ENPM665 – Final

Om Venugopal

omvg@umd.edu

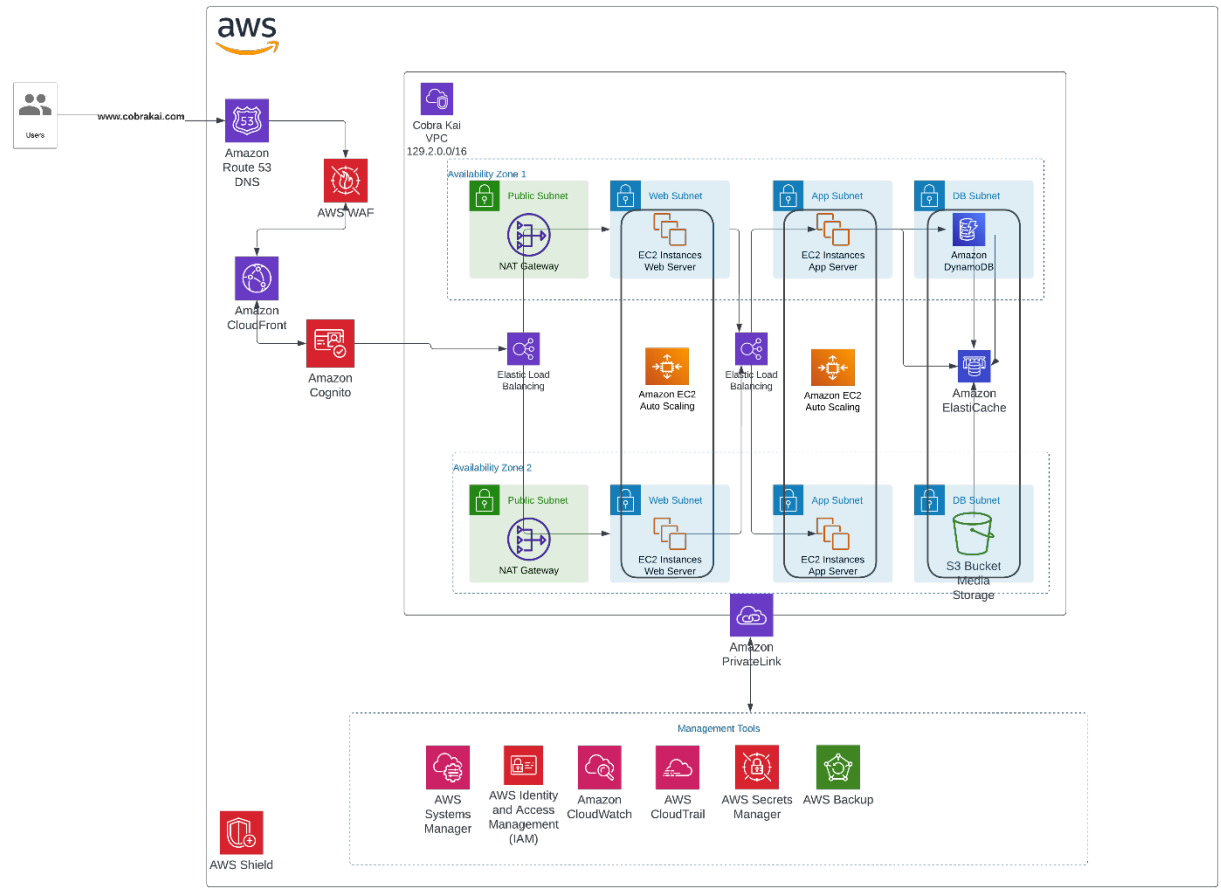
118116655

Contents

Introduction

The goal of this document is to help Cobra Kai build secure web application for its business. The documents explain the technical features of the recommendations that was provided and aims to help cobra Kai setup a successful secure working web application.

The proposed architecture meets all the requirements put forth by CobraKai. It aims to reduce latency, provide security against DDoS, meets their scaling and resiliency needs. The incoming traffic requests from the internet is handled by Amazon 53 and the CloudFront CDN which are all protected by the AWS Shield and the AWS WAF. The Firewall allows only the traffic that meets the configuration to the CDN to the Amazon Cognito. This is where the user authentication takes place. Once authenticated the user requests are routed to the web server. The request goes to the private subnet through the NAT gateway in the public subnet. The media is stored in the S3 bucket and metadata in the DynamoDB. The CDN streams the on-demand video to the user with high processing speed. Other services such as AWS IAM, AWS CloudWatch, CloudTrail, AWS Systems Manager are configured to keep the application secure and updated regularly. The logs are maintained and stored for auditing needs.



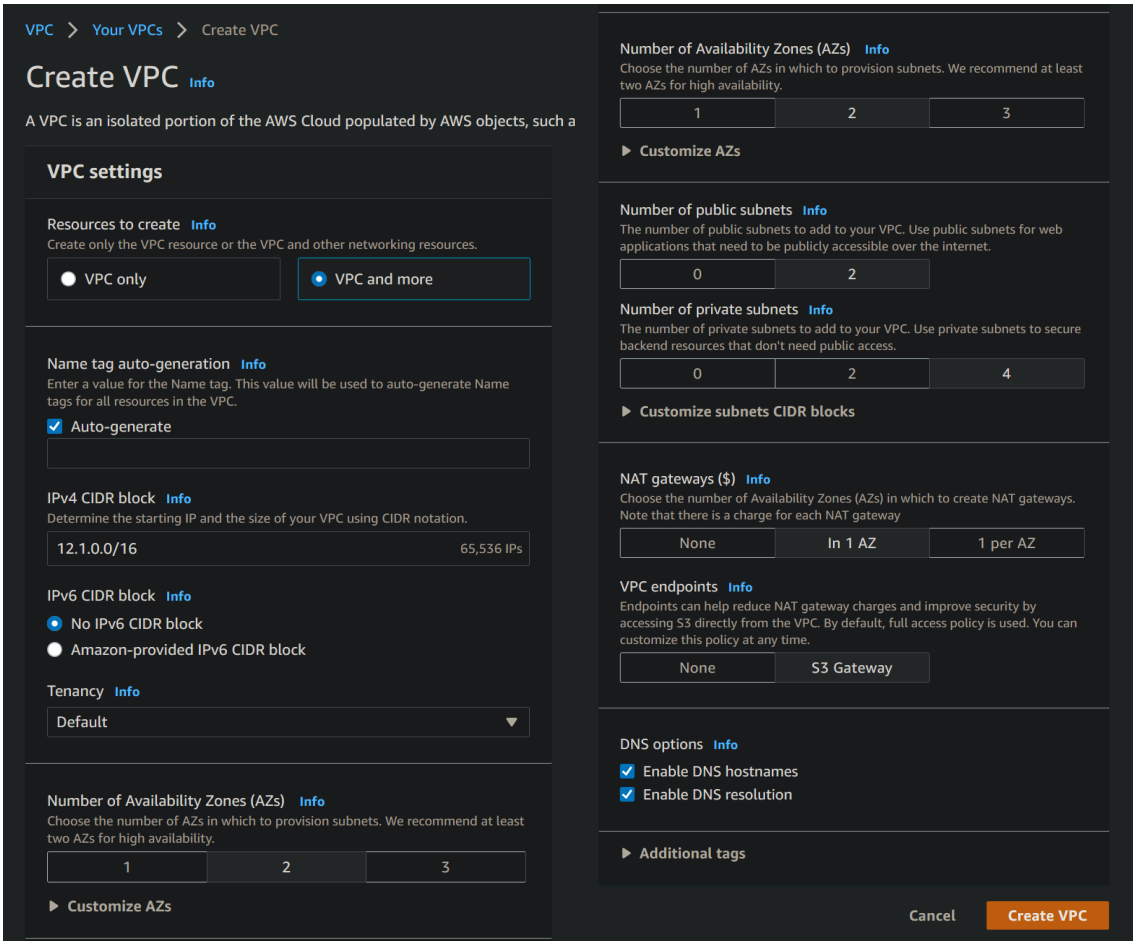
Recommendation 1: Web Application processing and on-scale demand

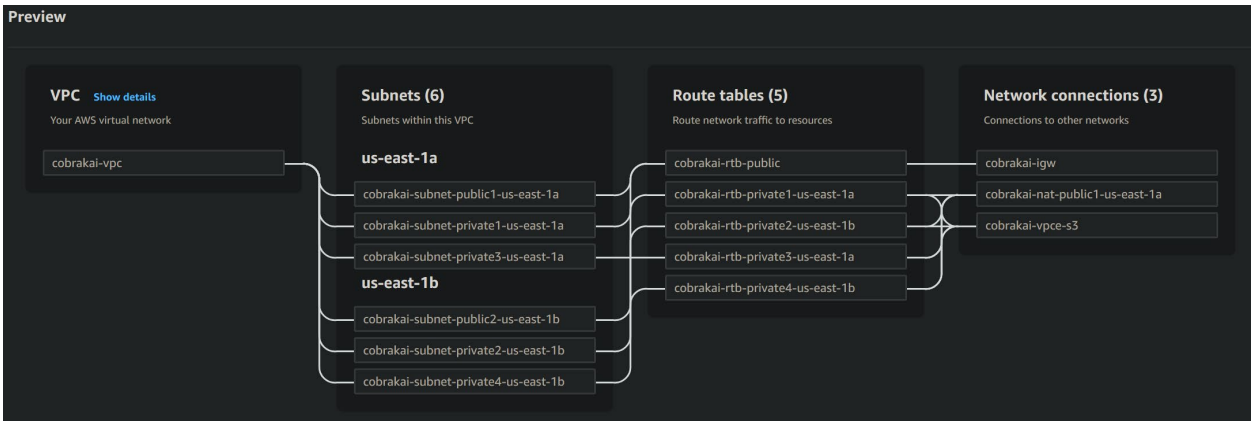
Setting up an VPC

To get started with, we create an Amazon VPC (Virtual Private Cloud) to configure and launch the services that are required to set up the Cobra Kai streaming platform. The VPC mimics the traditional datacenter setup that Cobra Kai initially had with better operational capabilities.

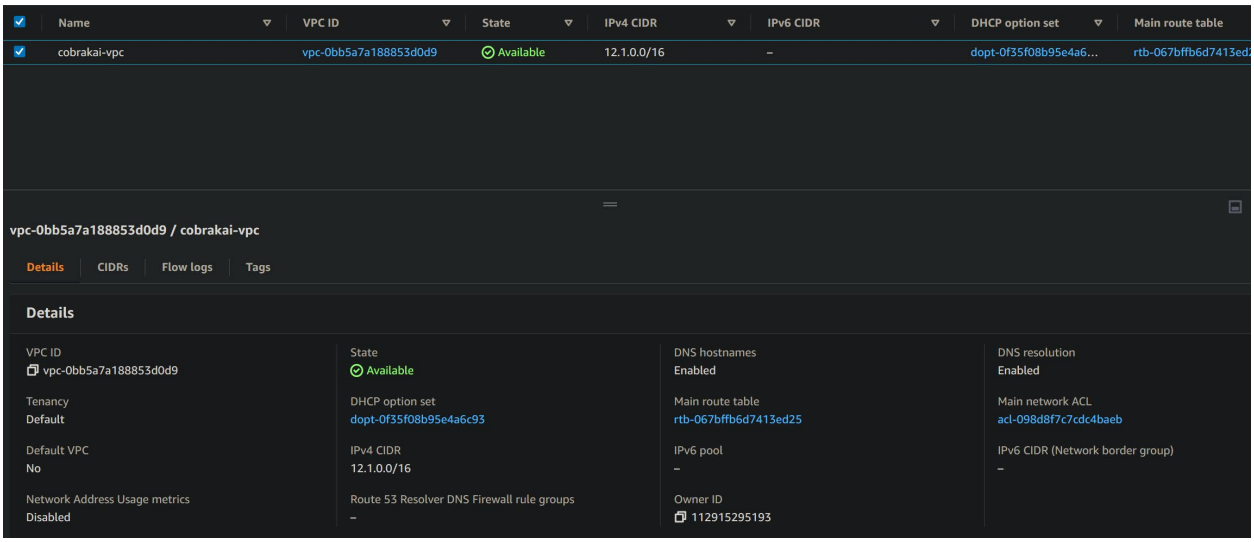
Open your AWS VPC Dashboard and click on Create VPC

1) Choose VPC and more, name the project as Cobra Kai
2) Set the IPv4 CIDR block to 129.2.0.0/16
3) Set IPv6 CIDR block to no
4) Availability zone 2
5) One NAT gateway
6) VPC endpoint to S3 gateway





The image above shows the network flow. The VPC contains two public subnets. One for each availability zone. And two private subnets in each zone. All the traffic from the subnets is routed via the route table.



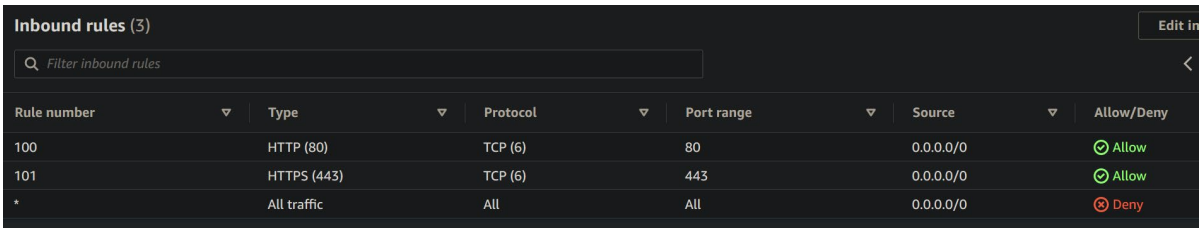
Since we choose VPC and more, amazon has done the work of creating a NAT, a Route table, Elastic IP and more. This reduces the work of the developers and reduces build time. Post creation of the VPC, it configures and sets up all the features from route table, NAT gateway, subnets.

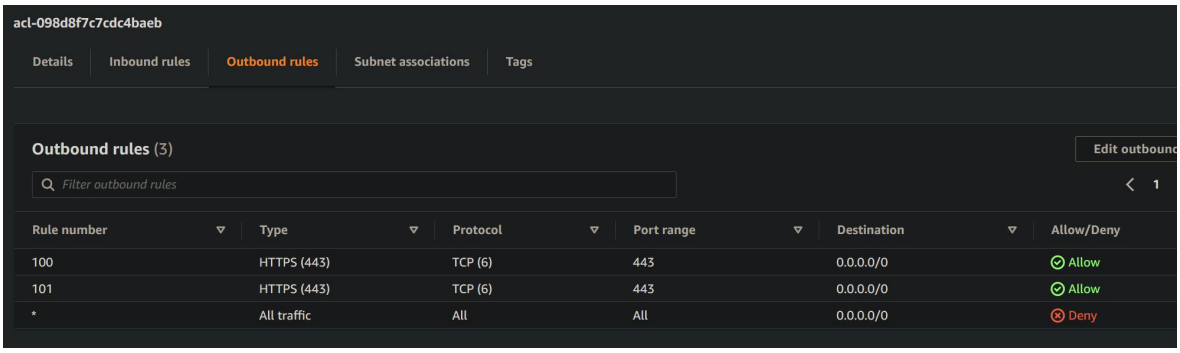
Post creation of the VPC, it is important to remember to configure a secure VPC. To do that we configure the traffic and the protocols.

First, Configure the network ACL ‘s

1) Choose Security - > Network ACL.

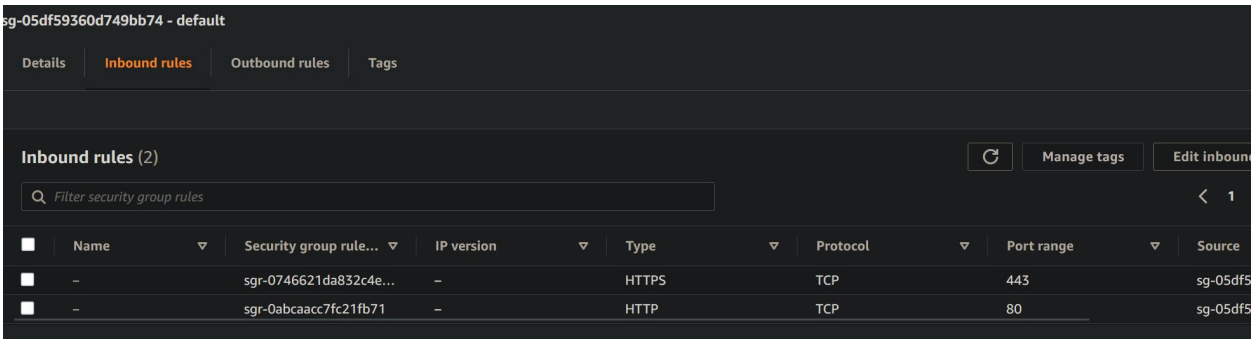
Edit the ACL to allow only HTTP / HTTPS traffic in both inbound and outbound rules.

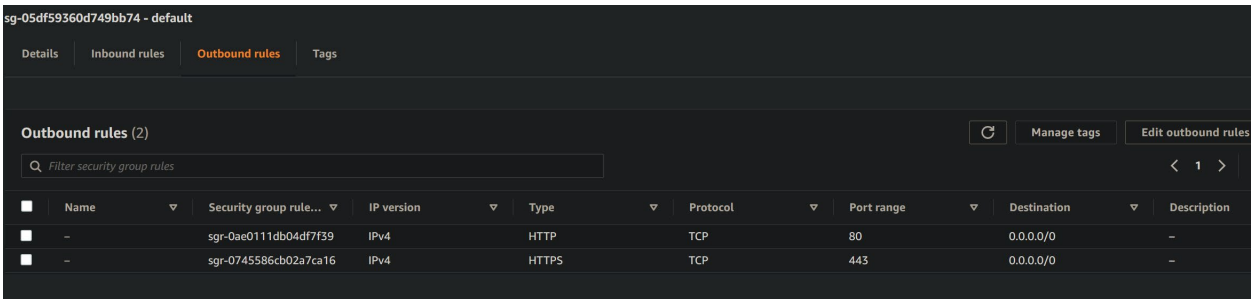




Secondly, we configure the security groups. The security group doubles as a virtual firewall for the EC2 instances by monitoring the inbound /outbound traffic. Configure the Security group to allow only HTTP/HTTPS traffic and we can also configure it to allow SSH traffic from the local IP address of the developers, if using AWS CLI.

1) Security -> Security Groups.
2) Configure the inbound and outbound rules for security group to allow only HTTP /HTTPS

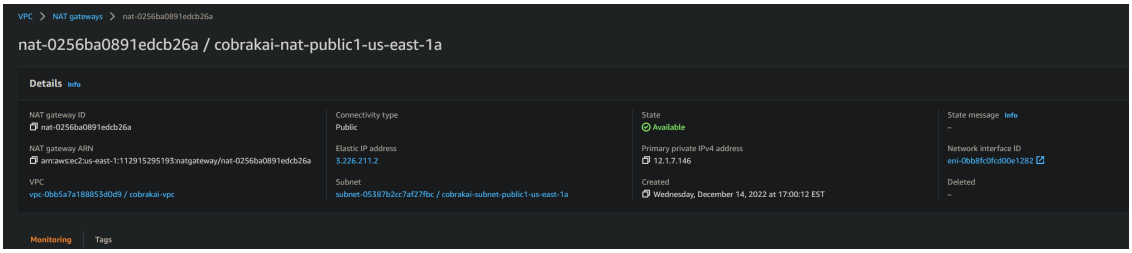




We need to create a NAT gateway to filter the traffic going to the private subnets.

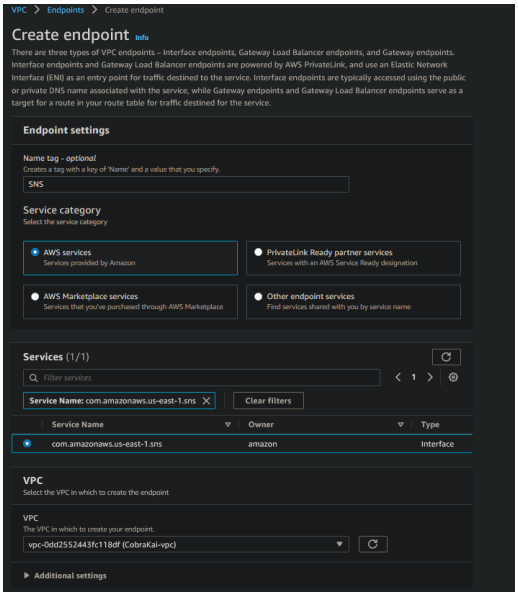
1) In the VPC Dashboard, Click on NAT gateway in the sidebar.
2) Click on create.

3) Select a subnet
4) Connectivity Type – private
5) Pick the Elastic IP that was allocated during the creation of the VPC
6) Create to launch the NAT gateway.



For the services to interact with the VPC, we need to create endpoints for each service.

1) Go to endpoints in the VPC Dashboard.
2) Create Endpoint (for sns)



3) Name the endpoint -> Select service category
4) For sns, we add the “com.amazonaws.us-east-1.sns”
5) Choose “Cobra-Kai” VPC and choose the subnet you want the service on.
6) Choose the default security group
7) Provide policies to endpoint access.
8) Create Endpoint.
9) Repeating the same process to add endpoints for the services below.

Endpoints (5) Info

Filter endpoints

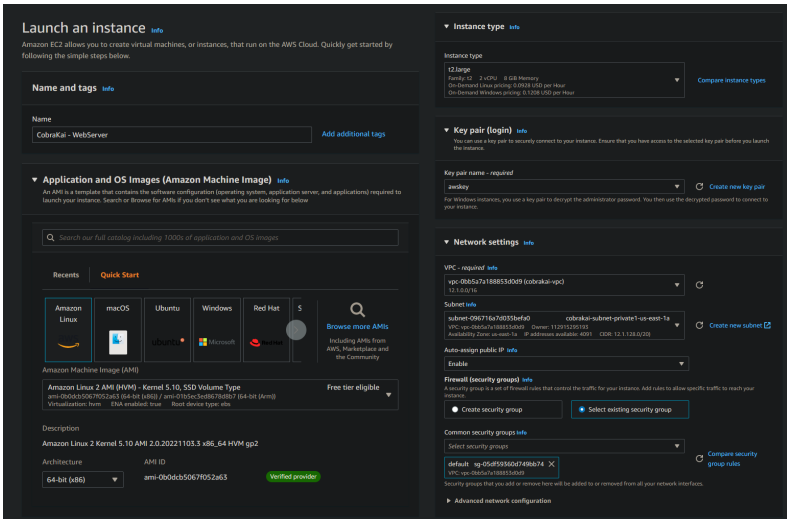
Name	VPC endpoint ID	VPC ID	Service name	Endpoint type
SNS	vpce-0b49d9e461ed959af	vpc-0dd2552443fc118df | CobraKai-vpc	com.amazonaws.us-east-1.sns	Interface
CobraKai-vpce-s3	vpce-0cd70e0e77bd31a06	vpc-0dd2552443fc118df | CobraKai-vpc	com.amazonaws.us-east-1.s3	Gateway
dynamodb	vpce-04429f21cca0ea375	vpc-0dd2552443fc118df | CobraKai-vpc	com.amazonaws.us-east-1.dynamodb	Gateway
systems-manager	vpce-0ce3d55c284ddb420	vpc-0dd2552443fc118df | CobraKai-vpc	com.amazonaws.us-east-1.secretsmanager	Interface
cloudwatch	vpce-06200f59c68951f1e	vpc-0dd2552443fc118df | CobraKai-vpc	com.amazonaws.us-east-1.cloudtrail	Interface

Creating an EC2

The Private subnets have a web server and an application server with elastic load balancing. We add a Web Server to the private subnet-1 and an Application Server to private subnet-3.

The steps below demonstrate on creating a web server and application server in availability zone 1. The same must be repeated in availability zone 2.

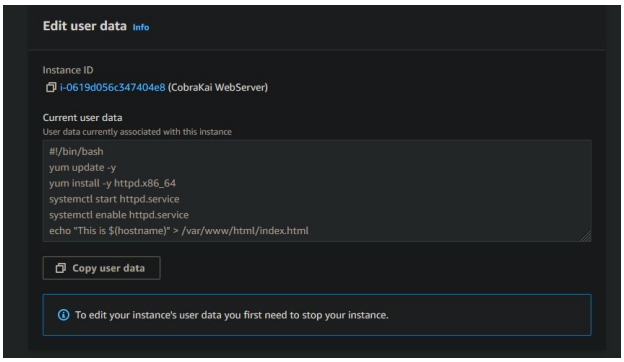
1) To create an ec2, EC2 dashboard and launch instance



Name the server. You have the option to choose any of the provided AMI's. (AMI are the virtual machines your server will run on) Here we process with Amazon Linux. Size as t2 since the webserver will have a significantly large amount of traffic. (Size can be configured post launching) Enable auto - assign IP.

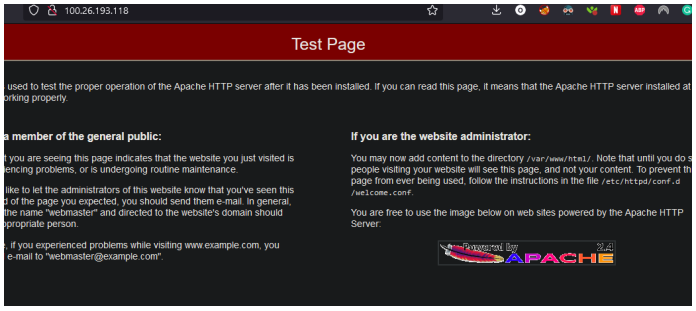
In VPC, choose the VPC that was previously created and edit the subnet to point to private subnet 1. Security group to use the existing security group that we have configured.

Under Advanced Details, add the following lines of code to the user data. This creates an Apache web server on your instance.



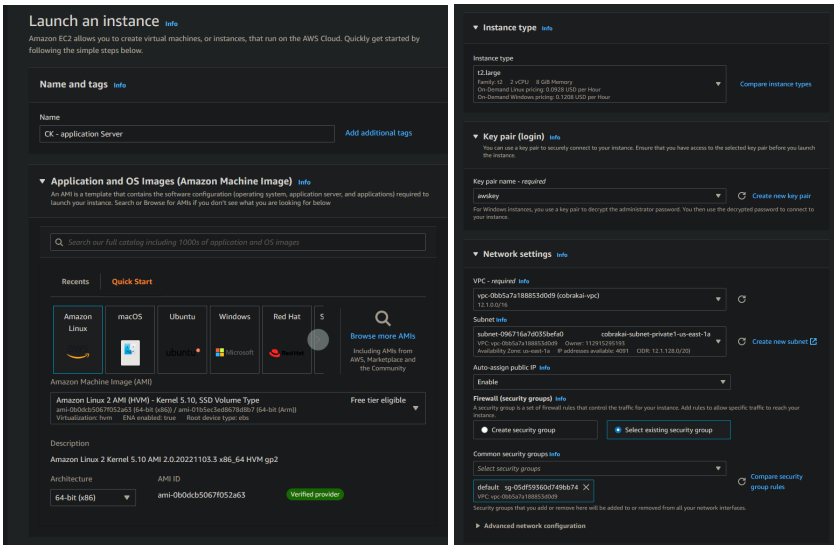
Click on Create. Wait for the EC2 to initialize. Post- initialization, click on the EC2 to view the public IPv4 address.

Load the IP on a web browser to test if the Apache server has been installed. The test page below indicates the successful installation of Apache web server.

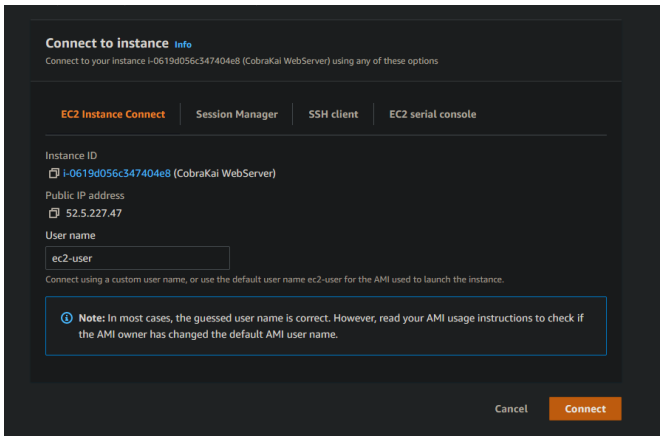


Create an application server.

Follow the same steps as the web server to create an application server.



Once the instance launches, connect to the instance using the connect button on the Dashboard.

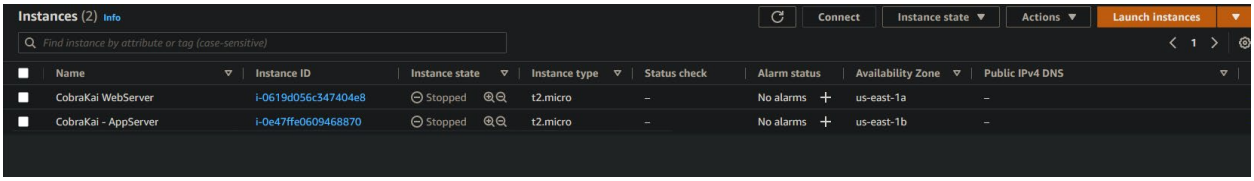


This opens the ec2 shell. Run the following commands to set up an application server on the EC2.

$ sudo su
$ yum install git
$ pip3 install Flask
$ git clone https://github.com/kts262/enpm809j
$ cd enpm809j
$ pip3 install gunicorn
$ python3 -m gunicorn -n 0.0.0.0:8000 enpm809j:app

The basic setup has been configured with the creation of two EC2 instances on the private subnet. The web server runs an Apache web server and the other with flask for the application server.

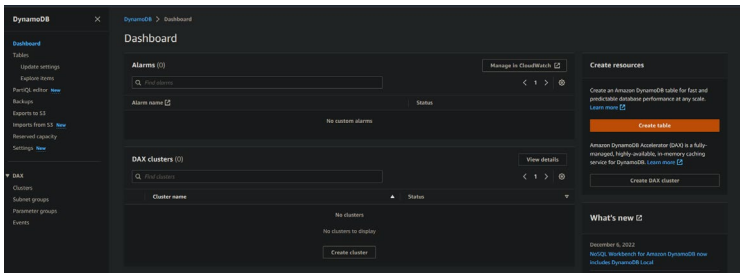
Repeat the same steps to set up the same in the availability zone 2

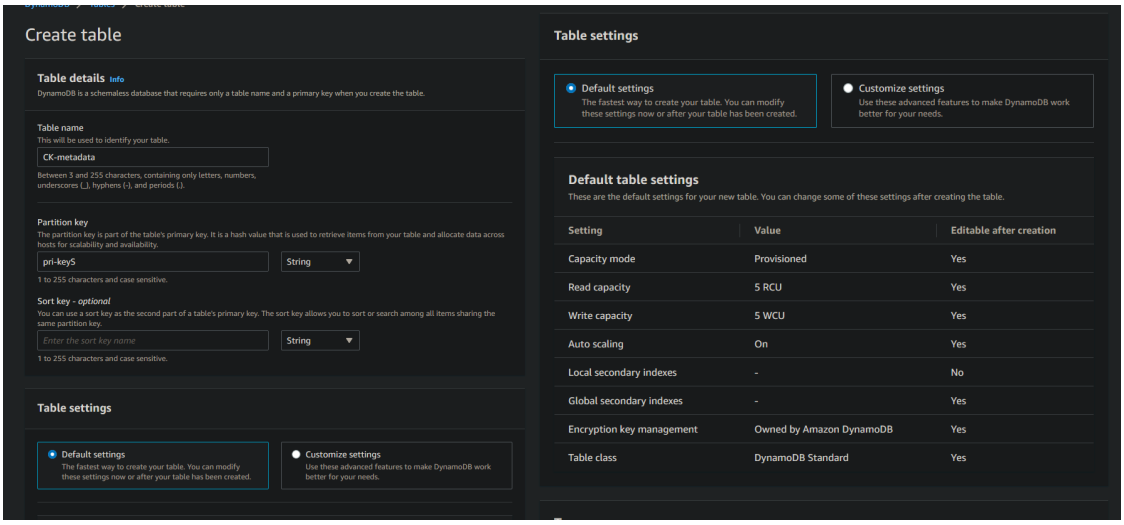


DynamoDB and S3 storage

To configure a database for storing backup's metadata and in-memory cache we use Amazon DynamoDB.

Amazon DynamoDB is a fast, NoSQL database service that is designed to run high speed flexible applications.





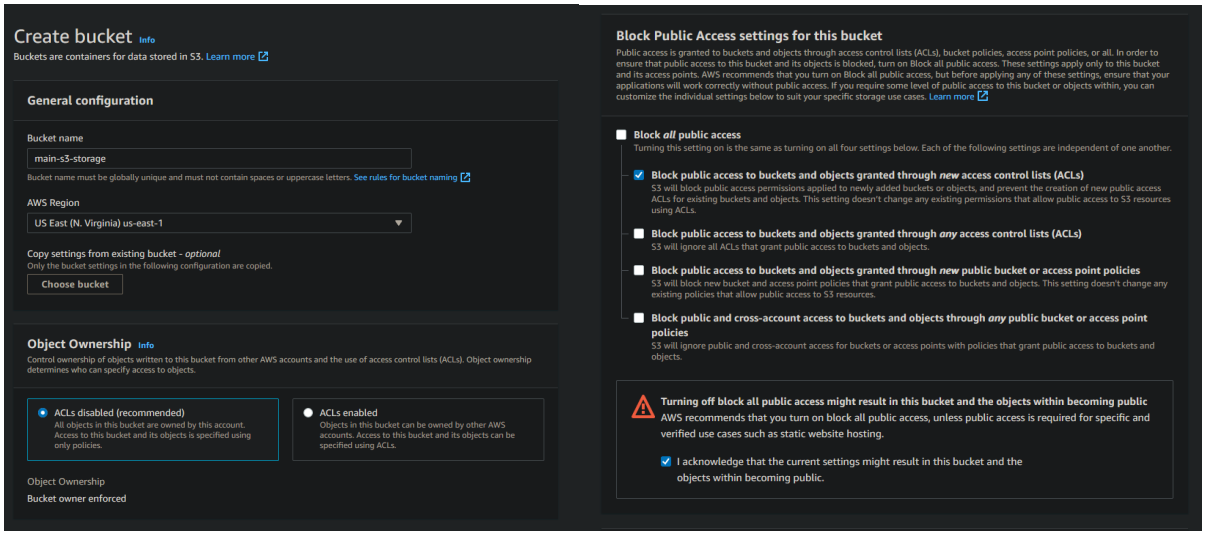
The database has been set up to one of the private subnets. The connections between the subnets are all balanced using a load balancer.

The S3 bucket is used of storing the video and act as the main storage for the application.

In the private subnet – 5, we add an S3 bucket.

To add an S3 bucket, go to S3 dashboard.

1) Create Bucket



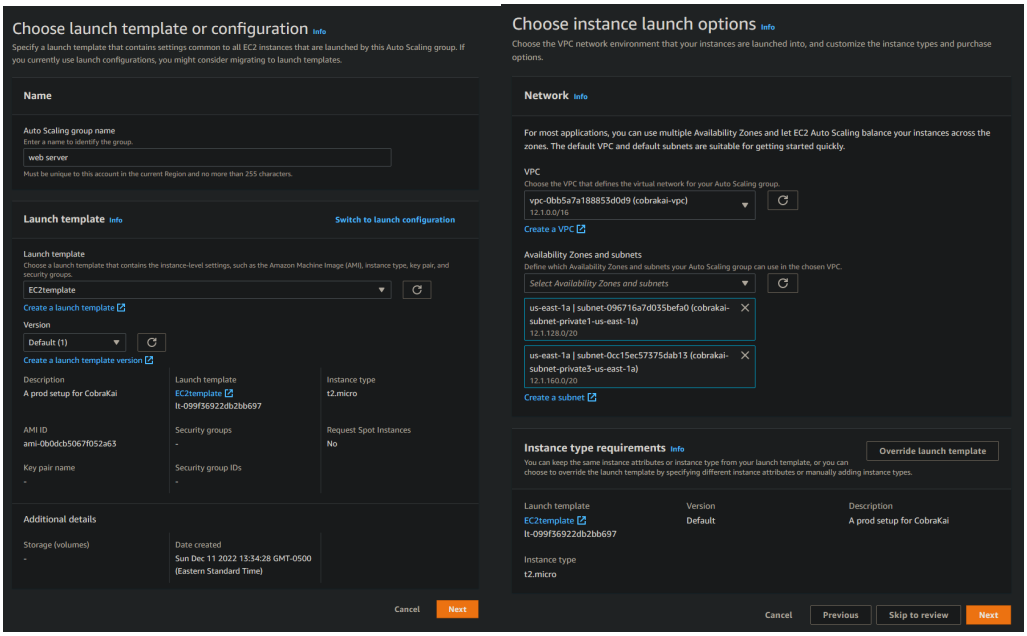
Auto-Scaling and Load Balancing

Now that we have setup our servers and the availability zones, we address the latency and auto-scaling issues Cobra Kai faced.

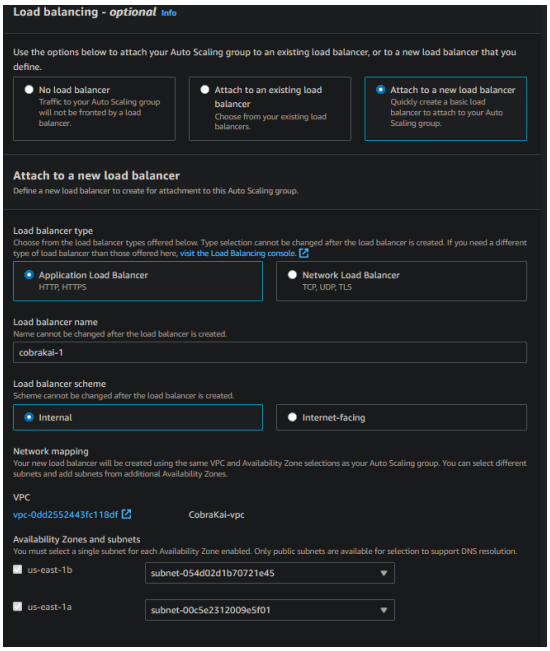
Setting up Auto-scaling with Elastic Load balancers will prevent the application from crashing when higher level of traffic come in. It will spin up new instances when on of the availability zone is dead or facing issue. The Auto-scaling group provides the option to create a Load balancer while setting up the group.

To setup an Auto-scaling group,

1) In the EC2 console go to Auto Scaling Group.
2) Name the group and select on Create a Launch Template.



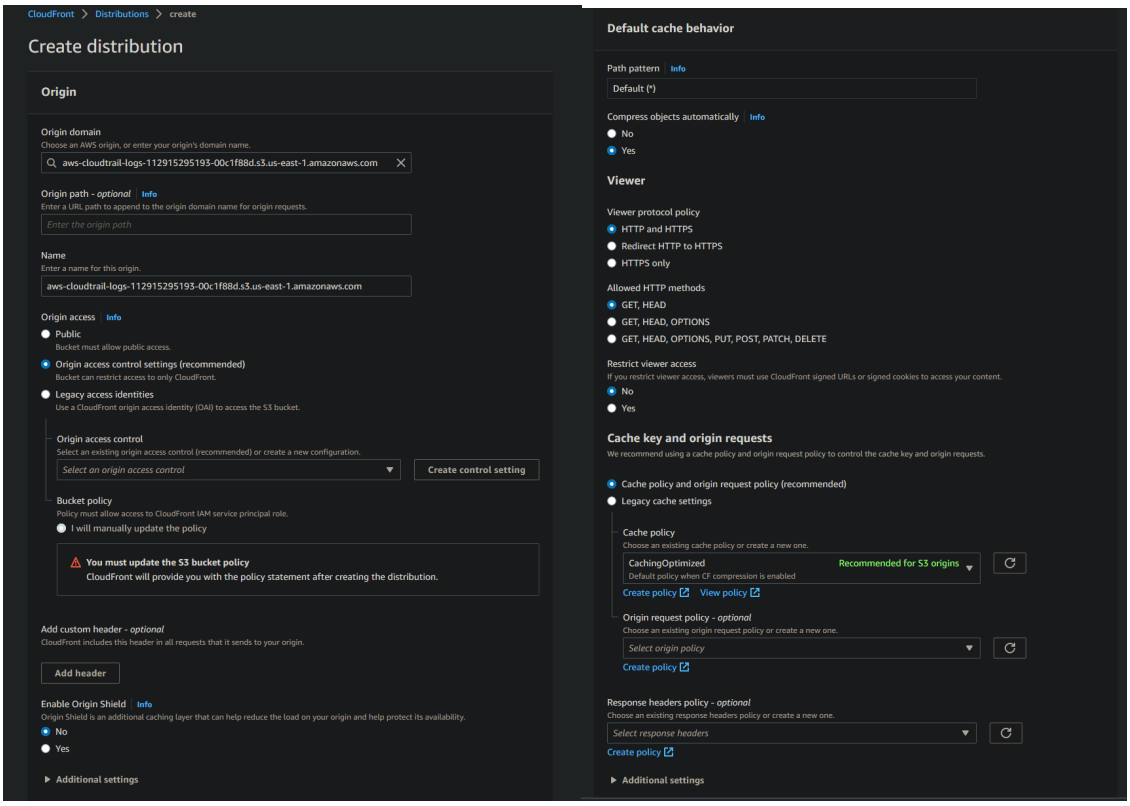
3) Pick the subnets the web servers are on in both the availability zones.
4) Select the new Load balancer and select the web server subnets in both zones.
5) Configure the groups and add policy's if you wish to.
6) Launch the group.
7) This initializes a Load balancer, a listener, and a target group.

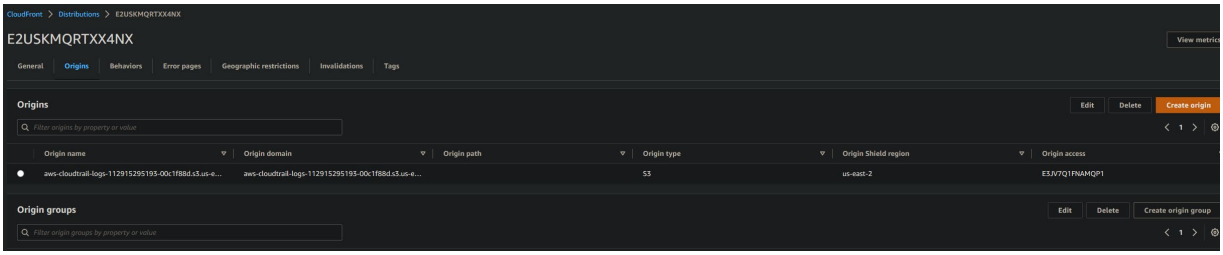


On-demand services

To meet CobraKai's on-demand needs we add a CloudFront CDN to provide live video streaming with improved latency. AWS CloudFront will require a S3 Bucket or a Load Balancer as an origin point to serve content from. We attach an S3 bucket that was created to store the media files to the CDN.

1) Go to the CloudFront Dashboard and click create distribution.
2) Choose the existing S3 bucket for origin
3) Set up the Origin Access Settings and create a control setting to prevent user's from accessing the URL. The S3 is to be used by the CDN only.
4) Set the protocols to HTTP GET and HEAD to prevent attacks.
5) Configure the ACL and launch the distribution.





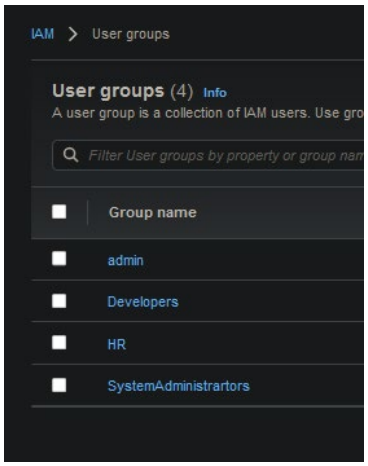
Recommendation 2: Identity Access Management and User Management

Identity Access Management

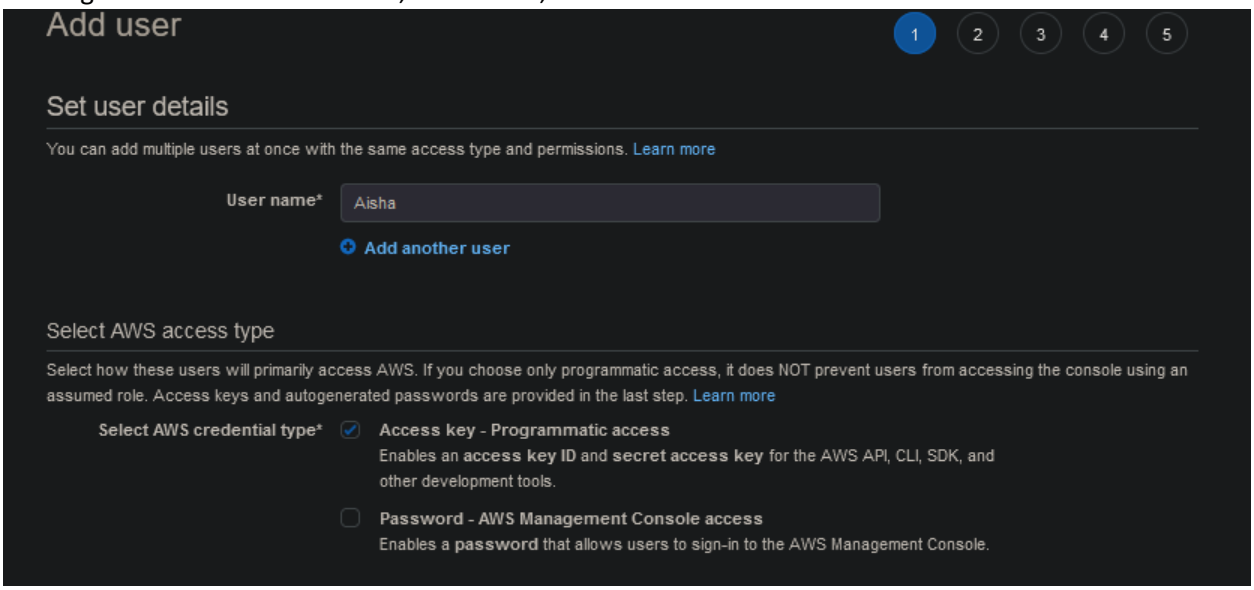
IAM was suggested during the proposal to securely control who has access to Cobra Kai's resources and who is authenticated (signed in) and authorized (has permissions). The granular permissions for Cobra Kai can be using the AWS IAM component to set different permissions to various people for various resources.

The steps below demonstrate on how to add users and configure the IAM for the employees of Cobra Kai. For e.g., purposes, we add Johnny Lawrence and Miguel Diaz to a user group named HR and give them restricted admin access.

1) Go to IAM dashboard.
2) Create Groups, defining the policies for each group. For. E.g., System Administrators have system admin policies. Repeat the same for Developer, HR, and admin.



3) To add users, Click Users -> Add User. Aisha is the CISO hence she gains programmatic access which gives access to all consoles, dashboard, CLI.



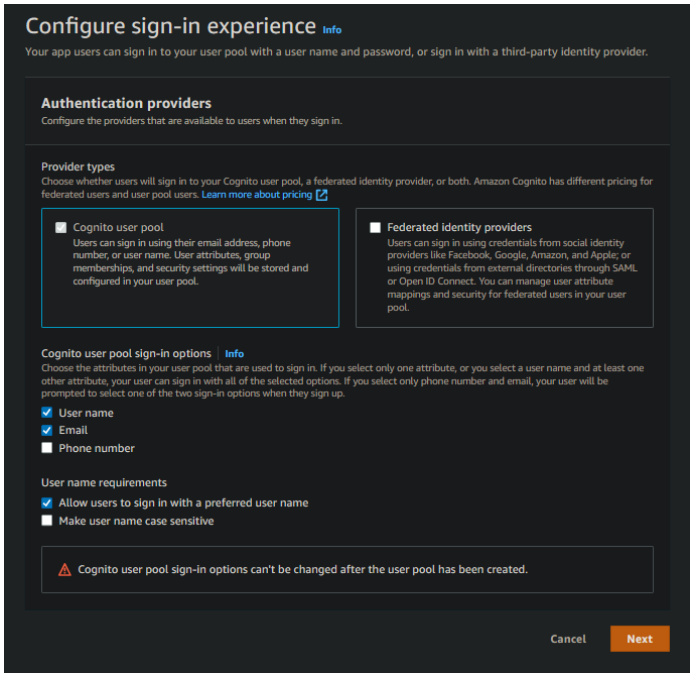
4) Repeat the same for the members
5) Developer Access – Eli
6) System administrator – Bert
7) Managerial access with read-only permissions – COO, CEO, Miguel
8) If more roles are to be delegated, it can be done using the IAM console.

The IAM has many features, we can strengthen the password policy for IAM, by going to Settings -> Password Policy and configuring it.

User Management and Policies

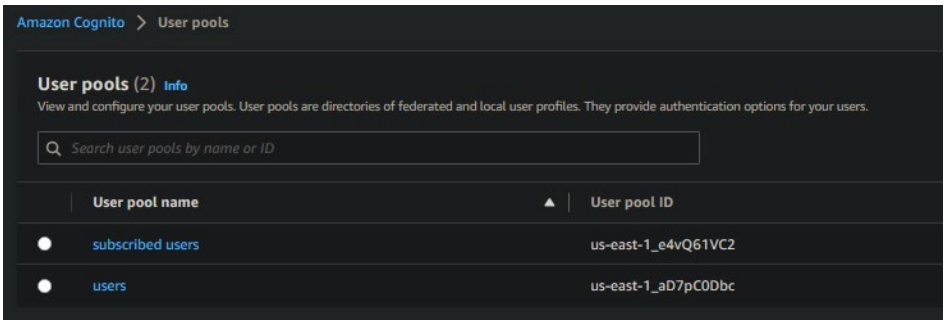
Amazon Cognito is a feature that helps to seamlessly integrate user authentication and authorization. It provides the IAM features for user's and helps secure the application.

1) Go to the Cognito dashboard.
2) Create User pools, The federation identity providers allow users to sign-up using third party such as google, apple, amazon.



3) Create a password policy
4) Set up MFA

5) Provide a User account Recovery Method
6) Configure the messaging delivery method for verification - Send email with Cognito
7) Associate the user pool with an IAM role
8) Additionally, you can add the IP address you want to block and add.
9) Create the user pool



Additionally, AWS Secrets manager can be used to store user sensitive details.

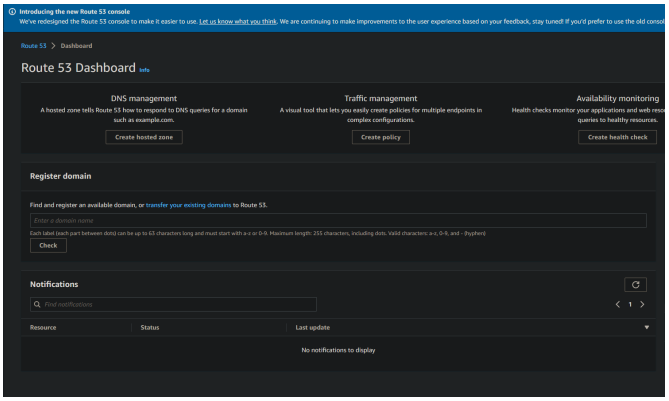
Recommendation 3: Security against DDoS attacks and Hardware failure

Cobra Kai had expressed their concerns over hardware failures. The advantage of moving to the cloud is that it doesn't require large amount of hardware to function. This in turn saves CobraKai the money in-terms of large storage devices and protecting the storage from any damages.

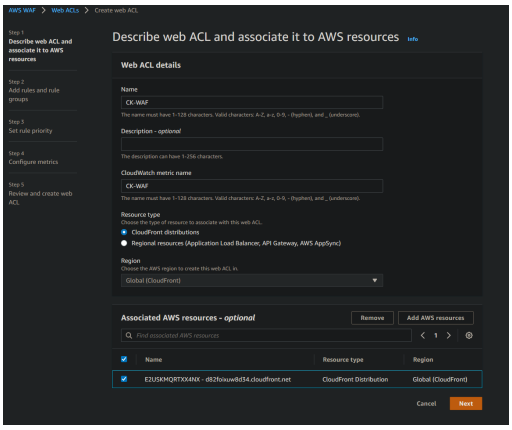
Another main concern of Cobra Kai was the attacks from its enemy Daniel Russo. Amazon by default uses AWF Shield for all applications that run on AWS. AWS Shield is a managed DDoS protection service that safeguards applications running on AWS. AWS Shield when combined with AWS WAF increases the protection against DDoS.

Amazon Route 53 is a DNS system that manages the domain naming system and provides DDoS protection.

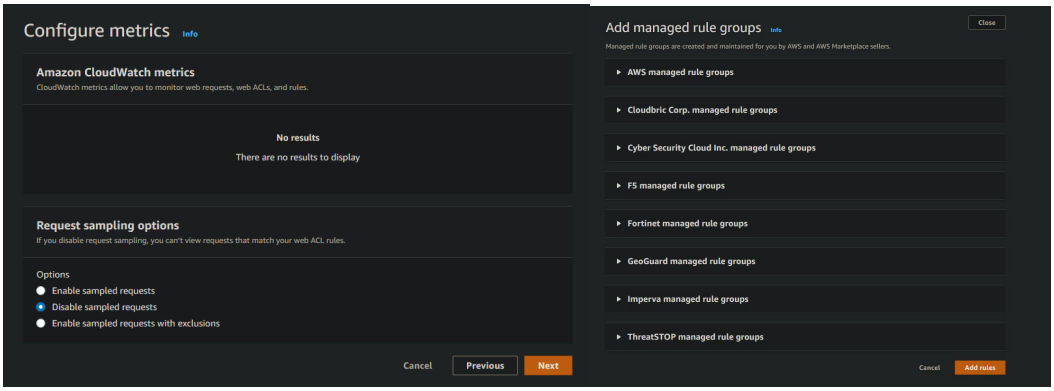
To set up the Amazon Route 53, go to the Route 53 dashboard. It provides the option to use an existing domain name or create a new one. It also allows to add policies and cloud monitoring to monitor the traffic that is to the application. The traffic is then forwarded to the AWS WAF.



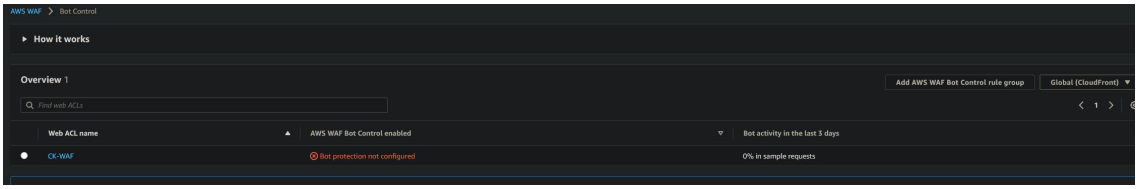
To configure an AWS WAF. Go to the AWS WAF Dashboard -> Add ACL.



The WAF is connected to the CDN. You can add rules to your WAF – ACL. AWS does provide the option to either customize your own or use amazon managed policies. Amazon offers policies that are secure and managed by third parties such as Fortinet. It is recommended to go with Amazon managed policies since it is configured to prevent attacks and bot traffic.
Enable cloud watch metrics to monitor the inbound/outbound traffic.



Another interesting option that WAF provides is that AWS also has a bot control setting to track the number of bot requests.



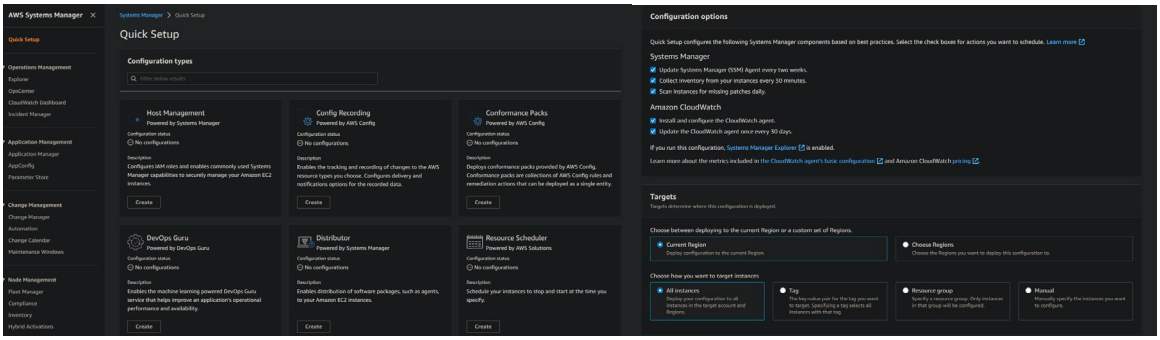
Route 53 along with AWS Shield forwards the malicious traffic to a random CDN and hiding the web application from malicious actors. To pass through the Amazon 53, the user needs to access the web application using set DNS names for e.g., www.cobrakai.com & media.cobrakai.com. anything other than these two would be terminated or rerouted to a random CDN.

Recommendation 4: Patching Strategy & Secure System Administration

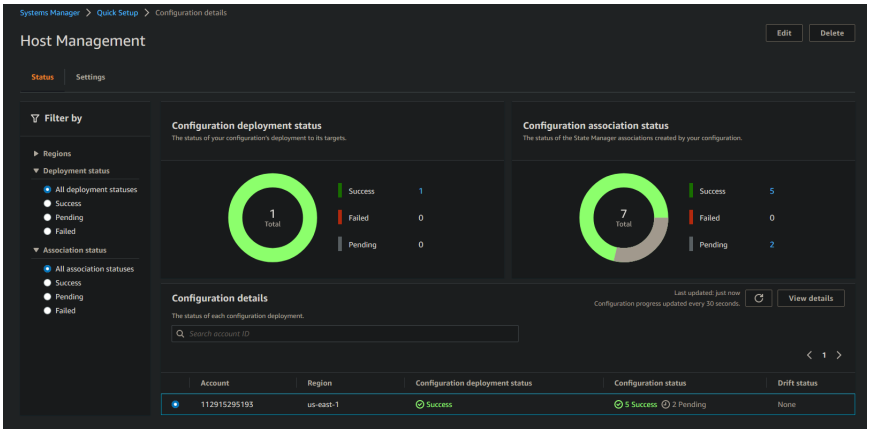
The AWS System Manager is one-in-all console that provides secure end-to-end management solution targeting both the Patching and Secure Administration issues that Cobra Kai had.

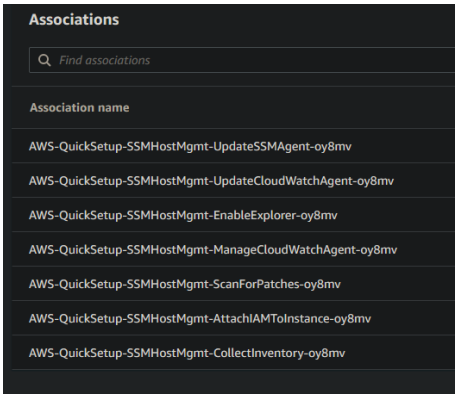
The Dashboard offers multiple configurations. We picked Host since it allows to monitor and configure all EC2 instances and monitor user details. You can set up multiple configurations based on the need. For demonstration purposes we proceed with the Host Management.

1) To add Systems Manager, go to the System Manager Dashboard and choose the configuration you prefer.
2) Enable Cloud Watch and pick the region and the instances you want to monitor.



This show that the Systems Manager is currently running on the VPC with the EC2 instances that we setup.

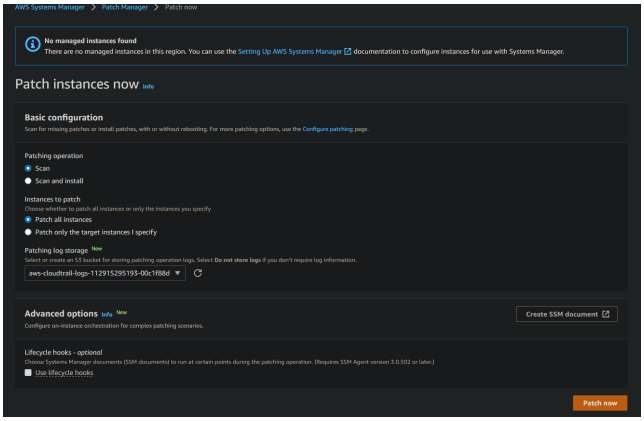




AWS System Manager offers a Patch Manager which can deploy patches simultaneously to applications and nodes across your organization. Additionally, you can monitor patch compliance account by account.

To setup a patch manager,

1) Choose Patch Manager from the Systems Manager Sidebar.
2) Create the default patch configuration
3) Choose the instances that you want to patch, specify the patch log storage.
4) Click on Patch Now.



The Patch Manager provides functionalities that help automate routine tasks such as automating the system scanning for vulnerabilities, updating patches, maintain logs of updates. Bert could schedule and configure the system manager to administer the system securely.

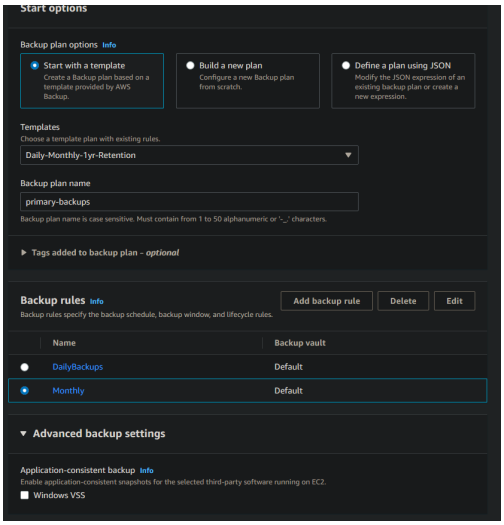
Recommendation 5: Backup Strategy

To improve Cobra Kai's backup strategy, we use AWS Backup with DynamoDB to build an automated secure backup.

AWS Backup helps automate the process of backing up the system on a regular basis. Bert has configuration controls. He could integrate the Backup Routines with the system manager to back-up data regularly. The Backup Drive can be accessed only by those with permissions thus maintain its integrity.

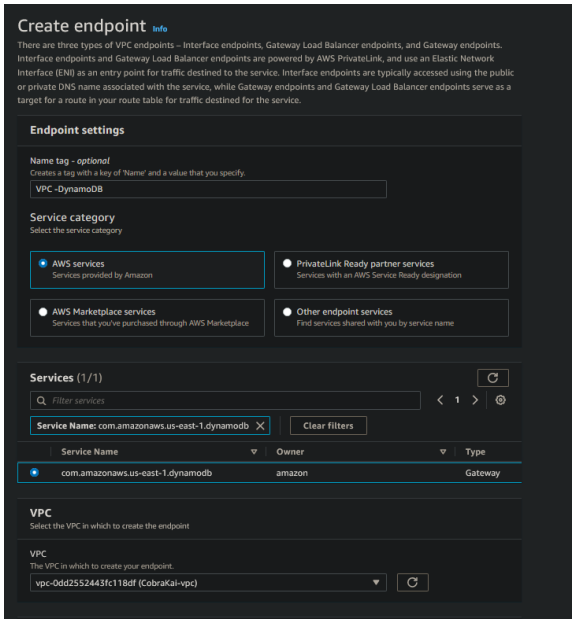
In AWS Backup Dashboard, click on Create Plan.

It provides you with the option to configure your own backup strategy or use one of the AWS Backup templates.

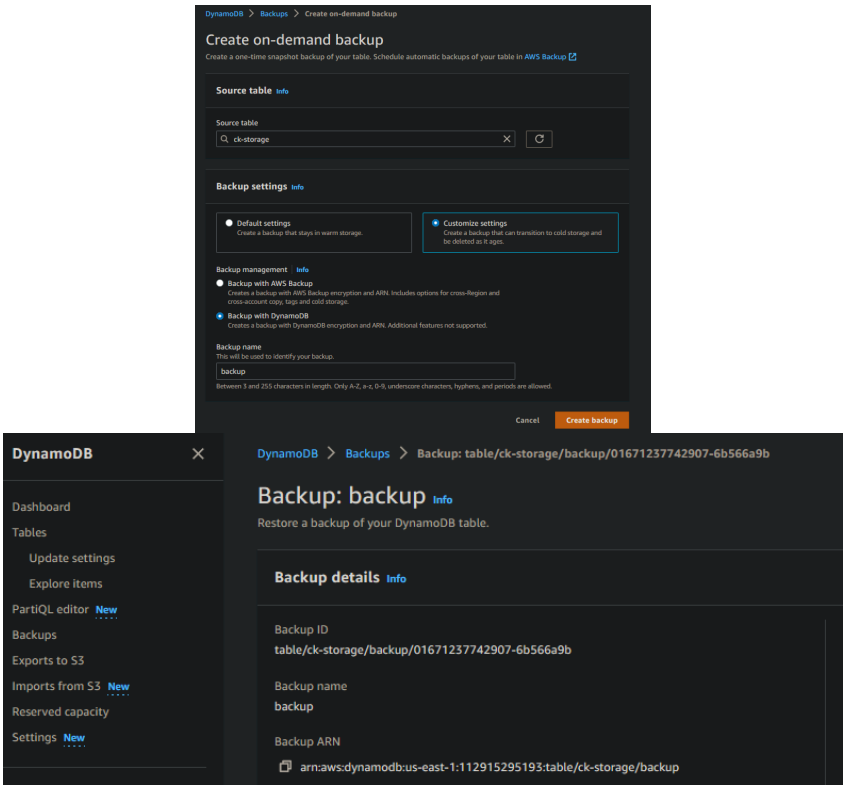


We can configure the backup to store the logs and the backup in the Amazon DynamoDB that we created in the basic setup. The data in the DynamoDB is encrypted and can be access only by those whose role have been configured in the IAM.

Create an endpoint in the VPC for the DynamoDB to connect with the AWS backup.



Go to your DynamoDB console to create an on-demand backup. The backup is created on the DynamoDB with the option to export to your S3 bucket.



Recommendation 6: PCI DSS and Compliance.

All the services that are proposed in the document should meet the PCI DSS Compliance requirements since Cobra Kai is processing user credit card details. It is to be noted that all Amazon services used in the application document are certified level 1 PCI DSS compliant.

AWS Firewall and AWS Shield is configured to monitor all the inbound and outbound traffic. The DNS 53 along with the WAF drops any suspicious packets. Further, the Firewall ACL is configured to allow only HTTP/HTTPS traffic.

The VPC offers isolated instances thus compartmentalizing the data. Each subnet in the VPC is protected and monitored to prevent any data leak. The security groups and NAT gateways are all configured to deny any suspicious traffic coming through.

The IAM console helps maintain the data integrity by granularizing the permission to the employees. This controls who has access to what. This makes sure the user sensitive information like credit card is not leaked. To harden the security, MFA has been setup for IAM and the Amazon Cognito, which denies access to anyone who isn't a user or an employee. Further, password policies have been set.

The AWS DynamoDB and S3 are used for storing cache and user details, both are encrypted and can be accessed only those who have the permissions.

The AWS secrets manager stores the user credit card information and sensitive data in a secure encrypted way.

AWS CloudWatch and CloudTrail log every user activity in the application. Any unusual activity will start an alarm and can be reversed back to the source. Additionally, tools have been provided to watch the metrics and monitor suspicious logs.

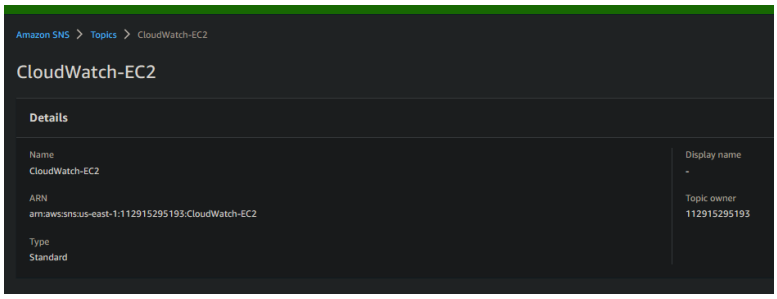
Recommendation 7: Logs.

For monitoring the infrastructure and to maintain the logs, we are integrating CloudWatch and CloudTrail to the VPC. CloudWatch maintains the log of all the actions that occurs in the VPC, thus allowing to monitor and review every single feature. We have enabled CloudWatch while configuring the features that have been used in the recommendations.

Amazon SNS is a service that allows to notify specified number of user when an alarm is triggered.

We create a topic for the EC-2utilization.

1) Go to Amazon SNS -> Create Topic
2) Name the topic, provide the encryption, and access control policies if required.
3) Create



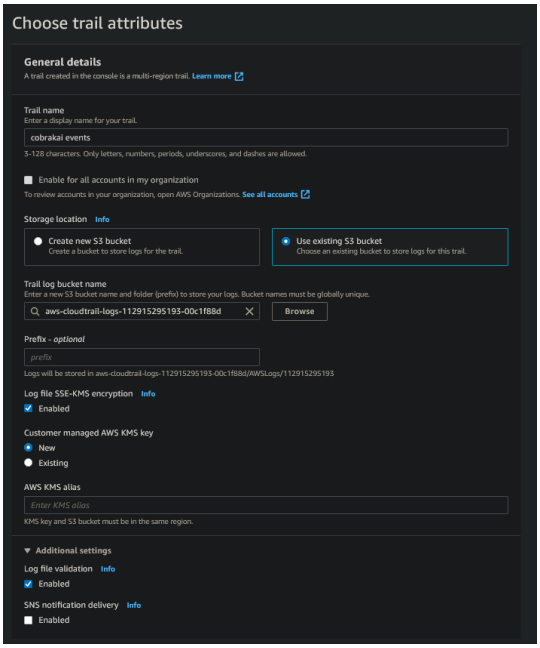
To set up CloudWatch Alarm for monitoring the CPU utilization of the EC2 instances,

1) In CloudWatch Dashboard, All Alarms - > Create Alarm.
2) Select Metric - > EC2 -> per instance metrics -> CPU Utilization
3) Choose Alarm -> existing CloudWatch Alarm
4) Name the alarm.
5) Create Alarm.

CloudTrail is generally used to maintain a track of user actions. It maintains a log events to meet the governance and audit needs for AWS accounts.

Create trail to monitor user actions and store the logs in the s3 bucket.

1) Go to CloudTrail Dashboard
2) Create Trail
3) Configure trail attributes, name the trail, select the existing s3 bucket to store the logs
4) Configure CloudWatch Logs
5) Create Trail



References

https://docs.aws.amazon.com/waf/latest/developerguide/setting-up-waf.html#setting-up-waf-iam

https://docs.aws.amazon.com/index.html

https://medium.com/@jamesaaronbanks/creating-an-ec2-instance-with-an-apache-web-server-38e6deda030d

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/WSL.html

https://aws.amazon.com/compliance/pci-dss-level-1-faqs/