ENPM665: MIDTERM

Submitted By: -
Ratan Gupta
rgut21@umd.edu
UID: 118195773

COBRA KAI
STRIKE FIRST – STRIKE HARD – NO MERCY

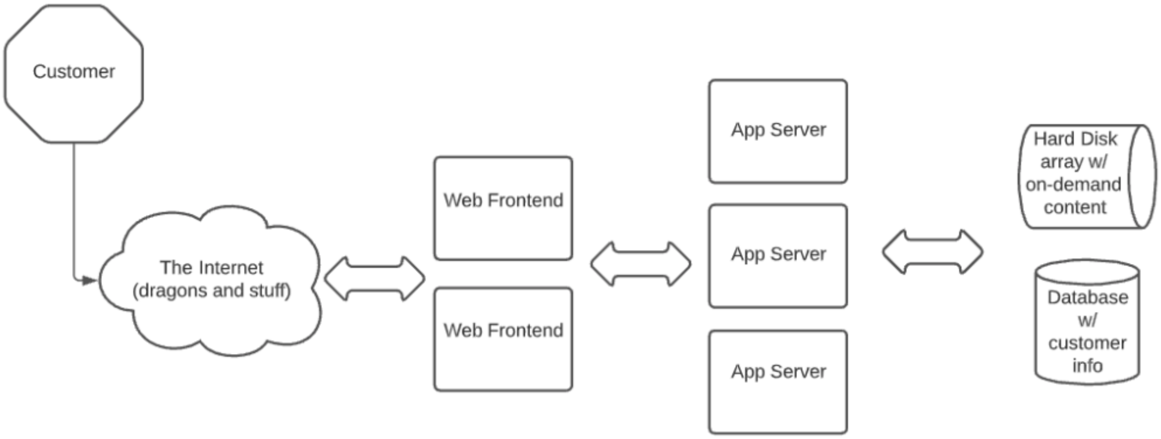
INTRODUCTION

This document serves as a short executive summary that provides recommendations on rearchitecting the Cobra Kai application and migrating it to the cloud. In this document, concepts and strategies to leverage the advantages of the cloud environment are discussed. Furthermore, concepts like identity and access management (IAM), resiliency, compliance, secure coding practices, secure system administration, and data protection are talked about to include security of the users and application.

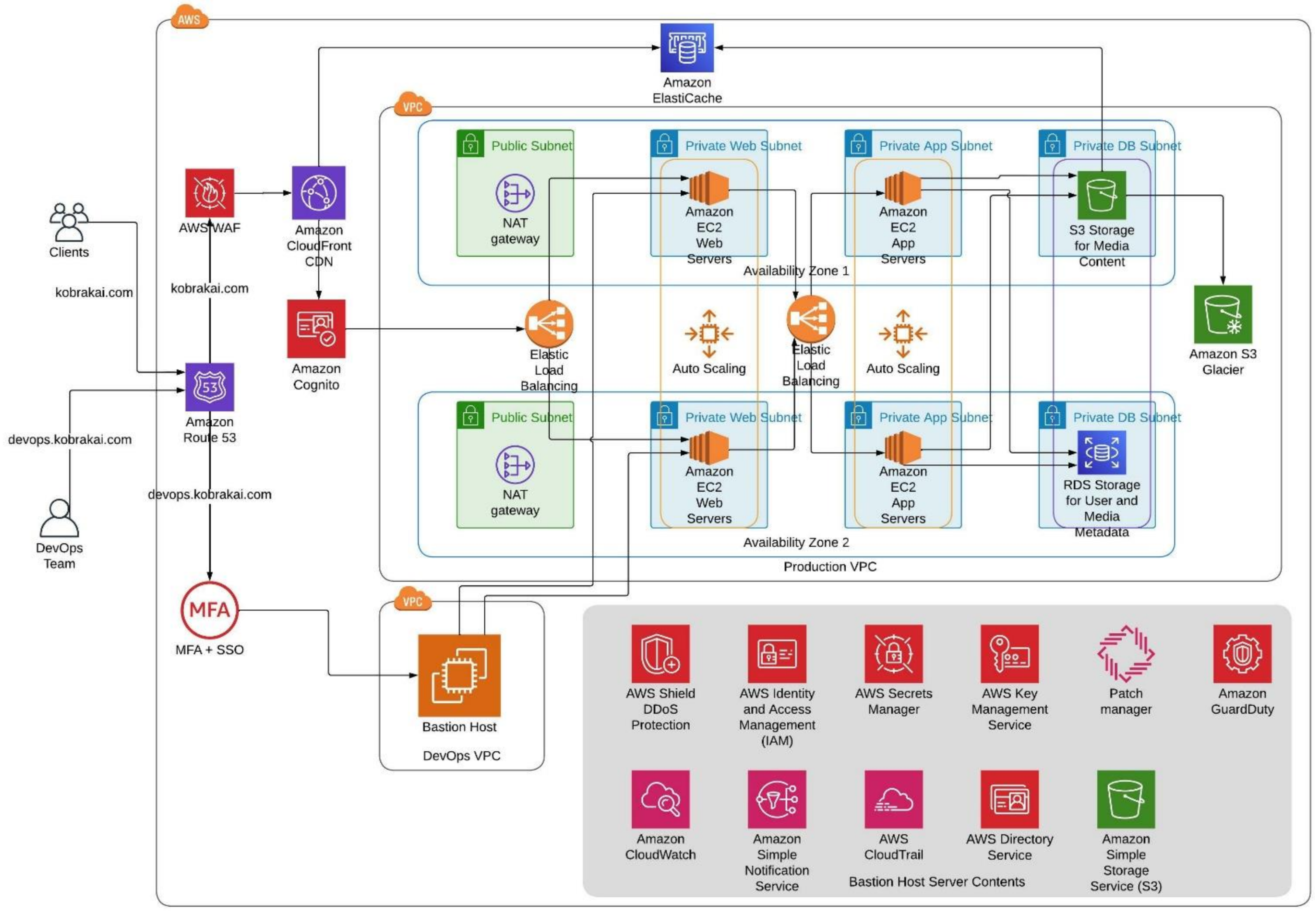
CURRENT ISSUES TO CONSIDER

- Cobra Kai does not currently have a patching strategy
- Cobra Kai does not currently have a backup strategy
- Cobra Kai does not currently have an account permission strategy; every user has the ability to run privileged commands on the web server if they want to
- Their entire website infrastructure is highly vulnerable to DDoS, hardware failures, and human error. It runs in a closet for crying out loud.
- The website has experienced DDoS attacks and compromise attempts they suspect comes from a rival dojo ran by Daniel LaRusso who with his deep pockets has become a persistent threat against Cobra Kai's l IT operations
- Customers have complained about slow streaming, downloads, and order processing
- Cobra Kai's platform is processing credit card data and stores customer PII (name, phone, email, address, and additional details about the customer)
- Cobra Kai's corporate IP range is 129.2.0.0/16 (it's not really but pretend it is).

CURRECT WEB ARCHITECTURE


Customer
The Internet (dragons and stuff)
Web Frontend
Web Frontend
App Server
App Server
App Server
Hard Disk array w/ on-demand content
Database w/ customer info

PROPOSED CLOUD ARCHITECTURE



PROPOSED DATA FLOW IN THE ARCHITECTURE

User Interaction with the Cloud

- Users will access the cloud application using the domain 'cobrakai.com'. All the HTTPS requests sent to this domain will be routed through Amazon Route 53. Amazon Route 53 is a scalable DNS service used by Amazon to connect user requests to applications hosted on AWS Cloud [1].
- Traffic will then flow through AWS WAF (Web Application Firewall). AWS WAF is a web application firewall that helps protect web applications or APIs against common web exploits that may affect availability, compromise security, or consume excessive resources. AWS WAF helps in blocking common attack patterns, SQL injections, cross-site scripting. AWS WAF can also block some of the traffic based on geolocation as well as IP addresses [2].
- The request will then be passed to Amazon CloudFront, which is Amazon's content delivery network. CDN reduces the latency of a request. Based on the request, the traffic will either be routed to Amazon ElastiCache, which is a managed in-memory data store and cache service [3], or to the EC2 instances running web and application servers.
- If the user needs to be authenticated before accessing the servers, the request will be forwarded to Amazon Cognito, which is a User Identity and Management service. It maintains the identity of all users and the level of access they are authorized to have.
- An Elastic Load Balancer will distribute incoming application traffic through NAT Gateways in public subnets and across multiple EC2 instances (web servers) in private subnets, existing in different Availability Zones.
  Multiple Availability Zones increase the availability of applications. Elastic Load Balancing helps in balancing the load between multiple servers. This increases speed and performance. It scales the load balancer capacity automatically depending on the changes in incoming traffic and monitors health of the registered targets to avoid unhealthy targets [4].
- To fulfill the request the Web Servers will send in a request to the App Servers hosting the web application through a load balancer. The Web and App Servers can scale based on the incoming traffic.
- The live streaming content and on-demand videos will be stored in Amazon Simple Storage Service (S3) bucket, which is a scalable and secure database. Frequently requested media will be cached in Amazon ElastiCache which was introduced previously. The live streaming and on-demand videos wilk be backed up in a long-term, durable, and secure backup service; Amazon S3 Glacier [5].
- The user and media metadata will be stored in Amazon Relational Database Service (RDS).

Administrative Management of the Cloud

- System admins and developers will login using Multi-Factor Authentication (MFA) and Single Sign On (SSO) services. Different access rules will be specified for the admins and developers using AWS Directory Service by creating Groups and Group Policies. This will enable admins and developers to have access to only those resources that they need to manage and nothing else.

- System admins and developers will access their AWS Console using the 'aws.cobrakai.com' domain.
- A Bastion Host will be set up in a DevOps VPC for the admins and developers to manage the application and will enable them to securely access web and app servers running in the Production VPC with their level of privilege.
- The Bastion Host is a Linux server running on an EC2 instance. Furthermore, the server will contain a myriad of AWS services required for compliance, security, resilience, etc.
- Activities, audits, operations, metrics, etc. will be logged in an S3 bucket.

Logging and Monitoring Recommendations

For logging and monitoring, Amazon CloudTrail, CloudWatch, and GuardDuty services will be used.

- Amazon CloudTrail enables governance, compliance, operational auditing, and risk auditing of the AWS account. With CloudTrail, we can log, continuously monitor, and retain account activity related to actions across the AWS infrastructure. CloudTrail provides event history of our AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services. All actions taken from any AWS user will be logged using CloudTrail. [6]
- Amazon CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, and provides a unified view of AWS resources, applications, and services that run on AWS and on-premises servers. The CloudWatch service will be used to create alarms on metrics and insights. CloudWatch can be used on all the servers to track all the audit logs. [7]
- Amazon GuardDuty is a threat detection service that continuously monitors malicious activity and unauthorized behavior to protect AWS accounts, workloads, and data stored in Amazon S3. GuardDuty will be used to monitor network traffic in this design. AWS GuardDuty sends alerts if users' IAM credentials or IAM roles have been compromised. It detects any unusual activities performed by users. GuardDuty also detects if any server has an open port and if someone is trying to reach that port from unusual locations or IP addresses. [8]

Other AWS services used

The following AWS services will be used for identity and access management (IAM), resiliency, compliance, secure coding practices, secure system administration, and data protection.

- Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication. In this design, we will leverage Amazon SNS to send out email alerts for any unusual activities. [11]

RECCOMMENDATIONS FOR CURRENT CONCERNS

Patching Strategy Recommendation

AWS Patch Manager will be used to patch the EC2 instances as well as the application. We can automate the process of patching using AWS Patch Manager so we can schedule the patching of the servers in the time when we don't expect any users to use the application. AWS Patch manager will be used to patch instances monthly. The Patch manager will automate the patching of managed instances with operating system updates as well as security and application updates. [9]

Backup Strategy Recommendation

Amazon S3 Glacier will provide a long-term, durable, and secure backup of the entire media content. Data can be stored for as long as decades using this service. Depending on the need, three different archive storage classes can be used. [5]

Account Permission Strategy Recommendation

For user access management the web application uses AWS Cognito which provides controlled access to the web application. Cognito has an identity store where it stores all the user identities which can scale on-demand. Cognito can be configured to send verification notifications or emails and can be used to provide authentication even with SAML or other identity providers.

Cognito can define permissions for the users and tokens can be used to interact with the EC2 Instances. This would ensure that no user is able to perform a function that they are not authorized for. User permissions are controlled using IAM defined roles and can be configured as per requirements. This setup will also hamper illegal access to the platform from the Daniel LaRusso dojo.

AWS Identity and Access Management will be used to assign roles to users and manage users that have access to the AWS Account. Only Johnny, Miguel, Aisha, and Hawk will have administrative access over the cloud environment.

AWS Secrets manager will store all the security keys used by the developers or the management staff. The secrets manager will also store all the keys that will be required for interconnection of services and for the codebase of the web application. It is necessary to make sure none of the security keys are hard coded into the web application codebase, they should be requested from the AWS Secrets Manager.

Strategy Recommendation to mitigate DDoS, Hardware Failures and Human Errors

AWS Web Application Firewall, Elastic Load Balancer and AWS Shield are used to prevent these errors. AWS shield which is being used with Elastic Load Balancer defends against most common, frequently occurring network and transport layer DDoS attacks that target website or applications. The advantage of Using AWS ELB is even if someone successfully performs DDoS attacks on the application, the Load Balancer will be able to auto scale servers as per demand so end user will not face any downtime.

Strategy Recommendation to mitigate Daniel LaRusso's DDoS attacks

Using AWS WAF, traffic coming from certain geolocations or from specific IP addresses can be blocked. So, on the Firewall we can block LaRusso's IP address which will render his attacks useless. In this solution, web servers reside in private subnet and behind Elastic Load Balancer. An attacker won't be able to reach the servers if we are using load balancer because load balancer will be the only thing that is publicly available. Attackers can't perform attacks on load balancer.

Strategy Recommendation to mitigate slow streaming, downloads, and order processing

The proposed solution uses AWS CloudFront and S3 bucket to mitigate this issue. AWS CloudFront is a fast Content Delivery Network (CDN). It will help deliver all media and application data and with low latency and high transfer rates. CloudFront will point to ElastiCache with static content and frequently requested data. This will resolve the issue of slow downloads, streaming and order processing.

PII and Credit Card Data Security Recommendations

Amazon Web Services (AWS) is certified as a PCI DSS Level 1 Service Provider. All the services that we are using such as Amazon S3, RDS, EC2, etc. are all PCI DSS Compliant.

The Payment Card Industry Data Security Standard (PCI DSS) is an information security standard merchants and service providers must comply with if they store, process, or transmit cardholder data. PCI DSS includes over 400 information security requirements, including requirements that apply to cloud infrastructure such as Amazon Web Services (AWS). [12]

With this proposed architecture, databases reside in private subnets which makes it publicly inaccessible. Database will only be connected to the app servers and all the connections will be blocked using Default Security Groups. Only App server IPs will be whitelisted using Security Groups and no other IPs. Databases will be monitored using CloudWatch for auditing and logging so we will be alerted if any non-privileged users run queries on the database or launch an attack.

Amazon Key Management Service will be used to encrypt all the data as rest. All the data will be encrypted using different KMS keys. KMS is also a single point service which keeps track of all the places the key has been used to decrypt data so this can help in security as well. [10]

REFERENCES

[1] https://aws.amazon.com/route53/

[2] https://www.nclouds.com/blog/security-apps-aws-web-application-firewall/

[3] https://en.wikipedia.org/wiki/Amazon_ElastiCache

[4] https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html

[5] https://docs.aws.amazon.com/amazonglacier/latest/dev/introduction.html

[6] https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html

[7] https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch_architecture.html

[8] https://aws.amazon.com/guardduty/

[9] https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-patch.html

[10] https://aws.amazon.com/blogs/aws/new-key-management-service/

[11] https://aws.amazon.com/sns/

[12] https://kirkpatrickprice.com/blog/aws-pci-compliance/