KubeCon CloudNativeCon North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022

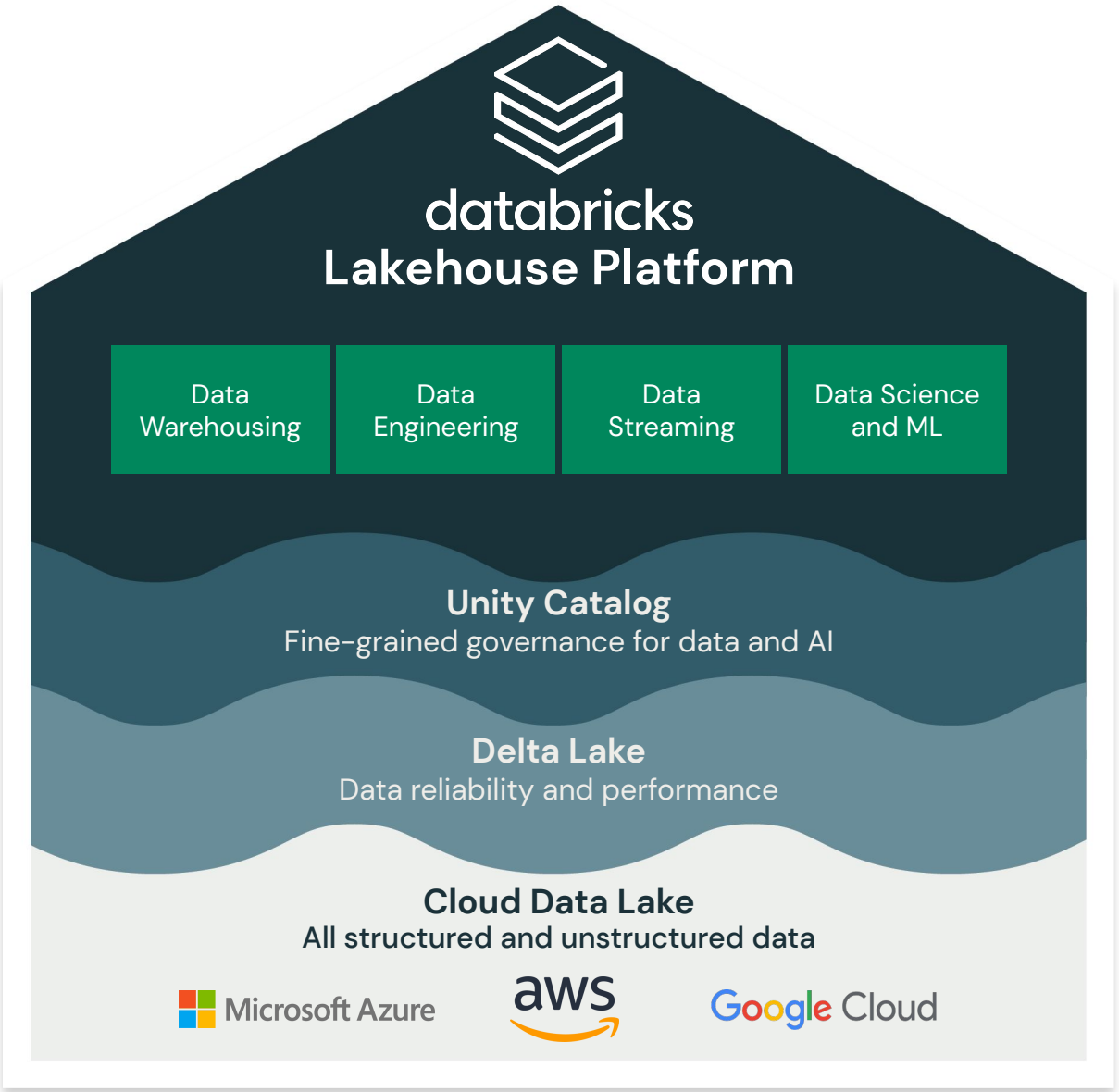
Migrating from single-node Kubernetes control plane to HA in production

Cong Yue and David Oppenheimer
Databricks

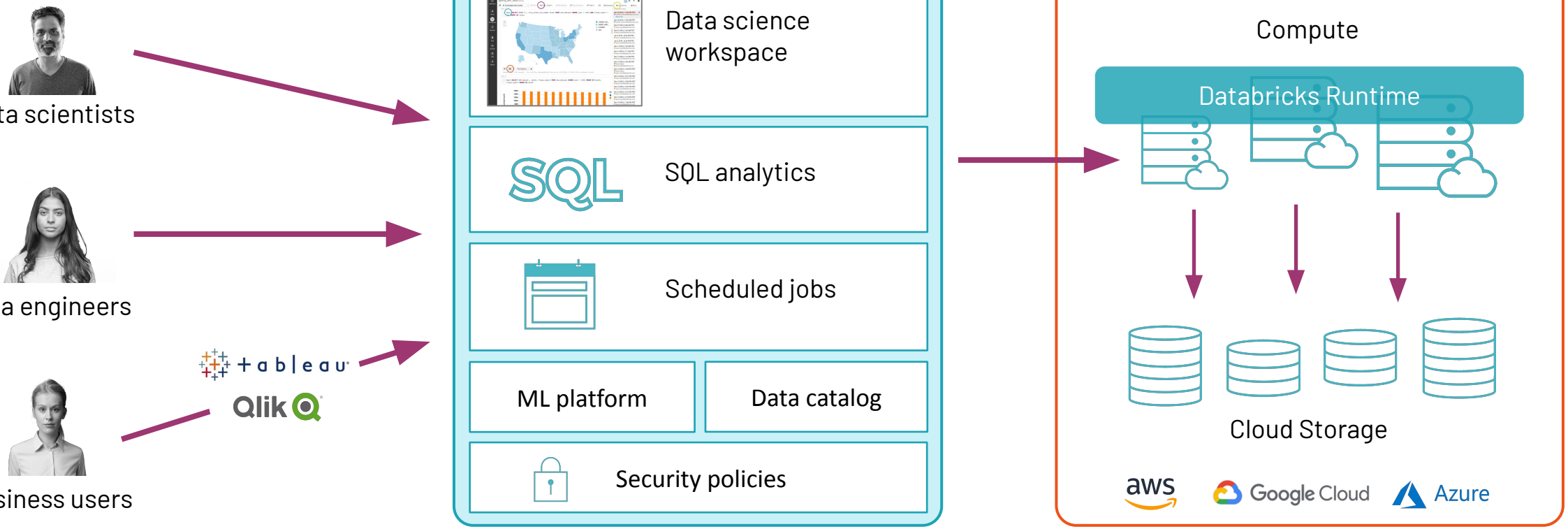
Outline

- Kubernetes at Databricks
- Non-HA control plane architecture
- HA control plane architecture & failure tolerance
- Non-HA -> HA migration process (and rollback)
- “Day 2” with HA
- Summary

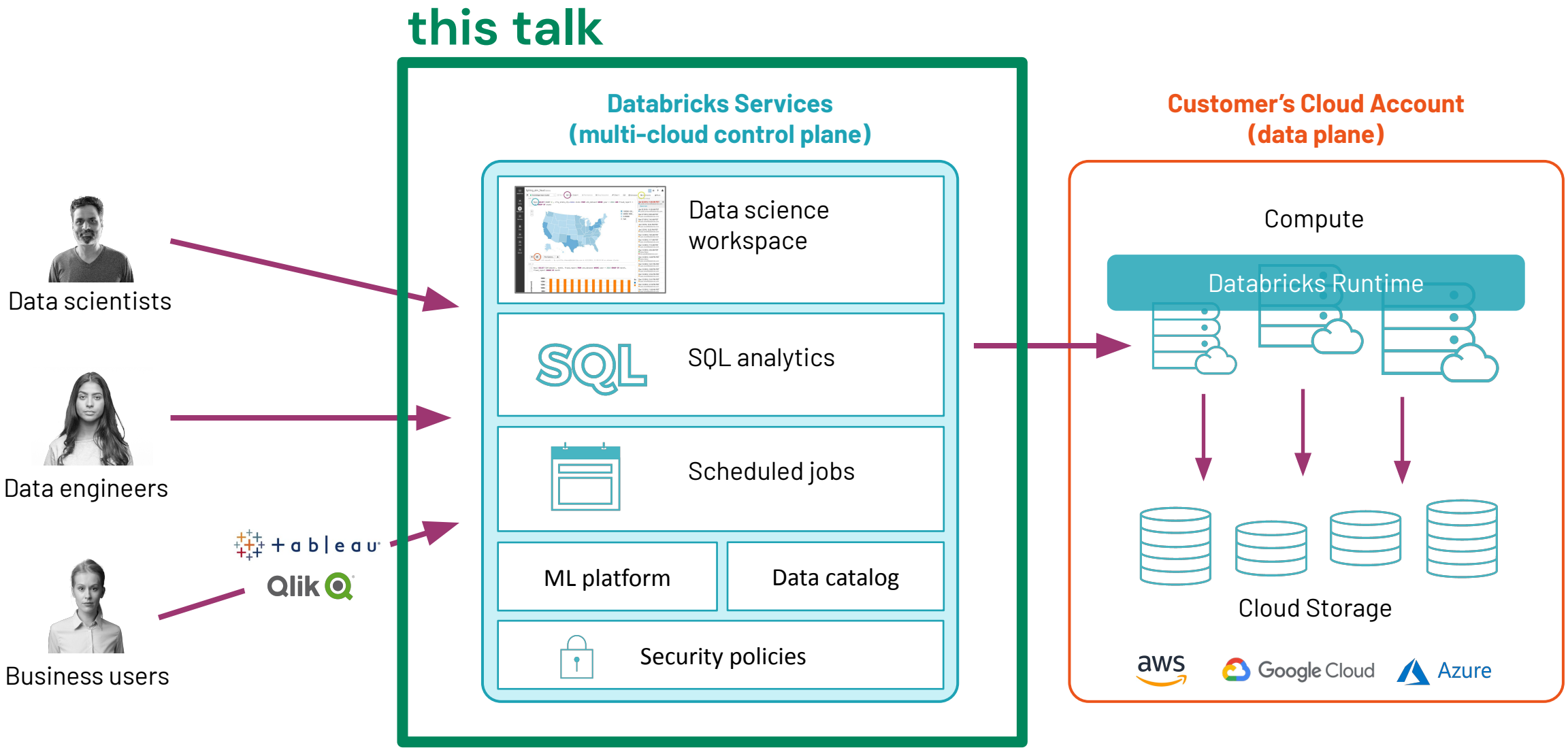
Databricks Lakehouse platform



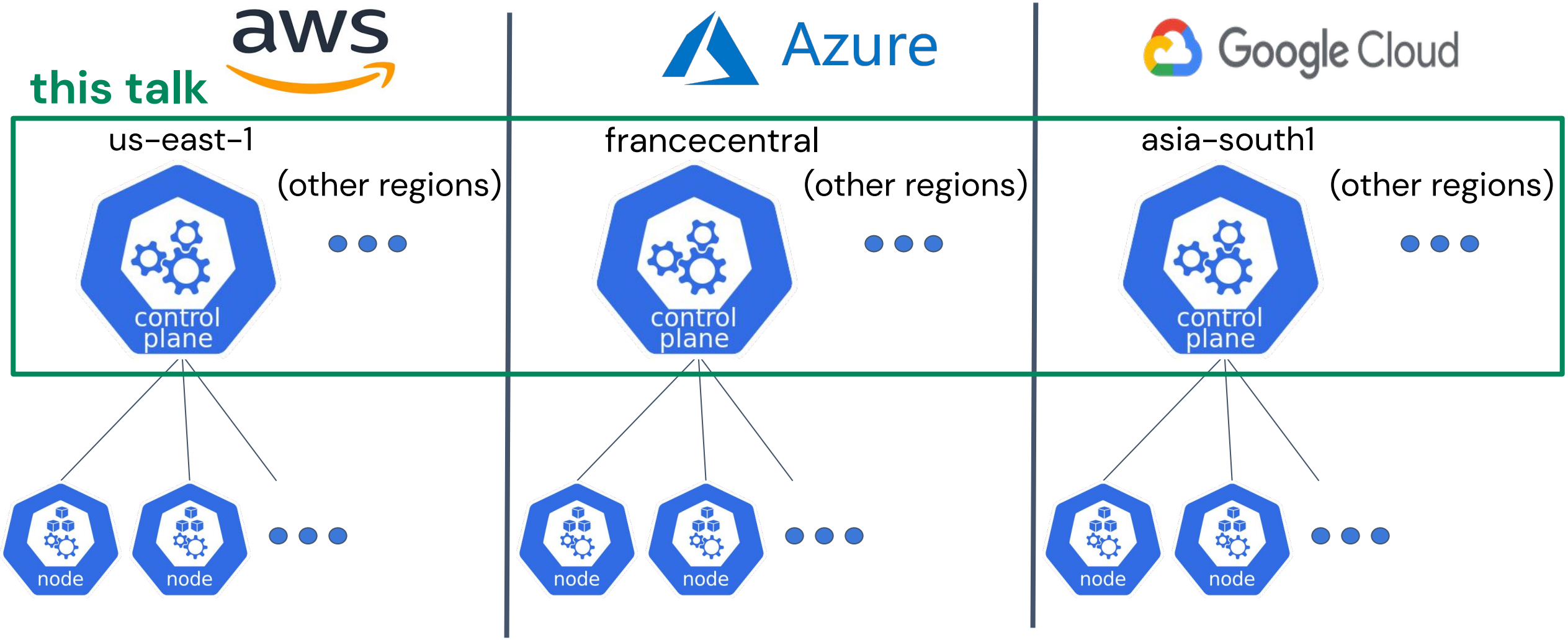
Databricks platform architecture



Databricks product architecture

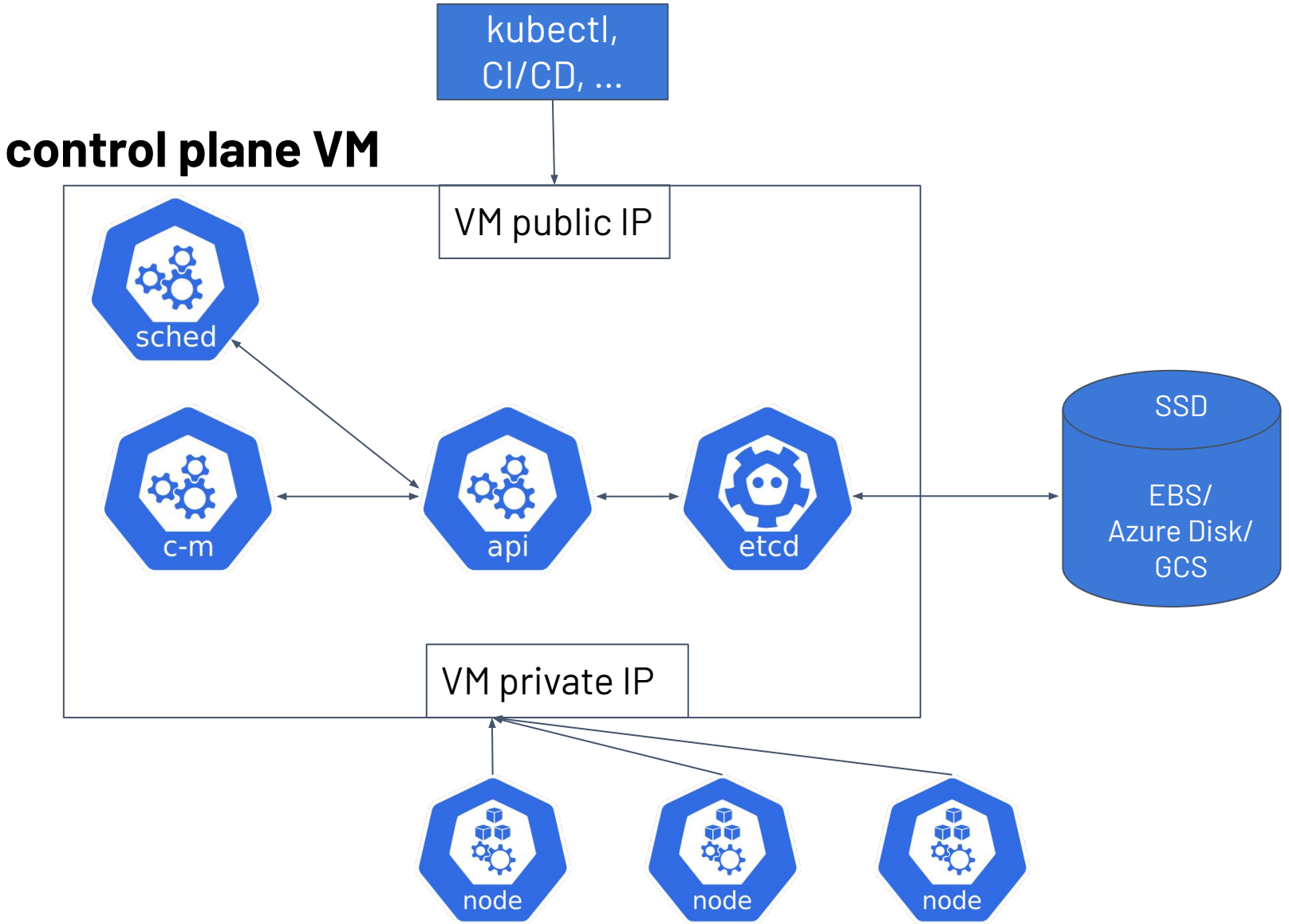


Multi-cloud control plane architecture

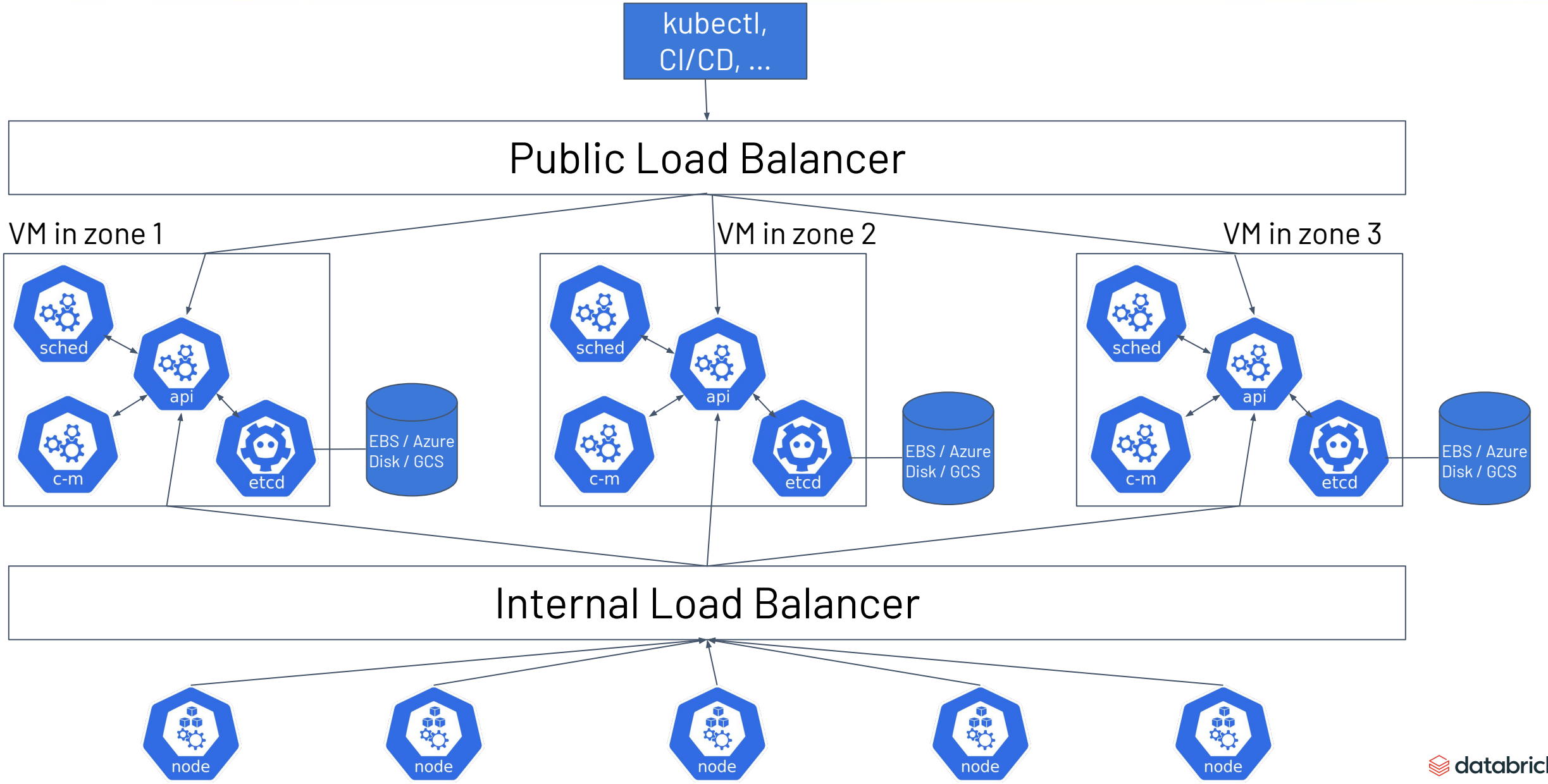


(total >60 production regions; self-managed k8s)

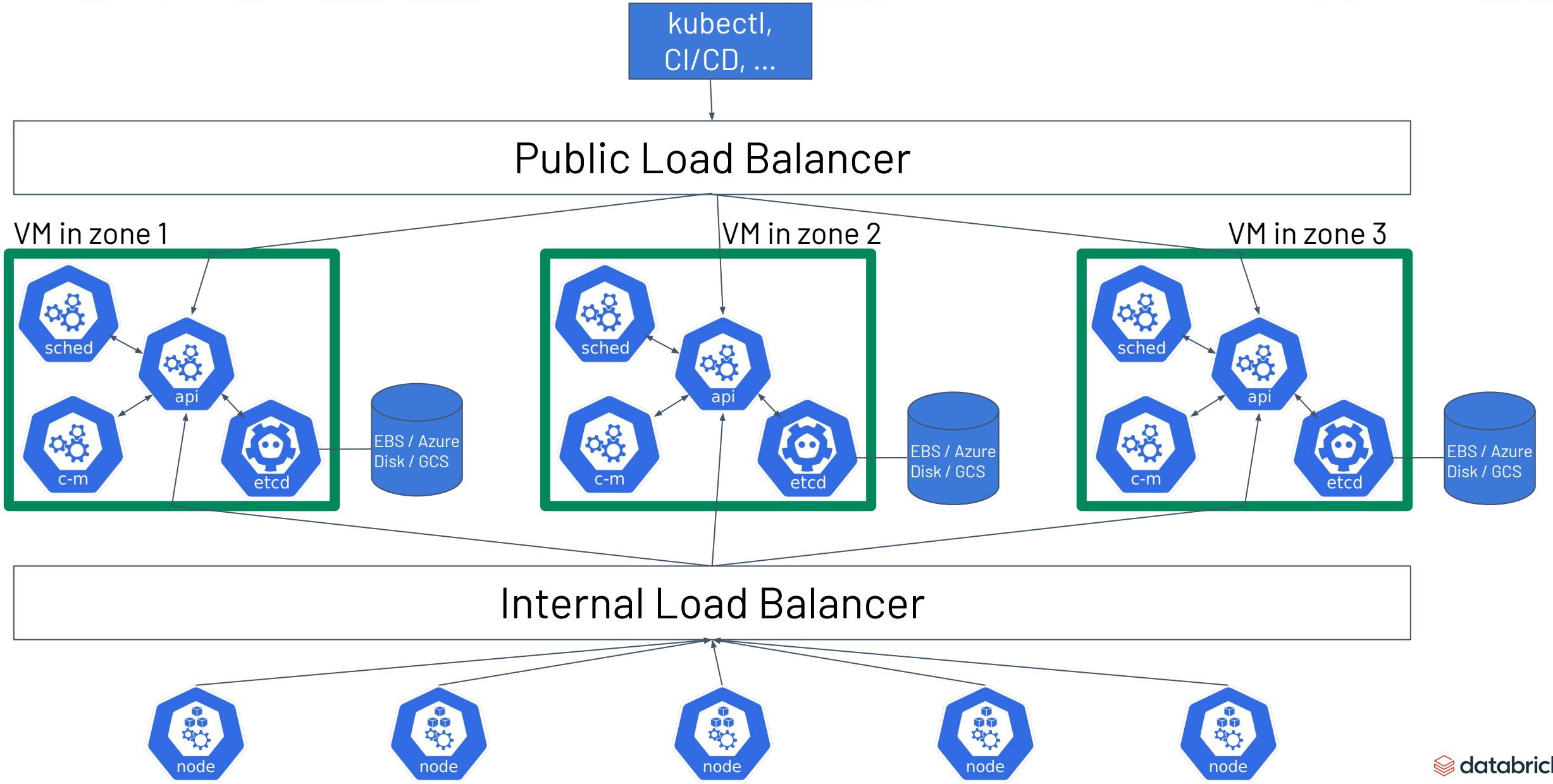
Non-HA cluster architecture



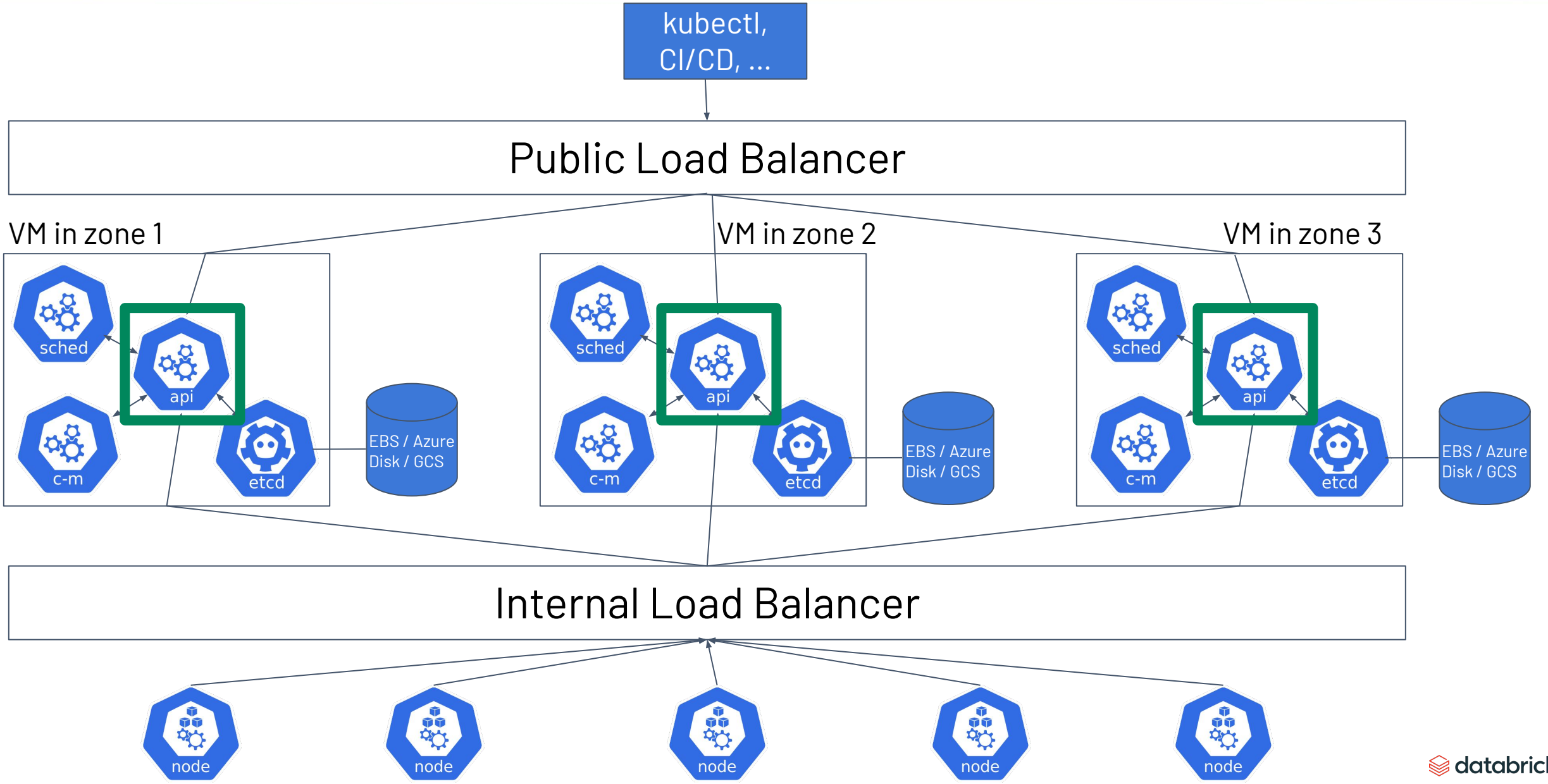
HA cluster architecture



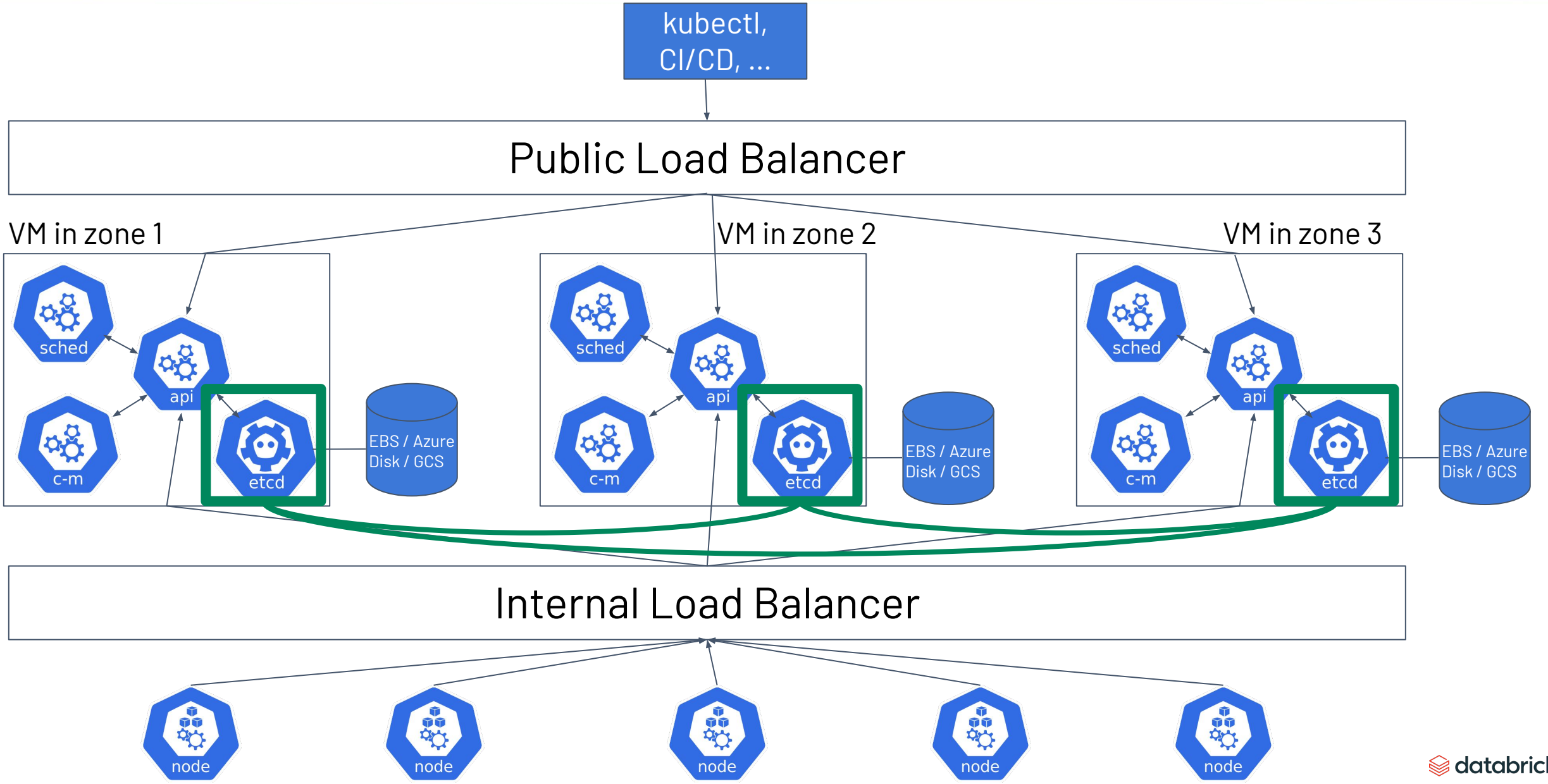
HA cluster architecture



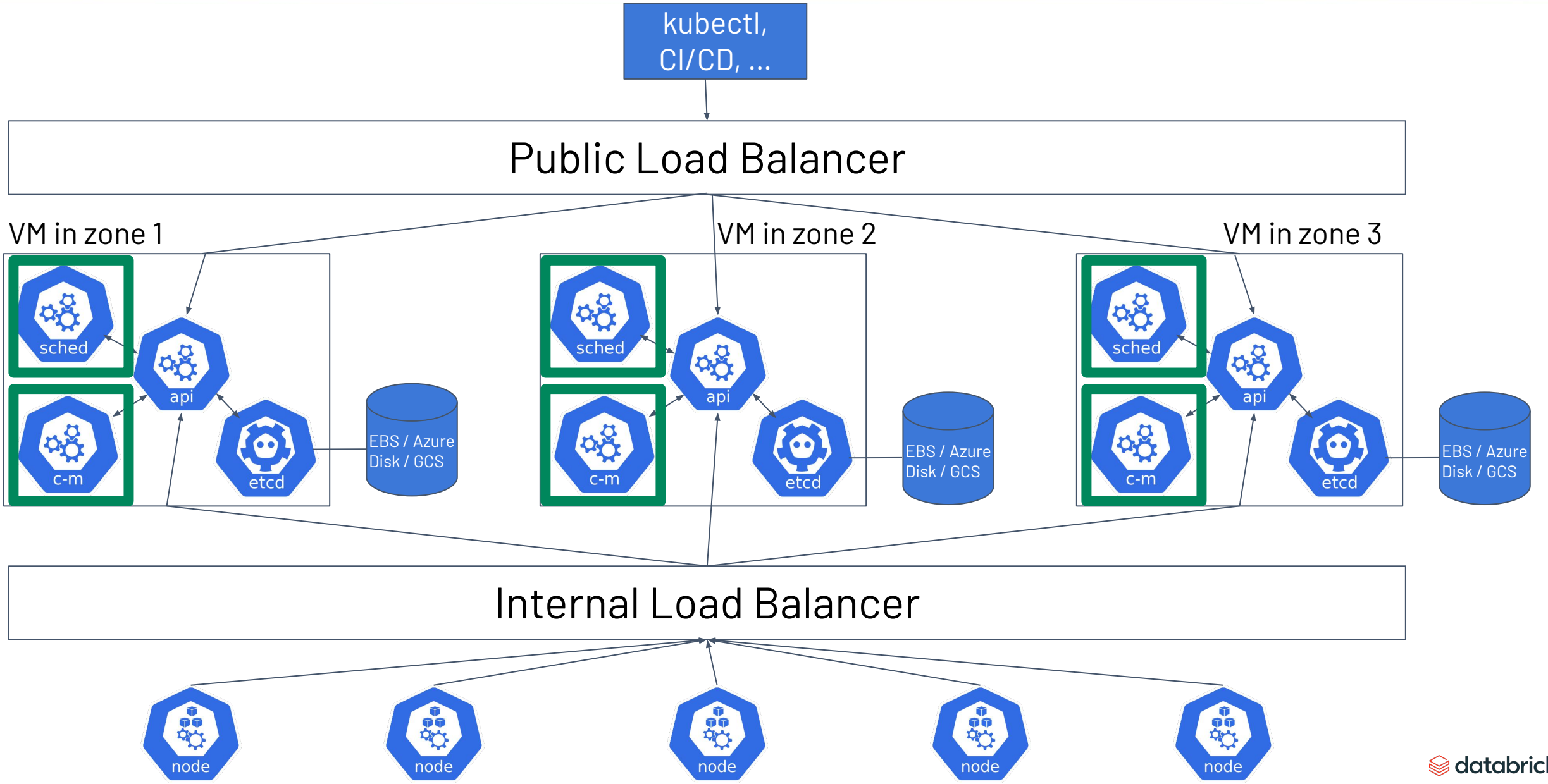
HA cluster architecture



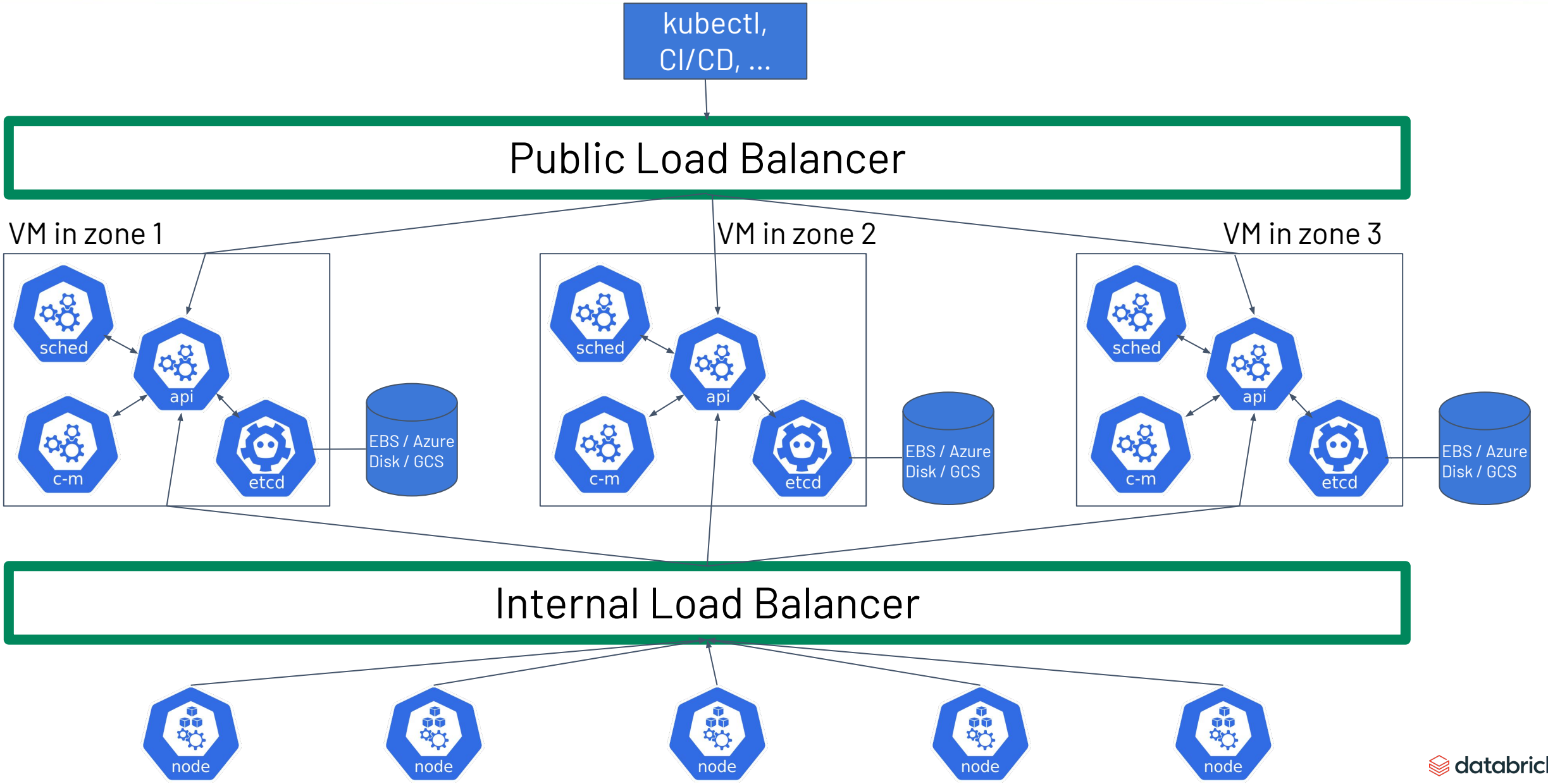
HA cluster architecture



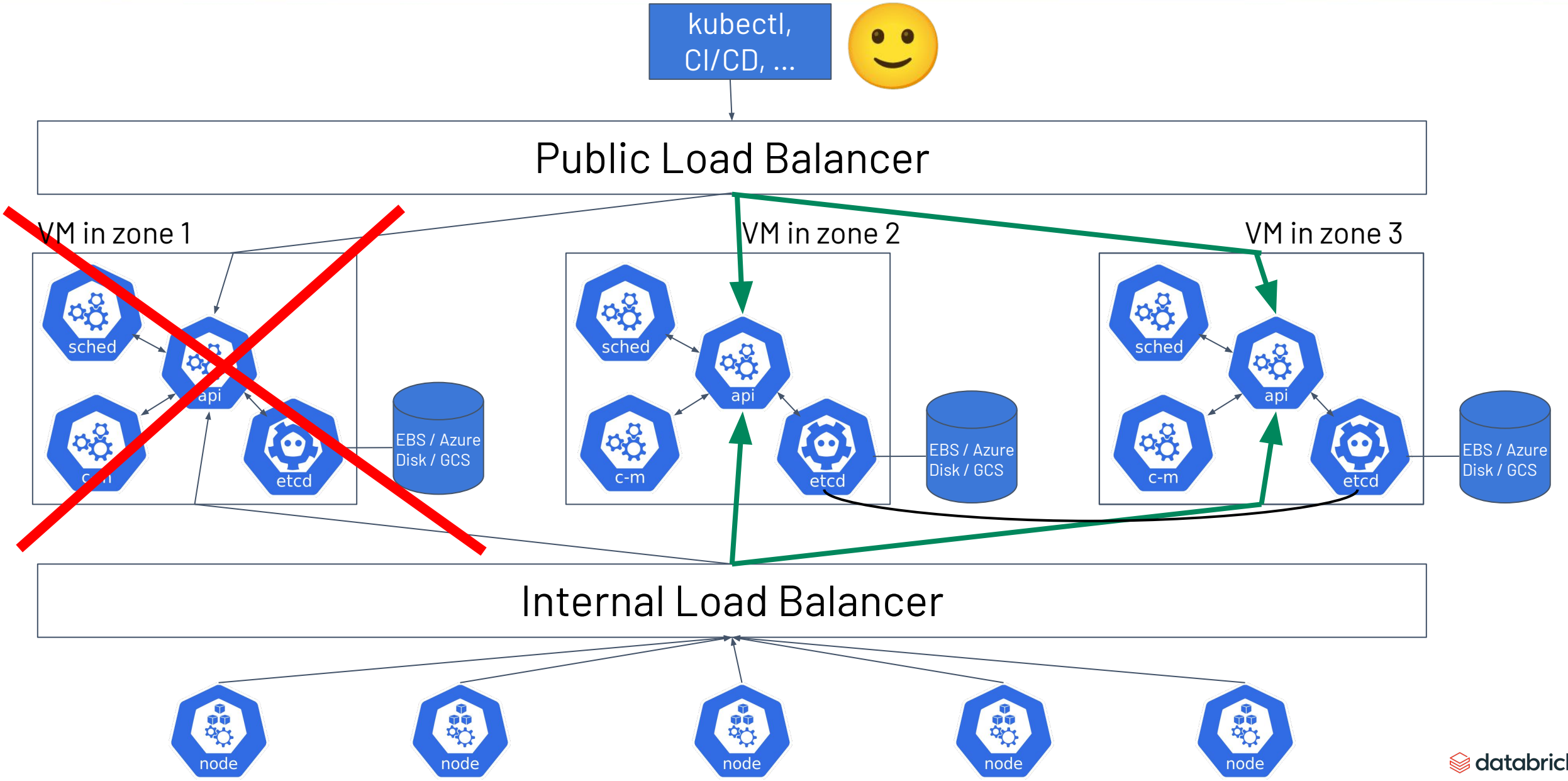
HA cluster architecture



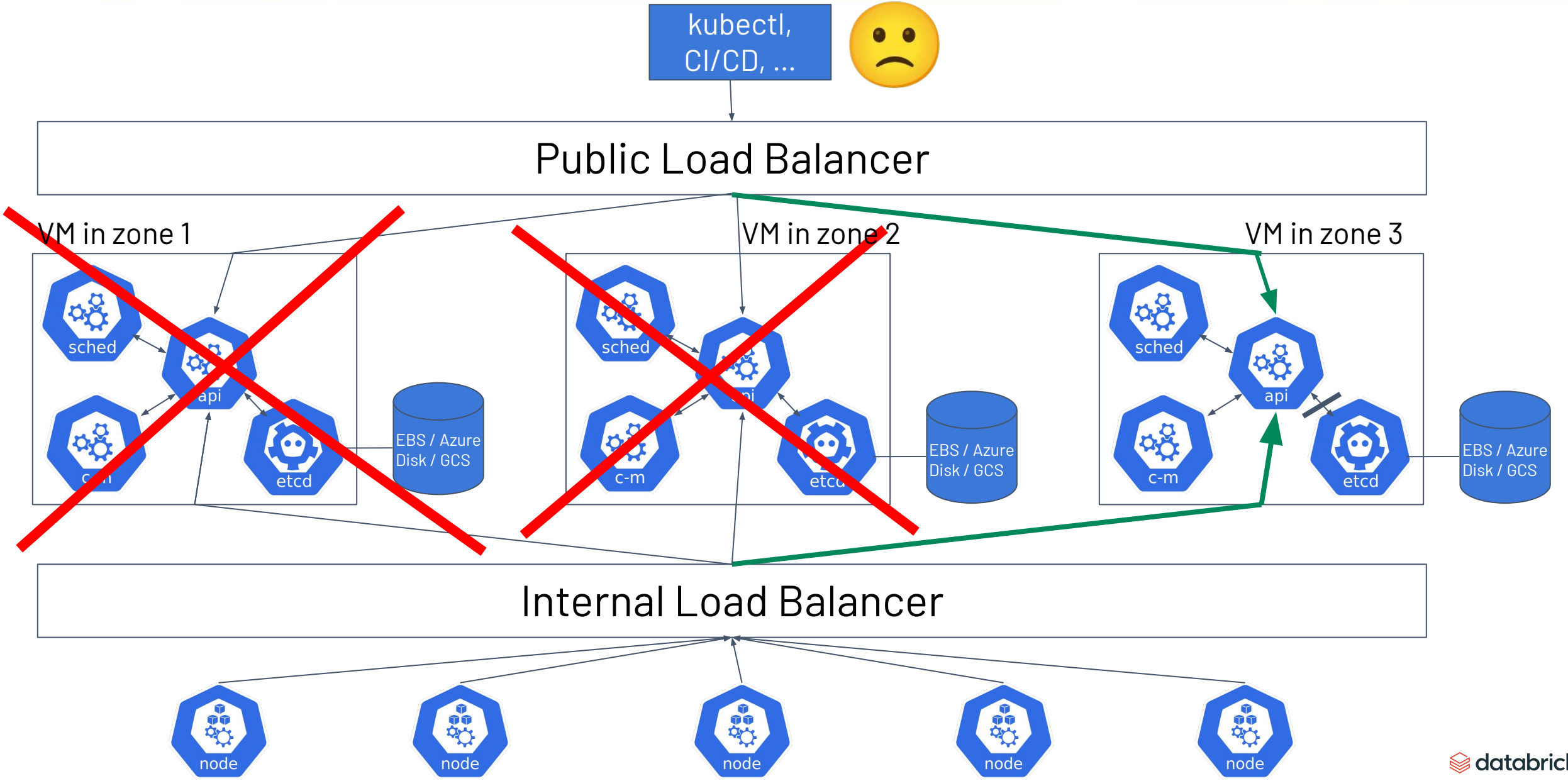
HA cluster architecture



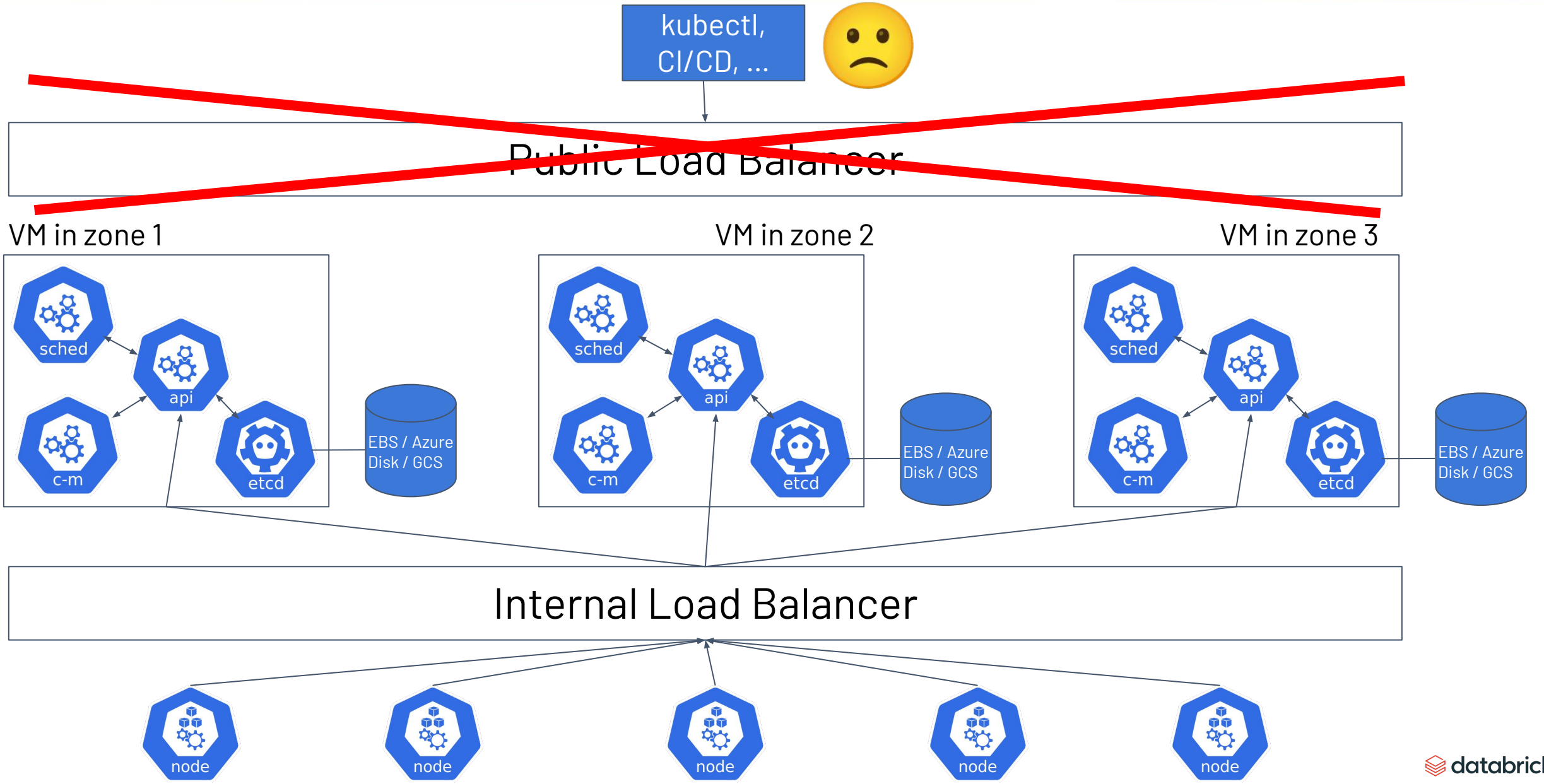
Single zone failure



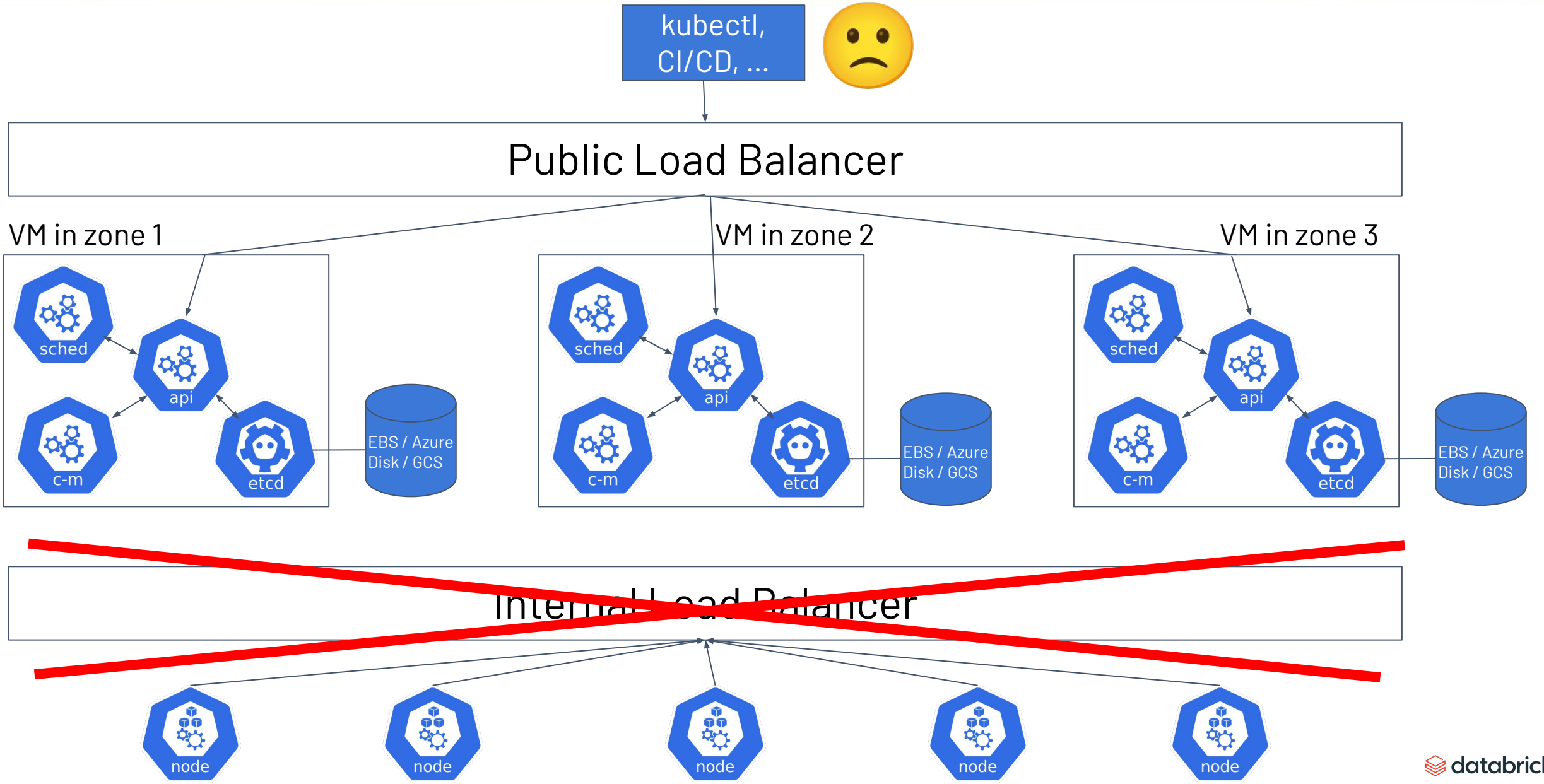
Two zone failures



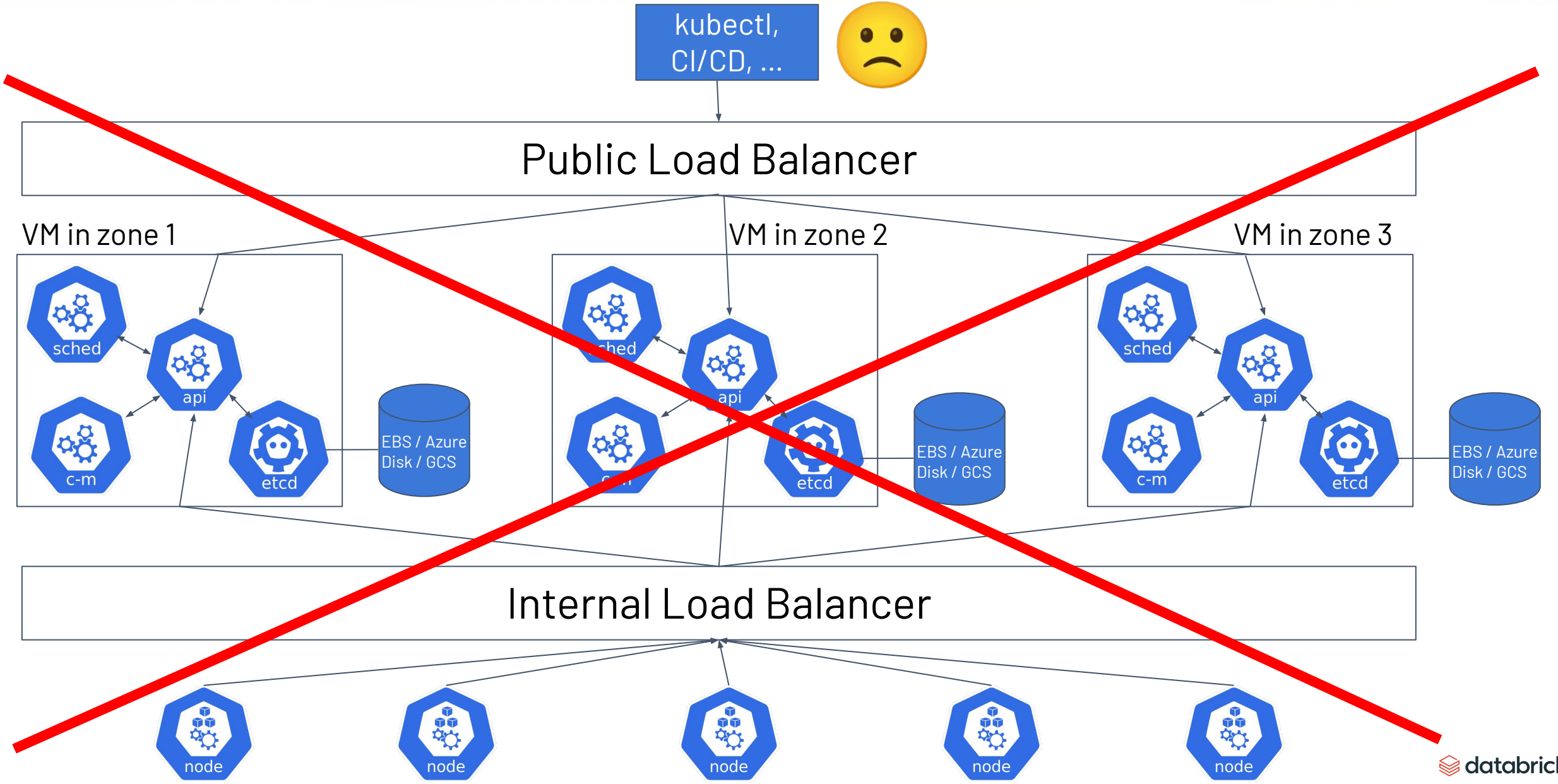
Public load balancer failure



Internal load balancer failure



Region failure



Migrating from non-HA to HA

Requirements for migration process

All workloads running on the cluster must continue to run during migration

No client reconfiguration (kubelets, workloads, external clients)

Per-cluster migration (and rollback) should be automated and safe

Migration across the fleet should be automated and safe

Non-HA to HA migration process

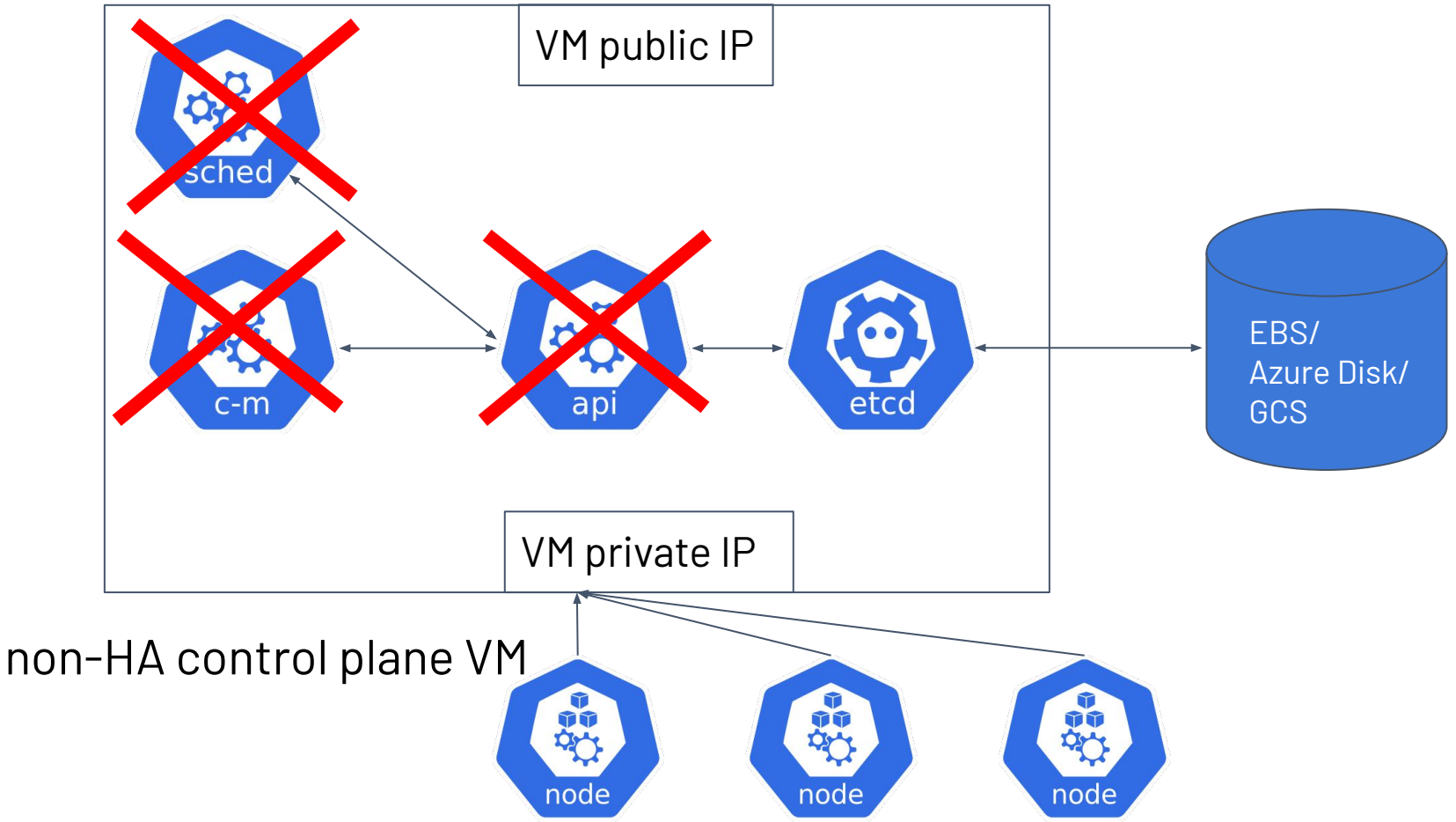
1. Snapshot cluster state & shutdown non-HA CP

2. Bootstrap HA CP without allowing mutating cluster state

3. Enable HA CP access, allowing mutating cluster state

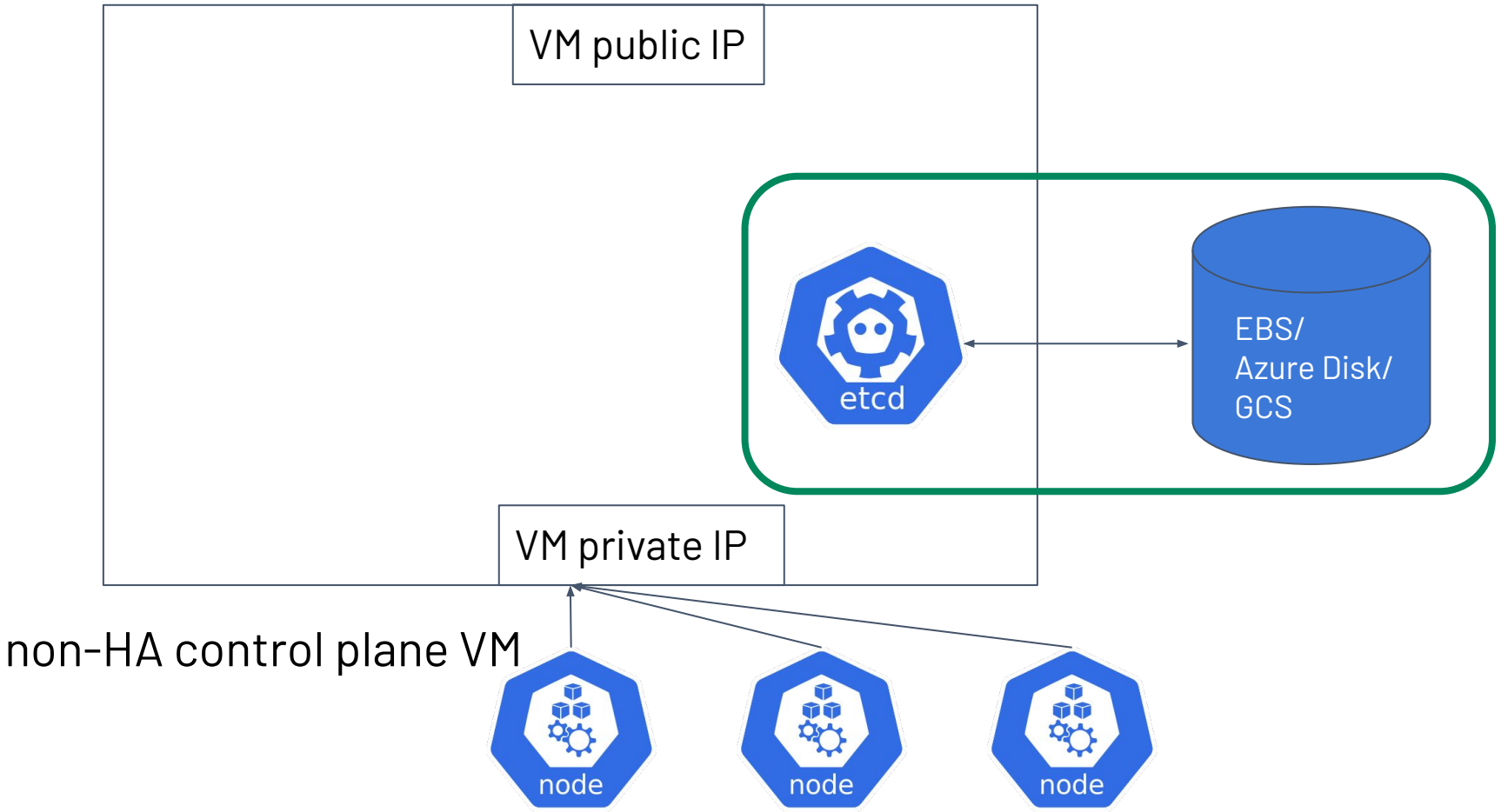
1. Snapshot cluster state

1.1. Terminate control plane pods to stop mutating cluster state in etcd



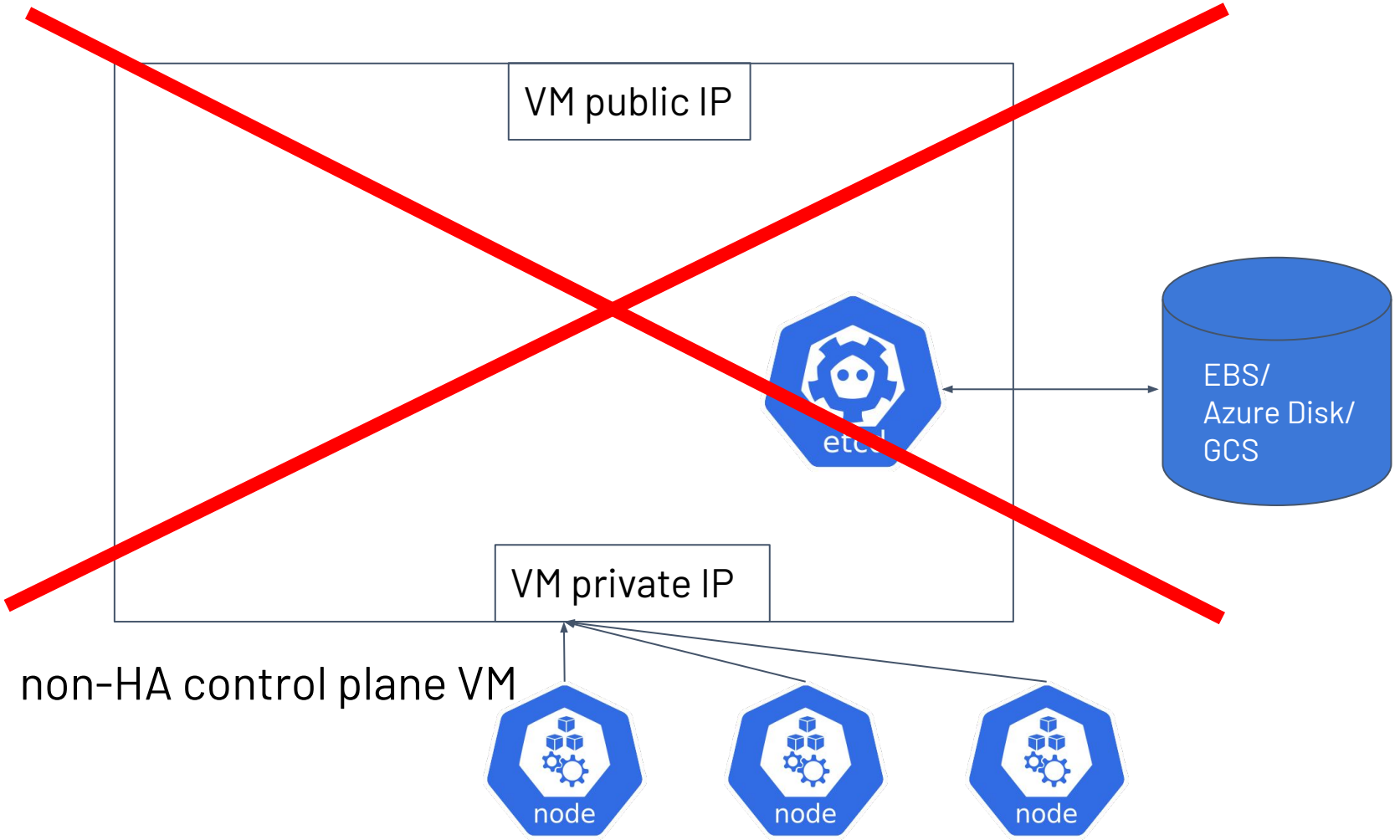
1. Snapshot cluster state

1.2. Use *etcdctl snapshot save* to take a snapshot of etcd state



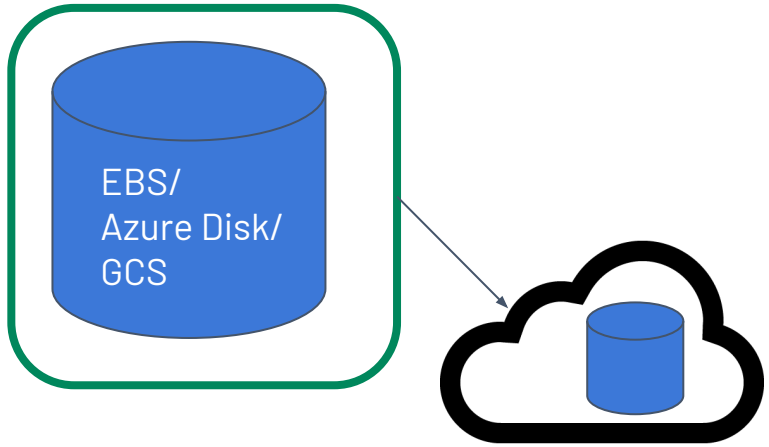
1. Snapshot cluster state

1.3. Shut down control plane VM (releases IPs)



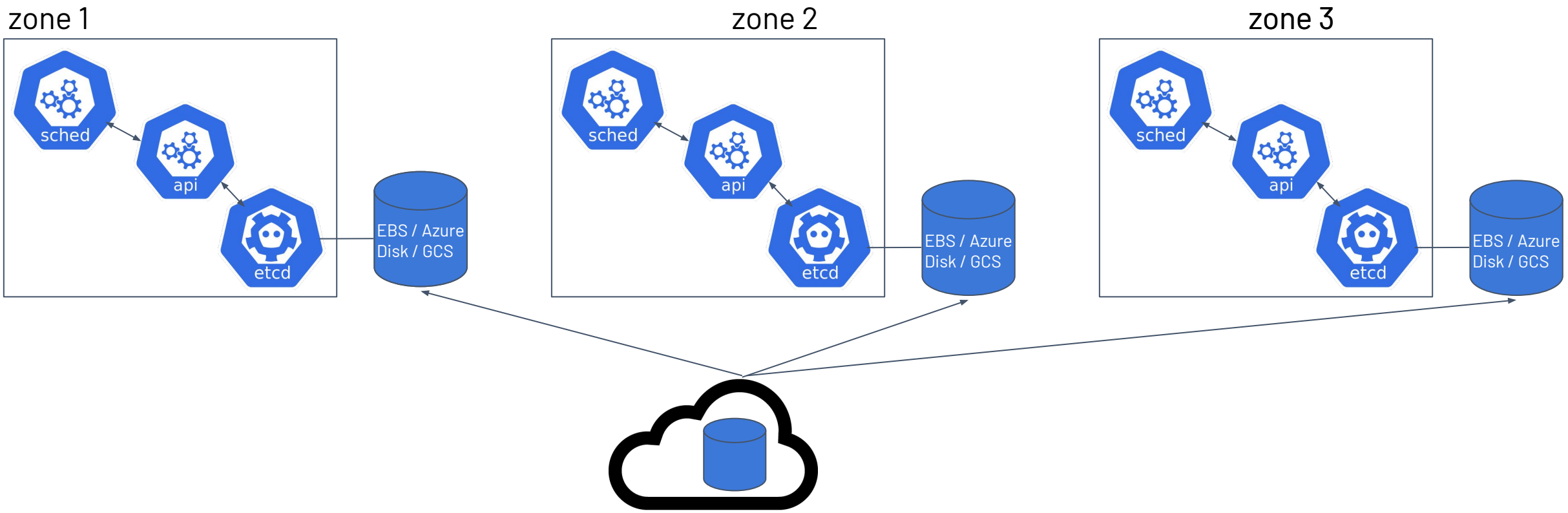
1. Snapshot cluster state

1.4. Take snapshot of etcd data disk using cloud provider API



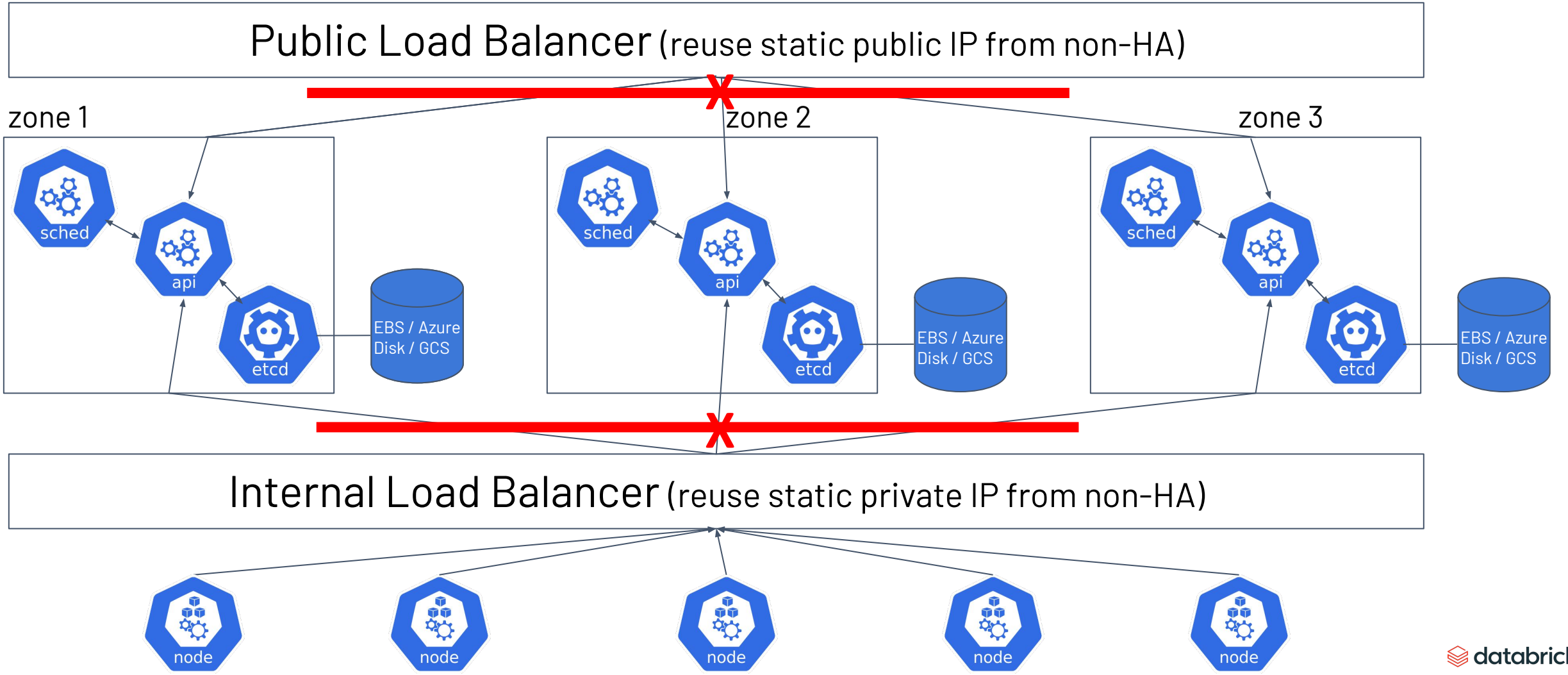
2. Bootstrap HA CP not allowing mutating state

2.1. Bootstrap 3 HA CP VMs from snapshot without starting controller-manager pods



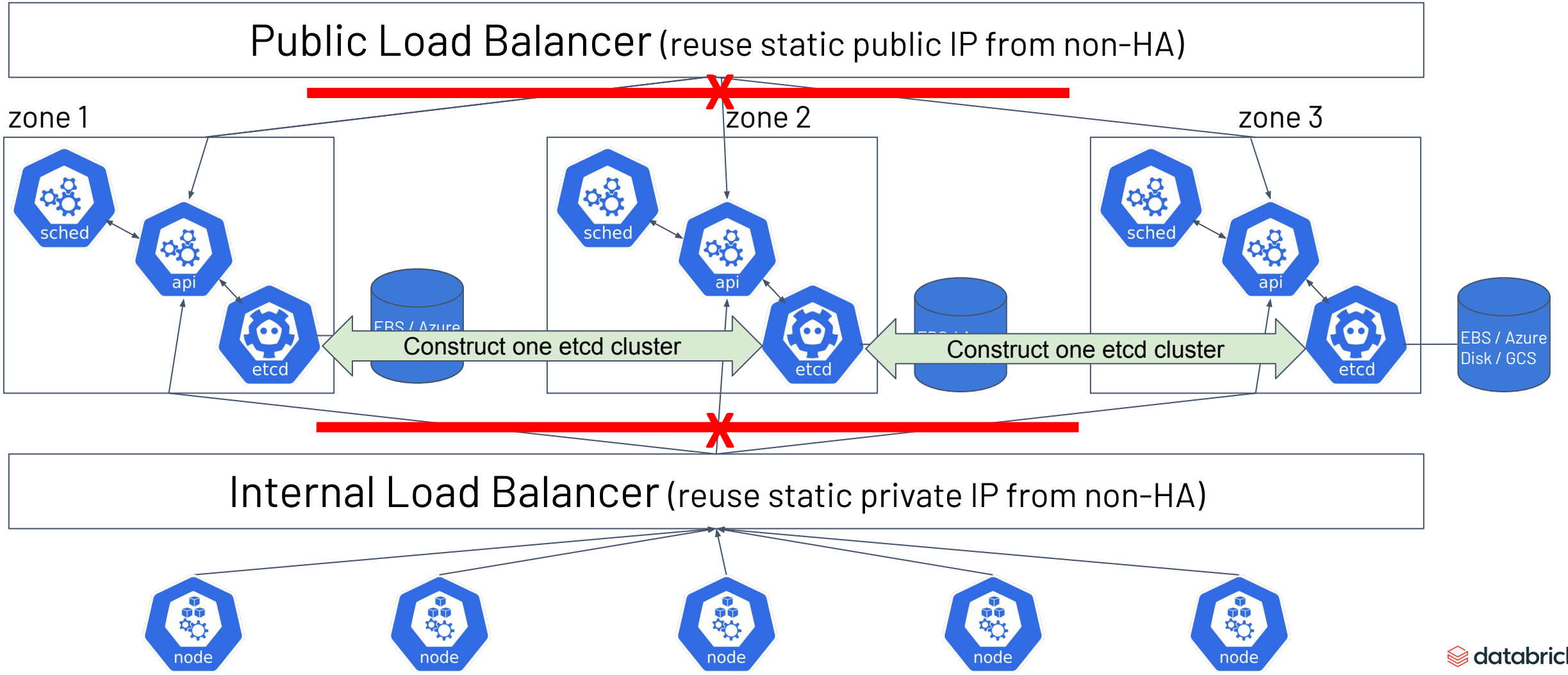
2. Bootstrap HA CP not allowing mutating state

2.2. Create Load balancers to HA CP VMs but not allowing access to api-server pods



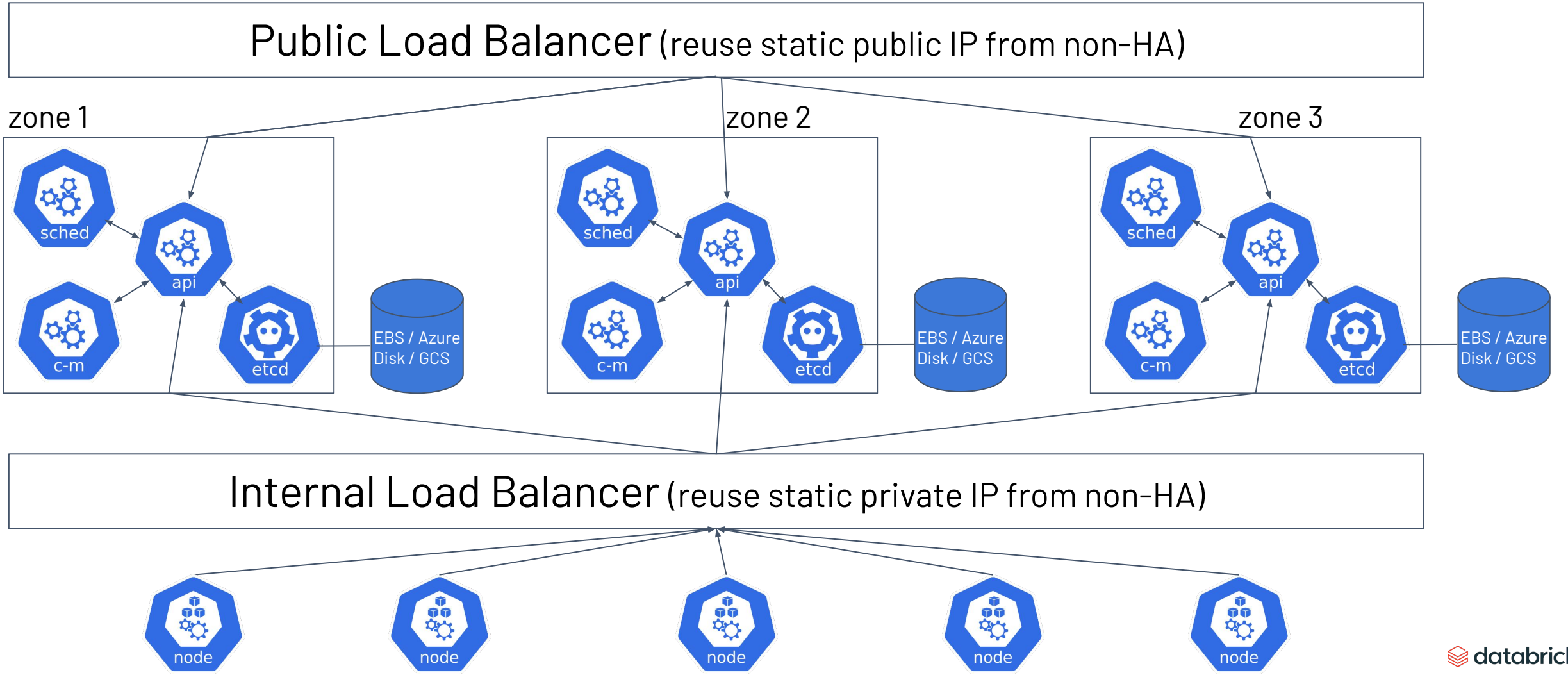
2. Bootstrap HA CP not allowing mutating state

2.3. Use *etcdctl snapshot restore* to restore etcd and configure new etcd cluster membership



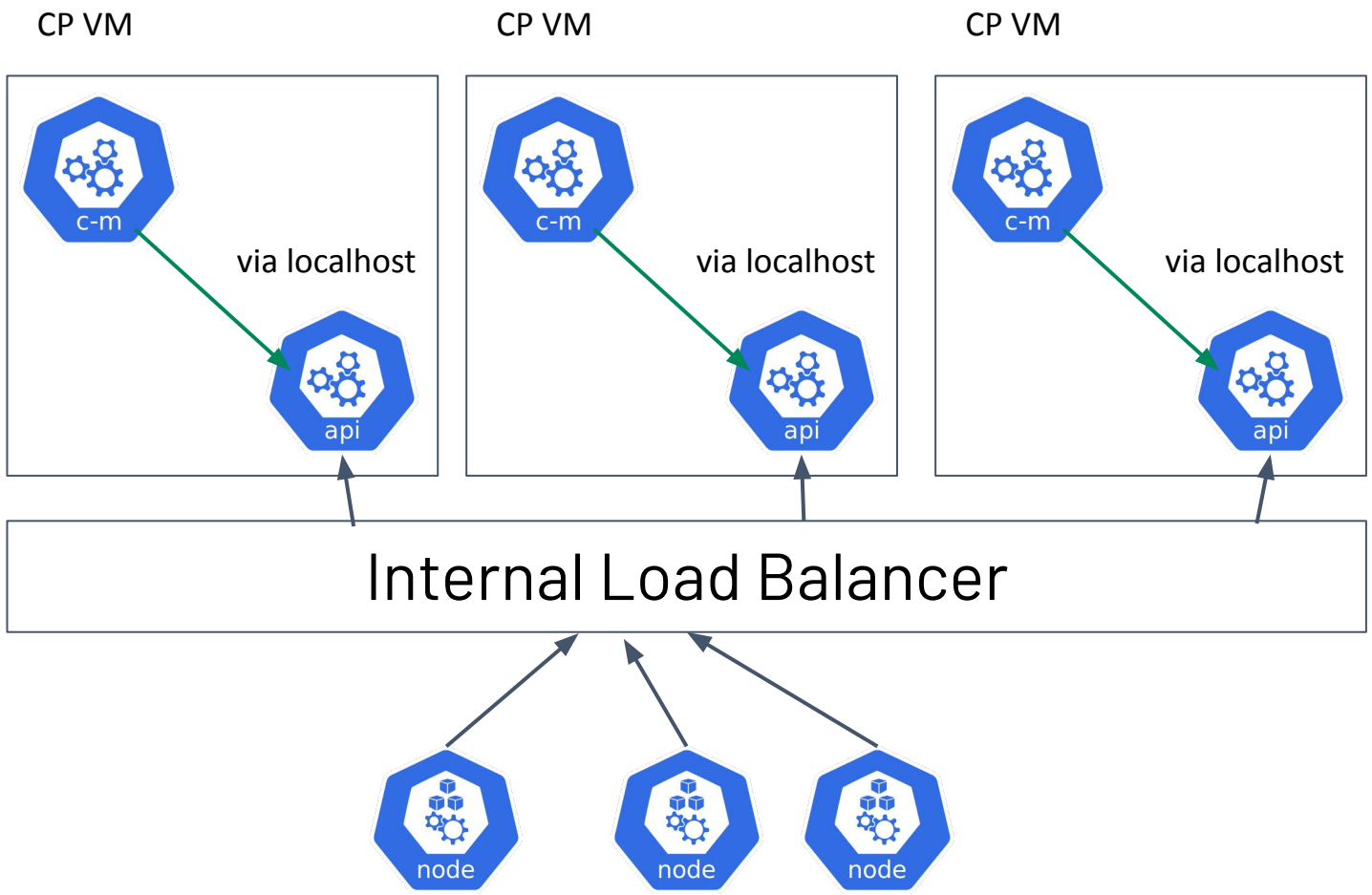
3. Enable HA CP with allowing mutating state

3.1. Start controller-manager pods and make both LBs accessible for api-server



Why not starting k-c-m before opening traffic

Avoid race condition between controller manager startup and kubelet heartbeat



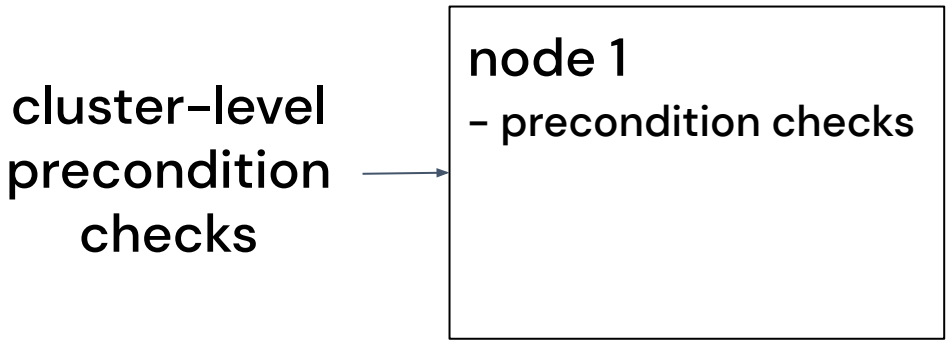
KubeCon | CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022

Day 2 with HA control plane

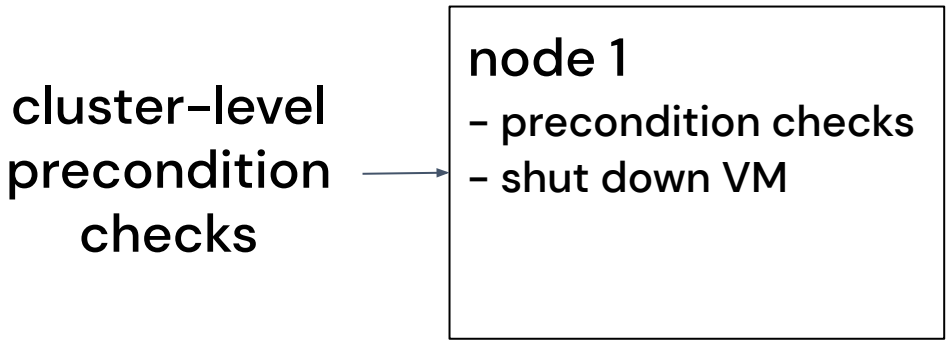
Control Plane k8s version upgrade with HA

cluster-level precondition checks

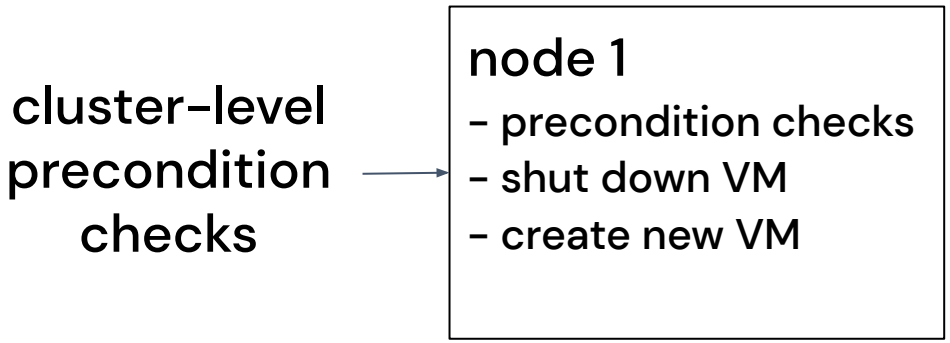
Control Plane k8s version upgrade with HA



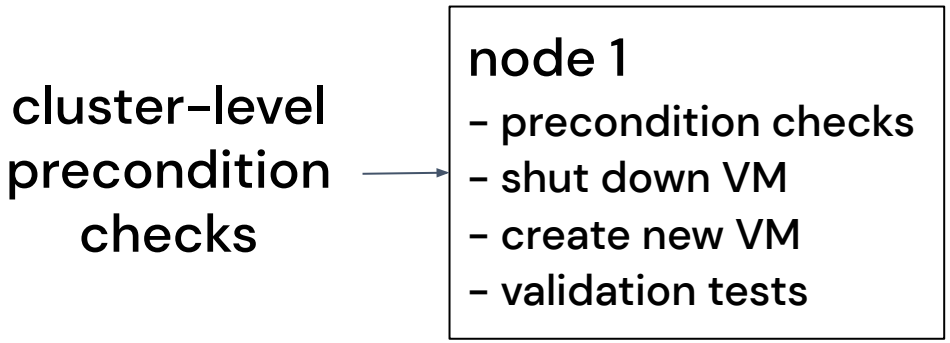
Control Plane k8s version upgrade with HA



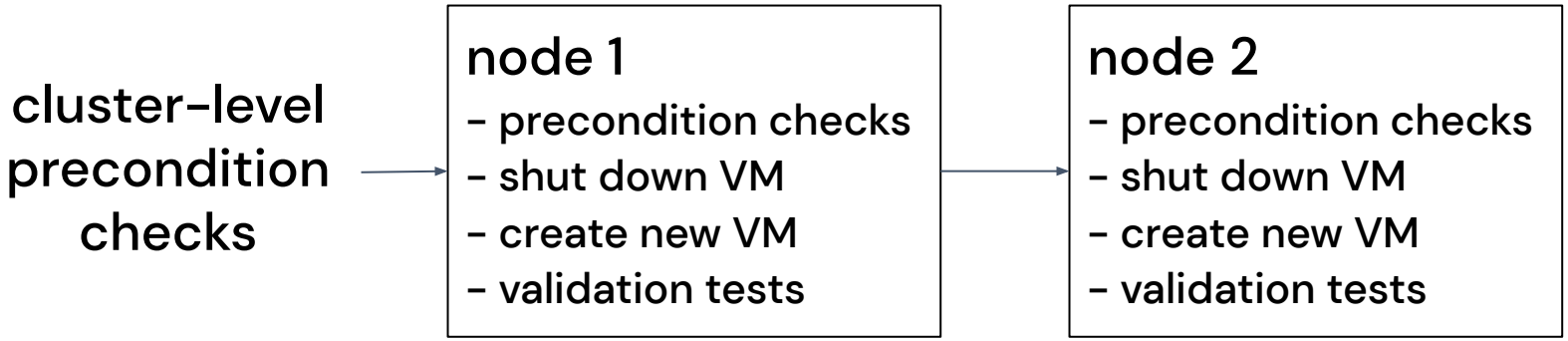
Control Plane k8s version upgrade with HA



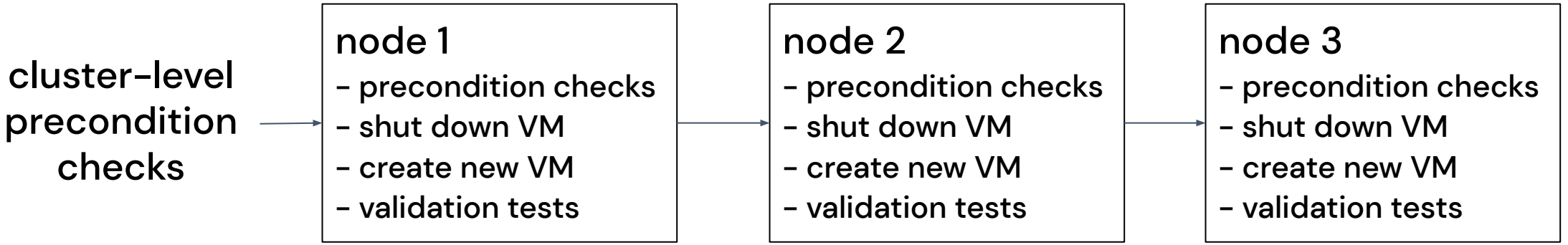
Control Plane k8s version upgrade with HA



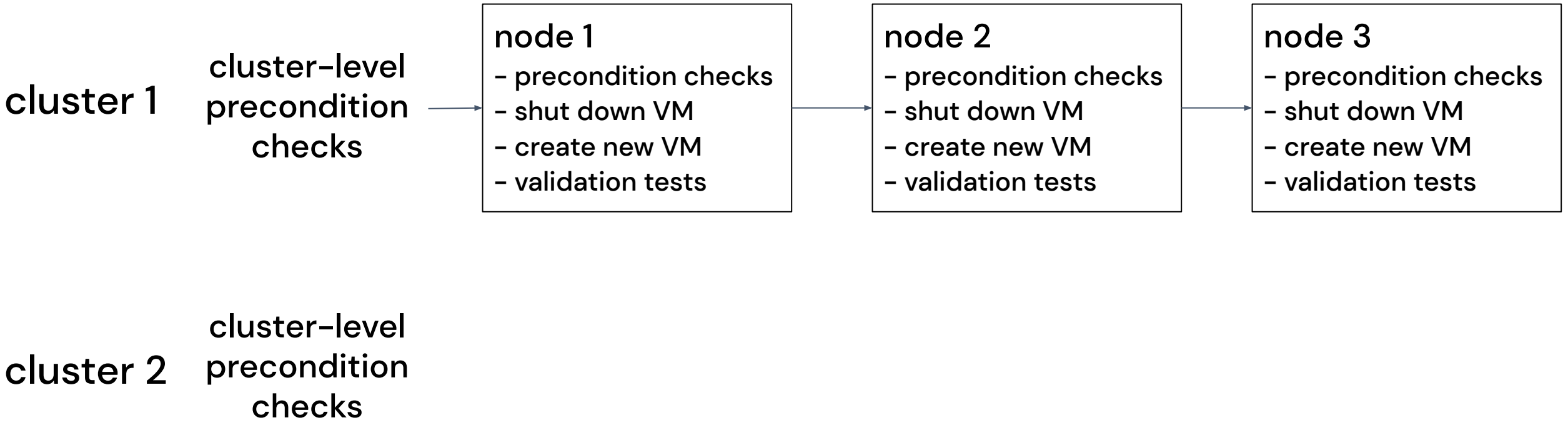
Control Plane k8s version upgrade with HA



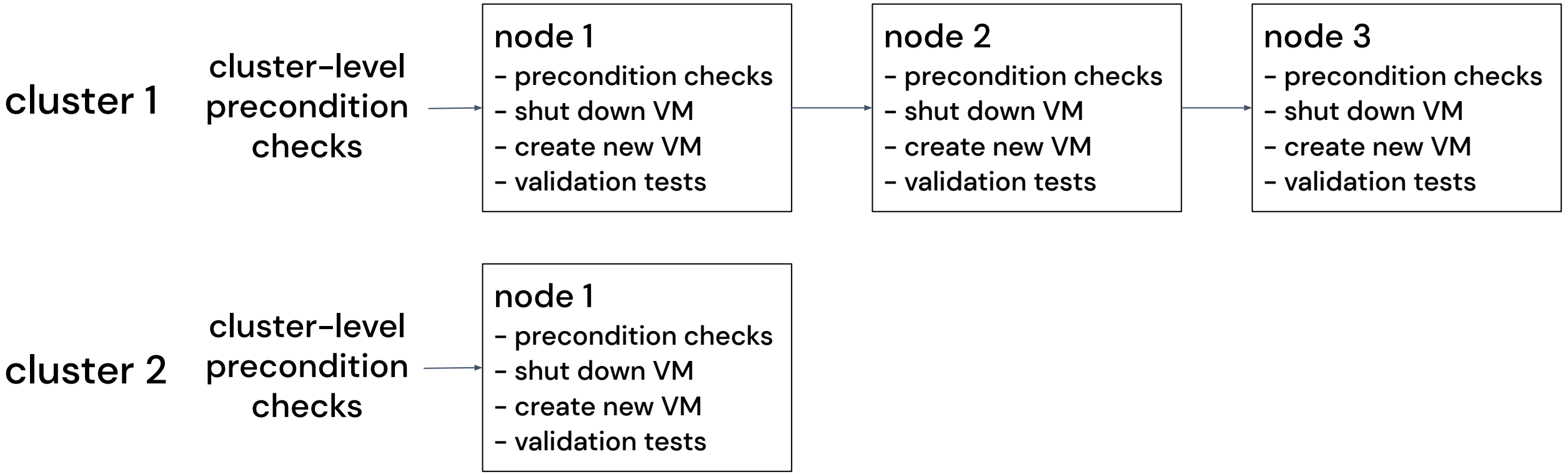
Control Plane k8s version upgrade with HA



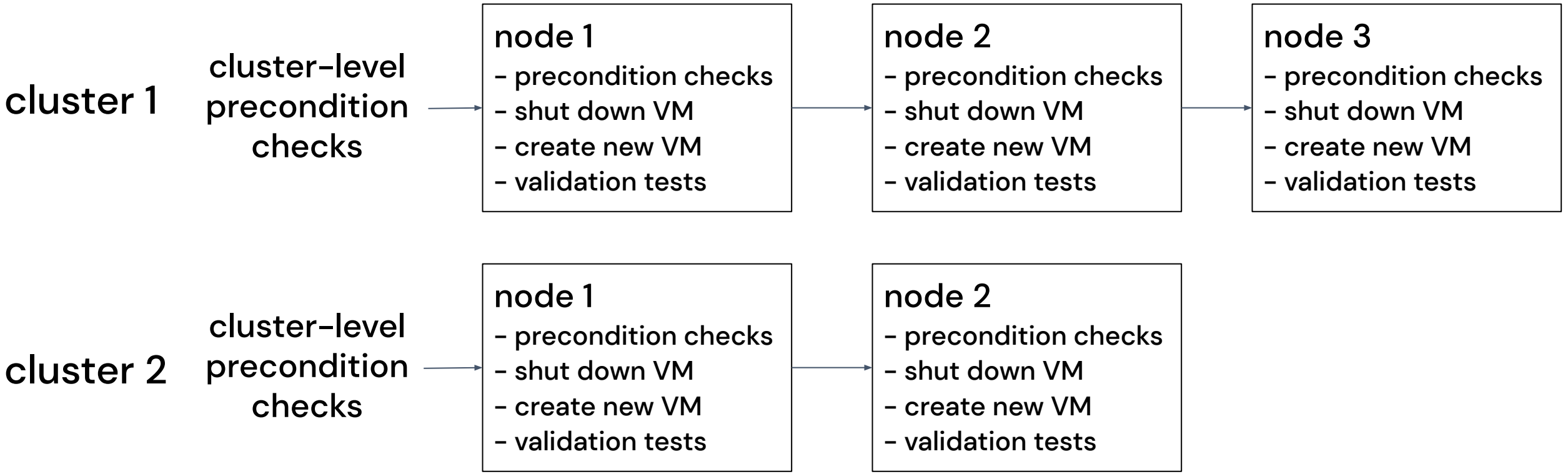
Control Plane k8s version upgrade with HA



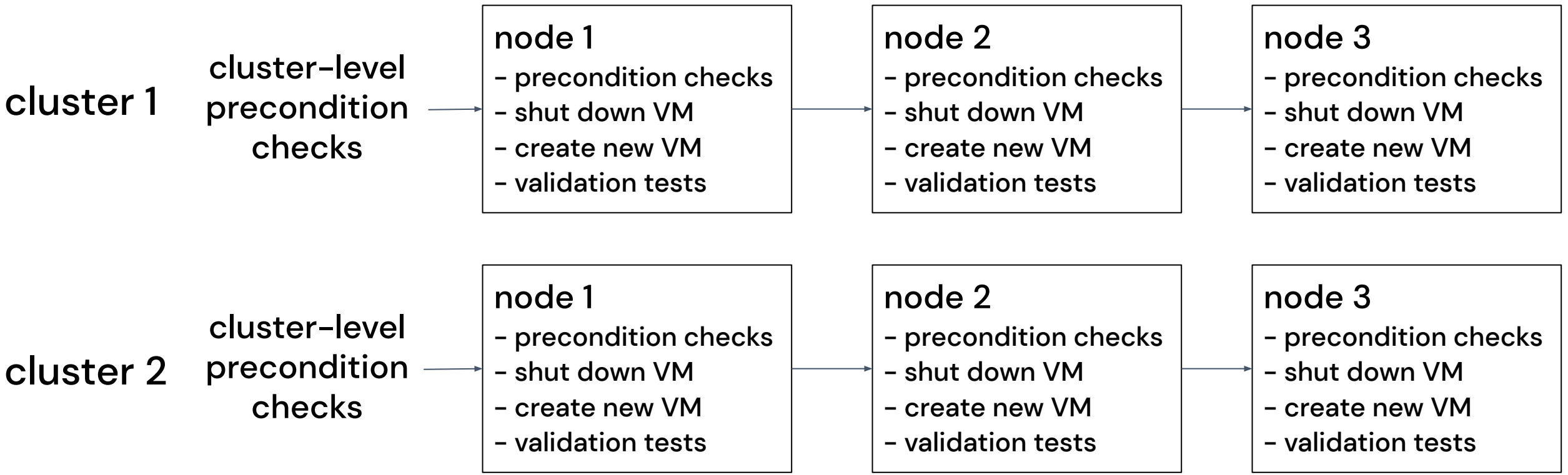
Control Plane k8s version upgrade with HA



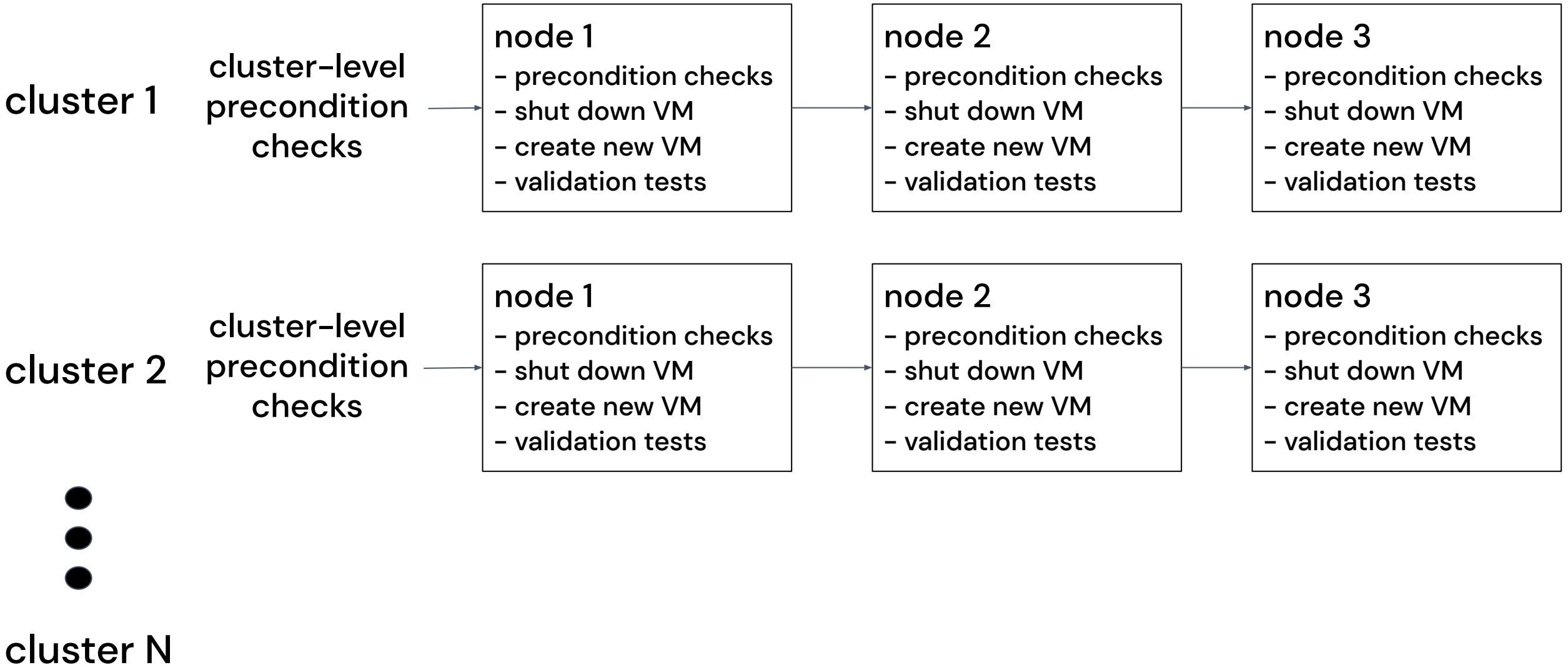
Control Plane k8s version upgrade with HA



Control Plane k8s version upgrade with HA



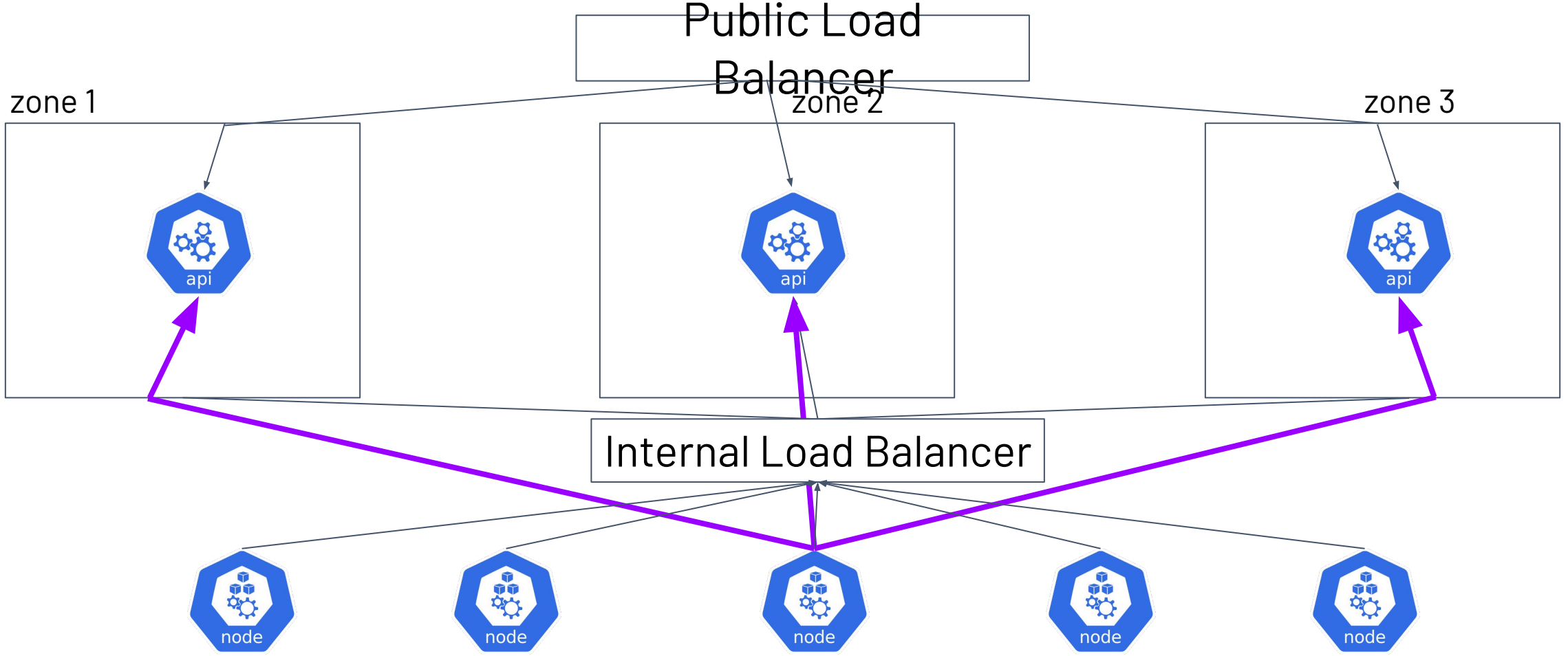
Control Plane k8s version upgrade with HA



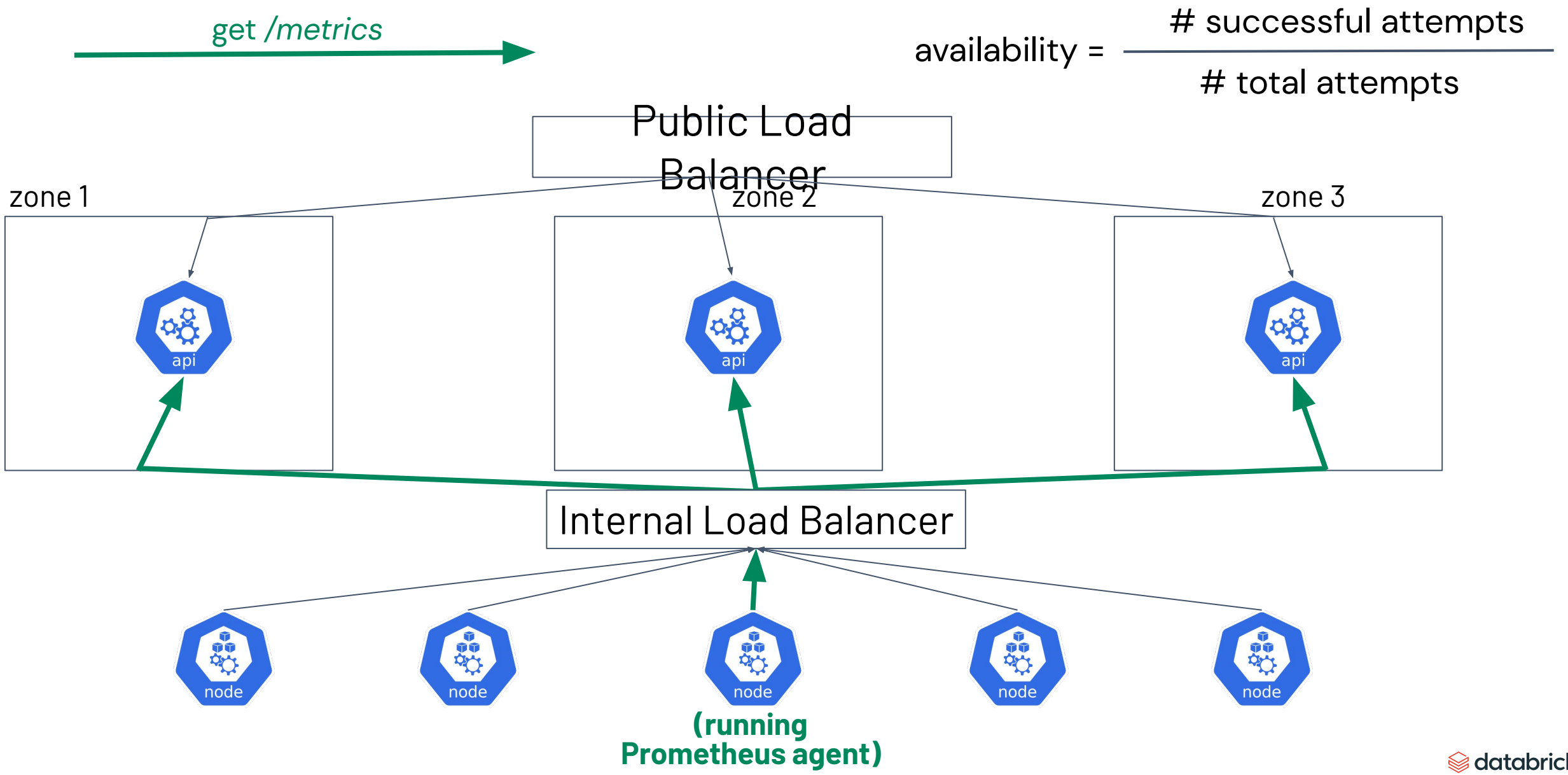
Monitoring and measuring availability

get *apiserver_request_total* from /metrics

$$\text{availability} = \frac{\#\ \text{successful calls}}{\#\ \text{total calls}}$$



Monitoring and measuring availability



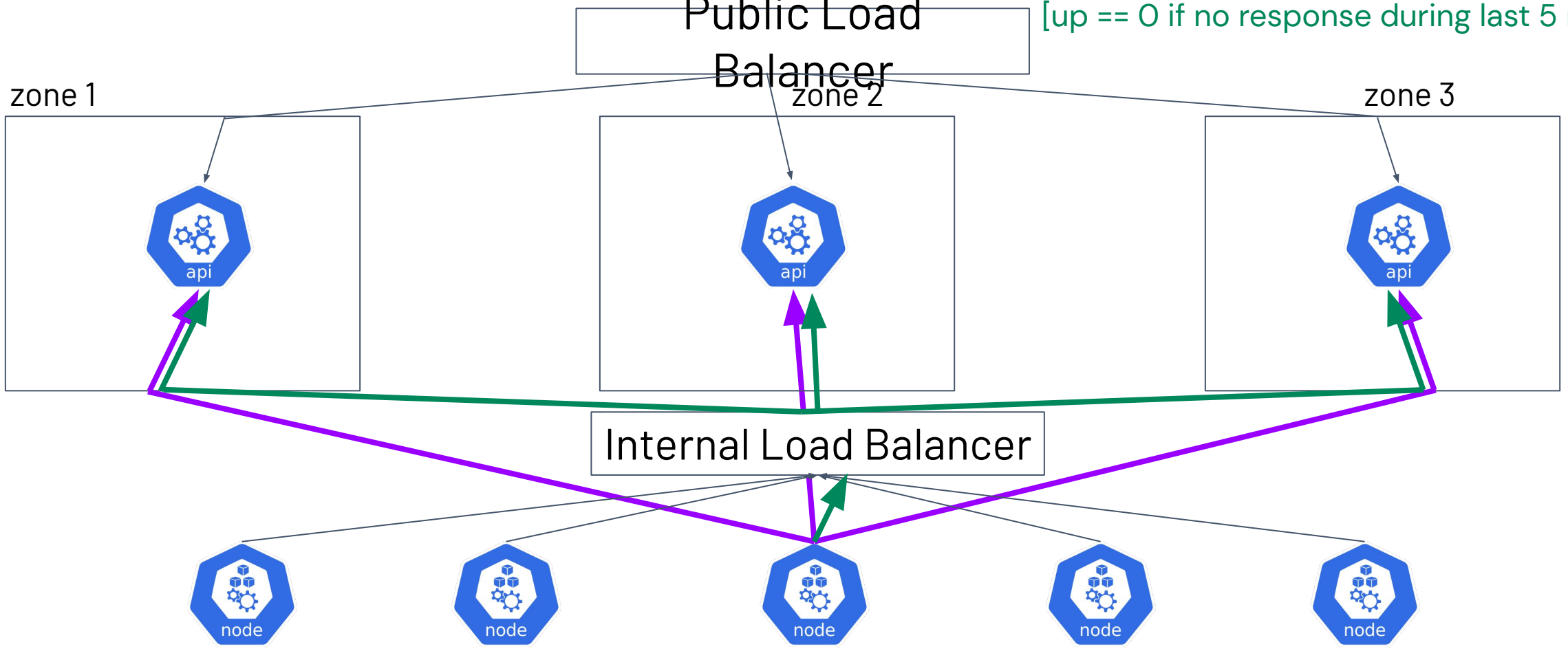
Monitoring and measuring availability

get *apiserver_request_total* from /metrics

get */metrics*

availability = $\frac{\text{\# successful calls}}{\text{\# total calls}}$ X up (0 or 1)

[up == 0 if no response during last 5 minutes]



Summary

Requirements for migration process

All workloads running on the cluster must continue to run during migration

- **snapshot single-node etcd + clone disk to create 2 new replicas**

Requirements for migration process

All workloads running on the cluster must continue to run during migration

- **snapshot single-node etcd + clone disk to create 2 new replicas**

No client reconfiguration (kubelets, workloads, external clients)

- **reuse single-node CP IP addresses as Load Balancer IP addresses**

Requirements for migration process

All workloads running on the cluster must continue to run during migration

- **snapshot single-node etcd + clone disk to create 2 new replicas**

No client reconfiguration (kubelets, workloads, external clients)

- **reuse single-node CP IP addresses as Load Balancer IP addresses**

Per-cluster migration (and rollback) should be automated and safe

- **migration script detects failures => roll back**
- **no state mutations until HA CP up and running => safe rollback**
- **ensure kubelets have heartbeated before enable controller-manager**

Requirements for migration process

All workloads running on the cluster must continue to run during migration

- **snapshot single-node etcd + clone disk to create 2 new replicas**

No client reconfiguration (kubelets, workloads, external clients)

- **reuse single-node CP IP addresses as Load Balancer IP addresses**

Per-cluster migration (and rollback) should be automated and safe

- **migration script detects failures => roll back**
- **no state mutations until HA CP up and running => safe rollback**
- **ensure kubelets have heartbeated before enable controller-manager**

Migration across the fleet should be automated and safe

- **pipeline for migration**
- **check pre-conditions and post-conditions for each cluster**

Additional lessons/observations

Availability metric should reflect end-user experience

- **% successful API requests, but zero if more than one CP node is down**

Additional lessons/observations

Availability metric should reflect end-user experience

- **% successful API requests, but zero if more than one CP node is down**

Additional benefit: HA CP makes upgrading to new k8s version zero-downtime

- upgrade one CP node at a time
- check for failures after each node, roll back if problem

Thank you!

https://www.databricks.com

https://www.databricks.com/blog/category/engineering