sig-autoscaling

Guy Templeton (co-lead)
Joseph Burnett (contributor)

Agenda

1. introduction
2. upcoming features
3. best practices
4. questions

Introduction

Horizontal Pod Autoscaler (HPA)
Scale in / out

Vertical Pod Autoscaler (VPA)
Scale up / down

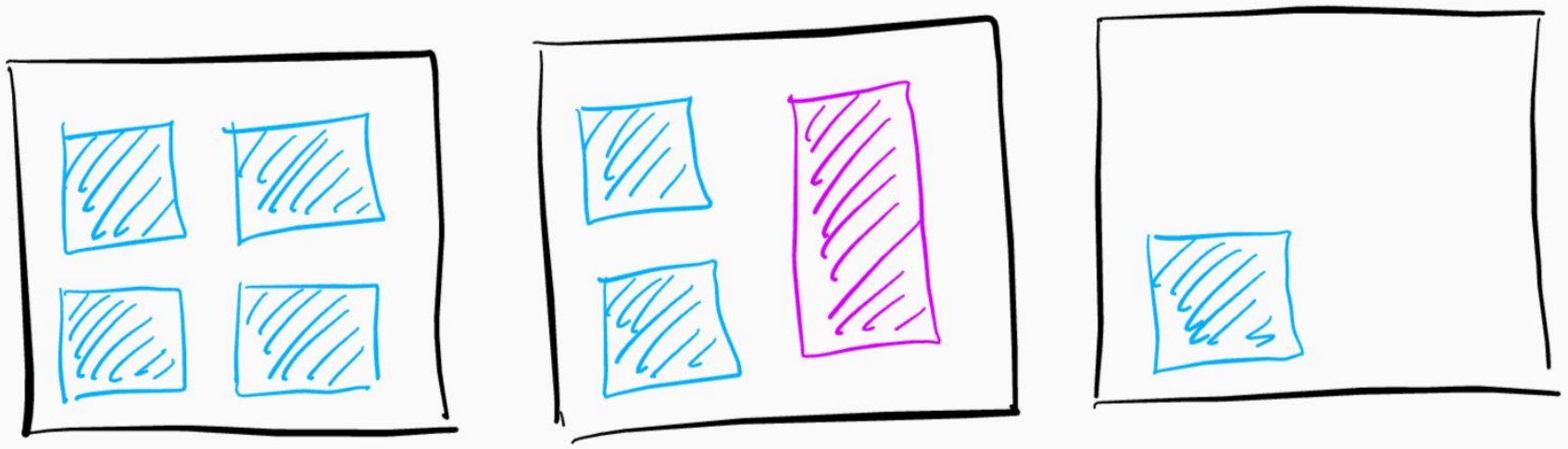
Workload (Pods)

Cluster (Nodes)

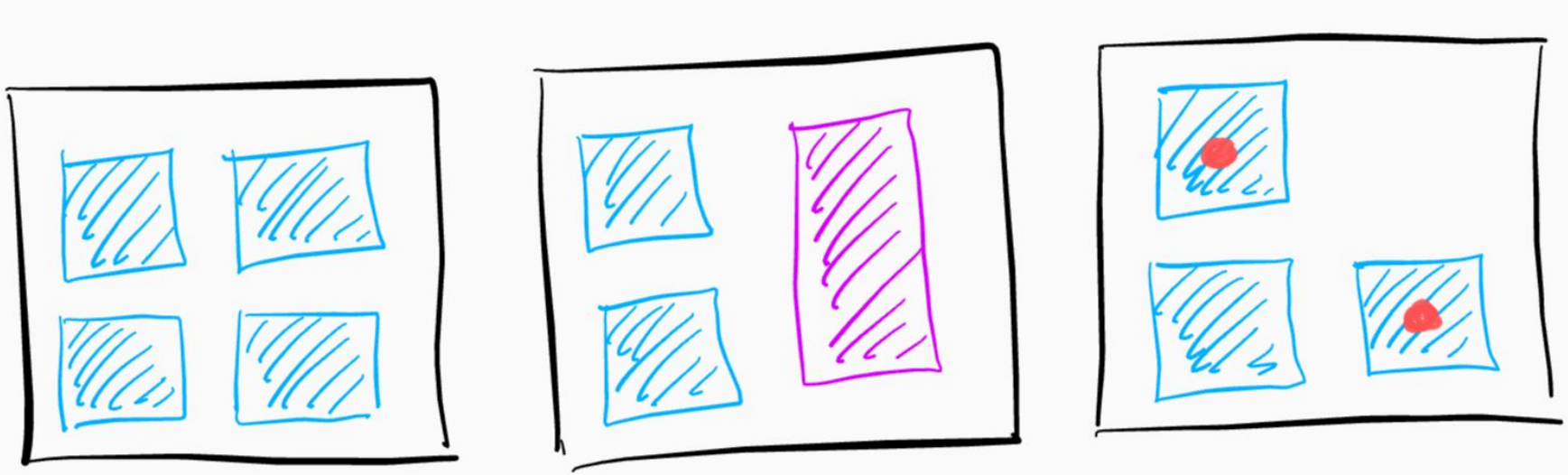
Cluster Autoscaler (CA)
Scale in / out

CA selects node pool
Scale up / down

Example

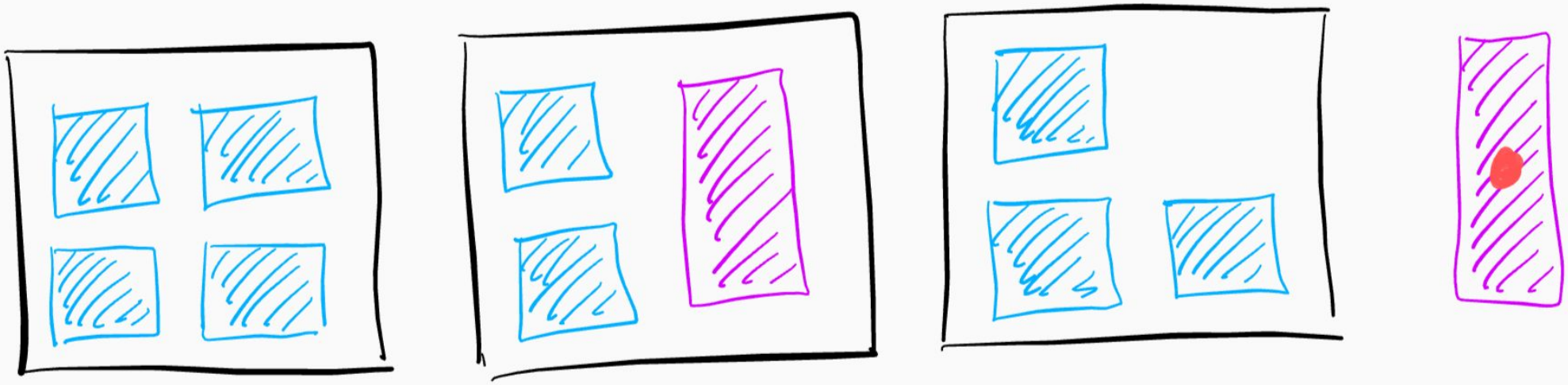


Example



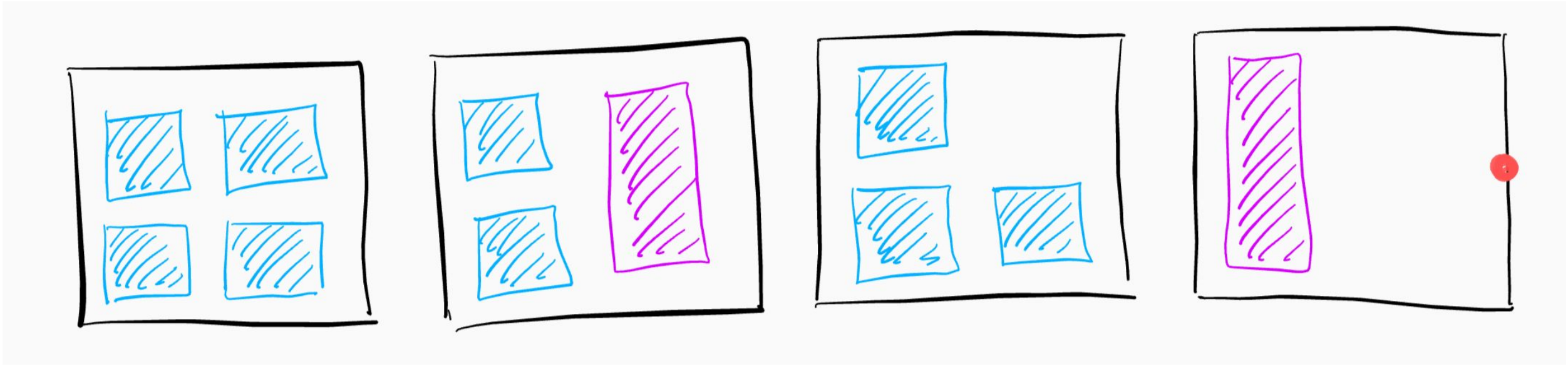
HPA scales out blue.

Example



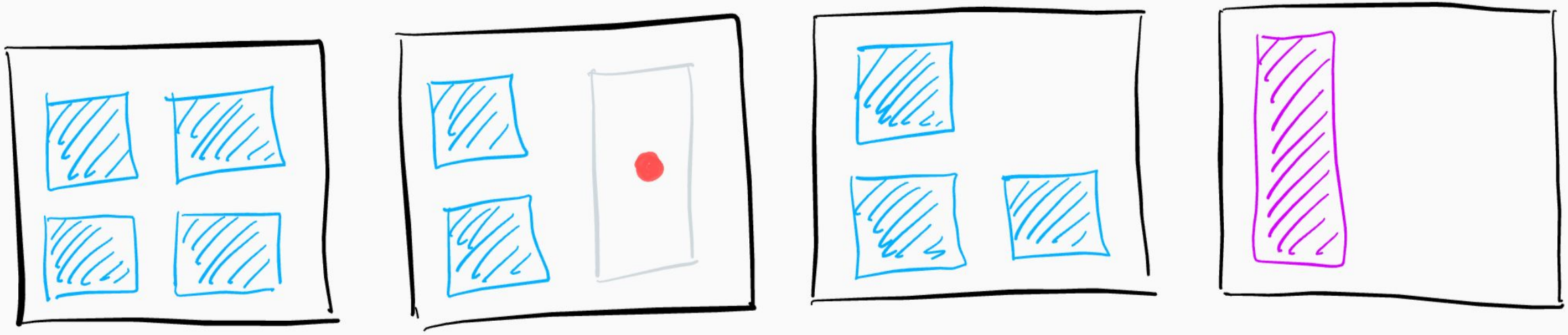
HPA scales out purple (unschedulable).

Example



CA scales out node pool. Purple is scheduled.

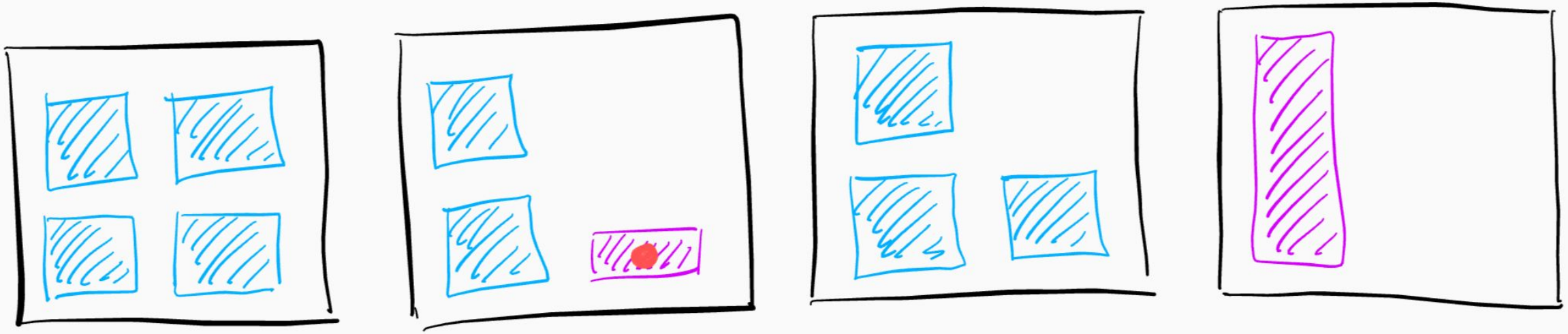
Example



VPA recommender scales down purple.

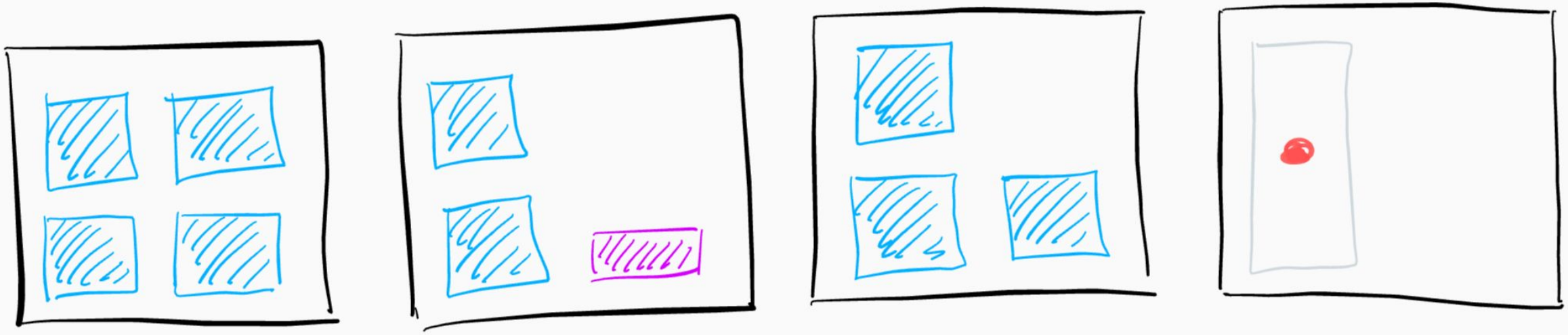
VPA updater deletes a pod.

Example



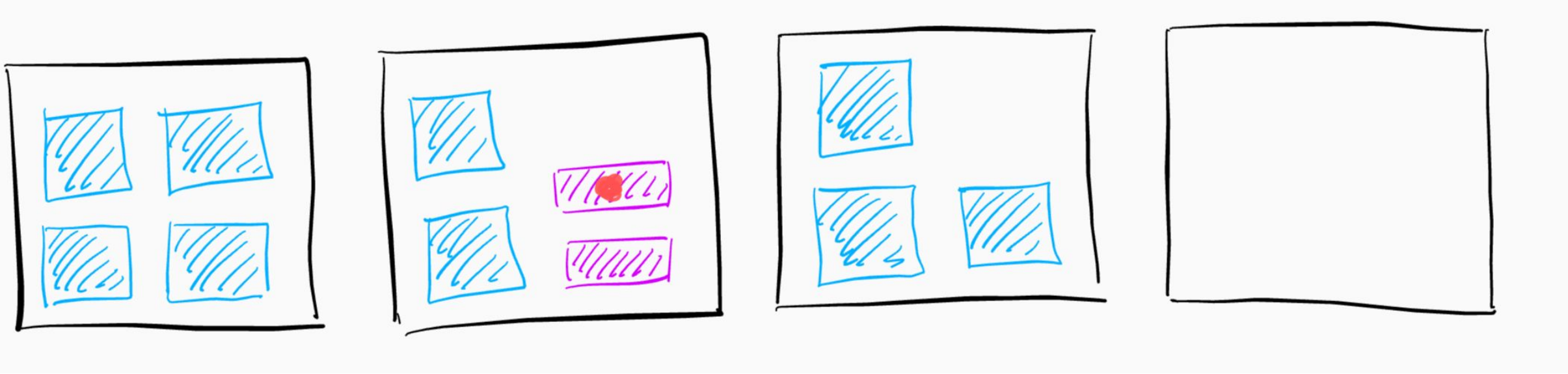
Purple pod is rescheduled with updated size.

Example



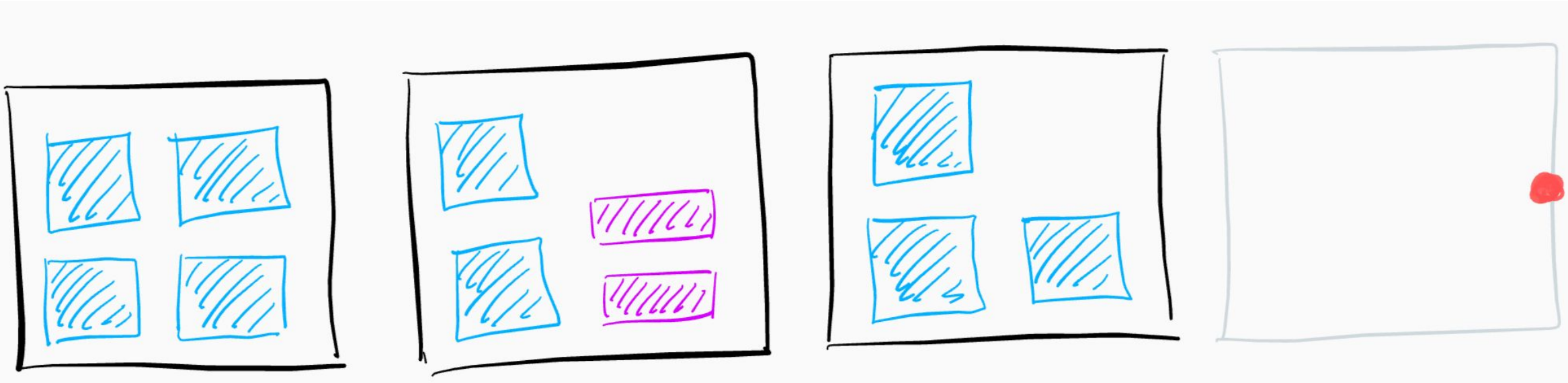
VPA updater deletes another purple pod.

Example



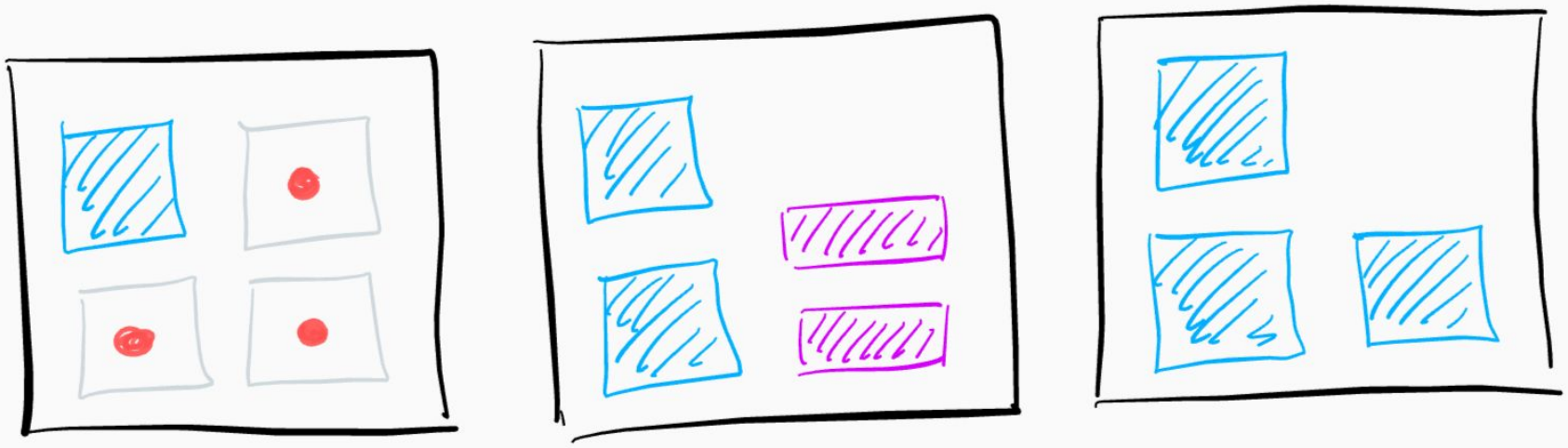
Purple pod is rescheduled with updated size.

Example



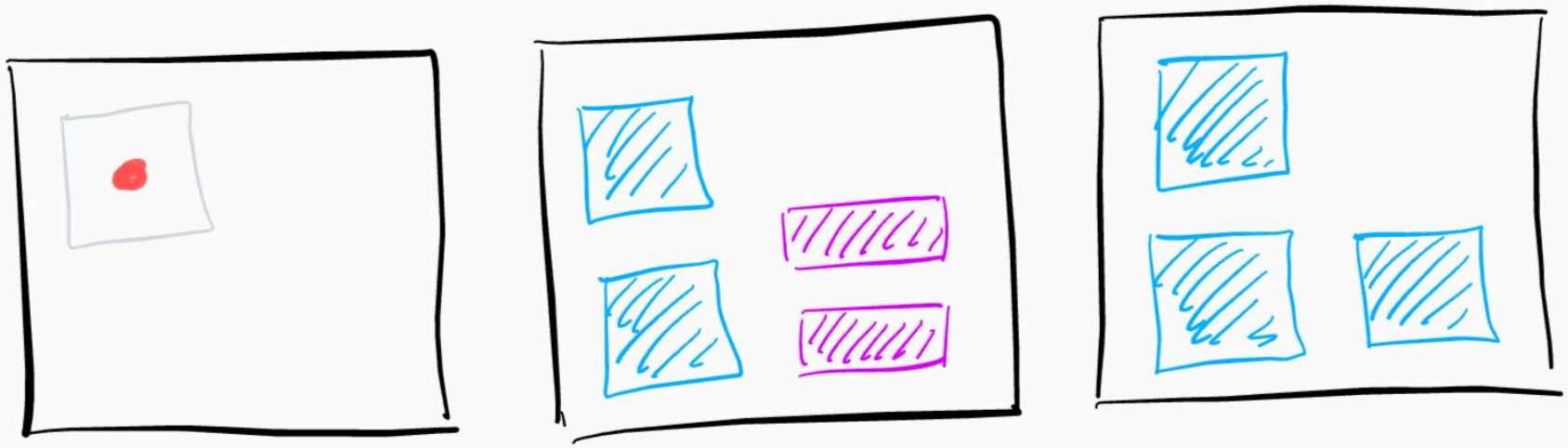
CA scales in empty node.

Example



HPA scales in blue.

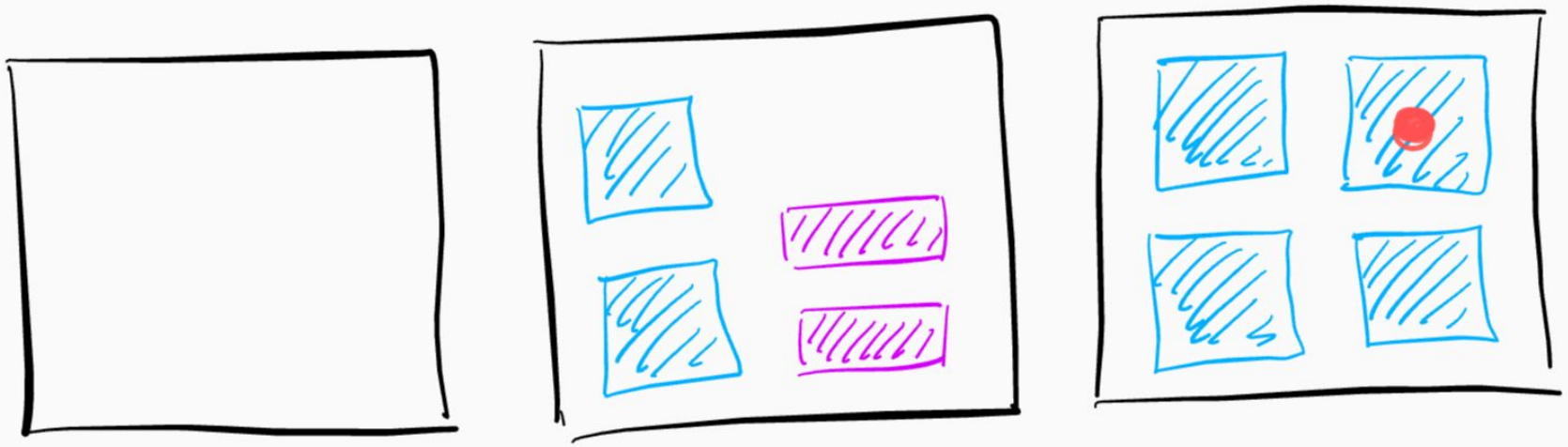
Example



CA consolidates blue.

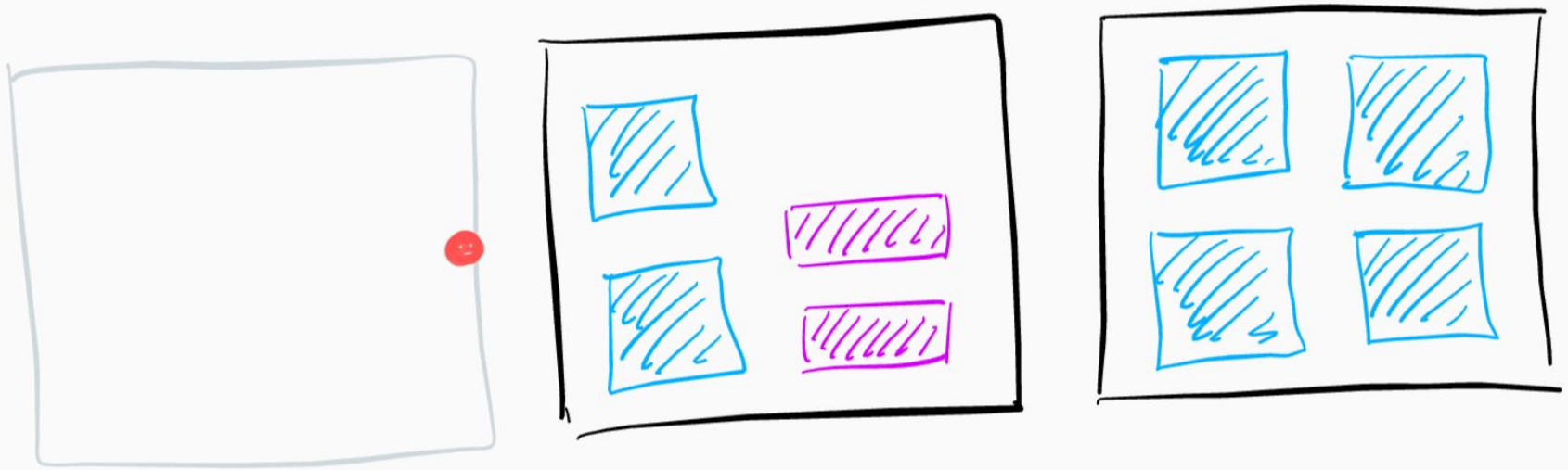
CA taints the node and evicts its pods.

Example



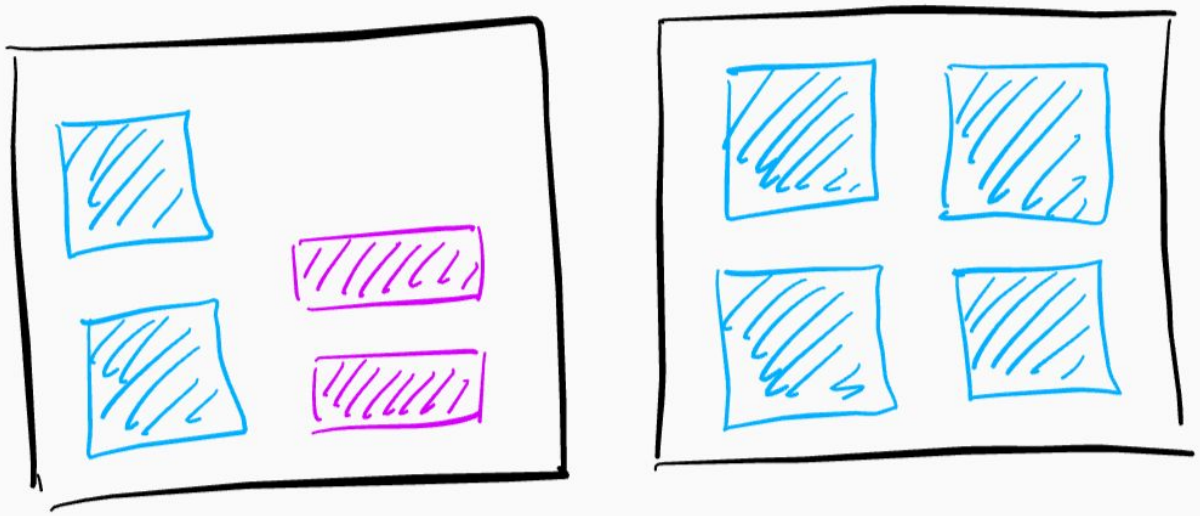
Blue pod is rescheduled.

Example



CA scales in node pool.

Example



Upcoming Features

1. hpa container resource targets [kep]
2. hpa v2 graduation to stable

Best Practices

- One of the key things is to enable developers to allow their services to scale gracefully and able to handle scaling on the cluster side

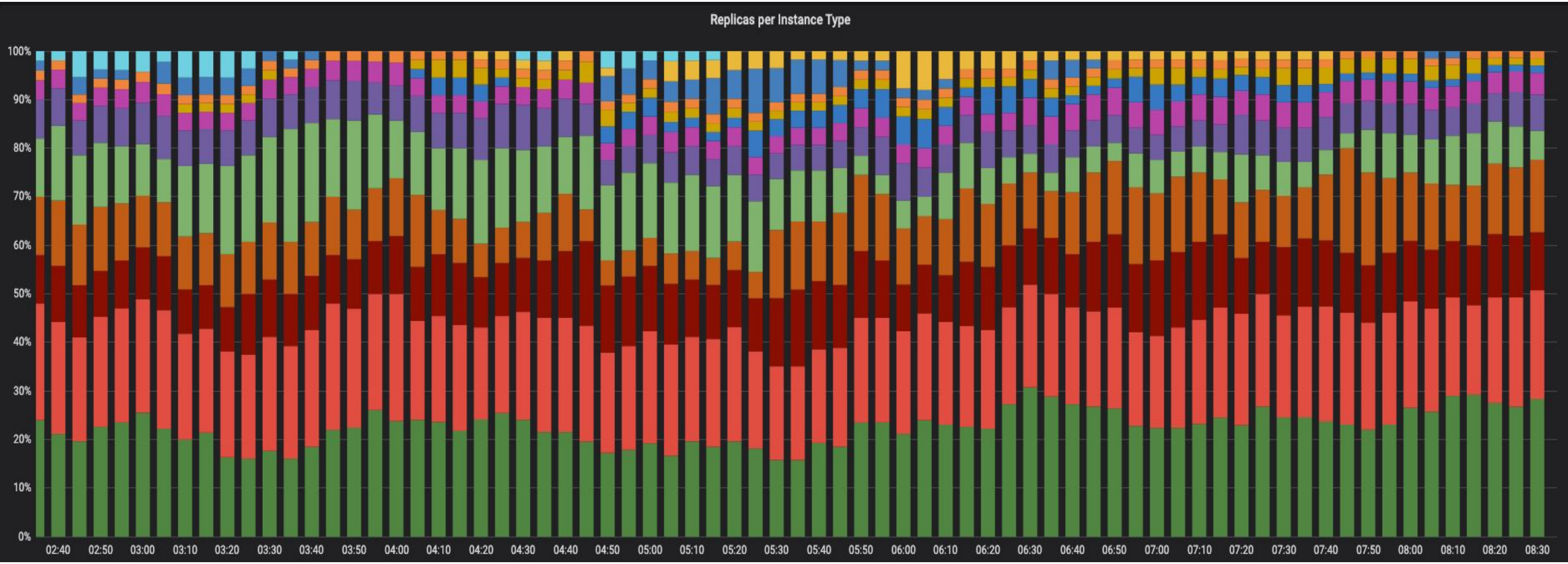
- Common options/settings to educate yourself/your developers about and provide good defaults for your environment:

  - Lifecycle hooks
  - Liveness and Readiness Probes
  - Pod annotations - cluster-autoscaler.kubernetes.io/safe-to-evict
  - Pod Disruption Budgets (PDBs)
  - Pod Priorities

Best Practices

- Horizontal Pod Autoscaling
  - Namespace level resource quotas
  - Metrics Sources for non-Resource Metric based autoscaling
  - Pod Affinities – these can either be node or pod based
  - Beware of the Pod level Resource Metric behaviour
  - Pod TopologySpreadConstraints – in beta as of 1.18

Best Practices



Best Practices

- Vertical Pod Autoscaling
  - Memory based vertical autoscaling isn't suitable for every language with default settings (I'm looking at you JVM based languages)
  - Combining Horizontal and Vertical pod autoscaling on the same metrics? - bad idea
  - Ensure ResourcePolicies are set (where appropriate)

Best Practices

- Cluster Autoscaling
  - Prioritise node startup time
  - Simulation inside the Cluster Autoscaler doesn’t always match reality
  - Ensure system and kubelet have enough resources on nodes

Resources

KubeCon CloudNativeCon North America 2020 Virtual

- Previous sig-autoscaling talks
  - Kubecon EU 2020 -- sig-autoscaling Deep Dive

- Best practice reading
  - Autoscaling Monzo
  - GKE's Best Scaling Practices


KEEP CLOUD NATIVE
EVERYWHERE
KubeCon
CloudNativeCon
North America 2020
Virtual