

Proxmox Virtualisation Environment

- Introduction to ProxMox GUI
- Introduction to storage
- Introduction to networking
- Backups and guest import / migration
- Cluster setup
- Ceph redundant shared storage
- High Availability
- CLI-tools and system performance

Introduction to Proxmox

- Proxmox is an open-source hyper-converged virtualization environment
- It has a bare-metal installer, a web-based remote management GUI, a HA cluster stack, unified cluster storage, and a flexible network setup
- It has commercial support packages available at a reasonable cost

Introduction to Proxmox

- Proxmox use the following underlying technologies:
 - KVM (type 1 hypervisor module for Linux)
 - QEMU hardware emulation
 - LXC Linux containers
 - Ceph replicated storage
 - Corosync Cluster Engine

KVM

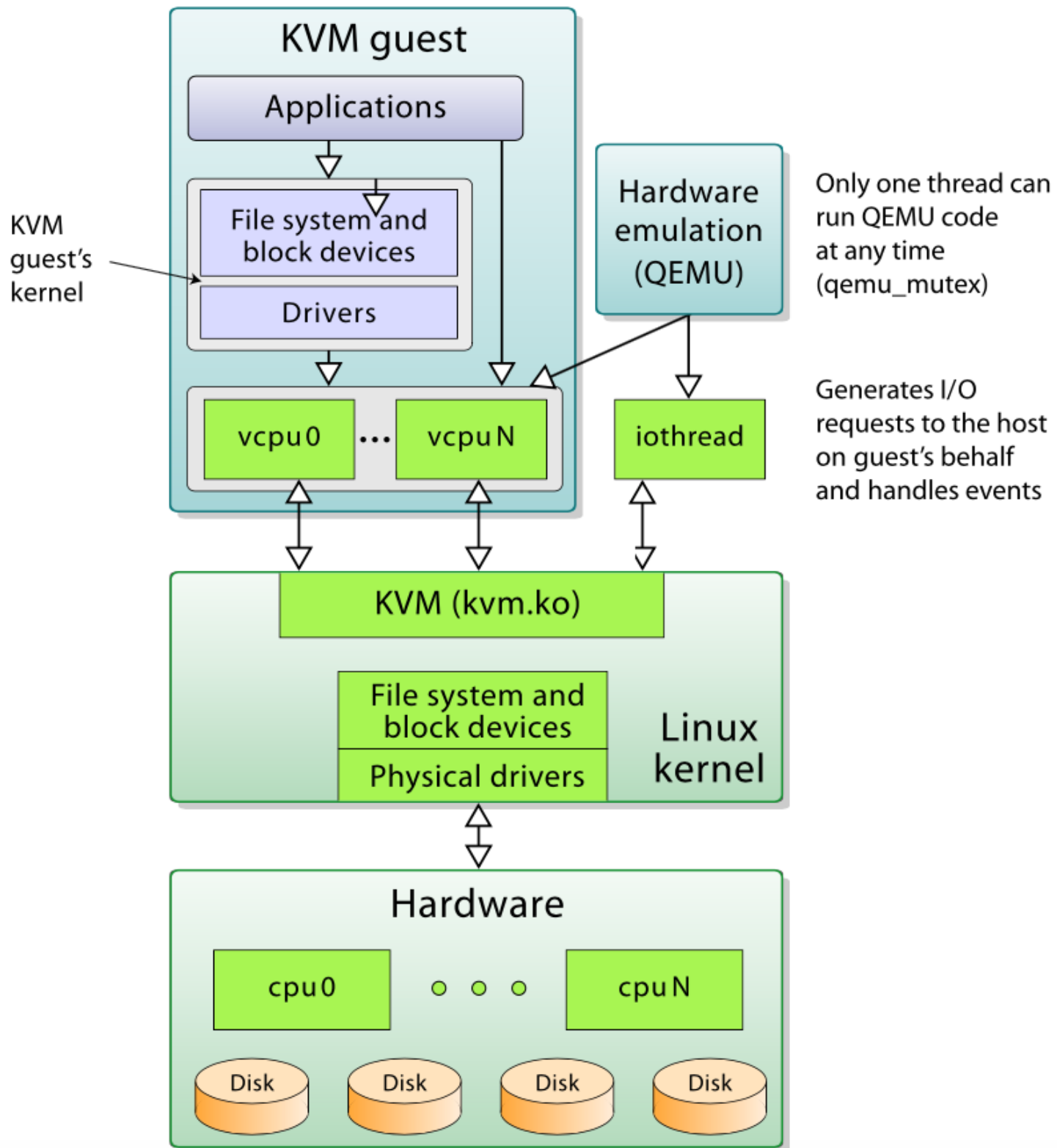
- Kernel-based Virtual Machine (KVM) is a virtualization infrastructure for the Linux kernel that turns it into a hypervisor
- KVM requires a processor with hardware virtualization extensions and a wide variety of guest operating systems work with KVM
- Supports a paravirtual Ethernet card, a paravirtual disk I/O controller using VirtIO, a balloon device for adjusting guest memory usage, and a VGA graphics interface

QEMU

- QEMU (Quick Emulator) is an open source hosted hypervisor that performs hardware virtualization
- Emulates CPUs through dynamic binary translation and provides a set of device models, enabling it to run a variety of unmodified guest operating systems
- Uses KVM Hosting mode in Proxmox where QEMU deals with the setting up and migration of KVM images

QEMU

- It is still involved in the emulation of hardware, but the execution of the guest is done by KVM as requested by QEMU
- Uses KVM to run virtual machines at near-native speed (requiring hardware virtualization extensions on x86 machines)
- When the target architecture is the same as the host architecture, QEMU can make use of KVM particular features, such as acceleration



LXC

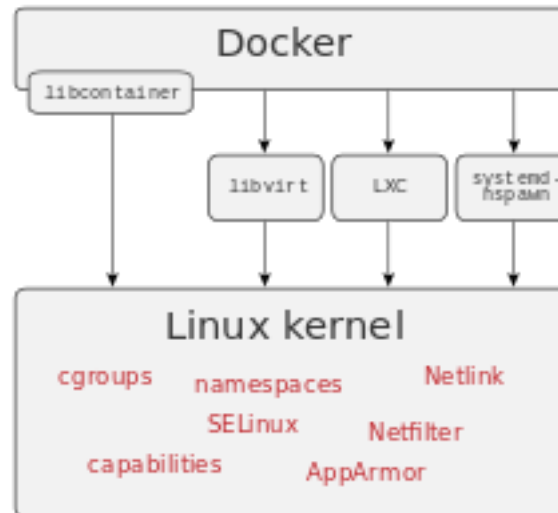
- LXC (Linux Containers) is an operating-system-level virtualization method for running multiple isolated Linux systems (containers) on a control host using a single Linux kernel.
- The Linux kernel provides the cgroups (control groups) functionality that allows limitation and prioritization of resources (CPU, memory, block I/O, network, etc.) without the need for starting any virtual machines.

LXC

- Provides namespace isolation functionality that allows complete isolation of an applications' view of the operating environment, including process trees, networking, user IDs and mounted file systems.
- LXC combines the kernel's cgroups and support for isolated namespaces to provide an isolated environment for applications.

LXC

- Docker can also use LXC as one of its execution drivers, enabling image management and providing deployment services.



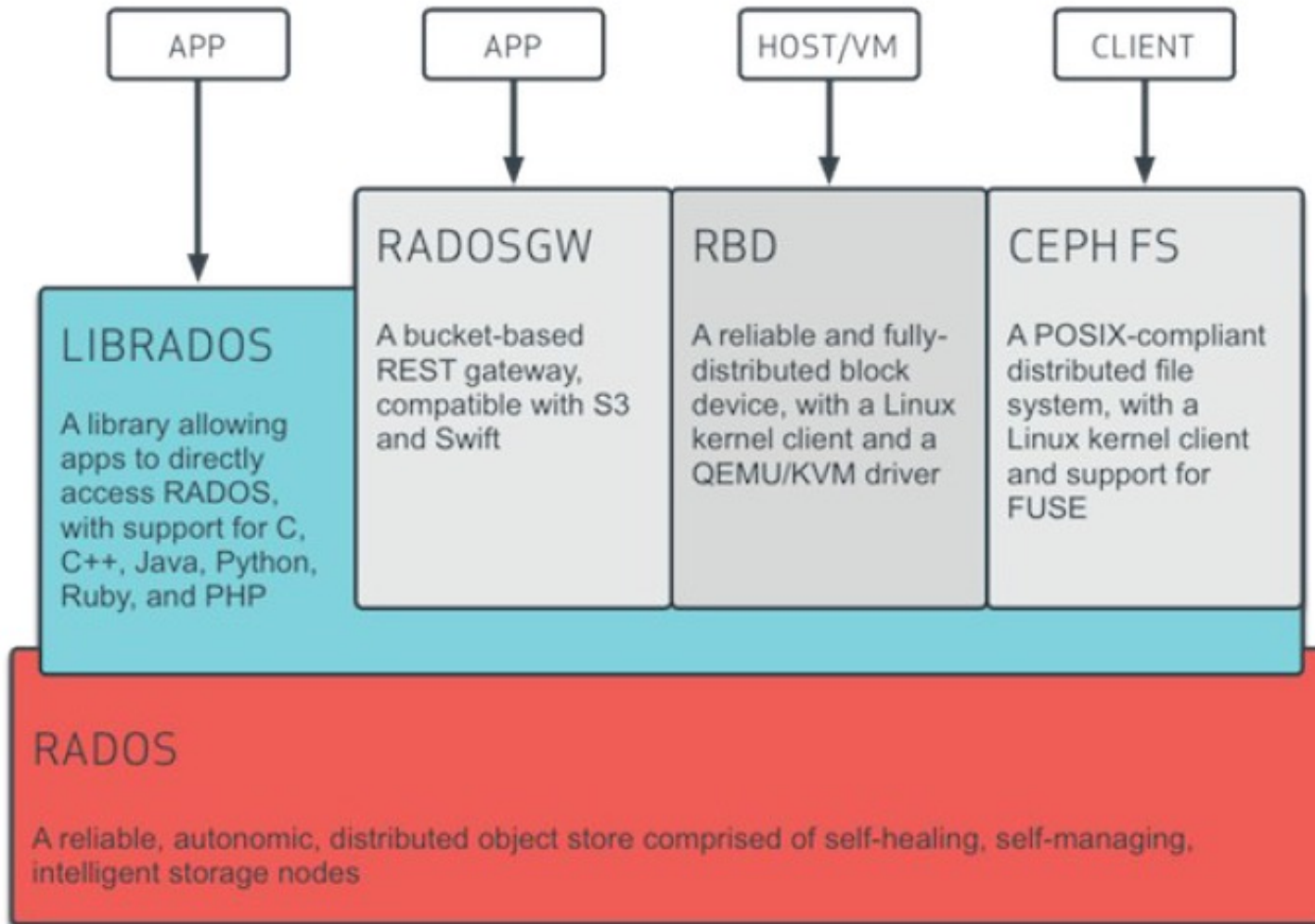
Ceph

- Ceph is a storage platform, that implements object storage on a single distributed computer cluster, and provides interfaces for object-, block- and file-level storage
- Ceph aims for completely distributed operation without a single point of failure, scalable to the exabyte level
- Ceph's software libraries provide client applications with direct access to the Reliable Autonomic Distributed Object Store (RADOS) object-based storage system

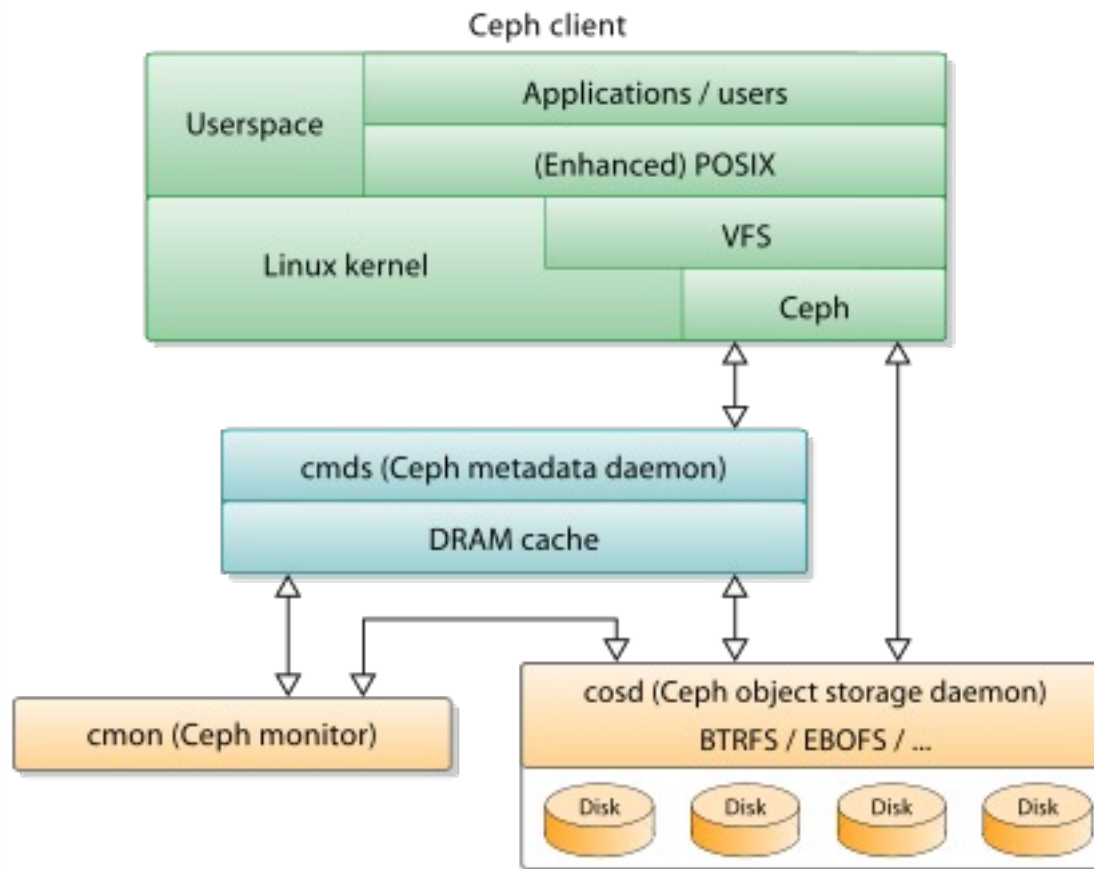
Ceph

- Ceph replicates data and makes it fault-tolerant, using commodity hardware and requiring no specific hardware support
- As a result of its design, the system is both self-healing and self-managing, aiming to minimize administration time and other costs
- When an application writes data to Ceph using a block device, Ceph automatically stripes and replicates the data across the cluster.
- Works well with KVM

Ceph Architecture



Ceph Internal Organisation



Ceph Network

- Create ceph ring0 network
 - each node must be reachable on ring0
 - check firewalls on each node
- Proxmox distribute their own ceph package as of V5.1
 - `pveceph install` will install latest stable repositories and packages
 - must be run on each node individually
 - `ceph init --network x.x.x.x/y` on first node only
 - `ceph createmon` on each node

Ceph OSDs

- Add disks as Object Storage Devices in each node
- Accurate network time is also very important to avoid 'clock slew'
 - the latest network time system daemon (system.time?) is much better than ntpdate
 - QEMU has a new time source driver which can be run in guests needing accurate time

Ceph Pools

- Pools are individual storage blocks
 - size is the number of replications (OSDs) per block
 - min-size is the minimum number of OSDs (replications) each block must be on to allow read-write status
 - 'add-storage' option automatically adds the storage block to the hosts rather than having to manually copy ceph keys to each host to allocate the storage

Ceph Pools

- The client (Proxmox) interacts with one OSD only
- This OSD then writes to and confirms write on each OSD in the block before confirming write completion
 - Writes are actually made to the journal rather than the block level device for speed
 - This primary OSD manages all interactions with both the client and the replication OSDs
 - In case primary manager is lost, a backup OSD will take over as primary

Ceph Pools

- Crush maps define the actual storage blocks, these are very complicated so don't change the default settings!
- As new OSDs are added, ceph will attempt to re-allocate data across blocks to improve access and availability
- If an OSD gets removed, ceph will rebalance data once OSD is marked as OUT (300 seconds by default)
 - use `ceph noout` to avoid rebalancing, e.g. for maintenance

Proxmox GUI

PROXMOX Virtual Environment 4.4-1/eb2d6f1e You are logged in as 'root@pam' [Help](#) [Create VM](#) [Create CT](#) [Logout](#)

Server View Node 'pve' [Restart](#) [Shutdown](#) [Shell](#) [More](#) [Help](#)

Server View: Datacenter > pve

- 100 (debian9-nfs)
- 101 (pve0)
- QNAP2-Backups (pve)
- QNAP2-ISOSImages
- local (pve)
- local-zfs (pve)

Node 'pve' Summary

Package versions

pve (Uptime: 00:59:06)

CPU usage	26.79% of 4 CPU(s)	IO delay	1.25%
Load average	1.08,1.02,0.70		
RAM usage	6.59% (4.14 GiB of 62.87 GiB)	KSM sharing	0 B
HD space(root)	0.01% (714.63 MiB of 7.01 TiB)	SWAP usage	0.00% (0 B of 8.00 GiB)

CPU usage

Server load

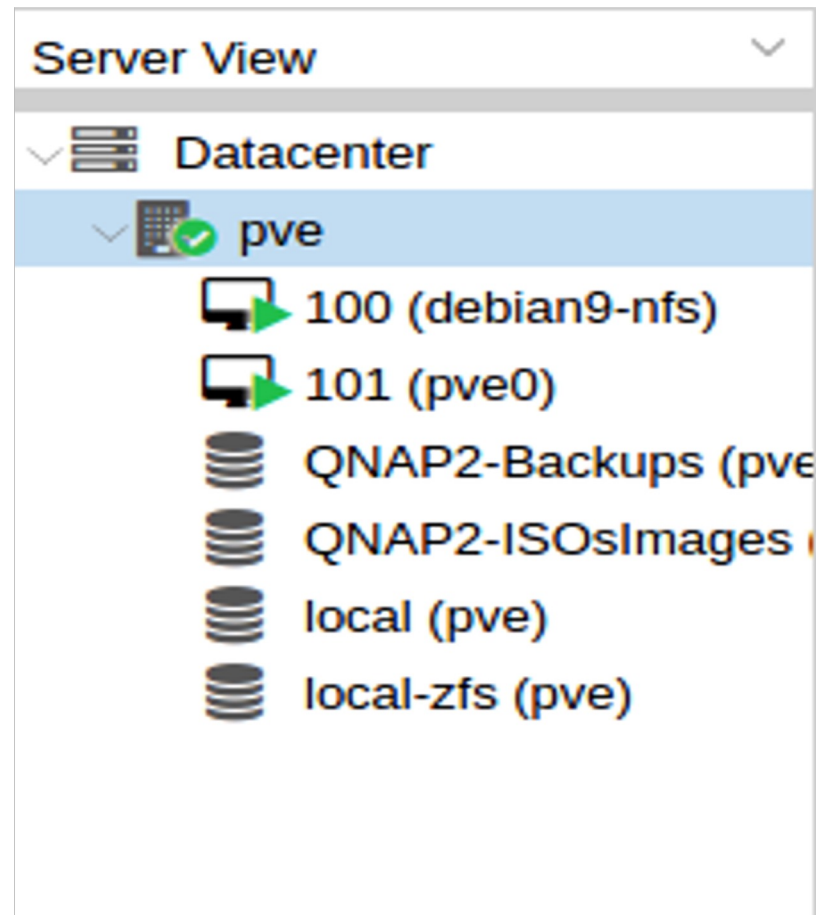
Memory usage

Tasks Cluster log

Start Time ↓	End Time	Node	User name	Description	Status
Dec 12 15:54:57		pve	root@pam	VM/CT 101 - Console	
Dec 12 15:54:56	Dec 12 15:55:07	pve	root@pam	VM 101 - Start	OK
Dec 12 15:54:33	Dec 12 15:54:37	pve	root@pam	VM/CT 101 - Console	Error: command '/bin/mc6 -l -...
Dec 12 15:54:33	Dec 12 15:54:37	pve	root@pam	VM/CT 101 - Console	Error: command '/bin/mc6 -l -...
Dec 12 15:54:32	Dec 12 15:54:37	pve	root@pam	VM 101 - Stop	OK

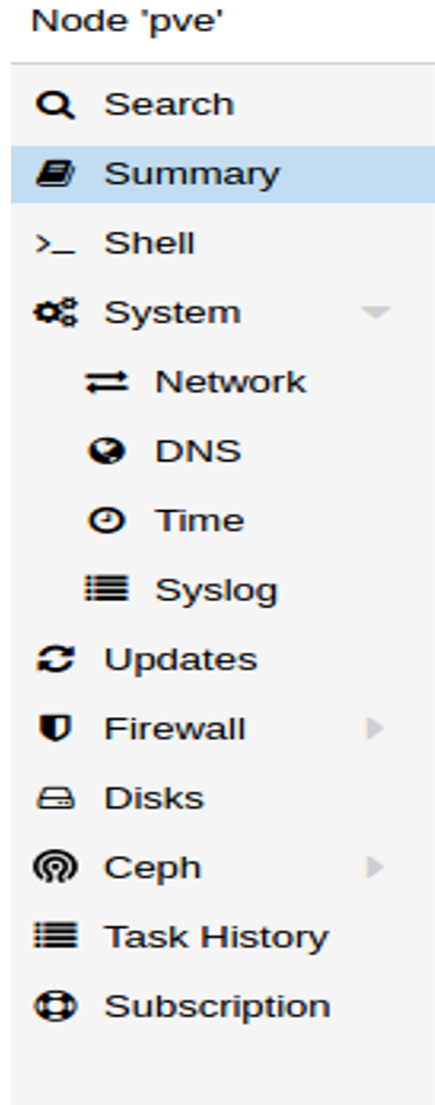
Elements of the GUI

Datacenter and Host overview



GUI elements cont'd

Node overview

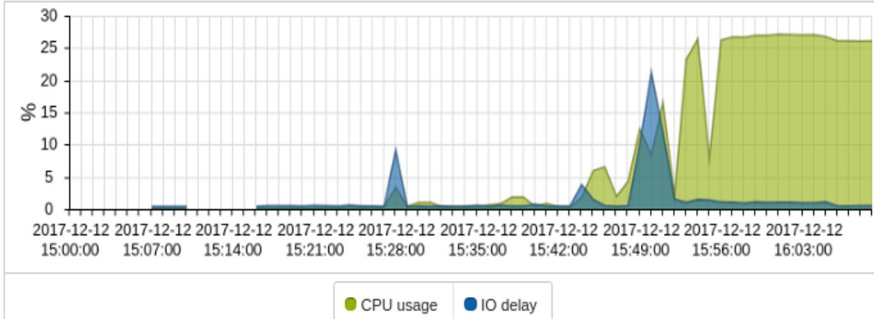


System Summary

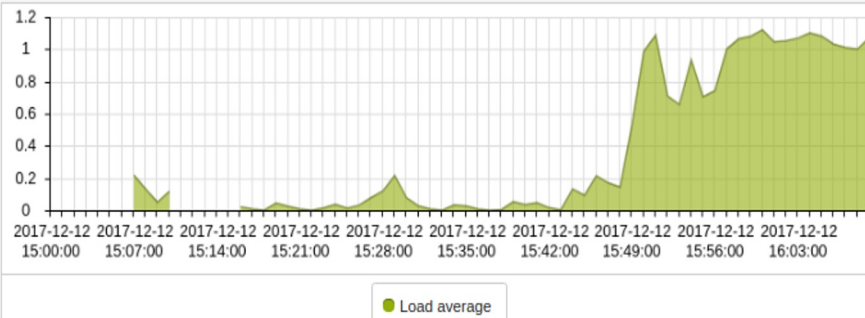
pve (Uptime: 01:04:21)

CPU usage	26.78% of 4 CPU(s)	IO delay	0.74%
Load average	1.21,1.07,0.83		
RAM usage	6.58% (4.14 GiB of 62.87 GiB)	KSM sharing	0 B
HD space(root)	0.01% (714.63 MiB of 7.01 TiB)	SWAP usage	0.00% (0 B of 8.00 GiB)
CPUs		4 x Intel(R) Xeon(R) CPU E5-2407 v2 @ 2.40GHz (1 Socket)	
Kernel Version		Linux 4.4.35-1-pve #1 SMP Fri Dec 9 11:09:55 CET 2016	
PVE Manager Version		pve-manager/4.4-1/eb2d6f1e	

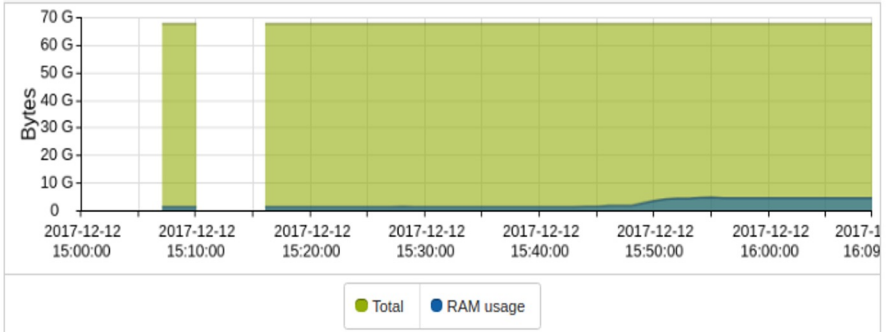
CPU usage



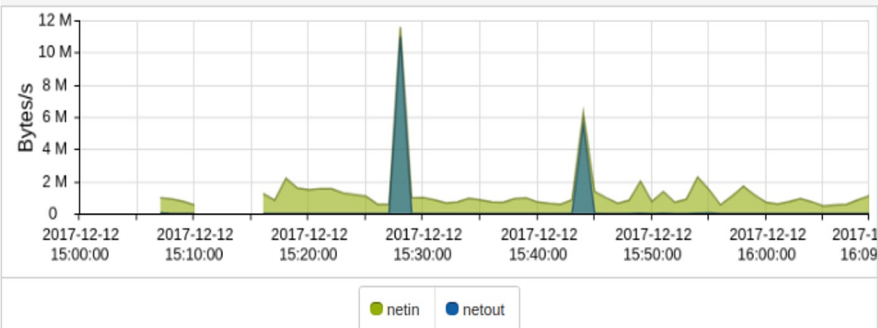
Server load





Memory usage





Network traffic




You are logged in as 'root@pam' 


 Help

 Create VM


 Create CT


 Logout

 Restart


 Shutdown

 Shell ▾

 More ▾

 Help

node 'pve' - Google Chrome

 Not secure | ~~https~~://172.21.200.104:8006/?console=shell&novnc=1&vmid=0&vmname

Connected (encrypted) to: VNC Command Terminal

root@pve:~#

System Services

<ul style="list-style-type: none">SearchSummaryShellSystemNetworkDNSTimeSyslogUpdatesFirewallDisksCephTask HistorySubscription	Start	Stop	Restart
	Name ↑	Status	Description
	corosync	dead	Corosync Cluster Engine
	cron	running	Regular background program processing daemon
	ksmtuned	running	Kernel Samepage Merging (KSM) Tuning Daemon
	postfix	running	LSB: Postfix Mail Transport Agent
	pve-cluster	running	The Proxmox VE cluster filesystem
	pve-firewall	running	Proxmox VE firewall
	pve-ha-crm	running	PVE Cluster Ressource Manager Daemon
	pve-ha-lrm	running	PVE Local HA Ressource Manager Daemon
	pvedaemon	running	PVE API Daemon
	pvefw-logger	running	Proxmox VE firewall logger
	pveproxy	running	PVE API Proxy Server
	pvestatd	running	PVE Status Daemon
	spiceproxy	running	PVE SPICE Proxy Server
	sshd	running	OpenBSD Secure Shell server
syslog	running	System Logging Service	
systemd-tim...	running	Network Time Synchronization	

VM Installation

- Hard drives
 - Virtio-scsi and scsi is best performance option
 - On windows VMs this can be a chore as it is necessary to use a second boot CD to install virtio drivers
 - No-cache is best compromise option for local disks
 - Write back is fastest, though unsafe on Ceph

VM installation

- Memory

- Fixed allocation with Ballooning is the best way to allocate RAM.
 - Over-provisioning is possible but dangerous as guests may crash if RAM not available
- Auto-allocation - required RAM may take up to 30sec to be available
- It is best to leave swap enabled as this way swap is a last option only before crashing...

VM Backups

- Backup types:
 - Snapshot leaves guest running and intercepts all write ops, writes them to backup if block is already backed up, then to guest but it slows guest IO down to speed of backup medium
 - Stop causes guest to shutdown, then restarts and does backup before making guest available

VM Migration

- Guests on local storage
 - migration must be done off line
 - any storage used in guest (eg ZFS) must be available on target node
- Guests can be live moved to shared, eg NFS or CEPH storage and then live migrated

VM Cloning

- Linked clones allow fast spin up of machines as only diverging blocks need to be written to disk
- Linked clones require file level storage systems, i.e. snapshot-able storage
- The conversion of a VM to a template sets the image as Read-Only

VM Imports

- OVA Import
 - Unpack the OVA, eg onto a NAS
 - `qm help importovf` for details of import command (available from proxmox 5.1)
- Disk Import
 - `qm help importdisk`
 - `vmdebootstrap` can be used to build debian disk images programatically
 - `qm help create` for details on creating VMs programatically

VM Imports

- NB Windows disk images will not have any virtio drivers installed by default
 - Hard disk types must be SATA
 - Network devices must be E1000
 - Spice-space spice-guest-tools can be used to install all virtio drivers into Windows images
 - Spice repository on github has the source code for the installation tools

User authentication

- PAM authentication
 - per machine authentication (may be possible to integrate radius, see forums)
- Proxmox authentication server
 - replicates authentication across all nodes

Proxmox Cluster

- The Proxmox VE cluster manager pvecm is a tool to create a group of physical servers. Such a group is called a cluster
- Uses the Corosync Cluster Engine for reliable group communication, and such clusters can consist of up to 32 physical nodes or more, dependent on network latency - must be $< 2\text{ms}$
- pvecm can be used to create a new cluster, join nodes to a cluster, leave the cluster, get status information and do various other cluster related tasks

Proxmox Cluster

Grouping Proxmox hosts into a cluster has the following advantages:

- Centralized, web based management of a multi-master cluster: each node can do all management task
- pmxcfs: database-driven file system for storing configuration files, replicated in real-time on all nodes using the corosync cluster engine
- Migration of VMs and Containers between physical hosts
- Fast deployment and cluster-wide services like firewall and HA (High Availability)

High Availability

- Items managed under HA are referred to as Resources
 - HA cluster is managed by 'pve-ha-crm.service'
 - Local HA resources are managed by 'pve-ha-lrm.service'
- Guest HA is managed either through dropdown on guest window, or HA options on Datacenter and this allows a guest VM to be automatically migrated or restarted on a different node if it is detected as down e.g. because of node failure or maintenance

High Availability

- Ensure 'pve-ha-crm' and 'pve-ha-lrm' are both running under 'node' -> 'services'
- All migrations and other actions on HA resources are managed by the HA daemon
- Task viewer only shows status of request to HA daemon to carry out task, not of actual task.

High Availability

- Migrations (generally but particularly under HA conditions) may fail due to a number of causes
 - guest has local attached storage which is not available on target node
 - guest has NUMA (non-uniform memory access) or other CPU settings not present on target node

High Availability

- Changing HA manager state for a VM will cause the VM state to change.
- If any node hosting a HA resource loses corosync quorum
 - 'pve-ha-lrm.service' will no longer be able to write to the watchdog timer service
 - after 60 seconds, the node will reboot
 - after a further 60 seconds, the VM will be brought up on a different node

HA Groups

- Group members will prefer selected nodes if available
 - If restricted is selected, members will ONLY run on selected nodes
 - guests will be stopped by the HA manager if the node(s) become unavailable
 - If nofailback is not selected, guests will try to migrate back to a preferred node once it becomes available again

Performance Benchmarking

- iperf to test network throughput
- systat to monitor system statistics
- iostat to test IO throughput
 - some bandwidth expectations would be
 - Spinning disk: 200MBps 1000IOPS
 - SSD: 400MBps 10000IOPS
 - google 'Thomas Krenn' for some good benchmarks
 - Emmanuel Kasper also has numbers on his own homepage