

Операционные системы Ч. II

Контейнерная виртуализация
и комплексные решения на основе
открытых технологий

Основа контейнерной виртуализации

Главная особенность - одно общее ядро для всех контейнеров. Это определяет основные свойства, накладывает некоторые ограничения и обеспечивает особые возможности

Достоинства и недостатки контейнерной виртуализации

Достоинства:

- не требуется аппаратная поддержка виртуализации;
- низкая стоимость;
- высокая производительность;
- высокая масштабируемость;
- высокая плотность размещения.

Недостатки:

- Основной — невозможность запуска различных ОС;
- Более низкий уровень изоляции;
- Неполная совместимость с некоторым ПО.

Примеры продуктов — Solaris Zones

Solaris 10 - global zone

APP 4

zone 1

APP 1

zone 2

APP 2

zone 3

APP 3



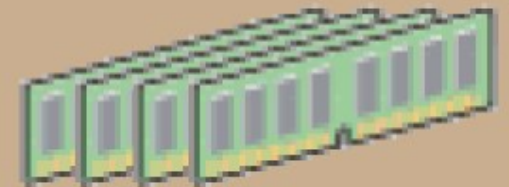
CPU



Disk



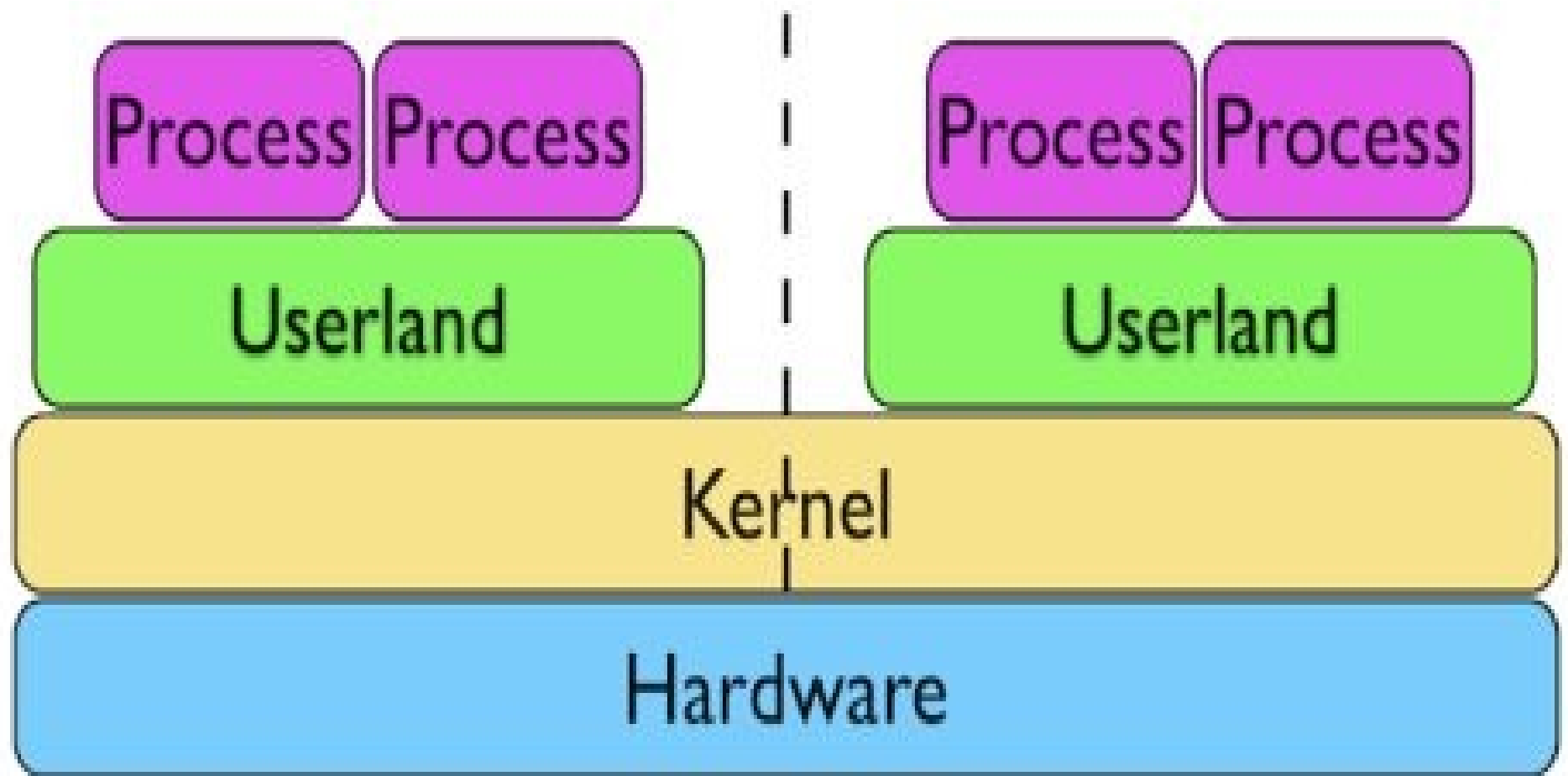
PCIe



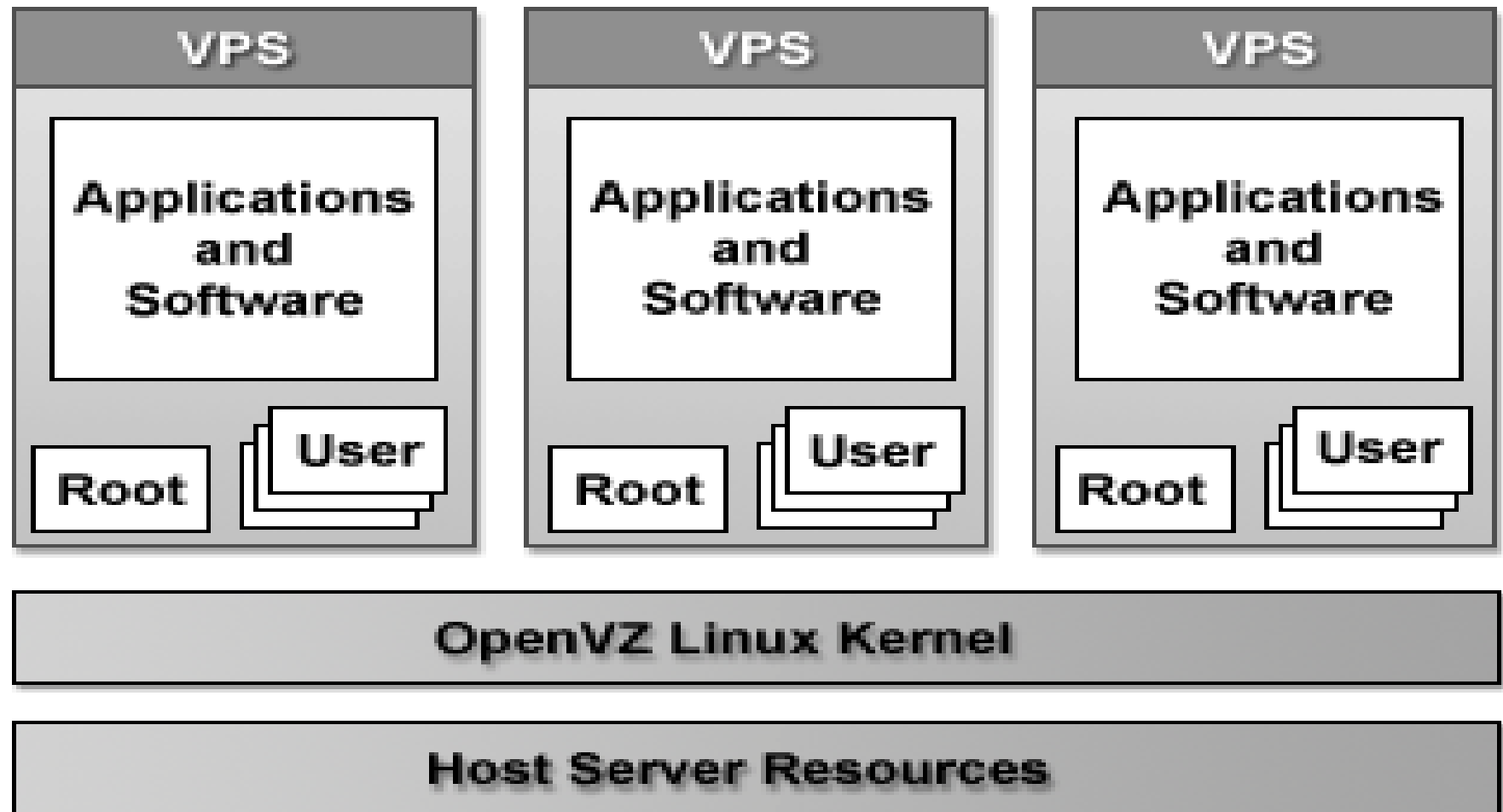
Memory

Hardware


FreeBSD Jails



Linux OpenVZ/Virtuozzo



Графическая панель управления OpenVZ

 **OpenVZ Web Panel** © Copyright 2008-2011 SoftUnity. All Rights Reserved.

Menu <<

General

Dashboard

Physical Servers

- note.lan
- rock.lan

IP Addresses

My Profile

Users

Requests

Tasks


Events Log





Logout


Help

Documentation


Support


 **Physical Server note.lan**


 OS Templates  Server Templates  Search  Up Level


Physical server statistics 


Parameter	Value
CPU load average	0.67, 0.60, 0.55
Disk usage, partition /	<div><div></div></div> 80%, 26.8 GB of 35.4 GB / 6.











Virtual servers list 

 Create virtual server

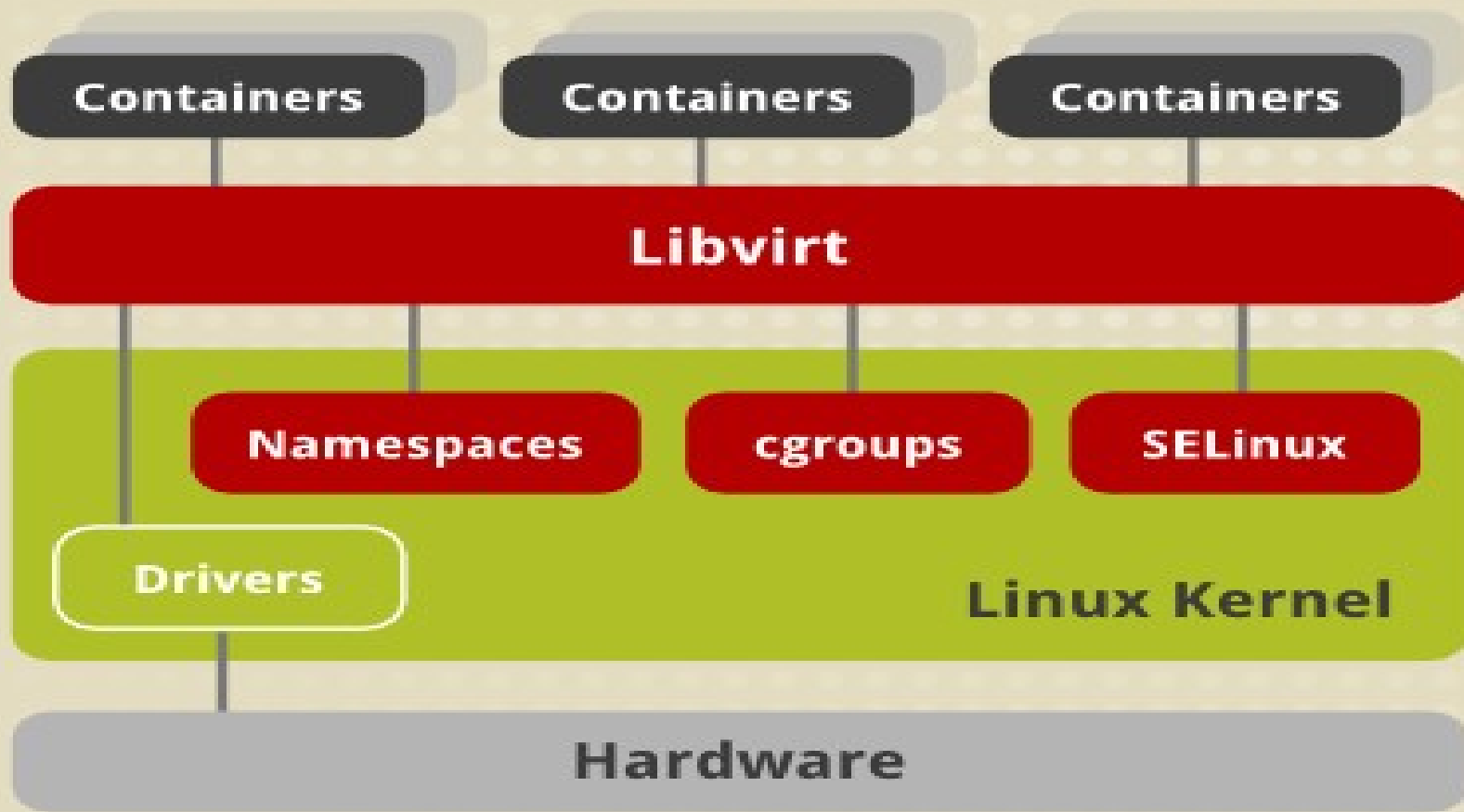
 Remove virtual server

 Change settings

 Cha

	State	VEID ▲	IP Address	Host Name
		<u>101</u>	192.168.0.101	web.note.lan
		Development server.		
		<u>102</u>	192.168.0.102	pm.lan
		<u>103</u>	192.168.0.103	site.lan
		Production web server.		
		<u>104</u>	192.168.0.104	web-proxy
		Frontend web server		
		<u>105</u>	192.168.0.105	teambox

Linux containers(LXC)



LXC Web Panel

LXC Web Panel

Logout (admin)

GENERAL

Overview

CONTAINERS

server1

server2

server3

server4

LXC SETTINGS

Networking

Check config

LXC WEB PANEL

Users

About

Ubuntu 12.04 (host)

Create CT

Clone CT

Reboot

CPU usage : 0.0%

Disk usage : 3,6G (14G free)

Memory usage : 370 / 7979 MB






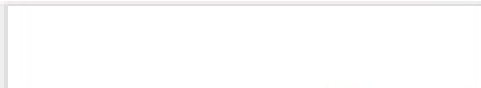












Uptime : 0 day(s) 0:28

Status	Name	Hostname	IP Address	Mem. usage	Actions
Running	server1	web	10.0.3.10	176 MB	Start Stop Freeze
	server2	sql	10.0.3.11	313 MB	Start Stop Freeze
Frozen	server3	voice	Undefined	4 MB	Start Stop Freeze
Stopped	server4	server4	Undefined		Start Stop Freeze

LXC VirtManager

Virtual Machine Manager

Open

Name	CPU usage
fridge.lappyfamily.net (LXC)	
 lab Running	
 lemon Running	
fridge.lappyfamily.net (QEMU)	
 WIN7HBI386EN Shutoff	
 WIN7PNX64EN Running	
 clonet-test-master Shutoff	
 corruption-test-clone Shutoff	
 fedora-12 Running	
 kubuntu-12.04 Shutoff	
 lucidity Shutoff	

Proxmox_VE — полная виртуализация через KVM + контейнеры LXC

PROXMOX Virtual Environment 4.4-1/eb2d6f1e Search You are logged in as 'root@pam' Help Create VM Create CT

Server View Datacenter

Search Summary Options Storage Backup Permissions Users Groups Pools Roles Authentication HA Firewall Support

117 (cent 101 (win 103 (jess 104 (whe 120 (win2 122 (free 124 (bare ceph3 (p ceph3-lx iso-temp local (pv local-lvm testlab-m ucs-zfs (pve-2-51 100 (deb 102 (fedc 114 (jess 108 (nets ceph3 (p ceph3-lx iso-temp local (pv local-lvm testlab-m pve-2-52

Datacenter Health

Status	Nodes	Ceph
	✓ Online 3 ✗ Offline 0	 HEALTH_OK

Cluster: mctestcluster1, Quorate: Yes

Guests

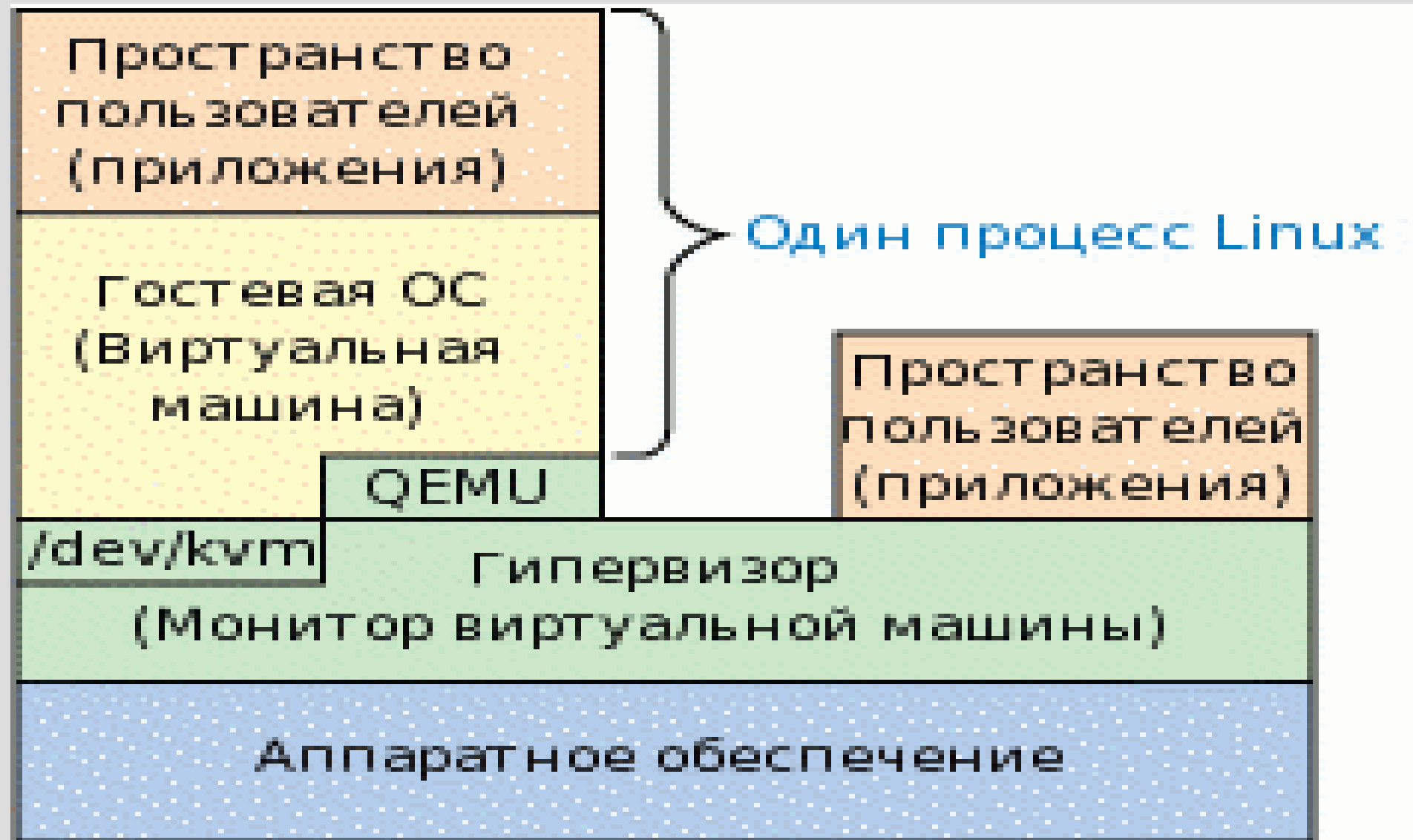
Virtual Machines	LXC Container
Running 8 Stopped 10	Running 6 Stopped 0

Cluster Resources

CPU	Memory	Storage
1% of 24 CPU(s)	22% 20.76 GiB of 93.93 GiB	30% 12.52 TiB of 41.88 TiB

Logs

Архитектура KVM



Управление KVM через Virtual Manager

Консоли VM и графики загрузки

The screenshot displays the Virtual Machine Manager (Virt-Manager) interface. The main window is titled "Менеджер виртуальных машин". The top menu bar includes "Приложения", "Переход", and "Система". The top status bar shows system information: "-5 °C", "en", and "Втр, 24 Фев, 21:08".

The left sidebar lists the virtual machines:

- Win10_Tech_Preview_32 (Работает)
- Win7_Lab_Use_Only (Работает)
- lin (Выключена)
- lin2 (Работает)

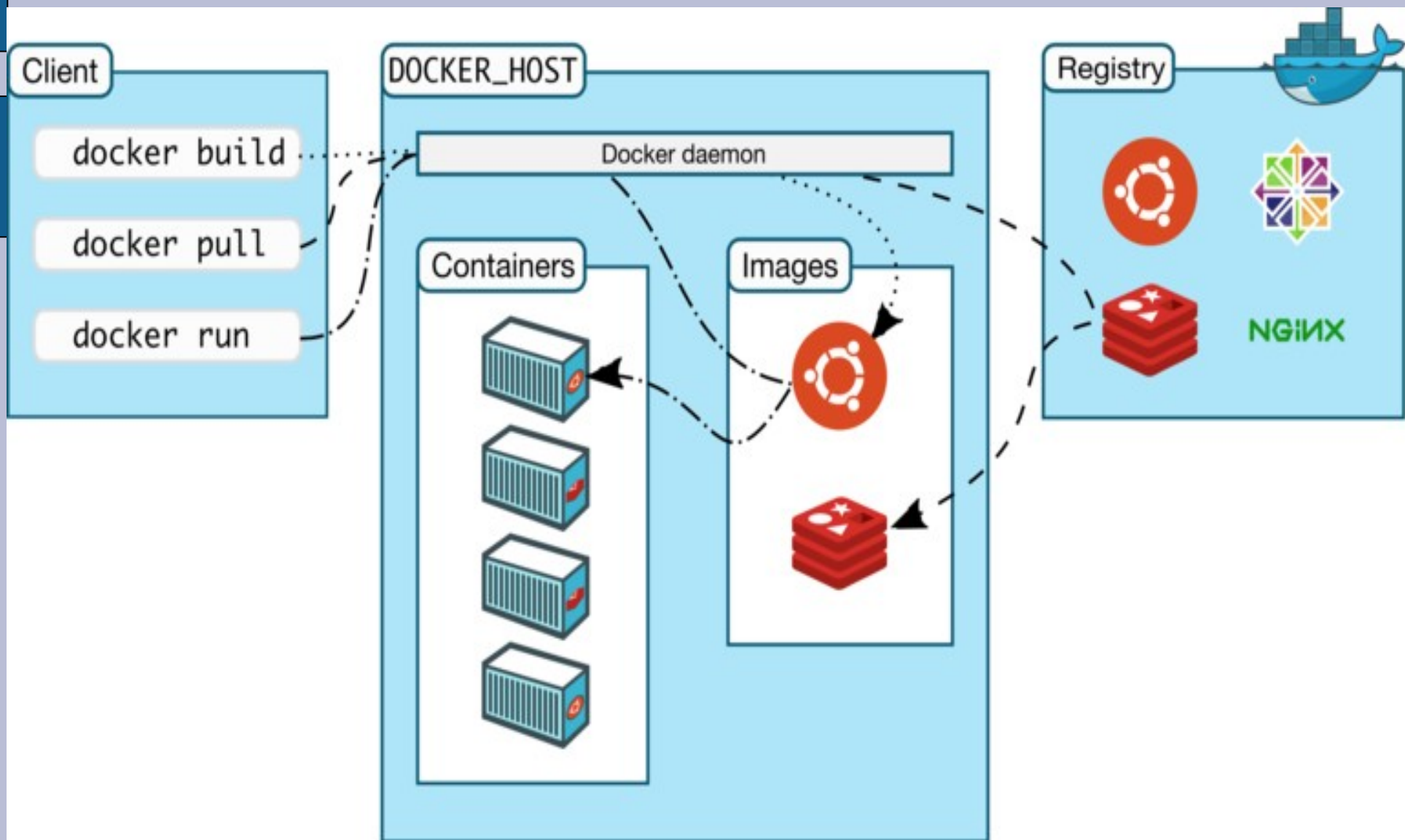
The right pane shows performance graphs for the selected VM (Win10_Tech_Preview_32). The graphs are organized into a grid with columns for "Использование CPU", "Host CPU usage", "Дисковый ввод/вывод", and "Сетевой ввод/вывод".

Below the main window, two smaller windows are visible:

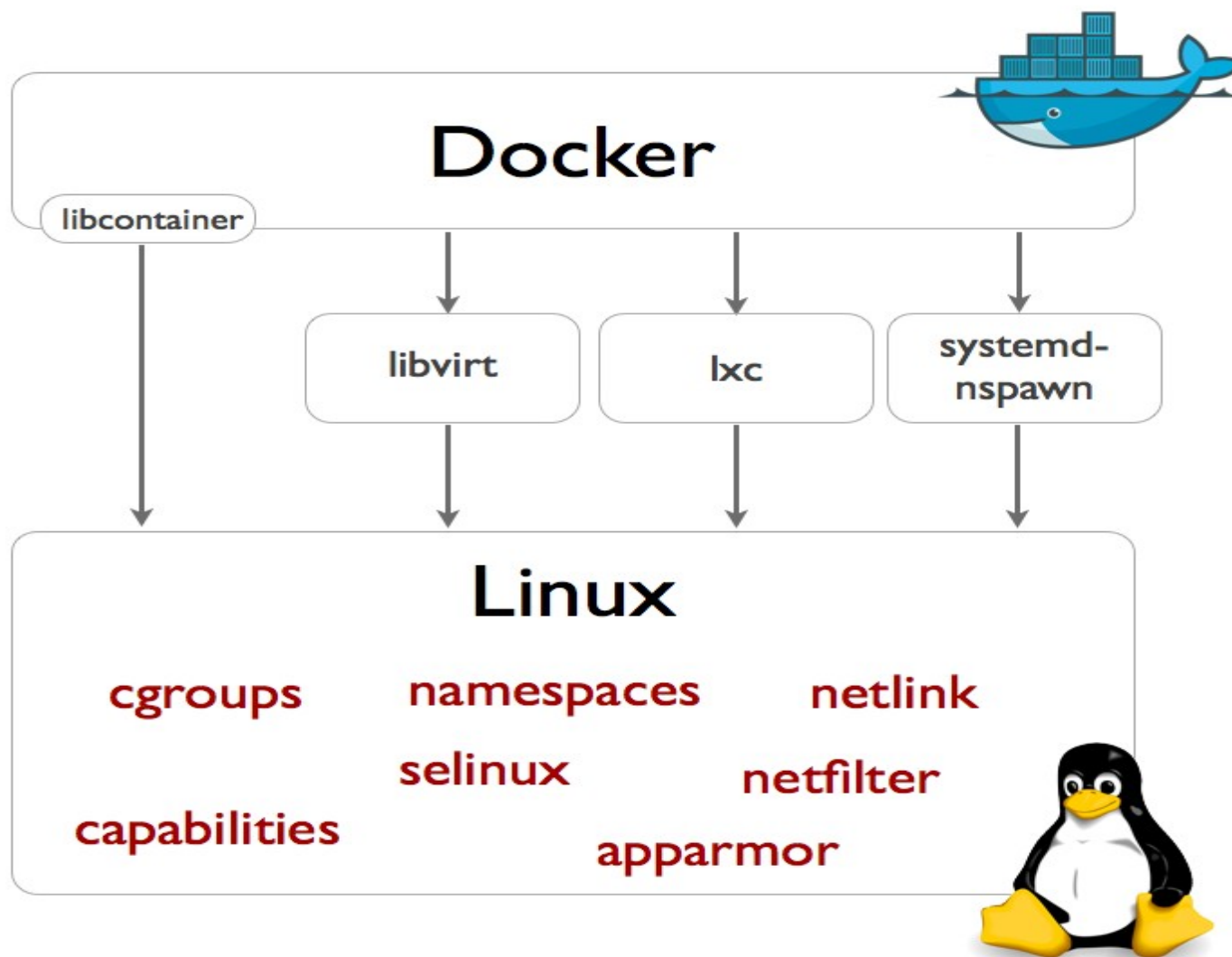
- "Win7_Lab_Use_Only Виртуальная машина" showing a Windows 7 desktop with the classic blue background and logo.
- "Win10_Tech_Preview_32 Виртуальная машина" showing a Windows 10 desktop with a waterfall background. A file explorer window is open, displaying a folder named "Корзина" (Recycle Bin).

The bottom taskbar shows the running applications: "Менеджер вирт...", "Win7_Lab_Use_O...", "Win10_Tech_Previ...", and "lin2 Виртуальная...". The system tray on the right shows the temperature (30 °C, 44 °C), network status, and CPU usage (800 MHz).

Архитектура Docker



Базовые технологии Docker



Базовые технологии Docker

NAMESPACES:

Docker использует namespaces следующим образом:

- **pid** namespace: изоляция процессов.
- **net** namespace: управление сетевыми интерфейсами.
- **ipc** namespace: управление доступом к ресурсам IPC.
- **mnt** namespace: управление точками монтирования файловых систем.
- **uts** namespace: изоляция идентификации ядра и версии. (UTS: Unix Timesharing System).

CGROUPS:

Cgroups ограничивает приложения определённым набором ресурсов.

Управление группами обеспечивает эффективное разделение ресурсов между контейнерами.

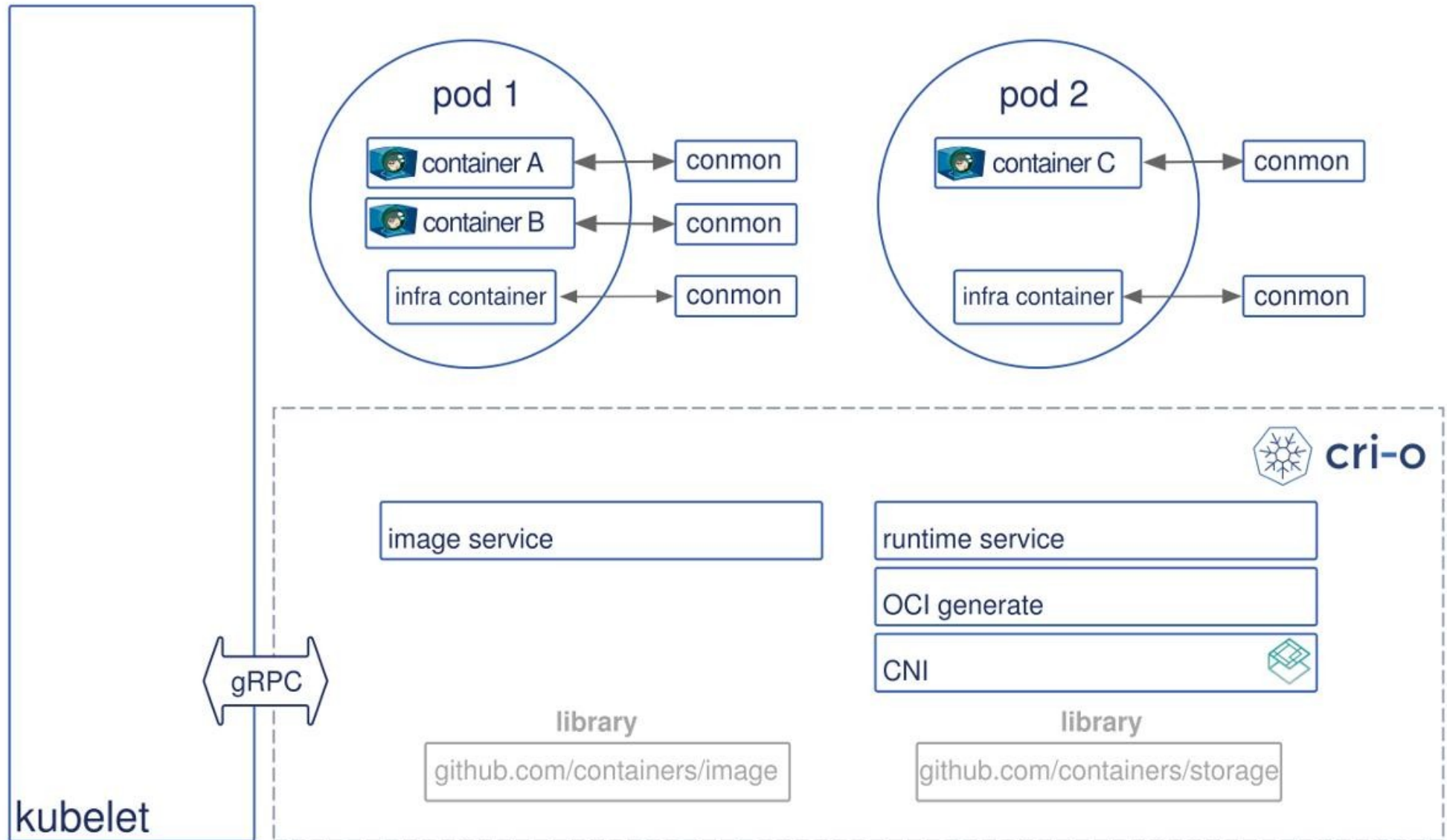
Union file systems:

Обеспечивают функционирование файловых систем в контейнерах. Docker может использовать различные варианты UnionFS - AUFS, btrfs, vfs, и DeviceMapper.

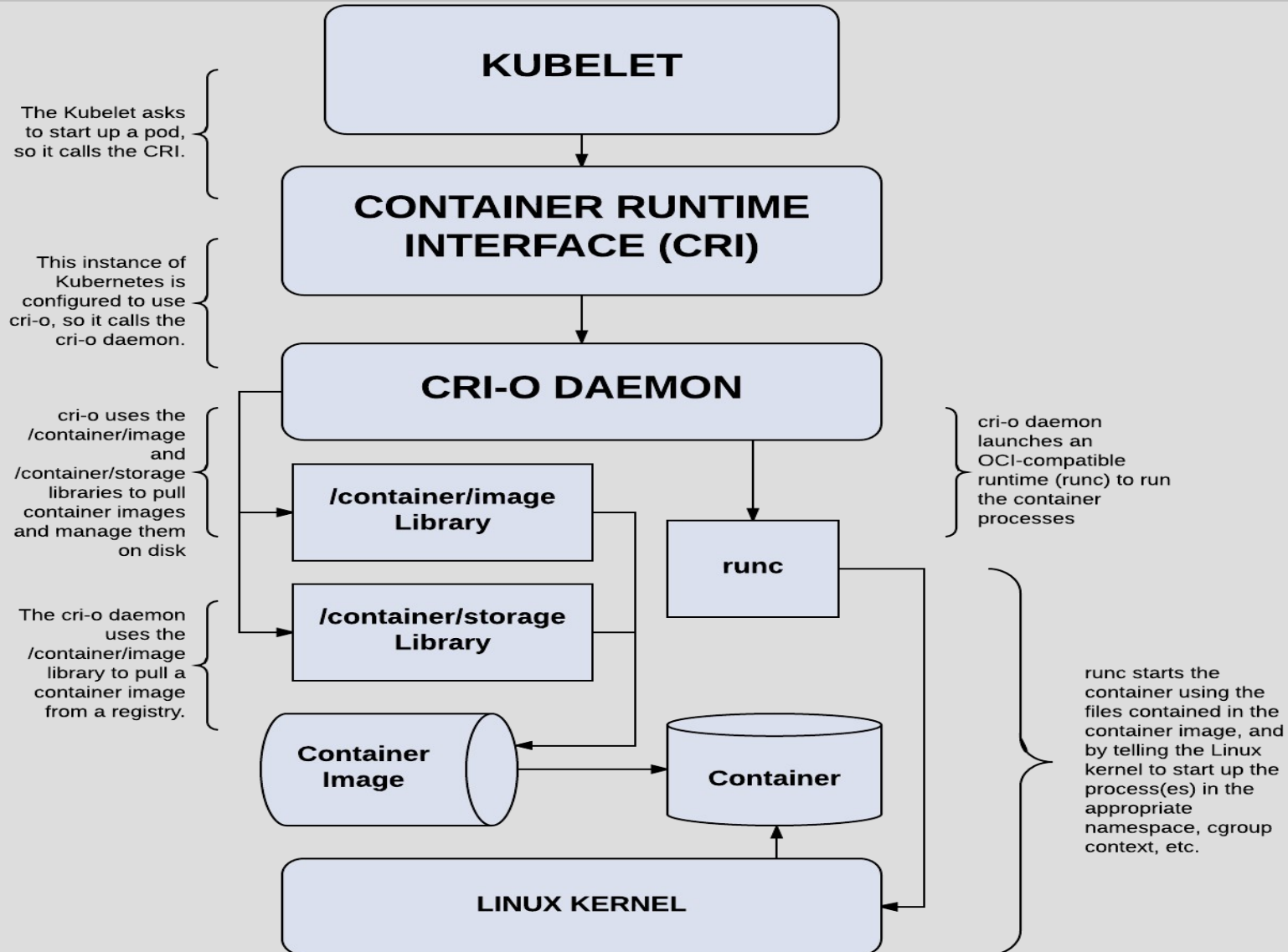
Container format:

Docker комбинирует namespaces, cgroups, и UnionFS в некоторую оболочку-
container format. По умолчанию в текущей версии container format —
libcontainer.

CRI-O и Kubernetes



Связь Kubernetes с runC и низкоуровневыми библиотеками



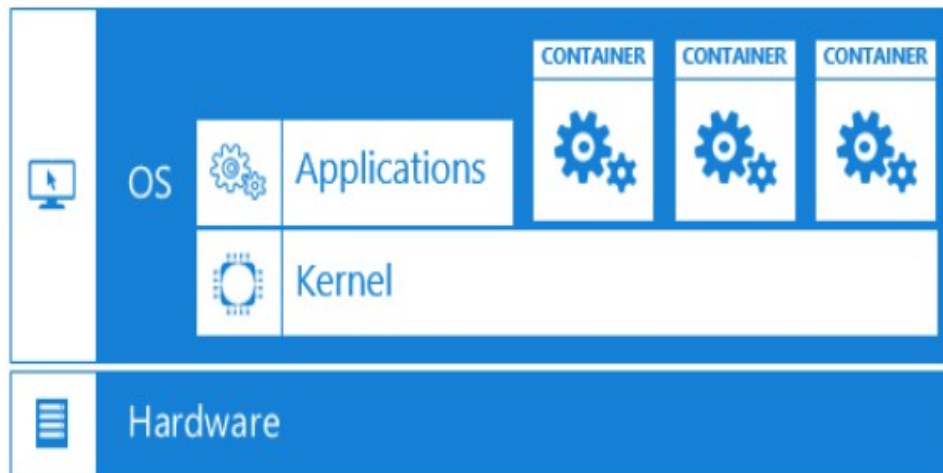
Функционирование CRI-O и Kubernetes

Kubernetes обращается к kubelet для запуска пода, а kubelet перенаправляет этот запрос к демону CRI-O через интерфейс Kubernetes CRI.

- CRI-O с помощью библиотеки `containers/image` забирает образ из реестра.
- CRI-O с помощью библиотеки `containers/storage` распаковывает загруженный образ в корневую файловую систему контейнера (хранимую в COW).
- Создав `rootfs` для контейнера, CRI-O с помощью утилиты `generate` из OCI Runtime Tools готовит JSON-файл, соответствующий спецификации OCI runtime и содержащий описание, как запустить контейнер.
- CRI-O запускает исполняемую среду (например, `runc`) с использованием созданной спецификации.
- Контейнер попадает в поле зрения мониторинга, осуществляемого с помощью утилиты `conmon` (отдельный процесс в системе), которая также обслуживает логирование для контейнера и записывает код завершения процесса контейнера.
- Сеть для пода настраивается с помощью любого из плагинов CNI.

Контейнеры Windows

Контейнеры = виртуализация на уровне операционной системы

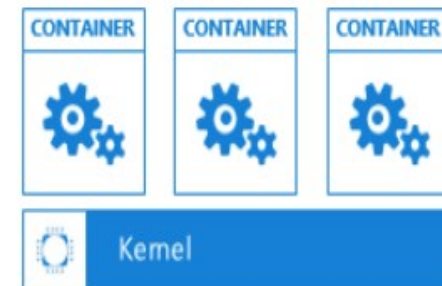


Традиционные виртуальные машины = аппаратная виртуализация



Контейнеры Windows Server

Максимальная скорость и плотность



Контейнеры Hyper-V

Изоляция и производительность

