

LibreOffice Online

Creación de imagen Docker sin restricciones de uso

20 de abril de 2020



Hoja de Control del Documento

Información del Documento			
Título	LibreOffice Online		
Asunto	Creación de imagen Docker sin restricciones de uso		
Nombre del fichero	DGTD_20200420_LibreOffice Online Docker.odt		
Versión	1.0		
Elaborado por	OTGP-CC-OTI	Fecha Elaboración	20/04/2020
Aprobado por		Fecha Aprobación	
Confidencialidad			

Control de Versiones			
Versión	Descripción de los cambios	Elaborado por	Fecha Elaboración
01.00	Elaboración inicial del documento	OTGP	20/04/2020

Lista de Distribución	
Apellidos, Nombre	Cargo / Función
Francisco Mesa Villalba	Adjunto Sv. Administración Electrónica
Equipo OTGP	



ÍNDICE

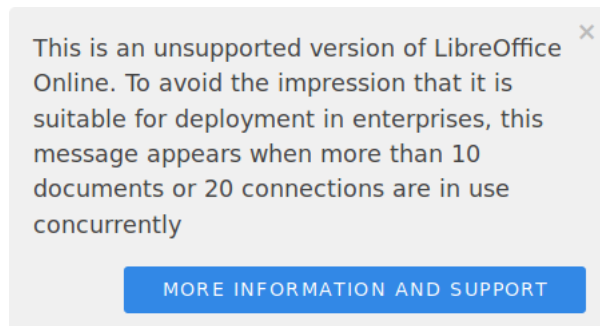
Sumario

1.Objeto.....	4
2.Introducción.....	4
3.Preparación del entorno.....	4
4.Compilación de LibreOffice.....	6
5.Compilación de LibreOffice Online.....	6
6.Creación de imagen Docker.....	7
7.Anexo A. Fichero Dockerfile.....	8
8.Anexo B. Fichero run-lool.sh.....	9



1. Objeto

Este documento describe las tareas a realizar para obtener una imagen Docker de LibreOffice Online sin el mensaje de advertencia relacionado con el soporte del producto en el ámbito empresarial.



2. Introducción

La Consejería de Hacienda, Industria y Energía tiene desplegada la herramienta LibreOffice Online en un entorno Red Hat Openshift.

El Servicio de Producción ha utilizado para este despliegue la imagen libreoffice/online:master basada en Ubuntu 18.04 LTS y disponible en <https://hub.docker.com/r/libreoffice/online/>. La imagen se ha creado a partir de los recursos disponibles en <https://cgit.freedesktop.org/libreoffice/online/tree/docker>.

Este documento describe las tareas a realizar para generar una imagen Docker personalizada de la herramienta LibreOffice Online, tomando como referencia la información proporcionada en el enlace anterior y configurando la compilación para obtener una versión sin las restricciones establecidas por defecto respecto al número de documentos y de conexiones concurrentes. Se abordará la preparación del entorno, compilación de la suite ofimática LibreOffice, compilación de la herramienta LibreOffice Online y finalmente creación de la imagen Docker.

3. Preparación del entorno

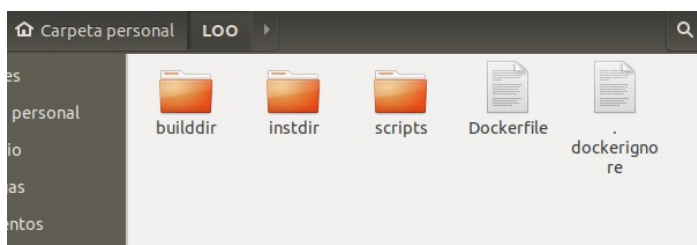
Con objeto de reducir las incidencias que se puedan producir en el proceso se utilizará Ubuntu 18.04 LTS para la compilación de los distintos proyectos y creación de la imagen.

Se realizarán las tareas siguientes como paso previo a la compilación:

- 1) Habilitar el repositorio de código fuente para el repositorio main.
 Editar el fichero `/etc/apt/sources.list` y eliminar el comentario de la línea `"deb-src http://es.archive.ubuntu.com/ubuntu/ bionic main restricted"`.
- 2) Refrescar el listado de paquetes disponibles para instalación tras la actualización de los repositorios.
`sudo apt-get update`
- 3) Instalar las dependencias de compilación de la suite ofimática LibreOffice.
`sudo apt-get build-dep libreoffice`
- 4) Instalar las dependencias de compilación de LibreOffice Online.
`sudo apt-get install python3-lxml python3-polib libcap-dev libpam0g-dev libpocoxml50 libpocoutil50 libpoconetssl50 libpoconet50 libpocojson50 libpocofoundation50 libpococrypto50 libpoco-dev`



- 5) Instalar la dependencia de compilación NPM de LibreOffice Online. Si bien este paquete se encuentra disponible en la distribución para su instalación, existe un conflicto con libpoco-dev que dificulta que ambos paquetes se puedan instalar simultáneamente en el sistema. Para salvar esta dificultad se opta por instalar el paquete npm desde NodeSource (<https://github.com/nodesource/distributions>).
 - a) `sudo apt-get install curl`
 - b) `curl -sSL https://deb.nodesource.com/gpgkey/nodesource.gpg.key | sudo apt-key add -`
 - c) `VERSION=node_13.x`
 - d) `DISTRO="$(lsb_release -s -c)"`
 - e) `echo "deb https://deb.nodesource.com/$VERSION $DISTRO main" | sudo tee /etc/apt/sources.list.d/nodesource.list`
 - f) `echo "deb-src https://deb.nodesource.com/$VERSION $DISTRO main" | sudo tee -a /etc/apt/sources.list.d/nodesource.list`
 - g) `sudo apt-get update`
 - h) `sudo apt-get install nodejs`
- 6) Instalar la herramienta GIT para descarga del código fuente.
`sudo apt-get install git`
- 7) Instalar la herramienta Docker.
`sudo apt-get install docker.io`
- 8) Crear carpeta de trabajo donde se descargará el código fuente de las aplicaciones, la configuración de la imagen de Docker, generación de binarios e imagen Docker. Para los comandos que se indican a continuación se asumirá que la carpeta de trabajo es /home/usuario/LOO, donde usuario es el identificador de la cuenta utilizada para acceder al sistema.



- a) Carpeta raíz.
`mkdir /home/usuario/LOO`
- b) Subcarpeta de almacenamiento del código fuente y binarios /home/usuario/LOO/builddir.
`mkdir /home/usuario/LOO/builddir`
- c) Subcarpeta de almacenamiento de binarios para instalación /home/usuario/LOO/instdir.
`mkdir /home/usuario/LOO/instdir`
- d) Subcarpeta con script de ejecución del servicio LibreOffice Online /home/usuario/LOO/scripts.
`mkdir /home/usuario/LOO/scripts/`
- e) Creación del fichero /home/usuario/LOO/scripts/run-lool.sh a partir del contenido de <https://cgjt.freedesktop.org/libreoffice/online/tree/docker/scripts/run-lool.sh>
- f) Creación del fichero /home/usuario/LOO/Dockerfile a partir del contenido de <https://cgjt.freedesktop.org/libreoffice/online/tree/docker/Dockerfile>
- g) Creación del fichero /home/usuario/LOO/.dockerignore a partir del contenido de <https://cgjt.freedesktop.org/libreoffice/online/tree/docker/.dockerignore>



4. Compilación de LibreOffice

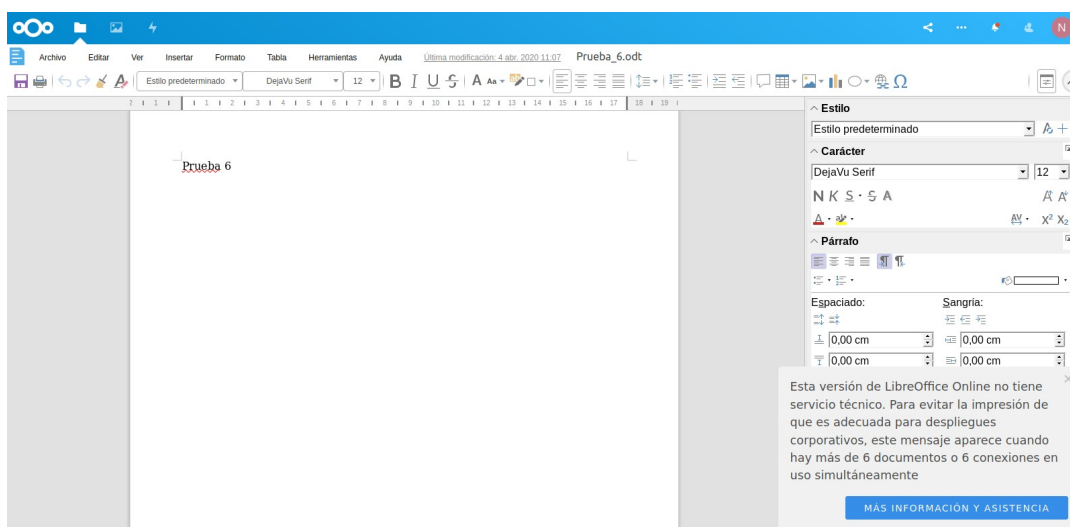
Se realizarán las tareas siguientes para la compilación e instalación del software:

- 1) Descargar el código fuente de la rama master en /home/usuario/LOO/builddir/libreoffice.
Nota: Para esta guía se utilizó el commit <https://git.libreoffice.org/core/+/-/393aa2b3dee9cca84a215635060b369a962ab179>.
 - a) `cd /home/usuario/LOO/builddir`
 - b) `git clone https://git.libreoffice.org/core libreoffice`
- 2) Configurar la compilación.
 - a) `cd /home/usuario/LOO/builddir/libreoffice`
 - b) `./autogen.sh --with-distro=LibreOfficeOnline --enable-release-build --with-vendor="Junta de Andalucía"`
- 3) Compilar el código.
`make build-nocheck`
- 4) Copiar los binarios generados a la carpeta de instalación.
 - a) `mkdir /home/usuario/LOO/instdir/opt`
 - b) `cp -a /home/usuario/LOO/builddir/libreoffice/instdir /home/usuario/LOO/instdir/opt/libreoffice`

5. Compilación de LibreOffice Online

Se realizarán las tareas siguientes para la compilación e instalación del software:

- 1) Descargar el código fuente de la rama master en /home/usuario/LOO/builddir/online.
Nota: Para esta guía se utilizó el commit <https://git.libreoffice.org/online/+/-/99787522581af296c25863f4ee1b24b170902a22>.
 - a) `cd /home/usuario/LOO/builddir`
 - b) `git clone https://git.libreoffice.org/online online`
- 2) Configurar la compilación. Los parámetros `with-max-connections` y `with-max-documents` determinan el límite máximo de conexiones y documentos concurrentes. A modo de ejemplo se muestra en la siguiente captura una configuración de máximo 6 conexiones y 6 documentos.



- a) `cd /home/usuario/LOO/builddir/online`

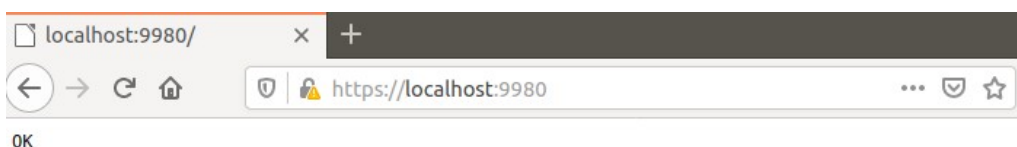


- b) `./autogen.sh`
- c) `./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var --enable-silent-rules --with-lokit-path=/home/usuario/LOO/builddir/libreoffice/include --with-lo-path=/opt/libreoffice --with-max-connections=100000 --with-max-documents=100000`
- d) Compilar el código.
`make`
- e) Copiar los binarios generados a la carpeta de instalación.
`DESTDIR=/home/usuario/LOO/instdir`
`make install`

6. Creación de imagen Docker

Se realizarán las tareas siguientes para la generación y prueba básica de la imagen Docker:

- 1) Generar la imagen utilizando Dockerfile descargado previamente (véase apartado “3. Preparación del entorno”).
 - a) `cd /home/usuario/LOO`
 - b) `sudo docker build --no-cache -t libreoffice/online:master .`
- 2) Comprobar la imagen generada.
 - a) Arranque de la imagen.
`sudo docker container run -t -d -p 127.0.0.1:9980:9980 -e "username=admin" -e "password=admin" libreoffice/online:master`
 - b) Comprobación de parámetros de configuración de la imagen relativos a concurrencia. Se listarán los contenedores cargados y se consultarán los logs del contenedor de libreoffice/online:master. En el log se comprobarán las líneas “Maximum concurrent open Documents limit: 100000 | wsd/LOOLWSD.cpp:1174” y “Maximum concurrent client Connections limit: 100000 | wsd/LOOLWSD.cpp:1175”.
`sudo docker container list`
`sudo docker logs <CONTAINER ID>`
 - c) Comprobación básica del funcionamiento del contenedor libreoffice/online:master. Al consultar al servicio se deberá obtener el mensaje “OK”.
`curl -insecure https://localhost:9980`



7. Anexo A. Fichero Dockerfile

Contenido del fichero Dockerfile obtenido de <https://cgit.freedesktop.org/libreoffice/online/tree/docker/Dockerfile>

```
# This file is part of the LibreOffice project.
#
# This Source Code Form is subject to the terms of the Mozilla Public
# License, v. 2.0. If a copy of the MPL was not distributed with this
# file, You can obtain one at http://mozilla.org/MPL/2.0/.

FROM ubuntu:18.04

# refresh repos otherwise installations later may fail
RUN apt-get update

# install LibreOffice run-time dependencies
# install adduser, findutils, openssl and cpio that we need later
# install an editor
RUN apt-get -y install locales-all libpng16-16 fontconfig adduser cpio findutils nano libpocoxml50 libpocoutil50 libpoconetssl50
libpoconet50 libpocojson50 libpocofoundation50 libpococrypto50 libcap2-bin openssl inotify-tools procps libxcb-shm0 libxcb-
render0 libxrender1 libxext6

# tdf#117557 - Add CJK Fonts to LibreOffice Online Docker Image
RUN apt-get -y install fonts-wqy-zenhei fonts-wqy-microhei fonts-droid-fallback fonts-noto-cjk

# copy freshly built LibreOffice master and LibreOffice Online master with latest translations
COPY /instdir /

# copy the shell script which can start LibreOffice Online (loolwsd)
COPY /scripts/run-lool.sh /

# set up LibreOffice Online (normally done by postinstall script of package)
RUN setcap cap_fowner,cap_mknod,cap_sys_chroot=ep /usr/bin/loolforkit
RUN adduser --quiet --system --group --home /opt/lool lool
RUN mkdir -p /var/cache/loolwsd && chown lool: /var/cache/loolwsd
RUN rm -rf /var/cache/loolwsd/*
RUN rm -rf /opt/lool
RUN mkdir -p /opt/lool/child-roots
RUN loolwsd-systemplate-setup /opt/lool/systemplate /opt/libreoffice >/dev/null 2>&1
RUN touch /var/log/loolwsd.log
# Fix permissions
RUN chown lool:lool /var/log/loolwsd.log
RUN chown -R lool:lool /opt/
RUN chown -R lool:lool /etc/loolwsd

EXPOSE 9980

# switch to lool user (use numeric user id to be compatible with Kubernetes Pod Security Policies)
USER 101

CMD bash /run-lool.sh
```



8. Anexo B. Fichero run-lool.sh

Contenido del fichero Dockerfile obtenido de <https://cgit.freedesktop.org/libreoffice/online/tree/docker/scripts/run-lool.sh>

```
#!/bin/bash
# This file is part of the LibreOffice project.
#
# This Source Code Form is subject to the terms of the Mozilla Public
# License, v. 2.0. If a copy of the MPL was not distributed with this
# file, You can obtain one at http://mozilla.org/MPL/2.0/.

export LC_CTYPE=en_US.UTF-8

# Fix domain name resolution from jails
cp /etc/resolv.conf /etc/hosts /opt/lool/systemplate/etc/

if test "${DONT_GEN_SSL_CERT-set}" == set; then
# Generate new SSL certificate instead of using the default
mkdir -p /opt/ssl/
cd /opt/ssl/
mkdir -p certs/ca
openssl rand -writerand /opt/lool/.rnd
openssl genrsa -out certs/ca/root.key.pem 2048
openssl req -x509 -new -nodes -key certs/ca/root.key.pem -days 9131 -out certs/ca/root.crt.pem -subj
"/C=DE/ST=BW/L=Stuttgart/O=Dummy Authority/CN=Dummy Authority"
mkdir -p certs/{servers,tmp}
mkdir -p certs/servers/localhost
openssl genrsa -out certs/servers/localhost/privkey.pem 2048
openssl req -key certs/servers/localhost/privkey.pem -new -sha256 -out certs/tmp/localhost.csr.pem -subj
"/C=DE/ST=BW/L=Stuttgart/O=Dummy Authority/CN=localhost"
openssl x509 -req -in certs/tmp/localhost.csr.pem -CA certs/ca/root.crt.pem -CAkey certs/ca/root.key.pem -CAcreateserial -
out certs/servers/localhost/cert.pem -days 9131
mv certs/servers/localhost/privkey.pem /etc/loolwsd/key.pem
mv certs/servers/localhost/cert.pem /etc/loolwsd/cert.pem
mv certs/ca/root.crt.pem /etc/loolwsd/ca-chain.cert.pem
fi

# Replace trusted host
perl -pi -e "s/localhost<\/host>/${domain}<\/host>/g" /etc/loolwsd/loolwsd.xml
perl -pi -e "s/<username (.*)>.*<\/username>/<username \1>${username}<\/username>/" /etc/loolwsd/loolwsd.xml
perl -pi -e "s/<password (.*)>.*<\/password>/<password \1>${password}<\/password>/" /etc/loolwsd/loolwsd.xml

# Restart when /etc/loolwsd/loolwsd.xml changes
[ -x /usr/bin/inotifywait -a /usr/bin/pkill ] && (
    /usr/bin/inotifywait -e modify /etc/loolwsd/loolwsd.xml
    echo "$(ls -l /etc/loolwsd/loolwsd.xml) modified -> restarting"
    pkill -f -signal 1 loolwsd
) &

# Start loolwsd
exec /usr/bin/loolwsd -version -o:sys_template_path=/opt/lool/systemplate -o:child_root_path=/opt/lool/child-roots -
o:file_server_root_path=/usr/share/loolwsd ${extra_params}
```

