Re-architecture and Migration of Cobra Kai

ENPM809J Omar Abdelati UID: 117556827

Plan Objective

This document provides an overview of a plan to migrate Cobra Kai onto the AWS cloud, with hopes of supplying all the necessary high-level knowledge regarding a possible cloud migration. A varying degree of technicality should be expected while going through this proposition. The benefits of each cloud element will be mentioned.

Please keep in mind that the plan was designed with a strict set of criterions, namely:

- Resiliency
- Access Management
- Data Protection
- Compliance
- Secure System Administration and Coding Practices

As a significant part of the overall objective, the criterions will each be met suitably.

The plan is to migrate Cobra Kai into the cloud in five phases. Each phase is outlined below.

Phase 1 – Testing

Test the current software with AWS. The idea here is to attempt a test forklift shift onto the AWS server. Should the software work as intended, or require just a few tweaks to get it working, then the test should guarantee a successful transition into the cloud using the same migration method. This test should take place using a small amount of test data (videos) and a single web-frontend RDS and a single EC2 instance (app server). The test should allow for a more durable environment that is not vulnerable to crashes.

Phase 2 – Data Migration

Once the testing in phase one checks out, a migration of data is due process. Since Cobra Kai is a streaming service with videos to offer, the best option that AWS can provide is an S3 storage system. The S3 storage system is great for quick access to read content and back-up storage. In this very case, Cobra Kai should opt to have both a main S3 bucket to store all of its on-demand content and a copy of the main S3 bucket for back-up that can only be accessed through a private VPN or granted access by specific VPC subnet. There will be two groups (This is to be discussed in further detail later) associated with S3 buckets; one for public (main) and the other for private (back-up).

A more affordable alternative to the S3 back-up bucket, would be Glacier. However, this is not recommended due to its inability to host static web content. S3 is much faster, which can be needed since both the main S3 bucket and the back-up must be accessed frequently enough to update (add newer video content). S3 is more than capable of keeping up with more frequent accesses without slowing down the overall process.

Next is customer data storage. The first option would be to use the Amazon Redshift data warehouse; however, this is might prove to be a little unnecessary. Instead, an EFS storage should be implemented with server-side encryption that can only be accessed through a VPN.

AWS provides the option to encrypt all of the information that is put onto the cloud, regardless of the storage method. Thus, as a necessary precaution, all the information and data will be encrypted; either en-route, prior, or when it reaches the server.

Here Cobra Kai should use the AWS import service. Amazon accepts data from USB 2.0 or eSata storage device that is shipped directly to AWS. AWS will upload the contents of the storage device to the S3 bucket.

Phase 3 – Application Migration

After moving data and organizing it accordingly, a necessary forklift shift of the application comes next. The reason for selecting the forklift shift over the hybrid method is due to the small scale of Cobra Kai's current servers. There should be very little to no issues with simply migrating over to the cloud without any re-structuring or layering of code; once again, phase one's test determines the decision to forklift shift the application.

At this stage, the frontend webservers should switch over to Amazon RDS, while the applications run on EC2 instances.

Phase 4 - Identity Access Management and additional Security Services

Section 1: Root User

Use the "root user" to create a user with all admin privileges. Give the root user and password to either Johnny Lawrence or Miguel Diaz (this depends on how Cobra Kai wants to set up their management hierarchy). A better alternative, should Cobra Kai agree to it, would be to split the username/passcode and MFA (Multi-factor Authentication. Mentioned below) between the two. The reason for creating the user with all admin privileges is for security reasons; for example, if user admin account credentials were compromised, the root user can then disable/delete the admin user.

Section 2: Privileged Cloud Access through IAM

Since the Cobra Kai doesn't have a lot of employees there is no need to leverage IAM groups through the AWS. However, each user/employee should only be given access to exactly what they need and nothing more. In this section of phase 4 Cloud migration a strict least privilege rule should be implemented, where each employee is granted enough access to get by their work and are granted more access, either temporarily through IAM roles, or through their user depending on the situation at hand. The chart below illustrates an overview of the initial setup of IAM in AWS:

Employee Name	**Johnny Lawrence**	**Miguel Diaz**	**Aisha Robinson**	**Hawk**	**Demetri**	**Bert**
Job Description	Founder	COO	CISO	CIO	Head Web Developer	System Administrator
Control IAM	🟩	🟩	🟨	🟥	🟥	🟥
CloudFront Cache	🟥	🟥	🟥	🟩	🟨	🟥
RDS Web Frontend	🟥	🟥	🟥	🟩	🟩	🟥
EC2 Application Server	🟥	🟥	🟥	🟩	🟨	🟩
Customer Information	🟩	🟩	🟩	🟥	🟥	🟥
S3 Bucket Main Content	🟥	🟥	🟩	🟩	🟨	🟥
S3 Bucket Backup Storage	🟥	🟥	🟩	🟨	🟨	🟥

🟩 Green	Grants Access
🟨 Yellow	Can permit access if needed
🟥 Red	Denies Access

Minor Suggestions

The creation of a Junior web developer role in case Hawk decides that Cobra Kai need extra help and hire a temporary developer. This role would only have access to either the EC2 application server or the RDS frontend.

The creation of IAM groups and security groups is now possible and suggested as Cobra Kai grows in size.

Section 3: Logging Services

For additional security S3 logging, CloudTrail, and CloudWatch can be leveraged from AWS's extra services.

- S3 Logging: Manages and monitors all requests to access any S3 buckets.
- CloudTrail: Monitors all user logins along with their details (IP address, time of the day, what content the user accessed)

- CloudWatch: Monitor EC2 applications and system wide performance. This is highly recommended for Aisha Robinson.

Section 4: Key Management System

Setting up a key management system through AWS is not mandatory due to Cobra Kai's size, however, it could prove helpful as Cobra Kai grows.

Section 5: Multi-Factor Authentication

Having MFA for each user and/or API key is mandatory for security.

Phase 5 – Optimizations and Security

Section 1: Security Implementations

Below are four steps that should be implemented for security purposes.

1- Planning: Security Policies and requirements. Establish guidelines for secure coding to prevent any unwanted segmentation faults, infinite loops, crashes. Plan out a sandbox instance for testing.

2- Create: Creation of any additional code should all be attempted within the sandbox EC2 or RDS instance before pushed onto any of the main servers or buckets.

3- Verify: Review the code and attempt to break. (This segment should be overseen by Aisha)

4- Optimize: Complete any optimal changes to the code run it through with the system administrator.

Section 2: Secure API Gateways

As part of the migration process securing any and all API gateways is mandatory for Cobra Kai's security. All API keys should be encrypted.

Section 3: AWS Virtual Private Cloud + NACL (Network Access Control List)

AWS VPC allows Cobra Kai to place a clear distinction between public and private subnets, adding an additional layer of security to the overall infrastructure. NACLs should also be implemented as a means to manage the inbound and outbound traffic of one or more subnets. ACLs and their rules should be configured depending on the service used. ACLs act as a firewall for the subnet that encompasses the service within the VPC.

Section 4: Sandbox Instance

As mentioned in section one, creating a sandbox instance is a necessary move towards Cobra Kai's growth. Duplicate the instance of the application and frontend webserver in attempt to re-engineer and improve them.

Section 5: AWS Network Firewall

Using the network firewall, Cobra Kai can place a permanent block on certain IP address that are known to be malicious.

Copyright Laws and Regulation Compliance

Cobra Kai should copyright any and all of its products. This includes, but is not limited to, any code produced by Cobra Kai and all of the episodes recorded for Cobra Kai. Cobra kai should only migrate licensed products to avoid facing copyright infringement cases.

Cobra Kai should trademark their Logo and brand name.

The Payment Card Industry (PCI) requires compliance with their Data Security Standard (DSS) for any entity that stores, processes, or transmit their customer's card data. As it stands, Cobra Kai will charge customers in exchange for its content; therefore, getting started with AWS Artifact is a must. A link to AWS Artifact can be found here: https://aws.amazon.com/artifact/getting-started/ A list of all the AWS services that are compliant with PCI DSS can be found here: https://aws.amazon.com/compliance/services-in-scope/. All of the services mentioned in this document comply with AWS PCI DSS.

Website Architecture

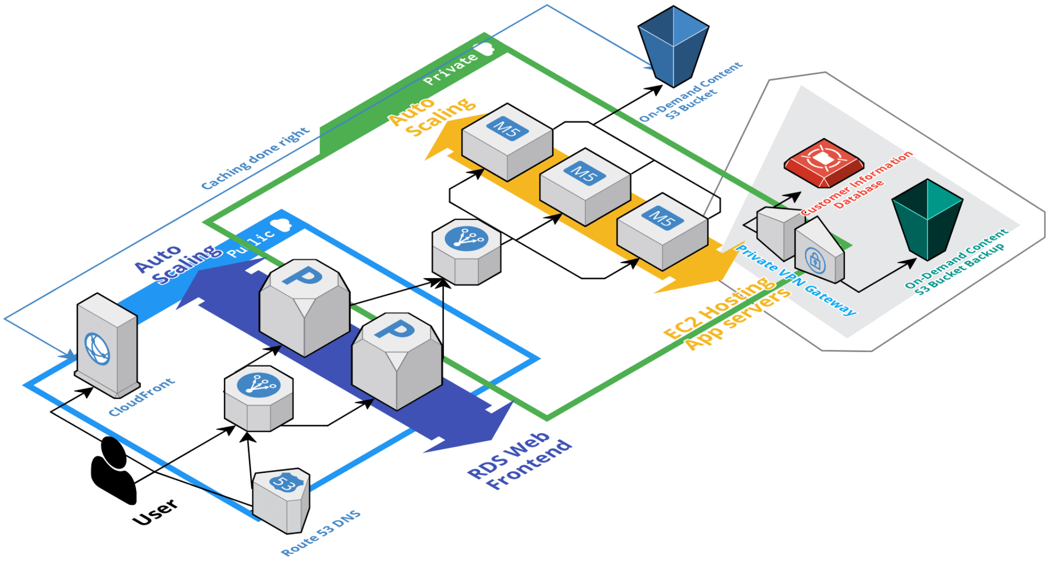

Figure 1 - Website Architecture. Created using Cloudcraft.co

Figure 1 displays the proposed website architecture. CloudFront should serve to provide Cobra Kai's users with a closer cached server, this, along with AWS S3, should speed up the streaming process for the viewers. Route S3 DNS is also added for newcomers, allowing them to connect to the frontend a lot faster than using any other DNS server. The Cloud will take on a similar website infrastructure that Cobra Kai currently has with slight adaptations to the new cloud environment.

Load Balancers: Serve to balance the network traffic between servers (for both frontend and EC2). Should also help prevent DDOS attempts.

VPC: A public and a private subnet adds an additional layer of security. In this vary case the load balancer and RDS frontend webservers are public, while the second load balancer, EC2 instances, and the main S3 bucket (blue bucket at the top) operate on a private subnet with routes that can only be found from the web server's network device.

In the right corner of the figure there exists a very private VPN gateway that shields the customer's information held in AWS EFS (Elastic File System) and the backup S3 bucket. This extra layer of security is deemed necessary after evaluating the importance of such data. Only privileged users can gain access to this area as depicted in the matrix. Both storage systems, S3 and EFS, are scalable.

Finally, AWS S3, EC2, EFS and RDS all provide flexibility and scalability. When Cobra Kai grows, your servers can grow with you! Without all too much effort an extra server here and there can be added with little effort using this architecture.

Future Possibilities

IAM Groups

As Cobra Kai grows in size, so will the number of its employees. Creating dedicated roles per employee will become tiring and difficult to automate. By shifting to groups, Cobra Kai can customize certain privileges within each group. For example, EC2 group members have access to the backend EC2 instances within the private VPC and access to nothing else unless they are a part of another group. In that sense, groups can be created depending on the AWS service, and even then, can be further customized.

Security Groups for VPC

Once more, this future possibility is concerned with the growth of the company and the likelihood of a greater number of employees. Security groups helps implement a layer of protection more efficiently and effectively for AWS instances. Up to five security groups can be created per instance. Each security group can be customized with rules that dictate the inbound and outbound traffic. After making certain security groups, system administrators can just place employees in their designated security group for each instance they work on.

Additional Roles for Temporary Employees and Contract Hires

AWS IAM Roles can be used to create a temporary position for employees working during a short-term. These roles evade having to create a new user and generate a new API every time a temporary worker or a short contract-bound employee is hired. Automating this process reduces the chances of giving more privilege to a new user on accident.

AWS Redshift Database for Data Collection and Analysis

A RedShift DB can be used as a business-intelligence segment of Cobra Kai. RedShift can be used for data mining, processing, and analysis of customer information, product sales, costs, etc. With the abundance of information collected and based on certain observations after analysis, Cobra Kai can make better suited business decisions to help develop the company and expand.