OWNCLOUD

Creating a file share & sync solution using OwnCloud and AWS

Created By:- vinay chawan

Project Objectives and Requirements

Recent research has found that a significant number of employees (40-75%) use Dropbox to share files both within and outside of their companies. Alarmingly, half of these Dropbox users are aware that this practice violates their company's rules, yet continue to do so. More than 40% of businesses have experienced the exposure of confidential information, and the estimated average cost of a data breach was $5.5 Million in 2011. These files, which contain sensitive company and customer data, are stored in a public cloud, beyond the control of the businesses - possibly even outside of the country. The potential for data leakage and security breaches is enormous, and companies must remain compliant with their policies and procedures for security and governance.

OwnCloud is a file server that enables secure storage, collaboration, and sharing. It's convenient to store files in the cloud, making them available on any device and easy to share. Many popular providers like Google, Apple, Facebook, Twitter, and Dropbox offer these services. However, with many of these vendors, files are stored and processed beyond users' control. For U.S. firms, files are subject to the Cloud Act and, therefore, to government snooping.

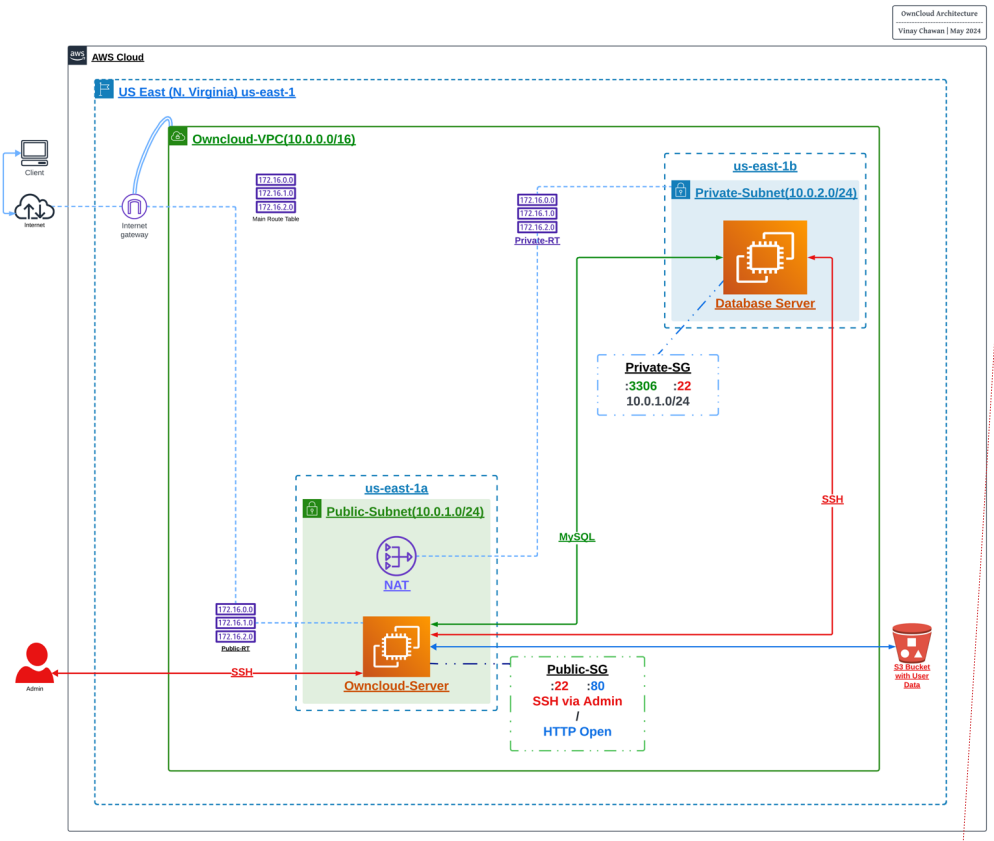
Built on open standards and modularity, ownCloud is suitable as a robust and security-oriented file platform for both proprietary and open-source software environments. As a result, ownCloud combines fully sovereign file storage with modern, efficient productivity tools while maintaining auditable source code and comprehensive release functions. Users can install ownCloud themselves or rent a managed instance.

By using OwnCloud, you can benefit from the conveniences of public clouds while being compliant and in control of your data.

Hence, we will use a bastion host to securely connect to the database of the OwnCloud server.

A bastion host is a server used to provide access to a private network from an external network, such as the Internet. Because of its exposure to potential attack, a bastion host must minimize the chances of penetration. For example, you can use a bastion host to mitigate the risk of allowing SSH.

Architecture Diagram



Commented [vc1]: Note: Main Route table is create by default, when a VPC is created.

Services Used

- **Networking Services:**
  - **VPC (Virtual Private Cloud)**
  - **Subnets**
  - **Route Table**
  - **Internet Gateway (IGW)**
  - **NAT Gateway**

- **Compute Services:**
  - **EC2(Elastic Compute Cloud)**
  - **Security Group**
  - **Ubuntu 22.04(Free Tier)**

- **Storage & IAM:**
  - **S3 Bucket (Simple Storage Service)**
  - **IAM (Identity Access Management)**

- **Security Considerations**:
  - VPC (10.0.0.0/16)
  - Database (Private IP)
  - IGW (Internet Gate for internet access) attachment on VPC for open world access to web application (OwnCloud Server). Also, for updating/installation of packages.
  - Open HTTP to OwnCloud Server for Web Applications access.
  - SSH via My IP for Admin on Public Subnet.
  - NAT for Setup/Update of packages on Database server via Public Subnet.
  - MySQL connection between Public and Private via port 3306.
  - SSH connection to Database server via Bastion Host (OwnCloud Server).
  - No public IP address assignment to the Database server.

Network Configuration/Setup

- VPC (Virtual Private Cloud):
  - Name: OwnCloud-VPC

- Subnet:
  - Public (us-east-1a) (10.0.1.0/24) (Enabled Public IP assignment)
  - Private (us-east-1b) (10.0.2.0/24) (Only Private IP)

- Route Table:
  - Create “Public” route table.
  - Create “Private” route table.

- Internet Gateway:
  - Create Internet gateway (IGW)

- NAT Gateway:
  - Create NAT gateway (NAT)

Commented [vc2]: Resource Map Page 18

Commented [vc3]: Page 10

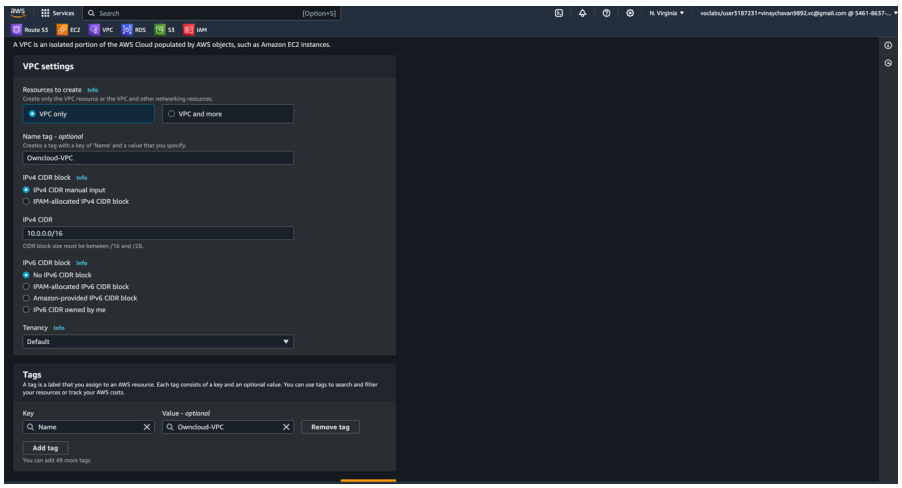
Commented [vc4]: Page 12

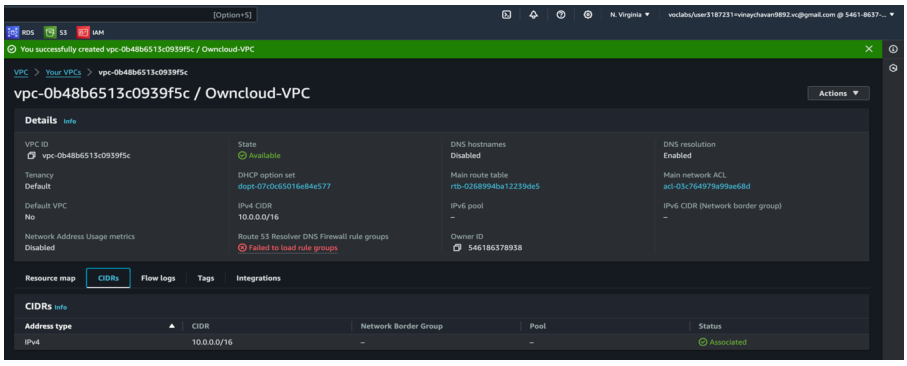
Commented [vc5]: Page 15

Installation and configuration process

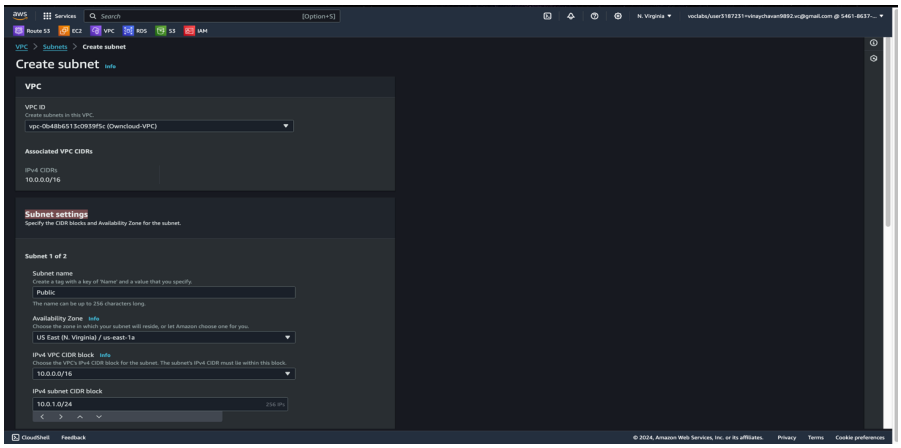
- Network Setup/Configuration

  - VPC:
    - Create Virtual Private Cloud (VPC) as OwnCloud-VPC.
    - Assign IPv4 CIDR as 10.0.0.0/16
    - Keep rest of the setting Default.
    - Add the Name tag with OwnCloud-VPC.
    - Click on “Create VPC”.



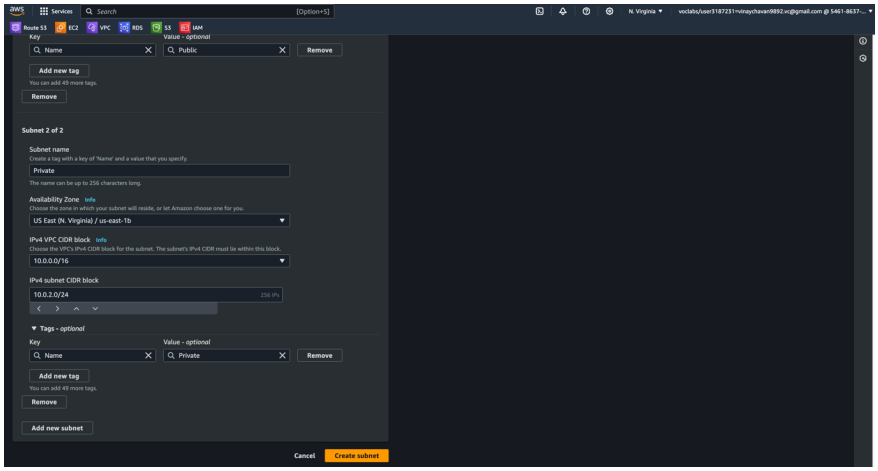


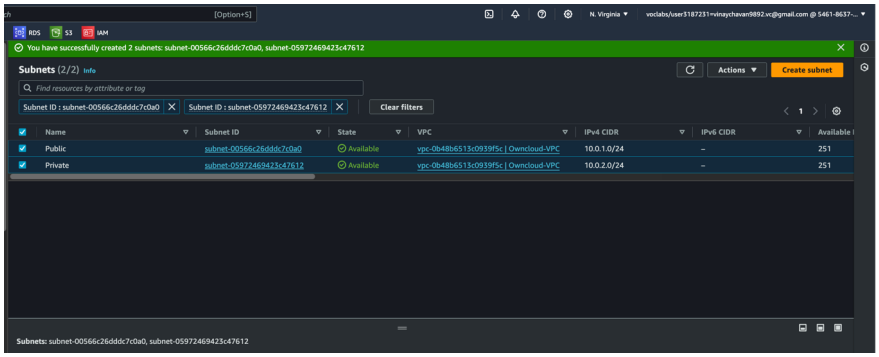
- Subnet:
  - Create two subnets (Public and Private).
  - Click on “Create Subnet”
  - Select the VPC(OwnCloud-VPC)
  - First subnet name is Public with AZ is us-east-1a and IPv4 is 10.0.1.0/24.
  - Tag Name as Public.
  - Click on “Add new Subnet”.
  - Second subnet name is Private with AZ is us-east-1b and IPv4 is 10.0.2.0/24
  - Tag Name as Private.
  - Click on “Create Subnet”.
  - Tick the Public Subnet.
  - Open “Actions”, Click on “Edit Subnet Settings”.
  - Tick on “Enable Auto Assign public IPv4 Address”.
  - Save.



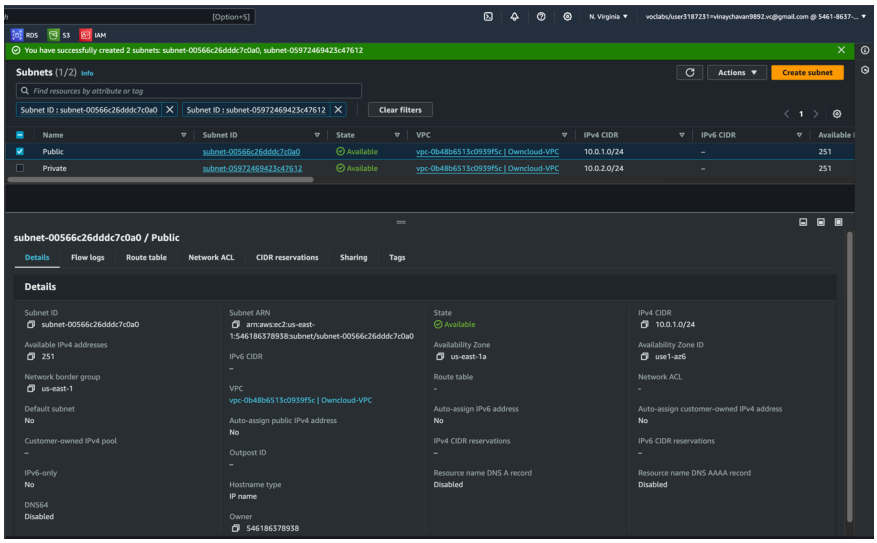
Commented [vc6]: Public Subnet

Commented [vc7]: Private Subnet

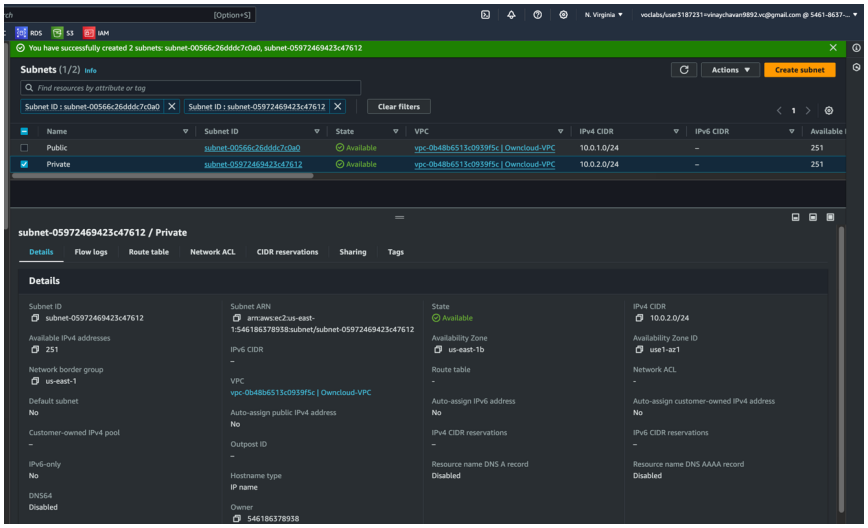

Services
Search
[Option+S]
N. Virginia
Route 53
EC2
VPC
RDS
S3
IAM
Key
Value - optional
Name
Public
Remove
Add new tag
You can add 49 more tags.
Remove
Subnet 2 of 2
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
Private
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1b
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16
IPv4 subnet CIDR block
10.0.2.0/24
256 IPs
Tags - optional
Key
Value - optional
Name
Private
Remove
Add new tag
You can add 49 more tags.
Remove
Add new subnet
Cancel
Create subnet

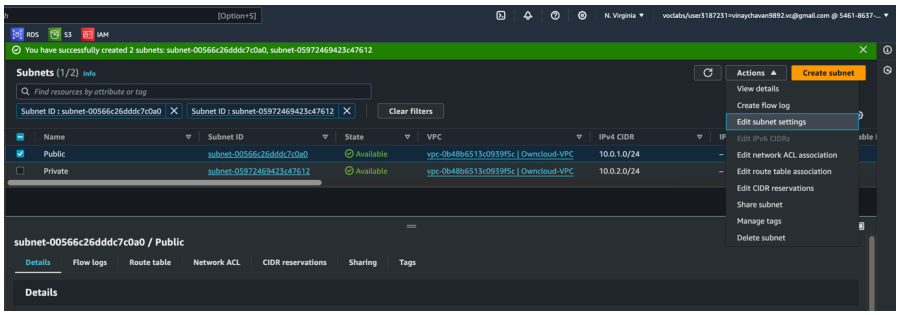

[Option+S]
N. Virginia
RDS
S3
IAM
You have successfully created 2 subnets: subnet-00566c26dddc7c0a0, subnet-05972469423c47612
Subnets (2/2) Info
Actions
Create subnet
Find resources by attribute or tag
Subnet ID : subnet-00566c26dddc7c0a0
Subnet ID : subnet-05972469423c47612
Clear filters
Name
Subnet ID
State
VPC
IPv4 CIDR
IPv6 CIDR
Public
subnet-00566c26dddc7c0a0
Available
vpc-0b48b6513c0939f5c | Owncloud-VPC
10.0.1.0/24
251
Private
subnet-05972469423c47612
Available
vpc-0b48b6513c0939f5c | Owncloud-VPC
10.0.2.0/24
251
Subnets: subnet-00566c26dddc7c0a0, subnet-05972469423c47612

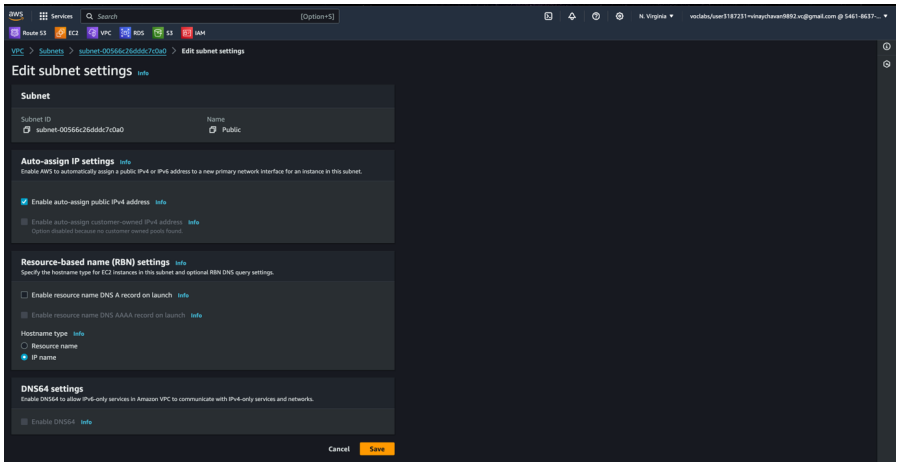
Public Subnet:



Private Subnet:




You have successfully created 2 subnets: subnet-00566c26dddc7c0a0, subnet-05972469423c47612
Subnets (1/2) Info
Actions
Create subnet
View details
Create flow log
Edit subnet settings
Edit IPv6 CIDRs
Edit network ACL association
Edit route table association
Edit CIDR reservations
Share subnet
Manage tags
Delete subnet
Clear filters
Public
Private
subnet-00566c26dddc7c0a0 / Public
Details
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags


Edit subnet settings Info
Subnet
Auto-assign IP settings Info
Enable auto-assign public IPv4 address Info
Resource-based name (RBN) settings Info
Enable resource name DNS A record on launch Info
Hostname type Info
Resource name
IP name
DNS64 settings
Cancel
Save

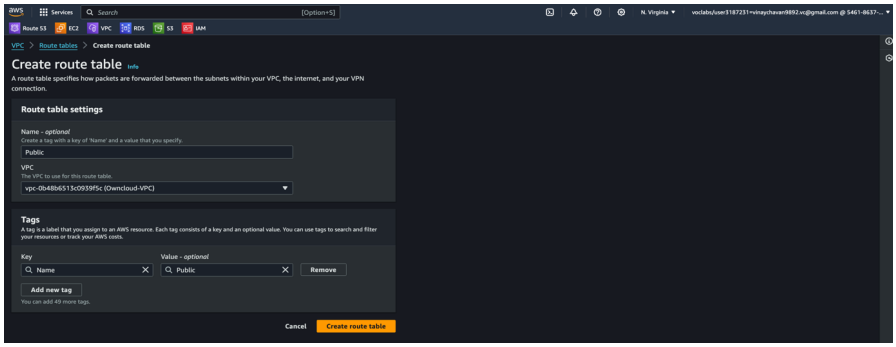
❖ Route Table:

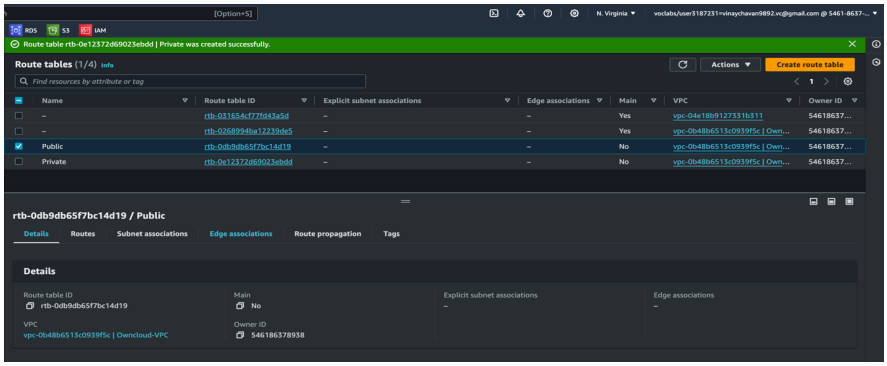
- Create two Route tables.
- Click on “Create Table”
- First table name is Public, Select the OwnCloud-VPC.
- Tag as per requirement.
- Click on “Create route table”.
- Select “Public” route table.
- Click on “Subnet Association”
- Click on “Edit subnet Associations”.
- Select “Public” Subnet and save Association.
- Again, click on “Create Table”.
- Second table name is Private, Select the OwnCloud-VPC.
- Tag as per requirement.
- Click on “Create route table”.

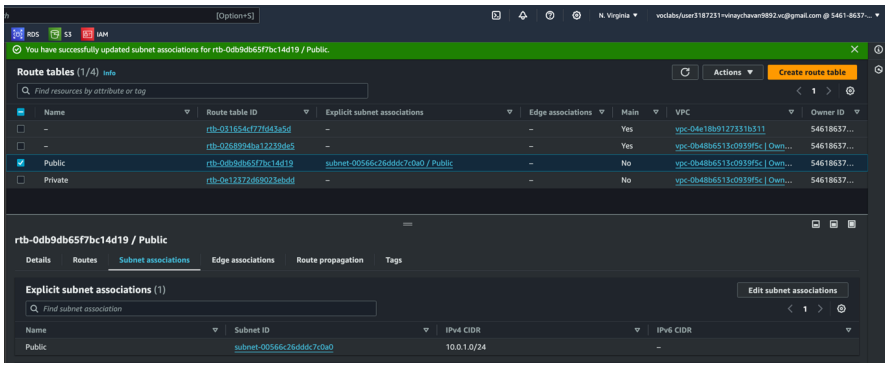
Commented [vc8]: Public Route Table

Commented [vc9]: Private Route table

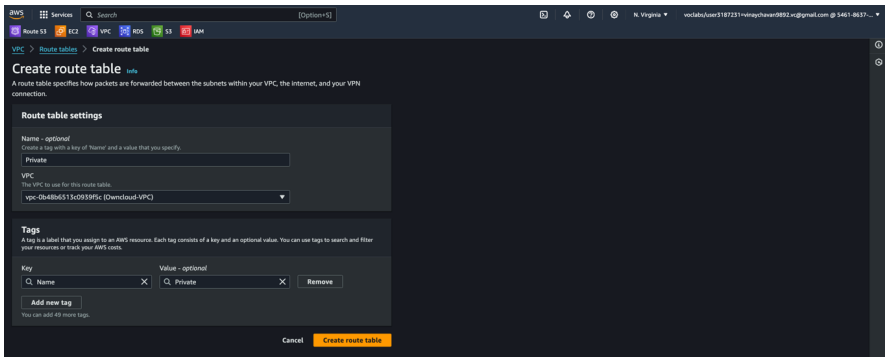
Public Route Table:

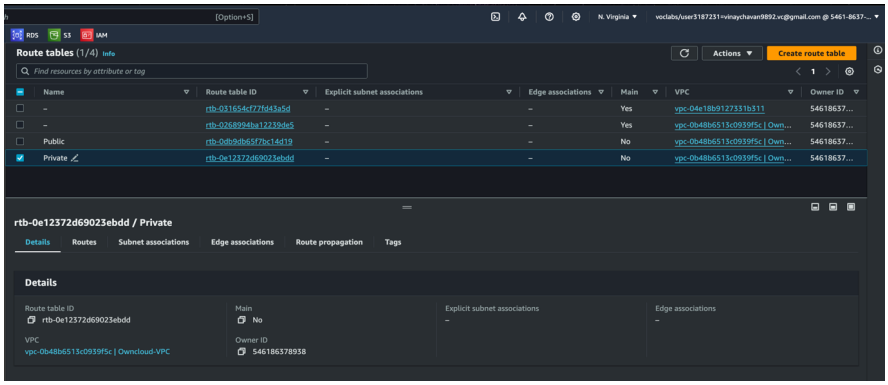


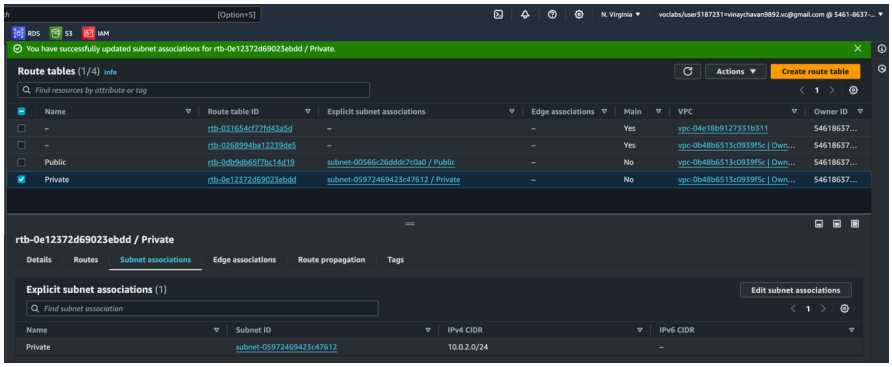

Route table rtb-0e12372d69023ebdd | Private was created successfully.
Route tables (1/4) Info
Actions
Create route table
Name
Route table ID
Explicit subnet associations
Edge associations
Main
VPC
Owner ID
rtb-031654cf77fd43a5d
rtb-0268994ba12239de5
Public
rtb-0db9db65f7bc14d19
Private
rtb-0e12372d69023ebdd
rtb-0db9db65f7bc14d19 / Public
Details
Routes
Subnet associations
Edge associations
Route propagation
Tags
Route table ID
rtb-0db9db65f7bc14d19
Main
No
Explicit subnet associations
Edge associations
VPC
vpc-0b48b6513c0939f5c | Owncloud-VPC
Owner ID
546186378938


You have successfully updated subnet associations for rtb-0db9db65f7bc14d19 / Public.
Route tables (1/4) Info
Actions
Create route table
Name
Route table ID
Explicit subnet associations
Edge associations
Main
VPC
Owner ID
rtb-031654cf77fd43a5d
rtb-0268994ba12239de5
Public
rtb-0db9db65f7bc14d19
subnet-00566c26dddc7c0a0 / Public
Private
rtb-0e12372d69023ebdd
rtb-0db9db65f7bc14d19 / Public
Details
Routes
Subnet associations
Edge associations
Route propagation
Tags
Explicit subnet associations (1)
Edit subnet associations
Name
Subnet ID
IPv4 CIDR
IPv6 CIDR
Public
subnet-00566c26dddc7c0a0
10.0.1.0/24

Private Route Table:

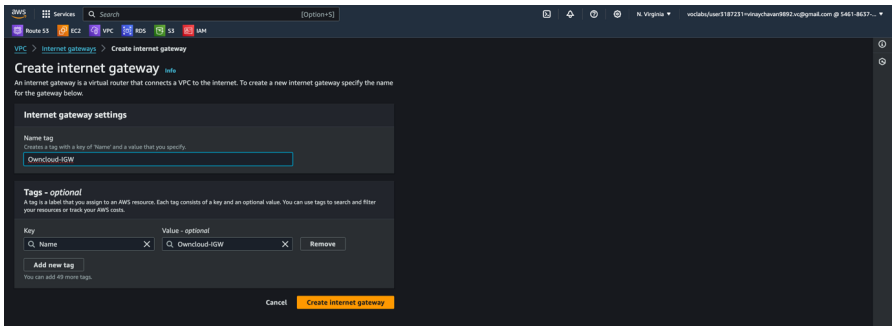


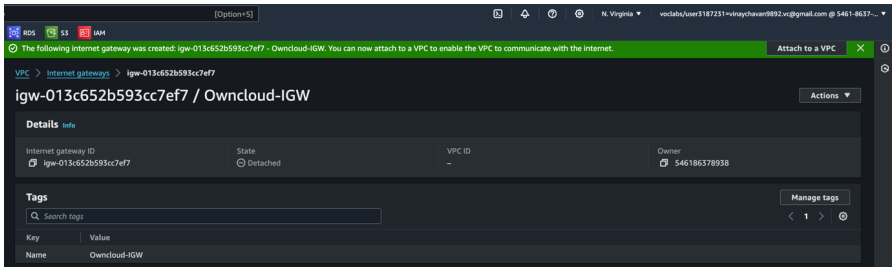


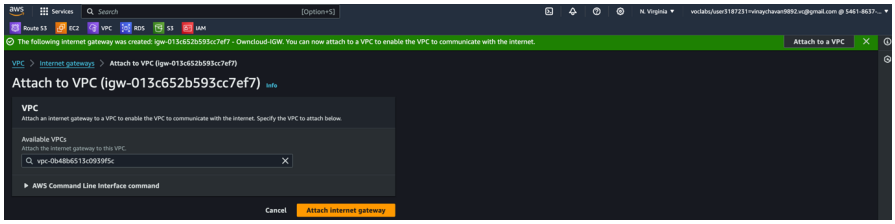


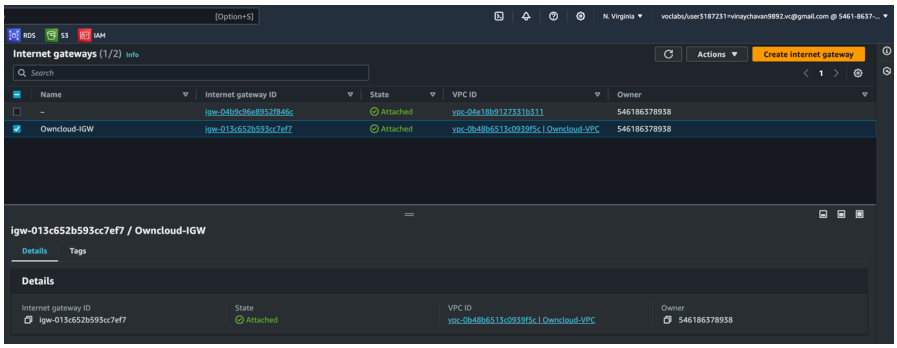
❖ **Internet Gateways (IGW)**

- Click on Internet Gateways.
- Click on “Internet gateway”.
- Name and tag the internet gateway as per requirement.
- Click on Create Internet gateway.
- Click on “Attach to a VPC”
- Select the OwnCloud-VPC.
- Click on Attach internet gateway.
- Click on Route tables.
- Select “Public” route table.
- Click on “Routes” tab.
- Click on Edit routes.
- Click on Add route.
- Select 0.0.0.0/0, Select “Internet gateway” and Select “IGW-ID”
- Then Save changes.

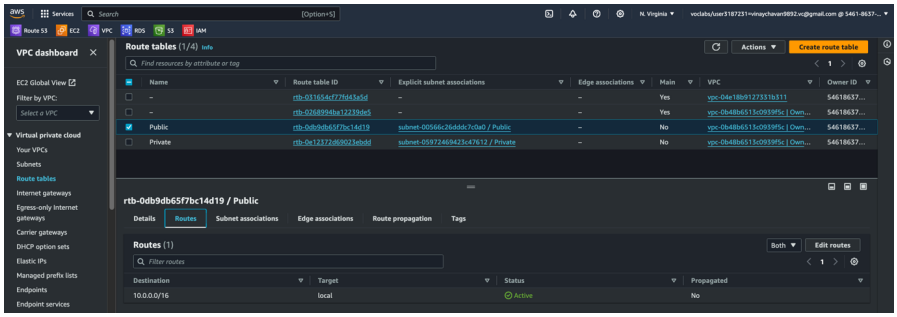


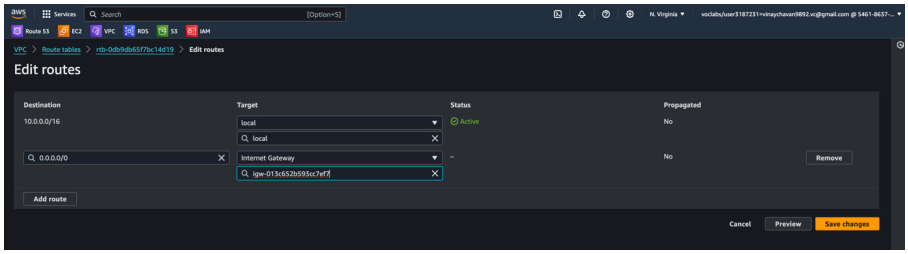


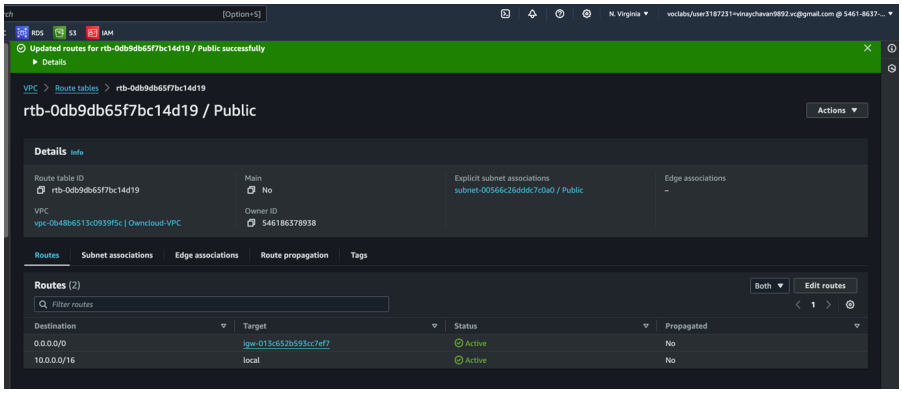




Adding IGW to Public route table:



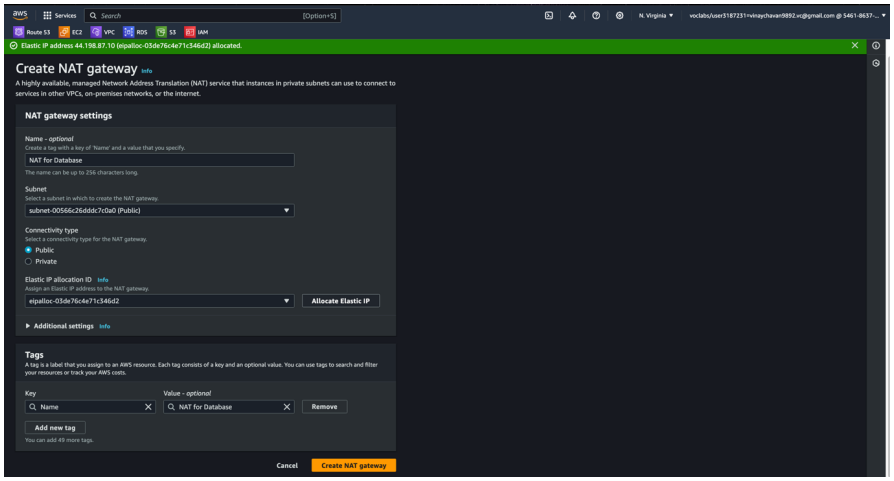


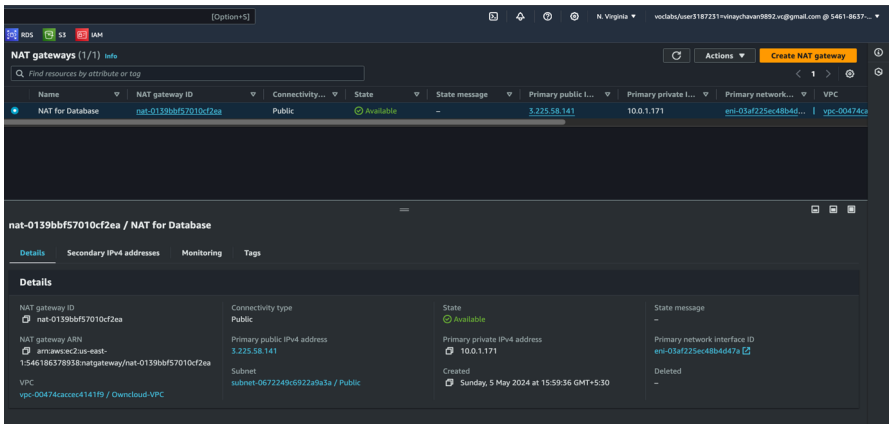


❖ NAT Gateways

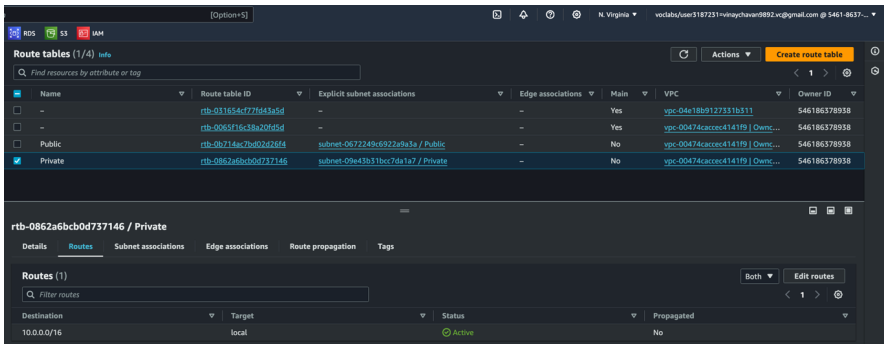
- Click on NAT Gateways.
- Click on Create NAT gateway.
- Provide a name to NAT gateway.
- Select “Public” Subnet.
- Connective Type is Public.
- Click on Allocate Elastic IP.
- Click On “Create NAT Gateway”.
- Then go to Route table.
- Select “Private” route table.
- Click on “Routes” tab.
- Click on Edit routes.
- Click on Add route.
- Select 0.0.0.0/0, Select “NAT gateway” and select “NAT for Database”.
- Then Save changes.

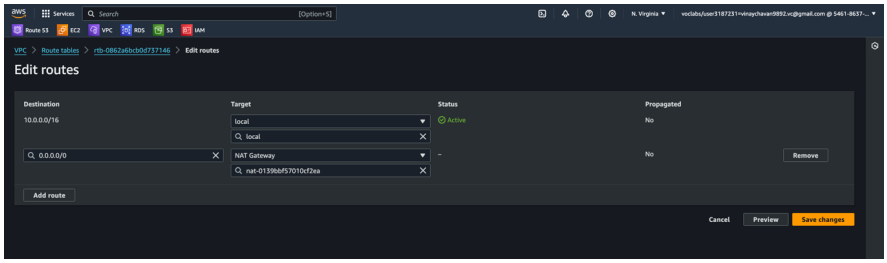
Commented [vc10]: Page 17


Elastic IP address 44.198.87.10 (eipalloc-05de76c4e71c346d2) allocated.
Create NAT gateway
A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.
NAT gateway settings
Name - optional
Create a tag with a key of 'Name' and a value that you specify.
NAT for Database
The name can be up to 256 characters long.
Subnet
Select a subnet in which to create the NAT gateway.
subnet-00566c26dddc7c0a0 (Public)
Connectivity type
Select a connectivity type for the NAT gateway.
Public
Private
Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
eipalloc-05de76c4e71c346d2
Allocate Elastic IP
Additional settings Info
Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
Key
Value - optional
Name
NAT for Database
Remove
Add new tag
You can add 49 more tags.
Cancel
Create NAT gateway

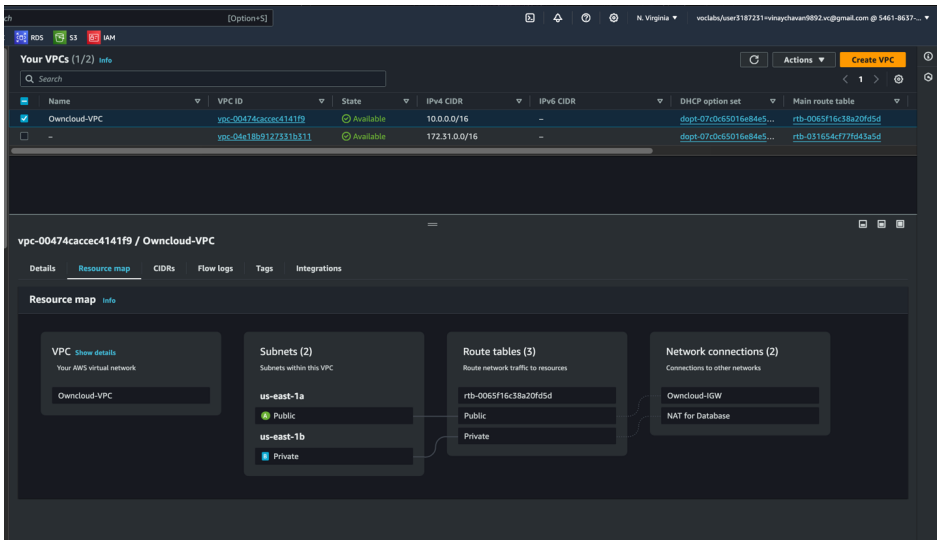

NAT gateways (1/1) Info
Actions
Create NAT gateway
Find resources by attribute or tag
Name
NAT gateway ID
Connectivity...
State
State message
Primary public I...
Primary private I...
Primary network...
VPC
NAT for Database
nat-0139bbf57010cf2ea
Public
Available
3.225.58.141
10.0.1.171
eni-03af225ec48b4d...
nat-0139bbf57010cf2ea / NAT for Database
Details
Secondary IPv4 addresses
Monitoring
Tags
Details
NAT gateway ID
nat-0139bbf57010cf2ea
Connectivity type
Public
State
Available
State message
-
NAT gateway ARN
arn:aws:ec2:us-east-1:546186378938:natgateway/nat-0139bbf57010cf2ea
Primary public IPv4 address
3.225.58.141
Primary private IPv4 address
10.0.1.171
Primary network interface ID
eni-03af225ec48b4d47a
VPC
vpc-00474caccec4141f9 / Owncloud-VPC
Subnet
subnet-0672249c6922a9a3a / Public
Created
Sunday, 5 May 2024 at 15:59:36 GMT+5:30
Deleted
-

Adding NAT Gateway to Private route table:





Owncloud-VPC Resource map:



Compute and Instances

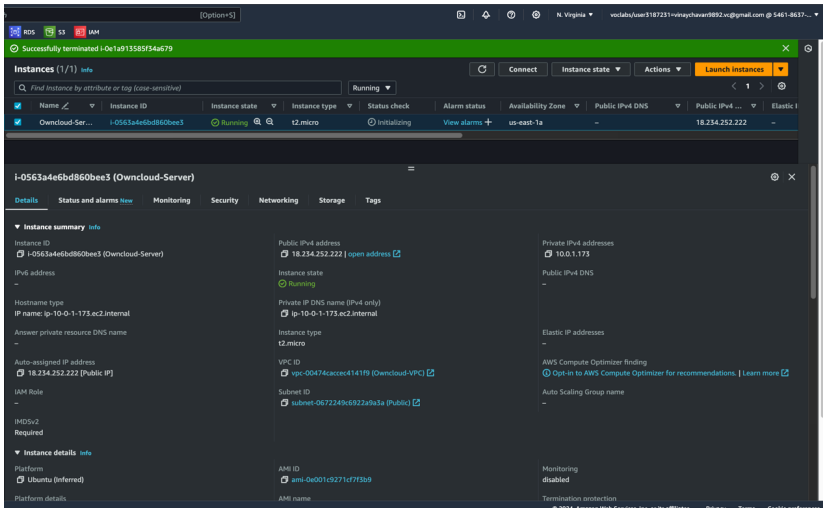
- **OwnCloud-Server:**
  - **Server Name: Owncloud-Server**
  - **AMI: Ubuntu 22.04**
  - **Key Pair creation**
  - **VPC: OwnCloud-VPC.**
  - **Subnet: Public - 10.0.1.0/24.**
  - **AZ:** us-east-1a.
  - Need IGW connection.
  - **Security Group as follow: SSH via Own IP and HTTP open (0.0.0.0).**
  - **Connection to S3 bucket for storage space.**

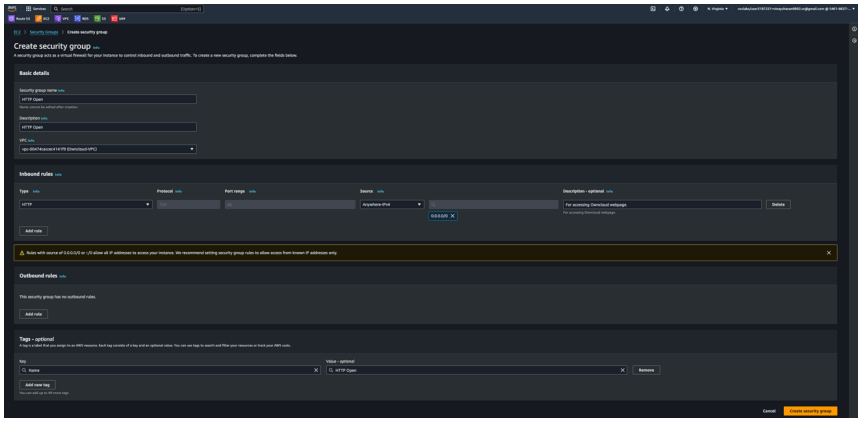
- **Database Server:**
  - **Server name: Database**
  - **AMI: Ubuntu 22.04**
  - **Key Pair creation**
  - **VPC: OwnCloud-VPC**
  - **Subnet: Private - 10.0.2.0/24**
  - **AZ:** us-east-1b
  - **Needs NAT package installation.**
  - **Security Group as follow: SSH via Bastion Host (Owncloud-Server) 10.0.1.0/24:22 and MySQL/Aurora:3306 Source:sg-0bfe9fe1cc50a3fb5/SSH via My IP.**

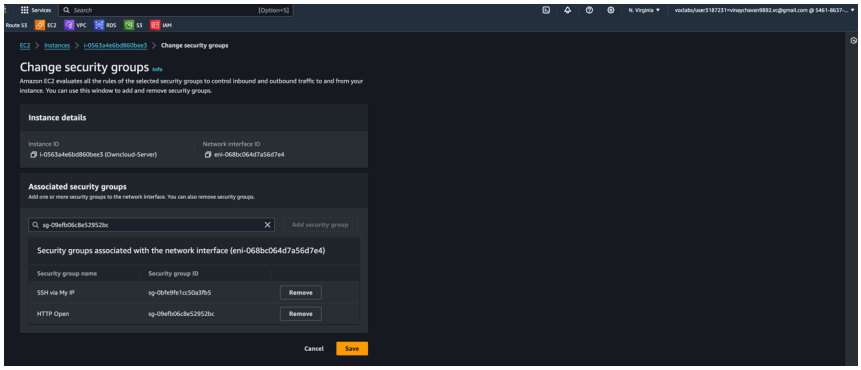
EC2(Elastic Compute Cloud)

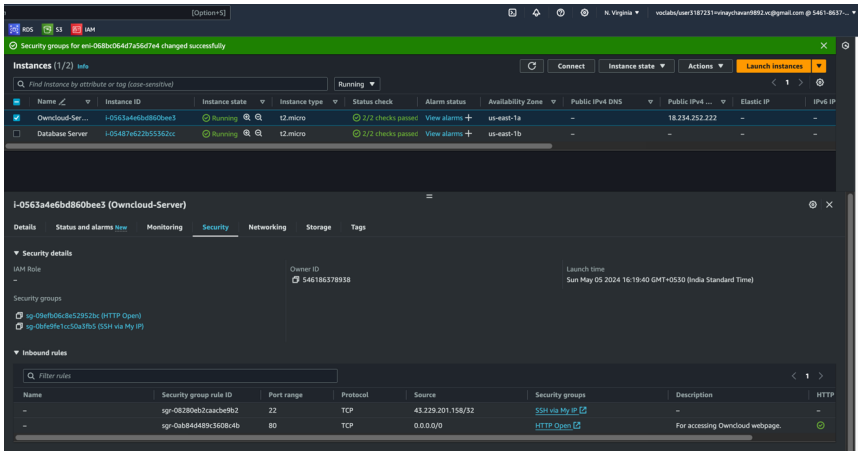
❖ Owncloud-Server

- First one is “Owncloud-Server”.
- Click on Launch instance in EC2.
- Name first instance as “Owncloud-Server”.
- Select AMI as “Ubuntu Server 22.04 LTS”
- Create a Key Pair.
- In Network Settings, select “OwnCloud-VPC”.
- Subnet as Public.
- Confirm the “Auto assign Public IP” is Enabled.
- Create a security group of “SSH via my IP”.
- SSH :22 Source type: My IP.
- Rest settings as default.
- Click on Launch Instance.
- Create New security group name “HTTP Open”.
- HTTP :80 Source: Anywhere-IPv4-0.0.0.0/0.
- Attach the HTTP open SG to Owncloud-Server instance.



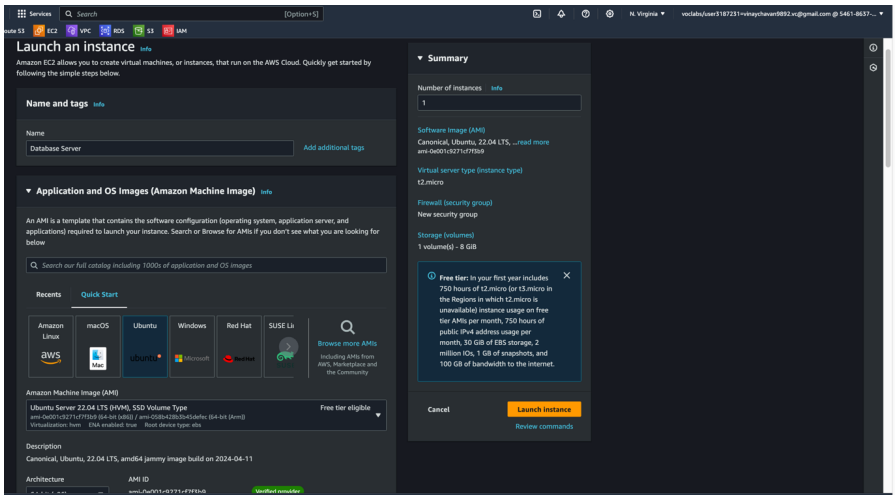

Create security group
Basic details
Inbound rules
Outbound rules
Tags - optional
Add rule
Add new tag
Cancel
Create security group

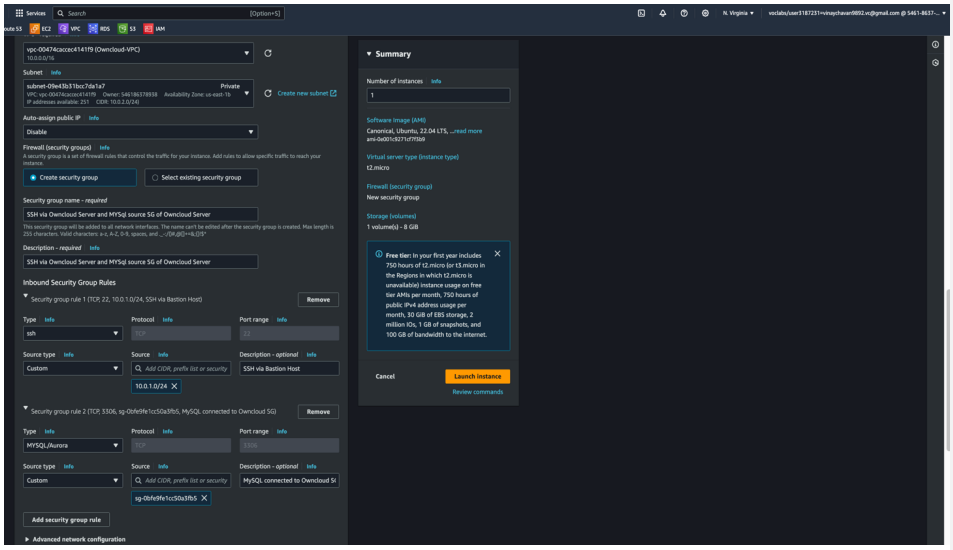

Change security groups
Amazon EC2 evaluates all the rules of the selected security groups to control inbound and outbound traffic to and from your instance. You can use this window to add and remove security groups.
Instance details
Instance ID
i-0563a4e6bd860bee3 (Owncloud-Server)
Network interface ID
eni-068bc064d7a56d7e4
Associated security groups
Add one or more security groups to the network interface. You can also remove security groups.
sg-09efb06c8e52952bc
Add security group
Security groups associated with the network interface (eni-068bc064d7a56d7e4)
Security group name
Security group ID
SSH via My IP
sg-0bfe9fe1cc50a3fb5
Remove
HTTP Open
sg-09efb06c8e52952bc
Remove
Cancel
Save

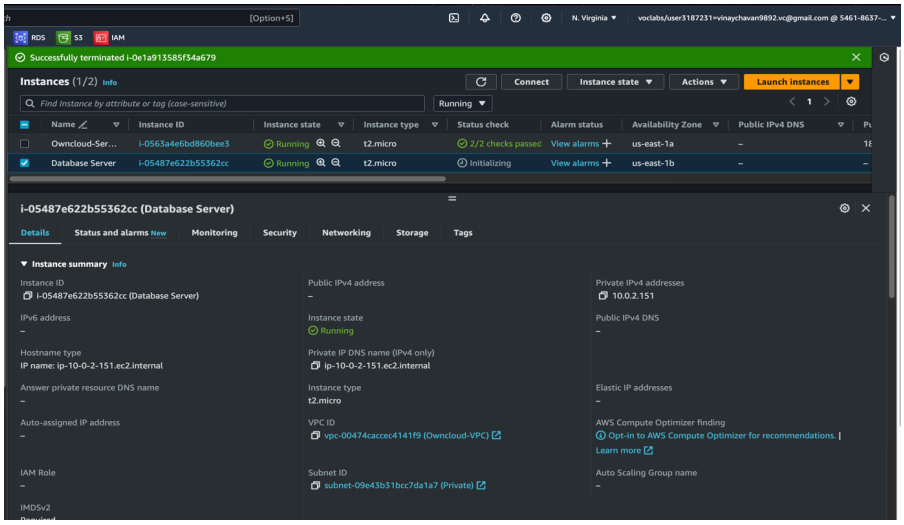

Security groups for eni-068bc064d7a56d7e4 changed successfully
Instances (1/2)
Connect
Instance state
Actions
Launch instances
Owncloud-Ser...
i-0563a4e6bd860bee3
Database Server
i-05487e622b55362cc
i-0563a4e6bd860bee3 (Owncloud-Server)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Security details
IAM Role
Owner ID
546186378938
Launch time
Sun May 05 2024 16:19:40 GMT+0530 (India Standard Time)
Security groups
sg-09efb06c8e52952bc (HTTP Open)
sg-0bfe9fe1cc50a3fb5 (SSH via My IP)
Inbound rules
Filter rules
Name
Security group rule ID
Port range
Protocol
Source
Security groups
Description
sgr-08280eb2caacbe9b2
22
TCP
43.229.201.158/32
SSH via My IP
sgr-0ab84d489c3608c4b
80
TCP
0.0.0.0/0
HTTP Open
For accessing Owncloud webpage.

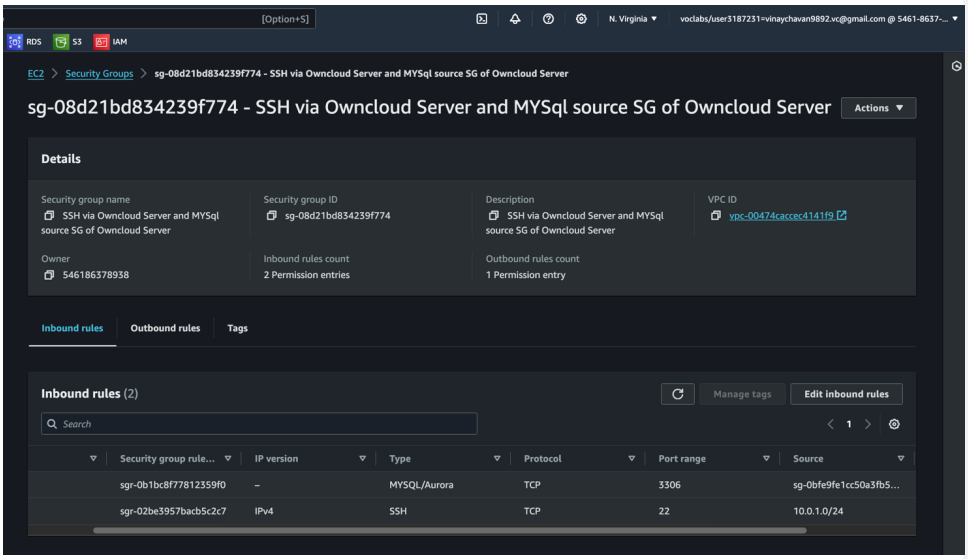
❖ Database Server

- Second is “Database Server”.
- Click on Launch instance in EC2.
- Name first instance as “Database Server”.
- Select AMI as “Ubuntu Server 22.04 LTS”
- Create a Key Pair.
- In Network Settings, select “OwnCloud-VPC”.
- Subnet as Private.
- Confirm the “Auto assign Public IP” is Disabled.
- Create a security group of “SSH via Owncloud Server and MYSql source SG of Owncloud Server”.
- SSH :22 Source:10.0.1.0/24.
- MySQL/Aurora :3306 Source:sg-0bfe9fe1cc50a3fb5 / SSH via My IP
- Rest settings as default.
- Click on Launch Instance.




Summary
Number of instances
Software Image (AMI)
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance
Review commands


Successfully terminated i-0e1a913585f34a679
Instances (1/2)
Connect
Instance state
Actions
Launch instances
Oncloud-Ser...
i-0563a4e6bd860bee3
Database Server
i-05487e622b55362cc
t2.micro
2/2 checks passed
Initializing
us-east-1a
us-east-1b
i-05487e622b55362cc (Database Server)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary
Instance ID
i-05487e622b55362cc (Database Server)
Public IPv4 address
Private IPv4 addresses
10.0.2.151
Instance state
Running
Public IPv4 DNS
Private IP DNS name (IPv4 only)
ip-10-0-2-151.ec2.internal
Hostname type
IP name: ip-10-0-2-151.ec2.internal
Instance type
t2.micro
Elastic IP addresses
VPC ID
vpc-00474caccec4141f9 (Oncloud-VPC)
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
Learn more
Subnet ID
subnet-09e43b31bcc7da1a7 (Private)
Auto Scaling Group name
IAM Role
IMDSv2


[Option+S]
N. Virginia
voclabs/user3187231=vinaychavan9892.vc@gmail.com @ 5461-8637-...
RDS
S3
IAM
EC2 > Security Groups > sg-08d21bd834239f774 - SSH via Owncloud Server and MYSql source SG of Owncloud Server
sg-08d21bd834239f774 - SSH via Owncloud Server and MYSql source SG of Owncloud Server
Actions
Details
Security group name
SSH via Owncloud Server and MYSql source SG of Owncloud Server
Security group ID
sg-08d21bd834239f774
Description
SSH via Owncloud Server and MYSql source SG of Owncloud Server
VPC ID
vpc-00474caccec4141f9
Owner
546186378938
Inbound rules count
2 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Tags
Inbound rules (2)
Manage tags
Edit inbound rules
Search
1
Security group rule...
IP version
Type
Protocol
Port range
Source
sgr-0b1bc8f77812359f0
-
MYSQL/Aurora
TCP
3306
sg-0bfe9fe1cc50a3fb5...
sgr-02be3957bacb5c2c7
IPv4
SSH
TCP
22
10.0.1.0/24

Setup/Configuration of Database Server and OwnCloud Server

Note: Perform below steps before setup/configuration of Servers.

(Database Server)

- Instance ID:- i-05487e622b55362cc
- Open an SSH client.
- Locate your private key file. The key used to launch this instance is Database.pem
- Run this command, if necessary, to ensure your key is not publicly viewable.
- chmod 400 "Database.pem"
- Connect to your instance using its Private IP:10.0.2.151
- Example: ssh -i "Database.pem" ubuntu@10.0.2.151

(Owncloud-Server)

- Instance ID:- i-0563a4e6bd860bee3
- Open an SSH client.
- Locate your private key file. The key used to launch this instance is Owncloud.pem
- Run this command, if necessary, to ensure your key is not publicly viewable.
- chmod 400 "Owncloud.pem"
- Connect to your instance using its Public IP:54.224.239.149
- Example: ssh -i "Owncloud.pem" ubuntu@54.224.239.149

Point's to be Noted:

- Since the Database server is in a Private Network by using Bastion host (Owncloud-Server), will need to take SSH.
- Hence need to send Database.pem to Owncloud-Server.
- Run this command, to send the Database.pem to Owncloud-Server.

scp -i Owncloud.pem ./Database.pem ubuntu@54.224.239.149:/home/ubuntu/Database.pem

❖ **Database Server**

- SSH to Bastion host (Owncloud-Server).
- Check the "Database.pem' in home directory /home/ubuntu.
- ls -l
- ssh -i "Database.pem" ubuntu@10.0.2.151
- sudo su
- sudo apt update
- sudo apt install mysql-server
- sudo systemctl start mysql.service
- sudo mysql
- ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';
- Exit
- sudo mysql_secure_installation

Commented [vc11]: Installation of MySQL Server.

Output

Securing the MySQL server deployment.
Connecting to MySQL using a blank password.
VALIDATE PASSWORD COMPONENT can be used to test passwords and improve security. It checks the strength of password.
and allows the users to set only those passwords which are secure enough. Would you like to setup VALIDATE PASSWORD component?
Press y|Y for Yes, any other key for No: Y
There are three levels of password validation policy:LOW Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 2
Please set the password for root here.
New password:
Re-enter new password:
Estimated strength of the password: 100
Do you wish to continue with the password provided? (Press y|Y for Yes, any other key for No) : Y
By default, a MySQL installation has an anonymous user, allowing anyone to log into MySQL without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother.
You should remove them before moving into a production environment.
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.
Normally, root should only be allowed to connect from'localhost'. This ensures that someone cannot guess at the root password from the network.
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.
By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing and should be removed before moving into a production environment.
Remove test database and access to it? (Press y|Y for Yes, any other key for No): y
- Dropping test database...
Success.
- Removing privileges on test database...
Success.
Reloading the privilege tables will ensure that all changes made so far will take effect immediately.
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.
All done!

- mysql -u root -p
- CREATE USER 'Admin'@'10.0.1.0/24' IDENTIFIED BY 'June#2024';
- create database owncloud;
  Query OK, 1 row affected (0.01 sec)
- MySQL [(none)]> show databases;
  +--------------------+
  | Database |
  +--------------------+
  | information_schema |
  | mysql |
  | owncloud |
  | performance_schema |
  | sys |
  +--------------------+
  5 rows in set (0.006 sec)
- GRANT ALL PRIVILEGES ON *.* TO 'Admin'@'10.0.1.0/24' WITH GRANT OPTION;
- FLUSH PRIVILEGES;
- Exit

Commented [vc12]: Creation of owncloud database for Owncloud Server.

- nano /etc/mysql/mysql.conf.d/mysqld.cnf
  #bind-address = 127.0.0.1
  #mysqlx-bind-address = 127.0.0.1
- systemctl restart mysql.service
- Exit
- Exit

Commented [vc13]: Put # for these entries in .cnf file.

Commented [vc14]: Exit from database server and go to Owncloud Server.

❖ **OwnCloud Server**

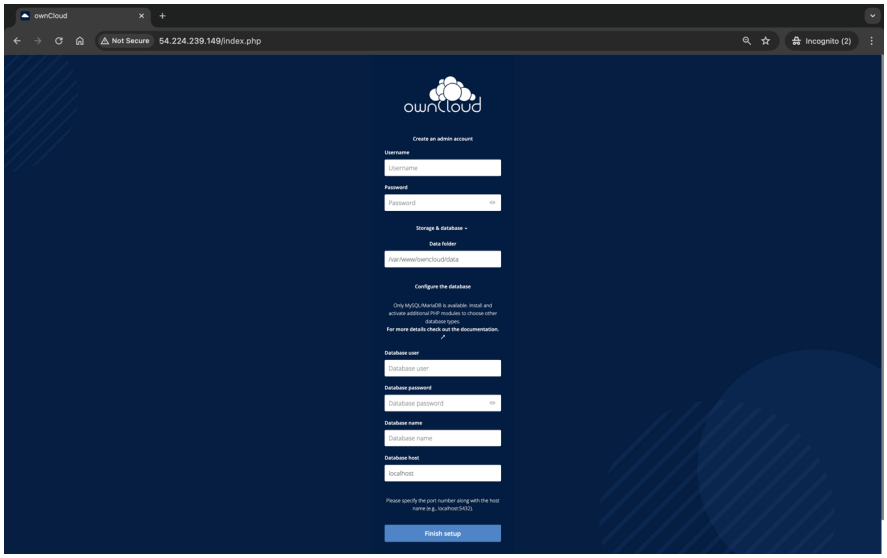
- sudo apt update
- sudo add-apt-repository ppa:ondrej/php -y
- sudo apt update
- sudo apt install -y apache2 libapache2-mod-php7.4 mariadb-server openssl redis-server wget php7.4 php7.4-imagick php7.4-common php7.4-curl php7.4-gd php7.4-imap php7.4-intl php7.4-json php7.4-mbstring php7.4-gmp php7.4-bcmath php7.4-mysql php7.4-ssh2 php7.4-xml php7.4-zip php7.4-apcu php7.4-redis php7.4-ldap php-phpseclib
- sudo a2enmod dir env headers mime rewrite setenvif
- sudo systemctl restart apache2
- root@ip-10-0-1-173:/var/www/html# rm *
- root@ip-10-0-1-173:/var/www/html# cd ..
- root@ip-10-0-1-173:/var/www# rmdir html
- root@ip-10-0-1-173:/var/www# sudo wget https://download.owncloud.com/server/stable/owncloud-complete-latest.tar.bz2
- sudo tar -xjf owncloud-complete-latest.tar.bz2
- sudo chown -R www-data. Owncloud
- root@ip-10-0-1-173:/var/www# rm owncloud-complete-latest.tar.bz2
- root@ip-10-0-1-173:/var/www# nano /etc/apache2/sites-enabled/000-default.conf
  DocumentRoot /var/www/owncloud
- sudo apt-get install mysql-client -y
- sudo systemctl restart apache2
- Exit

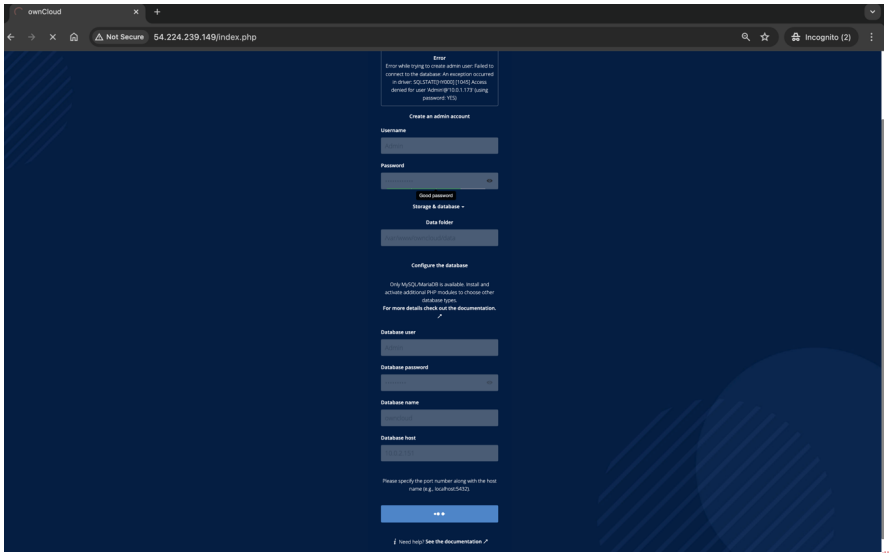
Commented [vc15]: Adding php repository

Commented [vc16]: Owncership to www-data

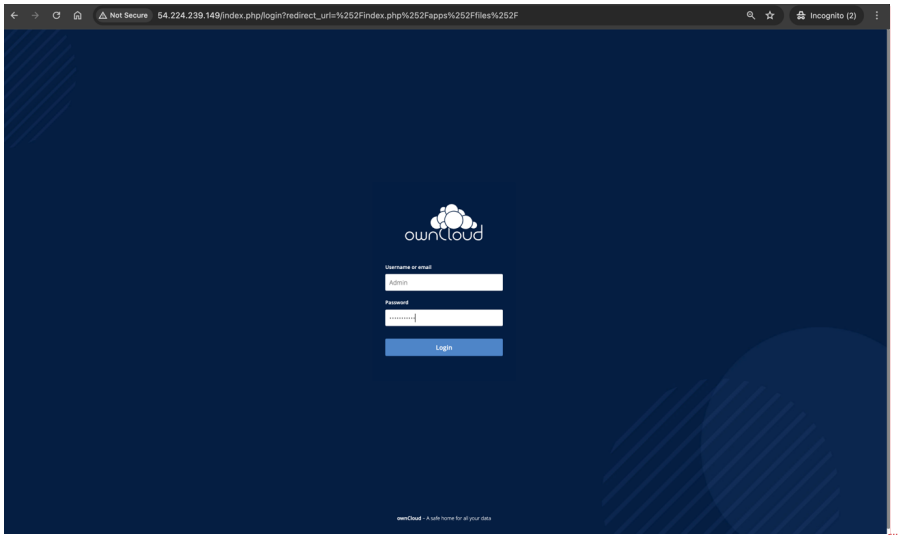
Commented [vc17]: Change the document root location.

- **Open the OwncCloud Page by using the public IP :- http://54.224.239.149/**

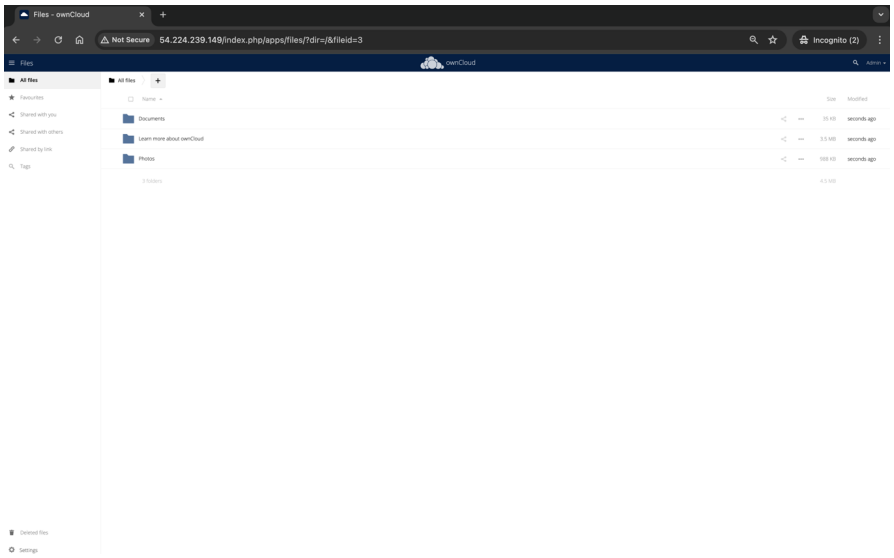




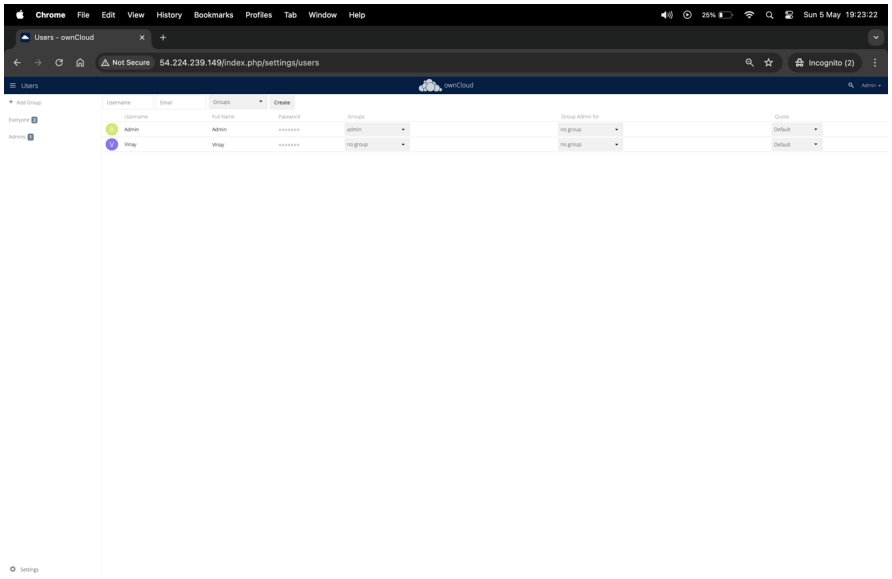
Commented [vc18]: Application setting up with Database and Admin account.



Commented [vc19]: Login to Admin account



Commented [vc20]: User creation.



Creating S3 bucket, IAM user and Mounting S3 bucket to Owncloud server(Using S3fs)

Commented [vc21]: Mounting S3 on OwnCloud Server for creating Backup and Restoration on mounted S3 bucket.
Also can be used for manually moving data to S3 bucket.

Note: As the creation of IAM users and policies is blocked on the Lab Account, the S3 bucket and IAM user/policies are created via a personal account belonging to FragShree.

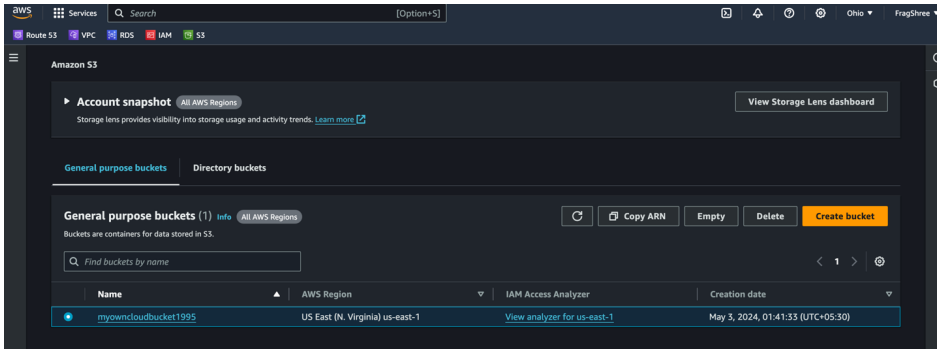
❖ **S3 Bucket and IAM User details:**

Commented [vc22]: Created on Personal ID.

- **AWS_ACCESS_KEY_ID:** AKIA4HMWHFHLABTDWMUX
- **AWS_SECRET_ACCESS_KEY**: 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
- **Region:** us-east-1
- **Bucket ARN**: s3://myowncloudbucket1995/

❖ **S3 bucket creation:**

- Step 1: Create an S3 Bucket:
  - **Log in to AWS**: Go to the AWS Management Console and log in with your credentials.
  - **Navigate to S3**: Select "Services" from the top bar, then choose "S3."
  - **Create a Bucket**:
  - Click the "Create bucket" button.
  - Give your bucket a unique name (it must be globally unique across AWS).
  - Select an appropriate region.
  - Configure other settings like versioning, encryption, and public access as needed.
  - Click "Create bucket."



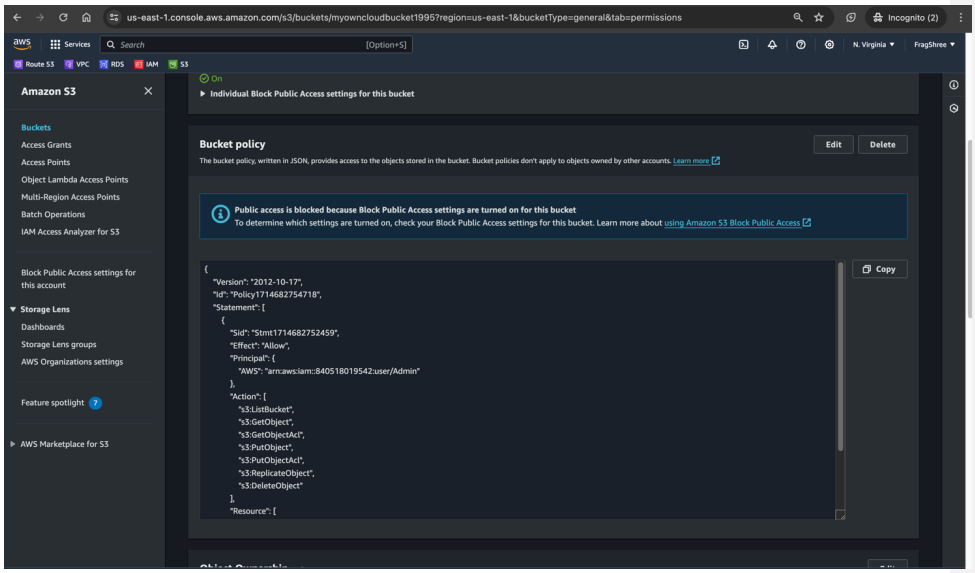
- Step 2: Set Bucket Permissions:
  - **Access Control**: Go to the "Permissions" tab in your new bucket. You can add specific policies or set bucket permissions to control who can access the bucket.

Bucket policy:

```
{
    "Version": "2012-10-17",
    "Id": "Policy1714682754718",
    "Statement": [
        {
            "Sid": "Stmt1714682752459",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::840518019542:user/Admin"
User(Admin)
            },
            "Action": [
                "s3:ListBucket",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:ReplicateObject",
```

Commented [vc23]: IAM User.

```
            "s3:DeleteObject"
        ],
        "Resource": [
            "arn:aws:s3:::myowncloudbucket1995",
            "arn:aws:s3:::myowncloudbucket1995/*"
        ]
    }
  ]
}
```

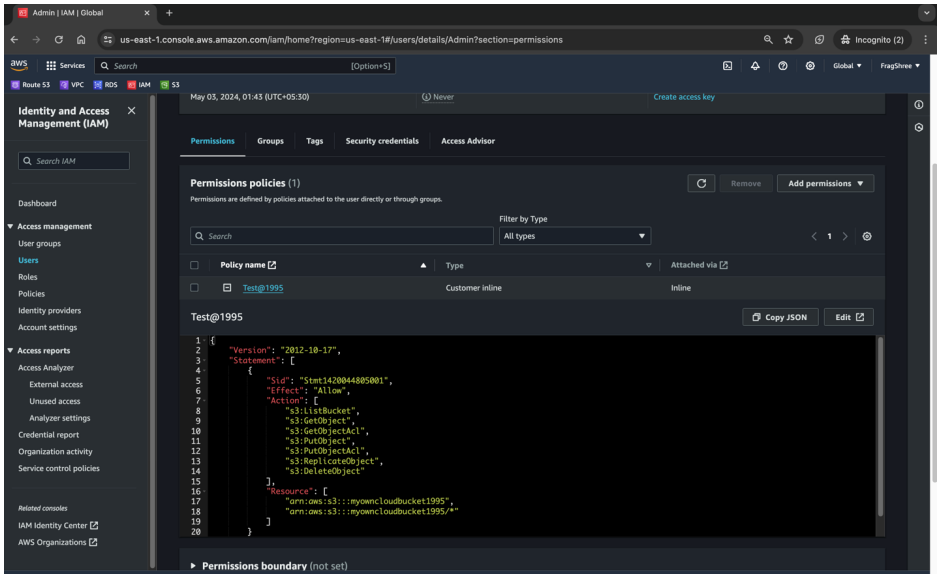


- **IAM user**: If you use an IAM role or user to access S3, ensure it has the required permissions.

IAM User Policy:

Commented [vc24]: Need to add the below Policy on IAM user Policy Tab.

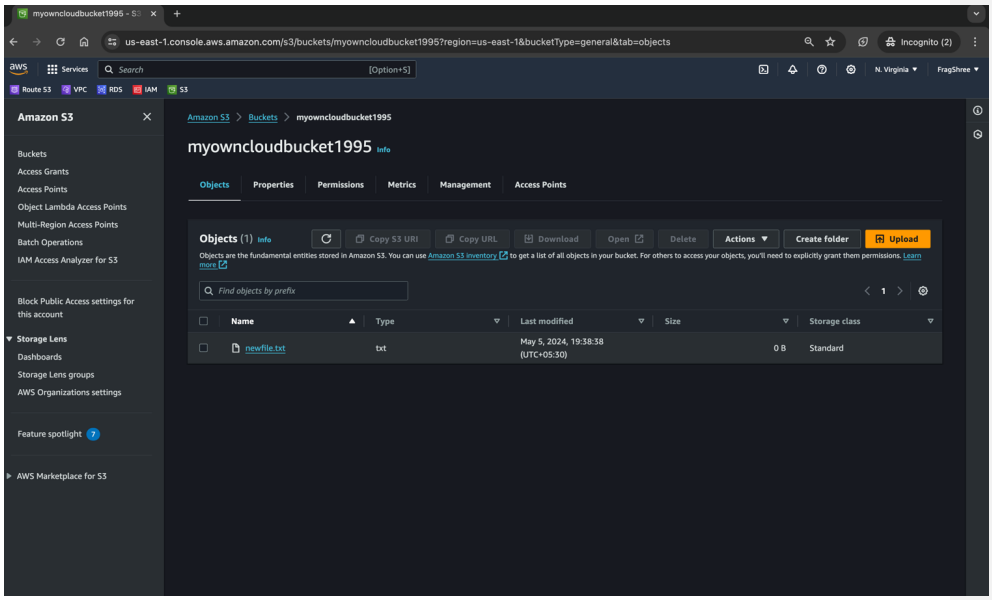
```
{
 "Version": "2012-10-17",
 "Statement": [
    {
       "Sid": "Stmt1420044805001",
       "Effect": "Allow",
       "Action": [
          "s3:ListBucket",
          "s3:GetObject",
          "s3:GetObjectAcl",
          "s3:PutObject",
          "s3:PutObjectAcl",
          "s3:ReplicateObject",
          "s3:DeleteObject"
       ],
       "Resource": [
          "arn:aws:s3:::myowncloudbucket1995",
          "arn:aws:s3:::myowncloudbucket1995/*"
       ]
    }
 ]
}
```



- Step 3: Launch an EC2 Instance
  - Mount S3 as a File System with s3fs.
  - If you want to mount your S3 bucket as a file system on your EC2 instance, you can use `s3fs`. This allows you to interact with S3 as if it were a mounted file system.
  - **Install s3fs**:
    - sudo apt-get update
    - sudo apt-get install s3fs
  - **Create AWS Credentials File**:
  - Create a file called “.passwd-s3fs” in your home directory, containing your Access Key and Secret Key:
  - echo "AKIA4HMWHFHLABTDWMUX:LR95eJ9Qn7mUSf73guYpJkfIFjhbCsSvM8z4bkt2" > ~/.passwd-s3fs
  - chmod 600 ~/.passwd-s3fs
  - cd /mnt
  - mkdir owncloud

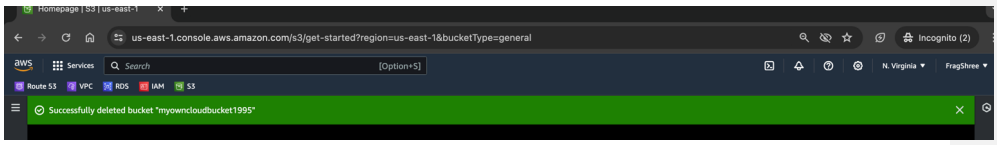
- s3fs myowncloudbucket1995 /mnt/owncloud/ -o passwd_file=~/.passwd-s3fs
- root@ip-10-0-1-173:/mnt# df -Th
  Filesystem Type Size Used Avail Use% Mounted on
  /dev/root ext4 7.6G 2.5G 5.1G 33% /
  tmpfs tmpfs 475M 0 475M 0% /dev/shm
  tmpfs tmpfs 190M 864K 190M 1% /run
  tmpfs tmpfs 5.0M 0 5.0M 0% /run/lock
  /dev/xvda15 vfat 105M 6.1M 99M 6% /boot/efi
  tmpfs tmpfs 95M 4.0K 95M 1% /run/user/1000
  s3fs fuse.s3fs 16E 0 16E 0% /mnt/owncloud
- root@ip-10-0-1-173:/mnt# cd owncloud/
- root@ip-10-0-1-173:/mnt/owncloud# touch newfile.txt
- root@ip-10-0-1-173:/mnt/owncloud# ls -l
  total 1
  -rw-r--r-- 1 root root 0 May 5 14:08 newfile.txt

Commented [vc25]: Mounted S3 Bucket

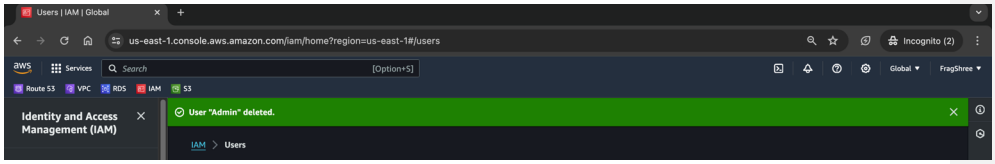


Termination of all Services

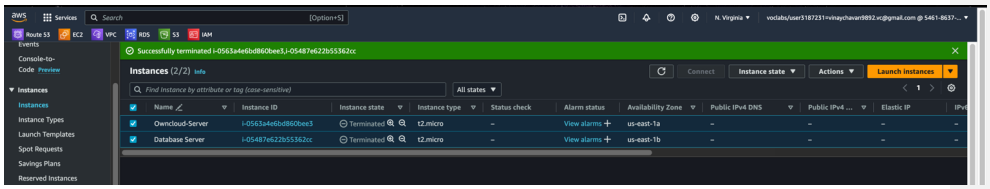
- S3 Bucket



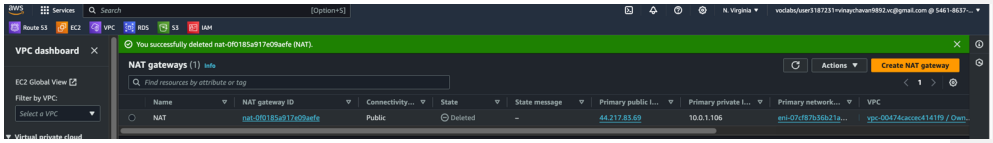
- IAM User.



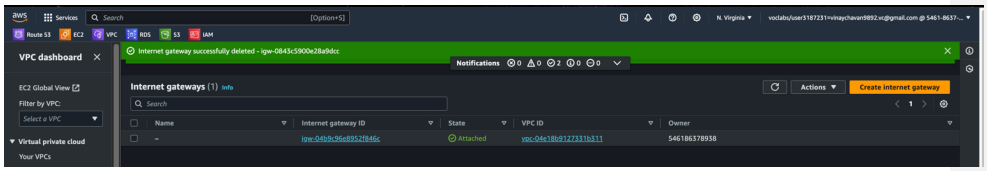
- EC2 instances.



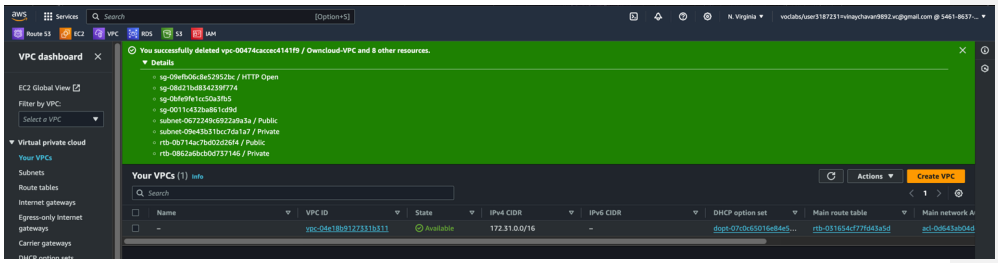
- NAT



- IGW



❖ VPC and 8 other resources.



THE END