

**ENPM665 FINAL 2022**  
**IMPLEMENTING COBRA KAI**

**DADHIJA PATEL**

**UID: 119186367**

**Overview:**

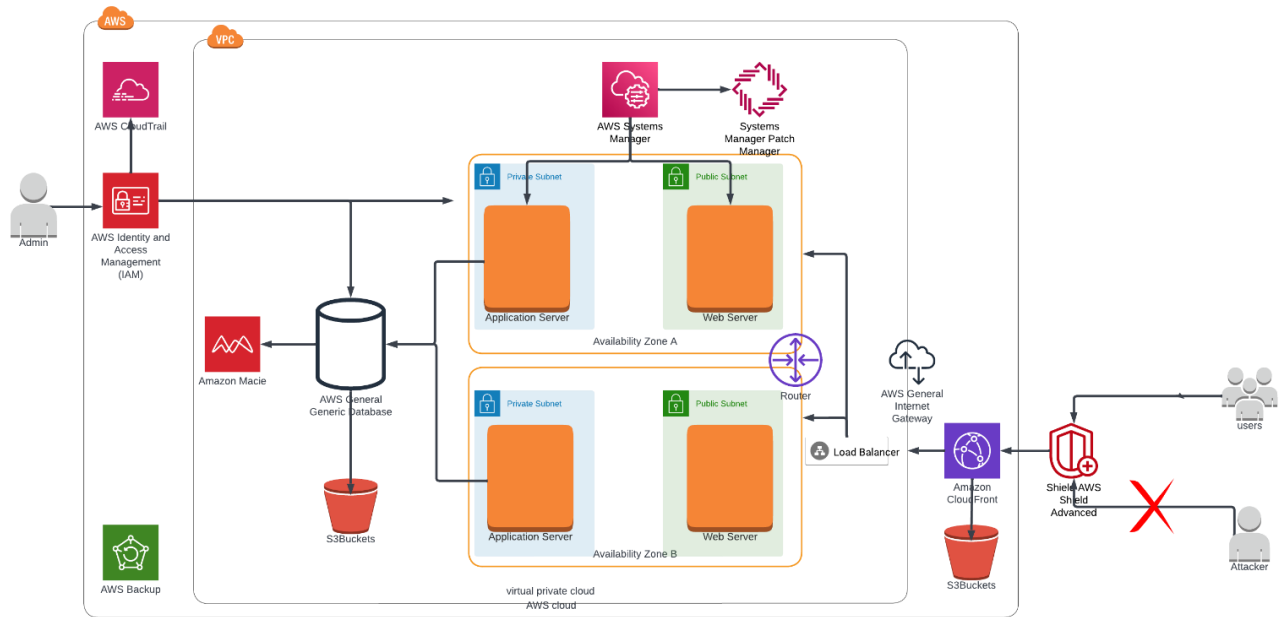
Cobra kai has made an online platform for students to learn through videos. The issue with this website is that it runs from a collection of on-prem servers vulnerable to cyber threats.

As a result, Cobra kai decided to move its platform to the cloud. This report shows the process of implementation of various recommendations that they wish to carry out.

It deals with the following topics:

1. Strategy of deployment
2. Access management (IAM)
3. Cloud trail
4. EC2
5. Patching (System patch manager)
6. S3 buckets and cloud front
7. Database
8. Backup (AWS standard backup)
9. Prevention of DDoS Attacks (AWS shield)
10. VPC
11. PCI compliance

## Diagram:



(Diagram taken from my midterm submission)

## Strategy for Deployment:

The best strategy for Cobra to follow is the refactoring strategy. Refactoring strategy involves creating an entire new infrastructure on the cloud. At the moment, there are multiple issues in every aspect of the structure of Cobra kai. Therefore, the best solution to build it up from scratch to prevent migrating any issues.

Refactoring is a costly and time consuming strategy but it utilizes all the benefits provided by a cloud platform. It is already expensive and not secure to maintain the current architecture of the website.

Refactoring takes a lot of time so we have to plan in such a way as to minimize disruption of service. We can build a temporary server with limited services until the migration process is complete.

The model to use for migration process is IaaS (Infrastructure as a Service). IaaS uses a third party service (here AWS) to provide a framework for application implementation on the cloud. The third party hosts servers and provides resources for the customers.

## Access Management (IAM):

AWS Identity and Access Management assigns particular roles, policies and permissions to different users. It ensures that no unauthorised person can run privileged commands on the web server.

It uses two-factor authentication and proper authorisation for security purposes. It keeps logs of all the actions on the server through CloudTrail.

The main users of Cobra kai website after its implementation are as follows:

1. Johnny Lawrence:  
No direct access to development tools of the website. He is more of a teacher and organizer.  
Permissions given:
  - ReadOnly
2. Miguel Diaz:  
Access to monitoring day to day activities. No access to technical side.  
Permissions Given:
  - CloudTrail\_ReadOnlyAccess
  - AmazonS3ReadOnlyAccess
  - DatabaseAdministrator
  - CloudFrontReadOnlyAccess
3. Aisha Robinson:  
Access to IAM logs, AWS shield, security operations  
Permissions Given:
  - CloudTrail\_FullAccess
  - AWSShieldDRTAccessPolicy (This policy enables the user to access services like CloudFront, EC2 instances, WAF and shield to ensure security of the application on cloud. This is the role of Aisha)
4. Hawk:  
Access to EC2 instances, VPC, CloudFront  
Hawk will have permissions related to website development.  
Permissions Given:
  - AmazonEC2FullAccess
  - AmazonVPCFullAccess
  - CloudFrontFullAccess
5. Demetri:  
Access to EC2 instances, VPC, CloudFront  
Demetri will have permissions related to website development.  
Permissions Given:
  - AmazonEC2FullAccess

- AmazonVPCFullAccess
- CloudFrontFullAccess

6. Bert:

Administrator access

Bert is the System administrator.

Permissions given:

- AdministratorAccess

## Steps for implementation:

1. Open AWS console and search for IAM in services.
2. On the side panel, click on Users.
3. Then in the page, click on add users.
4. In the first dialogue box, give a name to the user and select the credential type

Add user

12345

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

Add another user

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type\*

☒ Access key - Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☒ Password - AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.

\* Required

Cancel
Next: Permissions

5. Then click on Next:Permissions button.
6. Then assign permissions according to the user.

▼ Set permissions

Add user to group

Copy permissions from existing user

Attach existing policies directly

Create policy

Filter policies ▼ AdministratorAccess UID\_119186367 Showing 4 results

	Policy name ▼	Type	Used as
<input checked="" type="checkbox"/>	AdministratorAccess	Job function	Permissions policy (1)
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	None
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
<input type="checkbox"/>	AWSAuditManagerAdministratorAccess	AWS managed	None

Cancel Previous Next: Tags

7. Next, review the user and click on create.

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	Bert_UID_119186367
AWS access type	Programmatic access and AWS Management Console access
Console password type	Autogenerated
Require password reset	Yes
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AdministratorAccess
Managed policy	IAMUserChangePassword

Cancel Previous Create user

8. Send credentials to the particular user through e-mail.

**Success**

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://963281487438.signin.aws.amazon.com/console>

Download .csv

	User	Access key ID	Secret access key	Password	Email login instructions
▶	✔ Bert_UID_11...	AKIA6ASANA ZHPSEDHAW E	***** Show	***** Show	Send email

9. Create all users in the same way.

**Users (6)** [Info](#)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

<input type="checkbox"/>	User name	Groups	Last activity	MFA	Password ...	Acti
<input type="checkbox"/>	<a href="#">Aisha_Robinson</a>	None	Never	None	✓ 4 minutes ago	✓
<input type="checkbox"/>	<a href="#">Bert_UID_119186367</a>	None	Never	None	✓ 9 minutes ago	✓
<input type="checkbox"/>	<a href="#">Dadhija</a>	<a href="#">Dadhija_Cloudsec</a>	✓ 38 days ago	None	None	⚠
<input type="checkbox"/>	<a href="#">Demetri</a>	None	Never	None	✓ Now	✓
<input type="checkbox"/>	<a href="#">Hawk</a>	None	Never	None	✓ 2 minutes ago	✓
<input type="checkbox"/>	<a href="#">Minguel_Diaz</a>	None	Never	None	✓ 5 minutes ago	✓

## Notes:

- We can copy permissions of one user to another. Here Hawk and Demetri have same permissions

▼ Set permissions

Select an existing user from which to copy policies and group membership.

Copy permissions from existing user

Showing 5 results		
User name	Groups	Attached policies
<input type="radio"/> <a href="#">Aisha_Robinson</a>	None	AWSShieldDRTAccessPolicy and 2 more
<input type="radio"/> <a href="#">Bert_UID_119186367</a>	None	AdministratorAccess and 1 more
<input type="radio"/> <a href="#">Dadhija</a>	<a href="#">Dadhija_Cloudsec</a>	None
<input checked="" type="radio"/> <a href="#">Hawk</a>	None	CloudFrontFullAccess and 3 more
<input type="radio"/> <a href="#">Minguel_Diaz</a>	None	DatabaseAdministrator and 4 more

[Cancel](#) [Previous](#) [Next: Tags](#)

- I have not made any groups as all users have different permissions. If multiple users have the same permissions, it is easier to make a user group with all those permissions and just assign the user to the relevant group. Users can stay in multiple groups at a time. If permission of a user changes, the group of the user can be changed easily.
- With the help of access advisor, we can check the services that can be accessed by the user as well as when those services are accessed.

## Summary

Delete user

User ARN `arn:aws:iam::963281487438:user/Bert_UID_119186367`Path `/`

Creation time 2022-12-07 18:41 EST

Permissions Groups Tags Security credentials Access Advisor

Access Advisor shows the services that this user can access and when those services were last accessed. Review this data to remove unused permissions. [Learn More](#)

**Allowed services** (339)

Access Advisor reports activity for services and EC2, IAM, Lambda, and S3 management actions. To view actions, choose the service name from the list. Recent service activity usually appears within 4 hours. Service activity is reported for the past 400 days. [Learn More](#)

**i** Last accessed information is available for EC2, IAM, Lambda, and S3 management actions.

## Cloud Trail:

CloudTrail monitors the activity of users in an AWS environment. It stores logs of all the actions taken within the cloud environment. It helps figure out inconsistencies in patterns to detect any malicious activity.

It can be integrated with IAM to store the records of users. It shows event history for events occurring during a 90-day period. Users can see these events and their details by going to the event history page in the dashboard of CloudTrail.

CloudTrail > Event history

**Event history** (100+) [Info](#)

Event history shows you the last 90 days of management events.

Lookup attributes

Read-only ☐ false

30m 1h 3h 12h Custom

<input type="checkbox"/>	Event name	Event time	User name	Event source	Resource type
<input type="checkbox"/>	<a href="#">AttachUserPolicy</a>	December 07, 2022, 18:48:24 (U...	root	iam.amazonaws.com	AWS::IAM::User,
<input type="checkbox"/>	<a href="#">CreateLoginProfile</a>	December 07, 2022, 18:47:01 (U...	root	iam.amazonaws.com	AWS::IAM::User
<input type="checkbox"/>	<a href="#">AttachUserPolicy</a>	December 07, 2022, 18:47:00 (U...	root	iam.amazonaws.com	AWS::IAM::User,
<input type="checkbox"/>	<a href="#">CreateAccessKey</a>	December 07, 2022, 18:47:00 (U...	root	iam.amazonaws.com	AWS::IAM::Acces
<input type="checkbox"/>	<a href="#">CreateUser</a>	December 07, 2022, 18:47:00 (U...	root	iam.amazonaws.com	AWS::IAM::User,
<input type="checkbox"/>	<a href="#">AttachUserPolicy</a>	December 07, 2022, 18:47:00 (U...	root	iam.amazonaws.com	AWS::IAM::User,

Trail is different from event history as it stores all the events (not just for the last 90 days). Users can create a trail through the CloudTrail console and store the data in an s3 bucket.

An additional feature of CloudTrail is CloudTrail lake. It enables users to run SQL queries on the recorded events. It optimizes the JSON-based system for easy retrieval of data.

## **EC2:**

Elastic Compute Cloud reduces the hardware required to run an application. It can deploy servers and configure AWS resources for the application while migration. It is scalable.

EC2 instances are useful in patching, backup and many other services provided by AWS. For example, a Linux EC2 instance can be used to keep track of the application and implement changes in the code when required. Specific roles can be assigned to instances for specific services on the cloud.

AWS deploys VPC in EC2 instances to provide network security and monitor network traffic. Various data availability zones can be created for protection and preservation of Data.

A key pair is generated with the creation of EC2 instance so every time the instance is accessed, this key pair can be used. It confirms authorization for trusted parties.

You can add a monitoring plan to the EC2 instance. This should include the resources and services that will be monitored.

We can create different tags to manage EC2 instances in the environment.



## **Patching (System Patch Manager):**

Currently, Cobra kai deals with its technical issues on the fly. Therefore, it needs a proper patching strategy that checks for faults at regular intervals and fixes them.

Patch manager monitors the instances (both windows and Linux) and automates the installation of missing patches. Moreover, it can work on individual instances or groups of instances. The user can create baselines or use predefined baselines for patching.

Patch manager implements automated approval of patches after a certain period (like seven days). Users can specify this period while creating baselines.

When patching is required, the patch manager does not install all the patches available but rather a small number of specific patches that will improve the security.

A feature of patches for Linux EC2 instances is that the patches are not only implemented from the specified baselines of the instance but also from various repositories.

While patching, there are approved and rejected patches. Updates happen through the approved patches, and rejected patches won't be executed.

## Steps for implementation:

1. Create a role in IAM to give access to system manager to an EC2 instance.
  - Go into IAM console and click on create role.
  - Select AWS service for EC2 instances and select the use case as EC2.

Trusted entity type

- ☒ **AWS service**  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- ☐ **AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- ☐ **Web identity**  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- ☐ **SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- ☐ **Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

**Use case**  
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

- ☒ **EC2**  
Allows EC2 instances to call AWS services on your behalf.
- ☐ **Lambda**  
Allows Lambda functions to call AWS services on your behalf.

- Give the permission of EC2RoleForSystemManager to the role.

Permissions policies (Selected 1/800) [Info](#)

Choose one or more policies to attach to your new role.

Search:  17 matches

Filters: "ssm" Clear filters

	Policy name	Type	Description
<input checked="" type="checkbox"/>	AmazonEC2Rolefor...	AWS m...	This policy will soon be deprecated. Please use AmazonSSMManagedIns...
<input type="checkbox"/>	AmazonSSMAutom...	AWS m...	Provides access to view automation executions and send approval deci...
<input type="checkbox"/>	AmazonSSMManag...	AWS m...	The policy for Amazon EC2 Role to enable AWS Systems Manager servic...
<input type="checkbox"/>	AmazonSSMDirecto...	AWS m...	This policy allows SSM Agent to access Directory Service on behalf of th...
<input type="checkbox"/>	AmazonSSMFullAcc...	AWS m...	Provides full access to Amazon SSM.

- Click on create role.

Roles (3) <a href="#">Info</a>			
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.			
<input type="text" value="Search"/> <span>&lt; 1 &gt;</span> <span>⚙️</span>			
<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linked Role)	-
<input type="checkbox"/>	<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linked Role)	-
<input type="checkbox"/>	<a href="#">EC2role_UID_119186367</a>	AWS Service: ec2	-

## 2. Create an EC2 instance

- Go to EC2 console and click on launch instance.
- Give a name to the instance and select the Amazon Machine Image.

- Select the role created in the first step.

**Advanced details** [Info](#)

**Purchasing option** [Info](#)  
☐ Request Spot Instances  
 Request Spot Instances at the Spot price, capped at the On-Demand price

**Domain join directory** [Info](#)  

Select
 ▼

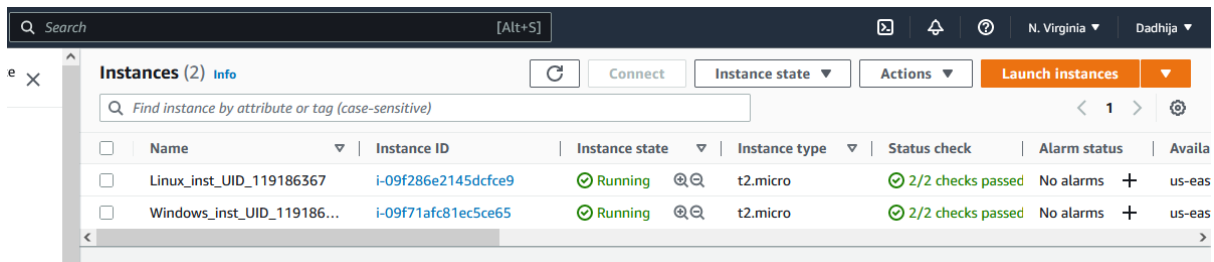
↻ [Create new directory](#)

**IAM instance profile** [Info](#)  

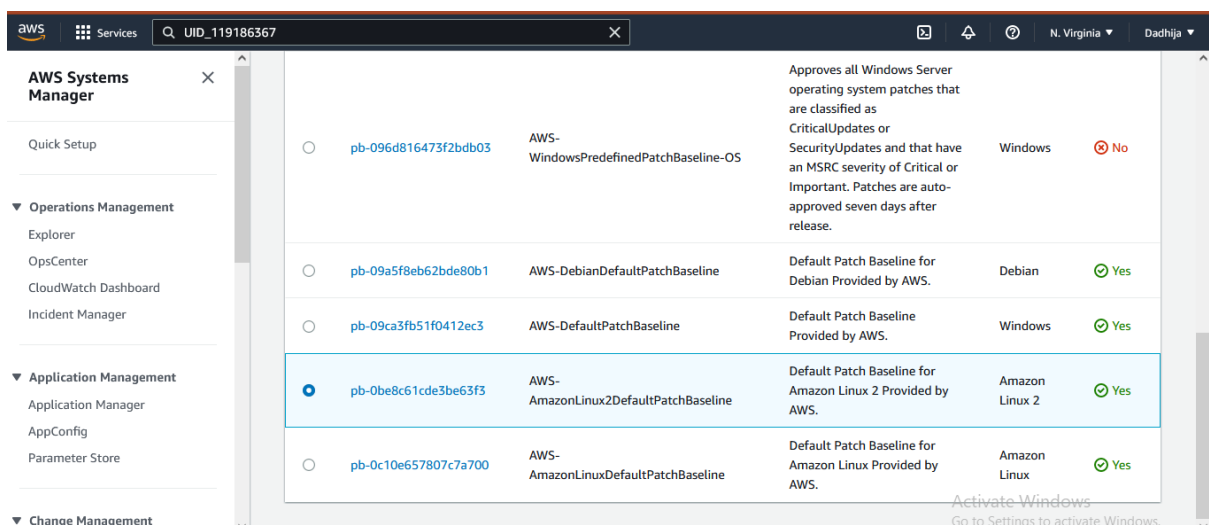
EC2role\_UID\_119186367  
 arn:aws:iam::963281487438:instance-profile/EC2role\_UID\_119186367
 ▼

↻ [Create new IAM profile](#)

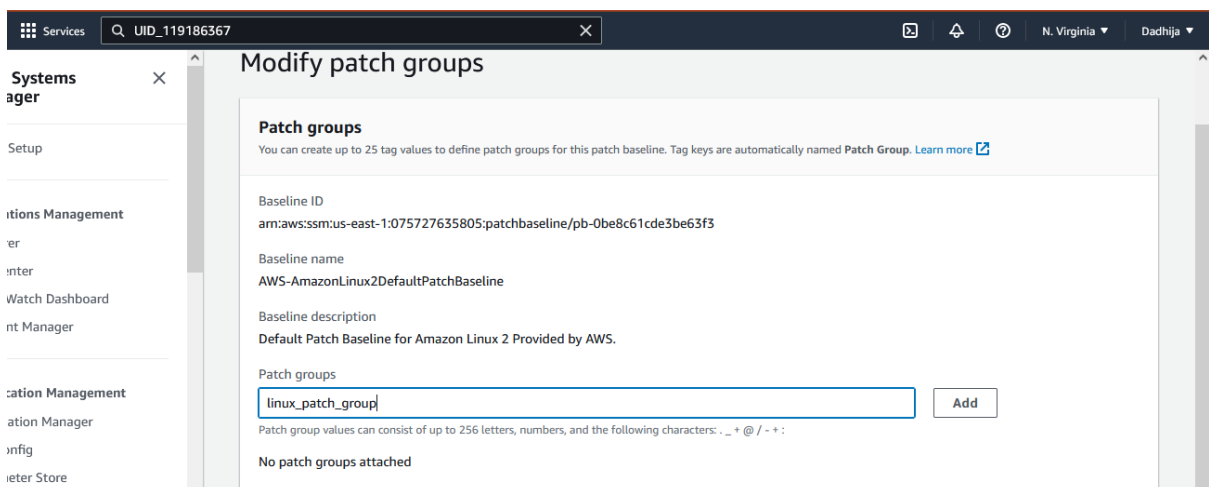
- Launch instance.



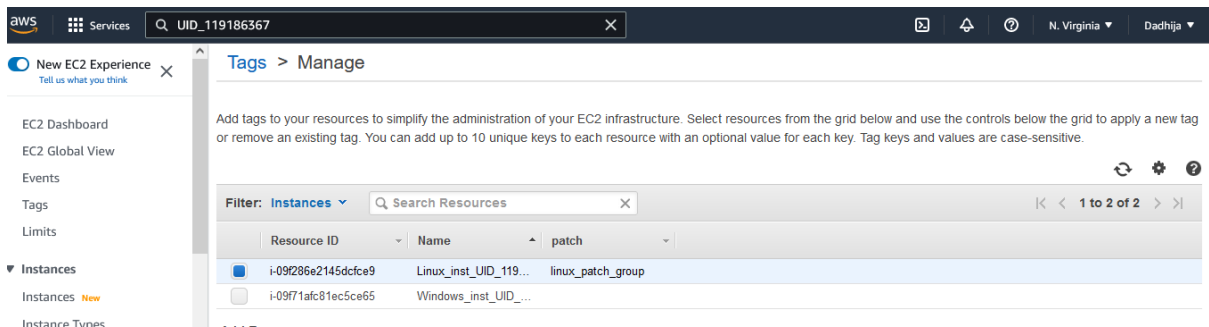
3. Go to AWS system manager and select patch manager.
4. A list of default patch baselines is given. Select a Linux based patch for Linux instances.



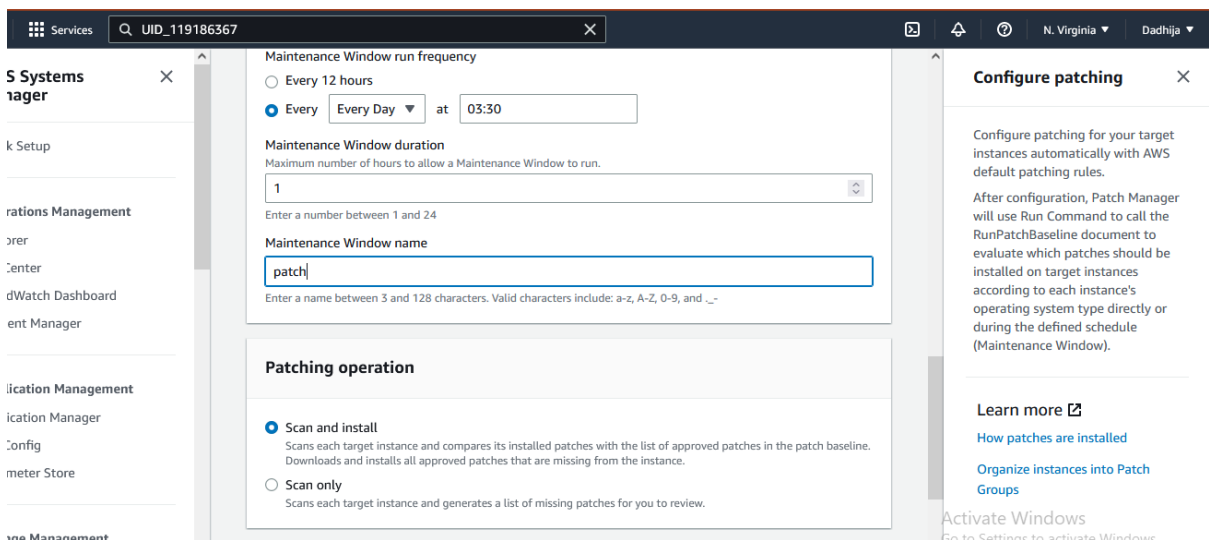
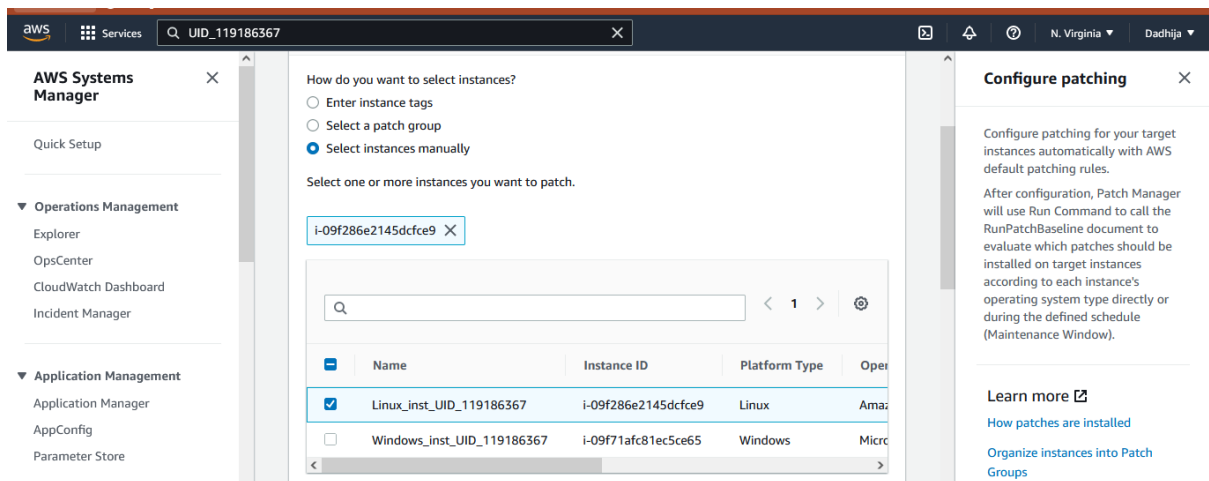
5. Create a patch group to assign to the instance. Go to modify patch groups and give a name to the group.



6. Attach tags to the EC2 instance according to the patch group.



## 7. Configure the patch. It is done by specifying the instances involved in the patch, monitoring times and maintenance frequency.



## 8. Create patch

Q UID\_119186367

Window ID: mw-073a804da1fb7f972

Description

Tasks

History

Targets

Tags

Window ID

mw-073a804da1fb7f972

Name

patch

Description

-

State

Enabled

Cron/Rate expression

cron(30 03 ? \* \*)

Duration

1 hour

Next execution time

Mon, Dec 12, 2022, 3:30:00 AM UTC

Cutoff point

0 hours before window closes

Window schedule timezone

-

Window start date

-

Window schedule offset

-

Window end date

-

Allow unregistered targets

Yes

Activ  
Go to

Q UID\_119186367

AWS Systems Manager > Maintenance Windows > Window ID: mw-073a804da1fb7f972 > Tasks

Window ID: mw-073a804da1fb7f972

EditDeleteActions

DescriptionTasksHistoryTargetsTags

Tasks

EditDeregister taskRegister tasks

Window task ID

Priority

Name

Task ARN

Type

Tan

bd0a2e6d-78e9-4616-af36-35d3c6a11827

1

PatchingTask

AWS-RunPatchBaseline

RUN\_COMMAND

1

## **S3 buckets and cloudfront:**

The CloudFront service ensures that cobra kai customers receive fast streaming. It happens because Cloudfront reduces the number of origin requests. As a result, data is only fetched from the server when required. This process reduces time. It also ensures that customers are not frustrated due to slow speed service.

S3 buckets store objects on the cloud platform. Consequently, CloudFront can retrieve data from the S3 buckets and save time.

Users don't have to worry about the size of the data stored in an S3 bucket as its self-scaling. It minimizes the use of servers in the application because all the static data is stored and retrieved from buckets. Consequently, the streaming speed increases and cost reduces.

The Cobra Kai website will store all its training videos in S3 buckets. Integrating S3 with CloudFront will further result in better management of the stored data. CloudFront is a content delivery network, so merging it with an S3 bucket rather than using S3 alone will yield better results.

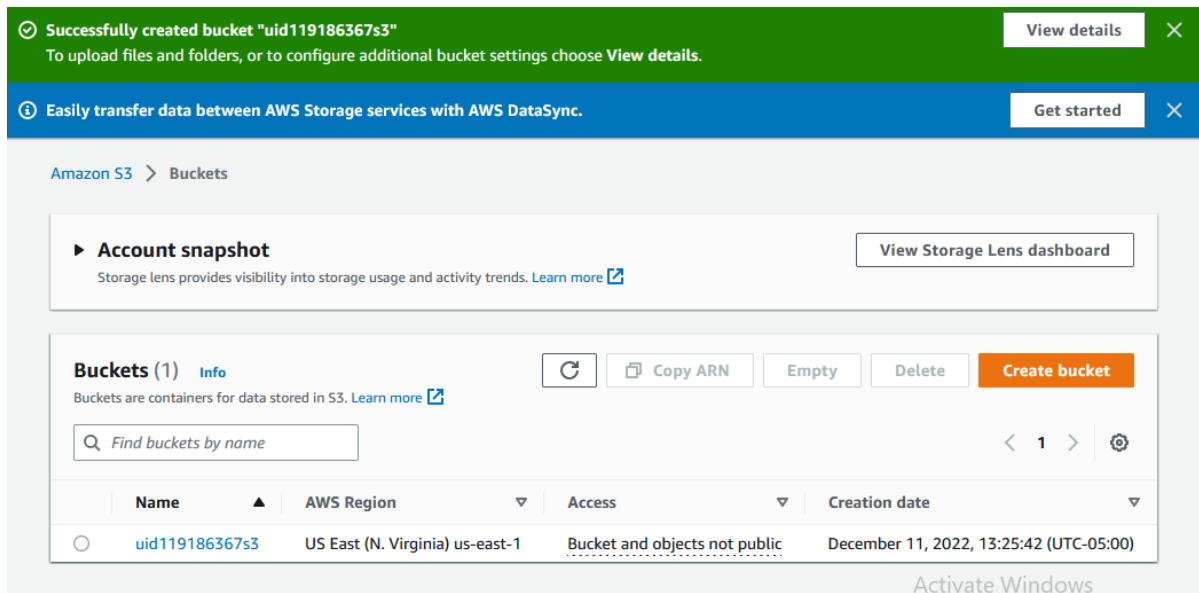
Sending data through CloudFront is more secure. CloudFront can protect against DDoS attacks as it combines with AWS shield, AWS WAF and firewall protection.

CloudFront has the options to restrict certain user requests and protocols to make the cloud environment more secure.

A CloudFront distribution enables the users to specify the source of data, and track and manage the delivery of this data. The CloudFront service uses edge devices to store and retrieve data so data transmission to the customers will be faster. It does not have to go to the origin for each and every request.

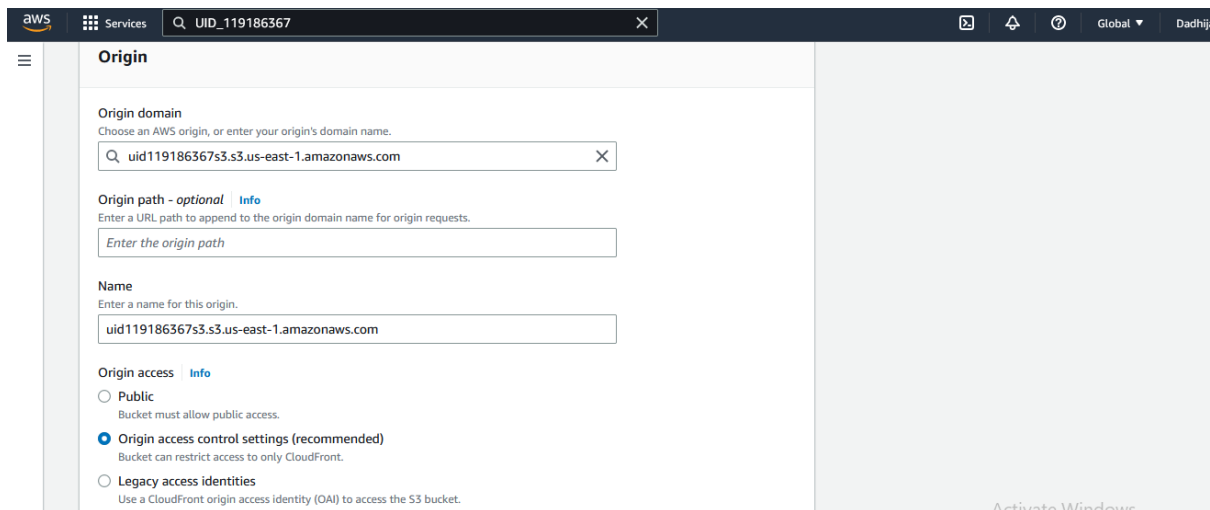
## **Steps for implementation:**

1. Open AWS and go to S3 console.
2. Click on create bucket.
3. Specify a bucket name and region.
4. Give properties to the bucket.
5. Click on create bucket.



All the data can be stored in S3 buckets. It can be uploaded from an on-prem device or be linked with one of the other AWS services.

6. Open CloudFront Console.
7. Click on create a CloudFront distribution.
8. Specify the origin properties. Give the origin domain as the bucket we just created. Users have to specify a path if the bucket is created from a different account but here that is not a requirement.



9. Specify details for the distribution like which HTTP methods to run, what access to provide and which edge locations should be considered in the distribution.



aws Services

Compress objects automatically [Info](#)

☒ 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

Activate Windows

aws Services

Choose the price class associated with the maximum price that you want to pay.

☒ Use all edge locations (best performance)  
☐ Use only North America and Europe  
☐ Use North America, Europe, Asia, Middle East, and Africa

**AWS WAF web ACL - optional**

Choose the web ACL in AWS WAF to associate with this distribution.

**Alternate domain name (CNAME) - optional**

Add the custom domain names that you use in URLs for the files served by this distribution.

③ To add a list of alternative domain names, use the [bulk editor](#).

**Custom SSL certificate - optional**

Associate a certificate from AWS Certificate Manager. The certificate must be in the US East (N. Virginia) Region (us-east-1).

[Request certificate](#)

**Supported HTTP versions**

Add support for additional HTTP versions. HTTP/1.0 and HTTP/1.1 are supported by default.

☒ HTTP/2  
☒ HTTP/3

Activate Windows  
Go to Settings to activate Windows

## 10. Create the CloudFront distribution

aws Services

CloudFront > Distributions

**Distributions (1)** [Info](#)

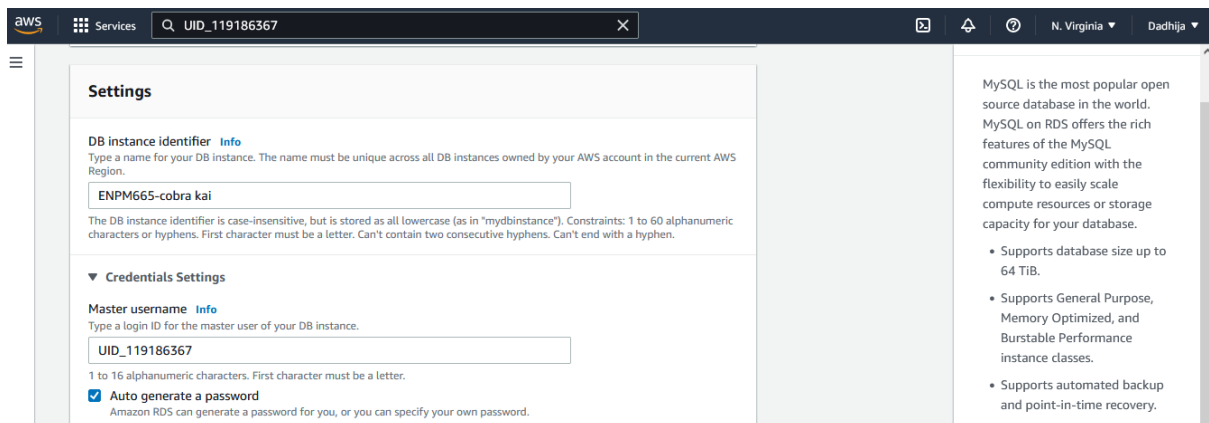
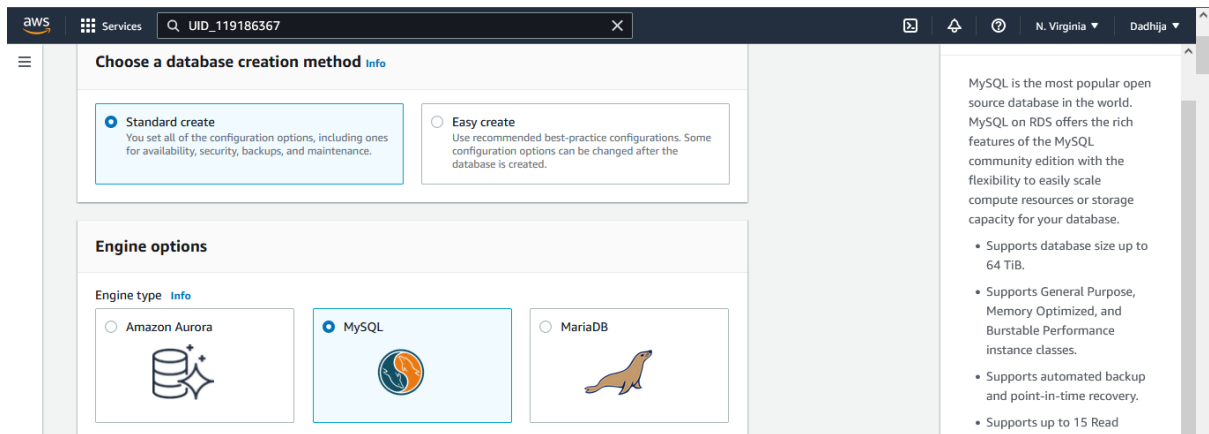
<input type="checkbox"/>	ID	Description	Domain name	Alternate dom...	Origins	Status	Last modified
<input type="checkbox"/>	E12OVOW09U1993	-	d8lvgataqf3v3.cl...	-	uid119186367s3.s3.us-	Enabled	Deploying

## Database:

The AWS database stores data from customers. AWS database is self-scaling, so we don't have to worry about the space the data requires. It monitors the data continuously.

## Steps for implementation:

1. Go to database console and choose a database type



2. Connect it to the EC2 instances that will store data.

aws Services Q UID\_119186367 X

≡

### Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

☐ Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

☒ Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

### EC2 Instance [Info](#)

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-007f6bcdc1cc8e39e

Backup\_inst ▼

### 3. Type in a name for the database

aws Services Q UID\_119186367 X

≡

Database options, encryption turned off, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

### Database options

Initial database name [Info](#)

CobraKai

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)

default.mysql8.0 ▼

Option group [Info](#)

default:mysql-8-0 ▼

### Backup

☒ Enable automated backups

Creates a point-in-time snapshot of your database

### 4. Configure additional settings

aws

Services

UID\_119186367

X

☰

⚠ Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#).

**Backup retention period** [Info](#)  
The number of days (1-35) for which automatic backups are kept.  

1 ▾

 day

**Backup window** [Info](#)  
The daily time range (in UTC) during which RDS takes automated backups.  

☐ Choose a window

☒ No preference

☒ Copy tags to snapshots

**Encryption**  

☐ Enable encryption  
Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. [Info](#)

aws

Services

UID\_119186367

X

🔍

☰

RDS service-linked role

ℹ Ensure that general, slow query, and audit logs are turned on. Error logs are enabled by default. [Learn more](#)

**Maintenance**  
Auto minor version upgrade [Info](#)  

☒ Enable auto minor version upgrade  
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

**Maintenance window** [Info](#)  
Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.  

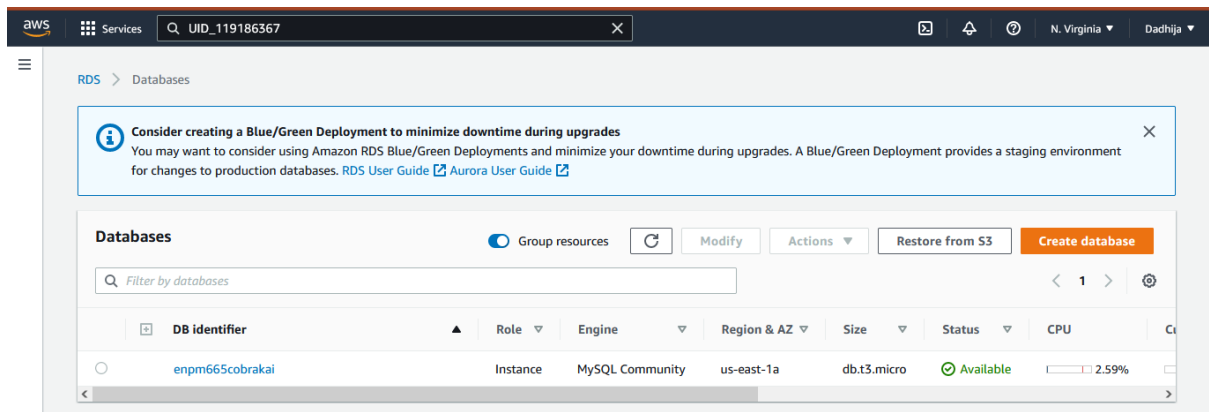
☐ Choose a window

☒ No preference

**Deletion protection**  

☐ Enable deletion protection  
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

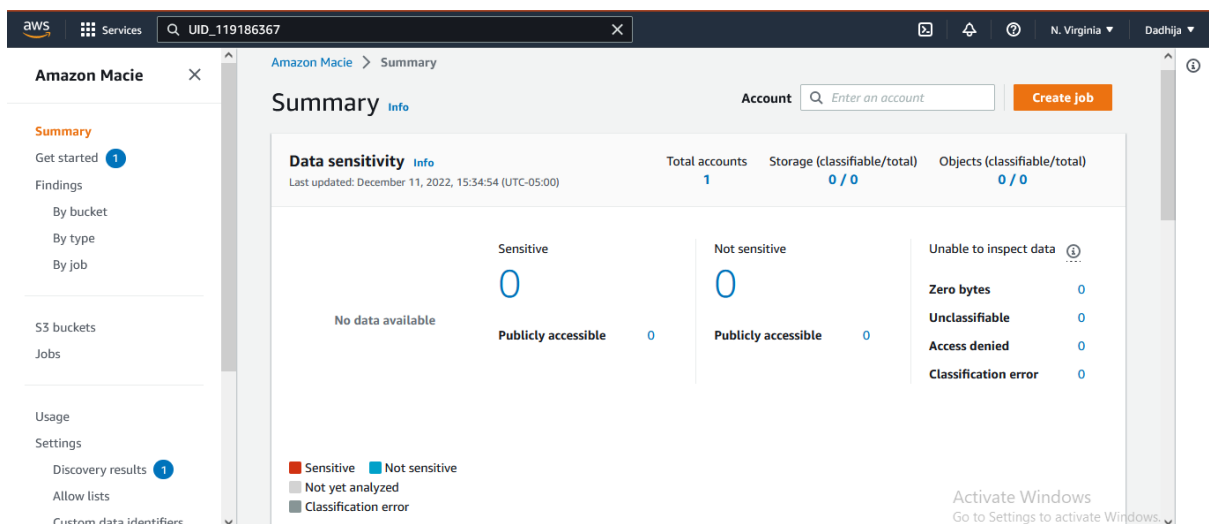
## 5. Create database



## Amazon Macie:

It protects sensitive data. It is a data security and data privacy service provided by AWS for secure storing of sensitive information of customers. It uses machine learning and pattern matching to secure the data.

Macie takes the data in S3 buckets and analysis it. It generates a report of the sensitivity of a data and enables the users to decide which data to protect. It monitors the data and alerts the user when it discovers any sensitive data. The above process can be combined with normal workflow for enhanced security.



## Backup:

Cobra kai does not have a proper backup strategy. Therefore, data protection can be done by AWS backup service. It supports the backup of EC2 instances, S3 buckets and database tables. AWS backup creates snapshots of data for backup.

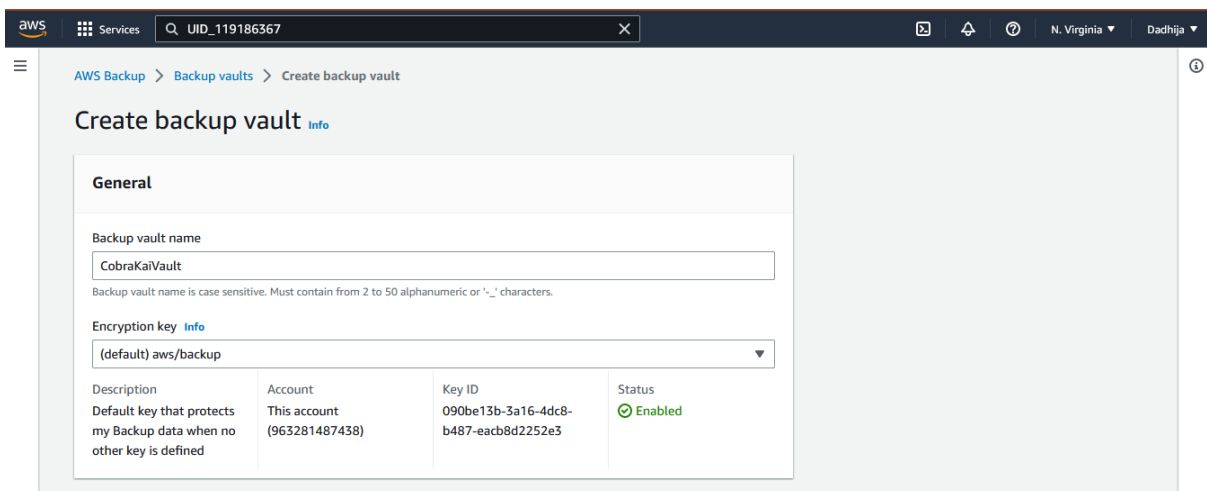
Backup works by creating and implementing a backup plan. The backup plan specifies the resources to be backed up and the frequency with which the backup should execute. During the first backup, all data is stored. From the second backup, only the changes are modified in the previously backed up data.

Backup vaults are useful in organising the backup plans. They have key encryption management for better security. It can give special permissions for access to all users. Therefore, an attacker can't perform malicious activity on the backup data. Recovery points are stored in vaults.

Backups can be restored without any damage. When a resource in the cloud needs to be restored, backup creates a new resource and configures it according to the stored data. This prevents the mauling or destruction of existing resources.

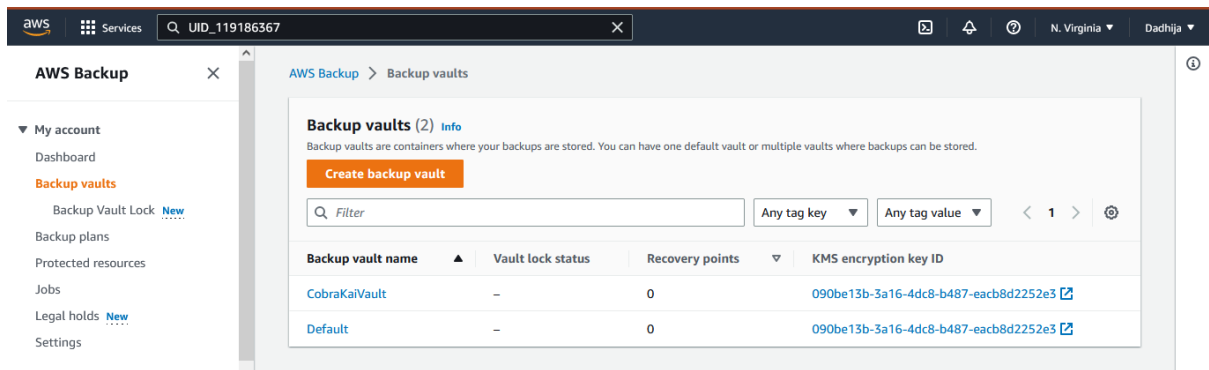
## Steps for implementation:

1. Create a backup vault.

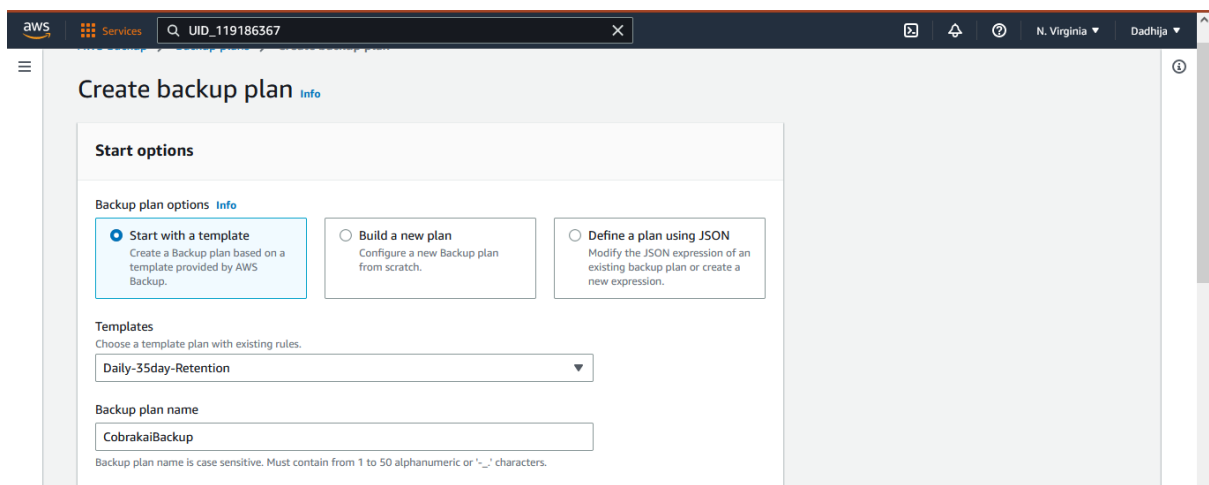


The screenshot shows the AWS Backup console interface for creating a new backup vault. The breadcrumb navigation indicates the path: AWS Backup > Backup vaults > Create backup vault. The main heading is 'Create backup vault' with an 'Info' link. Below this, there is a 'General' section with the following details:

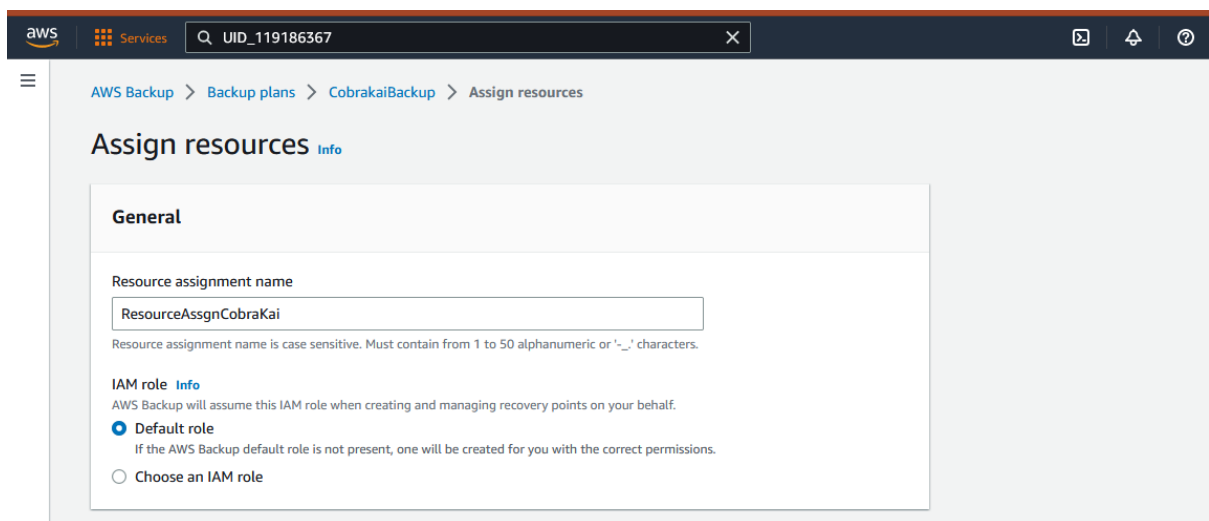
General			
Backup vault name			
<input type="text" value="CobraKaiVault"/>			
<small>Backup vault name is case sensitive. Must contain from 2 to 50 alphanumeric or '-' characters.</small>			
Encryption key <a href="#">Info</a>			
<input type="text" value="(default) aws/backup"/>			
Description	Account	Key ID	Status
Default key that protects my Backup data when no other key is defined	This account (963281487438)	090be13b-3a16-4dc8-b487-eacb8d2252e3	Enabled



2. Go to the create plan page in backup console.
3. Click on create plan.



4. Select the backup vault that was created in the first step to define the conditions of the backup
5. Assign AWS resources that are to be backed up. (here I have chosen all active resources to backup)



**Resource selection** [Info](#)

Assign resources to this Backup plan using tags and resource IDs.

**1. Define resource selection** [Info](#)

Protect all resources or specify resources by type or ID.

☒ **Include all resource types**  
Protect all resource types that are enabled in your account.

☐ **Include specific resource types**  
Choose resources by type or specify individual resources by ID.

**2. Refine selection using tags - optional** [Info](#)

Filter resources by tags. For multiple tags, resources will only be assigned to the backup plan if they satisfy all tag conditions.

Key	Condition for value	Value
patch	All values	Enter value

[Add tags](#)

You can add up to 29 more tags.

## 6. Click on create backup

**AWS Backup**

[AWS Backup](#) > [Backup plans](#) > CobrakaiBackup

**CobrakaiBackup** [Delete](#) [View JSON](#)

**Summary**

Backup plan name CobrakaiBackup	Version ID MTlhMzc2N2UtNjQ5VS00NGU 2LThIMWEtMzU4NDIjY2UyYjY 5	Last modified December 11, 2022, 14:57:01 (UTC-05:00)	Last runtime -
Backup plan ID 7e344d98- e08b-4195-8986-82ddaa44d1 2d			

The backup rules are defined in the vault created or the default vault that is already provided.



## **AWS shield:**

Cobra kai has a particular threat from DDoS attacks. Therefore, its prevention is of the utmost importance. We can use AWS shield to mitigate DDoS threats.

A targeted DDoS attack overflows the network with requests which leads to Denial of service for legitimate users. The goal is to build an architecture that is resilient towards such floods. In other words, a DDoS attack makes no impact on the functioning of the website.

AWS shield standard:

It protects against level 3 and level 4 DDoS attacks. It protects against network and transport layer attacks. It is an automatically configured service provided free of cost by the AWS cloud platform.

It is built into the AWS services that we use on the cloud platform.

AWS shield advance:

It is a service used to protect against targeted DDoS attacks. It monitors data across all amazon resources and looks for availability threats. It handles such threats by reducing traffic and re-routing techniques. It includes AWS WAF and firewall management.

Steps for implementation:

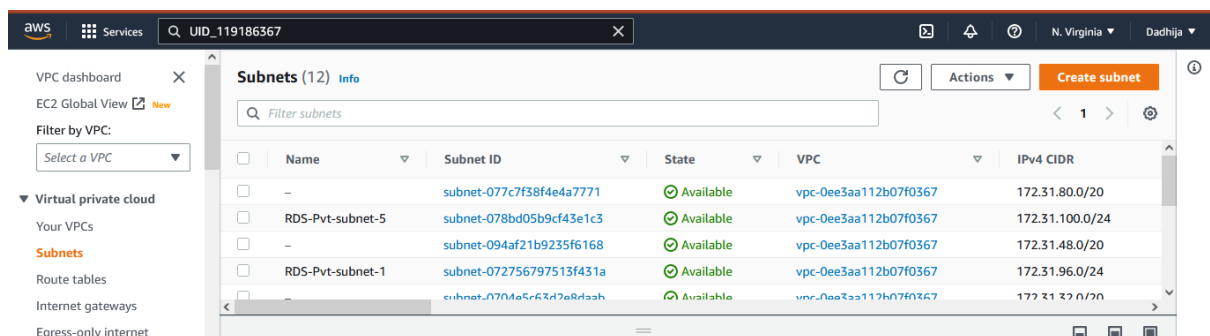
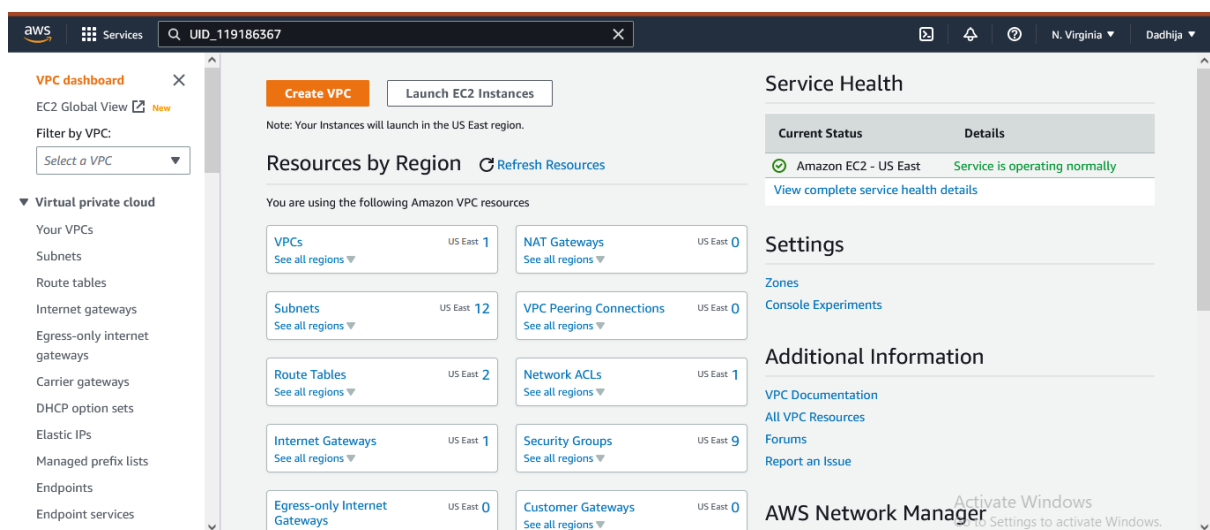
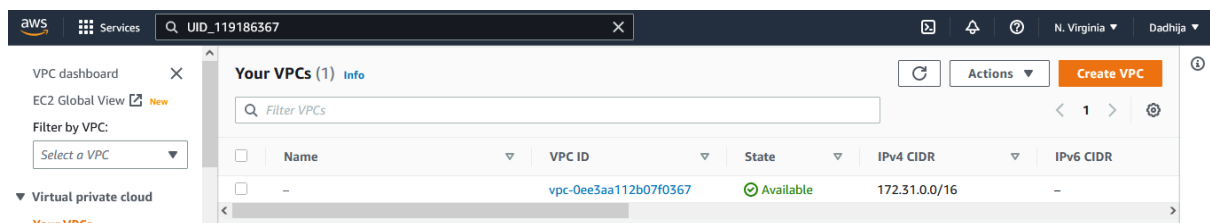
1. Sign into your AWS console.
2. Go to WAF and shield
3. Click on subscribe to AWS shield advanced
4. Select your choice of plan for subscription.
5. Select the resources to protect.
6. Configure shield response team support.
7. Review

## VPC:

VPC is used to create zones in the cloud environment that keep the resources isolated from the rest of the cloud. Different availability zones in VPC ensure that users do not face issues when one of the subnets experiences failures. It can assign public and private subnets to servers as required.

AWS VPC has default subnets as well as custom subnets that users can create according to their requirement.

AWS already consists of a default VPC to use during creation of instances.



## **PCI:**

Cobra kai provides online learning courses. Enrolment in the courses involves some payment from the subscribers. At the moment, Cobra kai does not have a secure method of storing the personal payment card information of customers.

Payment Card Industry Data Security Standard ensures that companies process and store credit card information in a secure environment. Having PCI compliance would make Cobra kai a more trusted website for its users.

AWS cloud platform is PCI compliant. As a result, it stores, processes and transmits sensitive information like credit card information and authentication codes securely. It prevents credit card information theft and gives a secure payment option.

Some AWS services that are PCI compliant are:

1. CloudFront
2. DynamoDB
3. EC2
4. Macie
5. Amazon S3
6. VPC

## References:

1. <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-getting-started.html>
2. <https://www.youtube.com/watch?v=ABtwRb9BFY4>
3. <https://aws.amazon.com/blogs/mt/patching-your-windows-ec2-instances-using-aws-systems-manager-patch-manager/>
4. <https://aws.amazon.com/getting-started/hands-on/create-mysql-db/>
5. <https://docs.aws.amazon.com/AmazonS3/latest/userguide/create-bucket-overview.html>
6. <https://aws.amazon.com/cloudfront/getting-started/S3/>
7. <https://www.reblaze.com/blog/ddos-protection/aws-shield-how-to-set-up-and-use-amazons-ddos-protection-service/>
8. [https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_users\\_create.html#id\\_users\\_create\\_console](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html#id_users_create_console)
9. <https://docs.aws.amazon.com/aws-backup/latest/devguide/getting-started.html>
10. <https://aws.amazon.com/compliance/pci-dss-level-1-faqs/>
11. <https://aws.amazon.com/macie/>
12. <https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-lake.html>
13. <https://github.com/kts262/enpm809j>
14. <https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-manager-how-it-works-release-dates.html>
15. <https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/>
16. <https://www.spiceworks.com/tech/cloud/articles/cloud-migration-strategy/>
17. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>
18. <https://docs.aws.amazon.com/>
19. My midterm submission