

CERTIFIED CLOUD AND SERVICE PROVIDER

TRACKING CAPACITY UTILIZATION WITH RED HAT STORAGE

March 2016

CONTENTS

INTRODUCTION.....	3
METERING.....	3
DETERMINING PEAK USAGE.....	4
CCSP CAPACITY TRACKING TOOL.....	4
INSTALLATION PROCESS.....	5
RHEL7.....	5
RHEL6.....	5
UBUNTU 14.04LTS.....	6
CONFIGURATION OVERVIEW.....	7
USAGE.....	7
WEB INTERFACE.....	8
CLI EXTRACT TOOL.....	9
APPENDICES.....	10
FILE LOCATIONS.....	10
MANUAL CAPACITY REPORTING.....	11
RED HAT CEPH STORAGE.....	11
RED HAT GLUSTER STORAGE.....	12

INTRODUCTION

Red Hat's Certified Cloud and Service Provider (CCSP) program provides Red Hat partners with everything they need to;

- enable your cloud or managed service
- offer Red Hat products on your cloud, and
- grow your business.

The partner program workflow covers a number of business and technical activities which serve to enable Red Hat partners to deliver solutions to customers based on Red Hat products. In any cloud environment one of the key activities that must be in place is efficient and accurate billing and metering processes.

This document focuses on the technical steps required to implement metering for the following Red Hat Storage (**RHS**) products

- Red Hat Ceph Storage (**RHCS**)
- Red Hat Gluster Storage (**RHGS**)

METERING

Partners may implement metering for RHS in two ways; shared or dedicated. Which method is used is dependent upon how the storage is to be consumed.

shared

Metering for a storage cluster in shared mode aligns to a multi-tenant deployment model. The metering strategy for a 'shared' cluster is based on tracking the raw capacity consumed (*raw_used*) within the storage cluster.

dedicated

A dedicated metering mode implies that the cluster being measured is only supporting a single customer/tenant. In this mode the metering metrics changes to account for the cluster size. In a Ceph cluster the cluster size is derived from the overall raw capacity of the cluster. For gluster, dedicated metering is based on the number of peers (nodes) in the cluster.

In both of these metering scenarios, usage **must** be reported as an observed **peak usage** over the monthly billing period.

DETERMINING PEAK USAGE

In order to track peak usage over a billing period, capacity data points must be taken at regular intervals and summarized. There are two supported strategies for extracting this data;

- *automated*, with the provided CCSP capacity tracking tool¹ (**recommended**)
- *manual*, by following the [manual capacity reporting](#) guidelines and implementing a local scripting solution.

Regardless of the approach implemented, the implementation should also support ad-hoc queries to account for missed reporting cycles.

CCSP CAPACITY TRACKING TOOL

The provided CCSP capacity tracking tool implements a daemon process which queries the storage cluster for capacity metrics at specific intervals. The data is then stored in a RRD data store and retained for six months(default). A single instance of the tool is installed into each cluster that needs to be monitored.

In addition to tracking and recording the capacity metrics, the tool also provides a web interface and standalone extract tool.

- the web interface provides a rolling 30 day view of the capacity metrics captured, and also shows the current peak values that would be used for billing purposes.
- the extract tool allows the admin to run ad hoc queries against the RRD data store to show the relevant billing metric over a given period.

The ccsp tool is available for RHEL6, RHEL7 and Ubuntu 14.04LTS².

1 <https://github.com/pcuzner/ccsp>

2 the Ubuntu platform has had limited testing

INSTALLATION PROCESS

The installation process varies slightly by platform, accounting for variances in system startup framework (systemd vs upstart) and package format (rpm or deb).

Installation is required on just one node within the storage cluster. Which node is chosen depends upon the cluster type. For a gluster environment, any node may be chosen that is a peer within the cluster. With Ceph, install the tool on either a monitor or an admin node.

RHEL7

1. Add the rhel-7-server-optional-rpms repo to the target server
2. For gluster monitoring only (you need to be subscribed to the rhs-3 channel)

```
# yum install gstatus
```

3. Install the ccsp rpm
4. Edit the configuration file at `/etc/rhs-usage.conf` to define the type of cluster to be monitored (**shared** is the default). Other settings like web interface and web port may also be changed.
5. Open the firewall port (example **assumes** the default of port 8080 is being used)

```
# firewall-cmd --zone=public --add-port=8080/tcp --permanent
# firewall-cmd --zone=public --add-port=8080/tcp
```

6. Refresh systemd, enable and start the service

```
# systemctl daemon-reload
# systemctl enable rhs-usage
# systemctl start rhs-usage
```

7. Confirm the service is operational by checking the status of the service and/or looking at the log file `/var/log/rhs-usage.log`

RHEL6

1. Add rhel-6-server-optional-rpms repo
2. For gluster monitoring only (you need to be subscribed to the rhs-3 channel)

```
# yum install gstatus
```

3. Install the ccsp rpm
4. Edit the configuration file at `/etc/rhs-usage.conf` to define the type of cluster to be monitored (**shared** is the default). Other settings like web interface and web port may also be changed.
5. There are no firewall considerations typically enabled on a RHEL6 RHGS build - if your environment has added them you will need to adapt the approach above to allow access to the web interface.

6. Check the service is accessible, and start it

```
# status rhs-usage
# start rhs-usage
```

7. Confirm the service is operational by checking the status of the service and/or looking at the log file `/var/log/rhs-usage.log`

UBUNTU 14.04LTS

To support deployments to the Ubuntu 14.04 LTS release a .deb package is available in the github repository. The package itself has been generated using the python stdeb module.

1. install the deb package

```
# sudo dpkg -i ccsp_0.XXX_all.deb
```

NB. if this fails with dependency issues, run `sudo apt-get -f install` and then retry the original command, or alternatively install the package with `gdebi` to automatically handle package dependencies.

2. Edit the configuration file at `/etc/rhs-usage.conf` to define the type of cluster to be monitored (**shared** is the default). Other settings like web interface, web port and sample frequency are also set in this file - but leave it at the defaults for now.
3. When the web component is running as part of the tool you may need to update the firewall to allow access to the relevant port
4. Check the service is accessible, and start it

```
# status rhs-usage
# start rhs-usage
```

5. Confirm the service is operational by checking the status of the service and/or looking at the log file `/var/log/rhs-usage.log`

CONFIGURATION OVERVIEW

The ccsp package installs a systemd or upstart service called rhs-usage. This service runs in two main threads; a data collector thread, and an optional web interface. By default, data collection is done every hour, with data points summarized every **4** hours. The summarized data is added to the RRD data store, which maintains a rolling **180** days worth of capacity and configuration data.

The configuration file (/etc/rhs-usage.conf) provides the runtime configuration for the rhs-usage service. Normal usage will just require changes to two parameters; `storage_type` and `run_mode`. These parameters determine the logic used when calculating a cluster's peak usage.

The example below shows the configuration file used to monitor an RHGS cluster, which for billing and accounting purposes is running in 'shared' mode.

```
[main]
# storage_type can be either gluster(default) or ceph
storage_type = gluster

# run mode is shared(default) or dedicated
run_mode = shared

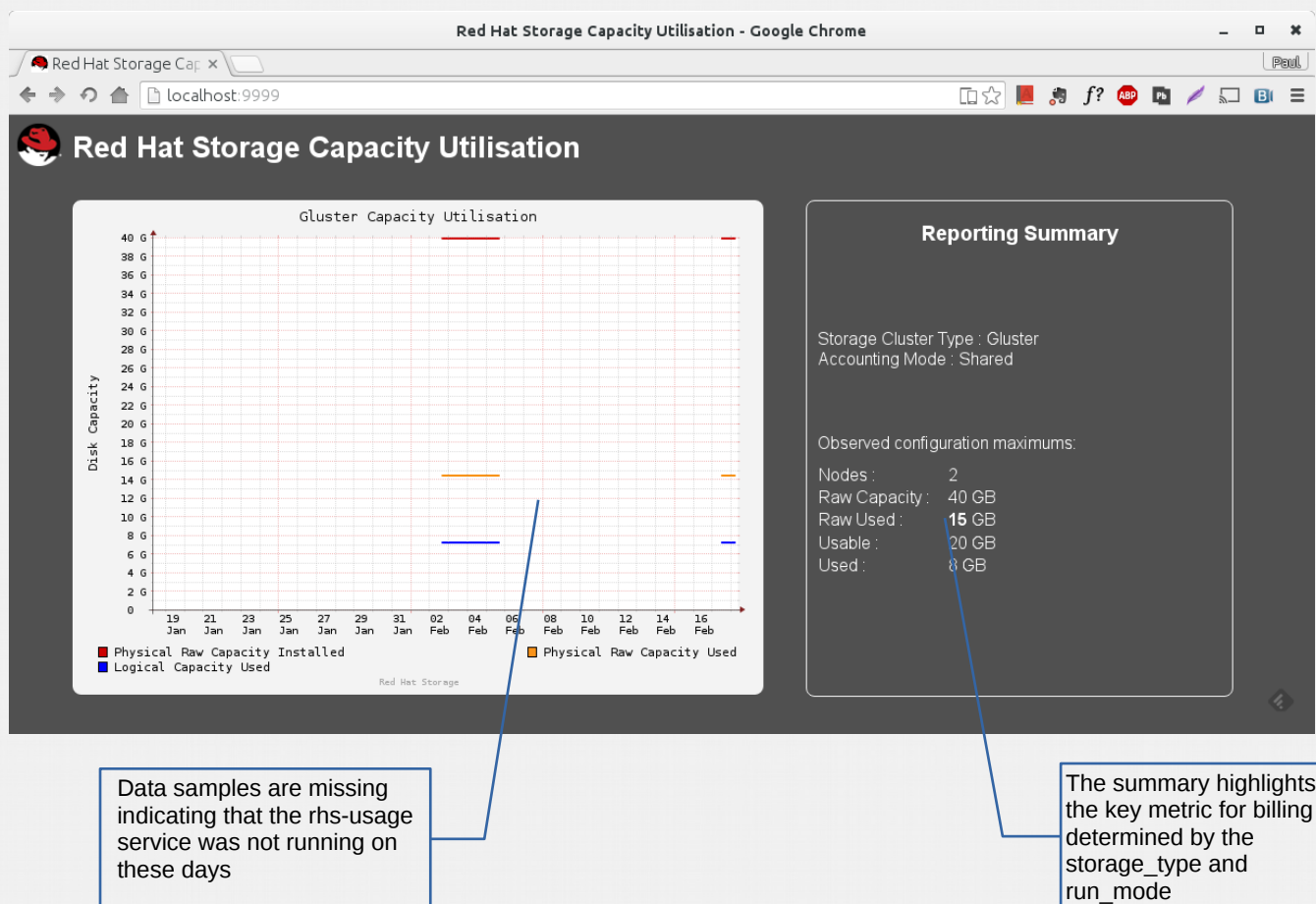
# start the web 'server' to show usage over the last month
web_server = y
web_server_port = 8080
```

USAGE

For the majority of the time, the rhs-usage service will just quietly collect and store capacity information for the relevant storage cluster. However, once a month the tool needs to be used to report the peak usage value back to Red Hat for charging purposes. There are two ways to determine the appropriate reporting value; the web interface or the CLI extract tool.

WEB INTERFACE

The web interface automatically starts with the rhs-usage service, if the configuration file sets the web_server parameter to 'y'. When this value is set, simply point your browser at the node running the rhs-usage service, specifying the relevant port number (*default is 8080*).



The above screenshot illustrates the following;

- data points from the last 30 days are shown, which help to identify potential capacity trends in the cluster's capacity
- when the rhs-usage service is not running, gaps will show up in the graph. Ideally the service should be active at all times.
- the reporting summary section shows
 - the run time configuration parameters
 - the current maximums over the 30 day window for each of the capacity metrics
 - the monthly billing metric to be reported back to Red Hat is highlighted (bold)

For general queries, this interface provides a simple overview of the capacity trends within the cluster over the last 30 days.

CLI EXTRACT TOOL

The interactive nature of the web interface may not work well with all partner work-flows. Where local automation or consolidated reporting across multiple clusters is required a different approach is needed.

The ccsp package provides a command line tool called `rhsextract`. This tool is designed to provide partners with a mechanism to extract data from the RRD data store on a more ad hoc basis. This allows local scripts to be written to aggregate the usage patterns across multiple RHS clusters.

```
[root@rhgs31u1-2 ~]# rhsextract -h
usage: rhsextract [-h] [-d] [-s START_DATE] [-e END_DATE] [-l LASTNDAYS] [-i]

optional arguments:
  -h, --help            show this help message and exit
  -d, --debug            run in debug mode
  -s START_DATE, --start START_DATE
                        start date for the extract (yyyy/mm/dd)
  -e END_DATE, --end END_DATE
                        end date for the extract(yyyy/mm/dd)
  -l LASTNDAYS, --last LASTNDAYS
                        go back this number of days from today
  -i, --info            show the first and last observation dates from the rrd
                        file
```

The tool reads the same configuration file as the `rhs-usage` service, which ensures that the `run_mode` and accounting logic is the same across both tools.

Here's an example extract which focuses on two specific days;

```
[root@rhgs31u1-2 ~]# rhsextract -s 2016/02/16 -e 2016/02/18
Using rrd filename of '/var/ccsp/rhs-usage.rrd'
Query looks ok to use
Billing Mode: SHARED
Max 'raw_used': 15 GB
```

This example shows that given the *shared* `run_mode` of the tool and the time period provided – the maximum raw capacity used within the cluster was **15GB**.

APPENDICES

FILE LOCATIONS

Path Name	Description
/var/ccsp/rhs-usage.rrd	rrd data file
/var/log/rhs-usage.log	Log file
/etc/rhs-usage.conf	run-time / environment configuration
/var/www/ccsp	static files that provide the web page

MANUAL CAPACITY REPORTING

Some partners may opt to implement their own data gathering techniques for CCSP metering and billing of RHS products. In this scenario, data collection, storage and reporting must be implemented to adhere to the following 5 key principles

1. data collection should adhere to Red Hat recommended practices
2. data gathering should take place at least hourly
3. data collected should be summarized, and peak usage for raw_capacity, raw_used, node count must be stored.
4. At least one summary record is required per day for each of these metrics
5. Data summarized over the billing period must show the peak usage values for a given period.

The following sections define the recommended best practice for data collection across the RHS products.

RED HAT CEPH STORAGE

For either a shared or dedicated cluster, the basis for capacity reporting is the Ceph CLI sub-command 'df'³. This command shows a Ceph cluster's disk capacity and utilization data, but which value to use varies depending upon the billing mode of either shared or dedicated.

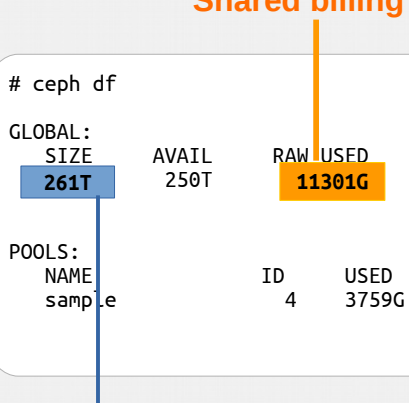
Consider the example below;

Shared billing mode

```
# ceph df
```

GLOBAL:	SIZE	AVAIL	RAW USED	%RAW USED
	261T	250T	11301G	4.21

POOLS:	NAME	ID	USED	%USED	MAX AVAIL	OBJECTS
	sample	4	3759G	1.40	85161G	965377



Dedicated billing mode

By default the 'df' sub-command will present the information in human readable format. In order to track and report Ceph statistics consistently for a given billing period these values need to be provided as GiB⁴.

Once the data is collected, the manual process will need to record and summarize the data in accordance with the principles identified above.

³ <https://access.redhat.com/documentation/en/red-hat-ceph-storage/1.3/red-hat-ceph-administration-guide/chapter-2-high-level-monitoring>

⁴ GiB is Gibibyte – capacity is calculated using base 2, instead of base 10.

RED HAT GLUSTER STORAGE

Extracting cluster capacity information from a glusterfs cluster should be done using the `gstatus`⁵ tool. This tool provides several output modes to summarize the current state of the cluster, including json.

`gstatus` should be run at predefined intervals on one of the RHGS nodes. The relevant data depends upon the billing mode of the environment – shared or dedicated. The example below shows the output of the `gstatus` command, and identifies which fields are important to which type of billing mode.

Dedicated billing mode

```
root@rhgs31u1-2 ~]# gstatus -o json
2016-03-01 15:45:32.960979 {"brick_count": 2, "bricks_active": 2, "glfs_version": "3.7.1", "node_count": 2,
"nodes_active": 2, "over_commit": "No", "product_name": "Red Hat Gluster Storage Server 3.1 Update 1",
"raw_capacity": 42920312832, "sh_active": 2, "sh_enabled": 2, "snapshot_count": 0, "status": "healthy",
"usable_capacity": 21460156416, "used_capacity": 15594610688, "volume_count": 1, "volume_summary":
[{"snapshot_count": 0, "state": "up", "usable_capacity": 21460156416, "used_capacity": 7797305344,
"volume_name": "repl"}]}
```

Shared billing mode

Once the data is collected, the manual process will need to record and summarize the data in accordance with the principles identified above. Also note that the capacity information provided by `gstatus` is in bytes, and the reporting metric needs to be presented as GiB.

⁵ https://access.redhat.com/documentation/en-US/Red_Hat_Storage/3.1/html/Administration_Guide/sect-gstatus_Command.html