Aayushi Dubey **UID: - 116934596**

ENPM809J

Final Project.

Reasons for Migration to the Cloud.

1. Cobra Kai wishes to develop a better patching strategy, backup strategy, and account permission strategy.
2. They would like to mitigate DDoS attacks, hardware failures. They have also experienced compromise attempts.
3. They want to provide better speed in streaming, downloads, and order processing, so customers don't face any problem. All the customer's details like name, phone number, email, credit card information is stored, and we want to keep them safe from rival dojo by Daniel LaRusso.

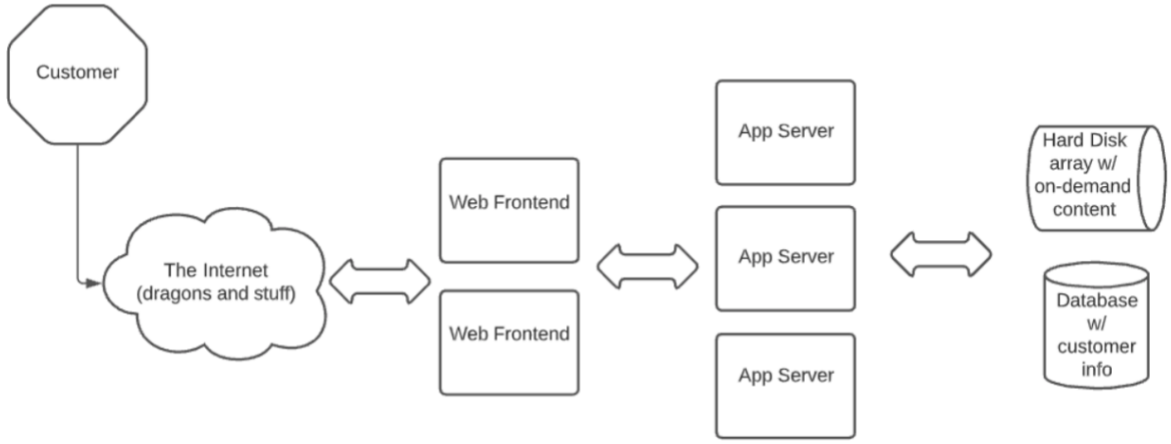
Development Team.

- **Johnny Lawrence** - The founder of Cobra Kai and the visionary disrupting karate and karate training with the introduction of his streaming platform for karate training
- **Miguel Diaz** - Chief Operating Officer. Miguel is the person in charge of daily operations for Cobra Kai and its streaming platform.
- **Aisha Robinson** - Chief Information Security Officer. Aisha is the enforcer for Cobra Kai, both in-person and online. Her security and risk-focused mindset helps her discover and mitigate risks before they are exploited. She is the reason you were hired to help with this move to the cloud.
- **Eli "Hawk" Moskowitz** -Chief Information Officer. "Hawk" is the brains behind the development of Cobra Kai's streaming platform.
- **Demetri** - Web developer. He heads the developer for implementing Hawk's vision.
- **Bert** - System administrator. Although small in stature, Bert is highly skilled when it comes to technical items

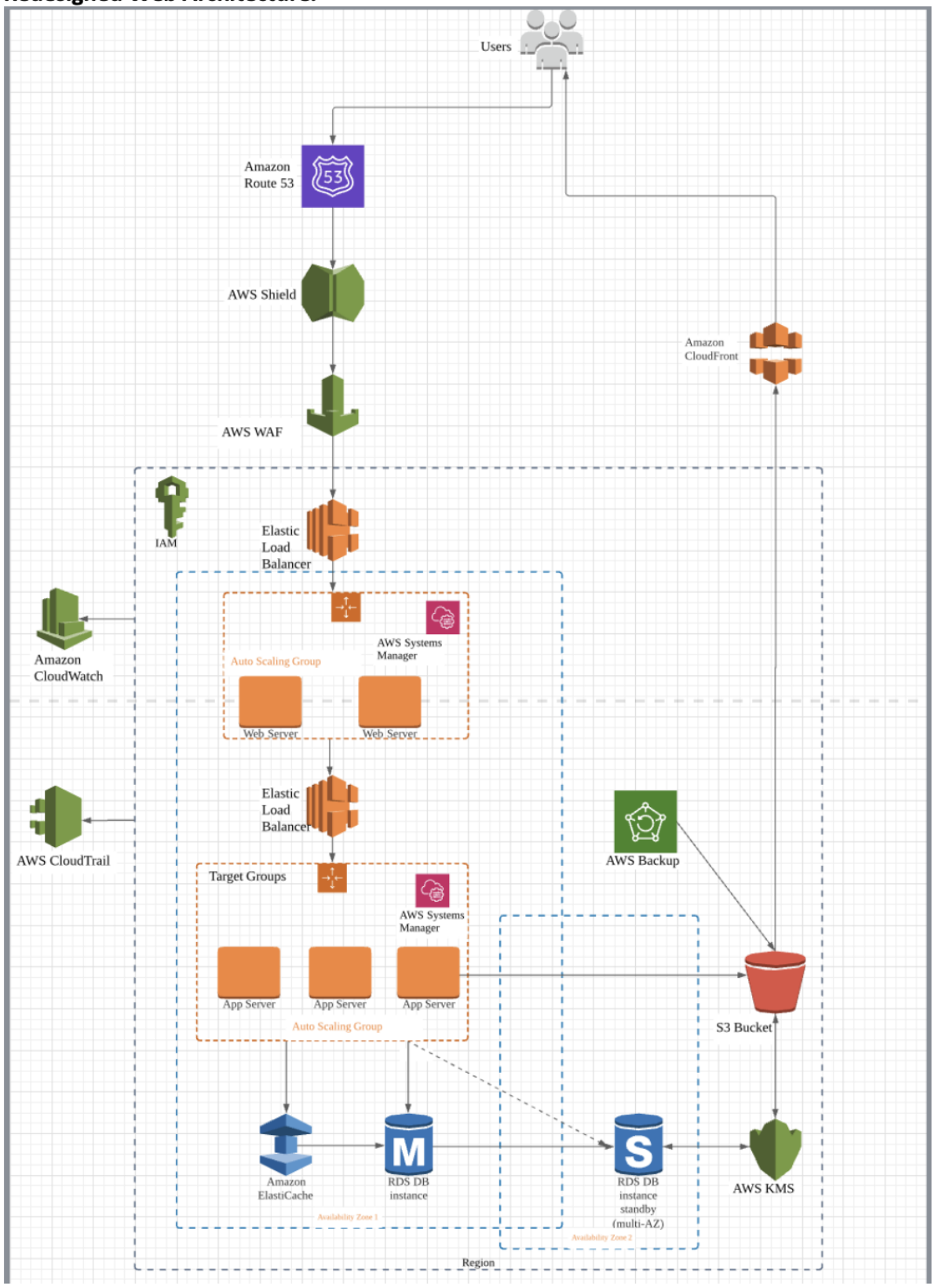
Issues to consider.

1. There are no account permissions to the team on the current website.
2. The backup strategy is not configured, resulting in loss of data.
3. No prevention against DDoS attacks.
4. High Latency.
5. Slow downloading while using the website.
6. No auto-scaling.
7. Patching strategy.

Current Website Architecture



Redesigned Web Architecture.

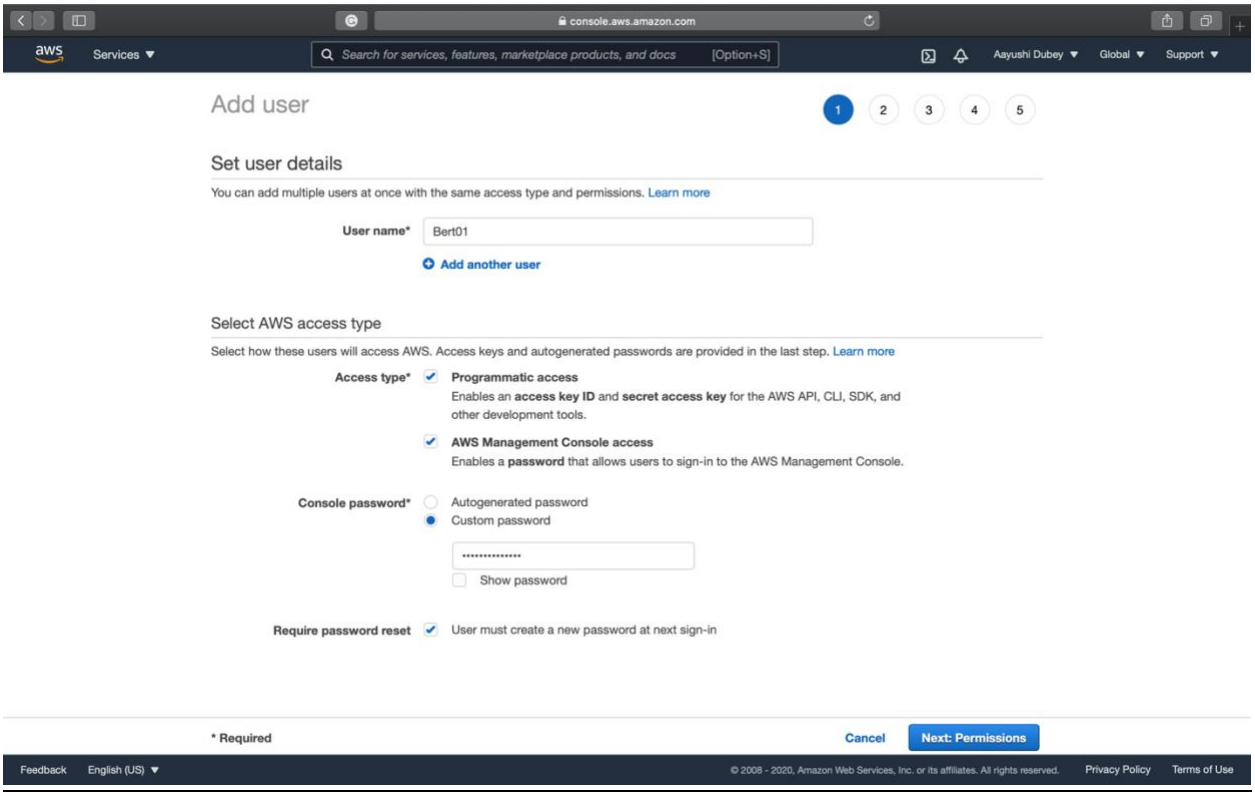


1. There are no account permissions to the team on the current website.

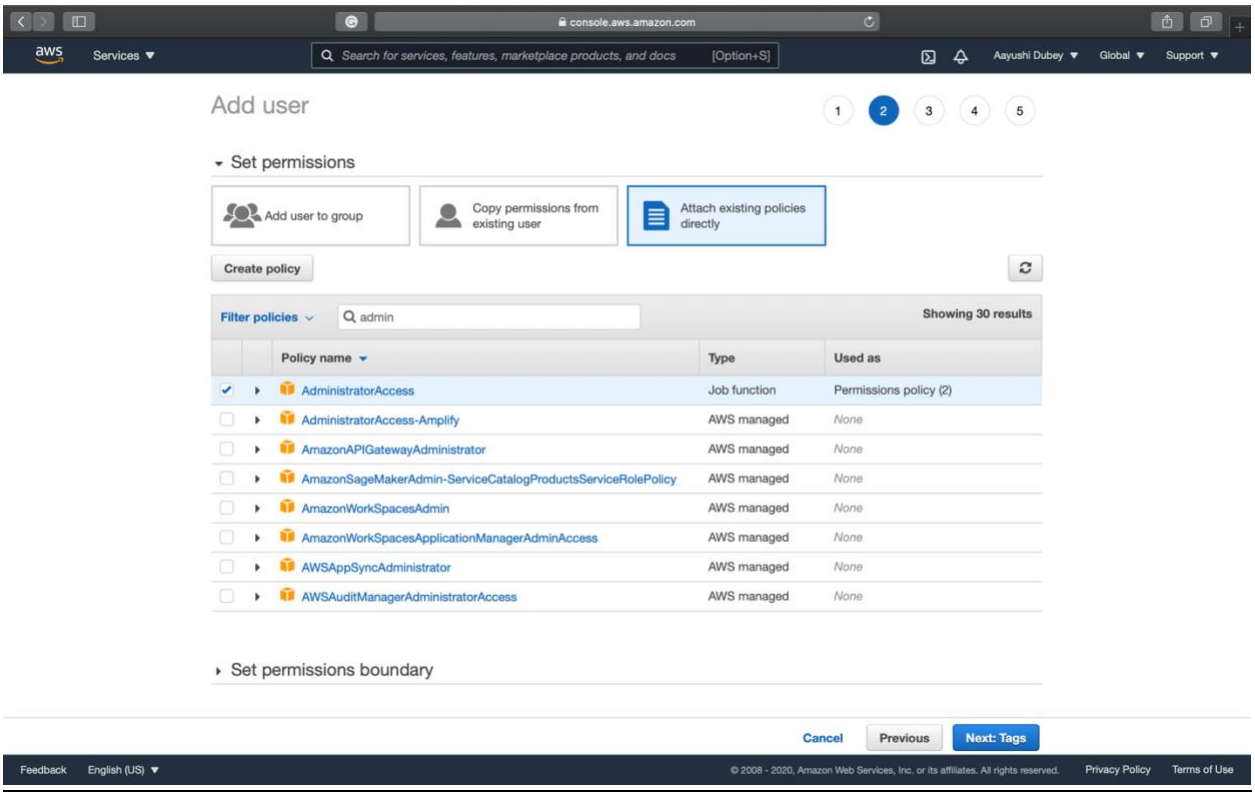
To resolve the issue of Cobra Kai having no account permissions for various users, who will need to have access to various AWS resources as well as won't have the permissions to other resources, we will use Identity and Access Management (IAM), which will be used to create Users and add to users into a group. Providing permissions to perform an action to a group explicitly provides those permissions to every user present in that group. If a User is present in multiple groups, then that user will inherit the permissions of both groups and can perform tasks as defined for both the groups.

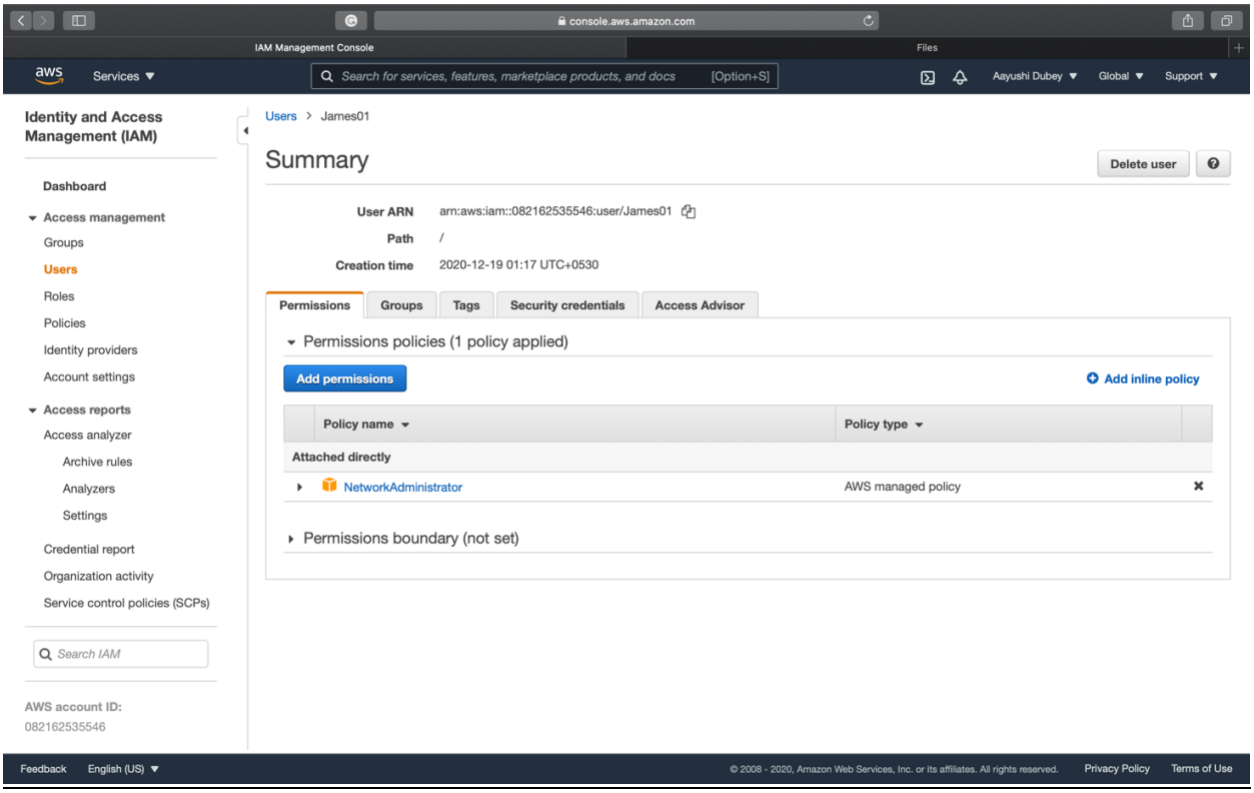
<u>Steps to configure IAM roles for various users to access various AWS Resources:</u>

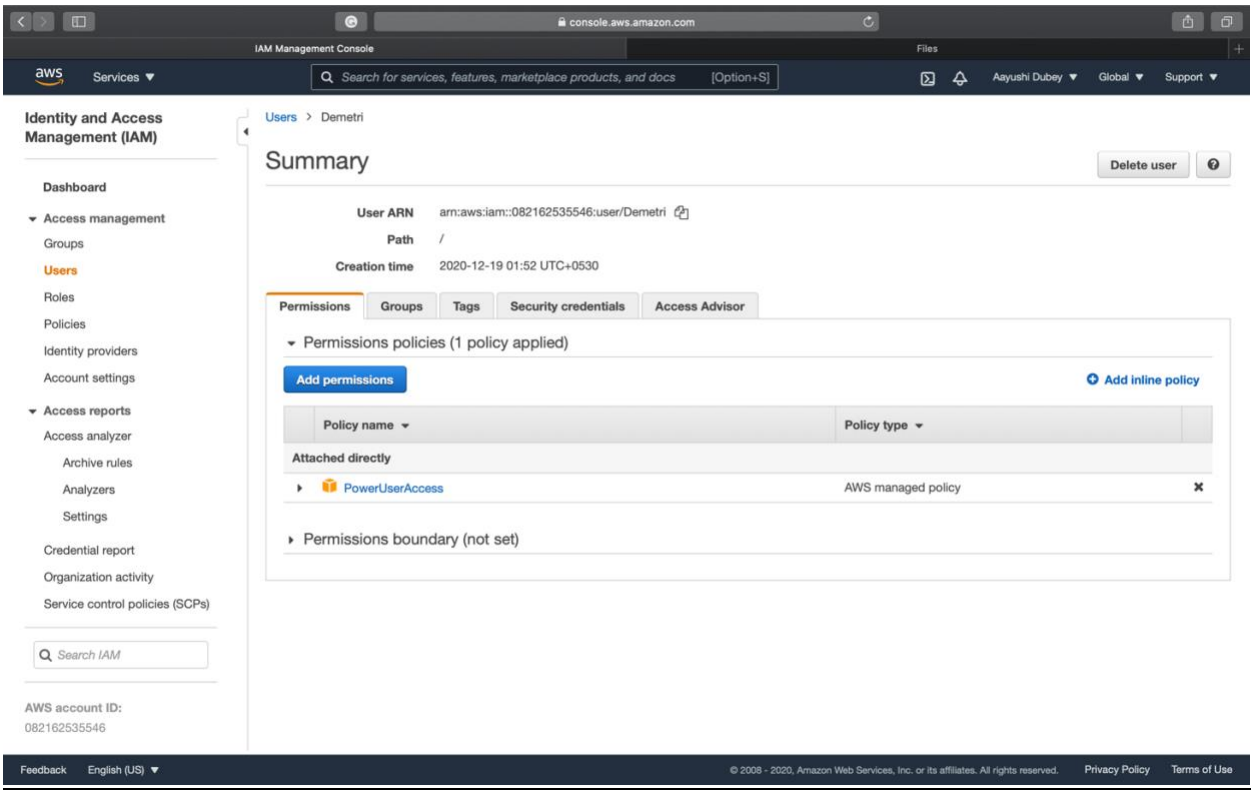
1. Go to Services and select IAM.
2. Choose User and click on Add User.



3. Click on Attach existing policies directly and choose the appropriate permissions.
4. Select Network Administrator for User "James01" and PowerUserAccess for user "Demetri."


console.aws.amazon.com
aws
Services
Search for services, features, marketplace products, and docs [Option+S]
Aayushi Dubey
Global
Support
Add user
1 2 3 4 5
Set permissions
Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy
Filter policies
admin
Showing 30 results
Policy name
Type
Used as
AdministratorAccess
Job function
Permissions policy (2)
AdministratorAccess-Amplify
AWS managed
None
AmazonAPIGatewayAdministrator
AWS managed
None
AmazonSageMakerAdmin-ServiceCatalogProductsServiceRolePolicy
AWS managed
None
AmazonWorkSpacesAdmin
AWS managed
None
AmazonWorkSpacesApplicationManagerAdminAccess
AWS managed
None
AWSAppSyncAdministrator
AWS managed
None
AWSAuditManagerAdministratorAccess
AWS managed
None
Set permissions boundary
Cancel
Previous
Next: Tags
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


console.aws.amazon.com
IAM Management Console
Files
aws
Services
Search for services, features, marketplace products, and docs [Option+S]
Aayushi Dubey
Global
Support
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 (SCPs)
Search IAM
AWS account ID:
082162535546
Users > James01
Summary
Delete user
User ARN
arn:aws:iam::082162535546:user/James01
Path
/
Creation time
2020-12-19 01:17 UTC+0530
Permissions
Groups
Tags
Security credentials
Access Advisor
Permissions policies (1 policy applied)
Add permissions
Add inline policy
Policy name
Policy type
Attached directly
NetworkAdministrator
AWS managed policy
Permissions boundary (not set)
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



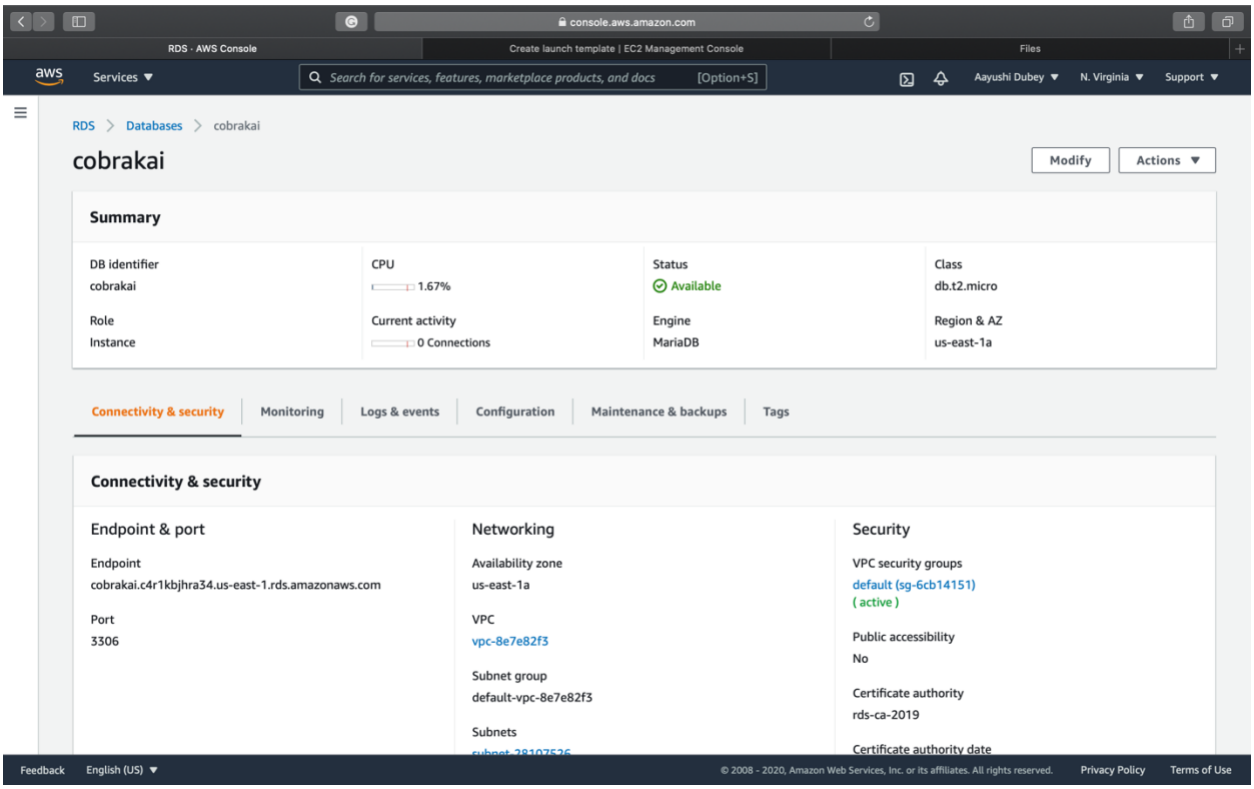
5. This summary page shows us the attached policies for each user and what AWS resources these users can access.

2. *The backup strategy is not configured, resulting in loss of data.*

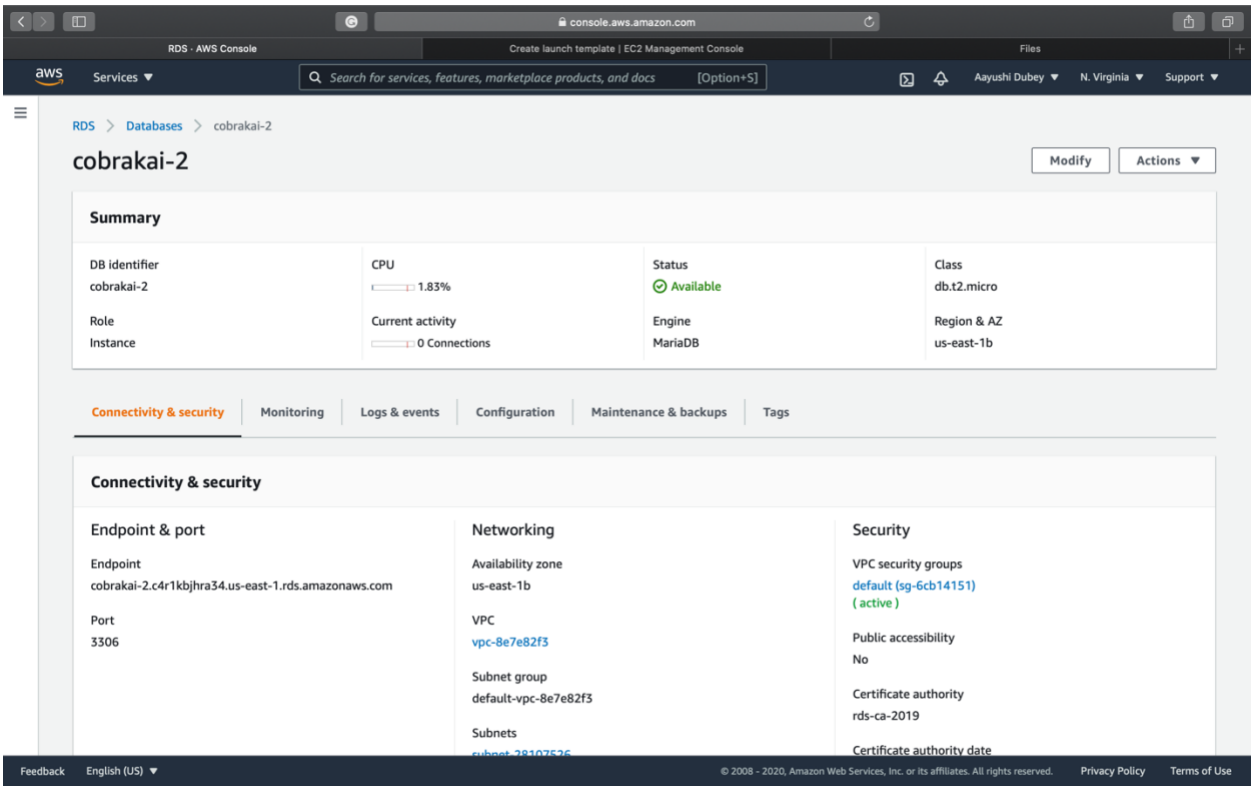
To resolve the issue of loss of data in case of natural disaster and mismanagement of data by the users, we will use Relational databases in different Availability zones. We will have relational databases in two different Availability Zones so that in the situation of any natural disaster, the customer's data will be protected with us, and hence this will increase the fault tolerance and provide high availability.

<u>Steps to configure relational databases in two different availability zones:</u>

1. Go to Services and select RDS.
2. Click on create Database.
3. Select Standard Create in Choose a database creation method.
4. Now choose MariaDB for the engine option.
5. Enter the Master Username and password.
6. Select the availability zone in us-east-1a.



7. Repeat the above steps to create a secondary database in a different Availability zone to make sure that if one database goes down, then we can use the backup database.

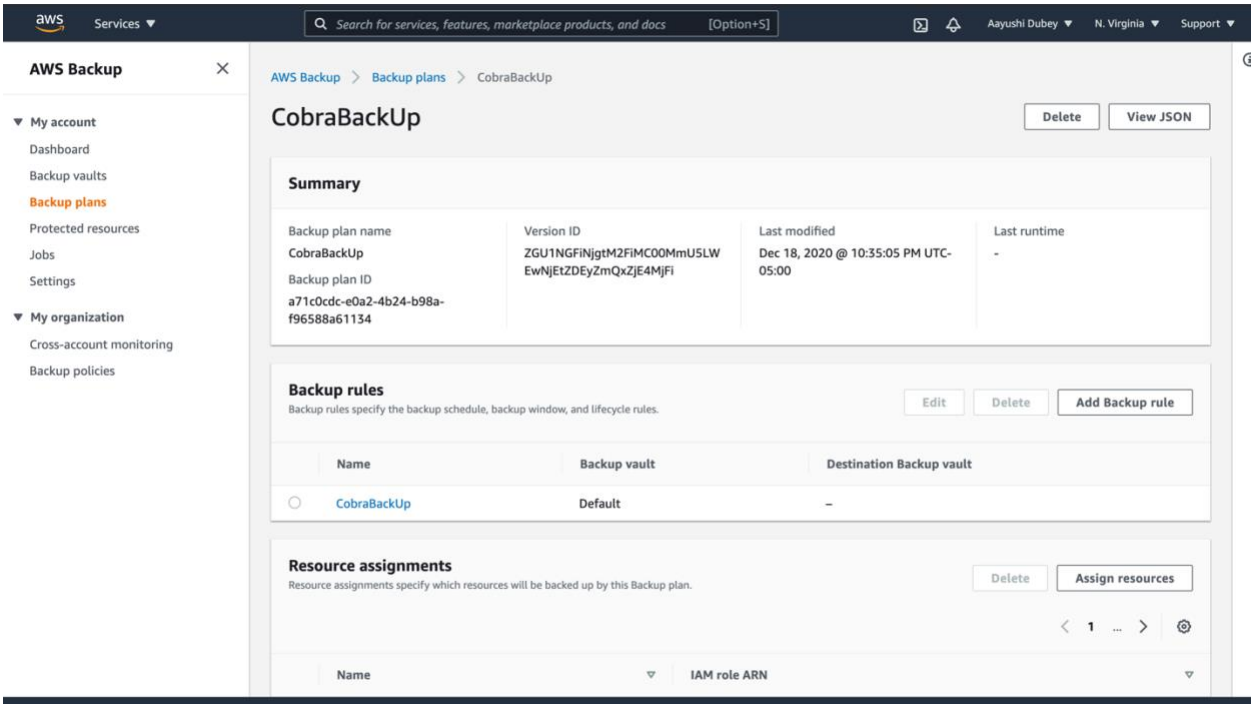


AWS Backup

AWS Backup is used to take the backup of various AWS resources at various defined intervals, whether it will be daily, weekly, or monthly. It will let define the Backup Vault, which will be further used for saving and storing the backup of those resources and which can be used for future use if in case any production system goes down.

Steps to configure AWS backup are as follows:

1. Go to services and choose AWS Backup.
2. Click on create a backup plan.
3. Now select either Start with a template, build a new plan, or Define a plan using JSON.
4. For our case select, to build a new plan.
5. Provide a name and the choose frequency of taking the backup.
6. Then Click on Create Plan.
7. The configuration of the AWS Backup will be as follows:

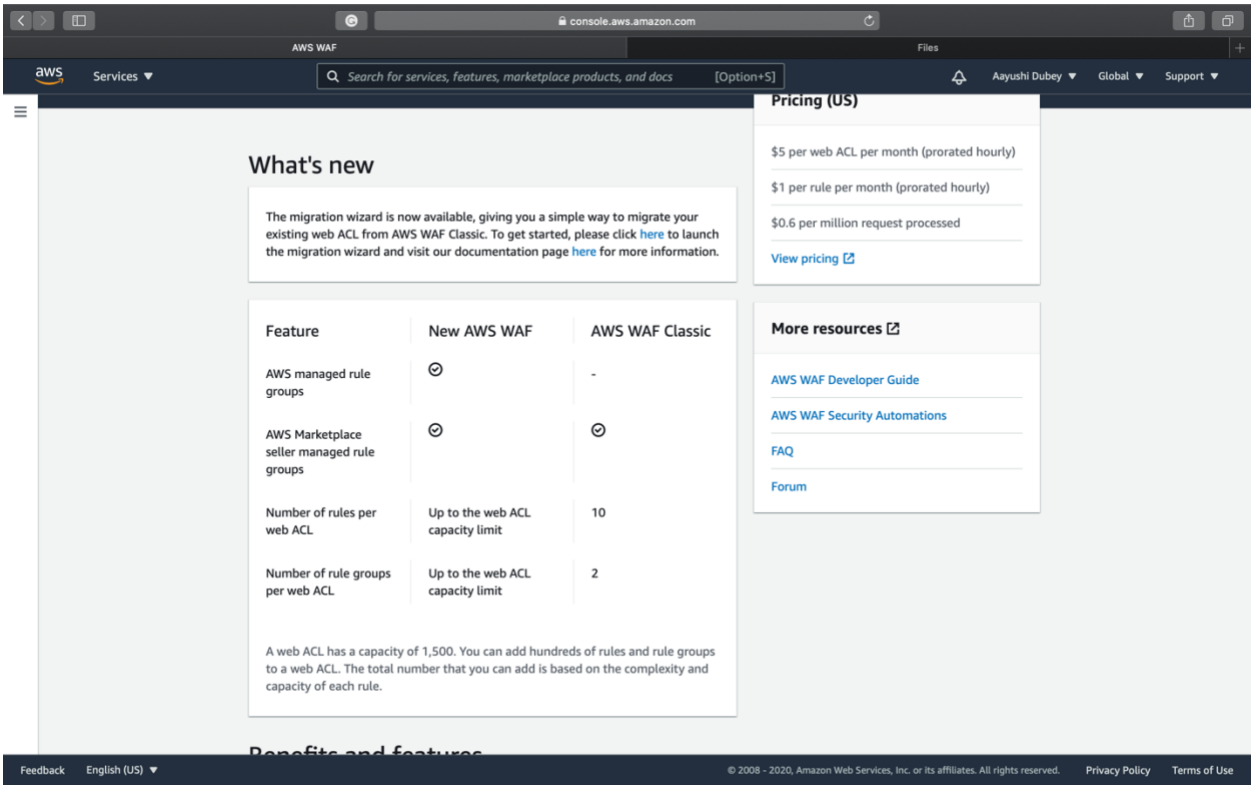


3. **No prevention against DDoS attacks.**

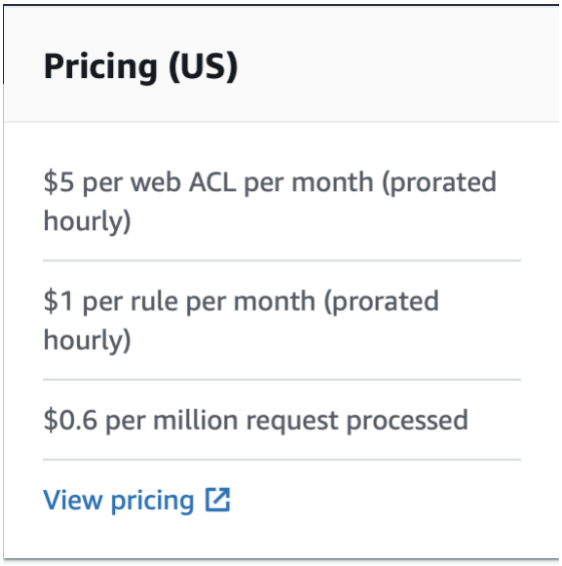
Cobra Kai web application does not have mechanism in place to prevent various DDoS attack that can happen on a Website as it the Web Application will be publicly available. To prevent DDoS attacks, we will utilize AWS Web Application Firewall (WAF) and AWS shield. Whenever an incoming data traffic from the internet passes these two-protection layers, i.e., AWS Shield and AWS WAF, then it can be safely said that the application hosted on the cloud is totally secure from the DDoS Attacks and other vulnerability.

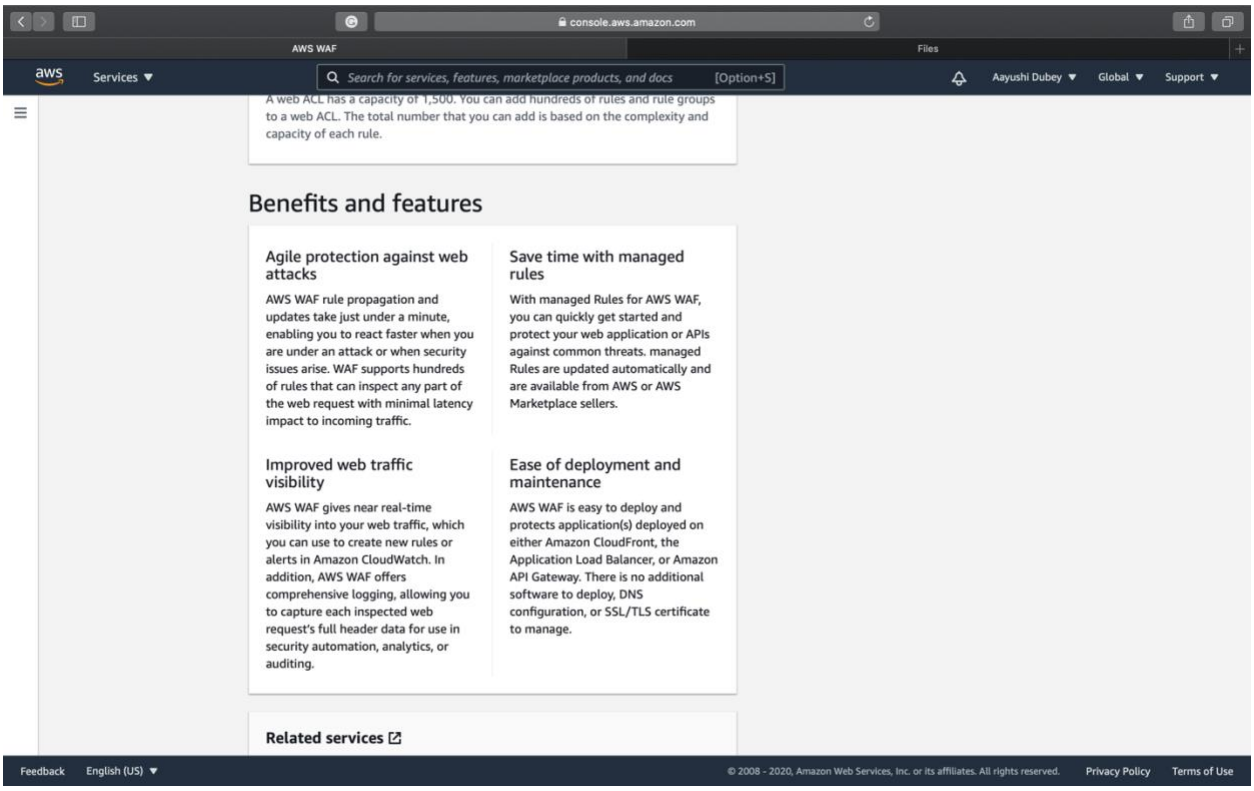
Steps to configure AWS Web Application Firewall (WAF) and AWS shield:

1. Go to Services and select WAF and Shield.
2. We will see that AWS WAF Classic and AWS shield standard are preconfigured with the AWS account as seen in the below console.

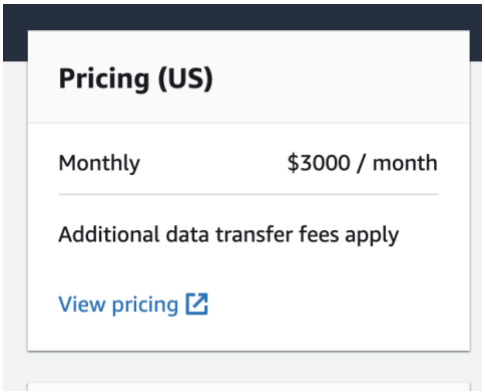


3. To configure New AWS WAF and create Access Control list we need to consider the AWS pricing for this.





4. The pricing for Shield Advanced is defined below.

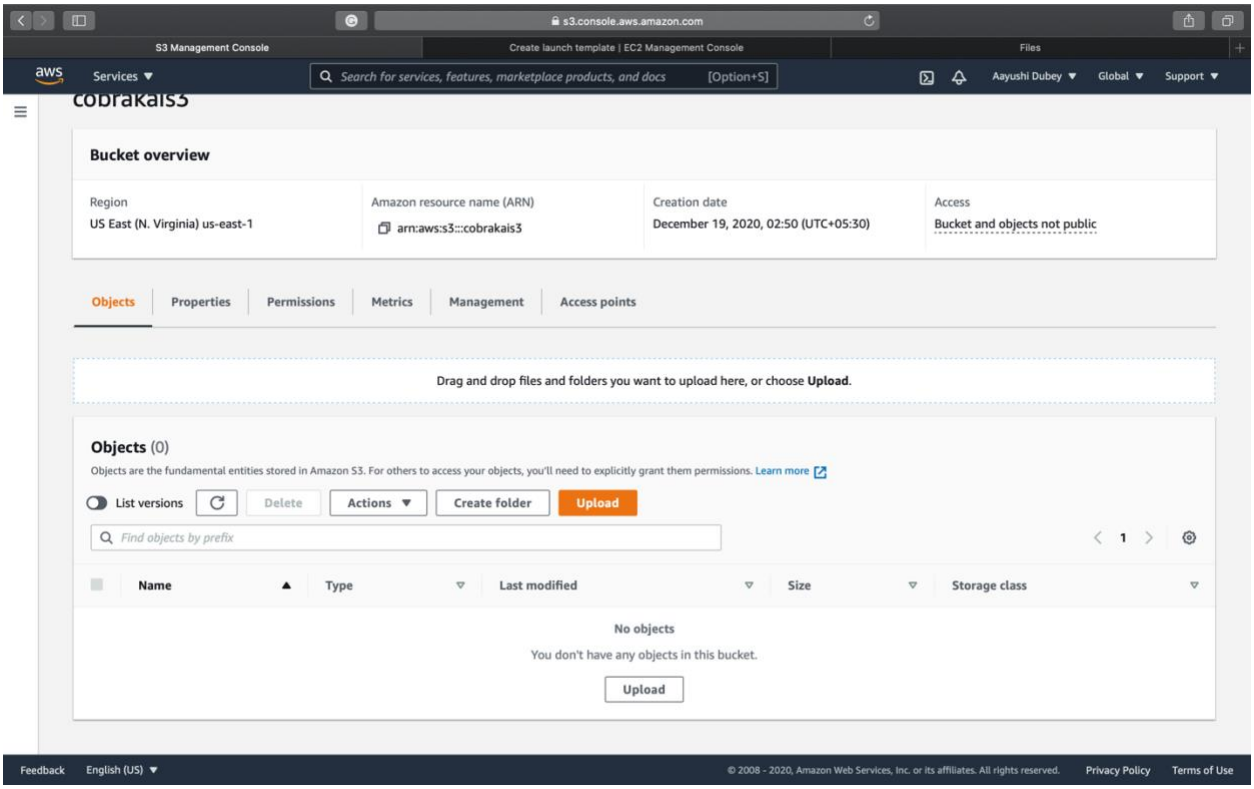


4. High Latency.

To ease out the problem of high latency associated with web application we will AWS Cloud Front which will be associated with AWS Simple Storage Service (S3) to provide seamless video experience for the users. AWS content delivery network uses Cloud Front to provide the requested data to the users. CDN is used to securely provide the requested contents like videos, charts, etc. to the customer with minimum latency, higher transfer speed and effective order processing. Cloud front eases out the delivery of that content by caching the content at the edge locations and providing it with minimum latency.

Steps to configure cloud front and associate it with S3 are as follows:

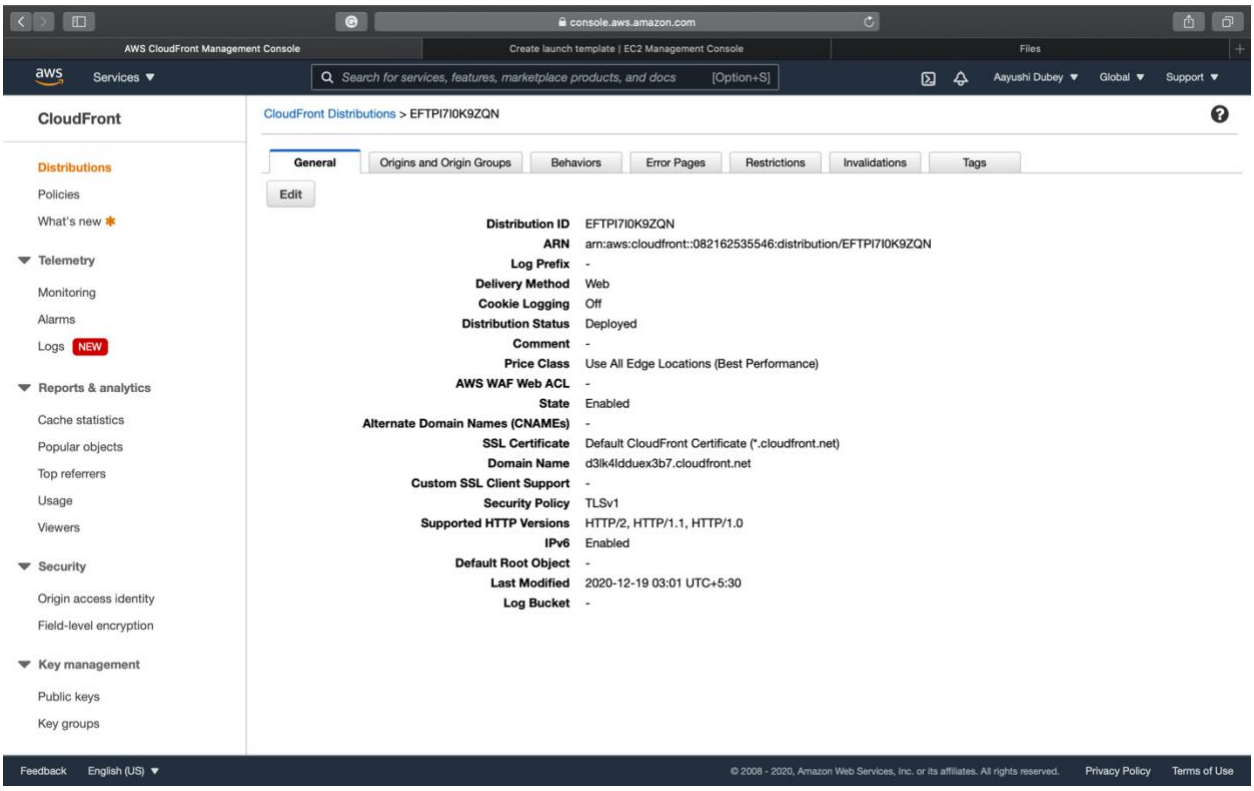
1. Go to services and select S3.
2. Click on create bucket and enter a unique bucket name.
3. Now select various settings for bucket versioning and Default encryption.
4. Provide an IAM role to the EC2 instance for S3FullAccess to access the objects stored in the S3 bucket.
5. The configured bucket is shown below.

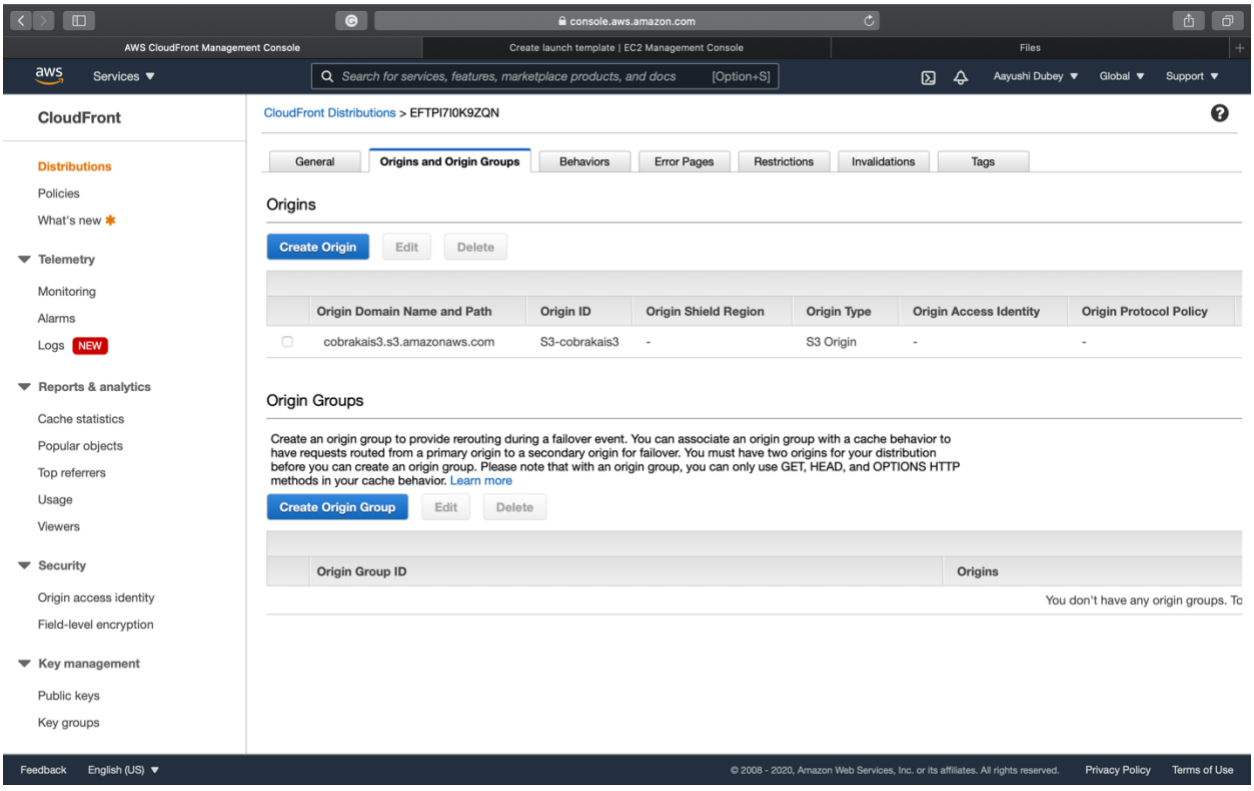


6. Now to configure S3 bucket with Content Delivery Network go to services and select CloudFront.
7. Click on Create Distribution and click on Get Started to create a Web distribution.
8. Select AWS S3 URL as Origin Domain Name.
9. Select HTTP and HTTPS for viewer protocol policy.
10. Select GET and HEAD for allowed HTTP methods.
11. Then click on Create Distribution.

Elastic Cache help makes it easy to host, run and maintain an in-memory cache into the cloud with minimal friction and also help in faster retrieval of information.

Elastic Cache supports two open-source in-memory caching engines: - Memcached and Redis. When a lot of people are queuing at the same time, it doesn't access the database again and again but creates cache memory to access it.

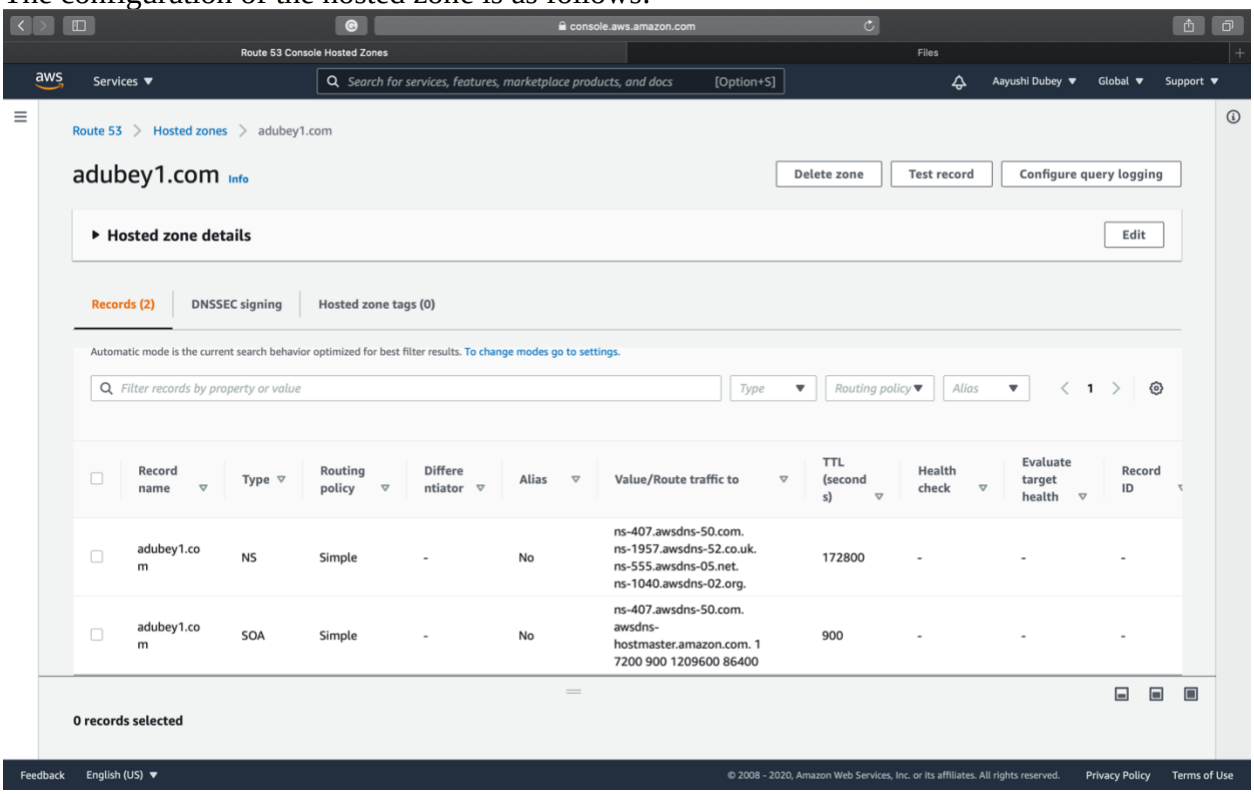

console.aws.amazon.com
AWS CloudFront Management Console
Create launch template | EC2 Management Console
Files
Services
Search for services, features, marketplace products, and docs [Option+S]
Aayushi Dubey
Global
Support
CloudFront
Distributions
Policies
What's new
Telemetry
Monitoring
Alarms
Logs NEW
Reports & analytics
Cache statistics
Popular objects
Top referrers
Usage
Viewers
Security
Origin access identity
Field-level encryption
Key management
Public keys
Key groups
CloudFront Distributions > EFTPI7I0K9ZQN
General
Origins and Origin Groups
Behaviors
Error Pages
Restrictions
Invalidations
Tags
Edit
Distribution ID EFTPI7I0K9ZQN
ARN arn:aws:cloudfront::082162535546:distribution/EFTPI7I0K9ZQN
Log Prefix -
Delivery Method Web
Cookie Logging Off
Distribution Status Deployed
Comment -
Price Class Use All Edge Locations (Best Performance)
AWS WAF Web ACL -
State Enabled
Alternate Domain Names (CNAMEs) -
SSL Certificate Default CloudFront Certificate (*.cloudfront.net)
Domain Name d3lk4ldduex3b7.cloudfront.net
Custom SSL Client Support -
Security Policy TLSv1
Supported HTTP Versions HTTP/2, HTTP/1.1, HTTP/1.0
IPv6 Enabled
Default Root Object -
Last Modified 2020-12-19 03:01 UTC+5:30
Log Bucket -
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


console.aws.amazon.com
AWS CloudFront Management Console
Create launch template | EC2 Management Console
Files
Services
Search for services, features, marketplace products, and docs [Option+S]
Aayushi Dubey
Global
Support
CloudFront
Distributions
Policies
What's new
Telemetry
Monitoring
Alarms
Logs NEW
Reports & analytics
Cache statistics
Popular objects
Top referrers
Usage
Viewers
Security
Origin access identity
Field-level encryption
Key management
Public keys
Key groups
CloudFront Distributions > EFTPI7I0K9ZQN
General
Origins and Origin Groups
Behaviors
Error Pages
Restrictions
Invalidations
Tags
Origins
Create Origin
Edit
Delete
Origin Domain Name and Path
Origin ID
Origin Shield Region
Origin Type
Origin Access Identity
Origin Protocol Policy
cobrakais3.s3.amazonaws.com
S3-cobrakais3
-
S3 Origin
-
-
Origin Groups
Create an origin group to provide rerouting during a failover event. You can associate an origin group with a cache behavior to have requests routed from a primary origin to a secondary origin for failover. You must have two origins for your distribution before you can create an origin group. Please note that with an origin group, you can only use GET, HEAD, and OPTIONS HTTP methods in your cache behavior. Learn more
Create Origin Group
Edit
Delete
Origin Group ID
Origins
You don't have any origin groups. To
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Additionally, we will configure route 53 to configure the hosted zones as well as the DNS server to host the website on.

Steps to configure Route 53 Hosted Zones are as follows:

1. Go to services and select Route 53.
2. Click on hosted zone and choose Create hosted zone.
3. Provide a DNS name as in this case we provided adubey1.com and a description to explain what purpose these hosted zone serves.
4. Now select the type of Hosted Zone and click create hosted zone
5. The configuration of the hosted zone is as follows:



5. **No AutoScaling.**

To prevent the issue of the Web Server and App server going down due to any hardware or software issues we will configure an AutoScaling Group which will automatically provisions new EC2 instances when load on one instance increases as well as when an instance goes down. The ASG either scales up that means can increase the number of instances or scales down the instance capacity which means decreasing the capacity in the AWS environment. Horizontal scaling means adding a greater number of instances or servers on the application when the load on one of the EC2 instance increase, and vertical scaling means to add more

capacity to the instances if there are only a limited number of instances present to use as servers. The application will face minimum downtime.

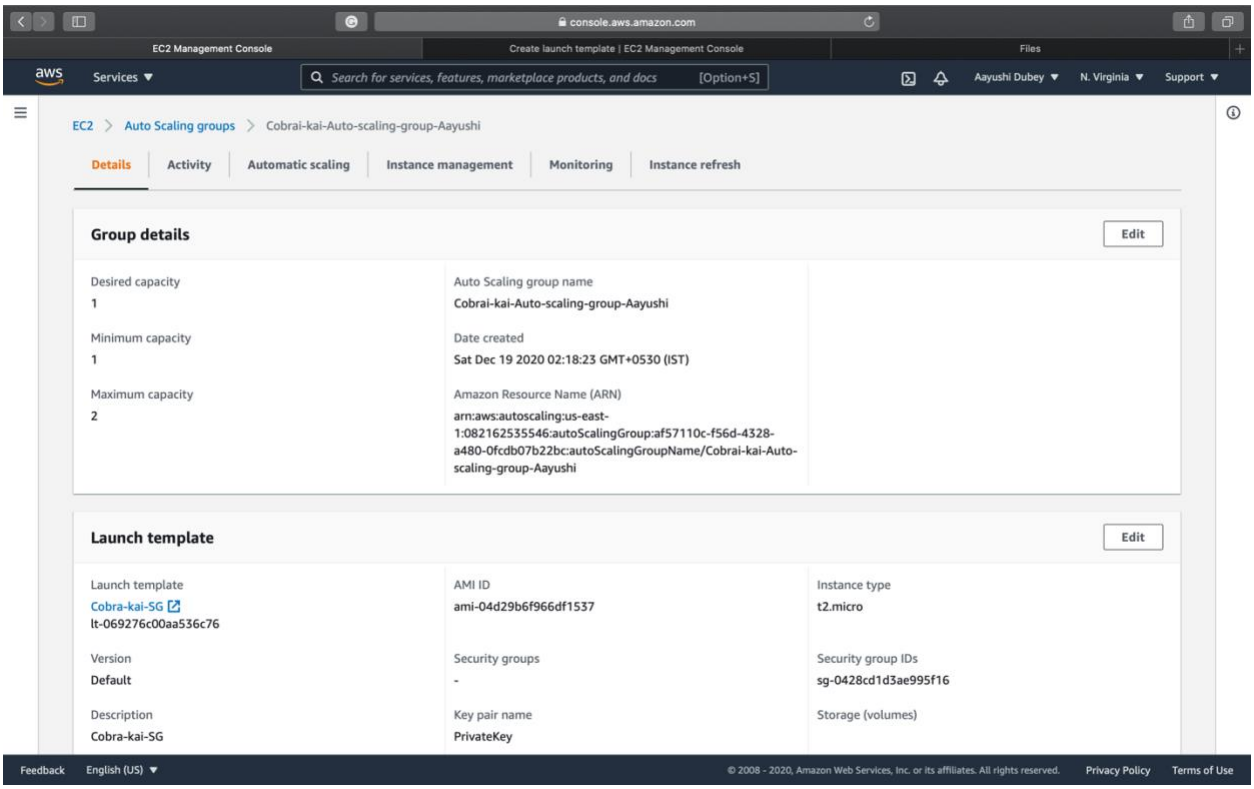
Steps to configure AutoScaling group with Load Balancer are as follows:

1. Click on Services and select EC2.
2. On the left panel select Auto Scaling group.
3. Click on create an Auto Scaling Group and provide a name.
4. Now click on Create a launch template to specify various configurations of an EC2 instance that need to be provisioned by the help of this Auto Scaling Group like instance type, Key pair, Networking platform.
5. Now select the template in the Launch template field.
6. Now select VPC and subnets in which we need to launch the EC2 instance.
7. Now Define the type of Load balancer which will get attached with the App and Webservers to distribute the incoming traffic to the instances.

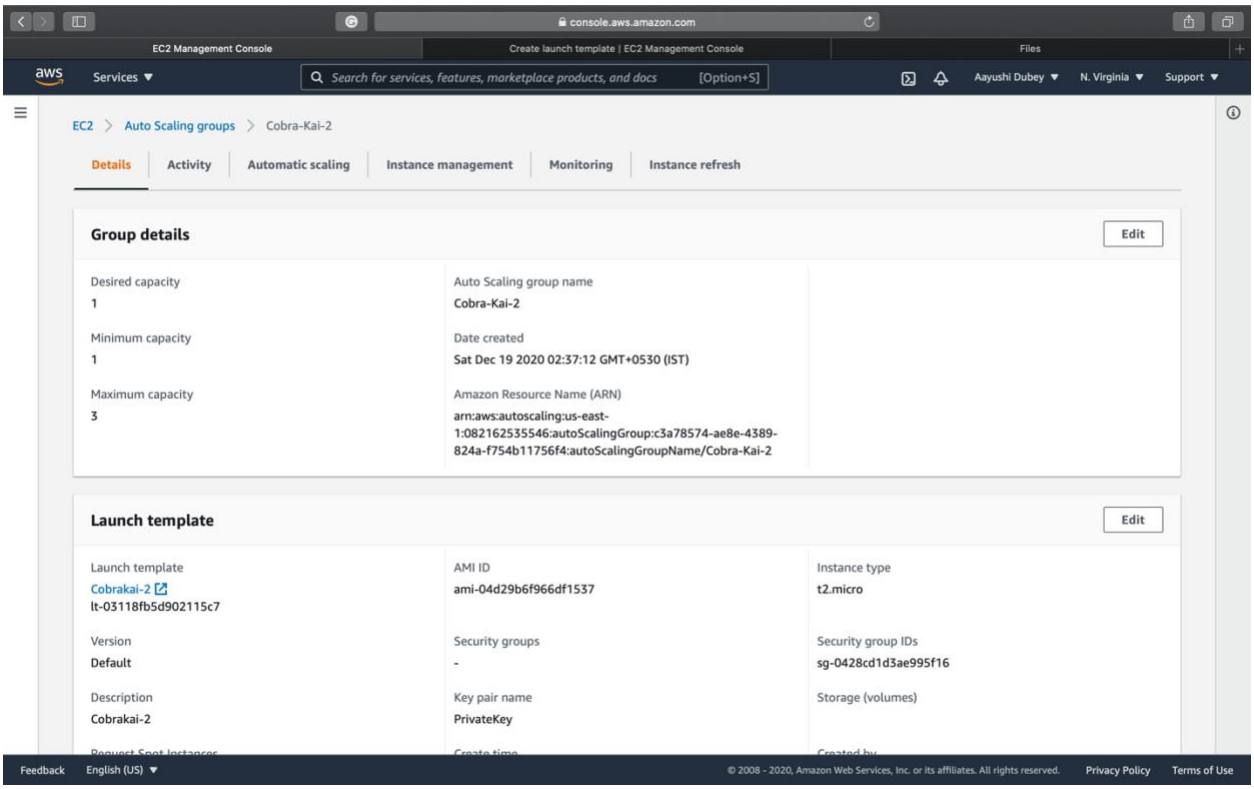
There are three type of load balancers that we can utilize here: Application Load Balancer which work with protocols like HTTP and HTTPS and operates at layer 7 of the OSI Model, Network Load balancer which works with protocols like TCP, UPD etc. and which works at layer 4 and Classic load balancer which works with both but misses out a lot of repositories as well as functionalities and soon will be no longer in use. These load balancers will help in distributing the incoming data traffic to various servers in order to decrease the load on one server and hence help in maintaining resiliency.

8. Then enter the desired capacity as 1, minimum capacity as 1 and maximum capacity for the app server as 3 and for the web server as 2.
9. The configured Auto Scaling Groups are shown below.

- Web Servers:

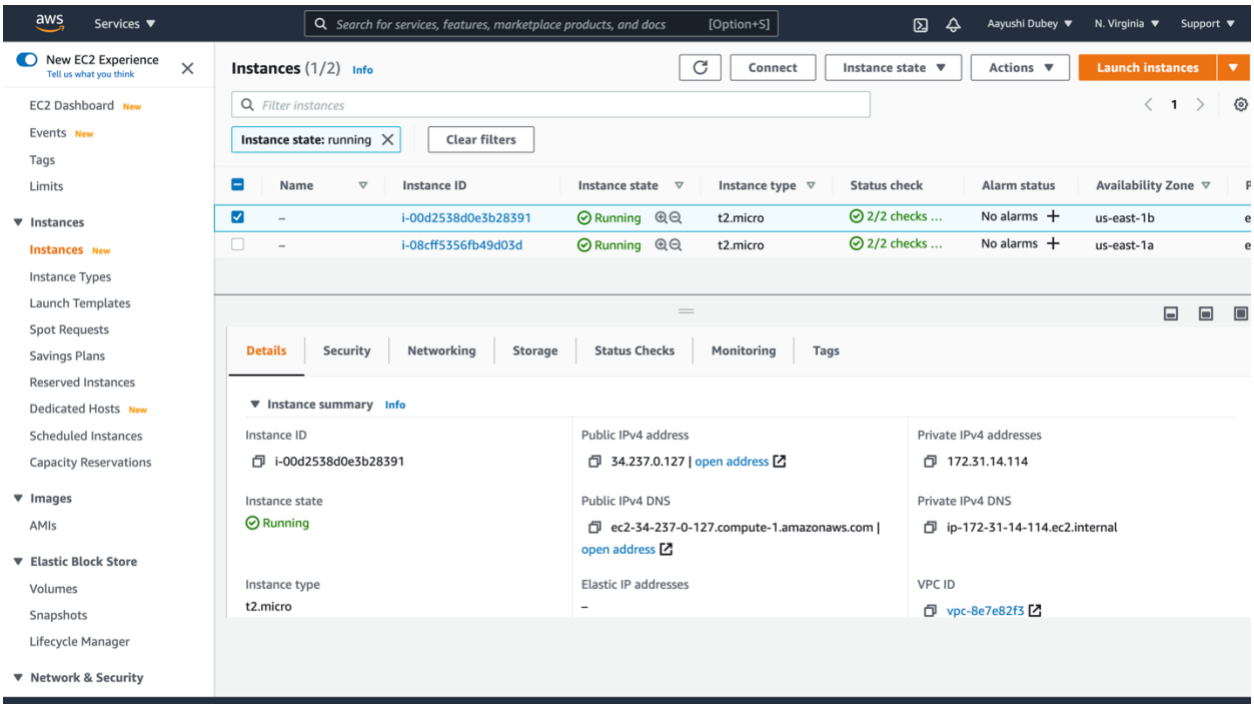

console.aws.amazon.com
EC2 Management Console
Create launch template | EC2 Management Console
Files
aws
Services
Search for services, features, marketplace products, and docs [Option+S]
Aayushi Dubey
N. Virginia
Support
EC2 > Auto Scaling groups > Cobrai-kai-Auto-scaling-group-Aayushi
Details
Activity
Automatic scaling
Instance management
Monitoring
Instance refresh
Group details
Edit
Desired capacity
1
Minimum capacity
1
Maximum capacity
2
Auto Scaling group name
Cobrai-kai-Auto-scaling-group-Aayushi
Date created
Sat Dec 19 2020 02:18:23 GMT+0530 (IST)
Amazon Resource Name (ARN)
arn:aws:autoscaling:us-east-1:082162535546:autoScalingGroup:af57110c-f56d-4328-a480-0fcdb07b22bc:autoScalingGroupName/Cobrai-kai-Auto-scaling-group-Aayushi
Launch template
Edit
Launch template
Cobra-kai-SG
lt-069276c00aa536c76
Version
Default
Description
Cobra-kai-SG
AMI ID
ami-04d29b6f966df1537
Security groups
-
Key pair name
PrivateKey
Instance type
t2.micro
Security group IDs
sg-0428cd1d3ae995f16
Storage (volumes)
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

- Web servers:

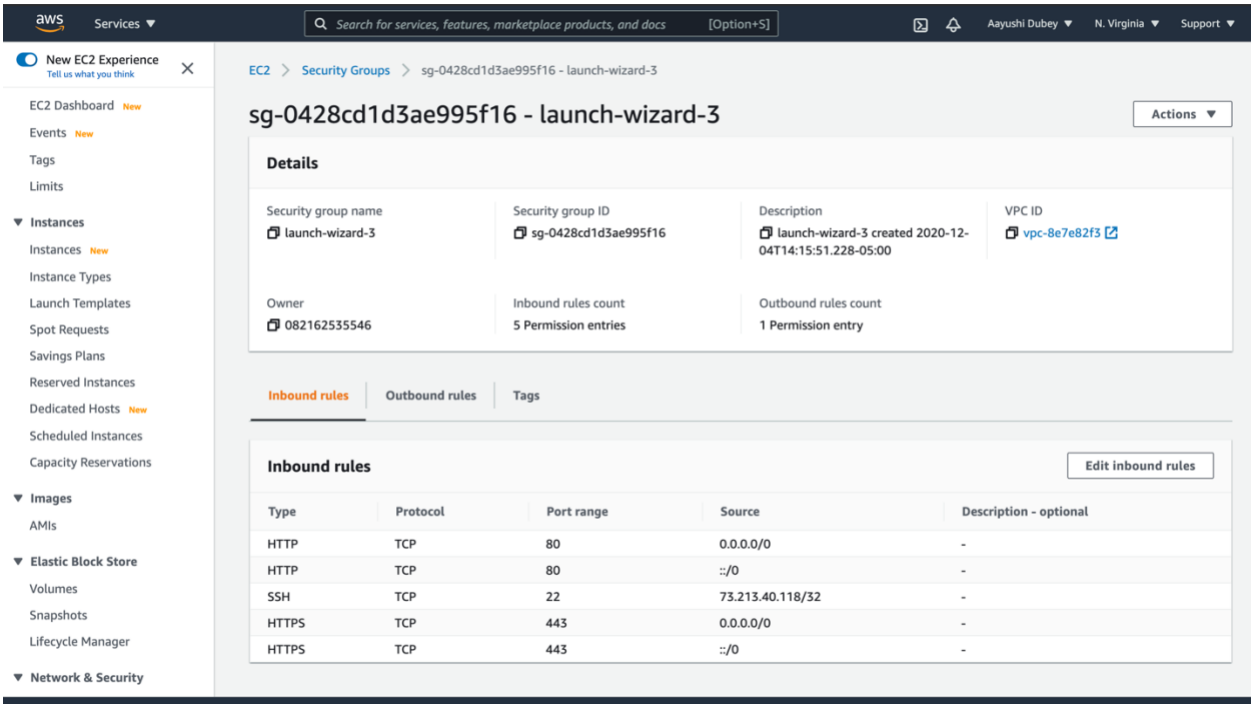


The created EC2 instance will have the following configurations as defined in the template for the Auto Scaling group.

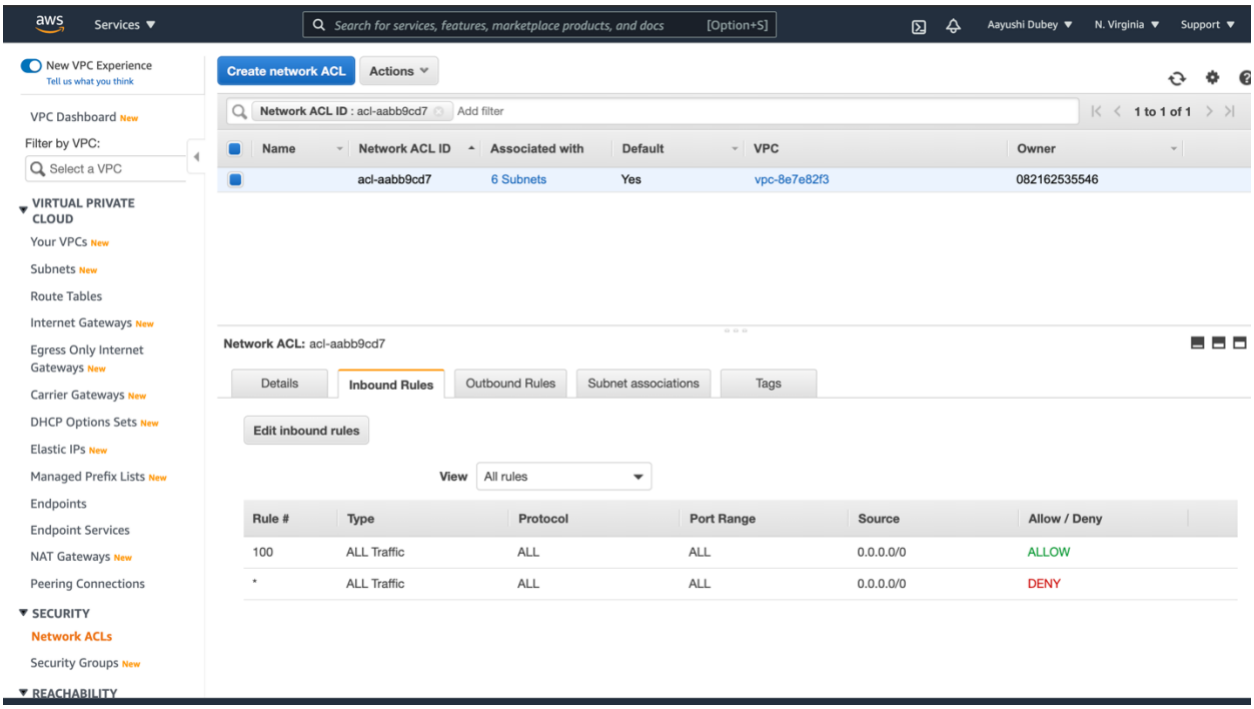
Details for the instances:



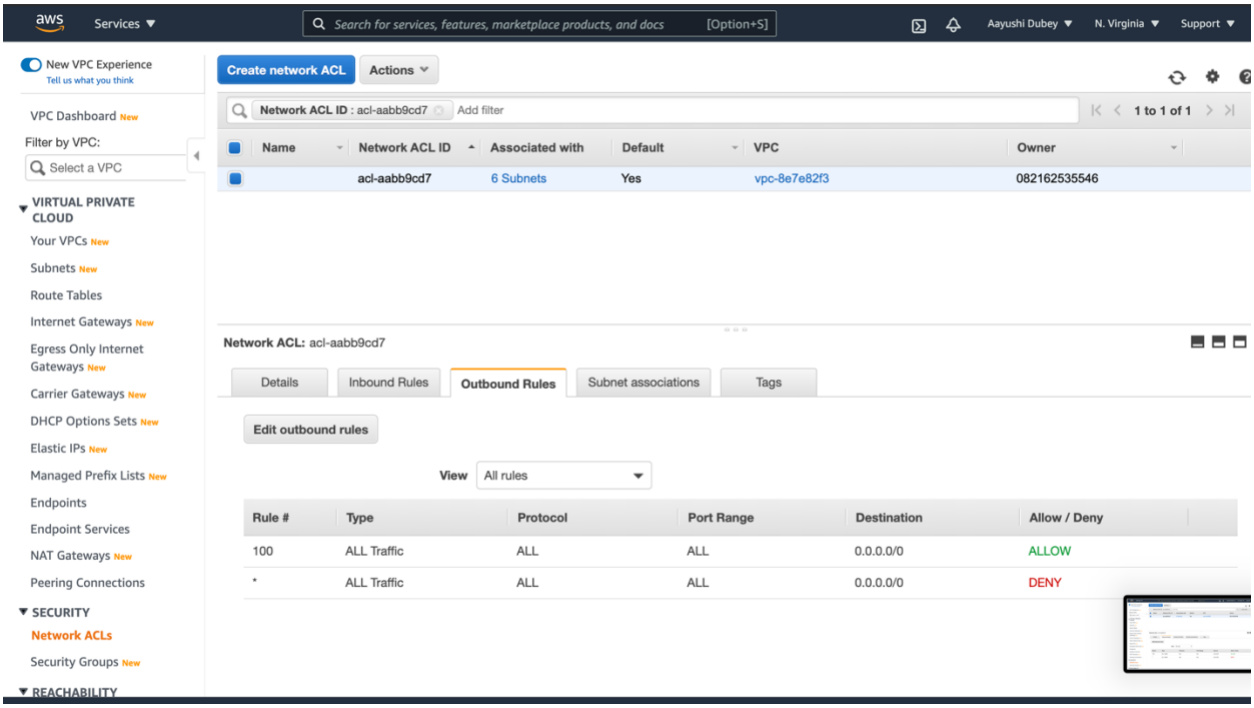
Security Group rules as defined for the connection from the internet as well as SSH connection to the instance:



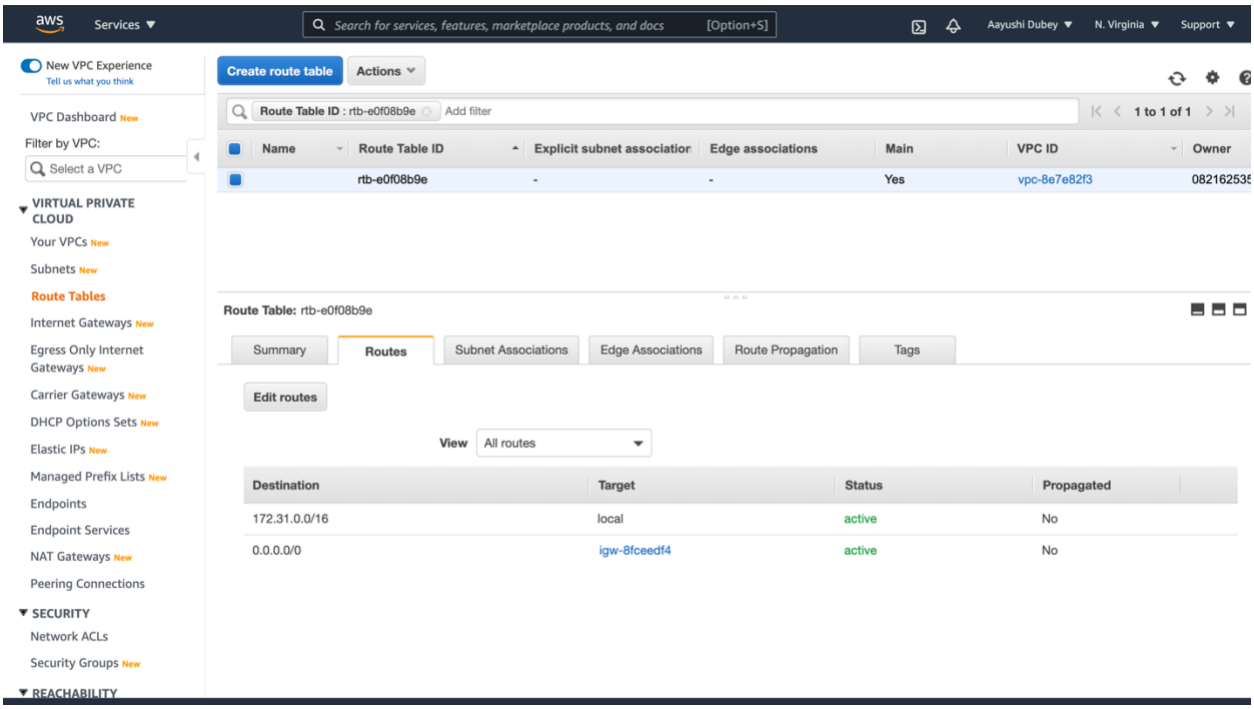
Inbound Rules for the Network ACLs for the instances are as follows:



Outbound Rules for the Network ACLs for the instances are as follows:



Configured route table for the EC2 instances with Internet Gateway configured:

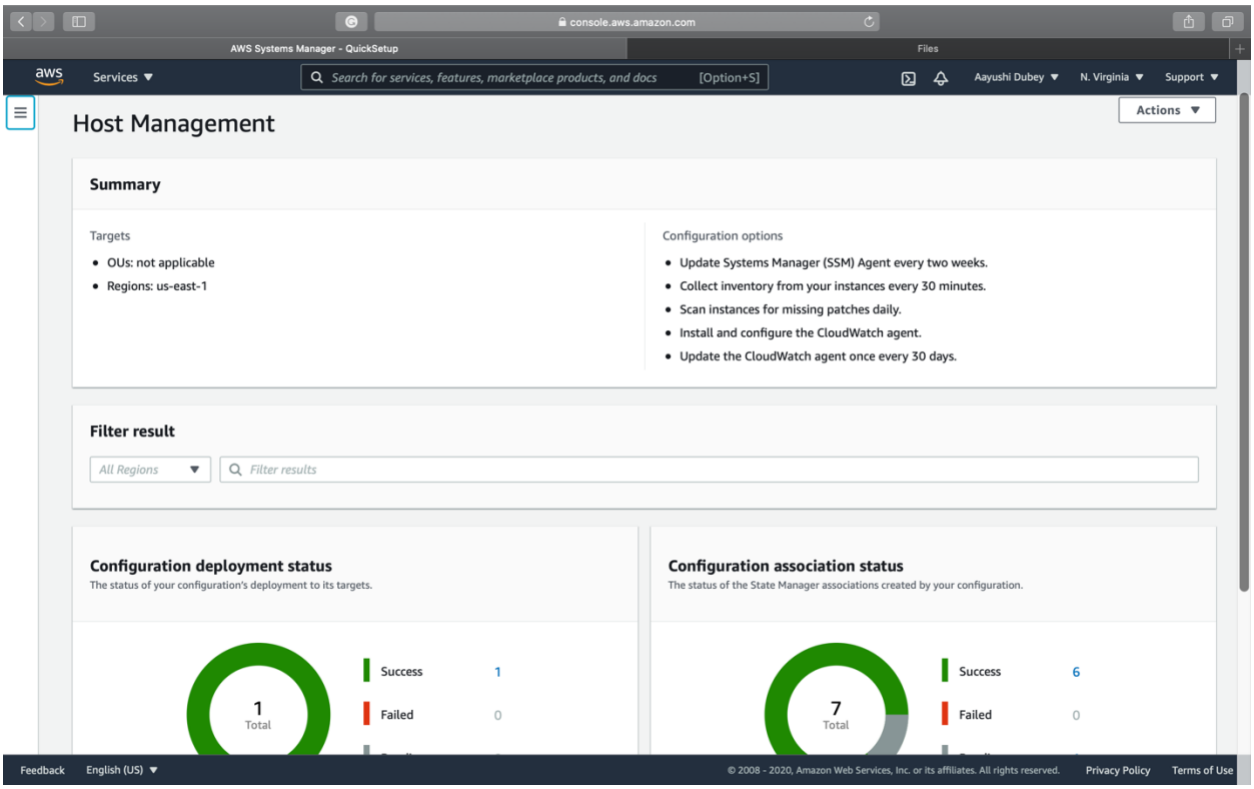


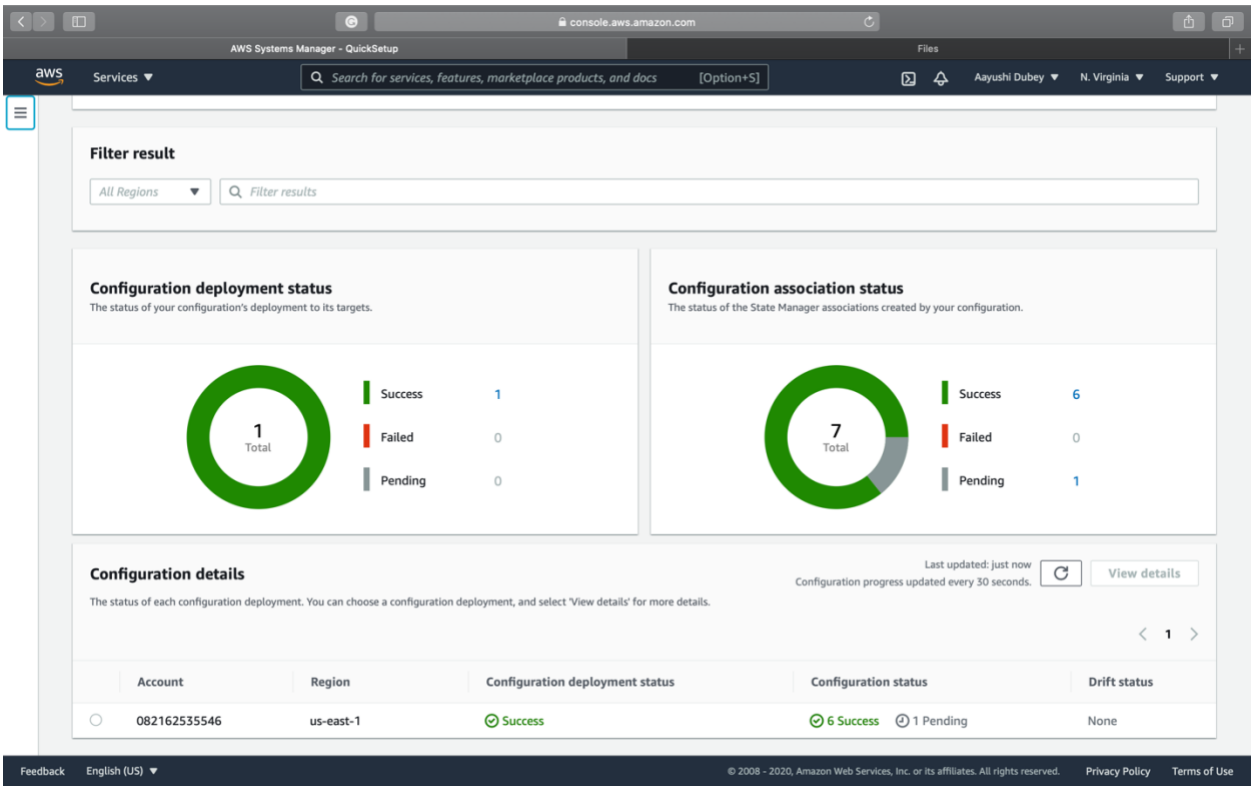
6. Patching strategy.

To configure the patching strategy to patch the deployed resources like the EC2 instances we will use AWS Systems Manager. AWS Systems Manager is a set of fully managed AWS services and capabilities. AWS Systems manager include Parameter store which provides secure and centralized storage for plain or encrypted text and separates data from the code and centralized it in single location to store data. Patching can be done using Patch manger which is one of the functionalities of Systems manager and which automates the process of patching managed instances with security related updates and can also patch fleets of EC2 and on-premises servers. This will help us to resolve the automatically patching issue.

Steps to configure AWS systems manager to automate the process of patching and automating various AWS documents:

1. Go to services and select Systems manager.
2. Now click on create and select host management.
3. Now select various configurations for Systems manager and Amazon CloudWatch.
4. Select target for the current region and target type as All Instances.
5. Click on create.
6. The configured patching strategy with configuration options and configuration deployment status are shown below:


console.aws.amazon.com
AWS Systems Manager - QuickSetup
Files
aws
Services
Search for services, features, marketplace products, and docs [Option+S]
Aayushi Dubey
N. Virginia
Support
Host Management
Actions
Summary
Targets
OUs: not applicable
Regions: us-east-1
Configuration options
Update Systems Manager (SSM) Agent every two weeks.
Collect inventory from your instances every 30 minutes.
Scan instances for missing patches daily.
Install and configure the CloudWatch agent.
Update the CloudWatch agent once every 30 days.
Filter result
All Regions
Filter results
Configuration deployment status
The status of your configuration's deployment to its targets.
1
Total
Success 1
Failed 0
Configuration association status
The status of the State Manager associations created by your configuration.
7
Total
Success 6
Failed 0
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

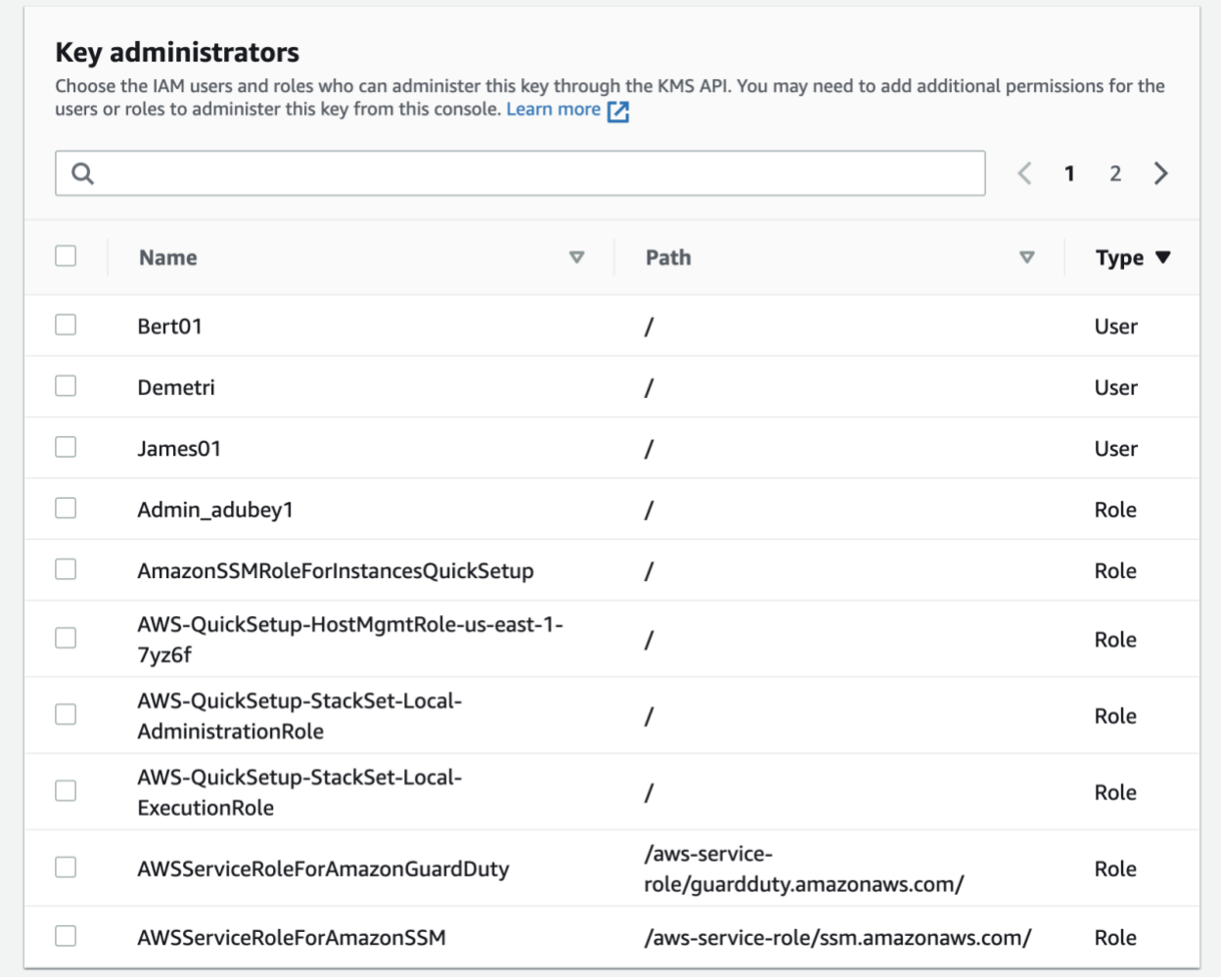

console.aws.amazon.com
AWS Systems Manager - QuickSetup
Files
aws
Services
Search for services, features, marketplace products, and docs [Option+S]
Aayushi Dubey
N. Virginia
Support
Filter result
All Regions
Filter results
Configuration deployment status
The status of your configuration's deployment to its targets.
1
Total
Success 1
Failed 0
Pending 0
Configuration association status
The status of the State Manager associations created by your configuration.
7
Total
Success 6
Failed 0
Pending 1
Configuration details
The status of each configuration deployment. You can choose a configuration deployment, and select 'View details' for more details.
Last updated: just now
Configuration progress updated every 30 seconds.
View details
1
Account
Region
Configuration deployment status
Configuration status
Drift status
082162535546
us-east-1
Success
6 Success
1 Pending
None
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

7. **Key Management System.**

This feature is used to encrypt the password and other confidential information that a user possesses while accessing the website like user history, credit card details which complies with PCI DSS compliance, passwords and various other details. KMS is used to both encrypt and decrypt the data stored in the S3 bucket as well as also helps in securing various other data that can be presented to the attackers for exploitation.

Steps to configure Key Management System are as follows:

1. Go to Services and click on KMS.
2. Now click on create key.
3. Now select the type of key i.e., either symmetric or asymmetric.
4. Now click on next and provide a name and a description for the KMS key.
5. Now Choose the IAM users and roles who can administer this key through the KMS API.



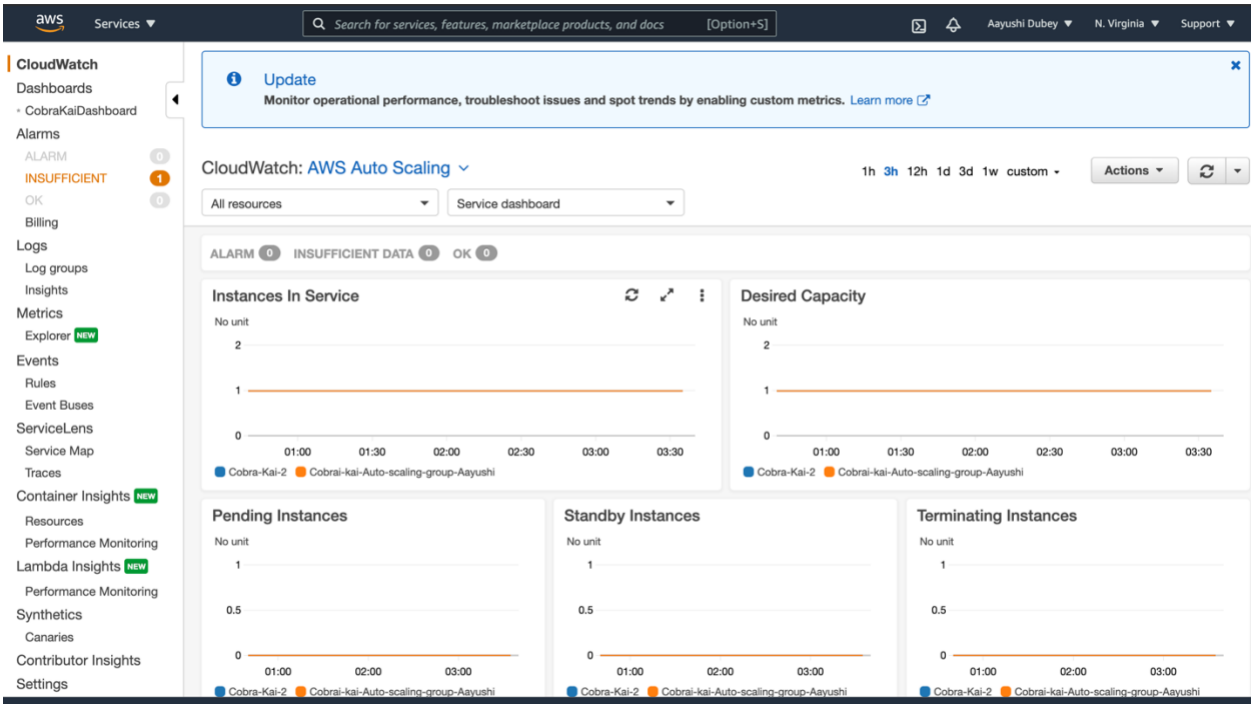
6. Now define key usage permissions for the above users and roles.
7. Click next, Review the policy and click on finish.

8. This will create the KMS key which will be used to encrypt and decrypt the user's data and will help in keeping it safely.

8. Cloud Watch

Cloud Watch is used for Monitoring and alerting the AWS system on user's account on various activities that happens on different AWS resources. Cloud watch logs can be easily and effectively used to troubleshoot issues related with EC2 instances. Cloud watch alarms can be used to perform various actions like SNS (for pushing a service out), for auto scaling policies and for various EC2 actions like launching and terminating an instance. Furthermore, Cloud Watch can be used to schedule automated builds of various templates, storing cloud watch logs to the S3 bucket, creating automated snapshots etc.

The CloudWatch dashboard for the previous configured AWS Auto Scaling is shown below:



9. Cloud Trail

Cloud trail is used to record different types of API calls made to different AWS resources including initiating, termination as well as modification of various AWS resources. AWS CloudTrail is used for auditing the tasks performed on these resources as well as also lists the user or the processes that makes the API call. Whenever an action fails to execute on an AWS service than CloudTrail can help determine what caused the API call to fail and what

troubleshooting actions can be performed to resolve the issue can be found easily by looking at the error codes.

The CloudTrail dashboard for the AWS console can be seen below: