CryptNDrive Administration Guide

August 23, 2018

Lybero.net

Version : v2.1.1-181-g2e9db11

Summary

1 Introduction

CryptnDrive is a secured web file sharing system providing end to end encryption through native web browser encryption of contents. This manual is split in 3 parts : the architecture of the system, the installation of the web server and finally the configuration of the system.

If you already have an instance and just want to change the configuration parameters, please jump directly to the Configuration chapter.

2 Architecture

The different components of the Transfer system are the following:

- A web access server in charge of communications between other components,
- A database storing all encrypted and unencrypted data (the metadata).
- Users web browsers.

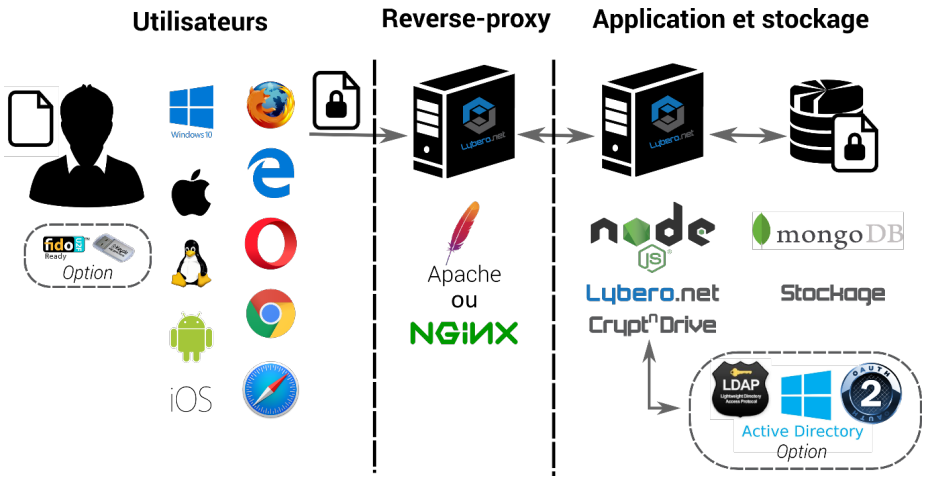


Figure 1: Technical architecture

2.1 Constraints

- Server or VM

  - A linux distribution (redhat, centos, debian, ubuntu, suse, slackware, . . .). Certified on Debian 8 and Centos 7.
  - No specific CPU requirements
  - At least 60MB of RAM per users connected simultaneously
  - Disk according to the expected size of the storage (typically 2GB / user with an account)
  - The server must be able to send mails either through a local smtp server or through a remote one (we detail the configuration after)
  - ssh root access for software installation, the www-data (for debian based distribution) or www (for rpm distributions) account will be used
  - Apache 2.2 or later installed and functional, used in reverse-proxy for access to the nodejs server. Reverse proxy ensures ssl encryption of communications with browsers.
  - A valid SSL certificate for clients' web browsers
  - Nodejs 6.9.X or 6.10.X installed or installable
- Network constraints:
  - Port 80 (http) and 443 (https) must be accessible from the outside. A redirect is done from the port 80 to the port 443 (in the apache configuration).
  - Port 22 (ssh) must be accessible from the outside
  - If the mongodb database is on a third-party server, we recommend a dedicated and reserved network interface for this purpose, ssh access and monitoring.

2.2 Multiple instance installation on a single server

It is possible to have multiple instance running on a single server. Typically an instance on https://drive.thecompany.com and other ones such as https://transfer.thecompany.com and https://transfer.company-drive.com. In this case, separated database will be used in the mongodb server as well as separated nodejs servers for each subdomain.

3 Installation on debian 8

These scripts are for the cryptndrive installation on debian 8.

3.1 Prerequisites

You must be root on the server to set up the application. If you are not, please, ask the person that is root to make the installation instead of you. In case you

are not root, we will not provide to you any installation support.

On the server you need to have the following applications:

- Apache2
- MongoDB
- certbot for ssl certificates
- pip for python
- termcolor module for python
- apt module for python
- curl

```
apt install apache2 mongodb-server certbot \
 python-pip python-termcolor python-apt curl
```

- Node.js 6.X or later and pm2 2.10.X or later

  ```
  curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash - \
  && apt install nodejs
  npm i -g pm2
  ```

- ovh module for python (optional)

  ```
  pip install ovh
  ```

You also have to create a lynvictus UNIX account

```
useradd -s /bin/bash -m -d /home/lynvictus -c "lynvictus user" -G sudo lynvictus
passwd lynvictus
```

3.2 get cryptndrive

Lybero.net has its own server to provide installation archives. We need to have your username and password. Please type the following command and send us (contact@lybero.net) the result (remembering your password).

```
htpasswd -n <votre login>
```

A password will be asked, you will then send us (contact@lybero.net) the password hash that we will enter our system.

You will then be able to recover the following file:

```
https://npmjs.lybero.net/repository/lynvictus-latest.tgz
```

This tar contains an already prepared version of the application with the modules. The full sequence to download is the following :

```
cd /root
mkdir Source
wget --no-check-certificate --user yourusername --password yourpassword \
https://npmjs.lybero.net/repository/lynvictus-latest.tgz
cd Source
tar -xvf ../lynvictus-latest.tgz
```

You then have in /root/Source a lynvictus-2.1.3 directory (from the cryptndrive version number). We will now copy (or move) this directory where it is needed by simply renaming it by the version number, then create an instance.

The content of lynvictus-2.1.3 is the following:

```
lynvictus-2.1.3
                |-- client
                |-- Documentation
                |-- Libs
                |-- mail
                |-- node_modules
                |-- notifServer.js
                |-- scripts
                        |-- Configure_Instance.py
                        |-- crypt.js
                        |-- install.py
                        |-- migrate.js
                        |-- OvhDnsEntry.py
                        |-- prerequisites.py
                        |-- template
                                |-- apache_conf
                                        |-- instance.conf
                                        |-- letsencrypt.conf
                                |-- instance_conf
                                        |-- default_yml.template
                |-- server.js
                |-- SharedLibs
```

To install the app, please use only python scripts in scripts/. All of these scripts need to be executed as root user and you can use –dry-run to see what the script will do without execution

3.3 Using NAT network

If your machine or virtual machine is beyond a NAT, then you need to do a special network configuration. In fact, the IP address corresponding to the domain of the drive does not correspond to the IP address provided by the DNS for this domain. In this case, we must add in the file /etc/hosts the line:

```
192.168.0.1 drive.lybero.net
```

Obviously replace 192.168.0.1 with the machine's local IP and drive.lybero.net with the domain name used for your drive.

3.4 Check the system before the installation

Use prerequisite.py to check your system

```
./prerequisites.py
```

It will check if all prerequisites are respected

- Check first if there is a lynvictus unix account, if no, the script create it
- Then check if apache2, mongodb-server and certbot are installed, if not, install with apt
- Finally, check if nodejs and pm2 are installed with the good version

At the end, you must have a Lynvictus user, apache2, mongodb-server and cerbot, nodejs v.6.X and pm2 v.2.10.X installed

3.5 Installation of the version

Use install.py scripts to install a version of the lynvictus app

```
./install.py --destination=/var/www/html/Lynvictus
```

- destination : path to destination folder

It will create a version folder (named like the version number of the app) and copy all the folder's files into the version folder and change owner to www-data:

```
/var/www/html/Lynvictus/2.1.3
                |-- client
                |-- Documentation
                |-- Libs
                |-- mail
                |-- node_modules
                |-- notifServer.js
                |-- scripts
                        |-- Configure_Instance.py
                        |-- crypt.js
                        |-- install.py
                        |-- migrate.js
                        |-- OvhDnsEntry.py
                        |-- prerequisites.py
                        |-- template
                                |-- apache_conf
                                        |-- instance.conf
                                        |-- letsencrypt.conf
                                |-- instance_conf
                                        |-- default_yml.template
                |-- server.js
                |-- SharedLibs

ls -al /destination/folder/version

drwxr-xr-x    9 www-data www-data  4096 mai   24 12:04 .
drwxr-xr-x   34 root     root      4096 mai   24 14:12 ..
drwxr-xr-x    9 www-data www-data  4096 mai   24 12:04 client
drwxr-xr-x    4 www-data www-data  4096 mai   24 12:04 Documentation
drwxr-xr-x    2 www-data www-data  4096 mai   24 12:04 Libs
drwxr-xr-x    2 www-data www-data  4096 mai   24 12:04 mail
drwxr-xr-x 1015 www-data www-data 36864 mai   24 12:04 node_modules
-rwxr-xr-x    1 www-data www-data  2633 mai   24 12:04 notifServer.js
drwxr-xr-x    3 www-data www-data  4096 mai   24 12:04 scripts
-rwxr-xr-x    1 www-data www-data 14903 mai   24 12:04 server.js
drwxr-xr-x    3 www-data www-data  4096 mai   24 12:04 SharedLibs
```

3.6 Instance creation

/!\ If the script failed during his execution, delete the destination folder before retrying ! /!\

If it's the first time using certbot, your email will be asked during the execution of the script (make sure to use -v option) However, make sure that your domain name is existing and valid, if not, certbot will not be able to get certificates

Use configureInstance.py to install a new instance or update an existing one, please, use the same destination as used in the install.py scripts

```
./configureInstance.py
        --name= <name>
        --domain-name
        [--port= <port number>]
        [--version=<version number to use>]
        [--destination=<destination where software is installed> ]
        [ --url= <instance url> ]
        [ --mongo_db= <mongodb database> or --mongo_url= <mongodb url> ]
```

- name : instance's name (without space)
- port : port to use with the instance (do not use 0 or greater than 65535) by default it will take the first available above 3000
- version : lynvictus version number to use (like 2.1.3) by default read the version from the PATH
- destination : path to destination folder (like /var/www) by default /var/www/html/Lynvictus
- domain-name : domain name ex: lybero.net
- url : frontal instance's url (you can use http or https but without space) (not mandatory, default value : https://'name'.lybero.net)
- mongo_db : mongoDB database name (without space) (not mandatory, default value : name of the instance)
- mongo_url : mongoDB database url (like mongodb://ipOrHostname:port/databaseName) (not mandatory, default value : mongodb://localhost:27017/'mongo_db')

It will creates a new instance folder into the destination folder or update it if exists and create a backup.

In this directory, there are symbolic links to the Lynvictus application folder (Libs, SharedLibs, client, mail, node_modules, notifServer.js, scripts and server.js). You can choose the version to use in using the –version option. The list of available version is the list of directories in /var/www/html/Lynvictus that are just numbers and points (for example 2.1.3). A config folder holding the single configuration file is also created in the directory.

In the config folder, there is the configuration of the instance (default.yml) generate by the script using the LynvictusInstanceConfig.template file where it need the name of the instance, the url, the port and the mongodb_url.

Also, the script produces a first apache configuration to get ssl certificate from let'sencrypt for the instance's virtualhost (so only listennig on the 80 port) (using LetsencryptinstanceConfig.template). The configuration will be activated and the apache server will be restart. After, a certbot commands (webroot) is used to get ssl certificates

Then, the script generates a second apache config file (erasing the first one) using ssl and a redirection of the 80 port to the 443, also, this configuration contains a reverse proxy to the instance (using –port) (using ApacheInstance-Config.template).

Finally, the script start (or restart) the server (server.js) and the notification server (notifServer.js) using pm2 with the lynvictus user. There are respectively named lynvictus-'name' and LynNotif-'name'

/!\ if you get an error a this point saying that server.js and notifServer.js have not been found, please delete the destination folder and restart the script /!\

```
/destination/folder/name
                |-- client -> ../../version/client
                |-- configs
                        |-- default.yml
                |-- Documentation -> ../../version/client
                |-- Libs -> ../../version/client
                |-- mail -> ../../version/mail
                |-- node_modules -> ../../version/node_modules
                |-- notifServer.js -> ../../version/notifServer.js
                |-- scripts -> ../../version/scripts
                |-- server.js -> ../../version/server.js
                |-- SharedLibs -> ../../version/SharedLibs
```

```
cat /destination.folder/name/configs/default.yml
```

```
  # The instance Name
  # -----------------
  instance: name

  # URL in case of
  # ---------------
  url : https://name.lybero.net

  # The server port(s)
  # ------------------
  daemons:
    standard:
      port: 3000
  # ssl:
  #   port: 3000
  #   key : ssl/server.key
  #   crt : ssl/server.crt
```


```
  initial:
    # --- Want to create initial users (alice / bob / charlie)
    # --- and theres filesets
    # -----------------
    withInitialDB: true
    # --- Want to fill the db with a lot of users
    # and filesets (the number)
    withInitialDBNum: 0

  # Database information
  # --------------------
  database:
    url: 'mongodb://localhost:27017/name'

cat /etc/apache2/site-available/name.lybero.net.conf

  <VirtualHost name.lybero.net:80>
        ServerName name.lybero.net
        ServerAdmin admin@mail.com
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        RewriteEngine on
        RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,QSA,R=permanent]
  </VirtualHost>

  <IfModule mod_ssl.c>
  <VirtualHost name.lybero.net:443>
        ServerName name.lybero.net
        ServerAdmin admin@mail.com
        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLCertificateFile /etc/letsencrypt/live/name.lybero.net/fullchain.pem
        SSLCertificateKeyFile /etc/letsencrypt/live/name.lybero.net/privkey.pem
        Include /etc/letsencrypt/options-ssl-apache.conf

        # ----------------------------------------------------------
        # --- redirection for lynvictus demo -----------------------
        # ----------------------------------------------------------
        ProxyPreserveHost On
        ProxyRequests off
```


```
        RewriteEngine On
        RewriteCond %{HTTP:Upgrade} =websocket [NC]
        RewriteRule /(.*)           ws://localhost:3000/$1 [P,L]

        ProxyPass / http://localhost:3000/ retry=1 acquire=port timeout=600 \
        Keepalive=On
        ProxyPassReverse / http://localhost:port/
        ProxyPassReverseCookiePath http://localhost:port https://name.lybero.net
        # ------------------------------------------------------------
  </VirtualHost>
  </IfModule>

sudo su - lynvictus -c "pm2 list"
(you will obtain something like this)

  /------------------------------------------------------------------
  ----------------------------------------\
  | App name         | id | mode | pid    | status | restart | uptime |
  cpu | mem       | user      | watching |
  |------------------------------------------------------------------
  ----------------------------------------|
  | Lynvictus-name   | 0  | fork | 1451   | online | 3       | 4h     |
  0%  | 66.3 MB   | lynvictus | disabled |
  \------------------------------------------------------------------
  ----------------------------------------/
  Use `pm2 show <id|name>` to get more details about an app
```

3.7 OVH DNS

OvhDnsEntry.py can creates, remove or verify a DNS OVH entry using their api

So, to execute, you need to install the latest release of Python wrapper: $ pip install ovh

As it's use the ovh API, you need to create a token for use it. see https://api.ovh.com/createToken/index.cgi

```
    OvhDnsEntry.py  --endpoint <endpoint>
            --application_key <application_key>
            --application_secret <application_secret>
            --consumer_key <consumer_key>
            -n <name> [--add | --delete | --verify]
            --target-cname or --target-ip
```


- endpoint: OVH endpoint
- application_key : OVH application_key
- application_secret : OVH application_secret
- consumer_key : OVH consumer_key
- name : entry's name
- target-cname : cname entry (CNAME)
  or
- target-ip : ip entry (A)

4 Installation on CentOS 7

4.1 Base

We start from a virgin Centos 7.4 virtual machine. I first install some useful tools and configure ssh to be able to connect by key. If you're not comfortable with linux, mc (midnight-commander) can really help you. Choose the editor you prefer.

```
yum install vim
yum install mc
vim ~/.ssh/authorized_keys
```

Do not forget to configure the ssh connection. I display the ~ / .ssh / id_rsa.pub file on my machine and copy the contents to /root/.ssh/authorized_keys.

4.2 /etc/hostname and /etc/hosts configuration

First the hostname file. Here we consider that the machine is named cryptndrive.fr.

```
cryptndrive.fr
```

now /etc/hosts file

```
127.0.0.1   localhost
51.255.165.34   cryptndrive cryptndrive.fr  vps523489.ovh.net   vps523489
```


4.3 Install apache / node / mongo / git

In centOS 7.4, the default nodejs package works directly.

```
yum install epel-release nodejs mongodb-server
#(assurez vous que node -v retourne 6.X)
service mongod start
service mongod status
service mongod stop
service mongod start
systemctl enable mongod
npm install pm2 -g
pm2 startup systemd
yum install mod_ssl python-certbot-apache
yum install httpd
systemctl enable httpd
yum install git
```

4.4 firewall

```
firewall-cmd --zone=public --list-all
firewall-cmd --zone=public --add-service=http --permanent
firewall-cmd --zone=public --add-service=https --permanent
firewall-cmd --zone=public --add-service=http
firewall-cmd --zone=public --add-service=https
firewall-cmd --reload
firewall-cmd --zone=public --list-all
```

4.5 get cryptndrive

We have our own server to access tar.gz. We need to have your username and password. Please type the following command and send us the result (remembering your password).

```
htpasswd -n <votre login>
```

A password will be asked, you will then send us the password hash that we will enter our system.

You will then be able to recover the following file:

```
https://npmjs.lybero.net/repository/lynvictus-latest.tgz
```


This tar contains an already mined and prepared version of the application with the modules.

```
cd /root
mkdir Source
cd Source
tar -xvf ../lynvictus-latest.tgz
```

You then have in / root / Source a lynvictus-2.1.3 directory (from the cryptndrive version number). We will now copy (or move) this directory where it is needed by simply renaming it by the version number, then create an instance.

4.6 Installing the reference instance

We want to be able to install different versions of the CryptnDrive application. For this, for each version, we install the different files and modules in a directory. Then we make symbolic links between the subdirectories of a version and an instance. This makes it possible to have a different configuration file for the instance in a basic way.

```
/var/www/html/Lynvictus
       |-- 2.1.3
              |-- client
              |-- Libs
              |-- .....
              |-- configs
       |-- instance
              |-- client (lien symbolique vers ../../2.1.3/client)
              |-- Libs  (lien symbolique vers ../../2.1.3/Libs)
              |-- ....
              |-- configs (contient un vrai fichier)
```

Copy of the version.

```
mkdir -p /var/www/html/Lynvictus
mkdir -p /etc/httpd/conf.d
cp -R lynvictus-2.1.3 /var/www/html/Lynvictus/2.1.3
```

We will create the “drive” instance on the server.

```
mkdir "/var/www/html/Lynvictus/drive"

SUBDIR_LIST=(client Libs mail node_modules notifServer.js scripts \
```


```
server.js SharedLibs ssl)

for SUBDIR in ${SUBDIR_LIST[*]}; do
    ln -s /var/www/html/Lynvictus/2.1.3/${SUBDIR} \
  /var/www/html/Lynvictus/drive/${SUBDIR}"
done
chown -R apache:apache /var/www/html/Lynvictus
```

4.7 Drive instance configuration

You have to modify the file '/var/www/hml/Lynvictus/drive/configs/default.yml'. It must indicate: the name of the instance (drive), the port on which the node server will run, the port on which the Mongo server runs and the name of the database. If you want to authenticate via Google, you must enter the clientID, clientSecret information provided by Google through the third-party application configuration interface

So you need to edit the following lines: * url : l'url du frontal * port : port sur lequel va écouter le drive * database: url: url de la base mongoDB

```
#The instance Name
#-----------------
instance: drive
# URL for creating mails url
# ---------------
url : https://drive.lybero.net
# The server port(s)
#-----------------
daemons:
  standard:
    port: 3000
initial:
  #--- Want to create initial users (alice / bob / charlie)
  #--- and their filesets
  #-----------------
  withInitialDB: true
  #--- Number of users and filesets to fill the db with initially
  #--- for test purpose
  withInitialDBNum: 0

#Database information
#--------------------
database:
  url: 'mongodb://localhost:27017/drive'
#auths:
```

```
#    local: {}
#    google:
#        clientID:  a_configurer_par_vos_soins_pour_une_authentification_avec_google
#        clientSecret: a_configurer_par_vos_soins_pour_une_authentification_avec_google
#        callbackURL: 'https://drive.cryptndrive.fr/oauthcallback'
```

4.8 Creating the lynvictus user and launching pm2

```
useradd -s /bin/bash -m -d /home/lynvictus -c "lynvictus user" lynvictus

sudo -u lynvictus --cwd /var/www/html/Lynvictus/drive \
 env PATH=$PATH:/usr/bin:/usr/local/bin \
 /usr/lib/node_modules/pm2/bin/pm2 start \
 /var/www/html/Lynvictus/drive/server.js \
 -n drive-server  -u lynvictus --hp /home/lynvictus
```

4.9 Apache configuration for drive

Here we consider that there is only one node.js server running on port 3000.
File /etc/httpd/conf.d/drive.conf

```
<VirtualHost drive.lybero.net:80>
        ServerName drive.lybero.net
        ServerAdmin your.mail@your.domain
        DocumentRoot /var/www/html

        ErrorLog logs/error.log
        CustomLog logs/access.log combined

        RewriteEngine on
        RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,QSA,R=permanent]
</VirtualHost>

<IfModule mod_ssl.c>
Listen 443 https
<VirtualHost drive.lybero.net:443>
        ServerName drive.lybero.net
        ServerAdmin your.mail@your.domain
        DocumentRoot /var/www/html

        ErrorLog logs/error.log
        CustomLog logs/access.log combined
```


```
        SSLEngine on
        SSLProtocol             all -SSLv2 -SSLv3
        SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:\
        ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:\
        DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:\
        ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:\
        ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:\
        ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:\
        DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:\
        DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA:\
        AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:\
        AES256-SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:\
        !PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA
        SSLHonorCipherOrder     on
        SSLCompression          off
        SSLOptions +StrictRequire
        # Add vhost name to log entries:
        LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" \
        vhost_combined
        LogFormat "%v %h %l %u %t \"%r\" %>s %b" vhost_common
        SSLCertificateFile /etc/letsencrypt/live/drive.lybero.net/fullchain.pem
        SSLCertificateKeyFile /etc/letsencrypt/live/drive.lybero.net/privkey.pem
        Include /etc/letsencrypt/options-ssl-apache.conf

        # -----------------------------------------------
        # --- redirection for lynvictus demo ----------
        # -----------------------------------------------
        ProxyPreserveHost On
        ProxyRequests off

        #RewriteEngine On
        #RewriteCond %{HTTP:Upgrade} =websocket [NC]
        #RewriteRule /(.*)           ws://localhost:3000/$1 [P,L]

        ProxyPass /ws  ws://localhost:3000/ws
        ProxyPassReverse /ws  ws://localhost:3000/ws

        ProxyPass / http://localhost:3000/ retry=1 acquire=3000 timeout=600 \
        Keepalive=On
        ProxyPassReverse / http://localhost:3000/
        ProxyPassReverseCookiePath http://localhost:3000 https://drive.lybero.net
        # -----------------------------------------------------------
</VirtualHost>
</IfModule>
```


If one also has a base domain with static files, one must not forget to create the corresponding apache configuration:

```
############## /etc/httpd/conf.d/cryptndrive.conf
<VirtualHost www.cryptndrive.fr:80>
    DocumentRoot /var/www/html
    ServerName www.cryptndrive.fr
</VirtualHost>
####################################
```

Restarting apache

```
service httpd restart
```


4.10 Nginx reverse proxy configuration

```
server {
  listen 443 ssl;
  server_name localhost
  ssl on;
  ssl_certificate /etc/apache2/certificates/localhost.crt;
  ssl_certificate_key /etc/apache2/certificates/localhost.key;
  location /ws {
          proxy_pass http://localhost:3000/ws;
          proxy_http_version 1.1;
          proxy_set_header Upgrade $http_upgrade;
          proxy_set_header Connection "Upgrade";
  }
  location / {
          proxy_pass http://localhost:3000;
  }
}
```

4.11 Network configuration

4.11.1 SELinux

In order for apache to connect to nodejs, it must be allowed to make a network connection. connection. The command is as follows:

```
/usr/sbin/setsebool -P httpd_can_network_connect 1
```

4.11.2 Special case of NAT

If your machine where your virtual machine is behind a NAT, then you have to do a special network configuration. In fact, the IP address corresponding to the domain of the drive does not correspond to the IP address provided by the DNS for this domain. In this case, we must add in the file / etc / hosts the line:

```
192.168.0.1 drive.lybero.net
```

Obviously replace 192.168.0.1 with the machine's local IP and drive.lybero.net with the domain name used for your drive.

4.11.3 mail configuration

The configuration of the mail can be done in 2 different ways:

- using only the parameters in the CryptNDrive application. By default, outgoing mail is considered to be sent on port 25 of localhost. However, the configuration settings for sending mail on the CryptNDrive server are editable. You have to go to the main menu, then Administration, then Configuration.
- or by directly configuring a local mail server as a relay to a third-party SMTP service.

We will examine this last configuration in the case of OVH. All you need to do is set up a default mail user for your domain in OVH, and authenticate with that user.

4.11.3.1 Configuring postfix to use OVH relay You need to have several postfix modules for this to work:

```
yum install libsasl2-modules
yum install cyrus-sasl cyrus-sasl-lib cyrus-sasl-plain
yum install cyrus-sasl-gssapi
yum install cyrus-sasl-ntlm
```

We will create a file / etc / postfix / sasl-passwords with the email address and the password of the person whose email was created in the OVH interface:

```
[pro1.mail.ovh.net]:587 cryptndrive@cryptndrive.fr:mettre_le_mot_de_passe_ici
```

You have to transform the file into a db:

```
postmap hash:sasl-passwords
```

Now you have to create the ssl key files for the connection:

```
cd /etc/ssl/certs
make genkey
```

In /etc/postfix/main.cf, the parameters to be modified are:

```
myhostname = cryptndrive.fr
inet_interfaces = $myhostname, localhost
mydestination =
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/localhost.crt
smtpd_tls_key_file=/etc/pki/tls/private/localhost.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtp_tls_security_level = may


relayhost = [pro1.mail.ovh.net]:587
smtp_sasl_auth_enable=yes
smtp_sasl_mechanism_filter = !gssapi, plain, login
smtp_sasl_password_maps=hash:/etc/postfix/sasl-passwords
smtp_sasl_security_options= noanonymous
```

You need a program to try:

```
yum install mutt
```

It only remains to try to send an email with mutt.

4.12 Get ssl certificats with Certbot

The apache configuration uses ssl certificates generated by Lets Encrypt. However, if you want to use your own certificates, simply modify the following lines:

```
SSLCertificateFile /chemin/vers/votre/certificat
SSLCertificateKeyFile /chemin/vers/votre/clé/privée
```

You must have created an account on Lets Encrypt and installed the package certbot. The documentation Let's Encrypt is very well made for these explanations. Being root, you have to run:

```
certbot certonly --apache --non-interactive \
--email your.email@your.domain --agree-tos \
--renew-by-default -d drive.cryptndrive.fr
```


Finally, you should not have forgotten the corresponding entries for DNS, in the DNS of your choice. At OVH, with a little automation, this can be done with a single post. Here is a small node program:

```
var ovh = require('ovh')({
  endpoint: 'ovh-eu',
  appKey: 'votre-app-key-voir-la-doc-ovh',
  appSecret: 'votre-app-scret',
  consumerKey: 'votre-consumer-key'
});

ovh.request('POST', '/domain/zone/cryptndrive.fr/record', {
        fieldType: 'A',
    subDomain: 'drive',
    target: '18.10.34.34',
        ttl: 0
}, function (err, result) {
  console.log("result : " + result);
  console.log(err || result);
});

# Do not forget to create the qdn in the DNS file on OVH
```

4.13 Debug

First install the necessary packages using the following command:

```
yum install tcpdump telnet bind-utils net-tools
```

To check that a port of the machine is open, just type:

```
netstat -an | grep '2000.*LISTEN'
```

(remplace 2000 for the choosen port)

If you get a return, it's because the port is open.

5 Configuration

5.1 Startup configuration

At the first start of the application you can login with this accounts:

User	Password	Description
`root`	`root`	Master account, with all rights (right of *manage* on the object *application*). You must change his password ! This account can be deleted after giving rights to somebody else.
`alice`	`alice`	A normal user. usable for testing.
`bob`	`bob`	A normal user. usable for testing.
`charlie`	`charlie`	A normal user. usable for testing.

All thoses users can be deleted.

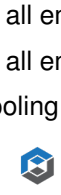
5.2 Application configuration

All the application can be access by the *Menu* (left up button), then *Administration*, then *Configuration*.

This menu and the ability to modify the configuration is attach to the right of *manage* on the object *application*. The first configured user with this right is `root`.

5.2.1 General configuration

Field	Description	Comment
`Instance`	Instance name	This field cannot be changed. It is just for information
`The application Name`	Application's name	You can change here the application name in the mair
`The compagny Name`	Company's name	In the *top bar* it is written `by the compagny Name`.
`A subtitle`	Subtitle	Not used.
`Tel for contact`	A phone number	In the signature of all emails, this is the contact phone
`Email contact`	An email	In the signature of all emails, this is the email for conta
`Pooling interval`	Delay	This is the cron pooling delay for emails and purge of I



Field	Description	Comment
`Logo`	An image	You can upload a image (any format) for the logo. The
`Mandatory quorum group`	Quorum group's name	This is the name of the quorum group for recovering e

5.2.2 client configuration

client configuration is about the theme by default and the available themes for users. You can modify existing themes, delete them, or create a new one by clicking on `+`.

For each theme, you can modify `spacing` (for placement) and `palette` (for colors).

All the application colors depends on themes except the slapshscreen (the loading page).

5.2.3 users configuration

users configuration describe the application behaviour about users created by an administrator (or auth by oauth2) and users self registered. You can add them to differents groups to manage differents rights.

- For `signin` users :

`signin` users are self registered users.

Field	Description	Comment
`enabled`	if users can sign in the application	You can decide if users can create accounts for
`Groups by default`	A list of groups names	Users will be added into thoses groups. You can
`Expiration delay`	The lifetime of thoses account	you can specify a lifetime of the account. 0 mean

- For `regular` users :

`regular` users are users created by administrators or oauth2 users (like google auth).

Field	Description	Comment
`enabled`	if users can sign in the application	You can decide if users can create accounts for

Field	Description	Comment
`Groups by default`	A list of groups names	Users will be added into thoses groups. You can
`Expiration delay`	The lifetime of thoses account	you can specify a lifetime of the account. `0` mean

5.2.4 notification configuration

This is the notification (chat between users and informations from server) configuration.

Field	Description	Comment
`Expiration`	a delay	How much time you will keep any notifications. `0` mean no expiration.

5.2.5 server configuration

This is the server configuration.

Field	Description	Comment
`websockets`	true or false	Say if you want websockets for speedup transfert. In some installations, websc

5.2.6 mail configuration

This is the email server configuration, for sending emails (SMTP).

- `connexion` configure the way to contact the email server (SMTP protocol)

Field	Description	Comment
`Host`	hostname or IP address	The SMTP server accepting connexion from the a
`Port`	port number	25, 587 or 465 in general.
`Secure`	if you are using TLS	if true the connection will use TLS when connecti
`ignoreTLS`	refuse server TLS	if this is true and secure is false then TLS is not u
`auth User`	the username	In case of a authenticated SMTP communication,
`auth Password`	the password	In case of a authenticated SMTP communication,

Field	Description	Comment
`tls rejectUnauthorized`	reject unknown certificates	If not false the server will reject any connection w

Field	Description	Comment
`Subject prefix`	Prefix in any subjects	You can add this string in the subject field (can be used for filteri
`from`	from email field	emil from
`replyTo`	replyTo email field	replyTo email field
`Digest delay`	delay	delay for digest email

5.2.7 auths configuration

The application can use external authentication (**oauth2**). The google authentication can be set up in this place

- `google` authentication

first you must set up for your server a Identity provider. Go in google developement interface :

https://console.developers.google.com/apis/

and fill a new identity with the following elements

1. Create an `Id client Oauth`
2. Select `Application web`
3. fill the following fields :

Field	Description
`Name`	A name for this Oauth2 provider
`javascript origin`	Your Server (example *https://drive.lybero.net*)
`callback redirect`	the callback URL , your server/oauthcallback (example *https://drive.lybero.net/oauthc*

And Google provides you two fields `clientId` and a `client Secret`:

Then you can enter in the configuration :

Field	Description	Comment
`The clientID provided by Google`	clientId	provided by Google
`The client secret provided by Google`	client Secret	provided by Google

5.2.8 errors configuration

You have the hability to send reports on application bug directly... where you want. Actually, only a slack is available.

if you enable this feature, if a javascript bug occured, on a client or on the serveur, a report is pushed into a slack channel.

Field	Description	Comment
`enable`	Toggle to yes for enabling it	
`slack webhookURL`	the URL provided by your slack configuration	it define the channel to post on your tea

For sending your bugs directly to lybero.net, you can user this skack webhookURL :

```
https://hooks.slack.com/services/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
```

5.3 Error reporting

Error reporting is usefull to find bugs on exotic configurations (browsers, architectures), wich cannot be tested in lab before publishing.

You can setup error reporting to lybero.net directly and permanently, or to your own server, or just during a debbuging session. as you want.

No sensitive datas are in the report (no password, passphrases, keys, files, etc...). only informations on the browser, the architecture, and the code file / line in trouble.

The error report contain the following information and nothing more (this is an example) :

```
<hostname>/<instanceName>
blob is undefined{
 "navigator": {
   "appCodeName": "Mozilla",
```



```
   "appName": "Netscape",
   "appVersion": "5.0 (X11)",
   "cookieEnabled": true,
   "language": "fr",
   "oscpu": "Linux x86_64",
   "product": "Gecko",
   "userAgent": "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.
 },
 "builder": "wallrich",
 "commit": "e1f0fc3",
 "builderhost": "kryha",
 "server": {
   "server": "kryha",
   "instance": "developement",
   "builder": "wallrich",
   "commit": "e1f0fc3",
   "builderhost": "kryha",
   "login": "ANONYMOUS",
   "userId": "0"
 }
}
auto_bom@http://localhost:3000/LybStores.js:33827:1
FileSaver@http://localhost:3000/LybStores.js:33834:12
saveAs@http://localhost:3000/LybStores.js:33899:11
downloadFile/<@http://localhost:3000/LybStores.js:12649:11
run@http://localhost:3000/polyfill.js:4257:22
notify/<@http://localhost:3000/polyfill.js:4270:30
flush@http://localhost:3000/polyfill.js:1386:9
```