

# Projeto

**C**HALLENGE **G**AME  
**I**NFO5 **3**C

**LSC**

Soluções e consultoria em TI e Negócios

## Índice

3 .....	Apresentação;
3 .....	Objetivo;
3 .....	Infraestrutura Atual;
4 .....	Método de Implementação;
4 .....	Requisitos de Software;
5 .....	Requisitos de Hardwares;
5 .....	Máquinas Virtuais;
6 .....	Infraestrutura Necessária;
7 .....	Implementação;
7 .....	Storage FreeNas;
9 .....	Criação do Pool;
12 .....	Servidor ESXi;
15 .....	Criação dos Datastores;
16 .....	Criação das VMs;
18.....	Instalação do Wordpress;
18 .....	Criação da infraestrutura Ansible;
18 .....	Criação do código;
24.....	Publicação do portal na rede onion.;
28.....	Publicação do projeto no Github;
132.....	Atualização do Repositório;
3.....	Gerenciamento da Aplicação(Código Fonte e Versionamento)
36.....	Custos de implantação

## **Apresentação**

Visamos atender à própria LSC na demanda de criação de um portal, on premises, baseado em Wordpress, publicado na rede onion, para ministração de curso de segurança da informação, denominado Projeto Challenge Game Infosec, que entrará para o portfólio da empresa.

## **Objetivo**

Criar o blog/site, publicado na deepweb para utilização no projeto de segurança da informação, Challenge Game Infosec, para utilização dos instrutores que ministrarão o curso, disponibilizado no portfólio da LSC.

O Blog/site será utilizado para apresentação da rede onion, aprendizados em testes de vulnerabilidade e de penetração web, podendo ser utilizados em aulas virtuais ou presenciais.

## **Infraestrutura Atual**

A Infraestrutura atual conta possuir um servidor com capacidade computacional ociosa, em condições de ser utilizado para esta feature. Por isso iremos utilizá-lo em paralelo a outros serviços existentes.

Visamos, com isso, atender simultaneamente dezenas de clientes e instrutores que poderão utilizar este serviço nos cursos que a LSC ministrará e, futuramente, objetivamos migrá-lo para uma nuvem pública.

## Método de implementação

Avaliamos a escolha entre a utilização de VMs ou containers para a implementação de nosso ambiente e , prezando a questão da segurança, pela natureza de nossa aplicação, que envolve publicação em rede onion e utilização para testes de vulnerabilidades e penetração, a melhor escolha, sem sombra de dúvidas é a utilização do ambiente em VMs.

Às VMs são um invólucro que podem ser completamente apartadas da rede e do host, enquanto o containers dividir o kernel do host. Isso poderia significar um problema de segurança para o projeto.

O portal será desenvolvido sobre uma instalação Wordpress com banco de dados MySQL, utilizando um playbook em Ansible para automatização da instalação do wordpress, seu pré requisitos e dependências.

## Requisitos de Software

Abaixo seguem as definição para a estrutura utilizada para a aplicação:

- VmWare ESXi 6;
- VSphere 6;
- FreeNas;
- Ubuntu 20.10;
- PHP versão 7.4.3;
- MySQL versão 8.0;
- Suporte HTTPS;
- Apache 2.4.41;
- Wordpress 5.6.2;

## Requisitos de Hardware

Os Hardwares , como mencionado, são preexistentes e possuem capacidade para operar a aplicação. Abaixo às configurações das máquinas.

### **Storage (Existente):**

Dell PowerEdge NAS Server:  
Procesador Intel® Xeon® E5-2690 v4 - 14 núcleos;  
RAM - 64 GB;  
HDDs - 4TB - RAW (3TB - RAID 5);  
Rede - Placa Ethernet Broadcom 5719 - 4 x 1Gb;  
OS - FreeNas;

### **Servidor Host (Existente):**

Servidor: PowerEdge R430;  
Disco: 4 unidades de 2,5" SSD de troca dinâmica, 1TB (Raid 5);  
Memória: DIMM DDR4 32GB;  
Processador: 1 Intel® Xeon® E5-2600 v4;  
Placas de Rede: 4 x LOM de 1 GbE;  
Fonte de energia: 2x Dell 1100w 09tmrf;  
*OBS.: Não serão mais necessários novos discos.*

## Máquinas Virtuais

Teremos 2 máquinas virtuais que serão usadas como máquina principal e backup *Falt Tolerance*, para migração do serviço em caso de falha.

### **Servidor Virtual 1:**

VM 1 (Wordpress)  
Sistema: Ubuntu 20.10  
Disco: 100 GB (Dinâmica);  
Memória: 16 GB (Dinâmica);  
VCPU: 2 (dedicadas);

### **Servidor Virtual 2 (Backup):**

VM 2 (Wordpress)  
Sistema: Ubuntu 20.10  
Disco: 100 GB (Dinâmica);  
Memória: 16 GB (Dinâmica);  
VCPU: 2 (dedicadas);

## Infraestrutura Necessária

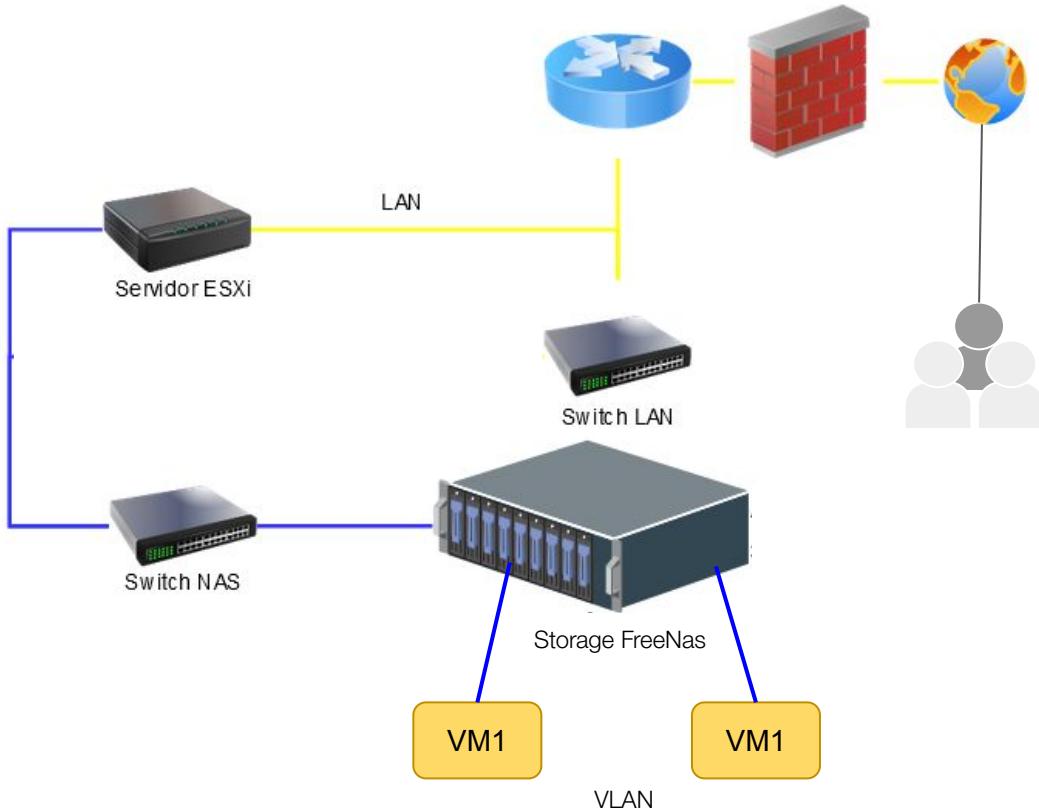
O diagrama a seguir mostra a infraestrutura do projeto, determinando que o servidor deve estar ligados a um switch NAS, utilizando as 2 portas GigaEthernet, da mesma forma se dá para a storage.

Estará disponibilizado na storage, 4 HDs de 1TB, arranjados em Raid 5, entregando 3 discos de 1TB onde utilizaremos 100 GB para cada máquina virtual.

As VMs estarão em VLANs diferentes da rede local para garantir a segurança na publicação do serviço wordpress.

O Vcenter será capaz de gerenciar às VMs, fazendo utilização do Vsphere FT, garantido que o serviço não seja interrompido durante o acesso dos instrutores.

O Vsphere FT será apasi de migrar os serviços da VM principal para a VM de backup caso um falha aconteça.



## Implementação

### Storage FreeNas

Como mencionado, escolhemos o FreeNas para o gerenciamento da nossa storage, e mesmo Free, é uma ferramenta robusta e confiável, o que a torna muito interessante ao projeto, pois reduz custos com licenciamento sem prejudicar a eficiência do projeto.

Abaixo documentamos a instalação do FreeNas e como devem ficar os arranjos de discos para a entrega às máquinas virtuais.

*OBS.: Mesmo sendo uma infraestrutura existente, mostraremos todo o processo de implementação da infraestrutura de VMS para fins de documentação.*



Executamos o FreeNas por meio de boot em um Pendrive.

Bootado o sistema, recebemos a tela de menu da instalação e escolhemos a opção 1;

Receberemos uma nova tela onde a opção que nos interessa é a opção 1(Install/Upgrade).



Feita a escolha na tela anterior, a instalação solicitará a escolha do disco onde o FreeNas será instalado.

Usando as setas do teclado caso tenham mais de uma opção, selecione OK e aperte Enter. O sistema será instalado no seu HD dedicado.



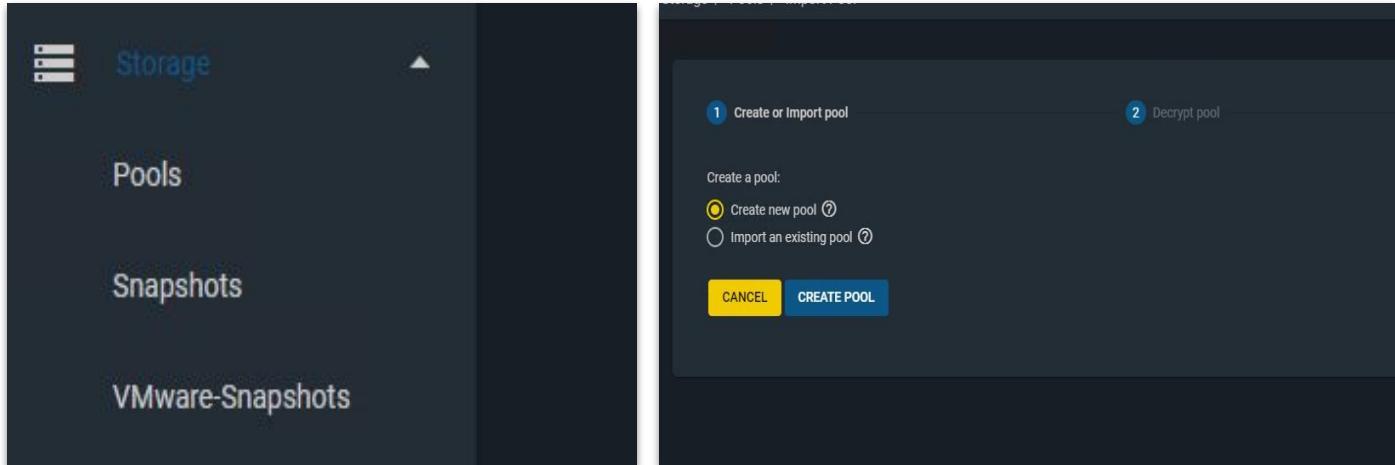
Terminada a instalação e inicialização, o Freenas mostrará uma tela de Setup. Caso o servidor já tenha IP ou esteja em DHCP, será mostrado nesta tela o IP de acesso a interface web de gerenciamento.

Ao entrar na interface, você poderá acessar o gerenciador mediante login e senha, que pode ser configurado na tela anterior.

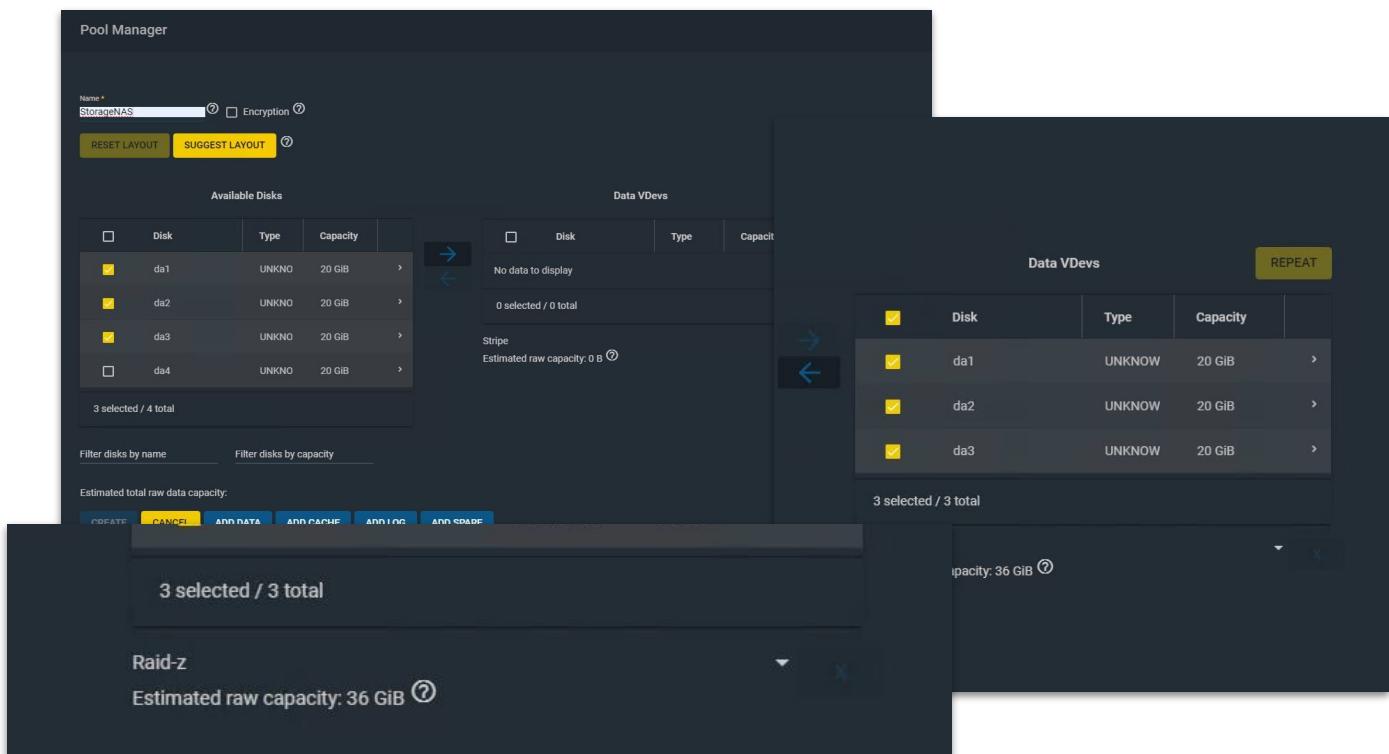
## Criação do Pool

Para a criação do Pool que arranjará os discos no formato que precisamos, vá ao menu Storage -> Pool.

Na próxima tela, Crie um novo Pool.



Selecionado os discos necessário para a criação do Raid 5, clicamos na seta azul para enviar os discos ao DATA VDevs, onde escolhemos a opção de arranjo, em nosso caso, Raid-Z2, que equivale ao Raid 5. \*OBS.: Imagens para fins de instrução. Não reflete a real capacidade do storage atual.



Realizada esta configuração para os discos necessários, teremos os discos empilhados no tamanho que projetamos. \*OBS.: Imagens para fins de instrução. Não reflete a real capacidade do storage atual.

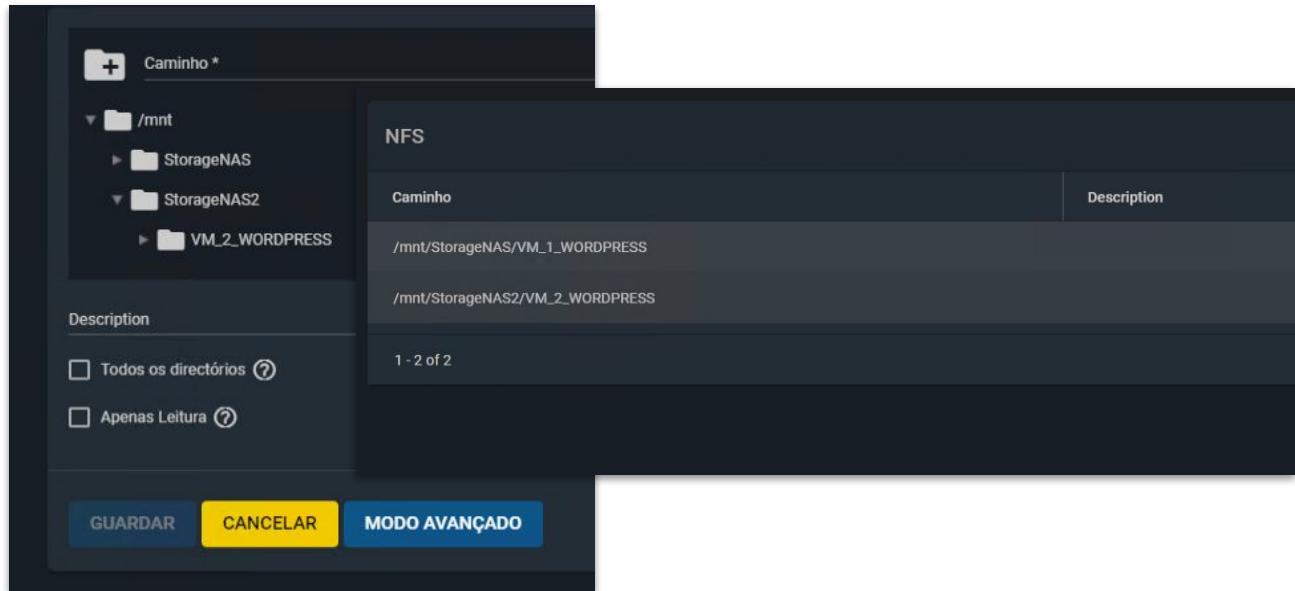
Name	Type	Used	Available	Compression	Compression Ratio	Readonly	Dedup	Comments
StorageNAS	dataset	663.13 MiB	33.87 GiB	lz4	1.13x	false	off	
StorageNAS2	dataset	372.97 KiB	34.52 GiB	lz4	1.00x	false	off	

Com o Pool criado, teremos que clicar no menu hambúrguer para adicionar um Dataset, criando uma identificação para este volume. Esta ação deve ser realizada a todos os Pools.

Na tela abaixo podemos ver a configuração do Dataset e exibição dos datastores e suas configurações de Dataset.

Name	Tipo	Used	Available	Compression	Compression Ratio	Readonly	Dedup	Comentários
StorageNAS	dataset	664 MiB	33.87 GiB	lz4	1.15x	false	off	
VM_1_WORDPRESS	dataset	117.22 KiB	33.87 GiB	Inherits (lz4)	1.00x	false	off	
StorageNAS2	dataset	1001.69 KiB	34.52 GiB	lz4	1.00x	false	off	
VM_2_WORDPRESS	dataset	117.22 KiB	34.52 GiB	Inherits (lz4)	1.00x	false	off	

Retornando ao Menu principal, em “Sharing / Unix Shares (NFS)”, criamos o compartilhamento NFS.

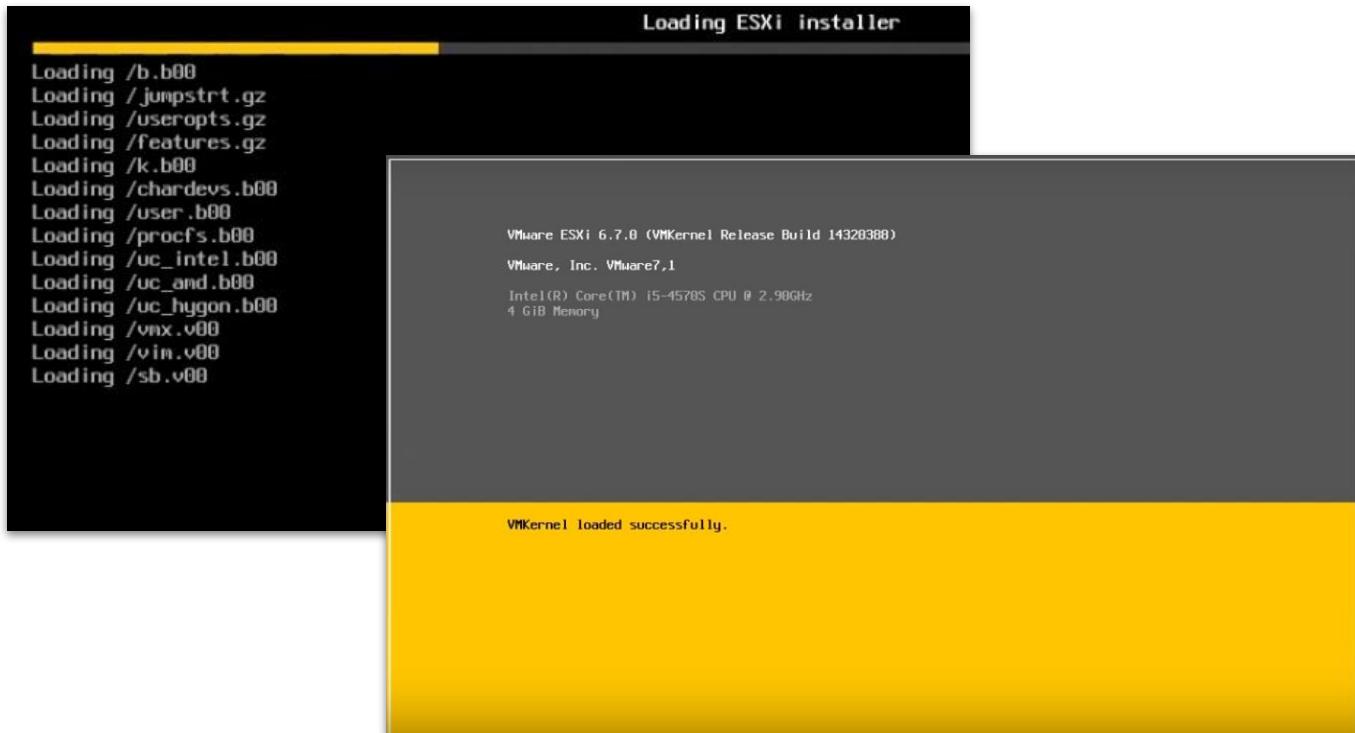


Também no menu hambúrguer do “path”, em “edit permissions”, Finalizamos dando direito de escrita ao servidor ESXi no storage e ver o caminho criado para o compartilhamento.

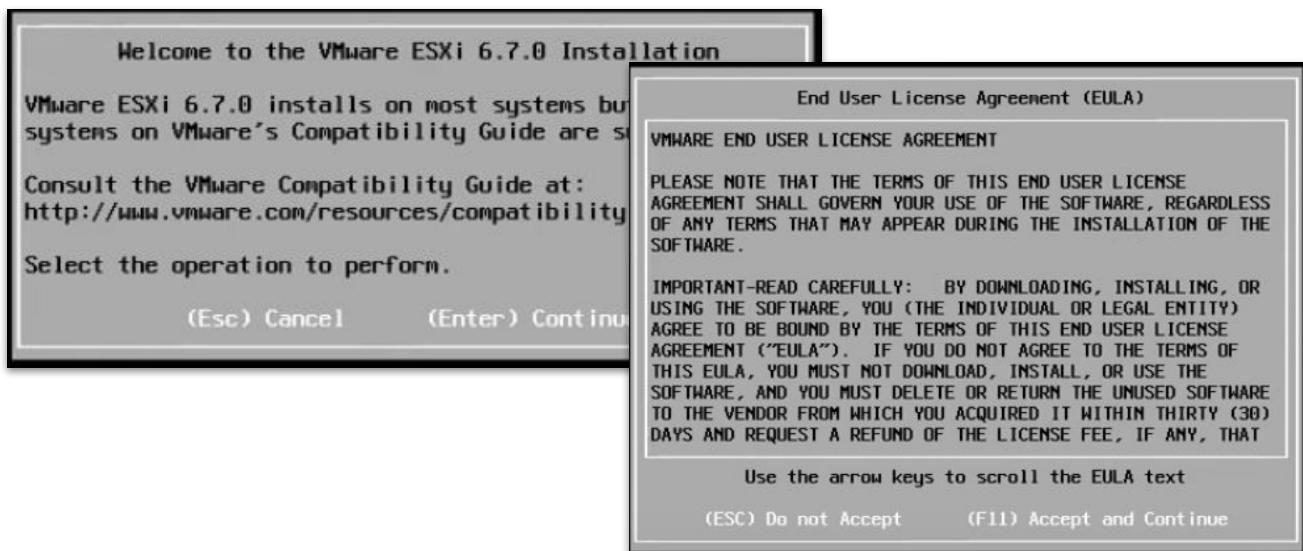
Este caminho será utilizado no mapeamento, no ESXi.

## Servidor ESXi

Para a instalação do ESXi 6.7, iniciamos o boot por disco nos servidores.



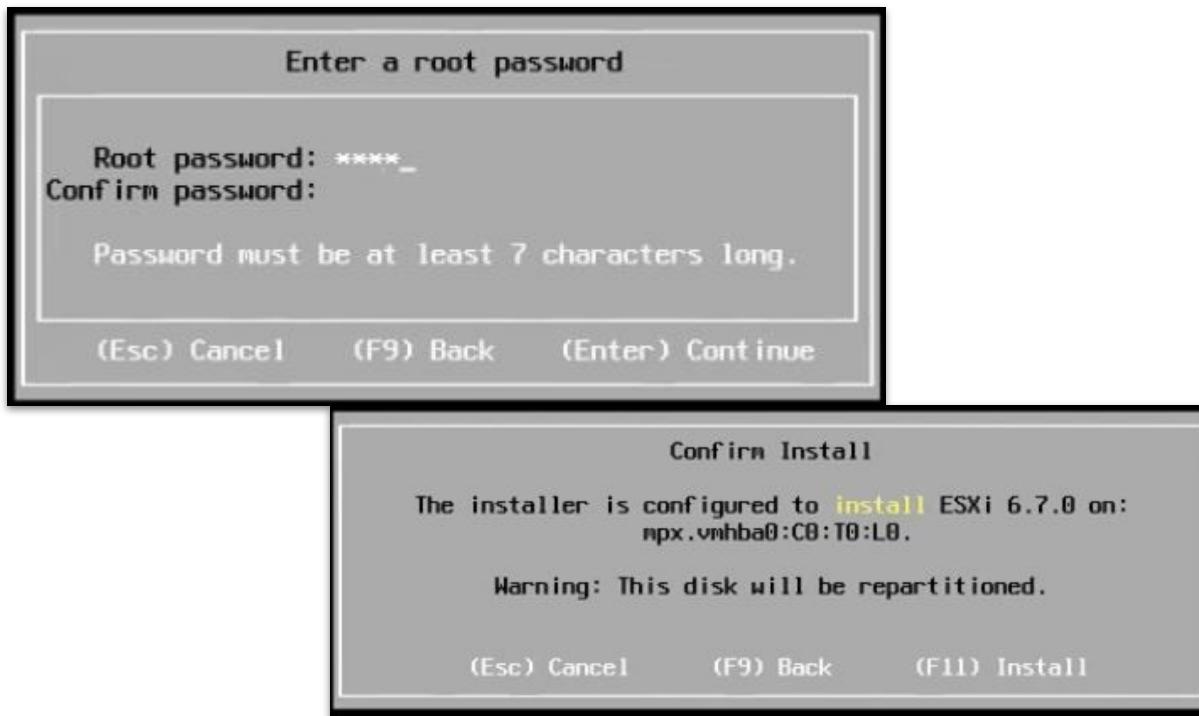
Haverão algumas telas de interação que só precisam da confirmação para prosseguimento, como evidenciado nas imagens.



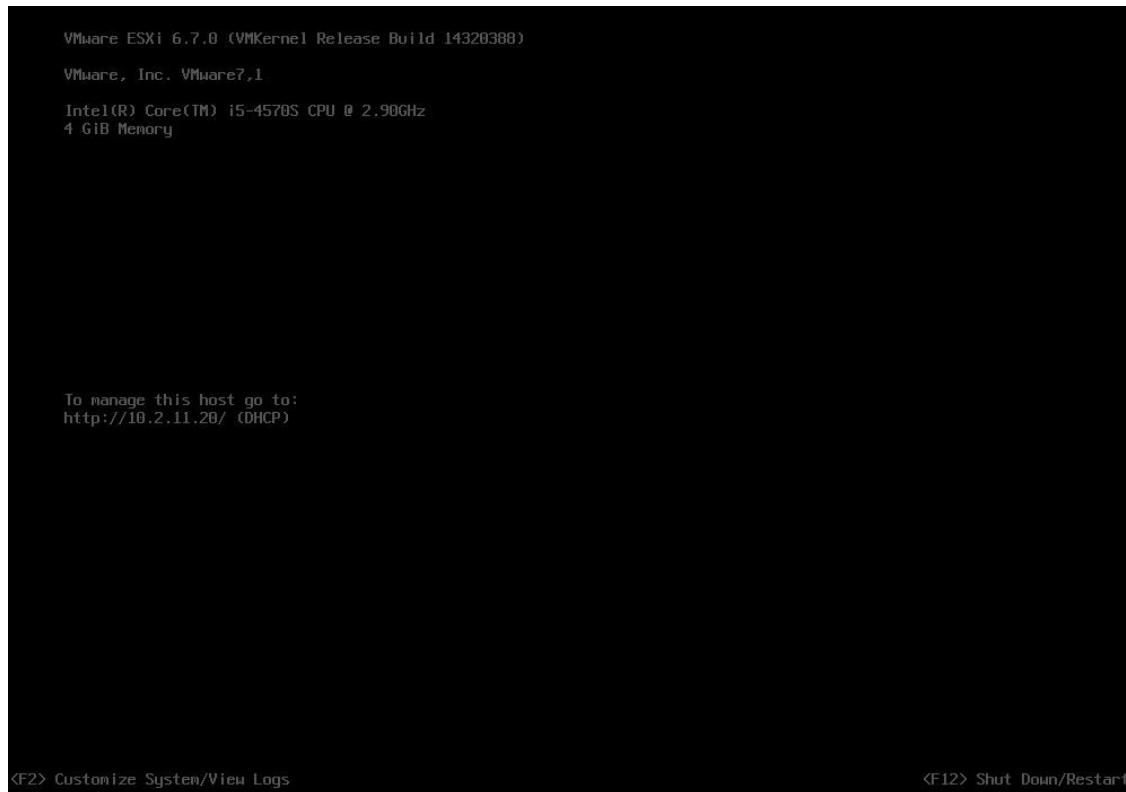
Após elas iremos para a tela onde escolhemos o disco onde será instalado o Sistema Operacional.



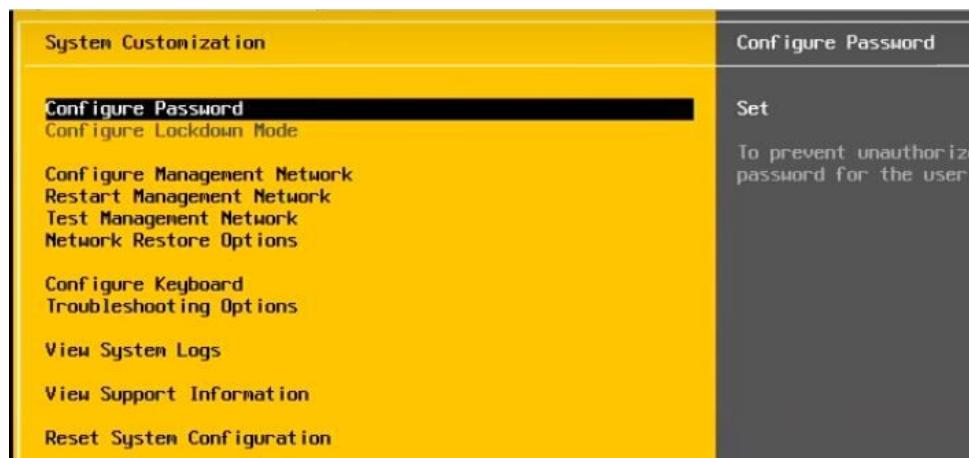
Criamos a senha de root para o sistema, prosseguimos e confirmamos a instalação.



Reiniciado após a instalação, vemos a tela do sistema. Caso o DHCP esteja ativo o servidor já terá um IP para acesso de sua interface web.



Clicando em F2 podemos acessar um menu de customização onde, entre outras coisas, o IP fixo pode ser definido.



Acessando o IP configurado, podemos ter acesso a interface de gerência do ESXi, finalizando sua instalação.

The screenshot shows the VMware ESXi host interface. In the top navigation bar, it says "localhost.lab.infnet.com.br - Virtual Machines". The left sidebar has sections for Host Manage, Monitor, Virtual Machines (4), Storage (1), and Networking (6). The main pane shows a table of virtual machines:

Virtual machine	Status	Used space	Guest OS	Host name	Host CPU	Host memory
vCenter	Normal	20 GB	Microsoft Windows Server 2012 (...)	Unknown	0 MHz	0 MB
VHC	Normal	16 GB	VMware Photon OS (64-bit)	Unknown	0 MHz	0 MB
VM 1 Wordpress	Normal	0 B	Ubuntu Linux (64-bit)	Unknown	0 MHz	0 MB
VM 2 Wordpress	Normal	0 B	Ubuntu Linux (64-bit)	Unknown	0 MHz	0 MB

Below this is the "Storage" section with tabs for Datastores, Adapters, Devices, and Persistent Memory. It shows three datastores:

Name	Drive Type	Capacity	Provisioned	Free	Type	Thin provisioning	Access
datastore1	Non-SSD	92.5 GB	76 GB	16.5 GB	VMFS6	Supported	Single
VM_1_Wordpress	Unknown	33.87 GB	116 KB	33.87 GB	NFS	Supported	Single
VM_2_WORDPRESS	Unknown	34.52 GB	116 KB	34.52 GB	NFS	Supported	Single

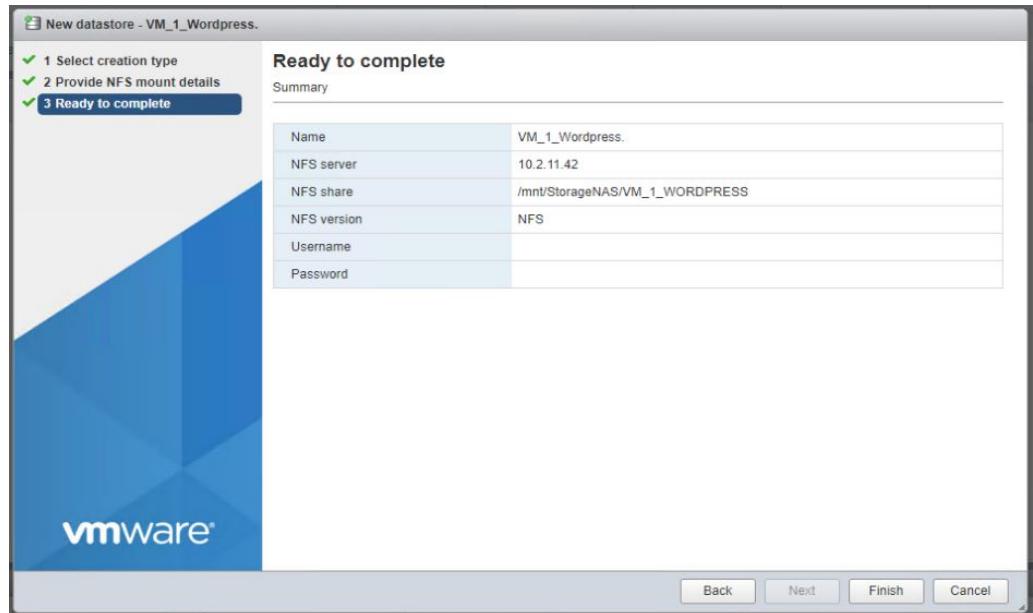
## Criação dos Datastores

Finalizada a instalação do ESXi, podemos usá-lo para a criação das Datastores, apontando às storages criadas no FreeNas. Elas, que serão entregues às máquinas Virtuais. As evidências abaixo mostram o apontamento efetuado, passo à passo.

The screenshot shows the "New datastore" creation wizard. Step 1: Select creation type. The user has chosen "Mount NFS datastore". The other options are: Create new VMFS datastore, Add an extent to existing VMFS datastore, Expand an existing VMFS datastore extent, and Mount NFS datastore.

E **New Datastore** iniciaremos a tela de criação. Escolheremos a montagem em NFS, conforme configuração no freenas.

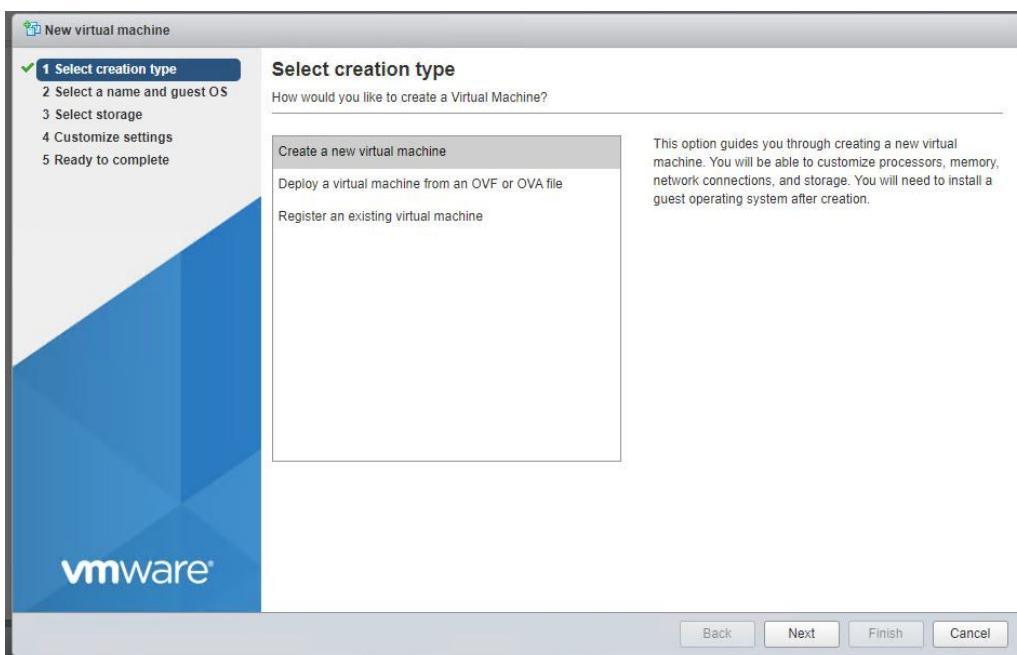
Após preencher, nome do Datastore, o DNS ou IP do servidor NAS e o caminho do compartilhamento NFS. Finalizamos a configuração do data store.

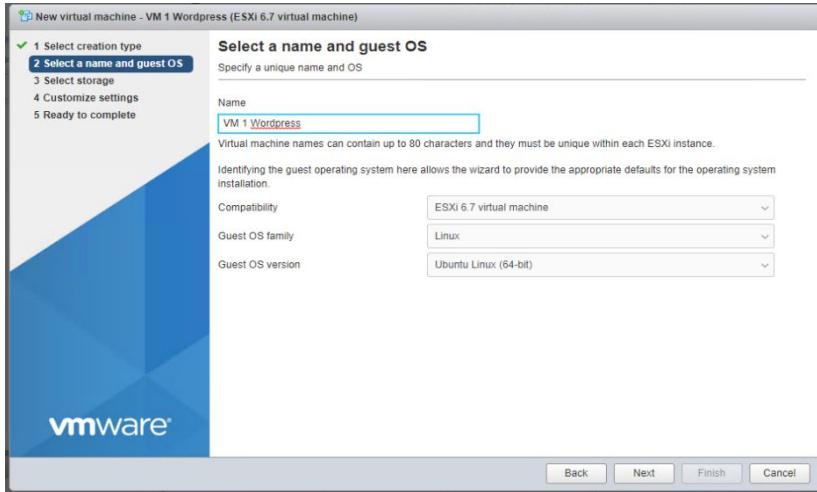


OBS.: Esta configuração se aplicará também ao segundo datastore.

## Criação das VMs

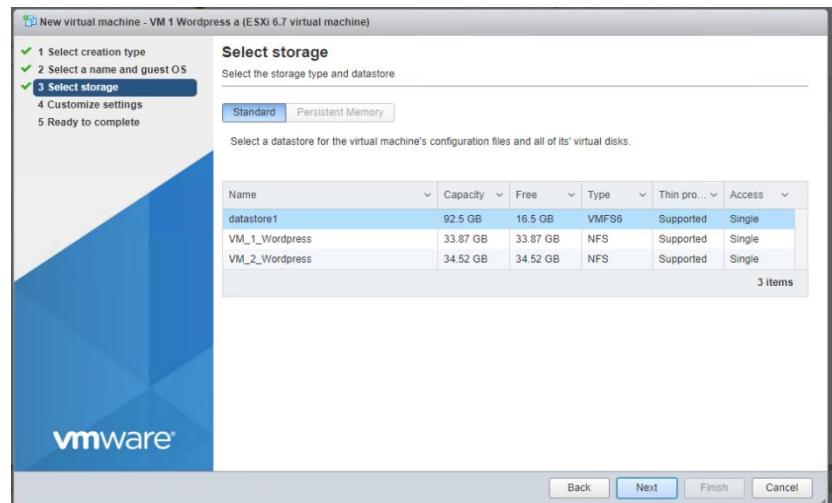
Para a criação das VMs o processo se assemelha. No Dashboard do ESXi, clique em **New Virtual Machine**.



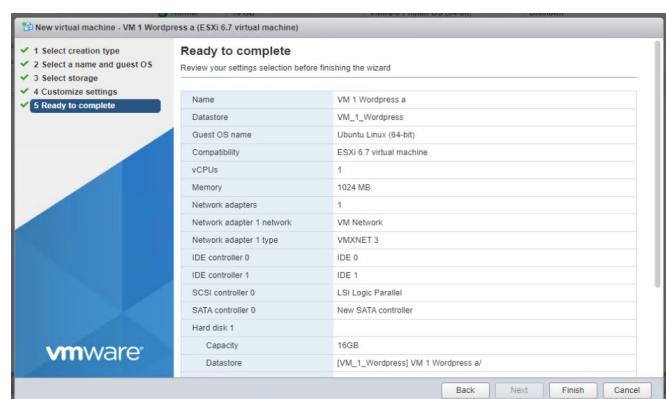


O próximo passo é escolher o nome da VM, selecionar a compatibilidade com hypervisor e o tipo de sistema operacional.

Com os Datastores já apontados, podemos criar os arquivos da VM, já em seu disco de destino na storage.



Configure a máquina virtual conforme o planejamento e finalize a criação. à exemplo do Datastore, repetiremos estas configurações para a criação da segunda VM.



## Instalação do Wordpress

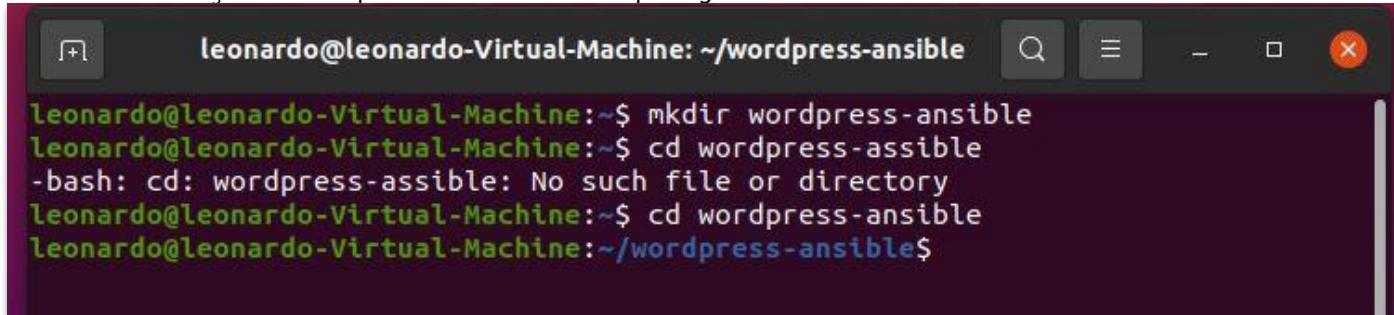
Nos pareceu irrelevante documentar a instalação do sistema operacional Ubuntu, por se tratar de uma atividade trivial.

Passaremos para as atividades de criação do Playbook e sua execução para implementação do projeto.

## Criação da Infraestrutura Ansible

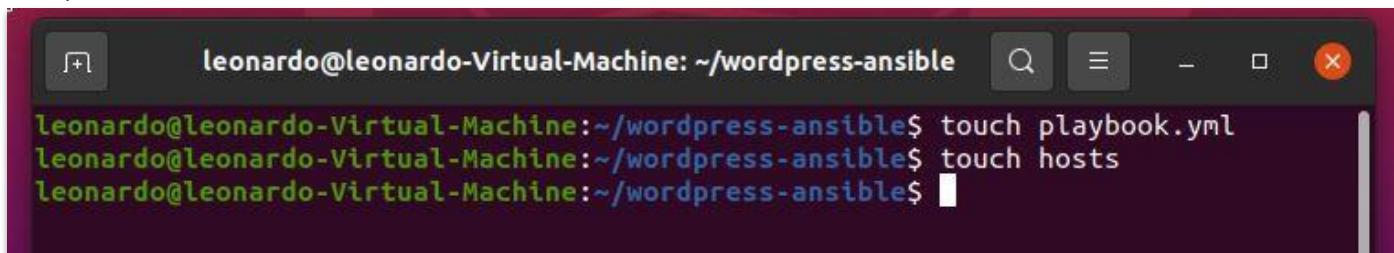
Abaixo serão documentados em imagens os comandos necessários para a criação da infraestrutura necessária ao funcionamento do playbook:

Criação da pasta raiz do projeto:



```
leonardo@leonardo-Virtual-Machine: ~/wordpress-ansible
leonardo@leonardo-Virtual-Machine:~$ mkdir wordpress-ansible
leonardo@leonardo-Virtual-Machine:~$ cd wordpress-ansible
-bash: cd: wordpress-assible: No such file or directory
leonardo@leonardo-Virtual-Machine:~$ cd wordpress-ansible
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible$
```

Criação dos arquivos Hosts (para o inventário dinâmico) e Playbook.yml, onde constará o código para a automatização da instalação completa da aplicação Wordpress e suas dependências:



```
leonardo@leonardo-Virtual-Machine: ~/wordpress-ansible
leonardo@leonardo-Virtual-Machine:~$ touch playbook.yml
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible$ touch hosts
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible$
```

Criação do diretório roles, onde serão criados, efetivamente, a infraestrutura da automatização do Ansible:

```
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible$ mkdir roles
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible$ cd roles
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible/roles$
```

O arquivo playbook.yml foi editado apontando os tipos de roles que serão implementadas:

Host: wordpress

remote\_user: leonardo

become: yes

roles:

-server

-php

-mysql

-wordpress

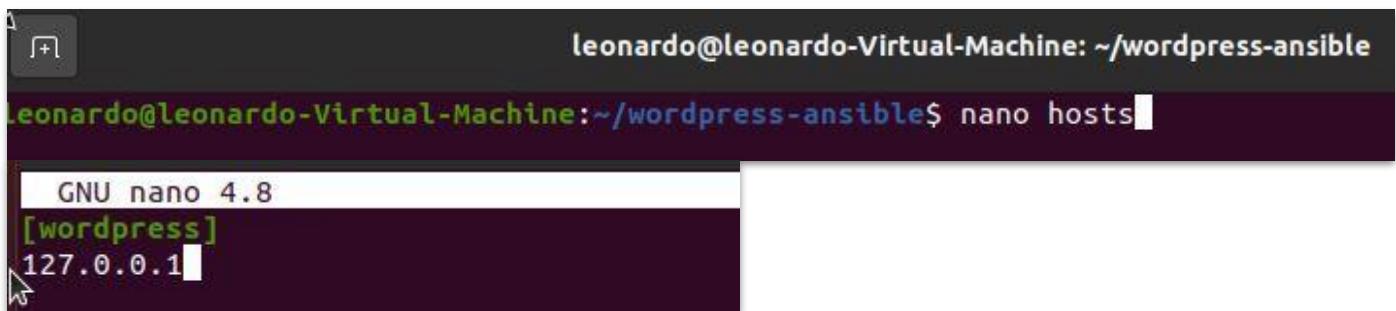
```
! playbook.yml
1   - hosts: wordpress
2     remote_user: leonardo
3     become: yes
4     roles:
5       - server
6       - php
7       - mysql
8       - wordpress
```

Dentro do diretório roles são aplicados os comandos abaixo, para a criação automática da infraestrutura de diretórios necessários:

```
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible/roles$ ansible-galaxy init
server
- Role server was created successfully
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible/roles$ ansible-galaxy init
php
- Role php was created successfully
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible/roles$ ansible-galaxy init
mysql
- Role mysql was created successfully
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible/roles$ wordpress
wordpress: command not found
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible/roles$ ansible-galaxy init
wordpress
- Role wordpress was created successfully
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible/roles$
```

- Role Server  
ansible-galaxy init server
- Mysql  
ansible-galaxy init mysql
- PHP  
ansible-galaxy init PHP
- Wordpress  
ansible-galaxy init wordpress

É necessário apontar a primeira máquina dentro do inventário, que será a própria máquina onde está o projeto. Para isso basta editar o arquivo hosts , introduzindo o IP localhost (Neste mesmo arquivo poderão ser apontadas máquinas remotas que precisarão ter o usuário e grupo aqui mostrado, para que o código possa ser executado):



The screenshot shows a terminal window with the following content:

```
leonardo@leonardo-Virtual-Machine: ~/wordpress-ansible
Leonardo@Leonardo-Virtual-Machine:~/wordpress-ansible$ nano hosts
```

GNU nano 4.8

[wordpress]

127.0.0.1

Editamos o arquivo main nas tasks Mysql, server e wordpress, para que, respectivamente, criasse o banco de dados e usuário com as permissões necessárias, fizesse a instalação dos pacotes necessários para o ambiente onde será instalado o wordpress e instalar o wordpress.

Editamos o arquivo main nas tasks Mysql, server e wordpress, para que, respectivamente, criasse o banco de dados e usuário com as permissões necessárias, fizesse a instalação dos pacotes necessários para o ambiente onde será instalado o wordpress e instalar o wordpress.

```
1 ...
2 # tasks file
3 ...
4 - name: Criar banco de dados MySQL
5   mysql_db:
6     - name: Atualiza o cache do MySQL
7       apt: update_cache=yes
8     - name: "Instalação do MySQL"
9       apt:
10      name:
11        - apache2
12        - mysql-server
13        - php-mysql
14        - php
15        - libapache2-mod-php
16        - mcrypt
17        - python3-mysqldb
18      state: present
19
20 - name: Baixa o WordPress
21   get_url:
22     url: https://wordpress.org/latest.tar.gz
23     dest: /tmp/wordpress.tar.gz
24     validate_certs: no
25
26 - name: Descompacta o WordPress
27   unarchive:
28     src: /tmp/wordpress.tar.gz
29     dest: /var/www/
30     copy: no
31
32 - name: Modifica permissão pasta WordPress
33   file:
34     path: /var/www/wordpress
35     owner: leonardo
36     group: www-data
37     mode: "0755"
38     recurse: yes
39
40 - name: Atualiza o site Apache padrão
41   lineinfile:
42     dest: /etc/apache2/sites-enabled/000-default.conf
43     regexp: "(.)+DocumentRoot /var/www/html"
44     line: "DocumentRoot /var/www/wordpress"
```

O Código está pronto para ser implementado. É necessário atentar para a identação do código a fim de evitar erros durante a execução.

## Execução do Código

Para executar o playbook.yml pelo terminal, é necessário estar na pasta onde está o arquivo. Pelo VS Code, ele iniciará dentro da pasta do projeto.

Execute o comando *\$ ansible-playbook -v -i playbook.yml*.

```
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible$ ansible-playbook -v -i hosts playbook.yml
Using /etc/ansible/ansible.cfg as config file

PLAY [wordpress] ****
TASK [Gathering Facts] ****
ok: [127.0.0.1]

TASK [server : Atualiza o cache apt de nosso servidor] ****
ok: [127.0.0.1] => {"cache_update_time": 1615645251, "cache_updated": false, "changed": false}

TASK [server : instalação dos pacotes para WordPress] ****
ok: [127.0.0.1] => {"cache_update_time": 1615645251, "cache_updated": false, "changed": false}

TASK [php : Instala as extensões PHP necessárias] ****
[DEPRECATION WARNING]: Invoking "apt" only once while using a loop via squash_actions is deprecated.
Instead of using a loop to supply multiple items and specifying name: "{{ item }}", please use name:
['php-gd', 'php-ssh2'] and remove the loop. This feature will be removed in version 2.11. Deprecation
warnings can be disabled by setting ansible_warnings=False in ansible.cfg.
ok: [127.0.0.1] => {"item": ["php-gd", "php-ssh2"], "ansible_loop_var": "item", "cache_update_time": 1615645251, "cache_updated": false, "changed": false, "item": ["php-gd", "php-ssh2"]}

TASK [mysql : Cria o banco de dados para o WordPress] ****
ok: [127.0.0.1] => {"changed": false, "db": "wordpress", "db_list": ["wordpress"]}

TASK [mysql : Cria um usuário de banco para o WordPress] ****
[WARNING]: Module did not set no_log for update_password
changed: [127.0.0.1] => {"changed": true, "msg": "Privileges updated", "user": "wordpress"}

TASK [wordpress : Baixa o WordPress] ****
changed: [127.0.0.1] => {"changed": true, "checksum_dest": null, "checksum_src": "76d1332fcfc5f8b17151b357999d1f758faf897", "dest": "/tmp/wordpress.tar.gz", "elapsed": 37, "gid": 0, "group": "root", "md5sum": "0d048d551e5d3c742b026d57e866f2d2", "mode": "0644", "msg": "OK (1574753 bytes)", "owner": "root", "size": 15747536, "src": "/home/leonardo/.ansible/tmp/ansible-tmp-1615648034.8143096-260557023654172/tmpf6lb3e_/", "state": "file", "status_code": 200, "uid": 0, "url": "https://wordpress.org/latest.tar.gz"}

T Screenshot s : Descompacta o WordPress ****
changed: [127.0.0.1] => {"changed": true, "dest": "/var/www/", "extract_results": {"cmd": "[/usr/bin/tar", "--extract", "-C", "/var/www/", "-z", "-f", "/tmp/wordpress.tar.gz"], "err": "", "out": "", "rc": 0}, "gid": 0, "group": "root", "handler": "TgzArchive", "mode": "0755", "owner": "root", "size": 4096, "src": "/tmp/wordpress.tar.gz", "state": "directory", "uid": 0}

TASK [wordpress : Atualiza o site Apache padrao] ****
ok: [127.0.0.1] => {"backup": "", "changed": false, "msg": ""}

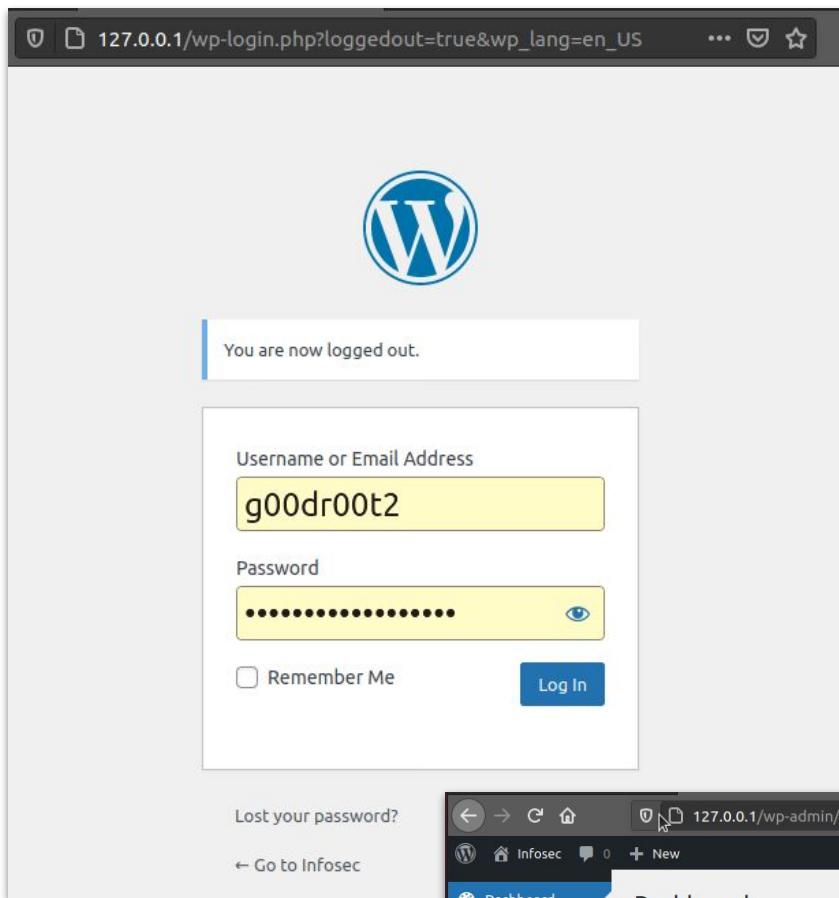
TASK [wordpress : Renomeia o arquivo de configuração de exemplo] ****
ok: [127.0.0.1] => {"changed": false, "cmd": ["mv", "/var/www/wordpress/wp-config-sample.php", "/var/www/wordpress/wp-config.php"], "rc": 0, "stdout": "skipped, since /var/www/wordpress/wp-config.php exists", "stdout_lines": ["skipped, since /var/www/wordpress/wp-config.php exists"]}

TASK [wordpress : Atualiza a configuração WordPress] ****
changed: [127.0.0.1] => {"item": {"regexp": "database_name_here", "line": "define('DB_NAME', 'wordpress');"}, "ansible_loop_var": "item", "changed": true, "item": {"line": "define('DB_NAME', 'wordpress');", "regexp": "database_name_here"}, "msg": "Line replaced"}
changed: [127.0.0.1] => {"item": {"regexp": "username_here", "line": "define('DB_USER', 'wordpress');"}, "ansible_loop_var": "item", "changed": true, "item": {"line": "define('DB_USER', 'wordpress');", "regexp": "username_here"}, "msg": "Line replaced"}
changed: [127.0.0.1] => {"item": {"regexp": "password_here", "line": "define('DB_PASSWORD', 'wp_db_password');"}, "ansible_loop_var": "item", "changed": true, "item": {"line": "define('DB_PASSWORD', 'wp_db_password');", "regexp": "password_here"}, "msg": "Line replaced"}

PLAY RECAP ****
127.0.0.1 : ok=12    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Esta tela será exibida caso nenhum erro ocorra durante a execução.

Neste momento todos os pacotes e dependências já estão instaladas e o wordpress já está funcional.

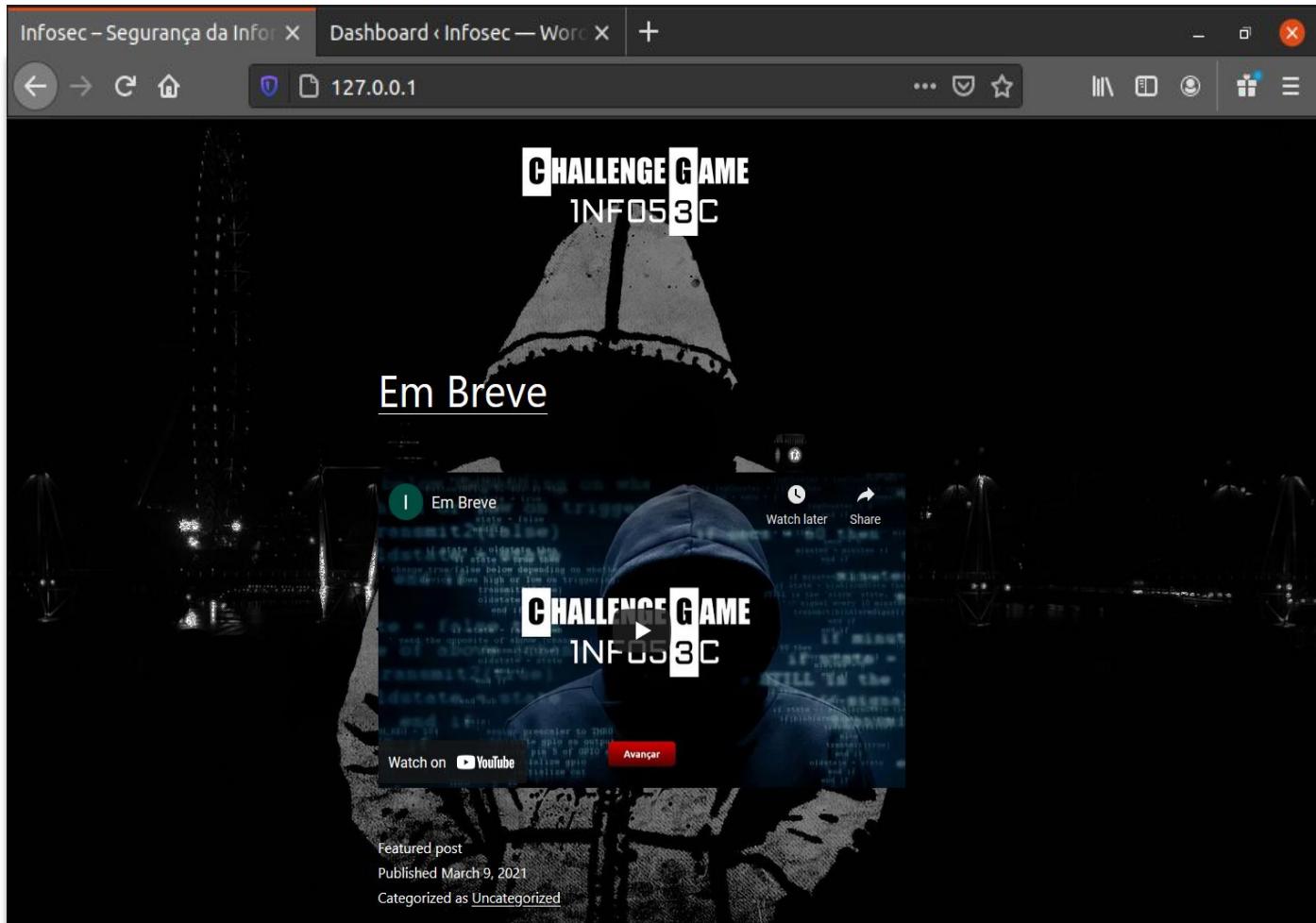


No portal  
WP-Admin foram  
feitas todas as  
customizações  
para a criação do  
portal Challenge  
Game Infosec..

Acesse o  
wordpress com  
seu login e já será  
possível fazer às  
customizações  
necessária para a  
implementação do  
projeto.

A screenshot of the WordPress dashboard at 127.0.0.1/wp-admin/. The dashboard features a "Welcome to WordPress!" message and links for "Get Started" and "Next Steps". It includes sections for "Site Health Status" (indicating "Should be improved") and "Quick Draft" (with fields for "Title" and "Content"). The left sidebar shows a navigation menu with options like Home, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings.

O Site está implementado.



## Publicação do Portal na rede Onion

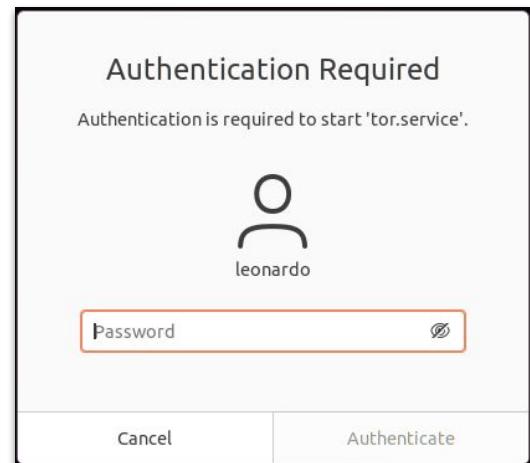
Nosso portal foi publicado na deepweb como forma de marketing para a cultura hacker e de Cybersecurity ao mesmo tempo em que será usado , tecnicamente, para ensinar sobre a própria deep web, rede onion, Hidden Service, Relays Tor e etc.

Para a publicação foram dados os seguintes passos:

```
leonardo@leonardo-Virtual-Machine:~$ sudo apt install tor
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libfprint-2-tod1 liblvm10
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libevent-2.1-7 tor-geoipdb torsocks
Suggested packages:
  mixmaster torbrowser-launcher socat tor-arm apparmor-utils obfs4proxy
The following NEW packages will be installed:
  libevent-2.1-7 tor tor-geoipdb torsocks
0 upgraded, 4 newly installed, 0 to remove and 2 not upgraded.
Need to get 2.577 kB of archives.
After this operation, 13,9 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://br.archive.ubuntu.com/ubuntu focal/main amd64 libevent-2.1-7 amd64 2.1.11-stable-1 [138 kB]
Get:2 http://br.archive.ubuntu.com/ubuntu focal/universe amd64 tor amd64 0.4.2.7-1 [1,410 kB]
Get:3 http://br.archive.ubuntu.com/ubuntu focal/universe amd64 torsocks amd64 2.3.0-2 [61,5 kB]
Get:4 http://br.archive.ubuntu.com/ubuntu focal/universe amd64 tor-geoipdb all 0.4.2.7-1 [968 kB]
Fetched 2.577 kB in 2s (1,633 kB/s)
Selecting previously unselected package libevent-2.1-7:amd64.
(Reading database ... 184176 files and directories currently installed.)
Preparing to unpack .../libevent-2.1-7_2.1.11-stable-1_amd64.deb ...
U Text Editor :libevent-2.1-7:amd64 (2.1.11-stable-1) ...
Selecting previously unselected package tor.
Preparing to unpack .../tor_0.4.2.7-1_amd64.deb ...
Unpacking tor (0.4.2.7-1) ...
Selecting previously unselected package torsocks.
Preparing to unpack .../torsocks_2.3.0-2_amd64.deb ...
Unpacking torsocks (2.3.0-2) ...
Selecting previously unselected package tor-geoipdb.
Preparing to unpack .../tor-geoipdb_0.4.2.7-1_all.deb ...
Unpacking tor-geoipdb (0.4.2.7-1) ...
[Progress: [==== 41% [#####
leonardo@leonardo-Virtual-Machine:~$ service tor start
```

Instale o Daemon do Tor pelo comando *\$ sudo apt install tor* e inicialize o serviço tor com *\$ service tor start*;

Faça a autenticação no sistema e com o comando *\$ service status*, confira a execução do serviço



```
leonardo@leonardo-Virtual-Machine:~$ service tor status
● tor.service - Anonymizing overlay network for TCP (multi-instance-master)
  Loaded: loaded (/lib/systemd/system/tor.service; enabled; vendor preset: enabled)
  Active: active (exited) since Sat 2021-03-13 12:51:35 -03; 3min 26s ago
    Main PID: 21387 (code=exited, status=0/SUCCESS)
      Tasks: 0 (limit: 4539)
     Memory: 0B
        CGroup: /system.slice/tor.service

mar 13 12:51:35 leonardo-Virtual-Machine systemd[1]: Starting Anonymizing overlay network for TCP (multi>
mar 13 12:51:35 leonardo-Virtual-Machine systemd[1]: Finished Anonymizing overlay network for TCP (multi>
lines 1-10/10 (END)
```

Edite o arquivo torrc no caminho /etc/tor/

```
root@leonardo-Virtual-Machine:/home/leonardo# nano /etc/tor/torrc
root@leonardo-Virtual-Machine:/home/leonardo# █
```

```
## HiddenServicePort x:y:z says to redirect re
## address y:z.

HiddenServiceDir /var/lib/tor/hidden_service/
HiddenServicePort 80 127.0.0.1:80

#HiddenServiceDir /var/lib/tor/other_hidden_se
#HiddenServicePort 80 127.0.0.1:80
#HiddenServicePort 22 127.0.0.1:22
█
##### This section is just for rela
#
```

Descomente  
o serviço  
*hiddenserviceport*  
80 e  
*HiddenServiceDir*  
*/var/lib/tor/hidden*  
*service/* para que o  
serviço

Hidden\_service seja criado na porta 80.

No diretório */var/lib/tor/hiddenservice/* é onde serão salvo às chaves de acesso do Hidden\_service.

Salve a edição e reinicie o daemon do Tor com o comando **\$ systemctl restart tor** e verifique seu status com o comando **\$ systemctl status tor**.

```
root@leonardo-Virtual-Machine:/home/leonardo# systemctl restart tor
root@leonardo-Virtual-Machine:/home/leonardo# systemctl status tor
● tor.service - Anonymizing overlay network for TCP (multi-instance-master)
   Loaded: loaded (/lib/systemd/system/tor.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sat 2021-03-13 16:19:26 -03; 15s ago
     Process: 23074 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 23074 (code=exited, status=0/SUCCESS)

mar 13 16:19:26 leonardo-Virtual-Machine systemd[1]: Starting Anonymizing overlay network for TCP (mult>
mar 13 16:19:26 leonardo-Virtual-Machine systemd[1]: Finished Anonymizing overlay network for TCP (mult>
lines 1-8/8 (END)
```

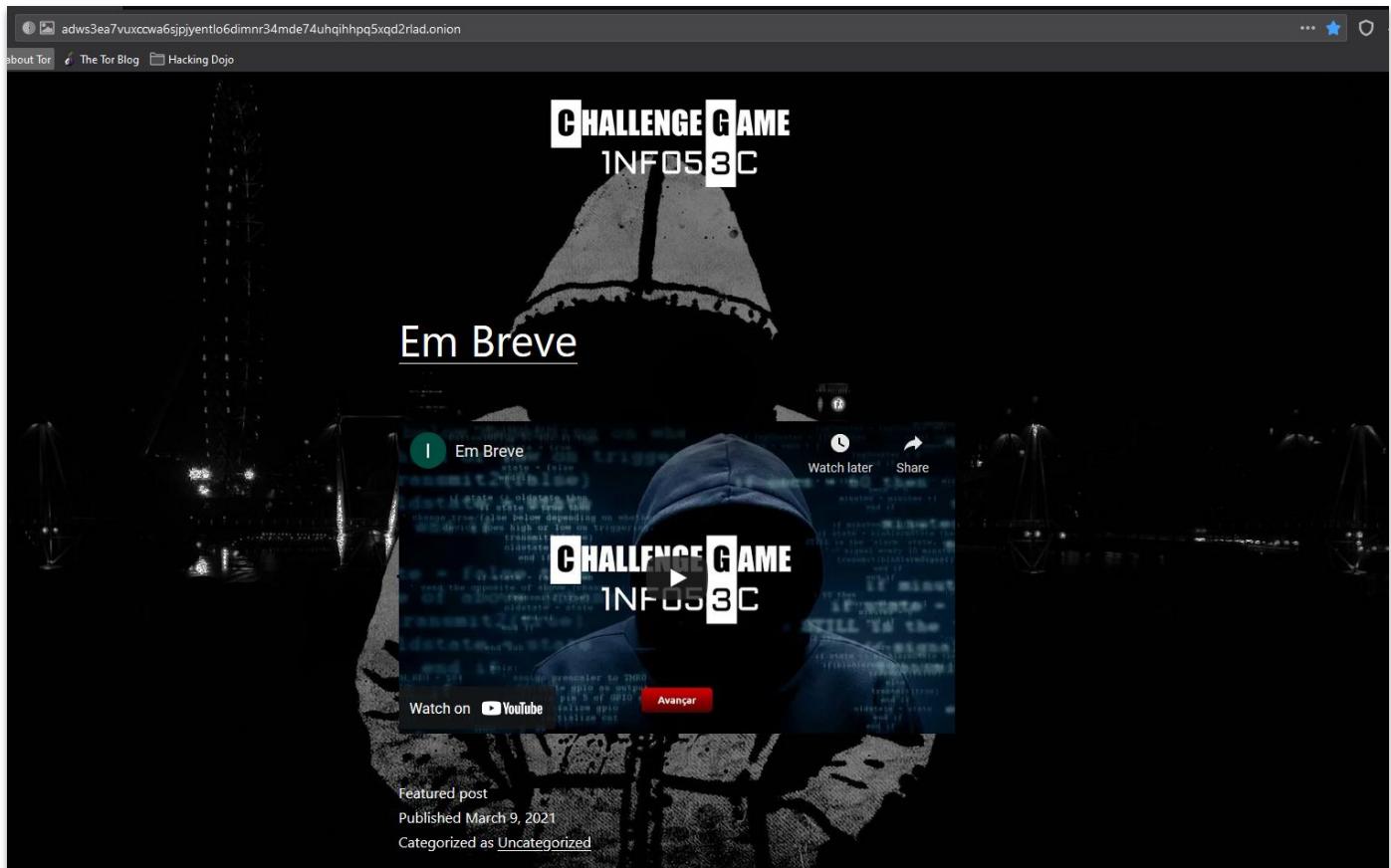
O hidden\_service estará ativo bastando buscar a URL da rede onion gerada no arquivo host dentro do caminho */var/lib/tor/hidden\_service*

Ao ler o arquivo se pode ver a URL típica e pouco amigável da rede onion:

```
root@leonardo-Virtual-Machine:/home/leonardo# cd /var/lib/tor/hidden_service
root@leonardo-Virtual-Machine:/var/lib/tor/hidden_service# ls
authorized_clients  hostname  hs_ed25519_public_key  hs_ed25519_secret_key
root@leonardo-Virtual-Machine:/var/lib/tor/hidden_service# cat hostname
adws3ea7vuxccwa6sjpjyentlo6dimnr34mde74uhqihhpq5xqd2rlad.onion
root@leonardo-Virtual-Machine:/var/lib/tor/hidden_service#
```

A url

<http://adws3ea7vuxccwa6sjpjyentlo6dimnr34mde74uhqihhpq5xqd2rlad.onion/> já pode ser acessada externamente utilizando configurações específicas para o proxy da rede Onion.



Alguns softwares ou Sistemas operacionais, como o Tor Browser e o sistema operacional Tails, trazem às configurações automáticas para navegar na rede tor, mas qualquer sistema operacional pode ser configurado para acessá-la.

## Publicação do Projeto no GitHub

Vamos manter o projeto publicado no github para que possa facilmente ser acessado, reutilizado e, porque não, adaptado para outras implantações e utilizações.

A primeira ação é a instalação do daemon git, com o comando `# sudo apt install git`.

A segunda ação é criar um repositório no github; O repositório foi criado e sua url é:

<https://github.com/LEOSCAMPOS/wordpress-ansible>

Para a iniciar a publicação é necessário seguir os comandos:

`git init` - Preparará o ambiente para operar com o git

```
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible$ git init
Reinitialized existing Git repository in /home/leonardo/wordpress-ansible/.git/
leonardo@leonardo-Virtual-Machine:~/wordpress-ansible$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hosts
    playbook.yml
    roles/
    workspace.code-workspace

nothing added to commit but untracked files present (use "git add" to track)
```

`git add` - Adicionar todos os arquivos do projeto ao sistema de versionamento:

```
root@leonardo-Virtual-Machine:/home/leonardo/wordpress-ansible# git add *
root@leonardo-Virtual-Machine:/home/leonardo/wordpress-ansible# git status
On branch master

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file:  hosts
  new file:  playbook.yml
  new file:  roles/mysql/.travis.yml
  new file:  roles/mysql/README.md
  new file:  roles/mysql/defaults/main.yml
  new file:  roles/mysql/handlers/main.yml
  new file:  roles/mysql/meta/main.yml
  new file:  roles/mysql/tasks/main.yml
  new file:  roles/mysql/tests/inventory
  new file:  roles/mysql/tests/test.yml
  new file:  roles/mysql/vars/main.yml
  new file:  roles/php/.travis.yml
  new file:  roles/php/README.md
  new file:  roles/php/defaults/main.yml
  new file:  roles/php/handlers/main.yml
  new file:  roles/php/meta/main.yml
  new file:  roles/php/tasks/main.yml
  new file:  roles/php/tests/inventory
  new file:  roles/php/tests/test.yml
  new file:  roles/php/vars/main.yml
  new file:  roles/server/.travis.yml
  new file:  roles/server/README.md
  new file:  roles/server/defaults/main.yml
  new file:  roles/server/handlers/main.yml
  new file:  roles/server/meta/main.yml
  new file:  roles/server/tasks/main.yml
  new file:  roles/server/tests/inventory
  new file:  roles/server/tests/test.yml
  new file:  roles/server/vars/main.yml
  new file:  roles/wordpress/.travis.yml
  new file:  roles/wordpress/README.md
  new file:  roles/wordpress/defaults/main.yml
  new file:  roles/wordpress/handlers/main.yml
  new file:  roles/wordpress/meta/main.yml
  new file:  roles/wordpress/tasks/main.yml
  new file:  roles/wordpress/tests/inventory
  new file:  roles/wordpress/tests/test.yml
  new file:  roles/wordpress/vars/main.yml
  new file:  workspace.code-workspace
```

*git commit -m - Execute o primeiro commit dos arquivos.*

```
root@leonardo-Virtual-Machine:/home/leonardo/wordpress-ansible# git commit -m "primeiro commit"
[master (root-commit) a98bc29] primeiro commit
 39 files changed, 633 insertions(+)
 create mode 100644 hosts
 create mode 100644 playbook.yml
 create mode 100644 roles/mysql/.travis.yml
 create mode 100644 roles/mysql/README.md
 create mode 100644 roles/mysql/defaults/main.yml
 create mode 100644 roles/mysql/handlers/main.yml
 create mode 100644 roles/mysql/meta/main.yml
 create mode 100644 roles/mysql/tasks/main.yml
 create mode 100644 roles/mysql/tests/inventory
 create mode 100644 roles/mysql/tests/test.yml
 create mode 100644 roles/mysql/vars/main.yml
 create mode 100644 roles/php/.travis.yml
 create mode 100644 roles/php/README.md
 create mode 100644 roles/php/defaults/main.yml
 create mode 100644 roles/php/handlers/main.yml
 create mode 100644 roles/php/meta/main.yml
 create mode 100644 roles/php/tasks/main.yml
 create mode 100644 roles/php/tests/inventory
 create mode 100644 roles/php/tests/test.yml
 create mode 100644 roles/php/vars/main.yml
 create mode 100644 roles/server/.travis.yml
 create mode 100644 roles/server/README.md
 create mode 100644 roles/server/defaults/main.yml
 create mode 100644 roles/server/handlers/main.yml
 create mode 100644 roles/server/meta/main.yml
 create mode 100644 roles/server/tasks/main.yml
 create mode 100644 roles/server/tests/inventory
 create mode 100644 roles/server/tests/test.yml
 create mode 100644 roles/server/vars/main.yml
 create mode 100644 roles/wordpress/.travis.yml
 create mode 100644 roles/wordpress/README.md
 create mode 100644 roles/wordpress/defaults/main.yml
 create mode 100644 roles/wordpress/handlers/main.yml
 create mode 100644 roles/wordpress/meta/main.yml
 create mode 100644 roles/wordpress/tasks/main.yml
```

*git remote add origin* - Aponta o projeto para o  
repositório criado no Github na url  
<https://github.com/LEOSCAMPOS/wordpress-ansible>

```
root@leonardo-Virtual-Machine:/home/leonardo/wordpress-ansible# git remote add origin https://github.com/LEOSCAMPOS/Wordpress-Ansible.git
```

Com o comando git push -u origin master os arquivos são enviados para o repositório.

```
root@leonardo-Virtual-Machine:/home/leonardo/wordpress-ansible# git push -u origin master
Username for 'https://github.com': nadinholsc@gmail.com
Password for 'https://nadinholsc@gmail.com@github.com':
Enumerating objects: 55, done.
Counting objects: 100% (55/55), done.
Compressing objects: 100% (26/26), done.
Writing objects: 100% (55/55), 6.20 KiB | 219.00 KiB/s, done.
Total 55 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/LEOSCAMPOS/Wordpress-Ansible.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

root Atualizacao		5 minutes ago	2
	roles	primeiro commit	4 days ago
	hosts	primeiro commit	4 days ago
	playbook.yml	Atualizacao	5 minutes ago
	workspace.code-work...	primeiro commit	4 days ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

 master  [wordpress-ansible / playbook.yml](#) [Go to file](#) 

 root Atualizacao Latest commit 2111a75 6 minutes ago  History

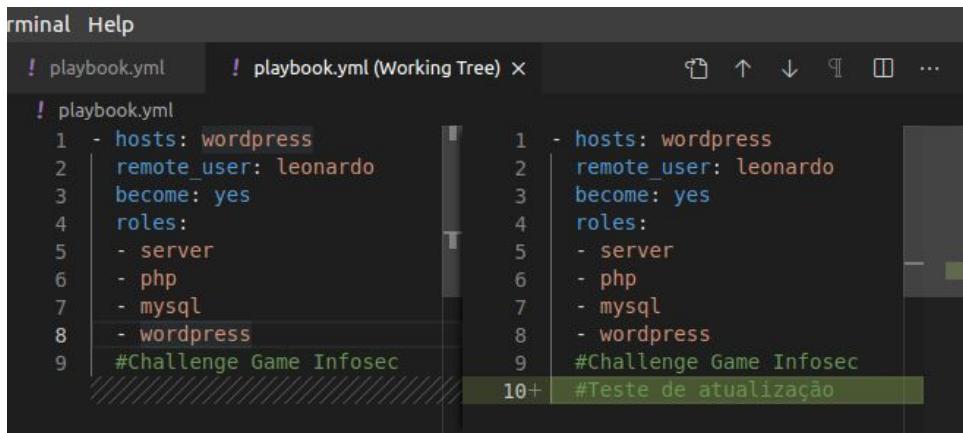
 0 contributors

9 lines (9 sloc) | 134 Bytes [Raw](#) [Blame](#)  

```
1 - hosts: wordpress
2   remote_user: leonardo
3   become: yes
4   roles:
5     - server
6     - php
7     - mysql
8     - wordpress
9   #Challenge Game Infosec
```

## Atualização do repositório

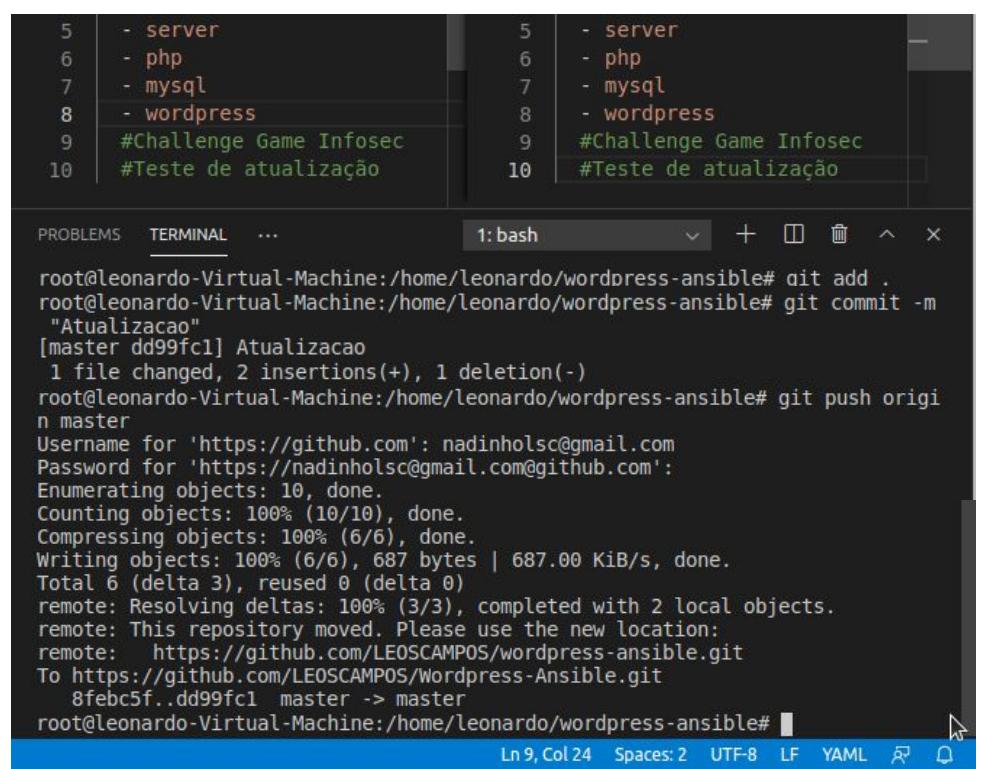
Para o teste de atualização do repositório, usamos o VS Code para realizar uma pequena alteração no código e posteriormente realizamos a sincronia para o repositório.



```
! playbook.yml      ! playbook.yml (Working Tree) X
! playbook.yml
1 - hosts: wordpress
2   remote_user: leonardo
3   become: yes
4   roles:
5     - server
6     - php
7     - mysql
8     - wordpress
9   #Challenge Game Infosec
10+ #Teste de atualização
```

Incluímos um pequeno comentário como teste de alteração no código.

Em seguida realizamos os comandos necessários para a sincronização do projeto local com o repositório no github.



```
PROBLEMS TERMINAL ...
1: bash
root@leonardo-Virtual-Machine:/home/leonardo/wordpress-ansible# git add .
root@leonardo-Virtual-Machine:/home/leonardo/wordpress-ansible# git commit -m "Atualizacao"
[master dd99fc1] Atualizacao
 1 file changed, 2 insertions(+), 1 deletion(-)
root@leonardo-Virtual-Machine:/home/leonardo/wordpress-ansible# git push origin master
Username for 'https://github.com': nadinholsc@gmail.com
Password for 'https://nadinholsc@gmail.com@github.com':
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 687 bytes | 687.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
remote: This repository moved. Please use the new location:
remote: https://github.com/LEOSCAMPOS/Wordpress-Ansible.git
To https://github.com/LEOSCAMPOS/Wordpress-Ansible.git
 8febc5f..dd99fc1 master -> master
root@leonardo-Virtual-Machine:/home/leonardo/wordpress-ansible#
```

Então podemos ver nos logs e no próprio código de que às alterações realizadas localmente, foram sincronizadas com o projeto remoto.

master ▾ [wordpress-ansible / playbook.yml](#) Go to file ⋮

 **LEOSCAMPOS** Update playbook.yml Latest commit 8febc5f now ⏲ History

1 contributor

10 lines (10 sloc) | 161 Bytes Raw Blame ⚪ 🗑

```
1 - hosts: wordpress
2   remote_user: leonardo
3   become: yes
4   roles:
5     - server
6     - php
7     - mysql
8     - wordpress
9   #Challenge Game Infosec
10  #Teste de atualização
```

[wordpress-ansible / playbook.yml](#) Newer ⏲ Older

100644 | 9 lines (9 sloc) | 134 Bytes Raw Normal view History

primeiro commit	7 days ago	1	- hosts: wordpress
		2	remote_user: leonardo
		3	become: yes
		4	roles:
		5	- server
		6	- php
		7	- mysql
Atualizacao	3 days ago	8	- wordpress
		9	#Challenge Game Infosec

## **Requisições de desenvolvimento/correções**

Após disponibilizado pela equipe de infra, o hardware, em condições, será entregue a equipe de desenvolvimento, que trabalhará em dividida em cenários de entregas distintos, com milestones divididos da seguinte maneira:

- Desenvolvimento e Correção

Ambiente de entrega do primeiro corpo do projeto e retorno de correção de bugs reportados.

- Homologação

Ambiente para teste das versões disponibilizadas, após correção, para validação interna.

- Entrega

Ambiente que disponibiliza ao usuário final o produto acabado.

## **Gerenciamento da Aplicação (Código Fonte e versionamento)**

Será disponibilizado, por meio do Github, toda a codificação para a implementação da infraestrutura lógica escrita no ansible, possibilitando todo o gerenciamento de versões.

A Equipe utilizará o VS Code para construir e compilar a infraestrutura lógica. O VS Code estará sincronizado com a conta Github para automatizar às versionamentos na ferramenta.

## Plano de implementação

Infraestrutura Lógica			
Criação e instalação das VMs e provisionamento dos discos	Dia 1	2	Equipe de Infra
Configuração do Ansible	Dia 3	2	Equipe de Infra
Criação dos ambientes de Milestones	Dia 5	2	Equipe de Infra
Configuração da página	Dia 07	05	Equipe de desenvolvimento e segurança
Publicação	Dia 11	1	Equipe de desenvolvimento e segurança

## Custos de implantação

<b>Custos de Software</b>			
<b>Software</b>	<b>Versão</b>	<b>Quant.</b>	<b>Custo</b>
VmWare ESXi	6	1	Licença existente
Vmware Vsphere	6	1	Licença existente
Ubuntu	20.10	2	Free
Wordpress	5.6	2	Free
FreeNas	10	1	Free
PHP	7.4	2	Free
MySQL	8	2	Free
Apache	2.4	2	Free
<b>Custos de Hardware</b>			
Dell PowerEdge NAS Server		1	Hardware Existente
PowerEdge R430		1	Hardware Existente

.