

# Orquestração de Containers

**Registro de eventos  
e monitoramento**

# Tópicos abordados

- Visualizando eventos
- Patterns para coleta de logs
- Monitoramento no kubernetes
- Monitoramento: exemplos

Invariavelmente, irão ocorrer erros e problemas durante a operação do *cluster*

Dominar as técnicas de visualização de eventos em containers e *pods* é fundamental

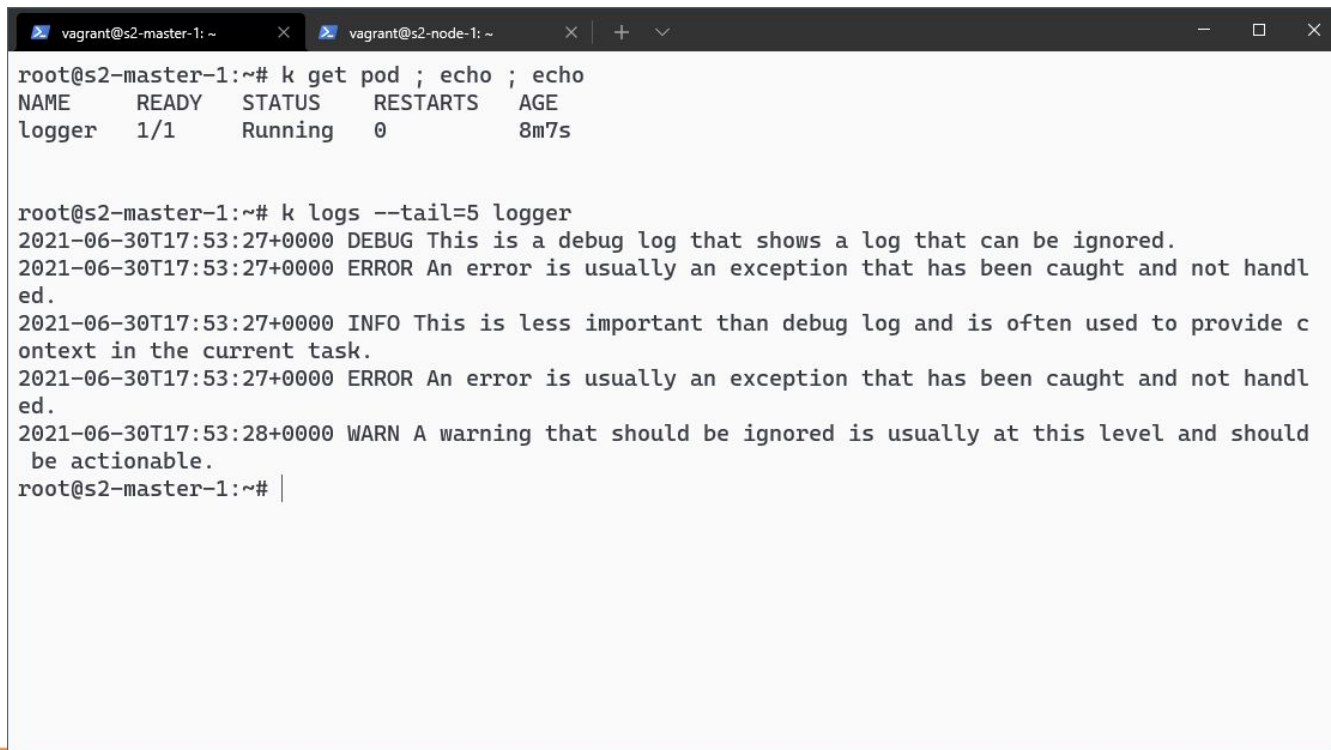
Etapa básica do processo de *troubleshooting*

Coleta de eventos de forma centralizada é o próximo passo de maturidade

## Visualizando eventos do *Kubernetes*

```
vagrant@s2-master-1: ~  
vagrant@s2-node-1: ~  
root@s2-master-1:~# k get events -n default | tail -n10  
21h          Normal      Started          pod/test-69f77cc749-tn8np    Started container ngi  
nx  
20m          Normal      SandboxChanged   pod/test-69f77cc749-tn8np    Pod sandbox changed,  
it will be killed and re-created.  
19m          Normal      Pulled          pod/test-69f77cc749-tn8np    Container image "ngin  
x:alpine" already present on machine  
19m          Normal      Created         pod/test-69f77cc749-tn8np    Created container ngi  
nx  
19m          Normal      Started         pod/test-69f77cc749-tn8np    Started container ngi  
nx  
17m          Normal      Killing         pod/test-69f77cc749-tn8np    Stopping container ng  
inx  
21h          Normal      SuccessfulCreate replicaset/test-69f77cc749    Created pod: test-69f  
77cc749-jfbvw  
21h          Normal      SuccessfulCreate replicaset/test-69f77cc749    Created pod: test-69f  
77cc749-krlbv  
21h          Normal      SuccessfulCreate replicaset/test-69f77cc749    Created pod: test-69f  
77cc749-tn8np  
21h          Normal      ScalingReplicaSet deployment/test              Scaled up replica set  
test-69f77cc749 to 3  
root@s2-master-1:~# |
```

## Visualizando eventos em *PODs*



```
vagrant@s2-master-1: ~  
vagrant@s2-node-1: ~  
root@s2-master-1:~# k get pod ; echo ; echo  
NAME      READY   STATUS    RESTARTS   AGE  
logger    1/1     Running   0           8m7s  
  
root@s2-master-1:~# k logs --tail=5 logger  
2021-06-30T17:53:27+0000 DEBUG This is a debug log that shows a log that can be ignored.  
2021-06-30T17:53:27+0000 ERROR An error is usually an exception that has been caught and not handled.  
2021-06-30T17:53:27+0000 INFO This is less important than debug log and is often used to provide context in the current task.  
2021-06-30T17:53:27+0000 ERROR An error is usually an exception that has been caught and not handled.  
2021-06-30T17:53:28+0000 WARN A warning that should be ignored is usually at this level and should be actionable.  
root@s2-master-1:~# |
```

## Visualizando eventos em PODs

E se eu tiver  
mais de um  
container?



```
root@s2-master-1:~# k apply -f manifests/multi-container-pod.yaml
pod/multi-container-pod created
root@s2-master-1:~# k get pods
NAME                READY   STATUS    RESTARTS   AGE
multi-container-pod  2/2     Running   0           12s
root@s2-master-1:~# k logs pods/multi-container-pod
Defaulted container "nginx-container" out of: nginx-container, alpine-container
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/08/01 01:39:19 [notice] 1#1: using the "epoll" event method
2023/08/01 01:39:19 [notice] 1#1: nginx/1.25.1
2023/08/01 01:39:19 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2023/08/01 01:39:19 [notice] 1#1: OS: Linux 5.10.0-21-amd64
2023/08/01 01:39:19 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/08/01 01:39:19 [notice] 1#1: start worker processes
2023/08/01 01:39:19 [notice] 1#1: start worker process 29
```



## Visualizando eventos em PODs

E se eu tiver mais de um container?



```
root@s2-master-1:~# k logs pods/multi-container-pod --container nginx-container
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/08/01 01:39:19 [notice] 1#1: using the "epoll" event method
2023/08/01 01:39:19 [notice] 1#1: nginx/1.25.1
2023/08/01 01:39:19 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2023/08/01 01:39:19 [notice] 1#1: OS: Linux 5.10.0-21-amd64
2023/08/01 01:39:19 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/08/01 01:39:19 [notice] 1#1: start worker processes
2023/08/01 01:39:19 [notice] 1#1: start worker process 29
root@s2-master-1:~#
root@s2-master-1:~# k logs pods/multi-container-pod --container alpine-container
done
done
done
done
```

## Visualizando eventos em pods

E se eu tiver  
múltiplas  
réplicas?



```
root@s2-master-1:~# k apply -f manifests/deploy-nginx.yaml
deployment.apps/deploy-nginx created
root@s2-master-1:~# k get pods
```

NAME	READY	STATUS	RESTARTS	AGE
deploy-nginx-567c687884-dzdp	0/1	ContainerCreating	0	3s
deploy-nginx-567c687884-gz944	0/1	ContainerCreating	0	3s
deploy-nginx-567c687884-vkcj6	0/1	ContainerCreating	0	3s

```
root@s2-master-1:~# k logs deployment/deploy-nginx
Found 3 pods, using pod/deploy-nginx-567c687884-dzdp
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/08/01 01:47:40 [notice] 1#1: using the "epoll" event method
2023/08/01 01:47:40 [notice] 1#1: nginx/1.25.1
2023/08/01 01:47:40 [notice] 1#1: built by gcc 12.2.1 20220924 (Alpine 12.2.1_git20220924-r4)
2023/08/01 01:47:40 [notice] 1#1: OS: Linux 5.10.0-21-amd64
2023/08/01 01:47:40 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/08/01 01:47:40 [notice] 1#1: start worker processes
2023/08/01 01:47:40 [notice] 1#1: start worker process 30
```



## Visualizando eventos em pods

E se eu tiver  
múltiplas  
réplicas?



Use stern!

```
root@s2-master-1:~# stern deployment/deploy-nginx
+ deploy-nginx-567c687884-gz944 > deploy-nginx
+ deploy-nginx-567c687884-dzdpr > deploy-nginx
+ deploy-nginx-567c687884-vkcj6 > deploy-nginx
deploy-nginx-567c687884-gz944 deploy-nginx /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, w
deploy-nginx-567c687884-gz944 deploy-nginx /docker-entrypoint.sh: Looking for shell scripts in /docker-
deploy-nginx-567c687884-gz944 deploy-nginx /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-lis
deploy-nginx-567c687884-vkcj6 deploy-nginx /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, w
deploy-nginx-567c687884-gz944 deploy-nginx 10-listen-on-ipv6-by-default.sh: info: Getting the checksum
deploy-nginx-567c687884-vkcj6 deploy-nginx /docker-entrypoint.sh: Looking for shell scripts in /docker-
deploy-nginx-567c687884-gz944 deploy-nginx 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPV
deploy-nginx-567c687884-dzdpr deploy-nginx /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, w
deploy-nginx-567c687884-dzdpr deploy-nginx /docker-entrypoint.sh: Looking for shell scripts in /docker-
deploy-nginx-567c687884-vkcj6 deploy-nginx /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-lis
deploy-nginx-567c687884-vkcj6 deploy-nginx 10-listen-on-ipv6-by-default.sh: info: Getting the checksum
```



E se o *kube-apiserver* estiver indisponível,  
ou o *control plane* estiver fora do ar?

```
root@s2-master-1:~# k get pods
The connection to the server 192.168.68.20:6443 was refused - did you specify the right host or port?
```

```
root@s2-master-1:~# docker ps -a | grep -i apiserver
root@s2-master-1:~# ls -l /etc/kubernetes/manifests/
total 12
-rw----- 1 root root 2409 Jul 12 12:51 etcd.yaml
-rw----- 1 root root 3392 Jul 12 12:51 kube-controller-manager.yaml
-rw----- 1 root root 1463 Jul 12 12:51 kube-scheduler.yaml
root@s2-master-1:~# mv kube-apiserver.yaml /etc/kubernetes/manifests/
root@s2-master-1:~# k get pods
```

NAME	READY	STATUS	RESTARTS	AGE
deploy-nginx-567c687884-dzdpr	1/1	Running	0	22m
deploy-nginx-567c687884-gz944	1/1	Running	0	22m
deploy-nginx-567c687884-vkcj6	1/1	Running	0	22m

## Uma integração melhor: *crictl*

Instalação e documentação disponíveis em:  
<https://kubernetes.io/docs/tasks/debug-application-cluster/crictl/>

```
root@s2-master-1:~# crictl --runtime-endpoint /var/run/cri-dockerd.sock ps
I0801 02:28:35.276779 213356 util_unix.go:103] "Using this endpoint is deprecated, please consider using full URL format" endpoint="/var/run/cri-dockerd.sock"
URL="unix:///var/run/cri-dockerd.sock"
I0801 02:28:35.304890 213356 util_unix.go:103] "Using this endpoint is deprecated, please consider using full URL format" endpoint="/var/run/cri-dockerd.sock"
URL="unix:///var/run/cri-dockerd.sock"
```

CONTAINER	IMAGE	CREATED	STATE	NAME	ATTEMPT	POD ID	POD
06e2a84571476 -s2-master-1	08a0c939e61b7	18 minutes ago	Running	kube-apiserver	0	18a9d901b5553	kube-apiserver
3eb532001845a ntrollers-674fff74c8-bnj5z	212faac284a2e	19 minutes ago	Running	calico-kube-controllers	10	40e18dd0c5789	calico-kube-co
e17b547691b90 r-manager-s2-master-1	7cffc01dba0e1	25 minutes ago	Running	kube-controller-manager	19	8826569f6985f	kube-controlle
6ad22dd6324e7 -s2-master-1	41697ceeb70b3	25 minutes ago	Running	kube-scheduler	3	07cdd5968fd61	kube-scheduler
0eb365ba56b3d 869d-v2sch	ead0a4a53df89	3 hours ago	Running	coredns	5	19563fa75d42e	coredns-5d78c9
a16732eb55f25 kq2	cae61b85e9b45	3 hours ago	Running	calico-node	5	1b8e906347e78	calico-node-zs
dd99fd7e1e964 z7	5780543258cf0	3 hours ago	Running	kube-proxy	5	9c72f1c6c24d6	kube-proxy-6gl
5c650c61c2e50 -1	86b6af7dd652c	3 hours ago	Running	etcd	5	3d19a950d601a	etcd-s2-master

## Visualizando eventos via *crictl*

```
vagrant@s2-master-1: ~  
vagrant@s2-node-1: ~  
root@s2-master-1:~# crictl logs --tail=10 ac2856d4648d8  
I0630 17:44:54.901699      1 shared_informer.go:247] Caches are synced for disruption  
I0630 17:44:54.902290      1 disruption.go:371] Sending events to api server.  
I0630 17:44:54.907916      1 shared_informer.go:247] Caches are synced for resource quota  
I0630 17:44:55.264449      1 shared_informer.go:247] Caches are synced for garbage collector  
I0630 17:44:55.289131      1 shared_informer.go:247] Caches are synced for garbage collector  
I0630 17:44:55.289218      1 garbagecollector.go:151] Garbage collector: all resource monitors ha  
ve synced. Proceeding to collect garbage  
E0630 18:02:54.418687      1 leaderelection.go:325] error retrieving resource lock kube-system/ku  
be-controller-manager: Get "https://192.168.68.20:6443/apis/coordination.k8s.io/v1/namespaces/kube  
-system/leases/kube-controller-manager?timeout=5s": context deadline exceeded  
E0630 18:17:16.331757      1 leaderelection.go:325] error retrieving resource lock kube-system/ku  
be-controller-manager: Get "https://192.168.68.20:6443/apis/coordination.k8s.io/v1/namespaces/kube  
-system/leases/kube-controller-manager?timeout=5s": context deadline exceeded  
E0630 18:30:01.488100      1 leaderelection.go:325] error retrieving resource lock kube-system/ku  
be-controller-manager: Get "https://192.168.68.20:6443/apis/coordination.k8s.io/v1/namespaces/kube  
-system/leases/kube-controller-manager?timeout=5s": context deadline exceeded  
E0630 18:31:03.787571      1 leaderelection.go:325] error retrieving resource lock kube-system/ku  
be-controller-manager: Get "https://192.168.68.20:6443/apis/coordination.k8s.io/v1/namespaces/kube  
-system/leases/kube-controller-manager?timeout=5s": context deadline exceeded  
root@s2-master-1:~#
```



## Uma boa solução, mas com limitações

Certamente, logs enviados pelos containers para *stdout* e *stderr* são úteis

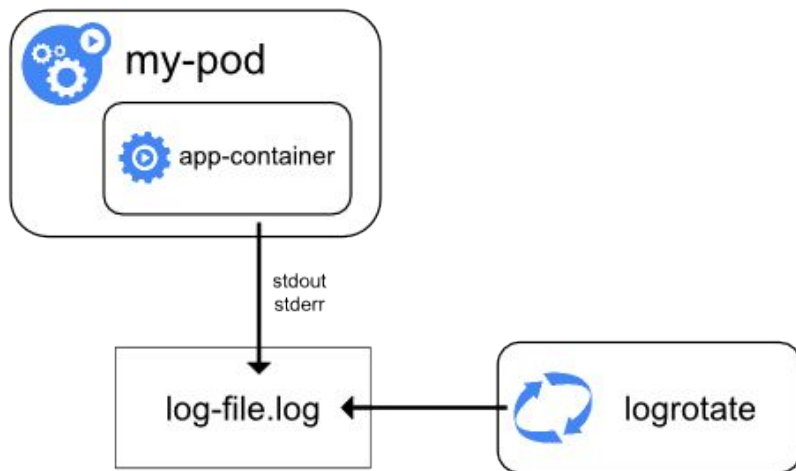
Contudo, não são suficientes: e se o container/pod/node tiverem um *crash*?

Num cluster, logs devem ter ciclo de vida e armazenamento independentes

Esse conceito é conhecido como *cluster-level logging*

<https://kubernetes.io/docs/concepts/cluster-administration/logging/>

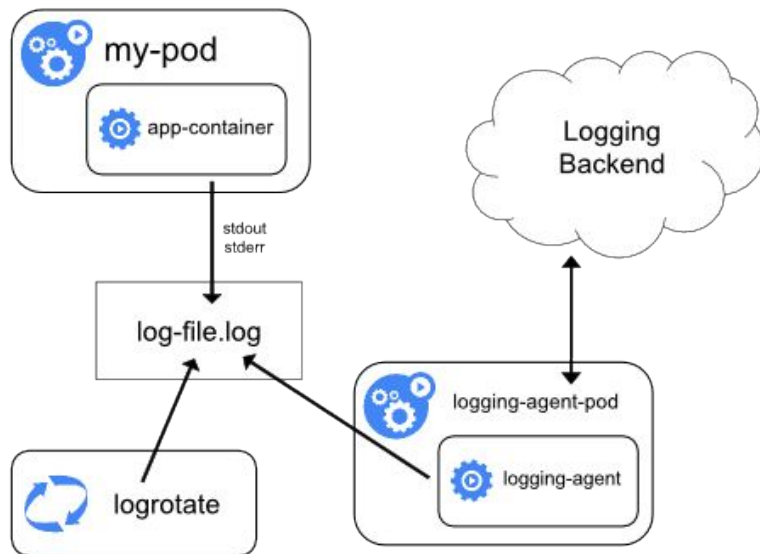
### Fácil, mas insuficiente: *logging no nível do node*



Precisamos verificar os logs  
em cada node:  
`/var/log/pods/`  
`/var/log/containers/`



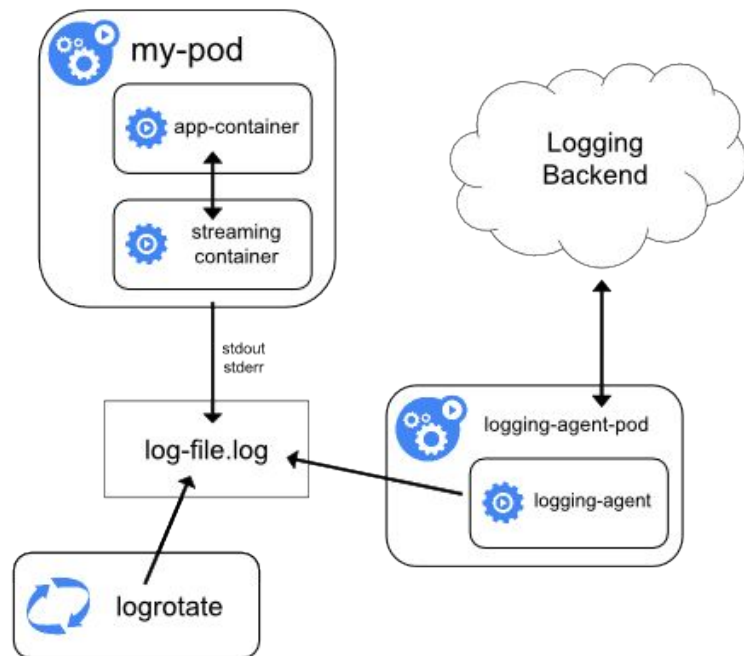
### Utilizando um agente de *logging* em cada *node*



Um agent roda em cada NODE implementado via *DaemonSet*

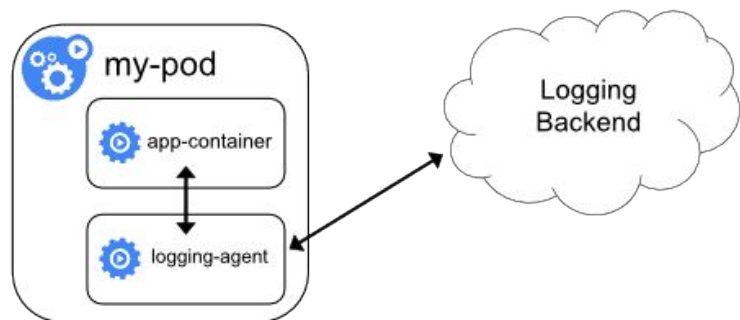
O logging-agent lê os logs do diretório de logs no NODE e envia para o backend

## Utilizando um streaming *sidecar* container



Permite a separação de *log streams*

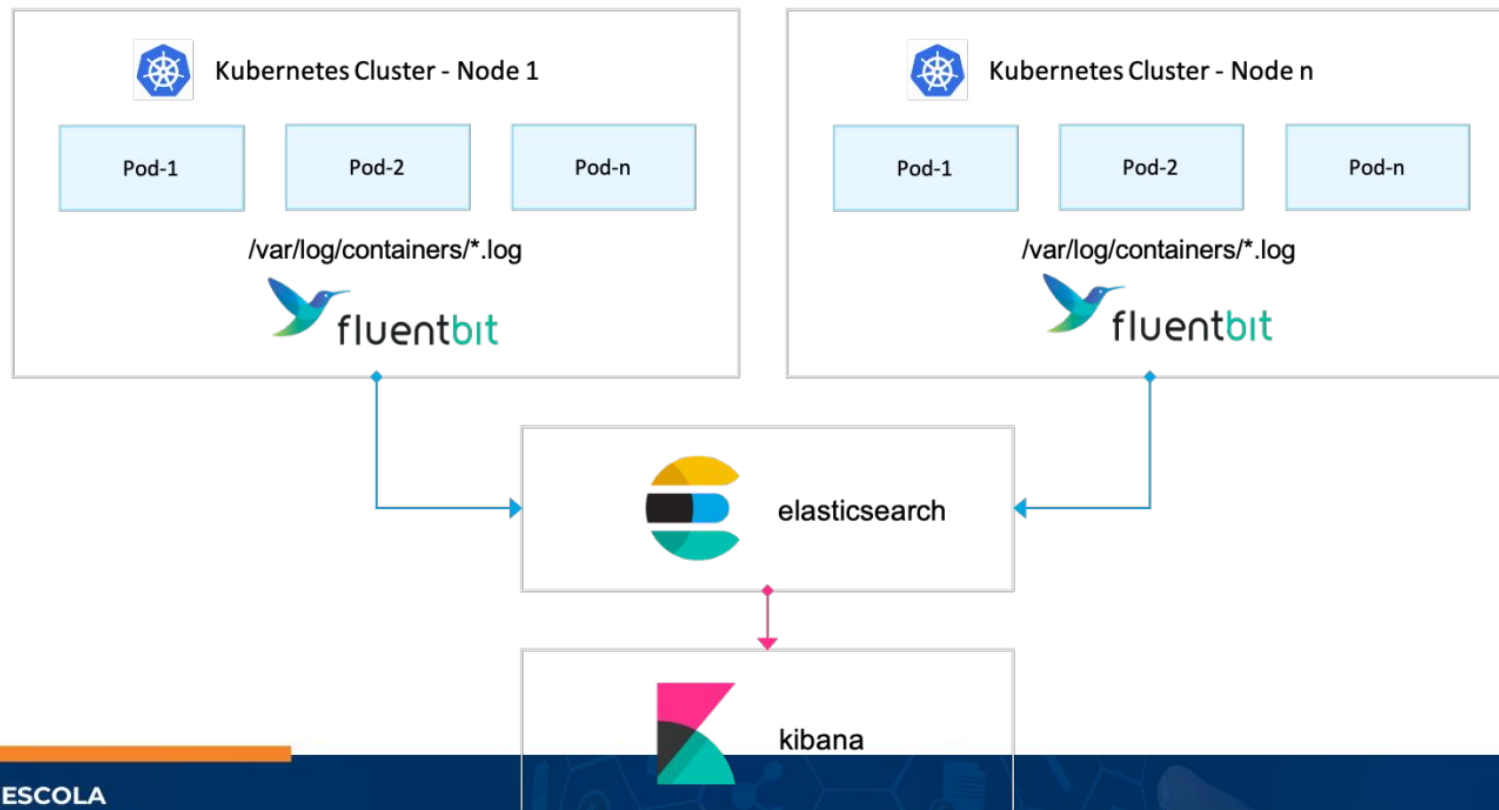
## Utilizando um *sidecar* com o agente de *logging*



Aumento no uso de recursos

Logs não são visíveis via  
*kubectl logs*

## Geralmente utilizado: agente de *logging* em *cada node*



Para escalar aplicações e ter um bom entendimento do ambiente, monitoramento é fundamental

Pode-se obter informações examinando containers, *pods*, serviços, e outros objetos

No Kubernetes, há dois *pipelines* de métricas para monitoramento

*Resource metrics*  
e  
*Full metrics*

## Resource metrics

*Pipeline com conjunto de métricas limitadas para alguns componentes do cluster*

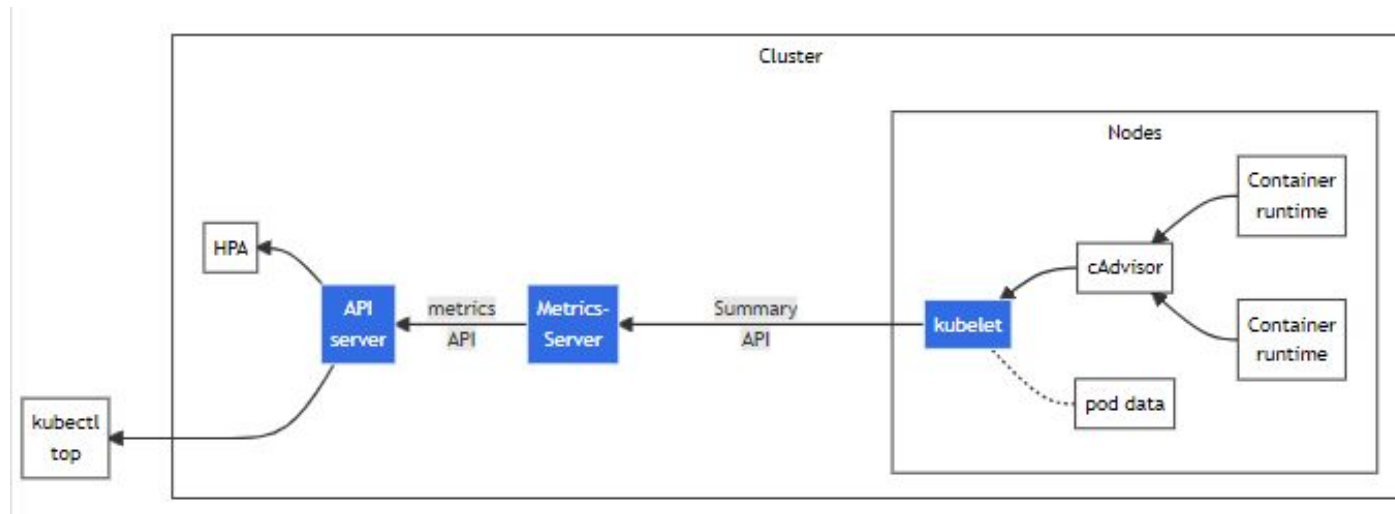
*Bons exemplos são o `kubectl top` e `Horizontal Pod Autoscaler`*

*Coletadas por servidor in-memory `metrics-server` e expostas via API `metrics.k8s.io`*

*Dados são coletados pelo `kubelet` a partir do CRI de cada *node**



## Resource metrics



<https://kubernetes.io/docs/tasks/debug/debug-cluster/resource-metrics-pipeline/>

## Instalação do *metrics-server*

```
# wget https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
```

adicione no Deployment, em args:  
--kubelet-insecure-tls

```
# kubectl apply -f components.yaml
```

```
# kubectl get deployment metrics-server -n kube-system
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
metrics-server	1/1	1	1	6m

```
# kubectl top node
```

<https://github.com/kubernetes-sigs/metrics-server>

## Full metrics

*Pipeline* que dá acesso a um conjunto mais rico de métricas

Também pode ser usado para auto-escalabilidade do *cluster*

Coletadas via *kubelet* e expostas via APIs *custom.metrics.k8s.io* ou *external.metrics.k8s.io*

Diversas soluções populares para visualização, como o **Prometheus**

<https://prometheus.io/>

## Algumas soluções populares

[Kubernetes Dashboard](#)

[k9s](#)

[Prometheus](#)

[cAdvisor](#)

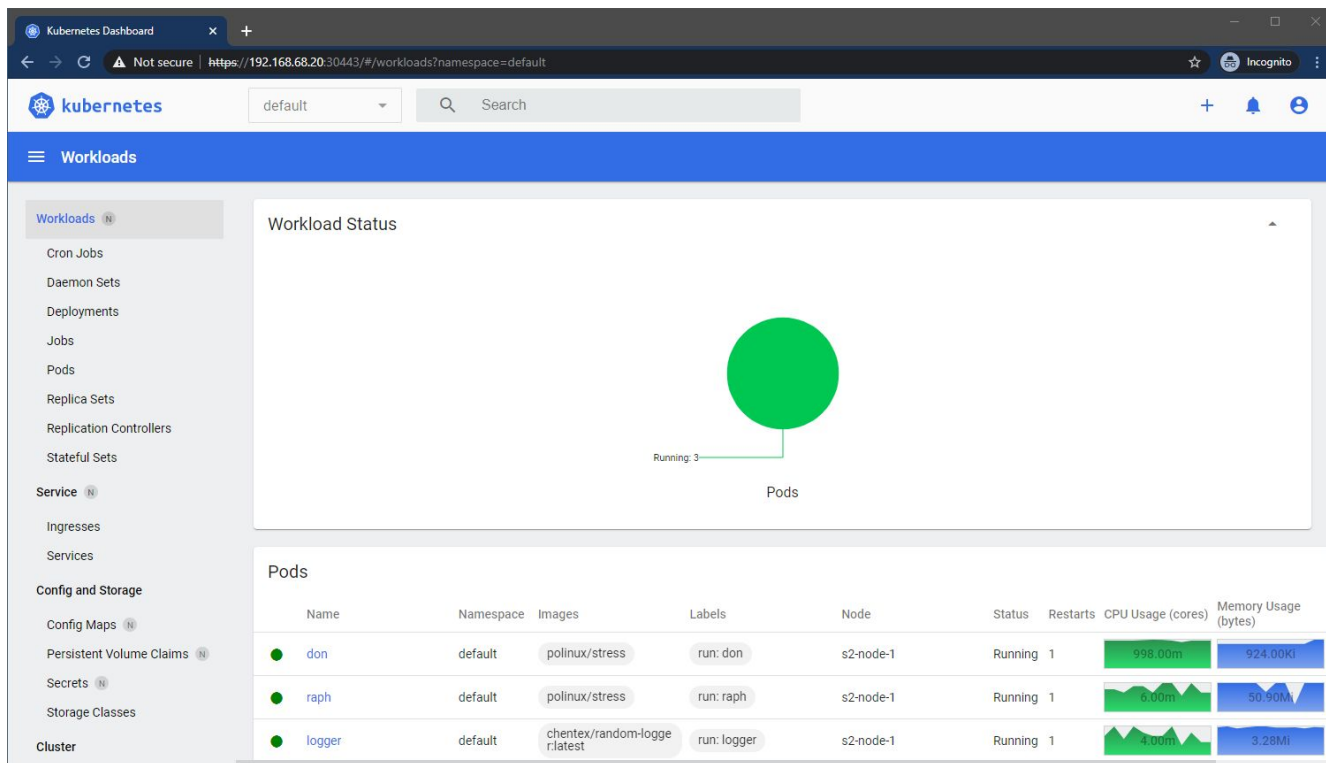
[Jaeger](#)

[Stack EFK](#)

[Datadog](#)

[New Relic](#)

## Kubernetes Dashboard



## Como operar com o Dashboard?

Vamos ver:

<https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/>  
<https://github.com/kubernetes/dashboard>



## Instalação kubernetes dashboard

kubectl apply -f

<https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml>

```
root@s2-master-1:~# k -n kubernetes-dashboard get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/dashboard-metrics-scraper-5cb4f4bb9c-ksgjk	1/1	Running	0	53s
pod/kubernetes-dashboard-6967859bff-fr2ld	1/1	Running	0	53s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/dashboard-metrics-scraper	ClusterIP	10.101.190.134	<none>	8000/TCP	53s
service/kubernetes-dashboard	ClusterIP	10.96.82.191	<none>	443/TCP	54s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/dashboard-metrics-scraper	1/1	1	1	53s
deployment.apps/kubernetes-dashboard	1/1	1	1	53s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/dashboard-metrics-scraper-5cb4f4bb9c	1	1	1	53s
replicaset.apps/kubernetes-dashboard-6967859bff	1	1	1	53s

## Instalação kubernetes dashboard

```
kubectl apply -f admin-user.yaml
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
```

# Instalação kubernetes dashboard

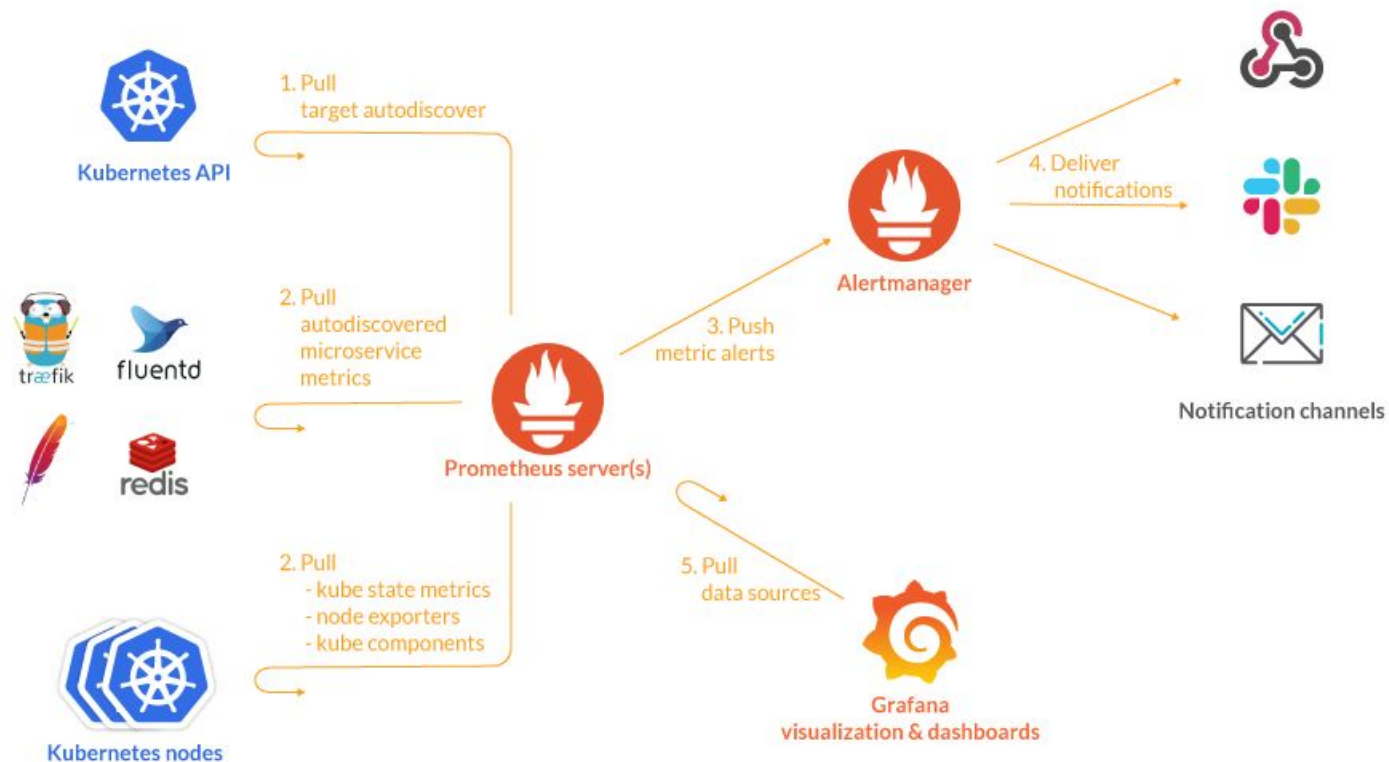
```
kubectl -n kubernetes-dashboard create token admin-user
```

```
root@s2-master-1:~# kubectl -n kuberneteshq create token admin-user eyJhbGciOiJIUzI1NiIsImtpZCI6IngyMkQ3eFdBOTY0OyUEZYdDcyOFN5aThVVDcXRKSzRyYjZlYW8xaEkifQ.eyJhdWQiOiJsiaHR0cHM6Ly9rdWJlcmlldGVzLmRlZmF1bHQuYXZlLnNsZXN0ZXIubG9jaWwuaXSwic2VudCkiOiJoanNkwODczMzk3LCJpYXQiOiJE20TA4Njk3OTcsImZscyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdmx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJuZXRlcy5pbypIeyJuYVwlcnBhY2UiOiJrdWJlcmlldGVzLWRhc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJhZG1lbGl1c2VyIiwidWlkIjoiazZyYWE4MzQtODRmNm00TtYyLWJlZDAZmIyZjc0YmE2MGUwIn19LCJuYmYiOiJE20TA4Njk3OTcsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDprdwJlcmlldGVzLWRhc2hib2FyZDphZG1lbGl1c2VyIn0.gz27ASUN0fuZilKA8YdAlUwcXomU56Q6QbkYOgWuxPTviFM1PZeIxOHAgKdM0N63InfGXsxWSZjXu_KSh0B2E7mbXSLMccsJ5GR-QoNP6K9wgBDpg0NKntx57_HuCR8-Et46ip7XB6ojPxhBOGoR83xgpNJL3cceG89efSxbQty4qfo4SSaDWEdyoGLKKqovZLV8yb4y3rYaru1c7N50T2noGYuIXUTG9KF-QL0y8wbkx-wCwpvBMLMbqlbOpsflavIQ5mgRTfvkJcgTezwCC9klwOmaFo8-8u2tmJ2cqZpTzkH24X4RUUpCPjNe6hvncnu9NMjzpzeJfCFB_0
```

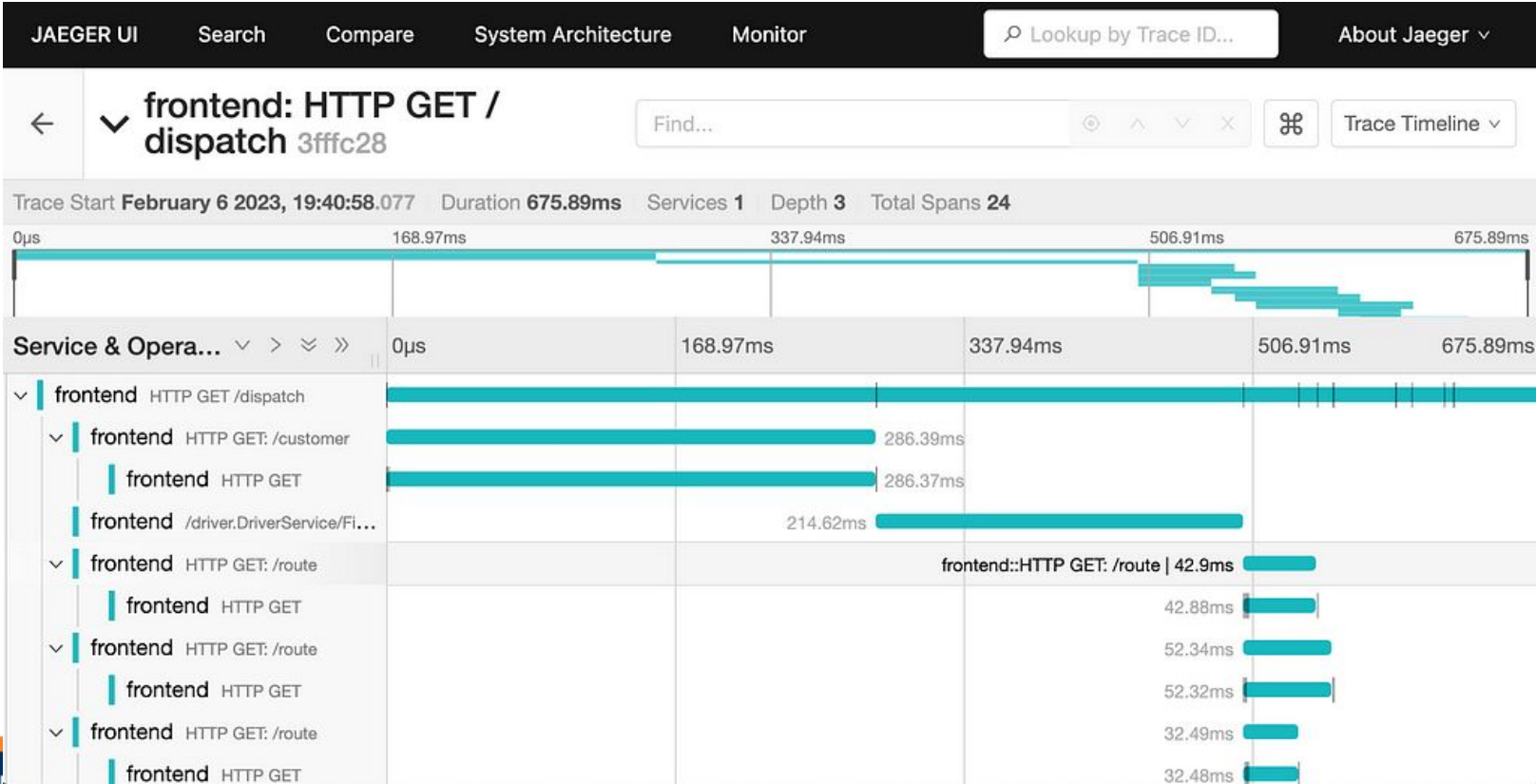
# Prometheus + Grafana



## Prometheus: arquitetura típica

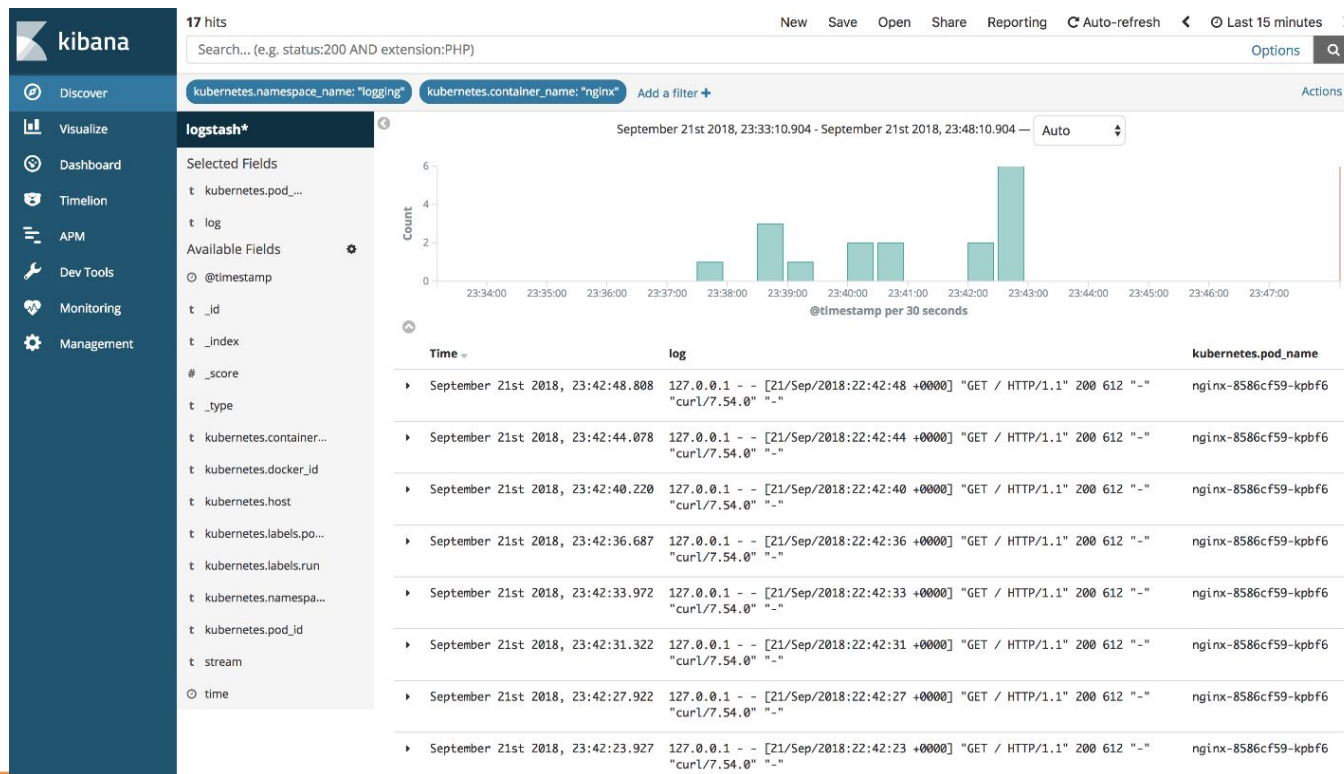


# Jaeger: rastreabilidade

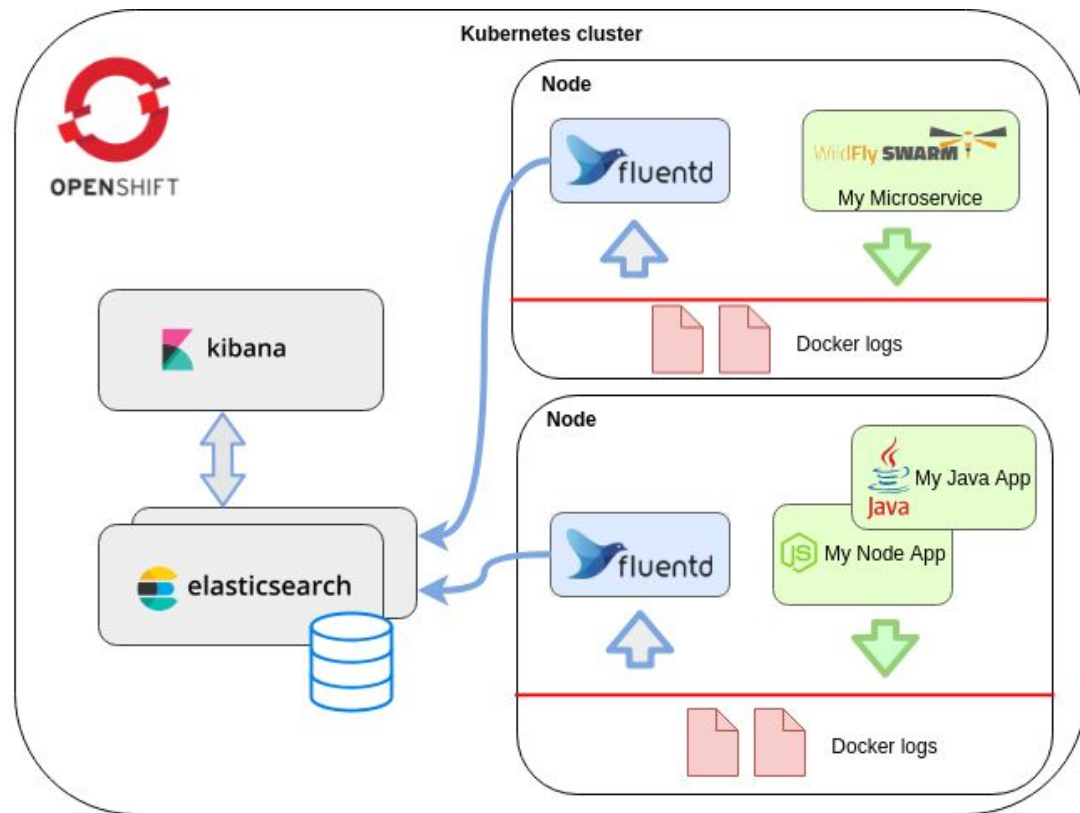




## EFK: dashboard



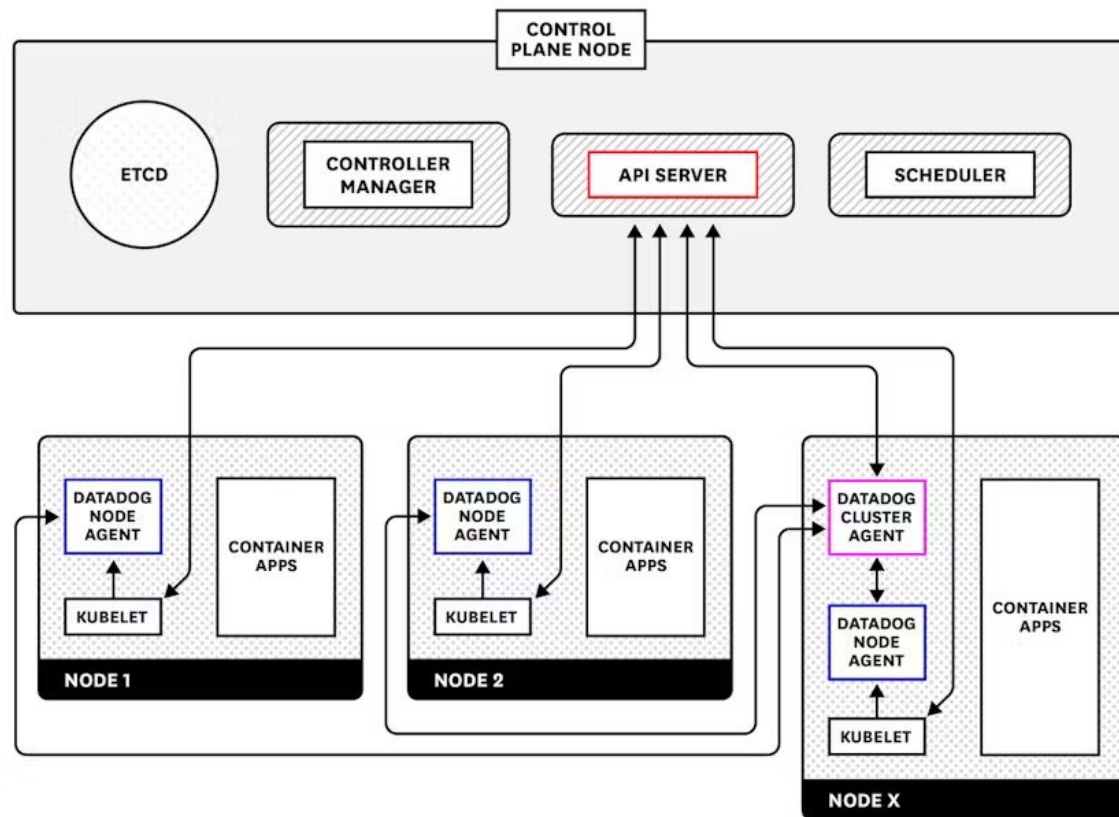
## EFK: arquitetura típica



## Datadog



# Datadog



# Tarefa 4

As atividades práticas desta sessão podem ser obtidas em formato HTML via:

<https://bit.ly/ads19-tarefas-s4>



ESCOLA  
SUPERIOR  
DE REDES

# Registro de eventos e monitoramento