



OpenNMS 101

Tarus Balog

tarus@opennms.com

<http://www.opennms.org/Training>

© 2017 The OpenNMS Group, Inc.

Module 1b: Installation via debs (Debian/Ubuntu)

Meridian vs. Horizon

- Both Require Java
- Both have some C++ code (jicmp, jrrd)
- Horizon – freely available downloads for apt (Debian/Ubuntu) and rpm (RHEL/CentOS)
- Meridian – password protected repository, rpm only (RHEL/CentOS)



Installation Guide

Installation Guide

Copyright (c) 2015 The OpenNMS Group, Inc. – OpenNMS 17.0.0, Last updated 2015-12-01 17:26:19 -05:00

1. Basic Installation of OpenNMS

The *OpenNMS* platform can be installed in several ways. This guide describes the installation of the platform on *RHEL*-, *Debian*- and *Microsoft Windows* based operation systems. Installable pre-compiled software packages are provided through *RPM* and *Debian* repository servers. Running *OpenNMS* requires the following components:

- Internet access to download and verify installation packages from public repository server
- Installed [Oracle Java 8](#) environment
- PostgreSQL 9.1+ data base
- Set link to section which describes to install with *RRDTool*. Optional [RRDtool](#) to persist long term performance data



OpenJDK 8 can be used, but for production and critical environments *Oracle Java 8* is recommended.



`${OPENNMS_HOME}` is referred to the path *OpenNMS* is installed to. On *RHEL-based* systems it is `/opt/opennms` on *Debian-based* systems it is `/usr/share/opennms`. The environment in *Microsoft Windows* can refer to `C:\Program Files\opennms`

With the *opennms* meta package all dependencies needed for the components mentioned above are maintained. The following sections describe how to install *OpenNMS* on a single system. Dependencies for *Java* and the *PostgreSQL* data base are maintained with the *opennms* meta installation package.

<https://docs.opennms.org/opennms/branches/develop/guide-install/guide-install.html>



Setting Up PostgreSQL

First, edit `/etc/postgresql/9.4/main/pg_hba.conf`:

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
host all all 127.0.0.1/32 trust
# IPv6 local connections:
host all all ::1/128 trust
```

Finally, restart the database for the change to take effect:

```
# sudo service postgresql restart
```

Tuning PostgreSQL

Database PostgreSQL [\[edit\]](#)

Shared Buffers [\[edit\]](#)

The default `shared_buffers` parameter in `postgresql.conf` is extremely conservative, and in most cases with modern servers, this can be significantly tweaked for a big performance boost, and drop in I/O wait time. This change will need to be in-line with kernel parameter changes to `shmmax`.

The PostgreSQL project wiki aggregates many good links in its [Performance Optimization](#) article. Among others linked from there, see [Postgres Wiki tuning page](#) and [this PostgreSQL performance page](#) for recommendations on this and other PostgreSQL settings.

If you want to put PostgreSQL on a different box then you want to change the SQL host look in `opennms-datasources.xml`. The PostgreSQL server will also need `iplike` installed and configured.

To clean up extra events out of the database try this [Event_Configuration_How-To#The_Database](#)

PostgreSQL 9.1 tuning [\[edit\]](#)

Summarized from [these blog posts](#).

For a system that has been running for some time, a good start is to determine what resources are available. Linux systems have a nice SNMP "System Memory Stats" graph to review how system memory is used.

Next, db size can be found:

```
opennms=# select pg_size_pretty(pg_database_size('opennms')) as db_size;
 db_size
-----
 691 MB
(1 row)
```

https://wiki.opennms.org/wiki/Performance_tuning#Database_PostgreSQL_2

Setting Up OpenNMS

First, as root, set the Java version (writes to /etc/opennms/java.conf):

```
# /usr/share/opennms/bin/runjava -s
runjava: Looking for an appropriate JRE...
runjava: Checking for an appropriate JRE in JAVA_HOME...
runjava: skipping... JAVA_HOME not set
runjava: Checking JRE in user's path: "/bin/java"...
runjava: found an appropriate JRE in user's path: "/bin/java"
runjava: value of "/bin/java" stored in configuration file
```

Next, run the installer (creates file /opt/opennms/etc/configured):

```
# /usr/share/opennms/bin/install -dis
```

```
=====
OpenNMS Installer
=====
```

```
Configures PostgreSQL tables, users, and other miscellaneous settings.
```

```
Upgrade completed successfully!
```



Uses of `iplike`

- `iplike` is a stored procedure, installed separately
- It allows for quick IP Address comparisons:
 - Can use wildcards
 - Can use ranges
 - Can use combinations
- Examples:
 - `10.10.1.*` (matches all in the 10.10.1.0 subnet)
 - `10.10.1.1-50` (matches the first 50 IP addresses)
 - `10.10.1,5,11-15.*` (matches all in the 10.10.1.0, 10.10.5.0 and 10.10.11.0 through 10.10.15.0 networks)
- In `psql`, use it as `iplike(ipaddr, 'pattern')`



Install `iplike` and Start OpenNMS

Finally, install the `iplike` stored procedure:

```
# install_iplike.sh
```

Then start OpenNMS:

```
# sudo service opennms start
```

Then you should be able to access the web page at:

```
http://localhost:8980/opennms
```