PIWIK PRO

ClickHouse Operator

ClickHouse

ClickHouse

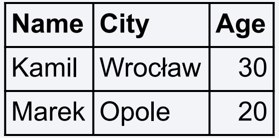
Column oriented database for online analytical processing

ClickHouse

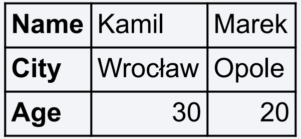
Column oriented database for online analytical processing

ClickHouse

- Column oriented database
  - Row - optimized for reading and writing rows efficiently
  - **Column** - optimized for reading and computing on columns efficiently
- Online analytical processing
  - OLTP - online **transaction** processing
  - **OLAP** - online **analytical** processing



Name	City	Age
Kamil	Wrocław	30
Marek	Opole	20

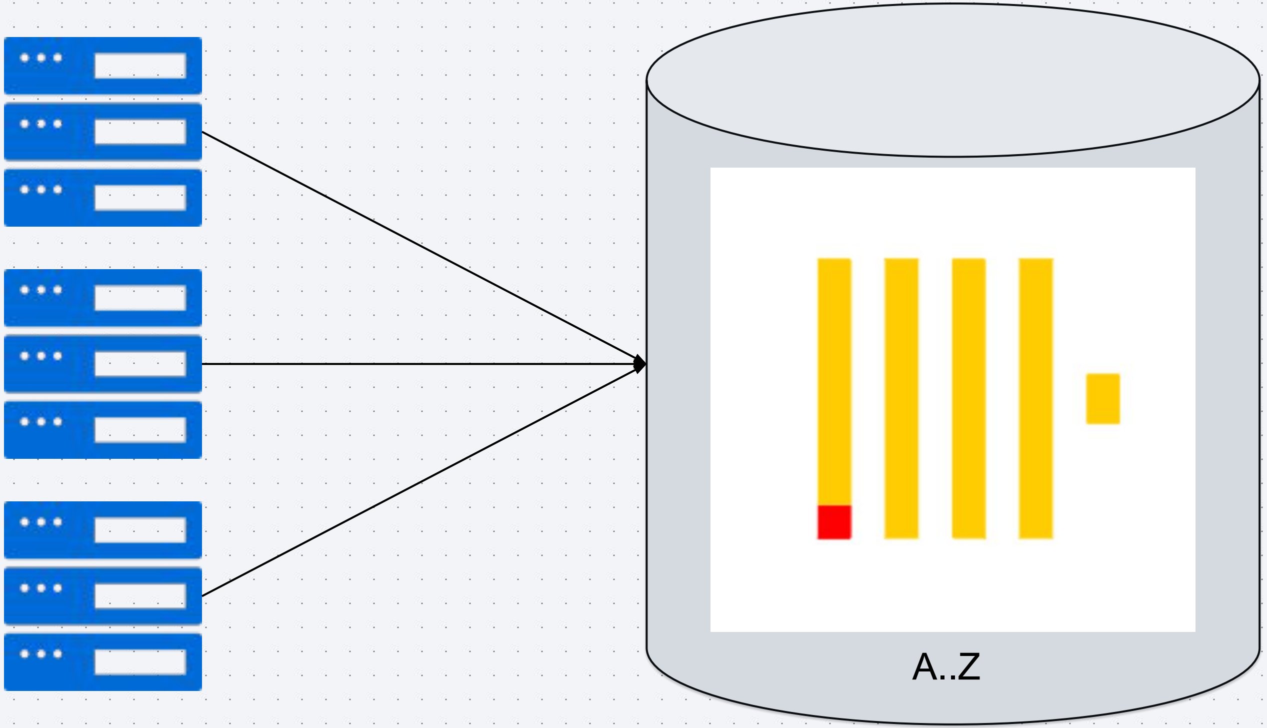


Name	Kamil	Marek
City	Wrocław	Opole
Age	30	20

ClickHouse

- Fast
- **Linear scalability** - scales to many petabytes
- Fault tolerant - **async master-master replication**
- Open source
- Developed by Yandex
- No transactions

Single server

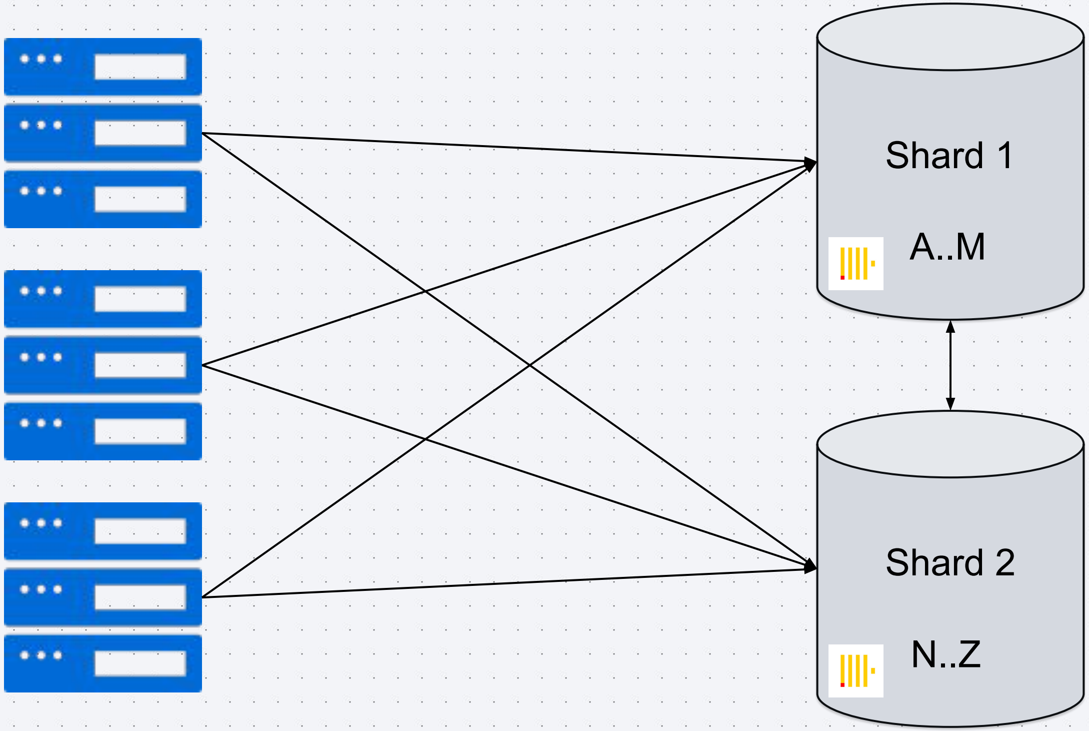


- Simple
- All applications use one server
- Not a real production setup

Problems:

- Limited scaling - only vertical
- Large dataset might not fit
- No High Availability

Sharding



- Different part of data on each server
- Distributed tables
- Any sharding key

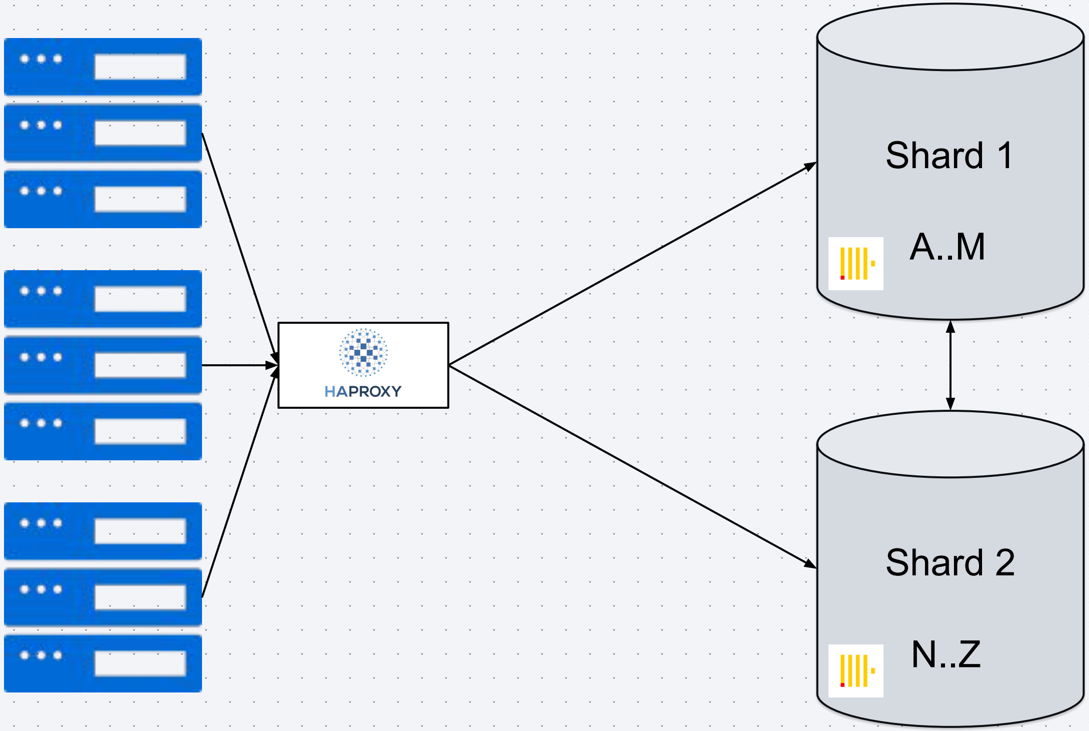
Problems:

✓ Horizontal and vertical scaling

✓ Large dataset might not fit

- No High Availability
- Load distribution on application side

Sharding



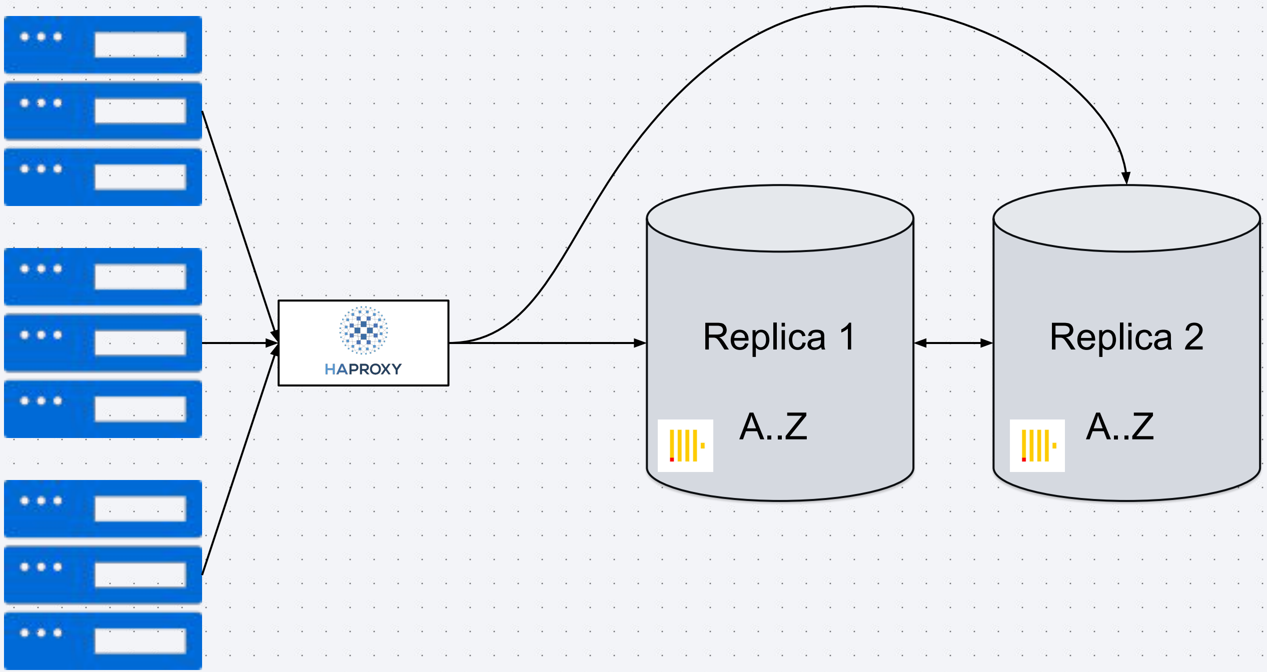
- Different part of data on each server
- Distributed tables
- Any sharding key

Problems:

- Horizontal and vertical scaling
- Large dataset might not fit
- No High Availability

✓ Load distribution on application side

Replication

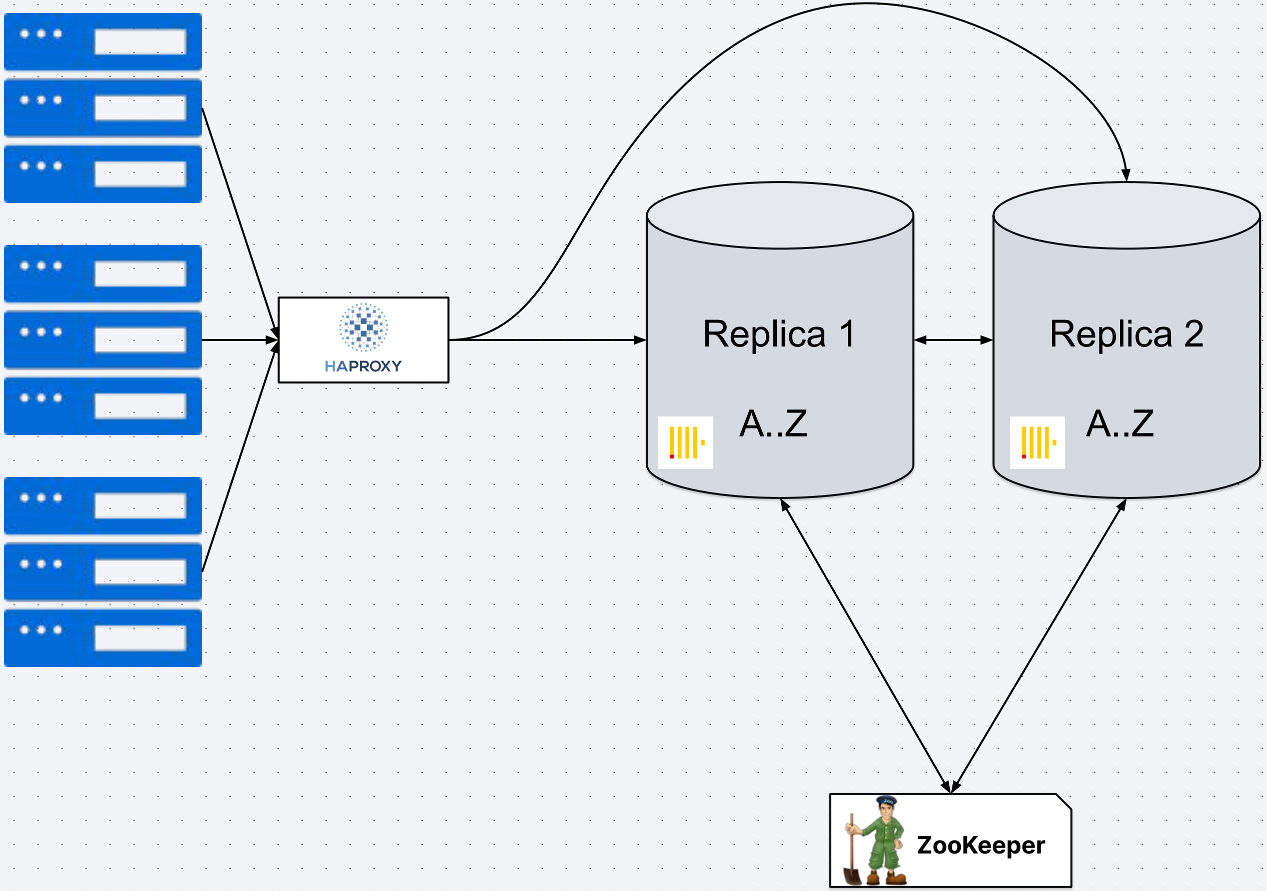


- Same data on all servers
- Each replica can handle reads
- **Master-master** replication - writes on all replicas

Problems:

✓ High Availability

Replication

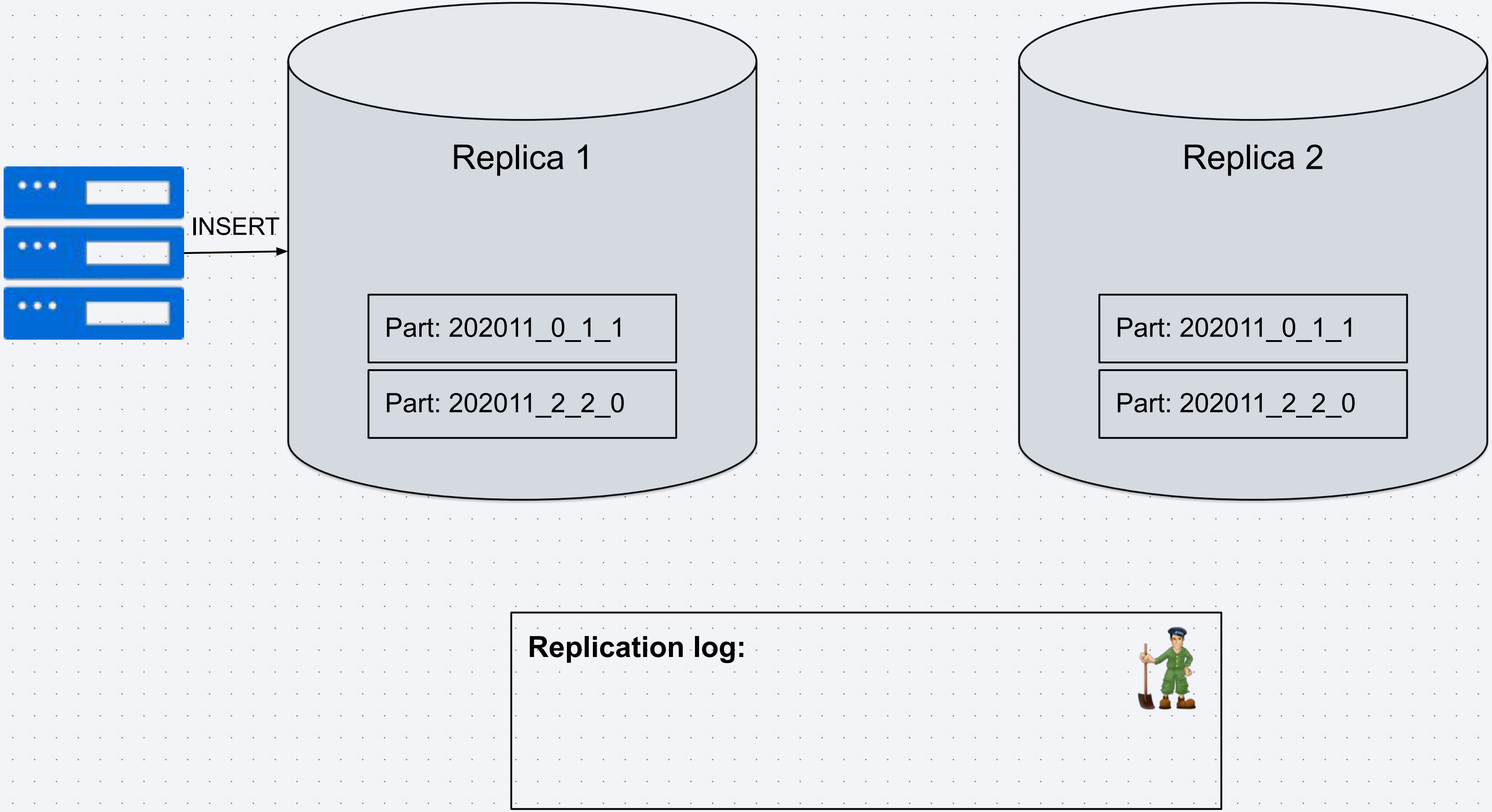


- Same data on all servers
- Each replica can handle reads
- **Master-master** replication - writes on all replicas
- ZooKeeper required for replication - only for inserts

Problems:

- High Availability

Replication in-depth - insert



INSERT

Replica 1

Part: 202011_0_1_1

Part: 202011_2_2_0

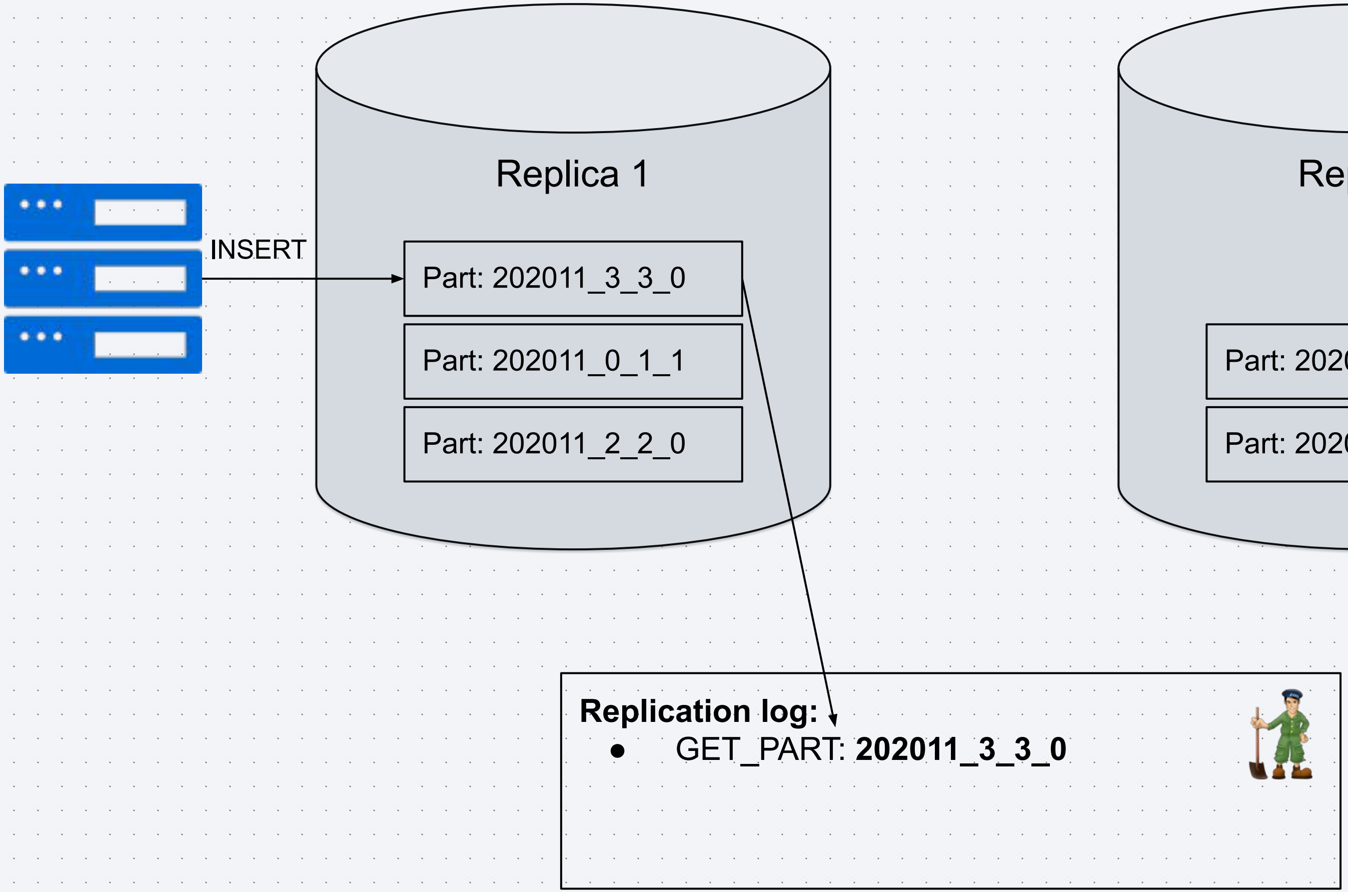
Replica 2

Part: 202011_0_1_1

Part: 202011_2_2_0

Replication log:

Replication in-depth - insert



INSERT

Replica 1

Part: 202011_3_3_0

Part: 202011_0_1_1

Part: 202011_2_2_0

Replica 2

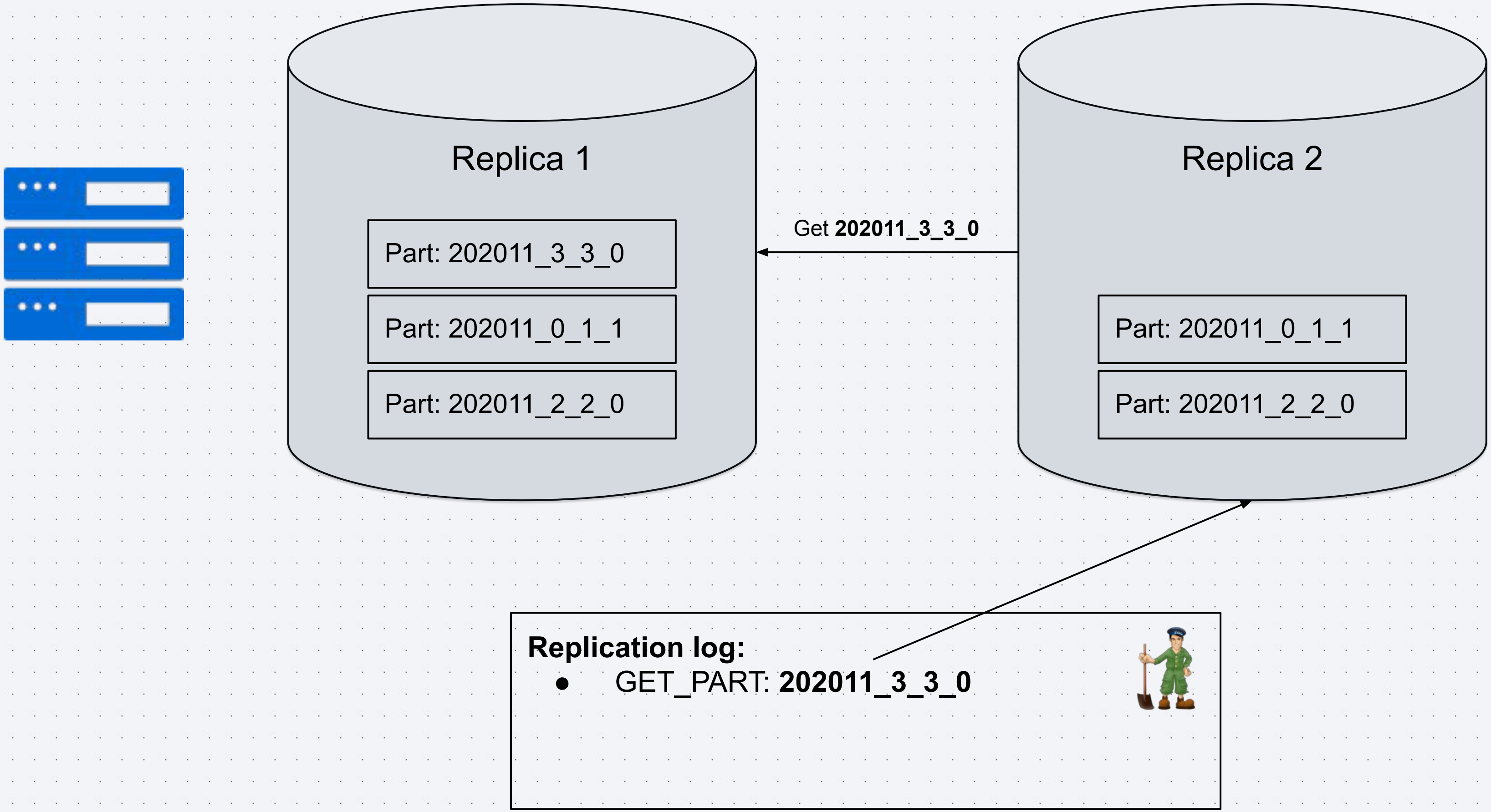
Part: 202011_0_1_1

Part: 202011_2_2_0

Replication log:

- GET_PART: **202011_3_3_0**

Replication in-depth - insert



Replica 1

Part: 202011_3_3_0

Part: 202011_0_1_1

Part: 202011_2_2_0

Get **202011_3_3_0**

Replica 2

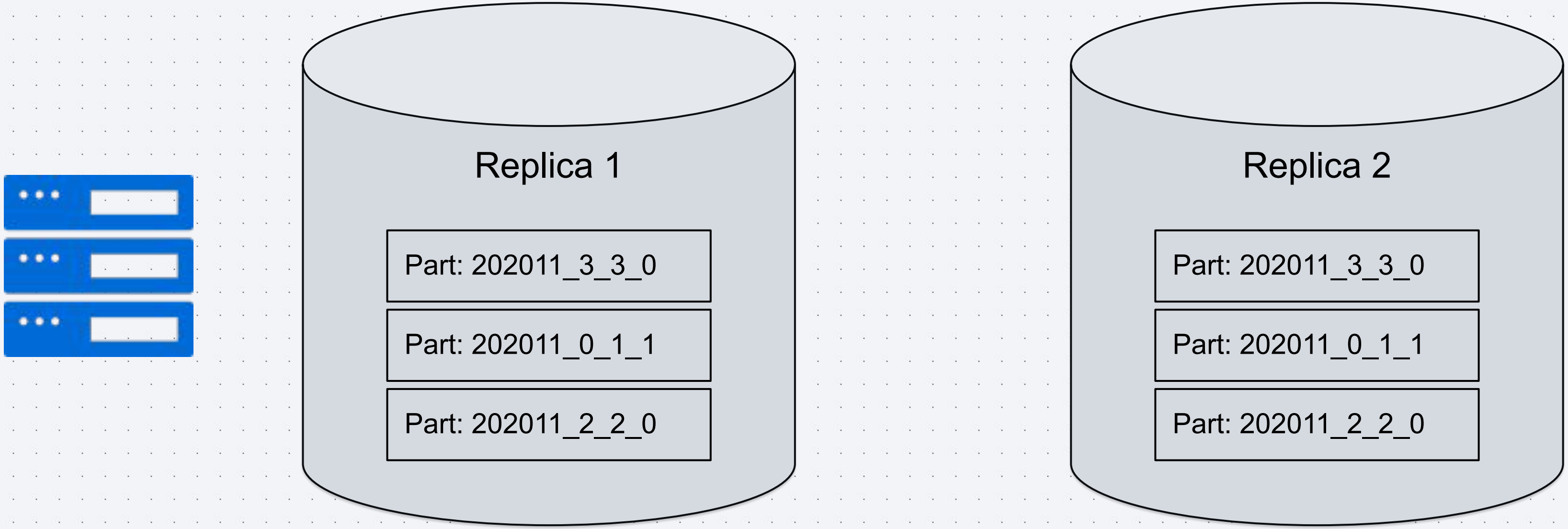
Part: 202011_0_1_1

Part: 202011_2_2_0

Replication log:

- GET_PART: **202011_3_3_0**

Replication in-depth - insert



Replica 1

Part: 202011_3_3_0

Part: 202011_0_1_1

Part: 202011_2_2_0

Replica 2

Part: 202011_3_3_0

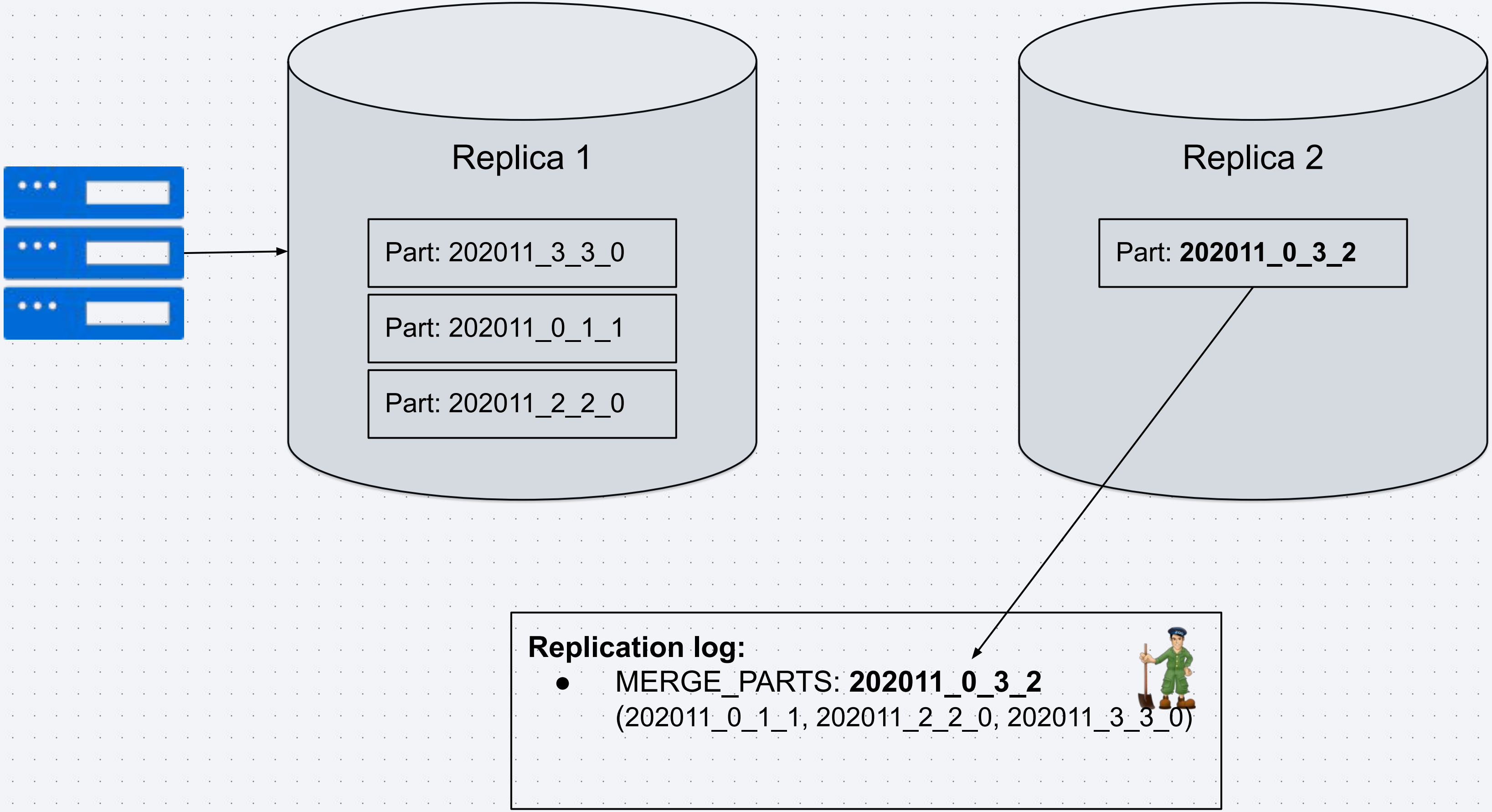
Part: 202011_0_1_1

Part: 202011_2_2_0

Replication log:

- GET_PART: **202011_3_3_0**

Replication in-depth - merging parts



Replica 1

Part: 202011_3_3_0

Part: 202011_0_1_1

Part: 202011_2_2_0

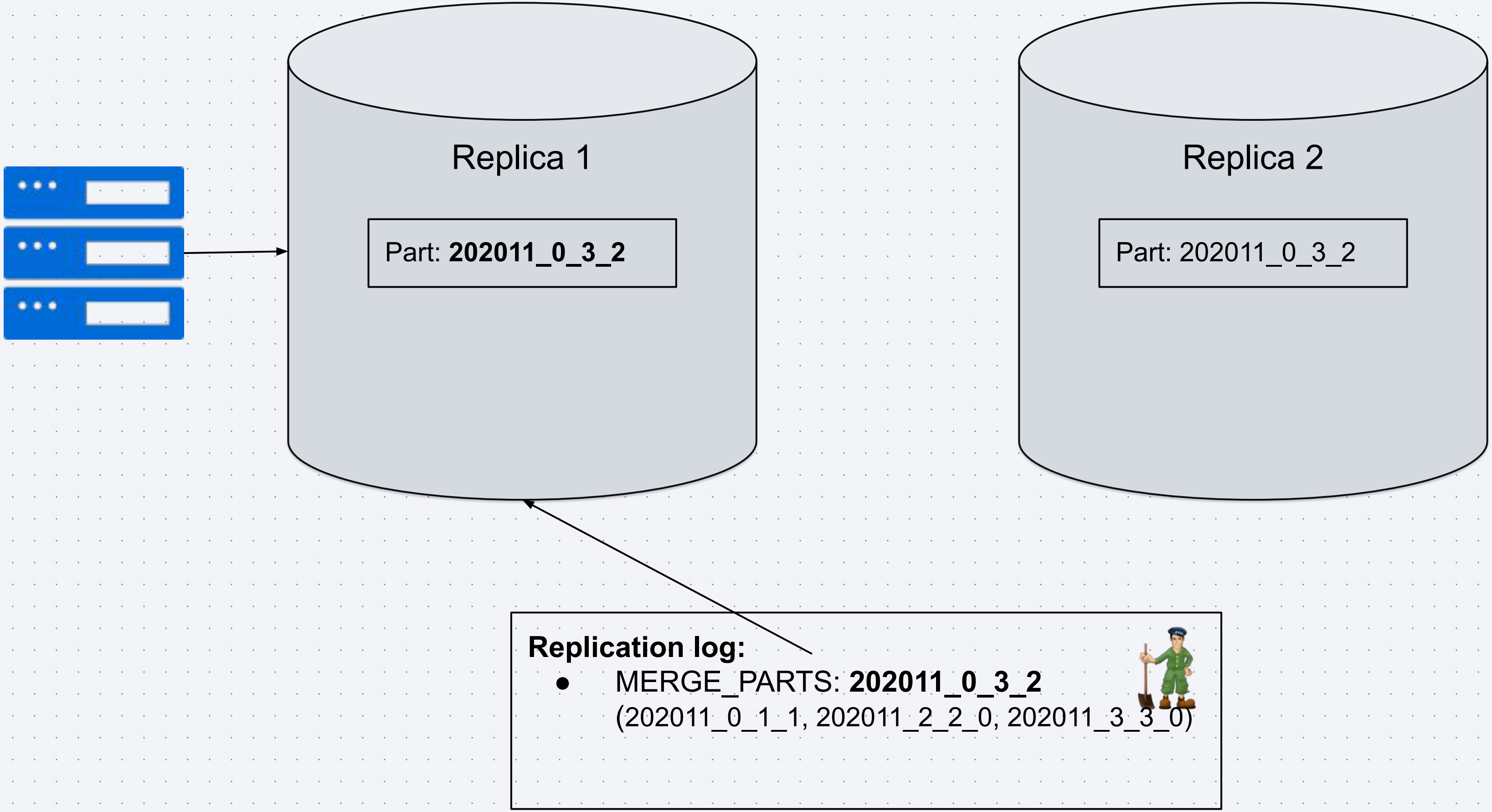
Replica 2

Part: **202011_0_3_2**

Replication log:

- MERGE_PARTS: **202011_0_3_2** (202011_0_1_1, 202011_2_2_0, 202011_3_3_0)

Replication in-depth - merging parts



Replica 1

Part: **202011_0_3_2**

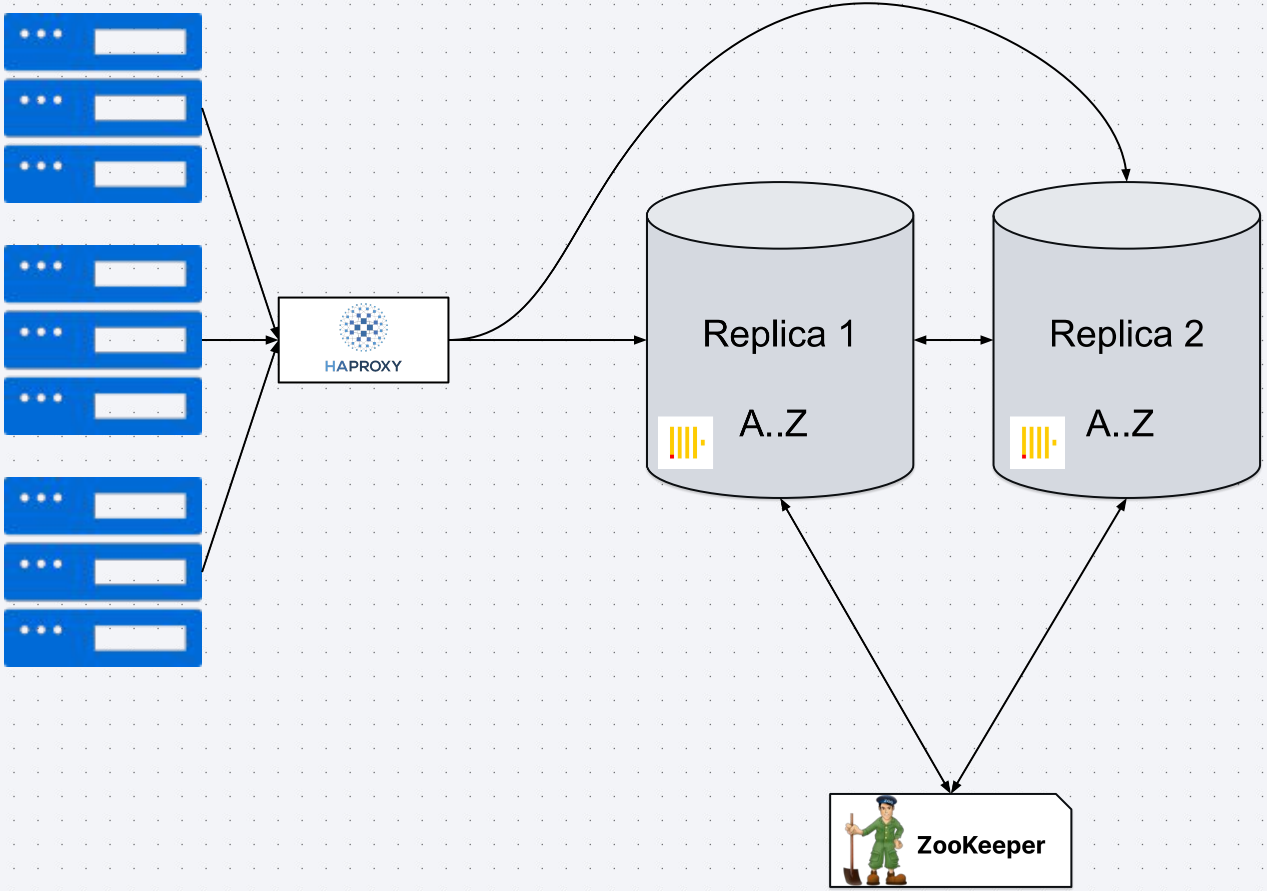
Replica 2

Part: 202011_0_3_2

Replication log:

- MERGE_PARTS: **202011_0_3_2** (202011_0_1_1, 202011_2_2_0, 202011_3_3_0)

Replication

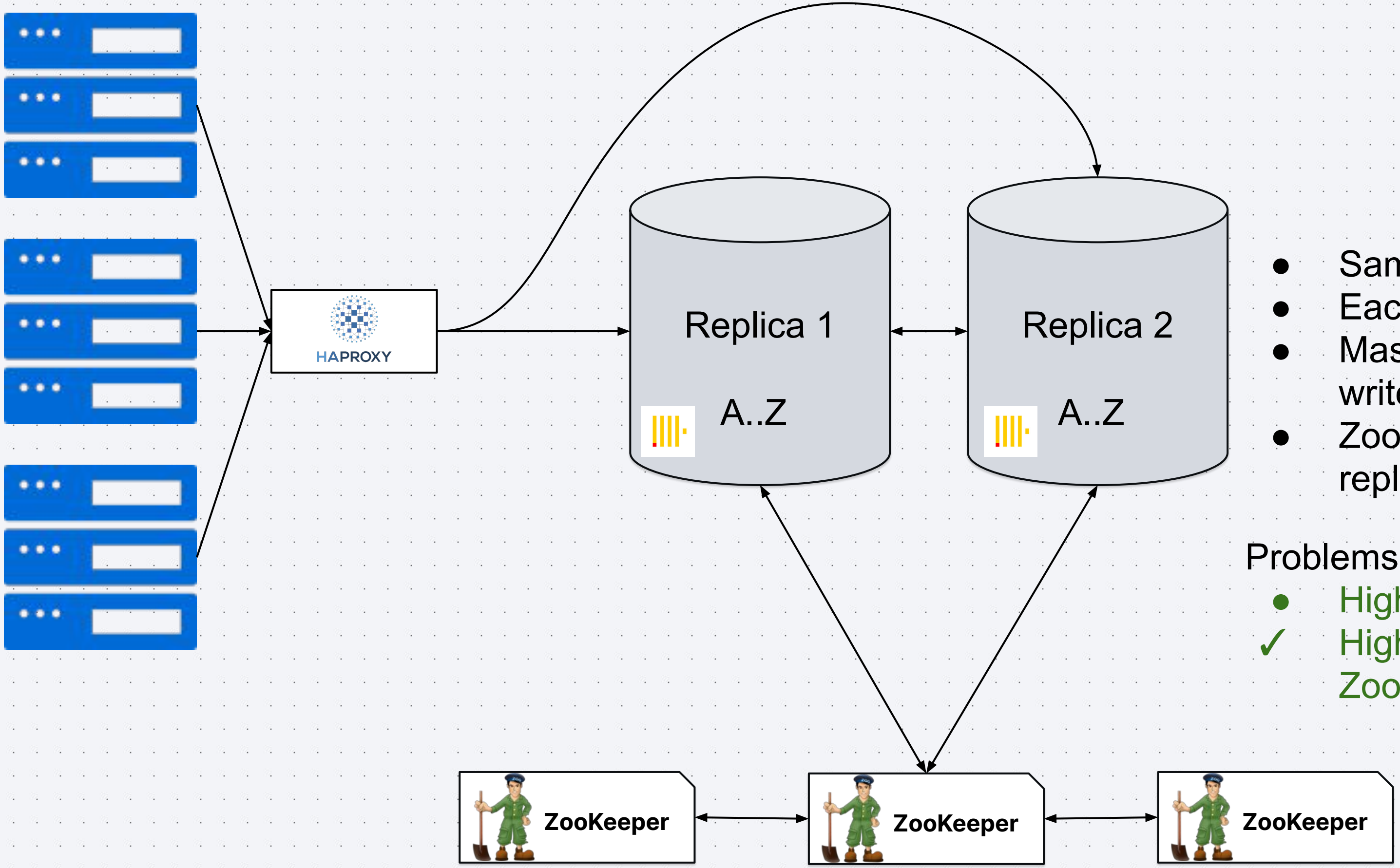


- Same data on all servers
- Each replica can handle reads
- Master-master replication - writes on all replicas
- ZooKeeper required for replication - only for inserts

Problems:

- 🟩 High Availability
- 🟥 High Availability for ZooKeeper

Replication



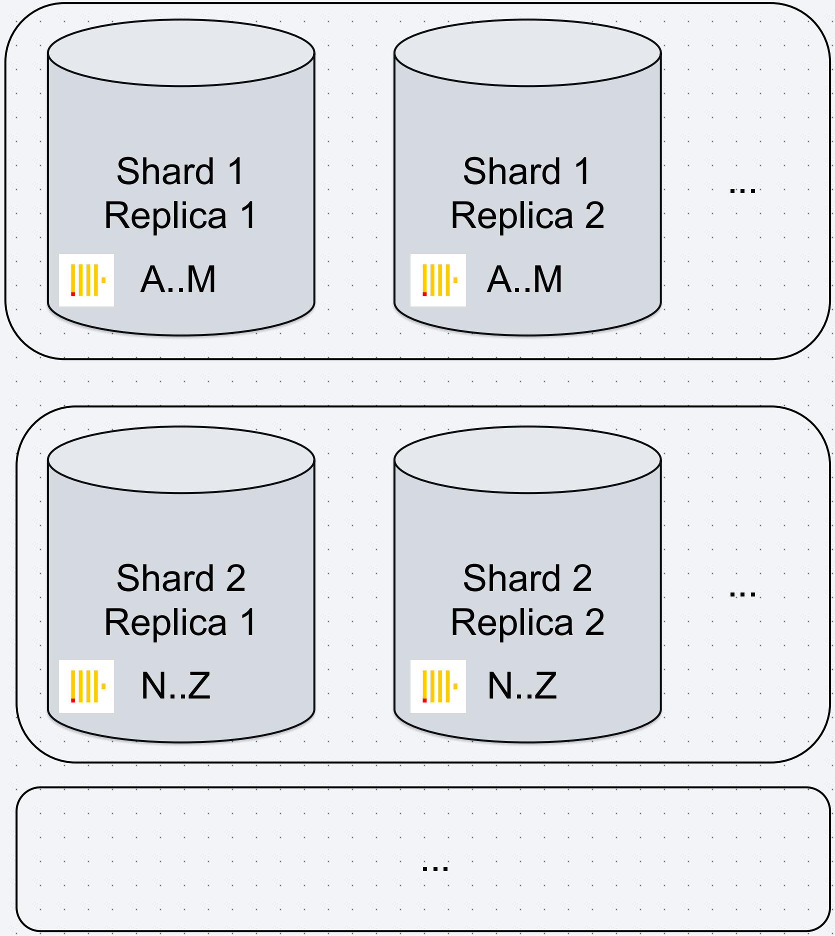
- Same data on all servers
- Each replica can handle reads
- Master-master replication - writes on all replicas
- ZooKeeper required for replication - only for inserts

Problems:

- High Availability

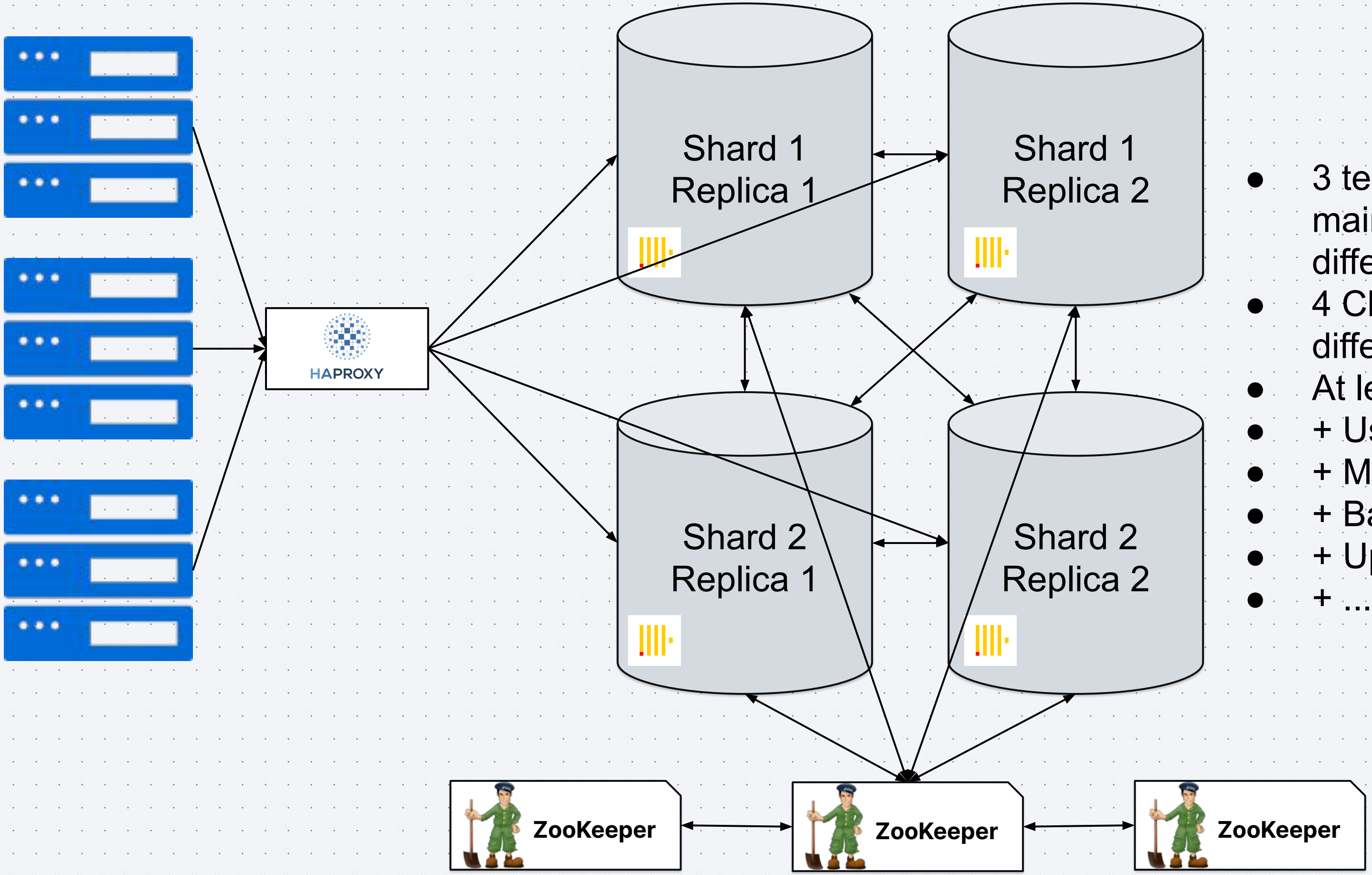
✓ High Availability also for ZooKeeper

Sharding & Replication



- Horizontal and vertical scaling
- High availability
- Master-master replication - writes and reads on all nodes
- Up to petabytes of data

Enterprise setup

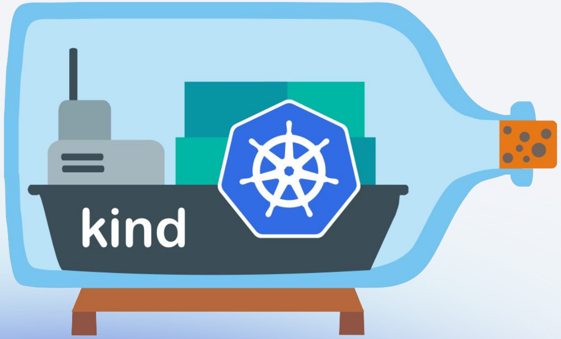


- 3 technologies - each maintained and scaled differently
- 4 ClickHouse servers in different roles
- At least 3 Zookeepers
- + Users mgmt
- + Monitoring
- + Backups
- + Upgrades
- + ...

Kubernetes (k8s)

How to run k8s locally

- minikube (minikube.sigs.k8s.io)
- kind (kind.sigs.k8s.io)
- k3d (k3d.io)



K8s operators

Operators in promil

- Creating MySQL database and users
- Creating Minio buckets
- Creating and configuration RabbitMQ queues
- Taking and restoring backups
- Deploying ClickHouse i Elasticsearch clusters
- Encryption of secrets (Sealed Secrets)

K8s operator - DIY

- Operator SDK (go, ansible, helm)
- Kopf (python)

Altinity
ClickHouse Operator

Who is behind it?

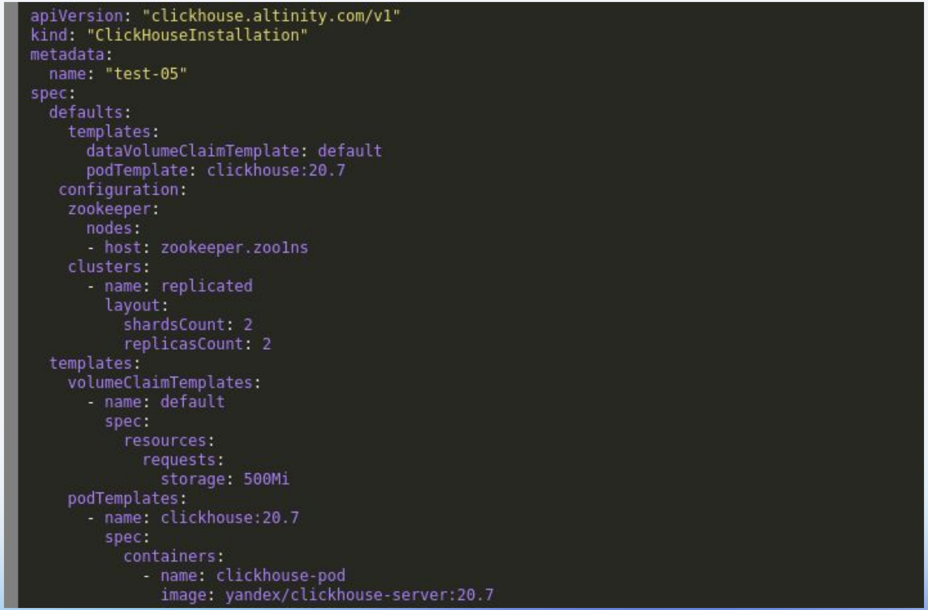
Altinity

- Altinity
  - Enterprise support for ClickHouse
  - Clickhouse feature development
- Founders:
  - Alexander Zaitsev
  - Peter Zaitsev and Vadim Tkachenko
- Great webinars on YouTube - check them out!

PERCONA

ClickHouseInstallation

Custom Resource



```yaml
apiVersion: "clickhouse.altinity.com/v1"
kind: "ClickHouseInstallation"
metadata:
  name: "test-05"
spec:
  defaults:
    templates:
      dataVolumeClaimTemplate: default
      podTemplate: clickhouse:20.7
   configuration:
    zookeeper:
      nodes:
      - host: zookeeper.zoo1ns
    clusters:
      - name: replicated
        layout:
          shardsCount: 2
          replicasCount: 2
  templates:
    volumeClaimTemplates:
      - name: default
        spec:
          resources:
            requests:
              storage: 500Mi
    podTemplates:
      - name: clickhouse:20.7
        spec:
          containers:
            - name: clickhouse-pod
              image: yandex/clickhouse-server:20.7
```

Database in container

- ✓ **Easier maintenance** - operators can use the same tools for application and database
- ✓ **Autoscaling** - adding additional replicas on hot hours
- ✓ **Similar setup on each platform** - Cloud/On-prem compatibility
- ✓ **Rollup upgrade and patching** - safe upgrades with easy way to rollback
- ✓ **More alike dev, staging and production setups** - more likely to identify issues before production deploy

Database in container

- ❌ Containers are meant to be **stateless**
- ❌ **Maintaining persistence** - after container shutdown data needs to be attached to another container (even on another node) in identical form
- ❌ **Availability** - Containers can be restarted at any time by oriestrator causing downtime
- ❌ Container will add an additional **performance overhead**
- ❌ **Security** - breach in other container may lead to compromising data
- ❌ **Maturity** - Is Kubernetes mature enough for running production databases?

Database in container

✓ Containers are meant to be **stateless**
Data mounted in container

✓ **Maintaining persistence** - after container shutdown data needs to be attached to another container (even on another node) in identical form
Assured with PersistentVolumes and Azure Managed Disks (on cloud)

✓ **Availability** - Containers can be restarted at any time by oriestrator causing downtime
Production setups always with HA

✗ Container will add an additional **performance overhead**
Yes, but how much?

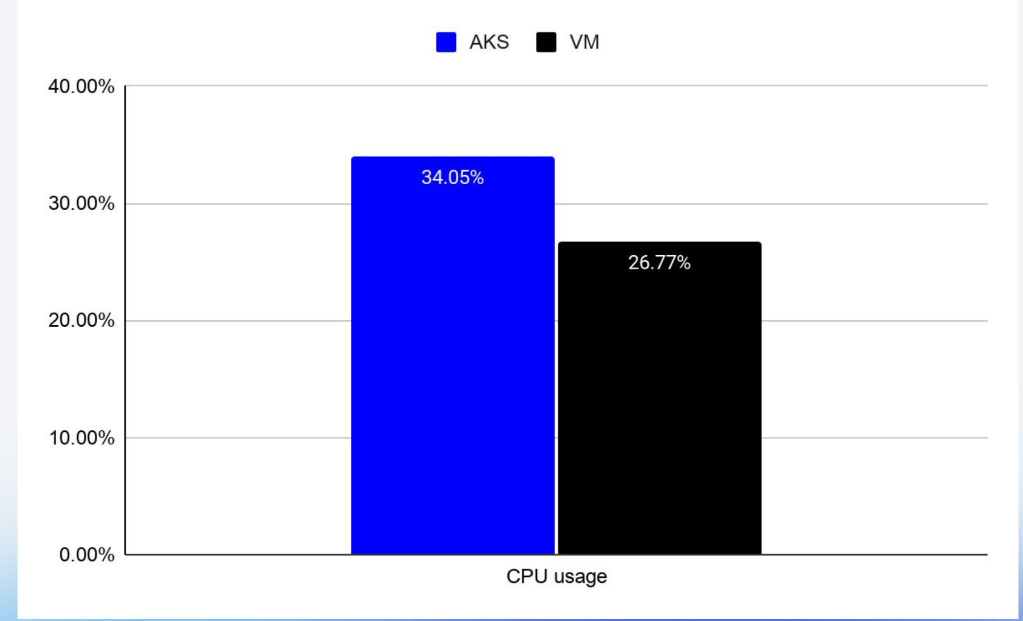
✓ **Security** - breach in other container may lead to compromising data
Separate k8s namespace and NetworkPolicies

✓ **Maturity** - Is Kubernetes mature enough for running production databases?
Altinity sells ClickHouse on their k8s in SaaS model

Performance container vs on host

benchmarks

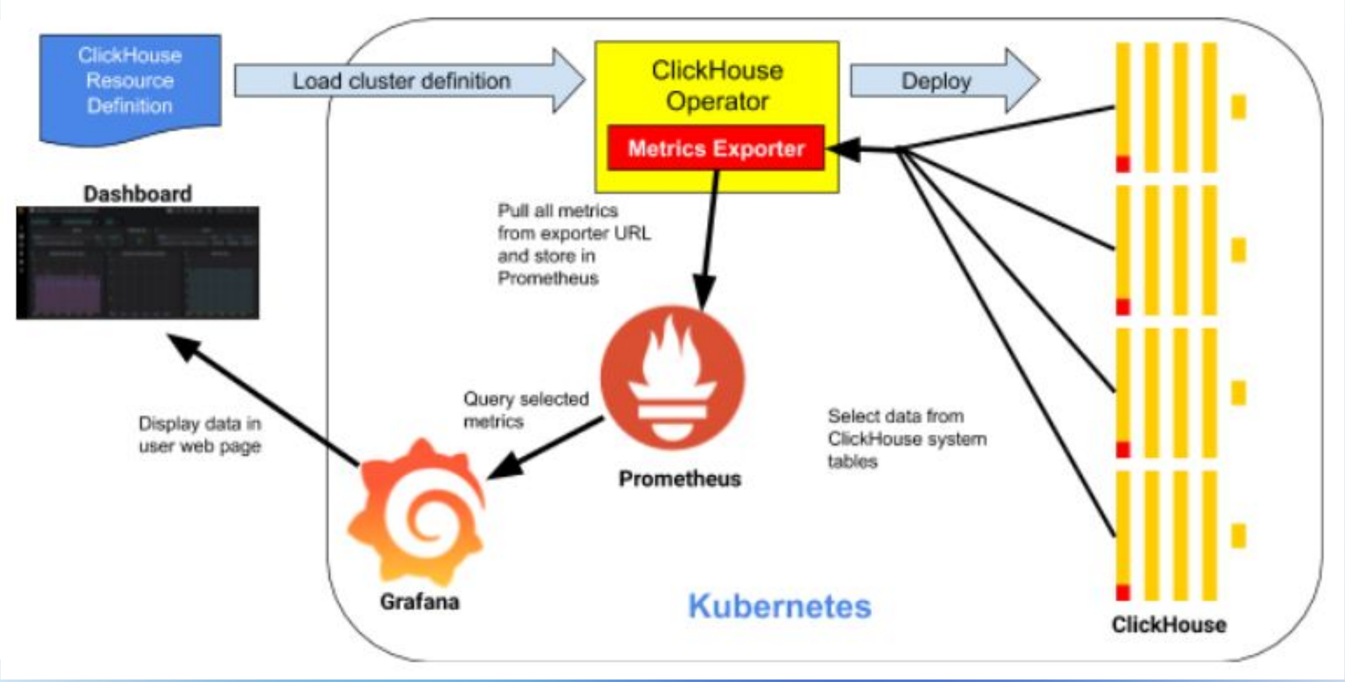
CPU usage



Monitoring

- ClickHouse Operator + Prometheus & Grafana
- Metrics, Alerts and Dashboards provided by Clickhouse Operator
- Automatically adjust metrics collection based on cluster settings

Monitoring


ClickHouse Resource Definition
Load cluster definition
ClickHouse Operator
Metrics Exporter
Deploy
Dashboard
Pull all metrics from exporter URL and store in Prometheus
Display data in user web page
Query selected metrics
Prometheus
Select data from ClickHouse system tables
Grafana
Kubernetes
ClickHouse

Source: https://altinity.com/wp-content/uploads/2020/07/Arch-Screenshotfrom2020-04-1517-15-27.png

Monitoring demo

Finito