ceph

Cephadm at Scale for Detractors: Why it's time to reconsider containerized Ceph deployments

Ernesto Puerta - IBM

Ernesto Puerta

epuertat@ibm.com

- Principal Software Eng. @ **IBM**
- **25 years'** experience (Telefonica, Alcatel-Lucent, Bell Labs, Nokia, Red Hat)
- **10 years** with Ceph (user & developer)
- Former **Ceph Dashboard** Lead (2020-2023)

My First Encounter with Ceph

2014, peacefully working at **ACME**...

... **Evil Corp** acquires **Small Storage HW Start-up**, provider of a core component of **ACME's star product**.

- Given 6 months to find a replacement for a **"99.999%-uptime <1ms-latency CDN-based video recording & streaming platform"**... Peanut budget (~~paid solutions~~). BTW, reuse legacy SGI servers.
- We explored the FOSS storage landscape: **NFS, Lustre, GlusterFS, Ceph**, ...
- Finally chose **Ceph (Hammer-Infernalis)**, and made all possible mistakes:
  - **Many small (5-node) clusters with bulky nodes (64 x 6 TB HDDs),**
  - **EC for read-intensive workloads**,
  - **Custom librados object-filesystem over HTTP** (Cassandra for object-dir grouping) instead of radosgw or cephfs,
  - **Über-finetuning**: sysctl hacks, ethtool, NUMA/IRQ affinity, xfs nobarrier, SR-IOV, undocumented kernel/HBA/NIC driver/ flags, ...
  - **Custom Ansible** deployer & upgrader,

It was a complete disaster, but I ended up ❤️ Ceph.

My First Encounter with Ceph

2014, peacefully working at **ACME**...

... **Evil Corp** acquires **Small Storage HW Start-up**, provider of a core component of **ACME's star product**.

- Given 6 months to find a replacement for a **"99.999%-uptime <1ms-latency CDN-based video recording & streaming platform"**... Peanut budget (~~paid solutions~~). BTW, reuse legacy SGI servers.
- We explored the FOSS storage landscape: **NFS, Lustre, GlusterFS, Ceph**, ...
- Finally chose **Ceph (Hammer-Infernalis)**, and made all possible mistakes:
  - **Many small (5-node) clusters with bulky nodes (64 x 6 TB HDDs),**
  - **EC for read-intensive workloads**,
  - **Custom librados object-filesystem over HTTP** (Cassandra for object-dir grouping) instead of radosgw or cephfs,
  - **Über-finetuning**: sysctl hacks, ethtool, NUMA/IRQ affinity, xfs nobarrier, SR-IOV, undocumented kernel/HBA/NIC driver/ flags, ...
  - **Custom Ansible** deployer & upgrader,

It was a complete disaster, but I ended up ❤️ Ceph.

A year later, the **project was shut down and everyone was fired**...

My First Encounter with Ceph

2014, peacefully working at **ACME**...

... **Evil Corp** acquires **Small Storage HW Start-up**, provider of a core component of **ACME's star product**.

- Given 6 months to find a replacement for a **"99.999%-uptime <1ms-latency CDN-based video recording & streaming platform"**... Peanut budget (~~paid solutions~~). BTW, reuse legacy SGI servers.
- We explored the FOSS storage landscape: **NFS, Lustre, GlusterFS, Ceph**, ...
- Finally chose **Ceph (Hammer-Infernalis)**, and made all possible mistakes:
  - **Many small (5-node) clusters with bulky nodes (64 x 6 TB HDDs),**
  - **EC for read-intensive workloads**,
  - **Custom librados object-filesystem over HTTP** (Cassandra for object-dir grouping) instead of radosgw or cephfs,
  - **Über-finetuning**: sysctl hacks, ethtool, NUMA/IRQ affinity, xfs nobarrier, SR-IOV, undocumented kernel/HBA/NIC driver/ flags, ...
  - **Custom Ansible** deployer & upgrader,

It was a complete disaster, but I ended up ❤️ Ceph.

A year later, the **project was shut down and everyone was fired**... but still ❤️ Ceph.

Agenda

- **Deploying** Ceph
- Cephadm for **Detractors**
  - **Intro** to Cephadm
  - Container **Myths**
- Managing **XXL Clusters**

ceph

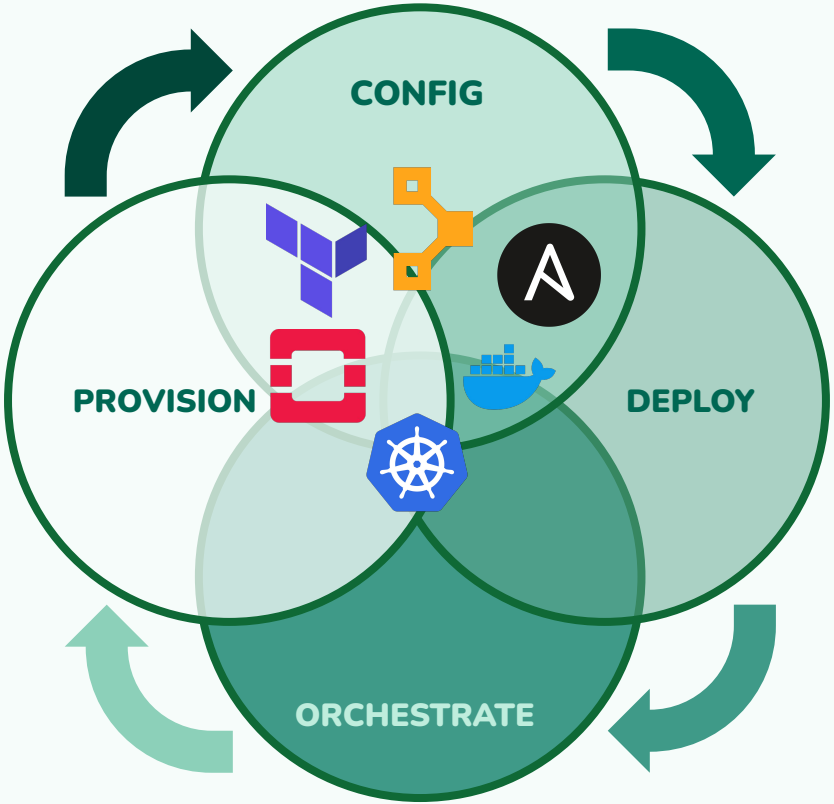
Deploying Ceph...

What I mean when I talk about Deployment

ceph

Day 0

Infrastructure
Resources
Hardware
VMs...



Day 1

System
Network
Security
Policies...

Performance
Logs
Metrics
Traces...

Day N

Releases
Services
Scale
Availability...

Day 2

In the Beginning... Was the Manual Deployment

Ceph - v0.1
Manual deployment
mkcephfs

2008

In the Beginning... Was the Manual Deployment

Ceph - v0.1
Manual deployment
mkcephfs

2008

```
podman run -ti ubuntu:14.04 bash
> apt-get update
> apt-get install -y curl automake make gcc g++-4.6 pkg-config libtool uuid-dev libkeyutils-dev
libcrypto++-dev libfuse-dev libedit-dev libboost-all-dev
> curl -L https://github.com/ceph/ceph/archive/refs/tags/v0.1.tar.gz -qo- | tar xvzf -
> cd ceph-01/
> ./autogen.sh
> CXX="/usr/bin/g++-4.6" CXXFLAGS="-w -pthread" LIBS="-lboost_system" ./configure
--without-tcmalloc --without-libatomic-ops
> make
...
make: *** [all-recursive] Error 1
```

In the Beginning... Was the Manual Deployment

Ceph - v0.1
Manual deployment
mkcephfs

2008

```
podman run -ti ubuntu:14.04 bash
> apt-get update
> apt-get install -y curl automake make gcc g++-4.6 pkg-config libtool uuid-dev libkeyutils-dev
libcrypto++-dev libfuse-dev libedit-dev libboost-all-dev
> curl -L https://github.com/ceph/ceph/archive/refs/tags/v0.40.tar.gz -qo- | tar xvzf -
> cd ceph-01/
> ./autogen.sh
> CXX="/usr/bin/g++-4.6" CXXFLAGS="-w -pthread" LIBS="-lboost_system" ./configure
--without-tcmalloc --without-libatomic-ops
> make
> mkcephfs -a -c mycluster.conf -k mycluster.keyring
...
```

A Chef for Ceph

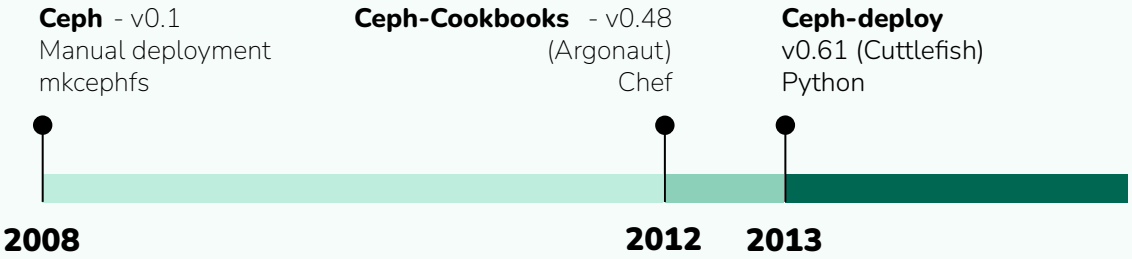
Ceph - v0.1
Manual deployment
mkcephfs

Ceph-Cookbooks - v0.48
(Argonaut)
Chef

2008

2012

Back to basics



The Year of the Deployers

Ceph - v0.1
Manual deployment
mkcephfs

Ceph-Cookbooks - v0.48
(Argonaut)
Chef

Ceph-deploy
v0.61 (Cuttlefish)
Python

Ceph-Ansible - v11.2.0 (Jewel)
Ansible Playbooks
DeepSea
Salt-based
Rook
Kubernetes

2008 **2012** **2013** **2016**

Timeline

Ceph - v0.1
Manual deployment
mkcephfs

Ceph-Cookbooks - v0.48
(Argonaut)
Chef

Ceph-deploy
v0.61 (Cuttlefish)
Python

Ceph-Ansible - v11.2.0 (Jewel)
Ansible Playbooks
DeepSea
Salt-based
Rook
Kubernetes

Cephadm v15
(Octopus)
Ceph-native container orchestrator

2008 **2012** **2013** **2016** **2020**

Not Invented Here:
Custom vs. Off-the-Shelf

Ceph - v0.1
Manual deployment
mkcephfs

Ceph-Cookbooks - v0.48
(Argonaut)
Chef

Ceph-deploy
v0.61 (Cuttlefish)
Python

Ceph-Ansible - v11.2.0 (Jewel)
Ansible Playbooks
DeepSea
Salt-based
Rook
Kubernetes

Cephadm v15
(Octopus)
Ceph-native container orchestrator

2008 **2012** **2013** **2016** **2020**

custom | OTS | custom | OTS | custom

Telluric Forces

Ceph - v0.1
Manual deployment
mkcephfs

Ceph-Cookbooks - v0.48
(Argonaut)
Chef

Ceph-deploy
v0.61 (Cuttlefish)
Python

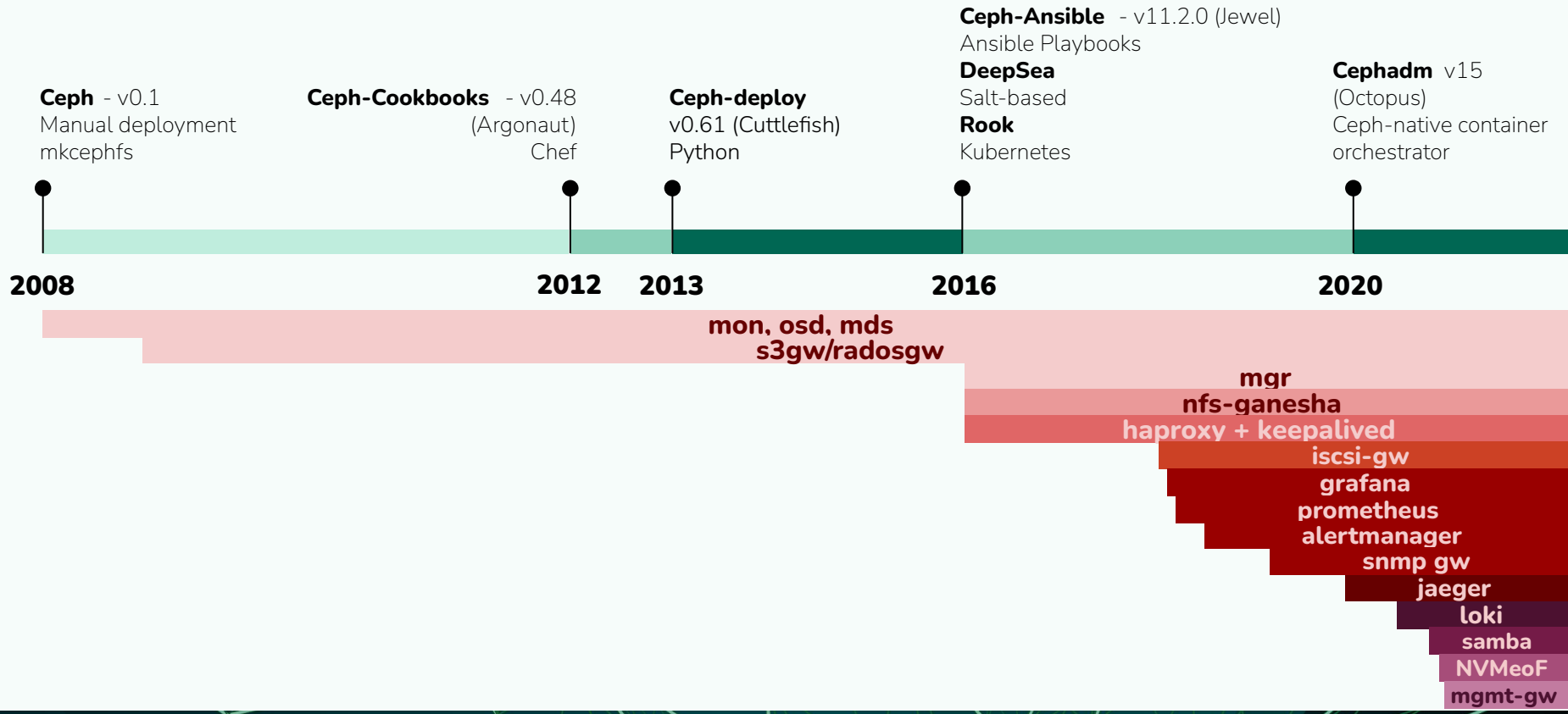
Ceph-Ansible - v11.2.0 (Jewel)
Ansible Playbooks
DeepSea
Salt-based
Rook
Kubernetes

Cephadm v15
(Octopus)
Ceph-native container orchestrator

2008 **2012** **2013** **2016** **2020**

sysv
systemd
distro packages (deb, rpm)
containers

The Cambrian Explosion... of Services



The (Deployment) Matrix

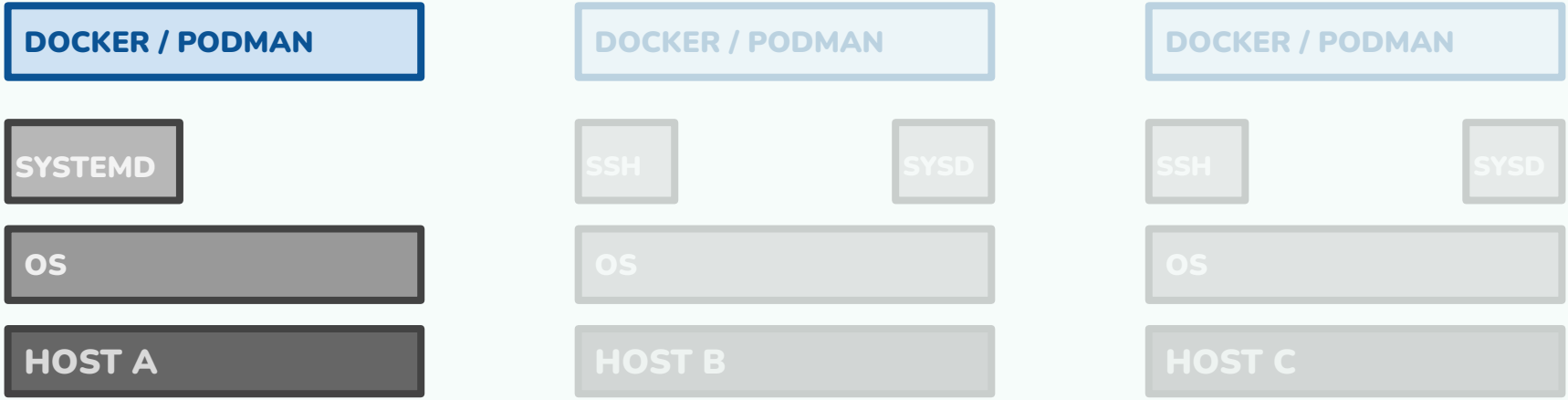
	mkcephfs †	Ceph-cookbook †	ceph-deploy †	Ceph-Ansible	DeepSea †	Rook	Cephadm
Who	Inktank	Inktank	Inktank	Red Hat	SUSE	CNCF-Ceph	**Ceph**
When	2008-2012	2012-2015	2013-2020	2016-	2016-2021	2016-	**2020-**
Tech	Bash	Chef / Ruby	Python	Ansible / Python	Salt	Golang / k8s	**Python**
Approach	Imperative	Imperative	Imperative	Declarative	Hybrid	Declarative	**Hybrid**
Model	Push (SSH)	Pull	Push (SSH)	Push (SSH)	Pull	Pull	**Push (SSH)**
License	LGPL-2.1	Apache-2.0	MIT	Apache-2.0	GPL-3.0	Apache-2.0	**LGPL-3.0**
Github	ceph/ceph/	ceph/ceph-cookbook	ceph/ceph-deploy	ceph/ceph-ansible	SUSE/DeepSea	rook/rook	ceph/ceph

ceph

Cephadm for Detractors: Intro to Cephadm

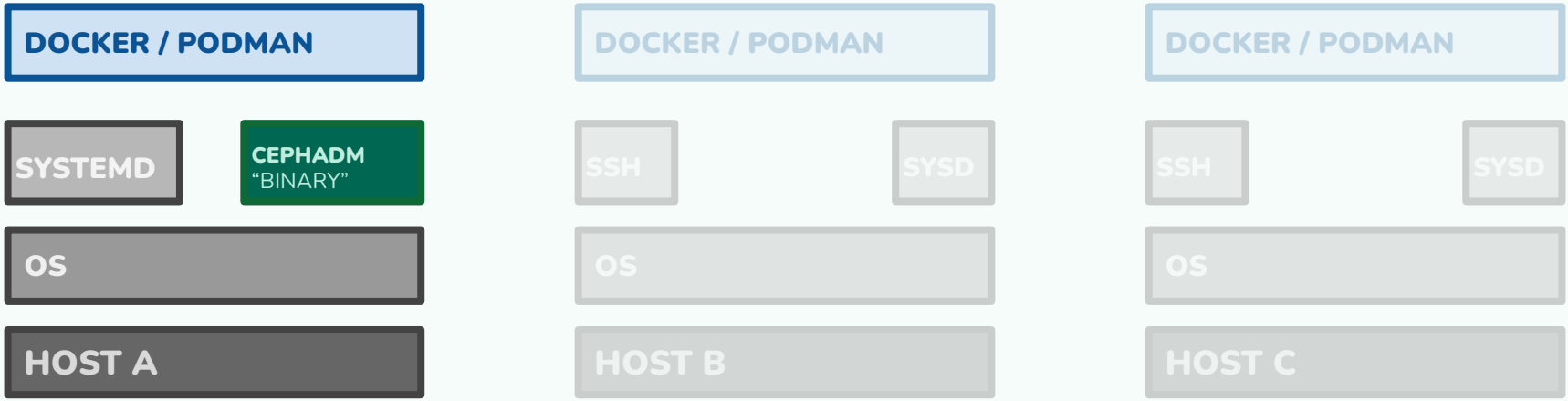
A Cephadm-managed Cluster

Day 1



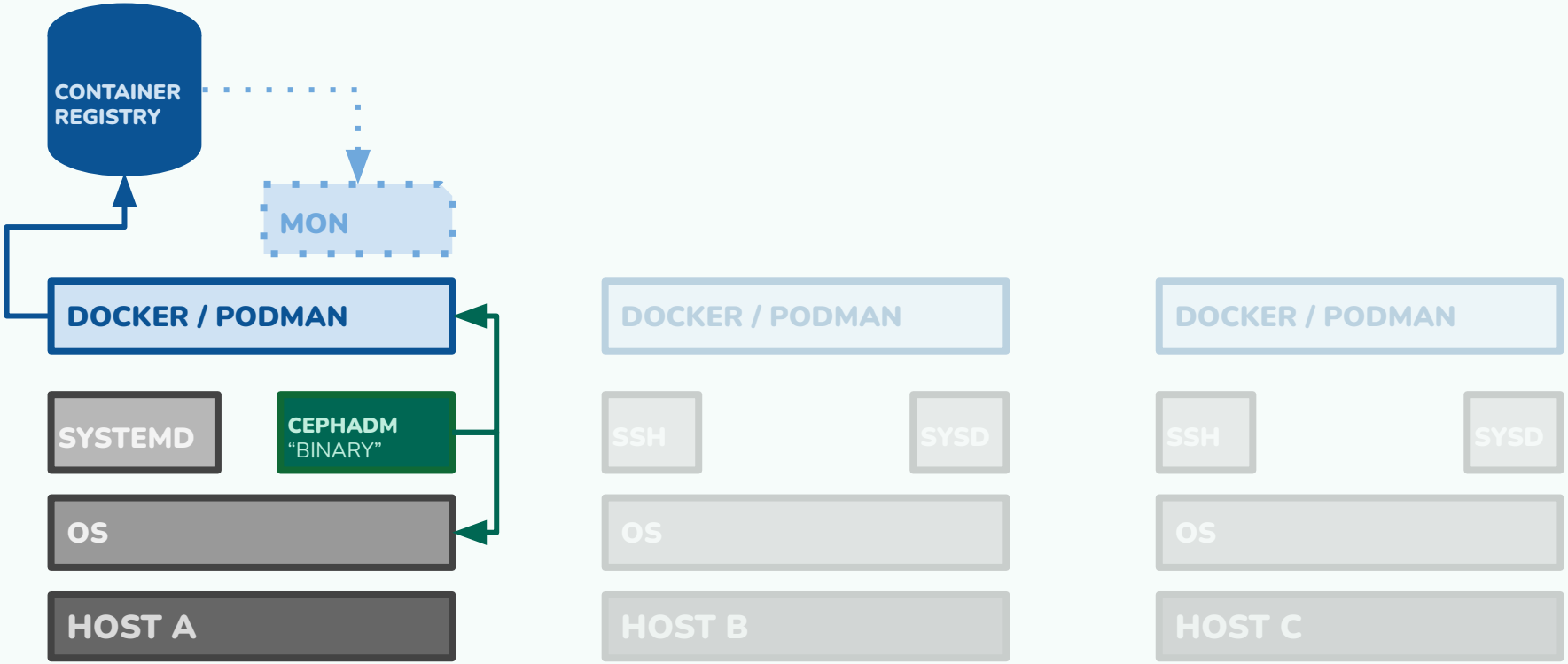
A Cephadm-managed Cluster

The “Binary”



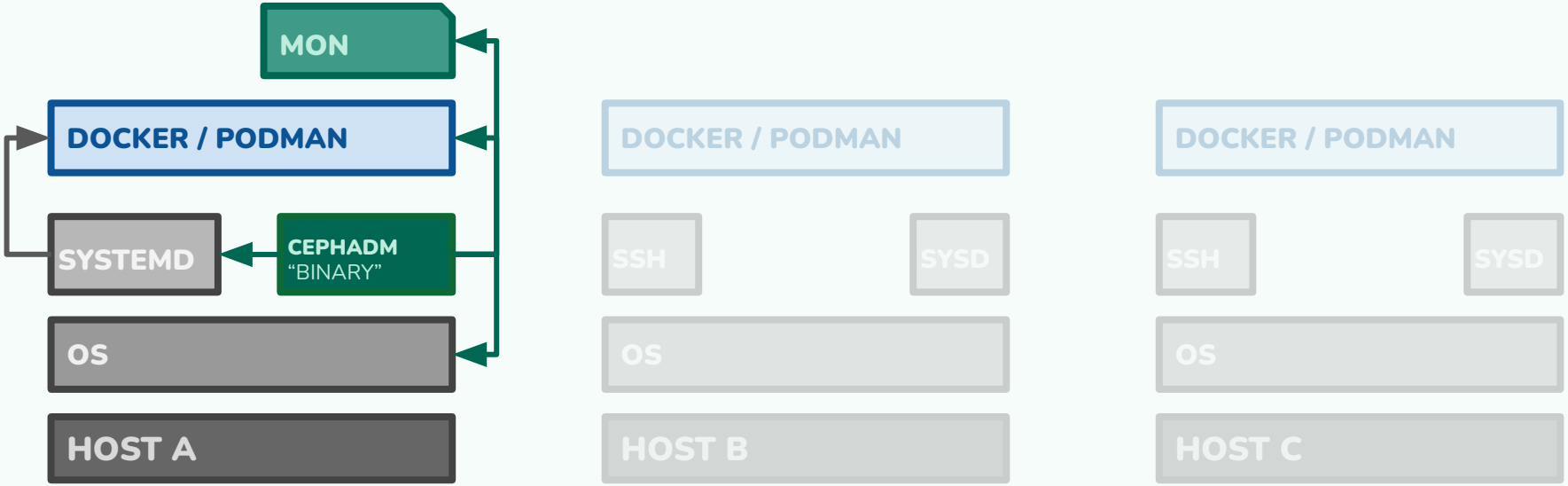
A Cephadm-managed Cluster

Bootstrap: Pulling Container Images...



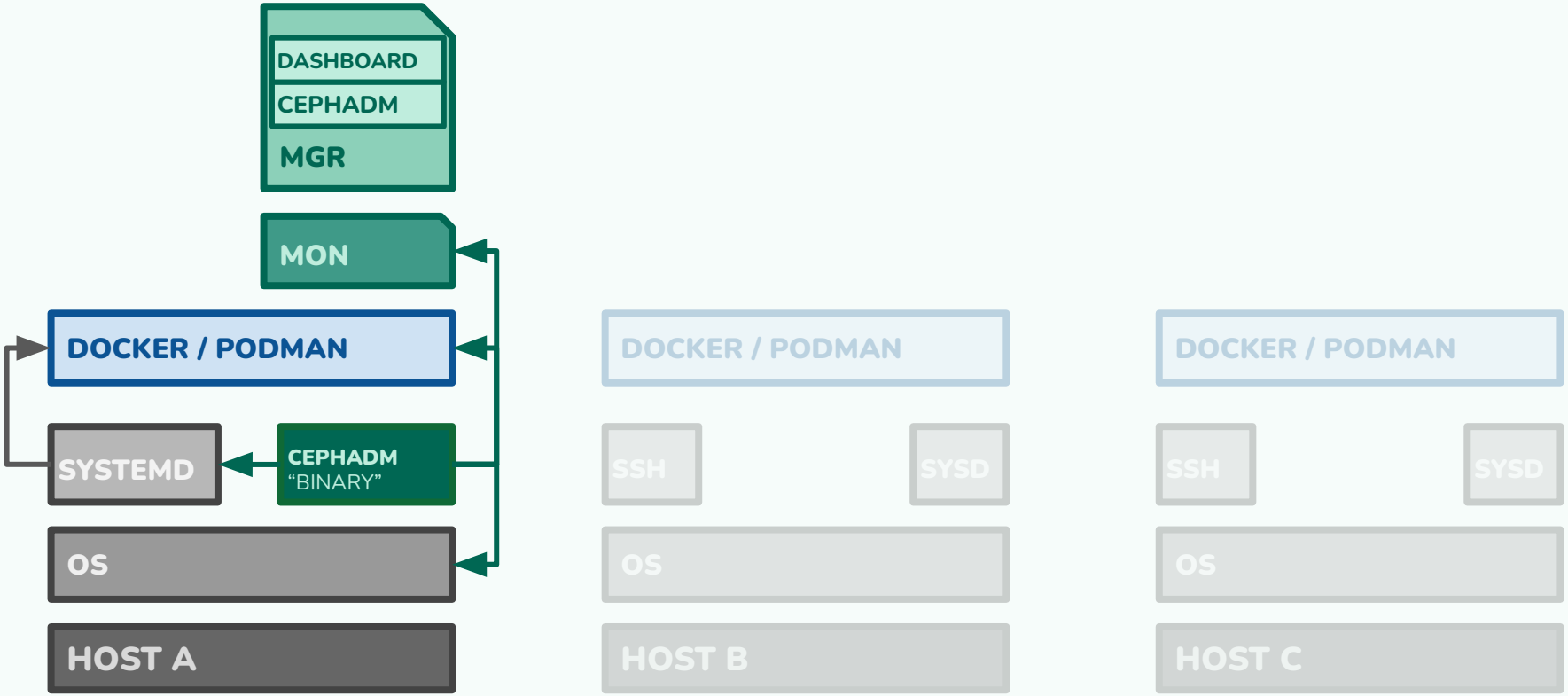
A Cephadm-managed Cluster

Bootstrap: Starting Ceph-Monitor...



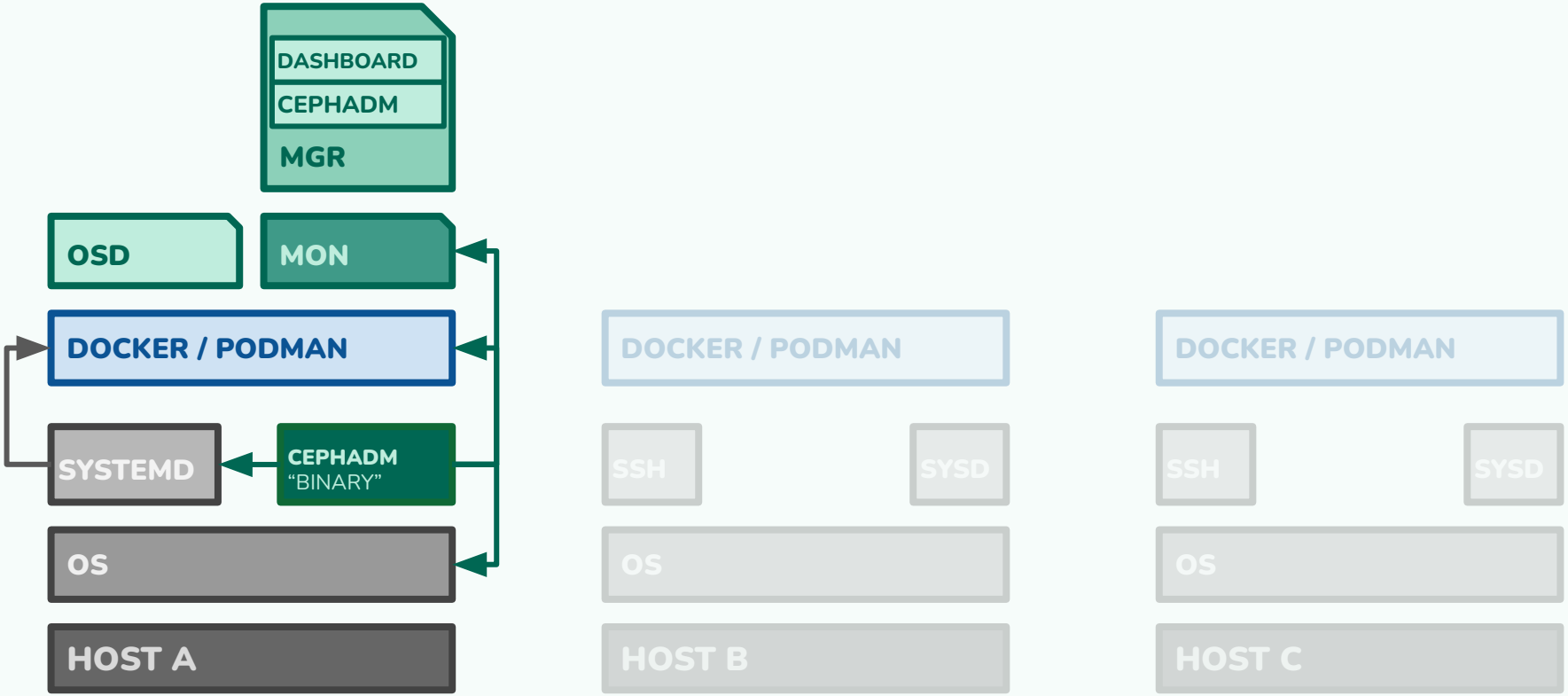
A Cephadm-managed Cluster

Bootstrap: Starting Ceph-Manager...



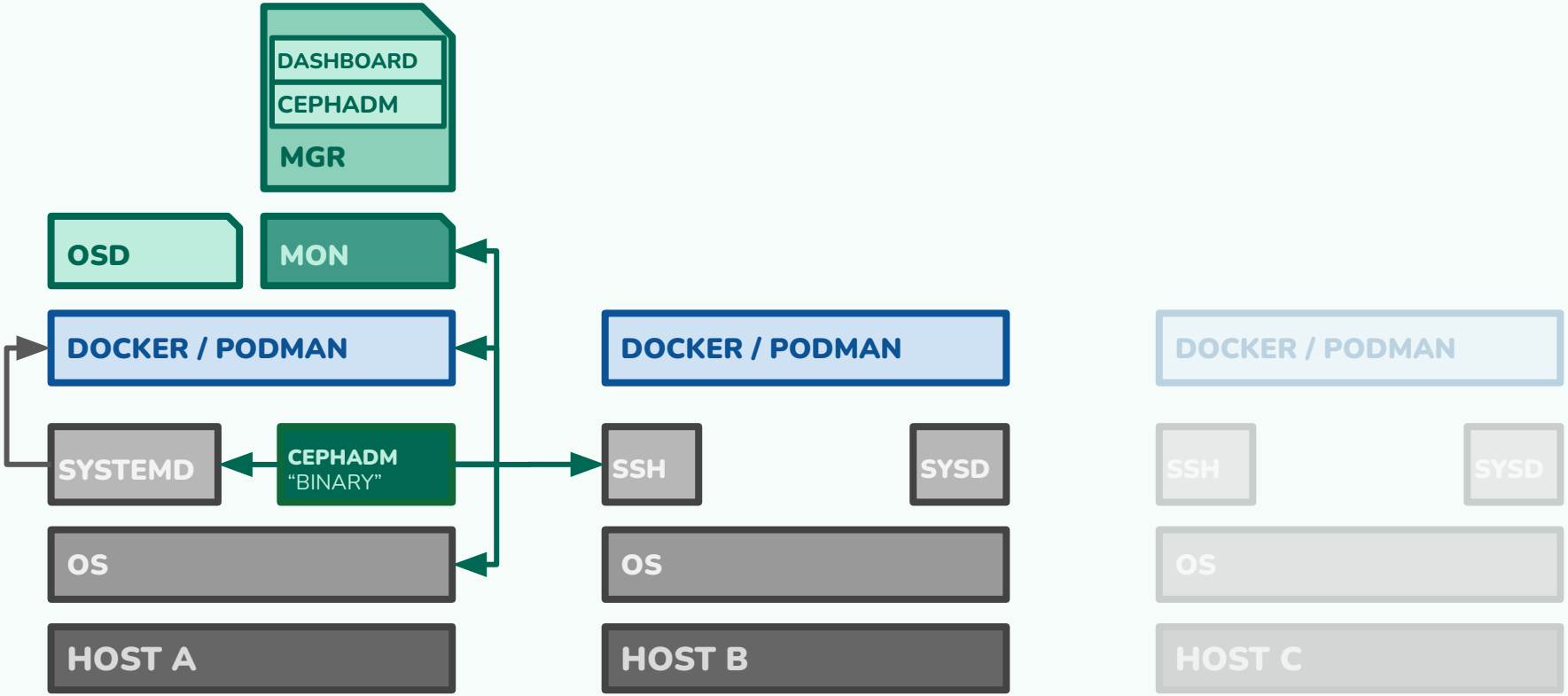
A Cephadm-managed Cluster

Bootstrap Finished: The “Seed” Cluster



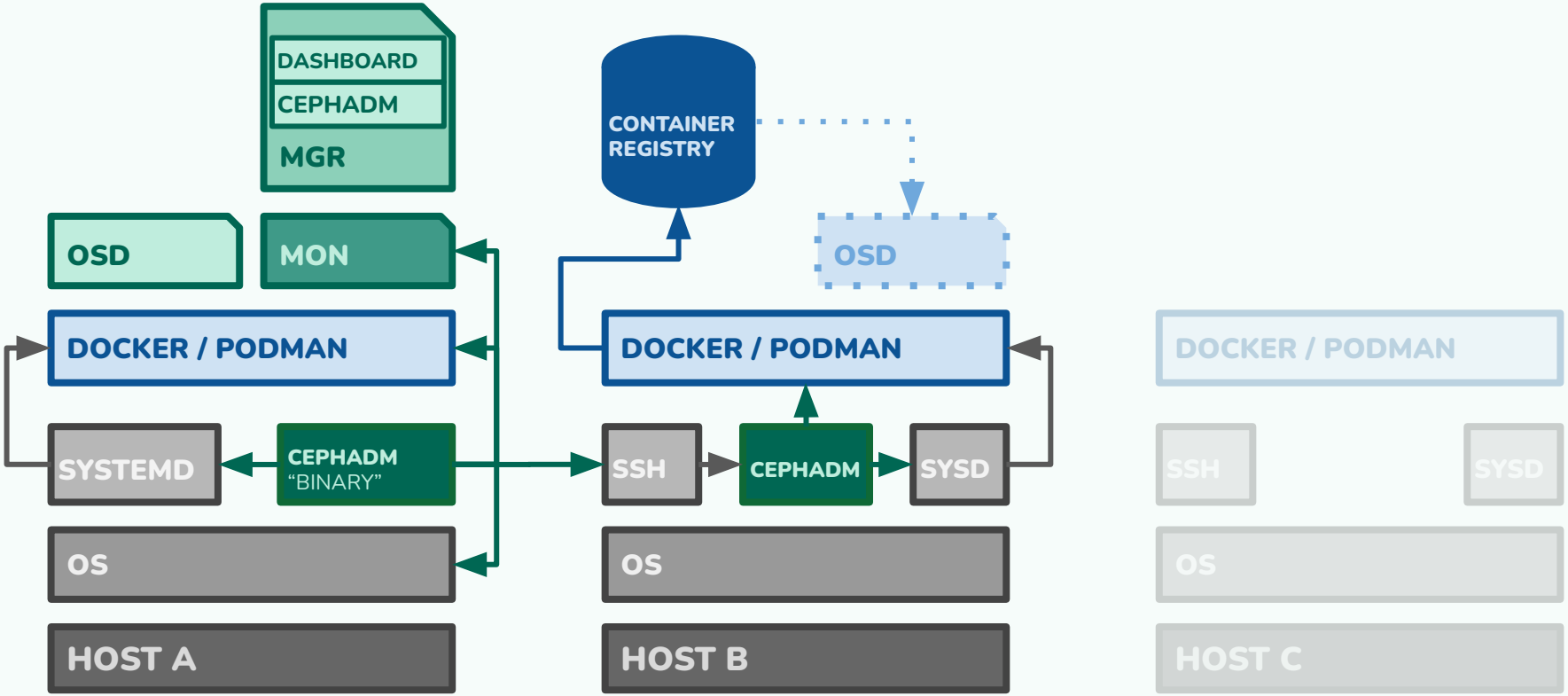
A Cephadm-managed Cluster

Expanding the Cluster: Add Host B



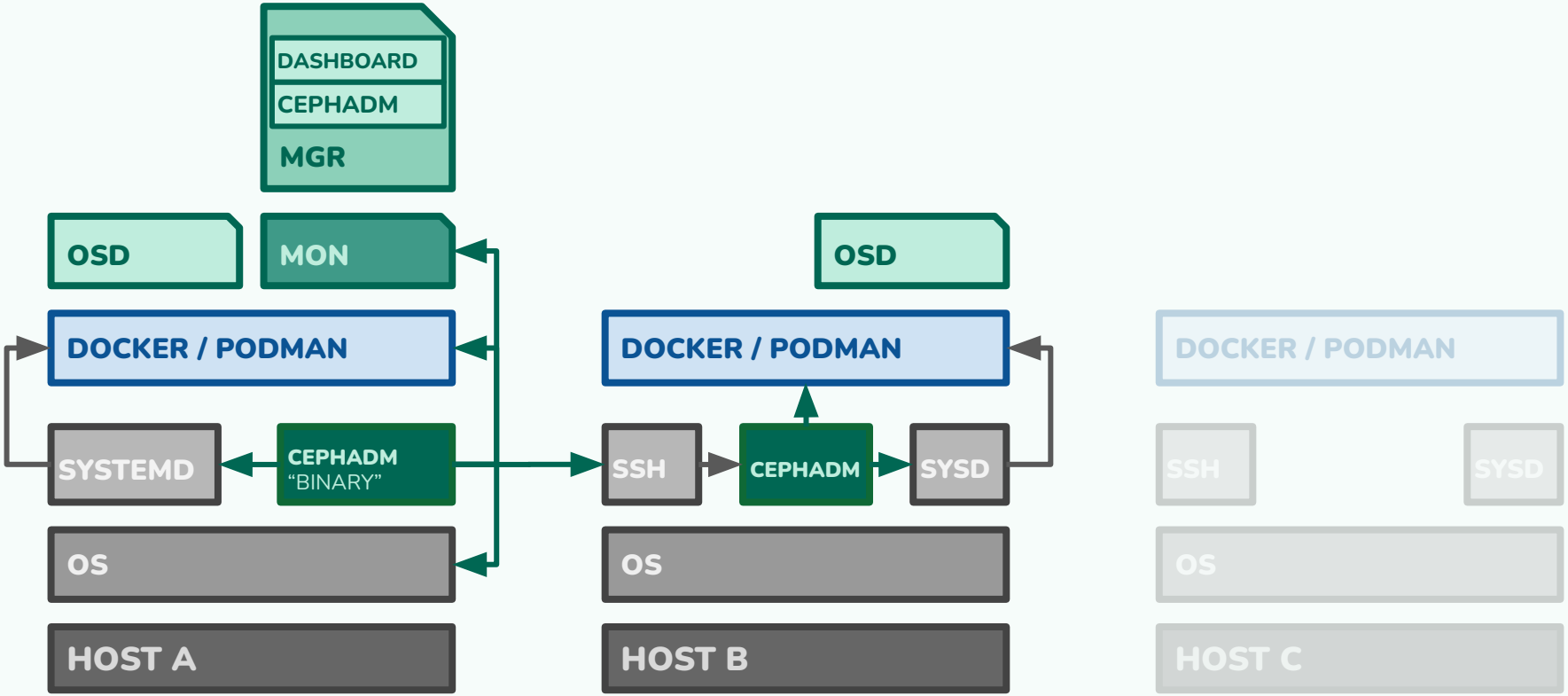
A Cephadm-managed Cluster

Expanding the Cluster - Host B: Pull Containers



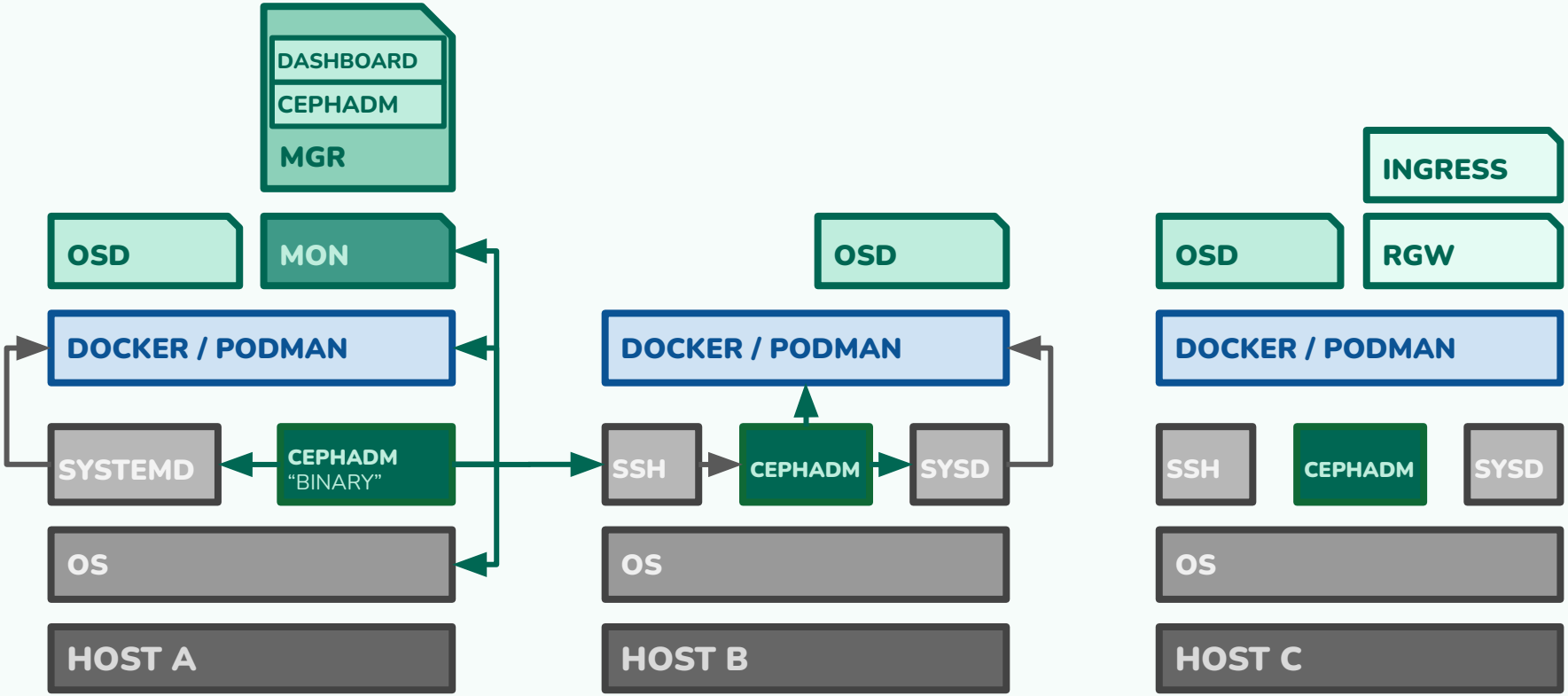
A Cephadm-managed Cluster

Expanding the Cluster - Host B: Deploy Services



A Cephadm-managed Cluster

A 3-Node Cluster



ceph

Cephadm for Detractors: Container Myths

4 Container Myths

1. *"(Lazy) developers just want to ship their computers." (**WOMM**)*
2. *"Containers are just Virtual Machines." (**Performance**)*
3. *"Containers are complex to use/manage/troubleshoot." (**Usability**)*
4. *"Containers are less secure than distro packages/bare metal/etc." (**Security**)*

4 Container Myths
“Works on my Machine”

Containers leverage best practices on CI (long-cherished in the Unix/Linux world):

- **Immutability**
- **Reproducibility**
- **Isolation**
- **Portability**

E.g.: before containers, many FOSS projects used chroots cages/jails to isolate build envs.

Ceph containers are **built from distro packages** (RPMs).

4 Container Myths
Performance

- **Containers are not a kernel thing**, they're **just processes** leveraging 4 userland-available tools in a **unified UX**:
  - **`namespaces`**: `mnt, uts, ipc, pid, net (lsns, ip-netns, nsenter, unshare, …)`
  - **`cgroups`**: `/sys/fs/cgroup`
  - **`overlayfs`**: `CoW`,
  - **`bridges`**,
- **Containers are not VMs** (not even paravirtualization):
  - **~0% drop for CPU, memory and network.** [1]
  - **Direct IO has 0% drop**. Overlayfs can be worse (~5%).
- **Containers are just Software-Defined Stuff**: aren't you using VLANs, SR-IOV, LVM, SDN, … or Ceph?

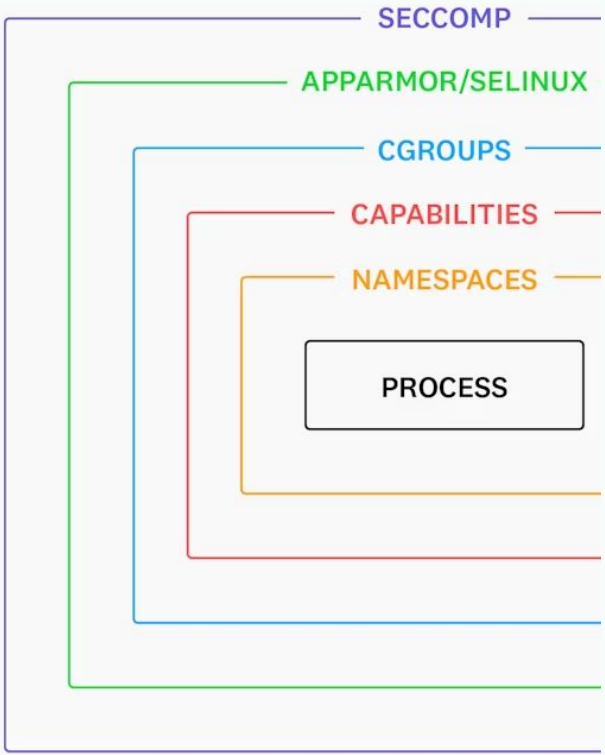
[1] "Bare-Metal vs. Hypervisors and Containers: Performance Evaluation of Virtualization Technologies for Software-Defined Vehicles", Long Wen, Markus Rickert, et al.

4 Container Myths
Usability

- **Open Container Initiative**: standard across Linux, Windows, Solaris, z/OS, ...
- Choose your **runtime**:
  docker, podman
- **Service lifecycle management**:
  `pull,run,start,stop,pause,unpause`
- **Debug/troubleshoot**:
  `logs/events,ps,attach/exec,inspect,diff`
- **Snaphots & manipulation:**
  `commit,cp,load,save,export,`
- **Improved `systemd` integration** with Podman Quadlets.

4 Container Myths
Security

- A **rootless container** starts with **less privileges** than any **regular user process**.
- **5 security layers**:
  - **seccomp**: limits what syscalls a process can invoke,
  - **AppArmor/SELinux**: limits how a process interacts with resources via fine-grained Mandatory Access Control (vs. Unix DAC).
  - **Cgroups**: limits how much resources a process can take.
  - **CAPS**: limits what system capabilities a process can access (vs. Unix Root-or-Nothing)
  - **Namespaces**: limits what a process can see.
- **Bare-metal vs. Container**: with containers you can have multiple versions of the same packages running in the same runtime (e.g.: for upgrades/rollbacks).



Picture from https://securitylabs.datadoghq.com/articles/container-security-fundamentals-part-2/

The Shape of Things to Come

- Bootable Containers (**bootc**)

```
FROM quay.io/centos-bootc/centos-bootc:stream9
RUN dnf install -y podman lvm2 chrony cephadm
...
```

- **Immutable** OS (RHEL Image Mode)
  - **Zero**-drift.
  - Only `/etc` and `/var/lib` are mutable (and changes can be tracked in a **GitOps** approach).
  - Layered upgrades/rollbacks. Multiple control points

ceph

Managing
An XXL Cluster

Ceph Cluster Sizes

<5 nodes	5-50 nodes	50-100 nodes	**>100 nodes**
<15 OSDs	15-1000 OSDs	1000-4000 OSDs	**>4000 OSDs**
	Most frequent	**<0.1%**	**<0.001%**

[1] Ceph Users Survey 2022: https://ceph.io/en/news/blog/2022/ceph-user-survey-results-2022/

The Tipping Point

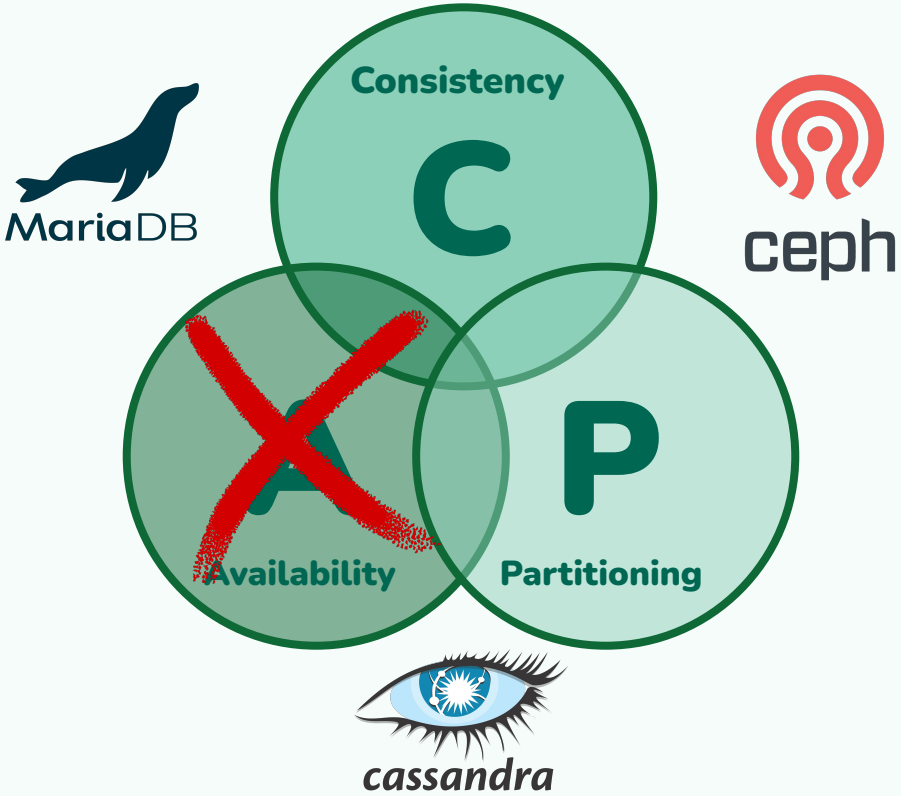
Unwritten **Law of Distributed Systems**:

> *"They scale horizontally, but fail vertically"*

This is especially true with **Strong-Consistency Systems**:

> Failing nodes are indistinguishable from a network partition.

The CAP Theorem



Brief History of Cephadm Scale Testing

Nov 2021: Pawsey Supercomputing Centre

Heavy Ceph users (**object**):

- 11 PB cluster
- 27 PB cluster

Test:

- Quincy
- 180 nodes
- 4,320 OSDs (22 OSD/host)
- 69 PB (raw)
- 100G NICs

Key Findings:

- **mgr**: too many perf-counters
  - **ceph-exporter**
- **Cephadm**:
  - high CLI lag (10-30s)
  - both polling and (EXPERIMENTAL) agent modes worked.
  - lack of filtering in `orch host ls` led to poor UX

Brief History of Cephadm Scale Testing

Jun 2022: Gibba@Ceph + Scalelab@Red Hat

3 Quincy environments:

- **Ceph Gibba**:
  - 40 hosts
  - 975 OSDs
- **Red Hat Scalelab:**
  - **“Logical Large Scale”**:
    - 127 hosts
    - 8,134 ODSs (NVMe)
  - **“Cephadm upgrade”**
    - 13 hosts
    - 832 OSDs

Key Findings:

- **cephadm**: issues running `orch` commands in multiple hosts.
  - Used Ansible to pre-provision basic host dependencies.

Next Challenge: 10k OSDs

- **Max recommended**: <4,000 OSDs
- **Beyond 4k OSDs**: careful design required (CPU, RAM, network bandwidth), and intensive mgmt & monitoring.
- **10k OSD challenge**: further code changes will be required (e.g.: Cephadm Agent).

ceph

Cephadm
Community

How to contribute

- Virtual **meetings**:
  - Ceph **Developer Monthly** (Wednesday)
  - Ceph **Users + Devs Monthly** (Thursdays)
- As a **user**:
  - Getting help:
    - ceph-users@lists.ceph.com,
    - Slack #ceph and #ceph-devel
  - Reporting issues or requesting features: tracker.ceph.com
- As a **developer**:
  - dev@ceph.io
  - github.com/ceph

Questions?

ceph

Thank you!