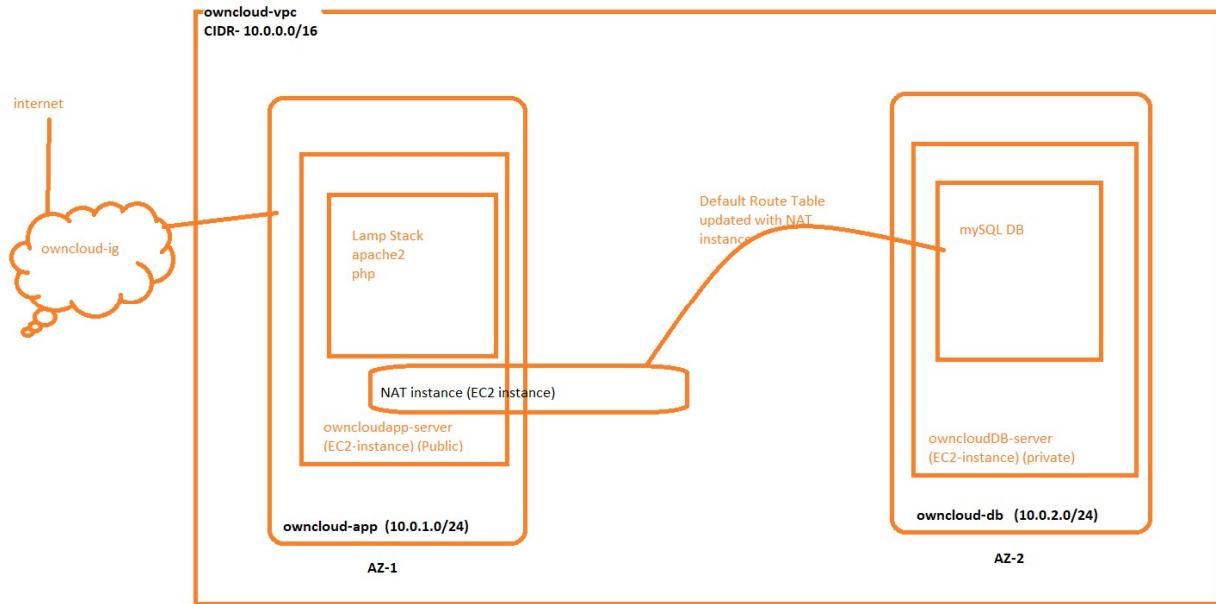


1. **Architecture**- Below is the architecture to create the scenario.



2. **Implementation Steps:** Below are the steps implementing it-

#### a- Create VPC and subnets

##### Creating owncloud-vpc with CIDR – 10.0.0.0/16

VPCs > Create VPC

Create VPC

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You must specify an IPv4 address range for your VPC. Specify the IPv4 address range as a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. You cannot specify an IPv4 CIDR block larger than /16. You can optionally associate an IPv6 CIDR block with the VPC.

Name tag	owncloud-vpc
IPv4 CIDR block*	10.0.0.0/16
IPv6 CIDR block	<input checked="" type="radio"/> No IPv6 CIDR Block <input type="radio"/> Amazon provided IPv6 CIDR block <input type="radio"/> IPv6 CIDR owned by me
Tenancy	Default

\* Required

**Create VPC** **Actions**

Filter by tags and attributes or search by keyword

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table	Main Network ACL	Tenancy	Default
vpc-b7f003dc	vpc-b7f003dc	available	172.31.0....	-	dopt-735f8a18	rtb-be47ffd5	acl-521cc239	default	Yes
<b>owncloud-vpc</b>	<b>vpc-0ccebc1c4a39a20362</b>	<b>available</b>	<b>10.0.0.0/16</b>	<b>-</b>	<b>dopt-735f8a18</b>	<b>rtb-00f0587cbad5c5af0</b>	<b>acl-07f523ca5a94db332</b>	<b>default</b>	<b>No</b>

Creating two subnets- 1. owncloud-app CIDR: 10.0.1.0/24 and 2. Owncloud-db CIDR: 10.0.2.0/24

Screenshot of the AWS Subnets > Create subnet page.

**Create subnet**

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag	owncloud-app		
VPC*	vpc-0cceeb1c4a39a20362		
Availability Zone	us-east-2a		
VPC CIDRs	CIDR	Status	Status Reason
10.0.0.0/16		associated	
IPv4 CIDR block*	10.0.1.0/24		

\* Required

**Create subnet** **Create**

**Subnets**

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Z.	Route table
<input checked="" type="checkbox"/> owncloud-db	subnet-00c6020a9b0feb161	available	vpc-0cceeb1c4a39a20362   owncloud-vpc	10.0.2.0/24	251	-	us-east-2b	use2-az2	rtb-00f0587cbad5c5af0
<input type="checkbox"/> owncloud-app	subnet-0173e43c4978bef00	available	vpc-0cceeb1c4a39a20362   owncloud-vpc	10.0.1.0/24	251	-	us-east-2a	use2-az1	rtb-00f0587cbad5c5af0
<input type="checkbox"/>	subnet-b5051edd	available	vpc-b7f003dc	172.31.0.0/20	4091	-	us-east-2a	use2-az1	rtb-be47ffd5
<input type="checkbox"/>	subnet-d183d2ab	available	vpc-b7f003dc	172.31.16.0/20	4091	-	us-east-2b	use2-az2	rtb-be47ffd5
<input type="checkbox"/>	subnet-fd1da4b1	available	vpc-b7f003dc	172.31.32.0/20	4091	-	us-east-2c	use2-az3	rtb-be47ffd5

**Description** **Flow Logs** **Route Table** **Network ACL** **Tags** **Sharing**

**Subnet: subnet-00c6020a9b0feb161**

Subnet ID	subnet-00c6020a9b0feb161	State	available
VPC	vpc-0cceeb1c4a39a20362   owncloud-vpc	IPv4 CIDR	10.0.2.0/24
Available IPv4 Addresses	251	IPv6 CIDR	-
Availability Zone	us-east-2b (use2-az2)	Route Table	rtb-00f0587cbad5c5af0
Network ACL	ad-07f523ca5a94db332	Default subnet	No
Auto-assign public IPv4 address	No	Auto-assign IPv6 address	No
Outpost ID	-	Owner	511554160123

## b- Creating IGW and update route table

Screenshot of the AWS Internet Gateways > Create internet gateway page.

**Create internet gateway**

**Internet gateway: igw-0bb973347093324aa**

Name	ID	State	VPC	Owner
<input checked="" type="checkbox"/> owncloud-ig	igw-0bb973347093324aa	attached	vpc-0cceeb1c4a39a20362   owncloud-vpc	511554160123
<input type="checkbox"/>	igw-d851ccb0	attached	vpc-b7f003dc	511554160123

**Description** **Tags**

ID	igw-0bb973347093324aa	Attached VPC ID	vpc-0cceeb1c4a39a20362   owncloud-vpc
State	attached	Owner	511554160123

Creating route table for owncloud-app subnet

Screenshot of the AWS Route Tables page showing the configuration of the 'owncloudapp-RT' route table.

**Route Table: rtb-0035346fc56d92925**

**Subnet Associations:**

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-0173e43c4978bef00...	10.0.1.0/24	-

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

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-00c6020a9b0feb1...	10.0.2.0/24	-

### Updating route table with owncloud-ig for internet access

Screenshot of the AWS Route Tables page showing the configuration of the 'owncloudapp-RT' route table.

**Route Table: rtb-0035346fc56d92925**

**Routes:**

View	All routes		
Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-0bb973347093324aa	active	No

- c- Creating EC2 instance – owncloudapp-server with ubuntu AMI having security group open for ssh, http and https and also associate it owncloud-app subnet**

Screenshot of the AWS EC2 Instances creation wizard Step 1: Choose an Amazon Machine Image (AMI).

**Step 1: Choose an Amazon Machine Image (AMI)**

<b>Red Hat Enterprise Linux 8 (HVM), SSD Volume Type</b> - ami-0a54ae4ef3b5f881 (64-bit x86) / ami-0ffd59b53e6797671 (64-bit Arm)	<input checked="" type="radio"/> 64-bit (x86)
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type Root device type: ebs Virtualization type: hvm ENA Enabled: Yes	<input type="radio"/> 64-bit (Arm)
<b>SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type</b> - ami-04c5bab51cc146925 (64-bit x86) / ami-02e73902018018171 (64-bit Arm)	<input checked="" type="radio"/> 64-bit (x86)
SUSE Linux Enterprise Server 15 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled. Root device type: ebs Virtualization type: hvm ENA Enabled: Yes	<input type="radio"/> 64-bit (Arm)
<b>Ubuntu Server 18.04 LTS (HVM), SSD Volume Type</b> - ami-07c1207a0d40bc3bd (64-bit x86) / ami-0a5ee0336de62011b (64-bit Arm)	<input checked="" type="radio"/> 64-bit (x86)
Ubuntu Server 18.04 LTS (HVM).EBS General Purpose (SSD) Volume Type. Support available from Canonical ( <a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a> ). Root device type: ebs Virtualization type: hvm ENA Enabled: Yes	<input type="radio"/> 64-bit (Arm)

## Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All Instance types Current generation Show/Hide Columns								
Currently selected: t2 micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)								
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2 nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro <span style="background-color: #00AEEF; color: white; padding: 2px;">Free tier eligible</span>	1	1	EBS only	-	Low to Moderate	Yes

1. Choose AMI   2. Choose Instance Type   3. Configure Instance   4. Add Storage   5. Add Tags   6. Configure Security Group   7. Review

## Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances <span style="color: #00AEEF;">(i)</span>	<input type="text" value="1"/> Launch into Auto Scaling Group <span style="color: #00AEEF;">(i)</span>
Purchasing option <span style="color: #00AEEF;">(i)</span>	<input type="checkbox"/> Request Spot instances
Network <span style="color: #00AEEF;">(i)</span>	<input type="text" value="vpc-0cce1c4a39a20362   owncloud-vpc"/> <span style="color: #00AEEF;">C</span> Create new VPC
Subnet <span style="color: #00AEEF;">(i)</span>	<input type="text" value="subnet-0173e43c4978bef00   owncloud-app   us-east-1"/> <span style="color: #00AEEF;">C</span> Create new subnet 251 IP Addresses available
Auto-assign Public IP <span style="color: #00AEEF;">(i)</span>	<input type="text" value="Use subnet setting (Enable)"/>
Placement group <span style="color: #00AEEF;">(i)</span>	<input type="checkbox"/> Add instance to placement group
Capacity Reservation <span style="color: #00AEEF;">(i)</span>	<input type="text" value="Open"/> <span style="color: #00AEEF;">C</span> Create new Capacity Reservation
IAM role <span style="color: #00AEEF;">(i)</span>	<input type="text" value="None"/> <span style="color: #00AEEF;">C</span> Create new IAM role
Shutdown behavior <span style="color: #00AEEF;">(i)</span>	<input type="text" value="Stop"/>

## Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type <span style="color: #00AEEF;">(i)</span>	Device <span style="color: #00AEEF;">(i)</span>	Snapshot <span style="color: #00AEEF;">(i)</span>	Size (GiB) <span style="color: #00AEEF;">(i)</span>	Volume Type <span style="color: #00AEEF;">(i)</span>	IOPS <span style="color: #00AEEF;">(i)</span>	Throughput (MB/s) <span style="color: #00AEEF;">(i)</span>	Delete on Termination <span style="color: #00AEEF;">(i)</span>	Encryption <span style="color: #00AEEF;">(i)</span>
Root	/dev/sda1	snap-0d2ae7855e80a405	<input type="text" value="8"/>	<input type="text" value="General Purpose SSD (gp2)"/> <span style="color: #00AEEF;">C</span>	100 / 3000	N/A	<input checked="" type="checkbox"/>	<input type="text" value="Not Encrypted"/>

[Add New Volume](#)

## Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic, you can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:  Create a new security group  
 Select an existing security group

Security Group ID	Name	Description
<input checked="" type="checkbox"/> sg-0c1b0fe6a497efc33	allow-https	allow HTTPS
<input checked="" type="checkbox"/> sg-00f2c701658990f59	allow-ssh-http	allow ssh and Http
<input type="checkbox"/> sg-05bc4d13c4f373b9a	allow-ssh-mysql	allow ssh and mysql
<input type="checkbox"/> sg-0573c6e69efce7100	default	default VPC security group

**Step 7: Review Instance Launch**

Security Group ID	Name
sg-00f2c701658990f59	allow-ssh-ht
sg-0c1b0fe6a497efc33	allow-https

All selected security groups inbound rules

Type	Protocol
HTTP	TCP
HTTP	TCP
SSH	TCP
SSH	TCP
HTTPS	TCP
HTTPS	TCP

Instance Details

Storage

Feedback English (US)

owncloud (1).pem

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair  
Key pair name: owncloud  
Download Key Pair

You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location**. You will not be able to download the file again after it's created.

Cancel Launch Instances

#### d- Creating owncloudDB-server EC2 instance-

**Step 3: Configure Instance Details**

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: Request Spot instances

Network: vpc-0ccebf1c4a39a20362 | owncloud-vpc Create new VPC

Subnet: subnet-00c6020a9b0feb161 | owncloud-db | us-east-1 Create new subnet  
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

Placement group: Add instance to placement group

Capacity Reservation: Open Create new Capacity Reservation

IAM role: None Create new IAM role

#### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group:  Create a new security group

Select an existing security group

Security Group ID	Name	Description	Actions
sg-0c1b0fe6a497efc33	allow-https	allow HTTPS	Copy to new
sg-00f2c701658990f59	allow-ssh-ht	allow ssh and Http	Copy to new
sg-05bc4d13c4f373b9a	allow-ssh-mysql	allow ssh and mysql	Copy to new
sg-0573c6e69efce7100	default	default VPC security group	Copy to new

#### e- Creating NAT-instance (EC2 instance)

[Cancel and Exit](#)

### Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	<input type="text" value="1"/>	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-0ccb1c4a39a20362   owncloud-vpc	<input type="button" value="Create new VPC"/>
Subnet	subnet-0173e43c4978bef00   owncloud-app   us-east-1	<input type="button" value="Create new subnet"/>
Auto-assign Public IP	<input type="checkbox"/> Use subnet setting (Enable)	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	Open	<input type="button" value="Create new Capacity Reservation"/>
IAM role	None <input type="button" value="Create new IAM role"/>	
Shutdown behavior	Stop <input type="button" value="Change"/>	

### Step 6: Configure Security Group

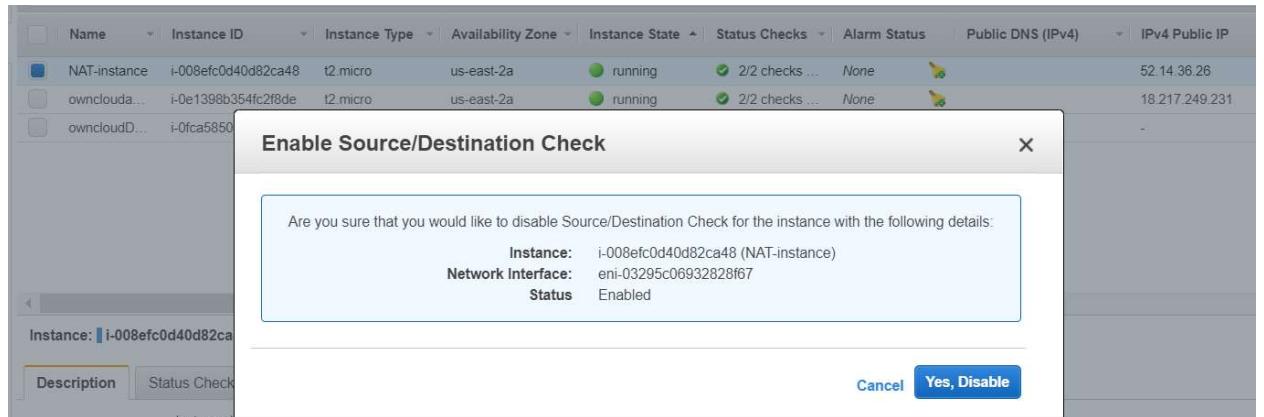
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow all unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:  Create a new security group  
 Select an existing security group

Security Group ID	Name	Description
<input type="checkbox"/> sg-0c1b0fe6a497efc33	allow-https	allow HTTPS
<input checked="" type="checkbox"/> sg-00f2c701658990f59	allow-ssh-http	allow ssh and Http
<input type="checkbox"/> sg-05bc4d13c4f373b9a	allow-ssh-mysql	allow ssh and mysql
<input type="checkbox"/> sg-0573c6e69efce7100	default	default VPC security group

### Step 7: Review Instance Launch

Disabling source/destination check for NAT-instance



#### f- Attaching NAT instance to default route table

Route Tables > Edit routes

#### Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	i-008efc0d40d82ca48	active	No

Add route \* Required

Cancel Save routes

Route Table ID	Explicit subnet association	Main	VPC ID
rtb-be47fd5	-	Yes	vpc-b7f003dc
rtb-0035346fc56d92925	subnet-0173e43c4978bef00	No	vpc-0ccebf1c4a39a20362   owncloud-vpc
rtb-00f0587cbad5c5af0	-	Yes	vpc-0ccebf1c4a39a20362   owncloud-vpc

Route Table: rtb-00f0587cbad5c5af0

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit routes

View All routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	eni-03295c06932828f67	active	No

Screenshot of the AWS VPC Dashboard showing the Subnets section. A subnet named 'owncloud-db' is selected.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone	Route table
owncloud-db	subnet-00c6020a9b0feb161	available	vpc-0ceb1c4a39a20362	10.0.2.0/24	250	-	us-east-2b	use2-az2	rtb-00f0587cbad5c5af0
owncloud-app	subnet-0173e43c4978bef00	available	vpc-0ceb1c4a39a20362	10.0.1.0/24	249	-	us-east-2a	use2-az1	rtb-0035346fcf
	subnet-b5051edd	available	vpc-b7f003dc	172.31.0.0/20	4091	-	us-east-2a	use2-az1	rtb-be47ffd5
	subnet-d183d2ab	available	vpc-b7f003dc	172.31.16.0/20	4091	-	us-east-2b	use2-az2	rtb-be47ffd5
	subnet-fd1da4b1	available	vpc-b7f003dc	172.31.32.0/20	4091	-	us-east-2c	use2-az3	rtb-be47ffd5

Route Table: rtb-00f0587cbad5c5af0

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	eni-03295c06932828f67

## g- Accessing Private instance (owncloudDB-server) via public instance (owncloudapp-server)

Screenshot of the AWS EC2 Instances page showing three instances: NAT-instance, owncloudapp-server, and owncloudDB-server.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
NAT-instance	i-008efc0d40d82ca48	t2.micro	us-east-2a	running	2/2 checks ...	None	
owncloudapp-server	i-0e1398b354fc2f8de	t2.micro	us-east-2a	running	2/2 checks ...	None	
owncloudDB-server	i-0fcfa585073377ad41	t2.micro	us-east-2b	running	2/2 checks ...	None	

Instance: i-0e1398b354fc2f8de (owncloudapp-server) Public IP: 18.217.249.231

Description Status Checks Monitoring Tags

Instance ID: i-0e1398b354fc2f8de	Public DNS (IPv4): -
Instance state: running	IPv4 Public IP: 18.217.249.231
Instance type: t2.micro	IPv6 IPs: -

```
29/04/2020 12:36.38 /home/mobaxterm/MyDocuments chmod 400 owncloud.pem

29/04/2020 12:36.54 /home/mobaxterm/MyDocuments ssh -i owncloud.pem ubuntu@18.217.249.231
Warning: Permanently added '18.217.249.231' (RSA) to the list of known hosts.
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1065-aws x86_64)

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

System information as of Wed Apr 29 07:07:38 UTC 2020

System load: 0.0          Processes:      86
Usage of /: 13.7% of 7.69GB Users logged in: 0
Memory usage: 15%          IP address for eth0: 10.0.1.139
Swap usage: 0%

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".

at MobaxTerm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```

```
ubuntu@ip-10-0-1-139:~$ cd /opt/
ubuntu@ip-10-0-1-139:/opt$ ls -al
total 8
drwxr-xr-x 2 root root 4096 Apr  8 17:35 .
drwxr-xr-x 23 root root 4096 Apr 29 06:28 ..
ubuntu@ip-10-0-1-139:/opt$ sudo chown ubuntu:ubuntu -R /opt
ubuntu@ip-10-0-1-139:/opt$ ls -al
total 8
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr  8 17:35 .
drwxr-xr-x 23 root  root 4096 Apr 29 06:28 ..
ubuntu@ip-10-0-1-139:/opt$ ls -lh
total 0
ubuntu@ip-10-0-1-139:/opt$ ls -alh
total 8.0K
drwxr-xr-x 2 ubuntu ubuntu 4.0K Apr  8 17:35 .
drwxr-xr-x 23 root  root 4.0K Apr 29 06:28 ..
ubuntu@ip-10-0-1-139:/opt$ exit
logout
Connection to 18.217.249.231 closed.
```

```
29/04/2020 12:40.31 /home/mobaxterm/MyDocuments
```

```

29/04/2020 12:40:31 /home/mobaxterm/MyDocuments > scp -i owncloud.pem ./owncloud.pem ubuntu@18.217.249.231:/opt/owncloud.pem
100% 1696 1.6KB/s 00:01

29/04/2020 12:42:00 /home/mobaxterm/MyDocuments > ssh -i owncloud.pem ubuntu@18.217.249.231
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1065-aws x86_64)

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

 System information as of Wed Apr 29 07:12:21 UTC 2020

 System load: 0.0 Processes: 87
 Usage of /: 13.9% of 7.69GB Users logged in: 0
 Memory usage: 15% IP address for eth0: 10.0.1.139
 Swap usage: 0%

0 packages can be updated.
0 updates are security updates.

Last login: Wed Apr 29 07:07:43 2020 from 196.207.102.129
ubuntu@ip-10-0-1-139:~$ 

```

Amazon EC2 Instances										
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
<input type="checkbox"/>	NAT-instance	i-008efc0d40d82ca48	t2.micro	us-east-2a	<span>●</span> running	<span>✓</span> 2/2 checks ...	None	<span>●</span>	52.14.36.26	-
<input type="checkbox"/>	owncloudapp-server	i-0e1398b354fc2f8de	t2.micro	us-east-2a	<span>●</span> running	<span>✓</span> 2/2 checks ...	None	<span>●</span>	18.217.249.231	-
<input checked="" type="checkbox"/>	owncloudDB-server	i-0fca585073377ad41	t2.micro	us-east-2b	<span>●</span> running	<span>✓</span> 2/2 checks ...	None	<span>●</span>	-	-

Instance: i-0fca585073377ad41 (owncloudDB-server) Private IP: 10.0.2.171	
Description	Status Checks
Instance ID	i-0fca585073377ad41
Instance state	running
Instance type	t2.micro
Finding	Opt-in to AWS Compute Optimizer for recommendations. <a href="#">Learn more</a>
Private DNS	ip-10-0-2-171.us-east-2.compute.internal
Private IPs	10.0.2.171
Secondary private IPs	
VPC ID	vpc-0cceeb1c4a39a20362 (owncloud-vpc)
Public DNS (IPv4)	-
IPv4 Public IP	-
IPv6 IPs	-
Elastic IPs	
Availability zone	us-east-2b
Security groups	allow-ssh-mysql, view inbound rules, view outbound rules
Scheduled events	No scheduled events
AMI ID	ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20200408/ami-07c1207a94d0b73b71

```

ubuntu@ip-10-0-1-139:~$ cd /opt/
ubuntu@ip-10-0-1-139:/opt$ ls -al
total 12
drwxr-xr-x  2 ubuntu  ubuntu  4096 Apr 29  07:11 .
drwxr-xr-x 23 root   root   4096 Apr 29  06:28 ..
-rw-----  1 ubuntu  ubuntu  1696 Apr 29  07:11 owncloud.pem
ubuntu@ip-10-0-1-139:/opt$ ssh -i owncloud.pem ubuntu@10.0.2.171
The authenticity of host '10.0.2.171 (10.0.2.171)' can't be established.
ECDSA key fingerprint is SHA256:UrV4lfGv+MxugkSS+tFfYSmw5wSjlyn0kZNaAYq+MAE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.171' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1065-aws x86_64)

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

 System information as of Wed Apr 29 07:13:58 UTC 2020

 System load:  0.0          Processes:      86
 Usage of /:   13.7% of 7.69GB   Users logged in:  0
 Memory usage: 15%           IP address for eth0: 10.0.2.171
 Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the

```

#Below command executed in owncloudDB-server

```

sudo apt-get update

sudo apt install mysql-server -y

sudo mysql_secure_installation

sudo mysql

CREATE DATABASE Employee;

use Employee;

create table Info (Empld INT, First_Name VARCHAR(20),Last_Name VARCHAR(20),Address
VARCHAR(20),City VARCHAR(15), Department VARCHAR(10),Salary INT);

insert into Info values ( "786" , "Ajeet" , "Parker" , "WestStreet" , "Las Vegas" , "Accounts" , 1000
);

CREATE USER 'Harry'@'localhost' IDENTIFIED BY 'Test$123';

GRANT ALL PRIVILEGES ON * . * TO 'Harry'@'localhost';

FLUSH PRIVILEGES;

exit

sudo service mysql status

```

```

mysql> use Employee;
Database changed
mysql> create table Info (EmpId INT, First_Name VARCHAR(20),Last_Name VARCHAR(20),Address VARCHAR(20),City VARCHAR(15), Department VARCHAR(10),Salary INT);
Query OK, 0 rows affected (0.06 sec)

mysql> insert into Info values ( "786" , "Ajeet" , "Parker" , "WestStreet" , "Las Vegas" , "Accounts" , 1000 );
Query OK, 1 row affected (0.01 sec)

mysql> CREATE USER 'Harry'@'localhost' IDENTIFIED BY 'Test$123';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON * . * TO 'Harry'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
ubuntu@ip-10-0-2-171:~$ sudo service mysql status
● mysql.service - MySQL Community Server
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2020-04-29 07:20:24 UTC; 3min 7s ago
    Main PID: 2590 (mysqld)
      Tasks: 29 (limit: 1152)
     CGroup: /system.slice/mysql.service
             └─2590 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

Apr 29 07:20:24 ip-10-0-2-171 systemd[1]: Starting MySQL Community Server...
Apr 29 07:20:24 ip-10-0-2-171 systemd[1]: Started MySQL Community Server.
ubuntu@ip-10-0-2-171:~$ █

```

## h- Configuration in public instance (owncloudapp-server)

```

ubuntu@ip-10-0-2-171:~$ exit
logout
Connection to 10.0.2.171 closed.
ubuntu@ip-10-0-1-139:/opt$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [928 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [318 kB]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [46.6 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [11.5 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1068 kB]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [332 kB]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [11.0 kB]
Get:16 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [4760 B]
Get:17 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [7516 B]
Get:18 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [4764 B]
Get:19 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [7060 B]
Get:20 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [4188 B]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [697 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [222 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [35.9 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [9188 B]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [657 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [218 kB]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [7176 B]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2764 B]
Fetched 18.6 MB in 4s (4855 kB/s)
Reading package lists... Done
ubuntu@ip-10-0-1-139:/opt$ █

```

sudo apt-get update

sudo apt-get install apache2

sudo service apache2 start

sudo service apache2 status



```
IP_ADDR=$(curl http://169.254.169.254/latest/meta-data/public-ipv4)
sudo chown ubuntu:ubuntu -R /var/www/html
sudo echo "This instance Having IP = $IP_ADDR" > /var/www/html/index.html
```



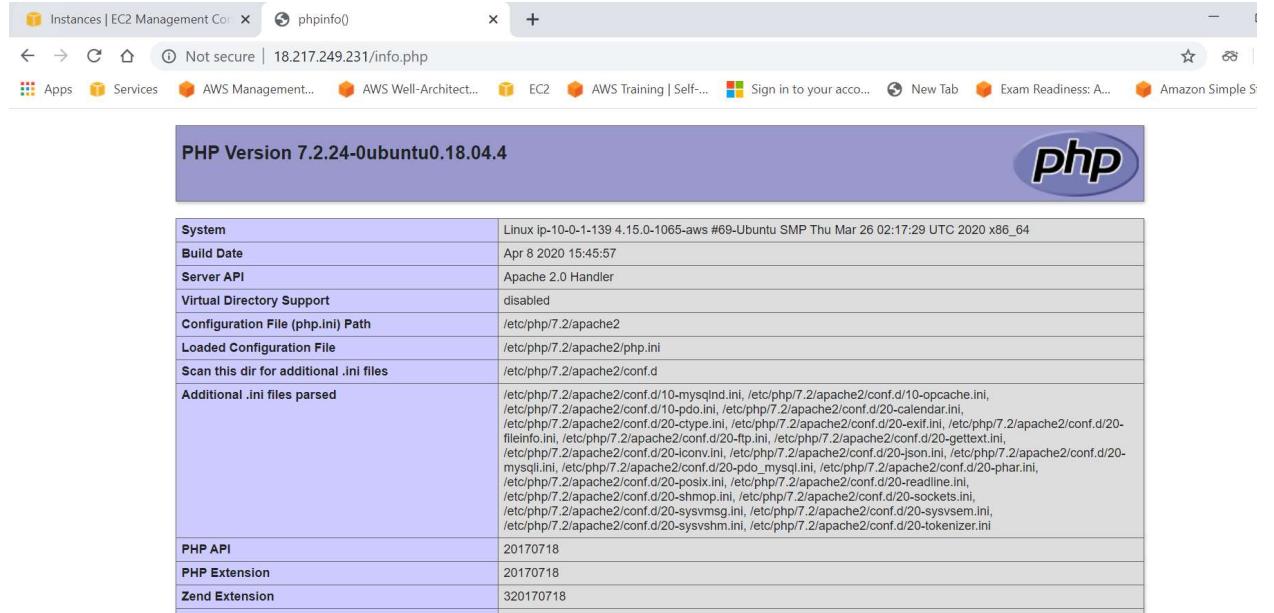
```
#install php server
sudo apt install php libapache2-mod-php php-mysql
#Make index.php as the default first load page
sudo nano /etc/apache2/mods-enabled/dir.conf
#Restart the web server
sudo systemctl restart apache2
sudo apt install php-cli
```

```
sudo nano /var/www/html/info.php
```

```
update-alternatives: using /usr/bin/phar7.2 to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar7.2 to provide /usr/bin/phar.phar (phar.phar) in auto mode

Creating config file /etc/php/7.2/cli/php.ini with new version
Setting up libapache2-mod-php7.2 (7.2.24-0ubuntu0.18.04.4) ...

Creating config file /etc/php/7.2/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php7.2
Setting up php-mysql (1:7.2+60ubuntu1) ...
Setting up libapache2-mod-php (1:7.2+60ubuntu1) ...
Setting up php7.2 (7.2.24-0ubuntu0.18.04.4) ...
Setting up php (1:7.2+60ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
ubuntu@ip-10-0-1-139:/opt$ sudo nano /etc/apache2/mods-enabled/dir.conf
ubuntu@ip-10-0-1-139:/opt$ sudo systemctl restart apache2
ubuntu@ip-10-0-1-139:/opt$ sudo apt install php-cli
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  php-cli
0 upgraded, 1 newly installed, 0 to remove and 15 not upgraded.
Need to get 3160 B of archives.
After this operation, 12.3 kB of additional disk space will be used.
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 php-cli all 1:7.2+60ubuntu1 [3160 B]
Fetched 3160 B in 0s (0 B/s)
Selecting previously unselected package php-cli.
(Reading database ... 57469 files and directories currently installed.)
Preparing to unpack .../php-cli_1%3a7.2+60ubuntu1_all.deb ...
Unpacking php-cli (1:7.2+60ubuntu1) ...
Setting up php-cli (1:7.2+60ubuntu1) ...
ubuntu@ip-10-0-1-139:/opt$ sudo nano /var/www/html/info.php
ubuntu@ip-10-0-1-139:/opt$
```



The screenshot shows a web browser window with the URL `18.217.249.231/info.php`. The page title is "Instances | EC2 Management Console". The page content displays the PHP Version 7.2.24-0ubuntu0.18.04.4 information table.

PHP Version 7.2.24-0ubuntu0.18.04.4	
System	Linux ip-10-0-1-139 4.15.0-1065-aws #69-Ubuntu SMP Thu Mar 26 02:17:29 UTC 2020 x86_64
Build Date	Apr 8 2020 15:45:57
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqli.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-filinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718

```
curl https://attic.owncloud.org/download/repositories/10.0/Ubuntu_18.04/Release.key | sudo apt-key add -
```

```
echo 'deb http://attic.owncloud.org/download/repositories/10.0/Ubuntu_18.04/' | sudo tee /etc/apt/sources.list.d/owncloud.list
```

```
sudo apt update
```

```

ubuntu@ip-10-0-1-139:/opt$ curl https://attic.owncloud.org/download/repositories/10.0/Ubuntu_18.04/Release.key | sudo apt-key add -
  % Total    % Received % Xferd  Average Speed   Time     Time      Current
                                           Dload  Upload Total Spent   Left Speed
100  4485  100  4485    0     0  6462      0 --:--:-- --:--:-- 6453
OK
ubuntu@ip-10-0-1-139:/opt$ echo 'deb http://attic.owncloud.org/download/repositories/10.0/Ubuntu_18.04/ /' | sudo tee /etc/apt/sources.list.d/owncloud.list
deb http://attic.owncloud.org/download/repositories/10.0/Ubuntu_18.04/ /
ubuntu@ip-10-0-1-139:/opt$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Ign:5 https://attic.owncloud.org/download/repositories/10.0/Ubuntu_18.04 InRelease
Get:6 https://attic.owncloud.org/download/repositories/10.0/Ubuntu_18.04 Release [984 B]
Get:7 https://attic.owncloud.org/download/repositories/10.0/Ubuntu_18.04 Release.gpg [481 B]
Get:8 https://attic.owncloud.org/download/repositories/10.0/Ubuntu_18.04 Packages [738 B]
Fetched 165 kB in 2s (68.6 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
15 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-10-0-1-139:/opt$ 

```

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

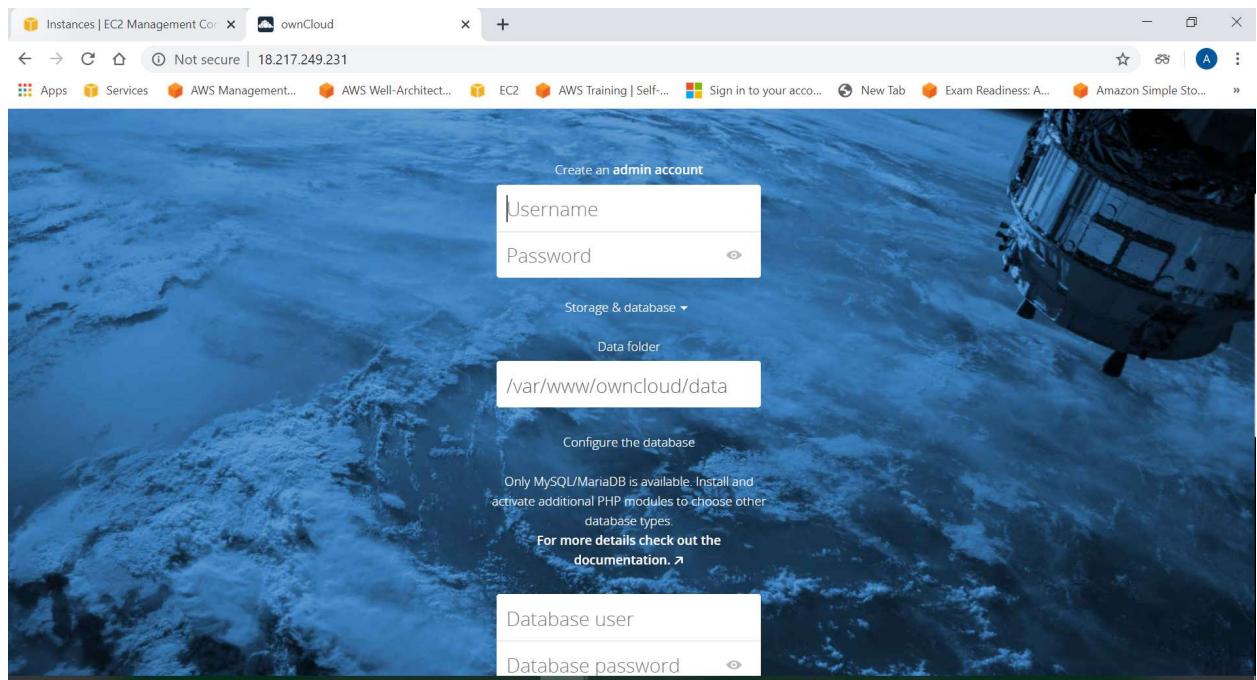
#Change default site directory to owncloud files directory using sudo user

sudo nano /etc/apache2/sites-enabled/000-default.conf

update directory root path to /var/www/owncloud

#restart the server

sudo systemctl reload apache2



3. **Lessons & Observations** -- In this project we successfully done the owncloud deployment and learn about NAT instance usage and LAMP stack.