**INSTALL SERVER ODCR55 di Linux Ubuntu/Kubuntu 19.04**

**Step 1: Setup PHP with Apache2 HTTP Server**

To use PHP with Apache2 HTTP server, you should first install Apache2… To do that, run the commands below:

sudo apt update

sudo apt install apache2

Allow Firewall

Check the available ufw application profiles:

sudo ufw app list

Output

Available applications:

Apache

Apache Full

Apache Secure

OpenSSH

Let’s enable the most restrictive profile that will still allow the traffic you’ve configured, permitting traffic on port 80 (normal, unencrypted web traffic):

sudo ufw allow 'Apache'

Verify the change:

sudo ufw enable

sudo ufw status

Output

Status: active

To Action From

-- ------ ----

OpenSSH ALLOW Anywhere

Apache ALLOW Anywhere

OpenSSH (v6) ALLOW Anywhere (v6)

Apache (v6) ALLOW Anywhere (v6)

After installing Apache2, the commands below can be used to stop, start and enable Apache2 service to always startup when the server boots up.

sudo systemctl stop apache2.service

sudo systemctl start apache2.service

sudo systemctl enable apache2.service

#### Step 2: Install PHP 7.3 with Apache2 Support

PHP 7.3 may not be available in Ubuntu default repositories… in order to install it, you will have to get it from third-party repositories.

Run the commands below to add the below third-party repository to upgrade to PHP / PHP-FPM 7.3

sudo apt-get install software-properties-common

sudo add-apt-repository ppa:ondrej/php

Then update and upgrade to PHP 7.3

sudo apt update

Next, run the commands below to install PHP 7.3 and related modules.

There are many PHP modules that perform different functions…. however, these are some popular ones that may be needed when developing PHP based websites…

sudo apt-get install php7.3 libapache2-mod-php7.3 php7.3-cli php7.3-mysql php7.3-gd php7.3-imagick php7.3-recode php7.3-tidy php-mongodb php7.3-xmlrpc

The line above will allow PHP to function with many popular PHP based websites and applications.

#### Step 3: Install MariaDB Database Server

MariaDB stands for M in LAMP and it’s a great place to start when looking at open source database server… Although MySQL was originally the default database server among Linux systems, MariaDB has taken over.. To install it run the commands below.

sudo apt-get install mariadb-server-10.3 mariadb-client

\*jika gagal install, sertakan versi yang direkomendasikan

After installing MariaDB database server, the commands below can be used to stop, start and enable MariaDB service to always start up when the server boots..

**On Ubuntu 18.04 LTS and 18.10**

sudo systemctl stop mariadb.service

sudo systemctl start mariadb.service

sudo systemctl enable mariadb.service

After that, run the commands below to secure MariaDB server by creating a root password and disallowing remote root access.

sudo mysql\_secure\_installation

When prompted, answer the questions below by following the guide.

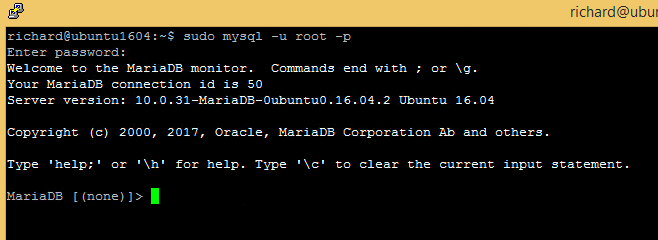
* Enter current password for root (enter for none): Just press the Enter
* Set root password? [Y/n]: Y
* New password: Enter password
* Re-enter new password: Repeat password
* Remove anonymous users? [Y/n]: Y
* Disallow root login remotely? [Y/n]: Y
* Remove test database and access to it? [Y/n]:  Y
* Reload privilege tables now? [Y/n]:  Y

Restart MariaDB server

To test if MariaDB is installed, type the commands below to logon to MariaDB server

sudo mysql -u root –p

Then type the password you created above to sign on… if successful, you should see MariaDB welcome message



#### Step 4: Install phpMyAdmin

Now that LAMP is in place, run the commands below to install phpMyAdmin…

sudo apt install phpmyadmin

When prompted to choose the web server, don’t select **apache2**…

+------------------------+ Configuring phpmyadmin +-------------------------+

| Please choose the web server that should be automatically configured to |

| run phpMyAdmin. |

| |

| Web server to reconfigure automatically: |

| |

| [x] apache2 |

| [ ] lighttpd |

| |

| |

| <ok> |

| |

+---------------------------------------------------------------------------+

When prompted again to allow debconfig-common to install a database and configure select Yes.

+------------------------+ Configuring phpmyadmin +-------------------------+

| |

| The phpmyadmin package must have a database installed and configured |

| before it can be used. This can be optionally handled with |

| dbconfig-common. |

| |

| If you are an advanced database administrator and know that you want to |

| perform this configuration manually, or if your database has already |

| been installed and configured, you should refuse this option. Details |

| on what needs to be done should most likely be provided in |

| /usr/share/doc/phpmyadmin. |

| |

| Otherwise, you should probably choose this option. |

| |

| Configure database for phpmyadmin with dbconfig-common? |

| |

| <Yes> <No> |

| |

+---------------------------------------------------------------------------+

You will be prompted to create a password for phpMyAdmin to register with the database… Please provide a password for phpmyadmin to register with the database…   After that, phpMyAdmin should be installed and ready to use..

#### Step 5: Connecting to phpMyAdmin

Now open your browser and browse to the hostname followed by **phpmyadmin**

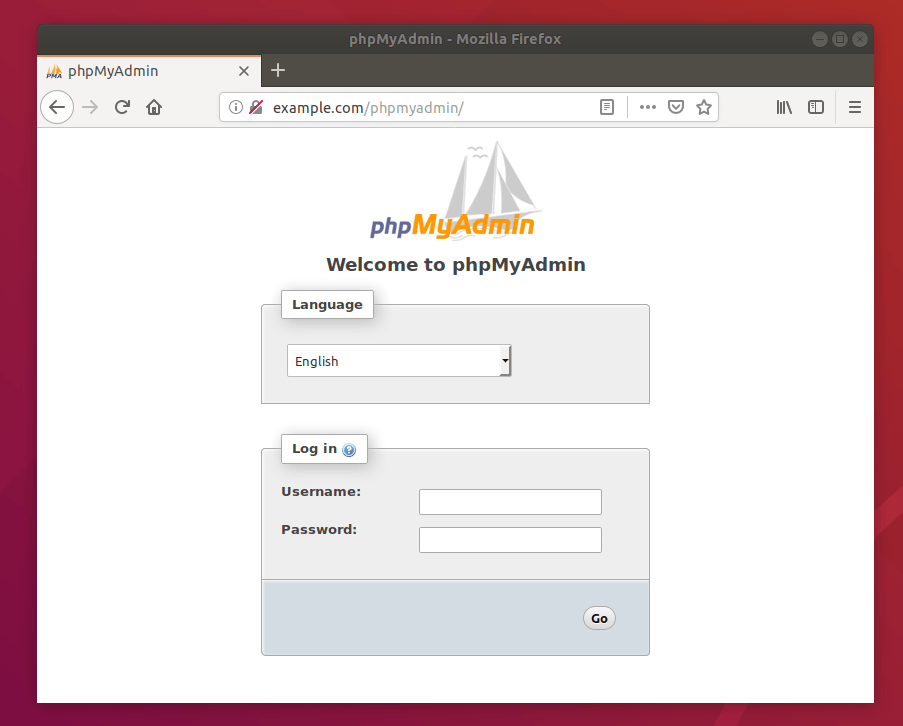
**http://localhost /phpMyAdmin**

**\*Jika phpMyAdmin tidak bisa diakses (error not found)**

**Sudo nano /etc/apache2/apache2.conf**

**Kemudian copy paste syntak berikut:**

**Include /etc/phpMyAdmin/apache.conf**



When you attempt to logon using MariaDB root account it will fail… That’s because MariaDB and MySQL have switch their authentication method to **auth\_socket**

The **auth\_socket** plugin authenticates users that connect from the localhost through the Unix socket file… which prevents users from connecting with password… So, you won’t be able to connecto via phpMyAdmin…

When you attempt to logon, you see the error “#1698 – Access denied for user ‘root’@’localhost’”

To fix that, run the commands below:

sudo mysql -u root -p

That should get you into the database server. After that, run the commands below to disable plugin authentication for the root user

Create database odcr55\_v3\_local\_dev;

use mysql;

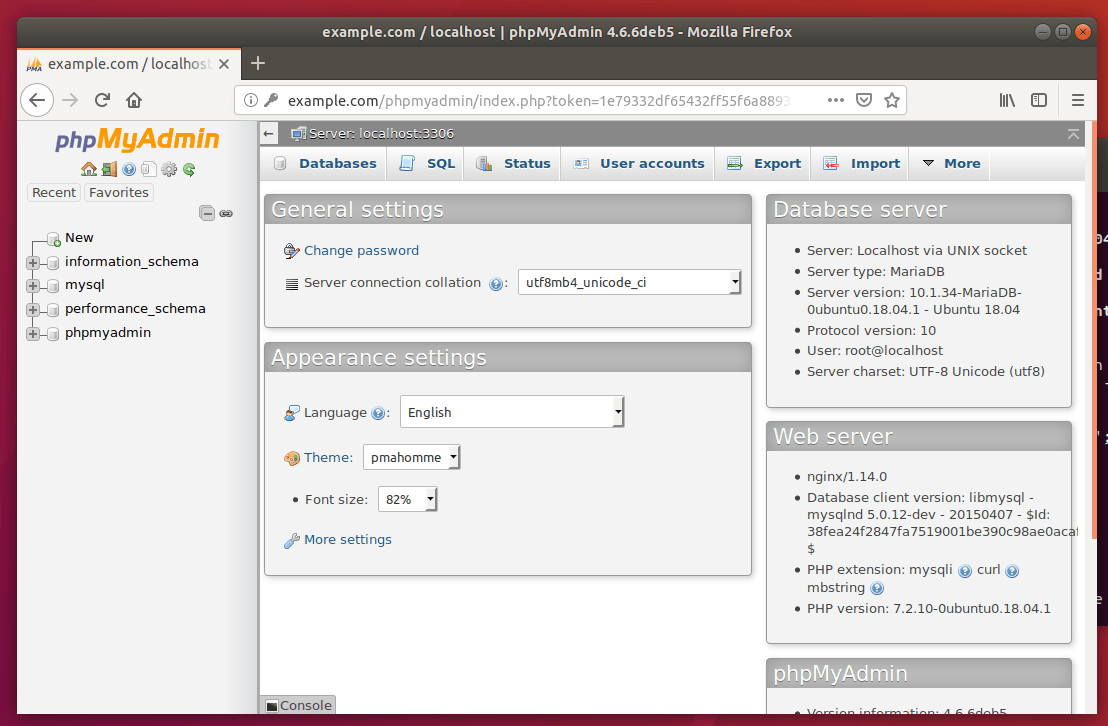
update user set plugin=’’ where User=’root’;

flush privileges;

exit

Restart and run the commands below to set a new password.

sudo systemctl restart mariadb.service



#### Step 6 — Installing MongoDB

First, update the packages list to have the most recent version of the repository listings:

sudo apt update

Now install the MongoDB package itself:

sudo apt install -y mongodb

This command installs several packages containing the latest stable version of MongoDB, along with helpful management tools for the MongoDB server. The database server is automatically started after installation.

Next, let’s verify that the server is running and works correctly.

#### Step 7 — Checking the Service and Database

The installation process started MongoDB automatically, but let’s verify that the service is started and that the database is working.

First, check the service’s status:

sudo systemctl status mongodb

You’ll see this output:

Output

mongodb.service - An object/document-oriented database

Loaded: loaded (/lib/systemd/system/mongodb.service; enabled; vendor preset: enabled)

Active: active (running) since Sat 2018-05-26 07:48:04 UTC; 2min 17s ago

Docs: man:mongod(1)

Main PID: 2312 (mongod)

Tasks: 23 (limit: 1153)

CGroup: /system.slice/mongodb.service

└─2312 /usr/bin/mongod --unixSocketPrefix=/run/mongodb --config /etc/mongodb.conf

According to systemd, the MongoDB server is up and running.

We can verify this further by actually connecting to the database server and executing a diagnostic command

Execute this command:

mongo --eval 'db.runCommand({ connectionStatus: 1 })'

This will output the current database version, the server address and port, and the output of the status command:

Output

MongoDB shell version v3.6.3

connecting to: mongodb://127.0.0.1:27017

MongoDB server version: 3.6.3

{

"authInfo" : {

"authenticatedUsers" : [ ],

"authenticatedUserRoles" : [ ]

},

"ok" : 1

}

A value of 1 for the ok field in the response indicates that the server is working properly.

Next, we’ll look at how to manage the server instance.

#### Step 8 — Managing the MongoDB Service

MongoDB installs as a systemd service, which means that you can manage it using standard systemd commands alongside all other sytem services in Ubuntu.

To verify the status of the service, type:

sudo systemctl status mongodb

You can stop the server anytime by typing:

sudo systemctl stop mongodb

To start the server when it is stopped, type:

sudo systemctl start mongodb

## **Step 9 -- Installing Git with Default Packages**

First, use the apt package management tools to update your local package index. With the update complete, you can download and install Git:

sudo apt update

sudo apt install git

You can confirm that you have installed Git correctly by running the following command:

git --version

Output

git version 2.17.1

With Git successfully installed, you can now move on to the [Setting Up Git](https://www.digitalocean.com/community/tutorials/how-to-install-git-on-ubuntu-18-04#setting-up-git) section of this tutorial to complete your setup.

#### Step 10 – Install Composer

## Step 1 — Installing the Dependencies

Before you download and install Composer, you’ll want to make sure your server has all dependencies installed.

First, update the package manager cache by running:

sudo apt update

Now, let’s install the dependencies. We’ll need curl in order to download Composer and php-cli for installing and running it. The php-mbstring package is necessary to provide functions for a library we’ll be using. git is used by Composer for downloading project dependencies, and unzip for extracting zipped packages. Everything can be installed with the following command:

sudo apt install curl php-cli php-mbstring git unzip

With the prerequisites installed, we can install Composer itself.

## Step 2 — Downloading and Installing Composer

Composer provides an [installer](https://getcomposer.org/installer), written in PHP. We’ll download it, verify that it’s not corrupted, and then use it to install Composer.

Make sure you’re in your home directory, then retrieve the installer using curl:

cd ~

curl -sS https://getcomposer.org/installer -o composer-setup.php

Next, verify that the installer matches the SHA-384 hash for the latest installer found on the [Composer Public Keys / Signatures](https://composer.github.io/pubkeys.html) page. Copy the hash from that page and store it as a shell variable:

HASH=544e09ee996cdf60ece3804abc52599c22b1f40f4323403c44d44fdfdd586475ca9813a858088ffbc1f233e9b180f061

Make sure that you substitute the latest hash for the highlighted value.

Now execute the following PHP script to verify that the installation script is safe to run:

php -r "if (hash\_file('SHA384', 'composer-setup.php') === '$HASH') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP\_EOL;"

You’ll see the following output.

Output

Installer verified

If you see Installer corrupt, then you’ll need to redownload the installation script again and double check that you’re using the correct hash. Then run the command to verify the installer again. Once you have a verified installer, you can continue.

To install composer globally, use the following command which will download and install Composer as a system-wide command named composer, under /usr/local/bin:

sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer

You’ll see the following output:

Output

All settings correct for using Composer

Downloading...

Composer (version 1.6.5) successfully installed to: /usr/local/bin/composer

Use it: php /usr/local/bin/composer

To test your installation, run:

composer

And you’ll see this output displaying Composer’s version and arguments.

Output

\_\_\_\_\_\_

/ \_\_\_\_/\_\_\_ \_\_\_\_ \_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_

/ / / \_\_ \/ \_\_ `\_\_ \/ \_\_ \/ \_\_ \/ \_\_\_/ \_ \/ \_\_\_/

/ /\_\_\_/ /\_/ / / / / / / /\_/ / /\_/ (\_\_ ) \_\_/ /

\\_\_\_\_/\\_\_\_\_/\_/ /\_/ /\_/ .\_\_\_/\\_\_\_\_/\_\_\_\_/\\_\_\_/\_/

/\_/

Composer version 1.6.5 2018-05-04 11:44:59

Usage:

command [options] [arguments]

Options:

-h, --help Display this help message

-q, --quiet Do not output any message

-V, --version Display this application version

--ansi Force ANSI output

--no-ansi Disable ANSI output

-n, --no-interaction Do not ask any interactive question

--profile Display timing and memory usage information

--no-plugins Whether to disable plugins.

-d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.

-v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

. . .

#### Step 11 – Add Node.js PPA

Node.js package is available in LTS release and the current release. It’s your choice to select which version you want to install on the system as per your requirements. Let’s add the PPA to your system to install Nodejs on Ubuntu.

**Use LTS Release :** At the last update of this tutorial, Node.js 10.15.3 is the LTS release available.

sudo apt-get install curl

curl -sL https://deb.nodesource.com/setup\_12.x | sudo -E bash -

For this tutorial, **I am using the latest current release** and added their PPA to my system.

#### Step 12 – Install Node.js on Ubuntu

You can successfully add Node.js PPA to Ubuntu system. Now execute the below command install Node on and Ubuntu using apt-get. This will also install NPM with node.js. This command also installs many other dependent packages on your system.

sudo apt-get install nodejs

#### Step 13 – Check Node.js and NPM Version

After installing node.js verify and check the installed version. You can find more details about current version on node.js [official website](https://nodejs.org/download/).

**node -v**

v12.10.0

Also, check the npm version

**npm -v**

6.10.3

#### Step 14 – Cloning Git Project

#### First open apache directory root with this command :

cd /var/www/

after that use git clone to download the project ODCR55\_V3. Run this command :

git clone https://gitlab-ci-token:2FWs5SbETAgZz5kgvH8E@gitlab.com/ws-project/proker-2019/cr55\_v3.git ODCR55\_V3

Wait until download finished.

#### Step 15 - Configuring Apache2

Now that we have installed Laravel, we move on to the step of configuring Apache webserver.

Next step is to give proper permissions to the project directory. For this, we need to enable access to it from the www-data group and to give it write permissions to the storage directory.

*sudo chown -R www-data:root /var/www/ODCR55\_V3*

*sudo chmod -R 755 /var/www/ODCR55\_V3*

Now go to the /etc/apache2/sites-available directory and use the following command to create a configuration file for our Laravel install.

*cd /etc/apache2/sites-available  
sudo nano laravel.conf*

Now add the following content to the file, copy and close it after saving.

<VirtualHost \*:80>

ServerAdmin webmaster@localhost

DocumentRoot /var/www/ODCR55\_V3/public

<Directory /var/www/ODCR55\_V3>

AllowOverride All

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Now we have to enable this newly created .conf file and disable the default .conf file that is installed with the default Apache install. Also, we need to enable mod\_rewrite so that permalinks can function properly.

*sudo a2dissite 000-default.conf  
sudo a2ensite laravel.conf  
sudo a2enmod rewrite  
sudo service apache2 restart*

#### Step 16 - Keeping the socket server running with supervisord

The websockets:serve daemon needs to always be running in order to accept connections. This is a prime use case for supervisor, a task runner on Linux.

First, make sure supervisor is installed.

# On Debian / Ubuntu

sudo apt install supervisor

sudo systemctl enable supervisor

Once installed, add a new process that supervisor needs to keep running. You place your configurations in the.

Open this directory :

cd /etc/supervisor/conf.d

Within that directory, create a new file called websockets.conf by run :

sudo nano websockets.conf

Then copy this content to file and save.

[program:websockets]

command=/usr/bin/php /var/www/ODCR55\_V3/artisan websockets:serve

numprocs=1

autostart=true

autorestart=true

Once created, instruct supervisor to reload its configuration files (without impacting the already running supervisor jobs).

sudo supervisorctl update

sudo supervisorctl start websockets

#### Step 17 – Setting and running project ODCR55

1. Open directory project :

cd /var/www/ODCR55\_v3

1. Checkout Production

sudo git checkout production

1. Run Composer

sudo composer install

1. Run NPM

sudo npm install

1. Buat database \*jika belum ada

* Akses localhost/phpMyAdmin
* Username:root password: odcr55
* New database : odcr55\_v3\_local\_dev

1. Copy file .env.example to .env

sudo cp .env.example .env

1. Setup .ENV dan Client ID waroeng

sudo nano .env

APP\_CLIENT\_ID=2706bdc6-cb1f-11e9-95fe-92d6d58b4819

Client ID menyesuaikan code tiap waroeng. Untuk mendapatkan client id akses app.pedasabis.com

APP\_LINUX=true

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=odcr55\_v3\_local\_dev

DB\_USERNAME=root

DB\_PASSWORD=odcr55

1. Run artisan

sudo php artisan key:generate

sudo php artisan config:cache

sudo php artisan config:clear

sudo composer dump-autoload

1. Run migration table

sudo php artisan migrate:refresh -> migrate untuk menghapus database dan meng create ulang database.

sudo php artisan migrate -> ini migrate biasa untuk update database

1. Download data master dari cloud

* open localhost/fetch on web browser
* Klik fetch cloud to local

1. Duplikat data master ke database mongo

Akses localhost/mongo on web browser

1. Tambahkan Hak Akses Printer. \*membutuhkan restart

Sudo usermod -a -G lp www-data

#### Step 18 – Update Patch ODCR55 \* jika diperlukan

* + - 1. Masuk ke directory ODCR55

Sudo cd /var/www/ODCR55\_V3

* + - 1. Hapus file sampah

Sudo git reset --hard

* + - 1. Download patch ODCR55 terbaru

Sudo git pull origin production

* + - 1. Migration database \*jika ada perubahan database

Sudo php artisan migrate

* + - 1. Fetch ulang data master

open localhost/fetch on web browser

Klik fetch cloud to local -> tunggu hingga selesai download

Akses localhost/mongo on web browser