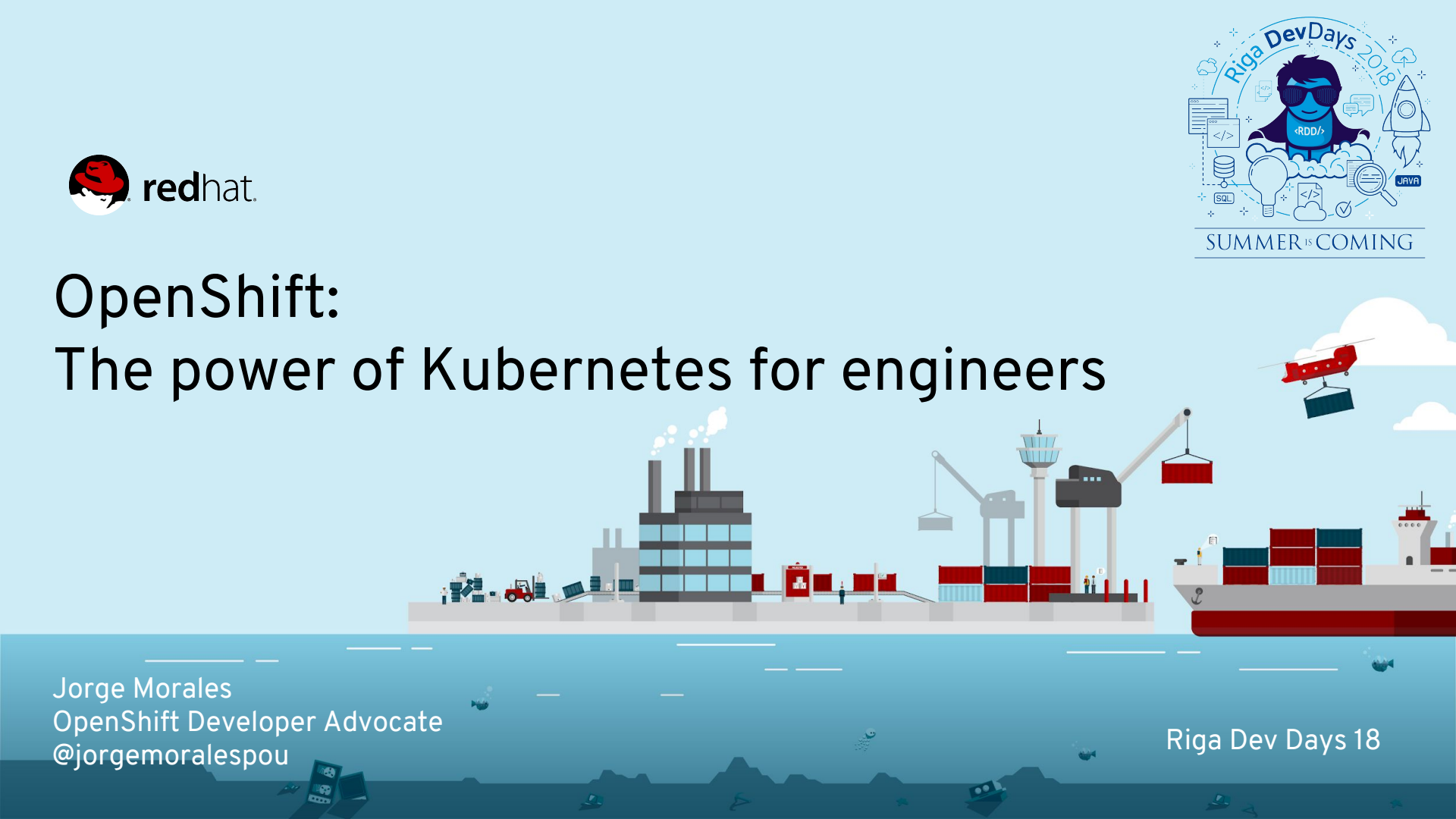



# OpenShift:
# The power of Kubernetes for engineers

Jorge Morales
OpenShift Developer Advocate
@jorgemoralespou

Riga Dev Days 18

# Me (aka Jorge)

- Spanish by nature and by language
- Work at Red Hat
- OpenShift Developer Advocate
- Mostly Java developer
- Obsessed with improving development experience

http://jorgemoral.es

@jorgemoralespou

github.com/jorgemoralespou

# Agenda

**09:30 – 11:00 Introduction to OpenShift**

11:00 – 11:30 Refreshment/Bio Break

**11:30 – 13:00 Hands on labs - part 1**

13:00 – 14:00 Lunch

**14:00 – 15:30 Hands on labs - part 2**

15:30 – 16:00 Refreshment/Bio Break

**16:00 – 17:00 Hands on labs - part 3**

**17:00 – 17:30 Recap**

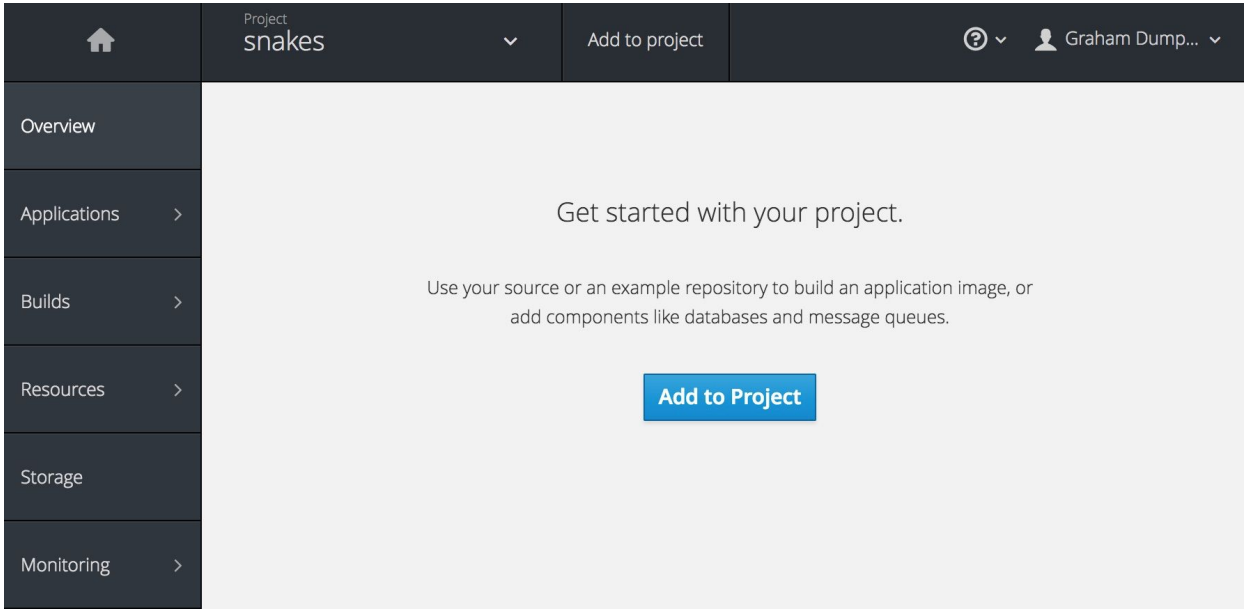
# INTRODUCTION TO OPENSHIFT

# Goals

1. Quick Introduction to Containers, Kubernetes, and OpenShift
2. Show You the Power of it All in Action!

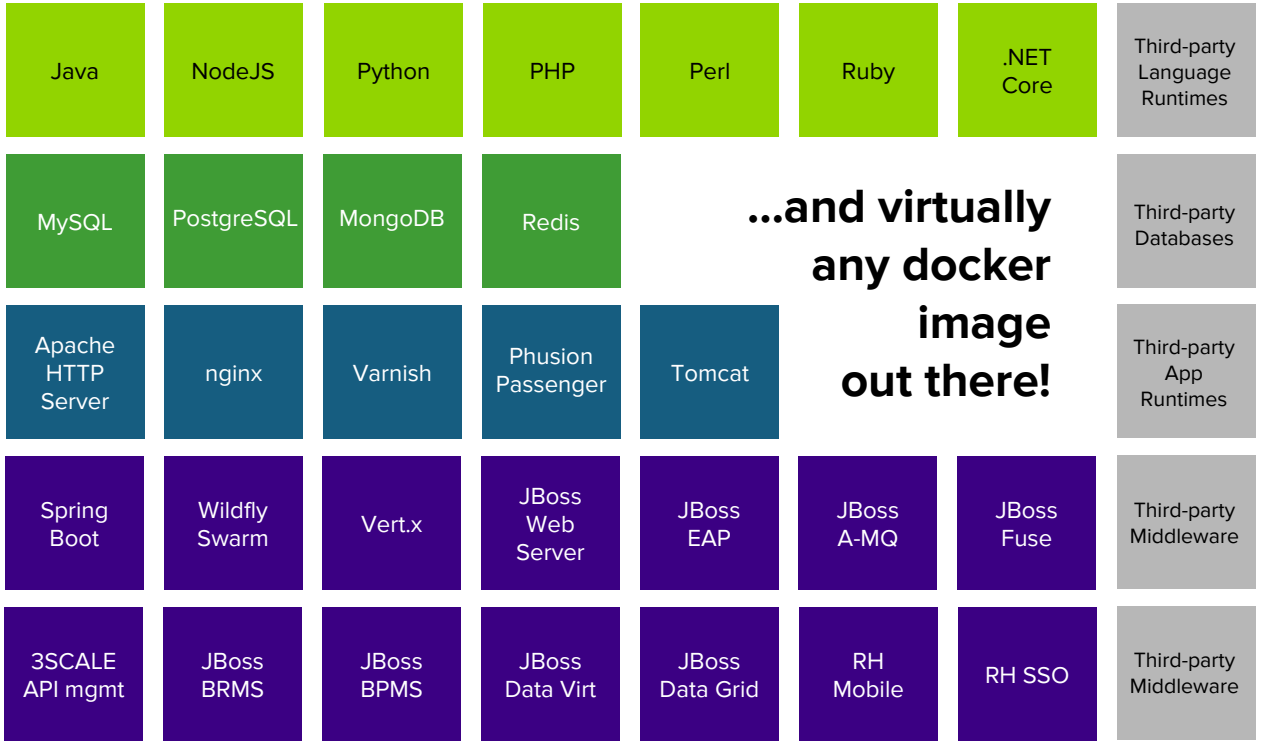
# What is it for?

# QUICK DEMO




redhat.

# TRUE POLYGLOT PLATFORM




**CrunchyData**

**GitLab**

**Iron.io**

**Couchbase**

**Sonatype**

**EnterpriseDB**

and many more

redhat.

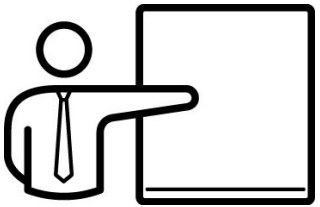
# But why?

BUSINESS

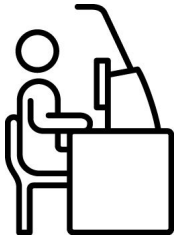
TECHNOLOGY

PRODUCT OWNER

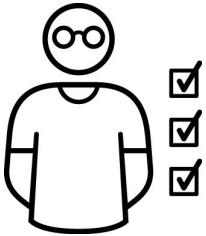
BUSINESS ANALYST



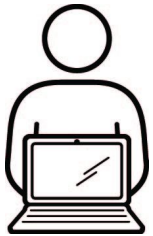
DEVELOPER



TESTER



OPERATIONS

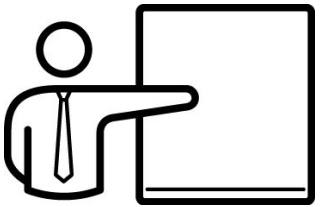


BUSINESS

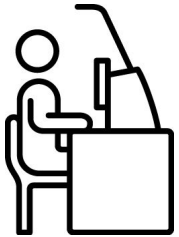
TECHNOLOGY

PRODUCT OWNER

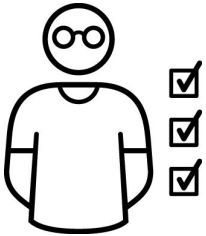
BUSINESS ANALYST



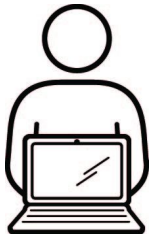
DEVELOPER



TESTER



OPERATIONS

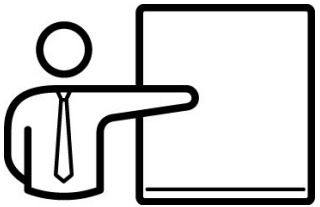


BUSINESS

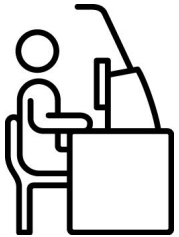
TECHNOLOGY

PRODUCT OWNER

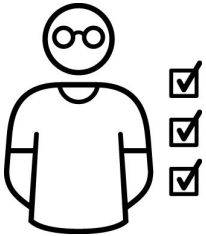
BUSINESS ANALYST



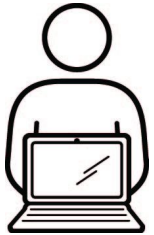
DEVELOPER



TESTER

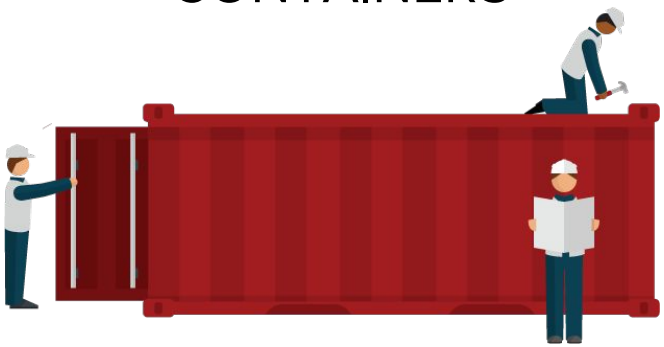


OPERATIONS



# A DRIVER

CONTAINERS



Adopting a container-based strategy helps applications to be easily shared and deployed.

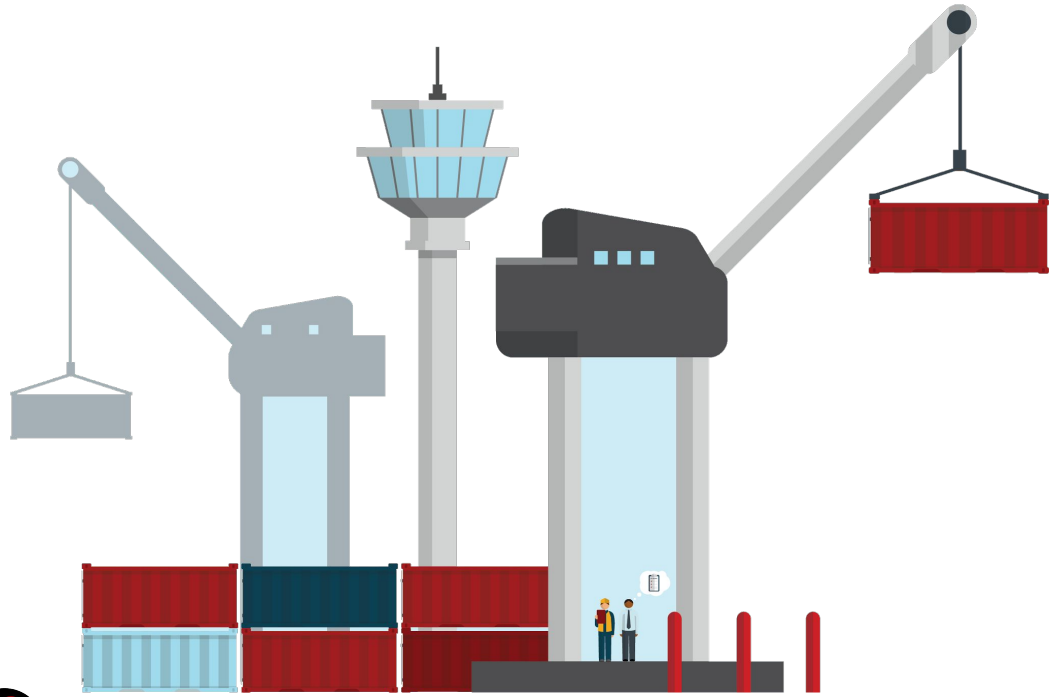
# APPLICATIONS REQUIRE **MANY** CONTAINERS.

## HOW DO YOU DELIVER AND MANAGE THEM AT SCALE?

redhat.

# BUILD, DEPLOY, AND MANAGE AT SCALE

## WITH RED HAT OPENSHIFT CONTAINER PLATFORM



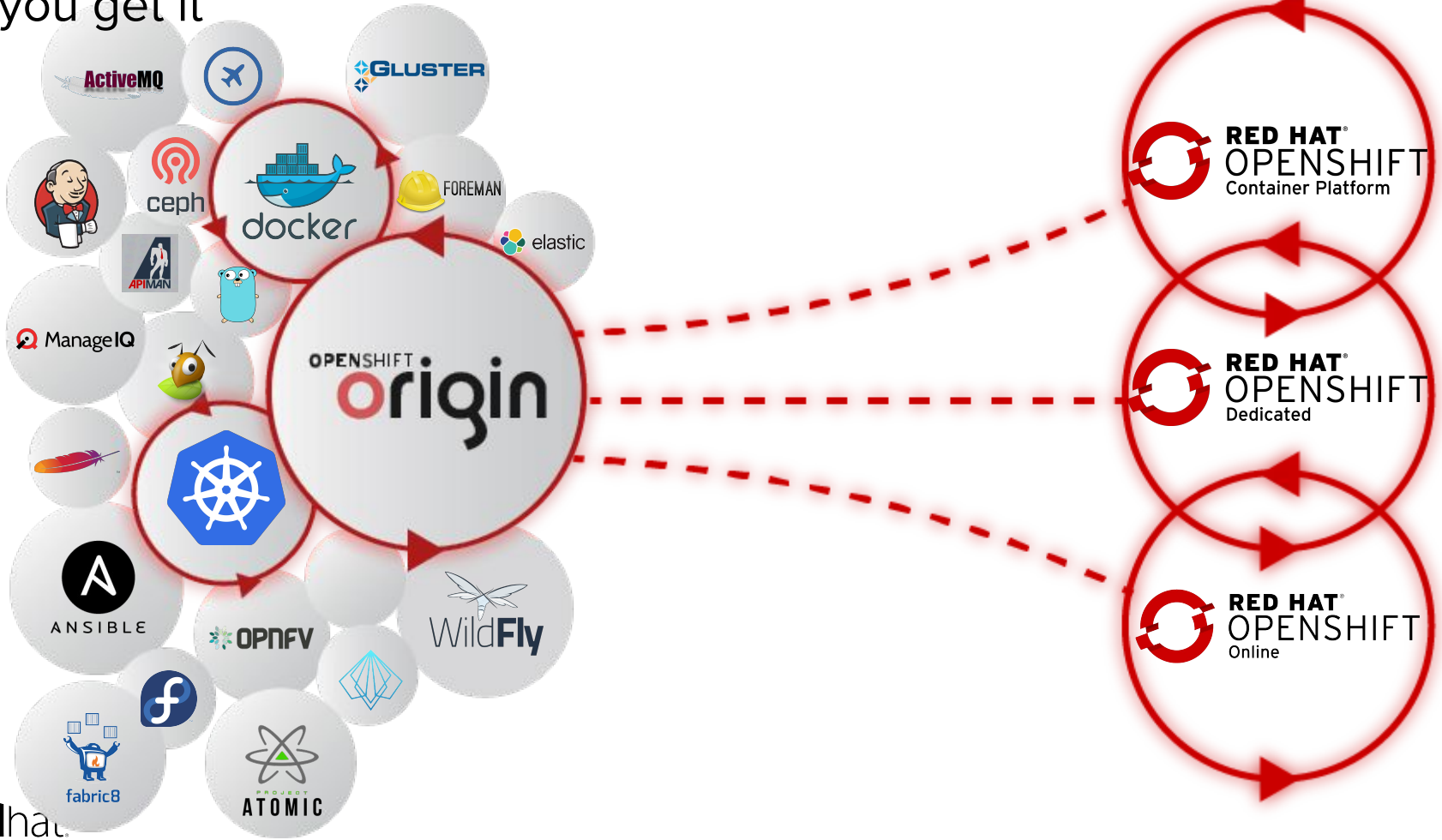
The industry's most secure and comprehensive enterprise-grade container platform based on industry standards, Docker and Kubernetes.

RED HAT® OPENSHIFT Container Platform

redhat.

# What is it built from?

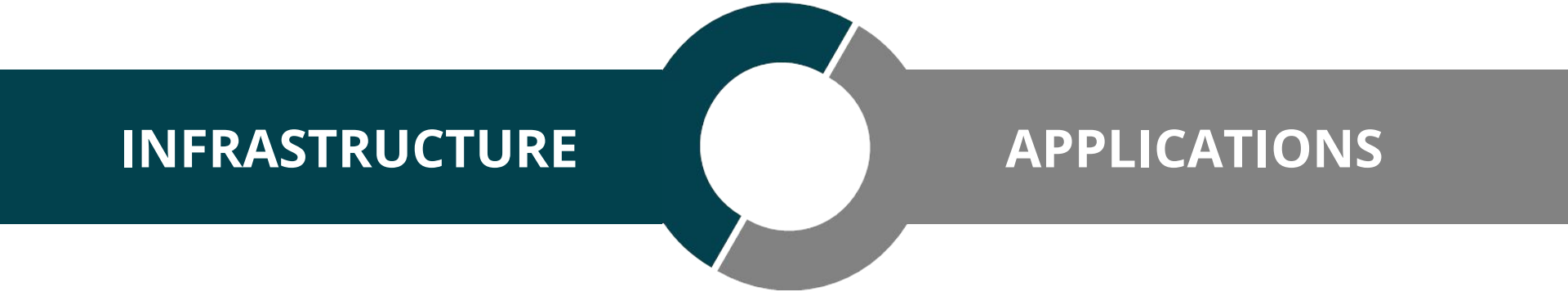
# How you get it



# Why containers?

# WHAT ARE CONTAINERS?

It Depends Who You Ask



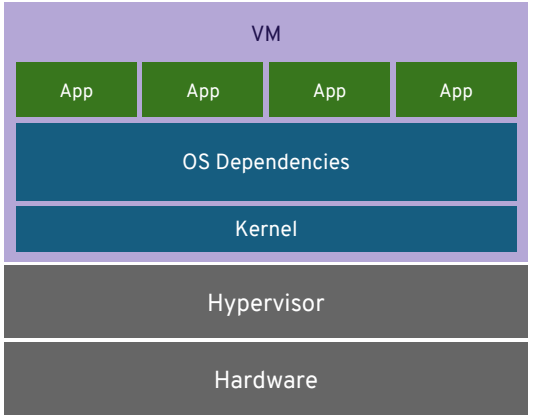

- Sandboxed application processes on a shared Linux OS kernel
- Simpler, lighter, and denser than virtual machines
- Portable across different environments

- Package my application and all of its dependencies
- Deploy to any environment in seconds and enable CI/CD
- Easily access and share containerized components

redhat.

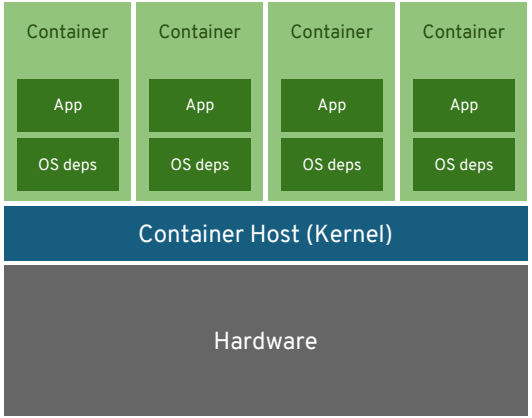
# VMs vs Containers

VIRTUAL MACHINES




virtual machines are isolated
apps are not

CONTAINERS




containers are isolated
so are the apps

redhat.

https://github.com/openshift-evangelists/workshop-summit/blob/master/nginx/Dockerfile

```
FROM centos

ADD nginx.repo /etc/yum.repos.d/

RUN yum update -y && yum install -y nginx

RUN mkdir -p /nginx
ADD nginx.sh /nginx/
RUN chmod 777 -R /nginx

EXPOSE 8080
WORKDIR /nginx

VOLUME ["/nginx/html"]
VOLUME ["/nginx/logs"]

CMD ["./nginx.sh"]
```

```
$ docker build -t app:v1 .

$ docker run app:v1
```

# Orchestration, Scheduling, Clustering, Shared Storage are All Still Needed

kubernetes

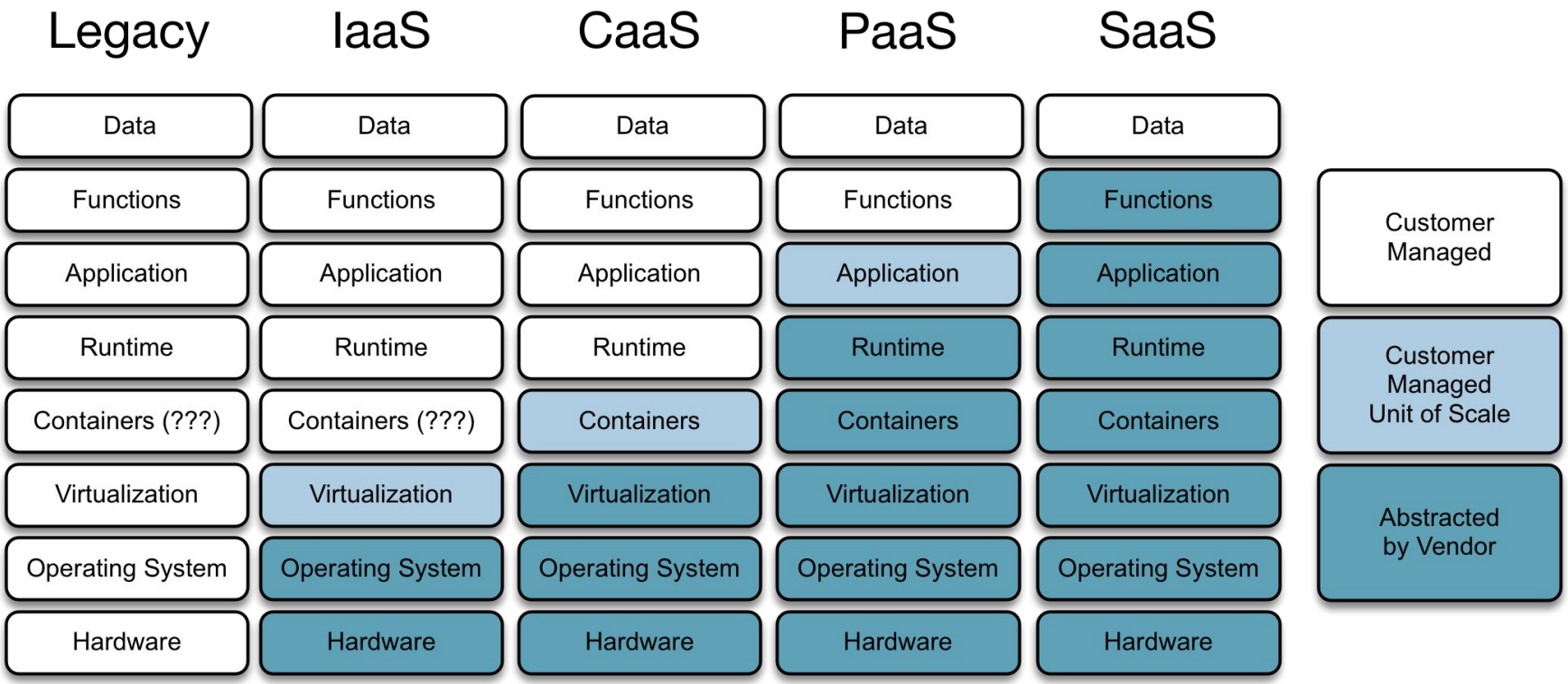
Kubernetes is an open-source platform designed to automate deploying, scaling, and operating application containers.

redhat

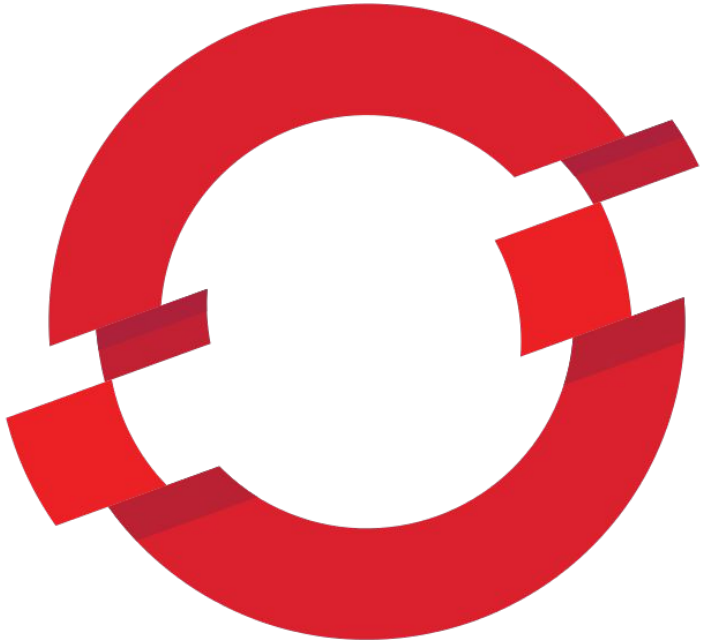
With Kubernetes, you are able to quickly and efficiently respond to customer demand:

- Deploy your applications quickly and predictably.
- Scale your applications on the fly.
- Roll out new features seamlessly.
- Limit hardware usage to required resources only.

# Need to provide a good experience for developers and not just IT operations

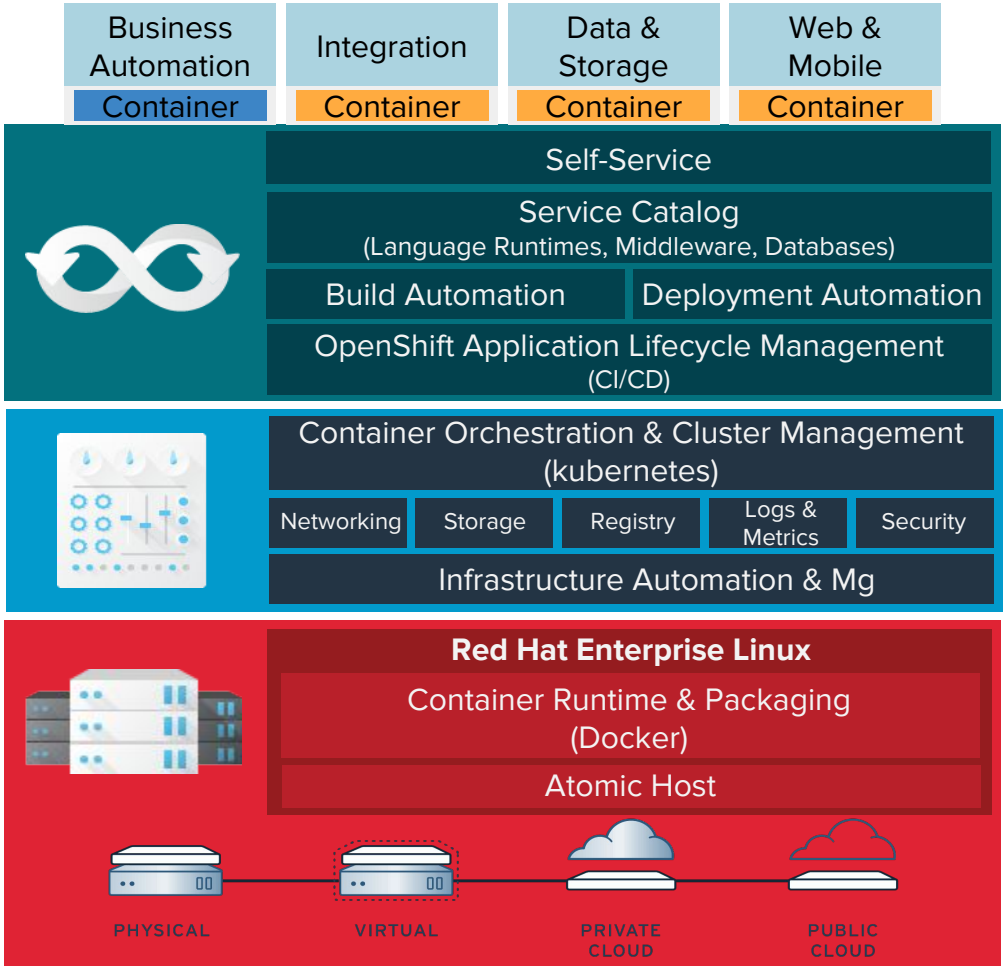

Legacy
IaaS
CaaS
PaaS
SaaS
Data
Functions
Application
Runtime
Containers (???)
Containers
Virtualization
Operating System
Hardware
Customer Managed
Customer Managed Unit of Scale
Abstracted by Vendor

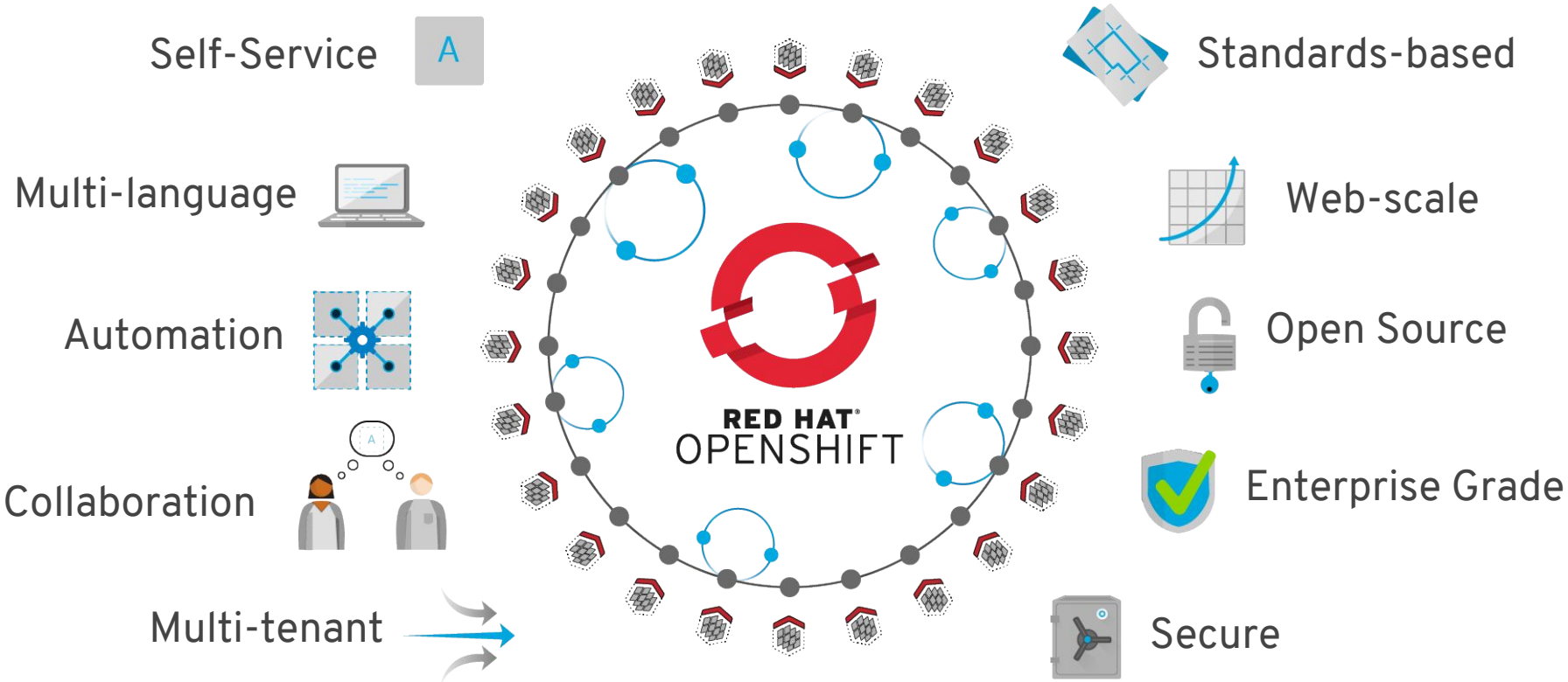
redhat.



# OPENSHIFT

redhat.

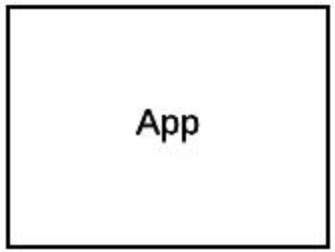

Business Automation
Container
Integration
Container
Data & Storage
Container
Web & Mobile
Container
Self-Service
Service Catalog
(Language Runtimes, Middleware, Databases)
Build Automation
Deployment Automation
OpenShift Application Lifecycle Management
(CI/CD)
Container Orchestration & Cluster Management
(kubernetes)
Networking
Storage
Registry
Logs & Metrics
Security
Infrastructure Automation & Mg
Red Hat Enterprise Linux
Container Runtime & Packaging
(Docker)
Atomic Host
PHYSICAL
VIRTUAL
PRIVATE CLOUD
PUBLIC CLOUD

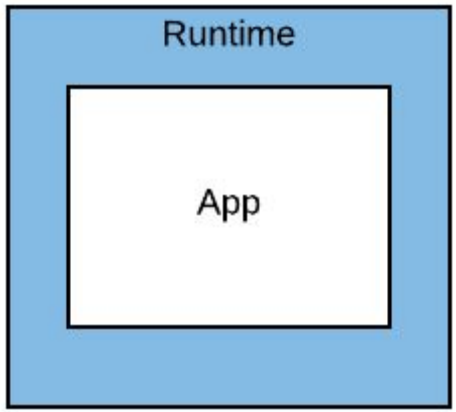

Self-Service
Multi-language
Automation
Collaboration
Multi-tenant
RED HAT
OPENSHIFT
Standards-based
Web-scale
Open Source
Enterprise Grade
Secure

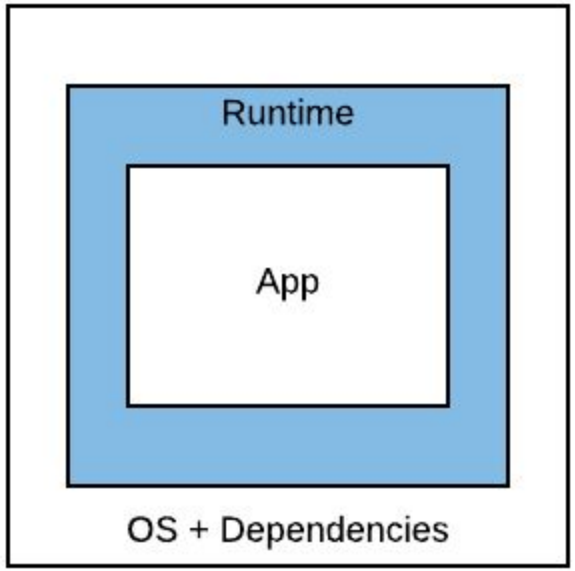
# So, why OpenShift

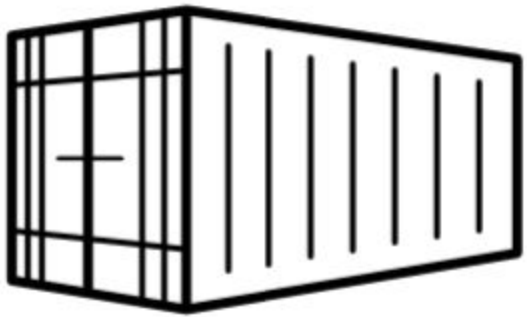
1. You want more out of containers
2. You want more out of Kubernetes
3. You are smart, focusing on the things you enjoy doing

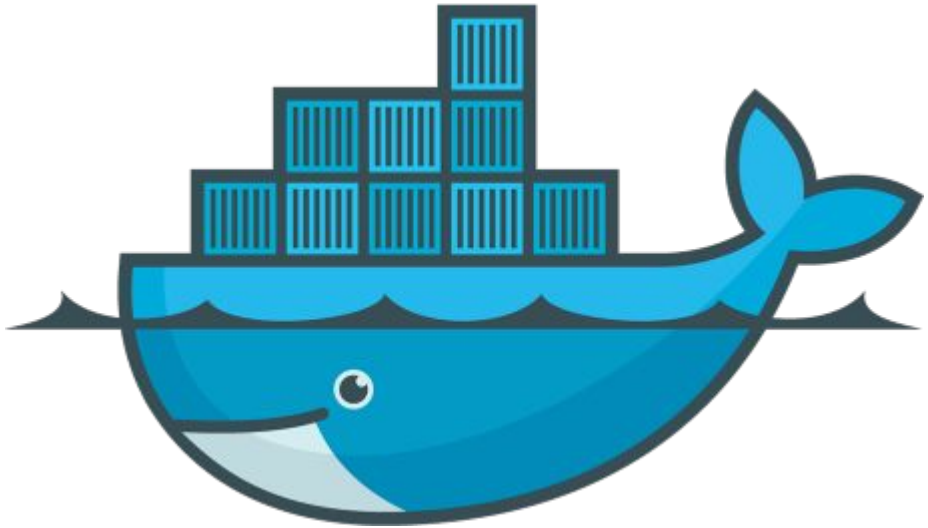
# How it works

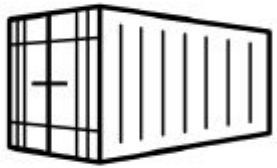

App


Runtime
App

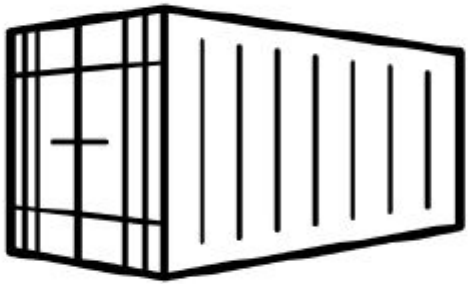

Runtime
App
OS + Dependencies





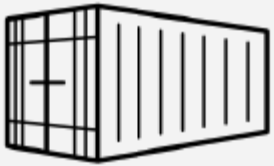


**1 cpu**
**512 MB**

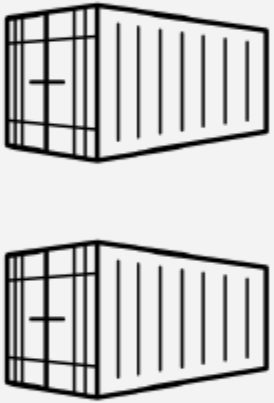


2 cpu
1 GB

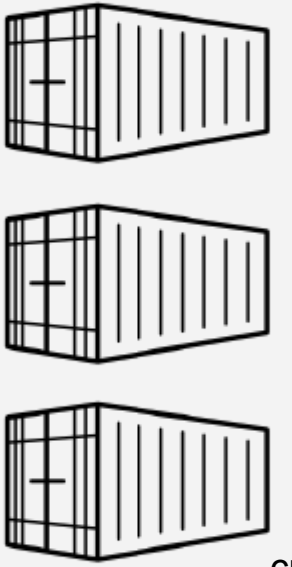
Pod



cpu=
mem=
ENV=
...

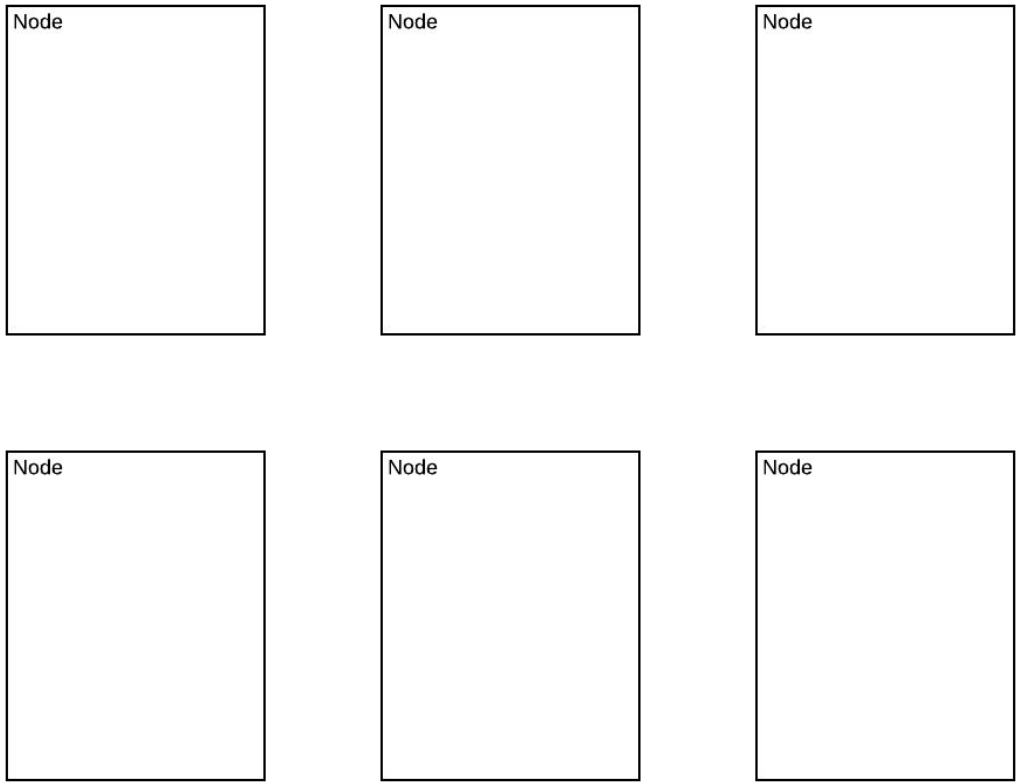


cpu=
mem=
ENV=
...

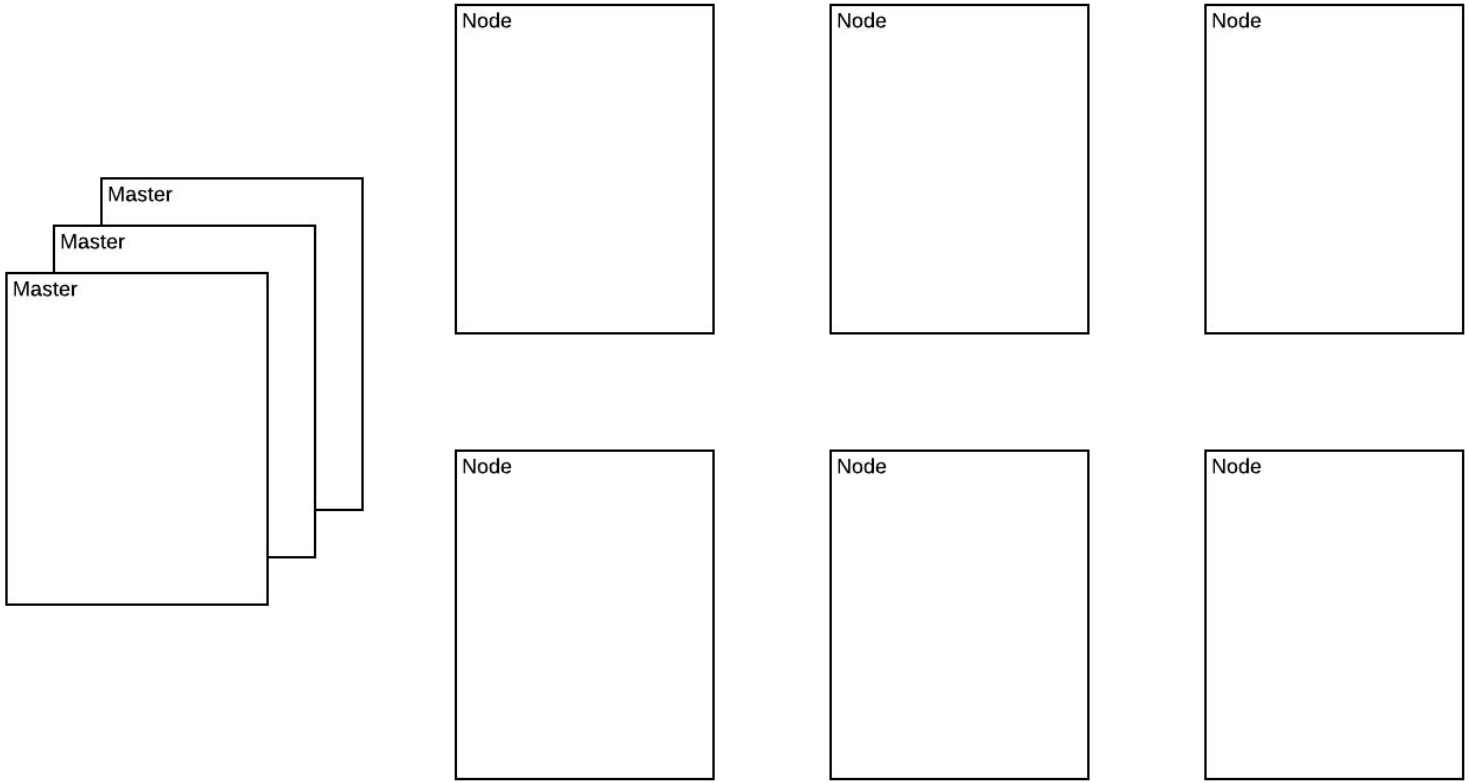


cpu=
mem=
ENV=
...

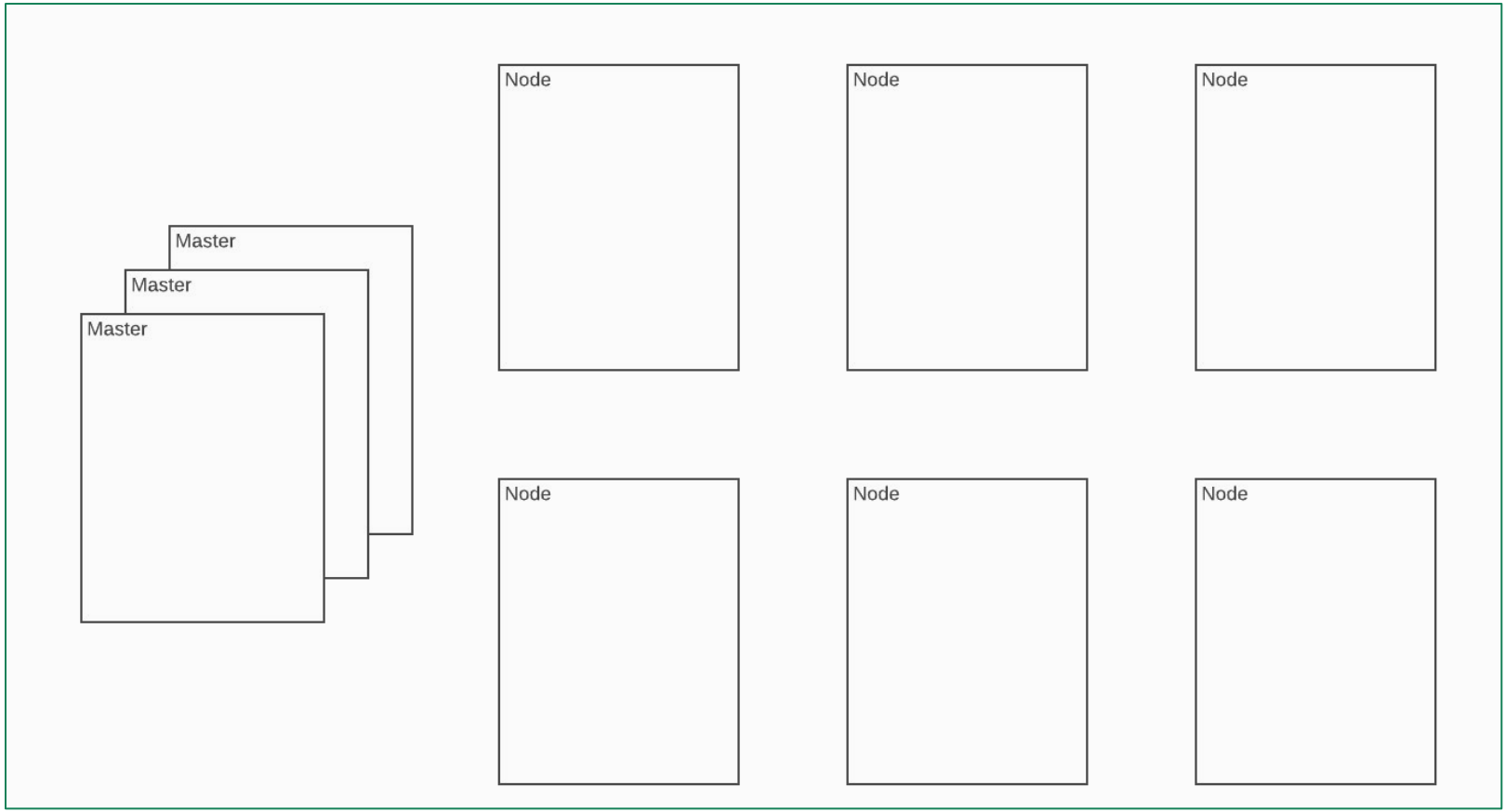
# Nodes



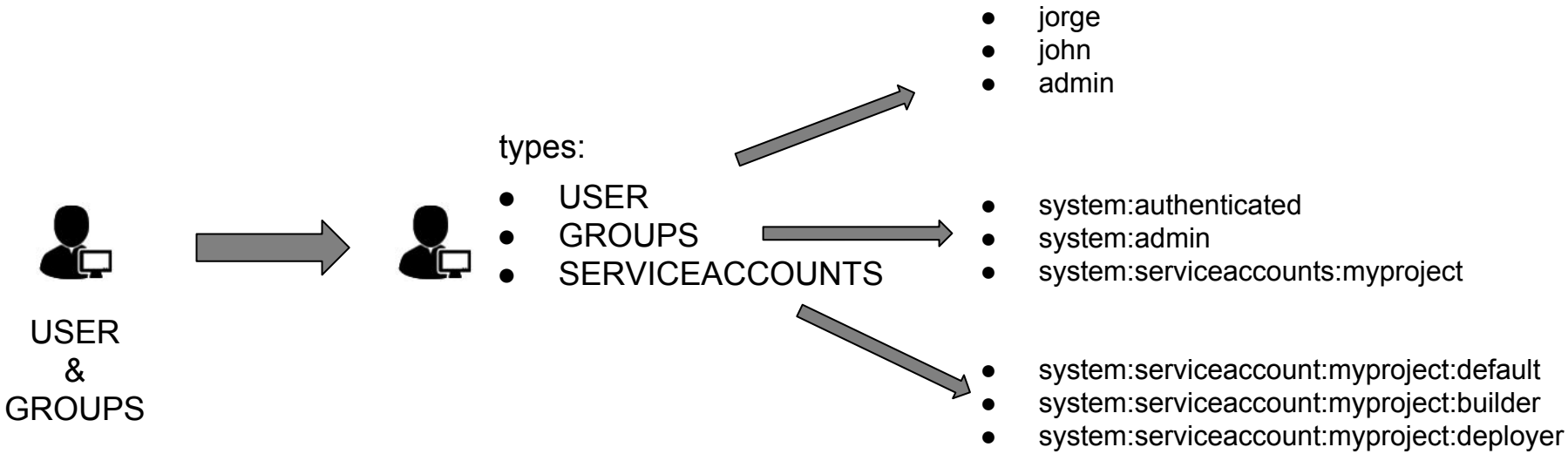
# Master nodes



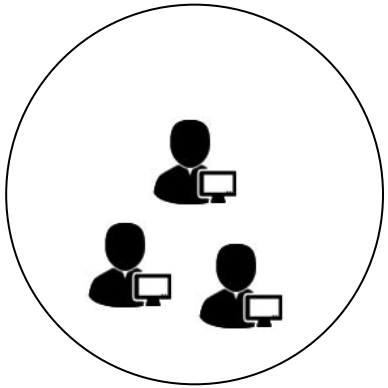
OpenShift Cluster



Users and Groups



Scopes / Multitenancy



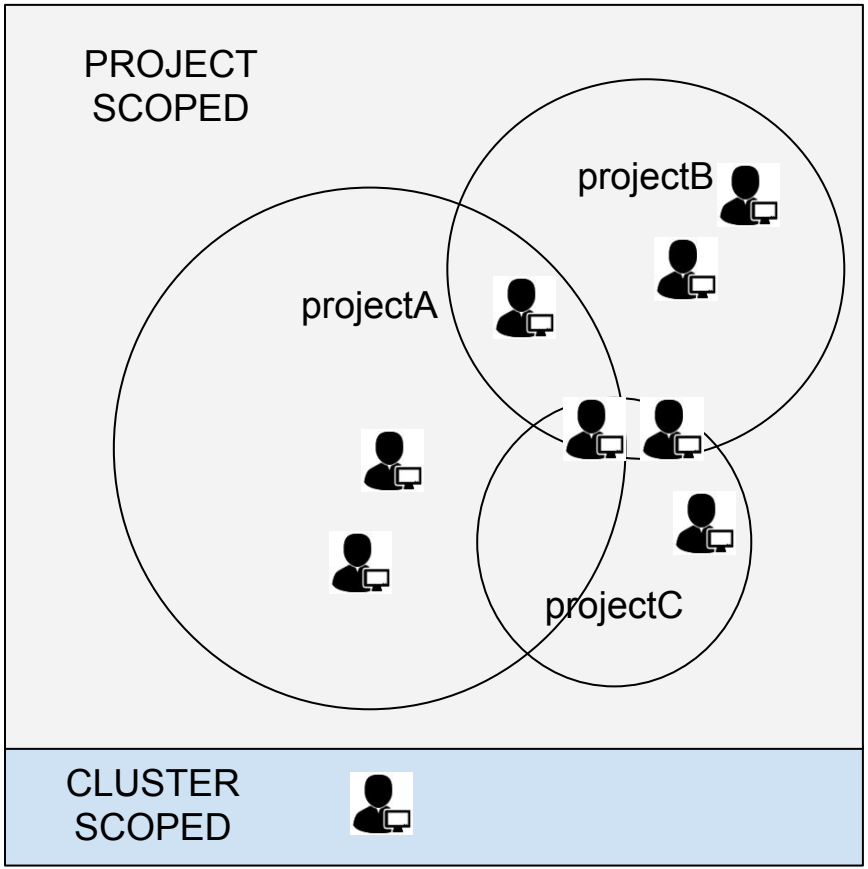
PROJECT (a.k.a namespace)

Constrains:

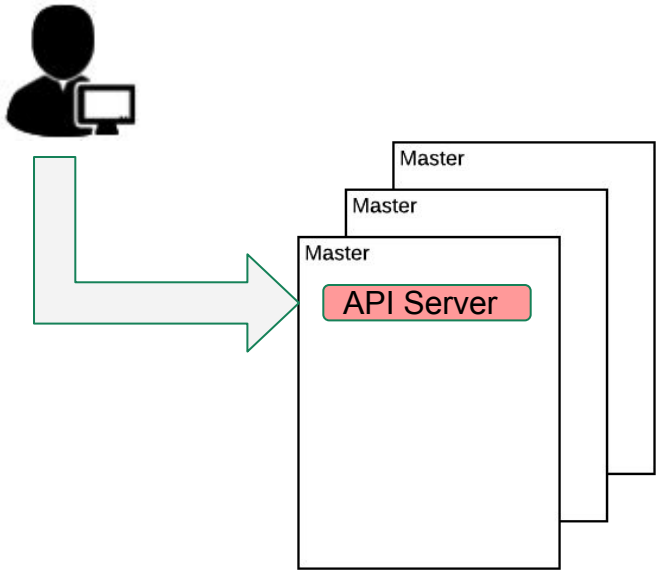
- Resources
- Visibility

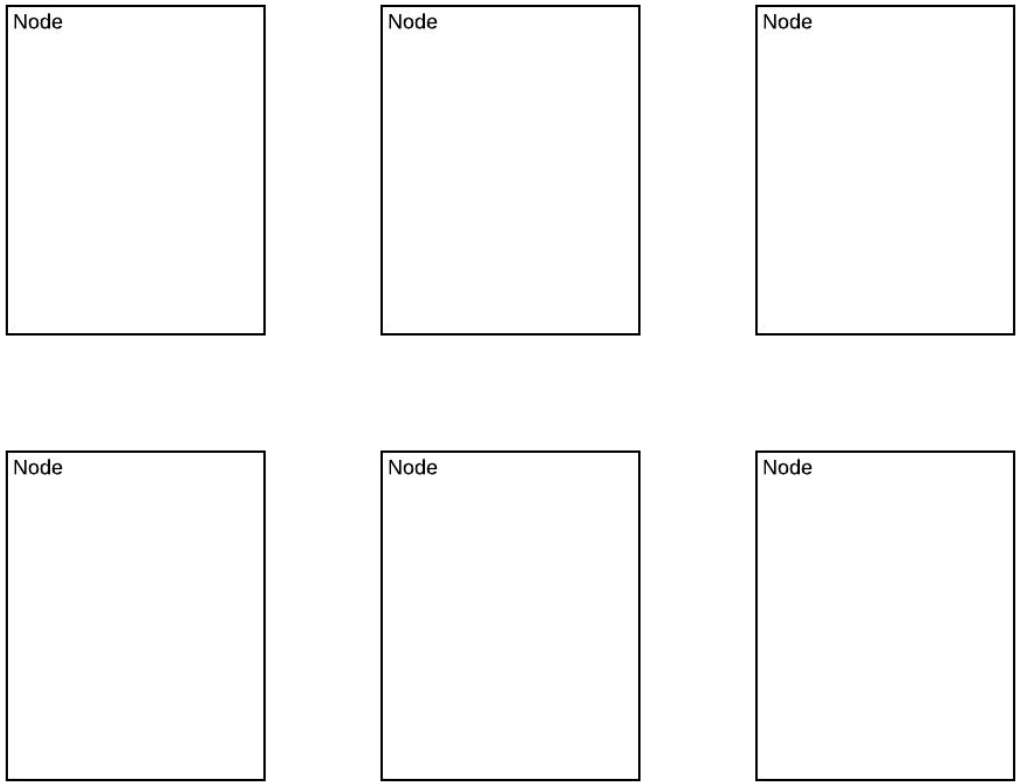
View project

View all

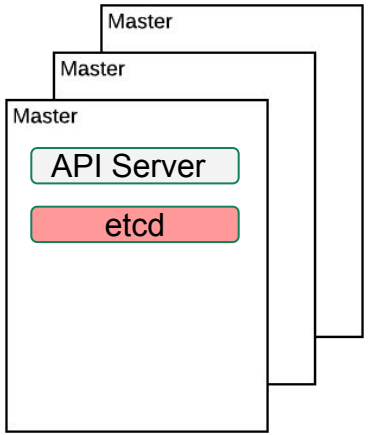


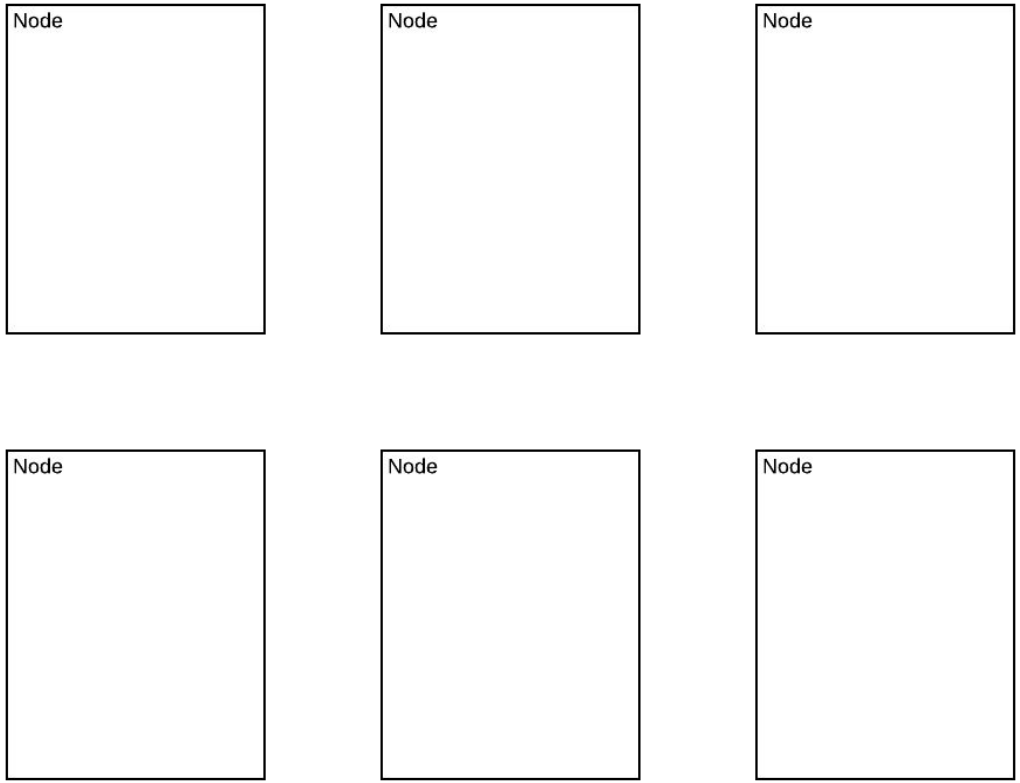
# Authn & Authz

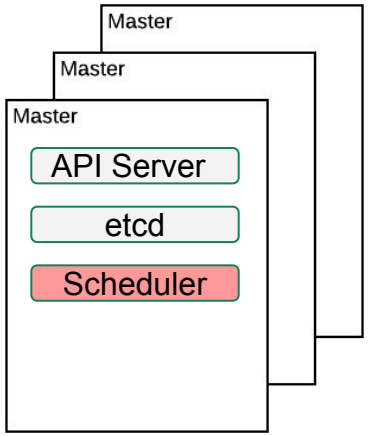

Master
Master
Master
API Server


Node
Node
Node
Node
Node
Node

Store state


Master
Master
Master
API Server
etcd


Node
Node
Node
Node
Node
Node




Node

Node

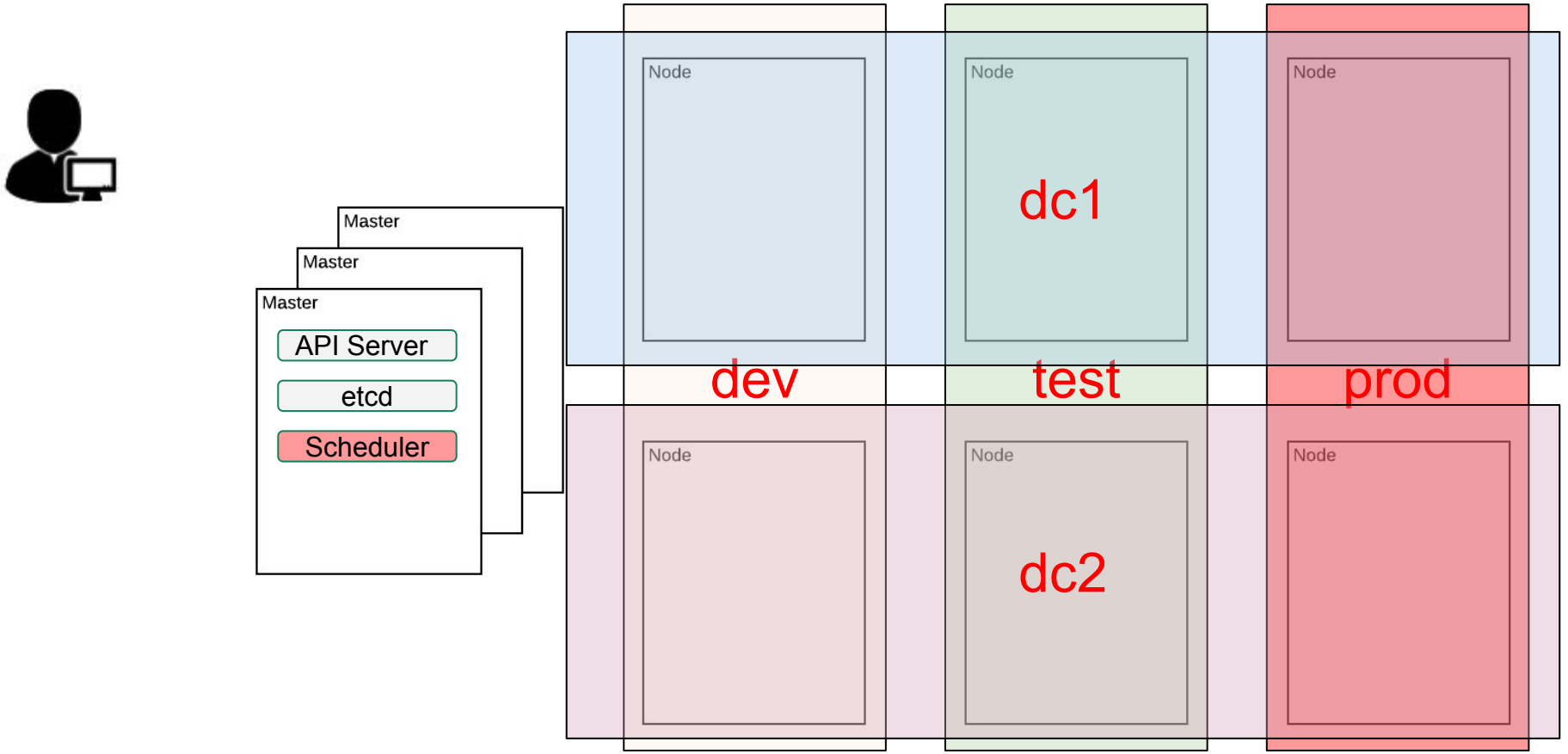
Node

Node

Node

Node

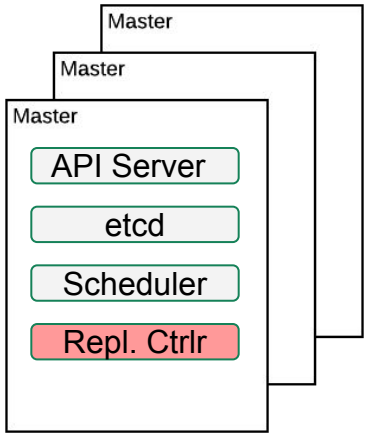
Workload placement

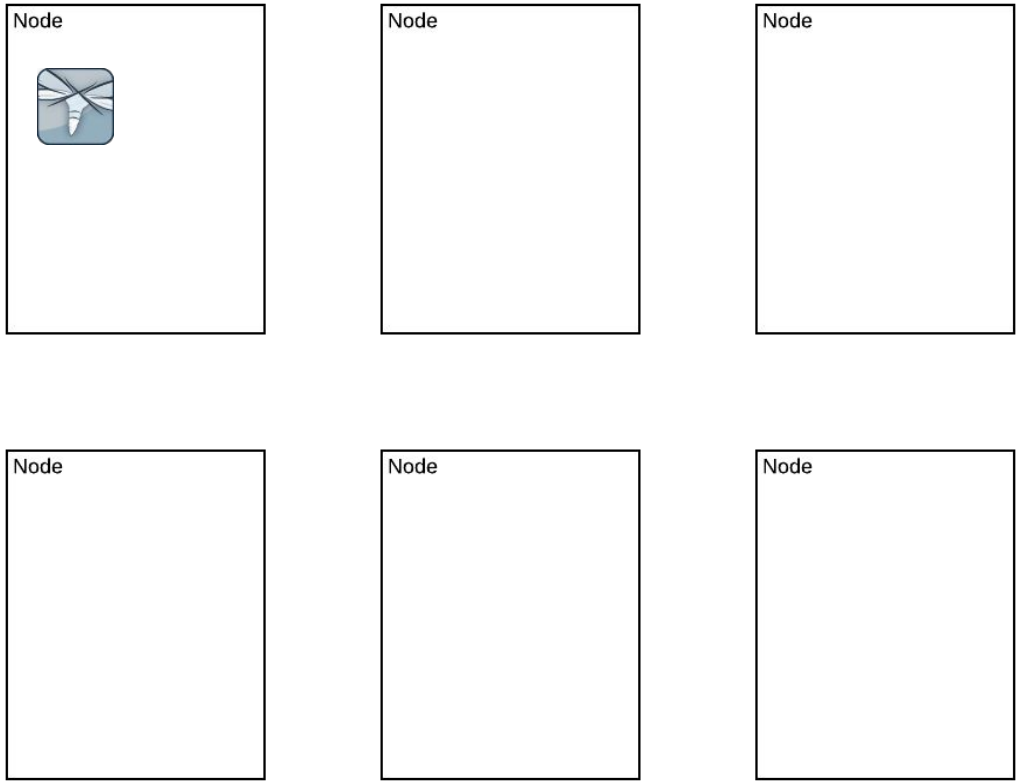


# Workloads

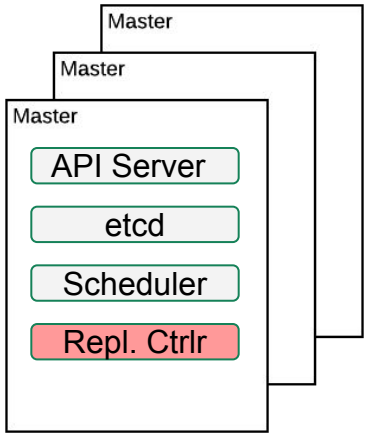
- Stateless
  - **ReplicaSet** (also known as ReplicationController)
- Stateful
  - **StatefulSet** (previously known as PetSet)
- Batch
  - **Jobs**
  - **CronJobs**
- Daemon
  - **DaemonSet**

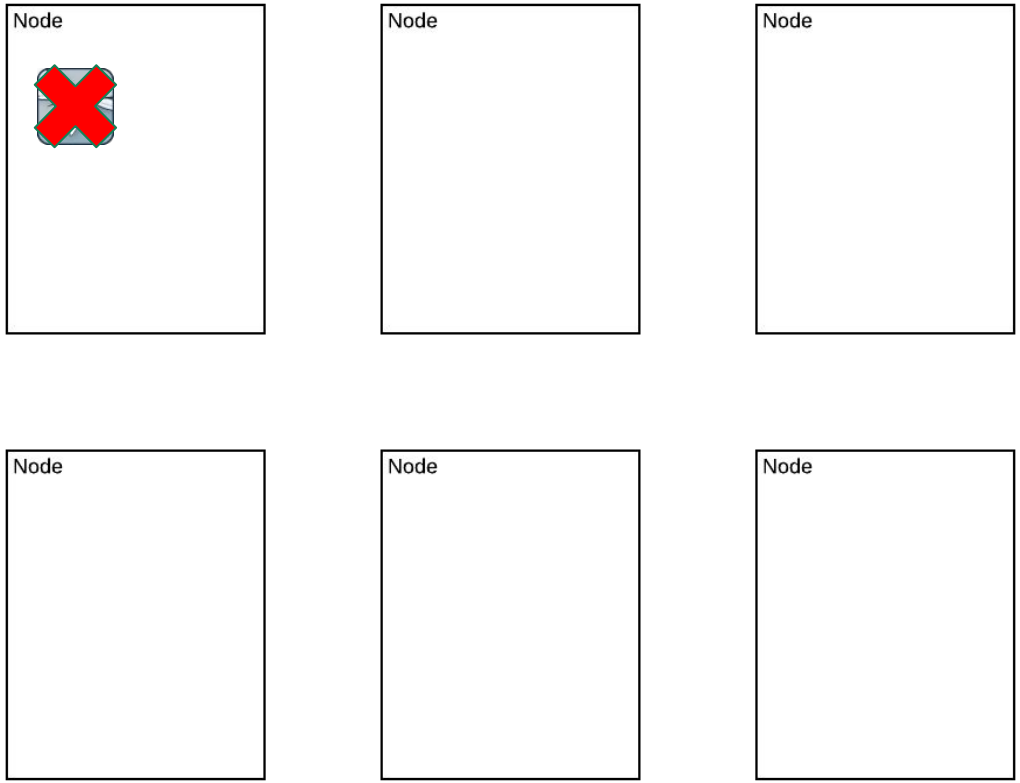
# Workload execution guarantees



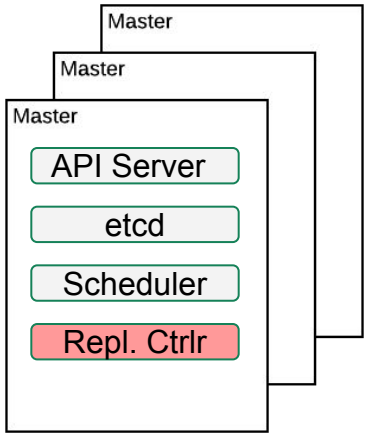



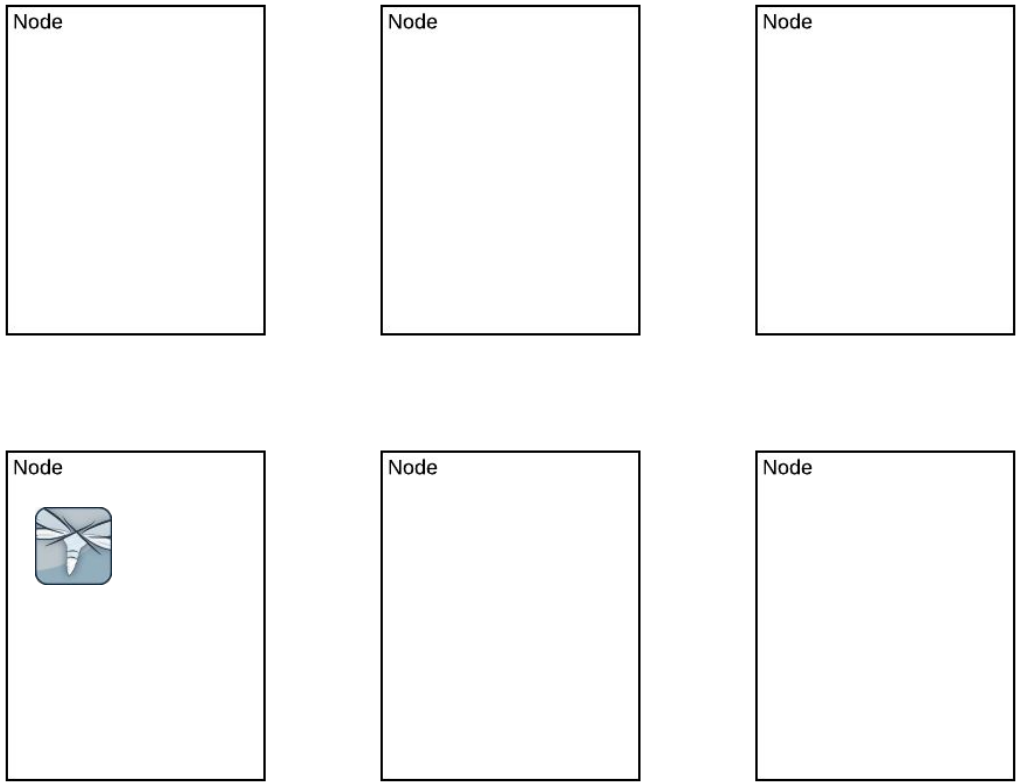
# Workload execution guarantees



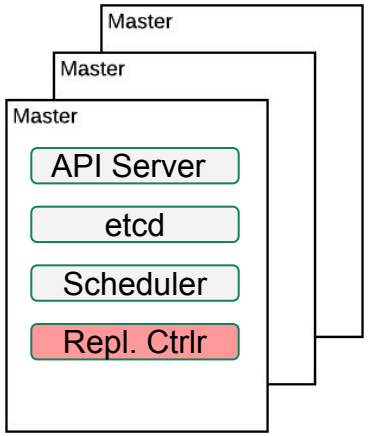



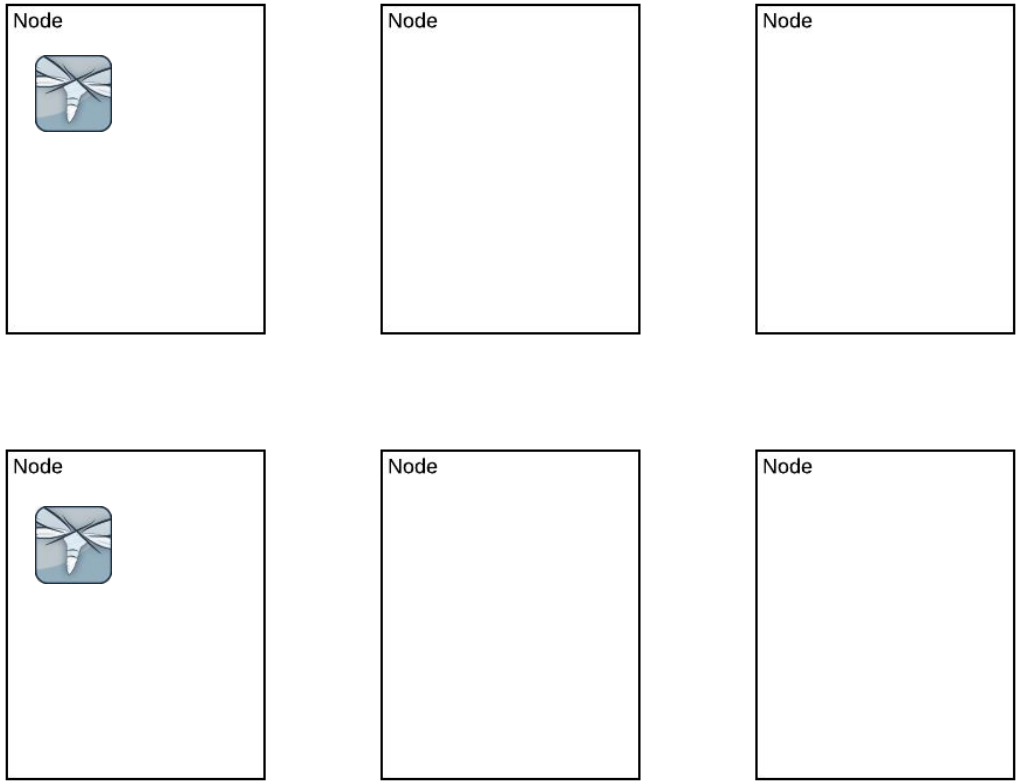
# Workload execution guarantees



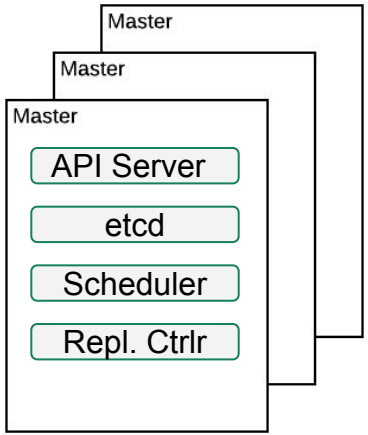



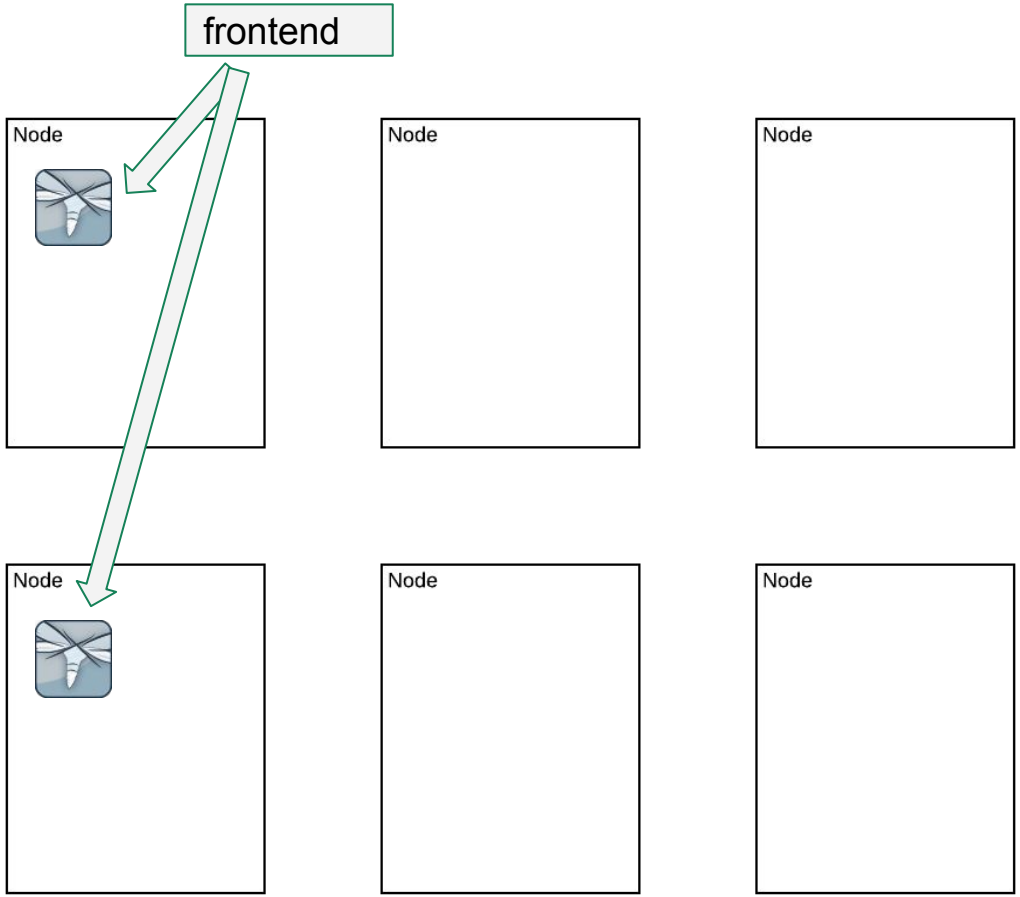
# Scalability



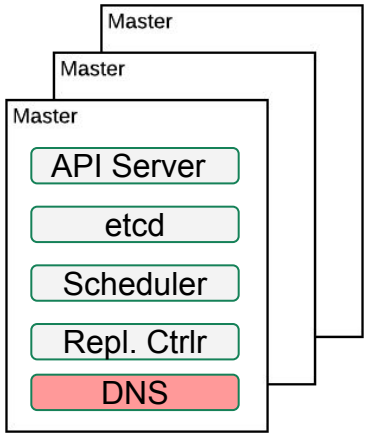



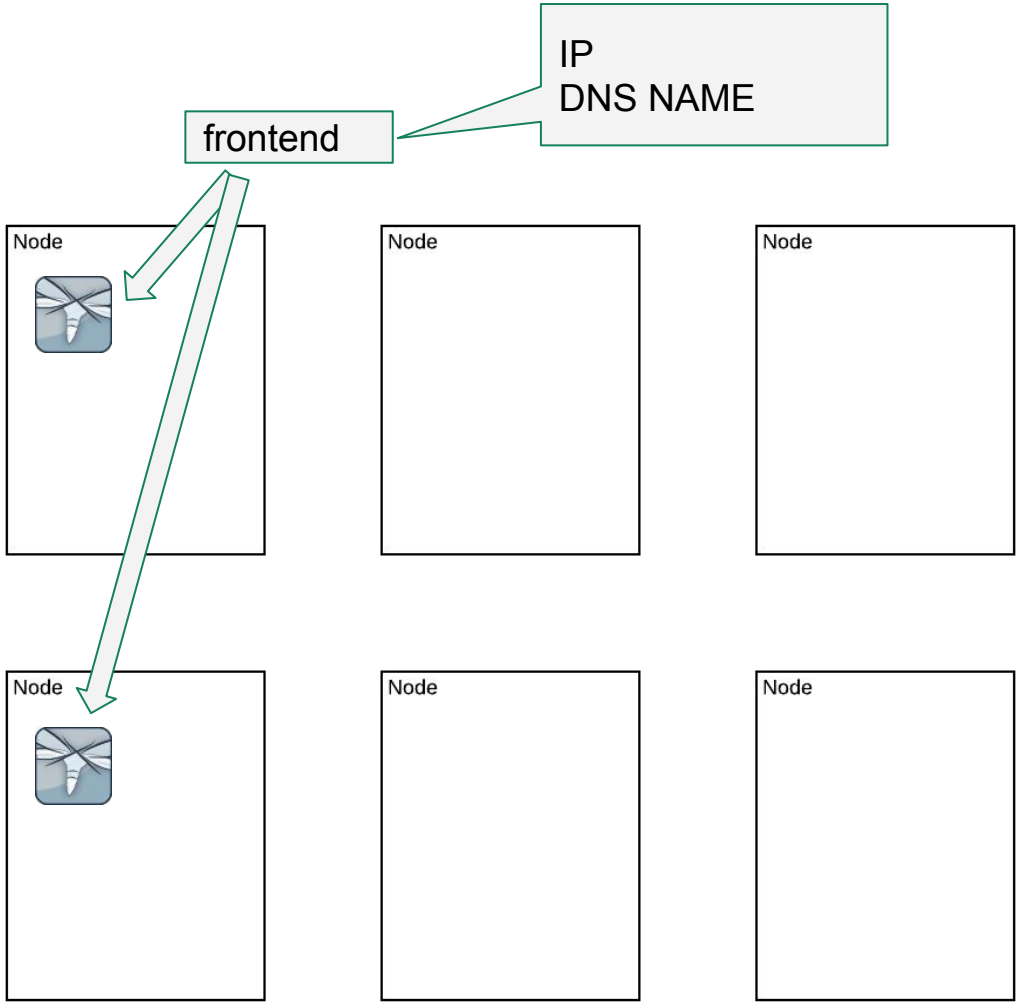
Service

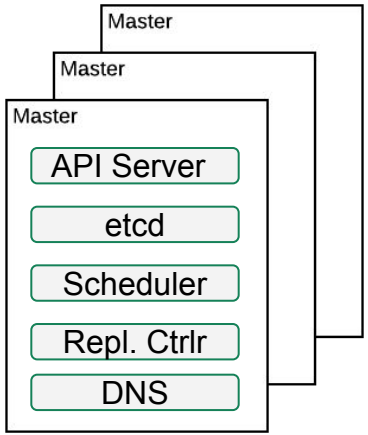

Master
Master
Master
API Server
etcd
Scheduler
Repl. Ctrlr


frontend
Node
Node
Node
Node
Node
Node

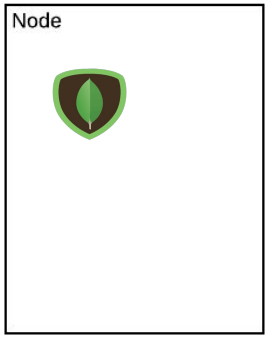
Service

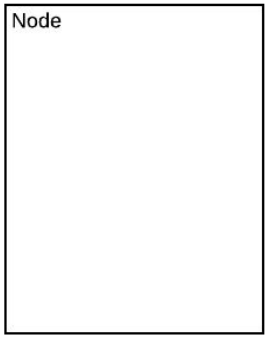

Master
Master
Master
API Server
etcd
Scheduler
Repl. Ctrlr
DNS


IP
DNS NAME
frontend
Node
Node
Node
Node
Node
Node

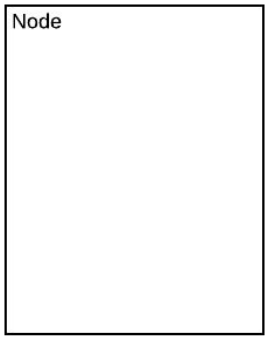

Master
Master
Master
API Server
etcd
Scheduler
Repl. Ctrlr
DNS

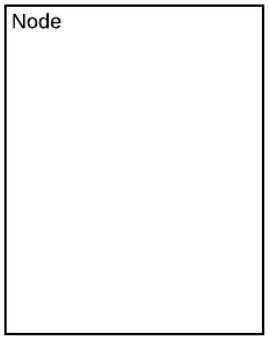

Node


Node

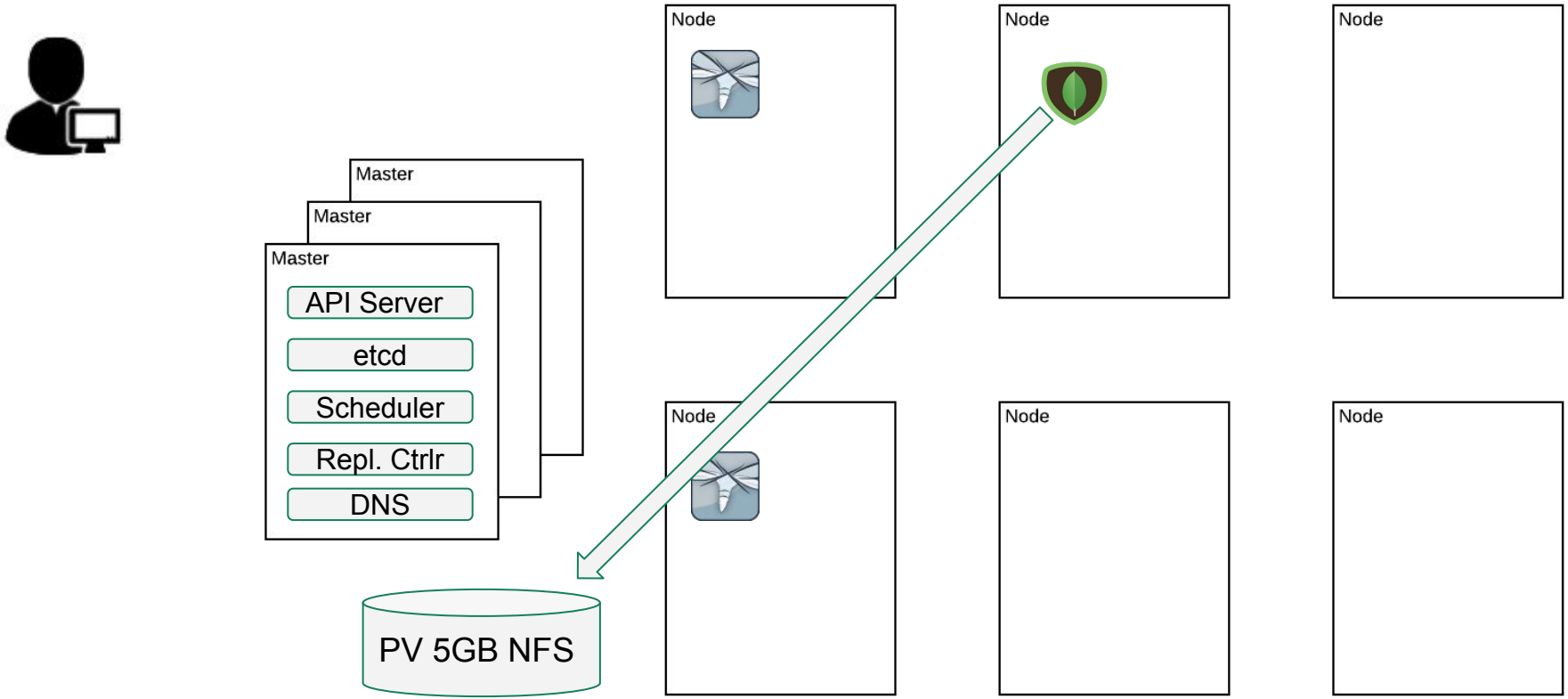

Node


Node

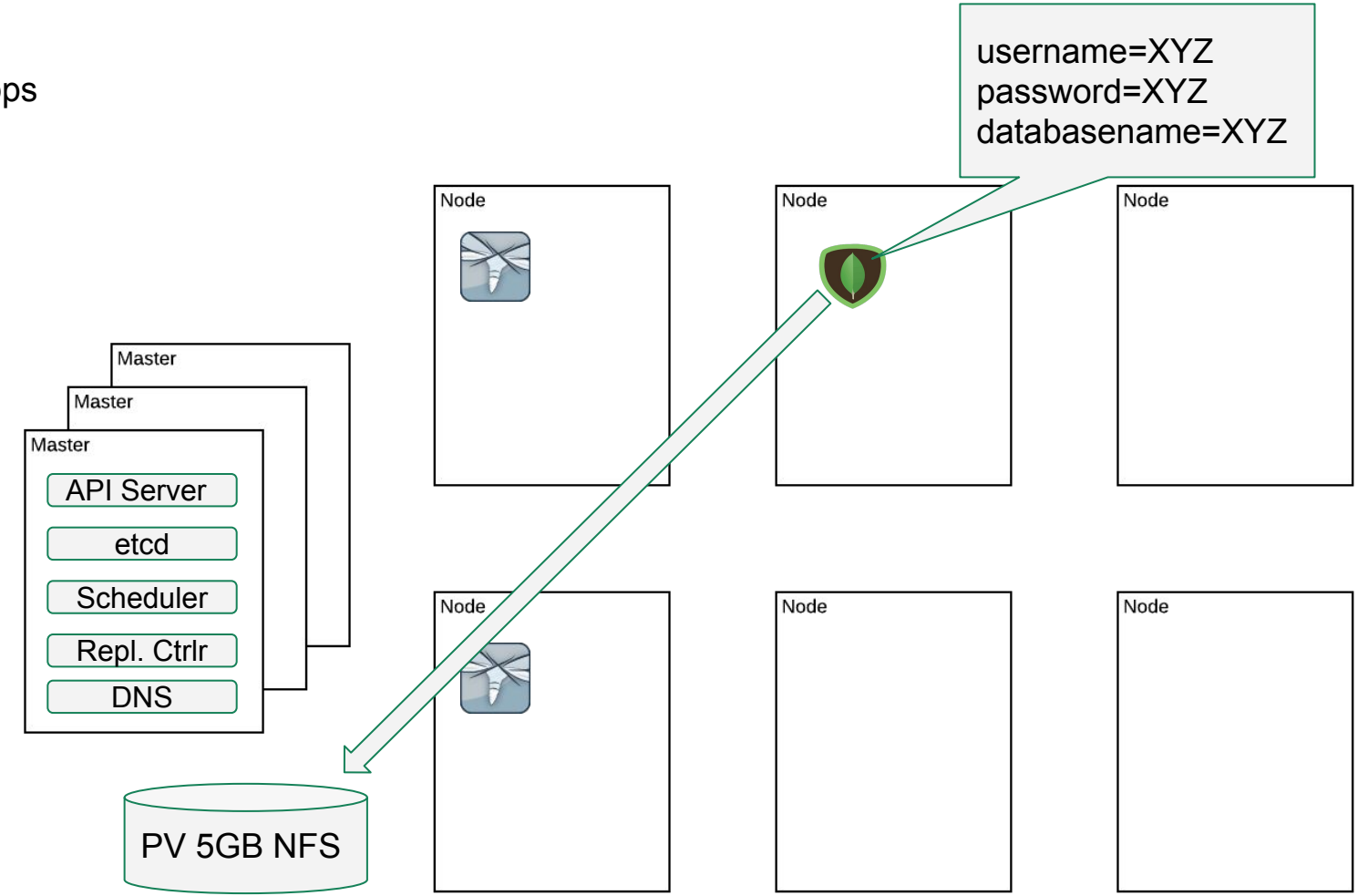

Node


Node

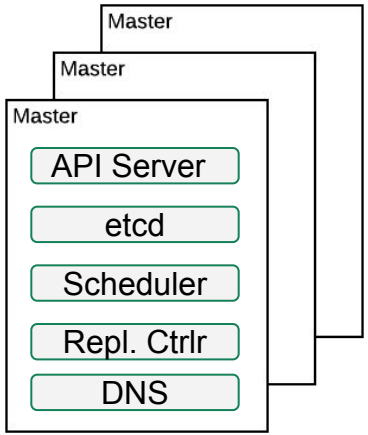
PersistentVolume and PersistentVolumeClaim

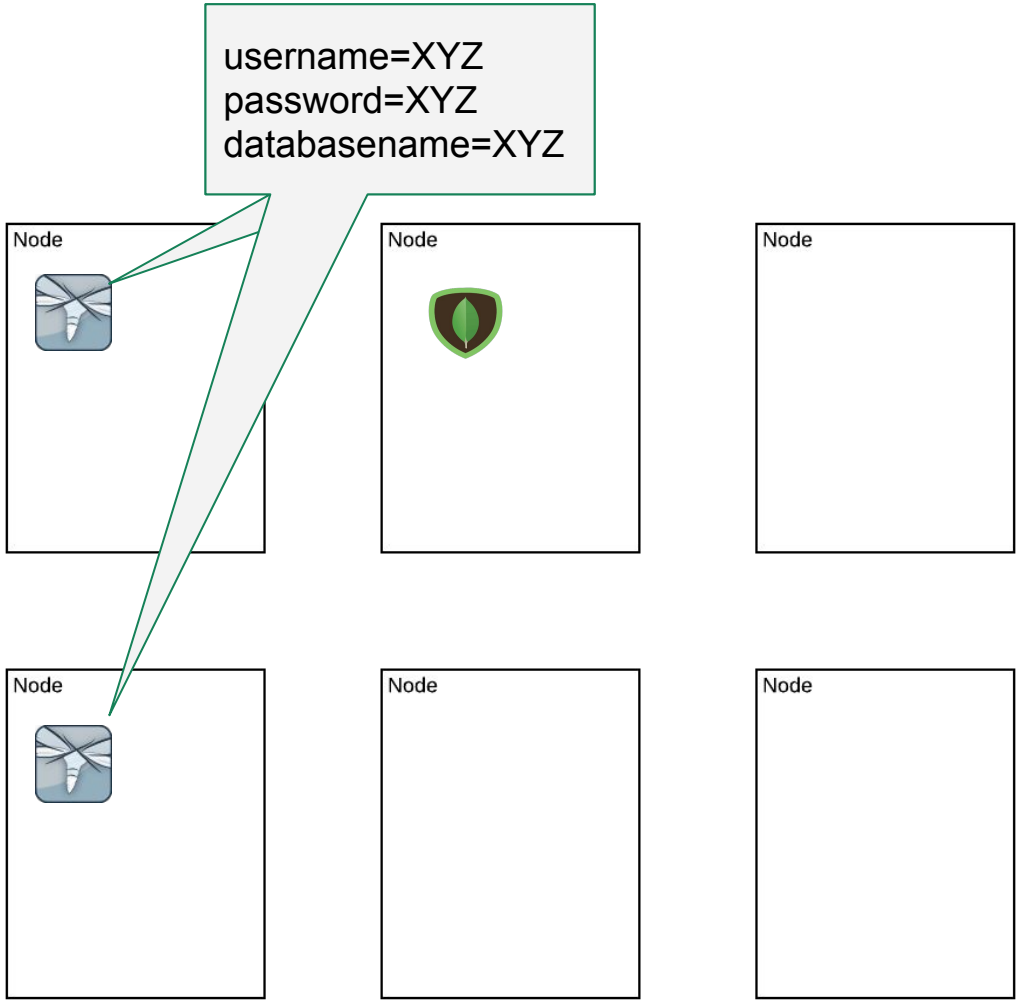


Configuring apps

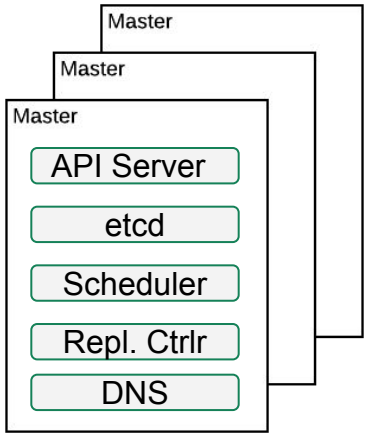


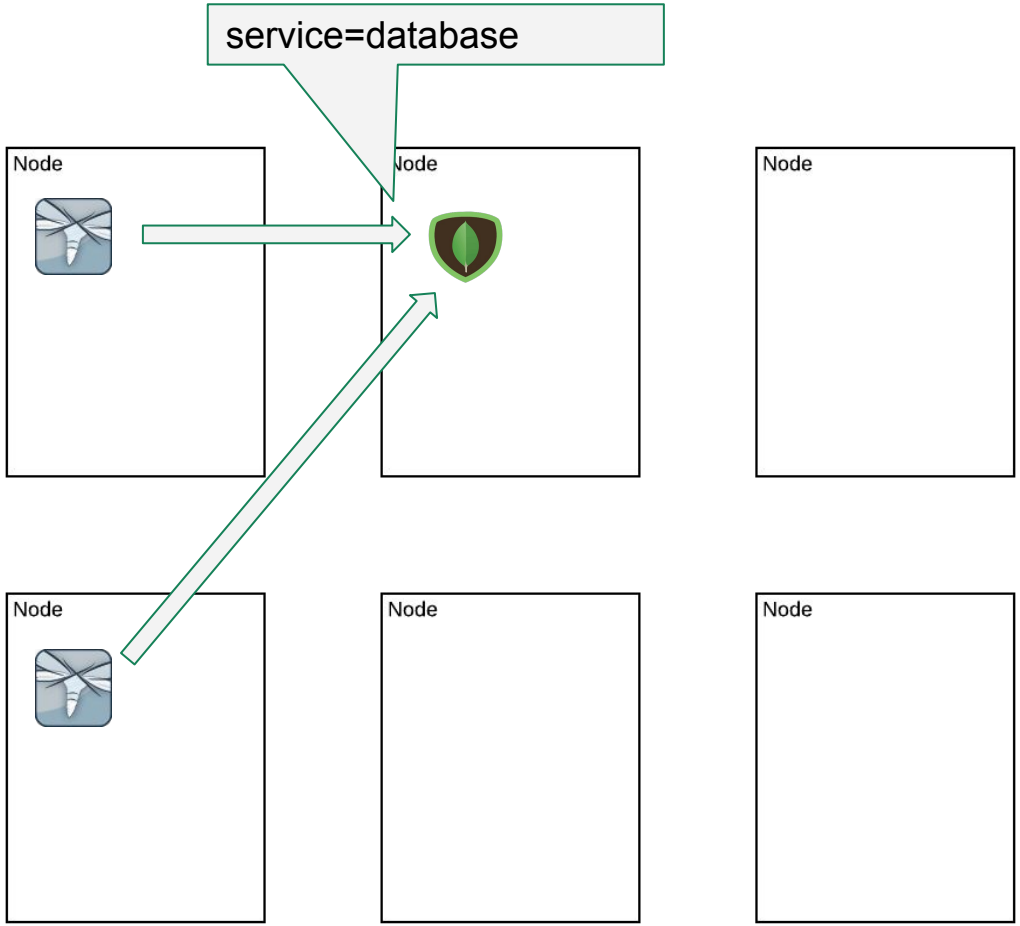
Linking services



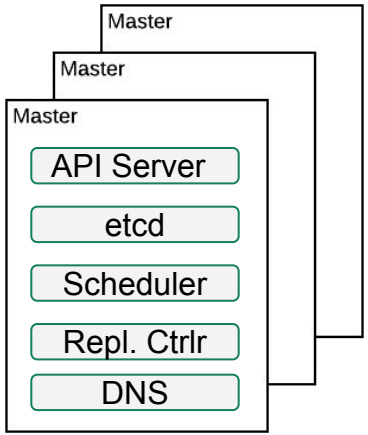



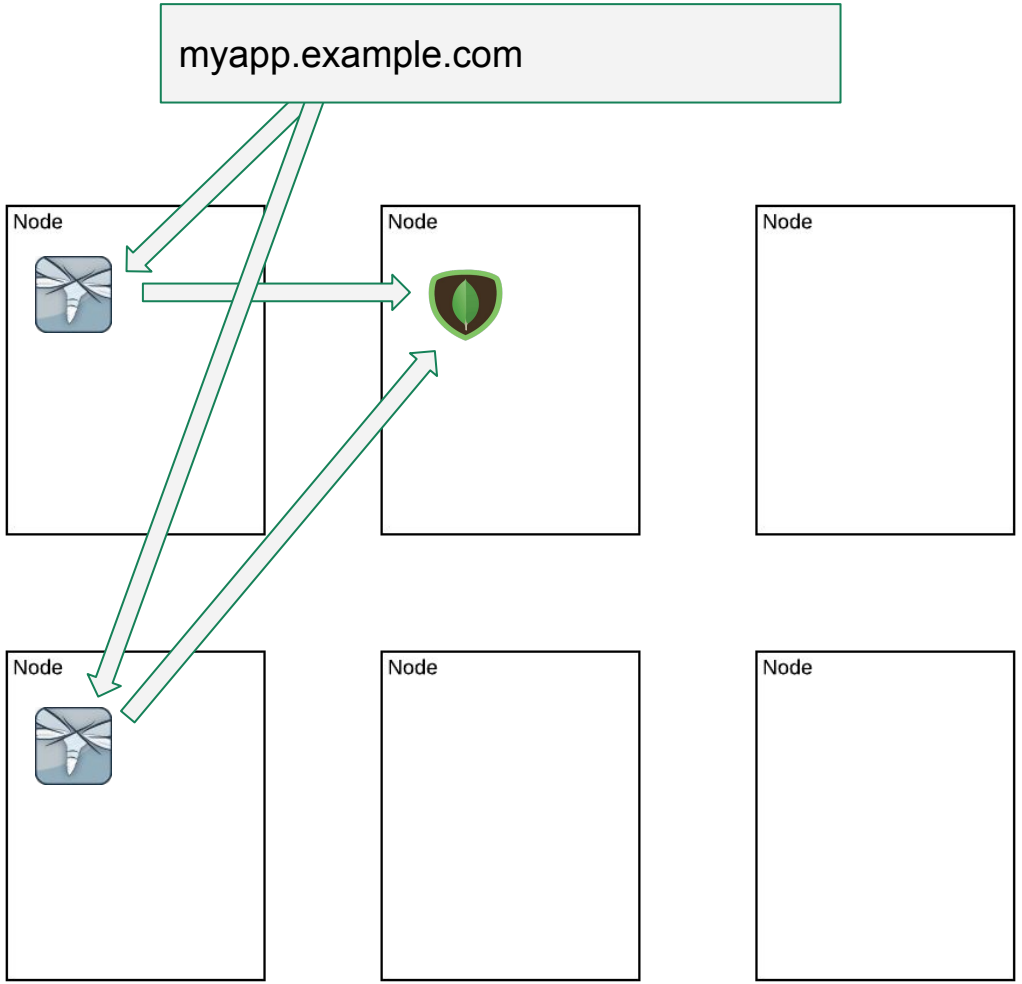
# Linking services



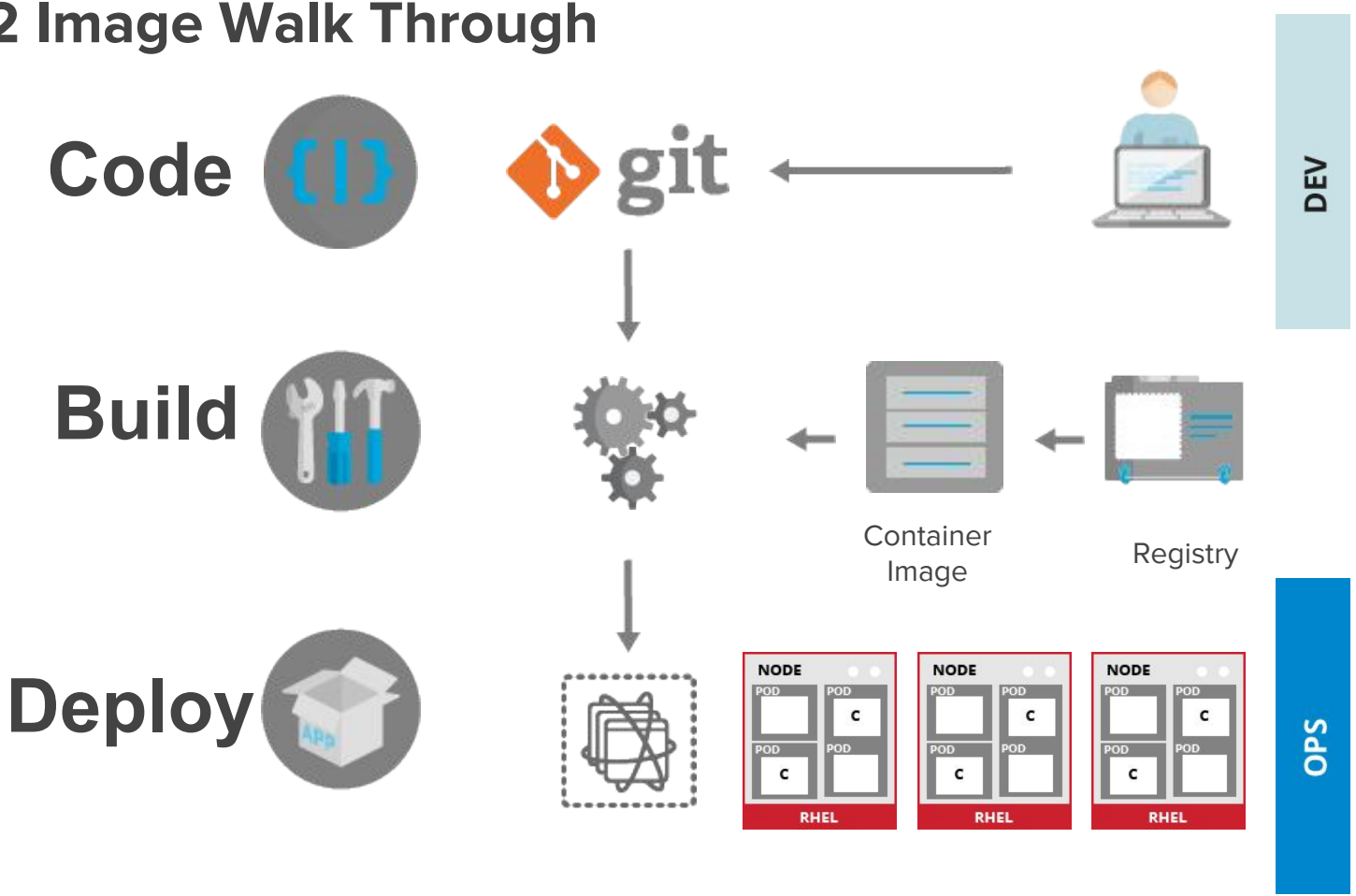



Routing


Master
Master
Master
API Server
etcd
Scheduler
Repl. Ctrlr
DNS


myapp.example.com
Node
Node
Node
Node
Node
Node

# Source 2 Image Walk Through



# Source 2 Image Walk Through

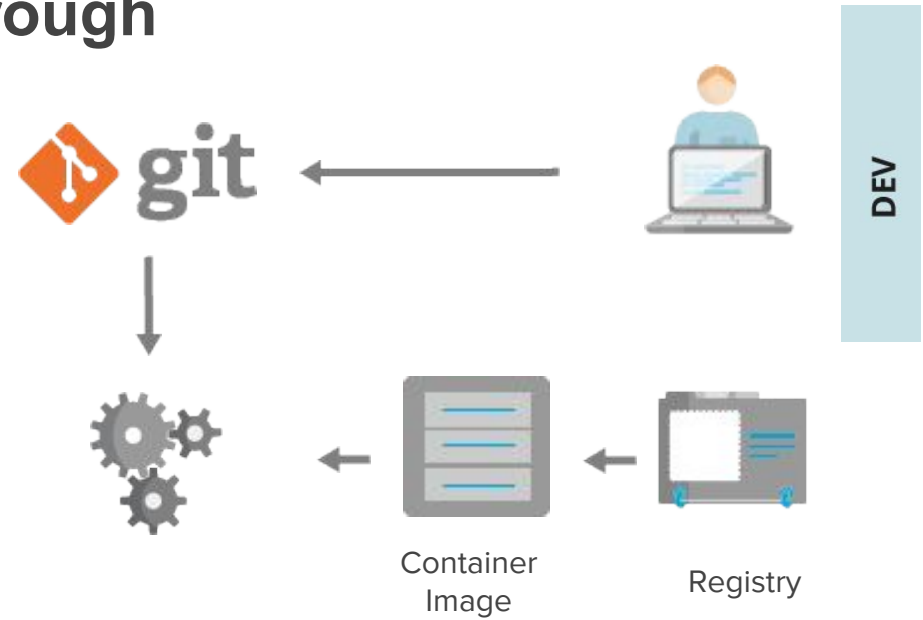

## Code

Developers can leverage existing development tools and then access the OpenShift Web, CLI or IDE interfaces to create new application services and push source code via GIT.  OpenShift can also accept binary deployments or be fully integrated with a customer's existing CI/CD environment.

# Source 2 Image Walk Through

## Build

OpenShift automates the Docker image build process with Source-to-Image (S2I). S2I combines source code with a corresponding Builder image from the integrated Docker registry. Builds can also be triggered manually or automatically by setting a Git webhook. Add in Build pipelines



# Source 2 Image Walk Through

## Deploy

OpenShift automates the deployment of application containers across multiple Node hosts via the Kubernetes scheduler. Users can automatically trigger deployments on application changes and do rollbacks, configure A/B deployments & other custom deployment types.

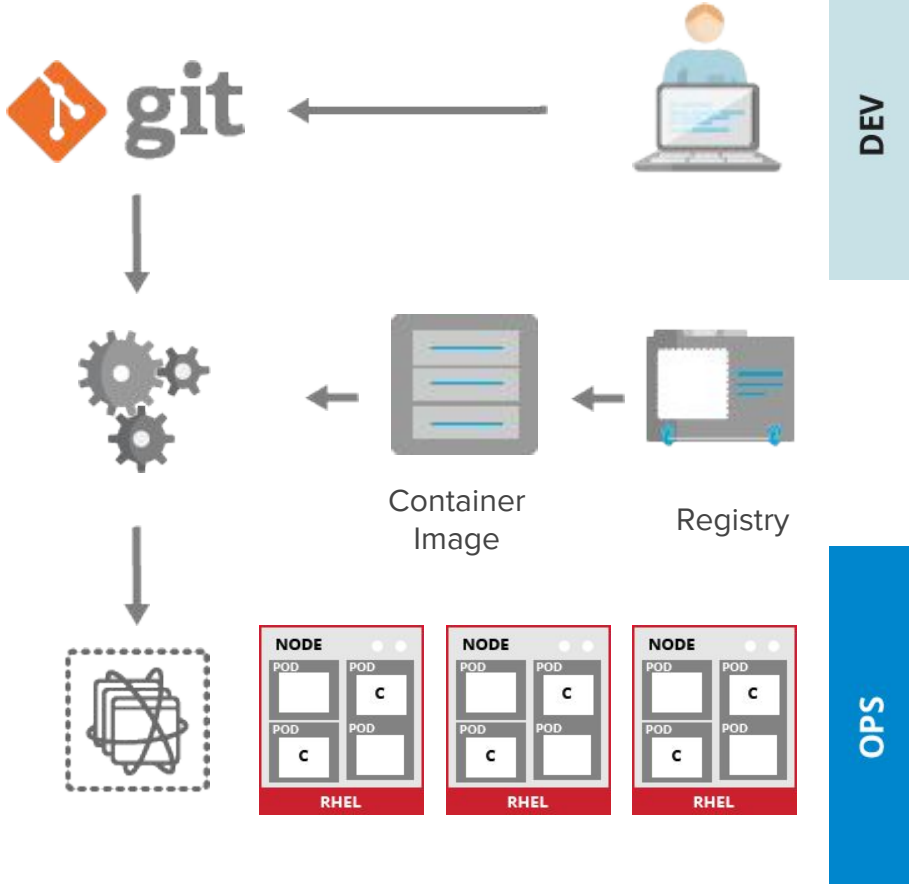
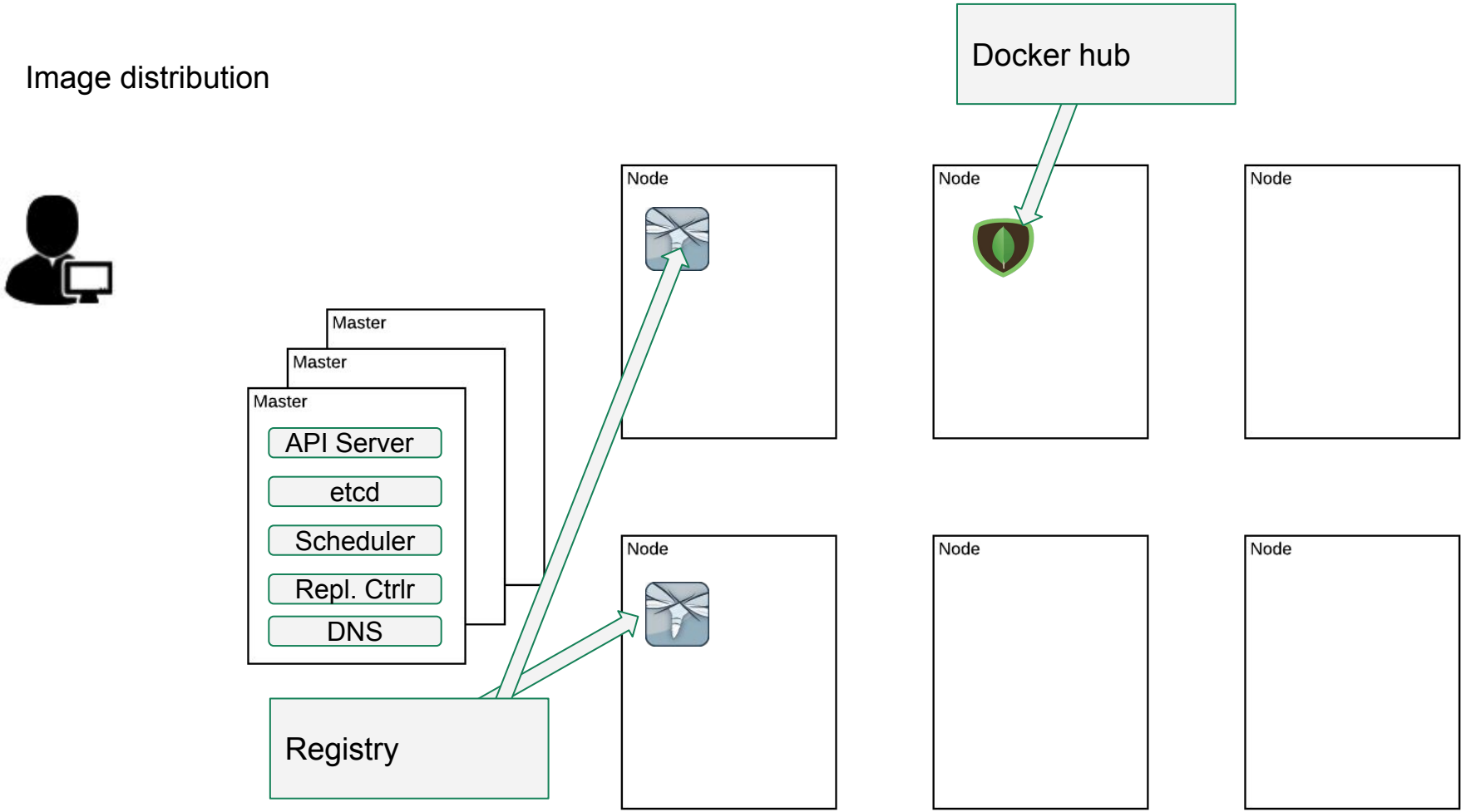


Image distribution



# Time to try it out

Hands on!

bit.ly/openshift-rigadevdays18

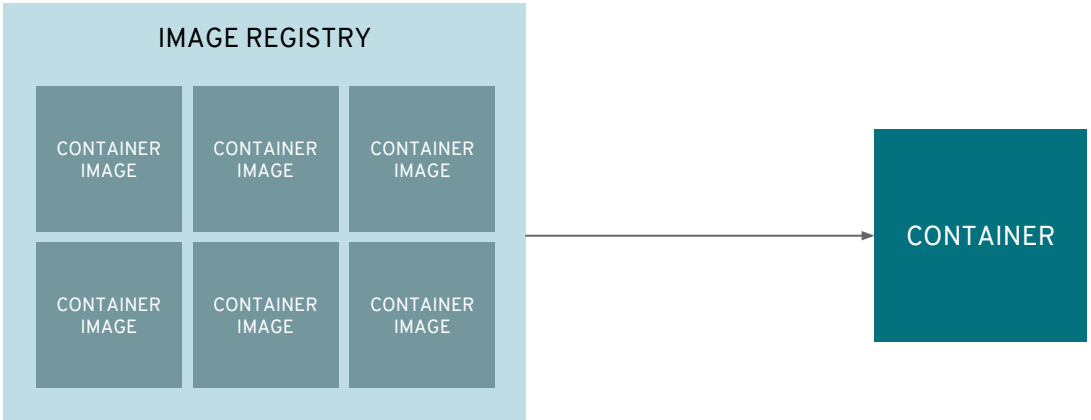
**NOTE: The platform is shared. Don’t abuse it**

# OPENSHIFT CONCEPTS SUMMARY RECAP

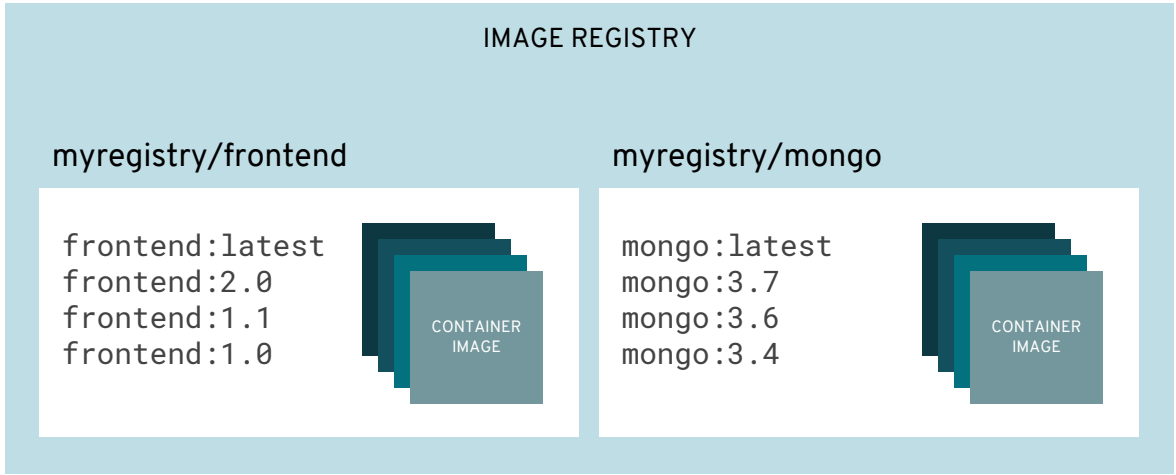
# A container is the smallest compute unit

# containers are created from container images

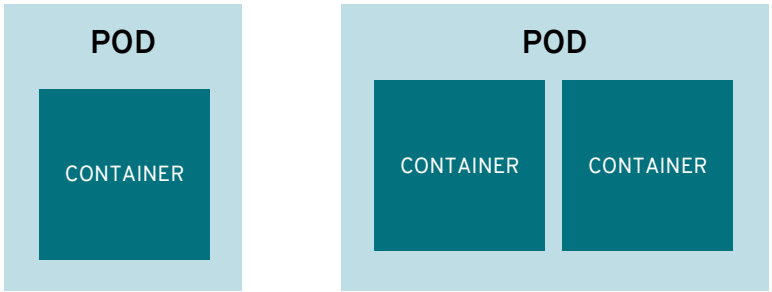
# container images are stored in an image registry



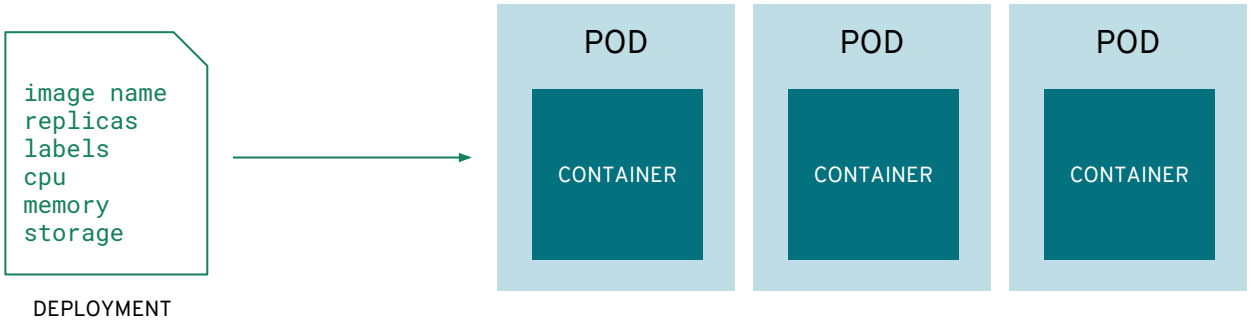
# an image repository contains all versions of an image in the image registry



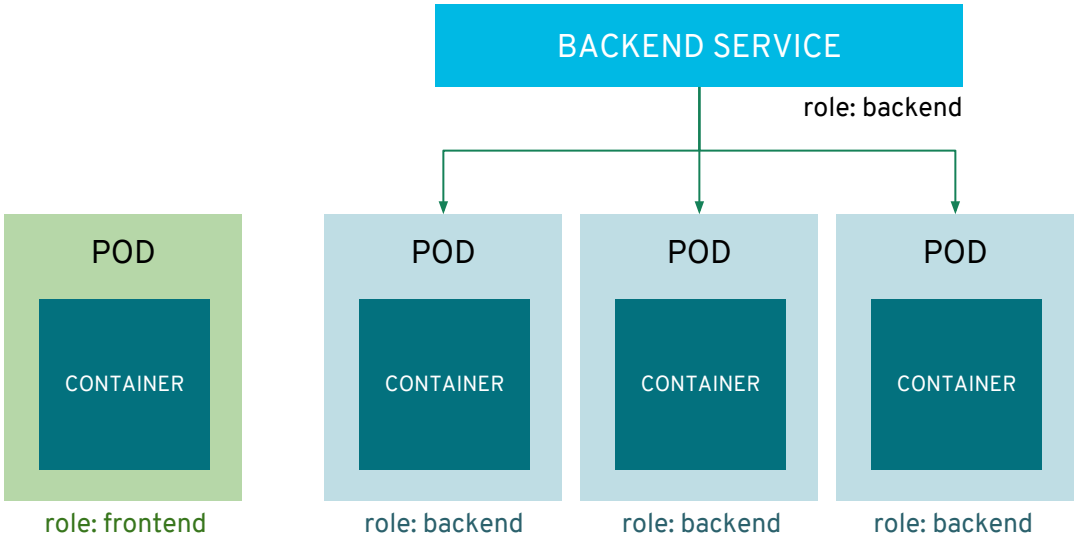
# containers are wrapped in pods which are units of deployment and management



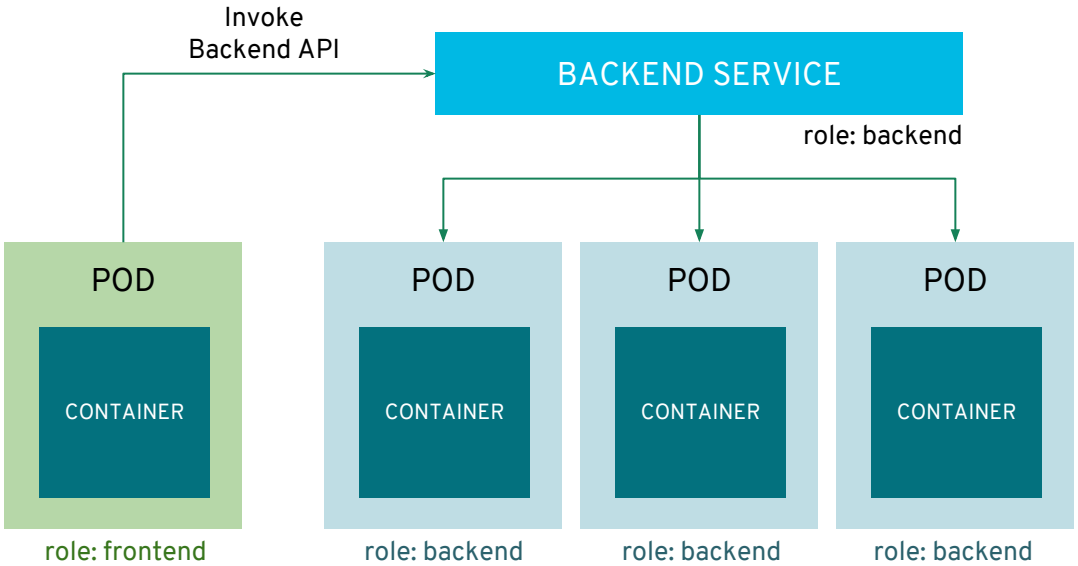
# pods configuration is defined in a `deployment`



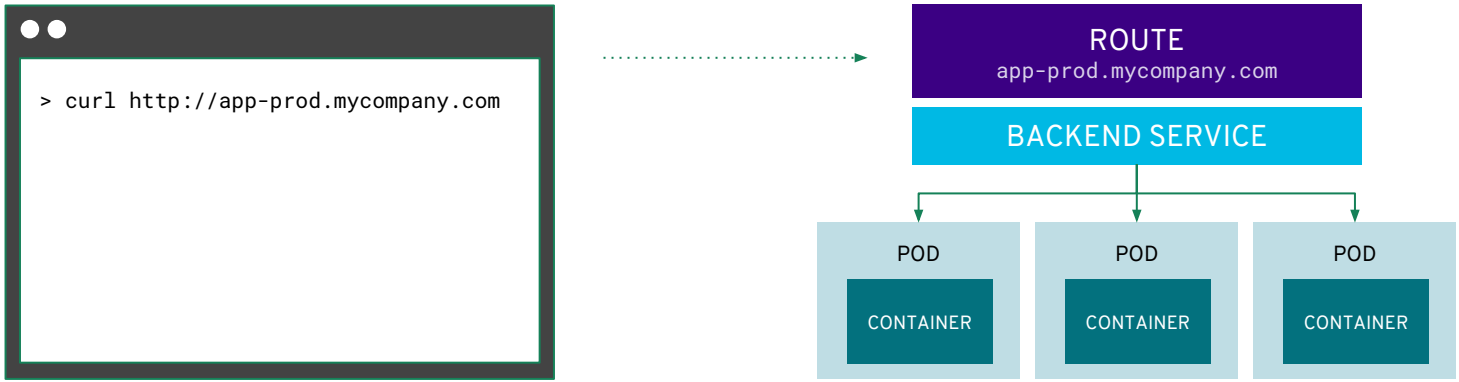
# services provide internal load-balancing and service discovery across pods



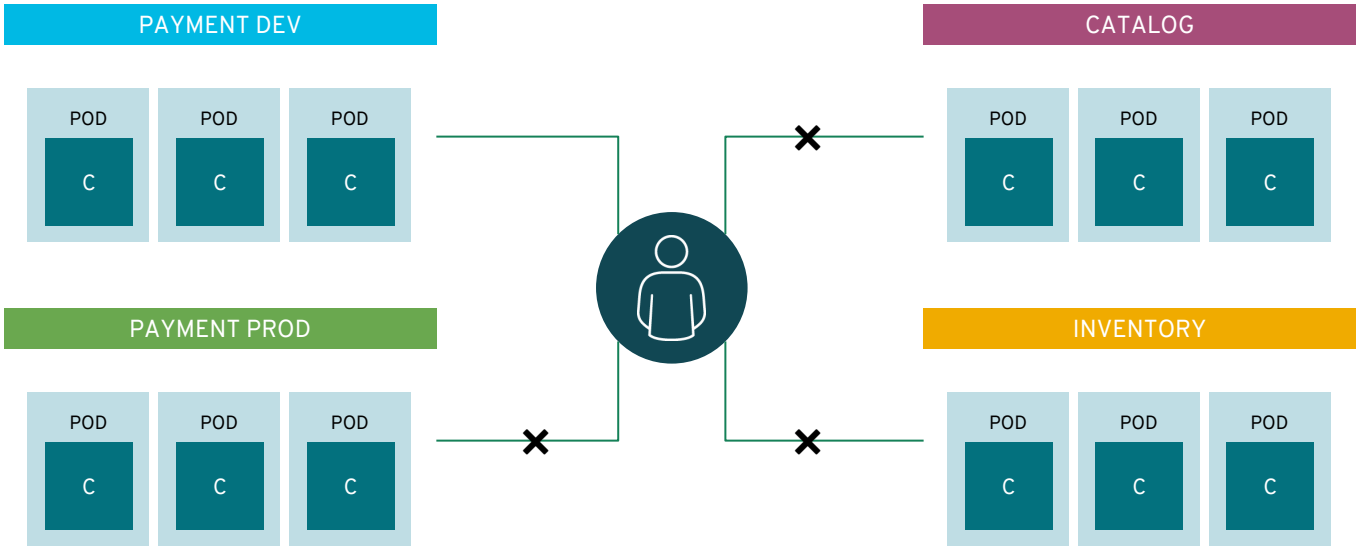
# apps can talk to each other via services



# routes add services to the external load-balancer and provide readable urls for the app

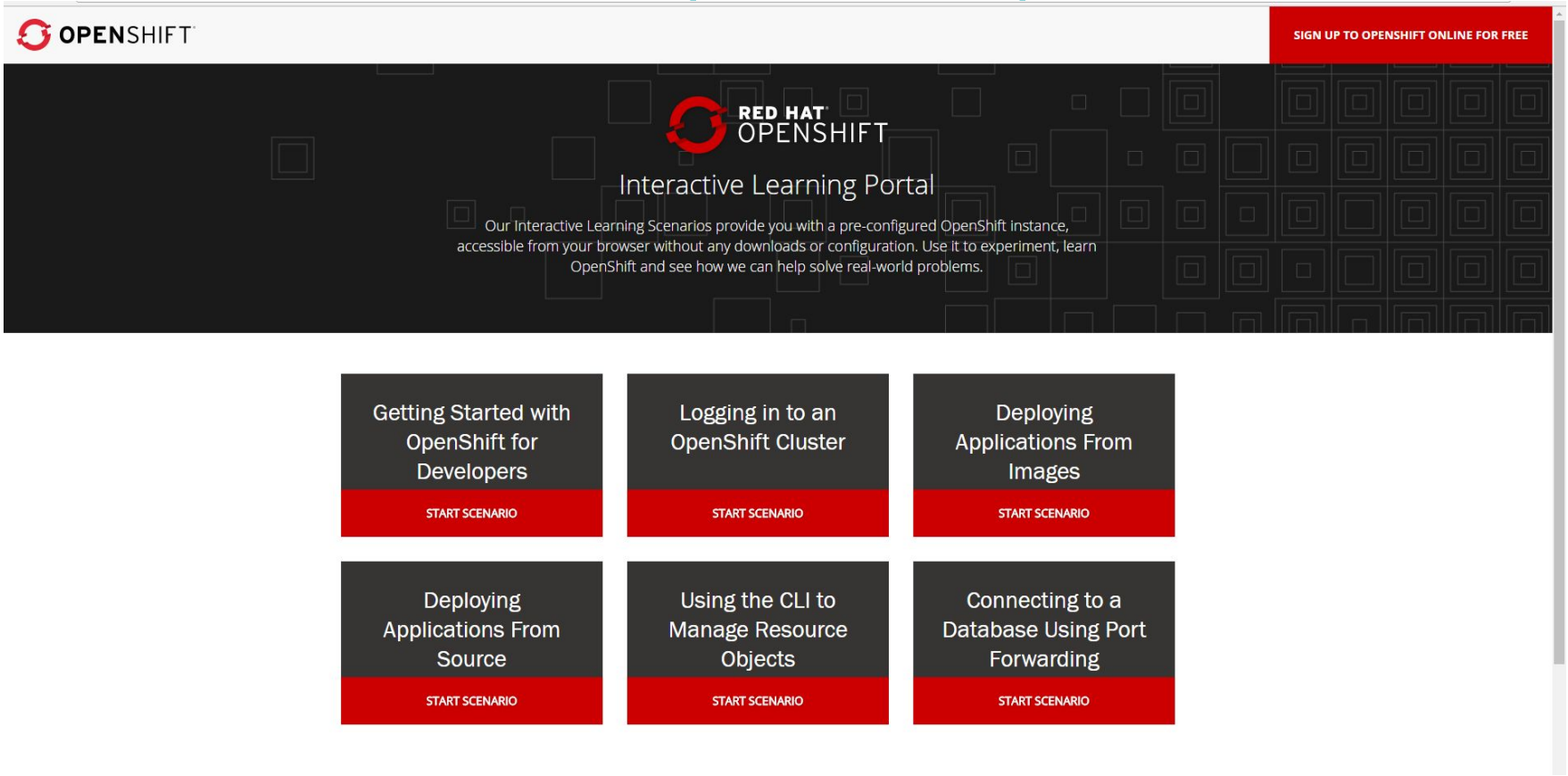


projects allow managing apps in isolation from other environments, teams, groups and departments



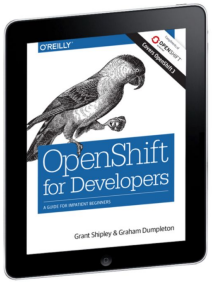
# Where to next

# Self Paced Tutorials - https://learn.openshift.com/




redhat.

# Developer Resources




- eBook (O'Reilly): DevOps with OpenShift
  https://www.openshift.com/promotions/devops-with-openshift.html
- eBook (O'Reilly): OpenShift for Developers:
  https://www.openshift.com/promotions/for-developers.html
- eBook (O'Reilly): Deploying to OpenShift:
  https://www.openshift.com/deploying-to-openshift/

# https://www.openshift.org/minishift/


redhat.

# The code for this labs

- Source code for the examples:
  - https://github.com/openshift-roadshow
- Source code for instructions:
  - https://github.com/openshift-labs/starter-guides

redhat.

# THANK YOU