

\*\*\*\*\*Etapas para criação de um cluster\*\*\*\*\*

## 1 - Escolher e Instalar o SO (neste caso Linux)

Mestre - Lubuntu

no1 - Lubuntu sem Interface Gráfica

no2 - Lubuntu sem Interface Gráfica

\*Definir todas as máquinas com o mesmo nome de usuário:

cluster@mestre

cluster@no1

cluster@no2

## 2 - Configurar Rede

### 2.1 Adicionar placa de rede (Local)

*\*Virtual Box\**

Arquivo>>Preferências >> Rede >> Criar Nova >> Editar Rede>>Renomear Rede -  
"RedeCluster">>Alterar CIDR da Rede para "192.168.1.0/24"

*\*Cada máquina/VM\**

Configurações >> Rede >> Adaptador 2 >> Habilitar Placa de Rede >> Conectado a: Rede NAT >>  
Nome: RedeCluster

### 1) Iniciar as máquinas

### 2) Verificar a placa de rede

todas as máquinas : ~\$ ifconfig

*\*\*realizei a consulta apenas no "mestre"  
placa identificada - "enp0s8"*

### 2.2 Instalar ferramentas adicionais (todas as máquinas/VMs)

#### 1) Instalar net-tools

todas as máquinas : ~\$ sudo apt-get install net-tools

#### 2) Instalar ifupdown (interface de rede on/off)

todas as máquinas : ~\$ sudo apt-get install ifupdown

### 2.3 Configurar placa de Rede

- usando o super usuário e o editor nano, editar o arquivo "interfaces" em /etc/network/interfaces

\$ sudo nano /etc/network/interfaces

\*\*\*\*GNU nano 4.8\*\*\*\*\*/etc/network/interfaces\*\*\*\*

# interfaces(5) file used by ifup(8) and ifdown(8)

# include files from etc/network/interfaces.d

source-directory /etc/network/interfaces.d

```
auto enp0s8
iface enp0s8 inet static
address 192.168.1.110 (Verificar em cada nó)
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
```

Ctrl+O para salvar e Ctrl+X para sair

\*\*\*\*\*

## 2.4 Iniciar a Rede configurada

### 1) Start placa de rede

todas as máquinas : ~\$ sudo ifup enp0s8

### 2) teste de start

todas as máquinas : ~\$ ifconfig

*\*verificar se o **inet** tem o mesmo valor de **address** do **/interfaces***

## 2.5 Teste de Ping (máquina principal)

cluster@mestre: ~\$ ping "ip" (sintaxe)

cluster@mestre: ~\$ ping 192.168.1.120 (nó1)

cluster@mestre: ~\$ ping 192.168.1.130 (nó2)

## 2.6 Colocar apelido nos ip`s das máquinas

usando superusuario editar e adicionar os ip`s e nomealos em /etc/hosts

### 1) mestre

cluster@mestre: ~\$ sudo nano /etc/hosts

\*\*\*\*GNU nano 4.8\*\*\*\*\*/etc/hosts\*\*\*\*\*

# Host addresses

127.0.0.1 localhost

127.0.1.1 mestre

192.168.1.110 mestre

192.168.1.120 no1

192.168.1.130 no2

# IPv6

: :1 localhost ip6-localhost ip6-loopback

ff02::1 ip6-allnodes

ff02::2 ip6-allrouters

Ctrl+O para salvar e Ctrl+X para sair

\*\*\*\*\*

### 1) no1

\$ sudo nano /etc/hosts

```
****GNU nano 4.8*****/etc/hosts*****
```

# Host addresses

127.0.0.1 localhost

127.0.1.1 no1

192.168.1.110 mestre

192.168.1.130 no2

# IPv6

::1 localhost ip6-localhost ip6-loopback

ff02::1 ip6-allnodes

ff02::2 ip6-allrouters

Ctrl+O para salvar e Ctrl+X para sair

```
*****
```

### 1) no2

\$ sudo nano /etc/hosts

```
****GNU nano 4.8*****/etc/hosts*****
```

# Host addresses

127.0.0.1 localhost

127.0.1.1 no2

192.168.1.110 mestre

192.168.1.120 no1

# IPv6

::1 localhost ip6-localhost ip6-loopback

ff02::1 ip6-allnodes

ff02::2 ip6-allrouters

Ctrl+O para salvar e Ctrl+X para sair

```
*****
```

## 2.6 Teste de Ping com nomes

### 1) mestre (máquina principal)

cluster@mestre: ~\$ ping no1

cluster@mestre: ~\$ ping no2

### 2) nó 1

cluster@no1: ~\$ ping mestre

cluster@no1: ~\$ ping no2

### 3) nó 2

cluster@no2: ~\$ ping mestre

cluster@no2: ~\$ ping no1

## 3 - Acesso Remoto (SSH)

### SSH (Secure Shell)

SSH é uma ferramenta de acesso remoto, que permite acessar uma máquina da rede e controlá-la remotamente.

### 3.1 Instalação

#### 1) Buscar/download repositório e instalação

p/ todas as máquinas da rede

todas as máquinas: ~\$ sudo apt-get install openssh-server

#### 2) Ativar serviço SSH

todas as máquinas: ~\$ sudo systemctl enable ssh

#### 3) Verificar instalação

todas as máquinas: ~\$ sudo systemctl status ssh

### 3.2 Configuração

#### 1) Gerar chave de acesso (mestre)

cluster@mestre: ~\$ ssh-keygen

*\*\*chave salva em /home/mestre/.ssh/id\_rsa*

#### 2) Realizar cópia da chave de acesso para os nós (mestre)

Entrar no diretório oculto .ssh onde a chave foi salva:

cluster@mestre: ~\$ cd .ssh

criar directorio .ssh nos nós:

cluster@no1: mkdir .ssh

cluster@no2: mkdir.ssh

fazer a cópia de id\_rsa.pub para os nós:

cluster@mestre: ~/.ssh\$ ssh-copy-id -i ~/.ssh/id\_rsa.pub no1

... continue connecting (yes/no/[fingerprint])? yes

no1@no1's password: \*\*\*\*

```
cluster@mestre: ~/.ssh$ ssh-copy-id -i ~/.ssh/id_rsa.pub no2
... continue connecting (yes/no/[fingerprint])? yes
no2@no2's password: ****
```

*\*\*no diretório .ssh dos nós, será gerado um arquivo chamado **authorized\_keys***

### 3.3 Teste de acesso (**mestre**)

```
cluster@mestre: ~$ ssh no1
cluster@no1: ~$ exit
```

## 4 - Configurar Compartilhamento (MPI):

### MPI (Machine)

(descrever o que é a biblioteca MPI)

#### 4.1 Instalação

todas as máquinas : ~\$ sudo apt-get install libopenmpi-dev

#### 4.2 Configuração

##### 1) Instalar ferramenta NFS (Network File System) - server:

cluster@mestre: ~\$ sudo apt-get install nfs-kernel-server

##### 2) Instalar arquivos padrões NFS-common:

todas as máquinas : ~\$ sudo apt-get install nfs-common

##### 3) Instalar mapeador de portas (PortMap):

todas as máquinas : ~\$ sudo apt-get install portmap

##### 4) Criar diretório a ser compartilhado:

todas as máquinas : ~\$ mkdir clusterdir

##### 5) Editar **exports** em **/etc/exports**

cluster@mestre: ~\$ sudo nano /etc/exports

```
****GNU nano 4.8*****/etc/exports*****
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes          hostname1(rw,sync,no_subtree_check)
hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4            gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes     gss/krb5i(rw,sync,no_subtree_check)
#
```

```
/home/cluster/clusterdir 192.168.1.0/24(rw,no_subtree_check,async,no_root_squash)
```

Ctrl+O para salvar e Ctrl+X para sair

```
*****
```

*\*rw - leitura e escrita*

*\*no\_subtree\_check - não criar subpastas*

*\*async - assíncrono, pode ser acessado por varias máquinas ao mesmo tempo*

*\*no\_root\_squash - não necessita de root para acessar*

## 6) Iniciar serviço de NFS:

```
cluster@mestre: ~$ systemctl enable nfs-kernel-server
```

## 7) Reiniciar e verificar status do serviços de NFS:

```
cluster@mestre: ~$ sudo /etc/init.d/nfs-kernel-server restart
```

```
cluster@mestre: ~$ systemctl status nfs-kernel-server
```

## 8) Teste de Pasta Compartilhada:

```
cluster@no`s: ~$ showmount -e 192.168.1.110
```

*\*\*192.168.1.110 - ip servidor*

## 9) Copiar e montar directorio da rede:

```
cluster@mestre: ~$ sudo mount -t nfs 192.168.1.110:/home/cluster/clusterdir
```

/home/cluster/clusterdir **\*\*\*verificar se necessita desse**

```
cluster@no1: ~$ sudo mount -t nfs 192.168.1.110:/home/cluster/clusterdir
```

/home/cluster/clusterdir

```
cluster@no2: ~$ sudo mount -t nfs 192.168.1.110:/home/cluster/clusterdir
```

/home/cluster/clusterdir

## 10) Editar fstab:

```
no`s@no`s: ~$ sudo nano /etc/fstab
```

```
****GNU nano 4.8****/etc/fstab*****
```

```
# /etc/fstab: static file system information.
```

```
#
```

```
# Use 'blkid' to print the universally unique identifier for a device; this may
```

```
# be used with UUID= as a more robust way to name devices that works even if
```

```
# disks are added and removed. See fstab(5).
```

```
#
```

```
# <file system>          <mount point> <type>  <options>  <dump>  <pass>
```

```
UUID=4a1ac170-1a4b-420c-93f2-11396fb3fb63 /                ext4      defaults  0 1
```

```
192.168.1.110:/home/cluster/clusterdir /home/cluster/clusterdir nfs rw,sync,hard,int 0 0
```

Ctrl+O para salvar e Ctrl+X para sair

```
*****
```

### 11) Gerar arquivo de teste:

```
cluster@mestre: ~$ cd clusterdir
cluster@mestre: ~$ sudo nano teste.txt
```

```
****GNU nano 4.8*****/etc/teste.txt*****
```

arquivo de teste

Ctrl+O para salvar e Ctrl+X para sair

```
*****
```

### 12) Acessar arquivo de teste:

```
cluster@no`s: ~$ cd clusterdir
cluster@no`s: ~/clusterdir$ ls
```

*\*\*verificar se o arquivo teste.txt está presente na pasta*

### 13) Ferramentas Adicionais

```
cluster@master: ~$ sudo apt-get install build-essential
```

*\*\*build-essential - compilador*

### 14) Configurar hostfile (MPI)

```
cluster@master: ~$ sudo nano .mpi_hostfile
```

```
****GNU nano 4.8*****.mpi_hostfile*****
```

```
localhost slots=2                #quantidade de núcleos a processar no mestre
```

```
no1 slots=2                      #quantidade de núcleos a processar no no1
```

```
no2 slots=1                      #quantidade de núcleos a processar no no2
```

Ctrl+O para salvar e Ctrl+X para sair

```
*****
```

## 5 - Rodar Cluster (MPI)

### 1) Compilar

```
cluster@master: : ~/clusterdir$ mpic++ nome_do_arquivo.c++ -o nome_arquivo_compilado
```

### 2) Rodar Localmente

```
cluster@master: : ~/clusterdir$ ./nome_arquivo_compilado
```

### 3) Rodar em rede

```
cluster@master: : ~/clusterdir$ mpirun -np 5 --hostfile ./mpi_hostfile ./nome_arquivo_compilado
```