

ENPM665 – Final

Om Venugopal

[omvg@umd.edu](mailto:omvg@umd.edu)

118116655

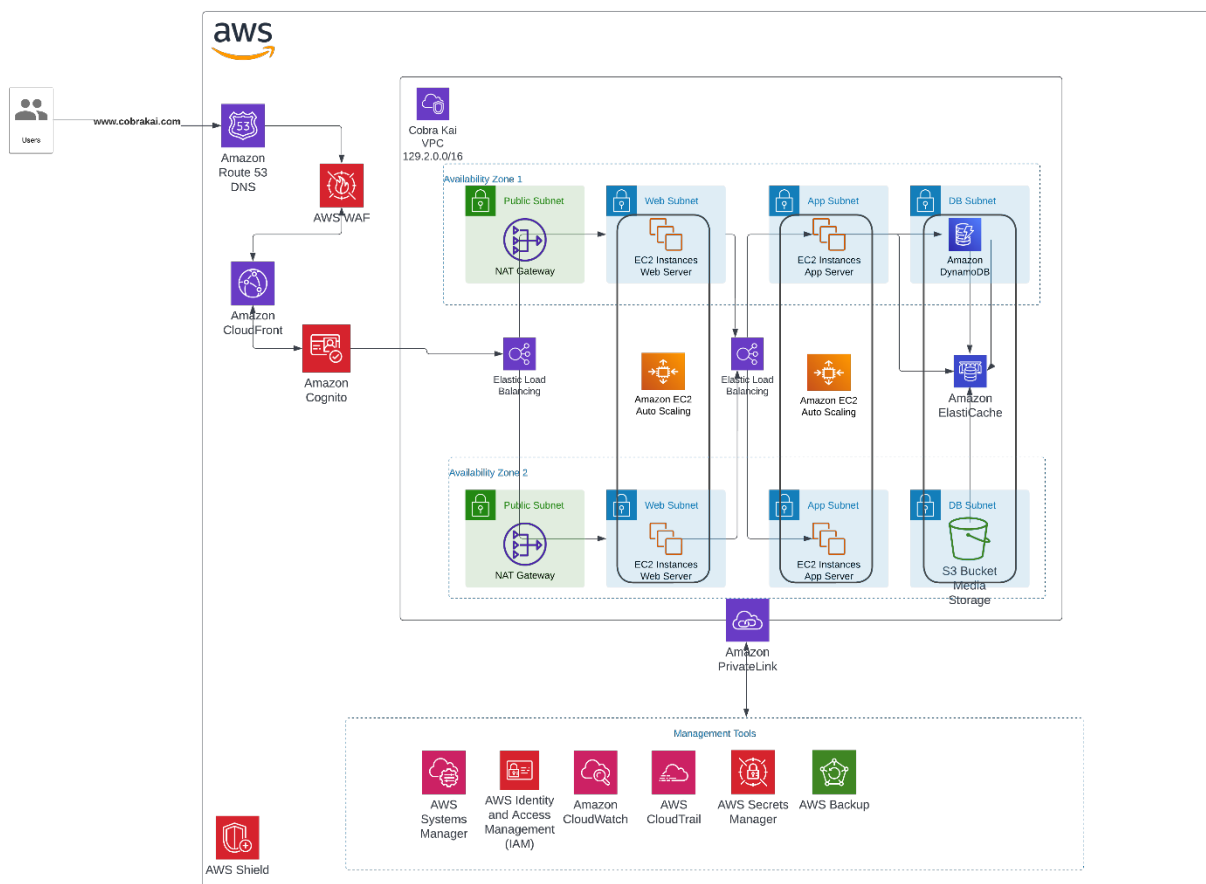
## Contents

Introduction .....	3
Recommendation 1: Web Application processing and on-scale demand .....	4
Setting up an VPC.....	4
Creating an EC2 .....	8
DynamoDB and S3 storage .....	10
Auto-Scaling and Load Balancing .....	12
On-demand services .....	13
Recommendation 2: Identity Access Management and User Management .....	14
Identity Access Management.....	14
User Management and Policies .....	15
Recommendation 3: Security against DDoS attacks and Hardware failure .....	17
Recommendation 4: Patching Strategy & Secure System Administration .....	19
Recommendation 5: Backup Strategy.....	20
Recommendation 6: PCI DSS and Compliance.....	22
Recommendation 7: Logs.....	23
References .....	24

## 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 S3 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

VPC > Your VPCs > Create VPC

## Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as

### VPC settings

**Resources to create** [Info](#)  
Create only the VPC resource or the VPC and other networking resources.

☐ VPC only ☒ VPC and more

**Name tag auto-generation** [Info](#)  
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

☒ Auto-generate

**IPv4 CIDR block** [Info](#)  
Determine the starting IP and the size of your VPC using CIDR notation.

12.1.0.0/16 65,536 IPs

**IPv6 CIDR block** [Info](#)

☒ No IPv6 CIDR block ☐ Amazon-provided IPv6 CIDR block

**Tenancy** [Info](#)

Default

**Number of Availability Zones (AZs)** [Info](#)  
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 2 3

► Customize AZs

**Number of public subnets** [Info](#)  
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 2

**Number of private subnets** [Info](#)  
The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 2 4

► Customize subnets CIDR blocks

**NAT gateways (\$)** [Info](#)  
Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway.

None In 1 AZ 1 per AZ

**VPC endpoints** [Info](#)  
Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

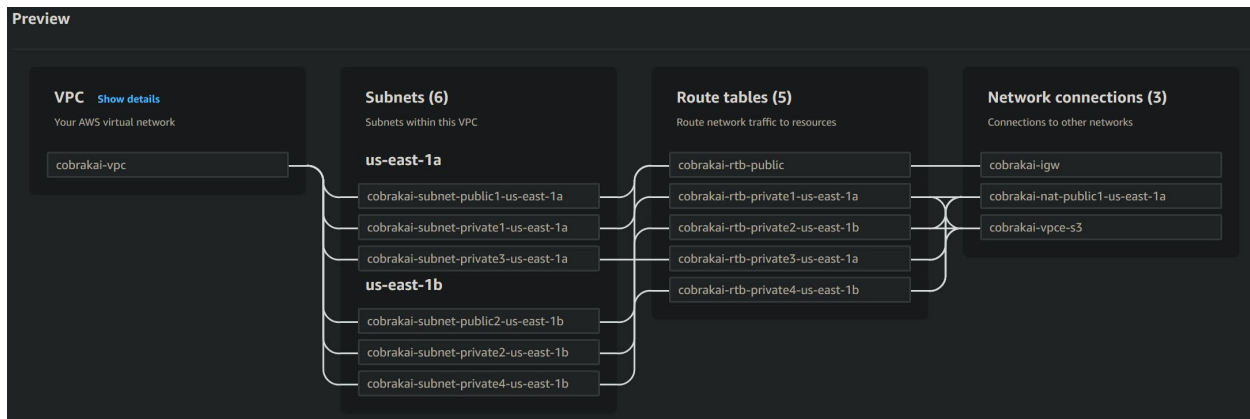
None S3 Gateway

**DNS options** [Info](#)

☒ Enable DNS hostnames ☒ Enable DNS resolution

► Additional tags

Cancel Create VPC



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.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table
cobrakai-vpc	vpc-0bb5a7a188853d0d9	Available	12.1.0.0/16	-	dopt-0f35f08b95e4a6...	rtb-067bffb6d7413ed25

vpc-0bb5a7a188853d0d9 / cobrakai-vpc			
Details	CIDRs	Flow logs	Tags
<b>Details</b>			
VPC ID vpc-0bb5a7a188853d0d9 Tenancy Default Default VPC No Network Address Usage metrics Disabled	State Available DHCP option set dopt-0f35f08b95e4a6c93 IPv4 CIDR 12.1.0.0/16 Route 53 Resolver DNS Firewall rule groups -	DNS hostnames Enabled Main route table rtb-067bffb6d7413ed25 IPv6 pool - Owner ID 112915295193	DNS resolution Enabled Main network ACL acl-098d8f7c7cdc4baeb IPv6 CIDR (Network border group) -

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.

**Inbound rules (3)** Edit inbound rules

Filter inbound rules

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
101	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

**acl-098d8f7c7cdc4baeb**

Details Inbound rules **Outbound rules** Subnet associations Tags

**Outbound rules (3)** Edit outbound rules

Filter outbound rules

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow
101	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

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

**sg-05df59360d749bb74 - default**

Details **Inbound rules** Outbound rules Tags

**Inbound rules (2)** Refresh Manage tags Edit inbound rules

Filter security group rules

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
-	sgr-0746621da832c4e...	-	HTTPS	TCP	443	sg-05df5
-	sgr-0abcaacc7fc21fb71	-	HTTP	TCP	80	sg-05df5

**sg-05df59360d749bb74 - default**

Details Inbound rules **Outbound rules** Tags

**Outbound rules (2)** Refresh Manage tags Edit outbound rules

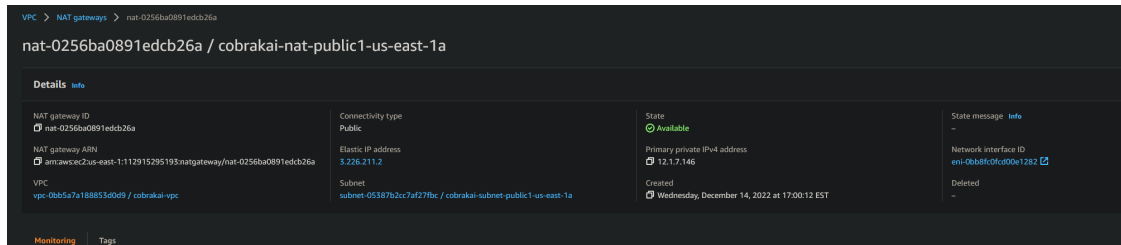
Filter security group rules

Name	Security group rule...	IP version	Type	Protocol	Port range	Destination	Description
-	sgr-0ae0111db04df7f39	IPv4	HTTP	TCP	80	0.0.0.0/0	-
-	sgr-0745586cb02a7ca16	IPv4	HTTPS	TCP	443	0.0.0.0/0	-

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) <a href="#">Info</a>						
	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.

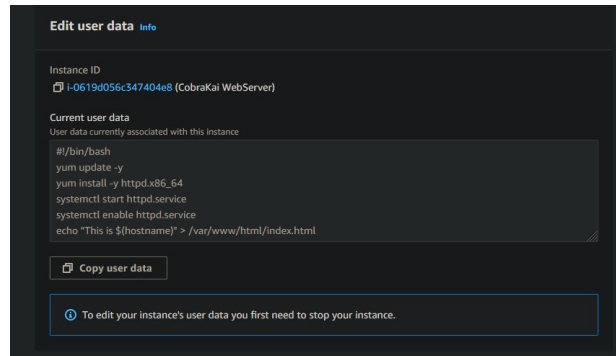
- 1) To create an ec2, EC2 dashboard and launch instance

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.

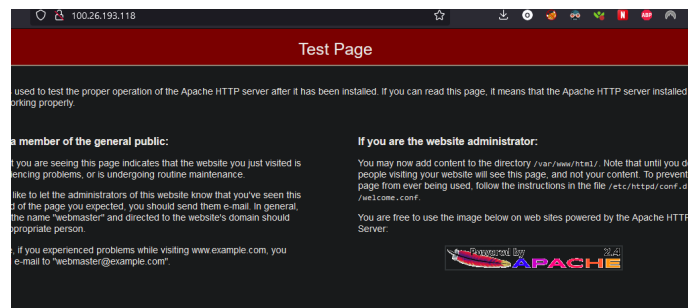
Under Advanced Details, add the following lines of code to the user data. This creates an Apache web server on your instance.





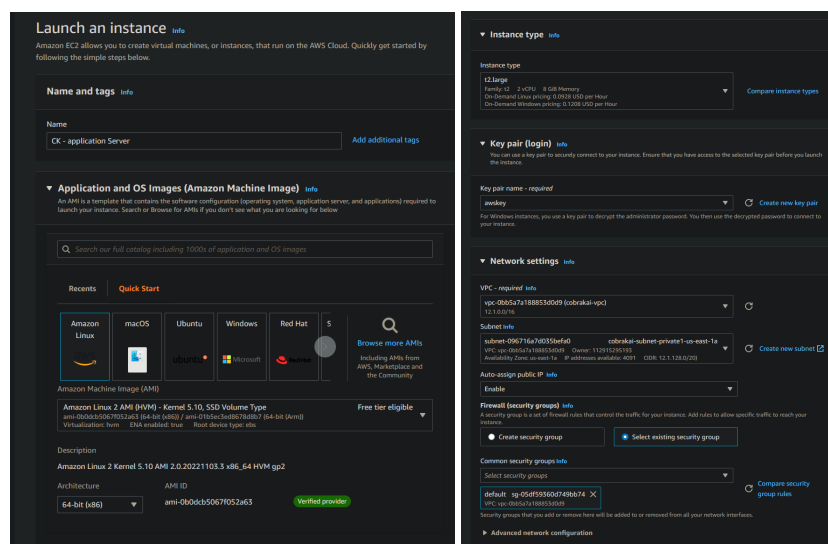
Click on Create. Wait for the EC2 to initialize. Post- initialization, click on the EC2 to view the public IPv4 address.

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.

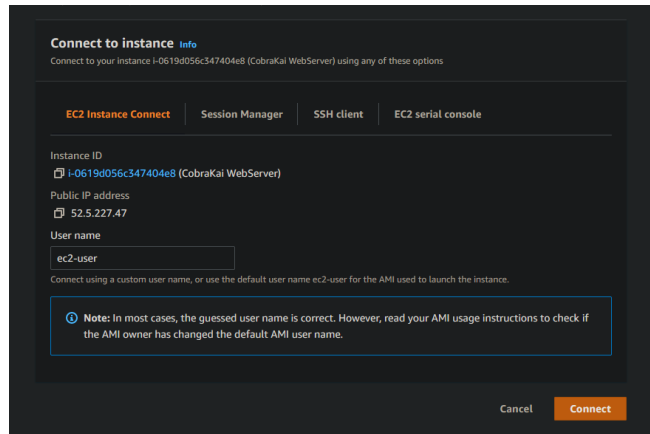


Create an application server.

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.

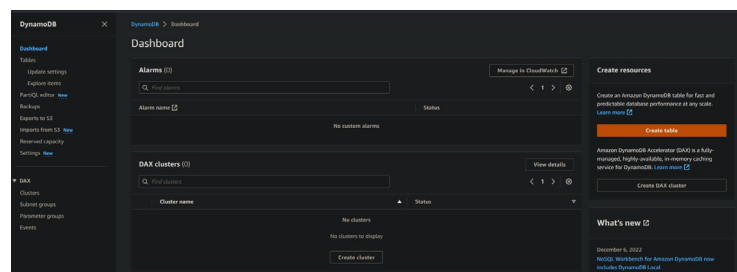
Repeat the same steps to set up the same in the availability zone 2

Instances (2) Info								
Find instance by attribute or tag (case-sensitive)								
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
	CobraKai WebServer	i-0619d056c347404e8	Stopped	t2.micro	–	No alarms +	us-east-1a	–
	CobraKai - AppServer	i-0e47fe0609468870	Stopped	t2.micro	–	No alarms +	us-east-1b	–

## 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.



**Create table**

**Table details** info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

**Table name**  
This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (\_), hyphens (-), and periods (.).

**Partition key**  
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

1 to 255 characters and case sensitive.

**Sort key - optional**  
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

1 to 255 characters and case sensitive.

**Table settings**

☒ **Default settings**  
The fastest way to create your table. You can modify these settings now or after your table has been created.

☐ **Customize settings**  
Use these advanced features to make DynamoDB work better for your needs.

**Default table settings**  
These are the default settings for your new table. You can change some of these settings after creating the table.

Setting	Value	Editable after creation
Capacity mode	Provisioned	Yes
Read capacity	5 RCU	Yes
Write capacity	5 WCU	Yes
Auto scaling	On	Yes
Local secondary indexes	-	No
Global secondary indexes	-	Yes
Encryption key management	Owned by Amazon DynamoDB	Yes
Table class	DynamoDB Standard	Yes

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

**Create bucket** info

Buckets are containers for data stored in S3. [Learn more](#)

**General configuration**

**Bucket name**

Bucket name must be globally unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

**AWS Region**

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

**Object Ownership** info

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.

**Turning off block all public access might result in this bucket and the objects within becoming public**  
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

## 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.

The screenshot displays the AWS Management Console interface for creating an Auto Scaling Group. It is divided into two main panels: 'Choose launch template or configuration' and 'Choose instance launch options'.

**Choose launch template or configuration:**

- Name:** The 'Auto Scaling group name' field is set to 'web server'.
- Launch template:** The 'Launch template' dropdown is set to 'EC2template'. The 'Version' is 'Default (1)'. The description is 'A prod setup for CobraKai'.
- Additional details:** The 'Storage (volumes)' section shows 'Sun Dec 11 2022 13:34:28 GMT-0500 (Eastern Standard Time)'.

**Choose instance launch options:**

- Network:** The 'VPC' dropdown is set to 'vpc-0bb5a7e188853d0d9 (cobrakai-vpc)'. The 'Availability Zones and subnets' section shows two selected subnets: 'us-east-1a | subnet-096716a7d035befa0 (cobrakai-subnet-private1-us-east-1a)' and 'us-east-1a | subnet-0cc15ec57375dab13 (cobrakai-subnet-private3-us-east-1a)'.
- Instance type requirements:** The 'Instance type' dropdown is set to 't2.micro'.

At the bottom of the console, there are navigation buttons: 'Cancel', 'Next', 'Previous', 'Skip to review', and 'Next'.

- 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.

**Load balancing - optional** [info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

☐ **No load balancer**  
 Traffic to your Auto Scaling group will not be fronted by a load balancer.

☐ **Attach to an existing load balancer**  
 Choose from your existing load balancers.

☒ **Attach to a new load balancer**  
 Quickly create a new load balancer to attach to your Auto Scaling group.

**Attach to a new load balancer**  
 Define a new load balancer to create for attachment to this Auto Scaling group.

**Load balancer type**  
 Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, [visit the Load Balancing console](#).

☒ **Application Load Balancer**  
 HTTP, HTTPS

☐ **Network Load Balancer**  
 TCP, UDP, TLS

**Load balancer name**  
 Name cannot be changed after the load balancer is created.

**Load balancer scheme**  
 Scheme cannot be changed after the load balancer is created.

☒ **Internal**

☐ **Internet-facing**

**Network mapping**  
 Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

**VPC**  
 [CobraKai-vpc](#)

**Availability Zones and subnets**  
 You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

☒ us-east-1b

☐ us-east-1a

## 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.

**Create distribution**

**Origin**

**Origin domain**  
Choose an AWS origin, or enter your origin's domain name.  
aws-cloudtrail-logs-112915295193-00c1f88d.s3.us-east-1.amazonaws.com

**Origin path - optional**  
Enter a URL path to append to the origin domain name for origin requests.  
Enter the origin path

**Name**  
Enter a name for this origin.  
aws-cloudtrail-logs-112915295193-00c1f88d.s3.us-east-1.amazonaws.com

**Origin access**  
☐ Public  
 Bucket must allow public access.  
☒ Origin access control settings (recommended)  
 Bucket can restrict access to only CloudFront.  
☐ Legacy access identities  
 Use a CloudFront origin access identity (OAI) to access the S3 bucket.

**Origin access control**  
Select an existing origin access control (recommended) or create a new configuration.  
Select an origin access control Create control setting

**Bucket policy**  
Policy must allow access to CloudFront IAM service principal role.  
☐ I will manually update the policy

**You must update the S3 bucket policy**  
CloudFront will provide you with the policy statement after creating the distribution.

**Add custom header - optional**  
CloudFront includes this header in all requests that it sends to your origin.  
Add header

**Enable Origin Shield**  
Origin Shield is an additional caching layer that can help reduce the load on your origin and help protect its availability.  
☒ No  
☐ Yes

**Additional settings**

**Default cache behavior**

**Path pattern**  
Default (\*)

**Compress objects automatically**  
☐ No  
☒ Yes

**Viewer**

**Viewer protocol policy**  
☒ HTTP and HTTPS  
☐ Redirect HTTP to HTTPS  
☐ HTTPS only

**Allowed HTTP methods**  
☒ GET, HEAD  
☐ GET, HEAD, OPTIONS  
☐ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

**Restrict viewer access**  
If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.  
☒ No  
☐ Yes

**Cache key and origin requests**  
We recommend using a cache policy and origin request policy to control the cache key and origin requests.

☒ Cache policy and origin request policy (recommended)  
☐ Legacy cache settings

**Cache policy**  
Choose an existing cache policy or create a new one.  
CachingOptimized Recommended for S3 origins  
Default policy when CF compression is enabled  
Create policy View policy

**Origin request policy - optional**  
Choose an existing origin request policy or create a new one.  
Select origin policy Create policy

**Response headers policy - optional**  
Choose an existing response headers policy or create a new one.  
Select response headers Create policy

**Additional settings**

**Origins**

Origin name	Origin domain	Origin path	Origin type	Origin Shield region	Origin access
aws-cloudtrail-logs-112915295193-00c1f88d.s3.us-east-1.amazonaws.com	aws-cloudtrail-logs-112915295193-00c1f88d.s3.us-east-1.amazonaws.com		S3	us-east-2	E3J7Q1P1NAHQ1

**Origin groups**

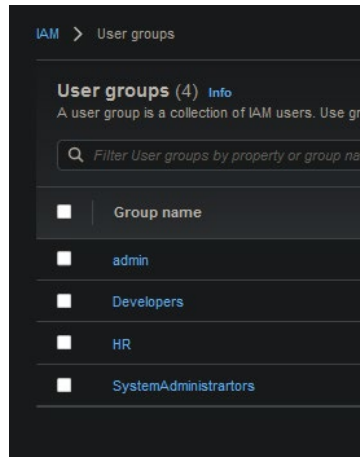
## Recommendation 2: Identity Access Management and User Management

### 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.

### Configure sign-in experience [Info](#)

Your app users can sign in to your user pool with a user name and password, or sign in with a third-party identity provider.

#### Authentication providers

Configure the providers that are available to users when they sign in.

**Provider types**  
Choose whether users will sign in to your Cognito user pool, a federated identity provider, or both. Amazon Cognito has different pricing for federated users and user pool users. [Learn more about pricing](#)

☒ **Cognito user pool**  
Users can sign in using their email address, phone number, or user name. User attributes, group memberships, and security settings will be stored and configured in your user pool.

☐ **Federated identity providers**  
Users can sign in using credentials from social identity providers like Facebook, Google, Amazon, and Apple, or using credentials from external directories through SAML or Open ID Connect. You can manage user attribute mappings and security for federated users in your user pool.

#### Cognito user pool sign-in options [Info](#)

Choose the attributes in your user pool that are used to sign in. If you select only one attribute, or you select a user name and at least one other attribute, your user can sign in with all of the selected options. If you select only phone number and email, your user will be prompted to select one of the two sign-in options when they sign up.

☒ User name  
☒ Email  
☐ Phone number

**User name requirements**

☒ Allow users to sign in with a preferred user name  
☐ Make user name case sensitive

Cognito user pool sign-in options can't be changed after the user pool has been created.

Cancel
Next

- 3) Create a password policy
- 4) Set up MFA

### Do you want to enable Multi-Factor Authentication (MFA)?

Multi-Factor Authentication (MFA) increases security for your app users. If you choose "Optional", individual users can have MFA enabled. You can only choose "Required" when initially creating a user pool, and if you do, all users must use MFA. Phone numbers must be verified if MFA is enabled. You can configure adaptive authentication on the Advanced security tab to require MFA based on the timing of user sign-in attempts. [Learn more about multi-factor authentication.](#)

Note: Separate charges apply for sending text messages.

☐ Off    ☐ Optional    ☒ Required

#### Which second factors do you want to enable?

Your users will be able to configure and choose any of the factors you enable. You must select at least one.

☐ SMS text message  
☒ Time-based One-time Password

#### How will a user be able to recover their account?

When a user forgets their password, they can have a code sent to their verified email or verified phone to recover their account. You can choose the preferred way to send codes below. We recommend not allowing phone to be used for their password reset and multi-factor authentication (MFA). [Learn more.](#)

☐ Email if available, otherwise phone, but don't allow a user to reset their password via phone if they are also using it for MFA.  
☒ Phone if available, otherwise email, but don't allow a user to reset their password via phone if they are also using it for MFA.  
☐ (Not recommended) Email only.  
☐ Phone only, but don't allow a user to reset their password via phone if they are also using it for MFA.  
☐ (Not recommended) Phone if available, otherwise email, and do allow a user to reset their password via phone if they are also using it for MFA.  
☐ None – users will have to contact an administrator to reset their passwords.

#### Which attributes do you want to verify?

Verification requires users to retrieve a code from their email or phone to confirm ownership. Verification of a phone or email is necessary to automatically confirm users and enable recovery from forgotten passwords. [Learn more about email and phone verification.](#)

☐ Email    ☐ Phone number    ☒ Email or phone number    ☐ No verification

If a user signs up with both a phone number and an email address, Cognito will only verify the phone number. For instructions on how to also verify the email address, please see the [documentation](#).

**You must provide a role to allow Amazon Cognito to send SMS messages**

You are currently in a sandbox environment for SMS messages. In order to send messages, go to [Amazon IAM](#) and follow the instructions to verify your phone numbers. You can then rotate your role to a production environment. You may be redirected to the [Amazon IAM console](#) or a [different region](#). [Learn more.](#)

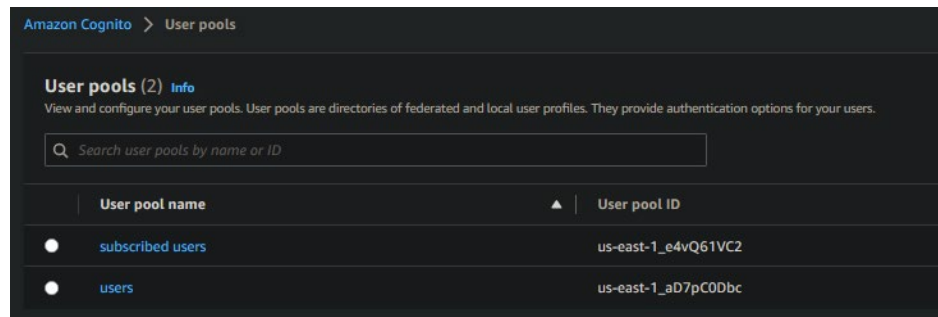
[or, Acknowledge to proceed](#)

In order to send SMS messages to US phone numbers, you must set up an [originating ID](#) in [Amazon Pinpoint](#). You may be redirected to the [Amazon Pinpoint console](#) in a [different region](#). [Learn more.](#)

Note: Separate charges may apply for sending SMS text messages. See the [Worldwide SMS pricing page](#) for more details.

- 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.

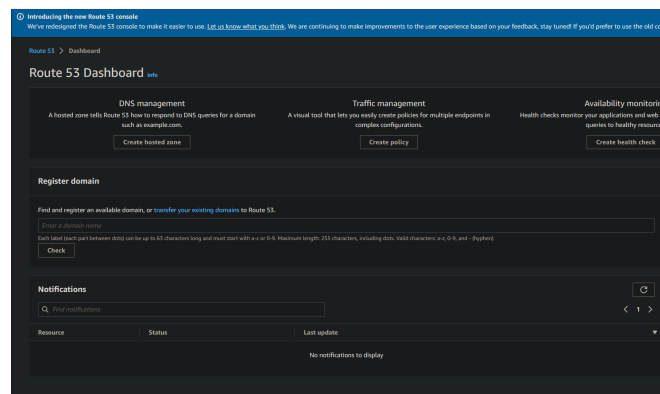
### Recommendation 3: Security against DDoS attacks and Hardware failure

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 AWS 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.

**Describe web ACL and associate it to AWS resources**

**Web ACL details**

**Name**  
CK-WAF

**Description - optional**  
The name must have 1-128 characters, valid characters: A-Z, a-z, 0-9, -, ., #, \$, %, &, ' (apostrophe), and \_ (underscore).

**CloudWatch metric name**  
CK-WAF

**Resource type**  
The name must have 1-128 characters, valid characters: A-Z, a-z, 0-9, -, ., #, \$, %, &, ' (apostrophe), and \_ (underscore).

**Resource type**  
Choose the type of resource to associate with this web ACL.

- ☒ CloudFront distributions
- ☐ Regional resource (Application Load Balancer, API Gateway, AWS AppSync)

**Region**  
Choose the AWS region to create this web ACL in.

Global (CloudFront)

**Associated AWS resources - optional**

Find associated AWS resources

Name	Resource type	Region
EUJXKHQRTXANK - d82f0awd54.cloudfront.net	CloudFront Distribution	Global (CloudFront)

Cancel Next

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.

**Configure metrics**

**Amazon CloudWatch metrics**  
CloudWatch metrics allow you to monitor web requests, web ACLs, and rules.

No results  
There are no results to display

**Request sampling options**  
If you disable request sampling, you can't view requests that match your web ACL rules.

**Options**

- ☐ Enable sampled requests
- ☒ Disable sampled requests
- ☐ Enable sampled requests with exclusions

Cancel Previous Next

**Add managed rule groups**

Managed rule groups are created and maintained for you by AWS and AWS Marketplace sellers.

- AWS managed rule groups
- Cloudbric Corp. managed rule groups
- Cyber Security Cloud Inc. managed rule groups
- F5 managed rule groups
- Fortinet managed rule groups
- GeoGuard managed rule groups
- Imperva managed rule groups
- ThreatSTOP managed rule groups

Cancel Add rules

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

**Bot Control**

**How it works**

**Overview 1**

Find web ACLs

Web ACL name	AWS WAF Bot Control enabled	Bot activity in the last 3 days
CK-WAF	Bot protection not configured	0% in sample requests

Add AWS WAF Bot Control rule group Global (CloudFront)

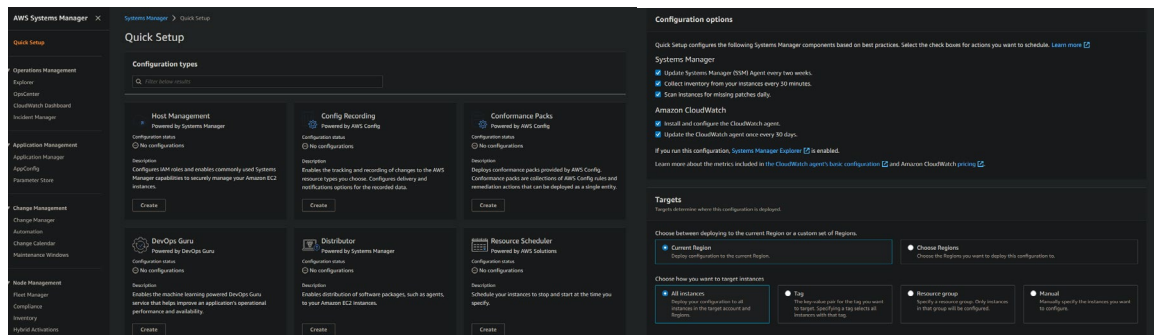
Route 53 along with AWS Shield forwards the malicious traffic to a random CDN and hiding the web application from malicious actors. To pass through the Amazon 53, the user needs to access the web application using set DNS names for e.g., www.cobrakai.com & media.cobrakai.com. anything other than these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

## Recommendation 4: Patching Strategy & Secure System Administration

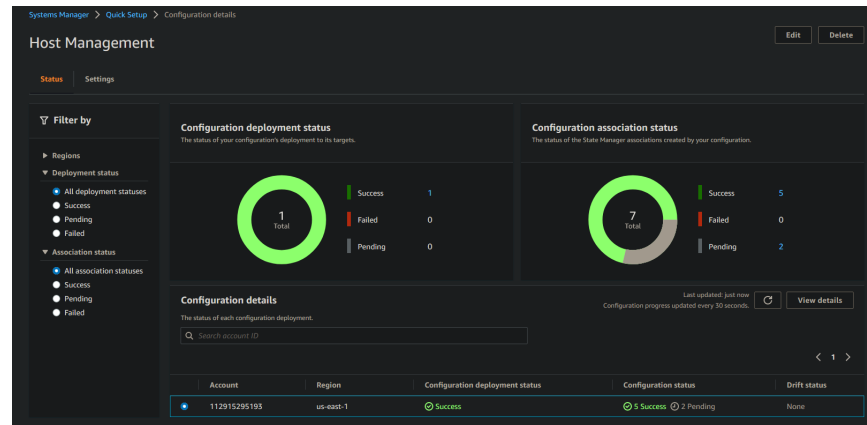
The AWS System Manager is one-in-all console that provides secure end-to-end management solution targeting both the Patching and Secure Administration issues that Cobra Kai had.

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

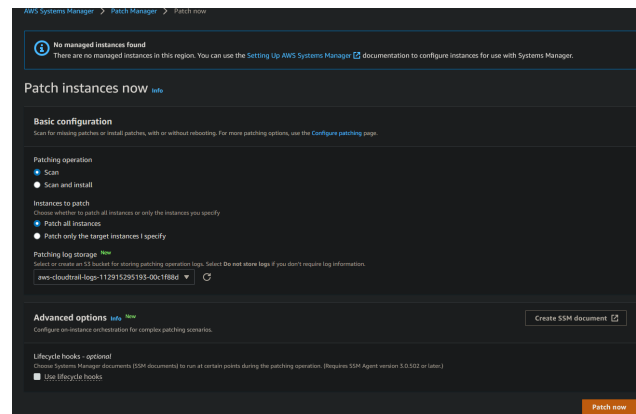


Associations	
Find associations	
Association name	
AWS-QuickSetup-SSMHostMgmt-UpdateSSMAgent-oy8mv	
AWS-QuickSetup-SSMHostMgmt-UpdateCloudWatchAgent-oy8mv	
AWS-QuickSetup-SSMHostMgmt-EnableExplorer-oy8mv	
AWS-QuickSetup-SSMHostMgmt-ManageCloudWatchAgent-oy8mv	
AWS-QuickSetup-SSMHostMgmt-ScanForPatches-oy8mv	
AWS-QuickSetup-SSMHostMgmt-AttachIAMToInstance-oy8mv	
AWS-QuickSetup-SSMHostMgmt-CollectInventory-oy8mv	

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.

**Start options**

**Backup plan options** [Info](#)

- Start with a template**  
Create a backup plan based on a template provided by AWS Backup.
- Build a new plan**  
Configure a new Backup plan from scratch.
- Define a plan using JSON**  
Modify the JSON expression of an existing backup plan or create a new expression.

**Templates**  
Choose a template plan with existing rules.  
Daily-Monthly-1yr-Retention

Backup plan name  
primary-backups  
Backup plan name is case sensitive. Must contain from 1 to 50 alphanumeric or '-', '\_' characters.

► Tags added to backup plan - optional

**Backup rules** [Info](#) [Add backup rule](#) [Delete](#) [Edit](#)  
Backup rules specify the backup schedule, backup window, and lifecycle rules.

Name	Backup vault
DailyBackups	Default
Monthly	Default

▼ **Advanced backup settings**

**Application-consistent backup** [Info](#)  
Enable application-consistent snapshots for the selected third-party software running on EC2.  
☒ Windows VSS

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.

**Create endpoint** [Info](#)

There are three types of VPC endpoints – Interface endpoints, Gateway Load Balancer endpoints, and Gateway endpoints. Interface endpoints and Gateway Load Balancer endpoints are powered by AWS PrivateLink, and use an Elastic Network Interface (ENI) as an entry point for traffic destined to the service. Interface endpoints are typically accessed using the public or private DNS name associated with the service, while Gateway endpoints and Gateway Load Balancer endpoints serve as a target for a route in your route table for traffic destined for the service.

**Endpoint settings**

**Name tag - optional**  
Creates a tag with a key of 'Name' and a value that you specify.  
VPC-DynamoDB

**Service category**  
Select the service category

- AWS services**  
Services provided by Amazon
- PrivateLink Ready partner services**  
Services with an AWS Service Ready designation
- AWS Marketplace services**  
Services that you've purchased through AWS Marketplace
- Other endpoint services**  
Find services shared with you by service name

**Services (1/1)**

Filter services

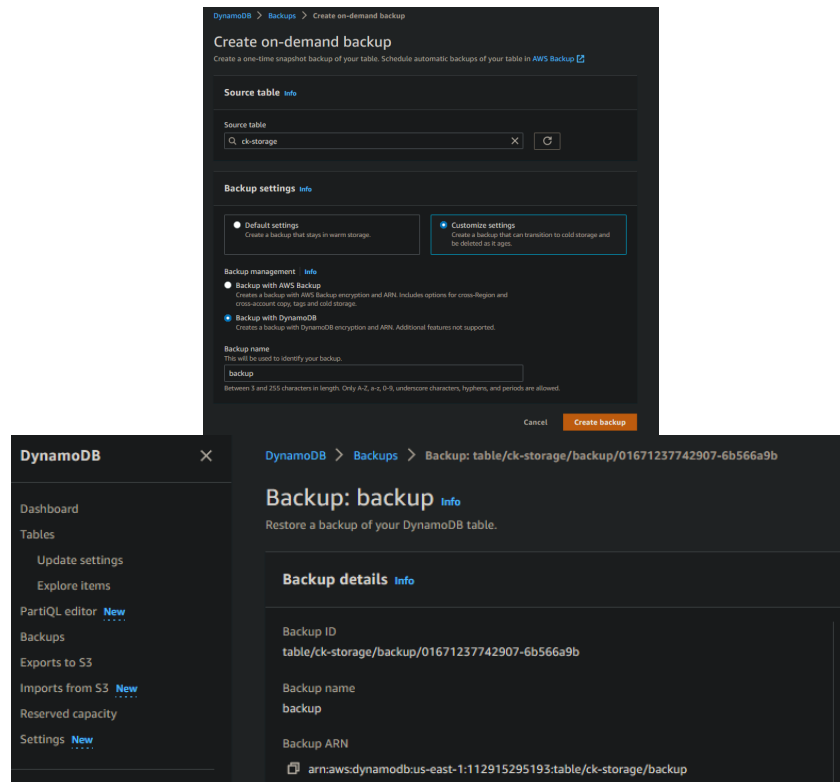
Service Name: com.amazonaws.us-east-1.dynamodb X Clear filters

Service Name	Owner	Type
com.amazonaws.us-east-1.dynamodb	amazon	Gateway

**VPC**  
Select the VPC in which to create the endpoint

**VPC**  
The VPC in which to create your endpoint.  
vpc-0dd2552443fc118df (CobraKali-vpc)

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.

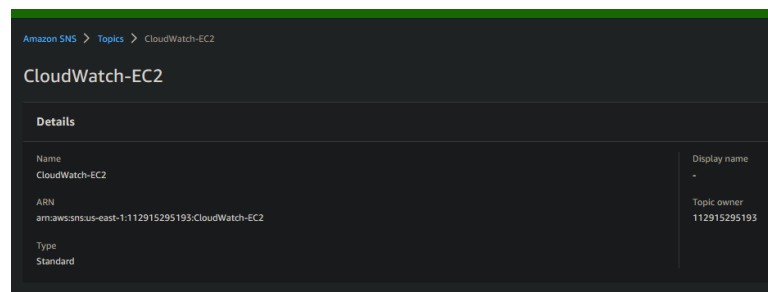
### Recommendation 7: Logs.

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 EC2 utilization.

- 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

The screenshot shows the 'Choose trail attributes' page in the AWS CloudTrail console. The page is divided into several sections:

- General details:** A trail created in the console is a multi-region trail. [Learn more](#)
- Trail name:** Enter a display name for your trail. The input field contains 'cobrakal events'. A note below states: '3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed.'
- Enable for all accounts in my organization:** A checkbox is present. Below it, a note says: 'To review accounts in your organization, open AWS Organizations. [See all accounts](#)'
- Storage location:** Two radio buttons are shown: 'Create new S3 bucket' (selected) and 'Use existing S3 bucket'. The 'Create new S3 bucket' option has a sub-note: 'Create a bucket to store logs for the trail.'
- Trail log bucket name:** Enter a new S3 bucket name and folder (prefix) to store your logs. Bucket names must be globally unique. The input field contains 'aws-cloudtrail-logs-112915295193-00c1f8bd'. A 'Browse' button is next to it. A note below states: 'Logs will be stored in aws-cloudtrail-logs-112915295193-00c1f8bd/AWSLog/112915295193'.
- Prefix - optional:** The input field contains 'prefix'.
- Log file S3E-KMS encryption:** A checkbox is checked and labeled 'Enabled'. A note [Info](#) is next to it.
- Customer managed AWS KMS key:** Two radio buttons are shown: 'New' (selected) and 'Existing'.
- AWS KMS alias:** The input field contains 'Enter KMS alias'. A note below states: 'KMS key and S3 bucket must be in the same region.'
- Additional settings:** A section with a dropdown arrow containing:
  - Log file validation:** A checkbox is checked and labeled 'Enabled'. A note [Info](#) is next to it.
  - SNS notification delivery:** A checkbox is unchecked. A note [Info](#) is next to it.

## 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/>