Configuring an Agent to Collect Data

Cloud Insights

Tony Lavoie
October 15, 2021

Table of Contents

Configuring an Agent to Collect Data

Cloud Insights uses Telegraf as its agent for collection of integration data. Telegraf is a plugin-driven server agent that can be used to collect and report metrics, events, and logs. Input plugins are used to collect the desired information into the agent by accessing the system/OS directly, by calling third-party APIs, or by listening to configured streams (i.e. Kafka, statsD, etc). Output plugins are used to send the collected metrics, events, and logs from the agent to Cloud Insights.

The current Telegraf version for Cloud Insights is **1.19.3**.

For accurate audit and data reporting, it is strongly recommended to synchronize the time on the Agent machine using **Network Time Protocol (NTP)** or **Simple Network Time Protocol (SNTP)**.

If you want to verify the installation files before instaling the Agent, see the section below on Verifying Checksums.

Installing an Agent

If you are installing a Service data collector and have not yet configured an Agent, you are prompted to first install an Agent for the appropriate Operating System. This topic provides instructions for installing the Telegraf agent on the following Operating Systems:

- Windows
- RHEL and CentOS
- Ubuntu and Debian
- macOS
- Kubernetes

To install an agent, regardless of the platform you are using, you must first do the following:

1. Log into the host you will use for your agent.
2. Log in to your Cloud Insights site and go to **Admin > Data Collectors**.
3. Click on **+Data Collector** and choose a data collector to install.

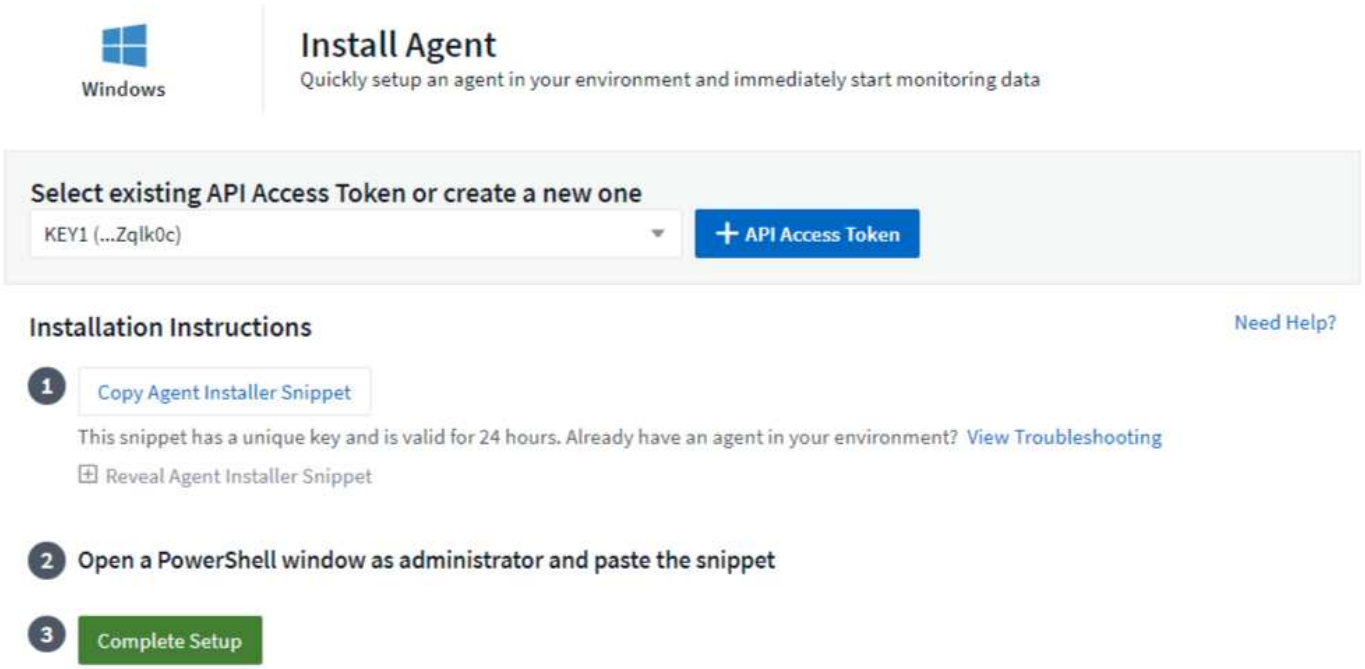
1. Choose the appropriate platform for your host (Windows, Linux, macOS, etc.)
2. Follow the remaining steps for each platform.

Once you have installed an agent on a host, you do not need to install an agent again on that host.

Once you have installed an agent on a server/VM, Cloud Insights collects metrics from that system in addition to collecting from any data collectors you configure. These metrics are gathered as "Node" metrics.

If you are using a proxy, read the proxy instructions for your platform before installing the Telegraf agent.

Windows



Pre-requisites:

- PowerShell must be installed
- If you are behind a proxy, follow the instructions in the **Configuring Proxy Support for Windows** section.

Steps to install agent on Windows:

1. Choose an Agent Access Key.
2. Copy the command block from the agent installation dialog. You can click the clipboard icon to quickly copy the command to the clipboard.
3. Open a PowerShell window
4. Paste the command into the PowerShell window and press Enter.
5. The command will download the appropriate agent installer, install it, and set a default configuration. When finished, it will restart the agent service. The command has a unique key and is valid for 24 hours.
6. Click **Finish** or **Continue**

After the agent is installed, you can use the following commands to start/stop the service:

```
Start-Service telegraf
Stop-Service telegraf
```

Configuring Proxy Support for Windows

The steps below outline the actions needed to set the *http_proxy/https_proxy* environment variables. For some proxy environments, users may also need to set the *no_proxy environment* variable.

For systems residing behind a proxy, perform the following to set the *https_proxy* and/or *http_proxy* environment variable(s) **PRIOR** to installing the Telegraf agent:

```
[System.Environment]::SetEnvironmentVariable("https_proxy",
"<proxy_server>:<proxy_port>",
[System.EnvironmentVariableTarget]::Machine)
```

Uninstalling the Agent

To uninstall the agent on Windows, do the following in a PowerShell window:

1. Stop and delete the Telegraf service:

```
Stop-Service telegraf
sc.exe delete telegraf
```

2. Remove the certificate from the trustore:

```
cd Cert:\CurrentUser\Root
rm E5FB7B68C08B1CA902708584C274F8EFC7BE8ABC
```

3. Delete the *C:\Program Files\telegraf* folder to remove the binary, logs, and configuration files
4. Remove the *SYSTEM\CurrentControlSet\Services\EventLog\Application\telegraf* key from the registry

Upgrading the Agent

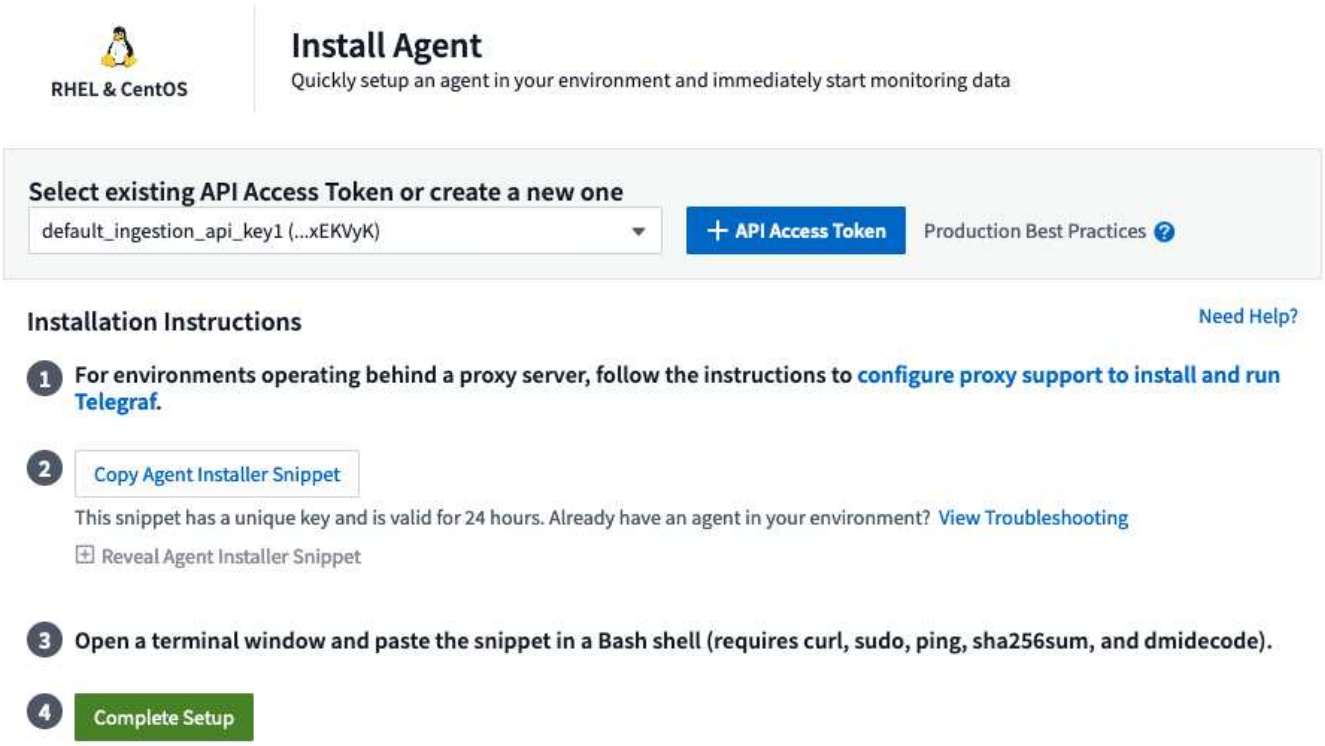
To upgrade the telegraf agent, do the following:

1. Stop and delete the telegraf service:

```
Stop-Service telegraf
sc.exe delete telegraf
```

2. Delete the *SYSTEM\CurrentControlSet\Services\EventLog\Application\telegraf* key from the registry
3. Delete *C:\Program Files\telegraf\telegraf.conf*
4. Delete *C:\Program Files\telegraf\telegraf.exe*
5. Install the new agent.

RHEL and CentOS



Pre-requisites:

- The following commands must be available: curl, sudo, ping, sha256sum, and dmidecode
- If you are behind a proxy, follow the instructions in the **Configuring Proxy Support for RHEL/CentOS** section.

Steps to install agent on RHEL/CentOS:

1. Choose an Agent Access Key.
2. Copy the command block from the agent installation dialog. You can click the clipboard icon to quickly copy the command to the clipboard.
3. Open a Bash window
4. Paste the command into the Bash window and press Enter.
5. The command will download the appropriate agent installer, install it, and set a default configuration. When finished, it will restart the agent service. The command has a unique key and is valid for 24 hours.
6. Click **Finish** or **Continue**

After the agent is installed, you can use the following commands to start/stop the service:

If your operating system is using systemd (CentOS 7+ and RHEL 7+):

```
sudo systemctl start telegraf
sudo systemctl stop telegraf
```

If your operating system is not using systemd (CentOS 7+ and RHEL 7+):

```
sudo service telegraf start
sudo service telegraf stop
```

Configuring Proxy Support for RHEL/CentOS

The steps below outline the actions needed to set the *http_proxy/https_proxy* environment variables. For some proxy environments, users may also need to set the *no_proxy environment* variable.

For systems residing behind a proxy, perform the following steps **PRIOR** to installing the Telegraf agent:

1. Set the *https_proxy* and/or *http_proxy* environment variable(s) for the current user:

```
export https_proxy=<proxy_server>:<proxy_port>
```

2. Create */etc/default/telegraf*, and insert definitions for the *https_proxy* and/or *http_proxy* variable(s):

```
https_proxy=<proxy_server>:<proxy_port>
```

Uninstalling the Agent

To uninstall the agent on RHEL/CentOS, in a Bash terminal, do the following:

1. Stop the Telegraf service:

```
systemctl stop telegraf (If your operating system is using systemd
(CentOS 7+ and RHEL 7+)
/etc/init.d/telegraf stop (for systems without systemd support)
```

2. Remove the Telegraf agent:

```
yum remove telegraf
```

3. Remove any configuration or log files that may be left behind:

```
rm -rf /etc/telegraf*
rm -rf /var/log/telegraf*
```

Upgrading the Agent

To upgrade the telegraf agent, do the following:

1. Stop the telegraf service:

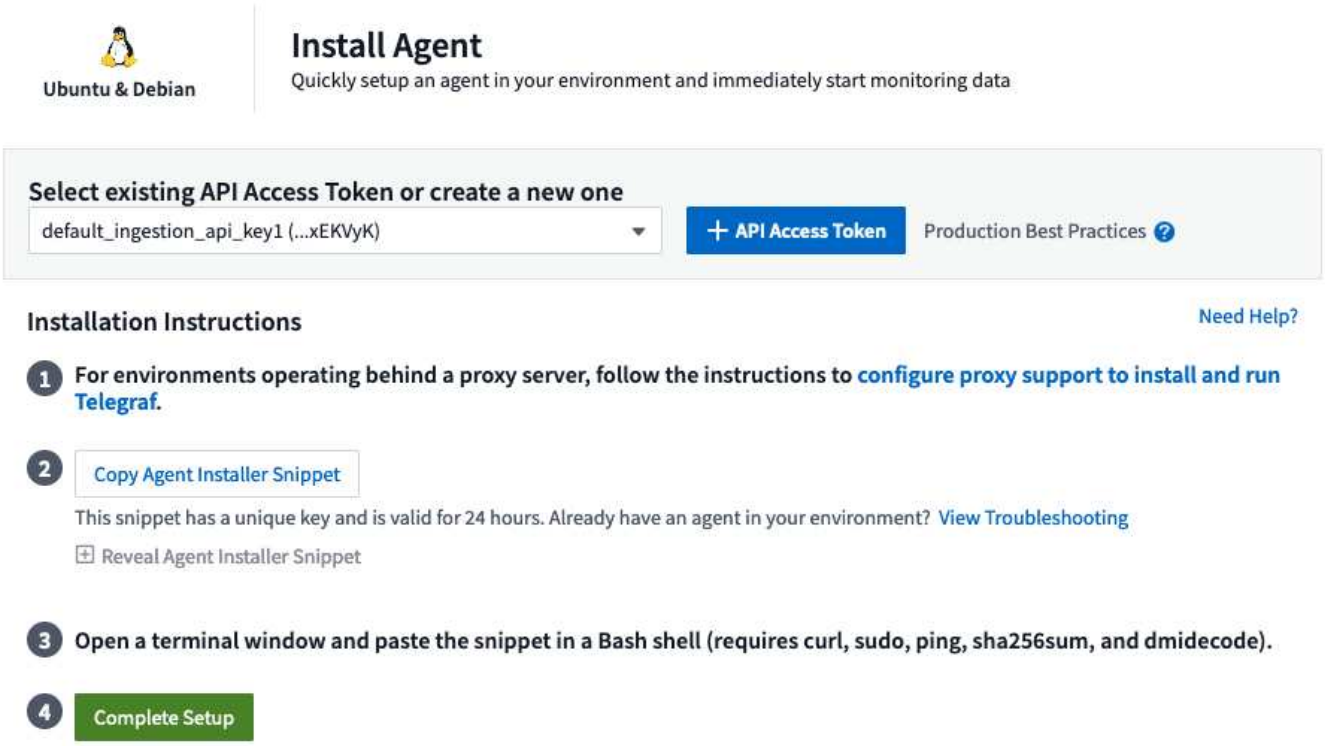
```
systemctl stop telegraf (If your operating system is using systemd
(CentOS 7+ and RHEL 7+)
/etc/init.d/telegraf stop (for systems without systemd support)
```

2. Remove the previous telegraf agent:

```
yum remove telegraf
```

3. Install the new agent.

Ubuntu and Debian



Pre-requisites:

- The following commands must be available: curl, sudo, ping, sha256sum, and dmidecode
- If you are behind a proxy, follow the instructions in the **Configuring Proxy Support for Ubuntu/Debian** section.

Steps to install agent on Debian or Ubuntu:

1. Choose an Agent Access Key.
2. Copy the command block from the agent installation dialog. You can click the clipboard icon to quickly copy the command to the clipboard.
3. Open a Bash window

4. Paste the command into the Bash window and press Enter.
5. The command will download the appropriate agent installer, install it, and set a default configuration. When finished, it will restart the agent service. The command has a unique key and is valid for 24 hours.
6. Click **Finish** or **Continue**

After the agent is installed, you can use the following commands to start/stop the service:

If your operating system is using systemd:

```
sudo systemctl start telegraf
sudo systemctl stop telegraf
```

If your operating system is not using systemd:

```
sudo service telegraf start
sudo service telegraf stop
```

Configuring Proxy Support for Ubuntu/Debian

The steps below outline the actions needed to set the *http_proxy/https_proxy* environment variables. For some proxy environments, users may also need to set the *no_proxy environment* variable.

For systems residing behind a proxy, perform the following steps **PRIOR** to installing the Telegraf agent:

1. Set the *https_proxy* and/or *http_proxy* environment variable(s) for the current user:

   ```
   export https_proxy=<proxy_server>:<proxy_port>
   ```

2. Create /etc/default/telegraf, and insert definitions for the *https_proxy* and/or *http_proxy* variable(s):

   ```
   https_proxy=<proxy_server>:<proxy_port>
   ```

Uninstalling the Agent

To uninstall the agent on Ubuntu/Debian, in a Bash terminal, run the following:

1. Stop the Telegraf service:

   ```
   systemctl stop telegraf (If your operating system is using systemd)
   /etc/init.d/telegraf stop (for systems without systemd support)
   ```

2. Remove the Telegraf agent:

```
dpkg -r telegraf
```

3. Remove any configuration or log files that may be left behind:

```
rm -rf /etc/telegraf*
rm -rf /var/log/telegraf*
```

Upgrading the Agent

To upgrade the telegraf agent, do the following:

1. Stop the telegraf service:

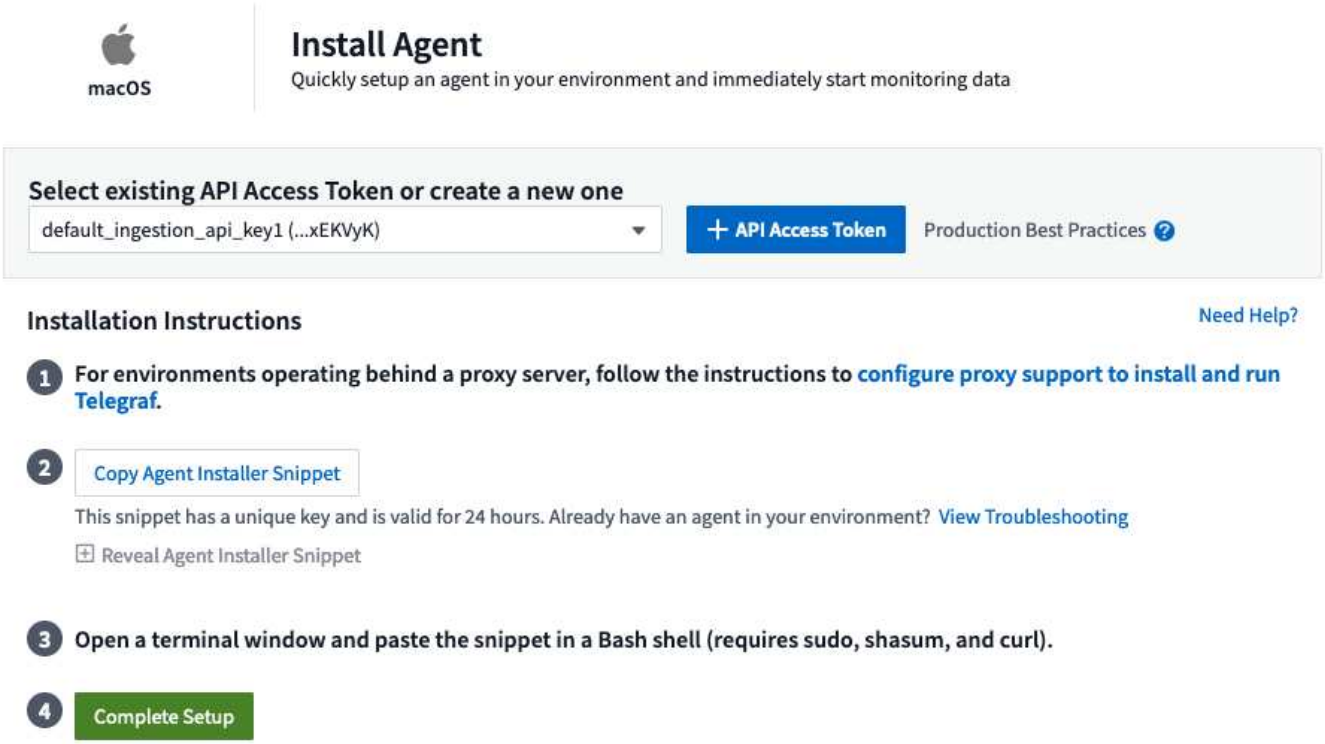
```
systemctl stop telegraf (If your operating system is using systemd)
/etc/init.d/telegraf stop (for systems without systemd support)
```

2. Remove the previous telegraf agent:

```
dpkg -r telegraf
```

3. Install the new agent.

macOS



Pre-requisites:

- The following commands must be available: curl, sudo, and shasum
- If you are behind a proxy, follow the instructions in the **Configuring Proxy Support for macOS** section.

Steps to install agent on macOS:

1. Choose an Agent Access Key.
2. Copy the command block from the agent installation dialog. You can click the clipboard icon to quickly copy the command to the clipboard.
3. Open a Bash window
4. Paste the command into the Bash window and press Enter.
5. The command will download the appropriate agent installer, install it, and set a default configuration. When finished, it will restart the agent service. The command has a unique key and is valid for 24 hours.
6. If you previously installed a Telegraf agent using Homebrew, you will be prompted to uninstall it. Once the previously installed Telegraf agent is uninstalled, re-run the command in step 5 above.
7. Click **Finish** or **Continue**

After the agent is installed, you can use the following commands to start/stop the service:

```
sudo launchctl start telegraf
sudo launchctl stop telegraf
```

Configuring Proxy Support for macOS

The steps below outline the actions needed to set the *http_proxy/https_proxy* environment variables. For some proxy environments, users may also need to set the *no_proxy environment* variable.

For systems residing behind a proxy, perform the following to set the *https_proxy* and/or *http_proxy* environment variable(s) for the current user **PRIOR** to installing the Telegraf agent:

```
export https_proxy=<proxy_server>:<proxy_port>
```

AFTER installing the Telegraf agent, add and set the appropriate *https_proxy* and/or *http_proxy* variable(s) in */Applications/telegraf.app/Contents/telegraf.plist*:

```
…
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
   <key>EnvironmentVariables</key>
   <dict>
          <key>https_proxy</key>
          <string><proxy_server>:<proxy_port></string>
   </dict>
   <key>Program</key>
   <string>/Applications/telegraf.app/Contents/MacOS/telegraf</string>
   <key>Label</key>
   <string>telegraf</string>
   <key>ProgramArguments</key>
   <array>
     <string>/Applications/telegraf.app/Contents/MacOS/telegraf</string>
     <string>--config</string>
     <string>/usr/local/etc/telegraf.conf</string>
     <string>--config-directory</string>
     <string>/usr/local/etc/telegraf.d</string>
   </array>
   <key>RunAtLoad</key>
   <true/>
</dict>
</plist>
…
```

Then, restart Telegraf after loading the above changes:

```
sudo launchctl stop telegraf
sudo launchctl unload -w /Library/LaunchDaemons/telegraf.plist
sudo launchctl load -w /Library/LaunchDaemons/telegraf.plist
sudo launchctl start telegraf
```

Uninstalling the Agent

To uninstall the agent on macOS, in a Bash terminal, run the following:

1. Stop the Telegraf service:

   ```
   sudo launchctl stop telegraf
   ```

2. Uninstall the telegraf agent:

   ```
   cp /Applications/telegraf.app/scripts/uninstall /tmp
   sudo /tmp/uninstall
   ```

3. Remove any configuration or log files that may be left behind:

   ```
   rm -rf /usr/local/etc/telegraf*
   rm -rf /usr/local/var/log/telegraf.*
   ```

Upgrading the Agent

To upgrade the telegraf agent, do the following:

1. Stop the telegraf service:

   ```
   sudo launchctl stop telegraf
   ```

2. Uninstall the previous telegraf agent:

   ```
   cp /Applications/telegraf.app/scripts/uninstall /tmp
   sudo /tmp/uninstall
   ```

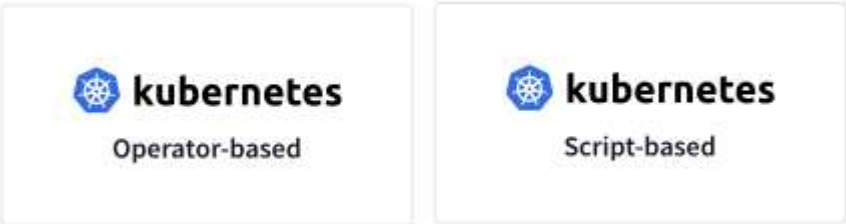
3. Install the new agent.

Kubernetes

Kubernetes offers two ways to collect data:

- Operator-based configuration. This is recommended for Kubernetes.
- Traditional script-based Agent installation

Installation instructions vary based on which tile you choose.



Operator-based installation is considered a *Preview* feature and is therefore subject to change.

Pre-requisites:

- The following commands must be available: curl, sudo, kubectl

  For best results, add these commands to the PATH.

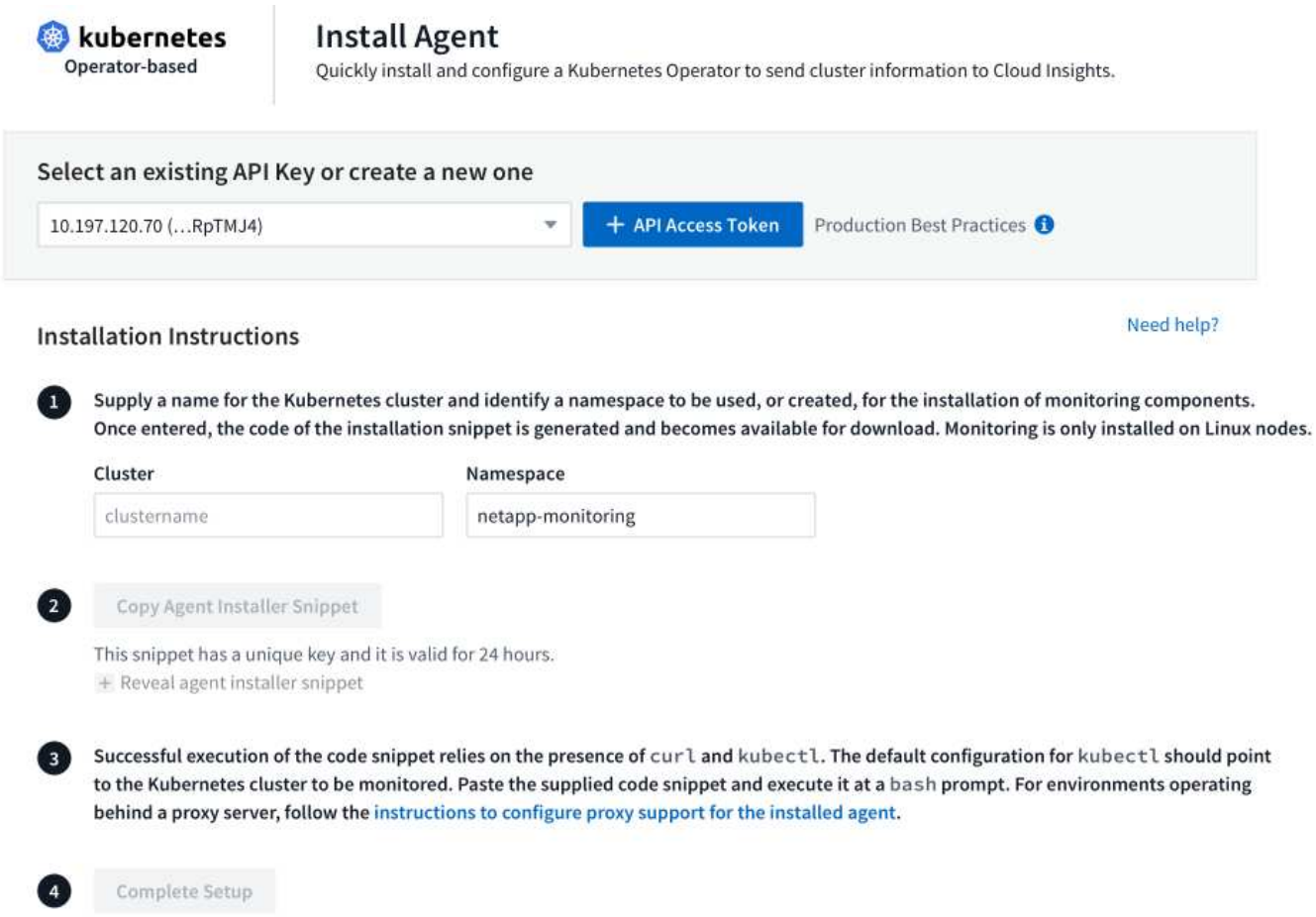
- kube-state-metrics must be installed. See below for more information. kube-state-metrics is automatically installed with Operator-based installation.
- If you are behind a proxy, follow the instructions in the **Configuring Proxy Support for Kubernetes** section.
- If you are running a Kubernetes variant that requires security context constraints, follow the instructions in the **Configuring the Agent to Collect Data from Kubernetes** section. Operator-based installation installs this for you.
- You must have permissions to create Kubernetes cluster roles and role bindings.
- Operator-based installation has been tested and is expected to work with AWS EKS 1.18 and OpenShift 3.11.

Monitoring is only installed on Linux nodes

Cloud Insights supports monitoring of Kubernetes nodes that are running Linux, by specifying a Kubernetes node selector that looks for the following Kubernetes labels on these platforms:

Platform	Label
Kubernetes v1.14 and above	Kubernetes.io/os = linux
Kubernetes v1.13 and below	beta.kubernetes.io/os = linux
Rancher + cattle.io as orchestration/Kubernetes platform	cattle.io/os = linux

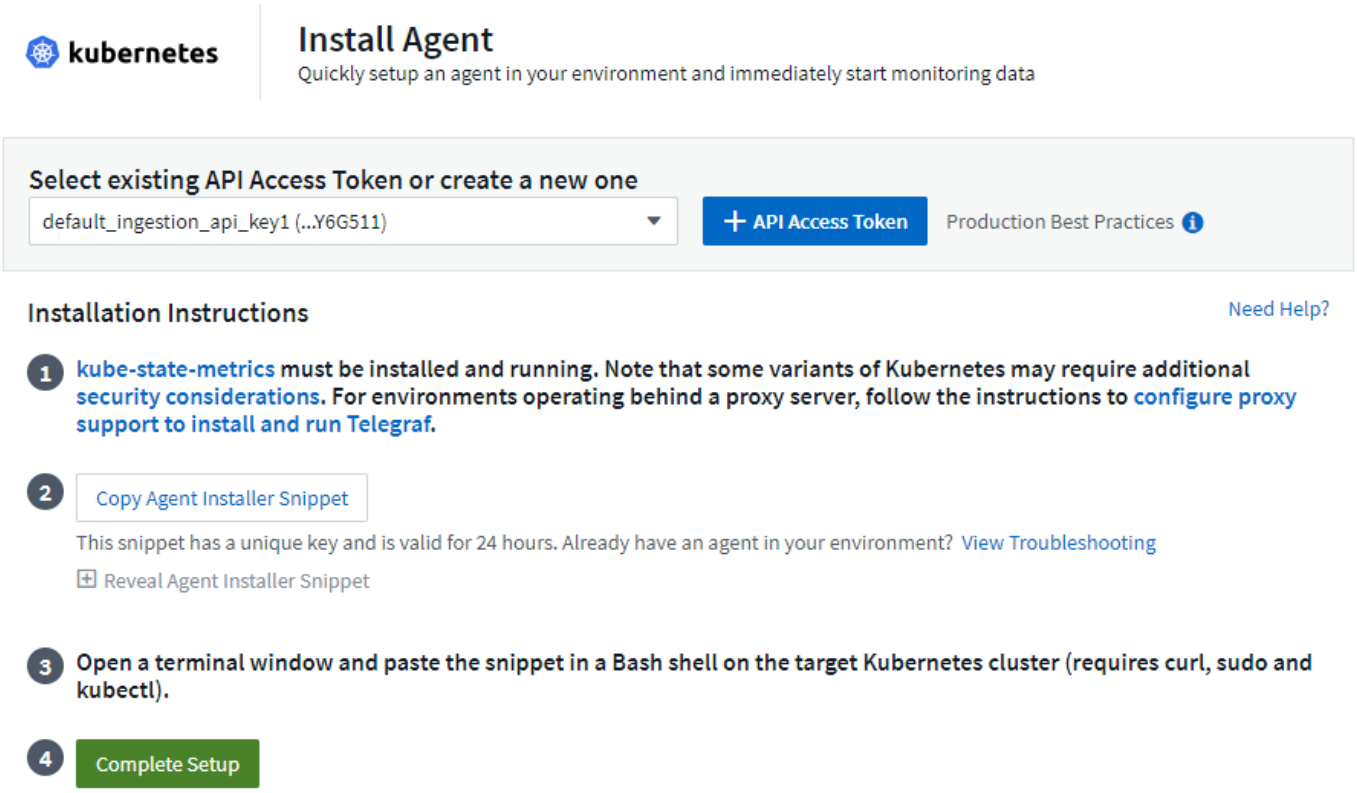
Operator-Based Installation



Steps to install Operator-based agent on Kubernetes:

1. Enter the cluster name and namespace.
2. Once these are entered, you can copy the Agent Installer snippet
3. Click the button to copy this snippet to the clipboard.
4. Paste the snippet into a *bash* window and execute it.
5. The installation proceeds automatically. When it is complete, click the *Complete Setup* button.

Script-Based Installation



Steps to install Script-based agent on Kubernetes:

1. Choose an Agent Access Key.
2. Click the **Copy Agent Installer Snippet** button in the installation dialog. You can optionally click the *+Reveal Agent Installer Snippet* button if you want to view the command block.
3. Paste the command into a *bash* window.
4. Optionally, you can override the namespace or provide the cluster name as part of the install command by modifying the command block to add one or both of the following before the final *./$installerName*
   - CLUSTER_NAME=<Cluster Name>
   - NAMESPACE=<Namespace>

   Scroll through the following example to see this in place in the command block:

```
installerName=cloudinsights-kubernetes.sh && token=<token> &&
key=c642e336-91f4-4c6f-8086-72faabd6aff6 &&
domain=tenant1.testk8.cloudinsights-test.netapp.com && curl -k -X GET
-H "Authorization: Bearer $token" -H "X-CloudInsights-ApiKey-Id:
$key" -o $installerName
https://$domain/rest/v1/lake/telegraf/platforms/installer?platform=ku
bernetes && chmod +x $installerName && sudo --preserve-env JWT=$token
DOMAIN_NAME=$domain API_KEY_ID=$key CLUSTER_NAME=TEST_CLUSTER
NAMESPACE=NEW-NAMESPACE ./$installerName
```

CLUSTER_NAME is the name of the Kubernetes cluster from Cloud Insights collects metrics, while *NAMESPACE* is the namespace to which the Telegraf agent will be deployed. The specified namespace will be created if it does not already exist.

5. When ready, execute the command block.
6. The command will download the appropriate agent installer, install it, and set a default configuration. If you have not explicitly set the *namespace*, you will be prompted to enter it. When finished, the script will restart the agent service. The command has a unique key and is valid for 24 hours.
7. When finished, click **Complete Setup**.

DaemonSet, ReplicaSet, and Stopping/Starting the agent

A DaemonSet and ReplicaSet will be created on the Kubernetes cluster to run the required Telegraf agents/pods. By default, these Telegraf agents/pods will be scheduled on both master and non-master nodes.

To facilitate stopping and restarting of the agent, generate the Telegraf DaemonSet YAML and ReplicaSet YAML using the following commands. Note that these commands are using the default namespace "ci-monitoring". If you have set your own namespace, substitute that namespace in these and all subsequent commands and files:

If you have set your own namespace, substitute that namespace in these and all subsequent commands and files:

```
kubectl --namespace ci-monitoring get ds telegraf-ds -o yaml >
/tmp/telegraf-ds.yaml
kubectl --namespace ci-monitoring get rs telegraf-rs -o yaml >
/tmp/telegraf-rs.yaml
```

You can then use the following commands to stop and start the Telegraf service:

```
kubectl --namespace ci-monitoring delete ds telegraf-ds
kubectl --namespace ci-monitoring delete rs telegraf-rs
```

```
kubectl --namespace ci-monitoring apply -f /tmp/telegraf-ds.yaml
kubectl --namespace ci-monitoring apply -f /tmp/telegraf-rs.yaml
```

Configuring Proxy Support for Kubernetes

The steps below outline the actions needed to set the *http_proxy/https_proxy* environment variables. For some proxy environments, users may also need to set the *no_proxy environment* variable.

For systems residing behind a proxy, perform the following to set the *https_proxy* and/or *http_proxy* environment variable(s) for the current user **PRIOR** to installing the Telegraf agent:

```
export https_proxy=<proxy_server>:<proxy_port>
```

AFTER installing the Telegraf agent, add and set the appropriate *https_proxy* and/or *http_proxy* environment variable(s) to the *telegraf-ds* daemonset and *telegraf-rs* replicaset.

```
kubectl edit ds telegraf-ds
```

```
…
        env:
        - name: https_proxy
          value: <proxy_server>:<proxy_port>
        - name: HOSTIP
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: status.hostIP
…
```

```
kubectl edit rs telegraf-rs
```

```
…
        env:
        - name: https_proxy
          value: <proxy_server>:<proxy_port>
        - name: HOSTIP
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: status.hostIP
…
```

Then, restart Telegraf:

```
kubectl delete pod telegraf-ds-*
kubectl delete pod telegraf-rs-*
```

Configuring the Agent to Collect Data from Kubernetes

Note: The default namespace for Script-based installation is *ci-monitoring*. For Operator-based installation, the default namespace is *netapp-monitoring*. In commands involving namespace, be sure to specify the correct

namespace for your installation.

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:ci-monitoring:monitoring-operator
    groups: []
```

Installing the kube-state-metrics server

Operator-based install handles the installation of kube-state-metrics. Skip this section if you are performing Operator-based installation.

It is strongly recommended to use kube-state-metrics version 2.0 or later in order to take advantage of the full feature set including the ability to link Kubernetes persistent volumes (PVs) to backend storage devices. Note also that with kube-state-metrics version 2.0 and above, Kubernetes object labels are not exported by default. To configure kube-state-metrics to export Kubernetes object labels, you must specify a metric labels "allow" list. Refer to the *--metric-labels-allowlist* option in the kube-state-metrics documentation.

Use the following steps to install the kube-state-metrics server (required if you are performing script-based installation):

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. ConfigMap Metrics
2. DaemonSet Metrics
3. Deployment Metrics
4. Ingress Metrics
5. Namespace Metrics
6. Node Metrics
7. Persistent Volume Metrics
8. Persistant Volume Claim Metrics
9. Pod Metrics
10. ReplicaSet metrics
11. Secret metrics
12. Service metrics
13. StatefulSet metrics

Uninstalling the Agent

Note that these commands are using the default namespace "ci-monitoring". If you have set your own namespace, substitute that namespace in these and all subsequent commands and files.

To uninstall the script-based agent on Kubernetes, do the following:

If the monitoring namespace is being used solely for Telegraf:

```
kubectl --namespace ci-monitoring delete
ds,rs,cm,sa,clusterrole,clusterrolebinding -l app=ci-telegraf
```

```
kubectl delete ns ci-monitoring
```

If the monitoring namespace is being used for other purposes in addition to Telegraf:

```
kubectl --namespace ci-monitoring delete
ds,rs,cm,sa,clusterrole,clusterrolebinding -l app=ci-telegraf
```

For Operator-based installation run the following commands:

```
kubectl delete ns netapp-monitoring
kubectl delete agent agent-monitoring-netapp
kubectl delete crd agents.monitoring.netapp.com
kubectl delete role agent-leader-election-role
kubectl delete clusterrole agent-manager-role agent-proxy-role agent-
metrics-reader
kubectl delete clusterrolebinding agent-manager-rolebinding agent-proxy-
rolebinding agent-cluster-admin-rolebinding
```

Upgrading the Agent

Note that these commands are using the default namespace "ci-monitoring". If you have set your own namespace, substitute that namespace in these and all subsequent commands and files.

To upgrade the telegraf agent, do the following:

1. Back up the existing configurations:

   ```
   kubectl --namespace ci-monitoring get cm -o yaml > /tmp/telegraf-
   configs.yaml
   ```

1. Uninstall the Agent (see above for instructions)
2. Install the new agent.

Verifying Checksums

The Cloud Insights agent installer performs integrity checks, but some users may want to perform their own verifications before installing or applying downloaded artifacts. To perform a download-only operation (as opposed to the default download-and-install), these users can edit the agent installation command obtained from the UI and remove the trailing “install” option.

Follow these steps:

1. Copy the Agent Installer snippet as directed.
2. Instead of pasting the snippet into a command window, paste it into a text editor.
3. Remove the trailing “--install” (Linux/Mac) or “-install” (Windows) from the command.
4. Copy the entire command from the text editor.
5. Now paste it into your command window (in a working directory) and run it.

Non-Windows:

- Download and install (default):

```
installerName=cloudinsights-kubernetes.sh … && sudo -E -H
./$installerName --download --install
```

- Download-only:

```
installerName=cloudinsights-kubernetes.sh … && sudo -E -H
./$installerName --download
```

Windows:

- Download and install (default):

```
!$($installerName=".\cloudinsights-windows.ps1") … -and
$(&$installerName -download -install)
```

- Download-only:

```
!$($installerName=".\cloudinsights-windows.ps1") … -and
$(&$installerName -download)
```

The download-only command will download all required artifacts from Cloud Insights to the working directory. The artifacts include, but may not be limited to:

- an installation script
- an environment file
- YAML files
- a signed checksum file (sha256.signed)
- a PEM file (netapp_cert.pem) for signature verification

The installation script, environment file, and YAML files can be verified using visual inspection.

The PEM file can be verified by confirming its fingerprint to be the following:

```
E5:FB:7B:68:C0:8B:1C:A9:02:70:85:84:C2:74:F8:EF:C7:BE:8A:BC
```

More specifically,

- Non-Windows:

  ```
  openssl x509 -fingerprint -sha1 -noout -inform pem -in netapp_cert.pem
  ```

- Windows:

  ```
  Import-Certificate -Filepath .\netapp_cert.pem -CertStoreLocation
  Cert:\CurrentUser\Root
  ```

The signed checksum file can be verified using the PEM file:

- Non-Windows:

  ```
  openssl smime -verify -in sha256.signed -CAfile netapp_cert.pem -purpose
  any
  ```

- Windows (after installing the certificate via Import-Certificate above):

  ```
  Get-AuthenticodeSignature -FilePath .\sha256.ps1 $result = Get-
  AuthenticodeSignature -FilePath .\sha256.ps1 $signer =
  $result.SignerCertificate Add-Type -Assembly System.Security
  [Security.Cryptography.x509Certificates.X509Certificate2UI]::DisplayCert
  ificate($signer)
  ```

Once all of the artifacts have been satisfactorily verified, the agent installation can be initiated by running:

Non-Windows:

```
sudo -E -H ./<installation_script_name> --install
```

Windows:

```
.\ cloudinsights-windows.ps1 -install
```

Troubleshooting Agent Installation

Some things to try if you encounter problems setting up an agent:

Problem:	Try this:
I already installed an agent using Cloud Insights	If you have already installed an agent on your host/VM, you do not need to install the agent again. In this case, simply choose the appropriate Platform and Key in the Agent Installation screen, and click on **Continue** or **Finish**.
I already have an agent installed but not by using the Cloud Insights installer	Remove the previous agent and run the Cloud Insights Agent installation, to ensure proper default configuration file settings. When complete, click on **Continue** or **Finish**.
I do not see a hyperlink/connection between my Kubernetes Persistent Volume and the corresponding back-end storage device. My Kubernetes Persistent Volume is configured using the hostname of the storage server.	Follow the steps to uninstall the existing Telegraf agent, then re-install the latest Telegraf agent. You must be using Telegraf version 2.0 or later.

Problem:	Try this:
I’m seeing messages in the logs resembling the following: E0901 15:21:39.962145 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: Failed to list *v1.MutatingWebhookConfiguration: the server could not find the requested resource E0901 15:21:43.168161 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: Failed to list *v1.Lease: the server could not find the requested resource (get leases.coordination.k8s.io) etc.	These messages may occur if you are running kube-state-metrics version 2.0.0 or above with Kubernetes version 1.17 or below. To get the Kubernetes version: *kubectl version* To get the kube-state-metrics version: *kubectl get deploy/kube-state-metrics -o jsonpath='{..image}'* To prevent these messages from happening, users can modify their kube-state-metrics deployment to disable the following Leases: *mutatingwebhookconfigurations validatingwebhookconfigurations volumeattachments resources* More specifically, they can use the following CLI argument: resources=certificatesigningrequests,configmaps,cronjobs,daemonsets, deployments,endpoints,horizontalpodautoscalers,ingresses,jobs,limitranges, namespaces,networkpolicies,nodes,persistentvolumeclaims,persistentvolumes, poddisruptionbudgets,pods,replicasets,replicationcontrollers,resourcequotas, secrets,services,statefulsets,storageclasses The default resource list is: "certificatesigningrequests,configmaps,cronjobs,daemonsets,deployments, endpoints,horizontalpodautoscalers,ingresses,jobs,leases,limitranges, mutatingwebhookconfigurations,namespaces,networkpolicies,nodes, persistentvolumeclaims,persistentvolumes,poddisruptionbudgets,pods,replicasets, replicationcontrollers,resourcequotas,secrets,services,statefulsets,storageclasses, validatingwebhookconfigurations,volumeattachments"

Problem:	Try this:
I installed or upgraded Telegraf on Kubernetes, but the Telegraf pods are not starting up. The Telegraf ReplicaSet or DaemonSet is reporting a failure resembling the following: Error creating: pods "telegraf-rs-" is forbidden": unable to validate against any security context constraint: [spec.volumes[2]: Invalid value: "hostPath": hostPath volumes are not allowed to be used]	Create a Security Context Constraint (refer to the Configuring the Agent to Collect Data from Kubernetes section above) if one does not already exist. Ensure the namespace and service account specified for the Security Context Constraint matches the namespace and service account for the Telegraf ReplicaSet and DaemonSet. kubectl describe scc telegraf-hostaccess |grep serviceaccount kubectl -n ci-monitoring --describe rs telegraf-rs | grep -i "Namespace:" kubectl -n ci-monitoring describe rs telegraf-rs | grep -i "Service Account:" kubectl -n ci-monitoring --describe ds telegraf-ds | grep -i "Namespace:" kubectl -n ci-monitoring describe ds telegraf-ds | grep -i "Service Account:"
I see error messages from Telegraf resembling the following, but Telegraf does start up and run: Oct 11 14:23:41 ip-172-31-39-47 systemd[1]: Started The plugin-driven server agent for reporting metrics into InfluxDB. Oct 11 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="failed to create cache directory. /etc/telegraf/.cache/snowflake, err: mkdir /etc/telegraf/.ca che: permission denied. ignored\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" Oct 11 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="failed to open. Ignored. open /etc/telegraf/.cache/snowflake/ocsp_response_cache.json: no such file or directory\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" Oct 11 14:23:41 ip-172-31-39-47 telegraf[1827]: 2021-10-11T14:23:41Z I! Starting Telegraf 1.19.3	This is a known issue. Refer to This GitHub article for more details. As long as Telegraf is up and running, users can ignore these error messages.

Additional information may be found from the Support page or in the Data Collector Support Matrix.