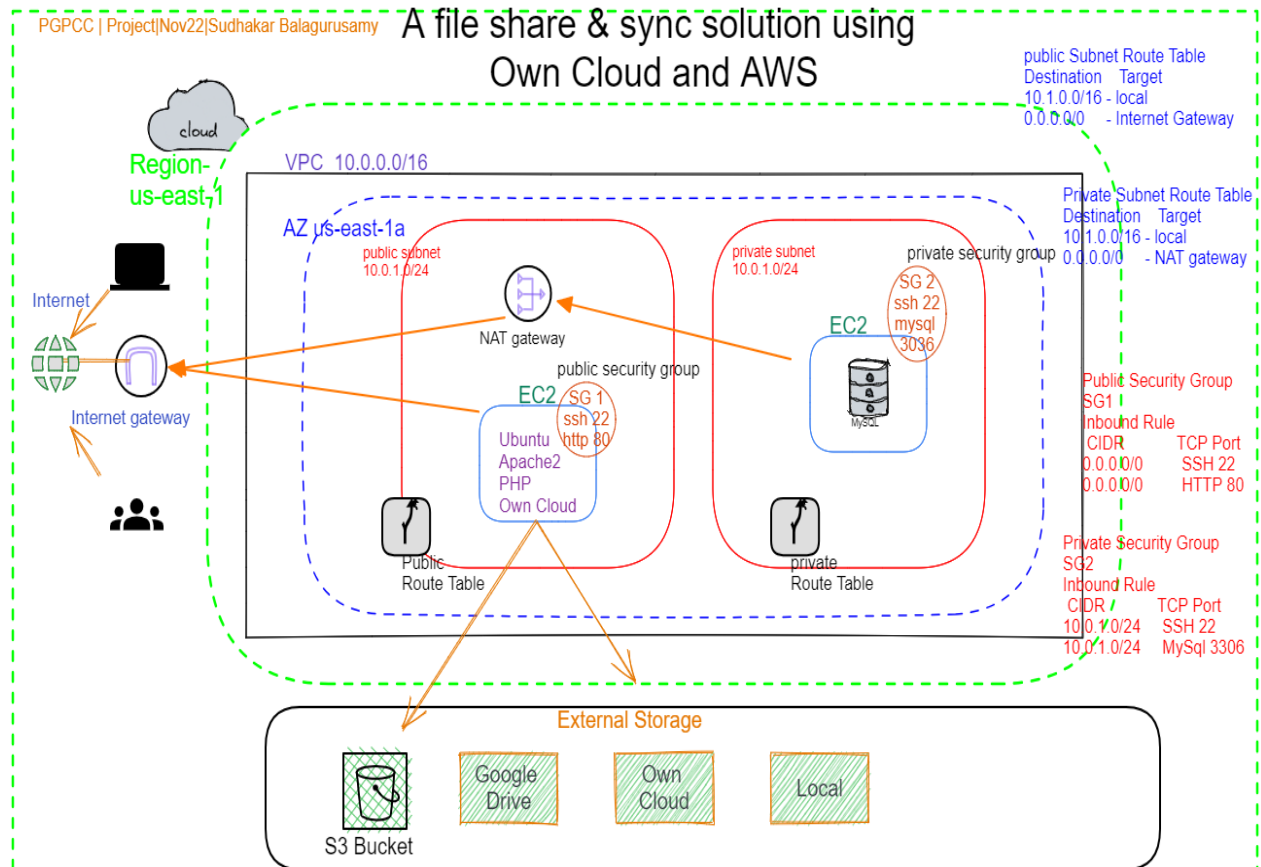


Architecture Diagram



Network Planning:

Create a VPC (10.0.0.0/16) with 2 subnets (10.0.1/24 and 10.0.2.0/24)

Public facing webserver will be in public the subnet (10.0.1/24) and database server will be in the private subnet (10.0.2.0/24)

Routing tables need to be created to direct the traffic to internet and NAT gateway and associated with subnets.

NAT gateway is created to provide internet access to private subnet.

Internet gateway is created for providing internet access to public subnet/VPC

Public Security group is created for SSH /HTTP access to EC2 instances in public subnet.

Private Security group is created for SSH /MYSQL access to EC2 instances in private subnet from public subnet.

The following scripts have been implemented for this project.

Location	Scripts	Description
CLI enabled bash terminal	Create_aws_resources_cli.bash	CLI commands to create the resources needed for this project. The region needs to be set as US-EAST-1 in the global configuration file.
DB Server (private subnet)	Install_mysql.bash	To install MySQL and create owncloud user in MYSQL
APP Server (public subnet)	Install_prerequisite.bash	Bash script to install Apache, PHP, Redis
APP Server (public subnet)	Config_apache2.bash	To configure apache2 for owncloud
APP Server (public subnet)	Install_owncloud.bash	To install own cloud component

Workflow Steps

Step 1

Create a new VPC :gl-vpc with a CIDR block 10.0.0.0/16	
AWS CLI (bash)	<pre>gl_vpc=\$(aws ec2 create-vpc \ --cidr-block 10.0.0.0/16 \ --query Vpc.VpcId \ --output text \</pre>

	--tag-specification "ResourceType=vpc,Tags=[{Key=Name,Value=gl-vpc}]")
AWS Console	<div><div>1. VPC</div><div>2. Your VPCs</div><div>3. Create VPC</div></div>

Screenshot

aws

Services

Search

[Alt+S]

EC2

VPC dashboard

EC2 Global View

Filter by VPC:

Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

Network ACLs

Security groups

Network Analysis

Reachability Analyzer

Network Access Analyzer

Your VPCs (1/2)

Filter VPCs

	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP
<input checked="" type="checkbox"/>	gl-vpc	vpc-07aa93bf04d9a3386	Available	10.0.0.0/16	-	dopt-0...
<input type="checkbox"/>	default-vpc	vpc-0d22a6d8fb758539d	Available	172.31.0.0/16	-	dopt-0...

vpc-07aa93bf04d9a3386 / gl-vpc

Details

CIDRs

Flow logs

Tags

Details

VPC ID	State	DNS hostnames
vpc-07aa93bf04d9a3386	Available	Disabled
Tenancy	DHCP option set	Main route table
Default	dopt-0a55dc01d08b23cdc	rtb-0ccbc5d58e62dd1a
Default VPC	IPv4 CIDR	IPv6 pool
No	10.0.0.0/16	-
Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups	Owner ID
Disabled	Failed to load rule groups	868974893492

Step 2

Create a public subnet with a CIDR block 10.0.1.0/24 in gl-vpc Enable Public IP for the resources on this subnet	
AWS CLI (bash)	<pre>gl_pub_subnet=\$(aws ec2 create-subnet \ --vpc-id \$gl_vpc \ --availability-zone us-east-1a \ --cidr-block 10.0.1.0/24 \ --query Subnet.SubnetId \ --output text \ --tag-specifications "ResourceType=subnet,Tags=[{Key=Name,Value=gl-pub-subnet}])" aws ec2 modify-subnet-attribute --subnet-id \$gl_pub_subnet --map-public-ip-on-launch</pre>
AWS Console	<ol style="list-style-type: none">1. VPC2. Subnets3. Create subnet

Screenshot

The screenshot displays the AWS Management Console interface. On the left, a navigation sidebar shows categories like 'Virtual private cloud', 'Security', and 'Network Analysis'. The main content area is titled 'subnet-0592d68c9222b69c9 / gl-pub-subnet' and contains a 'Details' section with various attributes. Below the details, there are tabs for 'Flow logs', 'Route table', 'Network ACL', 'CIDR reservations', 'Sharing', and 'Tags'. The 'Flow logs' tab is currently selected, showing a search bar and a table header for flow log data.

Virtual private cloud

- Your VPCs
- Subnets**
- Route tables
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- Endpoints
- Endpoint services
- NAT gateways
- Peering connections

Security

- Network ACLs
- Security groups

Network Analysis

- Reachability Analyzer
- Network Access Analyzer

subnet-0592d68c9222b69c9 / gl-pub-subnet

Details

Subnet ID subnet-0592d68c9222b69c9	Subnet ARN arn:aws:ec2:us-east-1:868974893492:subnet/subnet-0592d68c9222b69c9	State Available
Available IPv4 addresses 251	IPv6 CIDR -	Availability Zone us-east-1a
Network border group us-east-1	VPC vpc-07aa93bf04d9a3386 gl-vpc	Route table rtb-0ccbc5d58e62dd1a
Default subnet No	Auto-assign public IPv4 address Yes	Auto-assign IPv6 address No
Customer-owned IPv4 pool -	Outpost ID -	IPv4 CIDR reservations -
IPv6-only No	Hostname type IP name	Resource name DNS A record Disabled
DNS64 Disabled	Owner 868974893492	

Flow logs | Route table | Network ACL | CIDR reservations | Sharing | Tags

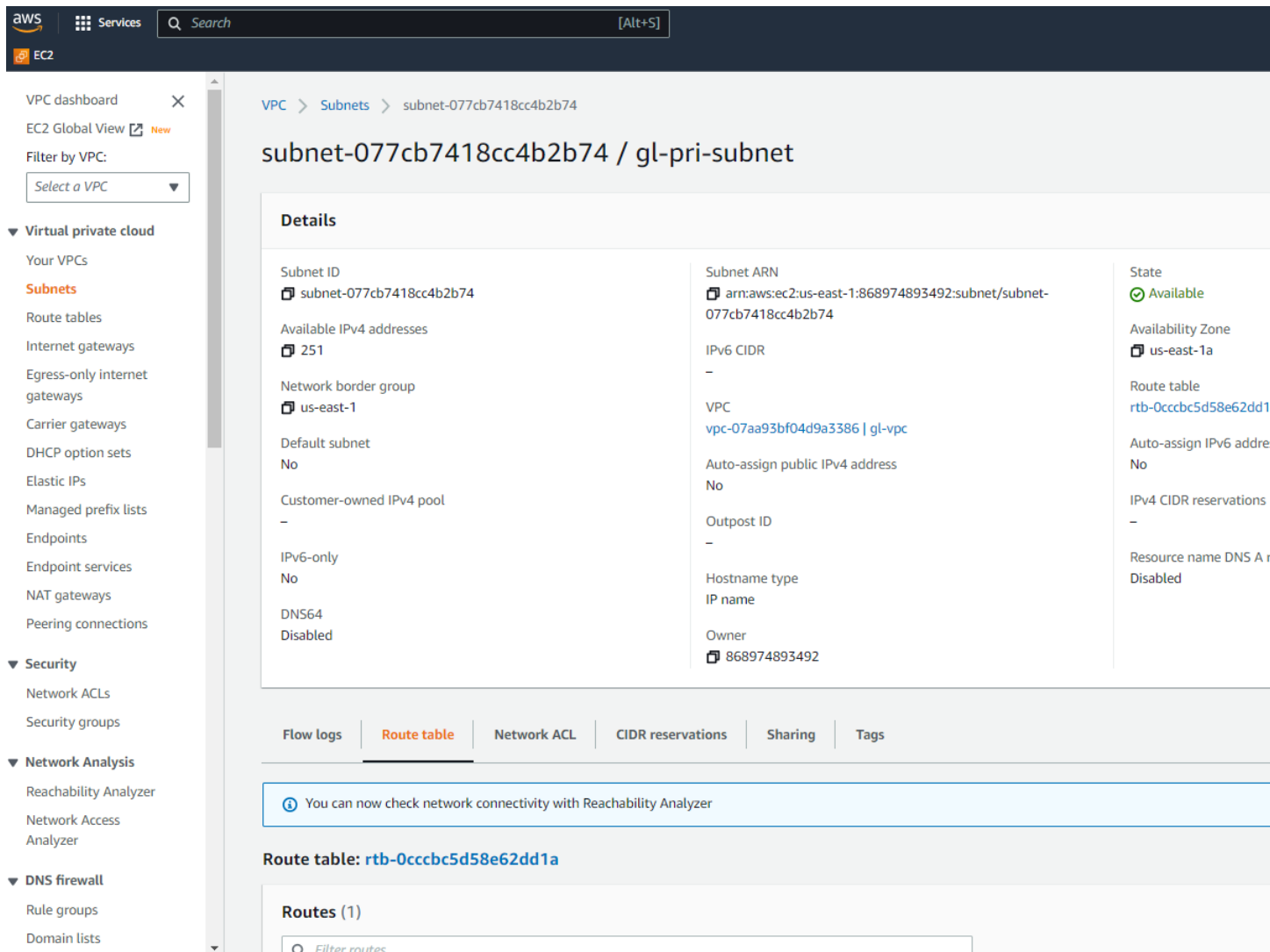
Filter flow logs

Name	Flow log ID	Filter	Destination type
------	-------------	--------	------------------

Step 3

Create a private subnet with a CIDR block 10.0.2.0/24 in gl-vpc	
AWS CLI (bash)	<pre>gl_pri_subnet=\$(aws ec2 create-subnet \ --vpc-id \$gl_vpc \ --availability-zone us-east-1a \ --cidr-block 10.0.2.0/24 \ --query Subnet.SubnetId \ --output text \ --tag-specifications "ResourceType=subnet,Tags=[{Key=Name,Value=gl-pri- subnet}])")</pre>
AWS Console	<ol style="list-style-type: none">1. VPC2. Subnets3. Create subnet

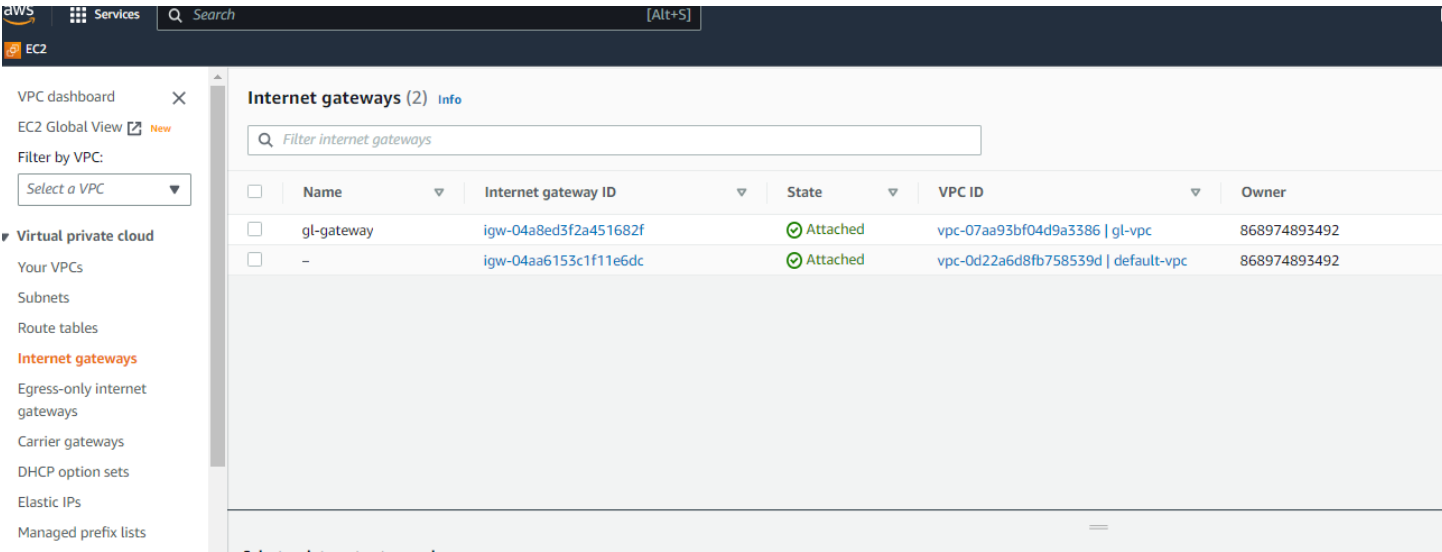
Screenshot



Step 4

Create an internet gateway gl-gateway and attach it to gl-vpc	
AWS CLI (bash)	<pre>gl_gateway=\$(aws ec2 create-internet-gateway --query InternetGateway.InternetGatewayId \ --output text --tag-specifications "ResourceType=internet- gateway,Tags=[{Key=Name,Value=gl-gateway}])" aws ec2 attach-internet-gateway --vpc-id \$gl_vpc --internet-gateway-id \$gl_gateway</pre>
AWS Console	<ol style="list-style-type: none"> 1. VPC 2. Internet gateways 3. Create internet gateway

Screenshot



Step 5

Create a route table for the public subnet to direct the traffic to gl-gateway And associate this to public subnet	
AWS CLI (bash)	<pre>gl_rt_gateway=\$(aws ec2 create-route-table --vpc-id \$gl_vpc --query RouteTable.RouteTableId --output text \ --tag-specifications "ResourceType=route-table,Tags=[{Key=Name,Value=gl-rt-gateway}])") #create an entry to route the 0.0.0.0/0 traffic to the gateway aws ec2 create-route --route-table-id \$gl_rt_gateway --destination-cidr-block 0.0.0.0/0 --gateway-id \$gl_gateway #associate the gl_rt_gateway to public subnet aws ec2 associate-route-table --subnet-id \$gl_pub_subnet --route-table-id \$gl_rt_gateway</pre>
AWS Console	<ol style="list-style-type: none"> VPC Route tables Create route table

Screenshot

Route table with subnet association

aws

Services

Search

[Alt+S]

EC2

VPC dashboard

EC2 Global View

Filter by VPC:

Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

Network ACLs

Security groups

Network Analysis

Reachability Analyzer

Network Access Analyzer

DNS firewall

VPC > Route tables > rtb-03d243ce029adfddf

rtb-03d243ce029adfddf / gl-rt-gateway

You can now check network connectivity with Reachability Analyzer

Details

Info

Route table ID

rtb-03d243ce029adfddf

VPC

vpc-07aa93bf04d9a3386 | gl-vpc

Main

No

Owner ID

868974893492

Explicit subnet associations

subnet-0592d68c9222b69c9 / gl-pu

Routes

Subnet associations

Edge associations

Route propagation

Tags

Explicit subnet associations (1)

Find subnet association

Subnet ID

subnet-0592d68c9222b69c9 / gl-pub-subnet

IPv4 CIDR

10.0.1.0/24

Subnets without explicit associations (1)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Find subnet association

Subnet ID

subnet-077cb7418cc4b2b74 / gl-pri-subnet

IPv4 CIDR

10.0.2.0/24

Routes in the public route table

- EC2

Virtual private cloud

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

Network ACLs

Security groups

Network Analysis

VPC dashboard

EC2 Global View

Filter by VPC:

Select a VPC

VPC > Route tables > rtb-03d243ce029adfdff

rtb-03d243ce029adfdff / gl-rt-gateway

You can now check network connectivity with Reachability Analyzer

Details

Info

Route table ID

rtb-03d243ce029adfdff

VPC

vpc-07aa93bf04d9a3386 | gl-vpc

Main

No

Owner ID

868974893492

Explicit subnet associations

subnet-0592d68c9222b69c9 / gl-

Routes

Subnet associations

Edge associations

Route propagation

Tags

Routes (2)

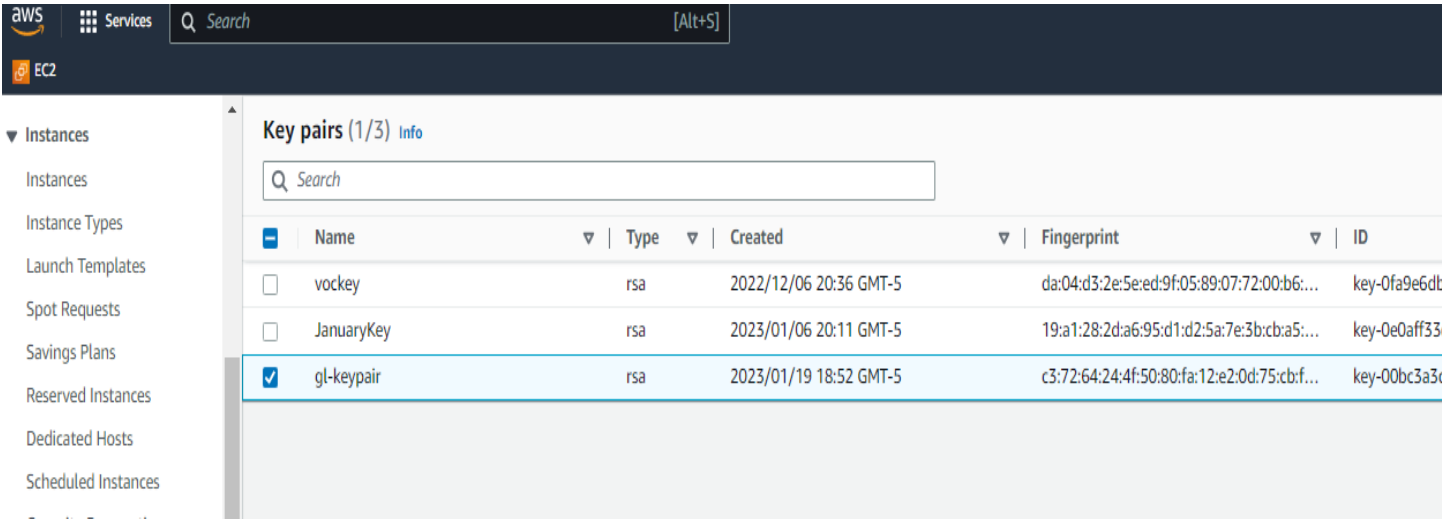
Filter routes

Both

Destination	Target	Status
0.0.0.0/0	igw-04a8ed3f2a451682f	Active
10.0.0.0/16	local	Active

	3. Create key pair
--	--------------------

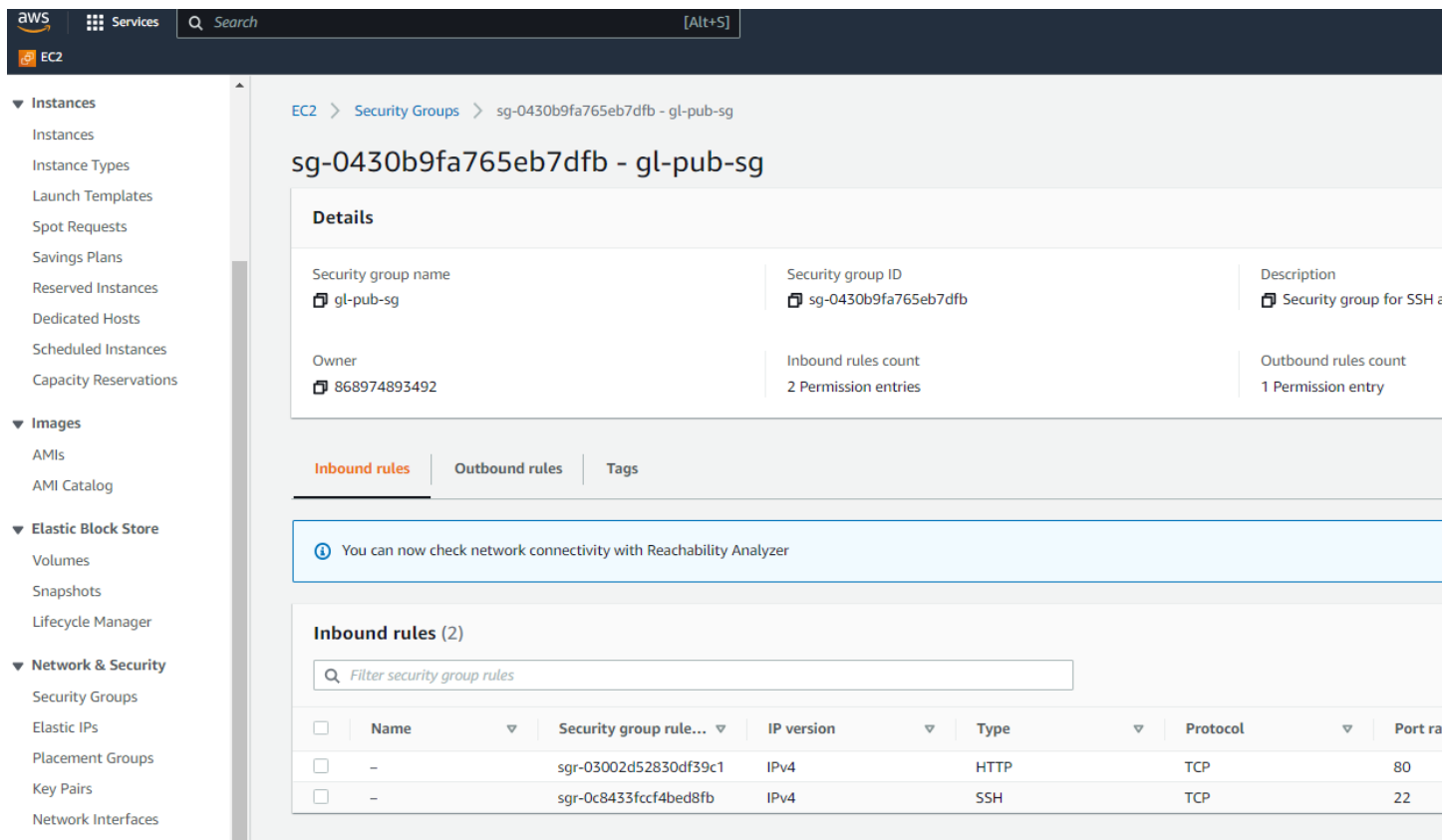
Screenshot



Step 7

Create a security group gl-pub-sg for public subnet EC2 instances. This allows SSH & HTTP ports to be accessed from internet	
AWS CLI (bash)	<pre>#create a security group for public subnet resources gl_pub_sg=\$(aws ec2 create-security-group --group-name gl-pub-sg -- description "Security group for SSH access" \ --vpc-id \$gl_vpc \ --query GroupId --output text) #authorize the security group for ssh access aws ec2 authorize-security-group-ingress --group-id \$gl_pub_sg --protocol tcp --port 22 --cidr 0.0.0.0/0 aws ec2 authorize-security-group-ingress --group-id \$gl_pub_sg --protocol tcp --port 80 --cidr 0.0.0.0/0</pre>
AWS Console	<ol style="list-style-type: none">1. EC22. Security Groups3. Create security group

Screenshot



Step 8

Create a security group gl-pri-sg for private subnet EC2 instances. This allows SSH & MySQL ports to be accessed from the public subnet

AWS CLI (bash)

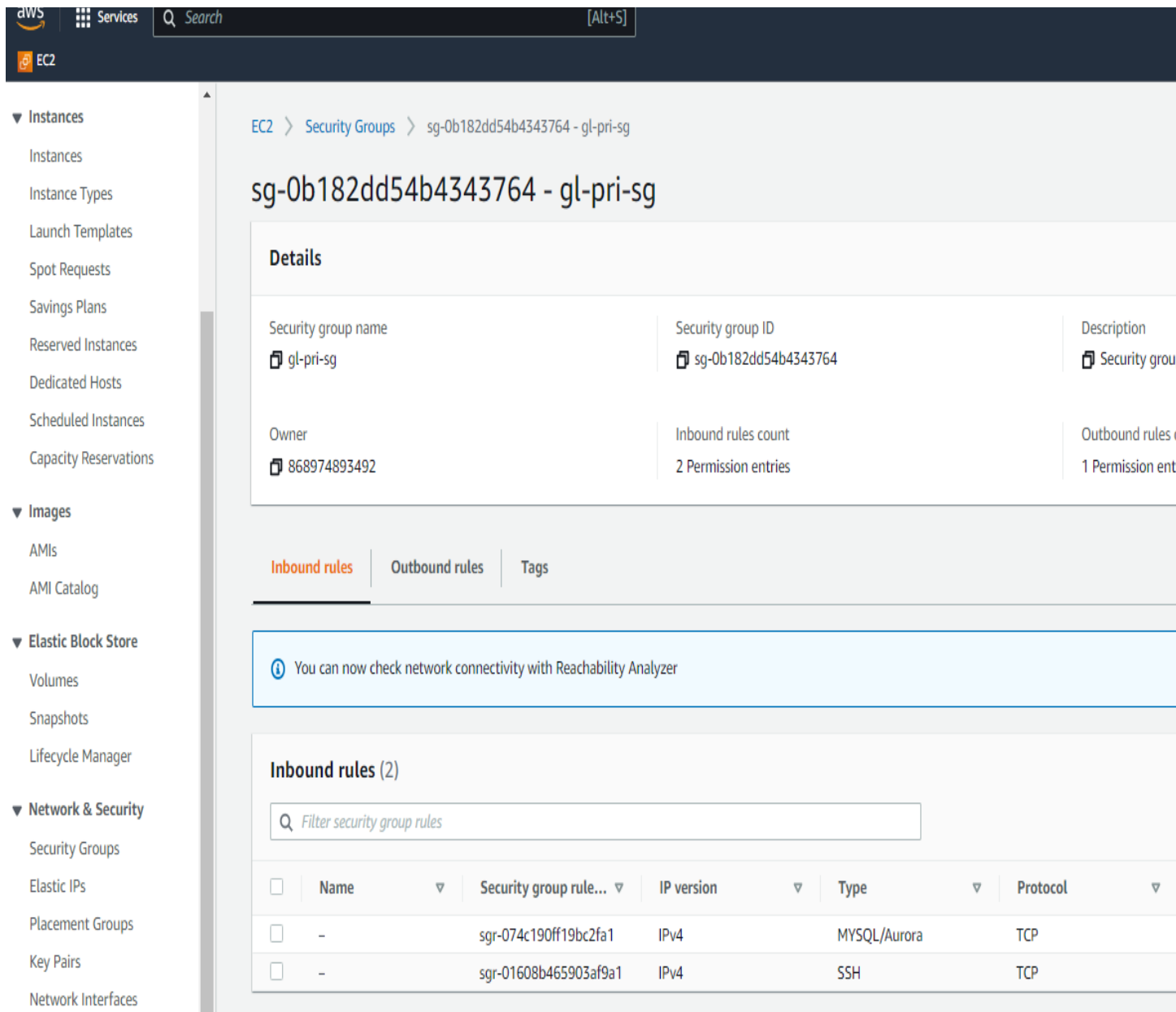
```
gl_pri_sg=$(aws ec2 create-security-group --group-name \
gl-pri-sg --description "Security group for SSH & MYSQL access from public
subnet" \
--vpc-id $gl_vpc \
--query GroupId --output text)

aws ec2 authorize-security-group-ingress --group-id $gl_pri_sg --protocol tcp
--port 22 --cidr 10.0.1.0/24
aws ec2 authorize-security-group-ingress --group-id $gl_pri_sg --protocol tcp
--port 3306 --cidr 10.0.1.0/24
```

AWS Console

1. [EC2](#)
2. [Security Groups](#)
3. [Create security group](#)

Screenshot



Step 9

Create an EC2 Ubuntu 18_04 instance (gl-app-server) to host own cloud application and webserver in the public subnet.

Associate the gl-pub-sg security group to it.

ami-08fdec01f5df9998f is used for 18_04 Ubuntu

AWS CLI (bash)	<pre>ubuntu_18_04="ami-08fdec01f5df9998f" gl_app_server=\$(aws ec2 run-instances --image-id \$ubuntu_18_04 --count 1 -- instance-type t2.micro --key-name \$gl_keypair \ --security-group-ids \$gl_pub_sg \ --subnet-id \$gl_pub_subnet \ --associate-public-ip-address \ --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=gl-app- server}]' \ --query Instances[0].InstanceId \ --output text) #get the public IP address gl_app_server_ip=\$(aws ec2 describe-instances --instance-id \$gl_app_server \ --query Reservations[0].Instances[0].PublicIpAddress \ --output text)</pre>
AWS Console	<ol style="list-style-type: none"> 1. EC2 2. Instances 3. Launch an instance

Screenshot

aws

Services

Search

[Alt+S]

EC2

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Load Balancing

Load Balancers

Target Groups

Auto Scaling

Launch Configurations

EC2 > Instances > i-0fc6ec71595aa51c4

Instance summary for i-0fc6ec71595aa51c4 (gl-app-server)

Updated less than a minute ago

Instance ID

i-0fc6ec71595aa51c4 (gl-app-server)

IPv6 address

-

Hostname type

IP name: ip-10-0-1-5.ec2.internal

Answer private resource DNS name

-

Auto-assigned IP address

54.167.63.86 [Public IP]

IAM Role

-

Public IPv4 address

54.167.63.86 | [open address](#)

Instance state

Running

Private IP DNS name (IPv4 only)

ip-10-0-1-5.ec2.internal

Instance type

t2.micro

VPC ID

vpc-07aa93bf04d9a3386 (gl-vpc)

Subnet ID

subnet-0592d68c9222b69c9 (gl-pub-subnet)

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

▼ Instance details

Platform

Ubuntu (Inferred)

Platform details

Linux/UNIX

Stop protection

Disabled

Instance auto-recovery

Default

AMI Launch index

0

Credit specification

standard

AMI ID

ami-08fdec01f5df9998f

AMI name

ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20221207

Launch time

Sat Jan 21 2023 22:05:11 GMT-0500 (Eastern Standard Time) (1 minute)

Lifecycle

normal

Key pair name

gl-keypair

Kernel ID

-

Step 10

Create an EC2 Ubuntu 18_04 instance (gl-db-server) to host My SQL server in the private subnet and associate the gl-pri-sg security group to it.

ami-08fdec01f5df9998f is used for 18_04 Ubuntu

AWS CLI (bash)	<pre>ubuntu_18_04="ami-08fdec01f5df9998f" gl_db_server=\$(aws ec2 run-instances --image-id \$ubuntu_18_04 --count 1 -- instance-type t2.micro --key-name \$gl_keypair \ --security-group-ids \$gl_pri_sg \ --subnet-id \$gl_pri_subnet \ --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=gl-db- server}]' \ --query Instances[0].InstanceId \ --output text)</pre>
AWS Console	<ol style="list-style-type: none">1. EC22. Instances3. Launch an instance

Screenshot

AWS Console

1. [EC2](#)
2. [Instances](#)
3. Launch an instance

Screenshot

The screenshot displays the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', a search bar, and a keyboard shortcut '[Alt+S]'. The left-hand navigation pane is expanded to 'Virtual private cloud', with 'NAT gateways' highlighted in orange. Below this, other options like 'Your VPCs', 'Subnets', and 'Route tables' are visible. The main content area is titled 'NAT gateways (1/1) Info'. It features a search bar labeled 'Filter NAT gateways' and a table with columns: Name, NAT gateway ID, Connectivity type, State, State message, and Elastic IP address. A single entry is shown with a blue status icon, a hyphen for Name, ID 'nat-01e584fcb48fa8019', 'Public' connectivity, 'Available' state (with a green checkmark), a hyphen for State message, and Elastic IP address '3.224.58.10'. Below the table, the selected NAT gateway 'nat-01e584fcb48fa8019' is detailed in the 'Details' tab. The details are organized into three columns: NAT gateway ID (nat-01e584fcb48fa8019), NAT gateway ARN (arn:aws:ec2:us-east-1:868974893492:natgateway/nat-01e584fcb48fa8019), and VPC (vpc-07aa93bf04d9a3386 / gl-vpc); Connectivity type (Public) and Subnet (subnet-0592d68c9222b69c9 / gl-pub-subnet); and State (Available with a green checkmark), Primary private IPv4 address (10.0.1.82), and Created date (Saturday, January 21, 2023 at 10:00 AM).

Name	NAT gateway ID	Connectivity type	State	State message	Elastic IP address
-	nat-01e584fcb48fa8019	Public	Available	-	3.224.58.10

Details

NAT gateway ID nat-01e584fcb48fa8019	Connectivity type Public	State Available
NAT gateway ARN arn:aws:ec2:us-east-1:868974893492:natgateway/nat-01e584fcb48fa8019	Elastic IP address 3.224.58.10	Primary private IPv4 address 10.0.1.82
VPC vpc-07aa93bf04d9a3386 / gl-vpc	Subnet subnet-0592d68c9222b69c9 / gl-pub-subnet	Created Saturday, January 21, 2023 at 10:00 AM

Step 12

**Create a route table gl-rt-pri for private subnet.
Add a route to direct the internet traffic to NAT gateway**

Associate this to private subnet	
AWS CLI (bash)	<pre>#create route table for private subnet gl_rt_pri=\$(aws ec2 create-route-table --vpc-id \$gl_vpc --query RouteTable.RouteTableId --output text \ --tag-specifications "ResourceType=route-table,Tags=[{Key=Name,Value=gl- rt-pri}])") #create an entry to route the 0.0.0.0/0 traffic to the nat gateway aws ec2 create-route --route-table-id \$gl_rt_pri --destination-cidr-block 0.0.0.0/0 --gateway-id \$gl_nat_gw #assoicate aws ec2 associate-route-table --subnet-id \$gl_pri_subnet --route-table-id \$gl_rt_pri</pre>
AWS Console	<ol style="list-style-type: none"> 1. VPC 2. Route tables 3. Create route table

Screenshot

Route table with routes to direct traffic to NAT gateway

aws

Services

Search

[Alt+S]

EC2

VPC dashboard

EC2 Global View

Filter by VPC:

Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

Network ACLs

Security groups

Network Analysis

Reachability Analyzer

Network Access Analyzer

VPC > Route tables > rtb-093112adaa1c214fb

rtb-093112adaa1c214fb / gl-rt-pri

You can now check network connectivity with Reachability Analyzer

Details

Info

Route table ID

rtb-093112adaa1c214fb

VPC

vpc-07aa93bf04d9a3386 | gl-vpc

Main

No

Owner ID

868974893492

Explicit subnet asso

-

Routes

Subnet associations

Edge associations

Route propagation

Tags

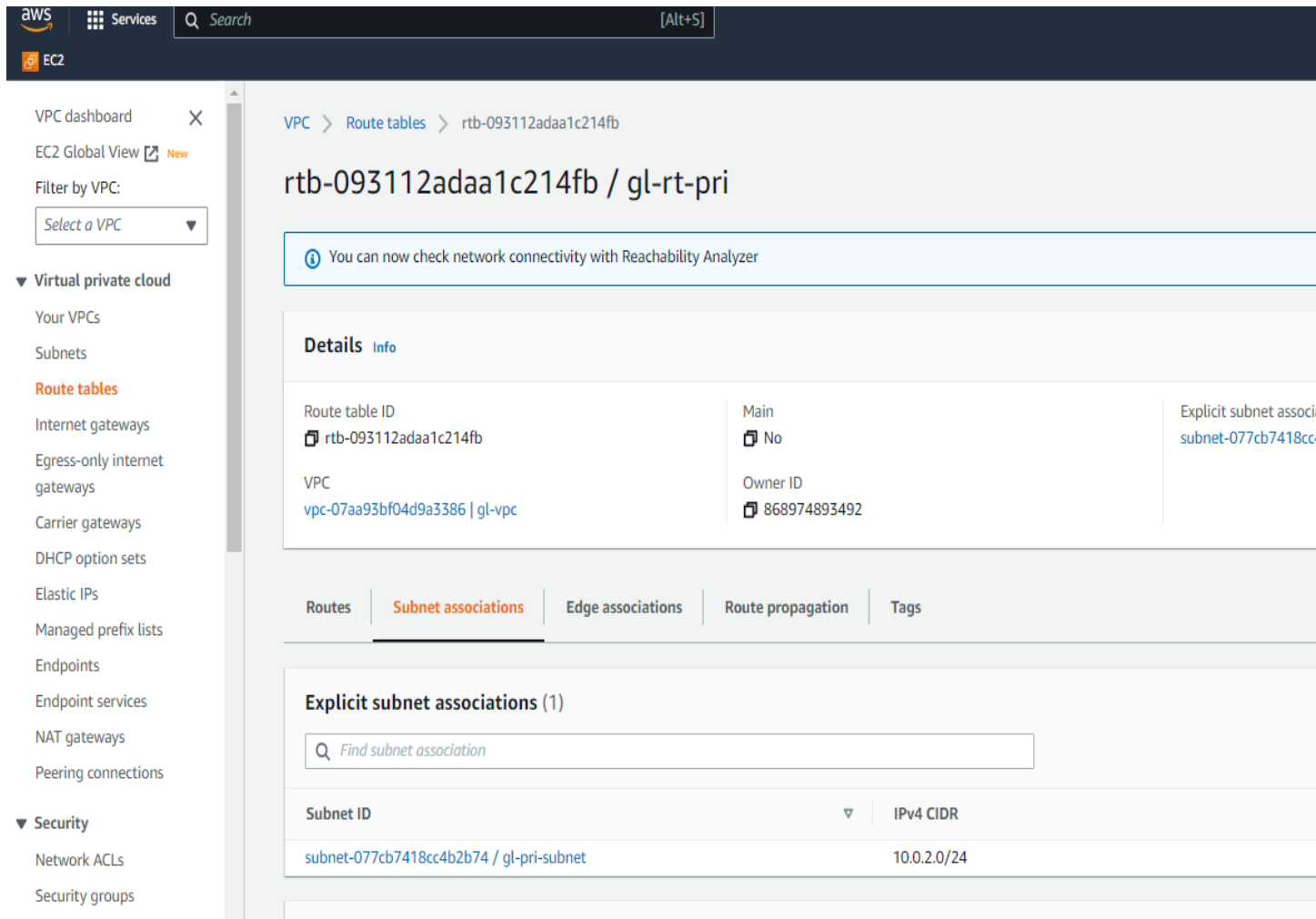
Routes (2)

Filter routes

Both

Destination	Target
0.0.0.0/0	nat-01e584fcb48fa8019
10.0.0.0/16	local

Route table with private subnet associated.



Step 13

Copy the keypair gl-keypair to app server EC2 instance in the public subnet.

SSH to app server from Local terminal.

SSH to db server from app server

Commands	<pre>#copy the keypair file to app server scp -i \${gl_keypair}.pem \${gl_keypair}.pem ubuntu@\${gl_app_server_ip}:.ssh ssh -i gl-keypair.pem ubuntu@\${gl_app_server_ip} ssh -i gl-keypair.pem ubuntu@10.0.2.252</pre>
-----------------	--

Screenshot for scp key , ssh to app server and DB server.

```
ubuntu@ip-10-0-2-252: ~  
sudhakar@UrsaMajor:~$ scp -i gl-keypair.pem gl-keypair.pem ubuntu@$gl_app_server_ip:  
sudhakar@UrsaMajor:~$ scp -i gl-keypair.pem gl-keypair.pem ubuntu@$gl_app_server_ip:  
gl-keypair.pem 100% 1675 62.4KB  
sudhakar@UrsaMajor:~$ ssh -i gl-keypair.pem ubuntu@$gl_app_server_ip  
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1092-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Sun Jan 22 04:51:07 UTC 2023  
  
System load:  0.0          Processes:      95  
Usage of /:   16.8% of 7.57GB Users logged in: 0  
Memory usage: 19%         IP address for eth0: 10.0.1.5  
Swap usage:   0%  
  
0 updates can be applied immediately.  
  
New release '20.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Sun Jan 22 03:22:59 2023 from 73.159.171.229  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-10-0-1-5:~$ ssh -i gl-keypair.pem ubuntu@10.0.2.252  
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1092-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Sun Jan 22 04:51:30 UTC 2023  
  
System load:  0.0          Processes:      96  
Usage of /:   25.6% of 7.57GB Users logged in: 0  
Memory usage: 37%         IP address for eth0: 10.0.2.252  
Swap usage:   0%  
  
0 updates can be applied immediately.  
  
New release '20.04.5 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
*** System restart required ***  
Last login: Sun Jan 22 03:26:08 2023 from 10.0.1.5  
ubuntu@ip-10-0-2-252:~$
```

Step 14

Install My SQL Server on the gl-db-server based on the instructions from the following link.

https://doc.owncloud.com/server/next/admin_manual/installation/quick_guides/ubuntu_22_04.html

Commands

```
#!/usr/bin/env bash

apt update && apt upgrade -y
apt install -y mysql-server

sed -i "[mysqld]/transaction-isolation = READ-COMMITTED\nperformance_schema = on"
/etc/mysql/mysql.conf.d/mysqld.cnf
sed -i "s/bind-address/#bind-address/"
/etc/mysql/mysql.conf.d/mysqld.cnf
sed -i "[mysqld]/bind-address = 0.0.0.0"
/etc/mysql/mysql.conf.d/mysqld.cnf

systemctl start mysql.service
mysql -u root -e "CREATE DATABASE IF NOT EXISTS owncloud; \
CREATE USER 'owncloud'@'localhost' IDENTIFIED BY 'pi=3.14159'; \
GRANT ALL PRIVILEGES ON *.* TO 'owncloud'@'localhost' WITH GRANT OPTION; \
FLUSH PRIVILEGES; \
CREATE USER 'owncloud'@'%' IDENTIFIED BY 'pi=3.14159'; \
GRANT ALL PRIVILEGES ON *.* TO 'owncloud'@'%' WITH GRANT OPTION; \
FLUSH PRIVILEGES; "
```

Screenshot for MySQL server installation

```
root@ip-10-0-2-252: ~  
root@ip-10-0-2-252:~# history  
 1 apt update && apt upgrade -y  
 2 apt install -y mysql-server  
 3 sed -i "\[mysqld\]/atransaction-isolation = READ-COMMITTED\nperformance_schema =  
 4 sed -i "s/bind-address/#bind-address/" /etc/mysql/mysql.conf.d/mysqld.cnf  
 5 sed -i "\[mysqld\]/abind-address = 0.0.0.0" /etc/mysql/mysql.conf.d/mysqld.cnf  
 6 vi /etc/mysql/mysql.conf.d/mysqld.cnf  
 7 grep bind /etc/mysql/mysql.conf.d/mysqld.cnf  
 8 systemctl start mysql.service  
 9 mysql -u root  
10 systemctl restart mysql.service  
11 exit  
12 history  
root@ip-10-0-2-252:~#
```

Step 15

Install Apache2, PHP 7.4 , Redis , Own Cloud on gl-appserver EC2 instance

based on the instructions from the following link.

https://doc.owncloud.com/server/next/admin_manual/installation/quick_guides/ubuntu_22_04.html

Commands to install

Prerequisite components

```
#!/usr/bin/env bash
```

```
apt update && apt upgrade -y  
add-apt-repository ppa:ondrej/php -y  
apt update && apt upgrade -y  
apt install -y \  
  apache2 \  
  libapache2-mod-php7.4 \  
  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  
apt-get install -y php7.4-smbclient
```


	<pre> echo "extension=smbclient.so" > /etc/php/7.4/mods- available/smbclient.ini phpenmod smbclient systemctl restart apache2 apt install -y \ unzip bzip2 rsync curl jq \ inetutils-ping ldap-utils\ smbclient #install mysql-client apt-get install -y mysql-client </pre>
Commands to configure Apache	<pre> #!/usr/bin/env bash FILE="/etc/apache2/sites-available/owncloud.conf" cat <<EOM >\$FILE <VirtualHost *:80> # uncomment the line below if variable was set #ServerName \$my_domain DirectoryIndex index.php index.html DocumentRoot /var/www/owncloud <Directory /var/www/owncloud> Options +FollowSymlinks -Indexes AllowOverride All Require all granted <IfModule mod_dav.c> Dav off </IfModule> SetEnv HOME /var/www/owncloud SetEnv HTTP_HOME /var/www/owncloud </Directory> </VirtualHost> EOM a2dissite 000-default a2ensite owncloud.conf echo "Enabled \$FILE in Apache" a2enmod dir env headers mime rewrite setenvif systemctl restart apache2 echo "apache2 has been restarted" </pre>

Commands to install Own Cloud

```
#!/usr/bin/env bash
#run this script as follows
#dbserver_ip=xxxxx install-owncloud.bash

if [ $dbserver_ip = "" ] ;
then
echo "exiting now. dbserver_ip is not defined";
exit;
fi

sec_admin_pwd="pi=3.14159"
echo $sec_admin_pwd > /etc/.sec_admin_pwd.txt

sec_db_pwd="pi=3.14159"
echo $sec_db_pwd > /etc/.sec_db_pwd.txt

FILE="/usr/local/bin/occ"

cat <<EOM >$FILE
#!/bin/bash
cd /var/www/owncloud
sudo -E -u www-data /usr/bin/php /var/www/owncloud/occ "$@"
EOM

chmod +x $FILE

echo "Created the helper file: $FILE"

cd /var/www/
wget https://download.owncloud.com/server/stable/owncloud-complete-latest.tar.bz2 && \
tar -xjf owncloud-complete-latest.tar.bz2 && \
chown -R www-data. owncloud

echo "Downloaded the OwnCloud repo"

occ maintenance:install \
--database "mysql" \
--database-host ${dbserver_ip} \
--database-name "owncloud" \
--database-user "owncloud" \
--database-pass ${sec_db_pwd} \
--data-dir "/var/www/owncloud/data" \
--admin-user "admin" \
--admin-pass ${sec_admin_pwd}

app_server_ip=$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4)
```

```

local_ip=$(curl -s http://169.254.169.254/latest/meta-data/local-ipv4)
client_ip=$(echo $SSH_CLIENT | awk '{ print $1}')

occ config:system:set trusted_domains 1 --value="$client_ip"
occ config:system:set trusted_domains 2 --value="$app_server_ip"
occ config:system:set trusted_domains 3 --value="$local_ip"

occ config:system:set files_external_allow_create_new_local --value 'true'

occ background:cron
echo "*/15 * * * * /var/www/owncloud/occ system:cron" \
| sudo -u www-data -g crontab tee -a \
/var/spool/cron/crontabs/www-data

echo "0 2 * * * /var/www/owncloud/occ dav:cleanup-chunks" \
| sudo -u www-data -g crontab tee -a \
/var/spool/cron/crontabs/www-data

echo "Configuring Memcache\APCu"

occ config:system:set \
    memcache.local \
    --value '\OC\Memcache\APCu'

    echo "Configuring Memcache\Redis"
occ config:system:set \
    memcache.locking \
    --value '\OC\Memcache\Redis'
occ config:system:set \
    redis \
    --value '{"host": "127.0.0.1", "port": "6379"}' \
    --type json

FILE="/etc/logrotate.d/owncloud"
cat <<EOM >$FILE
/var/www/owncloud/data/owncloud.log {
    size 10M
    rotate 12
    copytruncate
    missingok
    compress
    compresscmd /bin/gzip
}
EOM

cd /var/www/
chown -R www-data. owncloud

```

	<pre>systemctl restart apache2 echo "apache2 has been restarted"</pre>
--	--

Screenshot for Apache2, PHP 7.4, Redis, Own Cloud Installation

```

65 history
66 history 30
root@ip-10-0-1-5:/var/www# history
 1 ll
 2 ll .ssh/
 3 ssh -i .ssh/gl-keypair.pem ubuntu@10.0.2.252
 4 exit
 5 apt update && apt upgrade -y
 6 add-apt-repository ppa:ondrej/php -y
 7 apt update && apt upgrade -y
 8 apt install -y apache2 libapache2-mod-php7.4 openssl redis-server wget php7.4 php7.
php7.4-gd php7.4-imagick php7.4-intl php7.4-json php7.4-mbstring php7.4-gmp php7.4-bcmath php7.4-
4-zip php7.4-apcu php7.4-redis php7.4-ldap php-pharseclib
 9 apt-get install -y php7.4-smbclient
10 echo "extension=smbclient.so" > /etc/php/7.4/mods-available/smbclient.ini
11 phpenmod smbclient
12 systemctl restart apache2
13 php -m |grep smb
14 systemctl restart apache2
15 apt install -y unzip bzip2 rsync curl jq inetutils-ping ldap-utils smbclient
16 apt-get install -y mysql-client
17 FILE="/etc/apache2/sites-available/owncloud.conf"
18 cat <<EOM >$FILE
<VirtualHost *:80>
# uncommment the line below if variable was set
#ServerName $my_domain
DirectoryIndex index.php index.html
DocumentRoot /var/www/owncloud
<Directory /var/www/owncloud>
    Options +FollowSymlinks -Indexes
    AllowOverride All
    Require all granted

<IfModule mod_dav.c>
    Dav off
</IfModule>

    SetEnv HOME /var/www/owncloud
    SetEnv HTTP_HOME /var/www/owncloud
</Directory>
</VirtualHost>
EOM

19 a2disssite 000-default
20 a2ensite owncloud.conf
21 a2enmod dir env headers mime rewrite setenvif
22 systemctl restart apache2
23 mysql
24 mysql -u owncloud -h 10.0.2.252 -p
25 history
26 export dbserver_ip=10.0.2.252
27 env|grep -i dbser
28 sec_admin_pwd="pi=3.14159"
29 echo $sec_admin_pwd > /etc/.sec_admin_pwd.txt
30 sec_db_pwd="pi=3.14159"
31 echo $sec_db_pwd > /etc/.sec_db_pwd.txt
32 FILE="/usr/local/bin/occ"

```

Screenshot2 for Apache2, PHP 7.4, Redis, Own Cloud Installation

```

root@ip-10-0-1-5: /var/www
30 sec_db_pwd="pi=3.14159"
31 echo $sec_db_pwd > /etc/.sec_db_pwd.txt
32 FILE="/usr/local/bin/occ"
33 cat <<EOM >$FILE
#!/bin/bash
cd /var/www/owncloud
sudo -E -u www-data /usr/bin/php /var/www/owncloud/occ "\$@"
EOM

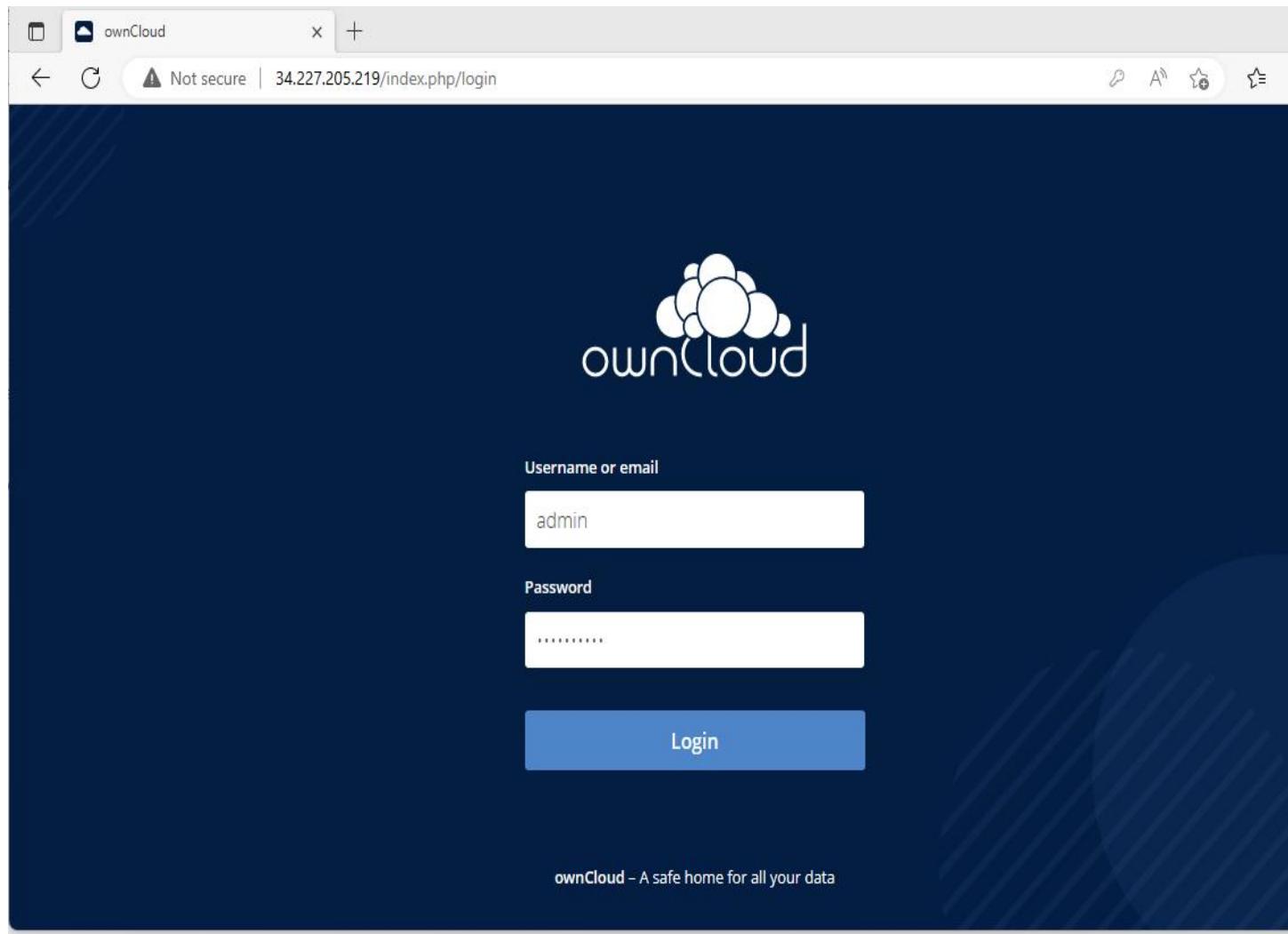
34 chmod +x $FILE
35 cd /var/www/
36 wget https://download.owncloud.com/server/stable/owncloud-complete-latest.tar.bz2 && tar -xj
&& chown -R www-data. owncloud
37 occ -h
38 env|grep -i sec_a
39 history
40 echo $sec_admin_pwd
41 app_server_ip=$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4)
42 local_ip=$(curl -s http://169.254.169.254/latest/meta-data/local-ipv4)
43 client_ip=$(echo $SSH_CLIENT | awk '{ print $1}')
44 occ config:system:set trusted_domains 1 --value=""
45 occ config:system:set trusted_domains 2 --value="$client_ip"
46 echo $client_ip
47 echo $local_ip
48 echo $app_server_ip
49 occ config:system:set trusted_domains 2 --value="$app_server_ip"
50 occ config:system:set trusted_domains 3 --value="$local_ip"
51 occ config:system:set files_external_allow_create_new_local --value 'true'
52 occ background:cron
53 echo "*/15 * * * * /var/www/owncloud/occ system:cron" | sudo -u www-data -g crontab te
-data
54 echo "0 2 * * * /var/www/owncloud/occ dav:cleanup-chunks" | sudo -u www-data -g cronta
/www-data
55 occ config:system:set memcache.local --value '\OC\Memcache\APCu'
56 occ config:system:set memcache.locking --value '\OC\Memcache\Redis'
57 occ config:system:set redis --value '{"host": "127.0.0.1", "port": "6379"}' --type
58 FILE="/etc/logrotate.d/owncloud"
59 cat <<EOM >$FILE
/var/www/owncloud/data/owncloud.log {
size 10M
rotate 12
copytruncate
missingok
compress
compresscmd /bin/gzip
}
EOM

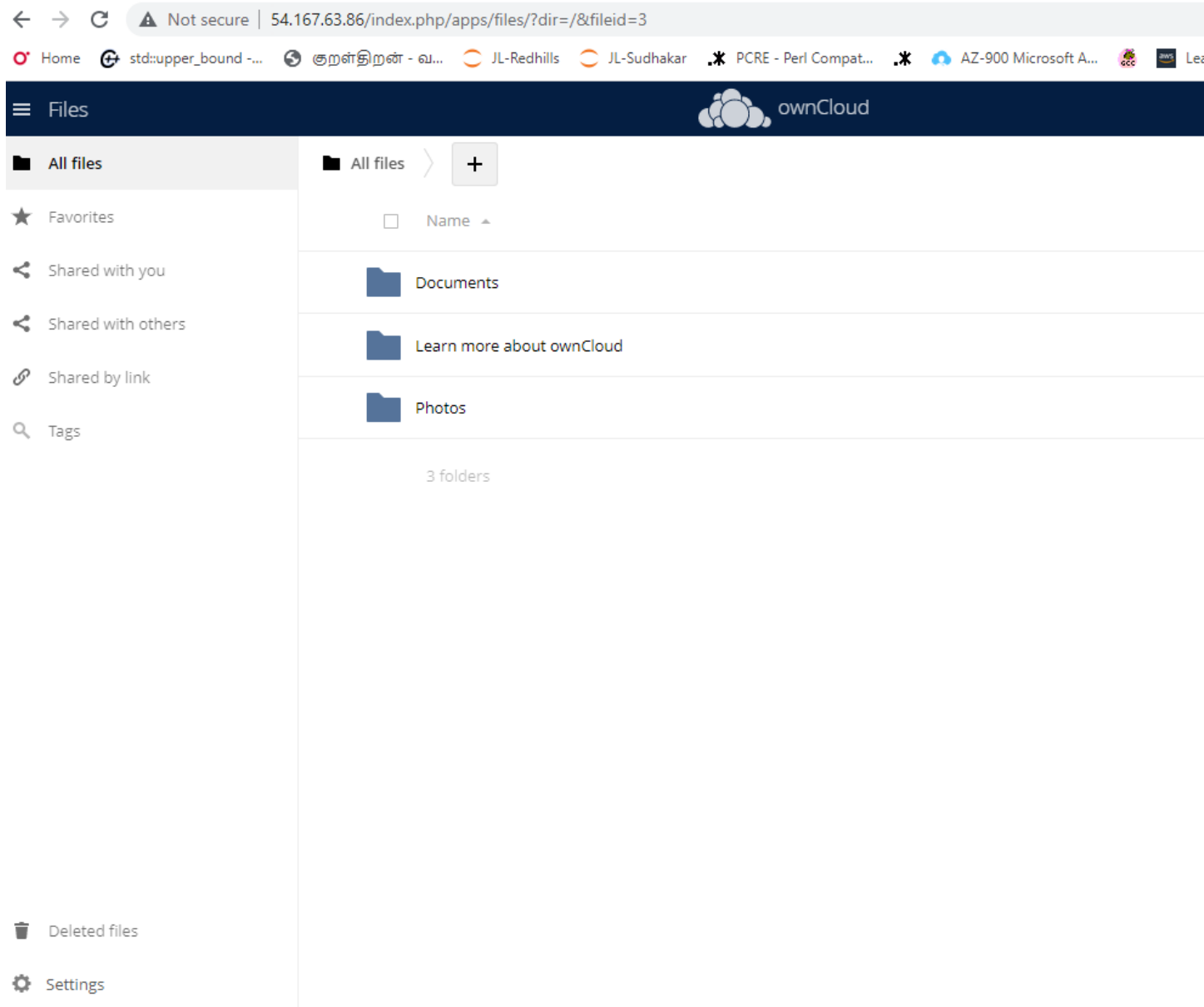
60 cd /var/www/
61 chown -R www-data. owncloud
62 systemctl restart apache2
63 history
64 history -h
65 history
66 history 30
67 history
root@ip-10-0-1-5: /var/www#

```

STEP 16 Login to OwnCloud (user = admin)

Screenshot





STEP 17: Install S3 External storage app in Own Cloud

After logging in as admin, navigate to Market/Storage/External Storage S3.

Install the external storage S3 plugin.

Screenshot

Market

Show all

App Bundles

CATEGORIES

Automation

Collaboration

Customization

External plugins

Games

Integration

Multimedia

Productivity

Security

Storage

Tools

SETTINGS

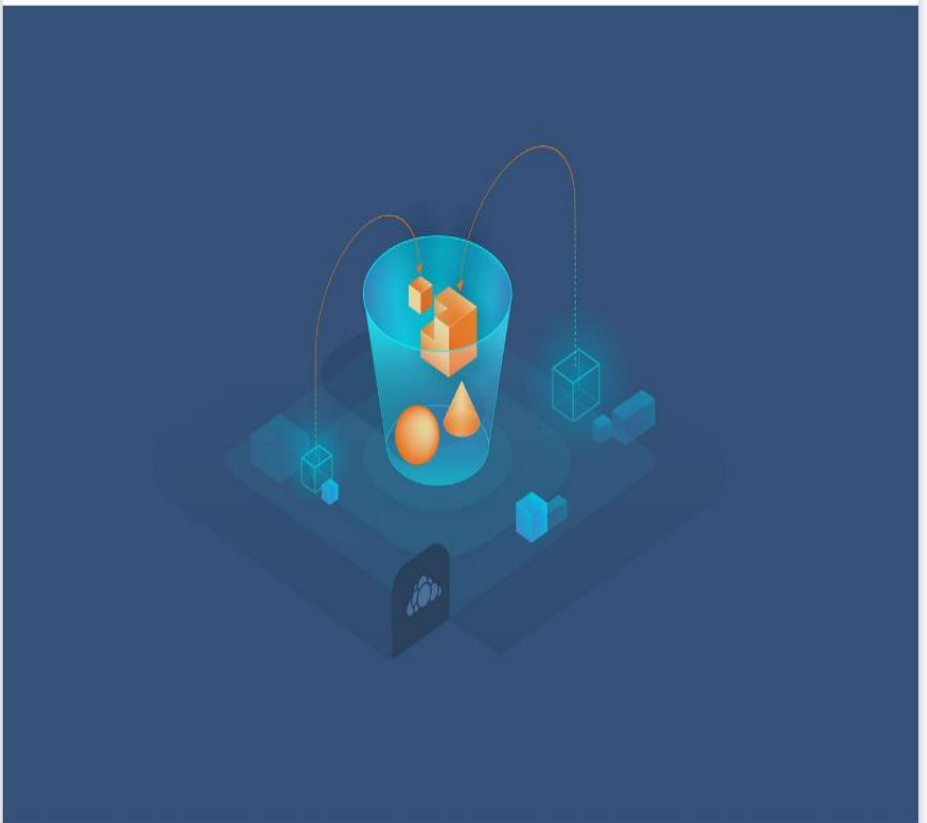
Add API Key

Clear cache

External Storage: S3

📁 storage

☆☆☆☆



With ownCloud you can leverage storage that already exists in your enterprise. Administrators can for example decide to store the most sensitive data on-premises and less sensitive data in the cloud while making all of your files available through a single user interface and enabling compliance with your chosen security and governance policies. This highly facilitates data infrastructure modernization in your business without the need of data migration or major investments in new storage systems. Using ownCloud it is possible to integrate a wide range of storage types and

Not secure | 54.167.63.86/index.php/apps/market/#/app/files_external_s3

Home std:upper_bound -... குறள்திறன் - வ... JL-Redhills JL-Sudhakar PCRE - Perl Compat... AZ-900 Microsoft A... Learner Lab How To Install Linux...

Market ownCloud

- Security
- Storage
- Tools
- SETTINGS
 - Add API Key
 - Clear cache

With ownCloud you can leverage storage that already exists in your enterprise. Administrators can for example decide to store the most sensitive data on-premises and less sensitive data in the cloud while making all of your files available through a single user interface and enabling compliance with your chosen security and governance policies. This highly facilitates data infrastructure modernization in your business without the need of data migration or major investments in new storage systems. Using ownCloud it is possible to integrate a wide range of storage types and make them accessible from all devices under the specified security policies in ownCloud. ownCloud works as an additional layer, providing users with a single point of access to their files on all storage locations.

This extension enables ownCloud Server to communicate with the widely spread S3 protocol (S3 HTTP API) to integrate object storages as external storages. Once an object storage is integrated by an administrator or the user in self-service, users see a folder structure that mirrors the hierarchy of the external storage. Users can then selectively sync files to their desktop or access them on mobile devices like any other file in ownCloud. The files will remain on the external storage but are now accessible in a secure and fully logged and traceable way through ownCloud.

Administrators and users (when enabled) can find external storage configuration options in the 'Storage' section of admin and user settings, respectively.

DEVELOPER	VERSION	LICENSE
ownCloud	2.1.0	GPLv2

App files_external_s3 installed successfully

Step 18: Add S3 buckets to External Storage using `aws_access_key_id` and `aws_secret_access_key`

Screenshot

← → ↻ Not secure | 54.167.63.86/index.php/settings/admin?sectionid=storage

Home std::upper_bound ~... குறள்திறன் - வ... JL-Redhills JL-Sudhakar * PCRE - Perl Compat... * AZ-900 Microsoft A... AWS

Settings ownCloud

Personal

- General
- Storage
- Sharing
- Security
- Additional

Admin

- Apps
- General
- Storage**
- Encryption
- Sharing
- Help & Tips
- Additional

External Storage

☒ Enable external storage

External storage has been disabled by the administrator

Folder name	External storage	Authentication	Configuration
			balagurusamy Hostname
			us-east-1
<input checked="" type="checkbox"/>	Balagurusamy	Amazon S3 compatible (SDK v3)	Access key ▾
			<input checked="" type="checkbox"/> Enable SSL
			<input type="checkbox"/> Enable Path Style
			AKIAXLWML4GQHKW

Folder name Add storage ▾

☐ Allow users to mount external storage

STEP 19: Create users & groups and grant access to S3 bucket.

Users; bluehills -Group blue

Redhills- Group red

Screenshot

← → ↻ ⚠ Not secure | 54.167.63.86/index.php/settings/admin?sectionid=storage

Home std::upper_bound ~... குறள்திறன் - வ... JL-Redhills JL-Sudhakar ✖ PCRE - Perl Compat... ✖ AZ-900 Microsoft A...

Settings External storage has been added successfully

Personal




- General
- Storage
- Sharing
- Security
- Additional

Admin

- Apps
- General
- Storage**
- Encryption
- Sharing
- Help & Tips
- Additional

External Storage

☒ Enable external storage

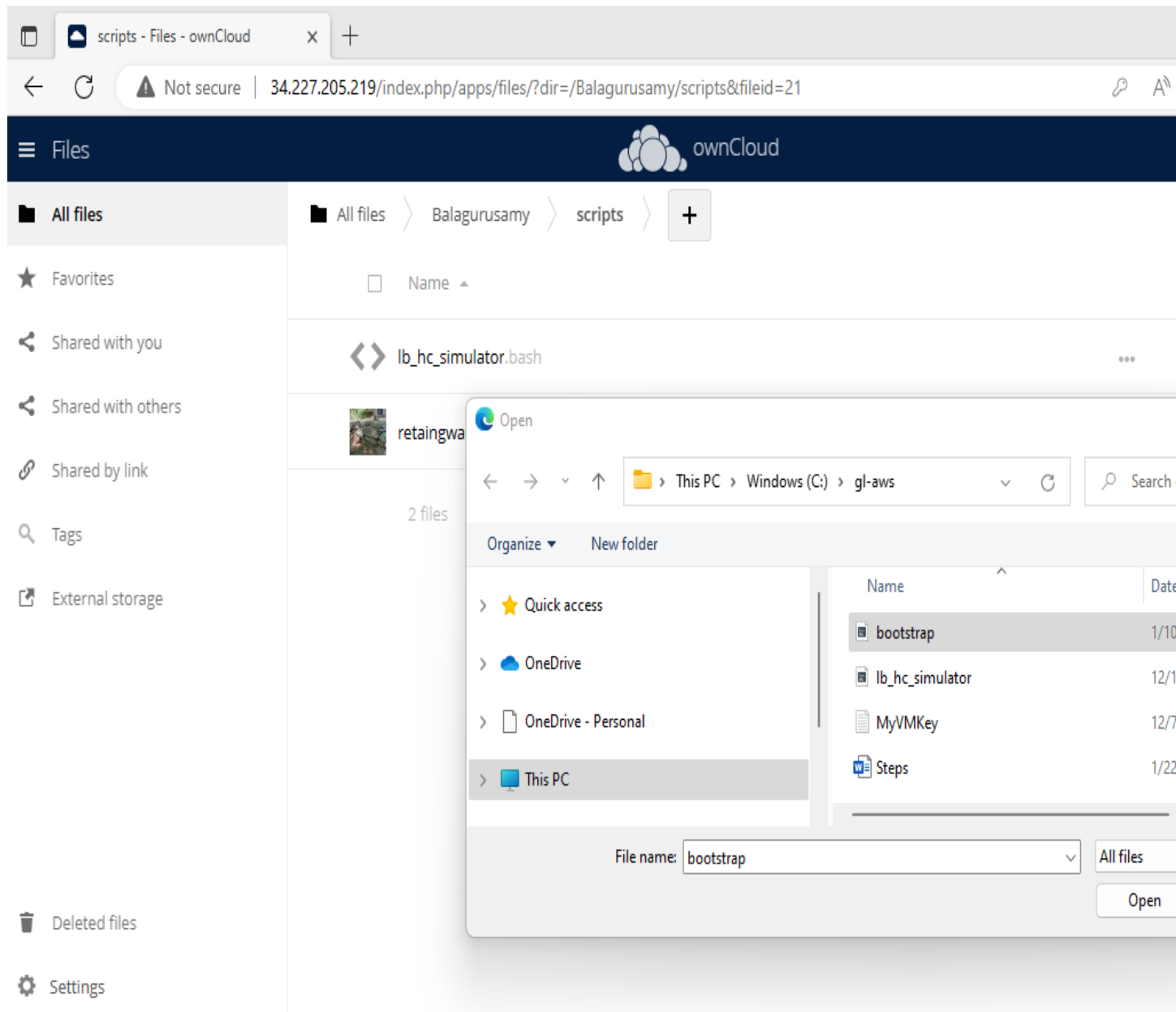
Folder name	External storage	Authentication	Configuration
 Balagurusamy	Amazon S3 compatible (SDK v3)	Access key ▾	<div>balagurusamy Hostname</div> <div>us-east-1</div> <div><input checked="" type="checkbox"/> Enable SSL</div> <div><input type="checkbox"/> Enable Path Style</div> <div>AKIAXLWML4GQHKW</div>
 red	Local	None ▾	/home/ubuntu/red
 blue	Local	None ▾	/home/ubuntu/blue
Folder name	Add storage ▾		

☐ Allow users to mount external storage

STEP 20: login in using user account : bluehills and upload some files to S3 bucket

Navigate to All Files > S3 Bucket (Balagurusamy) and click + button to upload.

Screenshot



STEP 21: Cleanup

Create a key pair gl-keypair

AWS CLI (bash)	<pre> #terminate the instance both app & db servers aws ec2 terminate-instances --instance-ids \$gl_app_server aws ec2 terminate-instances --instance-ids \$gl_db_server #delete NAT gateway aws ec2 delete-nat-gateway --nat-gateway-id \$gl_nat_gw #delete security groups aws ec2 delete-security-group --group-id \$gl_pub_sg aws ec2 delete-security-group --group-id \$gl_pri_sg #delete subnets aws ec2 delete-subnet --subnet-id \$gl_pub_subnet aws ec2 delete-subnet --subnet-id \$gl_pri_subnet #delete EC2 route tables aws ec2 delete-route-table --route-table-id \$gl_rt_gateway aws ec2 delete-route-table --route-table-id \$gl_rt_pri #detach internet-gateway aws ec2 detach-internet-gateway --internet-gateway-id \$gl_gateway --vpc-id \$gl_vpc #delete internet-gateway aws ec2 delete-internet-gateway --internet-gateway-id \$gl_gateway #delete vpc aws ec2 delete-vpc --vpc-id \$gl_vpc #release address aws ec2 release-address --allocation-id \$gl_nat_elastic_ip </pre>
AWS Console	<ol style="list-style-type: none"> 1. Terminate Instances 2. Delete NAT GATEWAY 3. Delete security groups 4. Delete Subnets 5. Delete route tables 6. Detach and delete Internet gateways 7. Delete vpc 8. Release elastic IP

Screenshots

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Successfully terminated i-04b879780c450d424,i-0fc6ec71595aa51c4

Instances (2) Info



Connect

Instance state

Find instance by attribute or tag (case-sensitive)

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status
<input type="checkbox"/>	gl-db-server	i-04b879780c450d424	Terminated	t2.micro	-
<input type="checkbox"/>	gl-app-server	i-0fc6ec71595aa51c4	Terminated	t2.micro	-

Internet gateway igw-04a8ed3f2a451682f successfully detached from vpc-07aa93bf04d9a3386

Internet gateways (1/2) Info

Filter Internet gateways

Name	Internet gateway ID	State	VPC ID	Owner
gl-gateway	igw-04a8ed3f2a451682f	Detached	vpc-07aa93bf04d9a3386	6893492
-	igw-04aa6153c1f11e6dc	Detached	vpc-07aa93bf04d9a3386	6893492

Delete internet gateway

Are you sure that you want to delete this internet gateway?

- igw-04a8ed3f2a451682f - gl-gateway

To confirm deletion, type *delete* in the field:

Cancel Delete internet gateway

Details

Internet gateway ID
igw-04a8ed3f2a451682f

State
Detached

VPC ID
-

Owner
6893492

aws Services Search [Alt+S] N. Virginia user230546

EC2 Select a VPC

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists

You successfully deleted nat-01e584fcb48fa8019.

NAT gateways (1/1) Info

Filter NAT gateways

	Name	NAT gateway ID	Connectivit...	State	State message	Elastic IP address	Pri
+	-	nat-01e584fcb48fa8019	Public	Deleting	-	3.224.58.10	10.

nat-01e584fcb48fa8019

Your VPCs (1/2) Info

Filter VPCs

Name

VPC ID

State

IPv4 CIDR

IPv6 CIDR

DHCP option set

gl-vpc

default-vpc

vpc-07aa93bf04d9a3386 / gl-vpc

Details

CIDRs

Flow logs

Details

VPC ID

vpc-07aa93bf04d9a3386

Tenancy

✔ Will be deleted

This VPC will be deleted permanently and cannot be recovered later:

Name

gl-vpc

VPC ID

vpc-07aa93bf04d9a3386

State

✔ Available

To confirm deletion, type *delete* in the field:

delete

Cancel

Delete

Elastic IP addresses (1/1)

Filter Elastic IP addresses

Name

Allocated IPv4 address

Type

Allocation ID

Reverse DNS record

Associate

-

3.224.58.10

Public IP

eipalloc-030a4551ec07dce15

-

-

3.224.58.10

Summary

Tags

Summary

Allocated IPv4 address

Type

Allocation ID

Reverse DNS record

Release Elastic IP addresses

✕

Will be released

If you release the following Elastic IP addresses, they will no longer be allocated to your account and you can no longer associate them with your resources.

Name

▲

IPv4 address

▼

Allocation ID

▼

Address type

▼

-

3.224.58.10

eipalloc-030a4551ec07dce15

Public

Cancel

Release