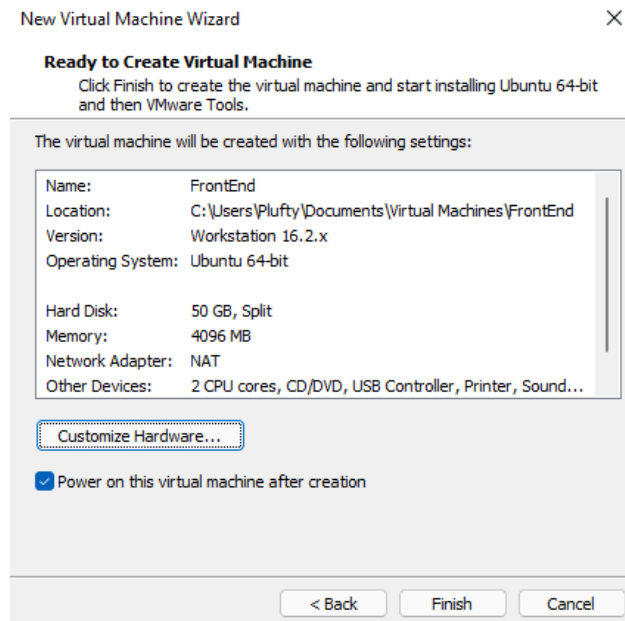


Gleidson Vinícius Gomes Barbosa - 6331

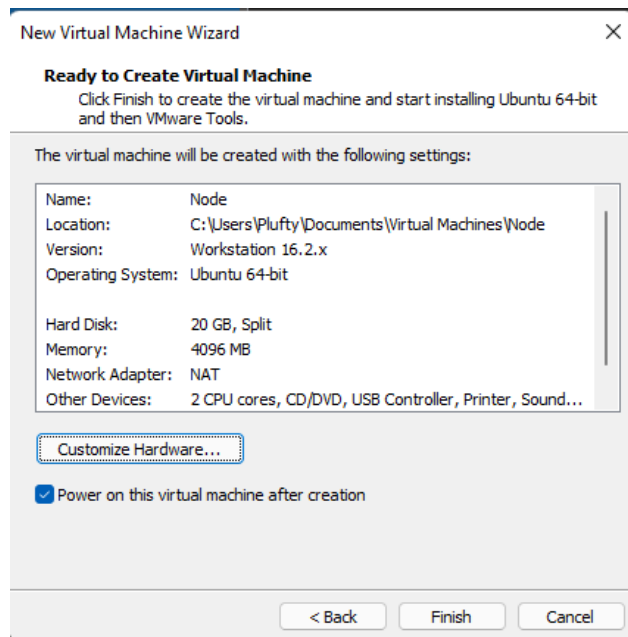
OpenNebula - Como fazer?

O procedimento será executado por mim em duas máquinas virtuais utilizando VMWare, então o primeiro passo será a criação de ambas as máquinas virtuais e ativar a virtualização das mesmas visto que será utilizado o KVM. Será instalado em ambas as máquinas o Ubuntu Server 20.04. Os procedimentos detalhados nesse relatório exceto pelos relacionados ao VMWare podem ser encontrados em :

- [Try OpenNebula on KVM/LXC/FC — OpenNebula 6.2.1 documentation](#)
 - [KVM Node Installation — OpenNebula 6.2.1 documentation](#)
1. Criando a VM FrontEnd que será a Master nesse processo, ela receberá o FrontEnd e será o servidor principal nesse processo, neste caso é recomendado criar uma VM com disco de 40GB devido ao requisito de um dos procedimentos futuros.

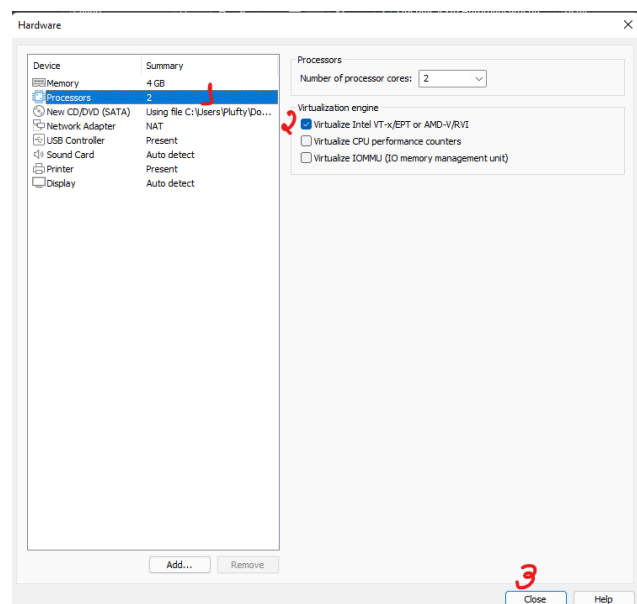
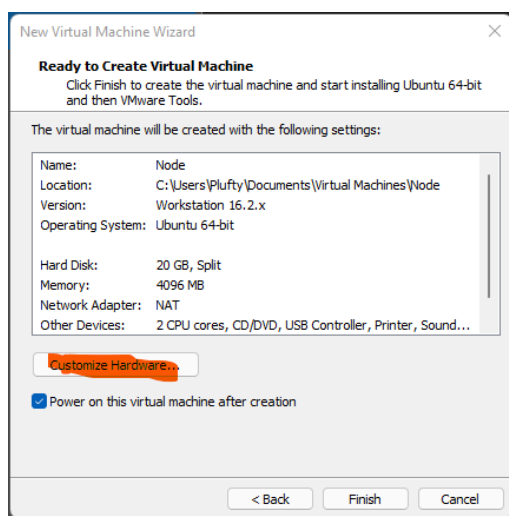


2. Criando a VM Node que será a máquina cliente desse processo, ela receberá o node do OpenNebula.

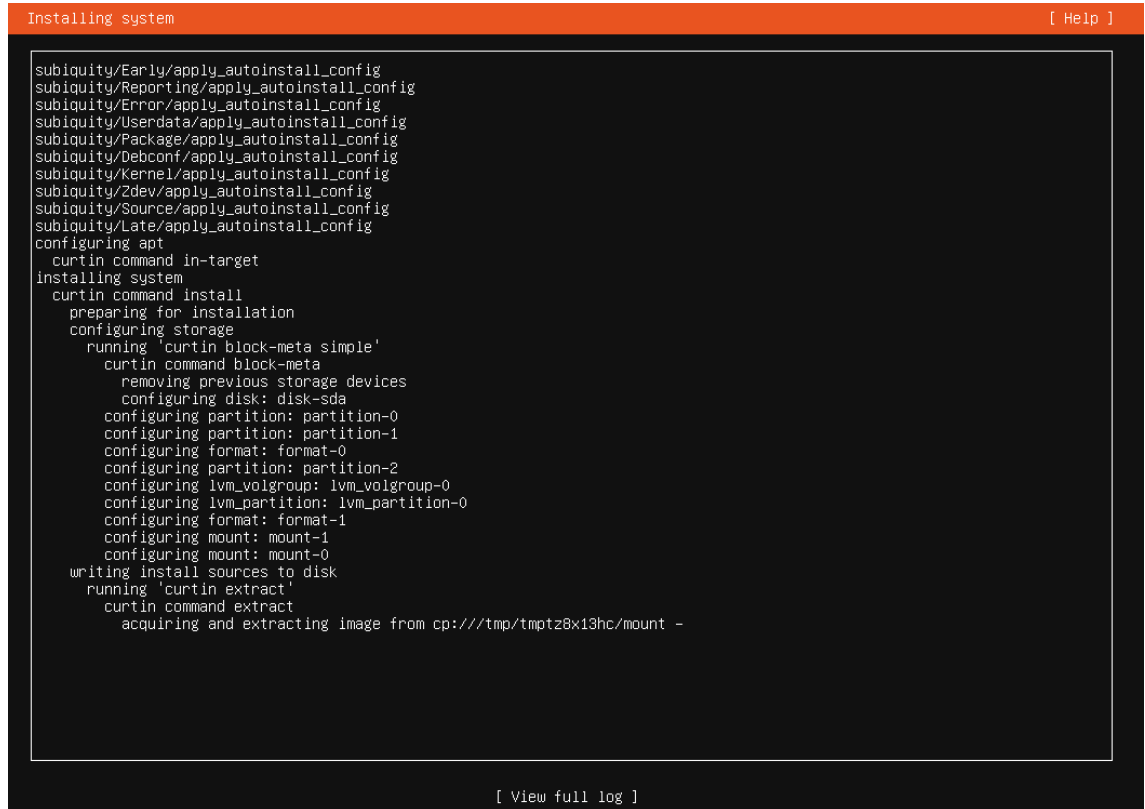


3.

4. Ativando a virtualização no VMWare, ao clicar em Customize Hardware, selecione Processors e ative a virtualização.



5. Instalação do SO



```
Installing system [ Help ]

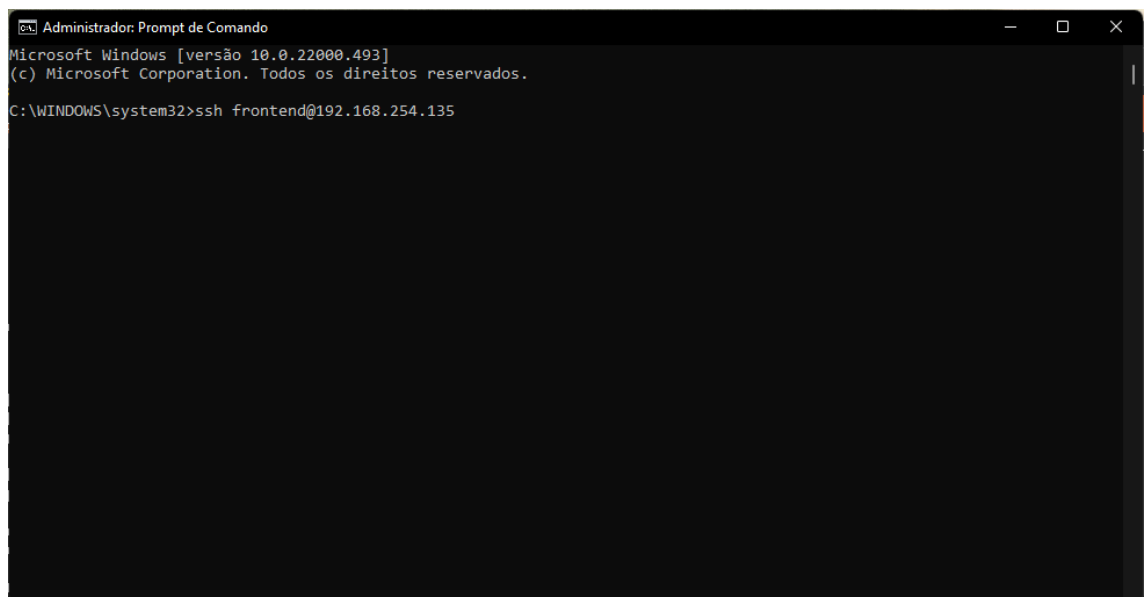
subiquity/Early/apply_autoinstall_config
subiquity/Reporting/apply_autoinstall_config
subiquity/Error/apply_autoinstall_config
subiquity/Userdata/apply_autoinstall_config
subiquity/Package/apply_autoinstall_config
subiquity/Debconf/apply_autoinstall_config
subiquity/Kernel/apply_autoinstall_config
subiquity/Zdev/apply_autoinstall_config
subiquity/Source/apply_autoinstall_config
subiquity/Late/apply_autoinstall_config
configuring apt
  curtin command in-target
installing system
  curtin command install
    preparing for installation
    configuring storage
      running 'curtin block-meta simple'
      curtin command block-meta
        removing previous storage devices
        configuring disk: disk-sda
        configuring partition: partition-0
        configuring partition: partition-1
        configuring format: format-0
        configuring partition: partition-2
        configuring lvm_volgroup: lvm_volgroup-0
        configuring lvm_partition: lvm_partition-0
        configuring format: format-1
        configuring mount: mount-1
        configuring mount: mount-0
    writing install sources to disk
      running 'curtin extract'
      curtin command extract
        acquiring and extracting image from cp:///tmp/tmp28x13hc/mount -

[ View full log ]
```

A instalação do OpenNebula FrontEnd

Neste momento observaremos apenas a máquina FrontEnd para instalar o módulo principal do OpenNebula.

1. Como estamos trabalhando com máquinas virtuais, estarei acessando primeiramente via SSH a máquina FrontEnd para trabalhar sem problemas de desempenho do VMWare.



```
Administrador: Prompt de Comando
Microsoft Windows [versão 10.0.22000.493]
(c) Microsoft Corporation. Todos os direitos reservados.
C:\WINDOWS\system32>ssh frontend@192.168.254.135
```

- ```
frontend@frontend:~$ sudo -i
[sudo] password for frontend:
root@frontend:~#
```

- ```

root@frontend-0 ~# curl -v https://github.com/Qemuhub/minione/releases/latest/download/minione
2022-03-09 22:03:18 -- https://github.com/Qemuhub/minione/releases/latest/download/minione
Resolving github.com (github.com)... 20.201.28.153
Connecting to github.com (github.com)|20.201.28.153|443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/Qemuhub/minione/releases/download/v6.2.0/minione [following]
2022-03-09 22:03:19 -- https://github.com/Qemuhub/minione/releases/download/v6.2.0/minione
HTTP redirect sent to github.com...403
HEAD /objects/githubusercontent.com/github-production-release-asset-26659e/15949252/246432c-6e2e-428e-ba7f-320d3cd8BDeC?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWMYVACW5GJ3A2F72622030RZU5-east-1:2022/03/09:aws_request_auth_data=20220309T221316Z%26X-Amz-Expires=3600&X-Amz-Signature=49f49f1e5dc50ff7f869825e1197acafbb506722a7a1a3580cbcf6807b08&X-Amz-SignedHeaders=host&content-disposition=attachment;filename=minione type=application/octet-stream [following]
2022-03-09 22:03:16 -- https://objects.githubusercontent.com/github-production-release-asset-26659e/15949252/246432c-6e2e-428e-ba7f-320d3cd8BDeC?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWMYVACW5GJ3A2F72622030RZU5-east-1:2022/03/09:aws_request_auth_data=20220309T221316Z%26X-Amz-Expires=3600&X-Amz-Signature=49f49f1e5dc50ff7f869825e1197acafbb506722a7a1a3580cbcf6807b08&X-Amz-SignedHeaders=host&content-disposition=attachment;filename=minione type=application/octet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.111.133, ...
HTTP request sent, awaiting response... 200 OK
Length: 49750 (49K) [application/octet-stream]
Saving to: 'minione'

minione          100%[*****]         48.5K   --.-KB/s   in 0.02s

2022-03-09 22:03:16 (2.40 MB/s) - "minione" saved [49750/49750]

root@frontend-0 ~#
```

- ```
root@frontend:~# sudo bash minione --frontend --force

Checks & detection
Checking augeas is installed SKIP will try to install
Checking free disk space IGNORE
Checking apt-transport-https is installed SKIP will try to install
Checking AppArmor SKIP will try to modify
Checking for present ssh key SKIP
Checking docker is installed SKIP will try to install
Checking python3-pip is installed SKIP will try to install
Checking ansible SKIP will try to install
Checking terraform SKIP will try to install
Checking unzip is installed SKIP will try to install

Main deployment steps:
Install OpenNebula frontend version 6.2
Install Terraform
Install Docker
Install augeas-tools apt-transport-https python3-pip unzip
Install pip 'ansible==2.9.9'
```

Durante a execução desse procedimento será solicitado ao usuário uma confirmação para continuar, apenas digite yes e continue.

```
Do you agree? [yes/no]:
yes

Installation
Updating APT cache OK
Install augeas-tools apt-transport-https python3-pip unzip OK
Updating PIP OK
Install from PyPI 'ansible==2.9.9' OK
Configuring repositories OK
Updating APT cache OK
Installing OpenNebula packages OK
Installing opennebula-provision package OK
Installing TerraForm OK
Create docker packages repository OK
Install docker OK
Start docker service OK
Enable docker service OK

Configuration
Generating ssh keypair in /root/.ssh-oneprovision/id_rsa OK
Add oneadmin to docker group OK
Update network hooks OK
Switching OneGate endpoint in oned.conf OK
Switching OneGate endpoint in onegate-server.conf OK
Switching keep_empty_bridge on in OpenNebulaNetwork.conf OK
Switching scheduler interval in oned.conf OK
Setting initial password for current user and oneadmin OK
Changing WebUI to listen on port 80 OK
Switching FireEdge public endpoint OK
Starting OpenNebula services OK
Enabling OpenNebula services OK
Add ssh key to oneadmin user OK
Update ssh configs to allow VM addresses reusing OK
Ensure own hostname is resolvable OK
Checking OpenNebula is working OK
Exporting [CentOS 7] from Marketplace to local datastore OK
Waiting until the image is ready OK
Updating VM template OK

Report
OpenNebula 6.2 was installed
Sunstone is running on:
 http://179.189.87.179/
FireEdge is running on:
 http://179.189.87.179:2616/
Use following to login:
 user: oneadmin
 password: 5UQ4MjEULn
```

Observe que ao fim deste procedimento nos é dado um Report com os dados de nossa máquina assim como usuário e senha do OpenNebula. Devemos lembrar de salvá-los, esses dados são essenciais, são eles: User: oneadmin Password: 5UQ4MjEULn

## Configurando Open Nebula Node

1. Como no caso da máquina FrontEnd, aqui também utilizarei o ssh e o modo root. Vide passos 1 e 2 do tópico anterior.

2. Importaremos o repositório do OpenNebula com os seguinte comandos:

```
wget -q -O- https://downloads.opennebula.org/repo/repo.key | sudo apt-key add -
```

```
root@node:~# wget -q -O- https://downloads.opennebula.org/repo/repo.key | sudo apt-key add -
OK
```

3. Após isso adicionaremos o repositório com o seguinte comando:

```
echo "deb https://downloads.opennebula.org/repo/6.2/Ubuntu/20.04 stable
opennebula" | sudo tee /etc/apt/sources.list.d/opennebula.list
```

```
root@node:~# echo "deb https://downloads.opennebula.org/repo/6.2/Ubuntu/20.04 stable opennebula" | sudo tee /etc/apt/sources.list.d/opennebula.list
deb https://downloads.opennebula.org/repo/6.2/Ubuntu/20.04 stable opennebula
```

Nesse ponto também utilizaremos o comando apt update para atualizar os pacotes.

```
root@node:~# apt update
Hit:2 http://br.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://br.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://br.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 http://br.archive.ubuntu.com/ubuntu focal-security InRelease
Ign:1 https://downloads.opennebula.io/repo/6.2/Ubuntu/20.04 stable InRelease
Get:6 https://downloads.opennebula.io/repo/6.2/Ubuntu/20.04 stable Release [1,754 B]
Get:7 https://downloads.opennebula.io/repo/6.2/Ubuntu/20.04 stable Release.gpg [490 B]
Get:8 https://downloads.opennebula.io/repo/6.2/Ubuntu/20.04 stable/opennebula amd64 Packages [4,849 B]
Fetched 7,093 B in 5s (1,353 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

4. Agora faremos a instalação do OpenNebula node com o comando

```
apt-get -y install opennebula-node-kvm
```

```
root@node:~# apt-get -y install opennebula-node-kvm
```

5. Após essa instalação, devemos editar o arquivo libvirt.conf para isso podemos usar o comando nano /etc/libvirt/libvirt.conf e devemos editá-lo da seguinte forma:

This is restricted to 'root' by default.

unix\_sock\_group = "oneadmin"

If not using PolicyKit and setting group ownership for access control, then you may want to relax this too.

unix\_sock\_rw\_perms = "0777"

Original

```
This is restricted to 'root' by default.
#unix_sock_group = "libvirt"
unix_sock_group = "oneadmin"

Set the UNIX socket permissions for the R/O socket. This is used
for monitoring VM status only

This setting is not required or honoured if using systemd socket
activation.

Default allows any user. If setting group ownership, you may want to
restrict this too.
#unix_sock_ro_perms = "0777"
unix_sock_ro_perms = "0777"

Set the UNIX socket permissions for the R/W socket. This is used
for full management of VMs

This setting is not required or honoured if using systemd socket
activation.

Default allows only root. If PolicyKit is enabled on the socket,
the default will change to allow everyone (eg, 0777)

If not using PolicyKit and setting group ownership for access
control, then you may want to relax this too.
unix_sock_rw_perms = "0777"
```

Novo

```
This is restricted to 'root' by default.
#unix_sock_group = "libvirt"
unix_sock_group = "libvirt"

Set the UNIX socket permissions for the R/O socket. This is used
for monitoring VM status only

This setting is not required or honoured if using systemd socket
activation.

Default allows any user. If setting group ownership, you may want to
restrict this too.
#unix_sock_ro_perms = "0777"
unix_sock_ro_perms = "0777"

Set the UNIX socket permissions for the R/W socket. This is used
for full management of VMs

This setting is not required or honoured if using systemd socket
activation.

Default allows only root. If PolicyKit is enabled on the socket,
the default will change to allow everyone (eg, 0777)

If not using PolicyKit and setting group ownership for access
control, then you may want to relax this too.
unix_sock_rw_perms = "0770"
```

6. Agora devemos reiniciar o serviço libvirt com o comando `systemctl restart libvirt`

```
root@node:~# systemctl restart libvirt
```

## Configurando SSH

1. O primeiro passo é entrar no usuário oneadmin na máquina FrontEnd e executar o comando
- `ssh-keyscan <frontend> <node> >> /var/lib/one/.ssh/known_hosts`
- onde <frontend> será o nome ou IP da máquina FrontEnd e <node> será o IP ou nome da máquina Node.

```
root@frontend:~# su - oneadmin
oneadmin@frontend:~$ ssh-keyscan 192.168.254.135 192.168.254.138 >> /var/lib/one/.ssh/known_hosts
192.168.254.138:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4
192.168.254.135:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4
192.168.254.135:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4
192.168.254.135:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4
192.168.254.135:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4
192.168.254.135:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4
192.168.254.138:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4
192.168.254.138:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4
192.168.254.138:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4
192.168.254.138:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4
192.168.254.138:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.4
oneadmin@frontend:~$
```

2. Realizaremos alteração da senha na máquina Node com o comando `passwd oneadmin`

```
root@node:~# passwd oneadmin
New password:
Retype new password:
passwd: password updated successfully
```

3. Para habilitar o acesso ssh sem senha ou passwordless ssh utilizaremos o seguinte comando na máquina FrontEnd `ssh-copy-id -i /var/lib/one/.ssh/id_rsa.pub <node>` onde <node> é o IP ou nome da máquina Node.

Logo após esse, também executaremos o comando

`scp -p /var/lib/one/.ssh/known_hosts <node>:/var/lib/one/.ssh/`

```
oneadmin@frontend:~$ ssh-copy-id -i /var/lib/one/.ssh/id_rsa.pub 192.168.254.138
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/var/lib/one/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
oneadmin@192.168.254.138's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '192.168.254.138'"
and check to make sure that only the key(s) you wanted were added.

oneadmin@frontend:~$ scp -p /var/lib/one/.ssh/known_hosts 192.168.254.138:/var/lib/one/.ssh/
known_hosts
100% 1686 2.3MB/s 00:00
```

4. Por fim o comando `scp -p /var/lib/one/.ssh/id_rsa <node>:/var/lib/one/.ssh/` finalizará a questão do passwordless SSH, lembrando que ele deve ser executado na máquina FrontEnd.

```
oneadmin@frontend:~$ scp -p /var/lib/one/.ssh/id_rsa 192.168.254.138:/var/lib/one/.ssh/
id_rsa
100% 2602 3.2MB/s 00:00
oneadmin@frontend:~$
```



5. Como último passo testaremos se nosso SSH sem senha está funcionando normalmente, conectando com ssh <node> e assim que conectar faremos ssh <frontend>, ao confirmar que o acesso foi realizado sem senha, basta dar o comand exit até voltar ao root. Lembrando que onde temos <node> e <frontend> entramos com nome ou IP das máquinas.

```
oneadmin@frontend:~$ ssh 192.168.254.138
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-104-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Wed 09 Mar 2022 11:58:51 PM UTC

System load: 0.0 Processes: 235
Usage of /: 45.8% of 9.78GB Users logged in: 1
Memory usage: 12% IPv4 address for ens33: 192.168.254.138
Swap usage: 0% IPv4 address for virbr0: 192.168.122.1

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

oneadmin@node:~$ ssh 192.168.254.135
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-104-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Wed 09 Mar 2022 11:58:58 PM UTC

System load: 0.16 Processes: 265
Usage of /: 36.7% of 23.74GB Users logged in: 1
Memory usage: 41% IPv4 address for docker0: 172.17.0.1
Swap usage: 0% IPv4 address for ens33: 192.168.254.135

 * Super-optimized for small spaces - read how we shrank the memory
 footprint of MicroK8s to make it the smallest full K8s around.

 https://ubuntu.com/blog/microk8s-memory-optimisation

4 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

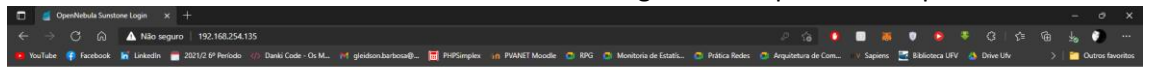
Last login: Wed Mar 9 23:55:09 2022 from 192.168.254.138
oneadmin@frontend:~$ exit
logout
Connection to 192.168.254.135 closed.
oneadmin@node:~$ exit
logout
Connection to 192.168.254.138 closed.
oneadmin@frontend:~$ exit
logout
root@frontend:~#
```

## Configurando Host



Ao fim do nosso tópico de instalação do OpenNebula FrontEnd já temos disponível a interface gráfica, mas optei por citá-la apenas aqui para não confundir as etapas.

1. A GUI pode ser acessada por qualquer máquina na rede utilizando o IP da máquina FrontEnd e seu acesso é feito com o usuário e senha gerados na primeira etapa.



Username  
oneadmin

Password  
\*\*\*\*\*

☒ Keep me logged in

Login

OpenNebula 6.2.0

2. Para adicionar o Host iremos em Infrastructure>Hosts e clicaremos no sinal de +

Dashboard

Instances

Templates

Storage

Network

Infrastructure

Clusters

Hosts

Zones

System

Settings

Not officially supported

OpenNebula 6.2.0

Hosts

oneadmin OpenNebula

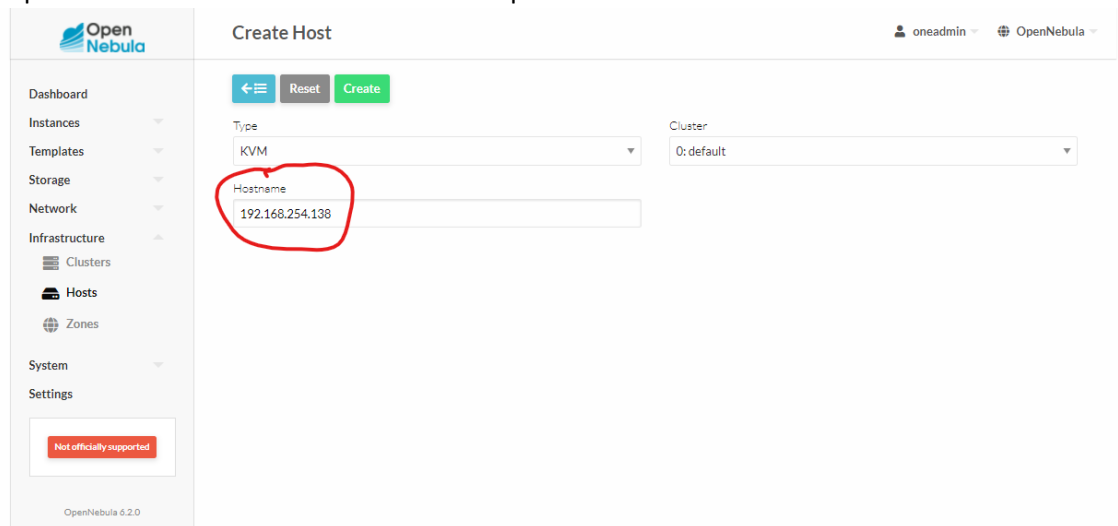
3 + Select cluster Enable Disable Offline

| ID                         | Name | Cluster | RVMs | Allocated CPU | Allocated MEM | Status |
|----------------------------|------|---------|------|---------------|---------------|--------|
| There is no data available |      |         |      |               |               |        |

Showing 0 to 0 of 0 entries

TOTAL 0 ON 0 OFF 0 ERROR

3. Aqui entraremos com o IP ou nome da máquina Node e clicaremos em Create.



Open Nebula

Create Host

oneadmin OpenNebula

Dashboard

Instances

Templates

Storage

Network

Infrastructure

Clusters

Hosts

Zones

System

Settings

Not officially supported

OpenNebula 5.2.0

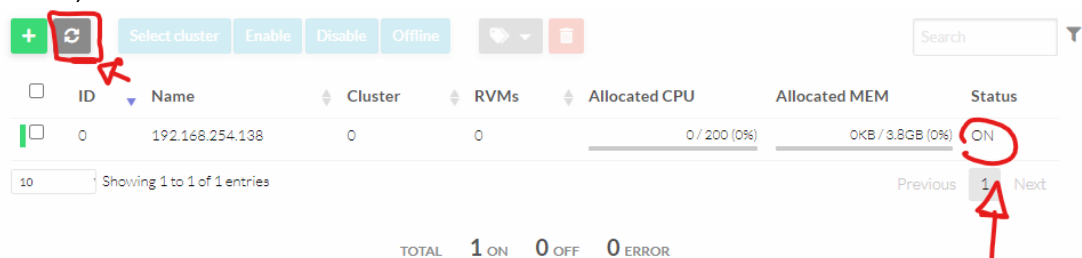
Type: KVM

Cluster: 0: default

Hostname: 192.168.254.138

Reset Create

4. Por fim, basta clicar em atualizar até seu host ficar em estado ON.



+ Refresh Select cluster Enable Disable Offline Search

| ID | Name            | Cluster | RVMs | Allocated CPU | Allocated MEM    | Status |
|----|-----------------|---------|------|---------------|------------------|--------|
| 0  | 192.168.254.138 | 0       | 0    | 0 / 200 (0%)  | 0KB / 3.8GB (0%) | ON     |

Showing 1 to 1 of 1 entries

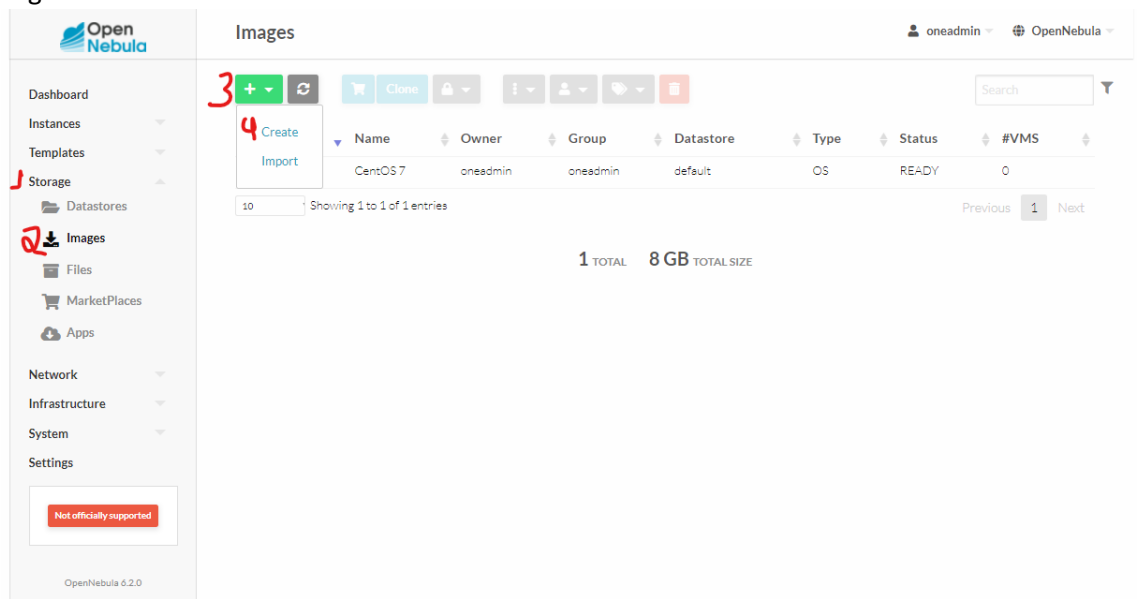
Previous 1 Next

TOTAL 1 ON 0 OFF 0 ERROR

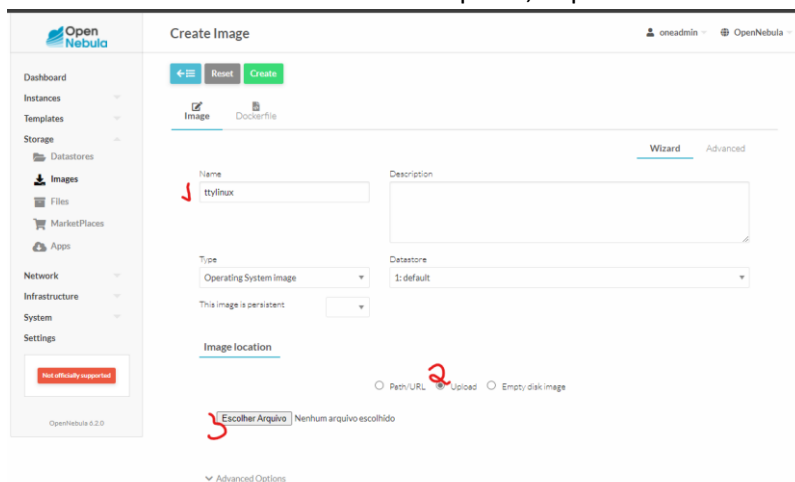
## Subindo uma VM no OpenNebula

Devido a limitações de espaço em minhas VM's não foi possível utilizar a imagem padrão do CentOS, então foi necessário upar um minimal Linux e gerar um novo template com a minha imagem.

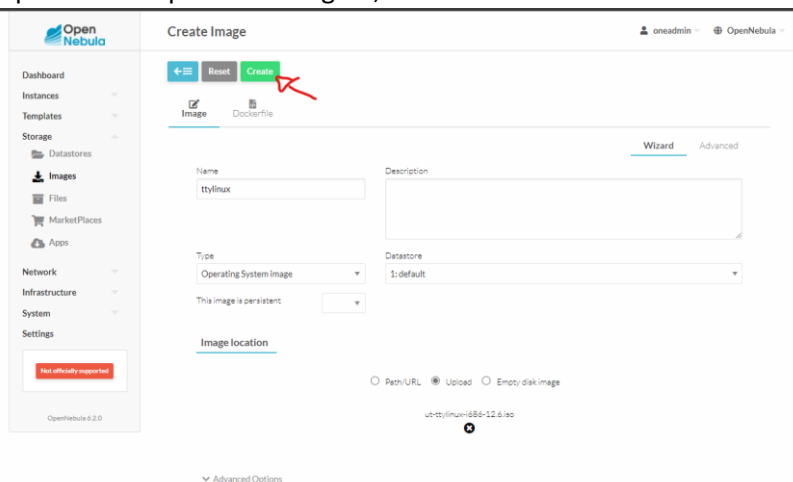
1. Para adicionar a nova imagem vamos a Storage>Images e clicaremos no + e em seguida em Create.



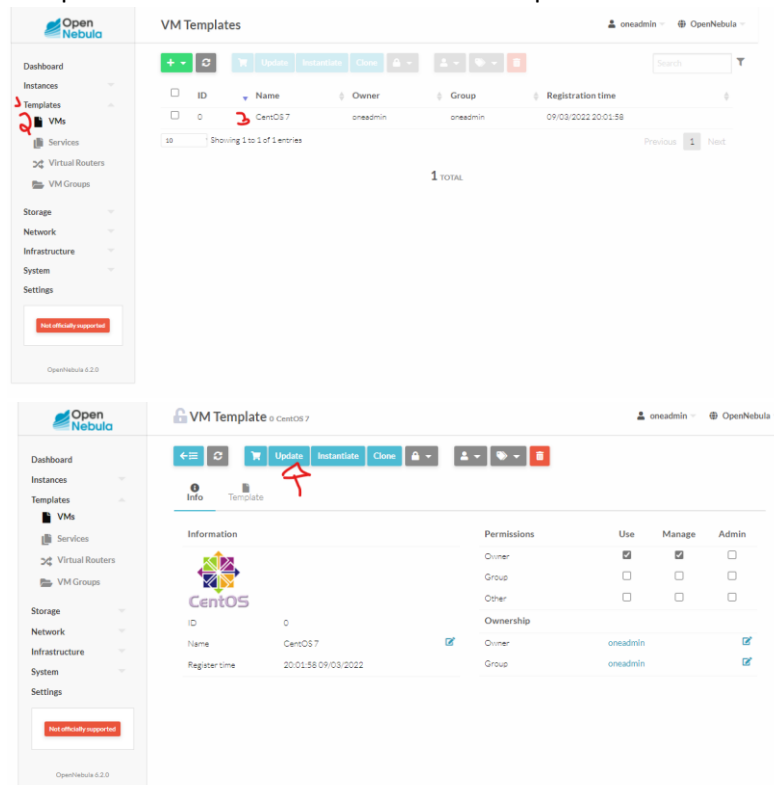
2. Preenchemos o nome e escolhemos upload, depois disso fazemos upload da imagem.



Após fazer o upload da imagem, clicamos em Create.



3. Após esse passo substituiremos a imagem no template padrão em Templates>VMs>CentOS7 e selecionamos Update.

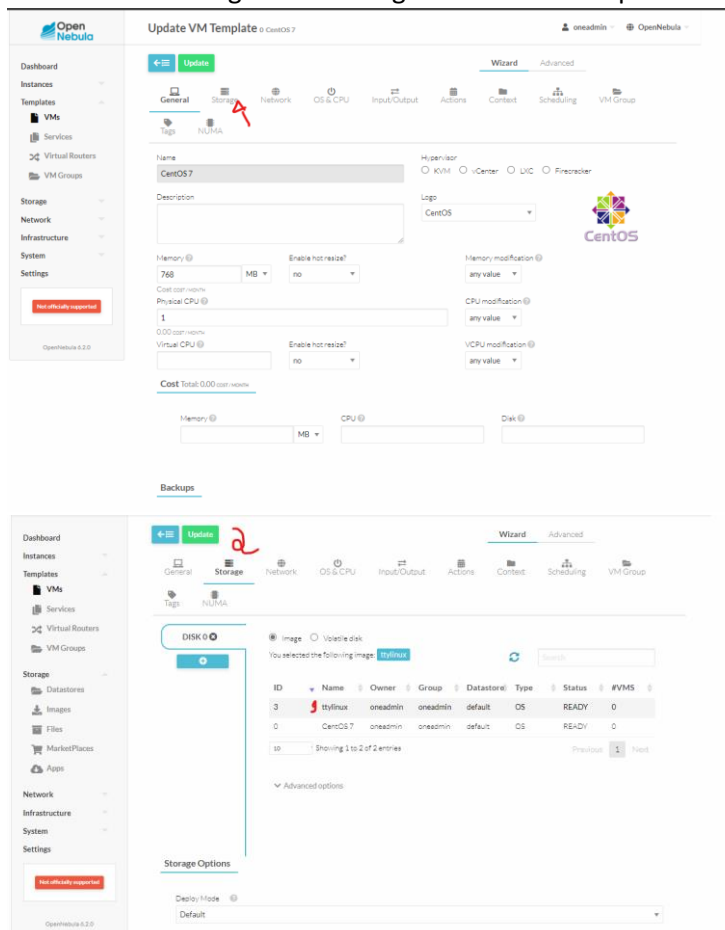


The screenshot shows the OpenNebula web interface. On the left is a sidebar with navigation links: Dashboard, Instances, Templates, VMs, Services, Virtual Routers, VM Groups, Storage, Network, Infrastructure, System, and Settings. The main panel is titled 'VM Templates' and shows a table with one entry: ID 0, Name CentOS7, Owner onedadmin, Group onedadmin, and Registration time 09/03/2022 20:01:58. Below the table, it says '1 TOTAL'. A red arrow points to the 'Update' button in the top toolbar. Below the table, there is a 'VM Template CentOS 7' section with tabs for 'Info' and 'Template'. The 'Info' tab is active, showing details for the CentOS7 template, including its ID, Name, Registration time, and a table of permissions and ownership.

| Information | Permissions | Use                                 | Manage                              | Admin                    |
|-------------|-------------|-------------------------------------|-------------------------------------|--------------------------|
| Owner       |             | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Group       |             | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |
| Other       |             | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |

| Ownership | Owner     | Group |
|-----------|-----------|-------|
| Owner     | onedadmin |       |
| Group     | onedadmin |       |

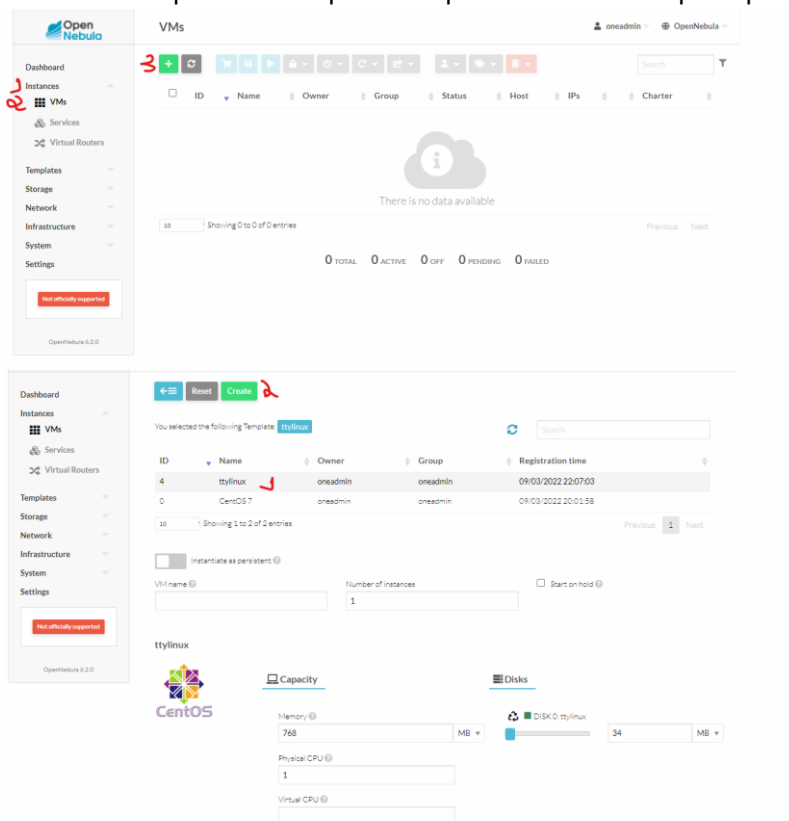
4. Trocaremos a imagem em storage e clicamos em update.



The screenshot shows the OpenNebula web interface for the 'Update VM Template' wizard. The 'Storage' tab is selected, and a red arrow points to the 'Storage' button in the top toolbar. The 'DISK 0' section is expanded, showing a table of available images. The table has columns: ID, Name, Owner, Group, Datastore, Type, Status, and #VMS. The first row shows ID 3, Name ttflinux, Owner onedadmin, Group onedadmin, Datastore default, Type OS, Status READY, and #VMS 0. The second row shows ID 0, Name CentOS7, Owner onedadmin, Group onedadmin, Datastore default, Type OS, Status READY, and #VMS 0. A red arrow points to the 'Update' button in the top toolbar. Below the table, there is a 'Storage Options' section with a 'Disk Mode' dropdown set to 'Default'.

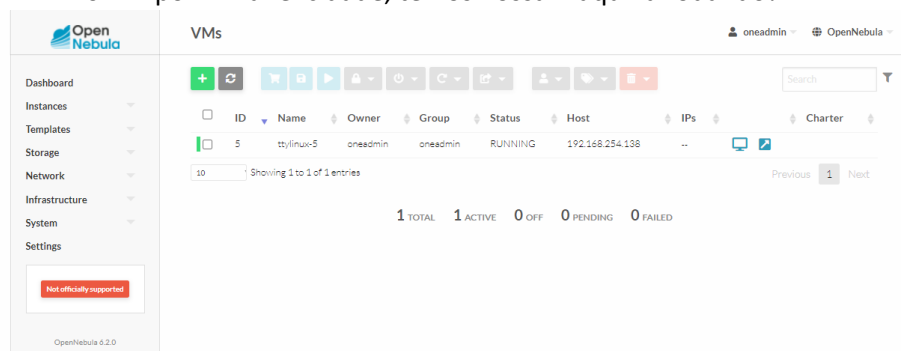
| ID | Name     | Owner     | Group     | Datastore | Type | Status | #VMS |
|----|----------|-----------|-----------|-----------|------|--------|------|
| 3  | ttflinux | onedadmin | onedadmin | default   | OS   | READY  | 0    |
| 0  | CentOS7  | onedadmin | onedadmin | default   | OS   | READY  | 0    |

5. Por fim vamos a Instances>VMs e clicamos em +, selecionamos nosso template e criamos a máquina. Obs: aqui eu troquei o nome do template para ttylinux.



The screenshot shows the OpenNebula web interface. On the left is a sidebar with navigation links: Dashboard, Instances, VMs, Services, Virtual Routers, Templates, Storage, Network, Infrastructure, System, and Settings. The main area is titled 'VMs' and shows a table with columns: ID, Name, Owner, Group, Status, Host, IPs, and Charter. The table is empty, displaying 'There is no data available'. Below the table, there are statistics: 0 TOTAL, 0 ACTIVE, 0 OFF, 0 PENDING, 0 FAILED. A red arrow points to the '+' button in the top toolbar. Another red arrow points to the 'Create' button in the top toolbar. Below the 'Create' button, there is a section for selecting a template. It shows 'You selected the following Template: ttylinux'. Below this, there is a table with columns: ID, Name, Owner, Group, and Registration time. The table contains two entries: ID 4, Name 'ttylinux', Owner 'oneadmin', Group 'oneadmin', Registration time '09/03/2022 22:07:03'; and ID 0, Name 'CentOS 7', Owner 'oneadmin', Group 'oneadmin', Registration time '09/03/2022 20:01:58'. Below the table, there are checkboxes for 'Instantiate as persistent' and 'Start on hold'. There are also input fields for 'VM name' and 'Number of instances' (set to 1). At the bottom, there is a section for 'Capacity' and 'Disks'. The 'Capacity' section shows 'Memory' (768 MB) and 'Physical CPU' (1). The 'Disks' section shows 'DISK0: ttylinux' (34 MB). The 'ttylinux' logo and 'CentOS' logo are also visible.

6. E por fim a felicidade, temos nossa máquina rodando!



The screenshot shows the OpenNebula web interface. On the left is a sidebar with navigation links: Dashboard, Instances, VMs, Services, Virtual Routers, Templates, Storage, Network, Infrastructure, System, and Settings. The main area is titled 'VMs' and shows a table with columns: ID, Name, Owner, Group, Status, Host, IPs, and Charter. The table contains one entry: ID 5, Name 'ttylinux-5', Owner 'oneadmin', Group 'oneadmin', Status 'RUNNING', Host '192.168.254.138', IPs '--', and Charter '--'. Below the table, there are statistics: 1 TOTAL, 1 ACTIVE, 0 OFF, 0 PENDING, 0 FAILED. A red arrow points to the '+' button in the top toolbar. Another red arrow points to the 'ttylinux-5' VM in the table.

