

ESCOLA SUPERIOR DE TECNOLOGIA DA INFORMAÇÃO



Graduação:	GESTÃO DA TECNOLOGIA DA INFORMAÇÃO
Bloco:	Arquitetura de Infraestrutura e Aplicações
Matéria:	Projeto de Bloco
Referência:	TP4 e TP5
Turma:	GRLGTI01C2-N2-L1
Aluno:	Silney Alvarenga Lopes

Sumário

1. Introdução.....	3
2. TP4 e TP5.....	3
3. Conclusão	9

1. Introdução

Objetivo neste documento é responder as questões colocadas no TP4 e TP5 do Projeto de Bloco.

2. TP4 e TP5

2.1. Preparando ambiente

A empresa observou, que é uma estratégia interessante construir mini ERPs com WordPress, segregado por departamentos e a equipe de programação cuidar da intercomunicação, backup e funcionamento entre eles (como módulos), devido ao baixo custo com a utilização de docker e recurso de máquina.

Baseado nisso, foi preciso desenvolver um código Ansible que automatize a instalação de um banco de dados e um wordpress para futuras implantações de mini ERPs de homologação e produção.

Começamos aqui o desenvolvimento:

2.2. Instalando o Docker

Foi criada uma VM dedicada ao Docker, os comandos utilizados para a instalação foram:

- `sudo apt update`
- `sudo apt install apt-transport-https ca-certificates curl software-properties-common`
- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg`
- `echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null`
- `sudo apt update`
- `sudo apt install docker-ce`

Evidências da Instalação:

```
silney@silney-infnet:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-03-23 23:39:37 -03; 18s ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Main PID: 3397 (dockerd)
      Tasks: 8
     Memory: 25.7M
        CPU: 895ms
    CGroup: /system.slice/docker.service
            └─3397 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

mar 23 23:39:36 silney-infnet dockerd[3397]: time="2023-03-23T23:39:36.760841562-03:00" level=info msg="[core] [Channel #4 SubC
mar 23 23:39:36 silney-infnet dockerd[3397]: time="2023-03-23T23:39:36.761476929-03:00" level=info msg="[core] [Channel #4] Cha
mar 23 23:39:36 silney-infnet dockerd[3397]: time="2023-03-23T23:39:36.908418910-03:00" level=info msg="Loading containers: sta
mar 23 23:39:37 silney-infnet dockerd[3397]: time="2023-03-23T23:39:37.242150705-03:00" level=info msg="Default bridge (/docke

silney@silney-infnet:~$ docker --version
Docker version 23.0.1, build a5ee5b1
```

2.3. Configurações VM Ansible

Criei uma pasta chamada TP4_TP5 vazia no GitHub e fiz o git clone para baixa-la na VM de Ansible.

```
silney@silney-infnet:~$ ls
Desktop Documents Downloads Music PBInfnet Pictures Public snap Templates Videos
silney@silney-infnet:~$ git clone https://github.com/silneylopes/TP4_TP5
Cloning into 'TP4_TP5'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
silney@silney-infnet:~$ ls
Desktop Documents Downloads Music PBInfnet Pictures Public snap Templates TP4_TP5 Videos
silney@silney-infnet:~$
```

Instalei o modulo Python Docker

```
sudo apt install python3-pip
pip3 install docker
```

```
silney@silney-infnet:~/TP4_TP5$ pip3 install docker
Defaulting to user installation because normal site-packages is not writeable
Collecting docker
  Downloading docker-6.0.1-py3-none-any.whl (147 kB)
    147.5/147.5 KB 3.9 MB/s eta 0:00:00
Requirement already satisfied: packaging>=14.0 in /usr/lib/python3/dist-packages (from docker) (21.3)
Collecting websocket-client>=0.32.0
  Downloading websocket_client-1.5.1-py3-none-any.whl (55 kB)
    55.9/55.9 KB 5.6 MB/s eta 0:00:00
Requirement already satisfied: urllib3>=1.26.0 in /usr/lib/python3/dist-packages (from docker) (1.26.5)
Collecting requests>=2.26.0
  Downloading requests-2.28.2-py3-none-any.whl (62 kB)
    62.8/62.8 KB 7.2 MB/s eta 0:00:00
Collecting charset-normalizer<4,>=2
  Downloading charset_normalizer-3.1.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (199
    199.3/199.3 KB 8.0 MB/s eta 0:00:00
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3/dist-packages (from requests>=2.26.0->
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-packages (from requests>=2.
Installing collected packages: websocket-client, charset-normalizer, requests, docker
WARNING: The script wsdump is installed in '/home/silney/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-scr
WARNING: The script normalizer is installed in '/home/silney/.local/bin' which is not on PATH.
```

Altereí o arquivo hosts da pasta Ansible inserindo o grupo docker com o endereço do servidor virtual que irá receber o Docker e suas variáveis.

```
silney@silney-infnet:~$ cd /etc/ansible/
silney@silney-infnet:/etc/ansible$ nano hosts
```

```
[docker]
192.168.31.254
[docker:vars]
ansible_user=silney
ansible_password=maicjc
ansible_become_password=maicjc
ansible_python_interpreter=/usr/bin/python3
^G Help      ^O Write Out  ^W Where Is
^X Exit      ^R Read File  ^_ Replace
```

Construção do Playbook

```
silney@silney-infnet:~/TP4_TP5$ nano playbook.yml
```

```

--
- hosts: docker
  vars:
    db_volume: mariadb
    wordpress: wordpress
  tasks:
    - name: Deploy MariaDB server
      docker_container:
        image: mariadb
        name: mariadb
        volumes:
          - "{{db_volume}}:/var/lib/mysql"
        env:
          MYSQL_ROOT_PASSWORD: password
          MYSQL_PASSWORD: infnet
          MYSQL_DATABASE: db_TP4_TP5
          MYSQL_USER: infnet

    - name: Deploy WordPress
      docker_container:
        image: wordpress
        name: wordpress
        restart_policy: always
        ports:
          - "8080:80"
        links:
          - "{{db_volume}}:/var/lib/mysql"
        volumes:
          - "{{wordpress}}:/var/www/html"
        env:
          MYSQL_PASSWORD: infnet
          MYSQL_DATABASE: db_TP4_TP5
          MYSQL_USER: infnet
          MYSQL_HOST: mariadb

```

Execução do código na VM Ansible para instalar na VM Docker
(192.168.31.254)

```

silney@silney-infnet:~/TP4_TP5$ export ANSIBLE_HOST_KEY_CHECKING=False
silney@silney-infnet:~/TP4_TP5$ ansible-playbook playbook.yml -l docker --ask-become-pass
BECOME password:

PLAY [docker] *****

TASK [Gathering Facts] *****
ok: [192.168.31.254]

TASK [Deploy MariaDB server] *****
changed: [192.168.31.254]

TASK [Deploy WordPress] *****
changed: [192.168.31.254]

PLAY RECAP *****
192.168.31.254      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Resultado na VM Docker


```
For more help on how to use Docker, head to https://docs.docker.com/go/guides/

silney@silney-infnet:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
silney@silney-infnet:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
aebe85500a30   wordpress "docker-entrypoint.s..." 17 seconds ago   Up 14 seconds   0.0.0.0:8080->80/tcp   wordpress
589b33d63eed   mariadb   "docker-entrypoint.s..." 21 seconds ago   Up 18 seconds   3306/tcp             mariadb
silney@silney-infnet:~$
```

Teste via browser


seguro 192.168.31.254:8080/wp-admin/setup-config.php?step=1&language=pt_BR

lar WLM



Abaixo você deve digitar os detalhes de conexão do banco de dados. Se você não tem certeza sobre isso, entre em contato com sua hospedagem.

Nome do banco de dados	<input type="text" value="db_TP4_TP5"/>	O nome do seu banco de dados que você deseja utilizar com o WordPress.
Nome de usuário	<input type="text" value="infnet"/>	Usuário do seu banco de dados.
Senha	<input type="text" value="infnet"/>	Senha do seu banco de dados.
Servidor do banco de dados	<input type="text" value="mariadb"/>	Você deve ser capaz de obter esta informação no seu servidor de hospedagem, caso localhost não funcione.
Prefixo da tabela	<input type="text" value="wp_infnet_pb"/>	Se quiser rodar várias instalações WordPress em um único banco de dados, mude isto.



Muito bem! Você concluiu esta parte da instalação. Agora o WordPress pode se comunicar com seu banco de dados. Se você estiver pronto, é hora de...



Bem-vindo (a)

Bem-vindo (a) à famosa instalação do WordPress em cinco minutos! Basta preencher as informações abaixo e você estará a poucos passos de usar a plataforma de publicação mais extensível e poderosa do mundo.

Informação necessária

Forneça as seguintes informações. Não se preocupe, você pode alterar estas configurações mais tarde.

Título do site

Nome de usuário

Nomes de usuário podem ter somente caracteres alfanuméricos, espaços, sublinhados, hífens, pontos e o símbolo @.

Senha [Show](#)
Very weak

Importante: Você precisará dessa senha para entrar. Guarde-a em um local seguro.

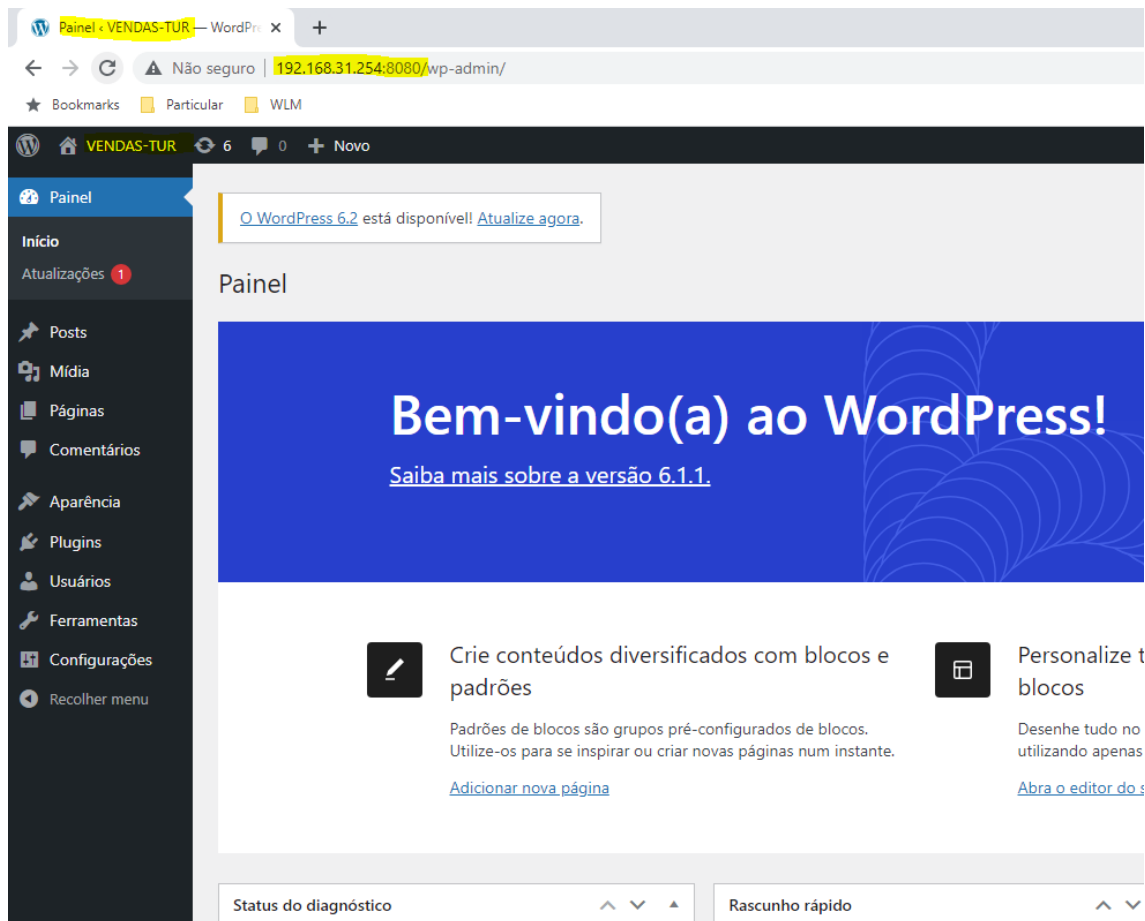
Confirme a senha ☒ Confirmar o uso de uma senha fraca

O seu e-mail

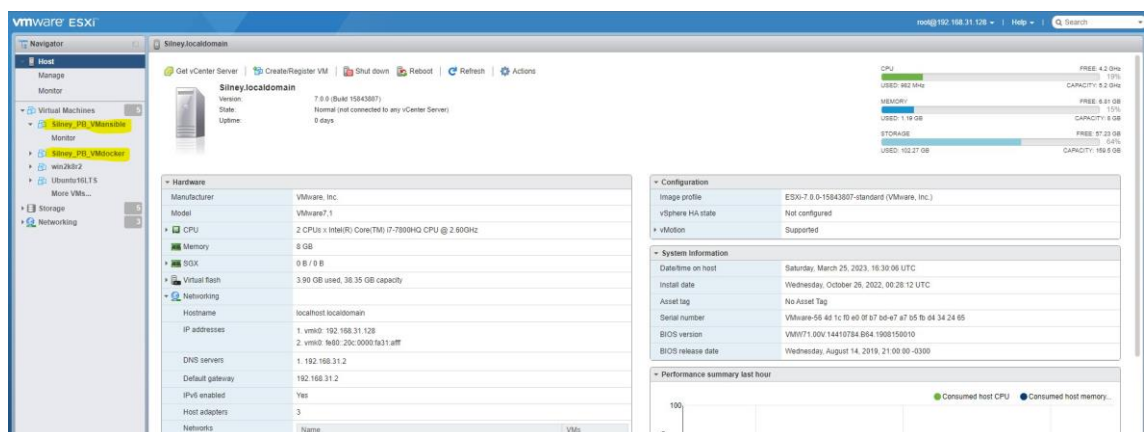
Verifique o seu endereço de e-mail antes de prosseguir.

Visibilidade nos motores de busca ☒ Evitar que mecanismos de busca indexem este site
Cabe aos mecanismos de busca atender esta solicitação.

[Instalar WordPress](#)



Tudo Executado em uma Nuvem Privada da VMware ESXi



Para finalizar estes TPs, executamos um git push para enviar os códigos para o repositório TP4_TP5 do GitHub


```
silney@silney-infnet:~/TP4_TP5$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 676 bytes | 676.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/silneylopes/TP4\_TP5.git
  2dec170..9b3ab17  main -> main
silney@silney-infnet:~/TP4_TP5$
```

Segue link: https://github.com/silneylopes/TP4_TP5

3. Conclusão

Nestes TPs, vimos o Ansible como ferramenta de configuração e automatização para servidores, Docker e o GitHub como ferramenta de documentação, versionamento e colaboração de códigos.