



OTRS Update and Installation Guide

发布 7.0

OTRS AG

2020 年 05 月 07 日

1	介绍	3
1.1	关于本手册	3
2	硬件和软件要求	5
2.1	硬件要求	6
2.2	软件要求	7
3	安装	9
3.1	准备：禁用 SELinux	9
3.2	第 1 步：解压并安装应用程序	10
3.3	第 2 步：安装其它程序和 Perl 模块	10
3.4	第 3 步：创建 OTRS 用户	10
3.5	第 4 步：激活默认配置文件	11
3.6	第 5 步：配置 Apache Web 服务器	11
3.7	第 6 步：设置文件权限	11
3.8	第 7 步：安装数据库	12
3.8.1	MySQL 或 MariaDB	12
3.8.2	PostgreSQL	13
3.8.3	完成数据库设置	13
3.9	第 8 步：设置 Elasticsearch 集群	14
3.10	第 9 步：启动 OTRS 守护进程和 Web 服务器	14
3.11	第 10 步：首次登录	15
3.12	第 11 步：设置 Systemd 文件	15
3.13	第 12 步：设置 Bash 自动补全(可选)	15
3.14	步骤 13：进一步信息	15
4	更新	17
4.1	第 1 步：停止所有相关服务和 OTRS 守护进程	17
4.2	第 2 步：备份文件和数据库	18
4.3	第 3 步：安装新版本	18
4.3.1	恢复原配置文件	18
4.3.2	恢复信件数据	19
4.3.3	恢复已安装的默认统计	19
4.3.4	设置文件权限	19
4.3.5	安装必需的程序和 Perl 模块	19
4.4	第 4 步：运行迁移脚本	19
4.5	第 5 步：更新已安装的软件包	20

4.6	第 6 步：启动服务	20
4.7	第 7 步：启动 OTRS 守护进程和 Web 服务器	20
4.8	第 8 步：手动迁移任务和更改	21
5	备份和恢复	23
5.1	备份	23
5.2	恢复	24
6	性能优化	25
6.1	工单索引模块	25
6.2	工单搜索索引	25
6.3	文件搜索	27
6.3.1	堆大小	27
6.3.2	磁盘分配	27
6.4	信件存储	28
6.5	归档工单	29
6.6	优化 Web 服务器	29
6.7	缓存	29
6.8	集群	30



此作品的版权归 OTRS AG 所有 (<https://otrs.com>)，德国 (法兰克福) 上乌瑟尔 Zimmersmühlenweg 路 11 号，61440。

根据 GNU 自由文档许可证 1.3 版或自由软件基金会发布的任何更新版本的条款，允许复制、分发和/或修改本文档; 没有不变的部分，没有封面文本，也没有封底文本。许可证的副本可以在 [GNU 网站](https://www.gnu.org/licenses/) 找到。

OTRS (Open Technology – Real Service) 是一个开源的工单请求系统，具有许多管理客户电话和电子邮件的功能。它在 GNU 通用公共许可证 (GPL) 下分发，并在各种 Linux 平台上进行过测试。您是否收到许多电子邮件，并希望与服务团队一起响应？你会爱上 OTRS 的！

1.1 关于本手册

本手册供系统管理员使用。这些章节描述了 OTRS 软件的安装和更新。

没有用于安装和更新的图形用户界面。系统管理员必须遵循以下章节中描述的步骤。

所有控制台命令看起来都像 `username> command-to-execute`。 `username` (用户名) 表示需要用来执行这些命令的操作系统的用户帐户。如果命令以 `root>` 开头，则必须以具有 `root` 权限的用户身份执行该命令。如果命令以 `otrs>` 开头，则必须按照为 OTRS 创建的用户执行命令。

警告： 复制命令并将其粘贴到 `shell` 时，不要选择 `username>` 部分，否则你会出错。

我们假定 OTRS 将被安装到 `/opt/otrs`。如果要安装到其它目录，则必须更改命令中的路径或创建指向此目录的符号链接。

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

硬件和软件要求

OTRS 可以安装在 Linux 和其他的 Unix 派生版本（例如 OpenBSD 或 FreeBSD）上，无法在 Microsoft Windows 上运行 OTRS。

要运行 OTRS，您还需要使用 Web 服务器作为反向代理和数据库服务器。除此之外，您应该在 OTRS 计算机上安装 Perl 和/或安装一些额外的 Perl 模块。

Perl 必须与 OTRS 安装在同一台机器上。数据库后端和 Web 服务器可以本地安装，也可以安装在其他主机上。

对于 Perl，您将需要一些额外的模块，可以使用 Perl shell 和 CPAN 安装，也可以通过操作系统的包管理器（rpm、yast、apt-get）安装。

OTRS 有一个控制台命令来检查环境和缺少的模块。

```
otrs> /opt/otrs/bin/otrs.CheckEnvironment.pl
```

如果缺少某些软件包，使用 --list 选项运行脚本，则可以获取当前操作系统安装这些软件包的命令。

```
otrs> /opt/otrs/bin/otrs.CheckEnvironment.pl --list
```

如果安装了所有需要的软件包，则环境检查脚本的输出将显示已安装的软件包和版本号。

```
Checking for Perl Modules:
  o Archive::Tar.....ok (v2.24)
  o Archive::Zip.....ok (v1.63)
  o Crypt::Eksblowfish::Bcrypt.....ok (v0.009)
  o Crypt::SSLeay.....ok (v0.73_06)
  o CryptX.....ok (v0.061)
  o Date::Format.....ok (v2.24)
  o DateTime.....ok (v1.50)
  o DBI.....ok (v1.641)
  o DBD::mysql.....ok (v4.046)
  o DBD::ODBC.....Not installed! Use: 'apt-get install -y↳
↳libdbd-odbc-perl' (optional - Required to connect to a MS-SQL database.)
```

(下页继续)

(续上页)

```

  o DBD::Oracle.....Not installed! Use: 'cpan DBD::Oracle'
->(optional - Required to connect to a Oracle database.)
  o DBD::Pg.....Not installed! Use: 'apt-get install -y
->libdbd-pg-perl' (optional - Required to connect to a PostgreSQL database.)
  o Digest::SHA.....ok (v5.96)
  o Encode::HanExtra.....ok (v0.23)
  o EV.....ok (v4.22)
  o IO::Socket::SSL.....ok (v2.060)
  o JSON::XS.....ok (v3.04)
  o List::Util::XS.....ok (v1.46_02)
  o LWP::UserAgent.....ok (v6.35)
  o Mail::IMAPClient.....ok (v3.39)
    o Authen::SASL.....ok (v2.16)
    o Authen::NTLM.....ok (v1.09)
  o Moose.....ok (v2.2011)
  o Net::DNS.....ok (v1.17)
  o Net::LDAP.....ok (v0.65)
  o Search::Elasticsearch.....ok (v6.00)
  o Specio.....ok (v0.42)
  o Specio::Subs.....ok (v0.42)
  o Template.....ok (v2.27)
  o Template::Stash::XS.....ok (undef)
  o Text::CSV_XS.....ok (v1.36)
  o Time::HiRes.....ok (v1.9741)
  o XML::LibXML.....ok (v2.0132)
  o XML::LibXSLT.....ok (v1.96)
  o XML::Parser.....ok (v2.44)
  o YAML::XS.....ok (v0.74)

Checking for External Programs:
  o GnuPG.....ok (v2.2.8)
  o npm.....ok (v5.8.0)
    o Node.js.....ok (v8.11.4)
  o OpenSSL.....ok (v1.1.1/OpenSSL)

```

2.1 硬件要求

硬件要求高度依赖于 OTRS 的使用情况。OTRS 可用于每月处理几张工单或每天处理数百张工单。存储要求还取决于工单数量和附件大小。

我们推荐使用如下配置的计算机：

- AMD 锐龙 7 3700X Octa 核心或同级别的 CPU
- 64GB 内存
- 2×1 TB NVMe SSD (软件 RAID 1)
- 千兆网卡

2.2 软件要求

Perl

- Perl 5.16.0 或更高版本
- 控制台命令 `/opt/otrs/bin/otrs.CheckEnvironment.pl` 列出的 Perl 包

Web 服务器

- Apache2
- nginx
- 可以用作反向代理的任何其他 Web 服务器

数据库

- MySQL 5.0 或更高版本
- MariaDB
- PostgreSQL 9.2 或更高版本
- Oraclr 10g 或更高版本

其它依赖

- Elasticsearch 6.x(不支持更高版本)
- Elasticsearch 模块 “analysis-icu” 和 “ingest-attachment”
- Node.js 8.9 或更高版本

浏览器

- Apple Safari 7 或更高版本
- Google Chrome
- Microsoft Internet Explorer 11
- Microsoft Edge
- Mozilla Firefox 32 或更高版本
- 其他支持 JavaScript 的任何现代 Web 浏览器

注解: OTRS 9 将不再支持 IE 浏览器。

本章描述了 OTRS 中央框架的安装和基本配置。

跟随本章的详细步骤来安装 OTRS 到你的服务器中，然后你就可以使用它的 WEB 界面来登录并管理系统了。

3.1 准备：禁用 SELinux

注解：如果你的系统使用了 SELinux，你需要禁用它，否则 OTRS 无法正常工作。

下面是如何在 RHEL/CentOS/Fedora 中禁用 SELinux。

1. 在文件 `/etc/selinux/config` 中配置 `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. 重启系统。重启后确认命令 `getenforce` 返回为 *Disabled*：

```
root> getenforce
Disabled
```

3.2 第 1 步：解压并安装应用程序

你可以下载 `otrs-x.y.z.tar.gz` 或 `otrs-x.y.z.tar.bz2`。将源存档（例如，使用 `tar`）解压到目录 `/opt` 中，并创建一个符号链接 `/opt/otrs`，指向 `/opt/otrs-x.y.z`。不要忘记替换版本号！

注解：默认情况下，某些系统中未安装 `bzip2` 包。确保在解压 `otrs-x.y.z.tar.bz2` 之前安装了 `bzip2`。

`otrs-x.y.z.tar.gz` 的解压命令：

```
root> tar -xzf otrs-x.y.z.tar.gz -C /opt
```

`otrs-x.y.z.tar.bz2` 的解压命令：

```
root> tar -xjf otrs-x.y.z.tar.bz2 -C /opt
```

建议创建一个名为 `/opt/otrs` 的符号链接，它始终指向最新的 OTRS 版本。使用符号链接可以轻松管理 OTRS 更新，因为您可以保留先前版本的目录，只需要更改符号链接。如果需要回退更新，改回符号链接的目标为原目录即可。

执行此命令以创建符号链接：

```
root> ln -fns /opt/otrs-x.y.z /opt/otrs
```

3.3 第 2 步：安装其它程序和 Perl 模块

使用以下脚本可以得到所有已安装的和必需的 CPAN 模块及其它外部依赖项的概览。

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

```

注解：请注意：OTRS 需要一个能工作的、安装了所有核心模块的 Perl。这些模块不在脚本的检查范围，在某些系统上（如 RHEL）默认没有安装这些核心模块，你可能需要安装一个 `perl-core` 软件包。

要安装必需和可选的软件包，可以使用 CPAN 或 Linux 发行版的软件包管理器。

执行此命令以获取用于安装缺少的依赖项的安装命令：

```
root> /opt/otrs/bin/otrs.CheckEnvironment.pl --list
```

OTRS 需要安装受支持的稳定版本的 Node.js。请参阅 [Node.js 安装说明](#)。

3.4 第 3 步：创建 OTRS 用户

为 OTRS 创建一个专用用户和组：

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

3.5 第 4 步：激活默认配置文件

OTRS 在 `$OTRS_HOME/Kernel/Config.pm.dist` 中自带了一个配置文件。您需要将它复制为不带 `.dist` 扩展名的文件来激活它。

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

3.6 第 5 步：配置 Apache Web 服务器

OTRS 附带一个内置 Web 服务器，在 Apache（或任何其他反向代理服务器）后面用作反向代理。正确运行需要一些 Apache 模块：

- mod_headers
- mod_proxy
- mod_proxy_http
- mod_proxy_wstunnel

在某些系统（如 Debian 和 SuSE）上，需要专门启用这些模块：

```
root> a2enmod headers
root> a2enmod proxy
root> a2enmod proxy_http
root> a2enmod proxy_wstunnel
```

几乎所有的 Apache 安装都包含一个 `conf.d` 目录。在 Linux 系统上，你通常可以在 `/etc/apache` 或 `/etc/apache2` 下找到这个目录。以 `root` 身份登录，切换到 `conf.d` 目录，并在 Apache 配置目录创建一个到 `/opt/otrs/scripts/apache2-httpd.include.conf` 的名为 `zzz_otrs.conf` 的软链接（以确保在其他配置之后加载 OTRS 配置）。

```
# Debian/Ubuntu:
root> ln -s /opt/otrs/scripts/apache2-httpd.include.conf /etc/apache2/sites-
→enabled/zzz_otrs.conf
```

现在，您可以重新启动 Web 服务器以加载新的配置设置。在大多数系统上，您可以使用以下命令执行此操作：

```
root> systemctl restart apache2.service
```

3.7 第 6 步：设置文件权限

请执行以下命令为 OTRS 设置文件和目录权限。它将尝试检测你的安装所需的正确的用户和组设置。

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

3.8 第 7 步：安装数据库

为了 OTRS 能正常使用，需要执行以下步骤设置数据库：

- 创建一个专用的数据库用户和数据库。
- 创建数据库结构。
- 插入初始数据。
- 在 Kernel/Config.pm 中配置数据库连接。

注解：请注意，OTRS 需要数据库存储编码为 utf8。

3.8.1 MySQL 或 MariaDB

以数据库管理员用户身份登录 MySQL 控制台：

```
root> mysql -uroot -p
```

创建一个数据库：

```
mysql> CREATE DATABASE otrs CHARACTER SET utf8;
```

MySQL 8 需要特殊的数据库用户处理，因为默认的 caching_sha2_password 只能用于安全连接。在 MySQL 8 中创建一个数据库用户：

```
mysql> CREATE USER 'otrs'@'localhost' IDENTIFIED WITH mysql_native_password
↳BY 'choose-your-password';
```

在 MySQL 8 之前的版本中创建一个数据库用户：

```
mysql> CREATE USER 'otrs'@'localhost' IDENTIFIED BY 'choose-your-password';
```

将用户权限分配给新数据库：

```
mysql> GRANT ALL PRIVILEGES ON otrs.* TO 'otrs'@'localhost';
mysql> FLUSH PRIVILEGES;
mysql> quit
```

在 shell 中运行以下命令以创建数据库结构并插入数据：

```
root> mysql -uroot -p otrs < /opt/otrs/scripts/database/otrs-schema.mysql.sql
root> mysql -uroot -p otrs < /opt/otrs/scripts/database/otrs-initial_insert.
↳mysql.sql
root> mysql -uroot -p otrs < /opt/otrs/scripts/database/otrs-schema-post.
↳mysql.sql
```

在 Kernel/Config.pm 中配置数据库设置：

```

$self->{DatabaseHost} = '127.0.0.1';
$self->{Database}     = 'otrs';
$self->{DatabaseUser} = 'otrs';
```

(下页继续)

(续上页)

```
$Self->{DatabasePw} = 'choose-your-password';
$Self->{DatabaseDSN} = "DBI:mysql:database=$Self->{Database};host=$Self->
->{DatabaseHost}";
```

注解：对于 MySQL，建议使用以下配置设置。请在文件 `/etc/my.cnf` 的 `[mysqld]` 部分下面添加以下行：

```
max_allowed_packet = 64M
query_cache_size = 32M
innodb_log_file_size = 256M
```

3.8.2 PostgreSQL

注解：我们假设，OTRS 和 PostgreSQL 服务器在同一台机器上运行，PostgreSQL 使用 *Peer* 身份验证方法。有关更多信息，请参阅 PostgreSQL 手册中的 [Client Authentication](#) 部分。

切换到 postgres 用户：

```
root> su - postgres
```

创建一个数据库用户：

```
postgres> createuser otrs
```

创建一个数据库：

```
postgres> createdb --encoding=UTF8 --owner=otrs otrs
```

在 shell 中运行以下命令以创建数据库结构并插入数据：

```
otrs> psql < /opt/otrs/scripts/database/otrs-schema.postgresql.sql
otrs> psql < /opt/otrs/scripts/database/otrs-initial_insert.postgresql.sql
otrs> psql < /opt/otrs/scripts/database/otrs-schema-post.postgresql.sql
```

在 Kernel/Config.pm 中配置数据库设置：

```
$Self->{DatabaseHost} = '127.0.0.1';
$Self->{Database} = 'otrs';
$Self->{DatabaseUser} = 'otrs';
$Self->{DatabasePw} = 'choose-your-password';
$Self->{DatabaseDSN} = "DBI:Pg:dbname=$Self->{Database};host=$Self->
->{DatabaseHost}";
```

3.8.3 完成数据库设置

要验证数据库设置，请运行以下命令：

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::Database::Check
Trying to connect to database 'DBI:Pg:dbname=otrs;host=localhost' with user
↳ 'otrs'...
Connection successful.
```

正确配置数据库后，请使用以下命令初始化系统配置：

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::Config::Rebuild
Rebuilding the system configuration...
Done.
```

注解：出于安全考虑，请通过生成随机密码来更改管理员用户 `root@localhost` 的默认密码：

```
otrs> /opt/otrs/bin/otrs.Console.pl Admin::User::SetPassword root@localhost
Generated password 'rtB98S55kuc9'.
Successfully set password for user 'root@localhost'.
```

您也可以选择设置自己的密码：

```
otrs> /opt/otrs/bin/otrs.Console.pl Admin::User::SetPassword root@localhost↳
↳ your-own-password
Successfully set password for user 'root@localhost'
```

3.9 第 8 步：设置 **Elasticsearch** 集群

OTRS 需要 **Elasticsearch 6.x** 的活动集群（不支持更高版本）。最简单的方法是在 OTRS 同一台主机上设置 **Elasticsearch** 并将其绑定到其默认端口。这样就不需要在 OTRS 中进一步配置。

此外，OTRS 需要在 **Elasticsearch** 中安装以下插件：

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

注解：之后重新启动 **Elasticsearch**，否则将无法构建索引。

要验证 **Elasticsearch** 安装，可以使用以下命令：

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::DocumentSearch::Check
Trying to connect to cluster...
Connection successful.
```

3.10 第 9 步：启动 **OTRS** 守护进程和 **Web** 服务器

新的 OTRS 守护进程负责处理 OTRS 中的任何异步和重复任务。内置的 OTRS Web 服务器进程处理从 Apache 移交的 Web 请求。这两个进程必须以 `otrs` 用户启动。

```
otrs> /opt/otrs/bin/otrs.Daemon.pl start
otrs> /opt/otrs/bin/otrs.WebServer.pl
```

3.11 第 10 步：首次登录

现在你可以通过 <http://localhost/otrs/index.pl> 登录系统了，使用用户 `root@localhost` 和之前生成的密码（见上）。

使用 <http://localhost> 访问外部人员界面。

3.12 第 11 步：设置 Systemd 文件

OTRS 附带了 `systemd` 示例配置文件，可用于确保在系统启动后自动启动 OTRS 守护进程和 Web 服务器。

```
root> cd /opt/otrs/scripts/systemd
root> for UNIT in *.service; do cp -vf $UNIT /usr/lib/systemd/system/;
↳systemctl enable $UNIT; done
```

通过本步骤，完成了系统的基本配置。

3.13 第 12 步：设置 Bash 自动补全（可选）

所有常规 OTRS 命令行操作都通过 OTRS 控制台界面进行。这为 `bash shell` 提供了自动补全功能，这使得查找正确的命令和选项变得更加容易。

您可以通过安装包 `bash-completion` 来激活 `bash` 自动补全。它会自动检测并加载 `otrs` 用户的 `/opt/otrs/.bash_completion` 文件。

重新启动 `shell` 后，您只需键入此命令，然后键入 `TAB`，即可列出所有可用命令：

```
otrs> /opt/otrs/bin/otrs.Console.pl
```

如果键入命令名的几个字符，按 `TAB` 键将显示所有匹配的命令。键入完整命令后，按 `TAB` 后将显示所有可能的选项和参数。

注解：如果遇到问题，可以通过在 `~/.bashrc` 文件中添加以下行来执行这些命令。

```
source /opt/otrs/.bash_completion
```

3.14 步骤 13：进一步信息

我们建议您阅读 [OTRS 性能优化](#) 部分。

注解：强烈建议在一台独立的测试主机上先进行升级测试。

从 **OTRS 7** 的早期版本更新 你可以直接从任何的先前版本升级到最新的补丁级别版本。

从 **OTRS 6** 更新 你可以从 **OTRS 6** 的任何补丁级别版本更新到 **OTRS 7** 最新的可用补丁级别版本。

从 **OTRS 5** 或更早版本更新 你不能从 **OTRS 5** 或更早版本直接更新到 **OTRS 7**。必须依次对所有可用的次要版本进行完全更新。例如，如果你从 **OTRS 4** 开始，则首先必须完整更新到 **OTRS 5**，然后更新到 **OTRS 6**，最后更新到 **OTRS 7**。

参见：

有关更新说明，请参阅以前版本的 **OTRS** 管理员手册。

4.1 第 1 步：停止所有相关服务和 **OTRS** 守护进程

请确保没有任何运行中的服务或 **CRON** 计划任务还在试图访问 **OTRS**，这取决于你的服务配置和 **OTRS** 版本。

```
root> systemctl stop postfix
root> systemctl stop apache2
```

如果您从 **OTRS 6** 进行重大更新，则需要停止旧的 **OTRS cron** 任务和守护进程（按此顺序）：

```
otrs> /opt/otrs/bin/Cron.sh stop
otrs> /opt/otrs/bin/otrs.Daemon.pl stop
```

如果在 **OTRS 7** 中执行补丁级别更新（使用新的 **systemd** 文件），请通过 **systemd** 停止 **OTRS** 服务：

```
root> systemctl stop otrs-daemon
root> systemctl stop otrs-webserver
```

4.2 第 2 步：备份文件和数据库

创建下列文件和目录的备份：

- Kernel/Config.pm
- Kernel/WebApp.conf (仅在 OTRS 7 的补丁级别更新的情况下，并且仅在文件被修改时)
- var/*
- 当然还有数据库

警告： 如果没有完整的系统备份，请不要继续。使用备份脚本。

4.3 第 3 步：安装新版本

注解： OTRS 7 不再提供 RPM 包，基于 RPM 的安装需要先卸载原有的 RPM 包(这不会删除数据库)，然后使用源文件安装。

你可以下载 `otrs-x.y.z.tar.gz` 或 `otrs-x.y.z.tar.bz2`。将源存档(例如，使用 `tar`)解压到目录 `/opt` 中，并创建一个符号链接 `/opt/otrs`，指向 `/opt/otrs-x.y.z`。不要忘记替换版本号！

注解： 默认情况下，某些系统中未安装 `bzip2` 包。确保在解压 `otrs-x.y.z.tar.bz2` 之前安装了 `bzip2`。

`otrs-x.y.z.tar.gz` 的解压命令：

```
root> tar -xzf otrs-x.y.z.tar.gz -C /opt
```

`otrs-x.y.z.tar.bz2` 的解压命令：

```
root> tar -xjf otrs-x.y.z.tar.bz2 -C /opt
```

建议创建一个名为 `/opt/otrs` 的符号链接，它始终指向最新的 OTRS 版本。使用符号链接可以轻松管理 OTRS 更新，因为您可以保留先前版本的目录，只需要更改符号链接。如果需要回退更新，改回符号链接的目标为原目录即可。

执行此命令以创建符号链接：

```
root> ln -fns /opt/otrs-x.y.z /opt/otrs
```

4.3.1 恢复原配置文件

- Kernel/Config.pm
- Kernel/WebApp.conf (仅在 OTRS 7 的补丁级别更新的情况下，并且仅在文件被修改时)

4.3.2 恢复信件数据

如果您将 OTRS 配置为在文件系统中存储信件数据，则必须将 `article` 目录恢复到 `/opt/otrs/var/` 或系统配置中指定的目录。

4.3.3 恢复已安装的默认统计

如果您有其它具有默认统计信息的软件包，则必须将带有后缀 `*.installed` 的统计信息 XML 文件恢复到 `/opt/otrs/var/stats`。

```
root> cd OTRS-BACKUP/var/stats
root> cp *.installed /opt/otrs/var/stats
```

4.3.4 设置文件权限

请执行以下命令为 OTRS 设置文件和目录权限。它将尝试检测你的安装所需的正确的用户和组设置。

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

4.3.5 安装必需的程序和 Perl 模块

请参阅安装指南中的第 2 步：安装其它程序和 Perl 模块 部分，其中介绍了如何验证外部依赖项，如 Perl 模块和 Node.js。

除此之外，OTRS 7 还需要 Elasticsearch 6.x 的活动集群（不支持更高版本）。请参阅安装指南中的第 8 步：设置 Elasticsearch 集群 部分。

4.4 第 4 步：运行迁移脚本

迁移脚本将对你的系统执行许多检查，并提供有关如何安装缺少的 Perl 模块等的建议（如果是必需的模块）。如果所有检查成功，将执行必要的迁移步骤。如果是补丁级别更新，也请运行此脚本。

运行这个迁移脚本：

```
otrs> /opt/otrs/scripts/DBUpdate-to-7.pl
```

警告： 如果此脚本无法正常工作，请不要继续升级过程。否则可能会发生故障或丢失数据。

迁移脚本还会检查 ACL 和系统配置设置是否正确。如果系统配置设置无效，脚本将为您提供通过从可能值列表中进行选择来修复它的机会。如果脚本以非交互模式运行，它将尝试自动修复无效设置。如果此操作失败，将要求您在迁移后手动更新设置。

如果存在过期的 ACL，系统将无法自动修复它们，并且需要由管理员进行更正。有关详细信息，请参阅手动更改的最后一步。

4.5 第 5 步：更新已安装的软件包

注解：OTRS 6 的软件包与 OTRS 7 不兼容，必须进行更新。

你可以使用下面的命令来更新所有已安装的软件包。这适用于从在线存储库提供的所有软件包。您稍后可以通过软件包管理器更新其他软件包（这需要 OTRS 守护进程在运行）。

```
otrs> /opt/otrs/bin/otrs.Console.pl Admin::Package::UpgradeAll
```

4.6 第 6 步：启动服务

OTRS 7 带有一个自己的内置 Web 服务器，在 Apache 后面用作反向代理（或任何其它反向代理服务器）。对于从 OTRS 6 的大版本更新，必须使用 `/opt/otrs/scripts/apache2-httpd.include.conf` 中的新版本更新 Apache 配置，如果它已被复制而不仅仅是链接。

还请注意，虽然不再需要“`mod_perl`”模块，但现在需要其它 Apache 模块：

- `mod_headers`
- `mod_proxy`
- `mod_proxy_http`
- `mod_proxy_wstunnel`

之后，可以启动服务。这取决于您的服务配置，这是一个示例：

```
root> systemctl start postfix
root> systemctl start apache2
```

注解：OTRS 守护进程是正确操作 OTRS 所必需的，例如发送电子邮件。请按照下一步中的说明激活它。

4.7 第 7 步：启动 OTRS 守护进程和 Web 服务器

OTRS 守护进程负责处理 OTRS 中的任何异步和重复任务。内置的 OTRS Web 服务器进程处理从 Apache 移交的 Web 请求。

OTRS 附带了 `systemd` 示例配置文件，可用于确保在系统启动后自动启动 OTRS 守护进程和 Web 服务器。

```
root> cd /opt/otrs/scripts/systemd
root> for UNIT in *.service; do cp -vf $UNIT /usr/lib/systemd/system/;
↪systemctl enable $UNIT; done
root> systemctl start otrs-daemon
root> systemctl start otrs-webserver
```

现有你可以登录到系统了。

4.8 第 8 步：手动迁移任务和更改

警告： 只有从 OTRS 6 的大版本更新才需要此步骤。

由于旧的客户界面屏幕不再存在，因此管理员需要手动更正某些 ACL。如果是这种情况，迁移脚本已通知您。

受影响的 ACL 是指在 Action(操作) 设置中引用不存在的客户界面屏幕的 ACL。这个前端 Action(操作) 规则需要用相应的 Endpoint(端点) 规则替换。下面包括一个可能的映射表。

操作	端点
CustomerTicket-Print	没有替换(功能缺失)
CustomerTicket-Zoom	ExternalFrontend::TicketDetailView
CustomerTicket-Process	ExternalFrontend::ProcessTicketCreate 或 ExternalFrontend::ProcessTicketNextStep
CustomerTicketMessage	ExternalFrontend::TicketCreate

OTRS 具有备份和还原的内置脚本。使用 `-h` 选项执行此脚本以获取更多信息。

5.1 备份

注解：要创建备份，otrs 用户需要有目标目录的写入权限。

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

脚本的输出结果：

```
Backup an OTRS system.

Usage:
 backup.pl -d /data_backup_dir [-c gzip|bzip2] [-r DAYS] [-t
 →fullbackup|nofullbackup|dbonly]

Options:
 -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.
 [-h]              - Display help for this command.

Help:
Using -t fullbackup saves the database and the whole OTRS home directory
 →(except /var/tmp and cache directories).
```

(下页继续)

(续上页)

```
Using -t nofullbackup saves only the database, /Kernel/Config* and /var_
↳directories.
```

```
With -t dbonly only the database will be saved.
```

```
Output:
```

```
Config.tar.gz           - Backup of /Kernel/Config* configuration files.
Application.tar.gz      - Backup of application file system (in case of full_
↳backup).
VarDir.tar.gz           - Backup of /var directory (in case of no full_
↳backup).
DataDir.tar.gz          - Backup of article files.
DatabaseBackup.sql.gz  - Database dump.
```

5.2 恢复

```
otrs> /opt/otrs/scripts/restore.pl -h
```

脚本的输出结果：

```
Restore an OTRS system from backup.
```

```
Usage:
```

```
restore.pl -b /data_backup/<TIME>/ -d /opt/otrs/
```

```
Options:
```

```
-b           - Directory of the backup files.
-d           - Target OTRS home directory.
[-h]        - Display help for this command.
```

下面是用于 OTRS 安装的性能增强技术清单，包括配置、编码、内存使用等。

6.1 工单索引模块

可以在系统配置设置 `Ticket::IndexModule` 中设置工单索引模块。工单队列视图的索引有两个后端模块：

Kernel::System::Ticket::IndexAccelerator::RuntimeDB 这是默认选项，并将从工单表中即时生成每个队列视图。在有 6 万个处理工单之前，系统都不会遇到性能问题。

Kernel::System::Ticket::IndexAccelerator::StaticDB 当您有超过 80,000 个处理中的工单时，应该使用这个最强大的模块。它使用一个额外的 `ticket_index` 表，用基于工单数据的关键字填充。在切换后端模块后使用下列命令生成初始索引：

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::Ticket::QueueIndexRebuild
```

6.2 工单搜索索引

OTRS 使用特殊的搜索索引对来自不同通信渠道的信件中的字段进行全文搜索。

使用这个命令生成初始索引：

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::Ticket::FulltextIndex --rebuild
```

注解：实际的信件索引通过后台的 OTRS 守护进程任务进行。虽然刚刚添加到系统中的信件被标记为立即进行索引，但其索引可能在几分钟后才可用。

有一些选项可用于微调搜索索引：

Ticket::SearchIndex::IndexArchivedTickets 定义归档工单是否包含在搜索索引中（默认情况下未启用）。建议在有归档工单的大型系统上保持较小的索引。如果这个选项启用了，全文搜索将能搜索到归档的工单。

Ticket::SearchIndex::Attribute 全文索引的基本设置。

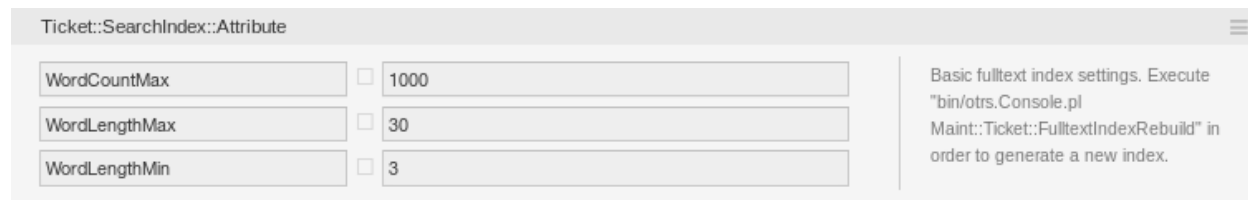


图 1: Ticket::SearchIndex::Attribute 设置

注解：运行以下命令以生成新索引：

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::Ticket::FulltextIndexRebuild
```

WordCountMax 定义用以构建索引的最大字数。例如，只有信件正文的前 1000 个单词存储在信件搜索索引中。

WordLengthMin 和 **WordLengthMax** 用作字长边界。只有具有这两个值之间长度的单词才会存储在信件搜索索引中。

Ticket::SearchIndex::Filters 全文索引正则表达式过滤器，用于删除部分文本。

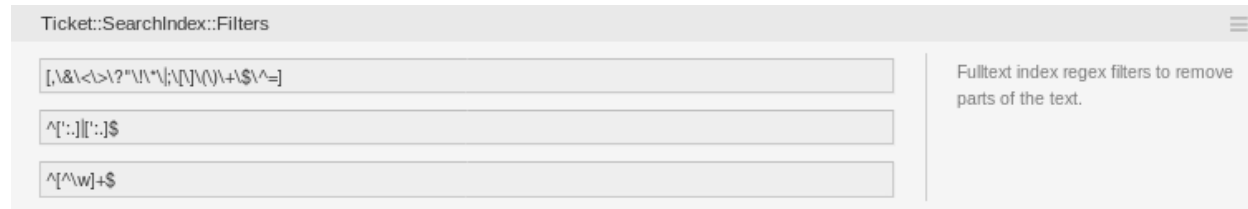


图 2: Ticket::SearchIndex::Filters 设置

定义了三个默认过滤器：

- 第一个过滤器剥离特殊的字符，如：, & < > ? ” ! * | ; [] () + \$ ^ =
- 第二个过滤器剥离使用以下字符之一开始或结束的单词： ‘ : 。
- 第三个过滤器剥离不包含字符的单词： a-z, A-Z, 0-9, _

Ticket::SearchIndex::StopWords 全文索引的英语停止词，这些词将从搜索索引中移除。

有一些语言定义了所谓的停止词。创建搜索索引时将跳过这些停止词。

参见：

如果您的语言不在系统配置设置中，或者您想要添加更多单词，则可以将它们添加到此设置：

- Ticket::SearchIndex::StopWords###Custom



图 3: Ticket::SearchIndex::StopWords###en 设置

6.3 文件搜索

OTRS 使用 Elasticsearch 进行文档搜索。有关 Elasticsearch 的概念、安装和使用的详细介绍，请遵循入门指南。

6.3.1 堆大小

Elasticsearch 是用 Java 编写的，因此可以在任何集群节点上的 Java 虚拟机 (JVM) 中运行。JVM 使用称为 *heap* (堆) 的一部分内存，其大小可以在配置文件 `jvm.options` 中配置。

堆最小和最大配置默认设置为 1 GB，可以使用以下选项进行修改：

- `Xms1g`：堆最小值为 1GB。
- `Xmx1g`：堆最大值为 1GB。

如果 `Xms` 的值低于 `Xmx`，则 JVM 将在超过当前限制时调整使用的堆的大小，直到达到 `Xmx` 的值。这样的大小调整导致服务暂停直到它完成，这可能降低搜索或索引动作的速度和反应性。因此，强烈建议将它们设置为相等的值。

警告： 如果超出了最大堆大小，则相关的群集节点将停止工作，甚至可能会关闭该服务。

设置的堆最大值越高，Elasticsearch 可以使用的内存越多，这也会增加 JVM 完成垃圾收集可能的暂停。因此，建议为 `Xmx` 设置一个不高于物理内存 50% 的值。

有关堆大小的更多信息和经验法则，请遵循 [官方文档](#)。

6.3.2 磁盘分配

在服务运行期间，Elasticsearch 会检查可用磁盘空间，因此决定是否将新分片分配给相关群集节点，甚至是将分片重新定位到特定节点之外。这种行为将由当前磁盘容量控制，可以在配置文件 `elasticsearch.yml` 中配置。随附一些重要的配置，它们具有良好但可能很重要的默认值：

`cluster.routing.allocation.disk.watermark.low` 默认值为 85%。如果超出此限制，Elasticsearch 将不会将更多分片分配给相关的群集节点。该节点的操作不受影响，仍然可以索引和搜索数据。

`cluster.routing.allocation.disk.watermark.high` 默认值为 90%。如果超出此限制，Elasticsearch 将尝试将现有分片重定位到具有足够可用空间的其它节点（如果有）。

`cluster.routing.allocation.disk.watermark.flood_stage` 默认值为 95%。如果超出此限制，Elasticsearch 会将所有索引的配置更新为只读索引块 `index.blocks.read_only_allow_delete`，其中至少有一个分片分配给相关的群集节点。从那时起，无法将新数据索引到此类索引并限制为仅搜索和删除操作。

注解：如果超出泛洪阶段并且某些索引配置为只读模式，则 Elasticsearch 不会自动更改此配置。如果相关磁盘由于手动操作而再次包含足够的可用空间，则需要手动将配置更改回正常模式。

有关磁盘水位线和基于磁盘的分片分配的更多信息，请参阅 [官方文档](#)。

6.4 信件存储

有两种不同的后端模块可用于存储电话、电子邮件和内部信件。使用的信件存储可以在设置 `Ticket::Article::Backend::MIMEBase::ArticleStorage` 中配置。

`Kernel::System::Ticket::Article::Backend::MIMEBase::ArticleStorageDB` 此默认模块将附件存储在数据库中。它也适用于多个前端服务器，但在数据库中需要很多存储空间。

注解：在大型安装环境中不要使用它。

`Kernel::System::Ticket::Article::Backend::MIMEBase::ArticleStorageFS` 使用此模块可在本地文件系统上存储附件。它很快，但如果您有多个前端服务器，则必须确保文件系统在服务器之间共享。将其放在 NFS 共享上或者最好是 SAN 或类似的解决方案中。

注解：在大型安装环境中推荐使用。

您可以在运行中从一个后端切换到另一个后端。您可以在系统配置中切换后端，然后运行此命令行实用程序将数据库中的信件放到文件系统上，反之亦然：

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

您可以使用 `--target` 选项指定目标后端。

注解：整个过程可能需要相当长的时间，具体取决于您拥有的信件数量以及可用的 CPU 资源和/或网络容量。

如果要在数据库中保留原有附件，可以激活系统配置选项 `Ticket::Article::Backend::MIMEBase::CheckAllStorageBackends` 以确保 OTRS 仍然可以找到它们。

6.5 归档工单

由于 OTRS 可以用作审核系统，删除关闭的工单可能不是一个好主意。因此，我们实现了一个可以让您归档工单的功能。

匹配某个条件的工单可以标记为“已归档”。这些工单在使用常规的工单搜索或运行一个自动任务时无法访问。系统本身不再需要处理大量的工单，而只考虑最近的工单就可以了。这在大型系统中能带来巨大的性能提升。

要使用归档功能：

1. 激活系统配置中的 `Ticket::ArchiveSystem` 设置。
2. 定义一个自动任务：
 - 在自动任务屏幕中点击添加任务按钮。
 - 任务设置：提供归档任务的名称。
 - 自动执行：为计划该任务选择合适的选项。
 - 选择工单：最好只将几个月前就关闭了的处于关闭状态的工单归档。
 - 更新/添加工单属性：设置归档选中的工单字段为归档工单。
 - 在页面底部保存该任务。
 - 点击概览表格中的运行该任务链接以查看受影响的工单。
 - 点击运行任务按钮。

注解：手动运行此任务最多可以修改 5000 个工单。

当你搜索工单时，系统默认搜索未归档的工单。

为了搜索归档的工单：

1. 打开工单搜索屏幕。
2. 设置归档搜索为未归档工单或所有工单。
3. 执行搜索。

6.6 优化 Web 服务器

OTRS 的内置 Web 服务器可以开箱即用地处理中小型安装。当 OTRS 同时为许多用户提供服务时，可能需要调整 Web 服务器配置，比如增加工作进程的数量。

Web 服务器配置文件位于 `Kernel/WebApp.conf`，并且所有设置均有文档说明。可以增加 `worker` 设置，以在有能力的服务器上部署更多进程，用于处理 HTTP 请求。

6.7 缓存

OTRS 在 `/opt/otrs/var/tmp` 中缓存了大量临时数据。请确保它使用高性能文件系统和存储。如果你有足够的 RAM，你也可以尝试将这个目录放在 `ramdisk` 上，如下所示：

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

注解：在 `/etc/fstab` 中添加永久性挂载点。

警告： 这将是一个非永久性存储，在服务器重启时会丢失，所有的会话（如果将它们存储在文件系统中）和缓存数据都将丢失。

参见：

还有一个集中的 基于 `memcached` 的缓存后端 可从 OTRS 集团购买。

6.8 集群

对于非常高的负载，可能需要在多个前端服务器的集群上运行 OTRS。这是一项复杂的任务，存在许多陷阱。因此，OTRS 集团专门为其托管的 OTRS 环境中的集群提供支持。