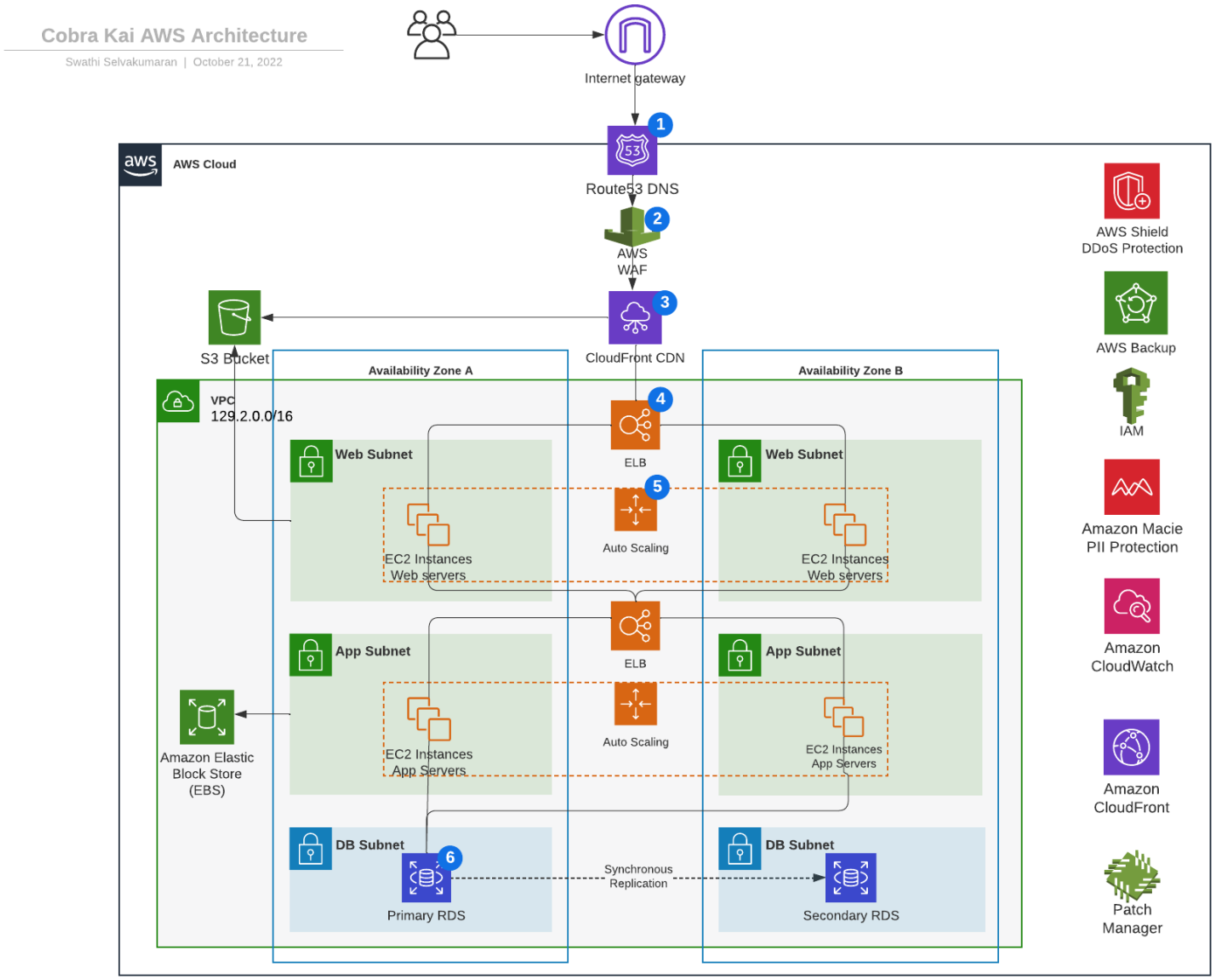
Cobra Kai Cloud Migration
ENPM665 – Final

Swathi Selvakumaran
119090206
swathi99@umd.edu

1.0 Introduction

The following documentation provides a high-level overview of the Cobra Kai application implementation in AWS. All the security key points have been considered while designing the application architecture.

2.0 Proposed Architecture



1. Amazon Route 53 routes the incoming user requests to the Cobra Kai application.
2. AWS Web Application Firewall (AWS WAF) validates the requests based on the defined rule set and forwards the Cobra Kai web requests to the Amazon CloudFront and Elastic load balancer.
3. AWS CloudFront handles the streaming of Cobra Kai tutorial videos to the users handling both static and dynamic stream content.

4. The Elastic Load Balancer automatically distributes the incoming cobra kai requests among the EC2 instances present in the availability zones.
5. Cobra Kai web servers and application servers are deployed on auto scaling groups and Amazon EC2 instances to automatically scale the application capacity.
6. The Amazon Relational Database Service securely stores the user and Cobra Kai application data.

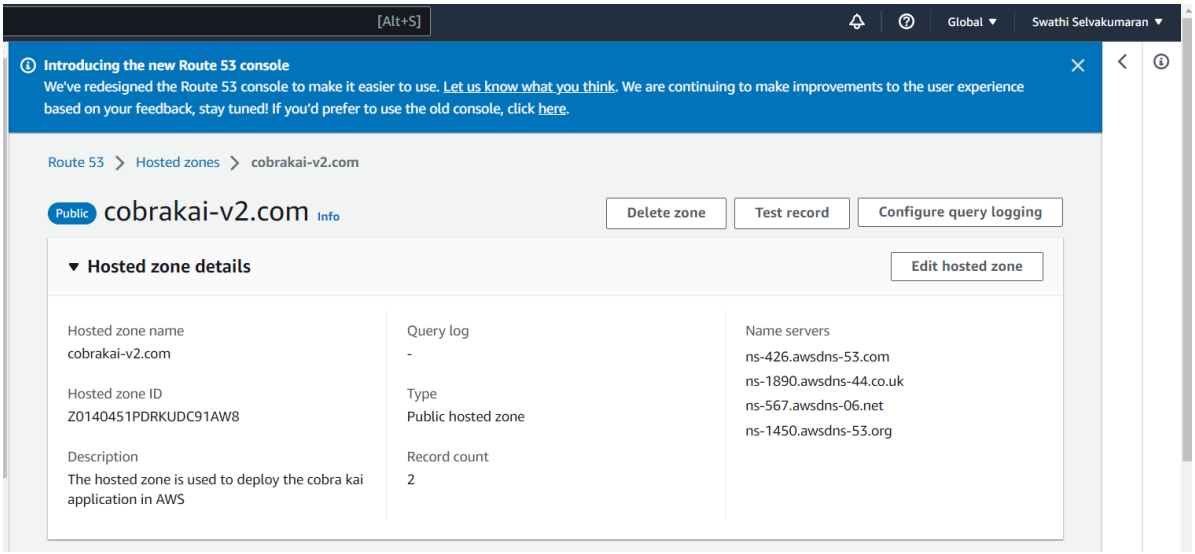
3.0 Amazon route 53

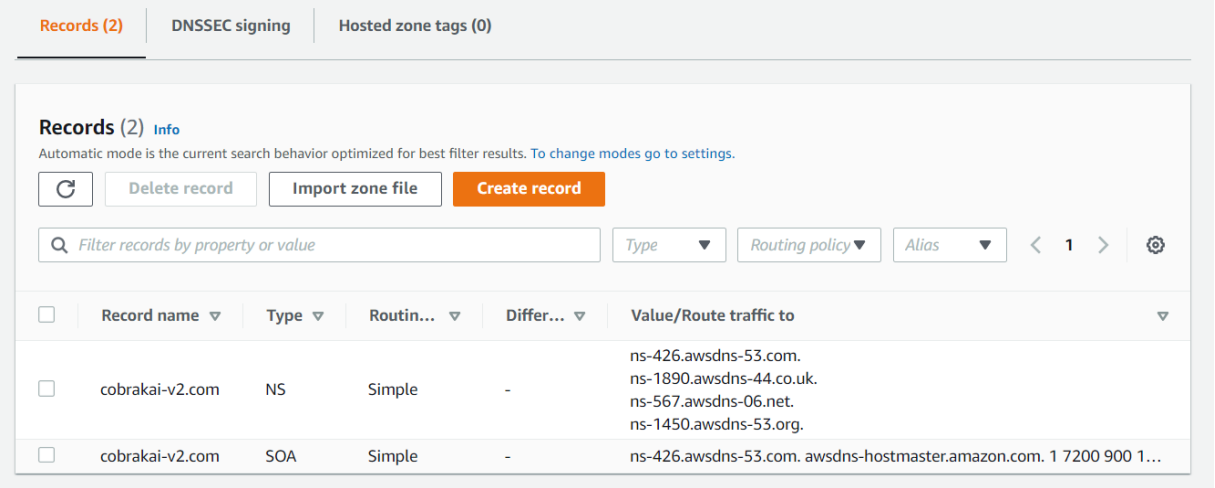
The user requests from the internet are routed to the Cobra Kai application hosted on AWS using the Amazon Route 53. Amazon Route 53 is a Domain Name System web service with reliable and cost-effective routing capability providing a connection between the end users and Cobra kai application. The user requests are routed to the AWS CloudFront and Elastic Load Balancer after verifying the request through the rules defined in AWS WAF. The domain name configured for the cobra kai application is: cobrakai-v2.com.

Domain Name registered for cobra kai application in AWS - cobrakai-v2.com

To configure DNS:

1. Select a unique domain name for Cobra Kai application and verify if it is valid.
2. Create the Domain name using AWS
3. Open Route 53 and enter the application domain name in the hosted zone and create the cobrakai-v2.com hosted zone.



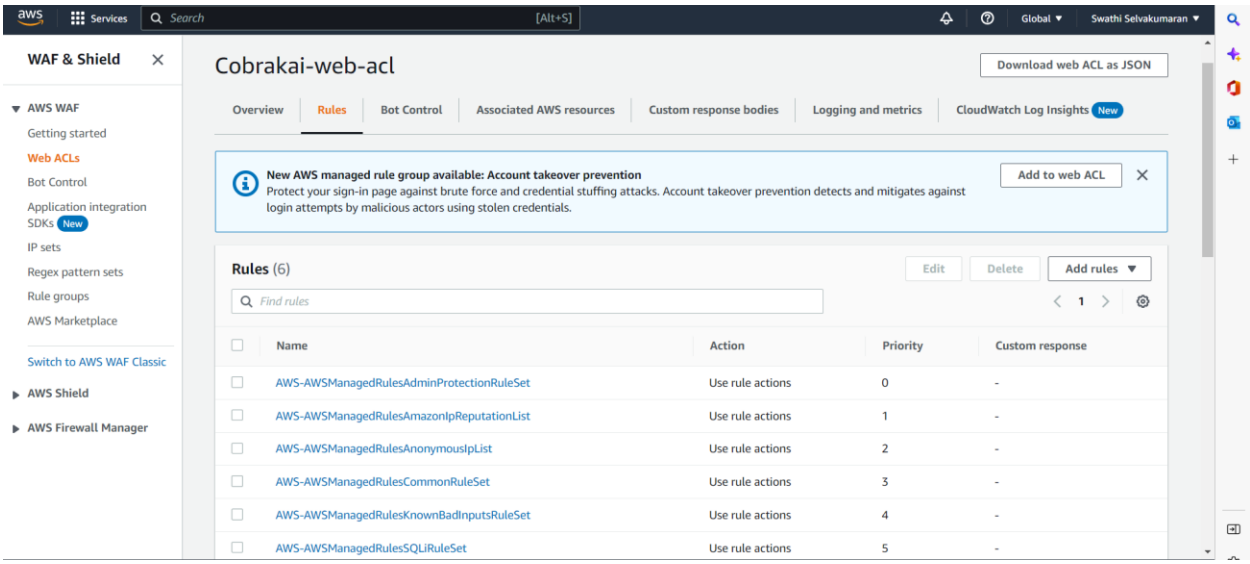


3.1 AWS Web Application Firewall (AWS WAF) - Preventing DDoS attacks

The AWS WAF is set in front of the AWS Cobra Kai application to monitor the web requests directed from the Route 53. These requests are validated by the rules defined int Web ACL. All the valid requests are then forwarded to the Amazon CloudFront distributions or an Application Load Balancer.

The rules defined in the Cobra Kai AWS WAF are:

- AWS-AWSManagedRulesAdminProtectionRuleSet
- AWS-AWSManagedRulesAmazonIpReputationList
- AWS-AWSManagedRulesAnonymousIpList
- AWS-AWSManagedRulesCommonRuleSet
- AWS-AWSManagedRulesKnownBadInputsRuleSet
- AWS-AWSManagedRulesSQLiRuleSet
- AWS-AWSManagedRulesWindowsRuleSet

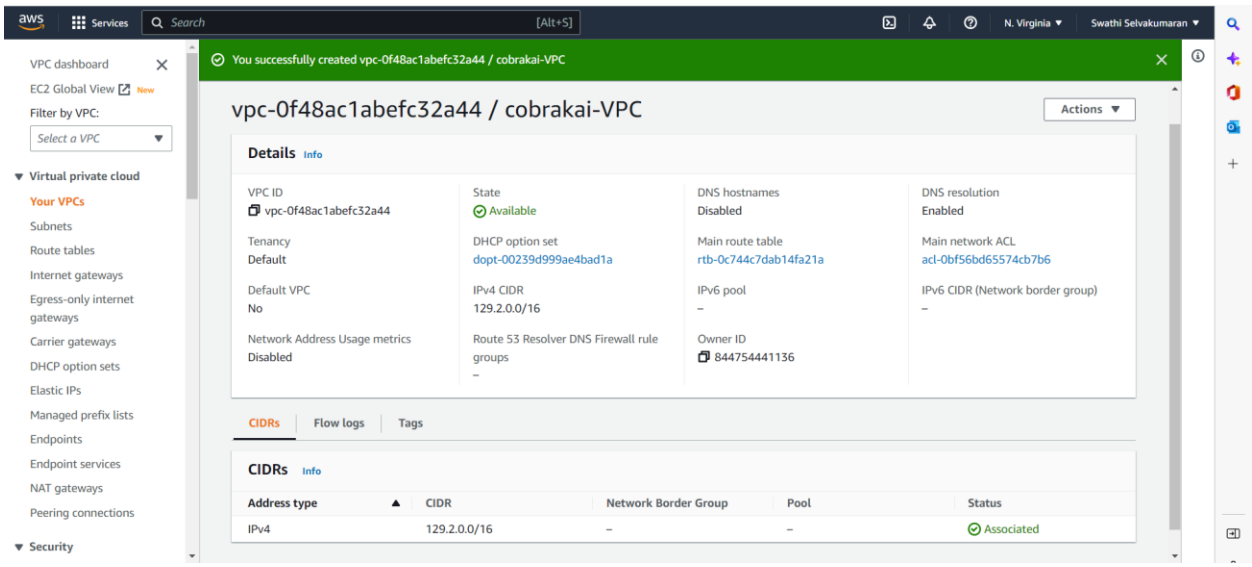


If the customer prefers to use a common firewall rule for multiple accounts and resources then the same rule set can be defined in the AWS Firewall Manager which will handle the administration and maintenance tasks across multiple accounts and resources.

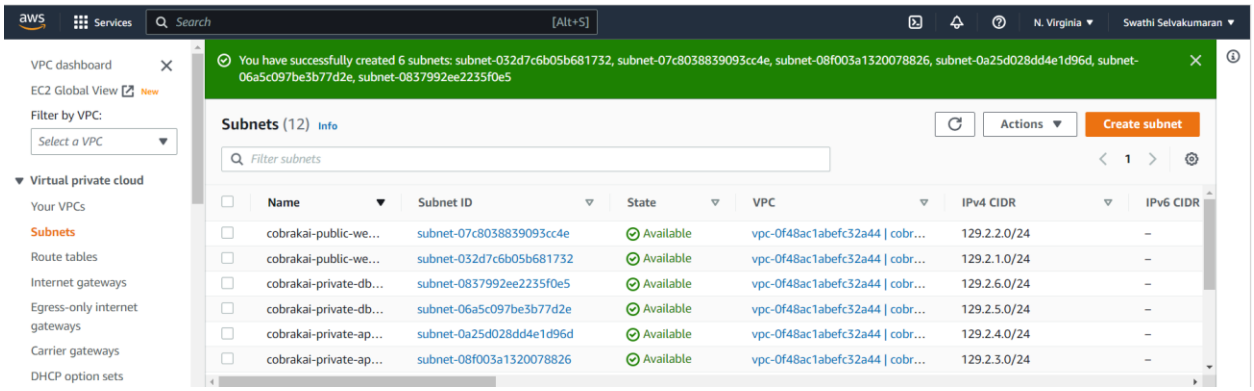
3.2 VPC

The AWS resources will be launched in the Amazon Virtual Private Cloud (VPC). For the Cobra kai application, the VPC is configured and will be launched in the corporate IP range 129.2.0.0/16. There are two availability zone for the cobra kai application. Each zone has a public web server, private app server and database. The incoming requests are distributed among the servers by the Elastic Load Balancer. The Amazon VPC should be configured by configuring the following:

1. ***Virtual Private Cloud*** – The VPC will be the data center for the Cobra kai application.
   To configure the VPC, open Amazon VPC service and select create a new VPC. Set the VPC name to cobrakai-VPC and configure the IP address range over which the application will be deployed as 129.2.0.0/16.

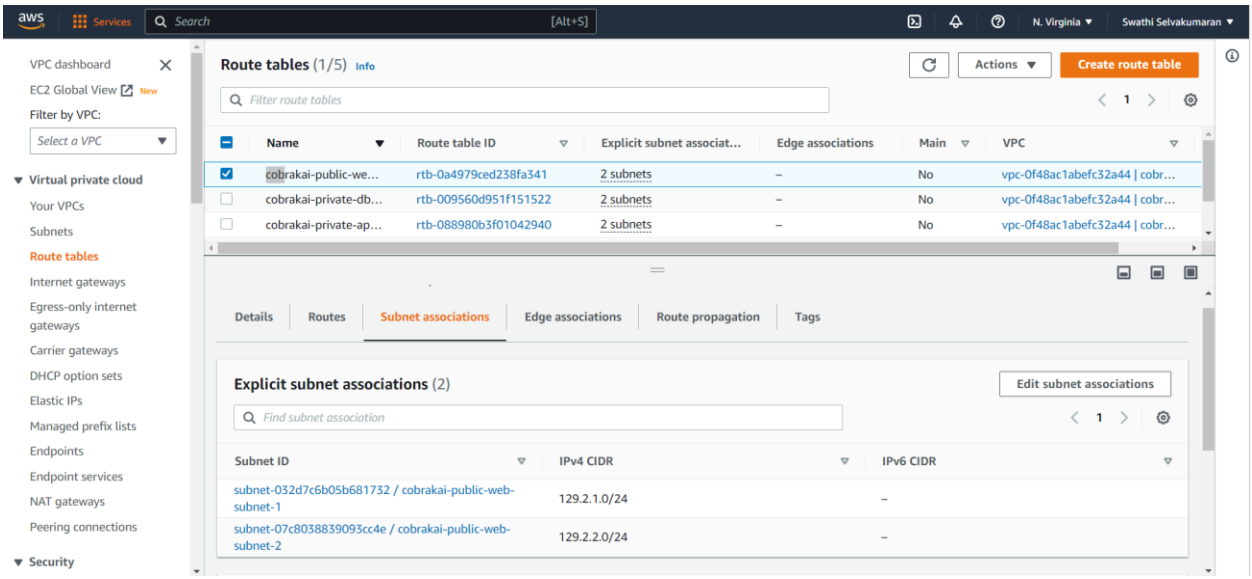


2. ***Subnet Configuration & IP Addressing*** – A specific range of IP addresses must be configured for the web server, app server and database. This configuration will define the range of IP addresses the request will be forwarded to for each server.
   For example, Here the configuration of the web and app servers to the following IP address range:
   1. cobrakai-public-web-server-subnet1 :- 129.2.1.0/24
   2. cobrakai-public-web-server-subnet2 :- 129.2.2.0/24
   3. cobrakai-private-app-server-subnet1 :- 129.2.3.0/24
   4. cobrakai-private-app-server-subnet1 :- 129.2.4.0/24
   5. cobrakai-private-db-subnet1 :- 129.2.5.0/24
   6. cobrakai-private-db-subnet2 :- 129.2.6.0/24



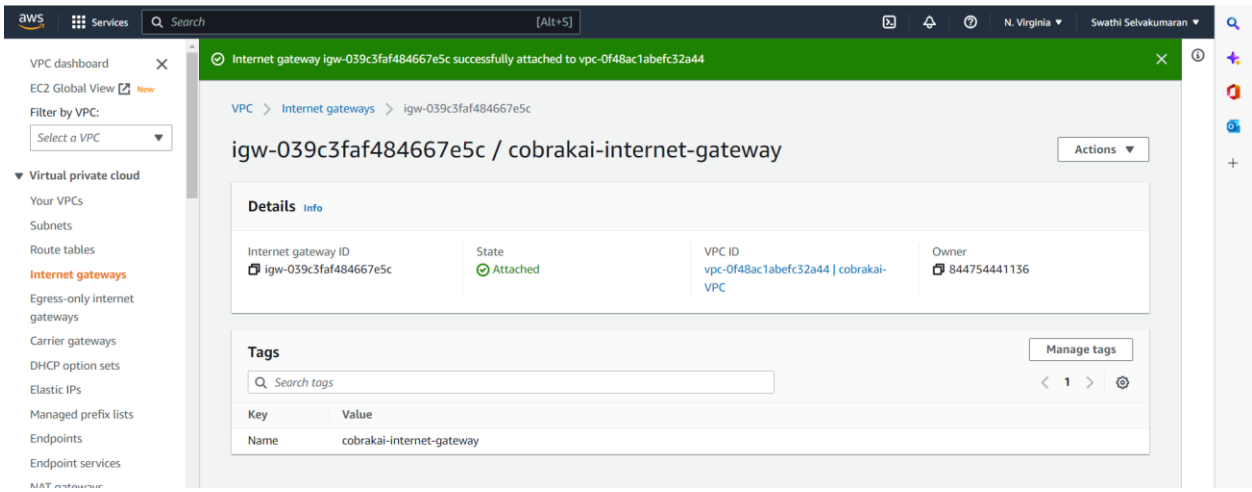
3. ***Routing*** – The route table consists of a set of rules that determines where the inbound and outbound subnet traffic is redirected to within the VPC.

Route tables must be created for the web server, app server and database and these tables must be configured to the respective subnets. The sample configuration of route tables can be found below.

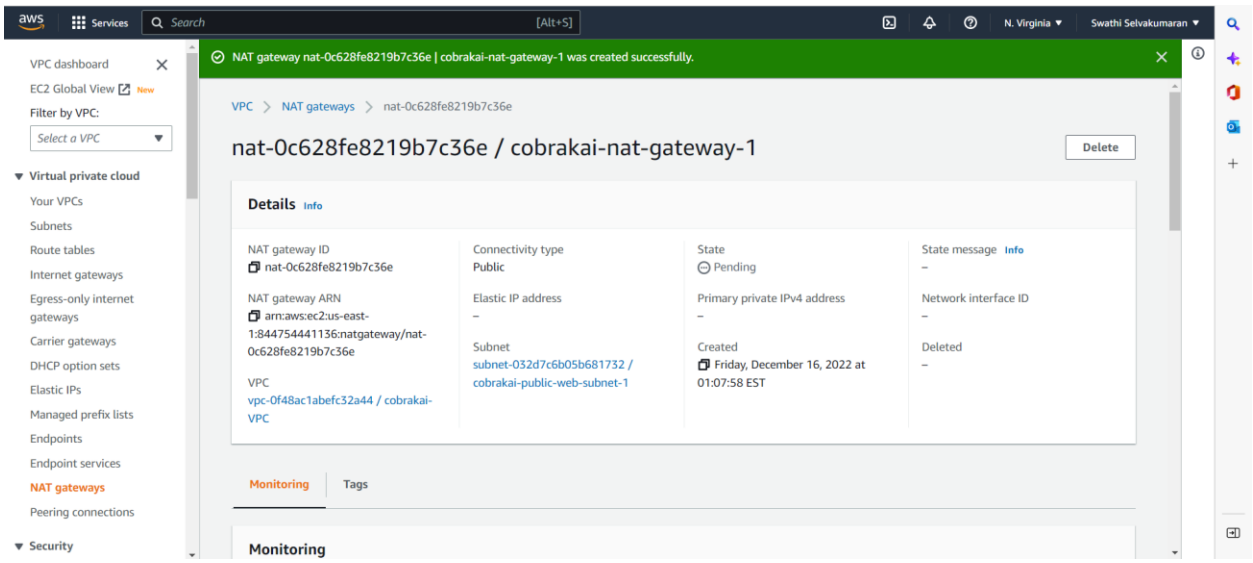


4. ***Gateways and end point*** – Gateway connect VPC to other networks and Endpoint connects to AWS services privately.

Internet gateway:



NAT gateway:

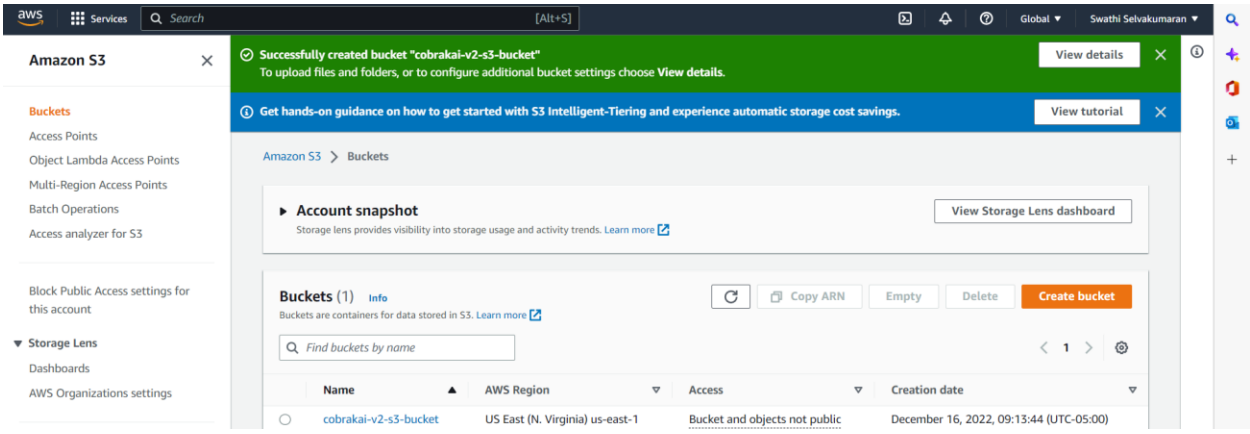


3.3.1 Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (Amazon S3) is an internet storage service. Data can be stored and retrieved with Amazon S3 at any time from any location on the internet. Amazon S3 is used for storage and backup in Cobra Kai Application.

A S3 bucket has been created for the Cobra Kai Application, named **cobrakai-v2-s3-bucket**. This bucket is configured to CloudFront and the Application web servers which is used for the storage and backup for the application.

The sample S3 bucket is:



3.3.2 Amazon CloudFront – Prevent Slow streaming

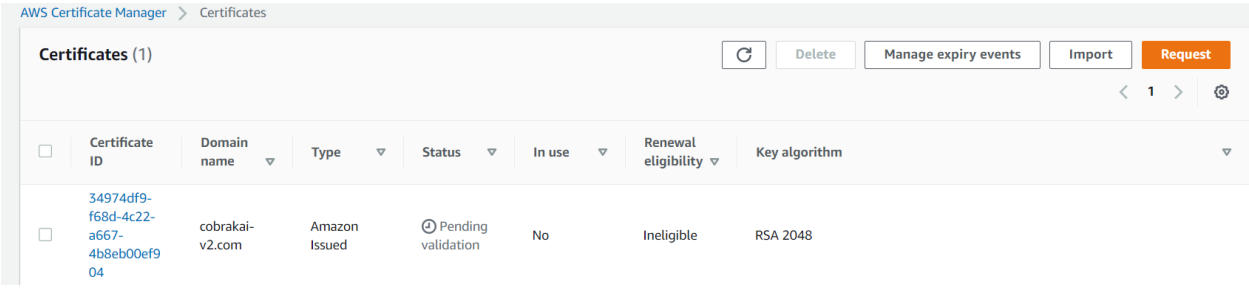
Amazon CloudFront allows for the high-speed distribution of both static and dynamic web content to users. The received requests are routed to the edge locations that offer the user requests with the least amount of delay. This guarantees that throughout content delivery, high performance is maintained.

Amazon CloudFront is configured with the Cobra Kai application for high-speed streaming of the Cobra Kai videos and for faster downloads and order processing.

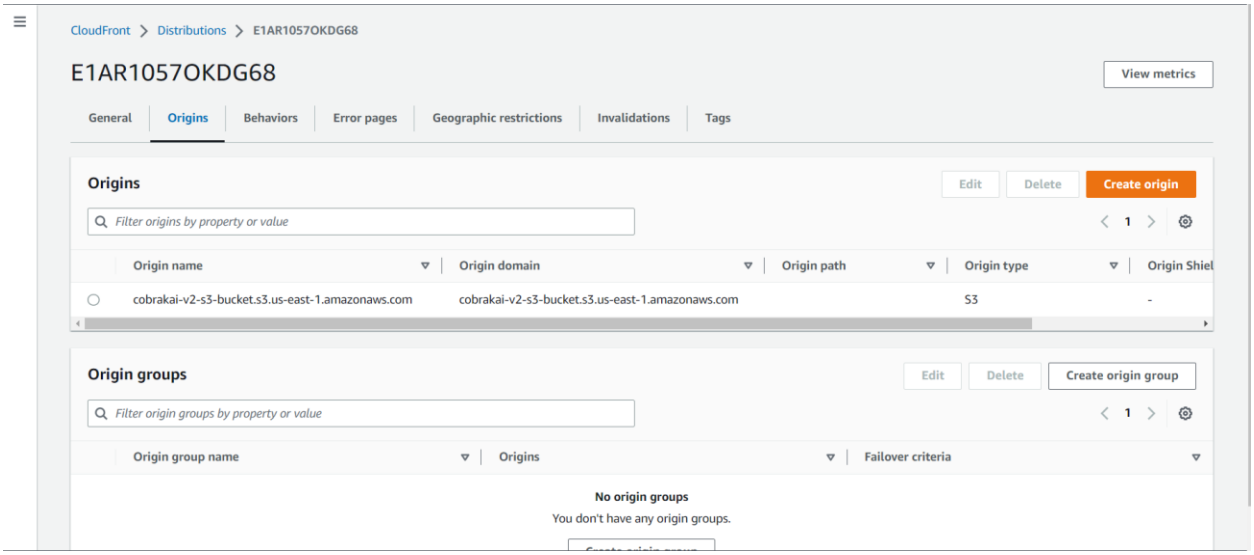
To Configure Amazon CloudFront:

1. Ensure that the S3 bucket for Cobra Kai application has been created as mentioned above.
2. Create the required .pem certificates using the Certificate Manager.

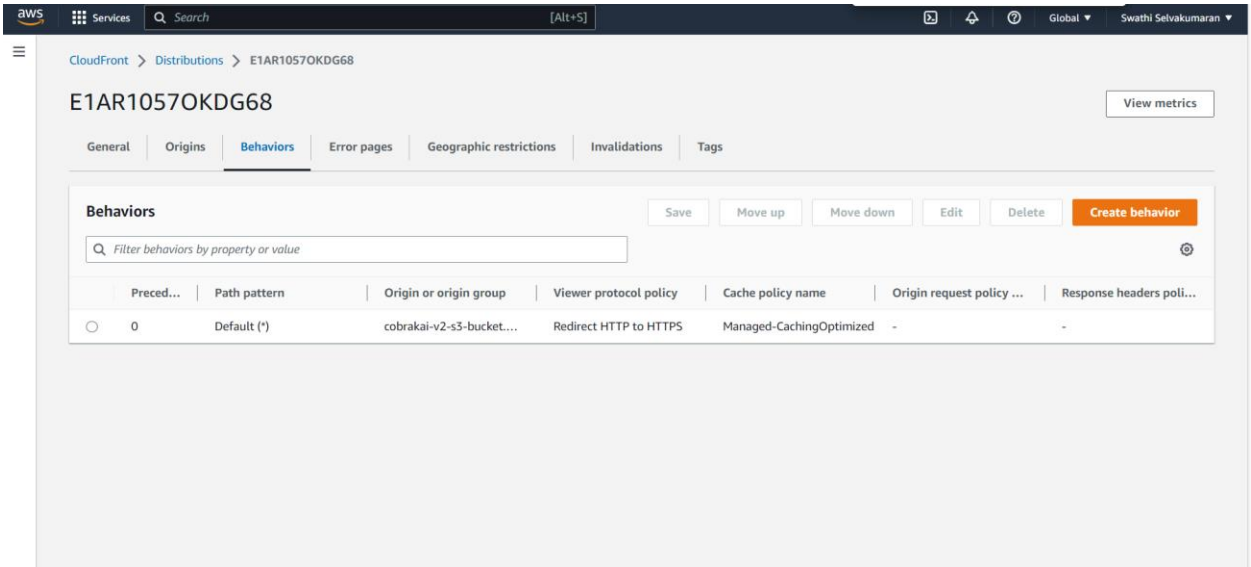
Certificate Manager:



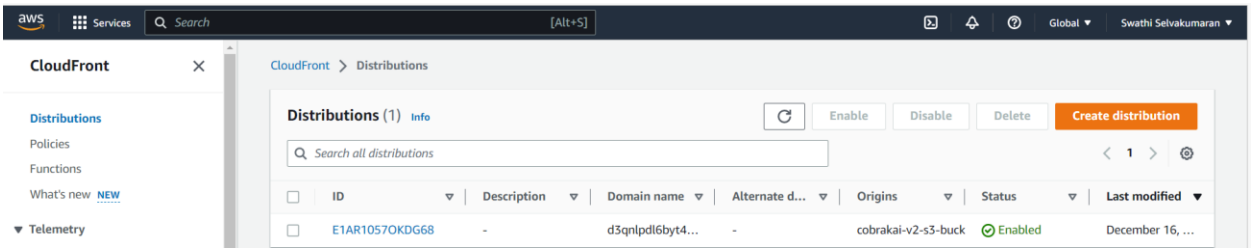
3. Go to CloudFront -> Distributions and create the Distribution using the ARN generated in the S3 bucket.



4. Configure the Cloud front to point to the S3 bucket, policy to Redirect all request from HTTP to HTTPS and configure the certificates.



5. After successful configuration create the CloudFront



3.4 Patching strategy – AWS System Manager Patch Management

AWS has automated patch manager that automates the patching process of the managing nodes. Patch manager handles the required security and operating system updates automatically. Patch Manager will automate the patching process for Cobra Kai Application by scanning all the instances for missing patches and installs any missing patches to the instances by using EC2 tags. AWS Codepipeline listens to the user inputs and once the code changes are provided the AWS Code build identifies the desired patching operation based on the target using resource groups and tags or patch groups. Patching can be then scheduled using the System manager to run in at predefined schedule. AWS System Manager controls the whole workflow of the patch operation.

Operations in patching strategy:

1. User defined, or default patch baseline is used
2. Select the instances to patch and configure the schedule and task.
3. Initiate the patching process.
4. Utilize Maintenance Windows to automate the patching
5. Keep track of patch status to guarantee compliance

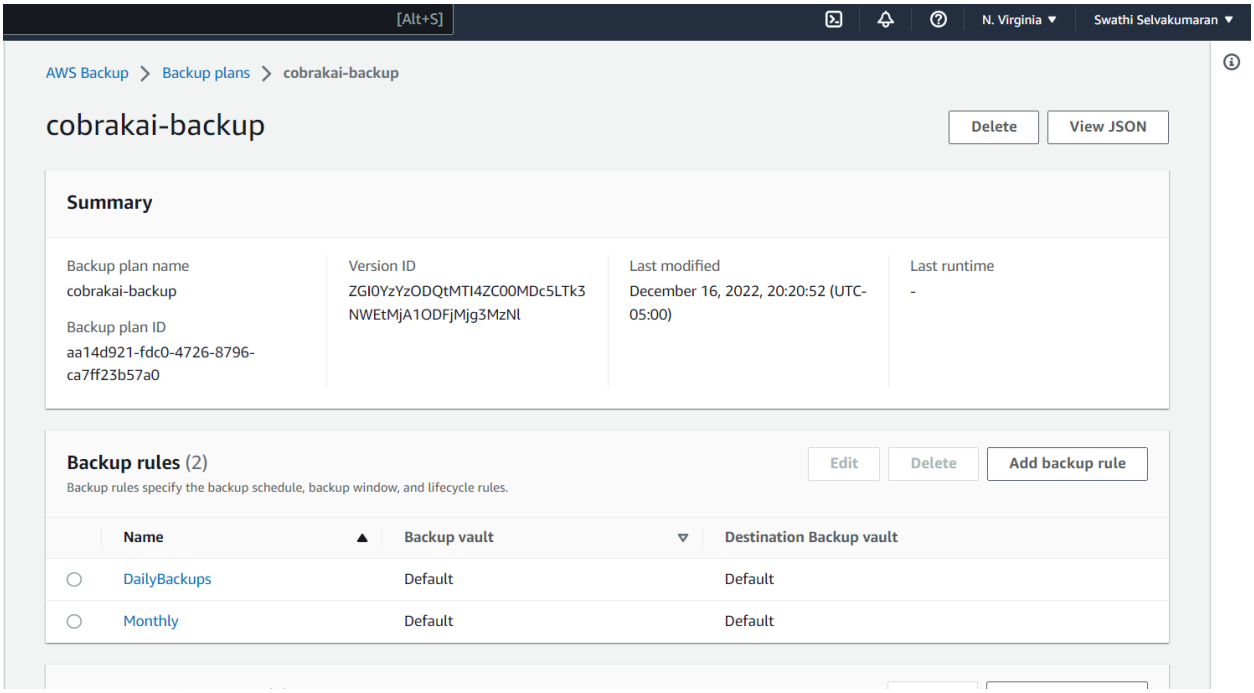
3.5 AWS Backup- Backup strategy

AWS Backup is a centralized service that automates data backup for all AWS services. The Cobra Kai application may use AWS Backup to centrally establish backup policies and track all resource backup activities.

To configure AWS Backup for the CobraKai application:

1. Navigate to AWS Backup service.
2. Create a new backup plan.
3. Assign the backup plan name as cobrakai-backup.
4. Assign the Backup rules based on the frequency of backup to take place.
5. Add the resources to which the backup plan must be implemented.

Here there are two backup rules for daily and monthly backup process. The resources included for back up are: EC2, Cloud front, S3 bucket, EBS. Resources can be added for backup if required.



3.6 AWS Shield - Preventing DDoS attacks

AWS Shield is a DDoS protection service available on AWS. AWS Shield Standard will protect the Cobra Kai application from DDoS attacks and prevents the AWS resources. AWS Shield is enabled by default, but the application resources must be added. Add the cobra kai application resources to be protected in the AWS Shield. If customer wants added protection the cobra kai application can be configured to the AWS Shield Advanced on subscription basis.

To configure the AWS, Shield Advanced: Go to AWS Shield service and opt for the AWS Shield Advanced plan configured for the Cobra Kai Application.

3.7 PII Protection and PCI DSS

3.7.1 Amazon Macie

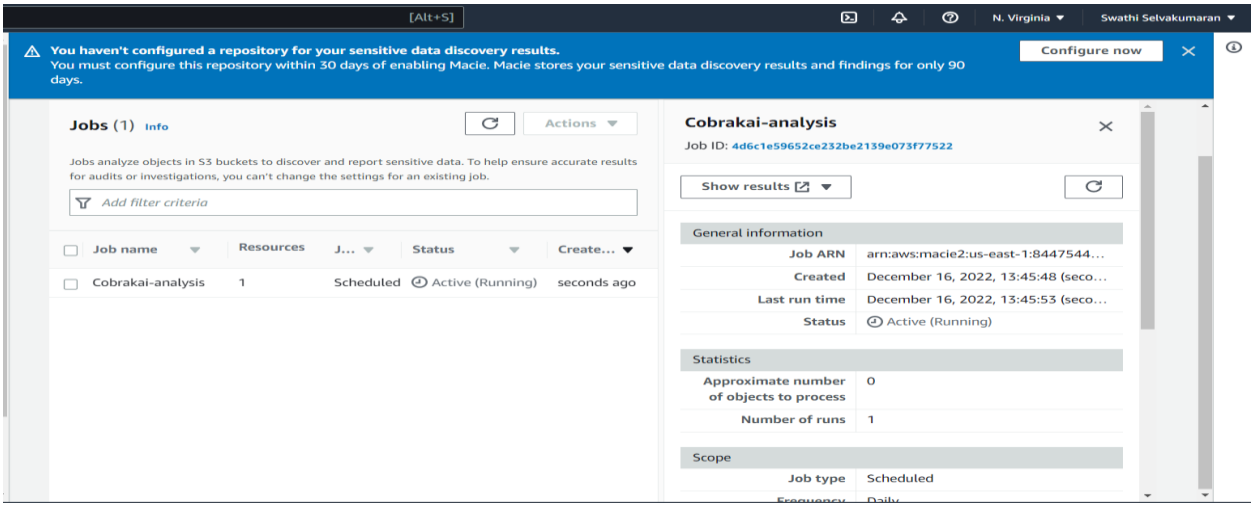
With the use of machine learning and pattern recognition, Amazon Macie's completely manages data security and privacy of the sensitive data stored in AWS. Amazon Macie continuously monitors and evaluates the Amazon S3 environment. Amazon Macie schedules data analysis of the application at regular intervals to ensure that the sensitive information is protected. Based on the that Amazon Macie generates the findings into a summary that consists of the following application data:

1. Maintains a list of sensitive datatypes that includes Personally Identifiable Information (PII) and other sensitive datatypes defined based on the privacy regulations of PCI DSS, GDPR and HIPAA.
2. Using Macie during data ingestion will determine the sensitivity of the data and protects it accordingly.
3. The findings of Macie are prioritized on the severity level categorized based on the data type, tags, encryption level, accessibility.

Cobra Kai Application should be integrated with Amazon Macie to protect the customers PII (name, phone, email, address and other customer details) and the Credit Card processing takes place based on the PCI DSS standards.

The Amazon Macie can be configured in the Cobra Kai Application by:

1. Enable Amazon Macie by selecting the service. This will enable the Connection for all the S3 buckets present.
2. Create a new job named Cobrakai-Analysis For monitoring the S3 buckets for sensitive data and set a scheduler for daily analysis.



3.7.2 PCI DSS certification – Cobra Kai compliance

Payment Card Industry Data Security Standard is a security standard set by the PCI security standards council. AWS has obtained the highest security certification as the PCI DSS Level 1 service provider. AWS does not directly store, transmit or process any credit card details. The AWS service provider will have to define the Cardholder Data Environment which can handle all the user credit card details.

To ensure the customer information security in Cobra Kai the PCI compliance must be defined as follows:

1. Building secure network and system – Data is secured by setting up the AWS Firewall around the Application.
2. Protecting Credit Card Information – Credit card details are encrypted during transactions as well as storage using nonreversible hash key.
3. Component security – The application should have regular scheduled scanning of all the components. The components should be updated regularly with latest security patches.
4. Access control – The Access to credit card information should be authenticated and authorized. Multi Factor Authentication should be set up to ensure secure payments.

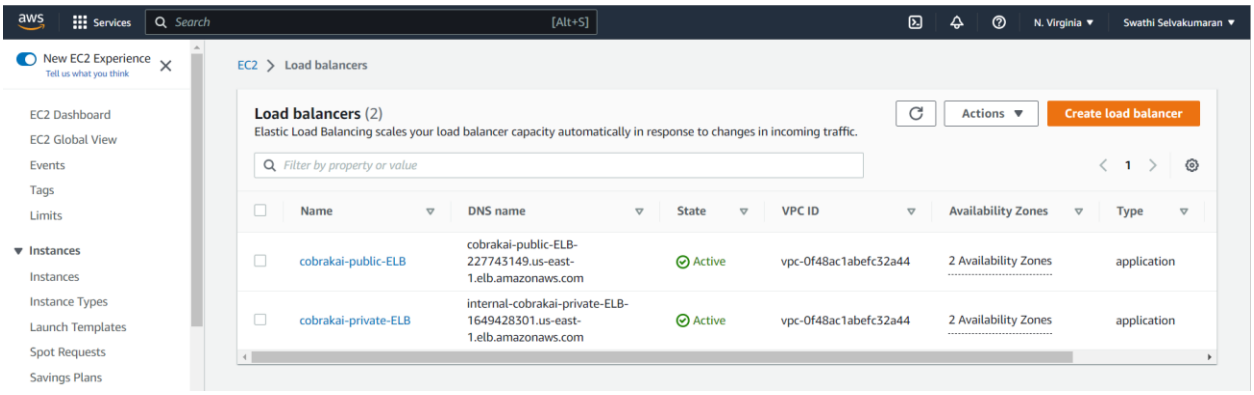
Cobra Kai Application can use secure third-party API that provides secure credit card payments.

3.8 Amazon EC2- Scalability

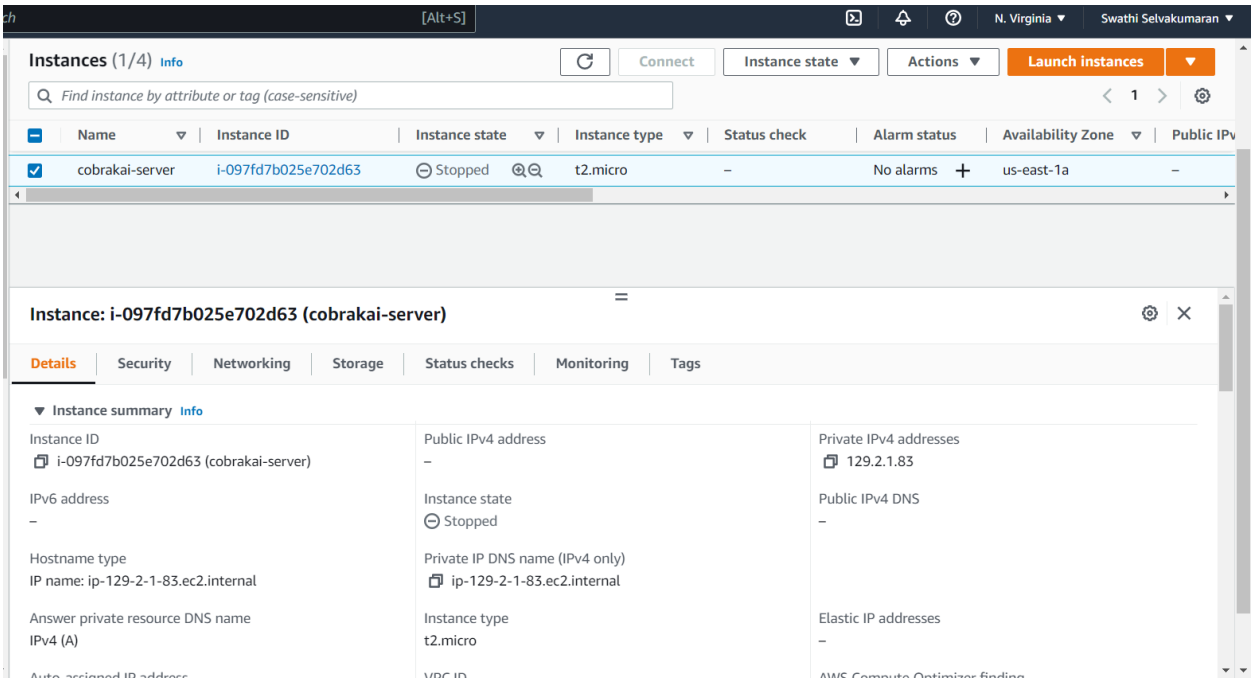
A scalable cloud computing capability is provided by Amazon Elastic Cloud Computing. The Cobra Kai Application should be hosted on the EC2 instances. The application must be connected to the Elastic Load balancer, Autoscaling and databases. EC2 instance will scale according to the required application capacity based on the demand preventing it from any hardware failures or human mistakes.

To configure EC2 instance for the cobra kai application:

1. Create public and private Elastic load Balancers for the web server and app server respectively. These servers must be configured to the VPC subnet zones created initially.



2. Select Launch Instances in the Amazon EC2.
3. Configure the instance as cobrakai-instance.
4. The configuration for EC2 instance will be Amazon Linux, t2.micro, select the VPC subnet to be configured, and create new key pair.
5. Launch the instance.

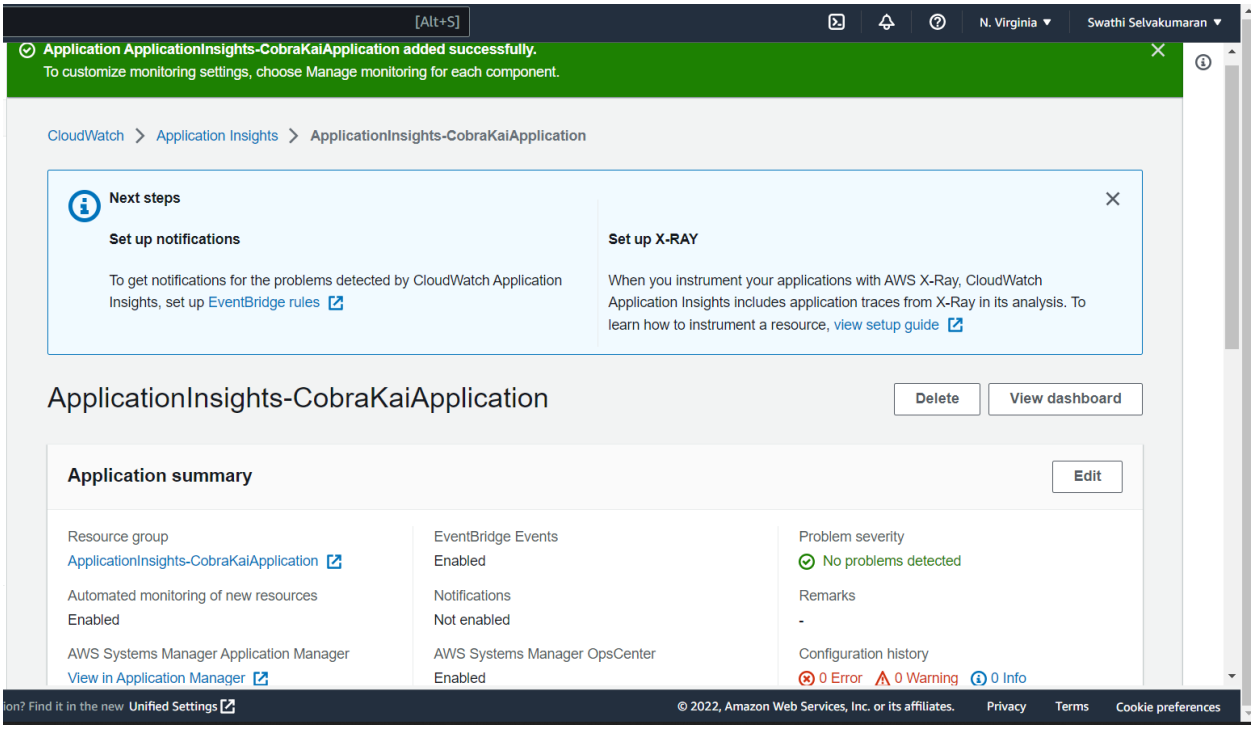


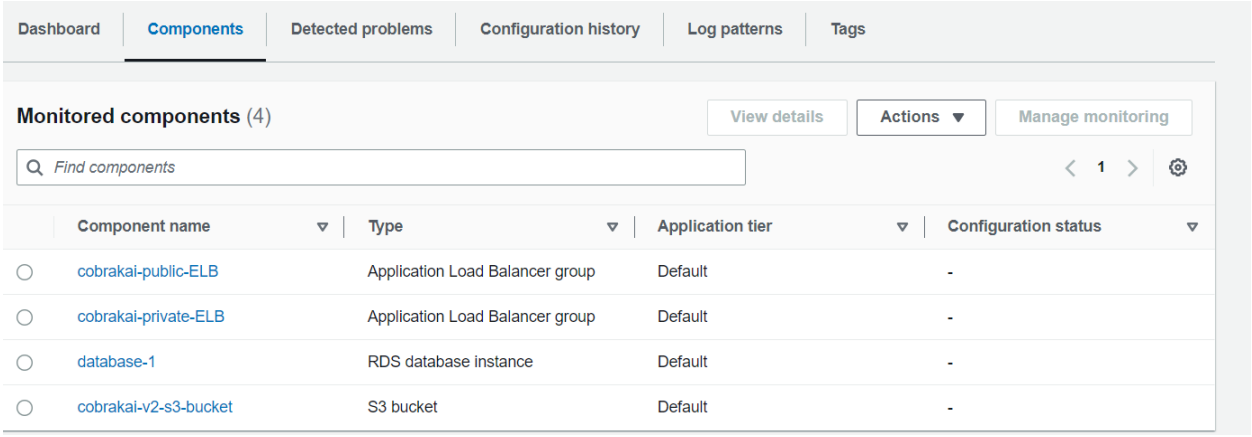
3.9 Amazon CloudWatch - Application Monitoring

The AWS resources utilized by the web application are tracked by Amazon CloudWatch. It gathers information, keeps logs, and monitors metrics. Utilizing Amazon CloudWatch, alerts may be configured to monitor that the thresholds for the metrics and actions are not exceeded. Amazon Cloud Watch must be configured in the Cobra Kai application for the logging and monitoring application, database, and workload.

To configure the Amazon Cloud watch in cobra kai:

1. Go to Amazon CloudWatch ->Application insights.
2. Add new Application for monitoring.
3. Create the application CobraKaiApplication
4. Add the cobra kai application S3 buckets, Elastic load balancer and database.
5. The Cloud watch will monitor the application.





3.10 Amazon EBS- Secure Storage

Scalable storage for Cobra Kai web application is provided through Amazon EBS. All the data in EBS is encrypted, and this encryption is maintained even when the data is transferred between EC2 instances. As a result, the Cobra Kai application will be protected from security vulnerabilities while streaming data.

3.11 Amazon Relational Database Service (RDS)

A scalable and maintainable relational database is created in the cloud by Amazon Relational Database Service (Amazon RDS). It carries out common database management tasks and offers an affordable, scalable capacity for a relational database. Cobra Kai is deployed on the Amazon RDS Data subnet.

To configure the RDS for Cobra Kai Application:

1. Go to the Amazon RDS and select Create database.
2. Select the database server of the customer choice.
3. Select the DB instances required.
4. Configure the Credentials and storage.
5. Add security groups to the database.
6. Configure the RDS with the Cobra Kai EC2 instance.
7. Create the database.

Services | Search [Alt+S] | N. Virginia | Swathi Selvakumaran

RDS > Databases > database-1

database-1

Modify | Actions

Summary

DB identifier	CPU	Status	Class
database-1	1.65%	Available	db.m6i.large
Role	**Current activity**	**Engine**	**Region & AZ**
Instance	0.00 sessions	MySQL Community	us-east-1a

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags

Connectivity & security

Endpoint & port

Endpoint
database-1.cvuhv1iu2ksn.us-east-1.rds.amazonaws.com

Networking

Availability Zone
us-east-1a

VPC

Security

VPC security groups
default (sg-0e5cf5e88d65b050f)
Active

Connectivity & security

Endpoint & port

Endpoint
database-1.cvuhv1iu2ksn.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1a

VPC
cobrakai-VPC (vpc-0f48ac1abefc32a44)

Subnet group
default-vpc-0f48ac1abefc32a44

Subnets
subnet-0837992ee2235f0e5
subnet-07c8038839093cc4e
subnet-08f003a1320078826
subnet-06a5c097be3b77d2e
subnet-032d7c6b05b681732
subnet-0a25d028dd4e1d96d

Network type
IPv4

Security

VPC security groups
default (sg-0e5cf5e88d65b050f)
Active

Publicly accessible
No

Certificate authority
rds-ca-2019

Certificate authority date
August 22, 2024, 12:08 (UTC-05:00)

Security group rules (2)

Filter by security group rules

Security group	Type	Rule
default (sg-0e5cf5e88d65b050f)	EC2 Security Group - Inbound	sg-0e5cf5e88d65b050f
default (sg-0e5cf5e88d65b050f)	CIDR/IP - Outbound	0.0.0.0/0

Replication (1)

Filter by replication

DB identifier	Role	Region & AZ	Replication source	Replication state	Lag
database-1	Instance	us-east-1a	-	-	-

3.12 AWS Identity and Access Management (IAM)

The access to the AWS services and resources is handled by the AWS Identity and Access Management (IAM). The access management for the Cobra Kai Leadership and development team should be configured as part of the AWS IAM.

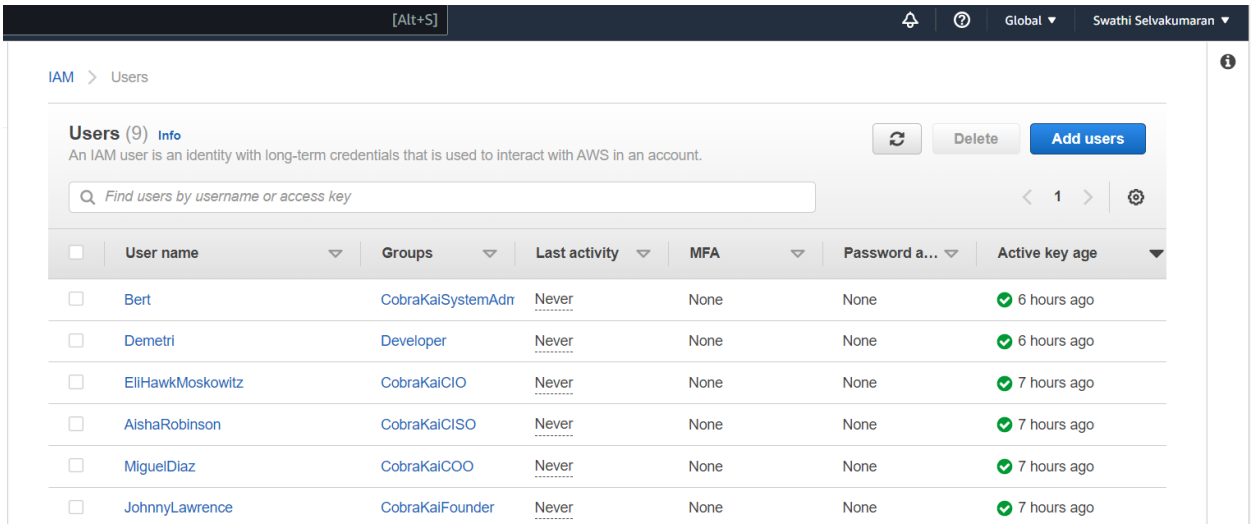
To manage the access management for the users:

1. Create the users in the AWS IAM ->users.
2. Create user groups and define the policies and roles based on specific group members.
3. Assign the users to respective user groups.
4. The users should be provided with least privileges possible based on their role to prevent possible security exploits.

The components of IAM are:

1. Central User Repository – stores and delivers Identity Information to the AWS services.
2. Authentication – Verifies the identity of the user with one or more authentication.
3. Authorization – Verifies if the user has permission to access the AWS service.
4. User Management – Manages the User lifecycle.

The following users were created, and the roles and permissions were assigned to the users.



3.12.1 Roles and Permissions for Cobra Kai leadership and development team

Johnny Lawrence

Job role: Founder of Cobra Kai

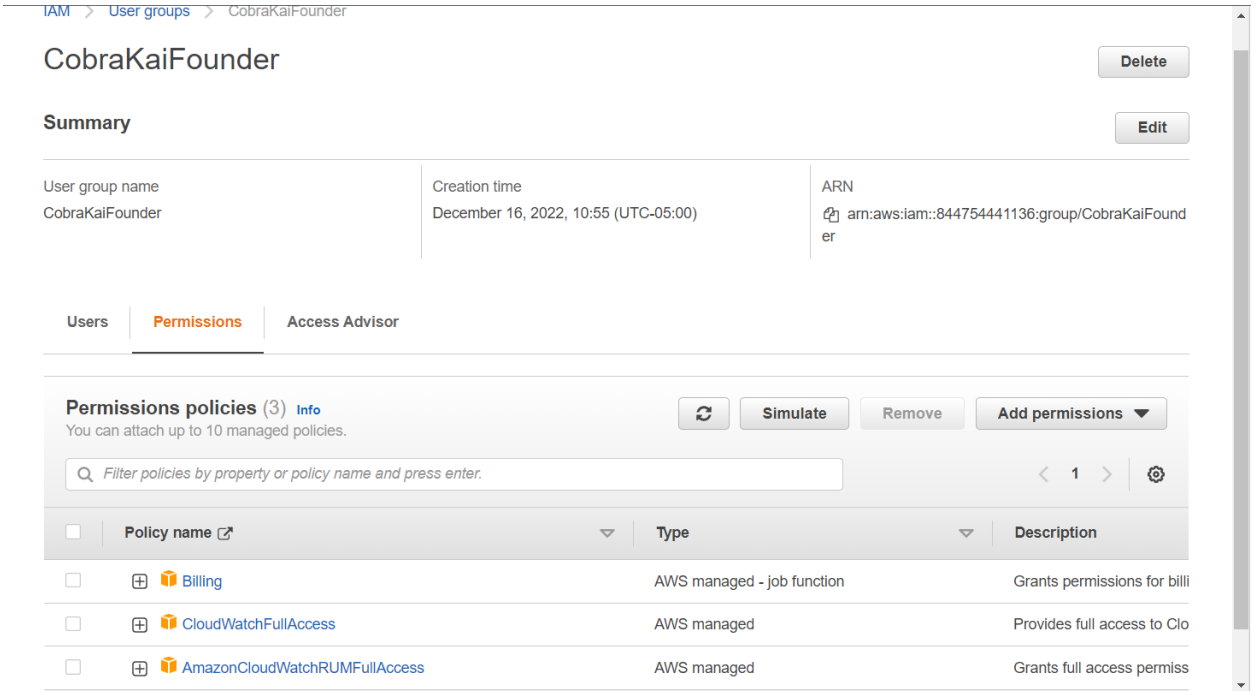
Privileges: As the founder of the Cobra Kai application is most exploitable person, He should be provided with the permission to view the organization expenses and monitor the events of the Cobra Kai application.

Roles: founder, visionary

Permissions:

The founder of organization should have the following permissions.

1. **Arn:aws:iam::aws:policy/job-function/Billing** – The following policy is assigned to the users who needs permission to view billing information, setting up payments and authorizing payments.
2. **arn:aws:iam::aws:policy/CloudWatchFullAccess –** This policy is assigned to user who need access to all application metrics and to track the customer information.



Miguel Diaz

Job role: Chief Operating Officer

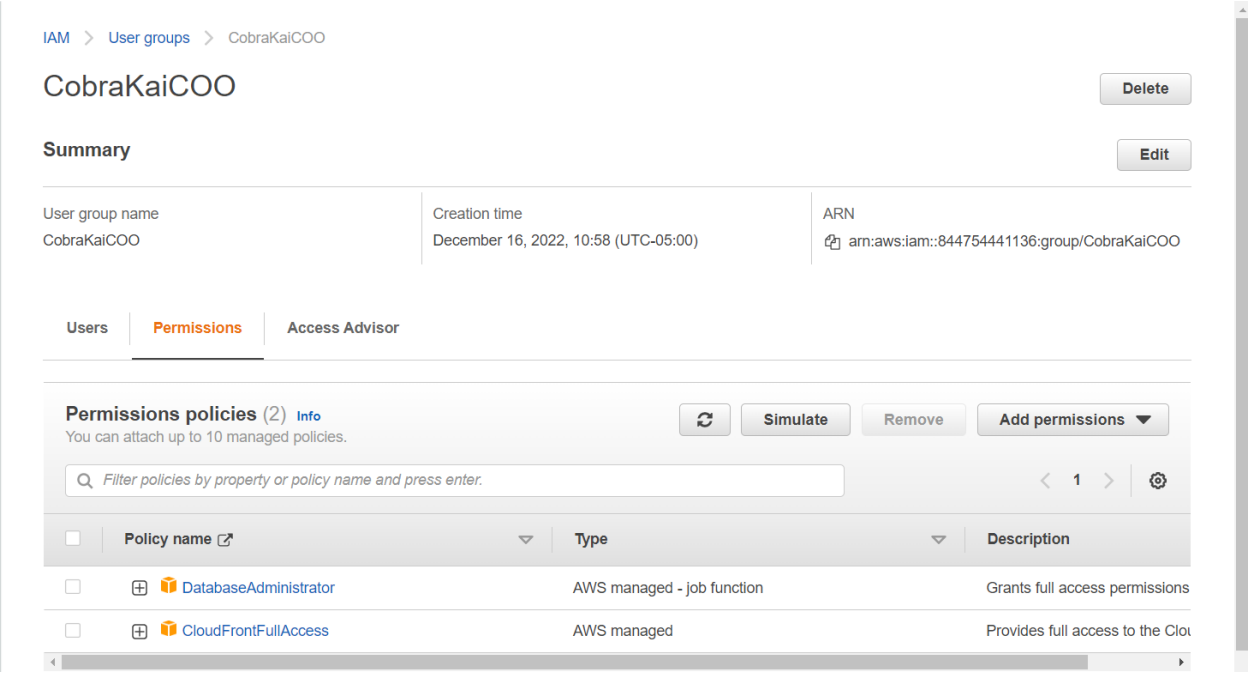
Roles: ChiefOperatingOfficer

Privileges: The Chief Operating Officer of Cobra Kai application should have access to the application streaming platform and the application

Permissions:

The COO of the organization should have the following policies:

1. arn:aws:iam::aws:policy/CloudFrontFullAccess – The user has complete access to the S3 buckets and CloudFront console for access the data.
2. arn:aws:iam::aws:policy/job-function/DatabaseAdministrator – The user has access to the Amazon RDS and they can establish, configure and maintain the application databases.



Aisha Robinson

Job role: Chief Information Security Officer – monitors and mitigates application vulnerabilities

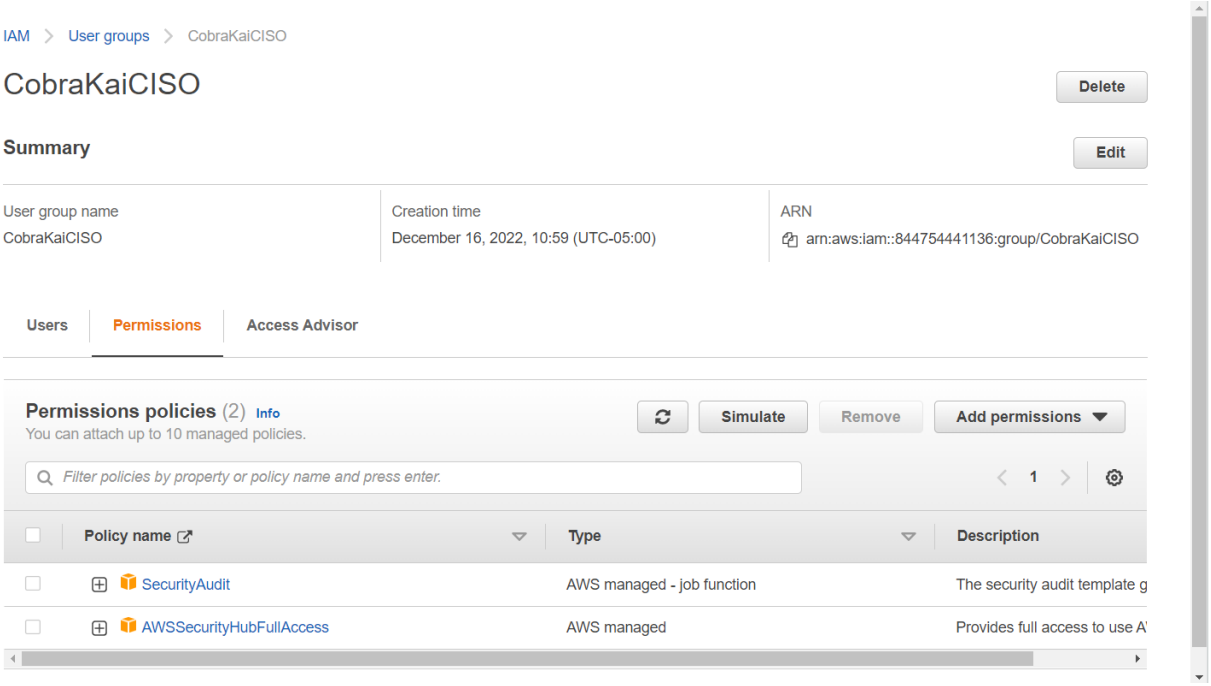
Roles: ChiefInformationSecurityOfficer

Privileges: monitor the application security

Permissions:

The CISO of the company should have the following policies:

1. arn:aws:iam::aws:policy/AWSSecurityHubFullAccess – The user should have the complete access to the application security hub and should be able to monitor any threats and prevent the application from any vulnerability.
2. arn:aws:iam::aws:policy/SecurityAudit – The user will be able to monitor the account for security compliance and will have access to the logs and events to check for security vulnerabilities.



Eli "Hawk" Moskowitz

Job role: Chief Information Officer – designs, develops and maintains the application codebase. Monitors the code checked-in.

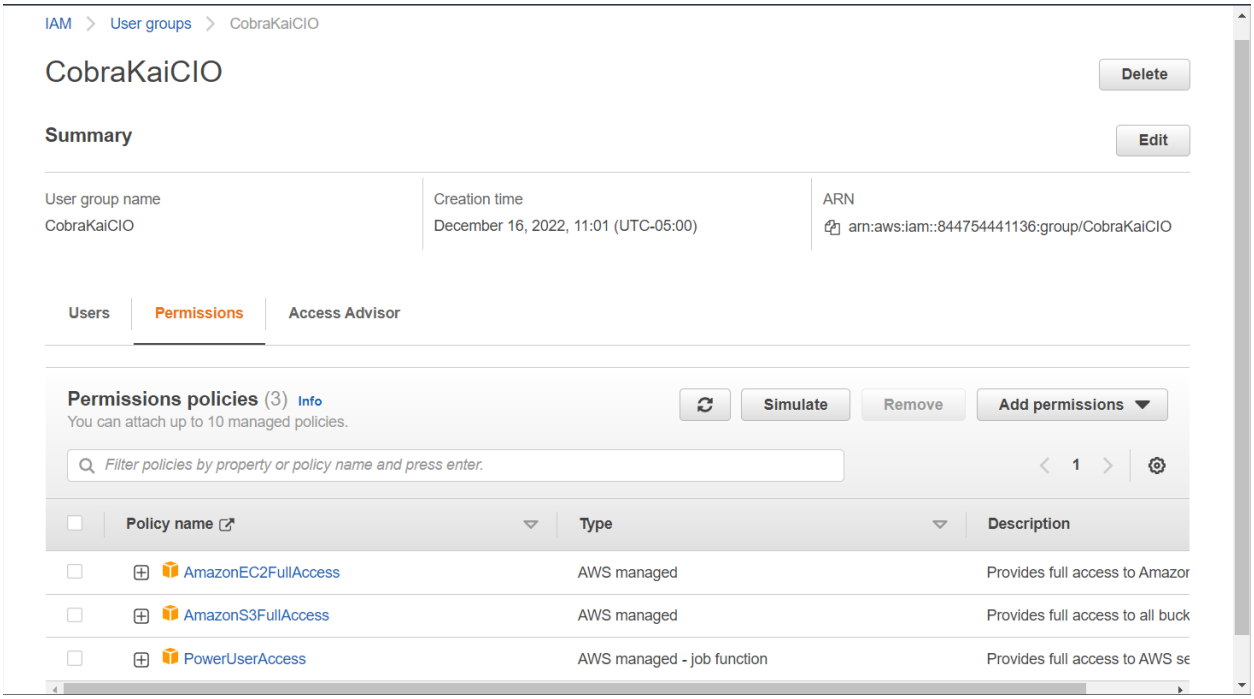
Roles: ChiefInformationOfficer

Privileges: Access to codebase, S3 buckets and EC2 instances

Permissions:

The CIO should have the following policies:

1. arn:aws:iam::aws:policy/PowerUserAccess – The user has all the permission to perform development tasks. The user can create and configure services and resources of the application.
2. arn:aws:iam::aws:policy/AmazonS3FullAccess - This policy provides full access to all the S3 buckets accessible through the AWS Management Console. "s3:*" denotes the complete user accessibility to S3 buckets of the application.
3. arn:aws:iam::aws:policy/AmazonEC2FullAccess - This policy provides full EC2 Access in any region. It provides accessibility to all ec2 instances, Elastic Load Balancing, Cloud Watch, Autoscaling.



Demetri

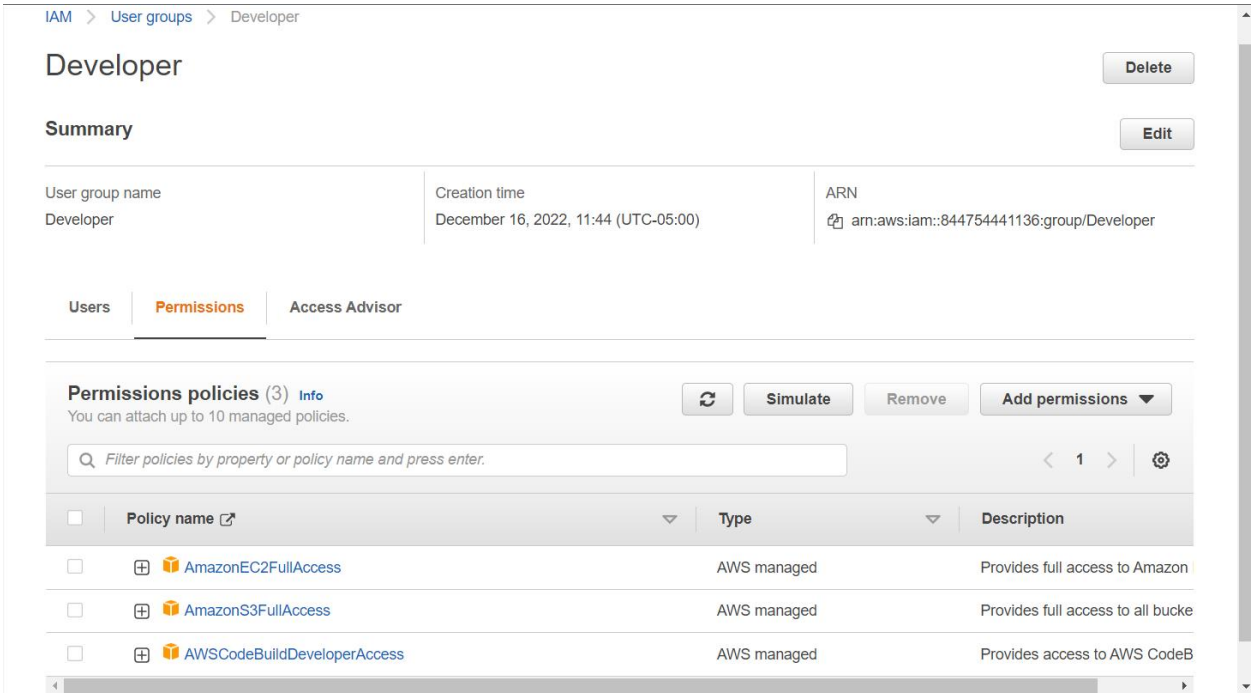
Job role: Web developer – develops the application code and maintains the application codebase.

Roles: developer

Privileges: Access to codebase, S3 buckets and EC2 instances

Permissions:

1. arn:aws:iam::aws:policy/AWSCodeBuildDeveloperAccess- The user has permission to view and edit the application codebase.
2. arn:aws:iam::aws:policy/AmazonS3FullAccess - This policy provides full access to all the S3 buckets accessible through the AWS Management Console. "s3:*" denotes the complete user accessibility to S3 buckets of the application.
3. arn:aws:iam::aws:policy/AmazonEC2FullAccess - This policy provides full EC2 Access in any region. It provides accessibility to all ec2 instances, Elastic Load Balancing, Cloud Watch, Autoscaling.



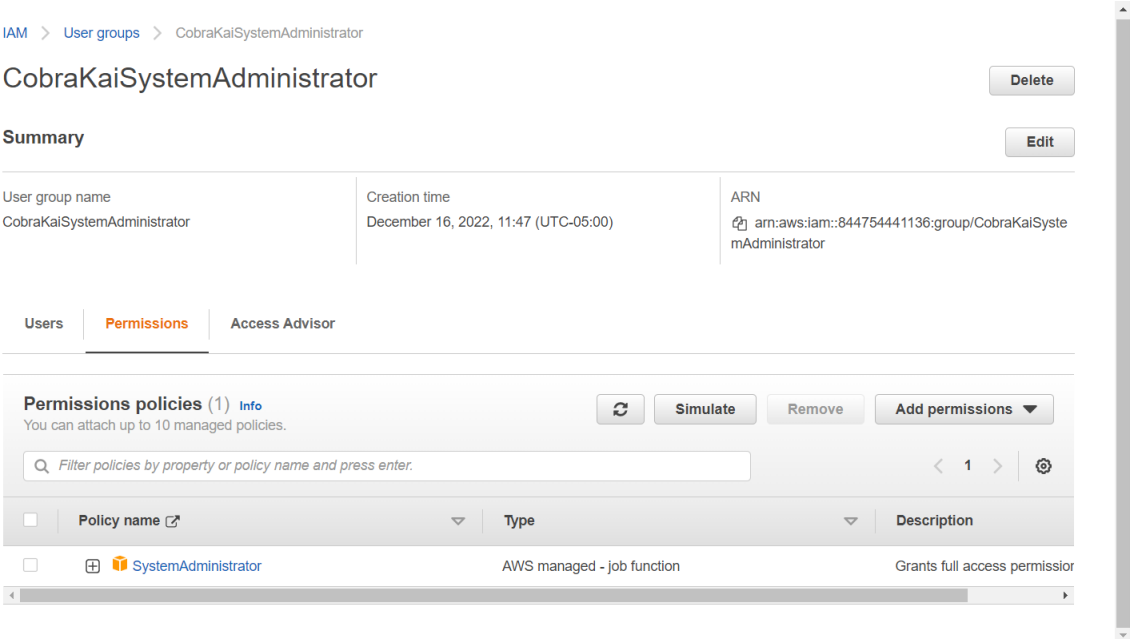
Bert

Job role: System administrator – highest security permission with access to all AWS services.

Roles: SystemAdministrator

Privileges: Complete System privileges.

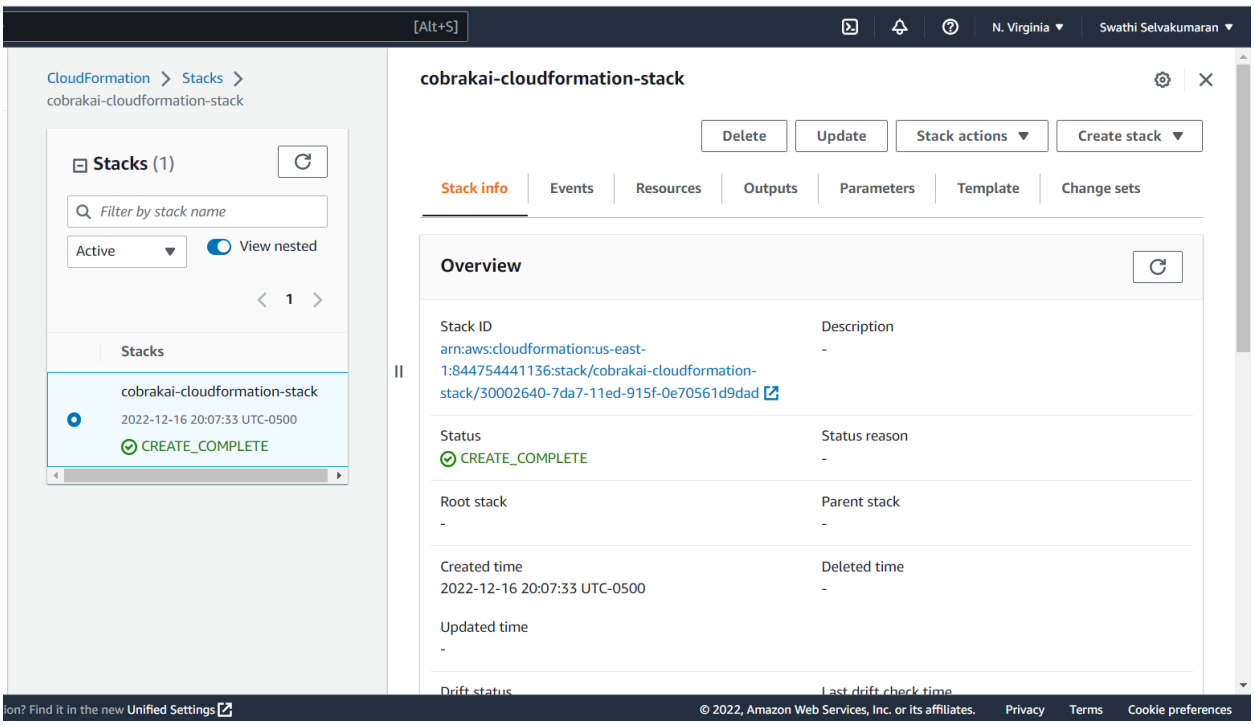
Permissions: arn:aws:iam::aws:policy/job-function/SystemAdministrator – The user has permission to all the AWS services like Amazon EC2, AWS IAM, Amazon RDS, AWS CodeCommit, AWS CodeDeploy, Amazon CloudWatch, and Amazon VPC. The policy also grants iam:GetRole and iam:PassRole for the following roles: ecr-sysadmin-*, rds-monitoring-role,ec2-sysadmin-*, lambda-sysadmin-*.



3.13 AWS CloudFormation

AWS CloudFormation is a template that consists of the AWS and third-party resources and provision them consistently. Using this template the multiple resources present in multiple AWS account and AWS region can be maintained in one stack. The Cloud Formation template for the Cobrakai Application was implemented by configuring the EC2 instances, VPC, RDS. This template can be used to immediately configure the resources in other AWS accounts.

Sample template: CobraKaiCloudFormationTemplate.pdf (Command Line)

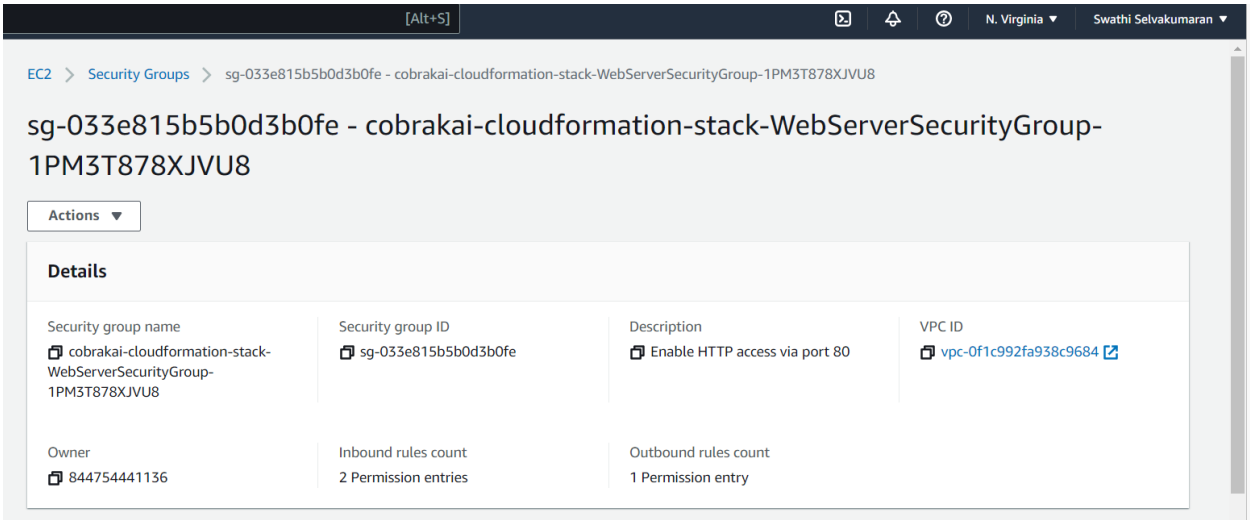


3.14 Autoscaling

AWS Auto Scaling will monitor the incoming requests and responses and adjusts the capacity to maintain a consistent performance with low cost.

3.15 Security Group

Security group is created for the EC2 instances to protect the instances. These rules ensure that only valid requests are received to the application.



Inbound rules | Outbound rules | Tags

Tags

Manage tags

1

Key	Value
aws:cloudformation:stack-name	cobrakai-cloudformation-stack
aws:cloudformation:stack-id	arn:aws:cloudformation:us-east-1:844754441136:stack/cobrakai-cloudformation-stack/30002640-7da7-11ed-915f-0e70561d9dad
aws:cloudformation:logical-id	WebServerSecurityGroup

Inbound rules | Outbound rules | Tags

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Outbound rules (1/1)

Manage tags | Edit outbound rules

Filter security group rules

1

Name	Security group rule...	IP version	Type	Protocol	Port range
–	sgr-0acb88ce849bf8f8d	IPv4	All traffic	All	All

Inbound rules | Outbound rules | Tags

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Inbound rules (2)

Manage tags | Edit inbound rules

Filter security group rules

1

Name	Security group rule...	IP version	Type	Protocol	Port range
–	sgr-0d7f5019a108088a4	IPv4	HTTP	TCP	80
–	sgr-076c33da8aef1a0c3	IPv4	SSH	TCP	22

References

[1] AWS Documentation - **AWS Documentation (amazon.com)**
[2] ENPM665 week 1 – week 13 Presentations.
[3] Lucid chart – AWS web application hosting.
[4] AWS Policies
[5] https://docs.aws.amazon.com/Route53
[6] https://docs.aws.amazon.com/AWSEC2
[7] https://docs.aws.amazon.com/IAM
[8] https://docs.aws.amazon.com/waf
[9] https://docs.aws.amazon.com/autoscaling/ec2
[10] https://docs.aws.amazon.com/AmazonRDS
[11] https://docs.aws.amazon.com/vpc