

**Instituto Infnet**



**TP 9**

**Projeto de bloco:  
Arquitetura de Infraestrutura de Aplicações**

**Professor: André Mattos**

**Felipe Santos de Paiva Rio**

**056950277-21**

## **Sumário:**

1	Introdução .....	3
2	TP1 .....	4-8
3	TP2 .....	9-19
4	TP3 .....	20-29
5	TP4 .....	30-38
6	TP5 .....	39-44
7	TP6 .....	45-49
8	TP7 .....	50-54
9	TP8 .....	55-69
10	TP9 .....	70-73
12	Conclusões .....	73
13	Repositório e Fontes .....	74

## **Introdução:**

Faça upload do conteúdo completo de seu Projeto de Bloco até aqui, incluindo todos os capítulos apresentados nos TPs 6, 7 e 8, mais uma versão inicial do capítulo 4/Conclusões.

Além dos requisitos apresentados para os capítulos anteriores, suas conclusões devem incluir:

- a. Se o prazo estabelecido no início do projeto foi adequado para a execução (não há problema se levou mais tempo que o planejado, mas se levou, explique as razões).
- b. Se os recursos planejados (quantidade de memória, disco etc.) foram suficientes para colocar a solução em funcionamento. Se não foram, explique as razões e sugira uma nova configuração.
- c. Se as funcionalidades previstas na solução original funcionaram a contento. Se não foram, explique as razões.
- d. Quais seriam as melhorias futuras que poderiam ser feitas no projeto executado, adotando soluções adicionais, aperfeiçoando instalações ou tornando o cenário mais complexo.
- e. Antes da entrega final, todo o conteúdo de seu trabalho (mesmo que sejam apenas documentos do Word e PDFs) deve ser armazenado em um repositório GitHub, cujo link deve estar contido no corpo do trabalho.

*\*Todo o processo foi executado em paralelo com as DRs de fundamentos, que tem as rubricas compartilhadas com os TPs de projeto de bloco.*

*\*\*Para reduzir o volume total, operações repetidas no ambiente em TPs distintos foram suprimidas.*

## **TP1**

### **Introdução:**

Para se iniciar a consultoria da implementação de um projeto em nuvem, que foi avaliado em uma entrevista com a empresa com o intuito de encaixar as necessidades do negócio da organização dentro do projeto a aplicação que irá passar pelo projeto será o WordPress.

### **Objetivos:**

Como uma avaliação envolve os pontos chaves da necessidade de performance e disponibilidade da aplicação, uma forma de ter reports de performance para usuários finais, um melhor controle da qualidade de entrega do código, com o nivelamento dos erros e ajustes de acordo com as requisições do usuário final todos os processos de TI devem estar alinhados com o time de DevOps.

### **Integração contínua, versionamento e homologação:**

O processo de desenvolvimento e implantação absorve muito tempo e recursos técnicos, e de acordo com as novas funcionalidades desenvolvidas, será adotado o modelo de Agile de Infraestrutura como Código, com a aplicação se iniciando baseada em uma Golden Image, com a porção de que a aplicação funciona sem problemas.

Os processos de compilação, instalação e homologação serão gerenciados utilizando o Atom e GitHUB, que é gratuito e centralizado em nuvem. Onde os desenvolvedores irão concentrar as atualizações ao código base como repositório de confiança, com versionamento onde a contribuição e compartilhamento é facilitado.

Todo o processo de receptação de novas requisições será executado junto com uma avaliação de *capacity planning*, mapeamento de dependências e ordenamento de prioridades.

Dentro de cada milestone como é base de um projeto ágil será executado uma reunião retrospectiva baseada em um período apenas de discussão, com o intuito de verificar o que foi não executado, o que foi executado, mas poderia ser melhor e onde melhorar a integração do time.

#### **Requisições de desenvolvimento, correções e escalabilidade:**

Todo o processo de desenvolvimento quando existe a necessidade de atualização ou expansão da base irá se voltar para a importância da virtualização, pois uma atualização do código da aplicação as vezes poderá ter modificações no sistema operacional, e para não gerar sobrecarga por pequenas mudanças o recurso de se ter um SDDC se dá importante para controlar a performance com reservas para expansão, ajustando utilização de armazenamento e poder de processamento em casos da necessidade de expansão do negócio.

#### **Ansible playbooks, o início:**

Com a abordagem que poderá ser reutilizada, repetida, de configuração simples e de polivalência entre os sistemas, a utilização na execução das tarefas via Ansible vai no objetivo de facilitar na construção dos playbooks e manter a distribuição sob controle sem a necessidade de *agents*.

Com o playbook pronto será possível confirmar um determinado push será feito com sucesso em um sistema. Os playbooks são uma forma de declarar configurações, orquestrar passos de qualquer ordem manual definida e iniciar diversas tarefas de forma síncrona ou assíncrona no ambiente.

A base do playbook será toda expressada em YAML onde cada tarefa irá chamar um modulo Ansible diferente, dependendo do sistema operacional o sistema é classificado como idempotente, onde a performance é a chave mestra, onde uma tarefa já executada não será feita novamente com o último status armazenado, sem a necessidade também de um servidor de gerenciamento (*on premises*).

Segue exemplos .yml editado na ferramenta Azure DevOPs para criação de serviços:

The screenshot shows the Azure DevOps interface with two code editors side-by-side.

**Left Editor (azure-pipelines.yml):**

```
1 # Starter pipeline
2 # Start with a minimal pipeline that you can customize to build and deploy your code.
3 # Add steps that build, run tests, deploy, and more:
4 # https://aka.ms/yaml
5
6 trigger:
7 - master
8
9 pool:
10  vmImage: 'ubuntu-latest'
11 demands: maven
12
13 #Your build pipeline references an undefined variable named 'Parameters.mavenPOMFile'. Create
14
15 steps:
16 - task: Maven@3
17  displayName: 'Maven Build'
18  inputs:
19    mavenPomFile: 'pom.xml'
20
21 - task: CopyFiles@2
22  displayName: 'Copy Files to: $(build.artifactstagingdirectory)'
23  inputs:
24    SourceFolder: '$(build.sourcesdirectory)'
25    Contents: |
26      **/*.war
27      **/*.yaml
28    TargetFolder: '$(build.artifactstagingdirectory)'
29
30 - task: PublishBuildArtifacts@1
31  displayName: 'Publish Artifact: drop'
32  inputs:
33    PathToPublish: '$(build.artifactstagingdirectory)'
```

**Right Editor (webapp.yml):**

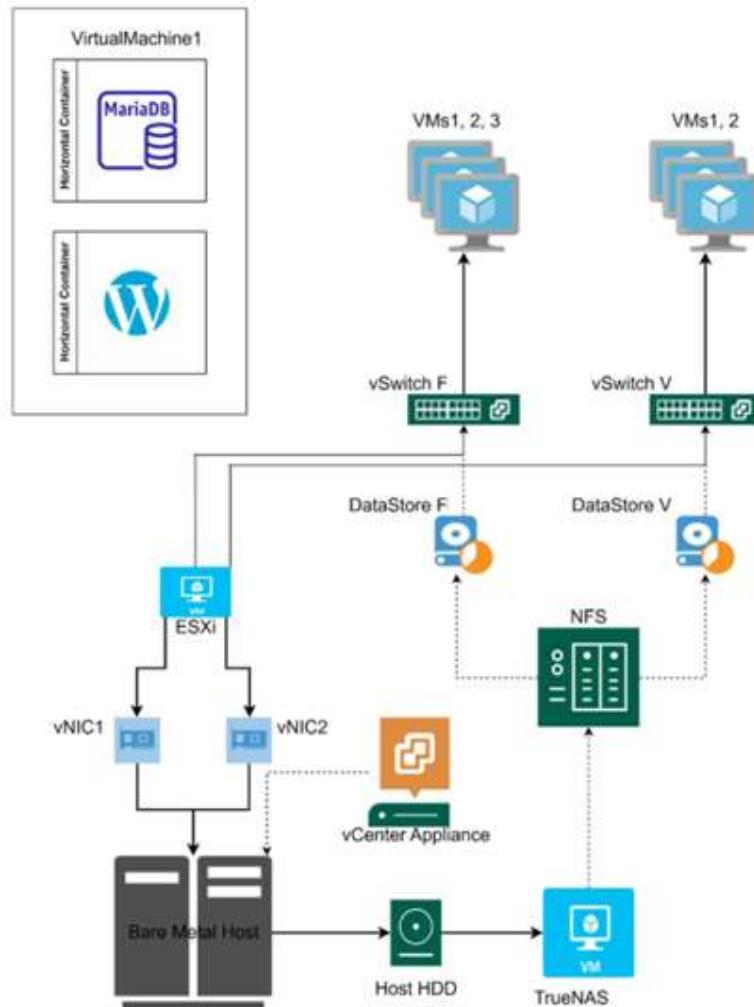
```
1 ...
2
3
4 - hosts: localhost
5
6
7 vars:
8   resource_group: __resourceGroupName__
9   location: __location__
10  webapp_name: __webappName__
11  mysqlserver_name: __mysqlServerName__
12  mysqlDb_name: __mysqlDbName__
13  admin_username: __mysqlAdmin__
14  admin_password: __mysqlAdminSecret__
15
16 tasks:
17 - name: Create a resource group
18   azure_rm_resourcegroup:
19     name: "{{ resource_group }}"
20     location: "{{ location }}"
21
22 - name: Create App Service on Linux with Java Runtime
23   azure_rm_webapp:
24     resource_group: "{{ resource_group }}"
25     name: "{{ webapp_name }}"
26     plan:
27       resource_group: "{{ resource_group }}"
28       name: myappplan
29       is_linux: false
30       sku: S1
31       number_of_workers: 1
32     frameworks:
33       - name: "java"
34         version: "1.8"
35         settings:
36           java_container: tomcat
37           java_container_version: "9.0"
```

### Infraestrutura atual:

Hoje a aplicação é executada exclusivamente via VCSA on-premises com as configurações onde o Host que executa a aplicação atualmente está dimensionado com:

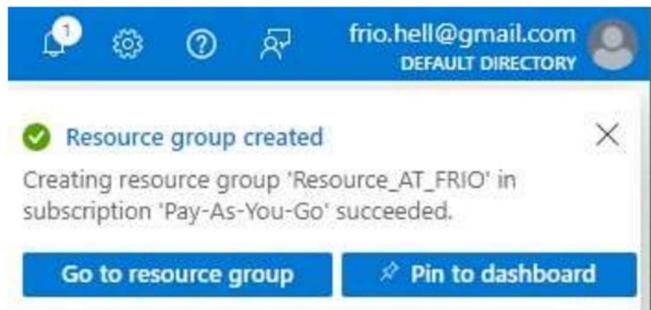
- -16 cores com tecnologia Hyper Threading.
- -64GB de RAM pois o operacional será simultâneo.

Onde a hospedagem inicial para separar a infra será configurado 1 host ESXi para suportar as virtualizações. Como IOPS no momento não é importante o armazenamento está por conta de 1 host TrueNAS prestando o serviço de compartilhamento via NFS, com separação de data stores e vSwitches distintos para segregação do tráfego de rede, garantindo todas as camadas lógicas.



## Esboço de implantação na nuvem (azure):

Como o processo final é para implementação na nuvem em IaaS onde a virtualização, servidores, storage e redes são de responsabilidade da pública, todo o projeto foi modificado para a infraestrutura abaixo com a extinção de alguns serviços mencionados que já estão incorporados nativamente, como armazenamento e a não mais necessidade de um host bare metal centralizado:



\*já criada previamente em formato PAYG pois não houve bônus para o ambiente Azure.

A configurações da infra na nuvem ficaram de acordo:

Os servidores de Active Directory e APP/RDS tem o hardware a seguir: 2vCPU, 8GB RAM, SSD Standard 128GB – 500IOPS – Windows 2016

O desktop cliente para acesso remoto tem o hardware a seguir: 2vCPU, 4GB RAM, HDD LRS 128GB – 350 IOPS – Windows 10

## TP2

### Hosts e instalação Ansible;

No host o ansible foi instalado:

```
[frio@ansihost ~]$ sudo yum install epel-release -y
[sudo] password for frio:
Last metadata expiration check: 1:50:43 ago on Sun 19 Mar 2023 04:50:12 PM -03.
Dependencies resolved.
=====
| Package           | Architecture | Version      | Repository | Size |
|=====|
| Installing:      |             |              |            |       |
|   epel-release     | noarch      | 8-11.el8    | extras     | 24 k  |
|=====|
Transaction Summary
=====
Install 1 Package
=====
Total download size: 24 k
Installed size: 24 k
Downloading Packages:
epel-release-8-11.el8.noarch.rpm
=====
Total
CentOS Stream 8 - Extras
Importing GPG key 0x8483C650:
Userid: CentOS-8 Official Signing Key <security@centos.org>
Fingerprint: 900B 70FA E1D7 C622 7FB6 48B2 05A5 55A3 84B3 C650
From : /etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
Key imported successfully
Running transaction test
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
  Installing : epel-release-8-11.el8.noarch
  Running scriptlet: epel-release-8-11.el8.noarch
  Verifying   : epel-release-8-11.el8.noarch
=====
Installed:
  epel-release-8-11.el8.noarch
```

```
[frio@el8host ~]$ sudo yum install ansible -y
Examining Packages for Enterprise Linux 8 - x86_64
Extra Packages for Enterprise Linux Modular 8 - x86_64
Last metadata expiration check: 0:00:01 ago on Sun 19 Mar 2023 06:41:51 PM -03.
Dependencies resolved.
=====
| Package           | Architecture | Version      | Repository | Size |
|=====|
| Installing:      |             |              |            |       |
|   ansible         | noarch      | 6.3.0-2.el8.1 | epel       | 35 M  |
|=====|
Installing dependencies:
ansible-core      x86_64      2.14.2-2.el8
ansible          x86_64      2.30.1-1.el8
python3          x86_64      3.9.16-1.module_el8.8.0+1243+5fsale61
python3-cffi       x86_64      1.14.3-2.module_el8.7.0+1178+0ba51308
python3-cryptography x86_64      3.1.3-2.module_el8.7.0+974+dc19af12
python3-pip        x86_64      21.0.3-2.module_el8.5.0+894+dc19af12
python3-lbs        x86_64      3.9.16-1.module_el8.8.0+1243+5fsale61
python3-pip-wheel  noarch      20.2.4-7.module_el8.7.0+1213+291b6551
python3-setuptools x86_64      3.11.0-1.module_el8.7.0+1178+dc19af12
python3-pyparser   noarch      2.20-3.module_el8.5.0+738+dc19af12
python3-pyyaml     x86_64      5.4.1-1.module_el8.6.0+930+10acc06f
python3-setuptools-wheel x86_64      50.3.2-4.module_el8.6.0+930+10acc06f
python3-setuptools-x  noarch      1.14.3-2.module_el8.5.0+738+dc19af12
sshpass          x86_64      1.09-4.el8
Installing weak dependencies:
python3-jmespath   noarch      0.9.0-11.el8
python3-setuptools noarch      50.3.2-4.module_el8.6.0+930+10acc06f
Enabling module streams:
python39
=====
Transaction Summary
=====
Install 18 Packages
=====
Total download size: 62 M
Installed size: 349 M
Dependencies resolved.
(1/18): python3-jmespath-0.9.0-11.el8.noarch.rpm
(2/18): python39-3.9.16-1.module_el8.8.0+1243+5fsale61.x86_64.rpm
=====
439 kB/s | 45 kB  00:00
1.5 MB/s | 33 kB  00:00
1.3 kB/s | 34 kB  00:00
```

## Corte de tela:

```
[root@frio /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-8
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : sshpass-1.09-4.el8.x86_64
1/18
Installing : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
2/18
Installing : python39-pip-wheel-20.2.4-7.module_el8.7.0+1213+291b551.noarch
3/18
Installing : python39-libffi-3.9.16-1.module_el8.8.0+1243+5fs1ae61.x86_64
4/18
Installing : python39-cffi-1.14.3-2.module_el8.8.0+1243+5fs1ae61.x86_64
5/18
Running scriptlet: python39-3.9.16-1.module_el8.8.0+1243+5fs1ae61.x86_64
6/18
Installing : python39-setuptools-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
7/18
Running scriptlet: python39-setuptools-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
8/18
Installing : python39-pip-20.2.4-7.module_el8.7.0+1213+291b551.noarch
9/18
Installing : python39-setuptools-50.3.2-4.module_el8.7.0+1213+291b551.noarch
10/18
Installing : python39-ply-3.11-10.module_el8.8.0+738+dc19af12.x86_64
11/18
Installing : python39-cpparser-2.14.3-2.module_el8.7.0+1178+0ba51308.x86_64
12/18
Installing : python39-pygments-2.10.0-11.module_el8.8.0+930+10acc06f.x86_64
13/18
Installing : python39-pyyaml-5.4-1.1.module_el8.8.0+930+10acc06f.x86_64
14/18
Installing : python3-jmespath-0.9.0-11.el8.noarch
15/18
Installing : git-core-2.39.1-1.el8.x86_64
16/18
Installing : ansible-6.3.0-2.el8.1.noarch
17/18
Installing : ansible-6.3.0-2.el8.1.noarch
18/18
Running scriptlet: ansible-6.3.0-2.el8.1.noarch
19/18
Verifying : ansible-6.3.0-2.el8.1.noarch
20/18
Verifying : git-core-2.39.1-1.el8.x86_64
21/18
Verifying : python3-jmespath-0.9.0-11.el8.noarch
22/18
Verifying : python39-3.9.16-1.module_el8.8.0+1243+5fs1ae61.x86_64
23/18
Verifying : python39-cffi-1.14.3-2.module_el8.8.0+1243+5fs1ae61.x86_64
24/18
Verifying : python39-cryptography-3.3.1-2.module_el8.5.0+738+dc19af12.x86_64
25/18
Verifying : python39-idna-2.10-3.module_el8.5.0+738+dc19af12.noarch
26/18
Verifying : python39-lbs-3.9.16-1.module_el8.8.0+1243+5fs1ae61.x86_64
27/18
Verifying : python39-setuptools-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
28/18
Verifying : python39-setuptools-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
29/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
30/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
31/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
32/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
33/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
34/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
35/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
36/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
37/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
38/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
39/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
40/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
41/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
42/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
43/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
44/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
45/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
46/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
47/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
48/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
49/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
50/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
51/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
52/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
53/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
54/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
55/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
56/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
57/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
58/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
59/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
60/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
61/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
62/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
63/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
64/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
65/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
66/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
67/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
68/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
69/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
70/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
71/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
72/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
73/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
74/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
75/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
76/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
77/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
78/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
79/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
80/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
81/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
82/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
83/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
84/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
85/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
86/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
87/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
88/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
89/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
90/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
91/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
92/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
93/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
94/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
95/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
96/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
97/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
98/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
99/18
Verifying : python39-setuptools-wheel-50.3.2-4.module_el8.6.0+930+10acc06f.noarch
100/18
Complete!
[frio@ansihost ~]$
```

E posteriormente verificado o sucesso da instalação:

```
Complete!
[frio@ansihost ~]$ ansible --version
ansible [core 2.14.2]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/frio/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible
  ansible collection location = /home/frio/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.9.16 (main, Jan 17 2023, 18:53:15) [GCC 8.5.0 20210514 (Red Hat 8.5.0-18)] (/usr/bin/python3.9)
  jinja version = 3.1.2
  libyaml = True
[frio@ansihost ~]$
```

## Criação e teste de conectividade;

Seguindo as boas práticas do manual do ansible foi feito um teste de conectividade logo após a criação do arquivo de inventario, coloquei 2 alvos para testar múltiplos acessos e agrupamento de hosts:

```
[root@ansihost AnsiblePBInfnetworkfrio]# vi inventory.txt
[root@ansihost AnsiblePBInfnetworkfrio]# cat inventory.txt
# Servidores Infra Infnet
target1 ansible_host=192.168.79.141 ansible_user=root ansible_ssh_pass=password1982
target2 ansible_host=192.168.79.142 ansible_user=root ansible_ssh_pass=password1982

#Agrupamento
[wordpressinfnet]
target1
target2
[root@ansihost AnsiblePBInfnetworkfrio]# vi /etc/ansible/ansible.cfg
[root@ansihost AnsiblePBInfnetworkfrio]# cat /etc/ansible/ansible.cfg | grep "host_key"
host_key_checking=False
;host_key_auto_add=False
;host_key_auto_add=
;host_key_checking=True
;record_host_keys=True
;host_key_checking=True
[root@ansihost AnsiblePBInfnetworkfrio]# ansible all --list-hosts -i inventory.txt
  hosts (2):
    target1
    target2
[root@ansihost AnsiblePBInfnetworkfrio]# ansible all -m ping -i inventory.txt
target1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
target2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
[root@ansihost AnsiblePBInfnetworkfrio]# ansible wordpressinfnet -m ping -i inventory.txt
target1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
target2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
[root@ansihost AnsiblePBInfnetworkfrio]#
```

## Arvore de arquivos e variáveis;

Foi criado localmente no host a arvore dos arquivos:

```
-rw-rw-r--, 1 trio trio 242 Mar 19 19:44 inventory.txt
[root@ansihost AnsiblePBInfnetworkfrio]# mkdir wordpress
[root@ansihost AnsiblePBInfnetworkfrio]# cd wordpress/
[root@ansihost wordpress]# mkdir roles vars files
[root@ansihost wordpress]# cd roles/
[root@ansihost roles]# mkdir -p apache/tasks php/tasks mysql/tasks wordpress/tasks firewall/tasks
[root@ansihost roles]# cd ..
[root@ansihost wordpress]# cd ..
[root@ansihost AnsiblePBInfnetworkfrio]# tree wordpress/
wordpress/
├── files
└── roles
    ├── apache
    │   └── tasks
    ├── firewall
    │   └── tasks
    ├── mysql
    │   └── tasks
    ├── php
    │   └── tasks
    └── wordpress
        └── tasks
vars
13 directories, 0 files
[root@ansihost AnsiblePBInfnetworkfrio]#
```

E o arquivo de variáveis do ambiente com usuários, senhas, base de dados e o que precisa ser instalado e caso precise de modificação concentrar as informações em um único arquivo:

```
[root@ansihost vars]# cat default.yml
#PHP ajustes
php_modules: [ 'php', 'php-curl', 'php-gd', 'php-mbstring', 'php-xml', 'php-xmlrpc', 'php-soap', 'php-intl', 'php-zip' ]

#MySQL ajustes
mysql_root_password: "password1982"
mysql_db: "wpdb"
mysql_user: "wpuser"
mysql_password: "password1982"

#HTTP FQDN e porta
http_host: "wp.infnet.com"
http_conf: "wp.infnet.com.conf"
http_port: "80"
```

## Templates;

Foram criados os templates do apache, com a função de instalar, iniciar o serviço pos boot, a criação de um diretório raiz e a configuração do host alvo junto do arquivo httpd.conf:

```
[root@ansihost tasks]# cat main.yml
- name: Install HTTP Packages
  yum: name=httpd update_cache=yes state=latest

- name: Start httpd service
  systemd: name=httpd state=started enabled=yes

- name: Apache Root
  file:
    path: "/var/www/{{ http_host }}"
    state: directory
    owner: "apache"
    group: "apache"
    mode: '0755'

- name: Apache VirtualHost
  template:
    src: "files/httpd.conf"
    dest: "/etc/httpd/conf.d/{{ http_conf }}"
    owner: root
    group: root
    mode: u=rw,g=r,o=r
[root@ansihost tasks]# [root@ansihost files]# cat httpd.conf
<VirtualHost *:{{ http_port }}>
  ServerAdmin webmaster@localhost
  ServerName {{ http_host }}
  ServerAlias www.{{ http_host }}
  DocumentRoot /var/www/{{ http_host }}/wordpress
  ErrorLog /var/log/httpd/error.log
  CustomLog /var/log/httpd/access.log combined

  <Directory /var/www/{{ http_host }}/wordpress>
    Options Indexes FollowSymLinks
    AllowOverride all
    Require all granted
  </Directory>
</VirtualHost>
[root@ansihost files]# [root@ansihost ~]#
```

## MySQL e Wordpress template:

Foi feito o template das configurações para o MySQL e Wordpress:

```
[root@ansihost tasks]# cat main.yml
# MySQL Configuration
- name: Install MySQL Packages
  yum: name={{ item }} update_cache=yes state=latest
  loop: [ 'mysql-server', 'php-mysqlnd', 'python3-PyMySQL' ]

- name: Start mysqld service
  systemd: name=mysql state=started enabled=yes

- name: Set MySQL root Password
  mysql_user:
    login_host: 'localhost'
    login_user: 'root'
    login_password: ''
    name: 'root'
    password: '{{ mysql_root_password }}'
    state: present

- name: Creates database for WordPress
  mysql_db:
    name: "{{ mysql_db }}"
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"

- name: Create MySQL user for WordPress
  mysql_user:
    name: "{{ mysql_user }}"
    password: "{{ mysql_password }}"
    priv: "{{ mysql_db }}.*:ALL"
    state: present
    login_user: root
    login_password: "{{ mysql_root_password }}"

[root@ansihost tasks]#
```

```
[root@ansihost tasks]# cat main.yml
# WordPress Configuration

- name: Download and unpack latest WordPress
  unarchive:
    src: https://wordpress.org/latest.tar.gz
    dest: "/var/www/{{ http_host }}"
    remote_src: yes
    creates: "/var/www/{{ http_host }}/wordpress"

- name: Set ownership
  file:
    path: "/var/www/{{ http_host }}"
    state: directory
    recurse: yes
    owner: apache
    group: apache

- name: Set permissions for directories
  shell: "/usr/bin/find /var/www/{{ http_host }}/wordpress/ -type d -exec chmod 750 {} \\;"

- name: Set permissions for files
  shell: "/usr/bin/find /var/www/{{ http_host }}/wordpress/ -type f -exec chmod 640 {} \\;"

- name: Copy sample config file
  command: mv /var/www/{{ http_host }}/wordpress/wp-config-sample.php /var/www/{{ http_host }}/wordpress/wp-config.php
  creates=/var/www/{{ http_host }}/wordpress/wp-config.php
  become: yes

- name: Update WordPress config file
  lineinfile:
    path: "/var/www/{{ http_host }}/wordpress/wp-config.php"
    regexp: "{{item.regexp}}"
    line: "{{item.line}}"
  with_items:
    - {'regexp': "define\\( 'DB_NAME', '(.+)' );", 'line': "define( 'DB_NAME', '{{mysql_db}}' );"}
    - {'regexp': "define\\( 'DB_USER', '(.+)' );", 'line': "define( 'DB_USER', '{{mysql_user}}' );"}
    - {'regexp': "define\\( 'DB_PASSWORD', '(.+)' );", 'line': "define( 'DB_PASSWORD', '{{mysql_password}}' );"}

- name: Restart httpd service
  systemd: name=httpd state=restarted
  become: yes
```

## PHP;

A instalação do PHP apesar de estar nas tasks foi feita externamente por um erro de chave do repositório.

```
[root@ansihost tasks]# cat main.yml
- name: Install PHP Remi Repository
  yum: name=http://rpms.remirepo.net/enterprise/remi-release-8.rpm update_cache=yes state=latest

- name: Enable PHP Remi Repository
  command: dnf module reset php -y
  command: dnf module enable php:remi-7.4 -y

- name: Install PHP Extensions
  yum: name={{ item }} update_cache=yes state=latest
  loop: "{{ php_modules }}"
[root@ansihost tasks]#
```

## Firewall (SELinux);

A instalação do PHP apesar de estar nas tasks foi feita externamente por um erro de chave.

```
[root@ansihost tasks]# cat main.yml
# Firewall Configuration
  - name: Disable SELinux Permanently (Reboot Required)
    selinux: state=disabled

  - name: Disable SELinux Without Reboot
    command: /sbin/setenforce 0

  - name: Configure Firewall
    firewalld: zone=public service=http permanent=yes state=enabled

  - name: Reload Firewall
    systemd: name=firewalld state=reloaded
[root@ansihost tasks]#
```

## Execução e problemas;

A execução do playbook e as telas a seguir:

```
[root@ansihost wordpress]# ansible-playbook playbook.yml -i inventory.txt
[WARNING]: While constructing a mapping from /home/frio/AnsiblePBInnetfrio/wordpress/roles/php/tasks/main.yml, line 4,
column 3, found a duplicate dict key (command). Using last defined value only.

PLAY [target1] ****
TASK [apache : Install HTTP Packages] ****
changed: [target1]

TASK [apache : Start httpd service] ****
changed: [target1]

TASK [apache : Apache Root] ****
changed: [target1]

TASK [apache : Apache VirtualHost] ****
changed: [target1]

TASK [php : Install PHP Remi Repository] ****
fatal: [target1]: FAILED! => {"ansible_facts": {"pkg_mgr": "dnf"}, "changed": false, "msg": "Failed to validate GPG signature for remi-release-8.7-2.el8.remi.noarch: Public key for remi-release-8z1_9gb17.rpm is not installed"}

PLAY RECAP ****
target1 : ok=4    changed=4    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

```
[root@ansihost wordpress]# ansible-playbook playbook.yml -i inventory.txt
PLAY [target1] ****
TASK [Apache : Install HTTP Packages] ****
ok: [target1]
TASK [Apache : Start httpd service] ****
ok: [target1]
TASK [Apache : Apache Root] ****
ok: [target1]
TASK [Apache : Apache VirtualHost] ****
ok: [target1]
TASK [mysql : Install MySQL Packages] ****
changed: [target1] => (item=mysql-server)
changed: [target1] => (item=php-mysqlnd)
changed: [target1] => (item=python3-PyMySQL)

TASK [mysql : Start mysqld service] ****
changed: [target1]

TASK [mysql : Set MySQL root Password] ****
changed: [target1]

TASK [mysql : Creates database for WordPress] ****
changed: [target1]

TASK [mysql : Create MySQL user for WordPress] ****
changed: [target1]

TASK [wordpress : Download and unpack latest WordPress] ****
changed: [target1]

TASK [wordpress : Set ownership] ****
changed: [target1]

TASK [wordpress : Set permissions for directories] ****
changed: [target1]

TASK [wordpress : Set permissions for files] ****
changed: [target1]

TASK [wordpress : Copy sample config file] ****
changed: [target1]

TASK [wordpress : Update WordPress config file] ****
changed: [target1] => (item={'regexp': "define\\(( 'DB_NAME', '(.)+' \\\\);, 'line': 'define( 'DB_NAME', 'wpdb' );"})
changed: [target1] => (item={'regexp': "define\\(( 'DB_USER', '(.)+' \\\\);, 'line': 'define( 'DB_USER', 'wpuser' );"})
changed: [target1] => (item={'regexp': "define\\(( 'DB_PASSWORD', '(.)+' \\\\);, 'line': 'define( 'DB_PASSWORD', 'password1982 ' );"})

TASK [wordpress : Restart httpd service] ****
changed: [target1]

TASK [firewall : Disable SELinux Permanently (Reboot Required)] ****
[WARNING]: SELinux state temporarily changed from 'enforcing' to 'permissive'. State change will take effect next reboot.
changed: [target1]
```

PHP foi removido da task e a continuidade da tela mostrando o sucesso do playbook:

```
TASK [firewall : Disable SELinux Permanently (Reboot Required)] ****
[WARNING]: SELinux state temporarily changed from 'enforcing' to 'permissive'. State change will take effect next reboot.
changed: [target1]

TASK [firewall : Disable SELinux Without Reboot] ****
changed: [target1]

TASK [firewall : Configure Firewall] ****
changed: [target1]

TASK [firewall : Reload Firewall] ****
changed: [target1]

PLAY RECAP ****
target1 : ok=20  changed=16  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

[root@ansihost wordpress]#
```

## PHP;

A instalação do PHP apesar de estar nas tasks foi feita externamente por um erro de chave e a instalação foi feita de forma manual aqui validando o serviço.:

```
Running scriptlet: php-common-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64 13/13
Running scriptlet: php-fpm-8.0.28-1.el8.remi.x86_64 13/13
Verifying : nginx-filesystem-1:1.14.1-9.module_el8.0.0+1060+3ab382d3.noarch 1/13
Verifying : php-fpm-8.0.28-1.el8.remi.x86_64 2/13
Verifying : php-mbstring-8.0.28-1.el8.remi.x86_64 3/13
Verifying : php-xml-8.0.28-1.el8.remi.x86_64 4/13
Verifying : oniguruma5php-6.9.8-1.el8.remi.x86_64 5/13
Verifying : php-cli-8.0.28-1.el8.remi.x86_64 6/13
Verifying : php-fpm-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64 7/13
Verifying : php-common-8.0.28-1.el8.remi.x86_64 8/13
Verifying : php-common-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64 9/13
Verifying : php-mysqlnd-8.0.28-1.el8.remi.x86_64 10/13
Verifying : php-mysqlnd-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64 11/13
Verifying : php-pdo-8.0.28-1.el8.remi.x86_64 12/13
Verifying : php-pdo-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64 13/13

Upgraded:
  php-cli-8.0.28-1.el8.remi.x86_64      php-common-8.0.28-1.el8.remi.x86_64      php-mysqlnd-8.0.28-1.el8.remi.x86_64
  php-pdo-8.0.28-1.el8.remi.x86_64

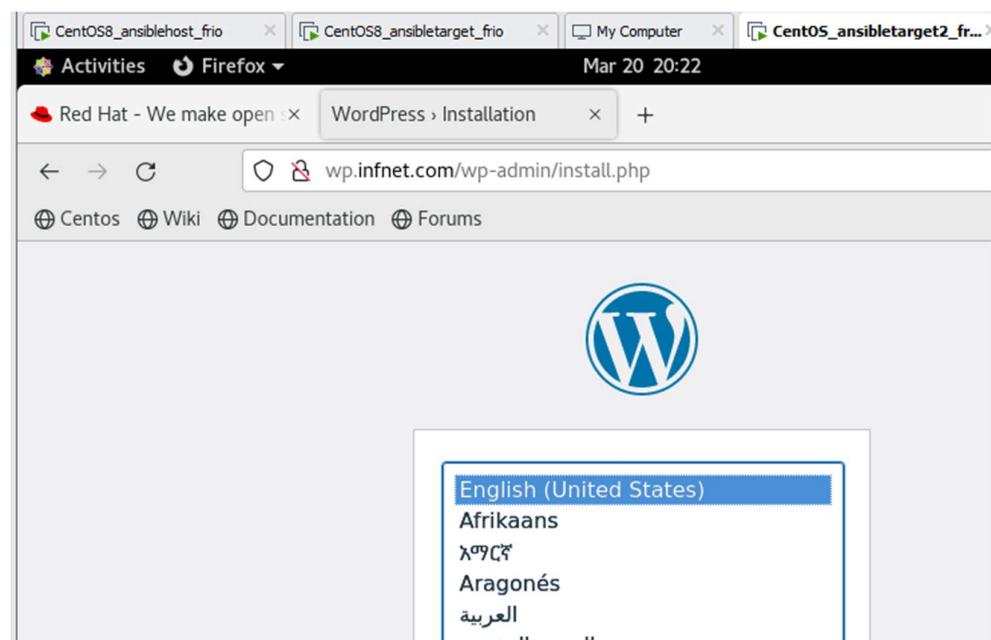
Installed:
  nginx-filesystem-1:1.14.1-9.module_el8.0.0+1060+3ab382d3.noarch      oniguruma5php-6.9.8-1.el8.remi.x86_64
  php-fpm-8.0.28-1.el8.remi.x86_64      php-mbstring-8.0.28-1.el8.remi.x86_64
  php-xml-8.0.28-1.el8.remi.x86_64

Complete!
[root@ansitarget /]# php -v
PHP 8.0.28 (cli) (built: Feb 14 2023 11:05:44) ( NTS gcc x86_64 )
Copyright (c) The PHP Group
Zend Engine v4.0.28, Copyright (c) Zend Technologies
[root@ansitarget /]#
```

## Validação do serviço wordpress localhost (GUI):

Com alterações no host para seguir o site e ficar uma instalação limpa,

e configurada linguagem, título e senha administrativa :



Activities Firefox Mar 20 20:23

Red Hat - We make open WordPress › Installation +

← → ⌂ ⌂ wp.infnet.com/wp-admin/install.php?step=1

Centos Wiki Documentation Forums

## Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and on your way to using the most extendable and powerful personal publishing platform in the world.

### Information needed

Please provide the following information. Do not worry, you can always change these settings later.

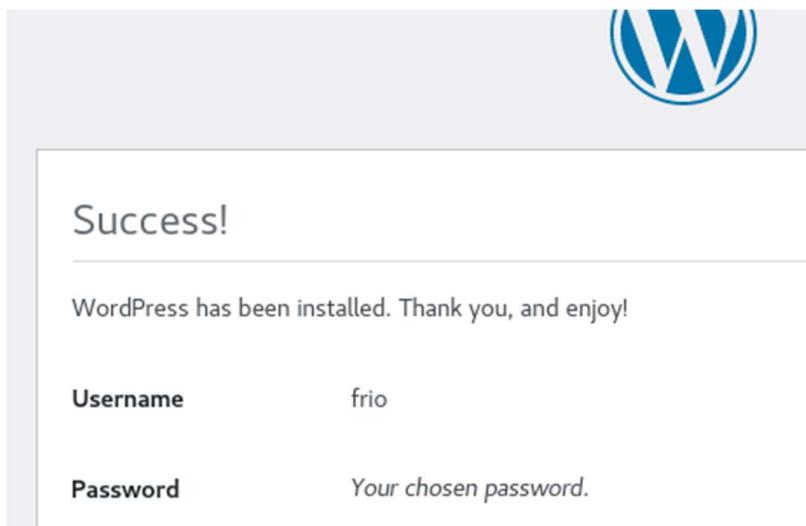
Site Title	Projeto Bloco Felipe Rio
Username	frio
Password	Admin01! Very weak
Confirm Password	<input checked="" type="checkbox"/> Confirm use of weak password
Your Email	felipe.rio@al.infnet.edu.br

Usetnames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

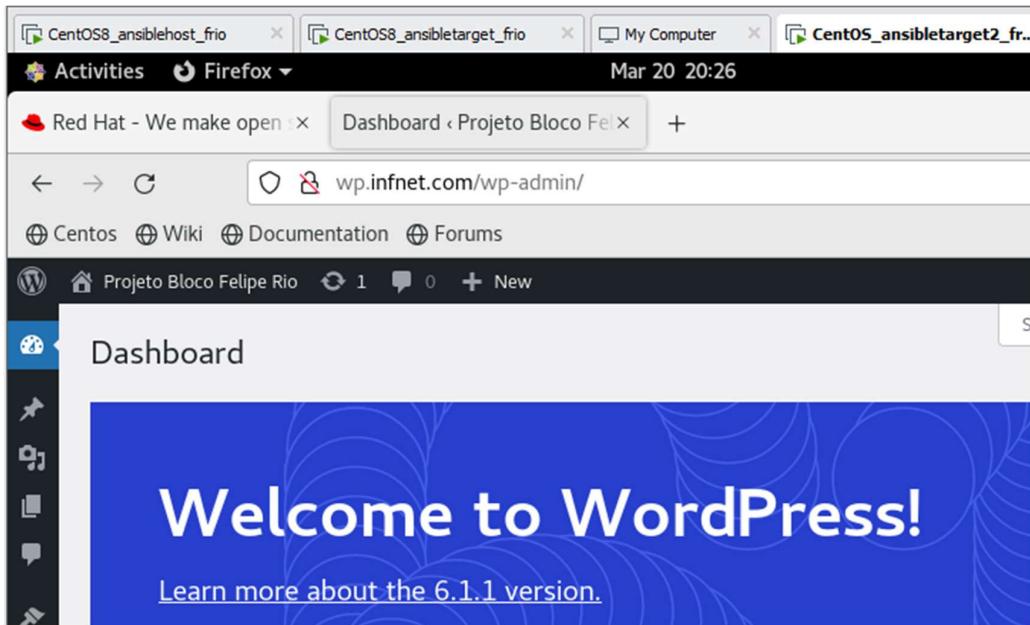
Important: You will need this password to log in. Please store it in a secure location.

Double-check your email address before continuing.

Posterior a configuração ativa:



Validando o acesso a página de administração para a criação e validação de um post:



Acesso externo para validar o post:

A screenshot of a web browser window titled "Projeto Bloco Felipe Rio". The address bar shows "192.168.79.141". The main content area displays a blog post titled "Hello world!". The post content is "Welcome to WordPress. This is your first post. Edit or delete it, then start writing!". The date of the post is "March 20, 2023".

## TP3

### Criação do profile;

A criação de um apelido e conta são voltadas para o Infnet com o endereço público.

Onde foram criados alguns arquivos manualmente para testar a plataforma web e alterado o README.md inicialmente.

The screenshot shows a GitHub repository page for 'felipe-infnet/projetoAIA'. The repository is public, has 1 branch, and 0 tags. The 'Code' tab is selected. The commit history shows four commits from 3 hours ago:

File	Message	Date	Commits
LICENSE	Initial commit	3 hours ago	last week
README.md	Update README.md	3 hours ago	last week
inventory.txt	Add files via upload	3 hours ago	3 hours ago
playbook.yml	Add files via upload	3 hours ago	3 hours ago

The README.md file content is as follows:

```
projetoAIA2023

Repositório criado com o objetivo de controlar o ambiente do projeto de bloco. O projeto é público e será usado como exemplo para a graduação de Gestão em TI da Fundação Infnet.

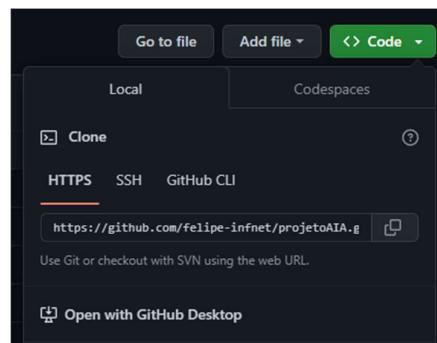
Instalação automatizada do Wordpress via ansible
```

## Configuração AnsibleHost:

Como toda a árvore foi criada localmente o servidor de host precisa ser configurado para se conectar ao serviço e posteriormente validada a configuração do .gitconfig:

```
[root@ansihost wordpress]# git config --global user.name "felipe-infnet"
[root@ansihost wordpress]# git config --global user.email "felipe.rio@al.infnet.edu.br"
[root@ansihost wordpress]# git config --list
user.name=felipe-infnet
user.email=felipe.rio@al.infnet.edu.br
[root@ansihost wordpress]# cat ~/.gitconfig
[user]
    name = felipe-infnet
    email = felipe.rio@al.infnet.edu.br
[root@ansihost wordpress]#
```

E em paralelo feito a cópia do endereço de clone no webservice:

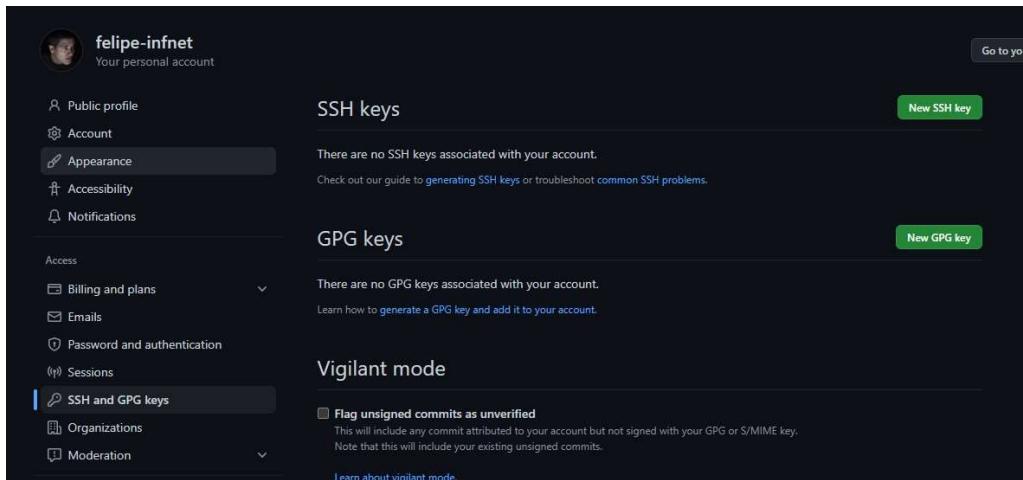


Foi verificado que o processo de autenticação por senha padrão estava bloqueado e seguindo o manual o serviço de SSH precisa ser configurado:

```
create mode 100644 vars/default.yml
[root@ansihost wordpress]# git remote add origin https://github.com/felipe-infnet/projetoAIA.git
[root@ansihost wordpress]# git push -u origin master
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-ht
tps-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/felipe-infnet/projetoAIA.git/'
[root@ansihost wordpress]#
```

## SSH:

Para se iniciar a configuração no menu settings web existe um campo de registro das chaves:



## SSH Configuração:

Seguindo o guia foi configurado o SSH no servidor gerando uma nova chave e verificado o serviço, e assim adicionando a chave em seguida:

```
[root@ansihost wordpress]# ssh-keygen -t ed25519 -C "felipe.rio@al.infnet.edu.br"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519.
Your public key has been saved in /root/.ssh/id_ed25519.pub.
The key's randomart image is:
+--[ED25519 256]--+
|%+o . .
|BB.. o = .
|++= o + = .
|++ o * + o
|E o =o= S o o
| +oo. o .
| . . o .
| ..o. .
| .o.....
+---[SHA256]----+
[root@ansihost wordpress]# eval "$(ssh-agent -s)"
Agent pid 56757
[root@ansihost wordpress]# ssh-add ~/.ssh/id_ed25519
Identity added: /root/.ssh/id_ed25519 (felipe.rio@al.infnet.edu.br)
```

A chave precisa ser copiada para o webservice:

```
[root@ansihost wordpress]# cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTExAAAINTWRcJkRM8MPBHxvhjhGuJoKoZNgMvmmhm9XP099RP/ felipe.rio@al.infnet.edu.br
[root@ansihost wordpress]# git clone git@github.com:repository-0:felipe-infnet/projetoAIA.git
Cloning into 'projetoAIA'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 29 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (29/29), 18.18 KiB | 18.18 MiB/s, done.
Resolving deltas: 100% (6/6), done.
[root@ansihost wordpress]#
```

E posteriormente criado um elemento:

The screenshot shows the GitHub 'SSH keys / Add new' page. On the left, there's a sidebar with account settings like Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, SSH and GPG keys (which is selected), Organizations, and Moderation. The main area has fields for Title ('AnsiblehostKey') and Key type ('Authentication Key'). A large text input box contains the copied SSH key:   
ssh-rsa  
AAAAAB3NzaC1yc2EAAAQABgQDVK29pj8i7WFTIVo2h8ZPtdirRSMGAI2/S3Qbbi56WTyRByoI4NmNm7DZ8zcfVcmjnhsS+2FexQqQOcDt+6IZTppe0AR/vT4c58iyGZsmotTjYvRKWjbf3KsrCYt/5xluMRgN78GMmYW+ECODSw00/O9dj6vvIZ324Vi7kaRTTTdmHdDLBc+BfK6oidU6hpS/lo8E9gg6Qf/UX6lnKrmLwvCfCf1WVvfqqXgFIUKr/1Nwg6XzqtjRmsabZD9WC9WdO2SJAtfSzwdTPq9L1FbSpQc2zKRxbWNNU+z+BP/E9pdmd8qf9e+Fchtn1LkBYJ/WKUFV6VLaknqspXvDFle6Xdw1F5ckDuerQtp25zenoMH0B9PYRnR6J0x6Kwt8+Bid88KHTxMzuEqRLBfyUBY+LSWfaY6ctNDI7OfslDs3jy4hr0ID+rITmLlyBExmvf3a+m21Cuvj5SBnIux/B8vEoqTojdhd5OsVwC/PX4SCU= root@ansihost.localdomain|

The screenshot shows the GitHub 'SSH Keys' page. It lists two authentication keys: 'AnsiblehostKey.141' and 'AnsiblehostKey'. Both keys were added on March 21, 2023, and have never been used. Each key entry includes a 'Delete' button. Below the keys, there's a note about generating SSH keys and troubleshooting common SSH problems. At the bottom, there's a 'New GPG key' button.

## Árvore de arquivos e commit:

No processo checando o status do git e comparando os diretórios:

```
remote: Total 29 (delta 0), reused 0 (delta 0), pack/reused 0
Receiving objects: 100% (29/29), 18.18 KiB | 18.18 MiB/s, done.
Resolving deltas: 100% (6/6), done.
[root@ansihost wordpress]# git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    projetoAIA/
nothing added to commit but untracked files present (use "git add" to track)
[root@ansihost wordpress]# ls -ltr
total 16
-rwxrwxrwx. 1 frio frio 242 Mar 19 19:44 inventory.txt
drwxrwxrwx. 7 root root 77 Mar 19 20:45 roles
drwxrwxrwx. 2 root root 25 Mar 20 17:47 vars
drwxrwxrwx. 2 root root 24 Mar 20 17:49 files
-rwxrwxrwx. 1 root root 166 Mar 20 18:33 playbook.yml
-rw-----. 1 root root 2610 Mar 21 18:07 felipe_ssh_keys.pub
-rw-r--r--. 1 root root 579 Mar 21 18:07 felipe_ssh_keys.pub.pub
drwxr-xr-x. 4 root root 104 Mar 21 19:09 projetoAIA
[root@ansihost wordpress]# cd projetoAIA/
[root@ansihost projetoAIA]# ls -ltr
total 48
-rw-r--r--. 1 root root 256 Mar 21 19:09 README.md
-rw-r--r--. 1 root root 166 Mar 21 19:09 playbook.yml
-rw-r--r--. 1 root root 35149 Mar 21 19:09 LICENSE
-rw-r--r--. 1 root root 242 Mar 21 19:09 inventory.txt
drwxr-xr-x. 2 root root 24 Mar 21 19:09 files
[root@ansihost projetoAIA]#
```

E iniciando o primeiro add/commit:

```
[root@ansihost projetoAIA]# git add -A
[root@ansihost projetoAIA]# git commit -m "primeiro commit Felipe"
>
[main 17c1612] primeiro commit Felipe
12 files changed, 147 insertions(+)
mode change 100644 => 100755 LICENSE
mode change 100644 => 100755 README.md
create mode 100644 felipe_ssh_keys.pub.pub
mode change 100644 => 100755 inventory.txt
mode change 100644 => 100755 playbook.yml
create mode 100644 roles/apache/tasks/main.yml
create mode 100644 roles/firewall/tasks/main.yml
create mode 100644 roles/mysql/tasks/main.yml
create mode 100644 roles/php/tasks/main.bak
create mode 100644 roles/php/tasks/main.yml
create mode 100644 roles/wordpress/tasks/main.yml
create mode 100644 vars/default.yml
[root@ansihost projetoAIA]#
```

## Push:

O processo de push sendo executado para validar as alterações no README.md:

```
[root@ansihost projetoAIA]# git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 312 bytes | 312.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:repository-0/felipe-infnet/projetoAIA.git
  431ee2e..c183d95 main -> main
branch 'main' set up to track 'origin/main'.
[root@ansihost projetoAIA]# cat README.md
# projetoAIA2023
Repositório criado com o objetivo de controlar o ambiente do projeto de bloco.
O projeto é público e será usado como exemplo para a graduação de Gestão em TI da Fundação Infnet.

Instalação automatizada do Wordpress via ansible
Primeiro pull! Agora Online!
[root@ansihost projetoAIA]#
```

The screenshot shows a GitHub repository page for 'felipe-infnet/projetoAIA'. The repository is public. The 'Code' tab is selected. At the top, it shows 'main' branch, '1 branch', and '0 tags'. Below that is a commit list from 'felipe-infnet' titled 'Primeiro commit agora online!' with commit ID 'c183d95' made 4 minutes ago. The commit details show changes to files like '.gitkeep', 'LICENSE', 'README.md', 'felipe\_ssh\_keys.pub', 'inventory.txt', and 'playbook.yml'. The 'README.md' file content is displayed below the commit list, showing the same text as the terminal output above. The right sidebar shows repository statistics: 12 commits, 0 issues, 0 pull requests, 0 actions, 0 projects, 0 wiki pages, 0 security vulnerabilities, 0 insights, and 0 settings.

```
felipe-infnet / projetoAIA [Public]
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
main 1 branch 0 tags Go to file Add file <> Code
felipe-infnet Primeiro commit agora online! c183d95 4 minutes ago 12 commits
files Delete .gitkeep 19 hours ago
roles primeiro commit Felipe 32 minutes ago
vars primeiro commit Felipe 32 minutes ago
LICENSE primeiro commit Felipe 32 minutes ago
README.md Primeiro commit agora online! 4 minutes ago
felipe_ssh_keys.pub pub primeiro commit Felipe 32 minutes ago
inventory.txt primeiro commit Felipe 32 minutes ago
playbook.yml primeiro commit Felipe 32 minutes ago
README.md
projetoAIA2023
Repositório criado com o objetivo de controlar o ambiente do projeto de bloco. O projeto é público e será usado como exemplo para a graduação de Gestão em TI da Fundação Infnet.
Instalação automatizada do Wordpress via ansible Primeiro pull! Agora Online!
```

Assim como o registro das alterações no projeto:

The screenshot shows a list of commits on the 'main' branch of a GitHub repository. The commits are as follows:

- validando tudo push (commit 938d576, 7 hours ago)
- push update infnet (commit f7b39b9, 8 hours ago)
- Primeiro commit agora online! (commit c183d95, 8 hours ago)
- Update README.md (commit 431ee2e, 9 hours ago)
- primeiro commit Felipe (commit 17c1612, 9 hours ago)

### Edicao online:

Efetuada uma edição online para verificar o processo:

The screenshot shows the 'README.md' file in the 'projetoAIA' repository. The content of the file is:

```
# projetoAIA2023
Repositório criado com o objetivo de controlar o ambiente do projeto.
O projeto é público e será usado como exemplo para a graduação de
Instalação automatizada do Wordpress via ansible
Primeiro pull! Agora Online!
Validacao de push online
```

The screenshot shows a GitHub repository page for 'felipe-infnet/projetoAIA'. The repository is public and has 1 branch and 0 tags. The main branch has 13 commits. The latest commit, 'f7b39b9 now', is a push update from Infnet. The commit history includes several files: 'files', 'roles', 'vars', 'LICENSE', 'README.md', 'felipe\_ssh\_keys.pub.pub', 'inventory.txt', and 'playbook.yml'. The README file contains the text: 'projetoAIA2023', 'Repositório criado com o objetivo de controlar o ambiente do projeto de bloco. O projeto é público e será usado como exemplo para a graduação de Gestão em TI da Fundação Infnet.', and 'Instalação automatizada do Wordpress via ansible Primeiro pull! Agora Online! Validação de push online'.

Executando um novo Push:

```
[root@ansihost projetoAIA]# git add -A
[root@ansihost projetoAIA]# git commit -m "validando tudo push"
[main 938d576] validando tudo push
 1 file changed, 1 insertion(+)
[root@ansihost projetoAIA]# git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 319 bytes | 319.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com-repository-0:felipe-infnet/projetoAIA.git
  f7b39b9..938d576  main -> main
branch 'main' set up to track 'origin/main'.
[root@ansihost projetoAIA]#
```

E validando as alterações online:

The screenshot shows a GitHub repository page for 'felipe-infnet/projetoAIA' (Public). The 'Code' tab is selected. The commit history shows 14 commits from 'felipe-infnet' pushed 1 minute ago. The commits are:

File	Commit Message	Time Ago
files	Delete .gitkeep	20 hours ago
roles	primeiro commit Felipe	1 hour ago
vars	primeiro commit Felipe	1 hour ago
LICENSE	primeiro commit Felipe	1 hour ago
README.md	validando tudo push	1 minute ago
felipe_ssh_keys.pub.pub	primeiro commit Felipe	1 hour ago
inventory.txt	primeiro commit Felipe	1 hour ago
playbook.yml	primeiro commit Felipe	1 hour ago

The README.md file content is:

```
projetoAIA2023

Repositório criado com o objetivo de controlar o ambiente do projeto de bloco. O projeto é público e será usado como exemplo para a graduação de Gestão em TI da Fundação Infnet.

Instalação automatizada do Wordpress via ansible Primeiro pull! Agora Online! Validação de push online.
```

## Pull:

No sentido inverso o arquivo README.md foi alterado online:

E executada a validação (CAT) do arquivo previa a um Pull da plataforma, validando novamente as alterações:

## TP4

### Desenvolvimento e criação das VMs;

É utilizada na criação das VMs todas as configurações de discos e rede disponibilizadas no ESXi.

*\*todo o processo de criação do ambiente será documentado no TP5 para não misturar rubricas*

*\*existem telas compartilhadas com o AT de nuvem privada*

### Primeiro boot e habilitação do ambiente Docker;

*\*foi notado que o Ubuntu 16LTS se comporta mal as vezes para instalar alguns componentes, decidi atualizar a VM para a versão 22.*

Por necessidade da empresa os containers de Banco de dados e da aplicação serão utilizados já que o webservice já se encontra instalado.

### Instalação Docker.IO via apt-get:

```
Processing triggers for man-db (2.10.2-1) ...
fflogMFluradainvest: $ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom libftd1-2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils glib-daemon-run | glib-daemon-sysvinit glib-doc
  git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 70,6 MB of archives.
After this operation, 307 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://br.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63,6 kB]
Get:2 http://br.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34,4 kB]
Get:3 http://br.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.0-0ubuntu1.1 [4,242 kB]
Get:4 http://br.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.5.9-0ubuntu3.1 [28,1 MB]
Get:5 http://br.archive.ubuntu.com/ubuntu jammy/universe amd64 docker.io amd64 20.10.12-0ubuntu4 [34,0 kB]
Get:6 http://br.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26,5 kB]
Get:7 http://br.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.5 [953 kB]
Get:8 http://br.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.5 [3.132 kB]
Get:9 http://br.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35,2 kB]
Fetched 79,6 MB in 3s (22,5 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 174251 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7-1ubuntu3_amd64.deb ...
Unpacking bridge-utils (1.7-1ubuntu3) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.0-0ubuntu1.1_amd64.deb ...
Unpacking runc (1.1.0-0ubuntu1.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.5.9-0ubuntu3.1_amd64.deb ...
Unpacking containerd (1.5.9-0ubuntu3.1) ...
Progress: [ 19%] [#########################################.....]
```

...corte de tela pós progresso de instalação:

```
Setting up docker.io (20.10.12-0ubuntu4) ...
Adding group `docker` (GID 137) ...
Done.
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.
Setting up git (1:2.34.1-1ubuntu1.5) ...
Processing triggers for man-db (2.10.2-1) ...
frio@VM1furadainvest:~$
```

### Testando o Docker;

Após o término da instalação foi testado o pull mais simples denominado “Hello World!” para garantir a funcionalidade da ferramenta e sua configuração:

```
frio@VM1furadainvest:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:c77be1d3a47d0caf71a82dd893ee61ce01f32fc758031a6ec4cf1389248bb833
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
frio@VM1furadainvest:~$
```

### Iniciado o pull e configuração de um banco de dados (MariaDB):

```
frio@VM1furadaInvest:/opt/mariadb$ sudo docker run -e MYSQL_ROOT_PASSWORD=infnet -e MYSQL_DATABASE=WordPress --name wordpressdb -v "$PWD/database":/var/lib/mysql -d mariadb:latest
Unable to find image 'mariadb:latest' locally
latest: Pulling from library/mariadb
0e3729cf69e0: Pull complete
0b2128efbd85: Pull complete
94c8ea0958ce: Pull complete
73cc9c81aeff: Pull complete
20bcac05cb43: Pull complete
c4a9b64b12f6: Pull complete
890f8c459a20: Pull complete
97535a02152a: Pull complete
Digest: sha256:8c15c3def7ae1b408c96d322a3cc0346dba9921964d8f9897312fe17e127b90
Status: Downloaded newer image for mariadb:latest
735ddbcb2cbb82dab2ff0d95155ac199171e5c2eade9fb38ff46cf583f049a
frio@VM1furadaInvest:/opt/mariadb$
```

### Validando o status de running do MariaDB:

```
frio@VM1furadaInvest:/opt/mariadb$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND       CREATED          STATUS          PORTS     NAMES
735ddbcb2cbb  mariadb:latest "docker-entrypoint.s..."  26 seconds ago   Up 25 seconds   3306/tcp   wordpressdb
frio@VM1furadaInvest:/opt/mariadb$
```

### Instalação do Portainer.IO:

Como demanda velocidade, foi instalada para administrar o Docker a tool chamada Portainer que viabiliza administração gráfica e administração remota, que é o caso da VM administrada no momento.

Mas, de qualquer forma seguindo o objetivo da avaliação, a aplicação também irá rodar em container, executando o pull:

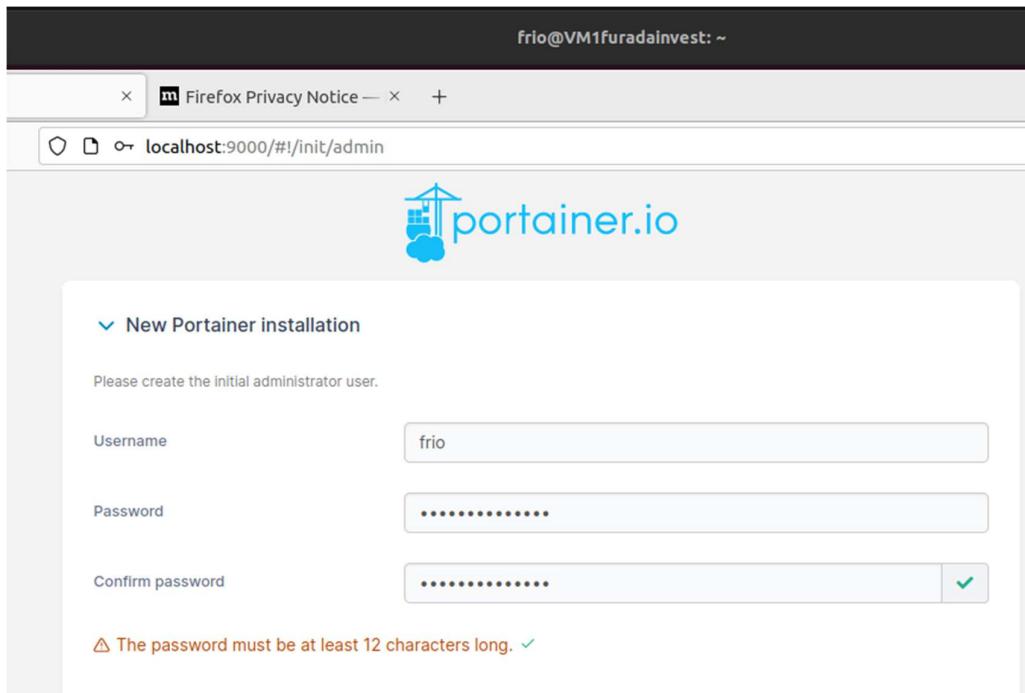
```
frio@VM1furadaInvest: $ sudo docker run -d -p 9000:9000 --name portainer --restart always -v /var/run/docker.sock:/var/run/docker.sock -v /home/frio/Portainer/data:/data portainer/portainer
Unable to find image 'portainer/portainer:latest' locally
latest: Pulling from portainer/portainer
772227786281: Pull complete
96fd13berfc87: Pull complete
6bad1d247b5b: Pull complete
b5d1b01b1d39: Pull complete
Digest: sha256:47b664434edf437badf7337e516e07f64477485c8ecc663ddabbe824b20c672d
Status: Downloaded newer image for portainer/portainer:latest
f845401dadde6af590432a0b6bc60cdf21d74d6927f3f65bb1357095f92f
frio@VM1furadaInvest: $
```

Validando a instalação e execução do que está em atividade até o momento:

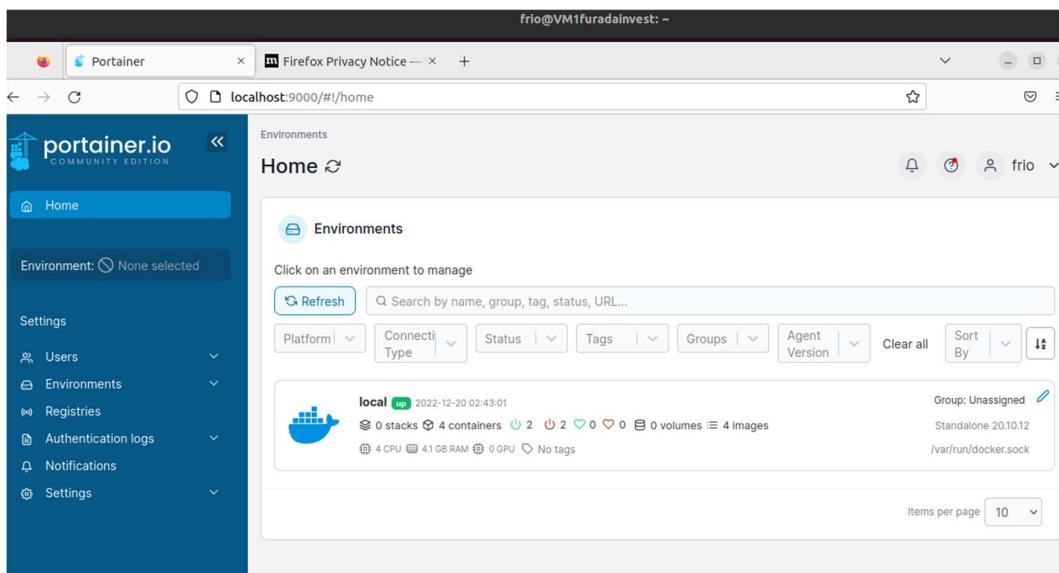
```
frio@VM1furadaInvest: $ sudo docker ps
CONTAINER ID   IMAGE      COMMAND       CREATED          STATUS          PORTS     NAMES
f845401dadde  portainer/portainer "/portainer"   2 minutes ago   Up 2 minutes   8000/tcp, 9443/tcp, 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp   portainer
er_efase9d695bd  mariadb:latest    "docker-entrypoint.s..."  11 minutes ago  Up 11 minutes  3306/tcp   wordpress
frio@VM1furadaInvest: $
```

## Configurando a instalação do Portainer.IO:

Logrando web já no mesmo servidor, mas com liberação de portas para ser feito remotamente foi criado uma conta de administração:



Validando o login e administração local:



Validando a instalação remotamente na rede:

The screenshot shows the Portainer.io Home page. On the left, there is a sidebar with the following navigation items: Home, Environment (None selected), Settings (Users, Environments, Registries, Authentication logs, Notifications, Settings). The main content area features a "Latest News From Portainer" section with a message about the New Portainer CE 2.16 Release. Below this is the "Environments" section, which displays a single environment named "local". The "local" environment details are as follows: Group: Unassigned, Status: Standalone 20.10.12, IP: /var/run/docker.sock. The environment contains 1 stack, 4 containers, 3 healthy, 1 warning, 0 errors, 2 volumes, and 5 images. It has 4 CPU, 4.1 GB RAM, 0 GPU, and no tags. A search bar and filters for Platform, Connect Type, Status, Tags, Groups, Agent Version, and Sort By are also present.

Posterior aos testes remotos, escolhida a aplicação, iniciando a instalação (pull) do wordpress via template:

The screenshot shows the Portainer.io Application templates list page. The sidebar on the left is identical to the previous screenshot, showing Home, Environment (local), Dashboard, App Templates (selected), Custom Templates, Stacks, Containers, Images, Networks, Volumes, Events, Host, Settings, and Users. The main content area is titled "Application templates list" and shows a search bar with the query "wordpd". A single template is listed: "WordPress" (stack). The template details are: WordPress setup with a MySQL database. There is a "Copy as Custom" button and a CMS label.

Parâmetros de configuração do DB MySQL:

The screenshot shows the Portainer.io interface. On the left, the sidebar menu is open, showing options like Home, local, Dashboard, App Templates (which is selected), Custom Templates, Stacks, Containers, Images, Networks, Volumes, Events, Host, Settings, and Users. The main panel is titled "Application templates list" and shows a "WordPress" template. It includes sections for "Information" (describing it as a WordPress instance connected to a MySQL database), "Configuration" (with fields for Name set to "infnetATfrio" and Database root password set to "Admin01!"), and "Access control" (with a toggle switch turned on, indicating access control is enabled). Below this, there are two buttons: "Administrators" (selected) and "Restricted". The "Administrators" button has a note: "I want to restrict the management of this resource to administrators only". The "Restricted" button has a note: "I want to restrict the management of this resource to users and/or teams".

Deploy em progresso:

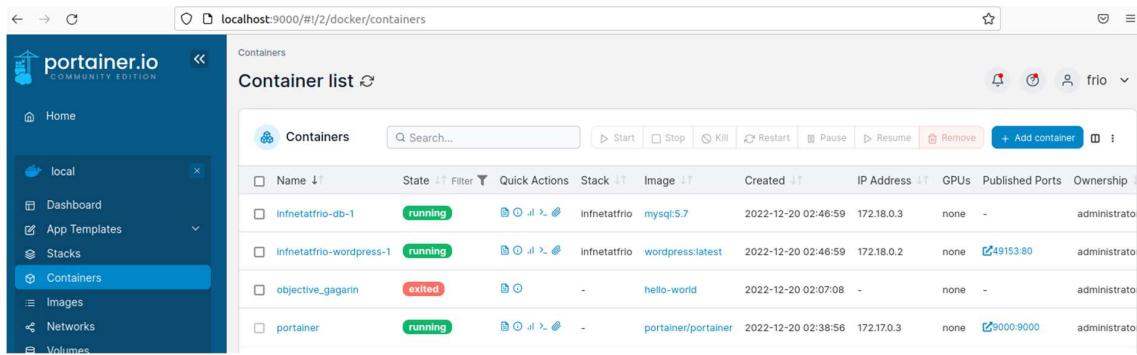
### Actions

Deployment in progress...

Verificado na notificação o sucesso:

The screenshot shows the Portainer.io notifications page. At the top, there's a header with icons for notifications, user profile, and dropdown. Below that is a "Notifications" section with a "Success" message: "Stack successfully deployed" with a green checkmark icon, timestamped "Just now". There's also a "Clear all" button. To the right of the notifications is a sidebar with "Add stack", "Ownership", and "administrators". At the bottom, there's a link "View all notifications".

Validação dos novos containers via GUI:



The screenshot shows the Portainer.io interface with the URL `localhost:9000#!/2/docker/containers`. The sidebar on the left shows the navigation menu with 'Containers' selected. The main area is titled 'Container list' and lists four containers:

Name	State	Image	Created	IP Address	Gpus	Published Ports	Ownership
infnetatfrio-db-1	running	infnetatfrio mysql:5.7	2022-12-20 02:46:59	172.18.0.3	none	-	administrador
infnetatfrio-wordpress-1	running	infnetatfrio wordpress:latest	2022-12-20 02:46:59	172.18.0.2	none	49153:80	administrador
objective_gagarin	exited	hello-world	2022-12-20 02:07:08	-	none	-	administrador
portainer	running	portainer/portainer	2022-12-20 02:38:56	172.17.0.3	none	9000:9000	administrador

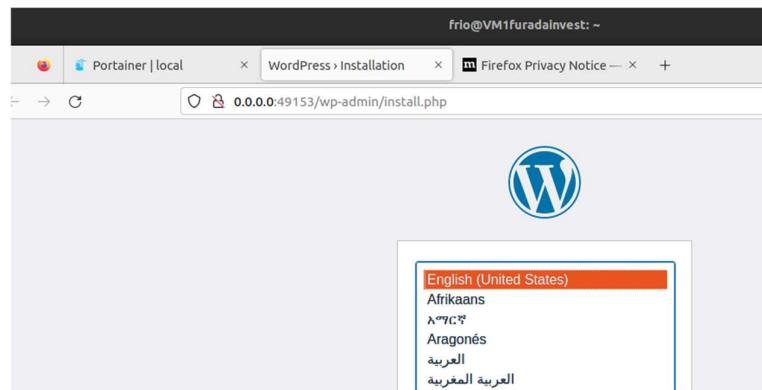
\*neste momento o MariaDB foi removido para testar a capacidade efetividade de administração do ambiente.

Validação dos novos containers via CLI:

```
frio@VM1furadalnvest: $ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
 NAMES
4d86f0a9ddb8      wordpress:latest   "docker-entrypoint.s..."   About a minute ago   Up About a minute   0.0.0.0:49153->80/tcp, :::49153->80/tcp
infnetatfrio-wordpress-1
76a57feb7a7d      mysql:5.7          "docker-entrypoint.s..."   About a minute ago   Up About a minute   3306/tcp, 33060/tcp
infnetatfrio-db-1
f845401dadde     portainer/portainer  "/portainer"           10 minutes ago     Up 9 minutes       8000/tcp, 9443/tcp, 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp
portainer
efase9d695bd      mariadb:latest      "docker-entrypoint.s..."   19 minutes ago     Up 19 minutes      3306/tcp
wordpressmari...
```

Validação de serviço WordPress:

Logando na página hospedada pela porta publicada escolhida diferente de 80 por segurança, configurado linguagem, título e senha administrativa :



The screenshot shows a Firefox browser window with the URL `0.0.0.0:49153/wp-admin/install.php`. The browser tabs include 'Portainer | local', 'WordPress > Installation', and 'Firefox Privacy Notice'. The main content of the page shows the WordPress logo and a language selection dropdown menu. The menu is currently set to 'English (United States)' and also lists other languages: Afrikaans, മലയാളം, Aragonés, العربية, and العربية المغربية.

Portainer | local    WordPress > Installation    Firefox Privacy Notice

0.0.0.0:49153/wp-admin/install.php?step=1

Welcome to the famous five-minute WordPress installation process! Just fill in the information needed and you'll be on your way to using the most extendable and powerful personal publishing platform.

### Information needed

Please provide the following information. Do not worry, you can always change these settings later.

**Site Title** Docker Rox InfnetAT\_FRIO

**Username** frio  
Usernames can have only alphanumeric characters, spaces, periods, and the @ symbol.

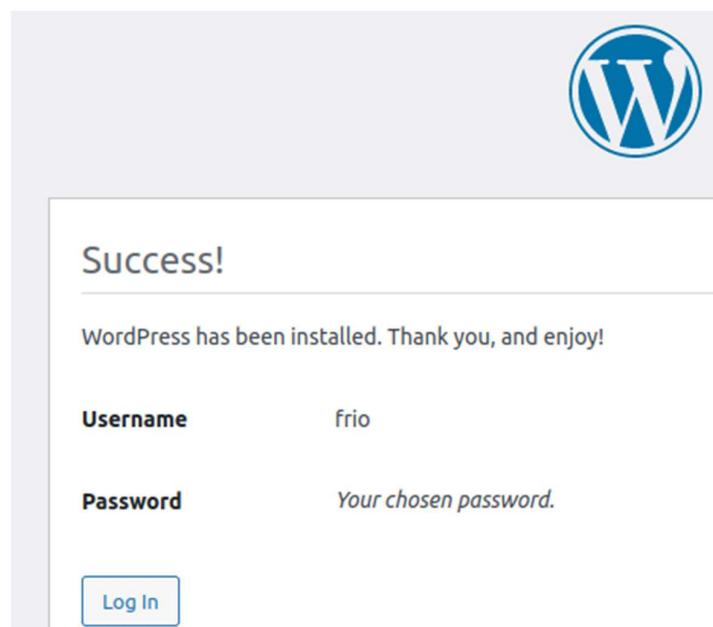
**Password** Admin01!  
Very weak  
 Hide

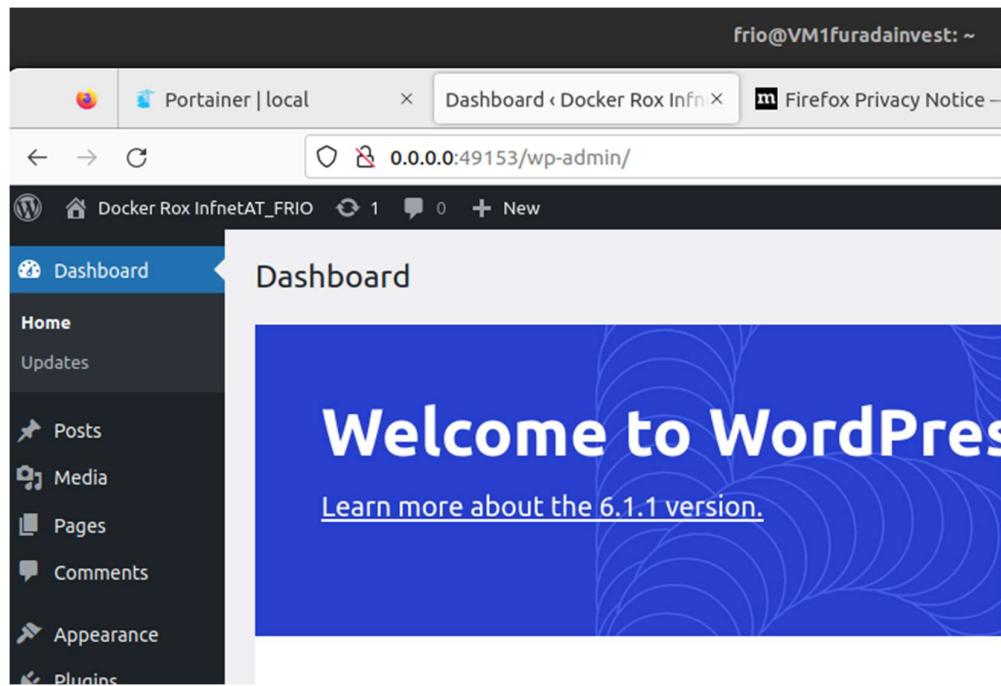
**Confirm Password**  Confirm use of weak password

**Your Email** felipe.rio@al.infnet.edu.br  
Double-check your email address before continuing.

**Search engine visibility**  Discourage search engines from indexing this site  
It is up to search engines to honor this request.

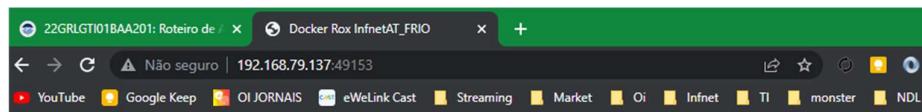
Posterior ao sucesso da configuração o processo de login na parte administrativa:





### Validando a parte administrativa localmente:

Os testes externos em uma máquina fora do ambiente virtualizado:



Docker Rox InfnetAT\_FRIO

Mindblown: a blog about philosophy.

Hello world!

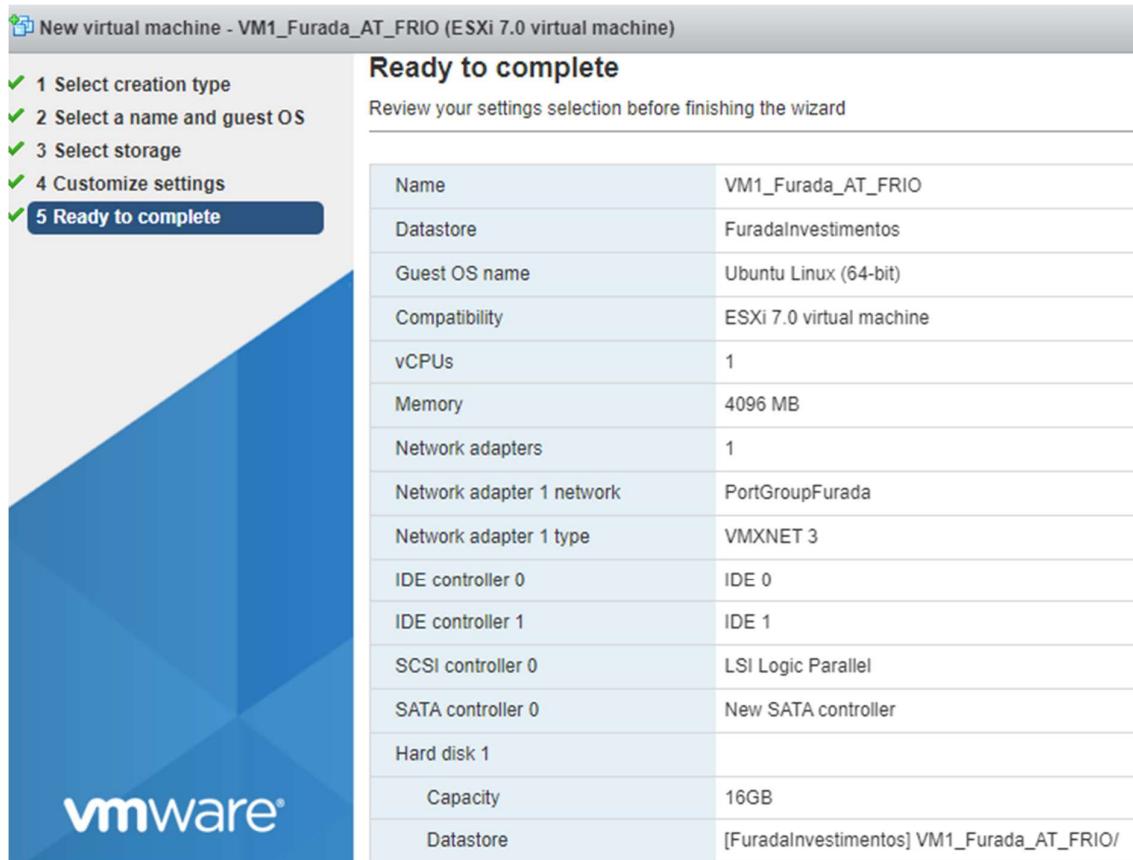
Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

December 20, 2022

## TP5

### Criação das VMs para instalação e configuração do Ansible;

Será utilizada na criação das VMs todas as configurações de discos e rede disponibilizadas no ESXi.



## Repassando os playbooks para Docker, LAMP e Wordpress;

\*foi notado que o Ubuntu 16LTS se comporta mal as vezes para instalar alguns componentes, então a imagem dos alvos foi trocada para o CentOS9 Stream.

### Execução do Playbook Docker:

Para instalar o Docker no host alvo, foi criado um inventário, e agrupado como [Docker] e solicitado pela permissão de root para execução:

```
[root@ansihost projetoAIAdocker]# ansible-playbook dockerinstall.yml -i inventory.txt --ask-become-pass
BECOME password:

PLAY [Instalando Docker] ****
TASK [Gathering Facts] ****
ok: [target2]
TASK [Adicionando repositorio Docker] ****
changed: [target2]
TASK [Instalando Docker package] ****
changed: [target2]
TASK [Starting docker service] ****
changed: [target2]
TASK [Docker SDK para Python3] ****
changed: [target2]
PLAY RECAP ****
target2 : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
[root@ansihost projetoAIAdocker]#
```

Com o final da execução do playbook de instalação do Docker, foi validado e \*testada a instalação do Docker no host alvo (target2):

```
[root@ansitarget2 ~]# docker ps -a
CONTAINER ID   IMAGE      COMMAND   CREATED     STATUS      PORTS      NAMES
[root@ansitarget2 ~]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:ffb13da98453e0f04d33a6eee5bb8e46ee50d08ebe17735fc0779d0349e889e9
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

\*A validação foi a verificação do Docker vazio e execução do pull do container hello-world

## Instalação LAMP/wordpress:

Posterior a validação do Docker foi executado o playbook para instalar os componentes necessários para o wordpress.

```
[root@ansihost projetoAIAdocker]# ansible-playbook dockerwordpress.yml -i inventory.txt --ask-become-pass
BECOME password:
PLAY [Deploy WordPress usando container] ****
TASK [Gathering Facts] ****
ok: [target2]
TASK [Start MySQL container] ****
changed: [target2]
TASK [Start WordPress container] ****
changed: [target2]
PLAY RECAP ****
target2 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
[root@ansihost projetoAIAdocker]#
```

\*disponível o código no repositório

Verificado no servidor o deploy e start com sucesso.

```
[root@ansitarget2 ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8e6d41c61aa7 wordpress:latest "docker-entrypoint.s..." 12 seconds ago Up 10 seconds 0.0.0.0:80->80/tcp wordpress
18ab752cc233 mysql:5.7 "docker-entrypoint.s..." 39 seconds ago Up 37 seconds 3306/tcp, 33060/tcp mysql
fd408fd16719 hello-world "/hello" About a minute ago Exited (0) About a minute ago
[root@ansitarget2 ~]# docker stats
CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS
8e6d41c61aa7 wordpress 0.0% 39.98MB / 1.712GB 2.28% 2.19KB / 0B 61.9MB / 0B 6
18ab752cc233 mysql 0.01% 202MB / 1.712GB 11.52% 2.45KB / 0B 152MB / 730MB 27
CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS
```

## Github push:

Como demanda solicitada um novo remote do git foi criado para executar o push:

```
[root@ansihost projetoAIAdocker]# ^C
[root@ansihost projetoAIAdocker]# git remote set-url origin https://github.com/felipe-infnet/projetoAIAdocker.git
[root@ansihost projetoAIAdocker]# ^C
```

## Ajuste de chave SSH:

Processo de ajuste da chave SSH do servidor e clone do repositório.

```
[root@ansihost projetoAIAdocker]# ssh-keygen -t ed25519 -C "felipe.rio@al.infnet.edu.br"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
/root/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519.
Your public key has been saved in /root/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:IVqe+tpJzT6+5ZzKhZBotds8Ei4571EZh19nmNp/iPc felipe.rio@al.infnet.edu.br
The key's randomart image is:
+--[ED25519 256]--+
|          o
| + + . + o
| * = * + o
| + B S o .
| . + X . o .
| = * B o . +
| B *.* . o
| .o*.*++ E
+----[SHA256]----+
[root@ansihost projetoAIAdocker]# eval "$(ssh-agent -s)"
Agent pid 4539
[root@ansihost projetoAIAdocker]# ssh-add ~/.ssh/id_ed25519
Identity added: /root/.ssh/id_ed25519 (felipe.rio@al.infnet.edu.br)
[root@ansihost projetoAIAdocker]# cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTES5AAAIK0Vi0bYTx2fP06besViGTF6k6r2Elk3j7sz7ny/9vH felipe.rio@al.infnet.edu.br
[root@ansihost projetoAIAdocker]# git clone git@github.com:repositorio-0:felipe-infnet/projetoAIAdocker.git
Cloning into 'projetoAIAdocker'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

```

Validação dos arquivos que irão subir com o add e um novo commit em seguida do push:

```
[root@ansihost projetoAIAdocker]# cd projetoAIAdocker/
[root@ansihost projetoAIAdocker]# ls -ltr
total 16
-rwxr-xr-x. 1 root root 234 Mar 24 19:13 inventory.txt
-rwxr-xr-x. 1 root root 2316 Mar 24 19:13 dockerwordpress.yml
-rw-r--r--. 1 root root 5 Mar 24 19:28 test
-rw-r--r--. 1 root root 84 Mar 24 19:28 README.md
[root@ansihost projetoAIAdocker]# git add -A
[root@ansihost projetoAIAdocker]# git commit -m "commit docker Felipe"
[main 09927e4] commit docker Felipe
 2 files changed, 96 insertions(+)
   create mode 100755 dockerwordpress.yml
   create mode 100755 inventory.txt
[root@ansihost projetoAIAdocker]# git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.20 KiB | 1.20 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:repositorio-0:felipe-infnet/projetoAIAdocker.git
 448e9a7..09927e4 main -> main
branch 'main' set up to track 'origin/main'.
[root@ansihost projetoAIAdocker]#
```

Checagem com git stats e verificação de que os arquivos estão online.

```
[root@ansihost projetoAIAdocker]# git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   dockerwordpress.yml
    modified:   inventory.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    dockerinstall.yml

no changes added to commit (use "git add" and/or "git commit -a")
[root@ansihost projetoAIAdocker]# git add -A
[root@ansihost projetoAIAdocker]# git commit -m "Updates docker install e wordpress container"
[main 97d1ecd] Updates docker install e wordpress container
 3 files changed, 43 insertions(+), 88 deletions(-)
 create mode 100644 dockerinstall.yml
[root@ansihost projetoAIAdocker]# git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1013 bytes | 1013.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com-repository-0:felipe-infnet/projetoAIAdocker.git
 09927e4..97d1ecd main -> main
branch 'main' set up to track 'origin/main'.
[root@ansihost projetoAIAdocker]#
```

## Validação GitHub:

Validação online do último commit:

The screenshot shows the GitHub repository page for 'projetoAIAdocker'. The repository has 1 branch and 0 tags. The most recent commit was made by 'felipe-infnet' on '2 days ago' with the message 'commit docker Felipe'. The commit hash is '09927e4'. The commit details show four files: README.md, dockerinstall.yml, dockerwordpress.yml, and inventory.txt, all updated '1 minute ago'. The README.md file content is visible below, describing it as a 'playbook simples sem roles para instalacao de docker e wordpress'.

File	Description	Updated
README.md	Initial commit	2 days ago
dockerinstall.yml	Updates docker install e wordpress container	1 minute ago
dockerwordpress.yml	Updates docker install e wordpress container	1 minute ago
inventory.txt	Updates docker install e wordpress container	1 minute ago
test	teste	2 days ago

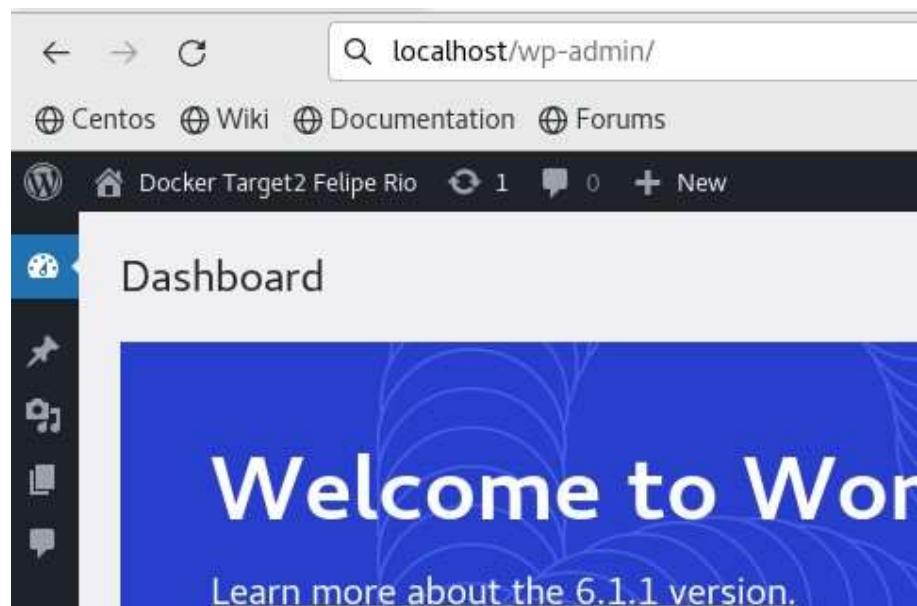
README.md

**projetoAIAdocker**

playbook simples sem roles para instalacao de docker e wordpress

### **Validação Wordpress:**

Posterior a publicação a validação no servidor.



## TP6

### Conteúdo:

A empresa precisa criar um ambiente escalável virtualizado com instâncias Linux preparadas para prover entrega de um framework Wordpress geograficamente distribuído pelo país.

O principal objetivo é prover infraestrutura virtualizada para permitir o desenvolvimento de aplicativos e *web hosting* no futuro, melhorando a disponibilidade de recursos e alavancando o desenvolvimento de projetos que serão controlados via GitHUB.

Essa empresa terá as seguintes atribuições para sua equipe de DevOPs.

- Ambiente com desktops virtuais via acesso remoto para criação dos layouts;
- Ambiente de staging separado de produção;
- Armazenamento de seu banco de dados centralizado;
- Backup da infra e snapshots dos servidores;
- Controle de versionamento do ambiente centralizado no GitHUB;
- Possibilidade de instâncias extras para dias de forte demanda.
- Firewall logico e roteamento segregado em cada uma das camadas.

### **Relevância:**

Uma solução de nuvem pública não se justifica, pois, por conta do ambiente de staging e um dos fatores determinantes é que a empresa já conta com um parque de máquinas compatível e ainda dentro de garantia.

Junto dos servidores os stakeholders querem total controle dos dados criados dentro da plataforma assim como o controle de custo do consumo indefinido.

### **Vantagens e benefícios:**

A vantagem da centralização de virtualização que foi salientada é em cima de um dos maiores problemas na instituição, onde todos os equipamentos são descentralizados, vinculados a políticas de domínio restritivas de utilização e acesso que impedem os alunos desenvolverem e instalarem aplicativos.

Em um ambiente de nuvem privada, a alocação de equipamentos perde sentido, o gerenciamento é centralizado e organizado. A instituição possui hardware de grande capacidade subaproveitado e o custo para o desenvolvimento e aplicação de um IaaS é praticamente nulo.

### **Gerenciamento e versionamento:**

A forma colaborativa de gerenciamento da versão de tudo que precisa ser lançado que será utilizada é o GITHUB, por ser fundamental para compartilhar de forma eficiente seu código. A ferramenta conta com controle de versão com histórico, facilitando a identificação e correção de erros, colaborativa, feita e gerenciada por uma grande comunidade e de alta integração com os maiores IDEs servindo também como portfólio do trabalho feito, com acesso global.

### **Ambiente para web hosting;**

O processo de execução do Wordpress básico será entre hardware e software;

- Servidor baseado OS Linux Ubuntu/CentOS com LAMP nas versões;
- Apache 2.4
- MySQL versão 8.0 (LTS 2026) ou MariaDB versão 10.11 (LTS até 2028);
- PHP versão 8;
- 64GB de RAM
- 2 cores com 6 HTs
- NAS/SAN com 500GBs livre

### **Descrição da aplicação Docker e Ansible;**

O Docker é uma plataforma baseada em containers que permite criar, implementar (deploy) e executar aplicativos em ambientes isolados. O serviço fornece uma maneira de encapsular um aplicativo e suas dependências dentro de um container, podendo ser implantado em qualquer lugar, independente do sistema subjacente. Com isso em vista do projeto, se torna mais fácil para os desenvolvedores efetuarem o deploy das aplicações e gerenciar os aplicativos em diferentes ambientes com os benefícios a seguir:

- Portabilidade: os contêineres Docker podem ser executados em qualquer sistema operacional, desde que o Docker esteja instalado, sem se preocupar com as diferenças de ambiente entre os sistemas;
- Isolamento: cada contêiner Docker é isolado do ambiente em que é executado, garantindo que não haja conflitos de dependência entre aplicativos e evitando que um aplicativo afete outro;
- Escalabilidade: os contêineres do Docker são leves e podem ser facilmente escalados horizontalmente para lidar com o aumento da carga de trabalho.

Já o Ansible é uma ferramenta de automação que permite configurar e gerenciar sistemas e aplicativos de forma programática. Iremos no projeto usar uma linguagem declarativa YAML para definir o estado desejado do sistema e, em seguida, aplicar em lote as alterações necessárias para alcançar esse estado.

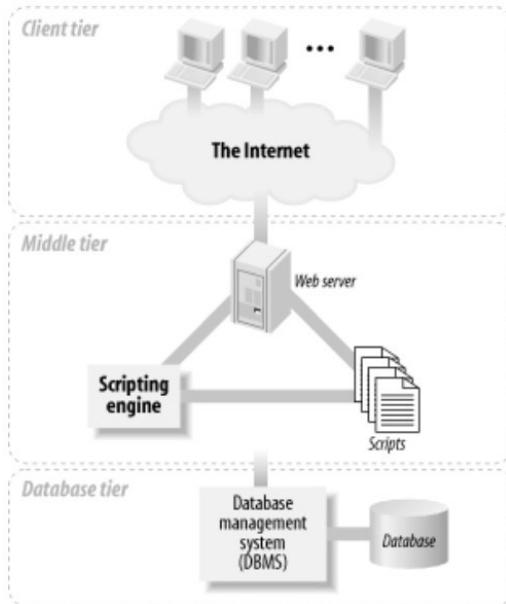
Alguns dos benefícios do uso do Ansible no projeto incluem:

- Automatização: o Ansible permite automatizar tarefas repetitivas de configuração e gerenciamento de sistemas e aplicativos, economizando tempo e reduzindo erros humanos.
- Flexibilidade: o Ansible pode ser usado para gerenciar sistemas em uma ampla variedade de plataformas, incluindo Linux/Unix, macOS e Windows.
- Escalabilidade: o Ansible pode ser usado para gerenciar muitos sistemas de forma centralizada e escalável.

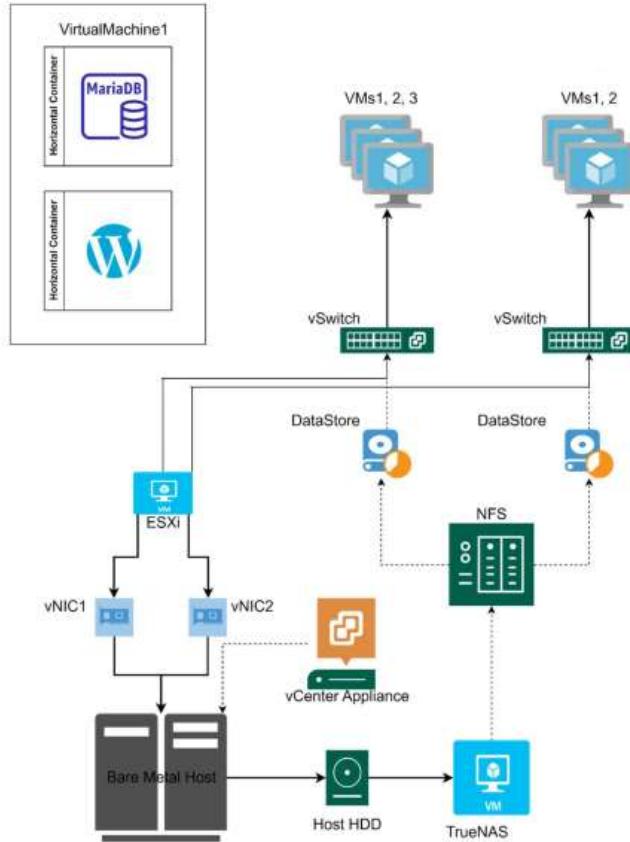
Em resumo, o uso do Docker e do Ansible pode ajudar a simplificar e automatizar a implantação e gerenciamento de aplicativos em ambientes Linux, tornando-os mais portáteis, escaláveis e fáceis de gerenciar.

### **Framework Wordpress;**

A estrutura de camadas se apresenta separada em tiers;



O cenário da organização da infra da aplicação terá um investimento de compra de licenças dos softwares da operação, custos de manutenção, resiliência térmica e elétrica do parque, assim como o treinamento do time para dar manutenção preventiva e corretiva no ambiente.



A aplicação Wordpress tem requisitos específicos, mas que serão bem atendidos com uma solução de virtualização ESXi com uma monitoração vCSA. A plataforma VMware, dentro do modelo separado por camadas, pode atender uma fácil escalabilidade dos recursos, podendo prover instalação ágil de novas instâncias, com o gerenciamento facilitado na camada servidores virtuais, isolamento da rede e ambiente de testes/staging. O balanceamento de carga pode ser configurado de maneira muito rápida, assim como a monitoração do consumo de processamento dos servidores e instâncias, podendo ainda utilizar a replicação e automação de backups/snapshots, de maneira que a empresa possa usufruir dos recursos disponíveis de forma rápida, descomplicada e transparente.

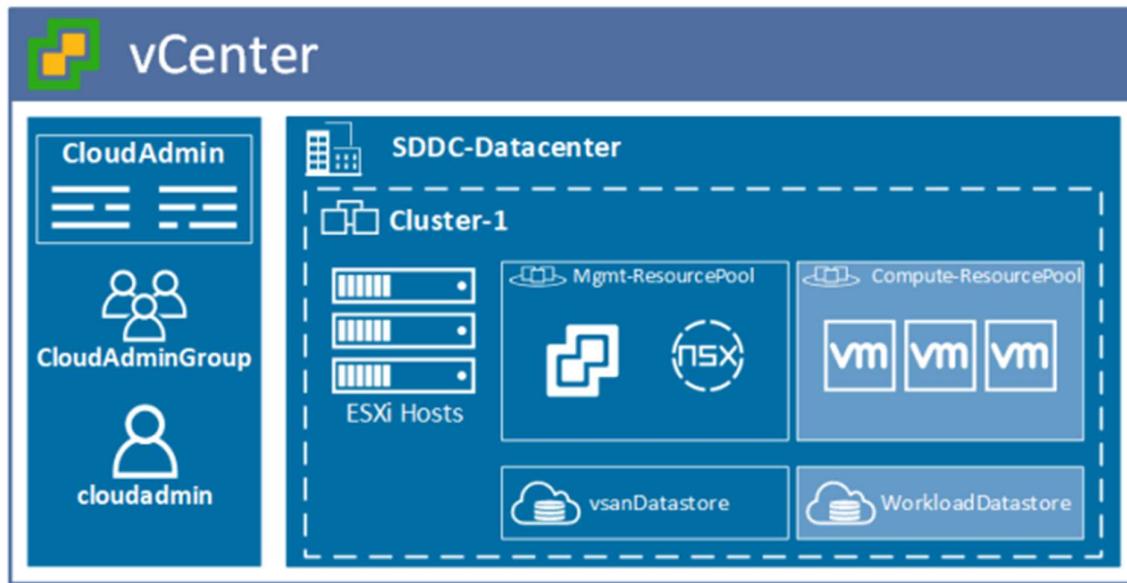
## TP7

### Ambiente ESXi;

Com o processo de infra escolhido, o VMware ESXi 7.0 tem diversos benefícios salientados para os stakeholders, alguns dos principais são:

1. Virtualização de Servidores: o VMware ESXi é uma solução de virtualização de servidores que permite executar várias máquinas virtuais em um único servidor físico. Isso reduz a necessidade de comprar e manter vários servidores físicos separados, o que pode reduzir significativamente os custos.
2. Consolidar Recursos: a virtualização também permite consolidar recursos de hardware, como CPU, RAM e armazenamento, em um único pool, que pode ser compartilhado por várias máquinas virtuais. Isso pode aumentar a eficiência e a utilização dos recursos disponíveis.
3. Gerenciamento Centralizado: o VMware ESXi permite o gerenciamento centralizado de todas as máquinas virtuais em um ambiente. Isso pode facilitar o gerenciamento, a implementação de atualizações e a solução de problemas, pois tudo pode ser feito a partir de um único local.
4. Alta Disponibilidade: o VMware ESXi possui recursos de alta disponibilidade que garantem que as máquinas virtuais continuem funcionando, mesmo em caso de falha de hardware. Isso é especialmente importante para aplicativos críticos, que precisam estar sempre disponíveis.
5. Migração de Máquinas Virtuais: o VMware ESXi permite migrar máquinas virtuais em tempo real de um servidor físico para outro, sem interrupção dos serviços. Isso pode ser útil para balancear a carga de trabalho ou para manutenção preventiva.
6. Segurança: o VMware ESXi possui recursos de segurança avançados, como criptografia, isolamento e segregação de rede, que podem ajudar a proteger as máquinas virtuais e os dados nelas armazenados.
7. Escalabilidade: o VMware ESXi é altamente escalável e pode suportar muitas máquinas virtuais em um único ambiente, o que permite que as organizações cresçam sem precisar se preocupar com limitações de hardware.

Um exemplo lógico do ambiente configurado com controle de permissão de acessos ao SDDC e os serviços designados.



#### **Hardware para o ambiente;**

Dentro do planejamento já parcialmente mapeado o processo de execução do Wordpress básico não teve mudanças;

- 64GB de RAM
- 2 cores com 6 HTs
- NAS/SAN com iniciais 500GBs livre

#### **Plano de implantação e aprovisionamento;**

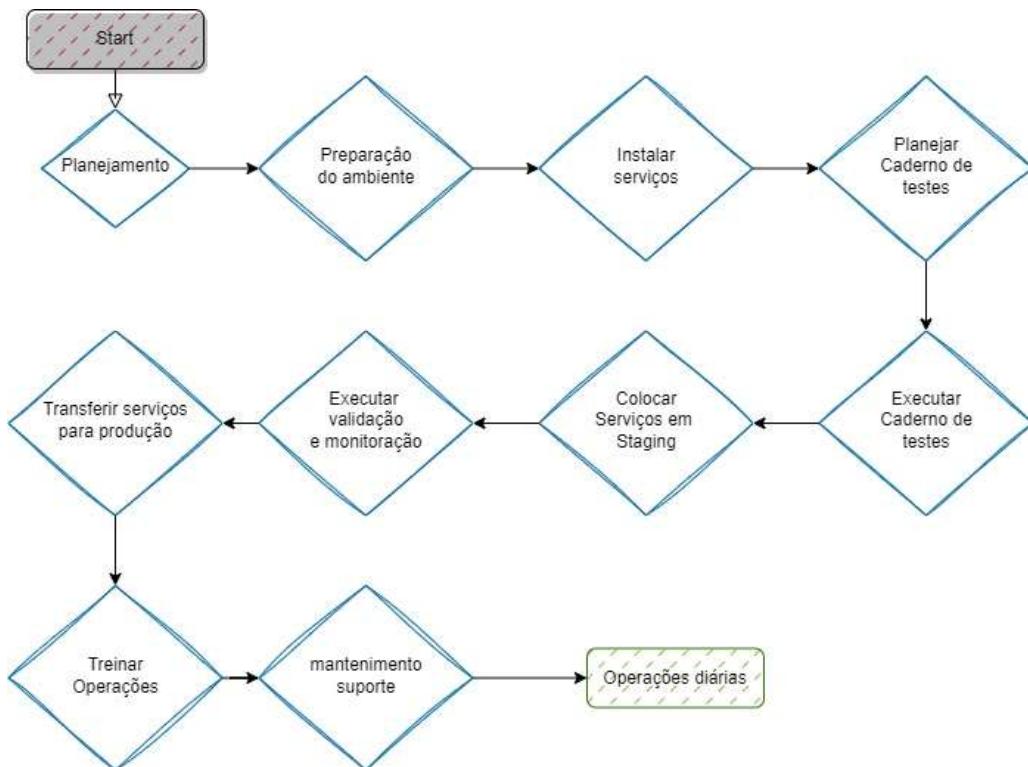
O plano de instalação e aprovisionamento irá seguir com o prazo inicial de 4 meses, onde o ambiente será avaliado e documentado com os padrões de hardware em quantitativo abaixo:

- Total de CPUs;
- Total de RAM alocada;
- Capacidade de IOPS e tecnologia dos discos disponíveis;
- Throughput da rede;
- Alocação do storage e rede FC iSCSI/SAN.

O diagrama de implantação vai seguir o cronograma da implantação, onde todas as premissas detectadas e acordadas no TAP serão estabelecidas. Após a documentação da infra disponível aquisição de novos componentes caso necessário o ambiente será preparado para receber as imagens. Em seguida os serviços do ESXi serão configurados.

Após a configuração, os testes para a funcionalidade serão elaborados, executados e colocados em staging para validação e monitoria, só será transferido para produção após as condições dos testes e monitoria estiverem a contento.

O treinamento para manutenimento e utilização será realizado e o hands-on dos serviços será realizada, fechando o ciclo do projeto.



## Reuniões e milestones;

O projeto vai ter reuniões de milestone a cada 15 dias e se espera que tenham os pontos de fases concluídos a cada 4 semanas de acordo com o cronograma registrado:

Cronograma do projeto					
Fases		Abril	Maio	Junho	Julho
<b>Planejamento</b>					
1	Planejamento				
1	Preparação do ambiente				
<b>Implantação/Deploy</b>					
2	Instalação dos serviços				
2	Planejamento caderno de testes				
2	Execução do caderno de testes				
2	Deploy staging				
2	Execução e monitoramento				
<b>Produção/Treinamento</b>					
3	Deploy para produção				
3	Treinar operação				
<b>Entrega</b>					
4	Hands-ON				
4	Mantenimento / Suporte				

Na proposta inicial até o ponto de pré-teste os servidores já vão rodar em cima do ESXi com os seus sistemas operativos no momento da implementação, que tem as etapas internas:

- Validação das últimas versões de Software LTS;
- Levantamento do custo de tempo para migrar os serviços;
- Documentação da instalação e do diagrama de utilização;
- Implementação ao GitHUB;
- Snapshot primário e posterior a cada etapa.
- Divulgação para funcionários da solução
- Correções e adequações

Com as fases internas realizadas a virtualização e provisionamento vão estar sob controle, atendendo as necessidades iniciais da empresa em um projeto futuro por crescimento se torna fácil escalar o serviço para a nuvem pública.

#### **Comparativo entre os serviços existentes:**

Por comparativo solicitado pelos stakeholders, o ambiente de nuvem publica Amazon (AWS) teve alguns fatos destacados:

Características	ESXi	AWS
SLA fornecedor	Suporte Vmware 24/7 com a licença, limitado a 60 tickets ao ano	VPC/EC2/S3/R53/RDS >95.0%
Controle e backup dos dados	Responsabilidade interna	Responsabilidade interna
Tecnologia Armazenamento	HDD SAS de disponibilidade imediata, liberado upgrade.	HDD ou SSD avaliado no custo
Segregação e escalabilidade	Depende de movimentação interna e controle de mudanças	De fácil acesso, porém o custo pode se tornar imprevisível pela escala de acessos.

A comparação do serviço oferecido pela Amazon não se encaixa do modelo de infraestrutura demandado no projeto. A Amazon apesar de ser pioneira na solução, ainda possui forte descrédito, por parte dos stakeholders. Na parte de S.I., ainda é apontada grande preocupação com relação à posse dos dados. Conforme estudo de premissas realizado, existe uma forte possibilidade de uso do virtualizado centralizado não para a produção direta, mas para a operação em paralelo para treinamento e testes das equipes, que lentamente vão incorporando as funcionalidades mais simples e fáceis de operar, por parte do suporte e por parte dos usuários.

No modelo de trabalho previsto em projeto, o ESXi inicialmente ainda é o mais adequado para a utilização on premises dos colaboradores que possuem a necessidade de rápida disponibilização de ambientes de testes. Nem todos os colaboradores ainda estão certificados para utilizar os serviços da AWS por terem dificuldade de obter conexão. É comum usuários perderem os dias trabalhados por terem problemas de segurança e com isso dias de trabalho perdidos dentro do cronograma de desenvolvimento.

## TP8

### Versões de componentes;

Dentro do projeto de implantação foram utilizados 2 tipos de sistemas operacionais, baseados em LAMP (Linux, Apache, MySQL/MariaDB e PHP) com um hypervisor/orquestrador e algumas ferramentas mencionadas abaixo;

- Sistema operacional Linux Ubuntu 16.04 (Xenial Xerus) e 22.04 (Jammy Jellyfish)
- Sistema operacional CentOS 7 e 9 Stream
- Hypervisor Orquestrador VMWare ESXi 7.0
- VMWare vSphere/vCSA 7
- Ansible 2.10.0
- YAML 1.2.2
- Docker 23.0.2
- Portainer CE 2.16.2
- Apache 2.4
- MySQL 8.0 (LTS 2026) e MariaDB 10.11 (LTS 2028)
- PHP 7.4 e 8.0 (dependendo do Linux usado)
- MobaXterm 22 (para acesso remoto)

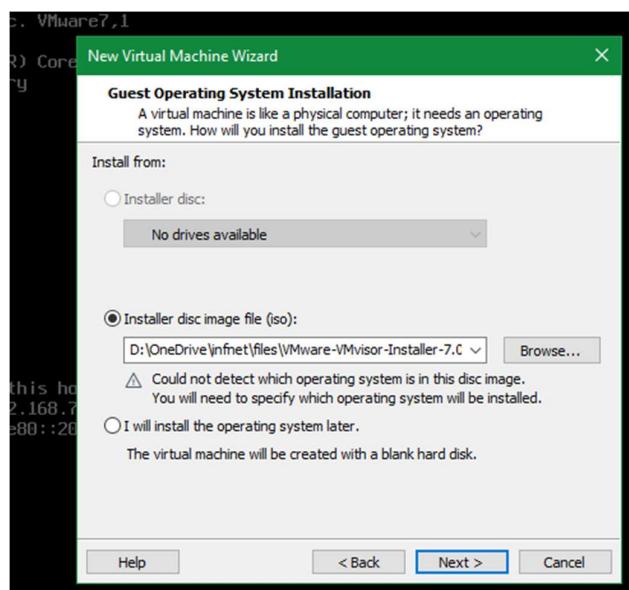
## **Implementação da infraestrutura de virtualização;**

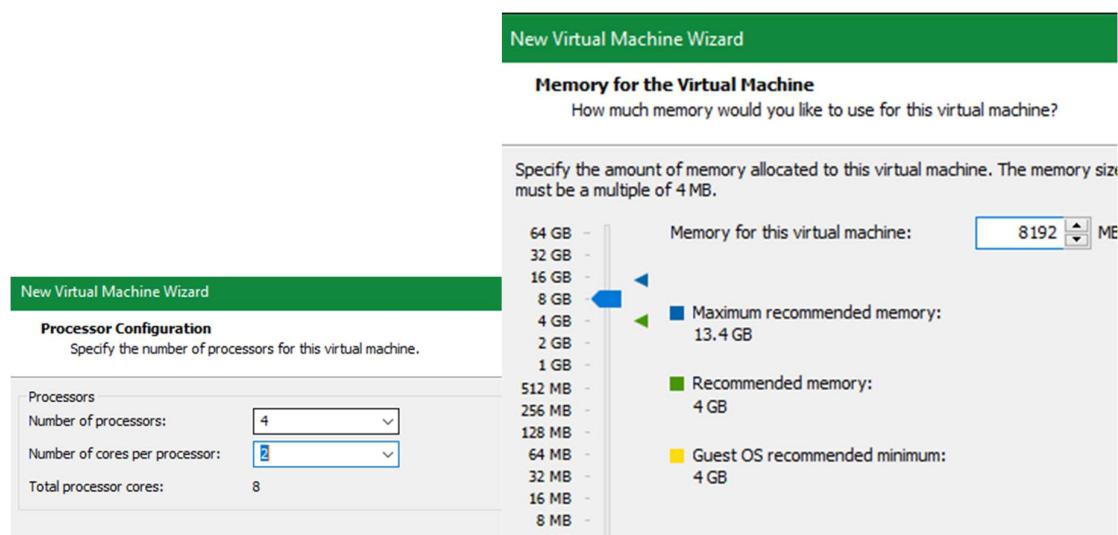
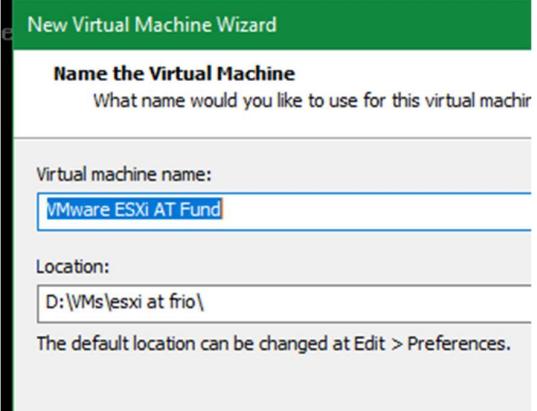
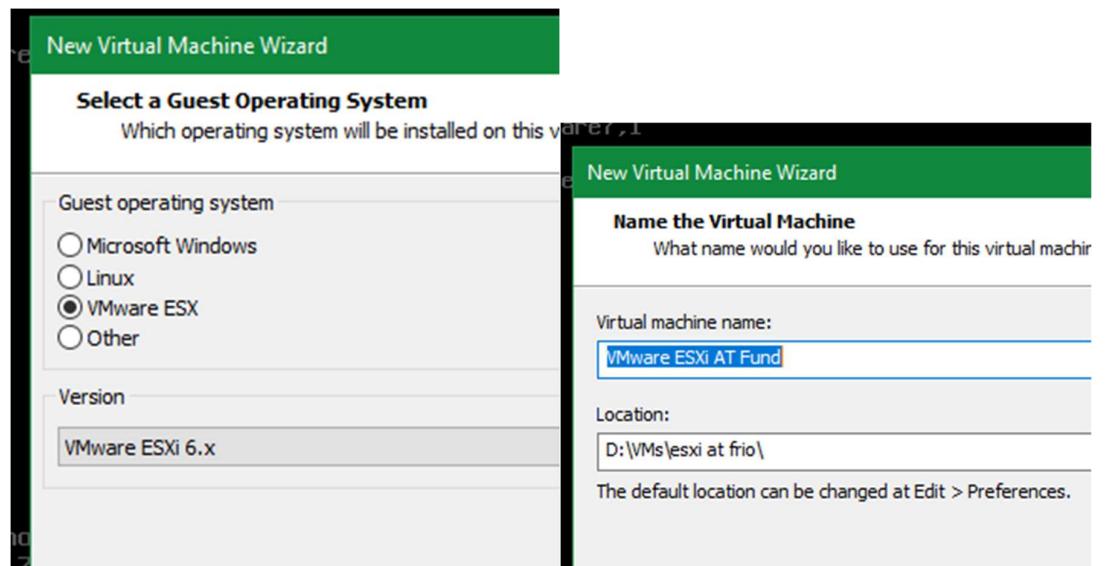
A escolha da feature VCSA foi baseada em Server Appliance, pois não requer licença adicional de Sistema Operacional pois a base é Linux e já configurada, com menos consumo de hardware para operar.

O esforço administrativo do banco de dados MariaDB/MySQL do ambiente de até 100 hosts é baixo, o método de deploy pode ser baseado em templates OVF ou OVA, nesta versão o Syslog collector e o Dump são serviços padrões, assim como o webclient facilitando a administração.

Os pontos negativos são não ter suporte para IPv6 e o serviço de Server heartbeat para segurança dos hosts e monitoria, o que não é o caso do projeto inicial.

O processo de instalação do ESXi segue com a inserção de uma ISO no ambiente, a seleção do tipo de sistema operacional, local de armazenamento, processadores e memoria disponibilizada, que não segue a documentação atual pois no host não tenho tanta memória.





Posteriormente a seleção da comunicação da controladora de rede e tipo de disco instalado.

The screenshot shows the 'Network Type' step of the 'New Virtual Machine Wizard'. On the left, under 'Network connection', there are four options:

- Use bridged networking
- Use network address translation (NAT)
- Use host-only networking
- Do not use a network connection

Below each option is a brief description. To the right, under 'Select I/O Controller Types', it says 'Which SCSI controller type would you like to use?'. It lists four options for 'SCSI Controller':

- BusLogic (Not available for 64-bit guests)
- LSI Logic
- LSI Logic SAS
- Paravirtualized SCSI (Recommended)

Finalizando o tamanho do disco a ser usado na instalação e o resumo antes do powerON

The screenshot shows the 'Ready to Create Virtual Machine' step of the 'New Virtual Machine Wizard'. On the left, under 'Specify Disk Capacity', it says 'How large do you want this disk to be?' with a input field containing '40'. Below it, 'Recommended size for VMware ESXi 6.x: 40 GB' is shown. There is a checkbox 'Allocate all disk space now.' followed by a note about performance. At the bottom, two radio button options are shown:

- Store virtual disk as a single file
- Split virtual disk into multiple files

A note below the second option states: 'Splitting the disk makes it easier to move the virtual machine, but may reduce performance with very large disk.'

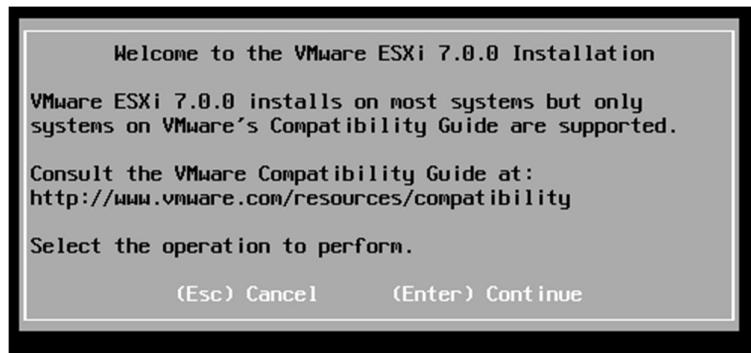
To the right, under 'Ready to Create Virtual Machine', it says 'Click Finish to create the virtual machine. Then you can install the operating system onto the disk.' A summary table shows the settings:

Name:	VMware ESXi AT Fund
Location:	D:\VMs\esxi at frio\
Version:	Workstation 15.x
Operating System:	VMware ESXi 6.x
Hard Disk:	40 GB, Split
Memory:	8192 MB
Network Adapter:	NAT
Other Devices:	8 CPU cores, CD/DVD, USB Controller

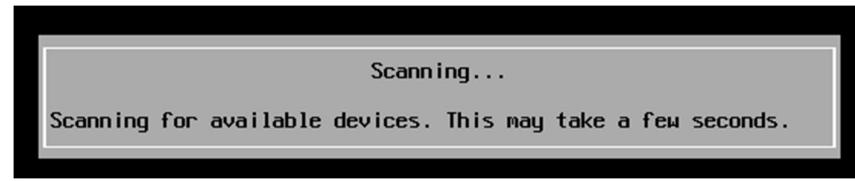
A 'Customize Hardware...' button is at the bottom.

## Efetuando o PowerON;

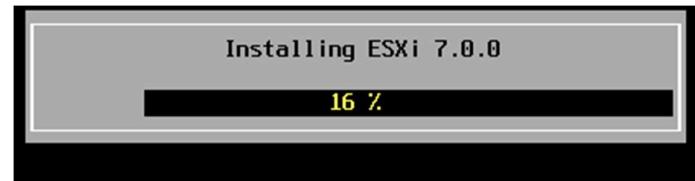
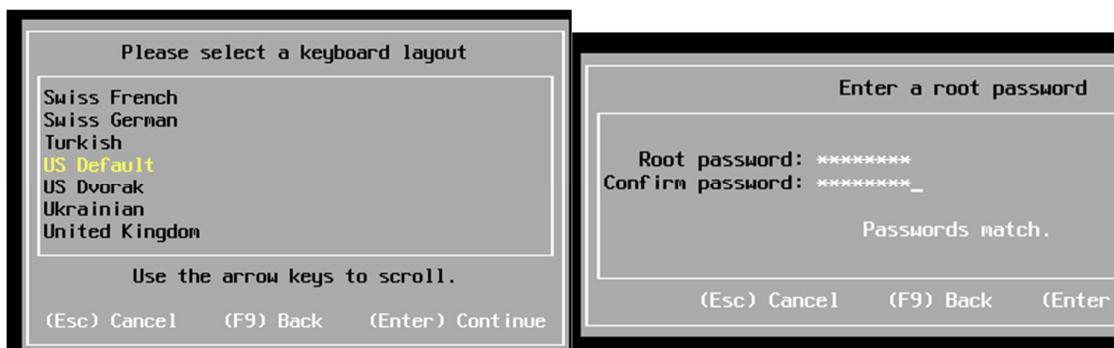
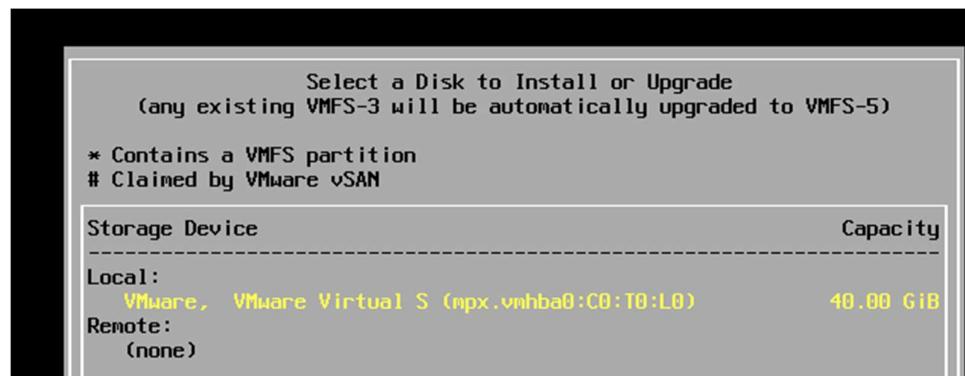
Com o servidor ligado, se inicia o processo de instalação seguindo as telas:



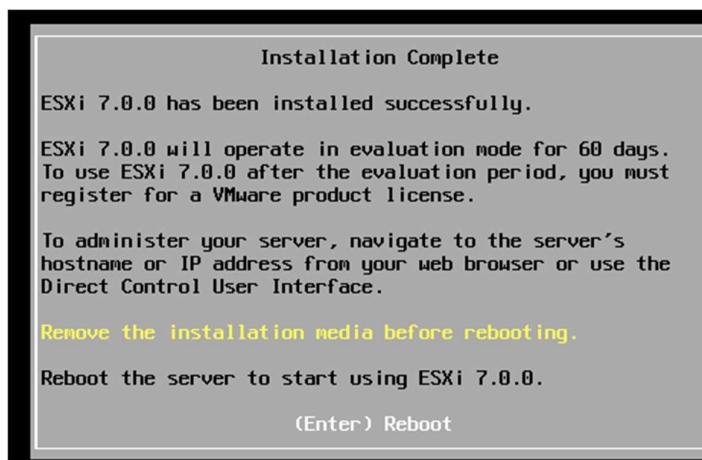
O processo de scan dos discos se inicia:



Com o processo da seleção de disco, tipo de teclado e senha administrativa (ROOT), posteriormente é informado que os dados no disco selecionado serão alterados por particionamento necessário.



Com a finalização concluída o processo de reboot pós remoção da mídia ISO do drive.



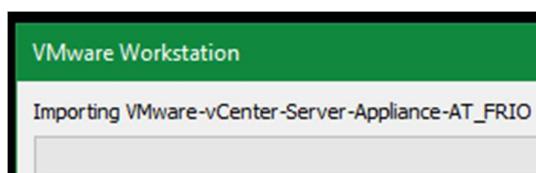
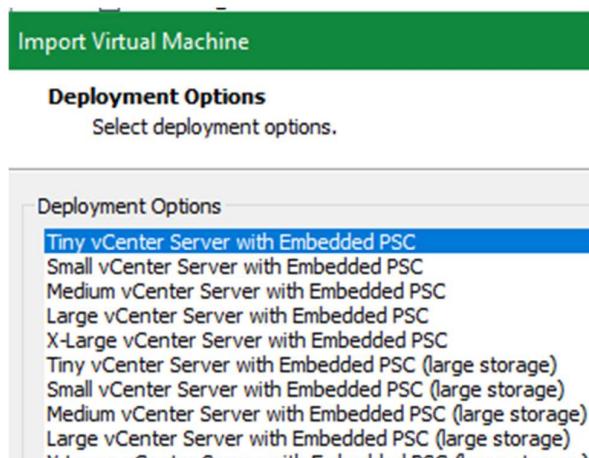
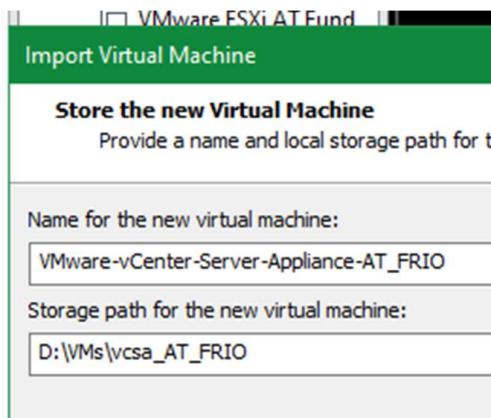
### O primeiro boot e verificação do serviço;

Acessando o ambiente posterior ao reboot:

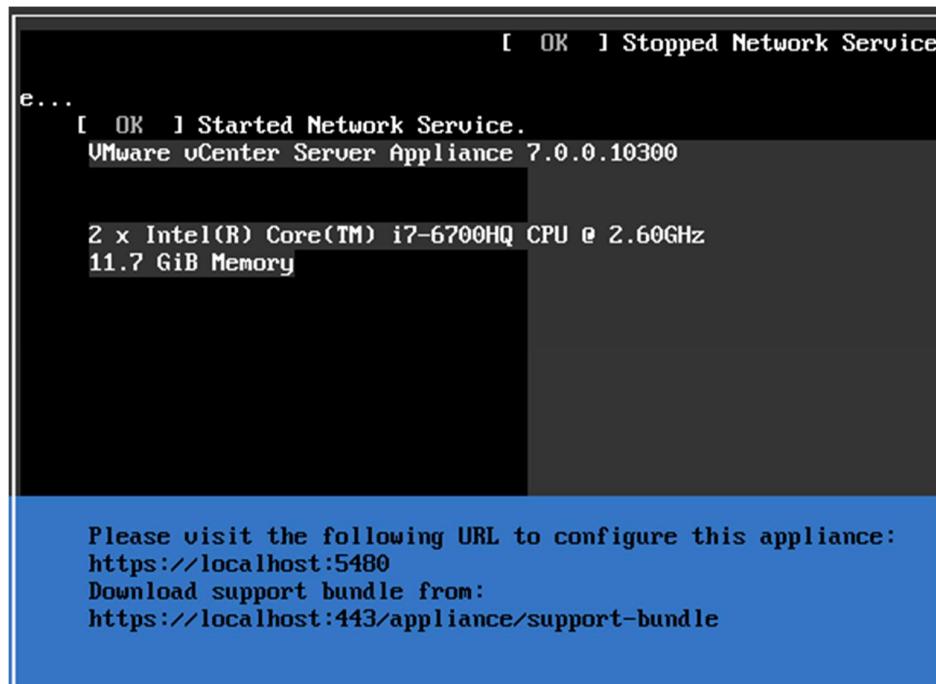


## VCSA Appliance install;

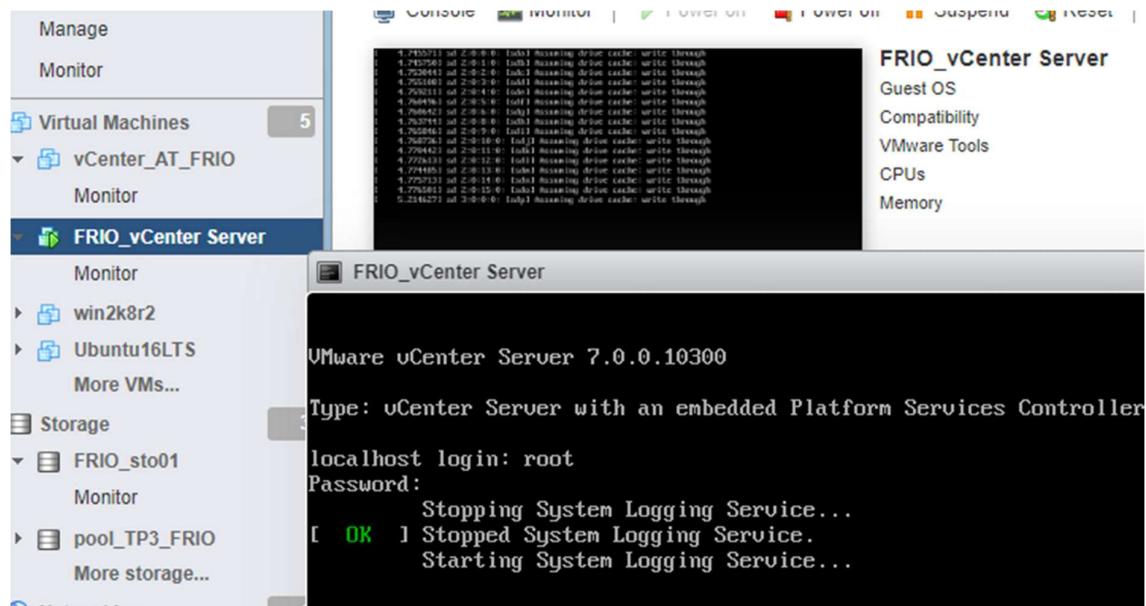
O processo de instalação OVF se dá com a utilização da ISO na criação de uma máquina virtual:



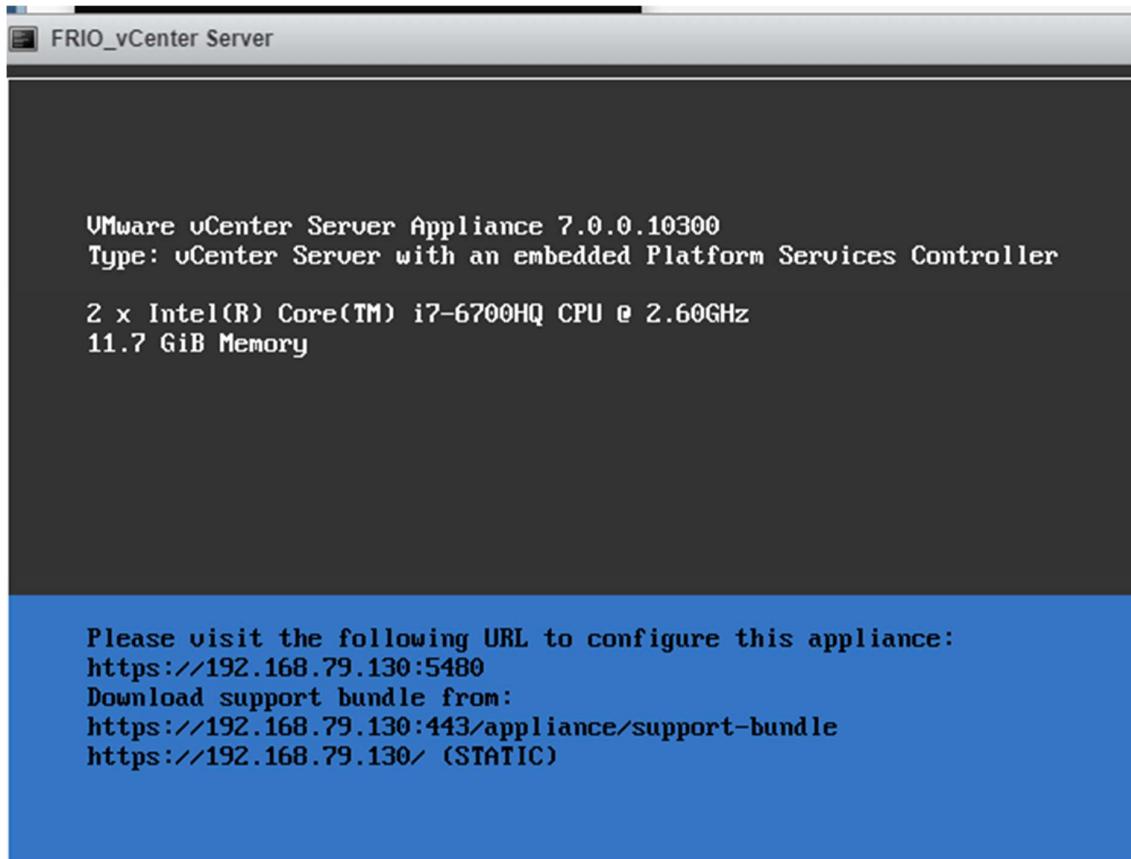
## O processo de subida dos serviços VCSA:



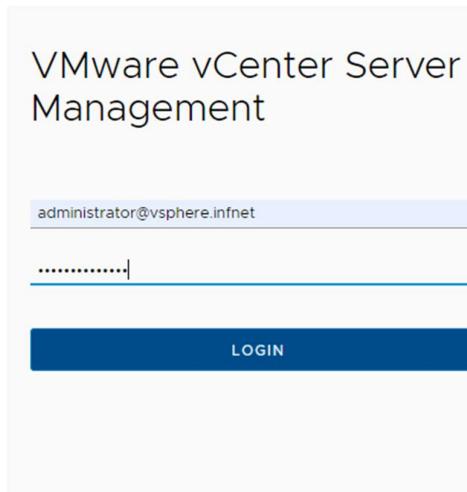
\*os arquivos da ISO VCSA em minha máquina foram corrompidos pós instalação em aula, tenho o servidor funcionando, mas não consegui documentar todo o processo de instalação.



**Validação de serviço:**



**Serviço VCSA UI acessível:**



Validação do método de instalação como appliance host:

The screenshot shows the vCenter Server Management interface. The top bar displays the URL as https://192.168.79.130:5480/ui/summary. The main content area is titled "Summary". On the left, there's a sidebar with links: Summary (which is selected), Monitor, Access, Networking, Firewall, and Time. The main panel shows a small icon of three cubes. To its right, detailed information is provided:  
Hostname: localhost  
Type: vCenter Server with an embedded Platform Services Controller  
Product: VMware vCenter Server Appliance  
Version: 7.0.0.10300  
Build number: 16189094

Aguardando o serviço subir por completo:

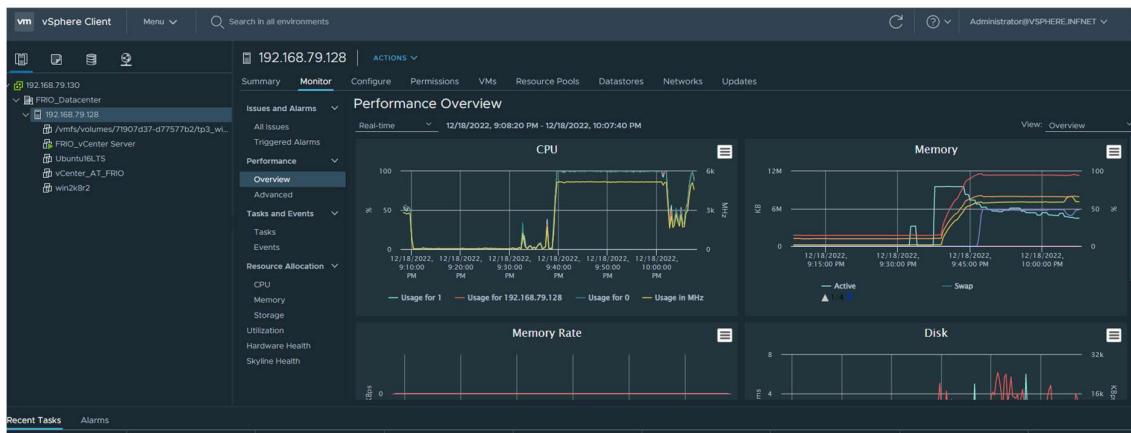
The screenshot shows a browser window with the URL https://192.168.79.130/ui/. The main content area displays the message: "The vSphere Client web server is initializing". Below this message, a smaller text states: "The vSphere Client web server is still initializing. Please try again shortly." At the bottom, there is a link: "VMware vSphere Documentation and Support".

## The vSphere Client web server is initializing

The vSphere Client web server is still initializing. Please try again shortly.

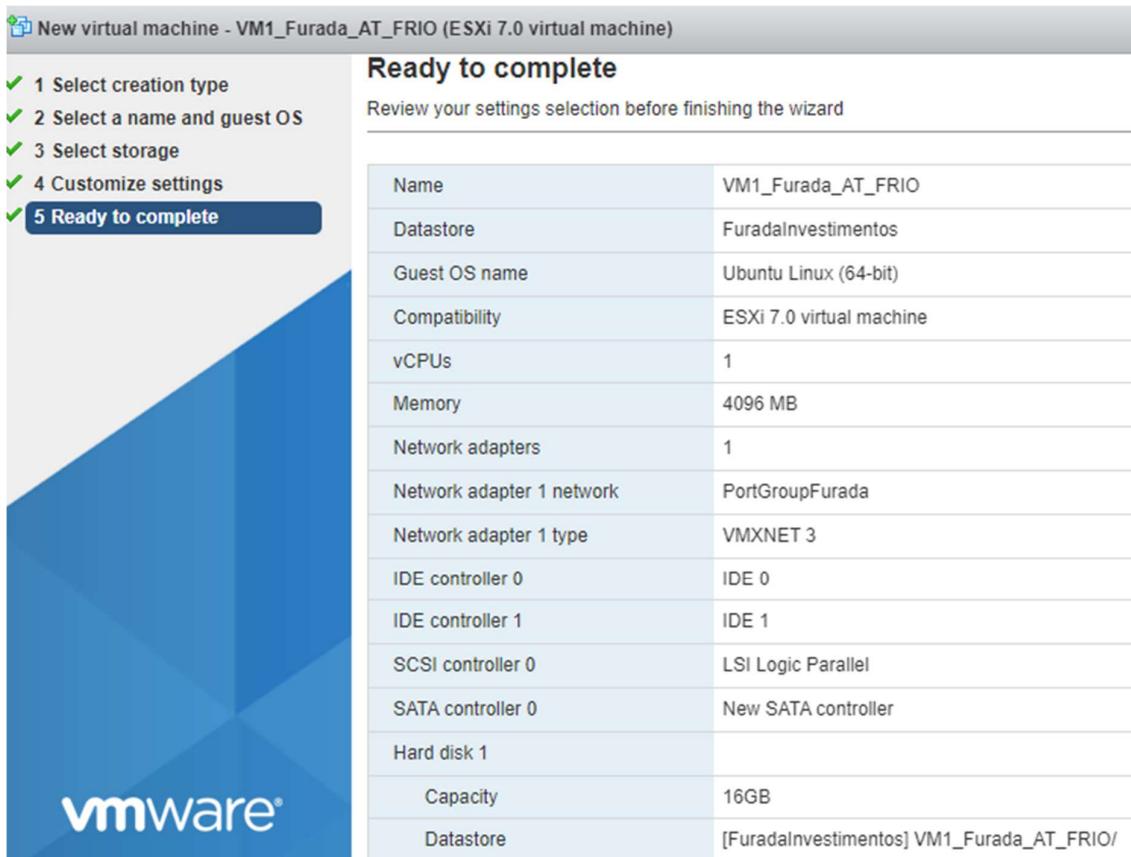
[VMware vSphere Documentation and Support](#)

## Validação da UI de administração:



## Criação das VMs para instalação e configuração do Ansible;

Será utilizada na criação das VMs todas as configurações de discos e rede disponibilizadas no ESXi.



## **Repassando os playbooks para Docker, LAMP e Wordpress;**

*\*foi notado que o Ubuntu 16LTS se comporta mal as vezes para instalar alguns componentes, então a imagem dos alvos foi trocada para o CentOS9 Stream.*

### **Execução do Playbook Docker:**

Para instalar o Docker no host alvo, foi criado um inventário, e agrupado como [Docker] e solicitado pela permissão de root para execução:

```
[root@ansihost projetoAIAdocker]# vi dockerinstall.yml
[root@ansihost projetoAIAdocker]# ansible-playbook dockerinstall.yml -i inventory.txt --ask-become-pass
BECOME password:

PLAY [Instalando Docker] ****
TASK [Gathering Facts] ****
ok: [target2]
TASK [Adicionando repositorio Docker] ****
changed: [target2]
TASK [Instalando Docker package] ****
changed: [target2]
TASK [Starting docker service] ****
changed: [target2]
TASK [Docker SDK para Python3] ****
changed: [target2]
PLAY RECAP ****
target2 : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
[root@ansihost projetoAIAdocker]#
```

Com o final da execução do playbook de instalação do Docker, foi validado e \*testada a instalação do Docker no host alvo (target2):

```
[root@ansitarget2 ~]# docker ps -a
CONTAINER ID   IMAGE      COMMAND   CREATED     STATUS      PORTS      NAMES
[root@ansitarget2 ~]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:ffb13da98453e0f04d33a6eee5bb8e46ee50d08ebe17735fc0779d0349e889e9
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

\*A validação foi a verificação do Docker vazio e execução do pull do container hello-world

### Instalação LAMP/wordpress:

Posterior a validação do Docker foi executado o playbook para instalar os componentes necessários para o wordpress.

```
[root@ansihost projetoAIAdocker]# ansible-playbook dockerwordpress.yml -i inventory.txt --ask-become-pass
BECOME password:
PLAY [Deploy WordPress usando container] ****
TASK [Gathering Facts] ****
ok: [target2]
TASK [Start MySQL container] ****
changed: [target2]
TASK [Start WordPress container] ****
changed: [target2]
PLAY RECAP ****
target2 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
[root@ansihost projetoAIAdocker]#
```

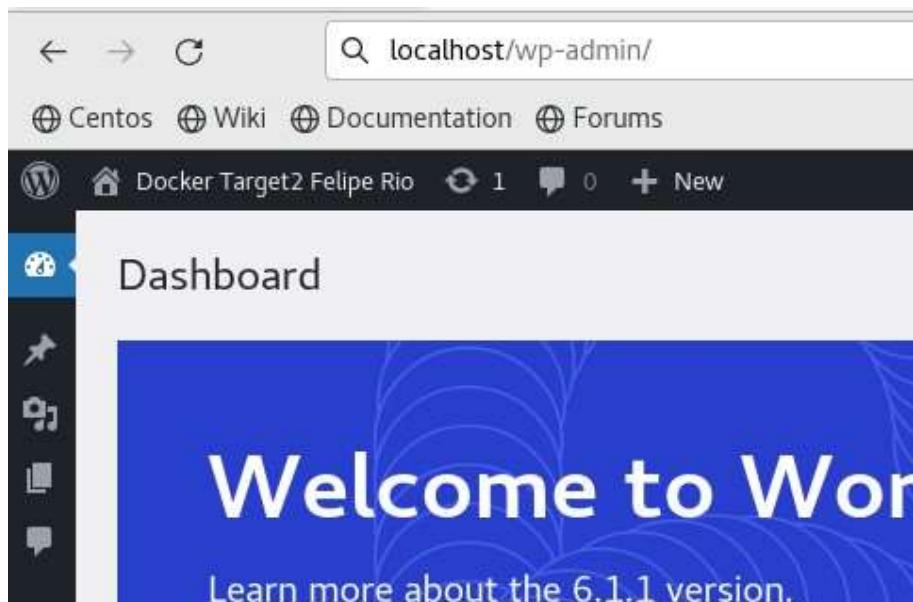
## Validação Wordpress:

Verificado no servidor o deploy e start com sucesso.

```
[root@ansitarget2 ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8e6d41c61aa7 wordpress:latest "docker-entrypoint.s..." 12 seconds ago Up 19 seconds 0.0.0.0:80->80/tcp wordpress
18ab752cc233 mysql:5.7 "docker-entrypoint.s..." 39 seconds ago Up 37 seconds 3306/tcp, 33060/tcp mysql
fd408fd16719 hello-world "/hello" About a minute ago Exited (0) About a minute ago thirsty_mccarthy

[root@ansitarget2 ~]# docker stats
CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS
8e6d41c61aa7 wordpress 0.0% 39.98MiB / 1.712GiB 2.28% 2.19KB / 0B 61.9MB / 0B 6
18ab752cc233 mysql 0.01% 202MiB / 1.712GiB 11.52% 2.45KB / 0B 152MB / 730MB 27
CONTAINER ID NAME CPU % MEM USAGE / LIMIT MEM % NET I/O BLOCK I/O PIDS
```

Posterior a configuração e publicação da validação no servidor.



## **TP9**

### **Tempo de projeto:**

De acordo com o estudo, o tempo do projeto não foi o suficiente. Como já era uma empresa em funcionamento, todas as janelas de manutenção precisavam de aprovação prévia com 7 dias e nunca encaixavam no cronograma. Por regras da empresa toda janela de manutenção depende da presença de alguém da infra interna e o fornecedor, com algumas janelas de manutenção dentro do projeto canceladas, porque o fornecedor não tinha acesso físico a infra ou a documentação de cabeamento do datacenter faltava alguma informação importante na MOP. E o último ponto de aprendizado é que para uma empresa já em funcionamento as reuniões precisam ser mais próximas, ao invés de 15 dias, semanalmente junto da defesa da necessidade das janelas de manutenção programadas.

### **Planejamento de recursos:**

De acordo com o planejamento e anamnese de acessos, a alocação de hardware do projeto não está suportando a carga de acessos e input de dados sendo que foram listados e com isso possíveis problemas futuros aparecem:

- Pouca memória e processamento no servidor que controla o container do banco de dados, o container precisa ser movido para outro vhost;
- Problemas de térmica, pois os racks extras para a expansão não foram avaliados pela equipe predial que cuida dos HVAC;
- Foi verificado que a segregação dos discos precisa de um controle maior de cota, pois o espaço está sendo mal utilizado para arquivos particulares, quebrando uma das regras básicas de S.I.;
- Custo crescente do contrato com o fornecedor do hardware, agora que a carga de acesso é alta, a falha dos discos se tornou mais evidente e frequente;
- Problemas de redundância elétrica no futuro, pois com a possível expansão de racks, o nobreak atual não segura a carga elétrica até a entrada do gerador.

### **Funcionalidades do ambiente:**

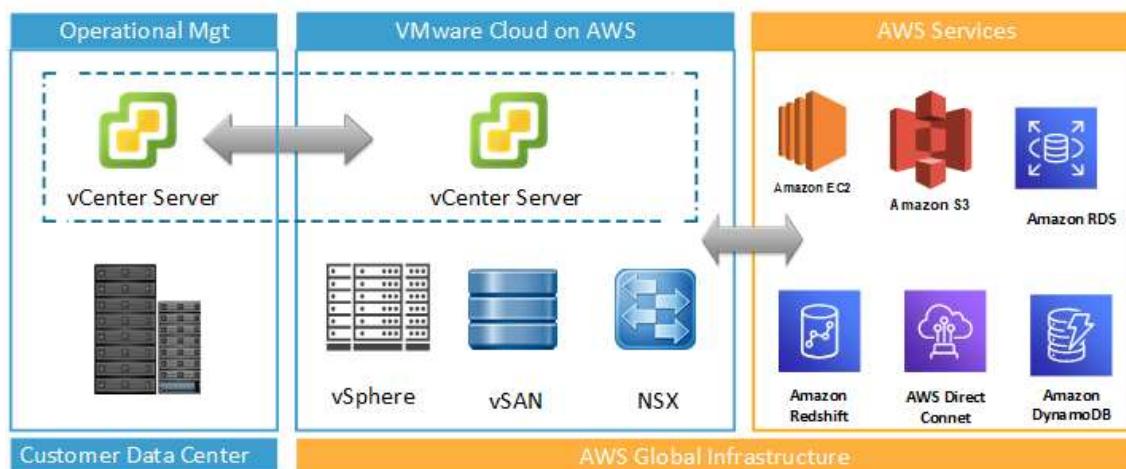
As funcionalidades e facilidades da virtualização ficaram a contento da equipe de DevOps em um certo ponto e dos stakeholders que tem o report direto deles, pois o contato direto com uma ferramenta tão prática adicionou velocidade ao processo, mas como todo o ciclo foi mal dimensionado e agora está sendo percebido o custo alto, a migração para a nuvem pública, onde a responsabilidade da infra sai da empresa, é eminente.

### **Melhorias futuras:**

Já discutido que em algum ponto da expansão o espaço físico teria um fim, e iria começar a limitar a capacidade de processamento e acessos do ambiente, de acordo os stakeholders aceitaram ouvir a proposta da nuvem pública.

Com todos os pontos de problemas mapeados e a tecnologia andando, no início do projeto foi salientado que o produto ESXi era o de melhor escolha, pela facilidade da proximidade da migração para a nuvem publica AWS, com a separação dos servidores e serviços descritos abaixo:

## VMware Cloud on AWS Services



Com o treinamento da equipe de DevOPs e suporte a infra executando todas as vantagens do serviço em nuvem publica poderão ser usufruídos em relação a virtualização local, onde teremos:

1. Escalabilidade: A AWS permite o aumento ou redução da capacidade dos recursos de computação, armazenamento e rede de forma rápida e fácil, de acordo com as necessidades atuais. Onde a empresa não precisa mais se preocupar em provisionar recursos físicos e lógicos com antecedência ou lidar com capacidade ociosa, térmica e elétrica;
2. Redução de custos: A AWS oferece modelos flexíveis que permitem o pagamento apenas pelos recursos utilizados. Removendo a responsabilidade com a manutenção e atualização de hardware, o que representa uma economia significativa de tempo e dinheiro;
3. Segurança: A AWS oferece recursos de segurança e conformidade de classe empresarial, incluindo criptografia de dados, monitoramento de segurança, gerenciamento de identidade e acesso e conformidade com padrões regulatórios mundiais;
4. Confiabilidade: A AWS oferece alta disponibilidade e escalabilidade automática, garantindo que as instâncias estejam sempre disponíveis manobrando a carga com picos de tráfego sem down-time ou gargalos;

**Link repositório:**

Link público para consultas e validações:

<https://github.com/felipe-infnet/projetoAIA>

<https://github.com/felipe-infnet/projetoAIAdocker>

**Fontes:**

[https://access.redhat.com/documentation/en-us/red\\_hat\\_openstack\\_platform/10/html/architecture\\_guide/components](https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/10/html/architecture_guide/components)

[https://docs.ansible.com/ansible/latest/getting\\_started/get\\_started\\_playbook.html](https://docs.ansible.com/ansible/latest/getting_started/get_started_playbook.html)

<https://vmusketeers.com/2015/03/22/windows-vcenter-vs-vcenter-server-virtual-appliance-vcsavcva/>

<https://www.makarenalabs.com/ansible-for-it-automation-wordpress-as-an-example/>

<https://www.altaro.com/vmware/vsphere-networking-basics-part-1/#:~:text=The%20VMkernel%20network%20interface%2C%20adapter,Tolerance%2C%20Management%20or%20perhaps%20vSAN>

<https://wa.aws.amazon.com/wellarchitected/2020-07-02T19-33-23/wat.concept.playbook.en.html>

[https://docs.ansible.com/ansible/latest/playbook\\_guide/playbooks\\_intro.html?extIdCarryOver=true&sc\\_cid=701f2000001OH7EAAW](https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_intro.html?extIdCarryOver=true&sc_cid=701f2000001OH7EAAW)

<https://www.techtarget.com/searchdatacenter/definition/SDDC-software-defined-data-center>

[https://docs.ansible.com/ansible/latest/reference\\_appendices/config.html](https://docs.ansible.com/ansible/latest/reference_appendices/config.html)

[https://docs.ansible.com/ansible/latest/reference\\_appendices/playbooks\\_keywords.html](https://docs.ansible.com/ansible/latest/reference_appendices/playbooks_keywords.html)

[https://docs.ansible.com/archive/ansible/2.3/playbooks\\_directives.html#play](https://docs.ansible.com/archive/ansible/2.3/playbooks_directives.html#play)

<https://linuxbuz.com/linuxhowto/what-is-yaml-in-ansible-and-how-it-works>

<https://stackoverflow.com/questions/3555681/why-are-my-php-files-showing-as-plain-text>

<https://www.linuxteacher.com/how-to-install-php-8-0-on-centos-8/>

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/managing-deploy-keys>

<https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/sudo-mode>

<https://opensource.com/article/19/7/create-pull-request-github>

<https://stackoverflow.com/questions/9270734/ssh-permissions-are-too-open>

<https://docs.github.com/en/get-started/getting-started-with-git/managing-remote-repositories>

<https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/creating-a-pull-request>

<https://www.portainer.io/docker-swarm-container-management-platform-gui>

<https://www.portainer.io/docker-swarm-container-management-platform-gui>

<https://wordpress.org/about/requirements>

[https://docs.ansible.com/ansible/latest/getting\\_started/get\\_started\\_playbook.html](https://docs.ansible.com/ansible/latest/getting_started/get_started_playbook.html)

<https://www.makarenalabs.com/ansible-for-it-automation-wordpress-as-an-example/>