OpenShift Monitoring and Logging

Definitions

Logging = collecting and analyzing **log** data

Monitoring = collecting and analyzing **metrics**

Note: both systems can, must and do support generating alerts

Logs

- Information from discrete events
- Unstructured text
  - may be formatted (syslog, common log, ...)
- Structured (e.g. json)
- Timestamped (?)

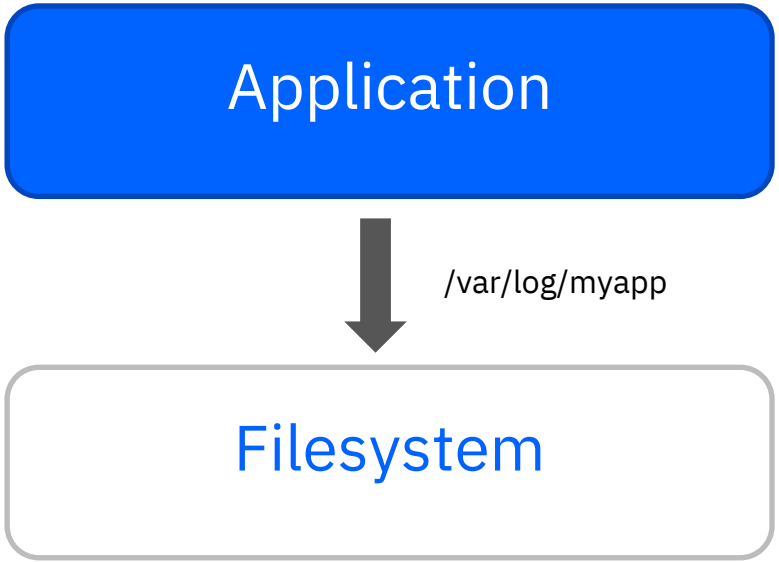
⇒ Text search, query + reduce

Metrics

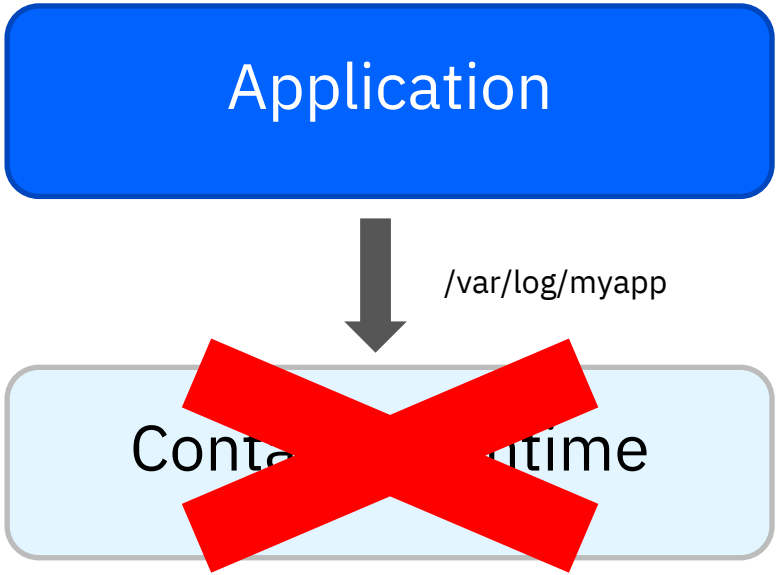
- Observables (CPU, memory, threads ...)
- Measured on specific intervals
  - every 10ms, 1s, 5mins ...
- Time series of data

⇒ Time-based graphs, gauges, charts

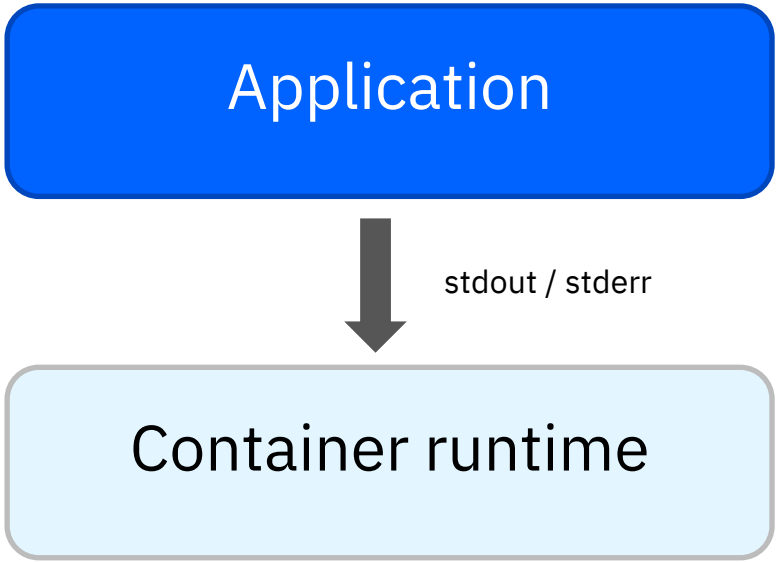
Application logging



Container logging -incorrect



Container logging 12-factor



Kubernetes logging

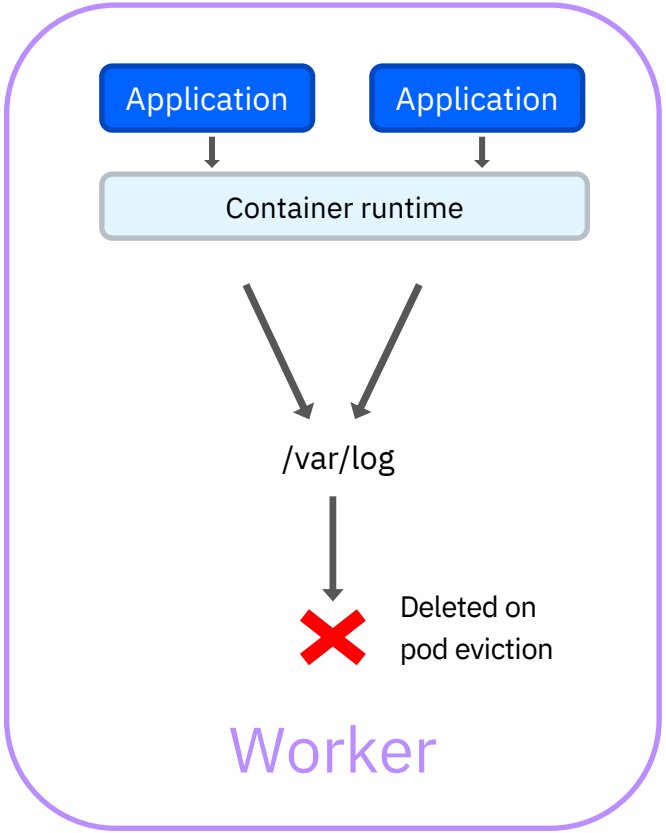
Scaling to a cluster of multiple applications

Application output – what is written to stdout / stderr

and...

Context in cluster:

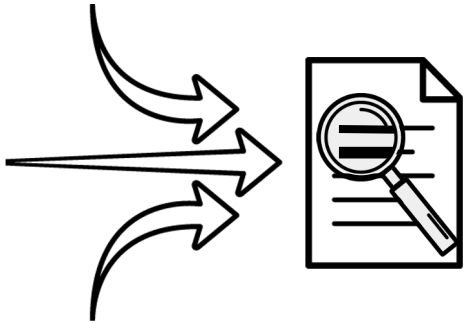
- Namespace
- Worker node
- Pod name
- Container in pod
- Labels




```
oc logs stockquote-1-41d88 –n mcsvcs-user001
```

Log management with EFK

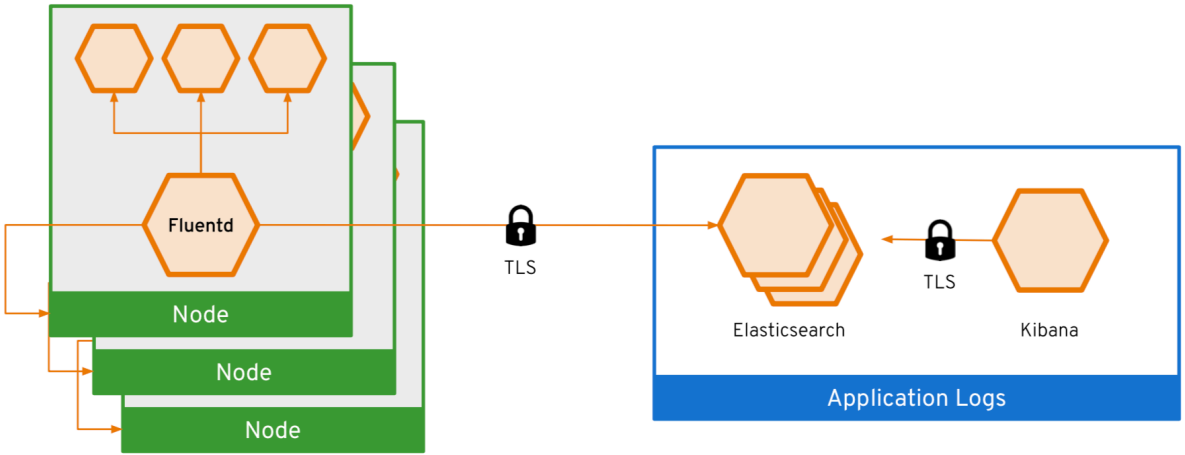
- EFK stack to aggregate logs for hosts and applications
  - **Elasticsearch:** search and analytics engine
  - **Fluentd:** gathers logs and sends to Elasticsearch
  - **Kibana:** web UI for Elasticsearch
- Access control
  - Cluster administrators can view all logs
  - Users can only view logs for their projects
- Ability to send logs elsewhere
  - External Elasticsearch, Splunk, etc



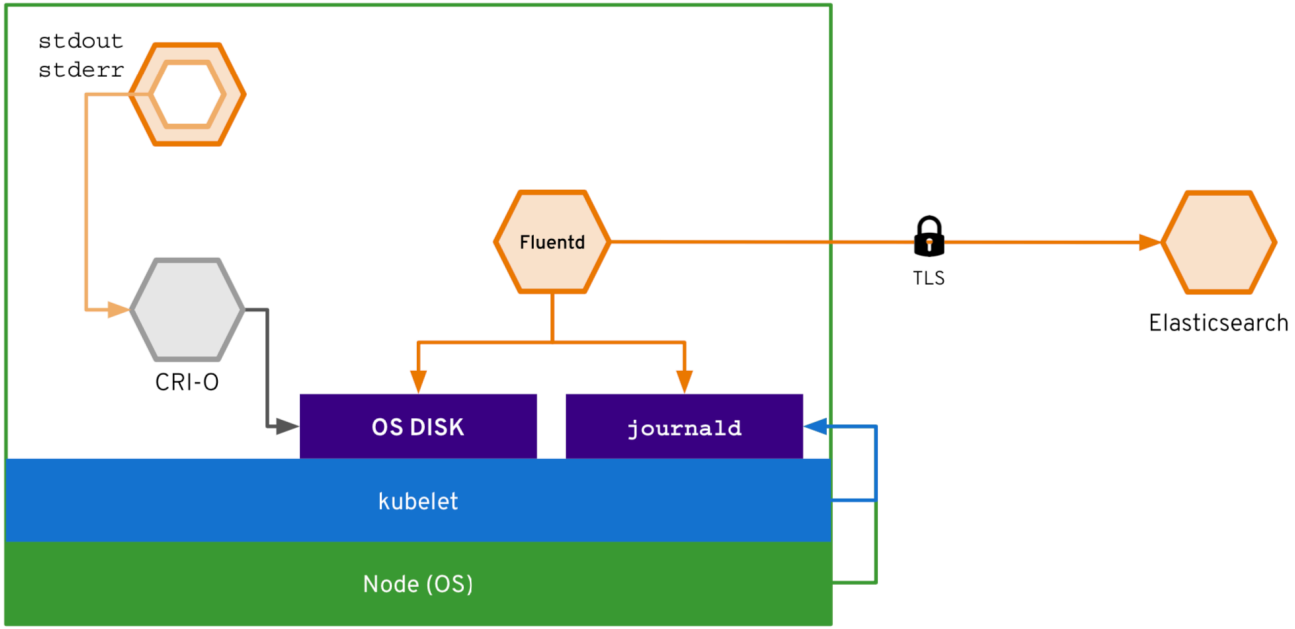
EFK is the default for OpenShift, other options:

- ElasticSearch, Logstash, Kibana (ELK)
- LogDNA – cloud log storage provider

EFK Log access



Log capture to Elastic



Aggregated logs

```
Nov 10 11:40:24 stockquote-1-4ld88 stockquote  Getting quote for stock IBM ...
Nov 10 11:40:24 stockquote-1-4ld88 stockquote  Calling API Connect with URL https://api.us-south.apiconnect.appdomain.cloud/ww-client-advocacy-workshop-dev/sb/stocks/IBM
Nov 10 11:40:25 stockquote-1-4ld88 stockquote  Quote returned from API Connect
Nov 10 11:40:25 stockquote-1-4ld88 stockquote  {"symbol":"IBM","date":"2019-11-08","time":1573246850446,"price":137.61}
Nov 10 11:40:25 stockquote-1-4ld88 stockquote  GET /stock-quote/IBM 200 567.753 ms - 72
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Watson initialization completed successfully!
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] SQL executeUpdate command completed successfully
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Updated IBM entry for Client2 in Stock table
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Adding IBM to portfolio for Client2
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Calling stock-quote microservice for MSFT
Nov 10 11:40:25 stockquote-1-4ld88 stockquote  Getting quote for stock MSFT ...
Nov 10 11:40:25 stockquote-1-4ld88 stockquote  Calling API Connect with URL https://api.us-south.apiconnect.appdomain.cloud/ww-client-advocacy-workshop-dev/sb/stocks/MSFT
Nov 10 11:40:25 stockquote-1-4ld88 stockquote  Quote returned from API Connect
Nov 10 11:40:25 stockquote-1-4ld88 stockquote  {"symbol":"MSFT","date":"2019-11-08","time":1573246800831,"price":145.96}
Nov 10 11:40:25 stockquote-1-4ld88 stockquote  GET /stock-quote/MSFT 200 608.844 ms - 73
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Watson initialization completed successfully!
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] SQL executeUpdate command completed successfully
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Updated MSFT entry for Client2 in Stock table
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Adding MSFT to portfolio for Client2
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Processed 2 stocks for Client2
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Releasing JDBC resources
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Released JDBC resources
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Watson initialization completed successfully!
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] SQL executeUpdate command completed successfully
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Returning {"owner": "Client2", "total": 2243.51, "loyalty": "BASIC", "balance": 30.0199999999999996, "commissions": 19.98, "free": 0, "nextCommission": 9.99,
    "sentiment": "Analytical", "stocks": {"IBM":{"symbol":"IBM","shares":11,"commission":9.99,"price":137.61,"total":1513.71,"date":"2019-11-08"},"MSFT":
    {"symbol":"MSFT","shares":5,"commission":9.99,"price":145.96,"total":729.8000000000001,"date":"2019-11-08"}}}
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Preparing to send a Kafka message
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Publishing to Kafka
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] TOPIC = stocktrader-user001
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] MESSAGE= {"id": "be6821b0-b59e-4a1a-bc93-fe7ae3320efe", "owner": "Client2", "symbol": "IBM", "shares": 6, "price": 137.61, "when": "2019-11-10 07:40:25.843",
    "commission": 0.0}
Nov 10 11:40:25 event-streams-consumer-1-4bgxs event-streams-consumer  Nov 10, 2019 7:40:25 PM com.ibm.hybrid.cloud.sample.stocktrader.eventstreamsconsumer.EventStreamsConsumer lambda$runConsumer$0
Nov 10 11:40:25 event-streams-consumer-1-4bgxs event-streams-consumer INFO INFO: Consumer Record:(stocktrader-user001, {"id": "be6821b0-b59e-4a1a-bc93-fe7ae3320efe", "owner": "Client2", "symbol": "IBM", "shares": 6, "pri
    137.61, "when": "2019-11-10 07:40:25.843", "commission": 0.0}, 0, 32)
Nov 10 11:40:25 event-streams-consumer-1-4bgxs event-streams-consumer  Nov 10, 2019 7:40:25 PM com.ibm.hybrid.cloud.sample.stocktrader.eventstreamsconsumer.EventStreamsConsumer insertStockPurchase
Nov 10 11:40:25 event-streams-consumer-1-4bgxs event-streams-consumer INFO INFO: In Mongo Connector insertStockPurchase
Nov 10 11:40:25 portfolio-1-2rlwd portfolio INFO [INFO    ] Delivered message to Kafka: {"id": "be6821b0-b59e-4a1a-bc93-fe7ae3320efe", "owner": "Client2", "symbol": "IBM", "shares": 6, "price": 137.61, "when": "2019-11-1
```

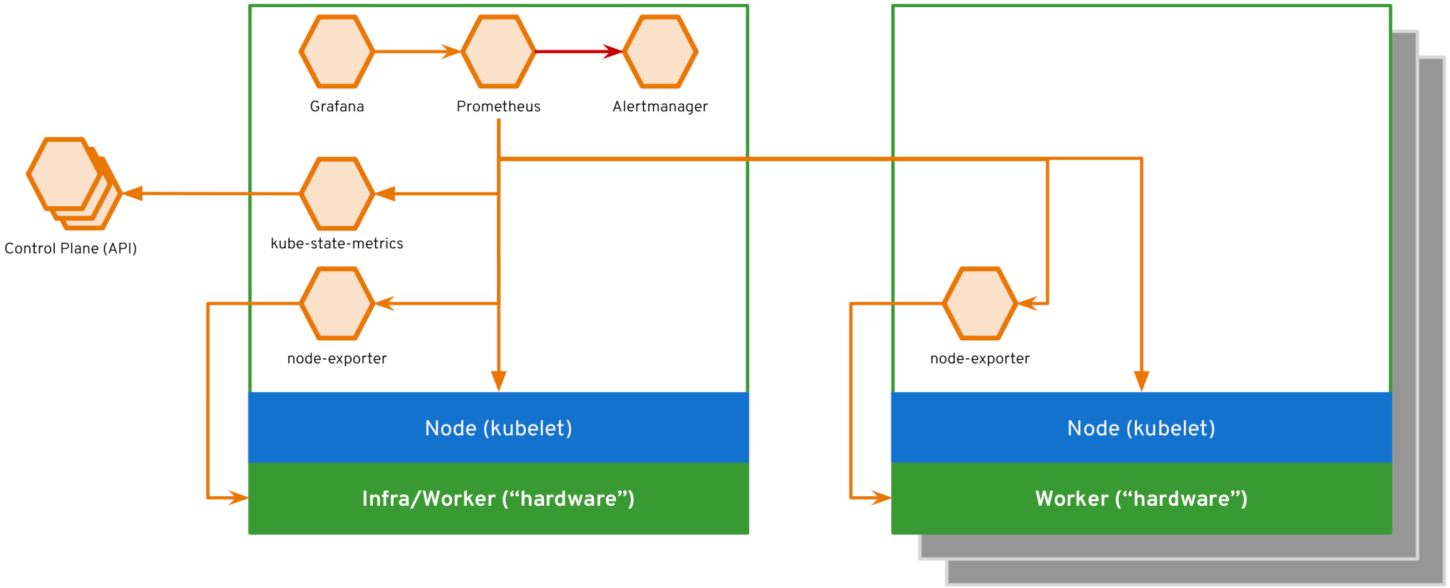
OpenShift Cluster Monitoring

Metrics collection and storage via Prometheus, an open-source monitoring system time series database.

Alerting/notification via Prometheus' Alertmanager, an open-source tool that handles alerts send by Prometheus.

Metrics visualization via Grafana, the leading metrics visualization technology.

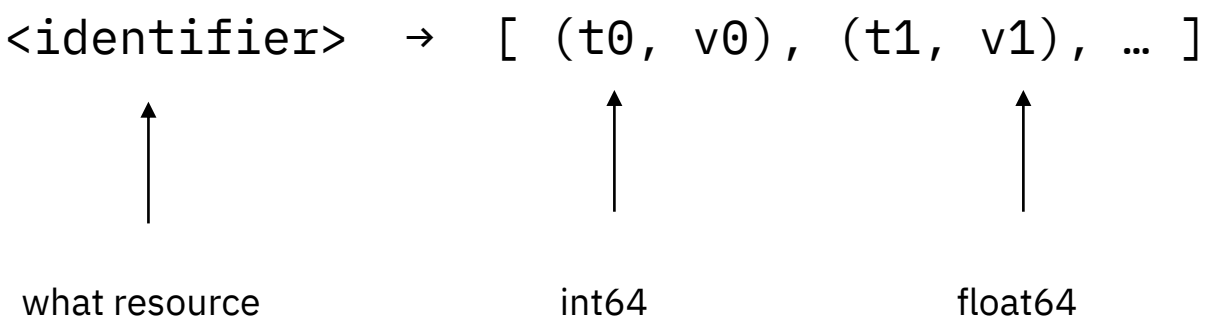
Architecture



Prometheus uses a “scraping” model – polling targets vs. push by agents

Prometheus Data Model

What is a time series?



Prometheus Data Model

Identifiers in a time series?

```
container_memory_usage_bytes{namespace="mcsvcs-user001",pod_name="trade-history-1-f9h5w"}
```

metric name

labels

- Flexible - could be anything
- No hierarchy
- Human readable

Prometheus Rules

Recording rules allow the definition of complex or frequently needed expressions

```
record: pod_name:container_cpu_usage:sum
expr: sum
  by(pod_name, namespace) (rate(container_cpu_usage_seconds_total{container_name!="",container_name!="POD",pod_name!=""}[5m]))
```

Prometheus Querying

How many pods in namespace mcsvcs-001 are running at more than 0.01 CPU (10 milicores)?

```
pod_name:container_cpu_usage:sum{namespace="mcsvcs-user001"} > 0.01
```

```
pod_name:container_cpu_usage:sum{namespace="mcsvcs-user001",
pod_name="mongodb-1-xgbqr"}                                    0.018

pod_name:container_cpu_usage:sum{namespace="mcsvcs-user001",
pod_name="trade-history-1-f9h5w"}                              0.020
```

Prometheus Alerts

Generate an alert when a node is project to run out of storage

```
alert: NodeDiskRunningFull
expr: '(node:node_filesystem_usage:
  > 0.85) and (predict_linear(node:node_filesystem_avail:[30m], 3600 * 2) <
  0)'
for: 10m
labels:
  severity: critical
annotations:
  message: Device {{ $labels.device }} of node-exporter {{ $labels.namespace }}/{{
    $labels.pod }} is running full within the next 2 hours.
```

Application monitoring

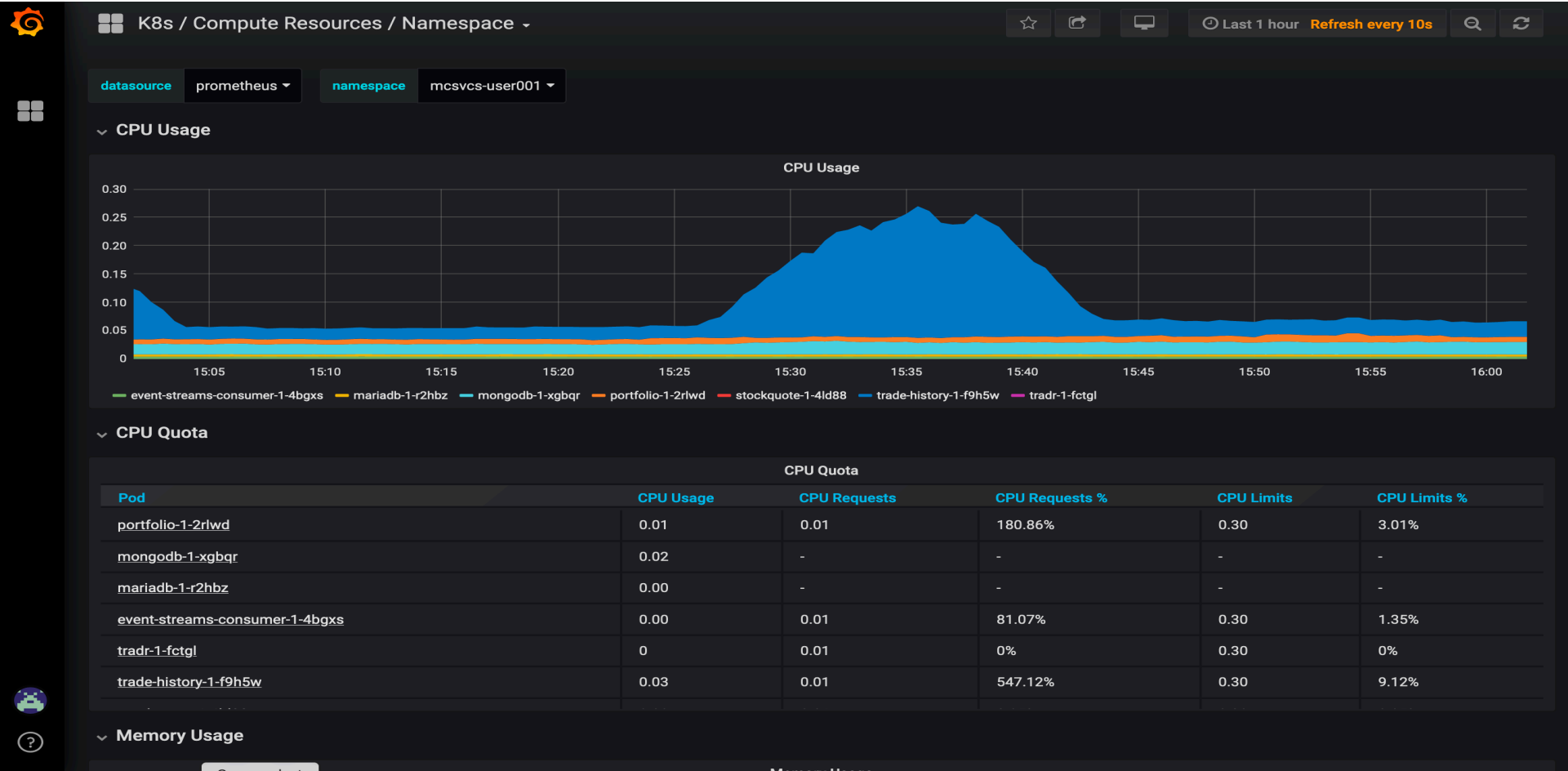
Pre-configured for cluster monitoring, but Prometheus is commonly used for applications too

```
sh-4.2$ curl http://portfolio:9080/metrics
# TYPE base:classloader_total_loaded_class_count counter
# HELP base:classloader_total_loaded_class_count Displays the total number of classes that have been loaded since the Java virtual machine has starte
d execution.
base:classloader_total_loaded_class_count 13745
# TYPE base:gc_global_count counter
# HELP base:gc_global_count Displays the total number of collections that have occurred. This attribute lists -1 if the collection count is undefined
 for this collector.
base:gc_global_count 45
# TYPE base:cpu_system_load_average gauge
# HELP base:cpu_system_load_average Displays the system load average for the last minute. The system load average is the sum of the number of runnabl
e entities queued to the available processors and the number of runnable entities running on the available processors averaged over a period of time.
 The way in which the load average is calculated is operating system specific but is typically a damped time-dependent average. If the load average i
s not available, a negative value is displayed. This attribute is designed to provide a hint about the system load and may be queried frequently. The
 load average may be unavailable on some platform where it is expensive to implement this method.
base:cpu_system_load_average 0.42
# TYPE base:thread_count counter
# HELP base:thread_count Displays the current number of live threads including both daemon and non-daemon threads.
base:thread_count 72
# TYPE base:classloader_current_loaded_class_count counter
# HELP base:classloader_current_loaded_class_count Displays the number of classes that are currently loaded in the Java virtual machine.
base:classloader_current_loaded_class_count 13706
# TYPE base:gc_scavenge_time_seconds gauge
# HELP base:gc_scavenge_time_seconds Displays the approximate accumulated collection elapsed time in milliseconds. This attribute displays -1 if the
collection elapsed time is undefined for this collector. The Java virtual machine implementation may use a high resolution timer to measure the elaps
ed time. This attribute may display the same value even if the collection count has been incremented if the collection elapsed time is very short.
base:gc_scavenge_time_seconds 7.843
# TYPE base:jvm_uptime_seconds gauge
# HELP base:jvm_uptime_seconds Displays the start time of the Java virtual machine in milliseconds. This attribute displays the approximate time when
 the Java virtual machine started.
base:jvm_uptime_seconds 19460.342
```

Grafana


K8s / USE Method / Cluster
Last 1 hour
Refresh every 10s
datasource
prometheus
CPU
CPU Utilisation
100.00%
75.00%
50.00%
25.00%
0%
15:00
15:10
15:20
15:30
15:40
15:50
10.191.61.67
10.191.61.69
CPU Saturation (Load1)
100.00%
75.00%
50.00%
25.00%
0%
15:00
15:10
15:20
15:30
15:40
15:50
10.191.61.67
10.191.61.69
Memory
Memory Utilisation
100.00%
75.00%
50.00%
25.00%
0%
15:00
15:10
15:20
15:30
15:40
15:50
10.191.61.67
10.191.61.69
Memory Saturation (Swap I/O)
13 MB/s
10 MB/s
8 MB/s
5 MB/s
3 MB/s
0 B/s
15:00
15:10
15:20
15:30
15:40
15:50
10.191.61.67
10.191.61.69
Disk

Grafana


K8s / Compute Resources / Namespace
Last 1 hour
Refresh every 10s
datasource
prometheus
namespace
mcsvcs-user001
CPU Usage
CPU Usage
0.30
0.25
0.20
0.15
0.10
0.05
0
15:05
15:10
15:15
15:20
15:25
15:30
15:35
15:40
15:45
15:50
15:55
16:00
event-streams-consumer-1-4bgxs
mariadb-1-r2hbz
mongodb-1-xgbqr
portfolio-1-2rlwd
stockquote-1-4ld88
trade-history-1-f9h5w
tradr-1-fctgl
CPU Quota
CPU Quota
Pod
CPU Usage
CPU Requests
CPU Requests %
CPU Limits
CPU Limits %
portfolio-1-2rlwd 0.01 0.01 180.86% 0.30 3.01%
mongodb-1-xgbqr 0.02 - - - -
mariadb-1-r2hbz 0.00 - - - -
event-streams-consumer-1-4bgxs 0.00 0.01 81.07% 0.30 1.35%
tradr-1-fctgl 0 0.01 0% 0.30 0%
trade-history-1-f9h5w 0.03 0.01 547.12% 0.30 9.12%
Memory Usage

When things go wrong #1

Kubernetes **Logs**

- Problems ***in*** the container

```
$ oc create deployment redis --image=redis:3.2.9
deployment.apps/redis created
$ oc exec –it redis-67455bbc75-klgrg redis-cli
127.0.0.1:6379> set foo bar
OK
127.0.0.1:6379> save
ERR

$ oc logs redis-67455bbc75-klgrg
1:C 11 Nov 00:16:14.784 # Warning: no config file specified,…
…
1:M 11 Nov 00:25:32.130 # Failed opening the RDB file dump.rdb (in server root dir /data) for
saving: Permission denied
```

When things go wrong #2

Kubernetes **Events**

- Problems ***with*** the container

```
$ oc create deployment foo --image=busyfoo:1.9
deployment.apps/foo created
$ oc get pods
NAME                   READY     STATUS         RESTARTS   AGE
foo-dcd7cd446-k85kh    0/1       ErrImagePull   0          9s
$ oc describe pod foo-dcd7cd446-k85kh
Name:               foo-dcd7cd446-k85kh
Namespace:          debugging
…
Events:
  Type     Reason     Age                From                    Message
  ----     ------     ----               ----                    -------
  Normal   Scheduled  36s                default-scheduler       Successfully assigned debugging/foo-dcd7cd446-k85kh to
10.191.61.69
  Normal   Pulling    21s (x2 over 35s)  kubelet, 10.191.61.69   pulling image "busyfoo:1.9"
  Warning  Failed     21s (x2 over 35s)  kubelet, 10.191.61.69   Failed to pull image "busyfoo:1.9": rpc error: code =
Unknown desc = Error reading manifest 1.9 in docker.io/library/busyfoo: errors:
denied: requested access to the resource is denied
unauthorized: authentication required
  Warning  Failed   21s (x2 over 35s)  kubelet, 10.191.61.69   Error: ErrImagePull
  Normal   BackOff  8s (x2 over 34s)   kubelet, 10.191.61.69   Back-off pulling image "busyfoo:1.9"
  Warning  Failed   8s (x2 over 34s)   kubelet, 10.191.61.69   Error: ImagePullBackOff
```

Summary

Effectively managing applications in a cluster requires a “big picture” approach

- Multiple application components
- Cluster health / monitoring
- Access control
- Retention policy

Useful OpenSource tools for logging and monitoring use cases