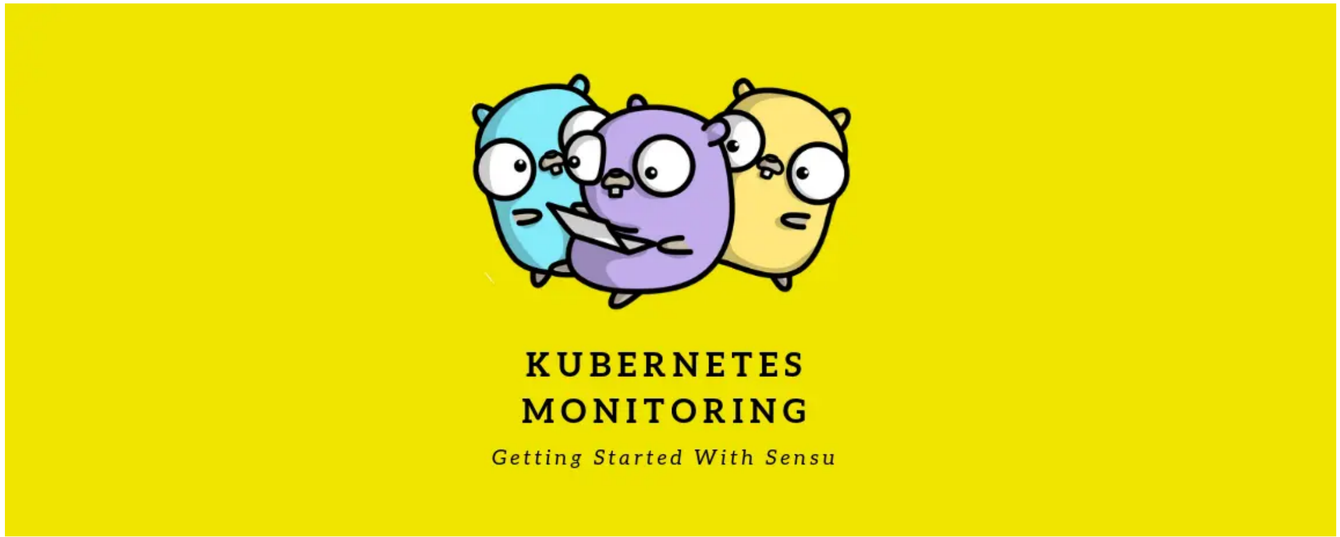
K — KUBERNETES

Kubernetes monitoring With Sensu: Setting up Container Sidecar Agent

by **Jef Spaleta** · August 27, 2019



The rise of containerized infrastructure has caused us to rethink the way we build and deploy our applications. But, with all that speed and flexibility comes challenges, especially when it comes to maintaining visibility into your (often multi-generational) infrastructure.

In this post, I'll discuss some of the challenges to monitoring containers and Kubernetes (the go-to for container orchestration), talk about some of the data sources and a few methods for monitoring Kubernetes, and walk you through a tutorial on monitoring containers with Sensu, a multi-cloud monitoring tool.

Container monitoring: the challenges

In a containerized world, your applications are constantly moving, making it even more difficult to keep tabs on them. It's also the norm to have distributed applications, which causes a whole new set of problems when it comes to

important to keep track of the labels and annotations associated with pods and containers.

Monitoring Kubernetes: the data sources

When it comes to monitoring Kubernetes, there are essentially four data sources you're plugging into your monitoring tool:

1. The Kubernetes hosts running the Kubelet. The most common way to get data out of these hosts is to use the Prometheus node exporter to scrape data from Kubernetes and expose system resource telemetry data on an HTTP endpoint.

2. The Kubelet metrics, which includes metrics for apiserver, kube-scheduler, and kube-controller-manager.

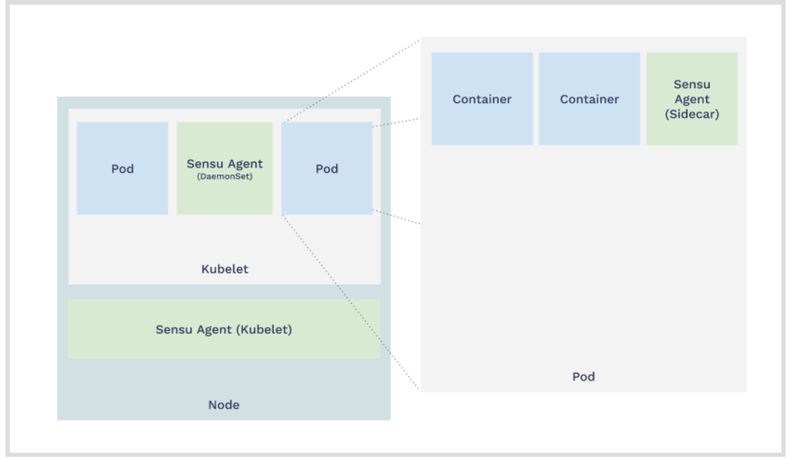
3. The Kubelet's built-in cAdvisor, which collects, aggregates, processes, and exports metrics for your running containers.

4. kube-state-metrics, which gives you data at the cluster level, such as all the pods you have configured and their current state.

Monitoring Kubernetes: the sidecar pattern

Of these data sources, we tend to prefer kube-state-metrics, as it gives you most of what you need without overloading you with information. I'll use that for my tutorial of container monitoring with Sensu, but wanted to note that Sensu can also scrape Prometheus metrics (but that's a whole other post — for now, feel free to check out our Prometheus collector asset).

Using the sidecar pattern, you can deploy a Sensu agent alongside your application container. A sidecar approach makes it possible for each Kubernetes pod to host your application container alongside other containers running support processes, such as the Sensu agent. Since all containers running inside a pod all share the same network space, your applications can talk to Sensu as if they were running in the same container.



Tutorial

In this tutorial, we'll install and configure sensuctl (Sensu's command-line tool), deploy the Sensu backend using a Kubernetes deployment, and deploy Sensu agent sidecars. This tutorial assumes you have Kubernetes installed. If you don't, take a look at setting up Minikube which will make things easier for the tutorial.

1. Deploy the Sensu backend using a Kubernetes deployment

Download the sensu-kube-demo repo. Use this sensu-backend.yaml file from the repo and the following command to deploy Sensu:

```
kubectl create -f go/deploy/sensu-backend.yaml
```

The `go/deploy/` path reference is relative to the top directory of the repository checkout.

You will also want to establish a Kubernetes ingress policy and host DNS configuration that will allow you to communicate with the sensu-backend running inside of the Kubernetes node. For the rest of the tutorial, I will assume you have configured things such that sensu.local tcp port 80 maps to the sensu-backend pod's TCP port 8080, the default port for the sensu-backend API service. The sensu-kube-demo repository includes an appropriate nginx ingress policy example.

2. [Optional] Install sensuctl on your workstation

sensuctl is a command-line tool for managing resources within Sensu. It works by

entities. The below instructions are for Ubuntu — check out our documentation for sensuctl installation on CentOS, Windows, and macOS.

To install sensuctl on Ubuntu:

1. Add the Sensu repository

```
curl -s
https://packagecloud.io/install/repositories/sensu/stable/script.deb.s
 | sudo bash
```

2. Install the sensu-go-cli package

```
sudo apt-get install sensu-go-cli
```

In order to use sensuctl from your workstation to communicate to the containerized sensu-backend, you'll need to make sure to configure your Kubernetes ingress controller and network DNS settings to allow access to the sensu-backend API outside of the Kubernetes cluster.

To save time, for this tutorial you can alternatively use the sensuctl provided in the sensu-backend container. In general you'll want to use sensuctl installed natively in your workstation environment for ease of operation.

3a. Configure sensuctl to use the built-in admin user on your workstation

If you are using the sensuctl configured on your workstation, enter the following to configure sensuctl:

```
sensuctl configure -n \
--username 'admin'\
--password 'P@assw0rd!'\
--namespace default\
--url 'http://sensu.local'
```

Please note you'll need to adjust the URL to match your Kubernetes cluster ingest configuration and DNS configuration.

3b. Configure sensuctl to use the built-in admin user inside the cluster

```
kubectl exec -it sensu-backend-<replace> -- /bin/sh
```

You can discover the correct pod name with:

```
kubectl get pods -l app=sensu-backend
```

```
sensuctl configure -n \
--username 'admin'\
--password 'P@assw0rd!'\
--namespace default\
--url 'http://localhost:8080'
```

Here you can use the localhost URL as if the sensu-backend were running locally, because your sensuctl is operating in the same pod, sharing the pod network with the sensu-backend container.

4. Create the Sensu namespace for the dummy application

All the agents used in this tutorial operate in a Sensu namespace named demo. We'll need to create the namespace now using sensuctl:

```
sensuctl namespace create demo
```

We can optionally reconfigure sensuctl to use the demo namespace as default:

```
sensuctl configure -n \
--username 'admin'\
--password 'P@assw0rd!'\
--namespace demo\
```

5. Deploy Sensu agent sidecars

Using the aforementioned sidecar pattern and the dummy.sensu.yaml file, enter the following to deploy Sensu agent sidecars for two example app instances using a Kubernetes deployment:

```
kubectl apply -f go/deploy/dummy.sensu.yaml
```

If you have set the default namespace to demo, use sensuctl entity list to see the agent containers, otherwise, use sensuctl entity list –namespace demo to set the namespace explicitly. Each agent is running in a separate pod as a sidecar to support the dummy application as per the deployment configuration in the

examine the running dummy service in each pod by using the dummy subscription, or a specific pod using the specific agent associated with each agent.

For more on container monitoring with Sensu — including using the Sensu web UI to view events and setting up workflows to Slack and InfluxDB — check out our interactive tutorial. Thanks for reading, and happy monitoring!

TAGS: Kubernetes Monitoring | Monitoring | Sensu

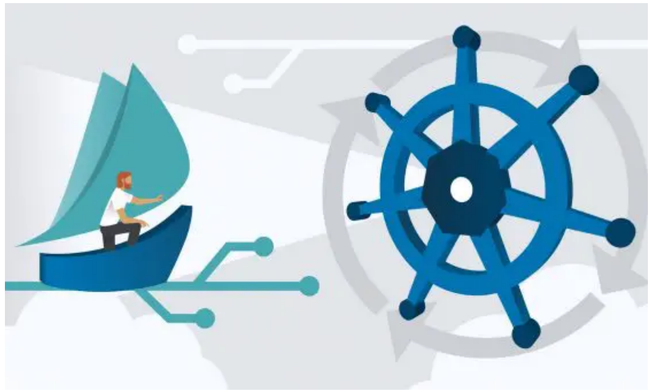
16 SHARES: SHARE 16 | TWEET

Jef Spaleta

Jef Spaleta is a Developer Advocate for Sensu and lives in Fairbanks, Alaska. His interests range from technology to coffee, curling and singing.

VIEW COMMENTS (0)

YOU MAY ALSO LIKE



T — TUTORIAL

Kubernetes Tutorials For Beginners: Getting Started Guide

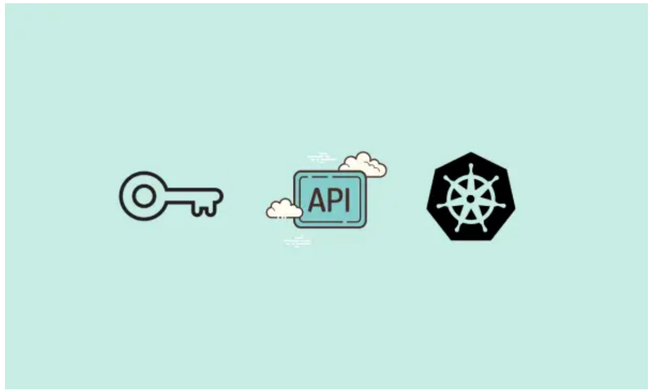
by **devopscube** · April 14, 2020

Official kubernetes (k8s) website says, Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications....

K — KUBERNETES

CKAD Exam Study Guide: A Complete Resource for CKAD Aspirants

by **Shishir Khandelwal** · June 28, 2021



K — KUBERNETES

How to Create kubernetes Role for Service Account

by **Bibin Wilson** · June 1, 2021

In this blog, you will learn how to create Kubernetes role for a service account and use it...

K — KUBERNETES

How To Setup Kube State Metrics on Kubernetes

by **Bibin Wilson** · November 3, 2019

In this blog we will look at what is Kube State Metrics and its setup on Kubernetes. What...

D — DEVOPS

How to Setup Prometheus Monitoring On Kubernetes Cluster

by **Bibin Wilson** · April 7, 2021

This article will guide you through setting up Prometheus on a Kubernetes cluster for monitoring the Kubernetes cluster....

K — KUBERNETES

Kubernetes Certification Coupon: 18% Off + $100 Off On Kubernetes Course bundle

by **devopscube** · July 20, 2021