

Creating a file share & sync solution using ownCloud and AWS

This project's goal is to set up the infrastructure necessary to run Owncloud applications with a MySQL database by utilizing AWS resources.

ownCloud:

ownCloud is an open source secure file sync and share solution which can help you gain control of this situation and enable you to create and deploy an enterprise scale file solution. ownCloud can run in your data center or on a public cloud, with its servers, storage etc completely managed and controlled by your IT team and management in accordance with your company's governance and security requirements.

Scenarion:

According to recent research, 40-75% of employees are using Dropbox to share files inside and outside of their businesses. Half of those Dropbox users do this even though they know it's against the rules. More than 40% of businesses have experienced the exposure of confidential information and the estimated average cost of a data breach equaled \$5.5 Million in 2011.



These files, containing sensitive company and customer data, are stored in a public cloud outside of the businesses' control – possibly even outside of the country. The potential for data leakage and security breaches is enormous and companies need to stay compliant with their own policies and procedures for security and governance.

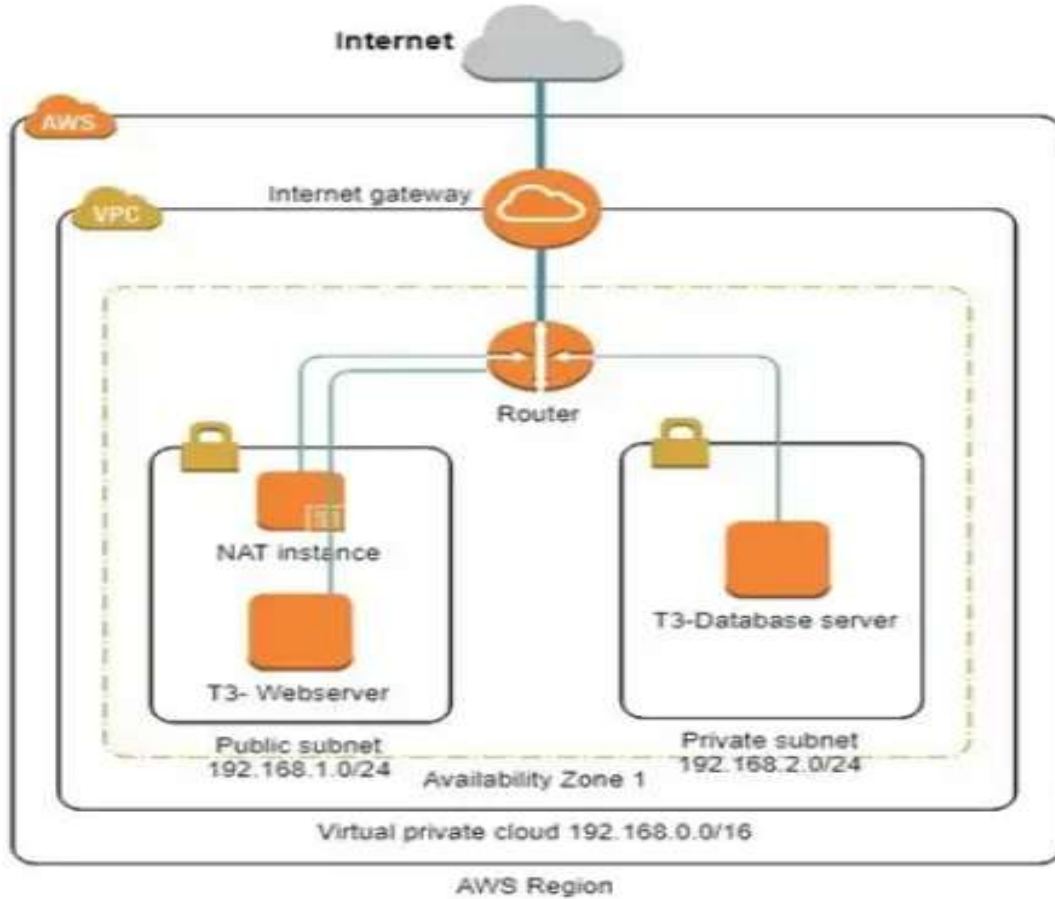
Objective:

Implement the ownCloud solution for a small workgroup, which can cater upto 150 users by using various AWS services. Your solution will be completely deployed on AWS.

Implementation Objective:

- Implement 2 different subnets (one public and the other private) in a custom VPC called owncloud-vpc.
- Install and configure MySQL database to run on the private subnet. This subnet should be associated with a security group that allows traffic to private subnet only from the public subnet.
- The ownCloud app should be installed in public subnet and MUST be configured to access a new database called owncloud-db (created by you) in the private subnet. Apache HTTP server should host ownCloud application in this subnet and must be configured with required PHP modules for ownCloud.

Network Diagram & Design Considerations



- The implementation will have two subnets (public and private) to separate web application server and database server.
- Custom VPC (*owncloud-project1*) will have CIDR 192.168.0.0/16
- Public subnet will have CIDR 192.168.1.0/24 called *Owncloud-Public*. Auto-assign IP address will be set as enabled.
- Private subnet will have CIDR 192.168.2.0/24 called *Owncloud-Private*.
- Custom route table (*OwnCloud-PublicRT*) will be assigned to Public subnet.
- Default route table (*Owncloud-defaultroute*) will be assigned to Private subnet.

- *OwnCloud-PublicRT* will have bi-directional internet route entry using Amazon internet gateway.
- *Owncloud-defaultroute* will have NAT instance route entry to facilitate internet connectivity to private subnet.
- Two T3.micro EC2 instances will be used with Ubuntu 18.* LTS Amazon machine images.
- Two security groups will be created.

Security Group:

A security group acts as a virtual firewall that controls the traffic for one or more instances. When we launch an instance, we can specify one or more security groups; otherwise, default security group is used.

To implement this two security groups are created:

Owncloud-public-SG:

This will be assigned to Owncloud web server EC2 instance in public subnet. It opens SSH port 22 for remote access and HTTP port 80 for web access. This opens unrestricted access to above ports for the world.

Basic details

Security group name
Info

Owncloud-public-SG

Name cannot be edited after creation.

Description
Info

Allows SSH access to developers

VPC
Info

vpc-05c0a05d5962f876c
X

Inbound rules
Info

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
SSH	TCP	22	Anywhere-I... 0.0.0.0/0 X		Delete
HTTP	TCP	80	Anywhere-I... 0.0.0.0/0 X		Delete
Add rule					

Outbound rules
Info

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info	
All traffic	All	All	Custom 0.0.0.0/0 X		Delete

Owncloud-Appserver-Traffic:

This will be assigned to database server EC2 instance in private subnet. This will enable restricted access to server from Public subnet only. It opens SSH port 22 for remote access and MYSQL DB port 3306 for remote database connection.

Basic details

Security group name [Info](#)

Owncloud-Appserver-Traffic

Name cannot be edited after creation.

Description [Info](#)

Allows SSH access to developers

VPC [Info](#)

Q vpc-0cc22345ed1024214

X

Inbound rules [Info](#)

Type [Info](#)

SSH

▼

Protocol [Info](#)

TCP

Port range [Info](#)

22

Source [Info](#)

Custom

▼

Q

192.168.1.0/24 X

Description - optional [Info](#)

Delete

Type [Info](#)

MYSQL/Aurora

▼

Protocol [Info](#)

TCP

Port range [Info](#)

3306

Source [Info](#)

Custom

▼

Q

192.168.1.0/24 X

Description - optional [Info](#)

Delete

Add rule

Outbound rules [Info](#)

Type [Info](#)

All traffic

▼

Protocol [Info](#)

All

Port range [Info](#)

All

Destination [Info](#)

Custom

▼

Q

0.0.0.0/0 X

Description - optional [Info](#)

Delete

Procedure:

1. Create Ubuntu 18.04 instance using 7 steps workflow. Open ports 80 and 22 using security group.

a. Choose ubuntu AMI

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

AMI from catalog

Quick Start

Amazon Machine Image (AMI)

ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20221201-ami-061dbd1209944525c

Free tier eligible
Verified provider

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Catalog	Published	Architecture	Virtualization	Root device type	ENA Enabled
Quickstart AMIs	2022-12-02T05:31:59.000Z	x86_64	hvm	ebs	Yes

Number of instances Info

Software Image (AMI)

Ubuntu Server 18.04 LTS (HVM),...[read more](#)
ami-061dbd1209944525c

Virtual server type (instance type)

t2.micro

Firewall (security group)

Owncloud-public-SG

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, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch Instance

b. Used custom VPC and public subnet.

The screenshot shows the 'Network settings' section of an AWS EC2 instance configuration. It includes a 'VPC' dropdown set to 'vpc-05c0a05d5962f876c (owncloud-project1)', a 'Subnet' dropdown set to 'subnet-05aa472137c731ba2' (labeled 'Owncloud-Public'), and an 'Auto-assign public IP' dropdown set to 'Enable'. The 'Firewall (security groups)' section has two radio buttons: 'Create security group' (unselected) and 'Select existing security group' (selected). Below this, a 'Common security groups' dropdown is set to 'Owncloud-public-SG sg-025cbac7769fe2f4d'. A 'Compare security group rules' link is visible. At the bottom, there is a link to 'Advanced network configuration'.

▼ Network settings Info

VPC - required Info

vpc-05c0a05d5962f876c (owncloud-project1) 192.168.0.0/16

Subnet Info

subnet-05aa472137c731ba2 Owncloud-Public
VPC: vpc-05c0a05d5962f876c Owner: 512497007024 Availability Zone: us-east-1a
IP addresses available: 251 CIDR: 192.168.1.0/24

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ Select existing security group

Common security groups Info

Select security groups

Owncloud-public-SG sg-025cbac7769fe2f4d X
VPC: vpc-05c0a05d5962f876c

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

► Advanced network configuration

c. Used below bootstrap script to update repository, add key file and install ownserver cloud server files automatically while the instance is launched.

```
#!/bin/bashapt-get update -y

apt-get install apache2 -y

apt-get install php libapache2-mod-php php-mysql curl -y

cd /tmp

wget

https://download.owncloud.org/download/repositories/production/Ubuntu 18.04/Release.key

apt-key add Release.key

echo 'deb
```

https://download.owncloud.org/download/repositories/production/Ubuntu_18.04/ /' | sudo tee /etc/apt/sources.list.d/owncloud.list

apt update

apt install php-bz2 php-curl php-gd php-imagick php-intl php-mbstring php-xml php-zip owncloud-files -y

User data [Info](#)

```
#!/bin/bash
apt-get update -y
apt-get install apache2 -y
apt-get install php libapache2-mod-php php-mysql curl -y
cd /tmp
wget
https://download.owncloud.org/download/repositories/production/Ubuntu_18.04
/Release.key
apt-key add Release.key
echo 'deb
https://download.owncloud.org/download/repositories/production/Ubuntu_18.04
/' | sudo tee /etc/apt/sources.list.d/owncloud.list
apt update
apt install php-bz2 php-curl php-gd php-imagick php-intl php-mbstring php-xml
php-zip owncloud-files -y
```

☐ User data has already been base64 encoded

d. Assigned below security group

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group

☒ Select existing security group

Common security groups [Info](#)

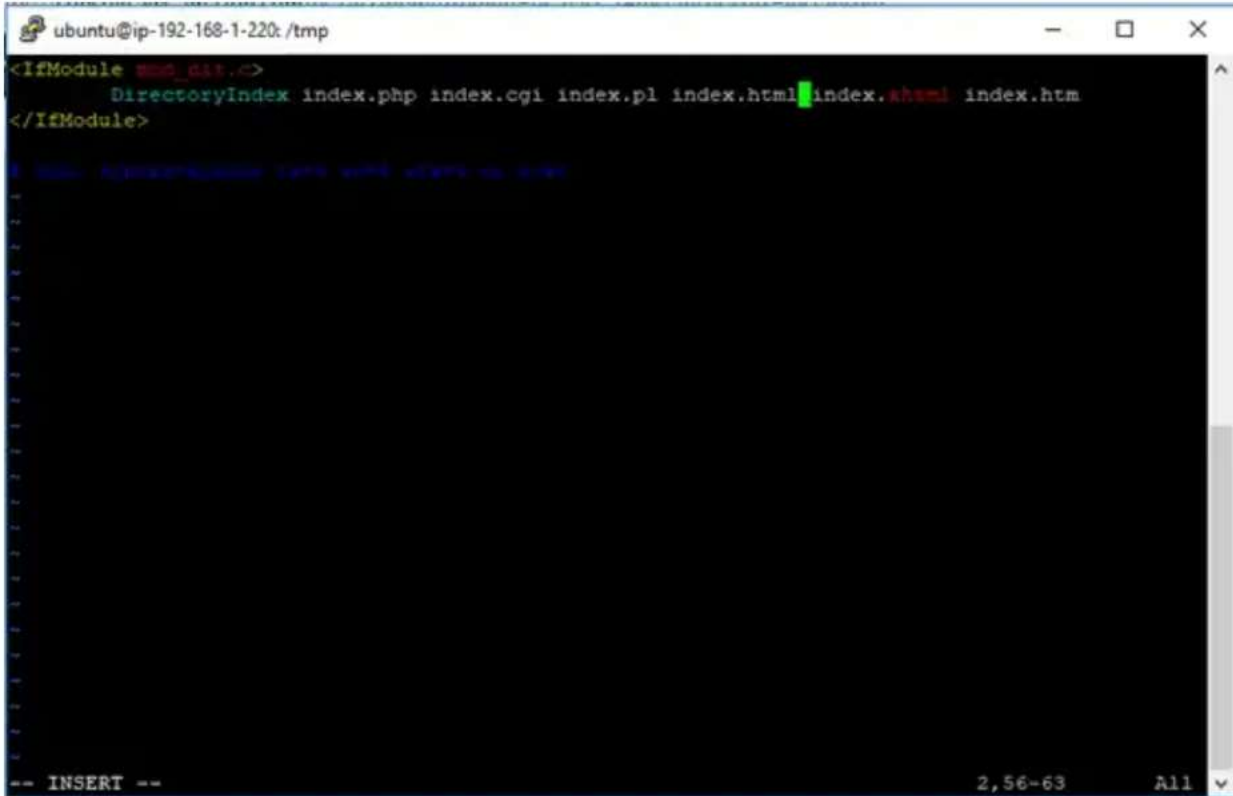
Select security groups

Owncloud-public-SG sg-025cbac7769fe2f4d X
VPC: vpc-05c0a05d5962f876c

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

2. Make index.php as the default first load page and restart the server.



The screenshot shows a terminal window with the title bar "ubuntu@ip-192-168-1-220: /tmp". The terminal content shows the configuration of the `DirectoryIndex` directive in the `httpd.conf` file. The current configuration is:

```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.cgi index.pl index.html index.shtml index.htm
</IfModule>
```

The cursor is positioned at the end of the line `index.shtml`. The terminal also shows a status bar at the bottom with the text "INSERT" and "2,56-63 All".

Implementing MySQL on Ubuntu EC2 instance in private subnet

MySQL database is deployed in private subnet which is accessible only by subnet of owncloud webserver.

Procedure:

1. Create Ubuntu 18.04 instance using 7 steps workflow.

```
#!/bin/bash
apt-get update -y
apt install mysql-server -y
```

Name

MySQL-Server

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

Recents

Quick Start

Recently launched

Currently in use

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20221201

ami-061dbd1209944525c

2022-12-02T05:31:59.000Z

architecture: 64-bit (x86)

Virtualization: hvm

ENA enabled: true

Root device type: ebs

Description

Canonical, Ubuntu, 18.04 LTS, amd64 bionic image build on 2022-12-01

Architecture

AMI ID

x86_64

ami-061dbd1209944525c

Verified provider

▼ Summary

Number of instances Info

1

Software Image (AMI)

Canonical, Ubuntu, 18.04 LTS, ...read more

ami-061dbd1209944525c

Virtual server type (instance type)

t2.micro

Firewall (security group)

Owncloud-Appserver-Traffic

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, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance


User data Info

```
#!/bin/bash
apt-get update -y
apt install mysql-server -y
```



2. Using Owncloud-Appserver-Traffic security group which allows traffic only from public subnet.

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-05c0a05d5962f876c (owncloud-project1)
192.168.0.0/16 ▼ 

Subnet [Info](#)

subnet-01a9c597429f80b94
Owncloud-Private
VPC: vpc-05c0a05d5962f876c Owner: 512497007024 Availability Zone: us-east-1a
IP addresses available: 251 CIDR: 192.168.2.0/24 ▼  [Create new subnet](#) 

Auto-assign public IP [Info](#)

Disable ▼


Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group

☒ Select existing security group

Common security groups [Info](#)

Select security groups ▼  [Compare security group rules](#)

Owncloud-Appserver-Traffic sg-0b3dc5fa2016f892a ✕
VPC: vpc-05c0a05d5962f876c

Security groups that you add or remove here will be added to or removed from all your network interfaces.

► Advanced network configuration

3. SSH to database server from primary but internet is not available yet because we still need to create NAT gateway

4. After setting up NAT Gateway and configuring route table, we are able to access internet.

```
ubuntu@ip-192-168-2-245: ~  
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [983 kB]  
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [299 kB]  
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [11.9 kB]  
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [5764 B]  
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [2512 B]  
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [1644 B]  
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [3732 B]  
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [1696 B]  
Get:20 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [464 kB]  
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [160 kB]  
Get:22 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [577 kB]  
Get:23 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [189 kB]  
Get:24 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [4008 B]  
Get:25 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2060 B]  
Fetched 17.7 MB in 4s (4950 kB/s)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
14 packages can be upgraded. Run 'apt list --upgradable' to see them.  
ubuntu@ip-192-168-2-245:~$
```

5. Created DB user and change bind address.

```
CREATE DATABASE owncloud

CREATE USER 'owncloud' @ 'localhost' IDENTIFIED BY 'password' ;

CREATE USER 'owncloud' @ '%' IDENTIFIED BY 'password' ;

GRANT ALL PRIVILEGES ON *.* to owncloud@localhost IDENTIFIED BY
'password' WITH GRANT OPTION;

GRANT ALL PRIVILEGES ON *.* to owncloud@'%' IDENTIFIED BY
'password' WITH GRANT OPTION;FLUSH PRIVILEGES;EXIT;
```

```

myqsig_idn]
socket      = /var/run/myqsigd/myqsigd.sock
size        = 0
}
myqsigd]
#
#
#
myqsig      = myqsig
pid-file    = /var/run/myqsigd/myqsigd.pid
confdir     = /var/run/myqsigd/myqsigd.conf
var         = /var
logdir       = /var
statefile    = /var/lib/myqsig
logdir       = /var
log-messages-dir = /var/log/myqsig
log-external-locking
#
#
#
#
bind-address = 0.0.0.0
}
#
my-buffer-size = 100
"/etc/myqsig/myqsig.conf.d/myqsig.conf" 104L 20110

```

Creating NAT Gateway for internet access to private subnet

This is similar to above. The only difference is using Amazon AMI for NAT. Public subnet needs to be used. Here is the screenshot of NAT instance:

NAT gateways (1/1) [Info](#)

Filter NAT gateways

	Name	NAT gateway ID	Connectivit...	State	State message
	Owncldw-NatGW	nat-0381c4c004fd1ac11	Public	Available	-

Here is the final view of EC2 dashboard:

Instances (2) [Info](#)

Find instance by attribute or tag (case-sensitive)

Connect Instance state Actions Launch instances

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	Webserver	i-0eae543a41ddc83b8	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	-	54.209.164.207	-
<input type="checkbox"/>	MySQL-Server	i-01ce25d948984b3ee	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	-	-	-

After this we need to add NAT gateway to private subnet route, refer here.

Subnets without explicit associations (2)

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

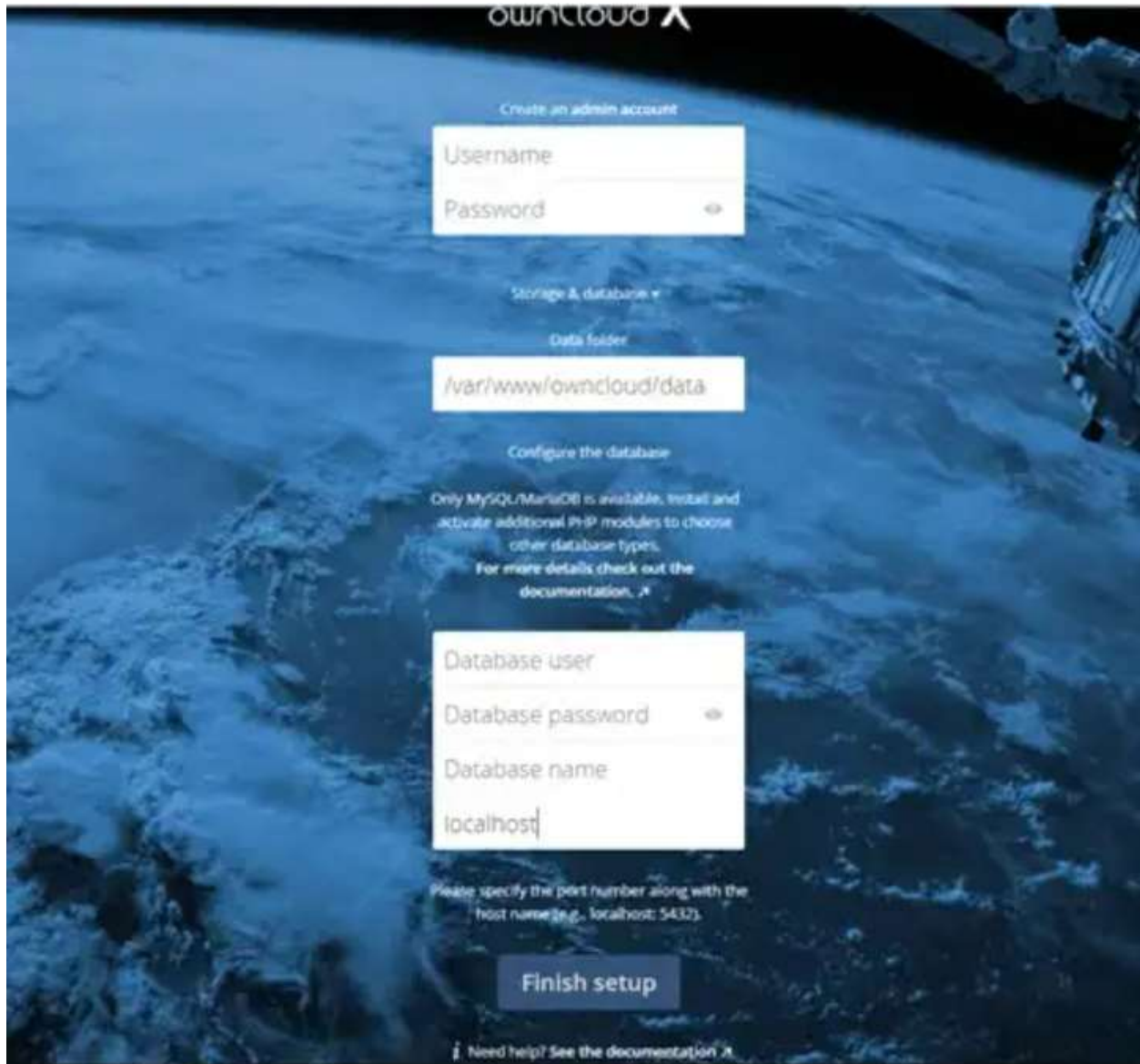
Find subnet association

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-05aa472137c731ba2 / Owncldw-Public	192.168.1.0/24	-
subnet-01a9c597429f80b94 / Owncldw-Private	192.168.2.0/24	-

Completing owncloud configuration

Once all prerequisites are in place and all instances are up and running below steps need to be taken to complete the configuration.

1. Access owncloud webpage at [http:// 54.209.164.207](http://54.209.164.207)

A screenshot of the OwnCloud configuration wizard interface. The background is a dark blue image of a ship's wake. The wizard is centered and consists of several steps. The first step is 'Create an admin account' with input fields for 'Username' and 'Password'. The second step is 'Storage & database' with a sub-step 'Data folder' showing the path '/var/www/owncloud/data'. The third step is 'Configure the database' with a note that only MySQL/MariaDB is available. It has input fields for 'Database user', 'Database password', and 'Database name' (containing 'localhost'). Below these is a note about specifying a port number. At the bottom is a 'Finish setup' button and a link to documentation.

owncloud

Create an admin account

Username

Password

Storage & database

Data folder

/var/www/owncloud/data

Configure the database

Only MySQL/MariaDB is available. Install and activate additional PHP modules to choose other database types. For more details check out the documentation.

Database user

Database password

Database name

localhost

Please specify the port number along with the host name (e.g., localhost:5432).

Finish setup

Need help? See the documentation

2. Enter admin user and password. Enter database values.
These values are taken from previous configuration.

Username: owncloud

Password: owncloud

Data folder: /var/www/owncloud/data

Database user: owncloud (created above)

Database password: password (created above)

Database name: owncloud (created above)

Database server IP: <IP OF database server in private subnet:port>

