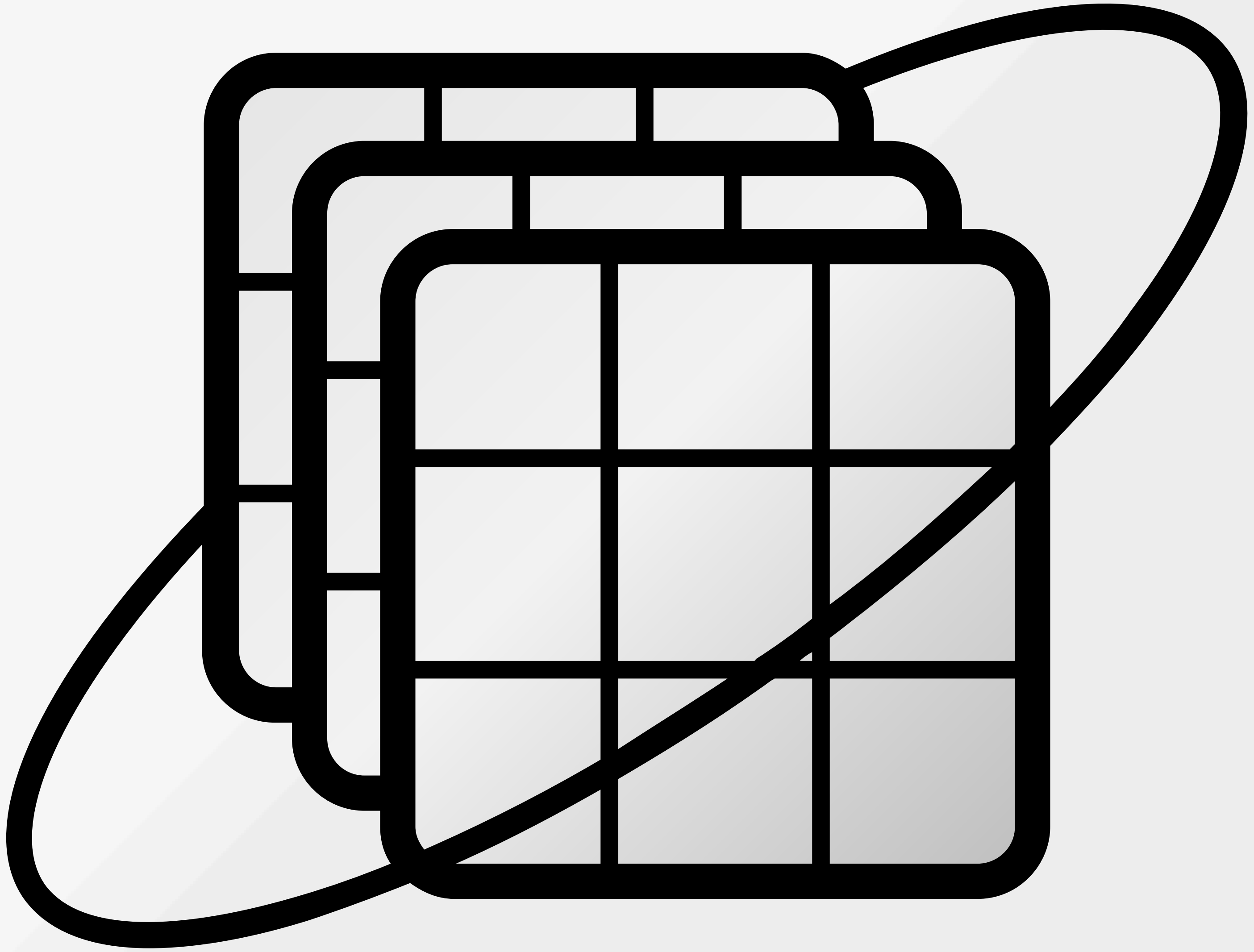
JAVA DEVELOPMENT IN THE AGE OF THE WHALE

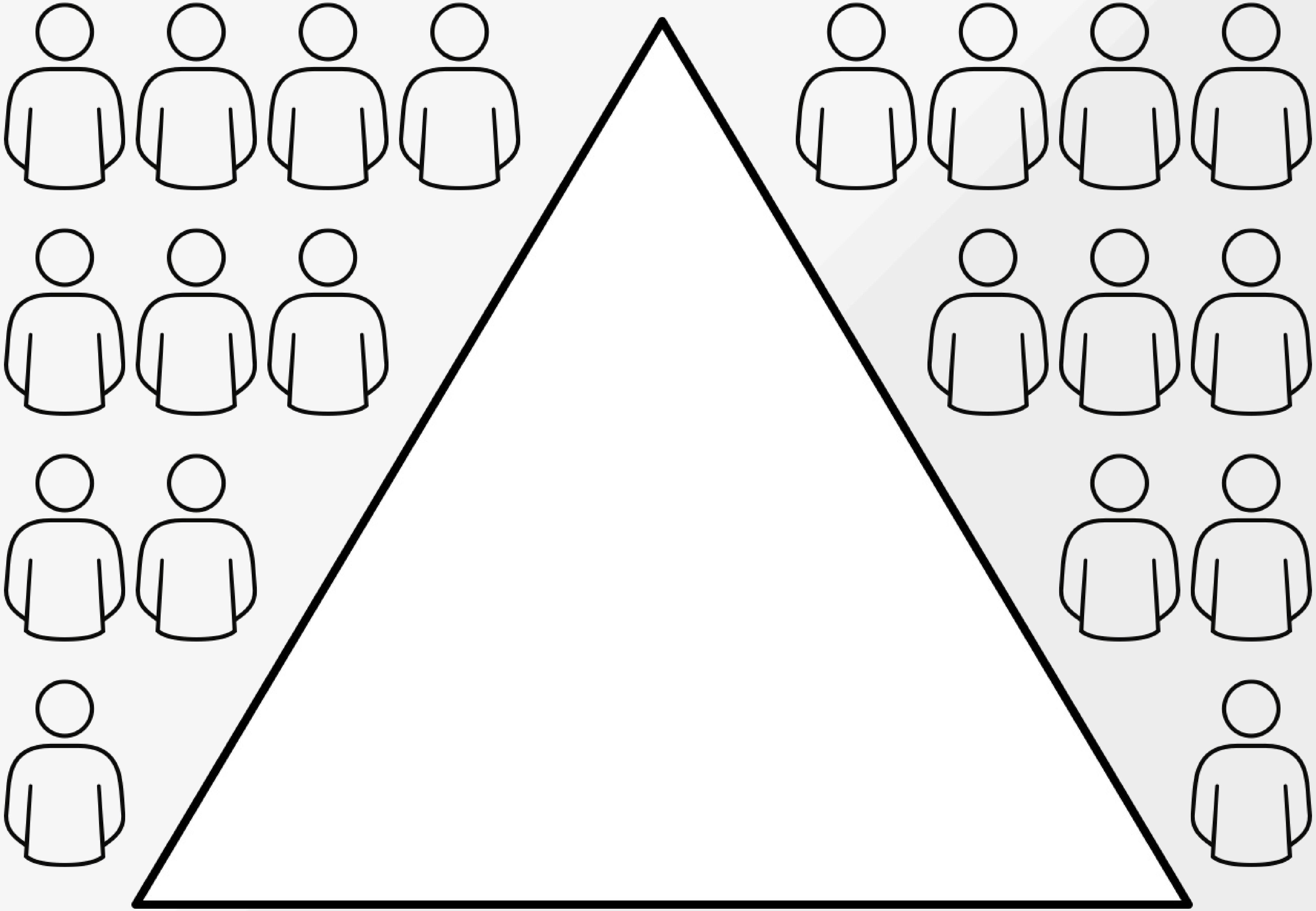
DevOpsCon 2017, München

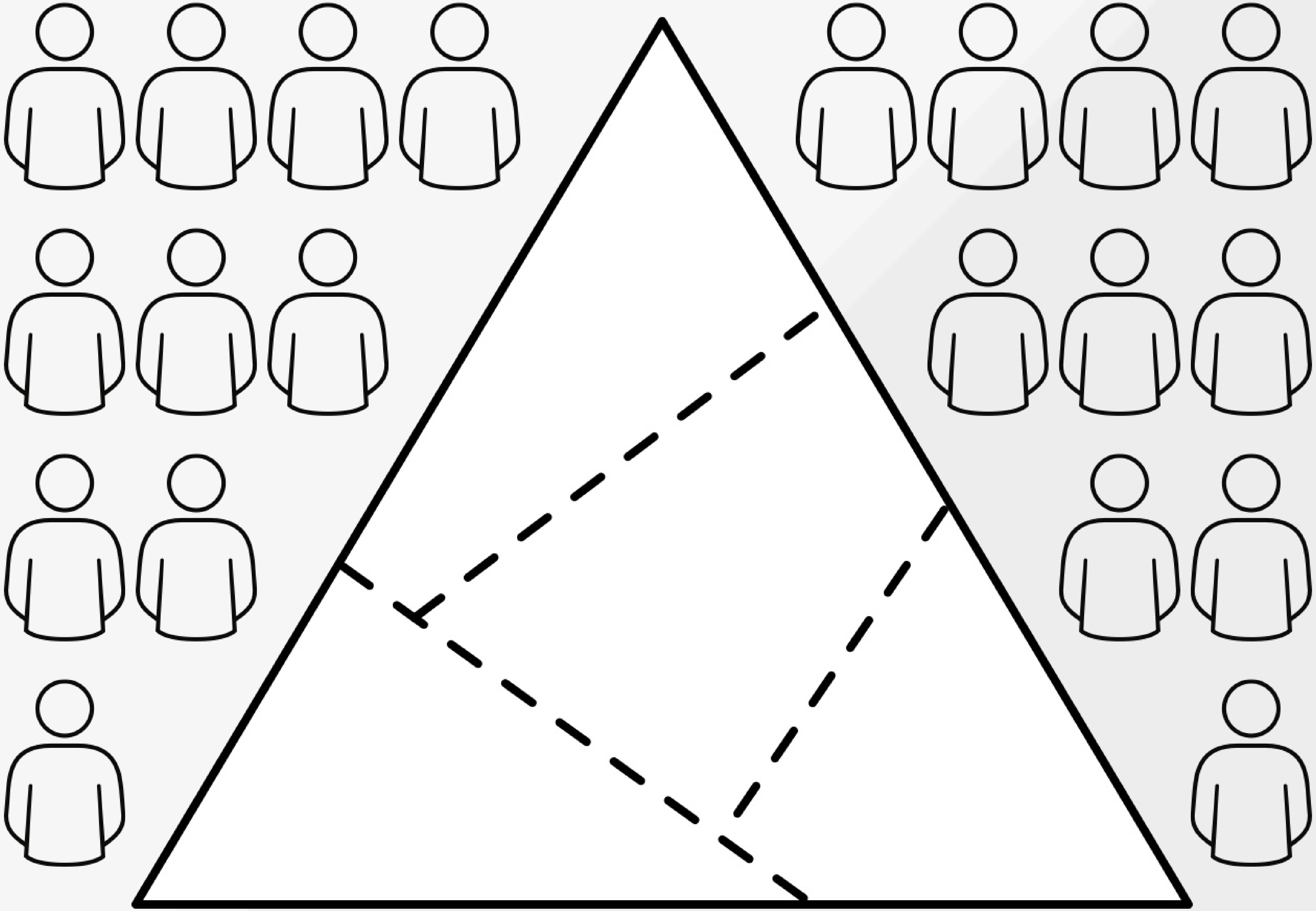
Dr. Roland Huβ, Red Hat, @ro14nd

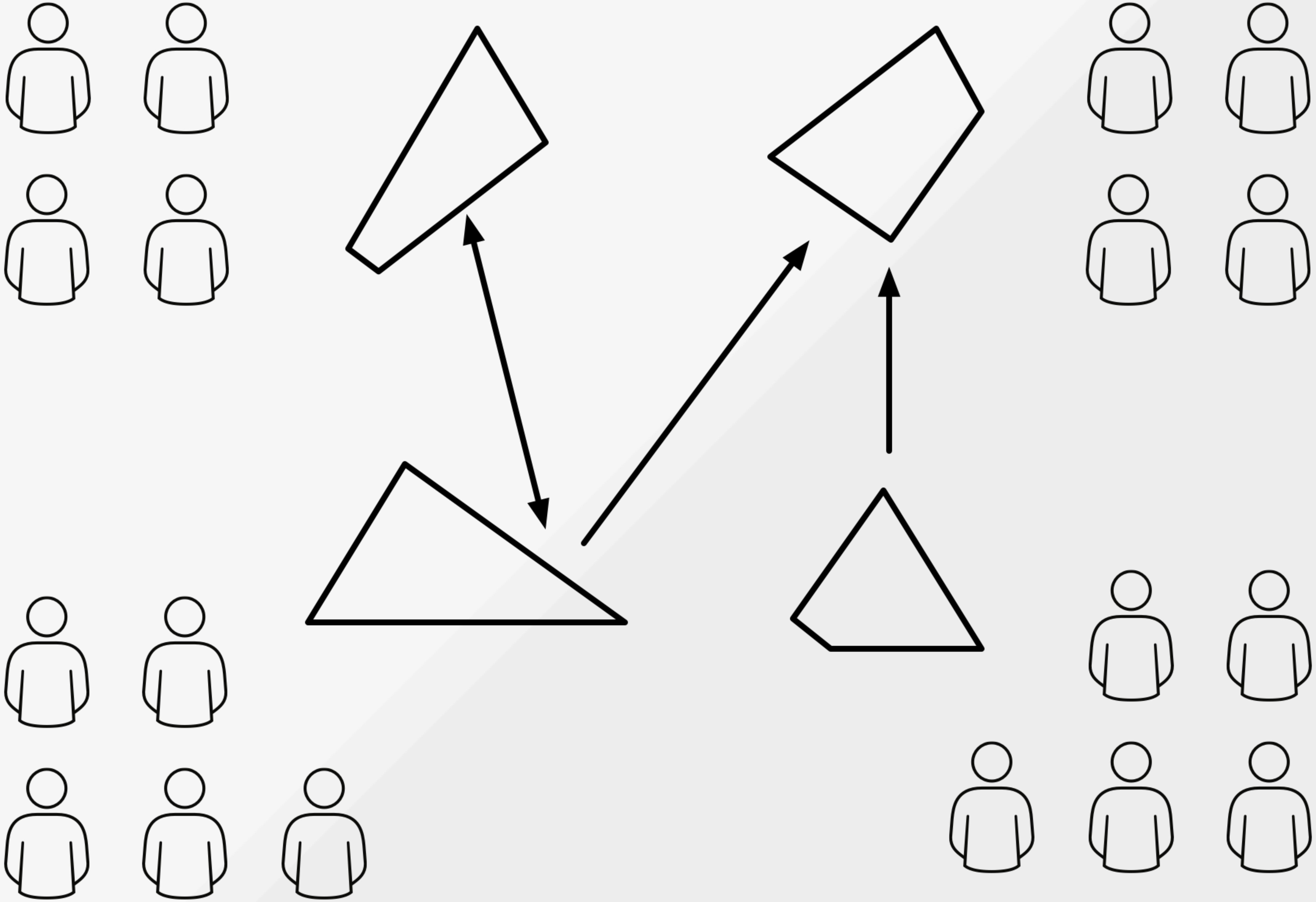
"**m**" for menu, "**?**" for other shortcuts

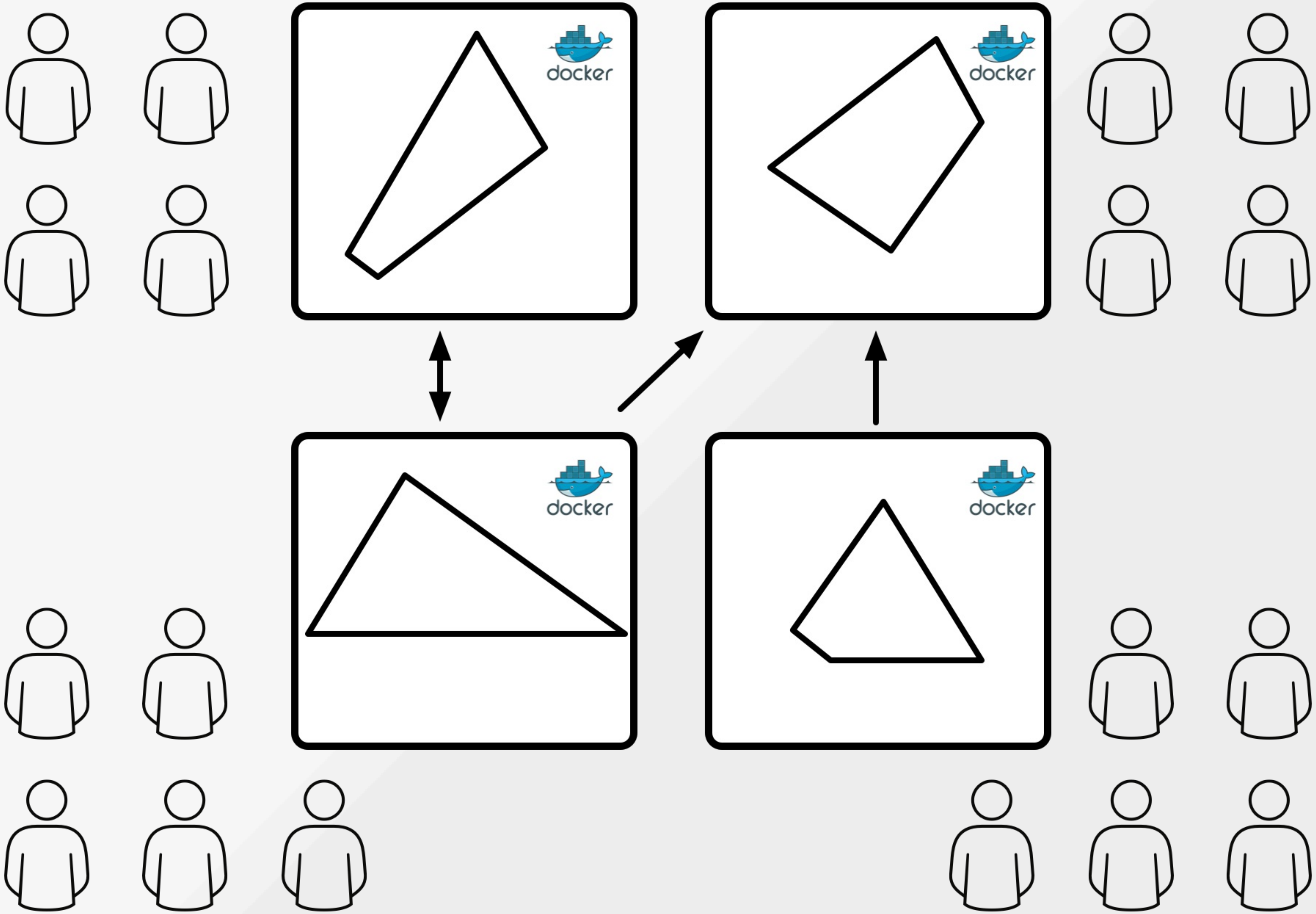


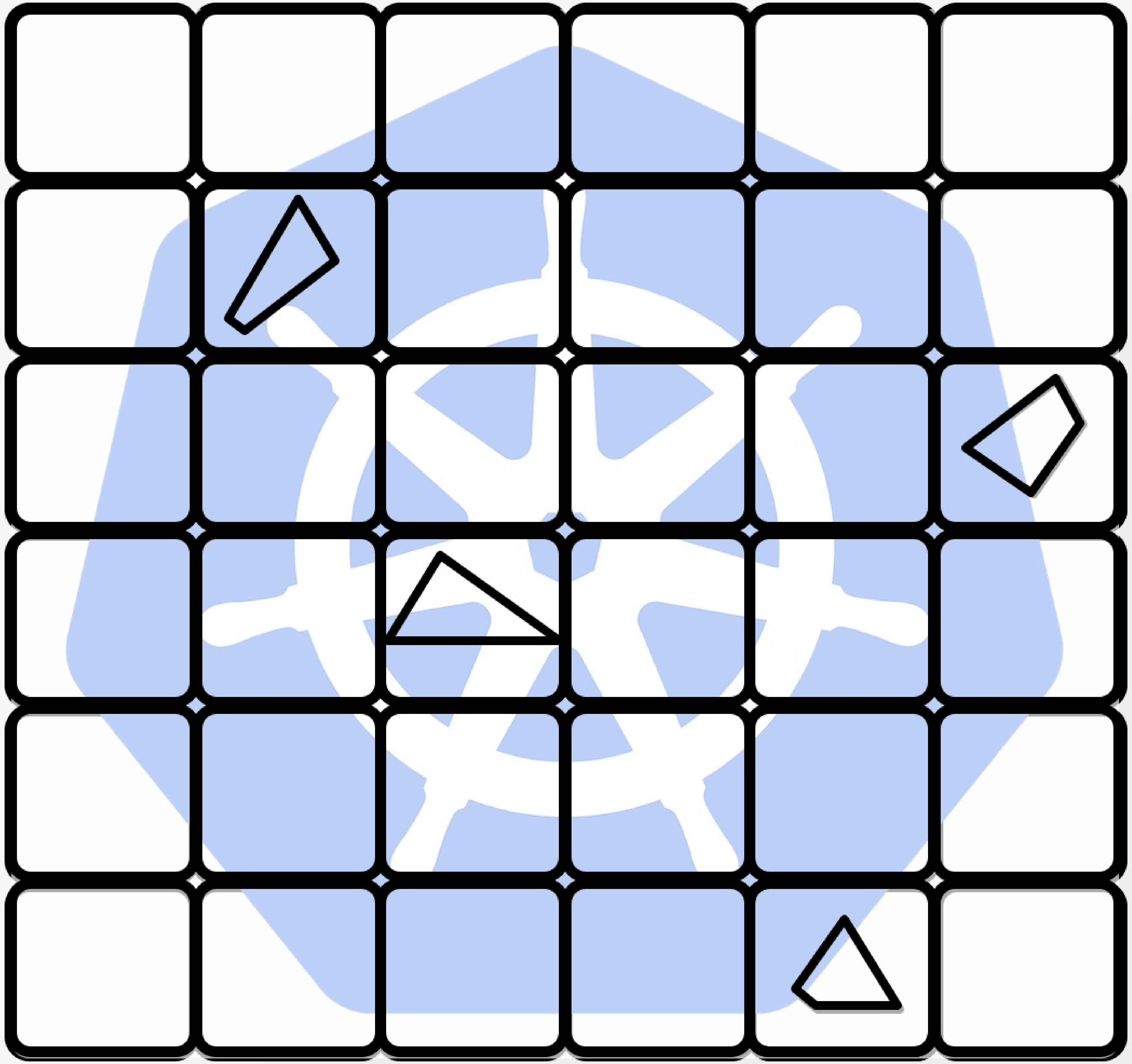
redhat







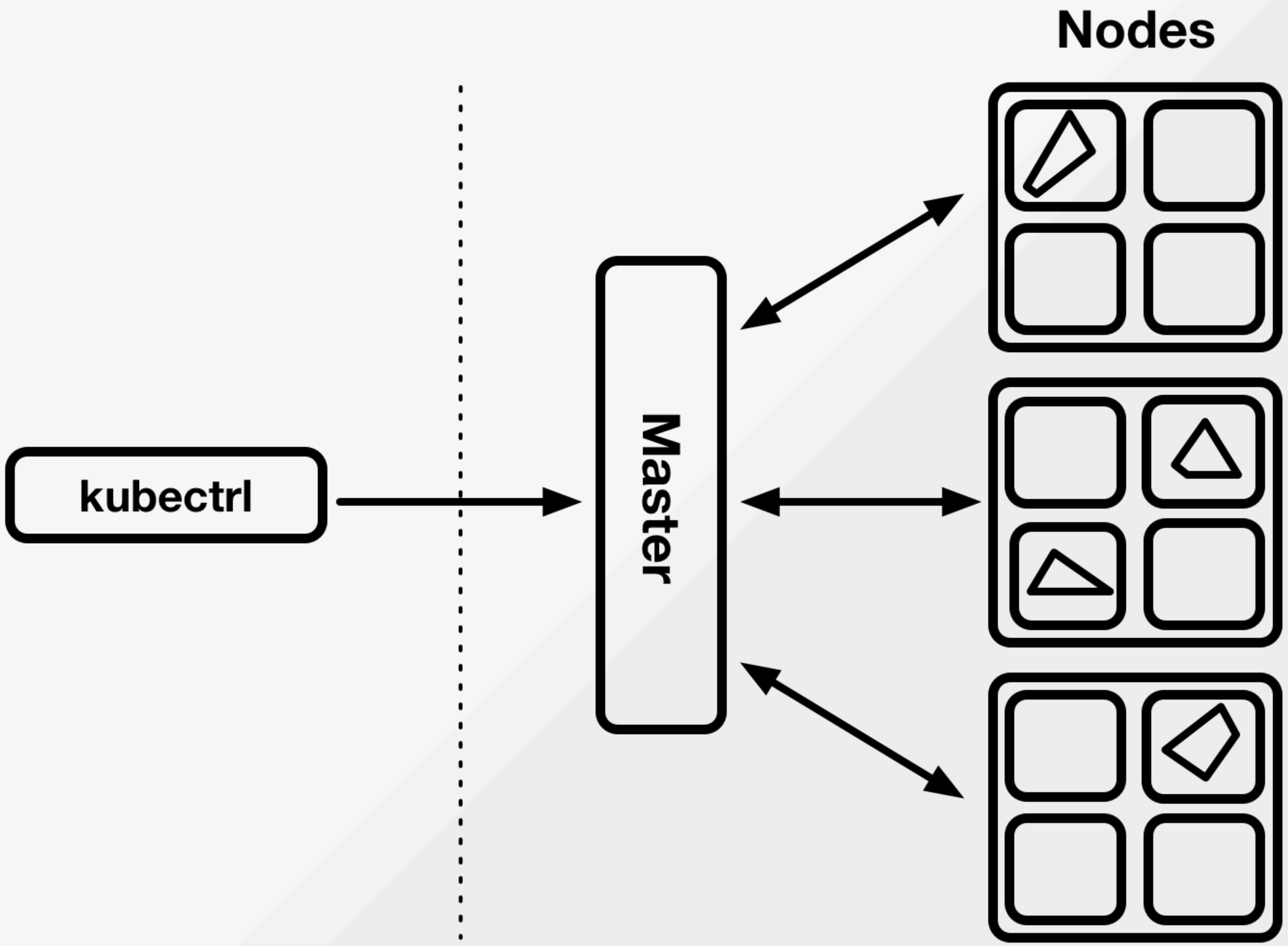

docker
docker
docker
docker



KUBERNETES

- Open Source orchestration system for containers
  - Scheduling
  - Horizontal scaling
  - Self-healing
  - Service discovery
  - Automated rollout and rollbacks

ARCHITECTURE



KUBERNETES IN THE CLOUD

- Google Container Engine (GKE)
- Azure Container Service (AKE)
- OpenShift Online
- AWS EC2, Digital Ocean, ...
  - Stackpoint.io
  - CoreOS Tectonic
  - Kubernetes Operations (kops)

RASPI CLUSTER

- 4 Raspberry Pi 3
- Wifi Router
- 6 Port USB charger
- 32 GB SD-Cards
- Costs: ~ 300 €
- Install via Ansible
- kubeadm

Full Story: https://ro14nd.de/kubernetes-on-raspberry-pi3

MINIKUBE

- Single-node Kubernetes cluster inside a VM
- No Docker daemon required
- Ideal for local development
- Supports DNS, NodePorts, Volumes, ...
- https://github.com/kubernetes/minikube

fabric8

FABRIC8

- **Microservices Platform** for Kubernetes & OpenShift
- Upstream projects for openshift.io
- Themes:
  - Continous Delivery
  - Management UI
  - Funktion
  - Quickstarts
  - Tooling

FABRIC8-MAVEN-PLUGIN

- Creates **Docker images** and **resource descriptors**
- Zero-configuration with **opinionated defaults**
- Full-configuration with **enrichable fragments**
- https://maven.fabric8.io

GOALS

fabric8:build	Build application images (Docker, S2I binary, S2I source)
fabric8:resource	Create Kubernetes and OpenShift resource descriptors
fabric8:apply	Apply resource descriptors to a running cluster

CONFIGURATION

- Zero Config
    - Opinionated Defaults
    - Limited configuration options
- XML Configuration
    - Restricted configuration syntax
- Resource Fragments
    - Most powerful
    - Verbose

ZERO CONFIG

- **Generators** for Image generation

```
<build>
  <plugins>
    <plugin>
      <groupid>io.fabric8</groupid>
      <artifactid>fabric8-maven-plugin</artifactid>
      <version>3.5.31</version>
    </plugin>
    <plugin>
      <groupid>org.springframework.boot</groupid>
      <artifactid>spring-boot-maven-plugin</artifactid>
    </plugin>
  </plugins>
</build>
```

RESOURCE FRAGMENTS

- Resource fragment

  src/main/fabric8/pong-rc.yml

```
spec:
 replicas: 1
 template:
  spec:
   containers:
   - name: pong
    ports:
    - containerPort: 8080
```

- **Enrichers** add missing pieces

GENERATORS

- Extract Docker image configuration from pom.xml
- Supported technologies:
  - Spring Boot
  - Wildfly Swarm
  - Fat Jars
  - Eclipse Vert.x
  - Webapps
  - Karaf

ENRICHERS

- Add default Kubernetes resources
- Update existing resources
- E.g.
    - Default **Deployment** and **Service**
    - Git information as labels
    - Add healthchecks
    - Add OpenShift routes
    - ....

PROFILES

- Named collection of enrichers and generators
- -Dfabric8.profile to select

raw	No enrichment
explicit	Only default objects
minimal	Small enrichments
aggregate	Combine resources from dependencies

K8S & OPENSHIFT

- Kubernetes:
  - Docker builds
  - Deployments
  - Ingress
- OpenShift
  - S2I & Docker Binary Builds
  - DeploymentConfig
  - ImageStream
  - Template

MISC

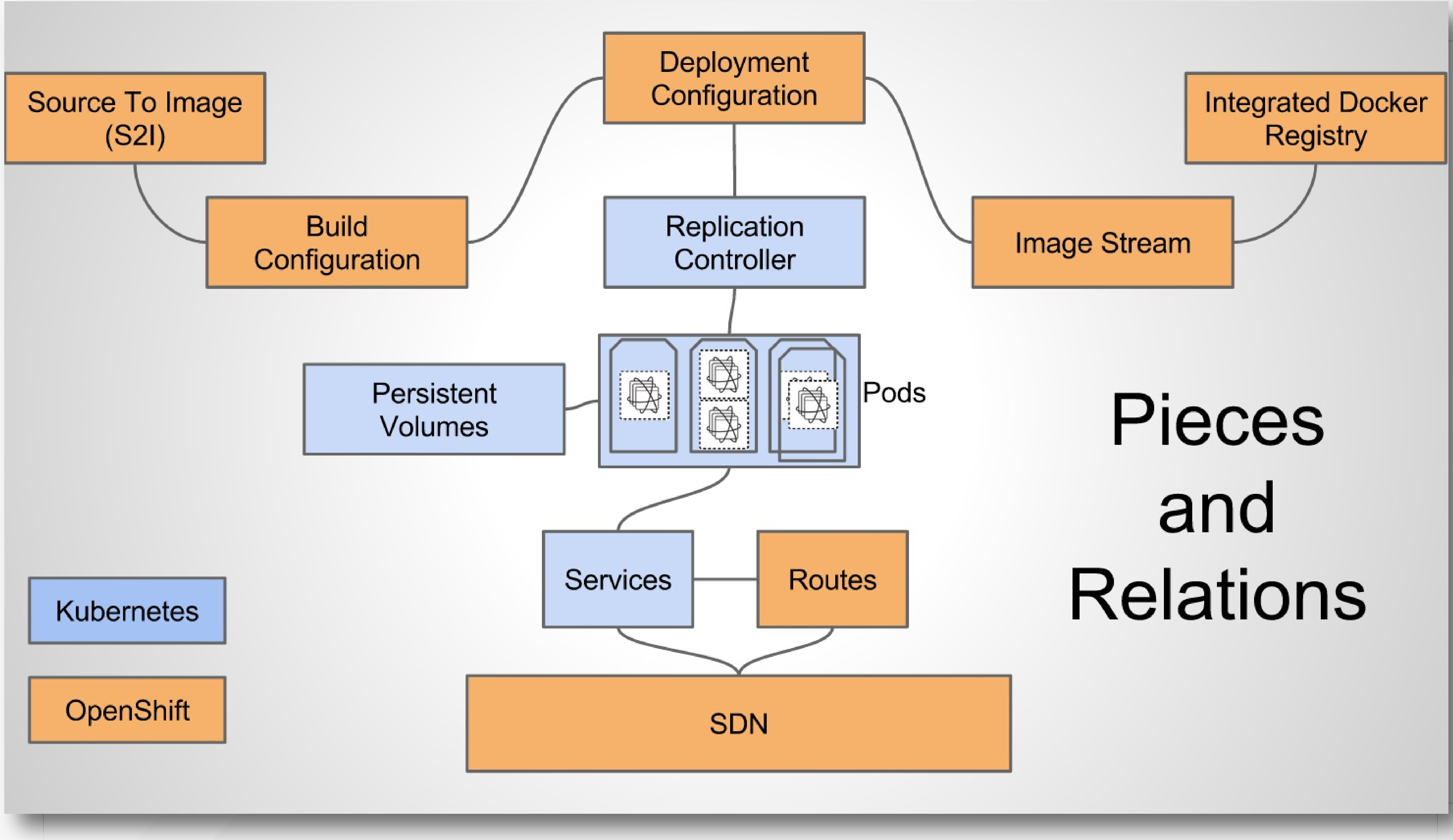
fabric8:install	Install local development environment
fabric8:cluster-start	Start minikube or minishift
fabric8:watch	Watch for changes and redeployments
fabric8:debug	Debug into pods

OPENSHIFT

OPENSHIFT

- Adds the **BUILD** to Kubernetes
- Infrastructure Services
  - Registry
  - Router
  - OAuth2 SSO
- Multi tenancy
- Management UI
- ...

OPENSHIFT EXTRAS



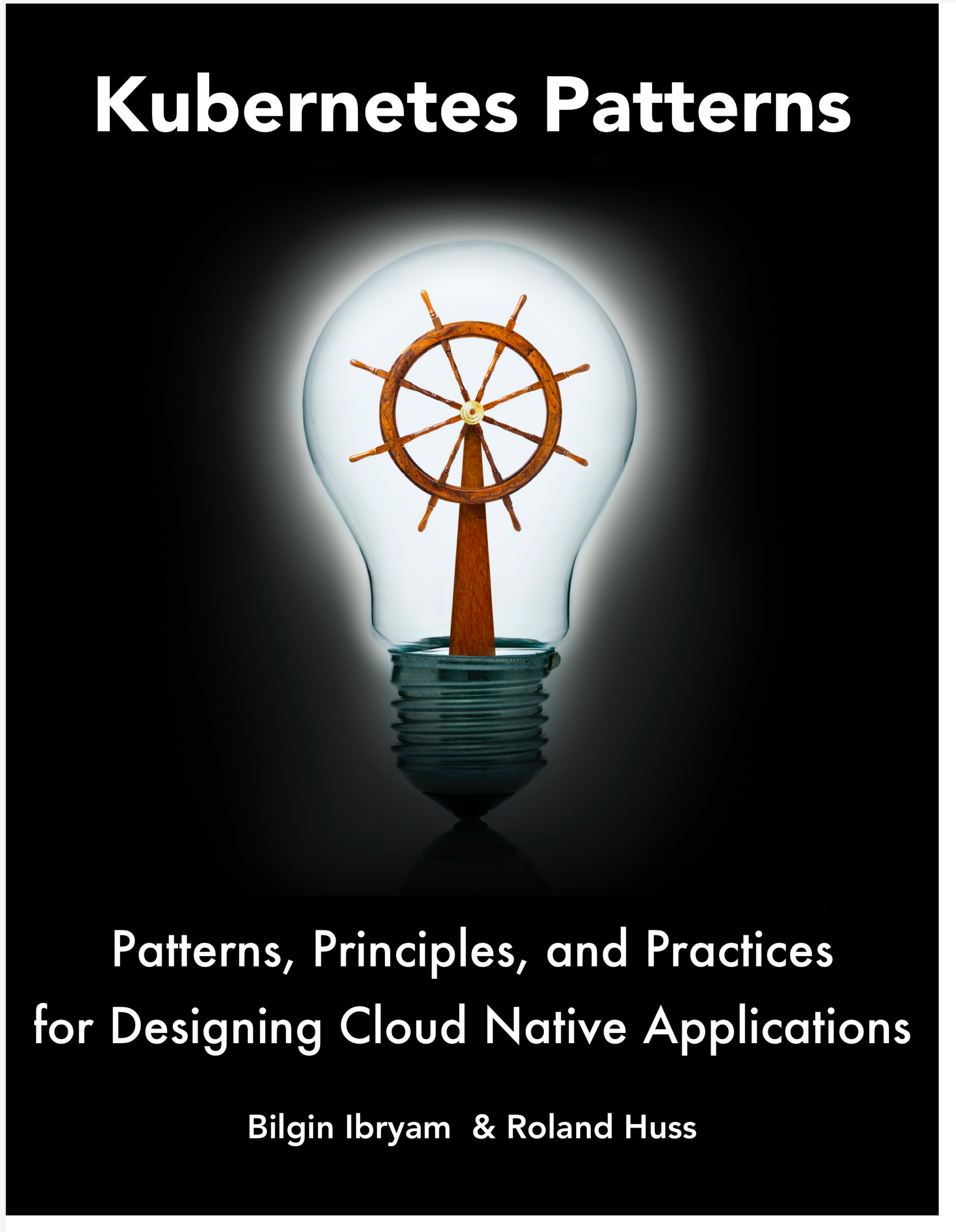
MINISHIFT

- Single-node OpenShift Origin cluster inside a VM
- Based on **oc cluster up**
- Supports routes, registry, s2i builds, ...
- https://github.com/minishift/minishift

WRAP UP

- Starting with Kubernetes can be almost as easy as with Docker
- Kubernetes and OpenShift are powerful orchestration platforms with enterprise grade features.
- Use **fabric8-maven-plugin** for Java apps

redhat.



Kubernetes Patterns

Patterns, Principles, and Practices
for Designing Cloud Native Applications

Bilgin Ibryam & Roland Huss

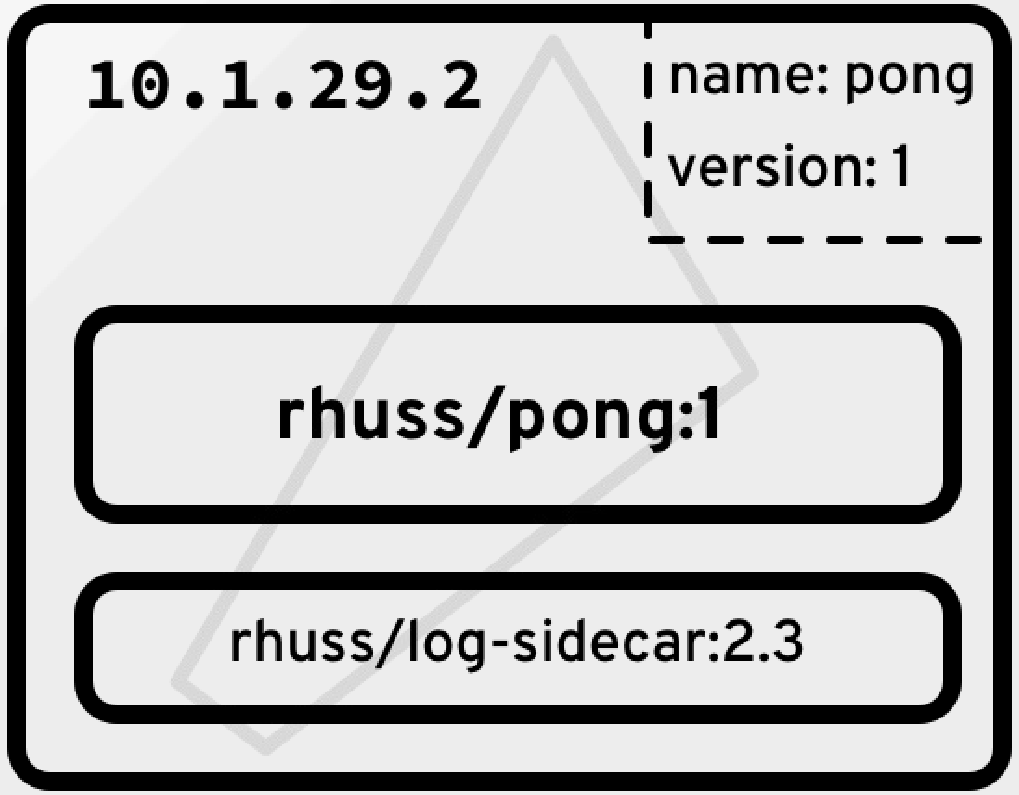
https://leanpub.com/k8spatterns

QUESTIONS ?

Twitter	@ro14nd
Slides	https://github.com/ro14nd-talks/kubernetes-for-java-developers

POD

- Kubernetes Atom
- One or more containers sharing:
  - IP and ports
  - Volumes
- Ephemeral IP address

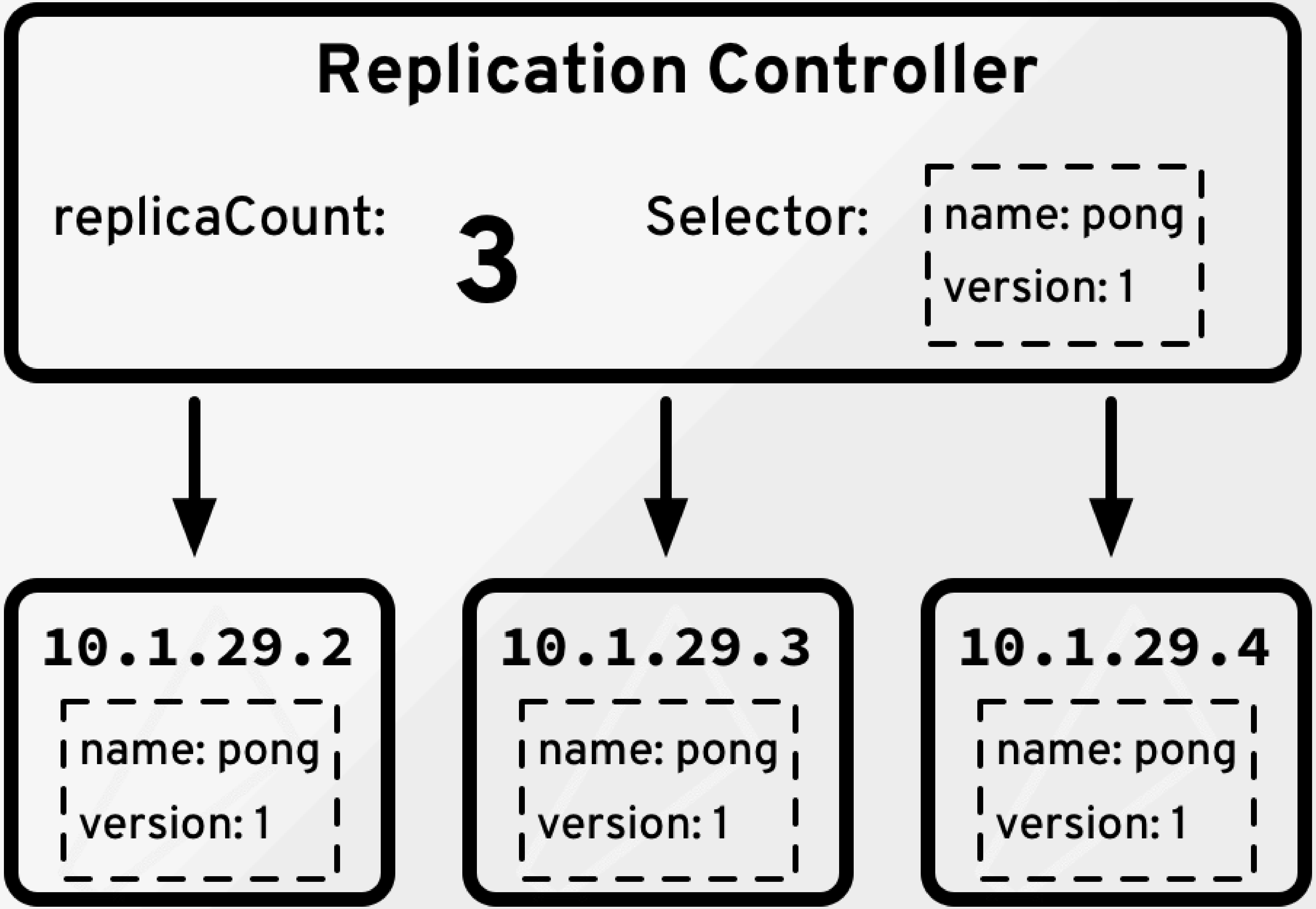


LABELS

- Metadata attachable to every resource object
- Used to categorize stuff
- Important for selectors
- "Freeform"

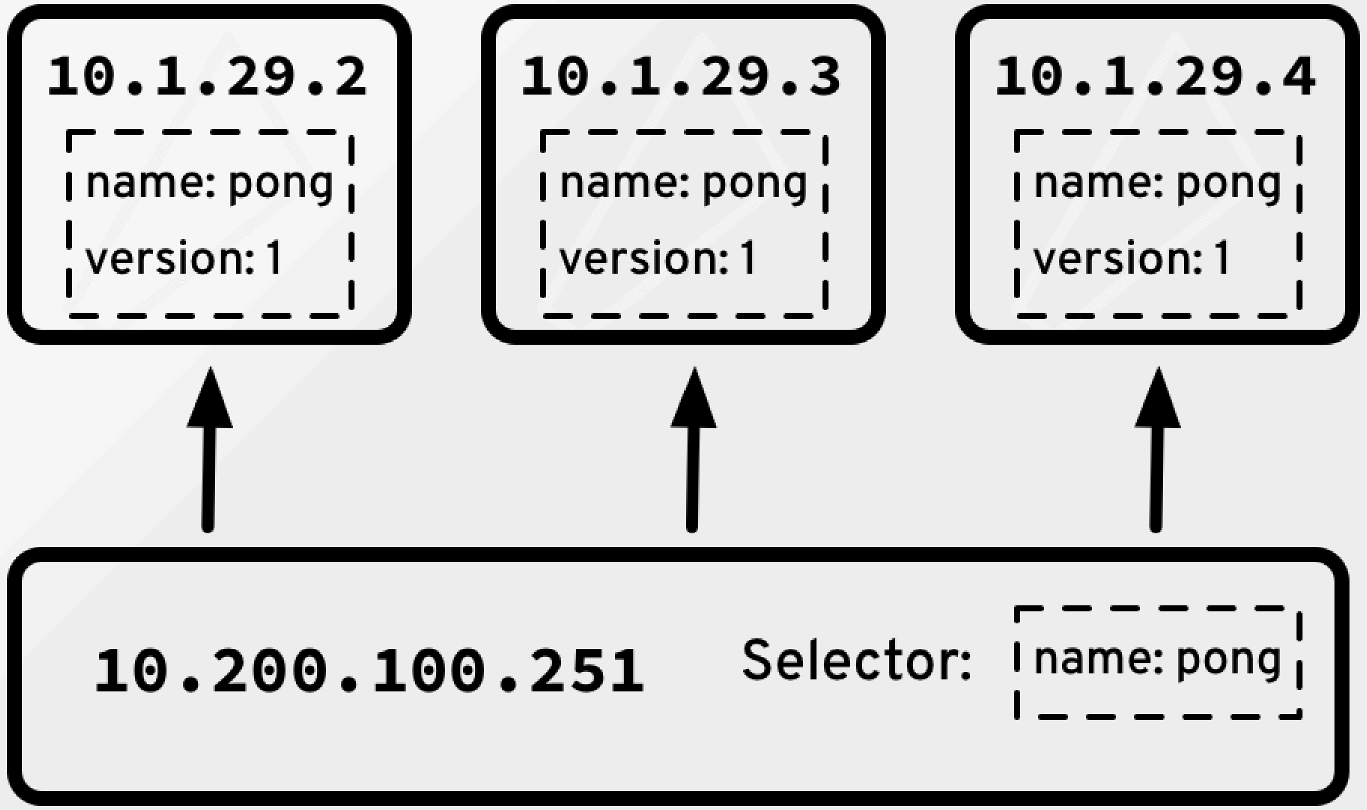
REPLICATION CONTROLLER

- Responsible for managing **Pods**
- **replicas** : Number of **Pod** copies to keep
- Label selector choose **Pods**
- Holds a template for creating new **Pods**


Replication Controller
replicaCount:
3
Selector:
name: pong
version: 1
10.1.29.2
name: pong
version: 1
10.1.29.3
name: pong
version: 1
10.1.29.4
name: pong
version: 1

SERVICE

- Proxy for a set of **Pods**
- **Pods** selected by **Label** selector
- Permanent IP address



ROLLING UPDATE

- kubectl rolling-update
- Downscale of old **replication controller**
- Upscale of new **replication controller**

VOLUMES

- Distributed storage
- Support types:
  - Local
  - NFS
  - Gluster
  - Ceph
  - ...

MISC FEATURES

- Secrets
- ConfigMaps
- ServiceAccounts
- Health & Liveness Checks
- Ingress