



# **OTRS Update and Installation Guide**

*Release 7.0*

**OTRS AG**

**07.05.2020**



<b>1</b>	<b>Einführung</b>	<b>3</b>
1.1	Über diese Anleitung . . . . .	3
<b>2</b>	<b>Hardware- und Software-Anforderungen</b>	<b>5</b>
2.1	Hardware Anforderungen . . . . .	6
2.2	Software-Anforderungen . . . . .	7
<b>3</b>	<b>Installation</b>	<b>9</b>
3.1	Vorbereitung: SELinux deaktivieren . . . . .	9
3.2	Schritt 1: Anwendung entpacken und installieren . . . . .	10
3.3	Schritt 2: Zusätzliche Programme und Perl-Module installieren . . . . .	10
3.4	Schritt 3: Einen OTRS-Benutzer erstellen . . . . .	11
3.5	Schritt 4: Standard-Konfigurationsdatei aktivieren . . . . .	11
3.6	Schritt 5: Apache-Webserver konfigurieren . . . . .	11
3.7	Schritt 6: Dateiberechtigungen festlegen . . . . .	12
3.8	Schritt 7: Die Datenbank einrichten . . . . .	12
3.8.1	MySQL oder MariaDB . . . . .	12
3.8.2	PostgreSQL . . . . .	13
3.8.3	Datenbank-Setup abschließen . . . . .	14
3.9	Schritt 8: Elasticsearch-Cluster einrichten . . . . .	14
3.10	Schritt 9: OTRS Daemon und Webserver starten . . . . .	15
3.11	Schritt 10: Die erste Anmeldung . . . . .	15
3.12	Schritt 11: Systemd-Dateien einrichten . . . . .	15
3.13	Schritt 12: Automatische Bash-Vervollständigung einrichten (optional) . . . . .	15
3.14	Schritt 13: Weitere Informationen . . . . .	16
<b>4</b>	<b>Aktualisieren</b>	<b>17</b>
4.1	Schritt 1: Alle relevanten Dienste und den OTRS Deamon beenden . . . . .	17
4.2	Schritt 2: Dateien und Datenbank sichern . . . . .	18
4.3	Schritt 3: Das neue Release installieren . . . . .	18
4.3.1	Alte Konfigurationsdateien wiederherstellen . . . . .	19
4.3.2	Artikeldaten wiederherstellen . . . . .	19
4.3.3	Bereits installierte Standardstatistik wiederherstellen . . . . .	19
4.3.4	Dateiberechtigungen festlegen . . . . .	19
4.3.5	Erforderliche Programme und Perl-Module installieren . . . . .	19
4.4	Schritt 4: Migrationsskript ausführen . . . . .	19
4.5	Schritt 5: Installierte Pakete aktualisieren . . . . .	20

4.6	Schritt 6: Ihre Dienste starten . . . . .	20
4.7	Schritt 7: OTRS-Daemon und Webserver starten . . . . .	21
4.8	Schritt 8: Manuelle Migrationsaufgaben und -änderungen . . . . .	21
<b>5</b>	<b>Sichern und Wiederherstellen</b>	<b>23</b>
5.1	Datensicherung . . . . .	23
5.2	Wiederherstellen . . . . .	24
<b>6</b>	<b>Leistungsverbesserung</b>	<b>25</b>
6.1	Ticket-Index-Modul . . . . .	25
6.2	Ticket-Suchindex . . . . .	25
6.3	Dokumentensuche . . . . .	27
6.3.1	Heap-Größe . . . . .	27
6.3.2	Festplattenzuordnung . . . . .	28
6.4	Artikelspeicherung . . . . .	28
6.5	Tickets archivieren . . . . .	29
6.6	Webserver optimieren . . . . .	30
6.7	Caching . . . . .	30
6.8	Clustering . . . . .	31



Dieses Werk ist urheberrechtlich geschützt von der OTRS AG (<https://otrs.com>), Zimmersmühlenweg 11, 61440 Oberursel, Deutschland.

Es ist erlaubt, dieses Dokument zu kopieren, verteilen und zu verändern unter Beachtung der Bedingungen der GNU Lizenz für freie Dokumentationen, Version 1.3 oder höher; ohne invariante Abschnitte, keine Titelseiten-Texte und keine Rückseiten-Texte. Eine Kopie der Lizenz finden Sie auf der [GNU website](#).



OTRS (Open Technology – Real Service) ist ein Open-Source-Ticketabfragesystem mit vielen Funktionen zur Verwaltung von Kunden-Telefonanrufen und E-Mails. Es wird unter der GNU General Public License (GPL) vertrieben und wurde auf verschiedenen Linux-Plattformen getestet. Erhalten Sie viele E-Mails und möchten diese mit einem Agenten-Team beantworten? Dann werden Sie OTRS lieben!

## 1.1 Über diese Anleitung

Dieses Handbuch ist für Systemadministratoren bestimmt. Die Kapitel beschreiben die Installation und Aktualisierung der OTRS-Software.

Es gibt keine grafische Benutzeroberfläche für die Installation und Aktualisierung. Systemadministratoren müssen die in den folgenden Kapiteln beschriebenen Schritte ausführen.

Alle Konsolenbefehle sehen aus wie `Benutzername> Auszuführender Befehl`. Der Benutzername gibt das Benutzerkonto des Betriebssystems an, das zur Ausführung des Befehls verwendet werden muss. Wenn ein Befehl mit `root>` beginnt, müssen Sie den Befehl als Benutzer mit Root-Berechtigungen ausführen. Wenn ein Befehl mit `otrs>` beginnt, müssen Sie den Befehl als Benutzer ausführen, der für OTRS erstellt wurde.

**Warnung:** Wählen Sie nicht `Benutzername>` aus, wenn Sie den Befehl kopieren und in die Shell einfügen. Andernfalls erhalten Sie eine Fehlermeldung.

Wir gehen davon aus, dass OTRS in `/opt/otrs` installiert wird. Wenn Sie OTRS in einem anderen Verzeichnis installieren möchten, dann müssen Sie den Pfad bei Befehlsaufruf ändern oder einen symbolischen Link zu diesem Verzeichnis erstellen.

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





---

## Hardware- und Software-Anforderungen

---

OTRS kann unter Linux und anderen Unix-Derivaten (z. B. OpenBSD oder FreeBSD) installiert werden. Das Ausführen von OTRS unter Microsoft Windows ist nicht möglich.

Um OTRS auszuführen, müssen Sie auch einen Webserver als Reverse Proxy und einen Datenbankserver verwenden. Ansonsten sollten Sie Perl installieren und / oder einige zusätzliche Perl-Module auf dem OTRS-Rechner installieren.

Perl muss auf demselben Rechner wie OTRS installiert sein. Das Datenbank-Backend und der Webserver können lokal oder auf einem anderen Host installiert werden.

Für Perl benötigen Sie einige zusätzliche Module, die entweder mit der Perl-Shell und CPAN oder über den Paketmanager Ihres Betriebssystems (rpm, yast, apt-get) installiert werden können.

OTRS verfügt über einen Konsolenbefehl zur Überprüfung der Umgebung und der fehlenden Module.

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

Wenn einige Pakete fehlen, können Sie einen Installationsbefehl für Ihr Betriebssystem erhalten, wenn Sie das Skript mit der Option `--list` ausführen.

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

Wenn alle benötigten Pakete installiert sind, werden in der Ausgabe des Umgebungsprüfungsskripts die installierten Pakete und die Versionsnummern angezeigt.

```
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)
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
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.)
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 Hardware Anforderungen

Die Hardware Anforderungen hängen stark davon ab, wie OTRS genutzt wird. OTRS kann genutzt werden, um ein paar Tickets pro Monat oder mehrere Tickets pro Tag zu verarbeiten. Die Anforderungen an den Speicherplatz hängen ebenfalls von der Anzahl der Tickets und der Größe der Anlagen ab.

Wir empfehlen die Verwendung einer Maschine mit:

- AMD Ryzen 7 3700X Octa core oder vergleichbare CPU
- 64 GB RAM
- 2 × 1 TB NVMe SSD (Software-RAID 1)
- Gigabit LAN

## 2.2 Software-Anforderungen

### Perl

- Perl 5.16.0 oder höher
- Perl-Pakete, die mit dem Konsolenbefehl `/opt/otrs/bin/otrs.CheckEnvironment.pl` aufgelistet werden

### Webserver

- Apache2
- nginx
- Jeder andere Webserver, der als Reverse Proxy verwendet werden kann

### Datenbanken

- MySQL 5.0 oder höher
- MariaDB
- PostgreSQL 9.2 oder höher
- Oracle 10g oder höher

### Andere Abhängigkeiten

- Elasticsearch 6.x (höhere Versionen werden nicht unterstützt)
- Elasticsearch-Module `analysis-icu` und `ingest-attachment`
- Node.js 8.9 oder höher

### Web Browser

- Apple Safari Version 7 oder höher
- Google Chrome
- Microsoft Internet Explorer 11
- Microsoft Edge
- Mozilla Firefox Version 32 oder höher
- Jeder andere moderne Webbrowser mit JavaScript-Unterstützung

---

**Bemerkung:** OTRS 9 wird den Internet Explorer nicht mehr unterstützen.

---



Dieses Kapitel beschreibt die Installation und die grundlegende Einrichtung des zentralen OTRS-Framework. Folgen Sie den Schritten in diesem Kapitel, um OTRS auf Ihrem Server zu installieren. Dann können Sie sich über die Weboberfläche anmelden und das System administrieren.

### 3.1 Vorbereitung: SELinux deaktivieren

---

**Bemerkung:** Wenn Ihr System SELinux benutzt, sollten Sie es deaktivieren. Andernfalls wird OTRS nicht korrekt funktionieren.

---

Hier finden Sie Informationen, wie Sie SELinux für RHEL/CentOS/Fedora deaktivieren können.

1. Konfigurieren Sie `SELINUX=disabled` in der Datei `etc/selinux/config`:

```
# 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. Starten Sie Ihr System neu. Vergewissern Sie sich nach dem Neustart, dass der Befehl `getenforce Disabled` zurückgibt:

```
root> getenforce
Disabled
```

## 3.2 Schritt 1: Anwendung entpacken und installieren

Sie haben nach dem Download entweder `otrs-x.y.z.tar.gz` oder `otrs-x.y.z.tar.bz2` erhalten. Entpacken Sie das Quellarchiv (zum Beispiel mit `tar`) in das Verzeichnis `/opt` und erstellen Sie einen symbolischen Link `/opt/otrs`, der auf `/opt/otrs-xyz` verweist. **Vergessen Sie nicht**, die Versionsnummern zu ersetzen!

---

**Bemerkung:** Das Paket `bzip2` ist in einigen Systemen standardmäßig nicht installiert. Stellen Sie sicher, dass `bzip2` installiert ist, bevor Sie `otrs-x.y.z.tar.bz2` entpacken.

---

Kommando zum Entpacken für `otrs-x.y.z.tar.gz`:

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

Kommando zum Entpacken für `otrs-x.y.z.tar.bz2`:

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

Es wird empfohlen, einen symbolischen Link mit dem Namen `/opt/otrs` zu erstellen, der immer auf die neueste OTRS-Version verweist. Die Verwendung eines symbolischen Links erleichtert die Verwaltung der OTRS-Updates, da Sie das Verzeichnis der vorherigen Version nicht ändern müssen. Lediglich der symbolische Link muss geändert werden. Wenn Sie das Update zurücksetzen müssen, können Sie das Ziel des symbolischen Links wieder ändern.

Führen Sie diesen Befehl aus, um einen symbolischen Link zu erstellen:

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

## 3.3 Schritt 2: Zusätzliche Programme und Perl-Module installieren

Verwenden Sie das folgende Skript, um einen Überblick über alle installierten und erforderlichen CPAN-Module und andere externe Abhängigkeiten zu erhalten.

```
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)
  ...
```

---

**Bemerkung:** Bitte beachten Sie, dass OTRS eine funktionierende Perl-Installation mit allen *core* Modulen wie dem Modul `version` erfordert. Diese Module werden vom Skript nicht explizit geprüft. Auf einigen Systemen wie RHEL, die die Perl-Core-Pakete standardmäßig nicht installieren, müssen Sie möglicherweise ein `perl-core`-Paket installieren.

---

Um die erforderlichen und optionalen Pakete zu installieren, können Sie entweder CPAN oder den Paketmanager Ihrer Linux-Distribution verwenden.

Führen Sie diesen Befehl aus, um einen Installationsbefehl zum Installieren der fehlenden Abhängigkeiten zu erhalten:

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

Für OTRS muss eine unterstützte stabile Version von Node.js installiert werden. Bitte beachten Sie die [Node.js installation instructions](#).

### 3.4 Schritt 3: Einen OTRS-Benutzer erstellen

Erstellen Sie einen dedizierten Benutzer für OTRS innerhalb der eigenen Gruppe:

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

### 3.5 Schritt 4: Standard-Konfigurationsdatei aktivieren

Es gibt eine gepackte OTRS-Konfigurationsdatei in `$OTRS_HOME/Kernel/Config.pm.dist`. Sie müssen sie aktivieren, indem Sie die Datei ohne die Dateinamenerweiterung `.dist` kopieren.

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

### 3.6 Schritt 5: Apache-Webserver konfigurieren

OTRS verfügt über einen eigenen integrierten Webserver, der hinter Apache als Reverse-Proxy (oder einem anderen Reverse-Proxy-Server) eingesetzt wird. Für den korrekten Betrieb werden einige Apache-Module benötigt:

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

Auf einigen Systemen wie Debian und SuSE müssen diese Module speziell aktiviert werden:

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

Die meisten Apache-Installationen enthalten ein `conf.d`-Verzeichnis. Auf Linux-Systemen finden Sie dieses Verzeichnis normalerweise unter `/etc/apache` oder `/etc/apache2`. Melden Sie sich als `root` an, wechseln Sie in das Verzeichnis `conf.d` und verknüpfen Sie die entsprechende Vorlage in `/opt/otrs/scripts/apache2-httpd.include.conf` mit einer Datei mit dem Namen `zzz_otrs.conf` im Apache-Konfigurationsverzeichnis (um sicherzustellen, dass es nach den anderen Konfigurationen geladen wird).

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

Jetzt können Sie Ihren Webserver neu starten, um die neuen Konfigurationseinstellungen zu laden. Auf den meisten Systemen können Sie dies mit diesem Befehl ausführen:

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

### 3.7 Schritt 6: Dateiberechtigungen festlegen

Führen Sie den folgenden Befehl aus, um die Datei- und Verzeichnisberechtigungen für OTRS festzulegen. Es wird versucht, die richtigen Gruppen- und Benutzereinstellungen zu ihrem Setup zu ermitteln.

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

### 3.8 Schritt 7: Die Datenbank einrichten

Die folgenden Schritte müssen ausgeführt werden, um die Datenbank für OTRS ordnungsgemäß einzurichten:

- Einen dedizierten Datenbank-Benutzer und eine Datenbank erstellen.
- Datenbank-Struktur erstellen.
- Initiale Daten einfügen.
- Datenbank-Verbindung in `Kernel/Config.pm` konfigurieren.

---

**Bemerkung:** Beachten Sie bitte, dass OTRS `utf8` zur Speichercodierung der Datenbank benötigt.

---

#### 3.8.1 MySQL oder MariaDB

Loggen Sie sich in die MySQL-Konsole als Datenbank Admin-Benutzer ein:

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

Datenbank erstellen:

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

Für MySQL 8 ist eine besondere Handhabung der Datenbank-Benutzer erforderlich, da das Standard-`caching_sha2_password` nur für sichere Verbindungen verwendet werden kann. Erstellen Sie einen Datenbank-Benutzer in MySQL 8:

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

Erstellen eines Datenbank-Benutzers in älteren MySQL Versionen:

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

Weisen Sie der neuen Datenbank Benutzerberechtigungen zu:



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

Führen Sie die folgenden Befehle in der Shell aus, um ein Schema zu erstellen und Daten einzufügen:

```
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
```

Konfigurieren Sie die Datenbank-Einstellungen in `Kernel/Config.pm`:

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

**Bemerkung:** Die folgenden Einstellungen zur Konfiguration werden für MySQL-Setups empfohlen. Fügen Sie bitte folgende Zeilen in `/etc/my.cnf` unterhalb des Abschnitts `[mysqld]` ein:

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

## 3.8.2 PostgreSQL

**Bemerkung:** Wir gehen davon aus, dass OTRS und PostgreSQL-Server auf demselben Rechner laufen und PostgreSQL die Authentifizierungsmethode *Peer* verwendet. Weitere Informationen finden Sie im Abschnitt [Client-Authentifizierung](#) im PostgreSQL-Handbuch.

Wechseln Sie zum Postgres-Benutzer:

```
root> su - postgres
```

Erstellen Sie einen Datenbank-Benutzer:

```
postgres> createuser otrs
```

Datenbank erstellen:

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

Führen Sie die folgenden Befehle in der Shell aus, um ein Schema zu erstellen und Daten einzufügen:

```
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
```

Konfigurieren Sie die Datenbank-Einstellungen in `Kernel/Config.pm`:

```
$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 Datenbank-Setup abschließen

Führen Sie den folgenden Befehl aus, um Ihr Datenbank-Setup zu überprüfen:

```
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.
```

Sobald die Datenbank korrekt konfiguriert ist, können Sie die Systemkonfiguration mit folgendem Befehl initialisieren:

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

**Bemerkung:** Ändern Sie bitte aus Sicherheitsgründen das Standardkennwort des Admin-Benutzers `root@localhost`, indem Sie ein zufälliges Passwort generieren:

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

Sie können auch Ihr eigenes Passwort festlegen:

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

## 3.9 Schritt 8: Elasticsearch-Cluster einrichten

OTRS erfordert einen aktiven Cluster von Elasticsearch 6.x (höhere Versionen werden nicht unterstützt). Der einfachste Weg ist, [Elasticsearch](#) auf demselben Host wie OTRS einzurichten und an seinen Standardport zu binden. Damit ist keine weitere Konfiguration in OTRS erforderlich.

Zusätzlich erfordert OTRS, dass Plugins in Elasticsearch installiert werden:

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

**Bemerkung:** Starten Sie Elasticsearch anschließend neu, da sonst keine Indizes aufgebaut werden.

Um die Installation von Elasticsearch zu überprüfen, können Sie den folgenden Befehl verwenden:

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

### 3.10 Schritt 9: OTRS Daemon und Webserver starten

Der neue OTRS-Daemon ist für die Behandlung asynchroner und wiederkehrender Aufgaben in OTRS verantwortlich. Der integrierte OTRS-Webserverprozess verarbeitet die von Apache übergebenen Webanforderungen. Beide Prozesse müssen vom `otrs`-Benutzer aus gestartet werden.

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

### 3.11 Schritt 10: Die erste Anmeldung

Jetzt können Sie sich unter <http://localhost/otrs/index.pl> als Benutzer `root@localhost` mit dem generierten Kennwort anmelden (siehe oben).

Verwenden Sie <http://localhost>, um auf das externe Interface zuzugreifen.

### 3.12 Schritt 11: Systemd-Dateien einrichten

OTRS enthält Beispiel Systemd-Konfigurationsdateien, mit denen sichergestellt werden kann, dass der OTRS-Daemon und der Webserver nach dem Systemstart automatisch gestartet werden.

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

Nach diesem Schritt ist das Basis-Systemsetup beendet.

### 3.13 Schritt 12: Automatische Bash-Vervollständigung einrichten (optional)

Alle regulären OTRS-Befehlszeilenoperationen finden über die OTRS-Konsolenschnittstelle statt. Dies bietet eine automatische Vervollständigung der Bash-Shell, wodurch das Finden der richtigen Befehle und

Optionen wesentlich vereinfacht wird.

Sie können die automatische Vervollständigung der Bash aktivieren, indem Sie das Paket `bash-completion` installieren. Das Feature erkennt und lädt automatisch die Datei `/opt/otrs/.bash_completion` für den `otrs`-Benutzer.

Nach dem Neustart der Shell können Sie einfach diesen Befehl gefolgt von TAB eingeben und es werden alle verfügbaren Befehle gelistet:

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

Wenn Sie einige Zeichen des Befehlsnamens eingeben, zeigt TAB alle übereinstimmenden Befehle an. Nach Eingabe eines vollständigen Befehls werden durch Drücken von TAB alle möglichen Optionen und Argumente angezeigt.

---

**Bemerkung:** Wenn Sie Probleme haben, können Sie folgende Zeile zu Ihrer `~/ .bashrc` hinzufügen, um die Befehle aus der Datei auszuführen.

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

---

### 3.14 Schritt 13: Weitere Informationen

Wir empfehlen Ihnen, das Kapitel OTRS *Leistungsverbesserung* zu lesen.

---

**Bemerkung:** Es wird empfohlen, zuerst ein Test-Update auf einem separaten Testsystem durchzuführen.

---

**Aktualisierung von einer früheren Version von OTRS 7** Sie können direkt von allen vorherigen Patch-Releases aktualisieren.

**Aktualisierung von OTRS 6** Sie können von jeder OTRS 6 Patch Version auf die neueste verfügbare OTRS 7 Patch Version aktualisieren.

**Aktualisierung von OTRS 5 oder früher** Sie können ein Update von OTRS 5 oder früheren Versionen auf OTRS 7 nicht direkt durchführen. Vollständige Updates für alle verfügbaren Minor Versionen müssen stattdessen nacheinander vorgenommen werden. Wenn Sie beispielsweise von OTRS 4 kommen, müssen Sie zuerst ein vollständiges Update auf OTRS 5, dann auf OTRS 6 und schließlich auf OTRS 7 durchführen.

**Siehe auch:**

Informationen zum Update finden Sie im Administratorhandbuch der vorherigen Versionen von OTRS.

### 4.1 Schritt 1: Alle relevanten Dienste und den OTRS Daemon beenden

Stellen Sie sicher, dass keine weiteren Dienste oder Cron-Jobs mehr vorhanden sind, die versuchen, auf OTRS zuzugreifen. Dies hängt von Ihrer Dienstkonfiguration und der OTRS-Version ab.

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

Wenn Sie ein Major Update von OTRS 6 durchführen, müssen Sie die alten OTRS-Cron-Jobs und den Daemon (in dieser Reihenfolge) stoppen:

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

Wenn Sie in OTRS 7 ein Update auf Patch-Ebene durchführen (mit den neuen systemd-Dateien), stoppen Sie die OTRS-Dienste über systemd:

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

## 4.2 Schritt 2: Dateien und Datenbank sichern

Erstellen Sie eine Sicherung der folgenden Dateien und Ordner:

- `Kernel/Config.pm`
- `Kernel/WebApp.conf` (nur bei Patch-Update von OTRS 7 und nur bei geänderter Datei)
- `var/*`
- ebenfalls die Datenbank

**Warnung:** Fahren Sie nicht ohne eine vollständige Sicherung Ihres Systems fort. Verwenden Sie dazu das Skript *Datensicherung*.

## 4.3 Schritt 3: Das neue Release installieren

**Bemerkung:** Mit OTRS 7 werden keine RPMs mehr bereitgestellt. RPM-basierte Installationen müssen geändert werden: Durch Deinstallieren des RPM (dabei wird Ihre Datenbank nicht gelöscht) und durch Verwendung der Quellarchive.

Sie haben nach dem Download entweder `otrs-x.y.z.tar.gz` oder `otrs-x.y.z.tar.bz2` erhalten. Entpacken Sie das Quellarchiv (zum Beispiel mit `tar`) in das Verzeichnis `/opt` und erstellen Sie einen symbolischen Link `/opt/otrs`, der auf `/opt/otrs-xyz` verweist. **Vergessen Sie nicht**, die Versionsnummern zu ersetzen!

**Bemerkung:** Das Paket `bzip2` ist in einigen Systemen standardmäßig nicht installiert. Stellen Sie sicher, dass `bzip2` installiert ist, bevor Sie `otrs-x.y.z.tar.bz2` entpacken.

Kommando zum Entpacken für `otrs-x.y.z.tar.gz`:

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

Kommando zum Entpacken für `otrs-x.y.z.tar.bz2`:

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

Es wird empfohlen, einen symbolischen Link mit dem Namen `/opt/otrs` zu erstellen, der immer auf die neueste OTRS-Version verweist. Die Verwendung eines symbolischen Links erleichtert die Verwaltung der OTRS-Updates, da Sie das Verzeichnis der vorherigen Version nicht ändern müssen. Lediglich der symbolische Link muss geändert werden. Wenn Sie das Update zurücksetzen müssen, können Sie das Ziel des symbolischen Links wieder ändern.

Führen Sie diesen Befehl aus, um einen symbolischen Link zu erstellen:

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

### 4.3.1 Alte Konfigurationsdateien wiederherstellen

- Kernel/Config.pm
- Kernel/WebApp.conf (nur bei Patch-Update von OTRS 7 und nur bei geänderter Datei)

### 4.3.2 Artikeldaten wiederherstellen

Wenn Sie OTRS so konfiguriert haben, dass Artikeldaten im Dateisystem gespeichert werden, müssen Sie den Ordner `article` nach `/opt/otrs/var/` (oder in dem in der Systemkonfiguration angegebenen Ordner) wiederherstellen.

### 4.3.3 Bereits installierte Standardstatistik wiederherstellen

Wenn Sie zusätzliche Pakete mit Standardstatistiken haben, müssen Sie die XML-Statistikdateien mit dem Suffix `*.installed` in `/opt/otrs/var/stats` wiederherstellen.

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

### 4.3.4 Dateiberechtigungen festlegen

Führen Sie den folgenden Befehl aus, um die Datei- und Verzeichnisberechtigungen für OTRS festzulegen. Es wird versucht, die richtigen Gruppen- und Benutzereinstellungen zu ihrem Setup zu ermitteln.

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

### 4.3.5 Erforderliche Programme und Perl-Module installieren

Bitte beachten Sie den Abschnitt [Schritt 2: Zusätzliche Programme und Perl-Module installieren](#) im Installationshandbuch, in dem erläutert wird, wie externe Abhängigkeiten wie Perl-Module und Node.js überprüft werden.

Darüber hinaus erfordert OTRS 7 auch einen aktiven Cluster von Elasticsearch 6.x (höhere Versionen werden nicht unterstützt). Weitere Informationen finden Sie in der Installationsanleitung im Abschnitt: [Schritt 8: Elasticsearch-Cluster einrichten](#).

## 4.4 Schritt 4: Migrationsskript ausführen

Das Migrationsskript führt viele Überprüfungen auf Ihrem System durch und gibt Ihnen Hinweise, wie Sie fehlende Perl-Module usw. installieren, falls dies erforderlich ist. Wenn alle Prüfungen erfolgreich waren, werden die erforderlichen Migrationsschritte ausgeführt. Bitte führen Sie dieses Skript auch für Updates auf Patch-Ebene aus.

Migrationsskript ausführen:

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

**Warnung:** Setzen Sie den Aktualisierungsvorgang nicht fort, wenn dieses Skript nicht ordnungsgemäß für Sie funktioniert hat. Andernfalls können Fehlfunktionen oder Datenverlust auftreten.

Das Migrationsskript überprüft auch, ob ACLs und Systemkonfigurationseinstellungen korrekt sind. Im Falle einer ungültigen Systemkonfigurationseinstellung bietet das Skript eine Möglichkeit zur Korrektur, indem es aus einer Liste möglicher Werte ausgewählt wird. Wenn das Skript in einem nicht interaktiven Modus ausgeführt wird, werden ungültige Einstellungen automatisch korrigiert. Wenn dies fehlschlägt, werden Sie aufgefordert, die Einstellung nach der Migration manuell zu aktualisieren.

Wenn veraltete ACLs vorhanden sind, kann das System sie nicht automatisch korrigieren. Sie müssen vom Administrator korrigiert werden. Weitere Informationen finden Sie im letzten Schritt für manuelle Änderungen.

### 4.5 Schritt 5: Installierte Pakete aktualisieren

---

**Bemerkung:** Pakete für OTRS 6 sind nicht kompatibel mit OTRS 7 und müssen aktualisiert werden.

---

Sie können den folgenden Befehl verwenden, um alle installierten Pakete zu aktualisieren. Dies funktioniert für alle Pakete, die in Online-Repositorys verfügbar sind. Sie können andere Pakete später über den Paketmanager aktualisieren (dies erfordert einen laufenden OTRS-Daemon).

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

### 4.6 Schritt 6: Ihre Dienste starten

OTRS 7 wird mit einem eigenen Webserver geliefert, der hinter Apache als Reverse-Proxy (oder einen anderen Reverse-Proxy-Server) verwendet wird. Für ein Major Update von OTRS 6 muss die Apache-Konfiguration mit der neuen Version in `/opt/otrs/scripts/apache2-httpd.include.conf` aktualisiert werden, wenn sie kopiert und nicht nur verlinkt wurde.

Bitte beachten Sie auch, dass während `mod_perl` nicht mehr benötigt wird, jetzt andere Apache-Module benötigt werden:

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

Danach können die Dienste gestartet werden. Dies hängt von Konfiguration Ihrer Dienste ab. Hier ein Beispiel:

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



**Bemerkung:** Der OTRS Daemon ist für den korrekten Betrieb von OTRS erforderlich, beispielsweise für das Senden von E-Mails. Bitte aktivieren Sie den Daemon wie im nächsten Schritt beschrieben.

## 4.7 Schritt 7: OTRS-Daemon und Webserver starten

Der OTRS-Daemon ist für die Behandlung asynchroner und wiederkehrender Aufgaben in OTRS verantwortlich. Der integrierte OTRS-Webserverprozess verarbeitet die von Apache übergebenen Webanforderungen. Beide Prozesse müssen vom `otrs`-Benutzer aus gestartet werden.

OTRS enthält Beispiel Systemd-Konfigurationsdateien, mit denen sichergestellt werden kann, dass der OTRS-Daemon und der Webserver nach dem Systemstart automatisch gestartet werden.

```
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
```

Jetzt können Sie sich in Ihrem System anmelden.

## 4.8 Schritt 8: Manuelle Migrationsaufgaben und -änderungen

**Warnung:** Dieser Schritt ist nur für ein Major Update von OTRS 6 erforderlich.

Da die alten Bildschirmmasken der Kundenoberfläche nicht mehr vorhanden sind, müssen einige ACLs vom Administrator manuell korrigiert werden. Das Migrationsskript hat Sie bereits informiert, wenn dies der Fall ist.

Betroffene ACLs sind solche, die sich auf eine nicht vorhandene Bildschirmmaske für die Kundenoberfläche in ihrer „Action“-Einstellung beziehen. Diese Frontend-Regel muss durch eine entsprechende Endpunkt-Regel ersetzt werden. Eine Tabelle mit möglicher Zuordnung finden Sie nachfolgend.

Aktion	Endpunkt
CustomerTicket-Print	Kein Ersatz (Funktion entfällt)
CustomerTicket-Zoom	ExternalFrontend::TicketDetailView
CustomerTicket-Process	ExternalFrontend::ProcessTicketCreate <b>oder</b> ExternalFrontend::ProcessTicketNextStep
CustomerTicketMessage	ExternalFrontend::TicketCreate



---

## Sichern und Wiederherstellen

---

OTRS verfügt über integrierte Skripte zum Sichern und Wiederherstellen. Führen Sie die Skripte mit der Option `-h` aus, um weitere Informationen zu erhalten.

### 5.1 Datensicherung

---

**Bemerkung:** Um eine Datensicherung zu erstellen, werden Schreibrechte für den `otrs` Benutzer im Zielverzeichnis benötigt.

---

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

Die Ausgabe des Skripts:

```
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:
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

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

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 Wiederherstellen

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

Die Ausgabe des Skripts:

```
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.
```

Es gibt eine Liste mit leistungsverbessernden Techniken für Ihre OTRS-Installation, einschließlich Konfiguration, Codierung, Speichernutzung und mehr.

### 6.1 Ticket-Index-Modul

Das Ticket-Index-Modul kann in der Systemkonfigurationseinstellung `Ticket::IndexModule` eingestellt werden. Es gibt zwei Backend-Module für den Index der Ticket-Queueansicht:

**Kernel::System::Ticket::IndexAccelerator::RuntimeDB** Das ist die Standardoption und generiert jede Queueansicht direkt aus der Ticket-Tabelle. Bei bis zu ca. 60.000 in Ihrem System offenen Tickets werden Sie damit keine Leistungsprobleme haben.

**Kernel::System::Ticket::IndexAccelerator::StaticDB** Das leistungsstärkste Modul sollte verwendet werden, wenn Sie über 80.000 offene Tickets haben. Es verwendet eine zusätzliche `ticket_index`-Tabelle, die mit auf Ticketdaten basierenden Schlüsselwörtern aufgefüllt wird. Verwenden Sie den folgenden Befehl, um einen Anfangsindex nach dem Wechsel der Backends zu erstellen:

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

### 6.2 Ticket-Suchindex

OTRS verwendet einen speziellen Suchindex, um Volltextsuchen über Felder in Artikeln aus verschiedenen Kommunikationskanälen durchzuführen.

Verwenden Sie diesen Befehl, um einen Anfangsindex zu erstellen:

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

**Bemerkung:** Die eigentliche Artikelindizierung erfolgt über einen OTRS-Daemon-Job im Hintergrund. Weil die Artikel, die gerade dem System hinzugefügt wurden, sofort für die Indizierung markiert werden, kann es sein, dass ihr Index erst innerhalb einiger Minuten verfügbar ist.

Es gibt einige Optionen zur Feinabstimmung des Suchindex:

**Ticket::SearchIndex::IndexArchivedTickets** Legt fest, ob archivierte Tickets in den Suchindex aufgenommen werden (standardmäßig nicht aktiviert). Dies ist ratsam, um den Index bei großen Systemen mit archivierten Tickets klein zu halten. Wenn diese Option aktiviert ist, werden archivierte Tickets durch Volltextsuche gefunden.

**Ticket::SearchIndex::Attribute** Grundlegende Einstellungen für den Volltextindex.

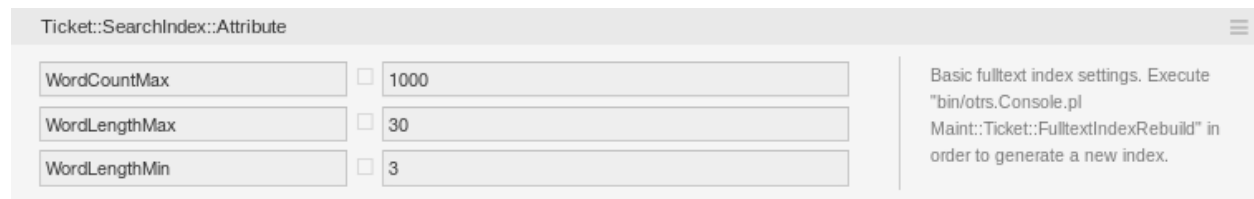


Abb. 1: Ticket::SearchIndex::Attribute Einstellung

**Bemerkung:** Führen Sie das folgende Kommando aus, um einen neuen Index zu generieren:

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

**WordCountMax** Definiert die maximale Anzahl von Wörtern, die zum Aufbau des Index verarbeitet werden. Zum Beispiel, dass nur die ersten 1000 Wörter des Artikelkörpers im Artikelsuchindex gespeichert werden.

**WordLengthMin und WordLengthMax** Wird als Wortlängenbegrenzung verwendet. Nur Wörter mit einer Länge zwischen diesen beiden Werten werden im Artikelsuchindex gespeichert.

**Ticket::SearchIndex::Filters** Reguläre Ausdrücke für den Volltextindex-Filter, um Teile des Textes zu entfernen.

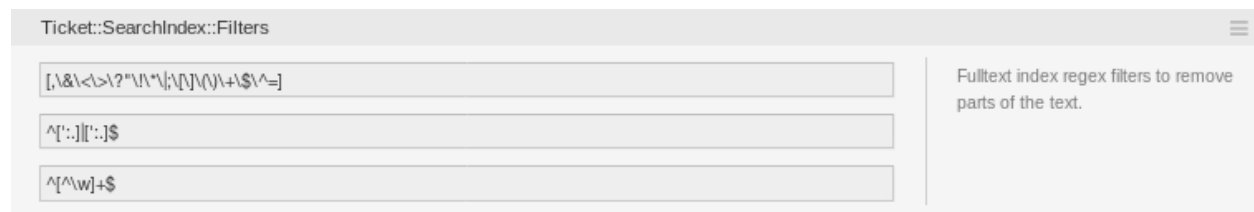


Abb. 2: Ticket::SearchIndex::Filters Einstellung

Es sind drei Standardfilter definiert:

- Der erste Filter entfernt Sonderzeichen wie: , & < > ? \* | ; [ ] ( ) + \$ ^ =
- Der zweite Filter entfernt Wörter die mit einem der folgenden Zeichen beginnen oder enden: , : .
- Der dritte Filter entfernt Wörter, die kein Wortzeichen enthalten: a-z, A-Z, 0-9, \_

Ticket::SearchIndex::StopWords###en

a

about

above

after

again

against

all

am

English stop words for fulltext index. These words will be removed from the search index.

Abb. 3: Ticket::SearchIndex::StopWords###en Einstellung

**Ticket::SearchIndex::StopWords** Englische Stopwörter für den Volltextindex. Diese Wörter werden aus dem Suchindex entfernt.

Für einige Sprachen sind sogenannte Stopwörter definiert. Diese Stopwörter werden beim Erstellen des Suchindex übersprungen.

**Siehe auch:**

Wenn Ihre Sprache nicht in den Systemkonfigurationseinstellungen enthalten ist oder Sie weitere Wörter hinzufügen möchten, können Sie diese mit dieser Einstellung hinzufügen:

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

## 6.3 Dokumentensuche

OTRS verwendet Elasticsearch für seine Dokumentensuche. Für eine gute Einführung in die Konzepte, die Installation und den Gebrauch von Elasticsearch folgen Sie bitte dem [Getting Started Guide](#).

### 6.3.1 Heap-Größe

Elasticsearch ist in Java geschrieben und läuft daher in einer Java Virtual Machine (JVM) auf jedem Clusterknoten. Eine solche JVM verwendet einen Teil des Speichers, genannt *heap*, dessen Größe in der Konfigurationsdatei `jvm.options` konfiguriert werden kann.

Die Heap-Minimum- und Maximum-Konfigurationen sind standardmäßig auf einen Wert von 1 GB eingestellt und können mit den folgenden Optionen geändert werden:

- `Xms1g`: Minimale Heap-Größe.
- `Xmx1g`: Maximale Heap-Größe.

Wenn der `Xms` einen niedrigeren Wert als `Xmx` hat, wird die JVM den verwendeten Heap bei jeder Überschreitung des aktuellen Limits skalieren, bis der Wert von `Xmx` erreicht ist. Eine solche Größenänderung bewirkt, dass der Dienst bis zum Ende pausiert, was die Geschwindigkeit und Reaktivität der Such- oder Indexierungsaktionen verringern kann. Daher wird dringend empfohlen, diese Konfigurationen auf einen gleichen Wert einzustellen.

**Warnung:** Wenn die maximale Heap-Größe überschritten wird, stoppt der zugehörige Clusterknoten seine Arbeit und kann den Dienst sogar herunterfahren.

Je höher der Heap-Maximalwert eingestellt ist, desto mehr Speicherplatz kann von Elasticsearch genutzt werden, was auch die möglichen Pausen für die Garbage Collection der JVM erhöht. Daher wird empfohlen, einen Wert für  $X_{mx}$  einzustellen, der nicht höher als 50% des physikalischen Speichers ist.

Für weitere Informationen und gute Faustregeln zur Größe des Heaps beachten Sie bitte die offizielle Dokumentation <<https://www.elastic.co/guide/en/elasticsearch/reference/current/heap-size.html>>‘\_\_.

### 6.3.2 Festplattenzuordnung

Während der Laufzeit des Dienstes überprüft Elasticsearch den verfügbaren Festplattenspeicher und entscheidet somit, ob dem zugehörigen Clusterknoten neue Shards zugewiesen oder sogar Shards von diesem speziellen Knoten entfernt werden. Dieses Verhalten wird durch die aktuelle Festplattenkapazität gesteuert und kann in der Konfigurationsdatei `elasticsearch.yml` konfiguriert werden. Anbei einige wichtige Konfigurationen, die mit guten Standardwerten ausgestattet sind, aber wichtig sein können:

**cluster.routing.allocation.disk.watermark.low** Standardwert von 85%. Wenn diese Grenze überschritten wird, weist Elasticsearch dem zugehörigen Clusterknoten nicht mehr Shards zu. Der Betrieb dieses Knotens wird nicht beeinflusst und die Daten können weiterhin indiziert und durchsucht werden.

**cluster.routing.allocation.disk.watermark.high** Standardwert von 90%. Wird diese Grenze überschritten, versucht Elasticsearch, bestehende Shards auf andere Knoten zu verschieben (wenn möglich), die genügend Platz haben.

**cluster.routing.allocation.disk.watermark.flood\_stage** Standardwert von 95%. Wird diese Grenze überschritten, aktualisiert Elasticsearch die Konfiguration aller Indizes auf schreibgeschützte Indexblöcke `index.blocks.read_only_allow_delete`, die mindestens einen Shard dem zugehörigen Clusterknoten zugeordnet haben. Seitdem ist es nicht mehr möglich, neue Daten auf solche Indizes zu indizieren und sich auf Such- und Löschkaktionen zu beschränken.

---

**Bemerkung:** Wenn der Speicherplatz überschritten wurde und bestimmte Indizes für den Nur-Lese-Modus konfiguriert sind, wird diese Konfiguration *nicht* automatisch von Elasticsearch geändert. Wenn die zugehörigen Laufwerke wieder genügend freien Speicherplatz enthalten, ist es aufgrund manueller Aktionen erforderlich, die Konfiguration manuell wieder in den Normalmodus zurückzusetzen.

---

Weitere Informationen zu den *Watermarks* der Festplatte und der plattenbasierten Shardzuweisung finden Sie in der offiziellen Dokumentation <https://www.elastic.co/guide/en/elasticsearch/reference/current/disk-allocator.html>.

## 6.4 Artikelspeicherung

Es gibt zwei verschiedene Backend-Module für die Artikelspeicherung von Telefon-, E-Mail- und internen Artikeln. Der verwendete Artikelspeicher kann in der Einstellung `Ticket::Article::Backend::MIMEBase::ArticleStorage` konfiguriert werden.

**Kernel::System::Ticket::Article::Backend::MIMEBase::ArticleStorageDB** Dieses Standardmodul speichert Anlagen in der Datenbank. Es funktioniert auch mit mehreren Frontend-Servern, erfordert jedoch viel Speicherplatz in der Datenbank.



---

**Bemerkung:** Verwenden Sie dies nicht bei großen Setups.

---

**Kernel::System::Ticket::Article::Backend::MIMEBase::ArticleStorageFS** Verwenden Sie dieses Modul, um Anlagen im lokalen Dateisystem zu speichern. Dies ist zwar schnell, aber wenn Sie über mehrere Frontend-Server verfügen, müssen Sie sicherstellen, dass das Dateisystem von den Servern gemeinsam genutzt wird. Legen Sie es auf eine NFS-Freigabe oder vorzugsweise auf ein SAN oder eine ähnliche Lösung.

---

**Bemerkung:** Empfohlen für große Setups.

---

Sie können zügig von einem Backend zum anderen wechseln. Sie können das Backend in der Systemkonfiguration wechseln und dann dieses Befehlszeilendienstprogramm ausführen, um die Artikel aus der Datenbank in das Dateisystem zu laden oder umgekehrt:

```
otrs> /opt/otrs/bin/otrs.Console.pl Admin::Article::StorageSwitch --target_
↪ArticleStorageFS
```

Sie können die Option `--target` verwenden, um das Ziel-Backend festzulegen.

---

**Bemerkung:** Der gesamte Prozess kann einige Zeit in Anspruch nehmen, abhängig von der Anzahl der Artikel, der verfügbaren CPU-Leistung und / oder der Netzwerkkapazität.

---

Wenn Sie alte Anhänge in der Datenbank behalten möchten, können Sie die Systemkonfigurationsoption `Ticket::Article::Backend::MIMEBase::CheckAllStorageBackends` aktivieren, um sicherzustellen, dass OTRS diese weiterhin findet.

## 6.5 Tickets archivieren

Da OTRS als revisionssicheres System betrieben werden kann, ist das Löschen von geschlossenen Tickets möglicherweise nicht empfehlenswert. Daher haben wir eine Funktion implementiert, mit der Sie Tickets archivieren können.

Tickets, die bestimmten Kriterien entsprechen, können als archiviert markiert werden. Auf diese Tickets wird nicht zugegriffen, wenn Sie eine reguläre Ticketsuche durchführen oder einen generischen Agentenjob ausführen. Das System selbst muss nicht mehr mit einer großen Anzahl von Tickets umgehen, da bei der Verwendung von OTRS nur die neuesten Tickets berücksichtigt werden. Dies kann bei großen Systemen zu einem enormen Leistungszuwachs führen.

So verwenden Sie die Archivfunktion:

1. Aktivieren Sie die Einstellung `Ticket::ArchiveSystem` in der Systemkonfiguration.
2. Definieren Sie einen `GenericAgent`-Job:
  - Klicken Sie im Bildschirm *GenericAgent* auf die Schaltfläche *Job hinzufügen*.
  - *Job-Einstellungen*: Geben Sie einen Namen für den Archivierungsjob an.
  - *Automatische Ausführung*: Wählen Sie die richtigen Optionen aus, um diesen Job zu planen.
  - *Tickets selektieren*: Es kann eine gute Idee sein, nur die Tickets in einem geschlossenen Zustand zu archivieren, die einige Monate zuvor geschlossen wurden.

- *Ticketattribute aktualisieren / hinzufügen*: Setzen Sie das Feld *Ausgewählte Tickets archivieren* auf *Tickets archivieren*.
- Speichern Sie den Job am Ende der Seite.
- Klicken Sie in der Übersichtstabelle auf den Link *Diese Aufgabe ausführen*, um die betroffenen Tickets anzuzeigen.
- Klicken Sie auf die Schaltfläche *Diesen Job ausführen*.

---

**Bemerkung:** Bis zu 5000 Tickets können durch manuelle Ausführung dieses Jobs geändert werden.

---

Bei der Suche nach Tickets werden standardmäßig nicht archivierte Tickets durchsucht.

So suchen Sie nach archivierten Tickets:

1. Öffnen Sie die Ticketsuche.
2. Setzen Sie *Archivsuche* auf *Nicht archivierte Tickets* oder *Alle Tickets*.
3. Führen Sie die Suche durch.

## 6.6 Webserver optimieren

Der integrierte Webserver von OTRS kann kleine und mittlere Setups sofort ausführen. Wenn OTRS von vielen Benutzer gleichzeitig genutzt wird, kann es erforderlich sein, die Webserverkonfiguration zu optimieren, um beispielsweise die Anzahl der Arbeitsprozesse zu erhöhen.

Die Konfigurationsdatei des Webservers befindet sich in `Kernel/WebApp.conf`, und alle Einstellungen sind dokumentiert. Die Einstellung für `Worker` kann erhöht werden, um mehr Prozesse für die Bereitstellung von HTTP-Anforderungen auf fähigen Servern bereitzustellen.

## 6.7 Caching

OTRS speichert viele temporäre Daten in `/opt/otrs/var/tmp`. Stellen Sie sicher, dass dieses Verzeichnis ein Hochleistungs-Dateisystem und -speicher verwendet. Wenn Sie über genügend RAM verfügen, können Sie auch versuchen, dieses Verzeichnis auf einer RAM-Disk wie folgt abzulegen:

```
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
```

---

**Bemerkung:** Fügen Sie einen dauerhaften Einhängpunkt in `/etc/fstab` hinzu.

---

**Warnung:** Dies ist ein nicht permanenter Speicher, der bei einem Neustart des Servers verloren geht. Alle Ihre Sitzungen (wenn Sie sie im Dateisystem speichern) und Ihre Cache-Daten gehen verloren.

**Siehe auch:**

Es gibt auch ein zentrales, auf [Memcached-basierendes Cache-Backend](#), das von der OTRS Group erworben werden kann.

## 6.8 Clustering

Bei sehr hohen Lasten kann es erforderlich sein, OTRS in einem Cluster mehrerer Frontend-Server zu betreiben. Dies ist eine komplexe Aufgabe mit vielen Fallstricken. Daher bietet die OTRS Group ausschließlich Unterstützung für Cluster in ihrer [managed OTRS-Umgebung](#).