**Three tier architecture using docker:**

Open the ec2 and make ssh:

Now

step1:

Download the docker on linux:

sudo yum install docker -y

sudo service docker start

sudo systemctl enable docker.service

step2:

create two networks for containers which acts as public and private subnet:

docker network create public-net

docker network create private-net

docker network ls

(its shows you created networks)

Step3:

Creating the docker container for nginx

Run following command:

docker run -d -p 80:80 --name nginx-container --network public-net nginx

docker ps

(also check it by using public ip of your server that nginx container is running [sg port 80 allow])

Step4:

Creating container php-fpm for second container

We have a customized php-fpm container for that lets pull it and create it with public and private network.

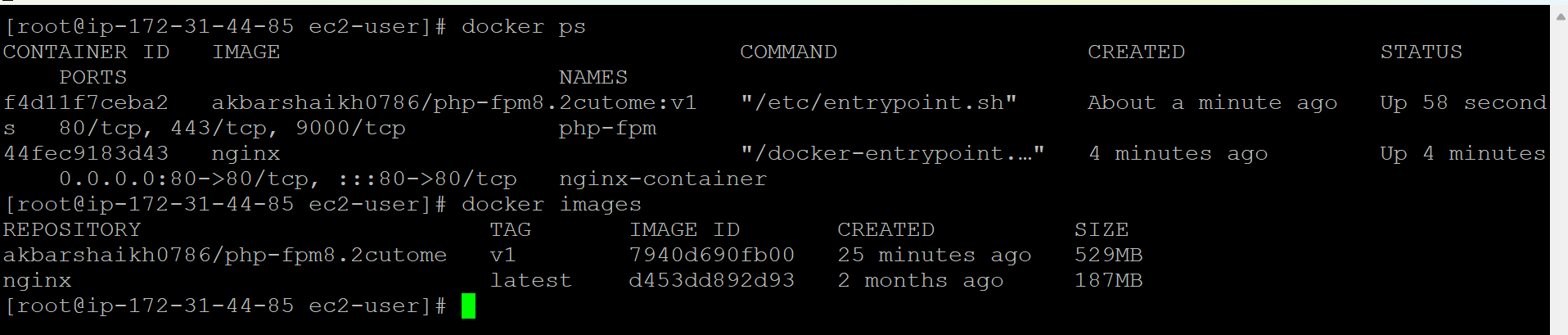
Run following command:

docker run -d --name php-fpm --network public-net akbarshaikh0786/php-fpm8.2cutome:v1

then list the images and container

docker ps

docker images ls



Then move to further step:

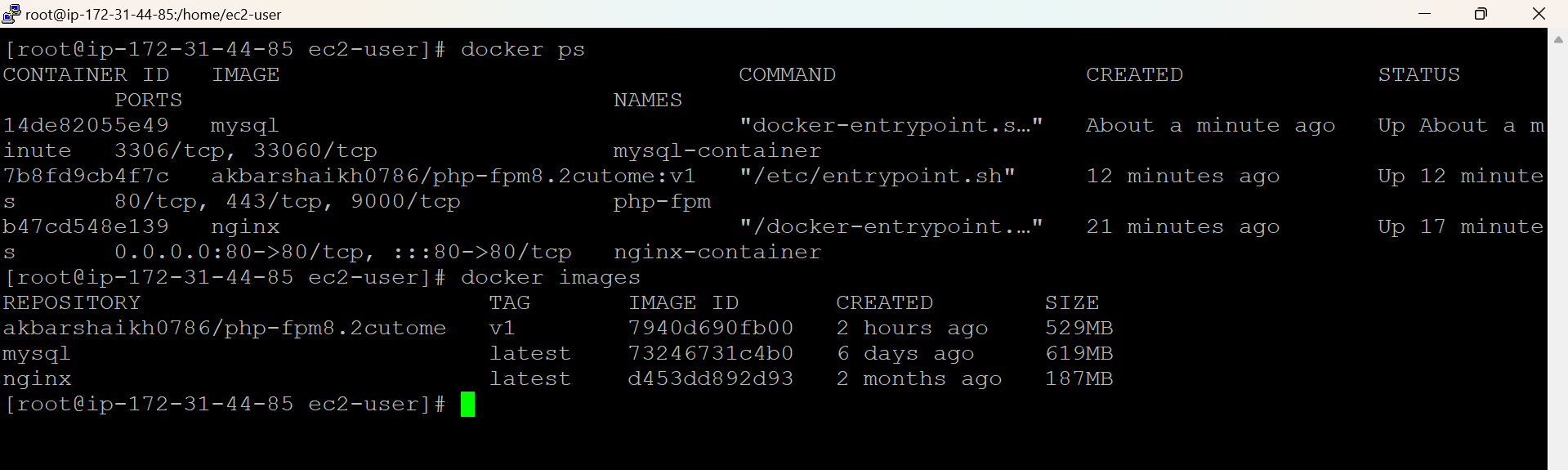
Step5:

Creating container for mysql which is in private network used as main db for our architecture.

Note for creating db we use enviromrnt variables root user pass and user as 3tier

Run the following command:

docker run -d --name mysql-container --network private-net -e MYSQL\_ROOT\_PASSWORD=Pass@123 -e MYSQL\_DATABASE=3tier mysql



Now add php-fpm in private network:

Run this command:

docker network connect private-net php-fpm

docker inspect php-fpm(172.19.0.3, 172.18.0.3)

docker inspect nginx-container(172.18.0.2)

now for next step we need to check out the ip of container in php-fpm and nginx so we can make proxy-pass of nginx to php-fpm.

Step6:

Change the configuration file for nginx container

For entering into container

Run this command:

docker exec -it nginx-container /bin/bash

------

apt-get update

apt-get install nano -y

after that change conf.

cd /etc/nginx/conf.d

ls

nano default.conf

copy this file :

server {

listen 80;

listen [::]:80;

server\_name localhost;

#access\_log /var/log/nginx/host.access.log main;

location / {

proxy\_pass <http://172.18.0.3>;

# add there public ip of php-fpm

# root /usr/share/nginx/html;

# index index.html index.htm;

}

#error\_page 404 /404.html;

# redirect server error pages to the static page /50x.html

#

# error\_page 500 502 503 504 /50x.html;

# location = /50x.html {

# root /usr/share/nginx/html;

#}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80

#

#location ~ \.php$ {

# proxy\_pass http://127.0.0.1;

#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000

#

#location ~ \.php$ {

# root html;

# fastcgi\_pass 127.0.0.1:9000;

# fastcgi\_index index.php;

# fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

# include fastcgi\_params;

#}

# deny access to .htaccess files, if Apache's document root

# concurs with nginx's one

#

#location ~ /\.ht {

# deny all;

#}

}

After that hit command for apply the conf changes:

service nginx reload

now the setup is ready so in databse lets add the fields according to our simple web view:

docker ps

docker exec -it mysql-container /bin/bash

now after that run following commands:

mysql -u root -p

enter password as(Pass@123)

show databases;

use 3tier;

now crete the table :

run this command:

create table user (id int auto\_increment not null primary key, name varchar(200), urls3 varchar(200));

desc user;

exit;

exit

step7:

lets create one s3 bucket to store the images uploaded by user.

Step8:

For uploading we need to install the docker composer which make connaection between the mysql and php so

Do follow

docker exec -it php-fpm /bin/bash

bydeault we enter as /var/www/mysite

cd ..

ls

cd html

rm index.html

nano index.html

copy this code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>3-Tier Architecture</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: white;

margin: 0;

padding: 0;

display: flex;

align-items: center;

justify-content: center;

height: 100vh;

}

.container {

max-width: 600px;

padding: 20px;

box-sizing: border-box;

background-color: grey;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1 {

text-align: center;

color: black;

}

label {

display: block;

margin: 10px 0 5px;

color: black;

}

input[type="text"],

input[type="email"],

textarea,

select {

width: 100%;

padding: 8px;

border: 1px solid #ccc;

box-sizing: border-box;

margin-bottom: 10px;

border-radius: 4px;

}

input[type="file"] {

margin-top: 5px;

}

input[type="submit"] {

width: 100%;

padding: 10px;

background-color: black;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

}

input[type="submit"]:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<div class="container">

<h1>Three Tier Architecture Using Docker And S3</h1>

<form action="upload.php" method="post" enctype="multipart/form-data">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required>

<label for="anyfile">Select image to upload:</label>

<input type="file" name="anyfile" id="anyfile">

<input type="submit" value="Upload Image" name="submit">

</form>

</div>

</body>

</html>

==================

nano upload.php

<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;

// Instantiate an Amazon S3 client.

$s3Client = new S3Client([

'version' => 'latest',

'region' => 'ap-south-1', //Add your bucket region here

'credentials' => [

'key' => 'AKIAR7MIJYJCCHJKKPXO', //Add your access key here

'secret' => 'dFBNdJhfH1QKq8xDldJRK5KRAUHhiykAovU7+oRG' //Add your secret key here

]

]);

// Check if the form was submitted

if($\_SERVER["REQUEST\_METHOD"] == "POST"){

// Check if file was uploaded without errors

if(isset($\_FILES["anyfile"]) && $\_FILES["anyfile"]["error"] == 0){

$allowed = array("jpg" => "image/jpg", "jpeg" => "image/jpeg", "gif" => "image/gif", "png" => "image/png");

$filename = $\_FILES["anyfile"]["name"];

$filetype = $\_FILES["anyfile"]["type"];

$filesize = $\_FILES["anyfile"]["size"];

// Validate file extension

$ext = pathinfo($filename, PATHINFO\_EXTENSION);

if(!array\_key\_exists($ext, $allowed)) die("Error: Please select a valid file format.");

// Validate file size - 10MB maximum

$maxsize = 10 \* 1024 \* 1024;

if($filesize > $maxsize) die("Error: File size is larger than the allowed limit.");

// Validate type of the file

if(in\_array($filetype, $allowed)){

// Check whether file exists before uploading it

if(file\_exists("uploads/" . $filename)){

echo $filename . " is already exists.";

} else{

if(move\_uploaded\_file($\_FILES["anyfile"]["tmp\_name"], "uploads/" . $filename)){

$bucket = '3tier-architecture'; //Add your bucket name here

$file\_Path = \_\_DIR\_\_ . '/uploads/'. $filename; // create uploads folder in html path

$key = basename($file\_Path);

try {

$result = $s3Client->putObject([

'Bucket' => $bucket,

'Key' => $key,

'Body' => fopen($file\_Path, 'r'),

'ACL' => 'public-read', // make file 'public'

]);

echo " Image uploaded successfully. Image path is: ". $result->get('ObjectURL');

} catch (Aws\S3\Exception\S3Exception $e) {

echo " There was an error uploading the file.\n";

echo $e->getMessage();

}

echo " Your file was uploaded successfully.";

}else{

echo "File is not uploaded";

}

}

} else{

echo " Error: There was a problem uploading your file. Please try again.";

}

} else{

echo " Error: " . $\_FILES["anyfile"]["error"];

}

}

$name=$\_POST["name"];

$s3url=$result->get("ObjectURL");

$servername = "mysql-container"; // Enter your database endpoint here

$username = "root"; // Enter your database username here

$password = "Pass@123"; // Enter your database password here

$dbname = "3tier"; // Enter your database name here

// Create connection

$conn = mysqli\_connect($servername, $username, $password, $dbname);

// Check connection

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

$sql = "INSERT INTO user(name,urls3) VALUES('$name','$s3url')";

if (mysqli\_query($conn, $sql)) {

echo " New record created successfully";

} else {

echo "Error: " . $sql . "<br>" . mysqli\_error($conn);

}

mysqli\_close($conn);

?>

Now in html directory create uploads folder

mkdir uploads

chmod 777 uploads

curl -sS https://getcomposer.org/installer | php

mv composer.phar /usr/local/bin/composer

ln -s /usr/local/bin/composer /usr/bin/composer

composer require aws/aws-sdk-php