



Kubernetes in Tuenti

Docker Meetup - 06/07/26

Jaime Soriano

github.com/jsoriano

Kubernetes in Tuenti

- Introduction
- Kubernetes
- New deployment workflow
- Summary, limitations and outcomes



01

Introduction

Background



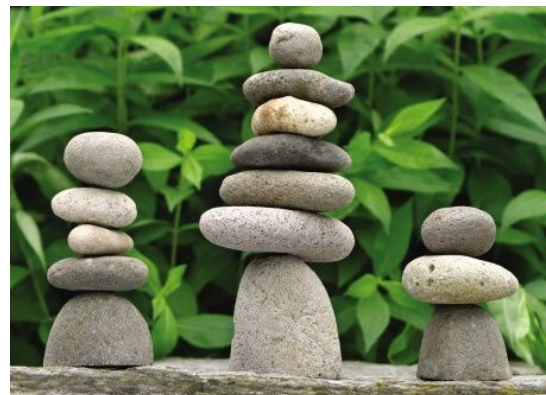
Social network in Spain
Millions of users



Communications platform
App + MVNO

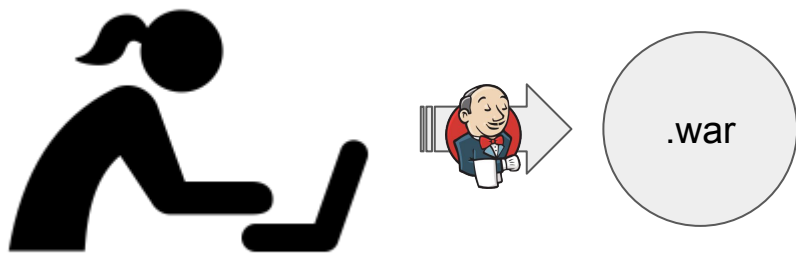
5 countries and several 100K
customers

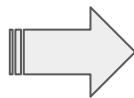
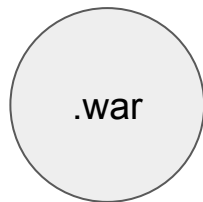


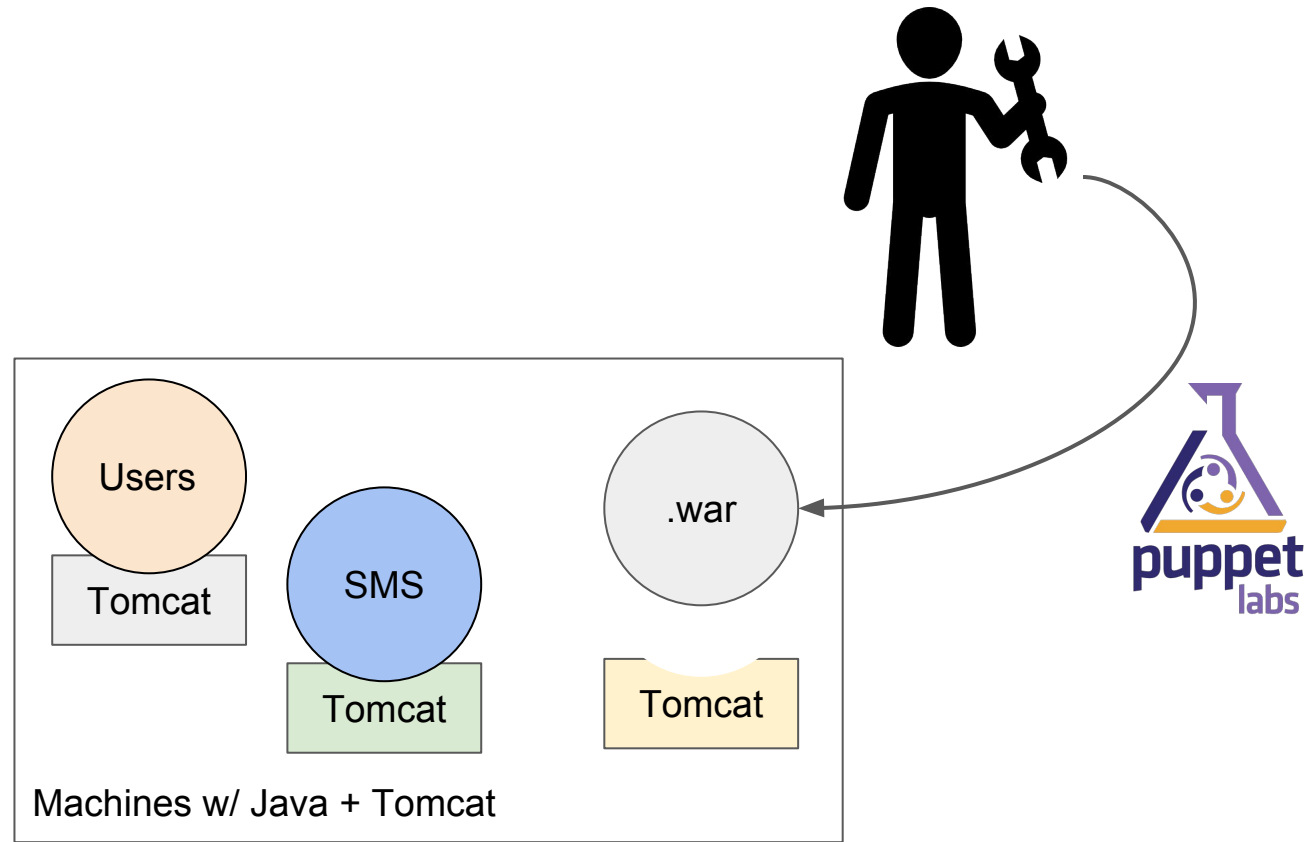


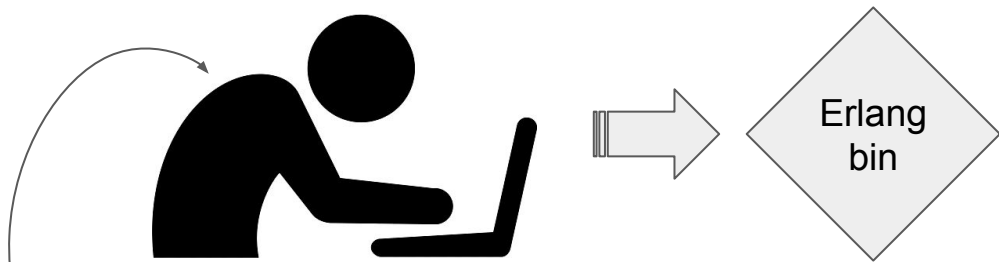
(Micro?) Services





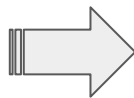
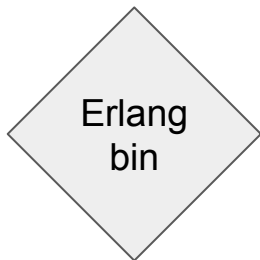






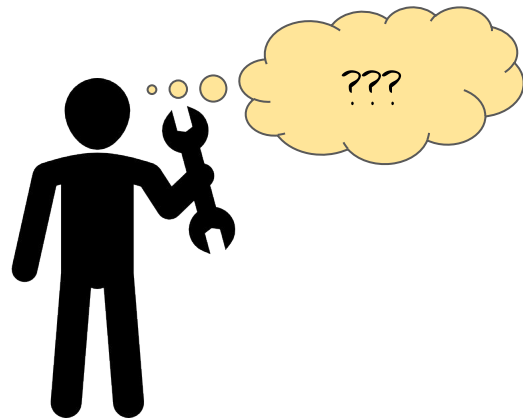
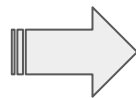
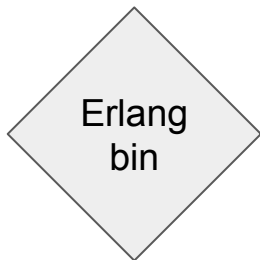
Erlang can cause
severe back pain

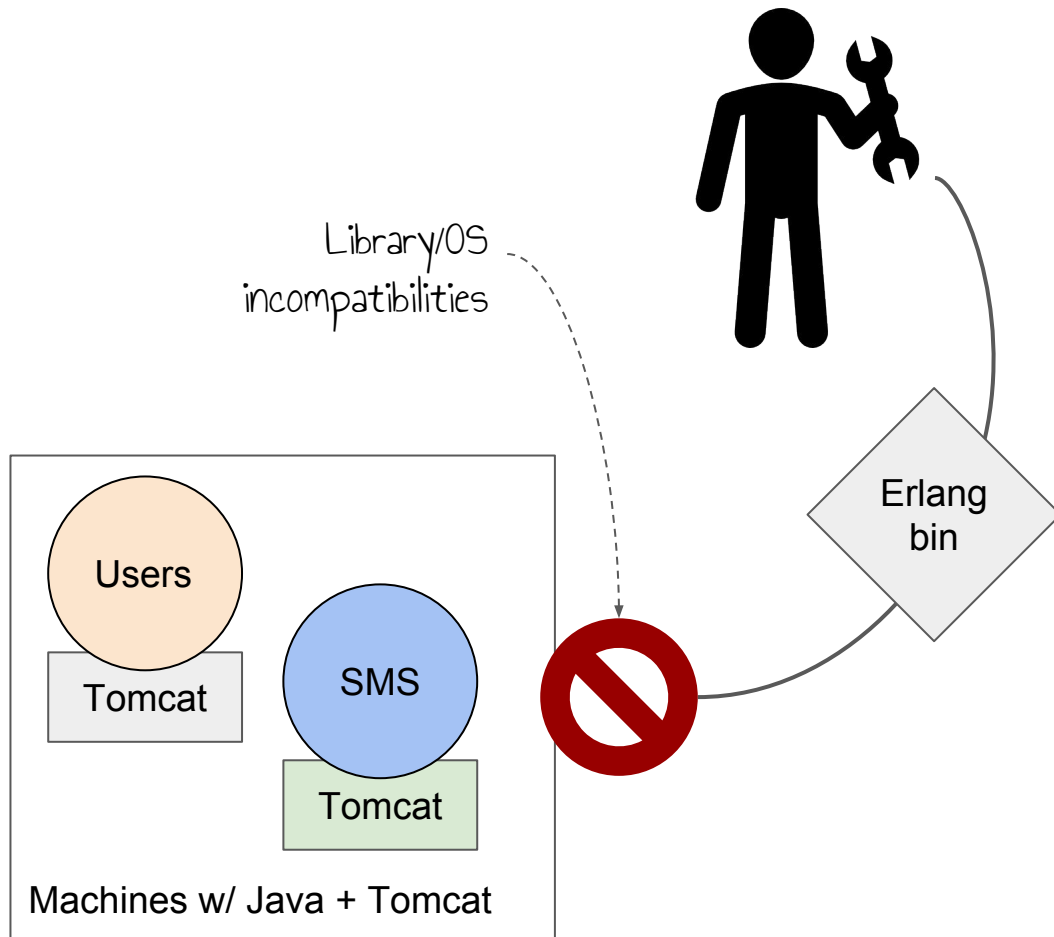


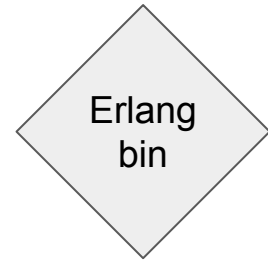
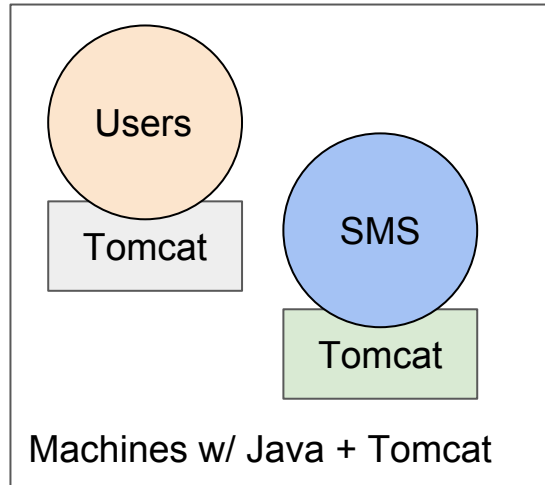
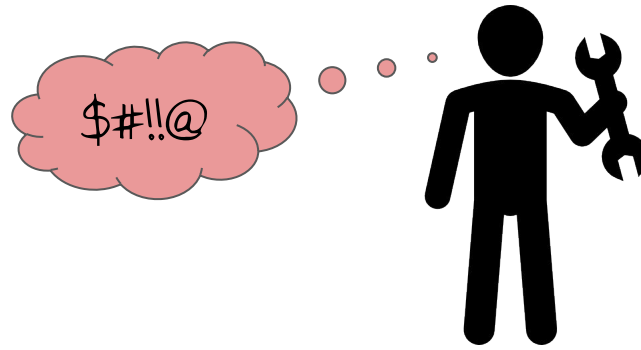


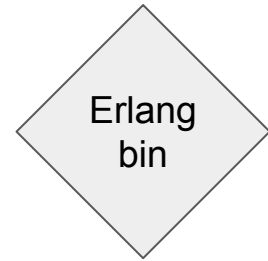
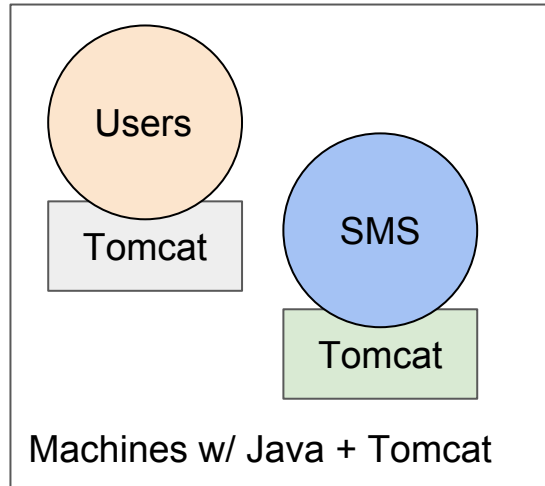
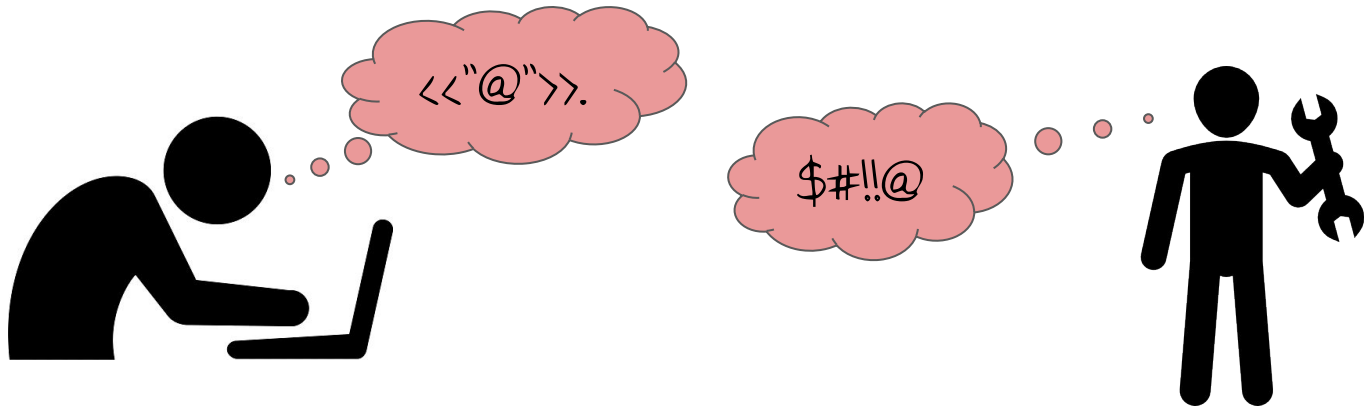
Unsuspecting SRE









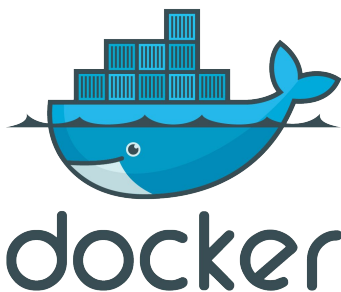


Problems

- Delays (two teams involved)
- Conflicts between services and environments
- Custom tools for deployments
- Knowledge barrier

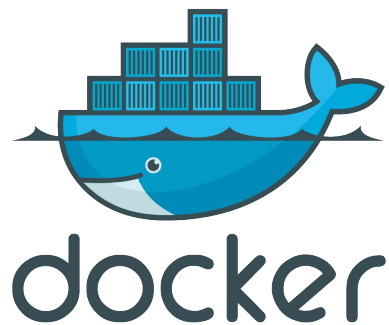


Inspirations



But...





Build, Ship, & Run
Any App, Anywhere

How to operate all this?



02

Kubernetes



kubernetes
by Google™

Kubernetes is an open-source system for automating deployment, operations, and scaling of containerized applications.



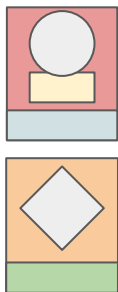
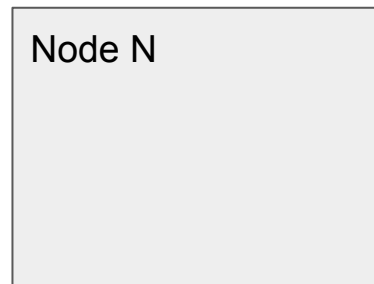
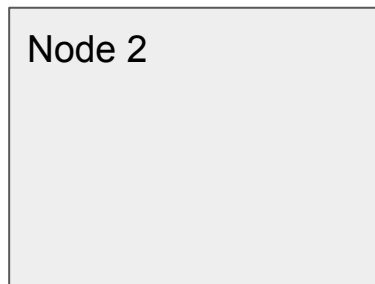
Node 1

Node 2

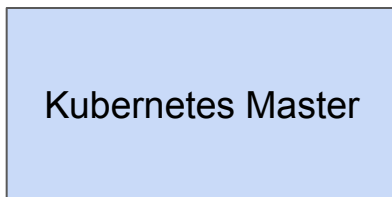
Node N

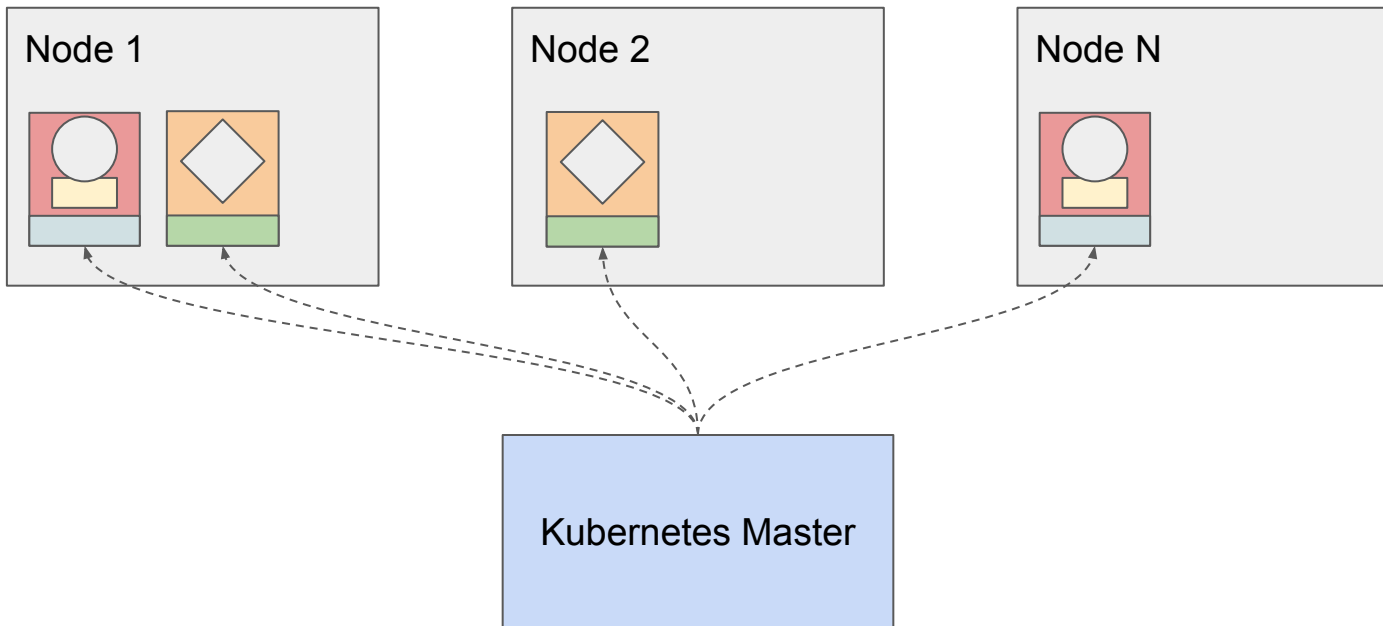
Kubernetes Master

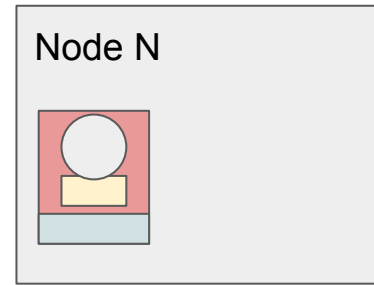
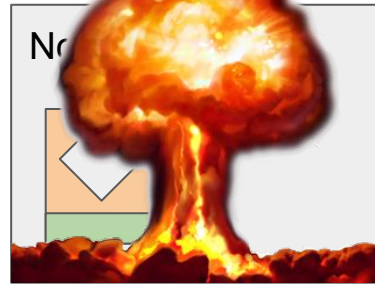
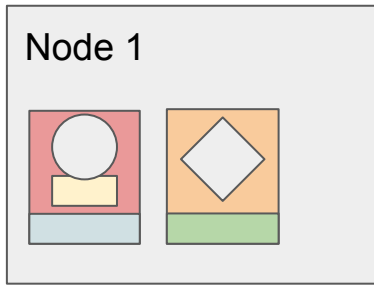




Deploy x2

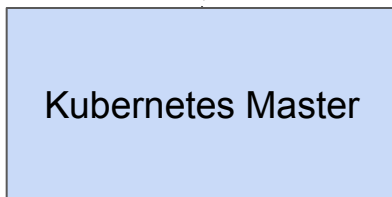
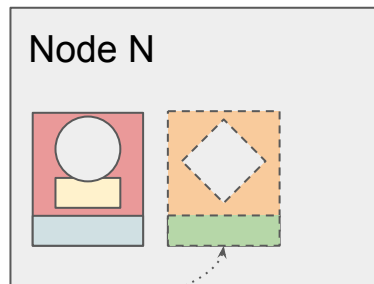
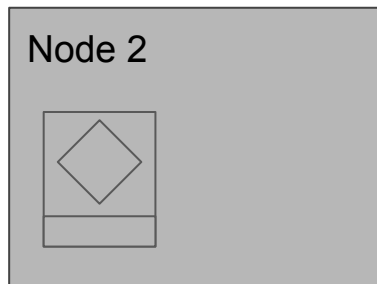
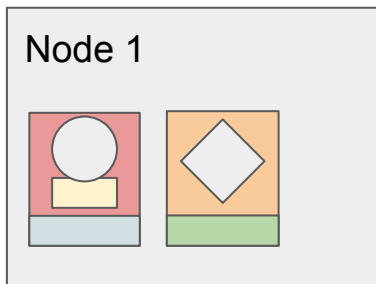


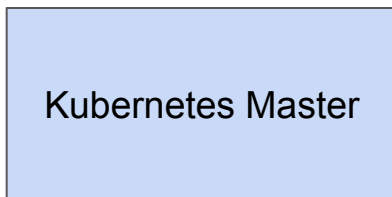
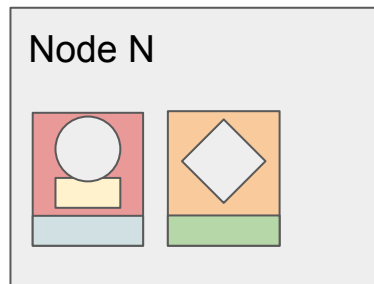
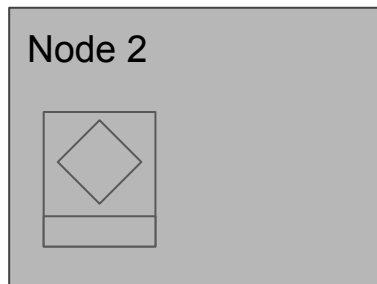
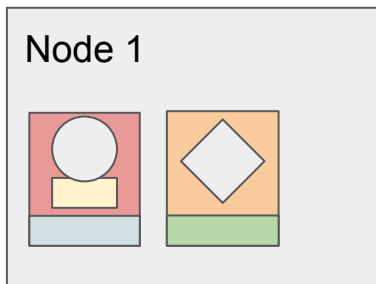




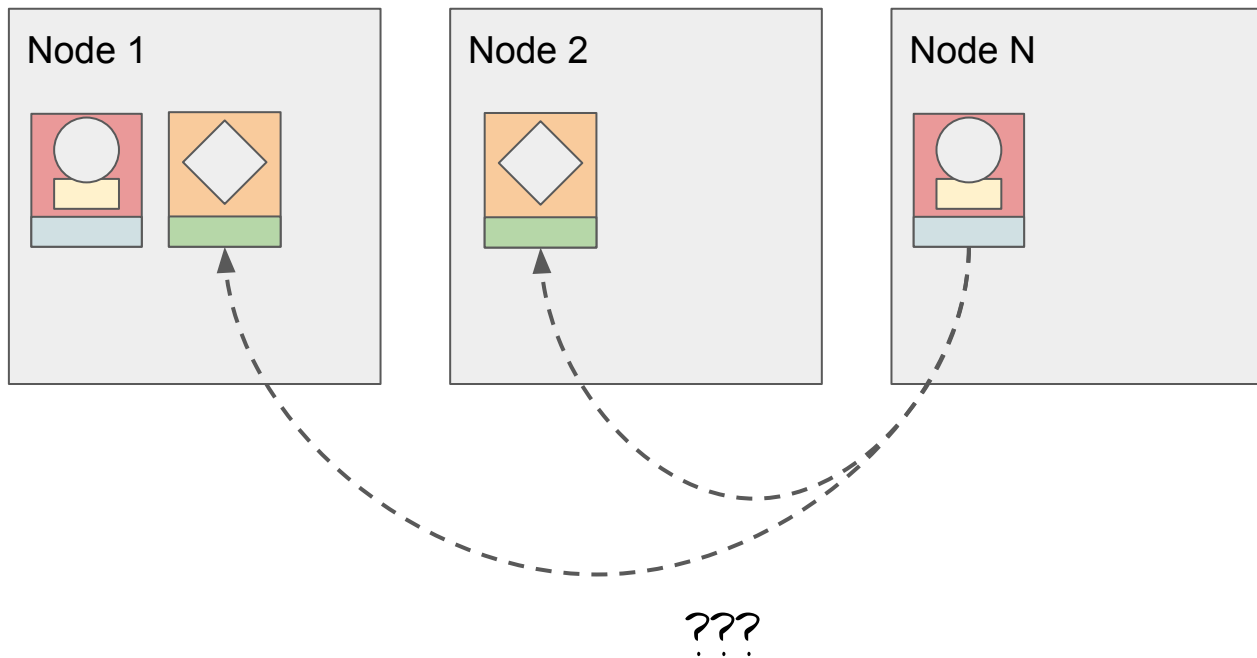
Kubernetes Master



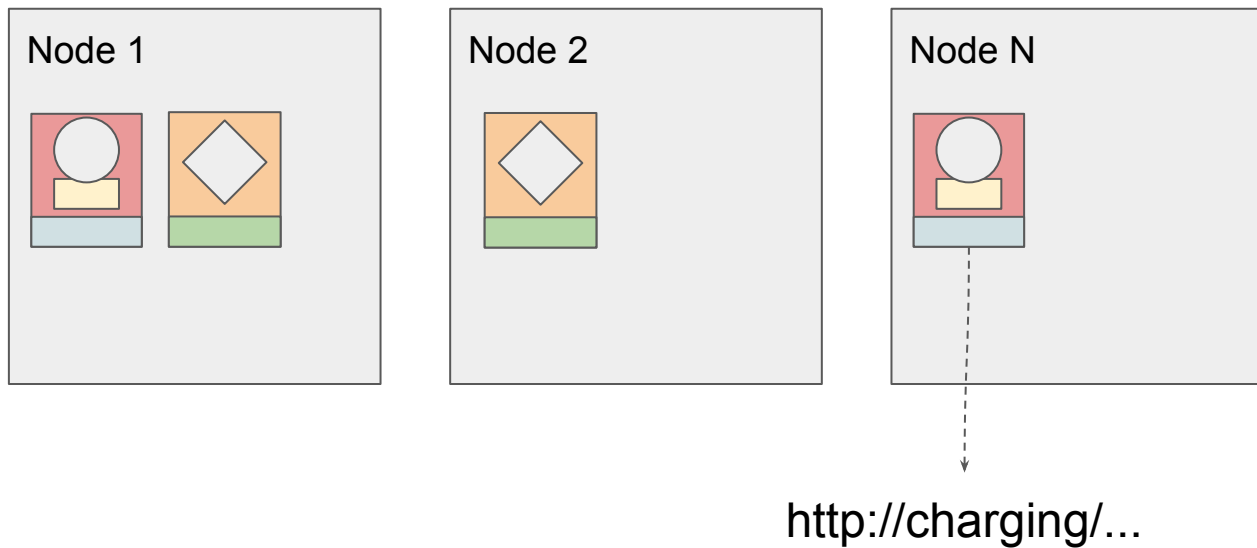




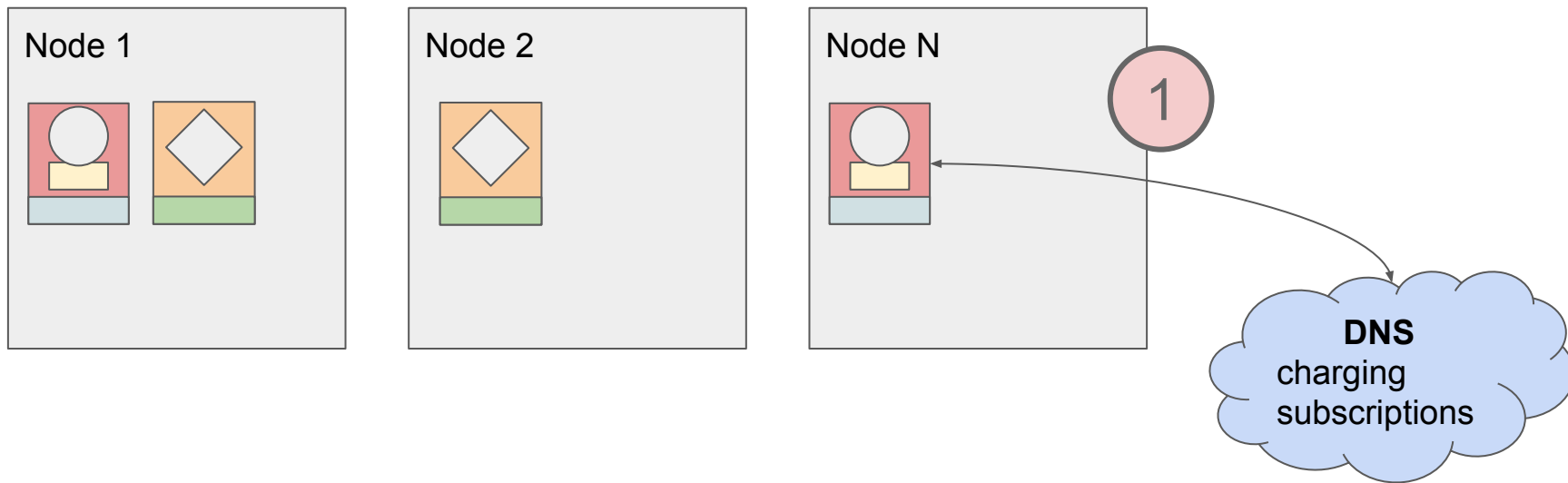
Service Discovery from the inside



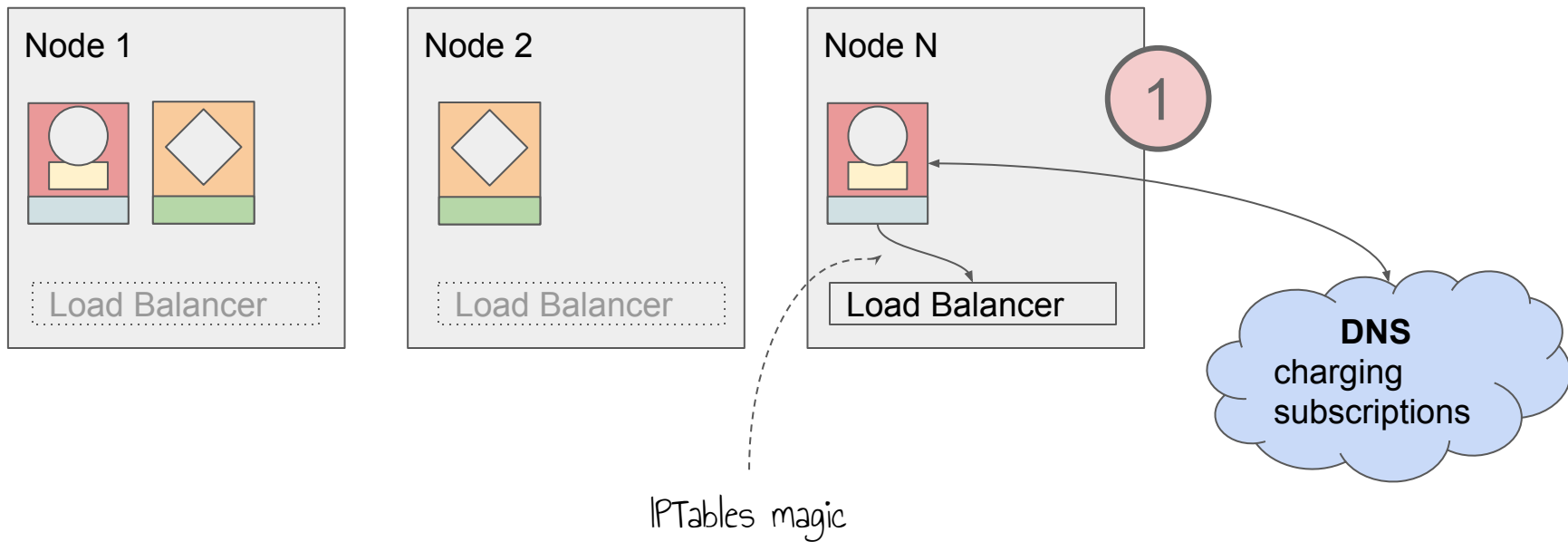
Service Discovery from the inside



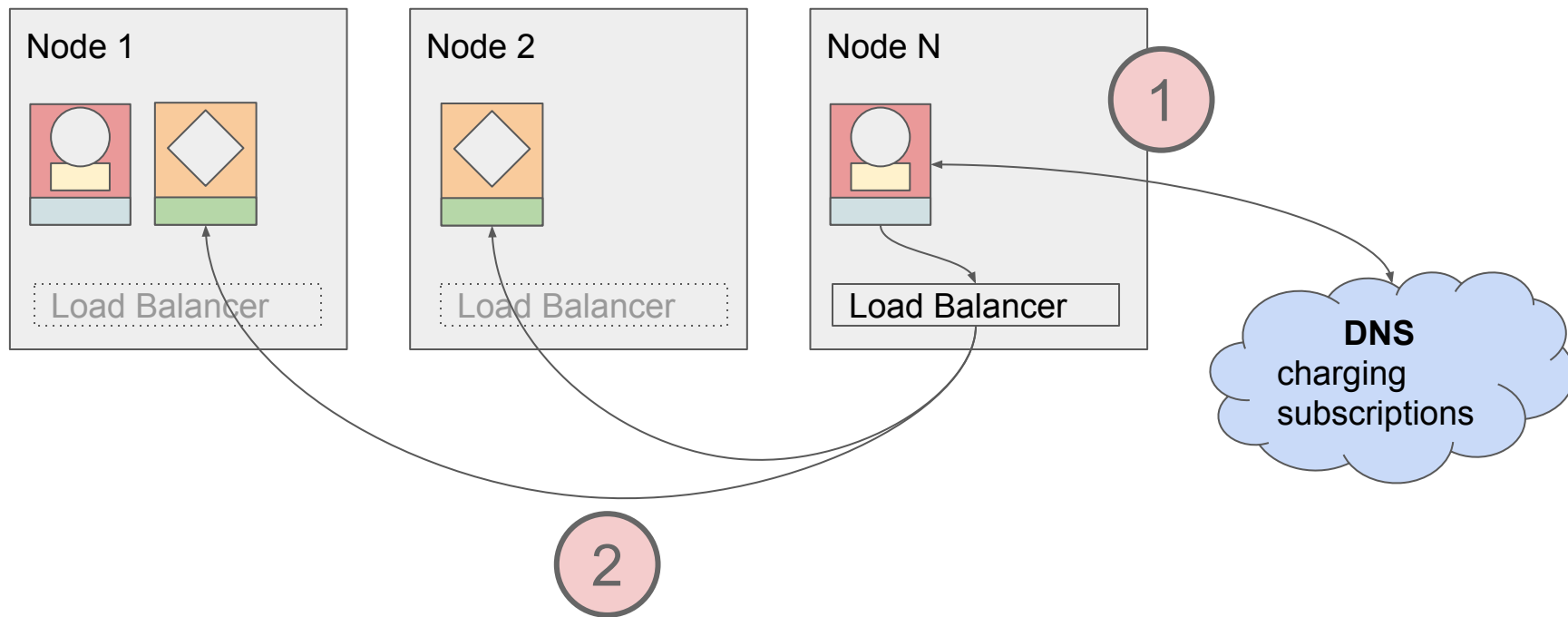
Service Discovery from the inside



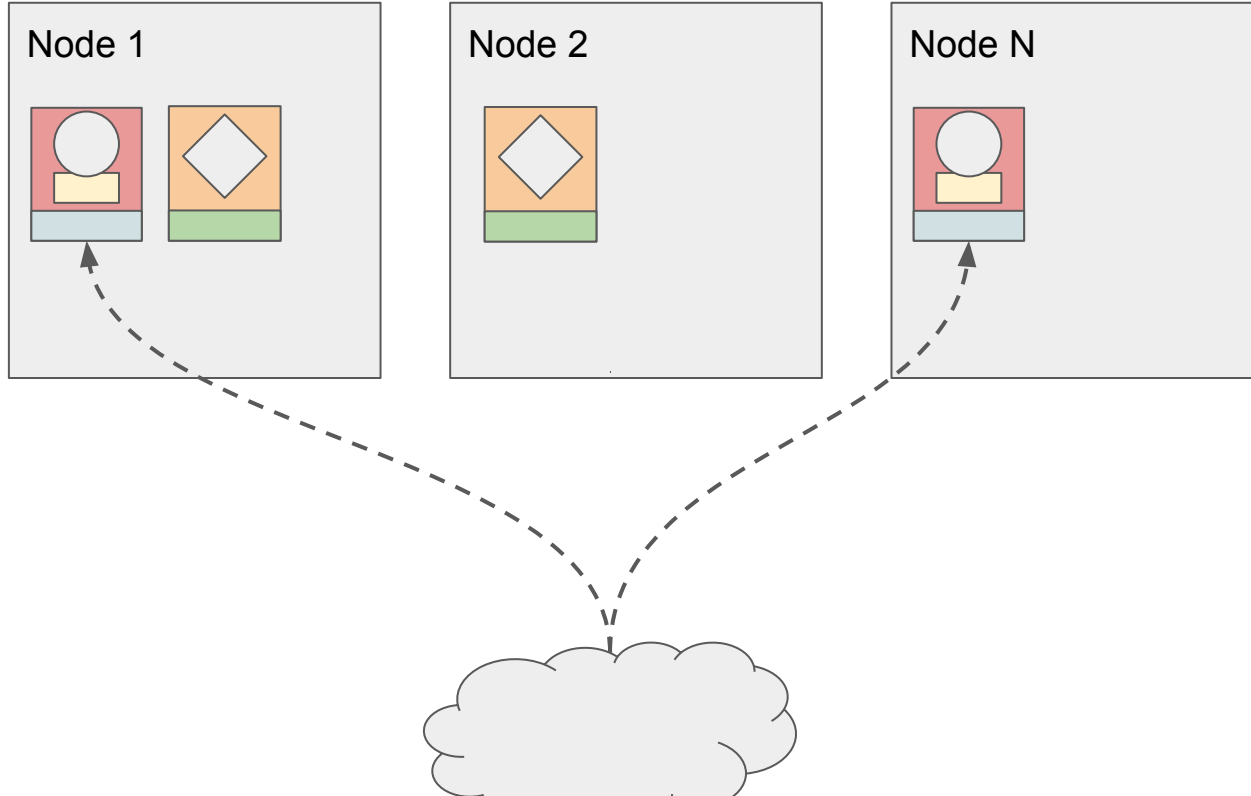
Service Discovery from the inside



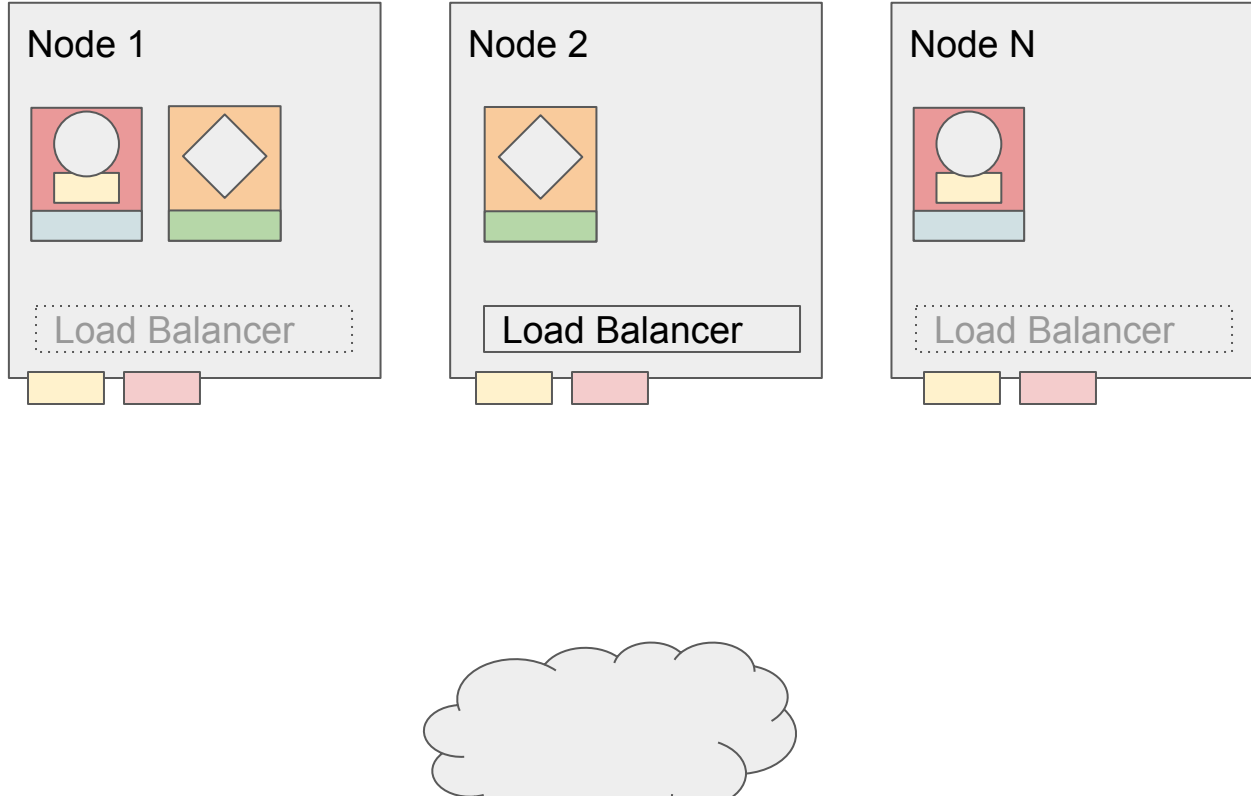
Service Discovery from the inside



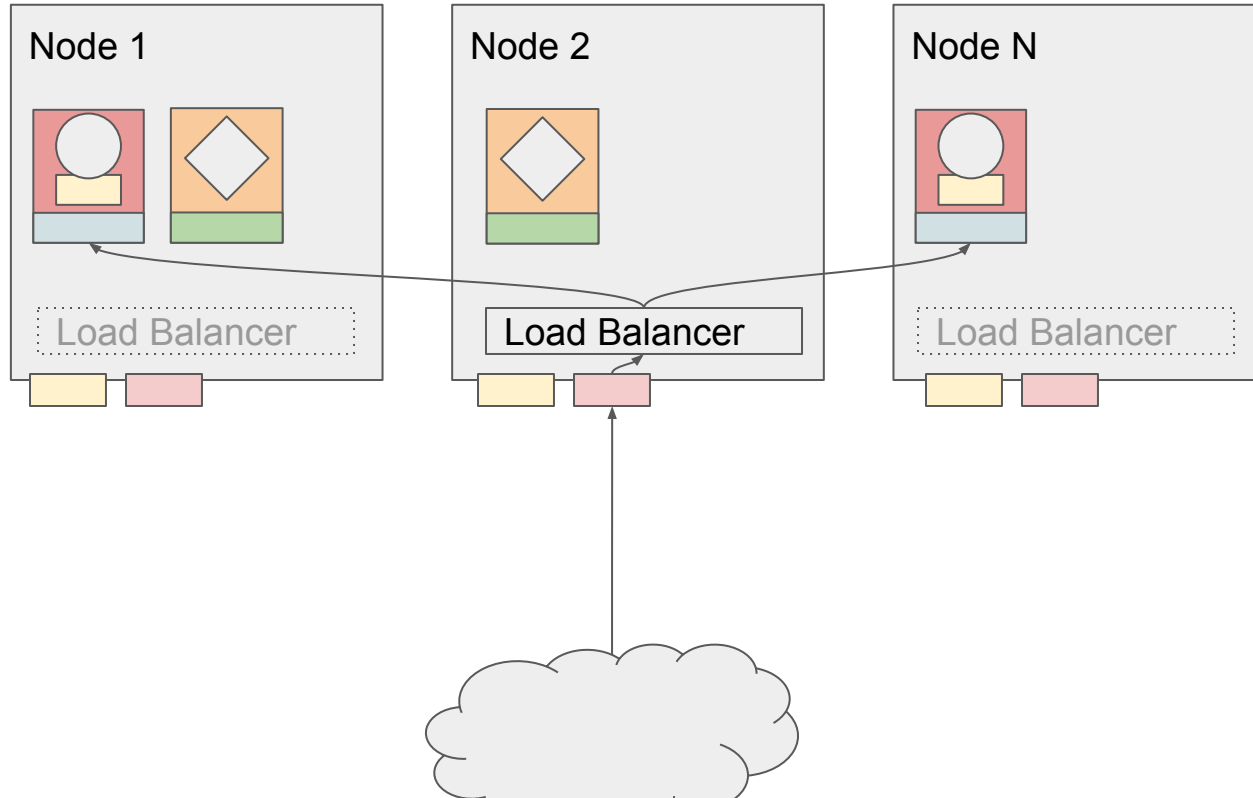
... and from the outside?



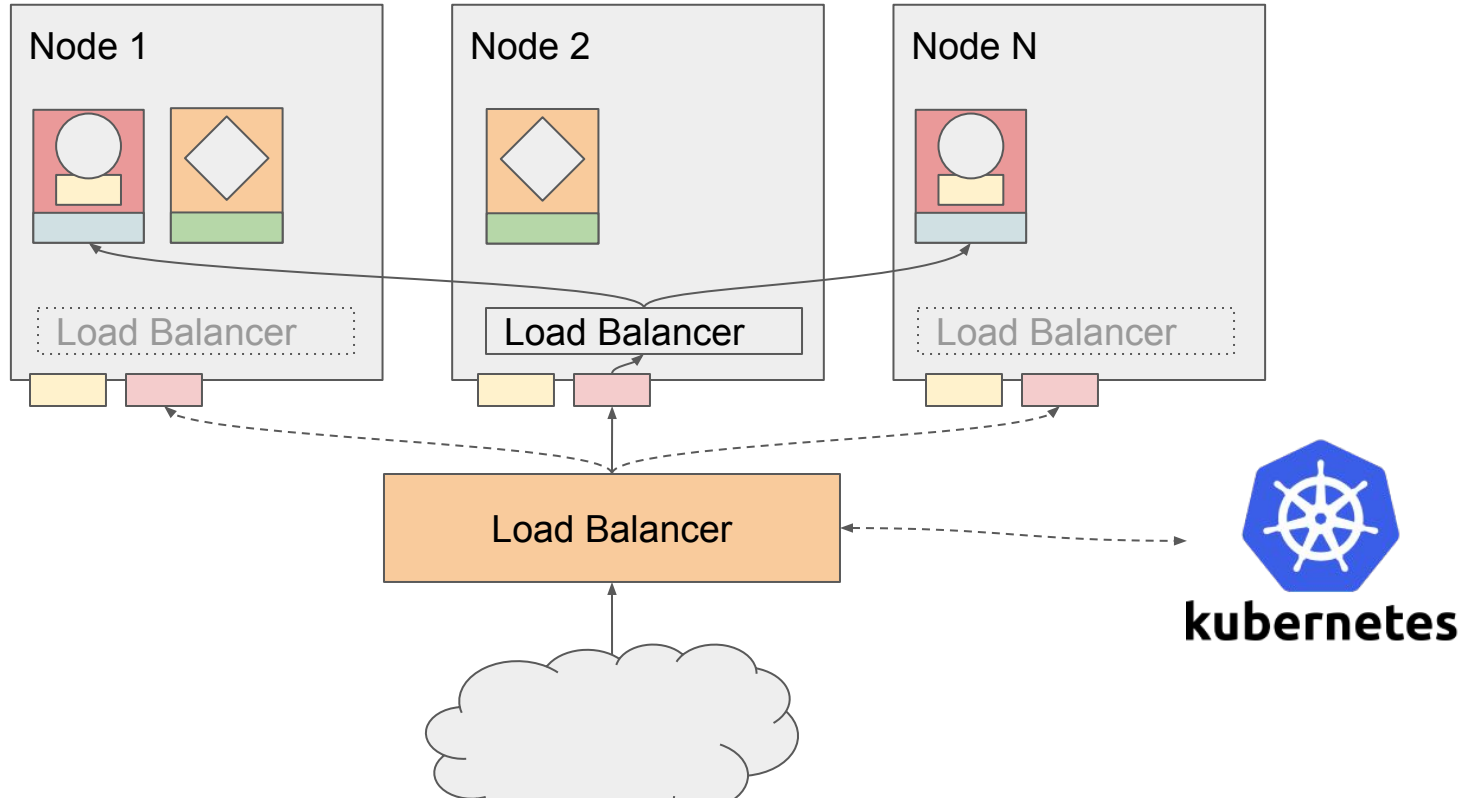
... and from the outside?



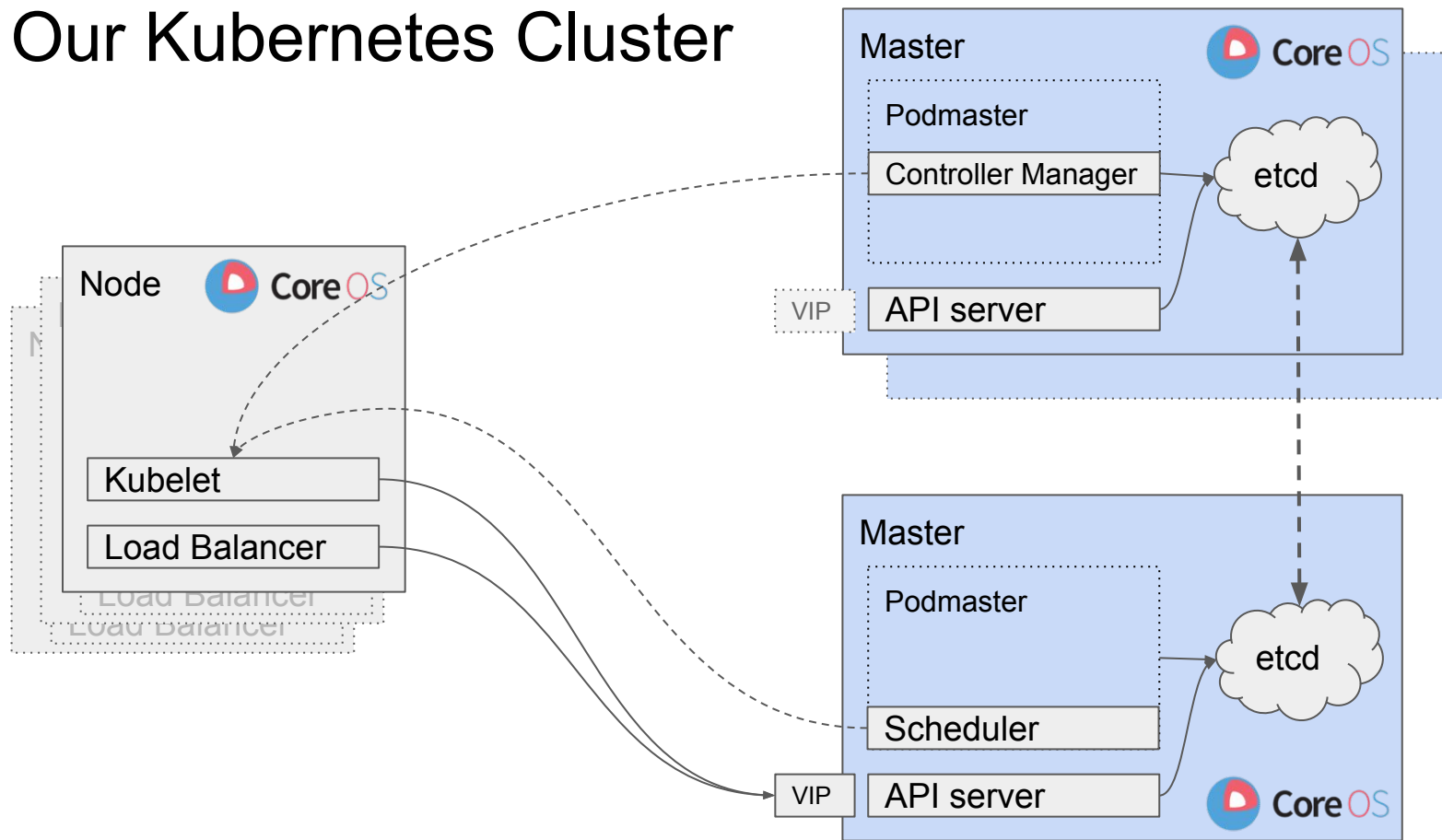
... and from the outside?



... and from the outside?



Our Kubernetes Cluster



Recap Kubernetes

- Schedule containers on machines
- Discovery of services using DNS and *automagic* load balancing
- High availability and no SPOF
- Great extensibility

Other features:

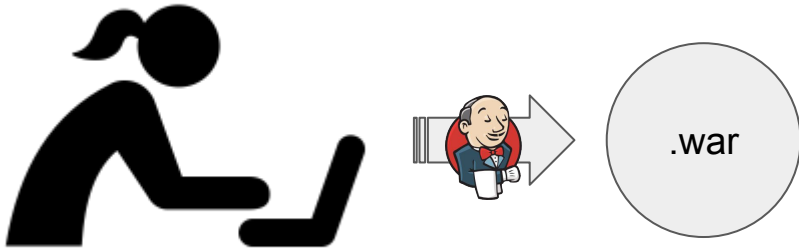
- Rolling updates
- Canary deployments



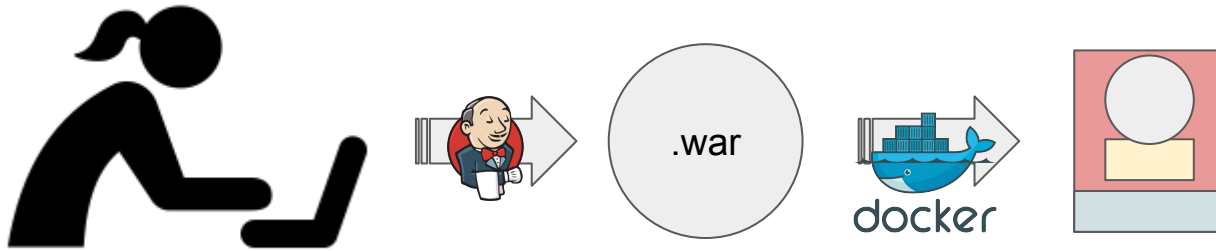
03

New deployment

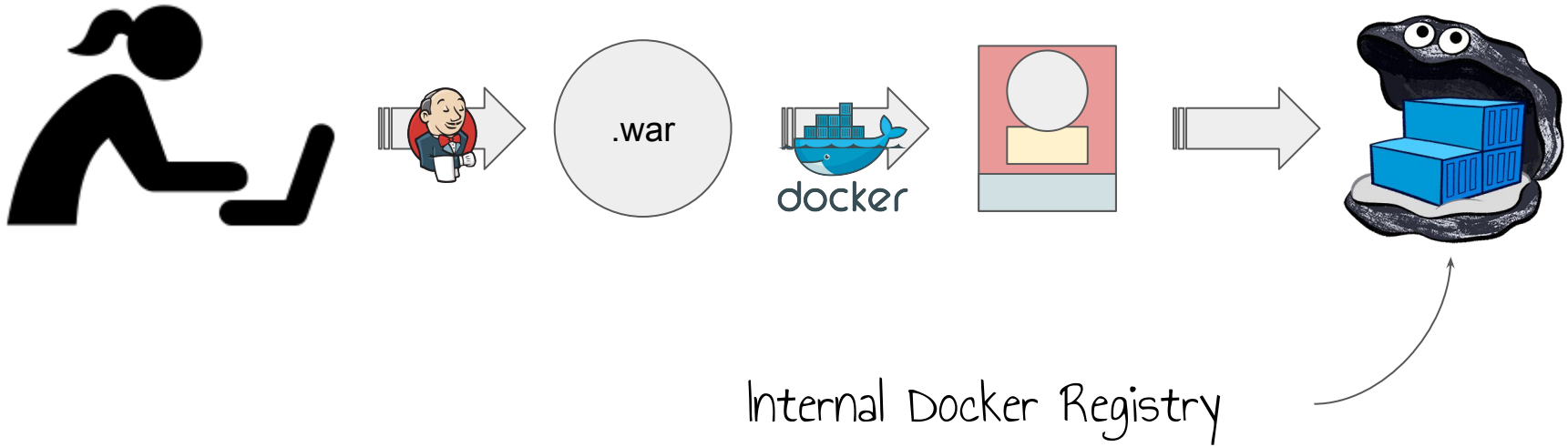
Build



Build

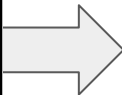


Build



Deploy

YAML definitions of
services



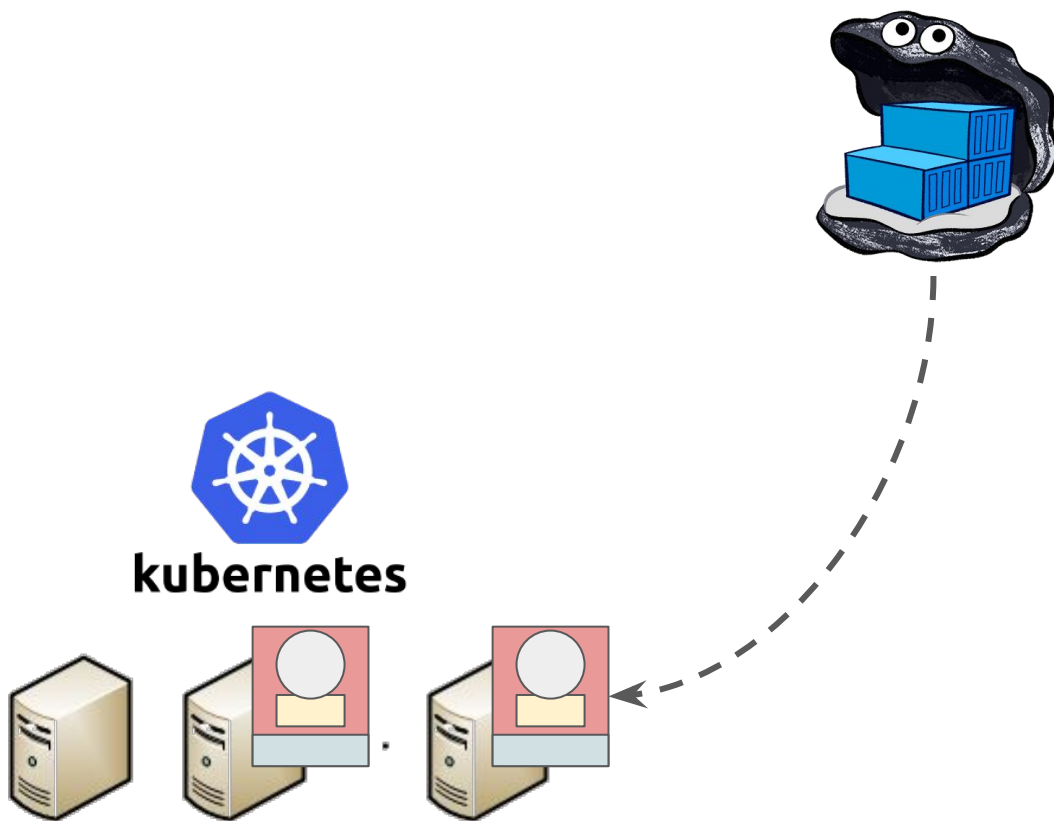
kubernetes

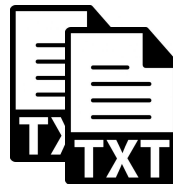
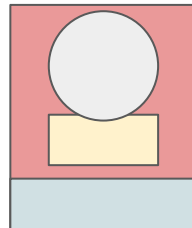
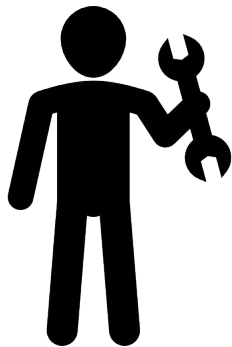


...



Deploy

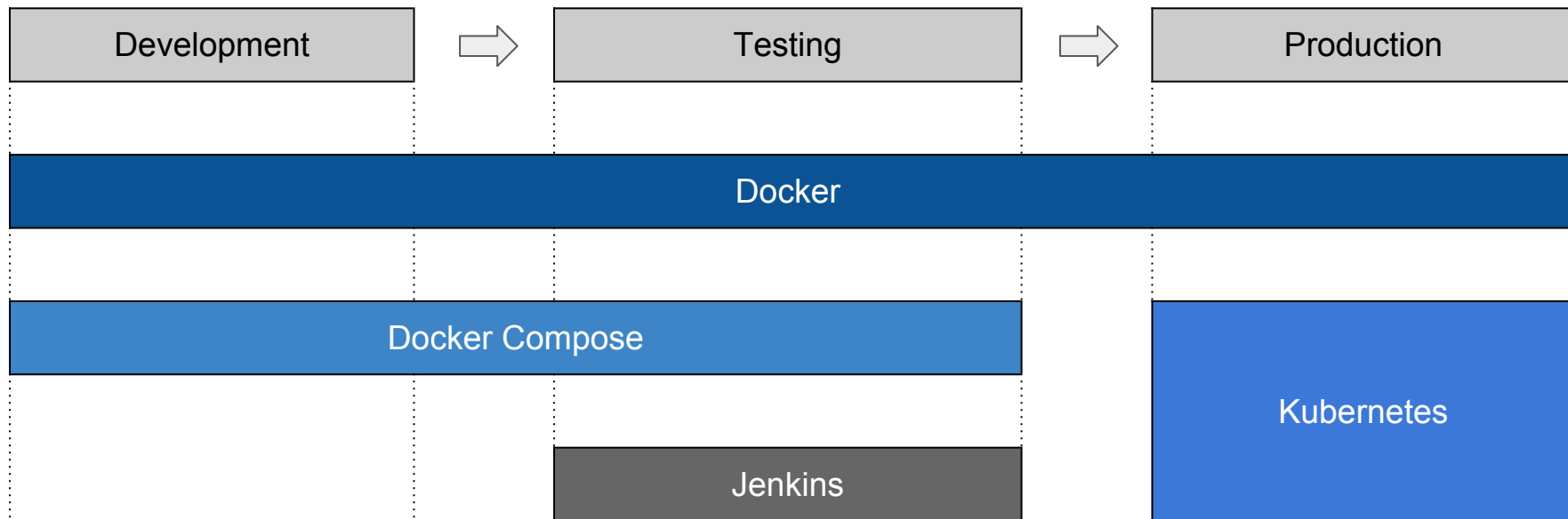




04

Summary

Global overview



Industrializing processes and re-using **the same tools for all projects**



Limitations

- Only stateless services by now
- A lot of people can do changes in the cluster
 - This is what we wanted, more agility!
 - But better audit logging is required!
- Not very polished configuration management
- Monitoring and alerting cannot be based on hostnames



Outcomes

- Faster time to deploy
- Clear responsibility division
 - Devs can deploy, run and monitor their services in production
 - SREs maintain infrastructure and tools **valid for all services**
- Better platform
 - Dynamic scaling
 - More customization per service (memory, cpu limits)
 - Homogeneous platform during Dev → CI → Production





github.com/tuenti

La compañía móvil que
se lo cuestiona todo