AWS Cloud Assignment

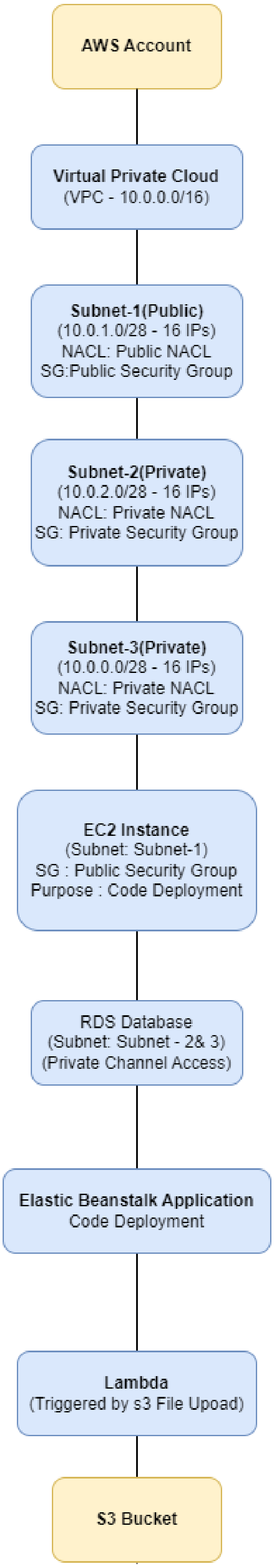
- Architecture Diagram

An architecture diagram in AWS illustrates the high-level design and components of a system deployed on the Amazon Web Services (AWS) cloud platform.

It typically includes key AWS services, such as compute instances, storage, databases, networking components, and other ancillary services.

The diagram visually represents how these components interact and communicate with each other to support the overall system functionality.

It may also depict security measures, load balancing, fault tolerance, and scalability aspects. The architecture diagram serves as a blueprint for system administrators, developers, and stakeholders to understand the system's structure, dependencies, and the flow of data or requests within the AWS environment.


AWS Account
Virtual Private Cloud
(VPC - 10.0.0.0/16)
Subnet-1(Public)
(10.0.1.0/28 - 16 IPs)
NACL: Public NACL
SG:Public Security Group
Subnet-2(Private)
(10.0.2.0/28 - 16 IPs)
NACL: Private NACL
SG: Private Security Group
Subnet-3(Private)
(10.0.0.0/28 - 16 IPs)
NACL: Private NACL
SG: Private Security Group
EC2 Instance
(Subnet: Subnet-1)
SG : Public Security Group
Purpose : Code Deployment
RDS Database
(Subnet: Subnet - 2& 3)
(Private Channel Access)
Elastic Beanstalk Application
Code Deployment
Lambda
(Triggered by s3 File Upoad)
S3 Bucket

- Component Description

This architecture diagram illustrates the following components and their relationships: -

1. **AWS Account**: The AWS account utilized for this assignment serves as the foundation for managing and accessing AWS services and resources.
2. **Virtual Private Cloud (VPC)**: In the assignment, a VPC named "my-vpc-1" was employed. It comprises three subnets: one public and two private. An internet gateway named "myInternetGateway" facilitates connectivity to the internet. The VPC has a CIDR of 10.0.0.0/16.
3. **Subnets**: Subnets within the VPC segregate portions of the defined CIDR block, enabling differentiated access rules and resource placement. This assignment employs the following subnets:-

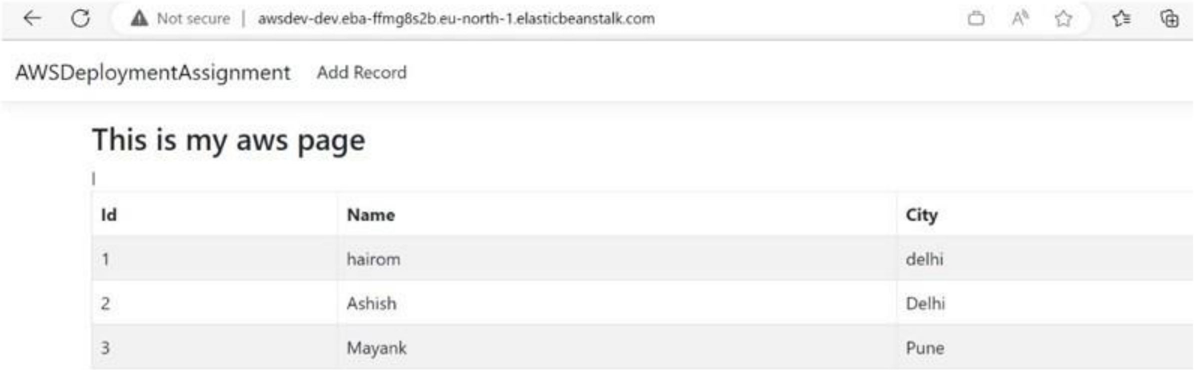
   Public Subnet: This subnet, named "publicSubnet," has a range of 16 IPs (10.0.1.0/28) and allows inbound and outbound internet traffic.

   Private Subnet 1: The "privateSubnet" restricts inbound internet traffic and is designated for storing the RDS instance. It has a range of 16 IPs (10.0.2.0/28).

   Private Subnet 2: "privateSubnet2" is created specifically for the RDS instance and covers at least 2 availability zones. It has a range of 16 IPs (10.0.0.0/28).
4. **EC2 Instance**: An EC2 virtual machine is deployed in the Public Subnet (Subnet 1). It runs the application code and communicates with the RDS database through a private channel. The associated security group is called "launch-wizard1."
5. **RDS Database**: The assignment utilizes an RDS database instance for the application code to communicate with. This managed MySQL database is accessible only through the private channel.
6. **Elastic Beanstalk**: AWS Elastic Beanstalk is employed to deploy the application code in a managed environment with auto-scaling capabilities.
7. **Lambda**: The AWS Lambda is a serverless compute service, to trigger a function whenever a file is uploaded to the S3 bucket. The Lambda function simply prints the name of the uploaded file.
8. **S3 Bucket**: An Amazon S3 bucket is used to store files and trigger the Lambda function whenever a file is uploaded to the bucket.

- Screenshots

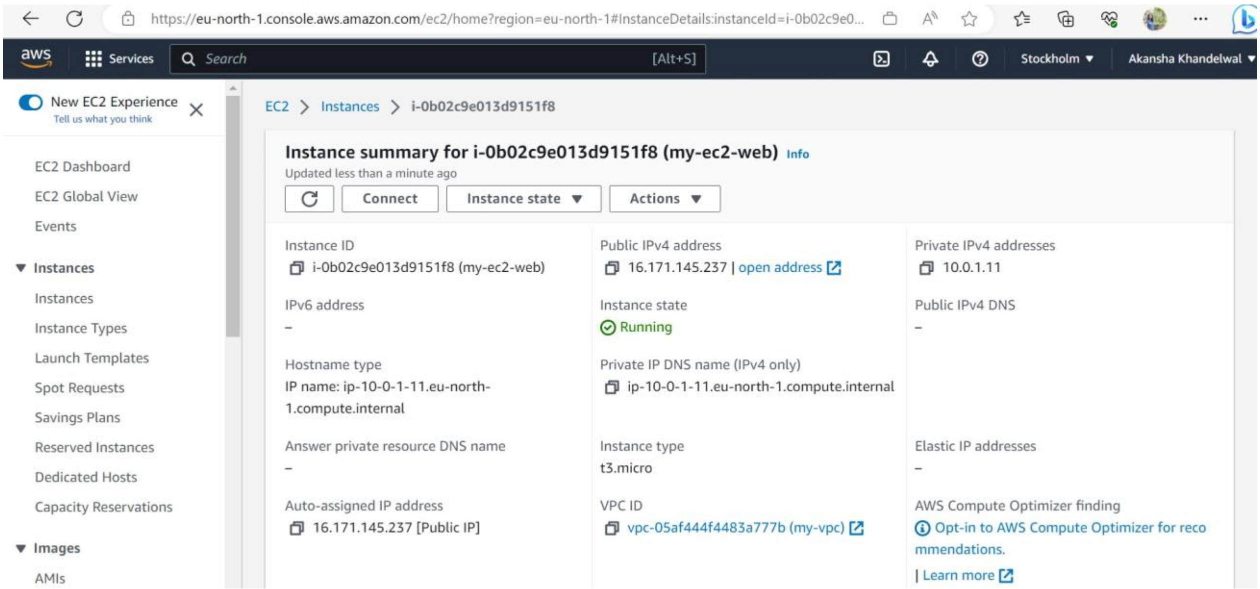
Running Demo of code deployed on VM:


Not secure | awsdev-dev.eba-ffmg8s2b.eu-north-1.elasticbeanstalk.com
AWSDeploymentAssignment
Add Record

This is my aws page

Id	Name	City
1	hairom	delhi
2	Ashish	Delhi
3	Mayank	Pune

EC2 instance created under my VPC:



Instance summary for i-0b02c9e013d9151f8 (my-ec2-web) Info

Updated less than a minute ago

Instance ID: i-0b02c9e013d9151f8 (my-ec2-web)

Public IPv4 address: 16.171.145.237 | open address

Private IPv4 addresses: 10.0.1.11

IPv6 address: –

Instance state: Running

Public IPv4 DNS: –

Hostname type: IP name: ip-10-0-1-11.eu-north-1.compute.internal

Private IP DNS name (IPv4 only): ip-10-0-1-11.eu-north-1.compute.internal

Answer private resource DNS name: –

Instance type: t3.micro

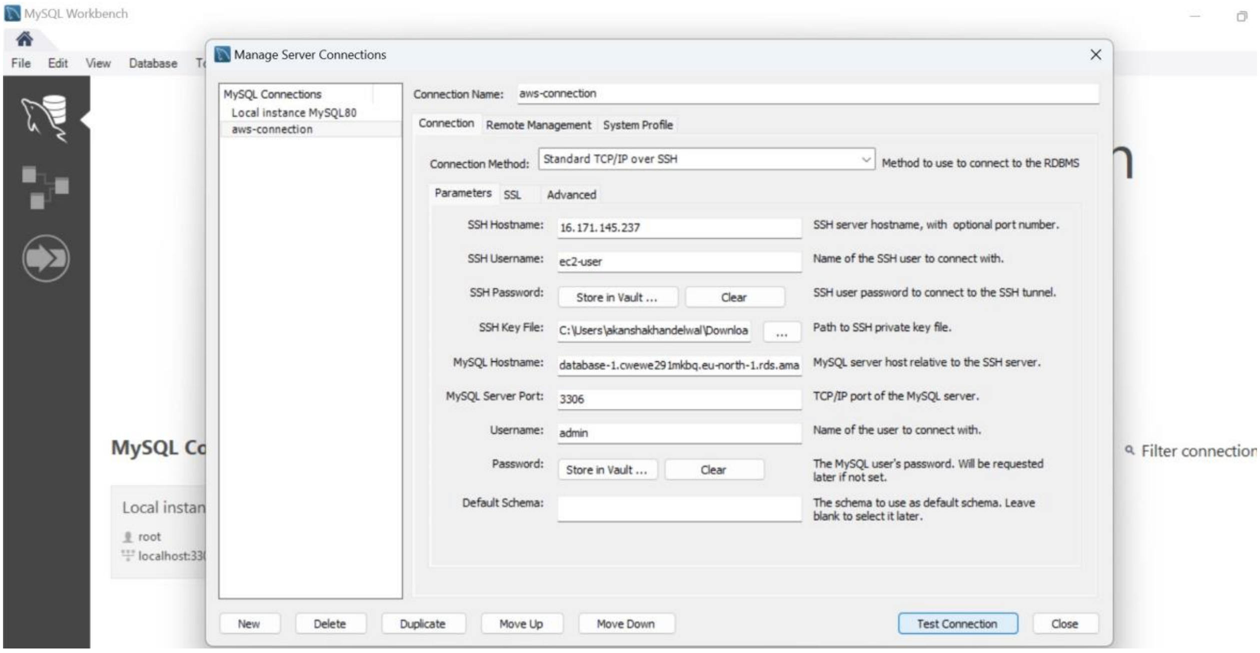
Elastic IP addresses: –

Auto-assigned IP address: 16.171.145.237 [Public IP]

VPC ID: vpc-05af444f4483a777b (my-vpc)

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Connection between RDS instance and EC2 instance(through private channel):



Connection Name: aws-connection

Connection Method: Standard TCP/IP over SSH — Method to use to connect to the RDBMS

SSH Hostname: 16.171.145.237 — SSH server hostname, with optional port number.

SSH Username: ec2-user — Name of the SSH user to connect with.

SSH Password: Store in Vault ... | Clear — SSH user password to connect to the SSH tunnel.

SSH Key File: C:\Users\akanshakhandelwal\Downloa — Path to SSH private key file.

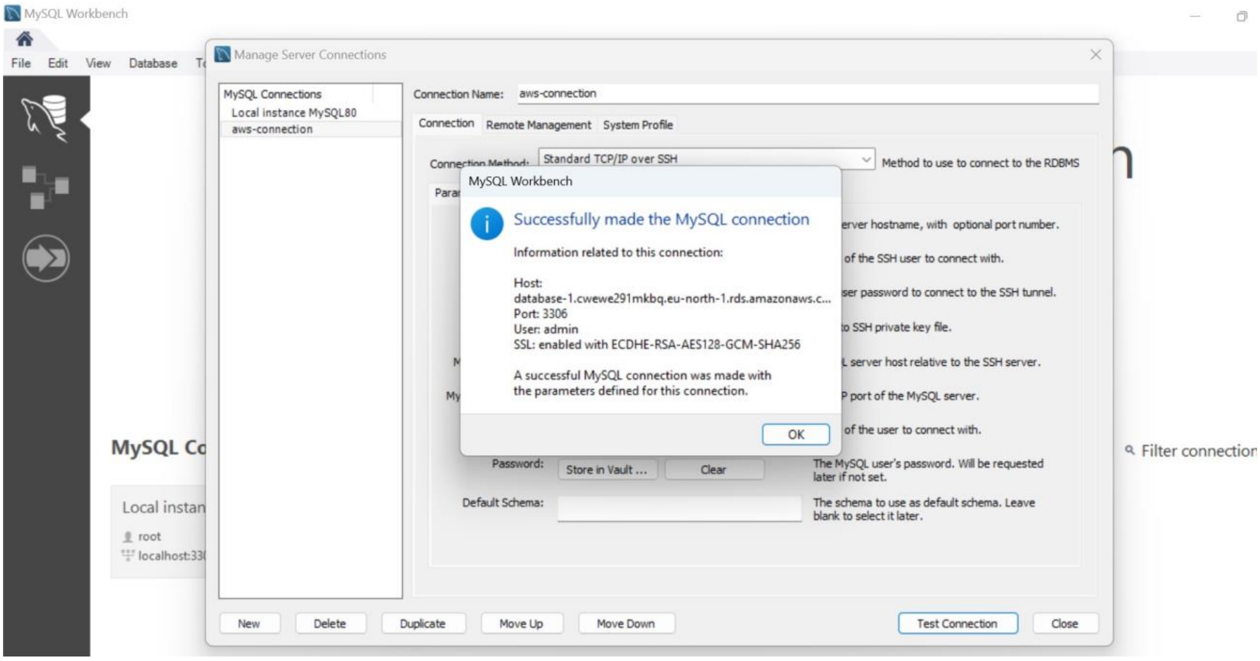
MySQL Hostname: database-1.cwewe291mkbq.eu-north-1.rds.ama — MySQL server host relative to the SSH server.

MySQL Server Port: 3306 — TCP/IP port of the MySQL server.

Username: admin — Name of the user to connect with.

Password: Store in Vault ... | Clear — The MySQL user's password. Will be requested later if not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.



Connection Name: aws-connection

Connection Method: Standard TCP/IP over SSH

MySQL Workbench

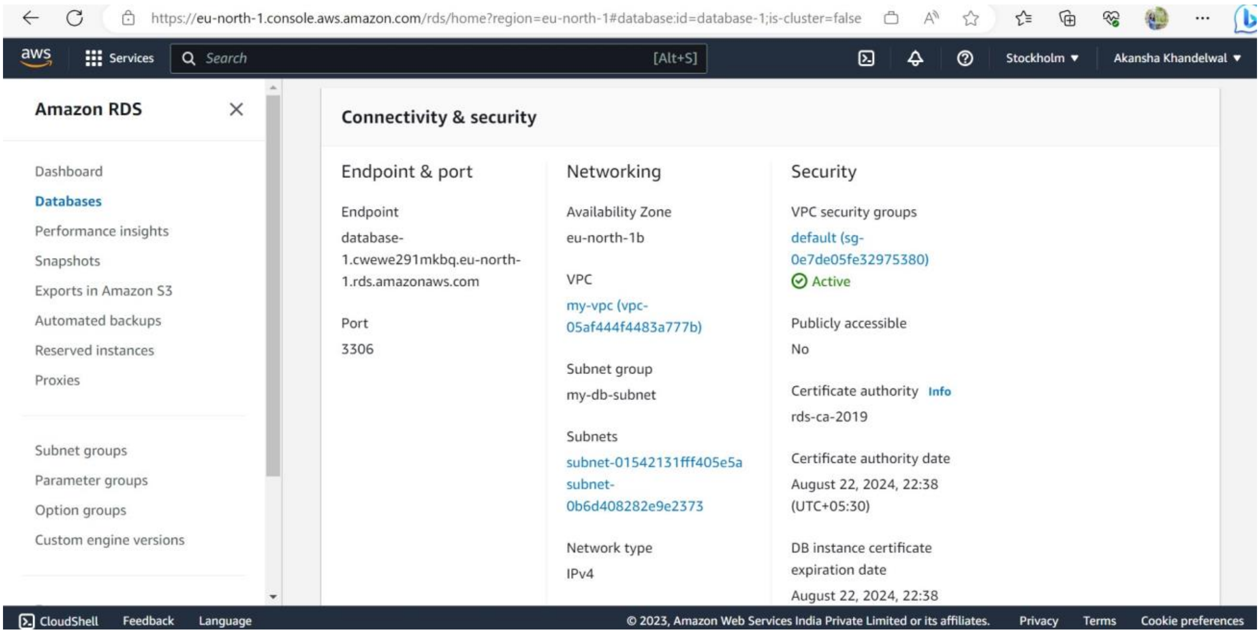
Successfully made the MySQL connection

Information related to this connection:

Host:
database-1.cwewe291mkbq.eu-north-1.rds.amazonaws.c...
Port: 3306
User: admin
SSL: enabled with ECDHE-RSA-AES128-GCM-SHA256

A successful MySQL connection was made with
the parameters defined for this connection.

RDS Instance:



Connectivity & security

Endpoint & port

Endpoint
database-1.cwewe291mkbq.eu-north-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
eu-north-1b

VPC
my-vpc (vpc-05af444f4483a777b)

Subnet group
my-db-subnet

Subnets
subnet-01542131fff405e5a
subnet-0b6d408282e9e2373

Network type
IPv4

Security

VPC security groups
default (sg-0e7de05fe32975380)
Active

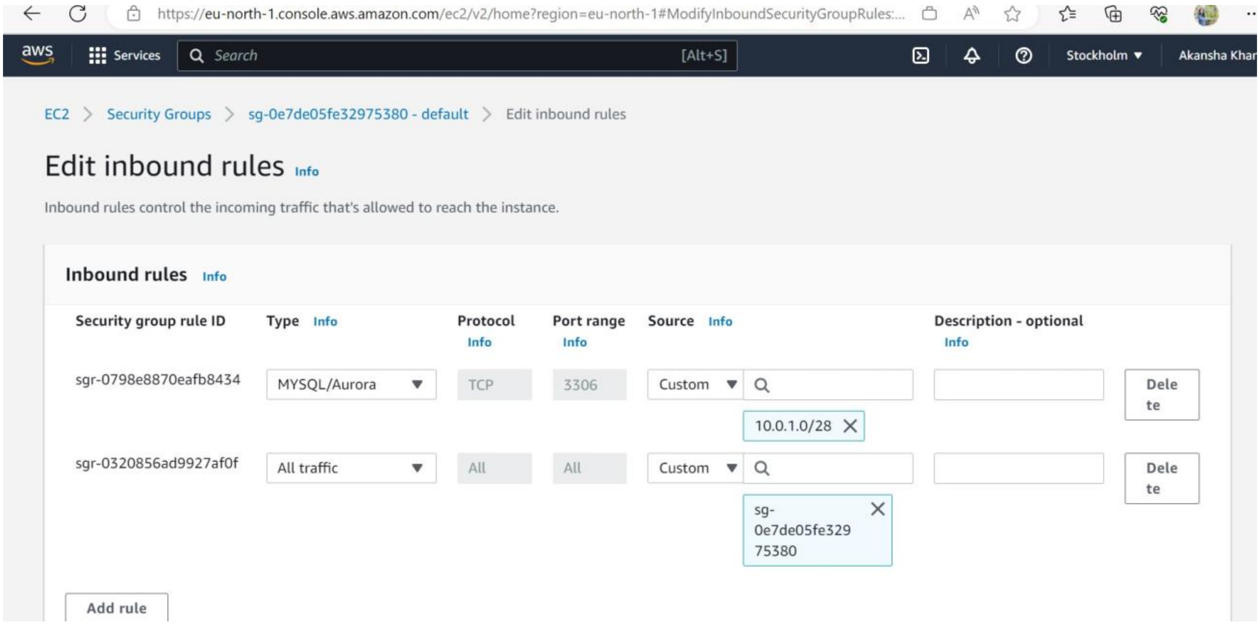
Publicly accessible
No

Certificate authority Info
rds-ca-2019

Certificate authority date
August 22, 2024, 22:38 (UTC+05:30)

DB instance certificate expiration date
August 22, 2024, 22:38

Security group:


https://eu-north-1.console.aws.amazon.com/ec2/v2/home?region=eu-north-1#ModifyInboundSecurityGroupRules:...
aws
Services
Search
[Alt+S]
Stockholm
Akansha Khar
EC2 > Security Groups > sg-0e7de05fe32975380 - default > Edit inbound rules
Delete
Delete
Add rule

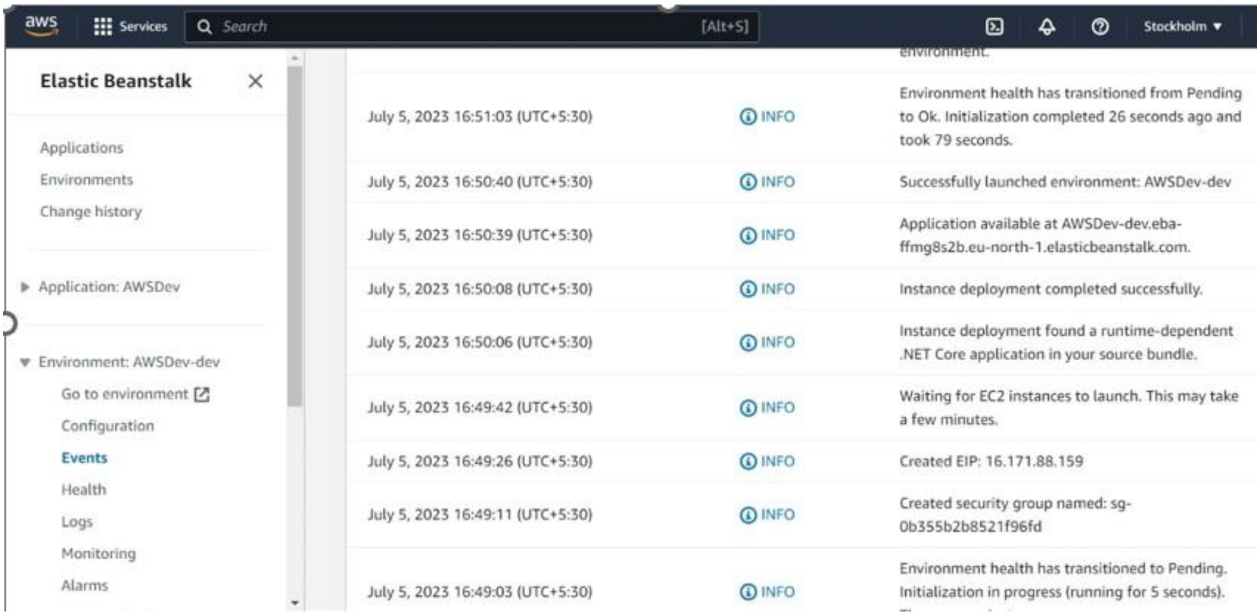
Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

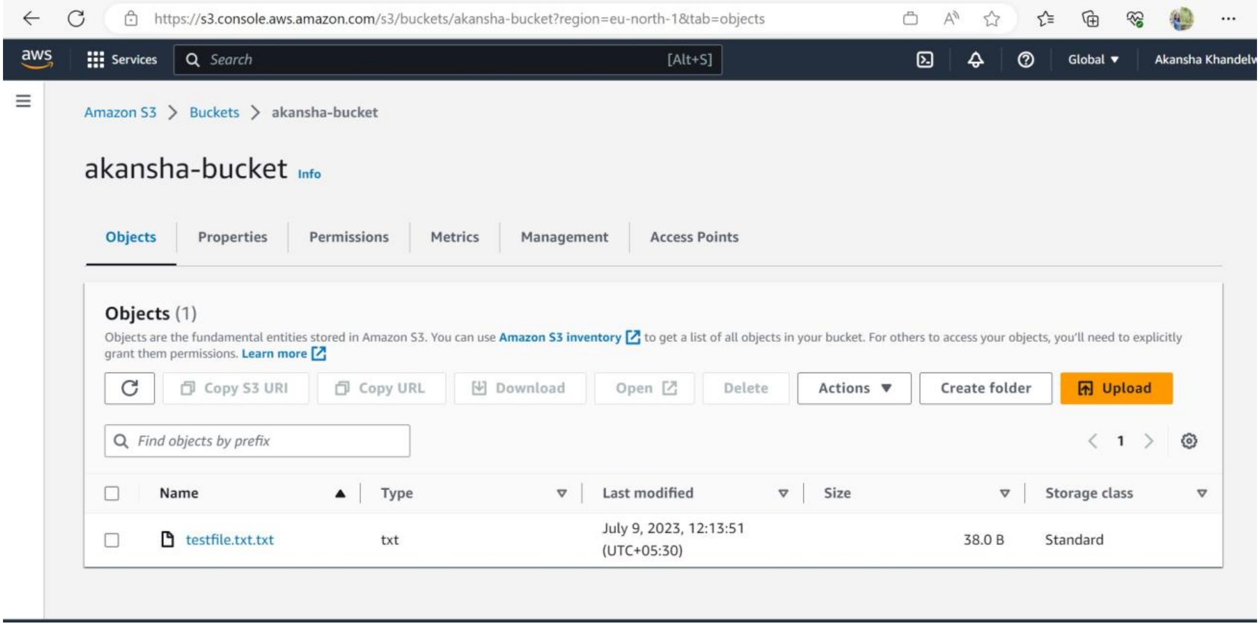
Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sgr-0798e8870eafb8434	MYSQL/Aurora	TCP	3306	Custom 10.0.1.0/28	
sgr-0320856ad9927af0f	All traffic	All	All	Custom sg-0e7de05fe32975380	

Elastic Beanstalk Environment:



		environment.
July 5, 2023 16:51:03 (UTC+5:30)	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 26 seconds ago and took 79 seconds.
July 5, 2023 16:50:40 (UTC+5:30)	INFO	Successfully launched environment: AWSDev-dev
July 5, 2023 16:50:39 (UTC+5:30)	INFO	Application available at AWSDev-dev.eba-ffmg8s2b.eu-north-1.elasticbeanstalk.com.
July 5, 2023 16:50:08 (UTC+5:30)	INFO	Instance deployment completed successfully.
July 5, 2023 16:50:06 (UTC+5:30)	INFO	Instance deployment found a runtime-dependent .NET Core application in your source bundle.
July 5, 2023 16:49:42 (UTC+5:30)	INFO	Waiting for EC2 instances to launch. This may take a few minutes.
July 5, 2023 16:49:26 (UTC+5:30)	INFO	Created EIP: 16.171.88.159
July 5, 2023 16:49:11 (UTC+5:30)	INFO	Created security group named: sg-0b355b2b8521f96fd
July 5, 2023 16:49:03 (UTC+5:30)	INFO	Environment health has transitioned to Pending. Initialization in progress (running for 5 seconds).

Bucket generated by Elastic Beanstalk:



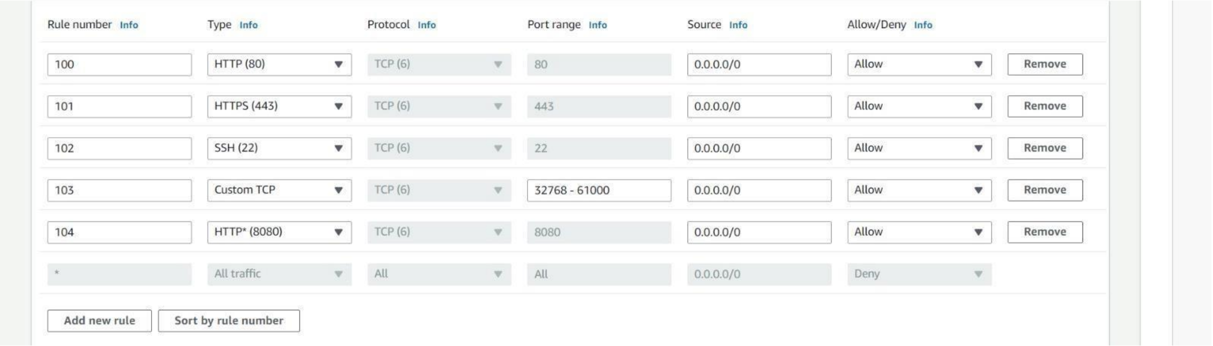
akansha-bucket Info

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

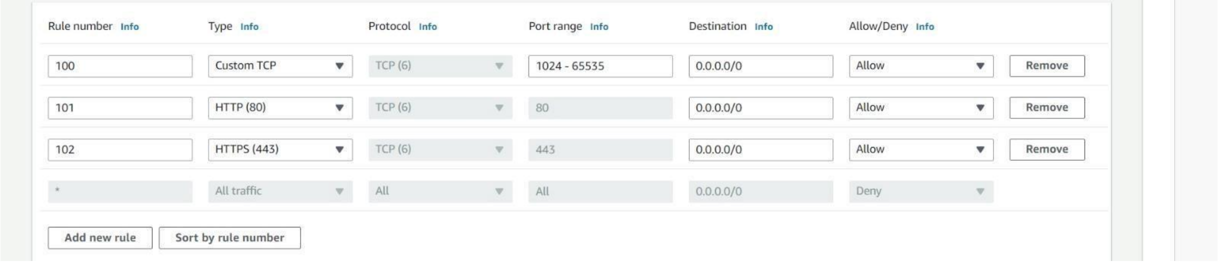
	Name	Type	Last modified	Size	Storage class
□	testfile.txt.txt	txt	July 9, 2023, 12:13:51 (UTC+05:30)	38.0 B	Standard

NACL:



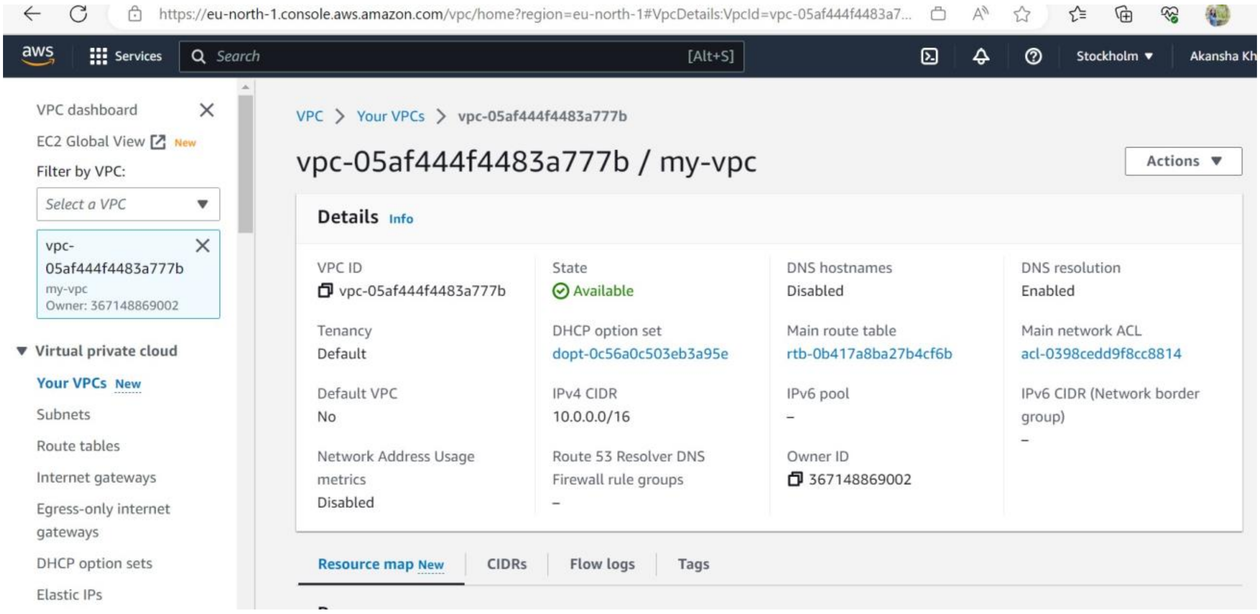
Rule number Info	Type Info	Protocol Info	Port range Info	Source Info	Allow/Deny Info	
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow	Remove
101	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow	Remove
102	SSH (22)	TCP (6)	22	0.0.0.0/0	Allow	Remove
103	Custom TCP	TCP (6)	32768 - 61000	0.0.0.0/0	Allow	Remove
104	HTTP* (8080)	TCP (6)	8080	0.0.0.0/0	Allow	Remove
*	All traffic	All	All	0.0.0.0/0	Deny	

SG:



Rule number Info	Type Info	Protocol Info	Port range Info	Destination Info	Allow/Deny Info	
100	Custom TCP	TCP (6)	1024 - 65535	0.0.0.0/0	Allow	Remove
101	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow	Remove
102	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow	Remove
*	All traffic	All	All	0.0.0.0/0	Deny	

VPC:



vpc-05af444f4483a777b / my-vpc

Details Info

VPC ID
vpc-05af444f4483a777b

State
Available

DNS hostnames
Disabled

DNS resolution
Enabled

Tenancy
Default

DHCP option set
dopt-0c56a0c503eb3a95e

Main route table
rtb-0b417a8ba27b4cf6b

Main network ACL
acl-0398cedd9f8cc8814

Default VPC
No

IPv4 CIDR
10.0.0.0/16

IPv6 pool
–

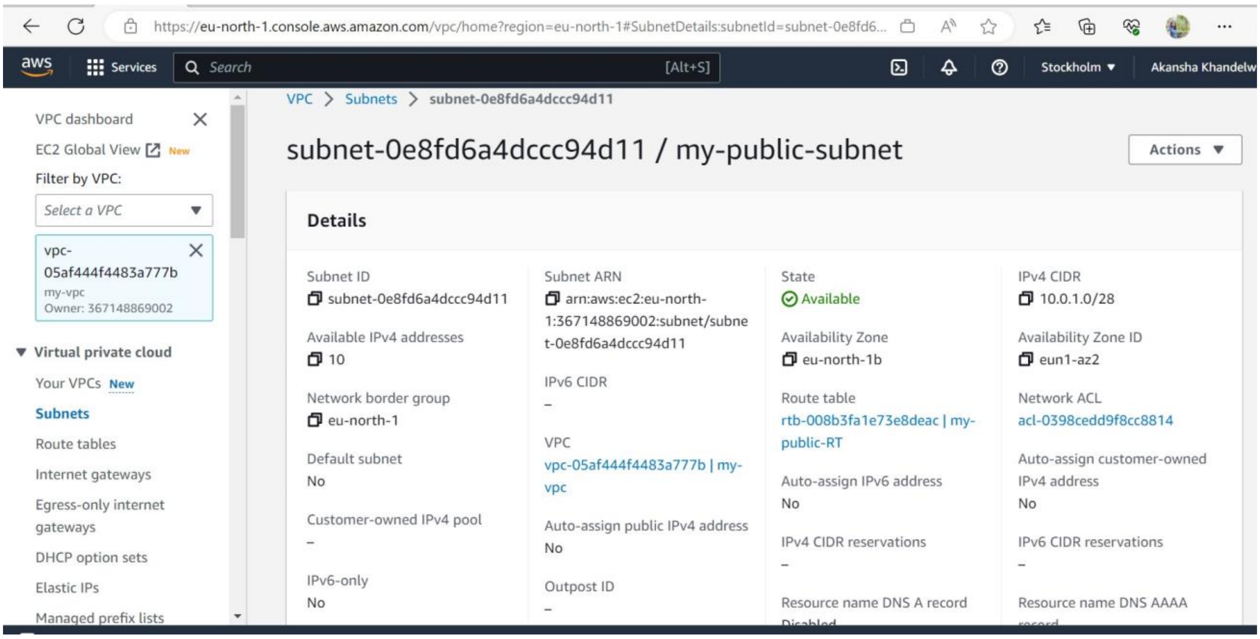
IPv6 CIDR (Network border group)
–

Network Address Usage metrics
Disabled

Route 53 Resolver DNS Firewall rule groups
–

Owner ID
367148869002

Public subnet:



subnet-0e8fd6a4dccc94d11 / my-public-subnet

Details

Subnet ID
subnet-0e8fd6a4dccc94d11

Subnet ARN
arn:aws:ec2:eu-north-1:367148869002:subnet/subnet-0e8fd6a4dccc94d11

State
Available

IPv4 CIDR
10.0.1.0/28

Available IPv4 addresses
10

Availability Zone
eu-north-1b

Availability Zone ID
eun1-az2

Network border group
eu-north-1

IPv6 CIDR
–

Route table
rtb-008b3fa1e73e8deac | my-public-RT

Network ACL
acl-0398cedd9f8cc8814

Default subnet
No

VPC
vpc-05af444f4483a777b | my-vpc

Auto-assign IPv6 address
No

Auto-assign customer-owned IPv4 address
No

Customer-owned IPv4 pool
–

Auto-assign public IPv4 address
No

IPv4 CIDR reservations
–

IPv6 CIDR reservations
–

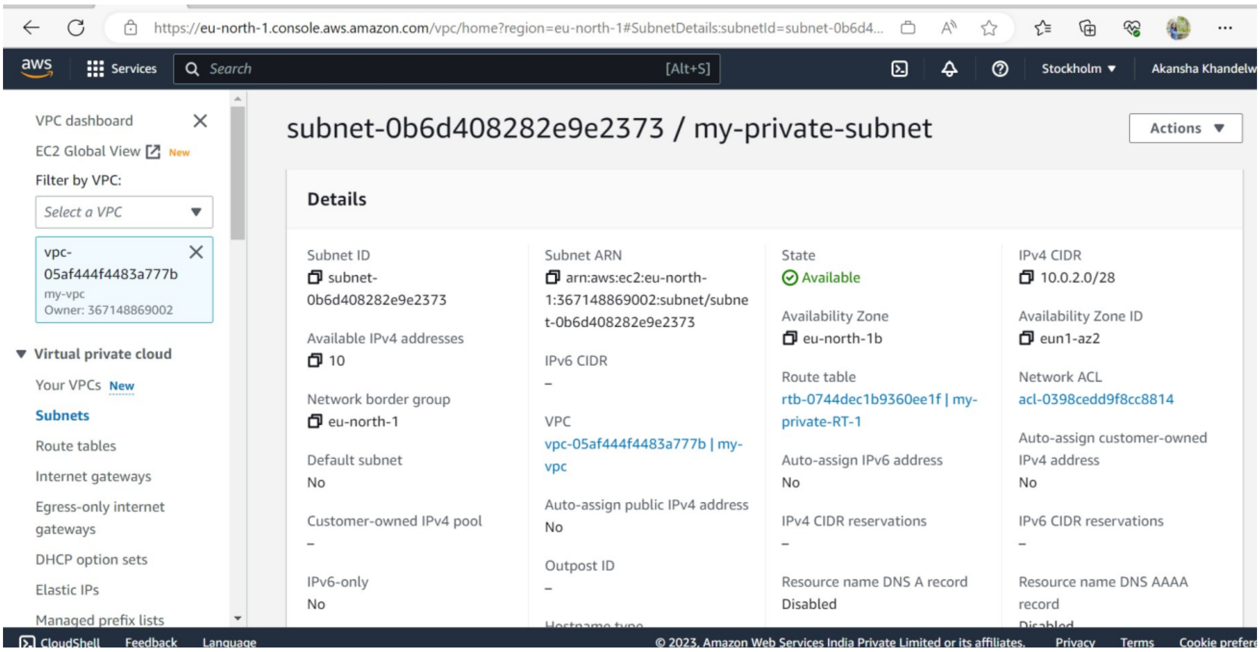
IPv6-only
No

Outpost ID
–

Resource name DNS A record
Disabled

Resource name DNS AAAA record

Private Subnet 1:



subnet-0b6d408282e9e2373 / my-private-subnet

Details

Subnet ID
subnet-0b6d408282e9e2373

Available IPv4 addresses
10

Network border group
eu-north-1

Default subnet
No

Customer-owned IPv4 pool
–

IPv6-only
No

Subnet ARN
arn:aws:ec2:eu-north-1:367148869002:subnet/subnet-0b6d408282e9e2373

IPv6 CIDR
–

VPC
vpc-05af444f4483a777b | my-vpc

Auto-assign public IPv4 address
No

Outpost ID
–

State
Available

Availability Zone
eu-north-1b

Route table
rtb-0744dec1b9360ee1f | my-private-RT-1

Auto-assign IPv6 address
No

IPv4 CIDR reservations
–

Resource name DNS A record
Disabled

IPv4 CIDR
10.0.2.0/28

Availability Zone ID
eun1-az2

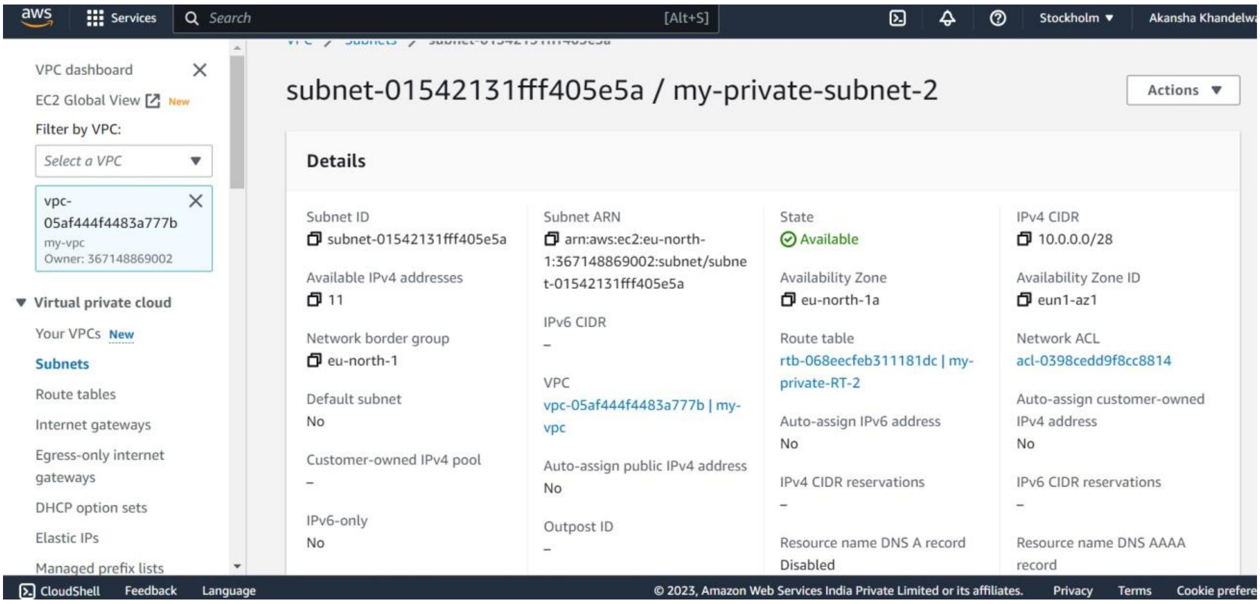
Network ACL
acl-0398cedd9f8cc8814

Auto-assign customer-owned IPv4 address
No

IPv6 CIDR reservations
–

Resource name DNS AAAA record

Private Subnet 2:



subnet-01542131fff405e5a / my-private-subnet-2

Details

Subnet ID
subnet-01542131fff405e5a

Available IPv4 addresses
11

Network border group
eu-north-1

Default subnet
No

Customer-owned IPv4 pool
–

IPv6-only
No

Subnet ARN
arn:aws:ec2:eu-north-1:367148869002:subnet/subnet-01542131fff405e5a

IPv6 CIDR
–

VPC
vpc-05af444f4483a777b | my-vpc

Auto-assign public IPv4 address
No

Outpost ID
–

State
Available

Availability Zone
eu-north-1a

Route table
rtb-068eecfeb311181dc | my-private-RT-2

Auto-assign IPv6 address
No

IPv4 CIDR reservations
–

Resource name DNS A record
Disabled

IPv4 CIDR
10.0.0.0/28

Availability Zone ID
eun1-az1

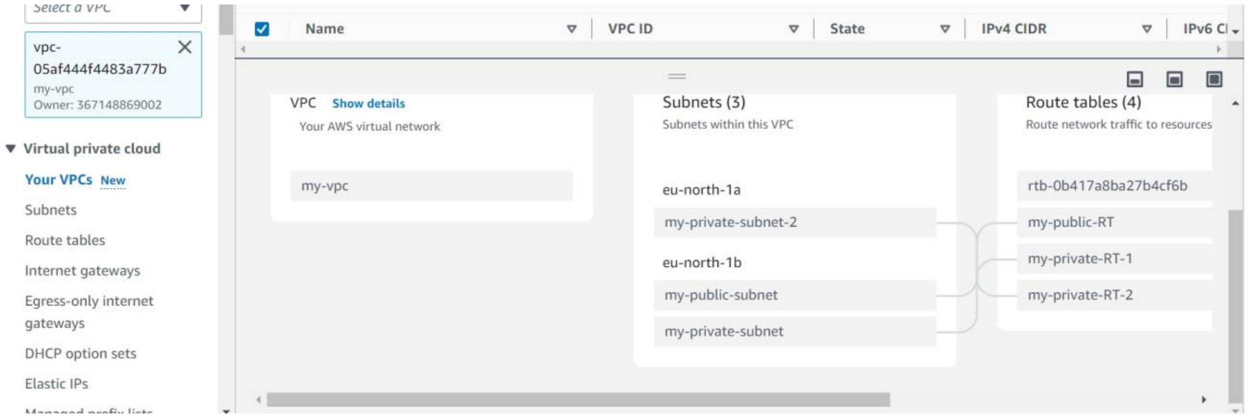
Network ACL
acl-0398cedd9f8cc8814

Auto-assign customer-owned IPv4 address
No

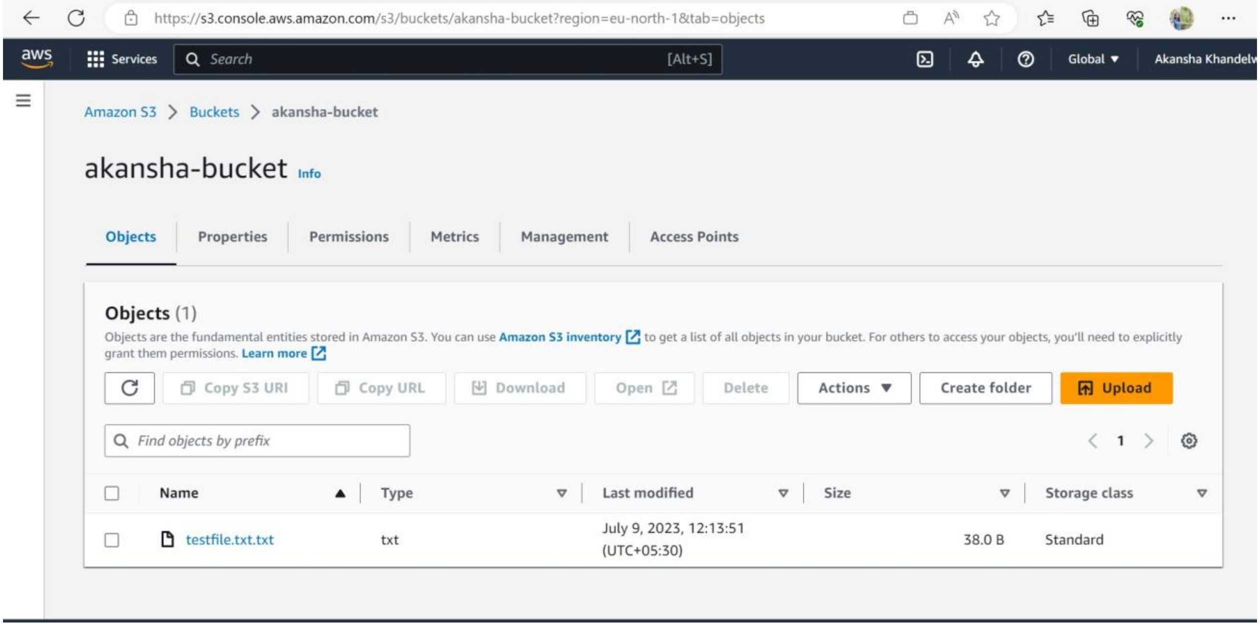
IPv6 CIDR reservations
–

Resource name DNS AAAA record

Resource map:



My bucket:

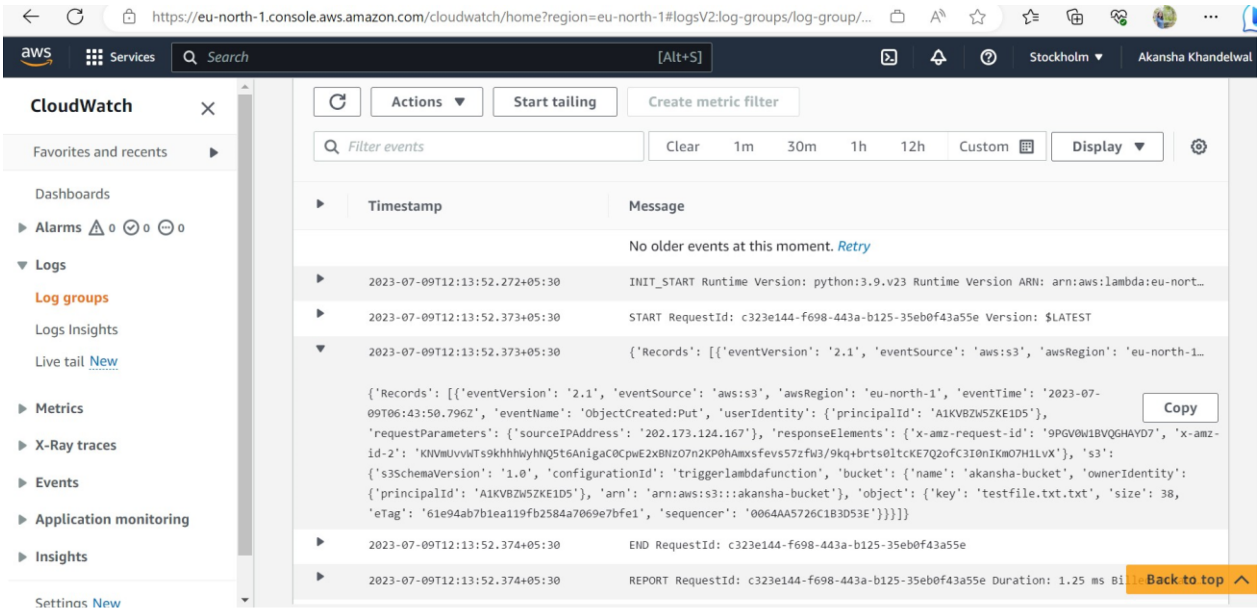


Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	testfile.txt.txt	txt	July 9, 2023, 12:13:51 (UTC+05:30)	38.0 B	Standard

Log events:



```
Timestamp                        Message
                                 No older events at this moment. Retry
2023-07-09T12:13:52.272+05:30    INIT_START Runtime Version: python:3.9.v23 Runtime Version ARN: arn:aws:lambda:eu-nort…
2023-07-09T12:13:52.373+05:30    START RequestId: c323e144-f698-443a-b125-35eb0f43a55e Version: $LATEST
2023-07-09T12:13:52.373+05:30    {'Records': [{'eventVersion': '2.1', 'eventSource': 'aws:s3', 'awsRegion': 'eu-north-1…

{'Records': [{'eventVersion': '2.1', 'eventSource': 'aws:s3', 'awsRegion': 'eu-north-1', 'eventTime': '2023-07-
09T06:43:50.796Z', 'eventName': 'ObjectCreated:Put', 'userIdentity': {'principalId': 'A1KVBZW5ZKE1D5'},
'requestParameters': {'sourceIPAddress': '202.173.124.167'}, 'responseElements': {'x-amz-request-id': '9PGV0W1BVQGHAYD7', 'x-amz-
id-2': 'KNVmUvvWTs9khhhWyhNQ5t6AnigaC0CpwE2xBNzO7n2KP0hAmxsfevs57zfW3/9kq+brts0ltcKE7Q2ofC3I0nIKmO7H1LvX'}, 's3':
{'s3SchemaVersion': '1.0', 'configurationId': 'triggerlambdafunction', 'bucket': {'name': 'akansha-bucket', 'ownerIdentity':
{'principalId': 'A1KVBZW5ZKE1D5'}, 'arn': 'arn:aws:s3:::akansha-bucket'}, 'object': {'key': 'testfile.txt.txt', 'size': 38,
'eTag': '61e94ab7b1ea119fb2584a7069e7bfe1', 'sequencer': '0064AA5726C1B3D53E'}}}]}

2023-07-09T12:13:52.374+05:30    END RequestId: c323e144-f698-443a-b125-35eb0f43a55e
2023-07-09T12:13:52.374+05:30    REPORT RequestId: c323e144-f698-443a-b125-35eb0f43a55e Duration: 1.25 ms Bi
```

- Scope and Assumptions

Scope:

The scope of this project is to develop and deploy an application on the AWS cloud infrastructure. This involves creating an AWS account, setting up a virtual network with specific requirements, deploying a virtual machine (VM) with an application, ensuring secure communication between the VM and the database (RDS) with the help of private subnet, deploying the application to Elastic Beanstalk service, and creating a Lambda function that gets triggered when any file is uploaded in an S3 bucket.

Assumptions:

- Infrastructure Requirements Only: Given that the project only provides information about the infrastructure requirements and does not disclose any details about the application's specific functionality or purpose, the development tasks will focus solely on setting up and configuring the necessary components to support the application. These tasks will include creating a development environment, configuring the database system, implementing backend and frontend functionality, setting up deployment and infrastructure management processes.

- AWS Account Access: The developer undertaking this project has the necessary credentials and access to create and configure AWS resources, including EC2 instances, VPCs, subnets, RDS instances, Elastic Beanstalk applications, and Lambda functions.

- Security Group and Network Access Control List (NACL) Rules: Appropriate security group and NACL rules are created to allow the necessary communication between resources while maintaining the required security and privacy.

- AWS Services Availability: All required AWS services (VPC, EC2, RDS, Elastic Beanstalk, S3, and Lambda) are available in the region chosen for deployment and can be accessed by the developer.

- S3 Bucket Configuration: The S3 bucket is created and provided necessary rights to the developer, and the developer has permissions to upload files to the bucket. The bucket should be properly configured to trigger the Lambda function upon file uploads.

- File Name Retrieval in Lambda: The Lambda function will be specifically designed to retrieve and print the name of the uploaded file. However, any other specific actions or operations on the uploaded file are considered to be outside the current scope of the project.

- Cost and Billing: The developer is aware of the costs associated with the services used in the project and takes necessary precautions to avoid incurring unexpected expenses.

- Backup and Disaster Recovery: The project does not specify any backup and disaster recovery requirements explicitly, indicating that these aspects are not within the scope of the project. Consequently, the main focus will be on deploying and ensuring the functionality of the application.