

YobiDrive

Installation manual

Version 1.0

Last updated: 26 February 2013

Table of Contents

1. Introduction.....	3
2. Prerequisites	5
2.1. Java virtual machine.....	5
2.1.1. Installation.....	5
2.1.2. Cryptography extensions.....	5
2.2. Tomcat server.....	6
2.2.1. Global configuration.....	6
2.2.2. AJP Connector.....	6
2.2.3. Clean the ROOT application	7
2.2.4. Enable custom HTTP return codes.....	7
2.3. Apache web-server.....	7
2.3.1. Global configuration.....	7
2.3.2. SSL installation.....	7
2.3.3. Mod_jk installation.....	8
2.3.4. Virtual host yobidrive.mycompany.com.....	9
3. YobiDrive core server installation.....	11
3.1. Yobifolders directory.....	11
3.2. Otherlibs directory and server configuration.....	12
3.3. WAR application deployment.....	12
4. YobiDrive FX web-sharing installation.....	14
4.1. Package installation.....	14
4.2. VirtualHost yobidrivefx.mycompany.com.....	15
4.3. Startup script.....	17
5. What to do next ?.....	19
5.1. Increase the registered number limitation.....	19
5.2. Customize your YobiDrive FX application.....	19
5.3. Build your own application on top of the YobiDrive server	20
6. Troubleshooting.....	20

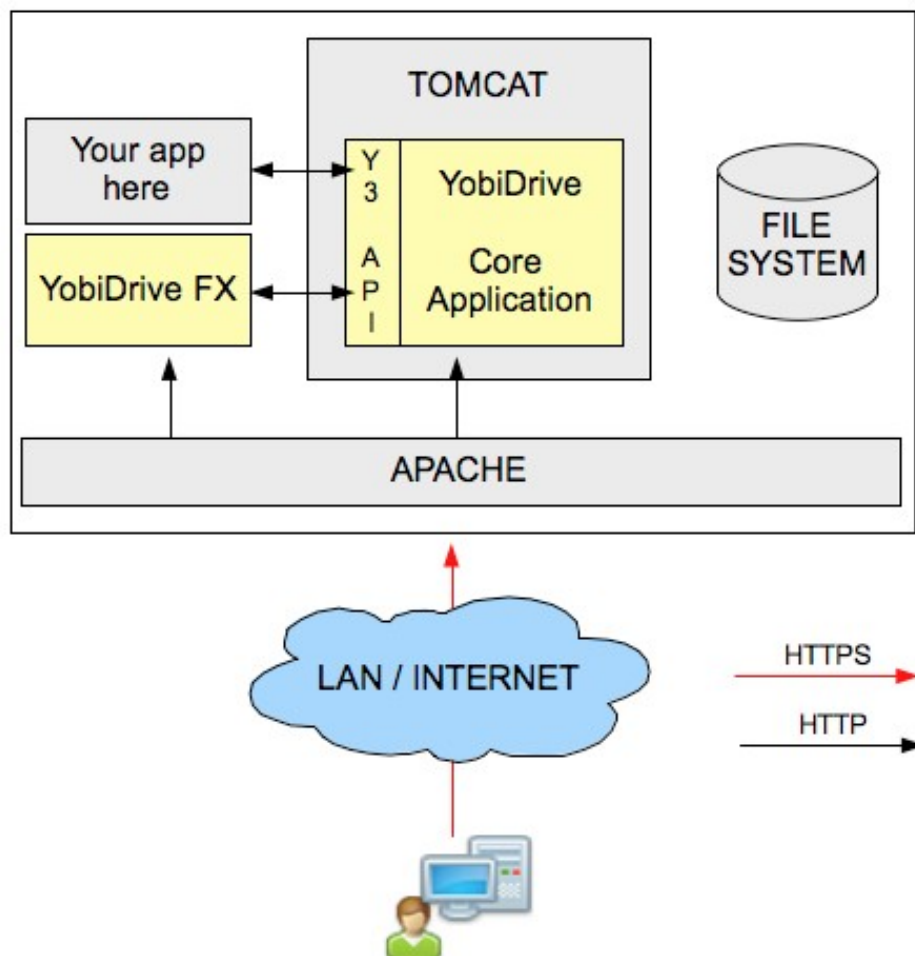
1. Introduction

This document explains all the steps required to set-up from ground-up a single node YobiDrive platform including:

- The YobiDrive core server application
 - Single node Key-Value storage
 - Y3 high level API to build your own customized services
- The YobiDrive FX professional file sharing application
 - Easy to use responsive HTML5 interface
 - Smart access rights management
 - Publication and traceability
 - Fully customizable (Apache 2.0 license)

The target system in this document is a freshly installed Debian 6.0 machine with only pre-requisite being a root ssh access to the machine. For the sake of simplicity both applications will be installed on the same machine but it's not a requirement.

The following drawing summarizes the different components of the system.

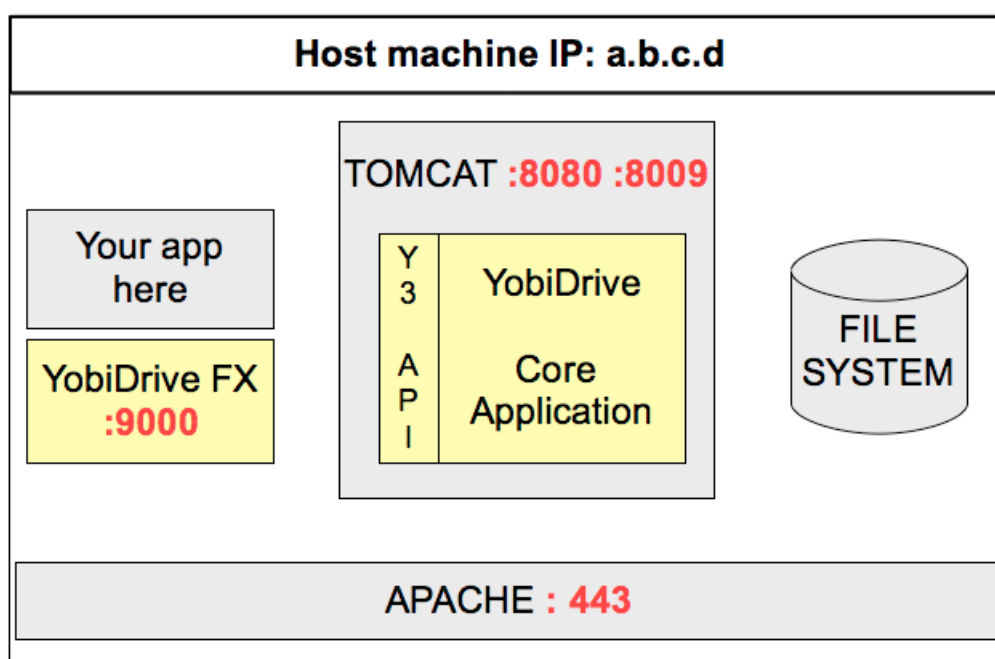




The Apache server will be responsible for handling HTTPS requests and route them internally. We will define two virtual hosts: *yobidrive.mycompany.com* and *yobidrivefx.mycompany.com* on the Apache machine. These are only examples and you are of course encouraged to use another names. Don't forget to update your DNS server accordingly, or if you are testing on your local machine add the required line in your hosts file.

The Tomcat server will be listening on port 8080 and 8009 and will host the YobiDrive server application. **It's really important to note that the single node version of YobiDrive doesn't manage data replication and thus you need to handle the backup of the file system manually.** A common solution is to use a storage area network instead of the local drive or to use a RAID array of drives.

The YobiDrive FX is a standalone lightweight Java application powered by the Play! Framework. This application will communicate with the YobiDrive server through HTTP calls to the Y3 API.



The above drawing shows in details the the target configuration (numbers in red are the TCP ports on which applications are listening) :

- Apache will listen on port 443 on two virtual hosts:
 - *yobidrivefx.mycompany.com*, for all traffic related to the YobiDrive FX application.
 - *yobidrive.mycompany.com* for all traffic related to YobiDrive core API calls
- YobiDrive FX will be listening on port 9000 through an Apache reverse proxy. It will communicate with the YobiDrive core server via HTTP calls on port 8080.
- Tomcat will listen on two ports:
 - 8009 (AJP) for all YobiDrive API calls made via *yobidrive.mycompany.com*
 - 8080 (HTTP) for calls made internally by YobiDrive FX.
 - Those two ports should be closed at the firewall level and only accessible from within the machine.



2. Prerequisites

We will start by installing the common components that will be required by YobiDrive. Depending on the machine you are installing on you may skip some steps.

Pre-requisites are:

- A SUN Java Virtual Machine version 6 with optionally the cryptography extensions
- A Tomcat 6 server
- An Apache 2 web-server

2.1. Java virtual machine

YobiDrive requires a Java Virtual Machine (JVM) version 6 to run.

2.1.1. Installation

In order to install the JVM on our Debian machine the first step is to add the non-free repository:

```
> nano /etc/apt/sources.list
```

Ensure that non-free is present in the first line, if it is not add it then save the file. Please note that the repository URL may be different in your installation.

```
GNU nano 2.2.4 File: /etc/apt/sources.list

deb http://mirror.ovh.net/debian/ squeeze main non-free
deb-src http://mirror.ovh.net/debian/ squeeze main

deb http://security.debian.org/ squeeze/updates main
deb-src http://security.debian.org/ squeeze/updates main
```

Now install the SUN JVM version 6:

```
> apt-get update
> apt-cache search sun-java6
> apt-get install sun-java6-bin sun-java6-jre
```

2.1.2. Cryptography extensions

By default YobiDrive tries to encrypt all data store to disk using the AES-256 algorithm. Nevertheless this level of encryption requires the Java Cryptography Extensions (JCE) to be installed. Due to US export control restriction, the JCE is not installed by default. Those extensions are not mandatory and if they are not installed YobiDrive will automatically fall back to 128bits encryption and warn you with a message in the application log file.

```
STARTUP ; Java Cryptography extensions are not installed, AES key will
be set to 128 bits
```

If you want 256 bits encryption it's your responsibility to carefully read license file and ensure that you have the right to install it in your country.



To install the JCE, you need to grab the installation package from Oracle website at the following url: <http://www.oracle.com/technetwork/java/javase/downloads/jce-6-download-429243.html>

You then need to replace the file *local_policy.jar* and *US_export_policy.jar* that are in the directory `/usr/lib/jvm/java-6-sun-1.6.0.26/jre/lib/security` by the one inside the file you downloaded from Oracle.

2.2. Tomcat server

YobiDrive requires a valid Tomcat 6 installation. If you don't have Tomcat installed on your machine you can install it using the following command:

```
> apt-get install tomcat6 tomcat6-admin
```

2.2.1. Global configuration

For simplicity we create links to our Tomcat Installation inside the `/opt` directory.

```
> /etc/init.d/tomcat6 stop
> mkdir /opt/tomcat
> cd /opt/tomcat
> ln -s /etc/tomcat6/ conf
> ln -s /usr/share/tomcat6/bin/ bin
> ln -s /usr/share/tomcat6/lib/ lib
> ln -s /var/lib/tomcat6/webapps webapps
> ln -s /var/log/tomcat6/ logs
```

We then need to create a user to access the management console used to deploy applications. This is done by editing the file *tomcat-users.xml*.

```
> nano /opt/tomcat/conf/tomcat-users.xml
```

Modify the user named tomcat, add the following roles and select a password.

```
<tomcat-users>
  <user username="tomcat" password="yourpassword" roles="tomcat,manager,admin,manager-gui" />
</tomcat-users>
```

2.2.2. AJP Connector

For better performances Apache will communicate with Tomcat via an AJP connector. We need to enable this connector by editing the *server.xml* file.

```
> nano /opt/tomcat/conf/server.xml
```

The connector is already defined in the file and you simple need to uncomment it:

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```



If for some reason you don't want to use AJP you can skip this step. In this case the configuration of the virtual host described at chapter 2.3.4 will be different as you will have to configure a reverse proxy (not detailed in this document).

2.2.3. Clean the ROOT application

YobiDrive server application will be deployed as the ROOT application of the server. To ensure that there are no access rights conflicts with the default ROOT application that comes with Tomcat we manually remove the application.

```
> rm -rf /opt/tomcat/webapps/ROOT
```

2.2.4. Enable custom HTTP return codes

YobiDrive API uses custom HTTP return codes. Newer versions of Tomcat disable this functionality by default and thus the file *catalina.properties* must be modified.

```
> nano /opt/tomcat/conf/catalina.properties
```

Add the following line to the file:

```
org.apache.coyote.USE_CUSTOM_STATUS_MSG_IN_HEADER = true
```

2.3. Apache web-server

YobiDrive requires a valid Apache 2 installation. If you don't have Apache installed on your machine you can install it using the following command:

```
> apt-get install apache2
```

2.3.1. Global configuration

We need to set UTF-8 as the default charset.

```
> nano /etc/apache2/conf.d/charset
```

Uncomment the line with AddDefaultCharset:

```
GNU nano 2.2.4 File: /etc/apache2/conf.d/charset

# Read the documentation before enabling AddDefaultCharset.
# In general, it is only a good idea if you know that all your files
# have this encoding. It will override any encoding given in the files
# in meta http-equiv or xml encoding tags.

AddDefaultCharset UTF-8
```

2.3.2. SSL installation

For security purposes all incoming traffic should be protected via HTTPS. In this document we will be using a self-signed certificate but if you already own one from a certification authority you should use it instead.



OpenSSL is required to generate the certificate, you can install it with the following command:

```
> apt-get install openssl
```

Certificate will be generated in directory `/etc/apache2/ssl` using the command:

```
> mkdir /etc/apache2/ssl
> openssl req -new -x509 -days 365 -nodes -out
/etc/apache2/ssl/apache.pem -keyout /etc/apache2/ssl/apache.key
```

You will just need to fill in some basic details:

```
Country Name (2 letter code) [AU]:LU
State or Province Name (full name) [Some-State]:Luxembourg
Locality Name (eg, city) []:Foetz
Organization Name (eg, company) [Internet Widgits Pty Ltd]:EZC Group
S.A.
Organizational Unit Name (eg, section) []:Labs
```

Then we enable the SSL module

```
> a2enmod ssl
```

Finally we modify the apache configuration to listen on port 443.

```
> nano /etc/apache2/ports.conf
```

Ensure the directives “NameVirtualHost *:443” and “Listen 443” are present:

```
NameVirtualHost *:80
Listen 80

<IfModule mod_ssl.c>
    # If you add NameVirtualHost *:443 here, you will also have to change
    # the VirtualHost statement in /etc/apache2/sites-available/default-ssl
    # to <VirtualHost *:443>
    # Server Name Indication for SSL named virtual hosts is currently not
    # supported by MSIE on Windows XP.
    NameVirtualHost *:443
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

If you start the server now, Apache will complain that there are no virtual host on port 443. It's normal as we will add it in the next steps.

2.3.3. Mod_jk installation

Mod_jk will be used to communicate with the Tomcat server. If you chose to skip the AJP connector installation in chapter 2.2.2, then you can skip this step too.

Install and activate mod_jk with the following commands:



```
> apt-get install libapache2-mod-jk
> a2enmod jk
```

First thing is to generate a `worker.properties` file for `mod_jk`. This file will be used to reference the available Tomcat servers.

```
> nano /etc/apache2/workers.properties
```

Paste the following content inside this file:

```
worker.list=proxy1
worker.proxy1.type=ajp13
worker.proxy1.host=127.0.0.1
```

Next thing is to give the path to this file to `mod_jk` by editing the `jk.conf` file:

```
> nano /etc/apache2/mods-available/jk.conf
```

Paste the following content inside this file:

```
JkWorkersFile          /etc/apache2/workers.properties
```

Finally we just have to disable/enable `mod_jk`.

```
> a2dismod jk
> a2enmod jk
```

2.3.4. Virtual host `yobidrive.mycompany.com`

This virtual host will redirect all requests to Tomcat through an AJP connector. If you chose to skip the AJP connector installation in chapter 2.2.2 then you have to set up a reverse proxy instead (not documented here).

```
> nano /etc/apache2/sites-available/yobidrive-ssl
```

Paste the following content inside the file, replacing `yobidrive.mycompany.com` with your desired hostname.

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
    ServerName yobidrive.mycompany.com
    AllowEncodedSlashes On

    JkMount /* proxy1
    JkOptions +ForwardURICompatUnparsed
    DocumentRoot /var/www-forbidden

    <Directory /var/www-forbidden/>
        Options Indexes FollowSymLinks MultiViews
```

```
        AllowOverride None
        Order allow,deny
        deny from all
    </Directory>

    SSLEngine on
    SSLCertificateFile    /etc/apache2/ssl/apache.pem
    SSLCertificateKeyFile /etc/apache2/ssl/apache.key

    <FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>

    BrowserMatch "MSIE [2-6]" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0
    BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>
</IfModule>
```

Enable the site and reload apache configuration:

```
> mkdir /var/www-forbidden
> a2ensite yobidrive-ssl
> /etc/init.d/apache2 reload
```

To test your installation you can point your browser to the URL:

<https://yobidrive.mycompany.com/manager/html>

If you are using a self-signed certificate your browser may warn you about it, simply trust the certificate to continue. You will then be able to sign to the Tomcat management console with the credentials you selected in chapter 2.2.1.



3. YobiDrive core server installation

The YobiDrive server package you downloaded is a zip file containing the following elements:

- *otherlibs* : directory with additional JAR files required by the YobiDrive application and a default configuration file.
- *yobifolders*: basic directory structure
- *ROOT.war*: server application to be deployed on Tomcat

Our target installation will be the following:

- */opt/otherlibs* will contains the additional JAR files
- */opt/yobifolders* will holds all the file manipulated by the YobiDrive application

First step is to upload the package to your server in the folder */opt*.

Then we simply unzip it.

```
> apt-get install unzip
> cd /opt
> unzip YobiDriveServerPackage.zip
```

The directories */opt/otherlibs* and */opt/yobifolders* will be created.

3.1. Yobifolders directory

The */opt/yobifolders* directory will contain all the data manipulated by the application and must be backed-up correctly. Indeed the single node installation of YobiDrive doesn't manage data replication and thus you need to handle the backup of the file system manually.

A common solution is to use a storage area network instead of the local drive and mount */opt/yobifolders* on this SAN.

You must ensure that the user running the Tomcat process (by default tomcat6) as write access on these folders.

```
> chown -R tomcat6:tomcat6 /opt/yobifolders/
```

If you didn't use the default */opt/yobifolders* directory then you need to modify a configuration file and ensure that the path to the base directory is correct. The command below opens the configuration in case of a standard installation, if you did not use */opt/yobifolders* as the base path you need to modify the command accordingly.

```
> nano /opt/yobifolders/_VOLDEMORT/config/server.properties
```

Look for the entries named *diskmap.log.path* and *diskmap.key.path* and ensure that they point to the good directories.

In our standard installation the values are:

```
diskmap.log.path=/opt/yobifolders/_VOLDEMORT/data
diskmap.key.path=/opt/yobifolders/_VOLDEMORT/index
```



3.2. Otherlibs directory and server configuration

The `/opt/otherlibs` directory contains all the additional libraries required by the application as well as the main server configuration file.

First step is to add this directory to the Tomcat server class-loader so that they can be found at runtime, this is done by editing the `catalina.properties` file.

```
> nano /opt/tomcat/conf/catalina.properties
```

Find the line with `common.loader` and add `"/opt/otherlibs,/opt/otherlibs/*.jar"` at the end of the line:

```
common.loader=${catalina.base}/lib,${catalina.base}/lib/*.jar,$
{catalina.home}/lib,$
{catalina.home}/lib/*.jar,/var/lib/tomcat6/common/classes,/var/lib/tomca
t6/common/*.jar,/opt/otherlibs,/opt/otherlibs/*.jar
```

Then we need to update the YobiDrive server configuration file.

```
> nano /opt/otherlibs/YobiFSProxy.cfg.xml
```

The configuration file is self-documented. You have to review the different points and make changes according to your configuration.

If you are using the standard directories path then you only need to fill the part with the SMTP server configuration, the YobiDrive FX web application base url and optionally generate another server key.

3.3. WAR application deployment

Before installing the WAR file we need to review the garbage collection and max memory settings of the Tomcat server.

As a minimum the max memory settings (Xmx) of the Tomcat server must be set to 768m. If the quantity of data your are handling increases then you will need to increase this setting too.

```
> nano /etc/init.d/tomcat6
```

Add the following line

```
JAVA_OPTS="-Xmx512m -XX:PermSize=128m -XX:MaxPermSize=256m -XX:
+UseConcMarkSweepGC"
```

just after the start option:

```
case "$1" in
  start)

    JAVA_OPTS="-Xmx512m -XX:PermSize=128m -XX:MaxPermSize=256m -XX:+UseConcMarkSweepGC"
    echo $JAVA_OPTS

    if [ -z "$JAVA_HOME" ]; then
      log_failure_msg "no JDK found - please set JAVA_HOME"
      exit 1
    fi
```



The last step is to install the WAR file via the Tomcat management console.

Restart the tomcat server:

```
> /etc/init.d/tomcat6 restart
```

Then access the Tomcat manager application via the following url:

```
https://yobidrive.mycompany.com/manager/html
```

1. Stop and undeploy the root application (the one with path “/”):
 1. Click on stop
 2. Click on undeploy
2. Upload the YobiDrive server application ROOT.war file (found in the installation package)
 1. Browse for the file
 2. Click on deploy

The file is now uploaded. Restart the tomcat server with:

```
> /etc/init.d/tomcat6 restart
```

If it's the first start of the server you should monitor the Tomcat log file to ensure the startup is fine.

```
> tail -f /opt/tomcat/logs/catalina.out
```

You should see several lines of information which are the sign of a correct startup:

```
INFO com.yobidrive.diskmap.DiskMapManager - Storage engine for store  
configuration / zone 0 ready  
INFO com.yobidrive.diskmap.DiskMapManager - Storage engine for store  
fsmetadata / zone 0 ready
```

If the Tomcat server looks OK then you can also have a look at the YobiDrive server log file.

```
> cat /opt/yobifolders/_LOGS/log_ALLDESKS.log  
STARTUP ; Server starting up: community edition 10 users  
STARTUP ; Server installation id: 0000687c52cb-7f52-xxxx
```

If you see those two lines it means that the YobiDrive server application is up and running.



4. YobiDrive FX web-sharing installation

The YobiDrive FX application is available either as an open source project or as a pre-build package. In this document we will install the pre-build package.

For more information on how to generate a pre-build package from the sources please refer to <https://code.google.com/p/yobidrive-fx/>

4.1. Package installation

First step is to upload the package to your server in the folder */opt*.

Then we simply unzip it.

```
> mkdir /opt/yobidrivefx
> mv /opt/yobidrive-fx.zip /opt/yobidrivefx/
> cd /opt/yobidrivefx/
> unzip yobidrive-fx.zip
```

We now simply need to generate an alternate configuration file with two entries and add it to the startup script.

```
> nano /opt/yobidrivefx/lib/yobidrivefx.conf
```

And put the following content inside this file:

```
include "application.conf"
yobidrive.urls.dlbylinkurl="https://yobidrive.mycompany.com/y3/link"
```

You of course need to replace *yobidrive.mycompany.com* with the name of the virtual host used to directly access the YobiDrive core application (virtual host defined in chapter 2.3.4).

Then edit the startup script to reference this configuration file and define the port to use:

```
> nano /opt/yobidrivefx/start
```

Ensure the start script contains the following:

```
#!/usr/bin/env sh
exec java -Dhttp.port=9000
-Dconfig.file=/opt/yobidrivefx/lib/yobidrivefx.conf $* -cp "`dirname
$0`/lib/*" play.core.server.NettyServer `dirname $0`
```

You can then start the application in background with:

```
> /opt/yobidrivefx/start &
```

A better solution is to create an init.d script as described in chapter 4.3



4.2. VirtualHost yobidrivefx.mycompany.com

This virtual host will act as a reverse proxy that will forward all the requests to the YobiDrive FX application.

```
> nano /etc/apache2/sites-available/yobidrivefx-ssl
```

Paste the following content inside the file, replacing **yobidrivefx.mycompany.com** with your desired hostname.

```
#Redirect all HTTP traffic to HTTPS
<VirtualHost *:80>
    ServerName yobidrivefx.mycompany.com
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule (.*?) https://%{HTTP_HOST}%{REQUEST_URI}
</VirtualHost>

<VirtualHost *:443>
    ServerName yobidrivefx.mycompany.com
    ProxyRequests Off

    <Proxy *>
        Order Allow,Deny
        Allow From All
    </Proxy>

    ProxyPreserveHost On

    #Upload directly done on the Y3 backend
    ProxyPass /files-ul http://127.0.0.1:8080/y3-upload
    ProxyPassReverse /files-ul http://127.0.0.1:8080/y3-upload
    #Y3 calls forwarded to the Y3 backend
    ProxyPass /y3 http://127.0.0.1:8080/y3
    ProxyPassReverse /y3 http://127.0.0.1:8080/y3

    #Everything else to the play application on port 9000
```

ProxyPass / http://127.0.0.1:9000/

ProxyPassReverse / http://127.0.0.1:9000/

RequestHeader set X-Forwarded-Proto "https"

AllowEncodedSlashes On

DocumentRoot /var/www-forbidden

<Directory /var/www-forbidden/>

Options Indexes FollowSymLinks MultiViews

AllowOverride None

Order allow,deny

deny from all

</Directory>

SSLEngine on

SSLCertificateFile /etc/apache2/ssl/apache.pem

SSLCertificateKeyFile /etc/apache2/ssl/apache.key

<FilesMatch "\.(cgi|shtml|phtml|php)\$">

SSLOptions +StdEnvVars

</FilesMatch>

<Directory /usr/lib/cgi-bin>

SSLOptions +StdEnvVars

</Directory>

BrowserMatch "MSIE [2-6]" \

nokeepalive ssl-unclean-shutdown \

downgrade-1.0 force-response-1.0

BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>



Enable the site and reload Apache

```
> a2enmod proxy
> a2enmod proxy_http
> a2enmod rewrite
> a2enmod headers
> a2ensite yobidrivefx-ssl
> /etc/init.d/apache2 restart
```

You can now access the application via the URL:

<https://yobidrivefx.mycompany.com>

The first time you will connect to the application you will be prompted to create the first administrator account.

4.3. Startup script

The following startup script allows you to easily start, stop and restart your YobiDrive FX instance.

```
> nano /etc/init.d/yobidrivefx
```

Paste the following content inside the file:

```
#!/bin/bash
APP_PATH="/opt/yobidrivefx"
start() {
    $APP_PATH/start &
}

stop() {
    kill `cat $APP_PATH/RUNNING_PID`
}

case "$1" in
    start)
        cd $APP_PATH
        echo `pwd`
        start > $APP_PATH/logs/application.log 2>&1 &
        echo "$APP started."
        ;;
    stop)

```

```
stop
echo "$APP stopped."
;;
restart)
stop
sleep 2
cd $APP_PATH
start > $APP_PATH/logs/application.log 2>&1 &
echo "$APP restarted."
;;
*)
N=/etc/init.d/play.$APP
echo "Usage: $N {start|stop|restart}" >&2
exit 1
;;
esac
exit 0
```

Make the file runnable

```
> chmod u+x /etc/init.d/yobidrivefx
```

And you can now start, stop or restart the application using

```
> /etc/init.d/yobidrivefx start
> /etc/init.d/yobidrivefx stop
> /etc/init.d/yobidrivefx restart
```



5. What to do next ?

You now have a fully working single node YobiDrive server and a professional-file sharing application running on top of it, but there is still more you can do.

5.1. Increase the registered number limitation

By default there is a limit in the maximum number of registered users on the YobiDrive Community edition server.

You can see this limit by looking inside the `/opt/yobifolders/_LOGS/log_ALLDESKS.log` file at server startup time:

```
Server starting up: community edition 10 users
```

You can ask for this limit to be removed by contacting us. It's completely free and will allow us to know you better. Before registering you will need to recover your installation id. You can recover it in three ways:

- By looking in the `log_ALLDESK.log` file
- By going to the url <https://yobidrive.mycompany.com/y3/version>
- By login into the YobiDrive FX application as an admin user, the installation id will be printed in the footer of the page

With this installation id simply go to the YobiDrive web-site at <http://www.yobidrive.com> and follow the instructions.

We will provide you a license file, i.e. `license.ezc.xml`. To install the license file you need to copy the file `license.ezc.xml` into the directory `/opt/yobifolders/SMARTFOLDERS/IMPORT`.

After the file is copied make sure that the tomcat6 user has write access on it.

```
> chown tomcat6 /opt/yobifolders/SMARTFOLDERS/IMPORT/license.ezc.xml
```

Wait a little and after a few seconds (between 30 seconds and 1 minutes), the file will be replaced by a `license.ezc.ok` and `license.ezc.res` files, meaning that the file has been correctly imported. If after several minutes you don't see anything please make sure that the YobiDrive server application is up and running.

You then only need to restart the tomcat server for the new license to take effect.

```
> /etc/init.d/tomcat6 restart
```

You can see it the `log_ALLDESKS.log` file if the new license has been taken into effect.

```
Server starting up: community edition 1000 users
```

If the log file shows entry like the one below, you probably have given us the wrong installation id.

```
Installed license is not valid for this installation
```

5.2. Customize your YobiDrive FX application

YobiDrive FX application is released under the open source Apache 2 license meaning that you can



freely change anything in it: from small changes like modifying colors, logo or texts to big changes like creating new screens or implementing new functionalities.

To download the source code and learn how to build a package please refer to the following site: <https://code.google.com/p/yobidrive-fx/>

5.3. Build your own application on top of the YobiDrive server

The YobiDrive FX file sharing is only an example of what can be built on top of the YobiDrive server platform.

The YobiDrive Y3 API is an HTTP REST that can be used with almost any programming languages.

To download the API documentation please visit <http://www.yobidrive.com>, to download the open source Java SDK (the one used to build YobiDrive FX) go to <https://code.google.com/p/yobidrive-y3sdk-java/>

If you have built an application on top of YobiDrive or have some requests about the API we would be glad to hear about it. Don't hesitate to contact us via our website <http://www.yobidrive.com>

6. Troubleshooting

```
catalina.out log file shows line of the form Problem creating output files in: /opt/yobifolders/_VOLDEMORT/data/bdb/je.info
```

Ensure you have given tomcat6 write permissions on directory */opt/yobifolders* as described in chapter 3.1

```
catalina.out log file shows OutOfMemoryError at initialization time.
```

Ensure you have set the Xmx parameter in the Tomcat startup script as described in chapter 3.3. If you have tweaked the server.properties file then the Xmx value may need to be increased.