

Final Migration Report



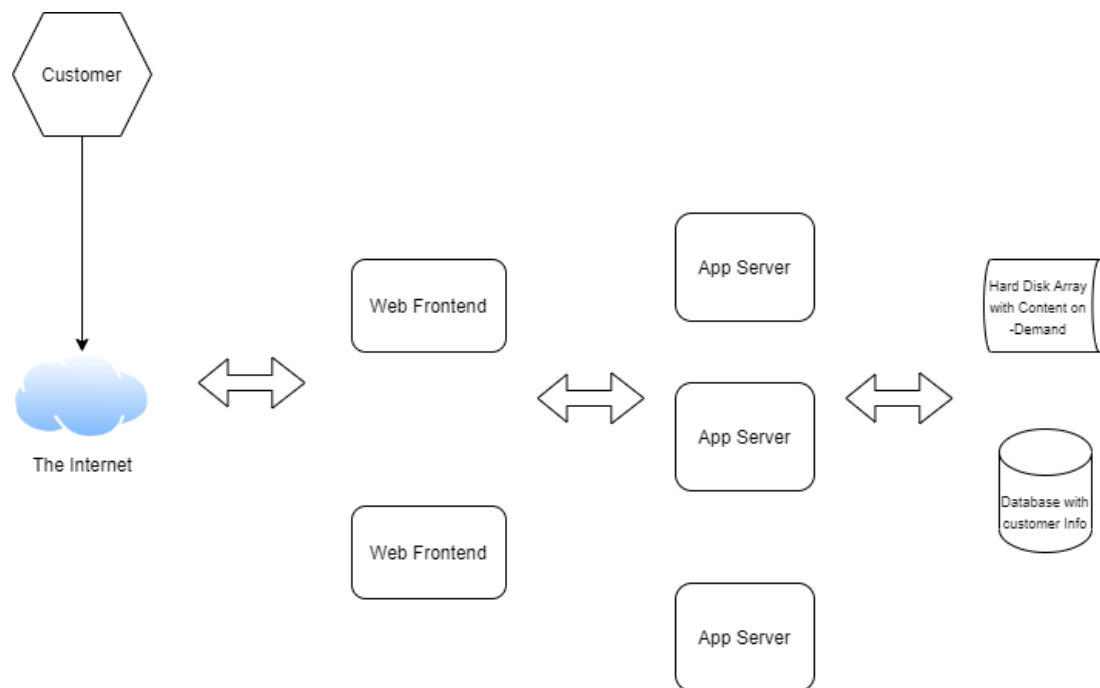
A. JAMES CLARK
SCHOOL OF ENGINEERING
ENPM809J Cloud Security Fall 2020

Student Name: Shoumit Karnik
University ID: 116717496

EXECUTIVE SUMMARY

In this document the implementation plan, which will be adopted to redesign the Cobra Kai web streaming service to take advantage of the benefits of moving to the cloud, has been outlined. Some basic implementation steps for a rudimentary cloud structure and some technical examples of how the cloud security posture would look like have been documented below.

CURRENT SYSTEM ARCHITECTURE



PROPOSED SOLUTION ROADMAP

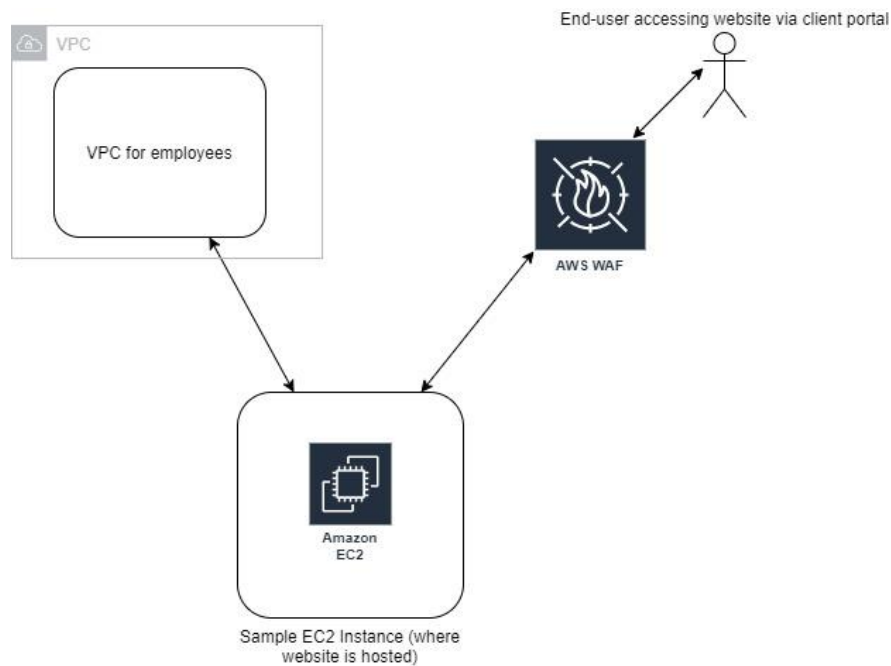
Some solution details have been provided subsequently. They include:

Sample Security Group configurations:

This part includes security group configurations, along with creating a VPC (Virtual Private Cloud) subnet for the employees.

The EC2 instance where the website will be migrated to will also be protected from external attacks using a Web Application Firewall (AWS WAF). This specifically takes care of OWASP top 10 web attacks like XSS(Cross Site Scripting), XSRF(Cross-Site Request Forgery) etc. (Because the website manages credit card information and has been known to be vulnerable to SQL injection)

The EC2 instance will also have iptables (internal firewall) running on it, to provide an internal layer of security for the instance.



For AWS WAF settings :

A set of Web ACL rules will be used to setup AWS WAF.

aws Services Search for services, features, marketplace products, and docs [Alt+S]

AWS WAF > Web ACLs > Create web ACL

Step 1 Describe web ACL and associate it to AWS resources

Step 2 Add rules and rule groups

Step 3 Set rule priority

Step 4 Configure metrics

Step 5 Review and create web ACL

Describe web ACL and associate it to AWS resources

Web ACL details

Name: ENPM809F_final

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Description - optional: Firewall facing the internet to protect from web based attacks

The description can have 1-256 characters.

CloudWatch metric name: ENPM809F_final

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

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

☐ CloudFront distributions

☒ Regional resources (Application Load Balancer, API Gateway, AWS AppSync)

The following rules will be selected to configure AWS WAF. Their priority can also be adjusted based on the security needs.

aws

Services

Search for services, features, marketplace products, and docs

[Alt+S]

Sho

Add rules and rule groups: Add managed rule groups

Step 3

Set rule priority

Step 4

Configure metrics

Step 5

Review and create web ACL

Name	Capacity	Action
Admin protection Contains rules that allow you to block external access to exposed admin pages. This may be useful if you are running third-party software or would like to reduce the risk of a malicious actor gaining administrative access to your application.	100	<input checked="" type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
Amazon IP reputation list This group contains rules that are based on Amazon threat intelligence. This is useful if you would like to block sources associated with bots or other threats.	25	<input checked="" type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
Anonymous IP list This group contains rules that allow you to block requests from services that allow obfuscation of viewer identity. This can include request originating from VPN, proxies, Tor nodes, and hosting providers. This is useful if you want to filter out viewers that may be trying to hide their identity from your application.	50	<input checked="" type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
Core rule set Contains rules that are generally applicable to web applications. This provides protection against exploitation of a wide range of vulnerabilities, including those described in OWASP publications.	700	<input checked="" type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
Known bad inputs Contains rules that allow you to block request patterns that are known to be invalid and are associated with exploitation or discovery of vulnerabilities. This can help reduce the risk of a malicious actor discovering a vulnerable application.	200	<input checked="" type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
Linux operating system Contains rules that block request patterns associated with exploitation of vulnerabilities specific to Linux, including LFI attacks. This can help prevent attacks that expose file contents or execute code for which the attacker should not have had access.	200	<input checked="" type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count

aws Services ▾

🔍 Search for services, features, marketplace products, and docs [Alt+S]

🔔 Shoumit Karnik ▾

☰

Known bad inputs Contains rules that allow you to block request patterns that are known to be invalid and are associated with exploitation or discovery of vulnerabilities. This can help reduce the risk of a malicious actor discovering a vulnerable application.	200	<input checked="" type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
Linux operating system Contains rules that block request patterns associated with exploitation of vulnerabilities specific to Linux, including LFI attacks. This can help prevent attacks that expose file contents or execute code for which the attacker should not have had access.	200	<input checked="" type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
PHP application Contains rules that block request patterns associated with exploiting vulnerabilities specific to the use of the PHP, including injection of unsafe PHP functions. This can help prevent exploits that allow an attacker to remotely execute code or commands.	100	<input checked="" type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
POSIX operating system Contains rules that block request patterns associated with exploiting vulnerabilities specific to POSIX/POSIX-like OS, including LFI attacks. This can help prevent attacks that expose file contents or execute code for which access should not been allowed.	100	<input type="radio"/> Add to web ACL
SQL database Contains rules that allow you to block request patterns associated with exploitation of SQL databases, like SQL injection attacks. This can help prevent remote injection of unauthorized queries.	200	<input checked="" type="radio"/> Add to web ACL <input type="radio"/> Set rules action to count
Windows operating system Contains rules that block request patterns associated with exploiting vulnerabilities specific to Windows, (e.g., PowerShell commands). This can help prevent exploits that allow attacker to run unauthorized commands or execute malicious code.	200	<input type="radio"/> Add to web ACL
WordPress application The WordPress Applications group contains rules that block request patterns	100	<input type="radio"/> Add to web ACL

aws

Services

Search for services, features, marketplace products, and docs

[Alt+S]

Shoumit Karnik

resources

Step 2

Add rules and rule groups

Step 3

Set rule priority

Step 4

Configure metrics

Step 5

Review and create web ACL

A rule defines attack patterns to look for in web requests and the action to take when a request matches the patterns. Rule groups are reusable collections of rules. You can use managed rule groups offered by AWS and AWS Marketplace sellers. You can also write your own rules and use your own rule groups.

Rules

If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

Edit

Delete

Add rules

<input type="checkbox"/>	Name	Capacity	Action
<input type="checkbox"/>	AWS-AWSManagedRulesAdminProtectionRuleSet	100	Use rule actions
<input type="checkbox"/>	AWS-AWSManagedRulesAmazonIpReputationList	25	Use rule actions
<input type="checkbox"/>	AWS-AWSManagedRulesAnonymousIpList	50	Use rule actions
<input type="checkbox"/>	AWS-AWSManagedRulesCommonRuleSet	700	Use rule actions
<input type="checkbox"/>	AWS-AWSManagedRulesKnownBadInputsRuleSet	200	Use rule actions
<input type="checkbox"/>	AWS-AWSManagedRulesLinuxRuleSet	200	Use rule actions
<input type="checkbox"/>	AWS-AWSManagedRulesSQLiRuleSet	200	Use rule actions

Web ACL rule capacity units used

The total capacity units used by the web ACL can't exceed 1500.

Web ACL rule capacity units used

The total capacity units used by the web ACL can't exceed 1500.

1475/1500 WCUs

Default web ACL action for requests that don't match any rules

Default action

☒ Allow

☐ Block

WAF & Shield

AWS WAF

Getting started

Web ACLs

IP sets

Regex pattern sets

Rule groups

AWS Marketplace

Switch to AWS WAF Classic

AWS Shield

AWS Firewall Manager

Introducing the new AWS WAF

We've improved the console and API experience and added AWS Managed Rules. [Learn more.](#)

Note: The previous version of AWS WAF is now named AWS WAF Classic. To access resources created with that version, switch to AWS WAF Classic.

Success

You successfully created the web ACL: ENPM809F_Final

AWS WAF > Web ACLs

Web ACLs

Info

US East (N. Virginia)

Copy ARN

Delete

Create web ACL

Find web ACLs

< 1 > ⚙

<input type="radio"/>	Name	Description	ID
<input type="radio"/>	ENPM809F_Final	-	f5780f8e-a87e-4e54-86c1-6bd38ae9ad3a

For Employee VPC settings the following steps will be followed:

Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

VPC with Public and Private Subnets

VPC with Public and Private Subnets and Hardware VPN Access

VPC with a Private Subnet Only and Hardware VPN Access

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

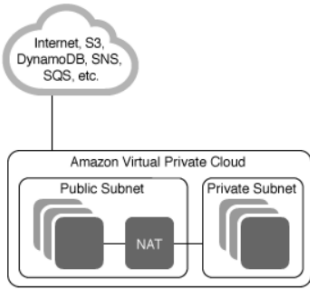
Creates:

A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via Network Address Translation (NAT). (Hourly charges for NAT devices apply.)

Important:

If you are using a Local Zone with your VPC [follow this link](#) to create your VPC.

Select



A VPC with Public and Private Subnets will be selected because we do not need Employee instances accessible from the internet, hence they will be included in the private subnet.

Step 2: VPC with Public and Private Subnets

IPv4 CIDR block:* 10.0.0.0/16 (65531 IP addresses available)

IPv6 CIDR block: ☒ No IPv6 CIDR Block
☐ Amazon provided IPv6 CIDR block
☐ IPv6 CIDR block owned by me

VPC name:

Public subnet's IPv4 CIDR:* 10.0.0.0/24 (251 IP addresses available)

Availability Zone:* No Preference ▾

Public subnet name:

Private subnet's IPv4 CIDR:* 10.0.1.0/24 (251 IP addresses available)

Availability Zone:* No Preference ▾

Private subnet name:

You can add more subnets after AWS creates the VPC.

Specify the details of your NAT instance ([Instance rates apply](#)).

Instance type:* t2.micro ▾

Key pair name: No key pair ▾

Service endpoints

Add Endpoint

Public subnet's IPv4 CIDR: 10.0.0.0/24 (251 IP addresses available)

Availability Zone: No Preference

Public subnet name: Public subnet

Private subnet's IPv4 CIDR: 10.0.1.0/24 (251 IP addresses available)

Availability Zone: No Preference

Private subnet name: Private subnet

You can add more subnets after AWS creates the VPC.

Specify the details of your NAT instance (Instance rates apply).

Instance type: t2.micro

Key pair name: No key pair

Service endpoints

Add Endpoint

Enable DNS hostnames: ☒ Yes ☐ No

Hardware tenancy: Default

Running NAT Instance (This may take a few minutes)...

Use a NAT gateway instead

Cancel and Exit **Back** **Create VPC**

The IP address ranges of the subnets created are:

Subnets (6) [Info](#)

[Refresh](#) [Actions](#) [Create subnet](#)

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6
<input type="checkbox"/>	-	subnet-50b8957b	Available	vpc-0e6f2676	172.31.48.0/20	-
<input type="checkbox"/>	Private subnet	subnet-0d98141b19fc19787	Available	vpc-0bdaf4d04d0c5e434	10.0.1.0/24	-
<input type="checkbox"/>	-	subnet-9ade04e2	Available	vpc-0e6f2676	172.31.16.0/20	-
<input type="checkbox"/>	-	subnet-cf589c85	Available	vpc-0e6f2676	172.31.32.0/20	-
<input type="checkbox"/>	Public subnet	subnet-0204284c3487bf68a	Available	vpc-0bdaf4d04d0c5e434	10.0.0.0/24	-
<input type="checkbox"/>	-	subnet-8653e6db	Available	vpc-0e6f2676	172.31.0.0/20	-

Next an instance is to be created for the website to be migrated to:

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start

My AMIs

AWS Marketplace

Community AMIs

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e472933a1395e172 (64-bit x86) / ami-0b0154d3d8011b0cd (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is approaching end of life on December 31, 2020 and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

☒ 64-bit (x86) ☐ 64-bit (Arm)

For the security group configurations, the considerations that have been taken into account are as follows:

- The website will be facing the internet, hence it will be open to all and hence the IP address configuration would be 0.0.0.0/24 for ports 80 and 5000.
- The employee private subnet will be used for the employee network. They will be given some special port access privileges like SSH and ICMP. This would help them with:
 - Accessing instances for changing/modifying code
 - Performing network diagnostics in case of system failures

Step 6: Configure Security Group

Assign a security group: ☒ Create a new security group

☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 10.0.1.0/24	SSH for employees
All ICMP - IPv	ICMP	0 - 65535	Custom 10.0.1.0/24	ICMP for employees
HTTP	TCP	80	Anywhere 0.0.0.0/0, ::/0	For customers to access website
Custom TCP	TCP	5000	Custom 0.0.0.0/0, ::/0	For customers to access website

Add Rule

For the internal EC2 instance security, as an added layer of security, iptables can be installed and configured on the instance where the website is running.

On the EC2 instance after SSH access, the following commands will be run:

```
sudo yum install iptables-services
```

```
sudo iptables -nvL
```

These commands install iptables and give a current state of the iptables firewall.

```
[ec2-user@ip-172-31-36-161 ~]$ sudo yum install iptables-services
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package iptables-services.x86_64 0:1.8.4-10.amzn2.1.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch      Version              Repository           Size
=====
Installing:
iptables-services      x86_64    1.8.4-10.amzn2.1.2   amzn2-core           58 k

Transaction Summary
=====
Install 1 Package

Total download size: 58 k
Installed size: 24 k
Is this ok [y/d/N]:
```

The rules added here will be as follows:


```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport 5000 -j ACCEPT
```

```
iptables -A INPUT -p icmp -s 10.0.1.0/24 -j ACCEPT
```

```
iptables -R INPUT 1 -p tcp -s 10.0.1.0/24 --dport 22 -j ACCEPT
```

```
[ec2-user@ip-172-31-36-161 ~]$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
[ec2-user@ip-172-31-36-161 ~]$ sudo iptables -A INPUT -p tcp --dport 5000 -j ACCEPT
[ec2-user@ip-172-31-36-161 ~]$ sudo iptables -A INPUT -p icmp -s 10.0.1.0/24 -j ACCEPT
[ec2-user@ip-172-31-36-161 ~]$ sudo iptables -R INPUT 1 -p tcp -s 10.0.1.0/24 --dport 22 -j ACCEPT
```

The rules have been updated and specific IP ranges have been let in for SSH and ICMP and all traffic is allowed for port 80 and 5000, since the website is running on these two ports.

```
[ec2-user@ip-172-31-36-161 ~]$ sudo iptables -nvL
Chain INPUT (policy ACCEPT 15 packets, 1084 bytes)
  pkts bytes target     prot opt in     out     source         destination
    0     0 ACCEPT     tcp  --  *      *       10.0.1.0/24    0.0.0.0/0
    0     0 tcp dpt:22  tcp  --  *      *       0.0.0.0/0      0.0.0.0/0
    0     0 tcp dpt:5000 tcp  --  *      *       0.0.0.0/0      0.0.0.0/0
    0     0 ACCEPT     icmp --  *      *       10.0.1.0/24    0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source         destination

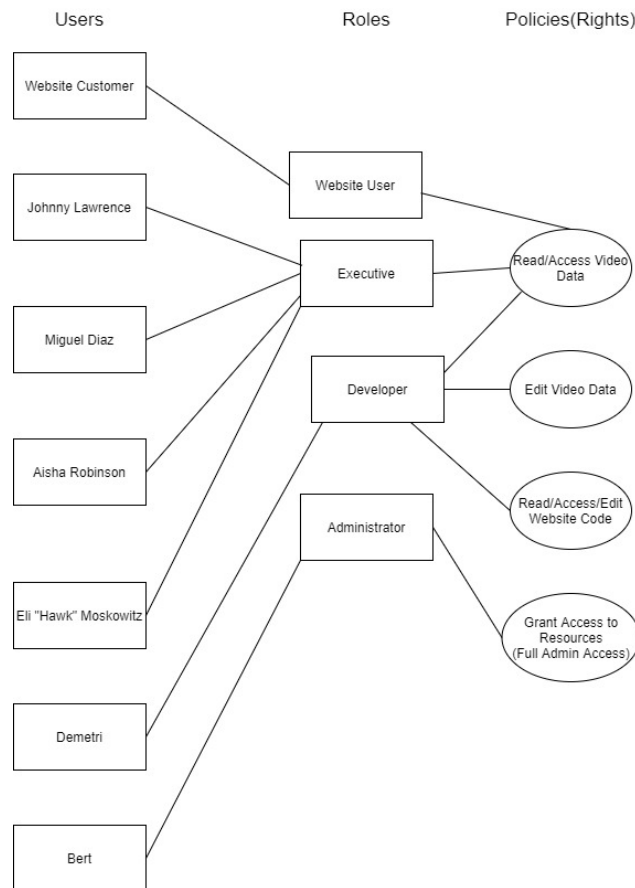
Chain OUTPUT (policy ACCEPT 9 packets, 2028 bytes)
  pkts bytes target     prot opt in     out     source         destination
[ec2-user@ip-172-31-36-161 ~]$
```

Note that these rules were based off the data provided in the sample VM and the data provided in the document:

```
enpm809j@ubuntu:~$ sudo netstat -tulpn | grep LISTEN
[sudo] password for enpm809j:
tcp        0      0 0.0.0.0:5000          0.0.0.0:*             LISTEN      862/python3
tcp        0      0 127.0.0.1:3306        0.0.0.0:*             LISTEN      871/mysqld
tcp        0      0 0.0.0.0:80           0.0.0.0:*             LISTEN      884/nginx -g daemon
tcp        0      0 0.0.0.0:22           0.0.0.0:*             LISTEN      851/sshd
tcp6       0      0 :::80                :::*                  LISTEN      884/nginx -g daemon
tcp6       0      0 :::22                :::*                  LISTEN      851/sshd
```

IAM policy/role configurations:

We have the following organizational structure:



These roles were outlined in the initial plan provided and we will be building our roles and policies based on that plan.

The AWS Identity and Access Management console will be used.

Various employee roles will be associated to our instance. Their configuration settings have been given below. The Website user role for the customers has not been configured, since they will be directly accessing the website. This will be done using a temporary key being provided to them from the EC2 instance and then later they can get read access to a bucket which will store the data.

Roles:

Three employee roles which would be created are:

Edit_Access_Role – Given to developers to fully access the EC2 instance and make changes to code or data.

Org_Admin Role – Given to the system administrator (Bert). This enables them to add/delete new users and make changes to the configurations of the system.

Read_Access_Role – Given to high level executives of the company, who only need to view the data or code but should not be given edit access

The permission policies attached to each role are as follows:

Edit_Access_Role :

Create role

1


Review

Provide the required information below and review this role before you create it.

Role name*
Use alphanumeric and '+=, @, _' characters. Maximum 64 characters.


Role description
Maximum 1000 characters. Use alphanumeric and '+=, @, _' characters.

Trusted entities AWS service: ec2.amazonaws.com



Policies  [AmazonEC2FullAccess](#) 

Policies attached: AmazonEC2FullAccess

Org_Admin Role:

 Services ▾

[Alt+S]

  Shoumit Karnik ▾

Create role

1

2

3

4

Review

Provide the required information below and review this role before you create it.

Role name*

Use alphanumeric and '+=, @, _' characters. Maximum 64 characters.

Role description

Maximum 1000 characters. Use alphanumeric and '+=, @, _' characters.

Trusted entities AWS service: ec2.amazonaws.com

Policies  [AdministratorAccess](#) 

 [AmazonEC2ContainerServiceFullAccess](#) 

Policies attached: AdministratorAccess, AmazonEC2ContainerServiceFullAccess

Read_Access_Role:

aws

Services

Search for services, features, marketplace products, and docs

[Alt+S]

Shoumit Karnik

Create role

1

2

3

4

Review

Provide the required information below and review this role before you create it.

Role name*

Use alphanumeric and '+', '@', '-', '_' characters. Maximum 64 characters.

Role description

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+', '@', '-', '_' characters.

Trusted entities

AWS service: ec2.amazonaws.com

Policies

AmazonEC2ReadOnlyAccess

Policies attached: AmazonEC2ReadOnlyAccess

Policies:

The policy details are as follows :

AmazonEC2FullAccess:

[Policies](#) > [AmazonEC2FullAccess](#)

Summary

Policy ARN

arn:aws:iam::aws:policy/AmazonEC2FullAccess

Description

Provides full access to Amazon EC2 via the AWS Management Console.

Permissions

Policy usage

Policy versions

Access Advisor

Policy summary

{ } JSON

Filter

Service	Access level	Resource	Request condition
Allow (6 of 264 services) Show remaining 258			
CloudWatch	Full access	All resources	None
EC2	Full access	All resources	None
EC2 Auto Scaling	Full access	All resources	None
ELB	Full access	All resources	None
ELB v2	Full access	All resources	None
IAM	Limited: Write	All resources	Multiple

This policy will be given to the developer. It ensures that the developer can make necessary changes to the EC2 instance but cannot override the Administrator privileges.

AdministratorAccess:

aws

Services

Search for services, features, marketplace products, and docs

[Alt+S]

Shoumit Kamik

Identity and Access Management (IAM)

Dashboard

Access management

Groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Organization activity

Service control policies

Policies > AdministratorAccess

Summary

Policy ARN

arn:aws:iam::aws:policy/AdministratorAccess

Description

Provides full access to AWS services and resources.

Permissions

Policy usage

Policy versions

Access Advisor

Policy summary

{ } JSON

Filter

Service	Access level	Resource	Request condition
Allow (264 of 264 services)			
Access Analyzer	Full access	All resources	None
Account	Full access	All resources	None
Activate	Full access	All resources	None
Alexa for Business	Full access	All resources	None
AMP	Full access	All resources	None
Amplify	Full access	All resources	None

AmazonEC2ContainerServiceFullAccess:

aws

Services

Search for services, features, marketplace products, and docs

[Alt+S]

Shoumit Kamik

Global

Identity and Access Management (IAM)

Dashboard

Access management

Groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Organization activity

Service control policies

Policies > AmazonEC2ContainerServiceFullAccess

Summary

Policy ARN

arn:aws:iam::aws:policy/AmazonEC2ContainerServiceFullAccess

Description

Provides administrative access to Amazon ECS resources.

Permissions

Policy usage

Policy versions

Access Advisor

Policy summary

{ } JSON

Filter

Service	Access level	Resource	Request condition
Allow (9 of 264 services) Show remaining 255			
CloudFormation	Limited: List, Read, Write	All resources	None
CloudWatch	Limited: Read	All resources	None
EC2	Limited: List, Read	All resources	None
EC2 Auto Scaling	Full: List, Read Limited: Write	All resources	None
Elastic Container Service	Full access	All resources	None
ELB	Full access	All resources	None

It is necessary to give the administrator full access to these resources, since they will be fully responsible for managing services like configurations, IAM etc. Hence the organization's amazon account's Administrator role needs to be assigned to them.

AmazonEC2ReadOnlyAccess:

identity and Access Management (IAM)

- Dashboard
- Access management
 - Groups
 - Users
 - Roles
 - Policies**
 - Identity providers
 - Account settings
- Access reports
 - Access analyzer
 - Archive rules
 - Analizers
 - Settings
 - Credential report
 - Organization activity

Policies > AmazonEC2ReadOnlyAccess

Summary

Policy ARN am:aws:iam::aws:policy/AmazonEC2ReadOnlyAccess

Description Provides read only access to Amazon EC2 via the AWS Management Console.

Permissions | Policy usage | Policy versions | Access Advisor

Policy summary | {} JSON

Q Filter

Service	Access level	Resource	Request condition
Allow (5 of 264 services) Show remaining 259			
CloudWatch	Limited: List, Read	All resources	None
EC2	Limited: List, Read	All resources	None
EC2 Auto Scaling	Full: List, Read	All resources	None
ELB	Full: List, Read	All resources	None
ELB v2	Full: Read	All resources	None

This role helps the executive staff to just have read permissions to the instance so that they can access the data and make sure that everything is set correctly. If they need any configuration or development changes, they should be contacting the respective staff.

The users listed above can be assigned the aforementioned roles, hence giving them the required access without providing more access than required.

Sample CloudFormation Templates:

Some very simple sample CloudFormation templates in YAML(YAML Ain't Markup Language) format have been outlined for the migration task. Note that this only has automated the security groups and the IAM roles for now.

For VPC automation:

CloudFormation YAML:

```

Description: This template deploys a VPC, with a public and
private subnet in the same availability zone. It deploys an
internet gateway, with a default
route on the public subnets. It deploys a NAT gateway and a
default routes for them in the private subnet.

Parameters:
  EnvironmentName:
    Description: An environment name that is prefixed to
resource names
    Type: String

  VpcCIDR:
    Description: Please enter the IP range (CIDR notation) for
this VPC
    Type: String
    Default: 10.0.0.0/16

  PublicSubnetCIDR:
    Description: Please enter the IP range (CIDR notation) for
the public subnet in the first Availability Zone
    Type: String
    Default: 10.0.0.0/24

  PrivateSubnetCIDR:
    Description: Please enter the IP range (CIDR notation) for
the private subnet in the first Availability Zone
    Type: String
    Default: 10.0.1.0/24

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VpcCIDR
      EnableDnsSupport: true
      EnableDnsHostnames: true

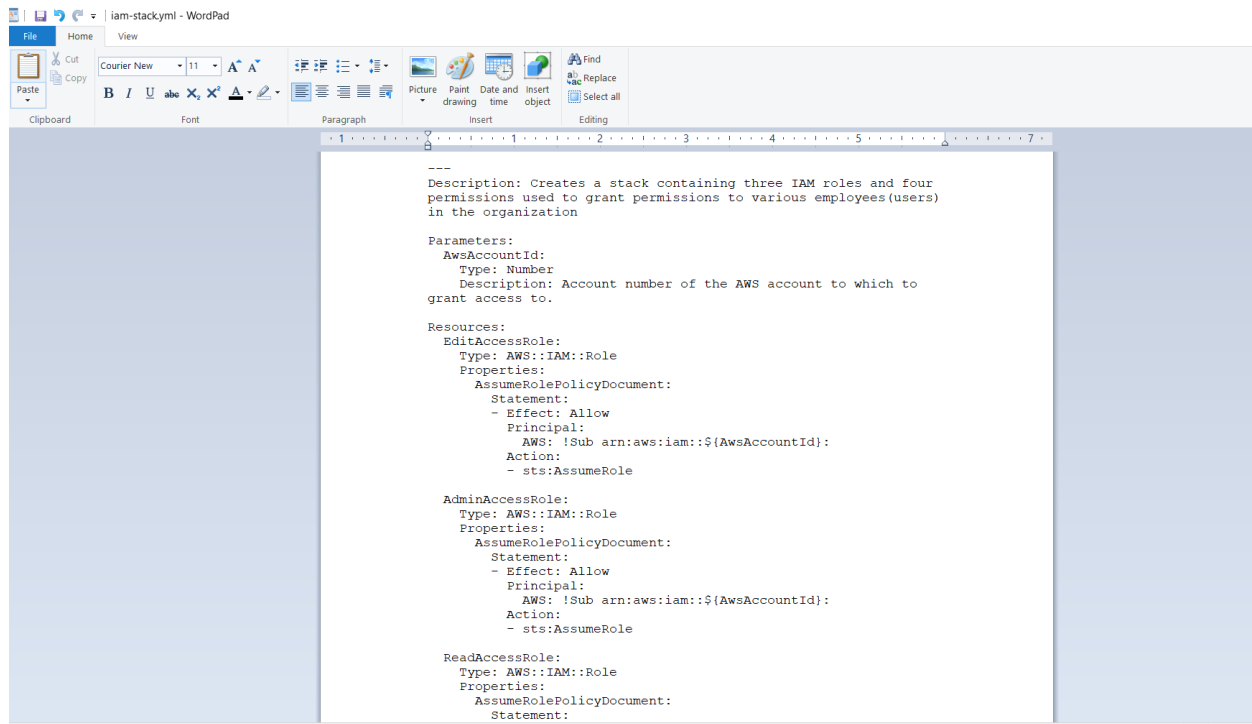
```

This file deploys the VPC and the security groups discussed above. It is in YAML format. It has been attached as a separate file. I have used the file given in the reference and modified it to fit the purpose of the project. [\[Ref.\]](#)

Note: The EC2 instance creation and iptables configuration still needs to be done manually.

For IAM Role Automation:

CloudFormation YML:



```
---
Description: Creates a stack containing three IAM roles and four
permissions used to grant permissions to various employees(users)
in the organization

Parameters:
  AwsAccountId:
    Type: Number
    Description: Account number of the AWS account to which to
grant access to.

Resources:
  EditAccessRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: Allow
            Principal:
              AWS: !Sub arn:aws:iam:${AwsAccountId}:
            Action:
              - sts:AssumeRole

  AdminAccessRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
          - Effect: Allow
            Principal:
              AWS: !Sub arn:aws:iam:${AwsAccountId}:
            Action:
              - sts:AssumeRole

  ReadAccessRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Statement:
```

This YML file can be used for the automated creation of the IAM roles and policies that were seen in the IAM policy/role configurations section. This file has also been attached as a separate file. The code has been taken from the various JSON (JavaScript Object Notation) files of the different policies and roles specified. The Account ID spaces however have been left blank so that Account credentials cannot be leaked. They should be filled in with the organization's Amazon account credentials and permissions.

These would be the initial cloud migration steps strictly from a security standpoint. These have been developed using a free AWS account. (Except for AWS WAF, it costed 0.6\$)

FUTURE SCOPE

Given more time and access to more resources in AWS which are mostly paid services the following can be done:

- A more detailed approach for the AWS CloudFormation Templates
- A server(NGINX) was seen in the code. A separate node can be created for that in AWS Console using Cloud Migration Hub.
- The system is DDoS(Distributed Denial of Service) resilient because of the AWS WAF right now, but to make it more efficient a distributed set of nodes and EC2 instances will be required as mentioned in initial document.
- Right now, the system does not use rolling keys and it is assumed that everyone will have their own key to access the system. This can be made more secure with the use of Amazon KeyStore as mentioned in the previous migration proposal.