

Running Production Kafka Clusters in Kubernetes

Balthazar Rouberol - Datadog

London Kafka Summit - May 2019



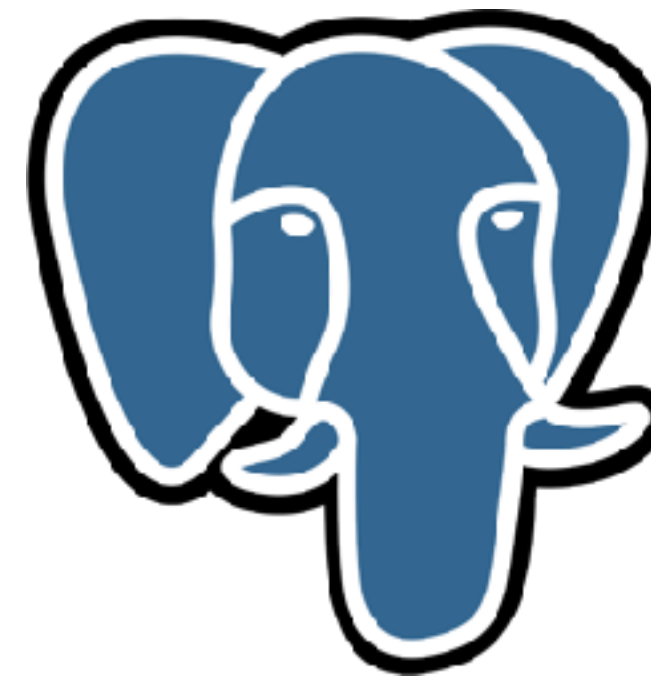
Agenda

- Why are we even doing this?
- Established Kafka tooling and practices at Datadog
- Description of important Kubernetes resources
- Deployment of Kafka in Kubernetes
- Description of routine operations

Who am I?



Data Reliability Engineer @ Datadog



Background

- New instance of Datadog in Europe
- Completely independant and isolated from the existing system
- Leave legacy behind and start fresh
- Have every team use it

Objectives

- Dedicated resources
- Local storage
- Clusters running up to hundreds of instances
- Rack awareness
- Unsupervised (when possible) operations backed by `kafka-kit` *
- Simplified configuration

* <https://github.com/datadog/kafka-kit>



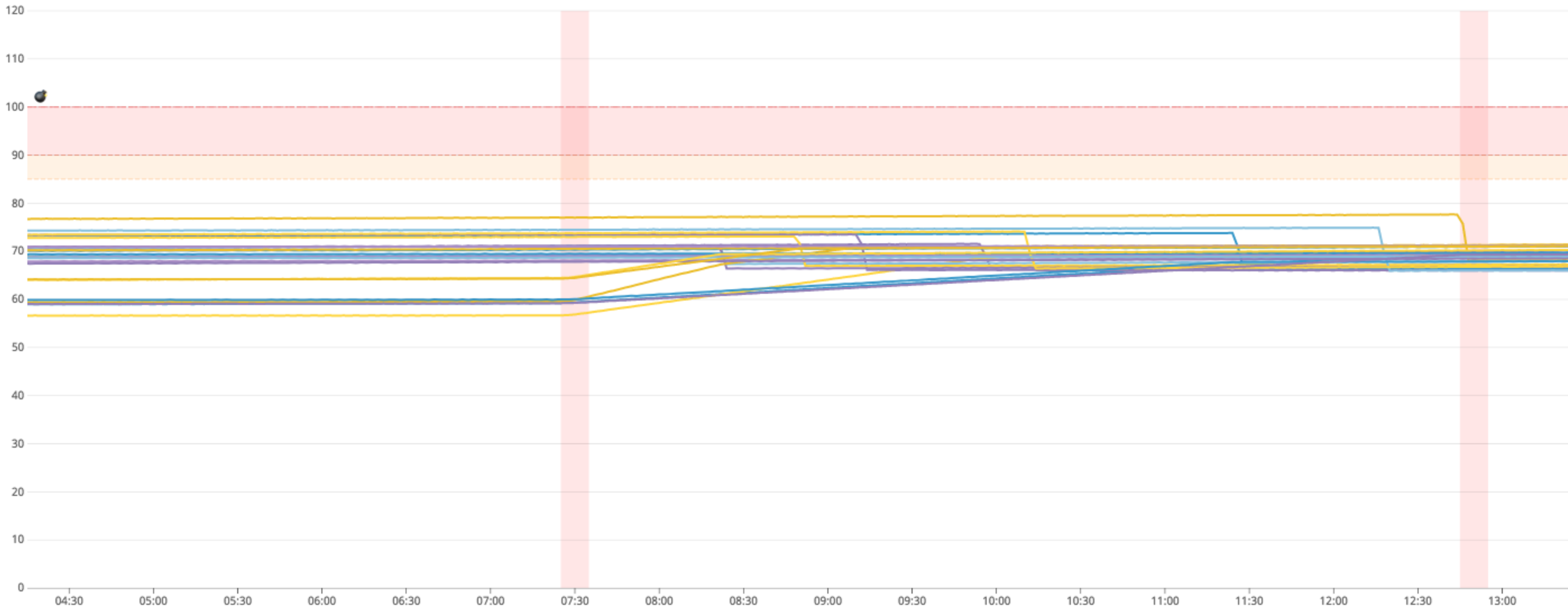
Tooling

Kafka-Kit: scaling operations

- <https://github.com/datadog/kafka-kit>
- `topicmappr`:
 - partition to broker mapping
 - failed broker replacement
 - storage-based cluster rebalancing

Disk Used (/data)

9 h Nov 14, 4:15 am - Nov 14, 1:24 pm



Kafka-Kit: scaling operations

- <https://github.com/datadog/kafka-kit>
- `topicmapper`:
 - partition to broker mapping
 - failed broker replacement
 - storage-based cluster rebalancing
- `autothrottle`: replication auto-throttling

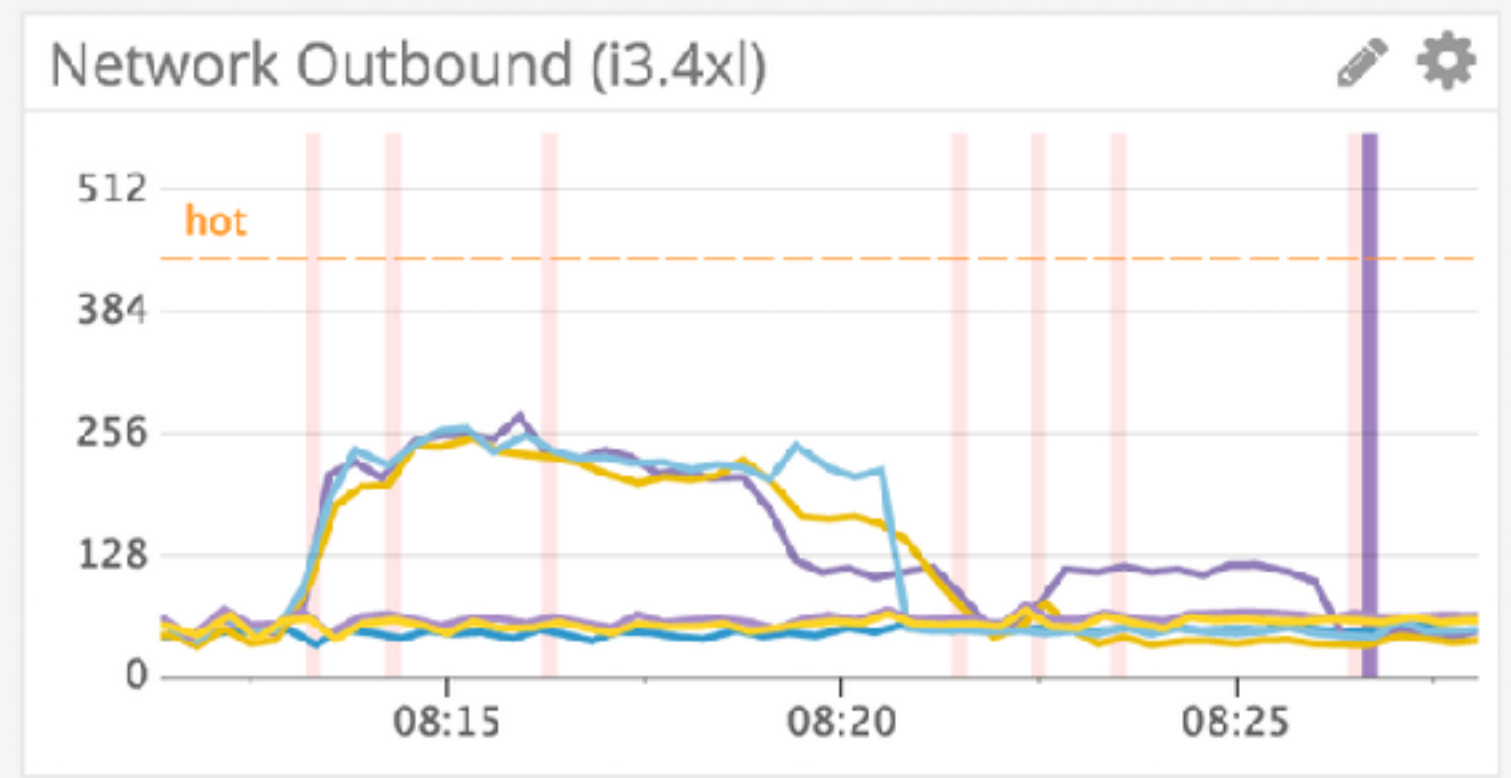
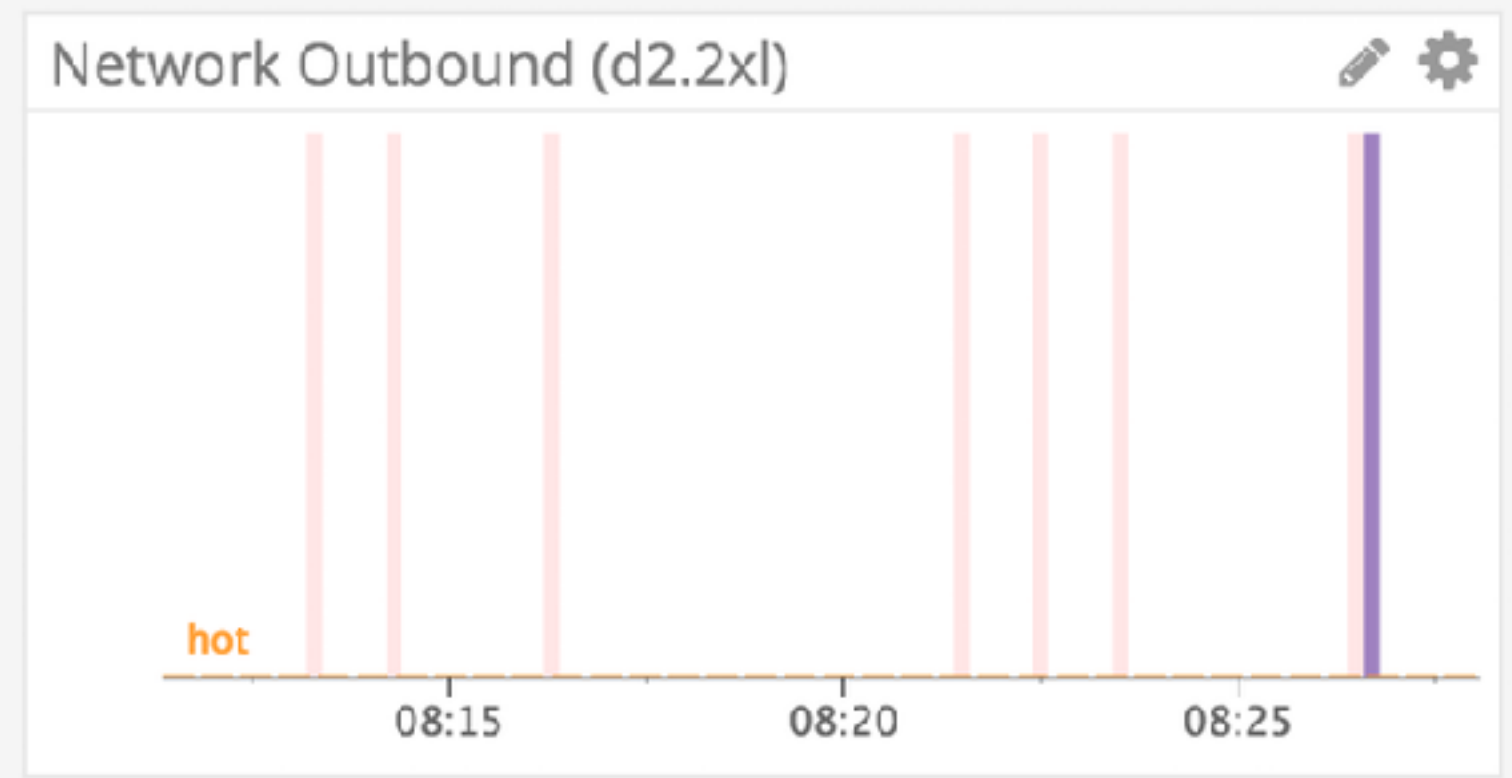
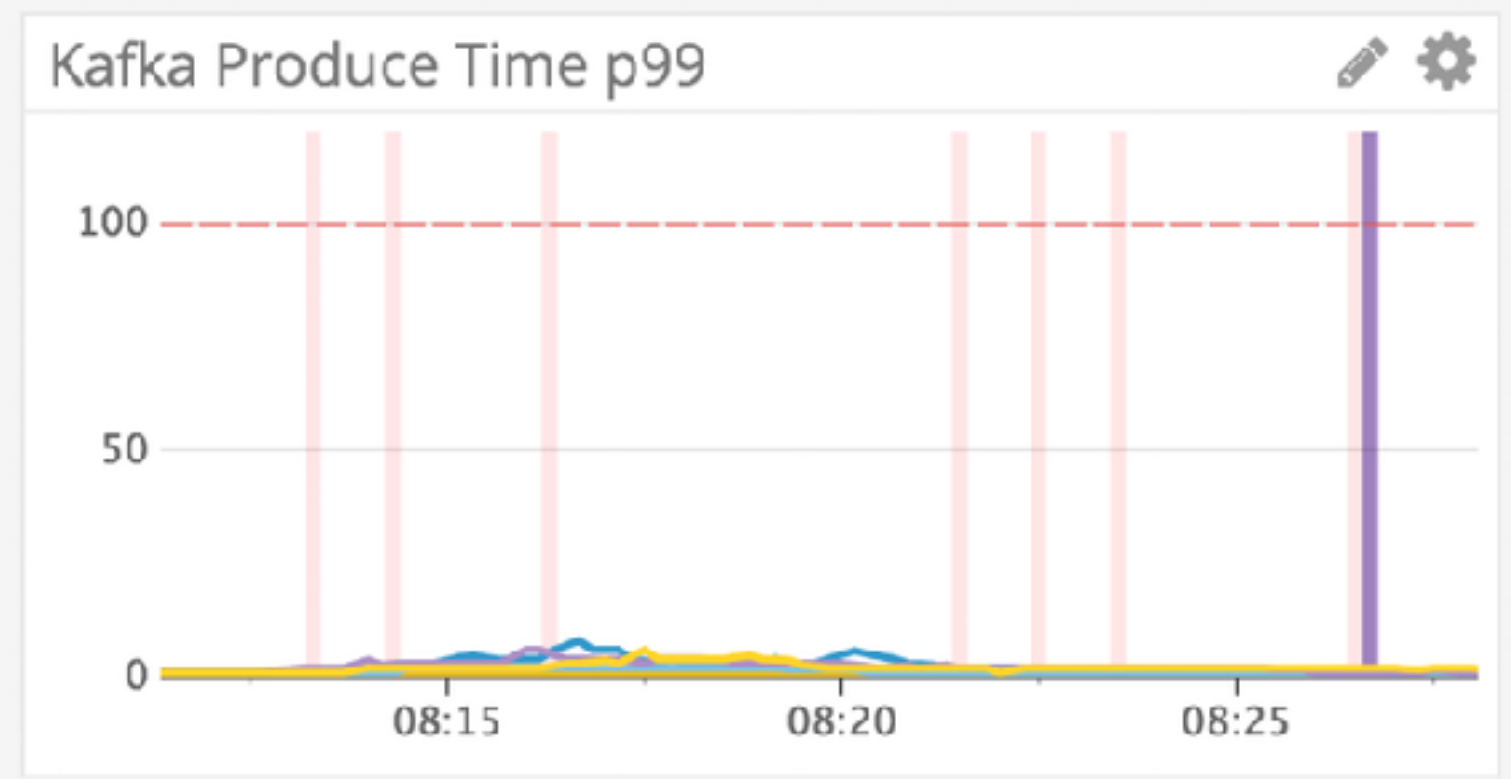
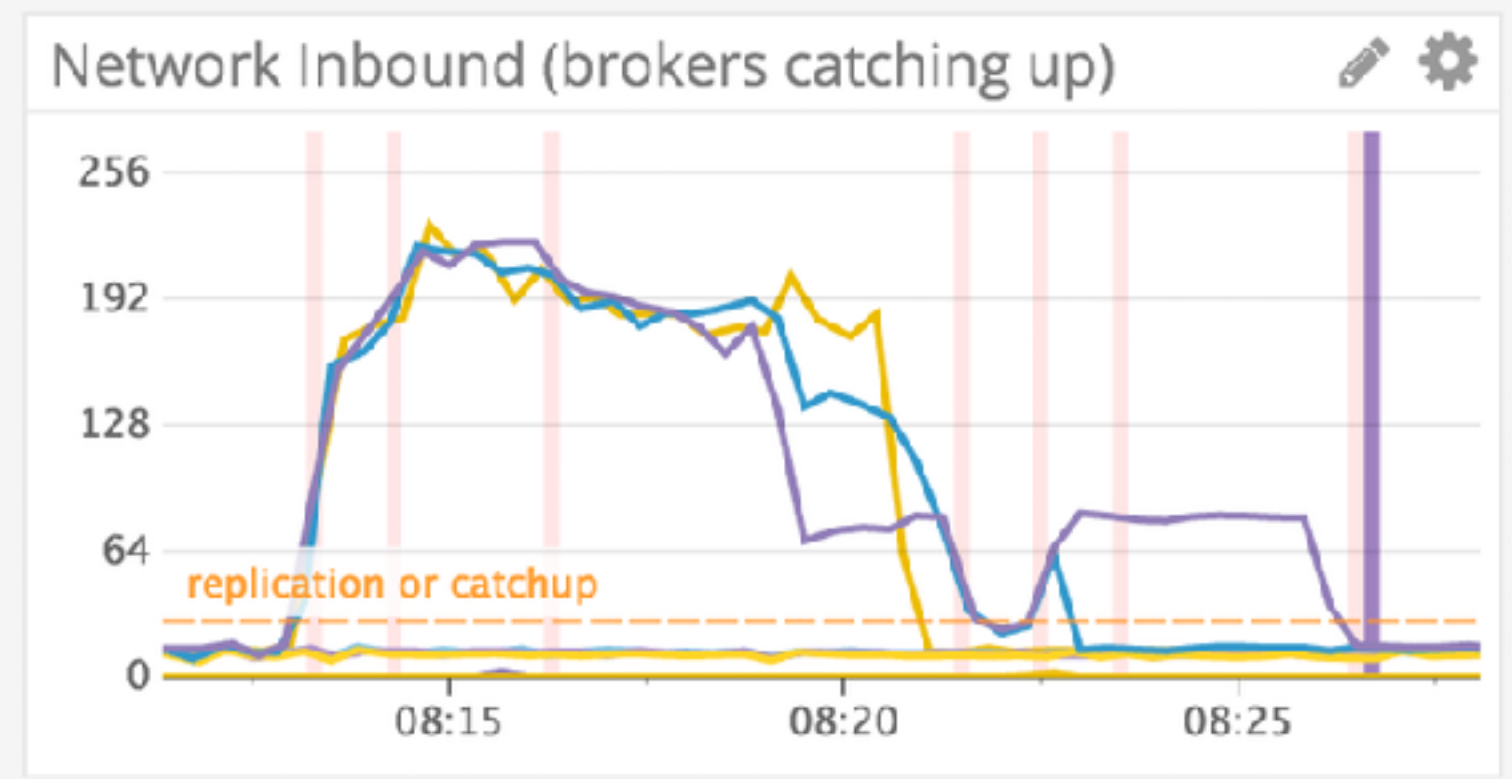
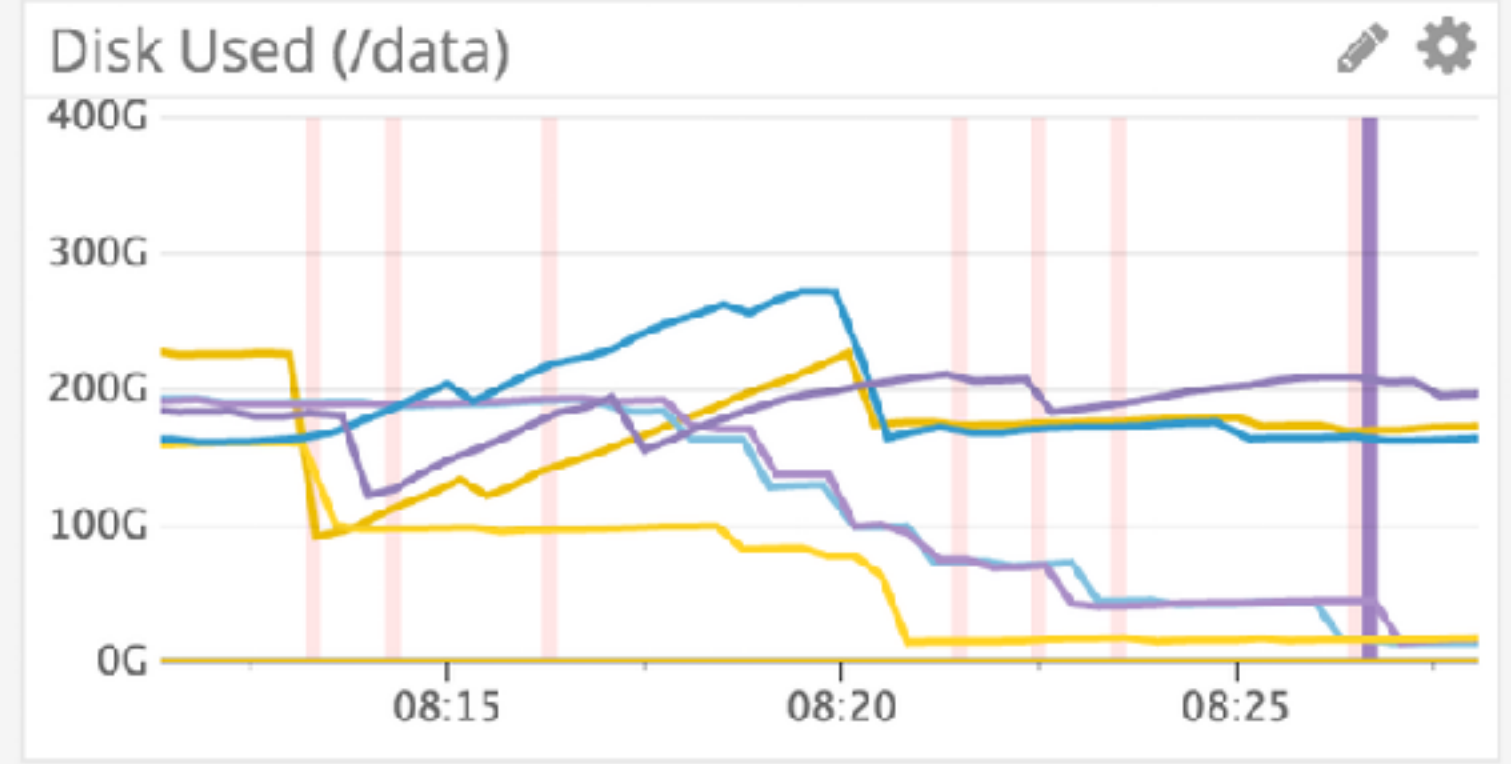
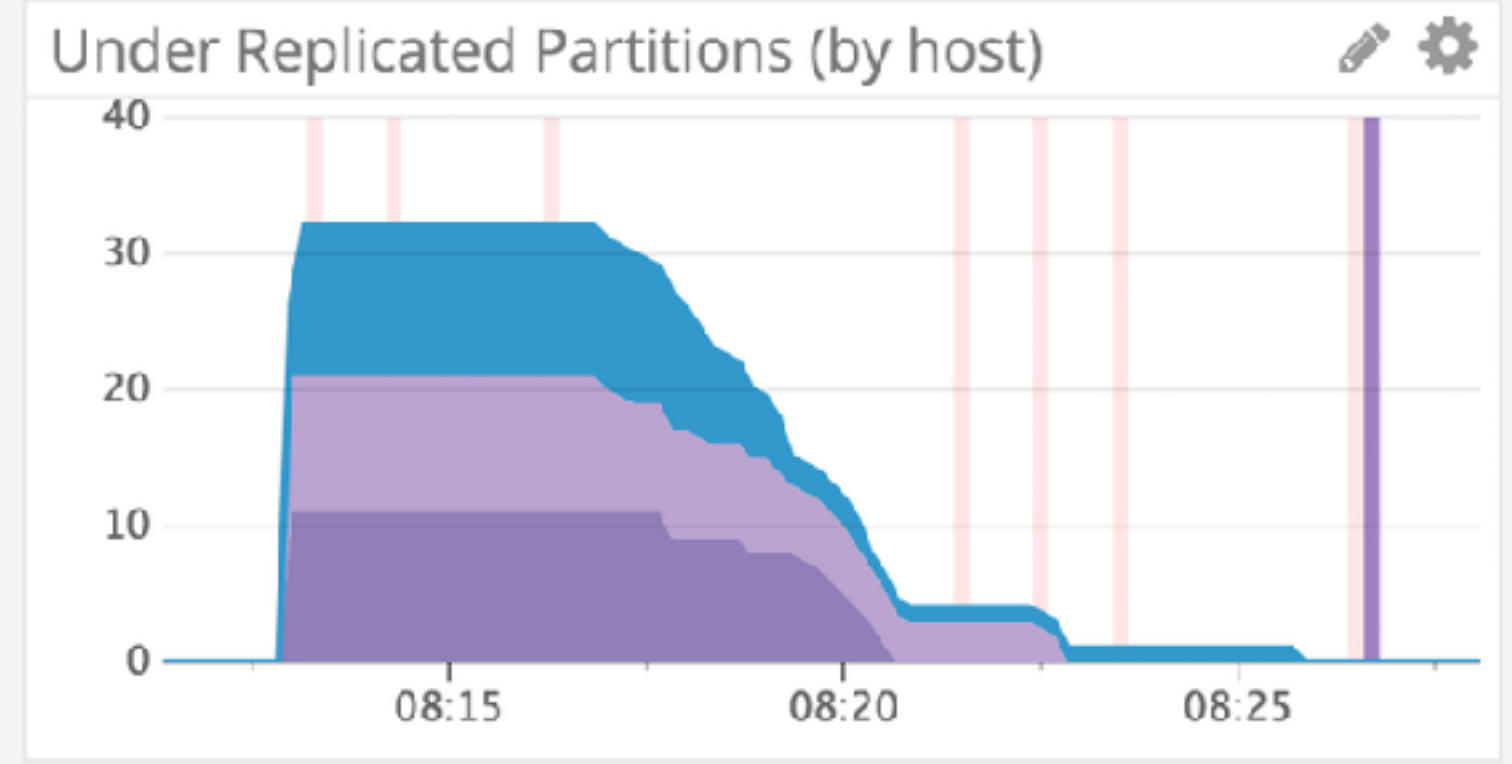
tags:name:kafka-autothrottle,cluster: [redacted]

Show 17m Feb 22, 8:11AM - Feb 22, 8:28AM



Events

- [kafka-autothrottle] Broker replication throttle removed**
Replication throttle removed on the following brokers: [1003 1004 1002 1007 1005 1006]
Thu Feb 22 2018 08:26:36 GMT-0700 (MST) · [Add comment](#) · [Lower priority](#)
- [kafka-autothrottle] Topics done reassigning**
Topics done reassigning: [redacted]
Thu Feb 22 2018 08:26:29 GMT-0700 (MST) · [Add comment](#) · [Lower priority](#)
- [kafka-autothrottle] Broker replication throttle set**
Replication throttle of 225.00MB/s set on the following brokers: [1004 1005 1007]
Topics currently undergoing replication: [redacted]
Thu Feb 22 2018 08:23:28 GMT-0700 (MST) · [Add comment](#) · [Lower priority](#)
- [kafka-autothrottle] Broker replication throttle set**
Replication throttle of 174.11MB/s set on the following brokers: [1003 1004 1005 1007]
Topics currently undergoing replication: [redacted]
Thu Feb 22 2018 08:22:26 GMT-0700 (MST) · [Add comment](#) · [Lower priority](#)
- [kafka-autothrottle] Broker replication throttle set**



Kafka-Kit: scaling operations

- <https://github.com/datadog/kafka-kit>
- `topicmapper`:
 - partition to broker mapping
 - failed broker replacement
 - storage-based cluster rebalancing
- `autothrottle`: replication auto-throttling
- pluggable metrics backend

Topic mapping

“Map”: assignment of a set of topics to Kafka brokers

```
map1: "events.*" => [1001,1002,1003,1004,1005,1006]
```

```
map2: "check_runs|notifications" => [1007,1008,1009]
```

Topic mapping

“Map”: assignment of a set of topics to Kafka brokers

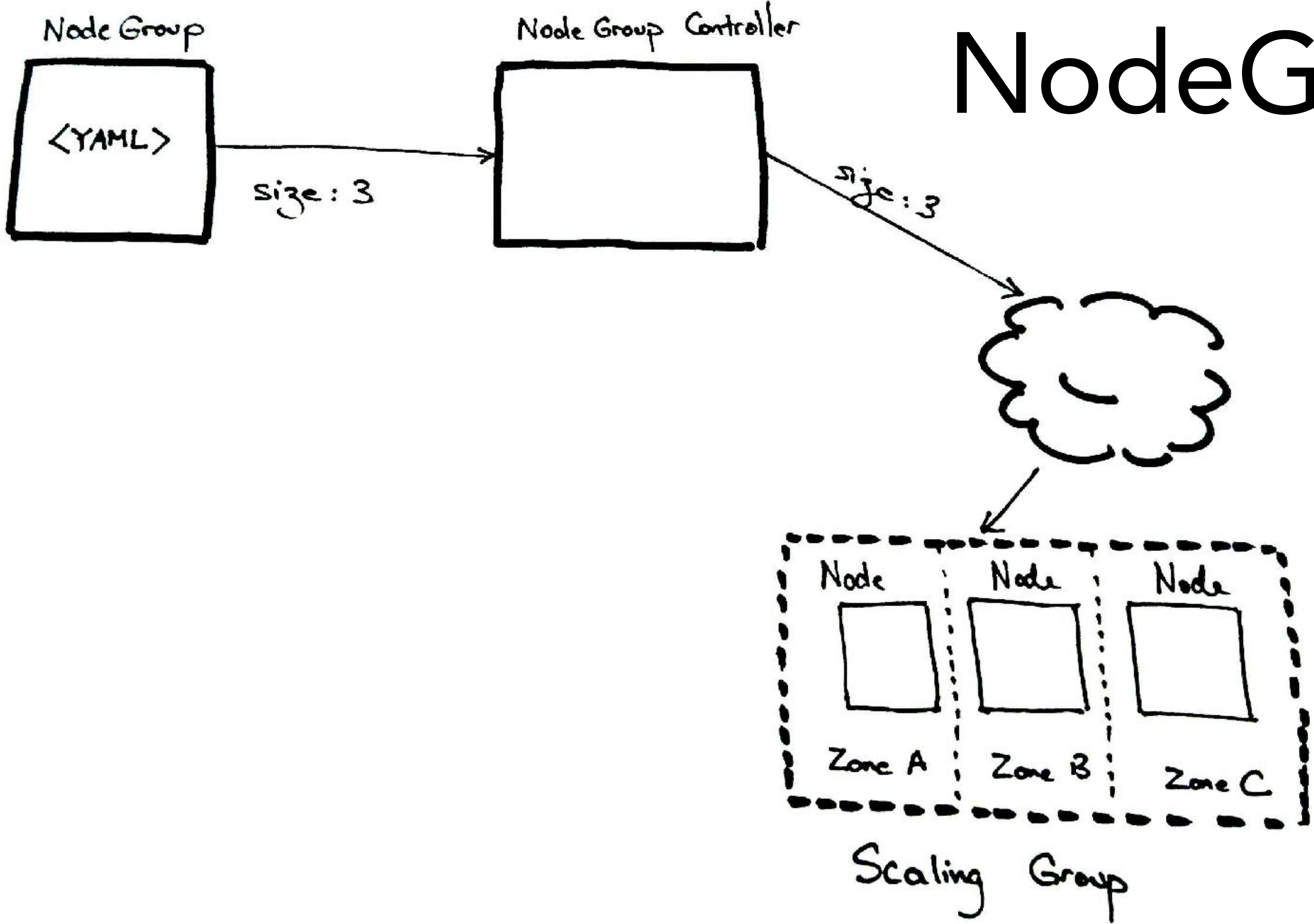
map1: "events.*" => 6x i3.4xlarge

map2: "check_runs|notifications" => 3x i3.8xlarge

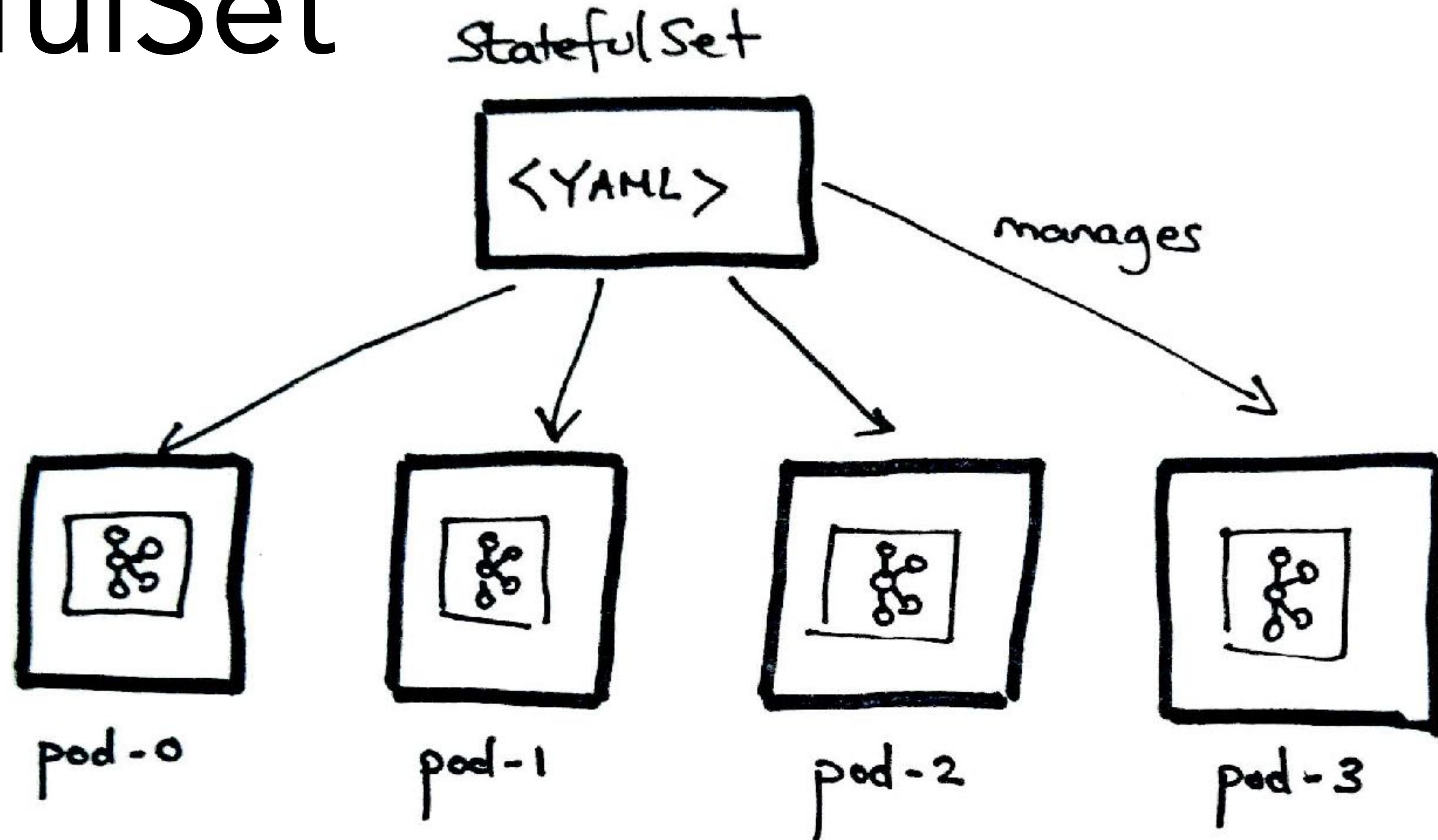


k8s concepts

NodeGroup

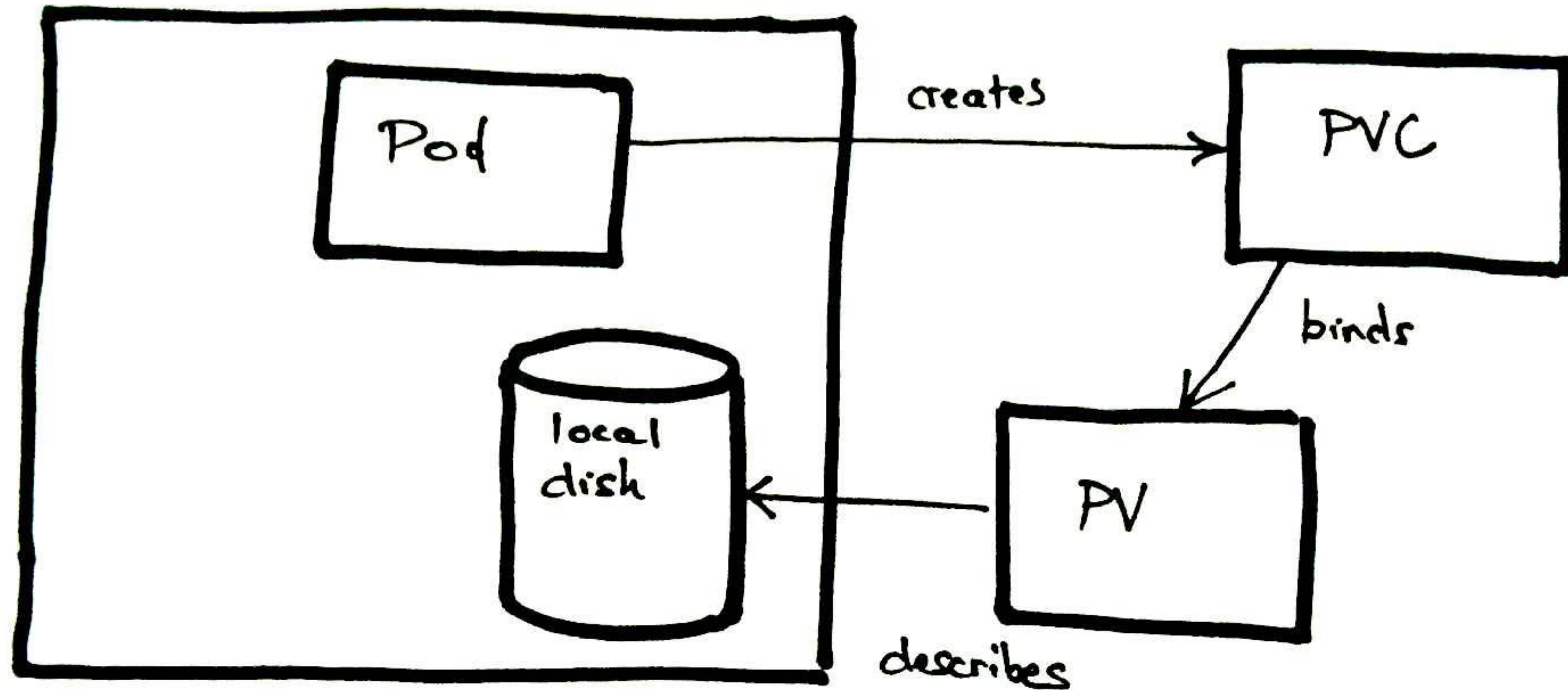


StatefulSet



Persistent Volume (Claim)

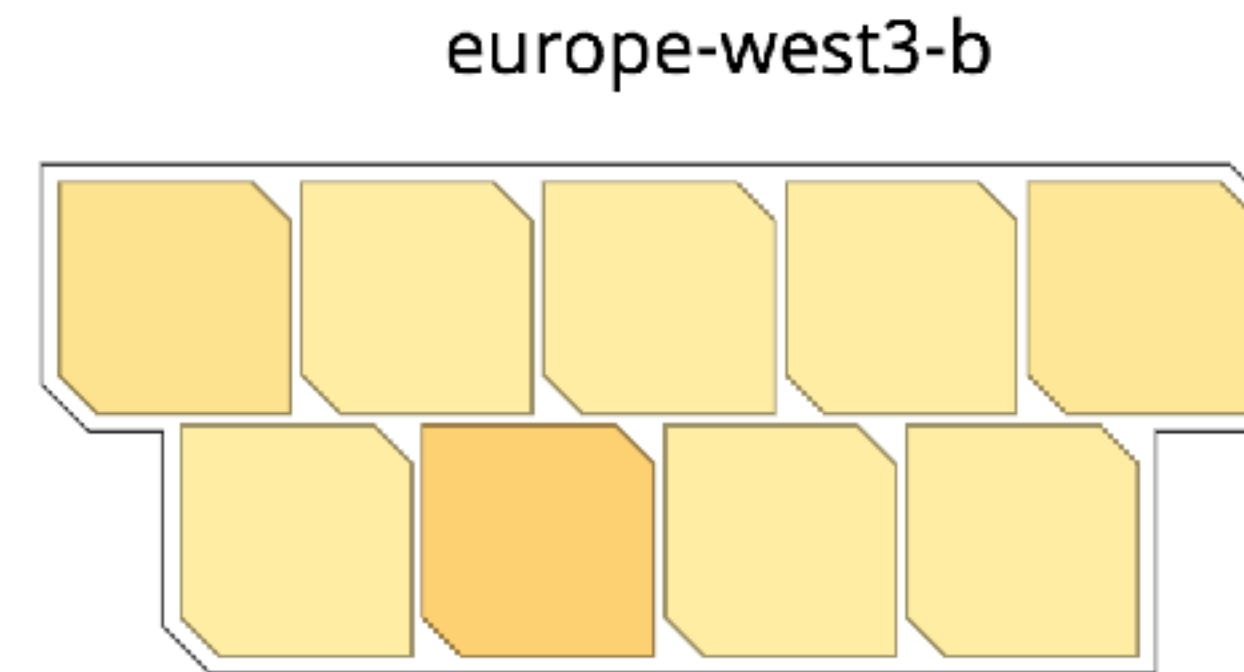
Node



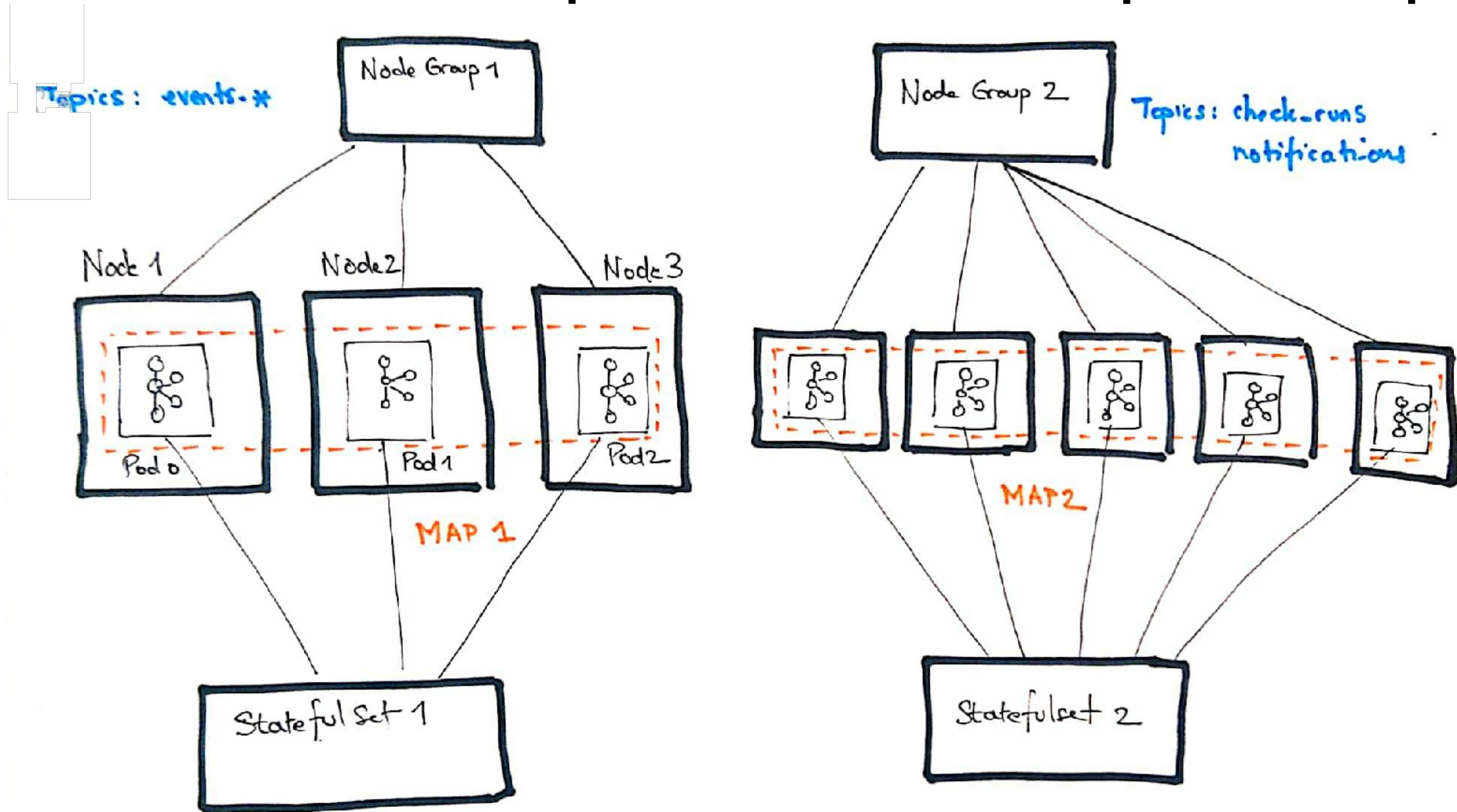
Cluster deployment in k8s

Broker deployment

One broker pod per node (`nodeAffinity` & `podAntiAffinity`)



One NodeGroup/StatefulSet per map



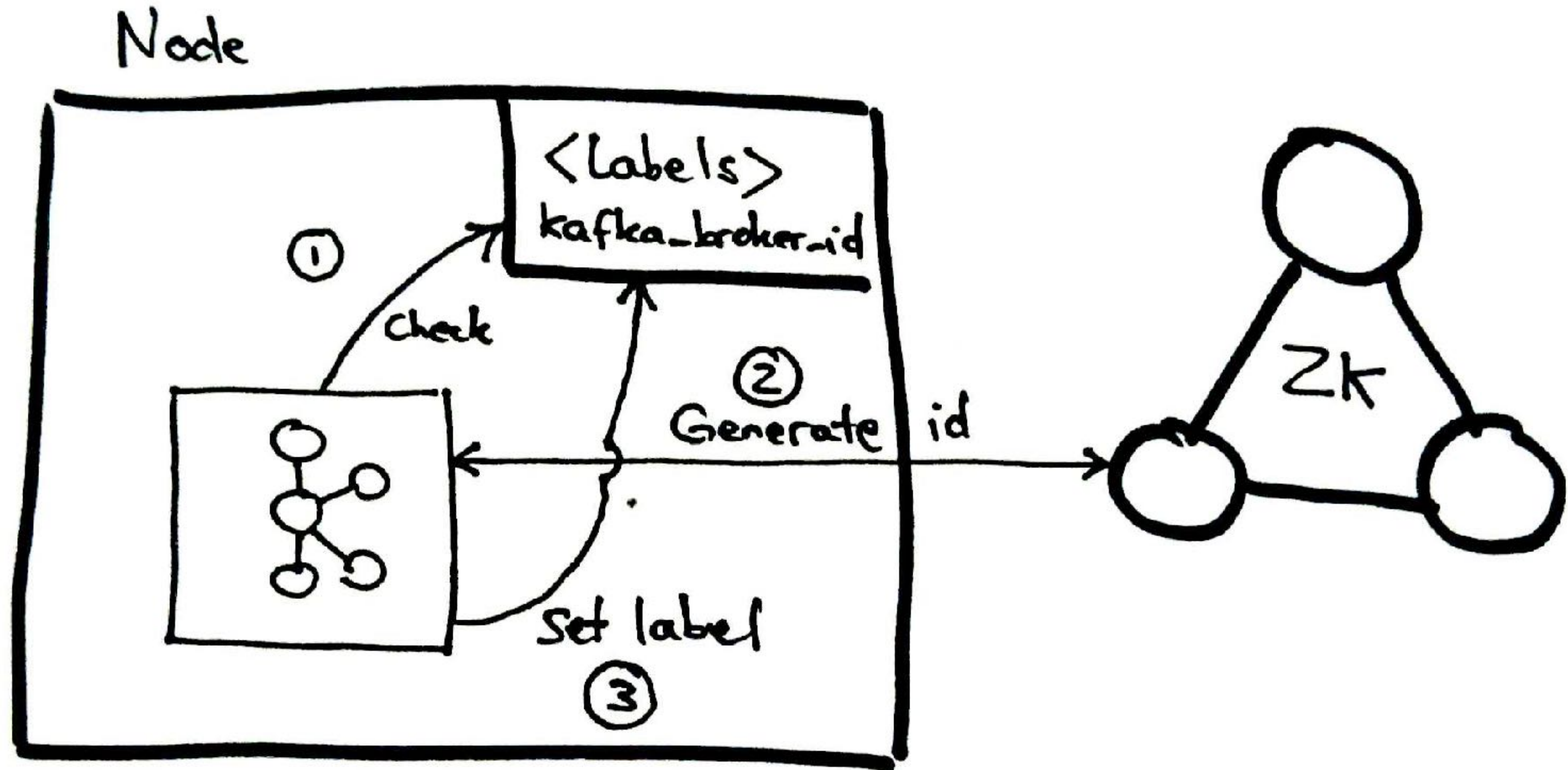
Data persistence and locality

- Instance store drives
- Data is persisted between pod restarts
- Data replicated on new nodes
- Rack-awareness: query zone of current node via k8s API at init



```
function set_broker_rack {  
    NODE_ZONE=$(kubectl get node $node_name \  
        -o=jsonpath='{.metadata.labels.failure-domain\.beta\.kubernetes\.io\/zone}')  
    if [[ "$NODE_ZONE" = "" ]]; then  
        echo "No failure domain could be read from the pod labels"  
        exit 1  
    else  
        echo "broker.rack=$NODE_ZONE" >> /etc/kafka/server.properties  
    fi  
}
```

Broker identity





```
function assign_kafka_broker_id {
    local broker_id=$(kubectl get node $NODE_NAME \
        -o=jsonpath='{.metadata.labels.kafka_broker_id}')

    if [[ "$broker_id" = "" ]]; then
        broker_id=$(/opt/kafka/bin/custom/assign-broker-id)
    fi

    echo "broker.id=$broker_id" >> /etc/kafka/server.properties
    kubectl label node --overwrite=true $NODE_NAME "kafka_broker_id=$broker_id"
}
```

Pod health and readiness

ZooKeeper:

- Liveness: port 2181 open?
- Readiness: leader/follower?

Kafka:

- Liveness: port 9092 open?
- Readiness: broker 100% in-sync?
- break the glass: forceable readiness (when 2 incidents coincide)

Safe rolling-restarts

Under-replicated and reassigned partitions / topic

Show 43 m Oct 25, 11:08AM - Oct 25, 11:52AM



This can go if we need to save time

Topic management

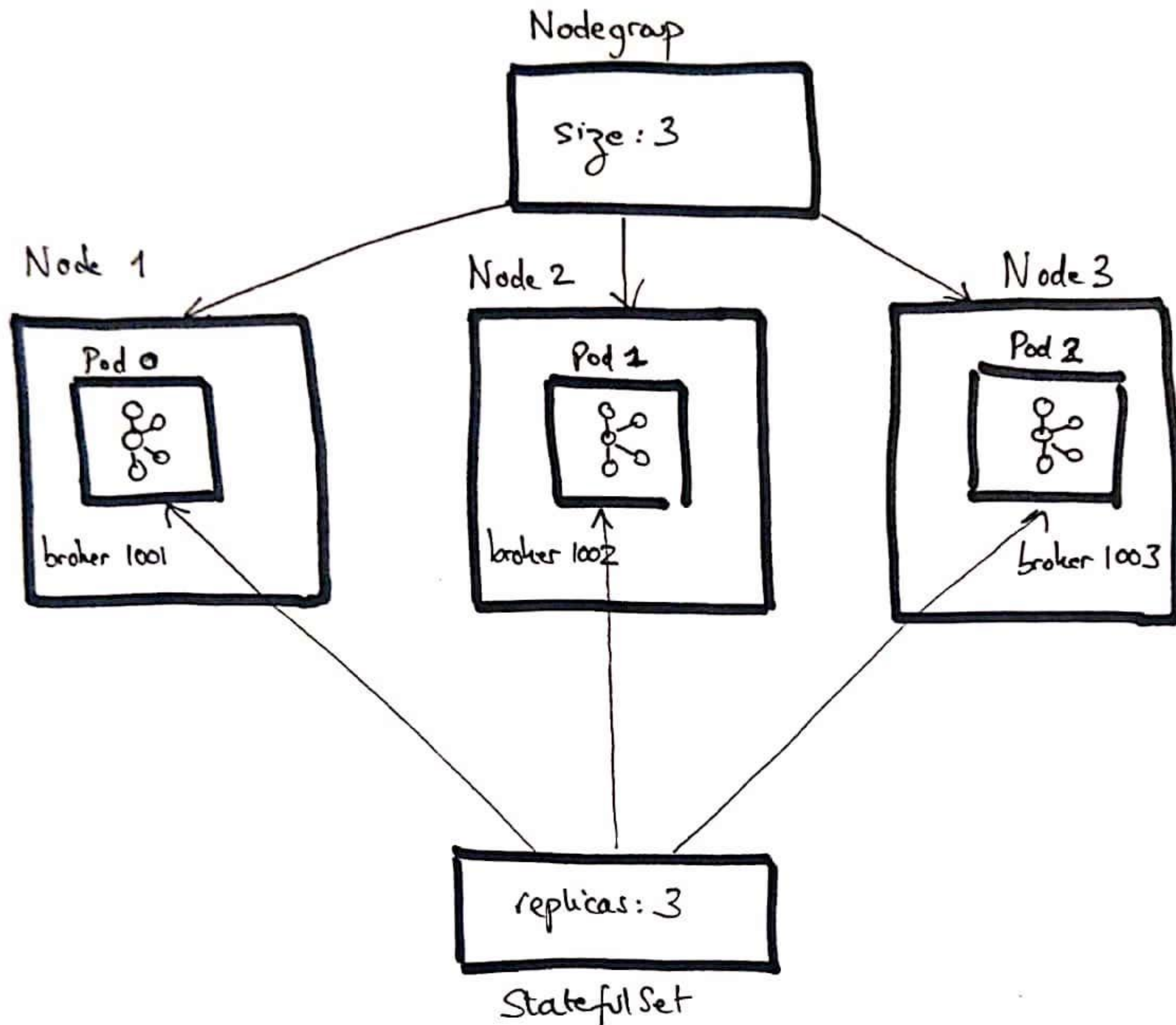
- Topic definition in a `ConfigMap`
- Regularly applied via a `CronJob`
- Broker ids/map resolved by looking up the k8s API

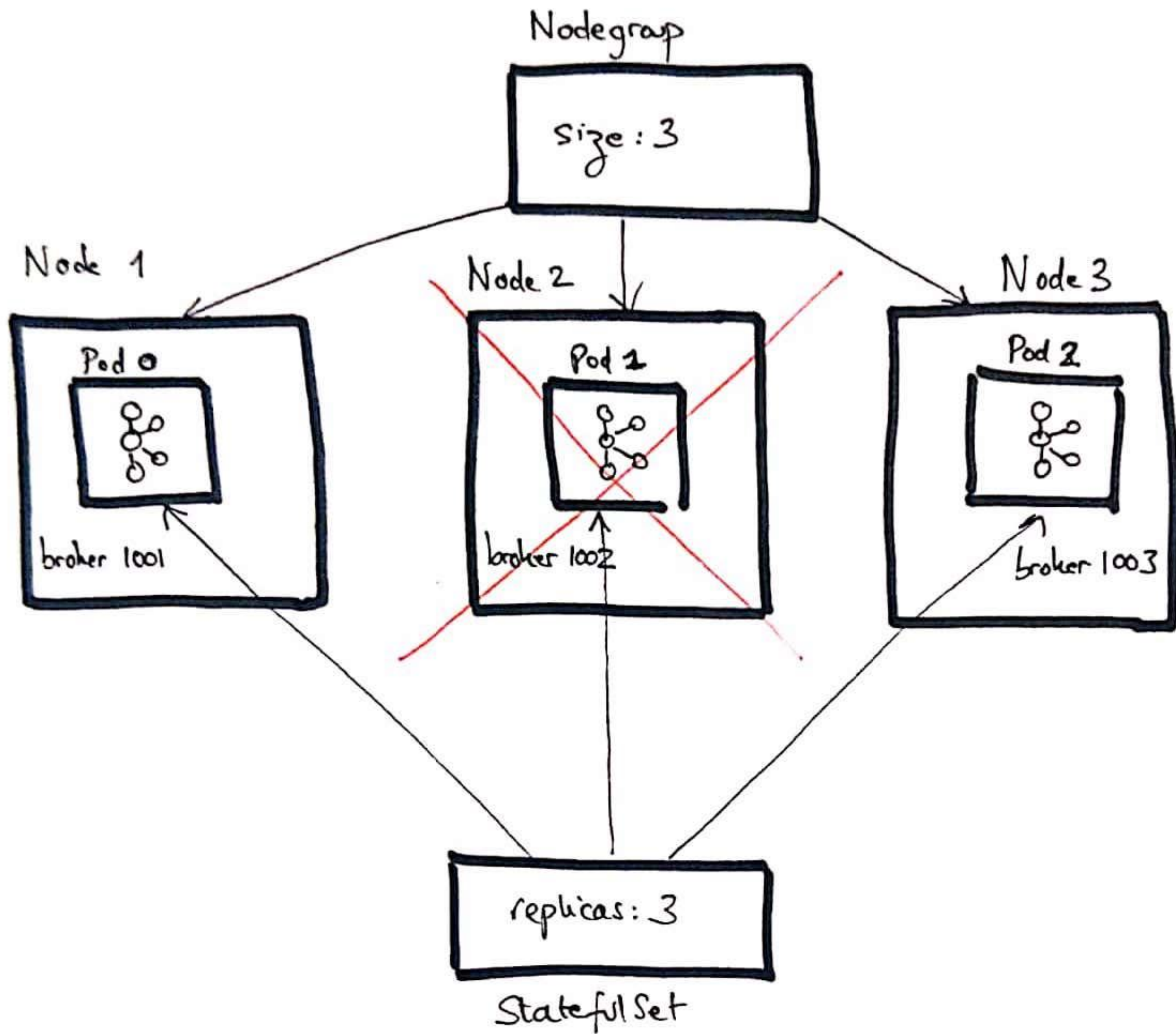
```
topics:  
  topic1:  
    partitions: 1  
    replication: 3  
    retention: 72  
    config:  
      cleanup.policy: compact
```

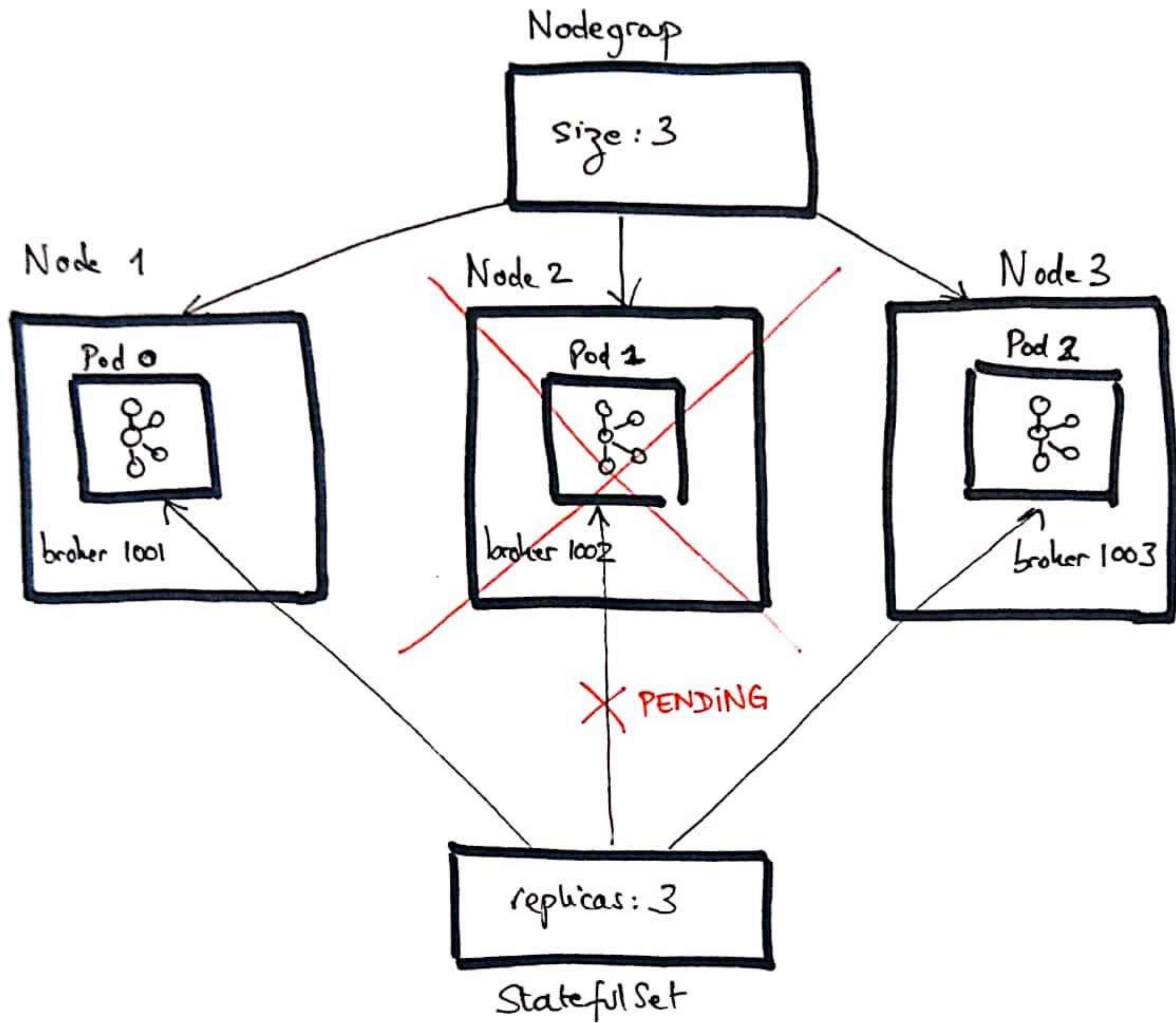


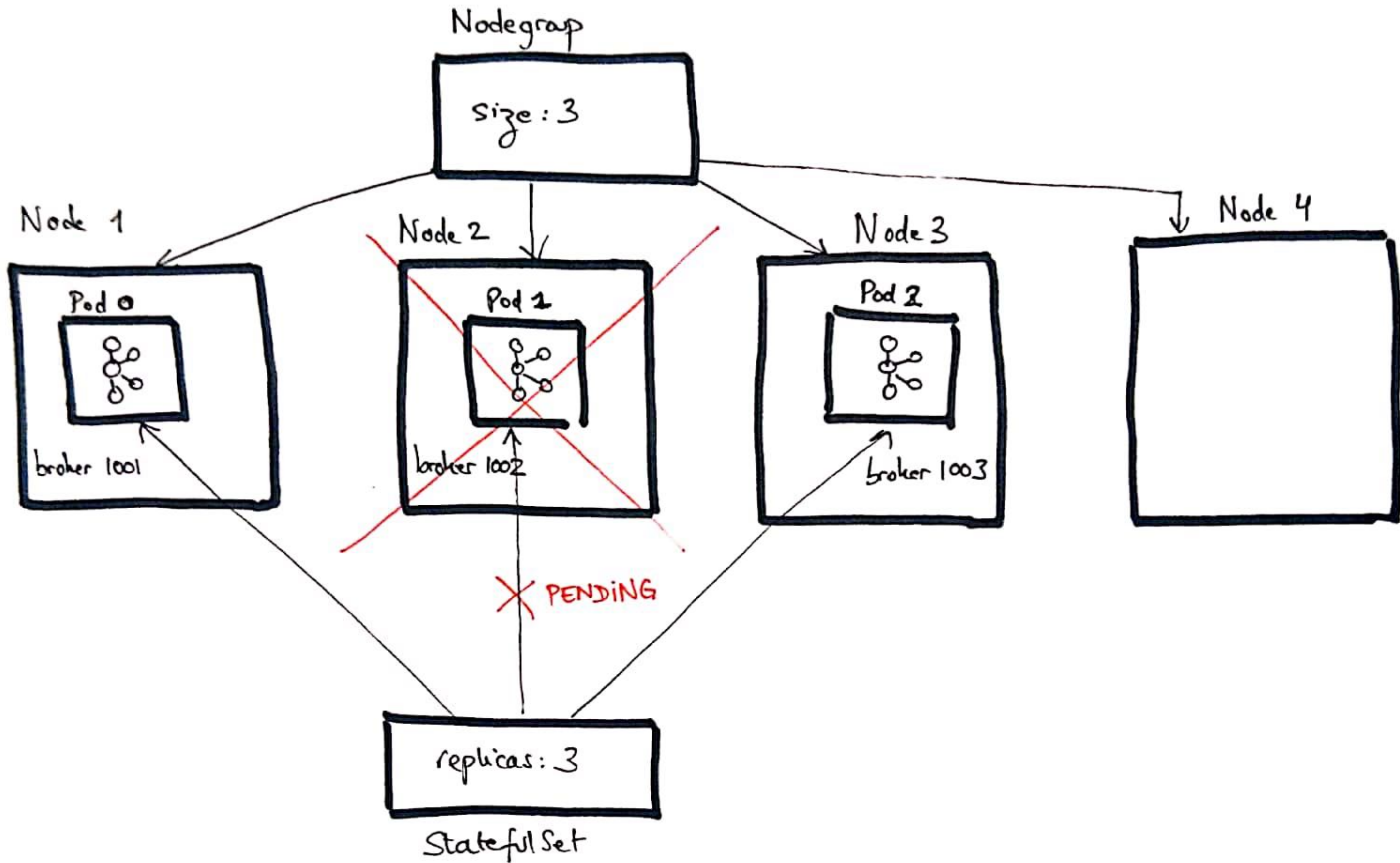
Operations

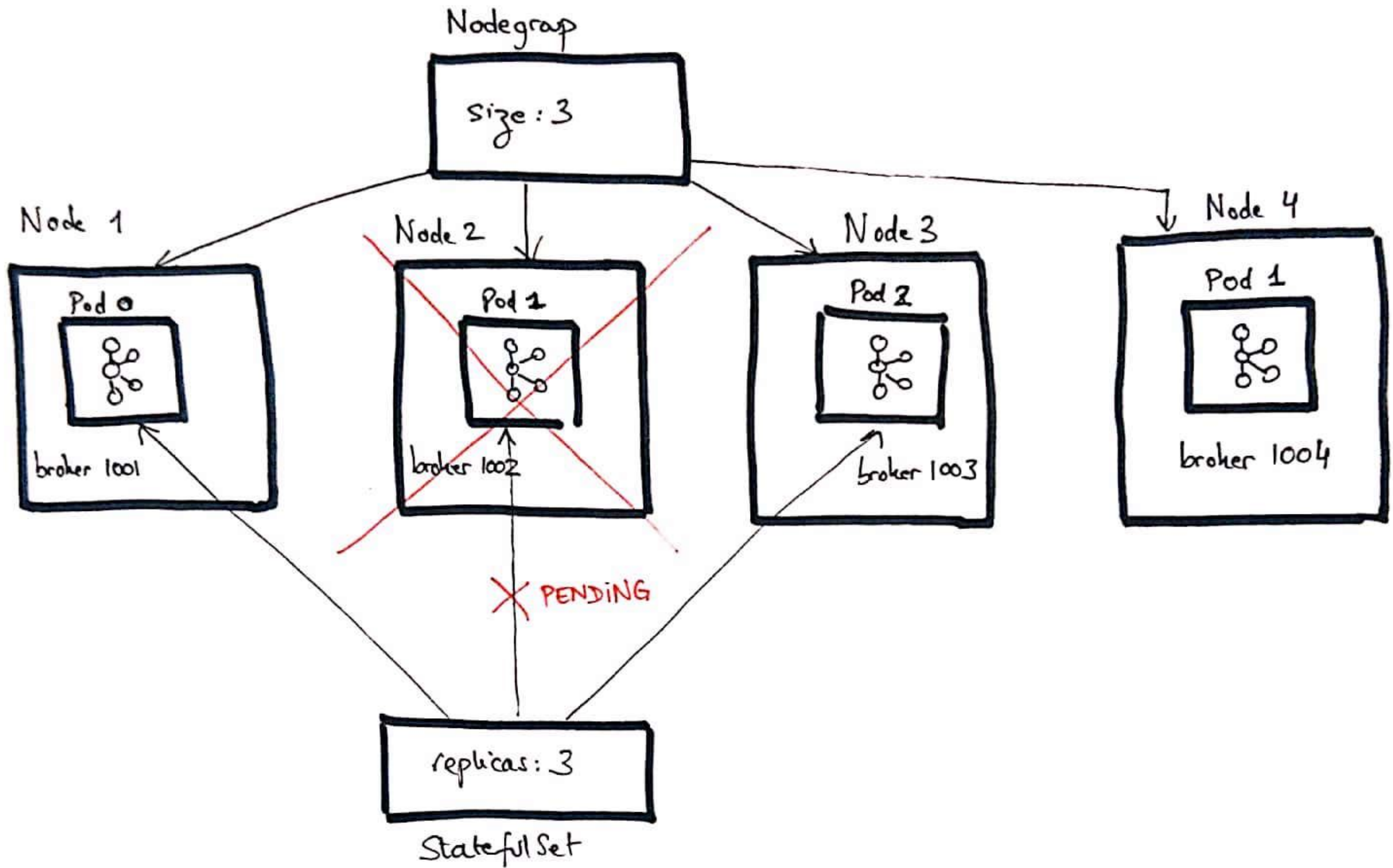
Broker replacement

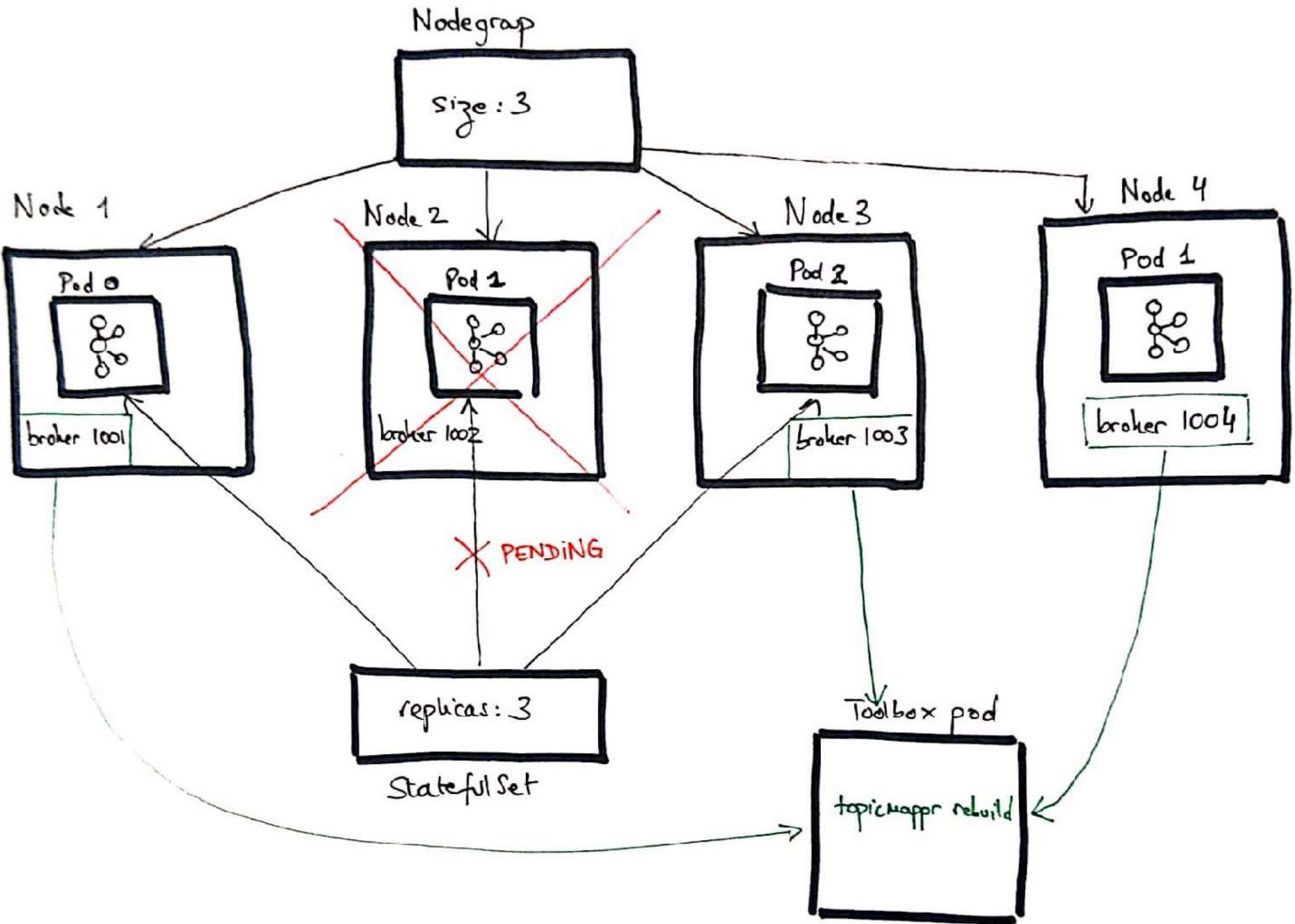


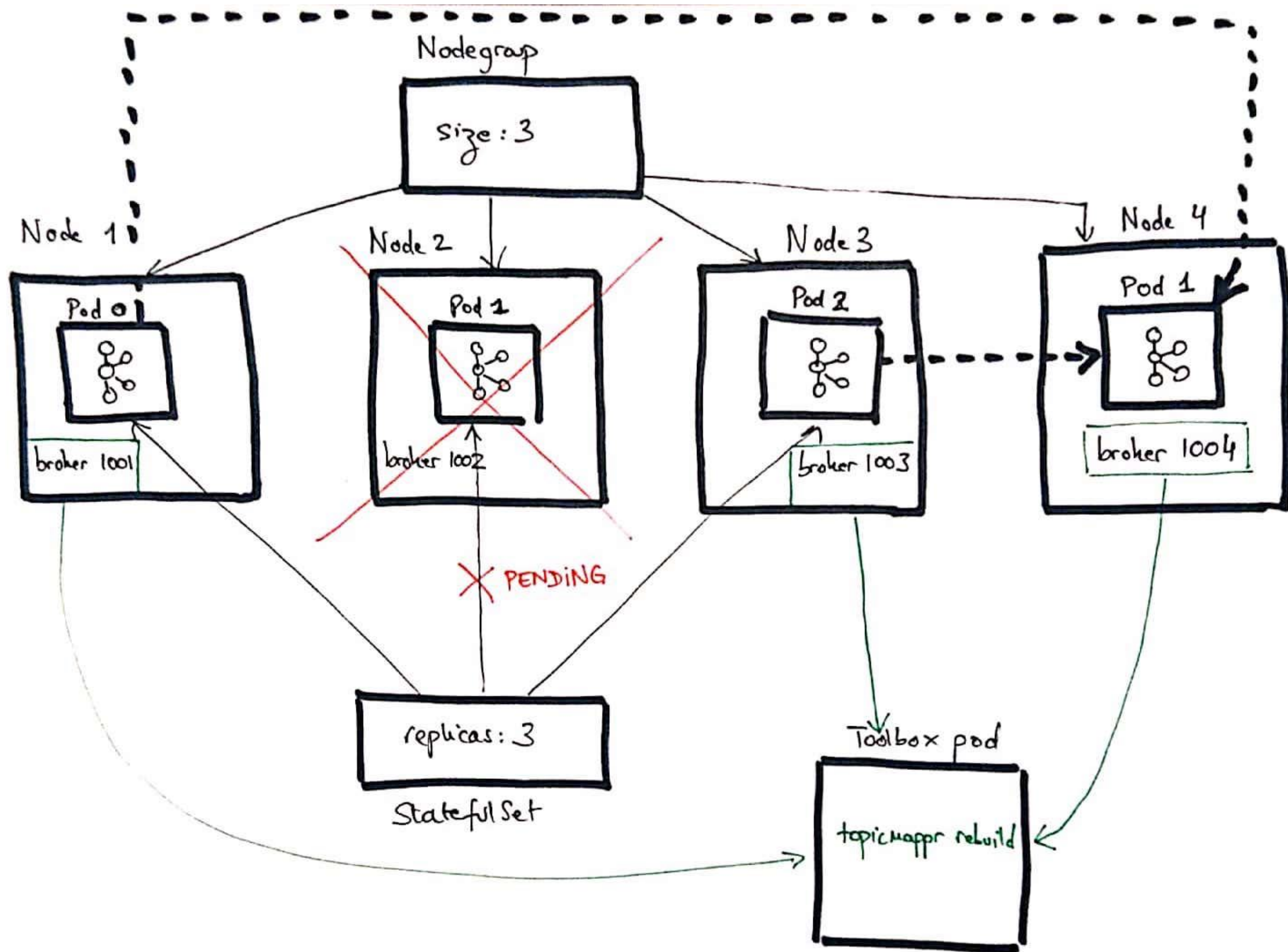




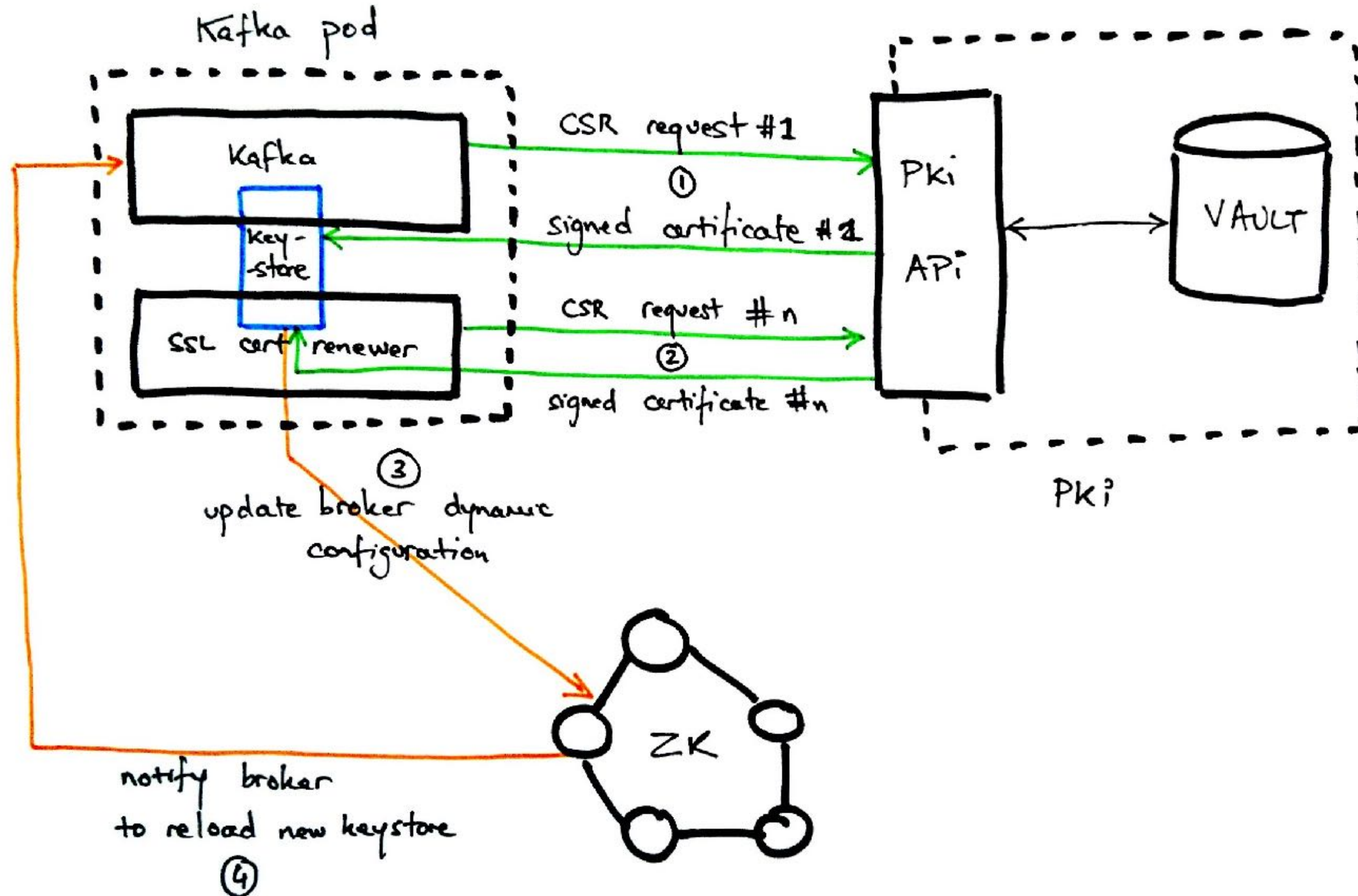






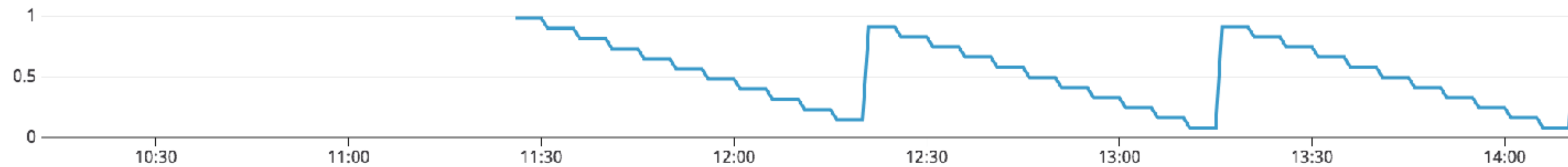


SSL certificate management



SSL certificate management

Timeseries Editor



1 Select your visualization

Timeseries Query Value Heat Map Scatter Plot Distribution Top List Change Host Map

2 Graph your data

[Graph Primer](#)

[Share](#)

[JSON](#)

[Edit](#)

Metric

vaultd.certificate_expiration...

from

\$datacenter x \$k8s_cluster x \$cluster x \$map x \$broker x kube_container_name:ssl-certificate-renewer x

Conclusion

- We twist the Kubernetes model to ensure dedicated resources
- We take advantage of Kubernetes' APIs to simplify configuration and operations
- The `kafka-kit` tooling works well in Kubernetes
- We gradually automate operations where possible

Thank you!

@brouberol

We're hiring!

<https://www.datadoghq.com/careers>

