

### List of Experiments for Cloud Computing Lab:

1. [Install Virtualbox/VMware Workstation with different flavours of linux or windows OS](#)
2. [Install a C compiler in the virtual machine created using virtual box and execute Simple Programs](#)
3. [Installation and configuration of own Cloud/nextCloud](#)
4. [Write a Program to Create, Manage and groups User accounts in own Cloud by Installing Administrative Features.](#)
5. [Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim](#)
6. [Create AWS Free Trial Account and Create & connect to Amazon EC2 Machine](#)
7. [Create S3 Bucket in AWS, Upload & Access a File, And Host a Simple Website](#)
8. [S3 Cross-Region Replication](#)
9. [Create & Manage EBS Volumes & Snapshots](#)
10. [Attach & Mount EBS Volume to EC2 Instance](#)

### Web Sources:

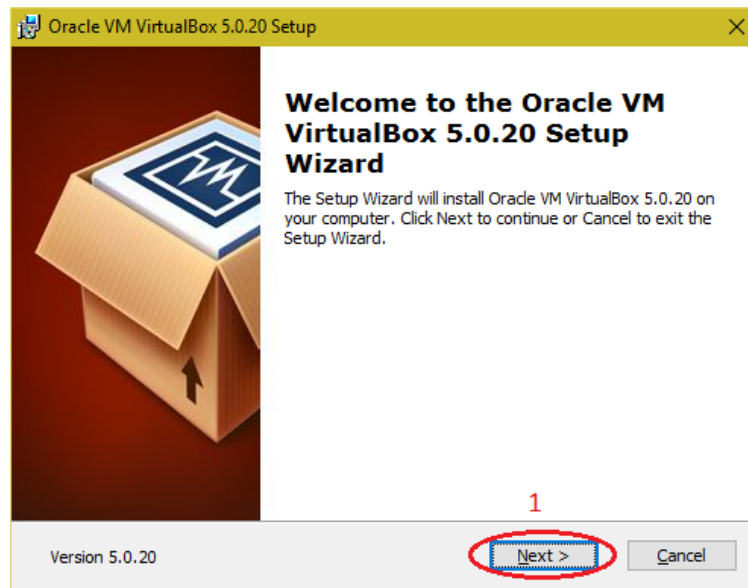
<https://www.vmware.com/in/products/workstation-player.html> - vmware  
<https://www.virtualbox.org/wiki/Downloads> - virtualbox  
<https://www.tecmint.com/install-owncloud-on-ubuntu/> - owncloud  
<https://www.tecmint.com/install-nextcloud-in-ubuntu/> - nextcloud  
<http://cloudbus.org/cloudsim/> - CloudSim  
<https://aws.amazon.com/> - AWS  
<https://techglimpse.com/cloud-modeling-simulation-toolkit-tutorial/> - CloudSim  
Installation Procedure

# **1. Install Virtualbox/VMware Workstation with different flavours of linux or windows OS**

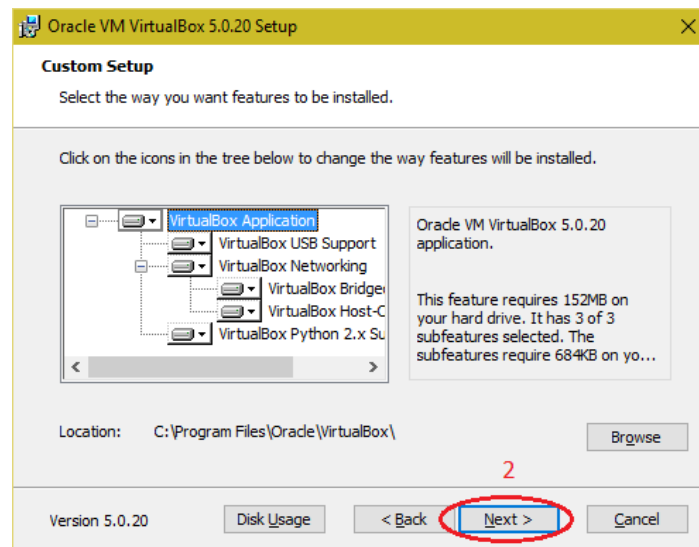
To install VirtualBox use the following steps:

Download VirtualBox setup for installation

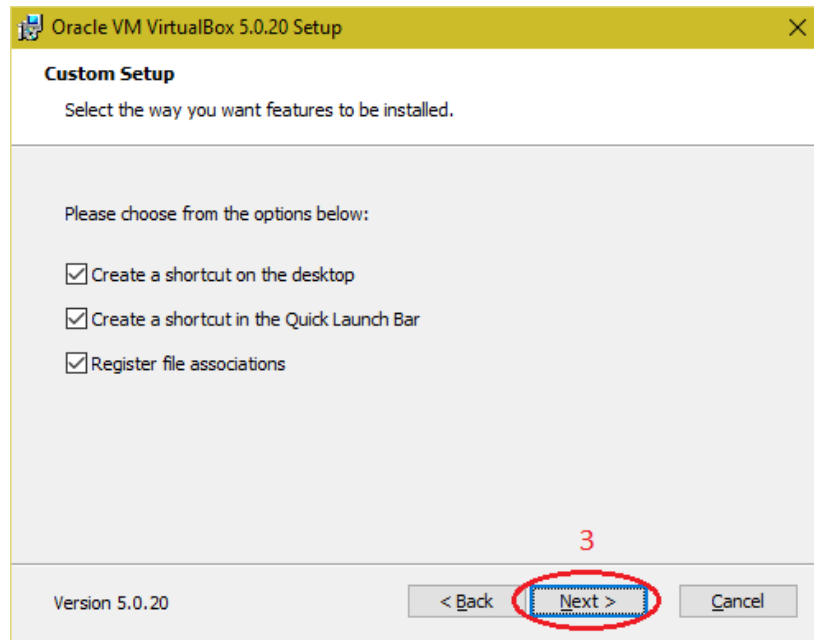
Run the Virtual Box setup and click on "Next" button.



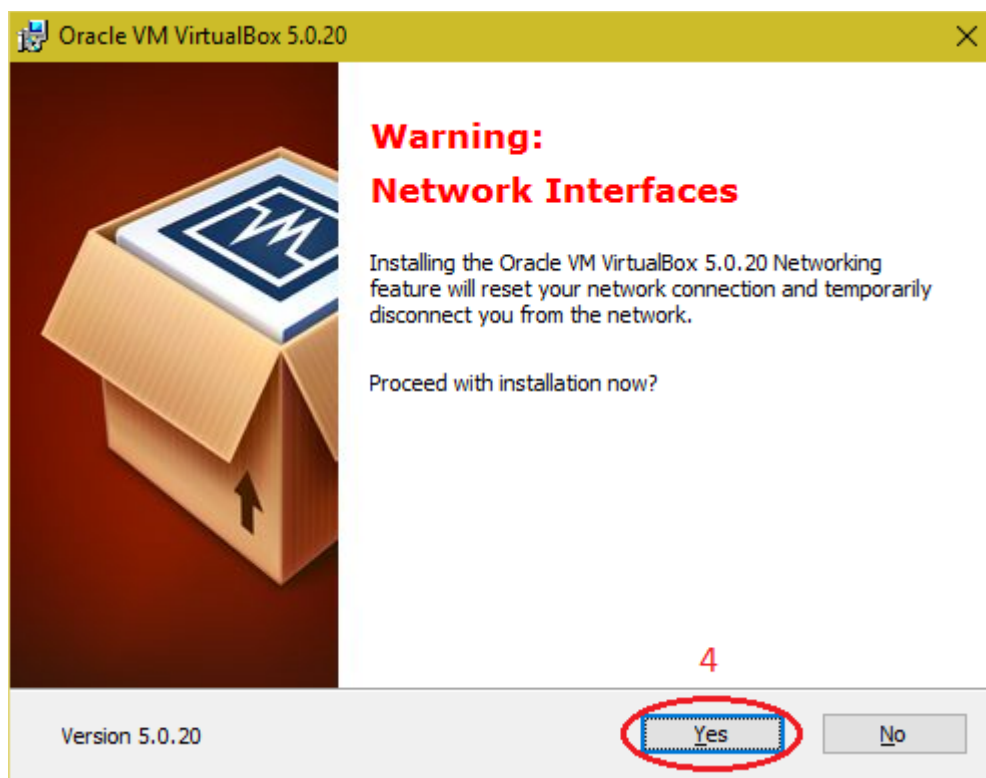
Click on "Next" button.



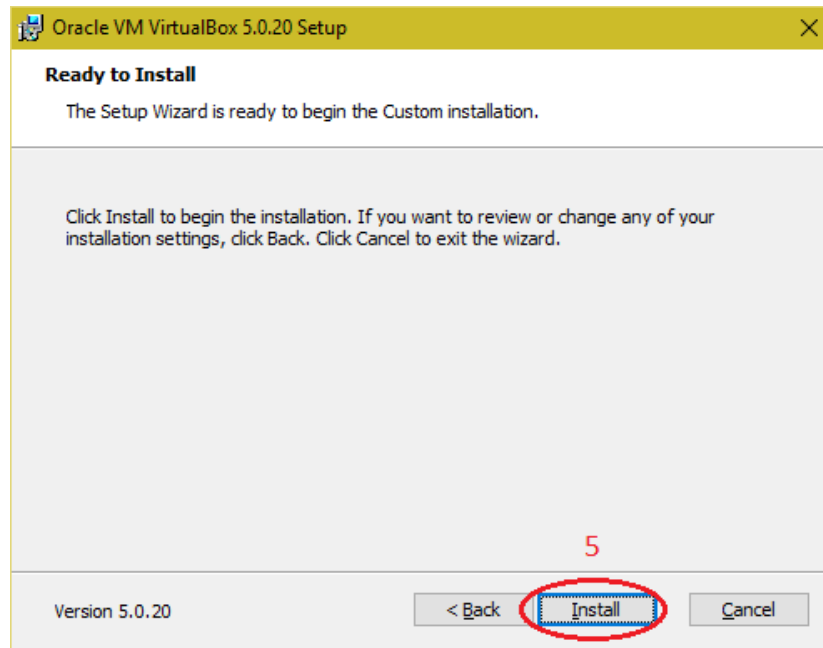
Click on "Next" button.



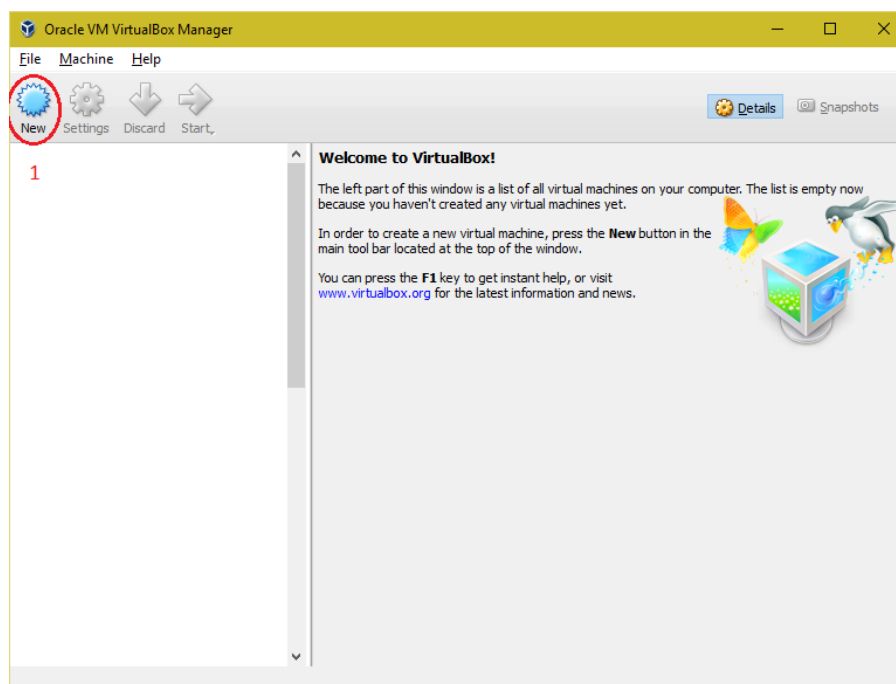
Click on "Yes" button.



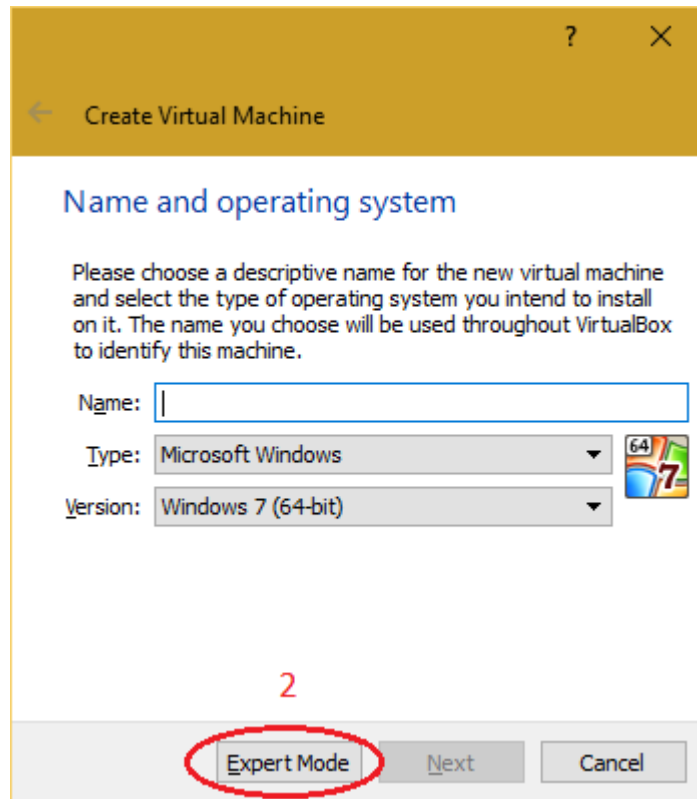
Click on "Install" button.



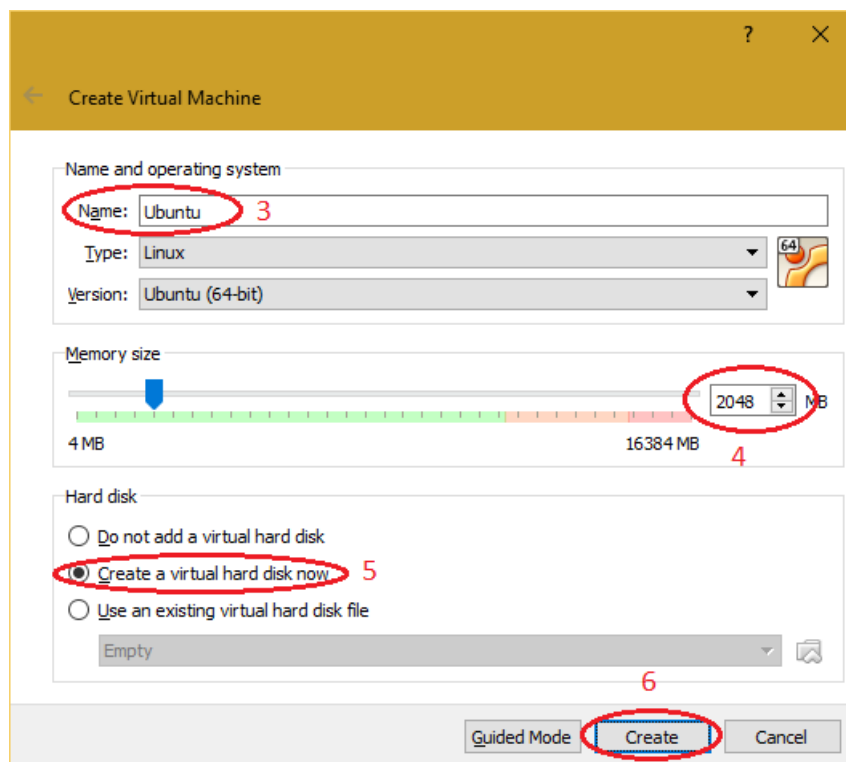
1. Open "Oracle VM VirtualBox Manager".



2. Click on "New" button and select "Expert Mode".



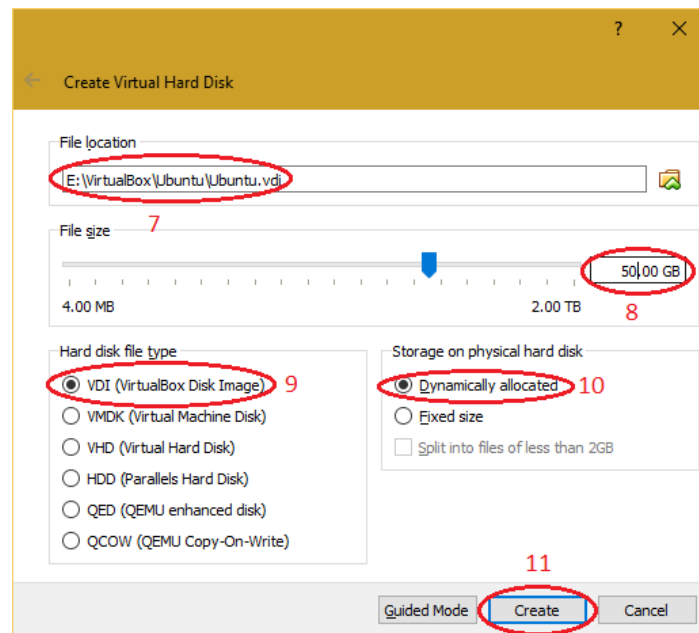
3. Provide the name and operating system information for virtual machine.



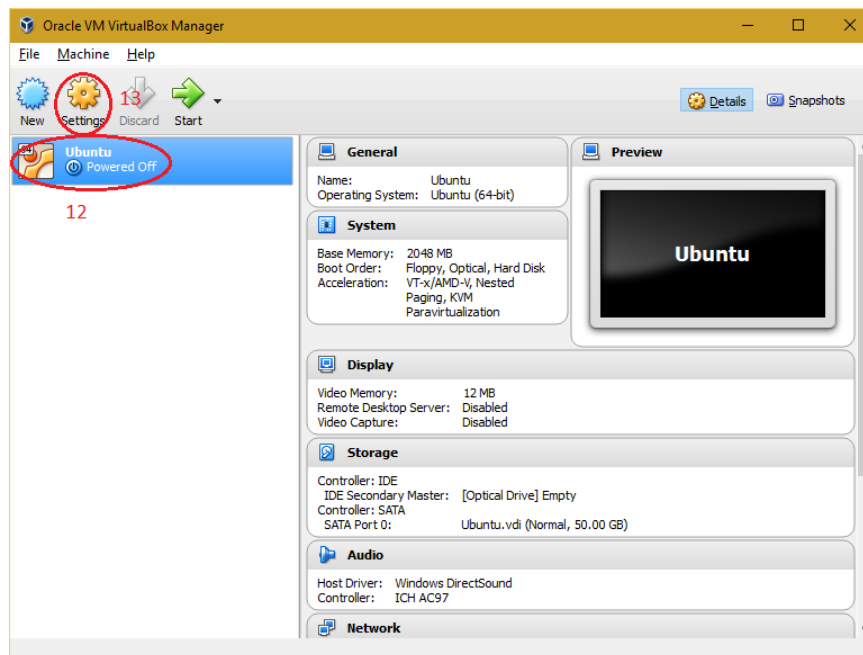
**Note:** Before installing 64-bit operating system, Intel VT-x/AMD-V must be enabled in "BIOS" on the system. To enable Intel VT-x/AMD-V, open BIOS and search for "Intel

Virtualization Technology" or "AMD-V", save the BIOS and boot the PC/Laptop.

4. Select the path for the virtual hard disk and click on "Create" button.

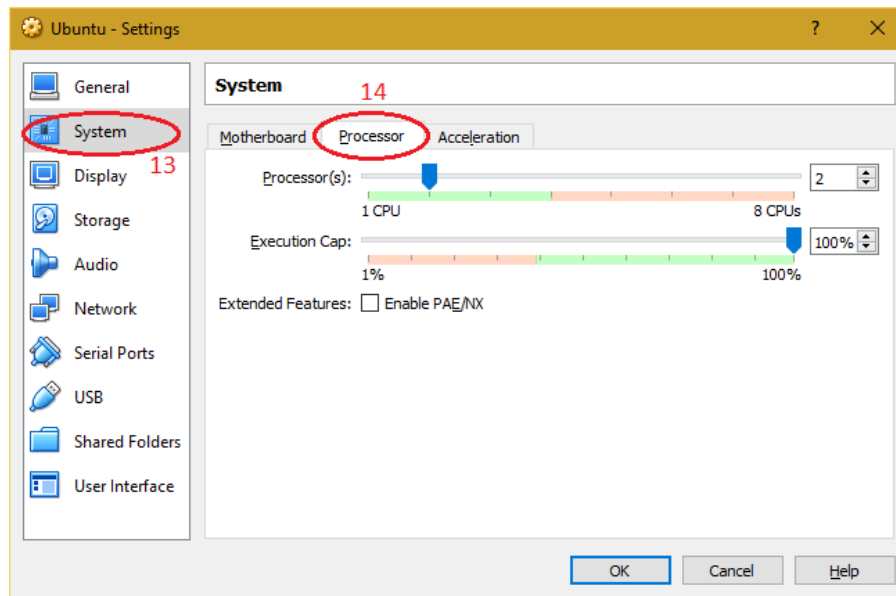


5. Select the virtual machine from the virtual box manager and click on "Settings" button.

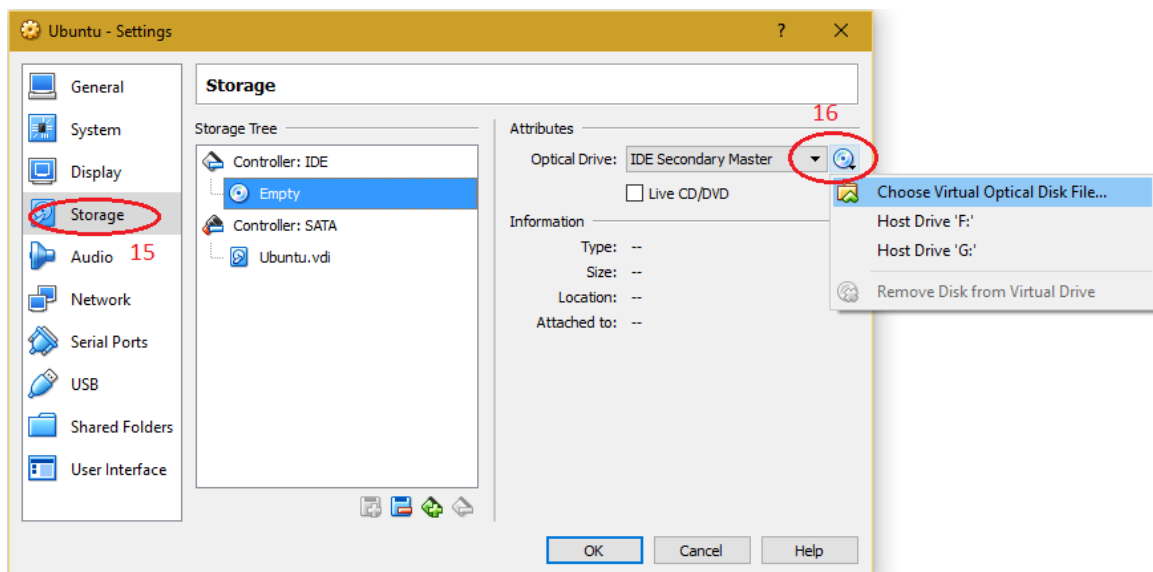


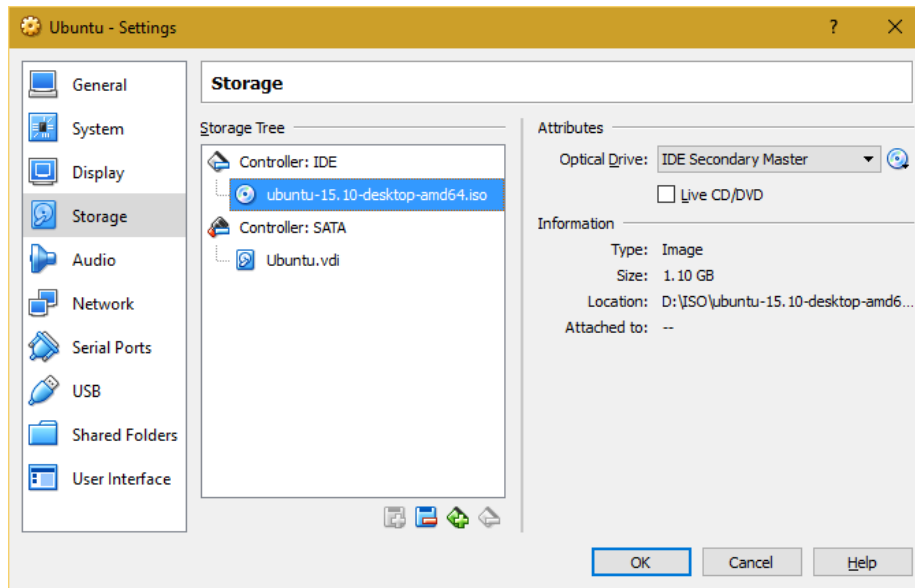
6. Select "System" and navigate to "Processor" tab to adjust number of processor of virtual

machine for better performance.

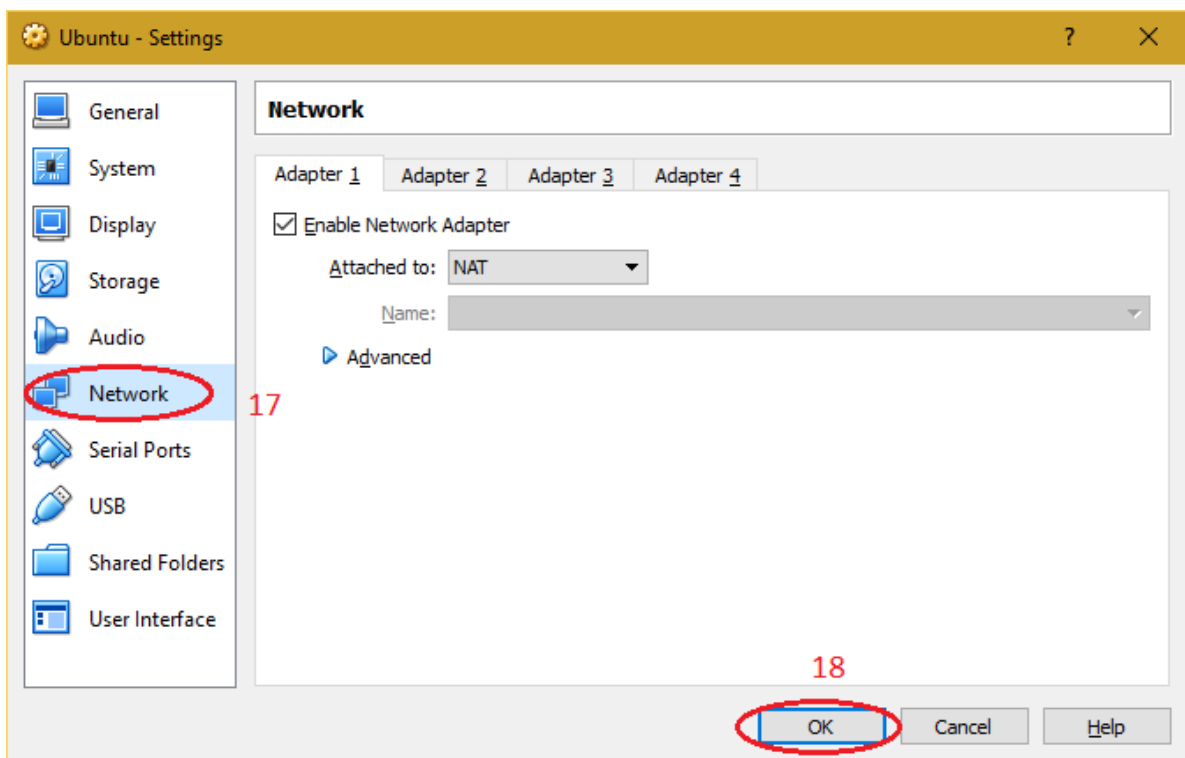


7. Select “Storage” and choose the installation media of Operating System (ISO/CD/DVD). Preferred Linux “.iso” can be downloaded from [CC ftp site](#). Also many different flavours of Linux are available on the internet – [Fedora](#), [CentOS](#), [Ubuntu](#), [Debian](#), [Mageia](#), [openSUSE](#), [Arch Linux](#), [Slackware Linux](#), etc.



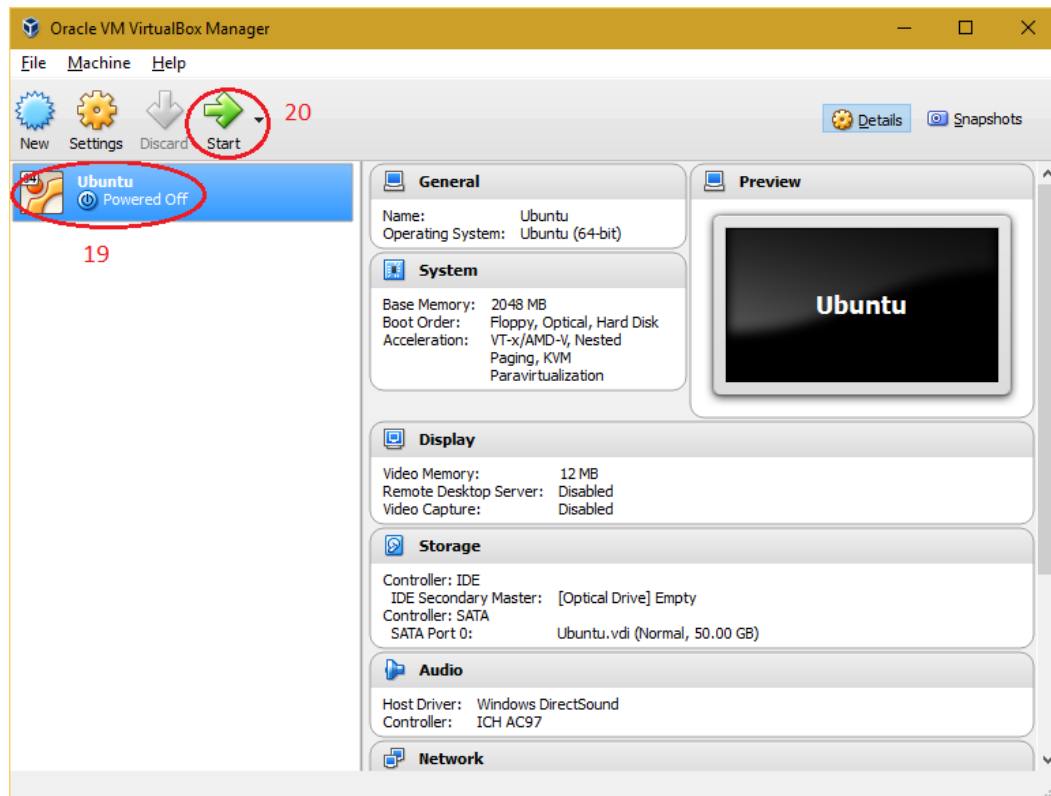


8. Select “Network” to make changes required for network setting of virtual machine and click on “OK”

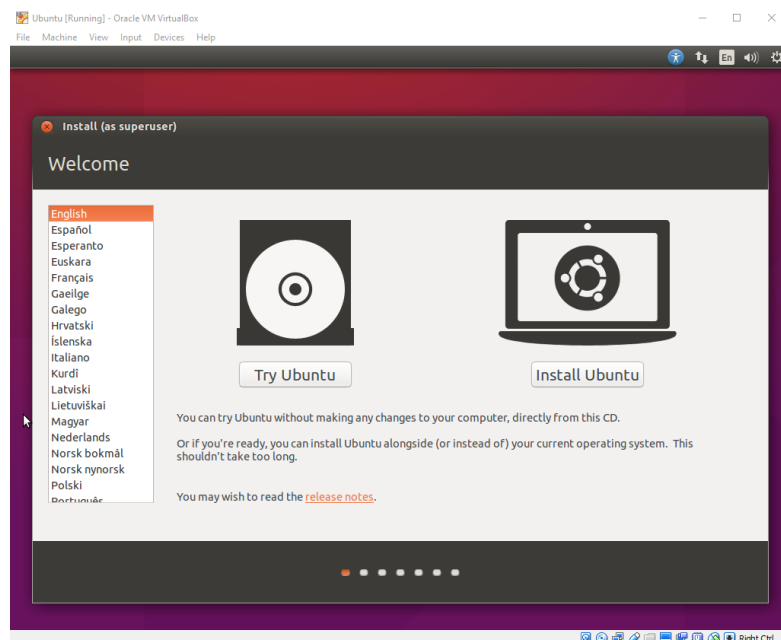




9. Select the created virtual machine and click on "Start" button.

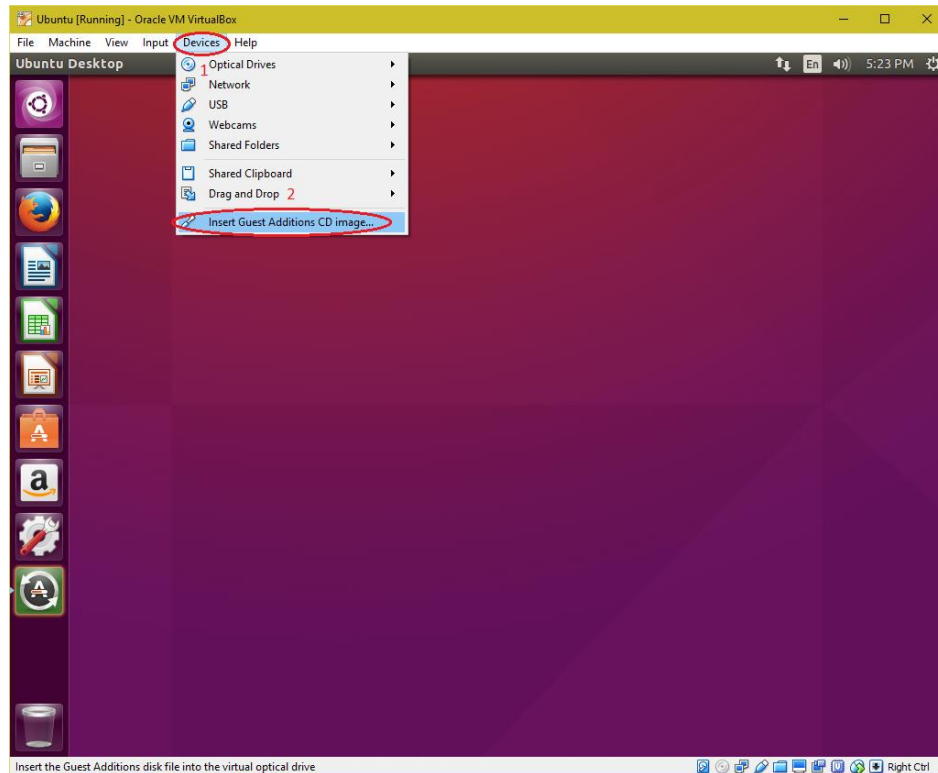


10. Proceed with the installation of operating system in virtual machine.

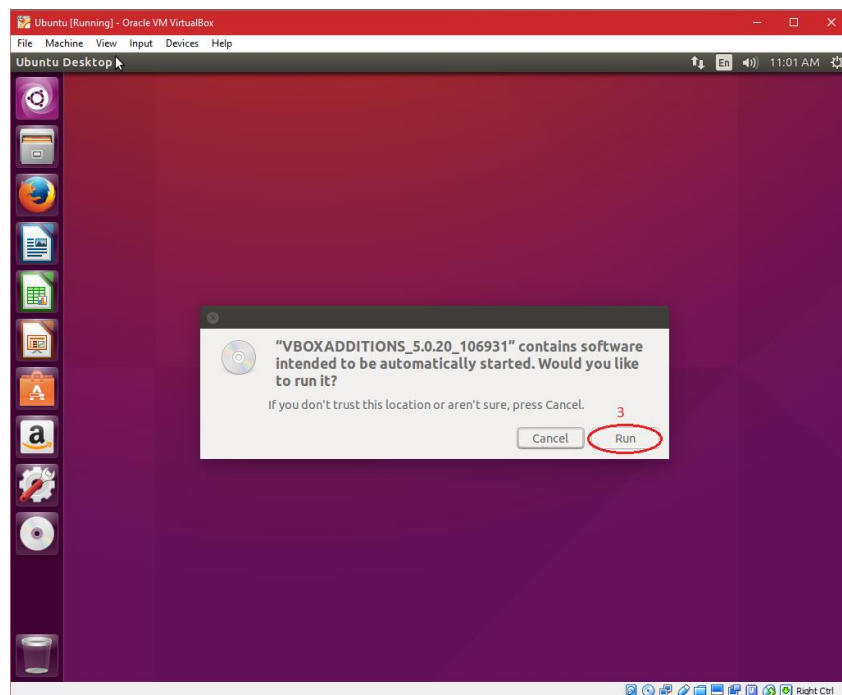


Installing "Guest Additions CD image" in Ubuntu virtual machine.

1. Login to user account on Ubuntu virtual machine.  
Select "Device > Insert Guest Addition CD image".

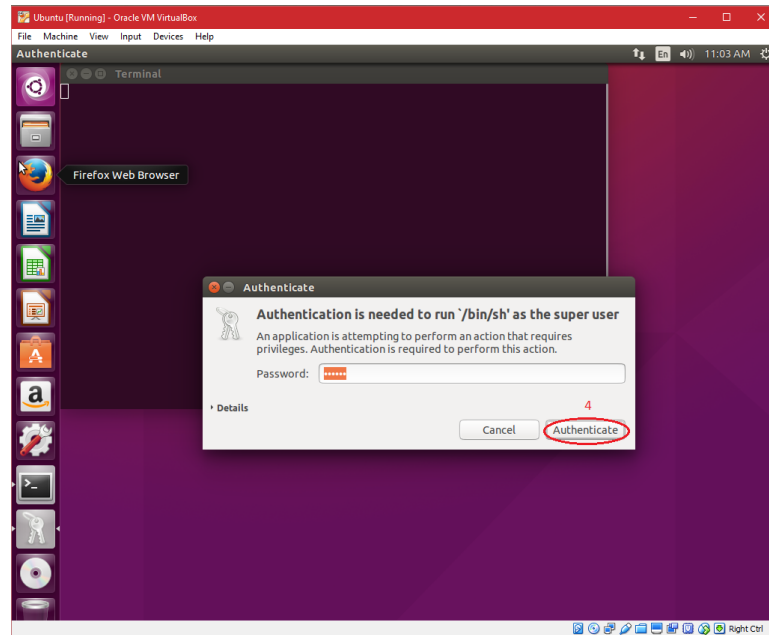


2. Click "Run" to install "Guest Additions".



3. Input the authentication of super user and click on "Authenticate". After completing

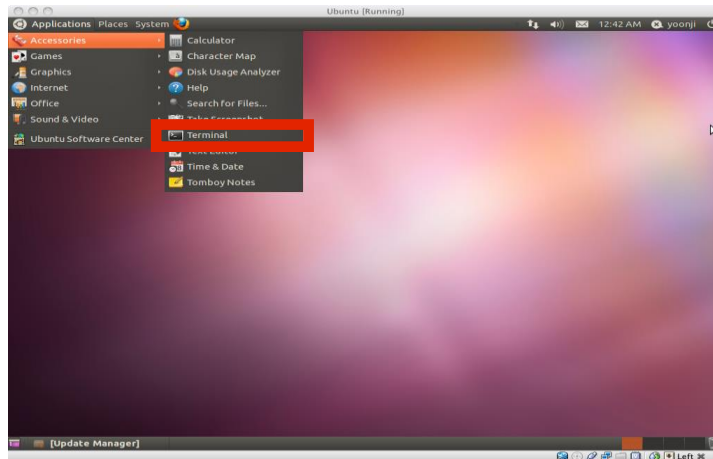
installtion "Reboot" the virtual machine.



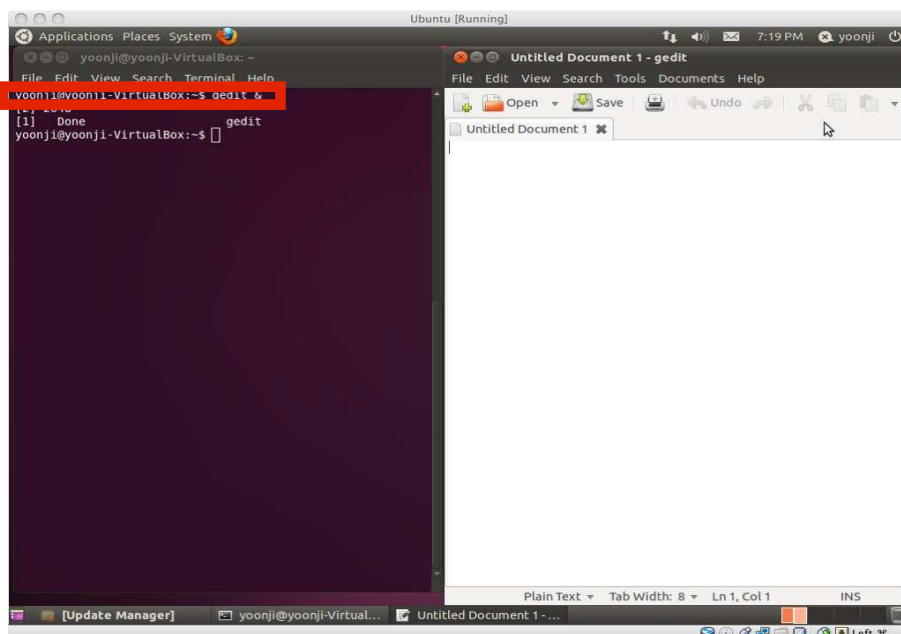
## **2. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs**

Follow the below steps:

1. Open Terminal (Applications-Accessories-Terminal)



2. Open gedit by typing “gedit &” on terminal  
(You can also use any other Text Editor application)



3. Type the following on gedit (or any other text editor)

```
#include<stdio.h> main()  
{  
printf("Hello World\n");  
}
```

4. Save this file as “helloworld.c”

5. Type “ls” on Terminal to see all files under current folder

6. Confirm that “helloworld.c” is in the current directory. If not, type cd  
DIRECTORY\_PATH to go to the directory that has “helloworld.c”

7. Type “gcc helloworld.c” to compile, and type “ls” to confirm that a new executable  
file “a.out” is created

8. Type “./a.out” on Terminal to run the program

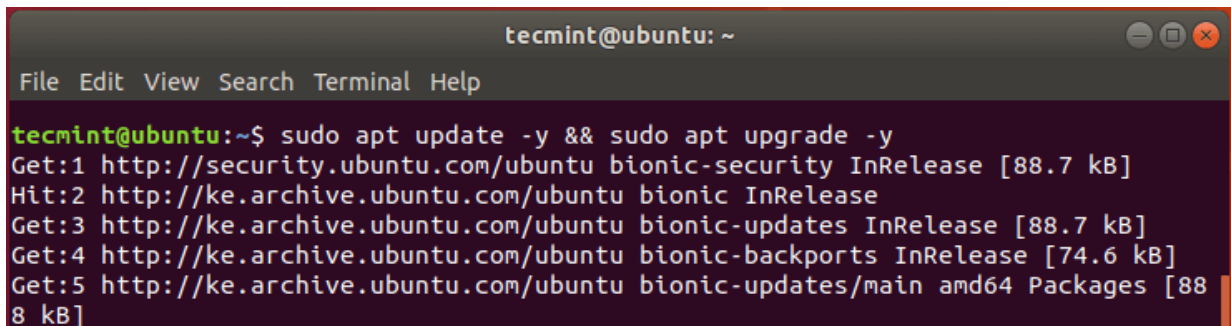
9. If you see “Hello World” on the next line,  
you just successfully ran your first C program!

### 3. Installation and configuration of own Cloud/nextCloud

#### Step 1: Update Ubuntu System Packages

Before getting started, update the system packages and repositories using the following commands:

```
$ sudo apt update -y && sudo apt upgrade -y
```



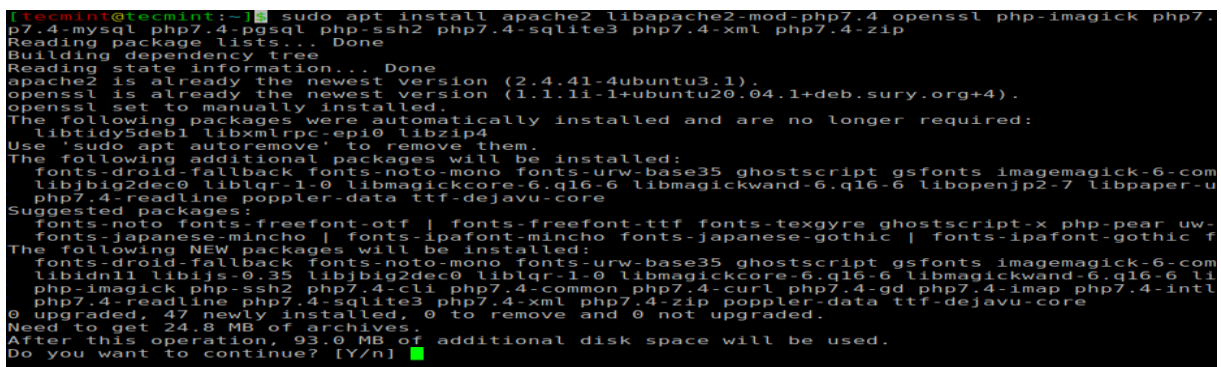
```
tecmin@ubuntu: ~  
File Edit View Search Terminal Help  
tecmin@ubuntu:~$ sudo apt update -y && sudo apt upgrade -y  
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]  
Hit:2 http://ke.archive.ubuntu.com/ubuntu bionic InRelease  
Get:3 http://ke.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]  
Get:4 http://ke.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]  
Get:5 http://ke.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [88  
8 kB]
```

Update Ubuntu System Packages

#### Step 2: Install Apache and PHP 7.2 in Ubuntu

**OwnCloud** is built on **PHP** and is typically accessed via a web interface. For this reason, we are going to install **the Apache** webserver to serve **Owncloud** files as well as **PHP 7.2** and additional PHP modules necessary for **OwnCloud** to function smoothly.

```
$ sudo apt install apache2 libapache2-mod-php7.2 openssl php-imagick php7.2-common  
php7.2-curl php7.2-gd php7.2-imap php7.2-intl php7.2-json php7.2-ldap php7.2-mbstring  
php7.2-mysql php7.2-pgsql php-smbclient php-ssh2 php7.2-sqlite3 php7.2-xml php7.2-zip
```



```
tecmin@tecmin:~$ sudo apt install apache2 libapache2-mod-php7.2 openssl php-imagick php7.2-common  
php7.2-mysql php7.2-pgsql php-ssh2 php7.2-sqlite3 php7.4-xml php7.4-zip  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
apache2 is already the newest version (2.4.41-4ubuntu3.1).  
openssl is already the newest version (1.1.1f-1+ubuntu20.04.1+deb.sury.org+4).  
openssl set to manually installed.  
The following packages were automatically installed and are no longer required:  
  libtidy5deb1 libxmlrpc-epi0 libzip4  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
  fonts-droid-fallback fonts-noto-mono fonts-urw-base35 ghostscript gsfonts imagemagick-6-com  
  libbig2dec0 liblqr-1-0 libmagickcore-6.q16-6 libmagickwand-6.q16-6 libopenjp2-7 libpaper-u  
  php7.4-readline poppler-data ttf-dejavu-core  
Suggested packages:  
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf fonts-texgyre ghostscript-x php-pear uw-  
  fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic f  
The following NEW packages will be installed:  
  fonts-droid-fallback fonts-noto-mono fonts-urw-base35 ghostscript gsfonts imagemagick-6-com  
  libbig2dec0 liblqr-1-0 libmagickcore-6.q16-6 libmagickwand-6.q16-6 libopenjp2-7 libpaper-u  
  php7.4-readline php7.4-curl php7.4-common php7.4-xml php7.4-zip poppler-data ttf-dejavu-core  
Need to get 24.8 MB of archives.  
After this operation, 93.0 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

Install Apache and PHP in Ubuntu

Once the installation is complete you can verify if **Apache** is installed by running the [dpkg command](#).

```
$ sudo dpkg -l apache2
```

From the output, we can see that we have installed **Apache** version **2.4.29**.

```
tecmin@ubuntu: ~  
File Edit View Search Terminal Help  
tecmin@ubuntu:~$ sudo dpkg -l apache2  
Desired=Unknown/Install/Remove/Purge/Hold  
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend  
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)  
||/ Name           Version           Architecture Description  
++-=====+-----+-----+-----+  
ii  apache2          2.4.29-1ubuntu1  amd64        Apache HTTP Server  
tecmin@ubuntu:~$
```

Check Apache Version in Ubuntu

To start and enable **Apache** to run on boot, run the commands.

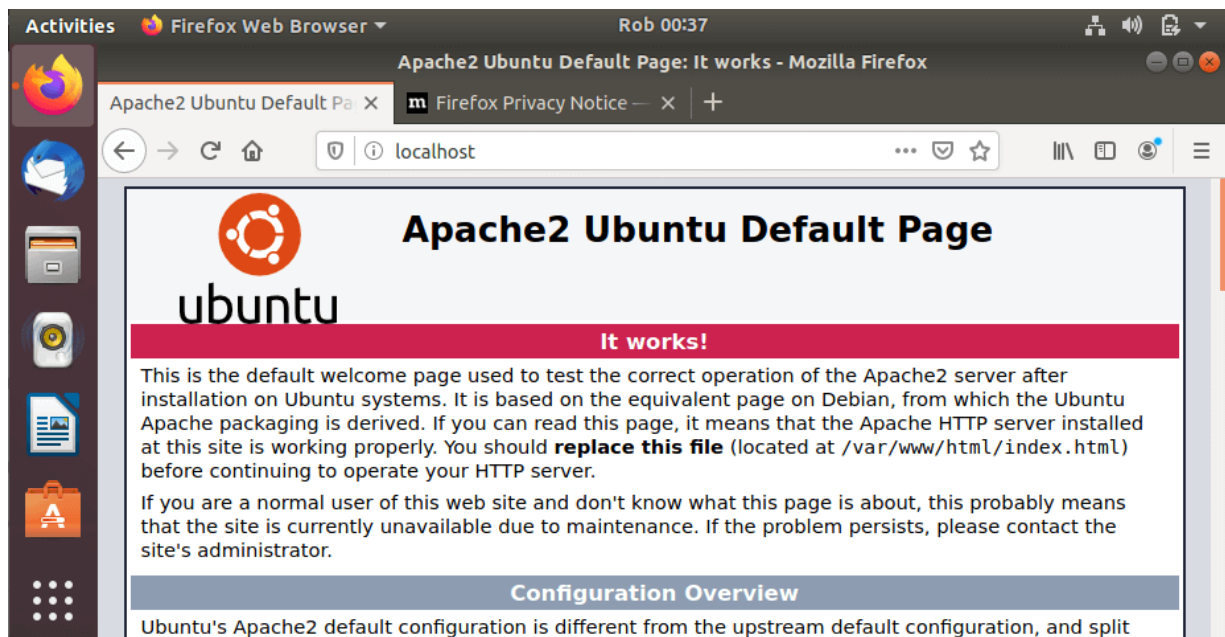
```
$ sudo systemctl start apache2
```

```
$ sudo systemctl enable apache2
```

Now head over to your browser and type in your server's IP address in the URL bar as shown:

http://server-IP

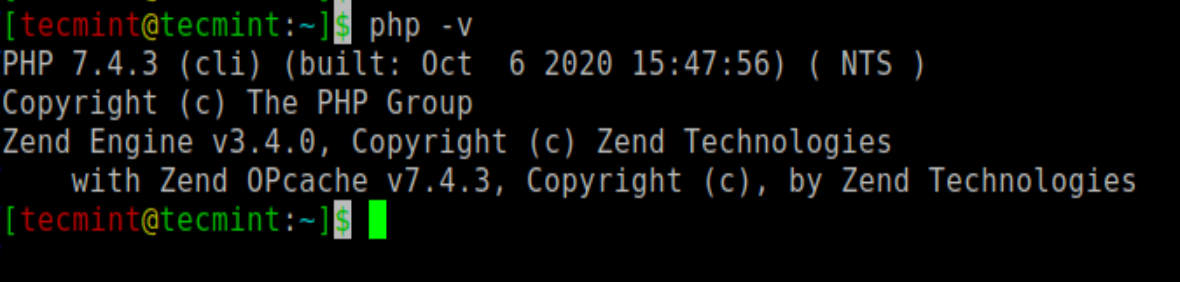
You should get a webpage below showing that **Apache** is installed and running.



Verify Apache Page in Ubuntu

To check if **PHP** is installed.

```
$ php -v
```

A terminal window with a black background and green text. The prompt is [tecmint@tecmint:~]\$. The command php -v has been executed, showing the following output: PHP 7.4.3 (cli) (built: Oct 6 2020 15:47:56) ( NTS ) Copyright (c) The PHP Group Zend Engine v3.4.0, Copyright (c) Zend Technologies with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies. The prompt is now [tecmint@tecmint:~]\$.

```
[tecmint@tecmint:~]$ php -v
PHP 7.4.3 (cli) (built: Oct 6 2020 15:47:56) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
[tecmint@tecmint:~]$
```

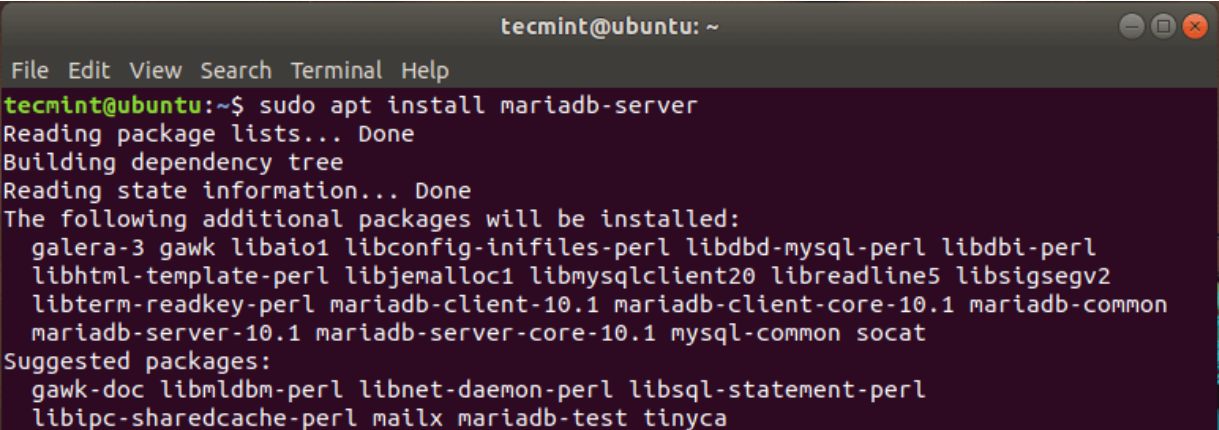
Check PHP Version in Ubuntu

### Step 3: Install MariaDB in Ubuntu

**MariaDB** is a popular open-source database server that is widely used by developers, database enthusiasts, and also in production environments. It's a fork of **MySQL** and has been preferred to **MySQL** since the takeover of **MySQL** by **Oracle**.

To install **the MariaDB** run.

```
$ sudo apt install mariadb-server
```

A terminal window titled 'tecmint@ubuntu: ~' with a dark purple background and green text. The prompt is tecmint@ubuntu:~\$. The command sudo apt install mariadb-server has been executed. The output shows: Reading package lists... Done, Building dependency tree, Reading state information... Done. It then lists additional packages to be installed: galera-3, gawk, libaio1, libconfig-inifiles-perl, libdbd-mysql-perl, libdbi-perl, libhtml-template-perl, libjemalloc1, libmysqlclient20, libreadline5, libsigsegv2, libterm-readkey-perl, mariadb-client-10.1, mariadb-client-core-10.1, mariadb-common, mariadb-server-10.1, mariadb-server-core-10.1, mysql-common, and socat. It also lists suggested packages: gawk-doc, libmldbm-perl, libnet-daemon-perl, libsql-statement-perl, libipc-sharedcache-perl, mailx, mariadb-test, and tinyca. The prompt is now tecmint@ubuntu:~\$.

```
tecmint@ubuntu: ~
File Edit View Search Terminal Help
tecmint@ubuntu:~$ sudo apt install mariadb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  galera-3 gawk libaio1 libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl
  libhtml-template-perl libjemalloc1 libmysqlclient20 libreadline5 libsigsegv2
  libterm-readkey-perl mariadb-client-10.1 mariadb-client-core-10.1 mariadb-common
  mariadb-server-10.1 mariadb-server-core-10.1 mysql-common socat
Suggested packages:
  gawk-doc libmldbm-perl libnet-daemon-perl libsql-statement-perl
  libipc-sharedcache-perl mailx mariadb-test tinyca
```

Install MariaDB in Ubuntu

By default, **MariaDB** is not secured and is prone to security breaches. We, therefore, need to perform additional steps to harden the MariaDB server.



To get started with securing your MySQL server, run the command:

```
$ sudo mysql_secure_installation
```

Hit **ENTER** when prompted for the root password and press 'Y' to set the root password.

```
tecmin@ubuntu: ~  
File Edit View Search Terminal Help  
tecmin@ubuntu:~$ sudo mysql_secure_installation  
  
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB  
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!  
  
In order to log into MariaDB to secure it, we'll need the current  
password for the root user. If you've just installed MariaDB, and  
you haven't set the root password yet, the password will be blank,  
so you should just press enter here.  
  
Enter current password for root (enter for none):  
OK, successfully used password, moving on...  
  
Setting the root password ensures that nobody can log into the MariaDB  
root user without the proper authorisation.  
  
Set root password? [Y/n] Y  
New password:  
Re-enter new password:  
Password updated successfully!  
Reloading privilege tables..  
... Success!
```

## Set MySQL Password in Ubuntu

For the remaining prompts, simply type 'Y' and hit **ENTER**.

```
tecmin@ubuntu: ~  
File Edit View Search Terminal Help  
Remove anonymous users? [Y/n] Y  
... Success!  
  
Normally, root should only be allowed to connect from 'localhost'. This  
ensures that someone cannot guess at the root password from the network.  
  
Disallow root login remotely? [Y/n] Y  
... Success!  
  
By default, MariaDB comes with a database named 'test' that anyone can  
access. This is also intended only for testing, and should be removed  
before moving into a production environment.  
  
Remove test database and access to it? [Y/n] Y  
- Dropping test database...  
... Success!  
- Removing privileges on test database...  
... Success!  
  
Reloading the privilege tables will ensure that all changes made so far  
will take effect immediately.  
  
Reload privilege tables now? [Y/n] Y  
... Success!  
  
Cleaning up...  
  
All done! If you've completed all of the above steps, your MariaDB  
installation should now be secure.
```

## Secure MySQL in Ubuntu

Your MariaDB server is now secured to a decent level.

### Step 4: Create an OwnCloud Database

We need to create a database for **Owncloud** to store files during and after installation. So log in to **MariaDB**.

```
$ sudo mysql -u root -p
```

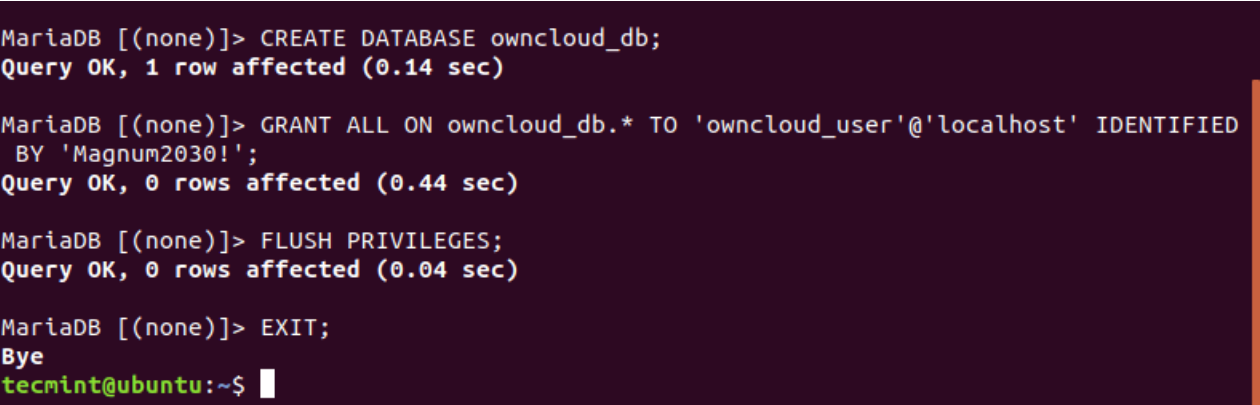
Run the commands below:

```
MariaDB [(none)]> CREATE DATABASE owncloud_db;
```

```
MariaDB [(none)]> GRANT ALL ON owncloud_db.* TO 'owncloud_user'@'localhost' IDENTIFIED BY 'StrongP@ssword';
```

```
MariaDB [(none)]> FLUSH PRIVILEGES;
```

```
MariaDB [(none)]> EXIT;
```



```
MariaDB [(none)]> CREATE DATABASE owncloud_db;
Query OK, 1 row affected (0.14 sec)

MariaDB [(none)]> GRANT ALL ON owncloud_db.* TO 'owncloud_user'@'localhost' IDENTIFIED BY 'Magnum2030!';
Query OK, 0 rows affected (0.44 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.04 sec)

MariaDB [(none)]> EXIT;
Bye
tecmint@ubuntu:~$
```

Create OwnCloud Database in Ubuntu

### Step 5: Download OwnCloud in Ubuntu

After creating the database, now [download the OwnCloud](https://download.owncloud.org/community/owncloud-10.4.0.zip) zipped file using the following [wget command](#).

```
$ sudo wget https://download.owncloud.org/community/owncloud-10.4.0.zip
```

Once downloaded, unzip the zipped package to the /var/www/ directory.

```
$ sudo unzip owncloud-10.4.0.zip -d /var/www/
```

Then, set permissions.

```
$ sudo chown -R www-data:www-data /var/www/owncloud/
```

```
$ sudo chmod -R 755 /var/www/owncloud/
```

## Step 6: Configure Apache for OwnCloud

In this step, we are going to configure **Apache** to serve OwnCloud's files. To do that, we are going to create a configuration file for **Owncloud** as shown.

```
$ sudo vim /etc/apache2/conf-available/owncloud.conf
```

Add the configuration below.

```
Alias /owncloud "/var/www/owncloud/"
```

```
<Directory /var/www/owncloud/>
```

```
Options +FollowSymlinks
```

```
AllowOverride All
```

```
<IfModule mod_dav.c>
```

```
Dav off
```

```
</IfModule>
```

```
SetEnv HOME /var/www/owncloud
```

```
SetEnv HTTP_HOME /var/www/owncloud
```

```
</Directory>
```

Save and close the file.   => :w =>qa! =>sudo no => :16 => j,k,l,o =>:wq

Next, you need to enable all the required Apache modules and the newly added configuration by running the commands below:

```
$ sudo a2enconf owncloud
```

```
$ sudo a2enmod rewrite
```

```
$ sudo a2enmod headers
```

```
$ sudo a2enmod env
```

```
$ sudo a2enmod dir
```

```
$ sudo a2enmod mime
```

For the changes to come into effect restart the Apache webserver.

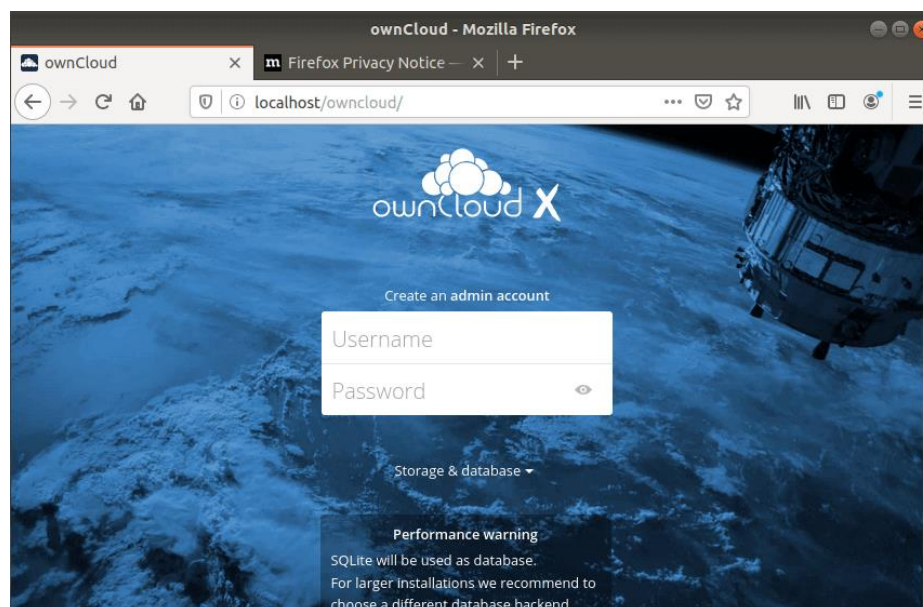
```
$ sudo systemctl restart apache2
```

### Step 7: Finalizing the OwnCloud Installation in Ubuntu

With all the necessary configurations finalized, the only part remaining is to install **OwnCloud** on a browser. So head out to your browser and type in your server's address followed by the /owncloud suffix.

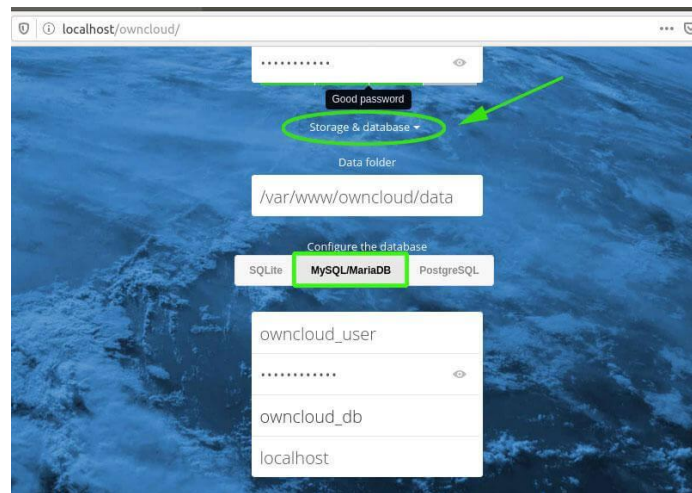
<http://server-IP/owncloud>

You will be presented with a web page similar to the one below.



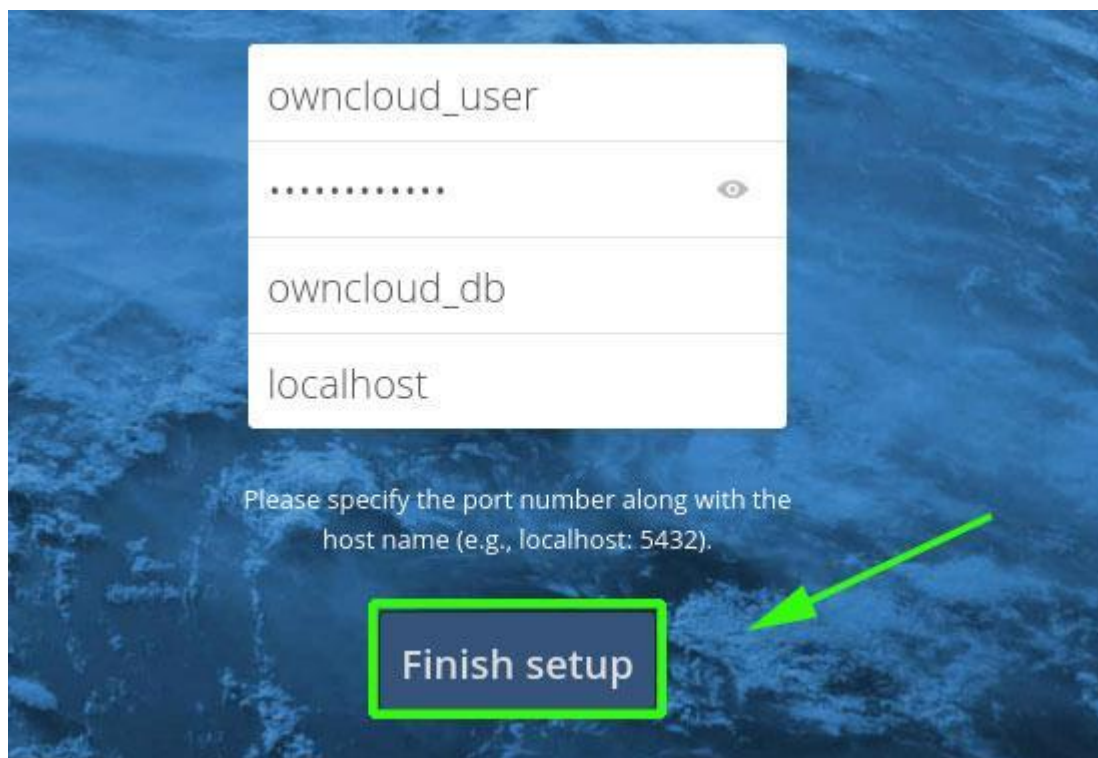
Create OwnCloud Admin Account

Just below, click on **'Storage and database'**. Select **'MySQL / MariaDB'** under the **'configure the database'** section and fill in the database credentials that you defined whilst creating the database for OwnCloud i.e database user, password of the database user, & database name.



Add OwnCloud Database Settings

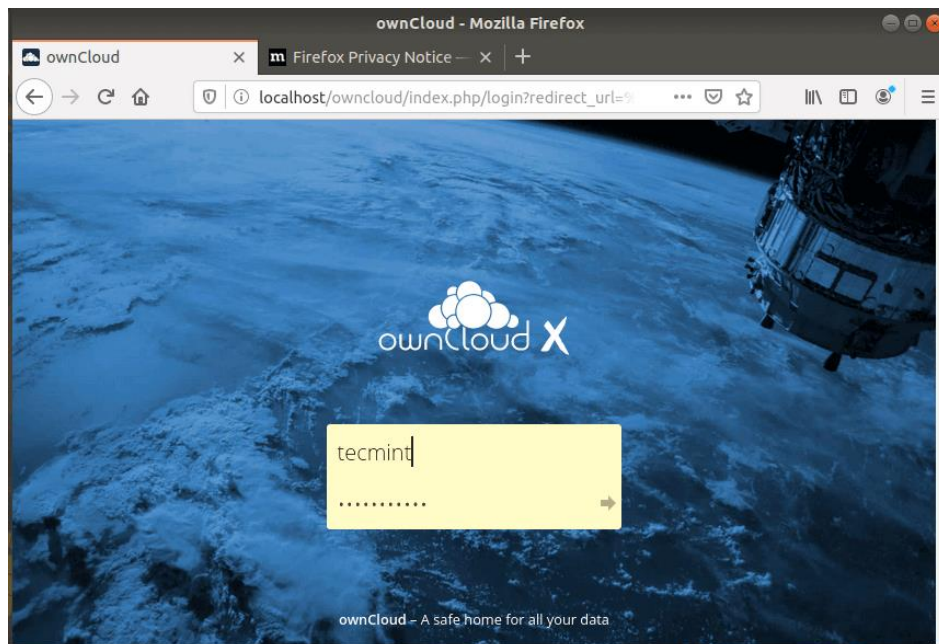
Finally, click **'Finish setup'** to wind up setting up Owncloud.



Finish OwnCloud Setup

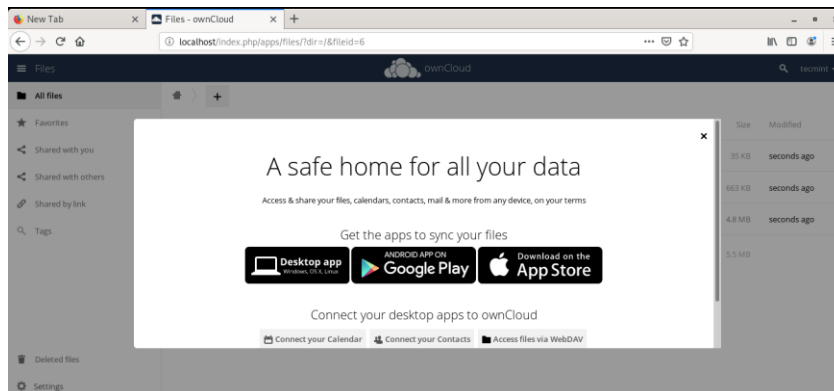


This takes you to the login screen as shown. Input the username and password defined earlier and hit ENTER.



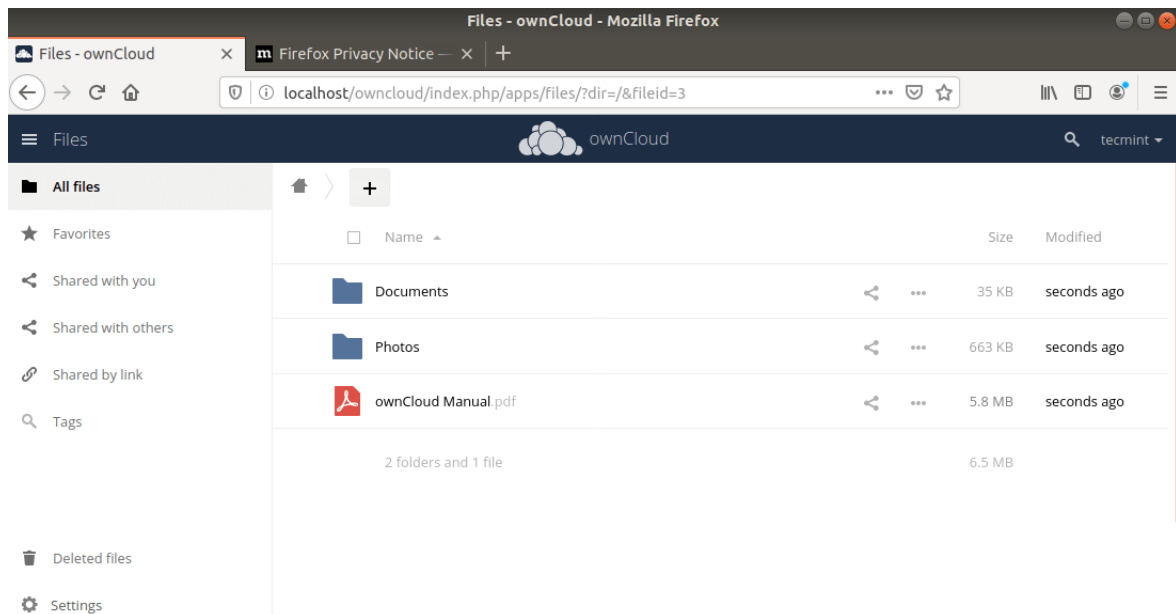
### OwnCloud Admin Login

A notification will be presented indicating other avenues that you can access OwnCloud from i.e iOS, Android & desktop App.



### OwnCloud Supported Platforms

Close the pop-up to access the dashboard as shown:

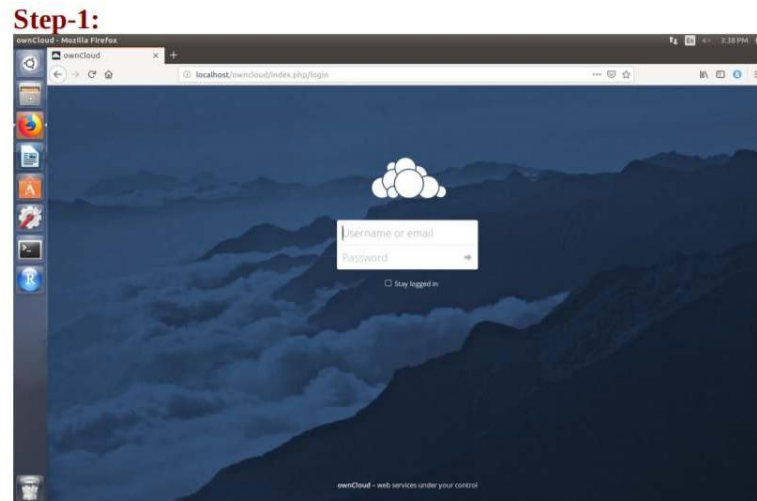


OwnCloud Dashboard

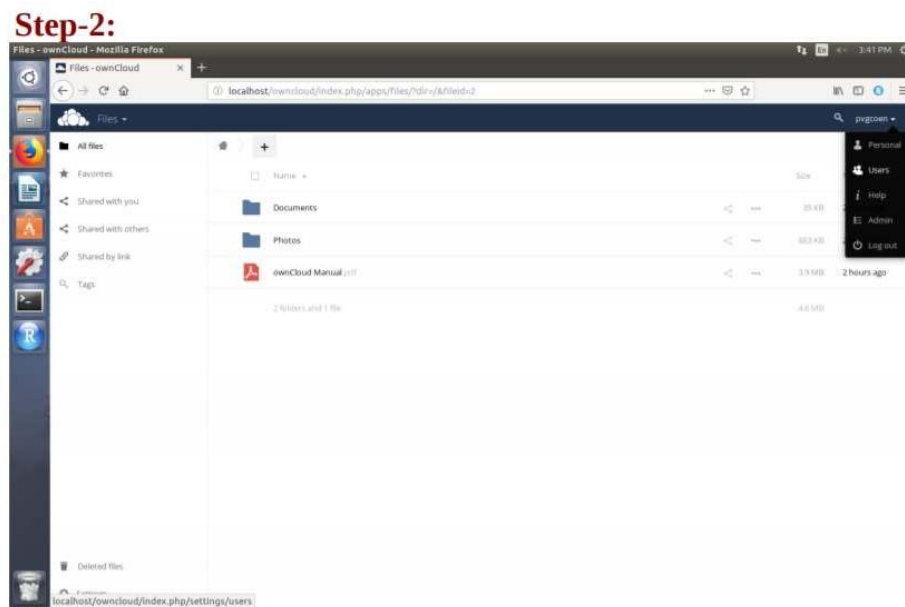
And that's it, guys! We have successfully installed the **OwnCloud** file sharing platform on **Ubuntu 18.04**.

#### **4. Write a Program to Create, Manage and groups User accounts in own Cloud by Installing Administrative Features.**

Step 1: Login into OwnCloud

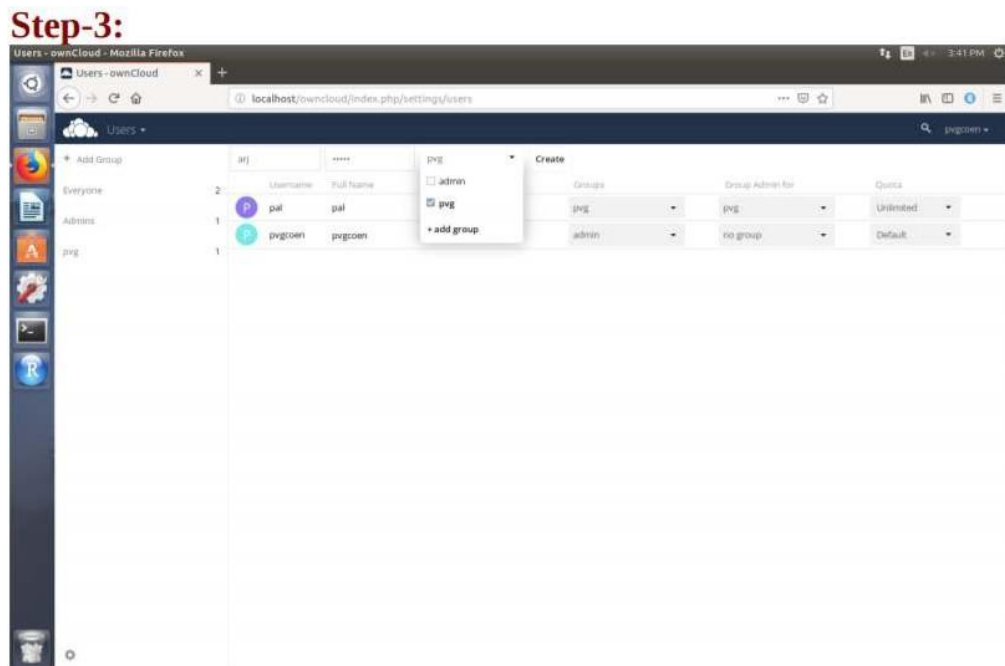


Step2: Click on Users



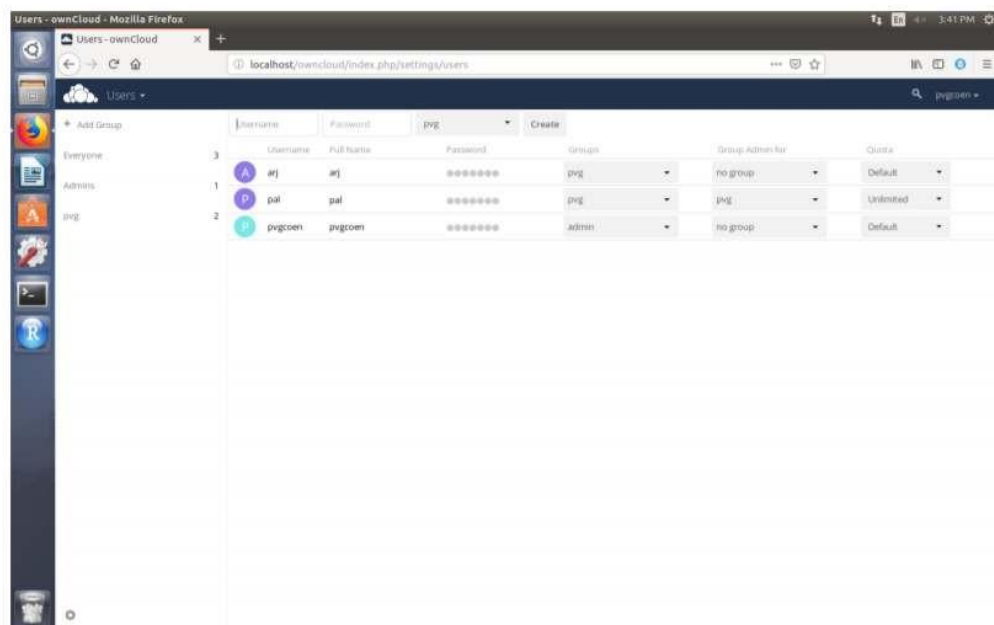


Step 3: Click in Create Group



Step 4: Set User Name and Password

### Step-4:



## **5. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim**

### **Step 1: Setting up the Prerequisites**

1. First of all we need to download the CloudSim and latest version of the Java Development Toolkit (JDK).

2. CloudSim requires a working Java installation. So, open up a terminal and run the following

```
1 sudo add-apt-repository ppa:webupd8team/java
2 sudo apt-get update && sudo apt-get install oracle-java8-installer
```

It will take some time to download and install so sit back and wait. Once it's done then we have to add the JAVA\_HOME to the Ubuntu environment. Run the following in a terminal to open up the */etc/environment* file.

```
1 sudo gedit /etc/environment
```

Now, append the following at the end of the file and save it:

```
JAVA_HOME="/usr/lib/jvm/java-8-oracle"
```

3. Now its time to install the CloudSim. Unpack the downloaded 'CloudSim-3.0.3.tar.gz' or 'CloudSim-3.0.3.zip' (let the name of the unpacked folder be 'cloudsim-3.0.3'). As you can see there is no makefile or install file inside the folder so it doesn't need to be compiled. Later if you want to remove the CloudSim, just remove the whole 'cloudsim-3.0.3' directory.

### **Step 2: Setting up the Environment**

Now comes the critical part, the most important part of the CloudSim setup is the setting up the paths to different classes and jar files correctly or you won't be able to run your programs

efficiently.

We need to set the 'CLASSPATH' variable which will contain the location of the class files and will be used by the CloudSim while executing an application. So we have to set two consecutive locations first one is the location of *gridsim.jar* file provided in the CloudSim and is used exclusively by the CloudSim applications and second one is the location where we have stored our programs.

We will set the CLASSPATH in the *.bashrc* file of the current user so open a terminal and run the following

```
1 sudo gedit /home/dhyan/.bashrc
```

Provide the password and add the following lines at the end of the opened file and save it.

```
CLASSPATH=".:/home/dhyan/Desktop/cloudsim-3.0.3/jars/*:  
/home/dhyan/Desktop/cloudsim-3.0.3/examples"  
export CLASSPATH
```

Now we need to reload the *.bashrc* file so close the all opened terminals (if any) and run the following

```
1 source ~/.bashrc
```

### **Step 3: Testing the Setup (Compiling and Executing a CloudSim Application)**

Finally now we can test whether our installation is successful or not. CloudSim includes some test example programs in the 'CloudSim\examples\gridsim\' folder that we can use to test our setup.

**1. Compiling a CloudSim program:** If you have followed this DIY then compiling a CloudSim program is pretty straightforward; the basic syntax for compilation is just similar to that of Java programs i.e. `javac filename.java` or `javac file_location/filename.java`. Let us compile the *Example2.java* included in 'CloudSim/examples/gridsim/example02/' folder. We will now run the following command in a new command prompt

```
1 javac /home/dhyan/Desktop/cloudsim-3.0.3/examples/org/cloudbus/cloudsim/examples/CloudSimExample1.java
```

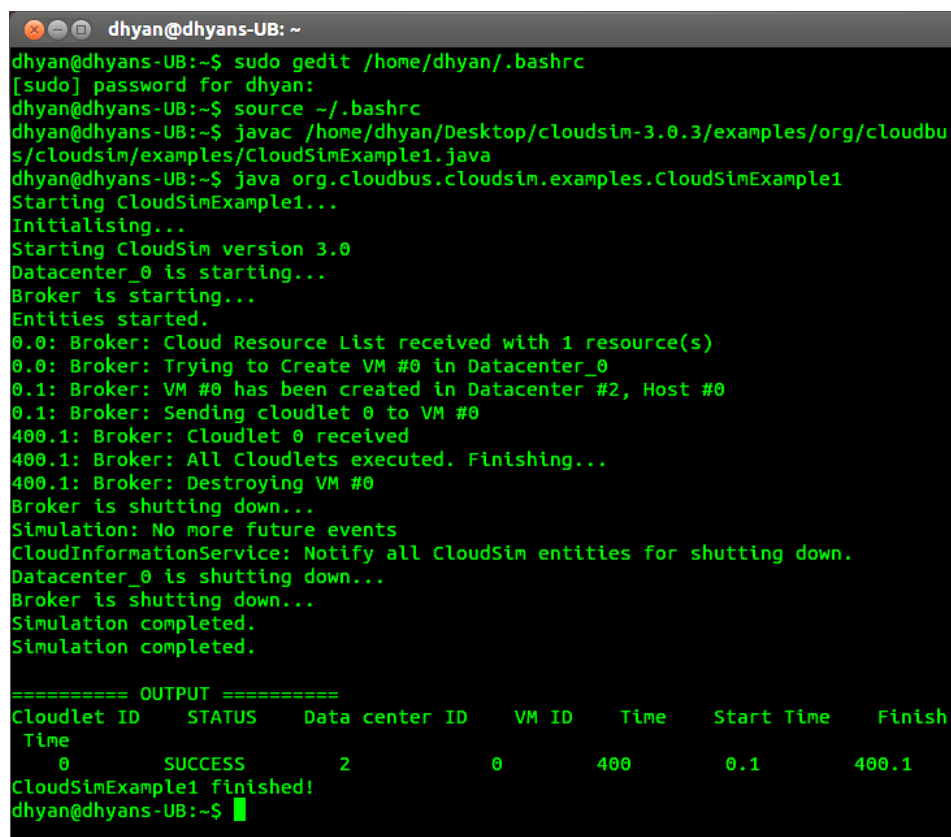
**2. Running the compiled program:** The syntax for running a compiled CloudSim program is similar to that of running a program in Java i.e. java filename. In our case we have to type (see image 1)

```
1 java org.cloudbus.cloudsim.examples.CloudSimExample1
```

OR if you want to save the output of your program to a file you can use the following

```
1 java org.cloudbus.cloudsim.examples.CloudSimExample1 > output.txt
```

**Note:** The examples given in the CloudSim uses the concept of packages hence it is advisable to go through the basics of packages in Java for a better understanding of the above stated commands.



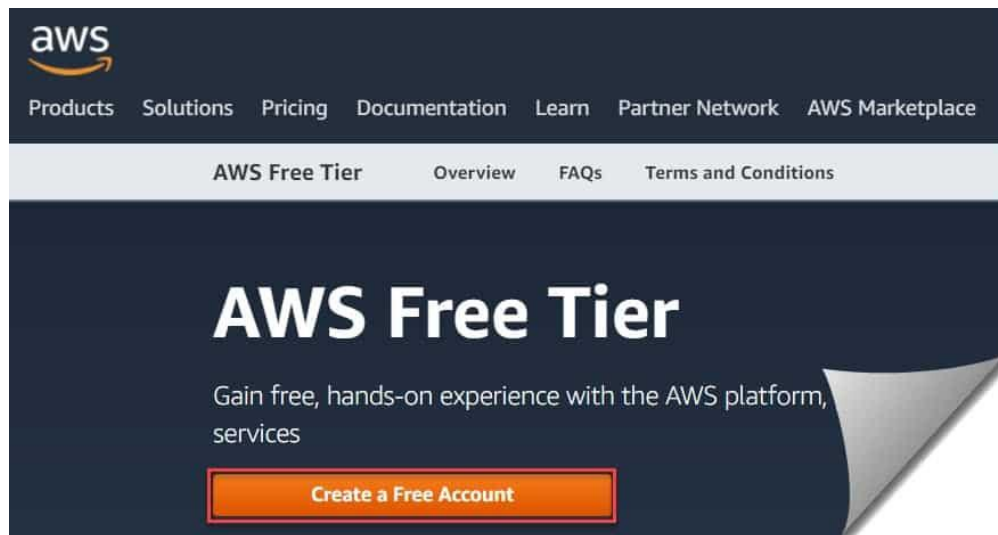
```
dhyan@dhyans-UB: ~
dhyan@dhyans-UB:~$ sudo gedit /home/dhyan/.bashrc
[sudo] password for dhyan:
dhyan@dhyans-UB:~$ source ~/.bashrc
dhyan@dhyans-UB:~$ javac /home/dhyan/Desktop/cloudsim-3.0.3/examples/org/cloudbus/cloudsim/examples/CloudSimExample1.java
dhyan@dhyans-UB:~$ java org.cloudbus.cloudsim.examples.CloudSimExample1
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish Time
0            SUCCESS      2              0      400     0.1         400.1
CloudSimExample1 finished!
dhyan@dhyans-UB:~$
```

## 6. Create AWS Free Trial Account and Create & connect to Amazon EC2 Machine

**Step.** Follow the below quick steps to **register for AWS free tier account** or create an aws account.

1. Open browser and navigate to the AWS signup Page.
2. Click on the Create a Free Account button as highlighted below.



3. On the Sign up for AWS page, provide the below details

- Email address: Provide a valid email address. Make sure you have not used the same email address before to register for an AWS account.
- Password: Provide a strong password.
- Confirm password: Reenter the same password for the confirmation.
- AWS account name: Provide a name for your AWS account. One point to note down here is that you can able to change the account name using the account settings page after the signup.

Finally, click on Continue (Step 1 of 5) button.

The image shows the 'Sign up for AWS' form. It has a title 'Sign up for AWS' and a sub-header 'Email address' with a note 'You will use this email address to sign in to your new AWS account.' Below this is a text input field containing 'rajurims86@gmail.com'. The next field is 'Password' with a masked input (dots). This is followed by a 'Confirm password' field, also masked. Below these is the 'AWS account name' field with a note 'Choose a name for your account. You can change this name in your account settings after you sign up.' The input field contains 'Azurelessons'. At the bottom of the form is an orange button labeled 'Continue (step 1 of 5)'. Below the button is a link that says 'Sign in to an existing AWS account'.

create an aws account

4. On the Contact Information section, provide the below details

- How do you plan to use AWS?: You can choose Personal or Business based on your need.
- Full Name: Provide your full name.
- Phone Number: You need to provide your phone number with your country code.
- Country or Region: Select your country from the dropdown.
- Address: You need to provide your complete address including your city, state, Postal Code, etc.
- Read and accept the terms and conditions of the AWS customer agreement.

Finally, click on Continue (Step 2 of 5) button to move to the next step.

The screenshot shows the AWS Free Tier offers and Contact Information section. The Free Tier offers section on the left lists three options: 'Always free' (Never expires), '12 months free' (Start from initial sign-up date), and 'Trials' (Start from service activation date). The Contact Information section on the right contains the following fields and options:


- How do you plan to use AWS?:** Radio buttons for 'Business - for your work, school, or organization' and 'Personal - for your own projects' (selected).
- Who should we contact about this account?:** (Empty field)
- Full Name:** Text input field containing 'Rajesh Kumar'.
- Phone Number:** Text input field containing '+918147'.
- Country or Region:** Dropdown menu showing 'India'.
- Address:** Text input field containing 'Flat-301'.
- City:** Text input field containing 'Bangalore'.
- State, Province, or Region:** Text input field containing 'Karnataka'.
- Postal Code:** Text input field containing '5'.
- Terms and Conditions:** A checkbox labeled 'I have read and agree to the terms of the AWS Customer Agreement' is checked.
- Continue (step 2 of 5):** A large orange button at the bottom right.

5. On the Billing Information section, provide the below details

- Credit or Debit card number: Provide your credit card number and make you have entered the correct one.
- Expiration date: You need to provide the expiration date of your credit card.
- Cardholder's name: Provide the name of the cardholder.
- CVV: Enter the correct CVV of your card.

- **Billing address:** You can choose the contact address that you have provided before or you can also add a new address by selecting the Use a new address radio button.
  - **Do You have a PAN?:** You can choose Yes and provide the PAN number or you can choose the No option and later you can add your PAN details on the tax settings page.
- Finally, click on Verify and Continue (step 3 of 5) button to move to the next step.


### Secure verification

 We will not charge for usage below AWS Free Tier limits. We temporarily hold INR 2 as a pending transaction for 3-5 days to verify your identity.

## Sign up for AWS

### Billing Information

Credit or Debit card number



AWS accepts all major credit and debit cards. To learn more about payment options, review our [FAQ](#).

Expiration date

Cardholder's name

CVV

Billing address

☒ Use my contact address

IN

☐ Use a new address

Do you have a PAN?

Permanent Account Number (PAN) is a ten-digit alphanumeric number issued by the Indian Income Tax Department. This 10-digit number is printed on the front of your PAN card.

☐ Yes

☒ No

You can go on the Tax Settings Page on Billing and Cost Management Console to update your PAN information.

**Verify and Continue (step 3 of 5)**

6. Now enter the OTP that you have received on your mobile for a transaction of 2 rupees and then click on the Next button. For me, it is 2 rupees as I have chosen India as my country. Based on your country you will get a very minimal transaction. Remember that this amount amazon will hold temporarily just to verify your identity and it might take 3 to 5 days to verify your identity.

Card Number

XXXX XXXX XXXX 5462

Merchant

AMAZON INTERNET SERVICES

Amount

INR 2.00

PLEASE AUTHENTICATE THE TRANSACTION USING:


**OTP**  
(One Time Password)


An OTP has been sent to your registered mobile

Enter OTP

Click here if you have not received an OTP

**NEXT** CANCEL

 Citigroup Privacy Promise Terms, Conditions, Caveats and small print  
Copyright © 2015, Citigroup



7. Now is the time to verify your Phone on the Confirm your Identity section. Provide the below details.

- How should we send you the verification code?: Select the text message radio button. You can also choose the Phone call option.
- Country or Region Code: Select your country or region code.
- Cell Phone Number: Provide the number of your cell phone.
- Security Check: Type the Exact captcha.

Click on the Send SMS button to receive the SMS on your mobile.

8. Enter the code you have received and then click on the Verify Code button.

9. It will show you now that “Your identity has been verified successfully.” Then click on the Continue button.

10. Now, on the next window you will see three plans.

- Basic Plan (Free)
- Developer Plan
- Business Plan

Select the plan based on your need. Remember that basic plan is free of cost and check the price before selecting the other two plans.

11. Finally, you will see now the Registration Confirmation page. It might take 30 minutes to 1 hour for the activation of your AWS account. You will receive an email confirmation for your AWS account activation.


Step: Follow the following steps to **Create & connect to Amazon EC2 Machine**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the EC2 console dashboard, in the **Launch instance** box, choose **Launch instance**, and then choose **Launch instance** from the options that appear.
3. Under **Name and tags**, for **Name**, enter a descriptive name for your instance.
4. Under **Application and OS Images (Amazon Machine Image)**, do the following:
  - a. Choose **Quick Start**, and then choose Amazon Linux. This is the operating system (OS) for your instance.
  - b. From **Amazon Machine Image (AMI)**, select an HVM version of Amazon Linux 2. Notice that these AMIs are marked **Free tier eligible**. An *Amazon Machine Image (AMI)* is a basic configuration that serves as a template for your instance.
5. Under **Instance type**, from the **Instance type** list, you can select the hardware configuration for your instance. Choose the t2.micro instance type, which is selected by default. The t2.micro instance type is eligible for the free tier. In Regions where t2.micro is unavailable, you can use a t3.micro instance under the free tier. For more information.
6. Under **Key pair (login)**, for **Key pair name**, choose the key pair that you created when getting set up.

### Warning



Do not choose **Proceed without a key pair (Not recommended)**. If you launch your instance without a key pair, then you can't connect to it.

7. Next to **Network settings**, choose **Edit**. For **Security group name**, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:
  - a. Choose **Select existing security group**.
  - b. From **Common security groups**, choose your security group from the list of existing security groups.
8. Keep the default selections for the other configuration settings for your instance.
9. Review a summary of your instance configuration in the **Summary** panel, and when you're ready, choose **Launch instance**.
10. A confirmation page lets you know that your instance is launching. Choose **View all instances** to close the confirmation page and return to the console.
11. On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is pending. After the instance starts, its state changes to running and it receives a public DNS name. If the **Public IPv4 DNS** column is hidden, choose the settings icon (  ) in the top-right corner, toggle on **Public IPv4 DNS**, and choose **Confirm**.
12. It can take a few minutes for the instance to be ready for you to connect to it. Check that your instance has passed its status checks; you can view this information in the **Status check** column.

### Connect using the Amazon EC2 console (browser-based client)

You can connect to an instance using the Amazon EC2 console (browser-based client) by selecting the instance from the console and choosing to connect using EC2 Instance Connect. Instance Connect handles the permissions and provides a successful connection.

### To connect to your instance using the browser-based client from the Amazon EC2 console

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
- In the navigation pane, choose **Instances**.
- Select the instance and choose **Connect**.
- Choose **EC2 Instance Connect**.
- Verify the user name and choose **Connect** to open a terminal window.

## 7. Create S3 Bucket in AWS, Upload & Access a File, And Host a Simple Website

### Step 1: Create a bucket

The following instructions provide an overview of how to create your buckets for website hosting. For detailed, step-by-step instructions on creating a bucket.

#### To create a bucket

1. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Choose **Create bucket**.
3. Enter the **Bucket name** (for example, **example.com**).
4. Choose the Region where you want to create the bucket.

Choose a Region that is geographically close to you to minimize latency and costs, or to address regulatory requirements. The Region that you choose determines your Amazon S3 website endpoint. For more information.

5. To accept the default settings and create the bucket, choose **Create**.
- 

### Step 2: Enable static website hosting

After you create a bucket, you can enable static website hosting for your bucket. You can create a new bucket or use an existing bucket.

#### To enable static website hosting

1. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In the **Buckets** list, choose the name of the bucket that you want to enable static website hosting for.
3. Choose **Properties**.
4. Under **Static website hosting**, choose **Edit**.
5. Choose **Use this bucket to host a website**.
6. Under **Static website hosting**, choose **Enable**.
7. In **Index document**, enter the file name of the index document, typically index.html.

The index document name is case sensitive and must exactly match the file name of the HTML index document that you plan to upload to your S3 bucket. When you configure a bucket for website hosting, you must specify an index document. Amazon S3 returns this index document when requests are made to the root domain or any of the subfolders. For more information.

8. To provide your own custom error document for 4XX class errors, in **Error document**, enter the custom error document file name.

The error document name is case sensitive and must exactly match the file name of the HTML error document that you plan to upload to your S3 bucket. If you don't specify a custom error document and an error occurs, Amazon S3 returns a default HTML error document. For more information.

9. (Optional) If you want to specify advanced redirection rules, in **Redirection rules**, enter JSON to describe the rules.

For example, you can conditionally route requests according to specific object key names or prefixes in the request. For more information.

10. Choose **Save changes**.

Amazon S3 enables static website hosting for your bucket. At the bottom of the page, under **Static website hosting**, you see the website endpoint for your bucket.

11. Under **Static website hosting**, note the **Endpoint**.

The **Endpoint** is the Amazon S3 website endpoint for your bucket. After you finish configuring your bucket as a static website, you can use this endpoint to test your website.


---

### Step 3: Edit Block Public Access settings

By default, Amazon S3 blocks public access to your account and buckets. If you want to use a bucket to host a static website, you can use these steps to edit your block public access settings.

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Choose the name of the bucket that you have configured as a static website.
3. Choose Permissions.
4. Under Block public access (bucket settings), choose Edit.
5. Clear Block all public access, and choose Save changes.

### Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#) 



#### Account settings for Block Public Access are currently turned on

[Account settings for Block Public Access](#) that are enabled apply even if they are disabled for this bucket.

☐ **Block *all* public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.

☐ **Block public access to buckets and objects granted through *new* public bucket or access point policies**

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Amazon S3 turns off Block Public Access settings for your bucket. To create a public, static website, you might also have to [edit the Block Public Access settings](#) for your account before adding a bucket policy. If account settings for Block Public Access are currently turned on, you see a note under **Block public access (bucket settings)**.

---

## Step 4: Add a bucket policy that makes your bucket content publicly available

After you edit S3 Block Public Access settings, you can add a bucket policy to grant public read access to your bucket. When you grant public read access, anyone on the internet can access your bucket.

### Important

The following policy is an example only and allows full access to the contents of your bucket. Before you proceed with this step to ensure that you understand the best practices for securing the files in your S3 bucket and risks involved in granting public access.

1. Under **Buckets**, choose the name of your bucket.
2. Choose **Permissions**.
3. Under **Bucket Policy**, choose **Edit**.
4. To grant public read access for your website, copy the following bucket policy, and paste it in the **Bucket policy editor**.

5. {

6. "Version": "2012-10-17",

```
7.  "Statement": [  
8.    {  
9.      "Sid": "PublicReadGetObject",  
10.     "Effect": "Allow",  
11.     "Principal": "*",  
12.     "Action": [  
13.       "s3:GetObject"  
14.     ],  
15.     "Resource": [  
16.       "arn:aws:s3:::Bucket-Name/*"  
17.     ]  
18.   }  
19. ]  
  
}
```

20. Update the Resource to your bucket name.

In the preceding example bucket policy, *Bucket-Name* is a placeholder for the bucket name. To use this bucket policy with your own bucket, you must update this name to match your bucket name.

21. Choose **Save changes**.

A message appears indicating that the bucket policy has been successfully added.

If you see an error that says Policy has invalid resource, confirm that the bucket name in the bucket policy matches your bucket name. For information about adding a bucket policy.

If you get an error message and cannot save the bucket policy, check your account and bucket Block Public Access settings to confirm that you allow public access to the bucket.

---

## Step 5: Configure an index document

When you enable static website hosting for your bucket, you enter the name of the index document (for example, **index.html**). After you enable static website hosting for the bucket, you upload an HTML file with this index document name to your bucket.

### To configure the index document

1. Create an index.html file.

If you don't have an index.html file, you can use the following HTML to create one:

```
<html xmlns="http://www.w3.org/1999/xhtml" >

<head>

  <title>My Website Home Page</title>

</head>

<body>

  <h1>Welcome to my website</h1>

  <p>Now hosted on Amazon S3!</p>

</body>

</html>
```

2. Save the index file locally.  
The index document file name must exactly match the index document name that you enter in the **Static website hosting** dialog box. The index document name is case sensitive. For example, if you enter index.html for the **Index document** name in the **Static website hosting** dialog box, your index document file name must also be index.html and not Index.html.
3. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
4. In the **Buckets** list, choose the name of the bucket that you want to use to host a static website.
5. Enable static website hosting for your bucket, and enter the exact name of your index document (for example, index.html). For more information.
6. After enabling static website hosting, proceed to step 6.

7. To upload the index document to your bucket, do one of the following:
    - Drag and drop the index file into the console bucket listing.
    - Choose **Upload**, and follow the prompts to choose and upload the index file.For step-by-step instructions.
  8. (Optional) Upload other website content to your bucket.
- 

## Step 6: Configure an error document

When you enable static website hosting for your bucket, you enter the name of the error document (for example, **404.html**). After you enable static website hosting for the bucket, you upload an HTML file with this error document name to your bucket.

### To configure an error document

1. Create an error document, for example 404.html.
  2. Save the error document file locally.

The error document name is case sensitive and must exactly match the name that you enter when you enable static website hosting. For example, if you enter 404.html for the **Error document** name in the **Static website hosting** dialog box, your error document file name must also be 404.html.
  3. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
  4. In the **Buckets** list, choose the name of the bucket that you want to use to host a static website.
  5. Enable static website hosting for your bucket, and enter the exact name of your error document (for example, 404.html). For more information.

After enabling static website hosting, proceed to step 6.
  6. To upload the error document to your bucket, do one of the following:
    - Drag and drop the error document file into the console bucket listing.
    - Choose **Upload**, and follow the prompts to choose and upload the index file.For step-by-step instructions.
- 

## Step 7: Test your website endpoint

After you configure static website hosting for your bucket, you can test your website endpoint.

1. Under **Buckets**, choose the name of your bucket.
2. Choose **Properties**.

3. At the bottom of the page, under **Static website hosting**, choose your **Bucket website endpoint**.

Your index document opens in a separate browser window.

You now have a website hosted on Amazon S3. This website is available at the Amazon S3 website endpoint. However, you might have a domain, such as example.com, that you want to use to serve the content from the website you created. You might also want to use Amazon S3 root domain support to serve requests for both `http://www.example.com` and `http://example.com`. This requires additional steps. For an example.

---

## Step 8: Clean up

If you created your static website only as a learning exercise, delete the AWS resources that you allocated so that you no longer accrue charges. After you delete your AWS resources, your website is no longer available. For more information.



## 8. S3 Cross-Region Replication

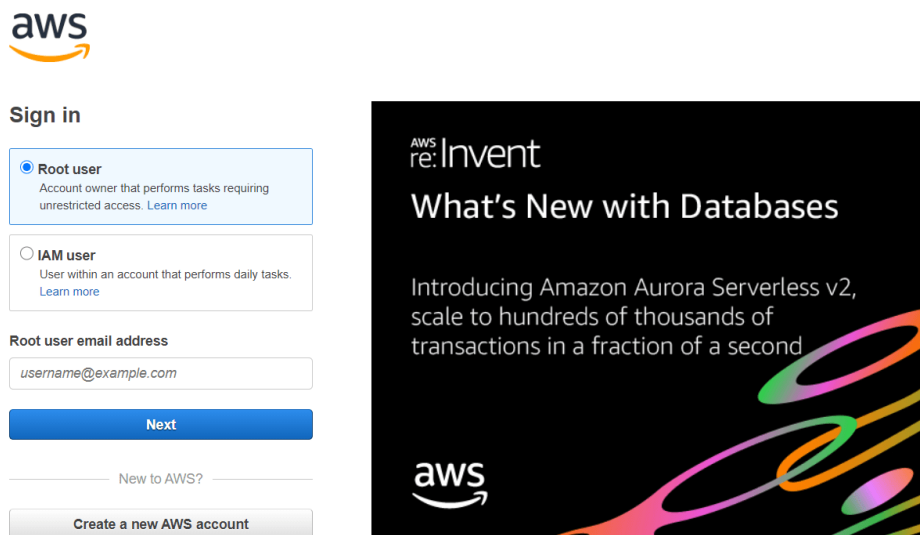
### Steps to Set Up Cross Region Replication in S3

You can implement Cross Region Replication in S3 using the following steps:

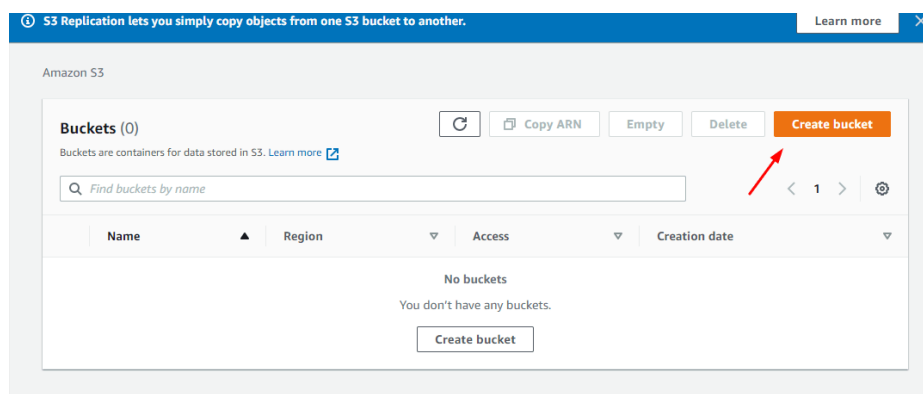
- [Step 1: Creating Buckets in S3](#)
- [Step 2: Creating an IAM User](#)
- [Step 3: Configuring the Bucket Policy in S3](#)
- [Step 4: Initializing Cross Region Replication in S3](#)

### Step 1: Creating Buckets in S3

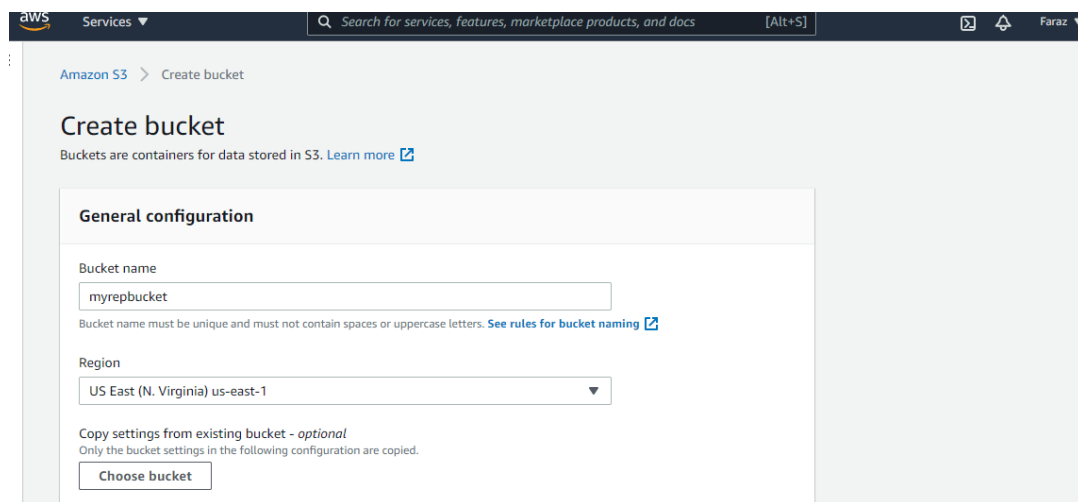
To start replicating data from your desired S3 bucket, you first need to log into the AWS management console for S3. To do this, go to the [official website](#) of AWS S3's management console and enter your credentials such as your username and password.



Once you've logged in, S3 homepage will now open up on your screen, where you need to click on the create a bucket option, found in the top right corner of your screen:



The “create a bucket” window will now open up on your screen, where you need to configure your new S3 bucket by providing details such as a unique name for your bucket and its region.



aws Services Search for services, features, marketplace products, and docs [Alt+S] Faraz

Amazon S3 > Create bucket

### Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

**General configuration**

Bucket name  
myrepbucket

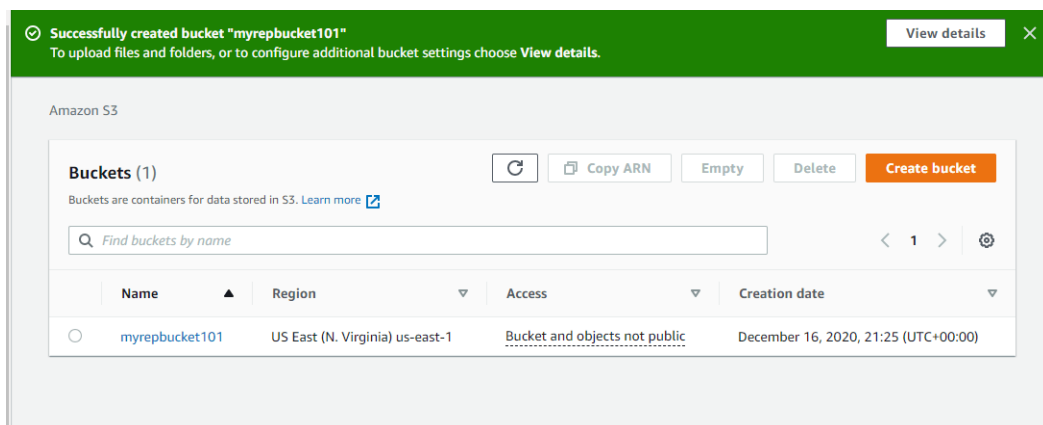
Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

Region  
US East (N. Virginia) us-east-1

Copy settings from existing bucket - optional  
Only the bucket settings in the following configuration are copied.

Choose bucket

You will now be able to see the newly created S3 bucket in the bucket details section as follows:



Successfully created bucket "myrepbucket101"  
To upload files and folders, or to configure additional bucket settings choose [View details](#).

Amazon S3

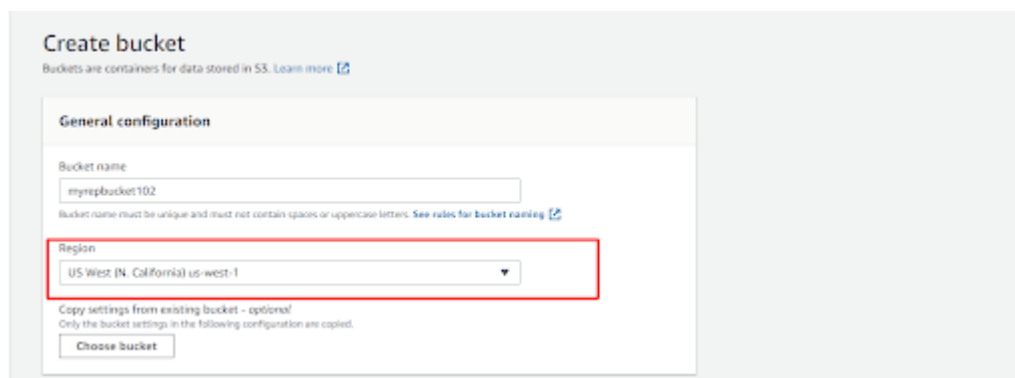
**Buckets (1)** [Refresh] [Copy ARN] [Empty] [Delete] **Create bucket**

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

	Name	Region	Access	Creation date
<input type="radio"/>	myrepbucket101	US East (N. Virginia) us-east-1	Bucket and objects not public	December 16, 2020, 21:25 (UTC+00:00)

To set up Cross Region Replication successfully, creating just one S3 bucket isn't enough and hence, you now need to set up another bucket in a different region.



Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

**General configuration**

Bucket name  
myrepbucket102

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

Region  
US West (N. California) us-west-1

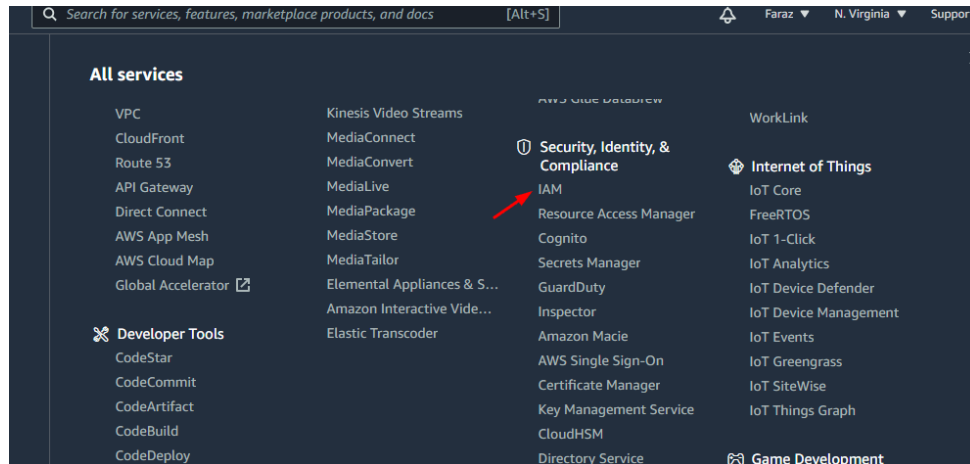
Copy settings from existing bucket - optional  
Only the bucket settings in the following configuration are copied.

Choose bucket

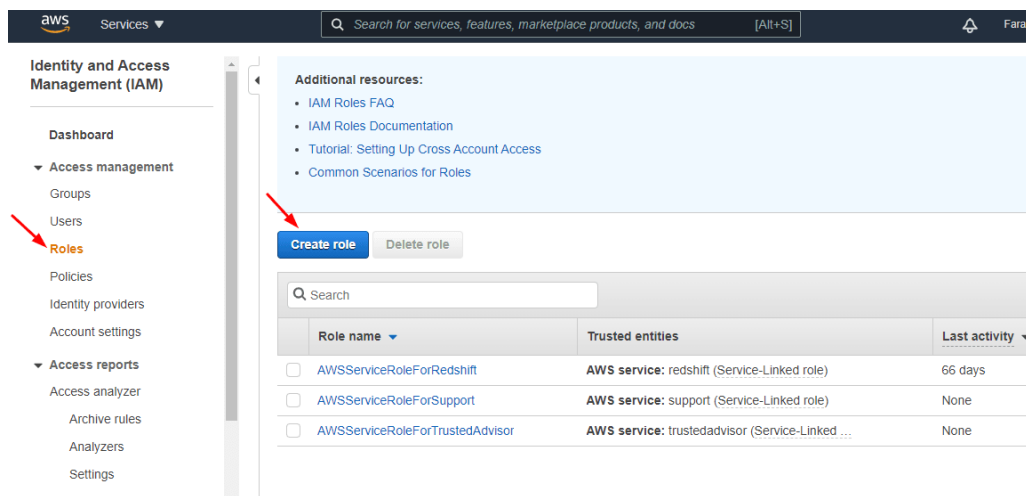
This is how you can create buckets in S3 to start setting up Cross Region Replication.

## Step 2: Creating an IAM User

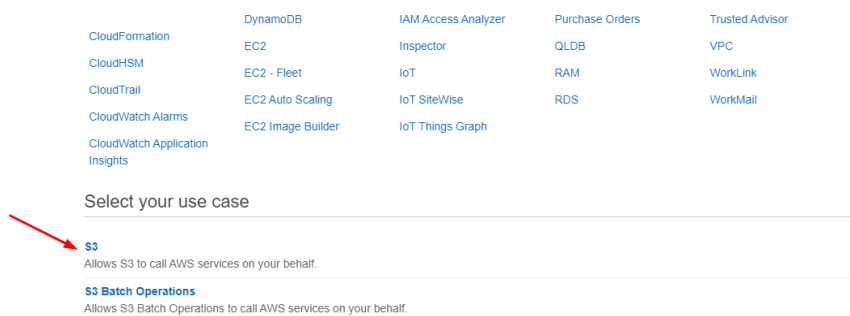
With your S3 buckets now ready, you now need to create an [IAM user](#). To do this, click on the IAM option, found in the main menu.



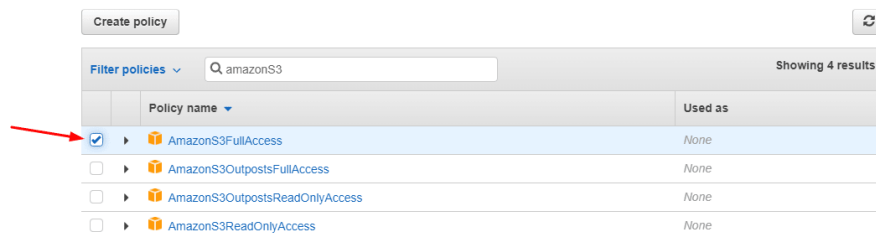
The IAM page will now open up on your screen, where you need to click on the roles option from the panel on the left and then click on the create role option.



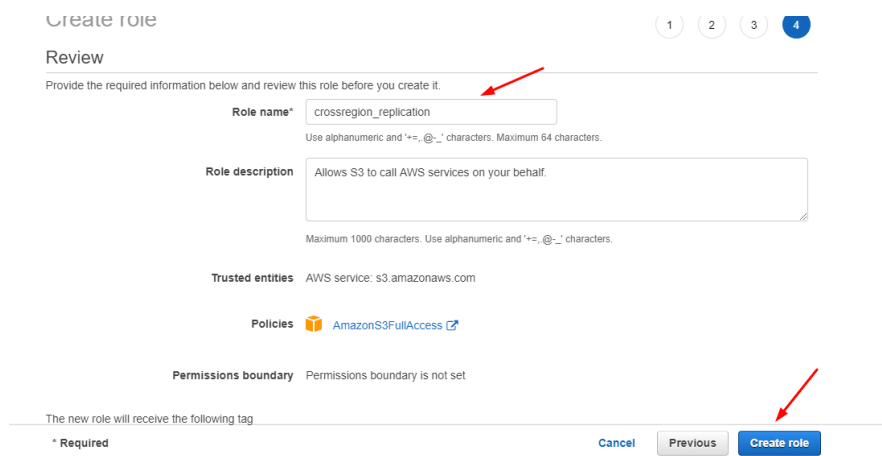
You now need to select S3 as your desired service and then choose “S3: Allow S3 to call AWS services on your behalf” as your use case.



Once you've selected the right use case and service, you now need to choose the role policy. To do this, use the search bar and search for "AmazonS3FullAccess" and select it:

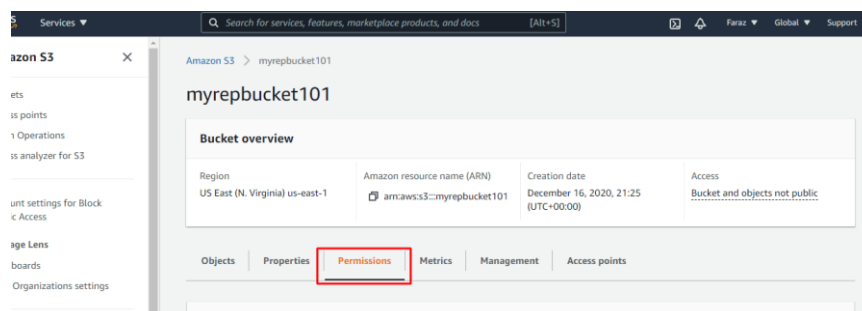


With your IAM role now ready and configured, the "review" window will now open up on your screen, where you'll be able to find all necessary information about your role. To complete the setup, click on the create role option, present in the bottom right corner of your screen.

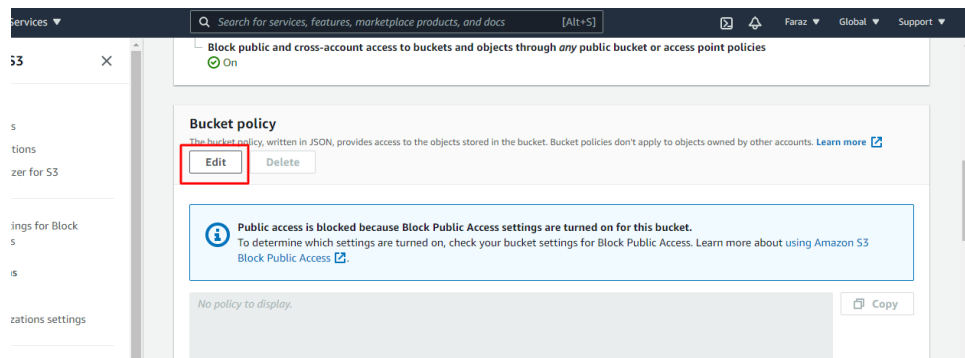


### Step 3: Configuring the Bucket Policy in S3

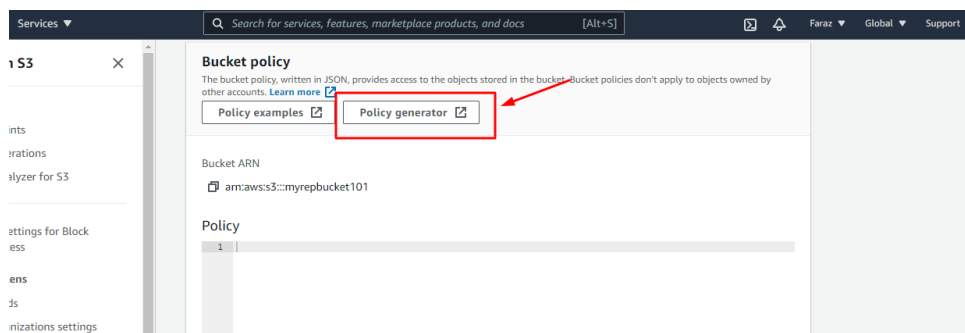
With your IAM role now set up, you now need to define the bucket policy that will help outline and decide the actions a user can perform. To configure the bucket policy, select the desired S3 bucket and click on the permissions option.



Locate the bucket policy section in the permissions tab and then click on the edit option as follows:



The bucket policy page will now open up on your screen, where you need to click on the [policy generator](#) option. In case you want to learn more about the AWS policy generator.



Once you’ve clicked on the policy generator option, the AWS policy generator window will now open up, where you need to choose the bucket policy. To do this, click on the policy drop-down list & select the “S3 Bucket Policy” option, and then click on the add statement option.

#### AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are [sample policies](#).

##### Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy SQS Queue Policy

##### Step 2: Add Statement(s)

A statement is the formal description of a single permission. See [a description of elements](#) that you can use in statements.

Effect ☒ Allow ☐ Deny

Principal

Use a comma to separate multiple values.

AWS Service Amazon SQS ☐ All Services ("\*")

Use multiple statements to add permissions for more than one service.

Actions -- Select Actions -- ☐ All Actions ("\*")

Amazon Resource Name (ARN)

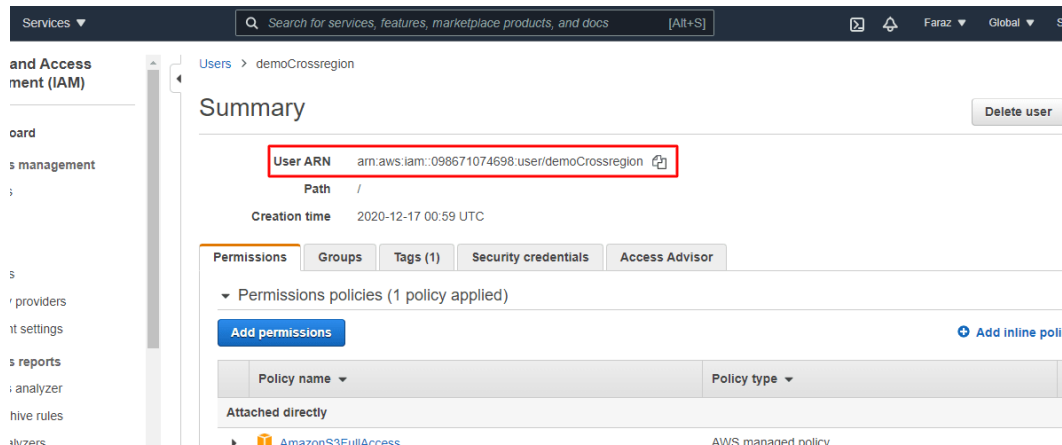
ARN should follow the following format: arn:aws:sqs:<region>:<account\_ID>:<queue\_name>.

Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement No Action selected. You must select at least one Action

You will now be able to find the IAM user [ARN value](#) in the summary section as follows:



Once you've configured the user ARN, you now need to set up the bucket ARN value. To configure the bucket policy and ARN, add three operations, namely, Get Object, Put Object & Delete Object and then click on the add statement option.

#### Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy S3 Bucket Policy

#### Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a [description of elements](#) that you can use in statements.

Effect ☒ Allow ☐ Deny

Principal

Use a comma to separate multiple values.

AWS Service Amazon S3 ☐ All Services ("\*")

Use multiple statements to add permissions for more than one service.

Actions 3 Action(s) Selected ☐ All Actions ("\*")

Amazon Resource Name (ARN)

ARN should follow the following format: arn:aws:s3:::<bucket\_name>/<key\_name>. Use a comma to separate multiple values.

[Add Conditions \(Optional\)](#)

[Add Statement](#)

The bucket policy statement will now appear on your screen as follows:

You added the following statements. Click the button below to Generate a policy.

Principal(s)	Effect	Action	Resource	Conditions
arn:aws:iam::098671074698:user/demoCrossregion	Allow	<ul style="list-style-type: none"><li>s3:DeleteObject</li><li>s3:GetObject</li><li>s3:PutObject</li></ul>	arn:aws:s3:::myrepbucket101	None

With your bucket statement now ready, click on the generate policy button. The newly created policy will now appear on your screen as follows:

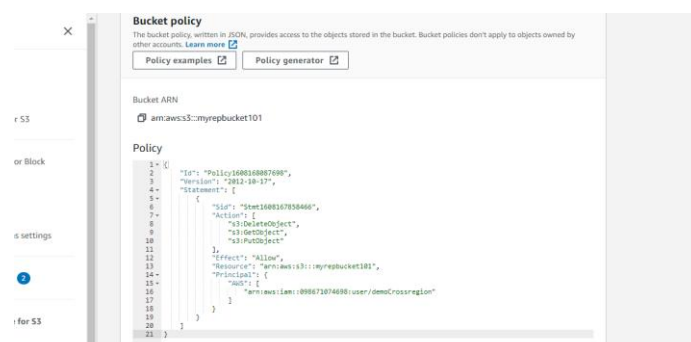


```

{
  "Id": "Policy1608168001400",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1608167858466",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::myrepbucket101",
      "Principal": {
        "AWS": [
          "arn:aws:iam::098671074698:user/demoCrossregion"
        ]
      }
    }
  ]
}

```

Copy the bucket policy and add it to your bucket policy list as follows:



You now need to repeat the same process for your second bucket.

```
{
  "Id": "Policy1608169434646",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1608169432435",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::myrepbucket102",
      "Principal": {
        "AWS": [
          "arn:aws:iam::098671074698:role/cross_region_demo1"
        ]
      }
    }
  ]
}
```

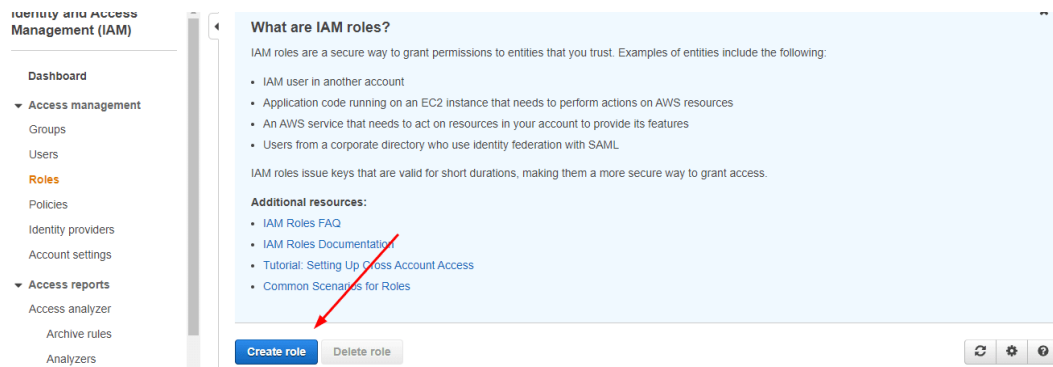
This is how you can set up the bucket policy in S3 to set up Cross Region Replication.

#### **Step 4: Initializing Cross Region Replication in S3**

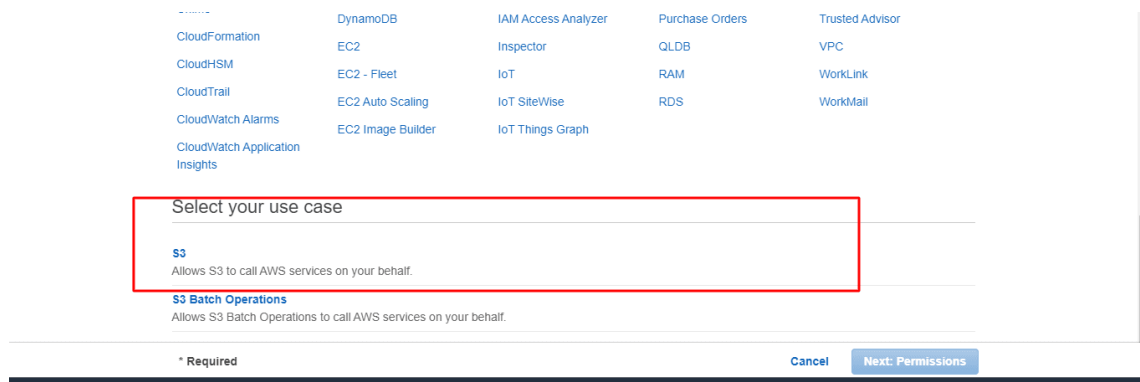
Once you've created your S3 buckets and have configured their policies, you can now perform a Cross Region Replication for your data in S3. To do this, you'll first have to create



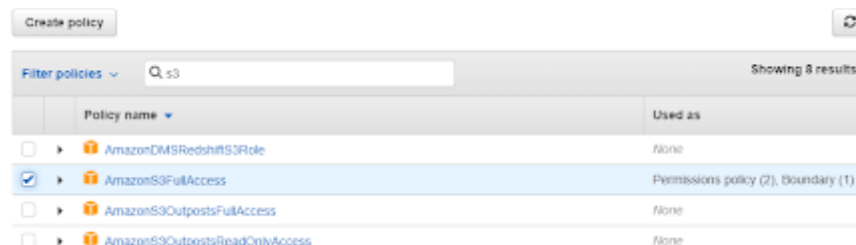
an IAM role for the user. To set up the IAM role, go to the roles page and click on the create role option present in the bottom of your screen:



Once you've clicked on it, you now need to select the use case for your IAM role as follows:



With your use case now set up, select the role policy permission as AmazonS3FullAccess.



Once you've made all the necessary configurations, the IAM role review page will open up on your screen, where you need to provide a unique name for your IAM role and then click on create. With your new IAM role in place, the bucket policies for both bucket 1 & 2 will get modified as follows:

### Bucket 1 Policy:

```
{
```

```
"Id": "Policy1608168001400",
```

```
"Version": "2012-10-17",
```

```
"Statement": [
```

```
{
  "Sid": "Stmt1608167858466",
  "Action": [
    "s3:DeleteObject",
    "s3:GetObject",
    "s3:PutObject"
  ],
  "Effect": "Allow",
  "Resource": "arn:aws:s3:::myrepbucket101",
  "Principal": {
    "AWS": [
      "arn:aws:iam::098671074698:user/demoCrossregion"
    ]
  }
}

{
  "Id": "Policy1608169434646",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1608169432435",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:PutObject"
      ]
    }
  ]
}
```

```
],
"Effect": "Allow",
"Resource": "arn:aws:s3:::myrepbucket102",
"Principal": {
  "AWS": [
    "arn:aws:iam::098671074698:role/cross_region_demo1"
  ]
}
}
```

**Bucket 2 Policy:**

```
{
  "Id": "Policy1608168001400",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1608167858466",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::myrepbucket101",
      "Principal": {
        "AWS": [
```

```
        "arn:aws:iam::098671074698:user/demoCrossregion"
    ]
}
}
]
}
{
    "Id": "Policy1608169434646",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1608169432435",
            "Action": [
                "s3:DeleteObject",
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::myrepbucket102",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::098671074698:role/cross_region_demo1"
                ]
            }
        }
    ]
}
```

To initialize the Cross Region Replication, click on the management option, present in the bucket details section and enable [bucket versioning](#) for both buckets.

**Create replication rule**

⚠ Replication requires versioning to be enabled for the source bucket. Enable object versioning on this bucket to continue creating the replication rule. [Enable Bucket Versioning](#)

**Replication rule configuration**

Replication rule name  
  
Up to 255 characters.

Status  
Choose whether the rule will be enabled or disabled when created.  
☒ Enabled

With bucket versioning now enabled, you now need to provide the name of your destination bucket as follows:

**Destination**

Destination  
You can replicate objects across buckets in different AWS Regions (Cross-Region Replication) or you can replicate objects across buckets in the same AWS Region (Same-Region Replication). You can also specify a different bucket for each rule in the configuration. [Learn more](#) or see [Amazon S3 pricing](#)

☒ Choose a bucket in this account  
☐ Specify a bucket in another account

Bucket name  
Choose the bucket that will receive replicated objects.  
 [Browse S3](#)

⚠ Replication requires versioning to be enabled for the destination bucket. Enable object versioning on this bucket to continue creating the replication rule or select a different bucket. [Enable bucket versioning](#)

Destination Region  
US East (N. Virginia) us-east-1

Now, click on the IAM role drop-down list and select the IAM role you created.

**IAM role**

IAM role  
 [Refresh](#) [View](#)

**Encryption**

Once you've selected the IAM role, click on the save option to bring the changes into effect. You now need to perform the same operation for your second bucket.

⊙ Replication configuration successfully updated  
Press the refresh button if changes to the configuration are not displayed.

Amazon S3 > myrepbucket102 > Replication rules

**Replication rules**  
Replication enables automatic and asynchronous copying of objects across buckets in the same or different AWS Regions. A replication configuration is a set of rules that define what options should be applied to a group of objects during replication.

**Replication configuration settings**  
Configuration settings affect all replication rules in the bucket. [Edit](#)

Source bucket myrepbucket102	IAM role cross_region_demo1
Source Region US West (N. California) us-west-1	

You can now verify the success of the replication process by checking the status of both buckets. The original bucket will now have a status value as “Completed” as follows:

<b>Etag</b>	098f6bcd4621d373cade4e832627b4f6
<b>Storage class</b>	Standard
<b>Server-side encryption</b>	AES-256
<b>Replication status</b>	COMPLETED
<b>Size</b>	

The replica bucket will now have the status value as “Replica” as follows:

<b>Etag</b>	098f6bcd4621d373cade4e832627b4f6
<b>Storage class</b>	Standard
<b>Server-side encryption</b>	AES-256
<b>Replication status</b>	REPLICA
<b>Size</b>	

This is how you can set up Cross Region Replication in S3

## 9. Create & Manage EBS Volumes & Snapshots

### To create a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Snapshots, Create snapshot**.
3. For **Resource type**, choose **Volume**.
4. For **Volume ID**, select the volume from which to create the snapshot.  
The **Encryption** field indicates the selected volume's encryption status. If the selected volume is encrypted, the snapshot is automatically encrypted using the same KMS key. If the selected volume is unencrypted, the snapshot is not encrypted.
5. (Optional) For **Description**, enter a brief description for the snapshot.
6. (Optional) To assign custom tags to the snapshot, in the **Tags** section, choose **Add tag**, and then enter the key-value pair. You can add up to 50 tags.
7. Choose **Create snapshot**.

You can copy snapshots, share snapshots, and create volumes from snapshots

### To copy a snapshot using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Snapshots**.
3. Select the snapshot to copy, and then choose **Actions, Copy snapshot**.
4. For **Description**, enter a brief description for the snapshot copy.

By default, the description includes information about the source snapshot so that you can identify a copy from the original. You can change this description as needed.

5. For **Destination Region**, select the Region in which to create the snapshot copy.
6. Specify the encryption status for the snapshot copy.

If the source snapshot is encrypted, or if your account is enabled for [encryption by default](#), then the snapshot copy is automatically encrypted and you can't change its encryption status.

If the source snapshot is unencrypted and your account is not enabled for encryption by default, encryption is optional. To encrypt the snapshot copy, for **Encryption**, select **Encrypt this snapshot**. Then, for **KMS key**, select the KMS key to use to encrypt the snapshot in the destination Region.

7. Choose **Copy snapshot**.

## To share a snapshot

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Snapshots**.
3. Select the snapshot to share, and then choose **Actions, Modify permissions**.
4. Specify the snapshot's permissions. *Current setting* indicates the snapshot's current sharing permissions.
  - To share the snapshot publicly with all AWS accounts, choose **Public**.
  - To share the snapshot privately with specific AWS accounts, choose **Private**. Then, in the **Sharing accounts** section, choose **Add account**, and enter the 12-digit account ID (without hyphens) of the account to share with.
5. Choose **Save changes**.



## 10. Attach & Mount EBS Volume to EC2 Instance

Follow the steps given below carefully for the setup.

**Step 1:** Head over to EC2 → Volumes and create a new volume of your preferred size and type.

Volume Type: General Purpose SSD (gp2) ⓘ ← Select the required Volume Type

Size (GiB): 100 (Min: 1 GiB, Max: 16384 GiB) ⓘ

IOPS: 300 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS) ⓘ

Throughput (MB/s): Not applicable ⓘ

Availability Zone\*: us-west-2a ⓘ

Snapshot ID: Select a snapshot ↺ ⓘ

Encryption: ☐ Encrypt this volume

Key	Value
App	Jenkins
Name	jenkins-date

Add Tag 48 remaining (Up to 50 tags maximum)

Cancel Create Volume

**Note:** Make sure the EBS volume and the instance are in the same zone.

**Step 2:** Select the created volume, right-click and select the “attach volume” option.

**Step 3:** Select the ec2 instance from the instance text box as shown below.

**Attach Volume**

Volume ⓘ vol-3113afe8 in ap-northeast-2a

Instance ⓘ i-5f2b41f8 in ap-northeast-2a

Device ⓘ /dev/sdf  
Linux Devices: /dev/sdf through /dev/sdp

Note: Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name through /dev/sdp.

**Step 4:** Now, login to your ec2 instance and list the available disks using the following command.

```
lsblk
```

The above command will list the disk you attached to your instance.

**Step 5:** Check if the volume has any data using the following command.

```
sudo file -s /dev/xvdf
```

If the above command output shows “/dev/xvdf: data“, it means your volume is empty.

**Step 6:** Format the volume to the ext4 filesystem using the following command.

```
sudo mkfs -t ext4 /dev/xvdf
```

Alternatively, you can also use the xfs format. You have to use either ext4 or xfs.

```
sudo mkfs -t xfs /dev/xvdf
```

**Step 7:** Create a directory of your choice to mount our new ext4 volume. I am using the name “newvolume“. You can name it something meaningful to you.

```
sudo mkdir /newvolume
```

**Step 8:** Mount the volume to “newvolume” directory using the following command.

```
sudo mount /dev/xvdf /newvolume/
```

**Step 9:** cd into newvolume directory and check the disk space to validate the volume mount.

```
cd /newvolume
```

```
df -h .
```

The above command should show the free space in the newvolume directory.

To unmount the volume, use the unmount command as shown below..

```
umount /dev/xvdf
```