Anonymouse Server Setup Guide

Here are some brief setup guides for initializing the web backend server for handling infrastructural API requests from the Anonymouse mobile client.

Most of the setup instructions (bash commands, etc) are based on Ubuntu Linux 16.04 x64 and above. Other Linux based systems might have slight to medium variations.

* Please execute below commands as root*

Installation

Pre-setup

Purchase an instance from cloud provider such as AWS or Digital Ocean, or setup a private server with public IP address.

Optional Install Ubuntu 16.04 x64 and setup the root user.

Update Ubuntu package repository

```
apt-get update
apt-get update -y
api-get install -y –force-yes software-properties-common
apt-add-repository ppa:nginx/development -y
apt-add-repository ppa:chris-lea/redis-server -y #add redis repo
apt-add-repository ppa:ondrej/apache2 -y #add apache repo
apt-add-repository ppa:ondrej/php -y #add php repo
sudo add-apt-repository 'deb [arch=amd64,i386] http://mirrors.accretive-networks.net/mariadb/repo/10.1/ubuntu
xenial main' #add mariadb(mysql) repo
apt-get update
apt-get install -y –force-yes build-essential curl fail2ban gcc git libmcrypt4 libpcre3-dev
make python2.7 python-pip sendmail supervisor ufw unattended-upgrades unzip whois zsh
```

Initialization setup

Recommended Disable password authentication for SSH login

```
sed -i "/PasswordAuthentication yes/d" /etc/ssh/sshd_config
echo "" | sudo tee -a /etc/ssh/sshd_config
echo "" | sudo tee -a /etc/ssh/sshd_config
echo "PasswordAuthentication no" | sudo tee -a /etc/ssh/sshd_config
ssh-keygen -A
```

```
service ssh restart
```

Create root SSH directory if necessary

```
if [ ! -d /root/.ssh ]
then
mkdir -p /root/.ssh
touch /root/.ssh/authorized_keys
fi
```

Recommended Create an user without root privilege

Create the user with name "sshuser"

```
useradd sshuser
mkdir -p /home/sshuser/.ssh
adduser sshuser sudo
```

Setup bash for the user

```
chsh -s /bin/bash sshuser
cp /root/.profile /home/sshuser/.profile
cp /root/.bashrc /home/sshuser/.bashrc
```

Update custom user password

```
usermod –password $(mkpasswd PASSWORD_HERE) sshuser
```

Enable root ssh key to login into sshuser

```
cp /root/.ssh/authorized_keys /home/sshuser/.ssh/authorized_keys
```

Create a new ssh key pair for the user

```
ssh-keygen -f /home/sshuser/.ssh/id_rsa -t rsa -N ' '
```

Setup directory permission

```
chown -R sshuser:sshuser /home/sshuser
chmod -R 755 /home/sshuser
chmod 700 /home/sshuser/.ssh/id_rsa
```

Recommended Set up firewall

Enable ssh, http and https

```
ufw allow 22
ufw allow 80
ufw allow 443
ufw –force enable
```

PHP related setups

Allow PHP-FPM to restart (makesure username: sshuser matches previous initialization)

```
echo "sshuser ALL=NOPASSWD: /usr/sbin/service php7.1-fpm reload" > /etc/sudoers.d/php-fpm
echo "sshuser ALL=NOPASSWD: /usr/sbin/service php7.0-fpm reload" >> /etc/sudoers.d/php-fpm
echo "sshuser ALL=NOPASSWD: /usr/sbin/service php5.6-fpm reload" >> /etc/sudoers.d/php-fpm
echo "sshuser ALL=NOPASSWD: /usr/sbin/service php5-fpm reload" >> /etc/sudoers.d/php-fpm
```

Install core PHP packages

```
apt-get install -y –force-yes php7.1-cli php7.1-dev \
php7.1-pgsql php7.1-sqlite3 php7.1-gd \
php7.1-curl php7.1-memcached \
php7.1-imap php7.1-mysql php7.1-mbstring \
php7.1-xml php7.1-imagick php7.1-zip php7.1-bcmath php7.1-soap \
php7.1-intl php7.1-readline php7.1-mcrypt
```

Install composer, a php dependency manager

```
curl -sS https://getcomposer.org/installer | php
mv composer.phar /usr/local/bin/composer
```

Miscellaneous PHP configs

```
sudo sed -i "s/error_reporting = .*/error_reporting = E_ALL/" /etc/php/7.1/cli/php.ini
sudo sed -i "s/display_errors = .*/display_errors = On/" /etc/php/7.1/cli/php.ini
sudo sed -i "s/memory_limit = .*/memory_limit = 512M/" /etc/php/7.1/cli/php.ini
sudo sed -i "s/;date.timezone.*/date.timezone = UTC/" /etc/php/7.1/cli/php.ini
```

Config PHP directory permissions

```
chmod 733 /var/lib/php/sessions
chmod +t /var/lib/php/sessions
```

Install Nginx web server & PHP-FPM (PHP handler program)

```
apt-get install -y –force-yes nginx php7.1-fpm
openssl dhparam -out /etc/nginx/dhparams.pem 2048 #prepare for ssl
```

PHP-FPM tweaks (sshuser must match!)

```
sed -i "s/error_reporting = .*/error_reporting = E_ALL/" /etc/php/7.1/fpm/php.ini
sed -i "s/display_errors = .*/display_errors = On/" /etc/php/7.1/fpm/php.ini
sed -i "s/;cgi.fix_pathinfo=1/cgi.fix_pathinfo=0/" /etc/php/7.1/fpm/php.ini
sed -i "s/memory_limit = .*/memory_limit = 512M/" /etc/php/7.1/fpm/php.ini
sed -i "s/;date.timezone.*/date.timezone = UTC/" /etc/php/7.1/fpm/php.ini
sed -i "s/^user = www-data/user = sshuser/" /etc/php/7.1/fpm/pool.d/www.conf
sed -i "s/^group = www-data/group = sshuser/" /etc/php/7.1/fpm/pool.d/www.conf
sed -i "s/;listen.owner.*/listen.owner = sshuser/" /etc/php/7.1/fpm/pool.d/www.conf
sed -i "s/;listen.group.*/listen.group = sshuser/" /etc/php/7.1/fpm/pool.d/www.conf
sed -i "s/;listen.mode.*/listen.mode = 0666/" /etc/php/7.1/fpm/pool.d/www.conf
sed -i "s/;request_terminate_timeout.*/request_terminate_timeout = 60/" /etc/php/7.1/fpm/pool.d/www.conf
sed -i "s/user www-data;/user sshuser;/" /etc/nginx/nginx.conf
sed -i "s/worker_processes.*/worker_processes auto;/" /etc/nginx/nginx.conf
sed -i "s/# multi_accept.*/multi_accept on;/" /etc/nginx/nginx.conf
sed -i "s/# server_names_hash_bucket_size.*/server_names_hash_bucket_size 128;/" /etc/nginx/nginx.conf
```

Config Nginx Gzip compressed transportation

```
cat > /etc/nginx/conf.d/gzip.conf << EOF
gzip_comp_level 5;
gzip_min_length 256;
gzip_proxied any;
gzip_vary on;
gzip_types
application/atom+xml
application/javascript
application/json
application/rss+xml
application/vnd.ms-fontobject
application/x-font-ttf
application/x-web-app-manifest+json
application/xhtml+xml
application/xml
font/opentype
image/svg+xml
image/x-icon
text/css
text/plain
text/x-component;

EOF
```

Restart PHP-FPM

```
if [ ! -z "$(ps aux | grep php-fpm | grep -v grep)" ]
then
service php7.1-fpm restart > /dev/null 2>&1
service php7.0-fpm restart > /dev/null 2>&1
service php5.6-fpm restart > /dev/null 2>&1
service php5-fpm restart > /dev/null 2>&1
fi
```

Nginx setup

Reload nginx

```
service nginx restart
service nginx reload
```

Add custom user to www-data group

```
usermod -a -G www-data sshuser
id sshuser
group sshuser
```

Optional NodeJS setup

```
curl –silent –location https://deb.nodesource.com/setup_6.x | bash -
apt-get update
sudo apt-get install -y –force-yes nodejs
npm install -g pm2
npm install -g gulp
npm install -g yarn
```

MySQL (MariaDB) setup

```
export DEBIAN_FRONTEND=noninteractive
```

Set up automated database root password

```
debconf-set-selections <<< "mariadb-server-10.0 mysql-server/data-dir select """
debconf-set-selections <<< "mariadb-server-10.0 mysql-server/root_password password PASSWORD_HERE"
debconf-set-selections <<< "mariadb-server-10.0 mysql-server/root_password_again password PASSWORD_HERE"
```

Install mysql

```
apt-get install -y mariadb-server
```

Grant permission for root and custom user (password same as above)

```
sed -i '/^bind-address/s/bind-address.=./bind-address = */' /etc/mysql/my.cnf
mysql –user="root" –password="PASSWORD_HERE" -e "GRANT ALL ON . TO root@'SERVER_IP' IDENTIFIED BY 'PASSWORD_HERE';"
mysql –user="root" –password="PASSWORD_HERE" -e "GRANT ALL ON . TO root@'%' IDENTIFIED BY 'PASSWORD_HERE';"
service mysql restart
mysql –user="root" –password="PASSWORD_HERE" -e "CREATE USER 'sshuser'@'SERVER_IP' IDENTIFIED BY
```

```
                                      'PASSWORD_HERE';"
mysql –user="root" –password="PASSWORD_HERE" -e "GRANT ALL ON . TO 'sshuser'@'SERVER_IP' IDENTIFIED
BY 'PASSWORD_HERE' WITH GRANT OPTION;"
mysql –user="root" –password="PASSWORD_HERE" -e "GRANT ALL ON . TO 'sshuser'@'%' IDENTIFIED BY
'PASSWORD_HERE' WITH GRANT OPTION;"
mysql –user="root" –password="PASSWORD_HERE" -e "FLUSH PRIVILEGES;"
```

Set MySQL Charset

```
echo "" >> /etc/mysql/my.cnf
echo "[mysqld]" >> /etc/mysql/my.cnf
echo "character-set-server = utf8" >> /etc/mysql/my.cnf
```

Other stuff

Install Redis for cache and queue

```
apt-get install -y redis-server
sed -i 's/bind 127.0.0.1/bind 0.0.0.0/' /etc/redis/redis.conf
service redis-server restart
```

Setup Supervisor for automated process management

```
systemctl enable supervisor.service
service supervisor start
```

Website Setup

Basically follow up starting from Step 3 in this tutorial. with a few exceptions:

- Replace *quickstart* with the name of our application, possibly *Anonymouse.* for every single occasion it appears.
- Git clone my repositoryhttps://github.com/ReactiveXYZ-Dev/Anonymouse-Api
- Before setting the DB_DATABASE, DB_USERNAME, and DB_PASSWORD in the .env file, make sure to create an initial database for our application with command like:

```
mysql –user="root" –password="MYSQL_ROOT_PASSWORD" -e "CREATE DATABASE anonymouse;"
```

- Make above DB configuration matches the settings during the setup of MySQL database.

- In the .env file, make sure below are set to enable debug mode:

```
APP_DEBUG=true
APP_ENV=local
```

- In the .env file, append these additional lines to enable additional features:

```
CACHE_DRIVER=redis
SESSION_DRIVER=redis
QUEUE_DRIVER=redis
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379
```

- Replace *example.com* with the actual domain of the website (assuming DNS has been configured correctly.) for every single occasion it appears.

- ***Notice:*** A tutorial about setting up TLS encryption (SSL) is also included in the end of the tutorial, please make sure you read it!

Initialization Setup

SSH into the server

```
ssh sshuser@IP_ADDRESS -i RSA_PRIVATE_KEY
```

Boot up redis

```
redis-server
```

Go to project directory and bootup queue listener

```
cd /var/www/html/PROJECT_NAME & php artisan queue:work
```

Support

Please contact jackie@reactive.xyz for assistance and support.