Kubernetes Data Collector

Cloud Insights

Tony Lavoie, Dave Grace
March 24, 2020

NetApp®

Table of Contents

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

Need Help?

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

```
#############################
## kubernetes
#############################

[[inputs.kubernetes]]
```

2. Replace <INSERT_KUBELET_ADDRESS> with the "$HOSTIP".

3. Replace <INSERT_KUBELET_PORT> with the applicable kubelet port. The typical port is 10250.

4. Replace <INSERT_K8S_CLUSTER_NAME> with the name of the Kubernetes cluster.

5. 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.

6. By default, the Telegraf agent will use the kubelet access token file found at /var/run/secrets/kubernetes.io/serviceaccount/token within each pod.

7. 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.

```
#############################
## kubernetes - prometheus
#############################

[[inputs.prometheus]]
```

8. Restart each applicable Telegraf pod.

```
kubectl delete pod <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
'{print $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

The following objects and their counters are collected:

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 Bytes (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: sudo kubectl --namespace monitoring get pods	Check if there were any errors deploying the DaemonSet: sudo kubectl --namespace monitoring describe ds telegraf-ds If there are errors related to SecurityContextConstraints, do the following: 1. Generate the Telegraf DaemonSet YAML sudo kubectl --namespace monitoring get ds telegraf-ds -o yaml > /tmp/telegraf-ds.yaml 2. Stop the Telegraf service sudo kubectl --namespace 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>

Additional information may be found from the Support page.