UNIVERSIDADE FEDERAL DE RORAIMA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
SISTEMAS OPERACIONAIS - DCC403

ALUNOS:
PHILIP MAHAMA AKPANYI
FRANCIS NUTEFE TSIGBEY

PROFESSOR: HERBERT OLIVEIRA ROCHA

VIRTUALIZAÇÃO DE SISTEMAS OPERACIONAIS (VB E CONTAINERS)

Introdução

A virtualização de sistemas operacionais é uma tecnologia cuja principal proposta é compartilhar os recursos do hardware de forma que ele execute em vários sistemas operacionais (iguais ou diferentes) e suas aplicações de forma simultânea e totalmente isoladas entre si. Com essa tecnologia, podemos fazer um melhor aproveitamento dos recursos computacionais novos ou existentes

.

Neste trabalho, usaremos uma máquina virtual e containers para mostrar a virtualização sistemas operacionais. Uma máquina virtual é um espaço virtual isolado com acesso ao hardware, onde funciona um sistema virtual. Em ambientes virtualizados, as máquinas virtuais simulam uma réplica física de uma máquina real. Os usuários têm a ilusão de que o sistema está disponível para seu uso exclusivo. O monitor de máquinas virtuais (*Virtual Machine Monitor - VMM*) é uma aplicação que implementa uma camada de virtualização, a qual permite que múltiplos sistemas operacionais funcionam sobre um mesmo *hardware* simultaneamente.

Fornecer um ambiente isolado dentro do sistema operacional de hospedagem é comumente conhecido como virtualização no nível do sistema operacional e esse ambiente isolado pode ser definido como *container*: Um container é um ambiente de execução autônomo que compartilha o kernel do sistema host e que é (opcionalmente) isolados de outros recipientes no sistema.

Instalação de software para criar containers (docker)

Para instalar o docker engine é simples. Acesse o terminal preferido do GNU/Linux e torne-se usuário root:

su - root e execute o comando abaixo:

wget -qO- https://get.docker.com/ | sh

Para criar o container temos que criar uma imagem primeira.

```
docker image build -t meuubuntu:nginx_auto

docker container run -it --rm meuubuntu:nginx dpkg -l nginx
```

Instalação e Criação de Máquina Virtual (VirtualBox)

Para mostrar a virtualização em máquinas virtuais, utilizamos o *Virtualbox* [1] como a ferramenta para instalar nossos sistemas operacionais.

No terminal de **Ubuntu**, siga as instruções abaixo para instalar o Virtualbox.

1. Atualize seu sistema antes de instalar

```
sudo apt-get update
sudo apt-get upgrade
```

2. Configuração do repositório de Apt

```
wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- | sudo apt-key add -

wget -q https://www.virtualbox.org/download/oracle_vbox.asc -O- | sudo apt-key add -
```

3. Acrescente Oracle VirtualBox PPA ao sistema de Ubuntu

sudo add-apt-repository "deb http://download.virtualbox.org/virtualbox/debian xenial contrib"

4. Finalmente, instalar o VirtualBox utilizando os seguintes comandos

```
sudo apt-get update
sudo apt-get install virtualbox-5.2
```

5. Para abrir, digite

```
virtualbox
```

Para instalar no **Windows**, é mais simples.

1. Baixe e execute o software (o link está na referência)
2. Aceite todas as condições. É normal que a rede se desconecte no processo da instalação.

Criação de Cluster com Máquinas Virtuais

Um cluster de computador pode ser um sistema simples de dois nós que apenas conecta dois computadores pessoais, ou pode ser um supercomputador muito rápido. A abordagem de agrupamento de computadores geralmente (mas nem sempre) conecta vários nós de computação prontamente disponíveis (por exemplo, computadores pessoais usados como servidores) por meio de uma rede local rápida. O sistema operacional, Cent OS (versão mínima), foi escolhido para criar o cluster de máquinas virtuais. Utilizamos esse sistema pela simplicidade e facilidade de uso.

Para criar cluster de máquinas [2], utilizamos duas máquinas virtuais (CentOS-1 e CentOS-2)

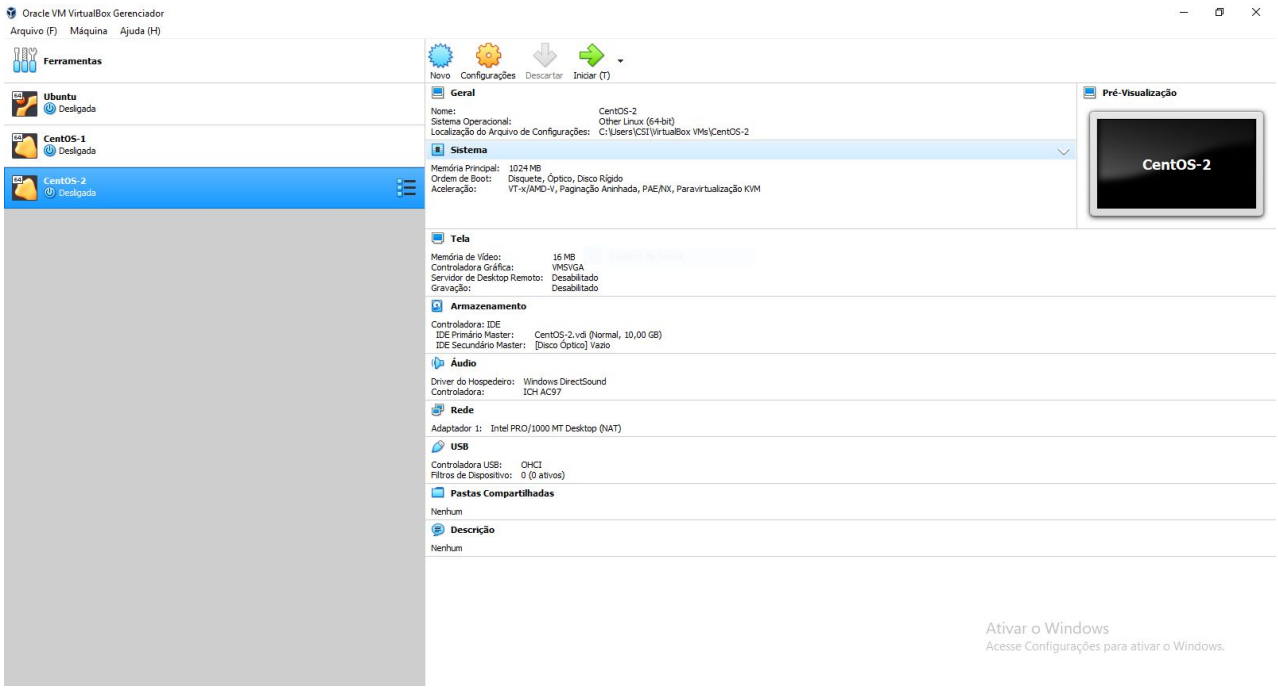

Figura 1 - Máquinas Virtuais

Precisamos criar e acessar a internet com DHCP - *Dynamic Host Configuration Protocol* que é um protocolo utilizado em redes de computadores que permite a estes obterem um endereço IP automaticamente. Para isso, abrimos o *command prompt* do Windows dentro do diretório **C:\Program Files\Oracle\VirtualBox**

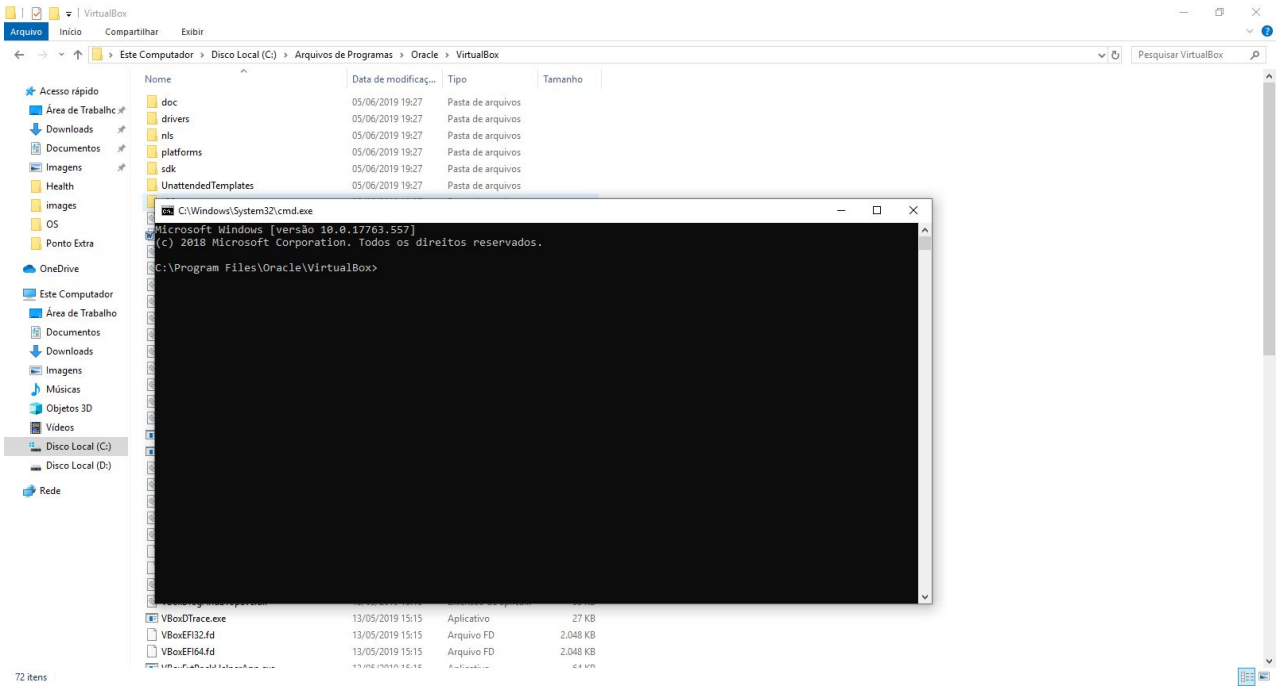

Figura 2 - Windows cmd

Vamos criar uma rede interna para fazer a função de DHCP com o seguinte comando

```
VBoxManage dhcpserver add --netname intnet --ip 10.0.1.1 --netmask 255.255.255.0 --lowerip 10.0.1.2 --upperip 10.0.1.200 --enable
```

Com o DHCP criado, voltamos para a configuração das nossas máquinas virtuais para fazer algumas alterações na rede

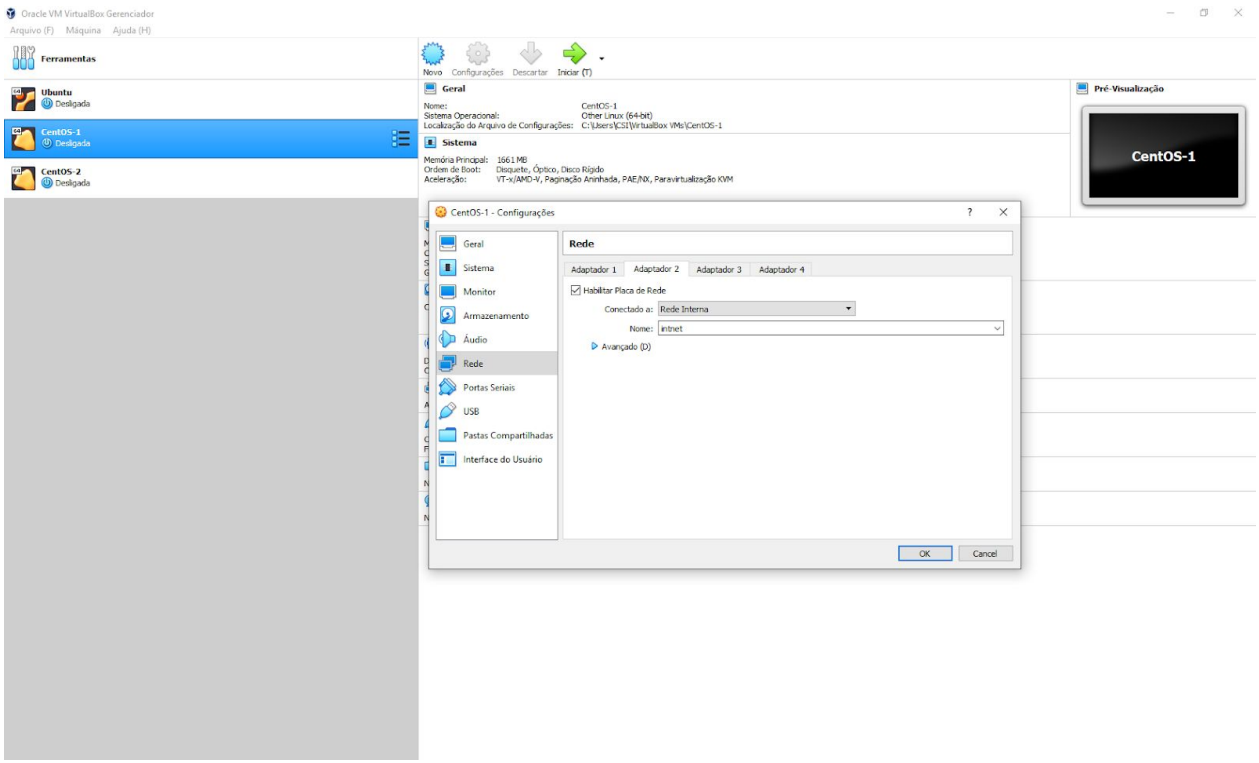

Figura 3 - Configuração das máquinas virtuais

Nas duas máquinas, configure a conexão de internet com os seguintes comandos

```
nmcli d
```

enp0s8 vai conectar mas enp0s3 vai desconectar. Isso é normal

Vamos abrir a configuração e modificar a última linha

```
vi /etc/sysconfig/network-scripts/ifcfg-enp0s3
```

Mude a última linha nas duas máquinas ONBOOT=yes
Reinicie a rede

```
systemctl restart network
```

Tente ping 8.8.8.8 (Google). Se conseguir, significa que temos acesso à internet. Agora, precisamos estabelecer um meio de comunicação de uma máquina (CentOS-1) para a outra (CentOS-2)

```
ip addr show
```

Verificando os endereços IPs na rede interna, vemos que cada máquina tem seu próprio endereço
CentOS-1 - 10.0.1.2
CentOS-2 - 10.0.1.3

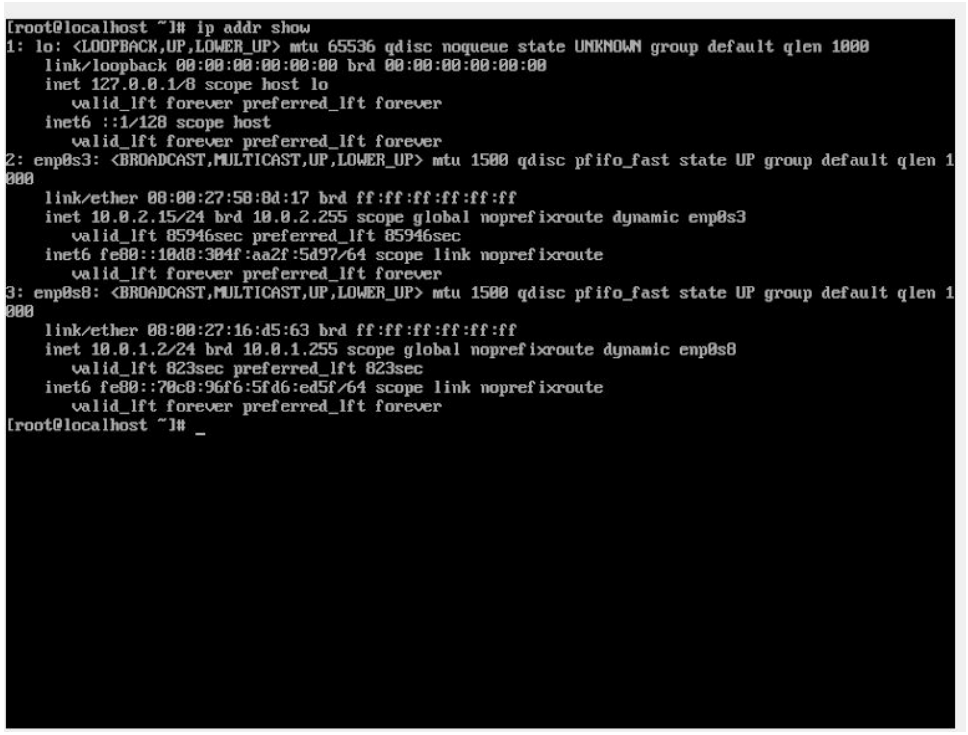

Figura 4 - Mostrando o endereço de IP da máquina CentOS-1 no enp0s8
Agora, configurando a máquina para poder ter comunicação entre as máquinas, vamos usar NFS Server (Network File System). Para isso, precisamos instalar algumas bibliotecas

```
yum install nfs-utils nfs-utils-lib -y
```

```
systemctl start rpcbind nfs-server
systemctl enable rpcbind nfs-server
```

Vamos criar uma pasta vazia que será usada como pasta compartilhada

```
mkdir /nfs
```

```
vi /etc/exports
```

Adicionamos a seguinte linha ao /etc/exports. Escrevemos o nome da pasta compartilhada e os endereços IP que queremos que a pasta compartilhada seja compartilhada.

Agora, nós carregamos as novas alterações em /etc/exports.

```
exportfs -a
```

Nós temos que mudar o firewall para permitir NFS e complementar os serviços.

```
firewall-cmd --permanent --zone=public --add-service=nfs
firewall-cmd --permanent --zone=public --add-service=mountd
firewall-cmd --permanent --zone=public --add-service=rpc-bind
```

```
firewall-cmd --reload
```

Vamos reiniciar
```
systemctl restart nfs
```

Na máquina #2, Criamos uma pasta onde a pasta compartilhada da máquina 1 será montada na máquina 2.

```
mkdir -p /nfs
```
Garantimos que podemos acessar o CentOS-1, o servidor NFS. Certifique-se de que os dois comandos a seguir não retornem erros.

```
showmount -e 10.0.1.2
```

```
rpcinfo -p 10.0.1.2
```

```
mount 10.0.1.2:/nfs /nfs
```

Com df -h, devemos ver que a montagem 10.0.1.2:/nfs foi criada na parte inferior. Se criarmos qualquer arquivo dentro de /nfs, então todas as máquinas conectadas podem ver o mesmo arquivo

```
df -h
```

Agora, testamos que a pasta compartilhada realmente funciona.

```
cd /nfs
```

```
touch arquivo_teste.txt
```

No CentOS-1, se nós navegamos para cd /nfs, veremos que o arquivo_teste.txt está dentro da pasta.

Quando você reinicia as duas máquinas virtuais, a pasta compartilhada NFS não estará lá. Precisamos definir uma maneira mais automática para o cliente NFS procurar a pasta NFS.

No cliente, nós mudamos um arquivo chamado /etc/fstab

```
vi /etc/fstab
```

Sempre que reiniciamos o cliente, podemos montar novamente a pasta compartilhada do NFS digitando mount -a.

```
mount -a
```

Feito tudo isso, nós conectamos 2 máquinas virtuais no VirtualBox e criamos um servidor e cliente NFS, que é necessário para pastas compartilhadas.

Vantagens e Desvantagens entre máquinas virtuais e containers

As Máquinas virtuais criam uma nova instância de um sistema operacional para cada execução de máquina virtual. Isso oferece vários benefícios, como a capacidade de executar um sistema completamente diferente do convidado, em comparação com o host mas também vem com muitos inconvenientes.

Em segundo lugar, as máquinas virtuais ocupam muito mais espaço no disco e são mais difíceis de manter. Os *containers* exigem apenas o aplicativo e suas dependências, enquanto o kernel é compartilhado entre eles. Como o sistema operacional já está em execução, iniciar um *container* tende a ser muito mais rápido do que iniciar uma máquina virtual. O kernel compartilhado pode nem sempre ser um benefício, pois, por exemplo, a execução de aplicativos do Windows em *containers* no Linux não é possível.

Referências

[1] - Baixar o VirtualBox https://www.virtualbox.org/wiki/Downloads
[2] - Como criar máquinas virtuais e habilitar comunicação entre elas
https://www.slothparadise.com/how-to-connect-virtual-machines-and-setup-nfs-server-part-1/