highly available elasticsearch

Mustafa Can Sevinc

10.01.2022

💼 My Elasticsearch Experience

- Since 6.2.4 to 7.2
- Various ES clusters like:
  - One-node clusters
  - Two-node clusters
  - Three-node clusters
  - Using Ceph RBD as massive storage in data roles
  - To sync metadata of objects in s3 buckets

📝 My Submission

I've done the first time while my assignment submission:

- Configure roles in a single nodeSet
- max_map_count using an initContainer instead of `node.store.allow_mmap: false`

Different on new elasticsearch versions:

- Optimized auto-configuring most of the settings
- The License

🎯 Features

- ☁️ Works on Kubernetes
- 🌀 High availability
- 🎭 Identical roles
- ✍️ README-driven
- 🧩 Kustomize-generated resources
- 🪄 Easily applicable

📐 Solution Architecture

- 💪 Resilience
- 🎭 Roles
- 💎 Sharding
- 💾 Storage
- 🧮 Memory & JVM Size
- 🥽 Virtual Memory
- 🛠️ Applying Custom Configuration
- 📈 Benchmark

📐 Solution Architecture

💪 Resilience

- Resilient if:
  - green,
  - at least two data nodes,
  - at least one replica for each shard,
  - at least thee master nodes,
  - load balancer
- Taking regular snapshots: SLM
- Design: Identical three nodes to ensure resilience to single-failure-node

📐 Solution Architecture

🎭 Roles

- master
- data
- ingest
- ml

📐 Solution Architecture

💎 Sharding

Aim for:

- Shards between 10GB and 50GB.
- Max 20 shards per GB of heap memory

Avoid:

- Unnecessary mapped fields by using explicit mapping

📐 Solution Architecture

💾 Storage

- Network-attached PersistentVolumes
- Local PersistentVolumes

🧮 Memory & JVM Size

Xms and Xmx should be

- Same with each other
- Set to no more than 50% of the total available RAM
- Less than 26GB

📐 Solution Architecture

🥽 Virtual Memory

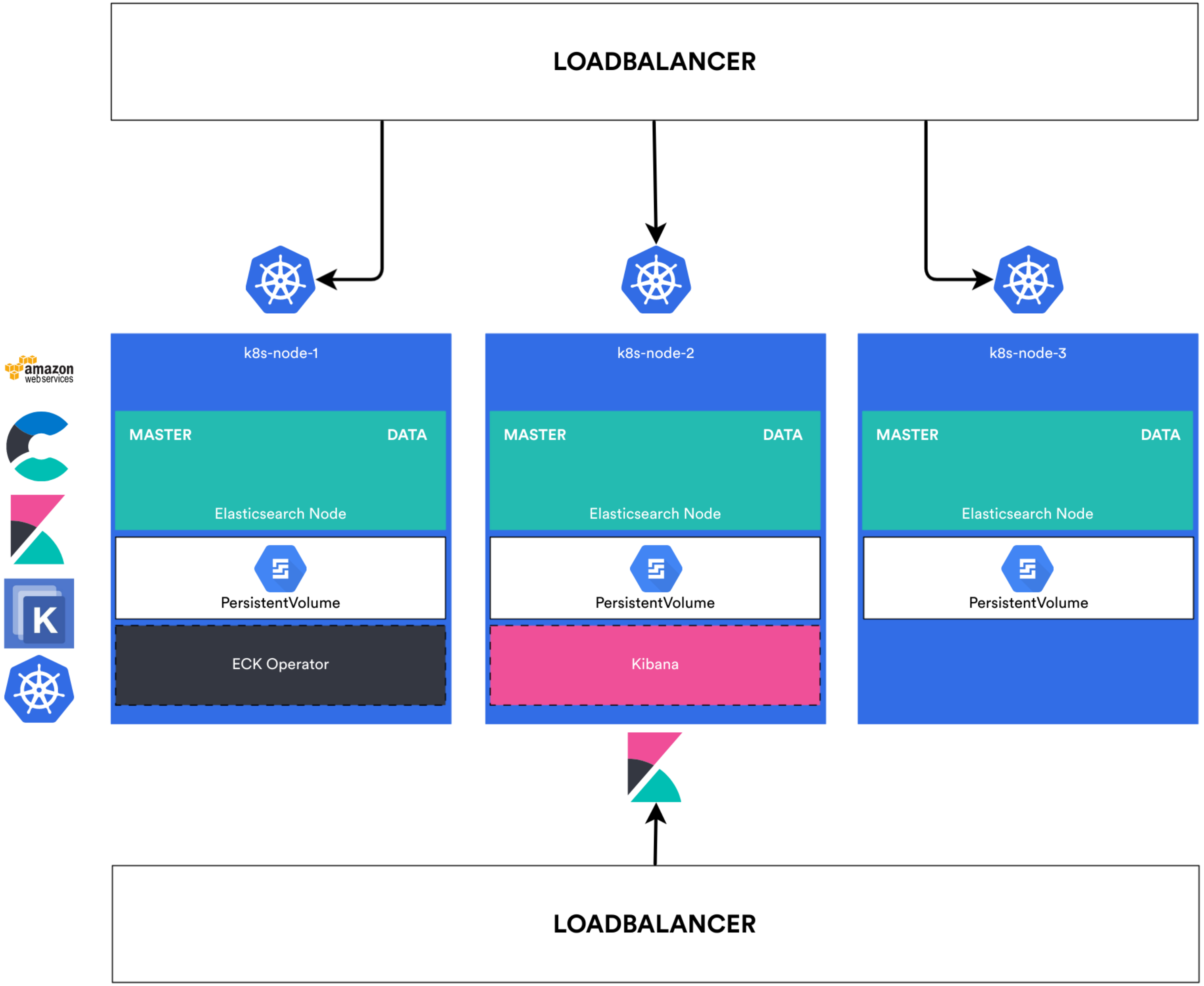
- Elasticsearch uses memory mapping.
- `vm.max_map_count` should be set to `262144`

🛠️ Applying Custom Configuration

- Create a custom image
- Use init containers

📈 Benchmark

- Rally can be used to size the cluster correctly



The Diagram of The Solution

💡 The Solution - Configuration

- ECK with **vanilla manifest files**
- Elasticsearch and Kibana with **kustomize**-generated file
- Configured using **initContainers**
- LoadBalancer
- Dynamic mapping option
- AWSElasticBlockStore
- Master & Data roles
- SLM Policy

💡 The Solution - Defaults

- Total shards per node
- JVM Heap Size Settings
- Update strategy
- PodDisruptionBudget configuration
- Node scheduling
- Readiness probe configuration
- PreStop hook configuration
- Security context configuration

🚀 Deployment

1. Install ECK Custom Resources
2. Install ECK Operator
3. Monitor the operator logs
4. Generate elasticsearch & kibana resources
5. Deploy elasticsearch & kibana
6. Verify everything is ready-to-use

Thanks for your time