elastic Elastic Cloud on kubernetes

Kubernetes logs and metrics with ECK

Eduardo González de la Herrán, Sr. Support Engineer

April 14, 2021

Kubernetes logs and metrics with ECK

Kubernetes observability with Elastic Stack and ECK

Kubernetes monitoring and logging solution with Elasticsearch, Kibana, Metricbeat, Filebeat. All orchestrated by ECK.

Elastic Stack monitoring on Kubernetes

Monitoring methods, different options, best practices and practical examples.

elastic

Introduction

Introduction

Main topics of the session

- Kubernetes Observability with Elastic Stack & ECK
- Elastic Stack Monitoring
- All Resources available at:

https://github.com/eedugon/eck-logs-and-metrics

- Not covered
  - Elasticsearch features, architecture layouts

elastic

Introduction

Software and versions used in the demo

- Kubernetes GKE v1.19.8
- ECK 1.4.0
- Elasticsearch, Kibana, Filebeat and Metricbeat 7.11.2
- Kube-state-metrics v2.0.0-rc0

- Local tools
  - Kubectx and kubens plugins
  - ‘k’, ‘kctx’ and ‘kns’ aliases :)

elastic

Introduction

Concepts and background:

- ECK
- Beats
- Kubernetes
  - DaemonSets / Deployments / StatefulSets
  - Labels and Annotations
- Elasticsearch & Kibana
  - ILM, Data streams

elastic

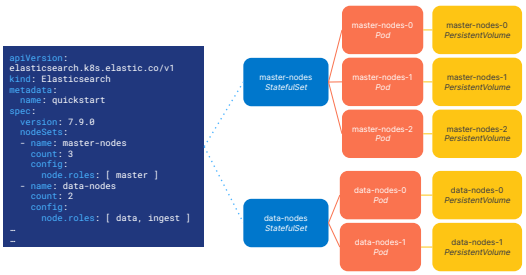
Introduction

ECK

- Orchestrator of Elastic Stack components
  - It started with Elasticsearch and Kibana and now supports also APM, Beats, Enterprise Search, etc.
  - It's a Kubernetes Operator (defines CRDs + runs a controller)

- Use case example: You create an "Elasticsearch" resource, and internally the operator will create and maintain a lot of different resources (StatefulSets, ConfigMaps, Secrets, ...)

```
apiVersion:
elasticsearch.k8s.elastic.co/v1
kind: Elasticsearch
metadata:
  name: quickstart
spec:
  version: 7.9.0
  nodeSets:
  - name: master-nodes
    count: 3
    config:
      node.roles: [ master ]
  - name: data-nodes
    count: 2
    config:
      node.roles: [ data, ingest ]
-
-
```



elastic

Introduction

ECK

- Challenges when running things on Kubernetes
  - Resource management
    - StatefulSets, Pods, Secrets, Services, PersistentVolumes
  - Day-2 operations
    - Configuration changes, version upgrades, scale up/down
  - Stateful workloads
    - Availability, consistency, volume management

elastic

Introduction

Beats

- Lightweight agents used to ship data to Elasticsearch (or another output)
  - Filebeat: Files / text content (logs)
    - Raw inputs or modules
  - Metricbeat: Metrics
    - Always configured via modules
  - Others: Auditbeat, Winlogbeat, Heartbeat, Packetbeat
- 1 module →N filesets | metricsets (check documents)

elastic

Introduction

Beats

- Beats Autodiscover
  - Allows dynamic configuration of inputs & modules
  - Hints based → input config received via pod annotations
  - Conditional templates → input config provided statically to the beat.
  - Autodiscover access Kubernetes API to get the list of running pods and metadata and keeps that in sync.
  - All events are enriched with Kubernetes metadata (`kubernetes.pod.name`, `kubernetes.namespace.name`, ...).

elastic

Introduction

Running Beats on Kubernetes

- Directly using the proposed manifests
  - Check "Running (File|Metric|Audit|etc)beat on Kubernetes" official docs for references and **updates**.
- Via helm charts
- Orchestrated by ECK:
  - ECK helps in multiple ways:
    - Mount data volume as a hostPath (requires elevated permissions)
    - Takes care of Elasticsearch output and authentication.
- As sidecar containers within a pod

Introduction

Running Elasticsearch & Kibana on Kubernetes

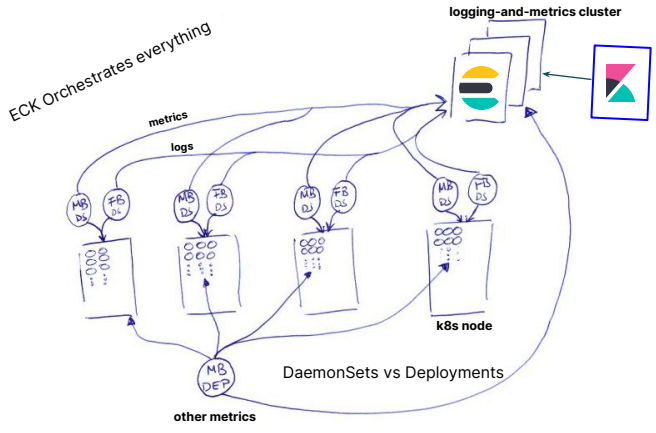
- Cooking / designing your own manifests
- Via helm charts
- Orchestrated by ECK (by far the best way).

Kubernetes Observability with the Elastic Stack

Kubernetes Observability

Overview

Architecture



elastic

Kubernetes Observability

Overview:

- Data Storage / Search Engine: Elasticsearch
- UI: Kibana
- Metrics:
  - Metricbeat DaemonSet
    - System and Kubernetes Modules
  - Metrics from users workloads (nginx, elasticsearch, ...)
- Logs:
  - Filebeat DaemonSet with autodiscover
  - Custom logs processing, modules, ...

elastic

Demo Time!

Infrastructure

Deploy ECK, namespaces and global roles, Elasticsearch and Kibana

Metrics

Deploy Metricbeat(s) to retrieve metrics from different components

Logs

Deploy Filebeat to ship all pods' logs to Elasticsearch

Stack Monitoring

Deploy different Stack Monitoring options

elastic

First thing's first: Deploy ECK, kube-state-metrics and RBAC

```
# If GKE cluster follow this
kubectl apply -f resources/01_infra
kubectl apply -f
resources/01_infra/external/kube-state-metrics-v2.0.0-rc.0/standard
```

- Raw manifests and updated instructions

Now let's deploy our logging-and-metrics Elasticsearch / Kibana cluster

Logging-and-metrics Cluster Manifest

```
kubectl apply -f
resources/02_k8s_monitoring/01_monitoring_logging-and-metrics_cluster.yaml
```

What do we have now?

Elasticsearch, Kibana, a lot of internal resources like Secrets, Services, etc

Let's access Kibana and take a look at the UI! All should be empty...

- Fetch elastic password, prepare local hostname resolution (optional)

We need logs and metrics!!!

Kubernetes Observability

Metrics

- Run Metricbeat in each Kubernetes node to fetch metrics via:
  - System module (OS metrics)
  - Kubernetes module
    - Retrieves metrics from different components (kubelet, kube-proxy, apiserver, kube-state-metrics, etc)
    - More than 20 metricsets.
    - Some are "local" (per node), others are unique (cluster level) → DaemonSet vs Depoyment (*)
    - HostNetwork = true & runs as root (for system metrics)
- Follow "Run Metricbeat on Kubernetes" for updates on default proposal and RBAC requirements.

Kubernetes Observability

Metrics

- Extras
  - Use Metricbeat Autodiscover for extra modules setup in new Deployments or DaemonSets. Do not use hostNetwork if not strictly needed.
- Advices:
  - Work on your own dasbhoards
    - Default dashboards are not updated very often (better to rely on Metrics UI or custom dashboards)

elastic

Demo Time!

Infrastructure

Deploy ECK, namespaces and global roles, Elasticsearch and Kibana

Metrics

Deploy Metricbeat(s) to retrieve metrics from different components

Logs

Deploy Filebeat to ship all pods' logs to Elasticsearch

Stack Monitoring

Deploy different Stack Monitoring options

elastic

Grabbing some metrics...

Kubernetes metrics

```
kubectl apply -f resources/02_k8s_monitoring/
```

Other metrics (elasticsearch and kibana example)

```
kubectl apply -f resources/02_k8s_monitoring/stack_monitoring
```

Explore them in Kibana!

- Observability UI
- Metrics UI
- System Overview (legacy dashboard)
- Kubernetes Overview Dashboard

+ Use discovery to explore data and Import the proposed dashboard

Kubernetes Observability

Logs

- Objective: Fetching the logs from (all / some) pods via Filebeat
  - Running Filebeat in each Kubernetes host with access to containers' log files (DaemonSet)
- Filebeat Autodiscover gives a lot of flexibility.
- HostNetwork = true (not strictly needed) & run as root needed.
- Follow "Run Filebeat on Kubernetes" for updates in the default manifest. This manifest usually offers these options:
  - Single input all pods log files
  - →**Hints based autodiscover fetching all logs by default**←

elastic

Kubernetes Observability

Logs

- Advanced use cases:
  - Using modules together with Autodiscover (check Elasticsearch and Kibana pods annotations!!!)
  - Processing logs via custom pipelines in Elasticsearch
  - Configure json processing
  - Ship logs to different indices based on namespace (watch out for total amount of indices and shards).
    - With ILM → bootstrap of rollover groups need to be done in advance
    - With Data Streams → bootstrap is done automagically.
  - What if an application is not logging to stdout/stderr?
    - Run filebeat as a sidecar container

elastic

Demo Time!

Infrastructure

Deploy ECK, namespaces and global roles, Elasticsearch and Kibana

Metrics

Deploy Metricbeat(s) to retrieve metrics from different components

Logs

Deploy Filebeat to ship all pods' logs to Elasticsearch

Stack Monitoring

Deploy different Stack Monitoring options

elastic

Let's log now!

Kubernetes logs

```
kubectl apply -f
resources/02_k8s_monitoring/11_logs_k8s_all_autodiscover.yaml
```

Explore the logs with Kibana Discovery and the provided dashboard

What do you see?

Check Kibana and Elasticsearch pods annotations

What if I want to parse my application logs into different fields?

You can follow this example to process your logs

```
kubectl apply -f resources/02_k8s_monitoring/extras/custom-formats
```

Visualize the results here

What if I want to split my data into multiple indices?

Sometimes that's not the best decision, but here you have a complete example explained!

```
kubectl apply -f resources/02_k8s_monitoring/extras/ns_data_streams
```

Visualize the results with:

- Discovery and `logs-fb-*` index pattern
- Checking created indices with DevTools (GET _cat/indices?v&s=index)

Kubernetes Observability Recap

Use Cases covered

Kubernetes Observability

Use Cases Covered

- Filebeats with and without hostNetwork: true + implications.
- Filebeat examples with hints based and conditional templates autodiscover (be careful with double processing same logs).
- Logging into namespace based indices and Elasticsearch Data Streams
- Custom logs (json and structured text)
- Using modules (ingress controller / elasticsearch / kibana)
- Metricbeat deployments with autodiscover to fetch specific metrics (Elasticsearch, Kibana and Beats).
- Visualize the overall logs and metrics sent to Elasticsearch (custom dashboard)

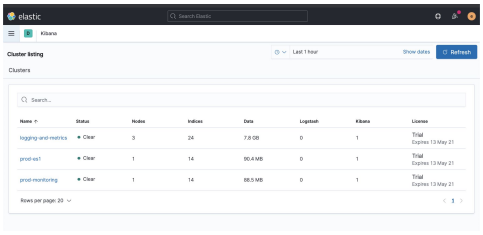
elastic

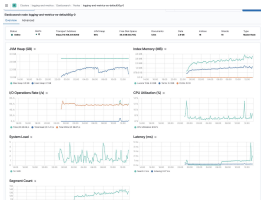
Elastic Stack Monitoring on Kubernetes

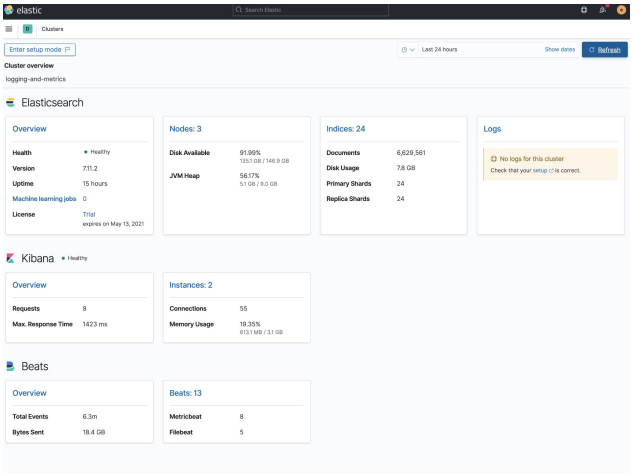
Elastic Logs and Metrics

Elastic Stack Monitoring on Kubernetes

What do we mean with that?







elastic

Elastic Stack Monitoring on Kubernetes

Elasticsearch & Kibana monitoring

- Metrics:
  - Collecting methods:
    - Metricbeat with modules (recommended)
    - Internal collectors
- Logs:
  - With filebeat modules (elasticsearch / kibana)

elastic

Elastic Stack Monitoring on Kubernetes

Monitoring methods

- Self-Monitoring
  - Not recommended for production
    - Specially logs (not recommended)
    - Same for filebeat logs
- Dedicated monitoring cluster (1-1)
- Centralized monitoring cluster (1-N)
  - Requires Gold license

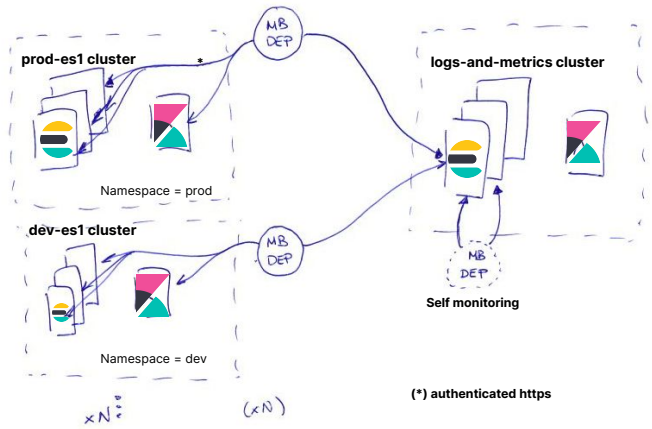
Official reference (baremetal / VMs)

elastic

Elastic Stack Monitoring on Kubernetes

Elasticsearch & Kibana Monitoring (metrics)

Centralized Monitoring (dedicated cluster) - metrics



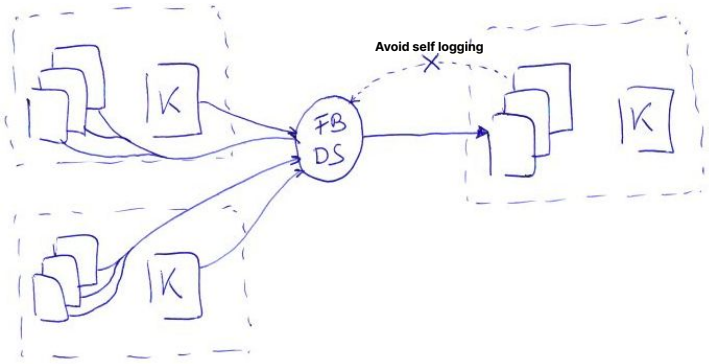
(*) authenticated https

elastic

Elastic Stack Monitoring on Kubernetes

Elasticsearch & Kibana Monitoring (logs)

Centralized Monitoring (dedicated cluster) - logs



elastic

Elastic Stack Monitoring on Kubernetes

Beats monitoring

- Beats metrics: collecting methods
  - Metricbeat with Beats module

```
http.enabled: true
http.port: 5067
http.host: 0.0.0.0
monitoring.enabled: false
```

→ hostNetwork implications!

    - Publish the port at pod level
    - Configure a Metricbeat instance with beat module to access that endpoint.
  - Internal collectors

```
monitoring.enabled: true
```

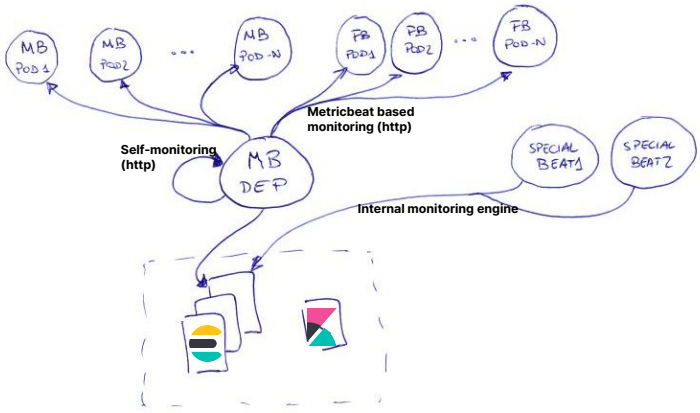
  - Legacy collector (deprecated): `xpack.monitoring.* (not used anymore)`

elastic

Elastic Stack Monitoring on Kubernetes

Beats Monitoring (metrics)

Beats Metrics



elastic

Elastic Stack Monitoring on Kubernetes

Monitoring Implementation

- Situations to avoid:
  - Self indexing elasticsearch logs + configure audit logs (exponential growth)
  - Be careful with indexing filebeat logs directly to same elasticsearch output (endless loops)

elastic

Demo Time!

Infrastructure

Deploy ECK, namespaces and global roles, Elasticsearch and Kibana

Metrics

Deploy Metricbeat(s) to retrieve metrics from different components

Logs

Deploy Filebeat to ship all pods' logs to Elasticsearch

Stack Monitoring

Deploy different Stack Monitoring options

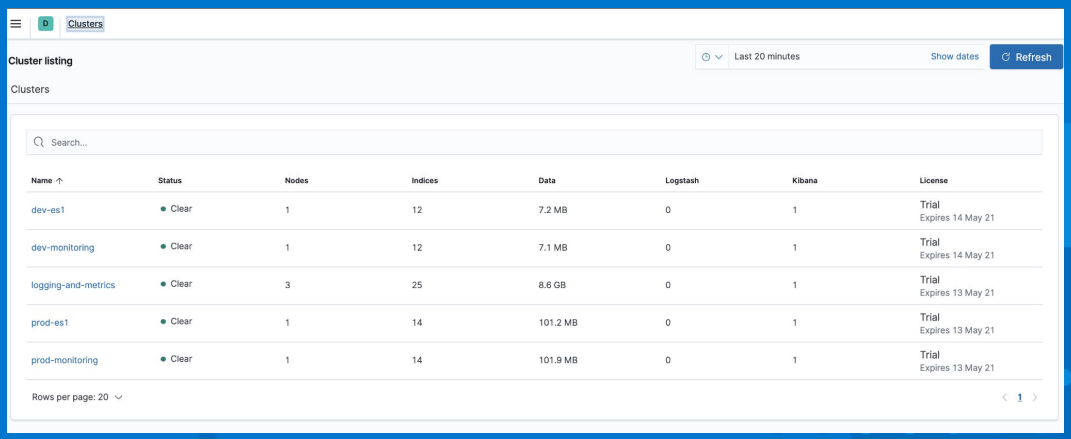
elastic

Stack Monitoring Demo (all types)

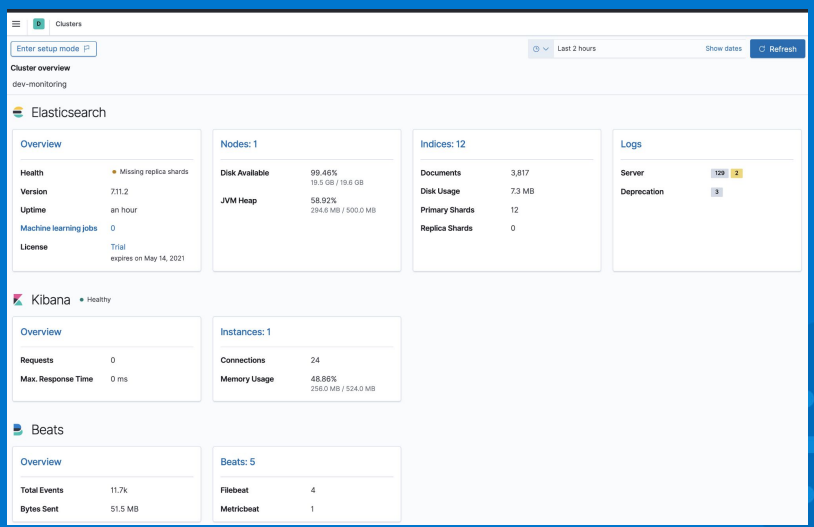
```
kubectl apply -f resources/03_prod
kubectl apply -f resources/03_prod/stack_monitoring/basic/self-monitoring
kubectl apply -f resources/03_prod/stack_monitoring/basic/dedicated-monitoring
kubectl apply -f resources/03_prod/stack_monitoring/enterprise/central-monitoring

And same for dev namespace!
```

Stack Monitoring Demo (all types)



Stack Monitoring Demo (all types)



Thanks!!!

What's next?

- Auditbeat, Heartbeat, ...
- Building your own visualizations
- Alerting
- Machine Learning and other ways to explore the data
- And much more!