Data Collector Reference - Services

Cloud Insights

NetApp April 25, 2020

 $This\ PDF\ was\ generated\ from\ https://docs.netapp.com/us-en/cloudinsights/concept_telegraf_display_options.html\ on\ April\ 25,\ 2020.\ Always\ check\ docs.netapp.com\ for\ the\ latest.$



Table of Contents

Da	ata Collector Reference - Services	1
	Telegraf Integration Data	1
	Node Data Collection	1
	ActiveMQ Data Collector	4
	Apache Data Collector	7
	Consul Data Collector	. 11
	Couchbase Data Collector	. 12
	CouchDB Data Collector	. 14
	Docker Data Collector	. 17
	Elasticsearch Data Collector	. 25
	Flink Data Collector	. 30
	Hadoop Data Collector	. 37
	HAProxy Data Collector	. 47
	JVM Data Collector	. 55
	Kafka Data Collector	. 59
	Kibana Data Collector	
	Kubernetes Data Collector	. 66
	Memcached Data Collector	. 74
	MongoDB Data Collector.	. 79
	MySQL Data Collector	. 81
	Netstat Data Collector	
	Nginx Data Collector	
	PostgreSQL Data Collector	
	Puppet Agent Data Collector	. 92
	Padis Data Collector	0.5

Data Collector Reference - Services

Telegraf Integration Data

The Cloud Insights Telegraf data collectors are powerful tools for gathering sourcereported metrics from your data sources.

Cloud Insights provides different transform options for metrics collected using Telegraf. When adding these metrics to a widget, you are presented with a drop-down giving the transform choices:

None

Data is displayed as is, with no manipulation.

Rate

Current value divided by the time range since the previous observation.

Cumulative

The accumulation of the sum of previous values and the current value.

Delta

The difference between the previous observation value and the current value.

Delta rate

Delta value divided by the time range since the previous observation.

Cumulative Rate

Cumulative value divided by the time range since the previous observation.

Node Data Collection

Cloud Insights gathers metrics from the node on which you install an agent.

Installation

- 1. From **Admin > Data Collectors**, click **+Data Collector**. Under **Hosts**, choose a platform.
- 2. Follow the instructions to configure the agent. The instructions vary depending on the type of Operating System or Platform you are using to collect data.

Objects and Counters

The following objects and their counters are collected as Node metrics:

Object:	Identifiers:	Attributes:	Datapoints:
Node Filesystem	Node UUID Device Path Type	Node IP Node Name Node OS Mode	Free Inodes Free Inodes Total Inodes Used Total Used Total Used
Node Disk	Node UUID Disk	Node IP Node Name Node OS	IO Time Total IOPS In Progress Read Bytes (per sec) Read Time Total Reads (per sec) Weighted IO Time Total Write Bytes (per sec) Write Time Total Writes (per sec) Current Disk Queue Length Write Time Read Time IO Time
Node CPU	Node UUID CPU	Node IP Node Name Node OS	System CPU Usage User CPU Usage Idle CPU Usage Processor CPU Usage Interrupt CPU Usage DPC CPU Usage

Object:	Identifiers:	Attributes:	Datapoints:
Object: Node	Identifiers: Node UUID	Node IP Node Name Node OS	Kernel Boot Time Kernel Context Switches (per sec) Kernel Entropy Available Kernel Interrupts (per sec) Kernel Processes Forked (per sec) Memory Active Memory Available Total Memory Available Total Memory Buffered Memory Cached Memory Commit Limit Memory Committed As Memory Dirty Memory Free Memory High Free Memory High Total Memory Huge Page Size Memory Huge Pages Free Memory Huge Pages Total Memory Low Free Memory Low Total Memory Page Tables Memory Shared Memory Shared Memory Swap Cached Memory Swap Free Memory Swap Free Memory Used Total Memory Used Memory Used Memory Vmalloc Chunk Memory Vmalloc Chunk Memory Wired Memory Writeback Total

Object:	Identifiers:	Attributes:	Datapoints:
Node Network	Network Interface	Node Name	Bytes Received
	Node UUID	Node IP	Bytes Sent
		Node OS	Packets Outboud
			Discarded
			Packets Outboud Errors
			Packets Received
			Discarded
			Packets Received Errors
			Packets Received
			Packets Sent

Setup and Troubleshooting information can be found on the Configuring an Agent page.

MacOS Memory Usage

Cloud Insights (via Telegraf) and macOS report different numbers for memory usage. Both Telegraf and the Mac activity monitor use metrics gathered from *vm_stat*, however the total memory usage is calculated differently for each.

Telegraf calculates *Memory Used Total* as follows:

```
Memory Used Total = Memory Total - Memory Available Total
```

Where *Memory Available Total* is derived from the sum of "Pages free" and "Pages inactive" in *vm_stat*.

The Mac activity monitor, on the other hand, calculates Memory Used as follows:

```
Memory Used = App Memory + Wired Memory + Compressed
```

Where:

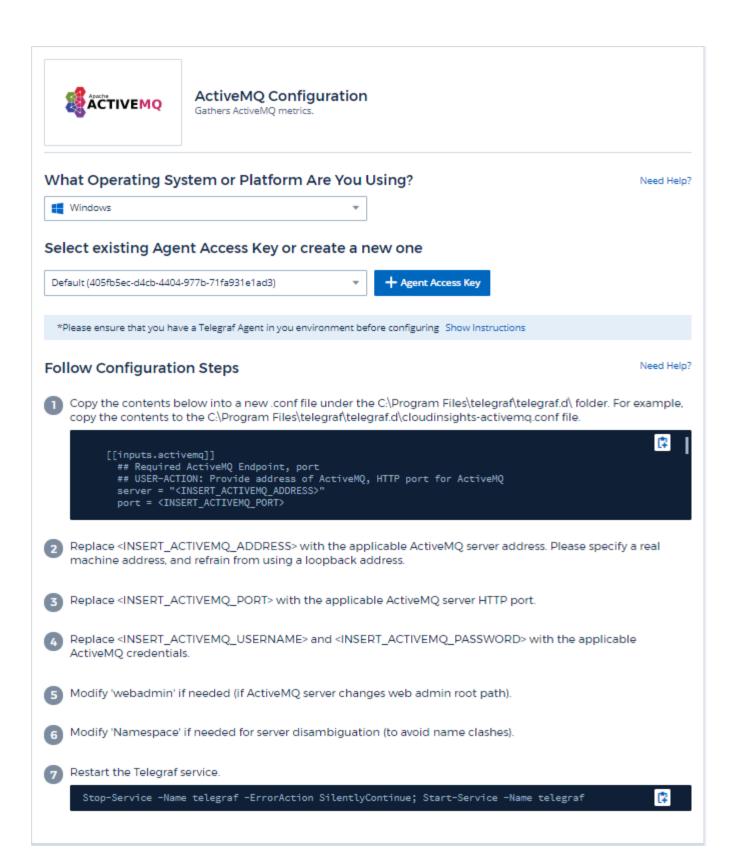
- *App Memory* is derived from the difference between "Anonymous pages" and "Pages purgeable" in *vm_stat*,
- Wired Memory is derived from "Pages wired down" in vm_stat, and
- *Compressed* is derived from "Pages occupied by compressor" in *vm_stat*.

ActiveMQ Data Collector

Cloud Insights uses this data collector to gather metrics from ActiveMQ.

Installation

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose ActiveMQ.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Information may be found in the ActiveMQ documentation

Objects and Counters

The following objects and their counters are collected:

Object:	Identifiers:	Attributes:	Datapoints:
ActiveMQ Queue	Namespace	Node Name	Consumer Count
	Queue	Node IP	Dequeue Count
	Port	Node UUID	Enqueue Count
	Server		Queue Size

Troubleshooting

Additional information may be found from the Support page.

Apache Data Collector

This data collector allows collection of data from Apache servers in your environment.

Pre-requisites

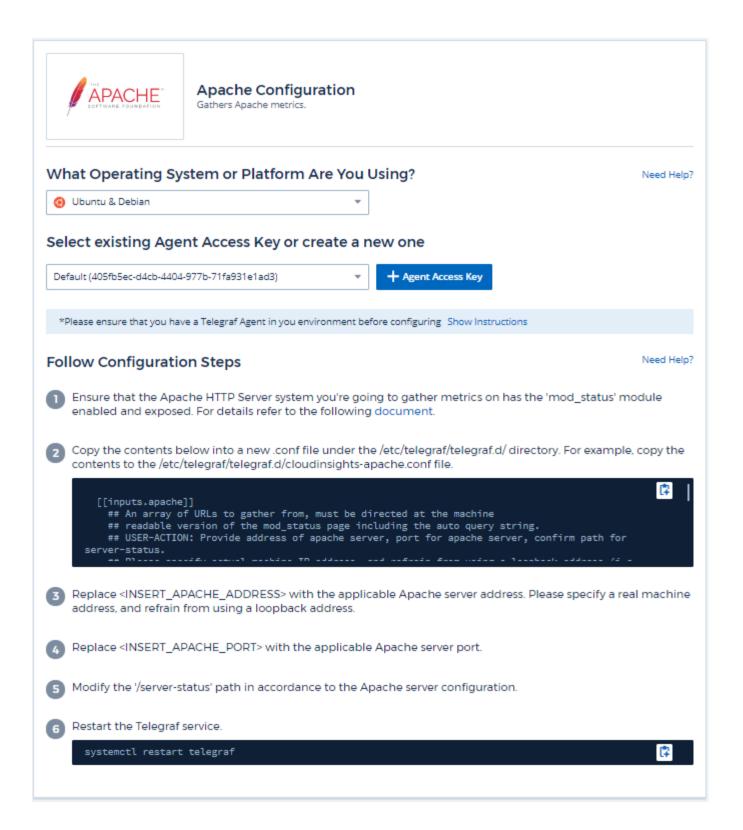
- You must have your Apache HTTP Server set up and properly running
- You must have sudo or administrator permissions on your agent host/VM
- Typically, the Apache *mod_status* module is configured to expose a page at the '/server-status?auto' location of the Apache server. The *ExtendedStatus* option must be enabled in order to collect all available fields. For information about how to configure your server, see the Apache module documentation: https://httpd.apache.org/docs/2.4/mod/mod_status.html#enable

Installation

1. From Admin > Data Collectors, click +Data Collector. Under Services, choose Apache.

Select the Operating System or Platform on which the Telegraf agent is installed.

- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Telegraf's plugin for Apache's HTTP Server relies on the 'mod_status' module to be enabled. When this is enabled, Apache's HTTP Server will expose an HTML endpoint that can be viewed on your browser or scraped for extraction of status of all Apache's HTTP Server configuration.

Compatibility:

Configuration was developed against Apache's HTTP Server version 2.4.38.

Enabling mod_status:

Enabling and exposing the 'mod_status' modules involves two steps:

- · Enabling module
- · Exposing stats from module

Enabling module:

The loading of modules is controlled by the config file under '/usr/local/apache/conf/httpd.conf'. Edit the config file and uncomment the following lines:

```
LoadModule status_module modules/mod_status.so

Include conf/extra/httpd-info.conf
```

Exposing stats from module:

The exposing of 'mod_status' is controlled by the config file under '/usr/local/apache2/conf/extra/httpd-info.conf'. Make sure you have the following in that configuration file (at least, other directives will be there):

```
# Allow server status reports generated by mod_status,
# with the URL of http://servername/server-status
<Location /server-status>
    SetHandler server-status
</Location>

# 
# ExtendedStatus controls whether Apache will generate "full" status
# information (ExtendedStatus On) or just basic information (ExtendedStatus
# Off) when the "server-status" handler is called. The default is Off.
# 
ExtendedStatus On
```

For detailed instructions on the 'mod_status' module, see the Apache documentation

Objects and Counters

Object:	Identifiers:	Attributes:	Datapoints:
Apache	Namespace	Node IP	Busy Workers
	Server	Node Name	Bytes per Request
		Port	Bytes per Second
		Parent Server Config	CPU Children System
		Generation	CPU Children User
		Parent Server MPM	CPU Load
		Generation	CPU System
		Server Uptime	CPU User
		Is Stopping	Asynchronous
			Connections Closing
			Asynchronous
			Connections Keep Alive
			Asynchronous
			Connections Writing
			Connections Total
			Duration per Request
			Idle Workers
			Load Average (last 1m)
			Load Average (last 15m)
			Load Average (last 5m)
			Processes
			Requests per Second
			Total Accesses
			Total Duration
			Total KBytes
			Scoreboard Closing
			Scoreboard DNS
			Lookups
			Scoreboard Finishing
			Scoreboard Idle Cleanup
			Scoreboard Keep Alive
			Scoreboard Logging
			Scoreboard Open
			Scoreboard Reading
			Scoreboard Sending
			Scoreboard Starting
			Scoreboard Waiting

Additional information may be found from the Support page.

Consul Data Collector

Cloud Insights uses this data collector to gather metrics from Consul.



This topic is considered Preview documentation and is subject to change.

Installation

1. From Admin > Data Collectors, click +Data Collector. Under Services, choose Consul.

If you haven't configured an Agent for collection, you are prompted to install an agent in your environment.

If you have an agent already configured, select the appropriate Operating System or Platform and click **Continue**.

2. Follow the instructions in the Consul Configuration screen to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.

Setup

Information may be found in the Consul documentation.

Objects and Counters for consul

The following objects and their counters are collected:

Object:	Identifiers:	Attributes:	Datapoints:
Consul	Namespace	Node IP	Critical
	Check ID	Node OS	Passing
	Service Node	Node UUID	Warning
		Node Name	
		Service Name	
		Check Name	
		Service ID	
		Status	

Troubleshooting

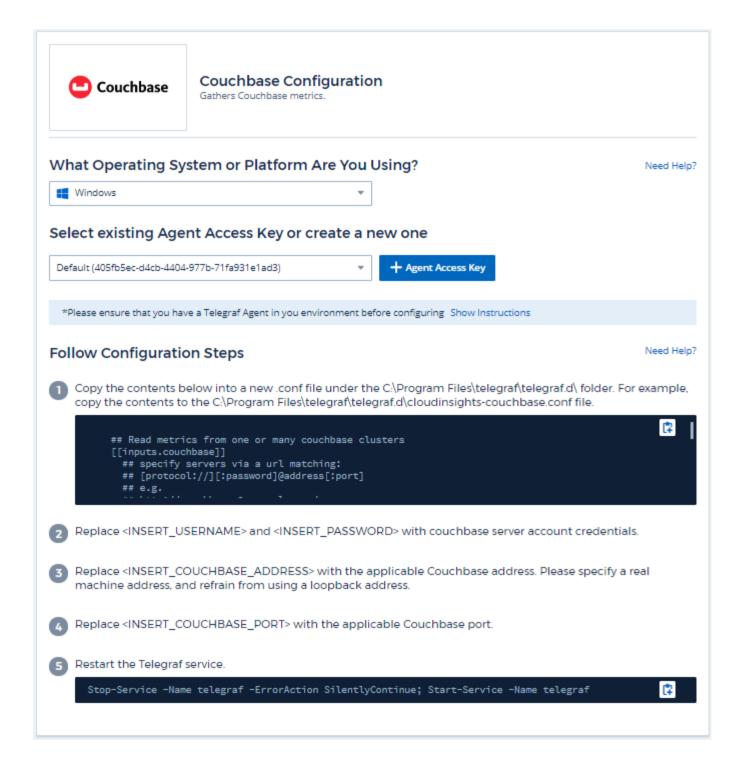
Additional information may be found from the Support page.

Couchbase Data Collector

Cloud Insights uses this data collector to gather metrics from Couchbase.

Installation

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose Couchbase.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Information may be found in the Couchbase documentation.

Objects and Counters

Object:	Identifiers:	Attributes:	Datapoints:
Couchbase Node	Namespace Cluster Couchbase Node Hostname	Node Name Node IP	Memory Free Memory Total
Couchbase Bucket	Namespace Bucket Cluster	Node Name Node IP	Data Used Data Fetches Disk Used Item Count Memory Used Operations Per Second Quota Used

Additional information may be found from the Support page.

CouchDB Data Collector

Cloud Insights uses this data collector to gather metrics from CouchDB.

Installation

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose CouchDB.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Information may be found in the CouchDB documentation.

Objects and Counters

Object:	Identifiers:	Attributes:	Datapoints:
CouchDB	Namespace	Node Name	Authentication Cache
	Server	Node IP	Hits
			Authentication Cache
			Miss
			Database Reads
			Database Writes
			Databases Open
			Open OS Files
			Max Request Time
			Min Request Time
			Httpd Request Methods
			Сору
			Httpd Request Methods
			Delete
			Httpd Request Methods
			Get
			Httpd Request Methods
			Head
			Httpd Request Methods
			Post
			Httpd Request Methods
			Put
			Status Codes 200
			Status Codes 201
			Status Codes 202
			Status Codes 301
			Status Codes 304
			Status Codes 400
			Status Codes 401
			Status Codes 403
			Status Codes 404
			Status Codes 405
			Status Codes 409
			Status Codes 412
			Status Codes 500

Additional information may be found from the Support page.

Docker Data Collector

Cloud Insights uses this data collector to gather metrics from Docker.



This topic is considered Preview documentation and is subject to change.

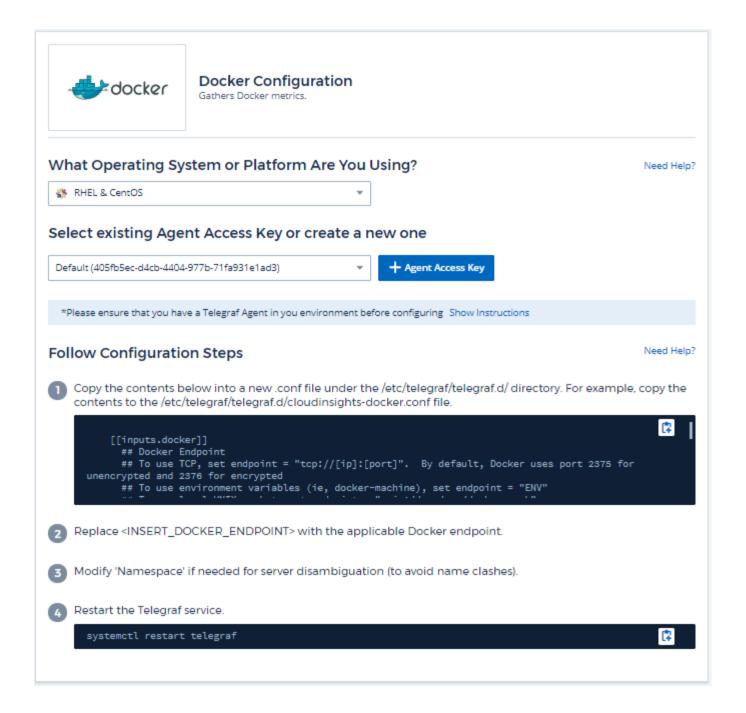
Installation

1. From Admin > Data Collectors, click +Data Collector. Under Services, choose Docker.

If you haven't configured an Agent for collection, you are prompted to install an agent in your environment.

If you have an agent already configured, select the appropriate Operating System or Platform and click **Continue**.

2. Follow the instructions in the Docker Configuration screen to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



The Telegraf input plugin for Docker collects metrics through a specified UNIX socket or a TCP endpoint.

Compatibility

Configuration was developed against Docker version 1.12.6.

Setting Up

Accessing Docker through a UNIX socket

If the Telegraf agent is running on baremetal, add the telegraf Unix user to the docker Unix group by running the following:

```
sudo usermod -aG docker telegraf
```

If the Telegraf agent is running within a Kubernetes pod, expose the Docker Unix socket by mapping the socket into the pod as a volume and then mounting that volume to /var/run/docker.sock. For example, add the following to the PodSpec:

```
volumes:
...
- name: docker-sock
hostPath:
path: /var/run/docker.sock
type: File
```

Then, add the following to the Container:

```
volumeMounts:
...
- name: docker-sock
mountPath: /var/run/docker.sock
```

Note that the Cloud Insights installer provided for the Kubernetes platform takes care of this mapping automatically.

Access Docker through a TCP endpoint

By default, Docker uses port 2375 for unencrypted access and port 2376 for encrypted access.

Objects and Counters

Object:	Identifiers:	Attributes:	Datapoints:
Docker Engine	Namespace	Node Name	Memory
	Docker Engine	Node IP	Containers
		Node UUID	Containers Paused
		Node OS	Containers Running
		Kubernetes Cluster	Containers Stopped
		Docker Version	CPUs
		Unit	Go Routines
			Images
			Listener Events
			Used File Descriptors
			Data Available
			Data Total
			Data Used
			Metadata Available
			Metadata Total
			Metadata Used
			Pool Blocksize

Object:	Identifiers:	Attributes:	Datapoints:
Docker Container	Namespace	Kubernetes Container	Memory Active
	Container Name	Hash	Anonymous
	Docker Engine	Kubernetes Container	Memory Active File
		Ports	Memory Cache
		Kubernetes Container	Memory Hierarchical
		Restart Count	Limit
		Kubernetes Container	Memory Inactive
		Termination Message	Anonymous
		Path	Memory Inactive File
		Kubernetes Container	Memory Limit
		Termination Message	Memory Mapped File
		Policy	Memory Max Usage
		Kubernetes Pod	Memory Page Fault
		Termination Grace	Memory Page Major
		Period	Fault
		Container Image	Memory Paged In
		Container Status	Memory Paged Out
		Container Version	Memory Resident Set
		Node Name	Size
		Kubernetes Container	Memory Resident Set
		Log Path	Size Huge
		Kubernetes Container	Memory Total Active
		Name	Anonymous
		Kubernetes Docker Type	Memory Total Active
		Kubernetes Pod Name	File
		Kubernetes Pod	Memory Total Cache
		Namespace	Memory Total Inactive
		Kubernetes Pod UID	Anonymous
		Kubernetes Sandbox ID	Memory Total Inactive
		Node IP	File
		Node UUID	Memory Total Mapped
		Docker Version	File
		Kubernetes IO Config	Memory Total Page
		Seen	Fault
		Kubernetes IO Config	Memory Total Page
		Source	Major Fault
		OpenShift IO SCC	Memory Total Paged In
		Kubernetes Description	Memory Total Paged Out
		Kubernetes Display	Memory Total Resident
		Name	Set Size
		OpenShift Tags	Memory Total Resident
		Kompose Service	Set Size Huge
		Pod Template Hash	Memory Total
		Controller Revision	Unevictable

Object:	Identifiers:	Attributes:	Datapoints:
Docker Container Block	Namespace	Kubernetes Container	IO Service Bytes
IO	Container Name	Hash	Recursive Async
	Device	Kubernetes Container	IO Service Bytes
	Docker Engine	Ports	Recursive Read
		Kubernetes Container	IO Service Bytes
		Restart Count	Recursive Sync
		Kubernetes Container	IO Service Bytes
		Termination Message	Recursive Total
		Path	IO Service Bytes
		Kubernetes Container	Recursive Write
		Termination Message	IO Serviced Recursive
		Policy	Async
		Kubernetes Pod	IO Serviced Recursive
		Termination Grace	Read
		Period	IO Serviced Recursive
		Container Image	Sync
		Container Status	IO Serviced Recursive
		Container Version	Total
		Node Name	IO Serviced Recursive
		Kubernetes Container	Write
		Log Path	
		Kubernetes Container	
		Name	
		Kubernetes Docker Type	
		Kubernetes Pod Name	
		Kubernetes Pod	
		Namespace	
		Kubernetes Pod UID	
		Kubernetes Sandbox ID	
		Node IP	
		Node UUID	
		Docker Version	
		Kubernetes Config Seen	
		Kubernetes Config	
		Source	
		OpenShift SCC	
		Kubernetes Description	
		Kubernetes Display	
		Name	
		OpenShift Tags	
		Schema Schema Version	
		Pod Template Hash	
		Controller Revision	
		Hash	

Object:	Identifiers:	Attributes:	Datapoints:
Docker Container	Namespace	Container Image	RX Dropped
Network	Container Name	Container Status	RX Bytes
	Network	Container Version	RX Errors
	Docker Engine	Node Name	RX Packets
		Node IP	TX Dropped
		Node UUID	TX Bytes
		Node OS	TX Errors
		K8s Cluster	TX Packets
		Docker Version	
		Container ID	

Object:	Identifiers:	Attributes:	Datapoints:
Docker Container CPU	Namespace	Kubernetes Container	Throttling Periods
	Container Name	Hash	Throttling Throttled
	CPU	Kubernetes Container	Periods
	Docker Engine	Ports	Throttling Throttled
		Kubernetes Container	Time
		Restart Count	Usage In Kernel Mode
		Kubernetes Container	Usage In User Mode
		Termination Message	Usage Percent
		Path	Usage System
		Kubernetes Container	Usage Total
		Termination Message	
		Policy	
		Kubernetes Pod	
		Termination Grace	
		Period	
		Kubernetes Config Seen	
		Kubernetes Config	
		Source	
		OpenShift SCC	
		Container Image	
		Container Status	
		Container Version	
		Node Name	
		Kubernetes Container	
		Log Path	
		Kubernetes Container	
		name	
		Kubernetes Docker Type	
		Kubernetes Pod Name	
		Kubernetes Pod	
		Namespace	
		Kubernetes Pod UID	
		Kubernetes Sandbox ID	
		Node IP	
		Node UUID	
		Node OS	
		Kubernetes Cluster	
		Docker Version	
		Kubernetes Description	
		Kubernetes Display	
		Name	
		OpenShift Tags	
		Schema Version	
		Pod Template Hash	

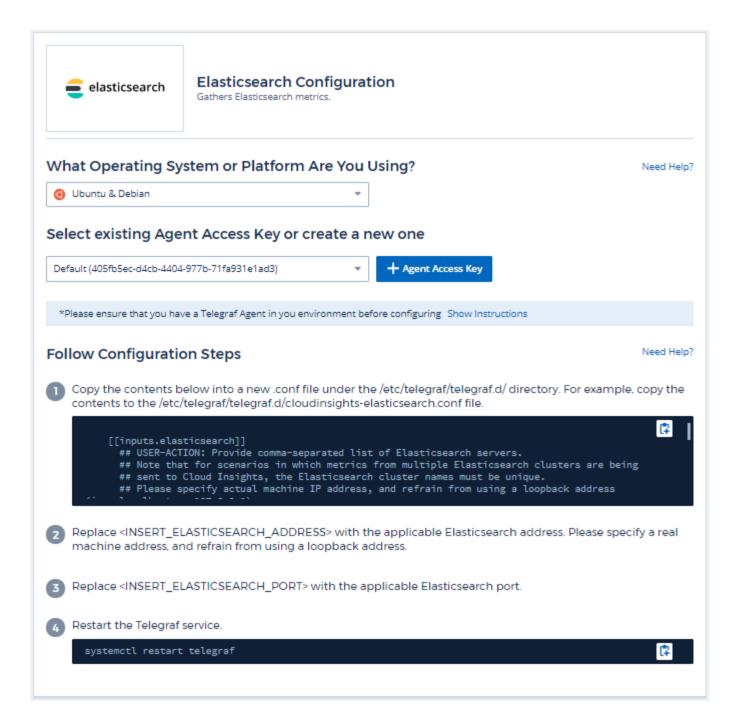
Problem:	Try this:
I do not see my Docker metrics in Cloud Insights after following the instructions on the configuration page.	Check the Telegraf agent logs to see if it reports the following error: E! Error in plugin [inputs.docker]: Got permission denied while trying to connect to the Docker daemon socket
	If it does, take the necessary steps to provide the Telegraf agent access to the Docker Unix socket as specified above.

Additional information may be found from the Support page.

Elasticsearch Data Collector

Cloud Insights uses this data collector to gather metrics from Elasticsearch.

- 1. From **Admin > Data Collectors**, click **+Data Collector**. Under **Services**, choose Elasticsearch.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Information may be found in the Elasticsearch documentation.

Objects and Counters

Object:	Identifiers:	Attributes:	Datapoints:
Elasticsearch Cluster	Namespace	Node IP	Master Node Count
	Cluster	Node Name	Total Node Count
		Cluster Status	Filesystem Data
			Available (bytes)
			Filesystem Data Free
			(bytes)
			Filesystem Data Total
			(bytes)
			JVM Threads
			OS Allocated
			Proccessors
			OS Available Processors
			OS Mem Free (bytes)
			OS Mem Free
			OS Mem Total (bytes)
			OS Mem Used (bytes)
			OS Mem Used
			Process CPU
			Indices Completion Size
			(bytes)
			Indices Count
			Indices Docs Count
			Indices Docs Count
			Indices Field Data
			Evictions
			Indices Field Data
			Memory Size (bytes)
			Indices Query Cache
			Count
			Indices Cache Size
			Indices Segments Count
			Indices Segments Doc
			Values Memory (bytes)
			Indices Shards Index
			Primaries Avg
			Indices Shards Index
			Primaries Max
			Indices Shards Index
			Primaries Min
			Indices Shards Index
			Replication Avg
			Indices Shards Index
			Replication Max
			Indices Shards Index

Object:	Identifiers:	Attributes:	Datapoints:
Elasticsearch Node	Namespace Cluster ES Node IP ES Node	Zone ID	Machine Learning Enabled Machine Learning Memory Machine Learning Max Open Jobs X-Pack Installed Breakers Accounting Estimated Size (bytes) Breakers Accounting Limit Size (bytes) Breakers Accounting Overhead Breakers Field Data Estimated Size (bytes) Breakers Field Data Limit Size (bytes) Breakers Field Data Limit Size (bytes) Breakers Field Data Coverhead Breakers Field Data Tripped Breakers Field Data Tripped Breakers In-Flight Sstimated Size (bytes) Breakers In-Flight Limit Size (bytes) Breakers In-Flight Overhead Breakers In-Flight Tripped Breakers In-Flight Tripped Breakers Parent Estimated Size (bytes) Breakers Parent Estimated Size (bytes) Breakers Parent Limit Size (bytes) Breakers Parent Overhead Breakers Parent Tripped Breakers Request Estimated Size (bytes)

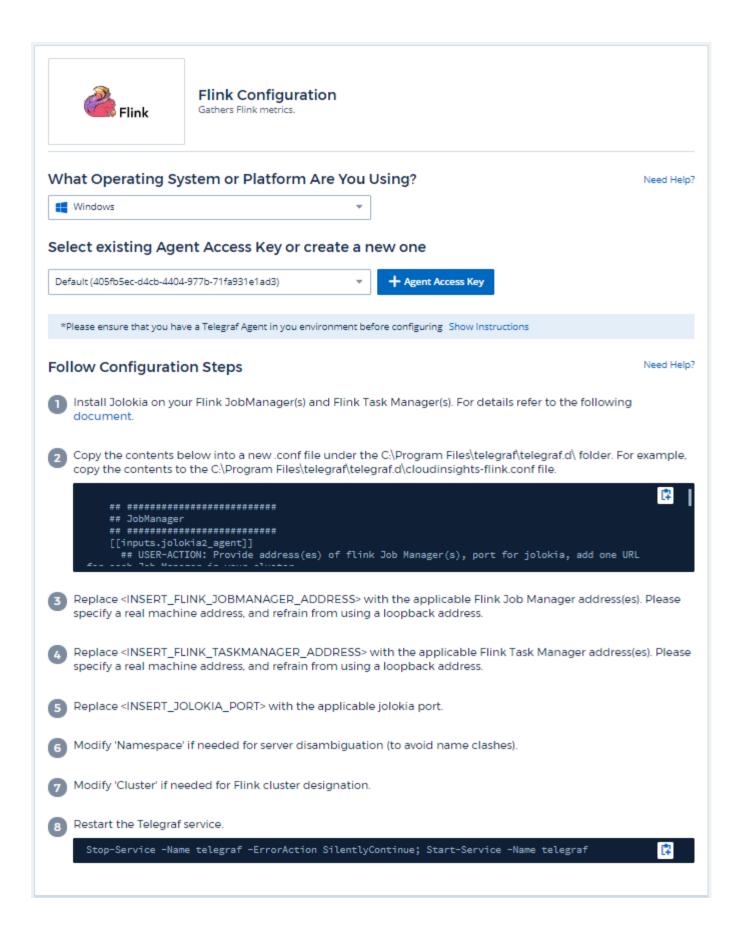
Additional information may be found from the Support page.

Flink Data Collector

Cloud Insights uses this data collector to gather metrics from Flink.

Installation

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose Flink.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



A full Flink deployment involves the following components:

JobManager: The Flink master. Coordinates a series of TaskManagers (slaves). In a High Availability setup, system will have more than one JobManager.

TaskManager: The Flink slaves. This is where Flink operators get executed.

The Flink plugin is based on the telegraf's Jolokia plugin. As such as a requirement to gather info from all Flink components, JMX needs to be configured and exposed via Jolokia on all components.

Compatibility

Configuration was developed against Flink version 1.7.0.

Setting Up

Jolokia Agent Jar

For all individual components, a version the Jolokia agent jar file must be downloaded. The version tested against was Jolokia agent 1.6.0.

Instructions below assume that downloaded jar file (jolokia-jvm-1.6.0-agent.jar) is placed under location '/opt/flink/lib/'.

JobManager

To configure JobManager to expose the Jolokia API, you can setup the following environment variable on your nodes then restart the JobManager:

```
export FLINK_ENV_JAVA_OPTS="-javaagent:/opt/flink/lib/jolokia-jvm-1.6.0-
agent.jar=port=8778,host=0.0.0.0"
```

You can choose a different port for Jolokia (8778). If you have an internal IP to lock Jolokia onto you can replace the "catch all" 0.0.0.0 by your own IP. Notice this IP needs to be accessible from the telegraf plugin.

TaskManager

To configure TaskManager(s) to expose the Jolokia API, you can setup the following environment variable on your nodes then restart the TaskManager:

```
export FLINK_ENV_JAVA_OPTS="-javaagent:/opt/flink/lib/jolokia-jvm-1.6.0-
agent.jar=port=8778,host=0.0.0.0"
```

You can choose a different port for Jolokia (8778). If you have an internal IP to lock Jolokia onto you can replace the "catch all" 0.0.0.0 by your own IP. Notice this IP needs to be accessible from the telegraf plugin.

Objects and Counters

Object:	Identifiers:	Attributes:	Datapoints:
Flink Task Manager	Cluster Namespace Server	Node Name Task Manager ID Node IP	Network Available Memory Segments Network Total Memory Segments Garbage Collection PS MarkSweep Count Garbage Collection PS MarkSweep Time Garbage Collection PS Scavenge Count Garbage Collection PS Scavenge Time Heap Memory Committed Heap Memory Init Heap Memory Used Thread Count Daemon Thread Count Peak Thread Count Thread Count Started
Flink Job	Cluster Namespace server Job ID	Node Name Job Name Node IP Last Checkpoint External Path Restarting Time	Downtime Full Restarts Last Checkpoint Alignment Buffered Last Checkpoint Duration Last Checkpoint Size Number of Completed Checkpoints Number of Failed Checkpoints Number of in Progress Checkpoints Number of Checkpoints Uptime

Object:	Identifiers:	Attributes:	Datapoints:
Flink Job Manager	Cluster	Node Name	Garbage Collection PS
	Namespace	Node IP	MarkSweep Count
	Server		Garbage Collection PS
			MarkSweep Time
			Garbage Collection PS
			Scavenge Count
			Garbage Collection PS
			Scavenge Time
			Heap Memory
			Committed
			Heap Memory Init
			Heap Memory Max
			Heap Memory Used
			Number Registered Task
			Managers
			Number Running Jobs
			Task Slots Available
			Task Slots Total
			Thread Count Daemon
			Thread Count Peak
			Thread Count
			Thread Count Total
			Started

Object:	Identifiers:	Attributes:	Datapoints:
Flink Task	Cluster Namespace Job ID Task ID	Server Node Name Job Name Sub Task Index Task Attempt ID Task Attempt Number Task Name Task Manager ID Node IP Current Input Watermark	Buffers In Pool Usage Buffers In Queue Length Buffers Out Pool Usage Buffers Out Queue Length Number Buffers In Local Number Buffers In Local Number Buffers In Local Per Second Count Number Buffers in Local Per Second Rate Number Buffers In Remote Number Buffers In Remote Per Second Count Number Buffers Out Number Buffers Out Per Second Count Number Buffers Out Per Second Count Number Buffers Out Per Second Count Number Bytes In Local Number Bytes In Local Per Second Count Number Bytes In Local Per Second Rate Number Bytes In Local Per Second Rate Number Bytes In Remote Number Bytes In Remote Number Bytes In Remote Per Second Count Number Bytes In Remote Per Second Count Number Bytes Out Number Bytes Out Number Bytes Out Per Second Count Number Bytes Out Per Second Count Number Bytes Out Per Second Count Number Records In Number Records In Number Records In Per Second Count Number Records In Per Second Count Number Records In Per Second Rate

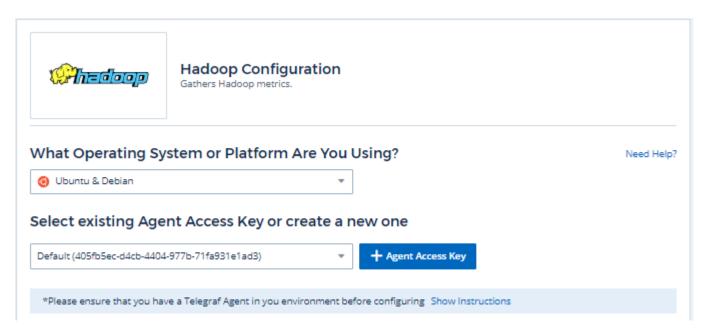
Object:	Identifiers:	Attributes:	Datapoints:
Flink Task Operator	Cluster	Server	Current Input
	Namespace	Node Name	Watermark
	Job ID	Job Name	Current Output
	Operator ID	Operator Name	Watermark
	Task ID	Sub Task Index	Number Records In
		Task Attempt ID	Number Records In Per
		Task Attempt Number	Second Count
		Task Name	Number Records In Per
		Task Manager ID	Second Rate
		Node IP	Number Records Out
			Number Records Out
			Per Second Count
			Number Records Out
			Per Second Rate
			Number Late Records
			Dropped
			Assigned Partitions
			Bytes Consumed Rate
			Commit Latency Avg
			Commit Latency Max
			Commit Rate
			Commits Failed
			Commits Succeeded
			Connection Close Rate
			Connection Count
			Connection Creation
			Rate
			Count
			Fetch Latency Avg
			Fetch Latency Max
			Fetch Rate
			Fetch Size Avg
			Fetch Size Max
			Fetch Throttle Time Av
			Fetch Throttle Time Ma
			Heartbeat Rate
			Incoming Byte Rate
			IO Ratio
			IO Time Avg (ns)
			IO Wait Ratio
			IO Wait Time Avg (ns)
			Join Rate
			Join Time Avg
			Last Heartbeat Ago

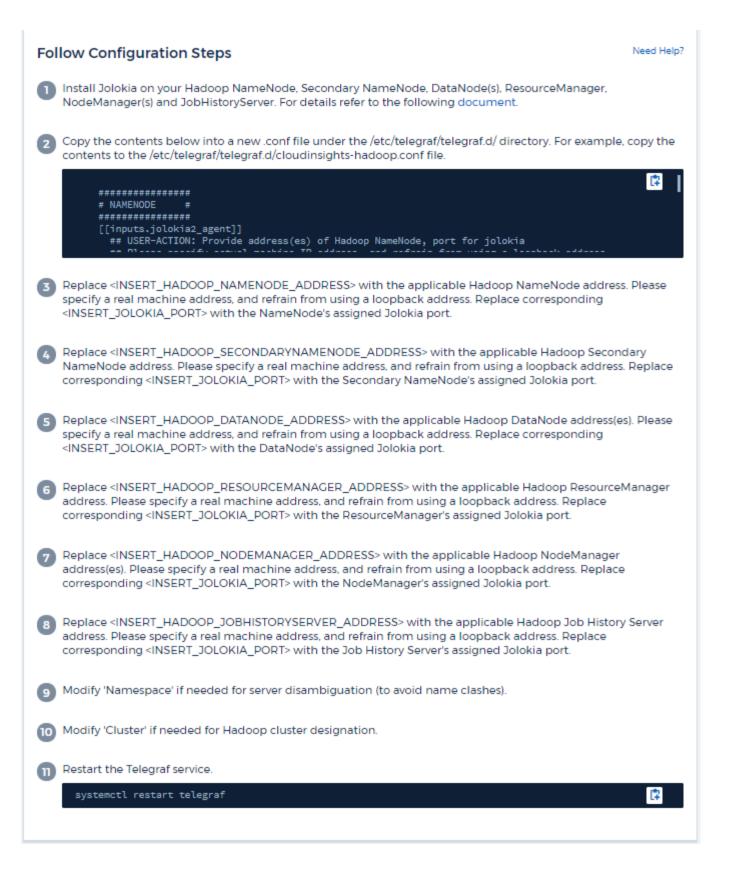
Additional information may be found from the Support page.

Hadoop Data Collector

Cloud Insights uses this data collector to gather metrics from Hadoop.

- 1. From **Admin > Data Collectors**, click **+Data Collector**. Under **Services**, choose Hadoop.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.





A full Hadoop deployment involves the following components:

• NameNode: The Hadoop Distributed File System (HDFS) master. Coordinates a series of DataNodes

(slaves).

- Secondary NameNode: a warm failover for the main NameNode. In Hadoop the promotion to NameNode does not occur automatically. Secondary NameNode gathers information from NameNode to be ready to be promoted when needed.
- DataNode: The HDFS slave. Actual owner for data.
- ResourceManager: The compute master (Yarn). Coordinates a series of NodeManagers (slaves).
- NodeManager: the resource for compute. Actual location for running of applications.
- JobHistoryServer: name says it all.

The Hadoop plugin is based on the telegraf's Jolokia plugin. As such as a requirement to gather info from all Hadoop components, JMX needs to be configured and exposed via Jolokia on all components.

Compatibility

Configuration was developed against Hadoop version 2.9.2.

Setting Up

Jolokia Agent Jar

For all individual components, a version the Jolokia agent jar file must be downloaded. The version tested against was Jolokia agent 1.6.0.

Instructions below assume that downloaded jar file (jolokia-jvm-1.6.0-agent.jar) is placed under location '/opt/hadoop/lib/'.

NameNode

To configure NameNode to expose the Jolokia API, you can setup the following in <HADOOP HOME>/etc/hadoop/hadoop-env.sh:

export HADOOP_NAMENODE_OPTS="\$HADOOP_NAMENODE_OPTS -javaagent:/opt/hadoop/lib/jolokia-jvm -1.6.0-agent.jar=port=7800,host=0.0.0.0 -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8000 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.password.file=\$HADOOP_HOME/conf/jmxremote.password"
You can choose a different port for JMX (8000 above) and Jolokia (7800). If you have an internal IP to lock Jolokia onto you can replace the "catch all" 0.0.0.0 by your own IP.
Notice this IP needs to be accessible from the telegraf plugin. You can use the option '-Dcom.sun.management.jmxremote.authenticate=false' if you don't want to authenticate. Use at your own risk.

Secondary NameNode

To configure the Secondary NameNode to expose the Jolokia API, you can setup the following in

export HADOOP_SECONDARYNAMENODE_OPTS="\$HADOOP_SECONDARYNAMENODE_OPTS

- -javaagent:/opt/hadoop/lib/jolokia-jvm-1.6.0-agent.jar=port=7802,host=0.0.0.0
- -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8002
- -Dcom.sun.management.jmxremote.ssl=false
- -Dcom.sun.management.jmxremote.password.file=\$HADOOP_HOME/conf/jmxremote.password"
 You can choose a different port for JMX (8002 above) and Jolokia (7802). If you have an internal IP to lock Jolokia onto you can replace the "catch all" 0.0.0.0 by your own IP.
 Notice this IP needs to be accessible from the telegraf plugin. You can use the option '-Dcom.sun.management.jmxremote.authenticate=false' if you don't want to authenticate. Use at your own risk.

DataNode

To configure the DataNodes to expose the Jolokia API, you can setup the following in <HADOOP_HOME>/etc/hadoop/hadoop-env.sh:

export HADOOP_DATANODE_OPTS="\$HADOOP_DATANODE_OPTS -javaagent:/opt/hadoop/lib/jolokia-jvm -1.6.0-agent.jar=port=7801,host=0.0.0.0 -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8001 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.password.file=\$HADOOP_HOME/conf/jmxremote.password"
You can choose a different port for JMX (8001 above) and Jolokia (7801). If you have an internal IP to lock Jolokia onto you can replace the "catch all" 0.0.0.0 by your own IP.
Notice this IP needs to be accessible from the telegraf plugin. You can use the option '-Dcom.sun.management.jmxremote.authenticate=false' if you don't want to authenticate. Use at your own risk.

ResourceManager

To configure the ResourceManager to expose the Jolokia API, you can setup the following in <HADOOP_HOME>/etc/hadoop/hadoop-env.sh:

export YARN_RESOURCEMANAGER_OPTS="\$YARN_RESOURCEMANAGER_OPTS
-javaagent:/opt/hadoop/lib/jolokia-jvm-1.6.0-agent.jar=port=7803,host=0.0.0.0
-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8003
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.password.file=\$HADOOP_HOME/conf/jmxremote.password"
You can choose a different port for JMX (8003 above) and Jolokia (7803). If you have an internal IP to lock Jolokia onto you can replace the "catch all" 0.0.0.0 by your own IP.
Notice this IP needs to be accessible from the telegraf plugin. You can use the option '-Dcom.sun.management.jmxremote.authenticate=false' if you don't want to authenticate. Use at your own risk.

NodeManager

To configure the NodeManagers to expose the Jolokia API, you can setup the following in <HADOOP_HOME>/etc/hadoop/hadoop-env.sh:

export YARN_NODEMANAGER_OPTS="\$YARN_NODEMANAGER_OPTS -javaagent:/opt/hadoop/lib/jolokia -jvm-1.6.0-agent.jar=port=7804,host=0.0.0.0 -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8004 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.password.file=\$HADOOP_HOME/conf/jmxremote.password"
You can choose a different port for JMX (8004 above) and Jolokia (7804). If you have an internal IP to lock Jolokia onto you can replace the "catch all" 0.0.0.0 by your own IP. Notice this IP needs to be accessible from the telegraf plugin. You can use the option '-Dcom.sun.management.jmxremote.authenticate=false' if you don't want to authenticate. Use at your own risk.

JobHistoryServer

To configure the JobHistoryServer to expose the Jolokia API, you can setup the following in <HADOOP_HOME>/etc/hadoop/hadoop-env.sh:

export HADOOP_JOB_HISTORYSERVER_OPTS="\$HADOOP_JOB_HISTORYSERVER_OPTS
-javaagent:/opt/hadoop/lib/jolokia-jvm-1.6.0-agent.jar=port=7805,host=0.0.0.0
-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8005
-Dcom.sun.management.jmxremote.password.file=\$HADOOP_HOME/conf/jmxremote.password"
You can choose a different port for JMX (8005 above) and Jolokia (7805). If you have an internal IP to lock Jolokia onto you can replace the "catch all" 0.0.0.0 by your own IP.
Notice this IP needs to be accessible from the telegraf plugin. You can use the option '-Dcom.sun.management.jmxremote.authenticate=false' if you don't want to authenticate. Use at your own risk.

Objects and Counters

Object:	Identifiers:	Attributes:	Datapoints:
Hadoop Secondary	Cluster	Node Name	GC Count
NameNode	Namespace	Node IP	GC Copies Count
	Server	Compile Info	GC Marks Sweep
		Version	Compact Count
			GC Number Info
			Threshold Exceeded
			GC Number Warning
			Threshold Exceeded
			GC Time
			GC Copy Time
			GC Marks Sweep
			Compact Time
			GC Total Extra Sleep
			Time
			Logs Error Count
			Logs Fatal Count
			Logs Info Count
			Logs Warn Count
			Memory Heap
			Committed
			Memory Heap Max
			Memory Heap Used
			Memory Max
			Memory Non Heap
			Committed
			Memory Non Heap Max
			Memory Non Heap Used
			Threads Blocked
			Threads New
			Threads Runnable
			Threads Terminated
			Threads Timed Waiting
			Threads Waiting

Object:	Identifiers:	Attributes:	Datapoints:
Hadoop NodeManager	Cluster Namespace Server	Node Name Node IP	Containers Allocated Memory Allocated Memory Allocated Oportunistic Virtual Cores Allocated Oportunistic Virtual Cores Allocated Memory Available Virtual Cores Available Virtual Cores Available Directories Bad Local Directories Bad Local Directories Bad Log Cache Size Before Clean Container Launch Duration Avg Time Container Launch Duration Number Of Operations Containers Completed Containers Failed Containers Failed Containers Reiniting Containers Reiniting Containers Reiniting Containers Rolled Back on Failure Containers Running Disk Utilization Good Local Directories Disk Utilization Good Local Directories Bytes Deleted Private Bytes Deleted Private Bytes Deleted Public Containers Running Opportunistic Bytes Deleted Total Shuffle Connections Shuffle Output Bytes Shuffle Outputs Failed Shuffle Outputs Ok GC Count GC Marks Sweep Compact Count GC Number Info

Object:	Identifiers:	Attributes:	Datapoints:
Hadoop ResourceManager	Identifiers: Cluster Namespace Server	Node Name Node IP	ApplicationMaster Launch Delay Avg ApplicationMaster Launch Delay Number ApplicationMaster Register Delay Avg ApplicationMaster Register Delay Number Register Delay Number NodeManager Active Number NodeManager Decomissioned Number NodeManager Decomissioning Number NodeManager Lost Number NodeManager Rebooted Number NodeManager Rebooted Number NodeManager Healthy Number NodeManager Healthy Number NodeManager Wemory Limit NodeManager Virtual Cores Limit Used Capacity Active Applications Active Users Aggregate Containers Allocated Aggregate Containers Preempted Aggregate Containers Preempted Aggregate Memory Seconds Preempted Aggregate Node Local Containers Allocated Aggregate Off Switch Containers Allocated Aggregate Ack Local Containers Allocated Aggregate Virtual Cores

Object:	Identifiers:	Attributes:	Datapoints:
Object: Hadoop DataNode	Identifiers: Cluster Namespace Server	Node Name Node IP Cluster ID Version	Transceiver Count Transmits in Progress Cache Capacity Cache Used Capacity DFS Used Estimated Capacity Lost Total Last Volume Failure Rate Blocks Number Cached Blocks Number Failed to Cache Blocks Number Failed to Uncache Volumes Number Failed to Uncache Volumes Number Failed Capacity Remaining GC Count GC Copies Count GC Marks Sweep Compact Count GC Number Info Threshold Exceeded GC Number Warning Threshold Exceeded GC Time GC Copy Time GC Marks Sweep Compact Time GC Total Extra Sleep Time Logs Error Count Logs Fatal Count Logs Warn Count Memory Heap Committed Memory Heap Max Memory Heap Used Memory Non Heap Committed Memory Non Heap Committed Memory Non Heap Max Memory Non Heap Max Memory Non Heap Max Memory Non Heap Max

Object:	Identifiers:	Attributes:	Datapoints:
Hadoop NameNode	Cluster Namespace Server	Node Name Node IP Transaction ID Last Written Time Since Last Loaded Edits HA State File System State Block Pool ID Cluster ID Compile Info Distinct Version Count Version	Block Capacity Blocks Total Capacity Used Capacity Used Non DFS Blocks Corrupt Estimated Capacity Lost Total Blocks Excess Heartbeats Expired Files Total File System Lock Queue Length Blocks Missing Blocks Missing Replication with Factor One Clients Active Data Nodes Decommissioning Dead Data Nodes Decommissioning Live Data Nodes Decommissioning Live Data Nodes Decommissioning Encryption Zones Number Data Nodes Entering Maintenance Files Under Construction Data Nodes Dead in Maintenance Files Under Construction Data Nodes Live in Maintenance Data Nodes Live in Maintenance Data Nodes Live in Maintenance Data Nodes Live Storages Stale Replication Pending Timeouts Data Node Message Pending Blocks Pending Deletion Blocks Pending Replication

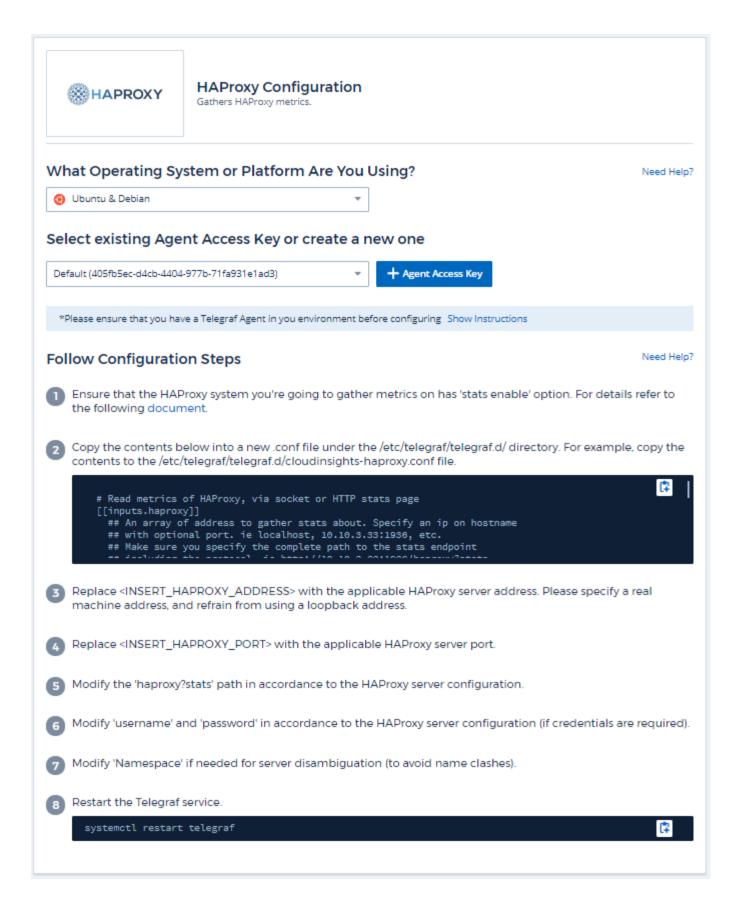
Object:	Identifiers:	Attributes:	Datapoints:
Hadoop	Cluster	Node Name	GC Count
JobHistoryServer	Namespace	Node IP	GC Copies Count
	Server		GC Marks Sweep
			Compact Count
			GC Number Info
			Threshold Exceeded
			GC Number Warning
			Threshold Exceeded
			GC Time
			GC Copy Time
			GC Marks Sweep
			Compact Time
			GC Total Extra Sleep
			Time
			Logs Error Count
			Logs Fatal Count
			Logs Info Count
			Logs Warn Count
			Memory Heap
			Committed
			Memory Heap Max
			Memory Heap Used
			Memory Max
			Memory Non Heap
			Committed
			Memory Non Heap Max
			Memory Non Heap Used
			Threads Blocked
			Threads New
			Threads Runnable
			Threads Terminated
			Threads Timed Waiting
			Threads Waiting

Additional information may be found from the Support page.

HAProxy Data Collector

Cloud Insights uses this data collector to gather metrics from HAProxy.

- 1. From **Admin > Data Collectors**, click **+Data Collector**. Under **Services**, choose HAProxy.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Telegraf's plugin for HAProxy relies on HAProxy Stats enablement. This is a configuration built into

HAProxy but it is not enabled out of the box. When enabled, HAProxy will expose an HTML endpoint that can be viewed on your browser or scraped for extraction of status of all HAProxy configurations.

Compatibility:

Configuration was developed against HAProxy version 1.9.4.

Setting Up:

To enable stats, edit your haproxy configuration file and add the the following lines after the 'defaults' section, using your own user/password and/or haproxy URL:

```
stats enable
stats auth myuser:mypassword
stats uri /haproxy?stats
```

The following is a simplified example configuration file with stats enabled:

```
global
  daemon
  maxconn 256
defaults
  mode http
  stats enable
  stats uri /haproxy?stats
  stats auth myuser:mypassword
  timeout connect 5000ms
  timeout client 50000ms
  timeout server 50000ms
frontend http-in
  bind *:80
  default_backend servers
frontend http-in9080
  bind *:9080
  default_backend servers_2
backend servers
  server server1 10.128.0.55:8080 check ssl verify none
  server server2 10.128.0.56:8080 check ssl verify none
backend servers_2
  server server3 10.128.0.57:8080 check ssl verify none
  server server4 10.128.0.58:8080 check ssl verify none
```

For complete and up to date instructions, see the HAProxy documentation.

Objects and Counters

Object:	Identifiers:	Attributes:	Datapoints:
HAProxy Frontend	Namespace	Node IP	Bytes In
_	Address	Node Name	Bytes Out
	Proxy	Proxy ID	Cache Hits
		Mode	Cache Lookups
		Process id	Compression Bytes
		Sessions Rate Limit	Bypassed
		Server id	Compression Bytes In
		Sessions Limit	Compression Bytes Out
		Status	Compression Responses
			Connection Rate
			Connection Rate Max
			Connections Total
			Requests Denied by
			Connection Rule
			Requests Denied by
			Security Concerns
			Responses Denied by
			Security Concerns
			Requests Denied by
			Session Rule
			Requests Errors
			Responses 1xx
			Responses 2xx
			Responses 3xx
			Responses 4xx
			Responses 5xx
			Responses Other
			Requests Intercepted
			Sessions Rate
			Sessions Rate Max
			Requests Rate
			Requests Rate Max
			Requests Total
			Sessions
			Sessions Max
			Sessions Total
			Requests Rewrites

Object:	Identifiers:	Attributes:	Datapoints:
HAProxy Server	Namespace	Node IP	Active Servers
	Address	Node Name	Backup Servers
	Proxy	Check Time to Finish	Bytes In
	Server	Check Fall Configuration	Bytes Out
		Check Health Value	Check Downs
		Check Rise	Check Fails
		Configuration	Client Aborts
		Check Status	Connections
		Proxy ID	Connection Average
		Last Change Time	Time
		Last Session Time	Downtime Total
		Mode	Denied Responses
		Process id	Connection Errors
		Server id	Response Errors
		Status	Responses 1xx
		Weight	Responses 2xx
			Responses 3xx
			Responses 4xx
			Responses 5xx
			Responses Other
			Server Selected Total
			Queue Current
			Queue Max
			Queue Average Time
			Sessions per Second
			Sessions per Second Max
			Connection Reuse
			Response Time Average
			Sessions
			Sessions Max
			Server Transfer Aborts
			Sessions Total
			Sessions Total Time
			Average
			Requests Redispatches
			Requests Retries
			Requests Rewrites
			Requests Rewilles

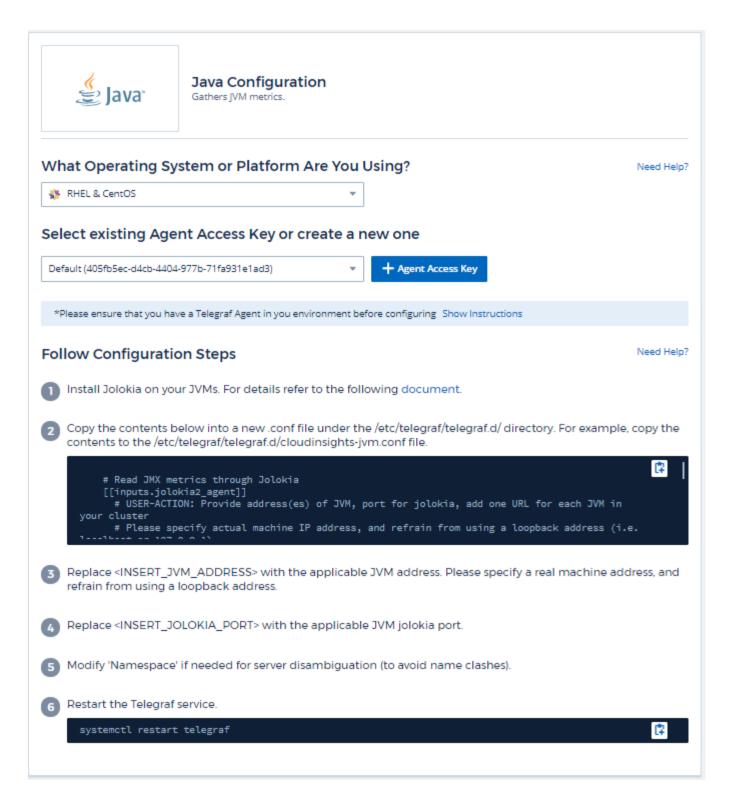
Object:	Identifiers:	Attributes:	Datapoints:
HAProxy Backend	Namespace	Node IP	Active Servers
	Address	Node Name	Backup Servers
	Proxy	Proxy ID	Bytes In
		Last Change Time	Bytes Out
		Last Session Time	Cache Hits
		Mode	Cache Lookups
		Process id	Check Downs
		Server id	Client Aborts
		Sessions Limit	Compression Bytes
		Status	Bypassed
		Weight	Compression Bytes In
			Compression Bytes Out
			Compression Responses
			Connections
			Connection Average
			Time
			Downtime Total
			Requests Denied by
			Security Concerns
			Responses Denied by
			Security Concerns
			Connection Errors
			Response Errors
			Responses 1xx
			Responses 2xx
			Responses 3xx
			Responses 4xx
			Responses 5xx
			Responses Other
			Server Selected Total
			Queue Current
			Queue Max
			Queue Average Time
			Sessions per Second
			Sessions per Second
			Max
			Requests Total
			Connection Reuse
			Response Time Average
			Sessions
			Sessions Max
			Server Transfer Aborts
			Sessions Total
			Sessions Total Time

Additional information may be found from the Support page.

JVM Data Collector

Cloud Insights uses this data collector to gather metrics from JVM.

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose JVM.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Information may be found in JVM documentation.

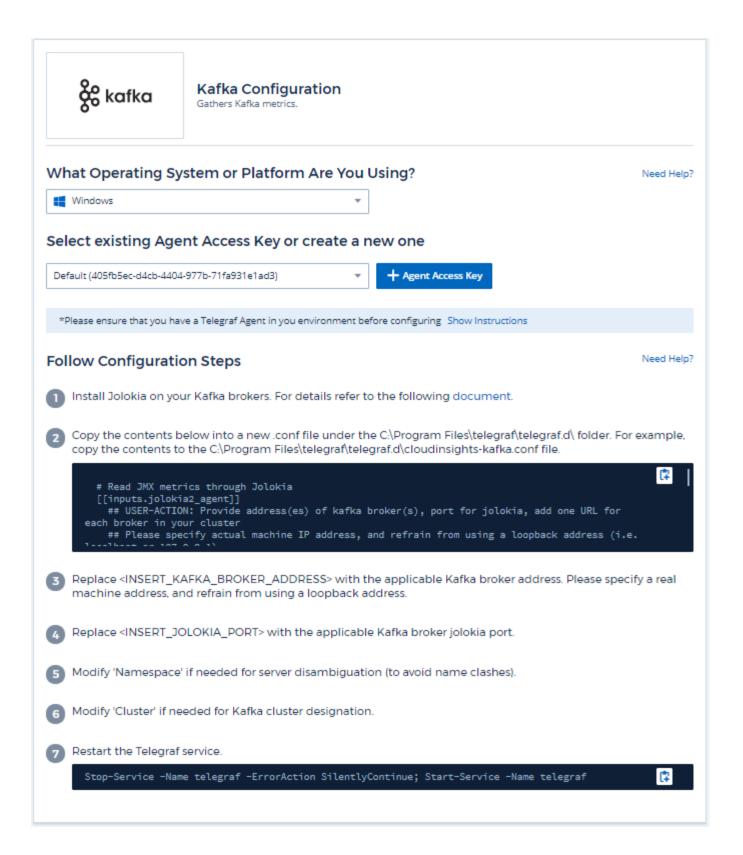
Objects and Counters

Additional information may be found from the Support page.

Kafka Data Collector

Cloud Insights uses this data collector to gather metrics from Kafka.

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose Kafka.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



The Kafka plugin is based on the telegraf's Jolokia plugin. As such as a requirement to gather info from all Kafka brokers, JMX needs to be configured and exposed via Jolokia on all components.

Compatibility

Configuration was developed against Kafka version 0.11.0.2.

Setting up

All the instructions below assume your install location for kafka is '/opt/kafka'. You can adapt instructions below to reflect your install location.

Jolokia Agent Jar

A version the Jolokia agent jar file must be downloaded. The version tested against was Jolokia agent 1.6.0.

Instructions below assume that the downloaded jar file (jolokia-jvm-1.6.0-agent.jar) is placed under the location '/opt/kafka/libs/'.

Kafka Brokers

To configure Kafka Brokers to expose the Jolokia API, you can add the following in <KAFKA_HOME>/bin/kafka-server-start.sh, just before the 'kafka-run-class.sh' call:

```
export JMX_PORT=9999
export RMI_HOSTNAME=`hostname -I`
export KAFKA_JMX_OPTS="-javaagent:/opt/kafka/libs/jolokia-jvm-1.6.0-
agent.jar=port=8778,host=0.0.0.0
-Dcom.sun.management.jmxremote.password.file=/opt/kafka/config/jmxremote.password
-Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=$RMI_HOSTNAME
-Dcom.sun.management.jmxremote.rmi.port=$JMX_PORT"
```

Note that example above is using 'hostname -I' to setup the 'RMI_HOSTNAME' environment variable. In multiple IP machines, this will need to be tweaked to gather the IP you care about for RMI connections.

You can choose a different port for JMX (9999 above) and Jolokia (8778). If you have an internal IP to lock Jolokia onto you can replace the "catch all" 0.0.0.0 by your own IP. Notice this IP needs to be accessible from the telegraf plugin. You can use the option '-Dcom.sun.management.jmxremote.authenticate=false' if you don't want to authenticate. Use at your own risk.

Objects and Counters

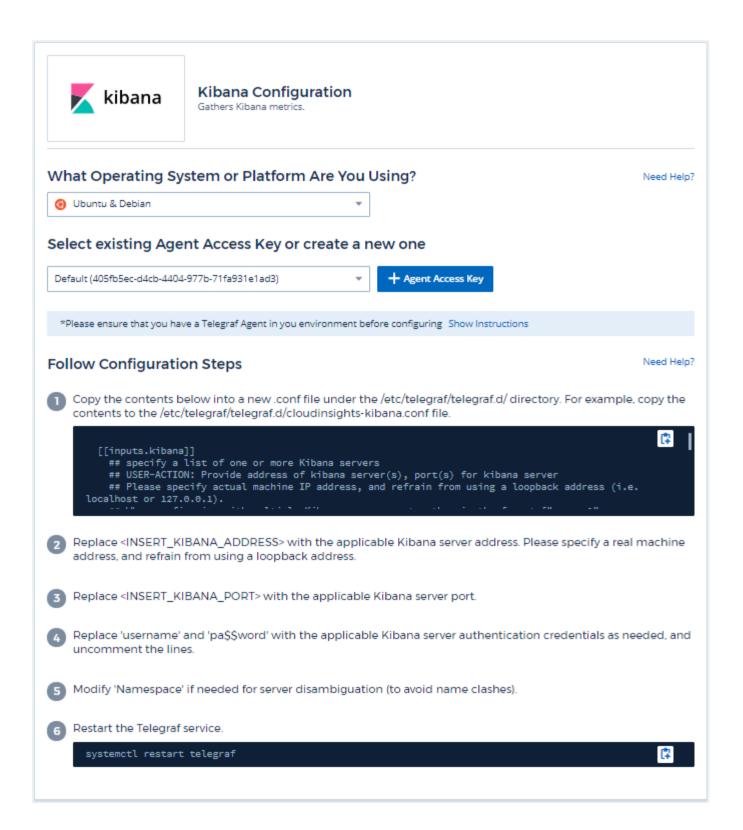
Object:	Identifiers:	Attributes:	Datapoints:
Kafka Broker	Cluster	Node Name	Replica Manager
	Namespace	Node IP	Fetcher Max Lag
	Broker		Zookeeper Client
			Connections
			Zookeeper Client
			Connections (15m rate)
			Zookeeper Client
			Connections (5m rate)
			Zookeeper Client
			Connections (mean rate)
			Zookeeper Client
			Connections (1m rate)
			Replica Manager
			Partition Count
			Thread Count Daemon
			Thread Count Peak
			Thread Count Current
			Thread Count Total
			Started
			Offline Partitions
			Produce Requests Total
			Time (50th Percentile)
			Produce Requests Total
			Time (75th Percentile)
			Produce Requests Total
			Time (95th Percentile)
			Produce Requests Total
			Time (98 Percentile)
			Produce Requests Total
			Time (999th Percentile)
			Produce Requests Total
			Time (99th Percentile)
			Produce Requests Total
			Time
			Produce Requests Total
			Time Max
			Produce Requests Total
			Time Mean
			Produce Requests Total
			Time Min
			Produce Requests Total
			Time Stddev
			Replica Manager ISR
			Shrinks

Additional information may be found from the Support page.

Kibana Data Collector

Cloud Insights uses this data collector to gather metrics from Kibana.

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose Kibana.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Information may be found in the Kibana documentation.

Objects and Counters

Object:	Identifiers:	Attributes:	Datapoints:
Kibana	Namespace Address	Node IP Node Name Version Status	Concurrent Connections Heap Max Heap Used Requests per Second Response Time Average Response Time Max Uptime

Additional information may be found from the Support page.

Kubernetes Data Collector

Cloud Insights uses this data collector to gather Kubernetes Pod, Node, and Container metrics.

Installation



It is strongly recommended to deploy the Telegraf agent as a DaemonSet and a ReplicaSet within the Kubernetes environment itself, by selecting *Kubernetes* as the platform during agent installation.

1. From Admin > Data Collectors, click +Data Collector. Under Services, choose Kubernetes.

If you haven't configured an Agent for collection, you are prompted to install an agent in your environment.

If you have an agent already configured, select the appropriate Operating System or Platform and click **Continue**.

2. Follow the instructions in the Kubernetes Configuration screen to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data. The example below shows the instructions for installing the Kubernetes data collector on the Kubernetes platform:

Follow Configuration Steps

Copy the contents below and append it to the data section of the telegraf-conf ConfigMap.



- Replace <INSERT_KUBELET_ADDRESS> with the "\$HOSTIP".
- Replace <INSERT_KUBELET_PORT> with the applicable kubelet port. The typical port is 10250.
- Replace <INSERT_K8S_CLUSTER_NAME> with the name of the Kubernetes cluster.
- If the Telegraf agent is running on one of the Kubernetes nodes, edit the telegraf-conf ConfigMap, replace <INSERT_K8S_CLUSTER_NAME> in the global_tags section with the name of the Kubernetes cluster, and uncomment the associated line.
- By default, the Telegraf agent will use the kubelet access token file found at /var/run/secrets/kubernetes.io/serviceaccount/token within each pod.
- If kube-state-metrics is installed and running within the Kubernetes cluster, copy the contents below and append it to the data section of the telegraf-conf-rs ConfigMap. Then, replace <INSERT_KUBE_STATE_METRICS_SERVER_IP> with the applicable kube-state-metrics server address, <INSERT_KUBE_STATE_METRICS_SERVER_PORT> with the applicable kube-state-metrics server port, and <INSERT_K8S_CLUSTER_NAME> with the name of the Kubernetes cluster.

Restart each applicable Telegraf pod.



Note that there are two different config maps that need updating: *telegraf-conf* for the DaemonSet and *telegraf-conf-rs* for the ReplicaSet. The latter is used to obtain kube-state-metrics data.

Setup

The Telegraf input plugin for Kubernetes collects metrics through the /stats/summary endpoint of the kubelet REST API as well as the kube-state-metrics server (if it exists).

Compatibility

Configuration was developed against Kubernetes version 1.9.1.

Configuring an Agent to Collect Data from Kubernetes

For Kubernetes environments, Cloud Insights deploys the Telegraf agent as a DaemonSet and a ReplicaSet. The pods in which the agents run need to have access to the following:

- hostPath
- configMap
- secrets

These Kubernetes objects are automatically created as part of the Kubernetes agent install command provided in the Cloud Insights UI. Some variants of Kubernetes, such as OpenShift, implement an added level of security that may block access to these components. The *SecurityContextConstraint* is not created as part of the Kubernetes agent install command provided in the Cloud Insights UI, and must be created manually. Once created, restart the Telegraf pod(s).

```
apiVersion: v1
   kind: SecurityContextConstraints
   metadata:
      name: telegraf-hostaccess
      creationTimestamp:
      annotations:
        kubernetes.io/description: telegraf-hostaccess allows hostpath volume mounts for
restricted SAs.
      labels:
        app: ci-telegraf
    priority: 10
   allowPrivilegedContainer: false
    defaultAddCapabilities: []
    requiredDropCapabilities: []
   allowedCapabilities: []
   allowedFlexVolumes: []
   allowHostDirVolumePlugin: true
    volumes:
    - hostPath
    - configMap
    - secret
    allowHostNetwork: false
    allowHostPorts: false
    allowHostPID: false
   allowHostIPC: false
   seLinuxContext:
      type: MustRunAs
   runAsUser:
      type: RunAsAny
   supplementalGroups:
      type: RunAsAny
   fsGroup:
      type: RunAsAny
   readOnlyRootFilesystem: false
    users:
    system:serviceaccount:monitoring:telegraf-user
    groups: []
```

Setting Up

For collecting Kubernetes metrics, the best practice is to deploy the Telegraf agent as a DaemonSet and a ReplicaSet within the Kubernetes environment of interest itself. The Cloud Insights agent install command does this if run on one of the Kubernetes nodes in the cluster. With the DaemonSet that is created, you can simply specify the "\$HOSTIP" environment variable for <INSERT_KUBELET_ADDRESS> in the telegraf-conf ConfigMap.

The pods in which the Telegraf agents run need to have access to a valid Kubernetes access token file in order to use the required kubelet API. To configure a Telegraf agent running outside the Kubernetes cluster of interest, you must manually generate this Kubernetes access token file, and configure the Telegraf agent to use this access token file.

To manually generate the Kubernetes access token, run the following in a Bash shell:

```
SECRET=$(sudo kubectl --namespace kube-system describe sa default |grep Tokens |awk '{print $2}')
```

```
TOKEN=\$(sudo kubectl --namespace kube-system describe secrets \$SECRET |grep ^token |awk '\$2\}')
```

To verify that the access token works as needed, run the following to confirm the kubelet API is accessible:

```
curl -v -X GET -H "Authorization: Bearer $TOKEN"
https://<KUBELET_ADDRESS>:<KUBELET_PORT>/stats/summary
```

To create the required access token file, run the following:

```
mkdir -p /var/run/secrets/kubernetes.io/serviceaccount/
```

```
echo -n $TOKEN | sudo tee /var/run/secrets/kubernetes.io/serviceaccount/token
```

By default, the Kubernetes input plugin configuration provided by CloudInsights is set up to look for the required access token file in the location used above. After performing the steps above, restart the Telegraf agent for the changes to take effect.

Objects and Counters

Object:	Identifiers:	Attributes:	Datapoints:
Kubernetes Container	Namespace Pod Container Cluster	Kubernetes Node Node Name Node OS Node UUID Node IP	CPU Nanoseconds CPU Usage Nanocores Memory Major Page Faults Memory Resident Set Size (RSS) Memory Working Set Memory Page Faults Memory Usage Root Filesystem Available Root Filesystem Capacity Root Filesystem Used
Kubernetes Node	Kubernetes Node Cluster	Node Name Node OS Node UUID Node IP	CPU Usage Nanocores CPU Usage Nanoseconds Filesystem Available Filesystem Total Filesystem Used Memory Available Memory Usage Memory Major Page Faults Memory Page Faults Memory Resident Set Size (RSS) Memory Working Set Network RX Errors (per sec) Network RX Bytes (per sec) Network TX Errors (per sec) Network TX Errors (per sec) Runtime Image Filesystem Available Runtime Image Filesystem Used Runtime Image Filesystem Capacity

Object:	Identifiers:	Attributes:	Datapoints:
Kubernetes Pod	Namespace Pod Cluster	Kubernetes Node Node Name Node IP Node OS Node UUID	Network TX Bytes (per sec) Network TX Errors (per sec) Network RX Bytes (per sec) Network RX Errors (per sec)
Kubernetes Pod Volume	Volume Pod Cluster Namespace	Kubernetes Node Node Name Node UUID Node IP Node OS	Available Capacity Used
Kubernetes System Container	System Container Kubernetes Node Cluster	Node Name Node IP Node OS Node UUID	CPU Usage Nanocores CPU Usage Core Nanoseconds Memory Major Page Faults Memory Page Faults Memory Resident Set Size (RSS) Memory Usage Memory Working Set Root Filesystem Available Root Filesystem Capacity Logs Filesystem Available Logs Filesystem Capacity

Installing the kube-state-metrics server

When you install the kube-state-metrics server you can enable collection of the following Kubernetes objects: StatefulSet, DaemonSet, Deployment, PV, PVC, ReplicaSet, Service, Namespace, Secret, ConfigMap, Pod Volume, and Ingress.

Use the following steps to install the kube-state-metrics server:

Steps

1. Create a temporary folder (for example, /tmp/kube-state-yaml-files/) and copy the .yaml files from

https://github.com/kubernetes/kube-state-metrics/tree/master/examples/standard to this folder.

2. Run the following command to apply the .yaml files needed for installing kube-state-metrics:

```
kubectl apply -f /tmp/kube-state-yaml-files/
```

kube-state-metrics Counters

Use the following links to access information for the kube state metrics counters:

- 1. Cronjob Metrics
- 2. DaemonSet Metrics
- 3. Deployment Metrics
- 4. Endpoint Metrics
- 5. Horizontal Pod Autoscaler Metrics
- 6. Ingress Metrics
- 7. Job Metrics
- 8. LimitRange Metrics
- 9. Namespace Metrics
- 10. Node Metrics
- 11. Persistent Volume Metrics
- 12. Persistant Volume Claim Metrics
- 13. Pod Metrics
- 14. Pod Disruption Budget Metrics
- 15. ReplicaSet metrics
- **16.** ReplicationController Metrics

Troubleshooting

Problem:	Try this:
I ran the Kubernetes agent installer command, but I do not see a Telegraf agent pod running via:	Check if there were any errors deploying the DaemonSet:
sudo kubectlnamespace monitoring get pods	sudo kubectlnamespace monitoring describe ds telegraf-ds
	If there are errors related to SecurityContextConstraints, do the following:
	1. Generate the Telegraf DaemonSet YAML
	sudo kubectlnamespace monitoring get ds telegraf-ds -o yaml > /tmp/telegraf-ds.yaml
	2. Stop the Telegraf service
	sudo kubectlnamespace monitoring delete ds telegraf-ds
	3. Create the necessary SecurityContextConstraint (see "Configuring Agent to Collect Data" section)
	4. Re-create the Telegraf DaemonSet
I configured Telegraf to obtain information about my Kubernetes cluster, but I don't see any information in Cloud Insights. I see "invalid header field value" errors in the Telegraf log file pertaining to the kubernetes input plugin I configured.	Ensure the referenced bearer_token file does not have a trailing newline. To verify, run the following command, and confirm that it returns 0: tail -c1 <bearer_token_file></bearer_token_file>

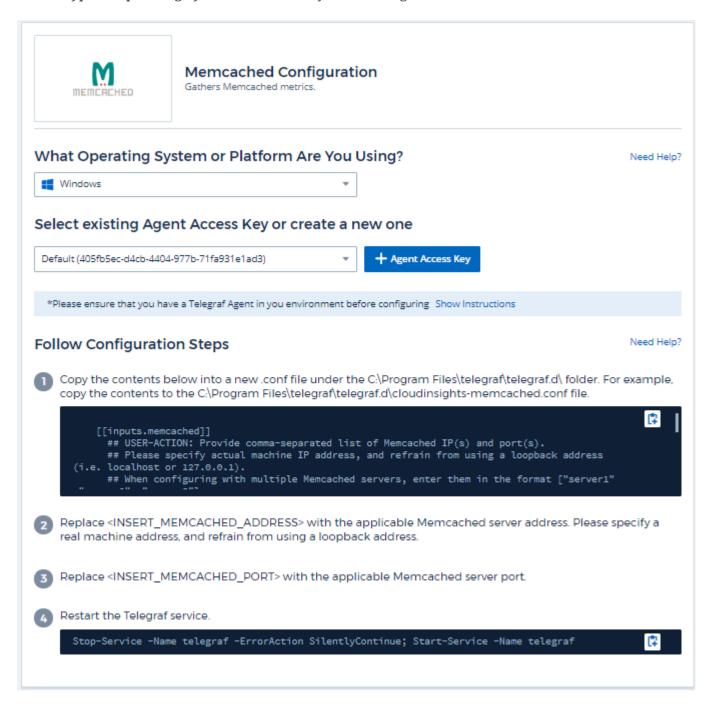
Additional information may be found from the Support page.

Memcached Data Collector

Cloud Insights uses this data collector to gather metrics from Memcached.

- From Admin > Data Collectors, click +Data Collector. Under Services, choose Memcached.
 Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a

- different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Information may be found in the Memcached wiki.

Objects and Counters

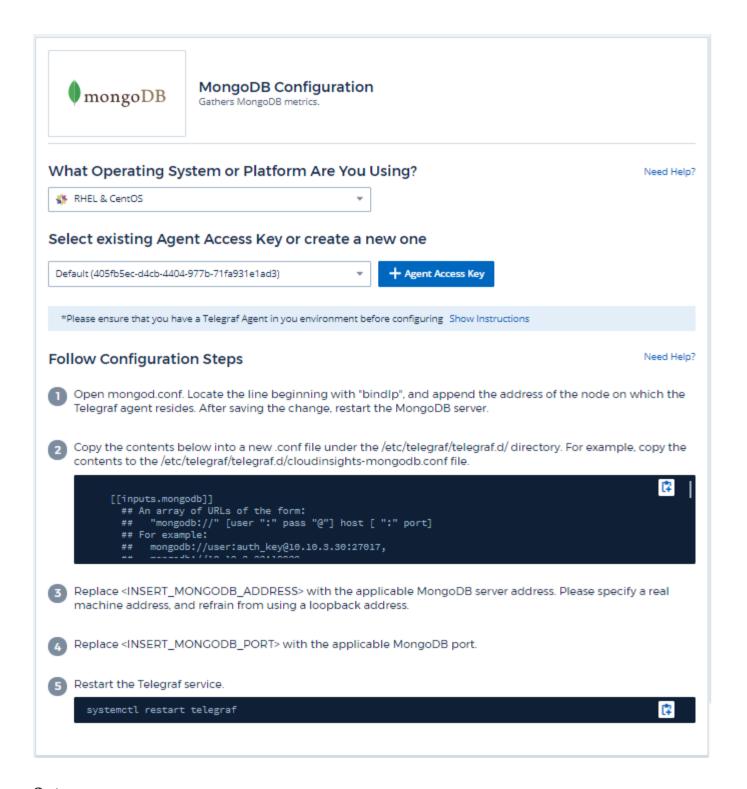
Object:	Identifiers:	Attributes:	Datapoints:
Memcached	Namespace	Node IP	Accepting Connections
	Server	Node Name	Handled Authentication
			Requests
			Failed Authentications
			Bytes Used
			Bytes Read (per sec)
			Bytes Written (per sec)
			CAS Badval
			CAS Hits
			CAS Misses
			Flush Reqs (per sec)
			Get Reqs (per sec)
			Set Reqs (per sec)
			Touch Reqs (per sec)
			Connection Yields (per
			sec)
			Connection Structures
			Open Connections
			Current Stored Items
			Decr Requests Hits (per
			sec)
			Decr Requests Misses
			(per sec)
			Delete Requests Hits
			(per sec)
			Delete Requests Misses
			(per sec)
			Items Evicted
			Valid Evictions
			Expired Items
			Get Hits (per sec)
			Get Misses (per sec)
			Used Hash Bytes
			Hash Is Expanding
			Hash Power Level
			Incr Requests Hits (per
			sec)
			Incr Requests Misses
			(per sec)
			Server Max Bytes
			Listen Disabled Num
			Reclaimed
			Worker Threads Count
			Total Opened

Additional information may be found from the Support page.

MongoDB Data Collector

Cloud Insights uses this data collector to gather metrics from MongoDB.

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose MongoDB.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Information may be found in the MongoDB documentation.

Objects and Counters

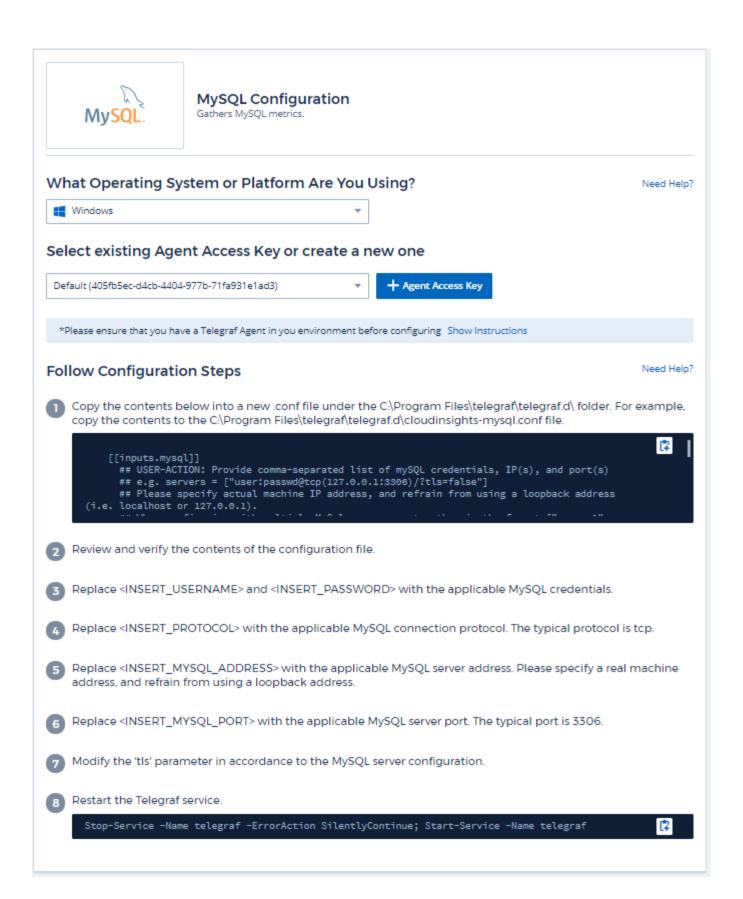
Object:	Identifiers:	Attributes:	Datapoints:
MongoDB	Namespace Hostname		
MongoDB Database	Namespace Hostname Database name		

Information may be found from the Support page.

MySQL Data Collector

Cloud Insights uses this data collector to gather metrics from MySQL.

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose MySQL.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Information may be found in the MySQL documentation.

Objects and Counters

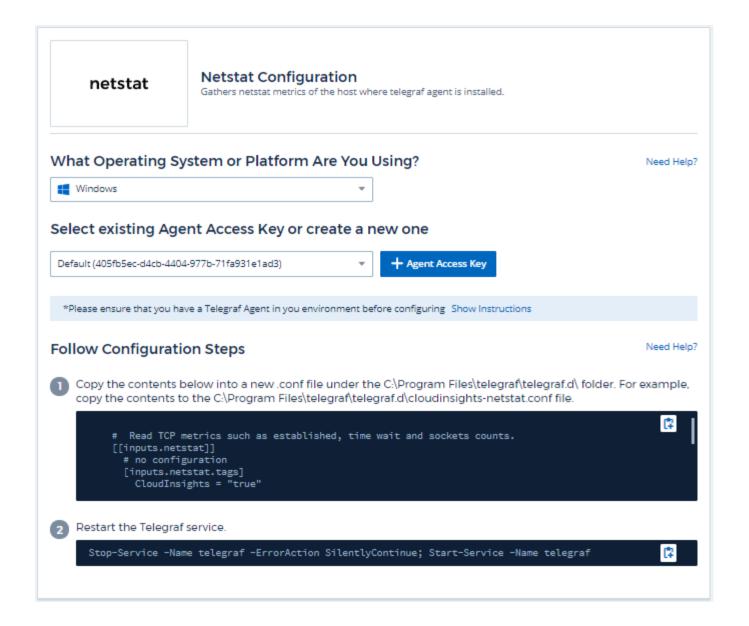
Object:	Identifiers:	Attributes:	Datapoints:
MySQL	Namespace	Node IP	Aborted Clients (per sec)
	MySQL Server	Node Name	Aborted Connects (per
			sec)
			RX Bytes (per sec)
			TX Bytes (per sec)
			Commands Admin (per
			sec)
			Commands Alter Event
			Commands Alter
			Function
			Commands Alter
			Instance
			Commands Alter
			Procedure
			Commands Alter Server
			Commands Alter Table
			Commands Alter
			Tablespace
			Commands Alter User
			Commands Analyze
			Commands Assign To
			Keycache
			Commands Begin
			Commands Binlog
			Commands Call
			Procedure
			Commands Change DB
			Commands Change
			Master
			Commands Change Repl
			Filter
			Commands Check
			Commands Checksum
			Commands Commit
			Commands Create DB
			Commands Create Event
			Commands Create
			Function Commands Create Index
			Commands Create Index
			Commands Create
			Procedure
			Commands Create
			Server
			Commands Create Table

Additional information may be found from the Support page.

Netstat Data Collector

Cloud Insights uses this data collector to gather Netstat metrics.

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose Netstat.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Objects and Counters

The following objects and their counters are collected:

Object:	Identifiers:	Attributes:	Datapoints:
Netstat	Node UUID	Node IP Node Name	

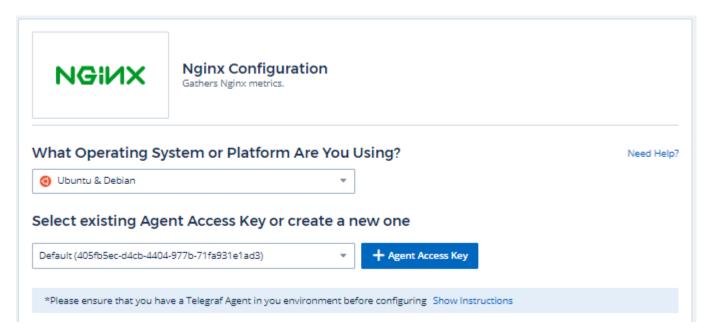
Troubleshooting

Additional information may be found from the Support page.

Nginx Data Collector

Cloud Insights uses this data collector to gather metrics from Nginx.

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose Nginx.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.





Nginx metric collection requires that Nginx http_stub_status_module be enabled.

Additional information may be found in the Nginx documentation.

Objects and Counters

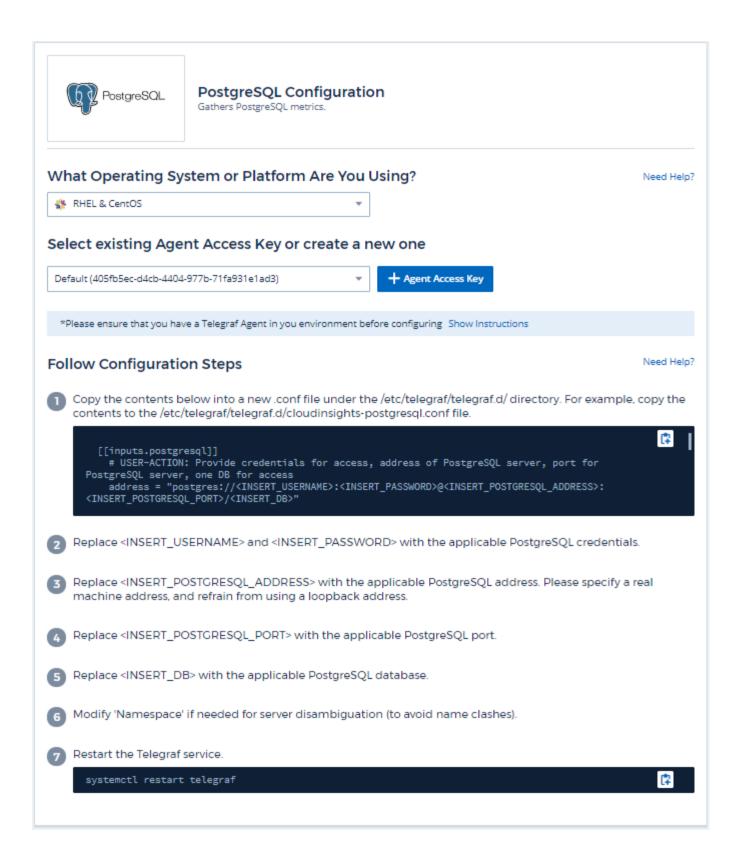
Object:	Identifiers:	Attributes:	Datapoints:
Nginx	Namespace Server	Node IP Node Name Port	Accepts Active Handled Reading Requests Waiting Writing

Additional information may be found from the Support page.

PostgreSQL Data Collector

Cloud Insights uses this data collector to gather metrics from PostgreSQL.

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose PostgreSQL.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Information may be found in the PostgreSQL documentation.

Objects and Counters

The following objects and their counters are collected:

Object:	Identifiers:	Attributes:	Datapoints:
PostgreSQL Server	Namespace Database Server	Node Name Node IP	Buffers Allocated Buffers Backend Buffers Backend File Sync Buffers Checkpoint Buffers Clean Checkpoints Sync Time Checkpoints Write Time Checkpoints Requests Checkpoints Timed Max Written Clean
PostgreSQL Database	Namespace Database Server	Database OID Node Name Node IP	Blocks Read Time Blocks Write Time Blocks Hits Blocks Reads Conflicts Deadlocks Client Number Temp Files Bytes Temp Files Number Rows Deleted Rows Fetched Rows Inserted Rows Returned Rows Updated Transactions Committed Transactions Rollbacked

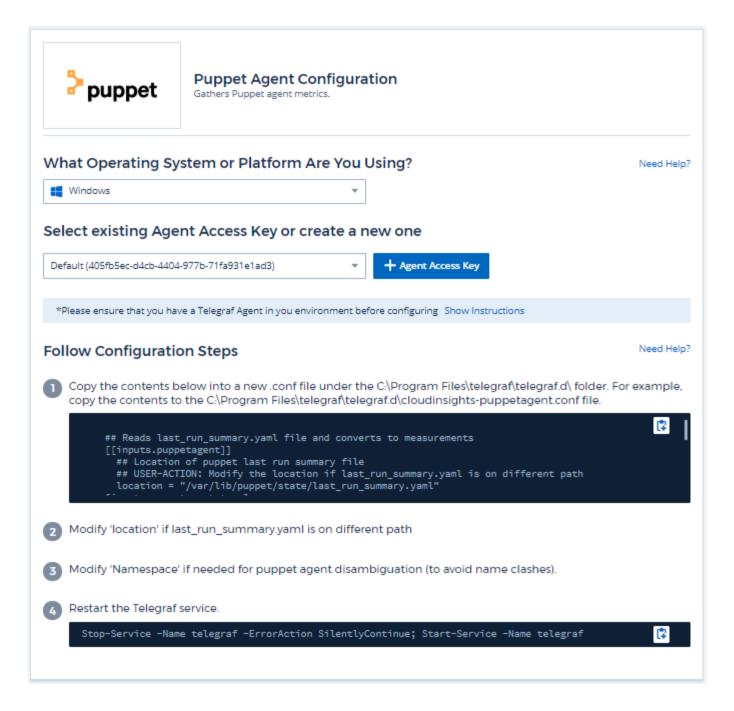
Troubleshooting

Additional information may be found from the Support page.

Puppet Agent Data Collector

Cloud Insights uses this data collector to gather metrics from Puppet Agent.

- 1. From Admin > Data Collectors, click +Data Collector. Under Services, choose Puppet.
 - Select the Operating System or Platform on which the Telegraf agent is installed.
- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Information may be found in the Puppet documentation

Objects and Counters

Object:	Identifiers:	Attributes:	Datapoints:

Puppet Agent	Namespace	Node Name	Changes Total
	Node UUID	Location	Events Failure
		Node IP	Events Success
		Version Configstring	Events Total
		Version Puppet	Resources Changed
			Resources Failed
			Resources Failed To
			Restart
			Resources Outofsync
			Resources Restarted
			Resources Scheduled
			Resources Skipped
			Resources Total
			Time Anchor
			Time Configretrieval
			Time Cron
			Time Exec
			Time File
			Time Filebucket
			Time Lastrun
			Time Package
			Time Schedule
			Time Service
			Time Sshauthorizedkey
			Time Total
			Time User

Additional information may be found from the Support page.

Redis Data Collector

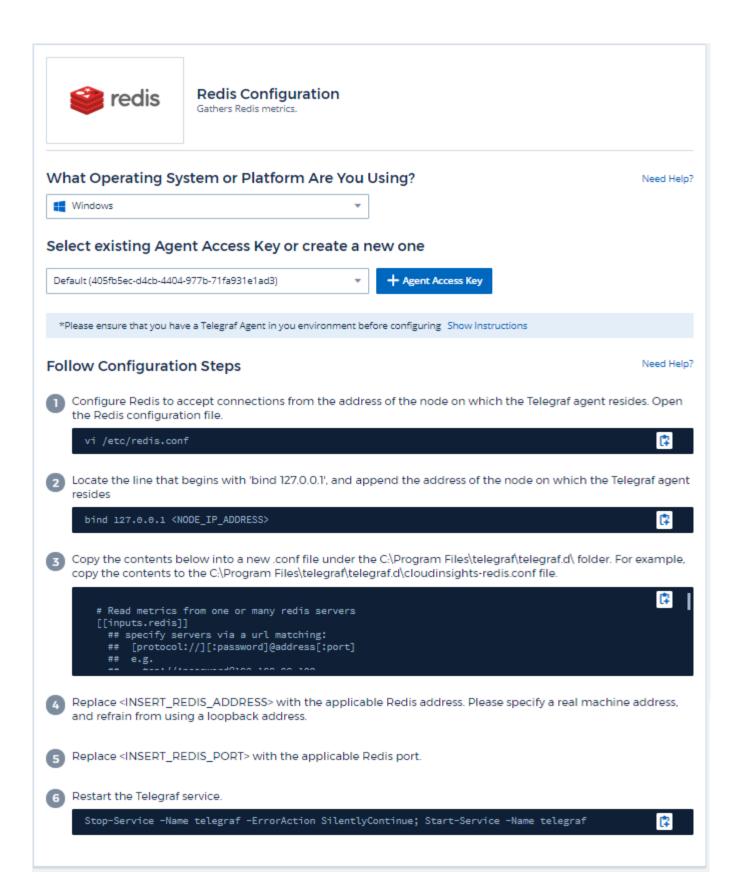
Cloud Insights uses this data collector to gather metrics from Redis. Redis is an open source, in-memory data structure store used as a database, cache, and message broker, supporting the following data structures: strings, hashes, lists, sets, and more.

Installation

1. From Admin > Data Collectors, click +Data Collector. Under Services, choose Redis.

Select the Operating System or Platform on which the Telegraf agent is installed.

- 2. If you haven't already installed an Agent for collection, or you wish to install an Agent for a different Operating System or Platform, click *Show Instructions* to expand the Agent installation instructions.
- 3. Select the Agent Access Key for use with this data collector. You can add a new Agent Access Key by clicking the + **Agent Access Key** button. Best practice: Use a different Agent Access Key only when you want to group data collectors, for example, by OS/Platform.
- 4. Follow the configuration steps to configure the data collector. The instructions vary depending on the type of Operating System or Platform you are using to collect data.



Information may be found in the Redis documentation.

Objects and Counters

The following objects and their counters are collected:

Object:	Identifiers:	Attributes:	Datapoints:
Redis	Namespace Server		

Troubleshooting

Additional information may be found from the Support page.

Copyright Information

Copyright © 2019–2020 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval systemwithout prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.