Student Presentation

Linux Containers (LXC)

Lightweight virtualisation alternative to VMs

Elvin Sindrilaru

CERN

Thematic CERN School of Computing 2014

CERN School of Computing

Outline

- Linux namespaces and control groups (cgroups)
- Linux containers (LXC)
- Docker – LXC high level wrapper
- Containers demo

CERN School of Computing

Linux namespaces

- The purpose of a namespace is to **wrap** a particular **global system resource** in an **abstraction** that makes it appear to the process within the namespace that they have their **own isolated instance of the global resource**.

- Currently 6 namespaces implemented in the Linux Kernel:
  - **Mount**
  - **UTS (Unix Time-sharing System)**
  - **IPC (Inter-process communication)**
  - **PID**
  - **Network**
  - **User namespace**

CERN School of Computing

Linux cgroups

- **Cgroups** allow allocating **resources** to **user-defined groups of processes** running on the system
- **Cgroup subsystems** (resources controllers) = kernel modules aware of cgroups which allocate varying levels of system resources to cgroups
- Everything is exposed through a **virtual filesystem**: /cgroups, /sys/fs/cgroup … - mountpoint may vary
- Currently up to 10 subsystems:
  - **blkio** – set limits on input/output access to/from block devices such as physical drives
  - **cpuset** – assign individual CPUs to tasks in a cgroup
  - **memory** – limits on memory used by tasks in a cgroup

CERN School of Computing

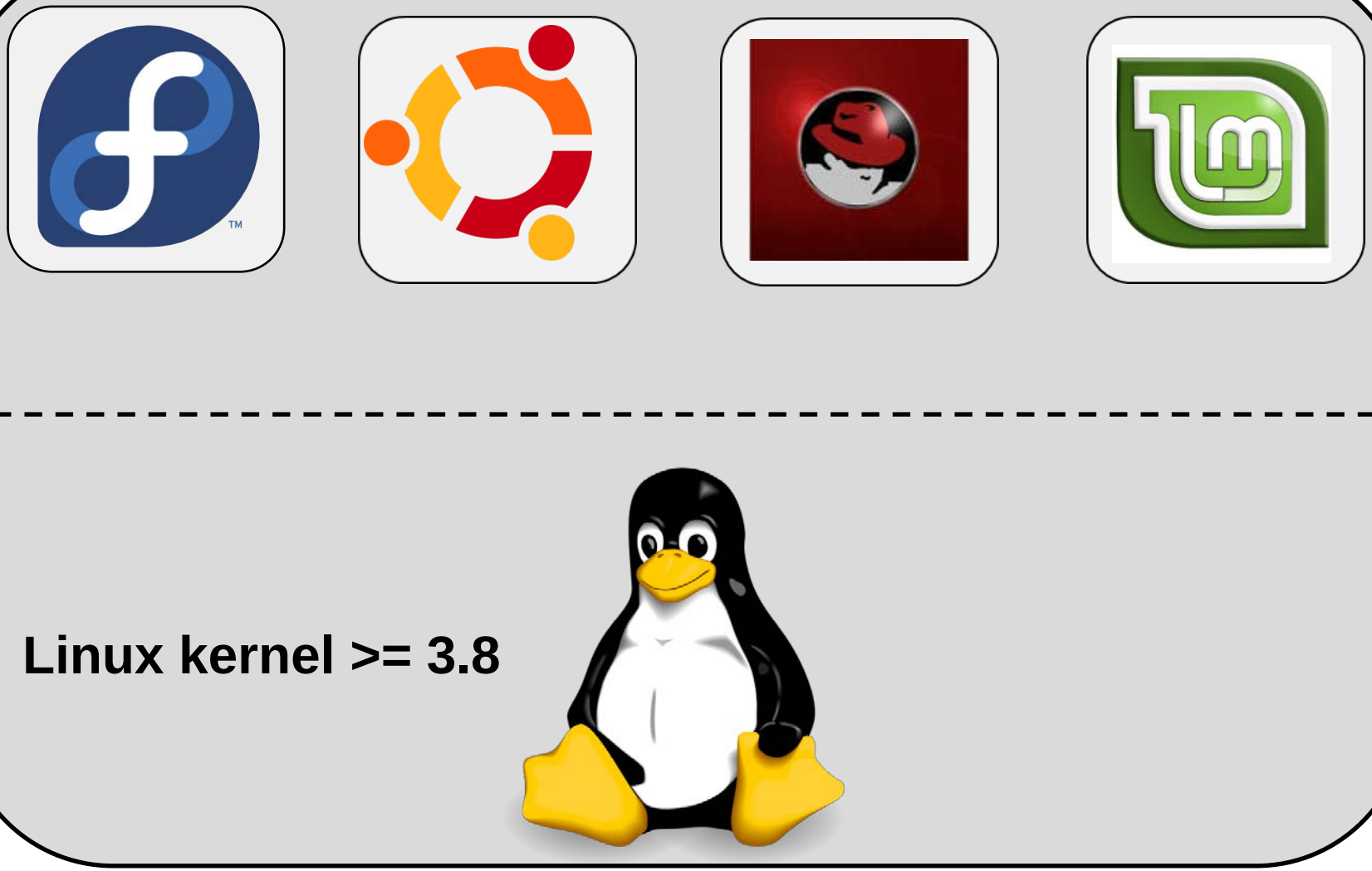
Linux containers - LXC

- **Containers**
  - tool for **lightweight virtualization**
  - provides a group of processes the **illusion** that they are the only ones running on the system
- **Advantages** in comparison to traditional VM:
  - Fast to deploy - seconds
  - Small memory footprint - MBs
  - Complete isolation without a hypervisor

Namespaces + Cgroups => Linux containers

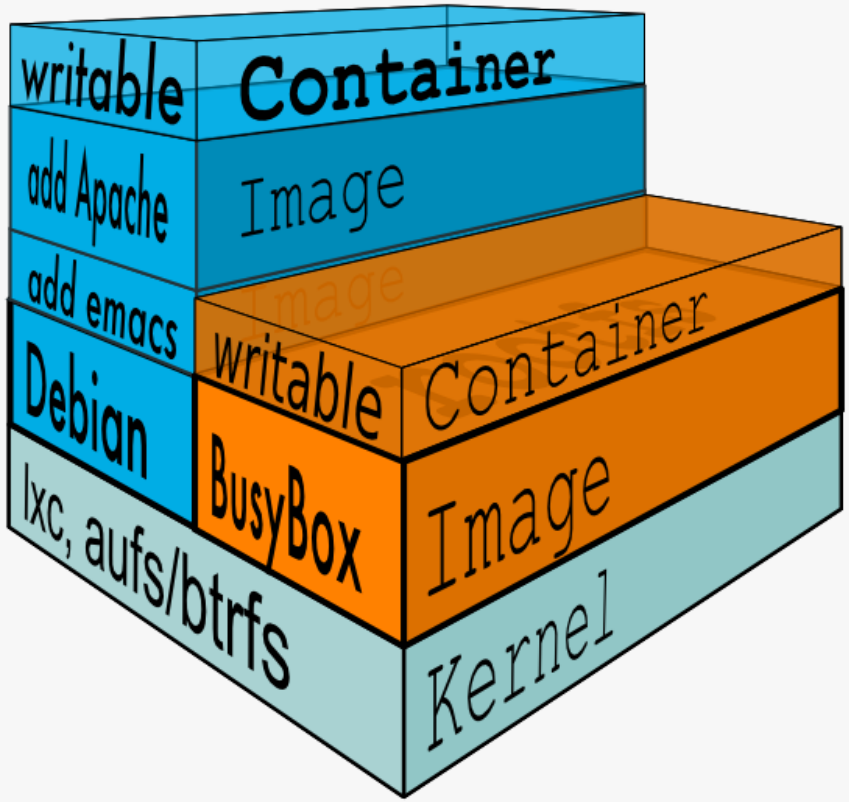
CERN School of Computing

Containers on a host machine



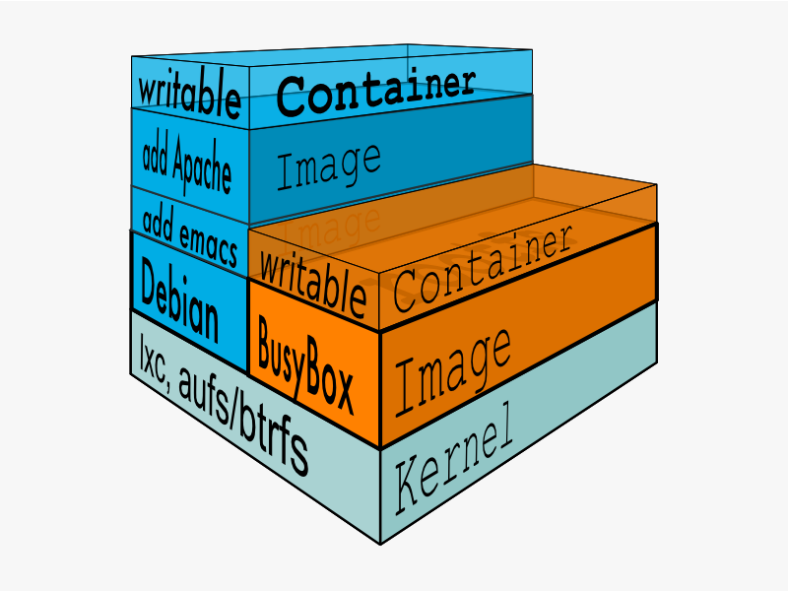
Docker– LXC wrapper

- "Open-source project to easily create **light-weight, portable, self-sufficient containers** from any application"



Docker – Layers

- **Union File System** – union of read-write layer and all read-only layers
- **Docker Image** - read-only layer, basically the root filesystem where lxc containers run in



- All modifications go into the RW layer

Docker Containers

- **Container**
  - Read-write layer
  - Information about Parent Image (RO layers)
  - Unique id + network configuration + resource limits
  - Exited container preserves the file system state but not the memory state
- Inside it looks like a VM, outside looks like a normal process
- Containers have state**: running / exited**
- Containers can be promoted to images: “docker commit”
- ➢ Takes a **snapshot** of the whole filesystem **(RW+RO)**

Containers Demo