**OTRS INSTALLATION AND MIGRATION**

**Otrs 5.0.10 installation on centos 6.5**

**PRE REQUISTICS**

**Step 1 :- Enable the epel (extra packages for enterprise linux) repository.**

**sudo yum install epel-release**

**yum update**

**Error: Cannot retrieve metalink for repository:**

**Note *: I solved this issue editing both /etc/yum.repos.d/epel.repo and /etc/yum.repos.d/epel-testing.repo files, commenting all entries starting with mirrorlist= and uncommenting all the entries starting with baseurl=***

**Step 2 :- Disable Selinux & Stop Firewall :**

**vi /etc/selinux/config**

**SELINUX=disabled**

**Save & Quit (:wq) and reboot**

**Step 3 :- Check ip tables**

**service iptables stop**

**chkconfig iptables off**

**Note: If Iptables service is on, then Adjust iptables to allow Apache Default Port 80.**

**vi /etc/sysconfig/iptables**

**Add the Following Line in Filter Table :**

**-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT**

**Save & Quit (:wq)**

**Service iptables restart**

**Step 4 :- Reboot the System : init 6**

**Save & Quit (:wq)**

**Steps**

**Log in as root user**

1. **Install Apache Web Server :**

*yum -y install httpd httpd-devel mod\_perl*

1. **Start Apache Service :**

*service httpd start*

*chkconfig httpd on*

1. **Install mysql -server**

*yum install mysql-server*

*Start mysql service as shown below:*

*service mysqld start*

*chkconfig mysqld on*

**By default, Mysql root user is blank. So let us setup the root user password as shown below:**

**mysql\_secure\_installation**

**NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL**

**SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!**

**In order to log into MySQL to secure it, we'll need the current**

**password for the root user.  If you've just installed MySQL, and**

**you haven't set the root password yet, the password will be blank,**

**so you should just press enter here.**

**Enter current password for root (enter for none): -----> Press Enter**

**OK, successfully used password, moving on...**

**Setting the root password ensures that nobody can log into the MySQL**

**root user without the proper authorisation.**

**You already have a root password set, so you can safely answer 'n'.**

**Change the root password? [Y/n] y -----> Enter Y and press Enter**

**New password: -----> Enter the new password**

**Re-enter new password: -----> Enter password again**

**Password updated successfully!**

**Reloading privilege tables..**

**... Success!**

**By default, a MySQL installation has an anonymous user, allowing anyoneto log into MySQL without having to have a user account created for them.  This is intended only for testing, and to make the installation go a bit smoother.  You should remove them before moving into a production environment.**

**Remove anonymous users? [Y/n] y -----> Press Enter**

**... Success!**

**Normally, root should only be allowed to connect from 'localhost'.  This**

**ensures that someone cannot guess at the root password from the network.**

**Disallow root login remotely? [Y/n] y -----> Press Enter**

**... Success!**

**By default, MySQL comes with a database named 'test' that anyone can**

**access.  This is also intended only for testing, and should be removed**

**before moving into a production environment.**

**Remove test database and access to it? [Y/n] y -----> Press Enter**

**- Dropping test database...**

**ERROR 1008 (HY000) at line 1: Can't drop database 'test'; database doesn't exist**

**... Failed!  Not critical, keep moving...**

**- Removing privileges on test database...**

**... Success!**

**Reloading the privilege tables will ensure that all changes made so far**

**will take effect immediately.**

**Reload privilege tables now? [Y/n] y -----> Press Enter**

**... Success!**

**Cleaning up...**

**All done!  If you've completed all of the above steps, your MySQL**

**installation should now be secure.**

**Thanks for using MySQL!**

MySQL has been installed successfully with default options. In order to work properly with OTRS, we have to add the following directives into the my.cnf file.

**vim /etc/my.cnf**

**[mysqld]**

*character-set-server=utf8*

*max\_allowed\_packet = 500M*

*query\_cache\_size =32M*

*innodb\_log\_file\_size =256M*

*datadir=/var/lib/mysql*

*socket=/var/lib/mysql/mysql.sock*

Save and close the file.

Next, restart mysql service using the following command:

**service mysqld restart**

**Create Database For OTRS :**

* *mysql -u root -p*
* *Enter Password:ccs#1234*
* *mysql> create database otrsdb;*
* *mysql> grant all privileges on otrsdb.\* to otrsuser@localhost identified by 'ccs#1234';*
* *mysql> flush privileges;*
* *mysql> \q*

**Download & Install OTRS RPM Package :**

* cd /mnt

wget http://ftp.otrs.org/pub/otrs/RPMS/rhel/6/otrs-5.0.10-01.noarch.rpm

yum -y install otrs-5.0.10-01.noarch.rpm

or

* download otrs tar or gz file as per requirement and archive it in /opt

Set ownership(chown otrs:apache) and proper permissions.

Because OTRS is written in Perl, it uses a number of Perl modules. We can check for missing modules by using the CheckModules.pl script included with OTRS.

* sudo /opt/otrs/bin/otrs.CheckModules.pl

You can install the missing modules with the yum commands provided in the output.Use the command below.

yum -y install "perl(Crypt::Eksblowfish::Bcrypt)" "perl(DBD::ODBC)" "perl(DBD::Pg)" "perl(Encode::HanExtra)" "perl(GD)" "perl(GD::Text)" "perl(GD::Graph)" "perl(JSON::XS)""perl(Mail::IMAPClient)" "perl(PDF::API2)" "perl(Text::CSV\_XS)" "perl(YAML::XS)"

**Note**:- Perl modules installations via cpan or make

refer

[perl module via cpan](https://www.linuxcloudvps.com/blog/how-to-install-perl-modules-using-cpan-on-linux/)

[manage multiple perl installations in your $HOME directory.](https://perlbrew.pl/)

* **Create OTRS User (In case of untaring otrs.x.x.x.tar.gz)**

Create user:

shell> useradd -d /opt/otrs -c 'OTRS user' otrs

Add user to webserver group (if the webserver is not running as the OTRS user):

shell> usermod -G www otrs

(SUSE=www, Red Hat/CentOS/Fedora=apache, Debian/Ubuntu=www-data)

* **Activate Default Config File**

There is one OTRS config file bundled in $OTRS\_HOME/Kernel/Config.pm.dist. You must activate it by copying it without the ".dist" filename extension.

shell> cp /opt/otrs/Kernel/Config.pm.dist /opt/otrs/Kernel/Config.pm

* **Check if all needed modules are installed**

shell> perl -cw /opt/otrs/bin/cgi-bin/index.pl

/opt/otrs/bin/cgi-bin/index.pl syntax OK

shell> perl -cw /opt/otrs/bin/cgi-bin/customer.pl

/opt/otrs/bin/cgi-bin/customer.pl syntax OK

shell> perl -cw /opt/otrs/bin/otrs.Console.pl

/opt/otrs/bin/otrs.Console.pl syntax OK

* **File Permissions**

Please execute the following command as root user to set the file and directory permissions for OTRS. It will try to detect the correct user and group settings needed for your setup.

shell> cd /opt/otrs/

shell> bin/otrs.SetPermissions.pl

* **Now Install OTRS Through Web Browser :**

Eg :http://192.168.0.121/otrs/installer.pl

* + - * Click on "Next".
      * Click on "Accept License & Continue".
      * Database Selection:
      * Type: MySQL
      * Install Type: Use an existing database for OTRS
      * Click on "Next".
      * Configure MySQL:
      * User: otrsuser
      * Password: password(ccs#1234)
      * Host: 127.0.0.1
      * Database Name: otrsdb
      * Click on "Check Database Settings".
      * Click on "Next".
      * Click on "Next".
      * System Settings:
      * AdminEmail: koushik@domain.com
      * Organization: Your Organization
      * Click on "Next".
      * Mail Configuration: For Now Skip this Step.
* **Now Login OTRS Admin Panel :**

*http://192.168.0.122/otrs/installer.pl*

*User :root@localhost*

*Password :mK5HutlLGmcMCcv6*

*Click on "Login".*

* **To Resolve OTRS Daemon is not running, Do the Following :**

*# useradd otrs*

*# chmod -Rf 777 /opt/otrs/Kernel/Config/Files/*

*# su - otrs*

*$ /opt/otrs/bin/otrs.Daemon.pl start*

*$ /opt/otrs/bin/Cron.sh start*

* **To Change Admin Password :**

http://192.168.0.122/otrs/index.pl

User: root@localhost

Password: LvMiOnXJAu2U7Vg6

* Click on "Edit Personal Peferences"
* Change Password:
* Current password: LvMiOnXJAu2U7Vg6
  + - New Password: Passw0rd@123
    - Verify Password: Passw0rd@123
* Click on "Update".

*To reset the root@localhost on OTRS 5 it´s necessary to get access to the server’s shell and execute the following command:*

***su -c "/opt/otrs/bin/otrs.Console.pl Admin::User::SetPassword root@localhost 123456" -s /bin/bash otrs***

*Now the new root password is 123456. This command is also used to reset the password of any other Agent, just overwrite “root@localhost” with the Agent’s login.*

***Setup bash autocompletion (optional)***

All regular OTRS commandline operations happen via the otrs Console interface bin/otrs.Console.pl. This provides an auto completion for the bash shell which makes finding the right command and options much easier.

You can activate the bash autocompletion by installing the package bash-completion. It will automatically detect and load the file /opt/otrs/.bash\_completion for the otrs user.

After restarting your shell, you can just type bin/otrs.Console.pl followed by TAB, and it will list all available commands. If you type a few characters of the command name, TAB will show all matching commands. After typing a complete command, all possible options and arguments will be shown by pressing TAB.

**OTRS 5.0.10 MIGRATION FROM CENTOS 6.5 TO CENTOS 8**

The previous server will be called centos 6.5, the destination server centos 8 in the following guide.

OTRS has its database on a MySQL server running on localhost.

**1. Stopping OTRS on CENTOS 6.5**

To not risk that during the transfer new tickets arrive or that an agent works on a ticket, OTRS should be stopped on CENTOS 6.5.

You can do this by stopping the cron service and Apache (or by simply disable the virtual host used for OTRS):

**/*etc/init.d/cron stop; /etc/init.d/apache2 stop***

**2. Create backup + transfer data on OLDSERVER**

You should now have a consistent state of your OTRS installation on CENTOS 6.5 so you can do a proper backup.

First the OTRS database needs to be dumped, then transferred to CENTOS 8:

**mysqldump –u root –p otrs > /tmp/otrs.sql; scp /tmp/otrs.sql root@NEWSERVER:/tmp/**

**either use winscp for database backup to centos 8**

Note that some OTRS databases can be huge, so it might make sense to zip/gzip/bzip the dump before the transfer.

Then the complete folder /opt/otrs (or wherever you have your OTRS installation installed) can be transferred to the new server.

I chose to pack the whole /opt/otrs into a tar.gz file to keep the file permissions and ownerships and then copy this file with scp:

**tar -czf /tmp/otrs.tar.gz /opt/otrs; scp /tmp/otrs.tar.gz root@NEWSERVER:/tmp/**

***OR BY BACKUP AND RESTORE METHOD***

***Backup***

*There are two types of data to backup: application files (e.g. the files in /opt/otrs), and the data stored in the database.*

*To simplify backups, the script scripts/backup.pl is included with every OTRS installation. It can be run to backup all important data (see Script below).*

*linux:/opt/otrs# cd scripts/*

*linux:/opt/otrs/scripts# ./backup.pl --help*

*backup.pl - backup script*

*Copyright (C) 2001-2019 OTRS AG, https://otrs.com/*

*usage: backup.pl -d /data\_backup\_dir/ [-c gzip|bzip2] [-r 30] [-t fullbackup|nofullbackup|dbonly]*

*linux:/opt/otrs/scripts#*

*Script: Getting help about the OTRS backup mechanism****.***

***Execute the command specified in the script below to create a backup:***

***linux:/opt/otrs/scripts# ./backup.pl -d /backup/***

***Backup /backup//2010-09-07\_14-28/Config.tar.gz ... done***

***Backup /backup//2010-09-07\_14-28/Application.tar.gz ... done***

***Dump MySQL rdbms ... done***

***Compress SQL-file... done***

***linux:/opt/otrs/scripts#***

***Script: Creating a backup.***

*All data was stored in the directory /backup/2010-09-07\_14-28/ (see Script below). Additionally, the data was saved into a .tar.gz file.*

***linux:/opt/otrs/scripts# ls /backup/2010-09-07\_14-28/***

***Application.tar.gz Config.tar.gz DatabaseBackup.sql.gz***

***linux:/opt/otrs/scripts***

***Restore***

*To restore a backup, the saved application data has to be written back into the installation directory, e.g. /opt/otrs. Also the database has to be restored*.

A script scripts/restore.pl (see Script below), which simplifies the restore process, is shipped with every OTRS installation. It supports MySQL and PostgreSQL.

*linux:/opt/otrs/scripts# ./restore.pl --help*

*restore.pl - restore script*

*Copyright (C) 2001-2019 OTRS AG, https://otrs.com/*

*usage: restore.pl -b /data\_backup/<TIME>/ -d /opt/otrs/*

*linux:/opt/otrs/scripts#*

*Script: Getting help about the restore mechanism*.

Data that is stored, for example, in the directory /backup/2010-09-07\_14-28/, can be restored with the command specified in the script below, assuming the OTRS installation is at /opt/otrs.

***linux:/opt/otrs/scripts# ./restore.pl -b /backup/2010-09-07\_14-28 -d /opt/otrs/***

***Restore /backup/2010-09-07\_14-28//Config.tar.gz ...***

***Restore /backup/2010-09-07\_14-28//Application.tar.gz ...***

***create MySQL***

***decompresses SQL-file ...***

***cat SQL-file into MySQL database***

***compress SQL-file...***

***linux:/opt/otrs/scripts#***

**3. Prepare stuff on NEWSERVER**

OTRS needs to run under its own system user, so this one needs to be created on CENTOS 8:

**useradd -r -d /opt/otrs -c 'OTRS user' -u 1001 otrs**

Make sure you use the same UID as used on OLDSERVER, if possible.

The new database (and database user) also needs to be created on NEWSERVER:

**mysql> create database otrs;**

**mysql> grant all on otrs.\* to 'otrs'@'localhost' identified by 'somePass';**

In case you don't remember the password, you can look it up on OLDSERVER in /opt/otrs/Kernel/Config.pm.

Furthermore I'd stop the cron service on NEWSERVER as well:

/etc/init.d/cron stop

**4. Import data on NEWSERVER(CENTOS 8)**

Let's start with the files. We can now unpack the tar.gz file into /opt:

***tar -xzf /tmp/otrs.tar.gz -C /opt/***

Verify the file ownerships. If they don't seem OK (e.g. a "ls -l /opt/otrs" doesn't show the user otrs), you should run

***/opt/otrs/bin/otrs.SetPermissions.pl.***

Then import the database:

mysql otrs < /tmp/otrs.sql

**5. Don't forget the cronjobs!**

This is most likely the step which could be forgotten the easiest. There are of course still the cron jobs on OLDSERVER(CENTOS 6.5), which need to run on NEWSERVER(CENTOS 8).

On OLDSERVER(CENTOS 6.5) show the crontab of user otrs:

crontab -l -u otrs

Copy the whole output and on NEWSERVER(CENTOS 8) insert it into the crontab:

crontab -e -u otrs

**6. Complete the migration**

Change DNS records to point to the new IP address, or inform your colleagues about the new domain/URL for OTRS.

It is recommended to do a quick testing on a secondary domain if OTRS works as it should. Only if the tests were successful, the "real" domain should then be changed or pointed to the migrated OTRS installation.

Once the tests were completed, you can start the cron service on NEWSERVER(CENTOS 8). This will start fetching the mails and create tickets again:

***/etc/init.d/cron start***

***Approximate time required for the migration:***

* ***For taking database backup of oldserver(centos 6.5) – around 6 hours***
* ***For taking backup using script(backup.pl) in centos 6.5 – around 10 hours***
* ***For creating tar ball of otrs 5.0.10 on centos 6.5 – around 3 hours***
* ***For installation and configuration of new latest centos 8 for migration of otrs from centos 6.5 to 8 – 3 hours***
* ***For untaring tar ball of otrs 5.0.10 on centos 6.5 to centos 8 – around 3 hours***
* ***For restoring otrs 5.0.10 to the new server (centos 8) – 14 hours***

***Problems faced during otrs installation and migration***

***1.Fails now with the error:***

*"Can't locate DateTime.pm in @inc (you may need to install the DateTime module) (@inc contains: /etc/perl /usr/local/lib/x86\_64-linux-gnu/perl/5.26.0 /usr/local/share/perl/5.26.0 /usr/lib/x86\_64-linux-gnu/perl5/5.26 /usr/share/perl5 /usr/lib/x86\_64-linux-gnu/perl/5.26 /usr/share/perl/5.26 /usr/local/lib/site\_perl /usr/lib/x86\_64-linux-gnu/perl-base) at /home/sk/bin/parseADSBxchg line 54.*

*BEGIN failed--compilation aborted at /home/sk/bin/parseADSBxchg line 54.*

*grep: ADS-B/\*.csv: No such file or directory"*

*I tried to install it using:*

*$ perl -MCPAN -e shell*

*cpan> install DateTime*

*cpan> q*

***2. Perl version problem***

*Make and Install*

*This is the procedure I used to compile Perl. I logged on as root for the entire process; you may wish to wait until the `make install` phase to switch to root, or do a `sudo make install`.*

*My system already had all the dependencies installed, so I can't give you a clear list of all the packages you'll need to be able to compile Perl. However, I know you'll need GNU Make and probably the GNU C Compiler. These can be installed quite easily on pretty much every Linux distribution using their package manager.*

*CLI:*

*yum install make*

*yum install gcc*

*Switch to root.*

*CLI:*

*su -*

*Enter the root password and hit enter.*

*Change to the directory you extracted Perl to. I extracted mine to my regular user's home directory.*

*CLI:*

*cd /home/user/perl-5.10.0*

*Configure the installation. This phase will make sure all the dependencies are installed and guess some default options. It's usually pretty good at guessing.*

*CLI:*

*sh Configure -de -Dusethreads*

*The -Dusethreads option is to compile the interpreter to support threads (which I recommend; there's no reason I can see why you SHOULDN'T compile with threads).*

*Run the standard make, make test, and make install commands. Note that `make install` must be executed as root, so if you're playing it safe and keeping root out of this until `make install`, you'll need to `sudo make install` instead (or switch to root, `cd` back to the source directory, and then do `make install`*

*CLI:*

*make*

*make test*

*make install*

*On my system, it installed the Perl 5.10.0 interpreter into /usr/local/bin/perl. Pay attention to what `make install` tells you to make sure you know where the Perl binary was moved to.*

*Post Configuration*

*By the time Perl 5.10 finished installing, I now have two different versions of Perl installed: the previous 5.8 version and my new 5.10. Like I said before, you may be tempted to try and remove version 5.8, but don't. I've still yet to get to the reason why.*

*On my system, /usr/local/bin/perl was version 5.10, but /usr/bin/perl was still the old 5.8 version. For general use, this was okay, because the new version of Perl showed up earlier in my $PATH statement, so that the `perl` command at the terminal executed the newer version. However, all of my CGI scripts on my web server were using /usr/bin/perl, the old version.*

*Code:*

*#!/usr/bin/perl -w*

*So this was a simple fix: just symlink the old binary to the new binary.*

*CLI:*

*su -*

*enter root's password*

*mv /usr/bin/perl /usr/bin/perl58*

*ln -s /usr/local/bin/perl /usr/bin/perl*

*What we did was we renamed the old perl to be "perl58" (so if you absolutely need it in the future, you can rename it back). Then we made a symbolic link to the new perl binary so that the location of the old binary still pointed to it.*

*Note, however, that Perl 5.8 and 5.10 have their own module paths, so if you installed any modules for Perl 5.8, you'll have to reinstall them for 5.10.*

***3.In my case the full error is:***

*Can't locate ExtUtils/MakeMaker.pm in @INC (@INC contains: /usr/local/lib64/perl5 /usr/local/share/perl5 /usr/lib64/perl5/vendor\_perl /usr/share/perl5/vendor\_perl /usr/lib64/perl5 /usr/share/perl5 .) at Makefile.PL line 24.*

*BEGIN failed--compilation aborted at Makefile.PL line 24.*

***The solution is easy, just run the following:***

***yum install perl-ExtUtils-MakeMaker -y***

***4.How to fix: fatal error: openssl/opensslv.h: No such file or directory in RedHat******7***

*To fix this problem, you have to install OpenSSL development package, which is available in standard repositories of all modern Linux distributions.*

*To install OpenSSL development package on Debian, Ubuntu or their derivatives:*

***$ sudo apt-get install libssl-dev***

*To install OpenSSL development package on Fedora, CentOS or RHEL:*

***$ sudo yum install openssl-devel***

***5.perl-DateTime is no longer part of the CentOS 8 base OS. You'll need to enable the PowerTools repository, i.e. as root***

*# yum config-manager --set-enabled PowerTools*

*# yum update*

*# yum repolist*

*repo id repo name status*

*PowerTools CentOS-8 - PowerTools*

***# yum install "perl(DateTime)"***

***References***

* <https://doc.otrs.com/doc/manual/admin/6.0/en/html/manual-installation-of-otrs.html>
* <https://doc.otrs.com/doc/manual/admin/6.0/en/html/backup-and-restore.html>
* <https://www.digitalocean.com/community/tutorials/how-to-set-up-a-help-desk-system-with-otrs-on-centos-7>

**Updation on progress**

Submitted By-

Clement P Abraham

Otrs 5.0.10 to latest

Praveen Mathew