



NetApp ToolChest

# NetApp Harvest Installation and Administration Guide 1.3

Christopher Madden, NetApp  
10 November 2016

## **Abstract**

This guide discusses installation and administration steps for the NetApp Harvest data collector.

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction to NetApp Harvest</b>	<b>4</b>
1.1	About NetApp Harvest	4
1.2	NetApp Harvest Features	4
1.3	NetApp Harvest Theory of Operation	5
1.4	Getting started with NetApp Harvest	5
<b>2</b>	<b>System Requirements</b>	<b>6</b>
2.1	Platform	6
2.2	Interoperability	7
2.3	Software	7
2.4	Protocol and port requirements	7
<b>3</b>	<b>Installing NetApp Harvest</b>	<b>9</b>
3.1	RPM based installation (RHEL, CentOS, etc)	9
3.2	DEB based installation (Ubuntu, Debian, etc)	11
3.3	Source based installation	12
<b>4</b>	<b>Upgrading NetApp Harvest</b>	<b>14</b>
4.1	RPM based upgrade (RHEL, CentOS, etc)	14
4.2	DEB based upgrade (Ubuntu, Debian, etc)	15
4.3	Source based upgrade	15
<b>4.</b>	<b>Preparing Monitored Systems</b>	<b>17</b>
a.	Enabling HTTPS and TLS (Data ONTAP 7-mode only)	17
b.	Creating user accounts	17
<b>5.</b>	<b>Configuring NetApp Harvest (basic)</b>	<b>21</b>
a.	Populate the configuration file	21
<b>6.</b>	<b>Integrating with Graphite</b>	<b>23</b>
a.	Setting frequency and retention in storage-schemas.conf	23
b.	Increase the maximum creates per minute in carbon.conf	23
c.	Use a blacklist to prevent visibility of temporary volumes [optional]	24
d.	Purging inactive metrics from Graphite [optional]	25
<b>7.</b>	<b>Integrating with Grafana</b>	<b>26</b>
a.	Creating a Grafana API key and configuration in Harvest	26
b.	Importing dashboards	28
c.	Exporting dashboards	29
<b>8.</b>	<b>Managing NetApp Harvest workers</b>	<b>30</b>
a.	Using netapp-manager	30
b.	Using netapp-worker	32

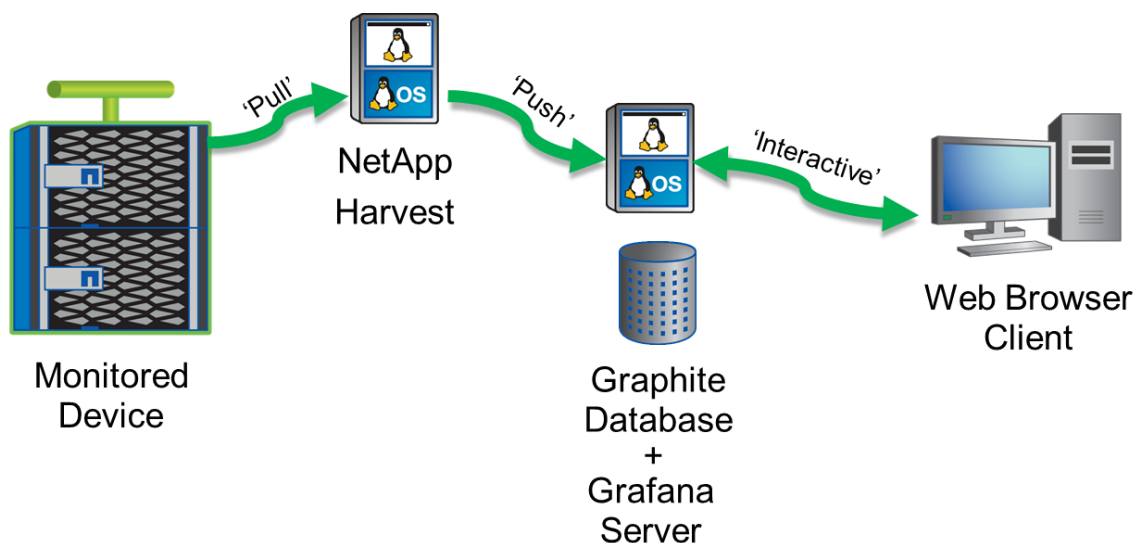
<b>9. Configuring NetApp Harvest (advanced)</b> .....	<b>33</b>
a. Populate the configuration file (advanced) .....	33
b. Working with collection templates (advanced) .....	37
c. OCUM capacity metrics with the OPM performance metrics data provider .....	40
d. perf-counters-utility .....	40
e. perf-counters-to-excel .....	41
<b>10. Threshold configuration</b> .....	<b>42</b>
<b>11. Troubleshooting</b> .....	<b>43</b>
a. "Can't locate NaServer.pm in @INC (you may need to install the NaServer module)" .....	43
b. "Required parameter [hostname] not defined" .....	44
c. "Unable to resolve hostname" .....	44
d. "Update of system-info cache DOT Version failed with reason: in Zapi::invoke, cannot connect to socket" .....	44
e. "Update of system-info cache DOT Version failed with reason: Authorization failed" .....	45
f. "Update of system-info cache DOT Version failed with reason: No response received from server" .....	45
g. "[lun] data-list poller next refresh at [2015-07-28 02:17:00] not scheduled because it occurred in the past" .....	45
h. Setting HTTP/1.0 because reverse hostname resolution (IP -> hostname) fails. To enable HTTP/1.1 ensure reverse hostname resolution succeeds.....	45
i. No best-fit collection template found (same generation and major release, minor same or less) found in [/opt/harvest/template/default]. Exiting; .....	46
j. "Poller status: status, secs=14400, api_time=1806, plugin_time=55, metrics=459594, skips=0, fails=0" .....	46
k. Additional support .....	47
<b>12. References</b> .....	<b>48</b>

# 1 Introduction to NetApp Harvest

## 1.1 About NetApp Harvest

NetApp Harvest is software that connects to a remote host, collects data, calculates and summarizes the data, and posts it to a metrics server in a graphite compatible format. It offers default collection templates for performance information from ONTAP and Data ONTAP 7-mode every 1 minute, and storage capacity information from OnCommand Unified Manager 6+ every 15 minutes. Further customization is possible in terms of the detail of data collected, the summarization that occurs, and the frequency of collection.

It is most commonly used together with [Graphite](#) as a metrics platform, and [Grafana](#) for user dashboards, as part of solution shown in the diagram below:



## 1.2 NetApp Harvest Features

NetApp Harvest uses the API interface to collect performance data from ONTAP and Data ONTAP 7-Mode, and capacity data from OnCommand Unified Manager.

Performance data collected includes:

- ONTAP
  - Node perspective: aggr, eth\_port, fcp\_port, MetroCluster fcvi adapter, storage adapter, processor, flashcache, flashpool, awa, system, vol, vol\_summary, waf1, cifs, nfsv3, nfsv4, nfsv41, iscsi, fcp, VAAI, Microsoft ODX
  - SVM perspective: cifs, nfsv3, nfsv4, nfsv41, iscsi, fcp, fcp\_lif, iscsi\_lif, lif, vol, vol\_summary, QoS, copy offload
- Data ONTAP 7-Mode
  - aggr, eth\_port, fcp\_port, storage adapter, flashcache, flashpool, cifs, nfsv3, iscsi, fcp, cifs domain, vscan

Capacity data collected for OnCommand Unified Manager supporting ONTAP includes:

- Node perspective: aggregate
  - capacity used, used percentage, growth rate, dedupe savings, compression savings, etc
- SVM perspective: SVM, volume, lun, qtree
  - capacity used, used percentage, reserved rate, fill rate, dedupe savings, compression savings, quota limit, etc

## 1.3 NetApp Harvest Theory of Operation

NetApp Harvest uses the NetApp SDK and APIs to collect performance metrics from Data ONTAP and capacity metrics from OnCommand Unified Manager.

Data is forwarded over TCP or UDP to a time series database in a Graphite compatible metrics format where it can be stored for any desired frequency and retention supporting reactive and proactive use cases.

After data collection but prior to forwarding to the time series database a plugin can be used to further manipulate, filter, or summarize data.

By default, Harvest will automatically detect the ONTAP or OCUM release and use the most appropriate collection template. It will also generate an appropriate metrics path hierarchy.

For flexibility, new counter manager objects or counters can be added, or default objects or counters can be disabled, to meet custom requirements. Any object or counter in ONTAP can be collected, usually with just a small configuration file change.

If Graphite is used as the metrics database, and Grafana for dashboards, a rich set of template driven dashboards can be imported for quick visualization of collected data.

## 1.4 Getting started with NetApp Harvest

To get started using NetApp Harvest you should perform a number of actions in sequence. A global outline of the steps are as follows:

1. [About NetApp Harvest](#)  
First understand the overall architecture. The major components are: Graphite time-series database, Grafana dashboard software, NetApp Harvest data collector.
2. [System Requirements](#)  
Decide on a system architecture for the major components. For a small installation (less than 10 nodes monitored) all services can easily run on a single VM. Install a Graphite and Grafana server, potentially using the Quick Start Guide. Install prerequisite software for NetApp Harvest and ensure network ports are open on the firewall.
3. [Installing NetApp Harvest](#) or [Upgrading NetApp Harvest](#)  
Install or upgrade the Harvest software on the host
4. [Preparing monitored systems](#)  
In order to collect from a system authentication credentials are required. It is a best practice to create dedicated accounts for Harvest with the minimum privileges needed. TLS must also be enabled on 7-Mode systems.
5. [Configuring NetApp Harvest \(basic\)](#)  
Each monitored system must be configured for collection.
6. [Integrating with Graphite](#)  
Graphite should be configured to expect metrics from Harvest and save them with a desired frequency and retention.
7. [Integrating with Grafana](#)  
Rich template driven dashboards can be loaded into Grafana to allow the system performance to be visualized.
8. [Managing NetApp Harvest workers](#)  
Start the collection workers using the manager.
9. [Troubleshooting](#)  
Should an error be encountered check this section for help to resolve

## 2 System Requirements

### 2.1 Platform

NetApp Harvest is currently written in Perl and can run on any operating system where Perl and the required packages are installed. However, to use full functionality of the software, most notably daemon support, a Linux/Unix platform is required.

Platform requirements vary based on the quantity of nodes being monitored, amount of instances monitored (i.e. # vols, luns, lifs, etc), and the frequency of monitoring.

With default collection parameters plan for the following resources per 10 monitored hosts or nodes:

- CPU: 1 vCPU
- Memory: 500MB RAM
- Disk: 1GB

If you have 40 monitored nodes you could use one poller host with 4x the above resources, or 4 poller hosts each as above. For small installations you may run NetApp Harvest on the same host as Graphite.

Typical bandwidth usage from Harvest to the monitored node is ~ 15Kbps, and from the monitored node to Harvest 90Kbps. Again, as instance count increases the bandwidth used will as well.

If you have remote nodes with many monitored instances (i.e. many vols, luns, lifs, etc) and significant network latency (20ms+) it may be beneficial to deploy a Harvest poller host local to those nodes and send metrics over the WAN to a central Graphite server. In this way the Harvest polls will not be unnecessarily delayed by network latency. To determine if having a local poller would be beneficial, test running Harvest from the remote site and compare the poll duration to the poll update frequency (use the Grafana Harvest dashboard or start `netapp-worker` with the `-v` flag). If the poll duration is much less than the frequency then it is fine to poll from the central site. But if not, placing the poller on a host near the monitored system is recommended.

## 2.2 Interoperability

The following software versions are known to work. Other software versions may work but have not explicitly been tested:

Software	Version	Notes
Graphite	0.9.13 (0.9.13-pre1) 0.9.14	Earlier versions work but lack a function (maxDataPoints) needed for high performance when working with the JSON api interface and large time spans
Grafana	3.0, 3.1.1	Dashboards are designed for Grafana 3.0. If using an older release some dashboard panels may not display as intended
NetApp Management SDK	5.3, 5.3.1, 5.4, 5.5, 5.6	
Data ONTAP 7-Mode	7.3, 8.0, 8.1, 8.2	
Clustered Data ONTAP	8.1, 8.2, 8.3, 9.0, 9.1	
OnCommand Unified Manager	6.1, 6.2, 6.3, 6.4, 7.0, 7.1	

## 2.3 Software

To use NetApp Harvest several software and ecosystem services must be available:

- An existing metrics database and dashboard, most often Graphite and Grafana
  - TIP:** See the guide “Quick Start: Installing Graphite and Grafana” available on the [NetApp Communities](#) for installation steps
- Synchronized time (NTP) across all components:
  - Monitored cluster / node / OCUM host
  - NetApp Harvest host
  - Graphite / Grafana server
- Software dependencies:

yum based system (RHEL, CentOS, etc)	<pre>sudo yum install -y epel-release unzip perl perl-JSON \ perl-libwww-perl perl-XML-Parser perl-Net-SSLeay \ perl-Time-HiRes perl-LWP-Protocol-https sudo yum install -y perl-Excel-Writer-XLSX</pre>
apt-get based system (Debian, Ubuntu, etc)	<pre>sudo apt-get -y install unzip perl libjson-perl libwww-perl \ libxml-parser-perl liblwp-protocol-https-perl \ libexcel-writer-xlsx-perl</pre>

## 2.4 Protocol and port requirements

Various network protocols and ports are required for NetApp Harvest to collect data from monitored hosts, post it to a Graphite metrics server, and for the client web browser to view the data in Graphite and Grafana. Below are details of the required network connectivity; ensure firewalls are configured to allow access below. Any item with a ‘\*’ can optionally be modified for port or protocol to match your environment requirements.

**Initiated by the NetApp Harvest host:**

Destination	Protocol	Port	Description
ONTAP Cluster management LIF, Data ONTAP 7-mode management IP, OCUM system IP	HTTPS with TLS	443*/TCP	Used to access the API and fetch performance and capacity information
Graphite Server IP	<a href="#">Graphite plaintext protocol</a>	2003*/TCP*	Used to post metrics to the Graphite metrics DB



### 3 Installing NetApp Harvest

NetApp harvest can be installed using a native Linux package (rpm and deb) or from source. Use the instructions from the appropriate subsection for your installation.

#### 3.1 RPM based installation (RHEL, CentOS, etc)

1. Install dependencies as found in the [Software](#) section.
2. Use your web browser to navigate to the NetApp Support site at <https://support.netapp.com> and download the following two software components:

1. [NetApp Management SDK for All Platforms](#)

**Note:** A form is required to download the software. Please complete and state that you want to use the SDK for Data ONTAP with Perl, and in the solution field enter “NetApp Harvest”

2. [NetApp Harvest](#)

3. Copy both software files to the /tmp directory on the Harvest poller host
4. Login the Harvest poller host and become root using sudo:

```
login as: user
password: *****
[user@host ~]# sudo -i
[root@host ~]#
```

5. Change into the /tmp directory and use yum to install the rpm replacing the '#' symbols with your actual version:

```
[root@host ~]# cd /tmp

[root@host tmp]# yum install -y netapp-harvest-#.#-#.noarch.rpm
Loaded plugins: fastestmirror, ovl
Examining netapp-harvest-1.3-1.noarch.rpm: netapp-harvest-1.3-1.noarch
Marking netapp-harvest-1.3-1.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package netapp-harvest.noarch 0:1.3-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch      Version      Repository      Size
=====
Installing:
netapp-harvest         noarch   1.3-1        /netapp-harvest-1.3-1.noarch  2.8 M

Transaction Summary
=====
Install 1 Package

Total size: 2.8 M
Installed size: 2.8 M
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
### Creating netapp-harvest group ... OK
### Creating netapp-harvest user ... OK
   Installing : netapp-harvest-1.3-1.noarch                1/1
### NetApp Manageability SDK is NOT installed.
### Download from the NetApp Support Site and install using (modify version if needed):
  unzip -j netapp-manageability-sdk-5.6.zip netapp-manageability-sdk-5.6/lib/perl/NetApp/*
-d /opt/netapp-harvest/lib
### Autostart NOT configured. Execute the following statements to configure autostart:
  sudo systemctl daemon-reload
  sudo systemctl enable netapp-harvest
```

```

### Next, configure your pollers in /opt/netapp-harvest/netapp-harvest.conf. Lastly,
start netapp-harvest by executing:
sudo systemctl start netapp-harvest
Verifying : netapp-harvest-1.3-1.noarch                                1/1

Installed:
  netapp-harvest.noarch 0:1.3-1

Complete!

[root@host tmp]#

```

6. Extract the Perl component of the SDK, replacing the 4 x '#' symbols with your actual version:

```

[root@host tmp]# unzip -j netapp-manageability-sdk-#.#.zip /
netapp-manageability-sdk-#.##/lib/perl/NetApp/* -d /opt/netapp-harvest/lib

Archive: netapp-manageability-sdk-5.6.zip
  inflating: /opt/netapp-harvest/lib/DfmErrno.pm
  inflating: /opt/netapp-harvest/lib/NaElement.pm
  inflating: /opt/netapp-harvest/lib/NaErrno.pm
  inflating: /opt/netapp-harvest/lib/NaServer.pm
continues...

```

7. Verify installation by executing the two main Harvest programs without any arguments to see their usage syntax. If an error is shown, check prerequisites or the [Troubleshooting](#) section for help.

a. NetApp Manager

```

[root@host tmp]# /opt/netapp-harvest/netapp-manager

Usage: /opt/netapp-harvest/netapp-manager {-status|-start|-stop|-restart|-import|-
export} [-poller <str>] [-group <str>] [-conf <str>] [-confdir <str>] [-logdir <str>]
[-h] [-v]

continues...

```

b. NetApp Worker:

```

[root@host tmp]# /opt/netapp-harvest/netapp-worker

Usage: /opt/netapp-harvest/netapp-worker -poller <str> [-conf <str>] [-confdir <str>]
[-logdir <str>] [-daemon] [-v] [-h]

continues...

```

8. Enable autostart:

RHEL or CentOS 6	[root@host netapp-harvest]# <b>chkconfig --add netapp-harvest</b>
RHEL or CentOS 7	[root@host netapp-harvest]# <b>systemctl daemon-reload</b> [root@host netapp-harvest]# <b>sudo systemctl enable netapp-harvest</b>

**Note:** Do not start the pollers (`service netapp-harvest start`) until you have configured Graphite with appropriate retention settings; see [Setting frequency and retention in storage-schemas.conf](#)

## 3.2 DEB based installation (Ubuntu, Debian, etc)

1. Install dependencies as found in the [Software](#) section.
2. Use your web browser to navigate to the NetApp Support site at <https://support.netapp.com> and download the following two software components:

3. [NetApp Management SDK for All Platforms](#)

**Note:** A form is required to download the software. Please complete and state that you want to use the SDK for Data ONTAP with Perl, and in the solution field enter “NetApp Harvest”

4. [NetApp Harvest](#)

3. Copy both software files to the /tmp directory on the Harvest poller host
4. Login the Harvest poller host and become root using sudo:

```
login as: user
password: *****
user@host:~# sudo -i
root@host:~#
```

5. Change into the /tmp directory and use apt-get to install the deb replacing the '#' symbols with your actual version:

```
root@host:~# cd /tmp

root@host:tmp# dpkg -i netapp-harvest_#.#_all.deb
dpkg -i netapp-harvest_1.3_all.debSelecting previously unselected package netapp-harvest.
(Reading database ... 10560 files and directories currently installed.)
Preparing to unpack netapp-harvest_1.3_all.deb ...
### Creating netapp-harvest group ... OK
### Creating netapp-harvest user ... OK
Unpacking netapp-harvest (1.3) ...
Setting up netapp-harvest (1.3) ...
### NetApp Manageability SDK is NOT installed.
### Download from the NetApp Support Site and install using (modify version if needed):
  unzip -j netapp-manageability-sdk-5.6.zip netapp-manageability-sdk-5.6/lib/perl/NetApp/*
-d /opt/netapp-harvest/lib
### Autostart NOT configured. Execute the following statements to configure autostart:
  sudo systemctl daemon-reload
  sudo systemctl enable netapp-harvest
### Next, configure your pollers in /opt/netapp-harvest/netapp-harvest.conf. Lastly,
start netapp-harvest by executing:
  sudo systemctl start netapp-harvest
Processing triggers for systemd (229-4ubuntu10) ...

root@host:tmp#
```

6. Extract the Perl component of the SDK, replacing the 4 x '#' symbols with your actual version:

```
root@host:tmp# unzip -j netapp-manageability-sdk-#.#.zip /
netapp-manageability-sdk-#.#/lib/perl/NetApp/* -d /opt/netapp-harvest/lib

Archive: netapp-manageability-sdk-5.6.zip
  inflating: /opt/netapp-harvest/lib/DfmErrno.pm
  inflating: /opt/netapp-harvest/lib/NaElement.pm
  inflating: /opt/netapp-harvest/lib/NaErrno.pm
  inflating: /opt/netapp-harvest/lib/NaServer.pm
continues...
```

- Verify installation by executing the two main Harvest programs without any arguments to see their usage syntax. If an error is shown, check prerequisites or the [Troubleshooting](#) section for help.

- NetApp Manager

```
root@host:tmp# /opt/netapp-harvest/netapp-manager

Usage: /opt/netapp-harvest/netapp-manager {-status|-start|-stop|-restart|-import|-
export} [-poller <str>] [-group <str>] [-conf <str>] [-confdir <str>] [-logdir <str>]
[-h] [-v]

continues...
```

- NetApp Worker:

```
root@host:tmp# /opt/netapp-harvest/netapp-worker

Usage: /opt/netapp-harvest/netapp-worker -poller <str> [-conf <str>] [-confdir <str>]
[-logdir <str>] [-daemon] [-v] [-h]

continues...
```

- Enable autostart:

Ubuntu 14.04	root@host:tmp# <code>update-rc.d netapp-harvest defaults</code>
Ubuntu 16.04 Debian 7, Debian 8	[root@host netapp-harvest]# <code>systemctl daemon-reload</code> [root@host netapp-harvest]# <code>sudo systemctl enable netapp-harvest</code>

**Note:** Do not start the pollers (`service netapp-harvest start`) until you have configured Graphite with appropriate retention settings; see [Setting frequency and retention in storage-schemas.conf](#)

### 3.3 Source based installation

- Install dependencies as found in the [Software](#) section.
- Use your web browser to navigate to the NetApp Support site at <https://support.netapp.com> and download the following two software components:
  - [NetApp Management SDK for All Platforms](#)

**Note:** A form is required to download the software. Please complete and state that you want to use the SDK for Data ONTAP with Perl, and in the solution field enter “NetApp Harvest”
  - [NetApp Harvest](#)
- Copy both software files to the `/tmp` directory on the Harvest poller host
- Login the Harvest poller host and become root using `sudo`:

```
login as: user
password: *****
[user@host ~]# sudo -i
[root@host ~]#
```

- Change into the `/tmp` directory, extract the NetApp Harvest software from the `.zip` file (replacing the `#` symbols with your actual version), and move the contents to the `/opt` directory:

```
[root@host ~]# cd /tmp

[root@host tmp]# unzip netapp-harvest-#.zip

Archive:  netapp-harvest-1.3.zip
  creating:  netapp-harvest/
  inflating:  netapp-harvest/CHANGES.txt
  inflating:  netapp-harvest/netapp-harvest.conf.example
continues...

[root@host tmp]# mv /tmp/netapp-harvest /opt/netapp-harvest
```

- Extract the Perl component of the SDK, replacing the 4 x `#` symbols with your actual version:

```
[root@host tmp]# unzip -j netapp-manageability-sdk-#.zip /
netapp-manageability-sdk-#.lib/perl/NetApp/* -d /opt/netapp-harvest/lib

Archive:  netapp-manageability-sdk-5.6.zip
  inflating:  /opt/netapp-harvest/lib/DfmErrno.pm
  inflating:  /opt/netapp-harvest/lib/NaElement.pm
continues...
```

- Verify installation by executing the two main Harvest programs without any arguments to see their usage syntax. If an error is shown, check prerequisites or the [Troubleshooting](#) section for help.

- NetApp Manager

```
[root@host netapp-harvest]# /opt/netapp-harvest/netapp-manager

Usage: /opt/netapp-harvest/netapp-manager {-status|-start|-stop|-restart|-import|-
export} [-poller <str>] [-group <str>] [-conf <str>] [-confdir <str>] [-logdir <str>]
[-h] [-v]

continues...
```

- NetApp Worker:

```
[root@host netapp-harvest]# /opt/netapp-harvest/netapp-worker

Usage: /opt/netapp-harvest/netapp-worker -poller <str> [-conf <str>] [-confdir <str>]
[-logdir <str>] [-daemon] [-v] [-h]

continues...
```

- Copy the default startup script to the `/etc/init.d` directory.

```
[root@host ~]# cp /opt/netapp-harvest/util/netapp-harvest /etc/init.d
```

- Add the service to the Linux system configuration and enable it to autostart:

Ubuntu 14.04	<code>root@host:tmp# update-rc.d netapp-harvest defaults</code>
RHEL or CentOS 6	<code>[root@host netapp-harvest]# chkconfig --add netapp-harvest</code>
RHEL or CentOS 7 Ubuntu 16.04 Debian 7, Debian 8	<code>[root@host netapp-harvest]# systemctl daemon-reload</code> <code>[root@host netapp-harvest]# sudo systemctl enable netapp-harvest</code>

**Note:** Do not start the pollers (`service netapp-harvest start`) until you have configured Graphite with appropriate retention settings; see [Setting frequency and retention in storage-schemas.conf](#)

## 4 Upgrading NetApp Harvest

NetApp harvest can be upgraded using a native Linux package (rpm and deb) or from source. Use the instructions from the appropriate subsection for your installation. All customized configuration files will be retained.

### 4.1 RPM based upgrade (RHEL, CentOS, etc)

1. Upgrade dependencies as found in the [Software](#) section.
2. Use your web browser to navigate to the NetApp Support site at <https://support.netapp.com> and download the following software component:

- [NetApp Harvest](#)

3. Copy the software file to the /tmp directory on the Harvest poller host
4. Login the Harvest poller host and become root using sudo:

```
login as: user
password: *****
[user@host ~]# sudo -i
[root@host ~]#
```

5. Change into the /tmp directory and use dpkg to install the deb package:

```
[root@host ~]# cd /tmp

[root@host ~]# yum upgrade -y netapp-harvest-1.3-1.noarch.rpm
Loaded plugins: fastestmirror, ovl
Setting up Upgrade Process
Examining netapp-harvest-1.3X2-1.noarch.rpm: netapp-harvest-1.3X2-1.noarch
Marking netapp-harvest-1.3X2-1.noarch.rpm as an update to netapp-harvest-1.3X1-1.noarch
Resolving Dependencies
--> Running transaction check
--> Package netapp-harvest.noarch 0:1.3X1-1 will be updated
--> Package netapp-harvest.noarch 0:1.3X2-1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch      Version      Repository      Size
=====
Updating:
netapp-harvest         noarch    1.3X2-1      /netapp-harvest-1.3X2-1.noarch  2.8 M

Transaction Summary
=====
Upgrade                1 Package(s)

Total size: 2.8 M
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
### Stopping NetApp Harvest
  Updating      : netapp-harvest-1.3X2-1.noarch                1/2
### NetApp Manageability SDK is NOT installed.
### Download from the NetApp Support Site and install using (modify version if needed):
  unzip -j netapp-manageability-sdk-5.5.zip netapp-manageability-sdk-5.5/lib/perl/NetApp/*
-d /opt/netapp-harvest/lib
### Starting NetApp Harvest after upgrade
  Cleanup      : netapp-harvest-1.3X1-1.noarch                2/2
  Verifying    : netapp-harvest-1.3X2-1.noarch                1/2
  Verifying    : netapp-harvest-1.3X1-1.noarch                2/2

Updated:
  netapp-harvest.noarch 0:1.3X2-1

Complete!
```

## 4.2 DEB based upgrade (Ubuntu, Debian, etc)

1. Upgrade dependencies as found in the [Software](#) section.
2. Use your web browser to navigate to the NetApp Support site at <https://support.netapp.com> and download the following software component:
  - [NetApp Harvest](#)
3. Copy the software file to the /tmp directory on the Harvest poller host
4. Login the Harvest poller host and become root using sudo:

```
login as: user
password: *****
user@host:~# sudo -i
root@host:~#
```

5. Change into the /tmp directory and use dpkg to install the deb package:

```
root@host:~# cd /tmp

root@host:tmp# dpkg -i netapp-harvest_1.3_all.deb
dpkg -i netapp-harvest_1.3X2_all.deb
(Reading database ... 12902 files and directories currently installed.)
Preparing to unpack netapp-harvest_1.3X2_all.deb ...
### Stopping NetApp Harvest
Unpacking netapp-harvest (1.3X2) over (1.3X1) ...
Setting up netapp-harvest (1.3X2) ...
### Starting NetApp Harvest after upgrade
Processing triggers for ureadahead (0.100.0-16) ...
root@host:/tmp #
```

## 4.3 Source based upgrade

1. Upgrade dependencies as found in the [Software](#) section.
2. Use your web browser to navigate to the NetApp Support site at <https://support.netapp.com> and download the following software component:
  - [NetApp Harvest](#)
3. Copy the software file to the /tmp directory on the Harvest poller host
4. Login the Harvest poller host and become root using sudo:

```
login as: user
password: *****
[user@host ~]# sudo -i
[root@host ~]#
```

5. Change into the /tmp directory, extract the NetApp Harvest software from the .zip file, and move the contents to the /opt/netapp-harvest-new directory:

```
[root@host ~]# cd /tmp

[root@host tmp]# unzip netapp-harvest-1.3.zip

Archive:  netapp-harvest-1.3.zip
  creating: netapp-harvest/
  inflating: netapp-harvest/CHANGES.txt
  inflating: netapp-harvest/netapp-harvest.conf.example
continues...

[root@host tmp]# mv /tmp/netapp-harvest /opt/netapp-harvest-new
```

- Copy the configuration files from your existing installation into the new installation

**Note:** The `\cp` ensures you don't use an alias for `cp` (which normally contains a `-i` requiring you to interactively accept overwrites)

```
[root@host tmp]# \cp /opt/netapp-harvest/template/* /opt/netapp-harvest-new/template
(ignore warning about omitting subdirectories)

[root@host tmp]# \cp /opt/netapp-harvest/cert/* /opt/netapp-harvest-new/cert

[root@host tmp]# \cp /opt/netapp-harvest/*conf /opt/netapp-harvest-new/
```

- Stop the existing NetApp Harvest pollers

```
[root@host tmp]# service netapp-harvest stop

Verify using ps that no pollers are running. If any are running kill them by PID manually.
[root@host tmp]# ps -ef | grep netapp
```

- Rename directories; update # with your current release version

```
[root@host opt]# cd /opt
[root@host opt]# mv netapp-harvest netapp-harvest-#. #
[root@host opt]# mv netapp-harvest-new netapp-harvest
```

- Start all pollers

```
[root@host opt]# service netapp-harvest start
```

- Update the Grafana dashboards to the latest versions, see [Integrating with Grafana](#) for instructions.

- Copy the default startup script to the `/etc/init.d` directory.

```
[root@host ~]# cp /opt/netapp-harvest/util/netapp-harvest /etc/init.d
```

- Add the service to the Linux system configuration and enable it to autostart:

Ubuntu 14.04	<code>root@host:tmp# update-rc.d netapp-harvest defaults</code>
RHEL or CentOS 6	<code>[root@host netapp-harvest]# chkconfig --add netapp-harvest</code>
RHEL or CentOS 7 Ubuntu 16.04 Debian 7, Debian 8	<code>[root@host netapp-harvest]# systemctl daemon-reload</code> <code>[root@host netapp-harvest]# sudo systemctl enable netapp-harvest</code>

**Note:** Do not start the pollers (`service netapp-harvest start`) until you have configured Graphite with appropriate retention settings; see [Setting frequency and retention in storage-schemas.conf](#)



## 4. Preparing Monitored Systems

Several steps should be taken on each monitored system to prepare it for collection by Harvest.

### a. Enabling HTTPS and TLS (Data ONTAP 7-mode only)

Harvest communicates with monitored systems exclusively via HTTPS using TLSv1 which is not enabled by default in Data ONTAP 7-mode. In contrast, it is always enabled (and cannot be disabled) on ONTAP and OnCommand Unified Manager. Execute the following steps on each Data ONTAP 7-mode system:

1. Verify SSL is configured:

```
netapp> secureadmin status ssl
sdt-7dot1b> secureadmin status
ssh2      - active
ssh1      - inactive
ssl       - active
```

If ssl is 'active' continue. If not, setup SSL and be sure to choose a Key length (bits) of 2048:

```
netapp> secureadmin setup ssl
SSL Setup has already been done before. Do you want to proceed? [no] yes
Country Name (2 letter code) [US]: NL
State or Province Name (full name) [California]: Noord-Holland
Locality Name (city, town, etc.) [Santa Clara]: Schiphol
Organization Name (company) [Your Company]: NetApp
Organization Unit Name (division): SalesEngineering
Common Name (fully qualified domain name) [sdt-7dot1a.nltestlab.hq.netapp.com]:
Administrator email: noreply@netapp.com
Days until expires [5475] :5475
Key length (bits) [512] :2048
```

2. Enable management via SSL and enable TLS:

```
netapp> options httpd.admin.ssl.enable on
netapp> options tls.enable on
```

### b. Creating user accounts

NetApp Harvest requires login credentials to access monitored hosts. Although a generic admin account can be used it is best practice to create dedicated monitoring accounts with least privilege access.

In the examples the user, groups, roles, etc, use a naming standard including the 'netapp-harvest' name. These can be modified as needed to match your environment standards.

#### i. Data ONTAP 7-mode

Data ONTAP 7-mode supports only username / password based authentication with NetApp Harvest. Login as a user with full administrative privileges and execute the following steps.

**Note:** Commands below should be entered on a single line but due to documentation formatting might include newlines therefore it is recommended to copy/paste into a text editor, remove any added newlines, and then copy/paste at the NetApp CLI.

1. Create the role with required capabilities:

```
useradmin role add netapp-harvest-role -c "Role for performance monitoring by NetApp Harvest" -a
login-http-admin,api-system-get-version,api-system-get-info,api-perf-object-*,api-ems-
autosupport-log
```

2. Create a group for this role:

```
useradmin group add netapp-harvest-group -c "Group for performance monitoring by NetApp Harvest"
-r netapp-harvest-role
```

### 3. Create a user that utilizes this role and enter the password when prompted:

```
useradmin user add netapp-harvest -c "User account for performance monitoring by NetApp Harvest"
-n "NetApp Harvest" -g netapp-harvest-group
```

The user is now created and can be configured for use by NetApp Harvest.

## ii. ONTAP (Clustered Data ONTAP)

ONTAP supports username / password and SSL certificate based authentication with NetApp Harvest. Login as a user with full administrative privileges and execute the following steps.

**Note:** Commands below should be entered on a single line but due to documentation formatting might include newlines therefore it is recommended to copy/paste into a text editor, remove any added newlines, and then copy/paste at the NetApp CLI.

### 1. Create a role with required capabilities:

```
security login role create -role netapp-harvest-role -access readonly -cmddirname "version"
security login role create -role netapp-harvest-role -access readonly -cmddirname "cluster
identity show"
security login role create -role netapp-harvest-role -access readonly -cmddirname "cluster show"
security login role create -role netapp-harvest-role -access readonly -cmddirname "system node
show"
security login role create -role netapp-harvest-role -access readonly -cmddirname "statistics"
security login role create -role netapp-harvest-role -access readonly -cmddirname "lun show"
security login role create -role netapp-harvest-role -access readonly -cmddirname "network
interface show"
security login role create -role netapp-harvest-role -access readonly -cmddirname "qos workload
show"
```

### 2. If you wish to use SSL certificate based authentication complete the following sub steps, otherwise continue to step 3).

- a. Login the NetApp Harvest Linux host, become root, change into the /opt/netapp-harvest/cert subdirectory, and generate a client certificate and private key using information from your environment:

**Note:** The 'Common Name' must match the username you create on the cDOT cluster later. Also, in this example the certificate will be valid for 10 years; adjust the days count according to your security requirements.

```
user@host ~# sudo -i

root@host ~# cd /opt/netapp-harvest/cert

root@host cert# openssl req -x509 -nodes -days 3650 -newkey rsa:1024 -keyout
netapp-harvest.key -out netapp-harvest.pem
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to '10yr.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:NL
State or Province Name (full name) [Some-State]:Noord-Holland
Locality Name (eg, city) [:]:Schiphol-Rijk
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NetApp BV
Organizational Unit Name (eg, section) []:PS
Common Name (e.g. server FQDN or YOUR name) []:netapp-harvest
Email Address []:netapp-harvest@netapp.com
```

A file named `netapp-harvest.pem` with the public certificate and a `netapp-harvest.key` containing your private key is created.

- b. Login the ONTAP clustershell as a user with full administrative privileges and add the public certificate. The entire contents of the pem file should be copied and pasted when requested. Also be sure to replace the argument of `-vserver` with the name of your cluster:

```
cluster::> security certificate install -type client-ca -vserver cluster
```

```
Please enter Certificate: Press <Enter> when done
-----BEGIN CERTIFICATE-----
MIICChDCCAe2gAwIBAgIJAKgurBmDXc3uMA0GCSqGSIb3DQEBBQUAMFsxCzAJBgNV
BAYTAk5MMRUwEwYDVoQHDAAeZWZhdWx0IENpdHkxHDAaBgNVBAoME0RlZmFlbHQQg
Q29tcGFueSBMdGQxZmFzAVBgnVBAMMDm5ldGFwcC1oYXJ2ZXN0MB4XDTE1MDYyNjEw
MTk1N1oXDTE1MDYyMzEwMTk1N1owWzELMAkGA1UEBhMCTkwxFtATBgNVBACMDERl
ZmFlbHQQgQ210eTEcMBoGA1UECgwTRGVmYXVsdCBDb21wYW55IEx0ZDEXMBUGA1UE
AwV0bWV0YXZlLWVhcnZlc3QwZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMyq
Qq6qXRW7czWRNHmfm1Zjpr0FV/VmOv0Brt9Ij7+tHYb+CcIKVYj/gv0RM8DGJ5L
X7VrdnnpINAu6tghBS6YOG2Nrlh9CRunBR91Hm2/DPKA7C0cNjg6EHuJkYLOVF21
nmRpdAXDURBfw89v1YrZz7ucLBqGX8SRqi0y0OvAgMBAAGjUDBOMB0GA1UdDgQw
BBTOMM2pC8HH0aK9ZRgW5OxOqcV7RDafbGnvHSMEGDAWgBTOMM2pC8HH0aK9ZRgW
5OxOqcV7RDAMBgnVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA4GBAFrg5HjXtZ8q
YkRcnCyekvdtFT1a18FyWjDUkRtldySyRgsdtwcf6BoYiVvEmjPVX2QR8n6u8G/R
Ii+6Mwt+OdWPTvzZX6k92ni3yDr0Ffghjj9V5+UZEK8aGHPnD4kpt/sAnJf3gbzO
WswIMiWH6mYaYLnkGDaze9UuXZcEuw4E
-----END CERTIFICATE-----
```

You should keep a copy of the CA-signed digital certificate for future reference.

- c. Enable SSL client authentication. Be sure to replace the argument of `-vserver` with the name of your cluster:

```
cluster::> security ssl modify -client-enabled true -vserver clustername
```

### 3. Create the user account on the ONTAP cluster.

#### *For Data ONTAP 8.2 or earlier:*

- Username/Password based (you will be prompted for the password):

```
security login create -username netapp-harvest -application ontapi -role netapp-
harvest-role -authmethod password
```

- SSL Certificate based:

```
security login create -username netapp-harvest -application ontapi -role netapp-
harvest-role -authmethod cert
```

#### *For ONTAP 8.3 or later:*

- Username/Password based (you will be prompted for the password):

```
security login create -user-or-group-name netapp-harvest -application ontapi -role
netapp-harvest-role -authmethod password
```

- SSL Certificate based:

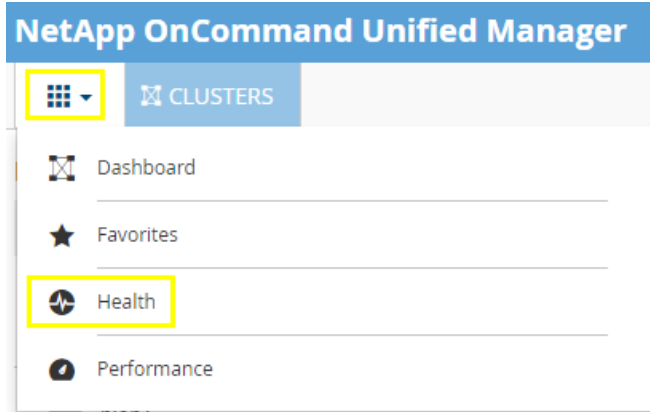
```
security login create -user-or-group-name netapp-harvest -application ontapi -role
netapp-harvest-role -authmethod cert
```

**Note:** The user created has the ability to login via the API only; login access via other application/login methods is expected to fail.

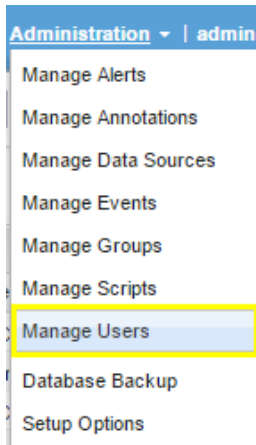
### iii. OnCommand Unified Manager (OCUM)

OCUM supports username / password based authentication with NetApp Harvest. Login the OCUM web interface as a user with full administrative privileges and execute the following steps.

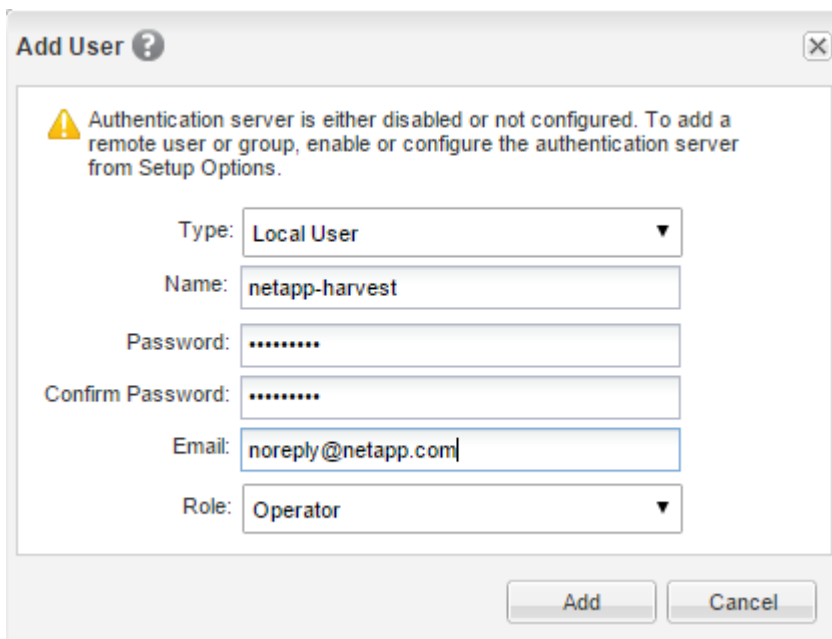
1. Access the 'Health' view:



2. Access the 'Administration' menu, and then 'Manage Users':



3. Enter parameters for the new user. A role of 'Operator' provides sufficient privilege for collection of storage capacity information as needed by Harvest:



## 5. Configuring NetApp Harvest (basic)

### a. Populate the configuration file

1. Login the collector host and if not logged in as the root superuser, become root using sudo:

```
[user@host ~]# sudo -i
```

2. Change directory to the installation directory

```
[root@host ~]# cd /opt/netapp-harvest
```

3. Copy the default example configuration file to your configuration file:

```
[root@host netapp-harvest]# cp netapp-harvest.conf.example netapp-harvest.conf
```

4. Use a text editor (nano, vi) to edit the configuration file. Values in all CAPITAL LETTERS must be modified to match your environment. Details related to Grafana parameters will be modified later in the [Integrating with Grafana](#) instructions so skip them now.

```
[root@host netapp-harvest]# nano netapp-harvest.conf
```

- a. Locate the [default] section of the file and modify the graphite\_server value to match your environment:

```
graphite_server = INSERT_IP_OR_HOSTNAME_OF_GRAPHITE_SERVER_HERE

example new value:

graphite_server = 10.64.31.76
```

- b. Continue in the [default] section of the file and set authorization credentials

For password authentication modify the username and password for your environment:

```
username = INSERT_USERNAME_HERE
password = INSERT_PASSWORD_HERE

example new values:

username = netapp-harvest
password = sEcReTr3aDoNlYpW
```

Or, to use SSL certificates ensure the .pem and .key files are in the /opt/netapp-harvest/cert directory and uncomment and complete the filenames for your environment:

```
#auth_type = ssl_cert
#ssl_cert = INSERT_PEM_FILE_NAME_HERE
#ssl_key = INSERT_KEY_FILE_NAME_HERE

example new values:

auth_type = ssl_cert
ssl_cert = netapp-harvest.pem
ssl_key = netapp-harvest.key
```

- c. Find the example section for a Data ONTAP system, copy and remove the '#' symbols, and modify the values to match your environment:

**Note:** No whitespace characters are allowed in the section header name or group fields

```
# [INSERT_CLUSTER_OR_CONTROLLER_NAME_HERE_EXACTLY_AS_SHOWN_FROM_CLI_PROMPT]
# hostname          = INSERT_IP_ADDRESS_OR_HOSTNAME_OF_CONTROLLER_OR_CLUSTER_LIF_HERE
# group             = INSERT_GROUP_IDENTIFIER_HERE

example new values:

[amsstor001]
hostname           = 10.64.31.100
group              = ams4
```

- d. Find the example section for an OCUM system, copy and remove the '#' symbols, and modify the values to match your environment:

```
# [INSERT_OCUM_SERVER_NAME_HERE]
# hostname          = INSERT_IP_ADDRESS_OR_HOSTNAME_OF_OCUM_SERVER
# group             = INSERT_GROUP_IDENTIFIER_HERE
# host_type         = OCUM
# data_update_freq = 900
# normalized_xfer   = gb_per_sec

example new values:

[nyclinx001]
hostname           = 10.24.31.243
group              = nycl
host_type          = OCUM
data_update_freq   = 900
normalized_xfer    = gb_per_sec
```

**Note:** If you only use OCUM capacity collection and do not want to collect performance information from the clusters you must still have an entry for each monitored cluster. Create an entry for each cluster as shown above (omitting login details) and set the parameter `host_enabled = 0`. These settings will allow NetApp Harvest to still include the group in the Graphite metrics hierarchy but will not collect performance data from the clusters.

5. Repeat step 4c and 4d as required for each cluster, 7-mode controller, or OCUM system. If a global username/password is not desired it can also be specified in each poller section.
6. Save the file and exit to the command prompt.

**Note:** Do not start the pollers (`./netapp-manager -start`) until you have configured Graphite with appropriate retention settings; see [Setting frequency and retention in storage-schemas.conf](#).

## 6. Integrating with Graphite

### a. Setting frequency and retention in storage-schemas.conf

Graphite, and specifically the Carbon service, receives metrics from NetApp Harvest and stores them in metrics db files in the whisper db format. The first time a metric is received the db file is created with frequency and retention settings found in the Carbon `storage-schemas.conf` configuration file.

This file includes one or more rules, each with a regular expression pattern that is tested against the incoming metric. If it matches, Carbon will create a db file with the frequency and retention defined in that rule. Because of this behavior be sure to put any 'catch all' entry (such as the `[default_1min_for_1day]`) at the bottom of the file.

Also, if you change the frequency and retention settings later it will only impact new db files created; existing metrics will continue to use the settings that were effective at the time the db file was created.

The following steps configure frequency and retention settings which are appropriate for NetApp Harvest sourced metrics:

1. Login the graphite host and if not logged in as the root superuser, become root using sudo:

```
[user@host ~]# sudo -i
```

2. Use a text editor (nano, vi) to edit the `storage-schemas.conf` file. Depending on your installation the location can vary, but is typically one of:

```
[root@host ~]# nano /etc/carbon/storage-schemas.conf  
  
or  
  
[root@host ~]# nano /opt/graphite/conf/storage-schemas.conf
```

- a. Add the following entries to `storage-schemas.conf` **above the default section `[default_1min_for_1day]`** to set metrics retention for the Harvest metrics hierarchy:

```
[netapp_perf]  
pattern = ^netapp(\.poller)?\.perf7?\  
retentions = 1m:35d,5m:100d,15m:395d,1h:5y  
  
[netapp_capacity]  
pattern = ^netapp(\.poller)?\.capacity\  
retentions = 15m:100d,1d:5y
```

The above entries result in metrics retention:

- i. Performance: 60 sec samples for 35 days, 5 min averages for 100 days, 15 min averages for 395 days, and 1 hour averages for 5 years.
  - ii. Capacity: 15 min samples for 100 days, 1 day averages for 5 years
  - iii. Retentions can be adjusted to match your environment, but do not adjust the pattern
7. Save the file and exit to the command prompt.
  8. The running Carbon service will automatically discover changes made to this file so no restart of the service is needed.

### b. Increase the maximum creates per minute in carbon.conf

Graphite, and specifically the Carbon service, receives metrics from NetApp Harvest and stores them in metrics db files. By default the Carbon service will throttle creation of new metric db files at 50 per minute to reduce the chance metric db updates will be starved or dropped. This value is overly conservative and can result in it taking hours for metrics to be visible upon adding a new monitored host. A value of 600 per minute is typically safe and provides visibility of a new metrics faster.

The following steps increase the maximum creates per minute:

1. Login the graphite host and if not logged in as the root superuser, become root using sudo:

```
[user@host ~]# sudo -i
```

2. Use a text editor (nano, vi) to edit the storage-schemas.conf file. Depending on your installation the location can vary, but is typically one of:

```
[root@host ~]# nano /etc/carbon/carbon.conf  
  
or  
  
[root@host ~]# nano /opt/graphite/conf/carbon.conf
```

- a. Locate the line with MAX\_CREATES\_PER\_MINUTE and modify it from 50 to 600:

```
MAX_CREATES_PER_MINUTE = 50  
  
change to  
  
MAX_CREATES_PER_MINUTE = 600
```

3. Save the file and exit to the command prompt.
4. Restart the Carbon cache process to make the changes active:

```
[root@host ~]# service carbon-cache stop  
  
Verify it stopped by running the following command repeatedly until it shows to be stopped. NOTE: It can take a few minutes if it has many cached metrics to flush to disk  
  
[root@host ~]# service carbon-cache status  
  
start again  
  
[root@host ~]# service carbon-cache start
```

### c. Use a blacklist to prevent visibility of temporary volumes [optional]

Harvest will discover all instances of objects such as lifs, volumes, and luns on the cluster. Some of these instances may be temporary, such as those used by backup jobs, and add undesired clutter and storage consumption. Graphite provides the ability to blacklist (block) incoming metrics that match a regular expression string and this feature can be used to block metrics for these temporary instances.

The following steps can be used to block metrics on temporary volumes that are often created by NetApp and NetApp partner software integrations:

1. Login the graphite host and if not logged in as the root superuser, become root using sudo:

```
[user@host ~]# sudo -i
```

2. Use a text editor (nano, vi) to edit the blacklist.conf file. Depending on your installation the location can vary, but is typically one of:

```
[root@host ~]# nano /etc/carbon/blacklist.conf  
  
or  
  
[root@host ~]# nano /opt/graphite/conf/blacklist.conf
```

3. Add the following contents to this file, save, and exit to the command prompt.

```
#  
# Exclusion for SnapProtect/CommVault Intellisnap  
# Clone volumes gets a "_CVclone" suffix  
#  
^netapp\.(capacity|perf7?)\..\.\vol\..\._CVclone\..+  
  
#  
# Exclusion for SnapCenter
```



```

# Clone volumes gets a "DDMMYYhhmss" suffix
#
^netapp\.(capacity|perf)\..\vol\..\+(0[1-9]|[12][0-9]|3[01])(0[1-9]|1[012])\d\d[0-9]{6}\..+

#
# Exclusion for SnapCreator
# This is from creating a clone manually in SnapCreator
# Clone volume gets a "cl_" prefix and a "_YYYYMMDDhhmss" suffix
#
^netapp\.(capacity|perf?)\..\vol\.cl_._+(19|20)\d\d(0[1-9]|1[012])(0[1-9]|[12][0-9]|3[01])[0-9]{6}\..+

#
# Exclusion for SnapDrive/SnapManager
# Clone volumes gets a "sdw_cl_" prefix
#
^netapp\.(capacity|perf?)\..\vol\.sdw_cl_._+\..+

#
# Exclusion for Metadata volumes that may also clutter menus.
#
# CRS volumes in SVM-DR or MetroCluster get a "MDV_CRS_" prefix
# Audit volumes get a "MDV_aud_" prefix
#
^netapp\.(perf)\..\vol\.MDV_CRS_._+\..+
^netapp\.(perf)\..\vol\.MDV_aud_._+\..+

```

#### d. Purging inactive metrics from Graphite [optional]

Graphite does not have an API to purge inactive metrics. As a consequence, if instances on the cluster (luns, volumes, lifs, etc) are deleted the associated metrics will not leading to stale metrics that clutter the UI forever. As a housekeeping practice many Graphite administrators configure a cronjob that purges inactive metrics files and parent directories if they become empty.

The following steps can be used to setup the purging of inactive metrics files:

1. Login the graphite host and if not logged in as the root superuser, become root using sudo:

```
[user@host ~]# sudo -i
```

2. Add a crontab entry. The following syntax will purge metrics with 120 days of inactivity, and any empty directories, every Sunday at 00:30:

```
[root@host ~]# crontab -e
```

***If using an installation installed from Source:***

```
30 0 * * 7 find /opt/graphite/storage/whisper -type f -mtime +120 -name \*.wsp -delete;
find /opt/graphite/storage/whisper -depth -type d -empty -delete
```

***or if using Ubuntu package:***

```
30 0 * * 7 find /var/lib/graphite/whisper -type f -mtime +120 -name \*.wsp -delete; find
/var/lib/graphite/whisper -depth -type d -empty -delete
```

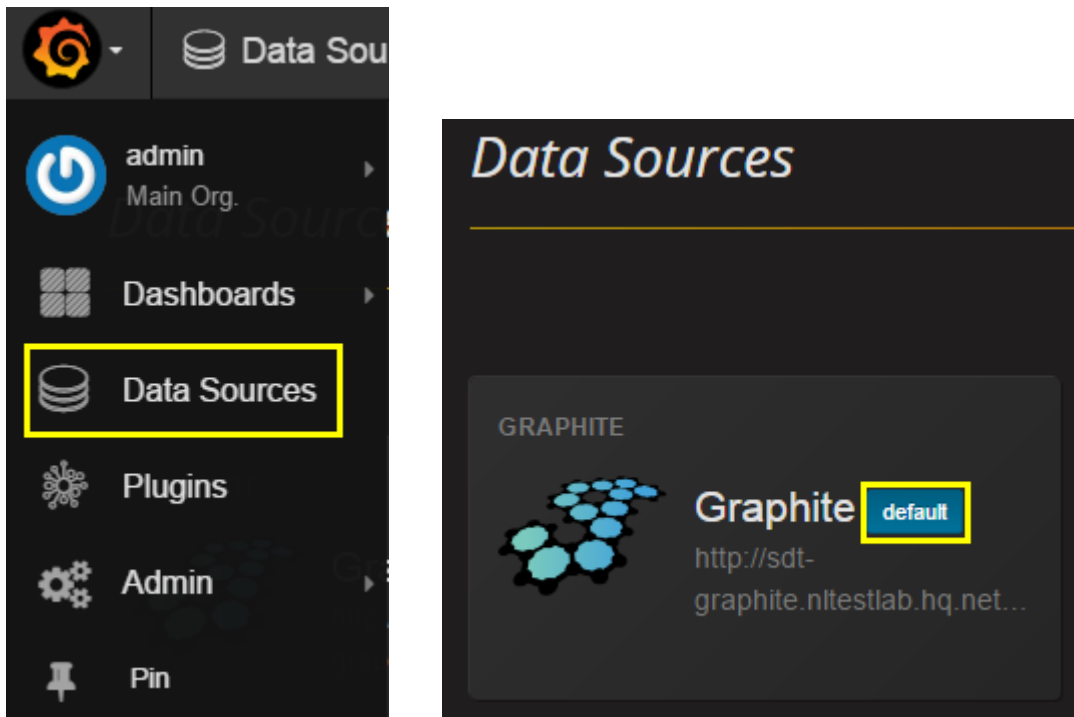
## 7. Integrating with Grafana

NetApp Harvest includes many rich template driven Grafana dashboards to quickly and easily visualize NetApp performance and capacity. It also includes the ability to bulk import and export dashboards to facilitate sharing or making a backup of your custom dashboards.

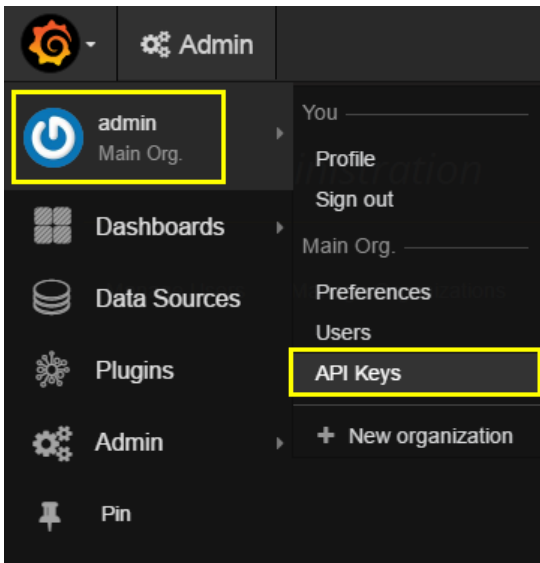
### a. Creating a Grafana API key and configuration in Harvest

Integration with Grafana requires creating an API key in Grafana and adding this into the Harvest configuration file. The following steps should be followed:

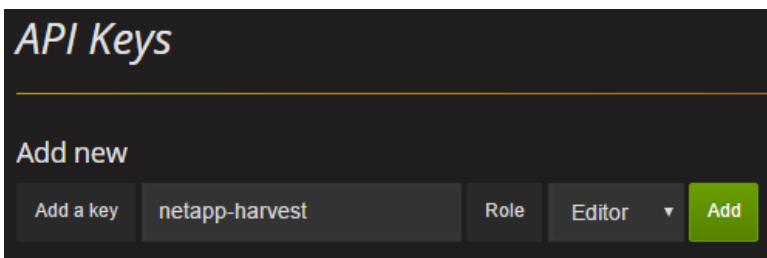
1. Login as an administrator into the Grafana Web Interface and click the Grafana logo, Data Sources, and verify a Graphite data source has been added and is set to the default:



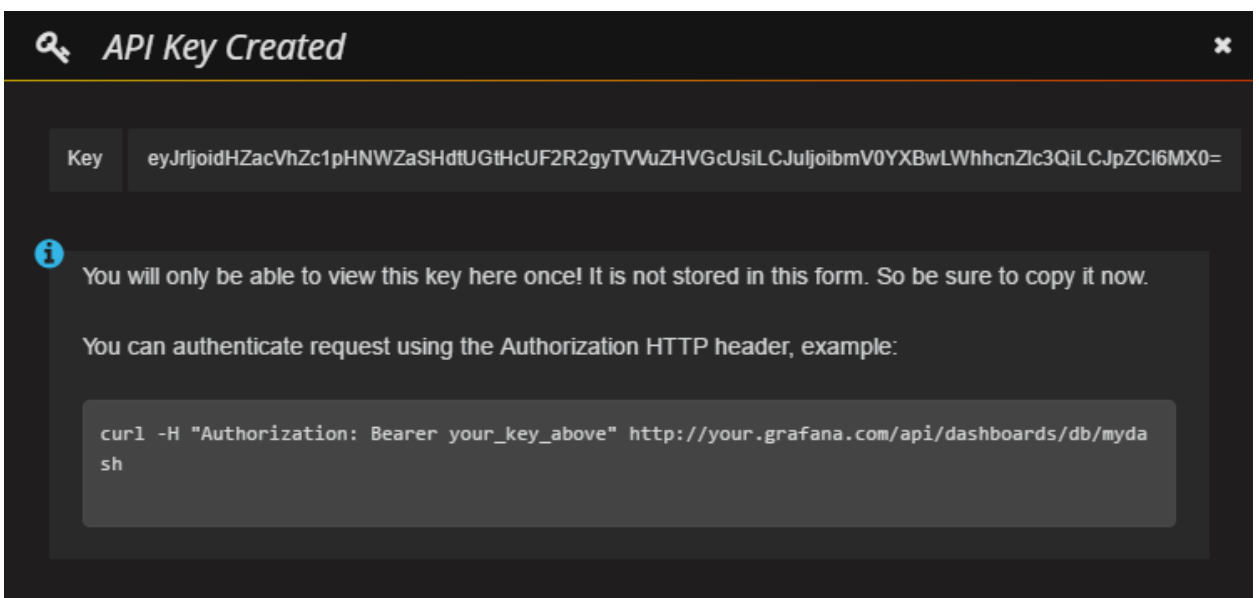
2. Add a new API key by clicking the main org and then API keys:



3. Enter a description, choose the Editor role, and click 'Add':



4. The API key will be displayed. Copy the entire string (including an '=' at the end) to be able to add to the conf file in the next step:



- Use a text editor (nano, vi) to edit the netapp-harvest configuration file and add the Grafana server details.

```
[root@host netapp-harvest]# nano /opt/netapp-harvest/netapp-harvest.conf
```

- Locate the [default] section of the file and modify the grafana\_api\_key value to match the key provided in the previous step, and modify the Grafana URL to match the URL used in your web browser:

```
[global]
grafana_api_key = INSERT_LONG_KEY_HERE
grafana_url = INSERT_URL_OF_GRAFANA_WEB_INTERFACE_HERE

example new values:

[global]
grafana_api_key = yJrIjoiMWhhUHIiLCJuIjoibmV0YXBwLWwhcnZlc3QiLCJpZCI6MX0=
grafana_url = http://localhost:3000
```

- Save the file and exit to the command prompt.

## b. Importing dashboards

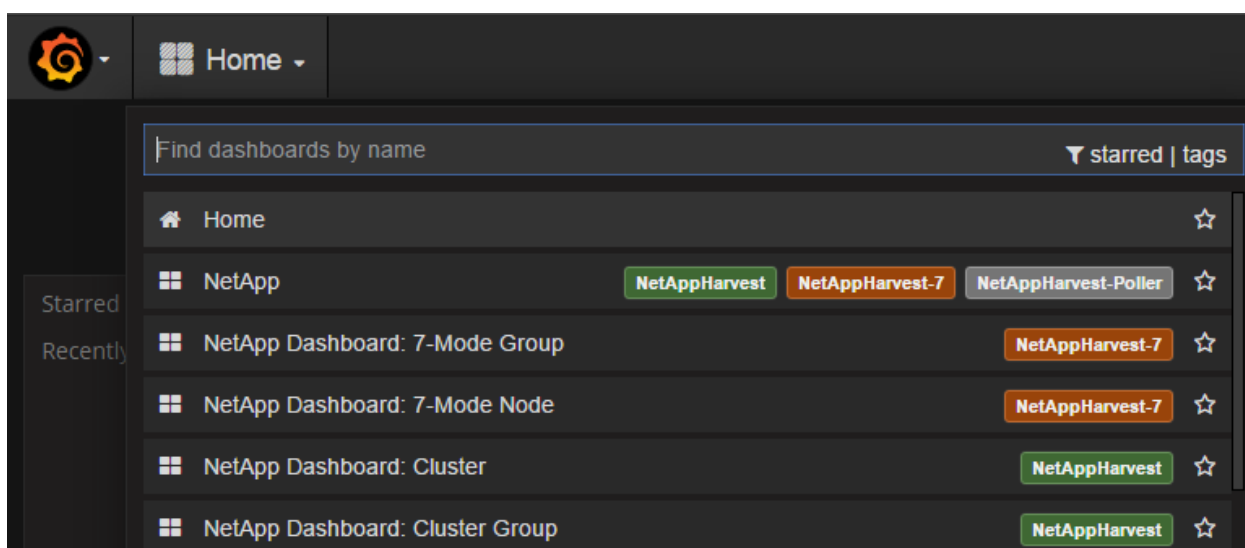
The program `netapp-manager` includes the ability to import dashboards to Grafana. The program will import all files in the `grafana` subdirectory and will overwrite any existing dashboards of the same name. For this reason it is strongly recommended to not customize any dashboards supplied by Harvest directly (i.e. adding or removing panels) but instead use 'save as' for your own copy. In this way on a subsequent Harvest upgrade updated dashboards can be imported and you will not lose any customizations.

**TIP:** If your environment contains only Clustered Data ONTAP you might like to remove the 7-mode dashboards from the `grafana` subdirectory prior to importing to avoid clutter from these unnecessary dashboards.

- Run the `netapp-manager -import` command to import all dashboards from the `grafana` subdirectory to the Grafana server

```
root@host:/opt/netapp-harvest# /opt/netapp-harvest/netapp-manager -import
[OK ] Will import dashboards to http://localhost:3000
[OK ] Imported dashboard [db_netapp.json] successfully
[OK ] Imported dashboard [db_netapp-dashboard-7-mode-node.json] successfully
continues...
```

- Refresh in the Grafana GUI and observe that all the dashboards have loaded:



### c. Exporting dashboards

The program `netapp-manager` includes the ability to export dashboards from Grafana to `.json` files. The program will export dashboards to the `grafana` subdirectory and will overwrite any existing dashboard files of the same name. By default all dashboards will be exported. To export a subset of dashboards that match a tag populate the `global` section `grafana_dl_tag` parameter with the tag name you wish to export.

The export feature can also be used to make a backup of your customized dashboards, or to share with other users on the NetApp Communities.

1. Run the `netapp-manager -export` command to download dashboards from the Grafana server to the `grafana` subdirectory

```
root@host:/opt/netapp-harvest# /opt/netapp-harvest/netapp-manager -export
[OK      ] Will export dashboards from http://localhost:3000
[OK      ] Exported [db/netapp] to dashboard file [db_netapp.json]
[OK      ] Exported [db/netapp-detail-graphite-server] to dashboard file [db_netapp-detail-
graphite-server.json]
continues...
```

## 8. Managing NetApp Harvest workers

NetApp Harvest consists of two main programs:

### 9. netapp-worker

- One process (background daemon) per monitored host. A monitored host can be a (7-mode node, a cDOT cluster, or an OCUM server. This process is responsible for data collection and metrics forwarding to Graphite

### 10. netapp-manager

- A management utility to start, stop, restart, and check status of one or more netapp-worker processes. It also provides the ability to bulk import and export dashboards from Grafana.

Each program uses a configuration file (default is netapp-harvest.conf) for the parameters of the requested action. Although it is typical to use netapp-manager to start and stop worker daemons, it is also possible to start workers directly as well.

#### a. Using netapp-manager

Command-line help for netapp-manager is shown by running the program with no arguments:

```
[root@host netapp-harvest]# ./netapp-manager

Usage: /opt/netapp-harvest/netapp-manager {-status|-start|-stop|-restart|-export|-import} [-poller <str>] [-group <str>] [-conf <str>] [-h]

PURPOSE:
  Stop/start/restart/status of netapp-worker poller processes and import/export Grafana dashboards
VERSION:
  1.3
ARGUMENTS:
  Required (one of):
    -status      Show status of all matching pollers
    -start       Start all matching pollers
    -stop        Stop all matching pollers
    -restart     Stop/start all matching pollers
    -export      Export dashboard json files from Grafana server
    -import      Import dashboard json files to Grafana server
  Optional:
    -poller <str>  Filter on poller names that match <str> (only valid for status/start/stop/restart)
    -group <str>  Filter on group names that match <str> (only valid for status/start/stop/restart)
    -conf <str>   Name of config file to use to find poller name
                  (default: netapp-harvest.conf)
    -h           Show this help
    -v           Show verbose output, and if starting pollers, start them with verbose logging
EXAMPLES:
  Check status of all pollers
  netapp-manager -status
  Start all pollers that are not already running
  netapp-manager -start
  Stop all pollers with poller name that includes netapp
  netapp-manager -stop -poller netapp
  Restart all pollers with poller name that includes netapp at group AMS
  netapp-manager -restart -poller netapp -group AMS
  Import all dashboards from the grafana subdirectory to Grafana
  netapp-manager -import
```

Start all enabled pollers in the netapp-harvest.conf configuration file that are not already running:

```
[root@host ~]# /opt/netapp-harvest/netapp-manager -start
STATUS      POLLER      GROUP
#####
[STARTED]   blob1      nl
```

```
[STARTED] blob2 nl
[STARTED] bumblebee ny
[STARTED] cmdemo uk
```

Stop all pollers in the `netapp-harvest.conf` configuration file that are running:

```
[root@host ~]# /opt/netapp-harvest/netapp-manager -stop
STATUS POLLER GROUP
#####
[STOPPED] blob1 nl
[STOPPED] blob2 nl
[STOPPED] bumblebee ny
[STOPPED] cmdemo uk
```

Check status of all pollers in the `netapp-harvest.conf` configuration file:

```
[root@host ~]# /opt/netapp-harvest/netapp-manager -status
STATUS POLLER GROUP
#####
[RUNNING] blob1 nl
[RUNNING] blob2 nl
[NOT RUNNING] bumblebee ny
[RUNNING] cmdemo uk
```

Restart all pollers with blob in the poller name in the `netapp-harvest.conf` configuration file:

```
[root@host ~]# /opt/netapp-harvest/netapp-manager -restart -poller blob
STATUS POLLER GROUP
#####
[STOPPED] blob1 nl
[STOPPED] blob2 nl
[STARTED] blob1 nl
[STARTED] blob2 nl
```

Import dashboards found in the `grafana` subdirectory to the Grafana instance specified in the `netapp-harvest.conf` configuration file:

```
root@host:/opt/netapp-harvest# /opt/netapp-harvest/netapp-manager -import
[OK ] Will import dashboards to http://localhost:3000
[OK ] Imported dashboard [db_netapp.json] successfully
[OK ] Imported dashboard [db_netapp-dashboard-7-mode-node.json] successfully
continues...
```

## b. Using netapp-worker

Command-line help for netapp-worker is shown by running the program with no arguments:

```
[root@host netapp-harvest]# ./netapp-worker
Usage: /opt/netapp-harvest/netapp-worker -poller <poller> [-conf <conf_file>] [-daemon] [-v] [-h]

PURPOSE:
  Collect performance data from Data ONTAP or OCUM and submit to Graphite.
VERSION:
  1.2X20
ARGUMENTS:
  Required:
    -poller <poller>   Poller section to run
  Optional:
    -conf <conf_file>  Name of conf_file to use to find poller_section
                       (default: netapp-harvest.conf)
    -h                 Output this help text
    -v                 Output verbose output to stdout and logfile
    -daemon            Daemonize process (linux only)
EXAMPLE:
Run poller netapp-1 interactively in verbose mode
netapp-worker -poller netapp-1 -v

Run poller netapp-2 as a daemon
netapp-worker -poller netapp-2 -daemon

Run poller netapp-5 from conf file test.conf as a daemon
netapp-worker -poller netapp-5 -conf test.conf -daemon
```

Start the nltl-infra-vmware poller from the netapp-harvest.conf configuration file interactively:

```
root@host:/opt/netapp-harvest# ./netapp-worker -poller nltl-infra-vmware
[2016-10-13 22:24:24] [NORMAL ] WORKER STARTED [Version: 1.3] [Conf: netapp-harvest.conf]
[Poller: nltl-infra-vmware]
[2016-10-13 22:24:24] [NORMAL ] [main] Poller will monitor a [FILER] at [nltl-infra-vmware:443]
[2016-10-13 22:24:24] [NORMAL ] [main] Poller will use [password] authentication with username
[admin] and password [*****]
[2016-10-13 22:24:24] [NORMAL ] [main] Collection of system info from [nltl-infra-vmware] running
[NetApp Release 8.3.2P2] successful.
[2016-10-13 22:24:24] [NORMAL ] [main] Found best-fit monitoring template (same generation and
major release, minor same or less): [cdot-8.3.2.conf]
[2016-10-13 22:24:24] [NORMAL ] [main] Added and/or merged monitoring template [/opt/netapp-
harvest/template/default/cdot-8.3.2.conf]
[2016-10-13 22:24:25] [NORMAL ] [main] Metrics will be submitted with graphite_root
[netapp.perf.nl.nltl-infra-vmware]
[2016-10-13 22:24:25] [NORMAL ] [main] Using graphite_meta_metrics_root
[netapp.poller.perf.nl.nltl-infra-vmware]
[2016-10-13 22:24:25] [NORMAL ] [main] Startup complete.  Polling for new data every [60]
seconds.
```

**Note:** The last line with 'Startup complete' informs you that the poller has completed startup tasks and is now entering the instance and data polling loops



## 9. Configuring NetApp Harvest (advanced)

This chapter describes additional advanced configuration possibilities of NetApp Harvest. For a typical installation this level of detail is likely not needed.

### a. Populate the configuration file (advanced)

In chapter 3, section [Populate the configuration file](#) included the minimum steps to add a monitored system. There are however many other parameters that can be modified in the configuration file which are explained in this section.

The configuration file uses an INI style format, with section headers enclosed in brackets, and parameter = value pairs to define parameters that belong to that section.

The headers (example: [header\_name] ) can consist of:

11. [global]: General settings. The section header name 'global' is a reserved name.
12. [default]: Settings which will be the default values for each admin created poller section. Any parameter that is common to all (or most) pollers can be set here for convenience. Parameters that vary between pollers can then be set explicitly (defined or overriding) in a specific poller definition. The section header name 'default' is a reserved name.
13. [monitored\_hostname]: Settings for an admin specified monitored system. The section header name is the name for the poller and can be any alphanumerical string and is usually the hostname of the monitored system. Also, when using the default `graphite_root`, it will be the system name submitted in the metrics path unless overridden by the parameter `display_name`.

The parameter value pairs (example: `hostname = 10.64.31.110`) specify the parameters to use in the configuration. When parsing a parameter = value pair the parameter will be set to the non-whitespace characters prior to the first = character, and the value will be set to the remainder of the line, with leading and trailing whitespace characters removed. Do not enclose your values in quotes as the quotes themselves would be included in the value.

Blank lines, and any line whose first non-whitespace character is a '#', will be ignored.

This table below lists all possible parameters for the 'global' section:

Parameter	Description	Example or possible values (default in <b>bold</b> )
<code>grafana_api_key</code>	Used to authenticate and import and export dashboards. Generate an API key from the Grafana UI.	<code>eyJrIjoiNXdsW9FCbmo2SDN2RGRuUndzWkNsSmxoOFdKdUttekkiLCJuIjoiZGFzaGJvYXJkIHRlc3RpbmciLCJpZCI6MX0=</code>
<code>grafana_url</code>	Used to connect with the Grafana server to import and export dashboards. Enter the URL as you would to access the server from your browser	<a href="http://server51:3000">http://server51:3000</a>
<code>grafana_dl_tag</code>	Used to selectively download dashboards. If a value is set only dashboards with that Grafana tag will be exported. If set to blank or the parameter is missing all dashboards will be exported.	<code>NetAppHarvest</code>

This table below lists all possible parameters for the 'default' and/or admin defined poller sections:

Parameter	Description	Example or possible values (default in <b>bold</b> )
graphite_enabled	Determines if metrics should be sent to Graphite (value of 1) or not (value of 0). Can be useful during testing if you want to view log messages but prevent any metrics from being sent to Graphite	0 <b>1</b>
graphite_server	IP address or hostname of the Graphite server. If multiple graphite servers are required use Graphite relay and have it duplicate metrics to multiple servers	<b>localhost</b> 10.10.10.100 graphite.acme.com <b>etc.</b>
graphite_port	The port number on the graphite_server which is listening for newline separated metrics	<b>2003</b> 2013
graphite_proto	The protocol on which the graphite_server is listening on the graphite_port	<b>tcp</b> udp
normalized_xfer	The raw counters in Data ONTAP and OCUM use varying units for data rate and capacities. To ease display and analysis harvest normalizes these to a unit of bytes, kilobytes, megabytes, or gigabytes depending on the value of this parameter. For provided dashboards normalize all FILER pollers to mb_per_sec and OCUM to gb_per_sec (which makes all capacity info in GB; ignore what _per_sec might imply)	b_per_sec kb_per_sec <b>mb_per_sec</b> gb_per_sec
normalized_time	The raw counters in Data ONTAP use varying time units. To ease display and analysis harvest normalizes these to a unit of microsec, millisec, or sec. For provided dashboards normalize all FILER pollers to millisec.	microsec <b>millisec</b> sec
graphite_root	<p>The Graphite metrics path prefix for the poller. It can contain hardcoded strings as well as poller section variables enclosed in {} (i.e. {parameter_name}).</p> <p>If the value is set to <code>default</code> or is blank then Harvest will automatically populate this path based on the group parameter and the OS type of the monitored system using prefix of:</p> <p>7-mode: <code>netapp.perf7.{group}.{display_name}</code>            cDOT: <code>netapp.perf.{group}.{display_name}</code>            OCUM (per cDOT host discovered):  <code>netapp.capacity.{group}.{display_name}</code></p> <p>The Grafana dashboards provided require a setting of <code>default</code></p>	<b>default</b> ntap.performance.NL.{display_name} <b>etc.</b>
graphite_meta_metrics_root	<p>The Graphite meta metrics path prefix for the poller. It can contain hardcoded strings as well as poller section variables enclosed in {} (i.e. {parameter_name}).</p> <p>If the value is set to <code>default</code> or is blank then Harvest will automatically populate this path based on the group parameter and the OS type of the monitored system using prefix of:</p> <p>7-mode: <code>netapp.poller.perf7.{group}.{display_name}</code>            cDOT: <code>netapp.poller.perf.{group}.{display_name}</code></p>	<b>default</b> ntap.poller.{group}.{display_name} <b>etc.</b>

Parameter	Description	Example or possible values (default in <b>bold</b> )
	<p>OCUM (per cDOT host discovered):  <code>netapp.poller.capacity.{group}.{display_name}</code></p> <p>The Grafana dashboards provided require a setting of <code>default</code></p>	
<code>host_type</code>	The host type of the monitored system. Required by the NetApp SDK to connect properly. A value of <b>FILER</b> is used for a Data ONTAP (7-mode or ONTAP) system, and a value of <b>OCUM</b> is used for a OnCommand Unified Manager 6.x system	<b>FILER</b> OCUM
<code>host_port</code>	The port number on the monitored host which is listening for SSL API requests.	<b>443</b> 8443 <b>etc.</b>
<code>host_enabled</code>	Determines if the poller is enabled (value of 1) or disabled (value of 0). Can be useful if you want to disable a poller from being started by <code>netapp-manager</code>	0 <b>1</b>
<code>template</code>	<p>Specifies a whitespace or comma separated list of collection templates. Each template will be loaded and merged in the order it is listed. Merging is done for 1<sup>st</sup> level keys, so if changing parameters within a key redefine all attributes of it.</p> <p>A template name of <code>default</code> is also supported in which case Harvest will discover the software release and use the most appropriate file in the <code>template/default</code> directory. Custom templates must be placed in <code>template/&lt;template_name&gt;</code></p> <p>As an example, to use the default template but disable LUN collection you could create a template <code>no_luns.conf</code>, which has a single 'lun' key in the poller section with <code>enabled =&gt; 0</code>. Then set the template to <code>default,no_luns.conf</code> and the effective config will be the default with lun collection disabled.</p>	<b>default</b> <code>default,no_luns.conf</code> <b>etc.</b>
<code>data_update_freq</code>	<p>The number of seconds between each successive data poll. For 7-mode and ONTAP 60 is typical. If collection does not complete in time ("skips" are reported in the logs) then it can be increased. Or, if a select set of metrics is desired at a more granular interval it can also be lowered.</p> <p>For OCUM it updates capacity metrics every 15 minutes so polling more frequently than 900 seconds would not be beneficial.</p>	<b>60</b> 900 <b>etc.</b>
<code>ntap_autosupport</code>	<p>Harvest can log the poller status messages (syslog or event message) to the 7-mode or ONTAP monitored system every 4 hours. If set to 1 this feature is enabled, if set to 0 it is disabled.</p> <p>For 7-mode the RBAC role required to submit an EMS message is minimal and it is recommended to set to be enabled.</p> <p>For cDOT the RBAC role required to submit an EMS message is substantial (system node *) and therefore if using an RBAC user is recommended to set to be disabled</p>	0 <b>1</b>

Parameter	Description	Example or possible values (default in <b>bold</b> )
latency_io_reqd	Latency metrics can be inaccurate if the IOP count is low. This parameter sets a minimum number of IOPs required before a latency metric will be submitted and can help reduce confusion from high latency but low IOP situations.	<b>10</b> 25 <b>etc.</b>
auth_type	Sets the authentication method for the poller. For 7-mode and OCUM only <code>password</code> is supported. For ONTAP <code>password</code> and <code>ssl_cert</code> are supported. Authentication type <code>password</code> uses username/password authentication and requires parameters <code>username</code> and <code>password</code> to be populated as well. Authentication type <code>ssl_cert</code> uses SSL certificates and requires parameters <code>ssl_cert</code> and <code>ssl_key</code> to be populated as well.	<b>password</b> <code>ssl_cert</code>
username	When using <code>auth_type</code> of <code>password</code> , this parameter contains the username of the user which Harvest should use to authenticate	<code>netapp-harvest</code> <code>monitor</code> <b>etc.</b>
password	When using <code>auth_type</code> of <code>password</code> , this parameter contains the password of the <code>username</code> which Harvest should use to authenticate. The <code>password</code> must be entered in plaintext. Because the user account can be one with least privilege access the risk of exposure is limited.	<code>netAPP135!</code> <code>dkd%#@\$=99</code> <b>etc.</b>
ssl_cert	When using <code>auth_type</code> of <code>ssl_cert</code> , this parameter contains the filename of the SSL public certificate located in the <code>cert</code> subdirectory ( <code>.pem</code> file) which Harvest should use to authenticate.	<code>netapp_ssl.pem</code> <code>monitor_pub.pem</code> <b>etc.</b>
ssl_key	When using <code>auth_type</code> of <code>ssl_cert</code> , this parameter contains the filename of the SSL private key located in the <code>cert</code> subdirectory ( <code>.key</code> file) which Harvest should use to authenticate.	<code>netapp.key</code> <code>monitor.key</code> <b>etc.</b>
display_name	The name of the monitored system to use in the path of submitted metrics. If this parameter does not exist or is blank it will be automatically be populated with the name from the poller section header. In most cases it is easiest to populate the section header with the short hostname of the device and not use this parameter, but if using multiple pollers for a single monitored system these headers must be unique you can instead use this parameter to locate metrics from both pollers under the same system name in the Graphite metrics path.	<code>cluster99</code> <code>ams4nasla</code> <b>etc.</b>
group	The group of the monitored system to use in the path of submitted metrics. It is preferred to use a short name to consume less screen space and spaces are not allowed; use dash or underscore instead if needed. In an earlier release of Harvest this parameter was named <code>site</code> ; if <code>site</code> is used it will be aliased to <code>group</code> internally. The default <code>graphite_root</code> uses <code>group</code> to help organize monitored systems. The default Grafana template driven dashboards also	<code>VDC2</code> <code>AWS-NW</code> <code>SFO8</code>

Parameter	Description	Example or possible values (default in <b>bold</b> )
	require a specific metrics path that includes <code>group</code> . If you have a small installation you can of course place all monitored systems at a single named group. For a large installation still only one level of group is recommended to maintain compatibility with the default Grafana dashboards.	

## b. Working with collection templates (advanced)

Each Harvest poller requires a collection template file that specifies the objects and counters that should be collected. The template file is set using the `template` parameter in the poller configuration. If missing or set to `default` it will cause Harvest to discover the software release and use the most appropriate file in the `template/default` subdirectory.

It is also possible to create your own customized template to collect more or less objects and counters, modify the metrics path, or use a plugin to tweak metrics or create meta metrics. Custom templates must be placed in the `template` subdirectory, and then the poller section parameter `template` must be set to include the name of your template file.

Multiple templates can be specified, separated with comma or whitespace. If multiple templates are listed each will be merged at the 1<sup>st</sup> level, in the order they are listed. So if redefining any keys be sure to include all parameters.

Some example templates are found in the `template/example` subdirectory, and the default templates in `template/default` can also be a great starting place to edit your own.

In the subsections we will review the file format and typical edits that might be desired.

### i. Collection template file format

The collection template is simply Perl code. Accordingly, syntax rules are quite strict requiring exact proper placement of brackets, curly braces, quotes, etc. If the file contains any syntax errors Harvest will terminate and the Perl error will be logged.

There are two hashes which are defined in the template:

14. `%poller`: Contains the collection details (objects, counters, plugins, etc)
15. `%override`: Contains parameter overrides for situations were counter metadata published by Data ONTAP is incorrect

#### 1. poller hash

The poller hash contains all collection details for objects and counters of the specific poller.

Here is an example for collection of the “processor” object type on cDOT:

```

1  %poller = (
2  'processor' =>
3  {
4      counter_list      => [ qw ( node_name node_uuid instance_name instance_uuid
5                             domain_busy processor_busy ) ],
6      graphite_leaf     => 'node.{node_name}.processor.{instance_name}',
7      plugin            => 'cdot-processor',
8      plugin_options    => {'cpu' => 0, 'domain' => 1},
9      enabled           => '1'
10 }
11 );

```

By line number (R = required, O = optional):

16. 1: (R) The reserved name `%poller` that defines the poller hash

17. 2: (R) The desired object name to collect, enclosed in single quotes
18. 3: (R) Opening curly brace for details of the object from line 2
19. 4 - 5: (R) the reserved name `counter_list => [qw (counter1 counter2 counter3 )]`
20. 6: (R) the reserved name `graphite_leaf => 'leaf.path'`. The `leaf.path` can also contain variables from the counter value results enclosed in `{}`. In this case the `node_name` and `instance_name` counters will be resolved for each counter instance when building the metrics path. Appended to the `graphite_leaf` is always the counter name itself, or in the case of counters which are 2D or 3D, an additional set of counter names using a period as separator. From the example it might be `<graphite_root>.node.cluster99-01.processor.processor0.processor_busy`.
21. 7: (O) the reserved name `plugin => 'plugin-name'`
22. 8: (O) the reserved name `plugin_options => {'opt1' => 1, 'opt2' => 0}`
23. 9: (O) the reserved name `enabled => '1'` (enabled) or `'0'` (disabled)
24. 10: (R) Closing curly brace for details of the object ending the section that began at line 3
25. 11: (R) Closing of entire poller list of objects; if more objects were also polled the block of lines 2-10 would be repeated, separated by a comma

In this example the 'processor' object metadata will be discovered (once at startup) and all instances of the 'processor' object will be discovered (and refreshed every 60min). Then, values will be collected for requested counters for all instances of the object (and refreshed every `data_update_freq`). These counter values are raw and will be transformed to a useful metric in various ways depending on object metadata discovered at startup. This transformation typically involves doing some math between the values of the current and previous poll. Thereafter the data is normalized to units (ex. `mb_per_sec`) specified in the poller configuration.

Because a plugin has been configured the metrics list, in graphite format, are sent to the plugin. The plugin is Perl code located in `plugin` subdirectory that can modify the metrics list by modifying the metrics path, removing elements, or create new 'meta' metrics. The `plugin_options` parameter give additional flexibility to control how the plugin behaves. In the case of `processor` the `cpu` flag allows per CPU metrics to be submitted or not, and the `domain` flag allows per CPU processing domain metrics to be submitted or not. Discussion of possible flags is typically found in the header of the plugin itself. In this case the plugin will calculate the per domain totals and remove the per CPU totals before passing the modified metrics list back to the main Harvest program for forwarding to Graphite.

## 2. override hash

The override hash contains overrides of counter metadata and can be used when the metadata configuration in Data ONTAP does not match the observed counter values. There are cases where a counter is defined as raw when it is actually a rate, or when the unit is defined as `b_per_sec` but is actually `kb_per_sec`. To correct these defects in Data ONTAP an override can be made which allows the standard poller logic to function correctly in spite of the metadata defect.

Here is an example for collection of the "nfsv3" object type on ONTAP:

```

1 %override = (
2 'nfsv3' =>
3 {
4     nfsv3_throughput      => { unit => b_per_sec },
5     nfsv3_read_throughput => { unit => b_per_sec },
6     nfsv3_write_throughput => { unit => b_per_sec },
7     access_total         => { properties => rate }
8 }
9 );

```

By line number (R = required, O = optional):

26. 1: (R) The reserved name `%override` that defines the override hash

- 27. 2: (R) The desired object name surrounded in single quotes
- 28. 3: (R) Opening curly brace for details of the object from line 2
- 29. 4 - 7: (R) the counter name `counter_name => { metadata => 'new_value' }`
- 30. 8: (R) Closing curly brace for details of the object ending the section that began at line 3
- 31. 9: (R) Closing of entire poller list of objects; if more objects were also polled the block of lines 2-8 would be repeated, separated by a comma

In the example above `nfsv3_throughput` counter was defined as `per_sec` instead of `b_per_sec`, and the `access_total` counter was defined as a delta whereas most users care about it in the form of a rate (i.e. `access_total` ops/sec). These edits allow Harvest to perform the correct math and compensate for counter metadata defects.

## ii. Disabling collection of an object type

From the [poller hash](#) section we see that each object has a field `enabled` that can have a value of 0 (disable collection) or 1 (enable collection). If some metrics are not needed or cause collection to run too long you can copy the appropriate default file for your monitored system from the `template/default` subdirectory to the `template` subdirectory, remove all object types that you want to leave unmodified, and set `enabled` to 0 for all object types that you want to disable. Then set the `template` for the poller in your configuration file to `default,disable.conf`.

## iii. Adding a new object type

In the [poller hash](#) section the structure required to collect an object was documented. If a new object is needed copy the appropriate default file for your monitored system from the `template/default` subdirectory to the `template` subdirectory to prepare for your edits. Next add a new object type to this custom template by duplicating an existing configuration block and replace the object type and counters with those desired. The [perf-counters-utility](#) can be very useful to browse the objects, instance names, counters and values in order to populate the parameters for the new object to collect. If additional modification of the metrics string is desired optionally create a plugin to modify. Then set the `template` for the poller in your configuration file to `default,extra.conf` and the default collection will be supplemented with your added object types.

## iv. Adding or removing counters for an existing object type

In the [poller hash](#) section the structure required to collect an object was documented. If you want to add or remove counters copy the appropriate default file for your monitored system from the `template/default` subdirectory to the `template` subdirectory to prepare for your edits. Next locate the object type and add or remove counters from the `counter_list` field as desired. The [perf-counters-utility](#) can be very useful to browse the counters and values in order to determine the counters you want to collect. Finally, set this new file as your `template` for the poller in your main configuration file.

## v. Creating a plugin

A plugin is run by Harvest after the metrics have been transformed but before they are sent to Graphite. In some cases a plugin does a simple job like fixing up a metric string to be more user friendly. In other cases a plugin will create roll-up summaries of data. An example of such roll-up exists for processor, the example from the [poller hash](#) section. Processor data is tracked per CPU, so for example we get Kahuna domain per CPU0-7. An administrator however cares about cumulative Kahuna domain utilization across all CPU, and not a per CPU basis. Using a plugin system any custom logic can be implemented to change the metrics submitted easily.

A plugin has access to any variable defined in Harvest although these must be declared using a Perl `our` statement. The variable that includes the metrics to be submitted is in the `@emit_items` array. This array allows an 'incoming' list of metrics to be read and manipulated in any way providing an 'output' list of metrics that should be set into that same `@emit_items`. See example plugins from the default configuration files for more information.

### c. OCUM capacity metrics with the OPM performance metrics data provider

The default OCUM template submits metrics in a hierarchy that is parallel to the performance metrics submitted by Harvest. If it is desired to use the OPM external data provider feature for performance data, and Harvest for capacity metrics, an example custom collection template can be used. This template is named `template/example/ocum-opm-hierarchy.conf`. To use it copy the file to `template/ocum-opm-hierarchy.conf`, and then add a poller section in the format:

```
[INSERT_OCUM_HOSTNAME_HERE]
hostname           = INSERT_IP_ADDRESS_OR_HOSTNAME_OF_OCUM_HOSTNAME
group              = INSERT_GROUP_IDENTIFIER_HERE
host_type          = OCUM
data_update_freq  = 900
normalized_xfer    = gb_per_sec
template           = ocum-opm-hierarchy.conf
graphite_root      = netapp-capacity.Clusters.{display_name}
graphite_meta_metrics_root = netapp-capacity-poller.INSERT_OCUM_HOSTNAME_HERE
```

### d. perf-counters-utility

The `perf-counters-utility` located in the `util` subdirectory can be useful to browse the counter manager system in Data ONTAP from the API. It can be useful when troubleshooting or preparing to collect a new object or additional counters to understand the data format and values that Harvest will receive and parse.

It connects to a live Data ONTAP system and displays:

32. objects
33. available instances [for a provided object]
34. available counters [for a provided object]
35. counter data in raw format [for a provided object and instance]

Command-line help for it is shown by running the program with no arguments:

```
Usage: perf-counters-utility -host <host> -user <user> -pass <pass> [-o|-in|-c|-d] [-f <family>]
[-n <name> | -u <uuid>]
PURPOSE:
  Collect performance data from Data ONTAP's cDOT performance counter
  subsystem that uses a hierarchy of: object-instance-counter
ARGUMENTS:
Required:
  -host      Hostname to connect with
  -user      Username to connect with
  -pass      Password to connect with
Required (one of):
  -o        Display object list
  -in       Display instance list
  -c        Display counter list
  -d        Output counter data
Required with -in, -c, -d:
  -f <family> Object family
Required (one of) with -d:
  -n <name>   Name of instance to graph
  -u <uuid>   Name of UUID to graph
Optional:
  -h        Output this help text
  -v        Output verbose output to stdout
EXAMPLE:
Display object list:
  perf-counters-utility -host sdt-cdot1 -user admin -pass secret -o
Display instance list:
  perf-counters-utility -host sdt-cdot1 -user admin -pass secret -in -f volume
Display counter list:
  perf-counters-utility -host sdt-cdot1 -user admin -pass secret -c -f volume
Display counter data for specific instance:
  perf-counters-utility -host sdt-cdot1 -user admin -pass secret -d -f volume -n vol0
```



## e. perf-counters-to-excel

The perf-counters-to-excel utility located in the `util` subdirectory can be useful to create an Excel spreadsheet of the counter inventory for a given Data ONTAP release. It can be useful when preparing to collect a new object or add additional counters.

The utility connects to a live Data ONTAP system and creates an Excel spreadsheet with one tab per object type. In each tab it then lists each available counter by name including a description and other metadata.

Command-line help for it is shown by running the program with no arguments:

```
Usage: perf-counters-to-excel -host <host> -user <user> -pass <pass> [-h]
PURPOSE:
  Creates an Excel file named perf_counters_<relname>.xlsx with object and counter details.
  Supports both 7-mode and cDOT
ARGUMENTS:
  -host          Hostname to connect with
  -user          Username to connect with
  -pass          Password to connect with
EXAMPLE:
  perf-counters-to-excel -host sdt-cdot1 -user admin -pass secret
```

Example output once opened in Excel and navigating to the processor tab:

name	desc	type	is_key	privilege-level	properties	unit	base-counter	label-info
cp_rupt_time	CP rupt time percentage			diag	percent	percent	processor_el	
cp_rupts	Number of CP rupts per second			diag	rate	per_sec	apsed_time	
domain_busy	Array of processor time in percentage spent in various domains	array		diag	percent	percent	processor_el	idle,kahuna,storage,ocol,nwk_exclusi
domain_switches	Number of domain switches per second			diag	rate	per_sec	apsed_time	
freebsd_cp_time	FreeBSD CP time array	array		diag	delta	none		USER,NICE,SYS,IN
grab_kahuna	Number of grab kahuna domain per second			diag	rate	per_sec	processor_el	

## 10. Threshold configuration

The provided Grafana dashboards include singlestat panels which are green/yellow/red depending on the value displayed. The thresholds implemented are found in the following table:

Object type	Parameter	Green	Yellow	Red
Volume/Node/SVM	Read Latency	0-20	20-30	30+
Volume/Node/SVM	Write Latency	0-2	2-10	10+
Volume/Node/SVM	Avg Latency	0-20	20-30	30+
Volume/Node/SVM	Other Latency	0-20	20-30	30+
Volume	% capacity Used	0-75	75-85	85+
Aggr	% capacity Used	0-75	75-85	85+
CPU	% Avg CPU busy	0-60	60-80	80+
Disk	% Busiest disk busy	0-60	60-80	80+
Disk	CP reads / writes	0-.5	.5-.75	.75+
Ethernet	Port Utilization	0-60	60-80	80+
iSCSI, FCP	Read Latency	0-20	20-30	30+
iSCSI, FCP	Write Latency	0-2	2-10	10+
iSCSI, FCP	Avg Latency	0-20	20-30	30+
NFS, CIFS	Read Latency	0-20	20-30	30+
NFS, CIFS	Write Latency	0-2	2-10	10+
NFS	Highest latency of any op type	0-20	20-30	30+
CIFS	Avg Latency	0-10	10-30	30+
LUN	Read latency	0-20	20-30	30+
LUN	Write latency	0-2	2-10	10+
Windows fileservices	AD auth latency	0-10	10-20	20+
Windows fileservices	Vscan latency	0-250	250-400	400+

## 11. Troubleshooting

Each NetApp Harvest poller has its own logfile in the `log` subdirectory (`/opt/netapp-harvest/log`). The logfile name is constructed from the poller section and configuration file name. Logfile content includes basic startup information, poller status messages (every 4 hours), and warnings or errors should they occur. If Harvest is run in the foreground (`netapp-worker` without the `-daemon` flag) this same content is also mirrored to standard output.

If the default logging level does not provide enough information to resolve an issue, verbose logging can also be enable by using the `-v` option. Verbose logging is quite detailed and generates a large logfile so only use it when diagnosing a specific issue.

No logfile management (rotation based on size or date) is done by Harvest. In a healthy configuration a single line of text is logged every 4 hours so if a logfile is large enough that it requires management there is some other problem which needs to be resolved.

When verifying a first time setup it can be useful to start the `netapp-worker` manually in the foreground (use Control-C to exit) and review the log messages in real-time. If there are errors resolve and try again repeatedly until collection is successful. Once collection is successful start as a background process using `netapp-manager`.

Specific errors and solutions are discussed in the following subsections.

### a. “Can't locate NaServer.pm in @INC (you may need to install the NaServer module)”

If a module cannot be loaded by `netapp-worker` this will not be reported in the logfile. You may see that when starting a poller via `netapp-manager` it immediately stops. If you start the poller via `netapp-worker` the output is:

```
[root@host ~]# /opt/netapp-harvest/netapp-worker -poller cluster99
Can't locate NaServer.pm in @INC (you may need to install the NaServer module) (@INC contains:
/opt/netapp-harvest/lib /etc/perl /usr/local/lib/perl/5.18.2 /usr/local/share/perl/5.18.2
/usr/lib/perl5 /usr/share/perl5 /usr/lib/perl/5.18 /usr/share/perl/5.18 /usr/local/lib/site_perl
.) at ./netapp-worker line 35.
BEGIN failed--compilation aborted at ./netapp-worker line 35.
[root@host ~]#
```

It stopped immediately because the `NaServer` module couldn't be located. The `NaServer.pm` file (and others from the SDK) were looked for in `lib` subdirectory but were not found. To resolve ensure that the SDK files are installed correctly into the `lib` subdirectory.

## b. “Required parameter [hostname] not defined”

There are several required parameters for each poller section. If a required parameter is not provided the poller will log an ERROR and terminate. Here is an example:

```
[root@host ~]# /opt/netapp-harvest/netapp-worker -poller cluster99
[2015-06-25 20:24:09] [NORMAL ] WORKER STARTED [Version: 1.2RC1] [Conf: netapp-harvest.conf]
[Poller: cluster99]
[2015-06-25 20:24:09] [ERROR ] [main] Required parameter [hostname] not defined
[root@host ~]#
```

To resolve check the configuration file section for the poller and ensure the parameter is listed (or is inherited from the [default] section) and resolve.

## c. “Unable to resolve hostname”

The hostname of each monitored system must be able to be resolved. During initialization a check is done to resolve and if not possible a WARNING will be logged:

```
[root@host ~]# /opt/netapp-harvest/netapp-worker -poller cluster99
[2015-06-26 11:06:07] [NORMAL ] WORKER STARTED [Version: 1.2RC1] [Conf: netapp-harvest.conf]
[Poller: cluster99]
[2015-06-26 11:06:07] [NORMAL ] [main] Poller will monitor a [FILER] at [cluster99:443]
[2015-06-26 11:06:07] [NORMAL ] [main] Poller will use [password] authentication with username
[netapp-harvest] and password [*****]
[2015-06-26 11:06:07] [WARNING] [connect] Unable to resolve hostname [cluster99]. Ensure correct
hostname resolution or use IP address instead.
[2015-06-26 11:06:07] [WARNING] [sysinfo] Update of system-info cache DOT Version failed with
reason: in Zapi::invoke, cannot connect to socket
[2015-06-26 11:06:07] [WARNING] [main] system-info update failed; will try again in 10 seconds.
```

The system will retry indefinitely to handle transient network failures, but if the hostname is incorrect or DNS is not configured it will never succeed. Resolve the issue by adding the hostname to DNS or the /etc/hosts file.

## d. “Update of system-info cache DOT Version failed with reason: in Zapi::invoke, cannot connect to socket”

Harvest uses the NetApp SDK to connect to the monitored system. If the SDK is unable to establish a TCP connection it will log a WARNING with ‘cannot connect to socket’:

```
[root@host ~]# /opt/netapp-harvest/netapp-worker -poller cluster99
[2015-07-08 22:53:48] [NORMAL ] WORKER STARTED [Version: 1.2RC1] [Conf: netapp-harvest.conf]
[Poller: cluster99]
[2015-07-08 22:53:48] [NORMAL ] [main] Poller will monitor a [FILER] at
[cluster99.nltestlab.hq.netapp.com:443]
[2015-07-08 22:53:48] [NORMAL ] [main] Poller will use [password] authentication with username
[netapp-harvest] and password [*****]
[2015-07-08 22:54:09] [WARNING] [sysinfo] Update of system-info cache DOT Version failed with
reason: in Zapi::invoke, cannot connect to socket
[2015-07-08 22:54:09] [WARNING] [main] system-info update failed; will try again in 10 seconds.
```

There are various reasons why Harvest cannot connect (incorrect IP address, a firewall is blocking communication, a mistake in routing, the cluster LIF is not reachable, etc) so do a test to connect directly to the port.

Use this test to connect to the port using nc (netcat):

```
[root@host ~]# nc -vz -w 5 cluster99.nltestlab.hq.netapp.com 443
nc: connect to 192.168.100.33 port 443 (tcp) timed out: Operation now in progress
```

The above shows there is no reachability to that cluster on port 443/tcp. In this case we determined that the firewall policy on the cluster was limiting SSL access to a specific management host [and it was not our Harvest host]. We added the harvest poller host to have access and tried again:

```
[root@host ~]# nc -vz -w 5 cluster99.nltestlab.hq.netapp.com 443
nc: connect to 192.168.100.33 port 443 (tcp) failed: Connection refused
```

Better, so we can now reach the system on port 443/tcp but it rejects access, this means there is no service listening on the port. We checked the cluster and the SSL server was disabled. We enabled it and tried again:

```
[root@host ~]# nc -vz -w 5 cluster99.nltestlab.hq.netapp.com 443
nc: connect to 192.168.100.33 443 port [tcp/https] succeeded!
```

With the above success Harvest should now be able to connect to the SSL port.

### e. “Update of system-info cache DOT Version failed with reason: Authorization failed”

Harvest must authenticate but if the configured credentials are incorrect or lack access a WARNING will be logged:

```
[root@host ~]# /opt/netapp-harvest/netapp-worker -poller cluster99
[2015-06-26 11:28:38] [NORMAL ] WORKER STARTED [Version: 1.2RC1] [Conf: netapp-harvest.conf]
[Poller: cluster99]
[2015-06-26 11:28:38] [NORMAL ] [main] Poller will monitor a [FILER] at [10.64.28.242:443]
[2015-06-26 11:28:38] [NORMAL ] [main] Poller will use [password] authentication with username
[netapp-harvest] and password [*****]
[2015-06-26 11:28:41] [WARNING] [sysinfo] Update of system-info cache DOT Version failed with
reason: Authorization failed
[2015-06-26 11:28:41] [WARNING] [main] system-info update failed; will try again in 10 seconds.
```

Verify the configured credentials (username/password, or SSL certificate details) are correct.

### f. “Update of system-info cache DOT Version failed with reason: No response received from server”

With Data ONTAP 7-mode TLS must be enabled. See the [Enabling TLS](#) section for more.

### g. “[lun] data-list poller next refresh at [2015-07-28 02:17:00] not scheduled because it occurred in the past”

Each object will update according to the configured `data_update_freq`. The poller updates each object (lun, volume, processor, etc) in a serialized manner. If cumulative time to collect all object types is greater than the `data_update_freq` then on some polls some objects will be skipped and log this message. If you see a message like this sporadically you may decide it is acceptable to miss a few polls here and there. If you see it regularly however you can investigate in the Grafana dashboard “NetApp Detail: Harvest Poller” to see collection times of each object type and analyze what to do.

If a single object is taking too long you can opt to not collect it, or to separate collection of it into a different dedicated poller, potentially with a different `data_update_freq`.

If many object types are taking a long time and collection is over a WAN you can setup a harvest collector local to the monitored system to reduce the impact of the WAN latency. If not over a WAN then look to configure a less frequent `data_update_freq` or split collection of the objects across multiple templates and multiple pollers.

It is also possible that you are using the NetApp Management SDK 4.0 and reverse hostname resolution is failing after a timeout; see [Setting HTTP/1.0 because reverse hostname resolution \(IP -> hostname\) fails](#).

### h. Setting HTTP/1.0 because reverse hostname resolution (IP -> hostname) fails. To enable HTTP/1.1 ensure reverse hostname resolution succeeds.

The NetApp Management SDK 5.4 introduced support for HTTP/1.1. This version of HTTP requires a Host: header in each request and the SDK will do a `gethostbyaddr` lookup on the hostname defined in the poller configuration. If this reverse hostname lookup fails Harvest will detect it and force HTTP/1.0

avoiding a http protocol request error (400 Bad Request). But, the SDK 5.4 still performs this `gethostbyaddr` lookup and if times out, instead of succeeding or failing quickly, it can extend each API request by several seconds. This issue is logged as NetApp Bug ID 935453.

To resolve ensure forward and reverse hostname resolution is possible (DNS entries or `/etc/hosts` file entry), or, use a SDK version prior to 5.4, such as 5.3.1.

**i. No best-fit collection template found (same generation and major release, minor same or less) found in [/opt/harvest/template/default]. Exiting;**

Harvest connects to a monitored system, learns the software release, and then attempts to find a suitable collection template. A suitable collection template is one with the same generation and major release, and minor release that is the same or less.

For example release 8.3.1RC1 is generation 8, major 3, minor 1 and acceptable templates in `template/default` subdirectory would be `cdot-8.3.1.conf` or `cdot-8.3.0.conf`, in that order. If neither exists this error will be generated and Harvest will exit.

The reason for this logic is oftentimes counters will be added, removed, or modified, and a new template is needed to map to these changes. But, it also means that Harvest must also be updated for successful collection of a new release family that is introduced.

If you encounter this error you should check if a new version of Harvest is available that supports the release natively and update to it. If a new Harvest version is not yet available, you can try to copy a similar release template in `template/default` subdirectory to match your new release (i.e. `cp ./template/default/cdot-8.2.0.conf ./template/default/cdot-8.3.0.conf`), start your poller, and check the log for errors modifying any template parameters as needed.

If you are running an internal release that does not use the generation/major/minor naming convention mentioned, instead copy the most similar release template to `template/custom-release.conf` and specify that filename explicitly in the `template` parameter of your poller configuration.

**j. “Poller status: status, secs=14400, api\_time=1806, plugin\_time=55, metrics=459594, skips=0, fails=0”**

Every 4 hours the poller logs a summary status message about the preceding time period. This information can be helpful to understand quickly the workload and health of each poller. If a poller has many skips or failures it is reason to look in more detail at the cause.

The fields can be interpreted as:

Field name	Description
<code>secs</code>	Number of seconds of activity which are summarized in the other fields; typically 14400s, or 4 hours
<code>api_time</code>	Cumulative seconds that the poller was waiting on the API to respond.
<code>plugin_time</code>	Cumulative seconds that the poller was waiting on plugins to process.
<code>metrics</code>	Cumulative number of metrics submitted by the poller
<code>skips</code>	Cumulative number of polls that were skipped because previous poll was still running
<code>fails</code>	Cumulative number of polls that failed

### An Example:

```
[2015-07-28 20:50:52] [NORMAL ] Poller status: status, secs=14400, api_time=1806, plugin_time=55, metrics=459594, skips=0, fails=0
```

The poller was active  $1806 + 55 = 1861$  seconds out of 14400 seconds, or about 13% of the time. This would indicate that there is plenty of 'budget' when the poller is idle waiting for the next poll.

A total of 459594 metrics was submitted, or 1914 per minute on average. This information could be useful when sizing Graphite to understand the quantity of updates submitted by the poller.

There were 0 skips indicating the poller never had to skip a poll because the previous was not yet complete. If there were skips the time and object impacted would also be logged at the time of the skip.

There were 0 failures indicating the poller always succeeded. If there were failures the exact reason would also be logged at the time of the failure.

### k. Additional support

NetApp Harvest is available for download from the NetApp Toolchest and uses a community support model. If you want to share tips with others, report a bug, or get help with your configuration please search and make posts to the 'OnCommand Storage Management Software Discussions' section on NetApp Communities:

36. <http://community.netapp.com/t5/OnCommand-Storage-Management-Software-Discussions/bd-p/oncommand-storage-management-software-discussions>

37. **Note:** Please use a tag of "netapp-harvest" to make it easily identifiable

## 12. References

The following references were used in this paper:

- NetApp ToolChest  
<http://mysupport.netapp.com/tools/index.html>
- NetApp Communities: OnCommand Storage Management Software Discussions  
<http://community.netapp.com/t5/OnCommand-Storage-Management-Software-Discussions/bd-p/oncommand-storage-management-software-discussions>
- NetApp Communities: OnCommand Storage Management Software Articles and Resources  
<http://community.netapp.com/t5/OnCommand-Storage-Management-Software-Articles-and-Resources/tkb-p/oncommand-storage-management-software-articles-and-resources>
- NetApp OnCommand Unified Manager Documentation  
<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=61373>
- NetApp OnCommand Performance Manager Documentation  
<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=61809>

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.