



Cloud-native data warehouses

A Gentle Introduction to **ClickHouse** on **Kubernetes**

Robert Hodges - July 2021

Presenter and Company Bio



Robert Hodges - Altinity CEO

30+ years on DBMS plus virtualization and security. Using Kubernetes since 2018.



Altinity

www.altinity.com

Enterprise provider for ClickHouse, a popular, open source data warehouse.
Implementors of ClickHouse
Kubernetes operator.

What is Kubernetes?

“Kubernetes is the new Linux”

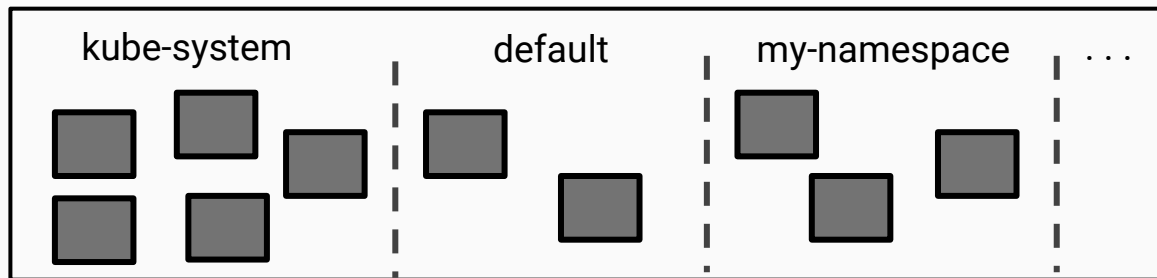
Actually it's an open-source platform that is:

- An orchestrator for container-based apps
- Infrastructure as code
- A great cluster manager
- A terminal endpoint for CI/CD automation

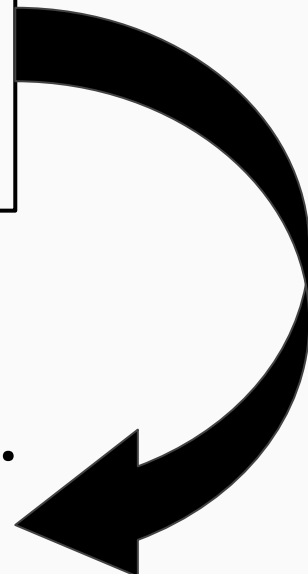
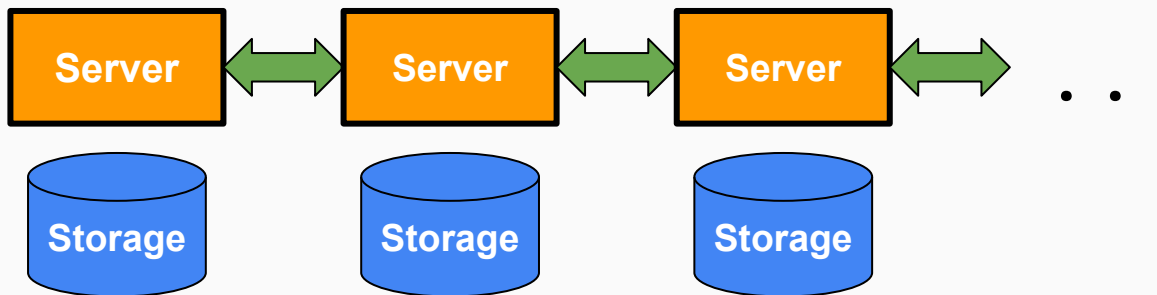
How Kubernetes manages applications



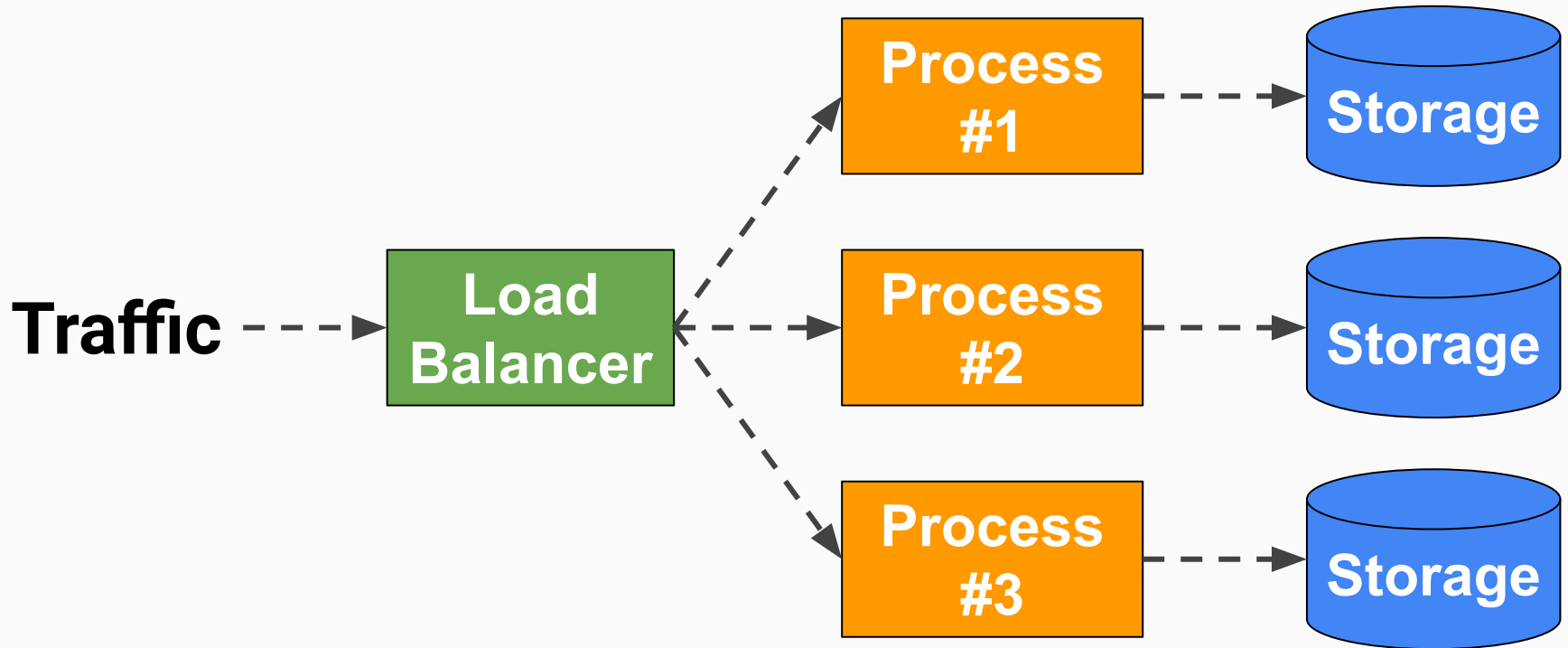
Kubernetes



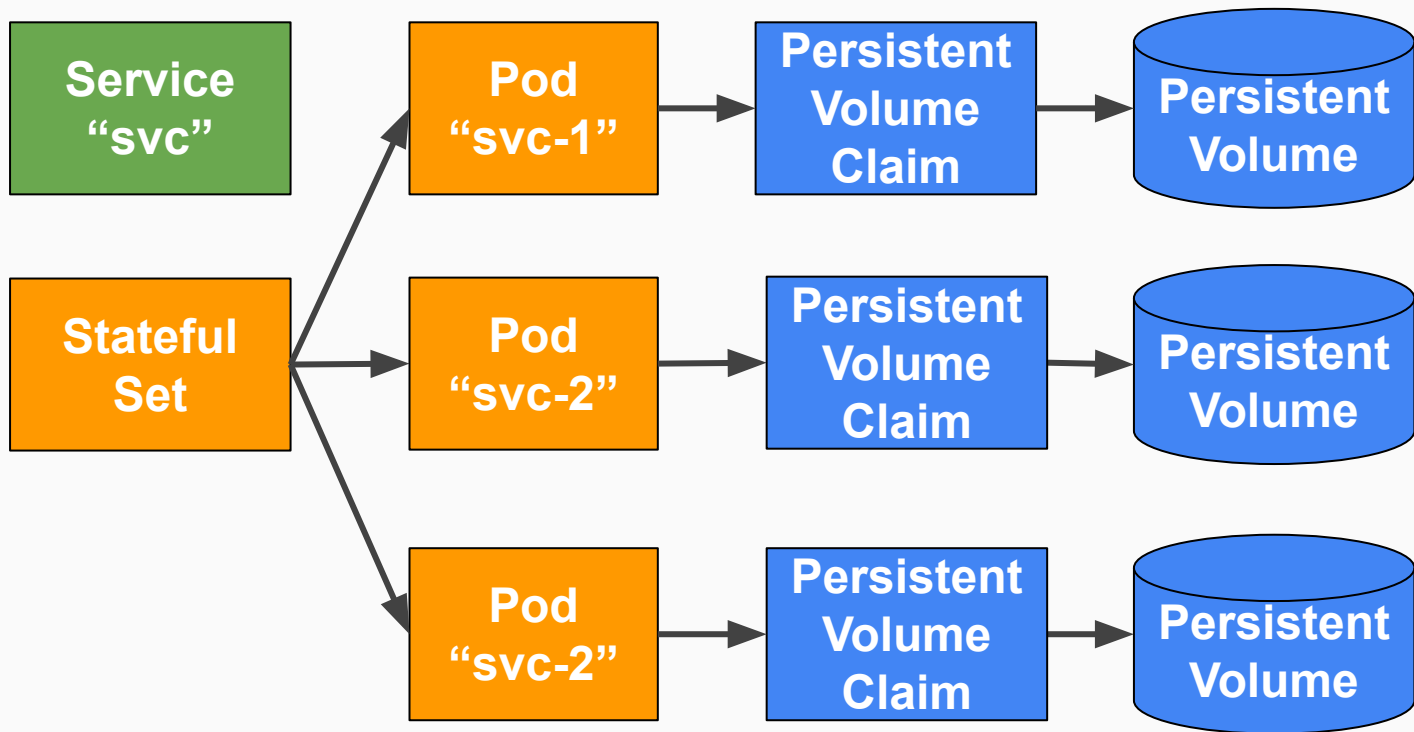
Physical Nodes



A typical distributed application



Defined using Kubernetes resources

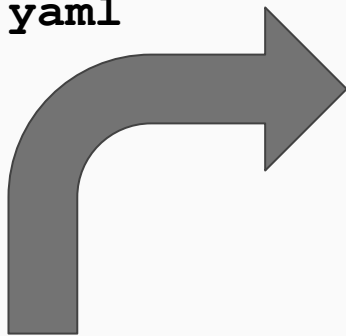


How does Kubernetes build applications?

```
kubectl apply -f  
somefile.yaml
```

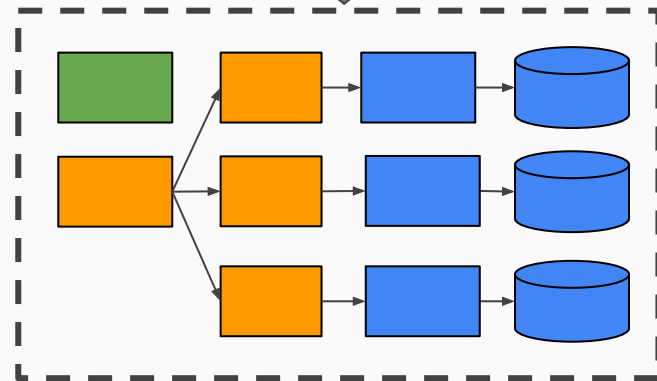
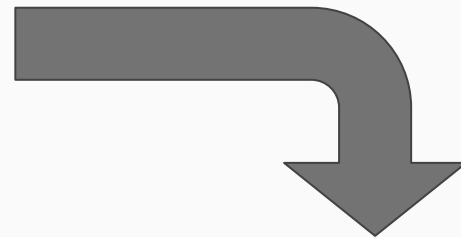


Specification(s) file



Kubernetes

“Adjust reality”

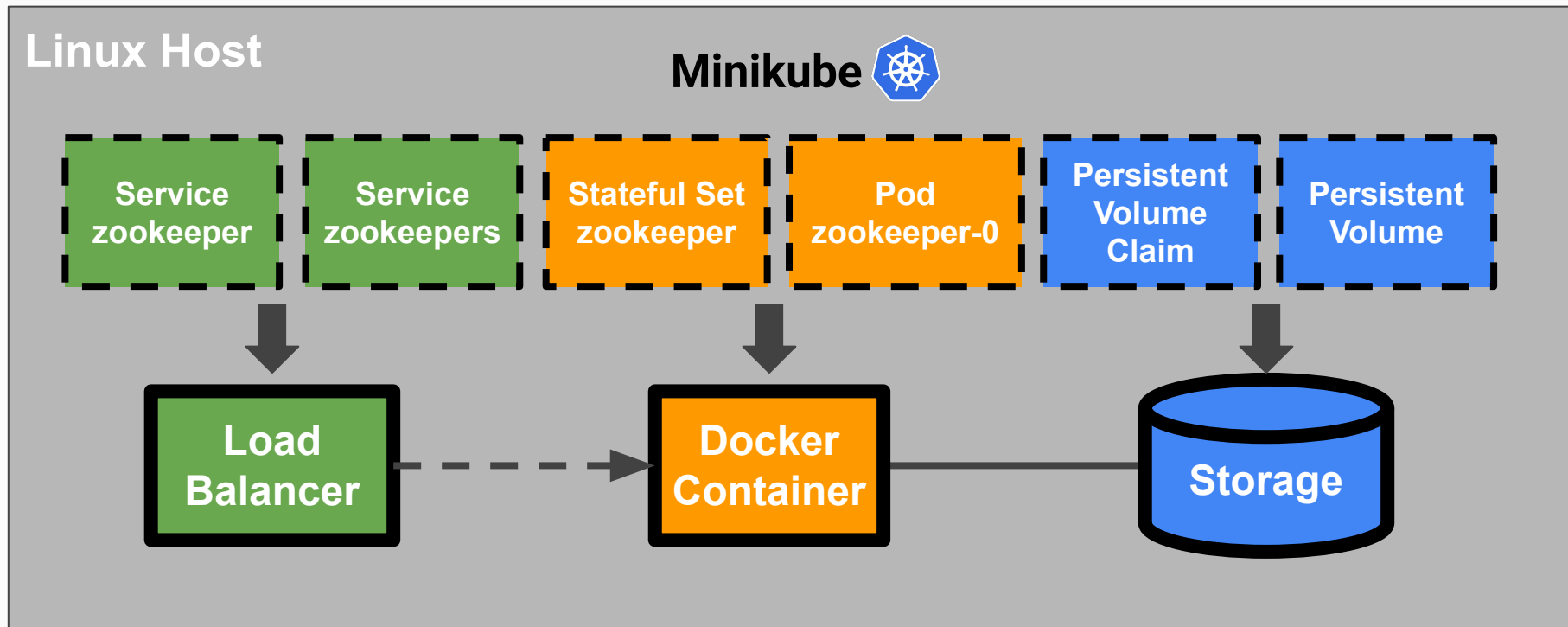


Some namespace

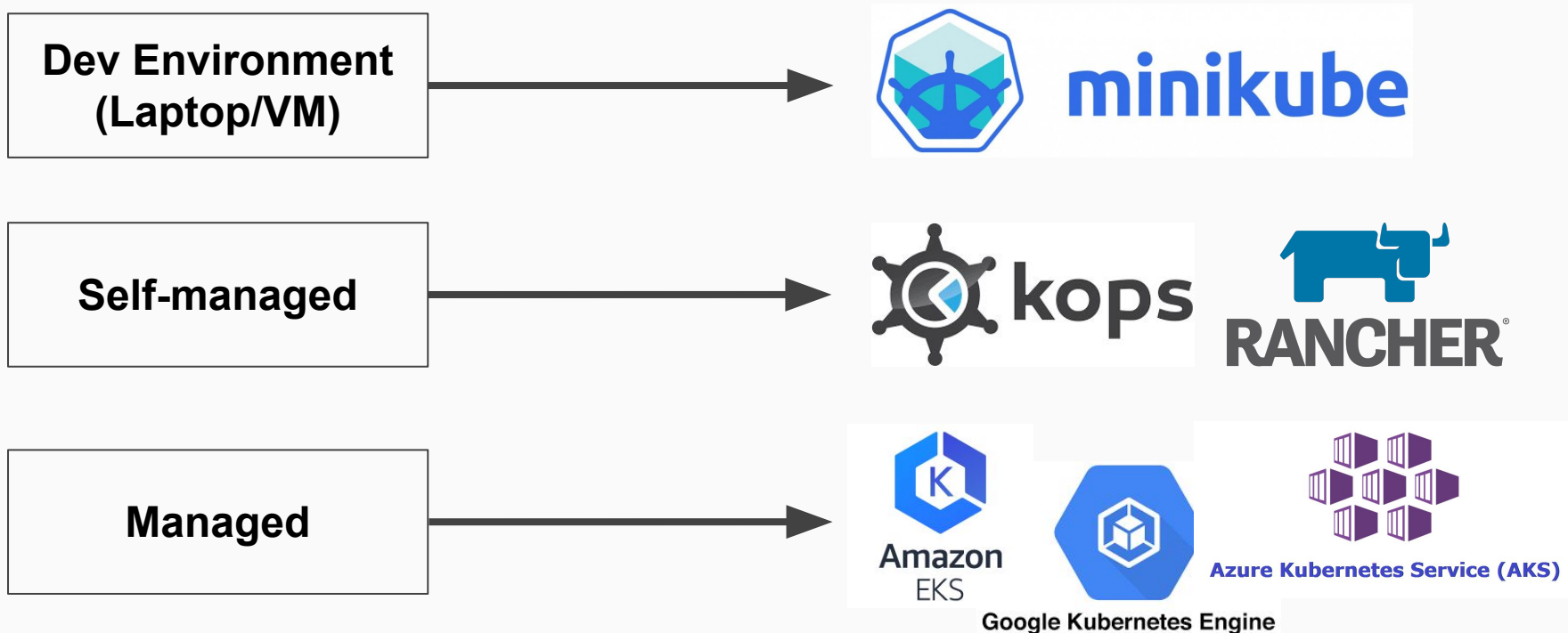
Demo time!

Deploying a stateful application on Kubernetes

Kubernetes entities map to host resources



Ways to run Kubernetes (non-exhaustive)



Running ClickHouse on Kubernetes

Introducing ClickHouse (in case you need it)

Understands SQL

Runs on bare metal to cloud

Compressed, columnar data

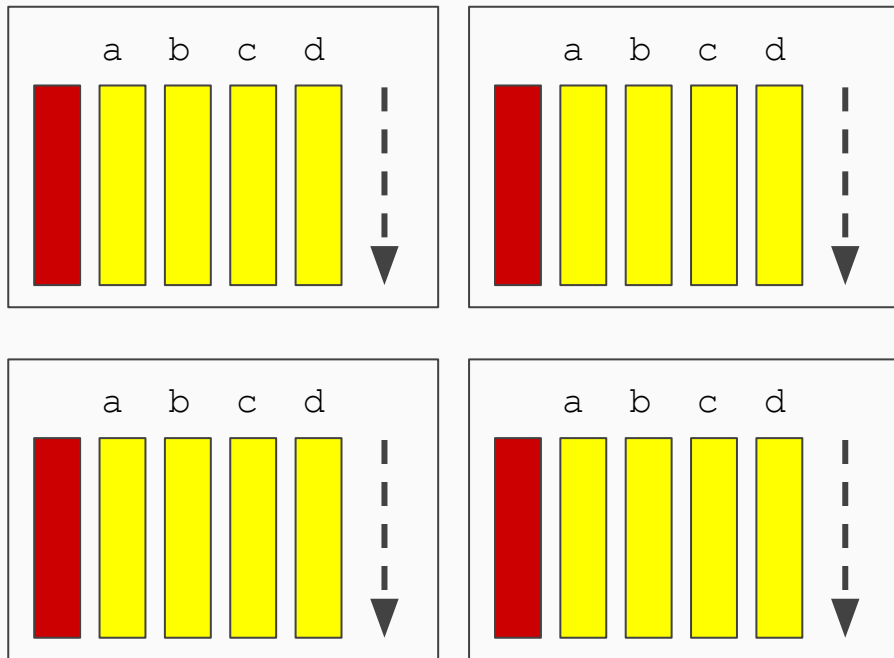
Parallel and vectorized execution

Built-in sharding and replication

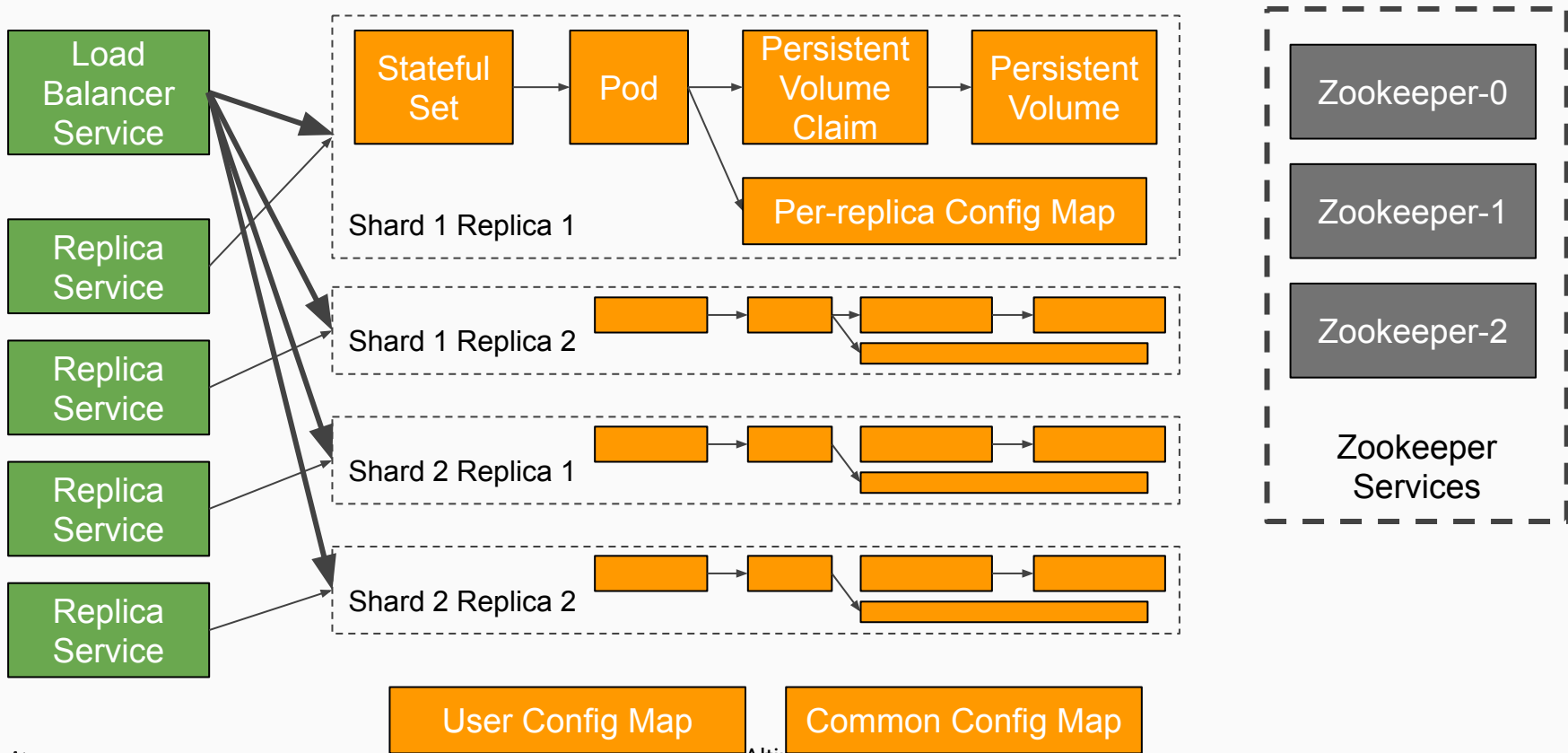
Linear scaling to 10s of petabytes

Open source (Apache 2.0)

And it's **really fast!**

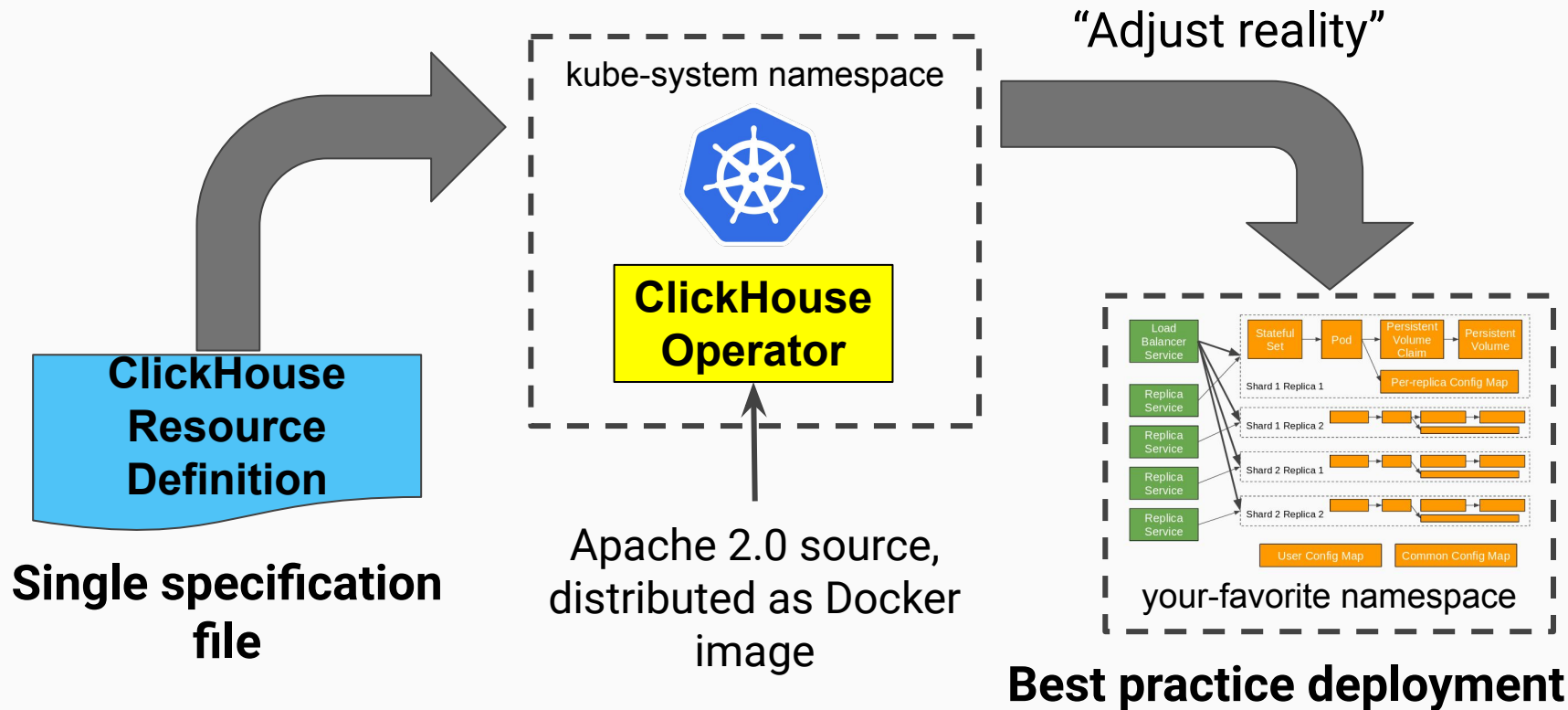


Problem: ClickHouse on Kubernetes is complex!



Solution: Kubernetes Operator

tes



Installing ClickHouse operator

Get operator custom resource definition:

```
wget \  
https://raw.githubusercontent.com/Altinity/clickhouse-operator/master/deploy/operator/clickhouse-operator-install.yaml
```

Install the operator:

```
kubectl apply -f clickhouse-operator-install.yaml
```

Installs in kube-system namespace by default

Installing Zookeeper

Get Zookeeper stateful set definition:

```
wget
```

```
https://raw.githubusercontent.com/Altinity/clickhouse-operator/master/deploy/zookeeper/quick-start-persistent-volume/zookeeper-1-node.yaml
```

Install Zookeeper.

```
kubectl create ns zk
```

```
kubectl apply -f zookeeper-1-node.yaml -n zk
```



Dev/test only!

We're ready to start working with data

**Time to bring up a
data warehouse!**

ClickHouse Resource Definition, part 1

```
apiVersion: "clickhouse.altinity.com/v1"
kind: "ClickHouseInstallation"
metadata:
  name: "k8s"
spec:
  configuration:
    clusters:
      - name: "demo"
        layout:
          shardsCount: 2
          replicasCount: 1
        templates:
          podTemplate: clickhouse
          volumeClaimTemplate: persistent
    zookeeper:
      nodes:
        - host: zookeeper.zk
          port: 2181
```

Resource type

Shards and replicas

Pod/storage details

Where is Zookeeper?

ClickHouse Resource Definition, part 2

```
templates:  
  podTemplates:  
    - name: clickhouse  
      spec:  
        containers:  
          - name: clickhouse  
            image: yandex/clickhouse-server:21.6  
  volumeClaimTemplates:  
    - name: persistent  
      reclaimPolicy: Retain  
      spec:  
        accessModes:  
          - ReadWriteOnce  
        resources:  
          requests:  
            storage: 10Gi
```

**ClickHouse
Docker version**



Don't delete storage!



Storage type and size



Accessing our creations

Safety first...Check PVs!

```
kubectl get pv  
--selector=clickhouse.altinity.com/namespace=default
```

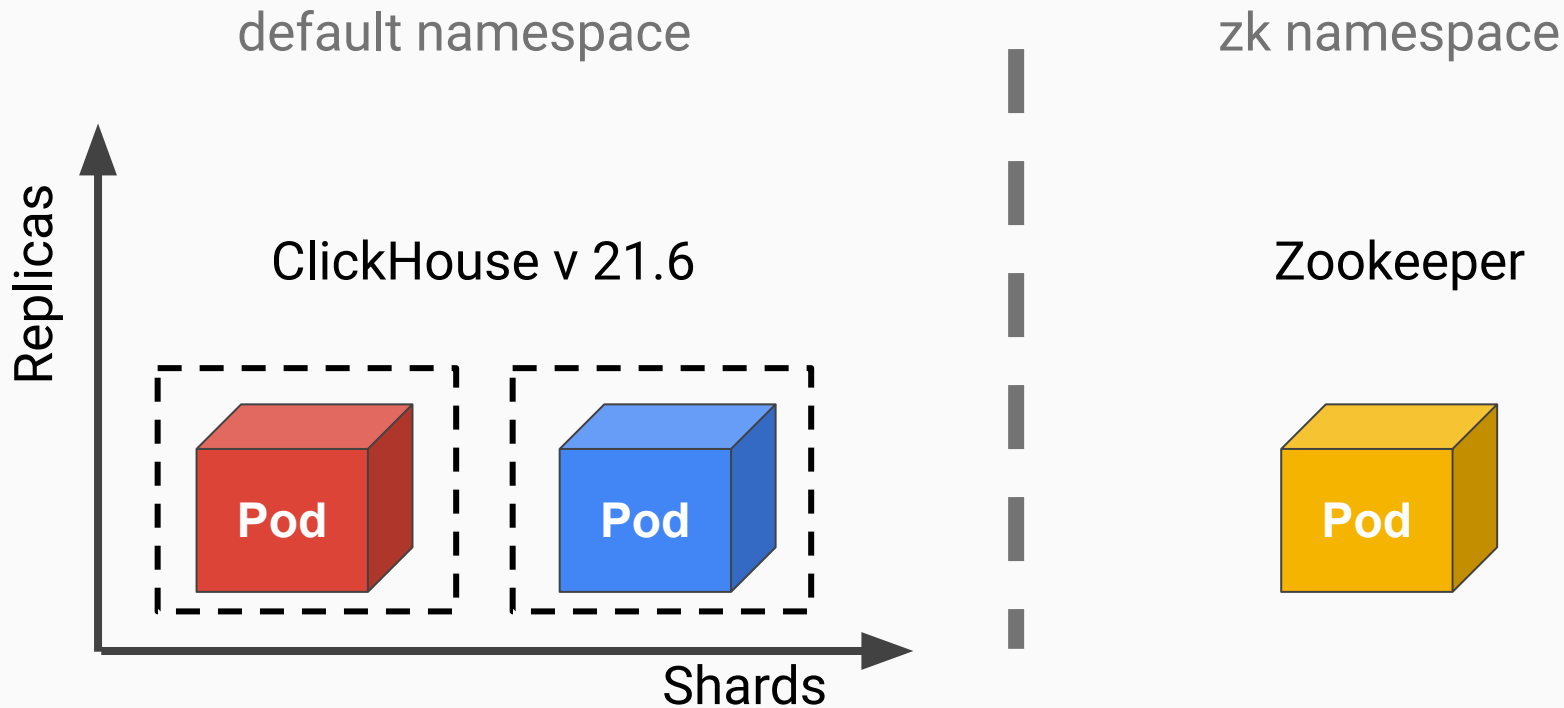
Access ClickHouse

```
kubectl exec -it chi-k8s-demo-0-0-0 clickhouse-client
```

Forward port to external network.

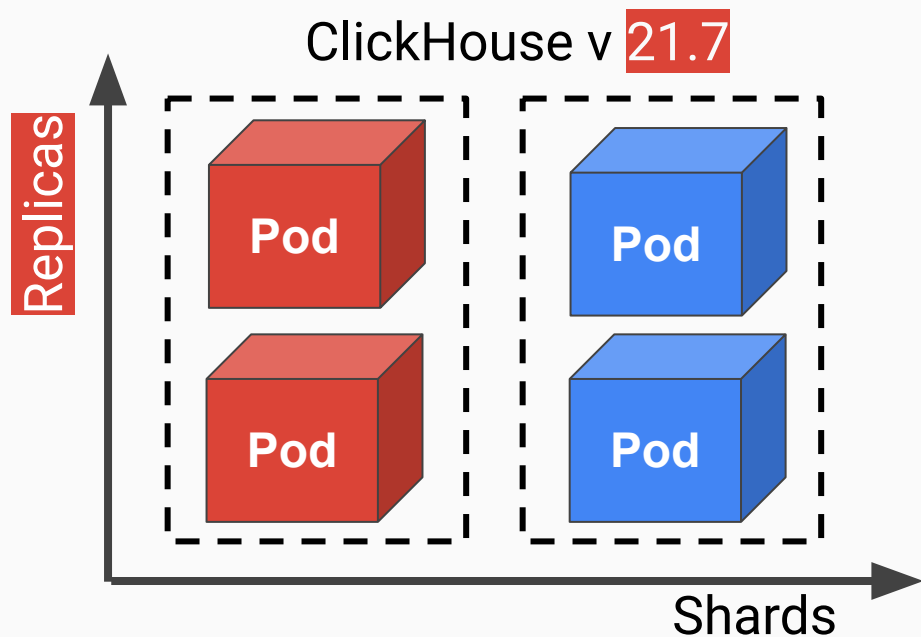
```
kubectl port-forward service/clickhouse-k8s 9000
```

What have we created?



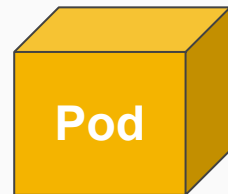
How can we make our creation better?

default namespace



zk namespace

Zookeeper



Increasing the number of replicas/shards

```
apiVersion:  
"clickhouse.altinity.com/v1"  
kind: "ClickHouseInstallation"  
metadata:  
  name: "k8s"  
spec:  
  configuration:  
    clusters:  
      - name: "demo"  
        layout:  
          shardsCount: 3  
          replicasCount: 2  
          . . .
```

More shards!



More replicas!



Adding a user

```
apiVersion: "clickhouse.altinity.com/v1"
kind: "ClickHouseInstallation"
. . .
spec:
  configuration:
    clusters: . . .
    zookeepers: . . .
    users:
      root/password_sha256_hex: 2bb80.....7a25b
      root/networks/ip:
        - ::1
        - 127.0.0.1
      root/quota: default
      root/access_management: 1
```

User definition



**Enable RBAC
management**



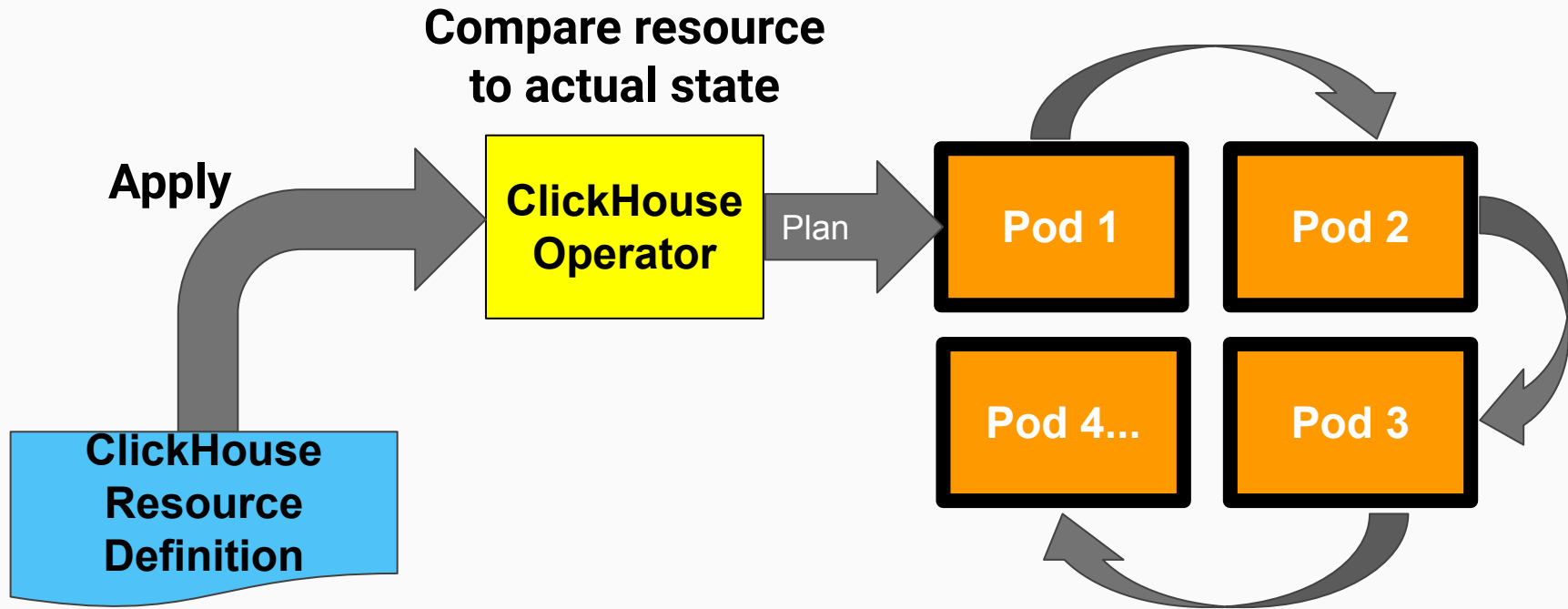
Changing the ClickHouse version

```
templates:  
  podTemplates:  
    - name: clickhouse  
      spec:  
        containers:  
          - name: clickhouse  
            image: yandex/clickhouse-server:21.7  
  volumeClaimTemplates:  
    . . .
```



**New
version**

Configuration changes applied automatically



Update resource definition

Upgrade pods sequentially

Updating the cluster

Learning more

More information and references

- [ClickHouse Kubernetes Operator](#)
- [Altinity Documentation](#)
 - Kubernetes operator & ClickHouse blog
- [Community docs on ClickHouse.tech](#)
 - Everything Clickhouse
- [Altinity Blog](#)
 - Articles about ClickHouse usage including Kubernetes

Thank you!

We're hiring

ClickHouse:

<https://github.com/ClickHouse/ClickHouse>

Altinity Website:

<https://www.altinity.com>

Altinity.Cloud

<https://altinity.com/cloud>