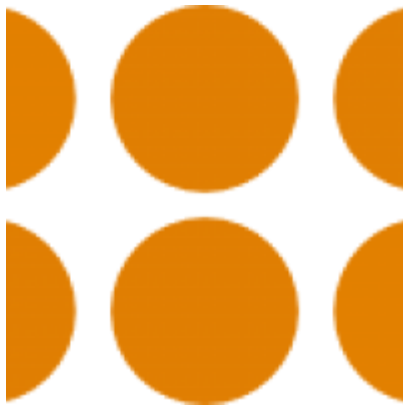




Author: powerdns community

[Scope of This Document](#)

[Glossary](#)

[nagios04 pdns 3.x on Centos 7](#)

[Docs](#)

[Step 1: PowerDNS with MySQL backend installation](#)

[2.5 phpmyadmin](#)

[Step 2: Setup MariaDB backend](#)

[Step 2.5 Install PowerDNS 4.x on Centos 7](#)

[Step 3: Configure PowerDNS](#)

[Step 4: Post-installation tasks](#)

[Step 5: Test how it works](#)

[Step 6: Recursor – installation and configuration](#)

[Step 7: PowerAdmin installation](#)

[Final thoughts](#)

[Inagios02 PowerDNS 4.1.x config](#)

[Create dns alias](#)

[Configure repository for Auth and Recursor](#)

[Install pdns and mysql backend](#)

[Configure Mariadb](#)

[Configure phpmyadmin](#)

[Make sure MariaDB is working](#)

[Setup MariaDB backend Create a database for PowerDNS, create a user for the database and grant him all privileges:](#)

[Step 3: Configure PowerDNS](#)

[Step 4: Post-installation tasks](#)

[Step 5: Test how it works](#)

[Check](#)

[Step 6: Recursor – installation and configuration](#)

[Dnsdist](#)

[Step 7: PowerAdmin installation](#)

[Final thoughts](#)

[Using pdns](#)

[Pdns using BIND as backend](#)

[lpm.test.lan integration](#)

[Miq integration](#)

[REST API](#)

[File listing](#)

[Install pdns using ansible](#)

[ERROR](#)

[pdns\[2064\]: Error receiving packet from recursor backend: Connection refused](#)

[File listing](#)

[Revision History](#)

[References](#)

Scope of This Document

We can easily just BIND but PowerDNS provides a great RESP API for DNS record CRUD action. This is great for automation.

Glossary

Name	Definition
Special Name	And its meaning here.

nagios04 pdns 3.x on Centos 7

Docs

1. Start
2. PDNS : [PowerDNS on CentOS 7 Guide](#)
3. End

Posted on March 17, 2016, By [Spiked Halo](#)

05/27/2017 Adapted by T.J. Yang for IaaS moto environment.

Although BIND is the most popular DNS-server in Linux, there are interesting alternatives such as PowerDNS. In our CentOS 7 PowerDNS installation guide we will provide all necessary instructions on how to install and configure PowerDNS and its web-front-end, Poweradmin. PowerDNS shows high performance and is distributed under the GPL license. It is the only solution, that stores zones in different backends like MySQL, PostgreSQL, LDAP and others. In addition, the PowerDNS Authoritative Server is the leading DNSSEC implementation. There are many advantages in comparison with BIND, such as: Choice of backend – store zones where you like – MySQL, PostgreSQL or just BIND-like flat files.

- Choice of replication principles – replicate with traditional master-slave scheme, or replicate back-ends
- Caching capabilities – Cache queries and packets
- Supermaster – when new zone is added to the supermaster, the slave will create the zone on itself and request an AXFR.

There is no such mechanism in BIND, for example

Next, we will install second service, Recursor, which is a high-end, high-performance resolving name server. It will be located on the same server as the authoritative DNS.

Enough reasons to try? For me – yes, and now – to the installation. Traditionally, I will use latest CentOS 7 minimal installation with firewalld and SELinux enabled for security reasons.

Step 1: PowerDNS with MySQL backend installation

1. First, add EPEL repository:

```
yum install epel-release.noarch
```

2. Next, proceed with MySQL(MariaDB) installation. In this step, we will install MariaDB, enable and start the service and make initial configuration through MySQL_secure_installation script.

```
# yum -y install mariadb-server mariadb
# systemctl enable mariadb.service
# systemctl start mariadb.service
# mysql_secure_installation
```

3. Configure mysql

```
[root@nagios04 ~]# mysql_secure_installation
```

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
```

```
In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, 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):
OK, successfully used password, moving on...
```

```
Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.
```

```
Set root password? [Y/n] Y
New password:
Re-enter new password: powerdns-password
Password updated successfully!
Reloading privilege tables..
... Success!
```

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB 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
... 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
... Success!
```

```
By default, MariaDB 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
- Dropping test database...
```

```

... Success!
- 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
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
[root@nagios04 ~]#

```

4. Check mysql process

```

[root@nagios04 ~]# ps -eaf |grep mysql
mysql      11381      1  0 10:02 ?          00:00:00 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
mysql      11538 11381  0 10:02 ?          00:00:00 /usr/libexec/mysqld --basedir=/usr
--datadir=/var/lib/mysql --plugin-dir=/usr/lib64mysql/plugin --log-error=/var/log/mariadb/mariadb.log
--pid-file=/var/run/mariadb/mariadb.pid --socket=/var/lib/mysql/mysql.sock
root       11795 11200  0 10:05 pts/0    00:00:00 grep --color=auto mysql
[root@nagios04 ~]#

```

5. End

2.5 phpmyadmin

```

scp /etc/httpd/conf.d/phpMyAdmin.conf root@salt01:/tmp/etc-httpd-conf.d-phpMyAdmin.conf
[aty010@nagios04 ~]$ egrep -v '^#|^$' /etc/httpd/conf.d/phpMyAdmin.conf
Alias /phpMyAdmin /usr/share/phpMyAdmin
Alias /phpmyadmin /usr/share/phpMyAdmin
<Directory /usr/share/phpMyAdmin/>
    AddDefaultCharset UTF-8
    <IfModule mod_authz_core.c>
        # Apache 2.4
        <RequireAny>
            Require all granted
        </RequireAny>
    </IfModule>
    <IfModule !mod_authz_core.c>
        # Apache 2.2
        Order Deny,Allow
        Deny from All
        Allow from 127.0.0.1
        Allow from ::1
    </IfModule>
</Directory>
<Directory /usr/share/phpMyAdmin/setup/>
    <IfModule mod_authz_core.c>
        # Apache 2.4

```

```

    <RequireAny>
    Require all granted
    </RequireAny>
</IfModule>
<IfModule !mod_authz_core.c>
    # Apache 2.2
    Order Deny,Allow
    Deny from All
    Allow from 127.0.0.1
    Allow from ::1
    Allow from All
</IfModule>
</Directory>
<Directory /usr/share/phpMyAdmin/libraries/>
    Order Deny,Allow
    Deny from All
    Allow from None
</Directory>
<Directory /usr/share/phpMyAdmin/setup/lib/>
    Order Deny,Allow
    Deny from All
    Allow from None
</Directory>
<Directory /usr/share/phpMyAdmin/setup/frames/>
    Order Deny,Allow
    Deny from All
    Allow from None
</Directory>
[aty010@nagios04 ~]$

```

2.5 [phpmyadmin](#)

```

[tjyang@nagios04 ~]$ sudo yum install -y phpmyadmin
sudo scp /etc/httpd/conf.d/phpMyAdmin.conf /etc/httpd/conf.d/phpMyAdmin.conf.old
sudo scp aty010@srepassword.test.lan:/etc/httpd/conf.d/phpMyAdmin.conf /etc/httpd/conf.d/phpMyAdmin.conf

```

Make sure MariaDB is working

```

[tjyang@nagios04 ~]$ ps -eaf |grep -i mysqld
mysql    16924      1  0 Jun30 ?        00:00:00 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
mysql    17025 16924  0 Jun30 ?        00:22:45 /usr/libexec/mysqld --basedir=/usr
--datadir=/var/lib/mysql --plugin-dir=/usr/lib64/mysql/plugin --log-error=/var/lib/mysql/nagios04.err
--pid-file=nagios04.pid
tjyang   63984 26120  0 10:35 pts/1    00:00:00 grep --color=auto -i mysqld
[tjyang@nagios04 ~]$

```

2.5 phpmyadmin

```
[tjyang@nagios04 ~]$ sudo yum install -y phpmyadmin
sudo scp /etc/httpd/conf.d/phpMyAdmin.conf /etc/httpd/conf.d/phpMyAdmin.conf.old
sudo scp aty010@srepassword.test.lan:/etc/httpd/conf.d/phpMyAdmin.conf /etc/httpd/conf.d/phpMyAdmin.conf
```

Make sure MariaDB is working

```
[tjyang@nagios04 ~]$ ps -eaf |grep -i mysqld
mysql    16924      1  0 Jun30 ?        00:00:00 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
mysql    17025 16924  0 Jun30 ?        00:22:45 /usr/libexec/mysqld --basedir=/usr
--datadir=/var/lib/mysql --plugin-dir=/usr/lib64/mysql/plugin --log-error=/var/lib/mysql/nagios04.err
--pid-file=nagios04.pid
tjyang   63984 26120  0 10:35 pts/1    00:00:00 grep --color=auto -i mysqld
[tjyang@nagios04 ~]$
```

Step 2: Setup MariaDB backend

1. Create a database for PowerDNS, create a user for the database and grant him all privileges:

```
[aty010@nagios04 ~]$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 5.5.56-MariaDB MariaDB Server

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]> show databases;
```

```
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| test |
| zabbix |
| zabbix_proxy |
+-----+
6 rows in set (0.00 sec)
```

```
MariaDB [(none)]> CREATE DATABASE powerdns;
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT ALL ON powerdns.* TO 'powerdns'@'localhost' IDENTIFIED BY
'powerdns-password';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]>
```

2. Next, setup the database created earlier – create scheme for it. We will follow [official instructions](#) from PowerDNS site:

```
USE powerdns;
CREATE TABLE domains (
  id          INT AUTO_INCREMENT,
  name        VARCHAR(255) NOT NULL,
  master      VARCHAR(128) DEFAULT NULL,
  last_check  INT DEFAULT NULL,
  type        VARCHAR(6) NOT NULL,
  notified_serial INT UNSIGNED DEFAULT NULL,
  account     VARCHAR(40) CHARACTER SET 'utf8' DEFAULT NULL,
  PRIMARY KEY (id)
) Engine=InnoDB CHARACTER SET 'latin1';

CREATE UNIQUE INDEX name_index ON domains(name);
```

```
CREATE TABLE records (
  id          BIGINT AUTO_INCREMENT,
  domain_id   INT DEFAULT NULL,
  name        VARCHAR(255) DEFAULT NULL,
  type        VARCHAR(10) DEFAULT NULL,
  content     VARCHAR(64000) DEFAULT NULL,
  ttl         INT DEFAULT NULL,
  prio        INT DEFAULT NULL,
  change_date INT DEFAULT NULL,
  disabled    TINYINT(1) DEFAULT 0,
  ordername   VARCHAR(255) BINARY DEFAULT NULL,
  auth        TINYINT(1) DEFAULT 1,
  PRIMARY KEY (id)
) Engine=InnoDB CHARACTER SET 'latin1';
```

```
CREATE INDEX nametype_index ON records(name,type);
CREATE INDEX domain_id ON records(domain_id);
CREATE INDEX ordername ON records (ordername);
```

```
CREATE TABLE supermasters (
  ip          VARCHAR(64) NOT NULL,
  nameserver  VARCHAR(255) NOT NULL,
  account     VARCHAR(40) CHARACTER SET 'utf8' NOT NULL,
  PRIMARY KEY (ip, nameserver)
) Engine=InnoDB CHARACTER SET 'latin1';
```

```
CREATE TABLE comments (
  id          INT AUTO_INCREMENT,
  domain_id   INT NOT NULL,
  name        VARCHAR(255) NOT NULL,
  type        VARCHAR(10) NOT NULL,
  modified_at INT NOT NULL,
  account     VARCHAR(40) CHARACTER SET 'utf8' DEFAULT NULL,
  comment     TEXT CHARACTER SET 'utf8' NOT NULL,
  PRIMARY KEY (id)
) Engine=InnoDB CHARACTER SET 'latin1';
```

```
CREATE INDEX comments_name_type_idx ON comments (name, type);
CREATE INDEX comments_order_idx ON comments (domain_id, modified_at);
```

```
CREATE TABLE domainmetadata (
```

```

    id                INT AUTO_INCREMENT,
    domain_id         INT NOT NULL,
    kind              VARCHAR(32),
    content           TEXT,
    PRIMARY KEY (id)
) Engine=InnoDB CHARACTER SET 'latin1';

CREATE INDEX domainmetadata_idx ON domainmetadata (domain_id, kind);

CREATE TABLE cryptokeys (
    id                INT AUTO_INCREMENT,
    domain_id         INT NOT NULL,
    flags             INT NOT NULL,
    active            BOOL,
    content           TEXT,
    PRIMARY KEY(id)
) Engine=InnoDB CHARACTER SET 'latin1';

CREATE INDEX domainidindex ON cryptokeys(domain_id);

CREATE TABLE tsigkeys (
    id                INT AUTO_INCREMENT,
    name              VARCHAR(255),
    algorithm          VARCHAR(50),
    secret            VARCHAR(255),
    PRIMARY KEY (id)
) Engine=InnoDB CHARACTER SET 'latin1';

CREATE UNIQUE INDEX namealgoindex ON tsigkeys(name, algorithm);

```

3. Check tables in powerdns DB.

```
MariaDB [powerdns]> show tables;
```

```

+-----+
| Tables_in_powerdns |
+-----+
| comments           |
| cryptokeys         |
| domainmetadata     |
| domains            |
| records            |
| supermasters       |
| tsigkeys           |
+-----+
7 rows in set (0.00 sec)

```

```
MariaDB [powerdns]>
```

4. End

Step 2.5 Install PowerDNS 4.x on Centos 7

1. os

```
sudo yum install epel-release yum-plugin-priorities &&
sudo rpm -e `rpm -qa |grep pdns`
sudo curl -o /etc/yum.repos.d/powerdns-rec-41.repo https://repo.powerdns.com/repo-files/centos-rec-41.repo
```

2. Install pdns and mysql backend

Backend is successfully installed – next, the PowerDNS and Recursor installation:

```
pdns-recursor
[tjyang@nagios04 ~]$ sudo yum -y install pdns pdns-backend-mysql pdns-tools
[tjyang@nagios04 ~]$ rpm -qa |grep pdns
pdns-recursor-4.1.3-2.el7.x86_64
pdns-tools-3.4.11-4.el7.x86_64
pdns-3.4.11-4.el7.x86_64
pdns-backend-mysql-3.4.11-4.el7.x86_64
[tjyang@nagios04 ~]$
```

3. End

Step 3: Configure PowerDNS

1. Now all is ready for configuration. First, configure PowerDNS:

```
# vi /etc/pdns/pdns.conf
[aty010@nagios04 ~]$ sudo egrep -v '^#|^$' /etc/pdns/pdns.conf
setuid=pdns
setgid=pdns
launch=bind
[aty010@nagios04 ~]$
```

2. Insert the following part of config at the beginning of the file, and don't forget to enter password for the database user created earlier:

```
launch=gmysql
gmysql-host=localhost
gmysql-user=powerdns
gmysql-password=powerdns-password
gmysql-dbname=powerdns
gmysql-dnssec=yes
loglevel=10
log-dns-queries=1
setuid=pdns
setgid=pdns
#launch=bind
```

3. Start and enable the service:

```
# systemctl enable pdns.service && systemctl start pdns.service
```

```
[root@nagios04 log]# systemctl status -l pdns
```

```
• pdns.service - PowerDNS Authoritative Server
   Loaded: loaded (/usr/lib/systemd/system/pdns.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2018-07-17 09:02:29 EDT; 5s ago
   Process: 3631 ExecStop=/usr/bin/pdns_control quit (code=exited, status=0/SUCCESS)
   Process: 3636 ExecStart=/usr/sbin/pdns_server --daemon (code=exited, status=0/SUCCESS)
   Main PID: 3637 (pdns_server)
   CGroup: /system.slice/pdns.service
           └─3637 /usr/sbin/pdns_server --daemon
```

```
Jul 17 09:02:29 nagios04 pdns[3637]: PowerDNS comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it according to the terms of the GPL version 2.
```

```
Jul 17 09:02:29 nagios04 pdns[3637]: Set effective group id to 985
```

```
Jul 17 09:02:29 nagios04 pdns[3637]: Set effective user id to 987
```

```
Jul 17 09:02:29 nagios04 pdns[3637]: Creating backend connection for TCP
```

```
Jul 17 09:02:29 nagios04 pdns[3637]: gmysql Connection successful. Connected to database 'powerdns' on
'localhost'.
```

```
Jul 17 09:02:29 nagios04 pdns[3637]: About to create 3 backend threads for UDP
```

```
Jul 17 09:02:29 nagios04 pdns[3637]: gmysql Connection successful. Connected to database 'powerdns' on
'localhost'.
```

```
Jul 17 09:02:29 nagios04 pdns[3637]: gmysql Connection successful. Connected to database 'powerdns' on
'localhost'.
```

```
Jul 17 09:02:29 nagios04 pdns[3637]: gmysql Connection successful. Connected to database 'powerdns' on
'localhost'.
```

```
Jul 17 09:02:29 nagios04 pdns[3637]: Done launching threads, ready to distribute questions
```

```
[root@nagios04 log]#
```

Step 4: Post-installation tasks

1. Create firewall rule to allow dns traffic and reload rules:

```
# firewall-cmd --zone=public --add-service=dns --permanent
# firewall-cmd --reload
```

Step 5: Test how it works

We have finished configuring our PowerDNS server – let's test how it works.

```
[root@nagios04 log]# nslookup www.ya.ru 127.0.0.1
Server:      127.0.0.1
Address:     127.0.0.1#53

** server can't find www.ya.ru.test.lan: REFUSED

[root@nagios04 log]#
```

As you can see, server refused to answer – and this is correct, because it is not authoritative for this domain. Now let's change config a little to get the answer. We will add a recursor entry in the config file and restart pdns:

```
# vi /etc/pdns/pdns.conf

recursor=8.8.8.8
allow-recursion=0.0.0.0/0

# systemctl restart pdns
```

We allowed the recursion and added the recursor IP-address. Let's check again:

```
[root@nagios04 log]# nslookup www.ya.ru 127.0.0.1 ;date
Server:      127.0.0.1
Address:     127.0.0.1#53

Non-authoritative answer:
www.ya.ru    canonical name = ya.ru.
Name:   ya.ru
Address: 87.250.250.242

Tue Jul 17 09:06:43 EDT 2018
[root@nagios04 log]#
[root@nagios04 log]# nslookup cnn.com 127.0.0.1;date
Server:      127.0.0.1
Address:     127.0.0.1#53

Non-authoritative answer:
Name:   cnn.com
Address: 151.101.65.67
Name:   cnn.com
Address: 151.101.129.67
Name:   cnn.com
Address: 151.101.193.67
Name:   cnn.com
Address: 151.101.1.67

Tue Jul 17 09:13:43 EDT 2018
[root@nagios04 log]#
```

As you can see, everything works fine! But what if we want to use authoritative DNS server as well as use primary and secondary servers, or to create split-brain DNS scenario?

And here another program product will help – the [Recursor](#). In the next steps, we will install and configure the Recursor.

Step 6: Recursor – installation and configuration

1. Install the recursor:

```
# yum -y install pdns-recursor
```

2. Now change the default config.

We will change default port – because the default port of recursor is 53 – and this port is already occupied with PowerDNS, so let's chose 8090, for example. Next, we add a forwarding rule: it means – forward all queries for . domain to 8.8.8.8 (primary) or if it is not reachable – to 8.8.4.4. (secondary). After making changes, start pdns-recursor and enable the service:

```
[root@nagios04 poweradmin-2.1.7]# egrep -v '^#|^$' /etc/pdns-recursor/recursor.conf
setuid=pdns-recursor
setgid=pdns-recursor
forward-zones=.=8.8.8.8;8.8.4.4
forward-zones-recurse=.=8.8.8.8;8.8.4.4
local-port=8090
local-address=100.64.5.234
allow-from=0.0.0.0/0
```

```
[root@nagios04 poweradmin-2.1.7]#date
Tue Jul 17 09:54:50 EDT 2018
[root@nagios04 poweradmin-2.1.7]#
```

```
# systemctl start pdns-recursor && systemctl enable pdns-recursor
# systemctl restart pdns-recursor
[root@nagios04 ~]# systemctl status -l pdns-recursor
```

```
• pdns-recursor.service - PowerDNS recursing nameserver
  Loaded: loaded (/usr/lib/systemd/system/pdns-recursor.service; enabled; vendor preset: disabled)
  Active: active (running) since Sat 2017-05-27 10:45:08 CDT; 19s ago
    Docs: man:pdns_recursor(1)
          man:rec_control(1)
  Main PID: 17051 (pdns_recursor)
    CGroup: /system.slice/pdns-recursor.service
            └─17051 /usr/sbin/pdns_recursor --daemon
```

```
May 27 10:45:08 nagios04 pdns_recursor[17051]: Set effective group id to 978
May 27 10:45:08 nagios04 pdns_recursor[17051]: Set effective user id to 983
May 27 10:45:08 nagios04 pdns_recursor[17051]: Launching 3 threads
May 27 10:45:08 nagios04 pdns_recursor[17051]: Done priming cache with root hints
May 27 10:45:08 nagios04 pdns_recursor[17051]: Done priming cache with root hints
May 27 10:45:08 nagios04 pdns_recursor[17051]: Enabled 'epoll' multiplexer
May 27 10:45:08 nagios04 pdns_recursor[17051]: Done priming cache with root hints
May 27 10:45:08 nagios04 pdns_recursor[17051]: Refreshed . records
May 27 10:45:08 nagios04 pdns_recursor[17051]: Refreshed . records
May 27 10:45:08 nagios04 pdns_recursor[17051]: Refreshed . records
```

```
[root@nagios04 poweradmin-2.1.7]# ps axf |grep "sbin/pdns" |grep -v grep;date
6873 ?        Ssl      0:00 /usr/sbin/pdns_server --daemon
```

```

24664 ?          Ssl      0:00 /usr/sbin/pdns_recursor --daemon=no --write-pid=no --disable-syslog
--log-timestamp=no
Tue Jul 17 09:55:23 EDT 2018
[root@nagios04 poweradmin-2.1.7]#

```

3. Change the PowerDNS configuration as well and restart the service:

```

[root@nagios04 poweradmin-2.1.7]# egrep -v '^#|^$' /etc/pdns/pdns.conf
launch=gmysql
gmysql-host=localhost
gmysql-user=powerdns
gmysql-password=powerdns-password
gmysql-dbname=powerdns
gmysql-dnssec=yes
loglevel=10
log-dns-queries=1
setuid=pdns
setgid=pdns
recursor=8.8.8.8
allow-recursion=0.0.0.0/0
recursor=100.64.5.234:8090
[root@nagios04 poweradmin-2.1.7]#

```

```
systemctl restart pdns
```

```

[root@nagios04 ~]# systemctl status -l pdns
• pdns.service - PowerDNS Authoritative Server
   Loaded: loaded (/usr/lib/systemd/system/pdns.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2018-07-31 18:43:52 EDT; 14s ago
     Process: 54458 ExecStop=/usr/bin/pdns_control quit (code=exited, status=0/SUCCESS)
     Process: 54463 ExecStart=/usr/sbin/pdns_server --daemon (code=exited, status=0/SUCCESS)
    Main PID: 54464 (pdns_server)
      CGroup: /system.slice/pdns.service
              └─54464 /usr/sbin/pdns_server --daemon

Jul 31 18:43:52 nagios04 pdns[54464]: Set effective group id to 985
Jul 31 18:43:52 nagios04 pdns[54464]: Set effective user id to 987
Jul 31 18:43:52 nagios04 pdns[54464]: DNS Proxy launched, local port 31620, remote 127.0.0.1:8090
Jul 31 18:43:52 nagios04 pdns[54464]: Creating backend connection for TCP
Jul 31 18:43:52 nagios04 pdns[54464]: gmysql Connection successful. Connected to database
'powerdns' on 'localhost'.
Jul 31 18:43:52 nagios04 pdns[54464]: About to create 3 backend threads for UDP
Jul 31 18:43:52 nagios04 pdns[54464]: gmysql Connection successful. Connected to database
'powerdns' on 'localhost'.
Jul 31 18:43:52 nagios04 pdns[54464]: gmysql Connection successful. Connected to database
'powerdns' on 'localhost'.
Jul 31 18:43:52 nagios04 pdns[54464]: gmysql Connection successful. Connected to database
'powerdns' on 'localhost'.
Jul 31 18:43:52 nagios04 pdns[54464]: Done launching threads, ready to distribute questions
[root@nagios04 ~]#

```

```

Jul 17 09:57:25 nagios04 pdns_server: Jul 17 09:57:25 Reading random entropy from '/dev/urandom'
Jul 17 09:57:25 nagios04 pdns[25583]: Reading random entropy from '/dev/urandom'
Jul 17 09:57:25 nagios04 pdns_server: Jul 17 09:57:25 Loading
'/usr/lib64/pdns/libgmysqlbackend.so'
Jul 17 09:57:25 nagios04 pdns[25583]: Loading '/usr/lib64/pdns/libgmysqlbackend.so'
Jul 17 09:57:25 nagios04 pdns_server: Jul 17 09:57:25 [gmysqlbackend] This is the gmysql backend
version 3.4.11 (Feb 1 2018, 16:32:30) reporting

```

```

Jul 17 09:57:25 nagios04 pdns[25583]: [gmysqlbackend] This is the gmysql backend version 3.4.11
(Feb 1 2018, 16:32:30) reporting
Jul 17 09:57:25 nagios04 pdns[25584]: This is a standalone pdns
Jul 17 09:57:25 nagios04 pdns[25584]: Listening on controlsocket in '/var/run/pdns.controlsocket'
Jul 17 09:57:25 nagios04 pdns[25584]: UDP server bound to 0.0.0.0:53
Jul 17 09:57:25 nagios04 pdns[25584]: TCP server bound to 0.0.0.0:53
Jul 17 09:57:25 nagios04 pdns[25584]: PowerDNS Authoritative Server 3.4.11
(jenkins@autotest.powerdns.com) (C) 2001-2016 PowerDNS.COM BV
Jul 17 09:57:25 nagios04 pdns[25584]: Using 64-bits mode. Built on 20180201163352 by
mockbuild@buildhw-08.phx2.fedoraproject.org, gcc 4.8.5 20150623 (Red Hat 4.8.5-16).
Jul 17 09:57:25 nagios04 pdns[25584]: PowerDNS comes with ABSOLUTELY NO WARRANTY. This is free
software, and you are welcome to redistribute it according to the terms of the GPL version 2.
Jul 17 09:57:25 nagios04 pdns[25584]: Set effective group id to 985
Jul 17 09:57:25 nagios04 pdns[25584]: Set effective user id to 987
Jul 17 09:57:25 nagios04 pdns[25584]: DNS Proxy launched, local port 32824, remote 127.0.0.1:8090
Jul 17 09:57:25 nagios04 pdns[25584]: Creating backend connection for TCP
Jul 17 09:57:25 nagios04 pdns[25584]: gmysql Connection successful. Connected to database
'powerdns' on 'localhost'.
Jul 17 09:57:25 nagios04 pdns[25584]: About to create 3 backend threads for UDP
Jul 17 09:57:25 nagios04 pdns[25584]: gmysql Connection successful. Connected to database
'powerdns' on 'localhost'.
Jul 17 09:57:25 nagios04 pdns[25584]: gmysql Connection successful. Connected to database
'powerdns' on 'localhost'.
Jul 17 09:57:25 nagios04 pdns[25584]: gmysql Connection successful. Connected to database
'powerdns' on 'localhost'.
Jul 17 09:57:25 nagios04 pdns[25584]: Done launching threads, ready to distribute questions

```

```

[root@nagios04 ~]# systemctl status -l pdns

```

```

• pdns.service - PowerDNS Authoritative Server
  Loaded: loaded (/usr/lib/systemd/system/pdns.service; enabled; vendor preset: disabled)
  Active: active (running) since Tue 2018-07-31 18:43:52 EDT; 1min 26s ago
  Process: 54458 ExecStop=/usr/bin/pdns_control quit (code=exited, status=0/SUCCESS)
  Process: 54463 ExecStart=/usr/sbin/pdns_server --daemon (code=exited, status=0/SUCCESS)
  Main PID: 54464 (pdns_server)
  CGroup: /system.slice/pdns.service
          └─54464 /usr/sbin/pdns_server --daemon

```

```

Jul 31 18:44:38 nagios04 pdns[54464]: Remote 127.0.0.1 wants 'www.ya.ru|A', do = 0, bufsize = 512:
packetcache MISS
Jul 31 18:44:38 nagios04 pdns[54464]: Distributor misses a thread (4<5), spawning new one
Jul 31 18:44:38 nagios04 pdns[54464]: gmysql Connection successful. Connected to database
'powerdns' on 'localhost'.
Jul 31 18:44:38 nagios04 pdns[54464]: Error receiving packet from recursor backend: Connection
refused
Jul 31 18:44:43 nagios04 pdns[54464]: Remote 127.0.0.1 wants 'www.ya.ru|A', do = 0, bufsize = 512:
packetcache MISS
Jul 31 18:44:43 nagios04 pdns[54464]: Error receiving packet from recursor backend: Connection
refused
Jul 31 18:44:48 nagios04 pdns[54464]: Remote 127.0.0.1 wants 'www.ya.ru|A', do = 0, bufsize = 512:
packetcache MISS
Jul 31 18:44:48 nagios04 pdns[54464]: Error receiving packet from recursor backend: Connection
refused
Jul 31 18:44:53 nagios04 pdns[54464]: Remote 127.0.0.1 wants 'www.ya.ru.test.lan|A', do = 0,
bufsize = 512: packetcache MISS
Jul 31 18:44:53 nagios04 pdns[54464]: Error receiving packet from recursor backend: Connection
refused
[root@nagios04 ~]#

```

4. Finally, let's check how it works:

```
# nslookup www.ya.ru 127.0.0.1
Server:      127.0.0.1
Address:     127.0.0.1#53

Non-authoritative answer:
www.ya.ru    canonical name = ya.ru.
Name:   ya.ru
Address: 213.180.204.3
Name:   ya.ru
Address: 93.158.134.3
Name:   ya.ru
Address: 213.180.193.3
```

As you can see, everything works fine through pdns-redirector.

But to manage zones and records effectively, we must add a web front-end to our installation. There are several different possibilities: [Poweradmin](#), [PowerDNS on Rails](#) and [JPower Admin](#). In this manual, I will choose Poweradmin because of support for most features of PowerDNS.

Step 7: PowerAdmin installation

1. Start

We will use Apache as our web-server, so install all necessary php modules and Apache itself:

```
#yum install httpd php php-devel php-gd php-imap php-ldap php-mysql php-odbc php-pear php-xml php-xmlrpc
php-mbstring php-mcrypt php-mhash gettext php-pear-DB php-pear-MDB2-Driver-mysql -y
```

2. Httpd service

Start and enable httpd service, and configure firewall rules. Do not forget – we left SELinux enabled, so we must edit the Boolean for httpd:

```
# systemctl enable httpd
# systemctl start httpd
# firewall-cmd --zone=public --add-service=http --permanent
# firewall-cmd --zone=public --add-service=https --permanent
# firewall-cmd --reload
# setsebool -P httpd_can_network_connect=true
```

3. www

```
[root@nagios04 ~]# grep ^DocumentRoot /etc/httpd/conf/httpd.conf
DocumentRoot "/var/www/html"
[root@nagios04 ~]#

[root@nagios04 ~]# grep nagios /etc/httpd/conf.d/ssl.conf
ServerName nagios04.test.lan
SSLCertificateFile /etc/pki/tls/certs/nagios03.test.lan/nagios03.test.lan.SST00531.crt
SSLCertificateKeyFile /etc/pki/tls/certs/nagios03.test.lan/nagios03.test.lan.key
[root@nagios04 ~]#
```

4. Install poweradmin from github

Download the package and extract files in /var/www/html:

```
# yum install wget -y
# cd /var/www/html/
# wget http://downloads.sourceforge.net/project/poweradmin/poweradmin-2.1.7.tgz
# tar xfv poweradmin-2.1.7.tgz
```

5. Start web-based installation by opening <http://nagios04.test.lan/poweradmin-2.1.7/install/> and Go to Step 2

6. End

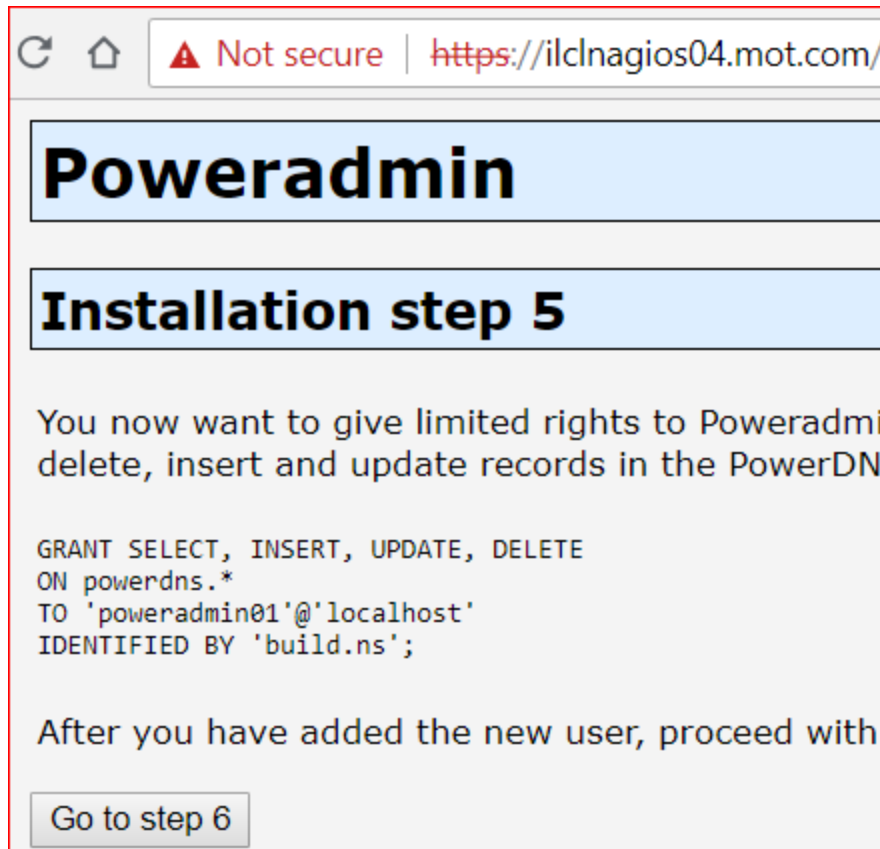
4.

5. Strange step, but – just proceed to Step 3

6. Provide credentials we created earlier – Username and password – for connection to MariaDB, database name – that we created earlier, and a password for administrative account:

7. Provide credentials for limited user, Hostmaster and NS records, and proceed to step 5:

8. Now we need to create that limited user from the previous step and grant him needed privileges. Back to console, enter MariaDB prompt and insert the code provided by web-installer:



```
# mysql -u root -p
```

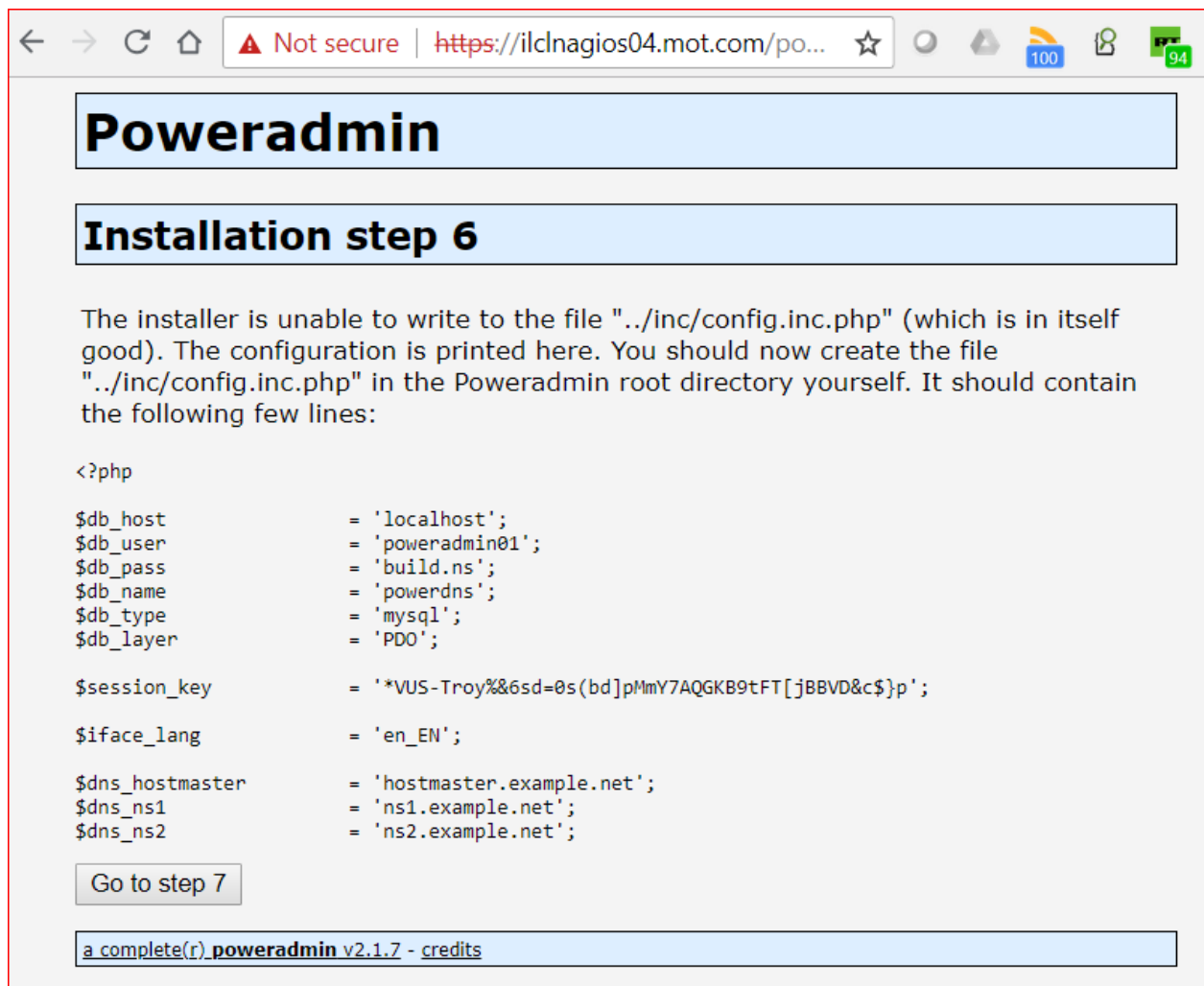
```
MariaDB [(none)]> GRANT SELECT, INSERT, UPDATE, DELETE ON powerdns.* TO  
'poweradmin01'@'localhost' IDENTIFIED BY 'powerdns-password';  
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [(none)]> FLUSH PRIVILEGES;  
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [(none)]> exit;
```

9. The installer will attempt to create its configuration file in `/var/www/html/poweradmin-2.1.7/inc`. If the script couldn't make these changes, we will have to make them manually. Create a file,

then paste the strings provided by web-installer, and then save-exit :



The screenshot shows a web browser window with the address bar displaying "https://ilclnagios04.mot.com/po...". The page title is "Poweradmin". The main heading is "Installation step 6". The text explains that the installer cannot write to the file "../inc/config.inc.php" and provides the configuration details. Below the text is a code block for the configuration file. At the bottom, there is a button "Go to step 7" and a link "a complete(r) poweradmin v2.1.7 - credits".

Poweradmin

Installation step 6

The installer is unable to write to the file "../inc/config.inc.php" (which is in itself good). The configuration is printed here. You should now create the file "../inc/config.inc.php" in the Poweradmin root directory yourself. It should contain the following few lines:

```
<?php

$db_host          = 'localhost';
$db_user          = 'poweradmin01';
$db_pass          = 'build.ns';
$db_name          = 'powerdns';
$db_type          = 'mysql';
$db_layer         = 'PDO';

$session_key      = '*VUS-Troy%&6sd=0s(bd]pMmY7AQGKB9tFT[jBBVD&c$)p';

$iface_lang       = 'en_EN';

$dns_hostmaster   = 'hostmaster.example.net';
$dns_ns1           = 'ns1.example.net';
$dns_ns2           = 'ns2.example.net';
```

[Go to step 7](#)

[a complete\(r\) poweradmin v2.1.7 - credits](#)

→ ↻ 🏠 ⚠ Not secure | https://ilcnnagios04.mot.com/po... ☆ 🔍

Poweradmin

Installation step 6

The installer is unable to write to the file "../inc/config.inc.php" (which is not good). The configuration is printed here. You should now create the file "../inc/config.inc.php" in the Poweradmin root directory yourself. It should contain the following few lines:

```
<?php

$db_host          = 'localhost';
$db_user          = 'poweradmin01';
$db_pass          = 'build.ns';
$db_name          = 'powerdns';
$db_type          = 'mysql';
$db_layer         = 'PDO';

$session_key      = '*VUS-Troy%&6sd=0s(bd)pMmY7AQGKB9tFT[jBBVD&c$}p';

$iface_lang       = 'en_EN';

$dns_hostmaster   = 'hostmaster.example.net';
$dns_ns1           = 'ns1.example.net';
$dns_ns2           = 'ns2.example.net';
```

[Go to step 7](#)

- 1 Test
- 2 # cd /var/www/html/poweradmin-2.1.7/inc
- 3 # touch config.inc.php
vi config.inc.php

Create config.inc.php

```
[root@nagios04 inc]# cat /var/www/html/poweradmin-2.1.7/inc/config.inc.php
<?php

$db_host          = 'localhost';
$db_user          = 'poweradmin01';
$db_pass          = 'powerdns-password';
$db_name          = 'powerdns';
$db_type          = 'mysql';
```

```

$db_layer                = 'PDO';

$session_key             = 'OrC)UBGYB@lZtzpebIr+ob_v+f@)jJq!)rqcZM#WF4Z$o';

$iface_lang              = 'en_EN';

$dns_hostmaster          = 'hostmaster.example.net';
$dns_ns1                 = 'ns1.example.net';
$dns_ns2                 = 'ns2.example.net';
[root@nagios04 inc]#

<?php

$db_host                 = 'localhost';
$db_user                 = 'poweradmin01';
$db_pass                 = 'powerdns-password';
$db_name                 = 'powerdns';
$db_type                 = 'mysql';
$db_layer                = 'PDO';

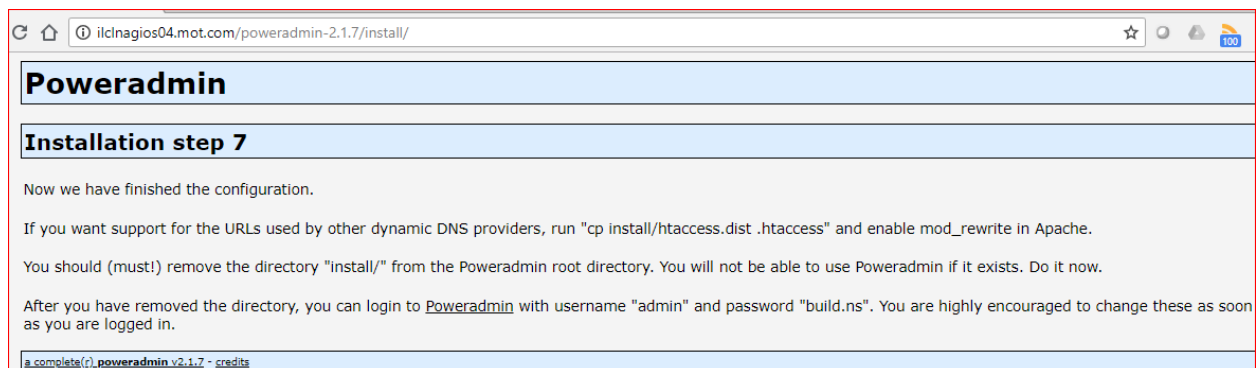
$session_key             = '+ka! (UGW%ER}pHr=9E_0TmR75t!D0zUAH37KSmmXQ{pHrX';

$iface_lang              = 'en_EN';

$dns_hostmaster          = 'nagios04.test.lan';
$dns_ns1                 = '100.64.21.50';
$dns_ns2                 = '100.65.128.130';

```

10. Now the final step – remove install folder. You will see a warning message, if you will try to access web-interface without deletion of that folder, but my advice – check the availability of <http://%your-IP-address%/poweradmin-2.1.7/> – and only after seeing this warning proceed with the deletion:



Now we have finished the configuration.

If you want support for the URLs used by other dynamic DNS providers, run "cp install/htaccess.dist .htaccess" and enable mod_rewrite in Apache.

You should (must!) remove the directory "install/" from the Poweradmin root directory. You will not be able to use Poweradmin if it exists. Do it now.

After you have removed the directory, you can login to Poweradmin with username "admin" and password "powerdns-password". You are highly encouraged to change these as soon as you are logged in.

```
[root@nagios04 poweradmin-2.1.7]# pwd;mv install install.old
/var/www/html/poweradmin-2.1.7
[root@nagios04 poweradmin-2.1.7]#
```

```
# rm -rf /var/www/html/poweradmin-2.1.7/install/
```

Final thoughts

So, for now, you have configured PowerDNS with PowerAdmin web-interface and Recursor on the single host. Everything went fine? Good. But there are some errors you can face, and this is how you can deal with them:

1. After finishing web-installer, you see an empty page or an 500 error.

Solution: check the file and folder permissions. Try to move the contents of /var/www/html/poweradmin-2.1.7 to the root of web-server.

2. After finishing web-installer, you see – Unable to connect to database. You check all the passwords – and they are correct. Solution: Try to avoid special characters li, and consider changing the password
3. End

Using pdns

1. Using admin/powerdns-password to login <https://nagios04.test.lan/poweradmin-2.1.7/>
2. Tbc
3. tbc

Pdns using BIND as backend

1. WHAT:
2. WHY: work with existing BIND instance.
3. tbc

Ipam.test.lan integration

Miq integration

REST API

1. Start
2. Doc
 - a. <https://ipmens.net/2015/01/24/nsedit-a-dns-zone-and-record-for-powerdns/>
 - b. <https://ipmens.net/2015/01/09/a-look-at-the-powerdns-rest-api/>
 - c. End

3. Config

```
[root@nagios04 ~]# egrep -v '^#|^$' /etc/pdns/pdns.conf
launch=gmysql
gmysql-host=localhost
gmysql-user=powerdns
gmysql-password=powerdns-password
gmysql-dbname=powerdns
gmysql-dnssec=yes
loglevel=1
log-dns-queries=0
setuid=pdns
setgid=pdns
recursor=8.8.8.8
allow-recursion=0.0.0.0/0
experimental-api-key=powerdns-123
webserver=yes
webserver-address=100.64.5.234
webserver-allow-from=0.0.0.0/0,::/0
[root@nagios04 ~]#
```

4. End

File listing

Install pdns using ansible

```
sudo yum install -y ansible
ansible-galaxy install wingundamth.pdns_recursor-ansible
git clone https://github.com/PowerDNS/pdns\_recursor-ansible.git
cd pdns_recursor-ansible

[tjyang@nagios04 github]$ cat ./hosts
nagios04.test.lan
[tjyang@nagios04 github]$

ansible-playbook -i hosts pdns41.yml --ask-pass -u root -k --check
--extra-vars='ansible_become_pass=powerdns-password'

[tjyang@nagios04 github]$ cat pdns41.yml
- hosts: all
  become: yes
  become_user: root
  become_method: ssh
  roles:
    - { role: powerdns.pdns }
  vars:
    pdns_config:
      master: true
      slave: false
      local-address: '100.64.5.234'
    pdns_backends:
      gmysql:
        host: 100.64.5.234
        port: 3306
        user: powerdns
        password: pdns-1234
        dbname: pdns
    pdns_mysql_databases_credentials:
      gmysql:
        priv_user: root
        priv_password: pdns-1234
        priv_host:
          - "%"
    pdns_install_repo: "{{ pdns_auth_powerdns_repo_41 }}"
[tjyang@nagios04 github]$
```

ERROR

1. info

Aug 10 05:41:50 pdns pdns_server: Remote 127.0.0.1 wants 'a.root-servers.net|A', do = 0, bufsize = 512: packetcache HIT

2. Aug 1 07:58:36 nagios04 pdns_server: Fatal error: Trying to set unknown parameter 'recursor'

a. solution

3. error

```
Jul 17 09:00:16 nagios04 pdns_server: Jul 17 09:00:16 Reading random entropy from '/dev/urandom'
Jul 17 09:00:16 nagios04 pdns[2521]: Reading random entropy from '/dev/urandom'
Jul 17 09:00:16 nagios04 pdns_server: Jul 17 09:00:16 Loading
'/usr/lib64/pdns/libmysqlbackend.so'
Jul 17 09:00:16 nagios04 pdns[2521]: Loading '/usr/lib64/pdns/libmysqlbackend.so'
Jul 17 09:00:16 nagios04 pdns_server: Jul 17 09:00:16 [gmysqlbackend] This is the gmysql backend
version 3.4.11 (Feb 1 2018, 16:32:30) reporting
Jul 17 09:00:16 nagios04 pdns[2521]: [gmysqlbackend] This is the gmysql backend version 3.4.11
(Feb 1 2018, 16:32:30) reporting
Jul 17 09:00:16 nagios04 pdns[2522]: This is a standalone pdns
Jul 17 09:00:16 nagios04 pdns[2522]: Listening on controlsocket in '/var/run/pdns.controlsocket'
Jul 17 09:00:16 nagios04 pdns[2522]: Fatal error: Trying to set unknown parameter 'mysql-host'
Jul 17 09:00:16 nagios04 systemd: pdns.service: main process exited, code=exited, status=1/FAILURE
Jul 17 09:00:16 nagios04 pdns_control: Fatal error: Unable to connect to remote
'/var/run/pdns.controlsocket': Connection refused
Jul 17 09:00:16 nagios04 systemd: pdns.service: control process exited, code=exited status=1
Jul 17 09:00:16 nagios04 systemd: Unit pdns.service entered failed state.
Jul 17 09:00:16 nagios04 systemd: pdns.service failed.
```

4. Tbc

```
May 27 10:28:46 nagios04 pdns[14624]: Listening on controlsocket in '/var/run/pdns.controlsocket'
May 27 10:28:46 nagios04 systemd[1]: pdns.service: main process exited, code=exited, status=1/FAILURE
May 27 10:28:46 nagios04 pdns[14624]: Fatal error: Trying to set unknown parameter 'gmysql-host'
May 27 10:28:46 nagios04 pdns_control[14626]: Fatal error: Unable to connect to remote
'/var/run/pdns.controlsocket': Connection refused
May 27 10:28:46 nagios04 systemd[1]: pdns.service: control process exited, code=exited status=1
May 27 10:28:46 nagios04 systemd[1]: Unit pdns.service entered failed state.
May 27 10:28:46 nagios04 systemd[1]: pdns.service failed.
```

Solution: #launch=bind

5. pdns

pdns[27126]: Error receiving packet from recursor backend: Connection refused

```
local-address=100.64.16.110
allow-from=100.64.16.0/24
```

```
In pdns.conf
#recursor=127.0.0.1:8090
recursor=100.64.16.110:8090
```

6. pdns[2064]: Error receiving packet from recursor backend: Connection refused

a. debug

```
[root@nagios04 ~]# cat /etc/pdns-recursor/recursor.conf
setuid=pdns-recursor
setgid=pdns-recursor
forward-zones=.=8.8.8.8;8.8.4.4
forward-zones-recurse=.=8.8.8.8;8.8.4.4
local-port=8090
local-address=100.64.5.234
allow-from=100.64.16.0/24
[root@nagios04 ~]#
```

b. solution

```
allow-from=0.0.0.0/0 in recursor.conf
```

c. End

7. tbc

File listing

1. Start

```
# https://cmikavac.net/2011/09/11/how-to-generate-an-ip-range-list-in-python/
def ipRange(start_ip, end_ip):
    start = list(map(int, start_ip.split(".")))
    end = list(map(int, end_ip.split(".")))
    temp = start
    ip_range = []

    ip_range.append(start_ip)
    while temp != end:
        start[3] += 1
        for i in (3, 2, 1):
            if temp[i] == 256:
                temp[i] = 0
                temp[i-1] += 1
        ip_range.append(".".join(map(str, temp)))

    return ip_range
# sample usage
ip_range = ipRange("192.168.0.0", "192.168.254.254")

for ip in ip_range:
    print(ip)
```

2. End

Revision History

Date	Name	Comment of changes
08/10/2018	T.J. Yang, tjyang2001@gmail.com	Initial notes on pdns

References

1. [centos-7-installing-and-configuring-powerdns-with-postgresql/](#)
2. Slides
 - a. <https://ds9a.nl/powerdns-denic.pdf>
 - b. End
3. PDNS frontend
 - a. <https://github.com/PowerDNS/pdns/wiki/WebFrontends>
4. Centos 7
 - a. <https://www.tecmint.com/install-powerdns-poweradmin-mariadb-in-centos-rhel/>
 - b. End
5. Command line util
 - a. <https://tailpuff.net/powerdns-script-add-new-zones-using-pdnsutil/>
 - b. End
6. Rest api
 - a. <https://stackoverflow.com/questions/42264248/inserting-a-records-using-the-powerdns-rest-api>
 - b. end
7. End.