

Kubernetes in Production

what can go wrong?!



Who we are

Jan Korger

jan.korger@egym.de

Jivko Vantchev

jivko.vantchev@egym.de

Beatrix Vad

beatrix.vad@egym.de

Gabrielle Juneau

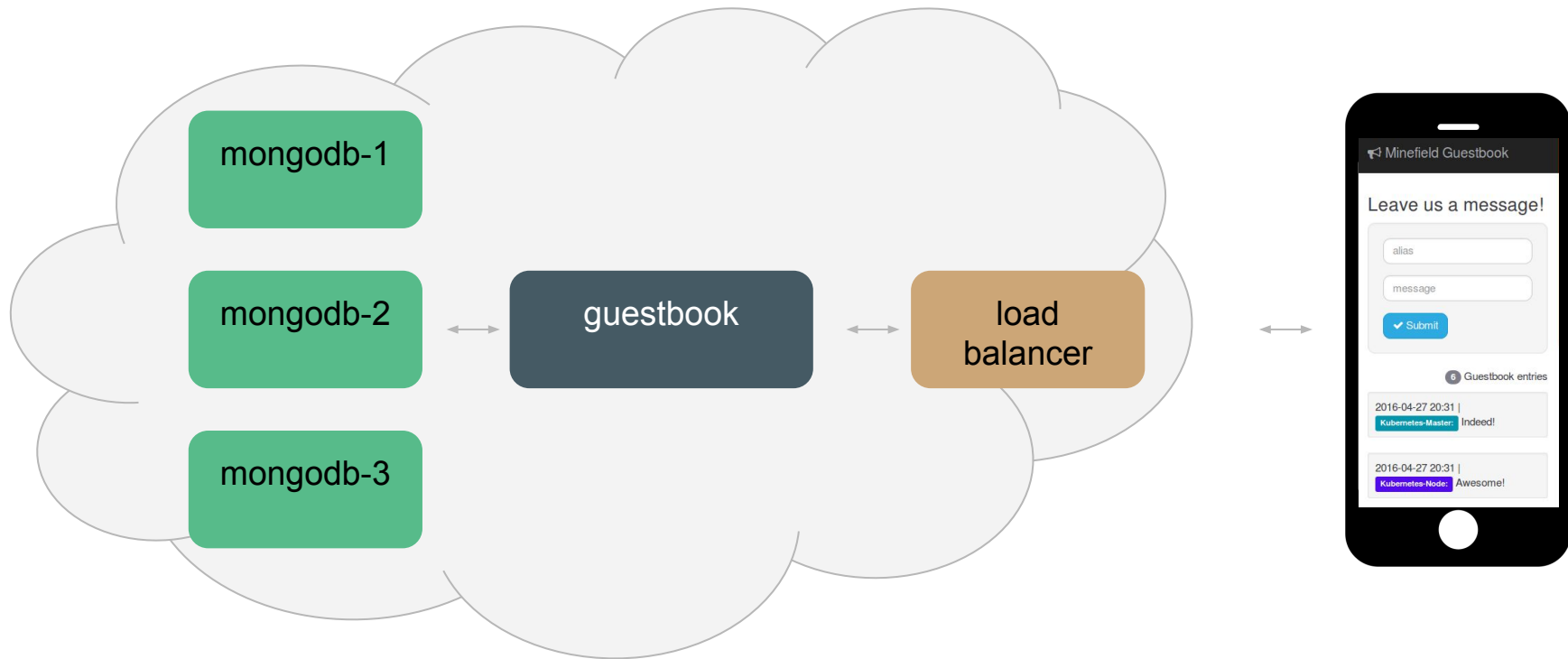
gabrielle.juneau@egym.de



Introduction Round

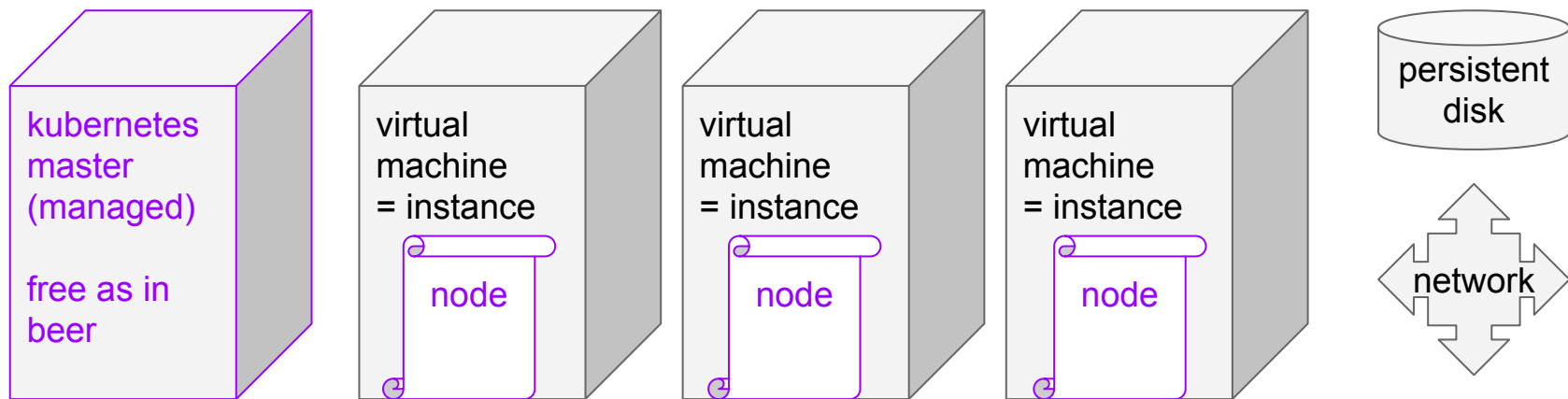
- Who is using Docker?
 - Who is using Docker in Production?
 - Anyone using Docker on Windows ? ;)
- Who is using Kubernetes?
 - Who is using Kubernetes in Production?
- Where do your containers live?
 - public cloud (GCE, AWS, other)

Kubernetes example



<https://github.com/egymgmbh/kubernetes-web-mongo-sample>

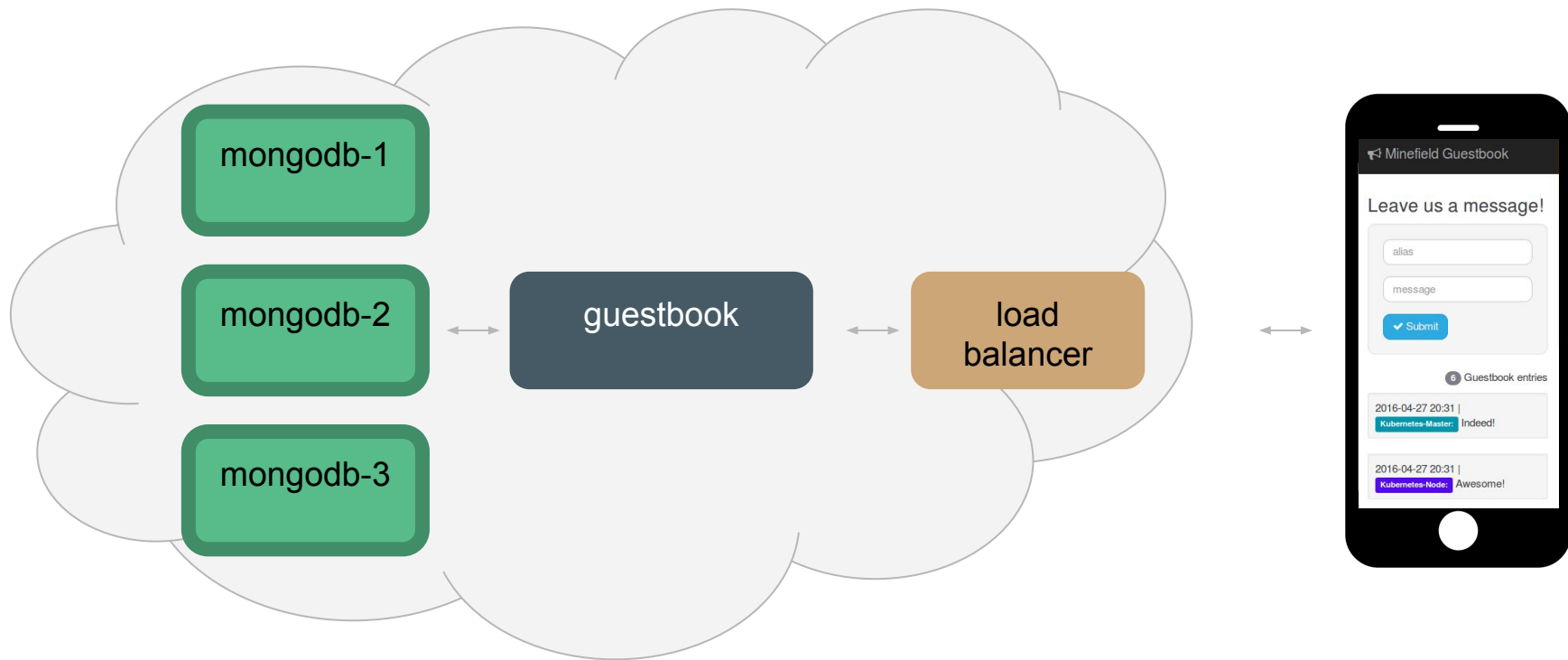
Kubernetes Cluster on Google Compute Engine



GCE = Google **C**ompute Engine = Infrastructure

GKE = Google Container Engine = **K**ubernetes

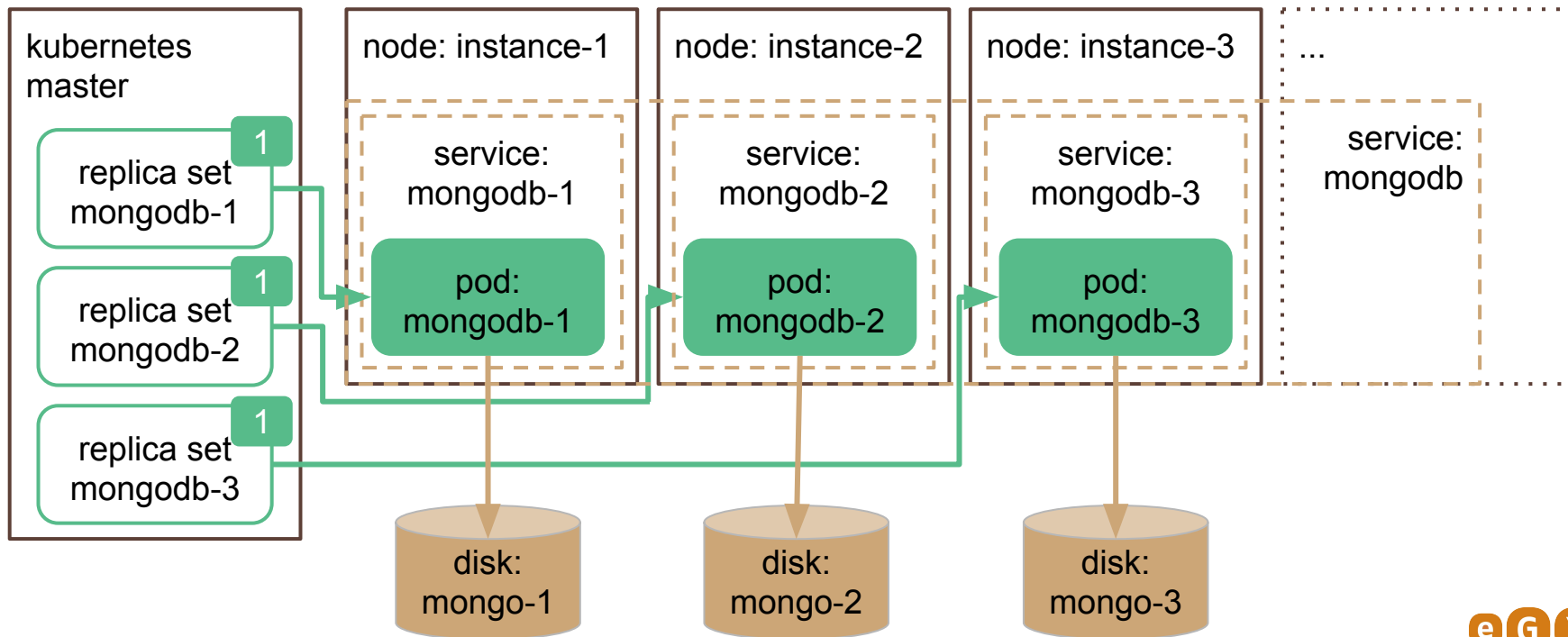
Kubernetes example: MongoDB Cluster



<https://github.com/egymgmbh/kubernetes-web-mongo-sample>

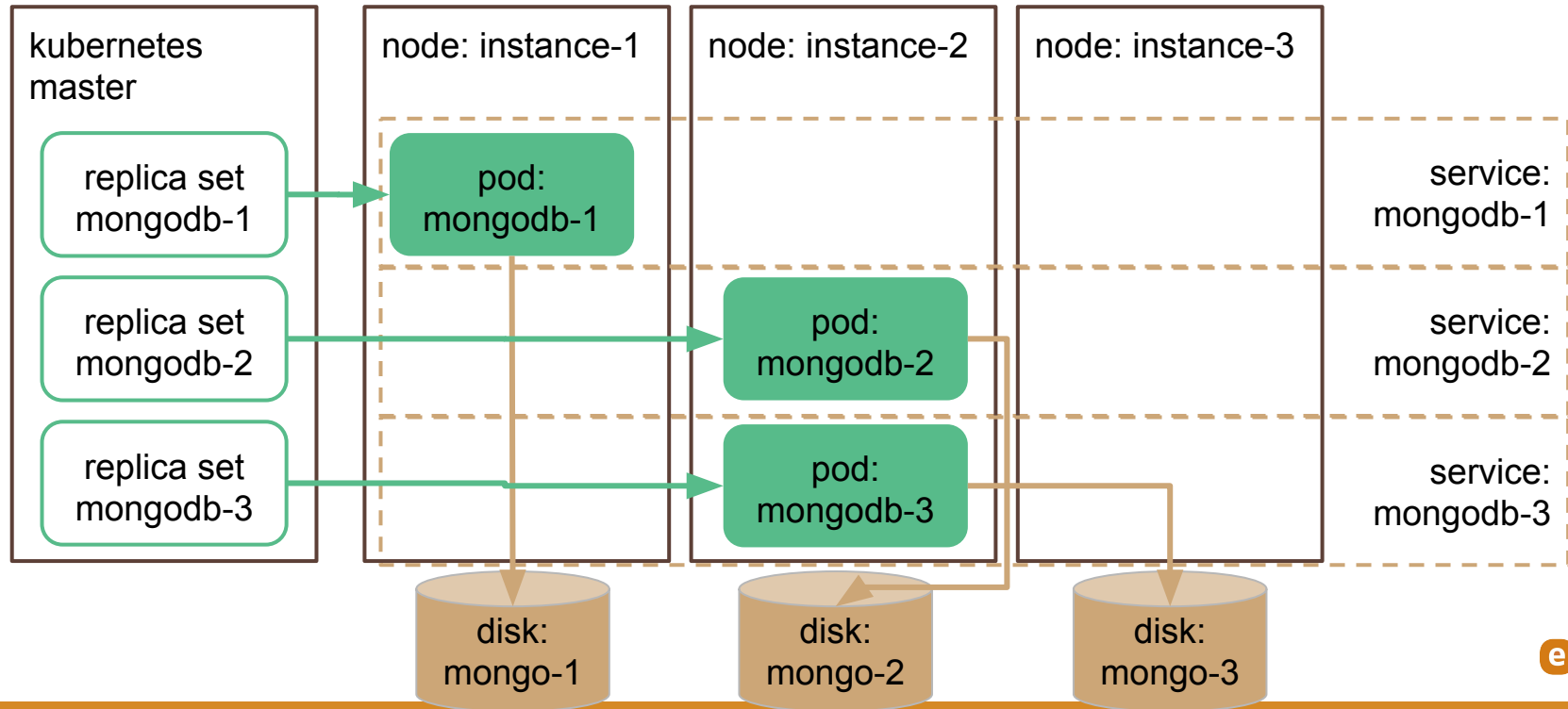
Running MongoDB cluster in a Kubernetes cluster

GCE instances = Kubernetes nodes



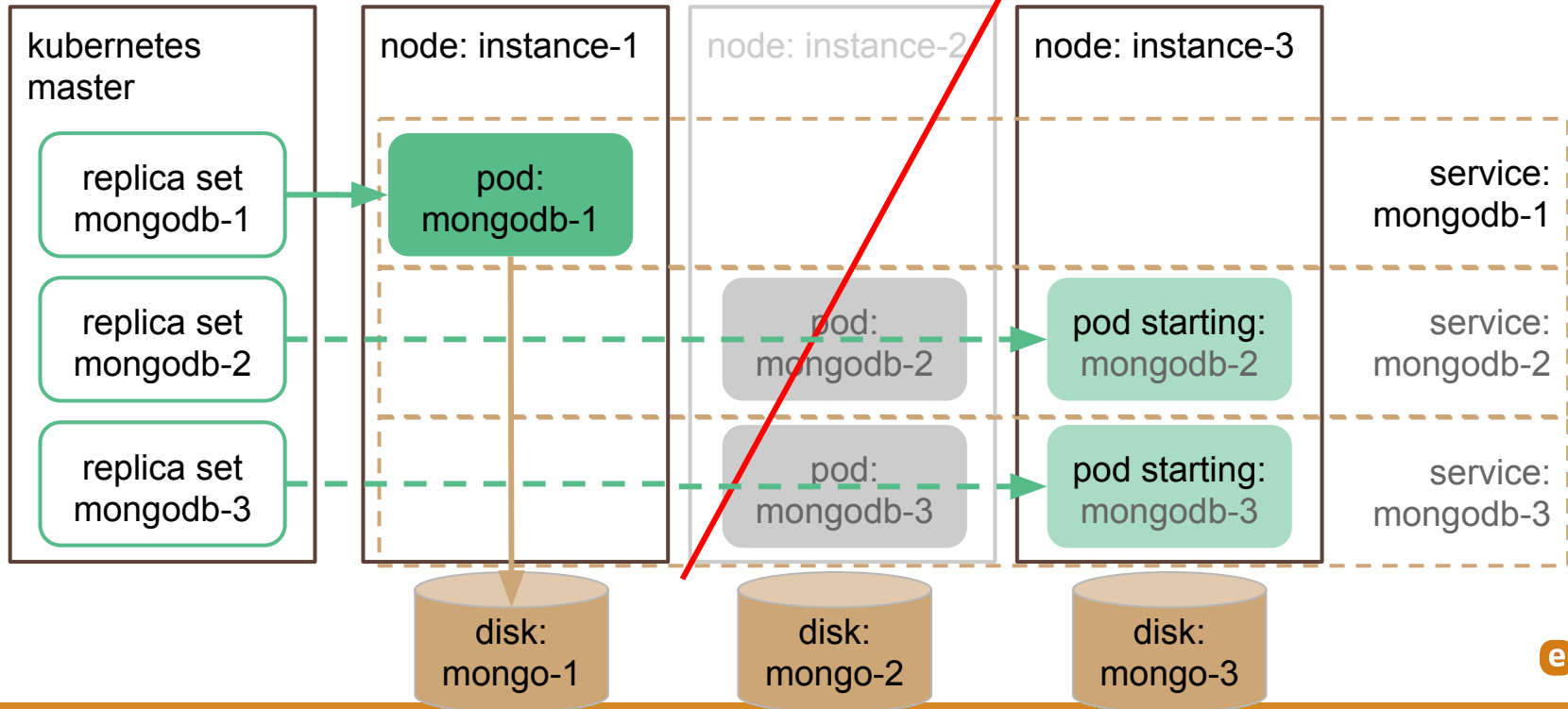
What will happen ...

GCE instances = Kubernetes nodes



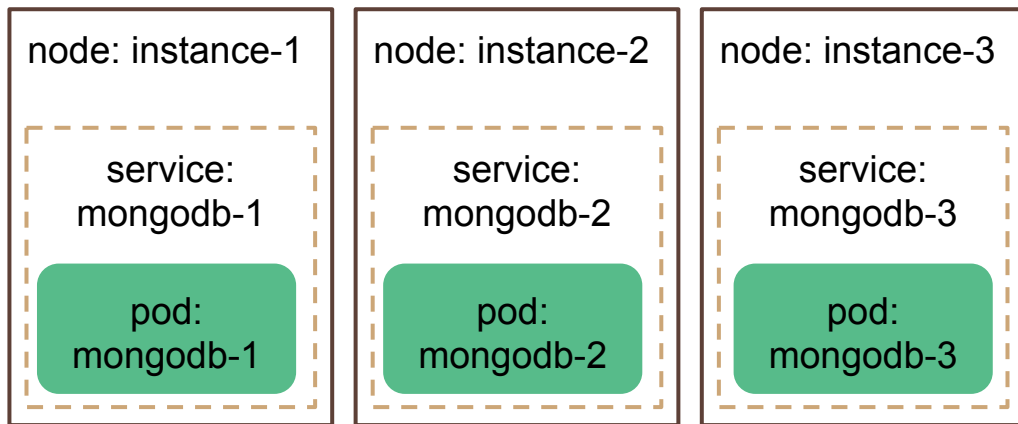
What happens next ...

GCE instances = Kubernetes nodes



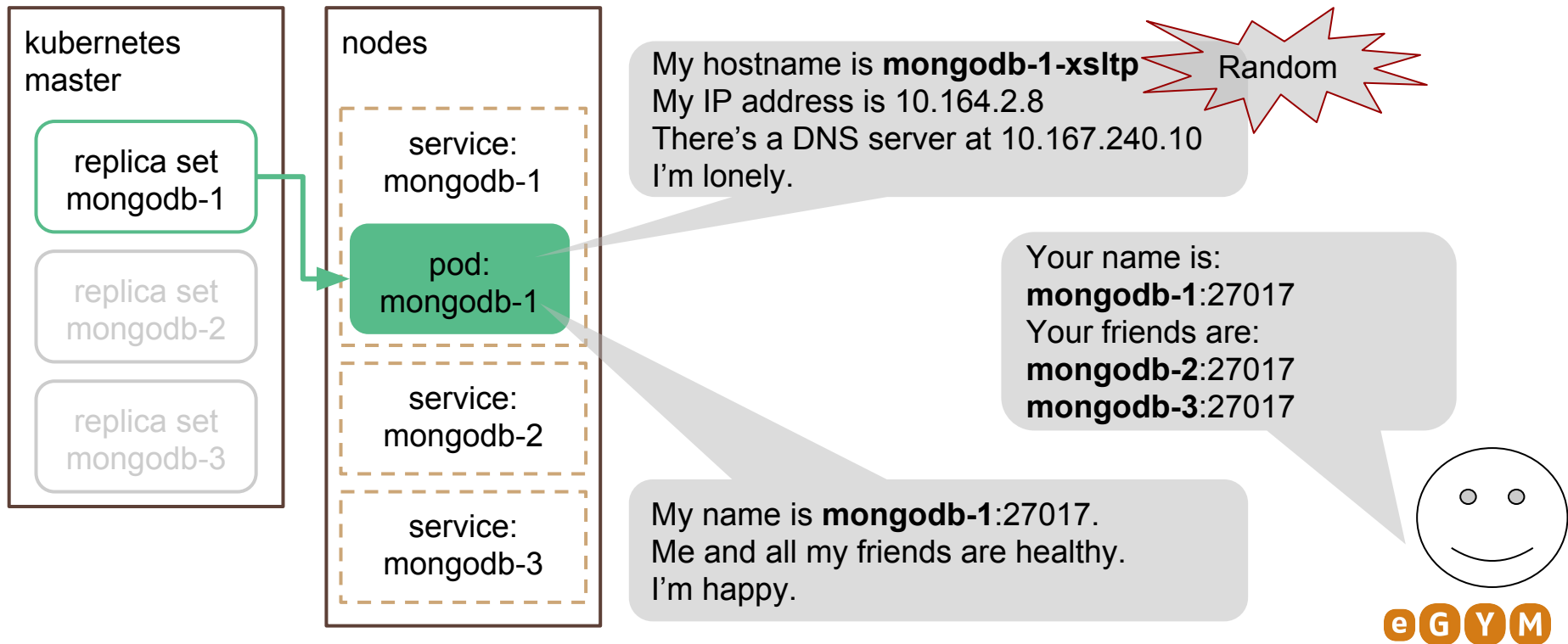
How to prevent clustering pods on single node?

GCE instances = Kubernetes nodes

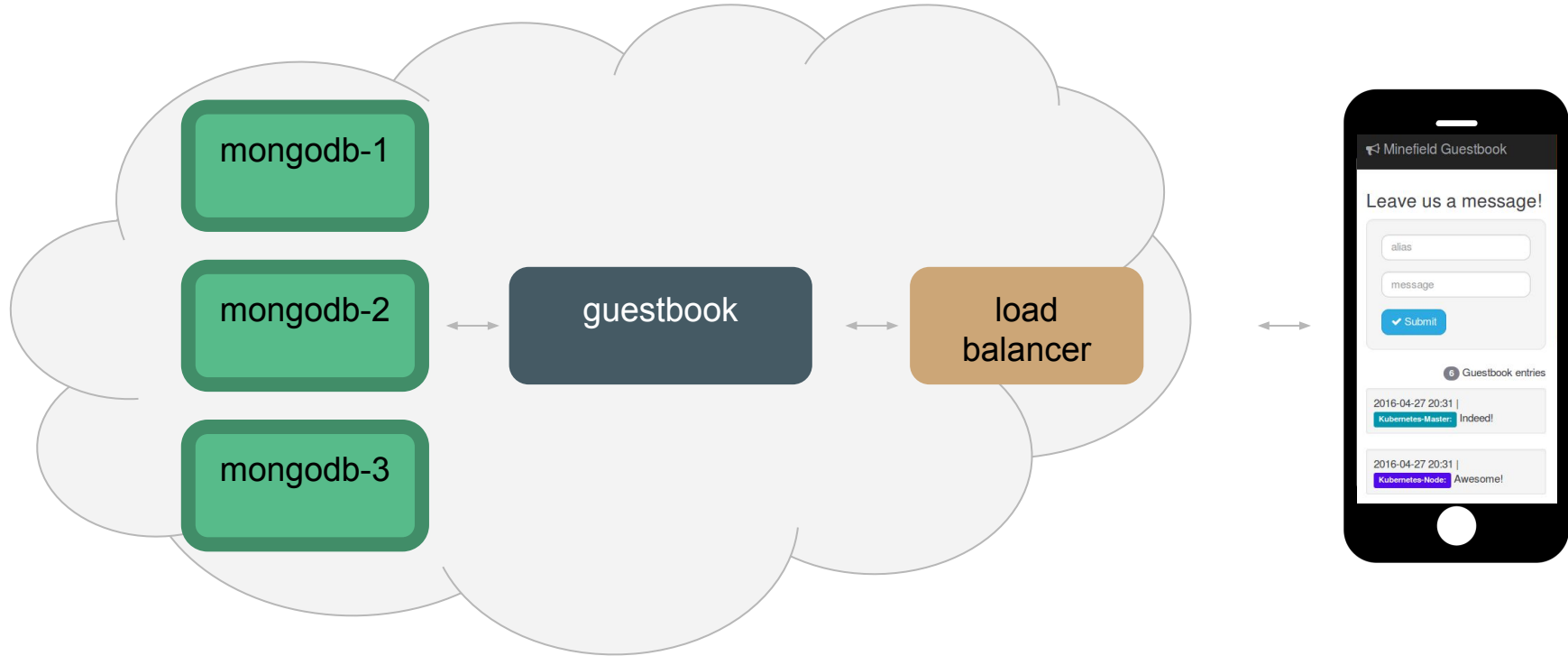


- ❑ [node labels & selectors](#)
- ❑ [node affinity \(alpha in v1.2\)](#)
- ❑ [pod sets \(probably in v1.3\)](#)

Configuring MongoDB to live inside Kubernetes

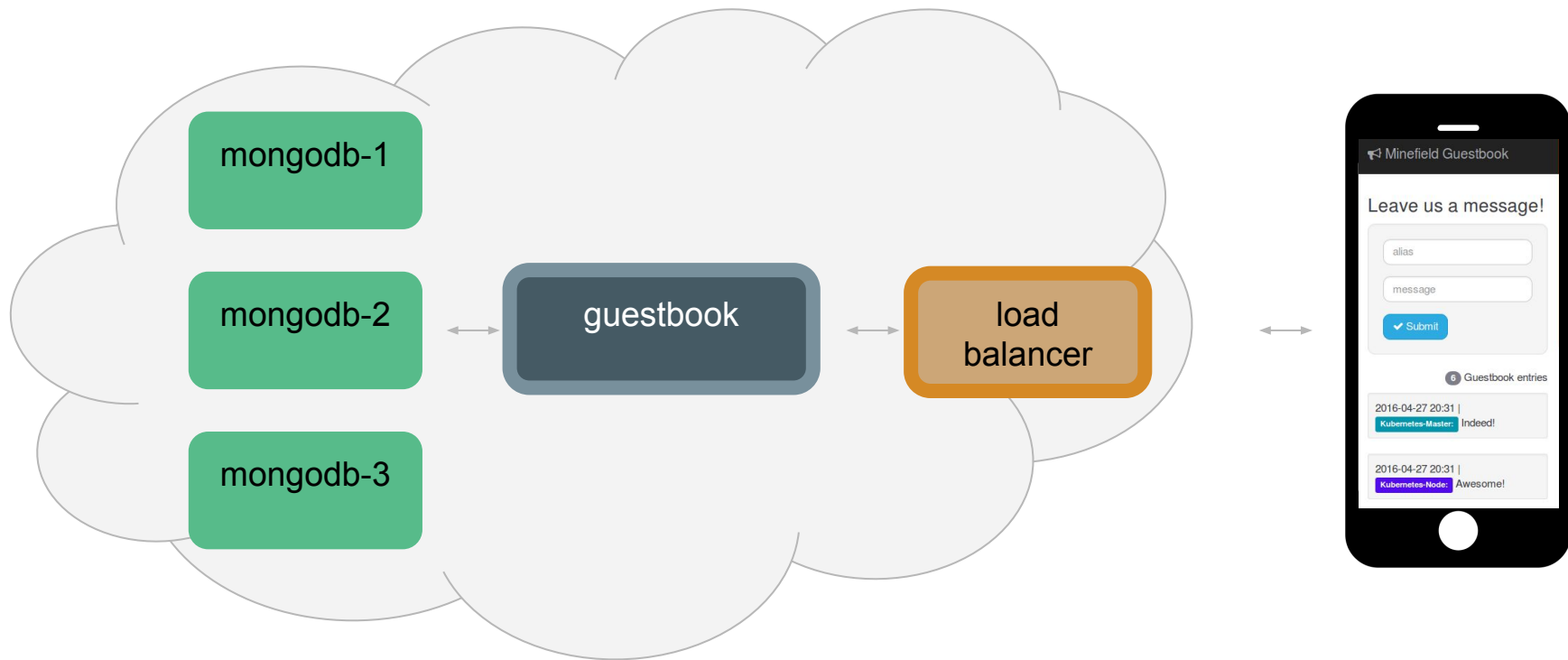


Demo



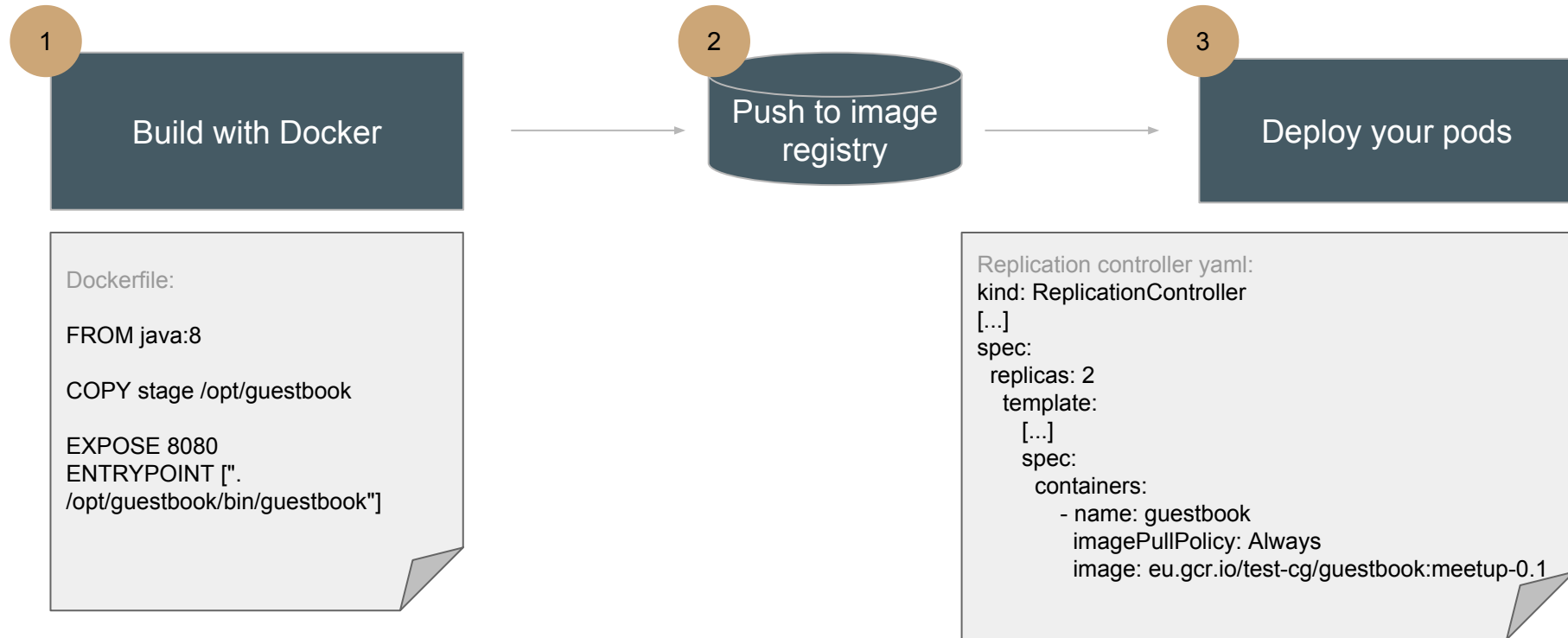
<https://github.com/egymgmbh/kubernetes-web-mongo-sample>

Kubernetes example: Guestbook

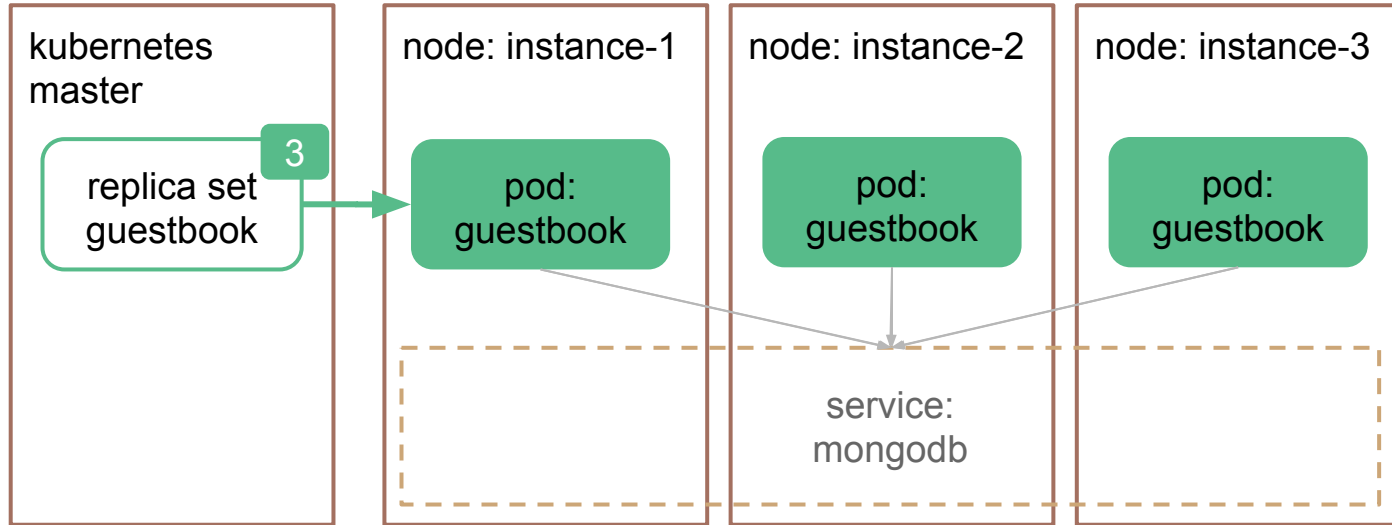


<https://github.com/egymgmbh/kubernetes-web-mongo-sample>

Deployment steps



Stateless application in a Kubernetes cluster



Exposing your application

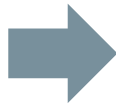
- **Service**

- ClusterIP (default)
- NodePort
- LoadBalancer



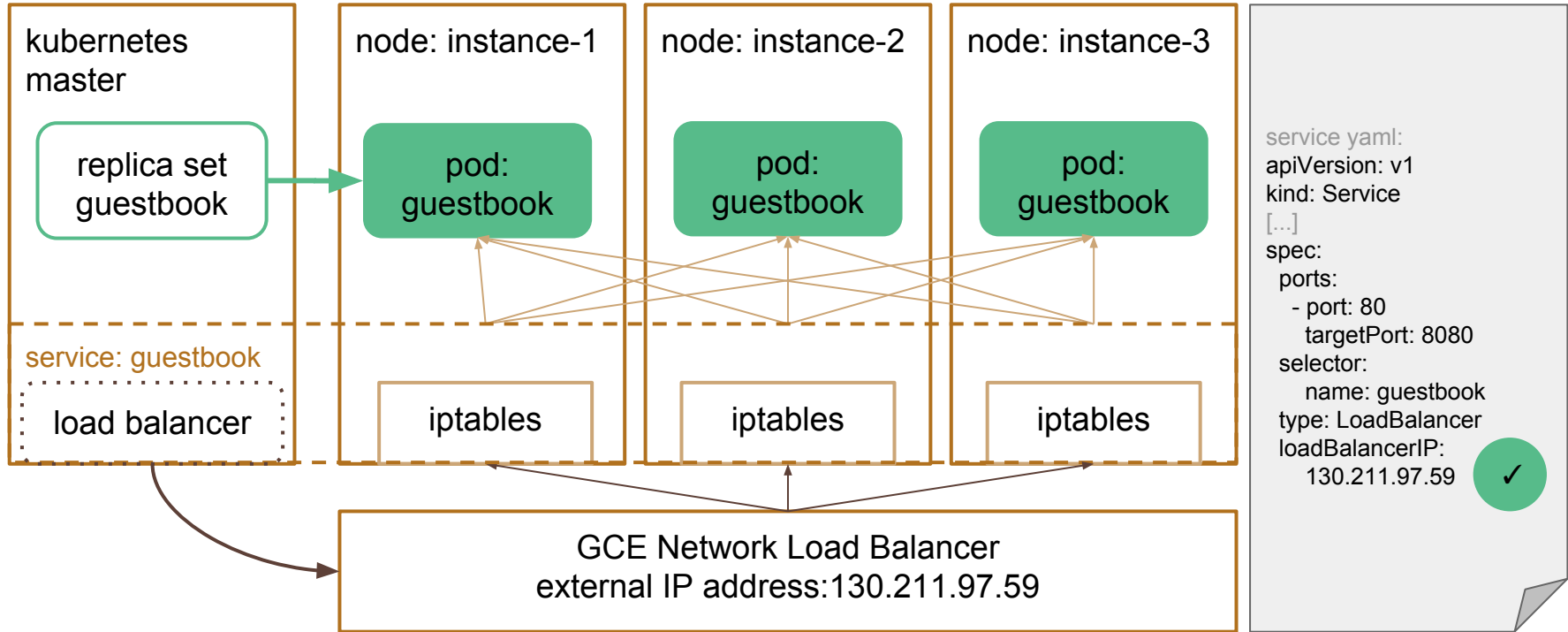
- **Ingress (v1.2)**

- Fanout
- VirtualHosts
- TLS termination
- Load balancing

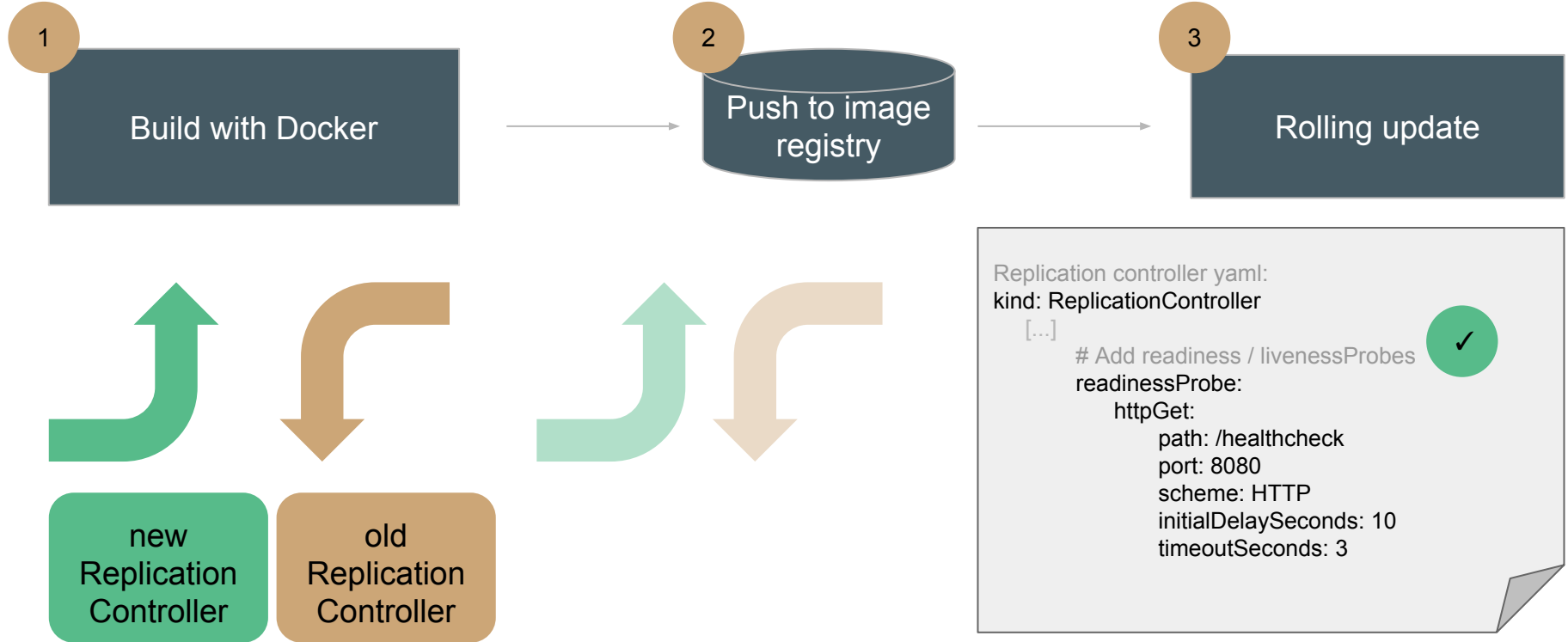


Ingress meets Let's Encrypt

Exposing your application: LoadBalancer

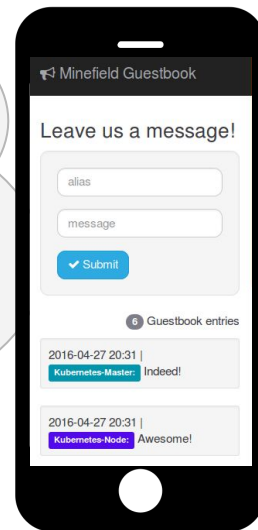


Rolling update

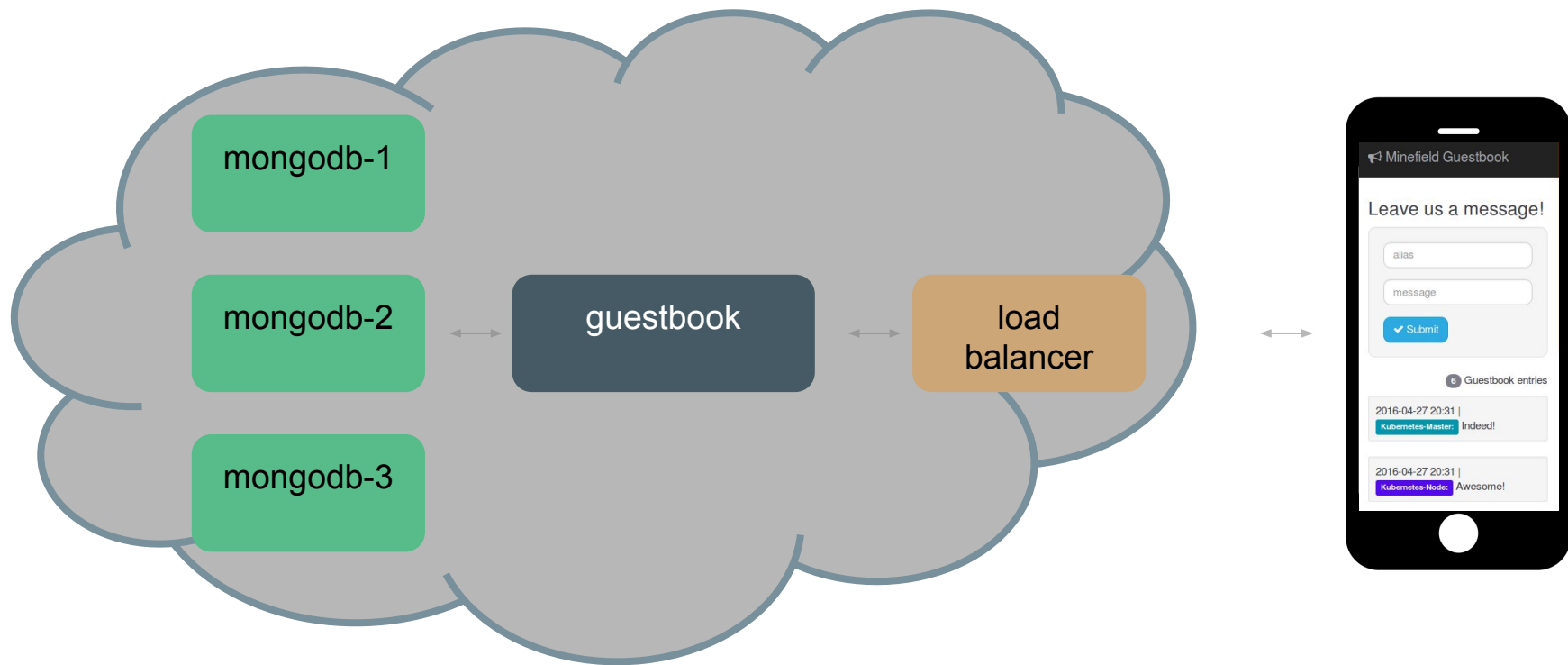


Demo

bit.ly/kubernetes-guestbook



Kubernetes example: Production



<https://github.com/egymgmbh/kubernetes-web-mongo-sample>

Kubernetes in Production

Q: For how long have we used it?

A: More or less from the first official release.

Q: What is running on Kubernetes?

A: Some internal services, as well as some customer facing ones (Web, APIs)

Q: So how is it going, were there any problems?

A: Well...

Kubernetes in Production

Things we have encountered so far:

- Nodes die unexpectedly. Not often but it happens. Redundancy with a good pod spread is required for no or minimal down time.
- GKE Cluster upgrades → mostly inevitable → better have 3 or more nodes.
- GKE Cluster upgrades *will* change your node IPs - update your firewall rules with external services.

Kubernetes in Production

- Kubernetes is a fast evolving ecosystem. You need to keep up and have to expect that what works today won't necessarily work the same way tomorrow → rolling updates with the same image tag.
- GCE and GKE are also evolving and improving. Logging is one example, but there are also some bugs popping up.
- There were/are some persistent GKE errors as well → LoadBalancer cannot be created, since the IP cannot be acquired (event though already in-use by it).

Kubernetes in Production

- Instance dies → sometimes the persistent disk is still associated with it → free it manually via Cloud Console (known [problem](#))

```
FailedMount   {kubelet gke-cluster-d683da1e-node-g78d}  
Unable to mount volumes for pod "mongodb-replica-3-test-controller-hfw1q_minefield":  
Could not attach GCE PD "mongo-db-test-disk-3". Timeout waiting for mount paths to be created.
```

You have to manually detach the PDs from the instance and k8s should auto-recover.

Kubernetes in Production

Some useful tips:

- Having automatic PD snapshots is nice to have (be careful to also delete old ones). Can be done via [service accounts](#).
- Monitoring and timely alerts are essential. Log-based Metrics are helpful! Accidents happen even in robust environments.
- Know your pods' [imagePullPolicy](#)!
- Keep an eye on that MongoDB Cluster! Replicas get removed (state REMOVED). You may need to force the reconfiguration.

Kubernetes in Production some Advice

- Keep an eye on node distribution (if you are not using any sort of [affinity](#))
 - especially when scaling up your cluster.
- It's a good idea to have readiness and liveness probes! Make them as lightweight as possible!

Overall: Kubernetes + GKE/GCE has some drawbacks but also a lot of advantages. Stability is sufficient and self-recovery works (in most cases).

Thank you for your attention!



www.egym.de

