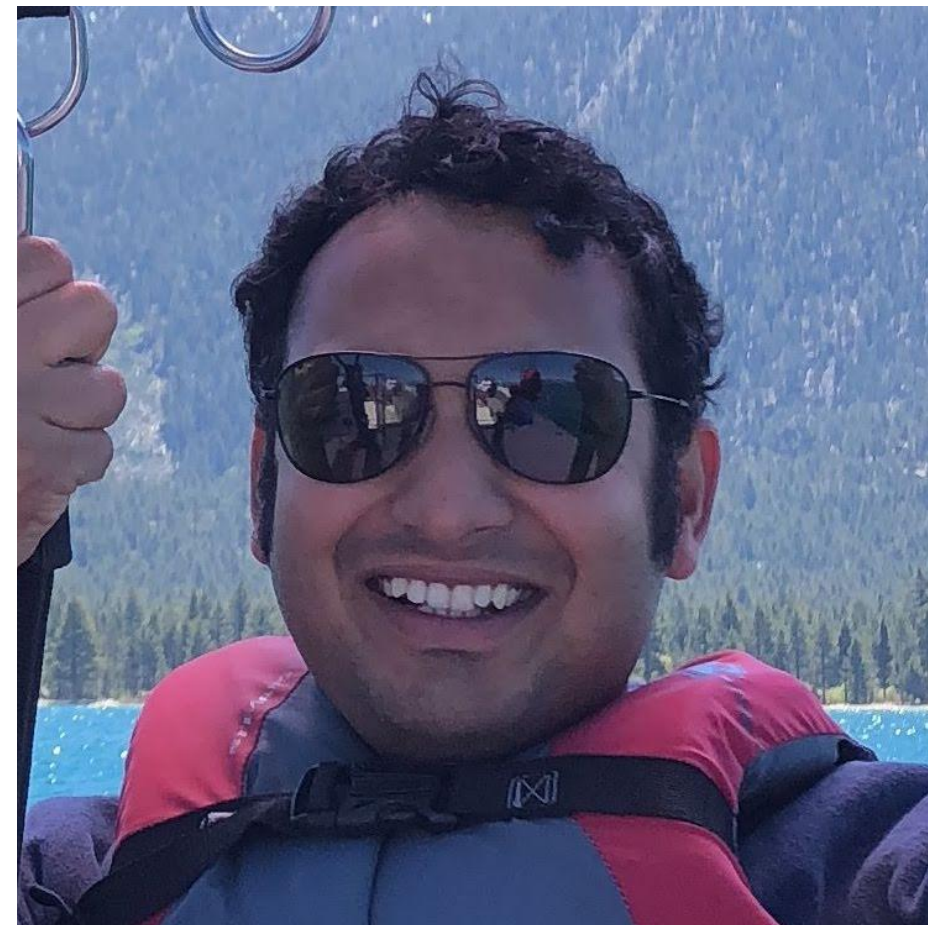


Docker on Docker: Leveraging Kubernetes In Docker EE



Brett Inman

Senior Infrastructure Engineer, Docker



Manish Tomar

Senior Software Engineer, Docker

Using Docker EE at Docker Inc

So?

- **We are a customer**
 - SaaS and internal workloads
- **Sharing what we've learned**
 - Best practices, tips
- **Dogfooding**
 - Better product

Agenda

- Background
- Planning Process for EE2.0
- Infrastructure Preparation
- Application Migration
- EE and Kube Features

Acronyms

- EE - Docker Enterprise Edition
- UCP - Universal Control Plane
- DTR - Docker Trusted Registry
- SaaS - Software as a Service, e.g. Docker Hub

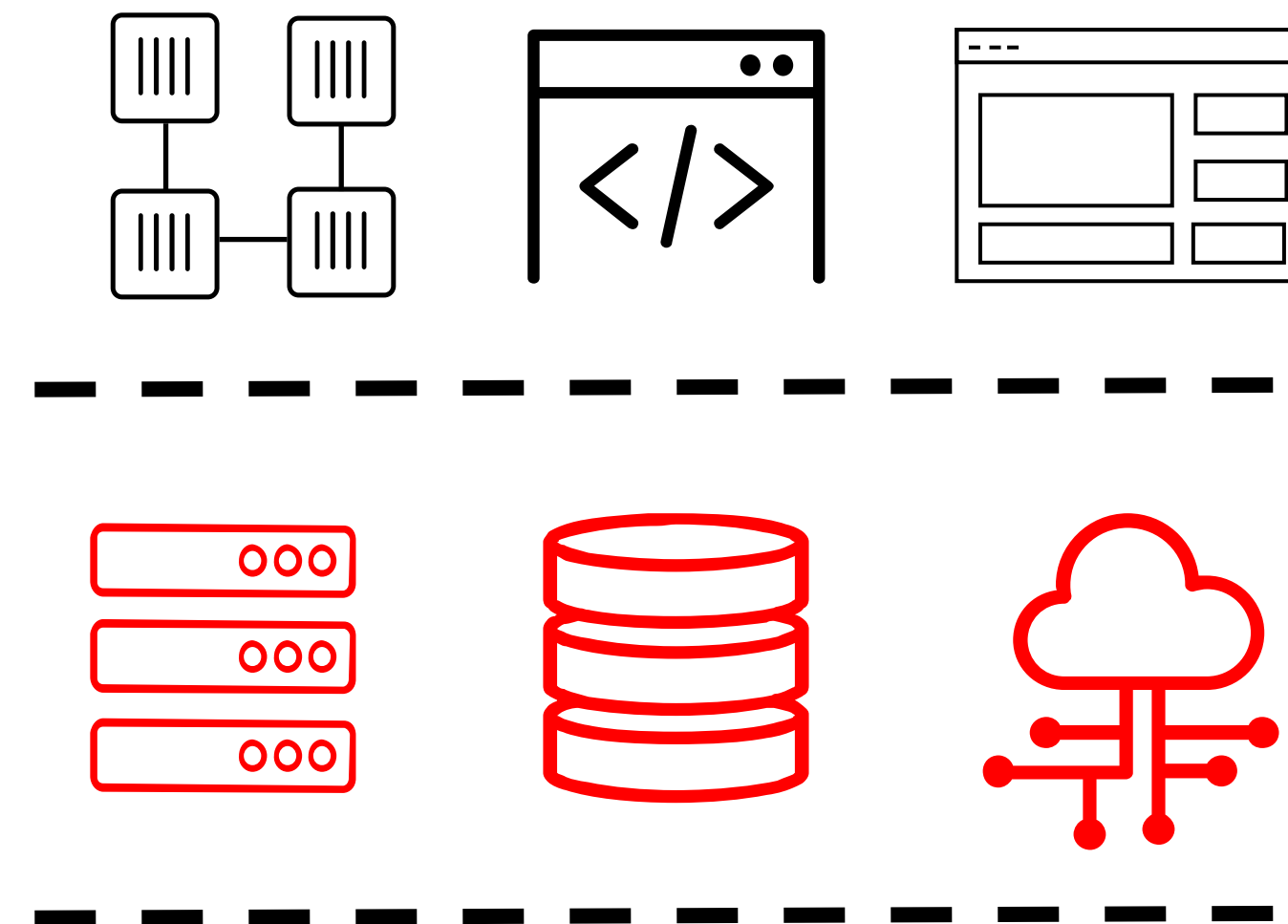
Disclaimer

Running on EE

- >200 hosts in production on EE
- >100 microservices
- Hub/Store/Cloud
 - 1 billion pulls every ~2 weeks
- Web properties (www, blog, etc)
- Build/CI systems

Role of Infrastructure

Provide a self-service
container platform.



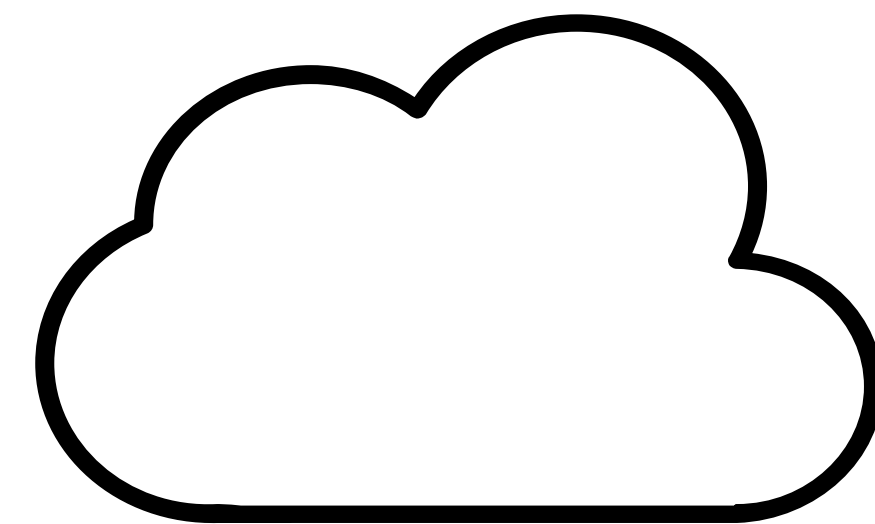
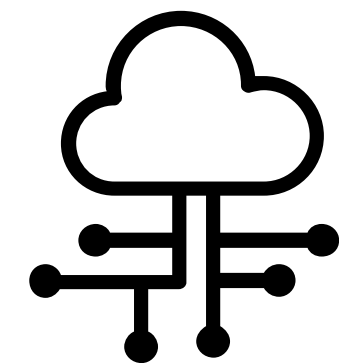
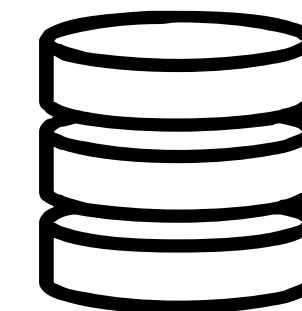
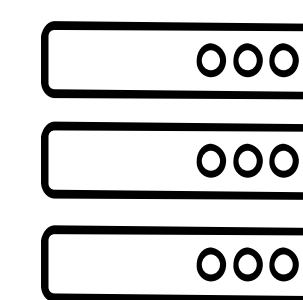
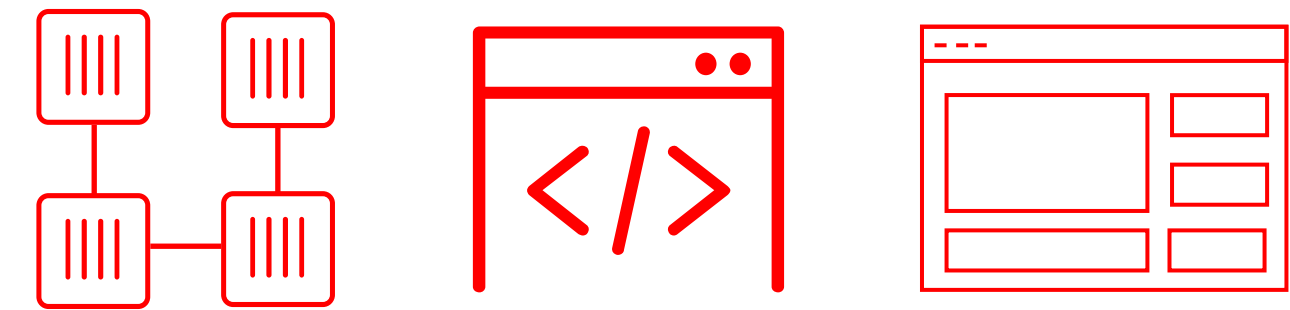
Role of Infrastructure

Interfaces for our users

- Docker EE - Swarm and Kubernetes
- Logging
- Metrics and Monitoring

Role of Dev

Use the platform to run applications and provide services.



Dogfooding

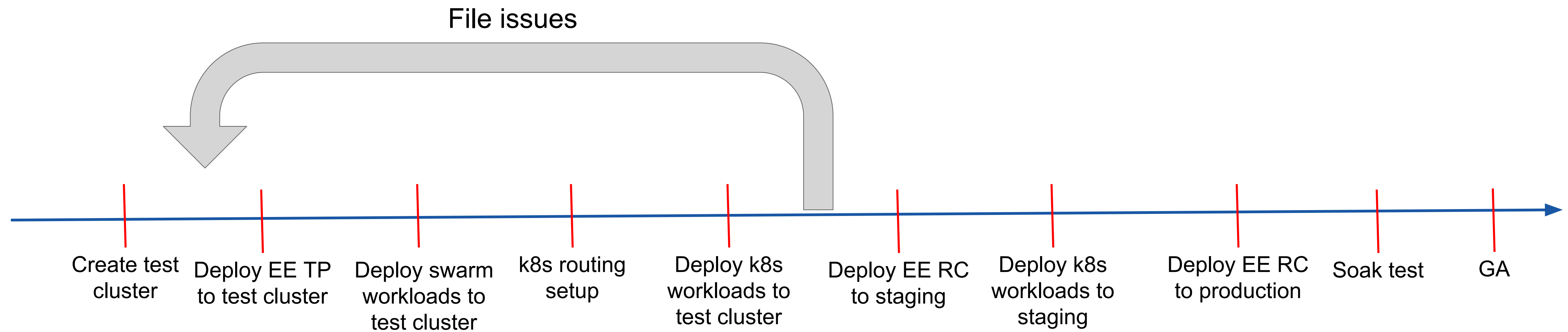
- Goal: provide real-world feedback pre-release
- Cross-functional
- Started in Summer 2016
- EE 2.0 - focus on Kubernetes

Planning for EE 2.0 / Kubernetes

Planning

- Milestones - assign owners and dates
- Weekly syncs
- Common slack channel #dogfooding
- Release blocker

Planning - milestones



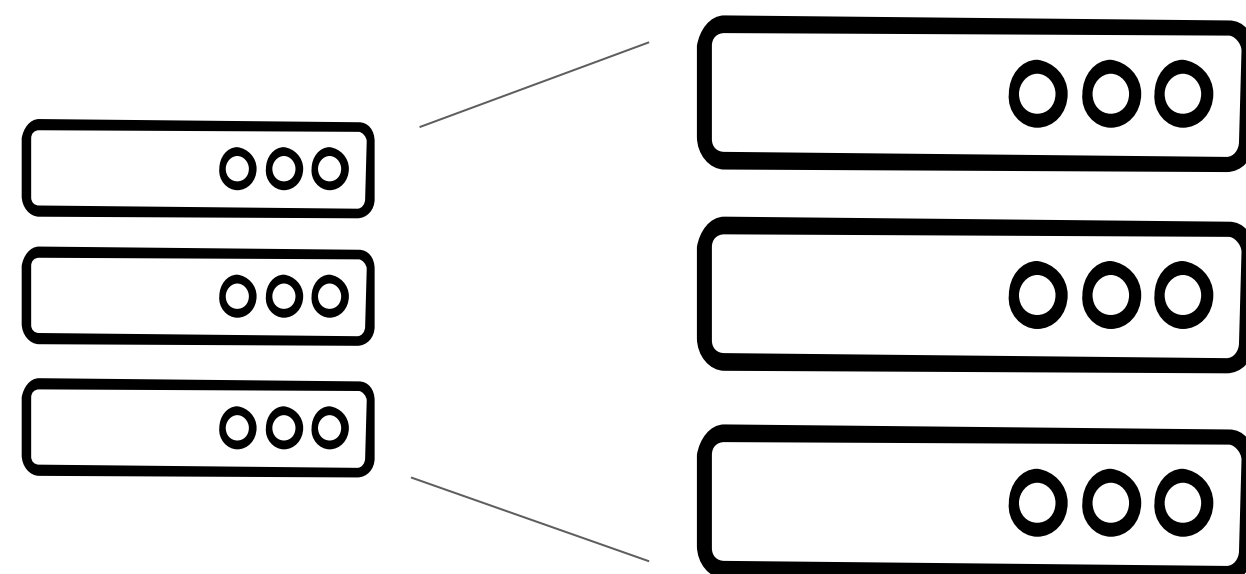
Infra prep for EE2 and Kubernetes

Test cluster

- Smaller clone of existing environments
- Make modifications to infra code
- PR changes back into stage/prod

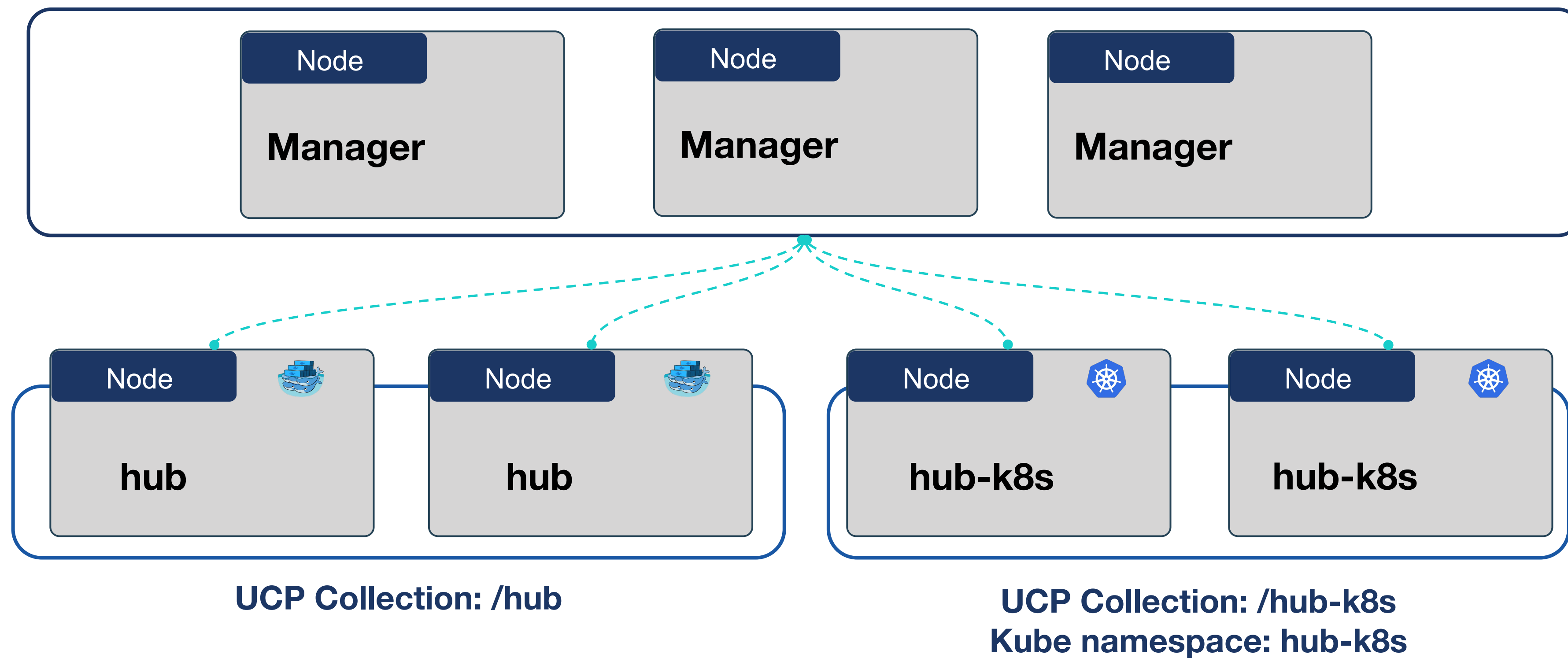


Sizing



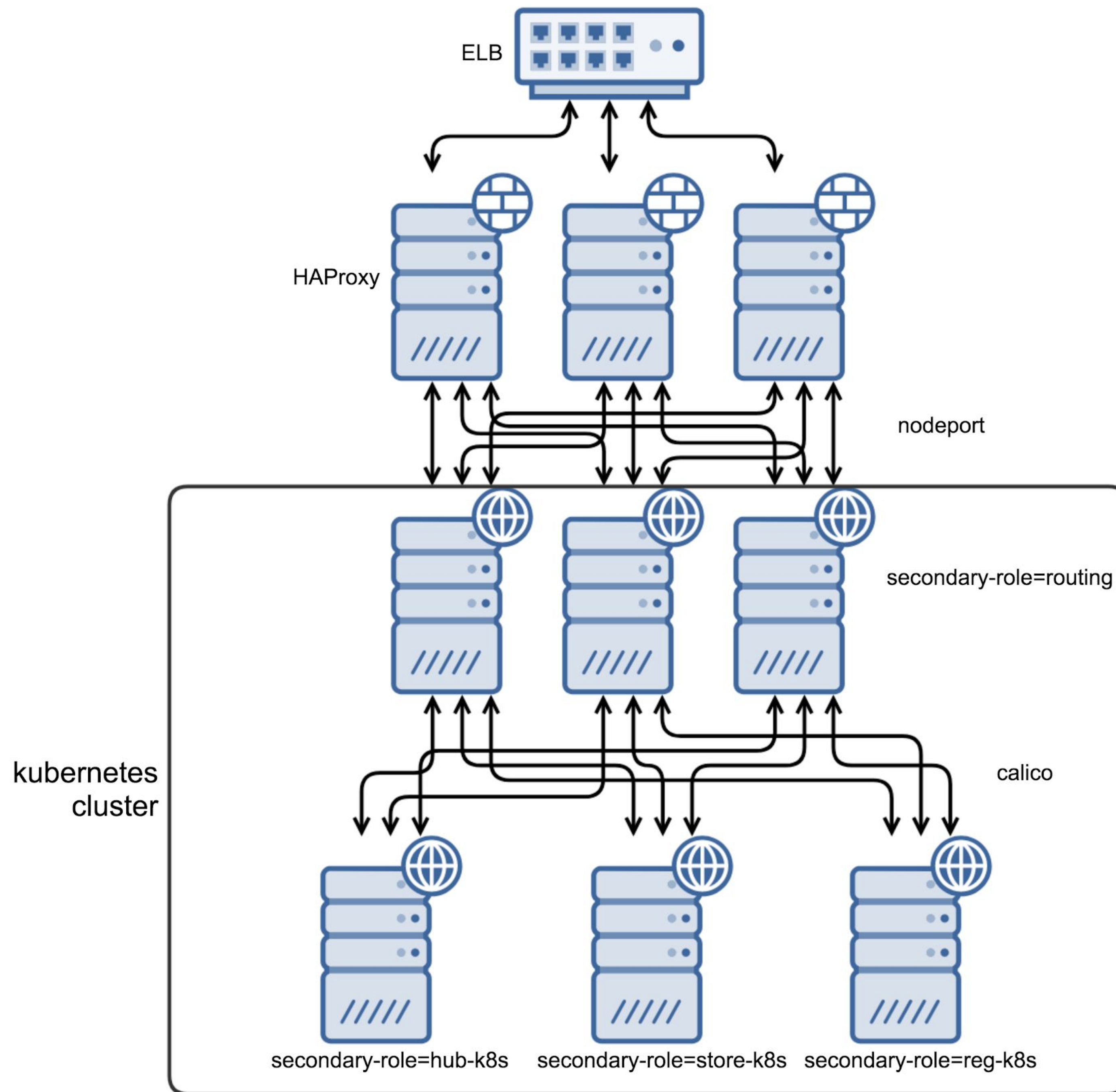
- UCP Managers now running kube containers
- Minimum: 8GB
- Recommended: 16GB

Hosts per orchestrator



Networking

- Calico
- Kubernetes concepts
 - Ingress Controllers
 - early, limited customization
 - Nodeport
 - simple building block



hub.docker.com:443



{ip_haproxy_node_x}:4321



nodeport
service

{ip_routing_node_y}:32775



Calico

{calico_ip_hub_pod_z}:80

Metrics

- Prometheus

- every in-house application exposes a /metrics endpoint
- exporters for third-party applications

- Service discovery

- consul
- kube api

Kubernetes service discovery in Prometheus

Prometheus scrape config

scrape_configs:

- job_name: 'kubernetes-pods'

kubernetes_sd_configs:

- role: pod

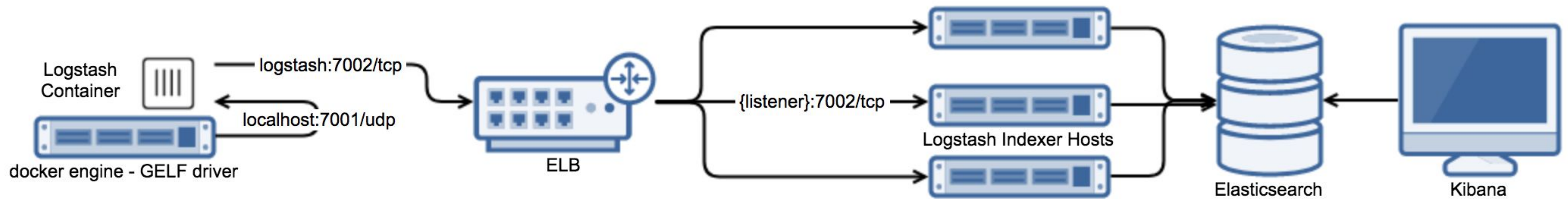
api_server: https://ucp.{{ env "STACK" }}.domain.io:6443

bearer_token: {{ KUBE_TOKEN }}

scheme: https

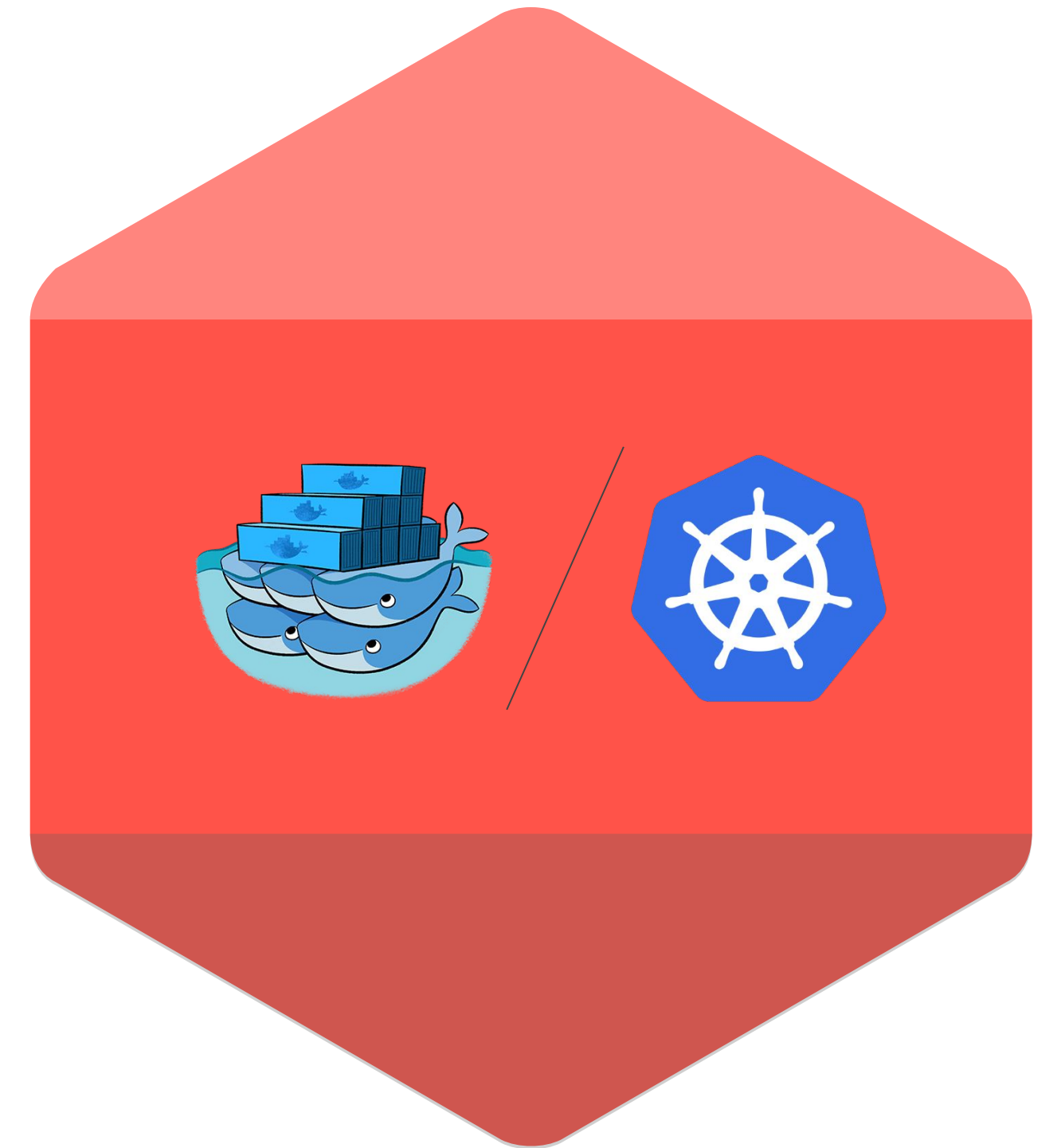
Logging

Logging happens at the engine level - no change



Upgrade and Migration

Upgrading UCP

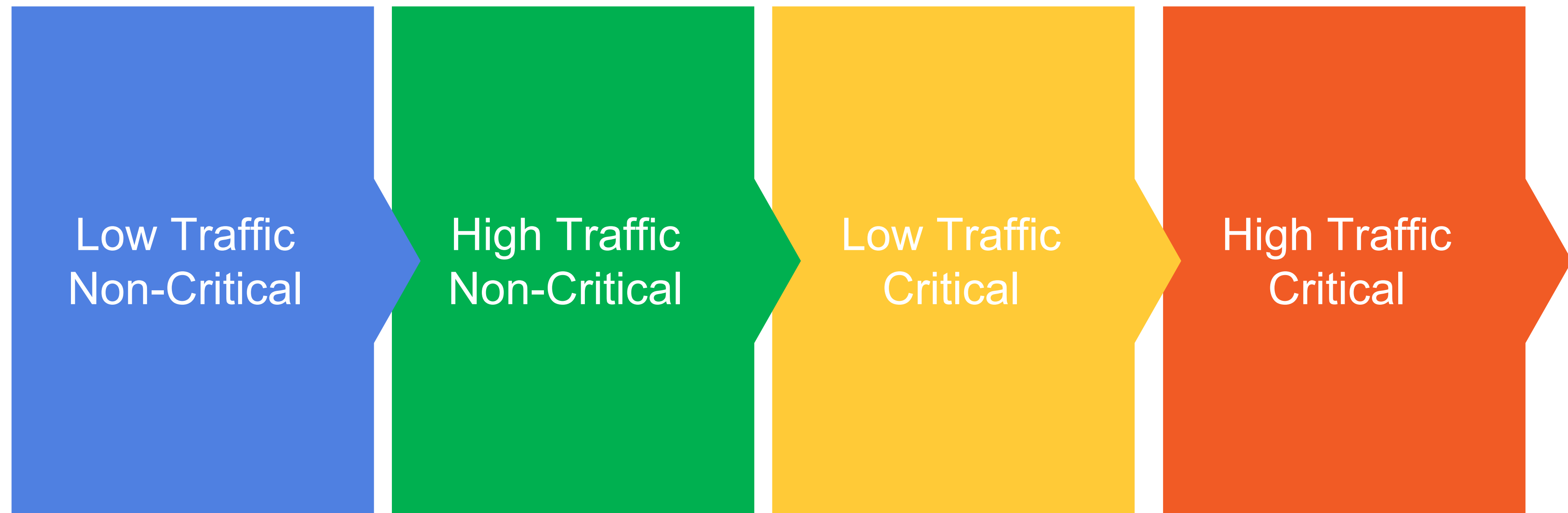


```
docker container run --rm -it \  
  --name ucp \  
  -v /var/run/docker.sock:/var/run/docker.sock \  
  docker/ucp:3.0.1 \  
  upgrade --interactive
```

Gotchas and Notes

- Images must be accessible on EVERY node
- No down nodes in UCP cluster
- Ubuntu 14.04 - see release notes for shared mounts
- Manager load balancing - kube API port 6443
- Kube DNS

Choosing applications



Choosing applications

Low Traffic
Non-Critical

notifications

Choosing applications



What didn't change

- No application code change
- No Dockerfile change
- No Environment changes

What did change

- K8S yaml files add
 - Deployment and Service object
- Registry secrets
- All in Source control

K8S deployment object

```
kind: Deployment
..
spec:
...
  template:
    metadata:
      labels:
        app: registry2sns
    spec:
      imagePullSecrets:
        - name: saasdeploycreds
      containers:
        - name: web
          image: docker/registry2sns:1093.0.0
          ports:
            ...
```

K8S Service object

```
apiVersion: v1
kind: Service
metadata:
  name: registry2sns
  labels:
    app: registry2sns
spec:
  type: NodePort
  ports:
    - port: 80
      nodePort: 32769
  selector:
    app: registry2sns
```

K8S Service object

```
apiVersion: v1
kind: Service
metadata:
  name: registry2sns
  labels:
    app: registry2sns
spec:
  type: NodePort
  ports:
    - port: 80
      nodePort: 32769
  selector:
    app: registry2sns
```

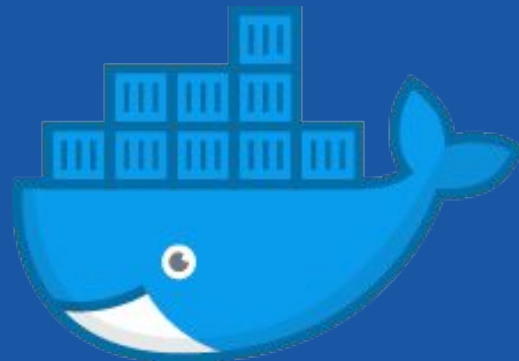
```
kind: Deployment
..
spec:
  ...
  template:
    metadata:
      labels:
        app: registry2sns
    spec:
      imagePullSecrets:
        - name: saasdeploycreds
      containers:
        - name: web
          image: docker/registry2sns:1093.0.0
          ports:
            ...
```

Testing App Migration to K8S

- Test requests against k8s pods
- Service backed by swarm containers and k8s pods
- Change in deploy tooling

Leveraging EE and Kubernetes

UCP Interfaces



Docker api

- Docker cli
- Swarm
- RBAC



Kubernetes api

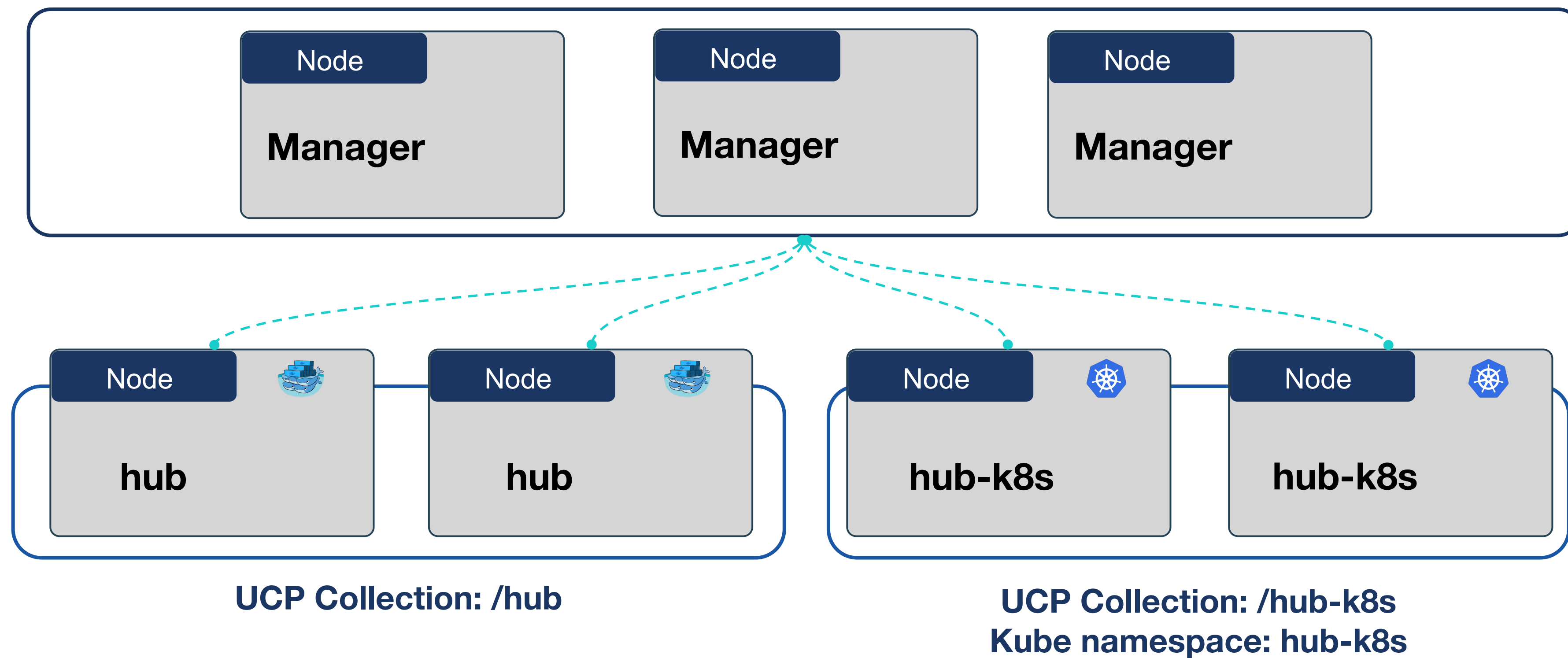
- kubectl
- k8s api server
- RBAC



Web UI

- Monitoring
- Configuration
- Single pane

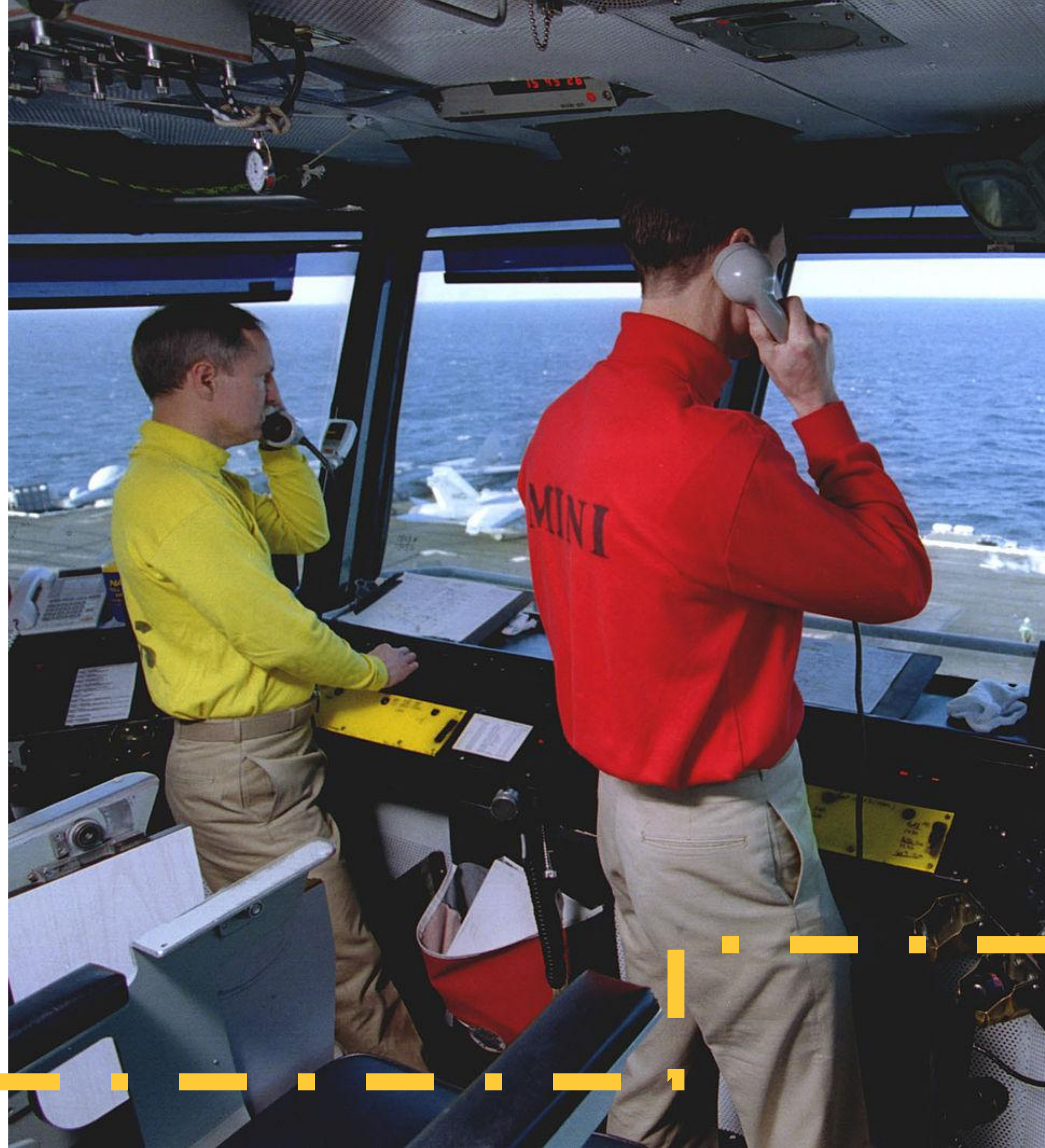
Resource Sets



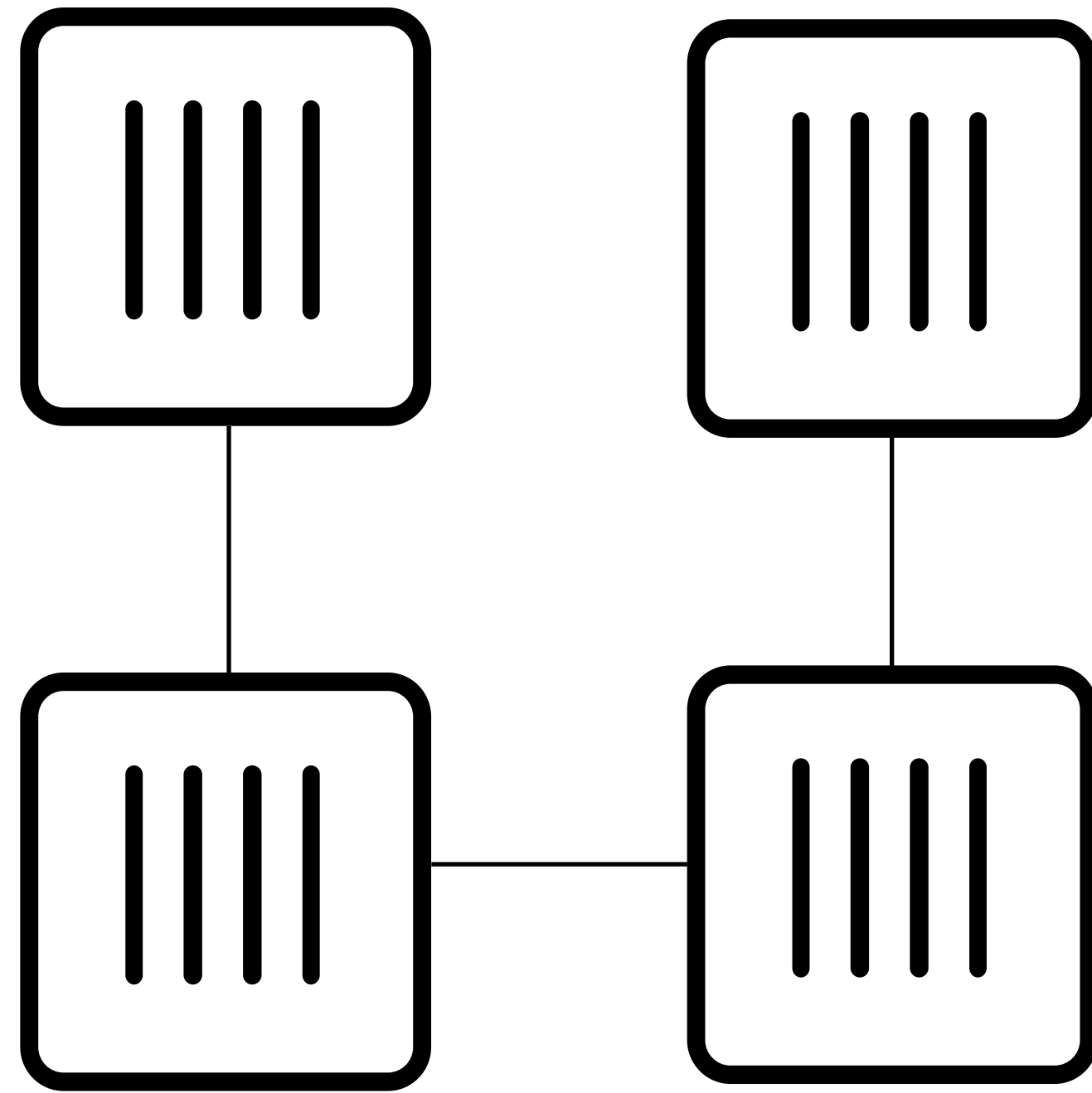
Airboss

Based on engine labels:

- Set node orchestrator
- Create UCP collection
- Add node to UCP collection
- Apply collection label in kube
- Create kube namespace
- Create annotation linking collection label to namespace



Pods



- Deploy containers together
- Useful for
 - breaking up monoliths
 - localhost
 - sharing volumes
 - metric exporters

CronJobs

- Batch processing
- Moving system services into containers

* / 5 * * * *

Takeaways

Dogfooding

10 P0/P1 fixes and improvements for EE2.0

162 total bugs and feature requests

Takeaways

- Planning - milestones, communication
- Infrastructure
 - Sizing
 - Resource sets
 - Routing
 - Metrics/Monitoring
 - Gotchas and Notes
- Migration process
- Leverage EE and Kube features

Go try it!

Hosted Trial: trial.docker.com

EE trial licenses: store.docker.com