“TEXTVAULT”

A CLOUD COMPUTING PROJECT REPORT

Submitted by

ADITHYA NAYAK (2023H1030092H)

ROHIT RAJ (2023H1030096H)

SHREYANK DUBEY (2023H1030081H)

JIGAR PATEL (2023H1030076H)

Under the guidance of

Dr. Subhrakanta Panda

Table Of Contents

Abstract

Our Project aims to provide users with a platform for creating and sharing notes online with custom expiry time. Users can also generate short links for their notes.This project also extends its functionality by offering an API.

Introduction

In the digital age, efficient communication and information sharing are paramount.

This project addresses this need by introducing a user-friendly platform for creating, sharing, and managing notes online.

Users can easily generate unique links for their notes, streamlining the sharing process. Moreover, recognizing the importance of seamless access, the project includes an API, enabling others to interact with its features effortlessly.

As we delve into the subsequent sections of this report, we will explore the intricacies of note creation, the user interface design, the unique link generation process, and the strategic implementation of the API. This introduction sets the stage for a detailed exploration of a project designed to meet the dynamic demands of modern communication, fostering convenience, and collaboration.

Techstack

NodeJs : Node.js is used for scalable and efficient server-side JavaScript execution, enabling non-blocking, event-driven architecture for high-performance web applications.

MongoDB : MongoDB is used for flexible and scalable document-oriented storage, providing a schema-less database solution for handling diverse data types in a distributed environment.

ExpressJs : Express.js is used to simplify and expedite the development of web applications by providing a minimal and flexible Node.js framework for building robust and scalable server-side applications.

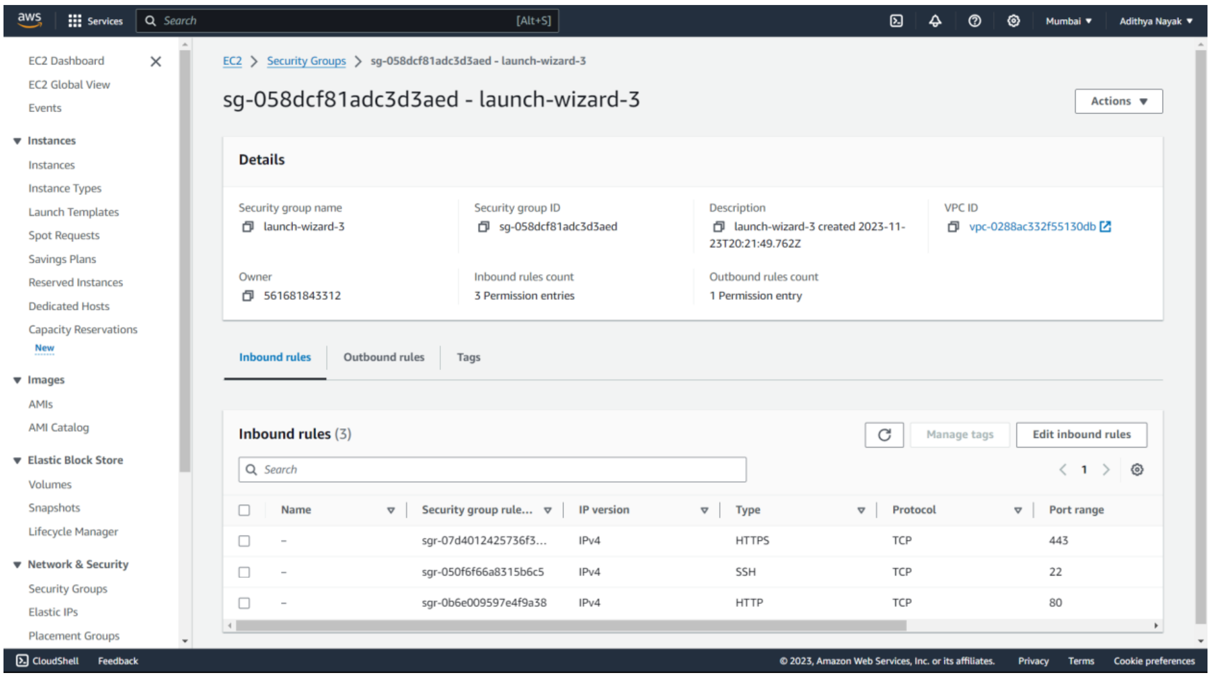
Html , Css , JavaScript : HTML, CSS, and JS are used together to create interactive and visually appealing web applications, with HTML for structure, CSS for styling, and JavaScript for dynamic behavior.

Git and Github : Git is used for version control, while GitHub serves as a platform for hosting and collaborating on Git repositories, facilitating efficient and collaborative software development.

Amazon Web Services : AWS is used for scalable and cost-effective cloud computing services, providing a wide range of infrastructure and tools to build, deploy, and manage applications.

Implementation

Step 1 : First we created a Security Group with the name "launch-wizard-3". Wich allows request from SSH,Http,Https services.



EC2 > Security Groups > sg-058dcf81adc3d3aed - launch-wizard-3

sg-058dcf81adc3d3aed - launch-wizard-3

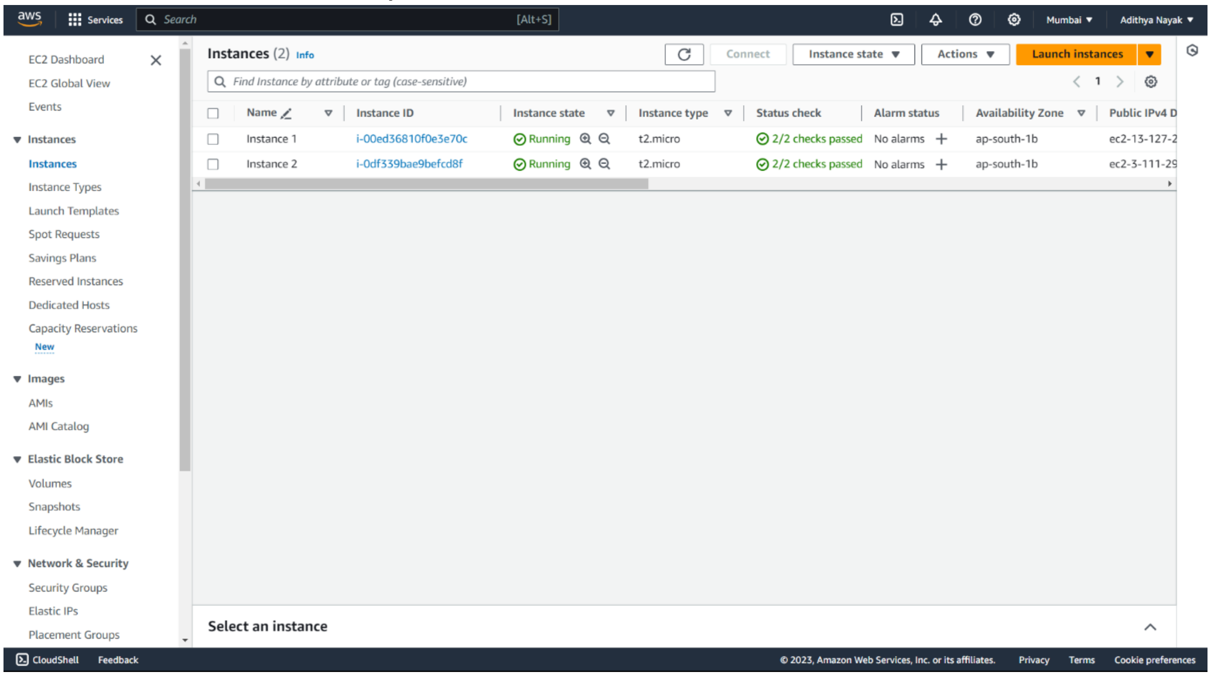
Details

Security group name launch-wizard-3	Security group ID sg-058dcf81adc3d3aed	Description launch-wizard-3 created 2023-11-23T20:21:49.762Z	VPC ID vpc-0288ac332f55130db
Owner 561681843312	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules (3)

Name	Security group rule...	IP version	Type	Protocol	Port range
–	sgr-07d4012425736f3...	IPv4	HTTPS	TCP	443
–	sgr-050f6f66a8315b6c5	IPv4	SSH	TCP	22
–	sgr-0b6e009597e4f9a38	IPv4	HTTP	TCP	80

Step 2 : We Created 2 instances under common security group that we created in first step.



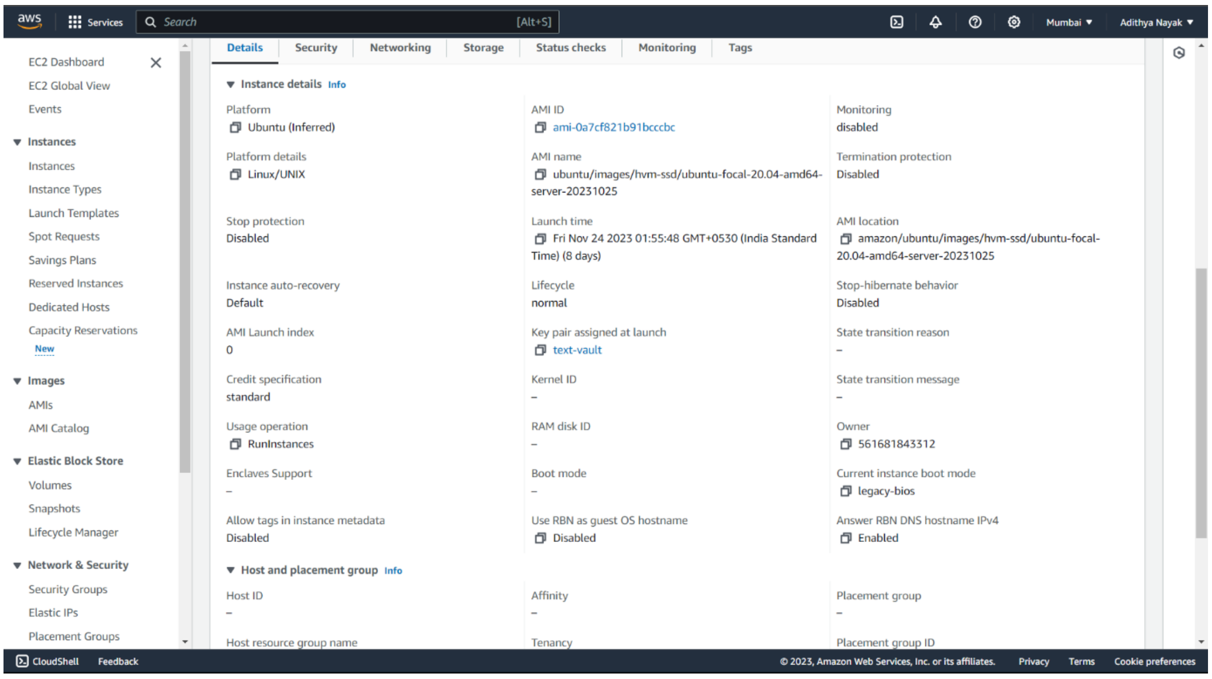
Instances (2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
Instance 1	i-00ed36810f0e3e70c	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-13-127-2
Instance 2	i-0df339bae9befcd8f	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-111-29

Select an instance

Instance details.

Link : Instance 1 and Instance 2



Details | Security | Networking | Storage | Status checks | Monitoring | Tags

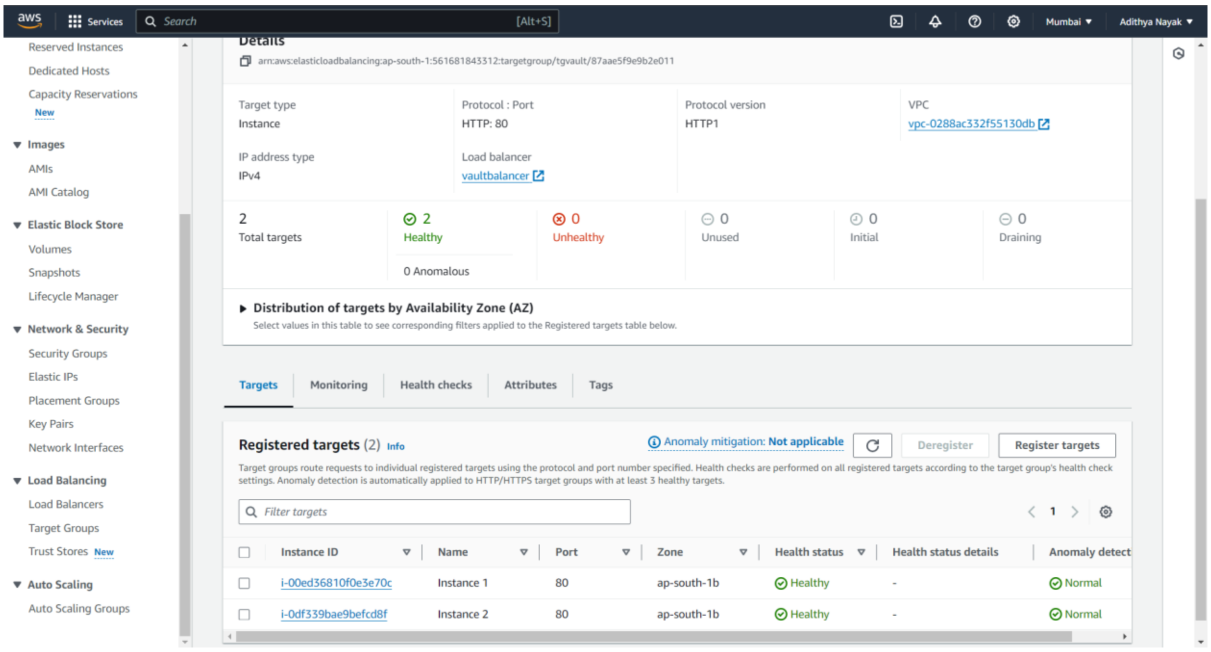
Instance details Info

Platform Ubuntu (Inferred)	AMI ID ami-0a7cf821b91bcccbc	Monitoring disabled
Platform details Linux/UNIX	AMI name ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-20231025	Termination protection Disabled
Stop protection Disabled	Launch time Fri Nov 24 2023 01:55:48 GMT+0530 (India Standard Time) (8 days)	AMI location amazon/ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-20231025
Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior Disabled
AMI Launch index 0	Key pair assigned at launch text-vault	State transition reason –
Credit specification standard	Kernel ID –	State transition message –
Usage operation RunInstances	RAM disk ID –	Owner 561681843312
Enclaves Support –	Boot mode –	Current instance boot mode legacy-bios
Allow tags in instance metadata Disabled	Use RBN as guest OS hostname Disabled	Answer RBN DNS hostname IPv4 Enabled

Host and placement group Info

Host ID –	Affinity –	Placement group –
Host resource group name	Tenancy	Placement group ID

Step 3 : We Created a target Group named "TGvault" for load balancer and then added 2 instances in the target group.



Details

arn:aws:elasticloadbalancing:ap-south-1:561681843312:targetgroup/tgvault/87aae5f9e9b2e011

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-0288ac332f55130db
IP address type IPv4	Load balancer vaultbalancer		

2 Total targets	2 Healthy 0 Anomalous	0 Unhealthy	0 Unused	0 Initial	0 Draining

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets | Monitoring | Health checks | Attributes | Tags

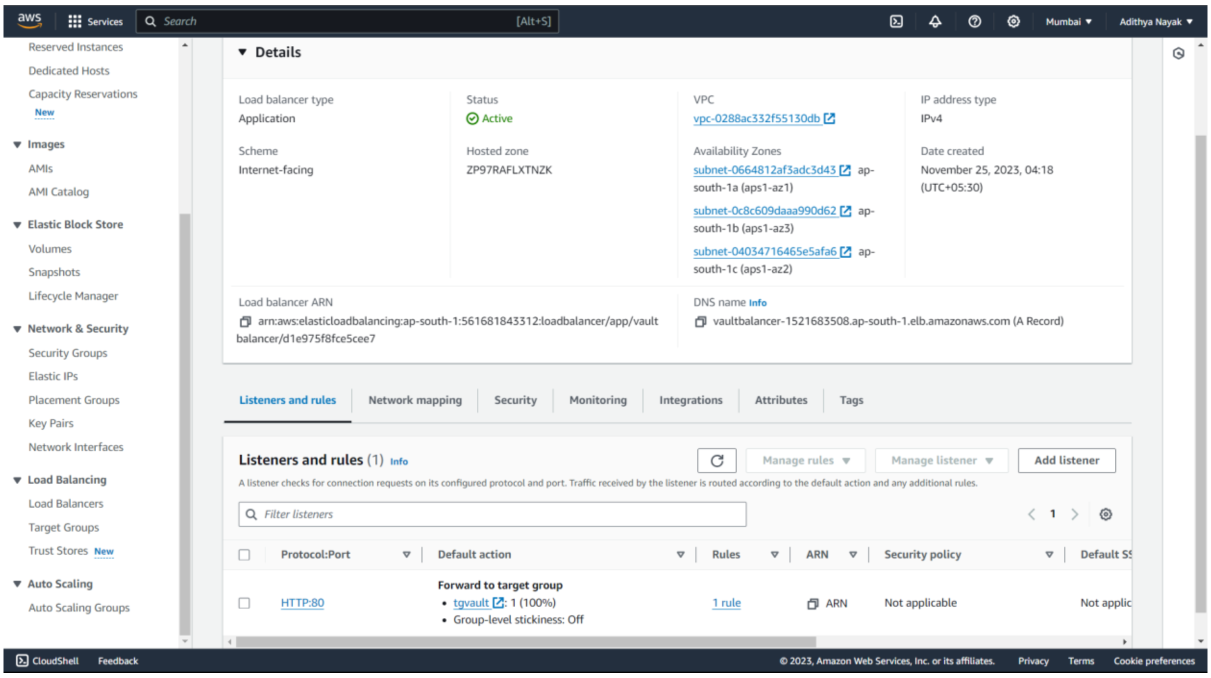
Registered targets (2) Info

Anomaly mitigation: Not applicable

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Instance ID	Name	Port	Zone	Health status	Health status details	Anomaly detect
i-00ed36810f0e3e70c	Instance 1	80	ap-south-1b	Healthy	-	Normal
i-0df339bae9befcd8f	Instance 2	80	ap-south-1b	Healthy	-	Normal

Step 4 : We Created a load balancer named “vaultbalancer”.



Details

Load balancer type: Application

Status: Active

VPC: vpc-0288ac332f55130db

IP address type: IPv4

Scheme: Internet-facing

Hosted zone: ZP97RAFLXTNZK

Availability Zones:
- subnet-0664812af3adc3d43 ap-south-1a (aps1-az1)
- subnet-0c8c609daaa990d62 ap-south-1b (aps1-az3)
- subnet-04034716465e5afa6 ap-south-1c (aps1-az2)

Date created: November 25, 2023, 04:18 (UTC+05:30)

Load balancer ARN: arn:aws:elasticloadbalancing:ap-south-1:561681843312:loadbalancer/app/vaultbalancer/d1e975f8fce5cee7

DNS name: vaultbalancer-1521683508.ap-south-1.elb.amazonaws.com (A Record)

Listeners and rules (1)

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Protocol:Port	Default action	Rules	ARN	Security policy	Default SS
HTTP:80	Forward to target group • tgvault: 1 (100%) • Group-level stickiness: Off	1 rule	ARN	Not applicable	Not applic

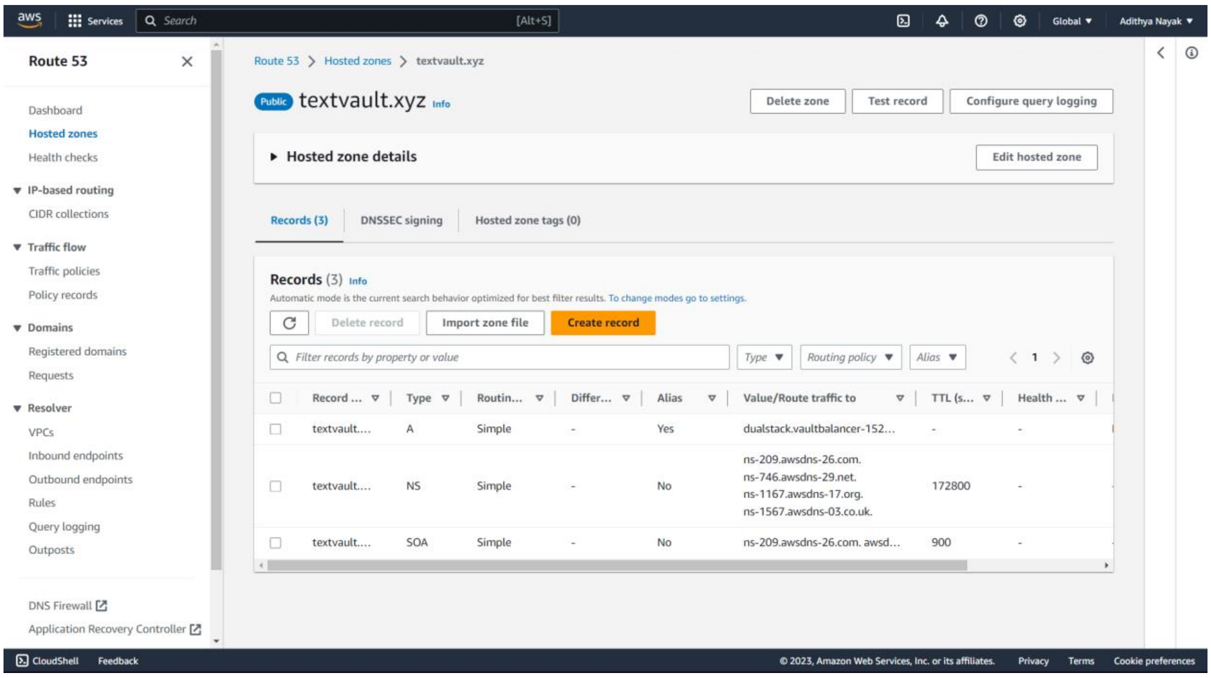
We can access our project through load balancers DNS link.

Link : Load Balancer

Step 5 : We Created a Public Hosted zone to get the Custom Nameserver . And then we bought a domain name “textvault.xyz” from GoDaddy.

After buying domain name from GoDaddy , we added the custom nameservers to domain which we got from hosted zone. This is how we connected our domain name to our project.

After that we created a records of type A in Route 53 which connects to Load balancer.


Route 53 > Hosted zones > textvault.xyz

textvault.xyz

Records (3)

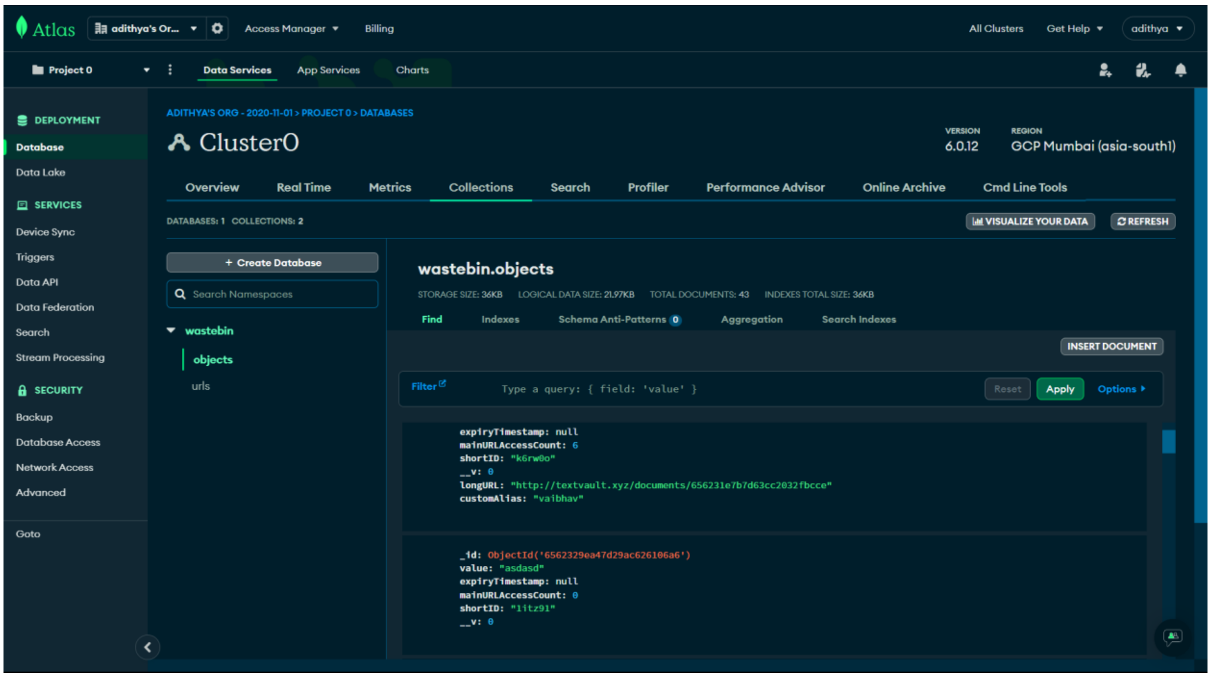
Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Record ...	Type	Routin...	Differ...	Alias	Value/Route traffic to	TTL (s...	Health ...
textvault....	A	Simple	-	Yes	dualstack.vaultbalancer-152...	-	-
textvault....	NS	Simple	-	No	ns-209.awsdns-26.com. ns-746.awsdns-29.net. ns-1167.awsdns-17.org. ns-1567.awsdns-03.co.uk.	172800	-
textvault....	SOA	Simple	-	No	ns-209.awsdns-26.com. awsd...	900	-

© 2023, Amazon Web Services, Inc. or its affiliates.

Step 6 : We cloned our github repository in both the instances. Then installed Nodejs , Npm , Mongodb , nginx , np2 module.

We connected our project to **MongoDB Atlas (Cloud Database)** using an API link that they had provided and implemented in our code.


ADITHYA'S ORG - 2020-11-01 > PROJECT 0 > DATABASES

Cluster0

VERSION 6.0.12 REGION GCP Mumbai (asia-south1)

DATABASES: 1 COLLECTIONS: 2

wastebin.objects

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 21.97KB TOTAL DOCUMENTS: 43 INDEXES TOTAL SIZE: 36KB

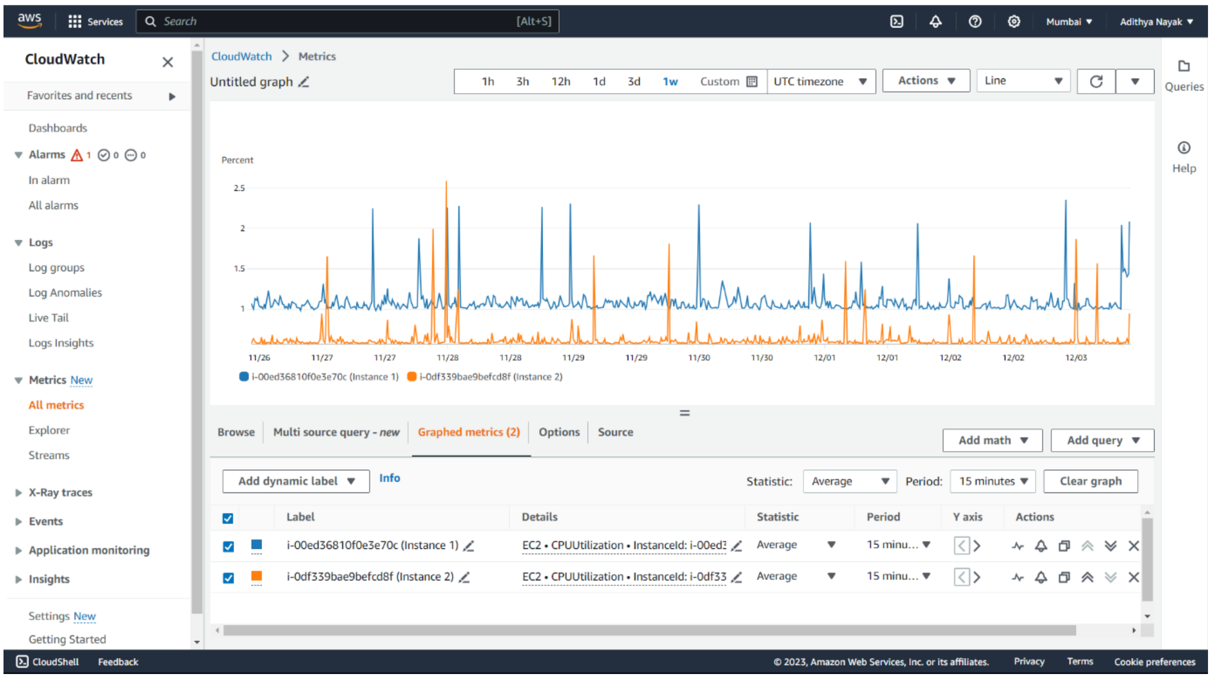
```
expiryTimestamp: null
mainURLAccessCount: 6
shortID: "k6rw0o"
__v: 0
longURL: "http://textvault.xyz/documents/656231e7b7d63cc2032fbcce"
customAlias: "vaibhav"
```

```
_id: ObjectId('6562329ea47d29ac626106a6')
value: "asdasd"
expiryTimestamp: null
mainURLAccessCount: 0
shortID: "1itz91"
__v: 0
```

Now our project was live : www.textvault.xyz

Step 7 : Now we implemented a **Cloud Watch** service in our instance. So that we can keep track of our instances and react accordingly .

Below image shows **CPU Utilization** of our instances.

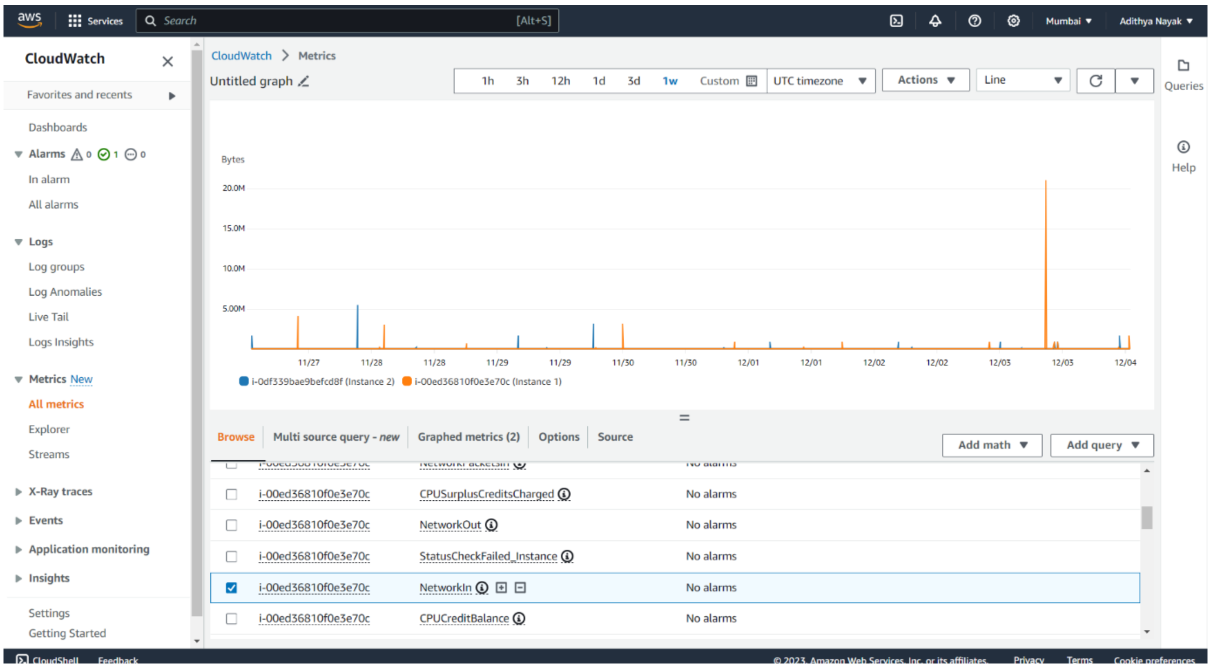


Add dynamic label ▼ Info

Statistic: Average ▼ Period: 15 minutes ▼ Clear graph

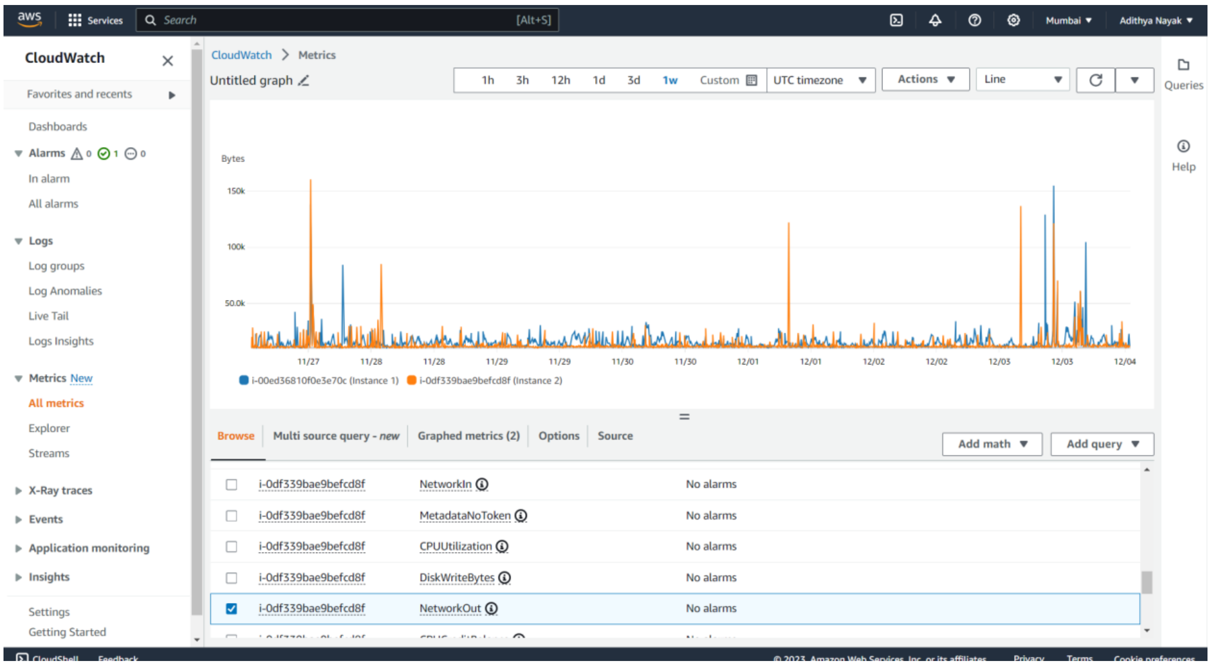
✓		Label	Details	Statistic	Period	Y axis	Actions
✓	🟦	i-00ed36810f0e3e70c (Instance 1)	EC2 • CPUUtilization • InstanceId: i-00ed3	Average	15 minu...		
✓	🟧	i-0df339bae9befcd8f (Instance 2)	EC2 • CPUUtilization • InstanceId: i-0df33	Average	15 minu...		

Below image shows **Bytes received** by instances.



☐	i-00ed36810f0e3e70c	CPUSurplusCreditsCharged	No alarms
☐	i-00ed36810f0e3e70c	NetworkOut	No alarms
☐	i-00ed36810f0e3e70c	StatusCheckFailed_Instance	No alarms
✓	i-00ed36810f0e3e70c	NetworkIn	No alarms
☐	i-00ed36810f0e3e70c	CPUCreditBalance	No alarms

Below image shows **Bytes sent** by instances.

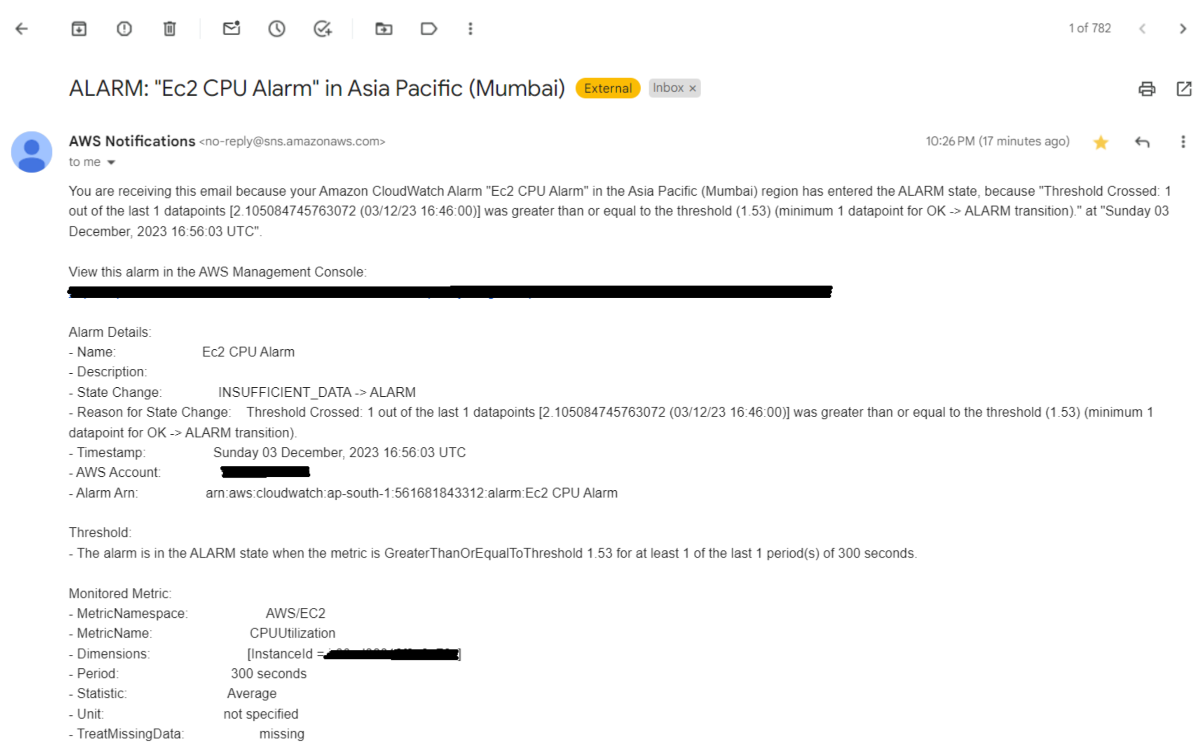


● i-00ed36810f0e3e70c (Instance 1) ● i-0df339bae9befcd8f (Instance 2)

☐	i-0df339bae9befcd8f	NetworkIn	No alarms
☐	i-0df339bae9befcd8f	MetadataNoToken	No alarms
☐	i-0df339bae9befcd8f	CPUUtilization	No alarms
☐	i-0df339bae9befcd8f	DiskWriteBytes	No alarms
☑	i-0df339bae9befcd8f	NetworkOut	No alarms

Step 8 : After a week of analysis we created a **Alarm in cloud watch**. Whenever **CPU Utilization goes beyond certain (In our case 1.53) threshold** , cloud watch sends an **Email** to us and can take certain predefined actions(like Stop , reboot , delete , auto scale instance.)

To test our implementation we kept the threshold low. After some time when threshold was reached , we got an Email with the detailed report of instance.



1 of 782

ALARM: "Ec2 CPU Alarm" in Asia Pacific (Mumbai) External Inbox ×

AWS Notifications <no-reply@sns.amazonaws.com>
to me

10:26 PM (17 minutes ago)

You are receiving this email because your Amazon CloudWatch Alarm "Ec2 CPU Alarm" in the Asia Pacific (Mumbai) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [2.105084745763072 (03/12/23 16:46:00)] was greater than or equal to the threshold (1.53) (minimum 1 datapoint for OK -> ALARM transition)." at "Sunday 03 December, 2023 16:56:03 UTC".

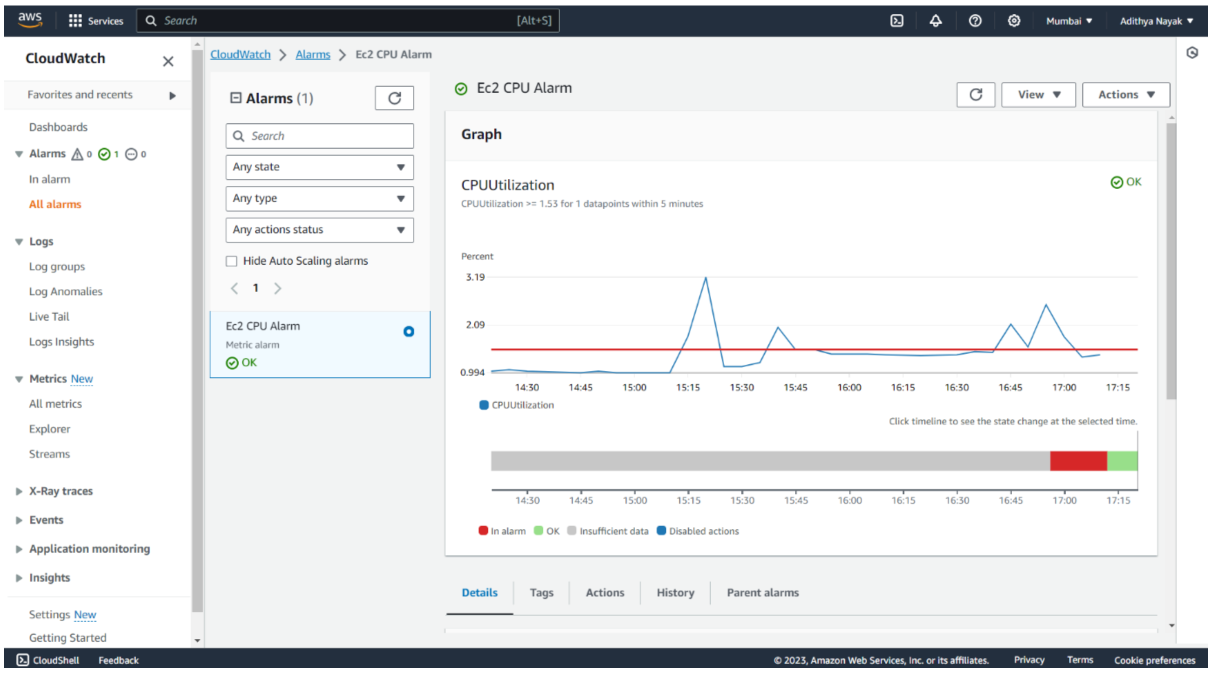
View this alarm in the AWS Management Console:
[redacted]

Alarm Details:
- Name: Ec2 CPU Alarm
- Description:
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [2.105084745763072 (03/12/23 16:46:00)] was greater than or equal to the threshold (1.53) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Sunday 03 December, 2023 16:56:03 UTC
- AWS Account: [redacted]
- Alarm Arn: arn:aws:cloudwatch:ap-south-1:561681843312:alarm:Ec2 CPU Alarm

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 1.53 for at least 1 of the last 1 period(s) of 300 seconds.

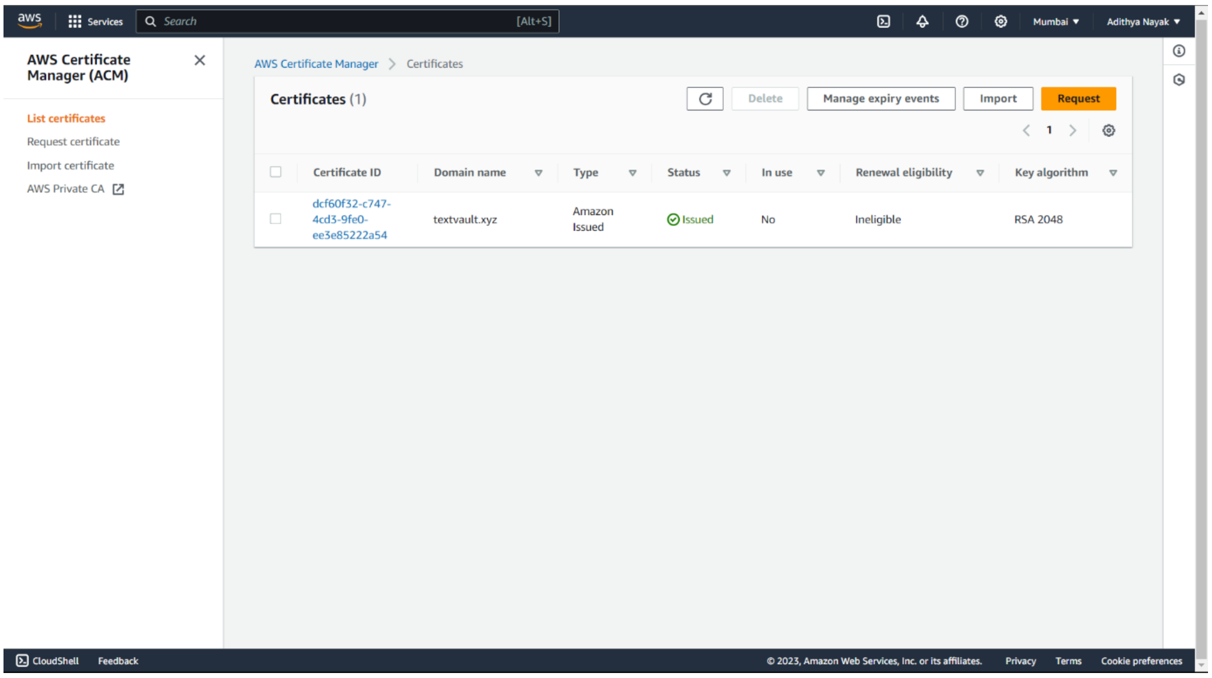
Monitored Metric:
- MetricNamespace: AWS/EC2
- MetricName: CPUUtilization
- Dimensions: [InstanceId = [redacted]]
- Period: 300 seconds
- Statistic: Average
- Unit: not specified
- TreatMissingData: missing

This is the Snapshot of system after the alarm situation.



Step 9 : After applying **Cloud watch ,** we used free **AWS Certificate manager** service to get **SSL certificate**. Using which we enabled HTTPS service in out site.

We created a certificate.


AWS Certificate Manager (ACM)

List certificates
Request certificate
Import certificate
AWS Private CA

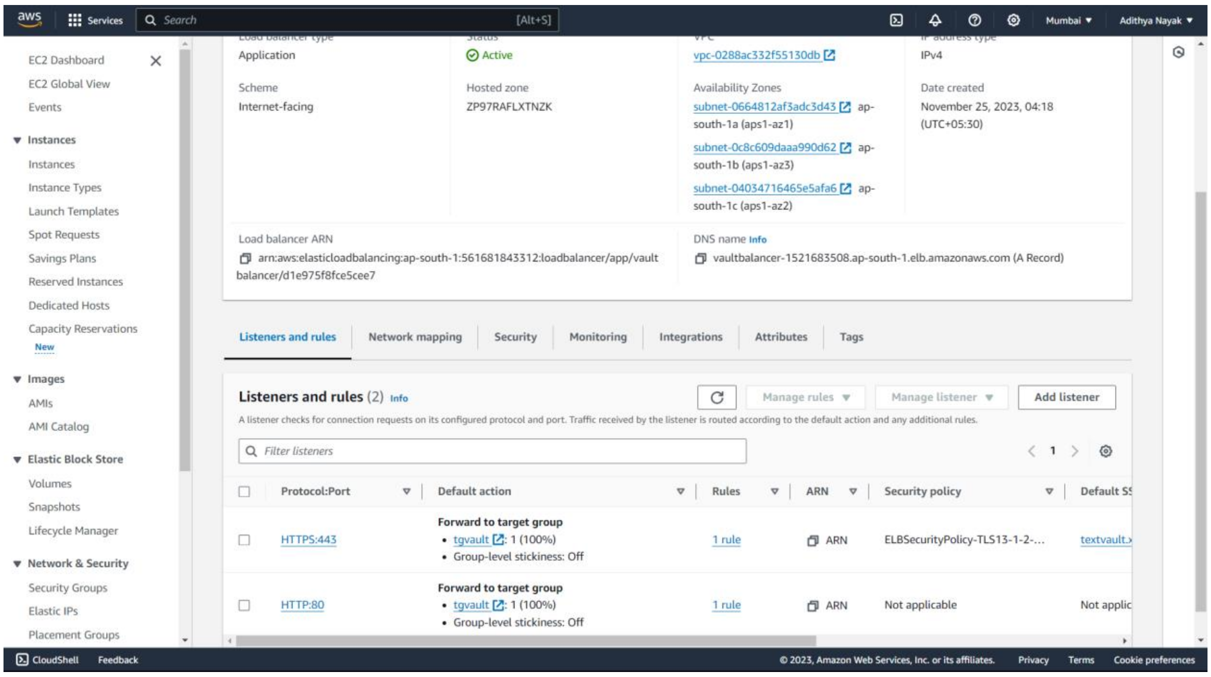
AWS Certificate Manager > Certificates

Certificates (1)

Certificate ID	Domain name	Type	Status	In use	Renewal eligibility	Key algorithm
dcf60f32-c747-4cd3-9fe0-ee3e85222a54	textvault.xyz	Amazon Issued	Issued	No	Ineligible	RSA 2048

© 2023, Amazon Web Services, Inc. or its affiliates.

Step 10 : We added the certificate in our load balancer's listners rules.


Load balancer type: Application
Status: Active
VPC: vpc-0288ac332f55130db
IP address type: IPv4
Scheme: Internet-facing
Hosted zone: ZP97RAFLXTNZK
Availability Zones: subnet-0664812af3adc3d43 ap-south-1a (aps1-az1); subnet-0c8c609daaa990d62 ap-south-1b (aps1-az3); subnet-04034716465e5afa6 ap-south-1c (aps1-az2)
Date created: November 25, 2023, 04:18 (UTC+05:30)
Load balancer ARN: arn:aws:elasticloadbalancing:ap-south-1:561681843312:loadbalancer/app/vault balancer/d1e975f8fce5cee7
DNS name: vaultbalancer-1521683508.ap-south-1.elb.amazonaws.com (A Record)

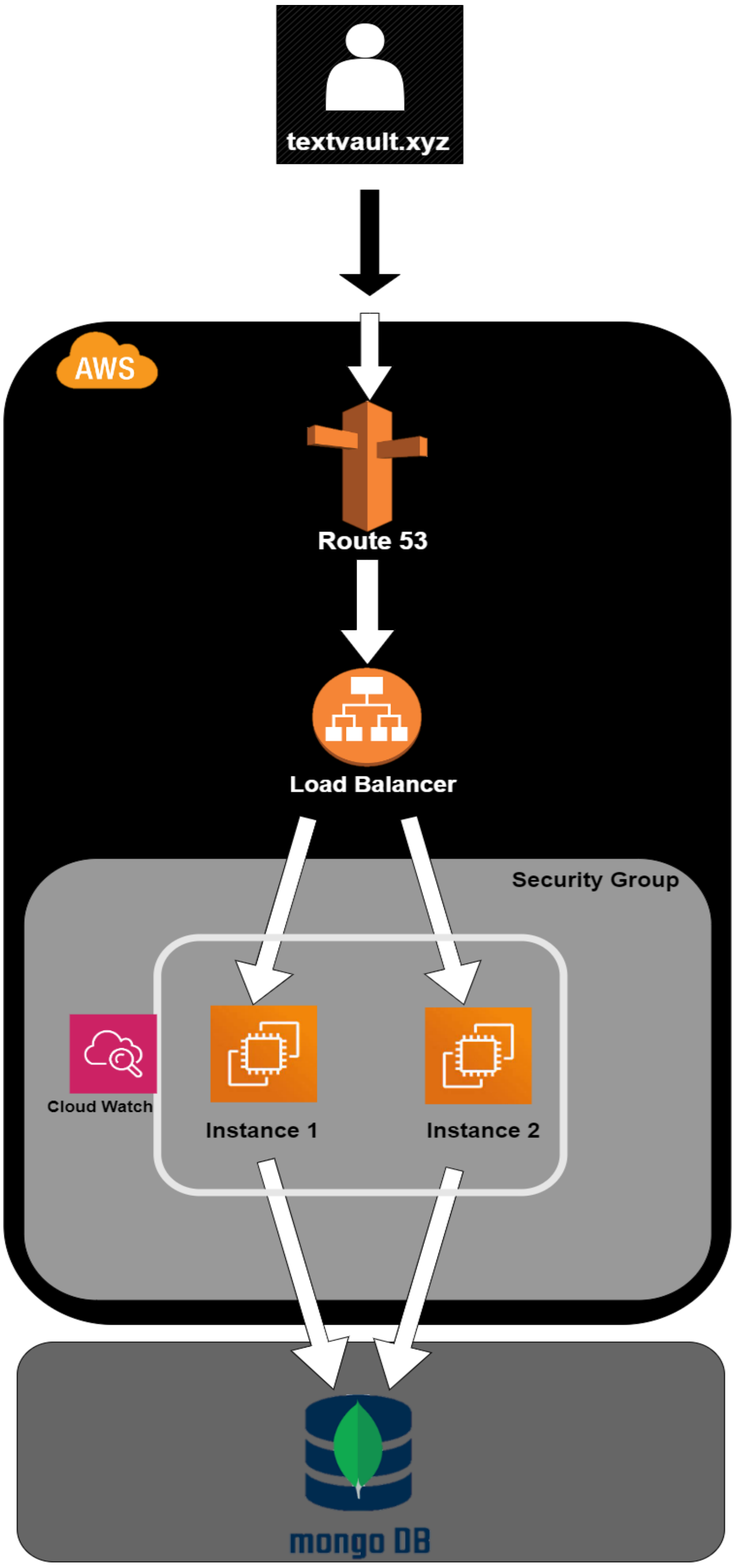
Listeners and rules (2)

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

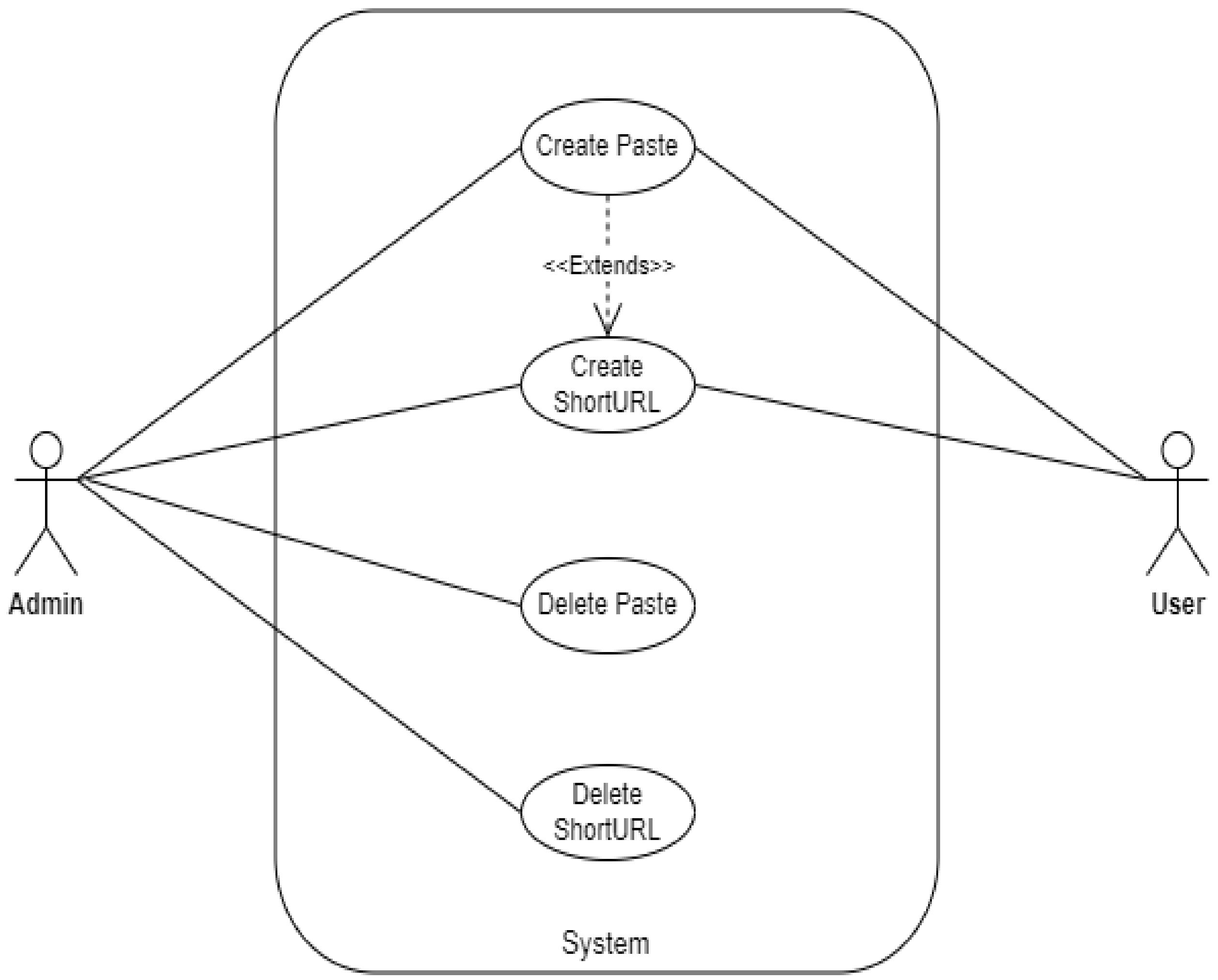
Protocol:Port	Default action	Rules	ARN	Security policy	Default SS
HTTPS:443	Forward to target group • tgvault: 1 (100%) • Group-level stickiness: Off	1 rule	ARN	ELBSecurityPolicy-TLS13-1-2-...	textvault.x
HTTP:80	Forward to target group • tgvault: 1 (100%) • Group-level stickiness: Off	1 rule	ARN	Not applicable	Not applic

© 2023, Amazon Web Services, Inc. or its affiliates.

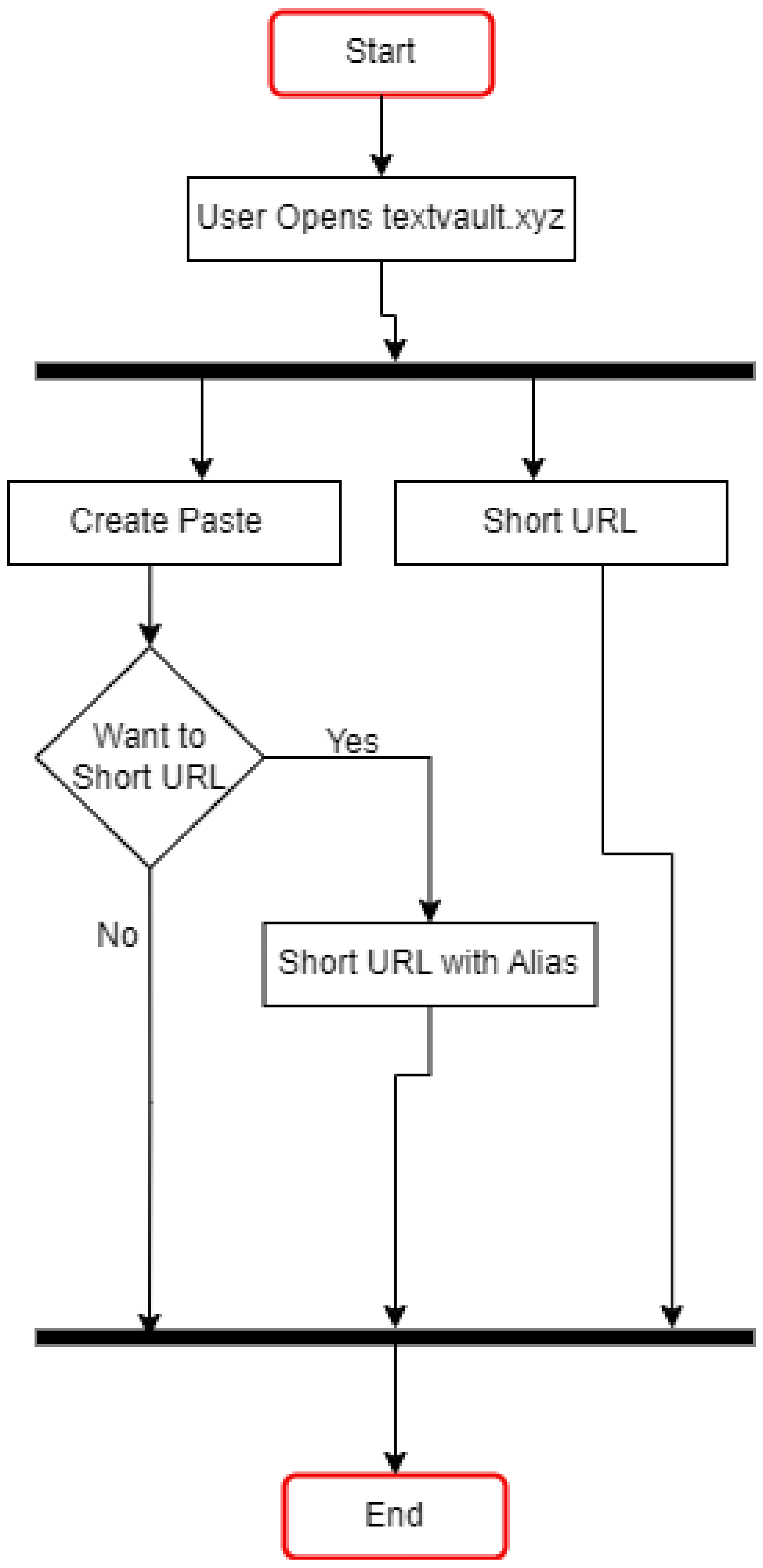
Implementation Diagram



Use Case Diagram



Activity Diagram

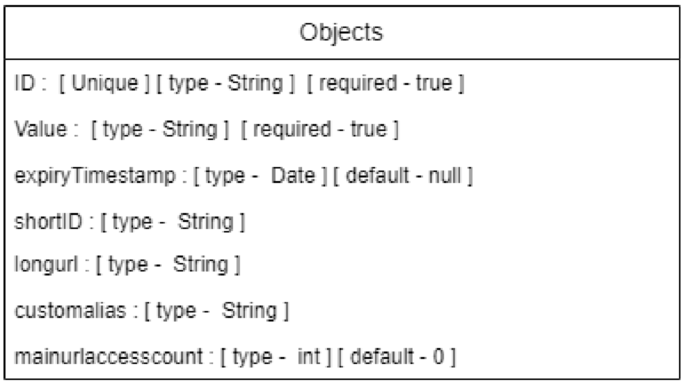


Data Dictionary

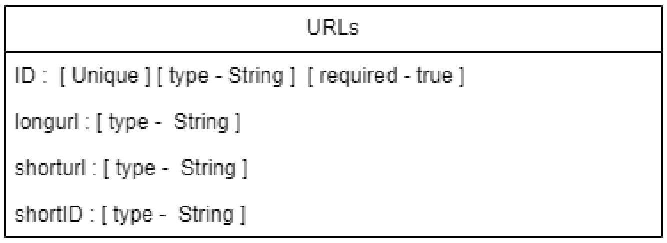
We have 2 Database in our Project

1) **Objects** : This contains paste , longurl , uniqueid of object , shortID(if generated short url) , customalias , url view count.

2) **Urls :** This database contains short url , long url (connected to short url) , shortID.

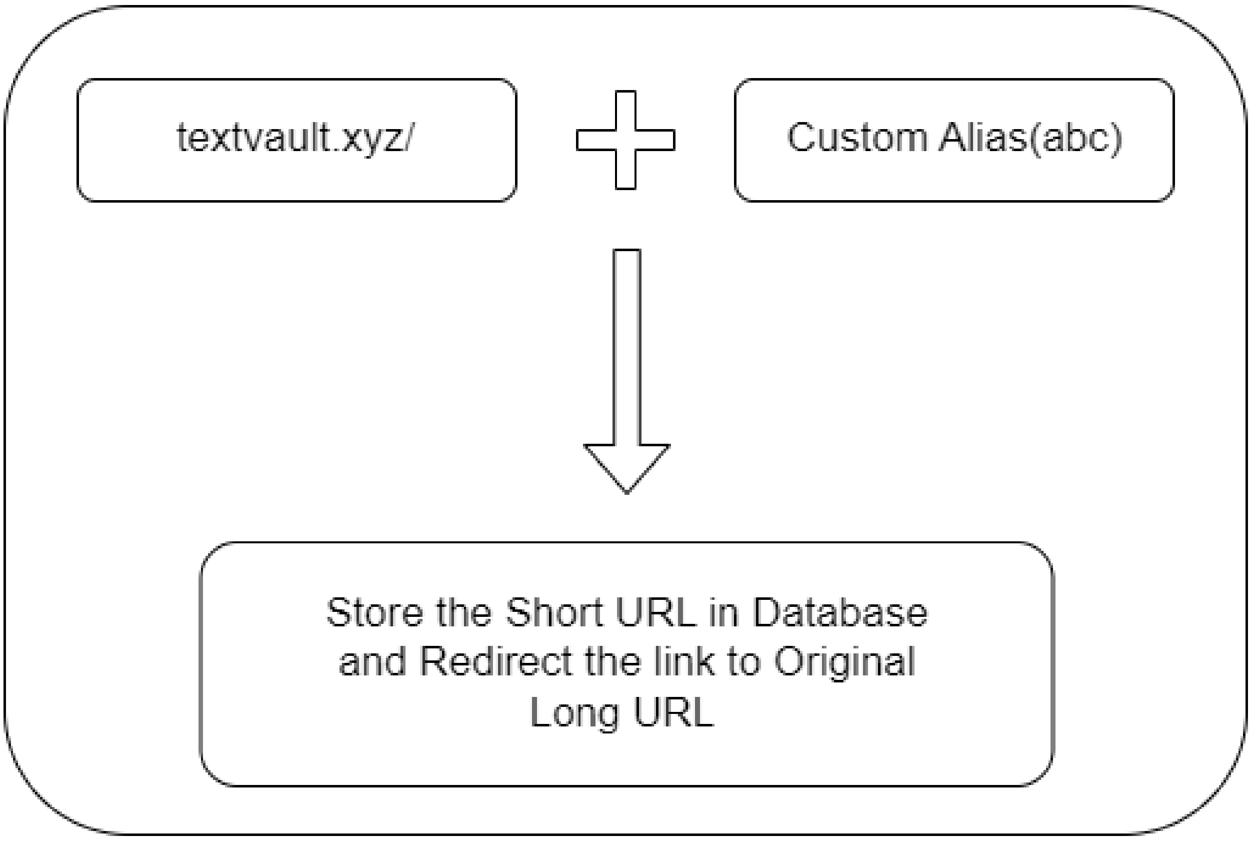


Objects
ID : [Unique] [type - String] [required - true]
Value : [type - String] [required - true]
expiryTimestamp : [type - Date] [default - null]
shortID : [type - String]
longurl : [type - String]
customalias : [type - String]
mainurlaccesscount : [type - int] [default - 0]



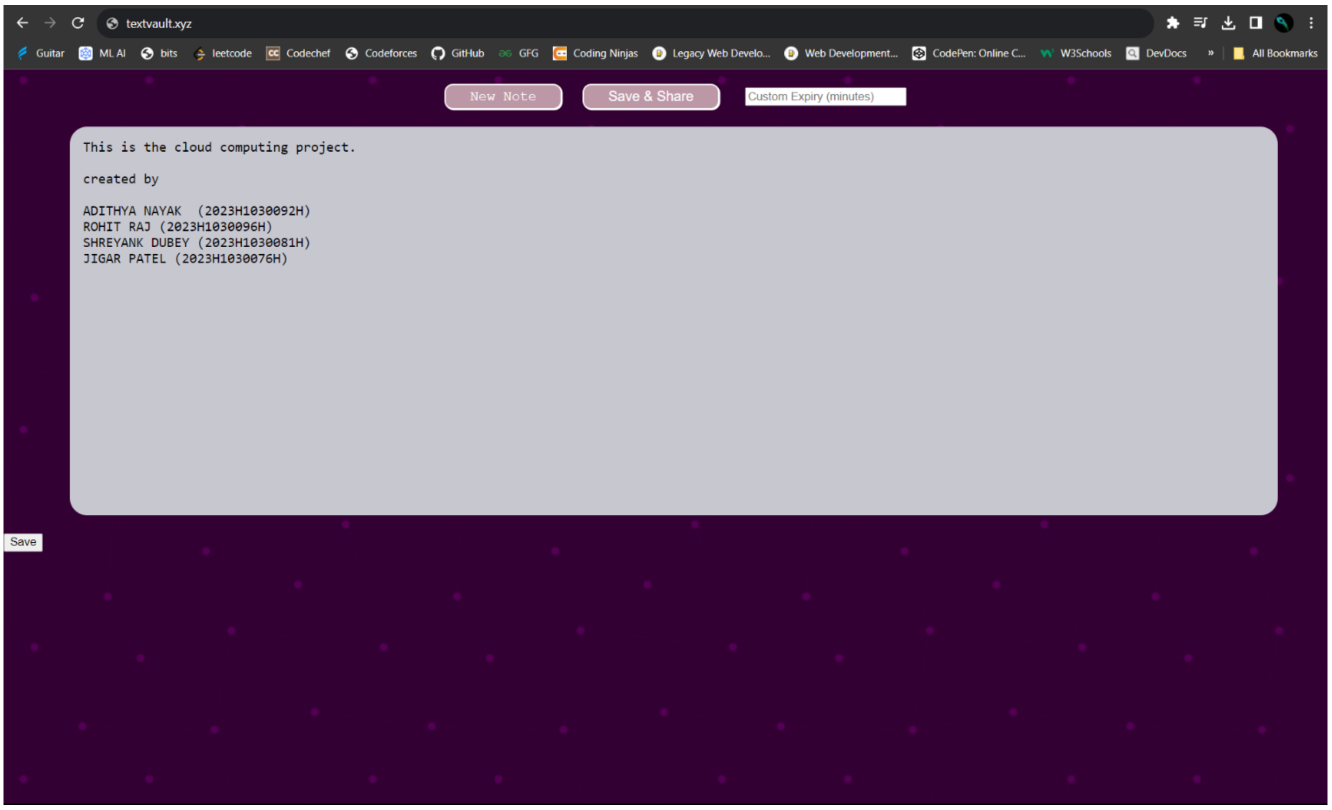
URLs
ID : [Unique] [type - String] [required - true]
longurl : [type - String]
shorturl : [type - String]
shortID : [type - String]

Logic Behind URL Shortner


textvault.xyz/
Custom Alias(abc)
Store the Short URL in Database and Redirect the link to Original Long URL

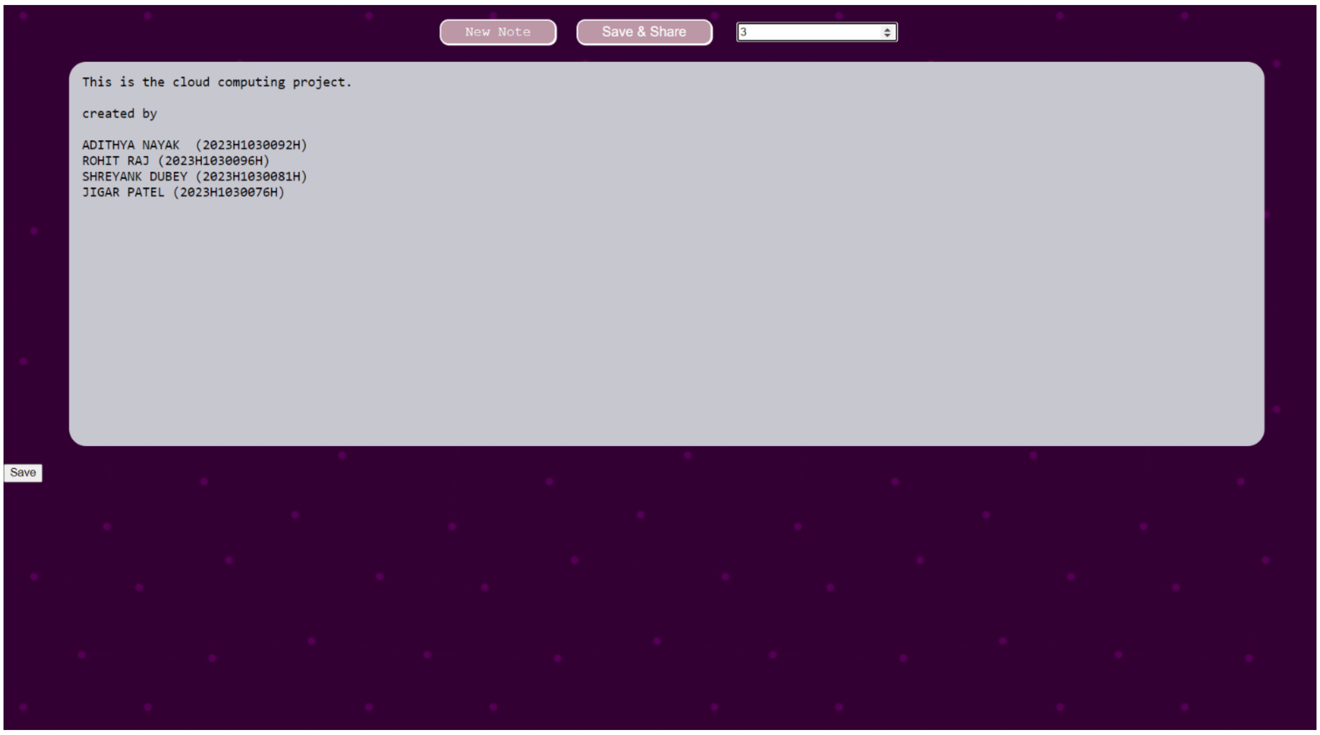
Features Of Our Project

1) Users Can **Create a note and then Share that note via unique link.**



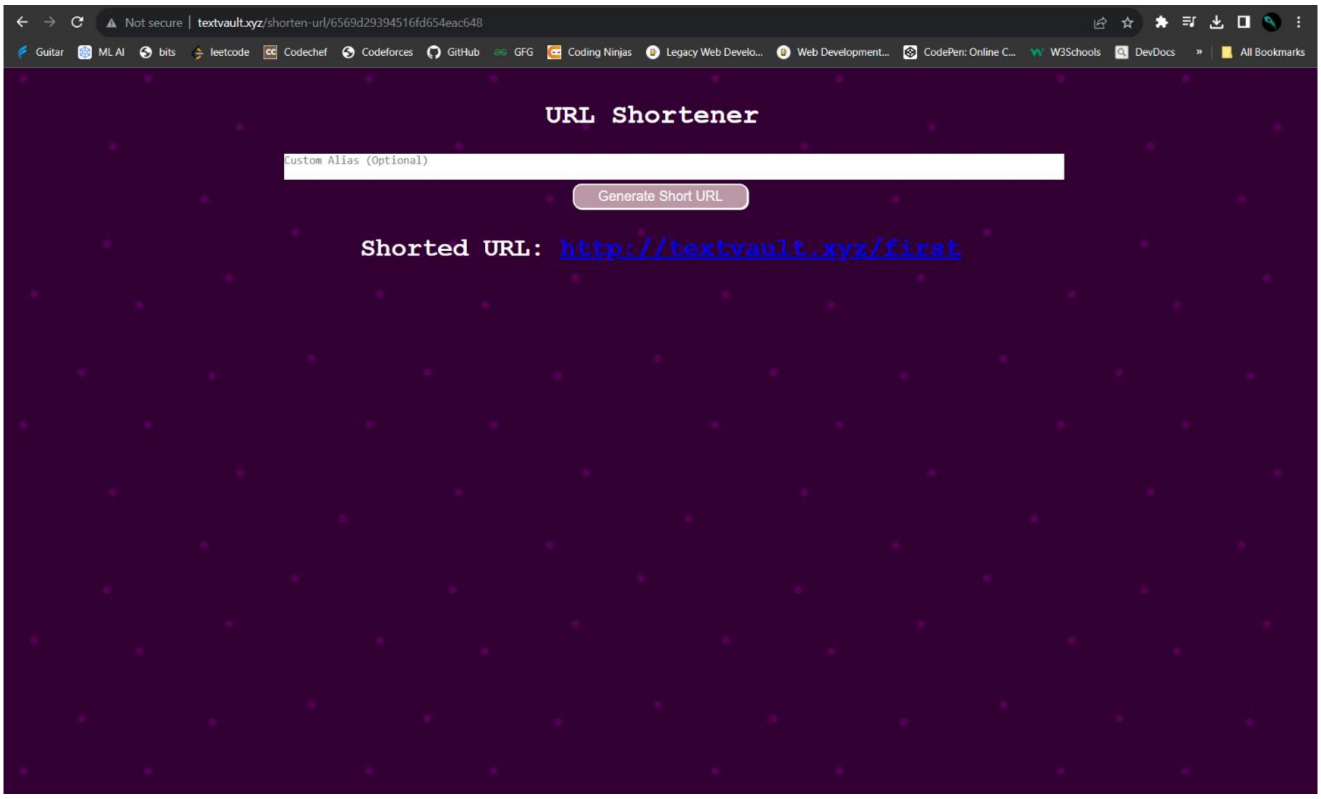
```
This is the cloud computing project.

created by

ADITHYA NAYAK  (2023H1030092H)
ROHIT RAJ (2023H1030096H)
SHREYANK DUBEY (2023H1030081H)
JIGAR PATEL (2023H1030076H)
```

2) Users can set the **custom time limit for the generated link**. After that time link will get expired and note will be not there to access.



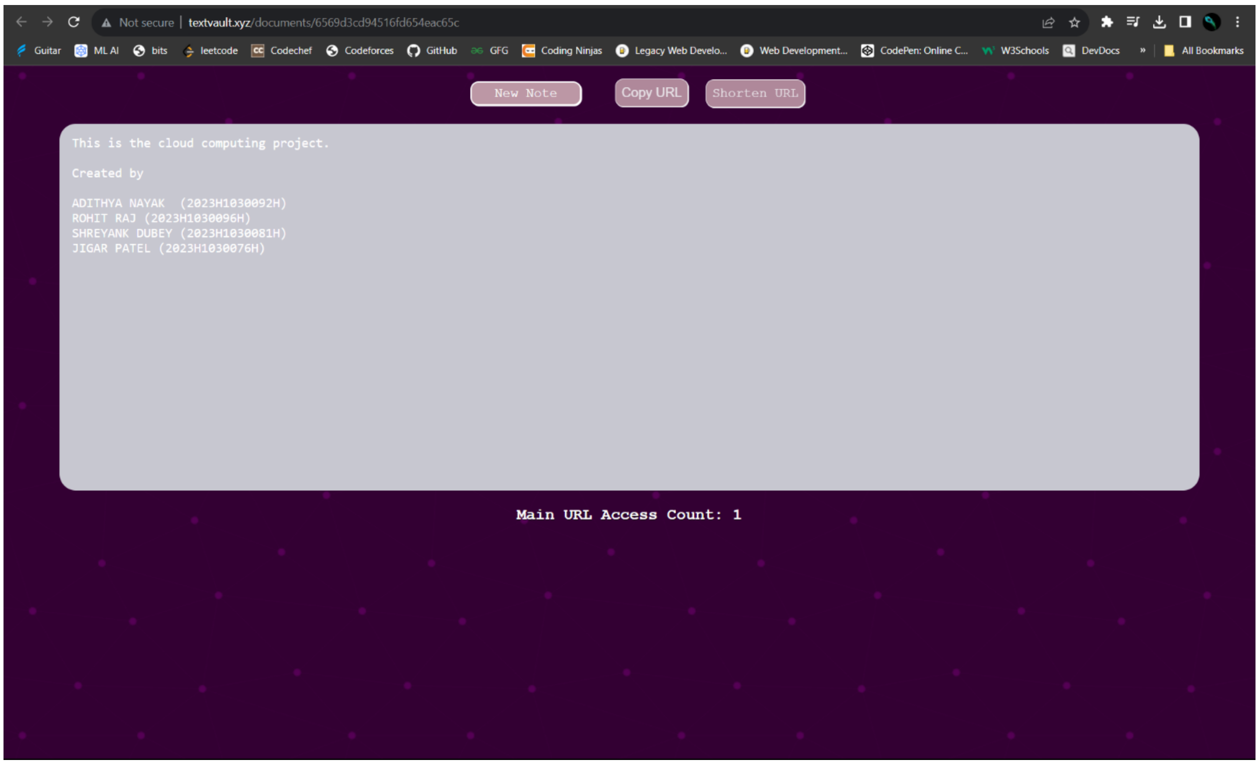
```
This is the cloud computing project.

created by

ADITHYA NAYAK  (2023H1030092H)
ROHIT RAJ (2023H1030096H)
SHREYANK DUBEY (2023H1030081H)
JIGAR PATEL (2023H1030076H)
```

3) User can generate **short url with custom alias** for their paste.


Not secure | textvault.xyz/shorten-url/6569d29394516fd654eac648
URL Shortener
Custom Alias (Optional)
Generate Short URL

Shorted URL: http://textvault.xyz/first

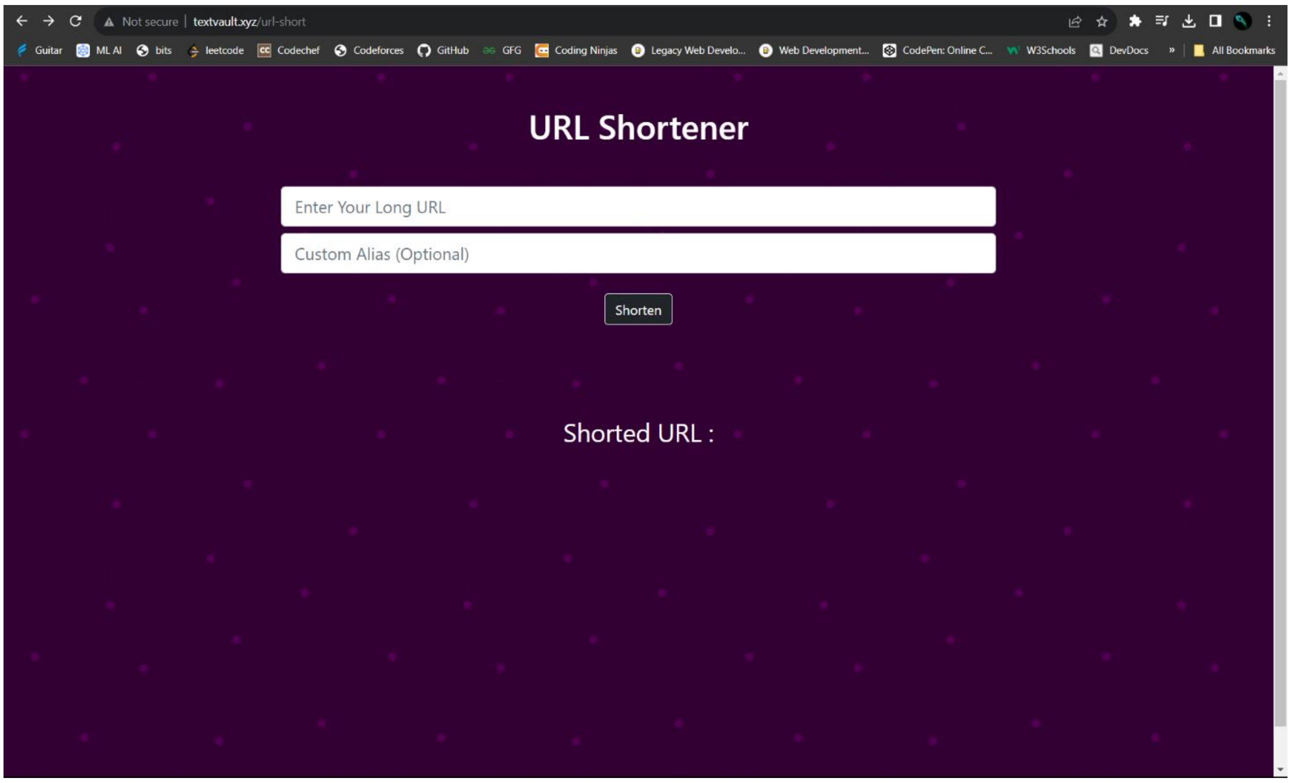
4) User Can see **how many time their Paste** is accessed.


Not secure | textvault.xyz/documents/6569d3cd94516fd654eac65c
New Note
Copy URL
Shorten URL

```
This is the cloud computing project.

Created by

ADITHYA NAYAK  (2023H1030092H)
ROHIT RAJ (2023H1030096H)
SHREYANK DUBEY (2023H1030081H)
JIGAR PATEL (2023H1030076H)
```

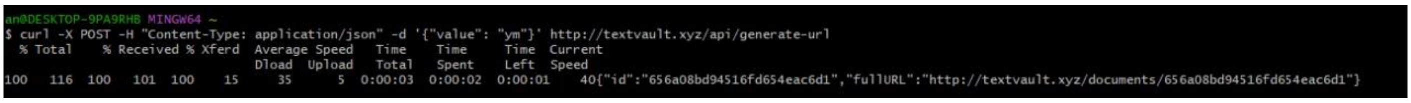
Main URL Access Count: 1

5) User Can directly use **URL Shortner** Service.



6) We **created a API of our project** so that other developers can use our project’s functionality through our API.

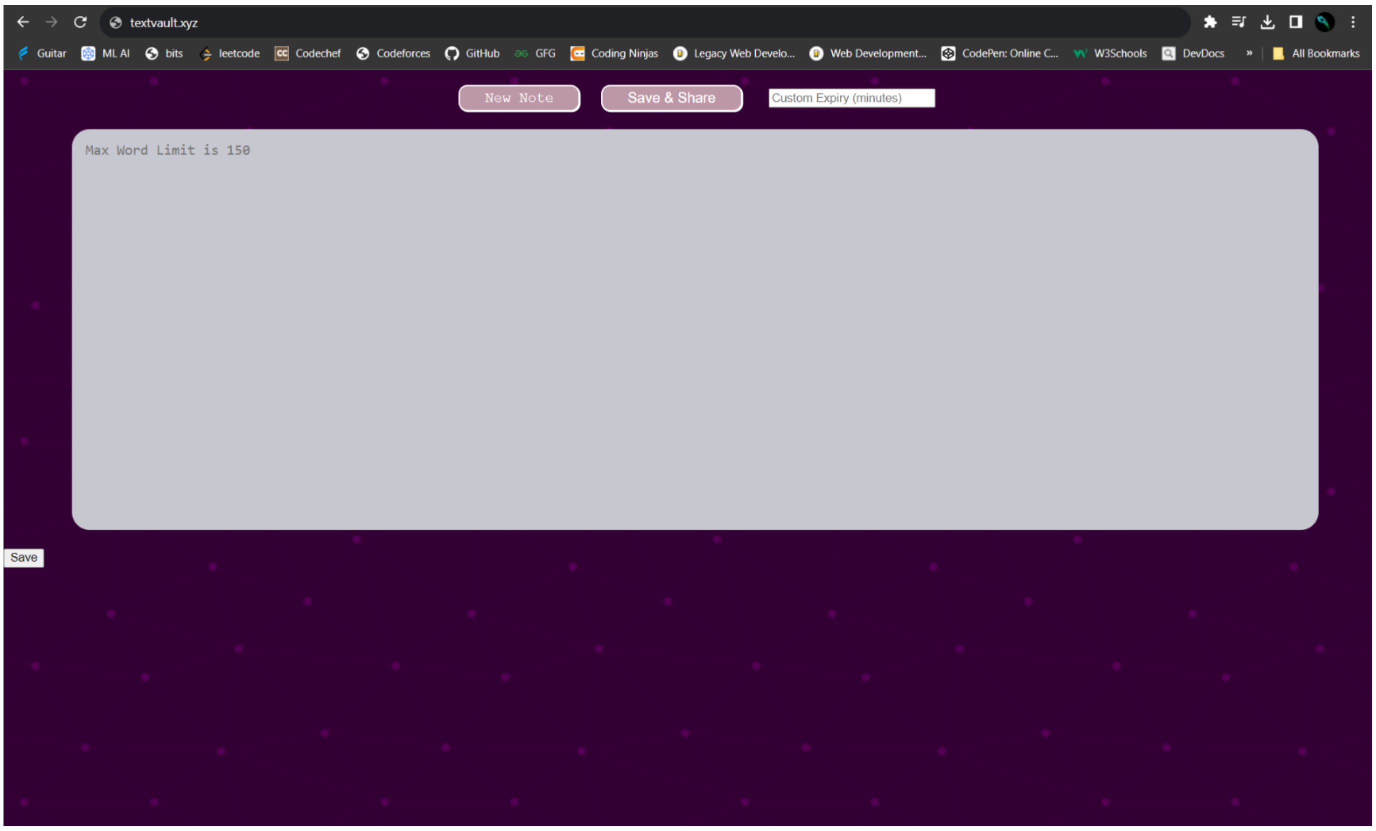
```
 $ curl -X POST -H "Content-Type: application/json" -d '{"value": "ym"}'
http://textvault.xyz/api/generate-url
```



```
an@DESKTOP-9PA9RHB MINGW64 ~
$ curl -X POST -H "Content-Type: application/json" -d '{"value": "ym"}' http://textvault.xyz/api/generate-url
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   116  100   101  100    15     35      5  0:00:03  0:00:02  0:00:01    40{"id":"656a08bd94516fd654eac6d1","fullURL":"http://textvault.xyz/documents/656a08bd94516fd654eac6d1"}
```

Design Consideration

1) **Text limitation** : We implemented a text limit of 150 words in our project. We can increase this limit in our future upgradation.



Max Word Limit is 150

2) **URL length limit** : We can set the length of url by using URL Shortner.

Perfomance Of Our Project

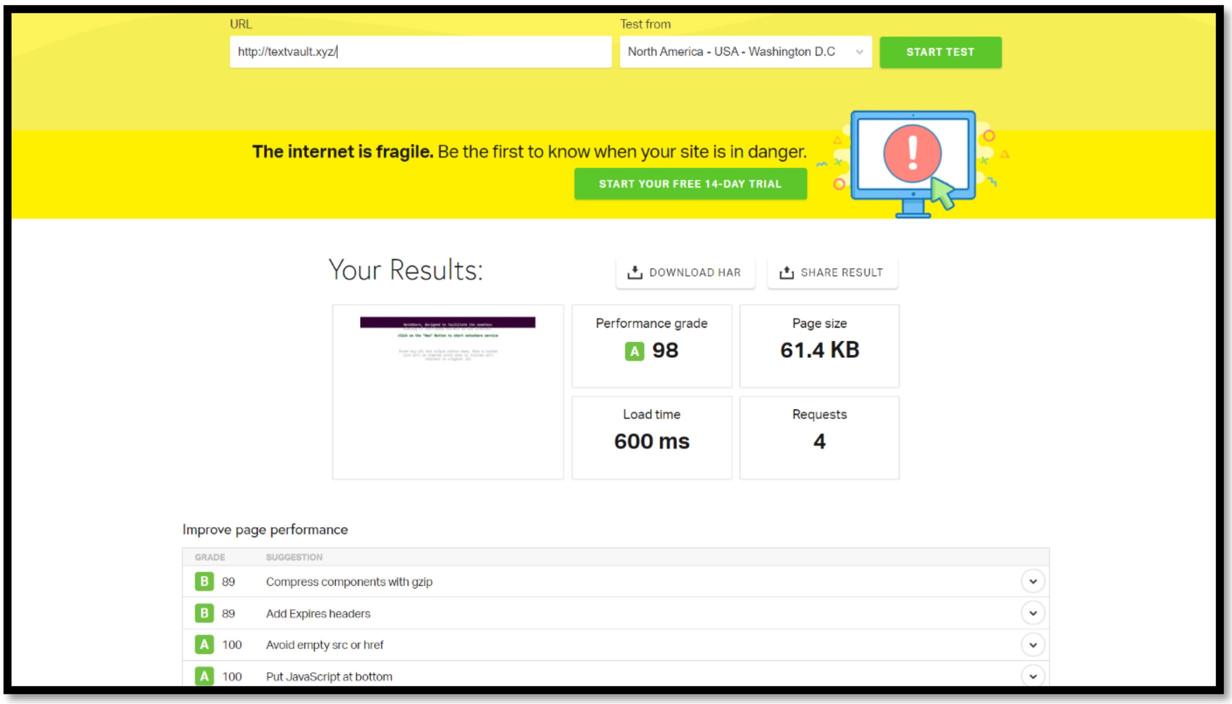
We Checked Our Site Perfomance via [tools.pingdom.com]

Perfomance Grade : A (98/100)

Load Time : 600 ms

Link of detailed report : [Report]

Page Size : 61.4 KB



URL: http://textvault.xyz/

Test from: North America - USA - Washington D.C

START TEST
The internet is fragile. Be the first to know when your site is in danger.
START YOUR FREE 14-DAY TRIAL
DOWNLOAD HAR
SHARE RESULT

Your Results:

Performance grade: A 98	Page size: 61.4 KB
Load time: 600 ms	Requests: 4

Improve page performance

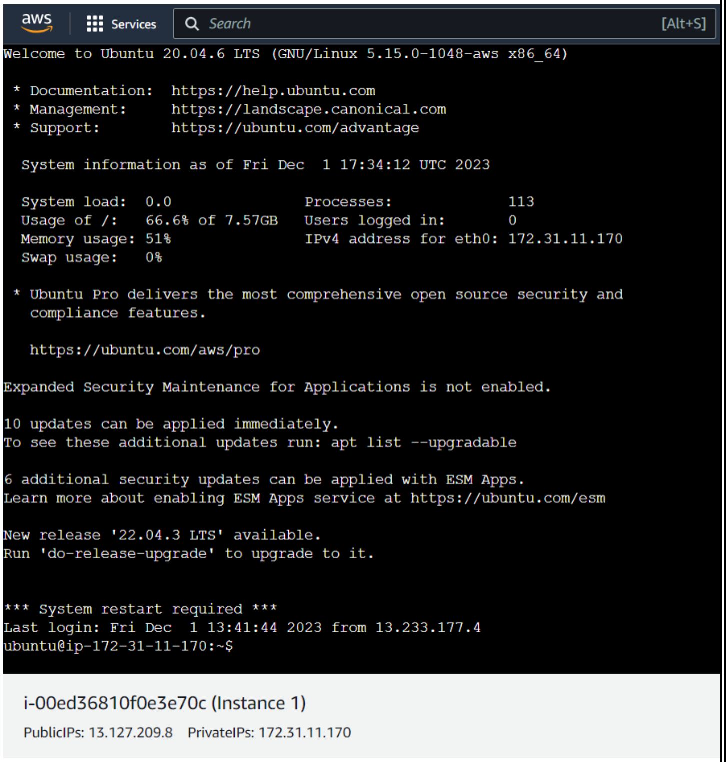
GRADE		SUGGESTION
B	89	Compress components with gzip
B	89	Add Expires headers
A	100	Avoid empty src or href
A	100	Put JavaScript at bottom

Memory And Storage Usage

Instance 1 :

Memory Use : 66.6% of 7.57 GB

Storage Use : 51%



```
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1048-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Dec  1 17:34:12 UTC 2023

  System load:  0.0                Processes:             113
  Usage of /:   66.6% of 7.57GB    Users logged in:       0
  Memory usage: 51%                IPv4 address for eth0: 172.31.11.170
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

10 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
Last login: Fri Dec  1 13:41:44 2023 from 13.233.177.4
ubuntu@ip-172-31-11-170:~$
```

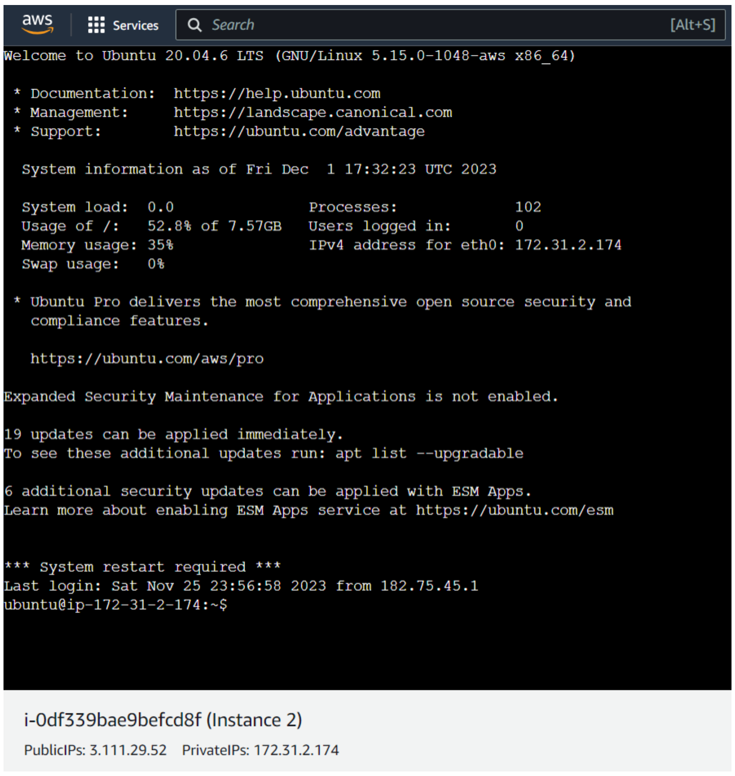
i-00ed36810f0e3e70c (Instance 1)

PublicIPs: 13.127.209.8   PrivateIPs: 172.31.11.170

Instance 2 :

Memory Use : 52.8% of 7.57 GB

Storage Use : 35%



```
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1048-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Dec  1 17:32:23 UTC 2023

  System load:  0.0                Processes:             102
  Usage of /:   52.8% of 7.57GB    Users logged in:       0
  Memory usage: 35%                IPv4 address for eth0: 172.31.2.174
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

19 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


*** System restart required ***
Last login: Sat Nov 25 23:56:58 2023 from 182.75.45.1
ubuntu@ip-172-31-2-174:~$
```

i-0df339bae9befcd8f (Instance 2)

PublicIPs: 3.111.29.52   PrivateIPs: 172.31.2.174

Future Work

Version 1.0 : In first version of our project we used MongoDB Atlas as our database container.

Version 2.0 : In next version we are thinking to make our project **Cost efficient**.

We will use DynamoDB and S3 to store our data and then use AWS Glacier storage class for cost reduction.

(If expiry time of notes is applied by user then don’t move data from S3 , but if note is permanently stored in S3 then move the data to **Glacier** after some fixed amount of time. Because if data access is not frequent then Glacier is the best option to choose.)

Right now we don’t have high traffic on our site. But we can use the **Auto Scaling** method to handle the future load.

References

1) AWS Route 53 Domain Name
2) Create AWS Load Balancer - Detailed video using the New AWS Console
3) Learn how to install AWS CloudWatch Agent on an EC2 instance
4) AWS Tutorial - Amazon CloudWatch Email Notification Alarm [Hands on Lab]
5) AWS Load Balancer HTTPS Setup with Route 53 and Certificate Manager & HTTP Redirect to HTTPS
6) Elastic Load Balancer Setup on Route 53 | Setup ELB on Route 53 | ELB-Alias Record
7) Top 50+ AWS Services Explained in 10 Minutes
8) Deploy NodeJS APP on AWS EC2 Instance | NodeJS on EC2 | Running NodeJS APP on AWS EC2