

Below are the steps to install Orangescrum Self Hosted version of Cloud Edition in CentOS Server

Last Updated: *May 26, 2017*

STEP 1

Download the Orangescrum Self Hosted version of Cloud Edition

Extract the archive file.

You will find the following list of files/folder

1. orangescrum - folder

You will find user license wise sql file(Ex. database.sql) inside orangescrum folder.

2. installationmanual.pdf

3. Perpetual license doc

Note: Strictly follow the below steps to install self hosted orangescrum.

STEP 2

Required packages to install: Php, MySQL, Apache

Requirement:

* Apache with `mod_rewrite`

* Enable curl in php.ini

* Change the 'post_max_size' and `upload_max_filesize` to 200Mb in php.ini

* PHP 5.5 or PHP 5.6

Required PHP Extensions: -

php5.6-gd

php5.6-curl

php5.6-common

php5.6-fpm

php5.6-cli

php5.6-imap
php5.6-intl
php5.6-ldap
php5.6-mysql
php5.6-snmp
php5.6-tidy
php5.6-mcrypt
php5.6-mbstring
php5.6-soap
php5.6-zip
php5.6-dba

* MySQL 4.1 or higher

* If STRICT mode is On, turn it Off.

wkhtmltopdf installation Process

```
sudo apt-get update  
sudo apt-get install wkhtmltopdf  
check version - wkhtmltopdf -V
```

Define the path for wkhtmltopdf in Constants.php
For Ex. ***define('PDF_LIB_PATH', '/usr/bin/wkhtmltopdf');***

STEP 3

- Extract the archive file.
- Upload folder (orangescrum) to your working directory (/var/www/html).
- Provide proper write permission to " app/tmp ", " app/webroot " and " app/Config " folders and their sub folders.

```
chmod -R 0777 app/Config
```

```
chmod -R 0777 app/tmp
```

```
chmod -R 0777 app/webroot
```

You can change the write permission of "app/Config" after installation procedure is completed.

STEP 4

Install MySQL: (Login as root user)

```
yum -y install mysql mysql-server
```

Then we will setup the system startup links for MySQL (MySQL starts automatically whenever the system boots):`chkconfig --levels 235 mysqld on`

Start the MySQL server:`/etc/init.d/mysqld start`

Setup the MySQL root password:`mysql_secure_installation`

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current password for the root user. If you've just installed MySQL, 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 MySQL root user without the proper authorisation.

Set root password? [Y/n] ← **ENTER**

New password: ← **yourrootsqlpassword**>

Re-enter new password: ← **yourrootsqlpassword**>

Password updated successfully!

Reloading privilege tables..

... Success!

By default, a MySQL installation has an anonymous user, allowing anyone to log into MySQL without having to have a user account created for them. 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, MySQL 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 MySQL installation should now be secure.

Thanks for using MySQL!

STEP 5

Login To MySQL & Create Database named "orangescrum" (you can rename to any name)

login to mysql:

```
[root@server ~]# mysql -u root -p
```

Enter password:

create the database:

```
mysql> create database orangescrum; (database name)
```

verify that it's there:

```
mysql> show databases;
```

create the user:

```
mysql> create user orangescrum; (database user name)
```

Grant all privileges while assigning the password:

```
mysql> grant all on orangescrum.* to 'orangescrum'@'localhost' identified by 'your_password';
```

Exit from the database:

```
mysql> exit
```

Import database sql file:

Navigate to /var/www/html/orangescrum directory by typing

cd /var/www/html/ orangescrum

```
[root@server ~]# mysql -u orangescrum -p orangescrum < database.sql
```

Enter password:

Login to the database and check whether your tables are created or not:

```
[root@server ~]# mysql -u orangescrum -p
```

Enter password:

```
mysql> show databases
```

if your database exist then trigger the below command:

```
mysql> use orangescrum;
```

```
mysql> show tables;
```

```
mysql> exit
```

```
mysql> exit
```

[Manage MySQL Databases \(Optional\)](#)

Install phpMyAdmin(To access database Graphically)

```
yum install epel-release
```

```
yum install phpMyAdmin
```

To access phpmyadmin in browser

Open the file

```
vim /etc/httpd/conf.d/phpMyAdmin.conf
```

And add

Require all granted

in the below patterns.

```
<IfModule mod_authz_core.c>
  # Apache 2.4
  <RequireAny>
    Require ip 127.0.0.1
    Require ip ::1
    Require all granted
  </RequireAny>
</IfModule>
<IfModule !mod_authz_core.c>
```

STEP 6

Install the Apache Server:

```
yum -y install httpd
```

Apache start in system boot time:

```
chkconfig --levels 235 httpd on
```

Start Apache Service:

```
/etc/init.d/httpd start Or service httpd start
```

STEP 7

Install the PHP Packages: (Recommended Version php 56)

Install Below Two Repository: -

```
wget http://dl.fedoraproject.org/pub/epel/7/x86\_64/e/epel-release-7-5.noarch.rpm
```

```
wget http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
```

```
rpm -Uvh remi-release-7*.rpm epel-release-7*.rpm
```

Enabling the Repository :

```
cd /etc/yum.repos.d
```

you should inside see a file called `remi.repo`.

Open the file in your favorite editor (Nano, Vi, vim etc), you'll see a number of sections. We need to make sure that the first section `[remi]` is enabled:

```
1 [remi]
2 name=Les RPM de remi pour Enterprise Linux 6 - $basearch
3 #baseurl=http://rpms.famillecollet.com/enterprise/6/remi/$basearch/
4 mirrorlist=http://rpms.famillecollet.com/enterprise/6/remi/mirror
5 enabled=1
6 gpgcheck=1
7 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi
```

Note the line `enabled=1` make sure this is set

If we want PHP 5.5 or PHP 5.6 we need to do a bit more work, further down in the `repo.repo` file you will see two additional sections `[remi-php55]` and `[remi-php56]`, decide which PHP version you want to install and then enable the correct. So for PHP 5.6 we would change to:

```
1 [remi-php56]
2 name=Les RPM de remi de PHP 5.6 pour Enterprise Linux 6 - $basearch
3 #baseurl=http://rpms.famillecollet.com/enterprise/6/php56/$basearch/
4 mirrorlist=http://rpms.famillecollet.com/enterprise/6/php56/mirror
5 # WARNING: If you enable this repository, you must also enable "remi"
6 enabled=1
7 gpgcheck=1
8 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi
```

`yum search php56`

```
yum install php56.x86_64 php56-php.x86_64 php56-php-cli.x86_64 php56-php-
common.x86_64 php56-php-dba.x86_64 php56-php-devel.x86_64 php56-php-
embedded.x86_64 php56-php-fpm.x86_64 php56-php-gd.x86_64 php56-php-imap.x86_64
php56-php-intl.x86_64 php56-php-ldap.x86_64 php56-php-mbstring.x86_64 php56-php-
mcrypt.x86_64 php56-php-pdo.x86_64 php56-php-mysql.x86_64 php56-php-pecl-
imagick.x86_64 php56-php-snmp.x86_64 php56-php-soap.x86_64 php56-php-tidy.x86_64
php56-php-xml.x86_64 php56-php-zip.x86_64 -y
```

Restart the Apache service:

```
/etc/init.d/httpd restart
```

Or

```
Service httpd restart
```

STEP 8

Setup the database information in `app/config/database.php`

Update the database connection details. (host, login, password and database name).

STEP 9

Virtual Host setup in Apache conf file.

```
Vi /etc/httpd/conf/httpd.conf
```

Add this below details to last line in `httpd.conf` file

```
<VirtualHost *:80>
ServerName localhost # eg: demo.orangescrum.com
DocumentRoot /var/www/html/orangescrum /
<Directory /var/www/html/orangescrum />
Options Indexes FollowSymLinks MultiViews
AllowOverride All
Order allow,deny
allow from all
</Directory>
</VirtualHost>
```

Save the file and Restart the Apache service

```
/etc/init.d/httpd restart / service httpd restart
```

STEP 10

General Configuration management:

MySQL:

- If STRICT mode is On, turn it Off.

Disable Strict mode on mysql for Centos/Fedora :-

```
vim /etc/my.conf
```

```
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
```

to

```
sql_mode=
```

```
sudo service mysql restart
```

PHP:

Enable curl in php.ini

Install the curl package to enable curl

```
sudo yum install php5-curl
```

And restart the apache server

```
sudo service httpd restart
```

Change the 'post_max_size' and 'upload_max_filesize' to 200Mb in php.ini

Open the `php.ini` file and change the `'post_max_size'` and `'upload_max_filesize'` to 200Mb

`Vim /etc/php.ini`

Make sure that, the `.htaccess` is working in your server.

STEP 11

Application Setup:

- Provide your valid Gmail ID and Password for SMTP email sending Or, you can use Sendgrid or Mandrill.
- Change the `'post_max_size'` and `'upload_max_filesize'` to 200Mb in `php.ini`
- **Open the `php.ini` file and change the `'post_max_size'` and `'upload_max_filesize'` to 200Mb**

`Vim /etc/php.ini`

- **Make sure that, the `.htaccess` is working in your server.**

SMTP (IMPORTANT)

Provide your valid Gmail ID and Password for SMTP email sending Or, you can use Sendgrid or Mandrill in `app/Config/Constants.php` file

FROM_EMAIL - this will be the default from email through out Orangescrum.

FROM_EMAIL_EC All the task created/updated notification email will be sent from this Email ID.

SUPPORT_EMAIL All Other Emails and support related Emails will be sent from this Email ID.

Please follow below steps to configure your SMTP:

Step 1:

Go to the path in your application: `app/Config/constants.php`

Step 2:

Set the following in Gmail SMTP:

```
define("SMTP_HOST", "ssl://smtp.gmail.com");
```

```
define("SMTP_PORT", "465");
```

```
define("SMTP_UNAME", "youremail@gmail.com");
```

```
define("SMTP_PASSWORD", "*****");
```

Please provide your smtp username and password to configure your SMTP.

If you are finding problem after setting the username and password, then please check the below:

Go to your php.ini file and enable the "php_openssl" in you PHP setting.

You need to enable extension=php_openssl.dll on php.ini file. If you are still facing the problem, then the port 465 is not available. Try with 587 port.

Otherwise, please contact with your server administrator.

Please run the following URL to check email is working or not:

http://YOURDOMAINNAME/cron/test_email/?to=emailId

Example:

http://localhost/orangescrum/cron/test_email/?to=emailId (if you are using localhost)

http://127.0.0.1/orangescrum/cron/test_email/?to=emailId (if you are using IP)

http://myprojects.orangescrum.com/cron/test_email/?to=emailId (if you are using any valid domain)

If the email is not working, then please follow the installation guide again.

Check Gmail security issues here. Please see below links, you have to change some configuration at email end.

<https://support.google.com/accounts/answer/6010255?hl=en>

<https://www.google.com/settings/security/lesssecureapps>

Also check app/tmp/logs/os-email.log file for any error.

Check SMTP Port:

If ports (465,587) are not enabled, then open it.

Still mail functionality is not working then try with changing the ports to (465,587, 25)

Add the hostname according to the mail server imap setting and Imap port. If ssl is enable then add no validate-cert parameter to the host name.

Then Add HOST Name, User Name & Password

To Use Own Email Server:

For own Email server: Add HOST_NAME, PORT, SMTP_UNAME, SMTP_PASSWORD in constant.php instead of gmailconfiguration.

Incorrect authentication data error:

Please double check the username and password.

If both are correct then check the server configuration. It's somehow blocking the server to connect to the mail server. We cannot do anything to fix this.

Also make sure you're not using 2 step verification and less secure app setting is ON.

FROM_EMAIL_EC All the task created/updated notification email will be sent from this Email ID.

SUPPORT_EMAIL All Other Emails and support related Emails will be sent from this Email ID.

STEP 12

Email Reply - Nohup Cron Job setup (Linux Server)

Make sure to do the following changes on the Email server connection details in the app/webroot/EmailReply.php file.

\$username- This will be the FROM_EMAIL_EC Email set on your app/Config/constants.php

All the task created/updated notification email will be sent from FROM_EMAIL_EC . When somebody will reply on that task created/updated notification email, the FROM_EMAIL_EC will get that Email in the inbox.

EmailReply.phppage is going to read the emails from FROM_EMAIL_EC and It will post to them as a reply to the respective tasks in Orangescrum.

\$password- Password of FROM_EMAIL_EC

client- Change it, if you are not using Gmail

After this setup, you can reply to a task created/updated notification email and that email reply will be posted to Orangescrum under that Task. This will help you to respond to a task while on the go from your Mobile.

(Assuming your Application is in "/var/www/html/orangescrum/ ")

Enable extension=php_imap in your php.ini file

Create a orangescrum.sh file in your server

```
vi orangescrum.sh (or, open that file to write the below code)#!/bin/bash
while(true)
```

```
do
cd /var/www/html/orangescrum/app/webroot
php q EmailReply.php 1>&2
sleep 1?
done
```

Give the execute permission for orangescrum.sh: `chmod +x orangescrum.sh`

Start the Nohup using the command: `nohup sh orangescrum.sh > customout.log &`

STEP 13

Browse the Orangescrum website <http://YourIpAddress> or server IP address or domain name.
Ex. <http://localhost/orangescrum> or **virtual host**

STEP 14

You will be asked to provide your Company Name, Email address and a Password to login and start using Orangescrum.

STEP 15

Advanced Setup

Google Signup, Google Login, Google Contact

To setup the Google Signup, Google Login, Google Contact, define the following details in the ***app/Config/constants.php file***

```
##### Google Keys (Login, Drive, Contacts)
#####
define("CLIENT_ID", "XXXXXXXXXXXX.apps.googleusercontent.com");
define("CLIENT_ID_NUM", "XXXXXXXXXXXX");
define("CLIENT_SECRET", "xXxXxXxxx_xXxXxXxxx");
define("API_KEY", "xXxXxXxxxXXXXXXXXXXXXXXXXxXxXxxx");
define("REDIRECT_URI", HTTP_ROOT . "users/googleConnect");
define("USE_GOOGLE", 0); //Set this parameter to 1, to use Google Login, Drive and Contacts
```

Dropbox Setup

```
##### Dropbox Key #####
define("DROPBOX_KEY", "xXxxXxxxXx");
```

```
define("USE_DROPBOX", 0); //Set this parameter to 1, to use Dropbox file sharing
```

AWS S3 Bucket Setup

```
define('USE_LOCAL', 1);  
##### AWS S3 Bucket #####  
define('USE_S3', 0); //Set this parameter to 1 to use AWS S3 Bucket ("0" for local storage)  
define('BUCKET_NAME', 'Bucket Name');  
define('DOWNLOAD_BUCKET_NAME', 'Download Bucket Name');  
define('awsAccessKey', 'XXXXXXXXXXXX');  
define('awsSecretKey', 'XXX/XXXXXXXXXXXXX/+XXXXXXXXXXXXX');
```

STEP 16

For Installing Node.js

There are 4 simple steps to install Node.js on your server. These are as follows:

1. Install Node.js and NPM
yum install npm node.js
yum --enablerepo=epel install npm node.js
yum install gcc-c++ make
yum install openssl-devel
2. Install socket.io using NPM
npm install socket.io
3. Install Forever
npm install forever -g
4. Run the notification.js (given with the In-App chat) file forever using the "Forever". (it should not stop)
5. Upload the node js file and go to that path
cd /var/www/html/orangescrum

Note: You can find the notification.js file in Orangescrum folder

- After installed all the above step please copy and paste the "notification.js" file (given with the chat add-on) in the same directory where the node_modules is installed
- **Run the below command to start node.js.**

forever start notification.js

Now node server is ready. You can test by checking the below URL

<http://your-domain:3002>

Ex. <http://localhost:3002>

<http://127.0.0.1:3002>

<http://myorange.com:3002>

If output is "Welcome to socket.io." then node.js is working fine and you can use that url in the constant.php as "NODE_HOST"

Note: you can find notification.js, node-js-installation-linux.pdf, node-js-installation-windows.pdf file in Orangescrum folder after install the In-App chat.

STEP 17

Profile Setting

- Go to Setting, click on My profile to set up your profile photo.
- Select Time zone for your account
- Click on Update to save the changes

Notifications Setting

- Go to Settings
- Click on Notifications under Profile Setting
- Select **"Yes"** to get notifications in your email inbox

- Click on Email Reports from the Personal setting
- Select Yes to get Email Reports

Company Setting

- Go to Setting
- Select **My Company** from the Company Settings
- Upload the **company logo** and click on Update to save changes

For Any Queries, Contact us: support@orangescrum.com