

Tech Brief

Monitoring Capacity Utilization via the CLI

This document describes how to use the `lsmetricstat` CLI command to monitor snapshot pool capacity utilization of your Actifio appliances.

- [What you can do with `lsmetricstat`](#) on page 2
- [Flow Metrics](#) on page 3
- [Store Metrics](#) on page 3
- [Example of `lsmetricstat` usage](#) on page 4
- [Retrieving Statistics by Metric Name or by Date Range](#) on page 4
- [Collecting Metric Statistics during a Scheduled Nightly Job](#) on page 5
- [Configuring Daily Cleanup Operations](#) on page 5
- [`lsmetricstat` Filters and Switches](#) on page 5

Introduction

Capacity utilization monitoring is important in understanding how Actifio system resources are used. Monitoring utilization also helps in managing changes to the environment from events, SLA changes, re-ingesting of applications, and load handling.

Capacity utilization metrics of an appliance can help in:

- Ensuring if there is enough headroom to meet SLAs.
- Provision new appliances and additional resources to meet SLAs.
- Review and analyze the historical utilization data to predict the application level impact due to corresponding appliance utilization.
- Forecast purchasing budget by analyzing the appliance, application type utilization history, and trends.
- On-board the new applications by learning the current utilization and expected app impact.

The Actifio appliance collects metrics:

- hourly (including both stats of currently running jobs and notification when the job has finished starting from the 0th minute of the hour)
- daily at midnight (appliance time)

Use the Actifio CLI command `udsinfo lsmetricstat` to see `metricstat` information. Details of this CLI command are in [lsmetricstat Filters and Switches](#) on page 5.

What you can do with lsmetricstat

Capacity utilization metrics can be collected at regular intervals or during the completion of every job. The Actifio appliance collects stats for capacity utilization monitoring for resource types such as Application, Group, Consistency Group, Pool, and Appliance.

Common use cases for capacity management include:

Capacity utilization monitoring

Capacity utilization monitoring is done with an intention of understanding how effectively system resources are used. Appliance utilization metrics can be used for viewing the current utilization of an individual appliance. These metrics are primarily used for detailed load balancing within the appliances as well as short-term capacity management.

Load changes handling

This metrics gathering can be done to ensure there are enough resources for ongoing operations during times of load change due to on-boarding of new applications, SLA changes, re-ingesting of applications, and other temporary load peaks such as garbage collection.

Capacity planning

Capacity planning involves forecasting resource needs based on current use and potentially expected events or changes.

Capacity planning is typically based on the data storage and helps you to understand the usage of snapshot or dedup pool storage or VDisk consumption. Capacity planning can be planned based on the saturation level for each application. Dedup Pool Storage is same as threshold of Dedup Pool. See [Configuring Daily Cleanup Operations](#) on page 5 for more detail.

Efficient data handling

A scheduled nightly job collects snappool metrics and provides summarized data. The data summary can be run based on the application and jobclass. See [Collecting Metric Statistics during a Scheduled Nightly Job](#) on page 5 for more detail.

Daily cleanup operations for platform cleanup from the stats table can be performed. You can configure the cleanup operations based on the time-interval and amount of data that is present in the metricstat table. See [Collecting Metric Statistics during a Scheduled Nightly Job](#) on page 5 for more detail.

Group application utilization metrics

This feature is useful for capacity planning for aggregated types of applications and for data flow between appliances. Applications can be grouped by type, by appliance, by host or server, by snapshot pool, or by using logical and consistency groups defined by the user.

View historical utilization metrics

Historical metrics are critical to understand system behavior and can give insight into the behavior of types or groups of applications and that can be used when on-boarding new applications of a similar type. You can view a graph of historical utilization metrics over various periods with various data frequency.

Flow Metrics

Use flow metrics to learn time-related appliance utilization details. These deal with throughput, especially data processing and movement. Flow metrics are useful for understanding system utilization when snapshot jobs move data into Actifio management. They typically occur within a defined window. Flow metrics can also be used for analyzing the amount of data that is processed by dedup engine every day.

Following flow metrics can be gathered for monitoring capacity utilization:

Ingest to snapshot pool

This is used to calculate the amount of data written to snapshot pool. It is a system-level write flow measurement used for performance diagnostics, forecasting and capacity planning.

```
udsinfo lsmetricstat -filtervalue metricname=DataIngest
```

StreamSnap throughput

This provides an aggregated system-level measure that is affected by the local snapshot pool, network, and the remote snapshot pool. These metrics can be used for troubleshooting the capacity planning related issues.

```
udsinfo lsmetricstat -filtervalue metricname=DataIngest
```

For example the metric names can be DataIngest | NetworkData | DataRead

Vault to cloud throughput

This is an aggregated system-level flow metric which evaluates the vault to cloud storage and is dependent on local snapshot pool and the target object storage.

```
udsinfo lsmetricstat -filtervalue metricname=DataRead
```

For example the metric name scan be DataIngest | DataRead

Store Metrics

Store metrics can help us gather utilization details that are not related to time, but are capacity based. Typically store metrics are useful for understanding the data storage. Store metrics can be gathered for analyzing usage of snapshot or dedup pool storage and VDisk consumptions.

The following Store metrics can be gathered for monitoring the capacity utilization:

Snapshot pool storage used

This store metric specifies the usage stats of each protected resource which consumes storage from Snapshot pool. Using this metric, user can see the amount of storage consumed at appliance level and also drill down to the next level to see the storage consumption by each host, application and application groups.

```
udsinfo lsmetricstat -filtervalue metricname=totalused
```

You can have safe limit of 80% of total Snap Pool.

VDisks used

This store measure indicates the VDisk utilization at appliance level as well as at specific application, application group, host, snap pool, organization, or universe.

```
udsinfo lsmetricstat -filtervalue metricname=vdiskcount
```

You can have up to 2048 VDisks for CDS, and 1000 to 5000 VDisks for Sky depending on your Sky license.

Example of lsmetricstat usage

```
$ udsinfo lsmetricstat -filtervalue grouptype=application
```

id	clusterid	appid	hostid	hostname	apptime	jobclass	poolid	poolname
97299	0	96388	96387	rkvm-01	rkvm-01	unknown	73	act_per_poolsakha
vdiskcount		1	Number	Daily	2016-08-03 06:52:58.222		2016-08-04 06:52:58.222	
2016-08-04 06:52:58.241			application					
97300	0	96388	96387	rkvm-01	rkvm-01	unknown	73	act_per_poolsakha
totalused		8192	Bytes	Daily	2016-08-03 06:52:58.222		2016-08-04 06:52:58.222	
2016-08-04 06:52:58.251			application					
97302	0	59798	59516	inband	C:\	unknown	73	act_per_poolsakha
vdiskcount		1	Number	Daily	2016-08-03 06:52:58.222		2016-08-04 06:52:58.222	
2016-08-04 06:52:58.269			application					
97303	0	59798	59516	inband	C:\	unknown	73	act_per_poolsakha
totalused		8192	Bytes	Daily	2016-08-03 06:52:58.222		2016-08-04 06:52:58.222	
2016-08-04 06:52:58.272			application					
97305	0	7258	7224	ravi-rhel66-2 /		unknown	73	act_per_poolsakha
vdiskcount		1	Number	Daily	2016-08-03 06:52:58.222		2016-08-04 06:52:58.222	
2016-08-04 06:52:58.299			application					
97306	0	7258	7224	ravi-rhel66-2 /		unknown	73	act_per_poolsakha
totalused		8192	Bytes	Daily	2016-08-03 06:52:58.222		2016-08-04 06:52:58.222	
2016-08-04 06:52:58.310			application					
97312	0	7258	7224	ravi-rhel66-2 /	FileSystem	unknown	0	
DataIngest		10468289	Bytes	Daily	2016-08-03 00:00:00.000		2016-08-03 23:59:59.000	
2016-08-04 06:52:58.471			application					
97313	0	59798	59516	inband	C:\	FileSystem	unknown	0
DataIngest		136571984	Bytes	Daily	2016-08-03 00:00:00.000		2016-08-03 23:59:59.000	
2016-08-04 06:52:58.479			application					

Retrieving Statistics by Metric Name or by Date Range

Use `lsmetricstat -metricname dataingest` to collect flow metric values based on the date range. If you do not provide any date range, then all flow metrics are reported.

Metricstats can be retrieved by providing the **metric name** or **date range** using `stattime` attribute using filter option.

Example:

Date Range option using `stattime`.

```
udsinfo lsmetricstat -filtervalue stattime\>"2016-08-02"\&stattime\<"2016-08-04"
```

Collecting Metric Statistics during a Scheduled Nightly Job

You can schedule a nightly job to collect metrics in a summarized format. The data can be summarized either by application or by jobclass.

- Grouping the data by appname and jobclass for stats between 12:00 AM to 11:59 PM.
- Differentiating the daily stats for aap, group, pool by using (stattype, apptype, and jobtype) attributes.
- Aggregating the stats for various resource types such as: Application, Group, Consistency Group, Pool and Appliance.
- Aggregating the stats for applications present in the group on daily basis. Nightly job of psrv can be summarized for all stats by UDP engine persists stats at group level.
- Aggregating the daily utilization stats for all pools based on corresponding pool IDs.

Total metric data per application is spanned across rows, so a nightly job will persist the data in the same table with stattype. This data can be used to generate historical reports.

When this summarized data merged with non summarized data, then based on following pointers, capacity utilization metrics can be analyzed.

Configuring Daily Cleanup Operations

- Configuring how long to save metric stats
- Hourly stats get purged after 14 days
- Daily stats get purged after 60 days
- You can change both settings

Metric stat data persists hourly and daily stats in the same table, that may increase the table size. Platform provides a customized cleanup schedule parameters through which you can clean up the old data.

The Actifio appliance maintains hourly statistics for 30 days and daily statistics for 90 days. You can use the following parameters to configure the duration time for cleanup operations.

Using `udstask setparameter` you can set values for these parameters.

```
udstask setparameter -param hourlystatexpirationindays -value 30
udstask setparameter -param dailystatexpirationindays -value 90
```

lsmetricstat Filters and Switches

The **lsmetricstat** command returns a concise list of stats, or a detailed view of stats that are collected for each application or for a group of resources.

-filtervalue *attrib=value*

Optional. Specifies that you want your report to display any or all of the list of valid filter attributes

The filter will be formed with an attribute and a value. If you specify more than one filter, they must be combined with '&' character (which must be escaped with '\').

For string filters, the only operator allowed is '='. You can also use wildcard character '*'. For example, to match metric stats with jobname begins with 'Job_0001', use '-filtervalue jobname=Job_0001*'.

Some filters allow only predefined constants. For example, stattype allows only Daily or Hourly stats, to match metrics with stattype "Daily" use '-filtervalue status=Daily'.

For number and date types, allowed operators are: =, >, >=, <, <=. To use <, <=, >, or >=, they need to be escaped with '\ ' or enclosed in ' ' or " ", as required by shell.

For example:

- -filtervalue appid\>0
- -filtervalue "appid>0"
- -filtervalue 'appid>0'

Date parameters startdate, enddate, and expiration can also use these operators.

For example:

- -filtervalue starttime\>"2010-01-01 00:00:00"
- -filtervalue starttime\>2010-01-01

Filter Attributes for udsinfo lsmetricstat

Filter	Switches
appid	None
appname	None
apptype	FileSystem Oracle SqlServerWriter Microsoft Exchange Writer Microsoft Hyper-V VSS Writer VMBackup
clusterid	None
endtime	None
groupype	[application jobclass hostid apptype poolid clusterid]
hostid	None
hostname	None
jobclass	[snapshot StreamSnap OnVault]
jobname	None
metricname	[DataIngest DataRead NetworkData vdiskcount totalused]
poolid	None
poolname	None
starttime	None
stattime	None
stattype	[Daily Hourly]
status	[succeeded failed canceled]