inovex

HdM Workshop

Kubernetes

Johannes M. Scheuermann &
Maximilian Bischoff

Stuttgart, 13. Dezember 2018

Johannes M. Scheuermann

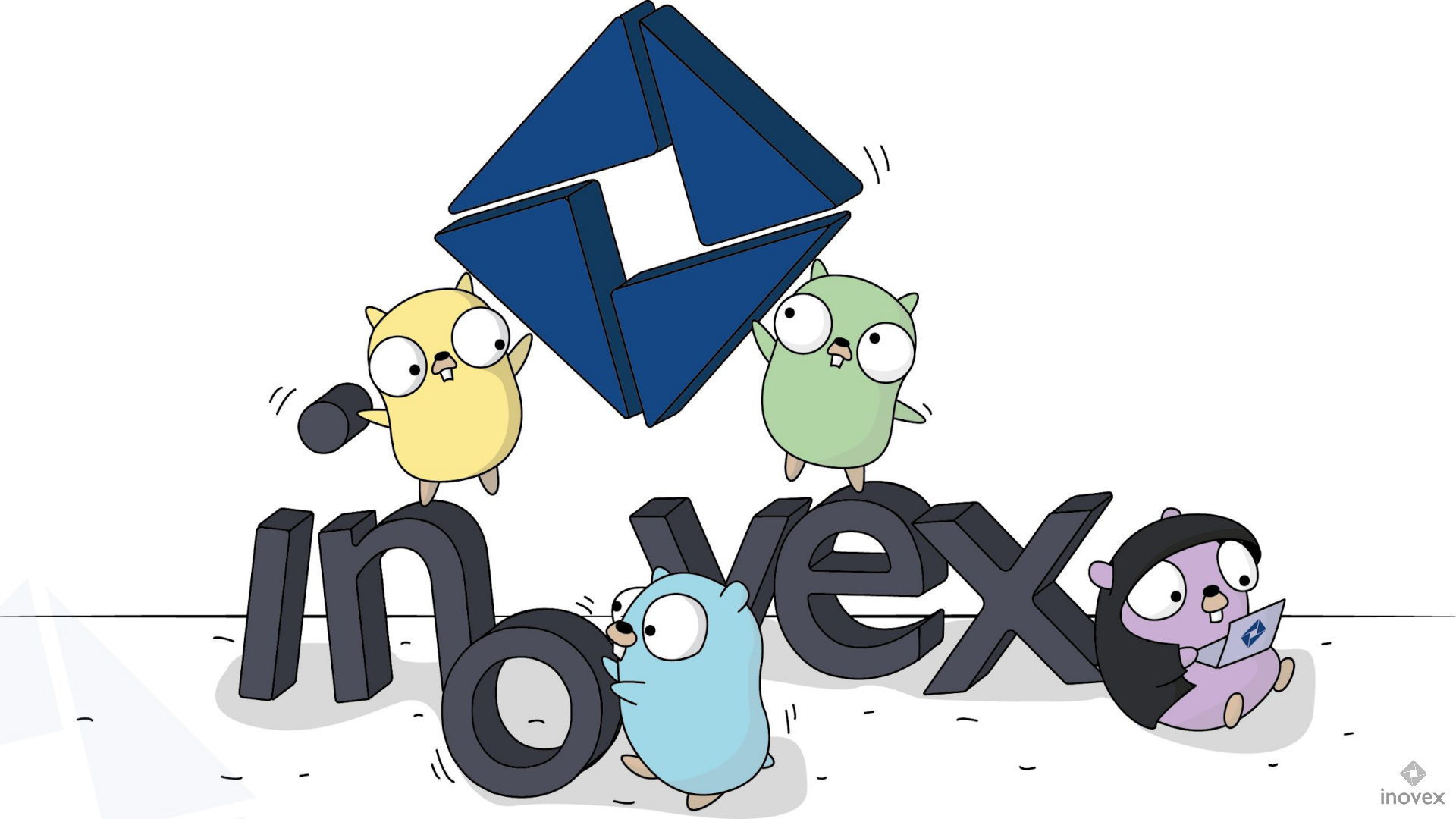
Macht Dinge in der Wolke

- Team ITO since 2014
- Kubernetes since 2014
- Certified Kubernetes Admin
- LF authorized Kubernetes Admin trainer
- Cloud technologies
- @johscheuer

Maximilian Bischoff

Macht Dinge in der Wolke

- Team ITO since 2018
- Kubernetes since 2018
- Masterthesis@HdM:
  Kubernetes Policy Validierung
- Cloud technologies


inovex

Agenda

- Containers
- Kubernetes - a high level overview
- Kubernetes concepts
- Kubernetes Auto-Scaling

- Hands on part: https://github.com/johscheuer/inovex_classes
- hdm_workshop_2018


What about you?

inovex

(Linux) Containers - What's inside?

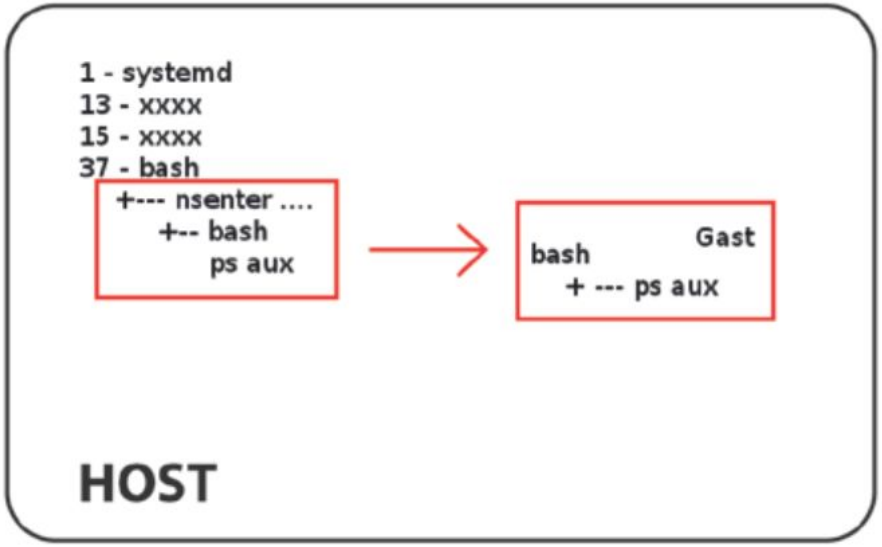
- chroot
- namespaces
- cgroups
- layered filesystem
- capabilities

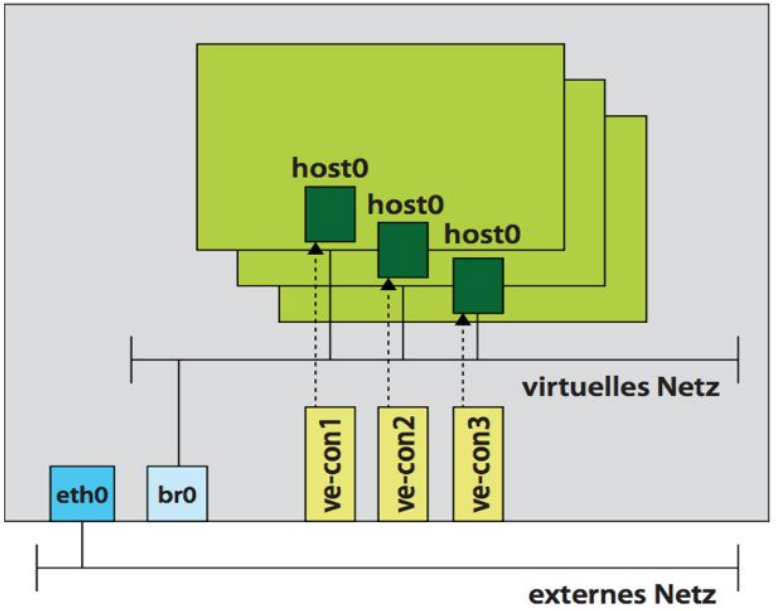
Chroot

```
$ tree /home/vagrant/
/home/vagrant/
├── jail
│   ├── bin
│   │   ├── bash
│   │   ├── ls
│   │   └── tree
│   ├── inside.jail
│   ├── lib
│   │   └── x86_64-linux-gnu
│   │       ├── libacl.so.1
│   │       ├── libattr.so.1
│   │       ├── libc.so.6
│   │       ├── libdl.so.2
│   │       ├── libpcre.so.3
│   │       ├── libselinux.so.1
│   │       └── libtinfo.so.5
│   └── lib64
│       └── ld-linux-x86-64.so.2
└── outside.jail

5 directories, 13 files
$
```

```
$ sudo chroot /home/vagrant/jail /bin/bash
$ tree /
/
|-- bin
|   |-- bash
|   |-- ls
|   `-- tree
|-- inside.jail
|-- lib
|   `-- x86_64-linux-gnu
|       |-- libacl.so.1
|       |-- libattr.so.1
|       |-- libc.so.6
|       |-- libdl.so.2
|       |-- libpcre.so.3
|       |-- libselinux.so.1
|       `-- libtinfo.so.5
`-- lib64
    `-- ld-linux-x86-64.so.2

4 directories, 12 files
$
```

Namespaces



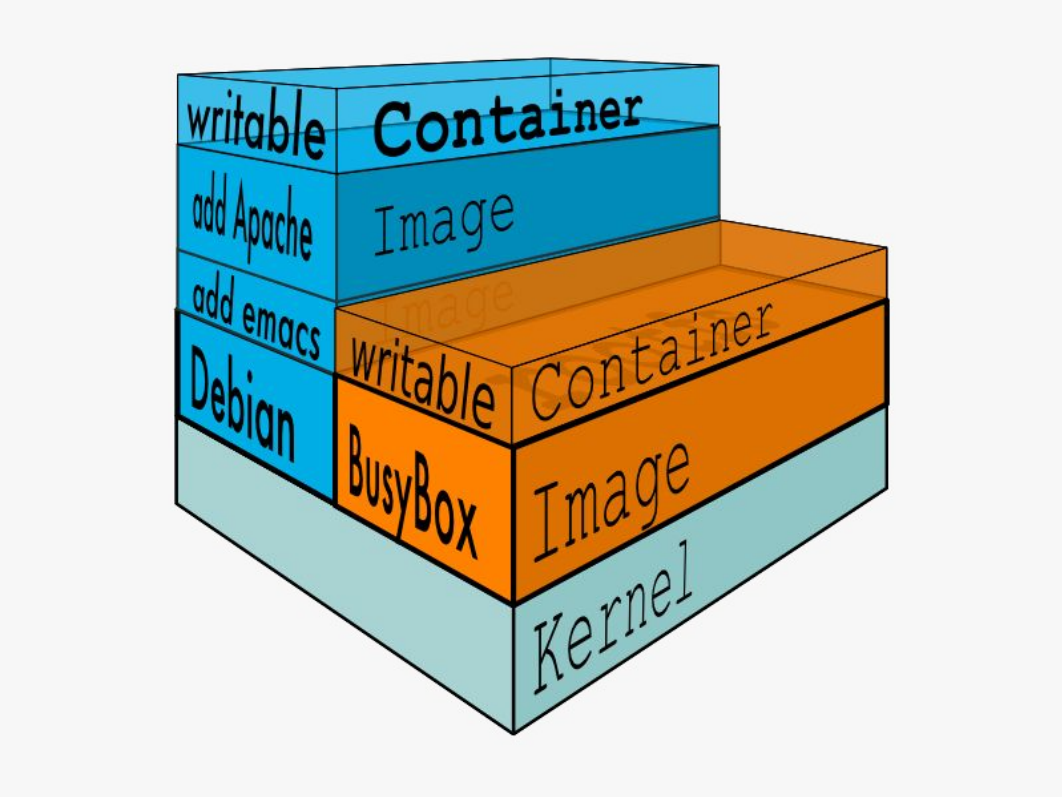


inovex

cgroups

- Hierarchy
- Resource limiting
  - CPU
  - Memory
  - …
- Prioritization
- Accounting
- Control
  - Freeze groups of processes (checkpointing)

Layered filesystem



inovex

Capabilities

- Two categories privileged/unprivileged process
  - privileged => UID == 0
  - unprivileged 0> UID != 0
- Privileged processes bypass kernel permission checks
- Since Linux Kernel 2.2 -> capabilities
  - Per-Thread attribute
- Some capabilities includes other capabilities
  - e.g. “CAP_SYS_ADMIN”

Source: http://man7.org/linux/man-pages/man7/capabilities.7.html

Docker - What's special?

- Uses "proven" technologies
- Made them developer friendly
  - Not only for Kernel hackers
  - Still most features are hard to understand
- Made the Container image distribution easier
  - Docker Hub
  - Docker images
  - "docker run -ti hello-world"

kubernetes

by Google

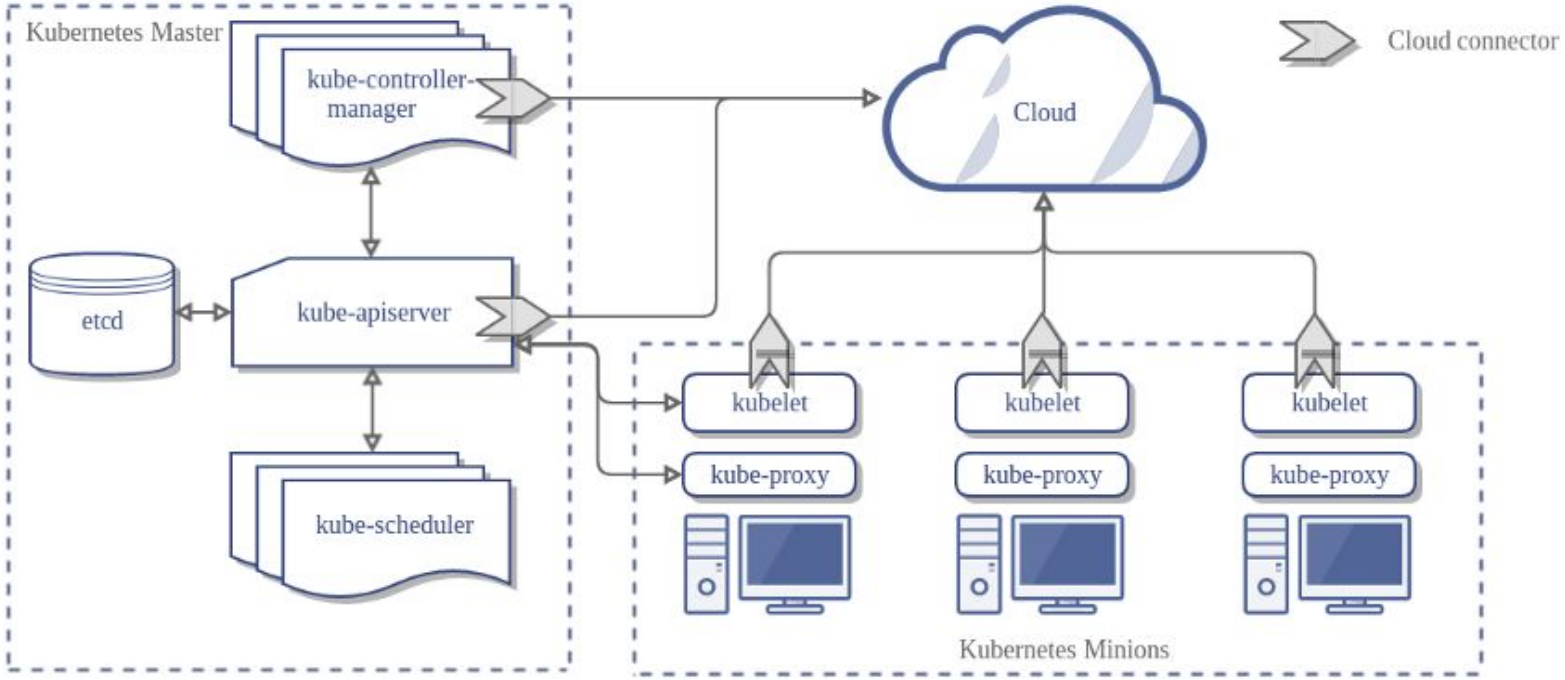
History of Kubernetes

- Built with lessons learned from Google
- Borg (https://research.google.com/pubs/pub43438.html)
- Omega (https://ai.google/research/pubs/pub41684)
- 2014: Open sourced
- 2015: Version 1.0
- Currently: v1.13.0


How to orchestrate a fleet?

“Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications”

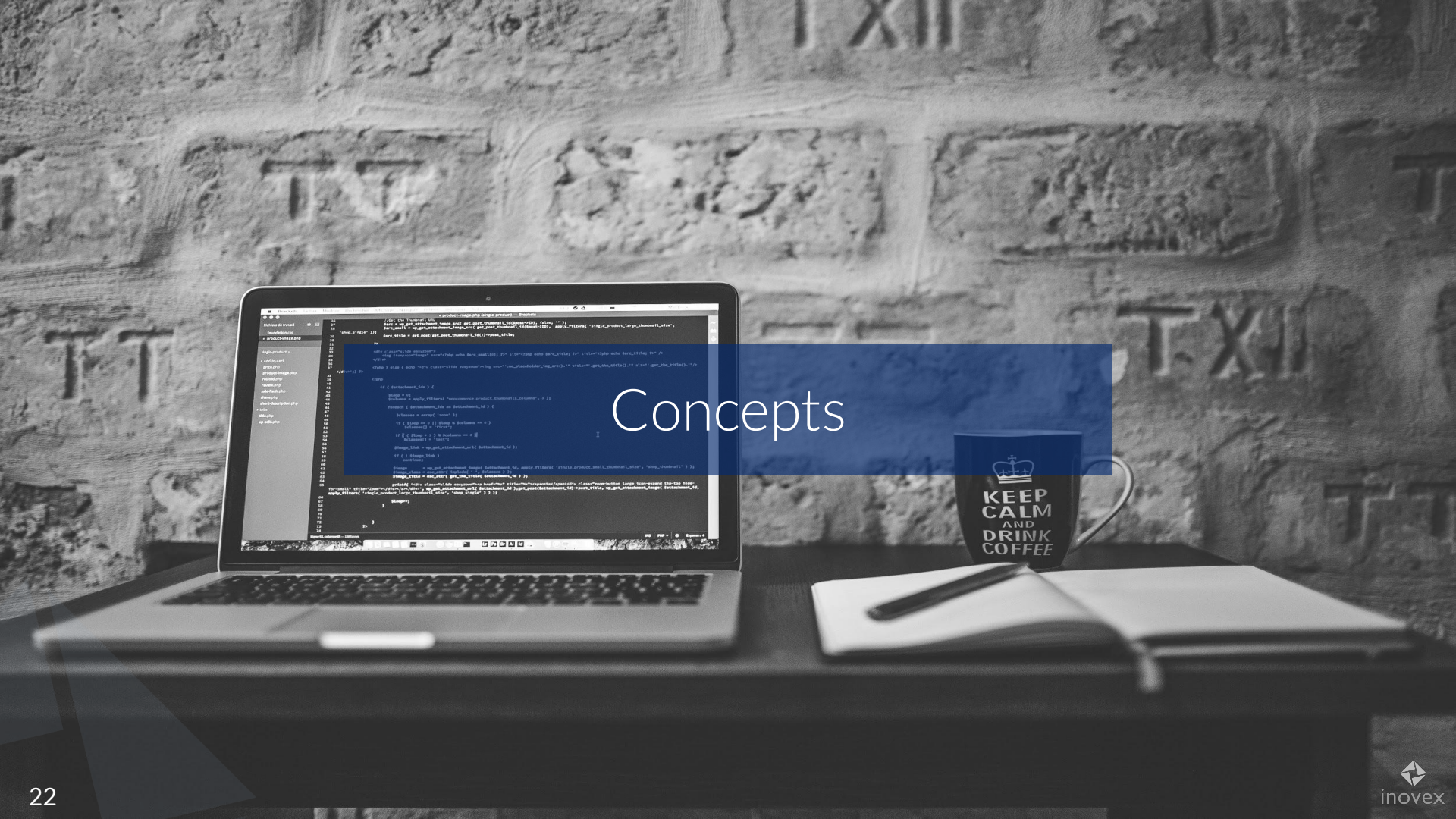
What is Kubernetes?



Architecture

- Each component has a dedicated task
  - Do one thing and do it well
- Each component can be built HA
- Each component can fail without breaking the system
  - except for the API server
  - except for the etcd

Working with Kubernetes
Hands-on



Kubernetes concepts

- Deployments
- ReplicaSets
- Pods
- Container
- DamonSets
- Jobs
- Services
- Ingress
- ...

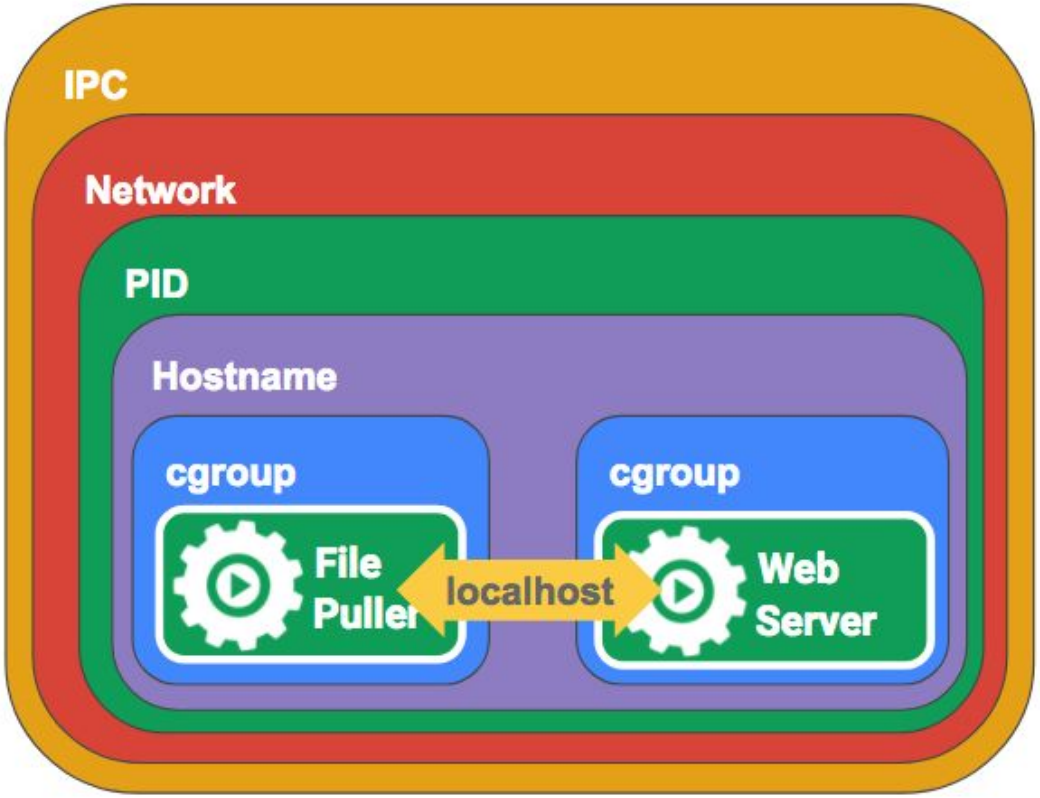
Deployments

- High-level abstraction
- Deployments are done on the server-side
- Generates and maintains ReplicaSets
- Works with labels
- Control-loop

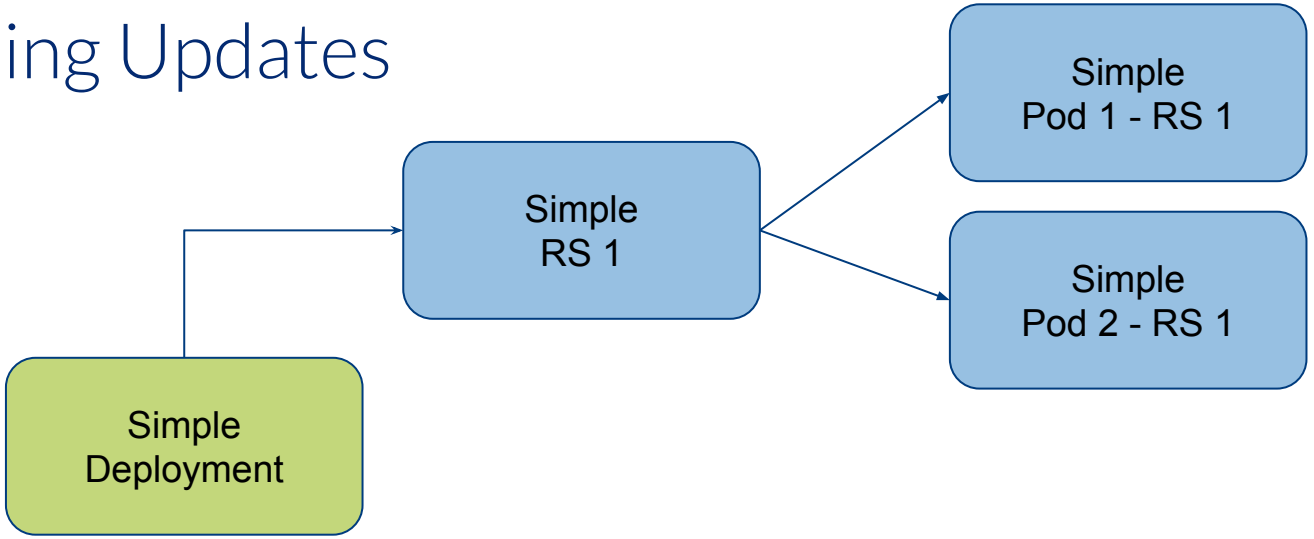
ReplicaSet

- Manages one or more Pods
- Control loop
- Ensures that enough replicas are available
- Works with labels
- Can be used without Deployments

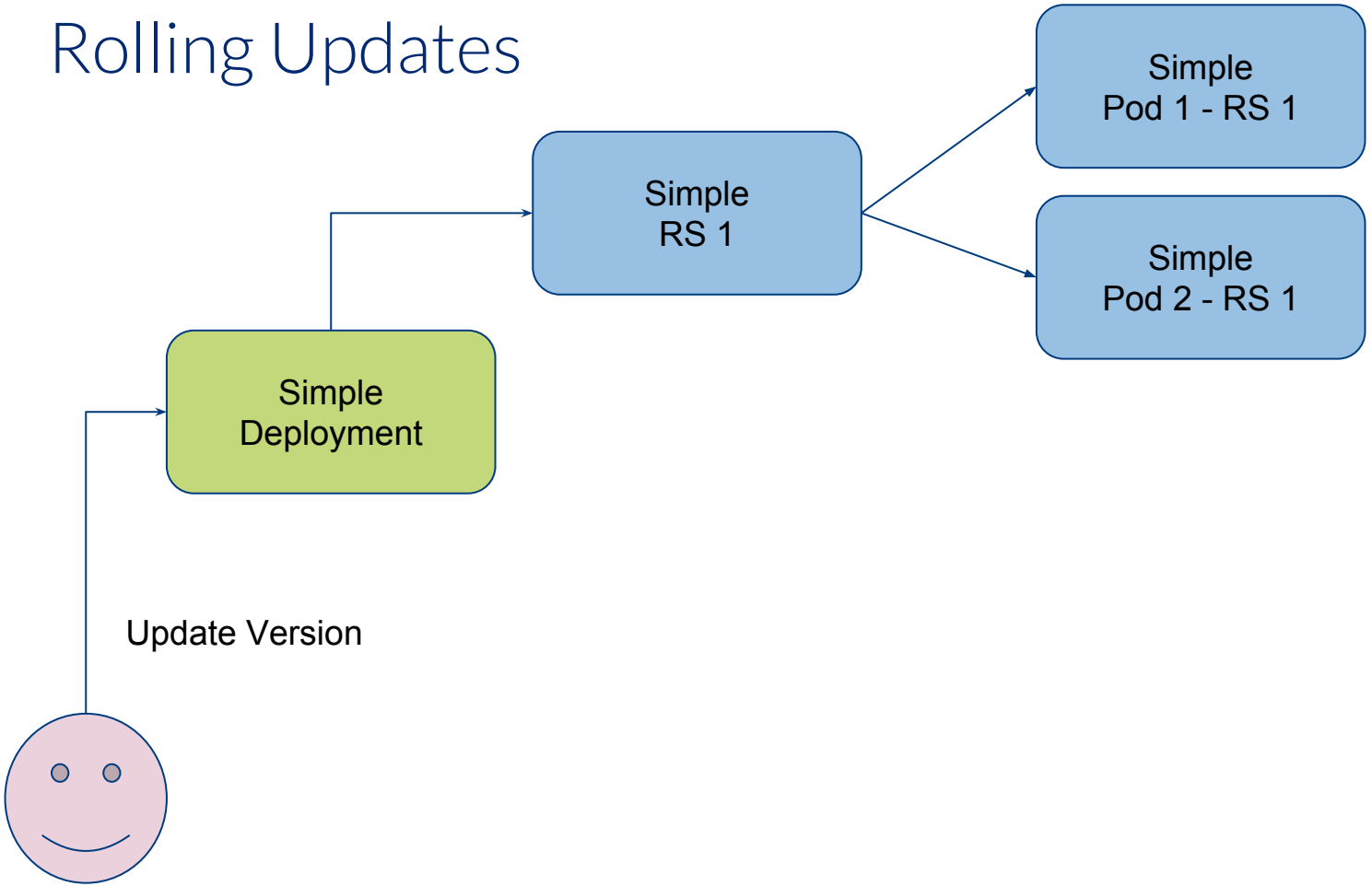
Pods


IPC
Network
PID
Hostname
cgroup
File
Puller
localhost
cgroup
Web
Server

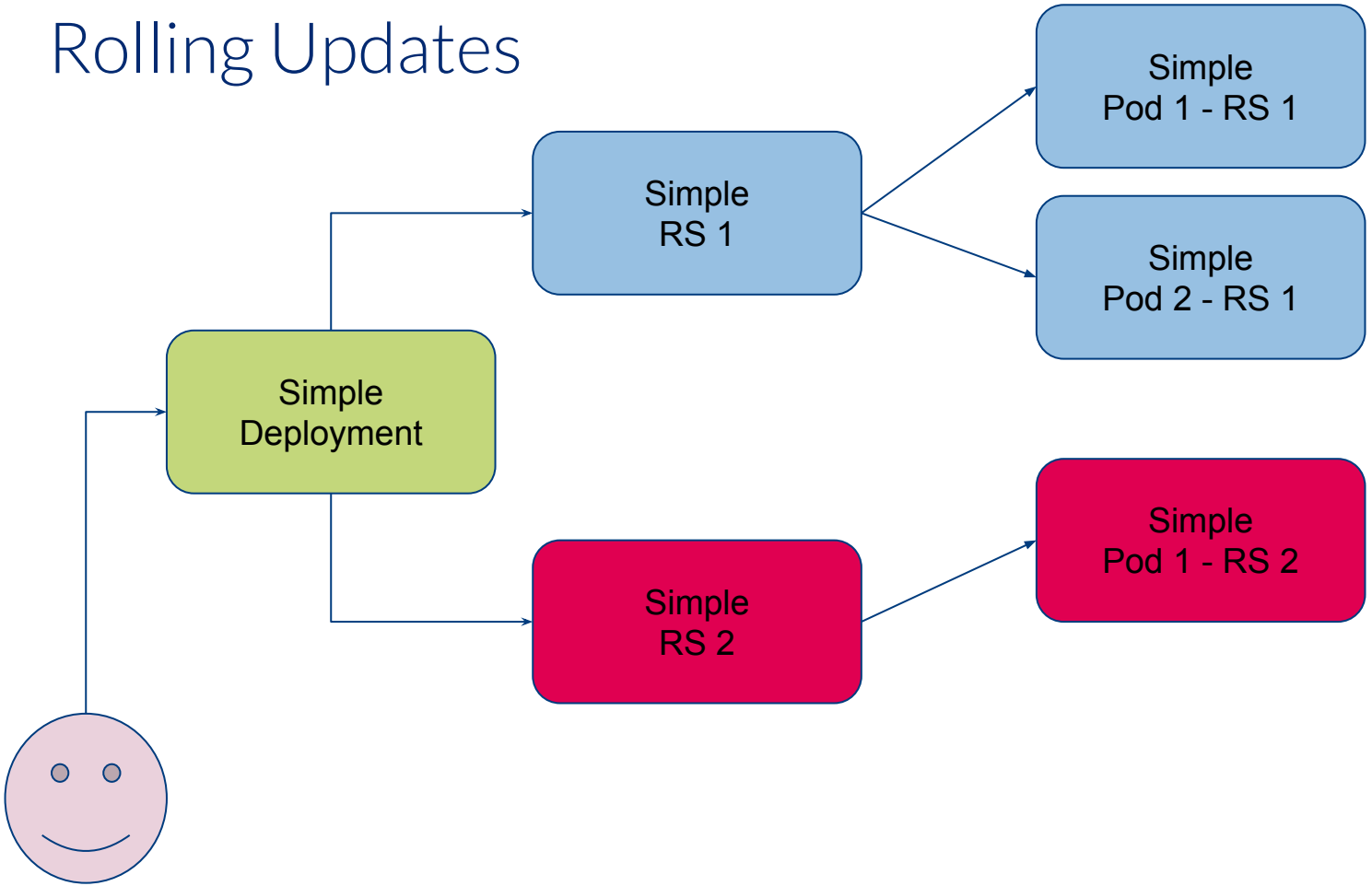
Rolling Updates



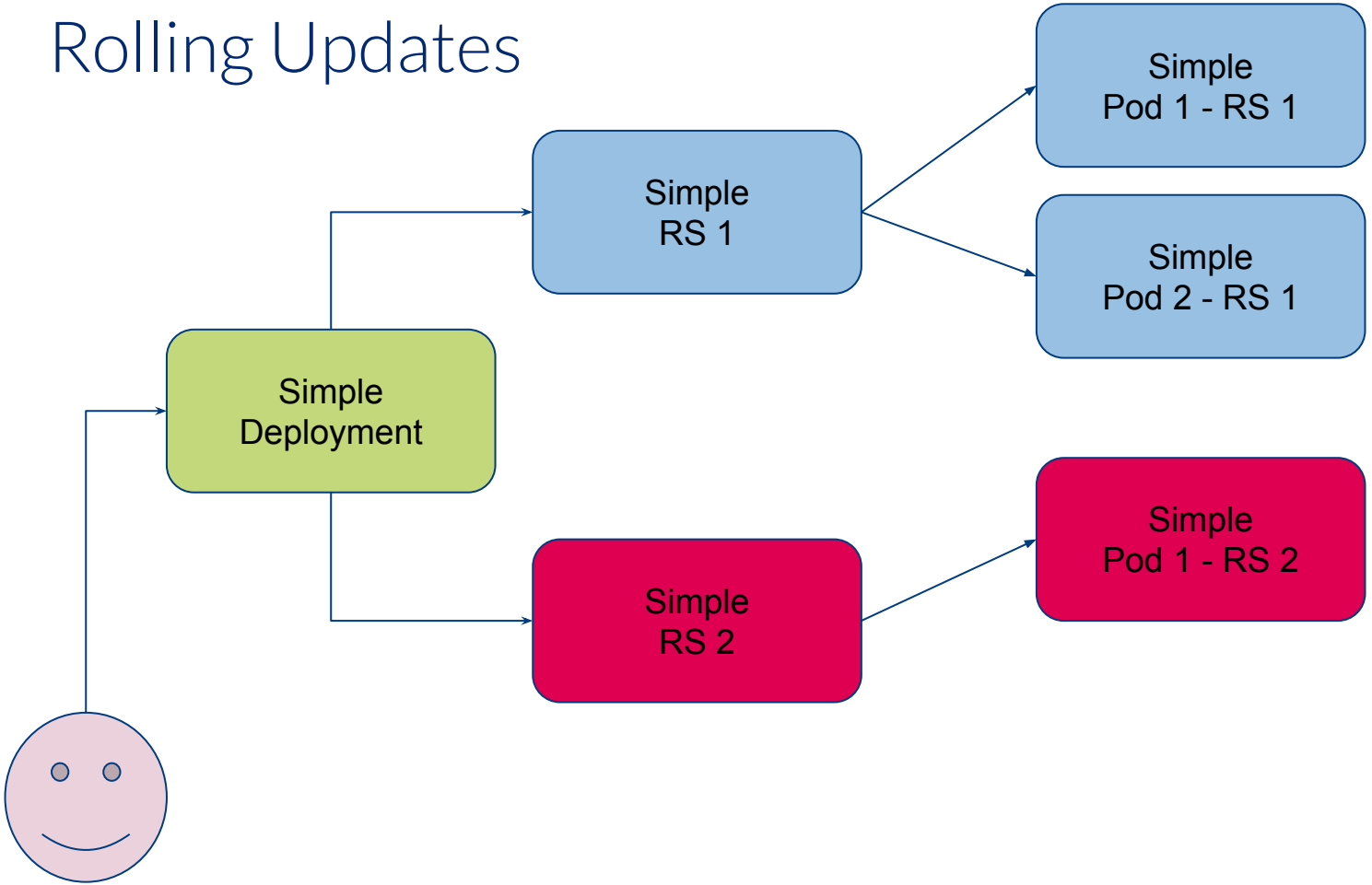
Rolling Updates



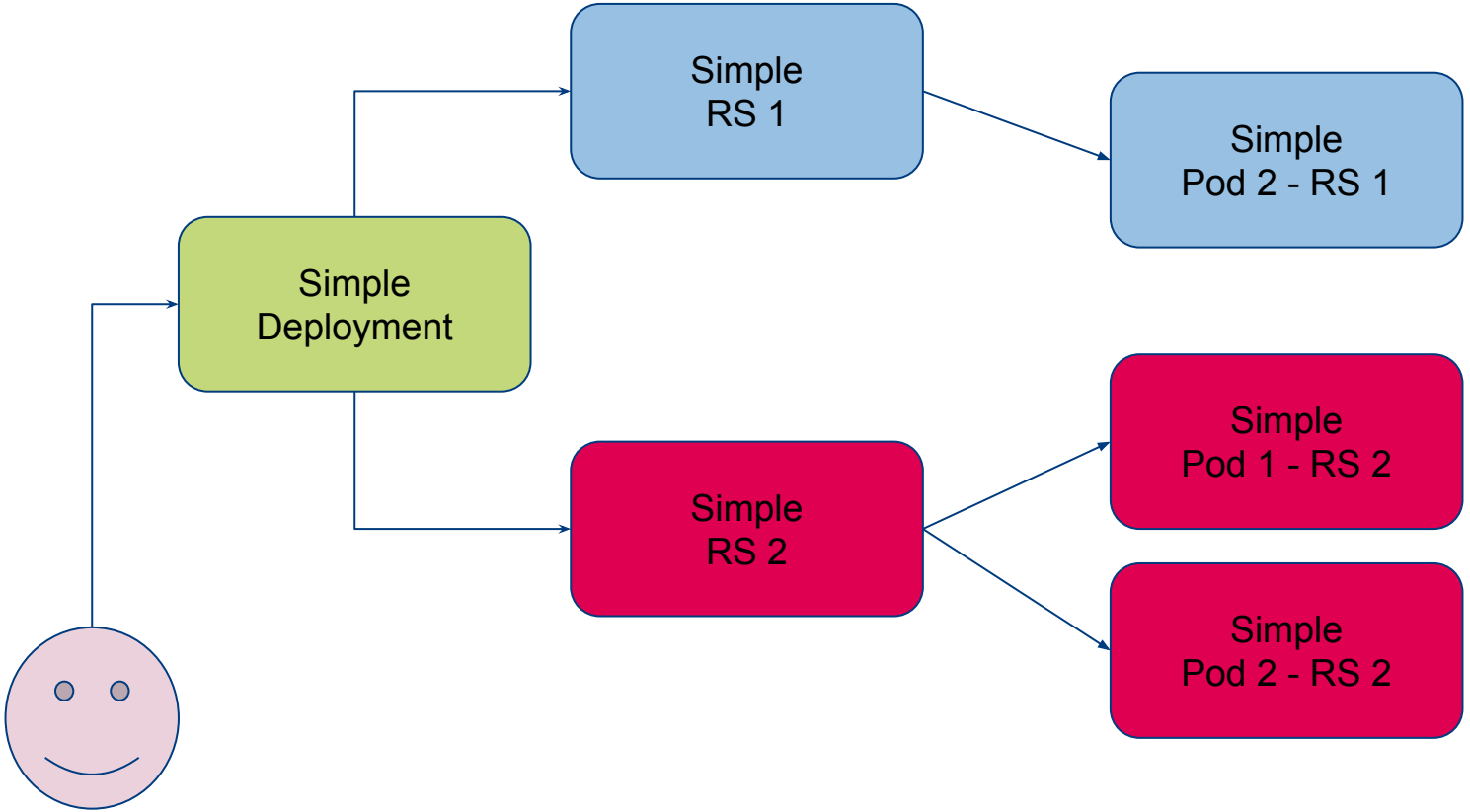
Rolling Updates



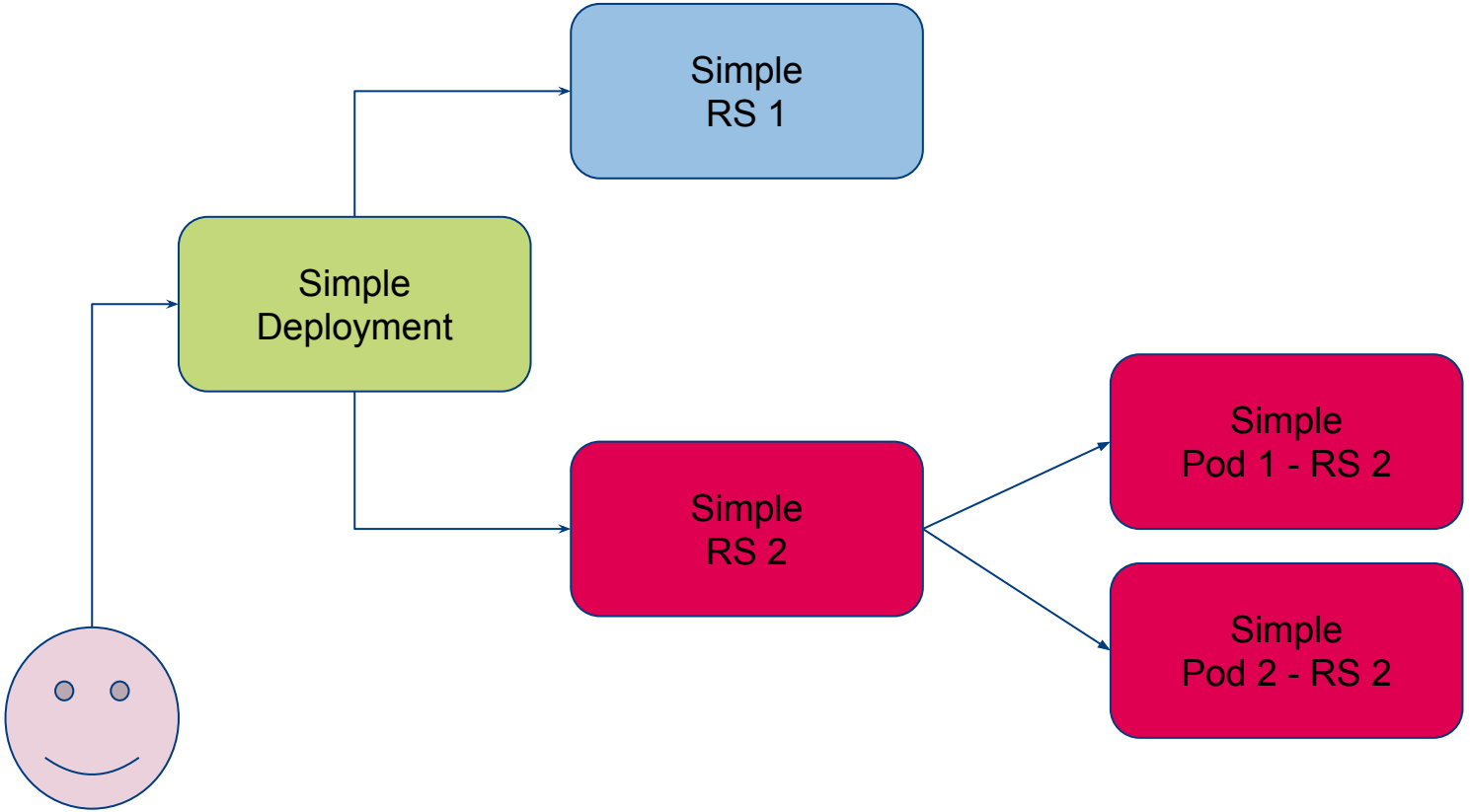
Rolling Updates



Rolling Updates



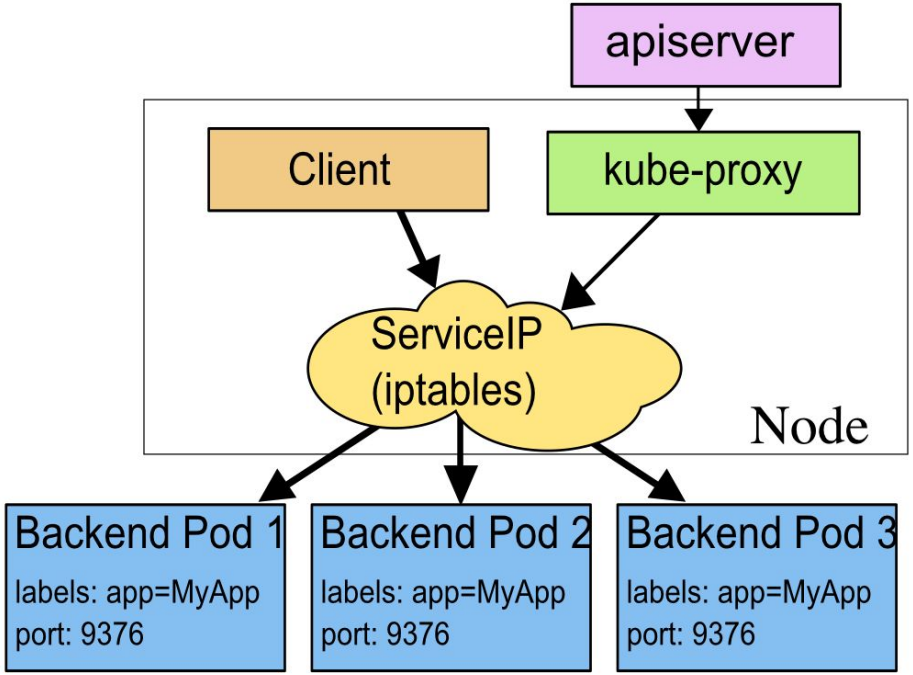
Rolling Updates



Service

- Provides a stable entrypoint inside the cluster
- Can expose services to the rest of the world
  - Different types
- Select endpoints based on labels
- With a CoreDNS a DNS entry will be created
  - Only internally

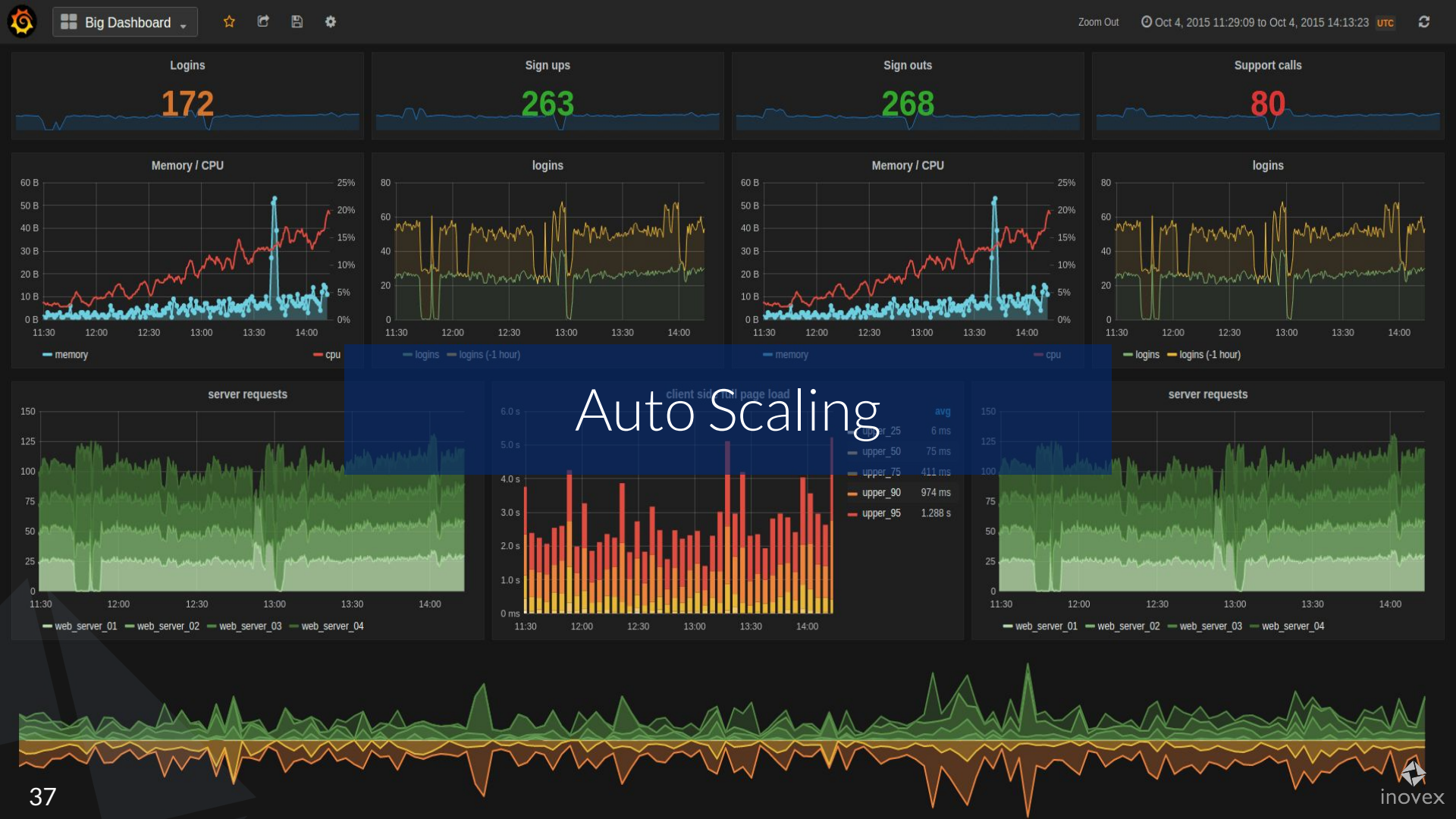
Service



Namespaces

- Logical separation of a cluster
  - e.g. dev/prod
- Can be used for “weak” Multi-Tenancy
- Allows to restrict resource usage (Quotas)
- With RBAC different permissions can be granted
- Default namespaces are present on each cluster

Hands on
Kubernetes concepts


Big Dashboard
Zoom Out
Oct 4, 2015 11:29:09 to Oct 4, 2015 14:13:23 UTC
Logins
172
Sign ups
263
Sign outs
268
Support calls
80
Memory / CPU
logins
Memory / CPU
logins
memory
cpu
logins
logins (-1 hour)
server requests
client side full page load
server requests
avg
upper_25 6 ms
upper_50 75 ms
upper_75 411 ms
upper_90 974 ms
upper_95 1.288 s
web_server_01
web_server_02
web_server_03
web_server_04
Auto Scaling

What do we need to auto scale?

Metrics

- Are collected by the “Metrics Server”
- Collects metrics for each node
- Collects metrics for each pod/container
- Needs to be installed additionally
- Works with the API Server Aggregation Layer
  - Metrics Server is an own API server
  - We can access the Metrics Server over the API Server

Hands on Autoscaling

Recap

What we did

- Used minikube for testing Kubernetes locally
- Played around with complete Manifests
- Looked at Kubernetes from a user perspective
- Basic Concepts
  - Namespaces
  - Deployments
  - Labels
  - Auto-Scaling

What we didn’t cover

- Networking (CNI, ingress, ...)
- Security (Policies, RBAC, …)
- Deployments over CI/CD
- Control Plane modifications (Scheduler tuning, ...)
- Logging
- Monitoring (only rudimentary)
- Use Docker to build your own container
- and much more ...

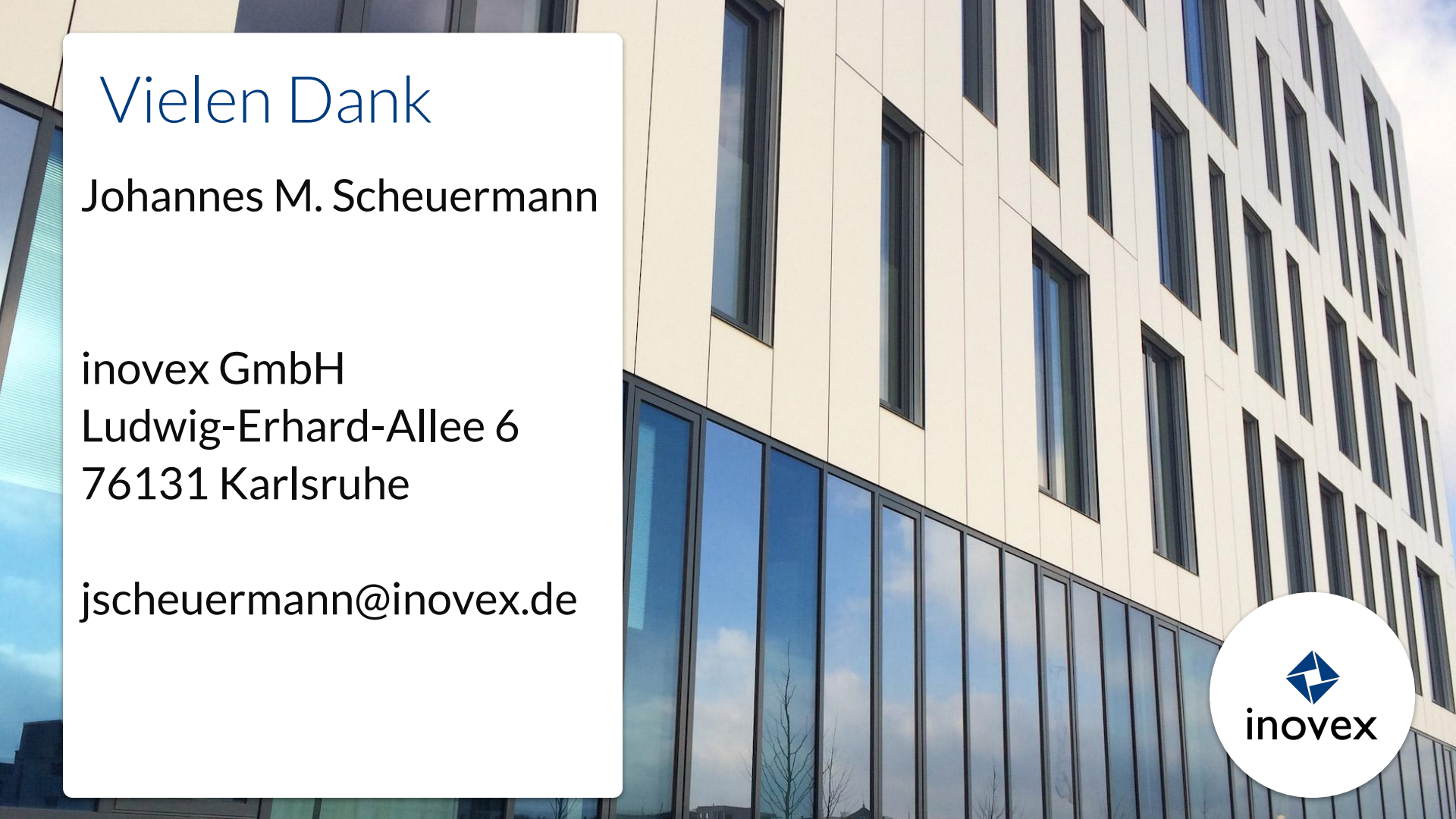
Use Kubernetes for your own project

- Managed service: GKE / AKS / EKS / ....
- Setup by your own (use kubeadm)
- The “hard way”:
  - https://kubernetes.io/docs/setup/scratch/
  - https://github.com/kelseyhightower/kubernetes-the-hard-way

- Thing about networking, storage, …

- https://github.com/johscheuer/inovex_classes/tree/master/class_4

Questions?



Vielen Dank

Johannes M. Scheuermann

inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe

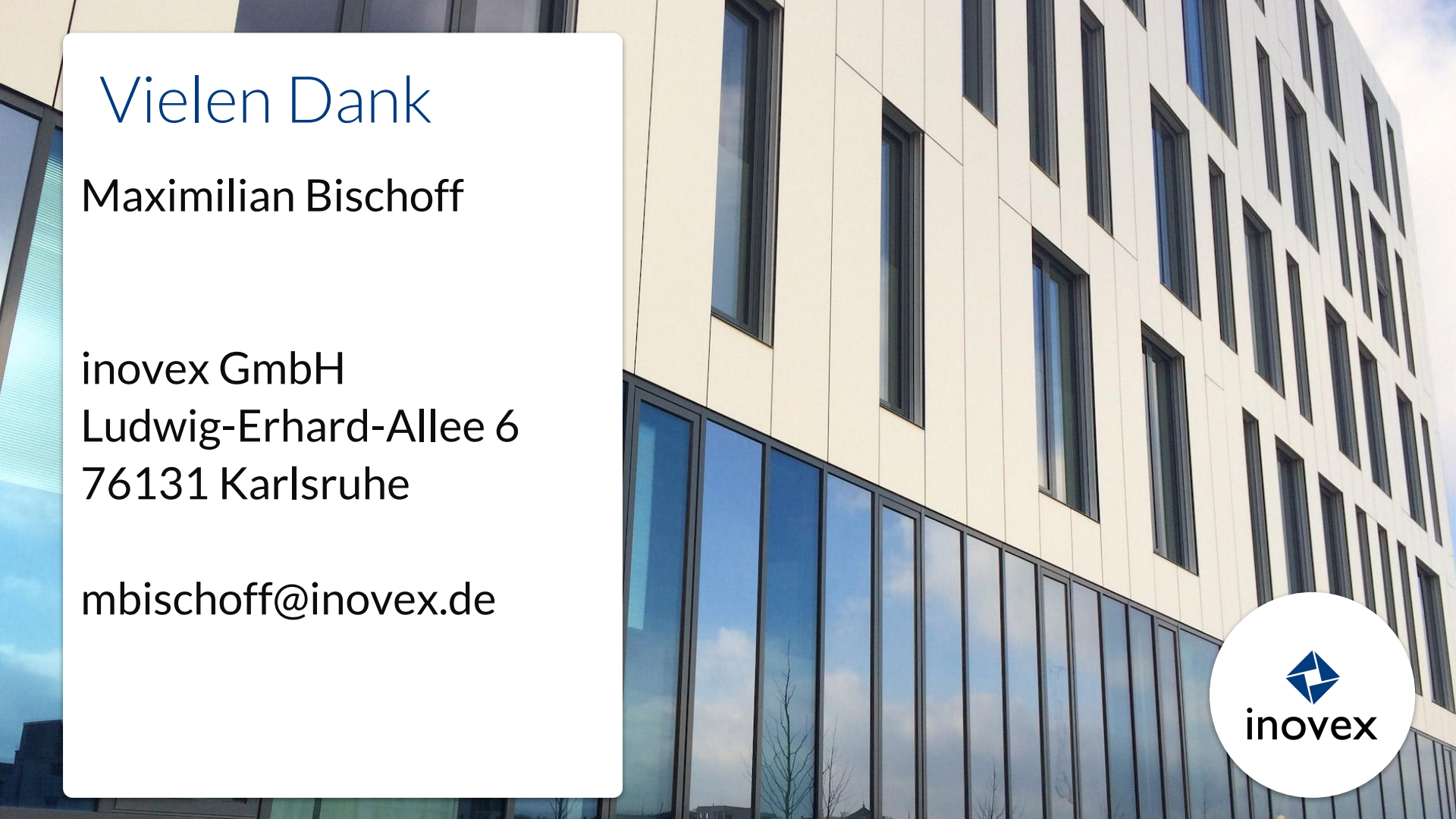
jscheuermann@inovex.de

Vielen Dank

Maximilian Bischoff

inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe

mbischoff@inovex.de



Reading list

- https://lwn.net/Articles/689856
- http://man7.org/linux/man-pages/man7/namespaces.7.html
- https://www.ianlewis.org/en/almighty-pause-container
- https://www.kernel.org/doc/Documentation/cgroup-v1/cgroups.txt
- https://kubernetes.io/docs/concepts/architecture/cloud-controller/
- https://coreos.com/etcd/docs/latest/learning/api_guarantees.html
- https://www.ianlewis.org/en/what-are-kubernetes-pods-anyway
- https://github.com/johscheuer/inovex_classes
- https://www.booleanworld.com/depth-guide-iptables-linux-firewall