Container Bootcamp

Microservice Operations

INNOQ

Operations

- Huge challenge
- Need to operate 50-100 Microservices
- More (virtual) system than an IT department might have

Provisioning

Provisioning

- Manual deployments too much effort
- …and too slow
- Manual deployments: hard to get right
- …and hard to reproduce

- So: Automate!

Provisioning Tools

- Your favorite package manager (apt-get, yum, HELM …)
- …but: (operating) system dependent

- Puppet , Chef, Ansible, Salt

- Your custom solution

Idempotent Provisioning

- Idea: Describe desired state of system after deployment
- Each install run gives the same result
- Machine fresh -> full install
- Machine up to date -> nothing

Idempotent Provisioning

- Problem: Complex
- No scripts, but declarations

- Problem: Might be incomplete

- Problem: Really idempotent?

Immutable Server

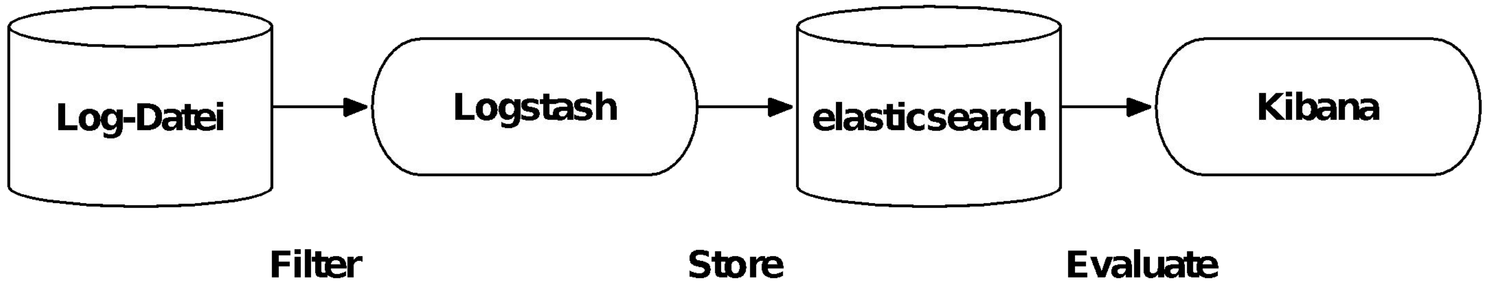
- Idea: Server cannot be changed
- Instead: Provide a complete new server for each update
- Technologies like Docker enable this

Log Analysis

Log Analysis

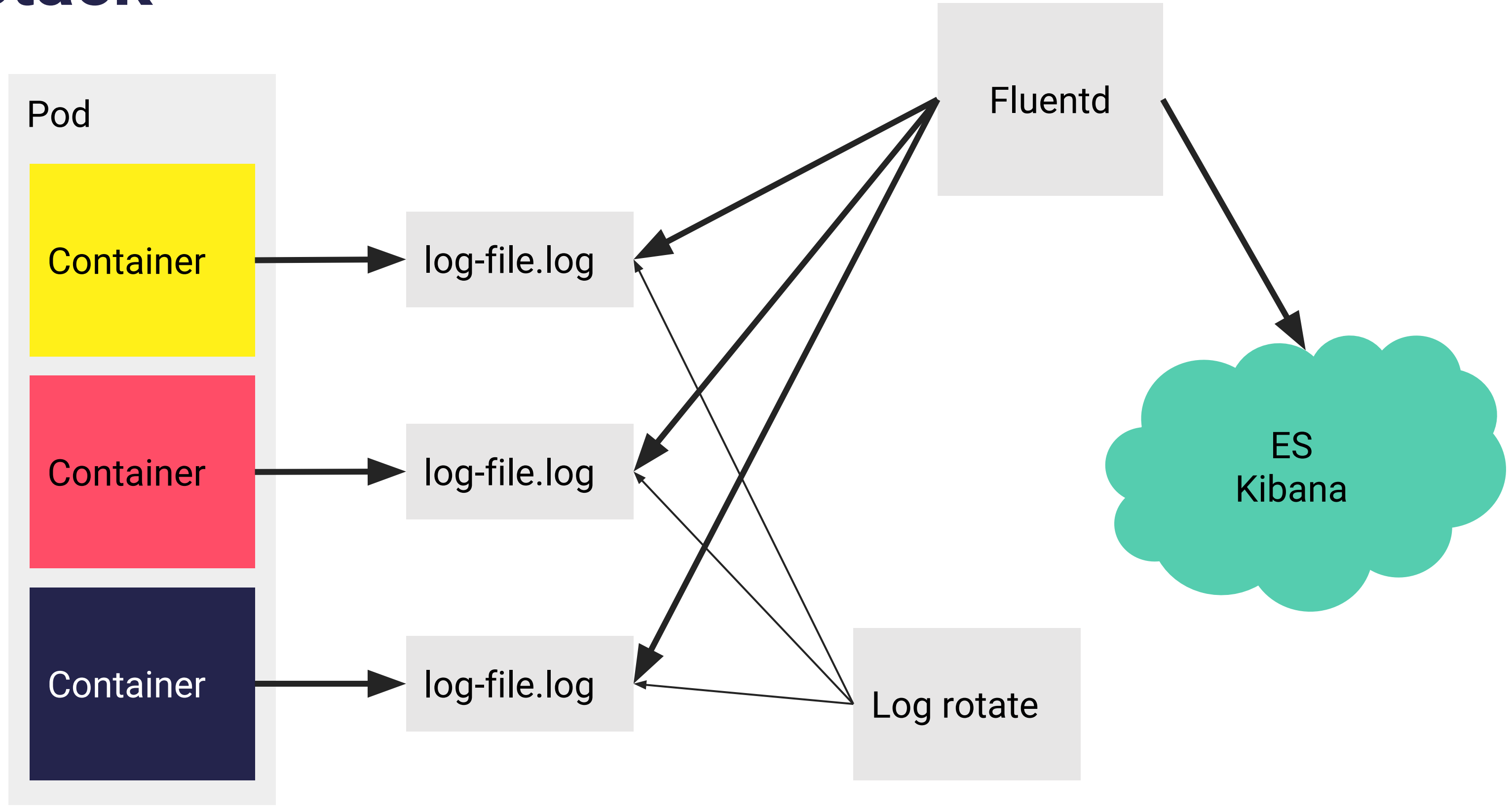
- Log Files are easy to analyze
- But: Microservices are distributed
- …so are logs
- Must analyze all logs
- Centralized log analysis

ELK Stack



- Logstash: JRuby
- Inputs, Parser, Outputs
- Elasticsearch: Java
- Kibana: JavaScript

EFK Stack


Pod
Container
Container
Container
log-file.log
log-file.log
log-file.log
Fluentd
ES
Kibana
Log rotate

Graylog

- Open Source
- Elasticsearch for storage
- MongoDB for Meta data
- GELF format for log messages

More Alternatives

- Splunk: commercial
- +Cloud

- Cloud : Loggly
- Sumo Logic
- Papertrail

Monitoring

App Metrics

- **Covers production relevant technical service metrics**
- **E.g. service quality degrading without a complete service failure**
  - Pull principle, for things like:
    - Number of HTTP 500 response codes in the last 10 minutes.
    - Average DB query duration
- **Could be used for Auto-Scaling**
- **f.e. Prometheus**
  - While queries should be written by service developer

Node Metrics

Node Metrics are still valuable, to get an insight to per node HW utilization

- Memory load
- Processes
- File descriptors
- Network throughput

Business Metrics

There might be a need for measuring business metrics.

Depending on the stakeholder this might require a different technical solution

Prometheus

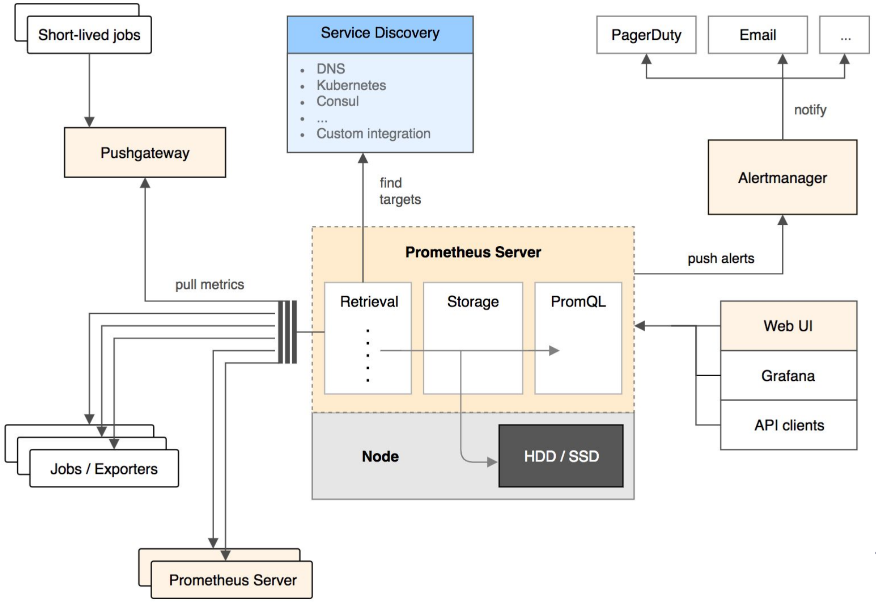
Is basically a Time Series Database

Polls data from a /metrics endpoint

Complex query language

Alerting

Visualization



Tracing

Tracing

- How is a request/workflow handled throughout the system?
- Trace call in each Microservice
- E.g. log + HTTP header

Standardize

- Consider to standardize
- Log format
- How logs are written
- How values are forwarded to monitoring
- Tracing
- Simplifies operations
- Individual system have no benefits

DevOps?

- DevOps = Development + Operations
- Close collaboration
- Ops skill become more important for Dev
- But: Do you need DevOps for Microservices?
- No – Ops define Macro-Architecture only

Conclusion

Conclusion

- Automate Provisioning
- Central log processing
- Central monitoring
- Standardize!
- DevOps beneficial but not required