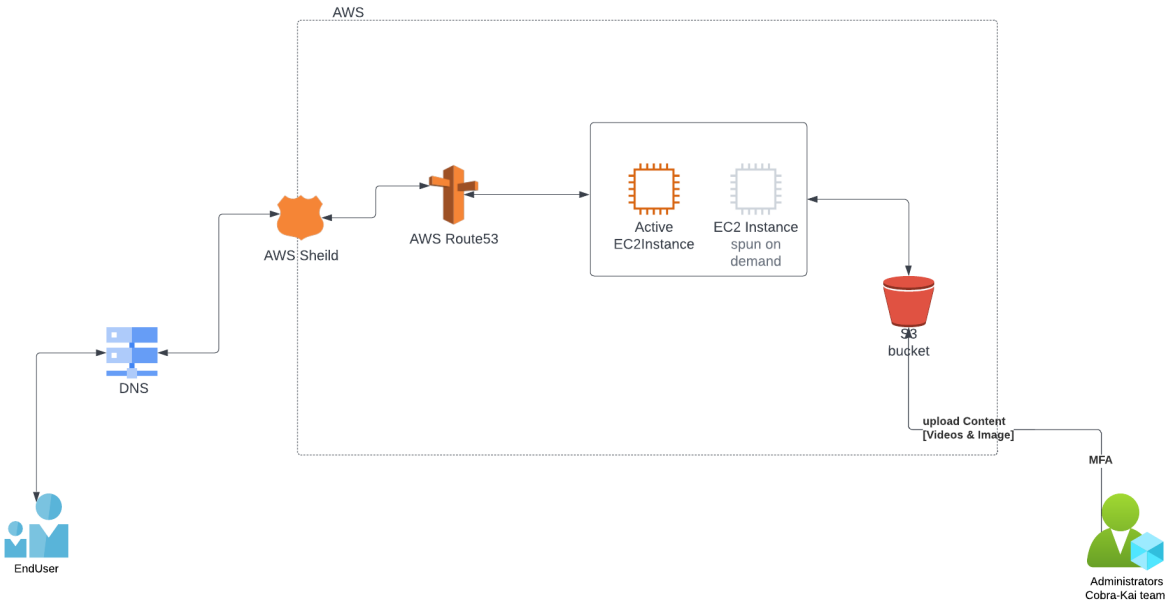
ENPM665- MIDTERM

The practice of transferring to the cloud has become increasingly important as more companies have switched from building and managing their own data centers to storing their applications and data in the cloud instead. A 2018 survey by IDG[International Data Group], a leading technology media company, stated that 73% of companies had applications or infrastructure in the cloud, with another 17% expected to make the move in the coming year.

I would like to strongly recommend moving the entire infrastructure to the cloud. I would like to propose an AWS based solution. Six important points are covered in order to convince Cobra Kai team to completely transfer to AWS.

1) Resiliency
2) IAM
3) Protecting data
4) Compliance
5) Against ddos attacks
6) Best secure practices

The following is diagram shows the detailed



Resilient systems:

A resilient system is one that continues to function despite failures of system components. We have many ways to achieve a resilient system, but the easiest way is to add redundancy to the application. The infrastructure layer (the cloud servers) is assumed to be prone to failure, and so redundancy is engineered into the architecture of the application. Some of the largest cloud vendors around operate in this fashion: they provide the infrastructure, you build the redundant applications.

Recommended AWS solution:

The most basic solution is a cloud-based architecture pattern that introduces Availability Zones (AZs) into the architecture to increase your system's resilience. The P1 pattern uses a Multi-AZ architecture where applications operate in multiple AZs within a single AWS Region. This allows your application to withstand AZ-level impacts.

cons:

P1 is low effort in several categories, but this comes at the expense of application recovery. If AZ is down, it will disrupt end users' access to the application while the new resources are being re-provisioned in a new AZ

Recommendation2:

P3 – Application portfolio distribution

The P3 pattern uses a multi-Region pattern to increase functional resilience. It distributes different critical applications in multiple Regions.

Let's assume a website requires little to no downtime because of its high volume traffic. Any time the website is down, it could result in lost revenue.These services are available to consumers via a mobile application or web-based applications. If the Region fails where the mobile application is deployed, customers can still access services via the other channels deployed in other Regions. Regional disruptions are rare, but implementing this pattern ensures your users retain access to business-critical services during disruptions. So even if the users can not access the videos through a web application, we could build a mobile application for Cobra kai that could be used without any disruption.

cons:

Operating an application portfolio that spans multiple Regions requires significant operational planning and management. Isolated functional elements may depend on common downstream systems and data sources that are deployed in a single Region. Therefore, Region-wide events might still cause disruption; however, the impact surface area is significantly reduced.

Cost factor:

Auto Scaling is enabled by Amazon CloudWatch and carries no additional fees. Each instance launched by Auto Scaling is automatically enabled for monitoring and the applicable Amazon Cloudwatch charges will be applied. CloudWatch is used to collect and track metrics

You can get started with Amazon CloudWatch for free. Most AWS Services (EC2, S3, Kinesis, etc.) send metrics automatically for free to CloudWatch. Many applications should be able to operate within these free tier limits.

The cost of S3 bucket (standard) is 0.023/GB for the first 50TB and 0.0.23$/GB for the next 450GB for each month. This should be sufficient for initial stages of the development of Cobra Kai which can be later scaled up accordingly.

IAM:

Cloud identity and access management (cloud IAM) is a security framework deployed in the cloud used to verify users and control their access rights, including issuing and denying access privileges

AWS solution:

Principal:

A principal is a person or application that can make a request for an action or operation on an AWS resource. The principal is authenticated as the AWS account root user or an IAM entity to make requests to AWS

The Identity and Access management for principal multi-factor authentication. In multi-factor authentication, two or more pieces of evidence must be presented to an authentication mechanism in order to gain access to a website or application.

AWS Identity and Access Management (IAM) helps you define what a principal entity is allowed to do in an account. Access management is often referred to as authorization. You manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS resources.

A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal uses an IAM entity (user or role) to make a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents.

If you manage permissions across multiple accounts, it is more difficult to manage permissions for your users. You can use IAM roles, resource-based policies, or access control lists (ACLs) for cross-account permissions.

You can organize IAM users into IAM groups and attach a policy to a group. In that case, individual users still have their own credentials, but all the users in a group have the permissions that are attached to the group. Use groups for easier permissions management

Cost factor:

IAM administrators control who can be authenticated (signed in) and authorized (have permissions) to use AWS Cost Management resources. IAM is an AWS service that you can use with no additional charge.

Protecting data:

In the cloud environment, data protection means securing a company's data wherever it is located, whether it is at rest or in motion, whether it is managed internally or externally. Protecting the data is a mandatory requirement since it helps the customers to gain more trust while shifting to the cloud.

Among the benefits of cloud data protection, it enables companies to:

- Secure applications and data across multiple environments while maintaining complete visibility into all user, folder and file activity.
- Better govern access.
- Define policies.
- Prevent and detect data loss and disruption.

Proposed AWS solution:

Services such as AWS Identity and Access Management (IAM) allow you to securely manage access to AWS services and resources. AWS CloudTrail and Amazon Macie enable compliance, detection, and auditing, while AWS CloudHSM and AWS Key Management Service (KMS) allow you to securely generate and manage encryption keys. AWS Control Tower provides governance and controls for data residency.

You can improve your ability to meet core security, confidentiality, and compliance requirements with our comprehensive services, whether that's through Amazon GuardDuty or our AWS Nitro System, the underlying platform for our EC2 instances. We've designed the Nitro System to have workload confidentiality and no operator access. With the Nitro System, there's no mechanism for any system or person to log in to EC2 servers, read the memory of EC2 instances, or access any data stored on instance storage and encrypted EBS volumes.

Cost factor:

A one-time charge of $0.033, which includes $0.009 for AWS Config to initially record 3 configuration items, at the rate of $0.003 per configuration item, and $0.002 for AWS Config to evaluate 2 rules, at the rate of $0.001 per evaluation (for the first 100,000 evaluations).

Compliance:

AWS continuously raises the bar on privacy safeguards with services and features that let you implement your own privacy controls, including advanced access, encryption, and logging features. We make it easy to encrypt data in transit and at rest using keys either managed by AWS or fully managed by you. You can bring your own keys that were generated and managed outside of AWS. We implement consistent and scalable processes to manage privacy, including how data is collected, used, accessed, stored, and deleted.

We only process customer data - that is any personal data you upload to your AWS account - under your documented instructions and do not access, use, or share your data without your agreement, except as required to prevent fraud and abuse, or to comply with law, as described in our AWS Customer Agreement and AWS GDPR Data Processing Addendum. Thousands of customers who are subject to GDPR, PCI, and HIPAA use AWS services for these types of workloads. AWS has achieved numerous internationally-recognized certifications and accreditations, demonstrating compliance with rigorous international standards, such as ISO 27017 for cloud security, ISO 27701 for privacy information management, and ISO 27018 for cloud privacy. We do not use customer data or derive information from it for marketing or advertising purposes.

Protection against DDOS attack:

To understand how to protect against DDoS attacks, let us see the most common types of attacks. There are three common types of DDoS attacks:

- Application-Layer Attacks consist of well-formed but malicious requests (HTTP GETs and DNS queries are popular) that are designed to consume application resources. For example, opening up multiple HTTP connections and reading the responses over the course of many seconds or minutes will consume excessive memory and prevent legitimate requests from being serviced.
- State-Exhaustion Attacks abuse stateful protocols and causes stress on firewalls and load balancers by consuming large numbers of per-connection resources.
- Volumetric Attacks disrupt networks by flooding them with more traffic than they can handle or by issuing fake queries that will flood an unsuspecting victim with a surprising amount of low-level “surprise” replies (also known as Reflection attacks).

AWS has preventive measures against DDOS attacks.

Route 53 is hosted at numerous AWS edge locations, creating a global surface area capable of absorbing large amounts of DNS traffic. Other edge-based services, including Amazon CloudFront and AWS Web Application Firewall, also have a global surface area and are also able to handle large amounts of traffic

Each edge location has many connections to the Internet. This allows for diverse paths and helps to isolate and contain faults. Route 53 also uses shuffle sharding and anycast striping to increase availability. With shuffle sharding, each name server in your delegation set corresponds to a unique set of edge locations. This arrangement increases fault tolerance and minimizes overlap between AWS customers. If one name server in the delegation set is not available, the client system or application will simply retry and receive a response from a name server at a different edge location

AWS Shield Standard is available to all AWS customers at no extra cost. It protects you from 96% of the most common attacks today, including SYN/ACK floods, Reflection attacks, and HTTP slow reads. This protection is applied automatically and transparently to your Elastic Load Balancers, CloudFront distributions, and Route 53 resources.

AWS Shield Advanced provides additional DDoS mitigation capability for volumetric attacks, intelligent attack detection, and mitigation for attacks at the application & network layers. You get 24×7 access to our DDoS Response Team (DRT) for custom mitigation during attacks,

advanced real time metrics and reports, and DDoS cost protection to guard against bill spikes in the aftermath of a DDoS attack.

Cost factor:

The AWS shield standard comes at no cost in aws free tier. And the AWS Route53 charges No fees for storage of up to 1,000 IP (CIDR) blocks which should be good to go for initial stages and can scale up if needed.

Best secure practices:

- ***Implement least privilege access***
    - When granting permissions, you decide who is getting what permissions to which Systems Manager resources. You allow specific actions that you want to allow on those resources. Therefore you should grant only the permissions that are required to perform a task. Implementing least privilege access is fundamental in reducing security risk and the impact that could result from errors or malicious intent.
- ***Use SecureString parameters to encrypt and protect secret data***
    - In Parameter Store, a capability of AWS Systems Manager, a SecureString parameter is any sensitive data that needs to be stored and referenced in a secure manner. If you have data that you don't want users to alter or reference in plaintext, such as passwords or license keys, create those parameters using the SecureString data type. Parameter Store uses an AWS KMS key in AWS Key Management Service (AWS KMS) to encrypt the parameter value.
- ***Define allowedValues and allowedPattern for document parameters***
    - You can validate user input for parameters in Systems Manager documents (SSM documents) by defining allowedValues and allowedPattern. For allowedValues, you define an array of values allowed for the parameter. If a user inputs a value that isn't allowed, the execution fails to start.
- ***Block public sharing for documents***
    - Unless your use case requires public sharing to be allowed, we recommend turning on the block public sharing setting for your SSM documents in the Preferences section of the Systems Manager Documents console.

Overall converting to AWS should be a very cost effective solution to achieve better quality of service and secure way to access the content for the end users.

References:

Reduce DDoS Risks Using Amazon Route 53 and AWS Shield
AWS Shield – Protect your Applications from DDoS Attacks
https://docs.aws.amazon.com/systems-manager/latest/userguide/security-best-practices.html
https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html
https://aws.amazon.com/compliance/programs/
https://docs.aws.amazon.com/IAM/latest/UserGuide/intro-structure.html
https://aws.amazon.com/blogs/architecture/understand-resiliency-patterns-and-trade-offs-to-architect-efficiently-in-the-cloud/