

Trabalho de Conclusão de Curso

Container Linux, uma Implementação Web Amigável

Marco Otávio Duarte de Almeida

Brivaldo Alves da Silva Junior

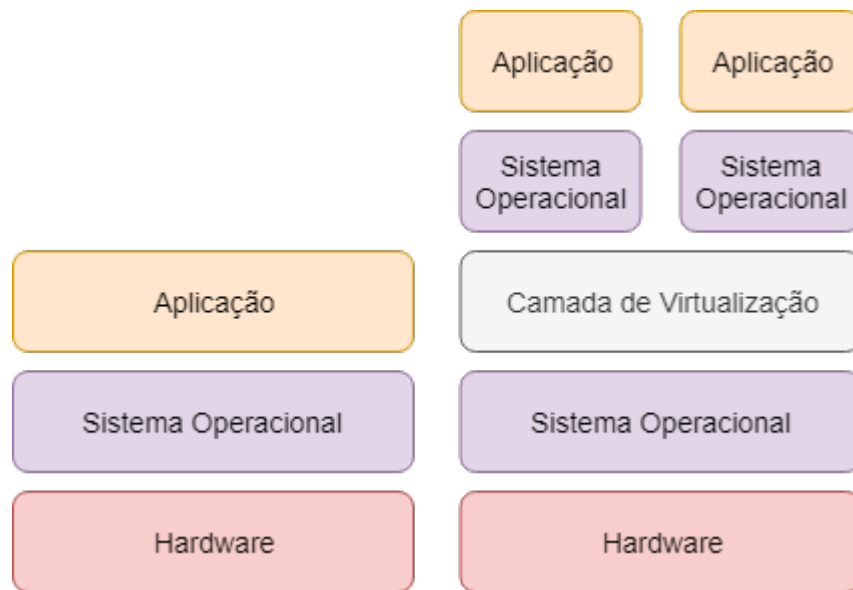
Motivação

- Fornecer aos usuários um ambiente seguro e rápido para fins de pesquisa, experimentação ou teste.
- Alternativas como VirtualBox, Docker. Vmware podem elevar o custo com *software* e *hardware*.
- Isso permitirá, por exemplo, aos acadêmicos da Facom a criação de ambientes para execução de projetos nas disciplinas de Banco de Dados, Redes de Computadores, Segurança de Redes etc.

O que é virtualização?

- Virtualização remete a ideia de máquinas potentes executando outras menores.
- É a técnica de separar uma aplicação ou sistema do (*software*) do meio físico (*hardware*).
- Essa ideia aparece a primeira vez quando os sistemas eram executados em *mainframes*.
- Com a rápida evolução dos computadores pessoais agora pode-se virtualizar diversos sistemas ou aplicativos afim de controlar ou balancear sites e serviços de alta disponibilidade.
- Normalmente utilizar um Hipervisor, *software* que cria e gerencia máquinas virtuais.

O que é virtualização?



Arquitetura tradicional vs. Arquitetura Virtual

- A "Camada de Virtualização" é responsável pela troca de informações entre o sistema hospedeiro e o sistema hospedado.

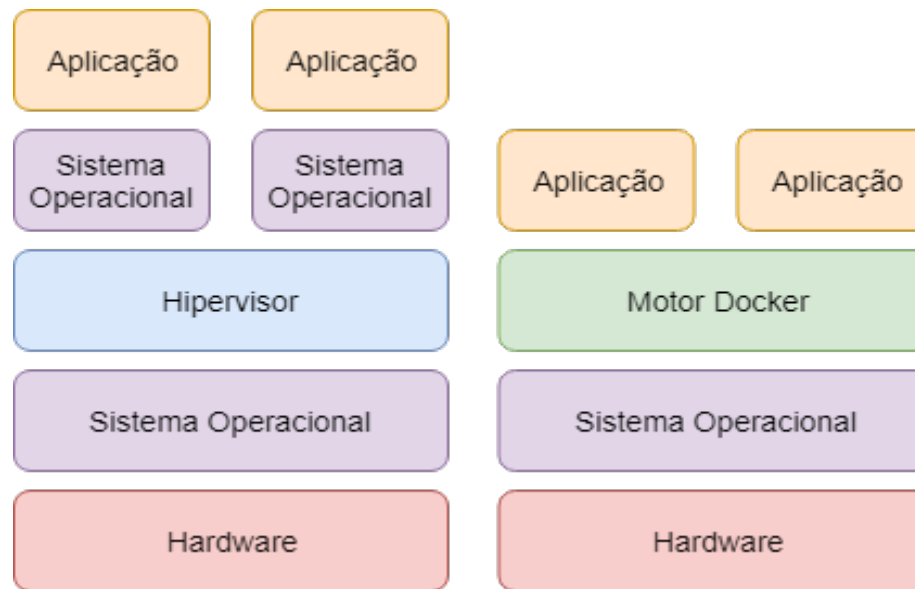
LXC – Linux Containers

- Tecnologia de container a nível de sistema operacional – sob o *Kernel Linux*.
- Possibilidade de execução de containers isolados.
- Totalmente virtualizado.
- Demais aplicativos podem executá-lo em forma de *sandbox*.
- Toda a troca de informação feita entre o sistema hospedeiro e o sistema hospedado é gerenciado pelo *Kernel Linux*.
- Excelente gerenciamento de recursos.

Docker

- Tecnologia de container.
- Fornece uma camada adicional de abstração.
- Permite empacotamento total da aplicação virtualizada.
- Altamente portátil. É possível alocar um ambiente em outro facilmente.
- Utiliza LXC (*Linux Containers*) no *backend*, é possível estabelecer limites de I/O, processador, memória etc.

Docker



Arquitetura Virtual vs. Arquitetura Docker

Vmware ESXi

- Hipervisor corporativo utilizado em servidores de grande porte.
- Não é um *software* de aplicação, é instalado diretamente no componentes computacionais, excluindo a necessidade de um sistema operacional.
- Problemas de I/O, sobrecarga de memória etc são mitigados visto que não há um sistema hospedeiro entre o *hardware* e o sistema hospedado.
- Alta disponibilidade de serviço.

Vmware ESXi

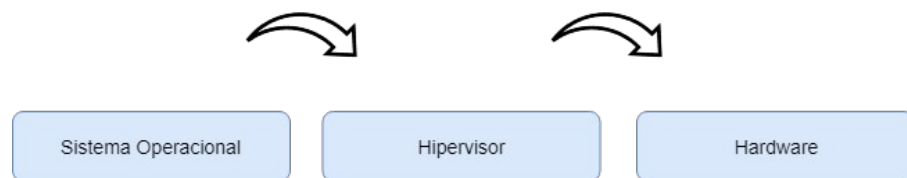


Arquitetura Vmware ESXi

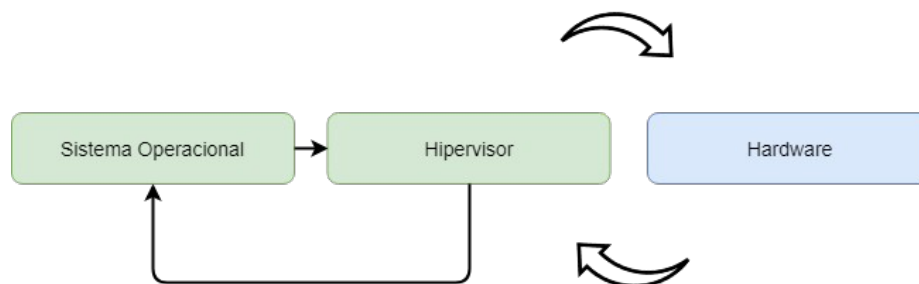
Xen

- Sistema de Para-Virtualização.
- Virtualização Total é aquela onde o há uma réplica do *hardware*.
- Todo comando executado é primeiramente interpretado pelo sistema hospedado, depois é passado par ao hipervisor, que interpreta novamente.
- Gasto de memória e processador.
- Em uma Para-Virtualização o sistema hospedado é modificado para entender que este é para virtualização. Logo, o hipervisor sempre é chamado a cada execução.

Xen



Fluxo de execução na virtualização total



Fluxo de execução na para-virtualização

LXDUI

- Nativamente o LXC/LXD não fornece uma interface amigável para gerenciamento de containers, a não ser por Interface por Linha de Comando.
- O LXDUI tem como objetivo fornecer uma interface web para criação e gerenciamento dos containers.
- Permite criação de usuários, *download* de images de sistemas (Ubuntu, Debian, CentOS etc), controle de recursos (memória, processador etc).

LXDUI - Modificações

- Criação de usuário.
- Permite a criação de usuários por dois tipos pré-definidos (User | Admin). Isso permitirá, futuramente, o isolamento total dos containers.

```
root@marco-Inspiron-7560:/home/marco/lxdui-master# python3 run.py user add -u usuario1 -p usuario1
root@marco-Inspiron-7560:/home/marco/lxdui-master# python3 run.py user list
User Accounts:
1. admin
2. usuario1
```

Criação de usuário - Antes

```
root@marco-Inspiron-7560:/home/marco/tcc-lxdui# python3 run.py user add -u usuario1 -p usuario1 -t admin
root@marco-Inspiron-7560:/home/marco/tcc-lxdui# python3 run.py user list
User Accounts:
1. usuario1. admin
```

Criação de usuário - Depois

LXDUI - Modificações

- Modo de execução.
- Isso permite que o LXDUI seja executado em modo administrador ou não.
- Quando executado em modo administrador não há restrições de visualização.
- Quando executado em modo usuário, todos os controles avançados são removidos, tais como:
 - Alteração de recursos.
 - Configuração de rede.
 - Configuração do *storage*.
 - Etc.

LXDUI - Modificações

- Configurações padrão.
- Definiu-se uma configuração padrão todos os containers criados, sendo permitida a modificação em modo administrador.

Função	Valor
<i>Persistent</i>	True
<i>Autostart</i>	True
<i>Memory</i>	64
<i>CPU Allocation</i>	100
<i>Hard Limit</i>	True

LXDUI - Desempenho

- Comparou-se o desempenho do LXDUI com o Virtualbox na seguinte máquina:
 - Processador Intel Core i5 7200 @ 2,50GHZ * 4.
 - 8GB memória ram.
 - SSD Samsung 850 256GB.
 - Placa Gráfica Intel HD Graphics 620 (Kaby Lake GT2).
 - Sistema Operacional Ubuntu 18.04.1 LTS.

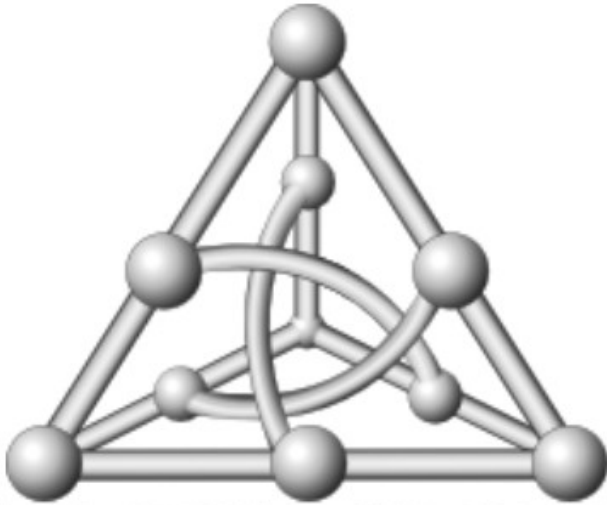
LXDUI - Desempenho

- LXDUI
 - Criação de 100 máquinas Ubuntu 17.10 minimal.
 - Tempo: 4 minutos e 54 segundos em média.
 - Consumo de memória: 337MB (2,8MB por nó) + 57MB do LXDUI.
 - Espaço em disco 400KB.
 - Situação: todo o sistema se mostrou estável.

LXDUI - Desempenho

- Virtualbox
 - Criação de 10 máquinas Ubuntu 17.10 minimal.
 - Tempo: 14 minutos e 58 segundos em média.
 - Espaço em disco: 2,6GB.
 - Situação: Houve congelamento do equipamento após a execução da décima máquina.

LXDUI - Login



FACOM-UFMS

Username

Password

LXDUI – Containers (User)



Logout

Containers

New Instance

Search:



Refresh List

JSON

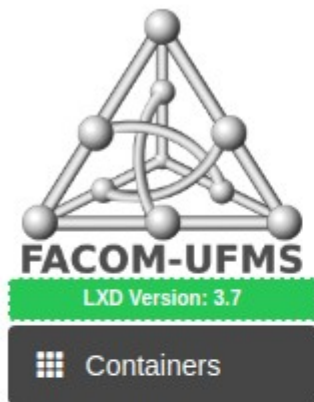
<input type="checkbox"/>	Name	Status	IP Address	Type	OS image	Actions
<input type="checkbox"/>	laughing-newton	Running	10.215.28.195	Persistent	alpine 3.4 amd64	
<input type="checkbox"/>	peaceful-northcutt-1	Running	10.215.28.155	Persistent	ubuntu artful amd64	
<input type="checkbox"/>	peaceful-northcutt-2	Running	10.215.28.158	Persistent	ubuntu artful amd64	

Showing 1 to 3 of 3 entries

List 10

Previous 1 Next

LXDUI – Containers (Admin)



Containers

Images

Profiles

Network

Storage Pools

Logout

Containers

▶ Start

■ Stop

↺ Restart

✖ Delete

⏸ Freeze

⏮ Unfreeze

➕ New Instance

Search:



Refresh List

JSON

<input type="checkbox"/>	Name	Status	IP Address	Type	OS image	Actions
<input type="checkbox"/>	laughing-newton	Running	10.215.28.195	Persistent	alpine 3.4 amd64	⋮
<input type="checkbox"/>	peaceful-northcutt-1	Running	10.215.28.155	Persistent	ubuntu artful amd64	⋮
<input type="checkbox"/>	peaceful-northcutt-2	Running	10.215.28.158	Persistent	ubuntu artful amd64	⋮

Showing 1 to 3 of 3 entries

List 10 ▼

Previous 1 Next

LXDUI - Containers



- Containers
- Images
- Profiles
- Network
- Storage Pools
- Logout

Container Details : **laughing-newton** Running

Stop Restart Freeze | Delete | Snapshot Clone Move Export Terminal

Details

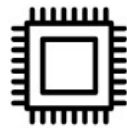
Networking

Snapshots

Advanced

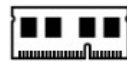
File Manager

JSON



CPU

10.3(sec)



Memory

current: 3.90MB

peak: 6.82MB



Disk

root: 308.00kB



Network caja

eth0: B 30.90kB | 2.86kB

P 257 | 21

Name : laughing-newton ✎

PID: 2800

Type : Persistent

Currently Running Processes : 9

Architecture : x86_64

OS Image : alpine 3.4 (amd64)

CPU Allocation : 100ms/100ms

Created On: 2018-12-10T21:56:38.003667832-03:00

Memory Allocation : 64MB

Is set to Autostart : Yes No

Profiles : default ✕ +

LXDUI - Imagens



Containers

Images

Profiles

Network

Storage Pools

Logout

Images

Local 2

Nightly 363

Remote 125

Launch Container(s)

Delete

Search:

JSON

<input type="checkbox"/>	OS	Description	Alias	Ver	Arch	Size
<input type="checkbox"/>	alpine	Alpine 3.4 amd64 (20180625_17:50)	N/A	3.4	amd64	2.00 MB
<input type="checkbox"/>	ubuntu	Ubuntu artful amd64 (20181002_07:42)	N/A	artful	amd64	118.19 MB

Showing 1 to 2 of 2 entries

List 10

Previous 1 Next


```
root@peaceful-northcutt-1:~# mkdir teste1 teste2 teste3
root@peaceful-northcutt-1:~# ls -l
total 2
drwxr-xr-x 2 root root 2 Dec 12 01:07 teste1
drwxr-xr-x 2 root root 2 Dec 12 01:07 teste2
drwxr-xr-x 2 root root 2 Dec 12 01:07 teste3
root@peaceful-northcutt-1:~# apt update
Hit:1 http://security.ubuntu.com/ubuntu artful-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu artful InRelease
Hit:3 http://archive.ubuntu.com/ubuntu artful-updates InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
root@peaceful-northcutt-1:~# apt install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  spell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 222 kB of archives.
After this operation, 766 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu artful/main amd64 nano amd64 2.8.6-3 [222 kB]
Fetched 222 kB in 2s (96.8 kB/s)
Selecting previously unselected package nano.
(Reading database ... 14167 files and directories currently installed.)
Preparing to unpack .../nano_2.8.6-3_amd64.deb ...
Unpacking nano (2.8.6-3) ...
Setting up nano (2.8.6-3) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
root@peaceful-northcutt-1:~# ping 10.215.28.158 -c 5
PING 10.215.28.158 (10.215.28.158) 56(84) bytes of data.
64 bytes from 10.215.28.158: icmp_seq=1 ttl=64 time=0.126 ms
64 bytes from 10.215.28.158: icmp_seq=2 ttl=64 time=0.115 ms
64 bytes from 10.215.28.158: icmp_seq=3 ttl=64 time=0.117 ms
64 bytes from 10.215.28.158: icmp_seq=4 ttl=64 time=0.129 ms
64 bytes from 10.215.28.158: icmp_seq=5 ttl=64 time=0.118 ms

--- 10.215.28.158 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4085ms
rtt min/avg/max/mdev = 0.115/0.121/0.129/0.005 ms
root@peaceful-northcutt-1:~#
```


Conclusão

- O LXD é um sistema altamente escalável pois elimina a camada de virtualização e, por consequência, a troca de informações entre o sistema hospedeiro e o sistema hospedado.
- Entretanto sistemas auxiliares devem ser desenvolvidos, assim como o LXDUI, para que o gerenciamento total seja possível.
- Por outro lado, alternativas como Docker e Virtualbox, podem ser utilizados com menos implementações.
- Porém o custo extra com recurso no hospedeiro deve ser levado em consideração.

Trabalhos Futuros

- Autenticação por serviço LDAP.
 - Atualmente não há uma base centralizada de usuários.
- Implementar controle e isolamento de usuários.
 - Usuários não visualizarem containers que não são próprios.