

YETI installation manual

Date: 14-09-2015

Author: Michael Furmur <m.furmur@gmail.com>

Contents

1 Overview	1
2 Databases installation	1
2.1 Install packages	1
2.2 Create databases	2
2.3 Check	2
2.4 Init schema and data	3
3 Management interface installation	3
3.1 Install packages	3
3.2 Configure database connection	4
3.3 Init databases data	4
3.4 Launch	5
3.5 Check	5

1 Overview

Minimal system installation requires following compoments:

- Routing database (postgresql-9.3) see [sec:Databases-installation]
- CDRs database (postgresql-9.3) see [sec:Databases-installation]
- Management interface (web interace written on RoR) see [sec:Management-interface-installation]
- DSS storage (Redis) see [sec:DSS-Storage-installation]
- Management server (configurations server) see [sec:Management-server-installation]
- Traffic switch (SBC based on SEMS) see [sec:Traffic-switch-server-installation]
- Load balancer (kamailio) see [sec:Load-balancer-installation]

2 Databases installation

System requires two databases: one for routing and one for CDRs.

We recommend to place routing database at the same host with management interface to reach best interface responsiveness by reducing network delay for database requests from web-interface.

2.1 Install packages

both routing and CDRs databases requires similar sets of connected repositories and installed packages.

Make sure that following repositories added on your system:

```
deb http://ftp.us.debian.org/debian/ wheezy main
deb http://security.debian.org/ wheezy/updates main
deb http://ftp.us.debian.org/debian/ wheezy-updates main
deb http://pkg.yeti-switch.org/debian wheezy/
deb http://apt.postgresql.org/pub/repos/apt/ wheezy-pgdg main
```

System repositories can be changed by editing of file: /etc/apt/sources.list

import public keys for repositories:

```
# apt-key adv --keyserver keys.gnupg.net --recv-key 9CEBFFC569A832B6
# apt-key adv --keyserver keys.gnupg.net --recv-key 7FCC7D46ACCC4CF8
```

Install packages:

```
# aptitude update && \
aptitude install postgresql-9.3 \
postgresql-contrib-9.3 \
postgresql-9.3-prefix \
postgresql-9.3-pgq3 \
postgresql-9.3-yeti \
skytools3 \
skytools3-ticker
```

2.2 Create databases

Create routing database:

```
# su - postgres
$ psql
postgres=# create user yeti encrypted password 'somepassword' superuser;
CREATE ROLE
postgres=# create database yeti owner yeti;
CREATE DATABASE
postgres=# \q
```

Create database to store CDR:

```
# su - postgres
$ psql
postgres=# create database cdr owner yeti;
CREATE DATABASE
postgres=# \q
```

Note: It's recommended to specify values for databases names, usernames, passwords differ from specified in this manual for security reasons.

For large installations is reasonable to place CDR database on dedicated server

2.3 Check

Check databases created and accessible:

```

root@evial:/# psql -h 127.0.0.1 -U yeti -d yeti
Password for user yeti: psql (9.3.9) SSL connection (cipher: DHE-RSA-AES256-GCM-SHA384,
Type "help" for help.

yeti=# \q
root@evial:/#

root@evial:/# psql -h 127.0.0.1 -U yeti -d cdr
Password for user yeti:
psql (9.3.9)
SSL connection (cipher: DHE-RSA-AES256-GCM-SHA384, bits: 256)
Type "help" for help.

cdr=# \q
root@evial:/#

```

Don't forget to make changes in `/etc/postgresql/9.3/main/pg_hba.conf` and apply them if you plan to access this databases from another hosts

2.4 Init schema and data

Look at [Configure database connection](#) and [Init databases data](#) for further databases initialization instructions.

3 Management interface installation

Server requirements:

- OS Debian 7 Wheezy with architecture amd64
- at least 1GB of RAM

3.1 Install packages

Make sure that following repositories added on your system:

```

deb http://ftp.us.debian.org/debian/ wheezy main
deb http://security.debian.org/ wheezy/updates main
deb http://ftp.us.debian.org/debian/ wheezy-updates main
deb http://pkg.yeti-switch.org/debian wheezy/
deb http://packages.dotdeb.org wheezy all
deb http://apt.postgresql.org/pub/repos/apt/ wheezy-pgdg main

```

System repositories can be changed by editing of file: `/etc/apt/sources.list`

import public keys for repositories:

```

# apt-key adv --keyserver keys.gnupg.net --recv-key 9CEBFFC569A832B6
# apt-key adv --keyserver keys.gnupg.net --recv-key E9C74FEEA2098A6E
# apt-key adv --keyserver keys.gnupg.net --recv-key 7FCC7D46ACCC4CF8

```

Install packages:

```

# aptitude update && aptitude install yeti-web

```

3.2 Configure database connection

To configure database connection edit file `/home/yeti-web/config/database.yml` Create `database.yml` file with the following content:

```
production:
  adapter: postgresql
  encoding: unicode
  database: yeti
  pool: 5
  username: yeti
  password: somepassword
  host: 127.0.0.1
  schema_search_path: >
    gui,public,switch,
    billing,class4,runtime_stats,
    sys,logs,data_import
  port: 5432
  #min_messages: warning

production_cdr:
  adapter: postgresql
  encoding: unicode
  database: cdr
  pool: 5
  username: yeti
  password: somepassword
  host: 127.0.0.1
  schema_search_path: 'cdr,reports,billing'
  port: 5432
  #min_messages: warning
```

Warning: you should specify correct addresses and credentials using those which you chose in previous section

3.3 Init databases data

To simplify work with databases use utility `yeti-db` To initialize empty databases:

```
# yeti-db init
# yeti-db --cdr init
```

To upgrade databases to the latest version:

```
# yeti-db apply_all
# yeti-db --cdr apply_all
```

You can check actual database versions:

```
# yeti-db version
# yeti-db --cdr version
```

Attention: During upgrade of the system which working in production command `apply_all` should not be used because this command intended to upgrade to the last version only for system without live traffic. Systems in production must be upgraded using command `apply` which applies just one update in a single

run. After each upgrade it is important to amend appropriate configuration files and restart all traffic switch instances. This approach provides zero-downtime upgrade procedure (without loss of traffic and CDRs)

3.4 Launch

After successful configuration of databases you finally can run software using following commands:

```
# /etc/init.d/yeti-web start
# /etc/init.d/yeti-cdr-billing start
# /etc/init.d/yeti-delayed-job start
```

This will run web-interface and CDR processing workers

3.5 Check

check if unicorn listens socket:

```
# netstat -ltn | grep unicorn
unix 2 [ ACC ] STREAM LISTENING 2535145 24728/unicorn.rb -E /tmp/yeti-unicorn.sock
```

check if nginx listens for appropriate sockets:

```
# netstat -ltn | grep nginx
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN 23627/nginx
tcp 0 0 127.0.0.1:6666 0.0.0.0:* LISTEN 23627/nginx
```

Log files to check for possible warnings/errors :

- /var/log/yeti-admin.log
- /var/log/yeti-cdr-billing.log
- /home/yeti-web/log/unicorn.stdout.log
- /home/yeti-web/log/unicorn.stderr.log

Try to open management interface in your favorite browser and login with default credentials:

```
user:   admin
password: 111111
```