



KubeCon



CloudNativeCon

Europe 2020

*Virtual*

# Capacity-Aware Dynamic Volume Provisioning for LVM Local Storage

*Kazuhito Matsuda, Satoru Takeuchi*

- Kazuhito Matsuda / Satoru Takeuchi
  - Software engineer @Cybozu (JP) / Kintone (US)
    - A groupware cloud service provider from Japan
  - We are constructing Kubernetes-based on-premise datacenter (Project: Neco)

- Existing local storage solutions
- Dynamic provisioning: Motivation & Challenges
- Capacity-aware volume scheduling
- Introducing TopoLVM
- Demo
- Wrap up

# Why we use “local” storage

## ■ Pros

- I/O performance
- Cost

## ■ Cons

- Topology limitation
- Redundancy

 Local storage is a reasonable choice  
**for I/O-bound applications**



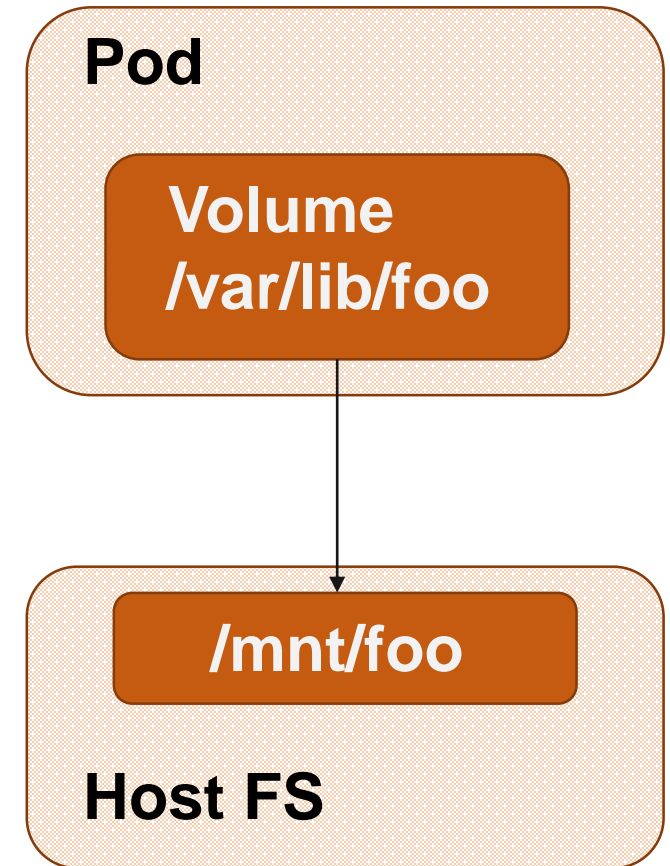
elastic



ROOK

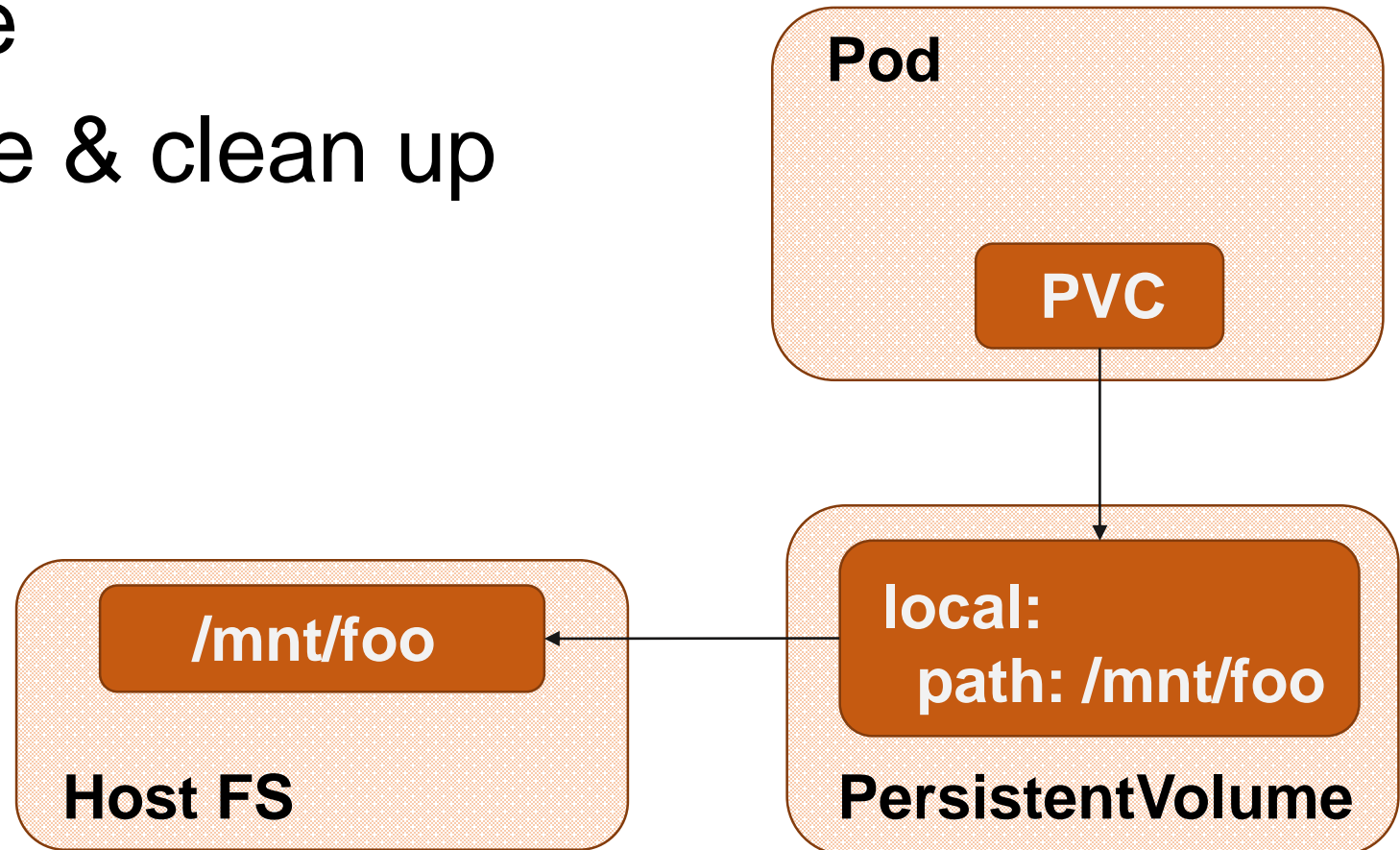
- `hostPath`
- Local persistent volume

- No portability
- Difficult to manage lifecycle
- **Not dynamic**



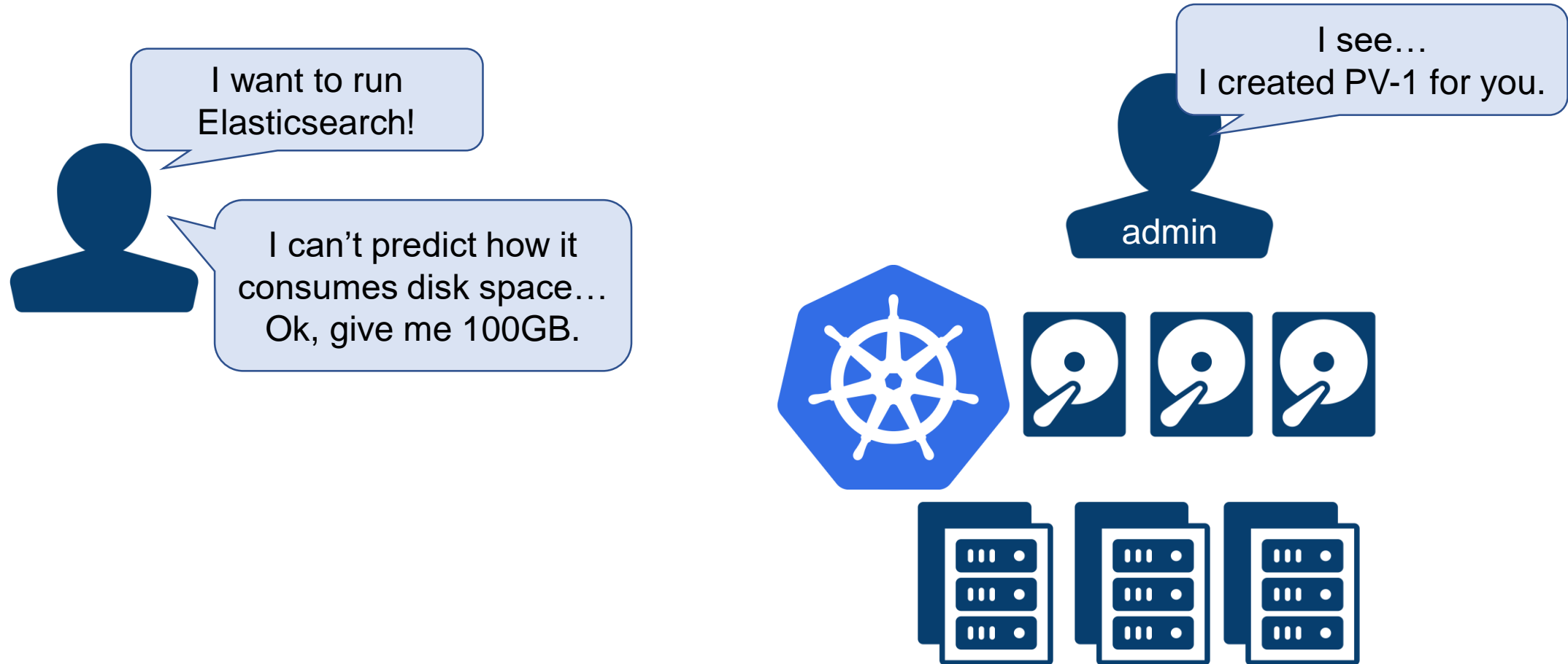
# Local persistent volume

- Local persistent volume
  - Topology-aware
  - Need to prepare & clean up
  - **Not dynamic**

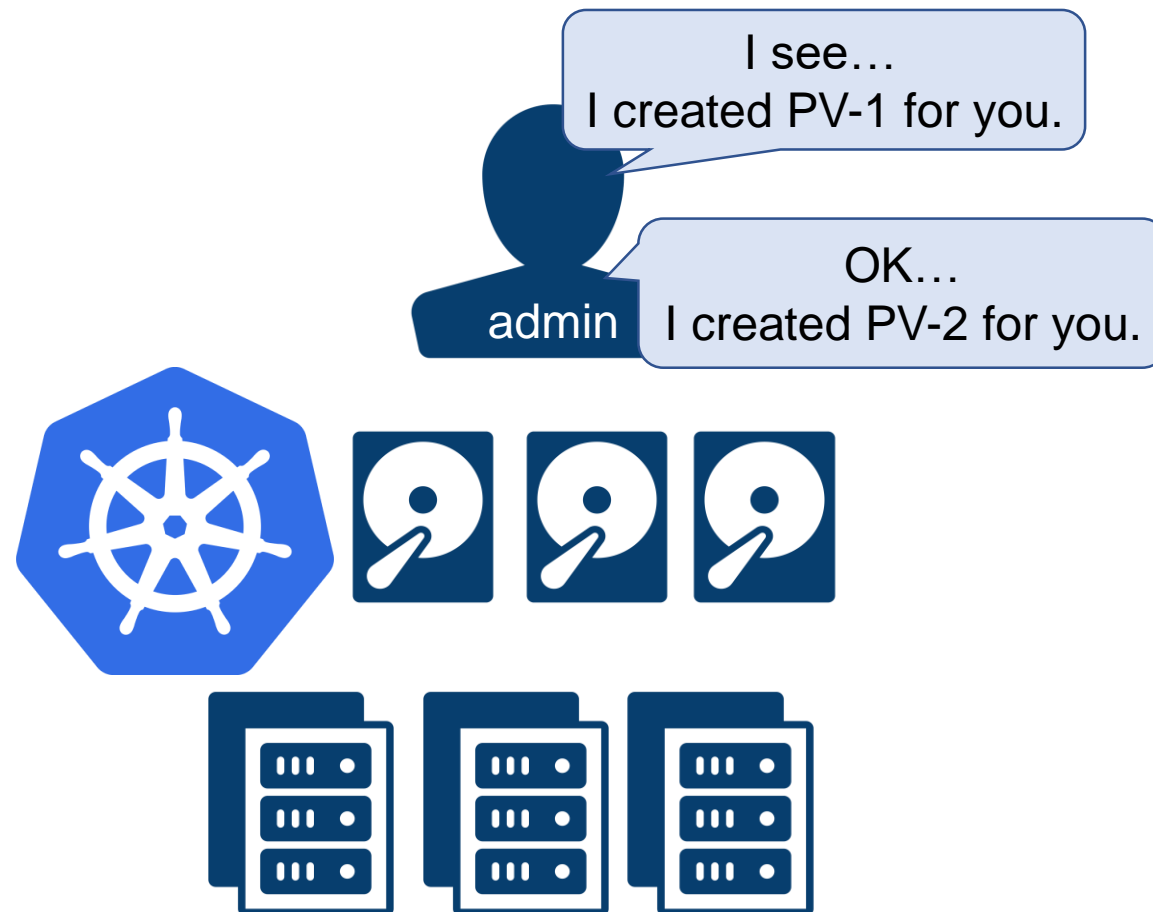
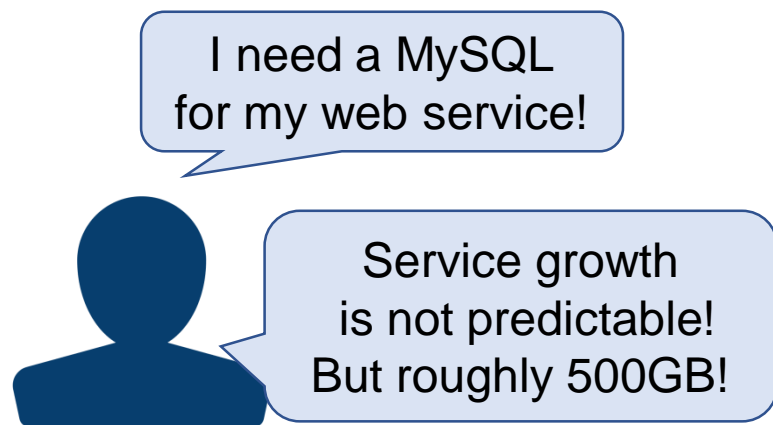
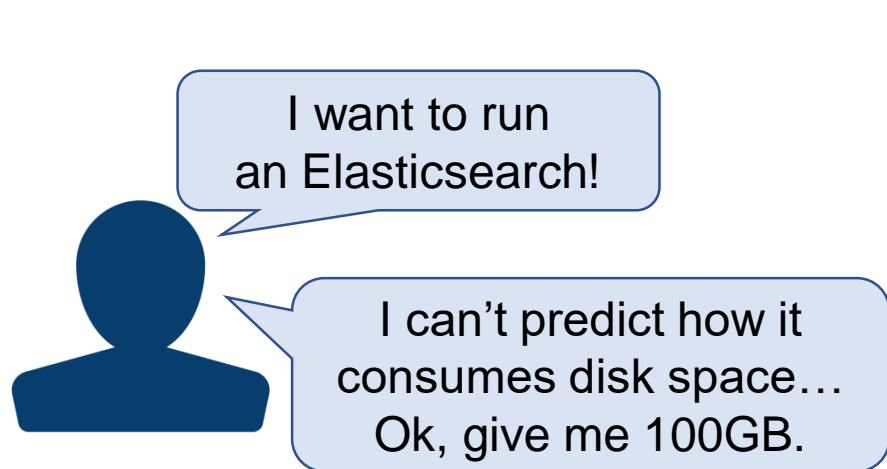




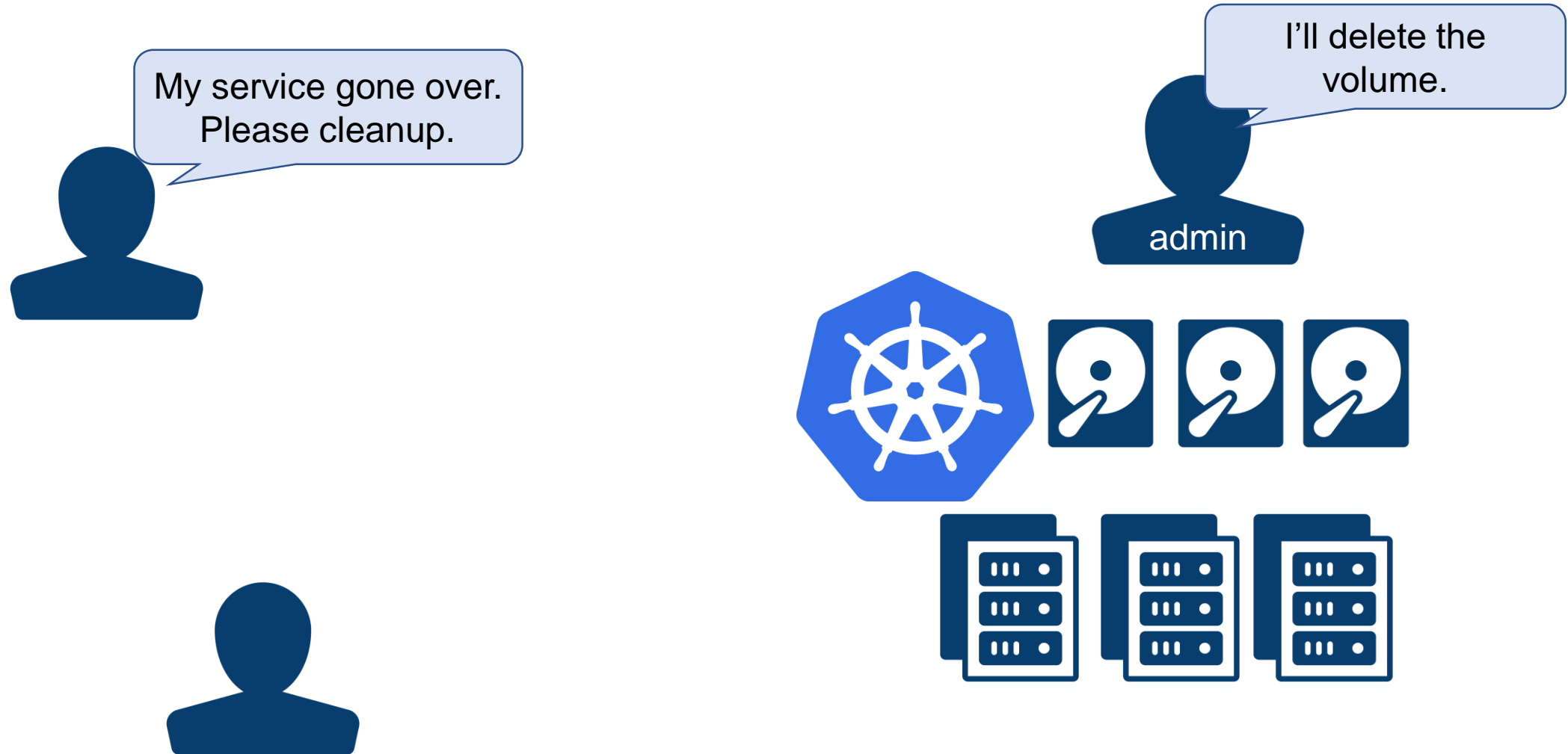
# Why dynamic provisioning



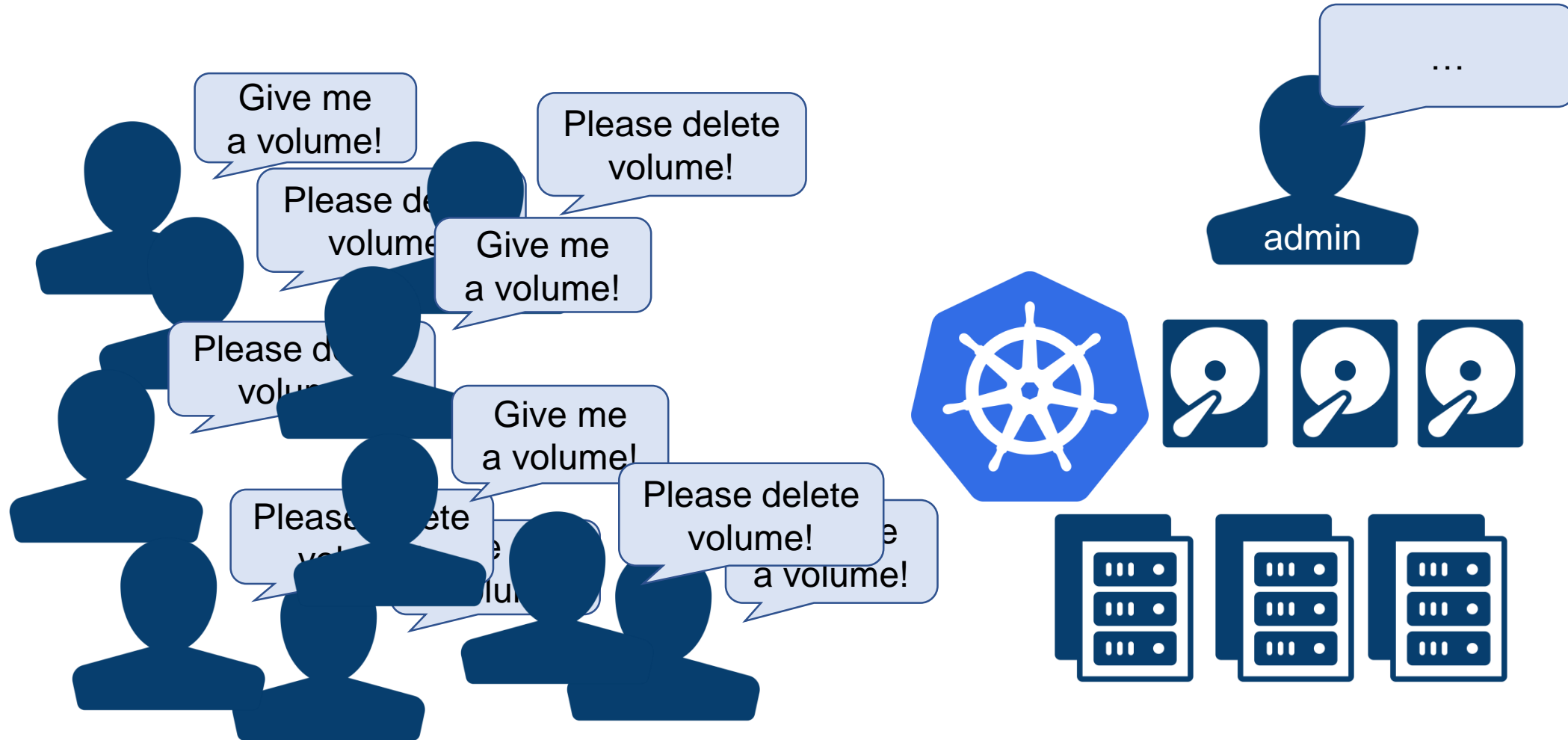
# Why dynamic provisioning



# Why dynamic provisioning



# Why dynamic provisioning



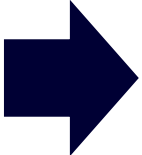
# Why dynamic provisioning



The dynamic provisioning feature eliminates the need for cluster administrators to pre-provision storage. Instead, it automatically provisions storage when it is requested by users.

quoted from: <https://kubernetes.io/docs/concepts/storage/dynamic-provisioning/>

- It provides:
  - Agility, Scalability, Accuracy

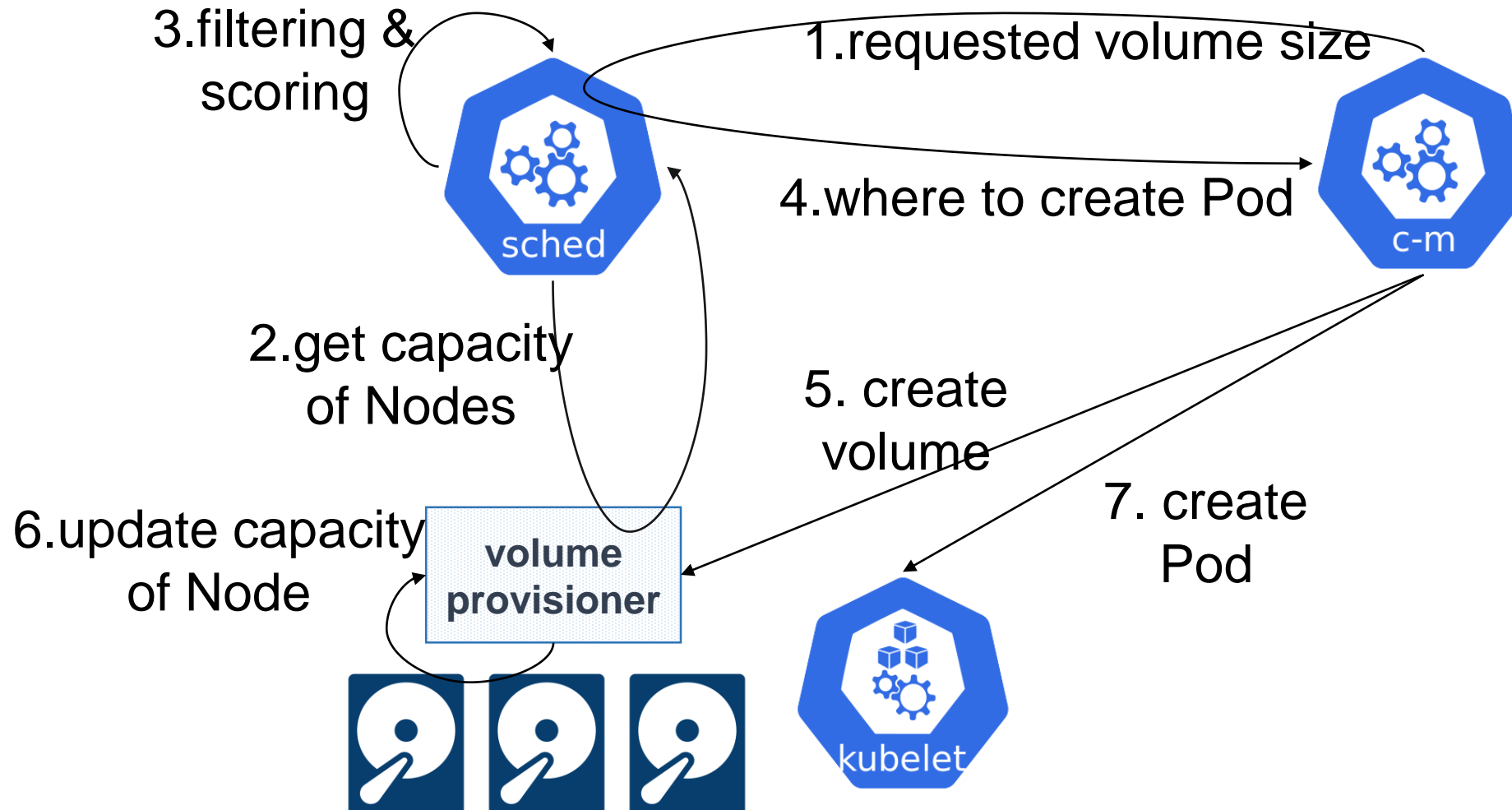
- Capacity-based filtering
    - We must **filter out nodes with insufficient storage space**
  - Room for resizing
    - A **node having larger free storage is more preferable**
-  **Capacity-awareness is the key!**



1. Gathering capacity metrics from Nodes
2. Filtering and scoring Nodes with the metrics by scheduler



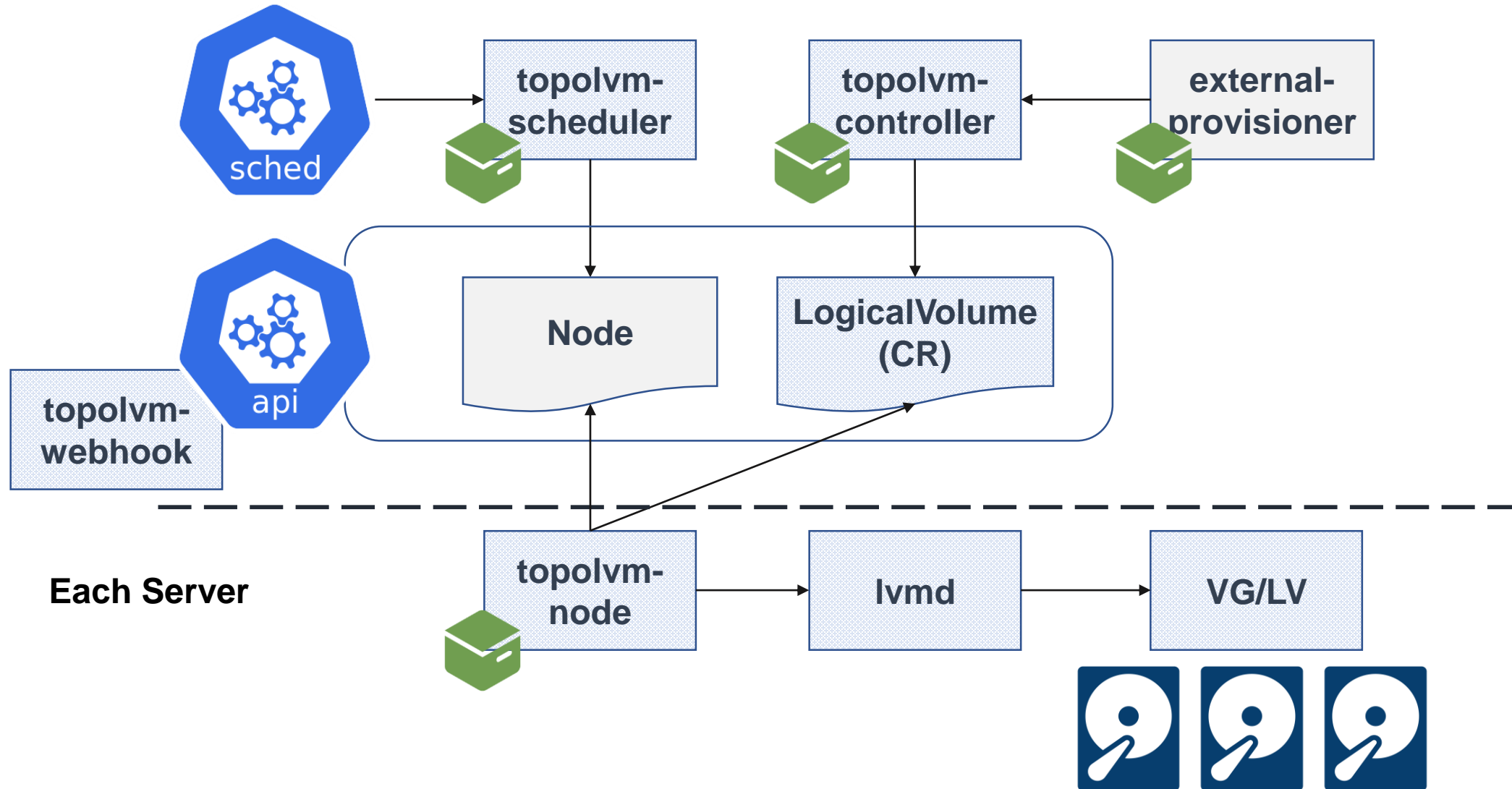
# Capacity-aware volume scheduling



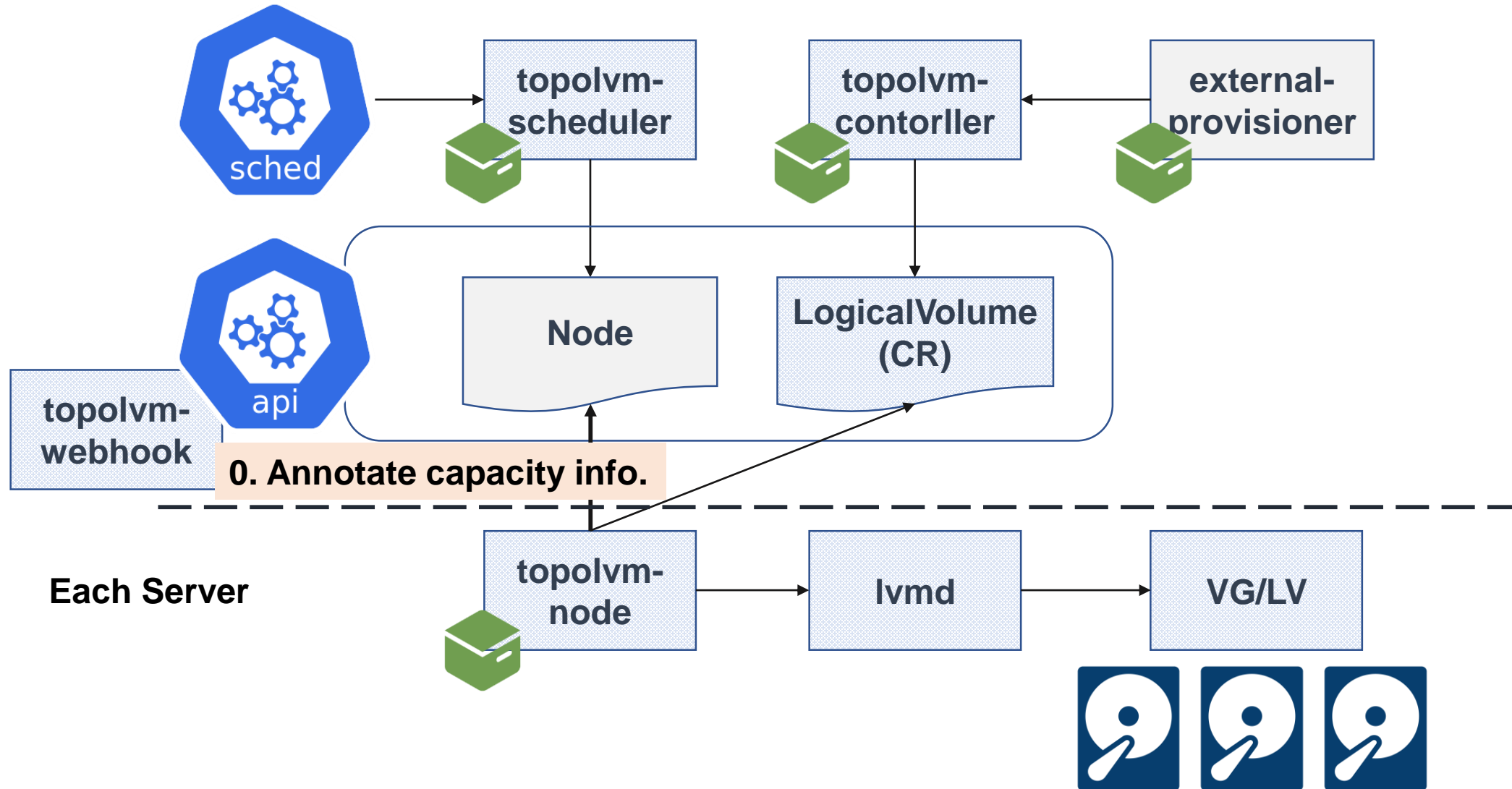
- KEP: Storage Capacity Constraints for Pod Scheduling
  - The discussion **ongoing**
  - <https://github.com/kubernetes/enhancements/pull/1353>

- LVM-based local storage driver conforming CSI
  - <https://github.com/kubernetes-csi/docs/blob/master/book/src/drivers.md>
- Features
  - Capacity-aware dynamic provisioning
  - Raw block volume
  - Online volume resizing

# Diagram

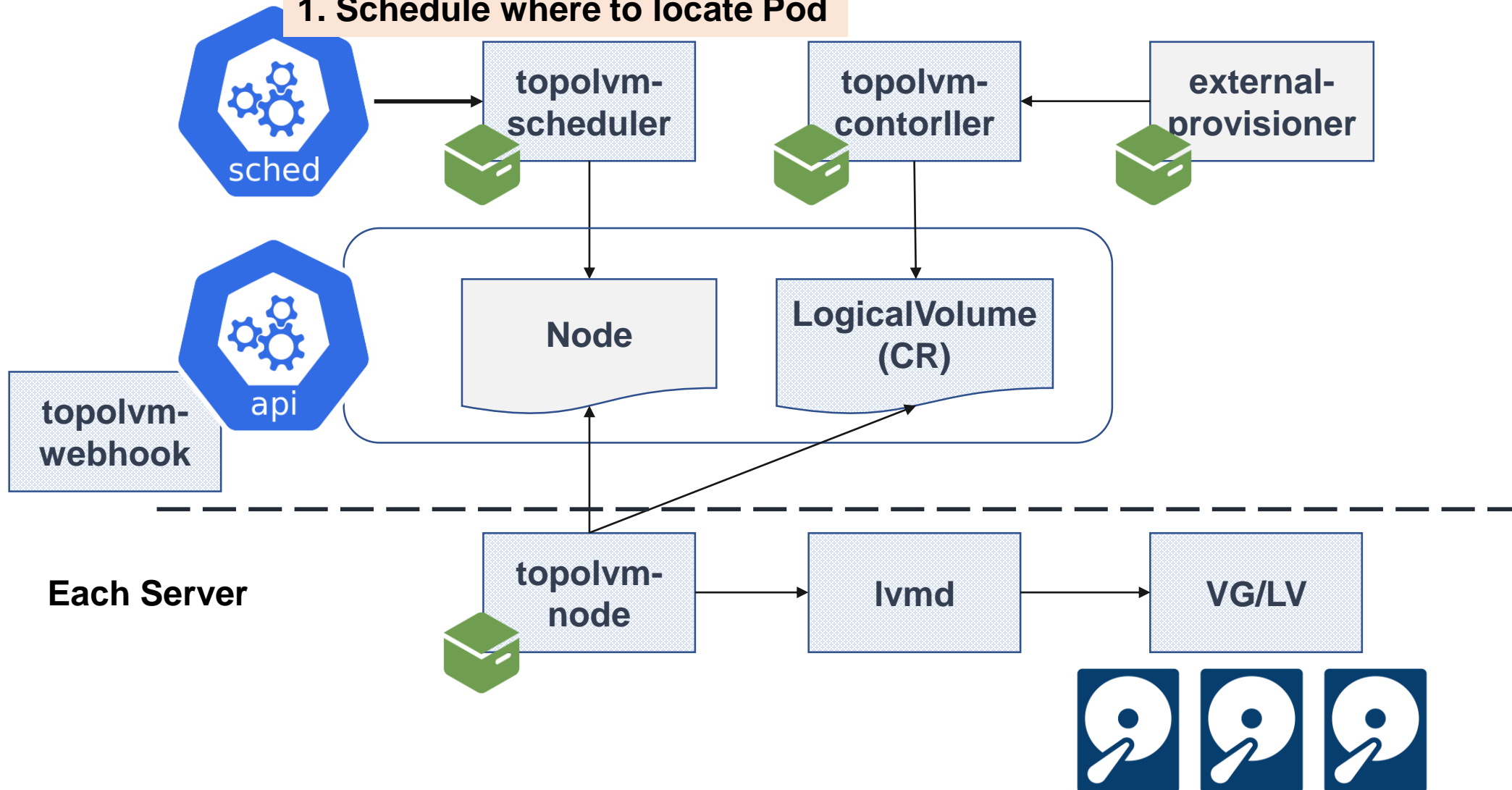


# Sequence of Provisioning



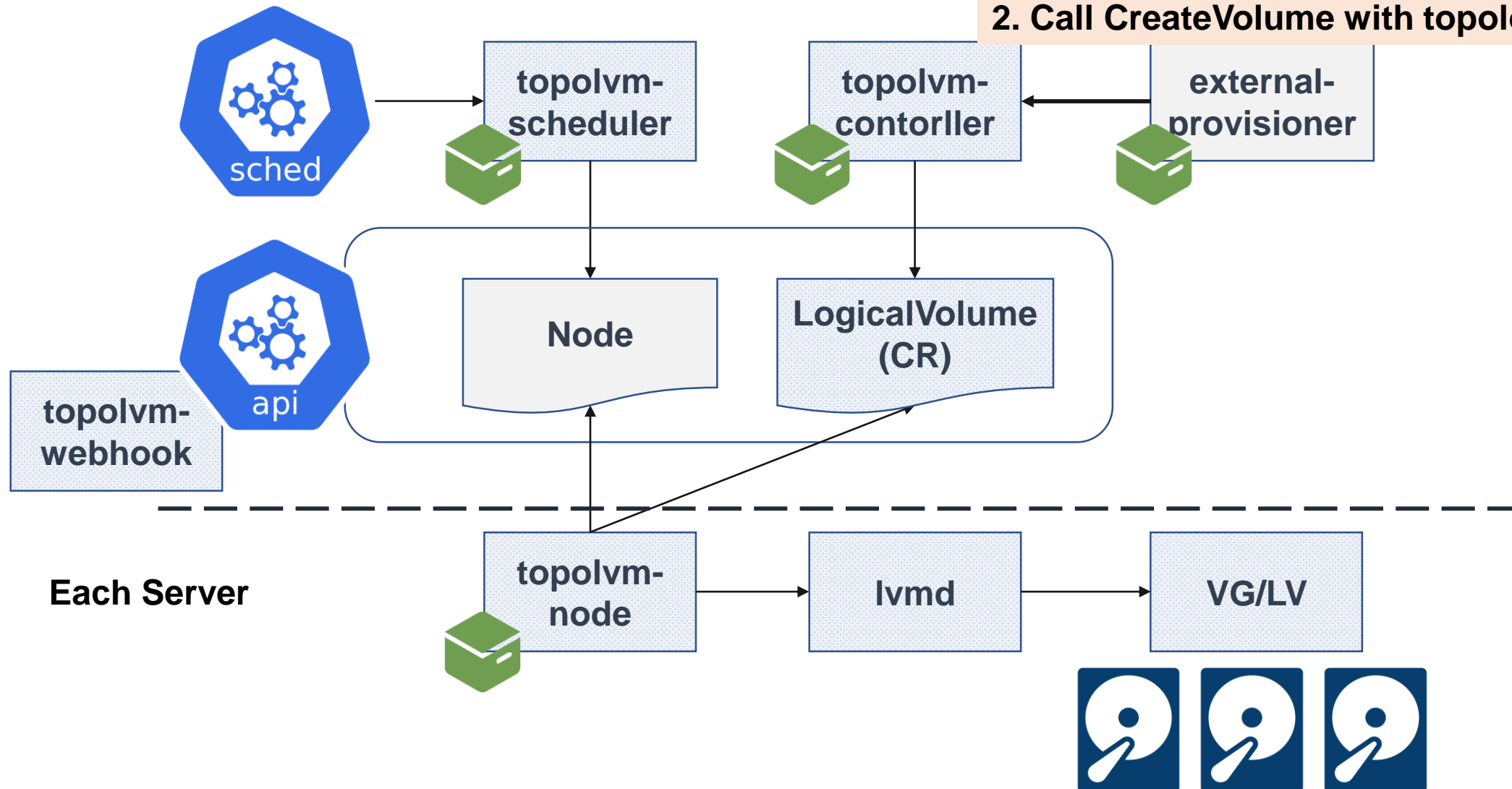
# Sequence of Provisioning

## 1. Schedule where to locate Pod

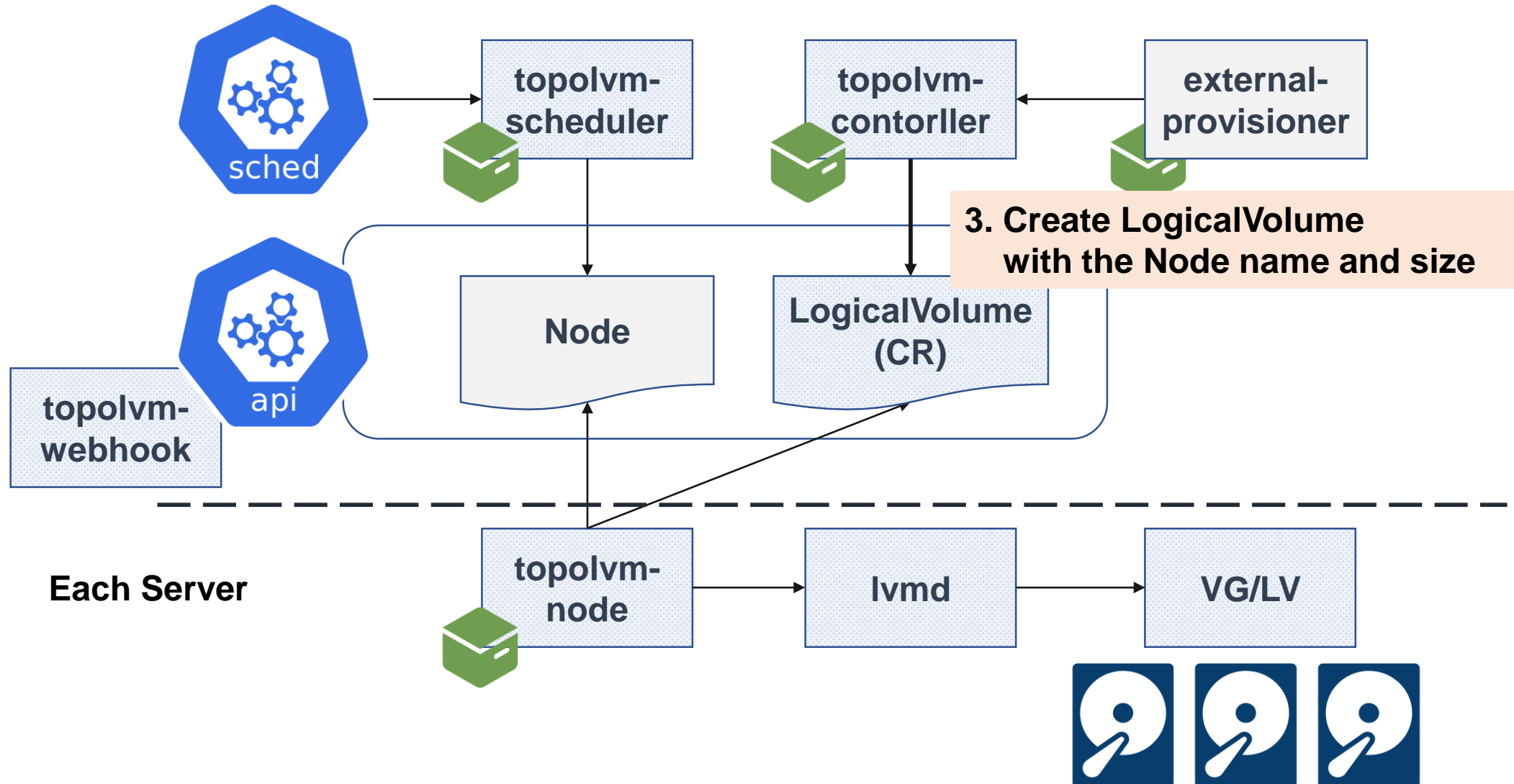


# Sequence of Provisioning

## 2. Call CreateVolume with topology key

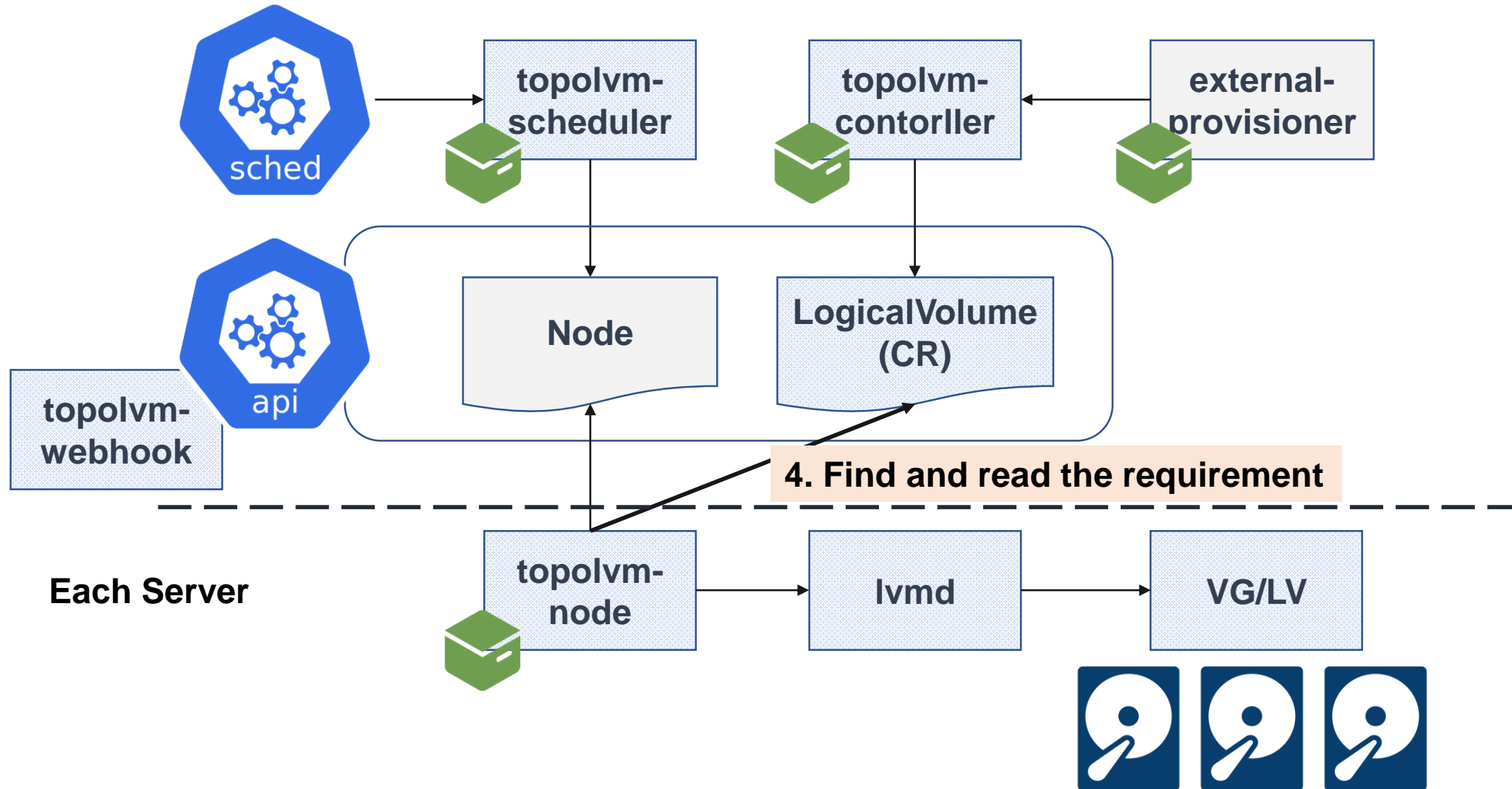


# Sequence of Provisioning

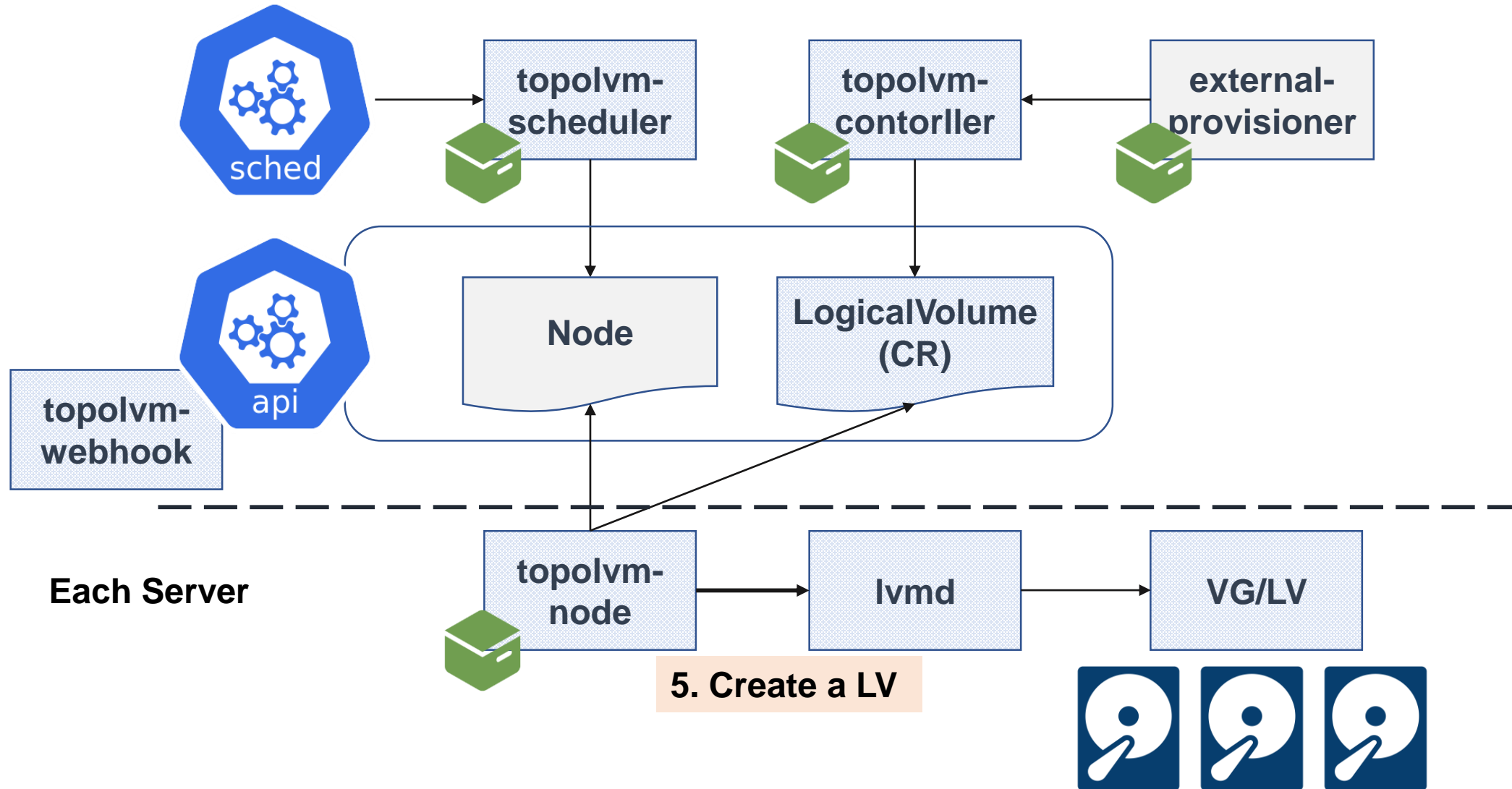




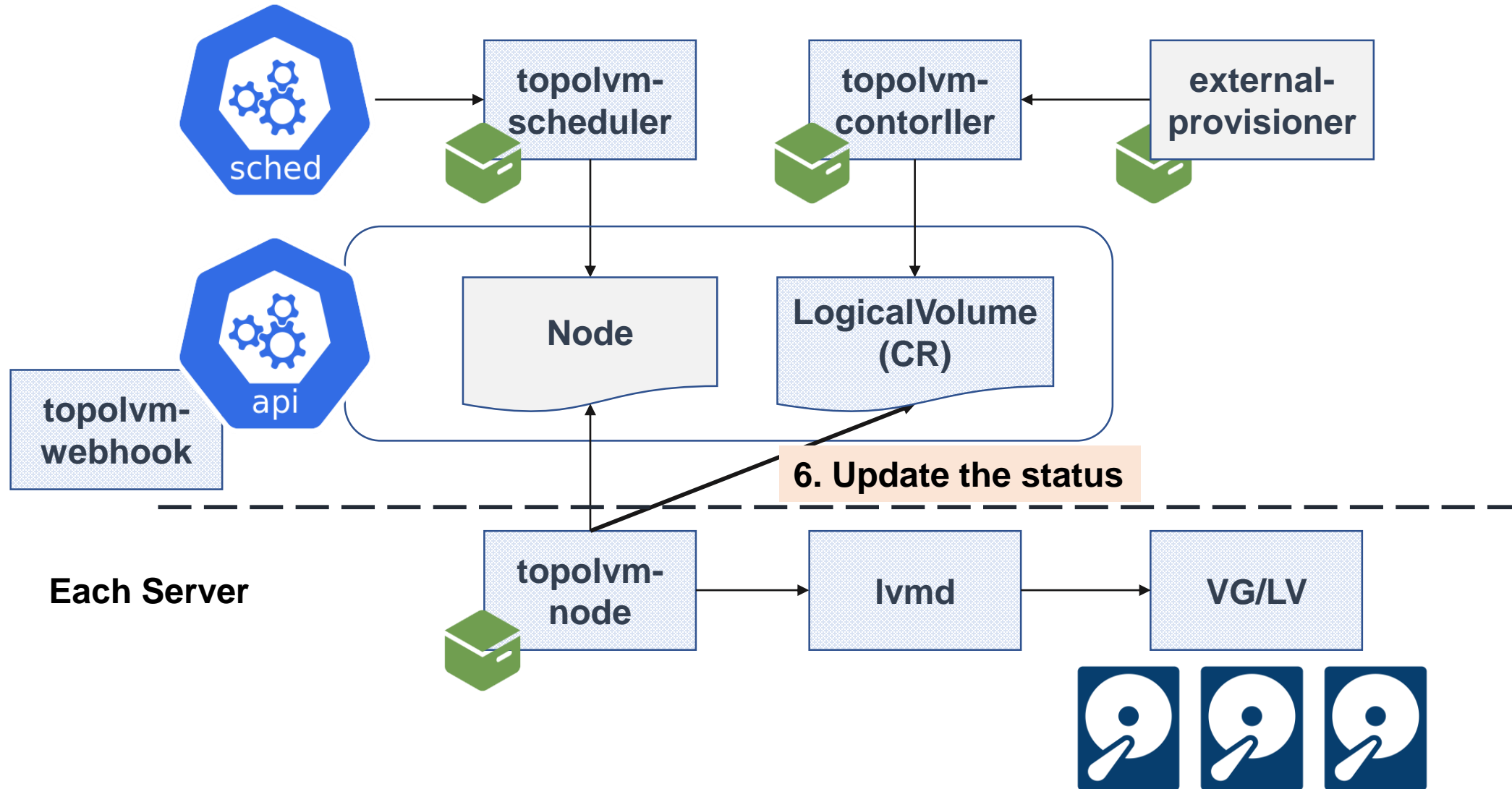
# Sequence of Provisioning



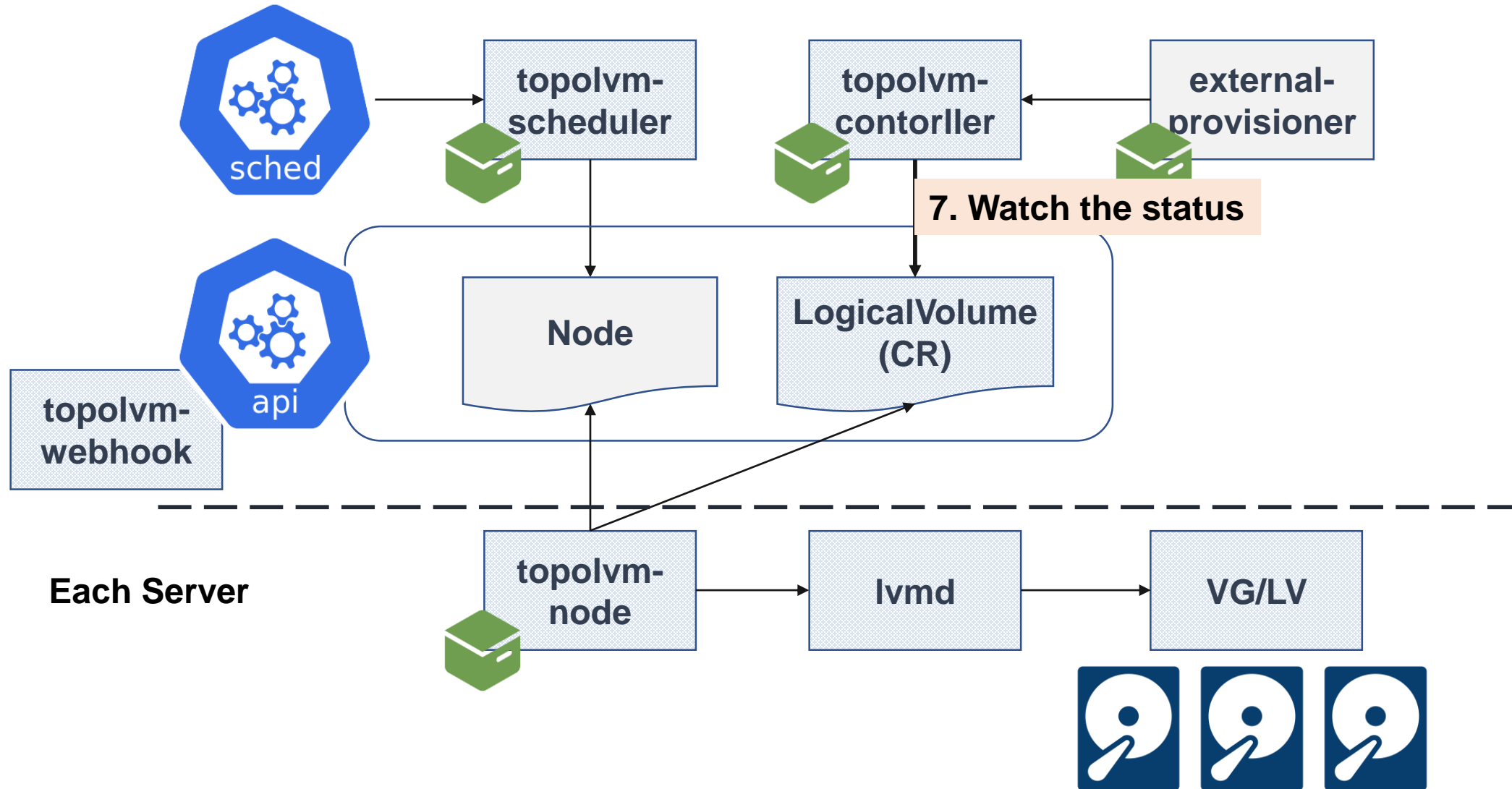
# Sequence of Provisioning



# Sequence of Provisioning

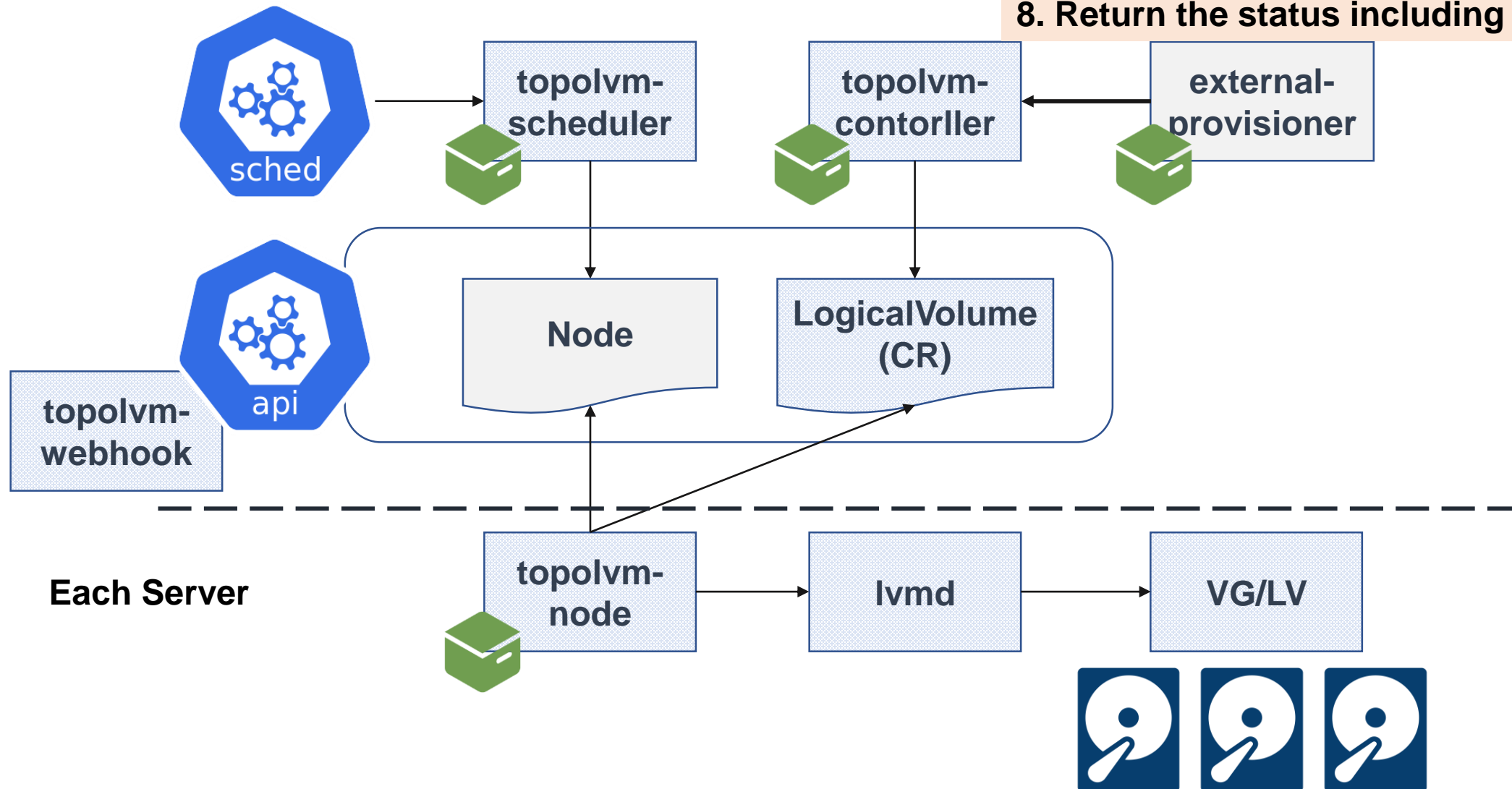


# Sequence of Provisioning

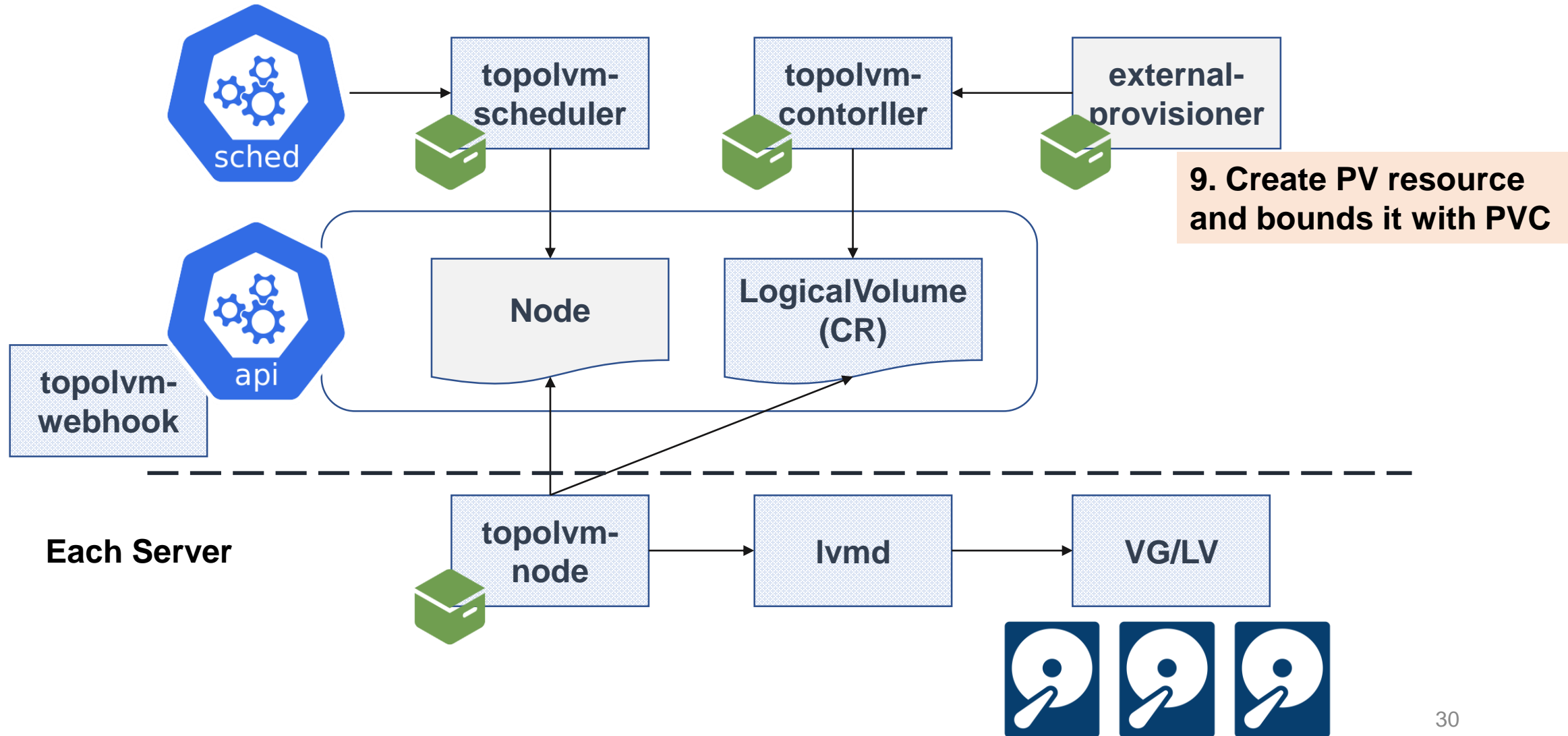


# Sequence of Provisioning

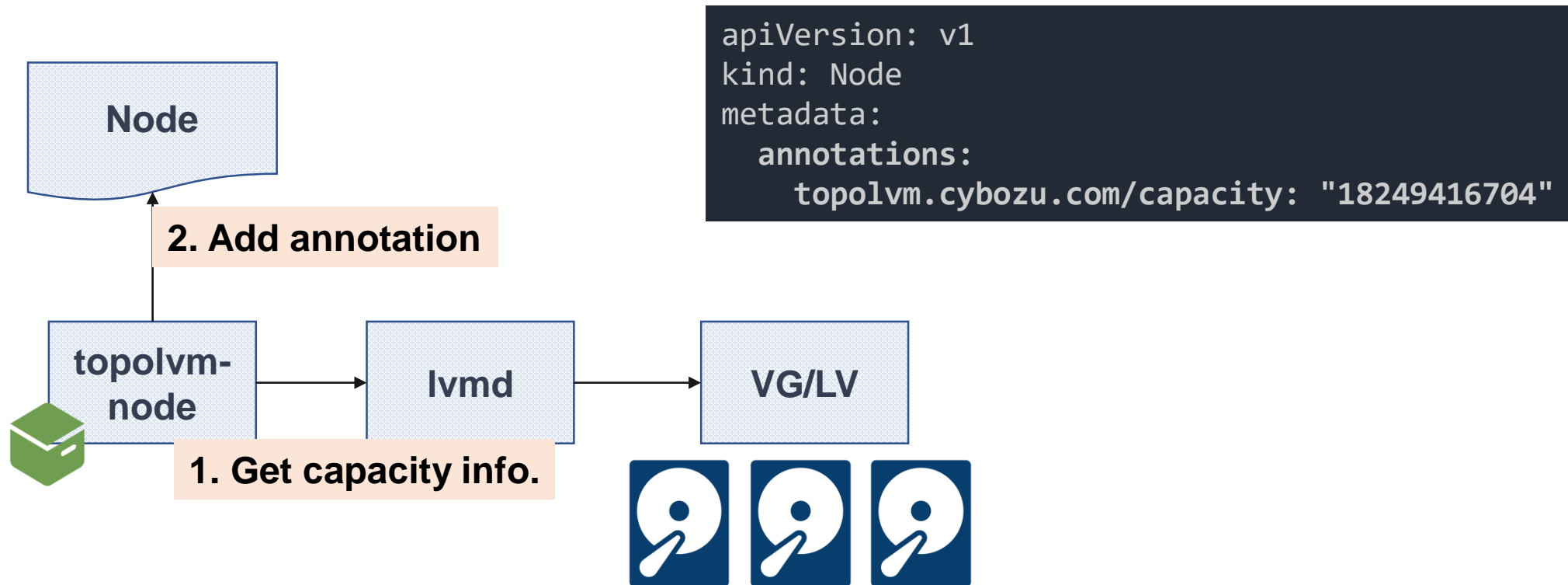
## 8. Return the status including LV info



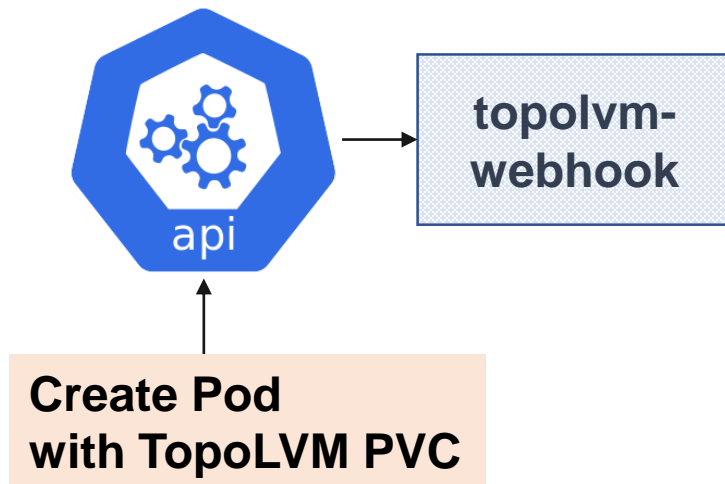
# Sequence of Provisioning



- Node Annotation by topolvm-node



- Mutating Pod by admission webhook



```
apiVersion: v1
kind: Pod
name: my-pod
namespace: default
spec:
  containers:
  - name: ubuntu
    resources:
      limits:
        topolvm.cybozu.com/capacity: "1073741824"
      requests:
        topolvm.cybozu.com/capacity: "1073741824"
```

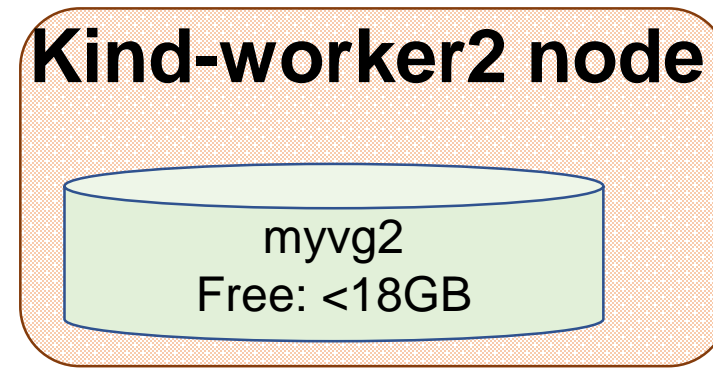
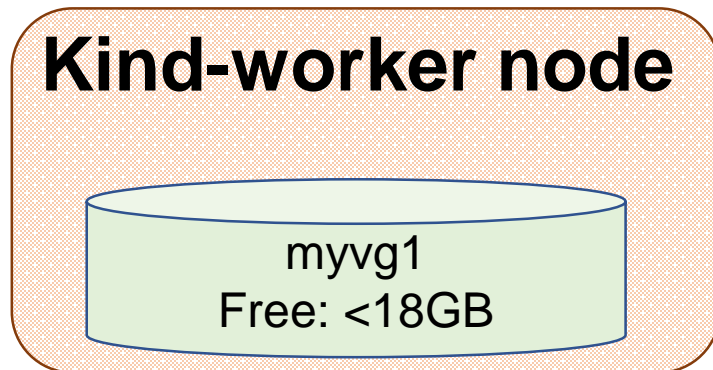




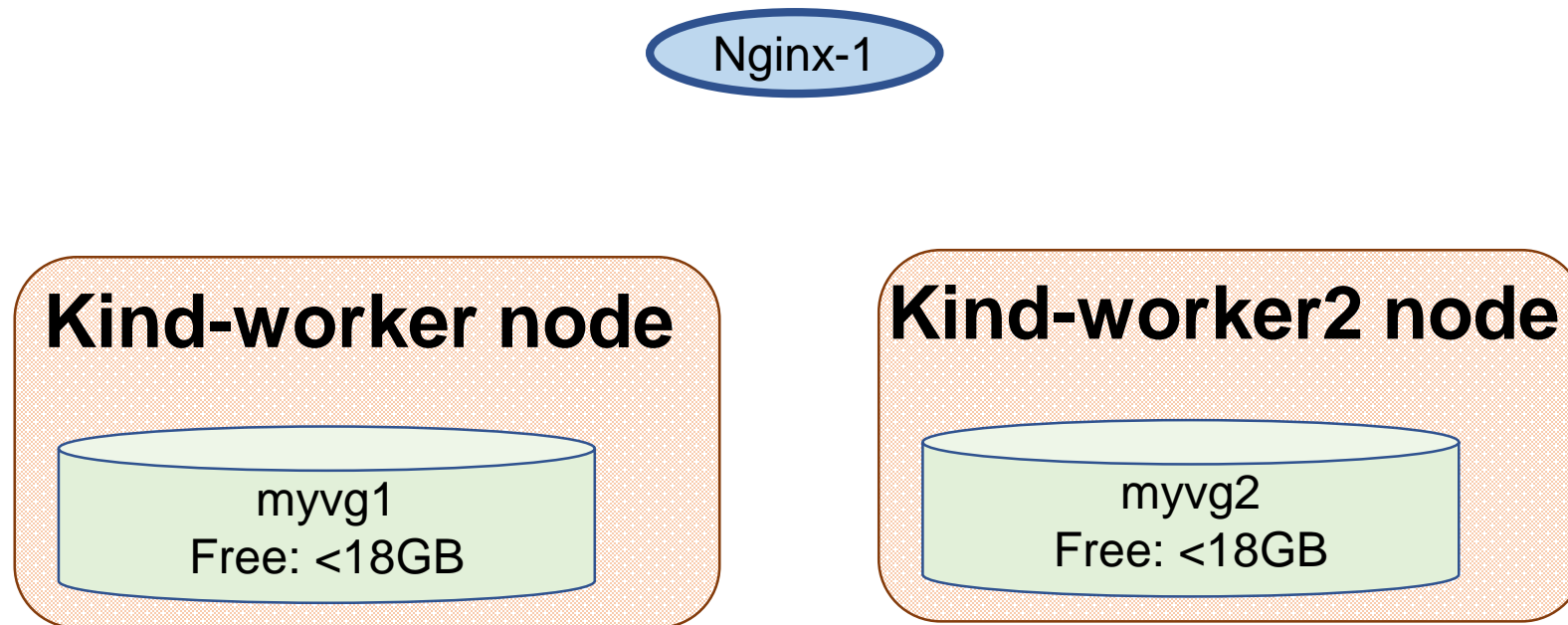


- **DOES NOT** provide specific redundancy
  - Because volumes are just located on the local disks
  - **Each application must be redundant itself**

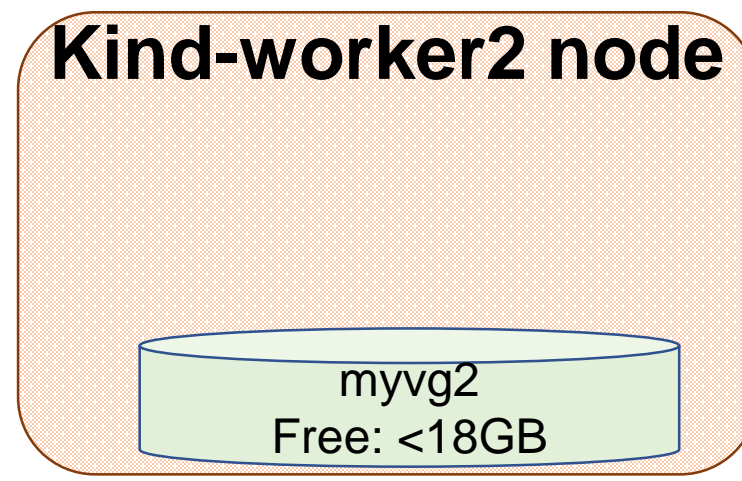
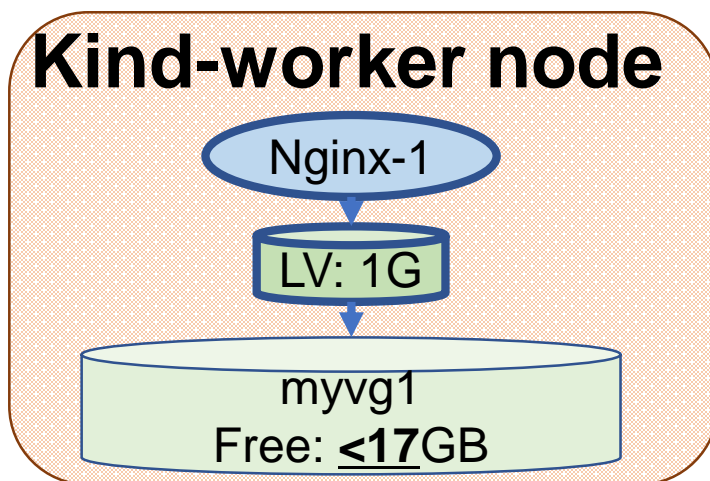
- **Features to be introduced**
  1. **Dynamic volume provisioning**
  2. **Capacity-aware Scheduling**
  3. **Online volume resizing**
- **Software and hardware configuration**



- **Schedule a pod (nginx-1)**
  - **Use a 1GiB volume**

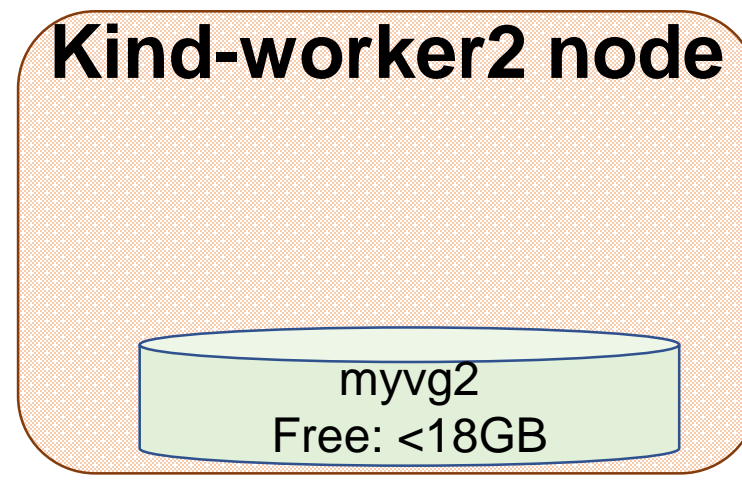
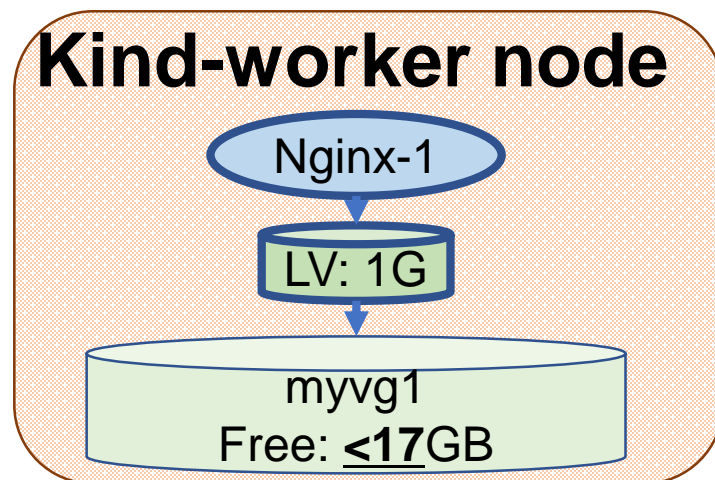


- **Expected result**
  - **PV is created dynamically**

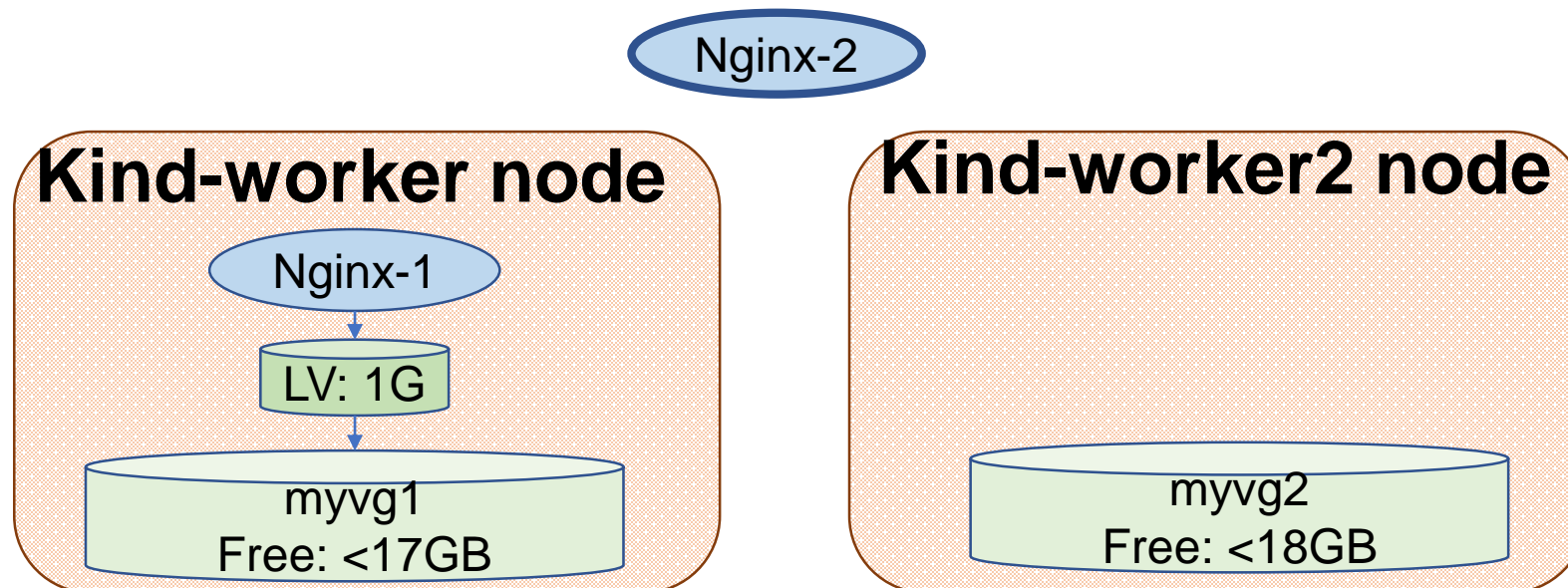


## ■ Actual result

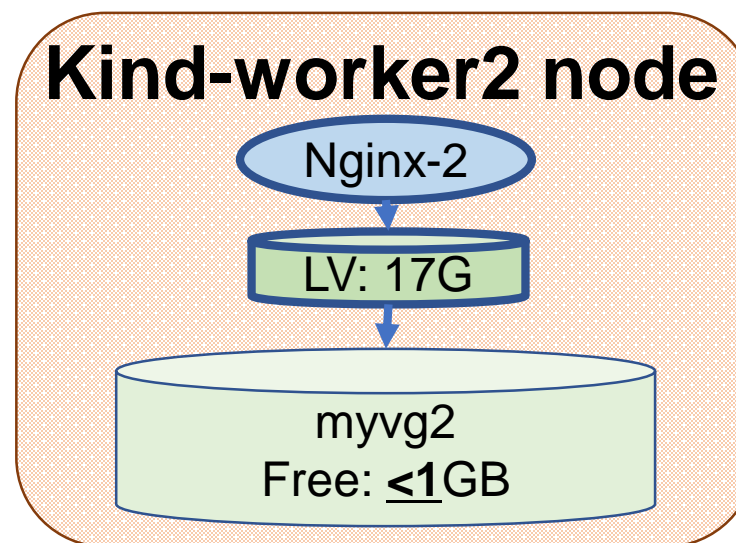
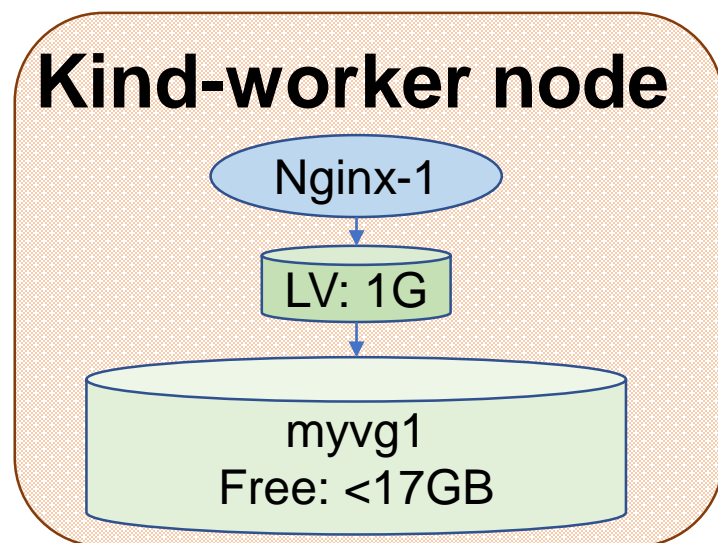
- ☑ PV is created dynamically



- Preparation: Exhaust kind-worker2's VG
- Schedule a pod (nginx-2) to kind-worker2
  - Use a 17 GB volume

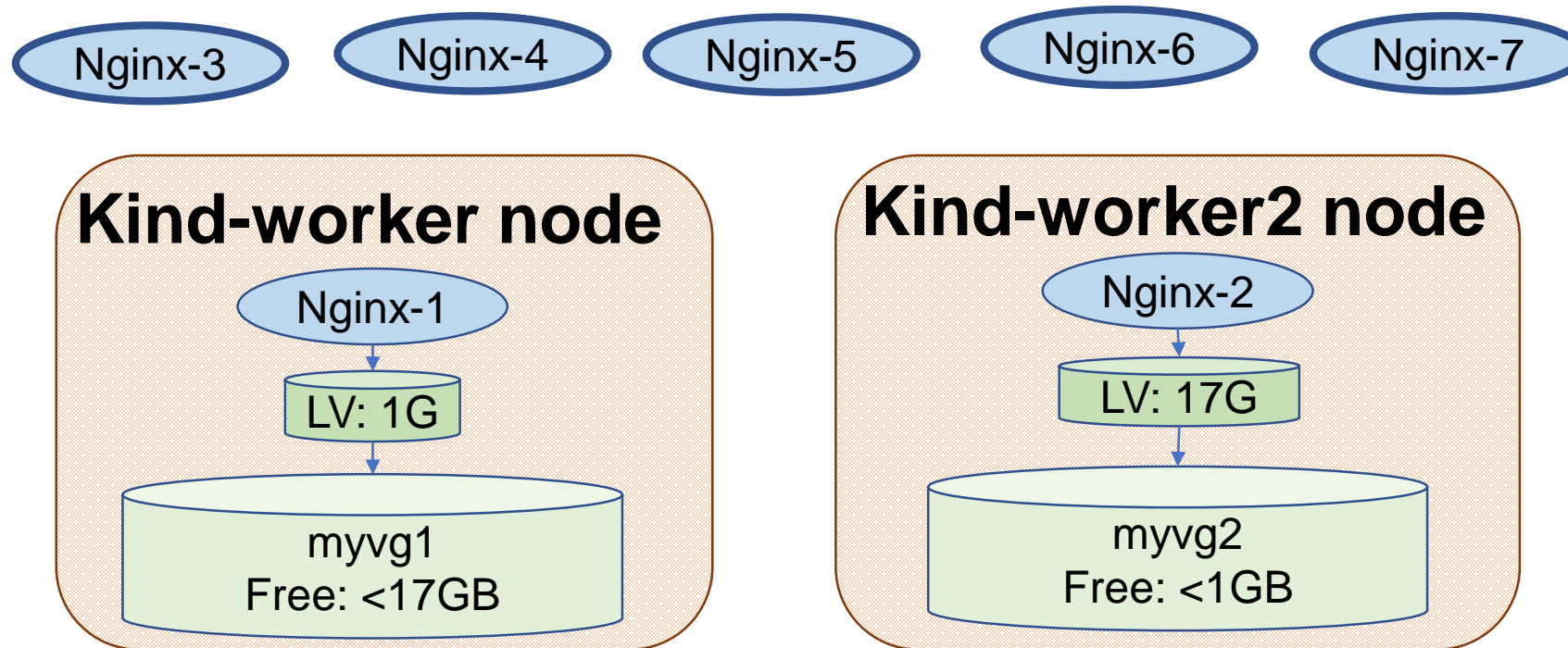


- **Kind-worker2's capacity is under 1GiB**



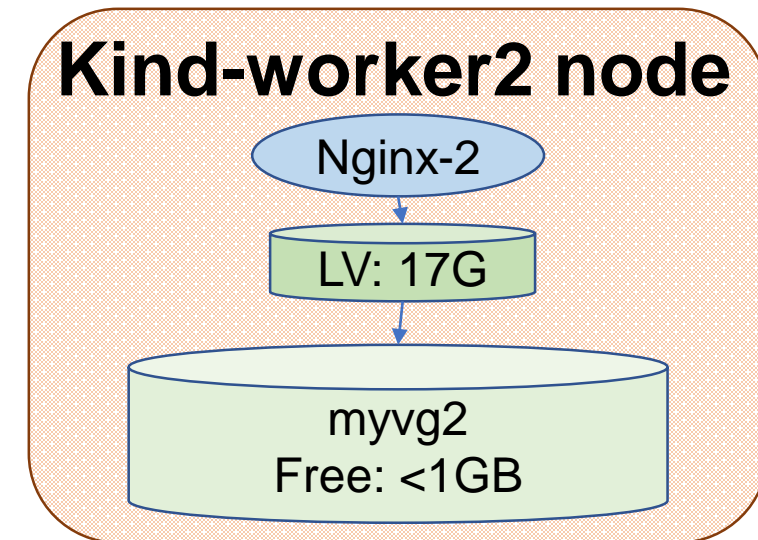
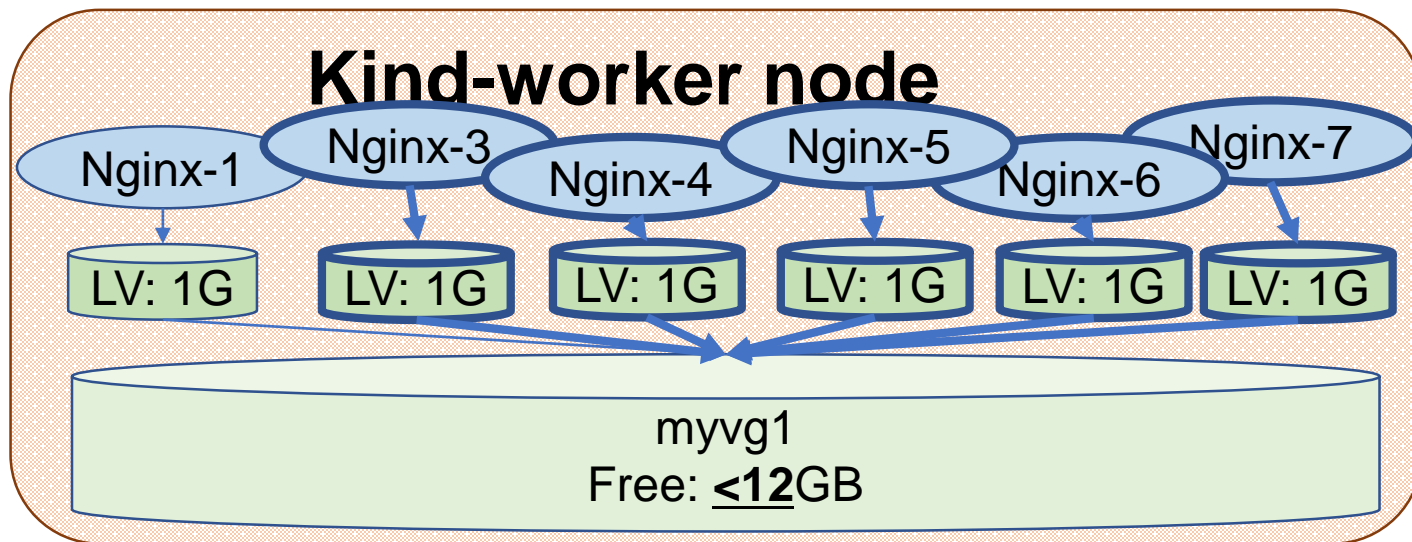


- Schedule many pods (nginx-[3-7])
- Use a 1GB volume



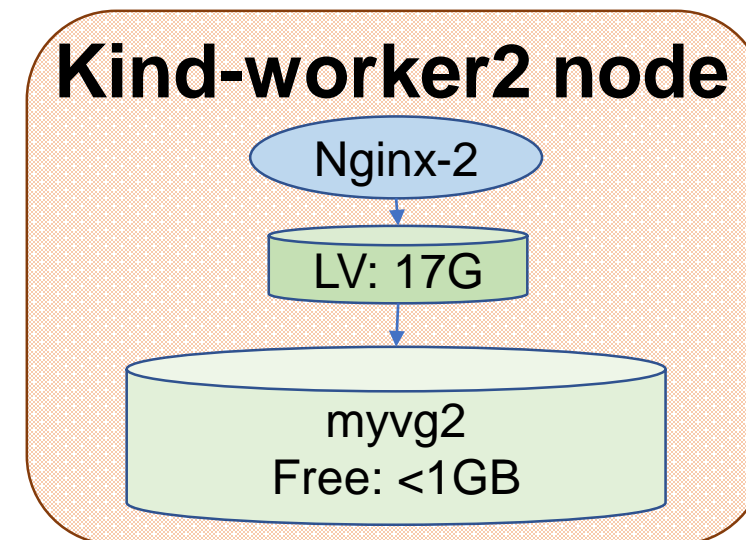
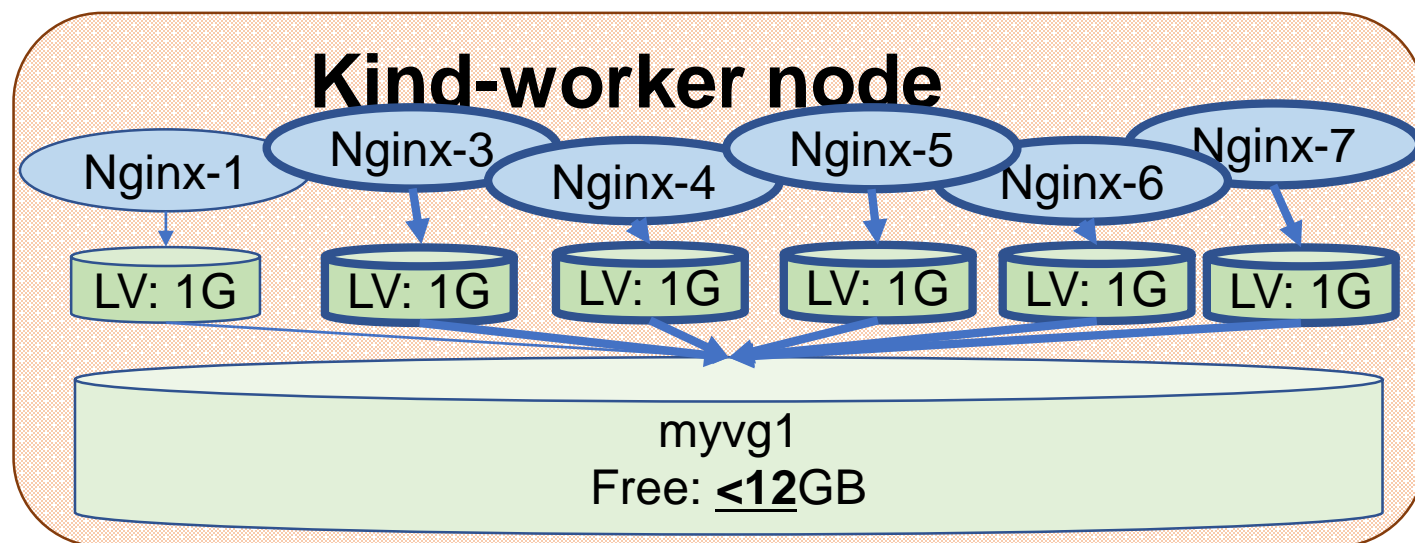
## ■ Expected result

- All pods are scheduled to kind-worker

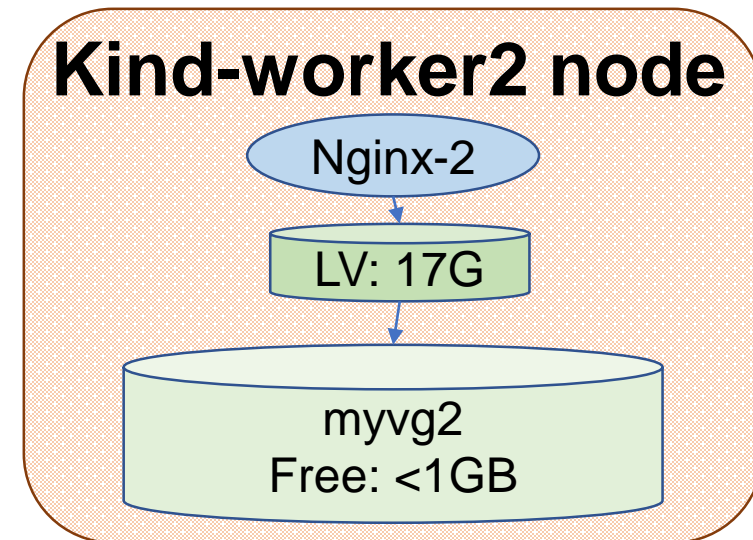
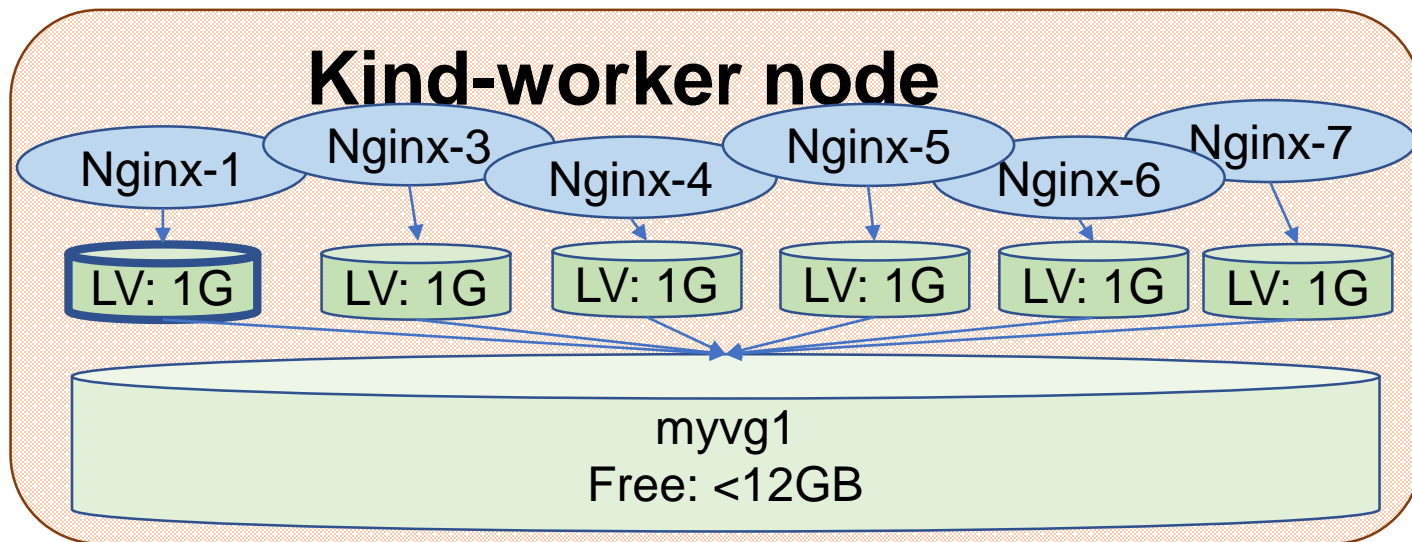


## ■ Actual result

- ☑ All pods are scheduled to kind-worker



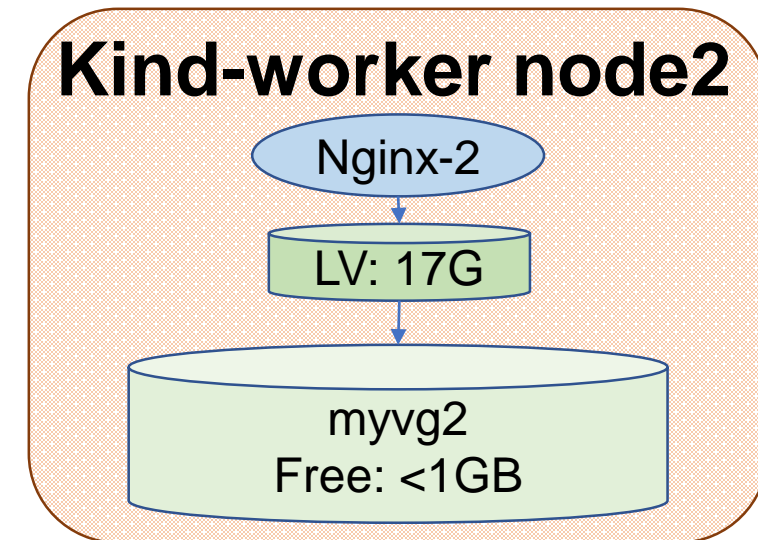
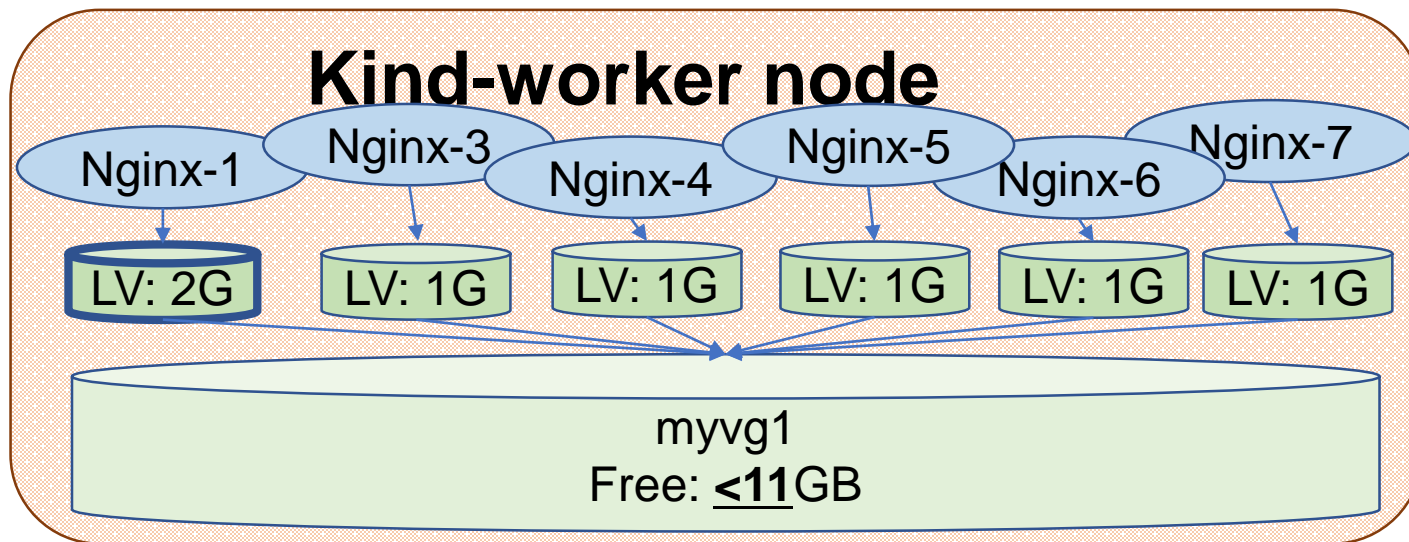
## ■ Expand nginx-1's volume to 2GiB



## ■ Expected result

□ Topo-pvc1 is resized

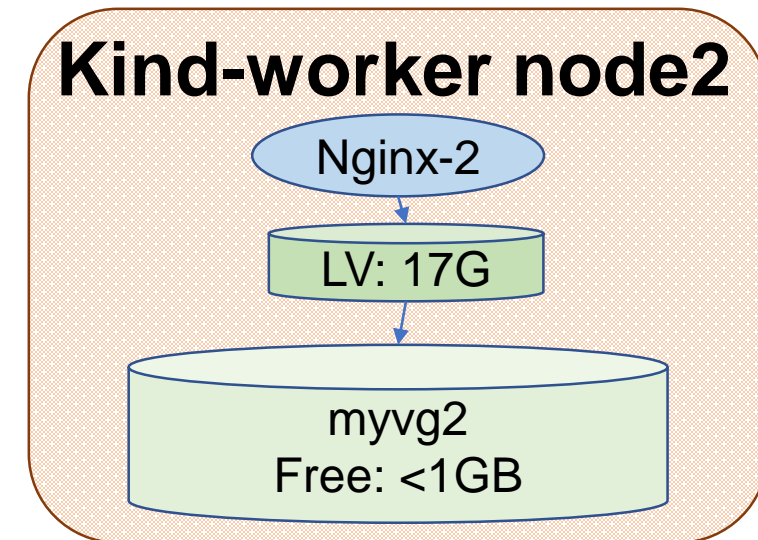
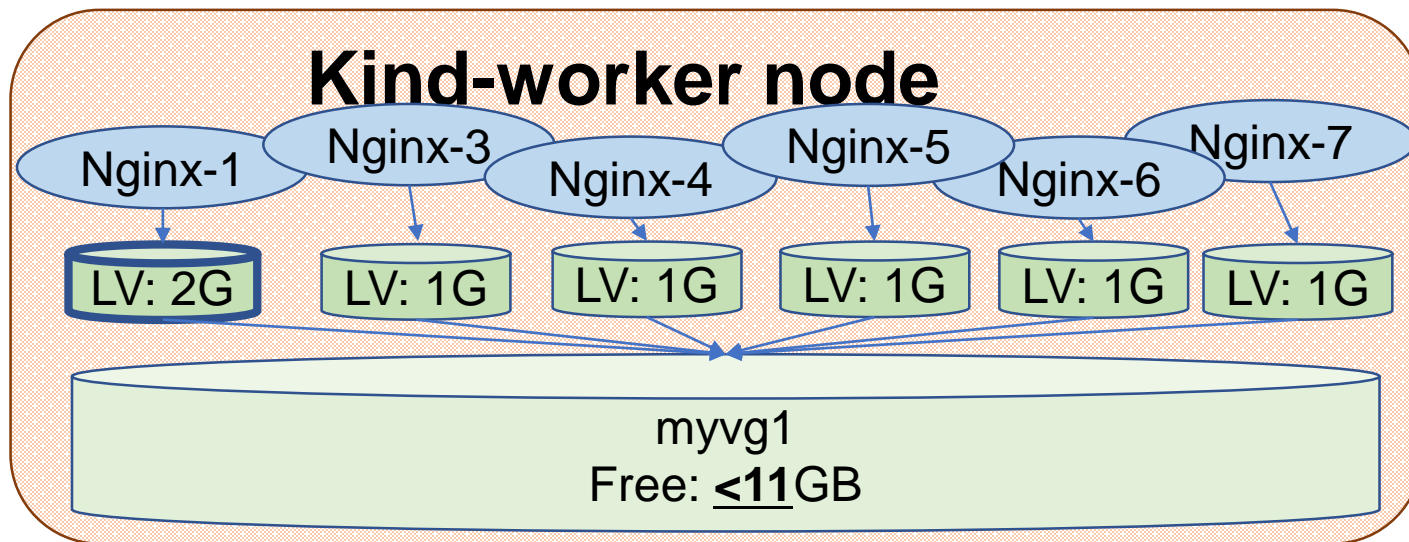
□ The corresponding filesystem is resized



## ■ Actual result

☑ Topo-pvc1 is resized

☑ The corresponding filesystem is resized



- **TopoLVM is a local storage dynamic provisioner based on LVM**
- **Enable capacity-aware Pod scheduling based on local storage**
- **Continue to develop targeting production use!**

## ■ GitHub

- <https://github.com/topolvm/topolvm>

- Including the manifests for practical deployment

## ■ Slack

- Please join from the invitation at README.md

## ■ Blog

- <https://blog.kintone.io/entry/topolvm>





KubeCon



CloudNativeCon

Europe 2020

*Virtual*

Thank You!  
and Any Questions?



KubeCon



CloudNativeCon

Europe 2020



*Virtual*



KEEP CLOUD NATIVE

CONNECTED

