

### Problem Statement:

Creating a Fileshare and sync solution using ownCloud and AWS Project: Own cloud Solution -

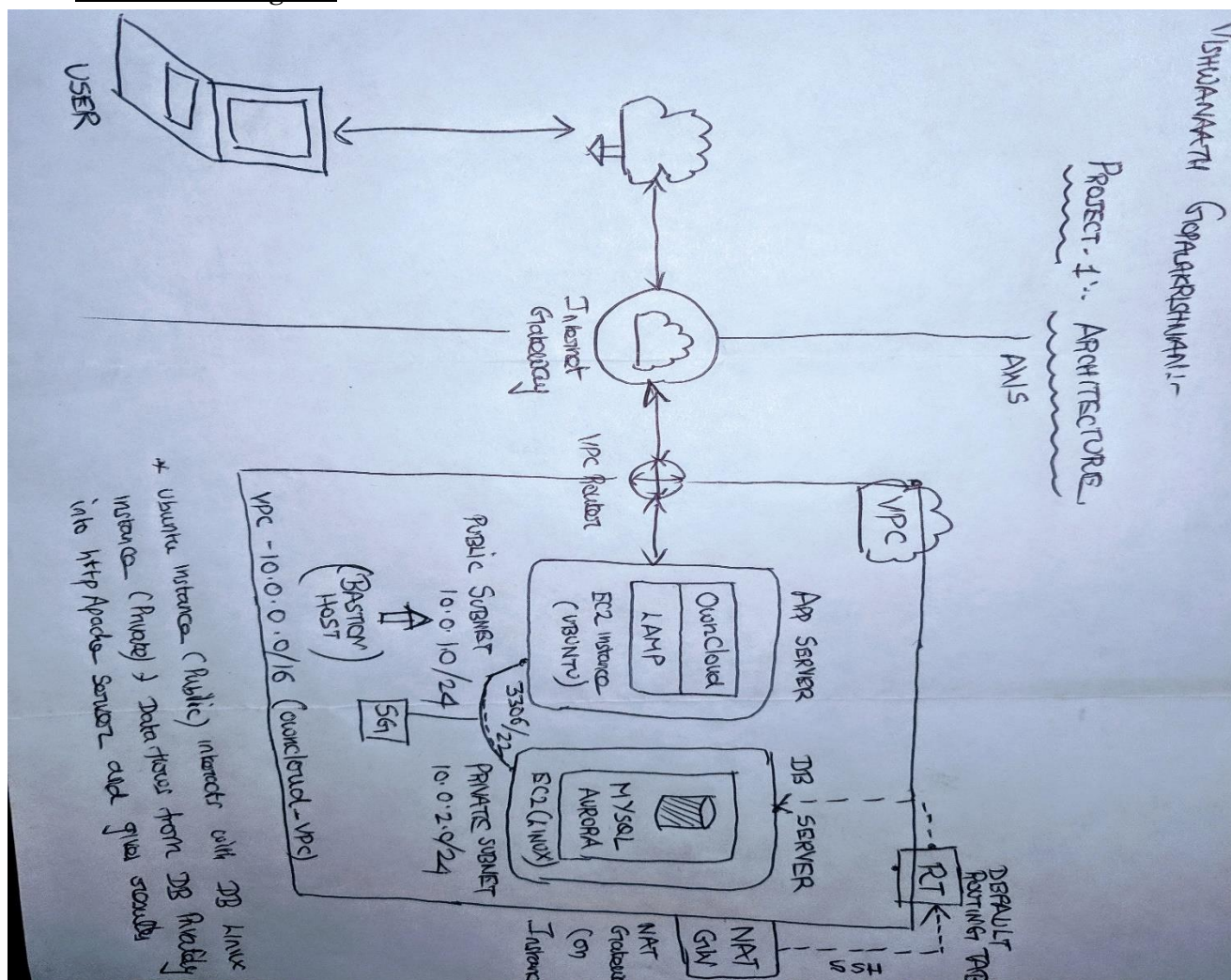
Vishwanaath Gopalakrishnan (18 Nov 2019)

Architecture and Screenshots listed in the document-

### Learnings & Observations:

- Private Subnet created to host the DB server instance
- NAT gateway used to establish connection to the Private subnet only from the Public subnet by setting up route tables and NAT gateway and letting it act as Bastion host

### Architecture Diagram:



### Implementation:

- Create a Custom VPC with the name owncloud-vpc with CIDR block 10.0.0.0/16

**VPC Dashboard**  
Filter by VPC:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table
owncloud-vpc	vpc-0b21730eb5c8b2bed	available	10.0.0.0/16	-	dopt-6aaa5810	rtb-05d4dca820dac00ea
def-vpc	vpc-0f603175	available	172.31.0.0/16	-	dopt-6aaa5810	rtb-463be738

**VPC: vpc-0b21730eb5c8b2bed**

Tags

Key	Value
Name	owncloud-vpc

- Create a Public Subnet with a CIDR block 10.0.1.0/24 that will allow 251 IP addresses under the owncloud-vpc. This is where the owncloud LAMP App server will be launched (AZ -1a)

**VPC Dashboard**  
Filter by VPC:

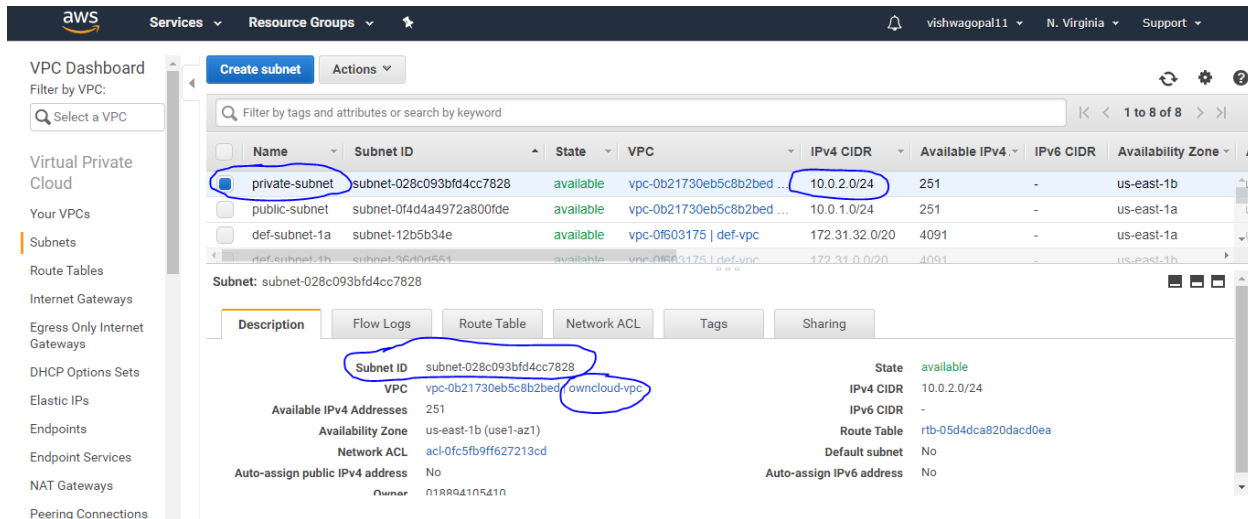
Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
public-subnet	subnet-0f4d4a4972a800fde	available	vpc-0b21730eb5c8b2bed ...	10.0.1.0/24	251	-	us-east-1a
def-subnet-1a	subnet-12b5b34e	available	vpc-0f603175   def-vpc	172.31.32.0/20	4091	-	us-east-1a
def-subnet-1b	subnet-36d0d551	available	vpc-0f603175   def-vpc	172.31.0.0/20	4091	-	us-east-1b
def-subnet-1c	subnet-4c070462	available	vpc-0f603175   def-vpc	172.31.80.0/20	4091	-	us-east-1c
def-subnet-1d	subnet-5b924616	available	vpc-0f603175   def-vpc	172.31.16.0/20	4091	-	us-east-1d
def-subnet-1e	subnet-ac25daa2	available	vpc-0f603175   def-vpc	172.31.64.0/20	4091	-	us-east-1e
def-subnet-1f	subnet-c4632bfa	available	vpc-0f603175   def-vpc	172.31.48.0/20	4091	-	us-east-1e

**Subnet: subnet-0f4d4a4972a800fde**

Description

Subnet ID: subnet-0f4d4a4972a800fde  
VPC: vpc-0b21730eb5c8b2bed | owncloud-vpc  
State: available  
IPv4 CIDR: 10.0.1.0/24

- Create a Private subnet with a CIDR block 10.0.2.0/24 that will allow 251 IP addresses also under the owncloud-vpc. This is where the MQ SQL Database server will be launched (AZ – 1b)



**VPC Dashboard**

Filter by VPC:

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

**Create subnet** Actions

Filter by tags and attributes or search by keyword

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
private-subnet	subnet-028c093bfd4cc7828	available	vpc-0b21730eb5c8b2bed	10.0.2.0/24	251	-	us-east-1b
public-subnet	subnet-0f4d4a972a800fde	available	vpc-0b21730eb5c8b2bed	10.0.1.0/24	251	-	us-east-1a
def-subnet-1a	subnet-125b5b34e	available	vpc-0f603175   def-vpc	172.31.32.0/20	4091	-	us-east-1a

Subnet: subnet-028c093bfd4cc7828

Description Flow Logs Route Table Network ACL Tags Sharing

Subnet ID: subnet-028c093bfd4cc7828

VPC: vpc-0b21730eb5c8b2bed | owncloud-vpc

Available IPv4 Addresses: 251

Availability Zone: us-east-1b (use1-az1)

Network ACL: acl-0fc5fb9ff627213cd

Auto-assign public IPv4 address: No

State: available

IPv4 CIDR: 10.0.2.0/24

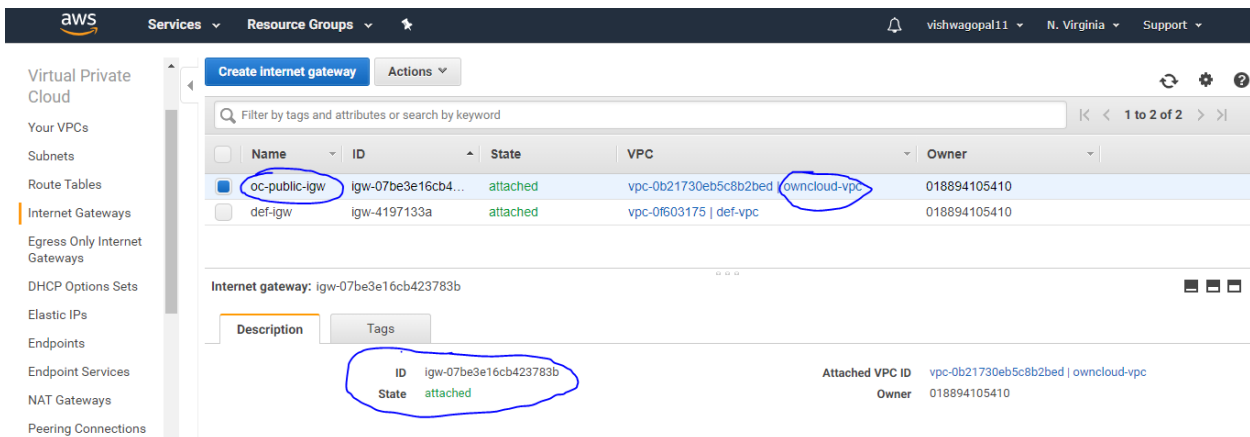
IPv6 CIDR: -

Route Table: rtb-05d4dca820dacd0ea

Default subnet: No

Auto-assign IPv6 address: No

- Connect the Public Subnet to the Internet using an Internet gateway.
- Create a Custom Internet gateway as oc-public-igw and associate it with the owncloud-vpc



**Create internet gateway** Actions

Filter by tags and attributes or search by keyword

Name	ID	State	VPC	Owner
oc-public-igw	igw-07be3e16cb423783b	attached	vpc-0b21730eb5c8b2bed   owncloud-vpc	018894105410
def-igw	igw-4197133a	attached	vpc-0f603175   def-vpc	018894105410

Internet gateway: igw-07be3e16cb423783b

Description Tags

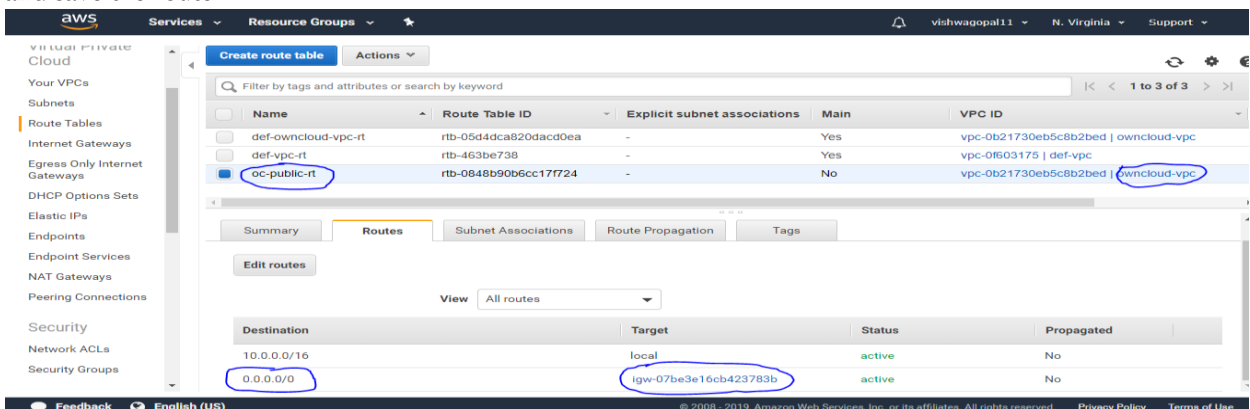
ID: igw-07be3e16cb423783b

State: attached

Attached VPC ID: vpc-0b21730eb5c8b2bed | owncloud-vpc

Owner: 018894105410

- Create a custom Route table oc-public-rt to connect the Public subnet in the VPC to the internet
- Add a global destination route CIDR block 0.0.0.0/0 and set the target to the Internet gateway oc-public-igw and save the route



**Create route table** Actions

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet associations	Main	VPC ID
def-owncloud-vpc-rt	rtb-05d4dca820dacd0ea	-	Yes	vpc-0b21730eb5c8b2bed   owncloud-vpc
def-vpc-rt	rtb-463be738	-	Yes	vpc-0f603175   def-vpc
oc-public-rt	rtb-0848b90b6cc17f724	-	No	vpc-0b21730eb5c8b2bed   owncloud-vpc

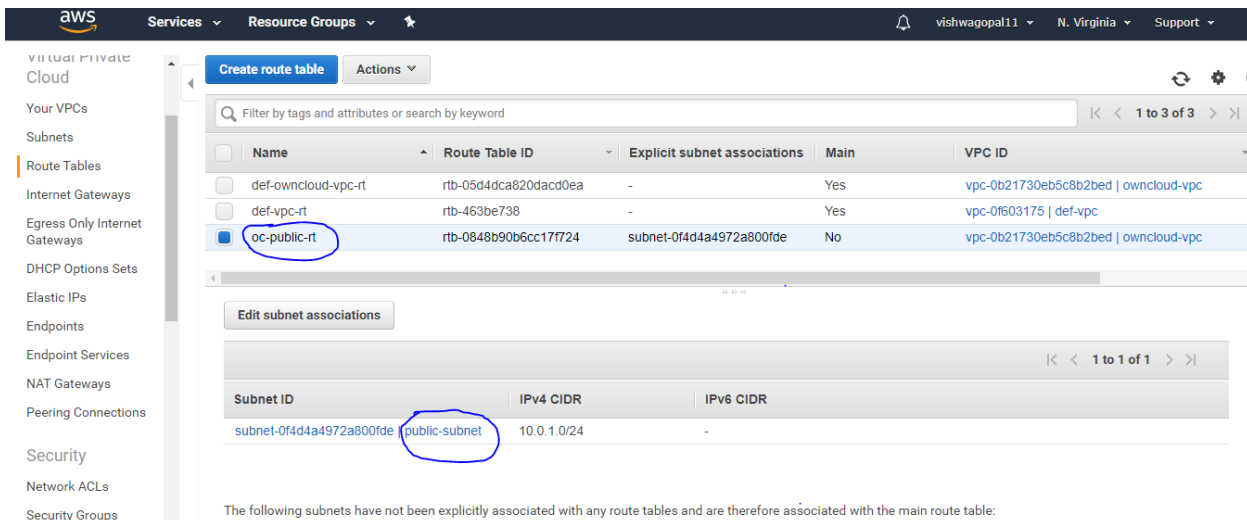
Summary Routes Subnet Associations Route Propagation Tags

Edit routes

View: All routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-07be3e16cb423783b	active	No

- Under Subnet associations, associate the route table created to the Public-subnet.



Virtual Private Cloud

Filter by tags and attributes or search by keyword

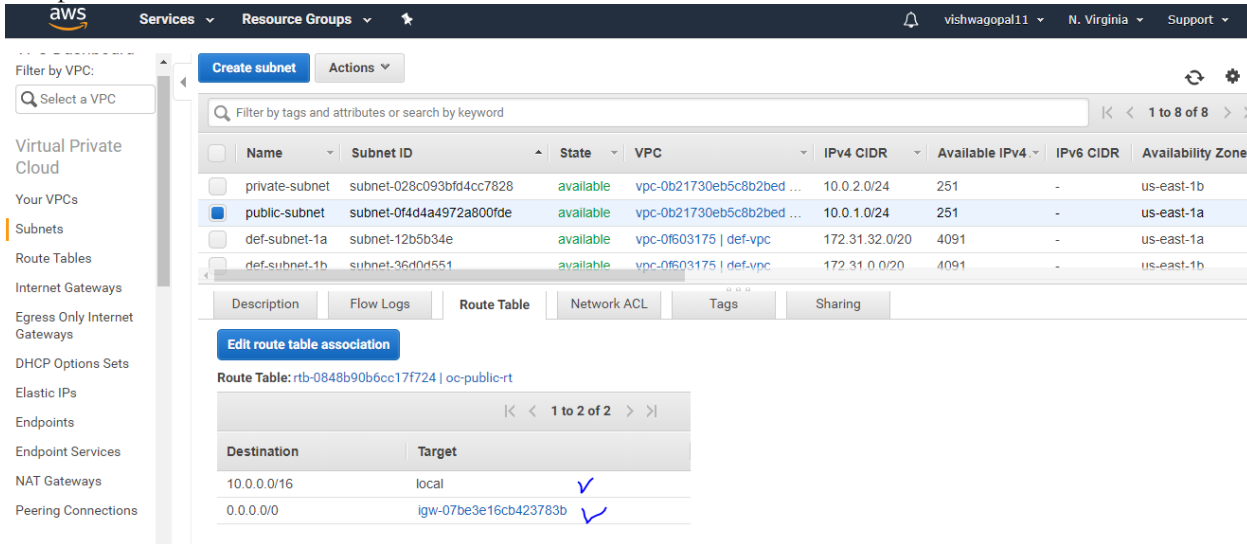
Name	Route Table ID	Explicit subnet associations	Main	VPC ID
def-owncloud-vpc-rt	rtb-05d4dca820dacd0ea	-	Yes	vpc-0b21730eb5c8b2bed   owncloud-vpc
def-vpc-rt	rtb-463be738	-	Yes	vpc-0f603175   def-vpc
oc-public-rt	rtb-0848b90b6cc17f724	subnet-0f4d4a4972a800fde	No	vpc-0b21730eb5c8b2bed   owncloud-vpc

Edit subnet associations

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-0f4d4a4972a800fde	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:

- The public subnet will now be associated with two route table entries as below



Virtual Private Cloud

Filter by tags and attributes or search by keyword

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
private-subnet	subnet-028c093bfd4cc7828	available	vpc-0b21730eb5c8b2bed ...	10.0.2.0/24	251	-	us-east-1b
public-subnet	subnet-0f4d4a4972a800fde	available	vpc-0b21730eb5c8b2bed ...	10.0.1.0/24	251	-	us-east-1a
def-subnet-1a	subnet-12b5b34e	available	vpc-0f603175   def-vpc	172.31.32.0/20	4091	-	us-east-1a
def-subnet-1b	subnet-36d0d551	available	vpc-0f603175   def-vpc	172.31.0.0/20	4091	-	us-east-1b

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

Edit route table association

Route Table: rtb-0848b90b6cc17f724 | oc-public-rt

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-07be3e16cb423783b


- Enable the auto assign IP settings for the Public subnet to allow the EC2 instance to get an IP address when launched

[Subnets](#) > Modify auto-assign IP settings

## Modify auto-assign IP settings

Enable the auto-assign IP address setting to automatically request a public IPv4 or IPv6 address for an instance launched in this subnet. You can override the auto-assign IP settings for an instance at launch time.

Subnet ID subnet-0f4d4a4972a800fde

Auto-assign IPv4 ☒ Enable auto-assign public IPv4 address 

\* Required

Cancel Save

The screenshot shows the AWS Management Console 'Create subnet' page. The 'public-subnet' is selected in the list. The 'Auto-assign public IPv4 address' checkbox is checked. The subnet details are as follows:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
public-subnet	subnet-0f4d4a4972a800fde	available	vpc-0b21730eb5c8b2bed	10.0.1.0/24	251	-	us-east-1a
private-subnet	subnet-028c093bfd4cc7828	available	vpc-0b21730eb5c8b2bed	10.0.2.0/24	251	-	us-east-1b
def-subnet-1f	subnet-c4632bfa	available	vpc-0f603175   def-vpc	172.31.48.0/20	4091	-	us-east-1e
def-subnet-1e	subnet-ac25daa2	available	vpc-0f603175   def-vpc	172.31.64.0/20	4091	-	us-east-1f

Subnet: subnet-0f4d4a4972a800fde

Property	Value
Subnet ID	subnet-0f4d4a4972a800fde
VPC	vpc-0b21730eb5c8b2bed   owncloud-vpc
Available IPv4 Addresses	251
Availability Zone	us-east-1a (use1-az6)
Network ACL	acl-0fc5fb9ff627213cd
Auto-assign public IPv4 address	Yes
Owner	018894105410
State	available
IPv4 CIDR	10.0.1.0/24
IPv6 CIDR	-
Route Table	rtb-0848b90b6cc17f724   oc-public-rt
Default subnet	No
Auto-assign IPv6 address	No

- Create EC2 instance in the Public subnet and proceed with the steps to setup the owncloud and LAMP applications in the EC2 instance as per the 7-step workflow

The screenshot shows the AWS Management Console 'Step 7: Review Instance Launch' page. The instance type is t2.micro, and the security group is oc-combo-sg. The instance details are as follows:

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	

- Setup security group to port 22 and port 28 as oc-combo-sg
- Ensure that the EC2 instance is associated with the correct VPC and subnet as below

The screenshot shows the AWS Management Console 'Step 3: Configure Instance Details' page. The network is vpc-0b21730eb5c8b2bed, and the subnet is subnet-0f4d4a4972a800fde. The instance details are as follows:

Property	Value
Number of instances	1
Purchasing option	Request Spot instances
Network	vpc-0b21730eb5c8b2bed   owncloud-vpc
Subnet	subnet-0f4d4a4972a800fde   public-subnet   us-east-1a
Auto-assign Public IP	Use subnet setting (Enable)
Placement group	Add instance to placement group
Capacity Reservation	Open
IAM role	None

- EC2 – Launch – 7 Step workflow - review

**aws** Services Resource Groups vishwagopal11 N. Virginia Support

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

### Step 7: Review Instance Launch

▼ Instance Details [Edit instance details](#)

Number of instances: 1 Purchasing option: On demand

Network: vpc-0b21730eb5c8b2bed  
Subnet: subnet-0f4d4a4972a800fde  
EBS-optimized: No  
Monitoring: No  
Termination protection: No  
Shutdown behavior: Stop  
Stop - Hibernate behavior: Disabled  
Capacity Reservation: open  
IAM role: None  
Tenancy: default  
T2/T3 Unlimited: Disabled  
Host ID: Off  
Affinity: Off  
Kernel ID: Use default  
RAM disk ID: Use default  
User data: Use subnet setting (Enable)  
Assign Public IP: Use subnet setting (Enable)  
Assign IPv6 IP: Use subnet setting (Enable)  
Network interfaces

Device	Network interface	Subnet	Primary IP	Secondary IP Addresses
eth0	New network interface	subnet-0f4d4a4972a800fde	Auto-assign	

[Cancel](#) [Previous](#) [Launch](#)

**aws** Services Resource Groups vishwagopal11 N. Virginia Support

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

### Step 7: Review Instance Launch

Affinity: Off  
Kernel ID: Use default  
RAM disk ID: Use default  
User data: Use subnet setting (Enable)  
Assign Public IP: Use subnet setting (Enable)  
Assign IPv6 IP: Use subnet setting (Enable)  
Network interfaces

Device	Network interface	Subnet	Primary IP	Secondary IP Addresses
eth0	New network interface	subnet-0f4d4a4972a800fde	Auto-assign	

▼ Storage [Edit storage](#)

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MiB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-02e105f83f77cd927	8	gp2	100 / 3000	N/A	Yes	Not Encrypted

▼ Tags [Edit tags](#)

Key	Value	Instances	Volumes
Name	http-appserver	✓	✓

[Cancel](#) [Previous](#) [Launch](#)

- Launch the EC2 instance after creating a keypair as owncloud.pem and download the keypair file to the system and ensure that the EC2 instance is created in the public subnet's IP range



The screenshot shows the AWS Management Console interface. On the left, the navigation menu includes EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, IMAGES, AMIs, Bundle Tasks, and ELASTIC BLOCK STORE. The main content area displays a table of EC2 instances. The instance 'http-appserver' is highlighted, showing its details in a 'Description' tab. The instance is in a 'pending' state, located in the 'us-east-1a' availability zone. Its public IP address is 52.91.36.185. The instance type is 't2.micro' and it is using the 'ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20191002' AMI. The key pair name is 'owncloud'.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
http-appserver	i-0b0ab4aabb2cdb81d	t2.micro	us-east-1a	pending	Initializing	None	52.91.36.185	52.91.36.185

**Description**

Instance ID	Instance state	Instance type	Elastic IPs	Availability zone	Security groups	Scheduled events	AMI ID	Platform	IAM role	Key pair name	Owner	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Private DNS	Private IPs	Secondary private IPs	VPC ID	Subnet ID	Network interfaces	Source/dest. check	T2/T3 Unlimited	EBS-optimized
i-0b0ab4aabb2cdb81d	pending	t2.micro	-	us-east-1a	oc-combo-sg. view inbound rules. view outbound rules	-	ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20191002 (ami-04b9e2b5572fa0d1)	-	-	owncloud	Owner	52.91.36.185	52.91.36.185	-	ip-10-0-1-163.ec2.internal	10.0.1.163	-	vpc-0b21730eb5c8b2bed (owncloud-vpc)	subnet-0f4d4a4972a800fde (public-subnet)	eth0	True	Disabled	False

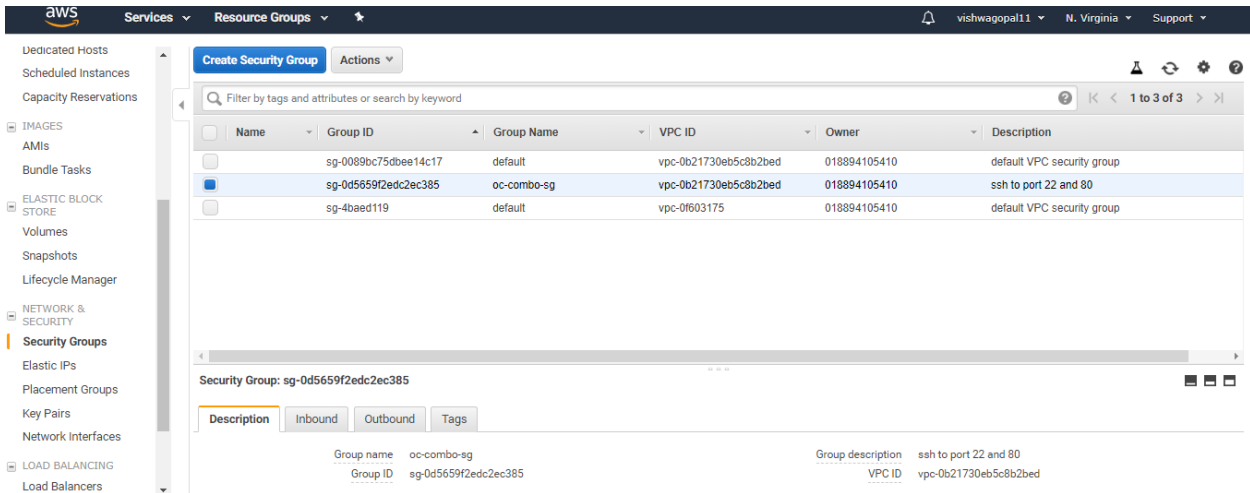
- SSH into the EC2 instance using the owncloud.pem file using the IPV4 public IP address 52.91.36.185 after ensuring the security group is setup for both ports 22 and 80

The screenshot shows the AWS Management Console interface. The instance 'http-appserver' is now in a 'running' state. The public IP address 52.91.36.185 is circled in blue. The instance has passed 2/2 status checks. The key pair name 'owncloud' is also circled in blue.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
http-appserver	i-0b0ab4aabb2cdb81d	t2.micro	us-east-1a	running	2/2 checks ...	None	52.91.36.185	52.91.36.185

**Description**

Instance ID	Instance state	Instance type	Elastic IPs	Availability zone	Security groups	Scheduled events	AMI ID	Platform	IAM role	Key pair name	Owner	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Private DNS	Private IPs	Secondary private IPs	VPC ID	Subnet ID	Network interfaces	Source/dest. check	T2/T3 Unlimited	EBS-optimized
i-0b0ab4aabb2cdb81d	running	t2.micro	-	us-east-1a	oc-combo-sg. view inbound rules. view outbound rules	No scheduled events	ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20191002 (ami-04b9e2b5572fa0d1)	-	-	owncloud	Owner	52.91.36.185	52.91.36.185	-	ip-10-0-1-163.ec2.internal	10.0.1.163	-	vpc-0b21730eb5c8b2bed (owncloud-vpc)	subnet-0f4d4a4972a800fde (public-subnet)	eth0	True	Disabled	False



- Once the security groups are created with appropriate ports, the console can be used to get into the instance

## Logging into the console

- SSHed into the instance and executed the below steps
- Install apache web server using following commands
  1. `sudo apt-get update`
  2. `sudo apt-get install apache2`

```
ubuntu@DESKTOP-49DC5BB:/mnt/c/users/vishw/downloads$ sudo su
[sudo] password for ubuntu:
root@DESKTOP-49DC5BB:/mnt/c/Users/Vishw/Downloads# ssh -i owncloud.pem ubuntu@52.91.36.185
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1051-aws x86_64)

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

System information as of Mon Nov 18 03:35:04 UTC 2019

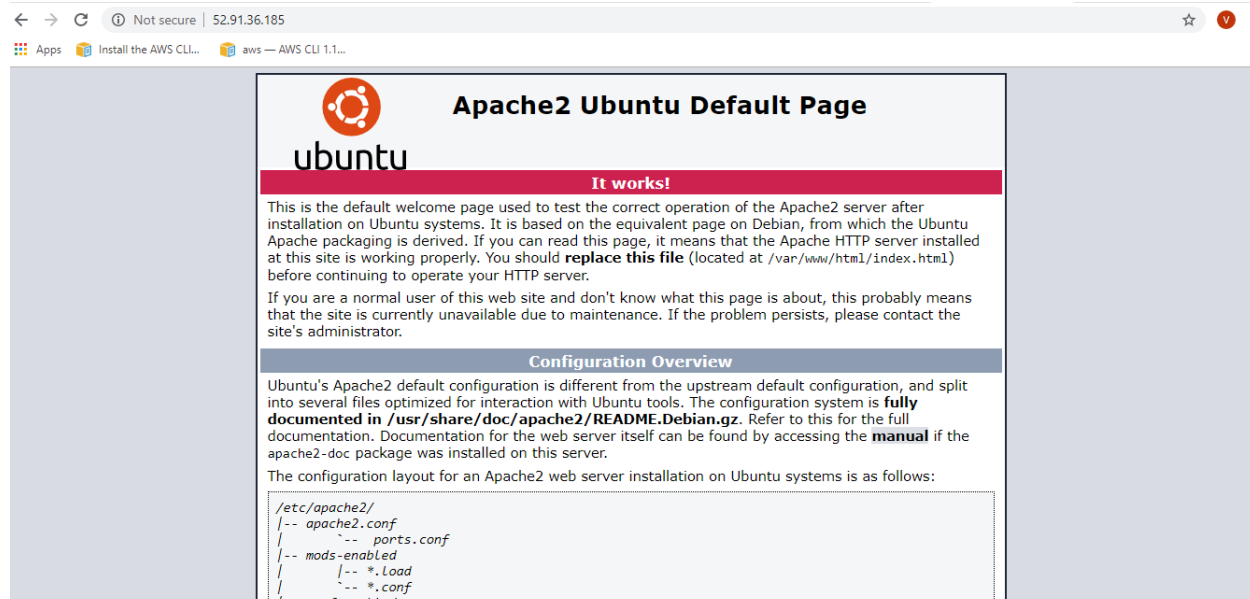
System load:  0.0               Processes:    88
Usage of /:   15.8% of 7.69GB   Users logged in:  0
Memory usage: 16%              IP address for eth0: 10.0.1.163
Swap usage:   0%

61 packages can be updated.
24 updates are security updates.

Last login: Mon Nov 18 03:15:09 2019 from 73.61.110.39
ubuntu@ip-10-0-1-163:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Fetched 88.7 kB in 0s (240 kB/s)
Reading package lists... Done
ubuntu@ip-10-0-1-163:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.29-1ubuntu4.11).
0 upgraded, 0 newly installed, 0 to remove and 47 not upgraded.
ubuntu@ip-10-0-1-163:~$
```



- Validate installation by accessing public ip of EC2 instance in browser



- Use the following commands to install php - `sudo apt install php libapache2-mod-php php-mysql`
- Make index.php as the default first load page 1. Edit `/etc/apache2/mods-enabled/dir.conf` file and make index.php as first access page `DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm` 2. Restart the web server - `sudo systemctl restart apache2`

```
ubuntu@ip-10-0-1-163:~$ sudo apt install php libapache2-mod-php  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libapache2-mod-php7.2 libsodium23 php-common php7.2 php7.2-cli php7.2-common php7.2-json php7.2-opcache php7.2-readline  
Suggested packages:  
  php-pear  
The following NEW packages will be installed:  
  libapache2-mod-php libapache2-mod-php7.2 libsodium23 php php-common php7.2 php7.2-cli php7.2-common php7.2-json php7.2-opcache php7.2-readline  
0 upgraded, 11 newly installed, 0 to remove and 47 not upgraded.  
Need to get 4013 kB of archives.  
After this operation, 17.6 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Abort.  
ubuntu@ip-10-0-1-163:~$
```

- Edit `/etc/apache2/mods-enabled/dir.conf` file and make index.php as first access page `DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm`

```

ubuntu@ip-10-0-1-163:/$ cat /etc/apache2/mods-enabled/dir.conf file
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>


# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
cat: file: No such file or directory
ubuntu@ip-10-0-1-163:/$ vi /etc/apache2/mods-enabled/dir.conf file
2 files to edit
ubuntu@ip-10-0-1-163:/$ ls -ltr /etc/apache2/mods-enabled/dir.conf file
ls: cannot access 'file': No such file or directory
lrwxrwxrwx 1 root root 26 Nov 18 03:20 /etc/apache2/mods-enabled/dir.conf -> ../mods-available/dir.conf
ubuntu@ip-10-0-1-163:/$ sudo su
root@ip-10-0-1-163:/# cat /etc/apache2/mods-enabled/dir.conf file
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
cat: file: No such file or directory
root@ip-10-0-1-163:/# vi cat /etc/apache2/mods-enabled/dir.conf file
3 files to edit
root@ip-10-0-1-163:/# vi /etc/apache2/mods-enabled/dir.conf file
2 files to edit
root@ip-10-0-1-163:/# cat /etc/apache2/mods-enabled/dir.conf file
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
cat: file: No such file or directory
root@ip-10-0-1-163:/#

```

- Restart using sudo systemctl restart apache2

 Select root@ip-10-0-1-163: /

```

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

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
cat: file: No such file or directory
root@ip-10-0-1-163:/# sudo systemctl restart apache2
root@ip-10-0-1-163:/# pwd
/
root@ip-10-0-1-163:/#

```

- Install owncloud on ubuntu 18.04 using the commands provided

1. curl https://download.owncloud.org/download/repositories/10.2/Ubuntu\_18.04/Release.key | sudo apt-key add -
2. echo 'deb http://download.owncloud.org/download/repositories/10.2/Ubuntu\_18.04/ /' | sudo tee /etc/apt/sources.list.d/owncloud.list
3. sudo apt update 4. sudo apt install php-bz2 php-curl php-gd php-imagick php-intl php-mbstring php-xml php-zip owncloud-files

```

root@ip-10-0-1-163: /
Setting up php7.2-xml (7.2.24-0ubuntu0.18.04.1) ...
Creating config file /etc/php/7.2/mods-available/dom.ini with new version
Creating config file /etc/php/7.2/mods-available/simplexml.ini with new version
Creating config file /etc/php/7.2/mods-available/wddx.ini with new version
Creating config file /etc/php/7.2/mods-available/xml.ini with new version
Creating config file /etc/php/7.2/mods-available/xmlreader.ini with new version
Creating config file /etc/php/7.2/mods-available/xmlwriter.ini with new version
Creating config file /etc/php/7.2/mods-available/xsl.ini with new version
Setting up php7.2-zip (7.2.24-0ubuntu0.18.04.1) ...
Creating config file /etc/php/7.2/mods-available/zip.ini with new version
Setting up php-bz2 (1:7.2+60ubuntu1) ...
Setting up libmagickcore-6.q16-3:amd64 (8:6.9.7.4+dfsg-16ubuntu6.8) ...
Setting up libgd3:amd64 (2.2.5-4ubuntu0.3) ...
Setting up php-zip (1:7.2+60ubuntu1) ...
Setting up php-xml (1:7.2+60ubuntu1) ...
Setting up php7.2-cli (7.2.24-0ubuntu0.18.04.1) ...
update-alternatives: using /usr/bin/php7.2 to provide /usr/bin/php (php) in auto mode
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 php7.2-phpdbg (7.2.24-0ubuntu0.18.04.1) ...
update-alternatives: using /usr/bin/phpdbg7.2 to provide /usr/bin/phpdbg (phpdbg) in auto mode

Creating config file /etc/php/7.2/phpdbg/php.ini with new version
Setting up libmagickwand-6.q16-3:amd64 (8:6.9.7.4+dfsg-16ubuntu6.8) ...
Setting up libavahi-client3:amd64 (0.7-3.1ubuntu1.2) ...
Setting up libcups2:amd64 (2.2.7-1ubuntu2.7) ...
Setting up php7.2-gd (7.2.24-0ubuntu0.18.04.1) ...

Creating config file /etc/php/7.2/mods-available/gd.ini with new version
Setting up php-imagick (3.4.3~rc2-2ubuntu4) ...
Setting up libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) ...
Setting up libcupsimage2:amd64 (2.2.7-1ubuntu2.7) ...
Setting up php-gd (1:7.2+60ubuntu1) ...
Setting up libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.12) ...
Setting up ghostscript (9.26~dfsg+0-0ubuntu0.18.04.12) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
root@ip-10-0-1-163: /#

```

- Accessing the IPV4 address 52.91.36.185 through address bar gives the required results as below

```

← → ↻ ⓘ Not secure | 52.91.36.185
Apps Install the AWS CLI... aws — AWS CLI 1.1...

<?php
/**
 * @author Jörn Friedrich Dreyer <jfd@butonic.de>
 * @author Lukas Reschke <lukas@statuscode.ch>
 * @author Morris Jobke <hey@morrisjobke.de>
 * @author Philipp Schaffrath <github@philippschaffrath.de>
 * @author RealRancor <fisch.666@gmx.de>
 * @author Robin Appelman <icewind@owncloud.com>
 * @author Sergio Bertolán <sbertolin@solidgear.es>
 * @author Thomas Müller <thomas.mueller@tmit.eu>
 * @author Vincent Petry <pvince81@owncloud.com>
 *
 * @copyright Copyright (c) 2018, ownCloud GmbH
 * @license AGPL-3.0
 *
 * This code is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License, version 3,
 * as published by the Free Software Foundation.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License, version 3,
 * along with this program. If not, see <http://www.gnu.org/licenses/>
 */

// Show warning if a PHP version below 7.0.7 is used, this has to happen here
// because base.php will already use 7.0 syntax.
if (\version_compare(PHP_VERSION, '7.0.7') === -1) {
    echo 'This version of ownCloud requires at least PHP 7.0.7<br/>';
    echo 'You are currently running PHP ' . PHP_VERSION . '. Please update your PHP version.';
    return;
}

```

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

1. edit /etc/apache2/sites-enabled/000-default.conf
2. update directory root path to /var/www/owncloud
3. restart the server - sudo systemctl reload apache2

```

← → ↻ ⓘ Not secure | 52.91.36.185
Apps Install the AWS CLI... aws — AWS CLI 1.1...

<?php
/**
 * @author Jörn Friedrich Dreyer <jfd@butonic.de>
 * @author Lukas Reschke <lukas@statuscode.ch>
 * @author Morris Jobke <hey@morrisjobke.de>
 * @author Philipp Schaffrath <github@philippschaffrath.de>
 * @author RealRancor <fisch.666@gmx.de>
 * @author Robin Appelman <icewind@owncloud.com>
 * @author Sergio Bertolán <sbertolin@solidgear.es>
 * @author Thomas Müller <thomas.mueller@tmit.eu>
 * @author Vincent Petry <pvince81@owncloud.com>
 *
 * @copyright Copyright (c) 2018, ownCloud GmbH
 * @license AGPL-3.0
 *
 * This code is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License, version 3,
 * as published by the Free Software Foundation.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License, version 3,
 * along with this program. If not, see <http://www.gnu.org/licenses/>
 */

// Show warning if a PHP version below 7.0.7 is used, this has to happen here
// because base.php will already use 7.0 syntax.
if (\version_compare(PHP_VERSION, '7.0.7') === -1) {
    echo 'This version of ownCloud requires at least PHP 7.0.7<br/>';
    echo 'You are currently running PHP ' . PHP_VERSION . '. Please update your PHP version.';
    return;
}

// Show warning if PHP 7.3 is used as ownCloud is not compatible with PHP 7.3

```

## DB Server Setup in EC2:

Setting up DB server on the Private Subnet and connecting via NAT gateway to allow interaction between the appserver and DB server

**aws** Services Resource Groups

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

### Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

▼ **AMI Details** [Edit AMI](#)

**Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-00eb20669e0990cb4** ✓

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root Device Type: ebs Virtualization type: hvm

▼ **Instance Type** [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

▼ **Security Groups** [Edit security groups](#)

**Security group name** public-subnet-traffic-only  
**Description** receive traffic from public subnet

- DB server – Private subnet – Linux instance review pages

**aws** Services Resource Groups

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

### Step 7: Review Instance Launch

▼ **Security Groups** [Edit security groups](#)

**Security group name** public-subnet-traffic-only  
**Description** receive traffic from public subnet

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	10.0.1.0/24	
MySQL/Aurora	TCP	3306	10.0.1.0/24	
All ICMP - IPv4	All	N/A	10.0.1.0/24	

▼ **Instance Details** [Edit instance details](#)

**Number of instances** 1 **Purchasing option** On demand

**Network** vpc-0b21730eb5c8b2bed ✓  
**Subnet** subnet-028c093b4d4cc7828 ✓  
**EBS-optimized** No  
**Monitoring** No  
**Termination protection** No

**Cancel** **Previous** **Launch**

**aws** Services Resource Groups

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

### Step 7: Review Instance Launch

▼ **Instance Details** [Edit instance details](#)

**Number of instances** 1 **Purchasing option** On demand

**Network** vpc-0b21730eb5c8b2bed  
**Subnet** subnet-028c093b4d4cc7828  
**EBS-optimized** No  
**Monitoring** No  
**Termination protection** No  
**Shutdown behavior** Stop  
**Stop - Hibernate behavior** Disabled  
**Capacity Reservation** open  
**IAM role** None  
**Tenancy** default  
**T2/T3 Unlimited** Disabled  
**Host ID** Off  
**Affinity** Off  
**Kernel ID** Use default  
**RAM disk ID** Use default  
**User data** Use default  
**Assign Public IP** Use subnet setting (Disable)  
**Assign IPv6 IP** Use subnet setting (Disable)  
**Network interfaces**

**Cancel** **Previous** **Launch**

## Step 7: Review Instance Launch

[RAW DISK I/O](#) [USE DEFAULT](#)  
[User data](#)  
[Assign Public IP](#) [Use subnet setting \(Disable\)](#)  
[Assign IPv6 IP](#) [Use subnet setting \(Disable\)](#)  
[Network interfaces](#)

Device	Network Interface	Subnet	Primary IP	Secondary IP Addresses
eth0	New network interface	subnet-028c093bfd4cc7828	Auto-assign	

▼ Storage [Edit storage](#)

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encrypted ⓘ
Root	/dev/xvda	snap-05a19c3561abd794a	8	gp2	100 / 3000	N/A	Yes	Not Encrypted

▼ Tags [Edit tags](#)

Key	Value	Instances ⓘ	Volumes ⓘ
Name	db-server	✓	✓

[Cancel](#) [Previous](#) [Launch](#)

- Key pair selection in the EC2 instance creation steps

Step 7: Review Instance Launch

[RAW DISK I/O](#) [USE DEFAULT](#)  
[User data](#)  
[Assign Public IP](#) [Use subnet setting](#)  
[Assign IPv6 IP](#) [Use subnet setting](#)  
[Network interfaces](#)

Device	Network Interface
eth0	New network interface

Storage

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ
Root	/dev/xvda	snap-05a19c3561abd794a

Tags

Key	Value	Instances ⓘ	Volumes ⓘ
Name	db-server	✓	✓

**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](#).

Choose an existing key pair

Select a key pair

owncloud

☒ I acknowledge that I have access to the selected private key file (owncloud.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) [Launch Instances](#)



The screenshot shows the AWS Management Console interface. On the left, the navigation menu includes EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, and IMAGES. The 'INSTANCES' section is expanded, showing a list of instances. Two instances are listed: 'http-appserver' and 'db-server'. The 'db-server' instance is highlighted with a blue circle. Below the list, the details for the 'db-server' instance are displayed. Key details include: Instance ID (i-074b72bb8d970d580), Instance Type (t2.micro), Availability Zone (us-east-1b), Instance State (running), Status Checks (2/2 checks passed), Alarm Status (None), Public DNS (IPv4) (52.91.36.185), Private DNS (ip-10-0-2-72.ec2.internal), Private IPs (10.0.2.72), VPC ID (vpc-0b21730eb5c8b2bed), Subnet ID (subnet-028c093bf64cc7828), Network interfaces (eth0), Source/dest. check (True), T2/T3 Unlimited (Disabled), EBS-optimized (False), Root device type (ebs), and Key pair name (owncloud). The 'owncloud' key pair name is circled in blue. The bottom of the console shows the footer with '© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.' and links to Privacy Policy and Terms of Use.

- The DB server is now created and in order to interact with the app server internet private connection is required
- This will need to be setup via NAT gateway by connecting to Public subnet that acts as the Bastion Host
- To SSH into the Private subnet DB instance – The .pem file is required.

```
ec2-user@ip-10-0-2-72:~
51 packages can be updated.
24 updates are security updates.

Last login: Mon Nov 18 03:35:05 2019 from 73.61.110.39
ubuntu@ip-10-0-1-163:~$
ubuntu@ip-10-0-1-163:~$
ubuntu@ip-10-0-1-163:~$ cd /opt/
ubuntu@ip-10-0-1-163:/opt$ sudo chown ubuntu:ubuntu -R /opt
ubuntu@ip-10-0-1-163:/opt$ ls -al
total 8
drwxr-xr-x 2 ubuntu ubuntu 4096 Oct  2 17:08 .
drwxr-xr-x 23 root     root   4096 Nov 18 06:01 ..
ubuntu@ip-10-0-1-163:/opt$ exit
logout
Connection to 52.91.36.185 closed.
root@DESKTOP-49DC588:/mnt/c/Users/Vishw/Downloads# ls -al *.pem
ls: cannot access '-': No such file or directory
ls: cannot access 'al': No such file or directory
-rw-r--r-- 1 root root 1700 Nov 18 06:01 gl.pem
-rw-r--r-- 1 root root 1700 Nov 18 06:01 glpem1 (1).pem
-rw-r--r-- 1 root root 1700 Nov 18 06:01 glpem1 (2).pem
-rw-r--r-- 1 root root 1700 Nov 18 06:01 greatlearning.pem
-rw-r--r-- 1 root root 1700 Nov 18 06:01 owncloud.pem
root@DESKTOP-49DC588:/mnt/c/Users/Vishw/Downloads# scp -i owncloud.pem ./owncloud.pem ubuntu@52.91.36.185:/opt
owncloud.pem
root@DESKTOP-49DC588:/mnt/c/Users/Vishw/Downloads#
root@DESKTOP-49DC588:/mnt/c/Users/Vishw/Downloads# ssh -i owncloud.pem ubuntu@52.91.36.185
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1051-aws x86_64)

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

System information as of Mon Nov 18 06:05:53 UTC 2019

System load:  0.01          Processes:      94
Usage of /:   21.5% of 7.69GB Users logged in: 0
Memory usage: 20%          IP address for eth0: 10.0.1.163
Swap usage:   0%

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

ec2-user@ip-10-0-2-72:~

```

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

*** System restart required ***
Last login: Mon Nov 18 05:59:32 2019 from 73.61.110.39
ubuntu@ip-10-0-1-163:~$ cd /opt
ubuntu@ip-10-0-1-163:/opt$ ls -l *.pem
-r-xr-xr-x 1 ubuntu ubuntu 1696 Nov 18 06:05 owncloud.pem
ubuntu@ip-10-0-1-163:/opt$ ssh -i owncloud.pem ec2-user@10.0.2.72
The authenticity of host '10.0.2.72 (10.0.2.72)' can't be established.
ECDSA key fingerprint is SHA256:UsS6978xtSvBzM2V6lzAE/sdONBoOih8RB9k8/LP5sM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.72' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!                @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0555 for 'owncloud.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "owncloud.pem": bad permissions
ec2-user@10.0.2.72: Permission denied (publickey).
ubuntu@ip-10-0-1-163:/opt$ sudo su
root@ip-10-0-1-163:/opt# ssh -i owncloud.pem ec2-user@10.0.2.72
The authenticity of host '10.0.2.72 (10.0.2.72)' can't be established.
ECDSA key fingerprint is SHA256:UsS6978xtSvBzM2V6lzAE/sdONBoOih8RB9k8/LP5sM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.72' (ECDSA) to the list of known hosts.

  _ | _ | _ )
  _ | ( _ | /
  _ | \ _ | _ |
                Amazon Linux AMI
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-10-0-2-72 ~]$

```

EC2 instances are accessed via ssh as per the screenshot

- As a first step we copy the owncloud.pem file from the ubuntu public subnet EC2 instance into an opt folder where we change the owner of the file after which we use the scp command to copy the file into the EC2 instance and then ssh into the private DB instance using the Private IP address
- Public EC2 instance – Ubuntu (Username: Ubuntu)
- Private EC2 instance – Linux (Username: ubuntu)
- The pem file can be SSHed using the super user role from either of the instances but it does not connect to the internet

```
[ec2-user@ip-10-0-2-72 ~]$ sudo yum update
Failed to set locale, defaulting to C
Loaded plugins: priorities, update-motd, upgrade-helper
Could not retrieve mirrorlist http://repo.us-east-1.amazonaws.com/latest/main/mirror.list error was
12: Timeout on http://repo.us-east-1.amazonaws.com/latest/main/mirror.list: (28, 'Connection timed out after 5001 milliseconds')
```

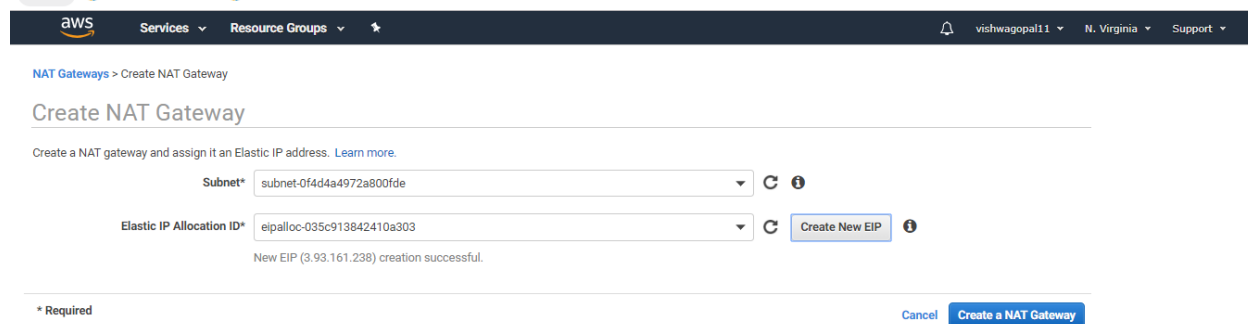
One of the configured repositories failed (Unknown),  
and yum doesn't have enough cached data to continue. At this point the only  
safe thing yum can do is fail. There are a few ways to work "fix" this:

1. Contact the upstream for the repository and get them to fix the problem.
2. Reconfigure the baseurl/etc. for the repository, to point to a working upstream. This is most often useful if you are using a newer distribution release than is supported by the repository (and the packages for the previous distribution release still work).
3. Disable the repository, so yum won't use it by default. Yum will then just ignore the repository until you permanently enable it again or use `--enablerepo` for temporary usage:  
  
`yum-config-manager --disable <repoid>`
4. Configure the failing repository to be skipped, if it is unavailable. Note that yum will try to contact the repo. when it runs most commands, so will have to try and fail each time (and thus. yum will be be much slower). If it is a very temporary problem though, this is often a nice compromise:  
  
`yum-config-manager --save --setopt=<repoid>.skip_if_unavailable=true`

```
Cannot find a valid baseurl for repo: amzn-main/latest
[ec2-user@ip-10-0-2-72 ~]$
```

## NAT Gateway creation:

To let the DB server connect to the internet, a NAT gateway needs to be created. And then a custom route table needs to be created that will route traffic only between the public EC2 instance which has the app server and the private DB server instance. For the current solution, the default route table will be used for the NAT gateway setup. NAT gateway created will need to be attached the owncloud VPC which has the internet connectivity.



aws Services Resource Groups

NAT Gateways > Create NAT Gateway

### Create NAT Gateway

Create a NAT gateway and assign it an Elastic IP address. [Learn more.](#)

Subnet\* subnet-Qf4d4a4972a800fde

Elastic IP Allocation ID\* eipalloc-035c913842410a303 [Create New EIP](#)

New EIP (3.93.161.238) creation successful.

\* Required [Cancel](#) [Create a NAT Gateway](#)

The NAT gateway created will need to be associated with a route table (selecting the default route table `def-owncloud-vpc-rt`).

The screenshot shows the AWS VPC Dashboard with the 'Create NAT Gateway' button highlighted. Below the table, the details for the NAT Gateway 'nat-03a46a944e1fc9257' are displayed. The status is 'available', the Elastic IP Address is '3.93.161.238', and the Private IP Address is '10.0.1.193'. The Network Interface ID is 'eni-0b71c633d8a4a8a33' and the VPC is 'vpc-0b21730eb5c8b2bed | owncloud-vpc'.

NAT Gateway ID	Status	Status Message	Elastic IP Address	Private IP Address	Network Interface	VPC
nat-03a46a944e1fc9257	available	-	3.93.161.238	10.0.1.193	eni-0b71c633d8a4a8a33	vpc-0b21730eb5c8b2bed   owncloud-vpc

**NAT Gateway: nat-03a46a944e1fc9257**

**Details**

NAT Gateway ID	nat-03a46a944e1fc9257	Status	available
Status Message	-	Elastic IP Address	3.93.161.238
Private IP Address	10.0.1.193	Network Interface ID	eni-0b71c633d8a4a8a33
VPC	vpc-0b21730eb5c8b2bed   owncloud-vpc	Subnet	subnet-0f4d4a4972a800fde   public-subnet
Created	November 18, 2019 at 1:32:02 AM UTC-5	Deleted	-

The screenshot shows the AWS VPC Dashboard with the 'Create subnet' button highlighted. Below the table, the details for the subnets are displayed. The subnets are 'public-subnet', 'private-subnet', and 'def-subnet-1f'. The status is 'available' for all. The Elastic IP Address is '3.93.161.238' and the Private IP Address is '10.0.1.193'.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone ID
public-subnet	subnet-0f4d4a4972a800fde	available	vpc-0b21730eb5c8b2bed   owncloud-vpc	10.0.1.0/24	249	-	us-east-1a	use1-az6
private-subnet	subnet-028c093bfd4cc7828	available	vpc-0b21730eb5c8b2bed   owncloud-vpc	10.0.2.0/24	250	-	us-east-1b	use1-az1
def-subnet-1f	subnet-c4632bfa	available	vpc-0f603175   def-vpc	172.31.48.0/20	4091	-	us-east-1e	use1-az3

Associate the default route table with the NAT gateway instance through subnet associations and adding the route to the destination:

The screenshot shows the AWS VPC Dashboard with the 'Create route table' button highlighted. Below the table, the details for the route tables are displayed. The route tables are 'def-owncloud-vpc-rt', 'def-vpc-rt', and 'oc-public-rt'. The status is 'active' for all. The Elastic IP Address is '3.93.161.238' and the Private IP Address is '10.0.1.193'.

Name	Route Table ID	Explicit subnet associations	Main	VPC ID	Owner
def-owncloud-vpc-rt	rtb-05d4dca820dacc0ea	-	Yes	vpc-0b21730eb5c8b2bed   owncloud-vpc	018894105410
def-vpc-rt	rtb-463be738	-	Yes	vpc-0f603175   def-vpc	018894105410
oc-public-rt	rtb-0848b90b6cc17f724	subnet-0f4d4a4972a800fde	No	vpc-0b21730eb5c8b2bed   owncloud-vpc	018894105410

**Route Table: rtb-05d4dca820dacc0ea**

**Summary**

**Edit routes**

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No

Route Tables > Edit routes

### Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	nat-03a46a944e1fc9257		No

Add route

\* Required

Cancel Save routes

VPC Dashboard

Filter by VPC: Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Create route table Actions

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet associations	Main	VPC ID	Owner
def-owncloud-vpc-rt	rtb-05d4dca820dacc0ea	-	Yes	vpc-0b21730eb5c8b2bed   owncloud-vpc	018894105410
def-vpc-rt	rtb-463be738	-	Yes	vpc-0f603175   def-vpc	018894105410
oc-public-rt	rtb-0848b90b6cc17f724	subnet-0f4d4a4972a800fde	No	vpc-0b21730eb5c8b2bed   owncloud-vpc	018894105410

Route Table: rtb-05d4dca820dacc0ea

Summary Routes Subnet 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	nat-03a46a944e1fc9257	active	No

Default VPC route table is now associated with the private subnet

VPC Dashboard

Filter by VPC: Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Create subnet Actions

Filter by tags and attributes or search by keyword

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone ID
public-subnet	subnet-0f4d4a4972a800fde	available	vpc-0b21730eb5c8b2bed ...	10.0.1.0/24	249	-	us-east-1a	use1-az6
private-subnet	subnet-028c093bfd4cc7828	available	vpc-0b21730eb5c8b2bed ...	10.0.2.0/24	250	-	us-east-1b	use1-az1
def-subnet-1f	subnet-c4632bfa	available	vpc-0f603175   def-vpc	172.31.48.0/20	4091	-	us-east-1e	use1-az3
def-subnet-1e	subnet-ac25daa2	available	vpc-0f603175   def-vpc	172.31.64.0/20	4091	-	us-east-1f	use1-az5
def-subnet-1d	subnet-5b924616	available	vpc-0f603175   def-vpc	172.31.16.0/20	4091	-	us-east-1d	use1-az4
def-subnet-1c	subnet-4c070462	available	vpc-0f603175   def-vpc	172.31.80.0/20	4091	-	us-east-1c	use1-az2
def-subnet-1b	subnet-36d0d551	available	vpc-0f603175   def-vpc	172.31.0.0/20	4091	-	us-east-1b	use1-az1
def-subnet-1a	subnet-12b5b34e	available	vpc-0f603175   def-vpc	172.31.32.0/20	4091	-	us-east-1a	use1-az6

Route Table: rtb-05d4dca820dacc0ea | def-owncloud-vpc-rt

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-03a46a944e1fc9257

Sudo yum update on the private instance now works

```
ec2-user@ip-10-0-2-72:~$ sudo yum update
Verifying : util-linux-2.23.2-59.29.amzn1.x86_64
Verifying : openssl-7.4p1-21.73.amzn1.x86_64
Verifying : tzdata-java-2019c-1.73.amzn1.noarch
Verifying : libmount-2.23.2-59.29.amzn1.x86_64
Verifying : openssl-clients-7.4p1-21.73.amzn1.x86_64
Verifying : 2:microcode_ctl-2.1-47.36.amzn1.x86_64
Verifying : libjpeg-turbo-1.2.90-8.16.amzn1.x86_64
Verifying : curl-7.61.1-12.93.amzn1.x86_64
Verifying : libjpeg-turbo-1.2.90-8.16.amzn1.x86_64
Verifying : ec2-net-utils-0.5-2.35.amzn1.noarch
Verifying : tzdata-2019a-1.71.amzn1.noarch
Verifying : libmount-2.23.2-59.29.amzn1.x86_64
Verifying : util-linux-2.23.2-59.29.amzn1.x86_64
Verifying : openssl-clients-7.4p1-21.73.amzn1.x86_64
Verifying : sudo-1.8.6p3-29.22.amzn1.x86_64
Verifying : python27-devel-2.7.16-1.129.amzn1.x86_64
Verifying : 2:microcode_ctl-2.1-47.36.amzn1.x86_64
Verifying : python27-2.7.16-1.129.amzn1.x86_64
Verifying : openssl-7.4p1-21.73.amzn1.x86_64
Verifying : libnghttp2-1.21.1-1.4.amzn1.x86_64
Verifying : libblkid-2.23.2-59.29.amzn1.x86_64
Verifying : ec2-utils-0.5-2.35.amzn1.noarch
Verifying : curl-7.61.1-12.93.amzn1.x86_64
Verifying : kernel-tools-4.14.138-89.102.amzn1.x86_64
Verifying : python27-libs-2.7.16-1.129.amzn1.x86_64
Verifying : libcurl-7.61.1-11.91.amzn1.x86_64
Verifying : libuuid-2.23.2-59.29.amzn1.x86_64
Verifying : openssl-server-7.4p1-21.73.amzn1.x86_64
Verifying : tzdata-java-2019a-1.71.amzn1.noarch

Installed:
kernel.x86_64 0:4.14.152-98.182.amzn1

Dependency Installed:
libsmartcols.x86_64 0:2.23.2-59.29.amzn1

Updated:
curl.x86_64 0:7.61.1-12.93.amzn1      ec2-net-utils.noarch 0:0.5-3.36.amzn1      ec2-utils.noarch 0:0.5-3.36.amzn1      kernel-tools.x86_64 0:4.14.152-98.182.amzn1
libblkid.x86_64 0:2.23.2-59.29.amzn1  libcurl.x86_64 0:7.61.1-12.93.amzn1      libjpeg-turbo.x86_64 0:1.2.90-8.16.amzn1      libmount.x86_64 0:2.23.2-59.29.amzn1
libnghttp2.x86_64 0:1.21.1-1.4.amzn1    microcode_ctl.x86_64 2:2.1-47.36.amzn1                    openssl.x86_64 0:7.4p1-21.73.amzn1
openssl-clients.x86_64 0:7.4p1-21.73.amzn1  openssl-server.x86_64 0:7.4p1-21.73.amzn1      python27-devel.x86_64 0:2.7.16-1.130.amzn1
python27-libs.x86_64 0:2.7.16-1.130.amzn1  sudo.x86_64 0:1.8.6p3-29.22.amzn1        tzdata.noarch 0:2019c-1.73.amzn1
tzdata-java.noarch 0:2019a-1.71.amzn1      tzdata.noarch 0:2019c-1.73.amzn1

Complete!
[ec2-user@ip-10-0-2-72 ~]$ sudo yum update
Failed to set locale, defaulting to C
Loaded plugins: priorities, update-motd, upgrade-helper
No packages marked for update
[ec2-user@ip-10-0-2-72 ~]$
```

Note: NAT instances can also be created to achieve the connectivity of the private subnet with a default VPC via a route table setup (NAT instances can be found via Community instances)

Validating the NAT gateway setup before updating the DB server instance via the console

Executing the below commands in the DB server console with `ec2-user@ip-10-0-2-72`

```
sudo apt-get install mysql-server
```

```
sudo mysql_secure_installation
```

```
sudo mysql
```

Create new DB and user with all privileges

```
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;
```

## Change the bind address

```
sudo vi /etc/mysql/mysql.conf.d/mysql.cnf
```

```
sudo systemctl restart mysql
```

## On web server install mysql client

```
mysql -uowncloud -h -p
```

Mysql-server DB successfully installed:

Screenshot accessing the DB installation in the console and the commands were executed

```
ec2-user@ip-10-0-2-72:~$ sudo yum install mysql-server
(7/13): perl-Data-Dumper-2.145-3.5.amzn1.x86_64.rpm
(8/13): perl-Net-Daemon-0.48-5.5.amzn1.noarch.rpm
(9/13): perl-PIRPC-0.2020-14.7.amzn1.noarch.rpm
(10/13): perl-IO-Compress-2.061-2.12.amzn1.noarch.rpm
(11/13): mysql55-libs-5.5.62-1.23.amzn1.x86_64.rpm
(12/13): perl-DBI-1.627-4.8.amzn1.x86_64.rpm
(13/13): mysql55-server-5.5.62-1.23.amzn1.x86_64.rpm
Total: 9.4 MB/s | 23 MB 00:00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : mysql55-libs-5.5.62-1.23.amzn1.x86_64 1/13
Installing : perl-Data-Dumper-2.145-3.5.amzn1.x86_64 2/13
Installing : mysql55-config-5.5.62-1.23.amzn1.x86_64 3/13
Installing : mysql55-libs-5.5.62-1.23.amzn1.x86_64 4/13
Installing : perl-Compress-Raw-Bzip2-2.061-3.11.amzn1.x86_64 5/13
Installing : perl-Net-Daemon-0.48-5.5.amzn1.noarch 6/13
Installing : 1:perl-Compress-Raw-Zlib-2.061-4.1.amzn1.x86_64 7/13
Installing : perl-IO-Compress-2.061-2.12.amzn1.noarch 8/13
Installing : perl-PIRPC-0.2020-14.7.amzn1.noarch 9/13
Installing : perl-DBI-1.627-4.8.amzn1.x86_64 10/13
Installing : perl-DBD-MySQL55-4.023-5.23.amzn1.x86_64 11/13
Installing : mysql55-server-5.5.62-1.23.amzn1.x86_64 12/13
Installing : mysql-server-5.5-1.6.amzn1.noarch 13/13
Verifying : perl-DBI-1.627-4.8.amzn1.x86_64 1/13
Verifying : mysql55-server-5.5.62-1.23.amzn1.x86_64 2/13
Verifying : perl-IO-Compress-2.061-2.12.amzn1.noarch 3/13
Verifying : mysql55-libs-5.5.62-1.23.amzn1.x86_64 4/13
Verifying : 1:perl-Compress-Raw-Zlib-2.061-4.1.amzn1.x86_64 5/13
Verifying : mysql55-libs-5.5.62-1.23.amzn1.x86_64 6/13
Verifying : mysql55-config-5.5.62-1.23.amzn1.x86_64 7/13
Verifying : perl-DBD-MySQL55-4.023-5.23.amzn1.x86_64 8/13
Verifying : perl-Net-Daemon-0.48-5.5.amzn1.noarch 9/13
Verifying : perl-Compress-Raw-Bzip2-2.061-3.11.amzn1.x86_64 10/13
Verifying : perl-PIRPC-0.2020-14.7.amzn1.noarch 11/13
Verifying : perl-Data-Dumper-2.145-3.5.amzn1.x86_64 12/13
Verifying : mysql-server-5.5-1.6.amzn1.noarch 13/13
Installed:
mysql-server.noarch 0:5.5-1.6.amzn1
Dependency Installed:
mysql-config.x86_64 0:5.5.62-1.23.amzn1      mysql55.x86_64 0:5.5.62-1.23.amzn1      mysql55-libs.x86_64 0:5.5.62-1.23.amzn1      mysql55-server.x86_64 0:5.5.62-1.23.amzn1
perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.11.amzn1      perl-Compress-Raw-Zlib.x86_64 1:2.061-4.1.amzn1      perl-DBD-MySQL55.x86_64 0:4.023-5.23.amzn1      perl-DBI.x86_64 0:1.627-4.8.amzn1
perl-Data-Dumper.x86_64 0:2.145-3.5.amzn1      perl-IO-Compress.noarch 0:2.061-2.12.amzn1      perl-Net-Daemon.noarch 0:0.48-5.5.amzn1      perl-PIRPC.noarch 0:0.2020-14.7.amzn1
Complete!
[ec2-user@ip-10-0-2-72 ~]$
```