caicloud
才云

Stability and Availability in Kubernetes

杨朝乐 才云科技

Agenda

- Understand Stability
- Meet Abnormal
- High-Availability in Kubernetes
- Deal with Abnormal
- Experience

Understand Stability

Apps Apps Apps

Deployment StatefulSet

Cluster

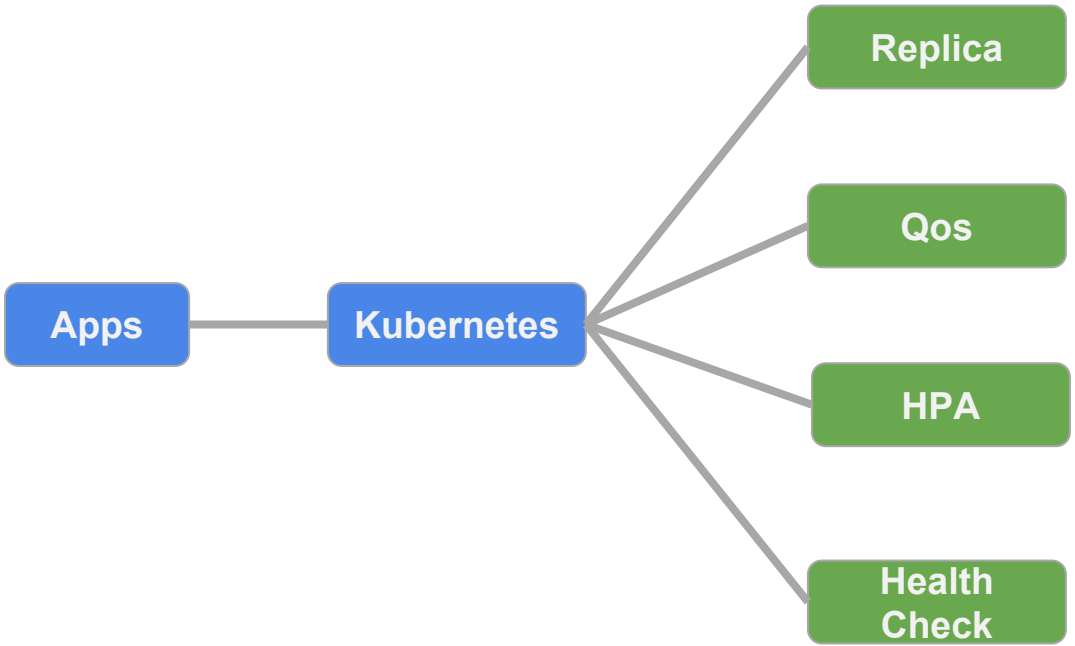
Node Node

Network Storage Resource Component

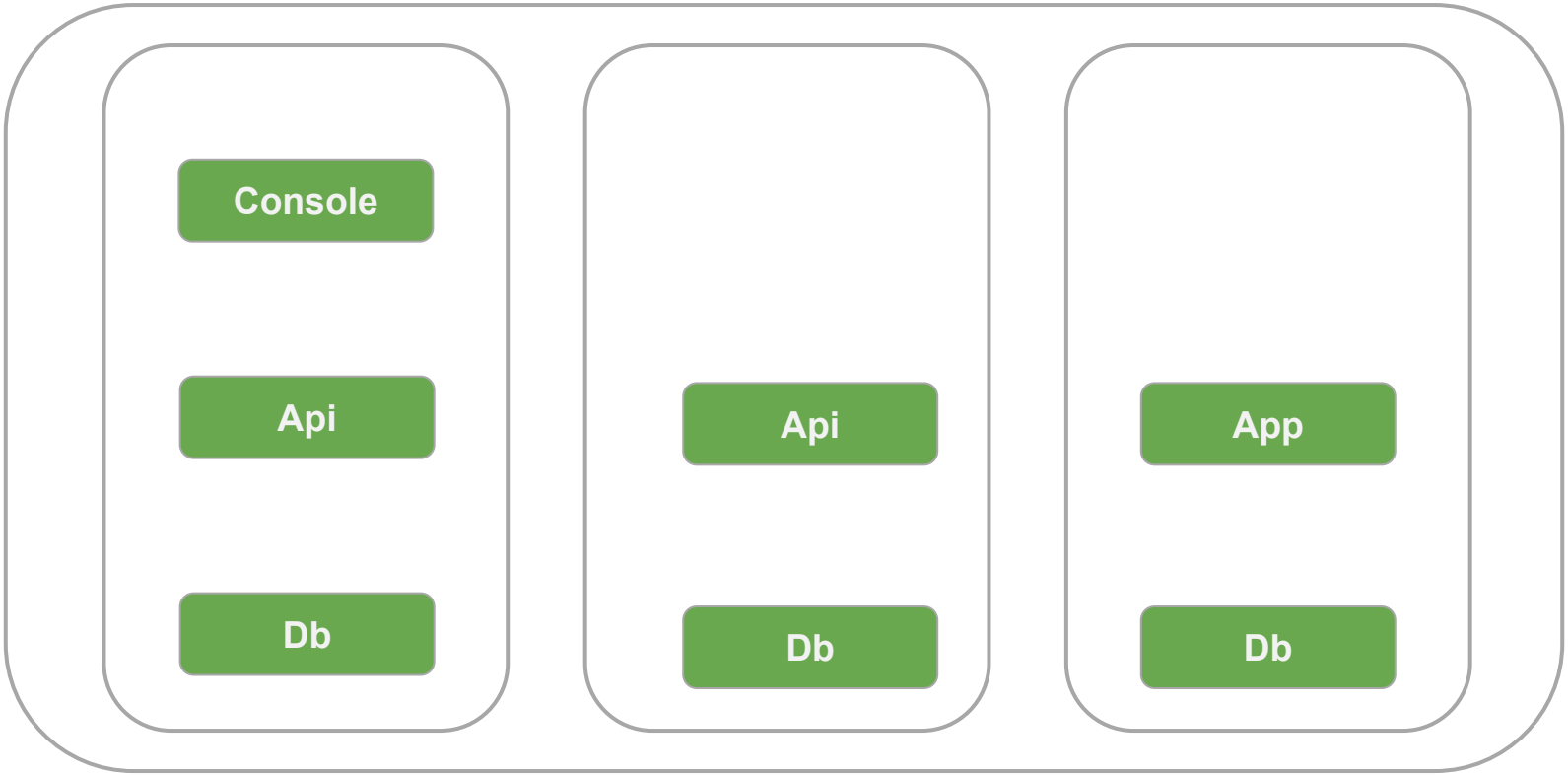
Hardware

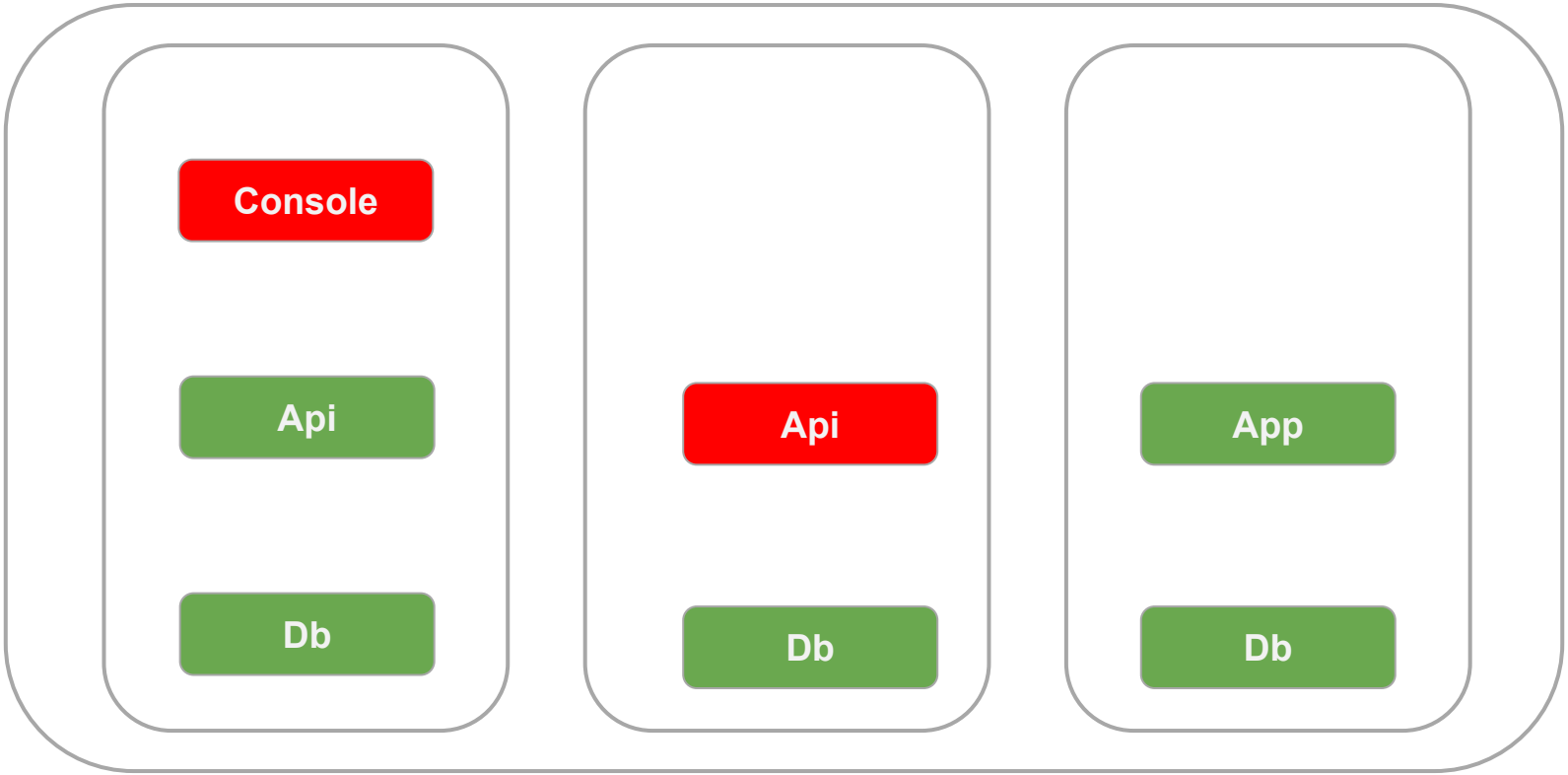
Credit: A. Aleksandravicius / Moment / Getty Images

Kubernetes offers



Everything works fine if everything works fine



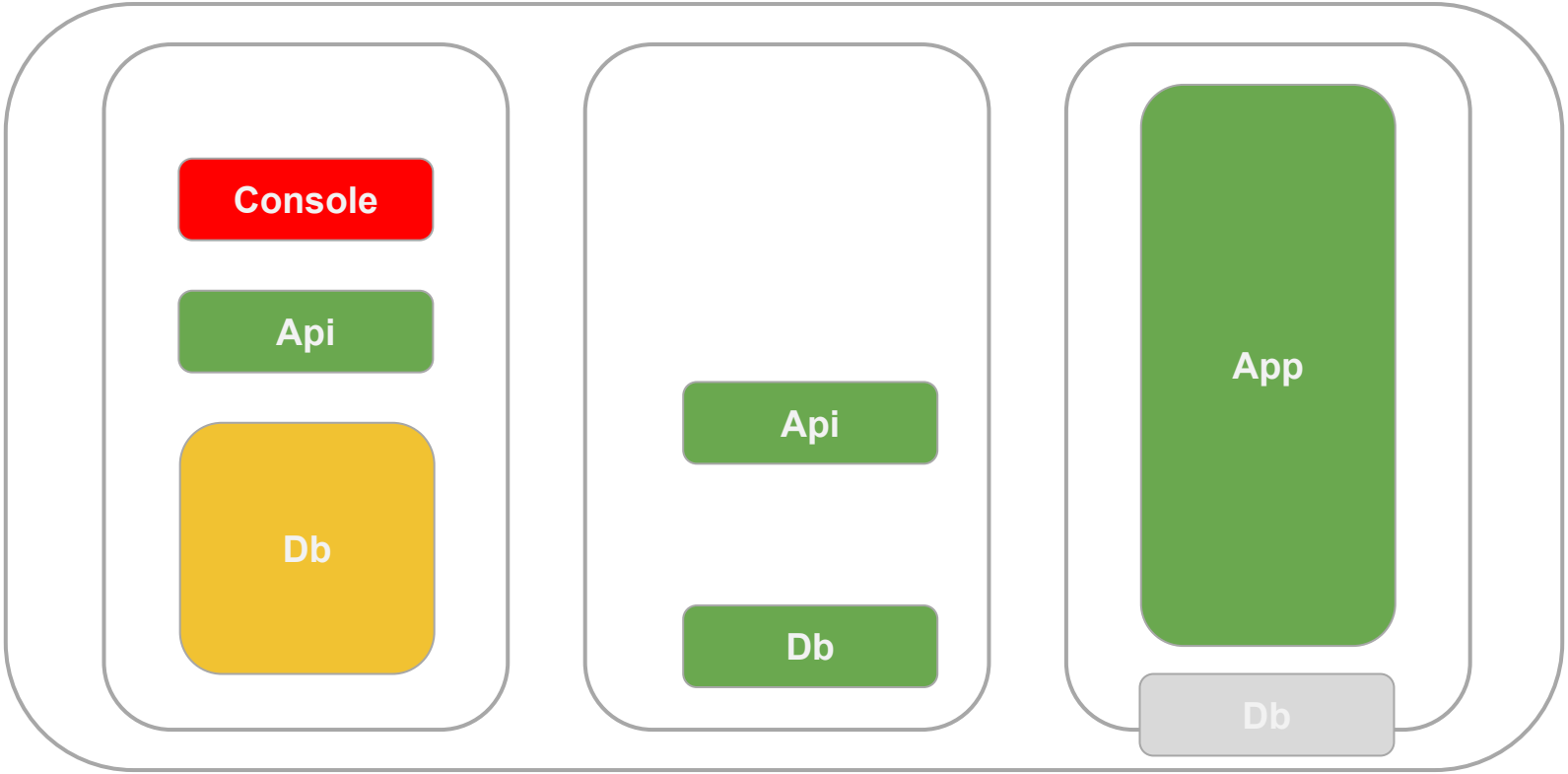


Case : Console crashed

- Service is down during console is down
- Console will run again to match **replica:1**
- Service is up when Console is up

Case : One Api is down(e.g. Deleted by accident)

- Service will go on since there still has one Api
- Api will be recreated to match **replica:2**



Case : Db take too many memory

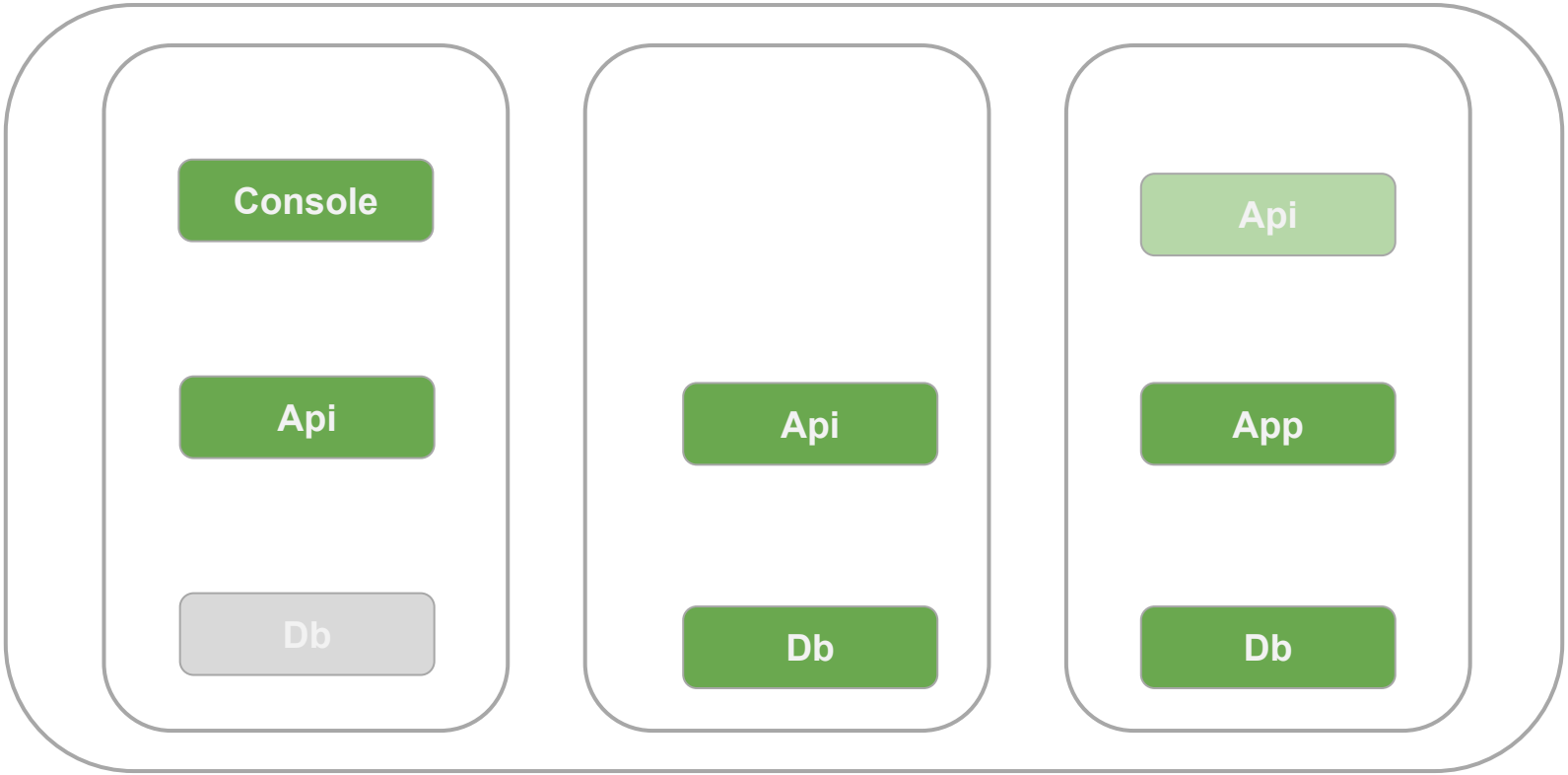
- Db itself may be OOM killed if all Qos is **BestEffort**
- Db itself may be OOM killed if resource **limit is too low**
- Console may be OOM killed if Db is **Guaranteed**

Case : App request too many memory

- Other pod can’t run one this node

QoS classes:

- Guaranteed
- Burstable
- BestEffort



Case : heavy load on API

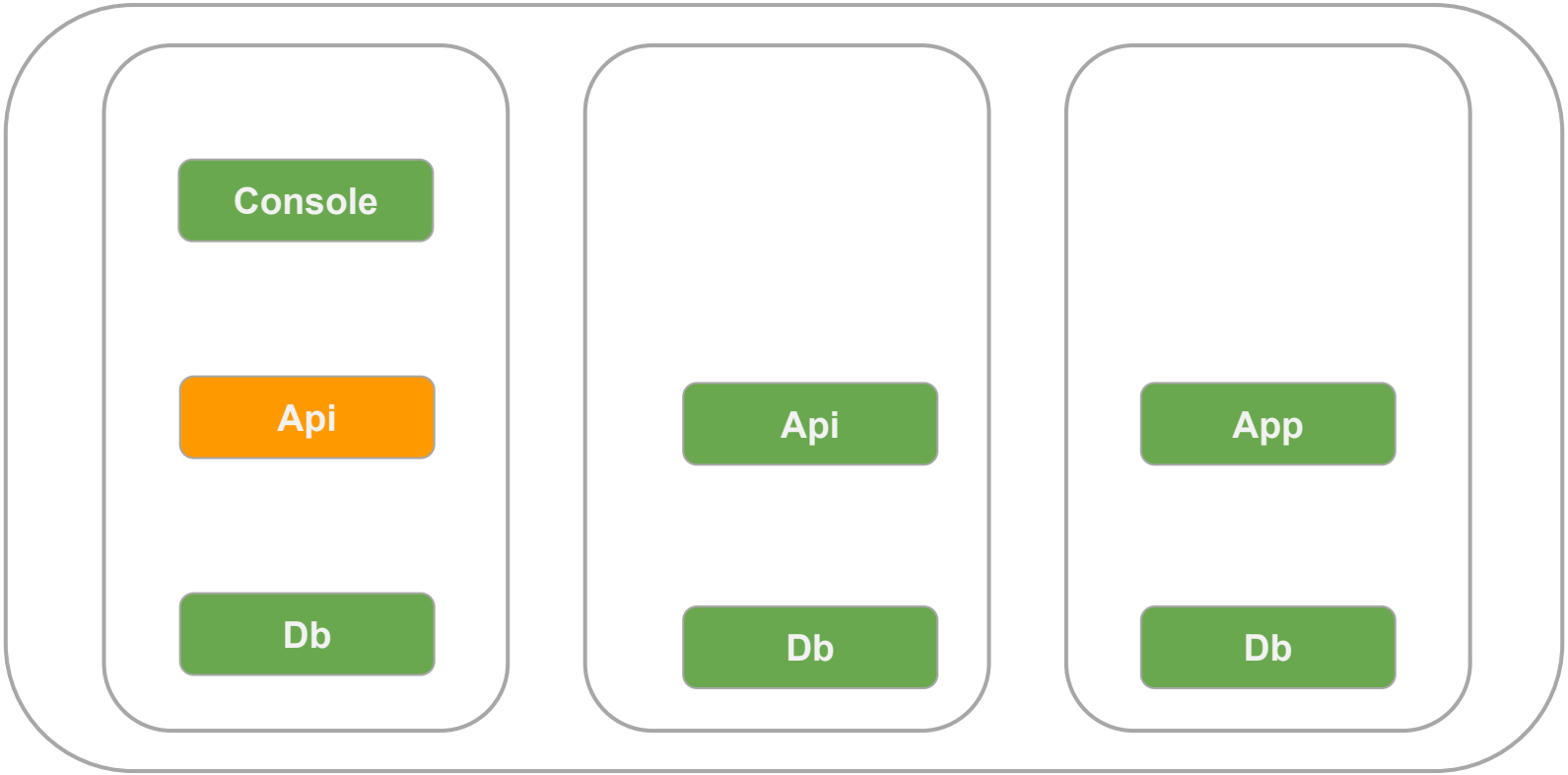
- Api will be scaled up (replica 2 -> 3)
- Lower load on Api

Case : low load on Db

- Db will be scaled down (replica 3 -> 2)
- Release more resource

More:

- Custom metric
- Cluster Autoscaler
- VPA



Case : Api is running but hang by deadlock

- Api will not work fine
- Api will be restarted if **liveness probes** is set

Case : Api is running and initializing

- Api will not work fine if request come during initializing
- No request will come if **readiness probes** is set

More:

- Configure Liveness and Readiness Probes

- Reserve Compute Resources for System Daemons
- Guaranteed Scheduling For Critical Add-On Pods

Allocatable : the amount of compute resources that are available for pods

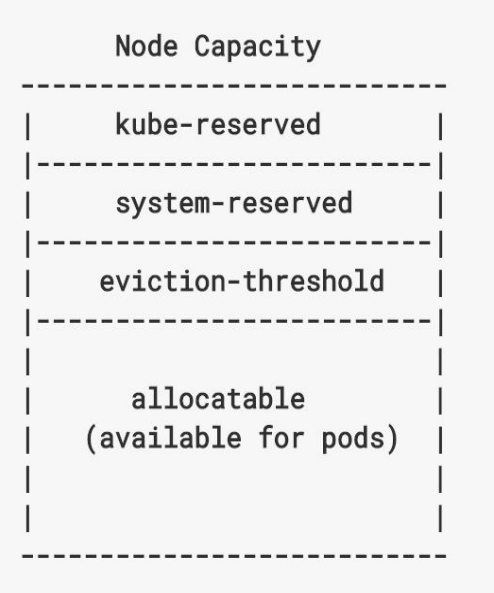
- CPU, memory and ephemeral-storage are supported
- The scheduler does not over-subscribe Allocatable

Kube-reserved:

- meant to capture resource reservation for kubernetes system daemons like the kubelet, container runtime, node problem detector

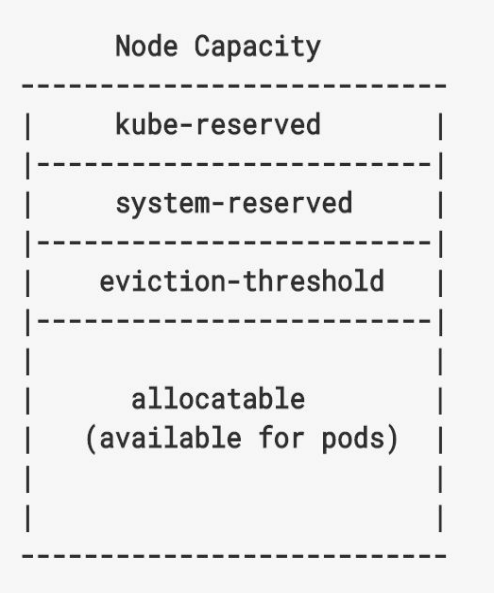
System-reserved:

- meant to capture resource reservation for OS system daemons like sshd, udev



Enabled by:

- Kubelet flag **--cgroups-per-qos**
- Kubelet flag **--system-reserved**
- Kubelet flag **--kube-reserved**
- Kubelet flag **--eviction-hard**



Guaranteed Scheduling For Critical Add-On Pods

Critical Add-On : Pods must run on a regular cluster node, e.g.

- Heapster, DNS
- Core components

Rescheduler:

- guaranteed scheduling of critical add-ons

Enabled by:

- Add Rescheduler static pod manifests
- **scheduler.alpha.kubernetes.io/critical-pod** annotation
- **[{"key":"CriticalAddonsOnly", "operator":"Exists"}]** tolerations

Meet abnormal

Apps Apps Apps

Deployment StatefulSet

Cluster

Node Node

Network Storage Resource Component

Hardware

Credit: A. Aleksandravicius / Moment / Getty Images

Meet abnormal

Case : Apps are not healthy

Case : Apps are not created

Case : Cluster is not healthy

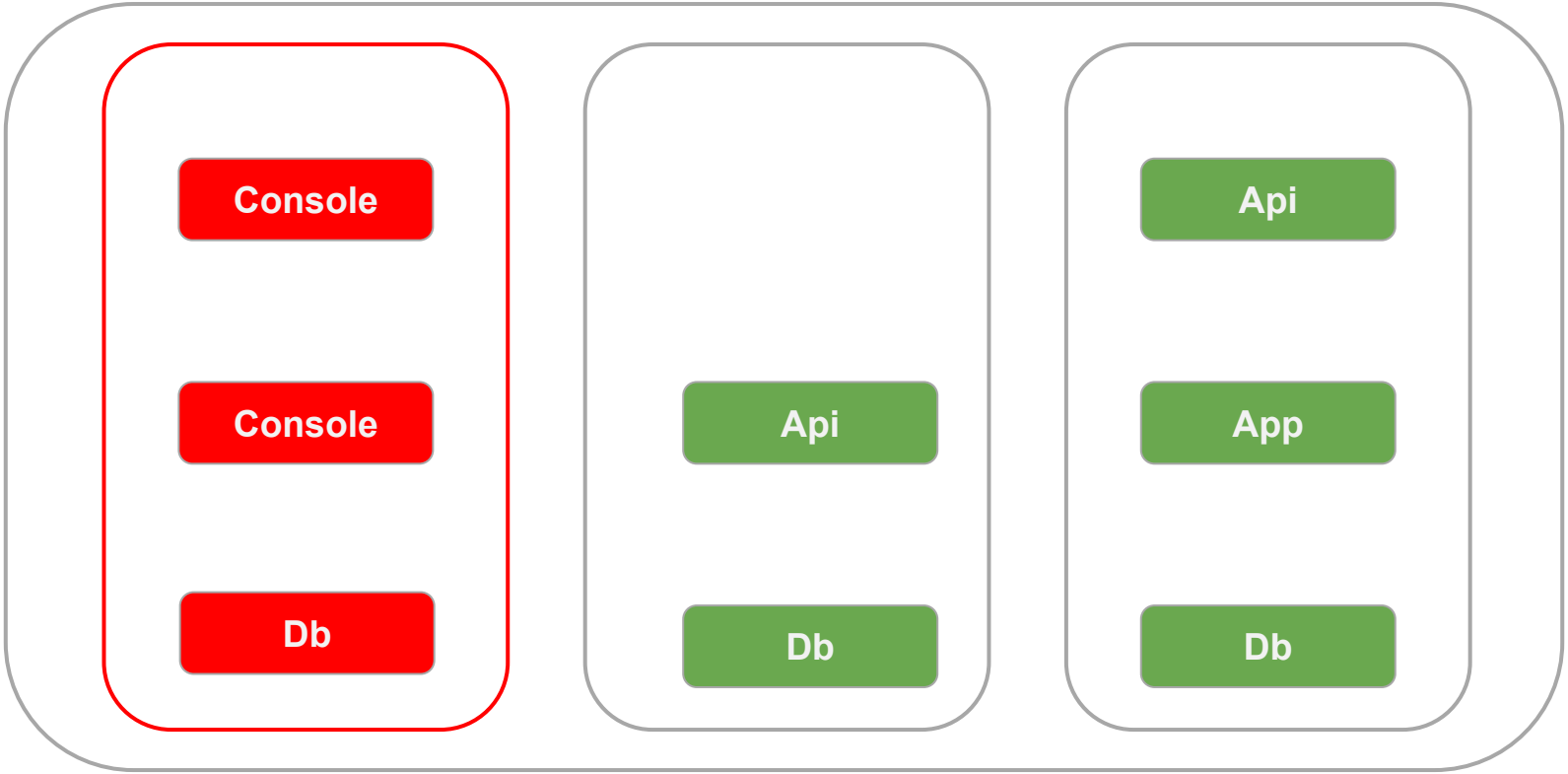
Case : Node is down

Case : Network is broken

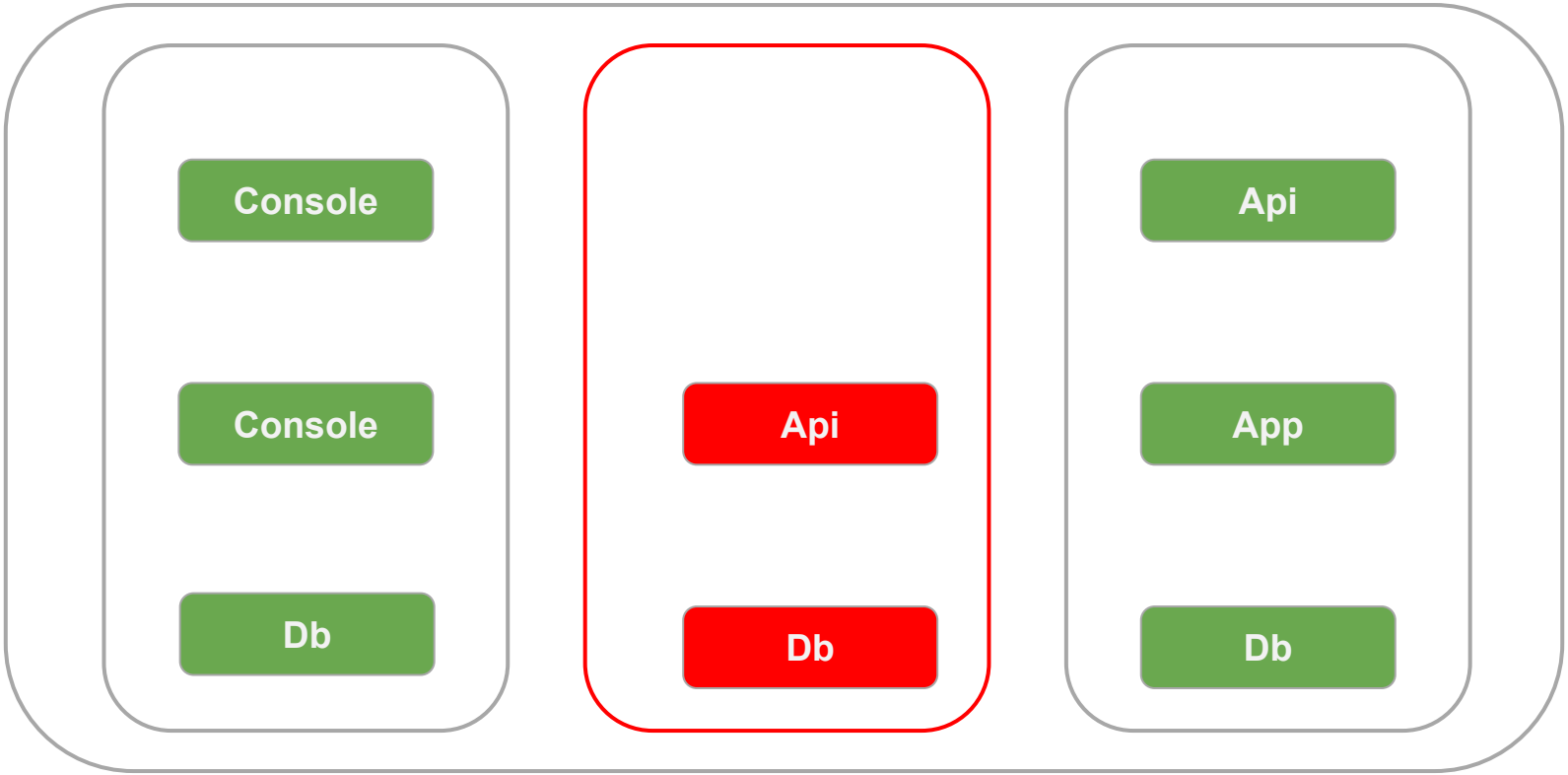
Case : Docker hang

Case : Hardware issues

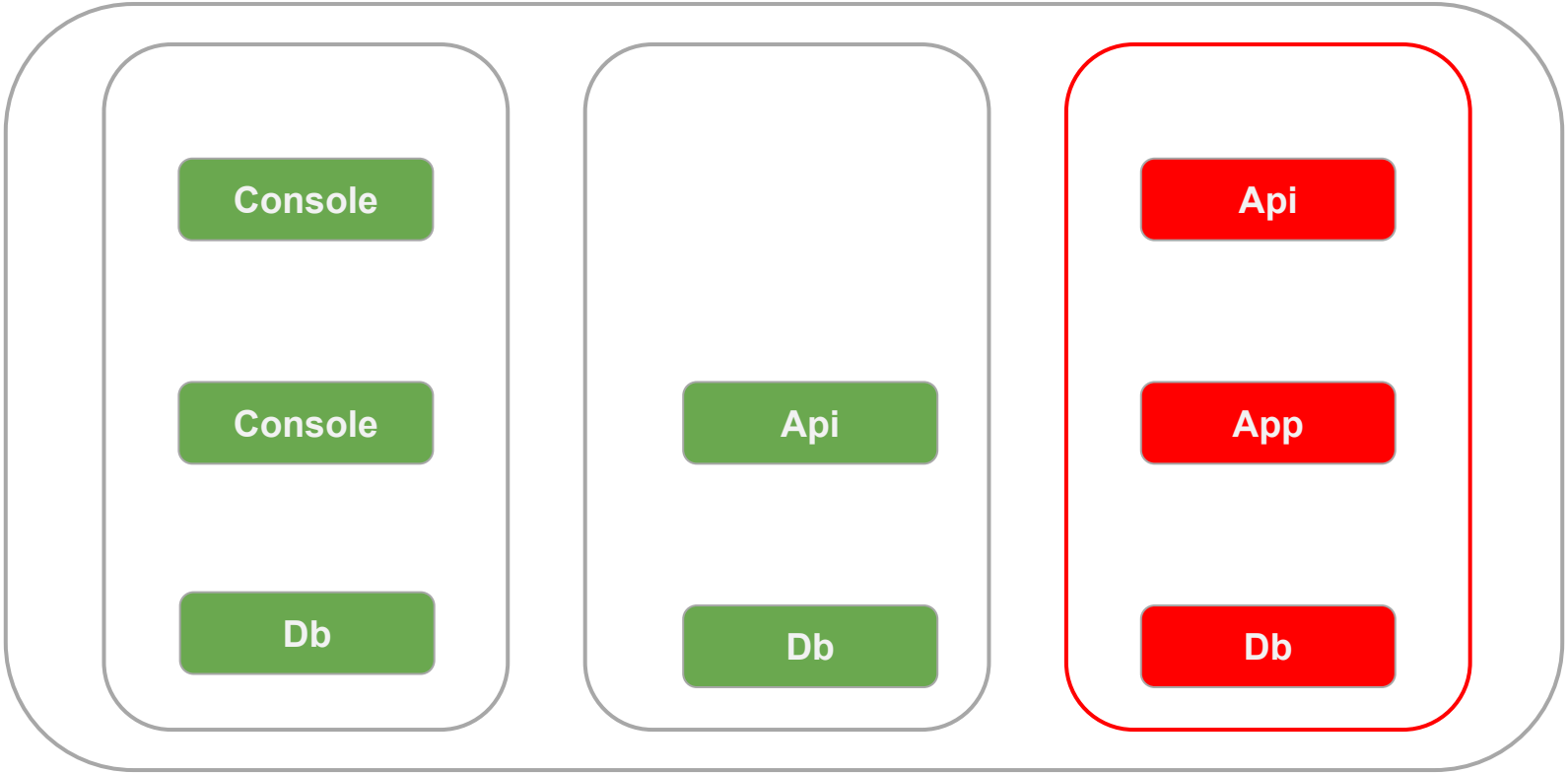
Case: Power issue



Case: Power issue



Case: Power issue



Case : Power issue

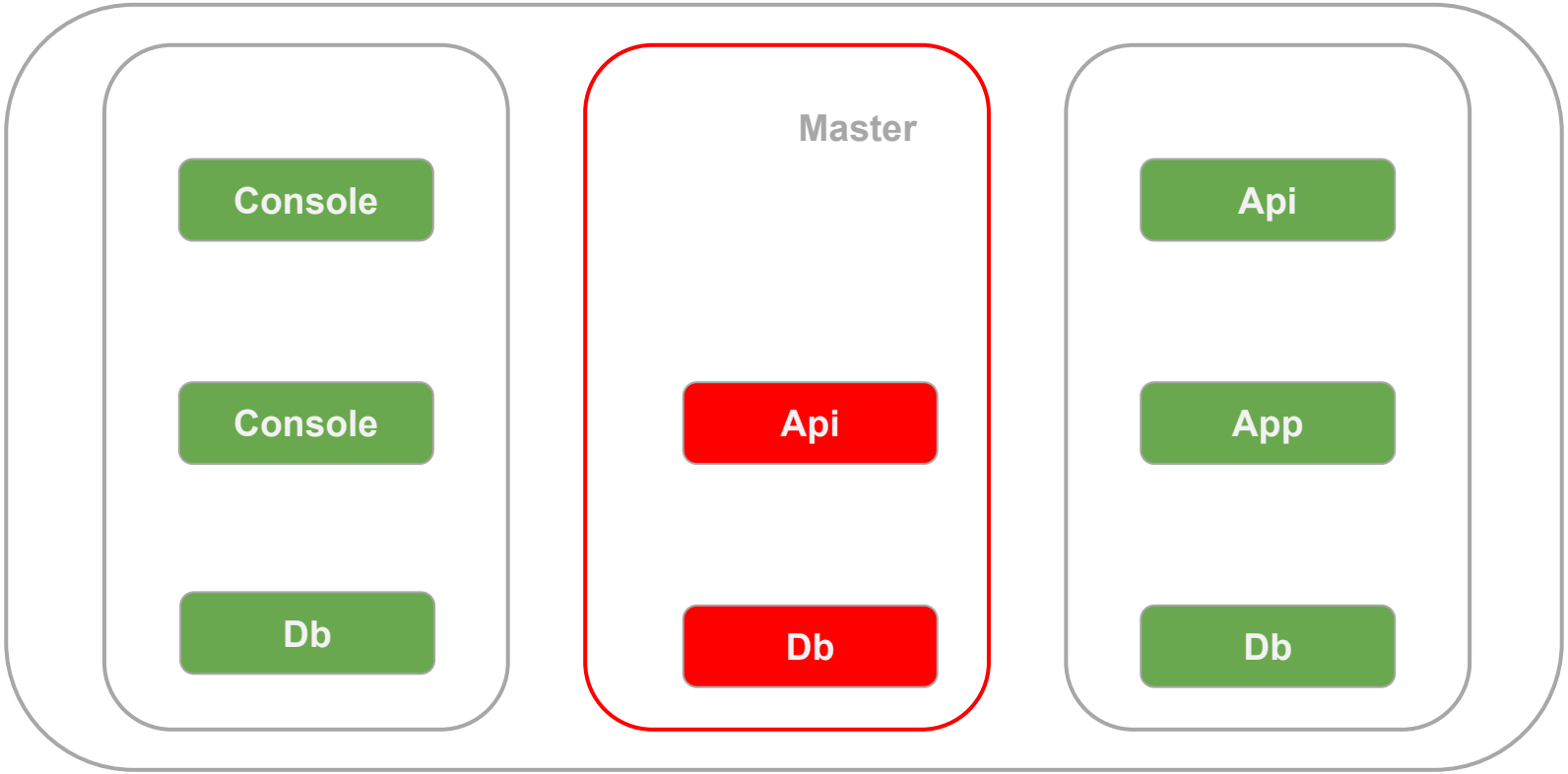
Downtime:

- Long if Replica = 1
- pod-eviction-timeout

Tips:

- Replica > 1
- podAntiAffinity
- Reschedule need time

Case: Power issue



Case : Power issue on master

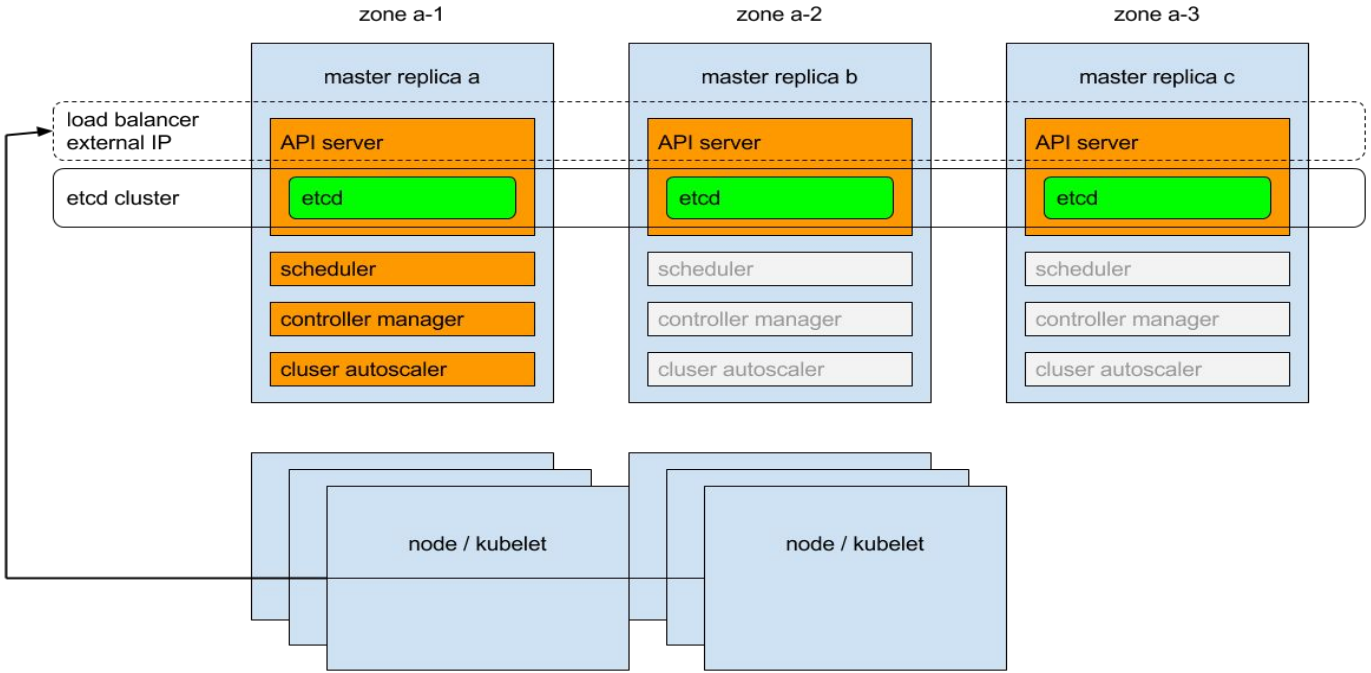
Single master:

- No reschedule
- Half service down even Replica > 1

Multi master:

- Just one node not ready
- Pod will be evicted and rescheduled

High-Availability in Kubernetes



Overview:

master:

- Etcd on each master as a cluster
- Apiserver talk to local etcd
- Leader election for controllers, scheduler, and cluster auto-scaler

load balancer:

- Maybe external provided by IaaS
- Maybe internal setup use keepalived and nginx

Node:

- Kubelet and proxy talk to load balancer VIP

- Kubeadm HA (high availability) checklist

The following is a checklist for kubeadm support for deploying HA-clusters. This a distillation of action items from:
https://docs.google.com/document/d/1lH9OKkFZMSqXCApmSXemEDuy9qlINdm5MfWWGrK3JYc/edit#heading=h.8hdxw3quu67g

but there may be more.

New Features:

- [x] Option enable active<>passive locking via ConfigMaps kubernetes/kubernetes#44857
  PR: kubernetes/kubernetes#45739
  - [x] Enable support to ComponentsConfigs to be loaded from ConfigMaps
    https://docs.google.com/document/d/1arP4T9Qkp2SovlJZ_y790sBeiWXDO6SG10pZ_UUU-Lc/edit?ts=59110d75#heading=h.xgjl2srtytjt
- [ ] Add support to standup etcd operator #254
  PR: kubernetes/kubernetes#45665
- [ ] Define exact initialization flow (remove explicit migration step)

Contentious:

- [ ] Handle the smart client vs. endpoints/proxy/apiserver discovery. (kubernetes/kubernetes#18174 & more)

Extending Support & Documentation:

- [ ] Update documentation to list supported best practices on LB configuration
  kubernetes/kubernetes.github.io#3607
- [ ] Allow input options to standup HA cluster using existing etcd bootstrap endpoint
  - [ ] kubernetes/kubernetes#44793
- [x] Make secrets more secure so we can pass secrets (see sig-auth) kubernetes/kubernetes#41939


jamiehannaford


luxas

Labels

area/HA
area/upgrades
kind/enhancement
priority/important-soon

Projects

None yet

Milestone

v1.10

11 participants

- Workarounds for the time before kubeadm HA becomes available

The planned HA features in kubeadm are not going to make it into v1.9 (see #261). So what can be done to make a cluster setup by kubeadm sufficiently HA?

This is what it looks like now:

- Worker nodes can be scaled up to achieve acceptable redundance.
- Without a working active/active or at least active/passive master setup, master failures are likely to cause significant disruptions.

Hence an active/active or active/passive master setup needs to be created (i.e. mimic what kubeadm would supposedly be doing in the futue):

1. Replace the local etcd pod by an etcd cluster of min. *2 x number-of-masters* size. This cluster could running on the OS rather than in K8s.
2. Set up more master instances. That's the interesting bit. The Kubernetes guide for building HA clusters (https://kubernetes.io/docs/admin/high-availability/) can be of help to understand what needs to be done. Here I'd like to have simple step-by-step instructions taking into consideration kubeadm-setup particularities in the end.
3. Not sure whether this is necessary: Probably set up haproxy/keepalived on the master hosts, move the original master's IP address plus SSL termination to it.

This seems achievable if converting the existing master instance to a cluster of masters (2) can be done (the Kubernetes guide for building HA clusters seems to indicate so). Active/active would be not more expensive than active/passive.

I am currently working on this. If I succeed I shall share what I find out here.


mbert


jamiehannaford

Labels

area/HA
documentation/content-gap
documentation/improvement
kind/enhancement
priority/important-soon

Projects

None yet

Milestone

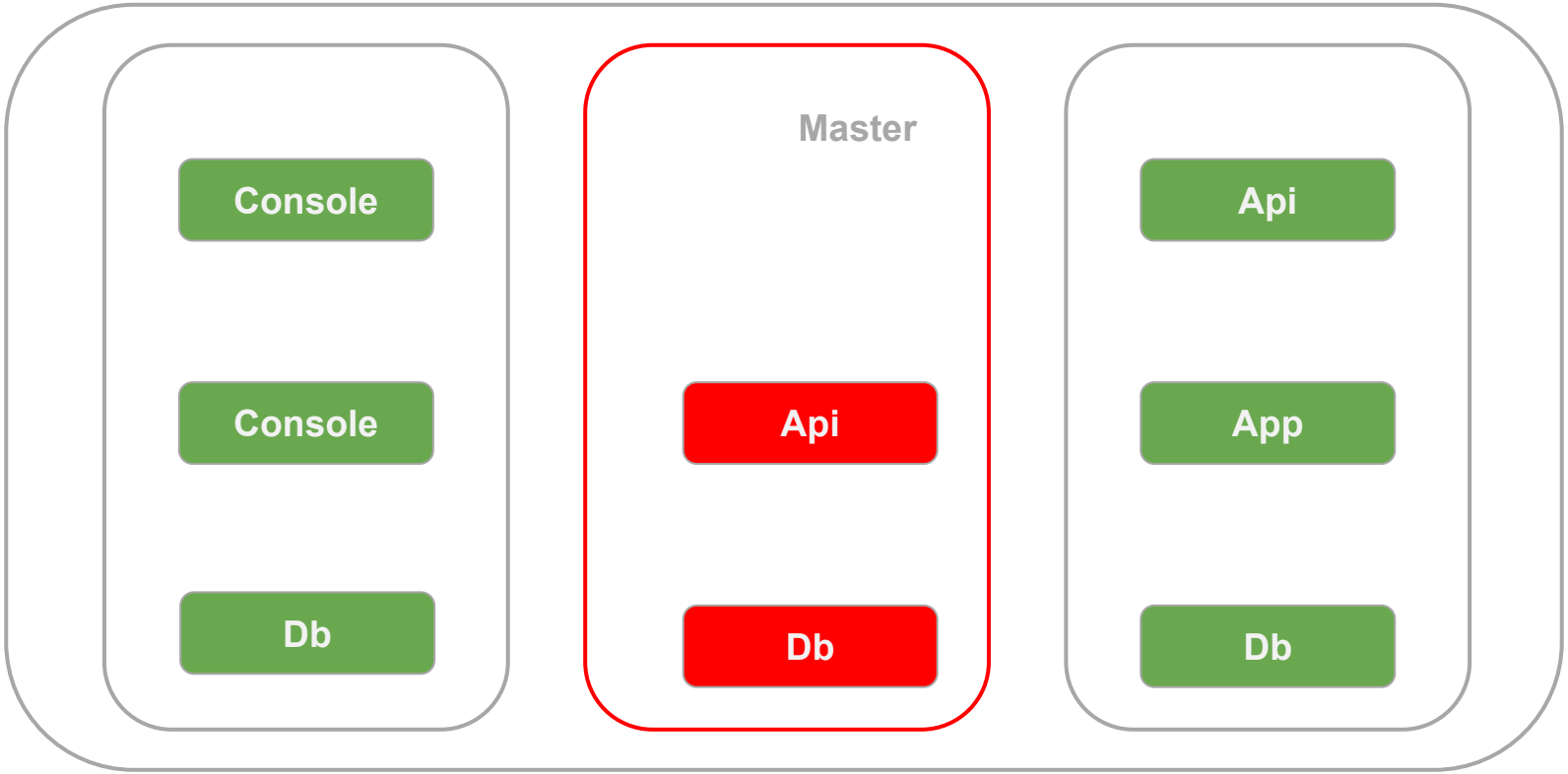
v1.10

12 participants

Links

- Set up High-Availability Kubernetes Masters
- Building High-Availability Clusters
- Proposal for a highly available control plane configuration for ‘kubeadm’ deployments
- Adding HA to kubeadm-deployed clusters
- Proposal for kubeadm self-hosted HA deployment

Back to Power issue on HA Master



Overview:

Etcd client:

- Apiserver may still connect bad etcd endpoint
- Different result when use kubectl on different node
- apiserver etcd v3 client does not close connections to dead etcd node

default/kubernetes svc:

- Bad apiserver endpoint may not be deleted
- session affinity enabled on default/kubernetes svc
- Fixed by add lease endpoint reconciler , not enabled by default

Workaround:

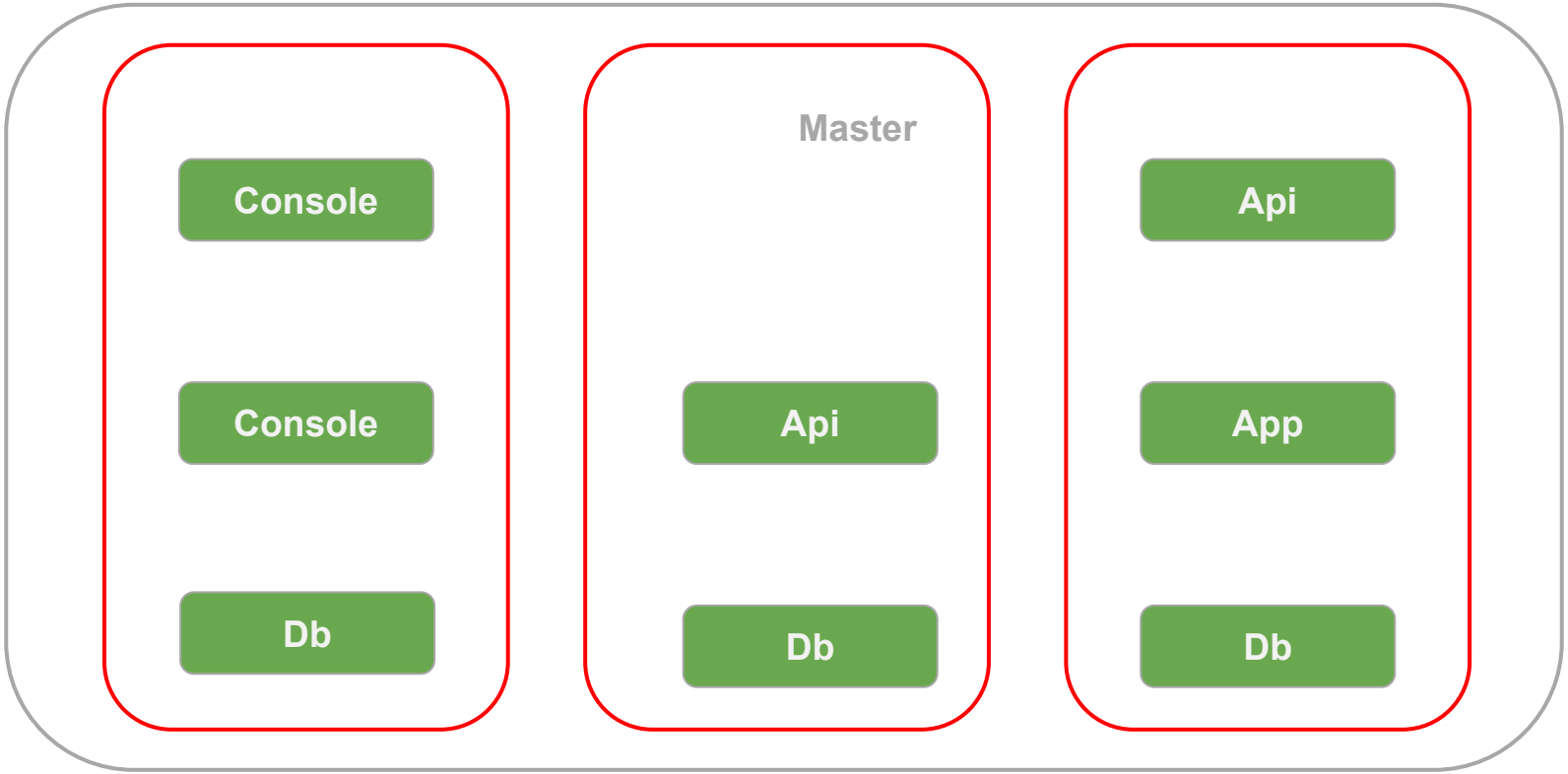
Etcd client:

- A daemon to restart etcd server if needed
- Cherry pick upstream code

default/kubernetes svc:

- A controller to remove bad apiserver endpoints
- Disable session affinity

Still with Power issue on HA Master



Still with Power issue on HA Master

Case:

All nodes/masters restart at the same time:

- Master get ready first, but LB still down
- Node become not ready
- All pods on node will be rescheduled on masters
- Heavy load on masters

All nodes/masters restart at the same time, but LB is fine:

- Heavy load on apiserver and etcd
- Load may never be normal when low io on etcd

Still with Power issue on HA Master

Tips:

All nodes/masters restart at the same time:

- Make master unschedulable or add taints

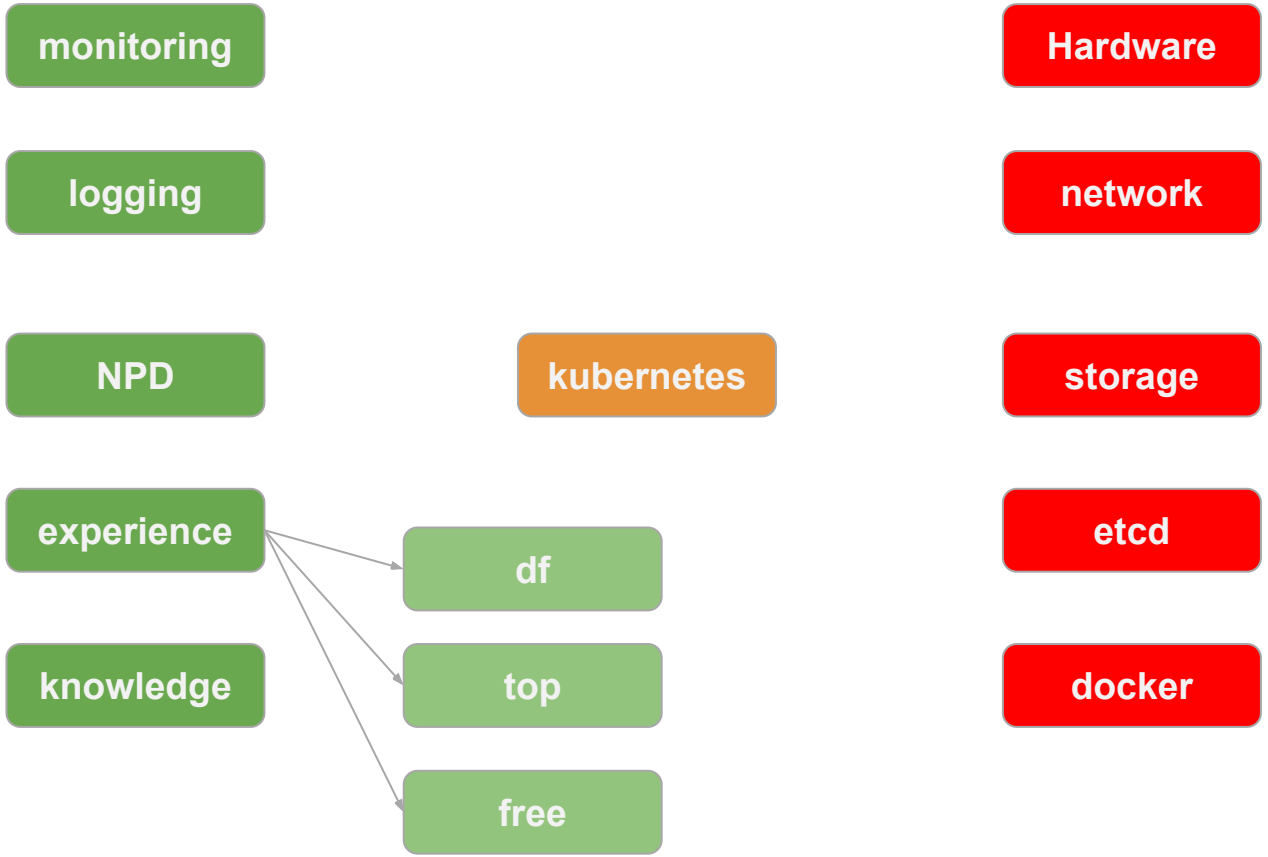
All nodes/masters restart at the same time, but LB is fine:

- Qos and traffic control

Deal with abnormal

Tools	symptom	Root Cause
nsenter	Pod pending	Kernel panic
describe	Pod terminating	code bug
Logs -p	Pod crash	Nfs hang
dmesg	Svc deny	docker hang
journalctl	Node notready	bind mount
tcpdump	app notwork	Network policy

Deal with abnormal



Q&A

caicloud
才云

Thanks！