

# SCALE

# YOUR

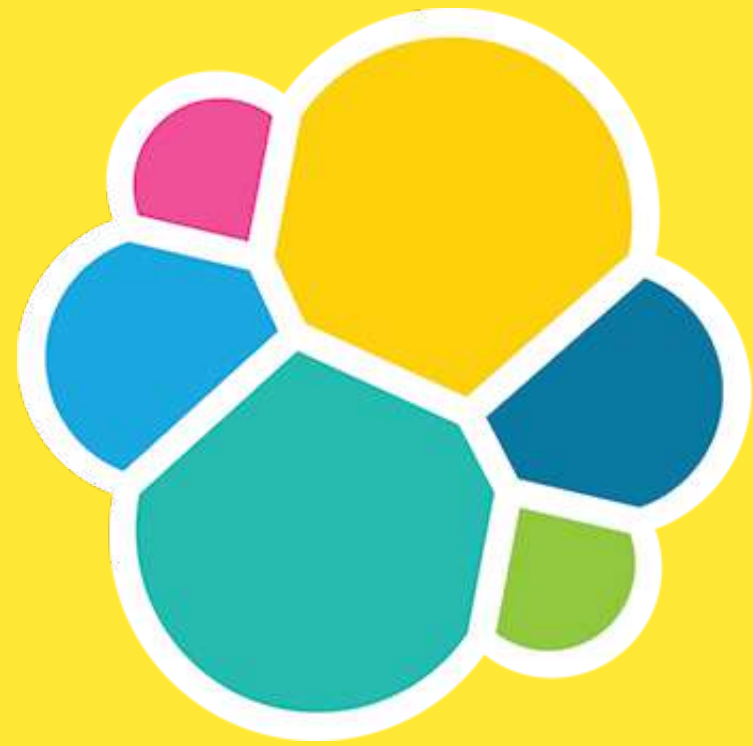


elasticsearch

# CLUSTER

# PHILIPP KRENN

# @XERAA



elastic

**DEVELOPER**



# **AGENDA**

## **LIFECYCLE MANAGEMENT**

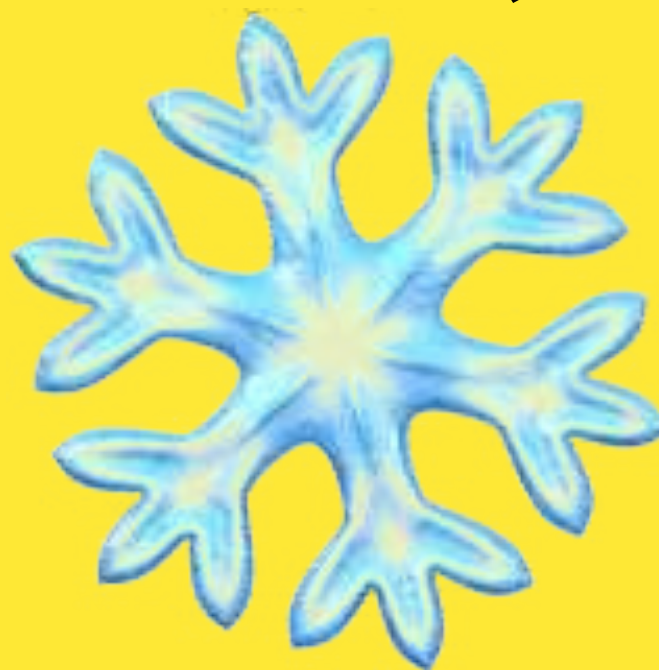
### **COLD STORAGE**

### **ROLLUPS**

*Start the Demo Setup*

# LIFECYCLE MANAGEMENT

# INDEX LIFECYCLE MANAGEMENT (ILM)



# FEATURES & ORDER

<https://github.com/elastic/elasticsearch/blob/7.9/x-pack/plugin/core/src/main/java/org/elasticsearch/xpack/core/ilm/TimeseriesLifecycleType.java>

```
static final List<String> ORDERED_VALID_HOT_ACTIONS = Arrays.asList(
    SetPriorityAction.NAME, UnfollowAction.NAME, RolloverAction.NAME,
    ForceMergeAction.NAME);
static final List<String> ORDERED_VALID_WARM_ACTIONS = Arrays.asList(
    SetPriorityAction.NAME, UnfollowAction.NAME, ReadOnlyAction.NAME,
    AllocateAction.NAME, ShrinkAction.NAME, ForceMergeAction.NAME);
static final List<String> ORDERED_VALID_COLD_ACTIONS = Arrays.asList(
    SetPriorityAction.NAME, UnfollowAction.NAME, AllocateAction.NAME,
    FreezeAction.NAME, SearchableSnapshotAction.NAME);
static final List<String> ORDERED_VALID_DELETE_ACTIONS = Arrays.asList(
    WaitForSnapshotAction.NAME, DeleteAction.NAME);
```

# TIMING

**MIN\_AGE: INDEX AGE OR ROLLOVER DATE**

**PREVIOUS PHASE MUST HAVE FINISHED**



# WHAT ABOUT ELASTIC CURATOR?

# SNAPSHOT LIFECYCLE MANAGEMENT (SLM)



# Demo

# PS: 7.0 IMPROVEMENT

INDEX.SEARCH.IDLE.AFTER: 30S

IFF DEFAULT INDEX.REFRESH\_INTERVAL

# COLD STORAGE

**RATIO HEAP : STORAGE**

**INDEX > FROZEN INDEX > CLOSED INDEX**

**FROZEN INDEX**  
**READ-ONLY**  
**NO MEMORY**

# THROTTLED THREAD POOL

1 PARALLEL SEARCH / NODE

100 IN QUEUE



# Demo

**PRE\_FILTER\_SHARD\_SIZE**

**ADDITIONAL ROUND TRIP**

**SKIP IMPOSSIBLE SHARDS**

**DATE RANGE FILTER**

# SEARCHABLE SNAPSHOTS

## SOON MORE...

# ROLLUPS



**think of the bytes**

# Demo

# CONCLUSION

# CODE

<https://github.com/xeraa/scale-elasticsearch>



**RECAP**  
**LIFECYCLE MANAGEMENT**  
**COLD STORAGE**  
**ROLLUPS**

**gcp.data.highio.1** Data Ingest Master

An I/O optimized Elasticsearch instance.

**Fault tolerance**

1 zone  2 zones  3 zones

**RAM per Node**

1 GB 2 GB 4 GB 8 GB 16 GB 32 GB 64 GB

Nodes  = 4 GB RAM per Zone

**Summary**

4 GB RAM 120 GB storage × 1 node × 2 zones = 8 GB RAM 240 GB storage

> User setting overrides

**gcp.data.highstorage.1** Data Ingest Master

A storage optimized Elasticsearch instance.

**Fault tolerance**

1 zone  2 zones  3 zones

**RAM per Node**

2 GB 4 GB 8 GB 16 GB 32 GB 64 GB

Nodes  = 4 GB RAM per Zone

**Summary**

4 GB RAM 400 GB storage × 1 node × 2 zones = 8 GB RAM 800 GB storage

> User setting overrides

Machine Learning 1 configuration

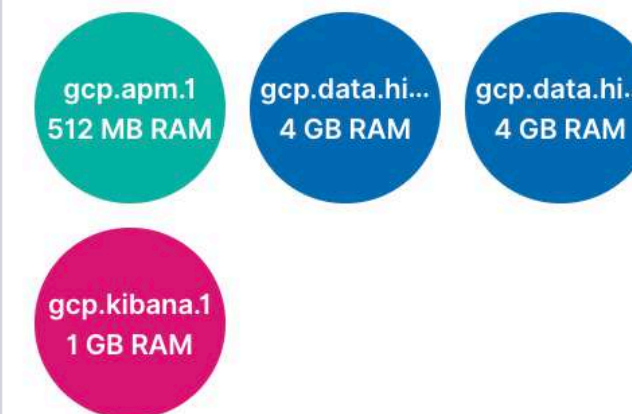
**gcp.ml.1** Machine Learning

## Summary

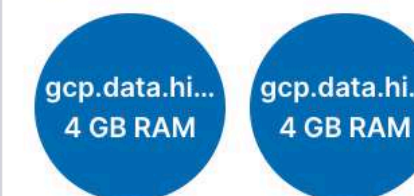
<b>Version</b>	v7.1.1
<b>ES data memory</b>	16 GB
<b>ES data storage</b>	1.02 TB
<b>Total memory</b>	17.5 GB
<b>Total storage</b>	1.02 TB
<b>Hourly rate</b>	\$0.7328
<b>Monthly rate</b>	\$534.94

## Architecture

### Zone 1



### Zone 2



- gcp.apm.1
- gcp.data.highio.1
- gcp.data.highstorage.1
- gcp.kibana.1

# Questions?

**PHILIPP KRENN**

**@XERAA**

