

Guia de configuração de Thin Clients Usando LTSP

Este guia prático mostrará aos alunos do curso de Ciência da Computação, matéria de Sistemas Distribuídos como configurar um servidor LTSP.

Configurando o Servidor:

Para realizar a configuração de um servidor LTSP é necessário baixar uma imagem Linux, recomendamos a distribuição Ubuntu Server 22.04.5 LTS (Jammy Jellyfish), onde ela pode ser baixada através deste link: <https://releases.ubuntu.com/22.04/ubuntu-22.04.5-live-server-amd64.iso> com esse link é feito o download direto da distribuição, que é encontrada no site oficial do Ubuntu:

https://releases.ubuntu.com/22.04/?_gl=1*1t8ramz*_gcl_au*MTUzODAzNDkwMS4xNzI2NTIwODc3&_ga=2.38122346.813318146.1728309647-1113315686.1726520874

Depois do download da ISO, você pode utilizar o virtual box para emular este servidor, mas você pode fazer a instalação normal, sem usar o virtual box. Esse tutorial será desenvolvido no virtual box.

De qualquer forma você irá precisar de duas placas de rede, sendo uma delas ****NIC 1 (NAT ou DHCP externo)**** que conecta o servidor à internet para atualizações e administração remota e outra placa de rede ****NIC 2 (Rede Interna)****: dedicada aos thin clients, com o LTSP fornecendo IPs via DHCP.

Após configurar as placas de rede, passa-se a configurar o servidor, com o Ubuntu já executando, recomenda-se utilizar o comando:

“sudo -i”

Assim pode-se executar todos os comandos sem ficar digitando a senha e sem utilizar o sudo para qualquer comando

É necessário garantir que todas as atualizações do Ubuntu estejam em dia, para isso o comando abaixo deve ser executado:

“apt update && sudo apt upgrade -y”

Após todas as atualizações estarem em dia, podemos começar a configurar o servidor LTSP

“apt install --install-recommends ltsp dnsmasq ipxe nfs-kernel-server openssh-server squashfs-tools ethtool net-tools epoptes curl”

Lembre-se de adicionar o seu usuário ao grupo “epoptes”

“gpsswd -a administrator epoptes” substitua administrador pelo seu nome de usuário.

Após todas as instalações, devemos configurar os ip's dual NIC, para isso deve-se editar o arquivo que está no diretório /etc/netplan/.yaml, utiliza-se o comando

“nano /etc/netplan/00-installer-config.yaml”

Dentro do arquivo, estaremos utilizando as seguintes interfaces, levando em consideração que a placa de rede nat tem o nome **enp0s3** e a placa de rede interna tem o nome de **enp0s8**. Caso as suas placas de rede não tenham esse nome, utilize os respectivos nomes para essas interfaces. Configure o arquivo desse jeito:

“

network:

version: 2

renderer: networkd

```
ethernets:
  enp0s3:
    dhcp4: yes
  enp0s8:
    dhcp4: no
    addresses:
      - 192.168.67.1/24
    nameservers:
      addresses:
        - 8.8.8.8
        - 1.1.1.1
```

“

Lembre-se de seguir corretamente as indentações de cada linha pois os arquivos .yaml são sensíveis as indentações. Ele deve ficar parecido com:

```
GNU nano 6.2 /etc/netplan/00-installer-config.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: yes
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.67.1/24
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
```

[Read 14 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location	M-U Undo
^X Exit	^R Read File	^N Replace	^U Paste	^J Justify	^_ Go To Line	M-E Redo

Aplique as mudanças com:

“netplan apply”

É importante digitar o comando para o dnsmasq:

“ltsp dnsmasq --proxy-dhcp=0”

Agora devemos configurar o dnsmasq para fornecer DHCP a rede interna, para isso execute o comando:

“nano /etc/dnsmasq.conf”

Dentro do arquivo dnsmasq.conf, no final do mesmo coloque as seguintes informações:

“

enable-tftp

tftp-root=/srv/tftp

“

Depois desses passos, vamos criar uma imagem chrootless do LTSP, usando o seguinte comando:

“ltsp image /”

Caso os arquivos vmlinuz e initrd.img não forem copiados corretamente, copie os mesmos manualmente com:

“cp /boot/vmlinuz-\$(uname -r) /srv/tftp/ltsp/vmlinuz”

“cp /boot/initrd.img-\$(uname -r) /srv/tftp/ltsp/initrd.img”

“cp /boot/vmlinuz-\$(uname -r) /srv/tftp/ltsp/x86_64/vmlinuz”

“cp /boot/initrd.img-\$(uname -r) /srv/tftp/ltsp/x86_64/initrd.img”

também

“cp /boot/vmlinuz-\$(uname -r) /srv/ltsp/images/x86_64/vmlinuz”

“cp /boot/initrd.img-\$(uname -r) /srv/ltsp/images/x86_64/initrd.img”

É de extrema importância dar as devidas permissões para os arquivos

“chmod 755 /srv/ltsp/images/x86_64/vmlinuz”

“chmod 755 /srv/ltsp/images/x86_64/initrd.img”

também

“chmod 755 /srv/tftp/ltsp/x86_64/vmlinuz”

“chmod 755 /srv/tftp/ltsp/x86_64/initrd.img”

“chmod 755 /srv/tftp/ltsp/vmlinuz”

“chmod 755 /srv/tftp/ltsp/initrd.img”

“chmod 755 /srv/tftp”

“chmod 755 /srv/ltsp”

Caso não há um diretório em **/srv/ltsp/images/x86_64** crie com:

“mkdir -p /srv/ltsp/images/x86_64”

Reinicie o ltsp

“ltsp initrd”

Após devemos configurar um menu iPXE com o comando

“ltsp ipxe”

Caso não copie os arquivos binários snponly.efi, pode copiar manualmente usando o comando

“cp /usr/lib/ipxe/snponly.efi /srv/tftp/ltsp/snponly.efi”

Agora vamos configurar o servidor de arquivos NFS para servir as imagens ou chroots pela rede com o comando:

“ltsp nfs”

Por último criamos os principais diretórios do LTSP em um único arquivo de imagem, vamos executar o seguinte comando:

“ltsp initrd”

Isso cria o arquivo **/srv/tftp/ltsp/ltsp.img**, que é transferido para os clientes durante o boot. Execute este comando novamente após atualizações de pacotes ou mudanças no **/etc/ltsp/ltsp.conf**.

É importante ajustar o arquivo ltsp.ipxe para fazer o boot correto:

“nano /srv/tftp/ltsp/ltsp.ipxe”

No fim do arquivo insira:

“

kernel /ltsp/vmlinuz

initrd /ltsp/initrd.img

boot

”

Após isso baixamos o x2go para garantir que os clientes utilizem os recursos do servidor como CPU e memória ram.

“apt install x2goserver x2goserver-xsession”

Devemos também configurar o ltsp.conf para usar as imagens certas, o kernel certo e também o x2go

“nano /etc/ltsp/ltsp.conf”

Dentro do arquivo ltsp.conf adicione as seguintes linhas:

“

[default]

KERNEL=/ltsp/vmlinuz

INITRD=/ltsp/initrd.img

XSESSION=x2go

XSERVER=auto

SSHFS=Y

KEEP_SYSTEM_SERVICES=sshd

NBD_SWAP=Y

“

É extremamente importante resetar o dnsmasq antes de ligar o cliente,

“systemctl restart dnsmasq”

Verifique o status do dnsmasq

“systemctl status dnsmasq”

Também para garantir, reinicie o ltsp

“ltsp initrd”

Com esses passos, você terá um servidor LTSP funcionando corretamente.

Configurando o cliente LTSP

Para configurar o cliente LTSP é necessário a criação de uma máquina. Tenha uma imagem Ubuntu 64 bits ou no virtual box crie uma máquina virtual com Ubuntu 64 bits.

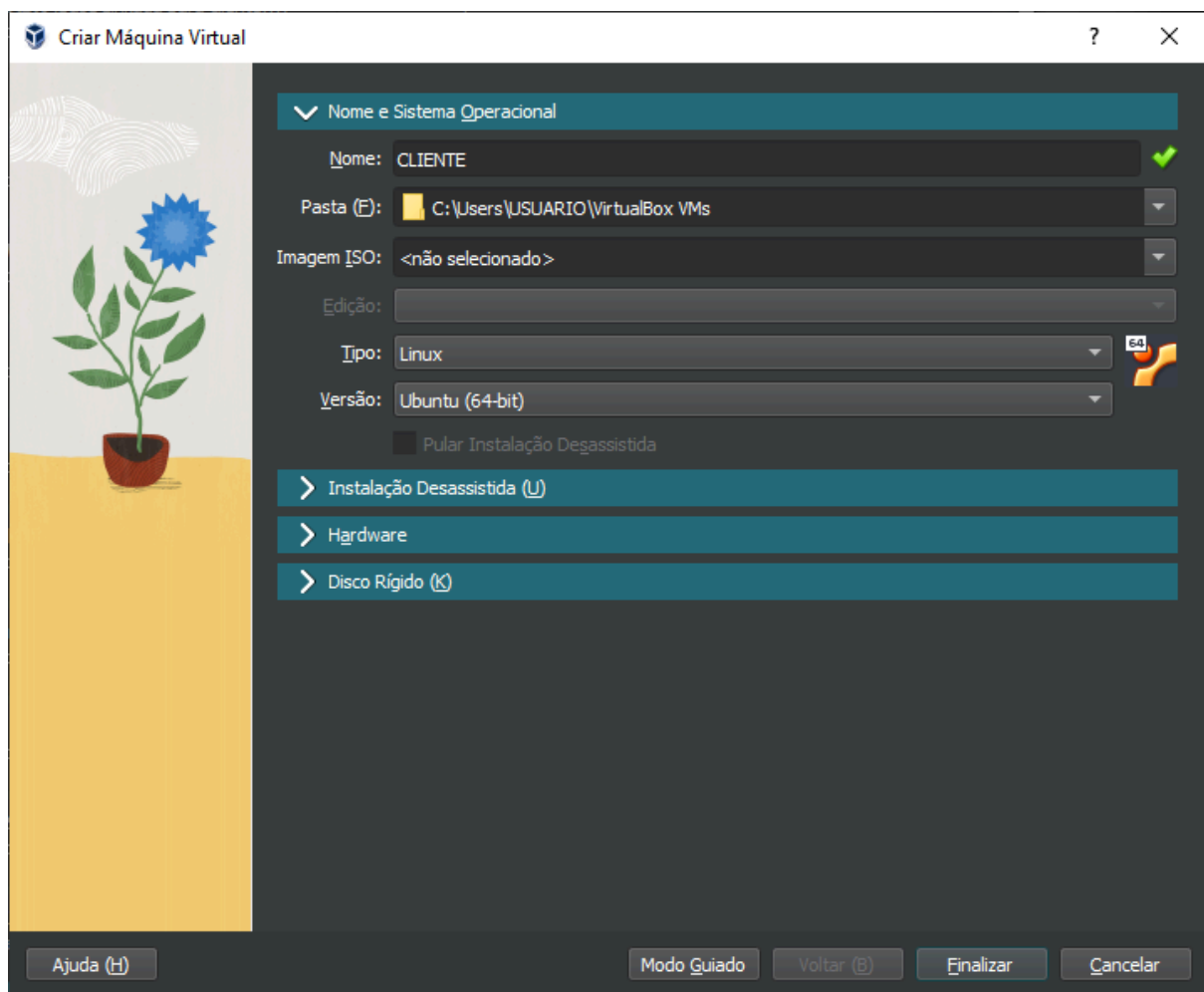
Essa máquina vai utilizar:

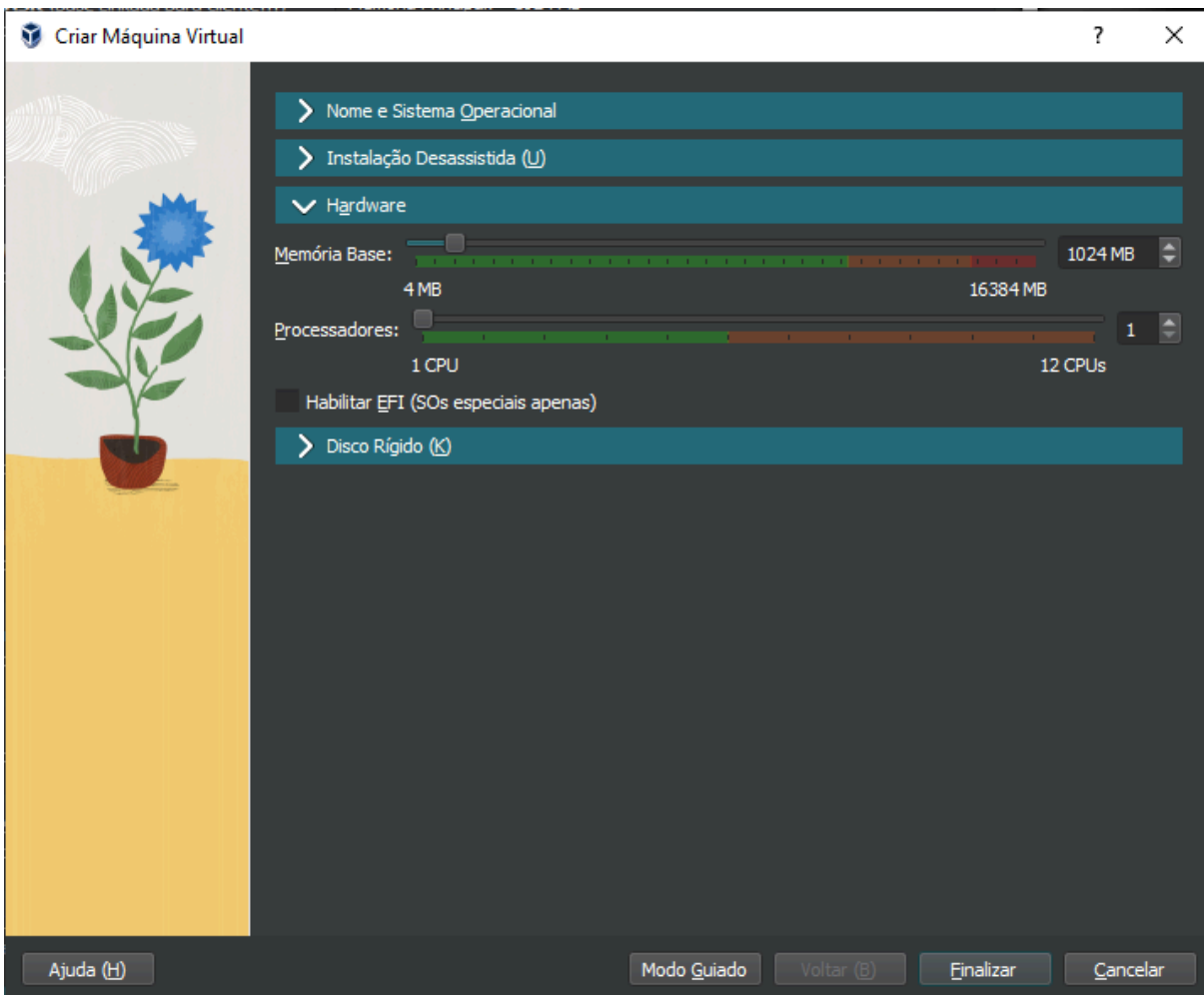
1024 mb de memória RAM;

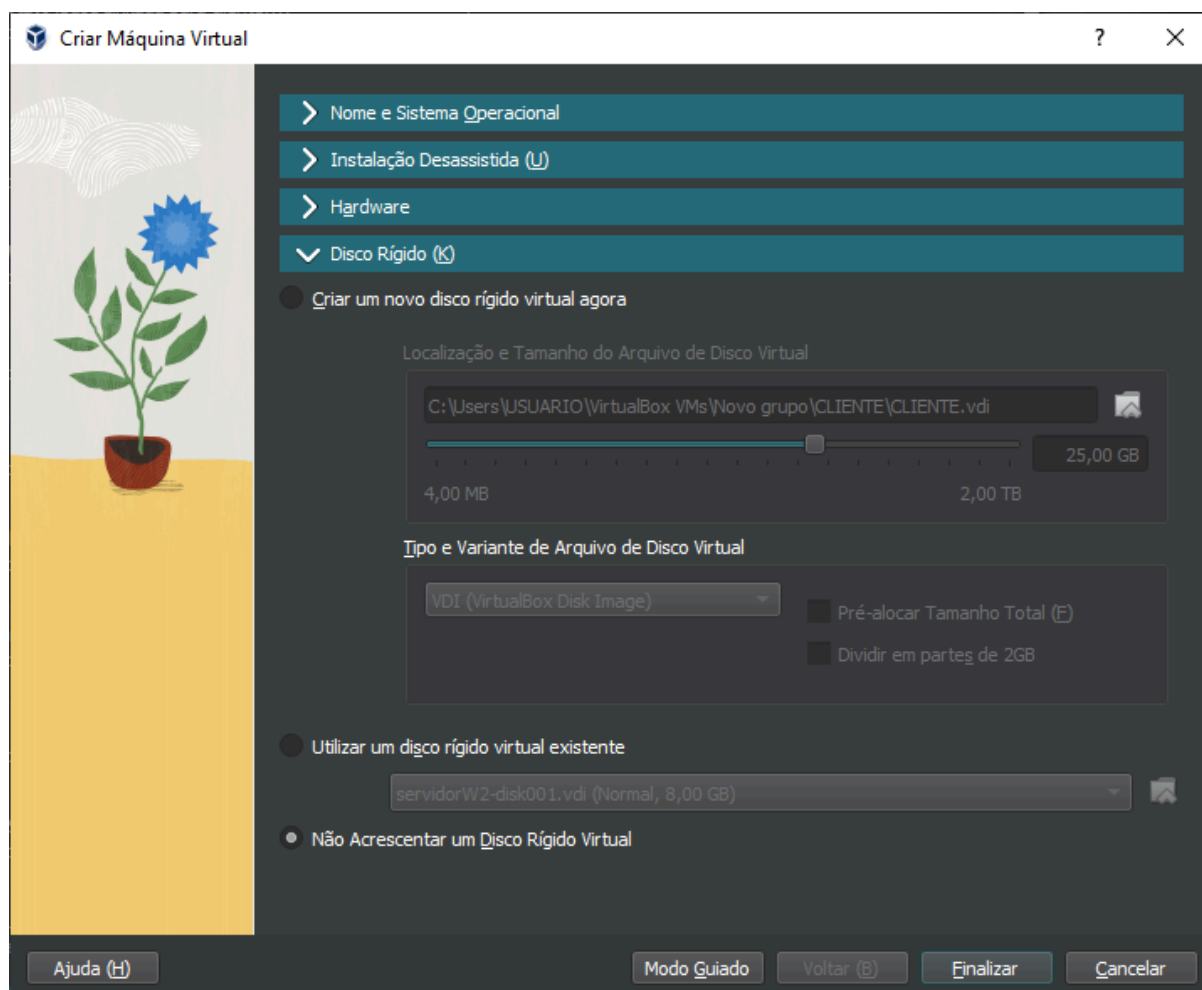
1 CPU;

Não terá disco rígido.

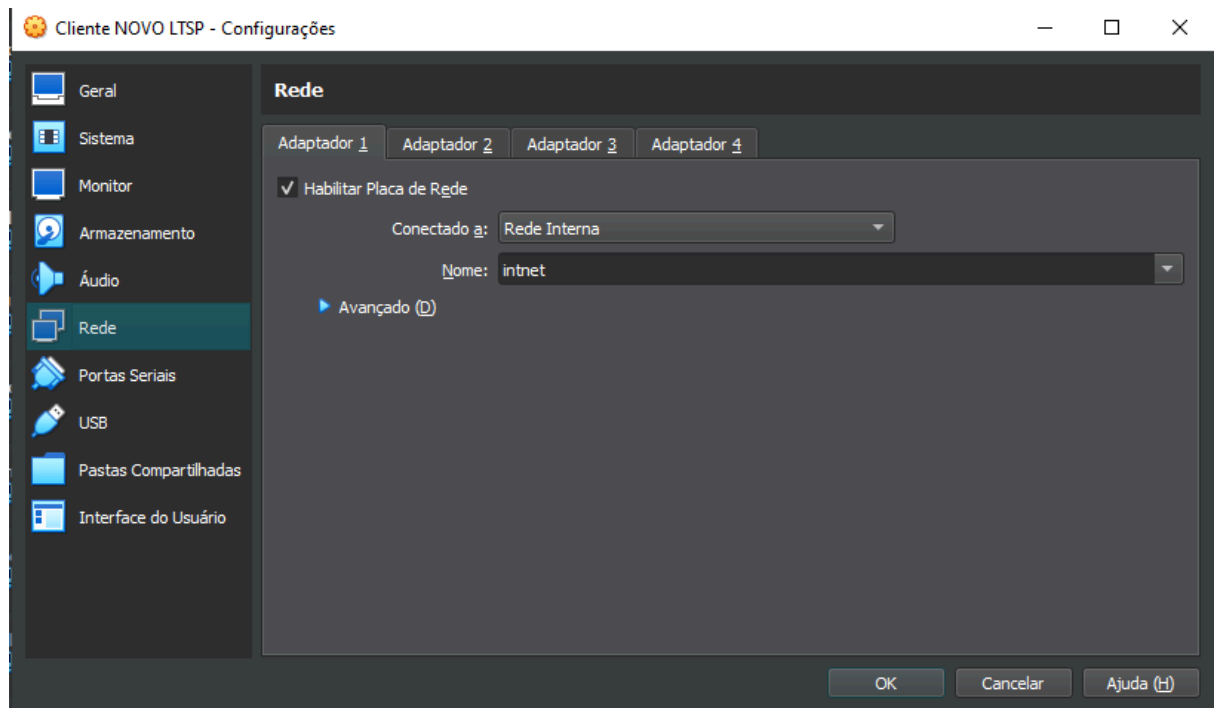
Esses são os passos no virtual box







Após isso, devemos configurar uma placa de rede interna para o cliente poder se conectar no servidor.



Esse cliente deve dar boot em rede, por tanto, a preferência deve ser sempre boot em rede, no virtualbox o procedimento é:


Na aba “Sistema” habilitar o boot em rede, desabilitar os outros e na ordem de boot, o boot em rede deve ser o primeiro.

Assim você terá um cliente e um servidor LTSP rodando, para fazer alguns testes com o sistema de arquivos ou comandos no servidor pode se usar os seguintes comandos no cliente:

```
“echo "Teste em casa" > /home/(nome do seu servidor)/teste_cliente.txt”
```

E no servidor podendo usar o cat vai aparecer algo parecido

```
“cat /home/(nome do seu servidor)/teste_cliente.txt”
```



```
root@leonardo-ltsp:~# cat /home/leonardo/teste_cliente.txt
Teste em casa
root@leonardo-ltsp:~# _
```

Outro teste para fazer no servidor é usando o seguinte comando para verificar se existe processos de usuário

“ps aux | grep (usuário)” substitua (usuário) pelo nome do client

Além de ver se o uso de memória no cliente está baixo, demonstrando que está usando do servidor

“free -h”