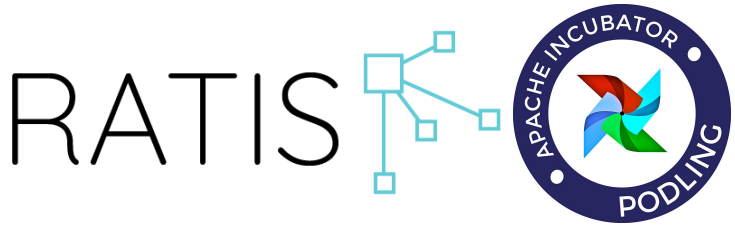
Hadoop
Ozone

Kubernetes

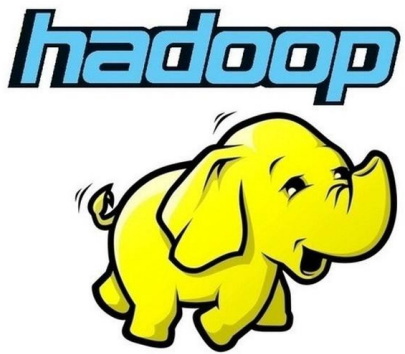
RATIS


RATIS
APACHE INCUBATOR
PODLING

RAFT library for Java

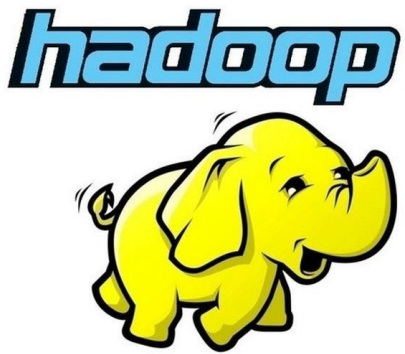
RATIS

RAFT library for Java

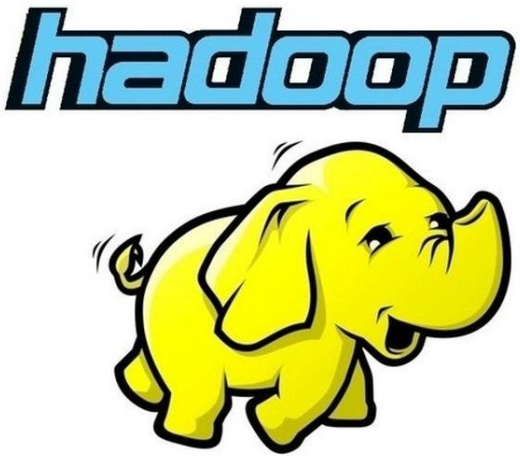


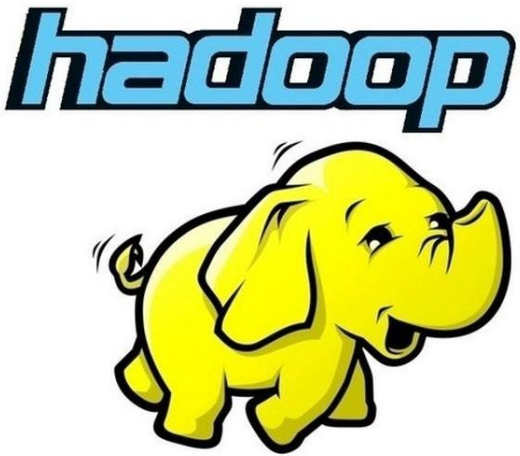
RATIS

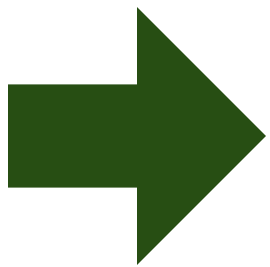
RAFT library for Java

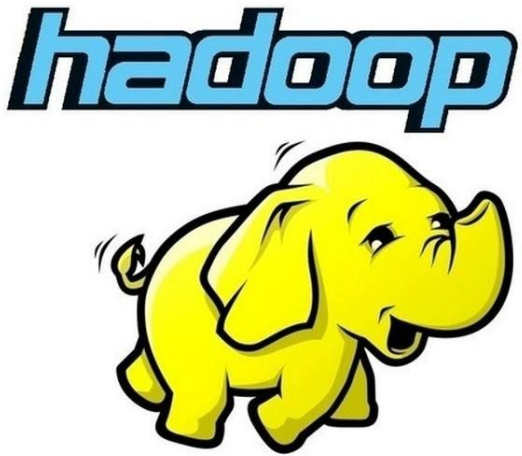


https://flokkr.github.io


hadoop

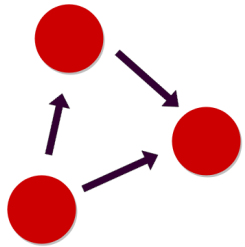

hadoop



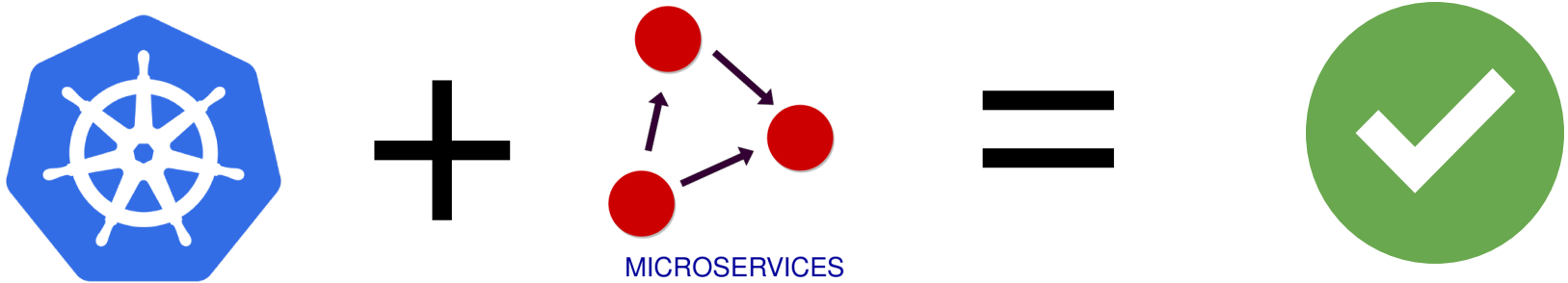

hadoop

Why

+



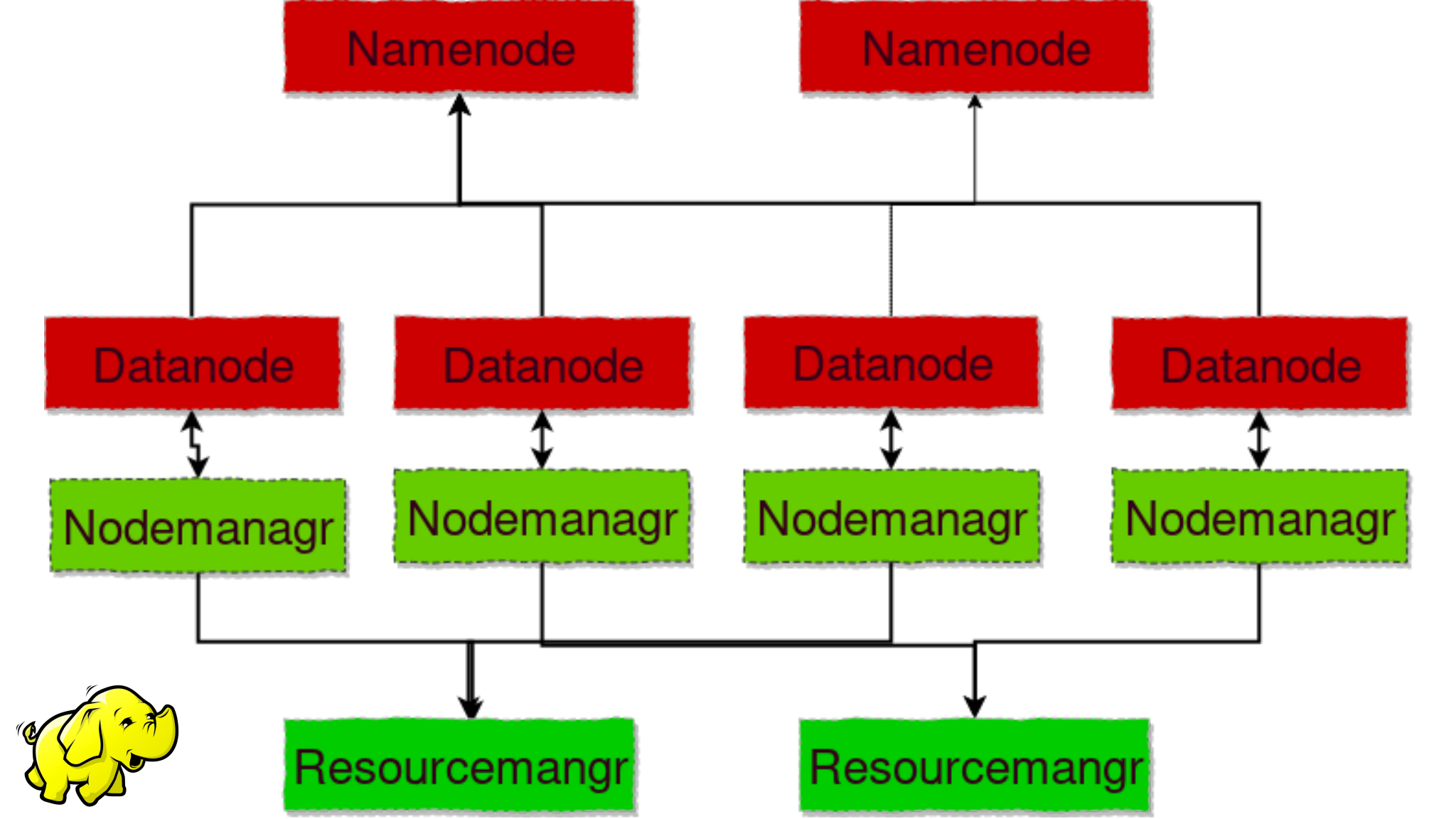
MICROSERVICES

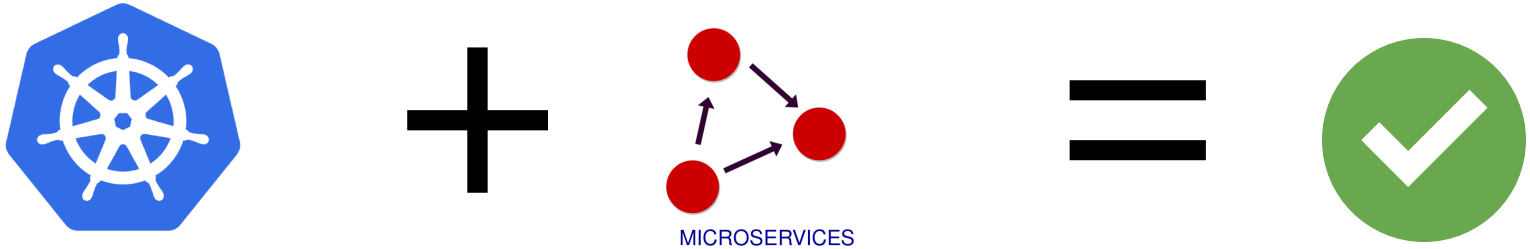

MICROSERVICES

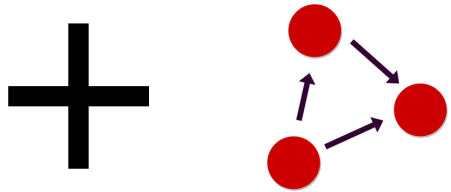
Microservices

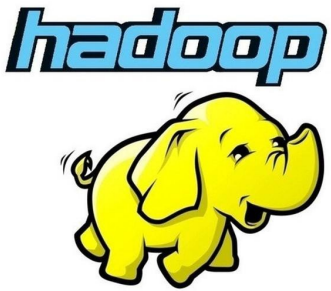
Microservices

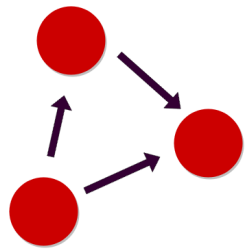
"microservice architectural style is an approach to developing a single application as a **suite of small services**, each running in its **own process** and **communicating** with **lightweight** mechanisms..." (Martin Fowler)


Namenode
Namenode
Datanode
Datanode
Datanode
Datanode
Nodemanagr
Nodemanagr
Nodemanagr
Nodemanagr
Resourcemangr
Resourcemangr

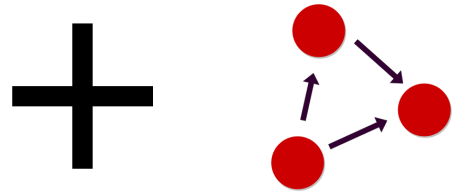

MICROSERVICES

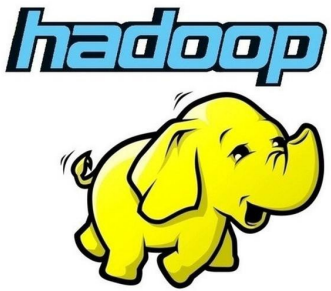

MICROSERVICES

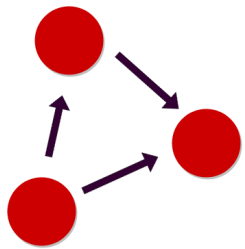

hadoop



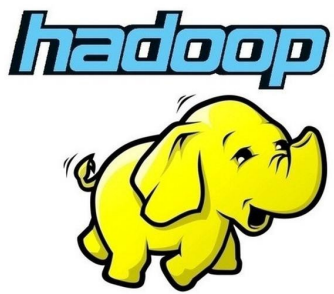
MICROSERVICES

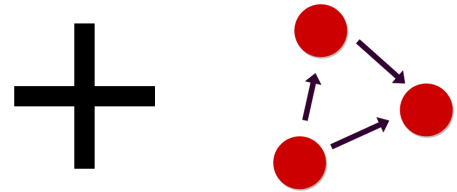


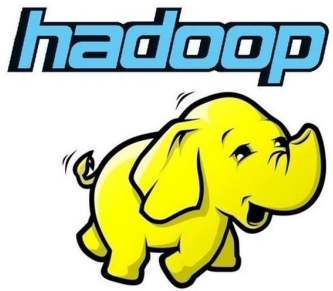


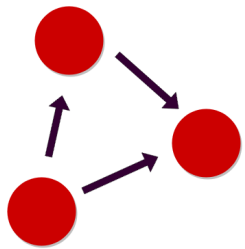


MICROSERVICES

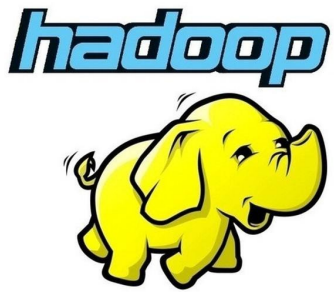


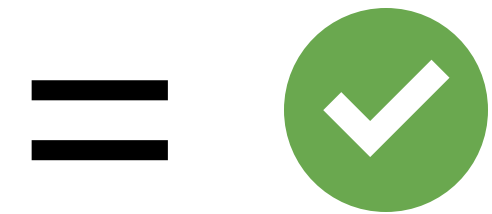


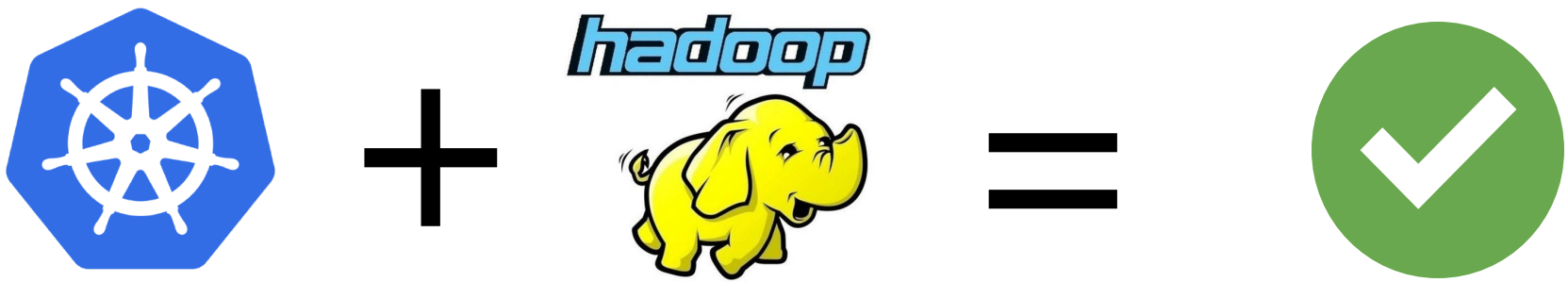




MICROSERVICES





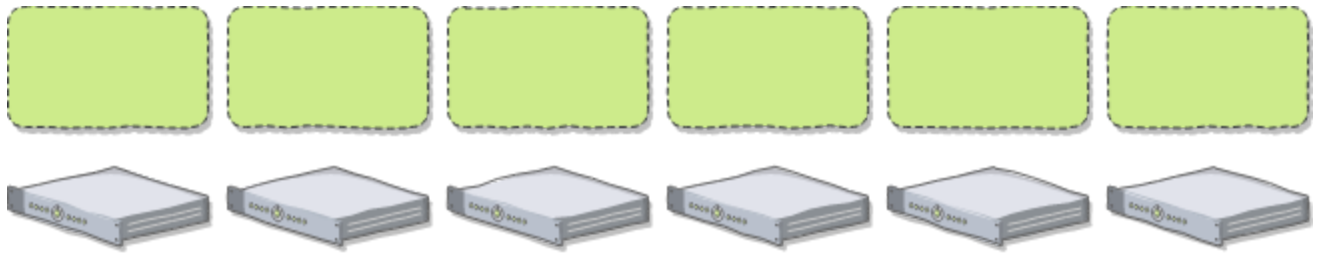

hadoop

What is Apache Hadoop

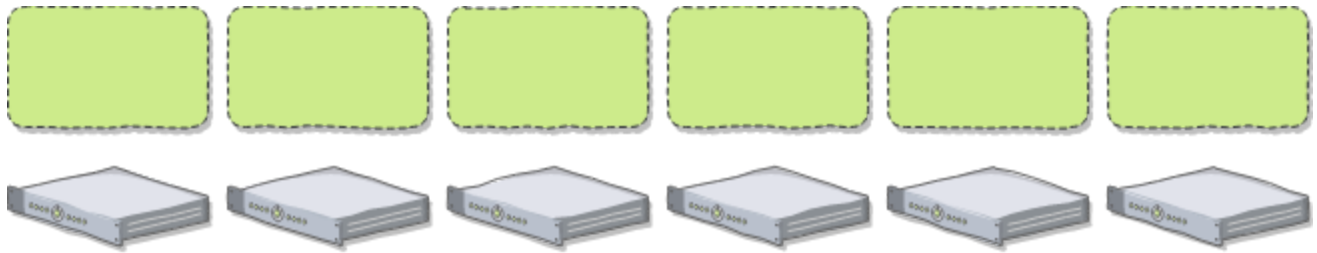
What is Apache Hadoop

in 60 seconds

Data

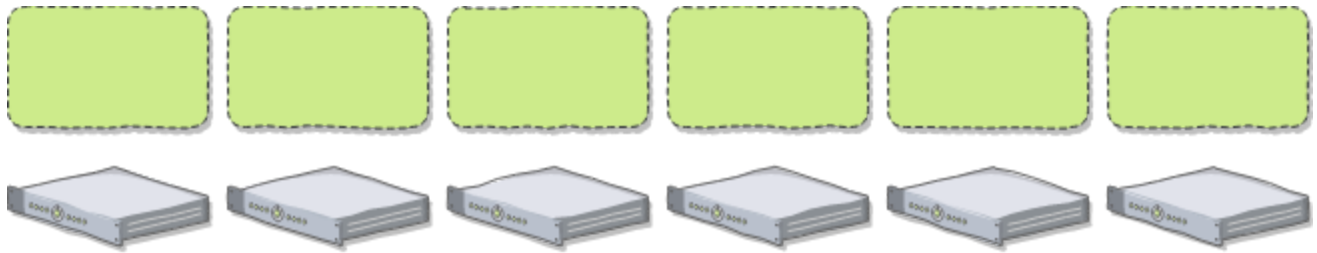


HDFS



HDFS

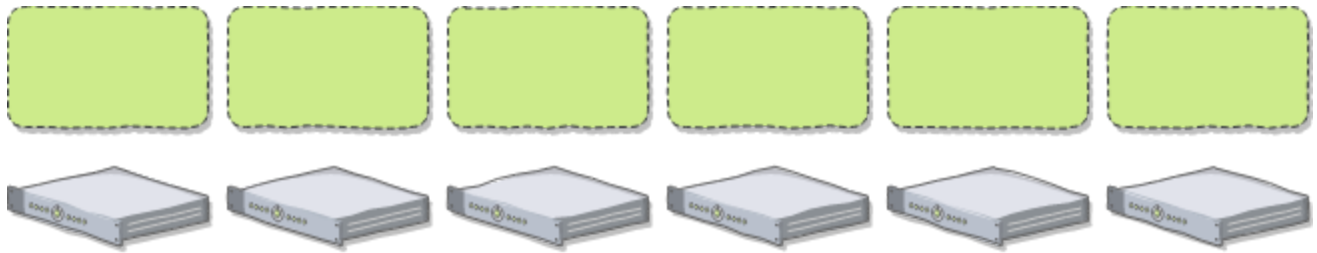
YARN



HDFS

YARN

Mapreduce

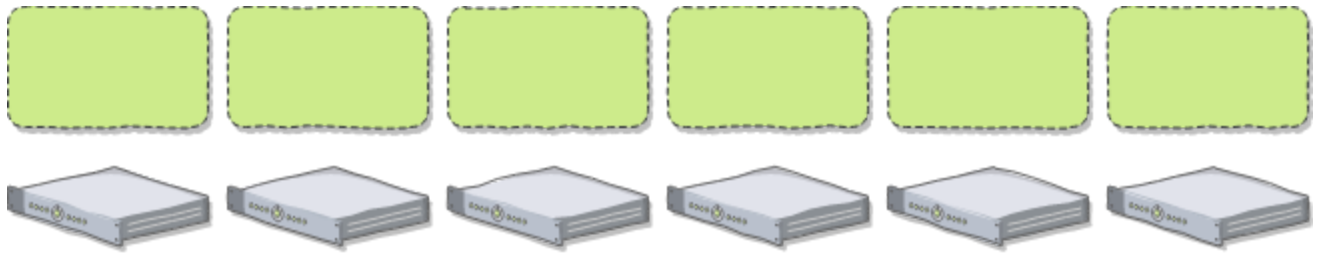


HDFS

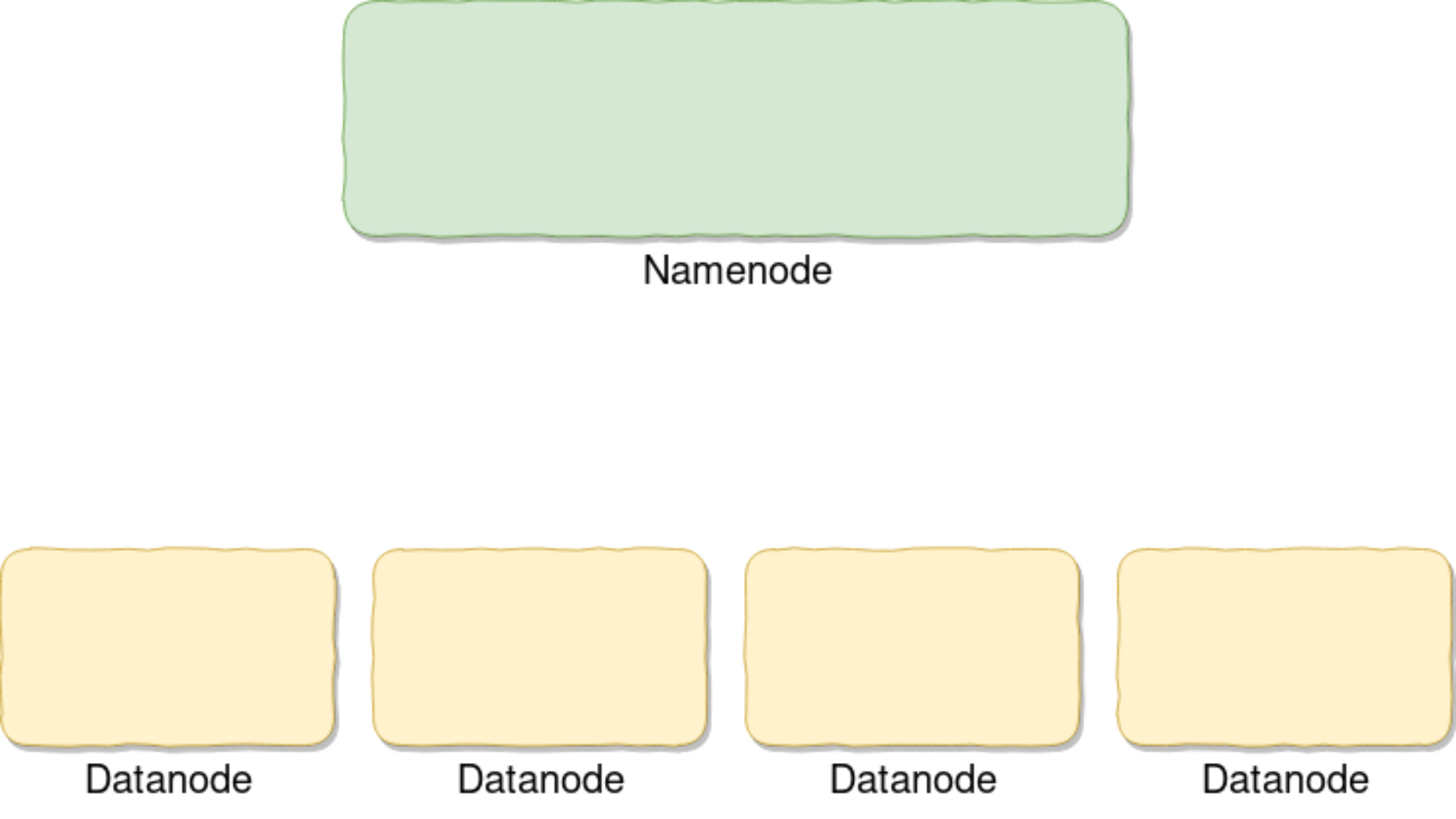
YARN

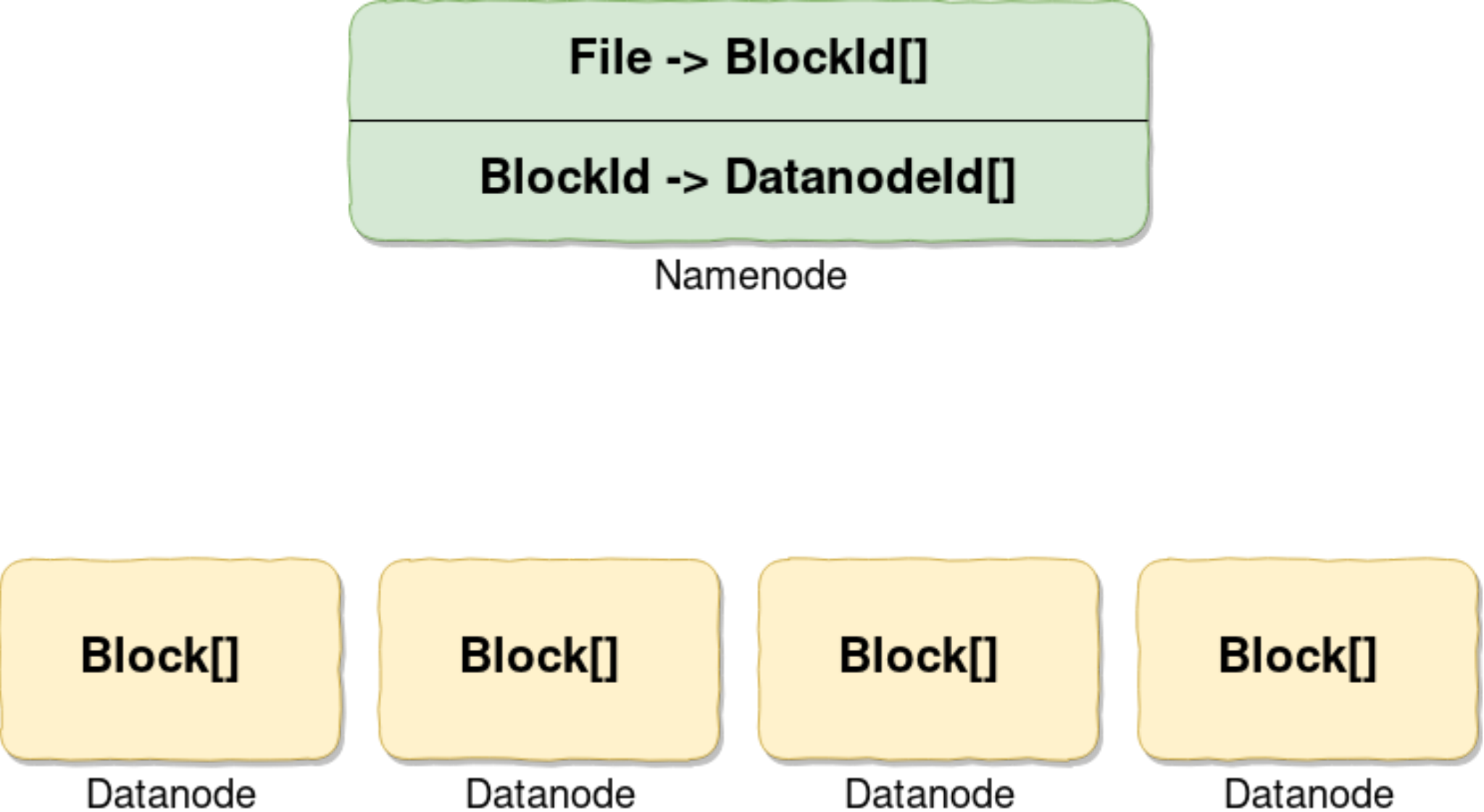
Mapreduce

Ozone + HDDS



Apache Hadoop HDDS+Ozone


Namenode
Datanode
Datanode
Datanode
Datanode


File -> BlockId[]
BlockId -> DatanodeId[]
Namenode
Block[]
Block[]
Block[]
Block[]
Datanode
Datanode
Datanode
Datanode

File -> BlockId[]

BlockId -> DatanodeId[]

Namenode

Block[]

Datanode

Block[]

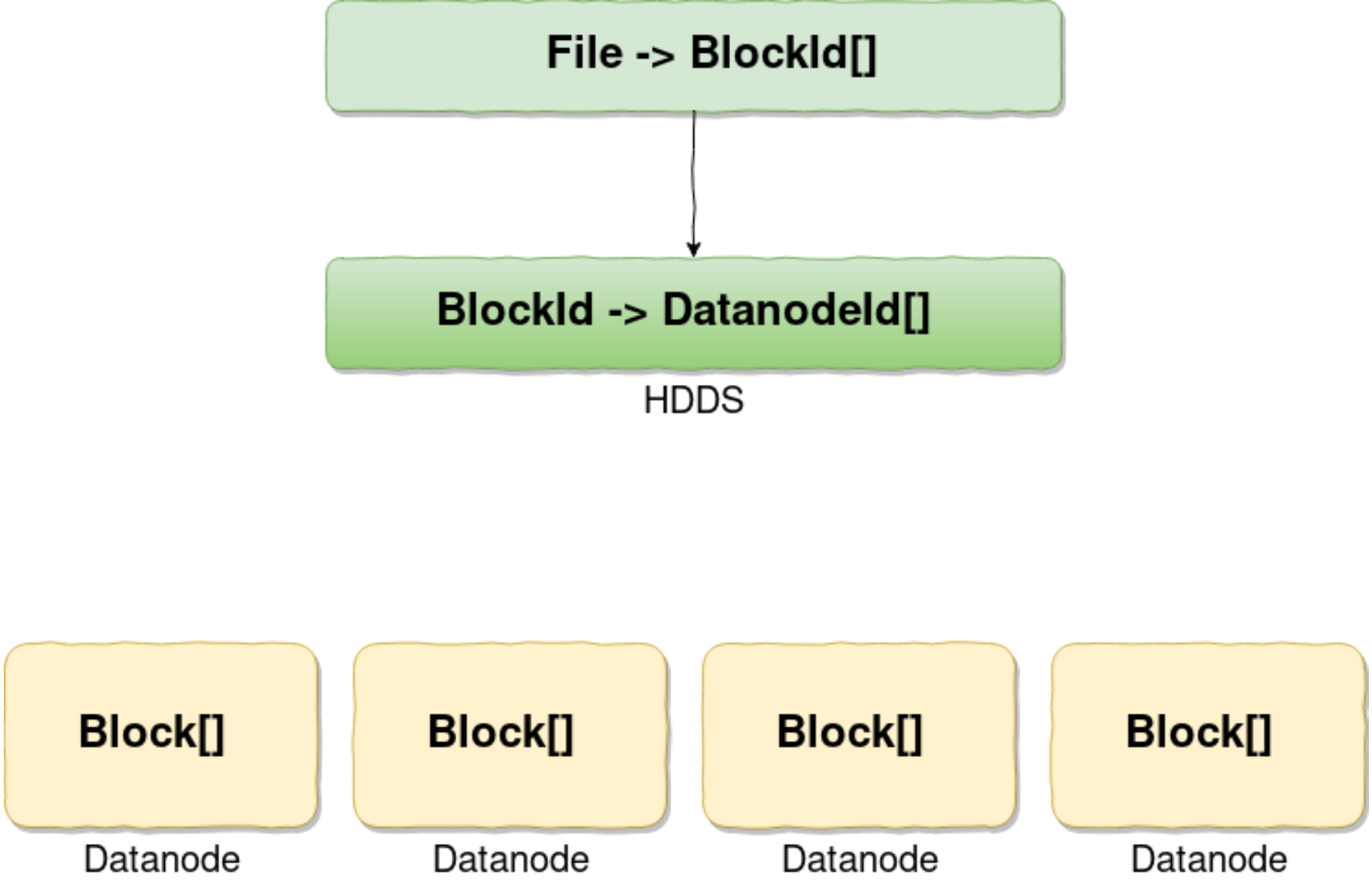
Datanode

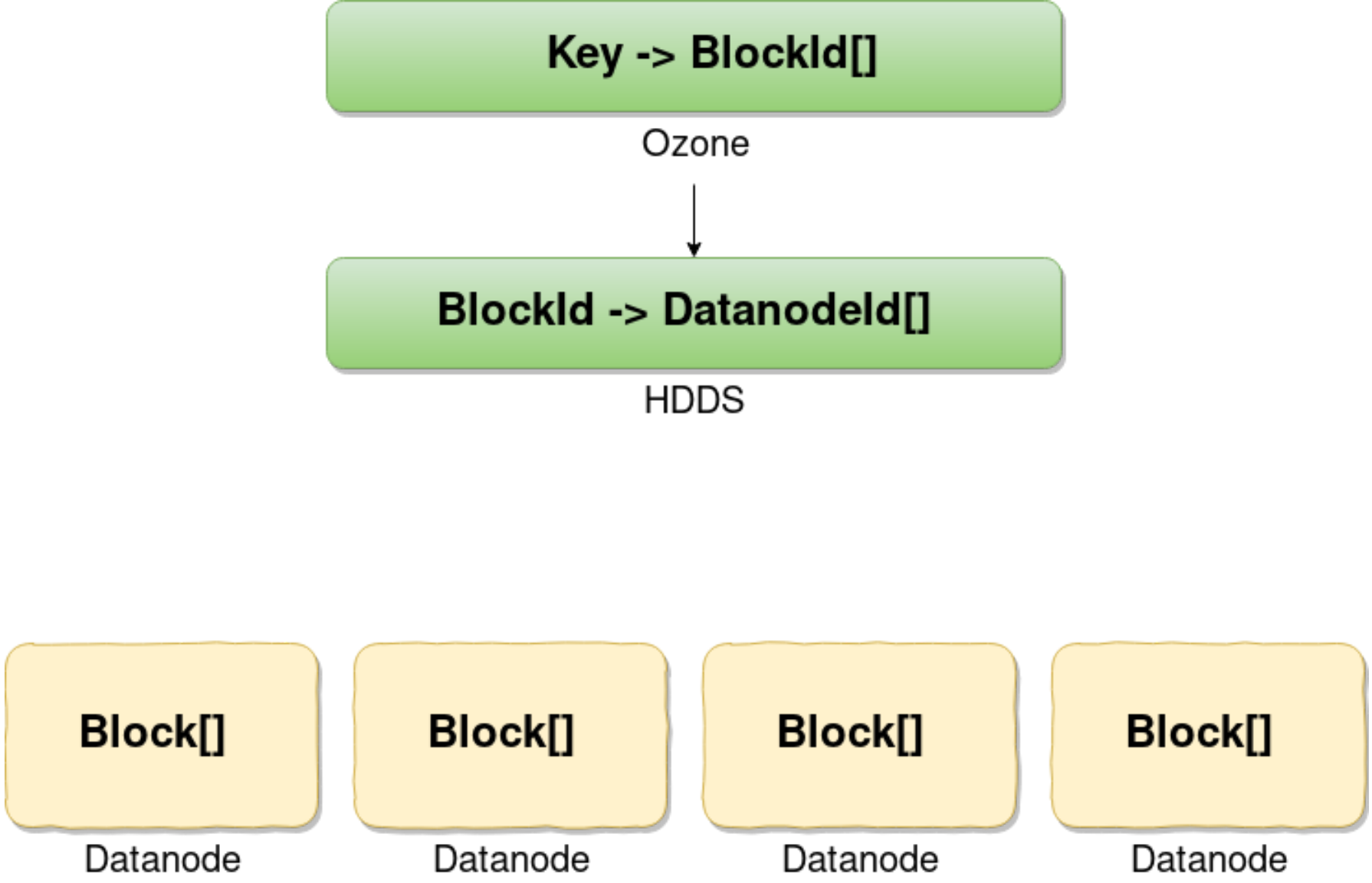
Block[]

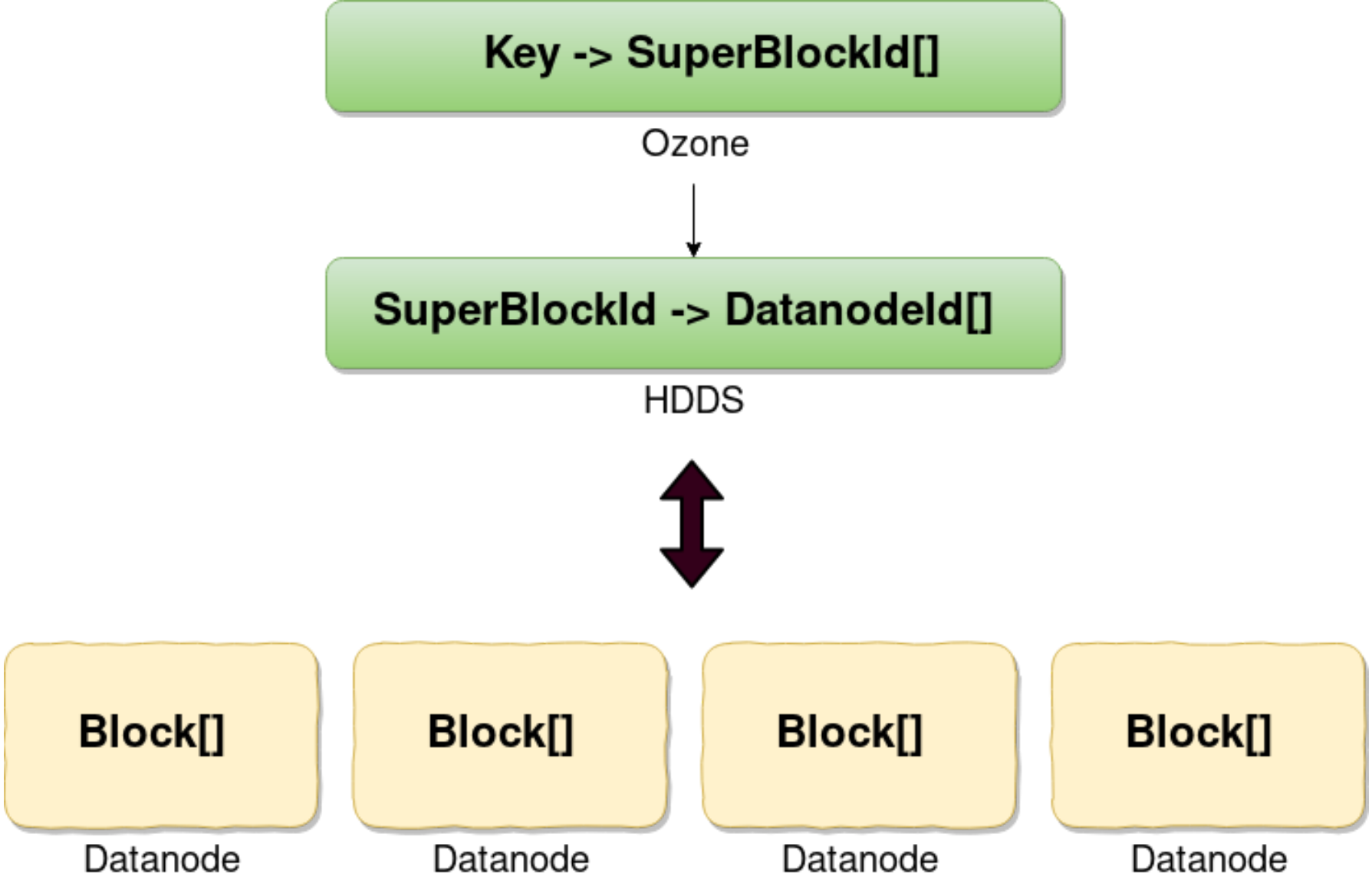
Datanode

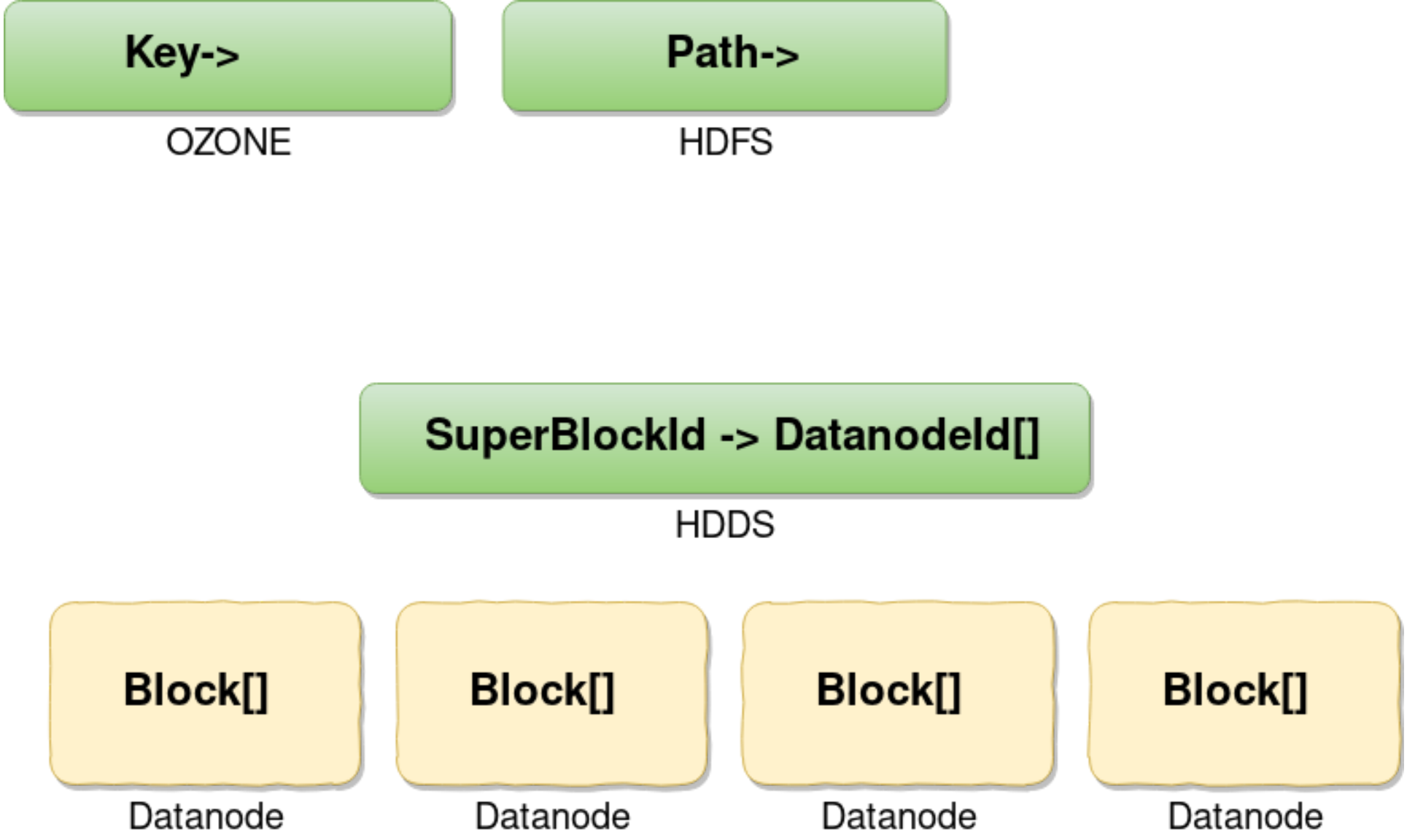
Block[]

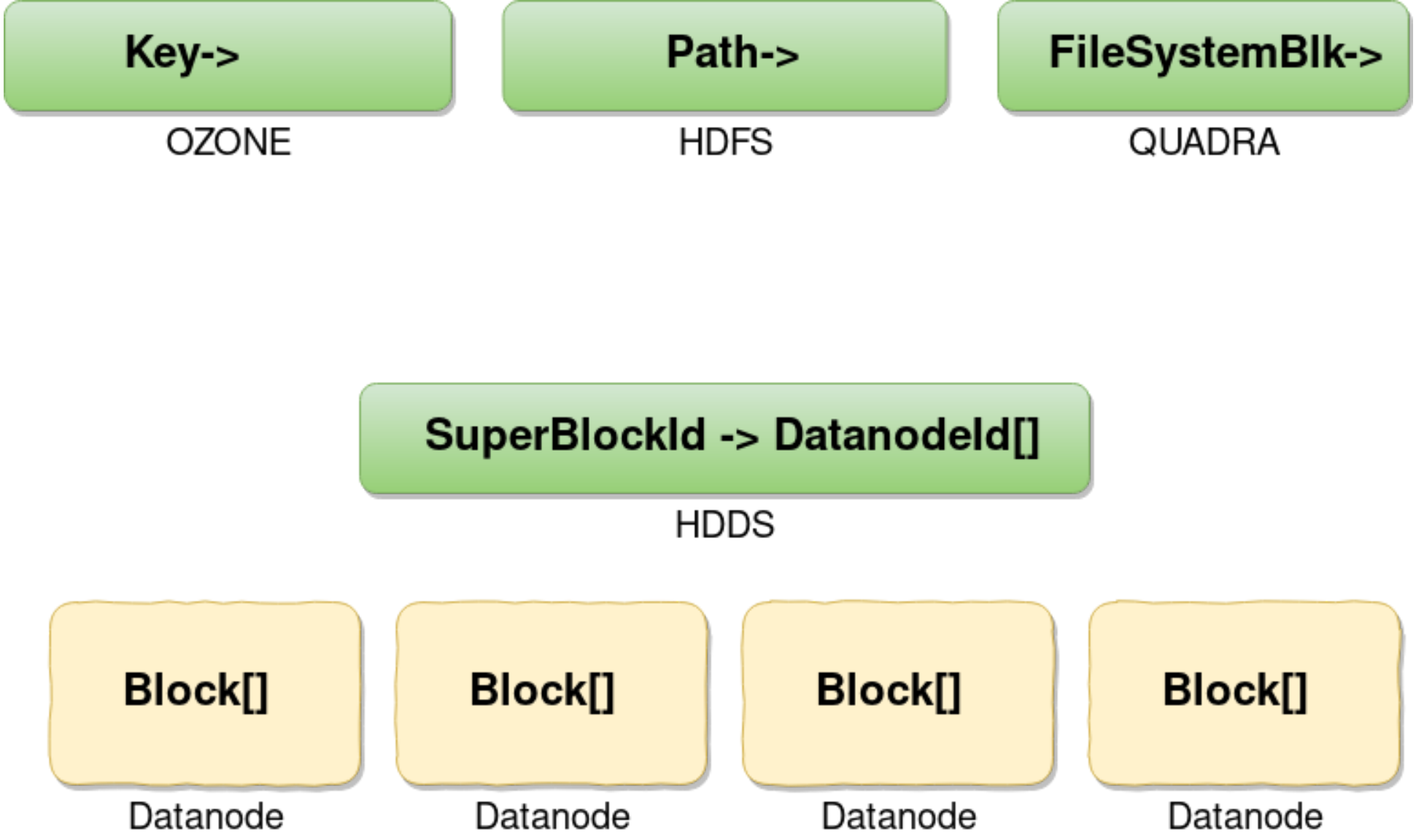
Datanode

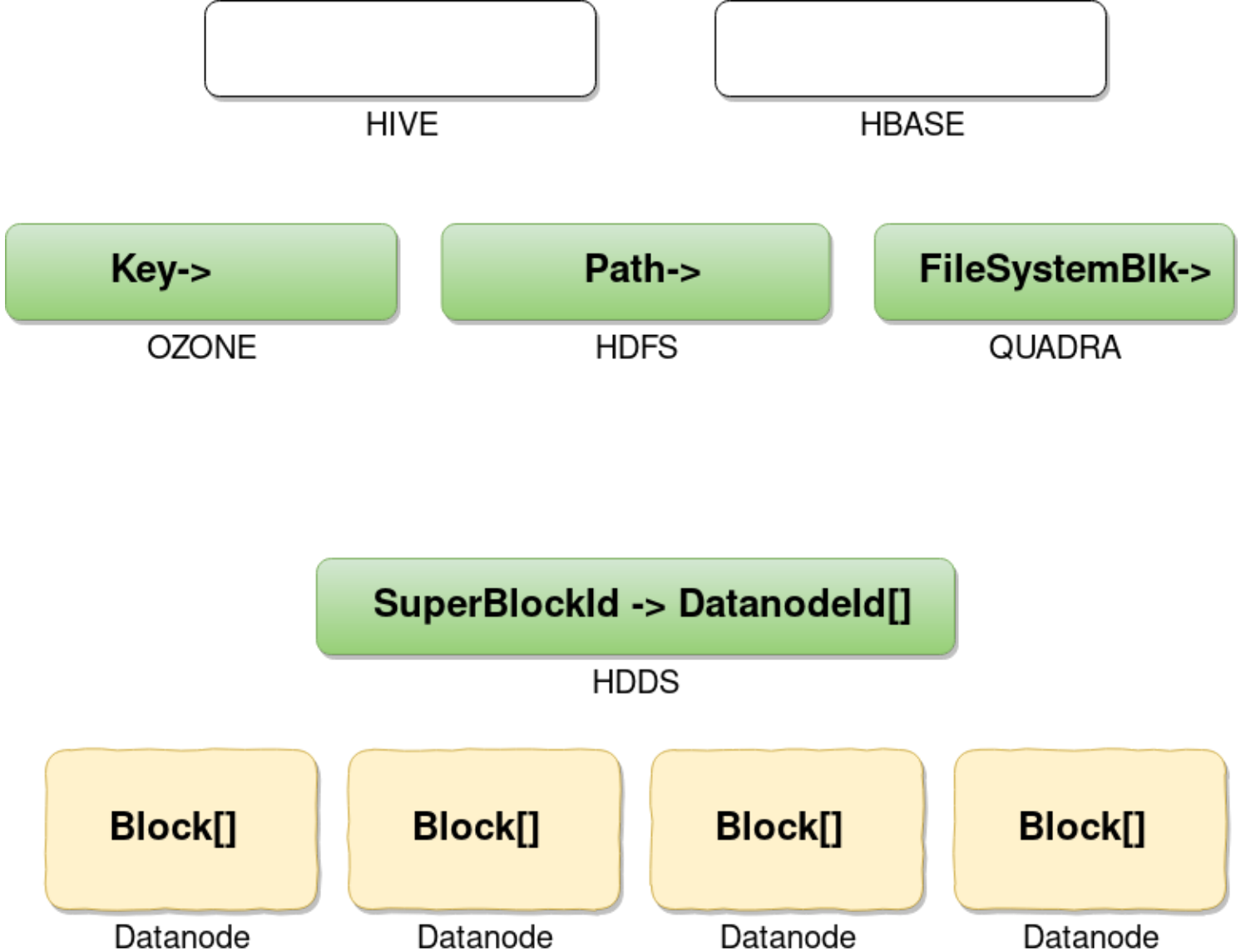

File -> BlockId[]
BlockId -> DatanodeId[]
HDDS
Block[]
Block[]
Block[]
Block[]
Datanode
Datanode
Datanode
Datanode


Key -> BlockId[]
Ozone
BlockId -> DatanodeId[]
HDDS
Block[]
Block[]
Block[]
Block[]
Datanode
Datanode
Datanode
Datanode


Key -> SuperBlockId[]
Ozone
SuperBlockId -> DatanodeId[]
HDDS
Block[]
Block[]
Block[]
Block[]
Datanode
Datanode
Datanode
Datanode


Key->
Path->
OZONE
HDFS
SuperBlockId -> DatanodeId[]
HDDS
Block[]
Block[]
Block[]
Block[]
Datanode
Datanode
Datanode
Datanode

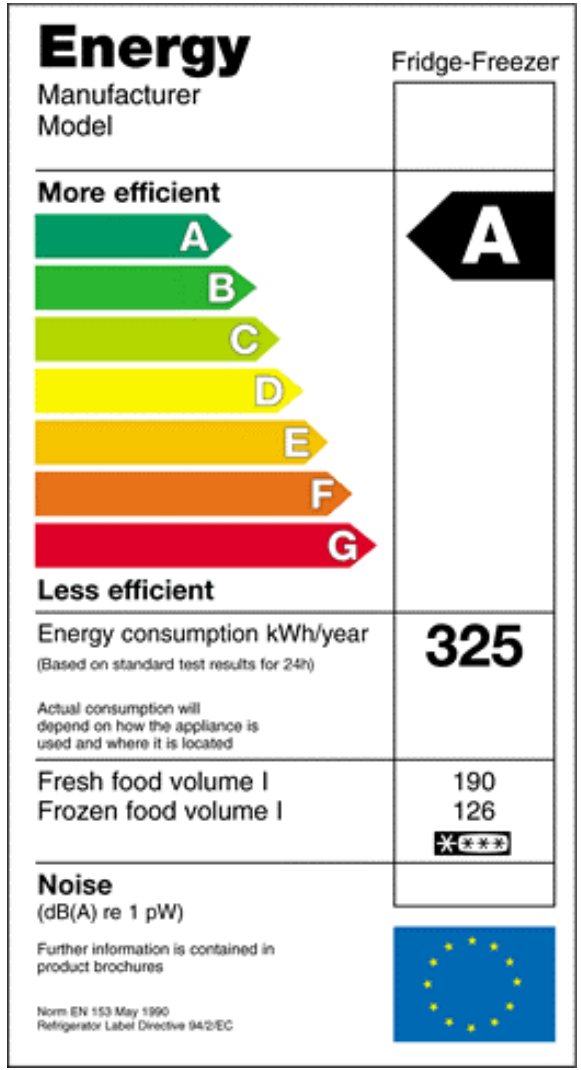

Key->
Path->
FileSystemBlk->
OZONE
HDFS
QUADRA
SuperBlockId -> DatanodeId[]
HDDS
Block[]
Block[]
Block[]
Block[]
Datanode
Datanode
Datanode
Datanode


HIVE
HBASE
Key->
Path->
FileSystemBlk->
OZONE
HDFS
QUADRA
SuperBlockId -> DatanodeId[]
HDDS
Block[]
Block[]
Block[]
Block[]
Datanode
Datanode
Datanode
Datanode

Hadoop  Docker

Dockerfile

```
FROM frolvlad/alpine-oraclejdk8
ADD hadoop-3.2.0.tar.gz /opt
WORKDIR /opt/hadoop
```


Energy
Fridge-Freezer
Manufacturer
Model
More efficient
A
B
C
D
E
F
G
A
Less efficient
Energy consumption kWh/year
(Based on standard test results for 24h)
325
Actual consumption will depend on how the appliance is used and where it is located
Fresh food volume l
190
Frozen food volume l
126
****
Noise
(dB(A) re 1 pW)
Further information is contained in product brochures
Norm EN 153 May 1990
Refrigerator Label Directive 94/2/EC

docker

Less efficient

Configuration management	
Source	
Preprocessing	
On change	
Provisioning, Scheduling	
Multihost support	
Scheduling	
Cluster definition	
Scaling	
Multi tenancy	
Failover	
Network	
Intraservice network	
DNS	
Service discovery	
Data locality	
Availability of the ports	

HADOOP-14898:
Dockerize Hadoop

for devs

for users

for users

for users

```
</configuration>
```

The dfs.client.failover.random.order set to true allows distributing the load evenly across the routers.

W... setting a u... can inte...t wi... -fs as a ...ular n...espa...

```
$ $HADOOP_HOM.../...s dfs ... hdfs://.../
/tmp
/data
```

This federated namespace can also be set as the default one at **core-site.xml** using fs.defaultFS.

Router configuration

One can add the configurations for Router-based federation to **hdfs-site.xml**. The main options are documented in hdfs-default.xml. The configuration values are described in this section.

RPC server

The RPC server to receive connections from the clients.

Property	Default	Description
dfs.federation.router.default.nameserviceId		Nameservice identifier of the default subcluster to monitor.
dfs.federation.router.rpc.enable	true	If true, the RPC service to handle client requests in the router is enabled.
dfs.federation.router.rpc-address	0.0.0.0:8888	RPC address that handles all clients requests.
dfs.federation.router.rpc-bind-host	0.0.0.0	The actual address the RPC server will bind to.
dfs.federation.router.handler.count	10	The number of server threads for the router to handle RPC requests from clients.
dfs.federation.router.handler.queue.size	100	The size of the queue for the number of handlers to handle RPC client requests.
dfs.federation.router.reader.count	1	The number of readers for the router to handle RPC client requests.
dfs.federation.router.reader.queue.size	100	The size of the queue for the number of readers for the router to handle RPC client requests.

Connection to the Namenodes

The Router forwards the client requests to the NameNodes. It uses a pool of connections to reduce the latency of creating them.

Property	Default	Description
dfs.federation.router.connection.pool-size	1	Size of the pool of connections from the router to namenodes.
dfs.federation.router.connection.clean.ms	10000	Time interval, in milliseconds, to check if the connection pool should remove unused connections.
dfs.federation.router.connection.pool.clean.ms	60000	Time interval, in milliseconds, to check if the connection manager should remove unused connection pools.

Admin server

The administration server to manage the Mount Table.

Property	Default	Description
dfs.federation.router.admin.enable	true	If true, the RPC admin service to handle client requests in the router is enabled.
dfs.federation.router.admin-address	0.0.0.0:8111	RPC address that handles the admin requests.
dfs.federation.router.admin-bind-host	0.0.0.0	The actual address the RPC admin server will bind to.
dfs.federation.router.admin.handler.count	1	The number of server threads for the router to handle RPC requests from admin.

State Store

The connection to the State Store and the internal caching at the Router.

Property	Default
dfs.federation.router.store.enable	true
dfs.federation.router.store.serializer	org.apache.hadoop.hdfs.server.federation.store.driver.impl.StateStoreSerializerPBImpl
dfs.federation.router.store.driver.class	org.apache.hadoop.hdfs.server.federation.store.driver.impl.StateStoreZooKeeperImpl
dfs.federation.router.store.connection.test	60000
dfs.federation.router.cache.ttl	60000
dfs.federation.router.store.membership.expiration	300000

Routing

Forwarding client requests to the right subcluster.

Property	Default
dfs.federation.router.file.resolver.client.class	org.apache.hadoop.hdfs.server.federation.resolver.MountTableResolver
dfs.federation.router.namenode.resolver.client.class	org.apache.hadoop.hdfs.server.federation.resolver.MembershipNamenodeResolver

Namenode monitoring

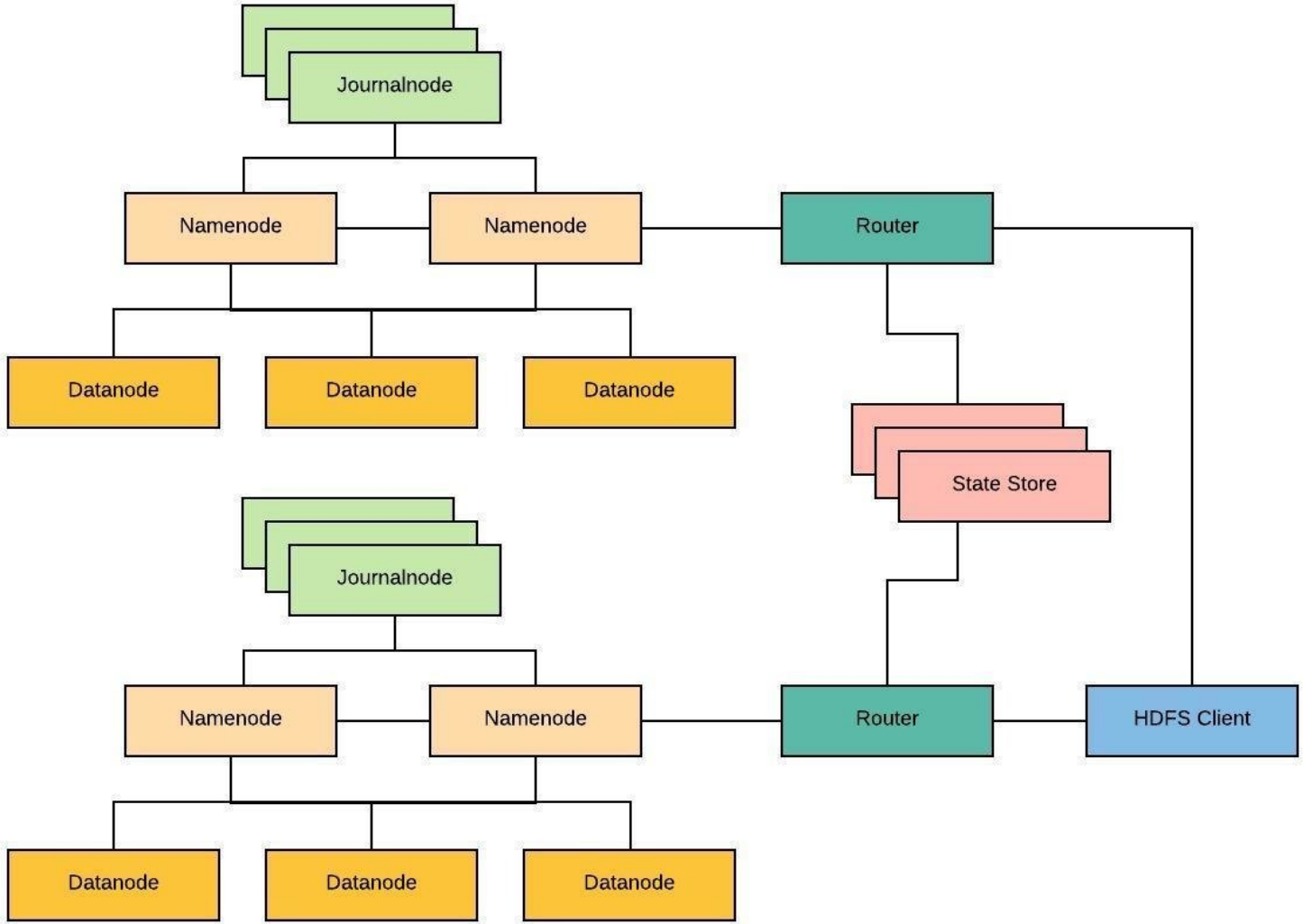
Monitor the namenodes in the subclusters for forwarding the client requests.

Property	Default	Description
dfs.federation.router.heartbeat.enable	true	If true, the Router heartbeats into the State Store.
dfs.federation.router.heartbeat.interval	5000	How often the Router should heartbeat into the State Store in milliseconds.
dfs.federation.router.monitor.namenode		The identifier of the namenodes to monitor and heartbeat.
dfs.federation.router.monitor.localnamenode.enable	true	If true, the Router should monitor the namenode in the local machine.

Note: The config *dfs.nameservice.id* is recommended to configure if *dfs.federation.router.monitor.localnamenode.enable* is enabled. This will allow the Router finding the local node directly. Otherwise, it will find the nameservice Id by matching namenode RPC address with the local node address. If multiple addresses are matched, the Router

Metrics

The Router and State Store statistics are exposed in metrics/JMX. These info will be very useful for monitoring. More metrics info can see Router RPC Metrics and State Store Metrics.


Journalnode
Namenode
Namenode
Router
Datanode
Datanode
Datanode
State Store
Journalnode
Namenode
Namenode
Router
HDFS Client
Datanode
Datanode
Datanode

flokkr / **runtime-compose**

Watch 2 | Star 3 | Fork 5

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

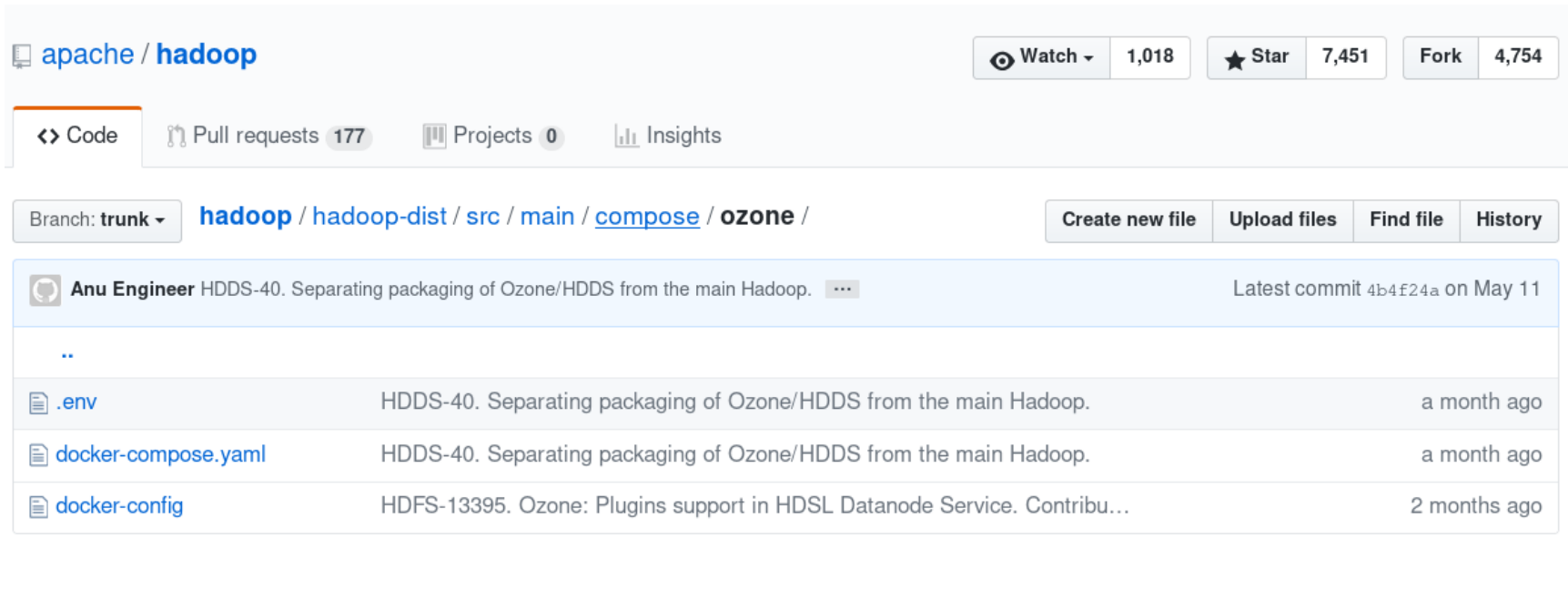
Branch: **master** **runtime-compose** / **hdfs-routerfeder** /

Create new file | Upload files | Find file | History

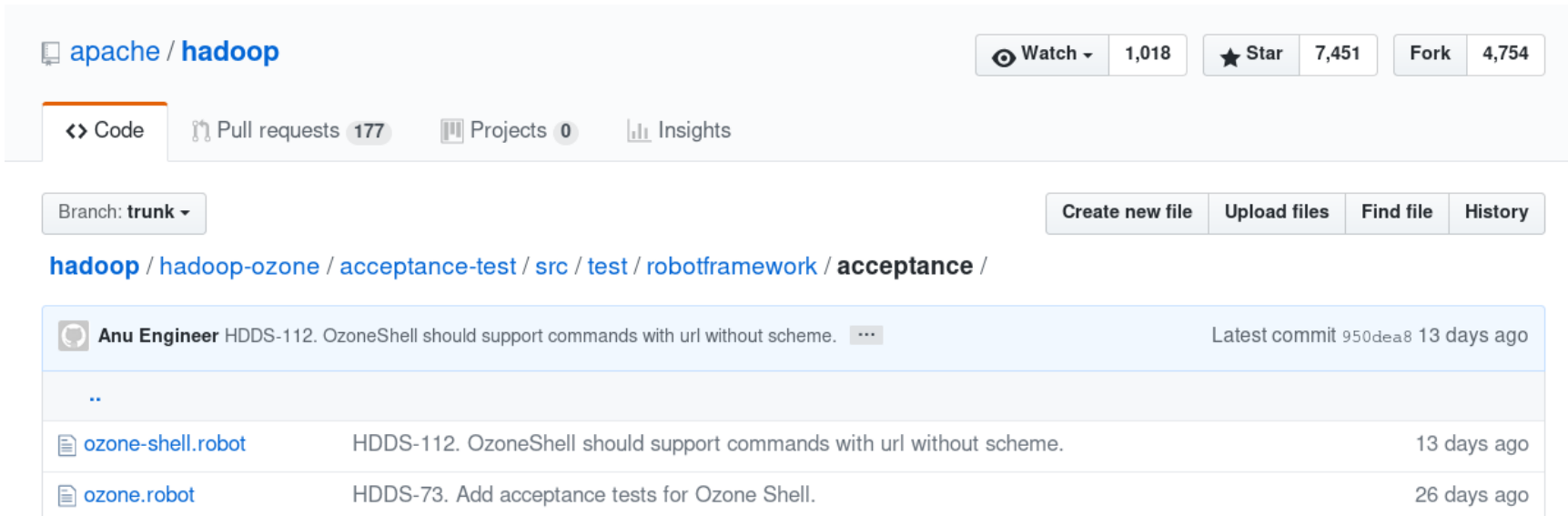
elek cleanup hdfs cluster configuration — Latest commit `2f1fcc0` on Jan 18

..		
.env	hdfs router based federation example	5 months ago
c1compose.yaml	cleanup hdfs cluster configuration	5 months ago
c1config	hdfs router based federation example	5 months ago
c2compose.yaml	cleanup hdfs cluster configuration	5 months ago
c2config	hdfs router based federation example	5 months ago
docker-compose.yaml	hdfs router based federation example	5 months ago
router1	hdfs router based federation example	5 months ago
router2	hdfs router based federation example	5 months ago

for development



for development



dev cluster

Base: simple empty image (apache/hadoop-runner)

Mount:

hadoop-dist/target/hadoop-3.2.0-SNAPSHOT --> /opt/hadoop

Goal:

Check dev version/ acceptance tests

feature demo

Base: Hadoop releases (apache/hadoop:3)

Mount:

n/a (hadoop is included)

Goal:

Documentation / reference configuration

HADOOP-14898

dev cluster

Base: simple empty image (apache/hadoop-runner)

Mount:

hadoop-dist/target/hadoop-3.2.0-SNAPSHOT --> /opt/hadoop

Goal:

Check dev version/ acceptance tests

feature demo

Base: Hadoop releases (apache/hadoop:3)

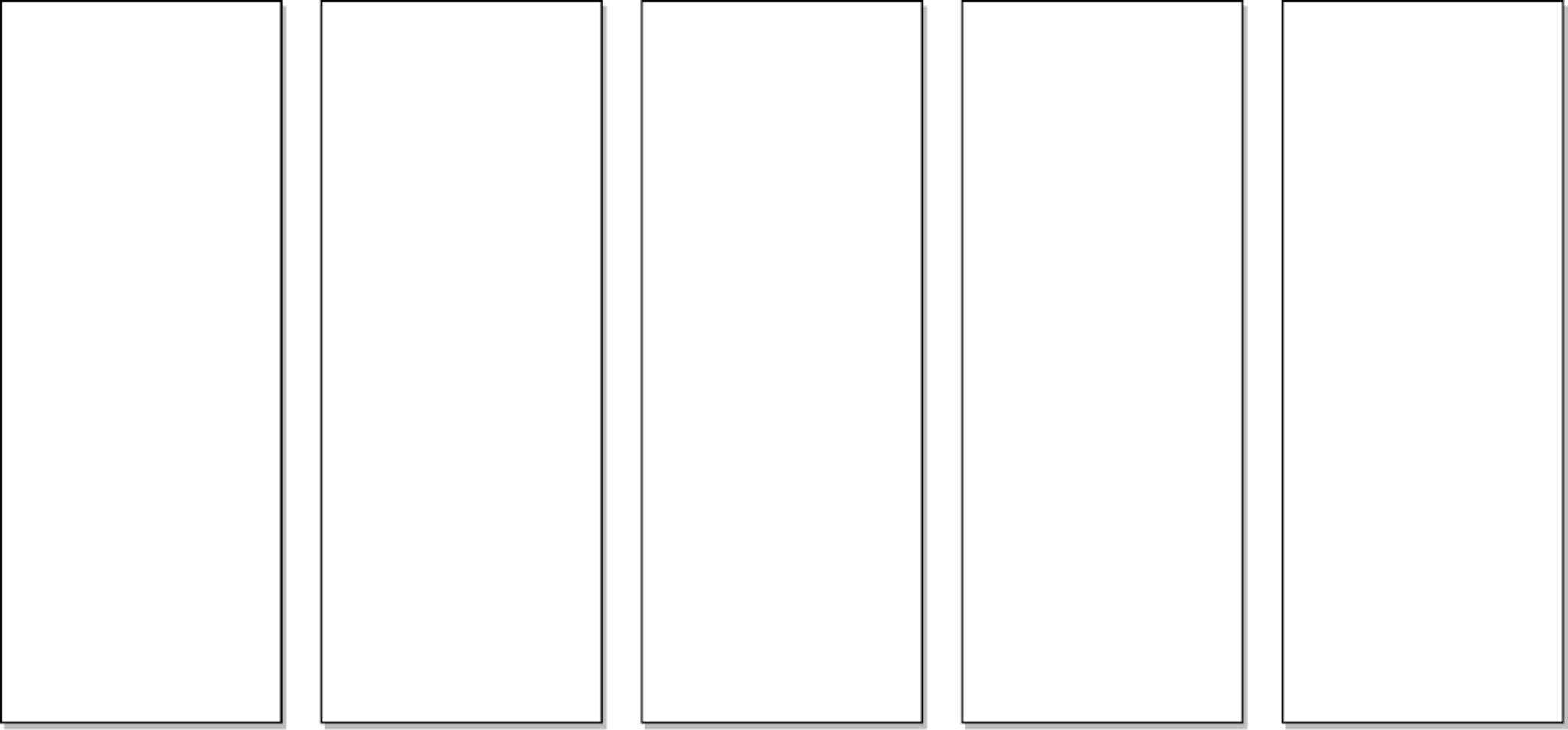
Mount:

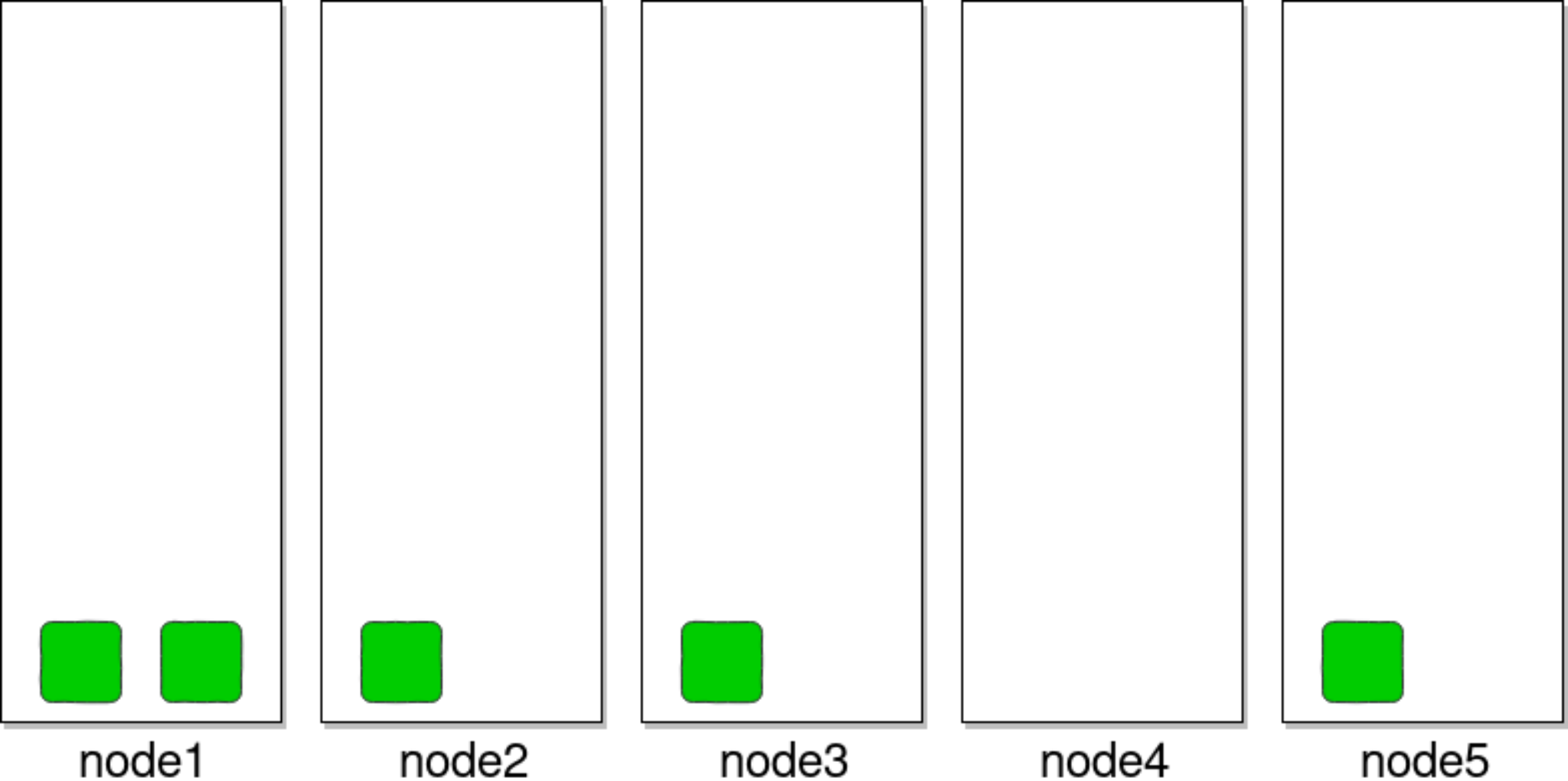
n/a (hadoop is included)

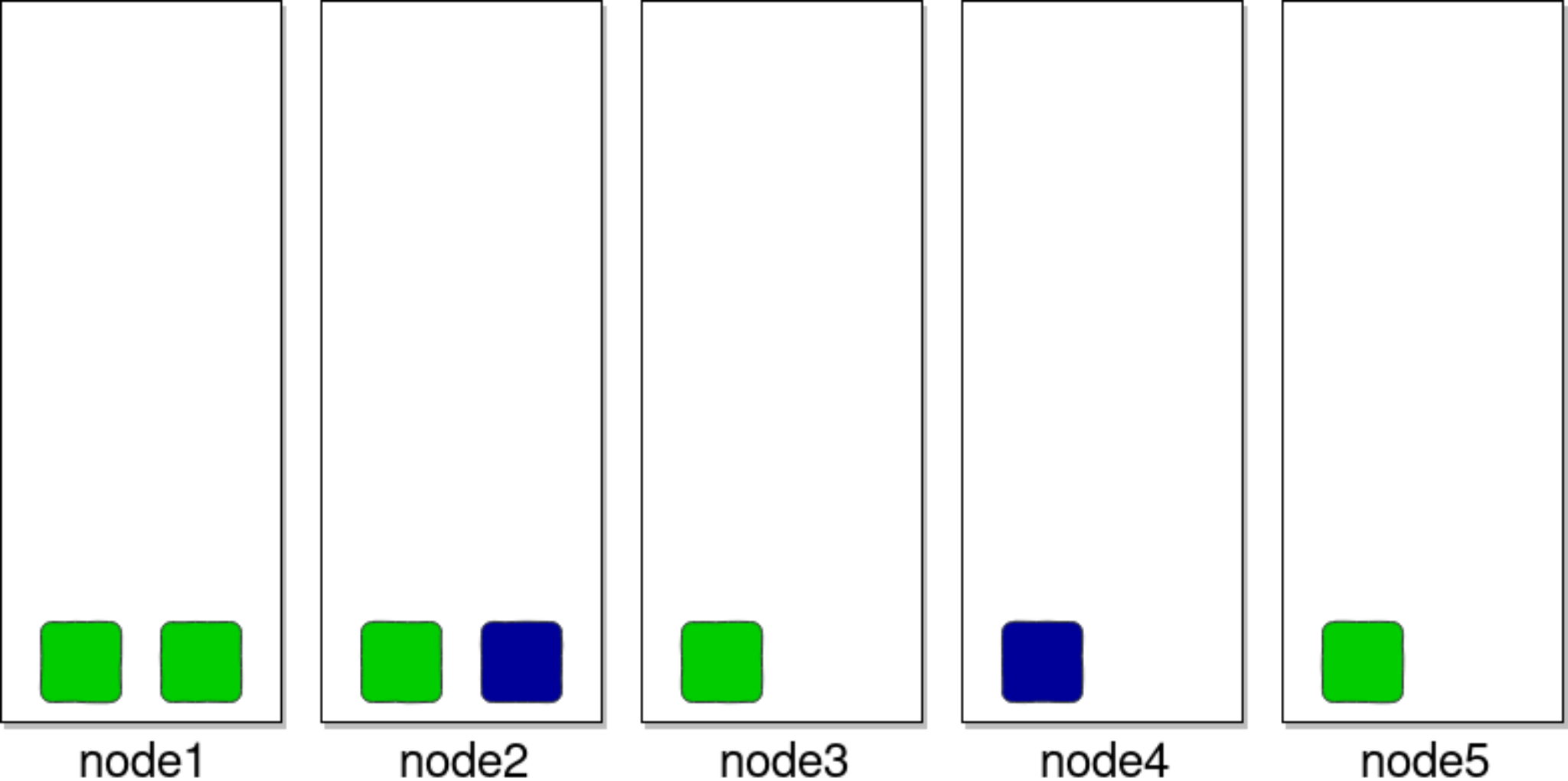
Goal:

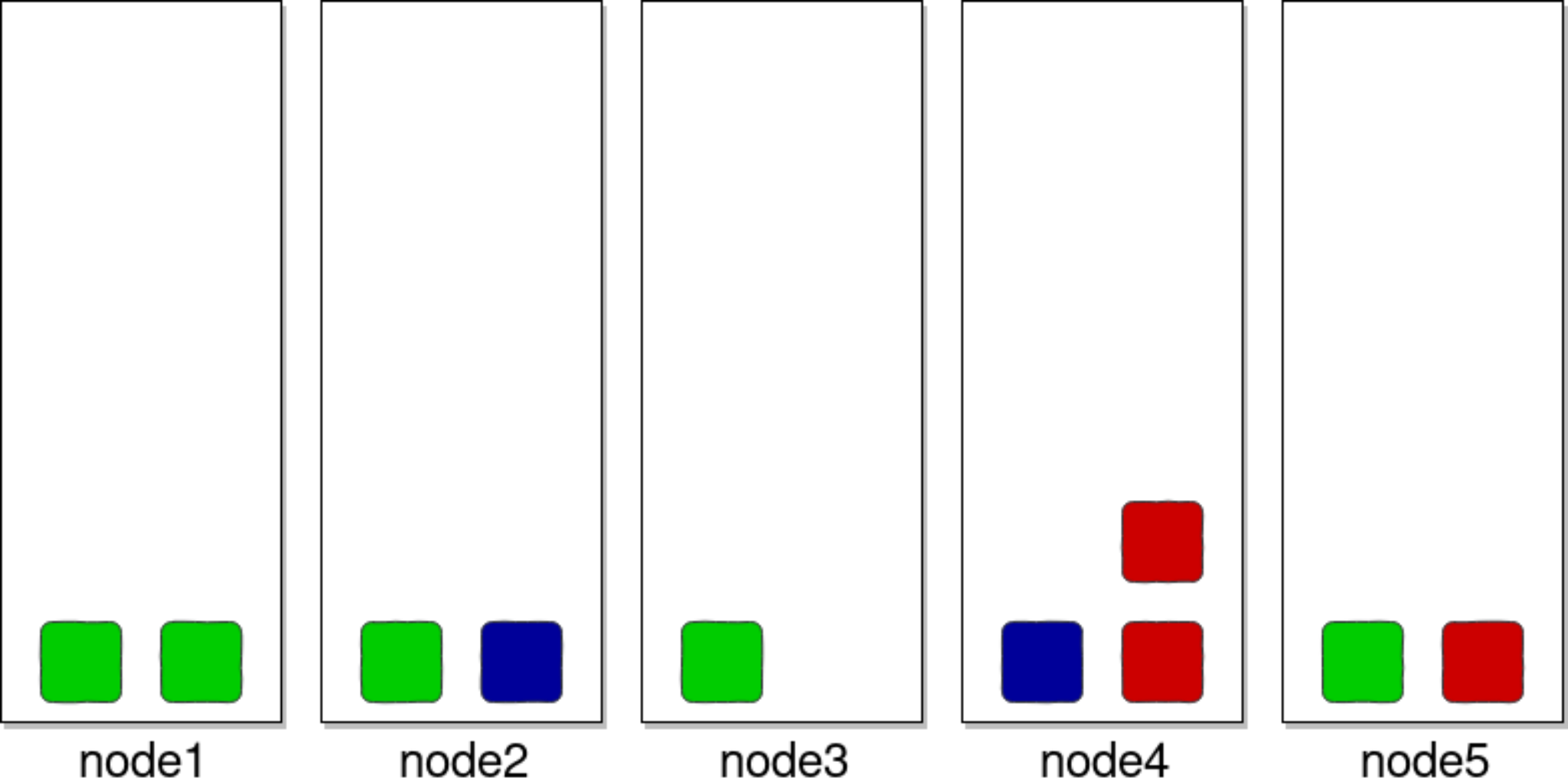
Documentation / reference configuration

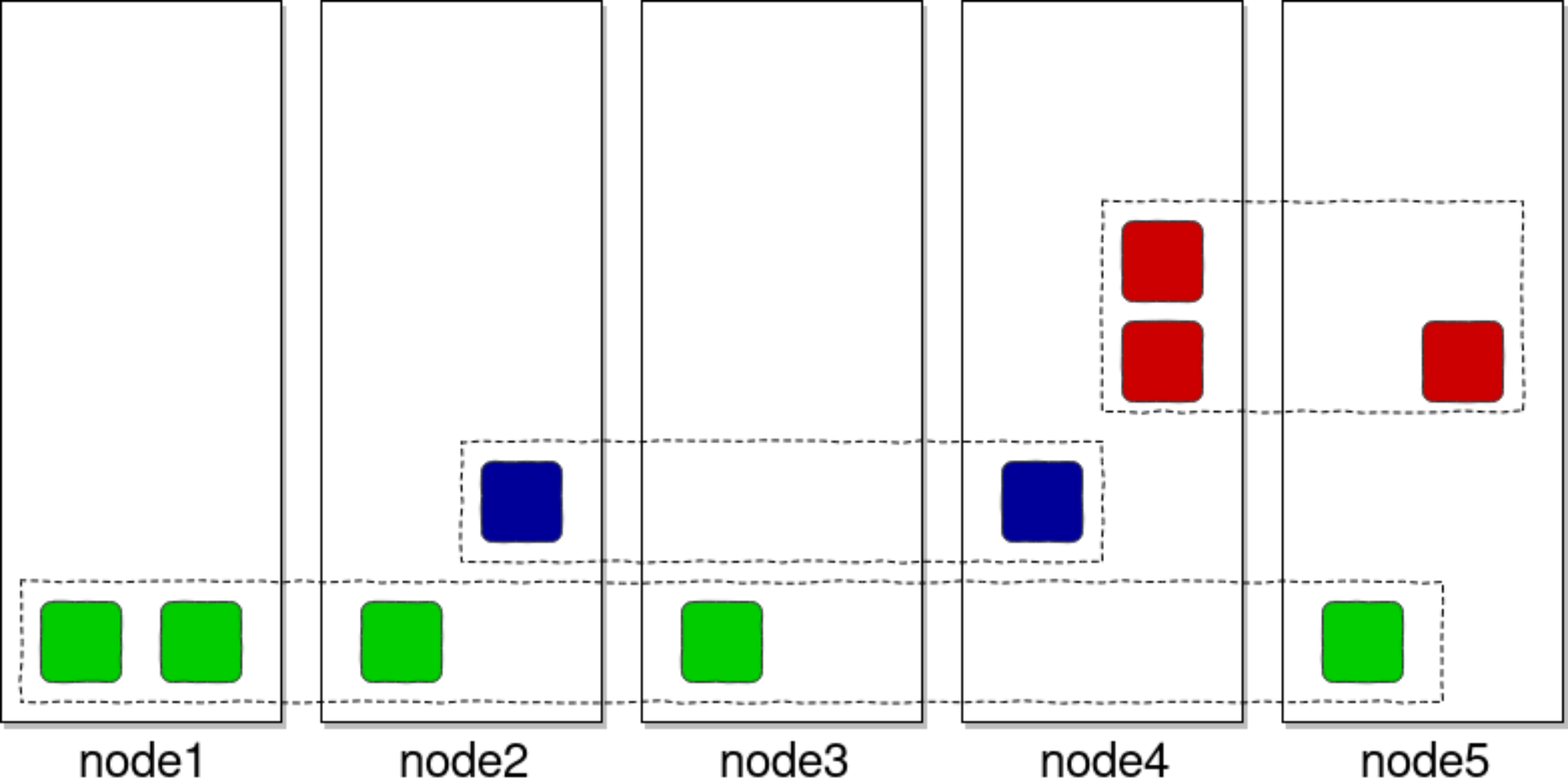
Hadoop  Kubernetes


node1
node2
node3
node4
node5


node1
node2
node3
node4
node5

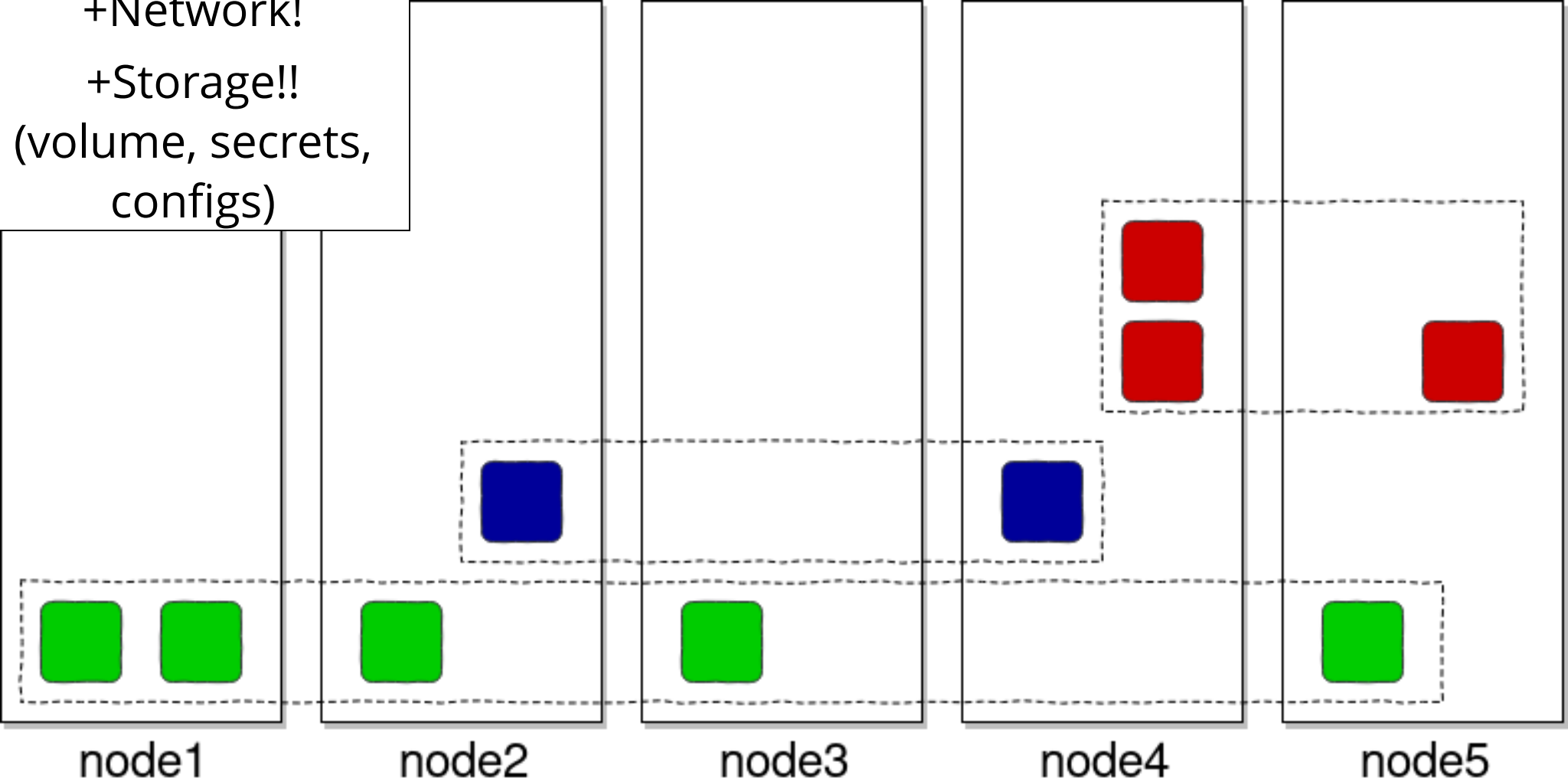

node1
node2
node3
node4
node5

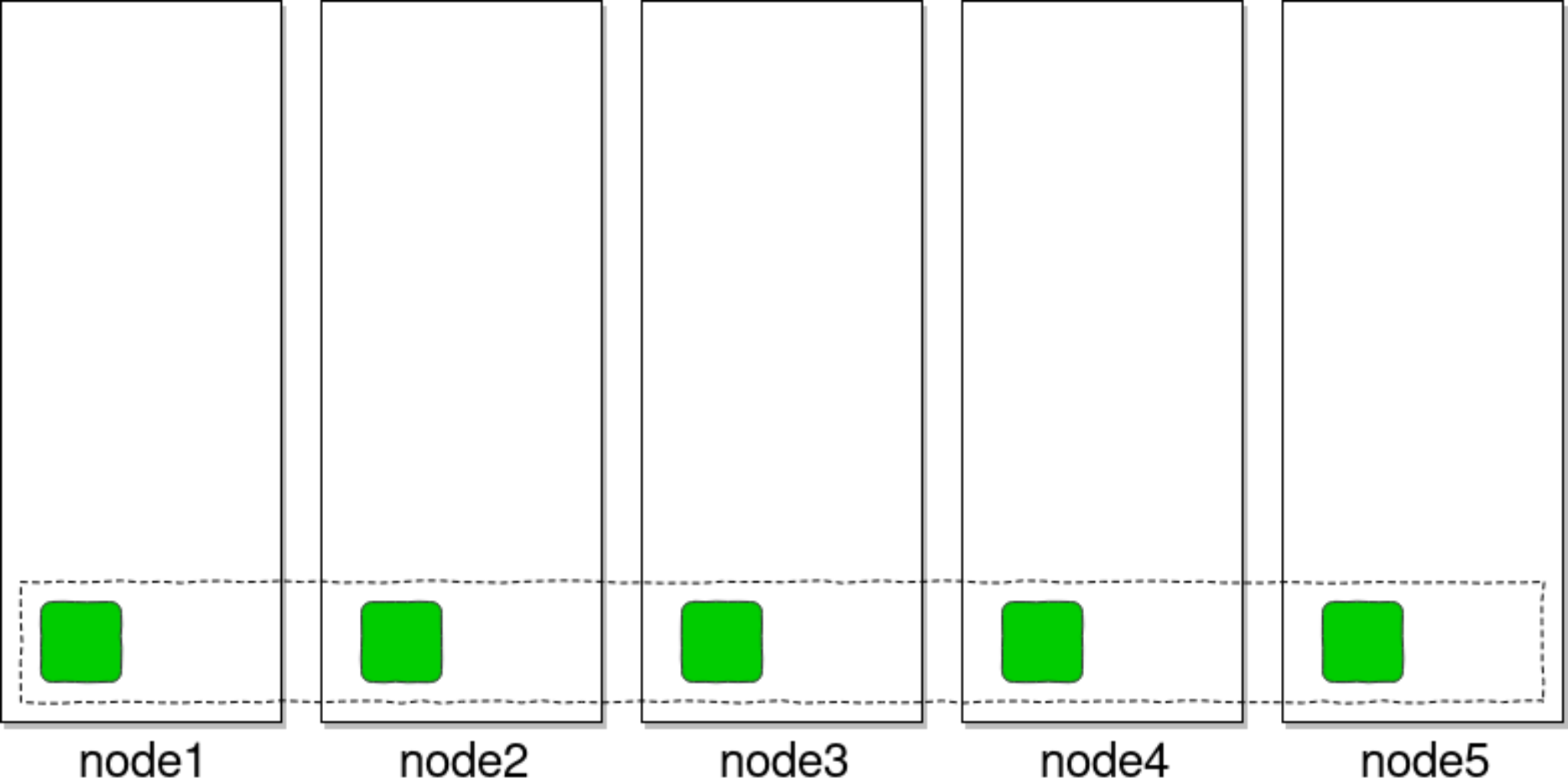

node1
node2
node3
node4
node5


node1
node2
node3
node4
node5

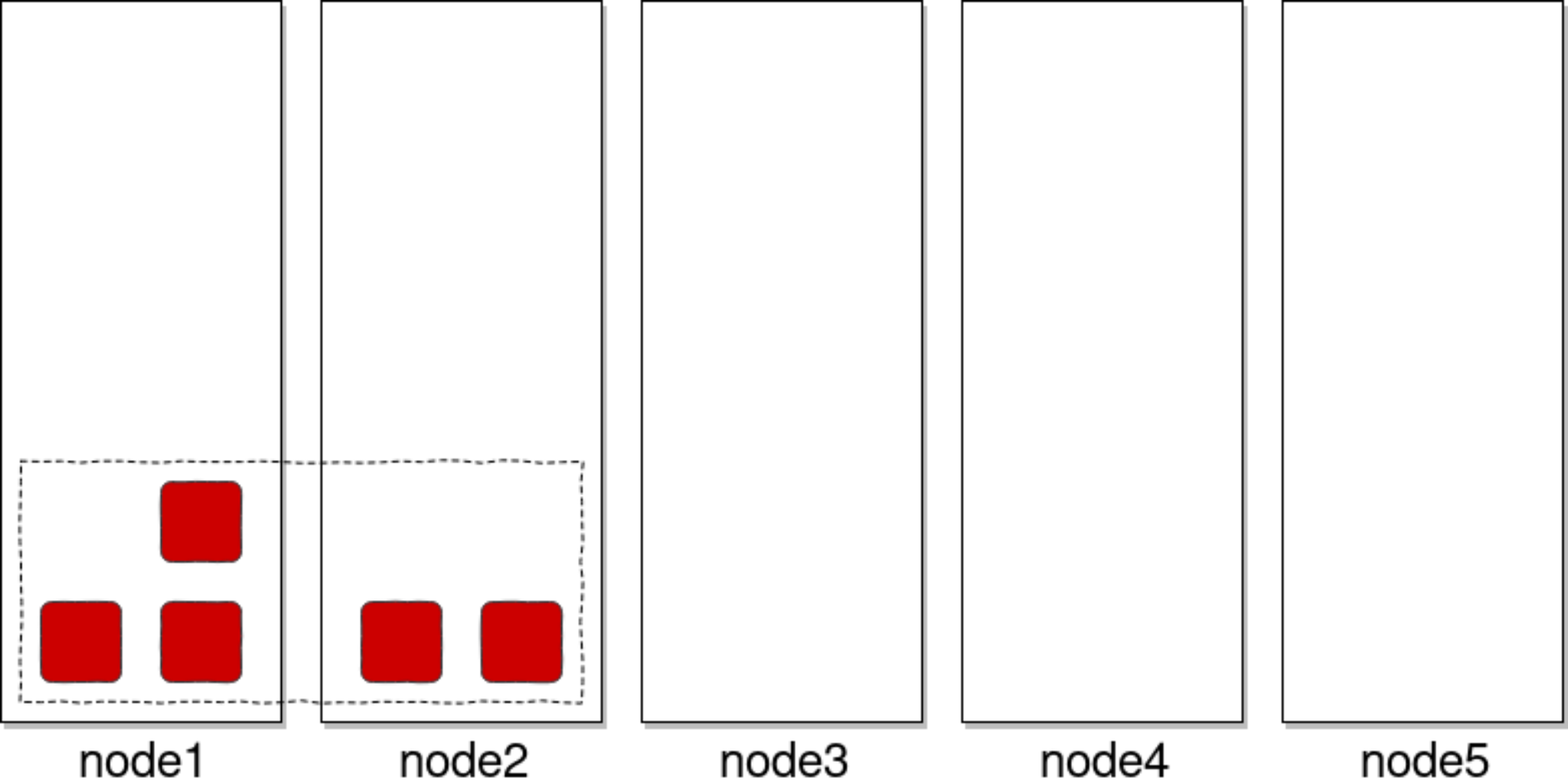
+Network!

+Storage!!
(volume, secrets, configs)

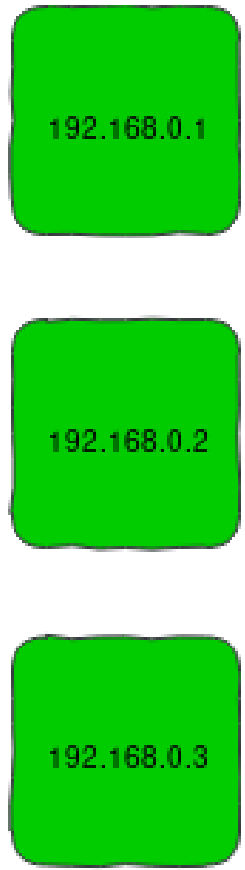


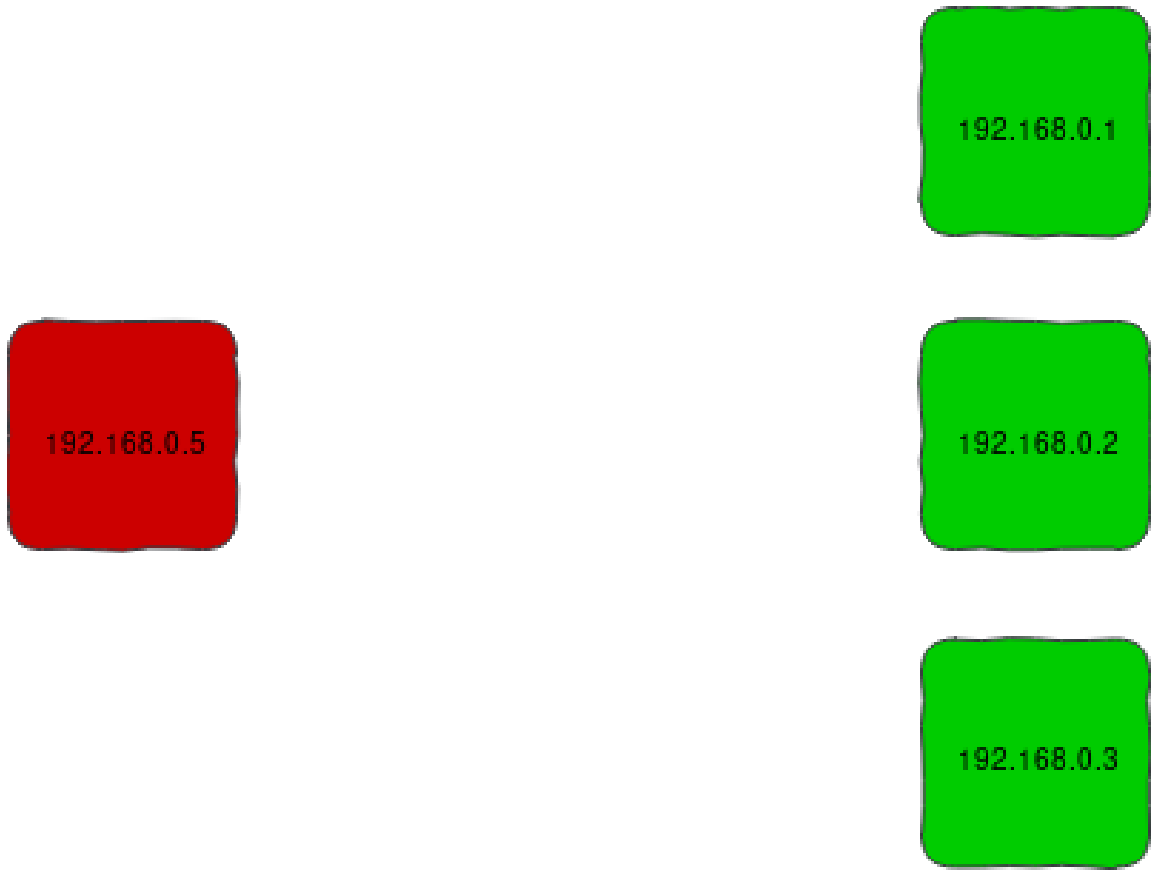


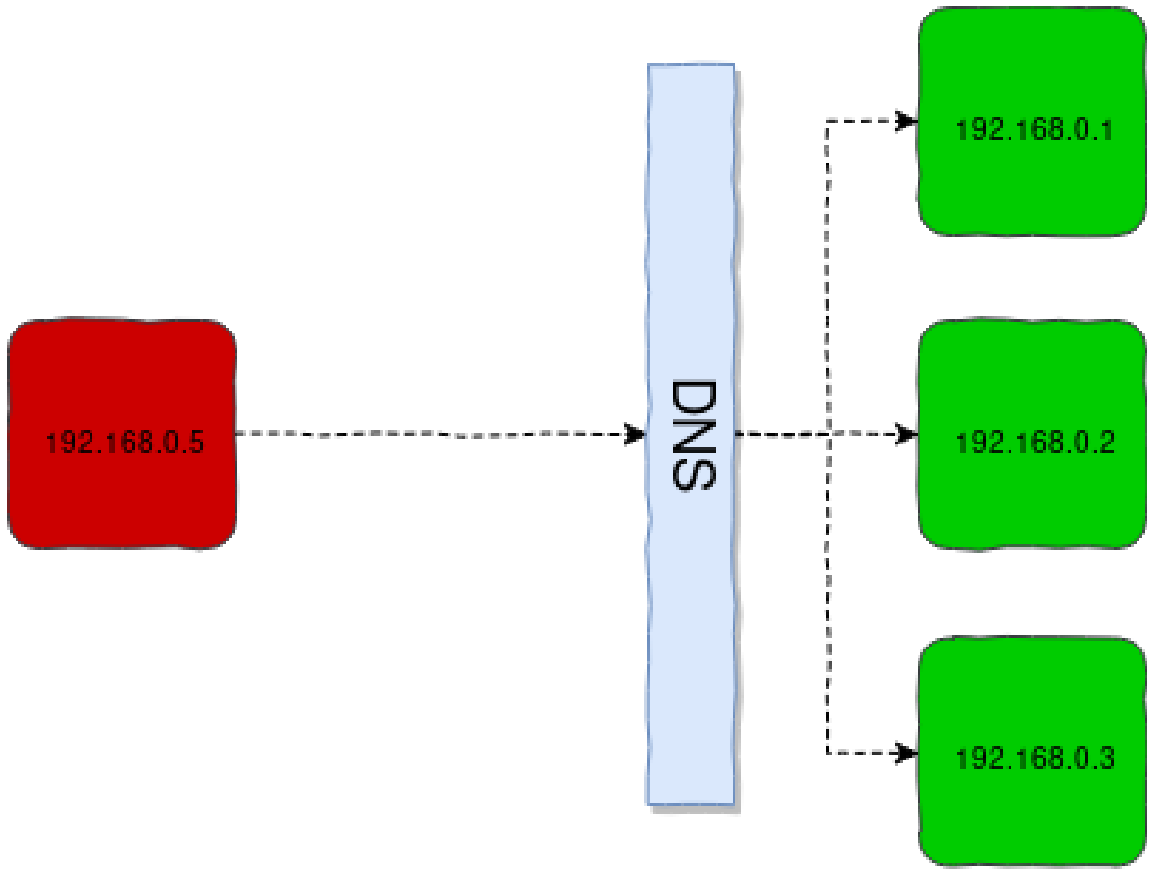
DaemonSet

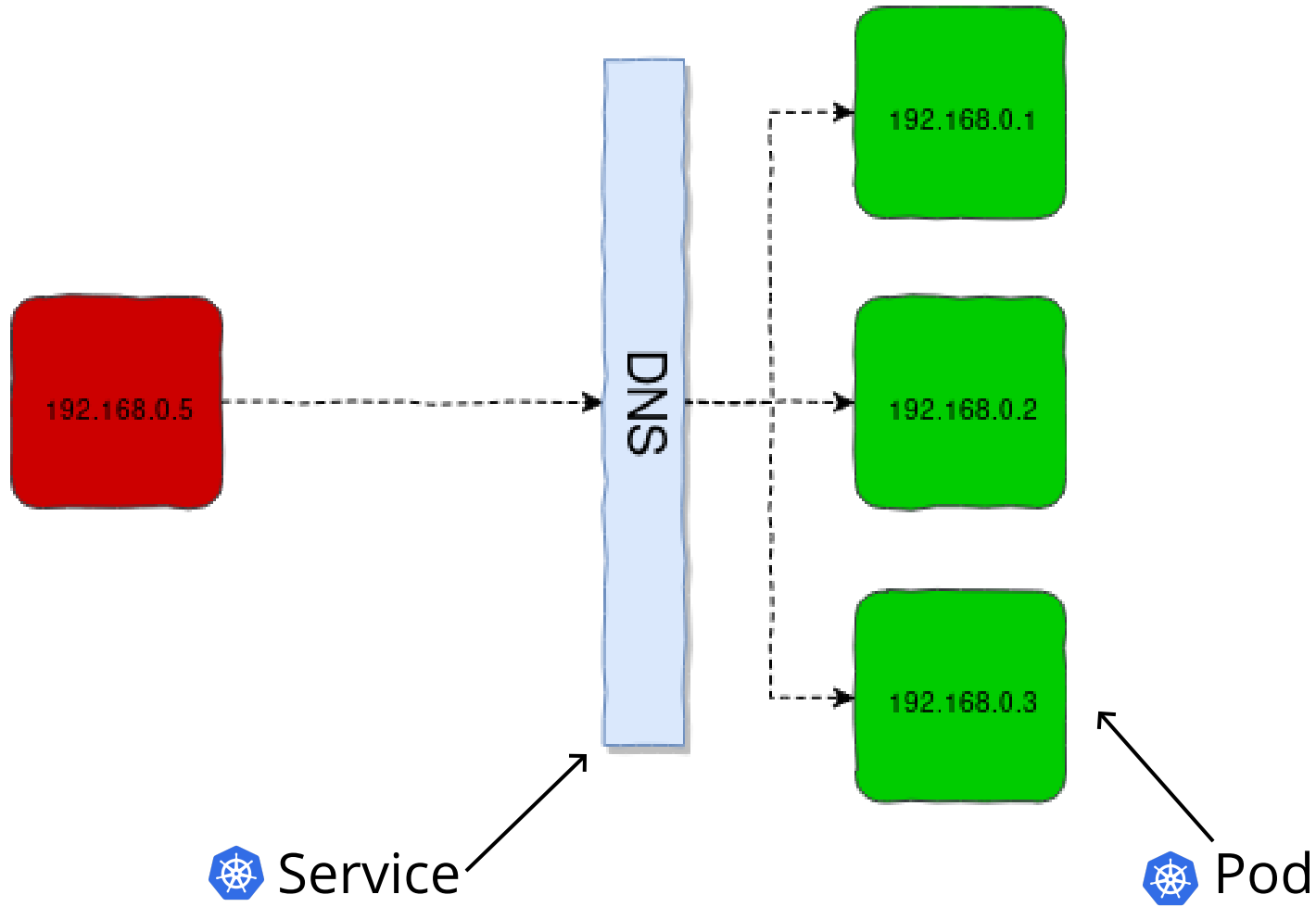


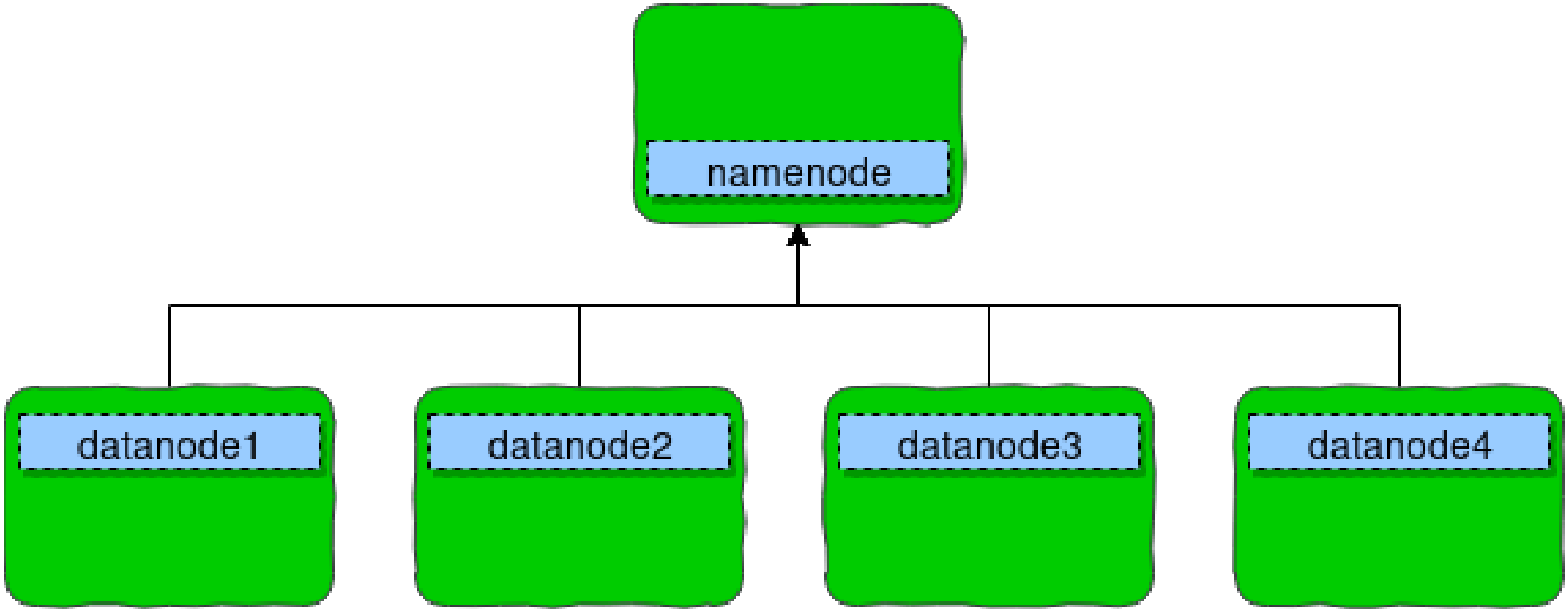
ReplicaSet


192.168.0.1
192.168.0.2
192.168.0.3

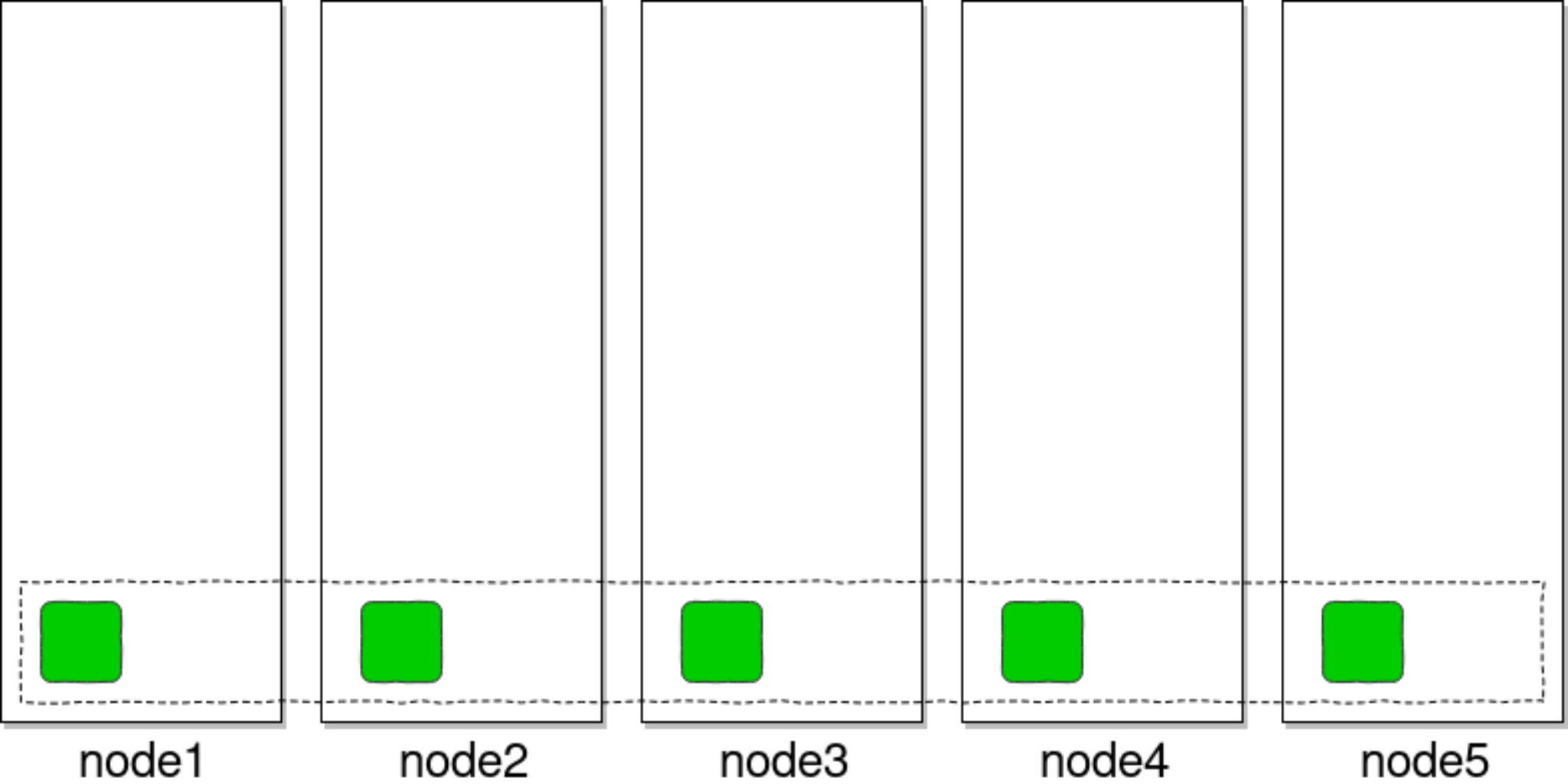

192.168.0.1
192.168.0.5
192.168.0.2
192.168.0.3


192.168.0.5
DNS
192.168.0.1
192.168.0.2
192.168.0.3


192.168.0.5
DNS
192.168.0.1
192.168.0.2
192.168.0.3
Service
Pod



StatefulSet



DaemonSet

Benefits of Hadoop + k8s?

Benefits of Hadoop + k8s?

Ecosystem

Flexibility

Ecosystem

Flexibility

Cloud-agnostic

Monitor Hadoop with Prometheus

Prometheus Alerts Graph Status Help

Enable query history

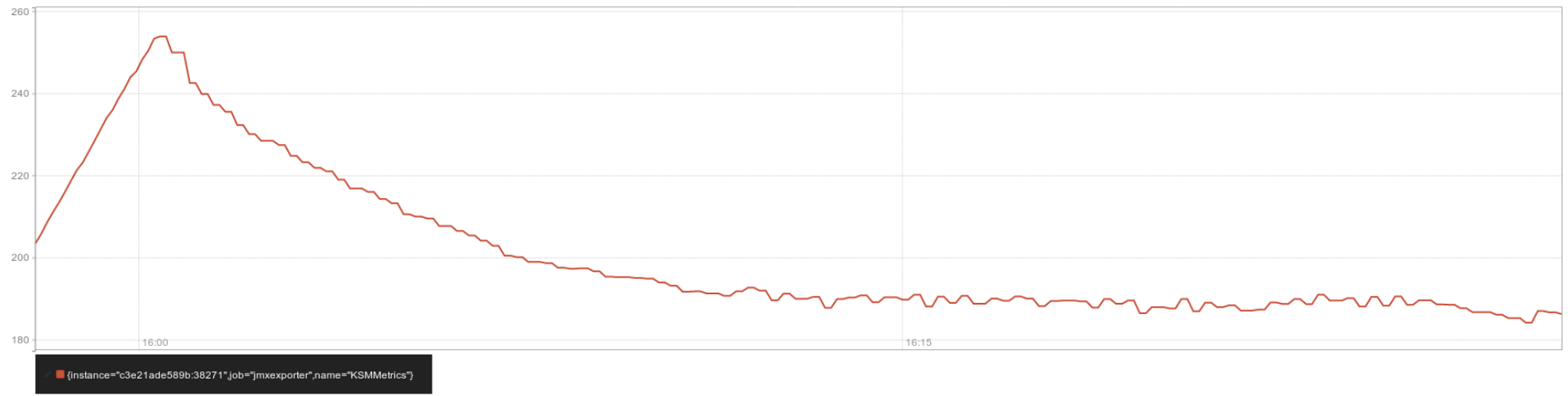
rate(Hadoop_KeySpaceManager_NumKeyCommits[10m])

Load time: 187ms
Resolution: 7s
Total time series:

Execute java_lang_OperatingSyste

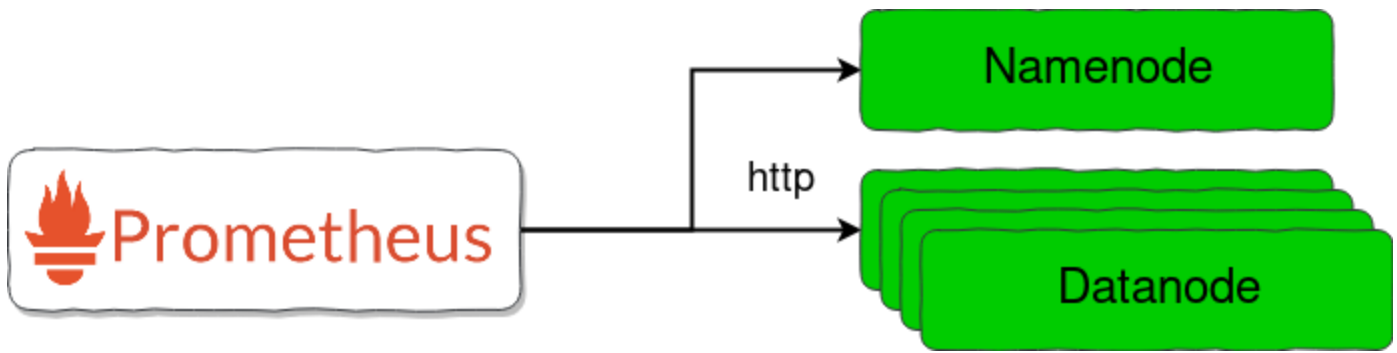
Graph Console

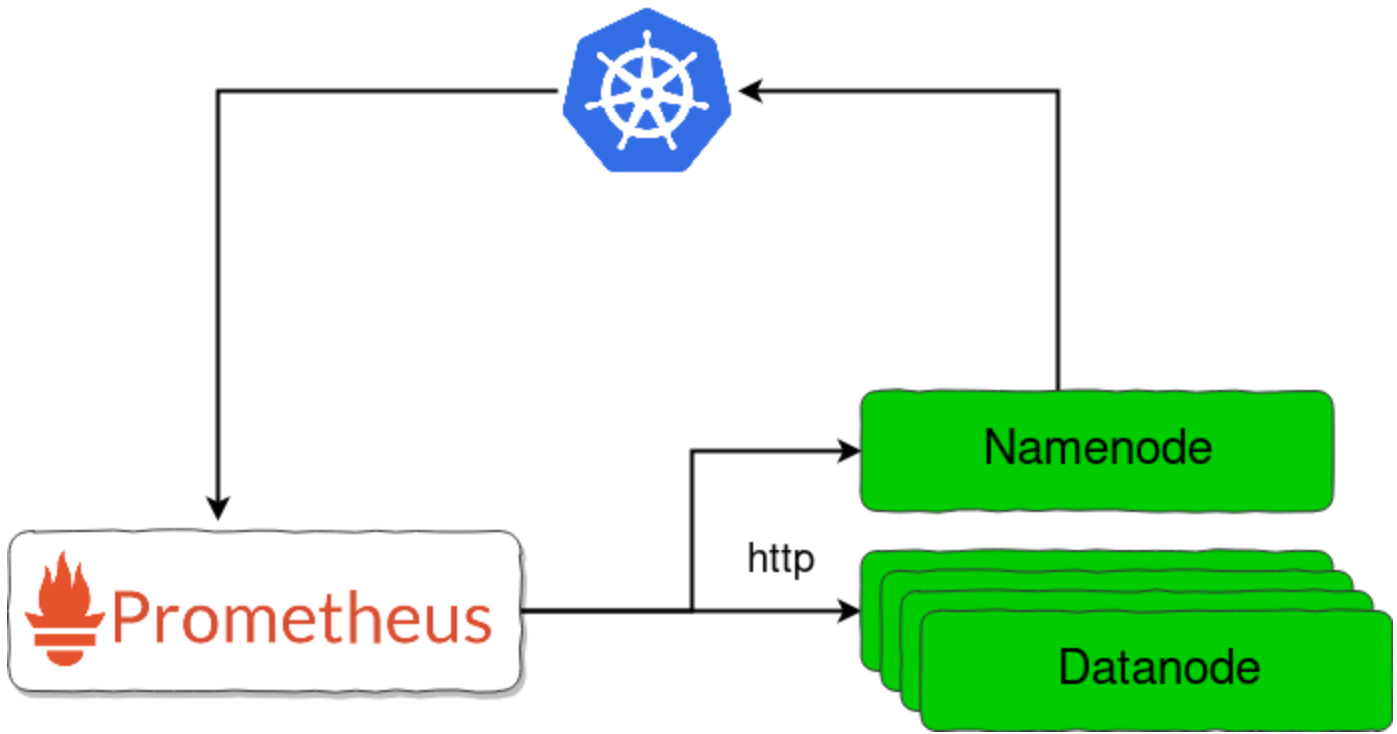
30m Until Res. (s) stacked



Remove Grap

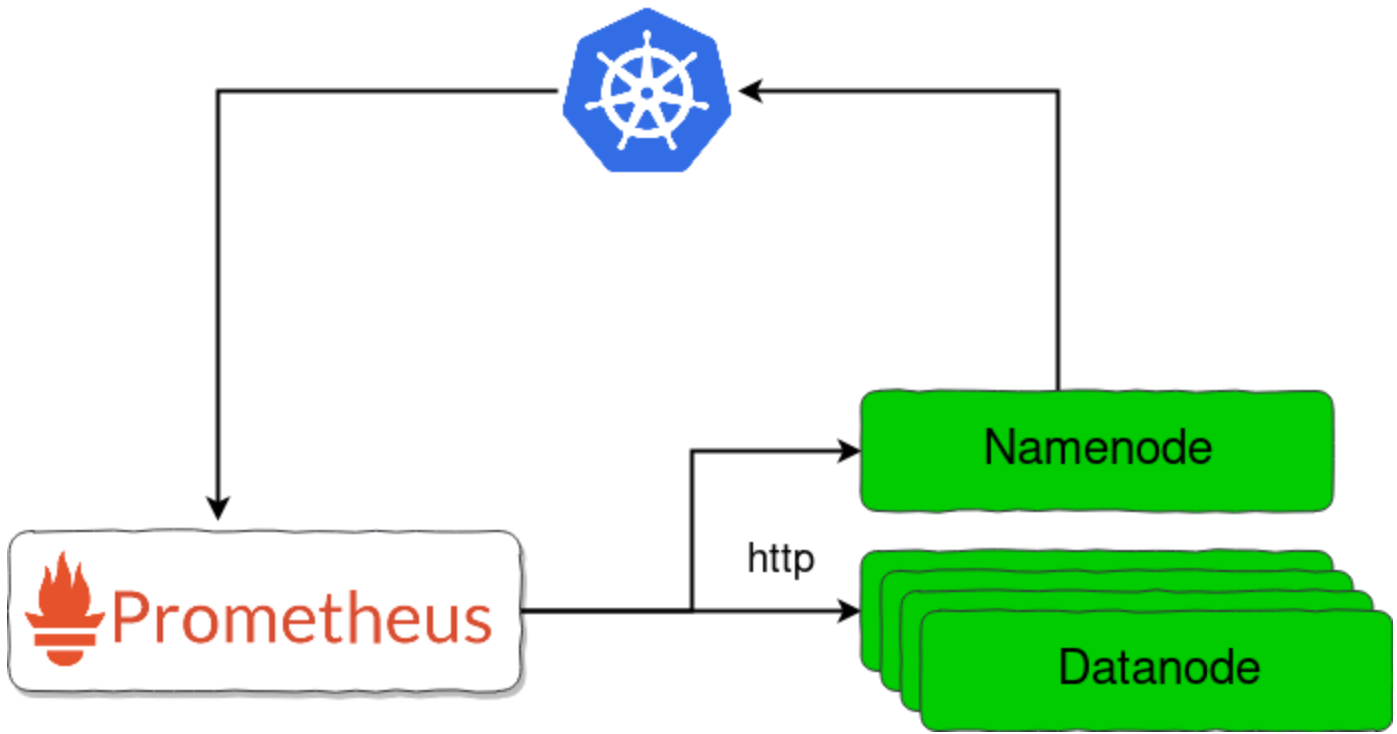
Add Graph

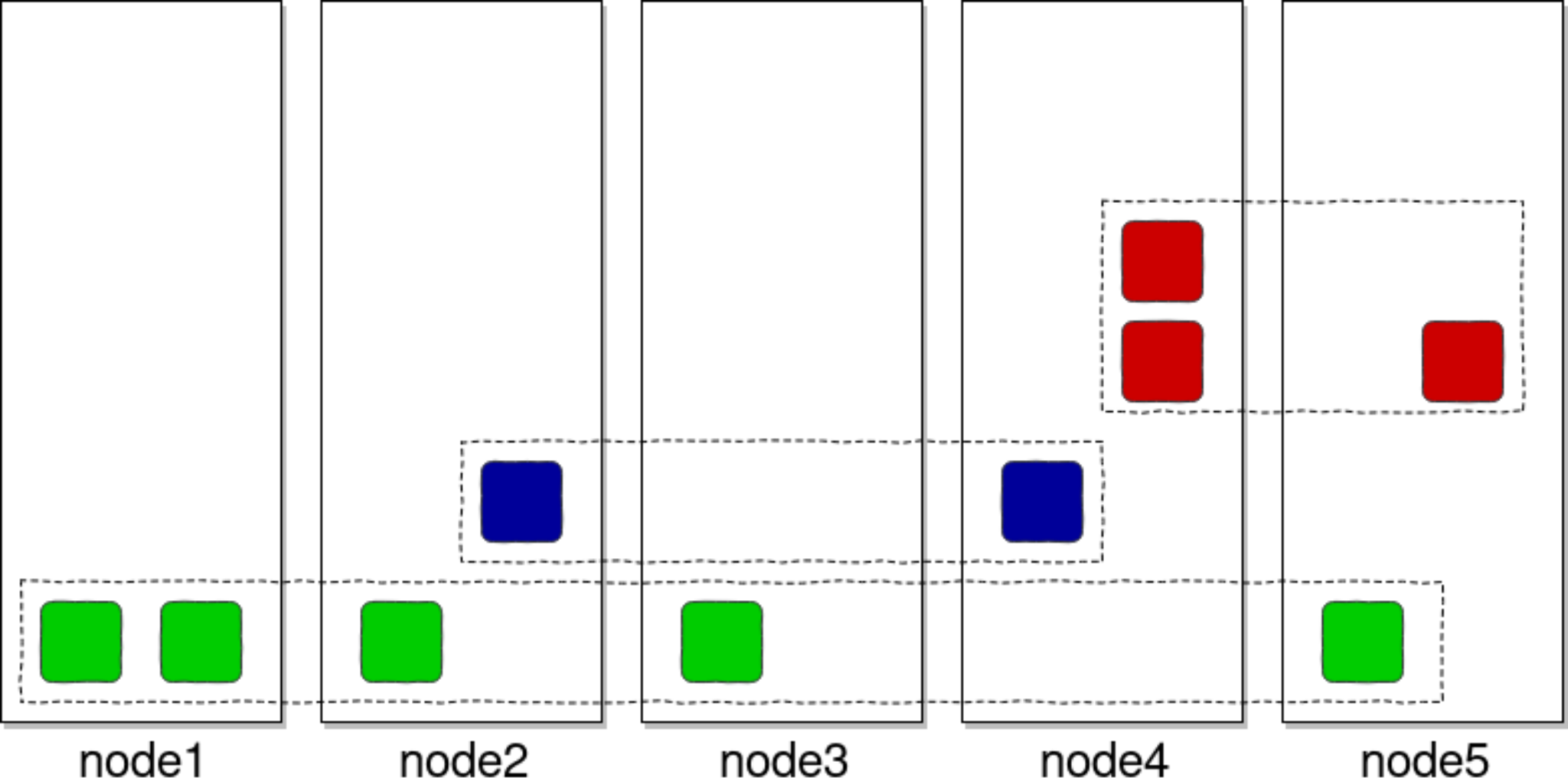

Namenode
http
Prometheus
Datanode

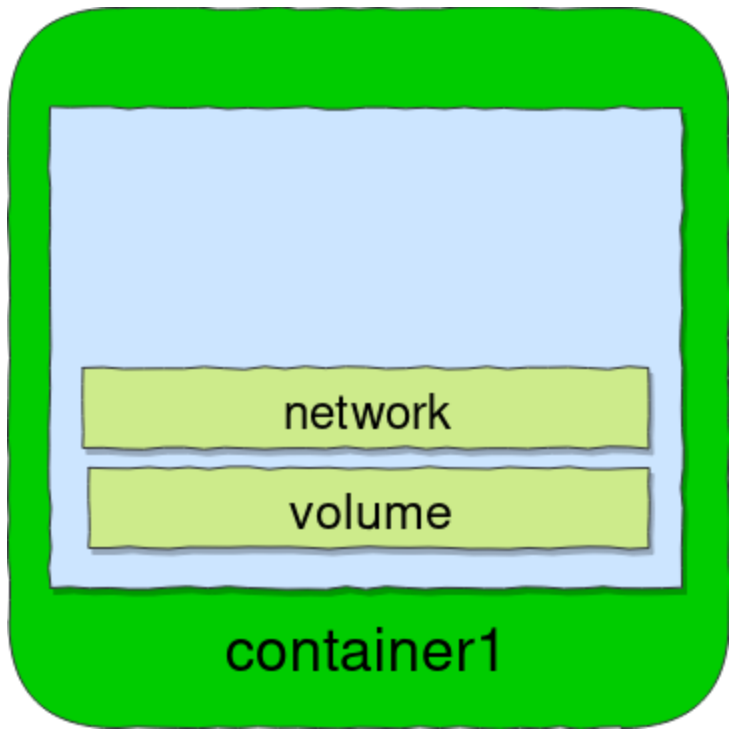

Namenode
http
Prometheus
Datanode

```yaml
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: ozone-hdfs-namenode
spec:
  serviceName: ozone2-hdfs-namenode
  replicas: 1
  template:
    metadata:
      labels:
        app: ozone
    spec:
      containers:
        - name: hdfs-namenode
          image: flokkr/ozone:2.1.0
          args: ["hdfs","namenode"]
```

```yaml
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: ozone-hdfs-namenode
spec:
  serviceName: ozone2-hdfs-namenode
  replicas: 1
  template:
    metadata:
      labels:
        app: ozone
      annotations:
        prometheus.io/scrape: "true"
        prometheus.io/port: "28942"
    spec:
      containers:
        - name: hdfs-namenode
          image: flokkr/ozone:2.1.0
          args: ["hdfs","namenode"]
```

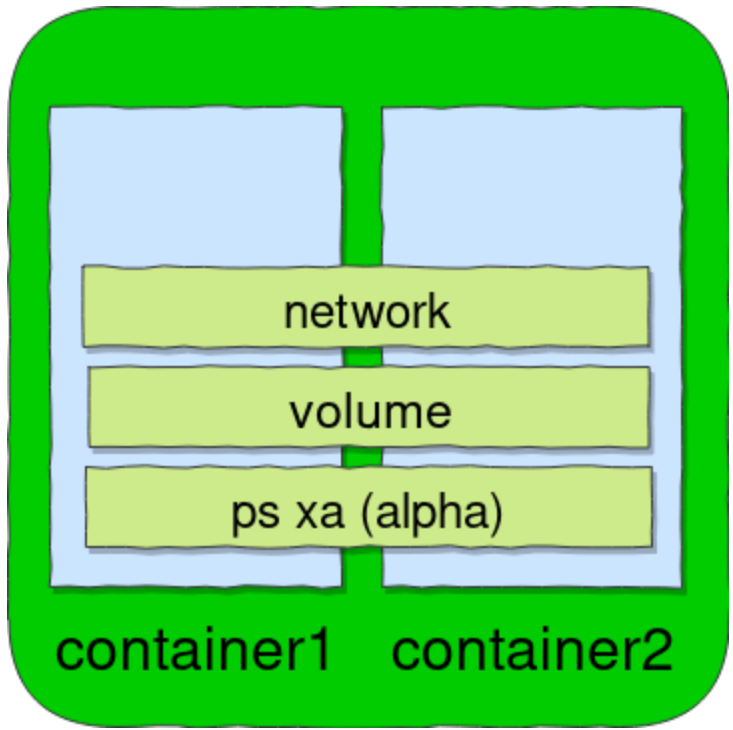

Namenode
http
Prometheus
Datanode


node1
node2
node3
node4
node5



Pod

Sidecar pattern



```yaml
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: ozone-hdfs-namenode
spec:
  serviceName: ozone2-hdfs-namenode
  replicas: 1
  template:
    metadata:
      labels:
        app: ozone
      annotations:
        prometheus.io/scrape: "true"
        prometheus.io/port: "28942"
    spec:
      shareProcessNamespace: true
      containers:
        - name: hdfs-namenode
          image: flokkr/ozone:2.1.0
          args: ["hdfs","namenode"]
        - name: jmxpromo
          image: flokkr/jmxpromo-sidecar
```

Prometheus Alerts Graph Status Help

Enable query history

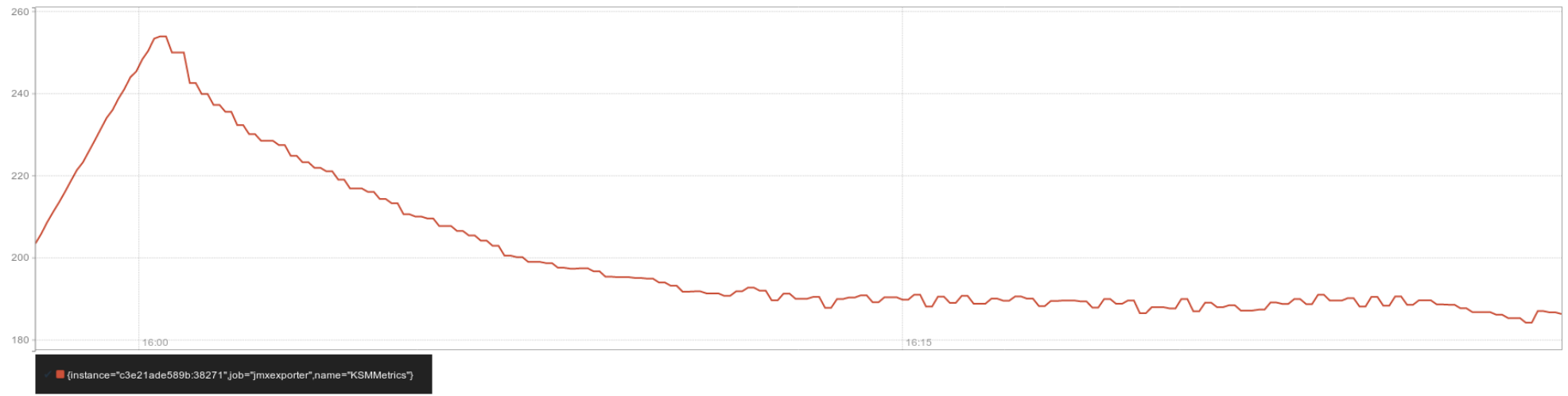
rate(Hadoop_KeySpaceManager_NumKeyCommits[10m])

Load time: 187ms
Resolution: 7s
Total time series:

Execute java_lang_OperatingSyste

Graph Console

30m Until Res. (s) stacked

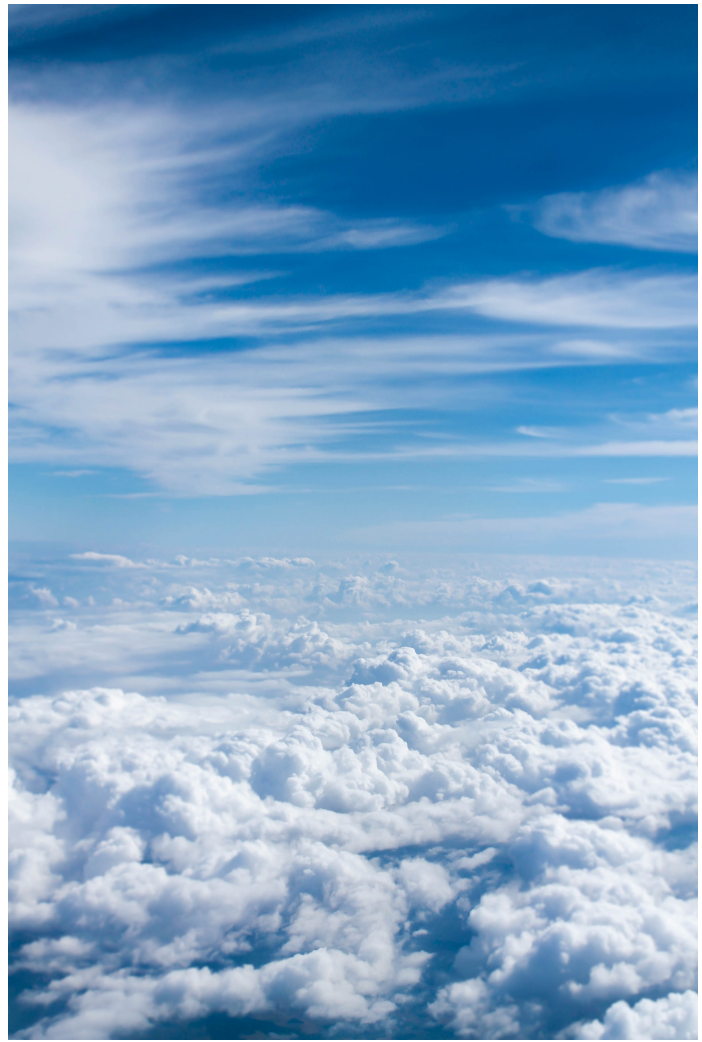


Remove Grap

Add Graph

Is Hadoop cloud native?

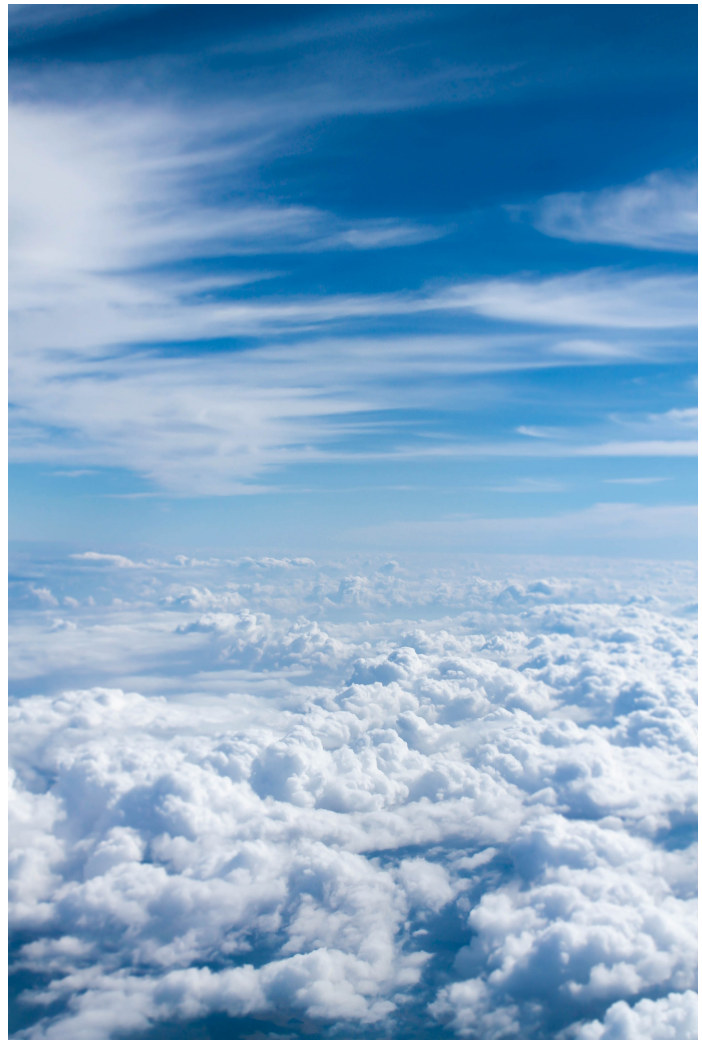
- DNS/IP handling. Shoud work
  - without DNS
  - with changing DNS
- More flexible configuration loading
- Reverse proxy friendly UI

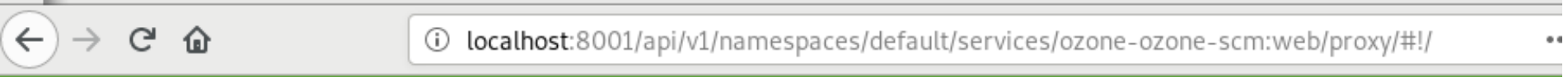


Is Hadoop cloud native?

Yes, with small modifications

- DNS/IP handling. Shoud work
  - without DNS
  - with changing DNS
- More flexible configuration loading
- Reverse proxy friendly UI





HDFS SCM Metrics Configuration Documentation Common tools

Overview

Started:	Jun 3, 2018 10:53:29 PM
Version:	3.2.0-SNAPSHOT, r9c4cbed8d19ec0f486af454de6b117d77a0a0b84
Compiled:	2018-06-03T18:41Z by jenkins from (HEAD detached at 9c4cbed)

JVM parameters

JVM:	Java HotSpot(TM) 64-Bit Server VM 25.161-b12
Input arguments:	["-Dproc_scm","-Djava.net.preferIPv4Stack=true","-Dhadoop.log.dir=/opt/hadoop/logs","-Dhadoop.l Dhadoop.root.logger=INFO,console","-Dhadoop.policy.file=hadoop-policy.xml","-Dhadoop.security.lo

Node counts

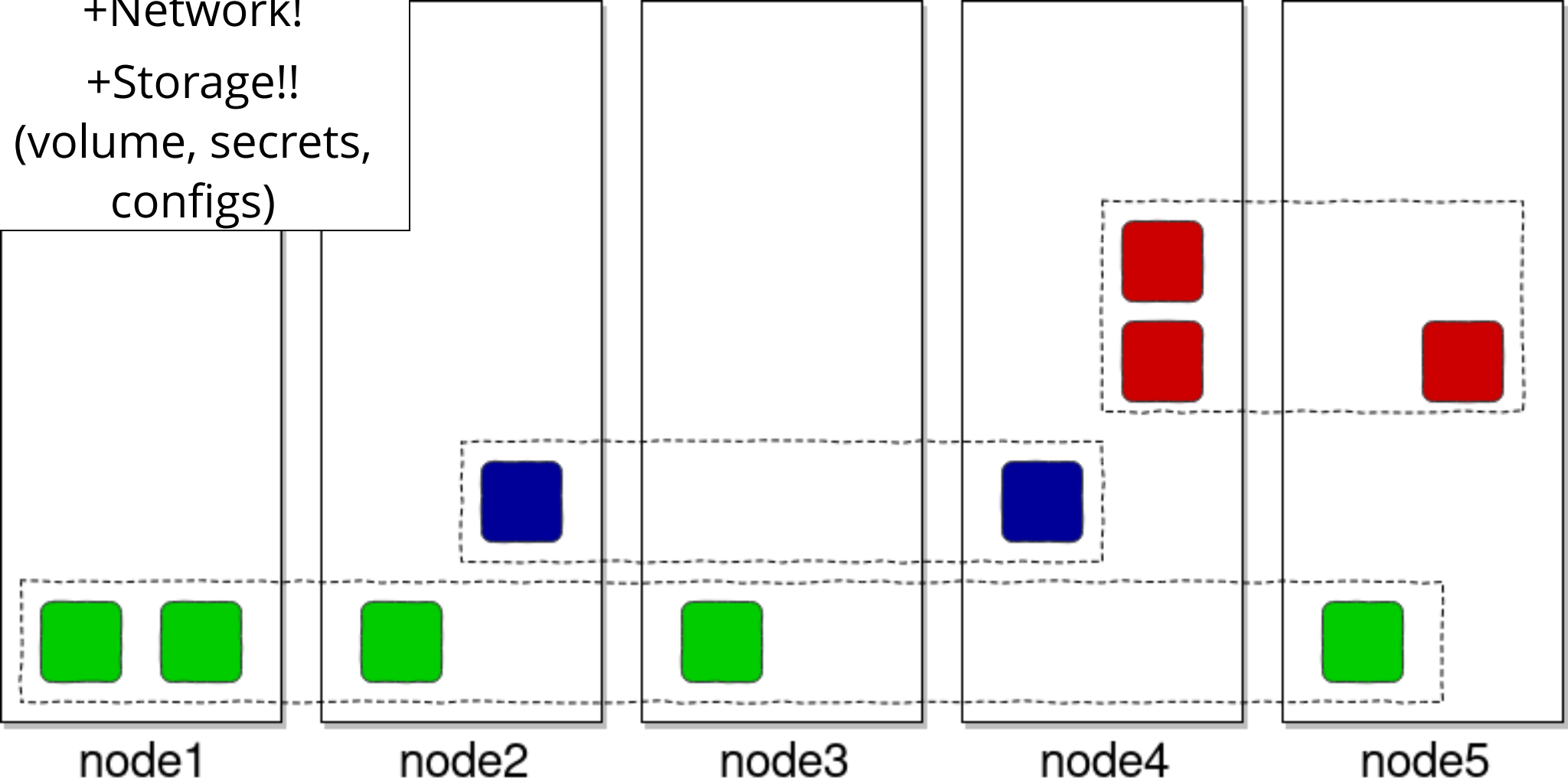
HEALTHY

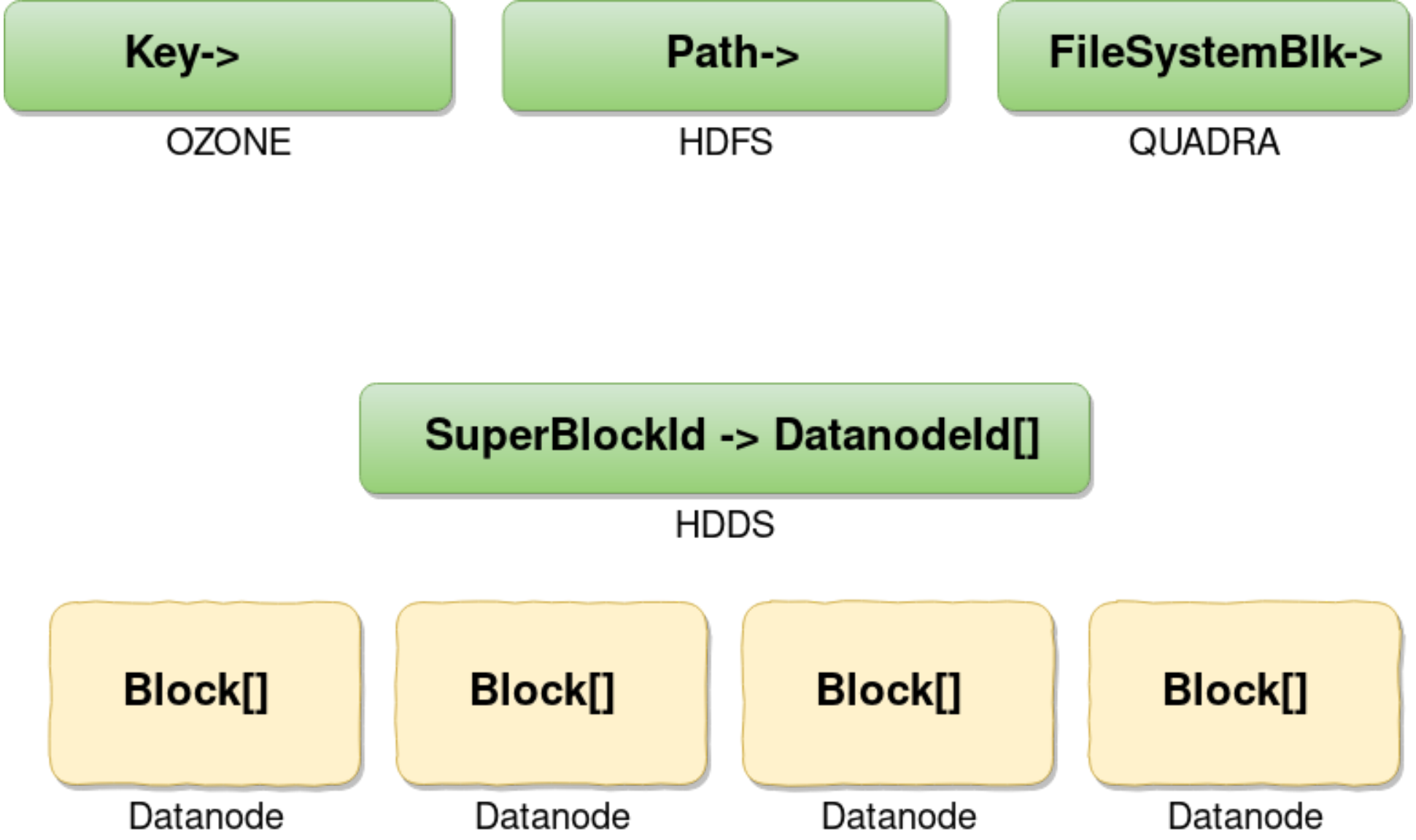
Hadoop  Kubernetes

Kubernetes  Hadoop

+Network!

+Storage!!
(volume, secrets, configs)




Key->
Path->
FileSystemBlk->
OZONE
HDFS
QUADRA
SuperBlockId -> DatanodeId[]
HDDS
Block[]
Block[]
Block[]
Block[]
Datanode
Datanode
Datanode
Datanode

Volume Plugin	Internal Provisioner	Config Example
AWSElasticBlockStore	✓	AWS
AzureFile	✓	Azure File
AzureDisk	✓	Azure Disk
CephFS	-	-
Cinder	✓	OpenStack Cinder
FC	-	-
FlexVolume	-	-
Flocker	✓	-
GCEPersistentDisk	✓	GCE
Glusterfs	✓	Glusterfs
iSCSI	-	-
Quobyte	✓	Quobyte
NFS	-	-
RBD	✓	Ceph RBD
VsphereVolume	✓	vSphere
PortworxVolume	✓	Portworx Volume
ScaleIO	✓	ScaleIO
StorageOS	✓	StorageOS
Local	-	Local

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: persistent
  labels:
    app: persistent
spec:
  replicas: 1
  selector:
    matchLabels:
      app: persistent
  template:
    metadata:
      labels:
        app: persistent
    spec:
      containers:
      - name: persistent
        image: alpine
        command: ["cat"]
        tty: true
        stding: true
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: persistent
  labels:
    app: persistent
spec:
  replicas: 1
  selector:
    matchLabels:
      app: persistent
  template:
    metadata:
      labels:
        app: persistent
    spec:
      containers:
      - name: persistent
        image: alpine
        command: ["cat"]
        tty: true
        stding: true
        volumeMounts:
        - mountPath: "/data"
          name: iscsi
      volumes:
      - name: iscsi
        persistentVolumeClaim:
           claimName: persistent
```

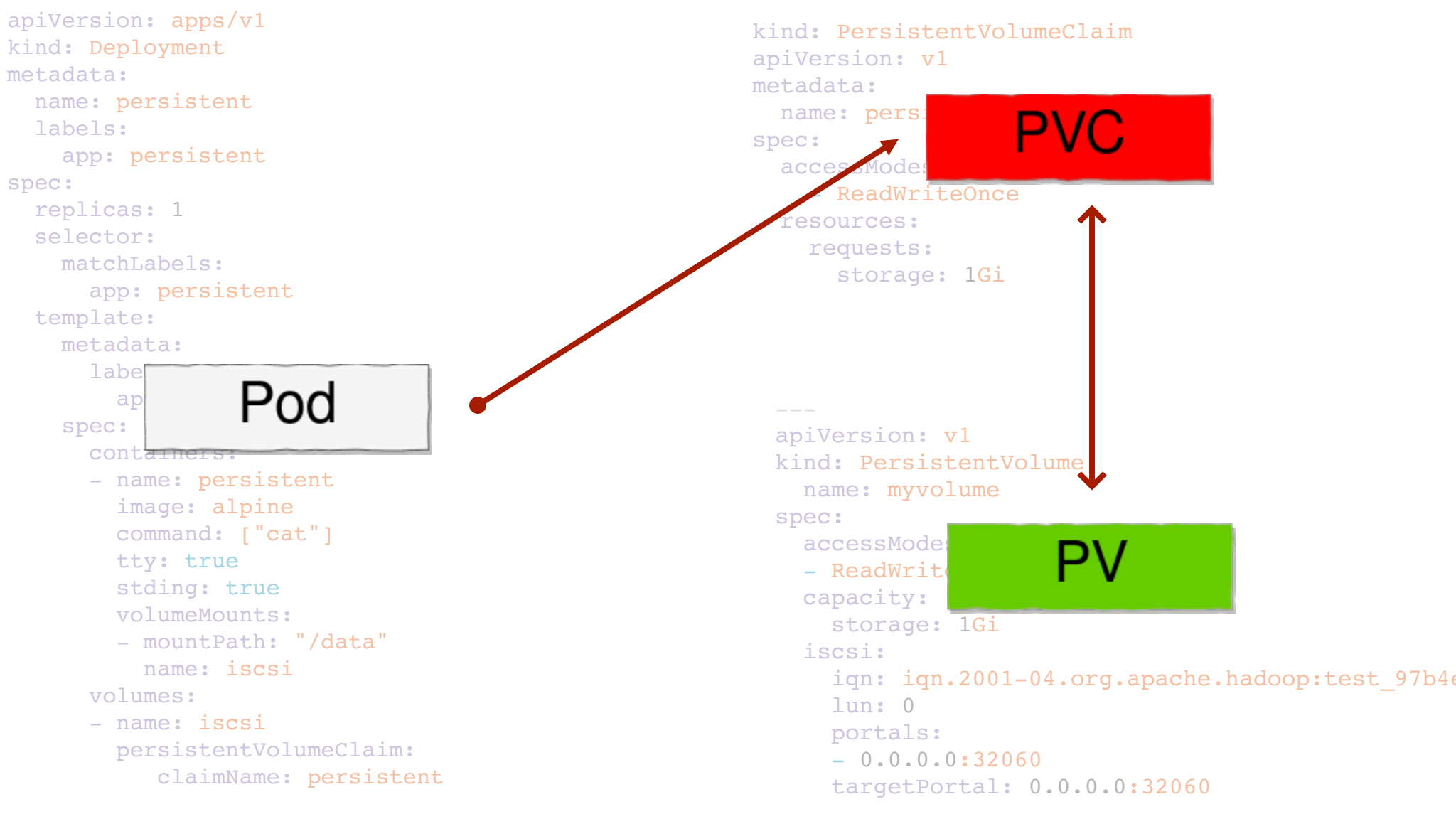
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: persistent
  labels:
    app: persistent
spec:
  replicas: 1
  selector:
    matchLabels:
      app: persistent
  template:
    metadata:
      labels:
        app: persistent
    spec:
      containers:
      - name: persistent
        image: alpine
        command: ["cat"]
        tty: true
        stding: true
        volumeMounts:
        - mountPath: "/data"
          name: iscsi
      volumes:
      - name: iscsi
        persistentVolumeClaim:
           claimName: persistent
```

```yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: persistent
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

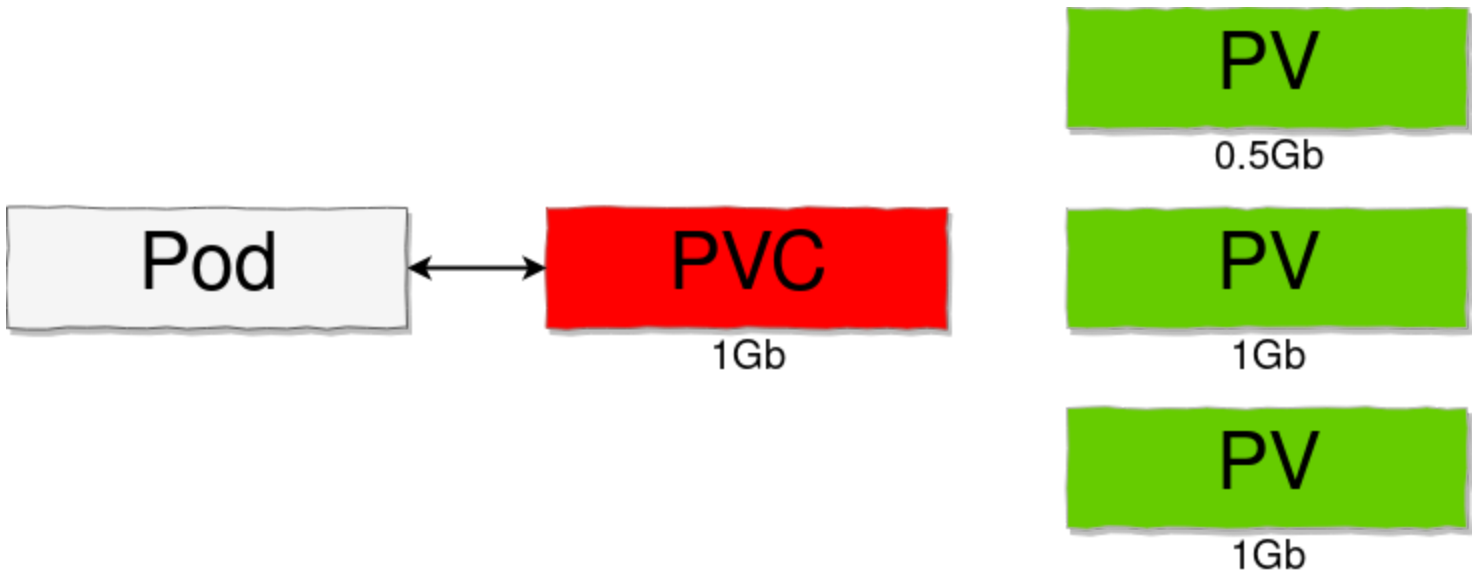

apiVersion: apps/v1
kind: Deployment
metadata:
name: persistent
labels:
app: persistent
spec:
replicas: 1
selector:
matchLabels:
app: persistent
template:
metadata:
labels:
app: persistent
spec:
containers:
- name: persistent
image: alpine
command: ["cat"]
tty: true
stding: true
volumeMounts:
- mountPath: "/data"
name: iscsi
volumes:
- name: iscsi
persistentVolumeClaim:
claimName: persistent
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: persistent
spec:
accessModes:
- ReadWriteOnce
resources:
requests:
storage: 1Gi


apiVersion: apps/v1
kind: Deployment
metadata:
name: persistent
labels:
app: persistent
spec:
replicas: 1
selector:
matchLabels:
app: persistent
template:
metadata:
labels:
app: persistent
spec:
containers:
- name: persistent
image: alpine
command: ["cat"]
tty: true
stding: true
volumeMounts:
- mountPath: "/data"
name: iscsi
volumes:
- name: iscsi
persistentVolumeClaim:
claimName: persistent
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: persistent
spec:
accessModes:
- ReadWriteOnce
resources:
requests:
storage: 1Gi
---
apiVersion: v1
kind: PersistentVolume
name: myvolume
spec:
accessModes:
- ReadWriteOnce
capacity:
storage: 1Gi
iscsi:
iqn: iqn.2001-04.org.apache.hadoop:test_97b4e
lun: 0
portals:
- 0.0.0.0:32060
targetPortal: 0.0.0.0:32060

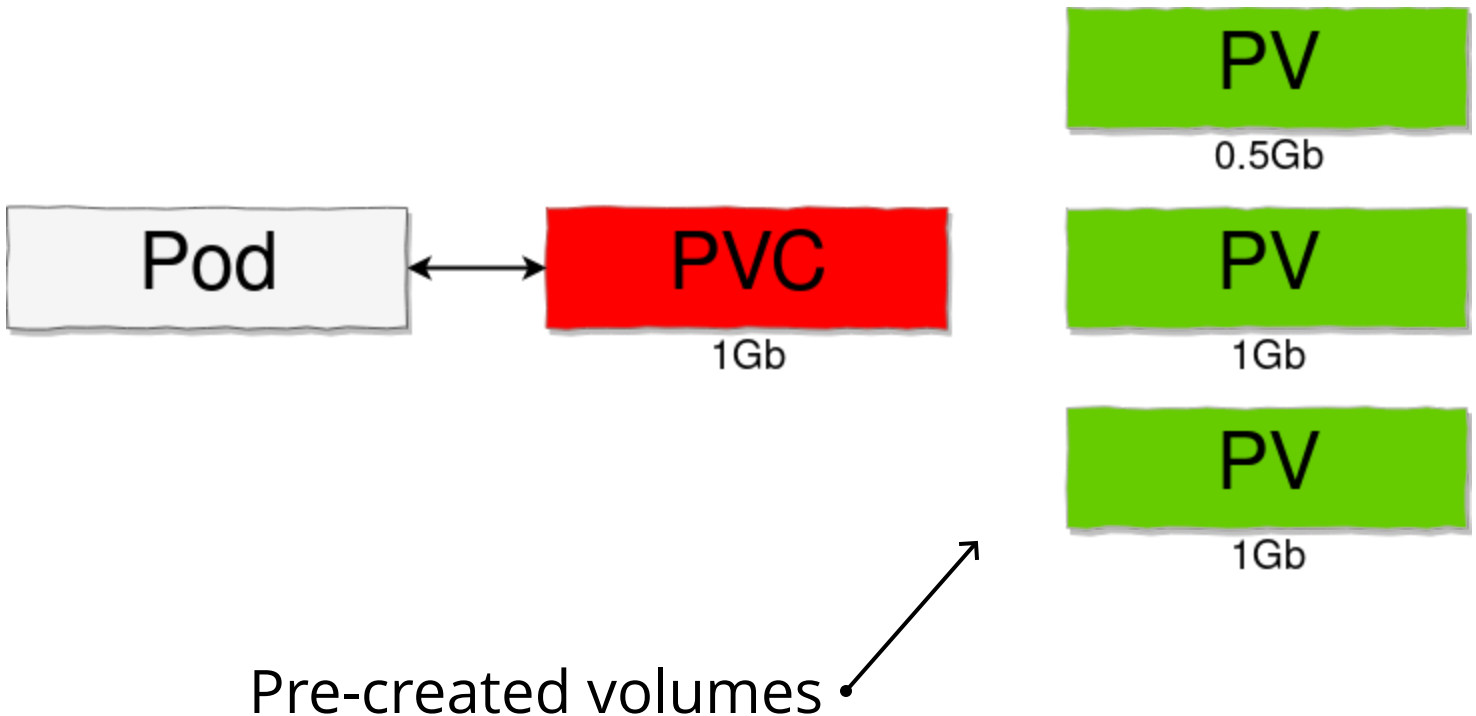

apiVersion: apps/v1
kind: Deployment
metadata:
name: persistent
labels:
app: persistent
spec:
replicas: 1
selector:
matchLabels:
app: persistent
template:
metadata:
labels:
app: persistent
spec:
containers:
- name: persistent
image: alpine
command: ["cat"]
tty: true
stding: true
volumeMounts:
- mountPath: "/data"
name: iscsi
volumes:
- name: iscsi
persistentVolumeClaim:
claimName: persistent
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: persistent
spec:
accessModes:
- ReadWriteOnce
resources:
requests:
storage: 1Gi
---
apiVersion: v1
kind: PersistentVolume
name: myvolume
spec:
accessModes:
- ReadWriteOnce
capacity:
storage: 1Gi
iscsi:
iqn: iqn.2001-04.org.apache.hadoop:test_97b4e
lun: 0
portals:
- 0.0.0.0:32060
targetPortal: 0.0.0.0:32060


apiVersion: apps/v1
kind: Deployment
metadata:
name: persistent
labels:
app: persistent
spec:
replicas: 1
selector:
matchLabels:
app: persistent
template:
metadata:
labe
ap
spec:
containers:
- name: persistent
image: alpine
command: ["cat"]
tty: true
stding: true
volumeMounts:
- mountPath: "/data"
name: iscsi
volumes:
- name: iscsi
persistentVolumeClaim:
claimName: persistent
Pod
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: pers
spec:
accessMode
- ReadWriteOnce
resources:
requests:
storage: 1Gi
PVC
---
apiVersion: v1
kind: PersistentVolume
name: myvolume
spec:
accessMode
- ReadWrit
capacity:
storage: 1Gi
iscsi:
iqn: iqn.2001-04.org.apache.hadoop:test_97b4e
lun: 0
portals:
- 0.0.0.0:32060
targetPortal: 0.0.0.0:32060
PV

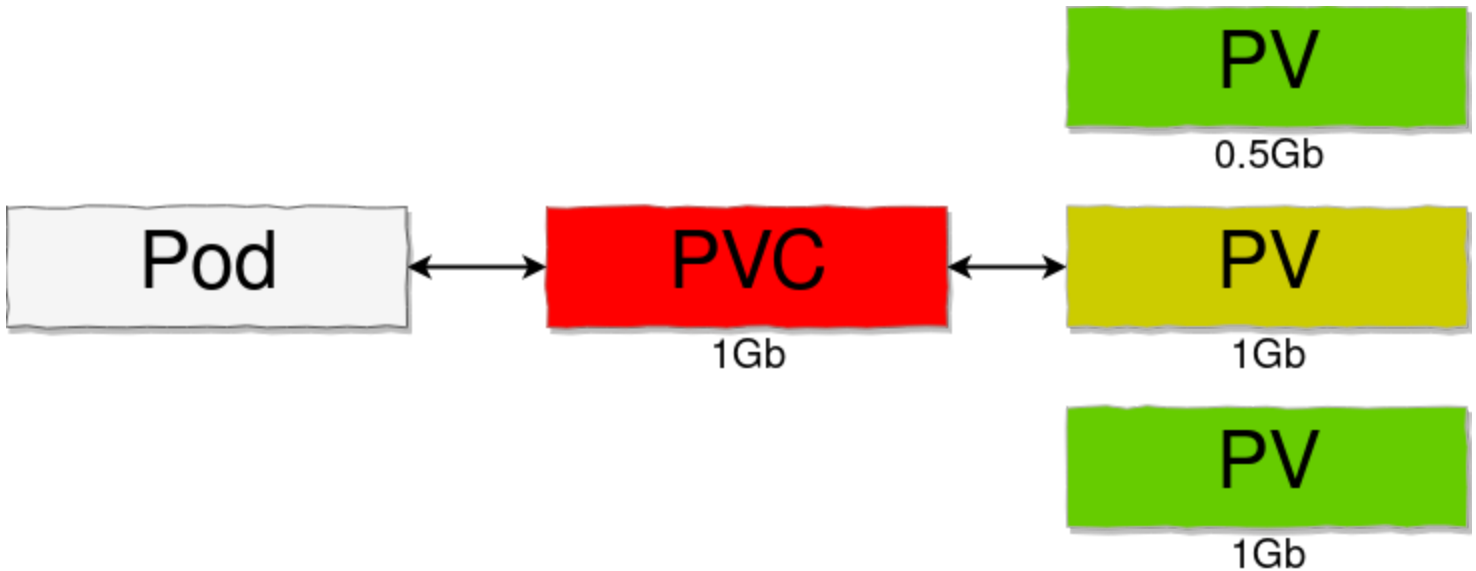
Static Provisinioning



Static Provisinioning

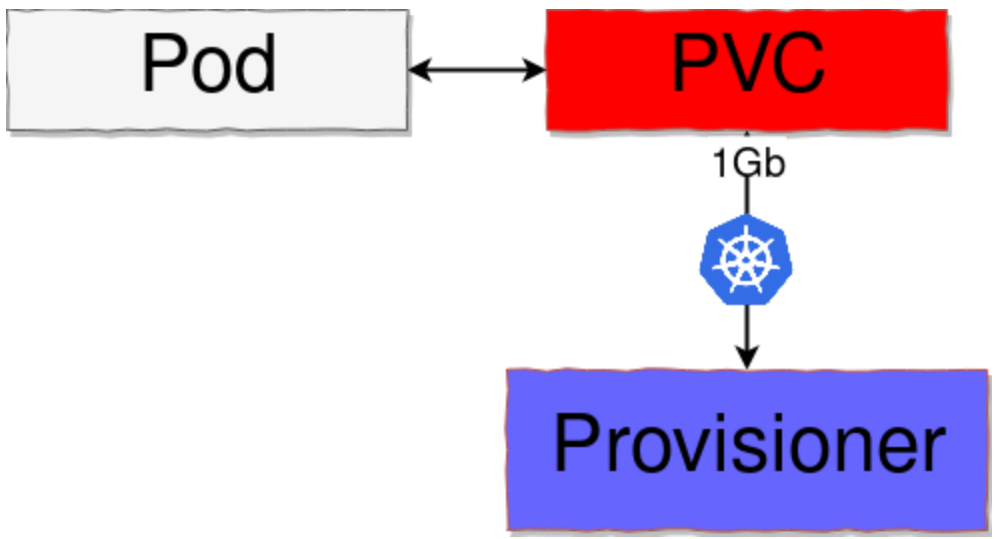


Static Provisinioning

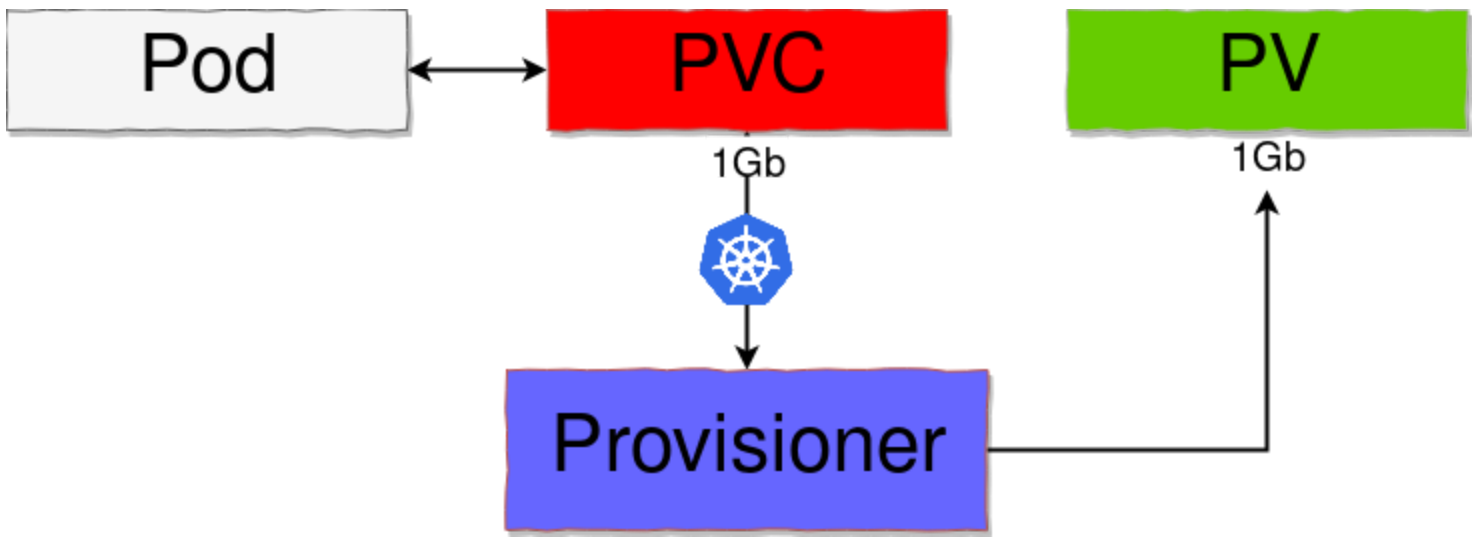


Dynamic Provisinioning

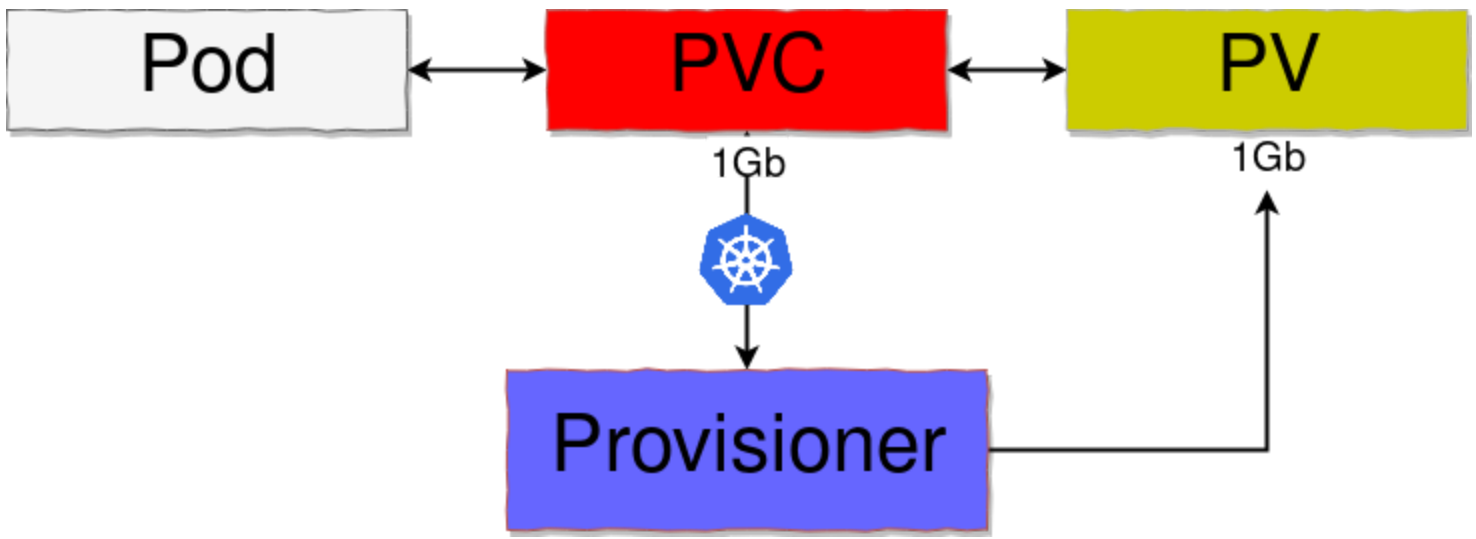
Dynamic Provisinioning

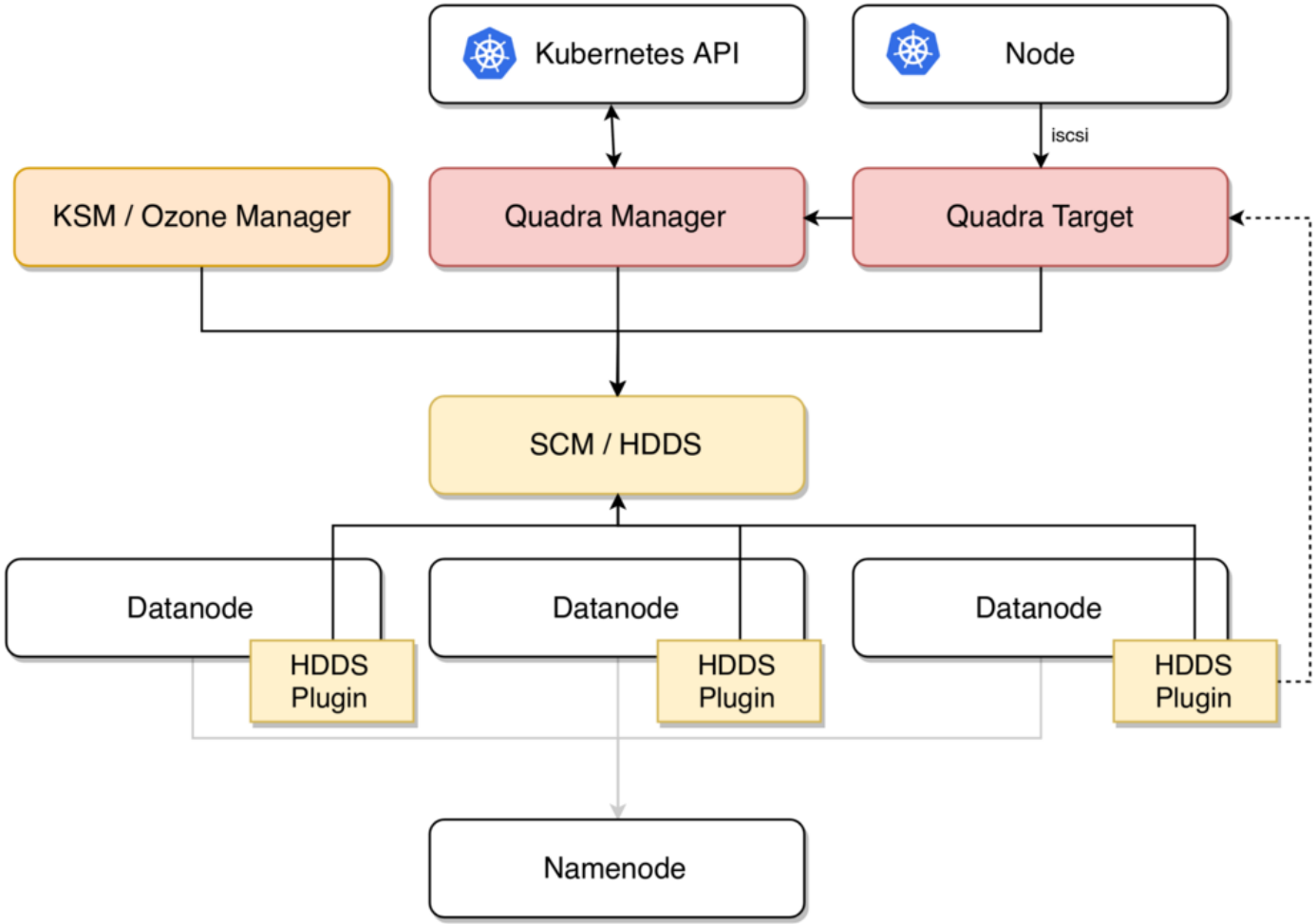


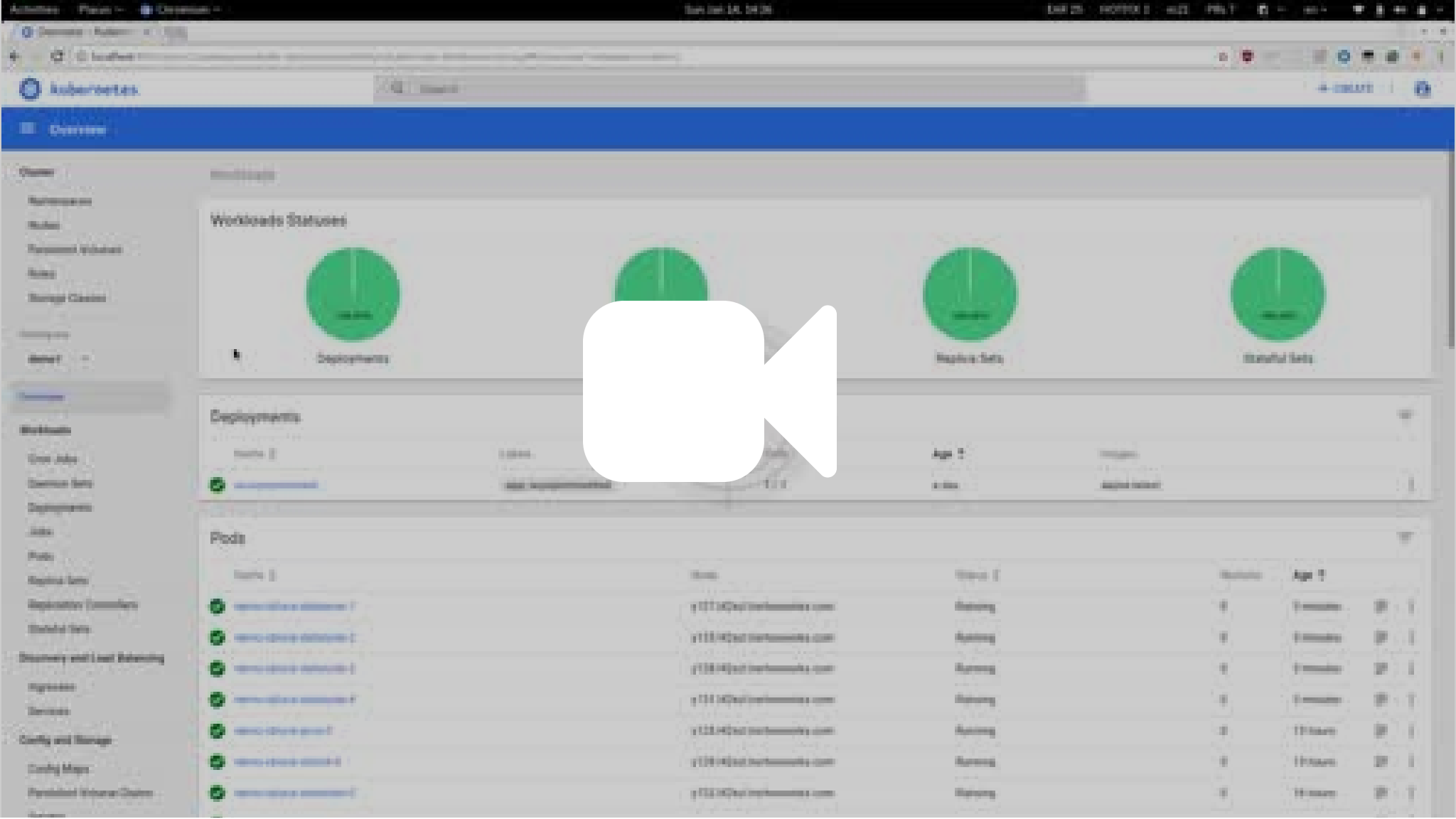
Dynamic Provisinioning



Dynamic Provisinioning




Kubernetes API
Node
iscsi
KSM / Ozone Manager
Quadra Manager
Quadra Target
SCM / HDDS
Datanode
Datanode
Datanode
HDDS Plugin
HDDS Plugin
HDDS Plugin
Namenode


kubernetes
Overview
Workloads Statuses
Deployments
Replica Sets
Stateful Sets
Deployments
Pods

Hadoop  Kubernetes

Kubernetes  Hadoop

Hadoop  Kubernetes

Kubernetes  Hadoop

Hadoop  Kubernetes

Kubernetes  Hadoop

Containerization helps a lot.

Hadoop  Kubernetes

Kubernetes  Hadoop

Containerization helps a lot.

Hadoop works
well in cloud-native
environments.

Hadoop  Kubernetes

Kubernetes  Hadoop

Containerization helps a lot.

Hadoop works
well in cloud-native
environments.

Could work even better with
minor improvements

Hadoop ❤ Kubernetes

Containerization helps a lot.

Hadoop works
well in cloud-native
environments.

Could work even better with
minor improvements

Kubernetes ❤ Hadoop

Hadoop Ozone/HDDS is

All in One:

- HDFS file system
- Object store filesystem
- Raw storage

Q&A

Márton Elek @anzix

~~https://ozone.hadoop.apache.org~~

https://flokkr.github.io (bigdata + containers project)

https://github.com/flokkr (source)

elek@apache.org