Open APIs for Open Minds

Scalable deployments of FIWARE

Stefan Wiedemann, Technical Lead & Architect
FIWARE Foundation

FIWARE

Topics

- Understanding of a scalable deployment
- Infrastructure
- Use-case Analysis
- Scaling the Orion-LD context broker
- Tooling and testing

Scalable deployments

Wikipedia:

“Scalability is the property of a system to handle a growing amount of work by adding resources to the system.”

Gartner:

“Scalability is the measure of a system’s ability to increase or decrease in performance and cost in response to changes in application and system processing demands.”

FIWARE

Scalable deployments

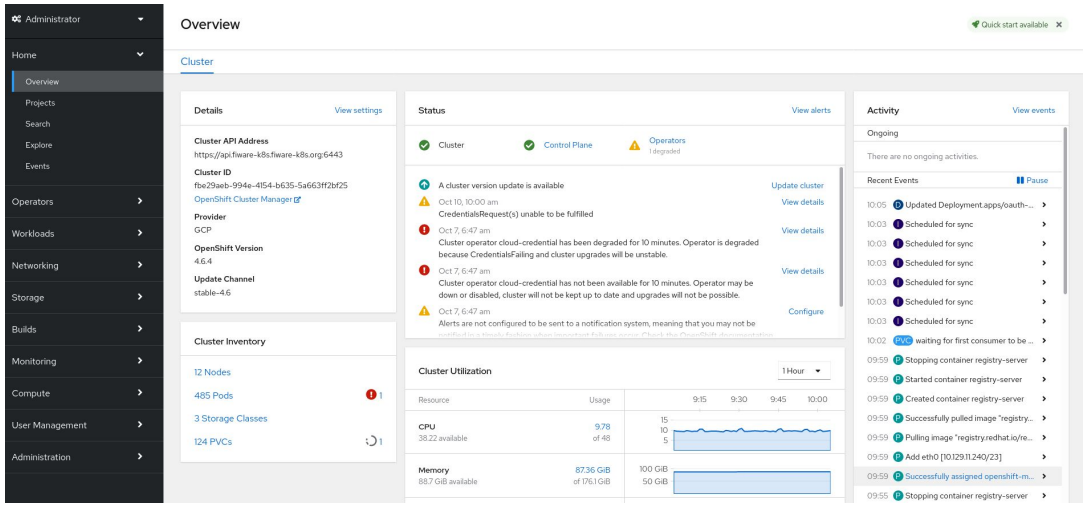
- constant trade-off between costs and performance
- no pure focus on growth
- continuously monitor and reconsider all decisions

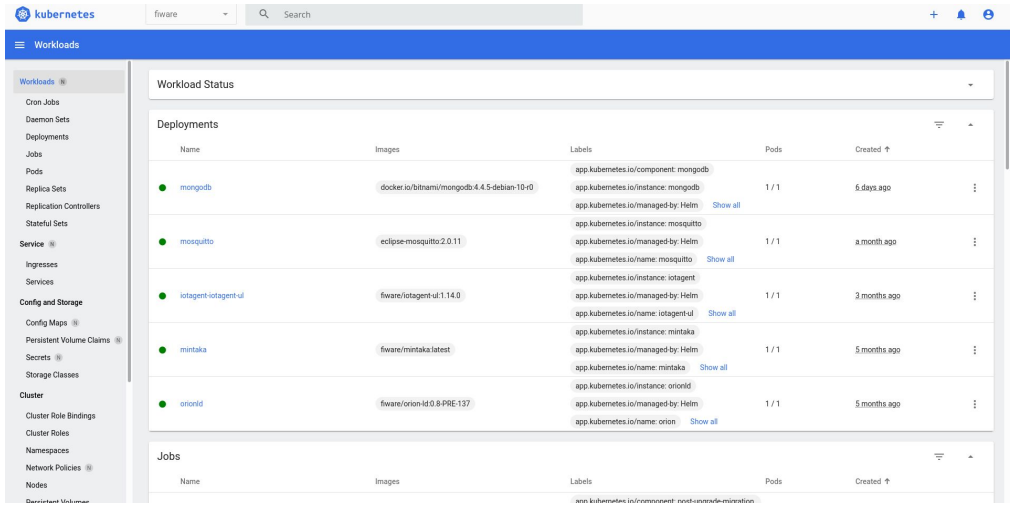
⟶Scalability needs flexibility

FIWARE

Infrastructure

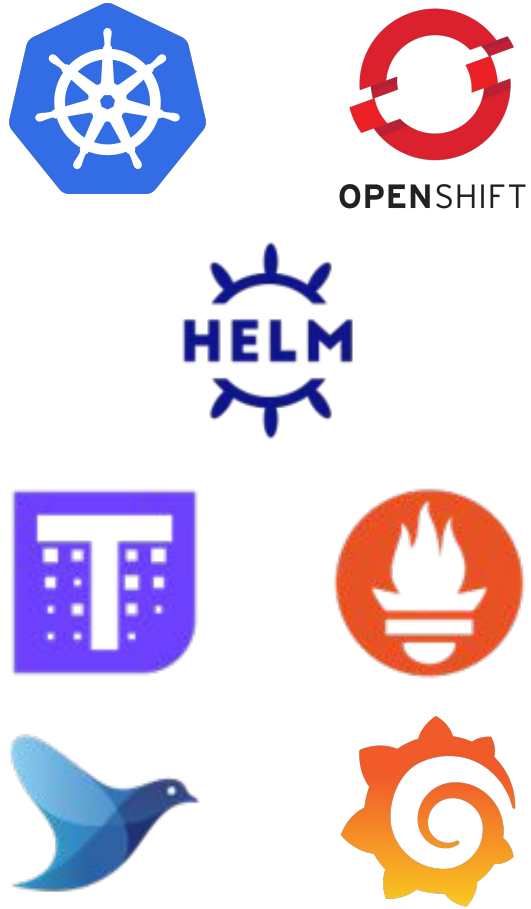
- Infrastructure is more than only computational resources
- Tooling needs to reflect the “flexibility”-goal and the ability to scale
- Should support workflows to quickly react on changing requirements





Infrastructure

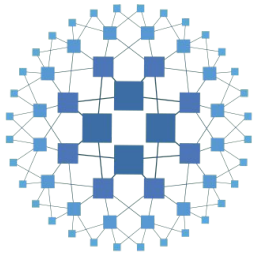
- Kubernetes as standardized abstraction layer and orchestration-tool
- Automate configurations and deployment
  - See https://github.com/FIWARE/helm-charts
- Operational tooling for
  - Logging
  - Monitoring
  - Alerting
- Know the capabilities of your infrastructure provider



FIWARE

Infrastructure

- Ingress-Controllers need the ability to scale
- SSDs will increase performance, but also the cost
- Managed services should be evaluated



FIWARE

Use-case analysis

- decide on the core metrics to optimize for

Read vs. Write

- Focus on high frequency updates
- Many parallel (retrieval) requests
- Mixture of both

Update size and parallelity

- Single entity/attribute updates
- Batch updates (of varying sizes)
- Few high-frequency clients
- Many parallel client

Use-case analysis

- decide on the features required for the use-case

Subscriptions

- Multiple subscribers
- Single/few subscribers
- Complex subscription queries
- Full data updates

Temporal representation

- Full history for every entity
- History only for a subset
- Reduced density enough?
- History retention/down sampling

Take a look at Orion Performance Tuning for detailed options.

Scaling Orion-LD

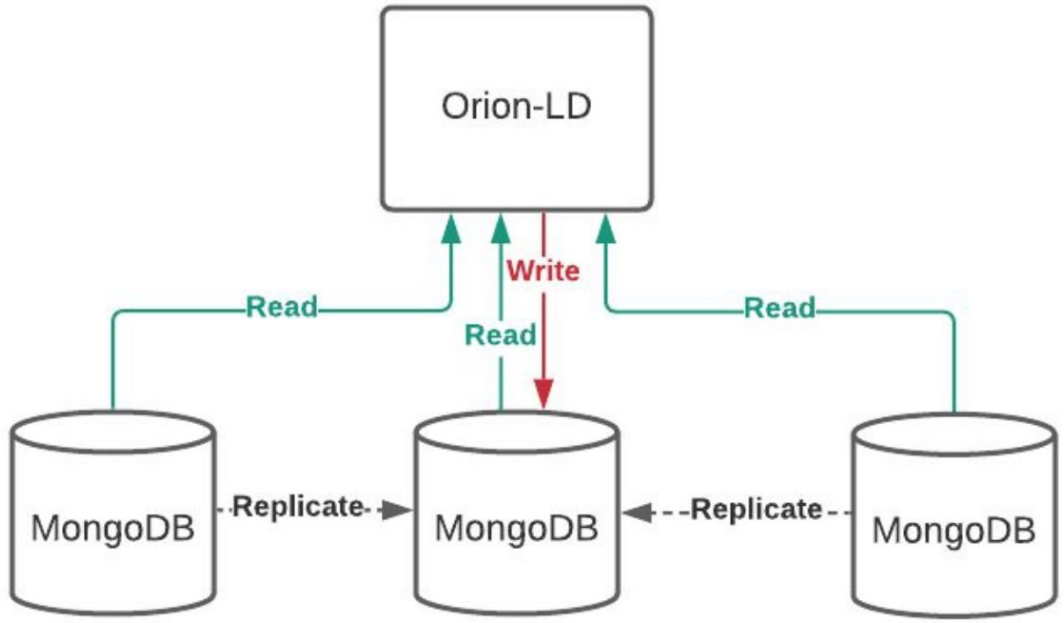
- Improve write performance

- Set “db.entities.createIndex({“_id.id”:1})” on MongoDB
- Use SSDs on MongoDB(if self-managed)
- Prefer scaling MongoDB vertical over horizontal
- Batch-Operations: Increase memory assigned to each replication of Orion

Scaling Orion-LD

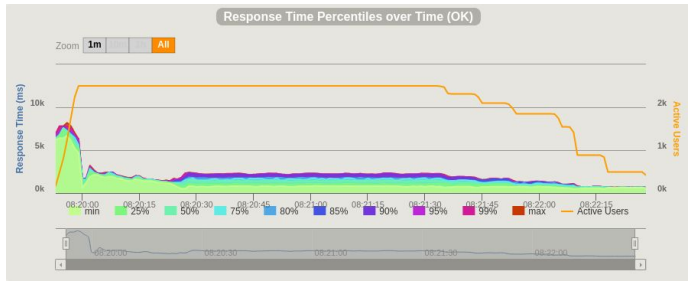
- Improve read performance

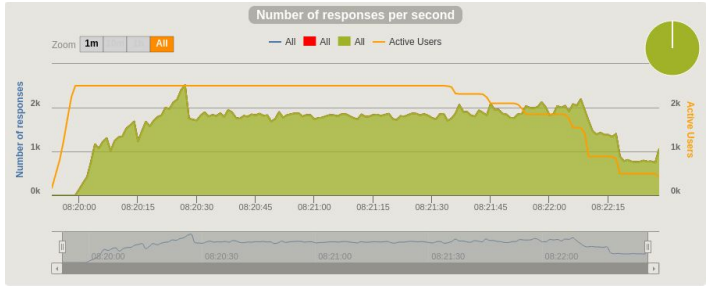
- Indexes fitting the common queries
- Use read replicas
- InMemory-engine for the read-replicas

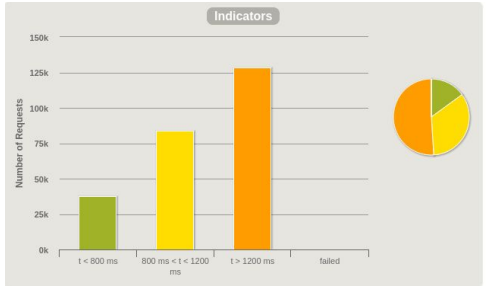


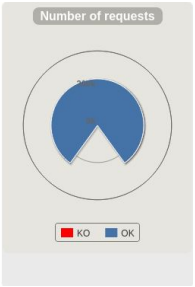
Tooling and testing

- Load-tests for the NGSI-LD/V2 context brokers
  - https://github.com/FIWARE/load-tests
  - Gatling framework, multiple scenarios implemented
  - reports and configurations for multiple sizes
- Execute tests from local(mvn, gatling reporter)
- Execute distributed in the cluster(kubernetes jobs)









Repositories

- https://github.com/FIWARE/helm-charts
- https://github.com/FIWARE/orion-loadtest

Thank you!

http://fiware.org
Follow @FIWARE on Twitter