Setting up a hyperconverged Proxmox cluster

Intro to Proxmox & Ceph / Resource planning / Step by step live demo

Author:
Sami Ait Ali Oulahcen

Nouakchott, Mauritania
17-22 February 2025

Plan

- Intro to the private cloud components
  - Compute
  - Storage
  - Network
- Resource planning
- Live demo

PROXMOX

Compute: PVE

What is Proxmox VE

- Proxmox Virtual Environment (PVE) is a complete, open-source virtualization management platform. It leverages many existing opensource projects to provide compute, network, and storage in a single solution.
- Compute: QEMU/KVM for VMs and LXC for containers
- Network: through the Linux network stack
- Storage: through Ceph w/ other options available

Specs & Features

- PVE is based on Debian Linux and licensed under the GNU AGPL v3. It runs on commodity hardware with x86_64 architecture
- It supports up to 8 CPU sockets (max 8192 logical CPUs) and up to 128 TiB RAM per node (max 64 PiB in total)
- Offers practically all features you’d need in a virtualization environment: full-featured GUI, User Management with 2FA, HA and clustering, Live Migration, Snapshots, Templates and Clones, and many other features

ceph

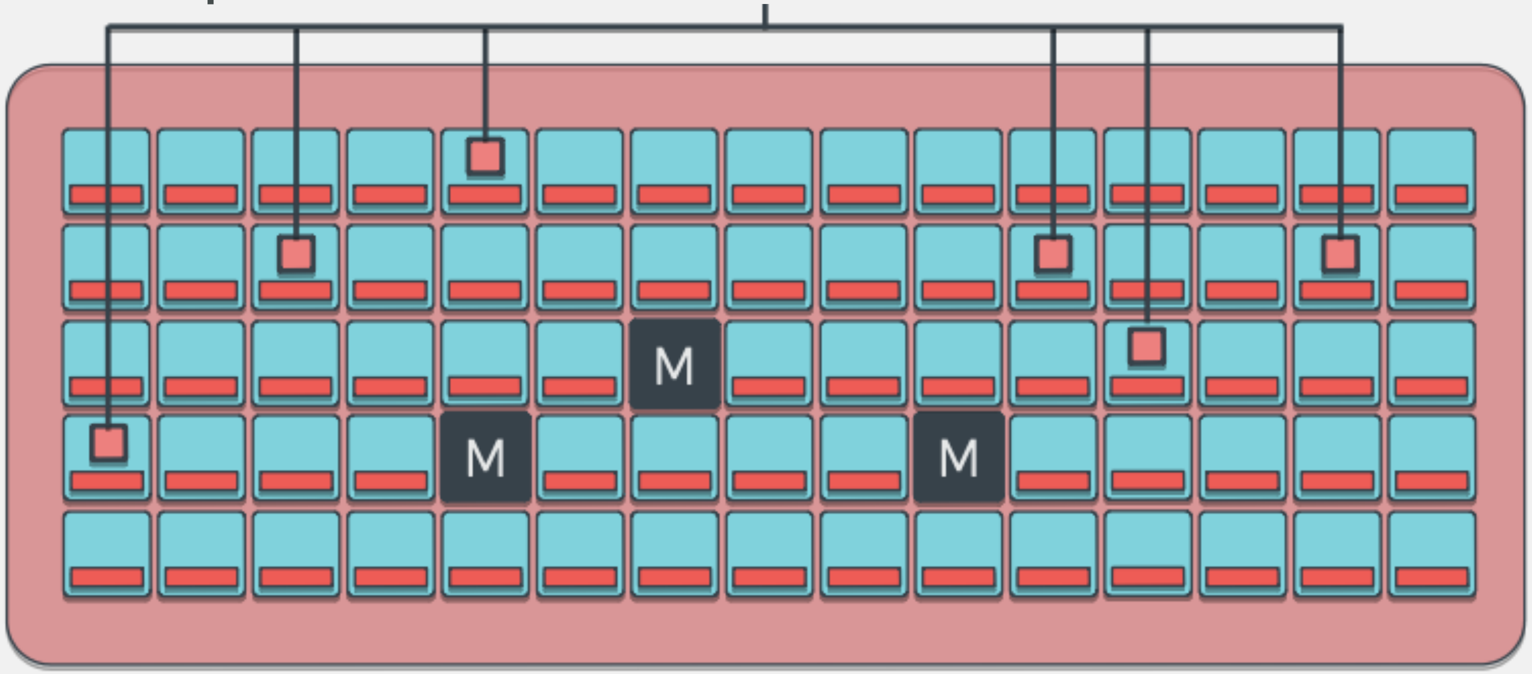
Storage: ceph

What is ceph ?

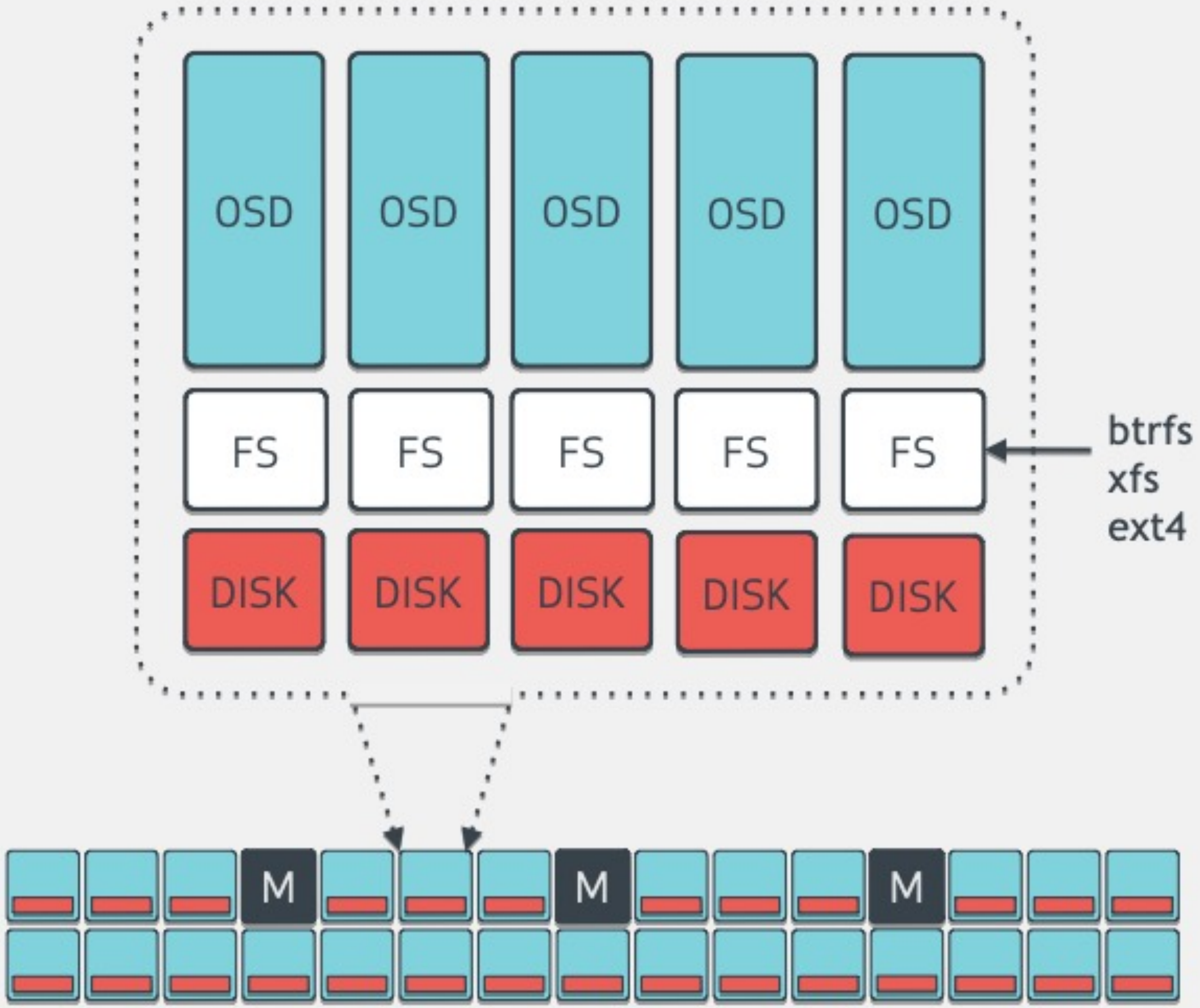
- Ceph is a distributed fault-tolerant opensource software storage platform designed to present object, block, and file storage.
- Ceph's main goals are to be completely distributed without a single point of failure, scalable to the exabyte level, and freely-available.

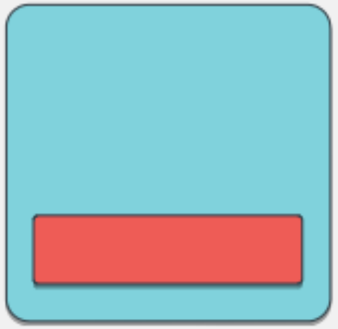
Ceph storage cluster

- Ceph does striping of individual files across multiple nodes for higher throughput, similar to RAID0
- Adaptive load balancing is supported: frequently accessed objects are replicated over more nodes.



Object Storage Daemons (OSDs) & Monitors (MONs)





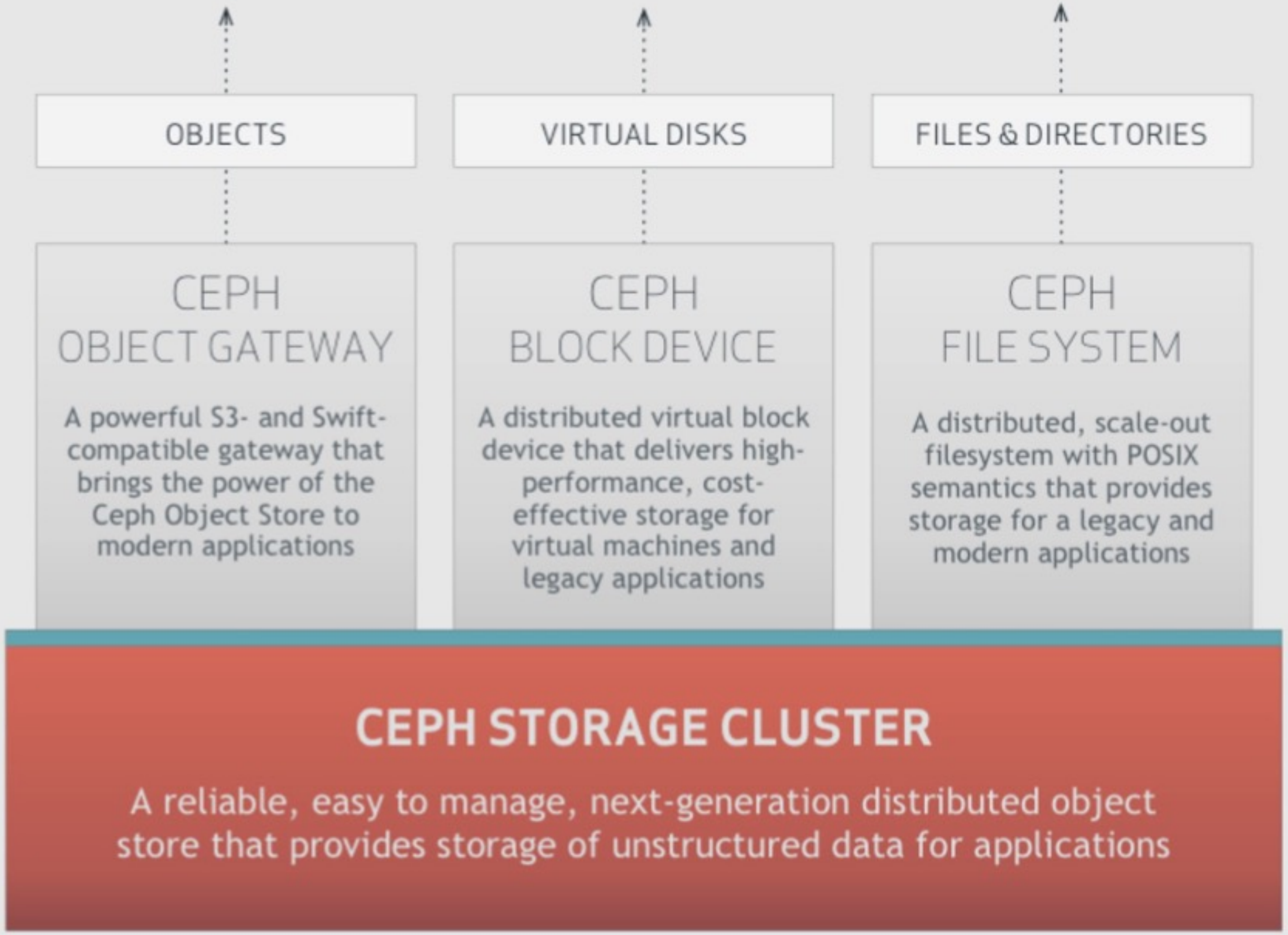
OSDs:

- 10s to 10000s in a cluster
- One per disk
  - (or one per SSD, RAID group…)
- Serve stored objects to clients
- Intelligently peer to perform replication and recovery tasks

Monitors:

- Maintain cluster membership and state
- Provide consensus for distributed decision-making
- Small, odd number
- These do **not** serve stored objects to clients

Storage types



OBJECTS

VIRTUAL DISKS

FILES & DIRECTORIES

CEPH OBJECT GATEWAY

A powerful S3- and Swift-compatible gateway that brings the power of the Ceph Object Store to modern applications

CEPH BLOCK DEVICE

A distributed virtual block device that delivers high-performance, cost-effective storage for virtual machines and legacy applications

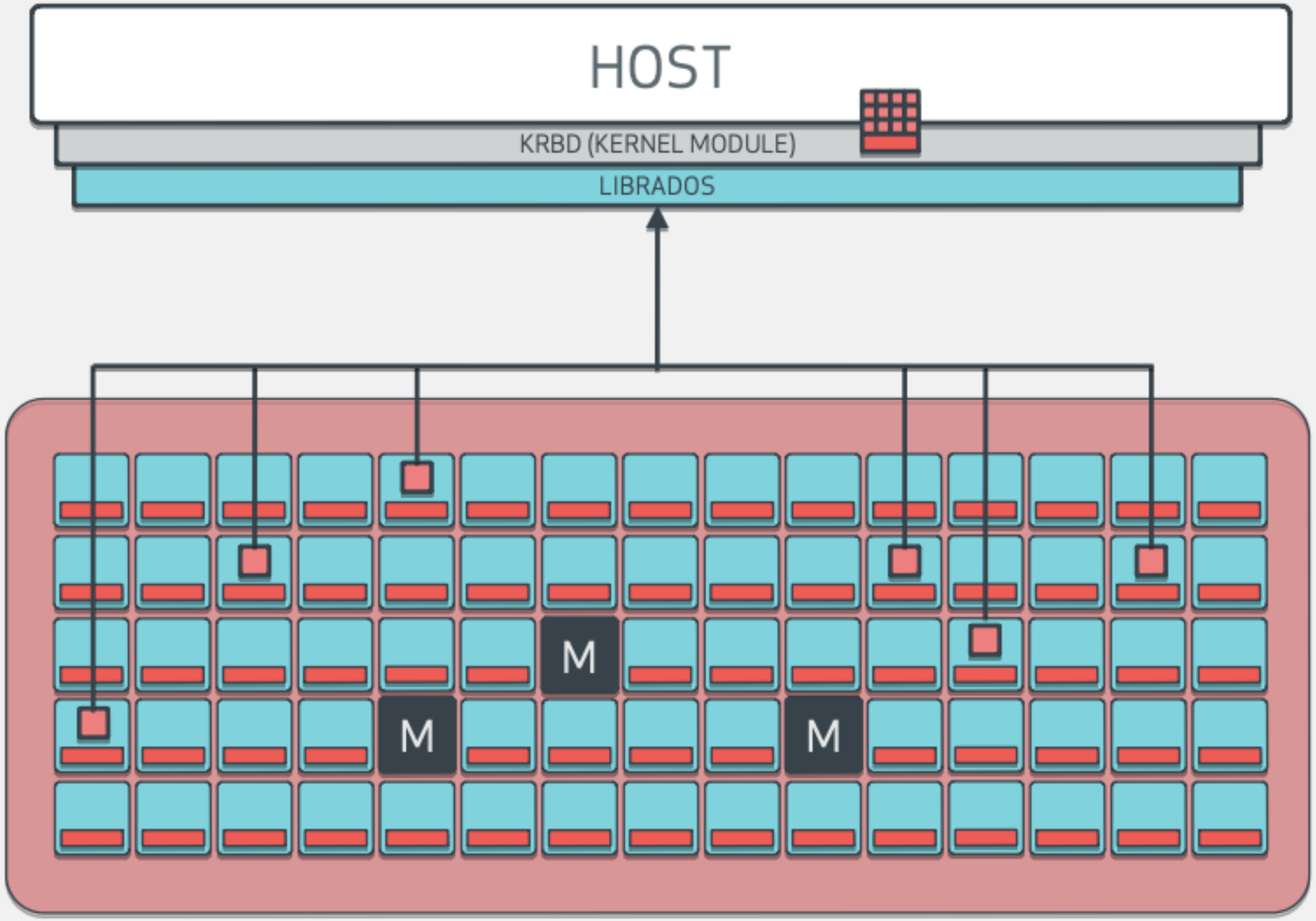
CEPH FILE SYSTEM

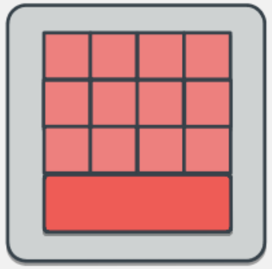
A distributed, scale-out filesystem with POSIX semantics that provides storage for a legacy and modern applications

CEPH STORAGE CLUSTER

A reliable, easy to manage, next-generation distributed object store that provides storage of unstructured data for applications

Ceph block storage

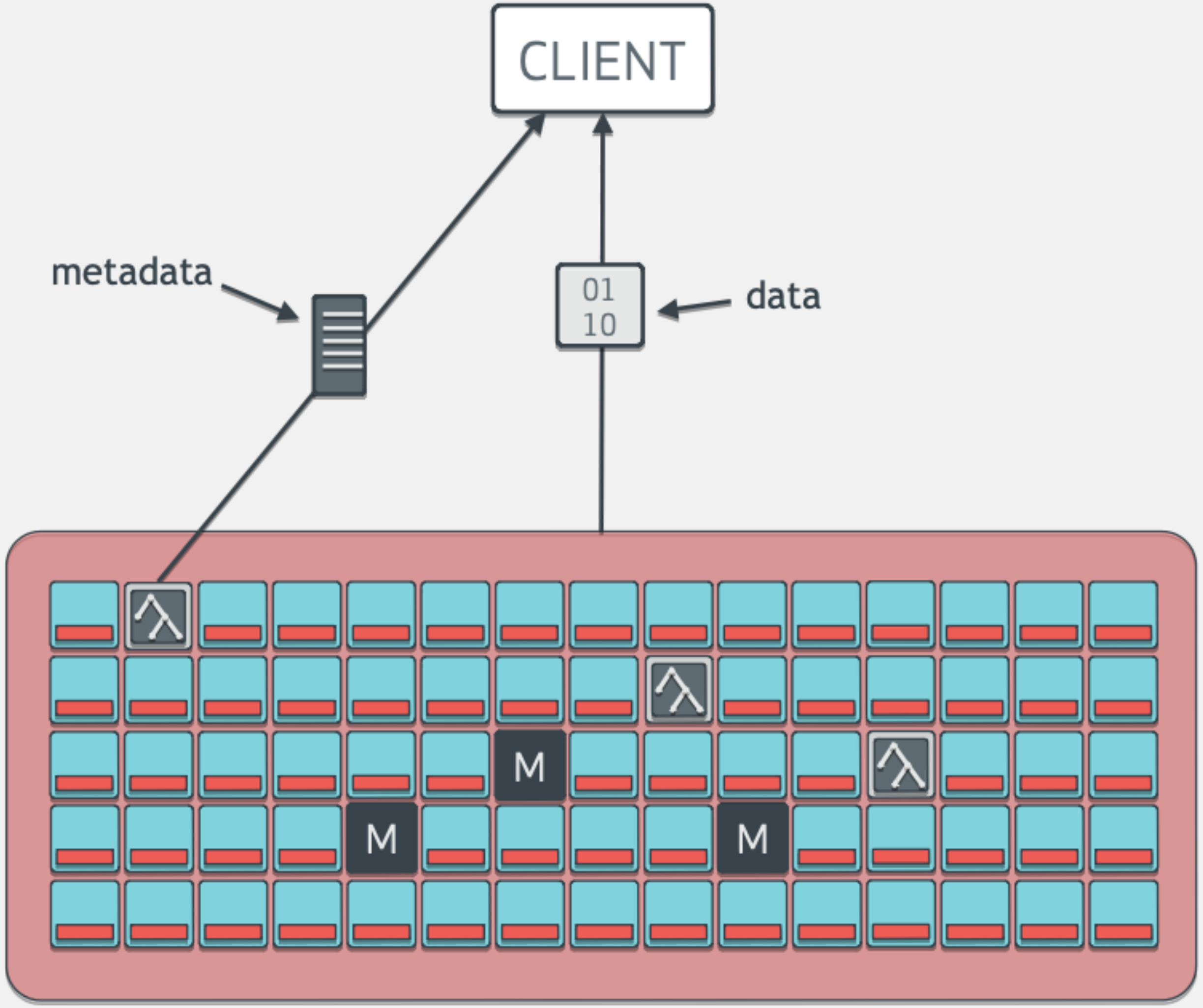




RADOS Block Device:

- Storage of disk images in RADOS
- Decouples VMs from host
- Images are striped across the cluster (pool)
- Snapshots
- Copy-on-write clones
- Support in:
  - Mainline Linux Kernel (2.6.39+)
  - Qemu/KVM
  - OpenStack, CloudStack

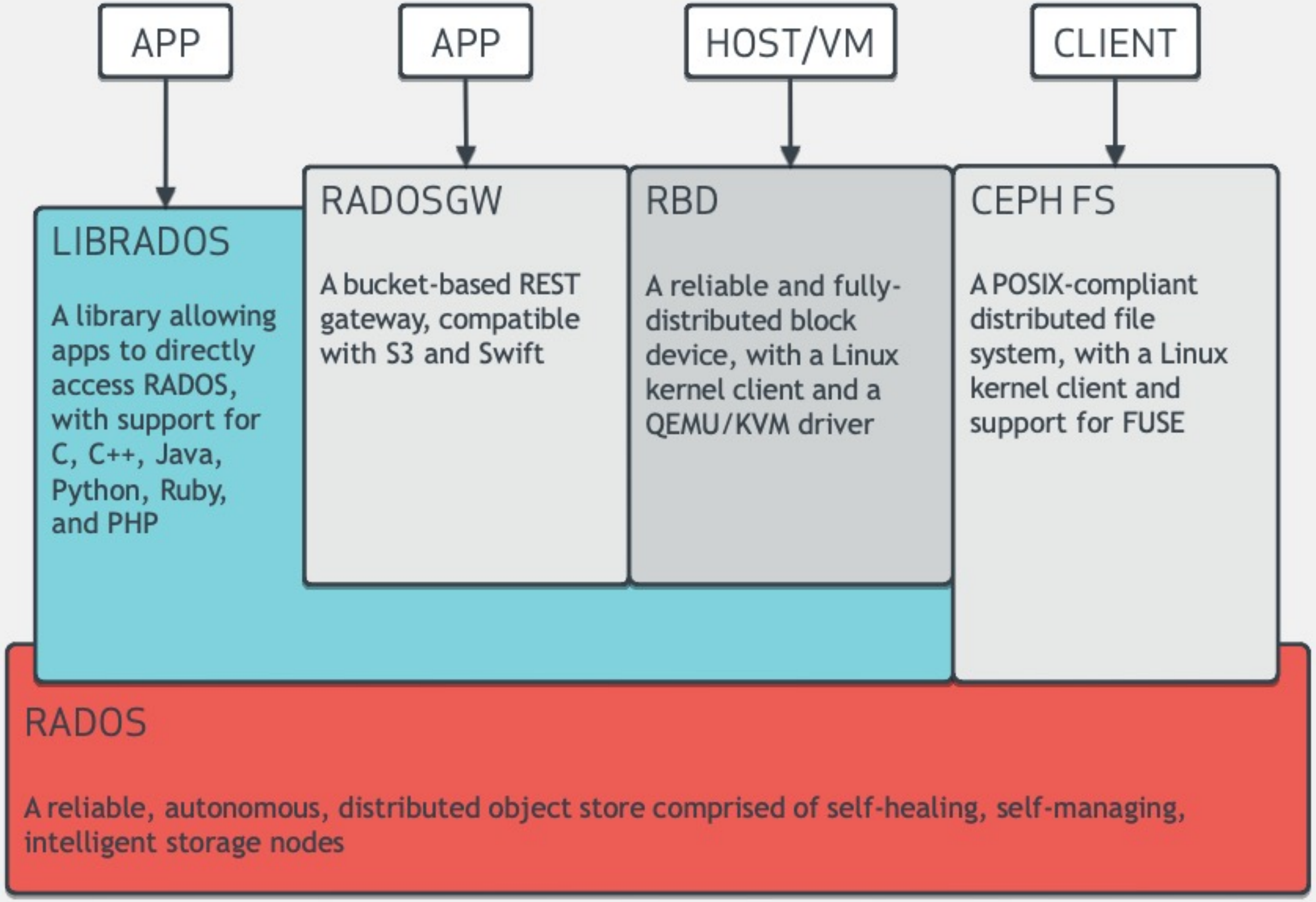
Ceph file system



Metadata Server

- Manages metadata for a POSIX-compliant shared filesystem
  - Directory hierarchy
  - File metadata (owner, timestamps, mode, etc.)
- Stores metadata in RADOS
- Does not serve file data to clients
- Only required for shared filesystem

Quick summary



APP

APP

HOST/VM

CLIENT

LIBRADOS

A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP

RADOSGW

A bucket-based REST gateway, compatible with S3 and Swift

RBD

A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver

CEPH FS

A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE

RADOS

A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

Quick summary

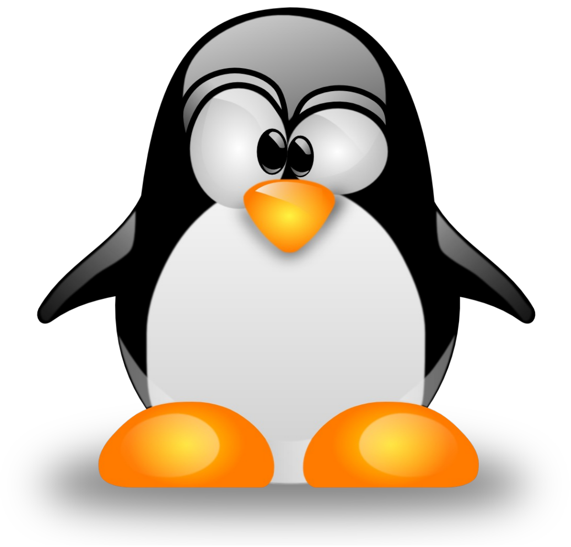
- Ceph RBD: we store VMs & containers here
- Ceph FS: we store ISO images here
- Ceph runs on 4 daemons:

OSD
actually stores the content of files on top of the local filesystem

MON
keep track of active and failed cluster nodes

MDS
stores the metadata of inodes and directories

RGW
exposes the object storage layer as an API



Network: the Linux kernel

Interfaces used by Proxmox

- Proxmox uses the Linux kernel for networking
- Bridge Interfaces: Used to connect multiple network interfaces into a single logical network segment, which is essential for virtualization
- VLAN Interfaces: Used to create isolated network segments within a single interface

Install Proxmox: planning

- How many VMs do you need ?
- Evaluate resources needed:
- CPU: 1-4 vCPUs per VM (remember 1 CPU core = 2 vCPUs)
  + extra CPU cores for system services
- RAM: 1-4GB per conventional VM/CT
  + 1GB RAM per 1TiB of data used by Ceph OSDs
  + extra RAM for system services
- Storage: SSD if you can afford
  Facture in replication (if you need 2TB get 4TB)
  No RAID controller, we’re using Ceph for replication

Install Proxmox: planning

- Network: Have at least 3 network interfaces

  Small cluster example: 10/25/100 GE interface for Ceph

  1 GE interface for management

  10 GE interface for VM networks

- Plan for an **odd number** of nodes to have quorum: 3 or 5 or 7

Install Proxmox: prepare servers

- Enable virtualization on BIOS if not enabled already
- Remove disks from RAID controller if you have one, use host bus adapter (HBA) instead
- Download the Proxmox Virtual Environment (PVE) ISO and burn it into a USB drive

Install Proxmox: cluster & networks

- Install Proxmox Virtual Environment (PVE) on all desired nodes

(Preferably an **odd number** of nodes to have quorum: 3 or 5 or 7)

- Create a cluster on one of the nodes
- Join all nodes to the cluster
- Configure network interfaces:

  - Management network: create bridge and configure IP addresses and gateways

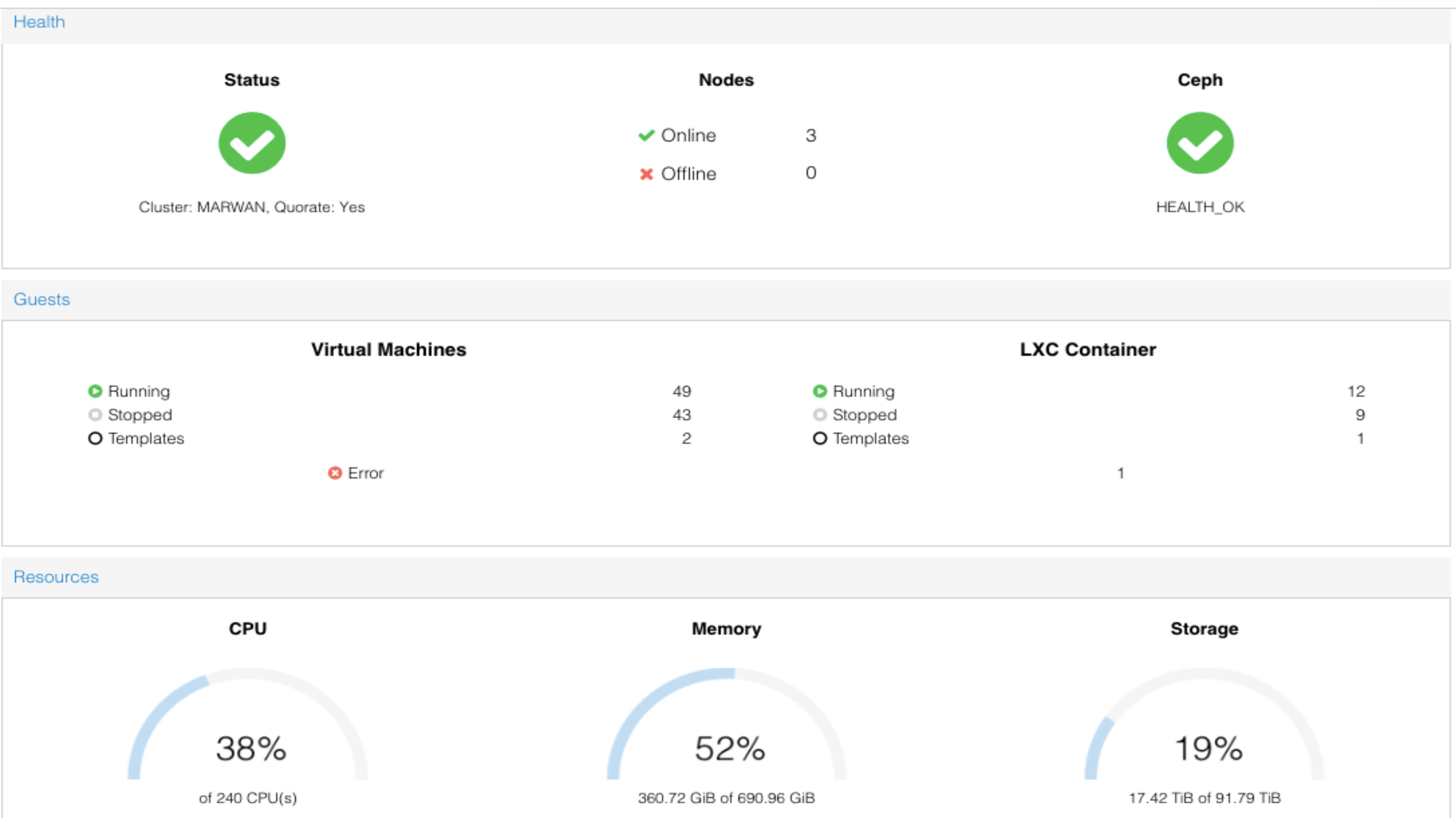
  - Ceph network: create bridge and configure IP addresses

  - VM networks: create VLANs then bridges (no IP addresses needed)

Install Proxmox: storage

- Install & configure Ceph via GUI on one of the nodes
- Install Ceph on the other nodes
- Add OSDs (1 drive = 1 OSD) on all the nodes that have storage
- Add monitors (MON) and managers (MGR) for redundancy
- Add metadata servers (MDS)
- Create CephFS storage for ISO images and container templates
- Create Ceph pool for VM disks storage

Install Proxmox: live showcase



Health

Status

✓

Cluster: MARWAN, Quorate: Yes

Nodes

✓ Online	3
✗ Offline	0

Ceph

✓

HEALTH_OK

Guests

Virtual Machines

● Running	49
○ Stopped	43
○ Templates	2

LXC Container

● Running	12
○ Stopped	9
○ Templates	1

✗ Error	1

Resources

CPU

38%

of 240 CPU(s)

Memory

52%

360.72 GiB of 690.96 GiB

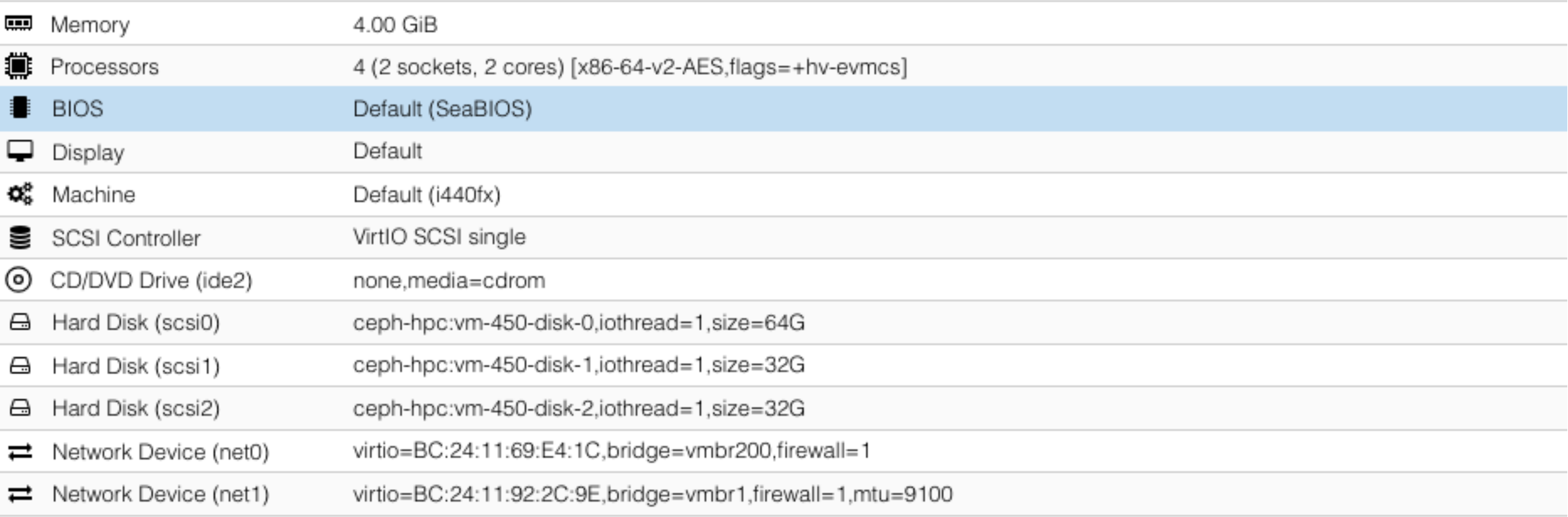
Storage

19%

17.42 TiB of 91.79 TiB

Annex

Proxmox Live showcase: node configuration



Memory	4.00 GiB
Processors	4 (2 sockets, 2 cores) [x86-64-v2-AES,flags=+hv-evmcs]
BIOS	Default (SeaBIOS)
Display	Default
Machine	Default (i440fx)
SCSI Controller	VirtIO SCSI single
CD/DVD Drive (ide2)	none,media=cdrom
Hard Disk (scsi0)	ceph-hpc:vm-450-disk-0,iothread=1,size=64G
Hard Disk (scsi1)	ceph-hpc:vm-450-disk-1,iothread=1,size=32G
Hard Disk (scsi2)	ceph-hpc:vm-450-disk-2,iothread=1,size=32G
Network Device (net0)	virtio=BC:24:11:69:E4:1C,bridge=vmbr200,firewall=1
Network Device (net1)	virtio=BC:24:11:92:2C:9E,bridge=vmbr1,firewall=1,mtu=9100

Acknowledgments

- https://pve.proxmox.com/wiki/Deploy_Hyper-Converged_Ceph_Cluster
- https://www.proxmox.com/en/products/proxmox-virtual-environment/comparison