

Cloudera

Overview

This page covers how to add and configure a Cloudera connection using GroundWork Cloud Hub. The connection requires a unique set of parameters (e.g., endpoint, credentials). You will need your GroundWork server and virtual environment connector parameters handy.

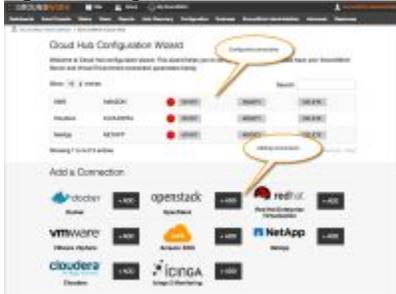
Contents

1.0 Adding a New Connection

The initial Cloud Hub screen is used to add, start, stop, modify, or delete available connectors. Follow the steps below to add a connection. You will need to create a new connection in this way for each region to be monitored.

1. Log in to GroundWork Monitor as an *Administrator*.
2. Select **GroundWork Administration** > **GroundWork Cloud Hub**.
3. Click **+Add** corresponding to the *Cloudera* connector icon.

Figure: Adding a connection



2.0 Configuring a Connection

In the configuration page you will need to enter both the GroundWork server and remote server parameters.

The data the GroundWork server receives comes from the remote virtualization server. The information is pulled from the API on a periodic basis based on the check interval that is set.

2.1 GroundWork Server Parameters

The Groundwork server is where CloudHub will store Cloudera metrics. Often, this is the same server as where CloudHub is running. However, CloudHub can also be run in a distributed environment, on its own node in a Groundwork cluster.

1. Here we enter the GroundWork server parameters, each described in the table below.

Figure: GroundWork server values

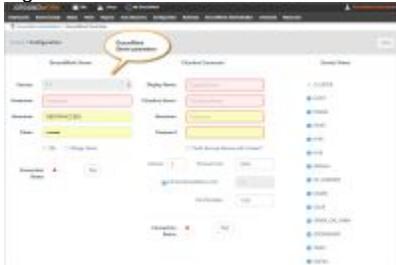


Table: GroundWork server values

Version	The Groundwork server version number. Usually you cannot change this value, and it will default to latest release installed. CloudHub can be configured to talk to versions going back to 7.0. CloudHub auto-detects which versions are available.
Hostname	The host name or IP address where a Groundwork server is running. A port number should not be entered here. If Groundwork is running on the same server, you can enter <i>localhost</i> .
Username	The provisioned Username granted API access on the GroundWork server.
Token	The corresponding API Token (password) for the given Username on the GroundWork server.
SSL	Check the SSL checkbox if your GroundWork server is provisioned with a secure HTTPS transport.
Merge Hosts	If checked, this option combines all metrics of same named hosts under one host. For example, if there is a Nagios configured host named <i>demo1</i> and a Cloud Hub discovered host named <i>demo1</i> , the services for both configured and discovered hosts will be combined under the hostname <i>demo1</i> (case-sensitive).
Test/Connection Status	After entering the GroundWork server parameters, click the <i>Test</i> button to test the connection. A dialog will be displayed with either a success message or, if the server cannot be contacted, an error message will be displayed with information describing why the connection failed. When a successful connection is made, the Connection Status button will change to green.

2.2 Remote Server Parameters

- Here we enter values for the remote Cloudera server, each described in the table below.

Figure: Remote server values

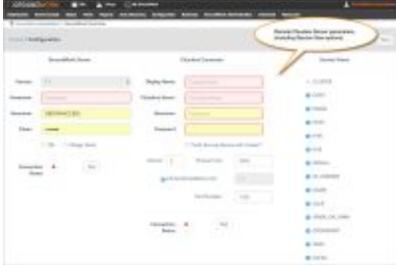


Table: Cloudera server values

Display Name	This is the configuration's name displayed in the list of Cloud Hub connectors on the Cloud Hub home page.
Cloudera Server	The host name or IP address where a Cloudera server is running. A port number should not be entered here.
Username	The provisioned Username granted API access on the Cloudera server.
Password	The corresponding Password for the given Username on the Cloudera server.
Prefix Service Names with Cluster?	A Cluster is a logical entity that contains a set of hosts and the service instances running on the hosts. If this directive is checked, the name of the service will be prefixed with the name of the cluster in the various GroundWork monitoring visualizations. This option is useful when are running two or more Cloudera clusters. Without this option checked, Cloudera services are stored as hosts with a host name directly corresponding to the Cloudera service name. For example, the HDFS service is stored in GroundWork as a host named <i>hdfs</i> . If this box is checked, the hostname will be prefixed by the name of the cluster it is running under. Given a cluster named <i>cluster1</i> , the hostname for HDFS will be stored as <i>cluster1-hdfs</i> . Similarly, the SOLR service will be stored as a hostname <i>cluster1-solr</i> . The default setting is set as disable. If you find Cloudera services are not being mapped to unique GroundWork hostnames, you can use this feature even with a single-cluster Cloudera deployment.
Interval	This is the metric gathering interval for collecting monitoring data from Cloudera and sending it to the GroundWork server. The value is in minutes.
Timeout (ms)	The connection timeout in milliseconds. Normally the default value 5000 is sufficient. When you have a slow network connection, you may want to increase the default value.
Infinite Retries	Check this box if you want CloudHub to infinitely retry connection to Cloudera when the connection fails. When this box is checked, the Retry Limit field is disabled. When this box is unchecked, the Retry Limit field is enabled.
Retry Limit	This entry is the number of retries for the connection and sets a limit on how many attempts are made after a failure. The number set indicates how many connections are attempted before the connection is left in an inactive state. At this point, the connection is suspended and you will need to manually restart it. When a retry limit is exhausted, all hosts managed by this connection are set to the monitor status <i>Unreachable</i> and all services for the matched hosts are set to the status of <i>Unknown</i> .

Port Number	The optional port number for the Cloudera server API. Default is 7180.
Service Views	Optional features of Cloudera are called Cloudera Services. These features are the core components, or services that are managed by Cloudera. Services include HBase, HDFS, Hive, Hue, Impala, KIndexer, Oozie, Solr, Spark, Zookeeper, Yarn, and Kafka. Each of these services has their own rich set of metrics. By default, all services are selected. If checked, service will be monitored. Cloudera also provides Cluster and Host metrics which can be optionally collected. If there are one or more clusters or hosts in the system, they will be automatically detected and collected. If you were collecting metrics for a service, and then unchecked that Cloudera service, the existing hosts and metrics stored in the GroundWork server will be deleted.
Test/Connection Status	After entering the Cloudera server parameters, click the <i>Test</i> button to test the connection. A dialog will be displayed with either a success message or, if the server cannot be contacted, an error message will be displayed with information describing why the connection failed. When a successful connection is made, the Connection Status button will change to green.

2. After the remote server parameters have been entered, click **Save** in the upper right corner to save and write the entries to an XML file in the GroundWork server `/usr/local/groundwork/config/cloudhub` directory. The Cloud Hub connector is assigned an agent ID and that in turn becomes a record locator in Foundation when you begin monitoring.
3. Next, validate both server configurations by selecting the **Test** buttons which will check if the connections are accessible with the given credentials. A dialog will be displayed with either a success message or, if the server cannot be contacted, an error message will be displayed with information describing why the connection failed. When a successful connection is made, the *Connection Status* buttons will change to green.
4. After the credentials have been validated select the **Metrics** link (top navigation) to start customizing metrics for the connection.

3.0 Navigating

From the Configuration page, navigations are on displayed in the top navigation bar:



From here, you can navigate to:

- Home - CloudHub home page
- Metrics - Metrics configuration page associated with this CloudHub connection

When creating a new Cloudera configuration, the Metrics link is not visible until you successfully save the configuration parameters:



Also, the **Save** button is not enabled until all required fields are validated. Here is a new configuration, where you will need to minimally enter the fields displayed in red:

- Groundwork server hostname
- Cloudera display name
- Cloudera server

Figure: Saving



Note that configuration changes are not saved until you click the **Save** button in the top navigation. If you make changes on the configuration page, and forget to save, you will be prompted:

Are you sure?



If you leave now, you will lose all your changes.
Are you sure you want to navigate away?

Yes

No

Once you are satisfied with your configuration settings, click **Save**, then click the **Metrics** link in the navigation bar to start customizing your metrics for this connection.

4.0 Determining Metrics To Be Monitored

1. The section below describes how to configure Cloudera metrics. When you are satisfied with the metric selections click **Save** to commit your changes to Cloud Hub.
2. Click **Home** to return to the main Cloud Hub panel.
3. Click **START** for the specific connector to begin the discovery and data collection process.

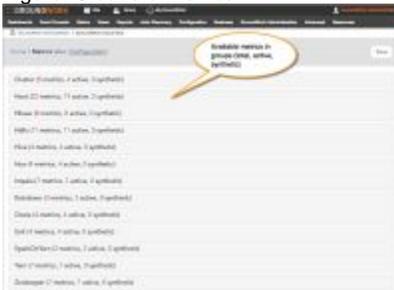
The Cloudera Metrics page is where you customize the lists of metrics being gathered for a connection. Out of the box, a complete list of metrics is provided for clusters, hosts, and Cloudera services. You can customize these metric lists by adding metrics to the list, deleting metrics, as well as creating calculated metric fields called Synthetic metrics.

The Metrics page is displayed in groups of metrics grouped by Cluster, Host, and Cloudera Service collections. The counts of metrics are displayed in the Group bar, and summarized by:

- Total metrics per group
- Active metrics per group
- Synthetic metrics per group

A metric is considered inactive if it is not monitored, (see section on Synthetics below).

Figure: Cloudera metrics



You can configure the metrics for any group by clicking on the group bar. For example, if we click on the bottom *Zookeeper* group bar, the display automatically expands to show all metrics for the Zookeeper Cloudera service:

Figure: Zookeeper Cloudera



Each row in the grid represents a metric. Metrics can be added, edited or deleted. You can directly edit metrics in the grid or use the advanced metric dialog by clicking the **Add** or **Edit** buttons and then configuring all properties of a metric in the dialog. When editing metrics in the grid directly, you will need to click the **Save** button in the top navigation to commit your changes to Cloud Hub. The UI will know if you made changes and remind you to save your changes if you forget.

Grid Fields

The grid displays the following fields:

Monitor?	Check this if you want to enable monitoring of this metric.
Graph?	Check this if you want to graph the values of this metric in time series
Metric Name	The exact Cloudera metric name or a Cloudera metric expression. This field is read-only. Click the <i>Edit</i> button to modify it.
Display Name	Overrides the metric name and stores the metric in GroundWork as a service with this name.
Warning Threshold	Metric value that will trigger a GroundWork Warning alert.
Critical Threshold	Metric value that will trigger a GroundWork Critical alert.

Leaving the threshold fields blank will disable threshold triggers.

Metrics come in two flavors: they can either be Normal or Synthetic metrics.

Normal Metrics

Normal metrics can be:

- Single Metric Names
- Computed Metric Names
- Health Checks metrics
- Configuration metrics (not monitored)

Single Metric Names

Figure: Single metric name with display name

The screenshot shows the 'Edit Metric' form with the following fields:

- Metric Name: load_5
- Metric Format String: %s
- Display Name: HostLoad5Minute
- Default Warning Threshold: 1
- Default Critical Threshold: 2
- Description: Host CPU Load over 5 minutes

 The 'Monitor' and 'Graph' checkboxes are checked, and 'Health Check' is unchecked.

In this example, we have a Host metric named *load_5*. This is the unique name of the metric in Cloudera. In the Display field, we renamed this metric to HostLoad5Minute. Renaming metrics is an optional feature. In this case, we renamed *load_5* to have a more descriptive metric name displayed in the GroundWork Status viewer. We recommend filling out the description field to describe the metric. This metric represents the Host CPU Load averaged over 5 minutes. We have also setup warning and critical thresholds. Note that the metric will be monitored and graphed.

Note that we never use dashes in metric names, only underscores. This is because dashes are not valid variable names in a Cloudera or synthetic expression.

As you type into the Metric Name field, the valid names of metrics available are automatically auto-suggested. This ensures that you use a valid Cloudera metric.

Computed Normal Metrics

Normal metrics can also be computed. They differ from Synthetic metrics in that the value of the metric is an expression, and it is computed on the Cloudera server, not by Cloud Hub.

Figure: Normal metrics - computed Cloudera expression

The screenshot shows the 'Edit Metric' form with the following fields:

- Metric Name: (physical_memory_used / physical_memory_total) * 100 as memory_usage_percent
- Metric Format String: %s
- Display Name: memory_usage_percent
- Default Warning Threshold: 80
- Default Critical Threshold: 80
- Description: Host Physical Memory Used Percentage

 The 'Monitor' and 'Graph' checkboxes are checked, and 'Health Check' is unchecked.

In this example, the Metric name is a computed Cloudera expression. The expression includes two Cloudera metrics: *physical_memory_used* and *physical_memory_total*. The expression takes the memory used metric, divides it by the total memory metric and multiplies that by 100 to return a computed metric named *memory_usage_percent*. The AS keyword is required. It defines an alias for the expression to uniquely name the metric:

```
(physical_memory_used / physical_memory_total) * 100 as memory_usage_percent
```



When working with computed metrics, make sure to include the AS clause (alias) in your computed expression. Aliases are required on computed metrics. Additionally, the metric Display name must match the alias.

The Metric Format String is an optional C-style formatting string. Here we limit the floating point number to 2 decimal places, and then append a percent sign to the computed metric value:

```
%.2f%%
```

See the section below on Example Formatting for more examples.

As you type into the Metric Name field, the valid names of metrics available are automatically auto-suggested. This ensures that you use a valid Cloudera metric name in your expression.

Health Check Metrics

Health Check metrics are a special type of metric that only report back Health Check status.

These metrics do not have numeric values, but instead have health check statuses that map to GroundWork statuses.

Figure: Health check metrics



Health Check metrics are flagged with the Health Check checkbox. As you type into the Metric Name field, the valid names of Health Check metrics available are automatically auto-suggested. This ensures that you use a valid Cloudera Health Check metric.

See the section below on Health Check Status Mappings for the complete list of health check status mappings.

Configuration Metrics

Configuration metrics are only used in synthetic computations. They are not reported back to the GroundWork server. To create a configuration metric, simply do not check the Monitor checkbox.

Configuration metrics are used in synthetic calculations, where the value is required, for example to perform a to megabyte or to gigabyte conversion, but you do not want to report back the byte value to the GroundWork server.

Figure: Configuration metrics



The Monitor check box is left unchecked. Note that we still use thresholds, as they are useful in the Synthetic Expression evaluator.

Synthetic Metrics

A Synthetic metric is a metric that is computed by Cloud Hub. It has one additional field, expression, that normal metrics do not have.

Figure: Synthetic metrics

The synthetic metric name is a simple metric name conforming to the GroundWork service name requirements. No spaces are allowed. By convention, we name synthetic metrics with the prefix *syn_*.

The Metric expression field contains the synthetic expression. In this example, we use a GroundWork function, *GW:GB2* to convert the value of the *physical_memory_used* Cloudera metric, a value in bytes, to a gigabyte value:

```
GW:GB2(physical_memory_used)
```

The Metric Format String is an optional C-style formatting string. Here we limit the floating-point number to 2 decimal places:

```
%.2f
```

See the section below on Example Formatting for more examples.

As you type into the Metric Expression field, the valid names of metrics available are automatically auto-suggested. This ensures that you use a valid Cloudera metric name in your expression. Synthetic expressions are limited to the Normal metrics defined for the current group. Additionally, the auto-suggest feature displays all GroundWork functions.

The Synthetic Expression

This field contains an actual programmable expression that is parsed by Cloud Hub. The expression is made up of:

- Normal Metrics (not health checks)
- Expression Operations (addition, subtraction, multiplication, division, parenthesis for grouping)
- GroundWork Functions
- Math Functions

Example expression with division and multiplier operators, parenthesis for grouping, and data conversion of integers to double values. The two normal Host metrics are *fd_open* and *fd_max*. Note that both normal metrics must be defined for this group. Other synthetic metrics cannot be included in a synthetic expression.

```
(GW:toDouble(fd_open) / GW:toDouble(fd_max)) * 100.0
```

The data types of Cloudera Metrics are typically floating point (double) values for any measurements. For counters, like the example above, are usually integers or longs. Consult the Cloud Hub documentation for a complete reference guide to metrics.

https://www.cloudera.com/documentation/enterprise/5-5-x/topics/cm_metrics.html

Type conversion is supported as GroundWork functions for both floating point and integer numbers. See the section below GroundWork Functions for a complete list.

The Expression Evaluator

The synthetics dialog has an expression evaluator to try out and test your expressions before saving them. The evaluator is displayed at the bottom of the dialog. Each variable in the expression is evaluated based on the check boxes.

Given the expression:
 $(GW:toDouble(fd_open) / GW:toDouble(fd_max)) * 100.0$

there are two variables, *fd_open* and *fd_max*. These variables are displayed in the Input Metric Values section of the dialog. There are three ways to evaluate the expression based on:

- Warning Threshold – Select the *Warning Threshold* option then click *Evaluate*
- Critical Threshold – Select the *Critical Threshold* option then click *Evaluate*

- **Override** – Enter values into the *Override Value* fields, then click *Evaluate*

Figure: Warning Threshold



The `fd_max` and `fd_open` metric fields are predefined with warning threshold values of 800 and 400. Clicking *Evaluate* yields the formatted output: 50.00% used.

Figure: Critical Threshold



The `fd_max` and `fd_open` metric fields are predefined with critical threshold values of 1000 and 600. Clicking *Evaluate* yields the formatted output: 60.00% used

Figure: Override Values



The `fd_max` and `fd_open` metric fields are entered with values of 2000 and 1500. Clicking *Evaluate* yields the formatted output: 75.00% used.

Cloudera Computed Examples

Cloudera computed (normal) metrics are calculated in the Cloudera server. Here are some examples of Cloudera computed metrics.

Example 1: Cloudera computes the memory usage of physical memory of a host

```
Computed Host Metric:
(physical_memory_used / physical_memory_total) * 100 as memory_usage_percent

Format:
%.2f%%
Display Name:
memory_usage_percentage
Warning Threshold: 85
Critical Threshold: 95
Description:
Host Physical Memory Used Percentage
```

Example 2: Cloudera converts bytes to MB for a host metric

```
Computed Host Metric:
physical_memory_used / 1048576 as memory_used_mb

Display Name:
memory_used_mb
Warning Threshold: 8182
Critical Threshold: 10240
Description:
Host Physical Memory Used in Megabytes
```

Example 3: Cloudera calculates Host CPU Load Percentage over 1 minute

```
Computed Host Metric:
cpu_user_rate / getHostFact(numCores, 1) * 100 as cpu_rate_user

Display Name:
cpu_rate_user
Warning Threshold: 75
Critical Threshold: 90
Description:
Host CPU Load Percentage over 1 Minute
```

Note that Cloudera currently has the following functions that can be used in a metric computation:

`dt(metric)` - Derivative with negative values.

The change of the underlying metric expression, per second.

Example:

```
dt(jvm_gc_count)
```

`dt0(metric)` - Derivative where negative values are skipped (useful for dealing with counter resets). The change of the underlying metric expression, per second.

Example:

```
dt0(jvm_gc_time_ms) / 10
```

`getHostFact(string factName, double defaultValue)` - Retrieves a fact about a host.

Example:

```
dt(total_cpu_user) / getHostFact(numCores, 2)
```

This example divides the results of `dt(total_cpu_user)` by the current number of cores for each host. If the number of cores cannot be determined, the default "2" will be used.

`getHostFact` currently supports one fact, `numCores`.

Synthetics examples

Example 1: Cloud Hub computes the physical memory used from bytes to GB with GW function

```
Metric Name:
syn_gb_memory_used
Expression:
GW:GB2(physical_memory_used)

Format:
%.2f% GB
Warning Threshold: 8
Critical Threshold: 10
Description:
Host Memory Used in GB
```

Example 2: Cloud Hub computes the physical memory used from bytes to GB with GW functions to convert integer values to double values

Metric Name:
 syn_fd_usage
Expression:
 $(GW:toDouble(fd_open) / GW:toDouble(fd_max)) * 100.0$
Format:
 %.2f%%
Warning Threshold: 700
Critical Threshold: 1000
Description:
 Percentage of File Descriptors Used

Example 3: Cloud Hub computes the percentage of memory used with the divideToPercentage function. Note this function returns an integer.

Metric Name:
 syn_physical_mem_percent
Expression:
 $GW:divideToPercentage(physical_memory_used, physical_memory_total)$
Format:
 %d %% used
Description:
 Percentage of Host Memory Used

GroundWork Functions

Table: Byte Conversion Functions Using Strict Hexadecimal Values (1024..)

GW:KB(bytes)	Convert bytes to kilobytes
GW:MB(bytes)	Convert bytes to megabytes
GW:GB(bytes)	Convert bytes to gigabytes
GW:TB(bytes)	Convert bytes to terabytes

Table: Byte Conversion Functions Using Decimal Values (1000..)

GW:KB2(bytes)	Convert bytes to kilobytes
GW:MB2(bytes)	Convert bytes to megabytes
GW:GB2(bytes)	Convert bytes to gigabytes
GW:TB2(bytes)	Convert bytes to terabytes

Table: Byte Conversion Functions Using Decimal Values (1000..)

GW:min(x,y)	Returns the minimum value of two numbers
GW:max(x,y)	Returns the maximum value of two numbers

Table: Type Conversion

GW:toDouble(m)	Converts a number to double precision
GW:toInteger	Converts a number to an integer
GW:toLong	Converts a number to a long integer

GW:scalePercentageUsed

This Function provides percentage usage synthetic values.

Calculates the usage percentage for a given used metric and a corresponding available metric.
Both the used metric and available metric can be scaled by corresponding scale factor parameters.

Example:

```
scalePercentageUsed(summary.quickStats.overallMemoryUsage,summary.hardware.memorySize, 1.0, 1.0)
```

Parameters:

used - Represents a 'used' metric value of how much of this resource has been used such as 'overallMemoryUsage'

available - Represents the totality of a resource, such as all memory available

usedScaleFactor - multiply usage parameter by this value, or pass in null to not scale. Passing in 1.0 will also not scale

availableScaleFactor - multiply available parameter by this value, or pass in null to not scale. Passing in 1.0 will also not scale

Returns the percentage usage as an integer

GW:scalePercentageUnused

This Function provides percentage unused/free synthetic values.

Calculates the unused(free) percentage for a given unused metric and a corresponding available metric.

Both the unused metric and available metric can be scaled by corresponding scale factor parameters.

Example:

```
scalePercentageUnused(summary.freeSpace,summary.capacity, 1.0, null, true)
```

Parameters:

unused - Represents a metric reference value of how much of this resource has not be used (free)

available - Represents the totality of a resource, such as all disk space available

usageScaleFactor - multiply usage parameter by this value, or pass in null to not scale. Passing in 1.0 will also not scale

availableScaleFactor - multiply available parameter by this value, or pass in null to not scale. Passing in 1.0 will also not scale

Returns the percentage not used (free) as an integer

GW:percentageUsed

This Function provides percentage usage synthetic values.

Calculates the usage percentage for a given used metric and a corresponding available metric.

Example:

```
scalePercentageUsed(summary.quickStats.overallMemoryUsage,summary.hardware.memorySize)
```

Parameters:

used - Represents a 'used' metric value of how much of this resource has been used such as 'overallMemoryUsage'

available - Represents the totality of a resource, such as all memory available

Returns the percentage usage as an integer

GW:percentageUnused

This Function provides percentage unused/free synthetic values.

Calculates the unused(free) percentage for a given unused metric and a corresponding available metric.

Both the unused metric and available metric can be scaled by corresponding scale factor parameters.

Example:

```
scalePercentageUnused(summary.freeSpace, summary.capacity)
```

Parameters:

unused - Represents a metric reference value of how much of this resource has not be used (free)

available - Represents the totality of a resource, such as all disk space available

Returns the percentage not used (free) as an integer

GW:divideToPercentage

Given two metrics, dividend and divisor divides them and returns a percentage ratio

Example:

```
GW:divideToPercentage(summary.quickStats.overallMemoryUsage,summary.hardware.memorySize)
```

Parameters:

dividend - typically a usage or free type metric

divisor - typically a totality type metric, such as total disk space

Returns the percentage ratio as an integer

GW:toPercentage

Turns a number such as .87 into an integer percentage (87). Also handles rounding of percentages

Example:

```
GW:toPercentage(summary.quickStats.overallMemoryUsage)
```

Parameters:

value - the value to be rounded to a full integer percentage

Returns the percentage value as an integer

Math Functions

Functions from Java Math library sample:

- min(n1,n2), max(n1,n2)
- abs
- cos, sin, tanexp, log, sort
- ceil, floor, round
- rint
- pow

See docs:

<https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html>

Math functions should be prefixed by:

Math:

Example:

```
Math:abs(metric)
```

Example Formatting

The formatting field uses standard C/Java style formatting strings. Typically, you will only be formatting one number, so the formatting strings should be very simple. Data types used are:

- Integer Numbers %d
- Floating Point Numbers %f

Example of formatting an integer value 2175:

| **Format String** | **Output** |

%d	2175
%05d	02175
%+5d	+2175
%,d	2,175
%d%% percent	2175% percent

Example of formatting a floating point value 3.141593:

Format String	Output
----------------------	---------------

DISABLED_HEALTH	WARNING
CONCERNING_HEALTH	WARNING
BAD_HEALTH	WARNING
GOOD_HEALTH	UP
STARTING	PENDING
STOPPING	DOWN
HISTORY_NOT_AVAILABLE	WARNING

Table: Host and Cloudera Service Status Mapping

Cloudera Host Status	Mapped to GroundWork Host Status
HISTORY_NOT_AVAILABLE	UNREACHABLE
NOT_AVAILABLE	UNREACHABLE
DISABLED	SUSPENDED
GOOD	UP
CONCERNING	WARNING
BAD	DOWN

Table: Metric Status Mapping

Cloudera Metric Status	Mapped to GroundWork Service Status
HISTORY_NOT_AVAILABLE	UNKNOWN
NOT_AVAILABLE	UNKNOWN
DISABLED	PENDING
GOOD	OK
CONCERNING	WARNING
BAD	CRITICAL