Configurazioni multi-site di CEPH

Fabrizio G. Ventola

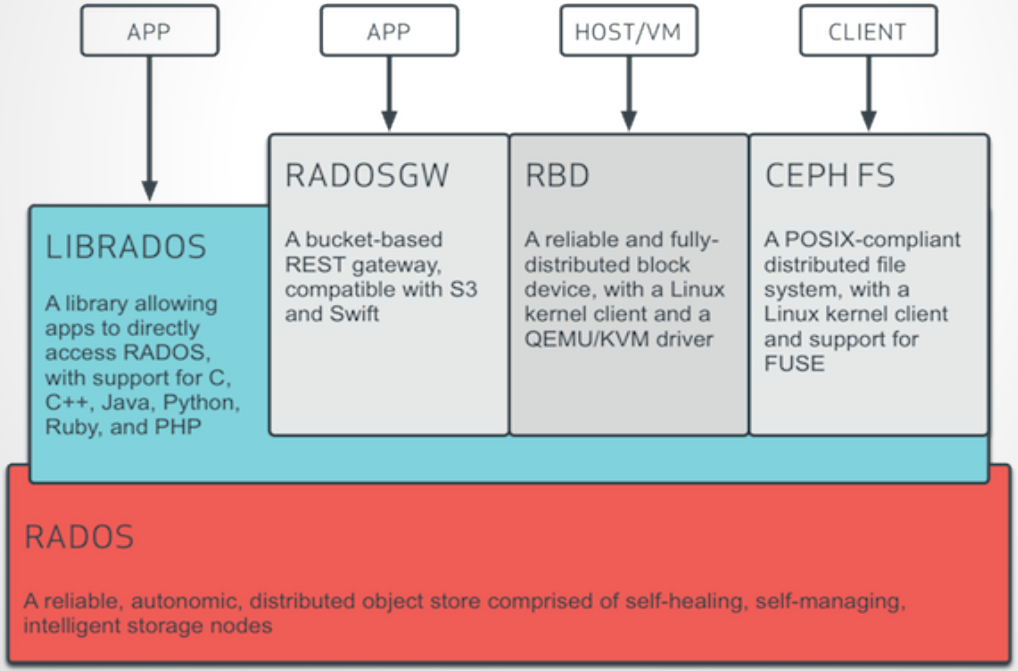
Tutorial days CCR - Napoli 19/12/2014

Outline

- intro Ceph
- Ceph consistency
- Ceph geo-replication models
- consistent geo-replication
- CRUSH map
- federated GWs
- REST geo-replication
- tests

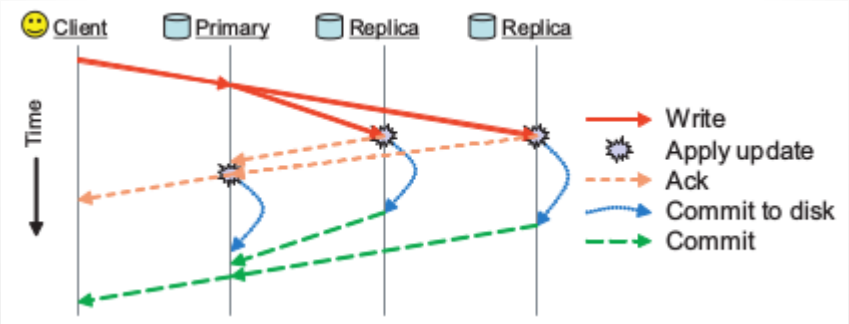
Ceph

Ceph **uniquely delivers object, block, and file storage in one unified system.**



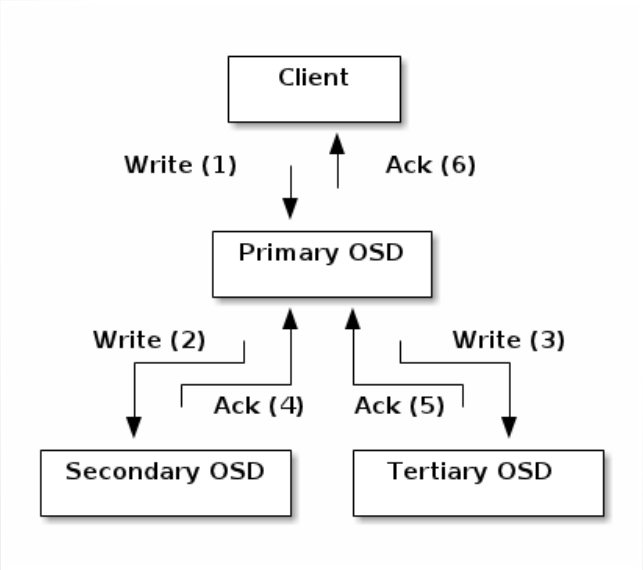
Ceph - consistency

Ceph implements strong consistency between OSDs...



Ceph - consistency

...according to replication factor



Ceph - eventual consistency

When?

- you don’t need strong consistency
- you want replicate your data geographically
- you don’t have a low latency (generally performant) link

How?

- Using Ceph RADOS Gateway (REST)

Ceph - eventual consistency

- Pay attention: reads **eventually** return the same value… remember **CAP theorem**:

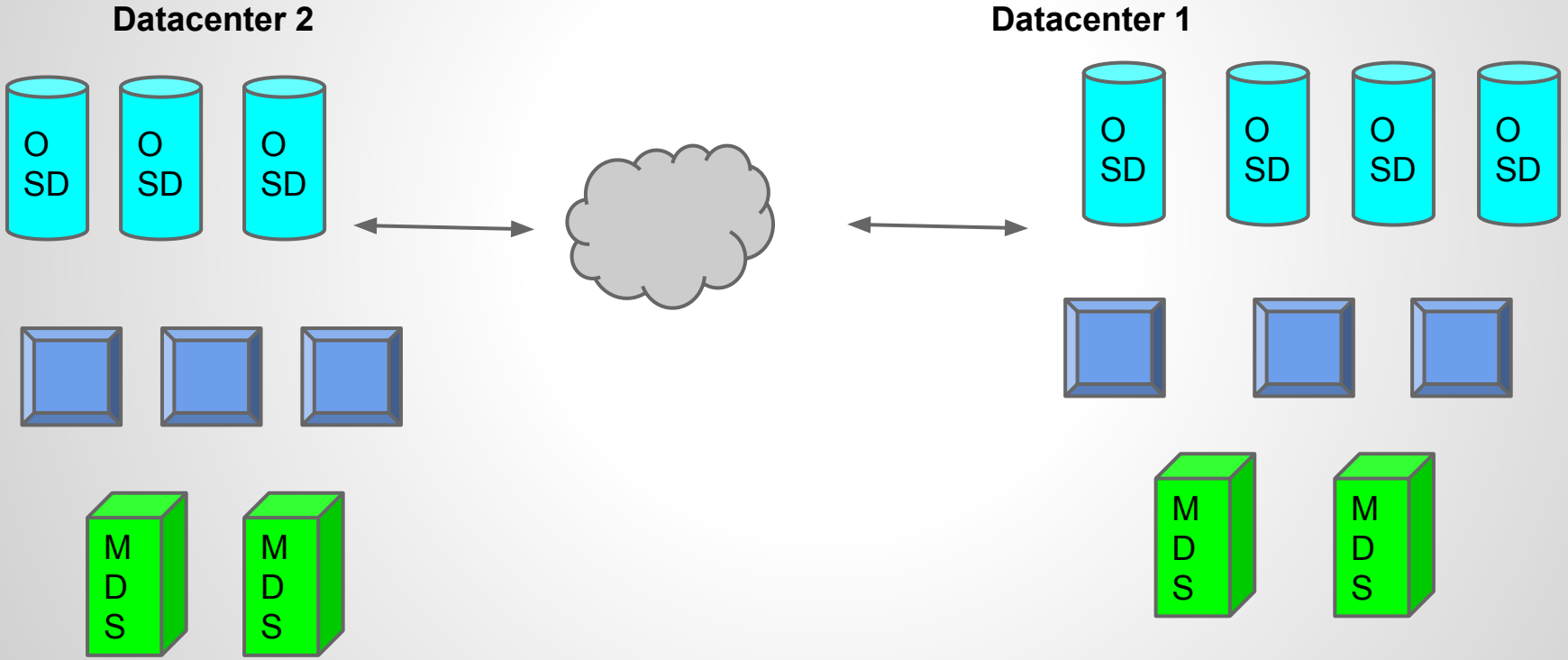
«...it’s impossible for a distributed system to simultaneously provide Consistency, Availability, Partition tolerance...»

- Relax consistency (affects performance on unstable link)

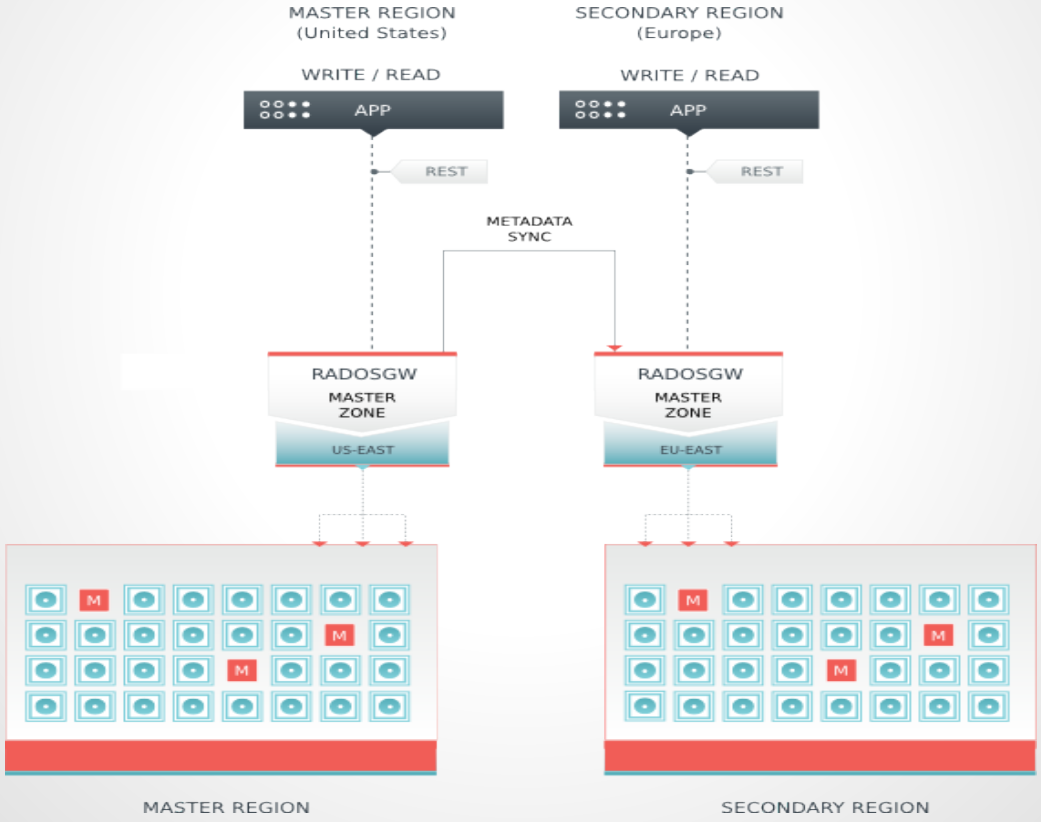
Ceph - eventual consistency

...but that's OK for **disaster recovery** (cold migration and similar use-cases)

Ceph - geo-replication model 1



Ceph - geo-replication model 2



Ceph - strong geo-replication

- You need performant link between involved sites (low latency)
- Deploy usually it’s not so easy (port 22 closed)
- Exploit CRUSH map and placement rules to define **extra datacenter** failure domains and placement strategies

CRUSH map

- CRUSH: dynamic HASH function
- Goals
  - avoid allocation tables
  - avoid data imbalance
  - avoid workload imbalance
  - doesn’t reshuffle if cluster map changes

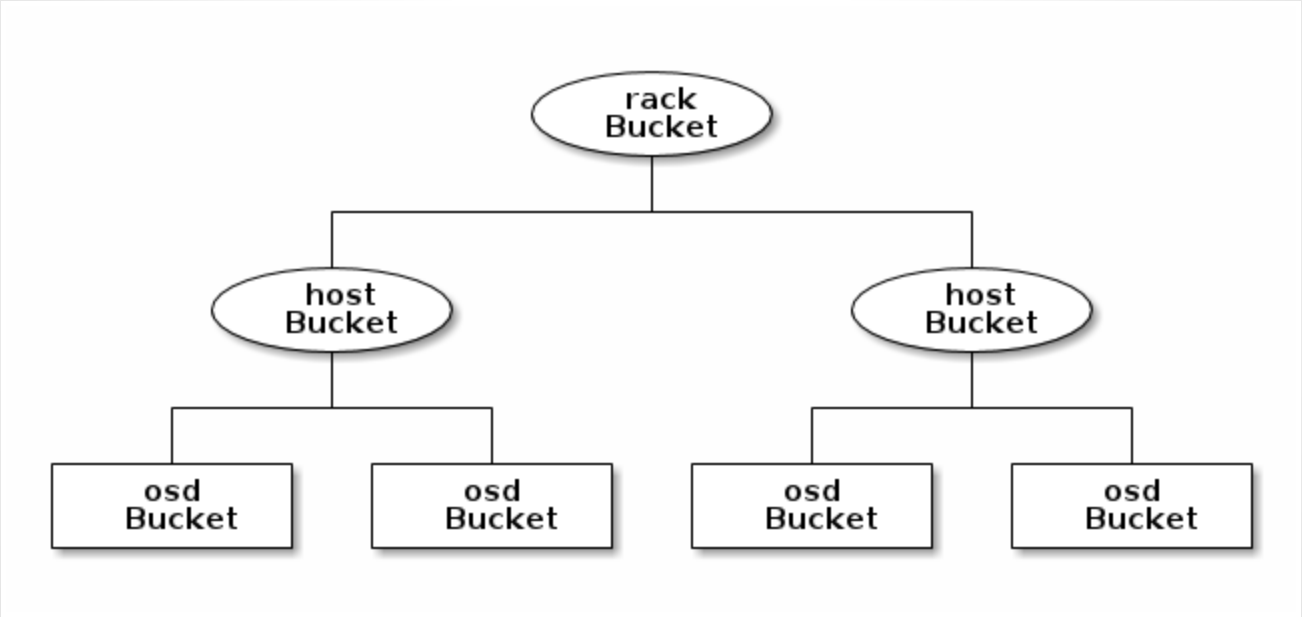
CRUSH function

- CRUSH maintains uniform distribution of workload and data minimizing data migration in case of cluster expansion or reduction
- Every node can compute CRUSH, clients too

CRUSH map

- Hierarchical cluster map
- Weighted tree where internal nodes are failure domains and leaves are OSDs eventually with different weights (size, speed, age…)

CRUSH map

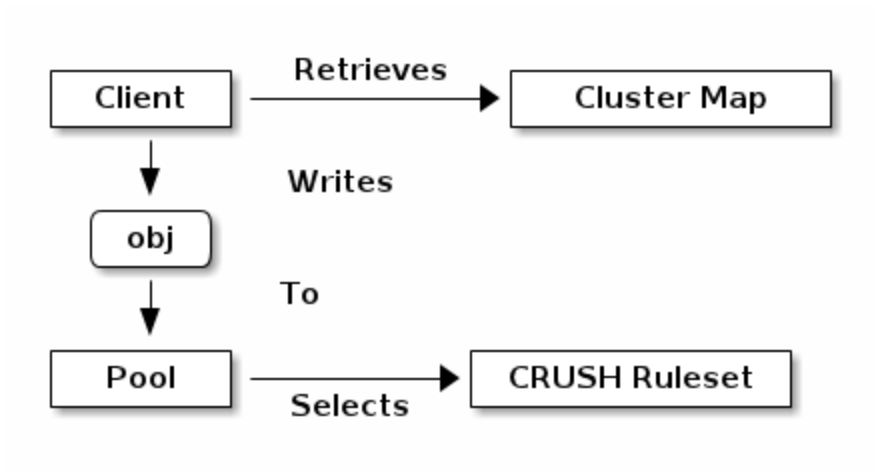

rack
Bucket
host
Bucket
host
Bucket
osd
Bucket
osd
Bucket
osd
Bucket
osd
Bucket

CRUSH map

- CRUSH map contains placement rules
- Rules define how much and how distribute replicas among the cluster map tree
- Placement rules are defined both for reliability and performance (define “fast pool”)
- We can associate any rule to a pool (but each pool must have just one rule)

CRUSH map

Ceph nodes and **clients** have an updated vision of the cluster sharing (in a lazy way) incremental updates of CRUSH map



CRUSH map

- CRUSH map sections:
  - parameters
  - devices
  - types (root, datacenter, room, row, rack, host, osd)
  - buckets
  - rules

CRUSH map

Buckets:

```
host anode-ba-infn-cloud {
        id -2           # do not change unnecessarily
        # weight 5.430
        alg straw
        hash 0  # rjenkins1
        item osd.0 weight 1.810
        item osd.1 weight 1.810
        item osd.2 weight 1.810
}
```

Buckets

```
host bnode-ba-infn-cloud {
        id -3           # do not change unnecessarily
        # weight 5.430
        alg straw
        hash 0  # rjenkins1
        item osd.3 weight 1.810
        item osd.4 weight 1.810
        item osd.5 weight 1.810
}
```

Buckets

```
host cnode-ba-infn-cloud {
        id -4           # do not change unnecessarily
        # weight 5.430
        alg straw
        hash 0  # rjenkins1
        item osd.6 weight 1.810
        item osd.7 weight 1.810
        item osd.8 weight 1.810
}
host node1-pd-infn-cloud {
... }
host node2-pd-infn-cloud {
… }
host node3-pd-infn-cloud { … }
```

Buckets

```
root bari {
        id -1           # do not change unnecessarily
        # weight 16.290
        alg straw
        hash 0  # rjenkins1
        item anode-ba-infn-cloud weight 5.430
        item bnode-ba-infn-cloud weight 5.430
        item cnode-ba-infn-cloud weight 5.430
}
```

Buckets

```
root padova {
        id -11          # do not change unnecessarily
        # weight 16.290
        alg straw
        hash 0  # rjenkins1
        item node1-pd-infn-cloud weight 5.430
        item node2-pd-infn-cloud weight 5.430
        item node3-pd-infn-cloud weight 5.430
}
```

CRUSH map ruleset

Rules:

```
rule pool_bari {
        ruleset 4
        type replicated
        min_size 0
        max_size 4
        step take bari
        step chooseleaf firstn 0 type host
        step emit
}
```

Rules

```
rule pool_padova {
        ruleset 5
        type replicated
        min_size 0
        max_size 4
        step take padova
        step chooseleaf firstn 0 type host
        step emit
}
```

Rules

```
rule pool_geo-replica {
        ruleset 6
        type replicated
        min_size 0
        max_size 4
        step take bari
        step chooseleaf firstn 2 type host
        step emit
        step take padova
        step chooseleaf firstn -2 type host
        step emit
}
```

Federated GWs

- A Ceph cluster can export a REST interface thank to RADOS Gateway module
- Main goal: exploit Ceph REST interface to replicate data among different Ceph clusters (through S3/Swift API)

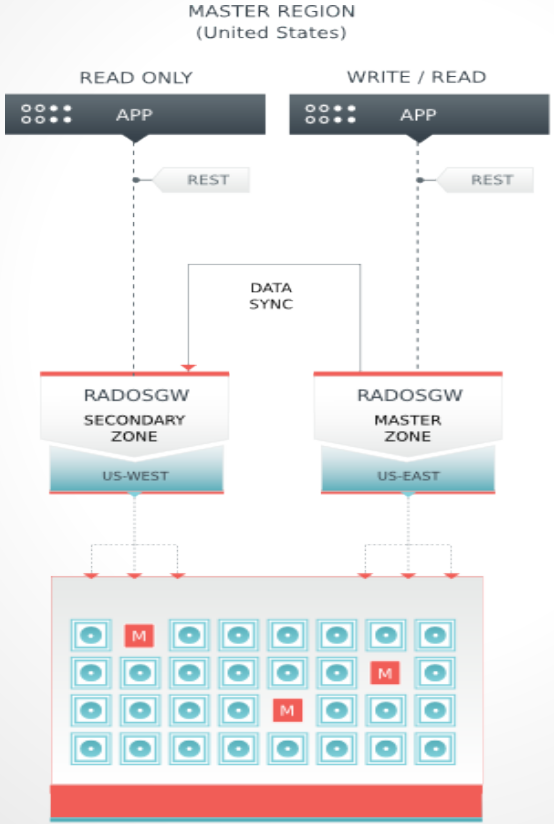
Federated GWs

- A RGW can participate in a federated architecture with multiple regions and multiple zones for a region
- A region represents a logical geographic area and contains one or more zones (only **one** region can act as a **master** in a federated cluster)
- A zone is a logical grouping of one or more RGWs (only **one** zone in a region can act as a **master**)
- RGW doesn't prevent you from writing to a secondary zone (**don't do it!**)

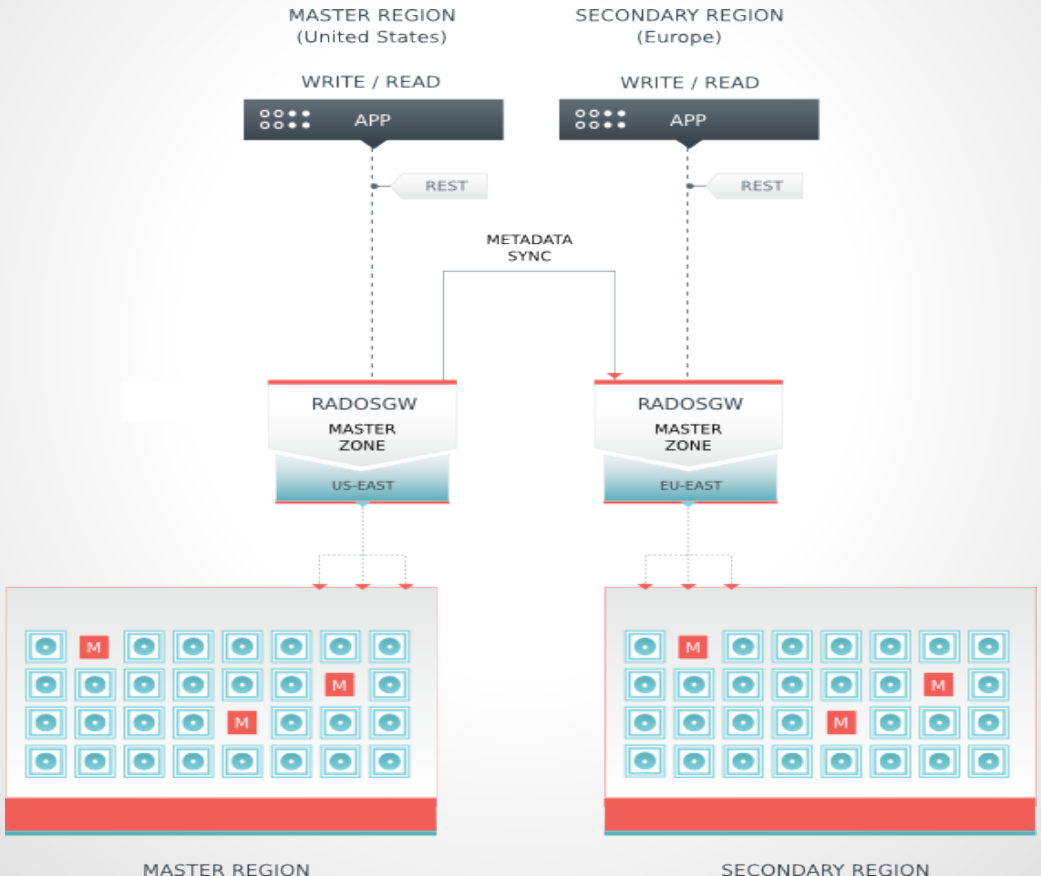
REST geo-replication

- One can run a separate Ceph cluster for each region or for each zone
- One can run multiple instance of RGW on one Ceph cluster
- All depends on your needs, degree of redundancy and isolation
- Common scenarios:
  - 1 region, 2 zones (master slave) inter-zone replication
  - 2 regions, 2 zones for each region (master/slave) and inter-region replication
  - N regions, ring replication

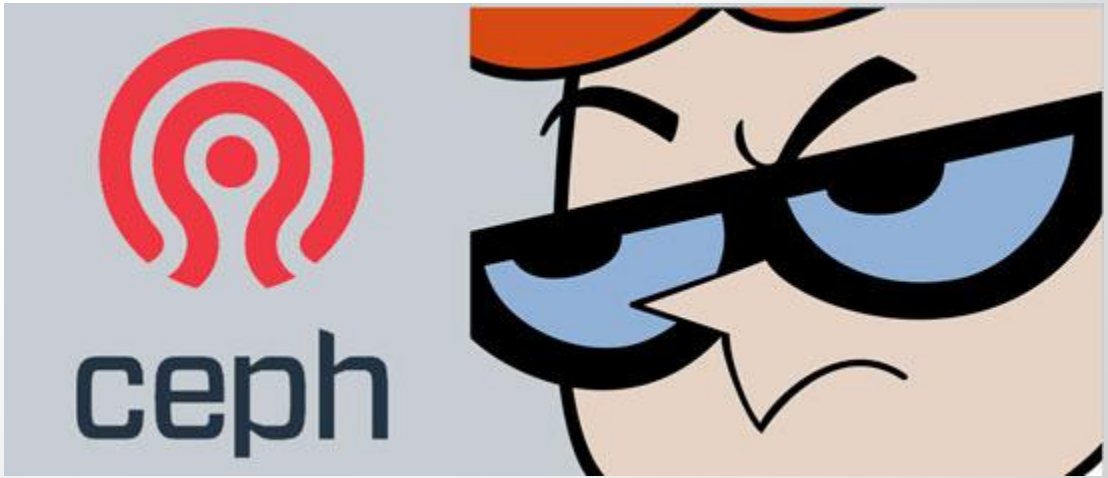
Inter-zone replication



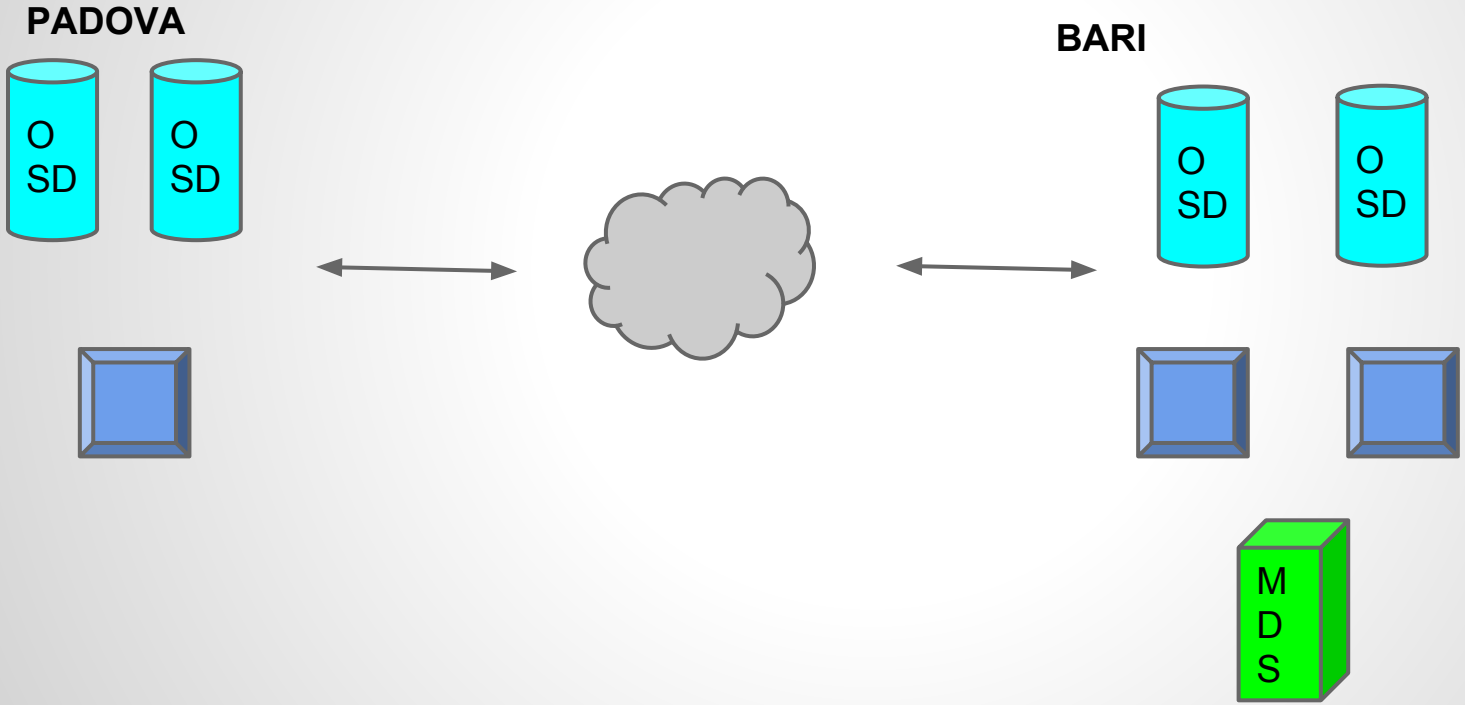
Inter-region replication



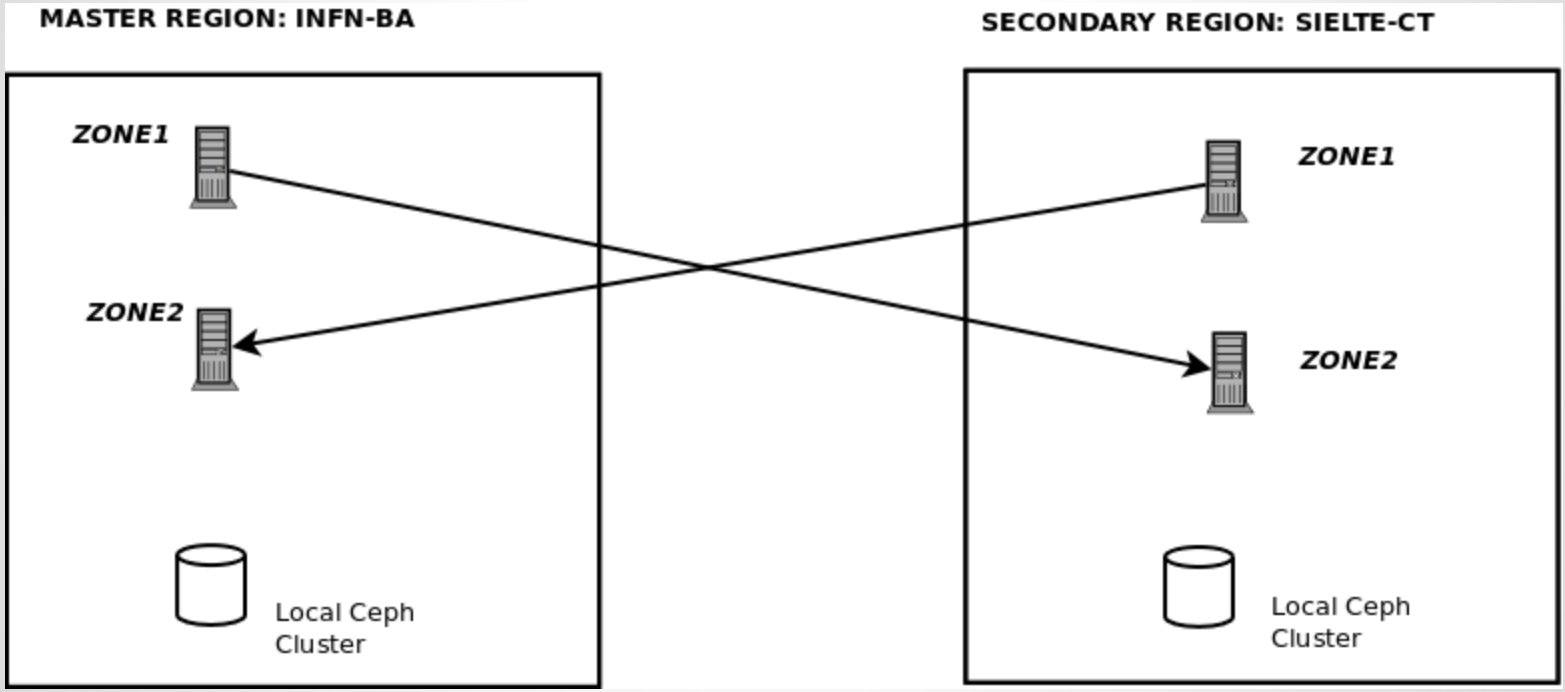
Tests



Test - consistent replica



Test REST replica



Tests

- Work in progress…preliminary results
- It’s not easy to deploy one cluster among 2 or more sites (for consistent replica)
- Network latency strongly affects Ceph RADOS protocol (both on block storage and object storage)

Configurazione multi-site di CEPH

Thank you for your attention!!!