

Deploy and Containers


Fish
jfsc@cin.ufpe.br


THE ATTENDANTS WILL LEARN...

- Container
 - Provisioning and Virtualization;
 - Scalability Issues;
 - Containers;
 - Namespaces;
 - Cgroups
 - Containers and the Pipeline
- Docker
 - HANDS ON!!! MOM

Scalability Issues

pls, me escrevam aqui 3 problemas relacionados a escalabilidade de aplicacoes


 **Júlio Melo** 7:30 AM
Seria perfeito se pudéssemos usar o Grad fish, porque eu não tenho laptop.


 **Rafael Jordão** 7:31 AM
Latência é um problema, tráfego interno no datacenter, custo geral é outro problema (já que não só o custo com expansão de máquinas mas também o custo operacional precisa ser levado em conta)



4

 **Fish Carvalho** 7:33 AM
@rfpj boa!


 **Augusto Lima** 8:18 AM
Disponibilidade também seria outro problema, visto que a depender do crescimento rápido de usuários e acessos na sua aplicação, se você não consegue escalar para atender essa demanda, eventualmente a aplicação ficará indisponível ou apresentar problemas de desempenho.

 **Pedro Neto** 8:35 AM
A arquitetura que sua aplicação se encontra também pode ser um problema de escalabilidade.
Se for centralizada, descentralizada ou híbrida os problemas/cenários são distintos

 **Thiago Santos** 9:09 AM
Dependendo da distância, em caso de sistema descentralizado, a largura de banda pode ser um gargalo para garantir a escalabilidade esperada.

new mess

 **Vinicius Garcia** 2:14 PM
@channel pra usar o lab hj vai ser complicado por conta da burocracia para instalacao do docker nas máquinas linux. por descuido meu eu não enviei a solicitação pra isso em tempo hábil...

 **Carlos Melo** 2:50 PM
Segurança, de que forma será tratado com o crescimento de usuários acessos e dados gerados para a aplicação ou pela aplicação.

Scalability Issues

| | Ships within ... | Manual deployment takes ... | Automated deployment takes ... | Boots in ... |
|-----------------------------------|-------------------------|------------------------------------|---------------------------------------|---------------------|
| Bare Metal | days | hours | minutes | minutes |
| Virtualization | minutes | minutes | seconds | less than a minute |
| Lightweight Virtualization | seconds | minutes | seconds | seconds |

image credit: Jerome Petazzoni from dotCloud)

<http://www.socallinuxexpo.org/sites/default/files/presentations/jerome-Scale11x%20LXC%20Talk.pdf>

Scalability Issues

On a typical physical server, with average compute resources, you can easily run:

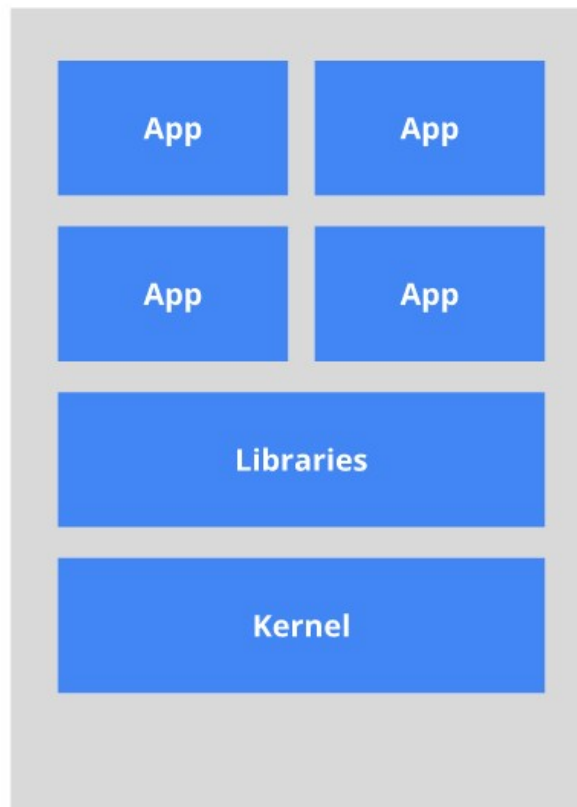
- 10-100 virtual machines
- 100-1000 containers

On disk, containers can be very light.
A few MB — even without fancy storage.

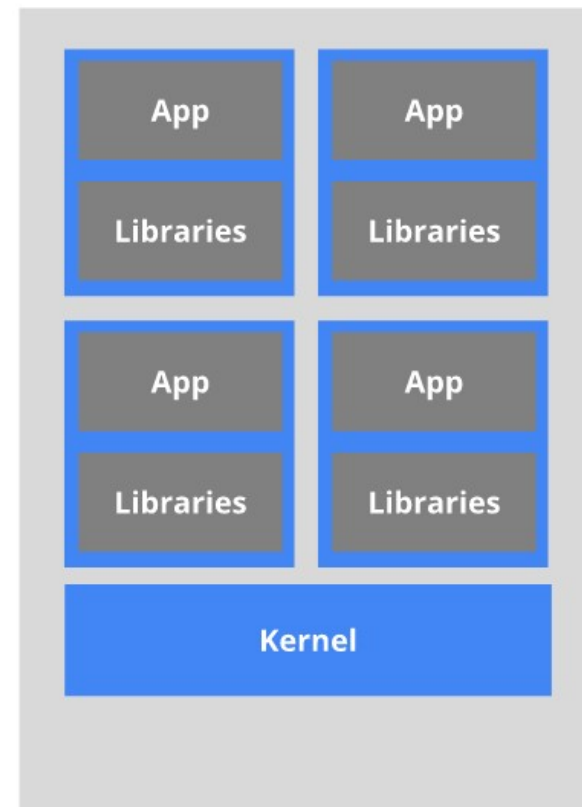
image credit: Jerome Petazzoni from dotCloud)

<http://www.socallinuxexpo.org/sites/default/files/presentations/Jerome-Scale11x%20LXC%20Talk.pdf>

Provisioning



*Heavyweight, non-portable
Relies on OS package manager*

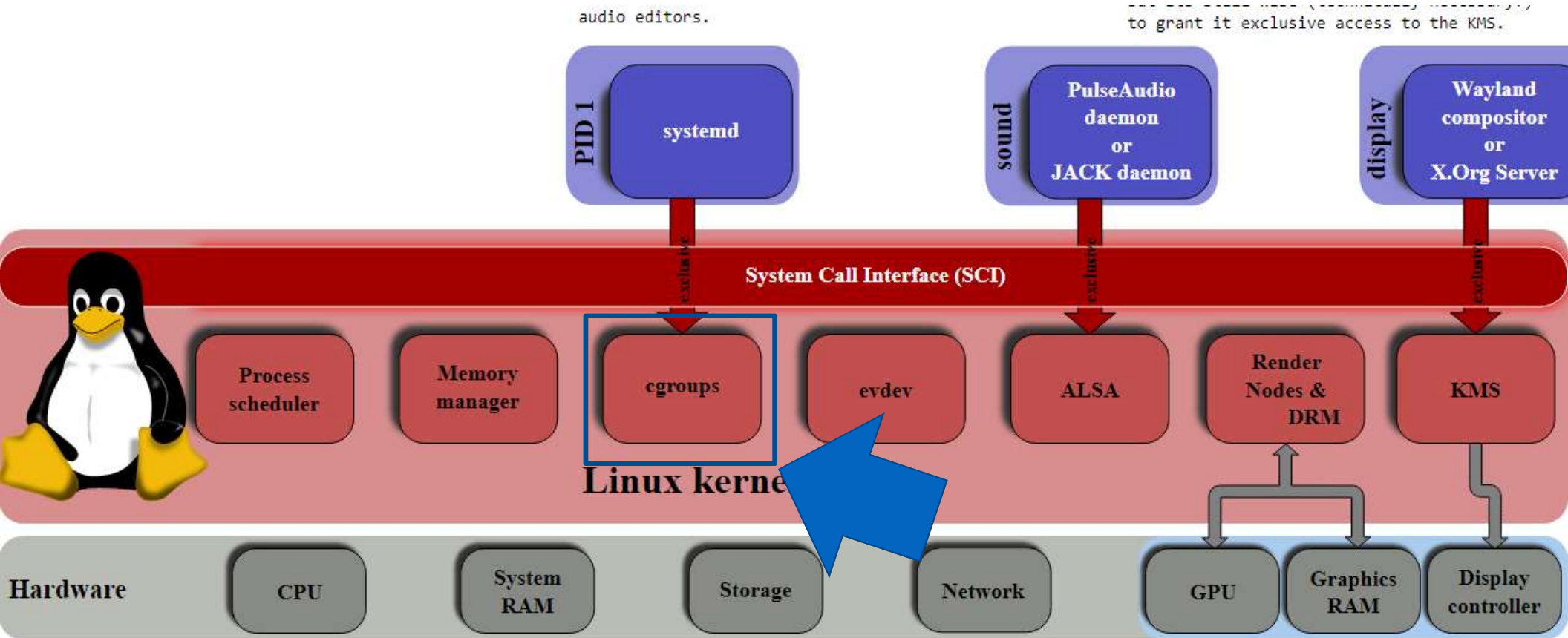


*Small and fast, portable
Uses OS-level virtualization*

Containers

- LxC (Linux Containers)
- Docker <=(OUR FOCUS)
- Warden (cloud foundry)
- Solaris Containers
- (Free) BSD Jails
- Linux V-Server
- Workload Partitions
- Sandboxie
- iCore Virtual Accounts
- WPARS
- rkt

Demystifying containers



Edited from wiki: <https://en.wikipedia.org/wiki/Cgroups>

What is Cgroup?

There are multiple efforts to provide process aggregations in the Linux kernel, mainly for resource-tracking purposes. Such efforts include cpusets, CKRM/ResGroups, UserBeanCounters, and virtual server namespaces. These all require the basic notion of a grouping/partitioning of processes, with newly forked processes ending up in the same group (cgroup) as their parent process.

The kernel cgroup patch provides the minimum essential kernel mechanisms required to efficiently implement such groups. It has

<https://www.kernel.org/doc/Documentation/cgroup-v1/cgroups.txt>

Demystifying containers

<https://jvns.ca/blog/2016/10/10/what-even-is-a-container/>