

Full Stack Visibility with Kubernetes in 15 Minutes

Mike Moore
Datadog Enterprise Solutions Engineer.



Trivia

Why do they call it **K8s!**?





Kubernetes with Datadog

A look from 50,000 feet

Integration Auto Discovery

LIVE CONTAINER MONITORING



Infrastructure Metrics
Custom and out-of-the-box

Logs & Events
Structured and unstructured

Tags/Metadata
Custom and out-of-the-box

Retention
15 months by default (extendable on request)

Granularity
No roll ups, full granularity

Secure Focused
Encrypted at Rest, SOC 2 Compliant

APM
Structured and unstructured

Highly Available
Spanning multiple AZ's

Encrypted Communication
Outbound only via HTTPS/443

System Metrics
80+ CPU, Disk, Load, Network, etc.

250+ Integrations
Kubernetes, Docker, AWS (Lambda, S3, etc.), Ansible, Kubernetes, MapR and more

High resolution
15 sec host granularity, 1 sec. custom and business metric granularity





Deploying Datadog in K8s

Let's start with the basics



Deployment Options

Host Based

- Additional Visibility
- Monitor Outside of K8s
- More Granular Configuration
- No Auto-Deployment without Config Management

Container Based

- Deploy as a *DaemonSet*
- Simplified Management
- Deploy Quickly Everywhere
- Easier to Deploy and Configure

https://docs.datadoghq.com/agent/kubernetes/host_setup/



Tagging and Other Best Practices

Making sense out of all of this data



Good Tagging Practices

What Should You Tag?

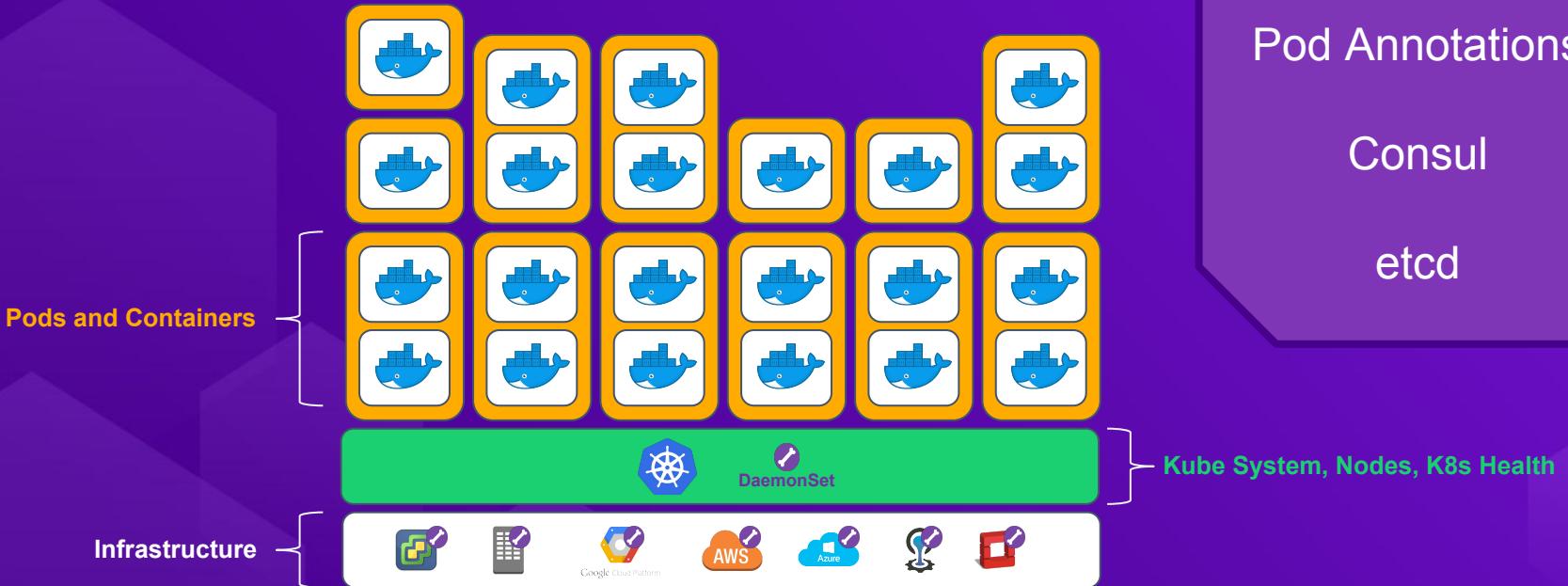
Applications	Roles
Services	Customers
Components	Business Units
Teams	Stores
Departments	Regions
Cost Centers	etc...

Why It Matters:

If we can't see or alert on what we want when we need to then monitoring anything has little point!

Where should we tag?

And what tags do we already get?





Kubernetes State Metrics

What are these?

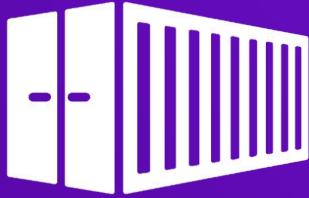
kube-state-metrics is a simple service that listens to the Kubernetes API server and generates metrics about the state of the objects.

It is not focused on the health of the individual Kubernetes components, but rather on the health of the various objects inside, such as deployments, nodes and pods.

How do I get them?

1. Download the [Kube-State manifests folder](https://github.com/kubernetes/kube-state-metrics/tree/master/kubernetes) from GitHub:
<https://github.com/kubernetes/kube-state-metrics/tree/master/kubernetes>
2. Apply them to your Kubernetes cluster:
`kubectl apply -f <KUBE_STATE_FOLDER>`

https://docs.datadoghq.com/agent/kubernetes/host_setup/



Building on the Basics

Going further with Integrations, Logs, and APM

Integrations in K8s

Getting Into the Middleware

Two Approaches to Integration

Mount a Local Directory

- Must deploy configuration to every node
- Have to manage configurations and track changes across nodes
- Config management can ease deployment.

Use Config Maps

- Manage configurations in Kubernetes configs
- Easy to track configurations
- Change configurations quickly and across-the cluster

<https://docs.datadoghq.com/agent/kubernetes/integrations/>

Live Container Monitoring

Monitoring Is All About Good Process

Setting Up Live Containers

Live Container Configuration

```
(...)  
  env:  
    (...)  
      - name: DD_PROCESS_AGENT_ENABLED  
        value: "true"  
  volumeMounts:  
    (...)  
      - name: passwd  
        mountPath: /etc/passwd  
        readOnly: true  
  volumes:  
    (...)  
      - hostPath:  
          path: /etc/passwd  
          name: passwd  
(...)
```



You'll get live processes
on your nodes too!

Processes

Containers

Search container...

kube_service

[Hide Controls](#) | Showing 1 - 2 of 2 matching containers

Pod

Search 2 pod_names

 k8satlantademo-f8f8fff59-f6dlx k8satlantademo-f8f8fff59-fl8xd

Deployment

Search 1 kube_deployment

 k8satlantademo

Service

Search 1 kube_service

 k8satlantademo

Namespace

k8satlantademo_k8satlantademo-f8f8fff59-f6dlx

HOST	START	CONTAINER NAME	CONTAINER IMAGE	
gke-k8satlantademo-def...	2 hours ago	k8satlantademo_k8satla...	...fetch-171516/fetch-y-fetch...	
TAGS				
container_id:e4b087f7e7ac	pod_name:k8satlantademo-f8f8fff59-f6dlx	host:gke-k8satlantademo-default-pool-02726fe4-9jv8.c.fetch-171516.internal	app:fetch	
cluster-location:us-central1-a	...			
COMMAND	PID	PPID	CPU %	RSS MEMORY
sh ddtrace-run python app.py -c DATADOG_ENV=demo --statsd-host dd-agent:8125	24279	24264	0 %	684 KB
└─ python app.py --statsd-host dd-agent:8125	24316	24279	66 %	726.23 MB

Total CPU % ▾ 15min 56 %

(63 %) RSS Memory ▾ 15min 681.6M B (754.3M B)



What you should see.

Logs in K8s

All Your Logs Are Belong To Us

Setting Up Log Collection

Basic Logs Configuration

```
(...)
env:
  (...)

  - name: DD_LOGS_ENABLED
    value: "true"
  - name: DD_LOGS_CONFIG_CONTAINER_COLLECT_ALL
    value: "true"
(...)
```

Tracking Our Read Location

```
(...)
volumeMounts:
  (...)

  - name: pointerdir
    mountPath: /opt/datadog-agent/run
(...)

volumes:
  (...)

  - hostPath:
      path: /opt/datadog-agent/run
      name: pointerdir
(...)
```

The left side of the interface features a histogram at the top with a y-axis from 0 to 250 and an x-axis from 13:19 to 13:25. Below it is a search bar labeled "Source: mongodb". A sidebar on the right contains filters: "Filters (25)" with a pencil icon, "Saved Views", and sections for "CORE", "Source" (with checkboxes for "fetch", "nginx", "docker", and "mongodb" which is checked), "Host", and "Service" (with a value of 16.59K).

Hide Controls | 21,897 results found

DATE ↓	HOST
Nov 13 13:33:41.465	gke-k8s-live-demo-d...
Nov 13 13:33:41.200	gke-k8satlantademo...
Nov 13 13:33:40.763	gke-k8s-live-demo-d...
Nov 13 13:33:40.727	gke-k8s-live-demo-d...
Nov 13 13:33:40.154	gke-k8s-live-demo-d...
Nov 13 13:33:39.959	gke-k8s-live-demo-d...
Nov 13 13:33:39.959	gke-k8s-live-demo-d...
Nov 13 13:33:39.959	gke-k8s-live-demo-d...

INFO Nov 13, 2018 at 13:33:39.061 (a few seconds ago) View in Context

HOST	SERVICE	SOURCE
gke-k8satlantademo-default-pool-02	k8satlantademo	mongodb
CONTAINER NAME	DOCKER IMAGE	POD NAME
k8s_mongo_mongo-2_default_6f73e...	mongo	mongo-2
TAGS		
kube_service:mongo source:mongodb service:k8satlantademo short_image:mongo image_tag:latest		
container_name:k8s_mongo_mongo-2_default_6f73e54a-e68b-11e8-95df-42010a800035_0 pod_name:mongo-2		
kube_namespace:default kube_container_name:mongo docker_image:mongo:latest ...		

2018-11-13T18:33:39.061+0000 I NETWORK [listener] connection accepted from 127.0.0.1:44764 #59268 (8 connections now open)

ATTRIBUTES

{
 db {

What you should see

APM in K8s

Getting Down To The Code



Setting Up APM and Trace Search

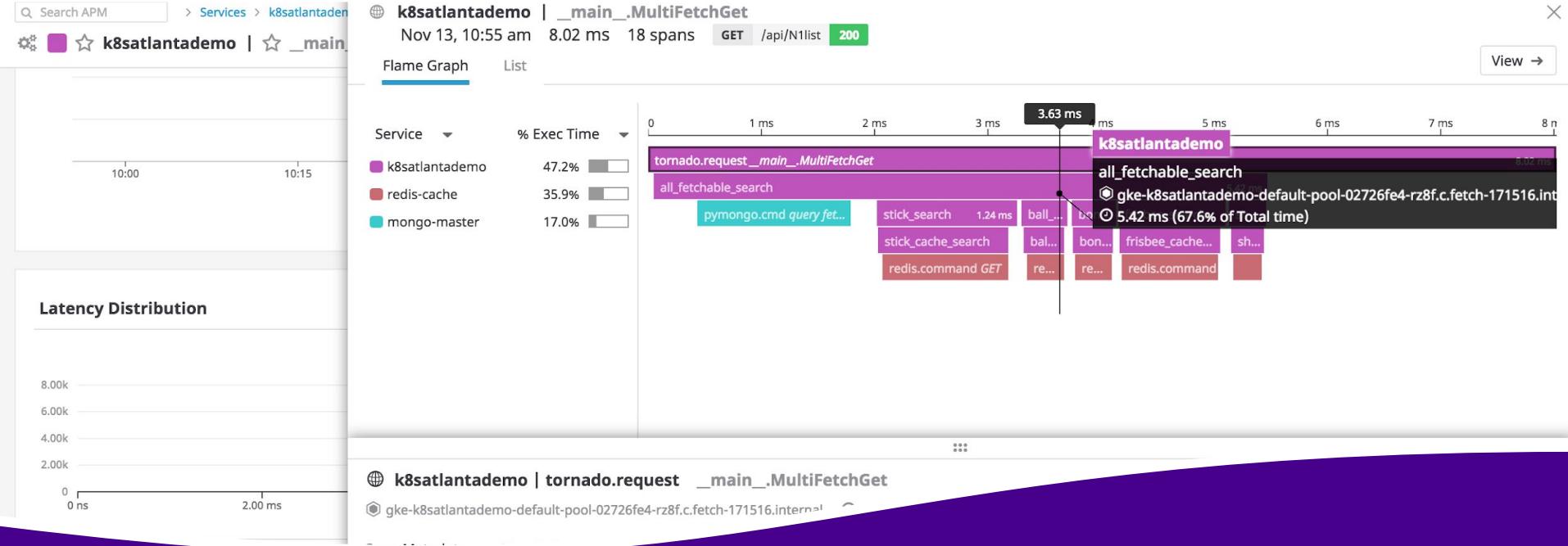
APM Configuration

```
(...)  
env:  
(...)  
- name: DD_APM_ENABLED  
  value: "true"  
(...)
```

Trace Search Configuration

```
(...)  
env:  
(...)  
- name: DD_APM_ANALYZED_SPANS  
  value: "YOURAPP|span.name=1"  
(...)
```

https://docs.datadoghq.com/agent/kubernetes/daemonset_setup/#trace-collection



What you should see.

Auto Discovery

Monitor Anything and Everything



Setting Up AutoDiscovery

Docker Labels

LABEL

```
"com.datadoghq.ad.check.names"="[<CHECK NAME>]"
```

LABEL

```
"com.datadoghq.ad.init.configs"="[<INIT CONFIG>]"
```

LABEL

```
"com.datadoghq.ad.instances"="[<INSTANCE CONFIG>]"
```

```
LABEL "com.datadoghq.ad.logs"="[<LOGS CONFIG>]"
```

Pod Annotations

annotations:

```
ad.datadoghq.com/<container identifier>.check.names: '[<CHECK NAME>]'  
ad.datadoghq.com/<container identifier>.init.configs: '[<INIT CONFIG>]'  
ad.datadoghq.com/<container identifier>.instances: '[<INSTANCE CONFIG>]'  
ad.datadoghq.com/<container identifier>.logs: '[<LOG CONFIG>]'
```

For Annotations, Autodiscovery identifies containers by *name*, NOT image (as it does for auto-conf files and key-value stores). That is, it looks to match <container identifier> to .spec.containers[0].name **not** .spec.containers[0].image

<https://docs.datadoghq.com/agent/autodiscovery/?tab=docker#template-source-kubernetes-pod-annotations>

AutoDiscovery with a Key/Value Store

Environment Variables

```
/datadog/  
  check configs/  
    docker image 1/ # container identifier,  
      - check names: [<CHECK NAME>]  
      - init configs: [<INIT CONFIG>]  
      - instances: [<INSTANCE CONFIG>]
```

etcd, Consul, Zookeeper

```
etcdctl mkdir /datadog/check configs/httpd  
  
etcdctl set /datadog/check configs/httpd/check names '[ "apache" ]'  
  
etcdctl set /datadog/check configs/httpd/init configs '[ {} ]'  
  
etcdctl set /datadog/check configs/httpd/instances  
'[{"apache_status_url":  
"http://%%host%%/server-status?auto"}]'
```

<https://docs.datadoghq.com/agent/autodiscovery/?tab=docker#template-source-key-value-store>



Demo Time

Let's see what this looks like in the wild...

Going further...

Advanced scaling and K8s cluster monitoring with Datadog

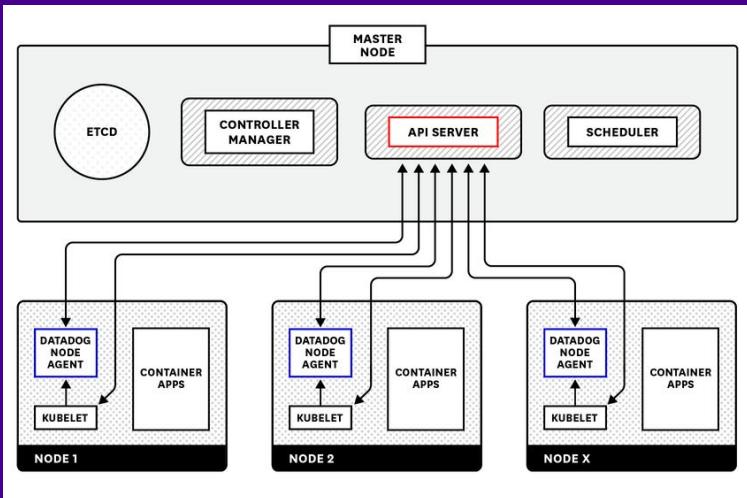


Datadog Cluster Agent

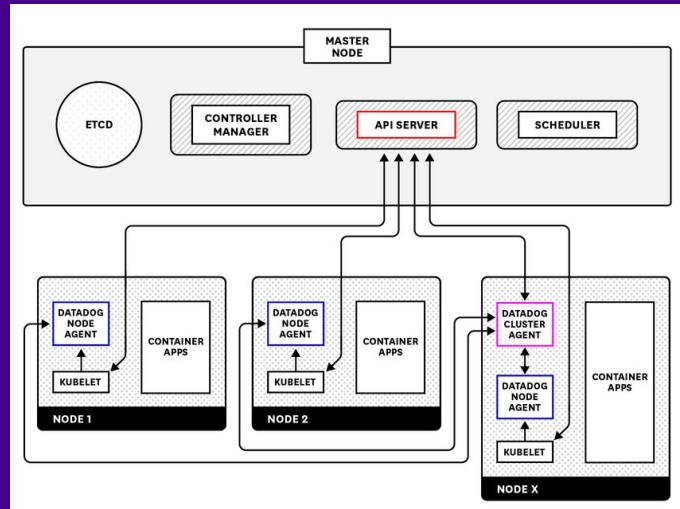
Scaling up to thousands of nodes gracefully...

Cluster Agent Concepts

Simple Deployment



With the Cluster Agent



<https://docs.datadoghq.com/agent/kubernetes/cluster/>

The background features a stylized illustration of several ships of different sizes and types sailing on a blue ocean under a sunset sky with orange and pink hues. One large ship on the right has the identifier "NETNX-265" on its side. Another ship in the foreground has "REDTS-606" on its side. A small boat with a steering wheel icon is also visible.

Autoscale With Any Metric

Whatever... I'll scale how I want!

Set Up Horizontal Pod Scaling

*Use the Datadog Cluster Agent
as an External Metrics Provider!*

Scale your services using any Datadog metric...

<https://www.datadoghq.com/blog/autoscale-kubernetes-datadog/>



Questions?

Thank you!



Mike Moore

michael.moore@datadoghq.com

Find this presentation and related code at:

<https://dtdg.co/ddk8s>



webofmike.com



[themsquared](#)



@themsquared