Creating Service Clouds with Kubernetes, CoreOS, Containers and Co

thomas@endocode.com

ENDOCODE

CONTAINER

ENDOCODE

CONTAINER

- container
  - resources
  - access rights
  - images
- formats
  - rkt
  - aci
- registries

ENDOCODE

CONTAINER VS VIRTUALIZATION

Topic	**Container**	**Virtualisation**	
Isolation	OS Level, OS namespaces	CPU Level: Ring 0/Ring 3	
foreign CPU	no	yes, with emulation	
foreign kernels, OS	no	yes	kernel is common
emulated devices	no	yes	security
host devices	direct	virtio driver	security
CPU performance	100%	95%	
IO performance	100%	<<100%	
root isolation	yes	yes	USER directive
CPU cache attacks	easy	possible	PoC ?

KUBERNETES

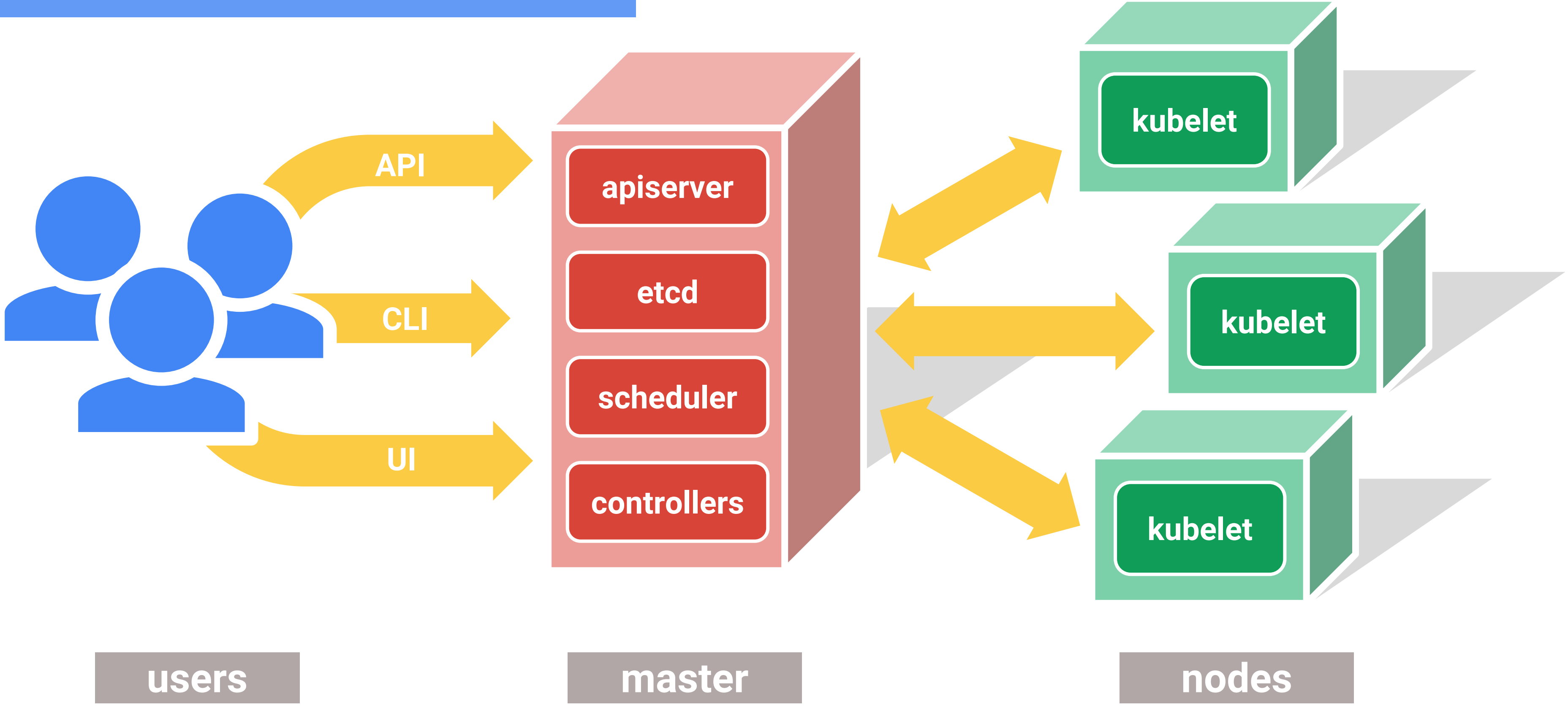
Greek for *“Helmsman”*; also the root of the words *“governor”* and *“cybernetic”*

- Runs and manages containers
- Inspired and informed by Google’s experiences and internal systems
- Supports multiple cloud and bare-metal environments
- Supports multiple container runtimes
- **100% Open source**, written in Go

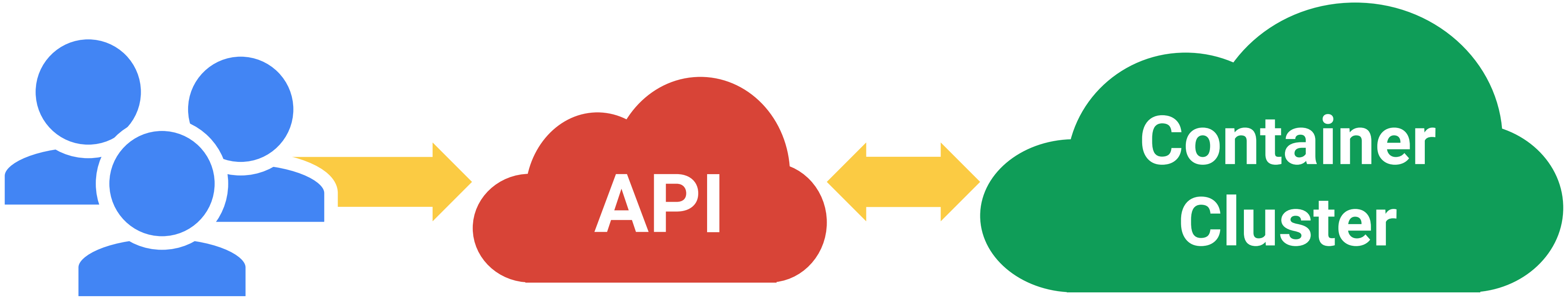
Manage <u>applications</u>, not machines

ENDOCODE

The 10000 foot view



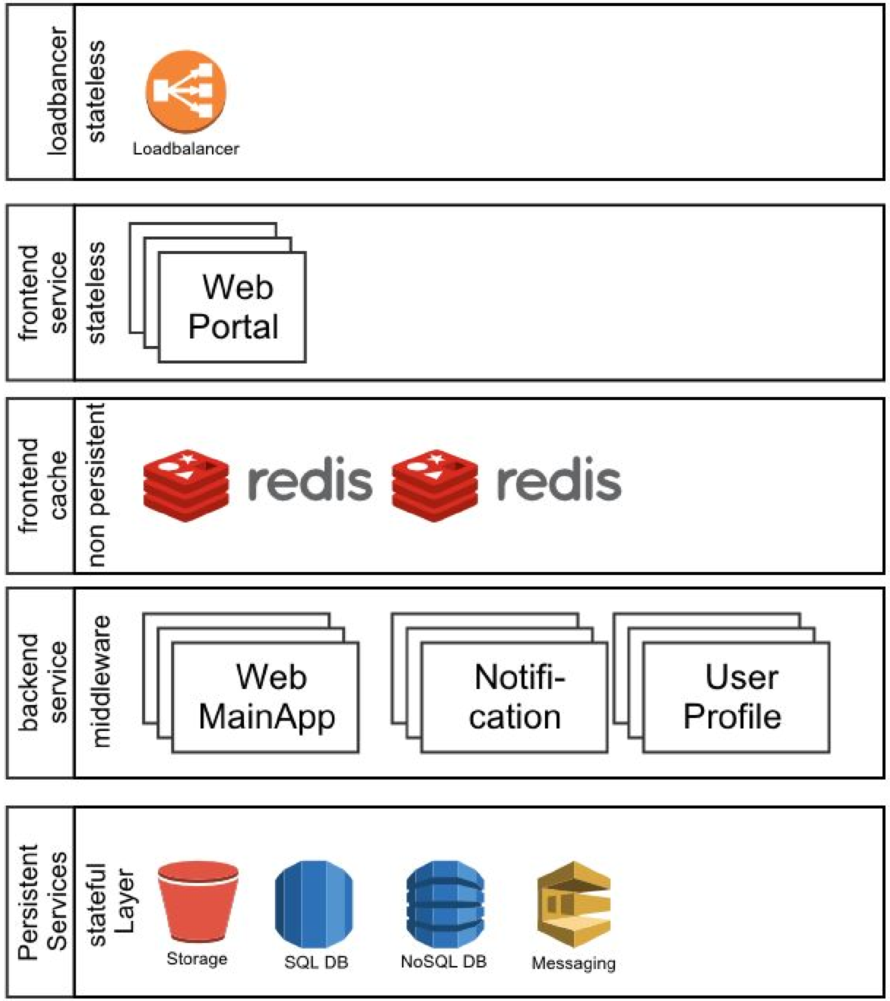
All you really care about



ARCHITECTURE

- strict layered architecture
  - separation of stateless services “cattle”
  - and persistent data “pets”

- inside the pods
  - developers are free to use what they want
  - contract is binding to the outside



ENDOCODE

Kubernetes Building Blocks

- Pods
- Replication Controller
- Services
- Persistence
    - External Ressources
    - Databases
        - Local Disk
        - Cloud Storage
        - Replication
        - Backup

ENDOCODE

STATELESS AND STATEFUL SERVICES

- where to keep state? A trade-off
    - provider → lock-in
    - self-managed → overhead
- cattle, no pets
- mindset: ephemeral deployment units

ENDOCODE

FROM VMs TO PODS

OS instances $\Rightarrow$ microservices in Pods

- pods are containers sharing the same fate
    - created together
    - running on same node
    - terminationg together
    - one network address
    - shared volumes

ENDOCODE

FROM VMs TO PODS

VM cluster $\Rightarrow$ Pods running on Kubernetes

- cattle: stateless containers
- pets: databases

configuration management $\Rightarrow$ separation of build time and run time

ENDOCODE

GETTING STARTED WITH KUBERNETES

ENDOCODE

PREREQUISITES

- Networkprovider
- Storage
- Loadbalancer Provider
- Ops Use Cases
  - add node
  - delete node
  - replace failing node
  - update kubernetes
- installation
  - cloud provider
  - bare metal

ENDOCODE

PUBLIC VS PRIVATE CLOUD

- GKE
  - ready to rock
  - scales very well
- AWS
  - need to set up the Ops cases
  - rich set of services
    - less need for pets
    - replicated dabases

- Your Private Cloud
  - loadbalancer providers
    - implement
    - manual
  - lots of storage providers
    - nfs
    - ceph
  - bare metal
    - local disks
    - SSDs

ENDOCODE

LAST OPS TASKS

- Backups
    - replication
        - database
        - storage
    - snapshots
        -

- Monitoring
    - basic
    - business KPIs
        - prometheus
        - graphite
        -
- Alerting
    - based on monitoring
    -

ENDOCODE