

**Surbo Chat Server Installation Guide**

**Ver: 2.3**

**Version History:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version No.** | **Release Date** | **Change Description** | **Updated By** | **Approved By** |
| 1.0 | 31-08-2019 | Initial Draft | Kanwar Pal | CTO |
| 2.0 | 18-04-2020 | Updated for latest versions | Kanwar Pal | CTO |
| 2.1 | 30-04-2020 | Pre-requisite section added | Mahender | CTO |
| 2.2 | 05-04-2020 | Updated | Kanwar Pal | CTO |
| 2.3 | 22-06-2020 | Updated (Mysql user permission) | Kanwar Pal | CTO |

**TABLE OF CONTENTS**

[1. Installation Pre-Requisites 4](#_Toc39594571)

[1.1. Software Versions 4](#_Toc39594572)

[1.2. Domain List 4](#_Toc39594573)

[1.3. Service & Port 4](#_Toc39594574)

[1.4. SSH User with Permission 4](#_Toc39594575)

[1.5. Mysql User 4](#_Toc39594576)

[1.6. Host list configured in /etc/hosts file 4](#_Toc39594577)

[2. Install Apache 5](#_Toc39594578)

[3. Install MySQL 5.7 6](#_Toc39594579)

[4. Install PHP 7.2 7](#_Toc39594580)

[5. Install Node.js and NPM Packages 8](#_Toc39594581)

[6. Install Redis 8](#_Toc39594582)

[7. Install Supervisor 10](#_Toc39594583)

[8. Install Laravel Echo Server and PM2 10](#_Toc39594584)

[9. Install Composer 11](#_Toc39594585)

[10. Add SSH-RSA Key in BitBucket 11](#_Toc39594586)

# Installation Pre-Requisites

## Software Versions

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Software / OS** | **Version** |
| 1 | Ubuntu | 18.04 LTS |
| 1 | Apache | 2.4.x |
| 2 | MySQL | 5.7.x |
| 3 | PHP | 7.2 |
| 4 | NodeJS | 10.x.x |
| 5 | NPM | 6.x.x |
| 6 | Redis | 5.0.x |
| 7 | Laravel Echo Server |  |

## Domain List

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Domain Name** | **Description** |
| 1 | chat.surbo.io | Livehcat |
| 2 | api.surbo.io | Bot & Chat Interaction |
| 3 | ace.surbo.io | Ace & Chat Interaction |

## Service & Port

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Service** | **Port** |
| 1 | Mysql | 6754 |
| 2 | Redis-server | 6389 |
| 3 | SSH | 1709 |
| 4 | Laravel Echo Server | 6002 |
| 5 | Email Service (Internal purpose) | 587 |

## SSH User with Permission

User : ace\_deploy

Permission : rwx at server document /var/www/html/

## Mysql User

Surbochat

Permission : To write file by mysql user and access apache user at /var/lib/mysql-files/

: Additional permission to read/write/execute stored procedure / function / triggers

Host list configured in /etc/hosts file

<<IP>> mysql.surbo.io

<<IP>> redis.surbo.io

# Install Apache

sudo apt-get update

sudo apt-get install apache2

**Hide Apache Version and Operating System**

sudo vi /etc/apache2/conf-enabled/security.conf

Change value of following keys as below

ServerSignature Off

ServerTokens Prod

**Disable Directory Listing and FollowSymLinks**

sudo vi /etc/apache2/apache2.conf

Put “-” before each tag directive in the line Options Indexes FollowSymLinks to become Options -Indexes -FollowSymLinks as shown below

<Directory /var/www/>

Options -Indexes -FollowSymLinks

AllowOverride None

Require all granted

</Directory>

**Secure Apache using mod\_security and mod\_evasive Modules**

sudo apt install libapache2-mod-security2 -y

sudo systemctl restart apache2

sudo apt install libapache2-mod-evasive -y

Select Internet in mail config

sudo systemctl restart apache2

**Disable TRACE HTTP Request**

sudo vi /etc/apache2/apache2.conf

Trace request is disabled by adding the line **TraceEnable off** in the config file /etc/apache2/apache2.conf

sudo systemctl restart apache2

# Install MySQL 5.7

sudo apt-get install mysql-server mysql-client

sudo mysql\_secure\_installation

|  |
| --- |
| Securing the MySQL server deployment.  Connecting to MySQL using a blank password.  VALIDATE PASSWORD PLUGIN can be used to test passwords  and improve security. It checks the strength of password  and allows the users to set only those passwords which are  secure enough. Would you like to setup VALIDATE PASSWORD plugin?  Press y|Y for Yes, any other key for No: **y**  There are three levels of password validation policy:  LOW Length >= 8  MEDIUM Length >= 8, numeric, mixed case, and special characters  STRONG Length >= 8, numeric, mixed case, special characters and dictionary file  Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: **2**  Please set the password for root here.  New password:  Re-enter new password:  Estimated strength of the password: 100  Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : **y**  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? (Press y|Y for Yes, any other key for No) : **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? (Press y|Y for Yes, any other key for No) : **y**  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? (Press y|Y for Yes, any other key for No) : **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? (Press y|Y for Yes, any other key for No) : **y**  Success.  All done! |

Run the below command to loads the time zone tables in the mysql database.

mysql\_tzinfo\_to\_sql /usr/share/zoneinfo | mysql -uroot -p mysql

# Install PHP 7.2

sudo apt-get install software-properties-common

sudo add-apt-repository ppa:ondrej/php

sudo apt-get update

sudo apt-get install php7.2

php -v

|  |
| --- |
| PHP 7.2.14-1+ubuntu16.04.1+deb.sury.org+1 (cli) (built: Jan 13 2019 10:05:18) ( NTS )  Copyright (c) 1997-2018 The PHP Group  Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies  with Zend OPcache v7.2.14-1+ubuntu16.04.1+deb.sury.org+1, Copyright (c) 1999-2018, by Zend Technologies |

**Install PHP 7.2 modules**

sudo apt-get install php-pear php7.2-curl php7.2-dev php7.2-gd php7.2-mbstring php7.2-zip php7.2-mysql php7.2-xml php7.2-bcmath php7.2-gd php7.2-zip

sudo a2enmod rewrite

sudo systemctl restart apache2

# Install Node.js and NPM Packages

curl -sL https://deb.nodesource.com/setup\_10.x | sudo bash -

sudo apt install nodejs

node -v

v10.15.1

npm -v

6.4.1

# Install Redis

Reference: https://www.digitalocean.com/community/tutorials/how-to-install-and-use-redis

sudo apt-get install build-essential

sudo apt-get install tcl8.5

wget http://download.redis.io/releases/redis-stable.tar.gz

tar xzf redis-stable.tar.gz

cd redis-stable

make

make test

sudo make install

To access the script move into the utils directory:

cd utils

sudo ./install\_server.sh

sudo apt-get install redis-server

sudo vi /etc/redis/redis.conf

supervised directive change from no to system

supervised systemd

Welcome to the redis service installer

This script will help you easily set up a running redis server

Please select the redis port for this instance: [6379]

Selecting default: 6379

Please select the redis config file name [/etc/redis/6379.conf]

Selected default - /etc/redis/6379.conf

Please select the redis log file name [/var/log/redis\_6379.log]

Selected default - /var/log/redis\_6379.log

Please select the data directory for this instance [/var/lib/redis/6379]

Selected default - /var/lib/redis/6379

Please select the redis executable path [/usr/local/bin/redis-server]

Selected config:

Port : 6379

Config file : /etc/redis/6379.conf

Log file : /var/log/redis\_6379.log

Data dir : /var/lib/redis/6379

Executable : /usr/local/bin/redis-server

Cli Executable : /usr/local/bin/redis-cli

Is this ok? Then press ENTER to go on or Ctrl-C to abort.

Copied /tmp/6379.conf => /etc/init.d/redis\_6379

Installing service...

Success!

Starting Redis server...

Installation successful!

sudo service redis\_6379 start

sudo service redis\_6379 stop

You can then access the redis database by typing the following command:

redis-cli

To set Redis to automatically start at boot, run:

sudo update-rc.d redis\_6379 defaults

Securing Redis

By default, Redis server allows connections from anywhere which is insecure. Binding to localhost will restrict access to the server itself and is a good first step to protecting your server.

Open the Redis configuration file for editing:

sudo nano /etc/redis/6379.conf

Locate this line and make sure it is uncommented (remove the `#` if it exists):

bind 127.0.0.1

Some of the commands that are considered dangerous include: FLUSHDB, FLUSHALL, KEYS, PEXPIRE, DEL, CONFIG, SHUTDOWN, BGREWRITEAOF, BGSAVE, SAVE, SPOP, SREM, RENAME, and DEBUG. This is not a comprehensive list, but renaming or disabling all of the commands in that list is a good starting point for enhancing your Redis server’s security.

To enable or disable Redis commands, open the configuration file once more:

sudo vi /etc/redis/redis.conf

rename-command FLUSHDB "XXFLUSHXX"

rename-command FLUSHALL "XXFLUSHALLXX"

rename-command DEBUG "XXDEBUGXX"

# rename-command CONFIG ""

rename-command SHUTDOWN XXSHUTDOWNXX

rename-command CONFIG XXCONFIGXX

#Install recent Redis version with external source package

sudo add-apt-repository ppa:chris-lea/redis-server

sudo apt-get update

sudo apt-get install redis-server

# Install Supervisor

sudo apt-get install supervisor

sudo systemctl enable supervisor

# Install Laravel Echo Server and PM2

sudo npm install -g pm2 laravel-echo-server

sudo mkdir /var/www/echo-server

cd /var/www/echo-server

sudo laravel-echo-server init

|  |
| --- |
| ubuntu@ip-172-31-2-205:~$ laravel-echo-server init  ? Do you want to run this server in development mode? No  ? Which port would you like to serve from? 6002  ? Which database would you like to use to store presence channel members? redis  ? Enter the host of your Laravel authentication server. https://chat.surbo.io  ? Will you be serving on http or https? http  ? Do you want to generate a client ID/Key for HTTP API? No  ? Do you want to setup cross domain access to the API? No  ? What do you want this config to be saved as? laravel-echo-server.json  Configuration file saved. Run laravel-echo-server start to run server. |

cd /var/www/echo-server

sudo vi pm2.json

Add following lines into pm2.json file

{

"name": "Surbo Chat",

"script": "laravel-echo-server",

"args": "start"

}

pm2 start pm2.json

Since everything is ok, we save the setup so pm2 will be able to restart everything once invoked with the resurrect command.

pm2 save

pm2 reset (in order to reload config later)

**Creating a systemd service for PM2**

sudo vim /etc/systemd/system/pm2-ubuntu.service

Add below content and save -

[Unit]

Description=PM2 process manager

Documentation=https://pm2.keymetrics.io/

After=network.target

[Service]

Type=forking

User=ubuntu

LimitNOFILE=infinity

LimitNPROC=infinity

LimitCORE=infinity

Environment=PATH=/usr/bin:/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

Environment=PM2\_HOME=/home/ubuntu/.pm2

PIDFile=/home/ubuntu/.pm2/pm2.pid

ExecStart=/usr/lib/node\_modules/pm2/bin/pm2 resurrect

ExecReload=/usr/lib/node\_modules/pm2/bin/pm2 reload all

ExecStop=/usr/lib/node\_modules/pm2/bin/pm2 kill

[Install]

WantedBy=multi-user.target

sudo systemctl daemon-reload

sudo systemctl enable pm2-ubuntu

# Install Composer

sudo apt-get install composer

# Add SSH-RSA Key in BitBucket

Set up SSH on macOS/Linux