inovex

inovex classes

Kubernetes

Johannes M. Scheuermann &
Timo Heinrichs

Köln, 27. September 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

Timo Heinrichs

Macht Dinge in der Wolke

- Team ITO since 2017
- Docker since v0.9
- Kubernetes since 2017
- Cloud technologies
- @theinrichs

Agenda

- Recap
- Kubernetes concepts
- Volumes and data

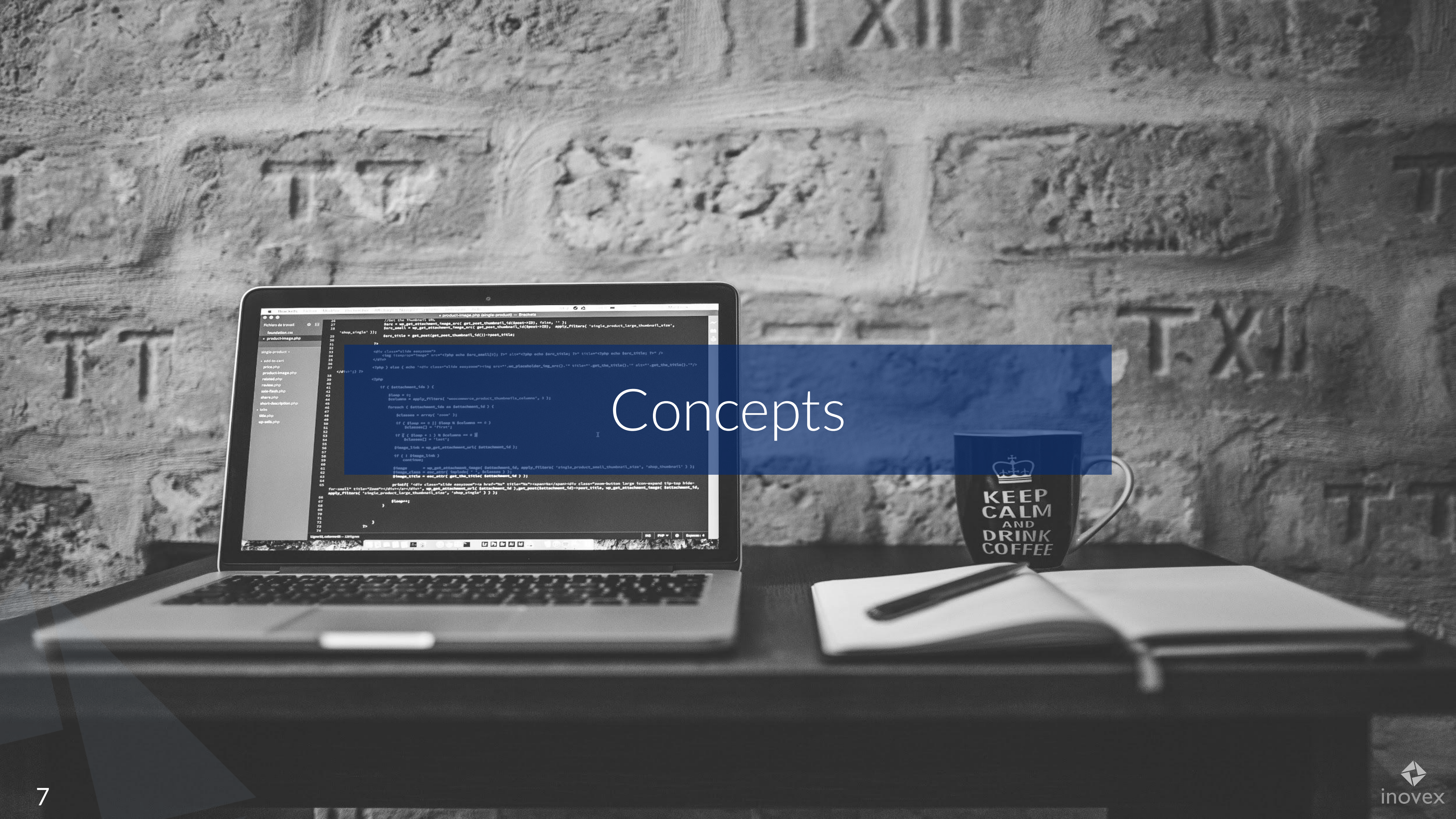
- Hands on part: https://github.com/johscheuer/inovex_classes

feedback

What we did

- History of Containers
- History of Kubernetes
- Kubernetes

- Feedback
  - “Micro” Service deployen
  - Autoscaling
  - Fail-Over



Concepts

Kubernetes concepts

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

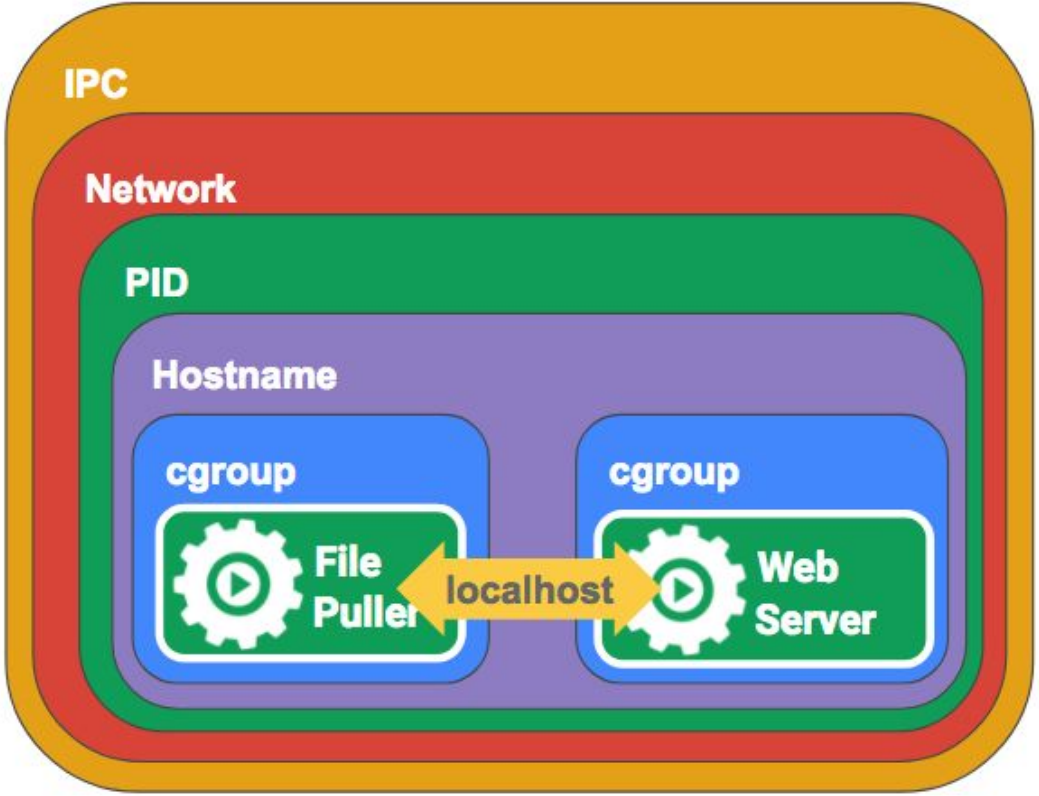
Deployments

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

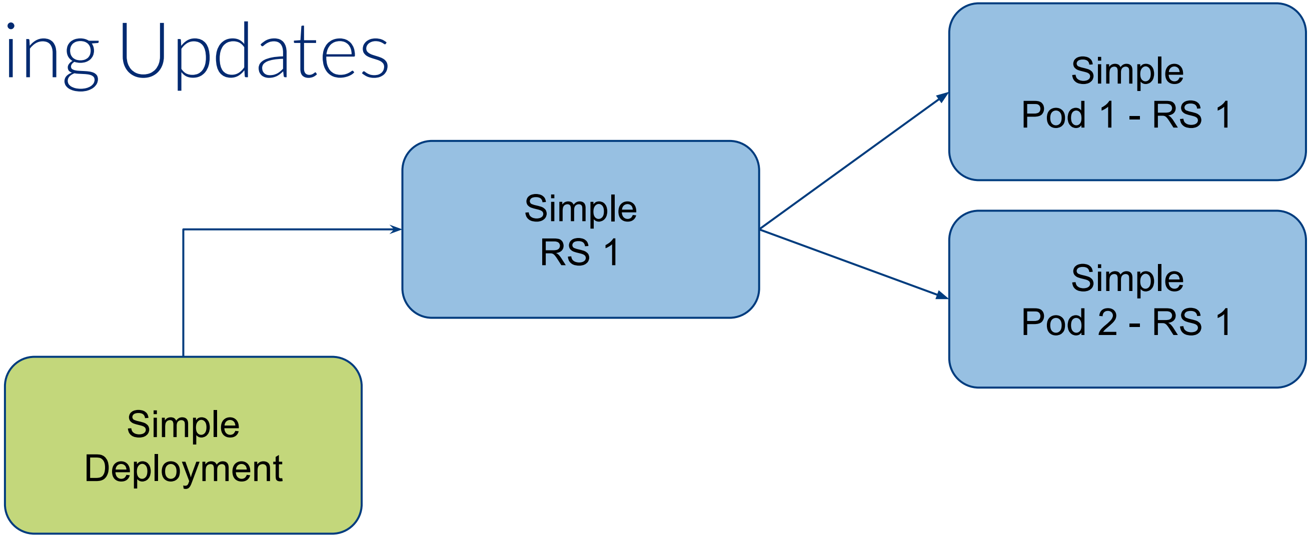
ReplicaSet

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

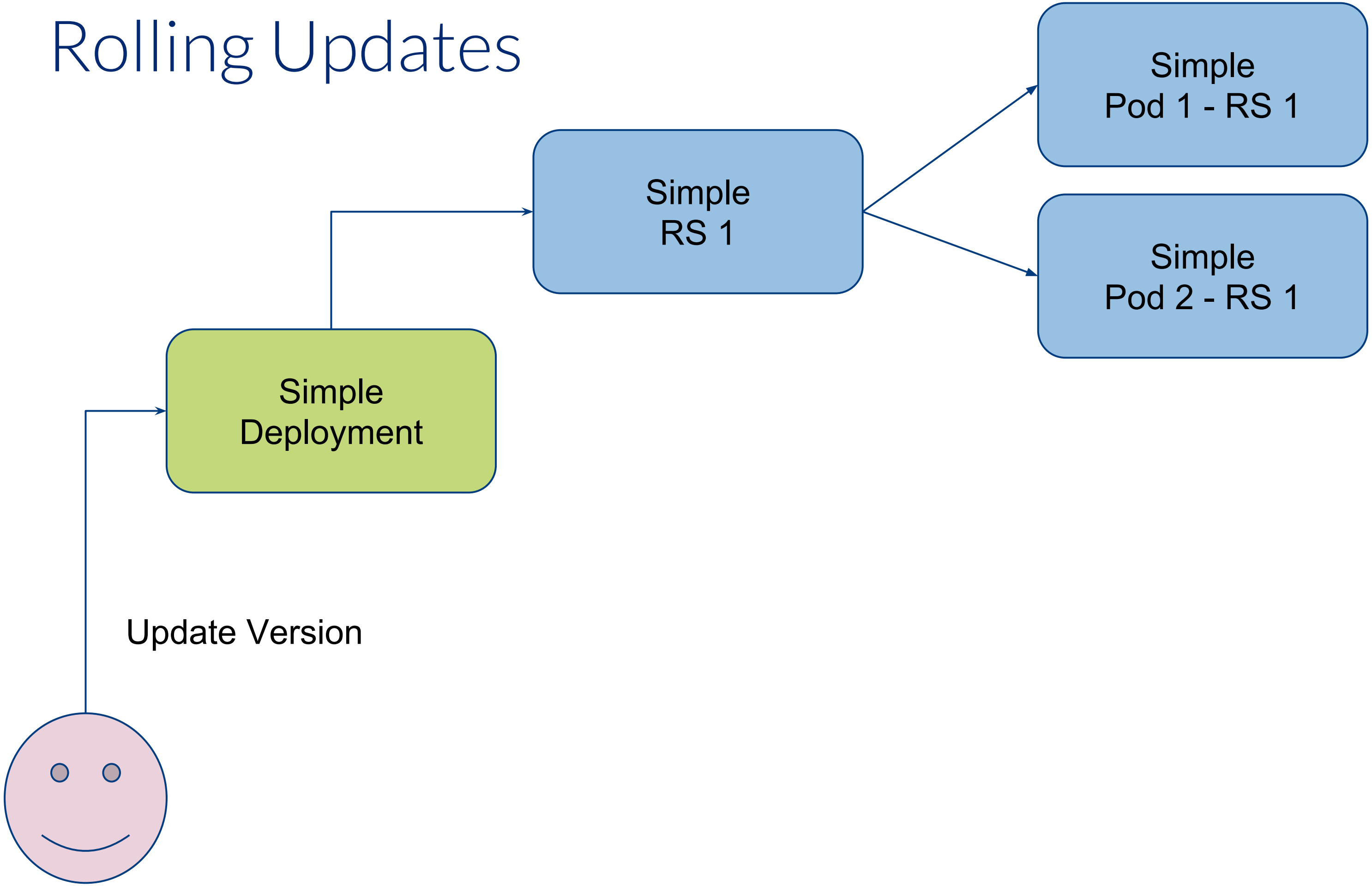
Pods (recap)


IPC
Network
PID
Hostname
cgroup
File Puller
localhost
cgroup
Web Server

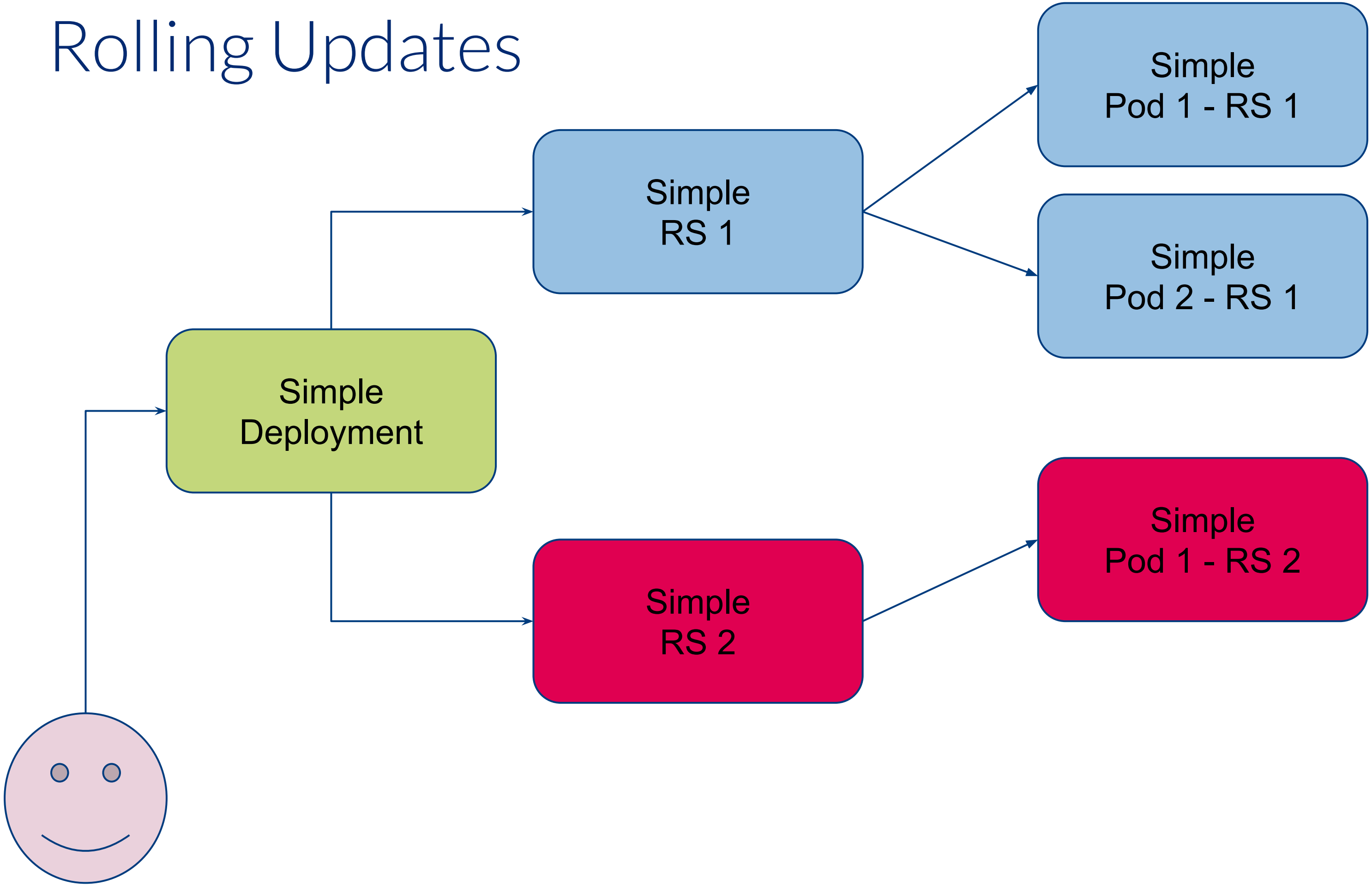
Rolling Updates



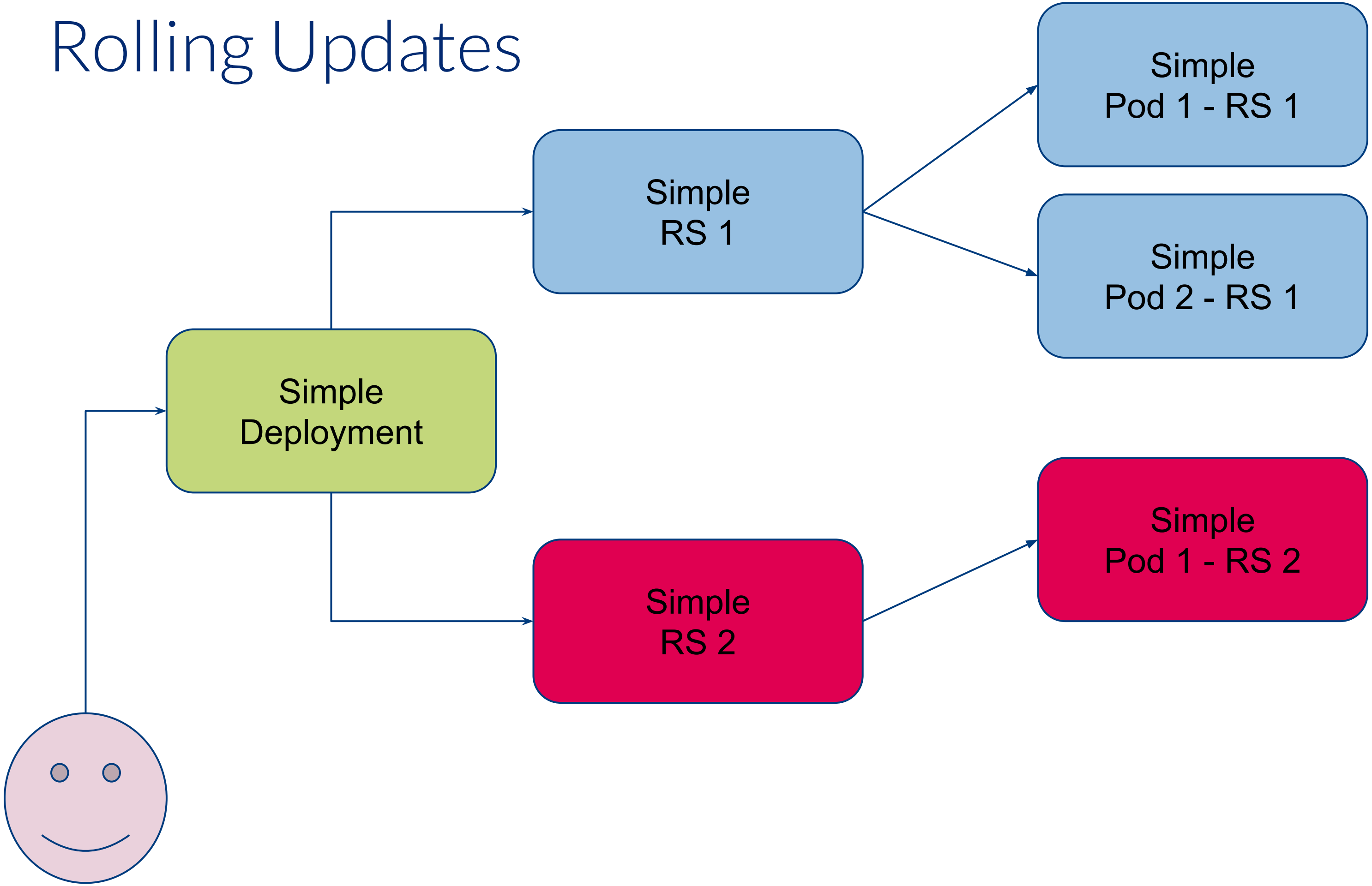
Rolling Updates



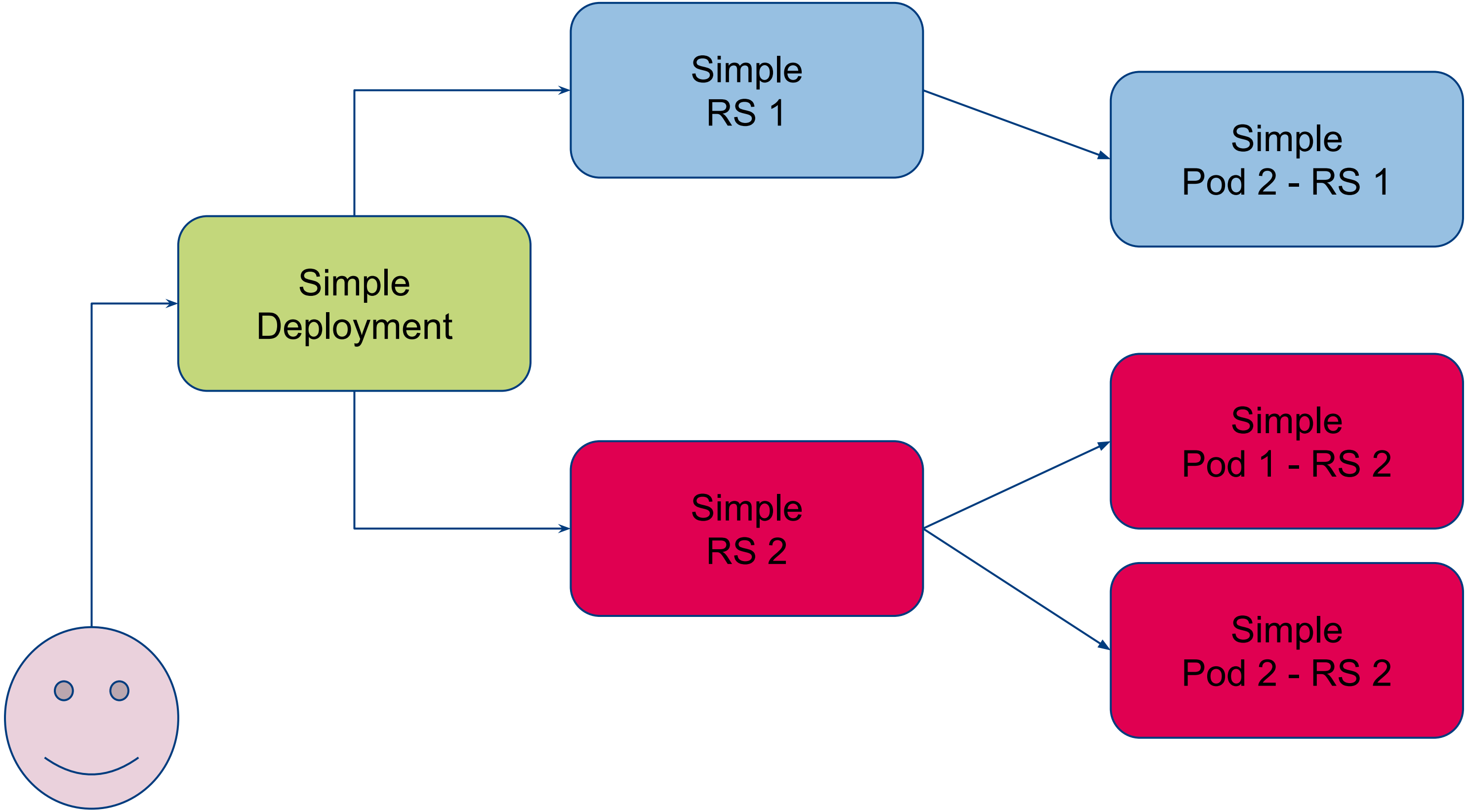
Rolling Updates



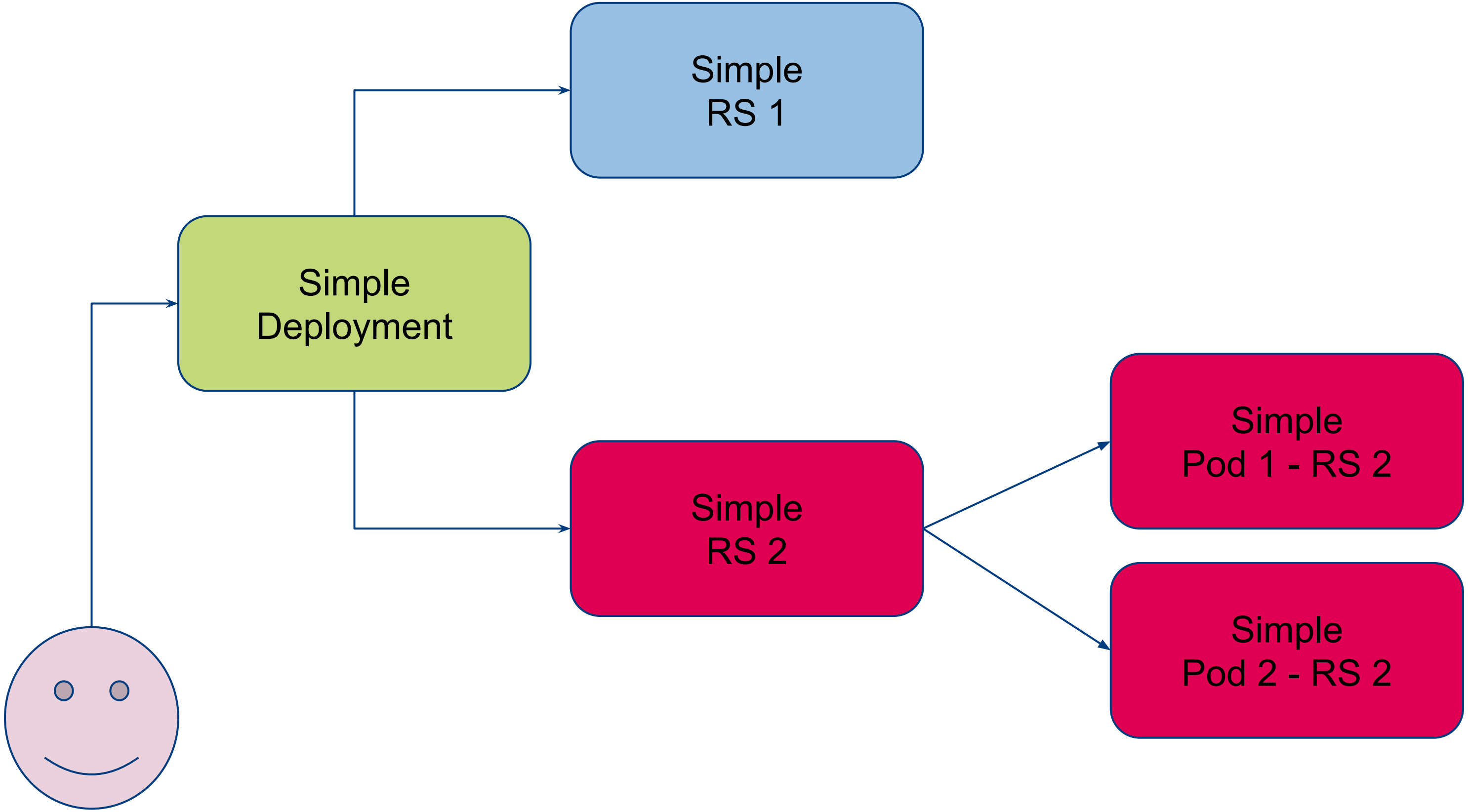
Rolling Updates



Rolling Updates



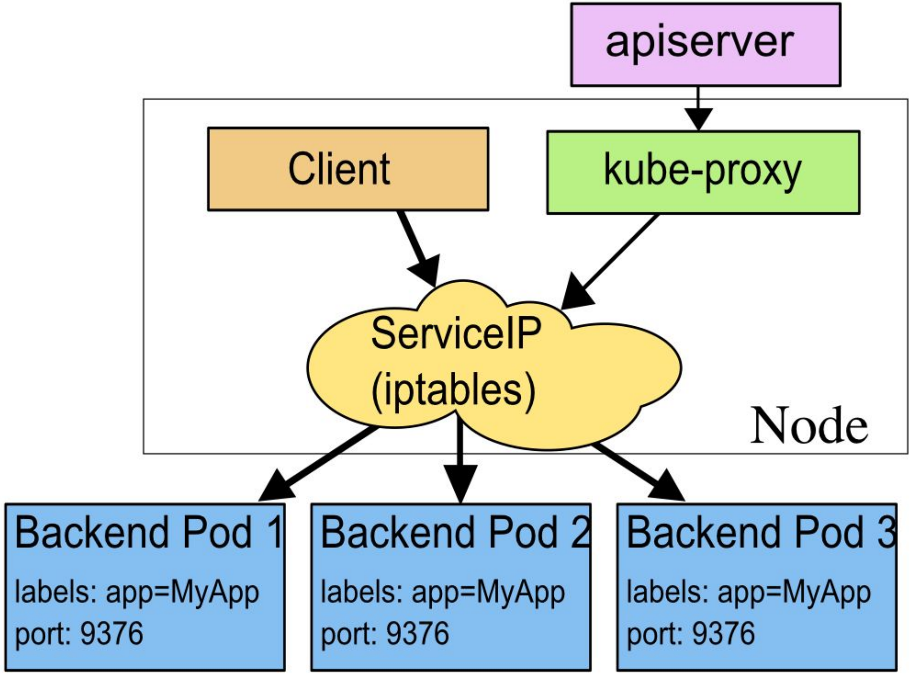
Rolling Updates



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/KubeDNS a DNS entry will be created
  - Only internally

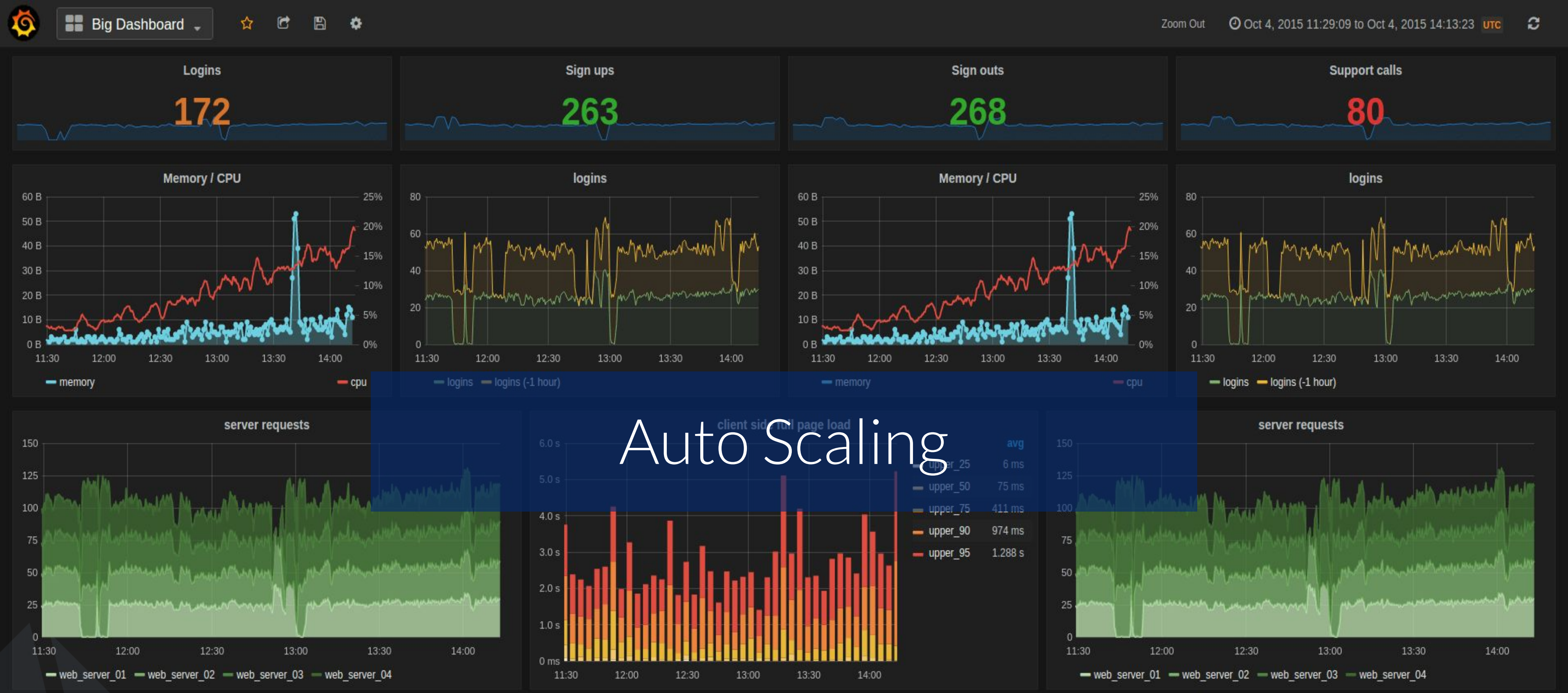
Service


apiserver
Client
kube-proxy
ServiceIP
(iptables)
Node
Backend Pod 1
labels: app=MyApp
port: 9376
Backend Pod 2
labels: app=MyApp
port: 9376
Backend Pod 3
labels: app=MyApp
port: 9376

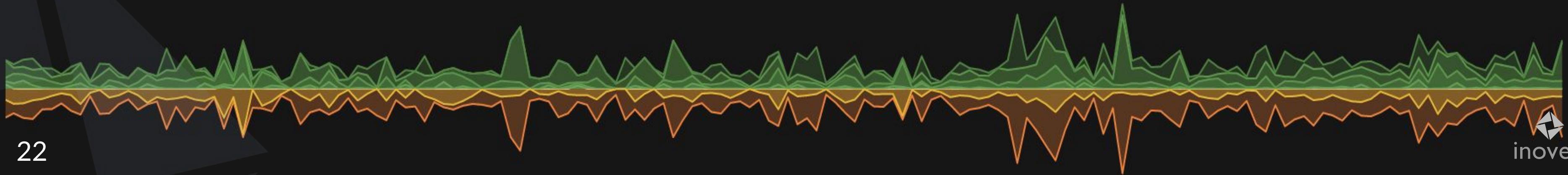
Namespaces

- 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

Hands on
Kubernetes concepts



Auto Scaling



inovex

What do we need to auto scale?

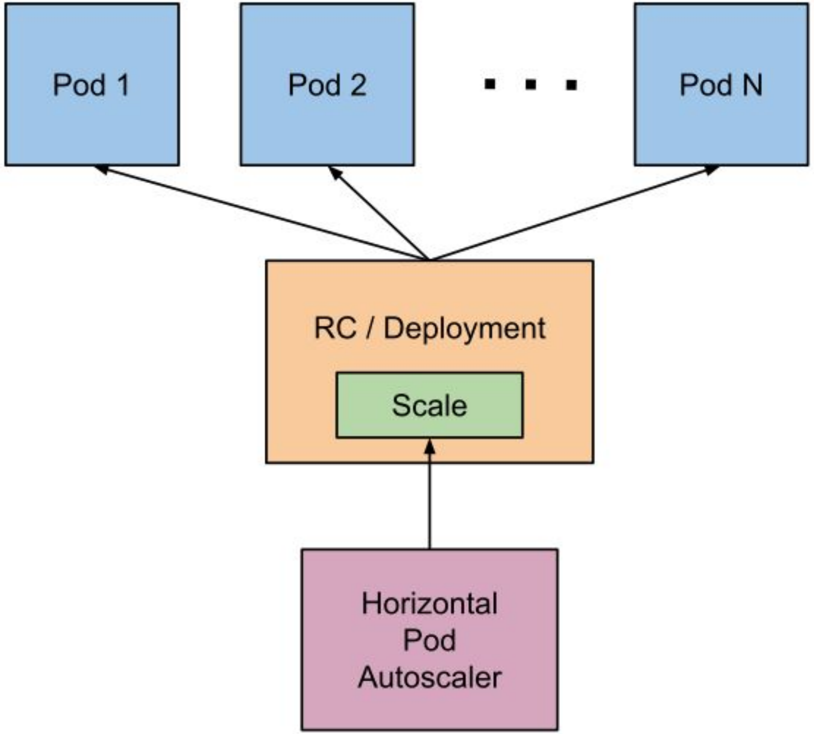
Metrics

- 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

Auto Scaling

- Deployments (horizontal)
  - Number of Pods based on CPU/Memory
  - Also custom metrics are supported
- Cluster (horizontal)
  - Number of nodes
  - Mostly “dump” algorithms are used
- Pods
  - Vertical Pod Autoscaler (change resource requests/limits)

Horizontal Pod Autoscaler



Hands on Autoscaling

Distributed Systems
What could possibly go wrong?

Outlook

- Volumes and Configuration
- Ingress
- Monitoring (Prometheus)

Questions?

Vielen Dank

Johannes M. Scheuermann

inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe

jscheuermann@inovex.de
0173 3181 058

Reading list

- https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale
- https://github.com/kubernetes/community/blob/master/contributors/design-proposals/autoscaling/horizontal-pod-autoscaler.md#autoscaling-algorithm