Apache Performance using Netdata on CentOS 7

Netdata is a free open source, simple yet powerful, and effective real-time system performance monitoring tool for Linux, FreeBSD and MacOS. It supports various plugins for monitoring general server status, applications, web services such as Apache or Nginx HTTP server and so much more.

Requirements:

1. A CentOS 7 Server or RHEL 7 Server with Minimal Install.
2. Apache HTTP server installation with mod_status module enabled.

Step 1: Install Apache on CentOS 7

1. First start by installing Apache HTTP server from the default software repositories using the YUM package manager.

```
# yum install httpd
```

2. After you have installed Apache web server, start it for the first time, check if it is up and running, and enable it to start automatically at system boot using following commands.

```
# systemctl start httpd
# systemctl enable httpd
# systemctl status httpd
```

3. If you are running a firewall for example firewalld, you need to open the ports 80 and 443 to allow web traffic to Apache via HTTP and HTTPS respectively, using the commands below.

```
# firewall-cmd --zone=public --permanent --add-port=80/tcp
# firewall-cmd --zone=public --permanent --add-port=443/tcp
# firewall-cmd --reload
```

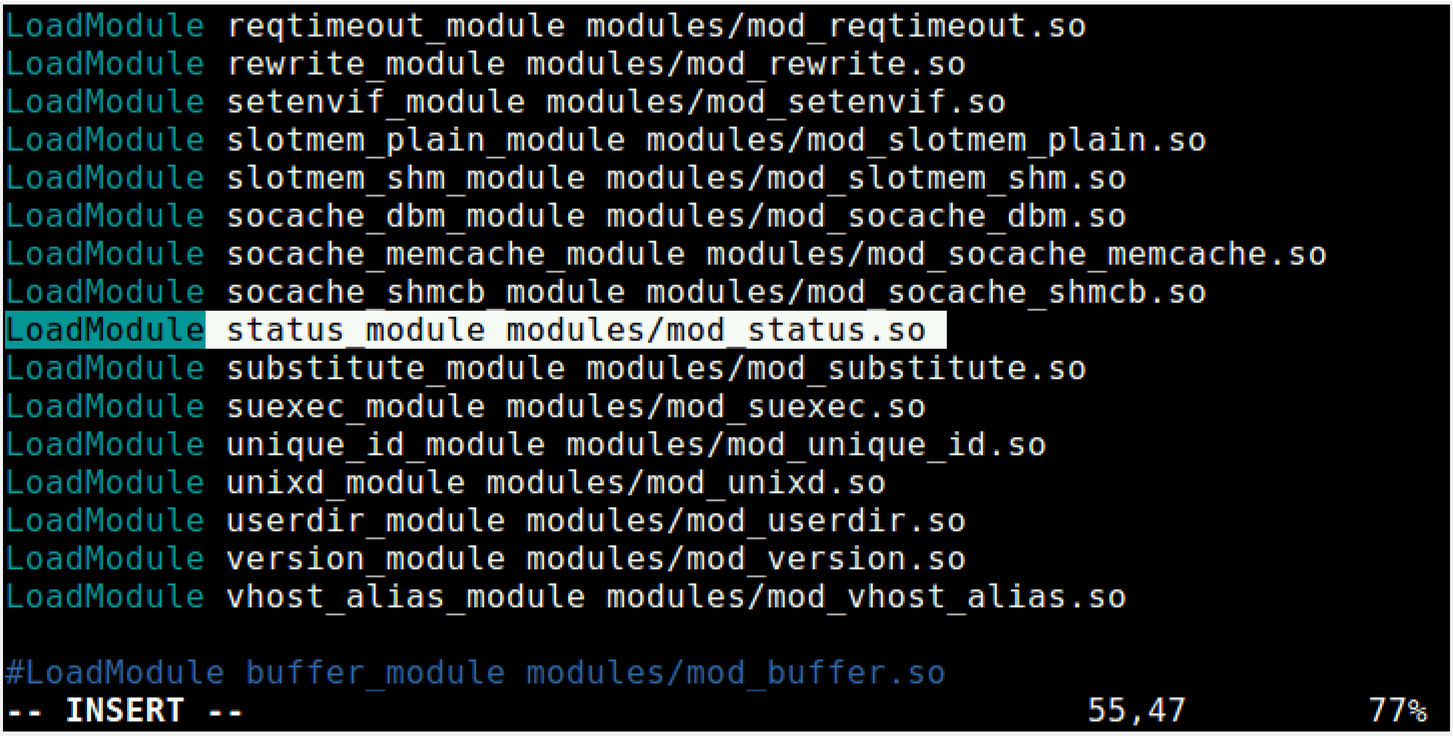
Step 2: Enable Mod_Status Module in Apache

4. In this step, you need to enable and configure mod_status module in Apache, this is required by Netdata for gathering server status information and statistics.

Open the file /etc/httpd/conf.modules.d/00-base.conf file using your favorite editor.

```
# vim /etc/httpd/conf.modules.d/00-base.conf
```

And ensure that the line below is uncommented to enable mod_status module, as shown in the screenshot.



```
LoadModule reqtimeout_module modules/mod_reqtimeout.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule slotmem_plain_module modules/mod_slotmem_plain.so
LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
LoadModule socache_dbm_module modules/mod_socache_dbm.so
LoadModule socache_memcache_module modules/mod_socache_memcache.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
LoadModule status_module modules/mod_status.so
LoadModule substitute_module modules/mod_substitute.so
LoadModule suexec_module modules/mod_suexec.so
LoadModule unique_id_module modules/mod_unique_id.so
LoadModule unixd_module modules/mod_unixd.so
LoadModule userdir_module modules/mod_userdir.so
LoadModule version_module modules/mod_version.so
LoadModule vhost_alias_module modules/mod_vhost_alias.so

#LoadModule buffer_module modules/mod_buffer.so
-- INSERT --                                              55,47          77%
```

Enable Mod_Status Module in Apache

5. Once you've enabled mod_status, next you need to create a `server-status.conf` configuration file for the Apache server status page.

```
# vim /etc/httpd/conf.d/server-status.conf
```

Add the following configuration inside the file.

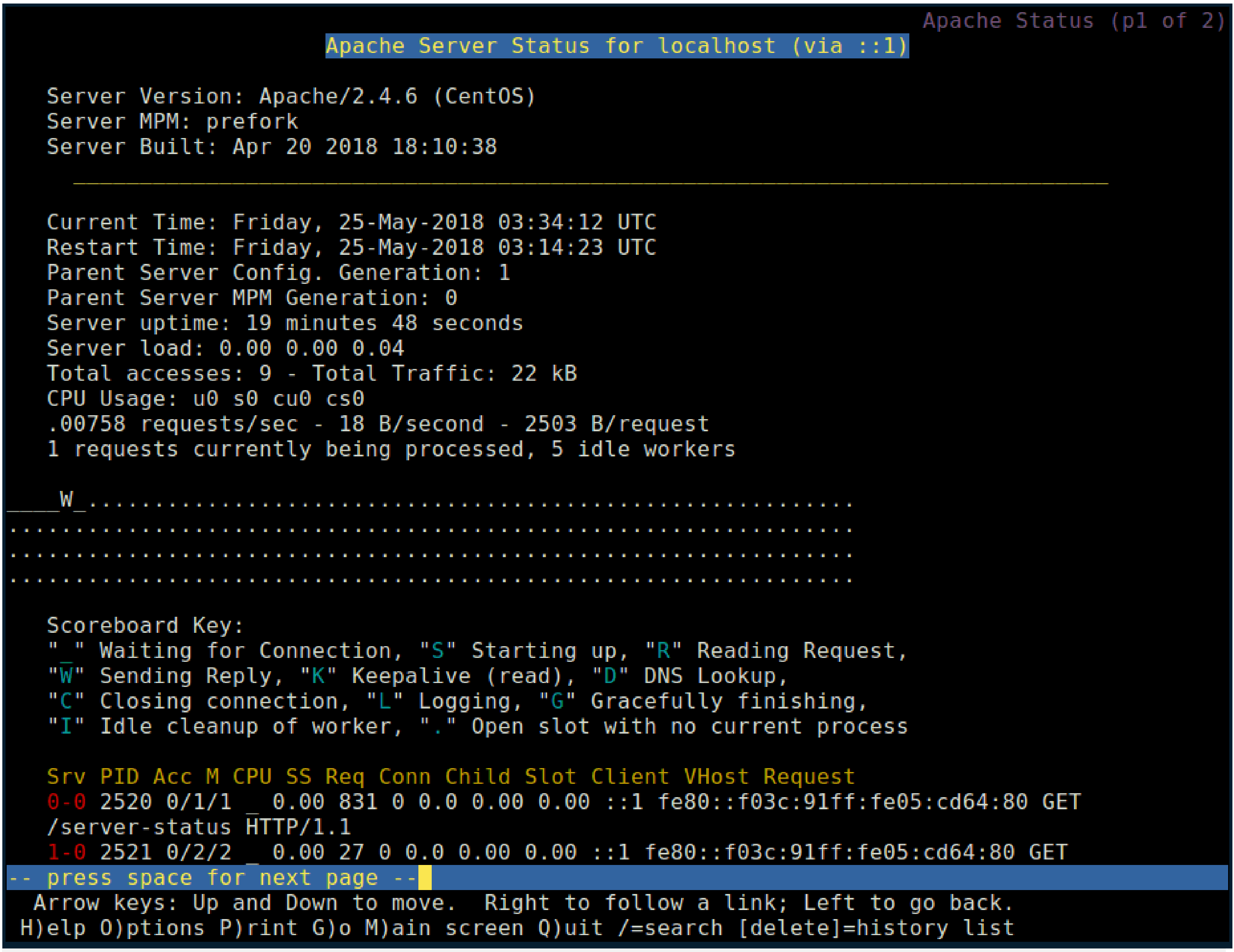
```
<Location "/server-status">
    SetHandler server-status
    #Require host localhost           #uncomment to only allow requests
from localhost
</Location>
```

Save the file and close. Then restart the Apache HTTPD service.

```
# systemctl restart httpd
```

6. Next, you need to verify that the Apache server status and statistics page is working well by using a command-line web browser such as lynx as shown.

```
# yum install lynx
# lynx http://localhost/server-status
```



```
                                                              Apache Status (p1 of 2)
                         Apache Server Status for localhost (via ::1)

   Server Version: Apache/2.4.6 (CentOS)
   Server MPM: prefork
   Server Built: Apr 20 2018 18:10:38

   Current Time: Friday, 25-May-2018 03:34:12 UTC
   Restart Time: Friday, 25-May-2018 03:14:23 UTC
   Parent Server Config. Generation: 1
   Parent Server MPM Generation: 0
   Server uptime: 19 minutes 48 seconds
   Server load: 0.00 0.00 0.04
   Total accesses: 9 - Total Traffic: 22 kB
   CPU Usage: u0 s0 cu0 cs0
   .00758 requests/sec - 18 B/second - 2503 B/request
   1 requests currently being processed, 5 idle workers

____W_..........................................................
................................................................
................................................................
................................................................

   Scoreboard Key:
   "_" Waiting for Connection, "S" Starting up, "R" Reading Request,
   "W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
   "C" Closing connection, "L" Logging, "G" Gracefully finishing,
   "I" Idle cleanup of worker, "." Open slot with no current process

   Srv PID Acc M CPU SS Req Conn Child Slot Client VHost Request
   0-0 2520 0/1/1 _ 0.00 831 0 0.0 0.00 0.00 ::1 fe80::f03c:91ff:fe05:cd64:80 GET
   /server-status HTTP/1.1
   1-0 2521 0/2/2 _ 0.00 27 0 0.0 0.00 0.00 ::1 fe80::f03c:91ff:fe05:cd64:80 GET
-- press space for next page --
  Arrow keys: Up and Down to move.  Right to follow a link; Left to go back.
 H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

Check Apache Server Status

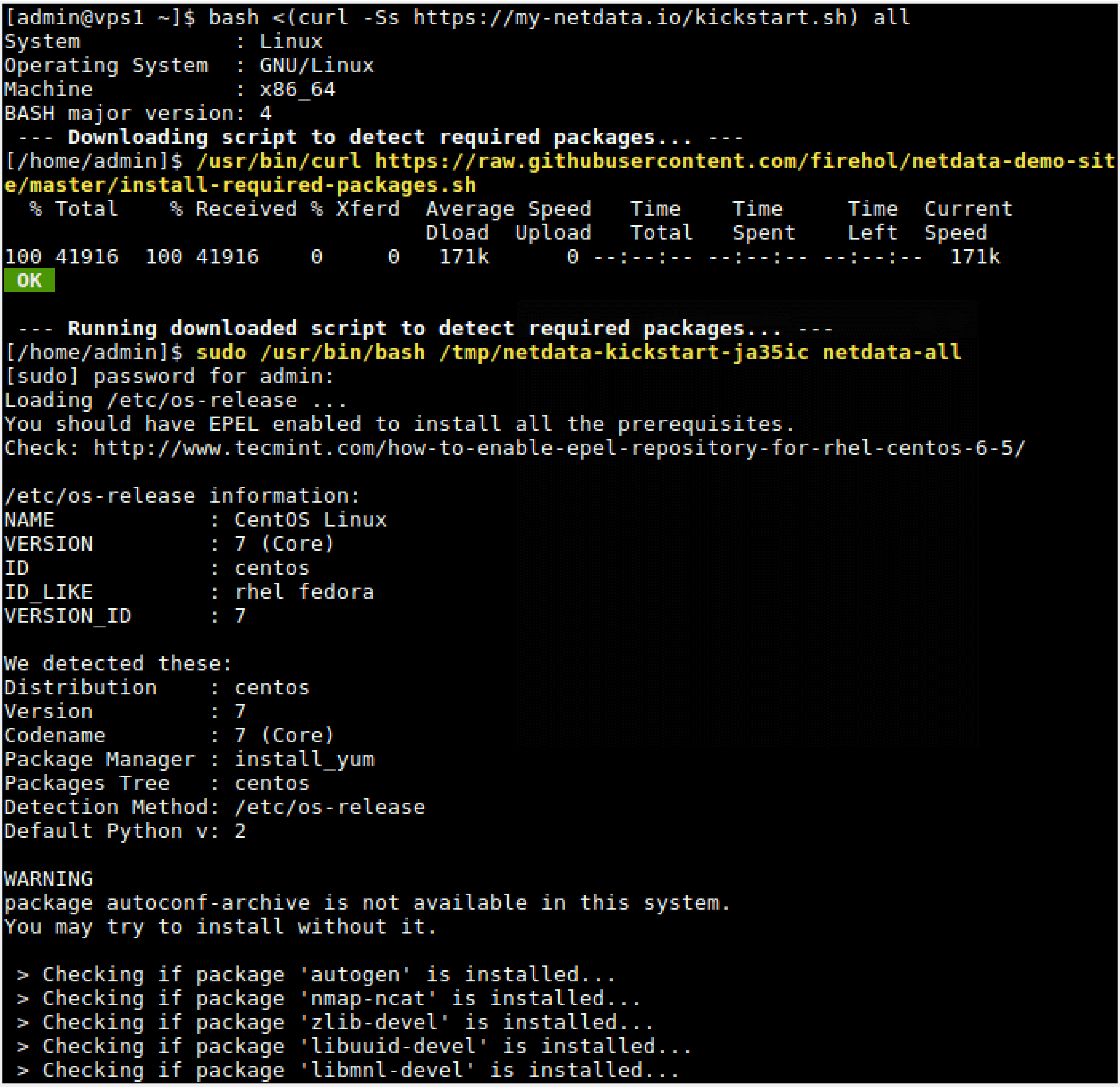
Step 3: Install Netdata on CentOS 7

7. Fortunately, there is a kickstarter shell script for painlessly installing netdata from its github repository. This one-liner script downloads a second script which checks your Linux distribution and installs the required system packages for building netdata, then downloads the latest netdata source tree; builds and installs it on your server.

You can start the kickstarter script as shown, the all flag allows for installing required packages for all netdata plugins including the ones for Apache HTTP server.

```
# bash <(curl -Ss https://my-netdata.io/kickstart.sh) all
```

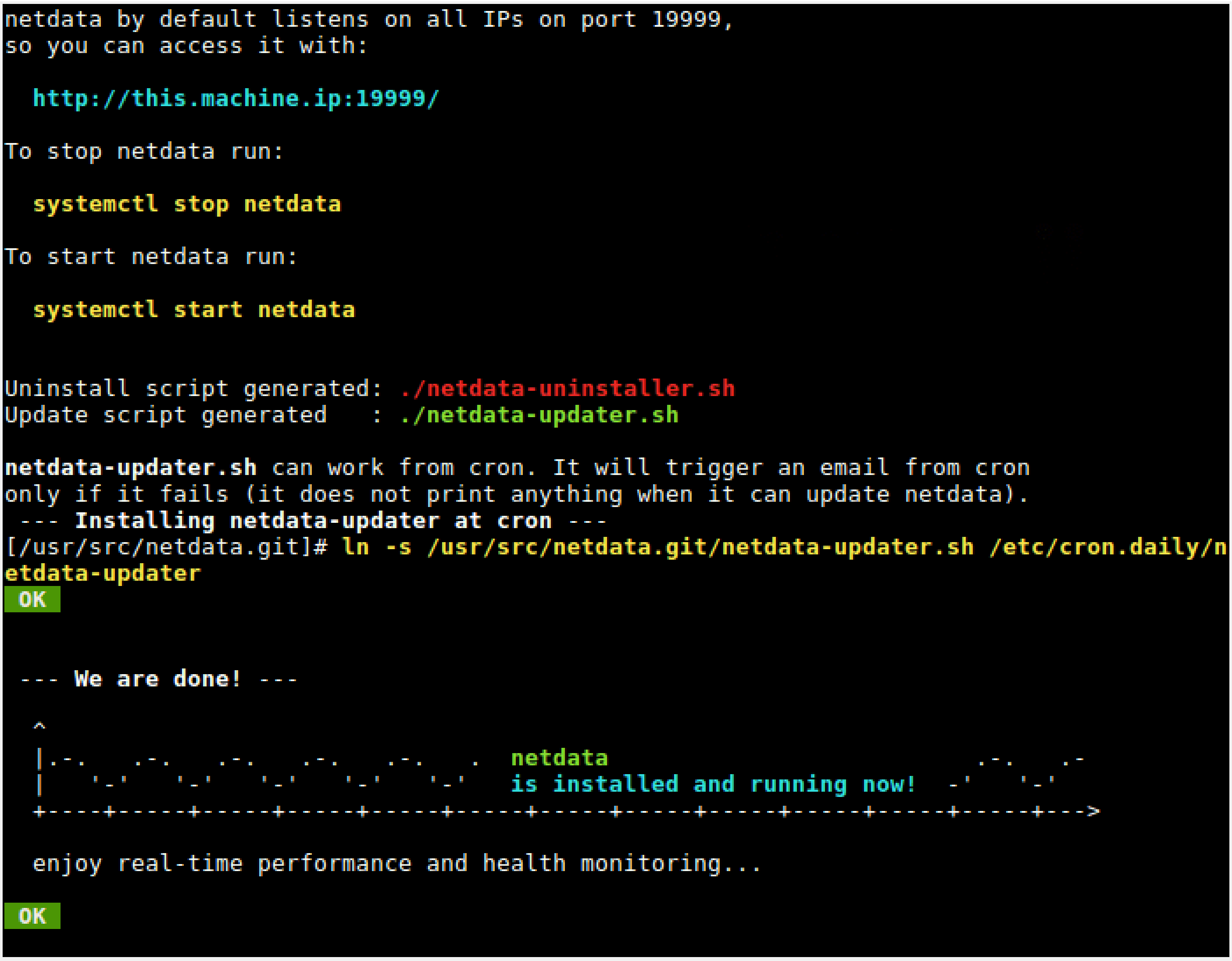
Note that if your not administering your system as root, you will be prompted to enter your user password for sudo command, and you will also be asked to confirm a number of functions by pressing [Enter].



```
[admin@vps1 ~]$ bash <(curl -Ss https://my-netdata.io/kickstart.sh) all
System            : Linux
Operating System  : GNU/Linux
Machine           : x86_64
BASH major version: 4
 --- Downloading script to detect required packages... ---
[/home/admin]$ /usr/bin/curl https://raw.githubusercontent.com/firehol/netdata-demo-sit
e/master/install-required-packages.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 41916  100 41916    0     0   171k      0 --:--:-- --:--:-- --:--:--  171k
 OK

 --- Running downloaded script to detect required packages... ---
[/home/admin]$ sudo /usr/bin/bash /tmp/netdata-kickstart-ja35ic netdata-all
[sudo] password for admin:
Loading /etc/os-release ...
You should have EPEL enabled to install all the prerequisites.
Check: http://www.tecmint.com/how-to-enable-epel-repository-for-rhel-centos-6-5/

/etc/os-release information:
NAME            : CentOS Linux
VERSION         : 7 (Core)
ID              : centos
ID_LIKE         : rhel fedora
VERSION_ID      : 7

We detected these:
Distribution    : centos
Version         : 7
Codename        : 7 (Core)
Package Manager : install_yum
Packages Tree   : centos
Detection Method: /etc/os-release
Default Python v: 2

WARNING
package autoconf-archive is not available in this system.
You may try to install without it.

 > Checking if package 'autogen' is installed...
 > Checking if package 'nmap-ncat' is installed...
 > Checking if package 'zlib-devel' is installed...
 > Checking if package 'libuuid-devel' is installed...
 > Checking if package 'libmnl-devel' is installed...
```

Install Netdata on CentOS 7

8. Once the script has completed building and installing netdata, it will automatically start the netdata service via systemd service manager and enables it to start at system boot.



```
netdata by default listens on all IPs on port 19999,
so you can access it with:

  http://this.machine.ip:19999/

To stop netdata run:

  systemctl stop netdata

To start netdata run:

  systemctl start netdata


Uninstall script generated: ./netdata-uninstaller.sh
Update script generated   : ./netdata-updater.sh

netdata-updater.sh can work from cron. It will trigger an email from cron
only if it fails (it does not print anything when it can update netdata).
 --- Installing netdata-updater at cron ---
[/usr/src/netdata.git]# ln -s /usr/src/netdata.git/netdata-updater.sh /etc/cron.daily/n
etdata-updater
 OK


 --- We are done! ---

  ^
  |.-.   .-.   .-.   .-.   .-.   .  netdata                          .-.   .-
  |   '-'   '-'   '-'   '-'   '-'   is installed and running now!  -'   '-'
  +----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+--->

  enjoy real-time performance and health monitoring...

 OK
```

Netdata Installation Summary

By default, netdata listens on port 19999, you will access the web UI using this port. So, open port 19999 in the firewall to access the netdata web UI.

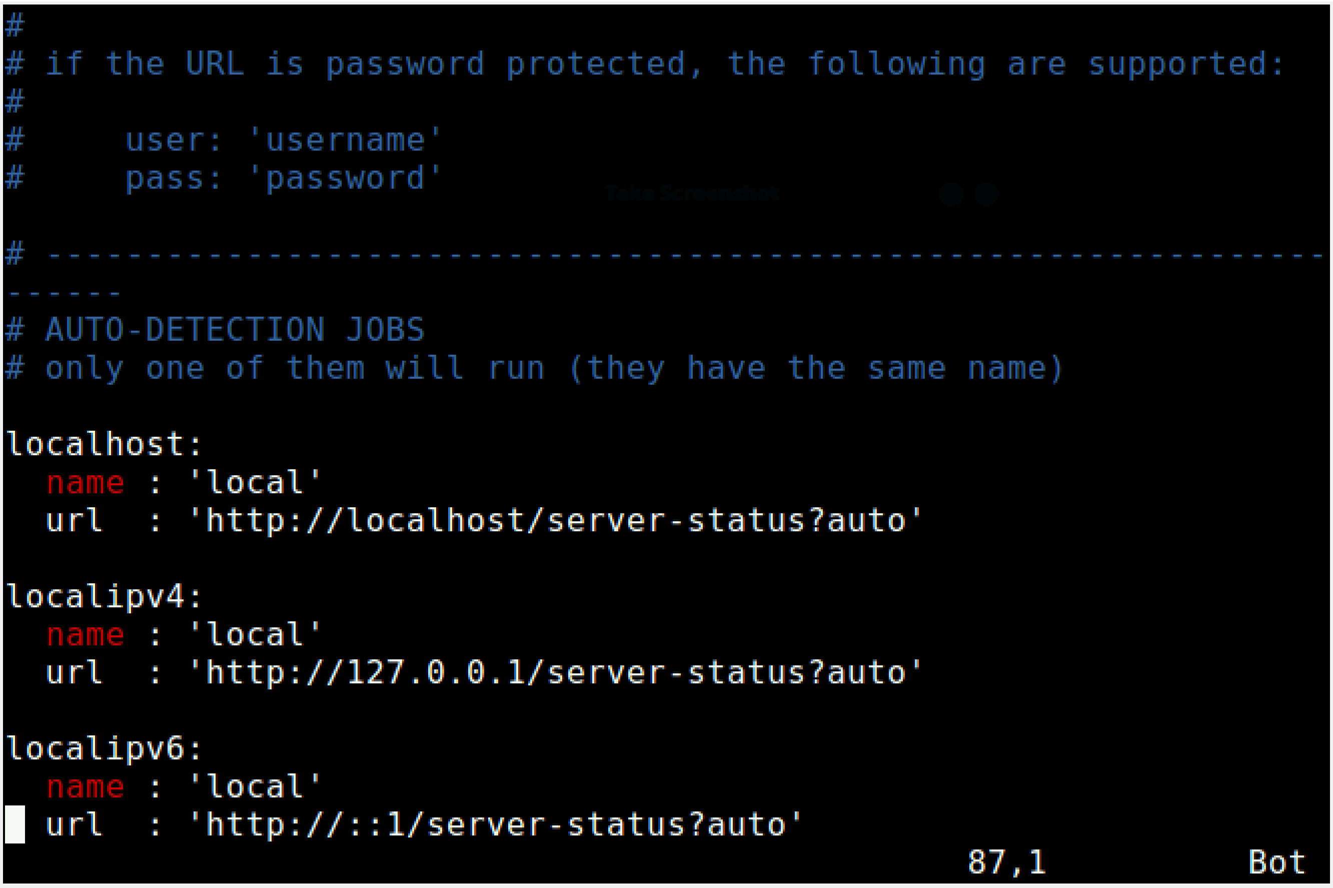
```
# firewall-cmd --permanent --add-port=19999/tcp
# firewall-cmd --reload
```

Step 4: Configure Netdata to Monitor Apache Performance

9. The netdata configuration for Apache plugin is /etc/netdata/python.d/apache.conf, this file is written in YaML format, you can open it using your favorite editor.

```
# vim /etc/netdata/python.d/apache.conf
```

The default configuration is just enough to get you started with monitoring your Apache HTTP server.



```
#
# if the URL is password protected, the following are supported:
#
#     user: 'username'
#     pass: 'password'

# ------------------------------------------------------------------
------
# AUTO-DETECTION JOBS
# only one of them will run (they have the same name)

localhost:
  name : 'local'
  url  : 'http://localhost/server-status?auto'

localipv4:
  name : 'local'
  url  : 'http://127.0.0.1/server-status?auto'

localipv6:
  name : 'local'
  url  : 'http://::1/server-status?auto'
                                                      87,1          Bot
```

Netdata Configuration for Apache

However, if you have read the documentation, and made any changes to it, restart the netdata service to effect the changes.

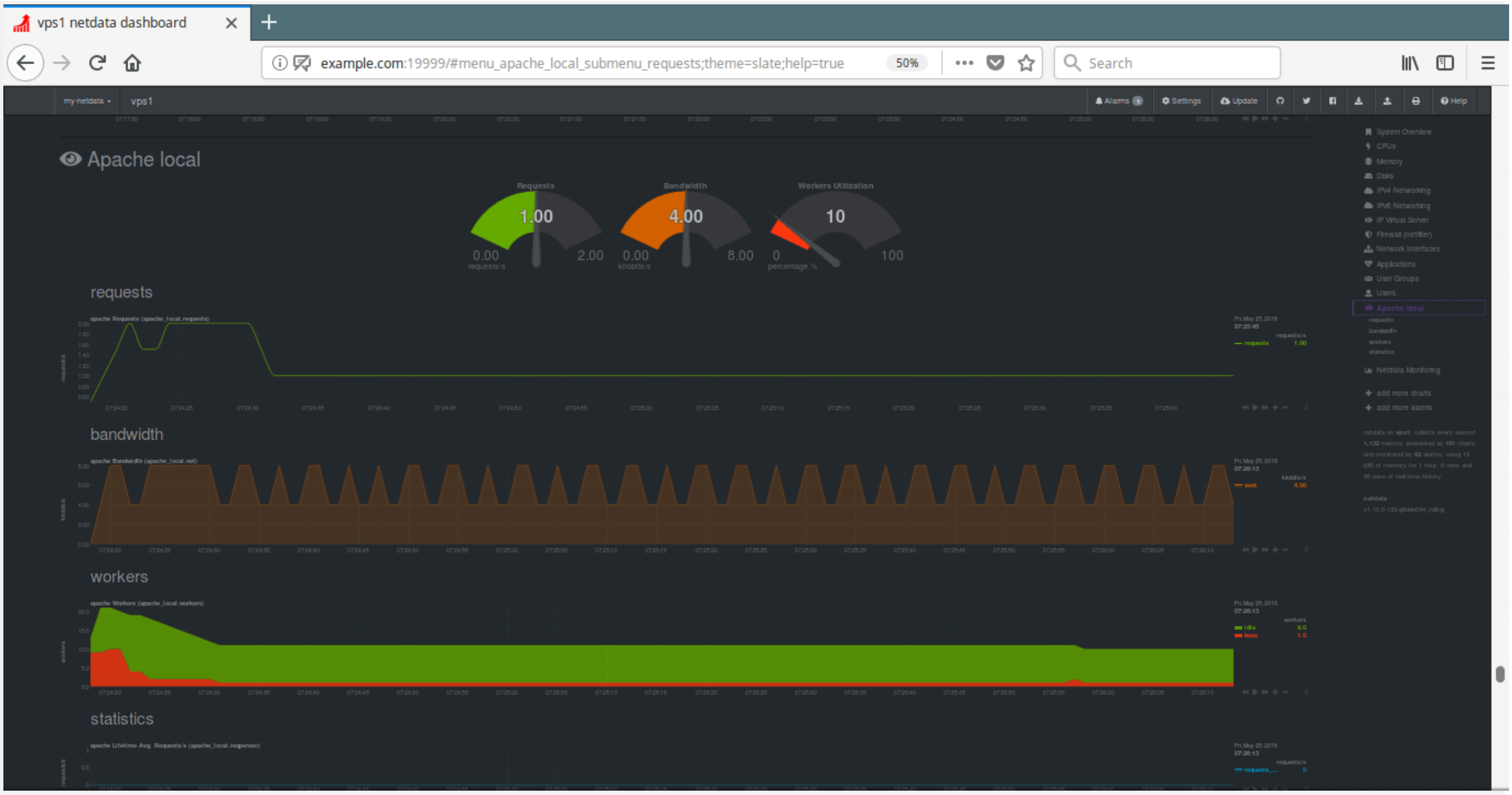
```
# systemctl restart netdata
```

Step 5: Monitor Apache Performance Using Netdata

10. Next, open a web browser and use the following URL to access the netdata web UI.

```
http://domain_name:19999
OR
http://SERVER_IP:19999
```

From the netdata dashboard, search for “Apache local” on the right hand side list of plugins, and click on it to start monitoring your Apache server. You will be able to watch visualizations of requests, bandwidth, workers, and other server statistics, as shown in the following screenshot.



Monitor Apache Performance Using Netdata

Install NetData Performance Monitoring Tool On Linux

Install NetData On Linux

Netdata can be installed on any Linux distributions that have Bash installed.

The easiest way to install Netdata is to run the following one-liner command from the Terminal:

```
#  bash <(curl -Ss https://my-netdata.io/kickstart-static64.sh) all
```

```
bash <(curl -Ss https://my-netdata.io/kickstart.sh) all
```

This will download and install everything needed to up and run Netdata.

Some users may not want to inject something directly into Bash without investigating it. If you don't like this method, you can follow the steps below to install it on your system.

On Arch Linux:

The latest version is available in the Arch Linux default repositories. So, we can install it with pacman using command:

```
$ sudo pacman -S netdata
```

On DEB and RPM-based systems

NetData is not available in the default repositories of DEB based (Ubuntu / Debian) or RPM based (RHEL / CentOS / Fedora) systems. We need to install NetData manually from its Git repository.

First install the required dependencies:

```
# Debian / Ubuntu
$ sudo apt-get install zlib1g-dev uuid-dev libuv1-dev liblz4-dev
libjudy-dev libssl-dev libmnl-dev gcc make git autoconf autoconf-archive
autogen automake pkg-config curl

# Fedora
$ sudo dnf install zlib-devel libuuid-devel libuv-devel lz4-devel
Judy-devel openssl-devel libmnl-devel gcc make git autoconf
autoconf-archive autogen automake pkgconfig curl findutils

# CentOS / Red Hat Enterprise Linux
$ sudo yum install epel-release
$ sudo yum install autoconf automake curl gcc git libmnl-devel
libuuid-devel openssl-devel libuv-devel lz4-devel Judy-devel lm_sensors
make MySQL-python nc pkgconfig python python-psycopg2 PyYAML zlib-devel

# openSUSE
$ sudo zypper install zlib-devel libuuid-devel libuv-devel liblz4-devel
judy-devel openssl-devel libmnl-devel gcc make git autoconf
autoconf-archive autogen automake pkgconfig curl findutils
```

After installing the required dependencies, install NetData on DEB or RPM based systems as shown below.

Git clone the NetData repository:

```
$ git clone https://github.com/netdata/netdata.git --depth=100
```

The above command will create a directory called ‘netdata’ in the current working directory.

Change to the ‘netdata’ directory:

```
$ cd netdata/
```

Finally, install and start NetData using command:

```
$ sudo ./netdata-installer.sh
```

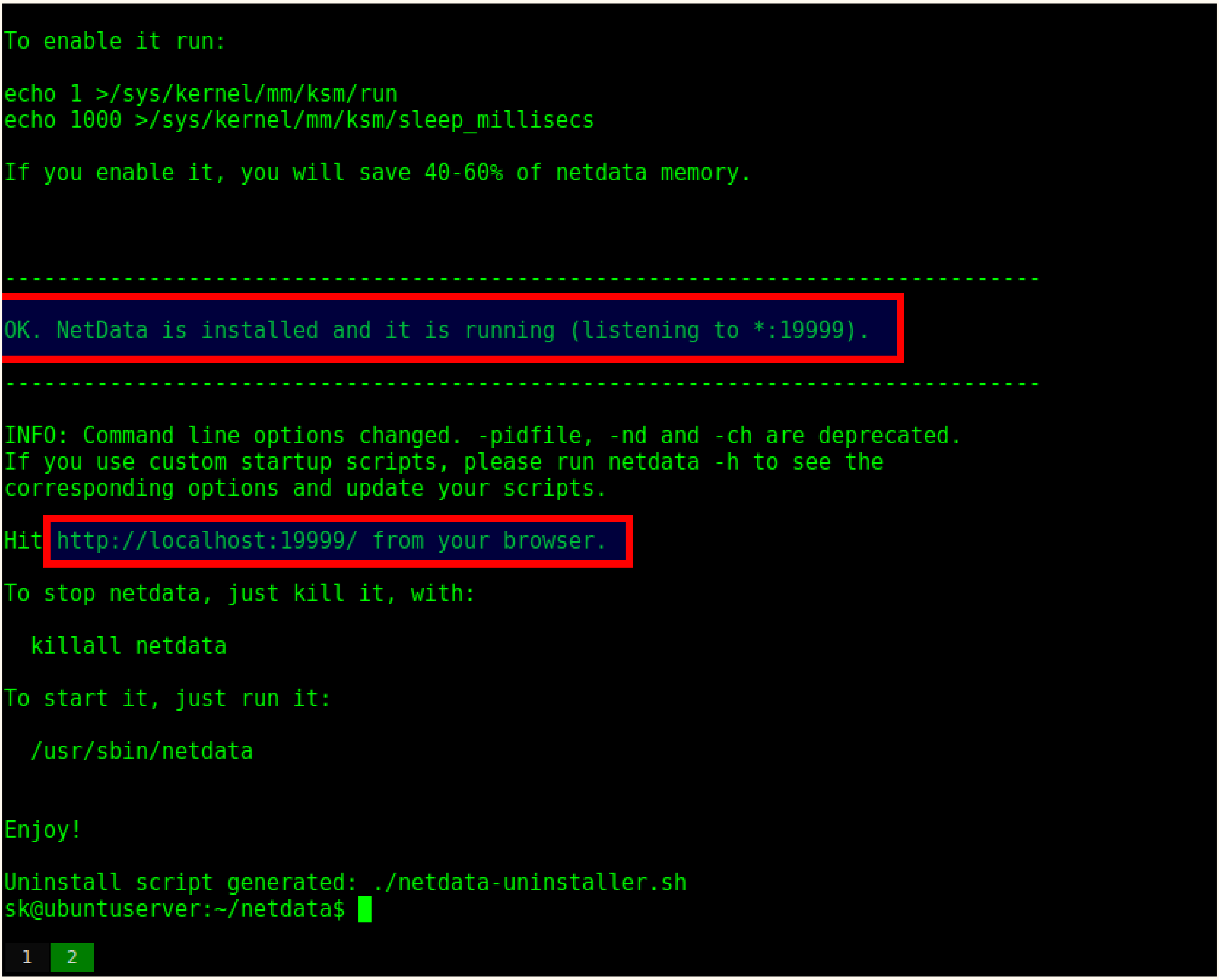
Sample output:

```
Welcome to netdata!
Nice to see you are giving it a try!

You are about to build and install netdata to your system.

It will be installed at these locations:

- the daemon at /usr/sbin/netdata
 - config files at /etc/netdata
 - web files at /usr/share/netdata
 - plugins at /usr/libexec/netdata
 - cache files at /var/cache/netdata
 - db files at /var/lib/netdata
 - log files at /var/log/netdata
 - pid file at /var/run
```

```
This installer allows you to change the installation path.
Press Control-C and run the same command with --help for help.


Press ENTER to build and install netdata to your system > ## Press ENTER key
```

After installing NetData, you will see the following output at the end:

```
--------------------------------------------------------------------------------------

OK. NetData is installed and it is running (listening to *:19999).

--------------------------------------------------------------------------------------

INFO: Command line options changed. -pidfile, -nd and -ch are deprecated.
If you use custom startup scripts, please run netdata -h to see the corresponding options and update your scripts.

Hit http://localhost:19999/ from your browser.

To stop netdata, just kill it, with:

killall netdata

To start it, just run it:

/usr/sbin/netdata



Enjoy!


Uninstall script generated: ./netdata-uninstaller.sh
```



```
To enable it run:

echo 1 >/sys/kernel/mm/ksm/run
echo 1000 >/sys/kernel/mm/ksm/sleep_millisecs

If you enable it, you will save 40-60% of netdata memory.



--------------------------------------------------------------------------------

OK. NetData is installed and it is running (listening to *:19999).

--------------------------------------------------------------------------------

INFO: Command line options changed. -pidfile, -nd and -ch are deprecated.
If you use custom startup scripts, please run netdata -h to see the
corresponding options and update your scripts.

Hit http://localhost:19999/ from your browser.

To stop netdata, just kill it, with:

  killall netdata

To start it, just run it:

  /usr/sbin/netdata


Enjoy!

Uninstall script generated: ./netdata-uninstaller.sh
sk@ubuntuserver:~/netdata$
```

Install NetData

NetData has been installed and started.

To install Netdata on other Linux distributions, refer the official installation instructions page.

Allow NetData default port via Firewall or Router

If your system stays behind any firewall or router, you must allow the default port 19999 to access the NetData web interface from any remote systems on the network,.

On Ubuntu / Debian:

```
$ sudo ufw allow 19999
```

On CentOS / RHEL / Fedora:

```
$ sudo firewall-cmd --permanent --add-port=19999/tcp
$ sudo firewall-cmd --reload
```

Starting / Stopping NetData

To enable and start Netdata service on systems that use Systemd, run:

```
$ sudo systemctl enable netdata
```

```
$ sudo systemctl start netdata
```

To stop:

```
$ sudo systemctl stop netdata
```

To enable and start Netdata service on systems that use Init, run:

```
$ sudo service netdata start
```

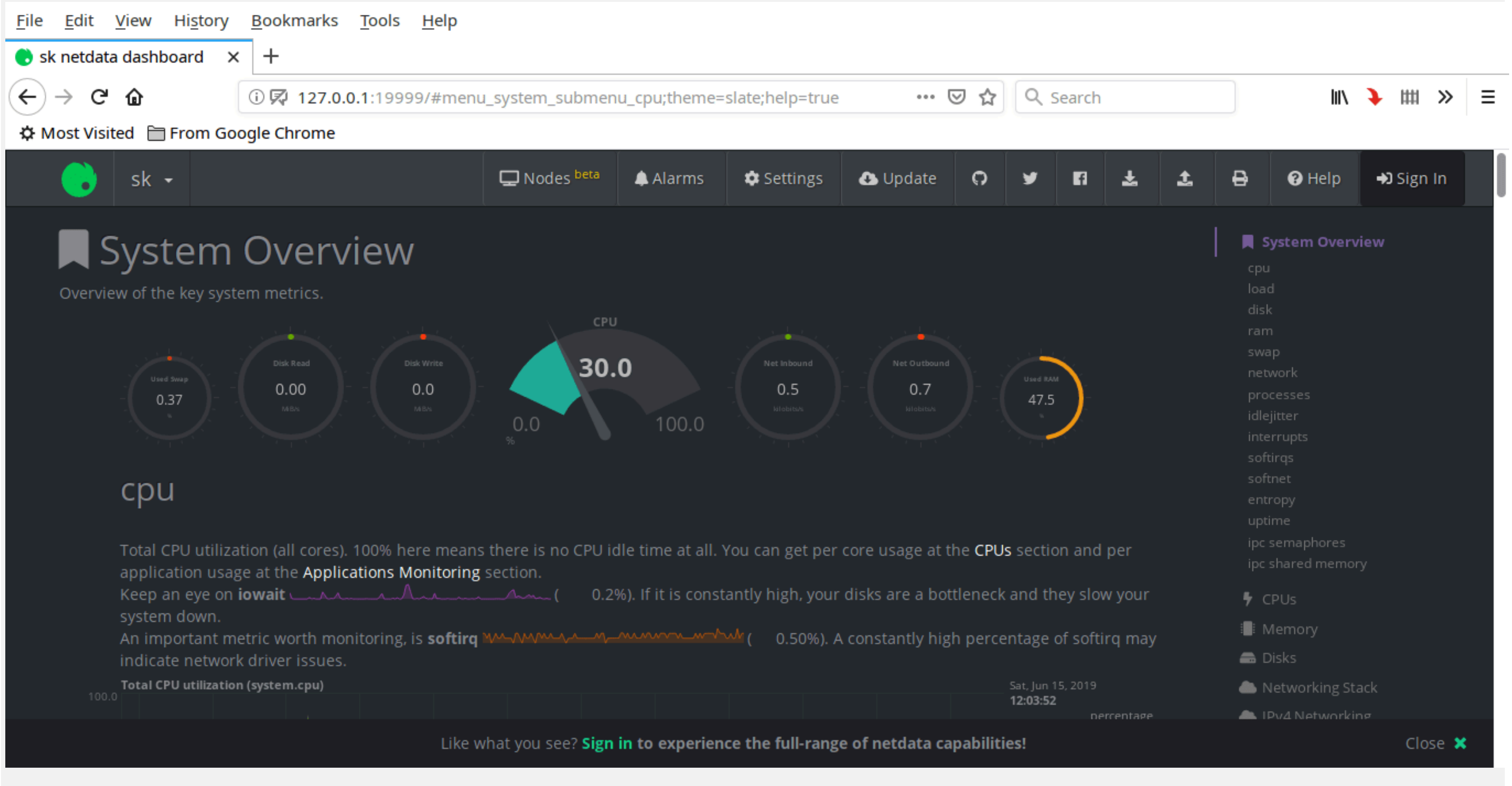
```
$ sudo chkconfig netdata on
```

To stop it:

```
$ sudo service netdata stop
```

Access NetData via Web browser

Open your web browser, and navigate to http://127.0.0.1:19999 or http://localhost:19999/ or http://ip-address:19999. You should see a screen something like below.



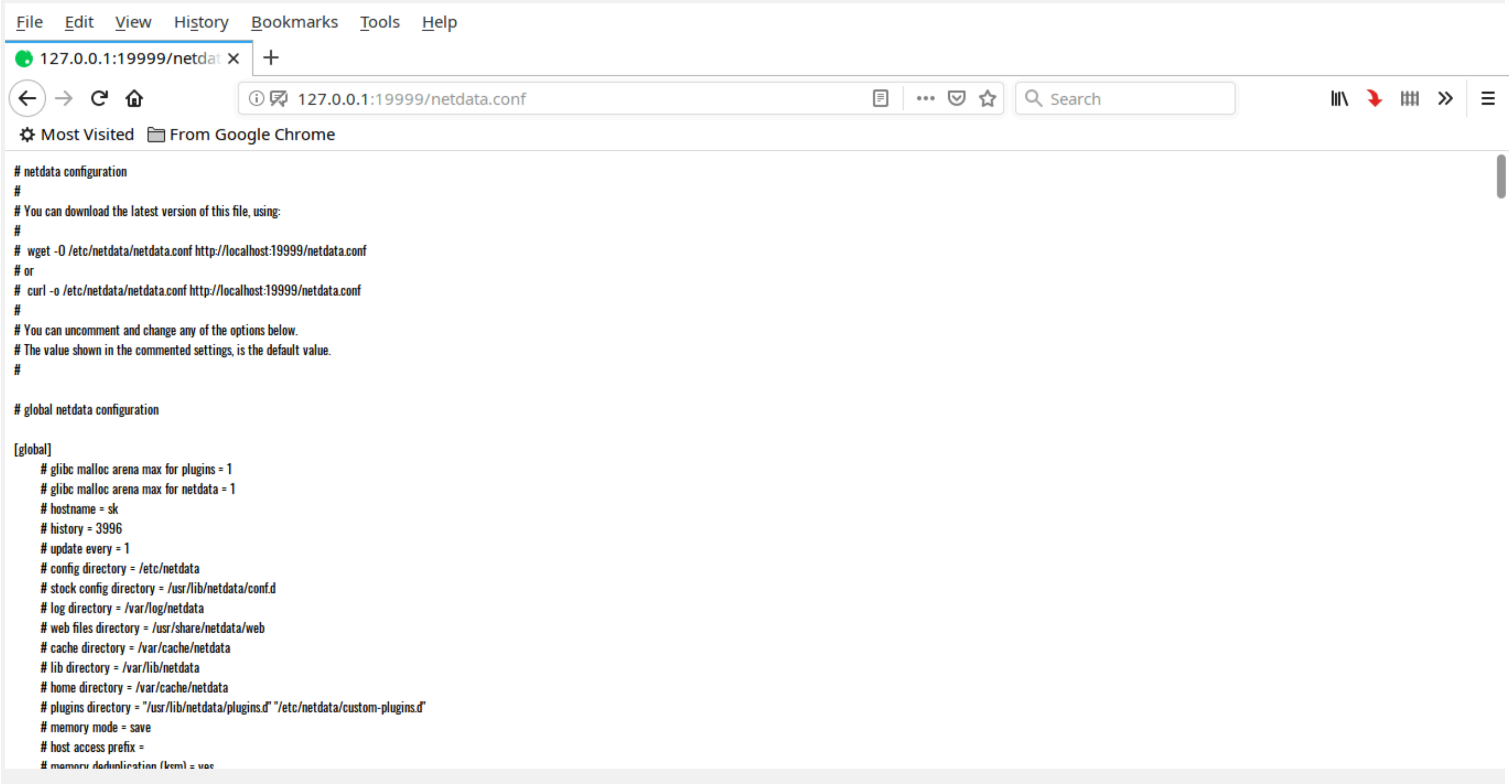
Total CPU utilization (all cores). 100% here means there is no CPU idle time at all. You can get per core usage at the **CPUs** section and per application usage at the **Applications Monitoring** section.
Keep an eye on **iowait** (0.2%). If it is constantly high, your disks are a bottleneck and they slow your system down.
An important metric worth monitoring, is **softirq** (0.50%). A constantly high percentage of softirq may indicate network driver issues.

Like what you see? **Sign in** to experience the full-range of netdata capabilities!

Netdata dashboard

From the dashboard, you will find the complete statistics of your Linux system. Scroll down to view each section.

You can download and/or view NetData default configuration file at any time by simply navigating to http://localhost:19999/netdata.conf.


File Edit View History Bookmarks Tools Help
127.0.0.1:19999/netdat
127.0.0.1:19999/netdata.conf
Search
Most Visited
From Google Chrome

```
# netdata configuration
#
# You can download the latest version of this file, using:
#
#  wget -O /etc/netdata/netdata.conf http://localhost:19999/netdata.conf
# or
#  curl -o /etc/netdata/netdata.conf http://localhost:19999/netdata.conf
#
# You can uncomment and change any of the options below.
# The value shown in the commented settings, is the default value.
#

# global netdata configuration

[global]
	# glibc malloc arena max for plugins = 1
	# glibc malloc arena max for netdata = 1
	# hostname = sk
	# history = 3996
	# update every = 1
	# config directory = /etc/netdata
	# stock config directory = /usr/lib/netdata/conf.d
	# log directory = /var/log/netdata
	# web files directory = /usr/share/netdata/web
	# cache directory = /var/cache/netdata
	# lib directory = /var/lib/netdata
	# home directory = /var/cache/netdata
	# plugins directory = "/usr/lib/netdata/plugins.d" "/etc/netdata/custom-plugins.d"
	# memory mode = save
	# host access prefix =
	# memory deduplication (ksm) = yes
```

Netdata configuration file

Updating NetData

In Arch Linux, just run the following command to update NetData. If the updated version is available in the repository, it will be automatically installed.

```
$ sudo pacman -Syyu
```

In DEB or RPM based systems, just go to the directory where you have cloned it (In our case it's netdata).

```
$ cd netdata
```

Pull the latest update:

```
$ git pull
```

Then, rebuild and update it using command:

```
$ sudo ./netdata-installer.sh
```

Suggested read:

- vnStat – Monitor Network Bandwidth In Linux and BSD
- How To Limit Network Bandwidth In Linux Using Wondershaper

Uninstalling NetData

Go to the location where you have cloned NetData.

```
$ cd netdata
```

Then, uninstall it using command:

```
$ sudo ./netdata-uninstaller.sh --force
```

In Arch Linux, the following command will uninstall it.

```
$ sudo pacman -Rns netdata
```

Resources:

- NetData website

- NetData GitHub page