

# Installing RADIUSdesk on Ubuntu 22.04 using Nginx

## Skills required for the installation

To install RADIUSdesk, you need sufficient knowledge and experience with Linux:

- Installing the Linux operating system.
- Edit text files from the terminal using a text editor such as Vi or Nano.
- Install packages from a repository.
- You must be familiar with how TCP/IP networks work.

## Background

- Nginx is a web server that seems to have overtaken Apache in terms of popularity and number of active websites on the Internet.
- It is new, lightweight, fast, highly scalable and capable of handling a large load without overloading your system.
- Nginx is the new Apache so to speak.
- This section describes the steps you need to take to get RADIUSdesk up and running with a LEMP stack on Ubuntu 22.04
  - A LEMP stack is one of those acronyms you can use to impress your friends. It stands for Linux NginX MySQL and PHP.



- We recently switched from CakePHPv3 to CakePHPv4.
- If you manage devices with MESHdesk and APdesk, make sure you also run the instructions that include cake3 so that your system is backwards compatible.
- The firmware on these devices might still point to the cake3 directory, so we need to include support for this directory.

## What do we need

- A standard Nginx installation on Ubuntu is actually very simple.
- The more complicated part is customising Nginx for the following tasks:

Requirement	Comment
Interpreting PHP scripts	We want the web server to call the PHP interpreter when a page with the .php extension is requested.
Access MySQL functions from PHP	Since we are setting up a LEMP server, we need to install a MySQL server and access it from PHP. We will install MariaDB, which is a direct replacement for MySQL.
Change the expiration date of HTTP headers to encourage caching.	We want files that do not change (e.g. CSS or images) to be cached on the client side to make the client experience more pleasant

Requirement	Comment
Compress the text before it is served to the client.	We can compress the text that flows between the client and the server and in this way reduce the number of bytes transmitted over the wire, which in turn should provide a more pleasant experience for the client
Enable rewrite rules in CakePHP for pretty URLs	CakePHP uses the .htaccess files in Apache to enable pretty URLs. Since Nginx does not support .htaccess files, we need to modify Nginx to behave the same way.

## HOWTO



- Please note that the behavior of the shell has changed in Ubuntu 22.04.
- It seems that it does NOT execute the multi-line insertion of commands.
- So unfortunately you have to copy these commands line by line.

### Add a sudo user

- We assume that you have installed a clean installation of Ubuntu 22.04 WITHOUT Apache.
- If you have not yet added a sudo user add one now.

```
# Add the system user
sudo adduser system
# Update the system to the latest
sudo usermod -aG sudo system
```

### Introduction to network technology under Ubuntu 22.04

- If you do not yet have a working network configuration on the server on which you want to perform the installation, please use this section as a reference, otherwise simply continue with the next section.
- Since there is a big difference between Ubuntu 16.04 and Ubuntu 22.04, we feel that this section will help those who need to get used to the new way of working.
- We assume that you have a bare VM (like the one from <https://www.osboxes.org/ubuntu-server/>)
- We also assume that you have used it to create a VM in Virtualbox and now only see the local loopback interface (127.0.0.1) when you enter the ip a command.
- To see which interfaces are available (although some may not yet be configured)

```
ip a
```

- On my system it lists three since I plan to use the VM also as a router with Coova Chilli running on the one interface. So we have **lo**, **enp0s3** and **enp0s8**.
- For now I will just configure both of those interfaces to be DHCP clients.

```
sudo vi /etc/netplan/50-cloud-init.yaml
```

- We edit the file to look like this (adapt to fit your system's interfaces)

```
# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  version: 2
  ethernets:
    enp0s3:
      addresses: []
      dhcp4: true
      optional: true
    enp0s8:
      addresses: []
      dhcp4: true
      optional: true
```

- Apply the network configuration using command:

```
sudo netplan --debug apply
```

- If all went well our VM will now have an IP Address (via DHCP) which we can use.

## ip addr

*#Feedback contains*

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether 08:00:27:fe:57:09 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.111/24 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 255675sec preferred_lft 255675sec
    inet6 fe80::a00:27ff:fefe:5709/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether 08:00:27:8c:d3:32 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a00:27ff:fe8c:d332/64 scope link
        valid_lft forever preferred_lft forever
```

- Now that we have a working network setup on our machine we can continue.

## Install Nginx

- We assume you have a clean install of Ubuntu 22.04 **WITHOUT** Apache installed.



- To remove Apache

```
sudo systemctl stop apache2.service  
sudo apt-get -y remove apache2
```

- Make sure it is up to date.

```
# Get the latest package lists  
sudo apt-get update  
# Update the system to the latest  
sudo apt-get upgrade
```

- Ensure the English language pack is installed

```
sudo apt-get -y install language-pack-en-base
```

- Install Nginx

```
sudo apt-get -y install nginx
```

- Ensure the web server starts up and is running

```
sudo systemctl stop nginx.service  
sudo systemctl start nginx.service
```

- Navigate to the IP Address of the server where you installed **Nginx** using a browser to ensure Nginx serves content e.g. <http://127.0.0.1>

## Configure Nginx to interpret .php files

### php-fpm

- The default installation of Nginx does not support serving .php files.
- We will install a program (actually a service) called **php-fpm**.
- This service waits for requests for interpretation.
- Install the php-fpm service by installing the default version 8.1 of the packages

```
sudo apt-get -y install php-fpm  
sudo systemctl enable php8.1-fpm  
sudo systemctl start php8.1-fpm
```

### Modify Nginx

- Now that the php-fpm service is installed, let's modify the default Nginx server to use it.
- Edit the default server file:

```
sudo vi /etc/nginx/sites-enabled/default
```

- Add *index.php* to this line:

```
# Add index.php to the list if you are using PHP
index index.php index.html index.htm index.nginx-debian.html;
```

- Enable PHP processing by leaving this section uncommented. Note that we are using the UNIX socket and that we are using 8.1 and not 7.4, which was originally specified in the configuration file.

```
# pass PHP scripts to FastCGI server
#
location ~ /\.php$ {
    include snippets/fastcgi-php.conf;
    #
    #    # With php-fpm (or other unix sockets):
    fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
    #    # With php-cgi (or other tcp sockets):
    #    fastcgi_pass 127.0.0.1:9000;
}
```

- Activate the hiding of .htaccess files.

```
# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
location ~ /\.ht {
    deny all;
}
```

- Reload the configuration of the Nginx web server

```
sudo systemctl reload nginx.service
```

- Create a test .php file to confirm that it does work

```
sudo vi /var/www/html/test.php
```

- Contents:

```
<?php
    phpinfo();
?>
```

- Navigate to <http://127.0.0.1/test.php> and check that the page displays the PHP information.

---

## Install MariaDB

### Why MariaDB?

- We have found that the version of MySQL that is delivered by default with Ubuntu 22.04 causes problems with RADIUSdesk.
- For this reason, we have installed MariaDB as an alternative.
- MariaDB is an open-source relational database management system that is often used as an

alternative to MySQL as the database part of the popular LAMP stack (Linux, Apache, MySQL, PHP/Python/Perl).

- It is intended as a drop-in replacement for MySQL.
- Make sure to provide a root password for the MariaDB database when asked for it if you are security conscious, otherwise just press the ESC key.

```
sudo apt-get -y install mariadb-server php8.1-mysql
sudo systemctl enable mariadb
sudo systemctl restart mariadb
sudo systemctl status mariadb
```

## Disable strict mode

- With Ubuntu 22.04, the bundled version of MariaDB is on version 10.3, which has introduced some strict modes that have some issues with the RADIUSdesk database implementation.
- We will disable the strict SQL mode in MariaDB by creating a new file `/etc/mysql/conf.d/disable_strict_mode.cnf`

```
sudo vi /etc/mysql/conf.d/disable_strict_mode.cnf
```

- Enter these two lines:

```
[mysqld]
sql_mode=IGNORE_SPACE,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION
```

- Save the file and restart the MySQL Server

```
sudo systemctl restart mariadb
```

---

## Performance tune Nginx

### Modify expiry date for certain files

- Edit the `/etc/nginx/sites-available/default` file:

```
sudo vi /etc/nginx/sites-available/default
```

- Add the following inside the server section:

```
location ~ ^/cake4/.\.(jpg|jpeg|gif|png|ico|js|css)$ {
    rewrite ^/cake4/rd_cake/webroot/(.*)$ /cake4/rd_cake/webroot/$1 break;
    rewrite ^/cake4/rd_cake/(.*)$ /cake4/rd_cake/webroot/$1 break;
    access_log off;
    expires max;
    add_header Cache-Control public;
}
```

- Add below only if you require backward compatibility (MESHdesk and APdesk).

```
location ~ ^/cake3/.\.(jpg|jpeg|gif|png|ico|js|css)$ {
    rewrite ^/cake3/rd_cake/webroot/(.*)$ /cake3/rd_cake/webroot/$1 break;
    rewrite ^/cake3/rd_cake/(.*)$ /cake3/rd_cake/webroot/$1 break;
    access_log off;
    expires max;
    add_header Cache-Control public;
}
```

- Reload Nginx:

```
sudo systemctl reload nginx.service
```

---

## Install RADIUSdesk

- In the first part, everything was prepared for the installation of **RADIUSdesk**.
- This part explains the steps for installing the latest **RADIUSdesk**.
- RADIUSdesk consists of three components.
  - **rd** directory with its contents contains all the HTML and JavaScript code and is used as the presentation layer.
  - **cake4** is a CakePHPv4 application and can be considered the engine room. Here the data is processed before being presented by the presentation layer.
  - **login** is a directory with various login pages which are centrally managed through the RADIUSdesk **Dynamic Login Pages** applet.
- Later we will create various symbolic links from locations inside the **rdcore** directory to locations inside the web server's document root directory.

## Required packages

- Make sure the following packages are installed.

```
sudo apt-get -y install php-cli php-mysql php-gd php-curl php-xml php-
mbstring php-intl php-sqlite3 git wget
sudo systemctl restart php8.1-fpm
```

- Check out the **rdcore** git repository.

```
cd /var/www
sudo git clone https://github.com/RADIUSdesk/rdcore.git
```

- This will create an **rdcore** directory that contains some subfolders.
- It is recommended that you also include the RD Mobile UI.
- Check out the **rd\_mobile** git repository.

```
cd /var/www
sudo git clone https://github.com/RADIUSdesk/rd_mobile.git
```

## Create softlinks

- We will create softlinks in the directory where Nginx provides the RADIUSdesk content.

```
cd /var/www/html
```

```
sudo ln -s ../rdcore/rd ./rd
sudo ln -s ../rdcore/cake4 ./cake4
#If backward compatibility is required for older firmware of MESHdesk
sudo ln -s ../rdcore/cake4 ./cake3
sudo ln -s ../rdcore/login ./login
sudo ln -s ../rdcore/AmpConf/build/production/AmpConf ./conf_dev
sudo ln -s ../rdcore/cake4/rd_cake/setup/scripts/reporting ./reporting
#For the RD Mobile UI
sudo ln -s ../rd_mobile/build/production/RdMobile ./rd_mobile
```

## Change ownership

- Change the ownership of the following files to www-data so Nginx can make changes to the files/directories

```
sudo mkdir -p /var/www/html/cake4/rd_cake/logs
sudo mkdir -p /var/www/html/cake4/rd_cake/webroot/files/imagecache
sudo mkdir -p /var/www/html/cake4/rd_cake/tmp
sudo chown -R www-data. /var/www/html/cake4/rd_cake/tmp
sudo chown -R www-data. /var/www/html/cake4/rd_cake/logs
sudo chown -R www-data. /var/www/html/cake4/rd_cake/webroot/img/realms
sudo chown -R www-data. /var/www/html/cake4/rd_cake/webroot/img/dynamic_details
sudo chown -R www-data. /var/www/html/cake4/rd_cake/webroot/img/dynamic_photos
sudo chown -R www-data. /var/www/html/cake4/rd_cake/webroot/img/access_providers
sudo chown -R www-data. /var/www/html/cake4/rd_cake/webroot/img/hardwares
sudo chown -R www-data. /var/www/html/cake4/rd_cake/webroot/files/imagecache
```

## The Database

- Make sure that the time zone on the server is set to UTC
- Fill the time zone data in the DB

```
#NOTE FAILING THIS STEP will break the RADIUS graphs
#There might be some error messages in the output which is fine - no need to be alarmed
sudo su
mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -u root mysql
```

- Create an empty database called *rd*

```
sudo su
mysql -u root
create database rd;
GRANT ALL PRIVILEGES ON rd.* to 'rd'@'127.0.0.1' IDENTIFIED BY 'rd';
GRANT ALL PRIVILEGES ON rd.* to 'rd'@'localhost' IDENTIFIED BY 'rd';
exit;
```

- Populate the database:



```
sudo mysql -u root rd < /var/www/html/cake4/rd_cake/setup/db/rd.sql
```



- RADIUSdesk is under active development and sometimes we add SQL patches.
- The SQL patches are located at **/var/www/html/cake4/rd\_cake/setup/db/**.
- These patches are non-destructive and you can run them against the database
- See the sample below

```
sudo mysql -u root rd <
/var/www/rdcore/cake4/rd_cake/setup/db/8.068_add_email_sms_histories.sql
```

## Configure Nginx

- Configure Nginx to rewrite some RdCore URLs starting with */cake4/rd\_cake*.
- Edit */etc/nginx/sites-enabled/default*

```
sudo vi /etc/nginx/sites-enabled/default
```

- Add this once section directly below **server\_name** item. (This is so that this rule is **hit** first for the reporting side. We do not use CakePHP for the reporting anymore due to performance issues.

```
server_name _;
location /cake4/rd_cake/node-reports/submit_report.json {
    try_files $uri $uri/ /reporting/reporting.php;
}
```

- If you need backward compatibility support (MESHdesk and APdesk) also add this section:

```
location /cake3/rd_cake/node-reports/submit_report.json {
    try_files $uri $uri/ /reporting/reporting.php;
}
```

- Add the following configuration block inside the server section (This you can add towards the end):

```
location /cake4/rd_cake {
    rewrite ^/cake4/rd_cake(.$) /cake4/rd_cake/webroot$1 break;
    try_files $uri $uri/ /cake4/rd_cake/index.php$is_args$args;
}
```

- If you need backward compatibility support (MESHdesk and APdesk) also add this section:

```
location /cake3/rd_cake {
    rewrite ^/cake3/rd_cake(.$) /cake3/rd_cake/webroot$1 break;
    try_files $uri $uri/ /cake3/rd_cake/index.php$is_args$args;
}
```

- Reload the Nginx:

```
sudo systemctl reload nginx
```

## Important URLs

- The following URLs are important to reach the UI
- To load the optimized UI, go to <http://127.0.0.1/rd/build/production/Rd/>
- If you want to serve the content directly out of the webroot, do the following:

```
sudo cp -R /var/www/html/rd/build/production/Rd/* /var/www/html/
```

- To load the RD Mobile UI, go to [http://127.0.0.1/rd\\_mobile](http://127.0.0.1/rd_mobile)

## Login Credentials

- By default you can log in with the following credentials

Username: **root** Password: **admin**

---

## Cron scripts

- **RADIUSdesk** requires some scripts that are executed at regular intervals to maintain a healthy and functioning system.
- To activate the cron scripts, execute the following command, which adds the **RADIUSdesk** cron scripts to the cron system

```
sudo cp /var/www/html/cake4/rd_cake/setup/cron/cron4 /etc/cron.d/
```

- If you want to change the default intervals at which the scripts are executed, simply edit the file `/etc/cron.d/cron3`.

## Add LETSENCRYPT certificate

- Instead of repeating the existing documentation, we simply add a URL with the appropriate instructions.
- You may want to do the following first before following the instructions in the URL

```
sudo apt-get update  
sudo apt-get -y install software-properties-common
```

- <https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-22-04>

## Next steps

- Be sure to also install **FreeRADIUS**
- [Install FreeRADIUS](#)

From:

<http://www.radiusdesk.com/wiki/> - **RADIUSdesk**

Permanent link:

[http://www.radiusdesk.com/wiki/getting\\_started/22\\_install\\_ubuntu\\_nginx](http://www.radiusdesk.com/wiki/getting_started/22_install_ubuntu_nginx)

Last update: **2023/11/22 09:58**

