Application configuration in containers

Johannes M. Scheuermann

Köln, 08.06.2016
Johannes M. Scheuermann
IT Engineering & Operations @ inovex

› Software-Defined Datacenters
› Container ecosystem
› Cloud technologies
› High Availability & Scalability
› @johscheuer

› https://github.com/johscheuer/Brownbags/tree/master/docker_app_config
What about you?
Application configuration

Configuration
  - Static
    - Image
  - Dynamic
    - Mount/Env
    - K/V store
Static configuration

• Inflexible
• Restart required
• Easy to setup
Image (on build)

- Good for default configuration
- Dev / Prod parity
- Easy distribution
- Needs a rebuilt on change
Start up time

• Mount from host
• Inject environment variable
  • Only simple key/values
  • Fixed from start time
  • echo/sed in configuration files
• Can overwrite default configuration
Dynamic configuration

• Flexible
• Application must support it
• Complexity
  • Configuration management
  • Key/Value store
Mount

• Mount config from host
• Configuration management
• Complex configuration
• Independent of Container
Key value store

- Very flexible
- Environmental variables
- Configuration files
  - Template files
- Separated setup needed
- Confd / Consul-template / envconsul
Environments

• Kubernetes
  • Configmap
  • Secrets
• Mesos / Marathon
  • Artifact Store?
• Swarm?
  • Environment variables
Distributed

• Not all solutions will work
• Mount doesn’t work
  • Config on all Nodes
• Key Value store
  • Separated setup
  • Additional complexity
Confd

- Many backends
- TOML (Tom's Obvious, Minimal Language)
- Polling/Watch/Onetime
- CMD to run after changes
Confd (high level)

Prod

Test

Host

etcd cluster

keys/prod/..
keys/test/..

prod

test
Consul template/env

- Similar to confd
- Only Consul
- HCL (HashiCorp Configuration Language)
- Supports plugins
- Rich feature set
Conclusion

• Find the right choice
  • What fits your application best?

• Combinations possible
  • See example

• No holy grail
We are hiring!

Become an inovexpert

www.inovexperts.com