

Installing RADIUSdesk on Ubuntu 20.04 using Nginx



- As from January 2023 we encourage you to run the latest RADIUSdesk code on Ubuntu 22.04 or higher.
- If you however are not able to do for whatever reason, use the URL below for **reference** instructions to replace PHP v7.4 with PHP v8.1 on Ubuntu 20.04.
- <https://www.cloudbooklet.com/how-to-install-or-upgrade-php-8-1-on-ubuntu-20-04/>
- The latest RADIUSdesk code requires a minimum of PHP v8.1.
- **The instructions on this page will not work with the latest RADIUSdesk code since they are for Ubuntu 20.04 and NOT 22.04.**

Skills Required to Install

To install RADIUSdesk you need sufficient knowledge and experience on Linux to:

- Install the Linux operating system
- Edit text files from the terminal using a text editor like **Vi** or **Nano**.
- Install packages from a repository.
- Be comfortable with the working of TCP/IP networking.

Background

- **Nginx** is a web server that seems to have overtaken Apache in terms of popularity and number of active sites on the Internet today.
- It is fresh, lightweight, fast, scales well and is able to take a lot of load without overwhelming your system.
- **Nginx** is the new Apache so to speak.
- This section will cover the steps you have to go through to get RADIUSdesk working with a **LEMP** stack on Ubuntu 20.04
 - * A LEMP stack is one of those acronyms you can impress your friends with. It stands for Linux NginX MySQL and PHP.



- We moved recently from CakePHPv3 to CakePHPv4.
- If you manage devices using MESHdesk and APdesk be sure to also execute the instructions that include cake3 in order for your system to have backwards compatibility.
- The firmware on these devices might still point to the **cake3** directory thus we need to include support for it

What do we require

- A standard **Nginx** install on Ubuntu is actually very simple.
- The part that is more involved is to tweak **Nginx** to do the following:

Requirement	Comment
Interpret PHP Scripts	We would like the web server to call the PHP interpreter when a page ending with .php is requested.
Be able to have access to the MySQL functions of PHP	Since we set up a LEMP server, we need to have a MySQL server installed and accessible from PHP.
Modify the expiry date of HTTP headers to encourage caching	We want files that does not change (e.g. CSS or images) to be cached on the client's side to make the client's experience more pleasant
Compress text before they are served to the client	We can compress the text that flows between the client and the server and in this way reduce the <i>over the line</i> bytes which in turn should also give the client a more pleasant experience
Enable rewrite rules in CakePHP for pretty URL's	CakePHP makes use of the .htaccess files in Apache to enable pretty URLs. Since Nginx does not support .htaccess files, we need to change Nginx to behave in the same way.

HOWTO

Add a sudo user

- We assume you have a clean install of Ubuntu 20.04 WITHOUT Apache installed.
- 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  
usermod -aG sudo system
```

Networking Introduction on Ubuntu 20.04

- If you do not yet have a working network configuration on the server you plan to do the installation on, please use this section as reference, else just proceed to the next section.
- Since there is such a huge difference between the way of doing things in Ubuntu 16.04 and Ubuntu 20.04 we felt that adding this section will help those who are getting used to this newer way of doing things.
- For this we assume you have a bare VM (like the ones from <https://www.osboxes.org/ubuntu-server/>)
- We also assume you used this to create a VM in Virtualbox and are now faced with only the local loopback interface (127.0.0.1) when issuing the **ip a** command.
- To see which interfaces are available (yet some might just 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 20.04 **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 install of **Nginx** does not support the serving of **.php** files.
- We will install a program (actually a service) called **php-fpm**.
- This service will listen for requests to interpret.
- Install the php-fpm service:

```
sudo apt-get -y install php-fpm  
sudo systemctl enable php7.4-fpm  
sudo systemctl start php7.4-fpm
```

Modify Nginx

- Now that the php-fpm service is installed we should change the default **Nginx** server to make use of 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;
```

- Activate PHP processing by un-commenting this this section. Note that we use the UNIX socket:

```
# 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/php7.4-fpm.sock;
    #    # With php-cgi (or other tcp sockets):
    #    fastcgi_pass 127.0.0.1:9000;
}
```

- Enable 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 **Nginx** web server's configuration

```
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 see if the page display the PHP info.

Install MariaDB

Why MariaDB?

- We discovered that the version of MySQL that comes bundled by default with Ubuntu 20.04 are breaking things on RADIUSdesk.
- For this reason we install MariaDB as an alternative.
- MariaDB is an open-source relational database management system, commonly used as an alternative for MySQL as the database portion of the popular LAMP (Linux, Apache, MySQL, PHP/Python/Perl) stack.
- It is intended to be a drop-in replacement for MySQL.
- Be sure to supply a root password for the MariaDB database when asked for it if you are security conscious else simply hit the ESC key.

```
sudo apt-get -y install mariadb-server php-mysql
sudo systemctl enable mariadb
sudo systemctl restart mariadb
sudo systemctl status mariadb
```

Disable strict mode

- With Ubuntu 20.04, the bundled release of MariaDB is at version 10.3 which introduced a few Strict modes which have some problems with RADIUSdesk database implementation.
- We will disable 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

- The first part prepared everything to install **RADIUSdesk**.
- This part will go through the steps to install 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 php7.4-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 containing some sub-folders.

Create soft links

- We will create soft links in the directory where Nginx will serve the RADIUSdesk contents.

```
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
```

Change Ownerships

- 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 the timezone on the server is set to UTC
- Populate the timezone data on 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
```

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/
```

Login Credentials

- By default you can log in with the following credentials

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

Cron Scripts

- **RADIUSdesk** requires a few scripts to run periodically in order to maintain a healthy and working system.
- To activate the cron scripts execute the following command, which will add **RADIUSdesk's** 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 get executed, just edit the `/etc/cron.d/cron3` file.

Add LETSENCRYPT certificate

- Rather than repeating existing documentation we will just add a URL with the instructions to do it.
- You might want to run the following first before going to 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-20-04>

Next steps

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

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

Permanent link:
https://www.radiusdesk.com/wiki/getting_started/20_install_ubuntu_nginx

Last update: **2023/01/12 21:37**

