



Kubernetes Master Course I

Homework dn.dev/kube-tutorial



Upcoming DevNation Schedule

| | India (English) UTC: 9:30 Local 15:00 | France (French) UTC 11:00 Local 13:00 | Spain (Spanish) UTC 13:00 Local 15:00 | Brazil (Portuguese) UTC 14:00 Local 11:00 | USA (English) UTC 16:00 Local 12:00 Eastern | Mexico (Spanish) UTC 18:00 Local 13:00 Central Standard |
|---------------|---|---|---|---|---|--|
| April 6 2020 | | | | Kubernetes I | Kubernetes I | |
| April 7 2020 | | | | Kubernetes 2 | Kubernetes 2 | |
| April 9 2020 | | | | | <u>Tech Talk Thursday</u> | |
| April 13 2020 | Kubernetes 1 | Kubernetes 1 | Kubernetes 1 | | | |
| April 14 2020 | Kubernetes 2 | Kubernetes 2 | Kubernetes 2 | | | |
| April 16 2020 | | | | | <u>Tech Talk Thursday</u> | |
| April 20 2020 | Istio | Istio | Istio | Istio | Istio | |
| April 21 2020 | Knative | Knative | Knative | Knative | Knative | |
| April 22 2020 | Tekton | Tekton | Tekton | Tekton | Tekton | |
| April 23 2020 | | | | | <u>Tech Talk Thursday</u> | |



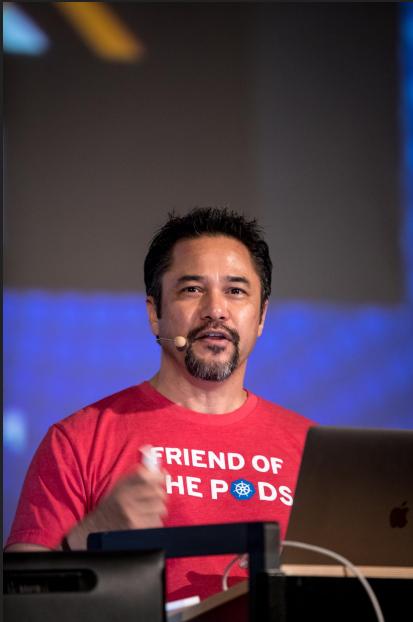
Red Hat
Summit

From here, anywhere

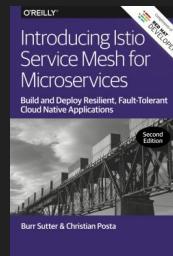
Red Hat Summit 2020
Virtual Experience
April 28-29

Immerse yourself in our free virtual event and find your inspiration at the intersection of choice and potential.

[Learn more](#)



- burr@redhat.com
- Featured speaker at technology events around the globe
- Java Champion since 2005
- Former President of the Atlanta Java User Group
- Founded the DevNexus conference
- Always looking for technologies that enable developers to deliver better software ever faster





@yanaga



Red Hat
Developer



DevNation



Burr Survey Link

<https://www.surveymonkey.com/r/S9868TY>

Edson Survey Link

<https://www.surveymonkey.com/r/2J9LDKZ>

Kamesh Sampath([devadvocate.yaml](#))



```
apiVersion: bio/v1
kind: DeveloperAdvocate
spec:
  title: "Director of Developer Experience @ Red Hat"
  gde:
    expert: cloud
    year: 2019
  javaDeveloper: true
  opensourceContributions:
    - Eclipse Che
    - Quarkus - Supersonic Subatomic Java
    - kubernetes/minikube: Run Kubernetes locally
    - Camel K
  social-handles:
    linkedin: https://linkedin.com/in/kameshsampath
    github: kameshsampath \(Kamesh Sampath\)
    twitter: Kamesh Sampath \(@kamesh\_sampath\)
    blogs: developers.redhat.com\(Kamesh Sampath\)
dn.dev/kubemaster1
```

kubectl email --address kameshs@redhat.com





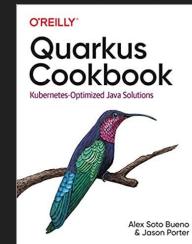
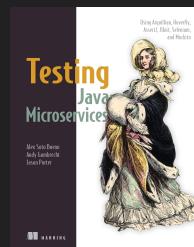
Red Hat
Developer

DevNation

Alex Soto (lordofthejars.com)



- [@alexsotob](https://twitter.com/alexsotob)
- asotobue@redhat.com
- Currently Red Hat's Director of Developer Experience
- Featured speaker at technology events around the globe
- A Java Champion since 2017
- Writer, University Professor, Radio collaborator
- A big fan of testing and continuous delivery in 21st century



Sebastien Blanc



- @sebi2706
- sblanc@redhat.com
- Currently Red Hat's Director of Developer Experience
- Featured speaker at technology events around the globe
- Java Champion
- Co-organizer of Riviera DEV
- Passion-Driven-Developer with one goal : share his passion by giving talks that are pragmatic, fun and focused on live coding



Agenda

Master Course I

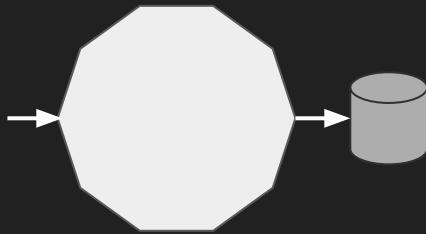
- Why Kubernetes
- What is Kubernetes
- Installation
- kubectl
- Pod
- ReplicaSet
- Deployment
- logs, stern
- Debugging with exec
- Apps as a Service
- Rolling Updates

Master Course 2

- Building Images
- Resource Limits
- Live & Ready
- Service Magic
- Blue/Green
- Env & ConfigMap
- Secrets
- Custom Resource Definitions
 - Pizzas & Apache Kafkas

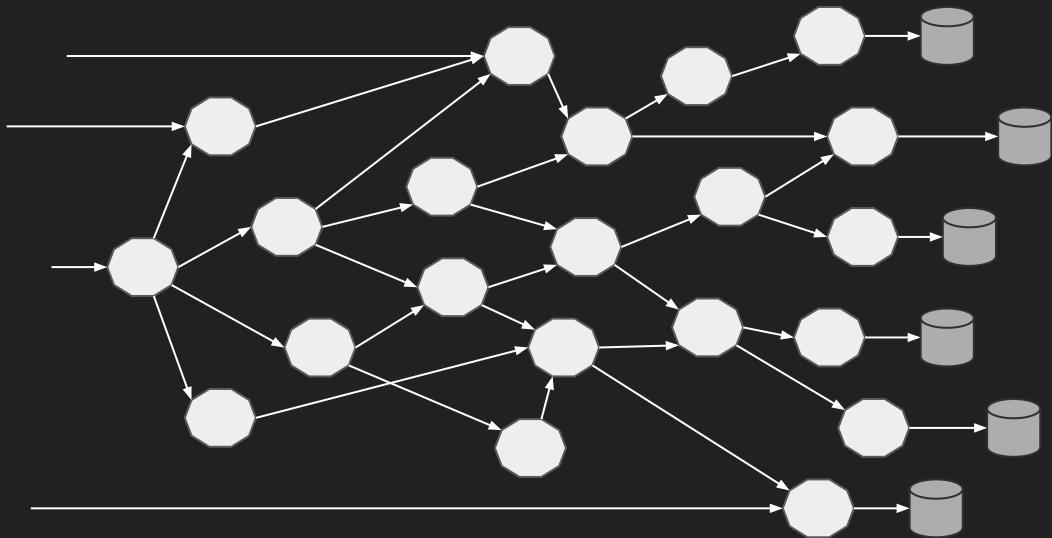
Why Kubernetes?

Old School



Love Thy Mono

New School





A Challenge

Have you ever had "/" vs "\" break your app? Or perhaps needed a unique version of a JDBC driver? Or had a datasource with a slightly misspelled JNDI name? Or received a patch for the JVM or app server that broke your code?

Containerize
Your
App

| | |
|----------------------|--|
| .war or .ear | |
| Custom Configuration | JDBC driver, datasource, JMS queue, users |
| Application Server | Weblogic 10.x.y, Tomcat 6.x.y, JBoss EAP 6.x.y |
| Java Virtual Machine | Java 1.6.6_45 or Java 1.7.0_67 |
| Operating System | Linux Kernel Version & Distribution |
| Server Hardware | |



Email

MyApp.war has been tested with the following

On my Windows 7 desktop

JDK 1.8.43

Wildfly 9

Configuration:

Datasource: MySQLDS

Tested with: mysql-connector-java-5.1.31-bin.jar

dn.dev/kubemaster1

Production Environment

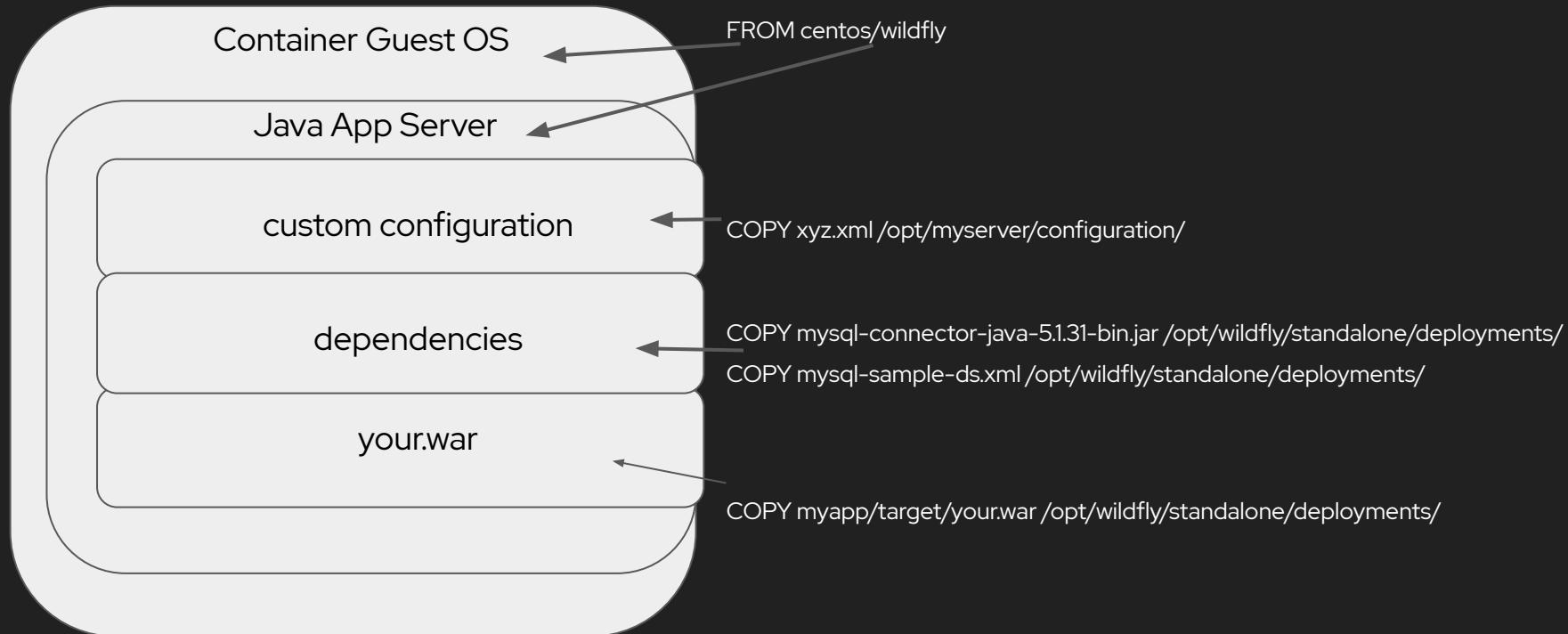
Red Hat Enterprise Linux 6.2

JRE 1.7.3

WebSphere 8.5.5

Oracle 9

Dockerfile



DevOps Challenges for Multiple Containers

- How to scale?
- How to avoid port conflicts?
- How to manage them on multiple hosts?
- What happens if a host has trouble?
- How to keep them running?
- How to update them?
- Rebuild Container Images?





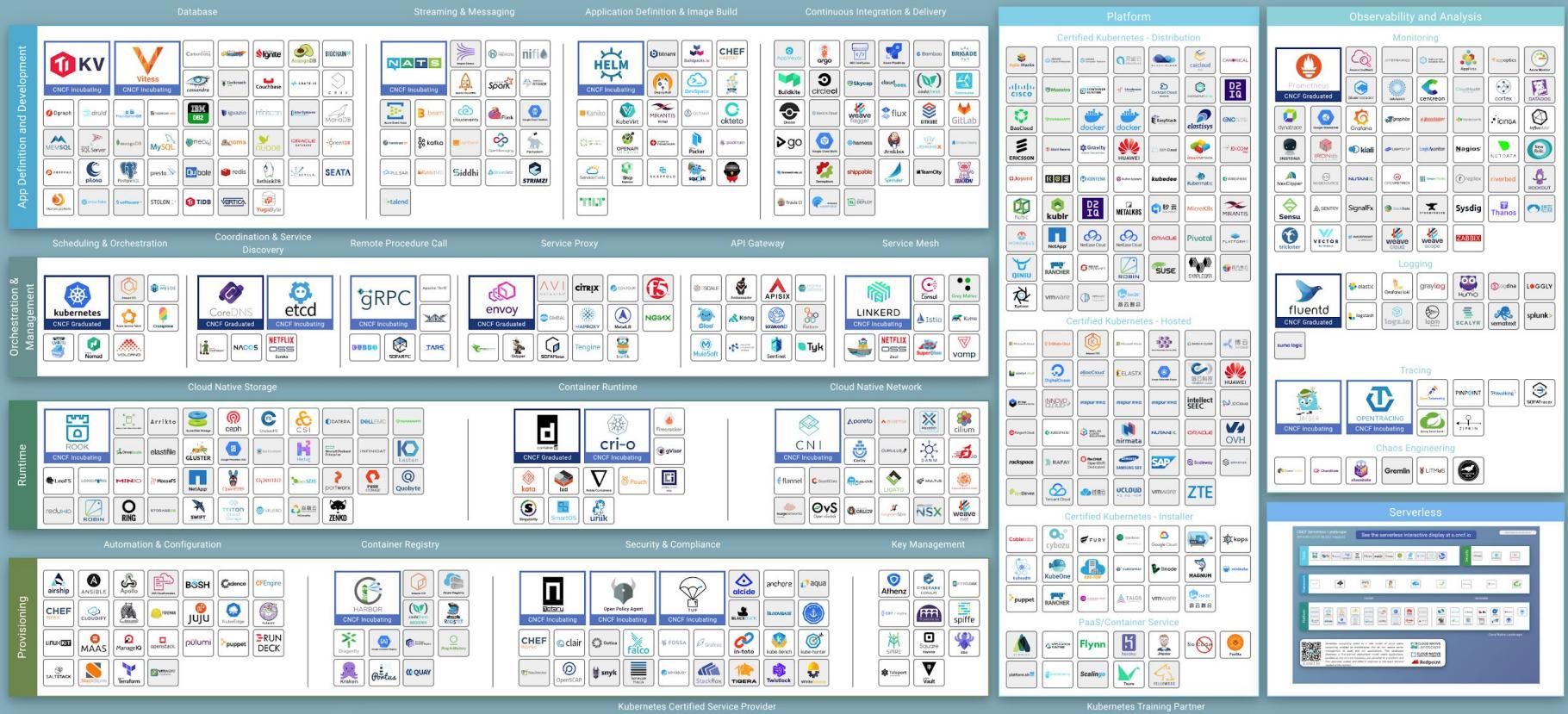
[Learn more](#)

Meet Kubernetes

- Greek for “Helmsman,” also the root of the word “Governor” (from latin: gubernator)
- Container orchestrator
- Supports multiple cloud and bare-metal environments
- Inspired by Google’s experience with containers
- Open source, written in Go
- Manage applications, not machines



OPENSHIFT



A QR code with a small blue icon in the center, which appears to be a camera or a similar symbol. The text "l.cncf.io" is printed below the QR code.

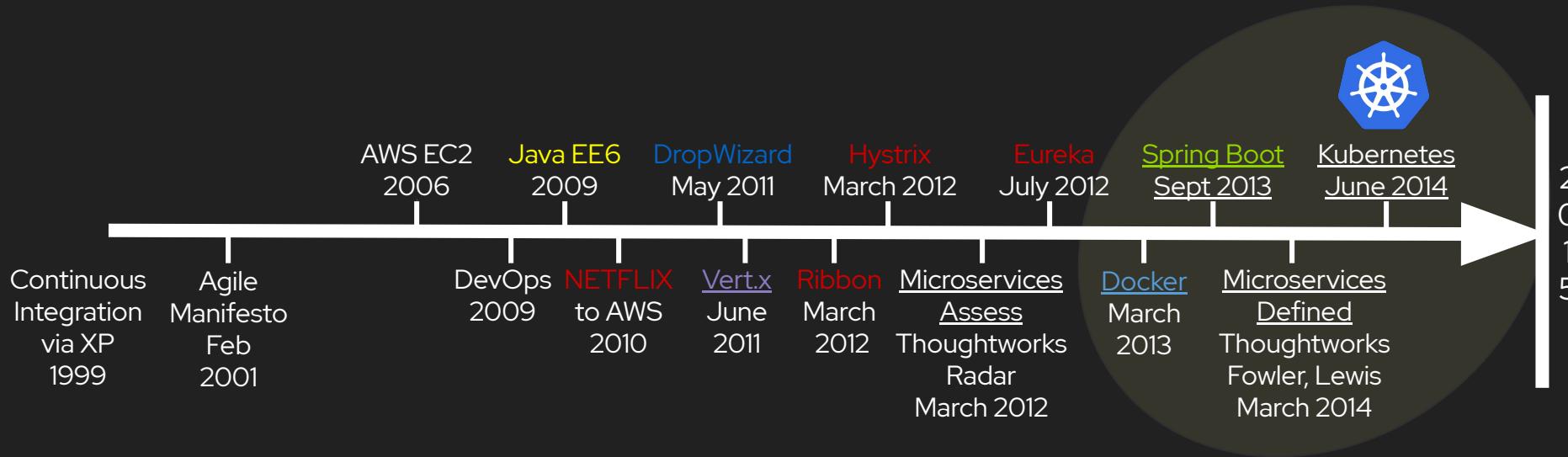



CLOUD NATIVE
 COMPUTING FOUNDATION

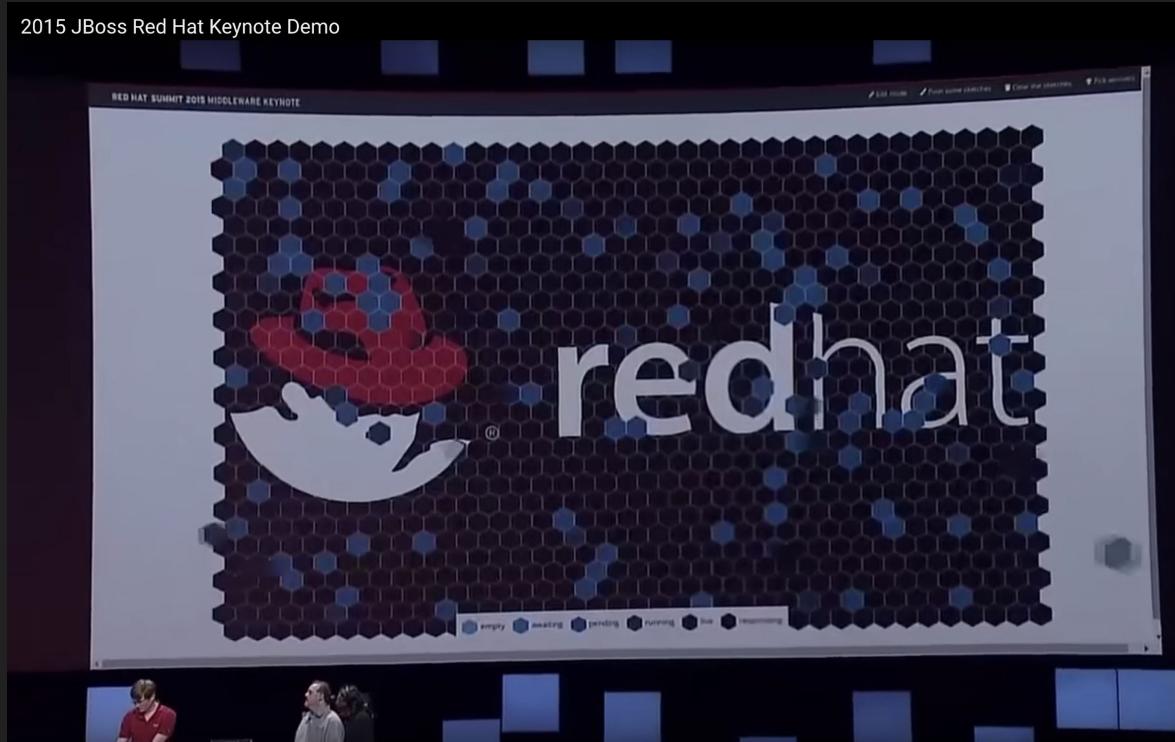
spec



History of Microservices



2015 JBoss Red Hat Keynote Demo



2015

Launch 1000+
Containers

Audience
Claims a
Container

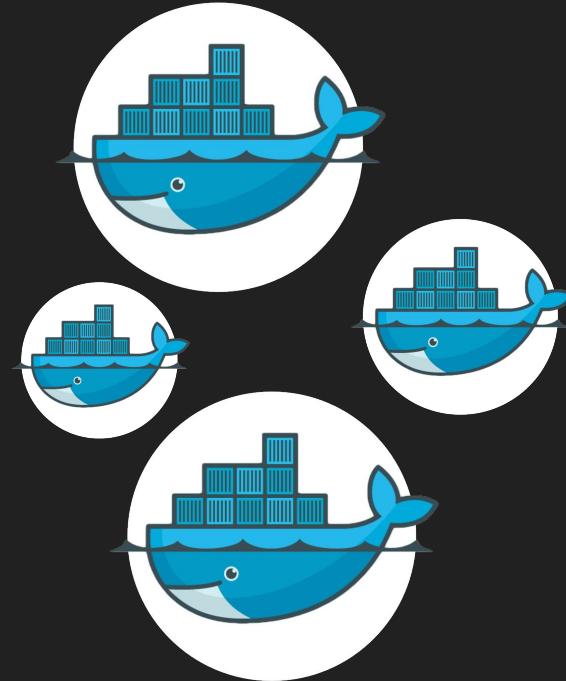
<https://www.youtube.com/watch?v=GCtpncAOEa0&feature=youtu.be&t=1031>

What is Kubernetes?

Pods

A group of whales is commonly referred to as a pod and a pod usually consists a group of whales that have bonded together either because of biological reasons or through friendships developed between two or more whales.

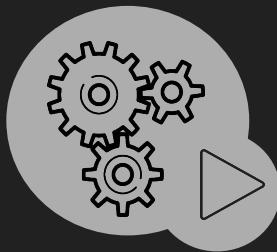
In many cases a typical whale pod consists of anywhere from 2 to 30 whales or more.*



*<http://www.whalefacts.org/what-is-a-group-of-whales-called/>

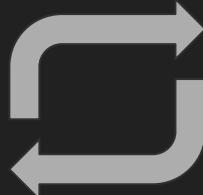
Kubernetes Terms

Pod



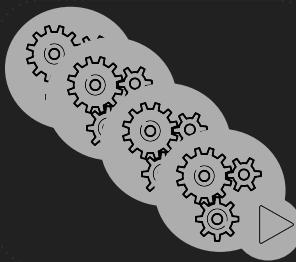
- ✓ 1+ containers
- ✓ Shared IP
- ✓ Shared storage (ephemeral)
- ✓ Shared resources
- ✓ Shared lifecycle

Replicaset/ Deployment



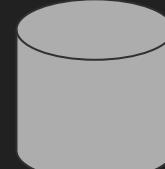
- ✓ The Desired State - replicas, pod template: health checks, resources, image

Service



- ✓ Grouping of pods (acting as one) has stable virtual IP and DNS name

Persistent Volume



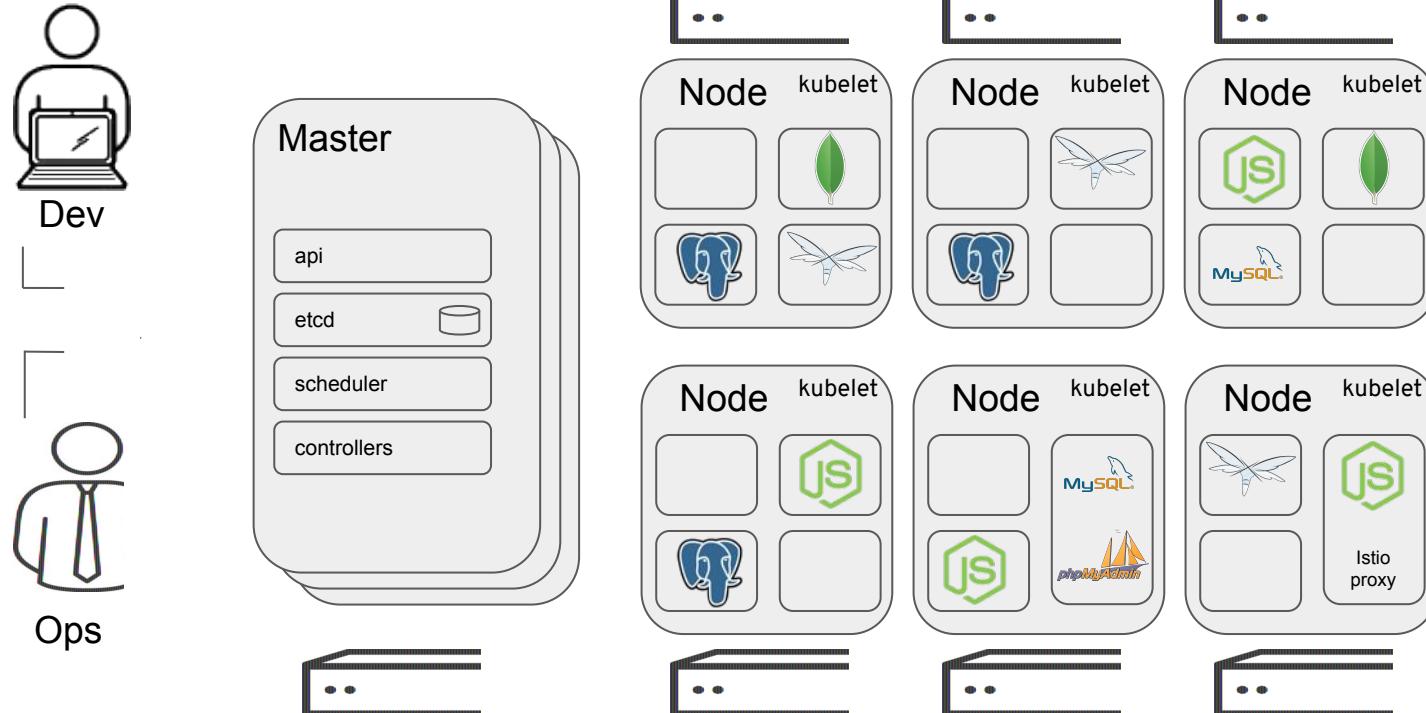
- ✓ Network available storage
- ✓ PVs and PVCs

Label



- ✓ Key/Value pairs associated with Kubernetes objects (env=production)

Kubernetes Cluster - Nodes





Red Hat
Developer

DevNation

Kubernetes Cluster - Declarative

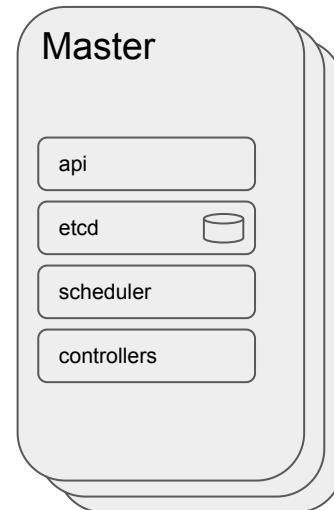


Dev

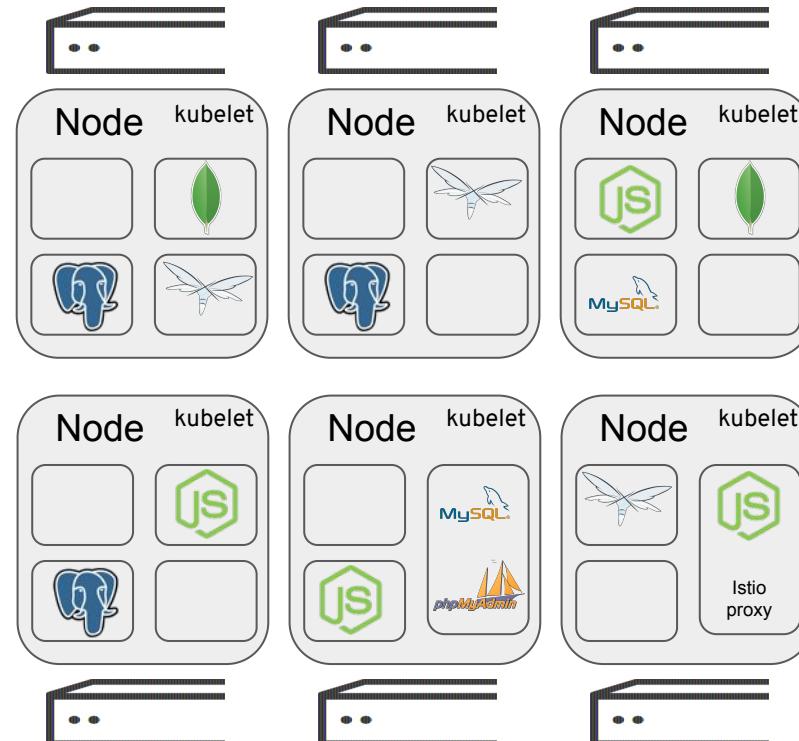
image: repo/mytomcat:v1
replicas: 4



Ops



..



dn.dev/kubemaster1

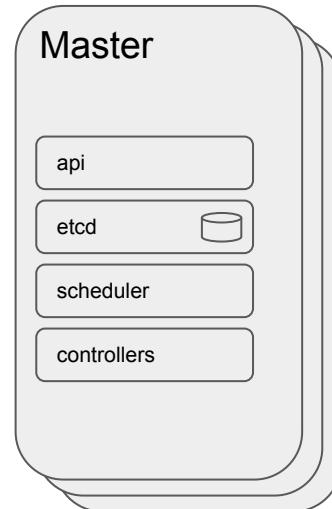
Kubernetes Cluster - 4 Tomcats



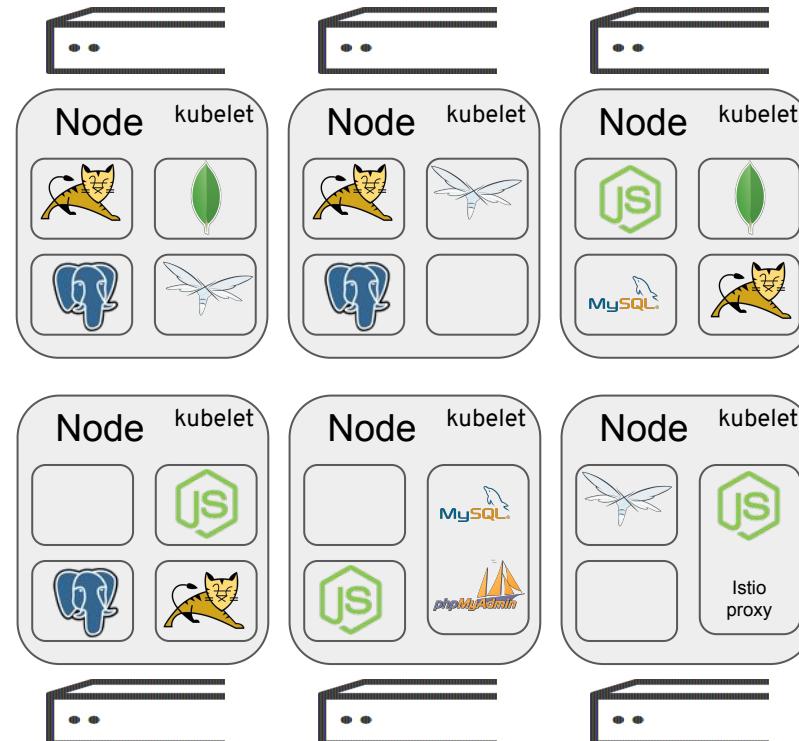
Dev



Ops



...



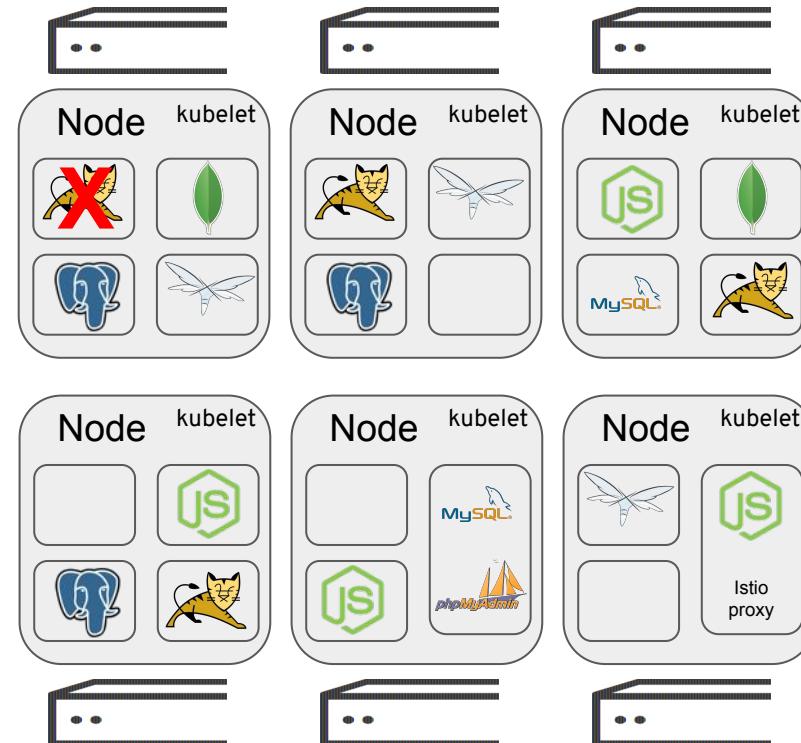
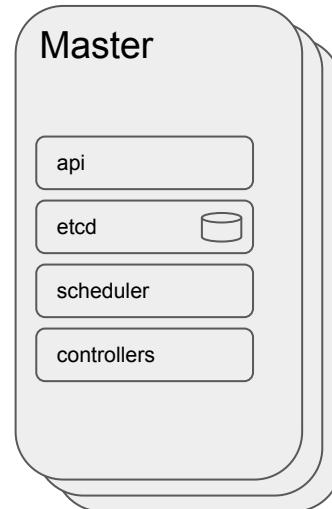
Kubernetes Cluster - Pod Fail



Dev



Ops



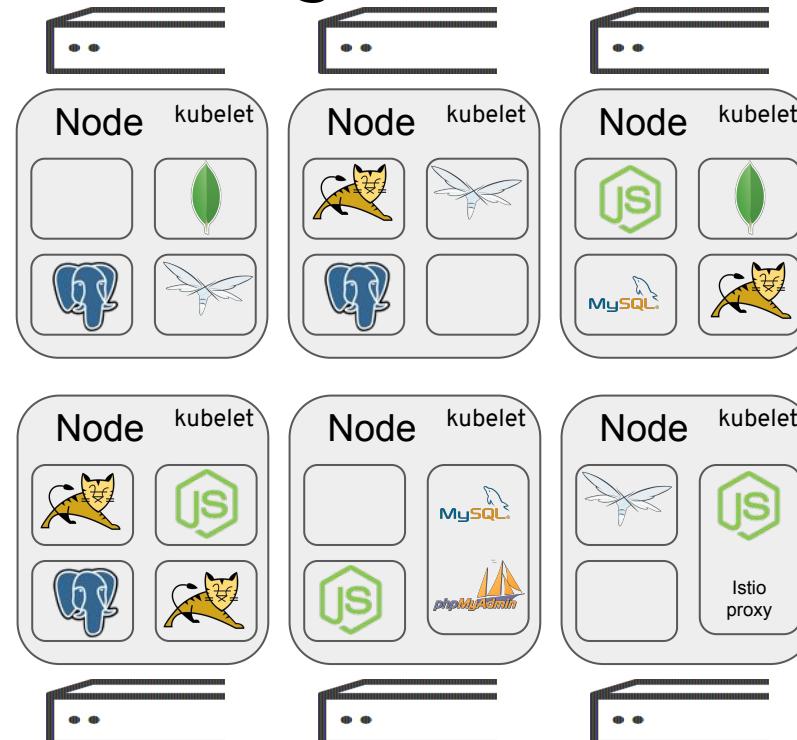
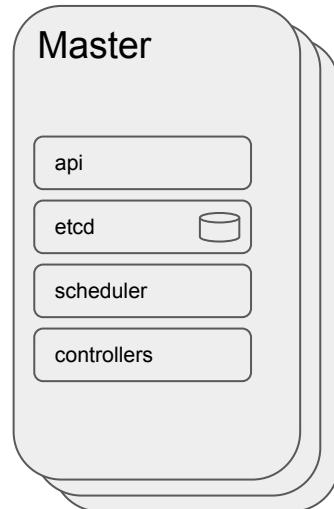
Kubernetes Cluster - Correcting



Dev



Ops



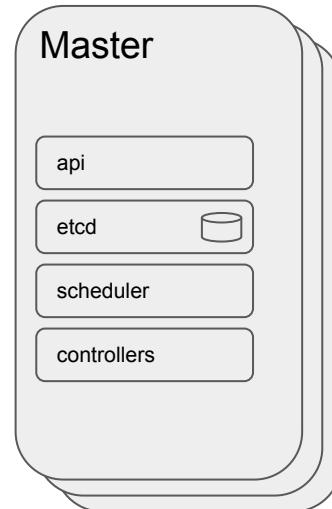
Kubernetes Cluster - Node Fail



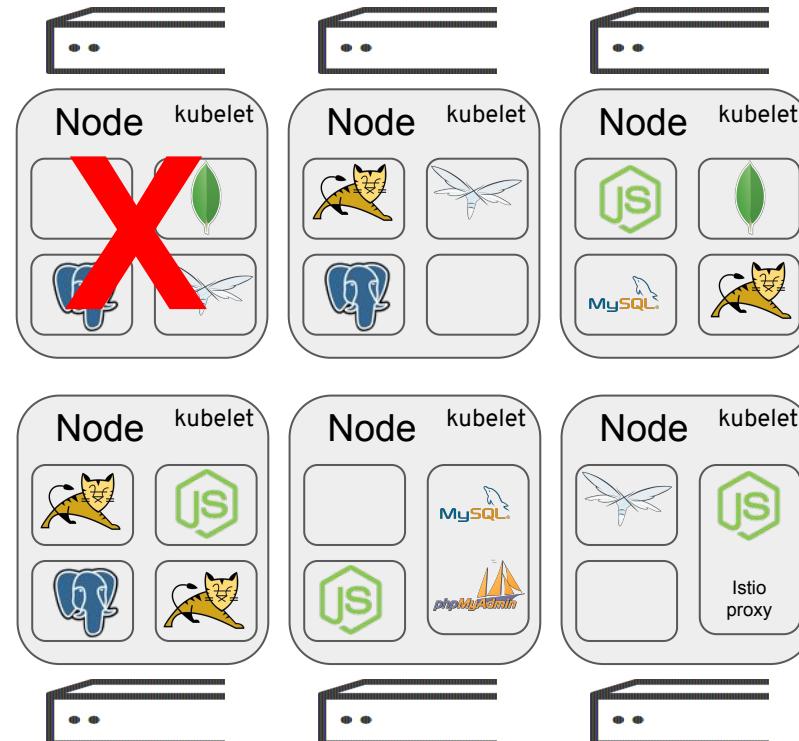
Dev



Ops



...



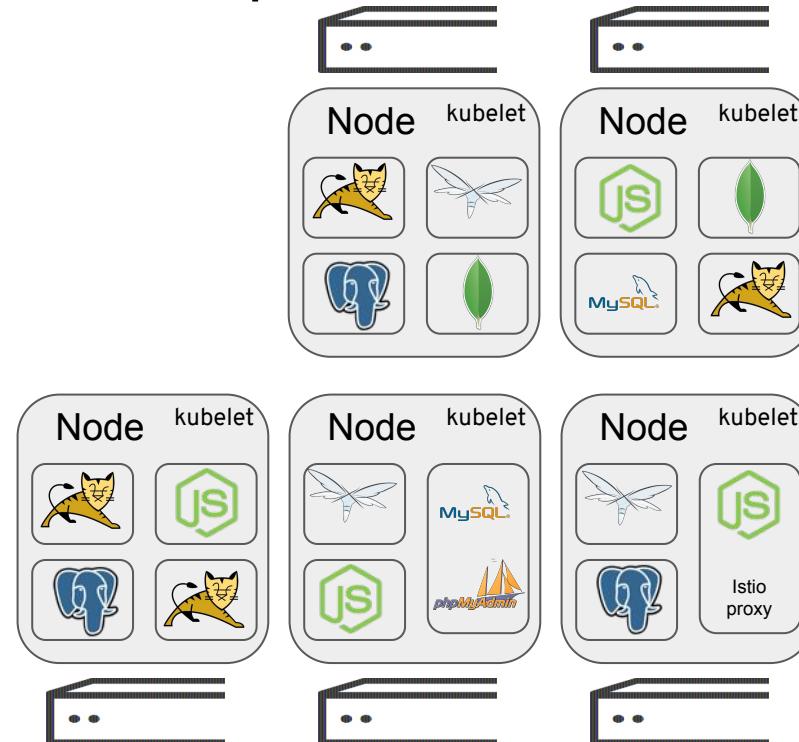
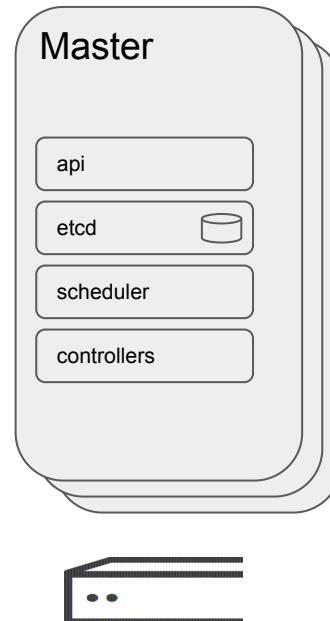
Kubernetes Cluster - Pods Replaced



Dev



Ops



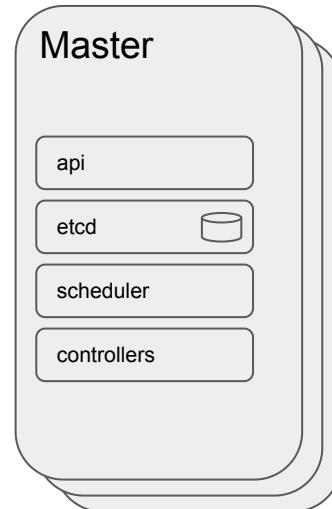
Kubernetes Cluster - New Node



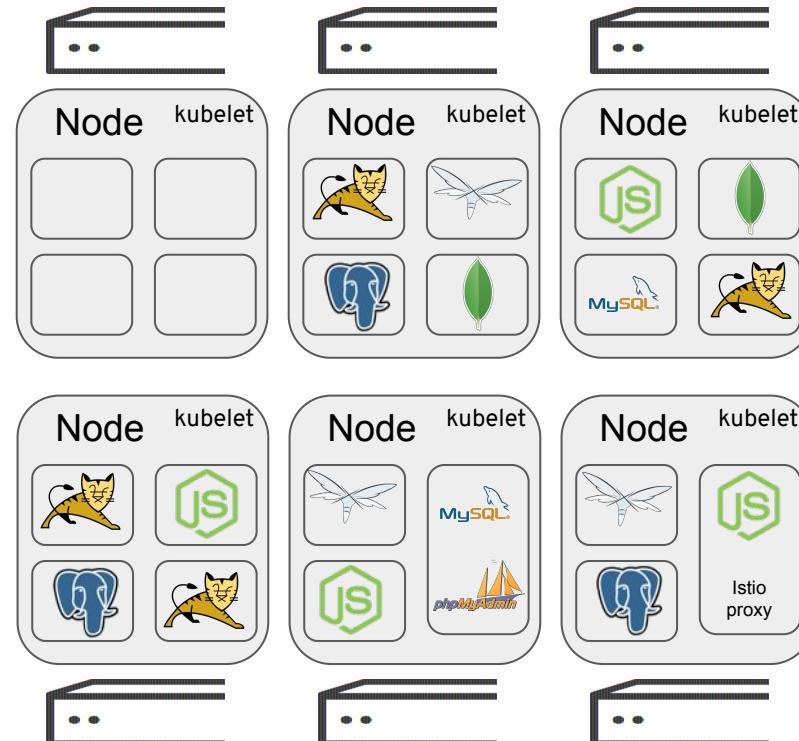
Dev



Ops



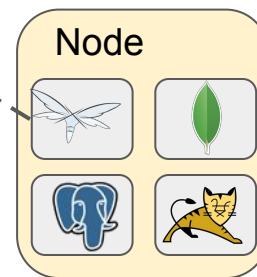
...



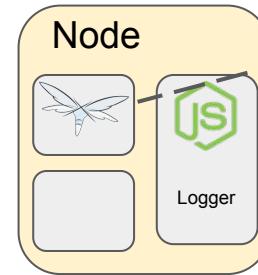
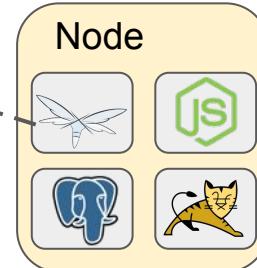
Labels

Labels

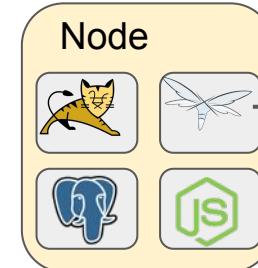
App: Cool
Env: Dev
Version: 1.0



App: Cool
Env: Prod
Version: 1.0

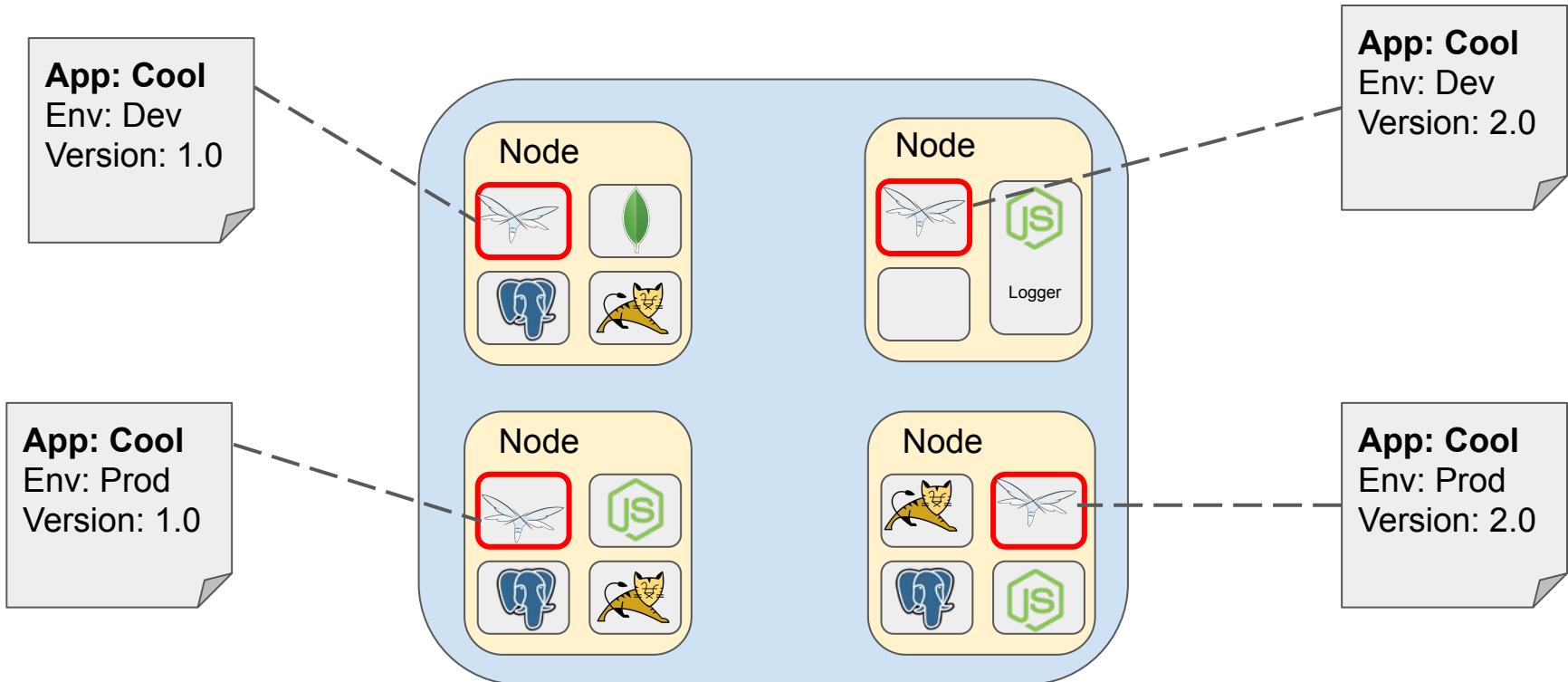


App: Cool
Env: Dev
Version: 2.0

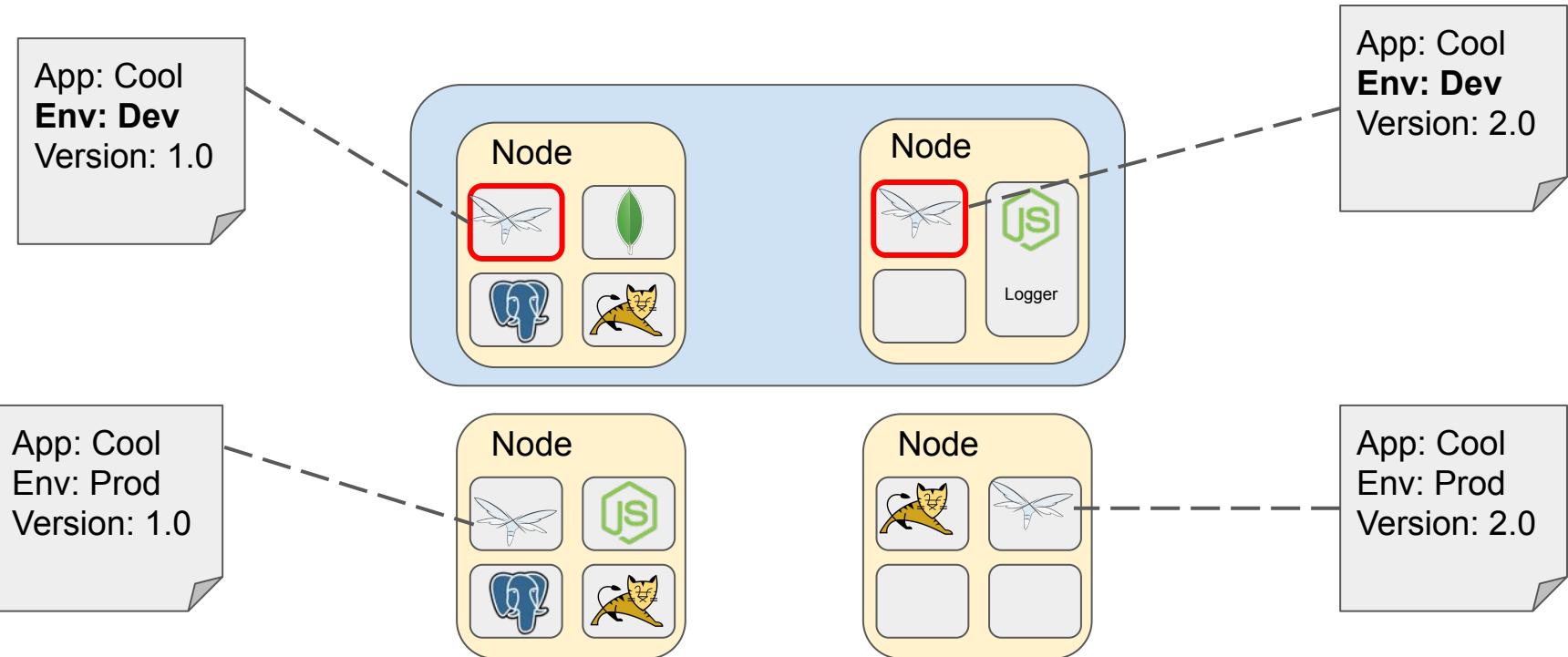


App: Cool
Env: Prod
Version: 2.0

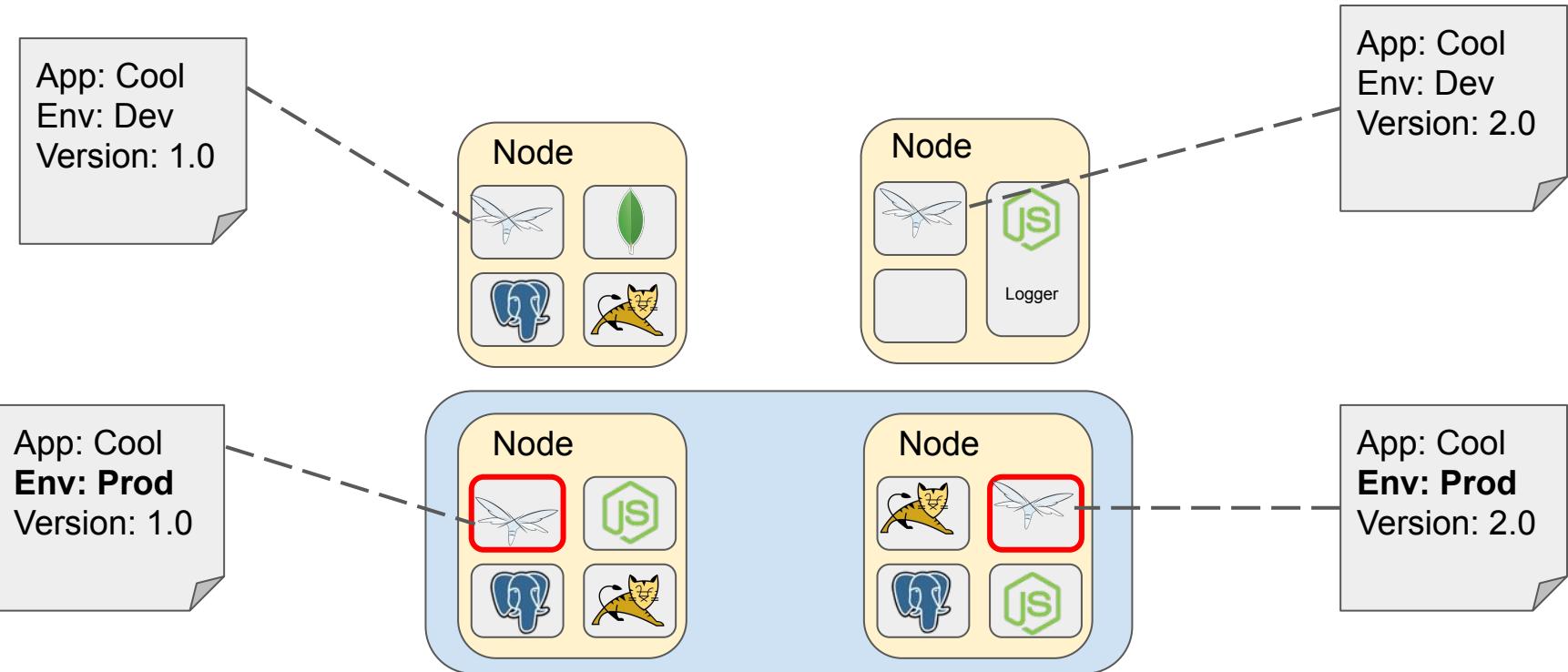
Labels App:Cool



Labels Env:Dev

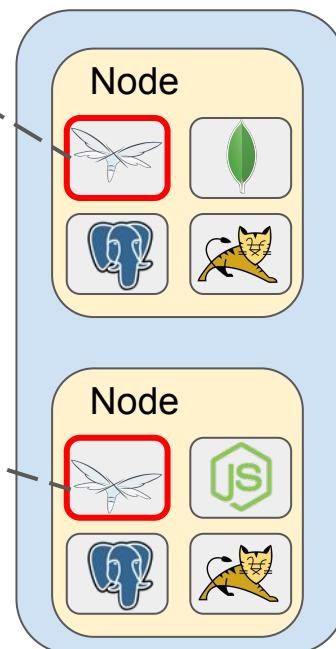


Labels Env:Prod

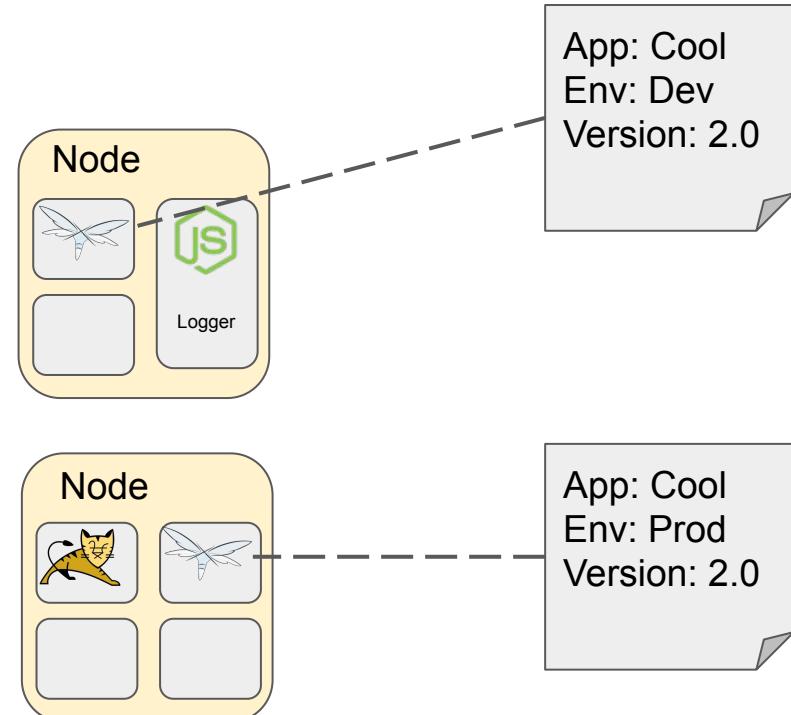


Labels Version:1.0

App: Cool
Env: Dev
Version: 1.0



App: Cool
Env: Prod
Version: 1.0



Installation

<https://redhat-developer-demos.github.io/kubernetes-tutorial/kubernetes-tutorial/installation.html>



Kubernetes Options

Localhost Development

- a. [minikube \(kubectl\) \(docs\)](#)
- b. [Docker for Mac/Windows](#)
- c. [CodeReady Containers](#)
- d. [K3s](#)
- e. [Micro8ks](#)
- f. [KinD](#)

Hosted Kubernetes Cluster

- a. [openshift.com/try from Red Hat](#)
- b. [GKE from Google Cloud Platform](#)
- c. [AKS from Microsoft Azure](#)
- d. [EKS from Amazon Web Services](#)
- e. [IKS from IBM Cloud](#)
- f. [Digital Ocean](#)

[Roll-your-own with kubeadm](#)

[Many others](#)



Developer Tools

- bash: documented commands are based on bash
- docker or podman: build images
- git: pull resources from github
- JDK & Apache Maven: if you wish to compile Java
- curl, tar: downloading and unpacking installations
- Homebrew for Mac: package installation tool
- kubectl: the primary tool you will be using
- minikube: on-laptop, single-node Kubernetes cluster

kubectl



kubectl commands

<https://kubernetes.io/docs/user-guide/kubectl/>

<https://kubernetes.io/docs/reference/kubectl/cheatsheet>

kubectl get namespaces

kubectl get pods -n mynamespace

kubectl run myboot --image=quay.io/burrsutter/myboot:v1 --port=8080

kubectl logs myboot-75f94d76fc-pl2l7

kubectl expose deployment --port=8080 myboot --type=LoadBalancer

kubectl scale deployment myboot --replicas=3

kubectl set image deployment myboot myboot=quay.io/burrsutter/myboot:v2

kubectl Exercises

<https://redhat-developer-demos.github.io/kubernetes-tutorial/kubernetes-tutorial/kubectl.html>

Pod, ReplicaSet, Deployment

Pod, ReplicaSet, Deployment

Exercises

<https://redhat-developer-demos.github.io/kubernetes-tutorial/kubernetes-tutorial/pod-rs-deployment.html>

Service

Service Exercises

<https://redhat-developer-demos.github.io/kubernetes-tutorial/kubernetes-tutorial/service.html>

Logs



logs

```
System.out.println("Java, where am I?");  
Or console.log("Node logs");
```

```
kubectl get pods  
kubectl logs microspringboot1-2-nz8f8  
kubectl logs microspringboot1-2-nz8f8 -p # last failed pod
```

```
OR stern (brew install stern)  
https://github.com/wercker/stern  
stern myboot
```

```
OR kail (https://github.com/boz/kail)  
kail -d myboot
```

Logs Exercises

<https://redhat-developer-demos.github.io/kubernetes-tutorial/kubernetes-tutorial/logs.html>

Resource Limits

Resource Requests/Limits

```
resources:  
  requests:  
    memory: "300Mi"  
    cpu: "250m" # 1/4 core  
  
  limits:  
    memory: "400Mi"  
    cpu: "1000m" # 1 core
```

Resource Limits Exercises

<https://redhat-developer-demos.github.io/kubernetes-tutorial/kubernetes-tutorial/resources.html>

exec



kubectl exec

"ssh" into your containers and explore

```
kubectl get pods --namespace=microworld  
kubectl exec -it --namespace=microworld $POD cat /sys/fs/cgroup/memory/memory.limit_in_bytes
```

Or

```
kubectl exec -it --namespace=microworld microspringboot1-2-nz8f8 /bin/bash  
ps -ef | grep java
```

Note: the following apply if using the fabric8 generated image, otherwise consult your Dockerfile

```
java -version
```

```
javac -version
```

```
# now find that fat jar
```

```
find / -name *.jar
```

```
cd /deployments (based on use of the fabric8 maven plugin)
```

```
ls
```

```
exit
```

exec Exercises

<https://redhat-developer-demos.github.io/kubernetes-tutorial/kubernetes-tutorial/exec.html>

Rolling Updates

Rolling Update Exercises

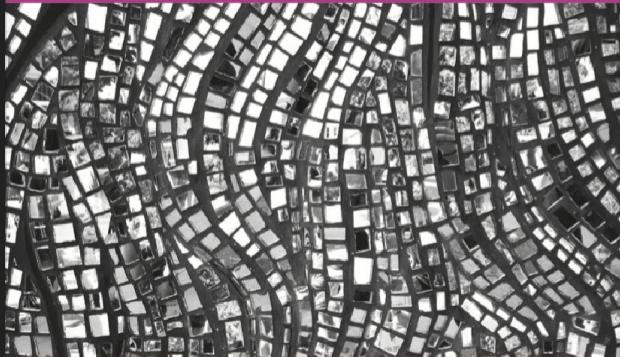
<https://redhat-developer-demos.github.io/kubernetes-tutorial/kubernetes-tutorial/rolling-updates.html>

Free Resources

O'REILLY®

Migrating to Microservice Databases

From Relational Monolith
to Distributed Data



Edson Yanaga

Compliments of
Red Hat
Developer

 Red Hat
Developer

 DevNation

[Download](#)

<http://bit.ly/mono2microdb>

O'REILLY®

Compliments of
Red Hat
Developer

Introducing Istio Service Mesh for Microservices

Build and Deploy Resilient, Fault-Tolerant
Cloud Native Applications



Second
Edition

Burr Sutter & Christian Posta

 Red Hat
Developer |  DevNation

[Download](#)

bit.ly/istibook

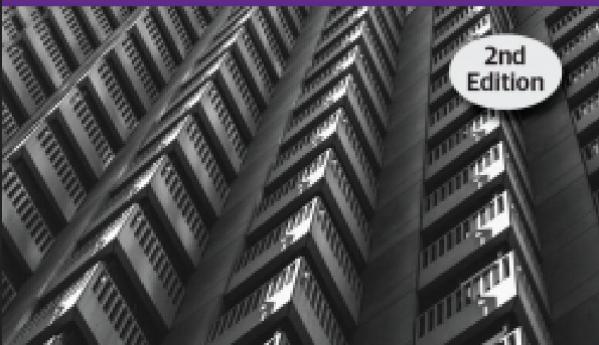
O'REILLY®

Compliments of
 Red Hat
Developer

Microservices for Java Developers

A Hands-On Introduction
to Frameworks & Containers

2nd
Edition



Rafael Benevides & Christian Posta



Red Hat
Developer

 DevNation

[Download](#)

bit.ly/javamicroservicesbook

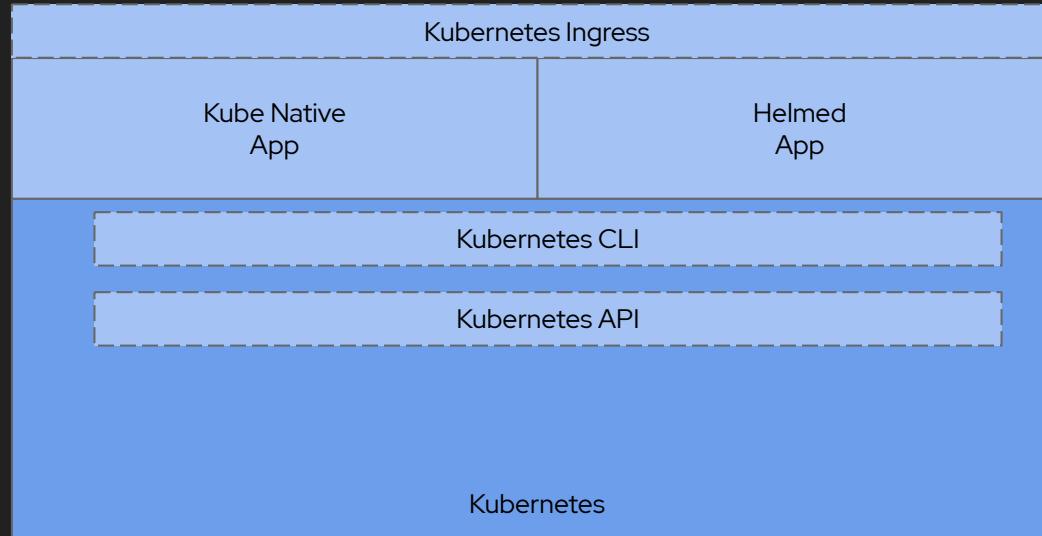
Coming Soon

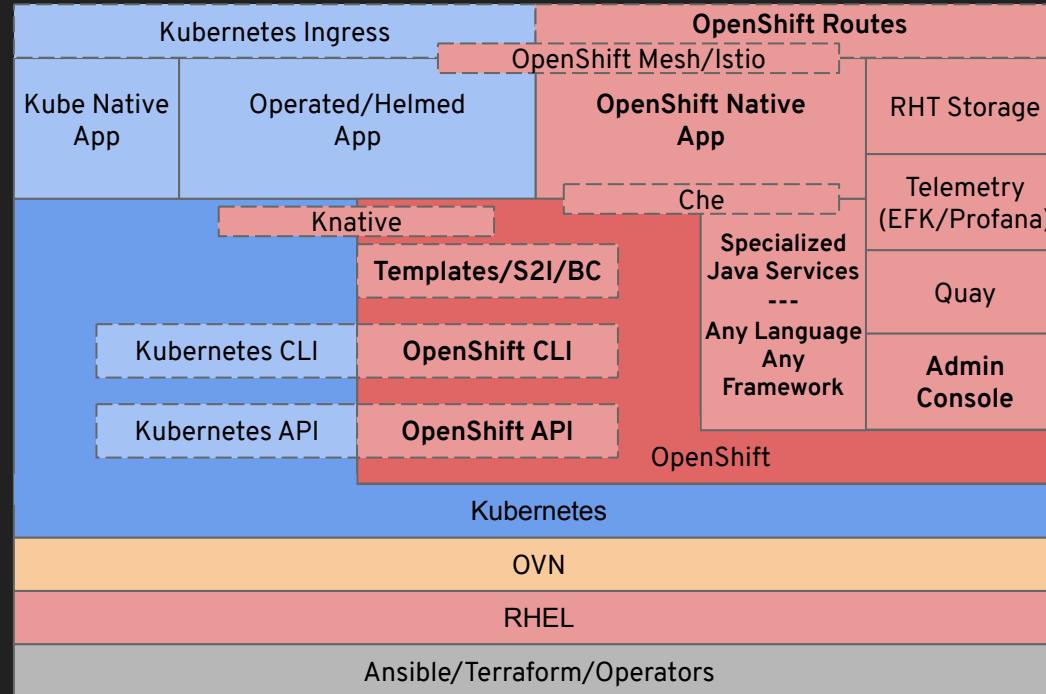


The End

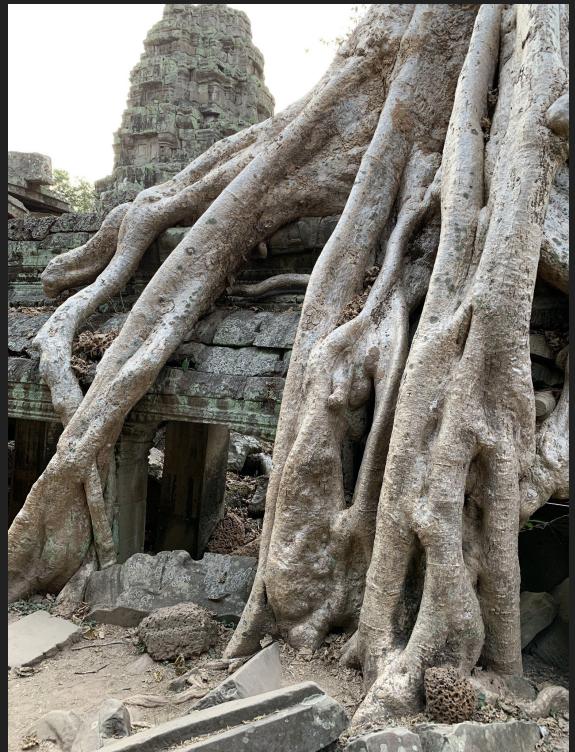
Backup Content

| | Kubernetes | OpenShift |
|--|------------|-----------|
| Multi-container, multi-host scheduling | ✓ | ✓ |
| Self-service provisioning | ✓ | ✓ |
| Service discovery | ✓ | ✓ |
| Persistent Storage | ✓ | ✓ |
| Multi-tenancy | | ✓ |
| Networking (SDN) | | ✓ |
| Image Registry | | ✓ |
| Image Build Tools | | ✓ |
| Metrics | | ✓ |
| Log Aggregation | | ✓ |
| Ingress | | ✓ |
| UX: Console, Service Catalog, "oc" | | ✓ |
| Secured by Default | | ✓ |

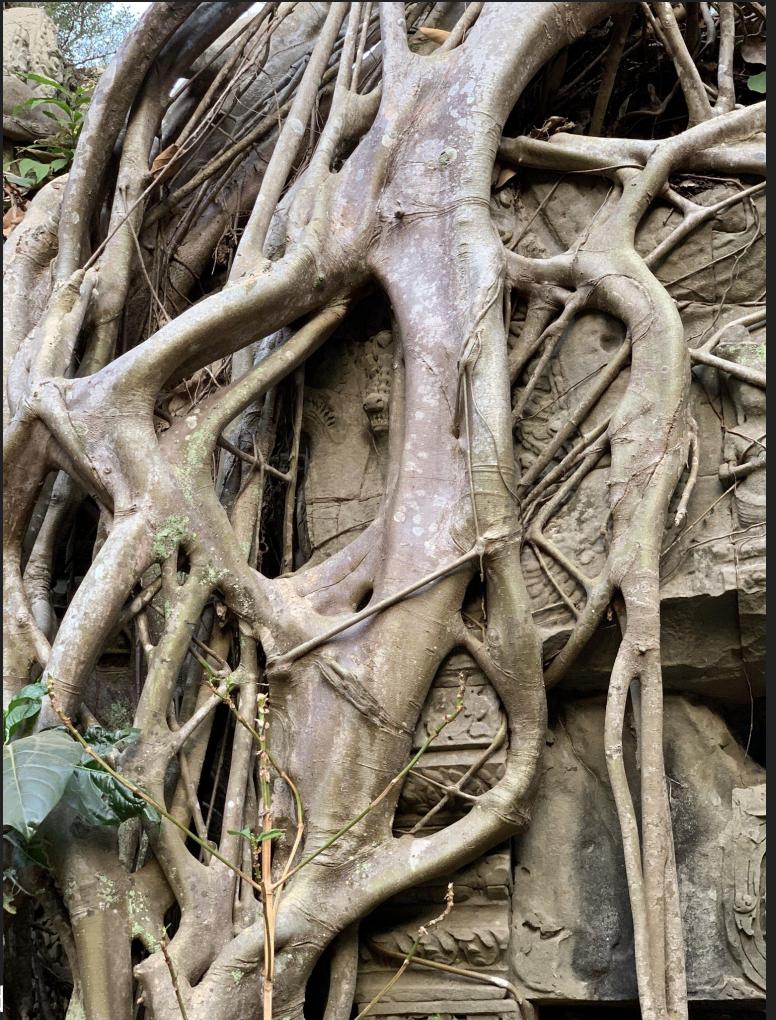




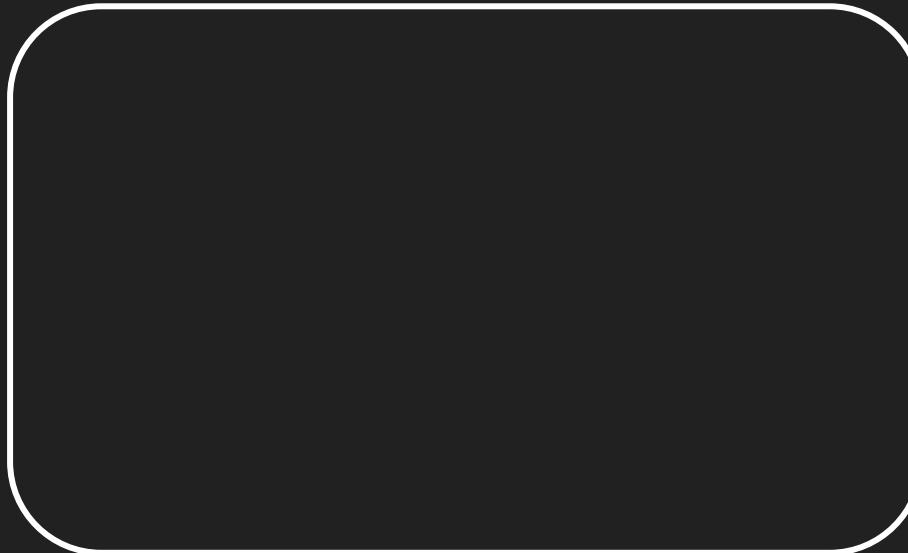
Strangler Strategy



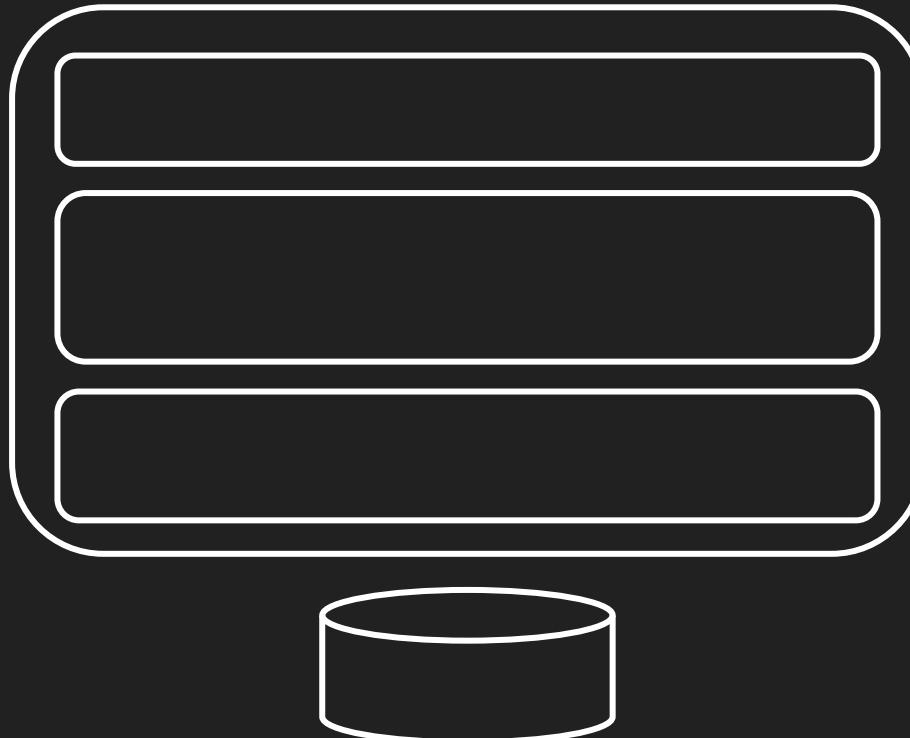
dn.15, Kambodscha



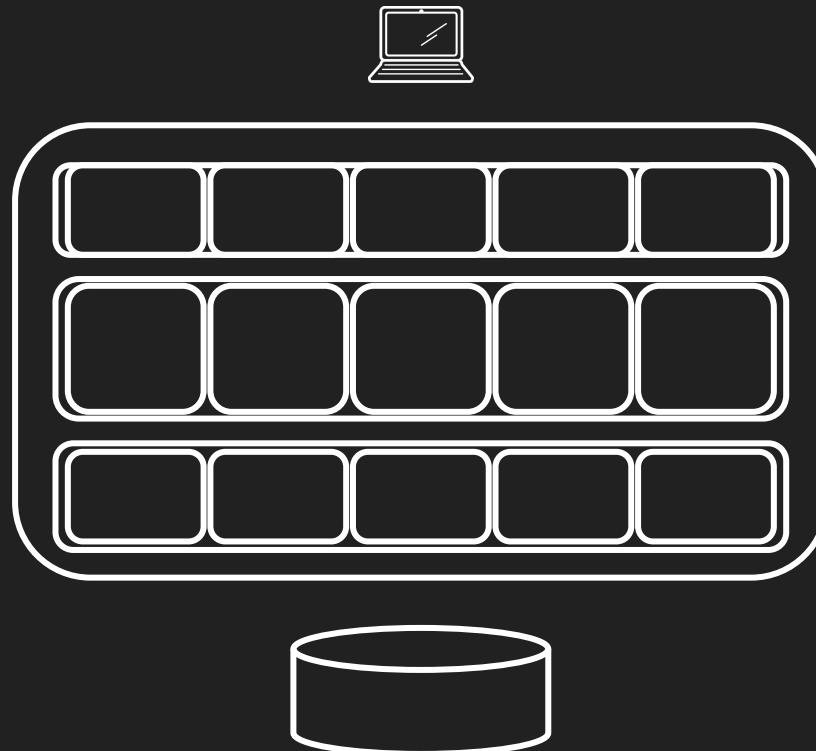
Application on Whiteboard



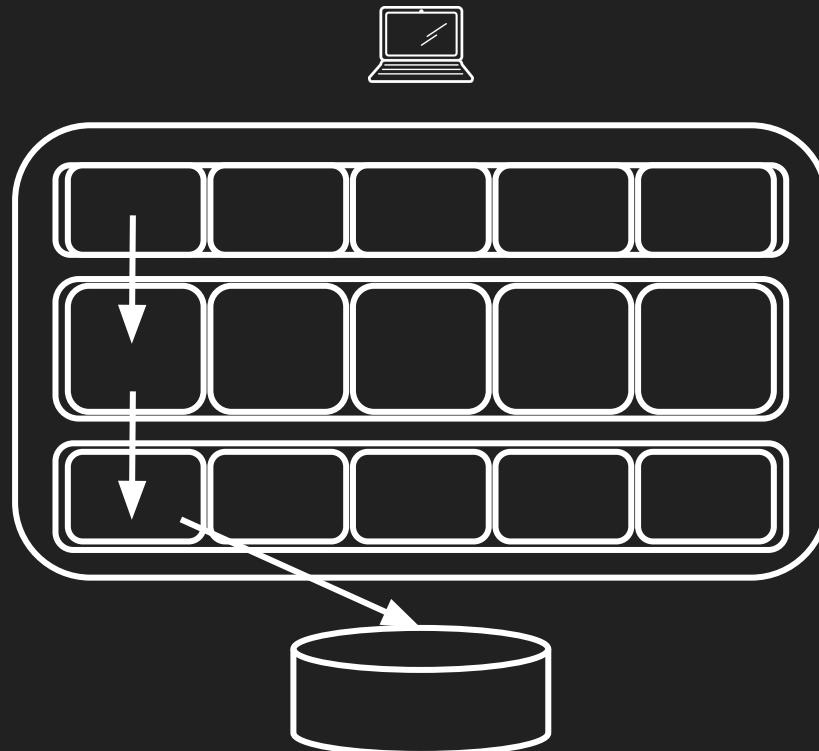
Whiteboard had 3 Tiers



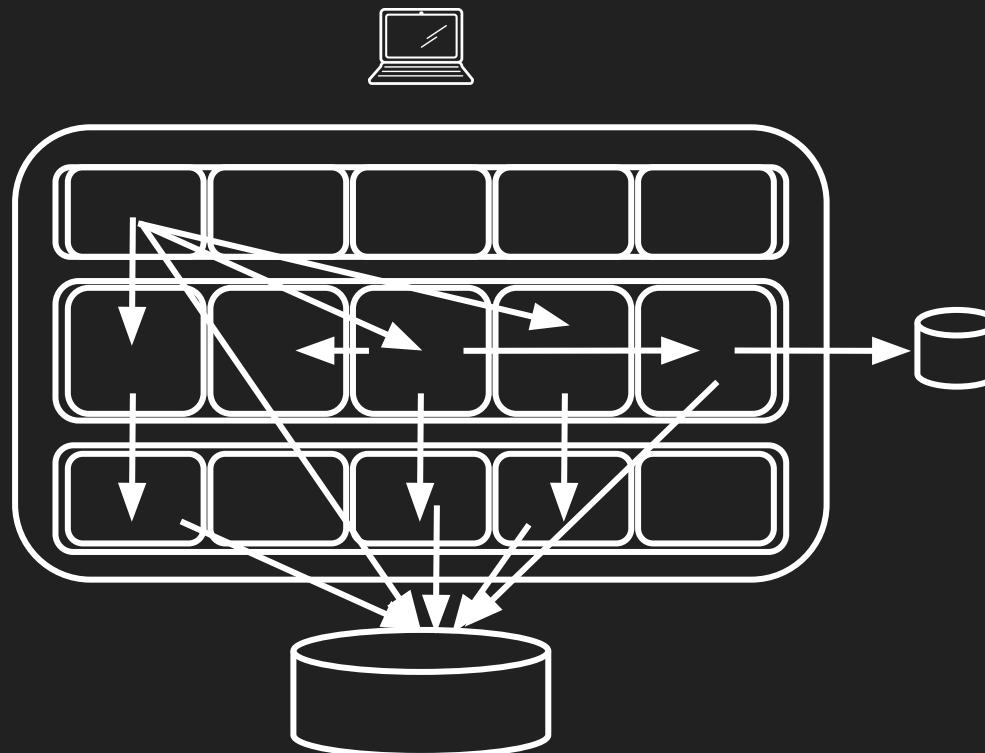
UI, Logic, Data



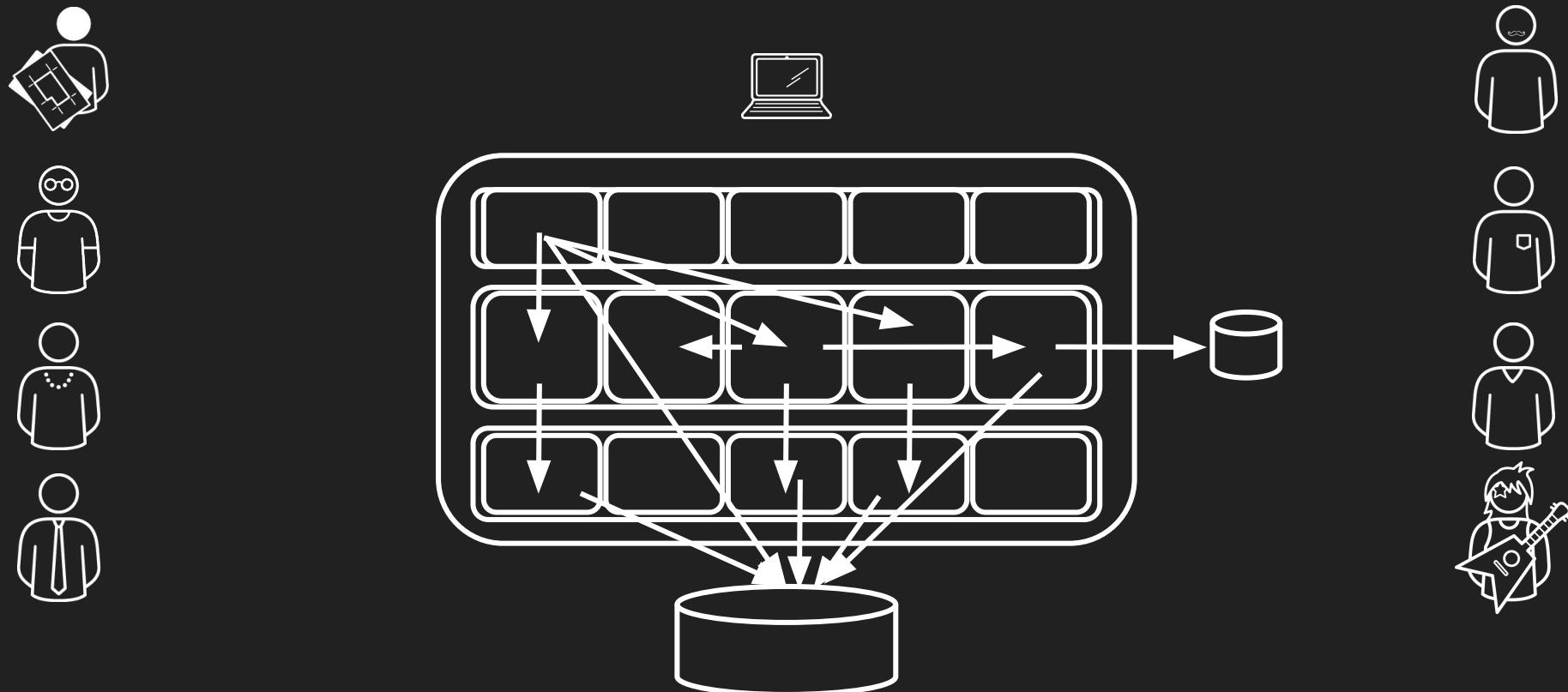
Clean Architecture



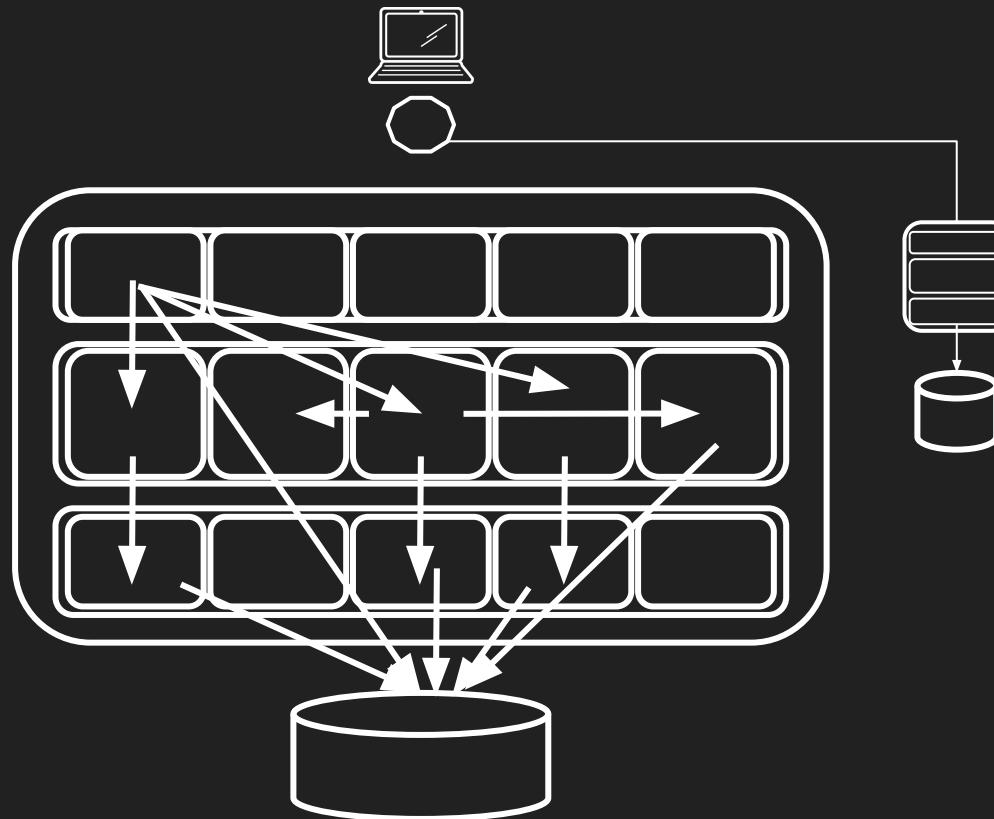
Real Life: Non-Majestic Monolith



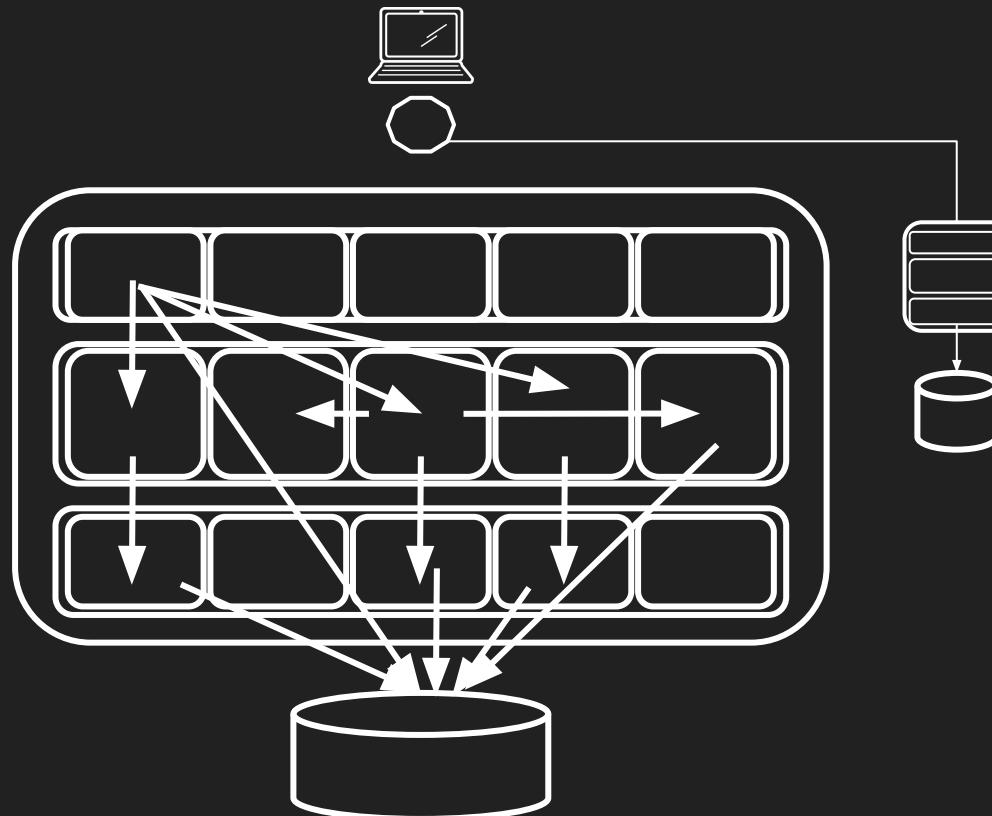
Large Team



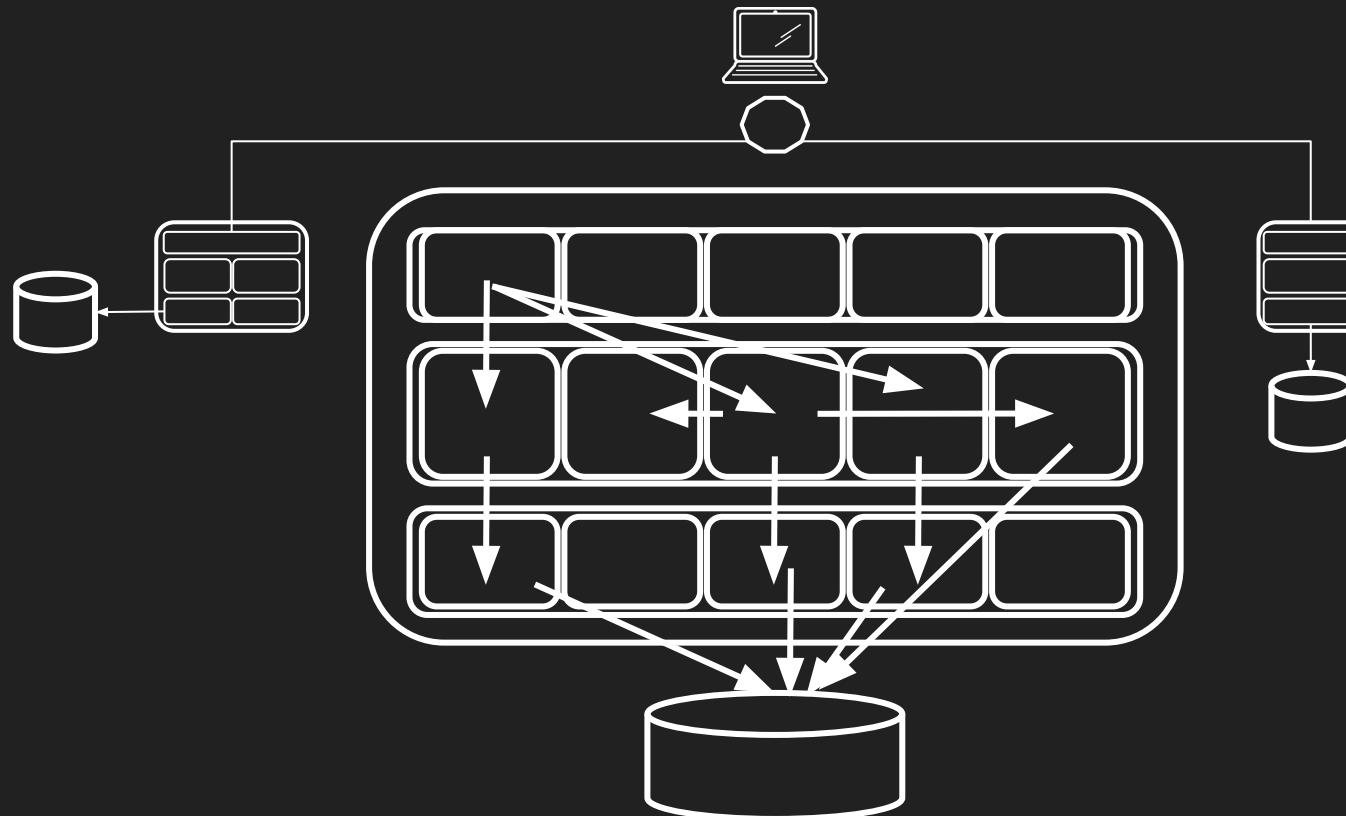
Strangle



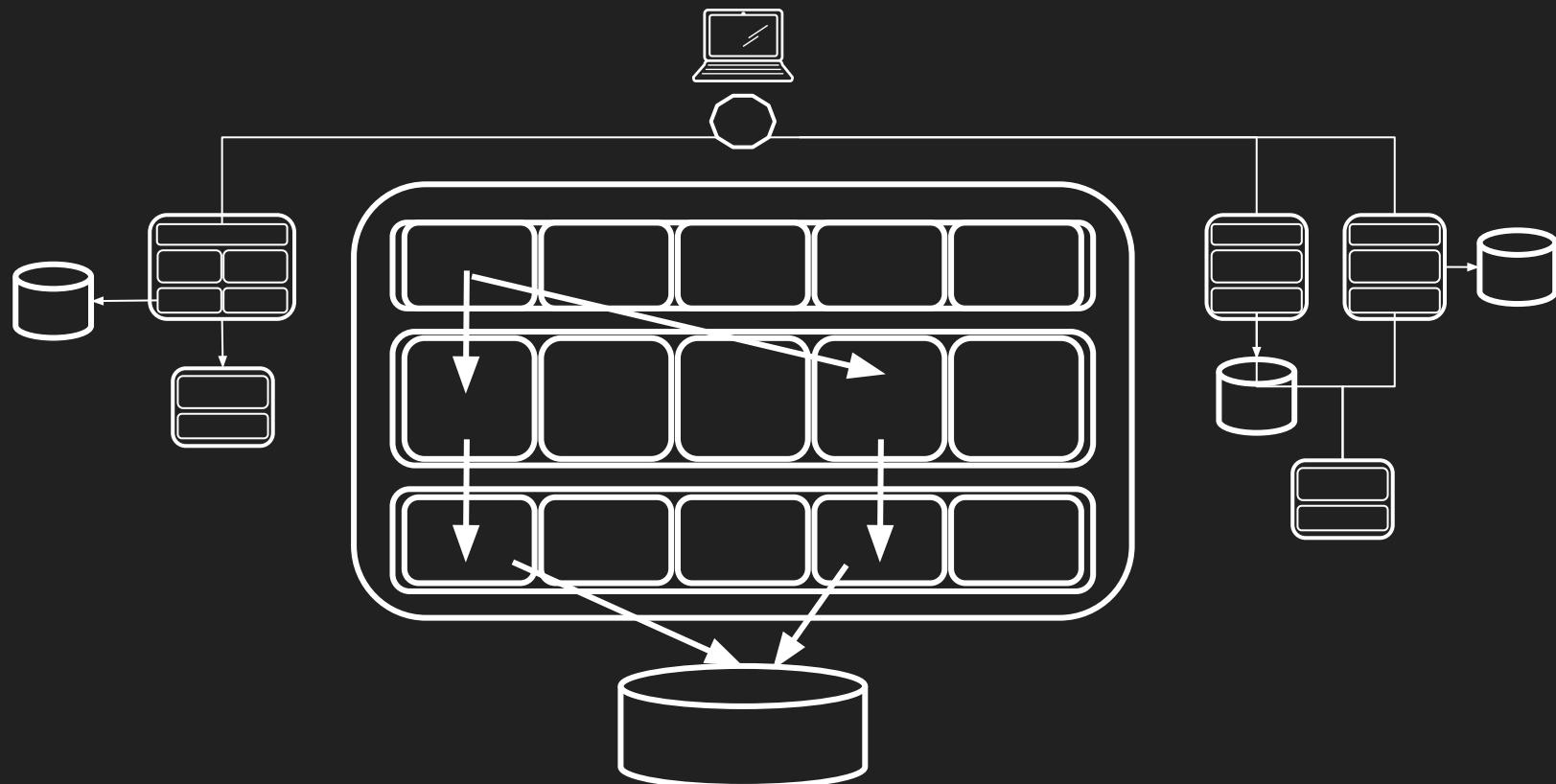
Strangle Hug



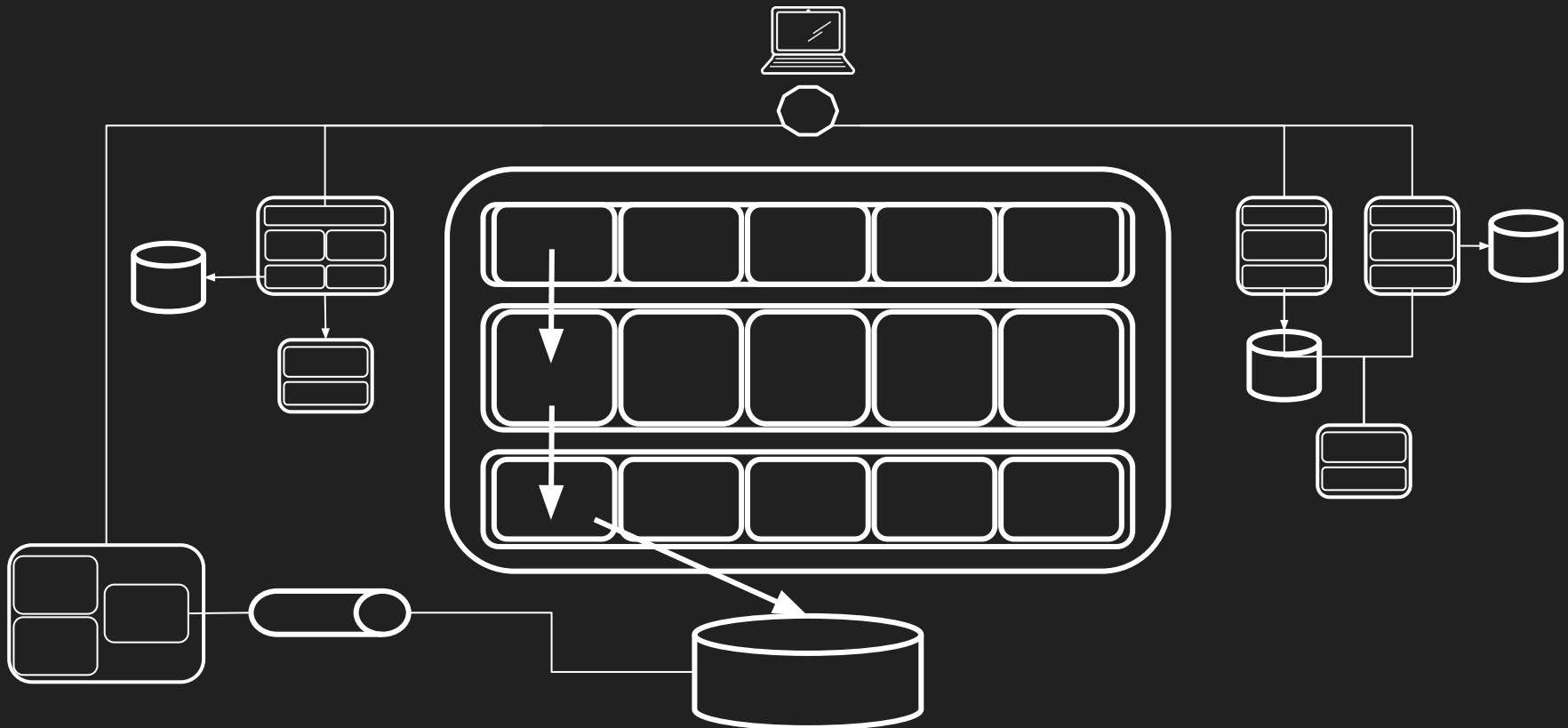
Friend Hug



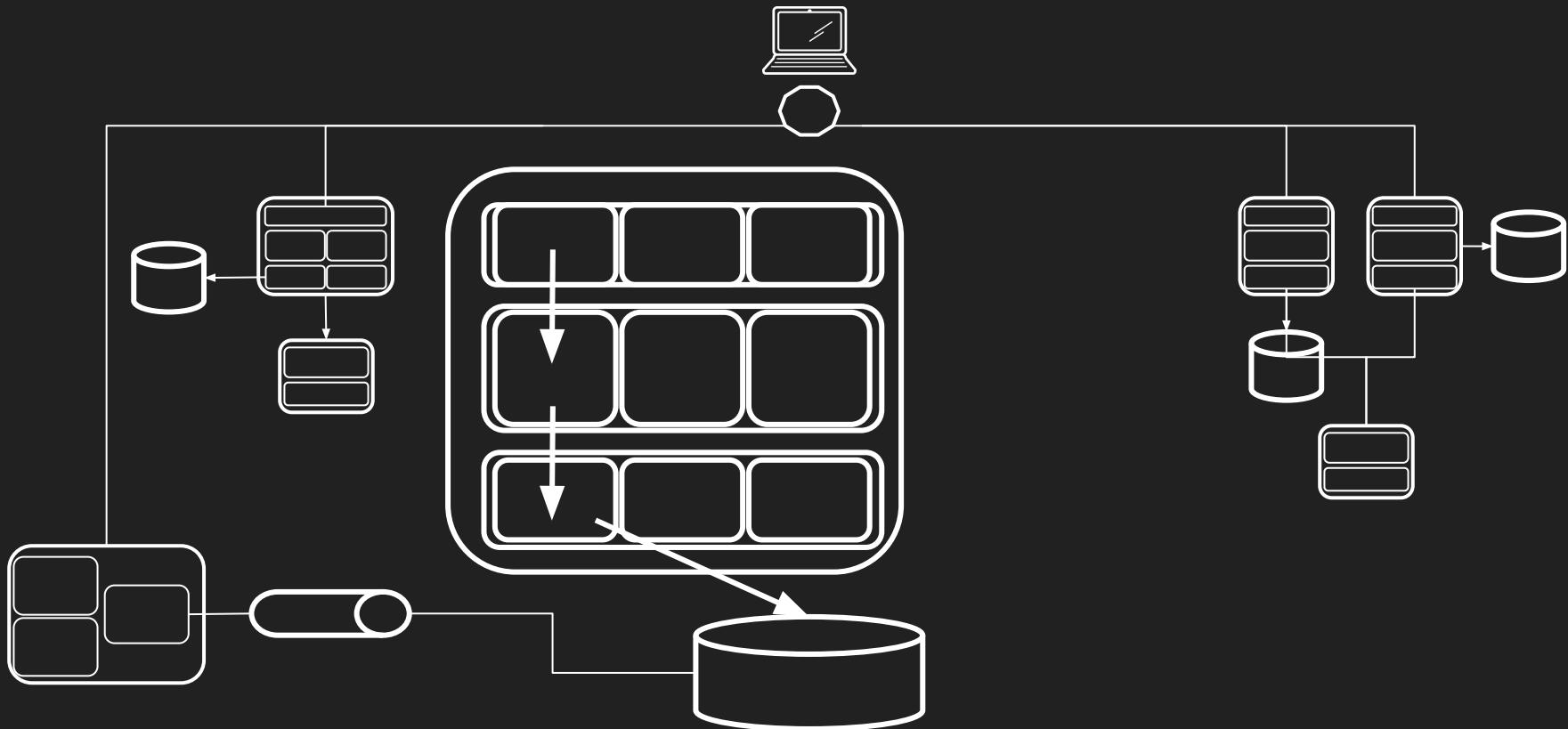
Family Hug



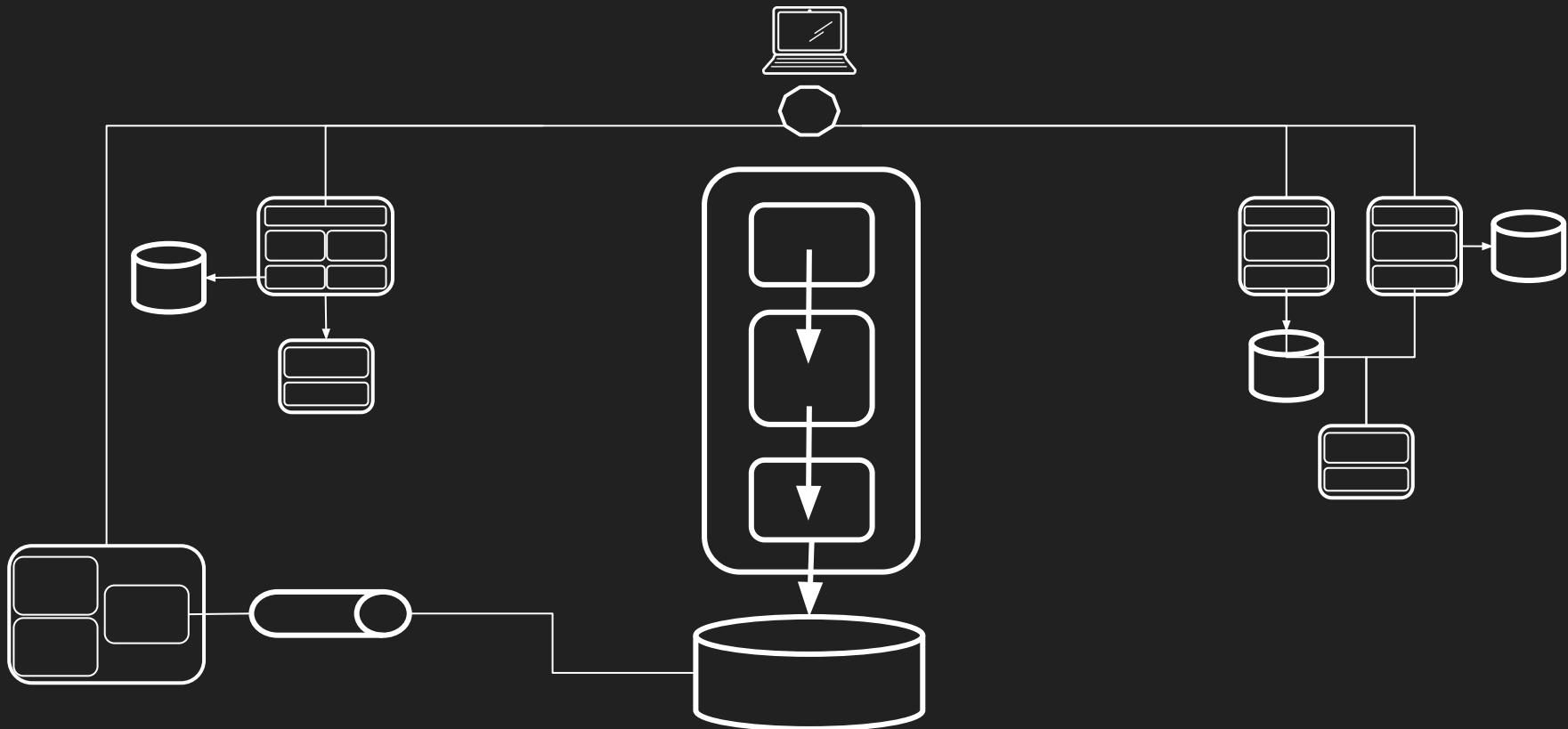
Bear Hug



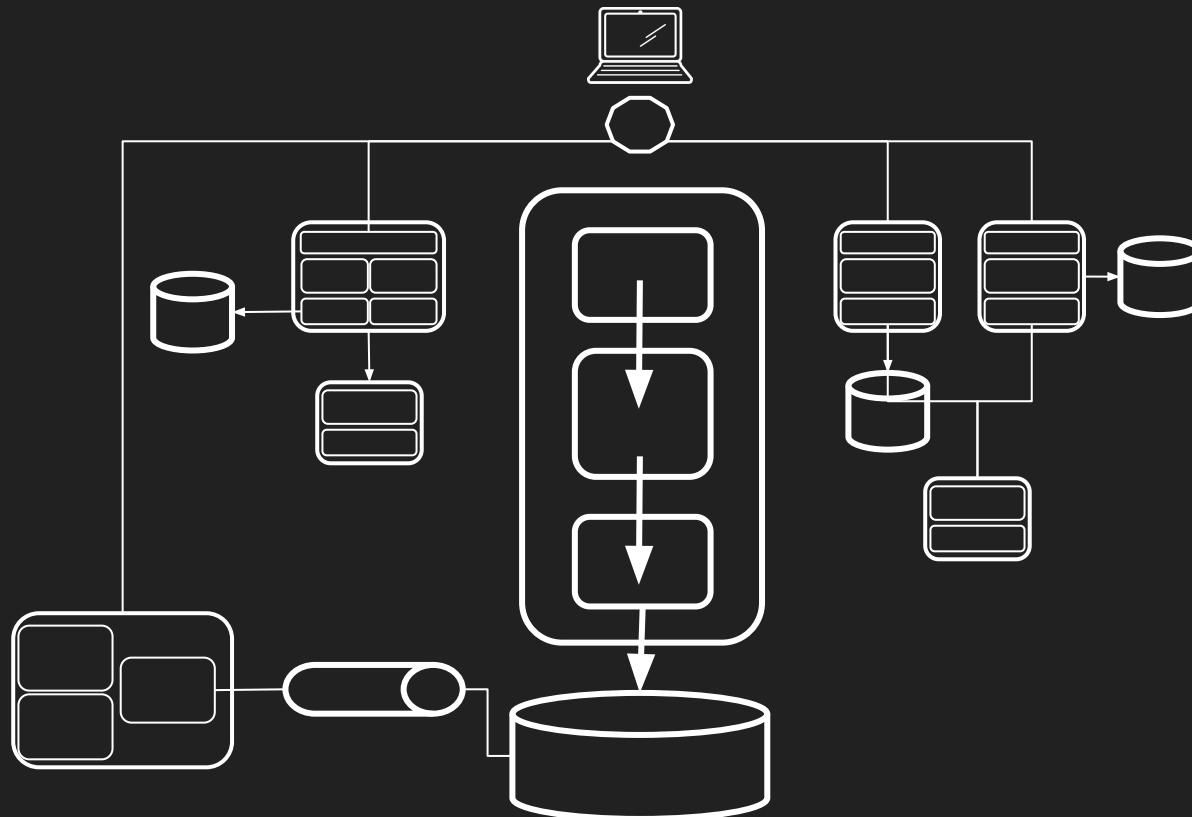
"Refactor"



Squeeze



Streamline



Embrace Others

