

**INSTITUTO INFNET
ESCOLA SUPERIOR DE TECNOLOGIA**



**Projeto de Bloco - Arquitetura e
Infraestrutura de aplicações**

TESTE DE PERFORMANCE – TP6, 7 e 8

ALUNO: André Muller Lima e Silva

Projeto de Bloco

Sumário

Capítulo 1 - Introdução ao projeto

- 1.1** - Introdução
- 1.2** - O que vai ser tratado e explicações sobre a empresa
- 1.3** - Justificativa da relevância do problema em questão
- 1.4** - Descrição da aplicação de será usada no projeto
- 1.5** - Desenho de como será organizado a solução de acordo com o sistema de virtualização utilizado

Capítulo 2 - Planejamento e descrição

- 2.1** - Descrição textual da solução de virtualização escolhida
- 2.2** - Comparação entre outras formas de implementar a solução, com ferramentas diferentes
- 2.3** - Planejamento passo a passo da implementação do projeto
- 2.4** - Cronograma do projeto

Capítulo 3 -

- 3.1** - Prints com o passo a passo para a execução do ambiente

Capítulo 4-

- 4.1** - Conclusão do projeto
- 4.2** - Melhoria futura
- 4.3** - Passos do commit para o Github
- 4.4** - Link do Github com o projeto

Capítulo 1

1.1 Introdução

O projeto da empresa ABC, do ramo editorial, consiste em criar uma página WEB para divulgação de novos lançamentos e futuros projetos feitos pela empresa, além de funcionar como porta de entrada para o time de vendas e também para recolher feedbacks de clientes, sobre serviços já prestados, e até sugestões.

A empresa não conta com um site comercial próprio, o foco da empresa é vendas físicas em livrarias e lojas onlines de outras empresas, amazon, americanas, saraiva e etc.

Com esse cenário, a diretoria da empresa percebeu que pelo fato de não termos nenhum canal de comunicação pela internet podíamos estar perdendo clientes, afinal estamos na era digital.

Com isso, foi pedido que o time de infraestrutura montasse um projeto para que fosse criado um website que suprisse as necessidades descritas pela diretoria, porém com os custos mais baixos possíveis, levando em consideração tudo que foi passado, a equipe de infraestrutura da empresa optou por utilizar a plataforma wordpress para fazer a implementação do website, por todas as facilidades que a aplicação promove.

1.2 O que vai ser tratado e explicações sobre a empresa.

Tipo de Negócio - Empresa ABC do ramo de editorial, empresa antiga, com poucas tecnologias implementadas, vendo que o mundo tem se transformado cada vez mais digital, temos que correr para tirar o atraso e equiparar nossa empresa com o mundo atual.

Será tratado neste projeto, a criação de um website para divulgação de novos lançamentos e futuros projetos feitos pela empresa, a partir da plataforma wordpress em cima de uma estrutura virtualizada VMware, já presente em nossa empresa.

1.3 - Justificativa da relevância do problema em questão

Com a inclusão de nossa página WEB, nós conseguiremos trazer o cliente para mais perto de nossa empresa, entendendo melhor o que os clientes necessitam e atendê-los de uma melhor forma, assim fidelizando os clientes já existentes e conseguindo atrair novos clientes tanto pela WEB, quanto pelas boas experiências de nossos clientes, além de ter maior conhecimento do nosso alcance em divulgação de novos produtos e projetos dos quais fazemos parte.

Entendendo que cada vez mais as empresas necessitam saber mais sobre seus clientes, a nova página WEB da ABC vem com o intuito de melhorar a experiência do nosso cliente como um todo,

ficando ciente de novos lançamentos, promoções em lojas parceiras, links diretos de marketplace para vendas em sites de outras empresas, canais de atendimento ao cliente pelo site com chat direto, e-mail e telefone.

Com todas essas funcionalidades funcionando, com certeza a empresa agrega valor ao negócio.

1.4 - Descrição da aplicação de será usada no projeto

Aplicação usada no projeto será o **Wordpress**, sistema livre e aberto de gestão de conteúdo para internet, plataforma usada para administrar nossa página WEB, aplicação distribuída (aplicação e banco de dados(MySQL)) gratuita e de código aberto, onde nossa equipe fará o gerenciamento do código fonte.

O desenvolvimento da página será feito por desenvolvedores terceirizados com acompanhamento direto de nossos desenvolvedores.

Pré requisitos:

Mundo físico - Dois servidores com 24 cores e 64 GBs de RAM e 10 TBs de armazenamento em disco, 4 interfaces de rede de 1GbE cada.

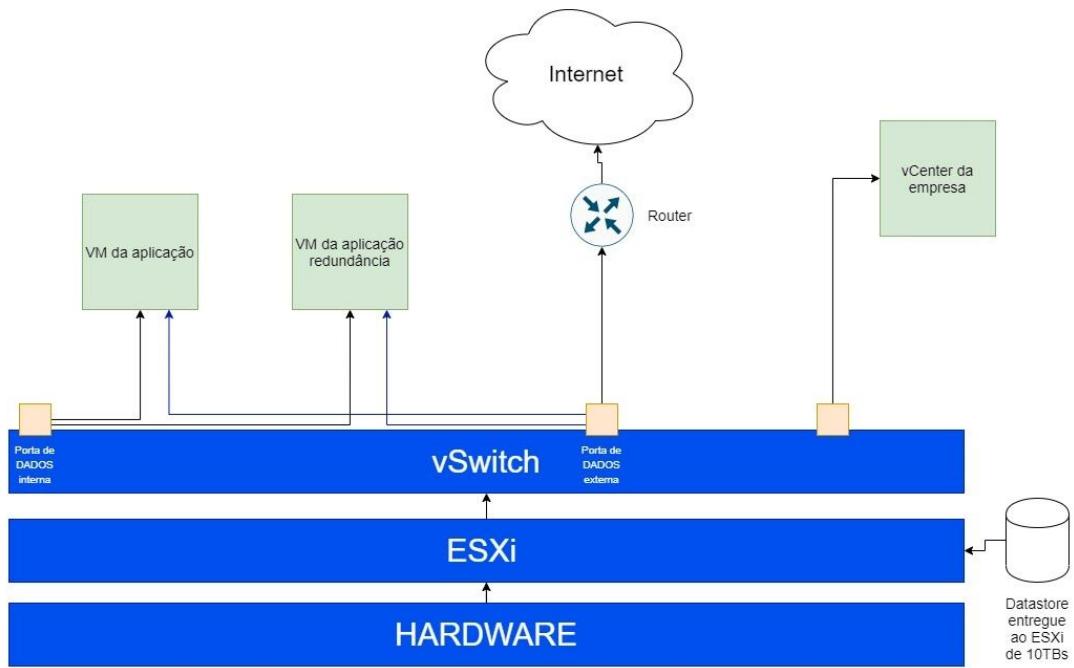
Mundo Virtual - Dois servidores virtuais Linux ubuntu, com 8 vCPUs, 16 GBs de RAM, 2 TBs de armazenamento e 2 vNics de 1GbE cada.

1.5 - Desenho de como será organizado a solução de acordo com o sistema de virtualização utilizado

O diagrama abaixo representa a infraestrutura que será usada nos dois servidores disponibilizados pela empresa para o projeto (visando redundância), ou seja cada servidor físico sustentará essa infraestrutura mostrada abaixo:

OBS: Cada item utilizado no desenho abaixo já foi descrito antes no projeto, número de CPU, RAM, armazenamento e etc.

OBS: Todas as conexões usadas no desenho devem ser consideradas duplas, visando a redundância no ambiente.



Capítulo 2

2.1 - Descrição textual da solução de virtualização escolhida

A arquitetura de virtualização que será utilizada para sustentar nossa aplicação será solução da VMware. Será implementado em dois servidores físicos redundantes entre si, da empresa o sistema operacional ESXi da VMware (A solução gratuita da VMware), onde nesse ESXi será criado as máquinas virtuais que levarão a aplicação (Wordpress), serão implementadas 2 VMs Ubuntu, redundantes entre si para o caso de falhas, esse ESXi será entregue ao vCenter da empresa para gestão do mesmo.

2.2 - Comparação entre outras formas

de implementar a solução, com ferramentas diferentes

Porquê foi escolhido a solução VMware ?

Foi escolhido a solução VMware pelo fato de já ser implementada na empresa, e por não ter custos extras, por estarmos usando a licença gratuita do ESXi e utilizando Linux para o ambiente.

Em comparação com outras tecnologias, como por exemplo, a AWS, Foi escolhida a VMware por questões financeiras principalmente, sabendo que na AWS teríamos que arcar com custos de serviços que seriam essenciais para o mínimo funcionamento do ambiente como a instância EC2 e um serviço de storage deles, como EBS, Glacier e S3 por exemplo.

Existe outras formas de tecnologia para serem usadas neste cenário, como a nuvem privada openstack, que poderia também ser usada em seu nível mais baixo, o packstack, com a função de rodar 2 máquinas virtuais que são o necessário para a aplicação funcionar de uma forma eficaz, porém implementar uma outra tecnologia para tão pouco uso não fazia sentido dentro do cenário atual da empresa, onde já é utilizado a solução VMware.

2.3 - Planejamento passo a passo da implementação do projeto

Implementação da nossa solução:

Etapa 1- Instalar nos servidores físicos a imagem do ESXi

Etapa 2- Executar todas as configurações para que o ESXi esteja em condições operacionais de funcionamento seguindo as normas de segurança da empresa, Configurando a rede, botando atrás do firewall e os demais procedimentos de segurança.

Etapa 3- Entregar esse ESXi para o vCenter que faz a gestão do

ambiente VMware da empresa.

Etapa 4- Fazer os testes no ESXi e garantir que está em pleno funcionamento.

Etapa 5 - Criar as 2 máquinas virtuais ubuntu server dentro do ESXi

Etapa 6 - Fazer as configurações do ubuntu server e deixar ele pronto para a instalação do wordpress

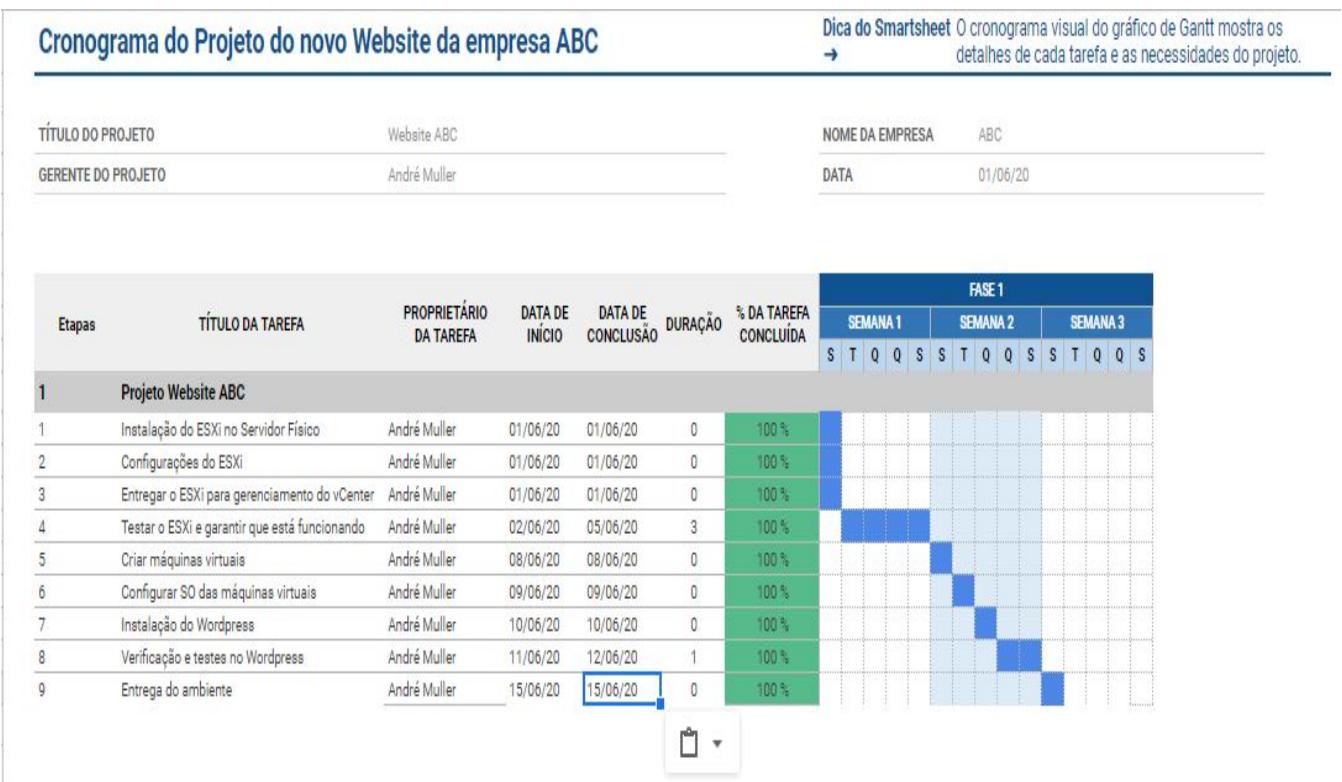
Etapa 7 - começa o processo de instalação do wordpress via container (que será documentado posteriormente no projeto).

Etapa 8 - Verificar se tudo foi instalado corretamente e executar os testes necessários

Etapa 9 - Entregar o ambiente pronto para a criação do novo Website.

2.4 - Cronograma do projeto

O projeto está cronometrado para terminar em 2 semanas, podendo terminar antes dependendo do nível de velocidade que as etapas descritas forem sendo feitas, praticamente dependendo do tempo hábil do analista escalado para executar as tarefas, assim como atrasos também podem acontecer devido a mesma causa, porém por ser um projeto simples a probabilidade de atraso é mínima.



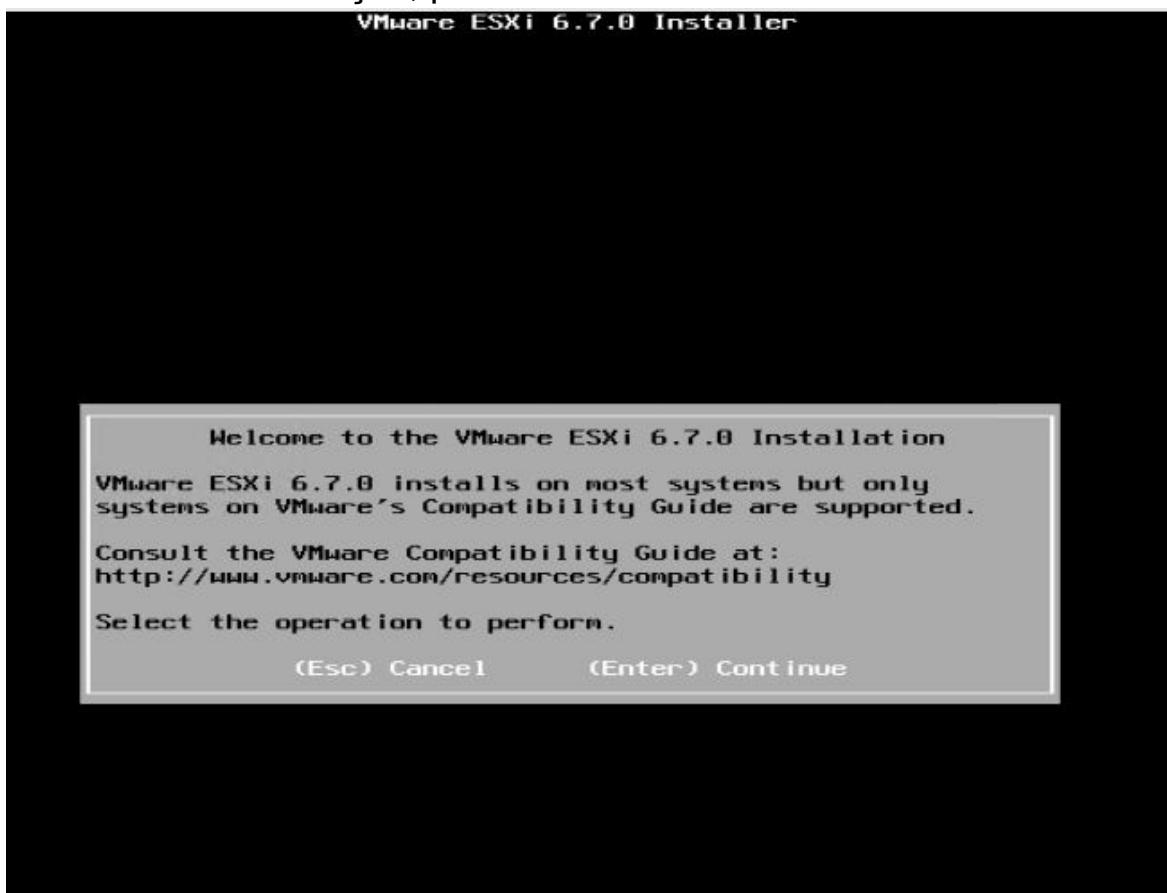
Capítulo 3

3.1 prints contendo o passo a passo para a execução do ambiente.

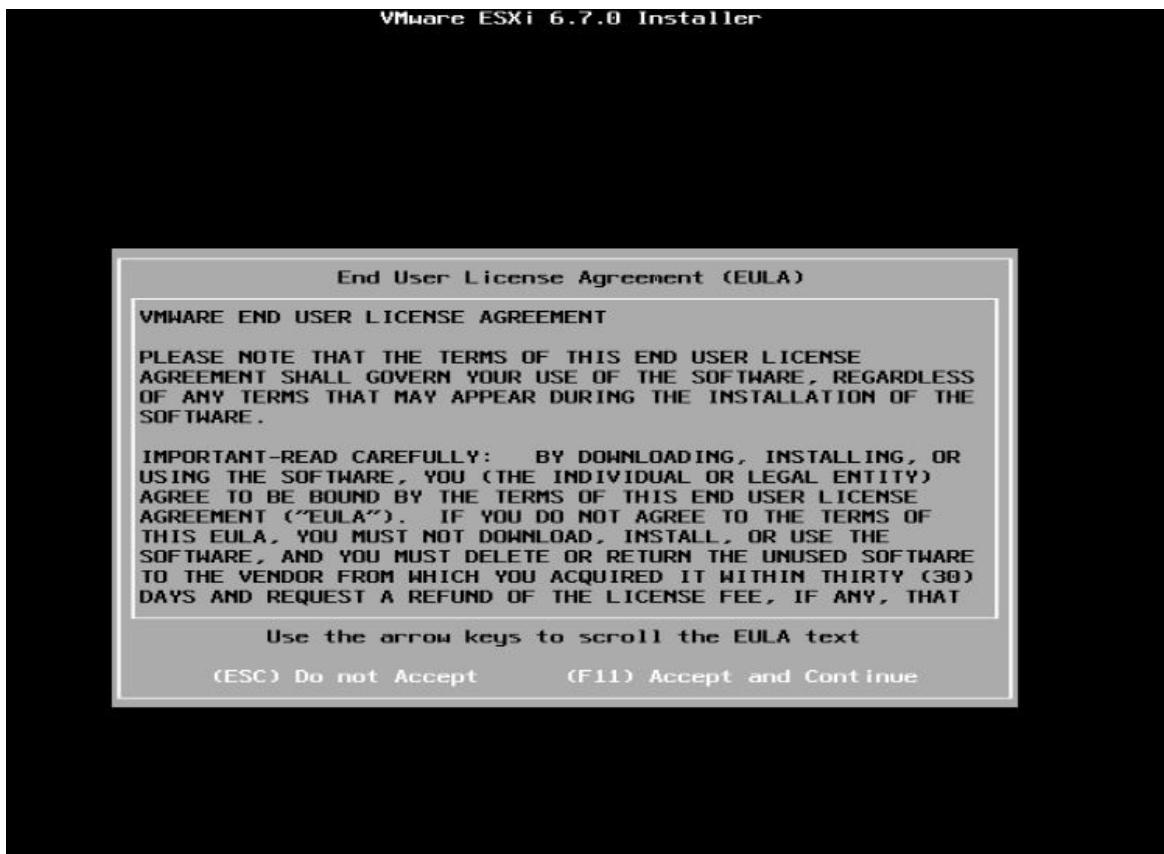
OBS: Todo o ambiente é redundante, os passos descritos abaixo devem ser feitos para os 2 servidores físicos disponibilizados pela empresa.

Instalação do ESXi -

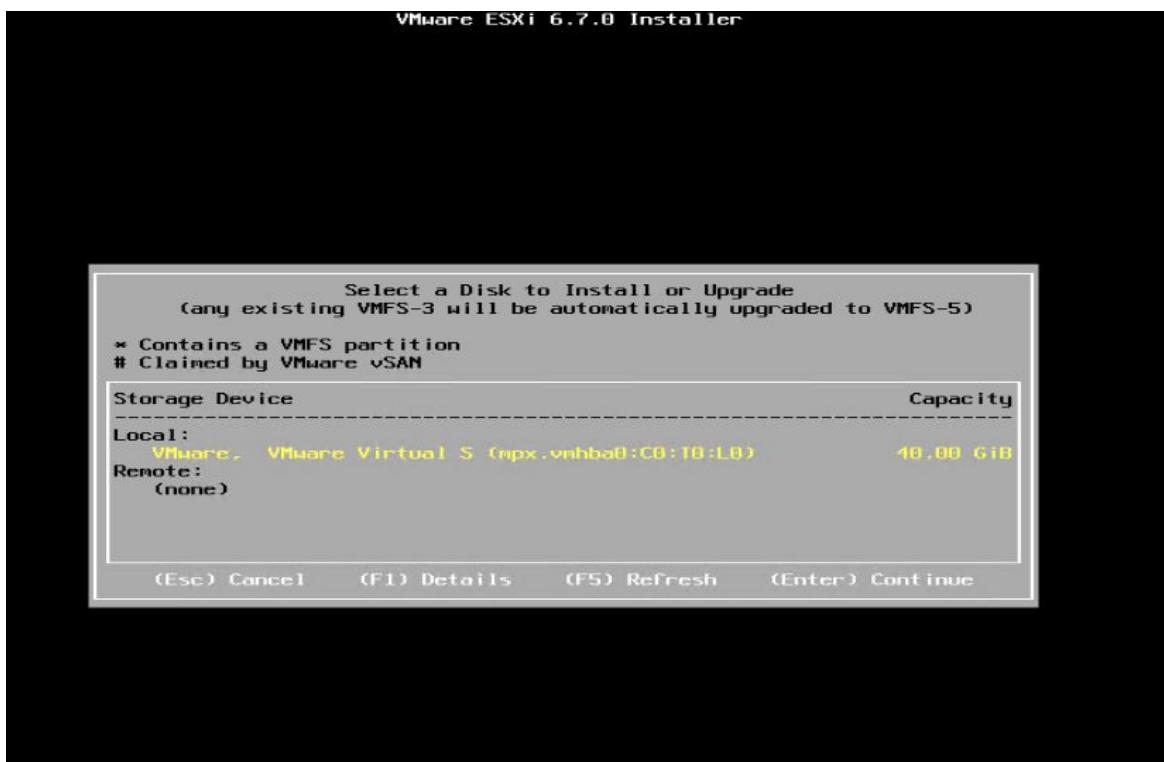
Tela inicial da instalação, pressione enter.



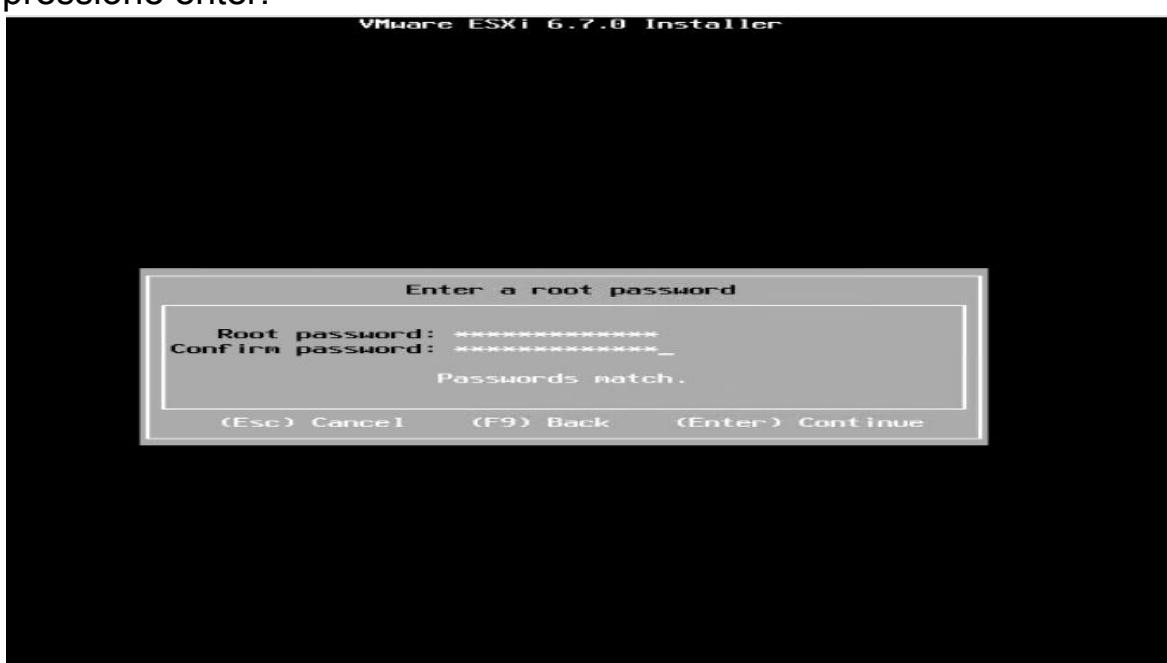
Tela da da End user license, aperte F11 accept and continue.



Parte onde selecionamos o disco onde iremos instalar o ESXi, selecione Local e pressione enter.



Tela onde criamos o password, crie um password forte com caracteres especiais, números e letras maiúsculas e minúsculas e pressione enter.



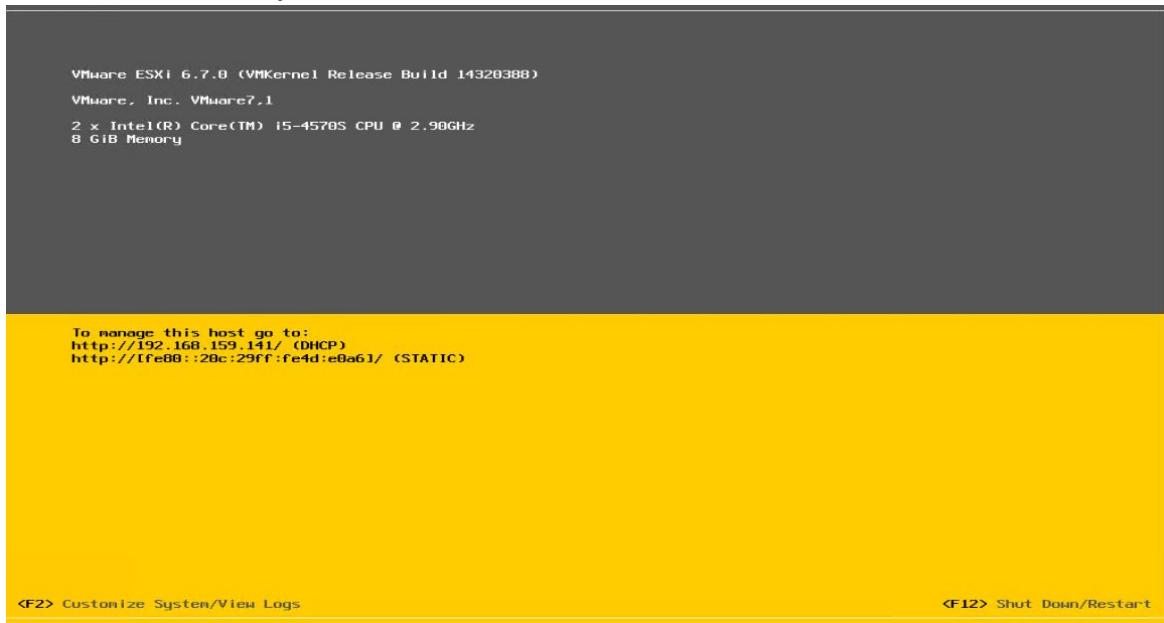
Tela da confirmação da instalação, pressione f11 Install



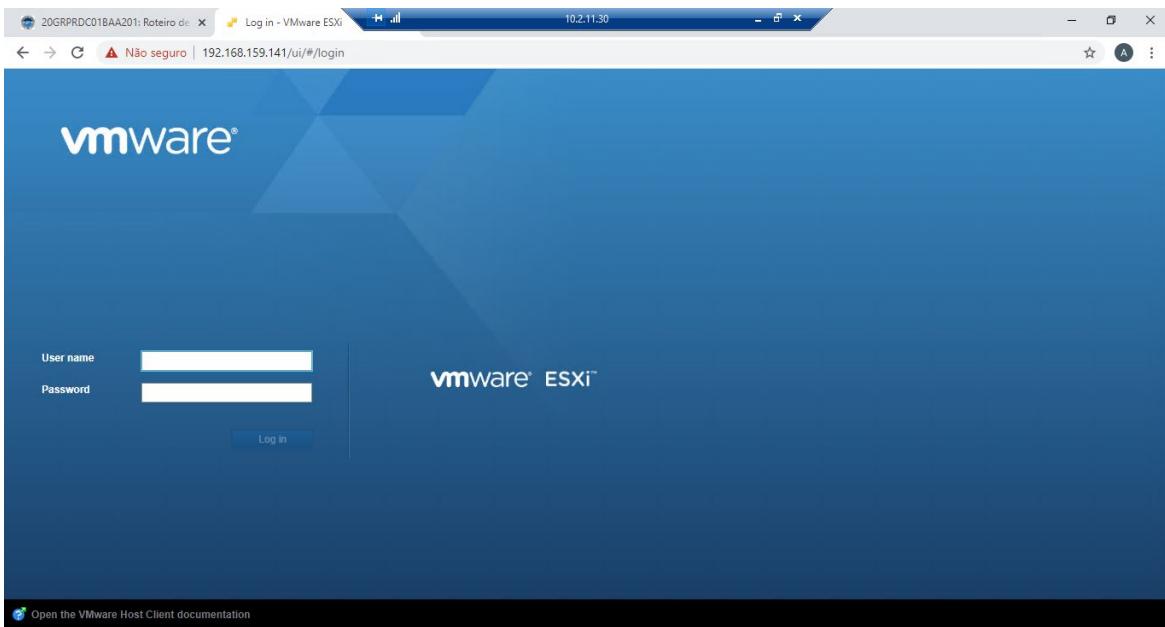
Tela da instalação completa, somente pressionar enter e rapidamente tirar o disco de boot no VMware.



Tela do ESXi instalado e devidamente pronto para ser acessado via browser, com o ip fornecido nessa tela: 192.168.159.141



Tela do browser, onde acessamos o ESXi para fazermos a gestão.



Tela com o ESXi aberto no browser, onde faremos as criações das VMs e adicionar o storage compartilhado.

Task	Target	Initiator	Queued	Started	Result	Completed
Update Options	localhost.localdomain	root		08/08/2020 14:13:35	Completed successfully	08/08/2020 14:13:35
Auto Start Power On	localhost.localdomain	root		08/08/2020 14:11:47	Completed successfully	08/08/2020 14:11:47

Agora faremos a configuração do storage e as configurações de compartilhamento NFS.

Temos a criação do volume de storage que iremos entregar ao ESXi, com RAID 5.

Storage / Pools / Create Pool

Pool Manager

Name * Encryption

Available Disks

<input checked="" type="checkbox"/>	Disk	Type	Capacity	>
<input checked="" type="checkbox"/>	da1	UNKNO	20 GiB	>
<input checked="" type="checkbox"/>	da2	UNKNO	20 GiB	>
<input checked="" type="checkbox"/>	da3	UNKNO	20 GiB	>
<input checked="" type="checkbox"/>	da4	UNKNO	20 GiB	>

4 selected / 4 total

Data VDevs

<input type="checkbox"/>	Disk	Type	Capacity	>
<input type="checkbox"/>				>

No data to display
0 selected / 0 total

Stripe
Estimated raw capacity: 0 B

Actions:

Pool Manager

Name * Encryption

Available Disks

<input type="checkbox"/>	Disk	Type	Capacity	>
<input type="checkbox"/>				>

No data to display
0 selected / 0 total

Data VDevs

<input type="checkbox"/>	Disk	Type	Capacity	>
<input type="checkbox"/>	da1	UNKNOWN	20 GiB	>
<input type="checkbox"/>	da2	UNKNOWN	20 GiB	>
<input type="checkbox"/>	da3	UNKNOWN	20 GiB	>
<input type="checkbox"/>	da4	UNKNOWN	20 GiB	>

0 selected / 4 total

Actions:

Abaixo temos o volume criado.

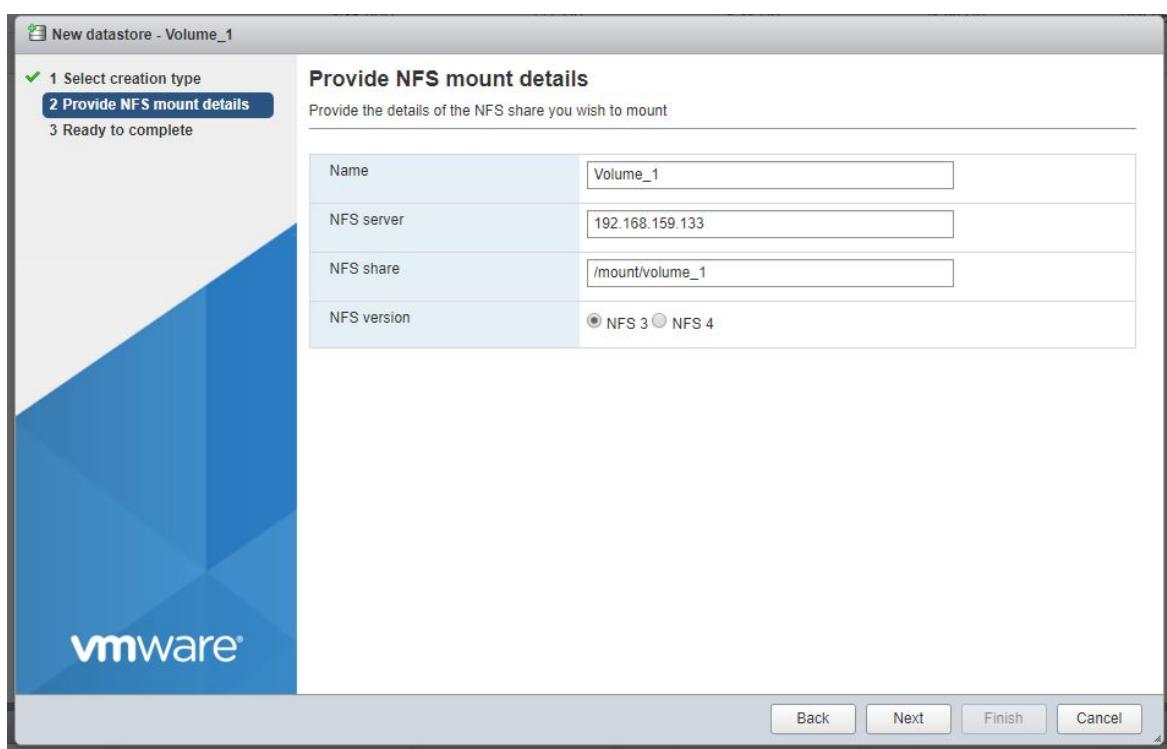
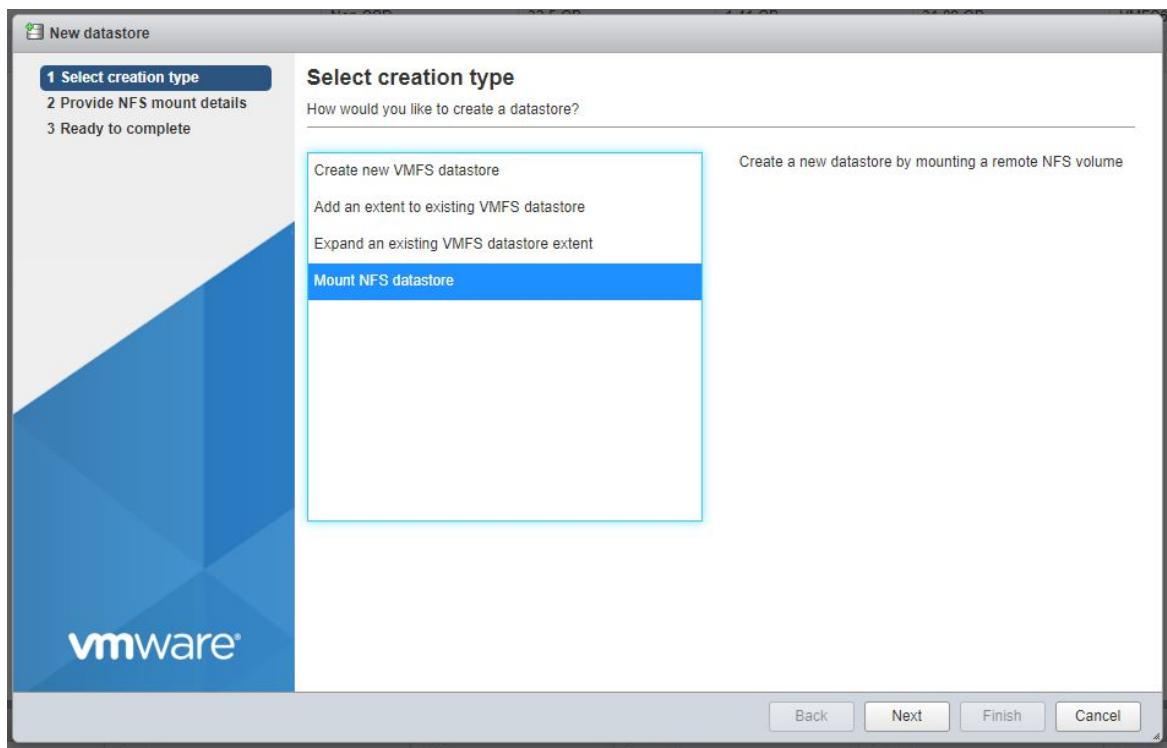
The screenshot shows the FreeNAS web interface under the 'Storage / Pools' section. A single pool, 'Volume_1', is listed as healthy. The table provides detailed information about the pool's configuration:

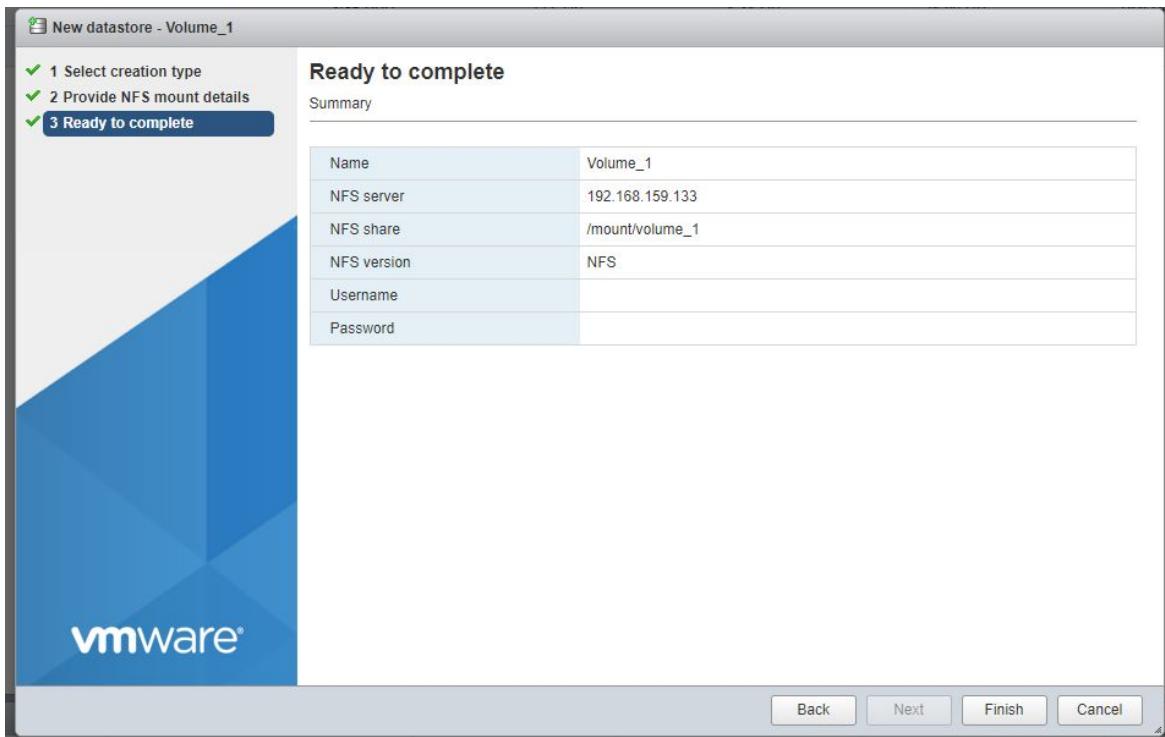
Name	Type	Used	Available	Compression	Compression Ratio	Readonly	Dedup	Comments
Volume_1	dataset	1.04 MiB	50.32 GiB	lz4	1.00x	false	off	

Agora nós criamos o compartilhamento NFS, para entregarmos ao ESXi.

The screenshot shows the FreeNAS web interface under the 'Sharing / NFS / Add' section. A new NFS share is being created at the path '/mnt/Volume_1'. The 'Description' section contains two checkboxes: 'All disks' and 'Read Only'. The 'SAVE' button is highlighted in yellow.

Abaixo temos a criação do Datastore no ESXi, onde usaremos o volume de storage criado anteriormente.

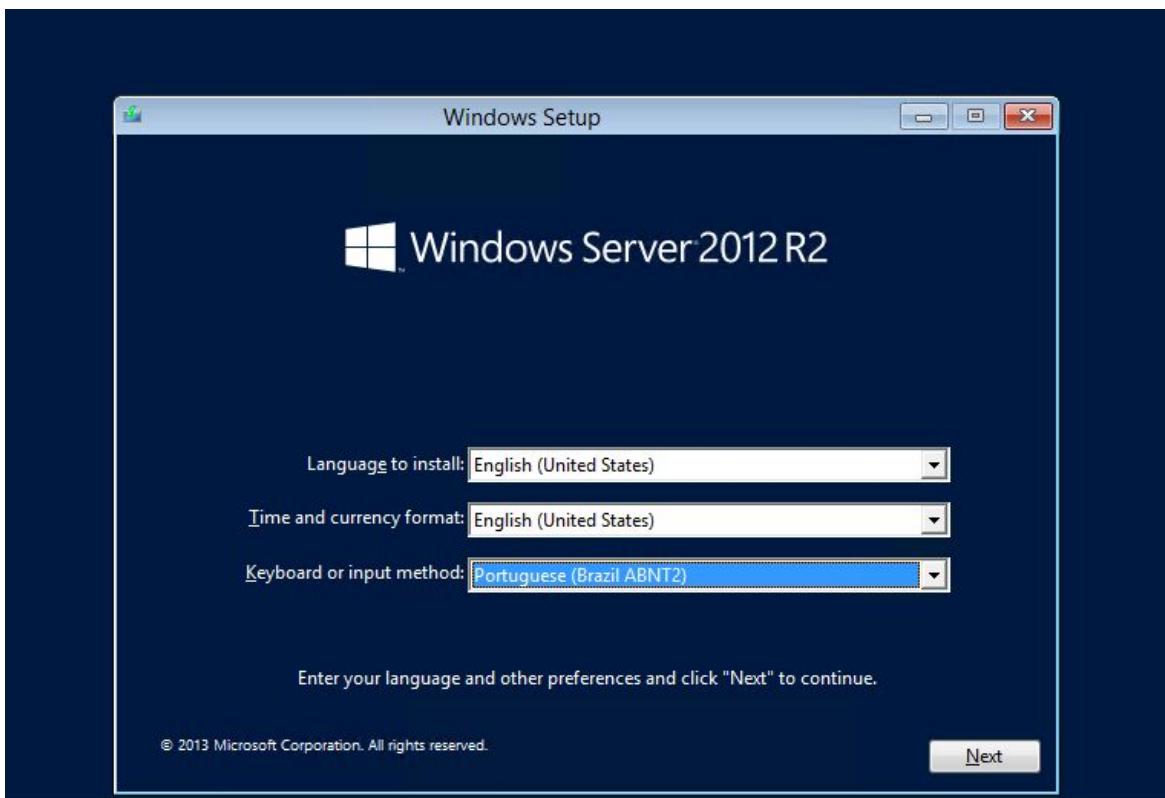




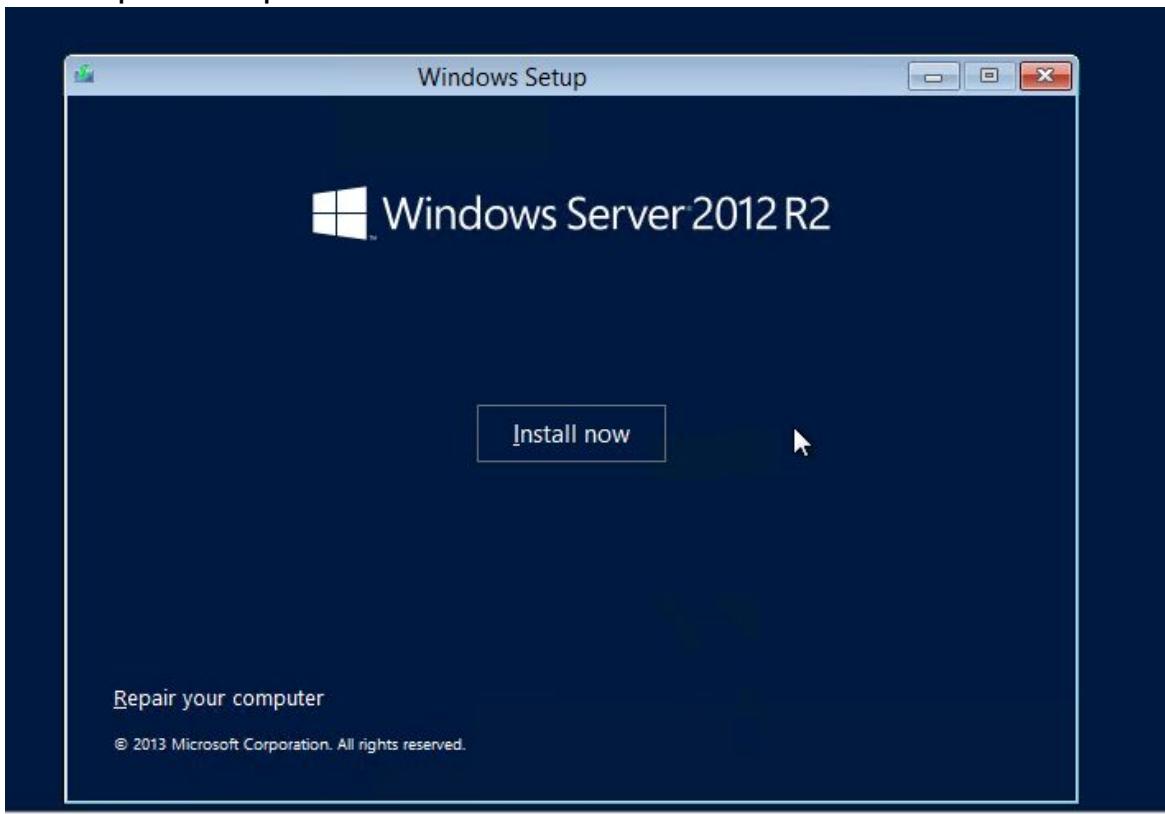
Nessa terceira etapa, subiremos um vCenter dentro de um windows server em cima do nosso ESXi, para que faça a gestão do ambiente.

Nessa primeira parte faremos a instalação do windows server, que no caso será o windows server 2012 R2

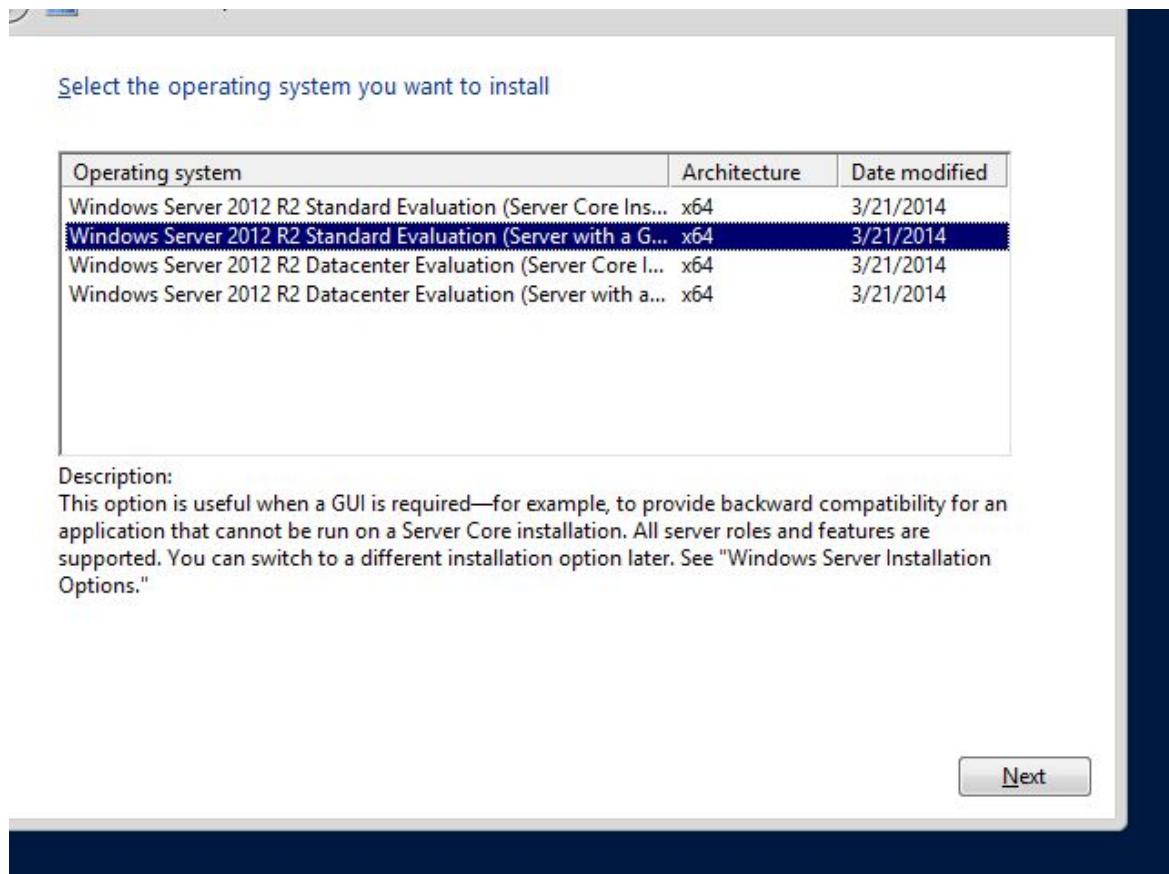
No print abaixo selecionamos o layout de teclado



Nessa parte só pressionar o install



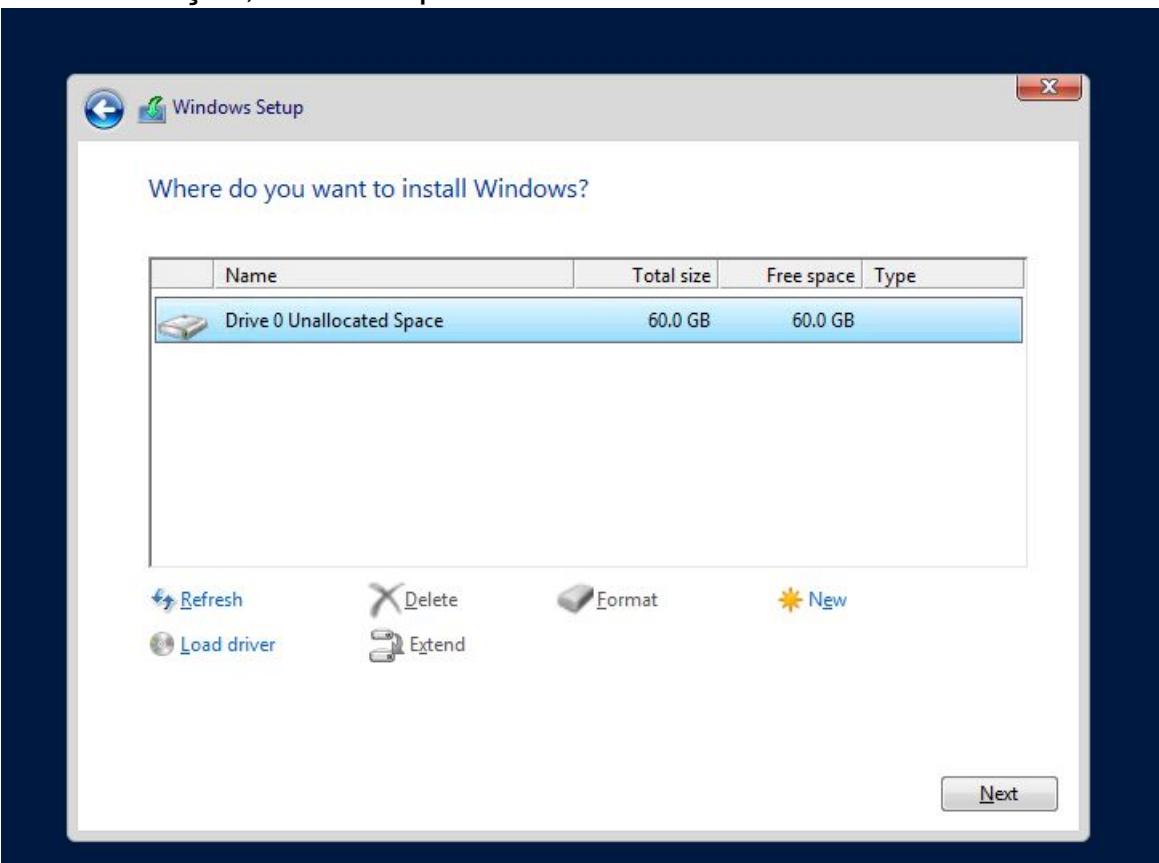
Nessa parte selecionamos qual windows iremos instalar, selecionamos o standard com a GUI, para termos a experiência com a interface gráfica



Nessa parte aceitamos o termo de licença e pressionamos next.



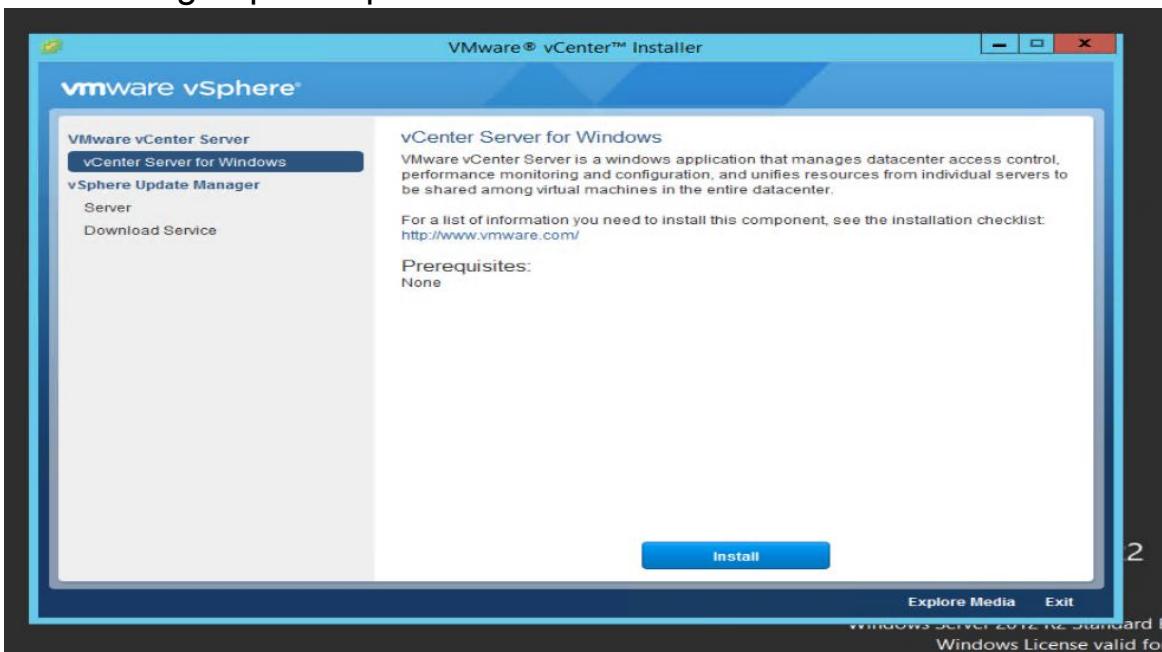
Selecionamos a unidade de disco para instalar, não precisa de nenhuma ação, somente pressionar next.



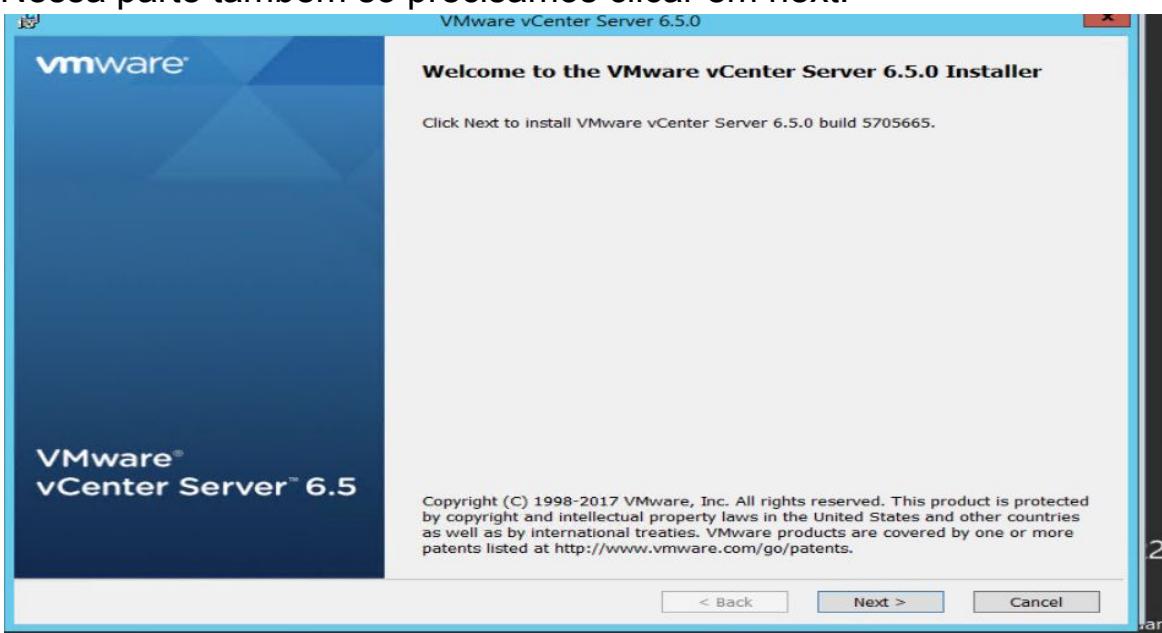
Depois que completar a instalação somente criar uma senha para o administrator do windows e seguir com a instalação do vCenter.

Após você selecionar a imagem iso do vCenter e colocar no VMware workstation no disco de boot, ele aparecerá no windows explorer, onde somente precisa clicar 2 vezes e fazer a instalação do vCenter que será documentada abaixo.

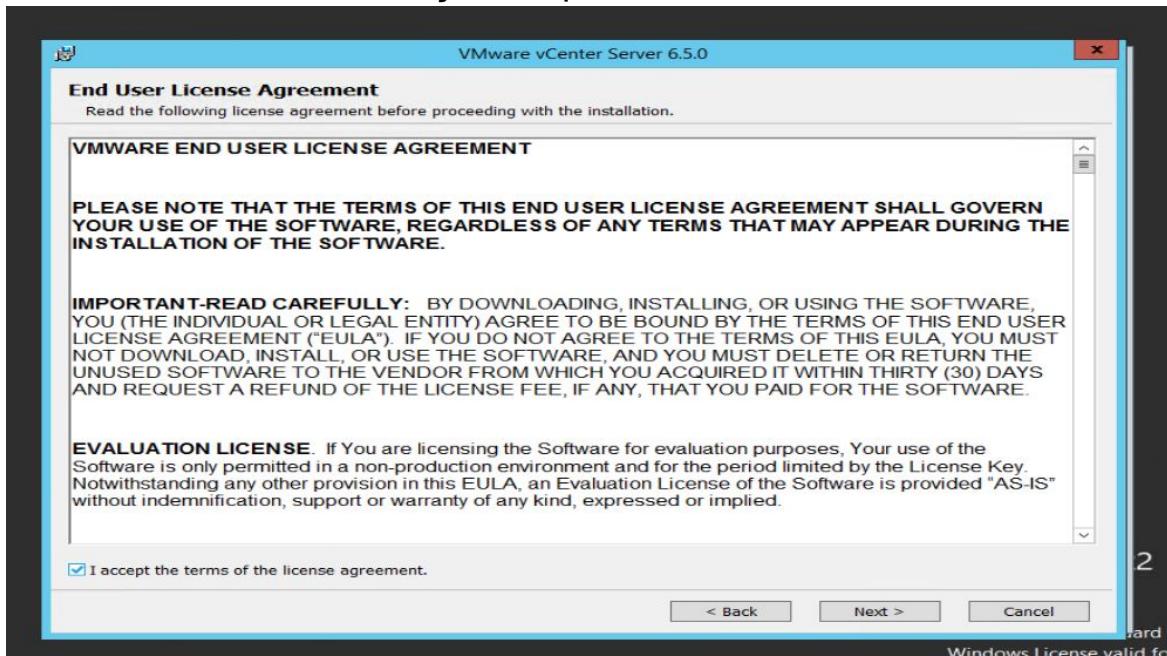
Na primeira tela não precisa de ação nenhuma apenas selecionar install e seguir para a próxima tela.



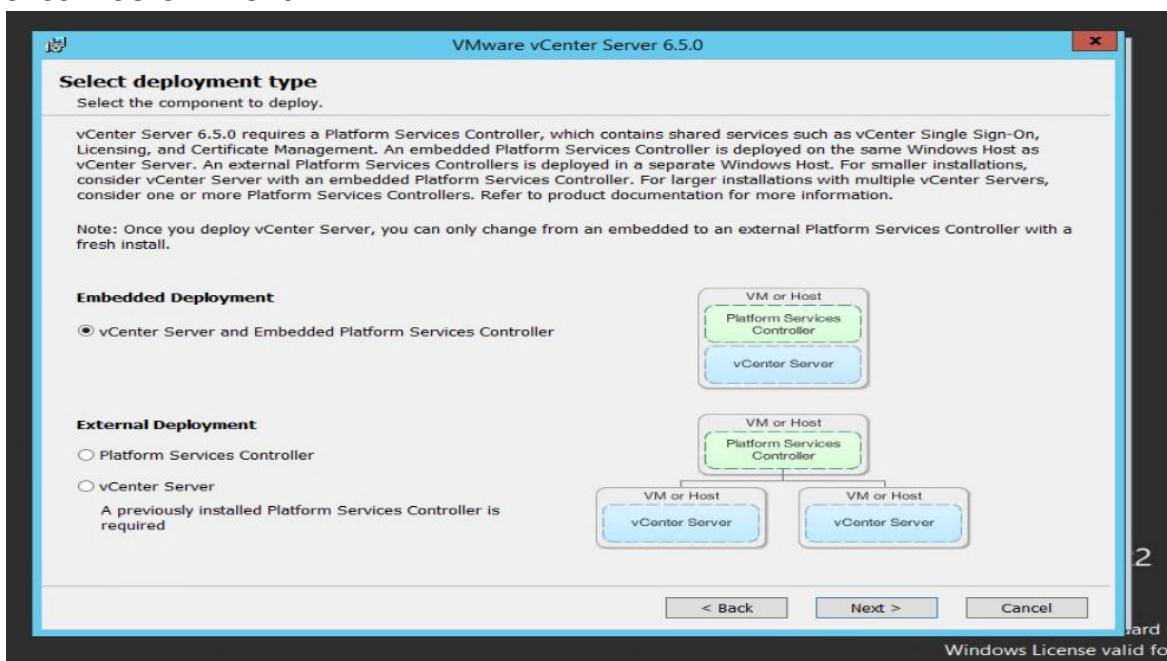
Nessa parte também só precisamos clicar em next.



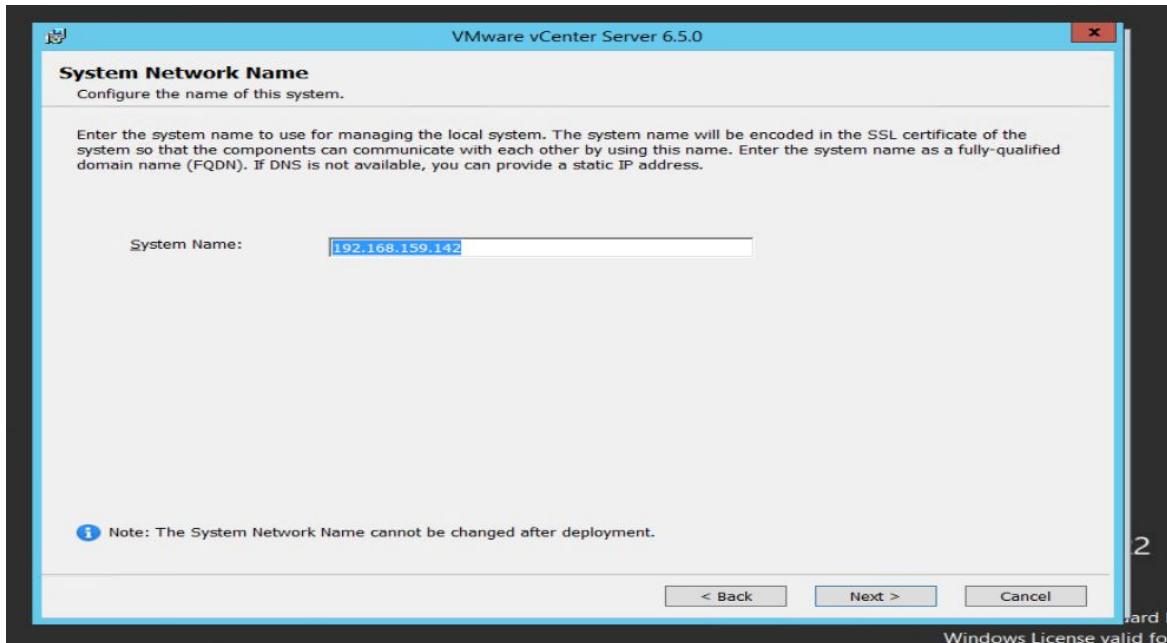
Aceita os termos da licença e clique em next.



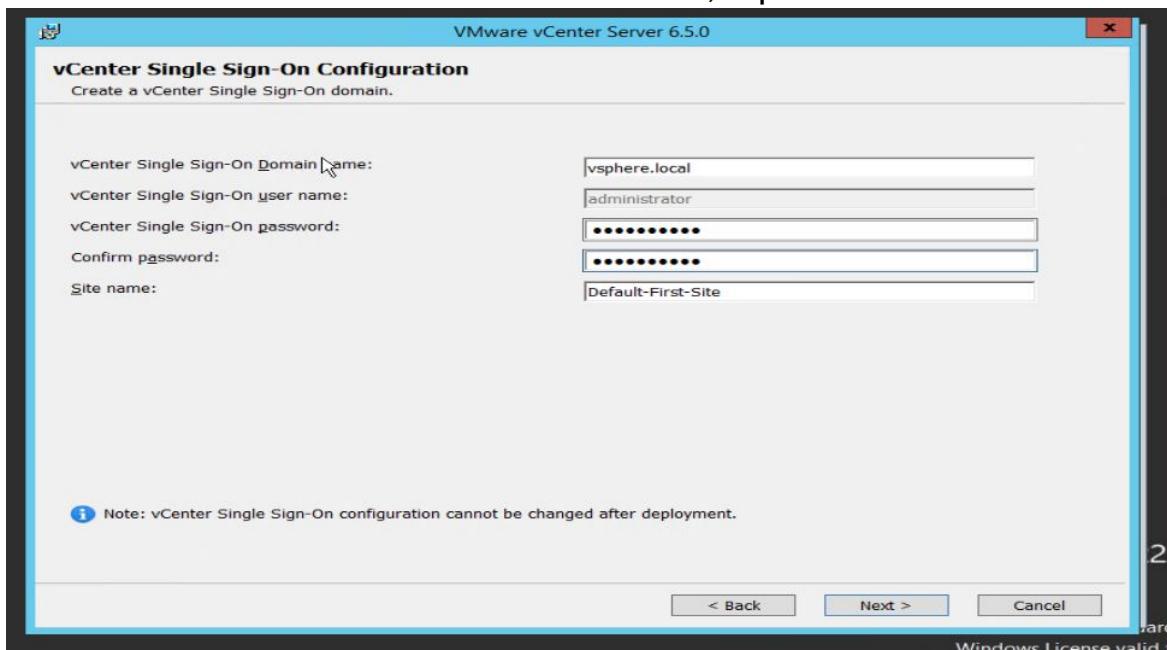
Nessa parte selecionamos a opção Embedded Deployment e clicamos em next.



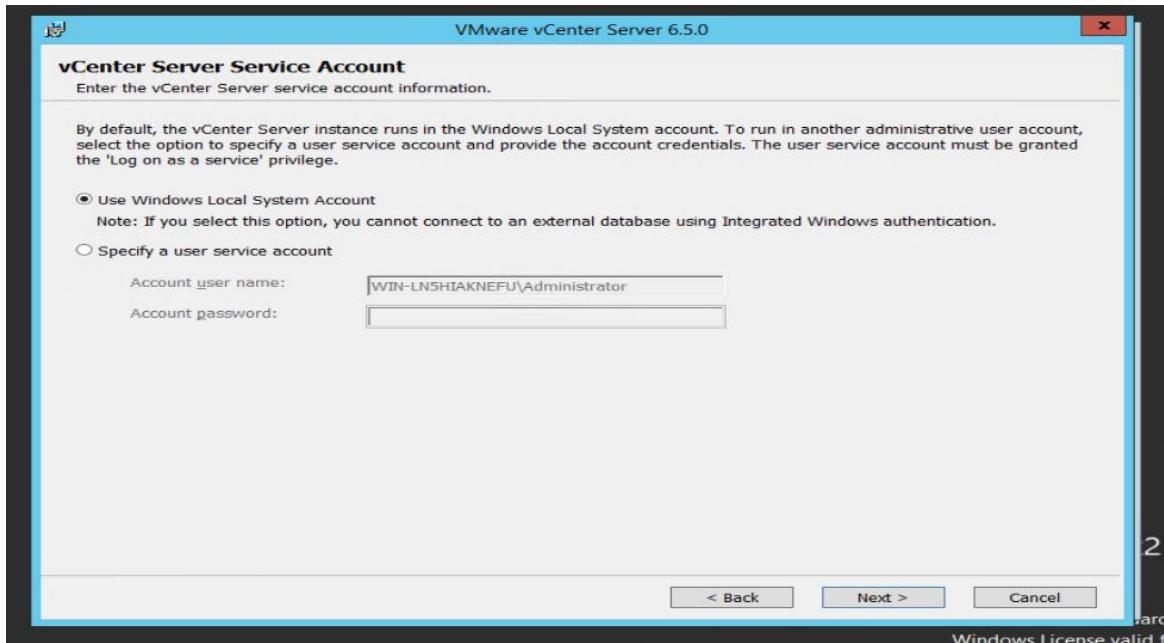
Nessa parte colocamos o ip da máquina windows na opção System Name e clicamos em next.



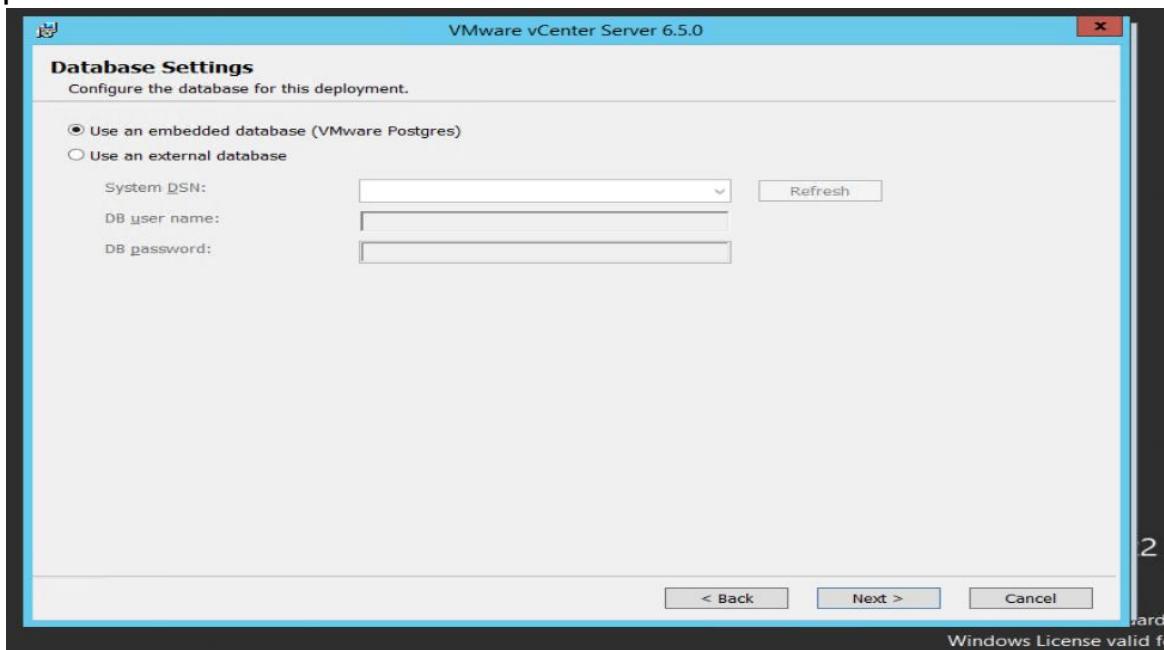
Criamos uma senha forte para o vCenter, com caracteres especiais, números e letras maiúsculas e minúsculas, e pressionar next.



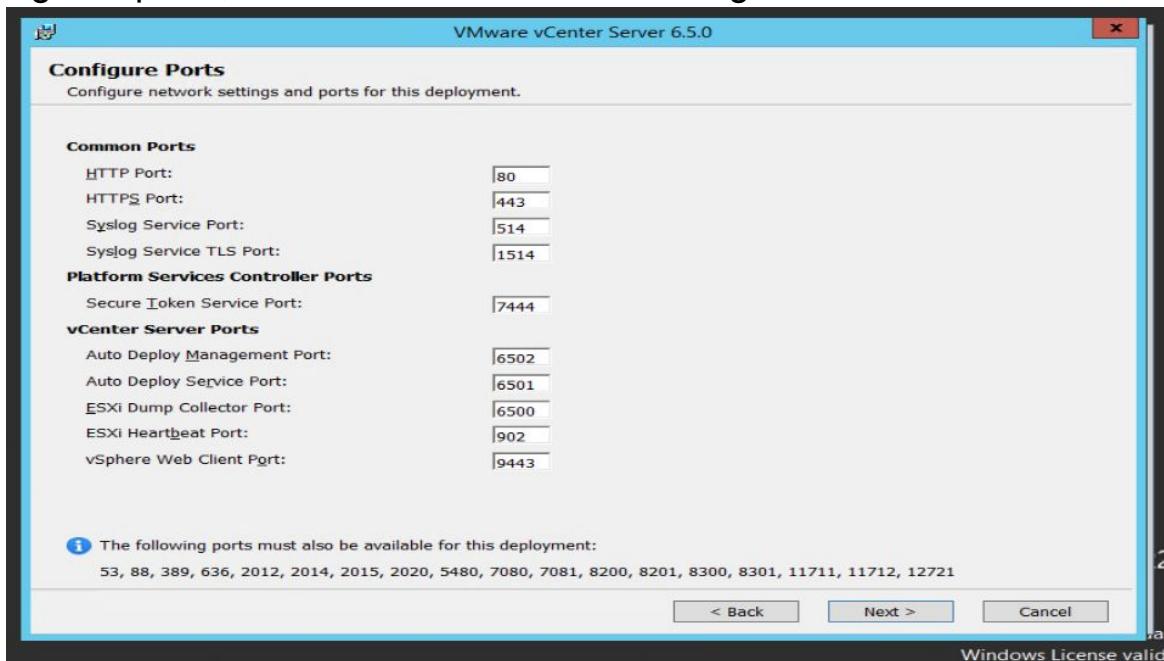
Nessa parte selecionamos a opção de use Windows Local System Account e clicamos em next.



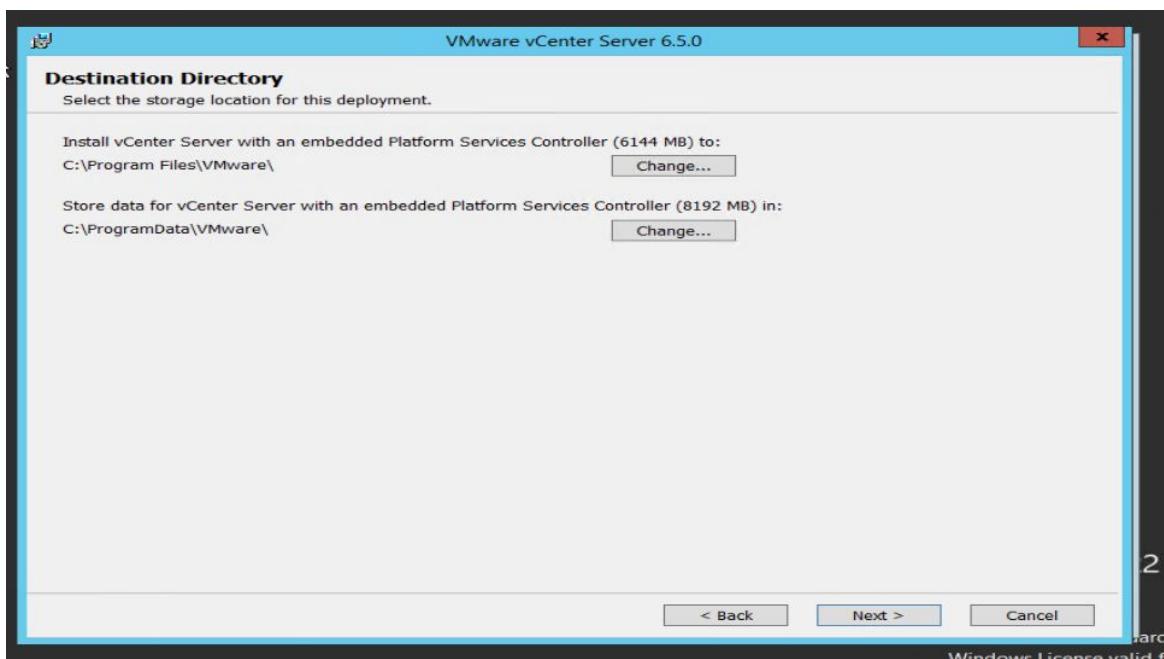
Nessa parte selecionamos a opção Use an Embedded database e pressionamos em next.



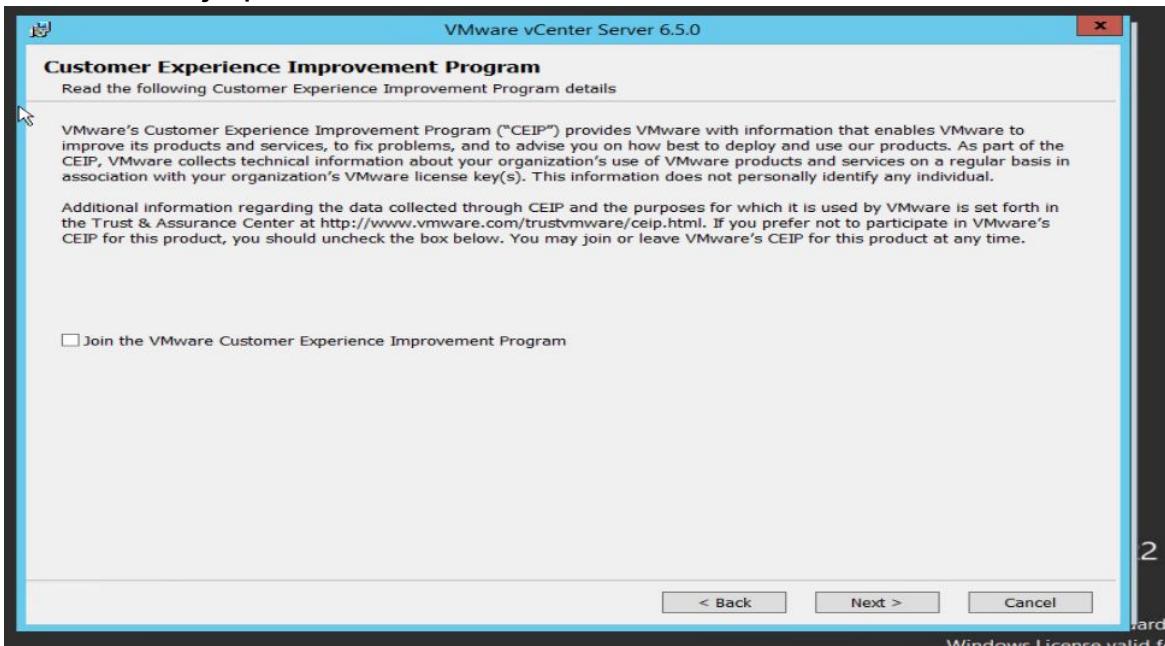
Nessa parte não precisamos tomar nenhum ação, usaremos as configurações já impostas pelo vCenter, se for o caso pode alterar alguma porta, se for necessidade do seu negócio.



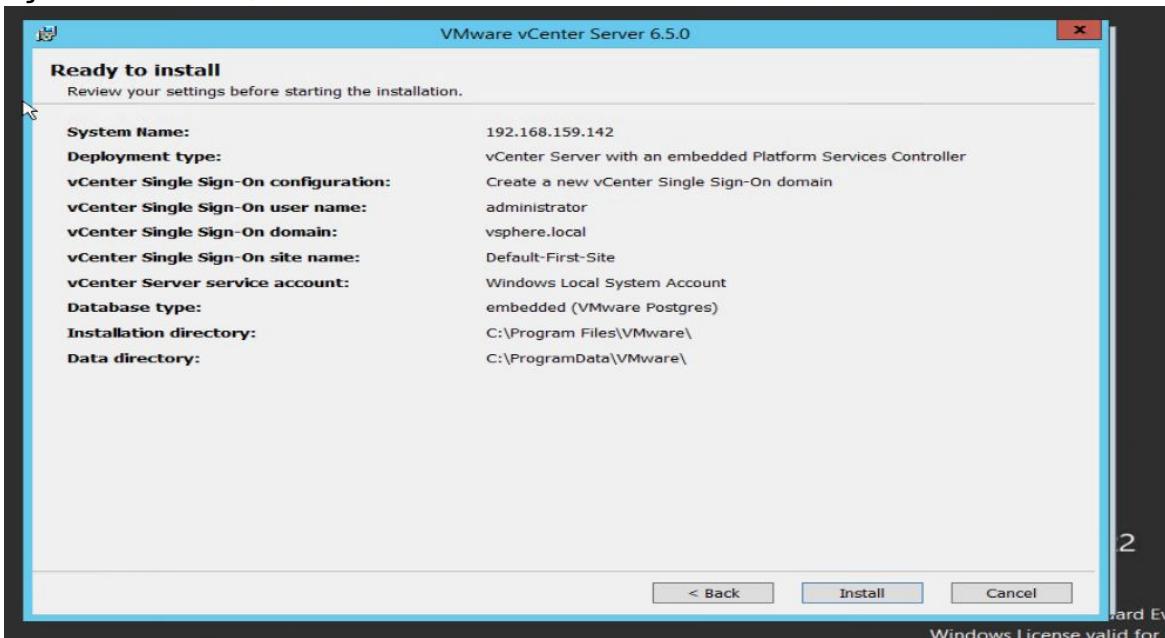
Também usaremos os caminhos já configurados, se no seu cenário for diferente só pressionar em change e direcionar para o diretório escolhido.



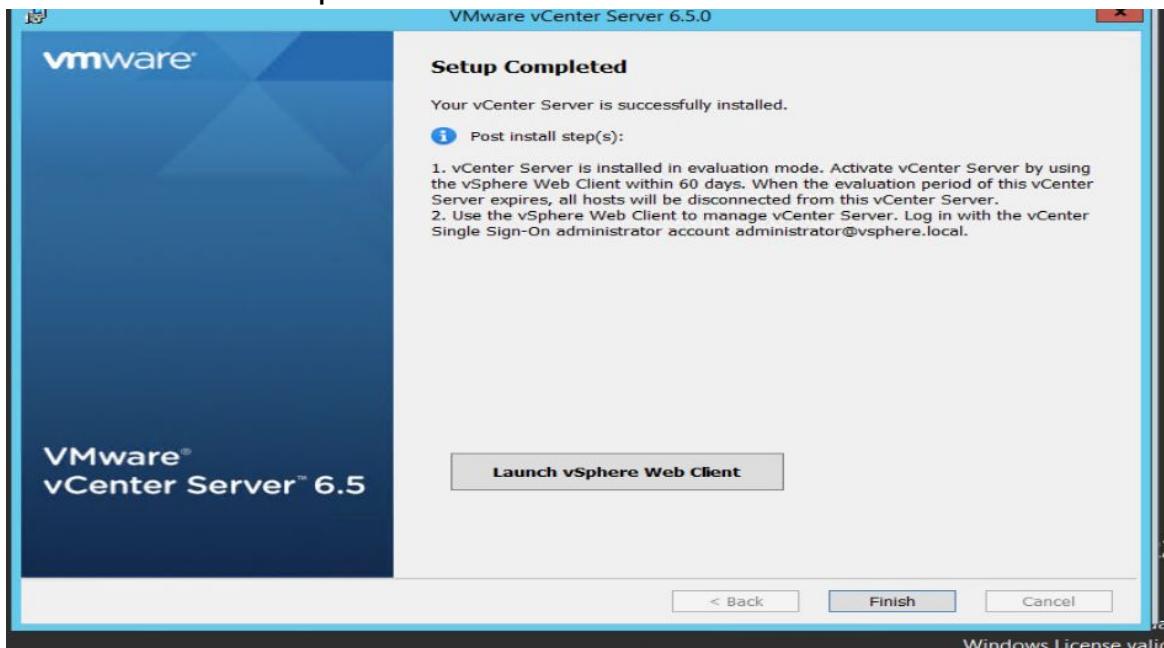
Nessa eu desmarco a opção de Join The VMware Customer Experience Improvement Program, somente pelo fato de não querer participar do programa, não faz diferença alguma na aplicação, se for do seu desejo pode deixar marcado.



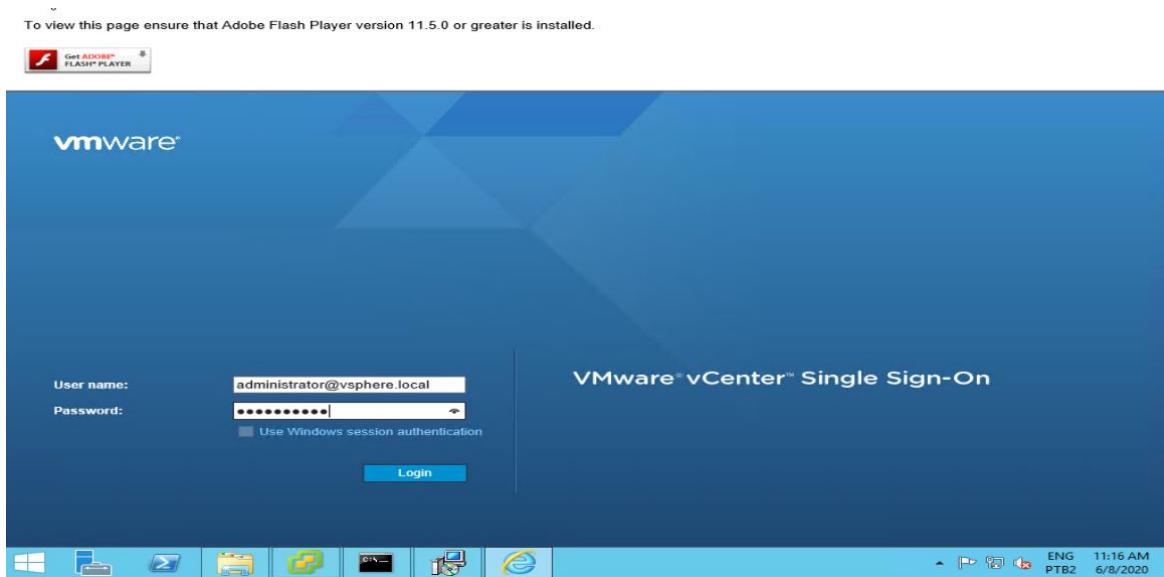
Nessa parte temos o review da instalação, também não precisa de ação nenhuma, só clicar em Install.



Essa é a tela que informa que tudo ocorreu bem, clicamos em Launch vSphere Web Client para acessar a web ou finish se for acessar a dashboard Web depois.



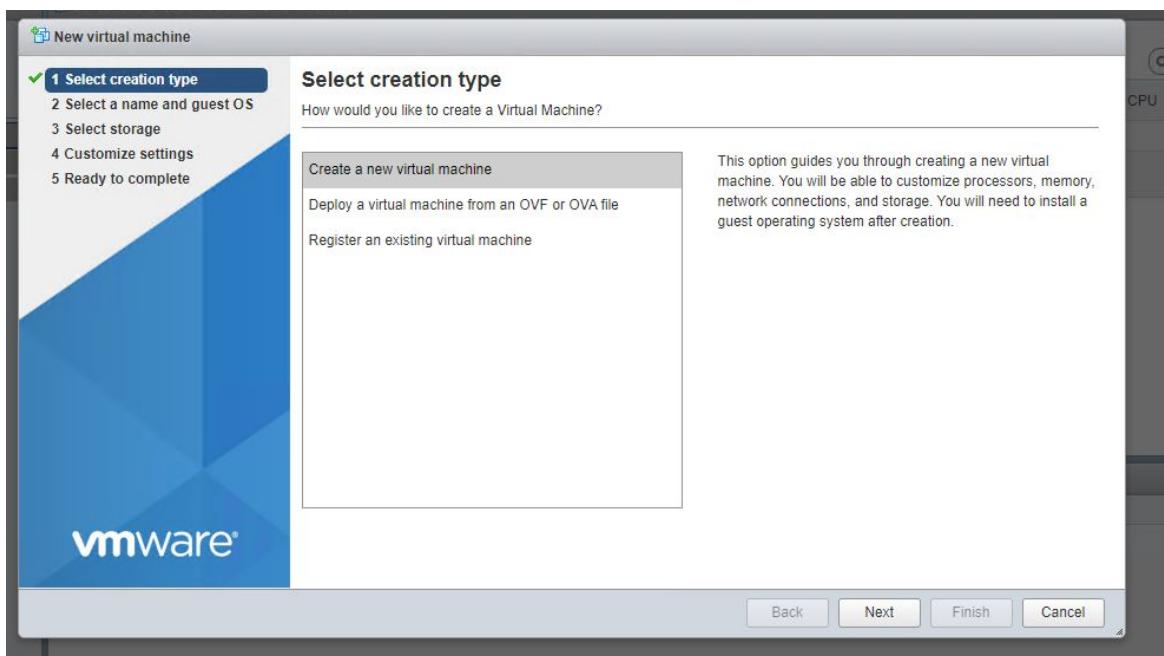
Nessa parte faremos o login para entrarmos no vCenter e termos a gestão dos ESXis entregues a ele, no caso não consegui logar pois não tinha o flash.



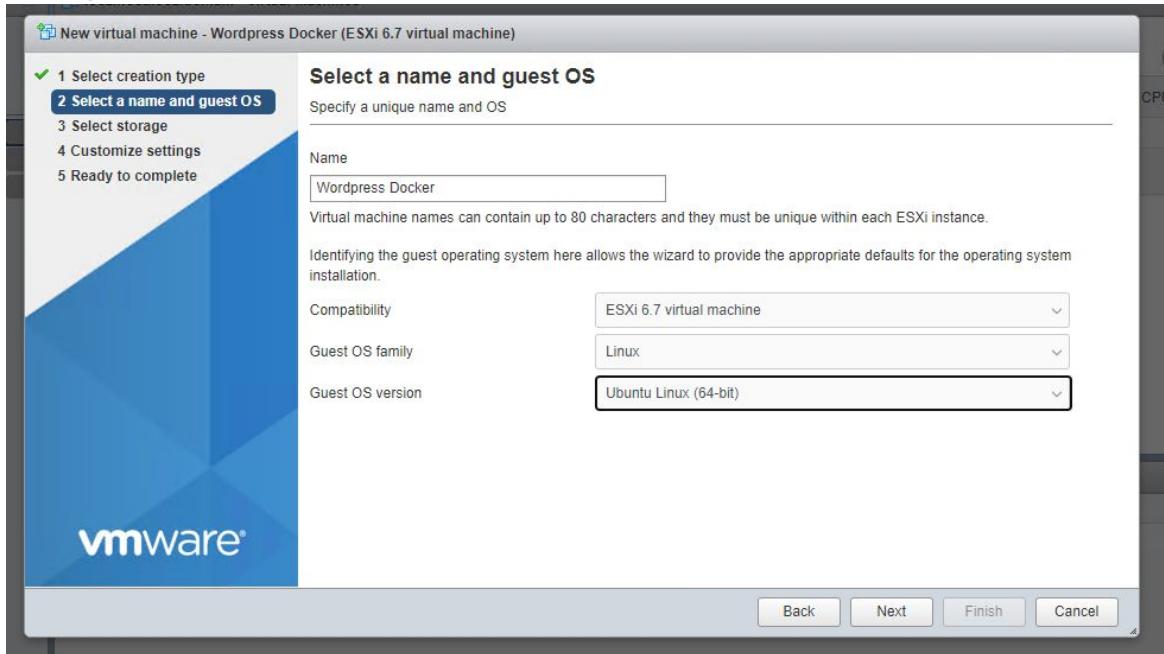
Na última etapa temos o passo a passo da instalação do wordpress via container e automatização ansible em cima do ubuntu, aplicação escolhida para esse projeto, será instalado em cima dos nossos servidores ESXi, cada servidor terá 2 VMs idênticas para termos maior redundância no ambiente.

Nos primeiros passos iremos criar a VM no ESXi e entrar na console para fazermos a instalação.

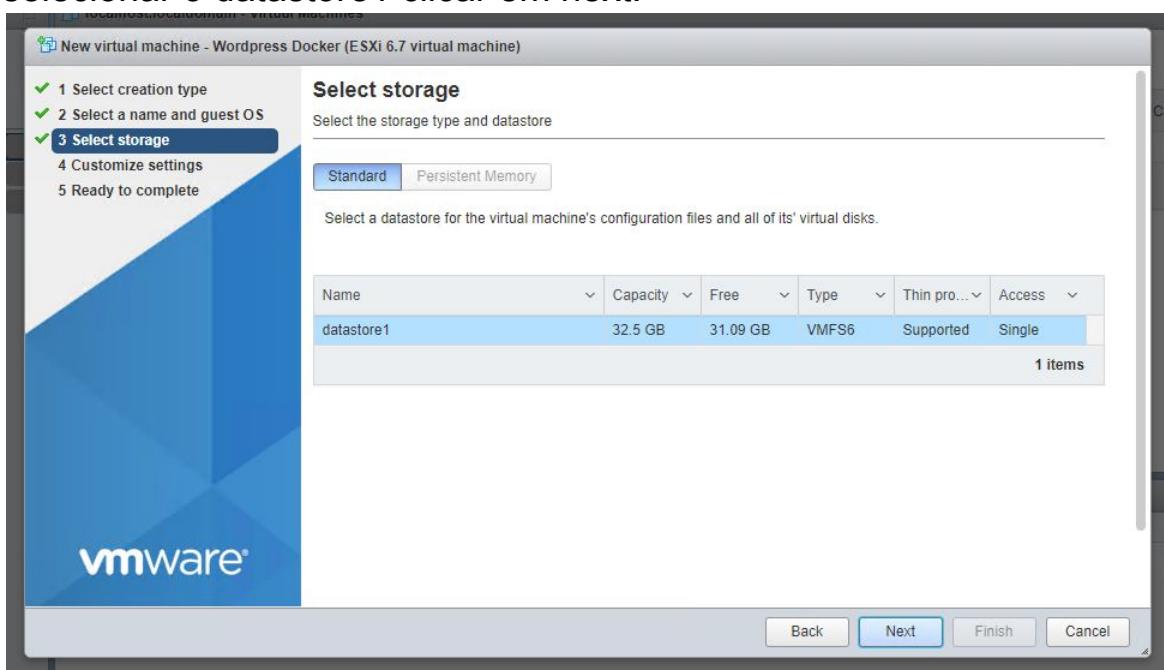
Primeiro passo é a criação da VM no ESXi, só clicar em create a new VM, e entrará nessa tela abaixo, onde selecionamos a opção Create a new Virtual Machine e clicamos em next.



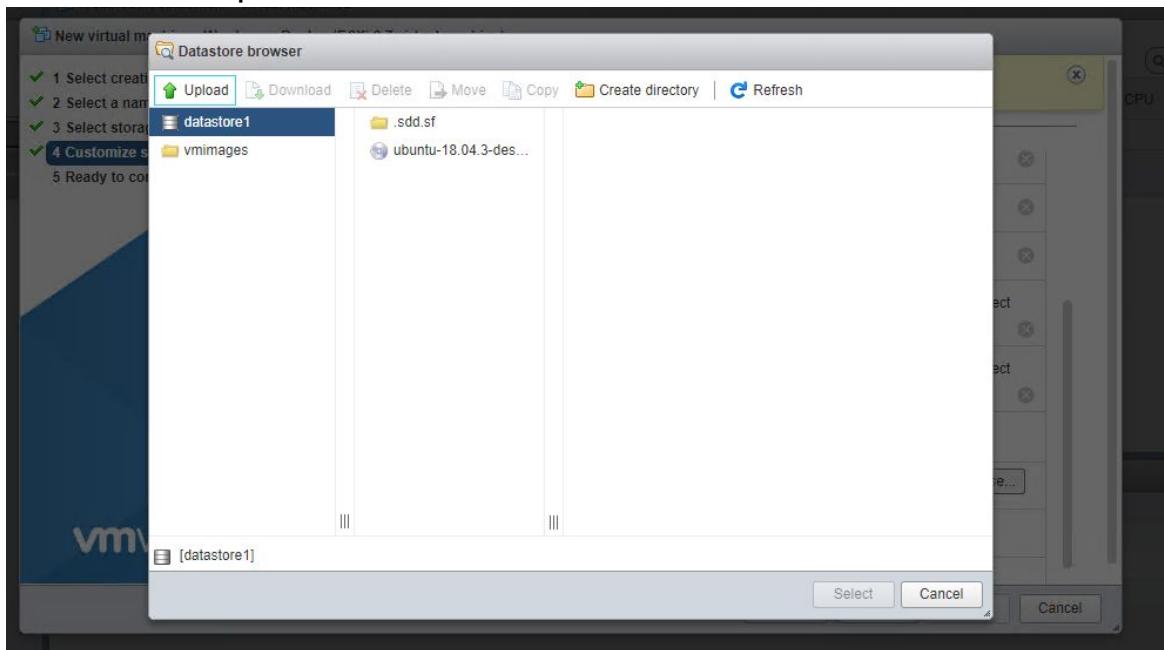
Nessa parte nós entraremos com o nome da VM, o tipo do sistema operacional e a versão dele, o resto é só deixar como já vem pré configurado, após isso, clicar em next.



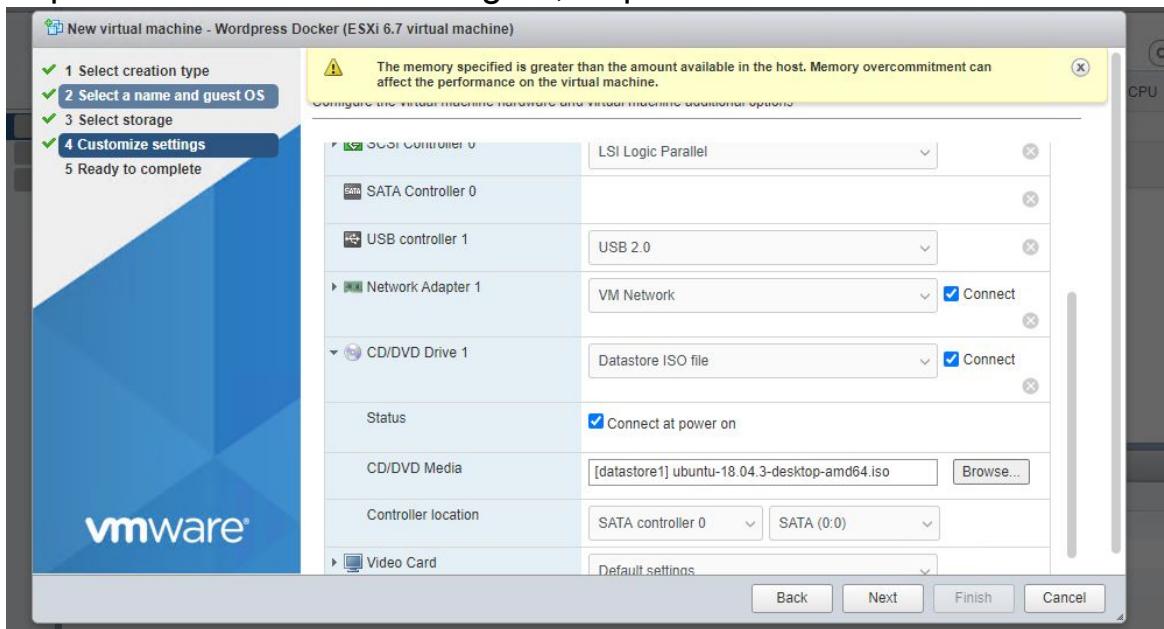
Nessa parte selecionamos o datastore onde queremos instalar a nossa VM, no nosso caso instalaremos no datastore1, após selecionar o datastore1 clicar em next.



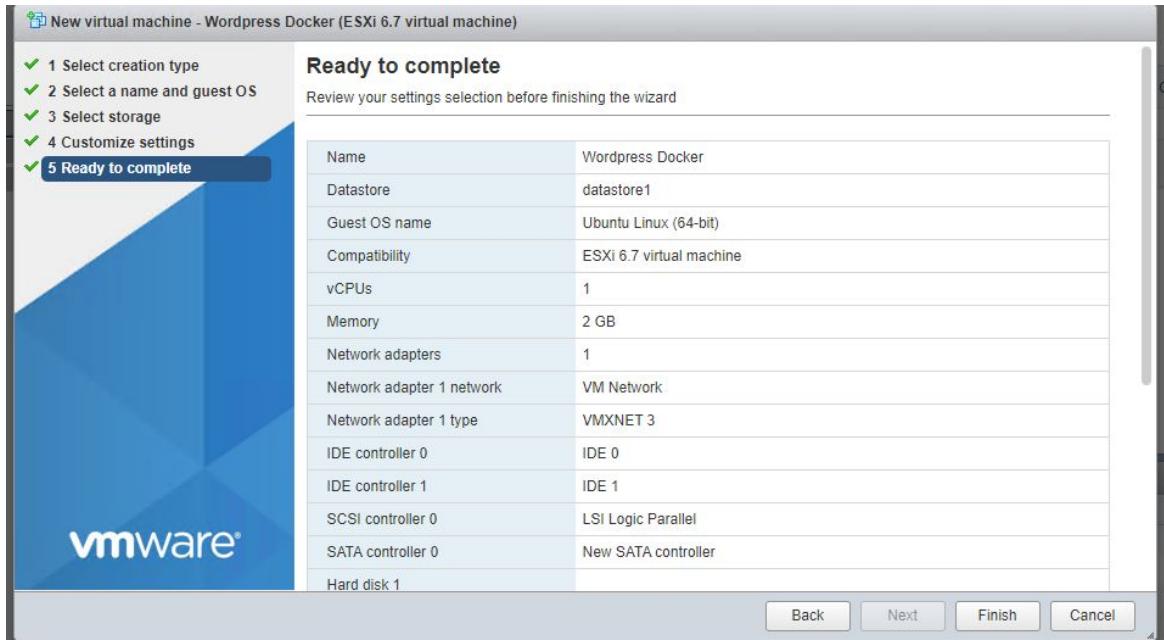
Nessa parte, em customize services, entraremos no DVD de boot da VM e faremos o upload da iso que será utilizada, no nosso caso o ubuntu desktop 18.04.



Depois de selecionado a imagem, clique em next.



Nessa parte temos o review da criação da VM, somente clicar em Finish.



Depois da VM criada é só selecionar ela e clicar em power on.

Depois de clicar em power on e ela iniciar, clicamos em console para começarmos a trabalhar na máquina.

Após clicarmos em console e entrarmos na tela da VM, só esperar o sistema operacional ser instalado e entrar com seu usuário e senha, pré configurado no VMware.



Depois, entrar no terminal do LINUX para fazermos a instalação dos componentes, primeira instalação será do net-tools, para termos visualização do nosso ip

```
analista@ubuntu:~$ sudo apt install net-tools
[sudo] password for analista:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 334 not upgraded.
Need to get 194 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 net-tools amd64 1.6
0+git20161116.90da8a0-1ubuntu1 [194 kB]
Fetched 194 kB in 1s (171 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 126435 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20161116.90da8a0-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Setting up net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
analista@ubuntu:~$
```

Depois faremos a instalação do serviço ssh com o comando
sudo apt-get install openssh-server

```
Creating config file /etc/ssh/sshd_config with new version
Creating SSH2 RSA key; this may take some time ...
2048 SHA256:az9q8rhVx0wtukDSmK52KCI4uLuCyzaZItWfElkFikI root@ubuntu (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:R1eqAgH3CM9sLocZm4lfyC34//hQCFewFToJIVm4Vs0 root@ubuntu (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:iPmqfPrAJ/9JZgAMcj1Q10FnuuZTCiTKitAu2Bmfoyo root@ubuntu (ED25519)
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/
systemd/system/ssh.service.
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.24) ...
Processing triggers for ufw (0.36-0ubuntu0.18.0.4.1) ...
analista@ubuntu:~$
```

Nessa parte criaremos a chave SSH com o comando abaixo.

```
analista@ubuntu:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/analista/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/analista/.ssh/id_rsa.
Your public key has been saved in /home/analista/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:MBSFJVq6pbDauOf0Y5Nei2KaKGjnKf+qn+gOfiNpk4c analista@ubuntu
The key's randomart image is:
+---[RSA 2048]----+
|   ==o
|   =..
|   . o +
|   o + o
|   . o S
|   +
|= * ..
| *E=*Bo .
| @@&@B+.
+---[SHA256]-----+
analista@ubuntu:~$
```

Com o comando abaixo nós sinalizaremos ao servidor que a chave libera logins remotos

```
analista@ubuntu:~$ cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
analista@ubuntu:~$
```

A partir de agora começaremos a instalação do ansible

Primeiro etapa é adicionar o repositório do ansible no nosso sistema, com o comando descrito abaixo.

```
analista@ubuntu:~$ sudo apt-add-repository ppa:ansible/ansible
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications—automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Press [ENTER] to continue or Ctrl-c to cancel adding it.

Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:2 http://ppa.launchpad.net/ansible/ubuntu bionic InRelease [15.9 kB]
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://ppa.launchpad.net/ansible/ubuntu/bionic/main i386 Packages [692 B]
Get:7 http://ppa.launchpad.net/ansible/ubuntu/bionic/main amd64 Packages [692 B]
Get:8 http://ppa.launchpad.net/ansible/ubuntu/bionic/main Translation-en [472 B]
Fetched 270 kB in 2s (113 kB/s)
Reading package lists... Done
analista@ubuntu:~$
```

Nessa parte instalaremos o python com o comando: sudo apt-get install python.

```
Selecting previously unselected package python2.7-minimal.
(Reading database ... 129408 files and directories currently installed.)
Preparing to unpack .../python2.7-minimal_2.7.17-1~18.04ubuntu1_amd64.deb ...
Unpacking python2.7-minimal (2.7.17-1~18.04ubuntu1) ...
Selecting previously unselected package python-minimal.
Preparing to unpack .../python-minimal_2.7.15~rc1-1_amd64.deb ...
Unpacking python-minimal (2.7.15~rc1-1) ...
Selecting previously unselected package python2.7.
Preparing to unpack .../python2.7_2.7.17-1~18.04ubuntu1_amd64.deb ...
Unpacking python2.7 (2.7.17-1~18.04ubuntu1) ...
Selecting previously unselected package libpython-stdlib:amd64.
Preparing to unpack .../libpython-stdlib_2.7.15~rc1-1_amd64.deb ...
Unpacking libpython-stdlib:amd64 (2.7.15~rc1-1) ...
Setting up python2.7-minimal (2.7.17-1~18.04ubuntu1) ...
Linking and byte-compiling packages for runtime python2.7...
Setting up python-minimal (2.7.15~rc1-1) ...
Selecting previously unselected package python.
(Reading database ... 129465 files and directories currently installed.)
Preparing to unpack .../python_2.7.15~rc1-1_amd64.deb ...
Unpacking python (2.7.15~rc1-1) ...
Setting up python2.7 (2.7.17-1~18.04ubuntu1) ...
Setting up libpython-stdlib:amd64 (2.7.15~rc1-1) ...
Setting up python (2.7.15~rc1-1) ...
Processing triggers for desktop-file-utils (0.23-1ubuntu3.18.04.2) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for gnome-menus (3.13.3-11ubuntu1.1) ...
Processing triggers for mime-support (3.60ubuntu1) ...
analista@ubuntu:~$
```

Nessa parte iremos atualizar os nossos pacotes para termos acesso aos programas do repositório.

```
analista@ubuntu:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:3 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic InRelease
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metadata [43.7 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Metadata [49.2 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Get:9 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Metadata [306 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 DEP-11 Metadata [279 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 DEP-11 Metadata [2,464 B]
Get:12 http://us.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 DEP-11 Metadata [7,968 B]
Fetched 943 kB in 2s (400 kB/s)
Reading package lists... Done
analista@ubuntu:~$
```

Abaixo nós fizemos a instalação do pacote ansible com o comando:
sudo apt-get install ansible

```
Setting up python-idna (2.6-1) ...
Setting up python-yaml (3.12-1build2) ...
Setting up python-asn1crypto (0.24.0-1) ...
Setting up python-crypto (2.6.1-8ubuntu2) ...
Setting up python-pyasn1 (0.4.2-3) ...
Setting up python-pkg-resources (39.0.1-2) ...
Setting up python-markupsafe (1.0-1build1) ...
Setting up python-httplib2 (0.9.2+dfsg-1ubuntu0.1) ...
Setting up python-cffi-backend (1.11.5-1) ...
Setting up python-six (1.11.0-2) ...
Setting up sshpass (1.06-1) ...
Setting up python-enum34 (1.1.6-2) ...
Setting up python-ipaddress (1.0.17-1) ...
Setting up python-setuptools (39.0.1-2) ...
Setting up python-jinja2 (2.10-1ubuntu0.18.04.1) ...
Setting up python-cryptography (2.1.4-1ubuntu1.3) ...
Setting up python-paramiko (2.0.0-1ubuntu1.2) ...
Setting up ansible (2.9.9-1ppa-bionic) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
analista@ubuntu:~$
```

A partir de agora iremos configurar o ansible para montar o ambiente, com o comando abaixo nós criamos o arquivo de inventário do ansible, onde iremos atribuir um servidor a ele.

```
analista@ubuntu:~$ 
analista@ubuntu:~$ 
analista@ubuntu:~$ 
analista@ubuntu:~$ sudo touch /etc/ansible/hosts
analista@ubuntu:~$
```

Agora nós iremos atribuir o servidor nesse inventário, entramos nele usando o gedit, com o comando sudo gedit /etc/ansible/hosts e dentro do arquivo nós botaremos a informação descrita no print abaixo no final do arquivo

```
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern you can specify
# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no

# leading 0s:

## db-[99:101]-node.example.com

[wordpress]
127.0.0.1
```

Depois disso faremos um teste com um comando para ver se está tudo certo.

```
analista@ubuntu:~$ ansible wordpress -m ping -u analista
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 127.0.0.1 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A future Ansible release will default to using the
discovered platform python for this host. See https://docs.ansible.com/ansible/
2.9/reference_appendices/interpreter_discovery.html for more information. This
feature will be removed in version 2.12. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.
127.0.0.1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
analista@ubuntu:~$
```

Agora nós instalaremos alguns pacotes de suporte que usaremos posteriormente na instalação para assinatura digital e download de pacotes.

```
(Reading database ... 136324 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.6.12ubuntu0.1_all.deb ...
Unpacking apt-transport-https (1.6.12ubuntu0.1) ...
Selecting previously unselected package libcurl4:amd64.
Preparing to unpack .../libcurl4_7.58.0-2ubuntu3.8_amd64.deb ...
Unpacking libcurl4:amd64 (7.58.0-2ubuntu3.8) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.58.0-2ubuntu3.8_amd64.deb ...
Unpacking curl (7.58.0-2ubuntu3.8) ...
Setting up apt-transport-https (1.6.12ubuntu0.1) ...
Setting up libcurl4:amd64 (7.58.0-2ubuntu3.8) ...
Setting up curl (7.58.0-2ubuntu3.8) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
analista@ubuntu:~$ █
```

Agora nós rodaremos esse comando que traz a chave de assinatura digital do projeto docker e adiciona ao nosso gerenciador de pacotes.

```
analista@ubuntu:~$  
analista@ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | su  
do apt-key add -  
OK  
analista@ubuntu:~$
```

Esse comando exibe a identificação da chave que adicionamos.

```
analista@ubuntu:~$ sudo apt-key fingerprint 0EBFCD88  
Pub rsa4096 2017-02-22 [SCEA]  
    9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88  
uid [ unknown] Docker Release (CE deb) <docker@docker.com>  
sub rsa4096 2017-02-22 [S]  
analista@ubuntu:~$
```

Nesse comando nós adicionamos o repositório do projeto docker ao nosso sistema, assim possibilitando a instalação os softwares disponibilizados pelo projeto.

```
analista@ubuntu:~$ sudo add-apt-repository \  
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \  
> $(lsb_release -cs) \  
> stable"  
Get:1 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]  
Get:2 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [12  
.1 kB]  
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic InRelease  
Hit:4 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease  
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease  
Hit:6 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease  
Hit:7 http://ppa.launchpad.net/ansible/ubuntu bionic InRelease  
Fetched 76.5 kB in 2s (37.8 kB/s)  
Reading package lists... Done  
analista@ubuntu:~$ █
```

Este comando vai baixar a lista de pacotes disponíveis no nossos repositórios configurados

```
analista@ubuntu:~$ sudo apt-get update  
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease  
Hit:2 https://download.docker.com/linux/ubuntu bionic InRelease  
Hit:3 http://ppa.launchpad.net/ansible/ubuntu bionic InRelease  
Hit:4 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease  
Hit:5 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease  
Hit:6 http://security.ubuntu.com/ubuntu bionic-security InRelease  
Reading package lists... Done  
analista@ubuntu:~$ █
```

Nessa parte nós finalmente instalaremos o sistema docker, com o comando: sudo apt-get install docker-ce

```
Preparing to unpack .../6-liberror-perl_0.17025-1_all.deb ...  
Unpacking liberror-perl (0.17025-1) ...  
Selecting previously unselected package git-man.  
Preparing to unpack .../7-git-man_1%3a2.17.1-1ubuntu0.7_all.deb ...  
Unpacking git-man (1:2.17.1-1ubuntu0.7) ...  
Selecting previously unselected package git.  
Preparing to unpack .../8-git_1%3a2.17.1-1ubuntu0.7_amd64.deb ...  
Unpacking git (1:2.17.1-1ubuntu0.7) ...  
Setting up aufs-tools (1:4.9+20170918-1ubuntu1) ...  
Setting up git-man (1:2.17.1-1ubuntu0.7) ...  
Setting up containerd.io (1.2.13-2) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service →  
/lib/systemd/system/containerd.service.  
Setting up liberror-perl (0.17025-1) ...  
Setting up cgroupsfs-mount (1.4) ...  
Setting up docker-ce-cli (5:19.03.11-0~ubuntu-bionic) ...  
Setting up pigz (2.4-1) ...  
Setting up git (1:2.17.1-1ubuntu0.7) ...  
Setting up docker-ce (5:19.03.11-0~ubuntu-bionic) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/  
systemd/system/docker.service.  
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/  
systemd/system/docker.socket.  
Processing triggers for libc-bin (2.27-3ubuntu1) ...  
Processing triggers for systemd (237-3ubuntu10.41) ...  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
Processing triggers for ureadahead (0.100.0-21) ...  
analista@ubuntu:~$ █
```

□ Agora nós automatizaremos todo o instanciamento do container wordpress e mySQL usando o ansible.

Primeiro de tudo, criar um diretório no LINUX para organizarmos o projeto, depois entrar no diretório e criar um arquivo texto para botarmos os comando YML dentro para rodar o playbook ansible, assim como está no print abaixo:

```
-->
- hosts: wordpress
  remote_user: analista
  become: yes
  tasks:
    - name: "Executa o container MySQL"
      docker_container:
        name: banco
        image: mysql:5.6
        env:
          MYSQL_ROOT_PASSWORD: senha123
    - name: "Executa o container WordPress"
      docker_container:
        name: wordpress
        image: wordpress
        links:
          - "banco:mysql"
      ports:
        - "80:80"
```

Entramos nesse diretório que foi criado.

```
analista@ubuntu:~$ cd wordpress-container/
analista@ubuntu:~/wordpress-container$ █
```

Em seguida rodamos o playbook que foi escrito anteriormente, dessa forma nós subimos todo nosso ambiente wordpress, que foi montado na “mão” anteriormente com apenas um comando.

ansible-playbook wordpress.yml

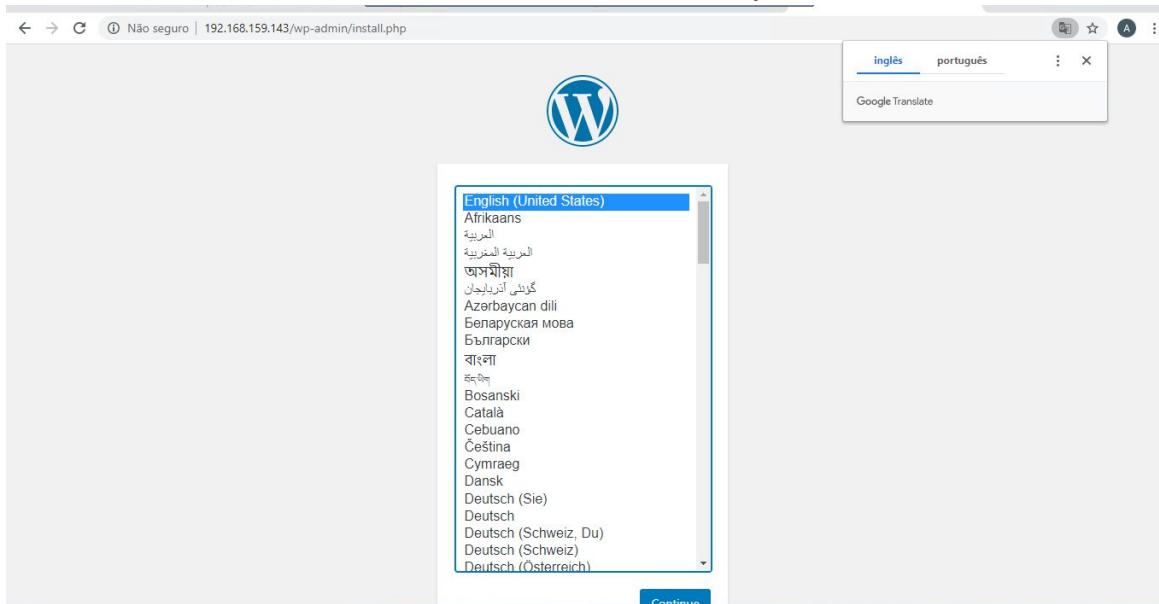
```
TASK [Gathering Facts] ****
*
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 127.0.0.1 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A future Ansible release will default to using the
discovered platform python for this host. See https://docs.ansible.com/ansible/
2.9/reference_appendices/interpreter_discovery.html for more information. This
feature will be removed in version 2.12. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.
ok: [127.0.0.1]

TASK [Executa o container MySQL] ****
*
ok: [127.0.0.1]

TASK [Executa o container WordPress] ****
*
ok: [127.0.0.1]

PLAY RECAP ****
*
127.0.0.1      : ok=3    changed=0    unreachable=0    failed=0
skipped=0      rescued=0   ignored=0
```

Depois desses passo, você pega o IP da sua máquina e faz um teste tentando acessar ele na WEB, como feito no print abaixo:



Criaremos rapidamente uma instalação do wordpress.

Olá

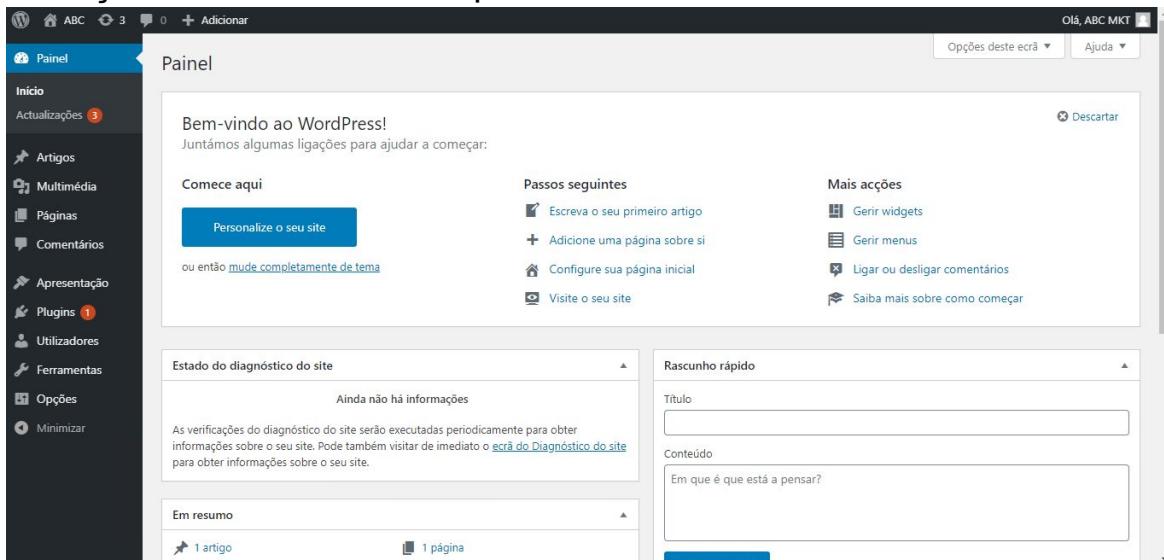
Bem-vindo ao famoso processo de instalação de cinco minutos do WordPress! Preencha a informação abaixo e estará pronto para usar a mais extensível e poderosa plataforma de publicação pessoal do mundo.

Informação necessária

Por favor forneça as seguintes informações. Não se preocupe, pode sempre alterar estas configurações mais tarde.

Título do site	ABC
Nome de utilizador	ABC MKT
Senha	ku0mh8R@#91T*pd0Dw
O seu endereço de email	andre.esilva@al.infnet.edu.br
Visibilidade nos motores de pesquisa	<input type="checkbox"/> Desencorajar os motores de pesquisa de indexarem este site Cabe aos motores de pesquisa honrar este pedido.

Depois de fazer os passos para instalar, chegamos nesse Painel, onde o time de desenvolvimento escalado no projeto terá acesso para a criação do WEBSITE da empresa ABC.



Capítulo 4

4.1 Conclusão

O projeto foi concluído dentro do prazo previamente planejado, como já era esperado, pelo fato de ser uma aplicação mais simples e com muito conteúdo na internet para o caso de precisar estudar possíveis erros na aplicação, o resto da nossa infraestrutura também era de uma certa forma básica, pois foi criada especialmente para esse projeto, então foram instalações limpas, sem ter outros softwares rodando no ambiente, uma vez que até a parte física do ambiente era nova.

O prazo indicado para o fim do projeto, foi muito bem calculado e de uma certa forma até lento, foram 15 dias de projeto, 15 contando com tudo que era preciso em mãos, e se tratava de um ambiente simples, claro que todos os passos foram feitos 2 vezes, pois se tratavam de 2 máquinas físicas, mas no geral eram instalações fáceis de se fazer.

Os recursos que foram disponibilizados no projeto, falo de recursos computacionais, também foram estudos e superestimados pelo projeto, nos dando uma grande folga para possíveis erros de cálculo, então foi tudo muito bem executado sem maiores problemas,

levando em consideração os recursos computacionais do projeto.

Todas as instalações foram feitas sem maiores problemas, o maior problema foram alguns erros na instalação do docker, no final do projeto, mas foram problemas muito baixos que foram rapidamente solucionados, assim não interferindo principalmente na qualidade e tempo do projeto.

4.2 - Melhorias futuras

Nesse espaço do projeto, será relatado alguns pontos de aprendizado que podem ser úteis para os próximos projetos.

1- Sempre quando for instalar máquinas virtuais, fazer snapshots de ponto cruciais da máquina, isso deixou de ser feito na máquina ubuntu e teve que ser reinstalado do zero, sorte que foi no início da instalação.

2 - Sempre superestimar o tempo do projeto, mesmo que seja pouco tempo a mais, isso dá mais margem para possíveis erros.

4.3 - Passos do commit para o Github

Primeiro passo é criar o novo repositório.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner

Repository name *



/ Projeto-de-Bloco-Final



Great repository names are short and memorable. Need inspiration? How about [upgraded-barnacle](#)?

Description (optional)

Repositório do Projeto de Bloco

 Public

Anyone can see this repository. You choose who can commit.

 Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾

Add a license: None ▾



Create repository

Nessa segunda tela, serve somente para pegar esse link do repositório e apontar ele no Linux para o commit.

Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/andreinfnet/Projeto-de-Bloco-Final.git>

Get started by creating a new file or [uploading an existing file](#). We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# Projeto-de-Bloco-Final" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/andreinfnet/Projeto-de-Bloco-Final.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/andreinfnet/Projeto-de-Bloco-Final.git
git push -u origin master
```

Nessas telas abaixo tem os comandos no passo a passo para a execução do commit.

```
analista@ubuntu:~/wordpress-PB$ git init
Initialized empty Git repository in /home/analista/wordpress-PB/.git/
analista@ubuntu:~/wordpress-PB$ git add *
analista@ubuntu:~/wordpress-PB$ analista@ubuntu:~/wordpress-PB$ setxkbmap -model abnt2 -layout br -variant abnt
2
analista@ubuntu:~/wordpress-PB$ analista@ubuntu:~/wordpress-PB$ analista@ubuntu:~/wordpress-PB$ git commit -m "commit do codigo"
[master (root-commit) 5a7a915] commit do codigo
 1 file changed, 19 insertions(+)
  create mode 100644 wordpress-PB
analista@ubuntu:~/wordpress-PB$ git remote add origin https://github.com/andreinfnet/Projeto-de-Bloco-Final.git
analista@ubuntu:~/wordpress-PB$ git push -u origin master
Username for 'https://github.com': andreinfnet
Password for 'https://andreinfnet@github.com':
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 435 bytes | 108.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/andreinfnet/Projeto-de-Bloco-Final.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
analista@ubuntu:~/wordpress-PB$
```

```
analista@ubuntu:~/wordpress-PB$ analista@ubuntu:~/wordpress-PB$ analista@ubuntu:~/wordpress-PB$ analista@ubuntu:~/wordpress-PB$ analista@ubuntu:~/wordpress-PB$ analista@ubuntu:~/wordpress-PB$ analista@ubuntu:~/wordpress-PB$ analista@ubuntu:~/wordpress-PB$ git clone https://github.com/andreinfnet/Projeto-de-Bloco-Final.git
Cloning into 'Projeto-de-Bloco-Final'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
analista@ubuntu:~/wordpress-PB$
```

Na tela abaixo temos a confirmação do sucesso do commit.

The screenshot shows a GitHub repository page for 'andreinfnet / Projeto-de-Bloco-Final'. The repository has 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. The latest commit was made by 'andreinfnet' 7 days ago, with the message 'commit do codigo'. A green button at the bottom right encourages adding a README.

andreinfnet / Projeto-de-Bloco-Final

Code Issues Pull requests Marketplace Explore

Unwatch 1 Star 0 Fork 0

Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

Repositório do Projeto de Bloco

Manage topics

-o 1 commit 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

andreinfnet commit do codigo Latest commit 5a7a915 7 days ago

wordpress-PB commit do codigo 7 days ago

Add a README

4.4 - Link do github com a documentação do projeto

<https://github.com/andreinfnet/Projeto-de-Bloco-Final>