



OTOB Installation Guide

Release 11.0

Rother OSS GmbH

5 | 13, 2024

1	简介	3
1.1	项目简介	3
2	环境搭建	5
2.1	环境	7
2.2	数据库	7
3	OTOBO	9
3.1	在Linux上安装SELinux	9
3.2	在1号服务器上安装OTOBO	10
3.3	在2号服务器上安装Perl	10
3.4	在3号服务器上安装OTOBO	11
3.5	在4号服务器上安装	11
3.6	在5号服务器上安装Apache	11
3.6.1	在5号服务器上安装SSL	12
3.6.2	在5号服务器上安装SSL	12
3.7	在6号服务器上安装	12
3.8	在7号服务器上安装	13
3.9	在8号服务器上安装Elasticsearch	14
3.9.1	在Ubuntu 18.04 LTS上安装Elasticsearch	14
3.9.2	在Linux上安装Elasticsearch	14
3.9.3	Elasticsearch	14
3.9.4	Elasticsearch	14
3.10	在9号服务器上安装	15
3.11	在10号服务器上安装	15
3.12	在11号服务器上安装OTOBO	15
3.13	在12号服务器上安装Cron	15
3.14	在13号服务器上安装Bash	15
3.15	在14号服务器上安装	16
4	使用 Docker 和 Docker Compose	17
4.1	简介	17
4.2	安装	18
4.2.1	1. Clone the otobo-docker repo	18
4.2.2	2. Create an initial .env file	18
4.2.3	3. Configure the password for the database admin user	18
4.2.4	4. Set up a volume with SSL configuration for the nginx webproxy (optional)	19

4.2.5	安装 Docker/Docker Compose	19
4.2.6	6. Install and start OTOBO	19
4.3	安装	20
4.3.1	Docker	20
4.3.2	Docker	20
4.3.3	Docker	20
4.4	配置	21
4.4.1	Nginx	21
4.4.2	在 Nginx 中配置 Kerberos	23
4.4.3	配置	23
4.4.4	配置	23
4.4.5	配置	23
4.4.6	在 OTOBO Docker Compose	24
4.4.7	在 OTOBO Docker 中	24
4.4.8	配置	25
4.4.9	配置	26
4.4.10	配置	26
4.5		27
5	Migration from OTRS 6 or OTRS 7 / ((OTRS)) Community Edition to OTOBO version 10.1	29
5.1		29
5.2		30
5.3	在 1 中安装 OTOBO	30
5.4	在 2 中安装 OTOBO	31
5.5	在 3 中安装 OTOBO	31
5.6	在 /opt/otrs 中配置	32
5.7	在 /opt/otrs 中配置 ssh	32
5.8	在 4 中安装 OTRS / ((OTRS))	32
5.8.1	安装 OTRS	33
5.8.2	配置	33
5.9	Docker	33
5.9.1	Docker / opt / otrs 中的 otobo_opt_otobo *	33
5.10	在 5 中配置	34
5.11	在 6 中配置	35
5.12		35
5.12.1	1. Login after migration not possible	35
5.12.2	2. Final page of the migration has a strange layout due to missing CSS files	35
5.12.3	3. MySQL	35
5.12.4	4. PostgreSQL	35
5.12.5	5.	36
5.13	在 7 中配置	36
5.13.1	1. Password policy rules	36
5.13.2	2. Docker 中的 cron	36
5.14		36
5.14.1	在 Oracle 中配置 Oracle	36
5.14.2	配置	38
6	配置	41
6.1	在 1 中安装 OTOBO	41
6.2	在 2 中配置	42
6.2.1	在 Ubuntu 中配置 MySQL	42
6.3	在 3 中配置	42
6.3.1	配置	42
6.3.2	配置	42

6.3.3	XXXXXXXXXXXXXXXXXXXX	43
6.3.4	XXXXXXXXXX	43
6.3.5	XX Apache XXXX	43
6.4	XX 4XXXXXXXXXXXX perl XX	43
6.5	XX 5 XXXXXXXXXXXXXXXXXXXX	43
6.6	XX 6XXXXXXXXXXXXXXXXXXXX 10.0 XXXX 10.1X	44
6.7	XX 7 XXXXXXXX	44
7	XXXXXXXXXX Docker XXX OTOBO 10	45
7.1	Updating the Docker Compose files	45
7.2	XX Docker Compose .env XX	46
7.3	XX Docker XXXX	46
7.4	XX OTOBO	46
8	XXXXXX	49
8.1	XX	49
8.2	XX	50
8.3	XX Docker XXXX OTOBO XXXX	50
9	XXXXXX	53
9.1	XX Docker XXXX OTOBO XXXX	53
10	OTOBO Docker XXXX Kerberos XXXX	55
10.1	XXXXXXXXXX	55
10.2	XX Active Directory Keytab XX	55
10.3	XXXX nginx XXXXXXX	57
10.4	XXXX OTOBO .env XX	57
10.5	XX OTOBO	58
10.6	XX OTOBO XX Kerberos XX	58
10.7	XXXXXXXXXX Kerberos SSO	58
10.8	XXXXXX	59
10.8.1	Kerberos XX	59
11	XXXXXXXXXXXXXXXXXXXX	61
11.1	XXXXXXXXXX	61
11.2	XXXXXXXXXX	61
11.2.1	XXXXXXXXXXXXXXXXXXXX	62
11.2.2	XXXXXXXXXXXXXXXXXXXX	63
12	XX CPAN XX Perl XX	65
12.1	XX Docker XXXX	65
13	XXXXXX	67
13.1	XXXXXX	67
13.2	XXXXXX	67
13.3	XXXXX	69
13.3.1	XXXX	69
13.3.2	XXXX	69
13.4	XXXXX	70
13.5	XXXXX	70
13.6	XX	71
13.6.1	XXXXX Redis Cache XXXX	71
13.6.2	XXXXXXXXXX	71
13.7	XX	72



OTRS AG (<https://otrs.com>), Zimmersmühlenweg 11, 61440 Oberursel, Germany.

Copyright © for modifications and amendments 2019-2022 ROTHER OSS GmbH (<https://otobo.de>), Oberwalting 31, 94339 Leiblfing, Germany

OTRS GNU GPL 1.3

Rother OSS GNU GPL 1.3

Rother OSS GmbH, (<https://otobo.de>), Oberwalting 31, 94339 Leiblfing, Germany.

OTRS AG (original version), Rother OSS GmbH (<https://otobo.de>)



OTOBO is licensed under the GNU GPL license. Linux is a registered trademark.

1.1 Introduction

OTOBO is a simple tool.

OTOBO is a simple tool.

OTOBO is a simple tool. "username> command-to-execute" "root>"

Warning: "username>"

OTOBO is a simple tool. /opt/otobo OTOBO is a simple tool.

```
root> ln -s /path/to/otobo /opt/otobo
```


CHAPTER 2

□□□□

OTOBO 支持 Linux Unix OpenBSD FreeBSD Microsoft Windows OTOBO

支持 OTOBO

支持 OTOBO Docker Docker Kubernetes

OTOBO Perl CPAN Perl Perl rpm yast apt-get

```
otobo> /opt/otobo/bin/otobo.CheckModules.pl --inst
```

```
--list
```

```
otobo> /opt/otobo/bin/otobo.CheckModules.pl --list | more
```

```
root
```

```
Required packages:
  o Archive::Tar.....ok (v2.32)
  o Archive::Zip.....ok (v1.67)
  o Const::Fast.....ok (v0.014)
  o Date::Format.....ok (v2.24)
  o DateTime.....ok (v1.51)
    o DateTime::TimeZone.....ok (v2.38)
  o Convert::BinHex.....ok (v1.125)
  o DBI.....ok (v1.643)
  o Digest::SHA.....ok (v6.02)
  o File::chmod.....ok (v0.42)
  o List::AllUtils.....ok (v0.15)
  o LWP::UserAgent.....ok (v6.26)
  o Moo.....ok (v2.003006)
  o namespace::autoclean.....ok (v0.29)
  o Net::DNS.....ok (v1.22)
```

- o Net::SMTP::SSL.....ok (v1.04)
- o Path::Class.....ok (v0.37)
- o Sub::Exporter.....ok (v0.987)
- o Template::Toolkit.....ok (undef)
- o Template::Stash::XS.....ok (undef)
- o Text::CSV.....ok (v1.95)
- o Text::Trim.....ok (v1.04)
- o Time::HiRes.....ok (v1.9760)
- o Try::Tiny.....ok (v0.30)
- o URI.....ok (v1.71)
- o XML::LibXML.....ok (v2.0207)
- o YAML::XS.....ok (v0.81)
- o Unicode::Collate.....ok (v1.27)
- o CGI::PSGI.....ok (v0.15)
- o DBIx::Connector.....ok (v0.56)
- o Path::Class.....ok (v0.37)
- o Plack.....ok (v1.0047)
- o Plack::Middleware::ForceEnv.....ok (v0.02)
- o Plack::Middleware::Header.....ok (v0.04)
- o Plack::Middleware::Refresh.....ok (undef)
- o Plack::Middleware::ReverseProxy..ok (v0.16)
- o Plack::Middleware::Rewrite.....ok (v2.101)
- o SOAP::Transport::HTTP::Plack.....ok (v0.03)

Recommended features for setups using apache:

- o ModPerl::Util.....ok (v2.000011)

Database support (installing one is required):

- o DBD::mysql.....ok (v4.050)

Various features for additional functionality:

- o Encode::HanExtra.....ok (v0.23)
- o Net::LDAP.....ok (v0.66)
- o Crypt::Eksblowfish::Bcrypt.....ok (v0.009)
- o XML::LibXSLT.....ok (v1.99)
- o XML::Parser.....ok (v2.46)

Features enabling communication with a mail-server:

- o Net::SMTP.....ok (v3.11)
- o Mail::IMAPClient.....ok (v3.42)
- o Authen::SASL.....ok (v2.16)
- o Authen::NTLM.....ok (v1.09)
- o IO::Socket::SSL.....ok (v2.067)

Optional features which can increase performance:

- o JSON::XS.....ok (v4.02)
- o Text::CSV_XS.....ok (v1.41)

Required packages if you want to use PSGI/Plack (experimental and advanced):

- o Gazelle.....ok (v0.49)
- o Linux::Inotify2.....ok (v2.2)
- o Plack::App::File.....ok (undef)

2.1 系统要求

OTOBOSystemRequirements OTOBOSystemRequirements

- CPU
- 4 GB RAM
- 10 GB Disk Space

OTOBOSystemRequirements

- 3 GHz Xeon CPU
- 8 GB RAM 16 GB
- 40 GB Disk Space

Note: OTOBOSystemRequirements OTOBOSystemRequirements

2.2 软件依赖

Perl

- Perl 5.24.0
- `"/opt/otobo/bin/otobo.CheckModules.pl"` Perl

Apache

- Apache HTTP 2.4

Database

- MySQL 5.6
- MariaDB
- PostgreSQL 9.2
- Oracle 10g

Search

- Elasticsearch 7.x
- Redis
- nginx SSL

Browsers

- Safari
- Chrome
- Internet Explorer 11
- Edge
- Mozilla Firefox

- `enableJavaScript`

OTOBO
OTOBO

Note: OTOBO 10.0.7 Docker Docker Compose OTOBO Docker Elasticsearch Redis Cache Docker <https://doc.otobo.org/manual/installation/10.1/en/content/installation-docker.html>

3.1 SELinux

Note: SELinux OTOBO

SELinux "sestatus" `getenforce`
"sestatus" SELinux SELinux "SELinux" SELinux
RHEL/CentOS/Fedora SELinux

1. `/etc/selinux/config` SELINUX=disabled

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

2. Verify that "getenforce" is disabled

```
root> getenforce
Disabled
```

3.2 Install OTOBO

Download the OTOBO tarball from <https://ftp.otobo.org/pub/otobo/> and extract it to /root/otobo-update

```
root> mkdir /opt/otobo-install && mkdir /opt/otobo # Create a temporary install directory
root> cd /opt/otobo-install # Change into the update directory
root> wget https://ftp.otobo.org/pub/otobo/otobo-latest-10.1.tar.gz # Download the latest OTOBO 10 release
root> tar -xzf otobo-latest-10.1.tar.gz # Unzip OTOBO
root> cp -r otobo-10.x.x/* /opt/otobo # Copy the new otobo directory to /opt/otobo
```

3.3 Install Perl

Install the Perl modules required by OTOBO from CPAN

Note: Debian users should install perl

```
apt-get install -y libarchive-zip-perl libtimedate-perl libdatettime-perl libconvert-binhex-perl libcgi-psgi-perl libdbi-perl libdbix-connector-perl libfile-chmod-perl liblist-allutils-perl libmoo-perl libnamespace-autoclean-perl libnet-dns-perl libnet-smtp-ssl-perl libpath-class-perl libsub-exporter-perl libtemplate-perl libtext-trim-perl libtry-tiny-perl libxml-libxml-perl libyaml-libyaml-perl libdbd-mysql-perl libapache2-mod-perl2 libmail-imapclient-perl libauthen-sasl-perl libauthen-ntlm-perl libjson-xs-perl libtext-csv-xs-perl libpath-class-perl libplack-perl libplack-middleware-header-perl libplack-middleware-reverseproxy-perl libencode-hanextra-perl libio-socket-ssl-perl libnet-ldap-perl libcrypt-eksblowfish-perl libxml-libxslt-perl libxml-parser-perl libconst-fast-perl
```

```
root> perl /opt/otobo/bin/otobo.CheckModules.pl -list
Checking for Perl Modules:
  o Archive::Tar.....ok (v1.90)
  o Archive::Zip.....ok (v1.37)
  o Crypt::Eksblowfish::Bcrypt.....ok (v0.009)
  ...
```

Note: OTOBO requires perl version "5.10.0" or higher. RHEL users should install perl-core

Install CPAN on Linux

Install the Perl modules required by OTOBO from CPAN


```
root> /opt/otobo/bin/otobo.CheckModules.pl --inst
```

Note: OTOBO's `otobo.CheckModules.pl` script checks for the following dependencies:

3.4 3. OTOBO

OTOBO requires the following dependencies:

```
root> useradd -r -U -d /opt/otobo -c 'OTOBO user' otobo -s /bin/bash
```

OTOBO requires the following dependencies:

```
root> usermod -G www-data otobo
(SUSE=www, Red Hat/CentOS/Fedora=apache, Debian/Ubuntu=www-data)
```

3.5 4. OTOBO

OTOBO requires the following dependencies: `$OTOBO_HOME/Kernel/Config.pm.dist` file. Copy the file to the following location:

```
root> cp /opt/otobo/Kernel/Config.pm.dist /opt/otobo/Kernel/Config.pm
```

3.6 5. Apache

OTOBO requires Apache 2.4.x with the `mod_perl` module installed. Linux distributions provide Apache 2.4.x with the `mod_perl` module installed.

```
# RHEL / CentOS:
root> yum install httpd mod_perl

# SuSE:
root> zypper install apache2-mod_perl

# Debian/Ubuntu:
root> apt-get install apache2 libapache2-mod-perl2
```

Apache 2.4.x requires the `mpm_prefork` module. Apache 2.4.x also requires the `a2dismod` and `a2enmod` utilities.

```
root> # check which MPM is active
root> apache2ctl -M | grep mpm_
```

`mpm_prefork` module is installed.

`mpm_event` module is not installed.

```
root> a2dismod mpm_event
```

OTOBO requires the `mpm_worker` module.

```
root> a2dismod mpm_worker
```

mpm_prefork

```
root> a2enmod mpm_prefork
```

OTOBO Apache a2enmod

```
root> a2enmod perl
root> a2enmod deflate
root> a2enmod filter
root> a2enmod headers
```

Note: Apache

Apache“conf.d“Linux“/etc/apache“ ““/etc/apache2“

3.6.1 SSL Apache

/opt/otobo/scripts/apache2-httpd.include.conf apache sites-available

```
# Debian/Ubuntu:
root> cp /opt/otobo/scripts/apache2-httpd.include.conf /etc/apache2/sites-available/
↪ zzz_otobo.conf
root> a2ensite zzz_otobo.conf
root> systemctl restart apache2
```

3.6.2 SSL Apache

/opt/otobo/scripts/apache2-httpd-vhost-80.include.conf` ` /opt/otobo/scripts/apache2-httpd-vhost-443.include.conf“ Apache“sites-available“

```
# Debian/Ubuntu:
root> cp /opt/otobo/scripts/apache2-httpd-vhost-80.include.conf /etc/apache2/sites-
↪ available/zzz_otobo-80.conf
root> cp /opt/otobo/scripts/apache2-httpd-vhost-443.include.conf /etc/apache2/sites-
↪ available/zzz_otobo-443.conf
```

SSL OTOBO Apache

```
root> a2ensite zzz_otobo-80.conf
root> a2ensite zzz_otobo-443.conf
```

```
root> systemctl restart apache2
```

3.7 6

OTOBO

```
root> /opt/otobo/bin/otobo.SetPermissions.pl
```

3.8 7

Linux MySQL MariaDB PostgreSQL Oracle
Linux MySQL

```
# RHEL / CentOS:
root> yum install mysql-server

# SuSE:
root> zypper install mysql-community-server

# Debian/Ubuntu:
root> apt-get install mysql-server
```

MySQL

MySQL 5.7 OTOBO
MySQL "root"

```
root> mysql -u root
root> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY
↳ 'NewRootPassword';
```

MariaDB 10.1

```
root> mysql -u root
root> update mysql.user set authentication_string=password('NewRootPassword') plugin=
↳ 'mysql_native_password' where user='root';
```

```
root> mysql -u root
root> UPDATE mysql.user SET password = PASSWORD('NewRootPassword') WHERE user = 'root
↳ ';
root> UPDATE mysql.user SET authentication_string = '' WHERE user = 'root';
root> UPDATE mysql.user SET plugin = 'mysql_native_password' WHERE user = 'root';
```

OTOBO

Note: MySQL "[mysqld]" MySQL "/etc/my.cnf"/etc/mysql/my.cnf /etc/mysql/mysql.conf.d/mysqld.cnf

```
max_allowed_packet = 64M
innodb_log_file_size = 256M
```

MySQL 8.0

```
query_cache_size = 32M
```

MySQL "[mysqld]" MySQL "/etc/my.cnf"/etc/mysql/my.cnf /etc/mysql/mysql.conf.d/mysqld.cnf

```
max_allowed_packet = 64M
```

“mysqLTuner” Github “https://github.com/major/MySQLTuner-perl” Debian Ubuntu

```
root> apt-get install mysqltuner
```

```
root> mysqltuner --user root --pass NewRootPassword
```

3.9 8Elasticsearch

OTOBO Elasticsearch Elasticsearch OTOBO

3.9.1 Ubuntu 18.04 LTS Elasticsearch

JDK

```
root> apt update
root> apt install openjdk-8-jdk
```

Elasticsearch

```
root> wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key
↪add -
root> echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee
↪/etc/apt/sources.list.d/elastic-7.x.list
root> apt update
root> apt -y install elasticsearch
```

3.9.2 Linux Elasticsearch

<https://www.elastic.co/guide/en/elasticsearch/reference/current/setup.html>

3.9.3 Elasticsearch

OTOBO Elasticsearch

```
root> /usr/share/elasticsearch/bin/elasticsearch-plugin install --batch ingest-
↪attachment
root> /usr/share/elasticsearch/bin/elasticsearch-plugin install --batch analysis-icu
```

3.9.4 Elasticsearch

Elasticsearch

OTOBO jvm “/etc/elasticsearch/jvm.options” JVM

```
-Xms4g
-Xmx4g
```

4-10GB

Note: <https://www.elastic.co/guide/en/elasticsearch/reference/current/setup.html>

Elasticsearch

```
root> systemctl restart elasticsearch
```

3.10 9

<http://localhost/otobo/installer.pl> "localhost"

3.11 10

<http://localhost/otobo/index.pl> "root@localhost"

3.12 11 OTOBO

OTOBO cron OTOBO OTOBO

```
otobo> /opt/otobo/bin/otobo.Daemon.pl start
```

3.13 12 OTOBO Cron

/opt/otobo/var/cron/*.dist cron OTOBO Daemon ".dist"

```
root> cd /opt/otobo/var/cron/
root> for foo in *.dist; do cp $foo `basename $foo .dist`; done

root> cd /opt/otobo/
root> bin/Cron.sh start
```

3.14 13 Bash

OTOBO Linux

"bash-completion" "otobo" "/opt/otobo/.bash_completion"

shell TAB

DOCKER Docker Compose

DOCKER Docker OTOBO OTOBO Docker OTOBO

- db MariaDB
- elastic OTOBO Elasticsearch
- redis Redis
- *Gazelle Perl
- nginx Nginx HTTPS

OTOBO

4.1

- Docker 19.03.08
- DockerCompose 1.25.0
- Git 2.25.1

Note: <https://www.digitalocean.com/community/tutorials/how-to-install-docker-compose-on-ubuntu-18-04>
<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04>

git Docker Docker Compose Ubuntu 20.04

```
root> apt-get install git docker docker-compose curl
root> systemctl enable docker
```

Git Docker

4.2

Docker root Docker Doc

4.2.1 1. Clone the otopo-docker repo

Docker https://hub.docker.com otopo-docker Github git 10.0.4

Note: */opt/otobo-docker*

```
docker_admin> cd /opt
docker_admin> git clone https://github.com/RotherOSS/otobo-docker.git --branch
<BRANCH> --single-branch
docker_admin> ls otopo-docker # just a sanity check, README.md should exist
```

4.2.2 2. Create an initial .env file

Docker Compose .env OTOBO TLS .docker_compose_env_http .docker_compose_env_https .docker_compose_env_http OTOBO HTTP .docker_compose_env_https OTOBO Nginx HTTPS .docker_compose_env_https_custom_nginx .docker_compose_env_https Nginx .docker_compose_env_https_kerberos .docker_compose_env_https Kerberos * .docker_compose_env_http_selenium .docker_compose_env_https_selenium Selenium

Note: ls -a

OTOBO HTTPS44380HTTPS OTOBO Web HTTP HTTPS Nginx HTTPS

```
docker_admin> cd /opt/otobo-docker
docker_admin> cp -p .docker_compose_env_https .env # or .docker_compose_env_http for
HTTP
```

4.2.3 3. Configure the password for the database admin user

.env

OTOBO_DB_ROOT_PASSWORD

Note: 安装OTOBO前，请确保系统已安装Docker及docker-compose，并执行以下命令：
bash“

4.3 部署

部署OTOBO

4.3.1 Docker部署

otobo_web_1 容器5000端口OTOBO Web容器
otobo_daemon_1 OTOBO容器 OTOBO容器
otobo_db_1 容器3306端口MariaDB
otobo_elastic_1 容器9200/9300端口Elasticsearch
otobo_redis_1 Redis容器
otobo_nginx_1 nginx容器HTTPS

4.3.2 Docker配置

Docker容器配置

otobo_opt_otobo 容器 web 容器 daemon 容器 /opt/otobo
otobo_mariadb_data 容器 **db** 容器 /var/lib/mysql
otobo_elasticsearch_data 容器 'elastic' 容器 /usr/share/elasticsearch/data
otobo_redis_data 容器 "redis" 容器
otobo_nginx_ssl 容器TLS

4.3.3 Docker环境变量

环境变量

MariaDB

otobo_db_root_password MariaDB 容器 db 容器

Elasticsearch

Elasticsearch容器<https://www.elastic.co/guide/zh-CN/elasticsearch/reference/7.8/docker.html#docker-prod-prerequisites>

OTOBO_Elasticsearch_ES_JAVA_OPTS 容器* OTOBO_Elasticsearch_ES_JAVA_OPTS = -Xms512m -Xmx512m *容器4G

Webserver

otobo_web_http_port 容器HTTP容器80容器HTTPS容器HTTPS

nginx

HTTPS

otobo_web_http_port HTTP 80 HTTPS

otobo_web_https_port HTTPS 443

otobo_nginx_ssl_certificate Nginx Webproxy SSL * OTOBO_NGINX_SSL_CERTIFICATE = / etc / nginx / ssl / acme.crt *

otobo_nginx_ssl_certificate_key Nginx Webproxy SSL * OTOBO_NGINX_SSL_CERTIFICATE_KEY = / etc / nginx / ssl / acme.key *

Kerberos *Nginx *

Kerberos Nginx

otobo_nginx_kerberos_keytab Kerberos keytab /etc/krb5.keytab

otobo_nginx_kerberos_config Kerberos /etc/krb5.conf krb5.conf.template

otobo_nginx_kerberos_service_name Kerberos

otobo_nginx_kerberos_realm Kerberos REALM /etc/krb5.conf

otobo_nginx_kerberos_kdc Kerberos kdc / AD /etc/krb5.conf

otobo_nginx_kerberos_admin_server Kerberos /etc/krb5.conf

otobo_nginx_kerberos_default_domain Kerberos /etc/krb5.conf

nginx_envsubst_template_dir Nginx

docker-compose

docker-compose

otobo_nginx_1 otopo_nginx_1 otopo_web_1
otobo_db_1 OTOBO

COMPOSE_FILE

COMPOSE_FILE * docker-compose / otopo-base.yml docker-compose / otopo-override-http.yml docker-compose / otopo-override-https.yml *

otobo_image_otobo, otopo_image_otobo_elasticsearch, otopo_image_otobo_nginx, ... Docker

4.4

4.4.1 Nginx

otobo_nginx_1 Nginx HTTPS Docker Nginx Docker https://hub.docker.com/_/nginx Nginx OTOBO

OTOBO /etc/nginx/template/otobo_nginx.conf.template Docker Nginx

otobo_nginx_ssl_certificate SSL

otobo_nginx_ssl_certificate_key SSL

otobo_nginx_web_host HTTP

otobo_nginx_web_port 80 HTTP 80

80 4.8000000000000000 SSL 80

Warning: OTOBO 10.0.4

otobo_nginx_1

```
# stop the possibly running containers
docker_admin> cd /opt/otobo-docker
docker_admin> docker-compose down

# create a volume that is initially not connected to otopo_nginx_1
docker_admin> docker volume create otopo_nginx_custom_config

# find out where the new volume is located on the Docker host
docker_admin> otopo_nginx_custom_config_mp=$(docker volume inspect --format '{{ .
↳Mountpoint }}' otopo_nginx_custom_config)
docker_admin> echo $otopo_nginx_custom_config_mp # just a sanity check
docker_admin> ls $otopo_nginx_custom_config_mp # another sanity check

# copy the default config into the new volume
docker_admin> docker create --name tmp-nginx-container rotheross/otobo-nginx-
↳webproxy:latest-10_0 # or latest-10_1, use the appropriate label
docker_admin> docker cp tmp-nginx-container:/etc/nginx/templates/otobo_nginx.conf.
↳template $otopo_nginx_custom_config_mp # might need 'sudo'
docker_admin> ls -l $otopo_nginx_custom_config_mp/otobo_nginx.conf.template # just
↳checking, might need 'sudo'
docker_admin> docker rm tmp-nginx-container

# adapt the file $otopo_nginx_custom_config_mp/otobo_nginx.conf.template to your needs
docker_admin> vim $otopo_nginx_custom_config_mp/otobo_nginx.conf.template
```

Warning: Nginx listen Web OTOBO 10.0.3 OTOBO 10.0.4 Nginx 10.0.3 80 443 OTOBO 10.0.4 8080 8443

docker-compose/otobo-nginx-custom-config.yml
COMPOSE_FILE** Nginx NGINX_ENVSUBST_TEMPLATE_DIR**

```
COMPOSE_FILE=docker-compose/otobo-base.yml:docker-compose/otobo-override-https.
↳yml:docker-compose/otobo-nginx-custom-config.yml
NGINX_ENVSUBST_TEMPLATE_DIR=/etc/nginx/config/template-custom
```

Docker Compose

```
docker_admin> docker-compose config | more
```

```
docker_admin> docker-compose up --detach
```

`https://hub.docker.com/_/nginx` “Nginx 1.19”

4.4.2 Nginx Kerberos

Kerberos .env `docker-compose/otobo-override-https-kerberos.yml` Docker compose Kerberos Nginx

.env `https://github.com/RotherOSS/otobo/blob/rel-10_1/scripts/nginx/kerberos/templates/krb5.conf.template` /etc/krb5.conf

/etc/krb5.conf /etc/krb5.conf .env OTOBO_NGINX_KERBEROS_CONFIG docker-compose/otobo-override-https-kerberos.yml

`*/etc/krb5.keytab*`

`** Kerberos SSO **`

`so-kerberos`

4.4.3

44380 HTTPS HTTP .env `OTOBO_WEB_HTTP_PORT` `OTOBO_WEB_HTTPS_PORT`

4.4.4

Docker compose HTTPS Docker compose `-scale`

```
docker_admin> docker-compose up --detach --scale db=0
```

`docker-compose/otobo-base.yml`

4.4.5

docker `otobo-docker <https://doc.otobo.org/manual/installation/10.1/en/content/installation-docker.html#clone-the-otobo-docker-repo>` `otobo-docker/docker-compose`

```
cd otobo-docker/docker-compose
```

Docker `otobo-base.yml`

```
for i in $(cat otobo-base.yml | grep image: | cut -d":" -f3,4 | sed -e "s-//1" -e"s/\}/<br>↵/g"); do docker pull $i; docker save $i -o $(echo $i | sed "s/\//-/g").docker; done
```

docker-compose docker-compose <https://en.wikipedia.org/wiki/Secure_copy_protocol>’__

docker

mariadb

```
docker load --input mariadb:10.5.docker
```

4.4.6 OTOBO Docker Compose

docker-compose/ otobo-docker ” <<https://doc.otobo.org/manual/installation/10.1/en/content/updating-docker.html#updating-the-docker-compose-files>>’_”

YAML

3306 docker compose

```
$ cat custom_db.yml
services:
  db:
    ports:
      - "0.0.0.0:3306:3306"
```

docker-compose YAML .env COMPOSE_FILE

```
COMPOSE_FILE=docker-compose/otobo-base.yml:docker-compose/otobo-override-http.
yml:custom_db.yml
```

docker-compose

```
$ docker-compose stop # if otobo is running
$ docker-compose up -d
```

4.4.7 OTOBO Docker

otobo_opt_otobo Docker /opt/otobo Perl /opt/otobo/local CPAN "Acme::123"

```
$ docker exec -it ${COMPOSE_PROJECT_NAME:=otobo}_web_1 bash
otobo@ce36ff89e637:~$ pwd
/opt/otobo
otobo@ce36ff89e637:~$ cpanm -l local Acme::123
--> Working on Acme::123
Fetching http://www.cpan.org/authors/id/N/NA/NATHANM/Acme-123-0.04.zip ... OK
Configuring Acme-123-0.04 ... OK
Building and testing Acme-123-0.04 ... OK
Successfully installed Acme-123-0.04
1 distribution installed
otobo@ce36ff89e637:~$
```

Docker

Debian Dockerfile OTOBO "docker commit "https://docs.docker.com/engine/reference/commandline/commit/"https://phoenixnap.com/kb/how-to-commit-changes-to-docker-image"

```

otobo UID 1000 otobo otobo
root
/opt/otobo_install/entrypoint.sh root
otobo
OTOBO

```

```

$ docker run rotheross/otobo:rel-10_0_10 /usr/games/fortune
/opt/otobo_install/entrypoint.sh: line 57: /usr/games/fortune: No such file or
↳directory

```

OTOBO root

```

$ docker run -it --user root --entrypoint /bin/bash --name otobo_orig rotheross/
↳otobo:rel-10_0_10
root@50ac203409eb:/opt/otobo# apt update
root@50ac203409eb:/opt/otobo# apt install fortunes
root@50ac203409eb:/opt/otobo# exit
$ docker ps -a | head

```

OTOBO

```

$ docker commit -c 'USER otobo' -c 'ENTRYPOINT ["/opt/otobo_install/entrypoint.sh"]'
↳otobo_orig otobo_with_fortune_cookies

```

OTOBO

```

$ docker run otobo_with_fortune_cookies /usr/games/fortune
A platitude is simply a truth repeated till people get tired of hearing it.
-- Stanley Baldwin

```

.env

4.4.8

Note: Docker

Docker git <https://github.com/RotherOSS/otobo>

- otobo.web.dockerfile
- otobo.nginx.dockerfile
- otobo.elasticsearch.dockerfile

bin/docker/build_docker_images.sh

```

docker_admin> cd /opt
docker_admin> git clone https://github.com/RotherOSS/otobo.git
docker_admin> # checkout the wanted branch. e.g. git checkout rel-10_0_11
docker_admin> cd otobo
docker_admin> # modify the docker files if necessary
docker_admin> bin/docker/build_docker_images.sh
docker_admin> docker image ls

```

Docker RELEASE local-<OTOBO_VERSION>

- docker-compose exec nginx nginx -s reload“ `nginx`

4.5 `nginx`

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXX

- Perl Maven Docker `Perl`
- Dockerfile `nginx`
- `nginx`

XXXXXXXXXX

- Ubuntu 18.04 LTS Docker Compose `nginx`
- Ubuntu 18.04 LTS Docker `nginx`
- `nginx` <<https://forums.docker.com/t/command-to-remove-all-unused-images/>>_
- Docker `IP`
- ‘`nginx`<<https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-nginx-in-ubuntu-18-04>>’_

XXXXXX

- ‘Docker `nginx`<<https://stackoverflow.com/questions/34814669/when-does-docker-image-cache-invalidation-occur>>’_
- `tcpdump` <<https://rmoff.net/2019/11/29/using-tcpdump-with-docker/>>’_
- `nginx`

Warning: Apache mod_perl OTRS OTOBO Apache OTRS a2query -s /etc/apache2/sites-available /etc/apache2/sites-enabled

```
*root@localhost# OTOBO Admin -> Packages OTOBO OPM
OPMOTRS“”OTOB
```

- OTRSHideShowDynamicField
- RotherOSSHideShowDynamicField
- TicketForms
- RotherOSS-LongEscalationPerformanceBoost
- Znuny4OTRS - AdvancedDynamicFields
- Znuny4OTRS-AutoSelect
- Znuny4OTRS-EscalationSuspend
- OTRSEscalationSuspend
- OTRSDynamicFieldDatabase
- OTRSDynamicFieldWebService
- OTRSBruteForceAttackProtection
- Znuny4OTRS-ExternalURLJump
- Znuny4OTRS-QuickClose
- Znuny4OTRS-AutoCheckbox
- OTRSSystemConfigurationHistory
- Znuny4OTRS-PasswordPolicy

The following OTOBO packages have been integrated into OTOBO 11.0. This means that they should not be installed in the target system when the target system is OTOBO 11.

- ImportExport

5.4 2 OTOBO ”

```
OTOBO OTOBO Admin -> System Configuration SecureMode
```

Note:

5.5 3 OTOBO

```
OTOBO Docker Docker
```

```
Docker otobo
```



```

docker_admin> # when docker_admin is root
docker_admin> rsync --recursive --safe-links --owner --group --chown 1000:1000 --
↳perms --chmod "a-wx,Fu+r,Du+rx" /opt/otrs/ $otobo_opt_otobo_mp/var/tmp/copied_otrs
docker_admin> ls -la $otobo_opt_otobo_mp/var/tmp/copied_otrs # just a sanity check

docker_admin> # when docker_admin is not root
docker_admin> sudo rsync --recursive --safe-links --owner --group --chown 1000:1000 --
↳perms --chmod "a-wx,Fu+r,Du+rx" /opt/otrs/ $otobo_opt_otobo_mp/var/tmp/copied_otrs
docker_admin> sudo ls -la $otobo_opt_otobo_mp/var/tmp/copied_otrs # just a sanity_
↳check

```

***** / opt / otobo / tmp / otrs *

5.10 5

http://localhost/otobo/migration.pl OTOBO "local-host"

Warning: *****

```

# native installation
root> service apache2 restart

# Docker-based installation
docker_admin> cd /opt/otobo-docker
docker_admin> docker-compose restart web
docker_admin> docker-compose ps # otobo_web_1 should be running again

```

Note: OTOBO Docker localhost OTRS /opt/otobo/var/tmp/copied_otrs OTRS *****

Note: OTRS OTRS Kernel/Config.pm otobo_db_1 Docker *****

Note: Docker Docker Docker 127.0.0.1 '127.0.0.1' OTRS OTRS Server 'hostname -all-ip-addresses' IP *****

Note: Docker otobo CREATE USER 'otrs_migration'@'%' IDENTIFIED BY 'otrs_migration'; `` ``GRANT SELECT, SHOW VIEW ON otrs.* TO 'otrs_migration'@'%'; DROP USER 'otrs_migration'@'%';

Kernel/Config.pm OTRS OTOBO /opt/otobo/Kernel/Config.pm LDAP *****

5.12.5 5. 5.12.5

OTOBOTO OTRSTicket::Frontend::AgentTicketQuickClose####State”OTOBOTO10“””**”**”

OTOBOTO

- “bin/otobo.Console.pl Admin::Config::ListInvalid“
- bin/otobo.Console.pl Admin::Config::FixInvalid
- bin/otobo.Console.pl Maint::Config::Rebuild migration.pl SecureMode

migration.pl

5.13 7

5.13.1 1. Password policy rules

OTOBOTO10“ PreferencesGroups ### Password”“ CustomerPersonalPreference #### Password”

“ PasswordMinSize “	8
“ PasswordMin2Lower2UpperCharacters “	
“ PasswordNeedDigit “	
“ PasswordHistory “	10
“ PasswordTTL “	30
“ PasswordWarnBeforeExpiry “	5
“ PasswordChangeAfterFirstLogin “	

5.13.2 2. Docker cron

OTOBOTO Docker cron Docker cron Docker cron OTRSTicket::Frontend::AgentTicketQuickClose####State”OTOBOTO10“””**”**”

5.14

5.14.1 Oracle Oracle

OracleETL Oracle

OTOBOTO Oracle Perl DBD::Oracle

Note: Oracle SDK DBD::Oracle

expdb impdbData Pump

Note: <https://github.com/bschmalhofer/otobo-ideas/blob/master/oracle.md>

1. `otobo`

`otobo` `otobo`

```
-- in the OTOBO database
DROP USER otobo CASCADE
```

2. `OTRS`

```
mkdir /tmp/otrs_dump_dir
```

```
-- in the OTRS database
CREATE DIRECTORY OTRS_DUMP_DIR AS '/tmp/otrs_dump_dir';
GRANT READ, WRITE ON DIRECTORY OTRS_DUMP_DIR TO sys;
```

```
expdp \"/sys/Oradoc_db1@//127.0.0.1/orclpdb1.localdomain as sysdba\" schemas=otrs_
↳directory=OTRS_DUMP_DIR dumpfile=otrs.dmp logfile=expdpotrs.log
```

3. `OTRS` "otobo"

```
impdp \"/sys/Oradoc_db1@//127.0.0.1/orclpdb1.localdomain as sysdba\" directory=OTRS_
↳DUMP_DIR dumpfile=otrs.dmp logfile=impdpotobo.log remap_schema=otrs:otobo
```

```
-- in the OTOBO database
-- double check
select owner, table_name from all_tables where table_name like 'ARTICLE_DATA_OT%_CHAT
↳';

-- optionally, set the password for the user otobo
ALTER USER otobo IDENTIFIED BY XXXXXX;
```

4. `otobo`

```
cd /opt/otobo
scripts/backup.pl --backup-type migratefromotrs # it's OK that the command knows only_
↳about the otobo database, only last line is relevant
sqlplus otobo/otobo@//127.0.0.1/orclpdb1.localdomain < /home/bernhard/devel/OTOBO/
↳otobo/2021-03-31_13-36-55/orclpdb1.localdomain_post.sql >sqlplus.out 2>&1
double check with `select owner, table_name from all_tables where table_name like
↳'ARTICLE_DATA_OT%_CHAT';
```

5. `otobo`

6. `5 migration.pl`

Note: `10.1` `OTOBO` `/opt/otobo/scripts/DBUpdate-to-10.1.pl` `stats_report` `data_storage`

5.14.2 数据库迁移

本节描述了如何从 OTRS 数据库迁移到 OTOBO 数据库。OTRS 数据库迁移到 OTOBO 数据库。

Note: 使用 Docker 安装 OTOBO。

Note: OTRS 使用 MySQL 数据库。

OTRS 数据库迁移到 OTOBO 数据库。

- 数据库字符集 utf8mb4
- 数据库引擎 InnoDB
- 数据库表名使用小写字母

OTRS 数据库迁移到 OTOBO 数据库。使用以下 SQL 脚本。

mysql -u root -p < /opt/otobo/bin/backup.pl * --db-name otrs --db-host=127.0.0.1 --db-user otrs --db-password "secret_otrs_password"

Warning: 使用 Docker 安装 OTOBO 时，请确保 /opt/otrs 目录存在。

```

otobo> cd /opt/otobo
otobo> scripts/backup.pl -t migratefromotrs --db-name otrs --db-host=127.0.0.1 --db-
↪user otrs --db-password "secret_otrs_password"

```

Note: 使用 Docker 安装 OTOBO 时，请确保 /opt/otobo 目录存在。

使用 bin/backup.pl 脚本迁移 OTRS 数据库。2021-04-13_12-13-04 数据库 SQL 脚本使用 mysql 命令。

```

otobo> cd <dump_dir>
otobo> mysql -u root -p<root_secret> otobo < otrs_pre.sql
otobo> mysql -u root -p<root_secret> otobo < otrs_schema_for_otobo.sql
otobo> mysql -u root -p<root_secret> otobo < otrs_data.sql
otobo> mysql -u root -p<root_secret> otobo < otrs_post.sql

```

使用 Docker 安装 OTOBO。

使用 Docker 安装 db 容器，使用 mysql 命令迁移 OTRS 数据库。root 用户 Docker 容器 .env 文件。

```

docker_admin> cd /opt/otobo-docker
docker_admin> docker-compose exec -T db mysql -u root -p<root_secret> otobo < /opt/
↪otobo/<dump_dir>/otrs_pre.sql
docker_admin> docker-compose exec -T db mysql -u root -p<root_secret> otobo < /opt/
↪otobo/<dump_dir>/otrs_schema_for_otobo.sql
docker_admin> docker-compose exec -T db mysql -u root -p<root_secret> otobo < /opt/
↪otobo/<dump_dir>/otrs_data.sql
docker_admin> docker-compose exec -T db mysql -u root -p<root_secret> otobo < /opt/
↪otobo/<dump_dir>/otrs_post.sql

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

```
otobo> mysql -u root -p<root_secret> -e 'SHOW DATABASES'
otobo> mysql -u root -p<root_secret> otobo -e 'SHOW TABLES'
otobo> mysql -u root -p<root_secret> otobo -e 'SHOW CREATE TABLE ticket'
```

XX Docker XXXX

```
docker_admin> docker-compose exec -T db mysql -u root -p<root_secret> -e 'SHOW
↳DATABASES'
docker_admin> docker-compose exec -T db mysql -u root -p<root_secret> otobo -e 'SHOW
↳TABLES'
docker_admin> docker-compose exec -T db mysql -u root -p<root_secret> otobo -e 'SHOW
↳CREATE TABLE ticket'
```

XXXXXXXXXXXXXXXXXXXXXXXXXXXX


```
root> su - otobo
otobo> cd /opt/otobo/
otobo> bin/Cron.sh stop
otobo> bin/otobo.Daemon.pl stop
```

6.2 2

“/opt/otobo”

6.2.1 Ubuntu MySQL

```
root> mkdir /root/otobo-update # Create a update directory
root> cd /root/otobo-update # Change into the update directory
root> cp -pr /opt/otobo otobo-prod-old # Backup the hole OTOBO directory
↳ to the update directory
root> mysqldump -u otobo -p otobo -r otobo-prod-old.sql # Backup the otobo database
↳ to otobo-prod-old.sql
```

Warning: “backup-restore”

6.3 3

https://ftp.otobo.org/pub/otobo/OTOBO“tar”“/root/otobo-update”

```
root> cd /root/otobo-update # Change into
↳ the update directory
root> wget https://ftp.otobo.org/pub/otobo/otobo-latest-10.1.tar.gz # Download he
↳ latest OTOBO 10.1 release
root> tar -xzf otobo-latest-10.1.tar.gz # Unzip OTOBO
root> cp -r otobo-10.1.x/* /opt/otobo # Copy the
↳ new otobo directory to /opt/otobo
```

6.3.1

OTOBO 10“Kernel/Config.pm”

```
root> cd /root/otobo-update
root> cp -p otobo-prod-old/Kernel/Config.pm /opt/otobo/Kernel/
root> cp -p otobo-prod-old/var/cron/* /opt/otobo/var/cron/
```

6.3.2

OTOBO“article” “/ opt / otobo / var /”


```
root> cd /root/otobo-update
root> cp -pr otobo-prod-old/var/article/* /opt/otobo/var/article/
```

6.3.3 複製檔案

複製檔案至 `*/opt/otobo/var/stats`

```
root> cd /root/otobo-update/otobo-prod-old/var/stats
root> cp *.installed /opt/otobo/var/stats
```

6.3.4 設定權限

設定 OTOBO 安裝目錄的權限

```
root> /opt/otobo/bin/otobo.SetPermissions.pl
```

6.3.5 設定 Apache 設定

將 OTOBO 安裝目錄的 `apache` 目錄下的 `10.1` 目錄下的 `CGI` 目錄下的 `PSGI` 目錄下的 `scripts/apache2-httpd-vhost-443.include.conf` 檔案複製到 `*/opt/otobo`

6.4 安裝 4 個 Perl 模組

OTOBO 安裝目錄的 `cpan` 目錄下的 `*/opt/otobo`

Note: Debian 系統安裝 Perl 模組

```
apt-get install -y libarchive-zip-perl libtimedate-perl libdatettime-perl libconvert-
↳binhex-perl libcgi-psgi-perl libdbi-perl libdbix-connector-perl libfile-chmod-perl
↳liblist-allutils-perl libmoo-perl libnamespace-autoclean-perl libnet-dns-perl
↳libnet-smtp-ssl-perl libpath-class-perl libsub-exporter-perl libtemplate-perl
↳libtemplate-perl libtext-trim-perl libtry-tiny-perl libxml-libxml-perl libyaml-
↳libyaml-perl libdbd-mysql-perl libapache2-mod-perl2 libmail-imapclient-perl
↳libauthen-sasl-perl libauthen-ntlm-perl libjson-xs-perl libtext-csv-xs-perl libpath-
↳class-perl libplack-perl libplack-middleware-header-perl libplack-perl libplack-
↳middleware-reverseproxy-perl libencode-hanextra-perl libio-socket-ssl-perl libnet-
↳ldap-perl libcrypt-eksblowfish-perl libxml-libxslt-perl libxml-parser-perl libconst-
↳fast-perl
```

```
root> su - otobo
otobo> perl /opt/otobo/bin/otobo.CheckModules.pl --list
```

6.5 安裝 5 個 Perl 模組

OTOBO 安裝目錄的 `cpan` 目錄下的 `*/opt/otobo`

 Updating Docker OTOBO 10

Docker OTOBO OTOBO OTOBO OTOBO OTOBO Docker OTOBO otobo_opt_otobo

OTOBO OTOBO

- Docker Compose
- Docker Compose
- Docker
- `docker volume inspect otobo_opt_otobo`
-

Note: In the sample commands below, the version **11.x.y**, corresponding to the tag **11_x_y**, is used as the example version. Please substitute it with the real version, e.g. **11.0.7**.

7.1 Updating the Docker Compose files

OTOBO Docker Compose

Note: <https://hub.docker.com/r/rotheross/otobo/tags>

```
# Change to the otobo docker directory
docker_admin> cd /opt/otobo-docker

# Get the latest tags
docker-admin> git fetch --tags
```

```
# Update OTOBO docker-compose repository to version 11.x.y.
docker-admin> git checkout rel-11_x_y
```

7.2 Docker Compose .env

```
.env OTOBO Docker OTOBO_IMAGE_OTOBO_OTOBO_IMAGE_OTOBO_ELASTICSEARCH
*OTOBO_IMAGE_OTOBO_NGINX*
```

7.3 Docker

Docker compose <https://hub.docker.com/r/rotheross/otobo/>

```
# Change to the otobo docker directory
docker_admin> cd /opt/otobo-docker

# fetch the new images, either 'latest-11_0', 'latest-10_1', 'latest-10_0' or the
↳specific version declared in .env
docker_admin> docker-compose pull
```

7.4 OTOBO

Warning: Please note that minor or major upgrades must always be carried out one after the other. If you would like to upgrade from version 10.0.* to the latest 11.0.*, please upgrade to 10.1 first and then to 11.0.

otobo_opt_otobo OTOBO

- Admin::Package::ReinstallAll
- ::::
- ::::
- ::::

```
# stop and remove the containers, but keep the named volumes
docker_admin> docker-compose down

# copy the OTOBO software, while containers are still stopped
docker_admin> docker-compose run --no-deps --rm web copy_otobo_next

# start containers again, using the new version and the updated /opt/otobo
docker_admin> docker-compose up --detach

# a quick sanity check
docker_admin> docker-compose ps

# complete the update, with running database
docker_admin> docker-compose exec web /opt/otobo_install/entrypoint.sh do_update_tasks
```

```
# inspect the update log
docker_admin> docker-compose exec web cat /opt/otobo/var/log/update.log

**# For minor or major release upgrades, you also have to run the upgrade script (for
↳example to upgrade from 10.1 to 11.0)**
root> docker exec -it otobo_web_1 perl scripts/DBUpdate-to-11.0.pl
```

Note: OTOBO 10.8 scripts/update.sh docker-compose pull

```
docker_admin> ./scripts/update.sh --help
docker_admin> ./scripts/update.sh

** For minor or major release upgrades, you also have to run the upgrade script (for
↳example to upgrade from 10.1 to 11.0)**
docker_admin> docker exec -it otobo_web_1 perl scripts/DBUpdate-to-11.0.pl
```


□□□□

OTOBOScript "otobo" help

8.1 □□

Note: OTOBO "otobo" help

```
otobo> /opt/otobo/scripts/backup.pl -h
```

□□□□□□□□

Backup an OTOBO system.

Usage:

```
backup.pl -d /data_backup_dir [-c gzip|bzip2] [-r DAYS] [-t
↪fullbackup|nofullbackup|dbonly]
backup.pl --backup-dir /data_backup_dir [--compress gzip|bzip2] [--remove-old-
↪backups DAYS] [--backup-type fullbackup|nofullbackup|dbonly]
```

Short options:

```
[-h]           - Display help for this command.
-d            - Directory where the backup files should place to.
[-c]         - Select the compression method (gzip|bzip2). Default: gzip.
[-r DAYS]    - Remove backups which are more than DAYS days old.
[-t]         - Specify which data will be saved
↪(fullbackup|nofullbackup|dbonly). Default: fullbackup.
```

Long options:

```
[--help]           - same as -h
--backup-dir       - same as -d
[--compress]      - same as -c
```



```
# create the backup directory on the host
docker_admin> mkdir otobo_backup

# create the Docker volume
docker_admin> docker volume create --name otobo_backup --opt type=none --opt device=
↳$PWD/otobo_backup --opt o=bind

# inspect the volume out of curiosity
docker_admin> docker volume inspect otobo_backup
```

~~~~~“otobo\_opt\_otobo“ ~ otobo\_backup~~~~Daemon~~~~~

```
# create a backup
docker_admin> docker run -it --rm --volume otobo_opt_otobo:/opt/otobo --volume otobo_
↳backup:/otobo_backup --network otobo_default rotheross/otobo:latest-10_0 scripts/
↳backup.pl -d /otobo_backup

# check the backup file
docker_admin> tree otobo_backup
```

~~~~~“<TIMESTAMP>“ ~~~~“2020-09-07\_09-38“~~~~~

```
# create a backup
docker_admin> docker run -it --rm --volume otobo_opt_otobo:/opt/otobo --volume otobo_
↳backup:/otobo_backup --network otobo_default rotheross/otobo:latest-10_0 scripts/
↳restore.pl -d /opt/otobo -b /otobo_backup/<TIMESTAMP>
```



```
# check the backup file
docker_admin>tree otobo_backup

.. note::

  --extra-dump-options="--single-transaction" prevents the database tables from being
  ↳locked, so OTOBO can still be used during the backup.
```

Note: `otobo` database

otobo database db MySQL CLI

MySQL otobo

```
mysql@4f7783595190:/$>DROP DATABASE otobo;
mysql@4f7783595190:/$>CREATE DATABASE otobo CHARACTER SET utf8mb4 COLLATE utf8mb4_
↳unicode_ci;
mysql@4f7783595190:/$>GRANT ALL PRIVILEGES ON otobo.* TO 'otobo'@'%';
```

`<TIMESTAMP>` "2020-09-07_09-38"

```
# restore a backup
docker_admin>docker run -it --rm --volume otobo_opt_otobo:/opt/otobo --volume otobo_
↳backup:/otobo_backup --network otobo_default rotheross/otobo:latest-10_0 scripts/
↳restore.pl -d /opt/otobo -b /otobo_backup/<TIMESTAMP>
```

OTOBO Docker Kerberos

Docker Docker Compose OTOBO Docker OTOBO

Note: AD LDAP Kerberos

10.1

Active Directory

Note: HTTP/fqdn.from.your.otobo.de”fqdn.from.your.otobo.de” A-Record DNS CNAME OTOBO URL CNAME A-Record

”HTTP/” Kerberos

”&”

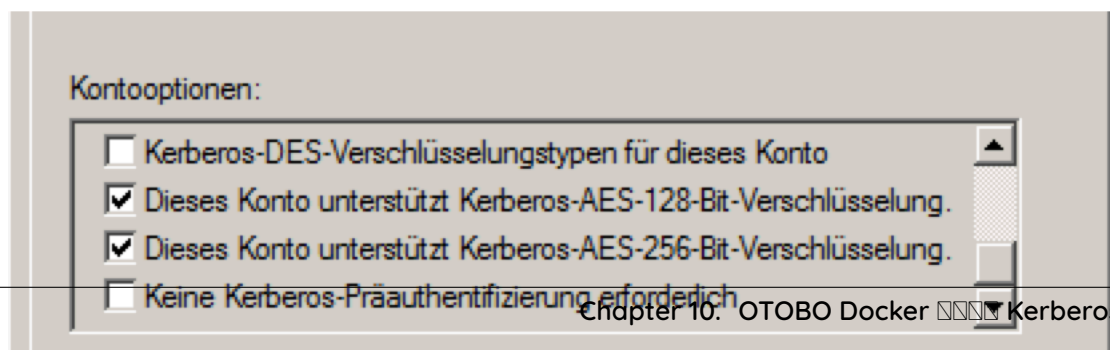
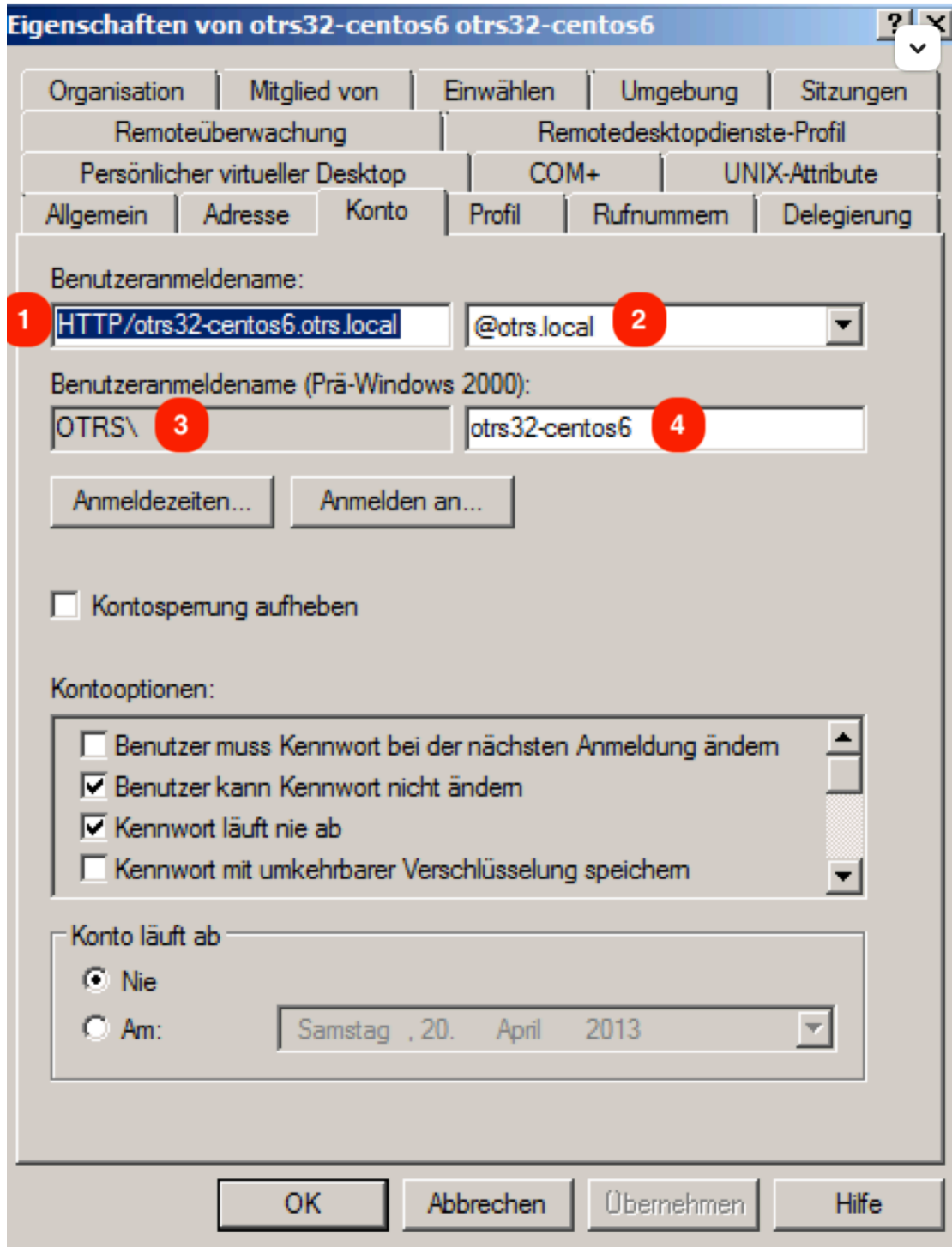
AD LDAP/AD

10.2 Active Directory Keytab

Active Directory (cmd) ktpass.exe

```
ktpass.exe -princ HTTP/otrs32-centos6.otrs.local@OTRS.LOCAL -mapuser OTRS\otrs32-  
centos6 -crypto All -pass Password -ptype KRB5_NT_PRINCIPAL -out c:\krb5.keytab
```

- -princ = HTTP/otrs32-centos6.otrs.local@OTRS.LOCAL -> 1+@+ 2'
- -mapuser = OTRSotrs32-centos6 (Username prä Win 2000) -> -> Picture Number 3++Picture Number.



- `-pass = $(cat /dev/urandom | tr -dc 'a-z0-9' | fold -w 32 | xargs echo) | sed 's/ /./g'`
- `-out = c:/krb5.keytab`

Note: `OTRS_LOCAL=@OTRS.LOCAL`

Configure `krb5.keytab` for OTOBO

```
# Create new directory
docker_admin> mkdir /opt/otobo-docker/nginx-conf

# Move the file krb5.keytab to the new directory (Attention, depending on where you
↳ have placed the krb5.conf file, the command below will change.)
docker_admin> mv ?/krb5.keytab /opt/otobo-docker/nginx-conf/krb5.keytab
```

10.3 Configure nginx

```
docker volume create otobo_nginx_custom_config
otobo_nginx_custom_config_mp=$(docker volume inspect --format '{{ .Mountpoint }}'
↳ otobo_nginx_custom_config)
docker create --name tmp-nginx-container rotheross/otobo-nginx-webproxy:latest-10_1
↳ (achtung: Versionsnummer)
docker cp tmp-nginx-container:/etc/nginx/templates /tmp
docker cp tmp-nginx-container:/etc/nginx/templates/otobo_nginx-kerberos.conf.template.
↳ hidden $otobo_nginx_custom_config_mp/otobo_nginx.conf.template
docker rm tmp-nginx-container
vim docker-compose/otobo-nginx-custom-config.yml
```

```
COMPOSE_FILE =>
docker-compose/otobo-nginx-custom-config.yml
NGINX_ENVSUBST_TEMPLATE_DIR=/etc/nginx/config/template-custom
```

10.4 Configure OTOBO .env

Configure `/opt/otobo-docker/.env` and `.env.tmp` for kerberos

```
# Stop OTOBO Container if running
docker_admin>cd /opt/otobo-docker
docker_admin>docker-compose down

# create a backup of the old .env file
docker_admin>mv /opt/otobo-docker/.env /opt/otobo-docker/.env.tmp

# create a new backupfile including kerberos settings
docker_admin>cp /opt/otobo-docker/.docker_compose_env_https_kerberos /opt/otobo-
↳ docker/.env
```

Configure `.env` for OTOBO_DB_ROOT_PASSWORD, OTOBO_NGINX_SSL_CERTIFICATE, OTOBO_NGINX_Kerberos

```
# Kerberos OTOBO_NGINX_KERBEROS_KEYTAB=/opt/otobo-docker/nginx-conf/krb5.keytab
```

```
# /etc/krb5.conf # OTOBO_NGINX_KERBEROS_CONFIG=/opt/otobo-docker/nginx-conf/krb5.conf
# Kerberos OTOBO_NGINX_KERBEROS_SERVICE_NAME=HTTP/otrs32-centos6.otrs.local # ->
1
# Kerberos REALM OTOBO_NGINX_KERBEROS_REALM=ROTHER-OSS.COM -> OTRS.LOCAL # ->
2
# /Kerberos kdc OTOBO_NGINX_KERBEROS_KDC=
# /Kerberos OTOBO_NGINX_KERBEROS_ADMIN_SERVER=rother-oss.com
# Kerberos OTOBO_NGINX_KERBEROS_DEFAULT_DOMAIN=otrs.local
```

10.5 OTOBO

Kerberos OTOBO

```
# Start OTOBO using docker-compose
docker_admin> docker-compose up -d
```

10.6 OTOBO Kerberos

AD Kernel/Config.pm LDAP
 Kerberos Kernel/Config/Defaults.pm Kerberos Kernel/Config.pm

```
$Self->{AuthModule} = 'Kernel::System::Auth::HTTPBasicAuth';
# In case you need to replace some part of the REMOTE_USER, you can
# use the following RegExp ($1 will be new login).
$Self->{'AuthModule::HTTPBasicAuth::ReplaceRegExp'} = '^(.+?)@.+?$';
```

10.7 Kerberos SSO

SSO

Chrome Edge Internet Explorer

Windows Internet

firefox "about:config"

- network.negotiate-auth.trusted-uris = https:// https://otobofqdn
- network.negotiate-auth.delegation-uris = http:// https://otobofqdn

10.8 验证

验证 Kerberos SSO 配置 NGINX 容器

```
# Check Container
docker_admin> docker ps
```

验证 NGINX 容器

```
# Check NGINX logs
docker_admin> docker logs otobo_nginx_1 -f
```

验证 NGINX 容器 NGINX 配置

```
# Login to the NGINX Container
docker_admin> docker exec -it otobo_nginx_1 bash

# Now please check if the krb5.conf file exists with your needed values
nginx_root> cat /etc/krb5.conf

# Now please check if the krb5.keytab file exists
nginx_root> cat /etc/krb5.keytab

# If not, please quit from the container and copy the file again using docker
docker_admin> docker cp /opt/otobo-docker/nginx-conf/krb5.keytab otobo_nginx_1:/etc/
↪krb5.keytab
```

10.8.1 Kerberos 验证

```
# Login to the NGINX Container
docker_admin> docker exec -it otobo_nginx_1 bash
```

验证 Kerberos 配置

```
env KRB5_TRACE=/dev/stdout kvno HTTP/otrs32-centos6.otrs.local@OTRS.LOCAL
klist -e
```

```
kinit -VV -k -t /etc/krb5.keytab HTTP/otrs32-centos6.otrs.local@OTRS.LOCAL
```

验证 Kerberos SSO 配置 NGINX 容器 AD 配置 keytab 配置 AD 配置 Kernel/Config.pm 配置

验证 SSO 配置 * 验证 * 验证 Kerberos SSO 配置

OTOBOTO

OTOBOTO OTOBOTO

Note: OTOBOTO CI OTOBOTO.css OTOBOTO "Admin -> System Configuration" "AgentLoginLogo" "AgentLogo" OTOBOTO

11.1 OTOBOTO

OTOBOTO Admin -> System Configuration OTOBOTO

- OTOBOTO
- OTOBOTO "OTOBOTO" "OTOBOTO" -> OTOBOTO "OTOBOTO" OTOBOTO

11.2 OTOBOTO

OTOBOTO OTOBOTO SCP OTOBOTOWinSCP OTOBOTO /tmp/ OTOBOTO

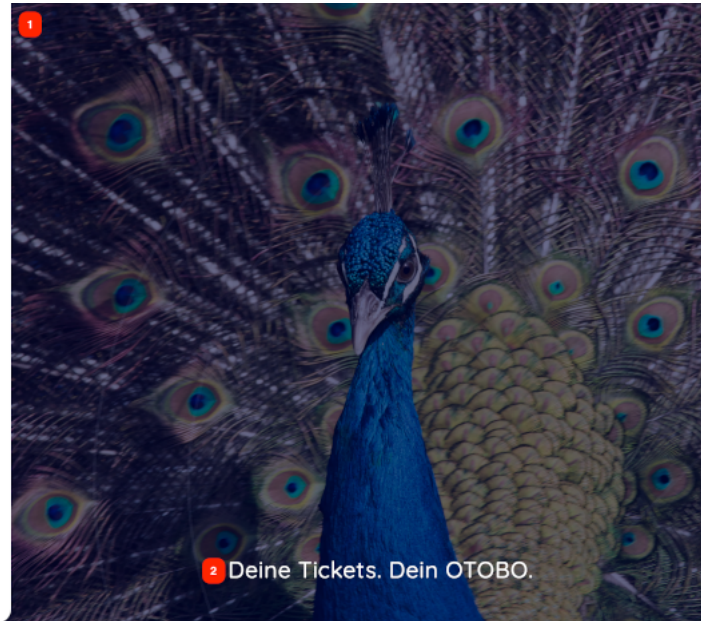
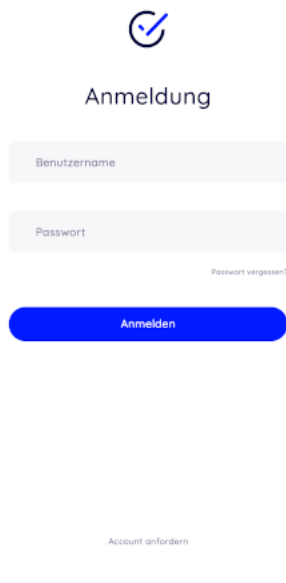
OTOBOTO Logo OTOBOTO OTOBOTO

```
**# Using OTOBOTO Docker Installation**
otobo_admin> docker cp /tmp/Logos.png otobo_web_1:/opt/otobo/var/httpd/htdocs/skins/
↳Customer/default/img/

**# Nativ installation in /opt/otobo/**
otobo_admin> cp /tmp/Logos.png /opt/otobo/var/httpd/htdocs/skins/Customer/default/img/
```

OTOBOTO Agent OTOBOTO "OTOBOTO" -> "OTOBOTO" OTOBOTO

11.2.1 背景画像を設定する



- 1 2 - 背景画像を設定する

背景画像を設定する

背景画像を設定する

背景画像を設定する**#oooLoginBG > .oooBG**

var/httpd/htdocs/skins/Custommer/default/css/Core.Login.css

```
#oooLoginBG > .oooBG {
    position: relative;
    width: 100%;
    height: 100%;
    /* opacity: 0.45; Disable opacity */
    background-size: cover;
    overflow: hidden;
}
```

背景画像を設定する

./Output/HTML/Templates/Standard/CustommerLogin.tt./Output/HTML/Templates/Standard/CustommerLogin.tt

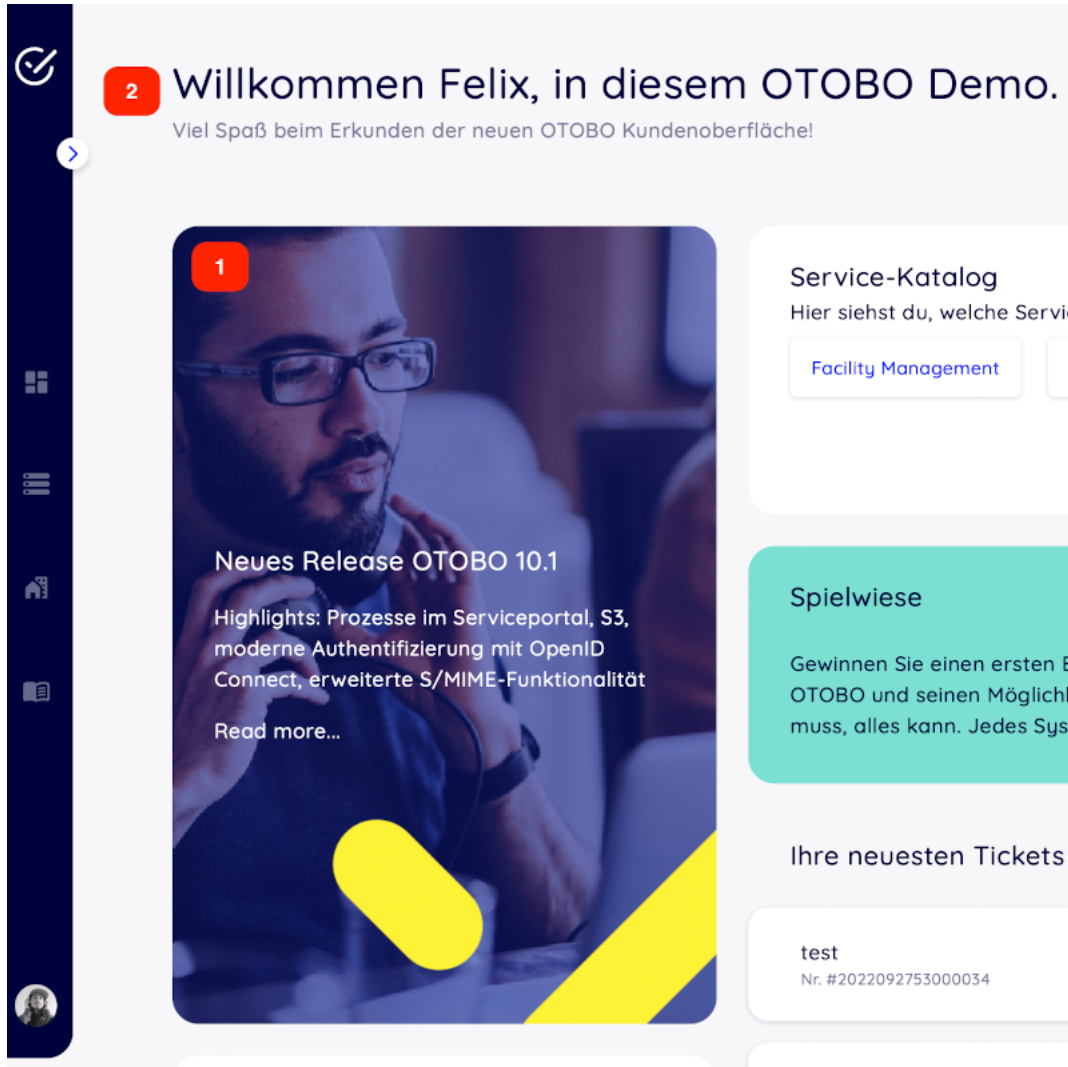
```
<!-- start login -->
<div id="oooLoginBG">
    <div class="oooBG" style="background-image: url([% Data.Background | html %]);">
# remove this line ->         <div id="oooBGSignet" style="background-image: url([%
->Config("Frontend::WebPath") %]common/img/otobo-signet_border.svg);"></div>
    </div>
    <h1>[% Translate(Data.LoginText) | html %]</h1>
</div>
```



Note: 背景画像を設定する opm 背景画像を設定する <https://doc.otobo.org/manual/developer/10.1/en/content/how-to-publish-otobo-extensions.html>

11.2.2

 "1" ->  "2"





- 1 -  CustomerDashboard::Tiles##FeaturedLink-01
- 2 -  CustomerDashboard::Configuration::Text

Note: 

CHAPTER 12

CPAN Perl

Perl CPAN <https://metacpan.org/>
cpanm cpanm <https://metacpan.org/pod/App::cpanminus>
Perl
“cpanm” root

```
root> cpanm Acme::Dice
```

```
otobo> perldoc -l Acme::Dice  
/usr/local/share/perl/5.30.0/Acme/Dice.pm
```

12.1 Docker

Docker OTOBO Docker
otobo_opt_otobo /opt/otobo/local
/opt/otobo/local Perl PERL5LIB PATH
OTOBO Perl /opt/otobo
Perl install --local-lib **web**

```
# starting a bash session in the container web  
docker_admin> cd /opt/otobo-docker/  
docker_admin> docker-compose exec web bash  
otobo@6ef90ed0cd0:~$ pwd  
/opt/otobo  
  
# installing the sample module Acme::Dice  
otobo@6ef90ed0cd0:~$ cpanm --local-lib local Acme::Dice
```

```
--> Working on Acme::Dice
Fetching http://www.cpan.org/authors/id/B/BO/BOFTX/Acme-Dice-1.01.tar.gz ... OK
Configuring Acme-Dice-1.01 ... OK
Building and testing Acme-Dice-1.01 ... OK
Successfully installed Acme-Dice-1.01
1 distribution installed

# confirm the installation directory
otobo@6ef90ed0cd0:~$ perldoc -l Acme::Dice
/opt/otobo/local/lib/perl5/Acme/Dice.pm

# locally installed module is found because the environment is preset accordingly
otobo@6ef90ed0cd0:~$ echo $PERL5LIB
/opt/otobo_install/local/lib/perl5:/opt/otobo/local/lib/perl5
otobo@6ef90ed0cd0:~$ echo $PATH
/opt/otobo_install/local/bin:/opt/otobo/local/bin:/usr/local/sbin:/usr/local/bin:/usr/
↪sbin:/usr/bin:/sbin:/bin
```


13.4 設定

OTOBOTicket :: Article :: Backend :: MIMEBase :: ArticleStorage

Kernel::System::Ticket::Article::Backend::MIMEBase::ArticleStorageDB

OTOBOTicket :: Article :: Backend :: MIMEBase :: ArticleStorageDB

Note: OTOBOTicket :: Article :: Backend :: MIMEBase :: ArticleStorageDB

Kernel::System::Ticket::Article::Backend::MIMEBase::ArticleStorageFS

OTOBOTicket :: Article :: Backend :: MIMEBase :: ArticleStorageFS NFS SAN

Note: OTOBOTicket :: Article :: Backend :: MIMEBase :: ArticleStorageFS

OTOBOTicket :: Article :: Backend :: MIMEBase :: ArticleStorageFS

```
otobo> /opt/otobo/bin/otobo.Console.pl Admin::Article::StorageSwitch --target ArticleStorageFS
```

OTOBOTicket :: Article :: Backend :: MIMEBase :: ArticleStorageFS

Note: OTOBOTicket :: Article :: Backend :: MIMEBase :: ArticleStorageFS CPU / OTOBOTicket :: Article :: Backend :: MIMEBase :: ArticleStorageFS

OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends

13.5 設定

OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends

OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends

OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends

1. OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends

2. OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends

- OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends
- OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends
- OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends
- OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends
- OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends
- OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends
- OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends
- OTOBOTicket :: Article :: Backend :: MIMEBase :: CheckAllStorageBackends

Note: 5000

- 1.
2. *
- 3.

13.6

Redis Cache

13.6.1 Redis Cache

1. Redis

Redis OTOBO Redis <https://redis.io/topics/quickstart>

2. Redis Perl Redis::Fast

Redis Redis Redis :: Fast Redis 2 otobo.CheckModules.pl -list

```
otobo> /opt/otobo/bin/otobo.CheckModules.pl --all
```

3. Redis OTOBO

OTOBO SysConfig -> OTOBO

| Setting | Description | Default value |
|-------------------------------|-----------------------------|----------------|
| Cache::Redis###Server | Redis server URL | 127.0.0.1:6379 |
| Cache::Redis###DatabaseNumber | Number of logical database | 0 |
| Cache::Redis###RedisFast | Use or not Redis::Fast | 0 |
| Cache::Module | Activate Redis Cache Module | DB (use Redis) |

13.6.2

OTOBO /opt/otobo/var/tmp RAM

```
otobo> /opt/otobo/bin/otobo.Console.pl Maint::Session::DeleteAll
otobo> /opt/otobo/bin/otobo.Console.pl Maint::Cache::Delete
root> mount -o size=16G -t tmpfs none /opt/otobo/var/tmp
```

Note: / etc / fstab

