Alibaba Cloud: Hands-On-Exercise

Application Deployment

Deploy web application sample to Alibaba Cloud. The minimum deployment should involve using of one ECS and one RDS instance.

Application Deployment

To deploy web application on Alibaba cloud (i.e., should involves using of one ECS and one RDS). I preferred to write a code in PHP language and built the CRUD operation where front-end s/w tools are installed on ECS and backend database deployed on RDS.

Following are the referral useful links to deploy the web application and to test it further as a pre-requisite:

On ECS over the CentOS image – I decided to installed following software and tools:

- Free Alibaba cloud account - https://www.alibabacloud.com/campaign/free-trial
- Apache: http://httpd.apache.org/
- PHP: http://www.php.net/
- MySQL: http://www.mysql.com/
- CentOS: http://www.centos.org/
- phpMyAdmin: http://www.phpmyadmin.net/
- Sample Web-application (**CRUD** based): https://github.com/RANBIJAY/CRUD-Operation-Alibaba-ECS-RDS

Suggestion: For best practice, it's always recommended to execute and run the sample application at your own environment first (i.e., local machine), before deploying it into the target cloud services.

Before starting of our hands-on exercise, let's do first benchmark set-up for both ECS and RDS services. During set-up process, users have a choice to opt either 'subscription' or 'pay-as-go' model depending on their needs.

I purchased super-high benchmark configuration for deployment of the sample web application and doing further R&D exercise in future assignment. The details are available below for reference:

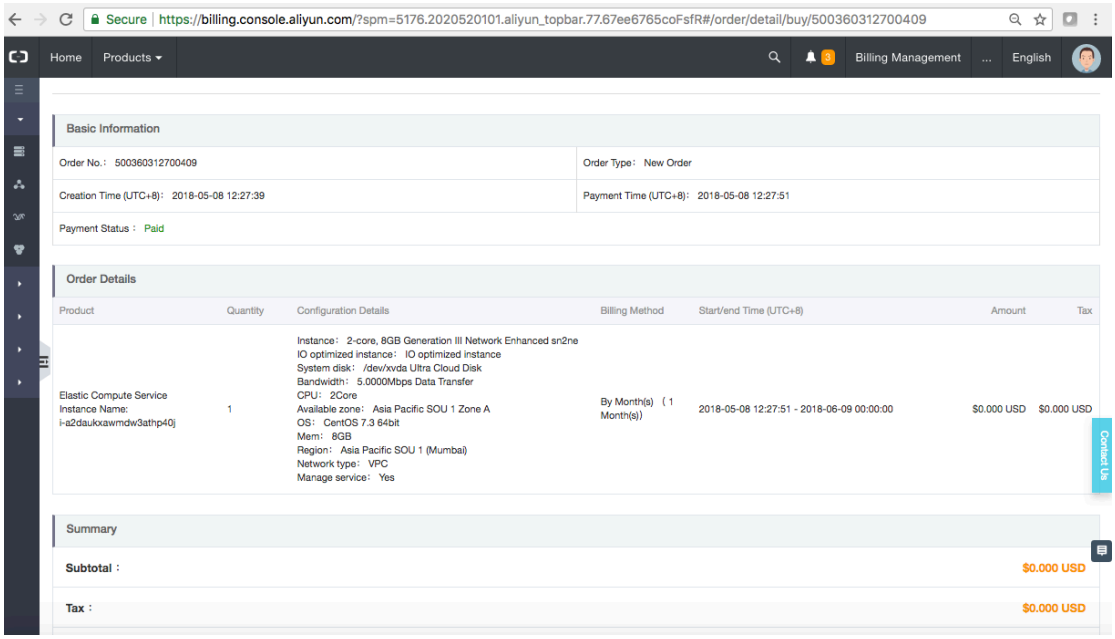
1. Benchmarking of ECS

For ECS instance, from the offered list of System, Marketplace and Custom Image, I preferred to go with CentOS system image for hands-on, as I am more comfortable to Linux based environment rather than Windows. Also, I have Mac machine for doing my personnel R&D or innovation for my personnel interest.

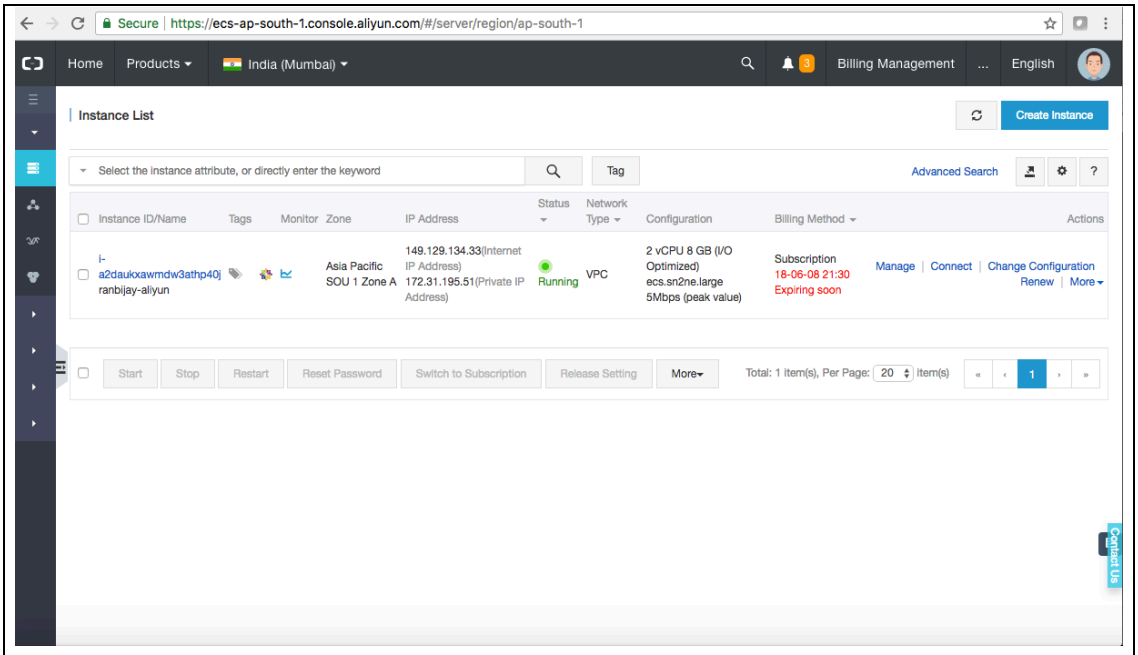
Product
Elastic Compute Service
Instance Name:
i-a2daukxawmdw3athp40j

Configuration Details

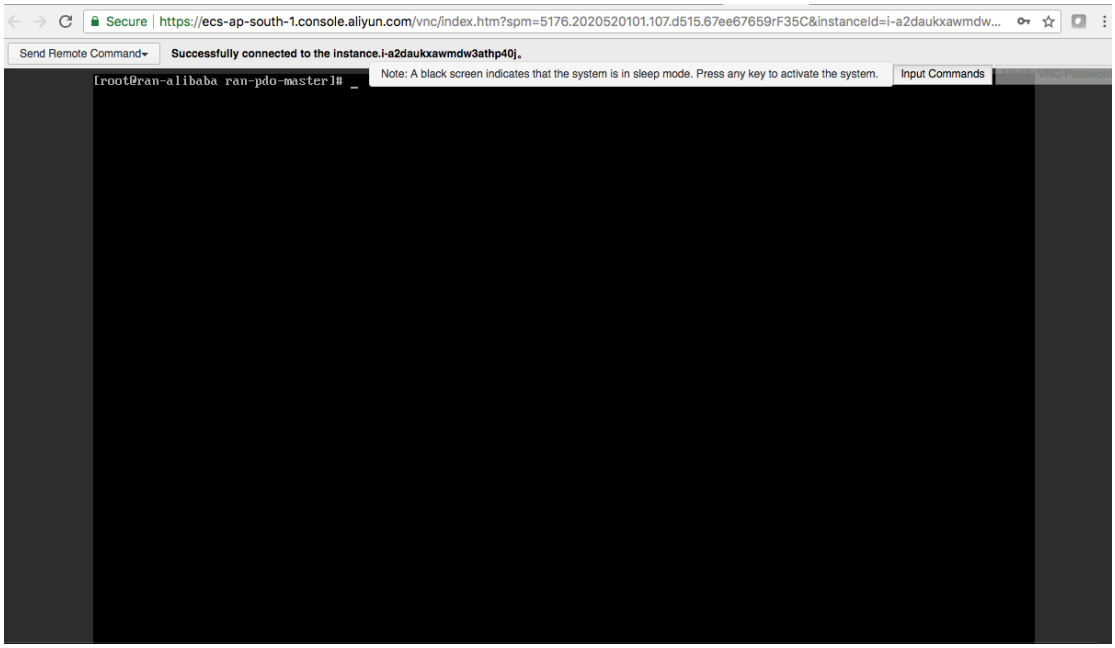
Instance： 2-core, 8GB Generation III Network Enhanced sn2ne
IO optimized instance： IO optimized instance
System disk： /dev/xvda Ultra Cloud Disk
Bandwidth： 5.0000Mbps Data Transfer
CPU： 2Core
Available zone： Asia Pacific SOU 1 Zone A
OS： CentOS 7.3 64bit
Mem： 8GB
Region： Asia Pacific SOU 1 (Mumbai)
Network type： VPC
Manage service： Yes



After ECS instance get activated, the status of instance been turned "Running" in status.



To test it further, how it works? Just clicked on "Connect" option at the right side of the Instance list portal, it opens the "Management console" for doing further exercise. Before getting the screen you need to provide **VNC** connection password. (Alert: The VNC connection password is viewable only once. Ensure you record this password and keep it in a secure location for subsequent login attempts.)



2. Benchmarking of RDS

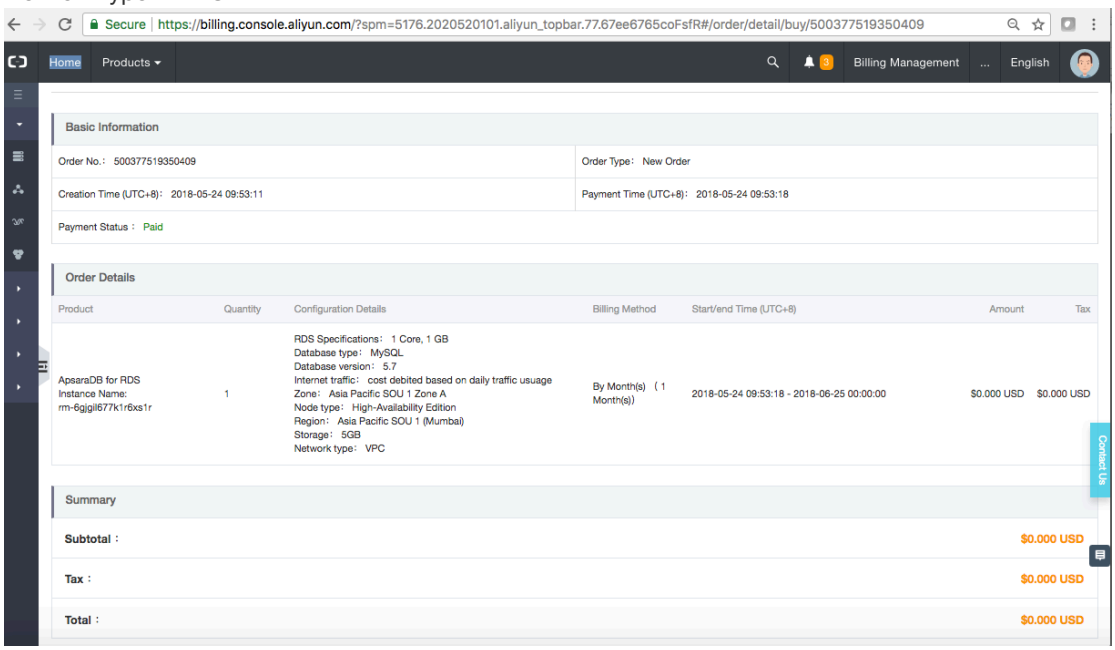
Product

ApsaraDB for RDS
Instance Name:
rm-6gjgil677k1r6xs1r

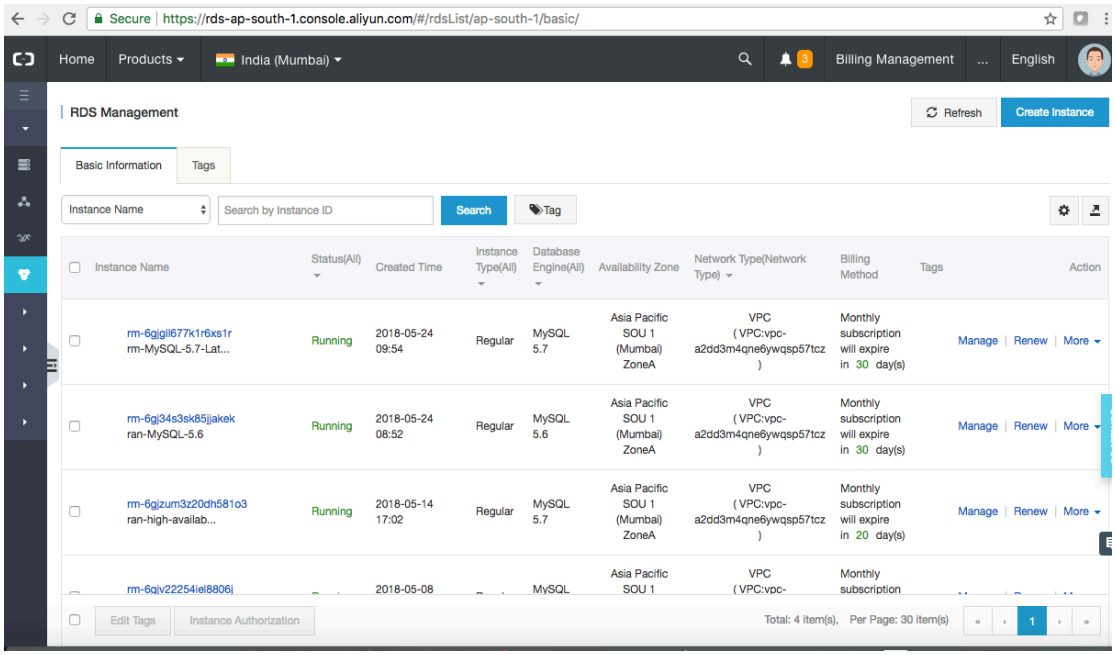
Configuration Details

RDS Specifications: 1 Core, 1 GB
Database type: MySQL
Database version: 5.7

Internet traffic： cost debited based on daily traffic usuage
Zone： Asia Pacific SOU 1 Zone A
Node type： High-Availability Edition
Region： Asia Pacific SOU 1 (Mumbai)
Storage： 5GB
Network type： VPC

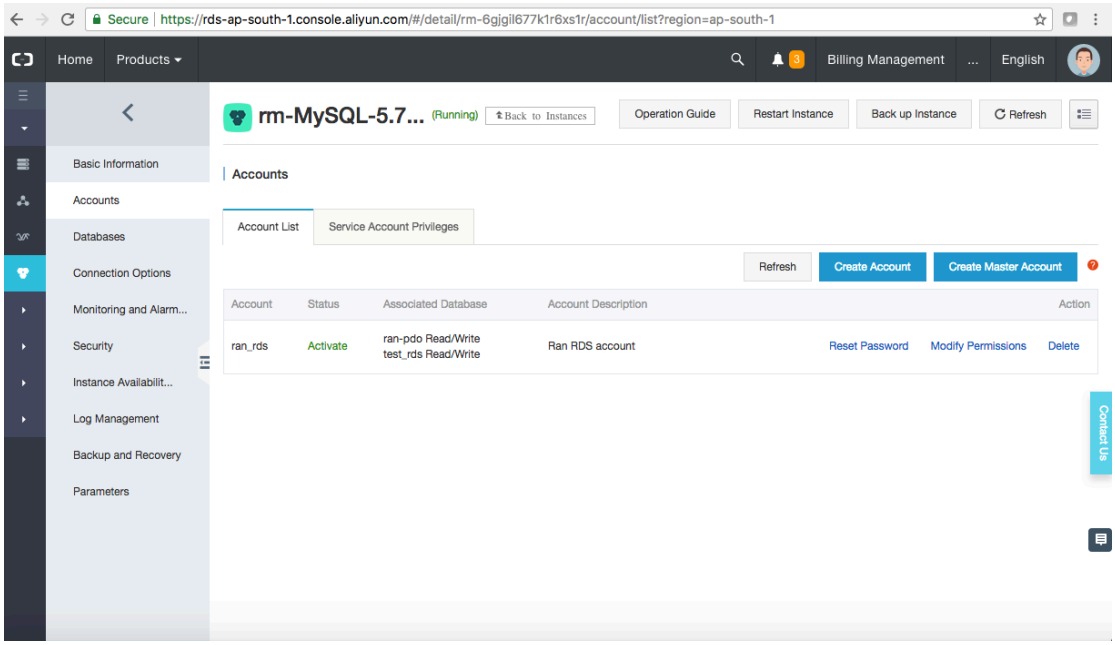


My current ApsaraDB for RDS Management portal looks like below image. Before doing any exercise to make sure the status of instance should be "Running".

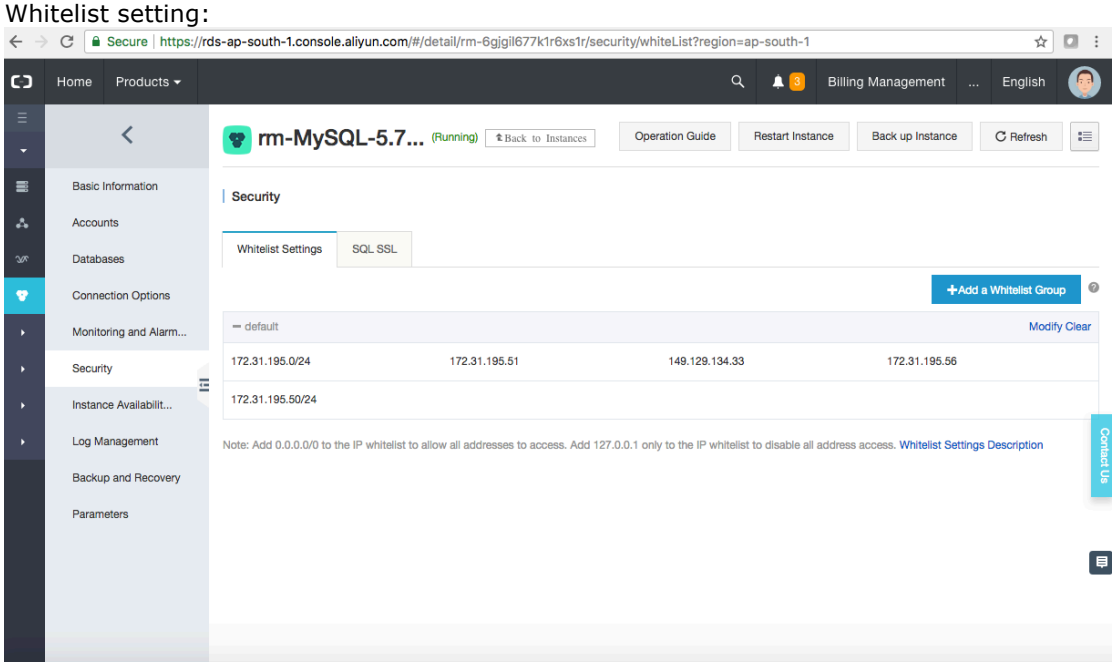


Since, I opted MySQL DB to use for my hands-on exercise. Related to MySQL, quick start, set-up and configuration details are well documented and available in Alibaba Cloud Document center for reference, helped me to do my hands-on in an easy manner.

Set-up MySQL root database user account as below [**Alert**: Creating master account activation will disappear account/database management from RDS instance portal, once the master account gets activated. Best practice here to create first database and account (ordinary one) and give privilege Read/Write to access them]



Whitelist setting:



Other product also explored and used for hands-on purpose, such as:

- Object Storage Service (to upload/download the sample web app)
- Cloud-Monitor Service

Steps of Hands-on Exercise:

Step1: Installing Apache, PHP 7.2 and MySQL on CentOS 7.4 System ECS Instance

1. Note

In this exercise, I use the hostname/Public_IP address with the IP 149.129.134.33.

I will add the EPEL repo here to install latest phpMyAdmin as follows:

[root@ran-alibaba /]# rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY*

[root@ran-alibaba /]# yum -y install epel-release

To edit files on the shell, I'll install the nano editor. If you prefer vi/vim for file editing, then skip this step.

[root@ran-alibaba /]# yum -y install nano

2. Installing MySQL / MariaDB

MariaDB is a MySQL fork of the original MySQL developer Monty Widenius. MariaDB is compatible with MySQL and I've chosen to use MariaDB here instead of MySQL. Run this command to install MariaDB with yum:

[root@ran-alibaba /]# yum -y install mariadb-server mariadb

Then we create the system startup links for MySQL (so that MySQL starts automatically whenever the system boots) and start the MySQL server:

[root@ran-alibaba /]# systemctl start mariadb.service

[root@ran-alibaba /]# systemctl enable mariadb.service

Set passwords for the MySQL root account:

[root@ran-alibaba /]# mysql_secure_installation

[root@ran-alibaba /]# mysql_secure_installation

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): <--ENTER

Ok, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB root user without the proper authorization.

[root@ran-alibaba /]# Set root password? [Y/n]

New password: <--yourmariadbpassword

Re-enter new password: <--yourmariadbpassword

Password updated successfully!

Reloading privilege tables.

... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB without having to have a user account created for him or her. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

Remove anonymous users? [Y/n] <--ENTER

... 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] <--ENTER

... 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] <--ENTER

- 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] <--ENTER

... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

[root@ran-alibaba /]#

3. Installing Apache

CentOS 7 ships with Apache 2.4. Apache is directly available as a CentOS 7 package, therefore we can install it like this:

[root@ran-alibaba /]# yum -y install httpd

Now configure your system to start Apache at boot time...

[root@ran-alibaba /]# systemctl start httpd.service

[root@ran-alibaba /]# systemctl enable httpd.service

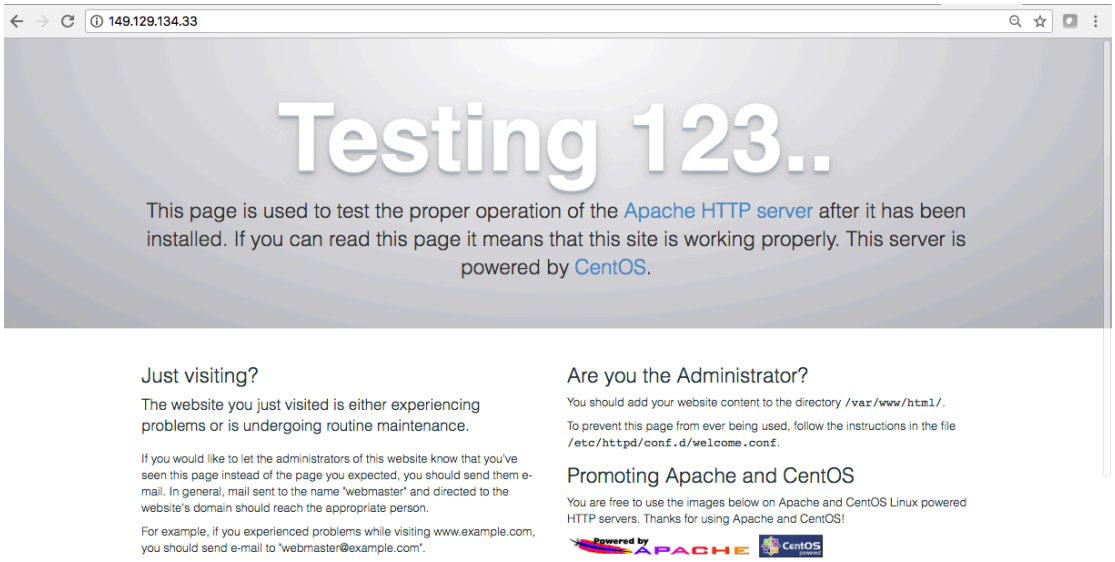
To be able to access the web server from outside, we have to open the HTTP (80) and HTTPS (443) ports in the firewall. The default firewall on CentOS is firewalld, which can be configured with the firewalld-cmd command.

[root@ran-alibaba /]# firewall-cmd --permanent --zone=public --add-service=http

[root@ran-alibaba /]# firewall-cmd --permanent --zone=public --add-service=https

[root@ran-alibaba /]# firewall-cmd --reload

Now direct your browser to the IP address of your server, in my case http://149.129.134.33/, and you should see the Apache placeholder page:



4. Installing PHP

The PHP version that ships with CentOS as default is quite old (PHP 5.4). Therefore I will show you in this exercise some options to install newer PHP versions like PHP 7.0 or 7.1 from Remi repository.

Add the Remi CentOS repository.

[root@ran-alibaba /]# rpm -Uvh http://rpms.remirepo.net/enterprise/remi-release-7.rpm

[root@ran-alibaba /]# Install yum-utils as we need the yum-config-manager utility.

[root@ran-alibaba /]# yum -y install yum-utils

and run yum update

[root@ran-alibaba /]# yum update

Now you have an option to choose which PHP version you want to use on the server. PHP version at a time with Apache mod_php. To install PHP, run this command:

```
[root@ran-alibaba /]# yum -y install php
```

To Install PHP 7.2, use:

```
[root@ran-alibaba /]# yum-config-manager --enable remi-php72
```

```
[root@ran-alibaba /]# yum -y install php php-opcache
```

We must restart Apache to apply the changes:

```
[root@ran-alibaba /]# systemctl restart httpd.service
```

5. Testing PHP / Getting Details About Your PHP Installation

The document root of the default website is /var/www/html. We will create a small PHP file (info.php) in that directory and call it in a browser to test the PHP installation. The file will display lots of useful details about our PHP installation, such as the installed PHP version.

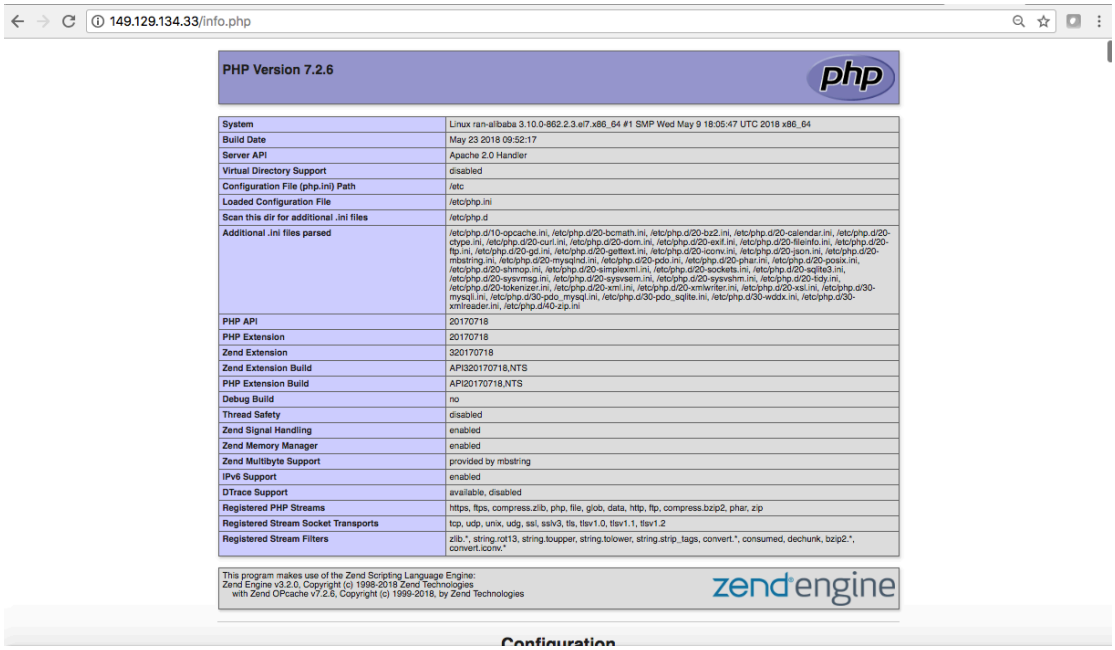
```
[root@ran-alibaba /]# nano /var/www/html/info.php  [install nano editor on CentOS: yum -y install nano]
```

In nano editor, type following code and exit.

```
<?php

phpinfo();
```

To switch from nano editor to root-command, in Mac OSX: press **fn+F2.** Now we call that file in a browser (e.g. http://149.129.134.33/info.php):



As you see, PHP 7.2 is working, and it's working through the Apache 2.0 Handler, as shown in the Server API line. If you scroll further down, you will see all modules that are

already enabled in PHP. MySQL is not listed there which means we don't have MySQL support in PHP yet.

6. Getting MySQL Support In PHP

To get MySQL support in PHP, we can install the php-mysqlnd package. It's a good idea to install some other PHP modules as well as you might need them for your applications. You can search for available PHP5 modules like this:

```
[root@ran-alibaba /]# yum search php
```

Pick the ones you need and install them like this:

```
[root@ran-alibaba /]# yum -y install php-mysqlnd php-pdo
```

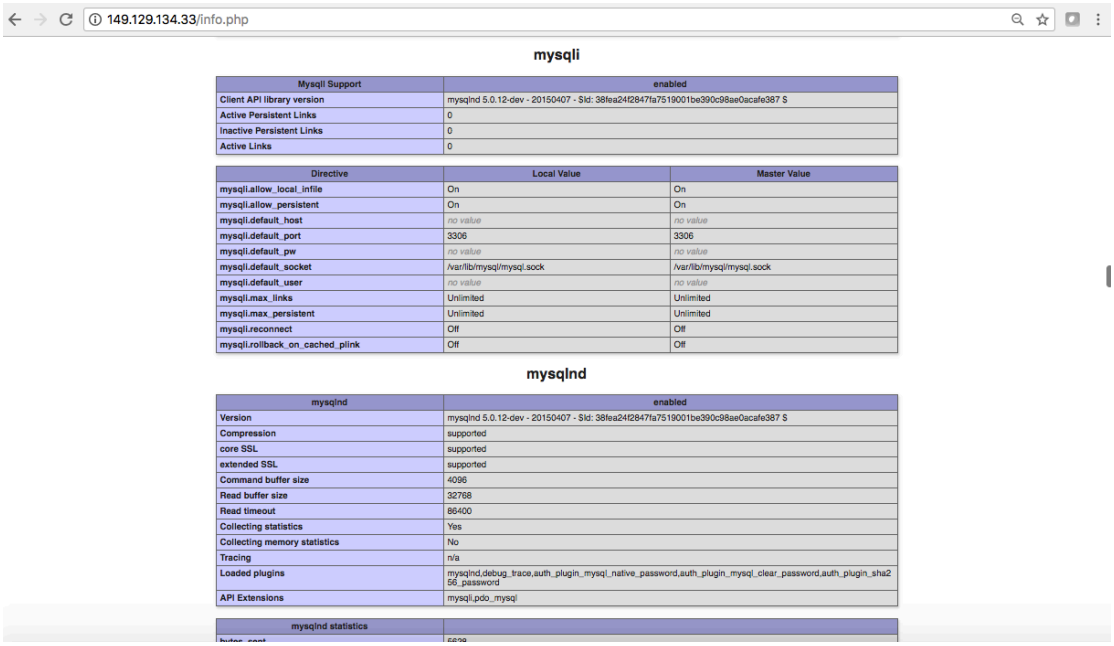
In the next step I have installed some common PHP modules that are required by CMS Systems like Wordpress, Joomla, and Drupal (for future exercise and learning):

```
[root@ran-alibaba /]# yum -y install php-gd php-ldap php-odbc php-pear php-xml php-xmlrpc php-mbstring php-soap curl curl-devel
```

Now restart Apache web server:

```
[root@ran-alibaba /]# systemctl restart httpd.service
```

Now reload http://149.129.134.33/info.php in your browser and scroll down to the modules section again. You should now find lots of new modules like curl etc. there:



If you don't need the php info output anymore, then delete that file for security reasons.

```
[root@ran-alibaba /]# rm /var/www/html/info.php
```

7. phpMyAdmin installation

phpMyAdmin is a web interface through which you can manage your MySQL databases.

phpMyAdmin can now be installed as follows:

```
[root@ran-alibaba /]# yum -y install phpMyAdmin
```

Now we configure phpMyAdmin. We change the Apache configuration so that phpMyAdmin allows connections not just from localhost (by commenting out the <RequireAny> stanza and adding the **'Require all granted'** line):

```
[root@ran-alibaba /]# nano /etc/httpd/conf.d/phpMyAdmin.conf
```

Inside the nano editor, the screens should be

```
[...]

Alias /phpMyAdmin /usr/share/phpMyAdmin

Alias /phpmyadmin /usr/share/phpMyAdmin

<Directory /usr/share/phpMyAdmin/>

 AddDefaultCharset UTF-8

 <IfModule mod_authz_core.c>

 # Apache 2.4

# <RequireAny>

# Require ip 127.0.0.1

# Require ip ::1

# </RequireAny>

 Require all granted

 </IfModule>

 <IfModule !mod_authz_core.c>

 # Apache 2.2

 Order Deny,Allow

 Deny from All

 Allow from 127.0.0.1

 Allow from ::1

 </IfModule>

</Directory>

<Directory /usr/share/phpMyAdmin/>
```

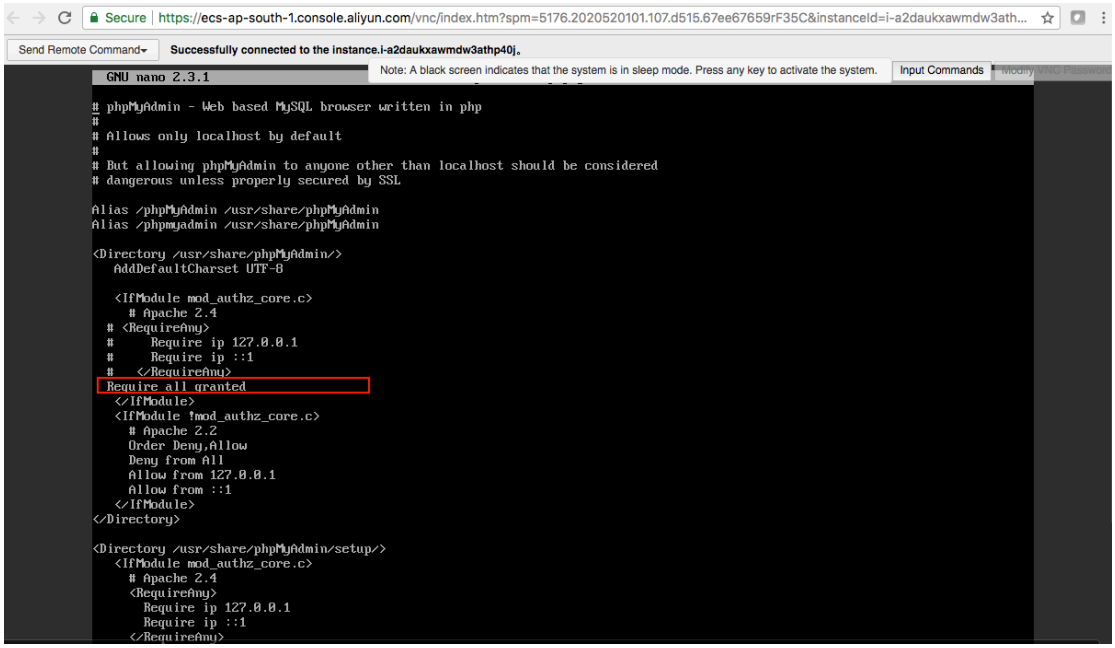
Options none

AllowOverride Limit

Require all granted

</Directory>

[...]



Next, we change the authentication in phpMyAdmin from cookie to http:

[root@ran-alibaba /]# nano /etc/phpMyAdmin/config.inc.php

[...]

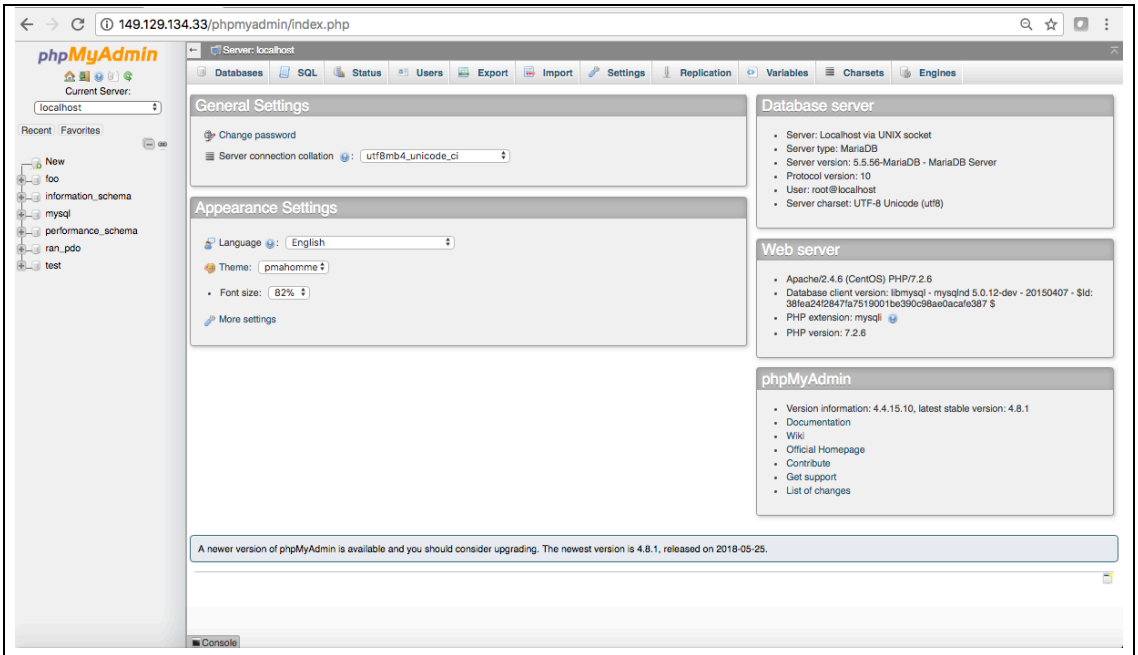
$cfg['Servers'][$i]['auth_type'] = 'http'; // Authentication method (config, http or cookie based)?

[...]

Restart Apache:

[root@ran-alibaba /]# systemctl restart httpd.service

Afterwards, you can access phpMyAdmin under **http://149.129.134.33/phpmyadmin/**:



Step2: phpmyadmin_connect_to_rds [Tutorial on connecting your phpmyadmin to an external Cloud RDS instance]

[root@ran-alibaba /]# nano /etc/phpMyAdmin/config.inc.php

--ADD LINES BELOW THE PMA CONFIG AREA AND FILL IN DETAILS--

$i++;

$cfg['Servers'][$i]['host'] = '<RDS-Endpoint>';

$cfg['Servers'][$i]['port'] = '3306';

$cfg['Servers'][$i]['socket'] = '';

$cfg['Servers'][$i]['connect_type'] = 'tcp';

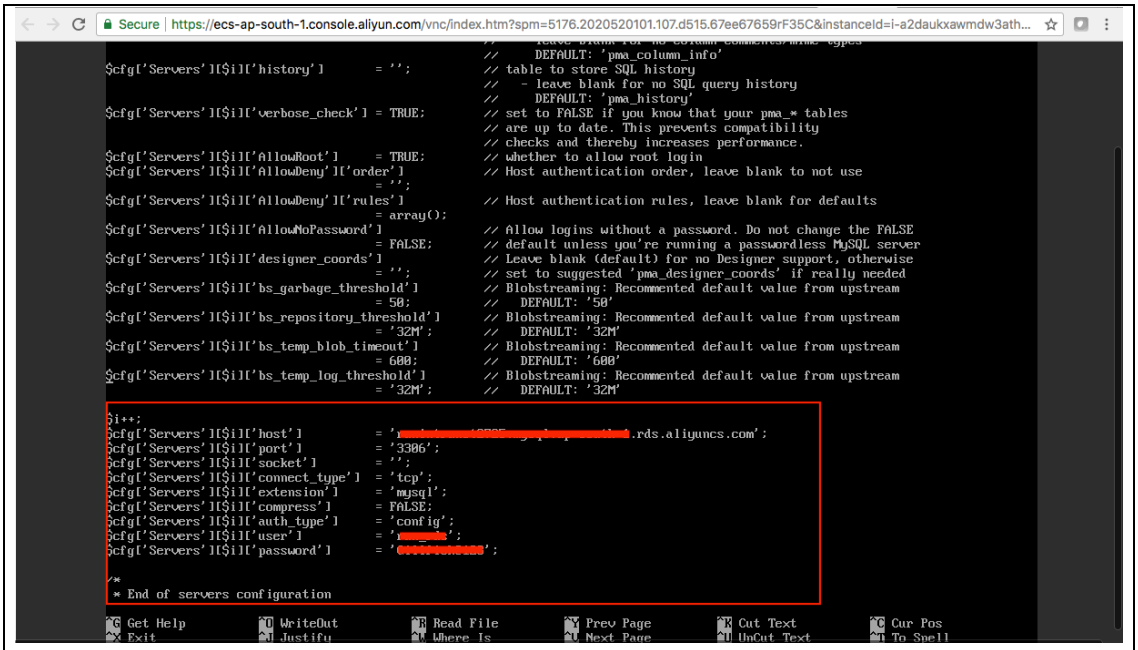
$cfg['Servers'][$i]['extension'] = 'mysql';

$cfg['Servers'][$i]['compress'] = FALSE;

$cfg['Servers'][$i]['auth_type'] = 'config';

$cfg['Servers'][$i]['user'] = '<Username>';

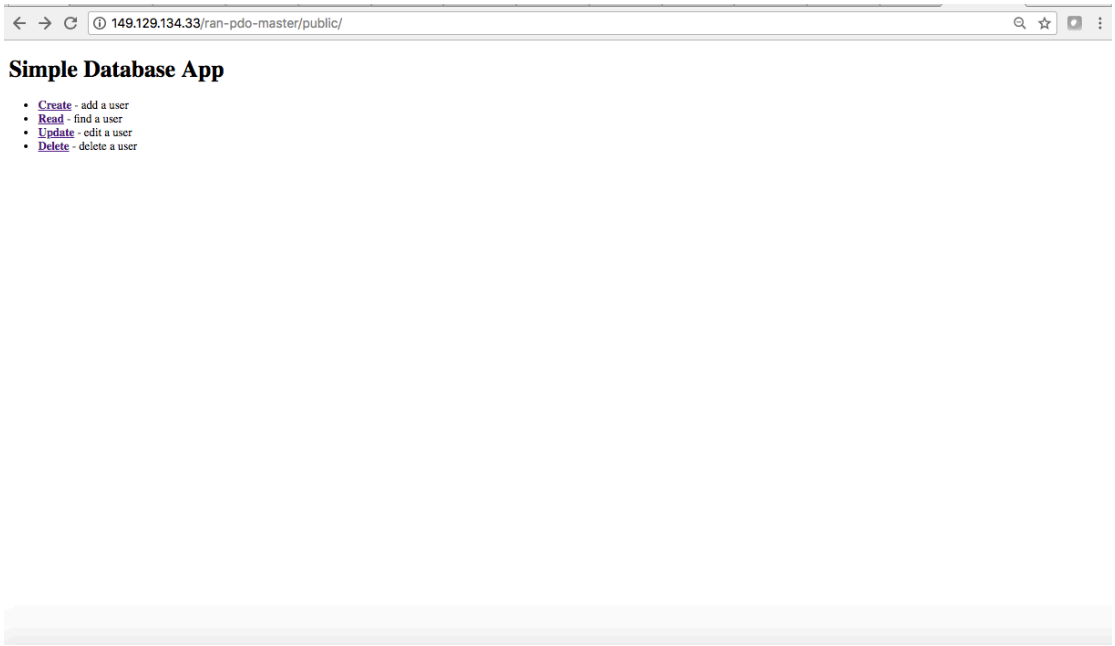
$cfg['Servers'][$i]['password'] = '<Password>';

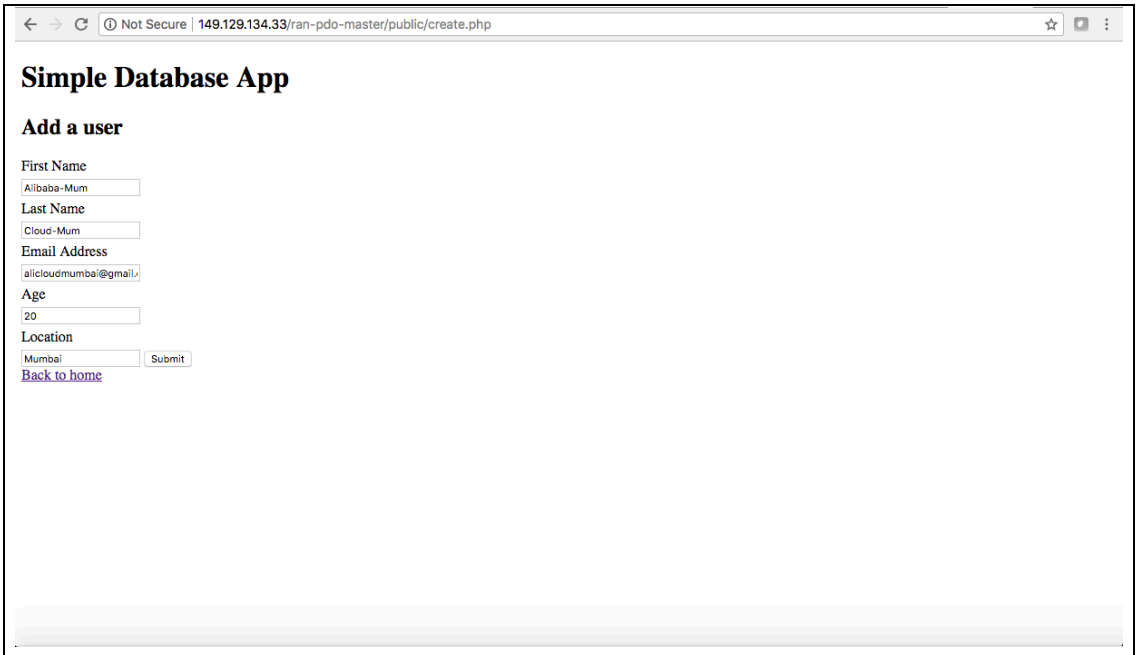


Step3: Testing Web-Application URL:

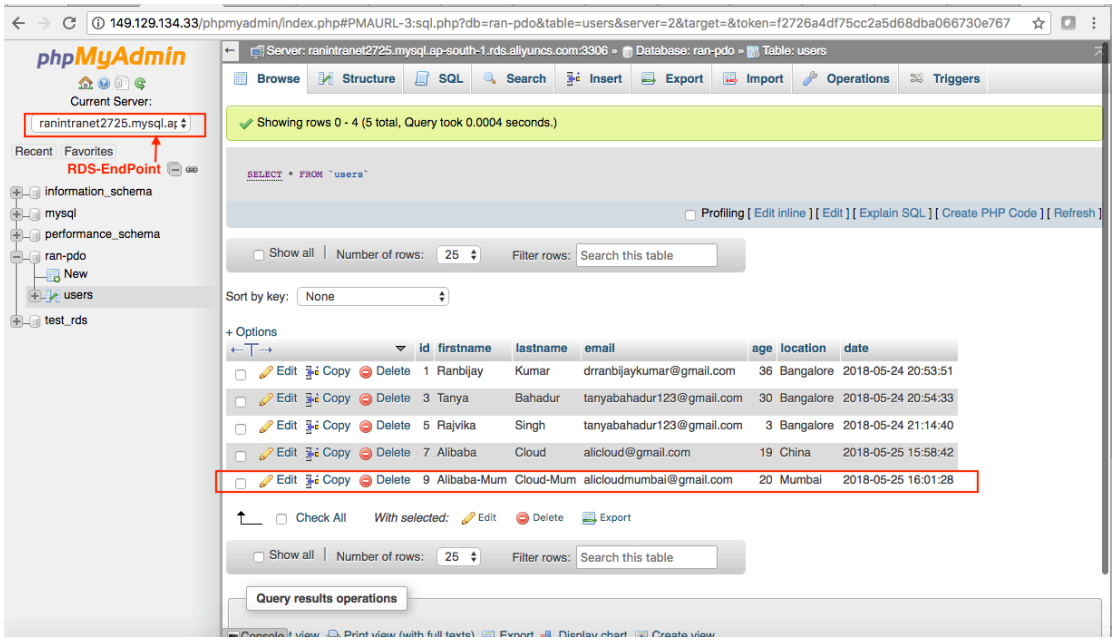
http://149.129.134.33/ran-pdo-master/public/

Screen-shot UI:




Not Secure | 149.129.134.33/ran-pdo-master/public/create.php
Simple Database App
Add a user
First Name
Alibaba-Mum
Last Name
Cloud-Mum
Email Address
alicloudmumbai@gmail.
Age
20
Location
Mumbai
Submit
Back to home

Backend:


149.129.134.33/phpmyadmin/index.php#PMAURL-3:sql.php?db=ran-pdo&table=users&server=2&target=&token=f2726a4df75cc2a5d68dba066730e767
phpMyAdmin
Current Server:
ranintranet2725.mysql.ap
Recent
Favorites
RDS-EndPoint
information_schema
mysql
performance_schema
ran-pdo
New
users
test_rds
Server: ranintranet2725.mysql.ap-south-1.rds.aliyuncs.com:3306 » Database: ran-pdo » Table: users
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Triggers
Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)
SELECT * FROM `users`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP Code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table
Sort by key: None
+ Options
id firstname lastname email age location date
Edit Copy Delete 1 Ranbijay Kumar drranbijaykumar@gmail.com 36 Bangalore 2018-05-24 20:53:51
Edit Copy Delete 3 Tanya Bahadur tanyabahadur123@gmail.com 30 Bangalore 2018-05-24 20:54:33
Edit Copy Delete 5 Rajvika Singh tanyabahadur123@gmail.com 3 Bangalore 2018-05-24 21:14:40
Edit Copy Delete 7 Alibaba Cloud alicloud@gmail.com 19 China 2018-05-25 15:58:42
Edit Copy Delete 9 Alibaba-Mum Cloud-Mum alicloudmumbai@gmail.com 20 Mumbai 2018-05-25 16:01:28
Check All With selected: Edit Delete Export
Show all | Number of rows: 25 Filter rows: Search this table
Query results operations