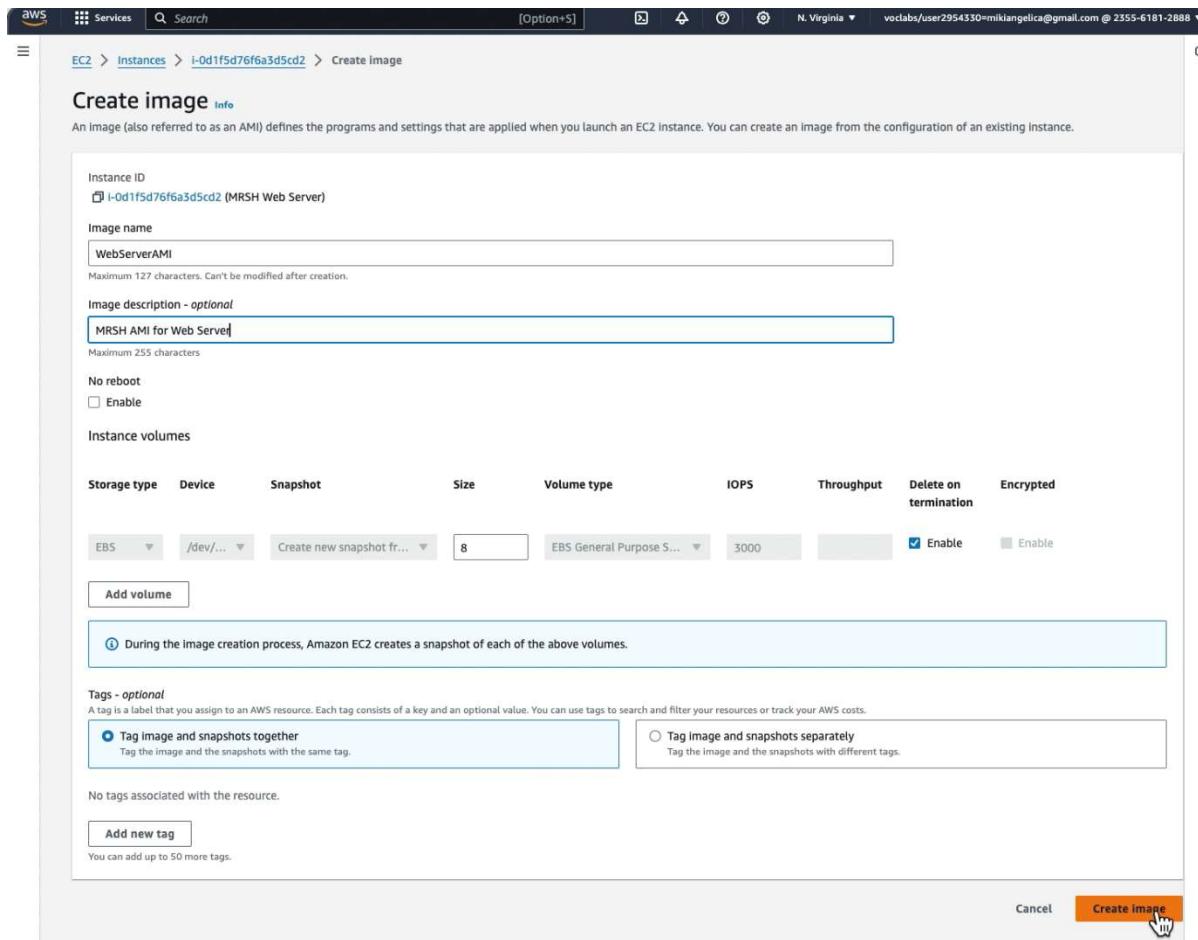


Figure 74: WebServerAMI Image

Next, we create the MRSConfig Launch Template for our Auto Scaling group to use when launching EC2 instances. This launch template will contain a default template instance containing information such as AMI (using the image created prior), instance type, key pair, and security group. We attach this launch template to our auto scaling group “MRSH Auto Scaling Group” and load balancer “MRSH_ELB”. In our Auto Scaling, we input a desired capacity of 2 instances, a minimum of 2, and a maximum of 6. The Auto Scaling feature will align with the Load Balancer and add or remove instances as needed, keeping between 2 and 6 instances at any one time.

EC2 > Launch templates > MRSHConfig

MRSHConfig (lt-07b805d4a3b19ec5c)

Actions ▾ Delete template

Launch template details

Launch template ID lt-07b805d4a3b19ec5c	Launch template name MRSHConfig	Default version 1	Owner arn:aws:sts::235561812888:assumed-role/voclabs/user2954330=miklangelica@gmail.com
--	------------------------------------	----------------------	--

Details | Versions | Template tags

Launch template version details

Actions ▾ Delete template version

Version 1 (Default)	Description -	Date created 2024-02-07T00:08:22.000Z	Created by arn:aws:sts::235561812888:assumed-role/voclabs/user2954330=miklangelica@gmail.com
------------------------	------------------	--	---

Instance details | Storage | Resource tags | Network interfaces | Advanced details

AMI ID ami-0f98bd076a53f1ed4	Instance type t2.micro	Availability Zone -	Key pair name vockey
Security groups -	Security group IDs sg-0e2f59833ff1301c8		

Figure 75: Launch Template MRSHConfig

Step 1
Choose launch templateStep 2
Choose instance launch optionsStep 3 - optional
Configure advanced optionsStep 4 - optional
Configure group size and scalingStep 5 - optional
Add notificationsStep 6 - optional
Add tagsStep 7
Review**Review** Info**Step 1: Choose launch template****Edit****Group details**Auto Scaling group name
MRSH Auto Scaling Group**Launch template**

Launch template	Version	Description
MRSHConfig	Default	lt-07b805d4a3b19ec5c

Step 2: Choose instance launch options**Edit****Network**Network
VPC
[vpc-057da990cd57d50c4](#)

Availability Zone Subnet

us-east-1a	subnet-07e01880deff01719	192.168.254.128/26
us-east-1b	subnet-0e30fcceb52971757	192.168.254.19

Instance type requirements

This Auto Scaling group will adhere to the launch template.

Step 3: Configure advanced options**Edit****Load balancing****Load balancer 1**

Name MRSHELB	Type Application/HTTP	Target group MRSHGroup1
---------------------------------	--------------------------	--

Figure 76: MRSH Auto Scaling Group

Scaling		
Minimum desired capacity 2	Maximum desired capacity 6	
Target tracking policy Policy type Target tracking scaling	Scaling policy name MRSHScalingPolicy	Execute policy when As required to maintain Average CPU utilization at 60
Take the action Add or remove capacity units as required	Instances need 300 seconds to warm up before including in metric	Scale in Enabled
Instance maintenance policy		
Replacement behavior No policy	Min healthy percentage -	Max healthy percentage -

Figure 77: Target Group MRSHGroup1 Scaling

VPC Considerations

Sunday, 19 October 2025 10:35 am

VPC Security Considerations

In a cloud environment, multiple VPC security features should be used to protect our instances and databases from attacks and unauthorised access. Figure 78 illustrates some common VPC security features.

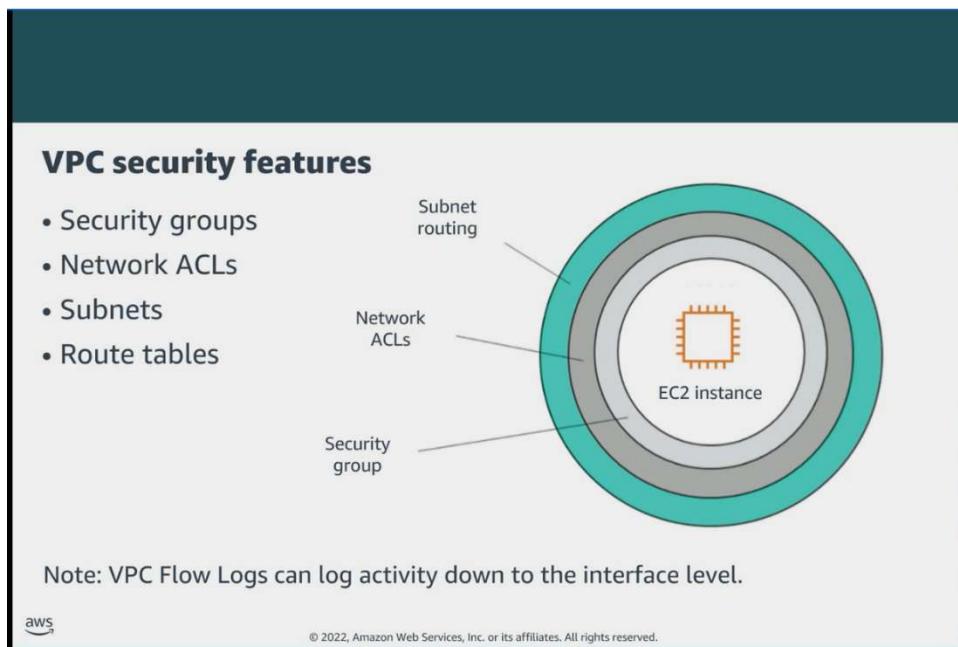


Figure 78: VPC security features

In the case of this AWS cloud environment, the following features were implemented:

Multiple Availability Zones (AZ): Subnets are spread across multiple AZs for high availability & durability, increased fault tolerance (in the event of an AZ/database instance failure) and scalability.

Use of private and public subnets: Private subnets enhance security as they are not directly exposed to the internet, thereby hindering potential unauthorised access attempts and/or cyber threats. Instances that do not need to be connected to the internet directly are placed in our private subnets. In this case, the DB Instance is located in the private subnets and connected to the EC2 web instance via security groups. All other access attempts and traffic are denied.

Security Groups: Security groups protect the instances by monitoring and controlling traffic in and out of it. The EC2 web server and RDS DB instances are assigned Security Groups that only allow the necessary traffic needed for their respective purpose.

Network ACLs: Network ACLs protect our subnets by controlling inbound and outbound traffic. The assigned ACL for our subnets allow certain inbound and outbound traffic, whilst denying all else.

CloudTrail and CloudWatch: These functions allow for monitoring and auditing of events and defined metrics within our cloudspace, sending out notification messages and activating alarms if need be, so that network administrators have high visibility of what is happening in the environment to take the necessary actions.

Other VPC Security Considerations

Due to either the account restrictions in place, or their unsuitability to this use case, the following are a few VPC security features that may be implemented in other cloud systems:

IAM: Due to account restrictions, I am unable to use IAM to create users, roles, and policies for restricted AWS service access within our VPCs.

Bastion Hosts: A bastion host withstands cybersecurity attacks and protects VPCs from security vulnerabilities and cybersecurity threats from the internet by filtering and directing user and network activity. Purposing an additional EC2 instance as a Bastion Host will incur costs and administrative duties and efforts.

Proxy Servers: Proxy servers are another layer of security as they act as a web filter for network users and the internal network. In our case, as we only have our web server on the cloud and they are usually placed on public subnets for direct access to the internet and lower costs than if they were to be placed in a private access (traffic through NAT gateway is charged).