



Instituto Infnet

CURSO DE GRADUAÇÃO EM REDE DE COMPUTADORES

## **PROJETO DE BLOCO**

### **Arquitetura e Infraestrutura de Aplicações**

#### **AVALIAÇÃO TP5**

*João Ricardo Cesar Teixeira de Araujo*

Turma: RDC2016

## ÍNDICE

<b>1. INTRODUÇÃO .....</b>	<b>3</b>
<b>2. SOBRE PLAYBOOK ANSIBLE .....</b>	<b>3</b>
<b>3. INSTALAÇÃO DO SISTEMA OPERATIVO .....</b>	<b>4</b>
3.1. SOBRE O UBUNTU 18.04 .....	4
3.2. SOBRE O ORACLE VM VIRTUAL BOX.....	4
3.3. INSTALAÇÃO DO UBUNTU .....	5
<b>4. INSTALAÇÃO DO ANSIBLE .....</b>	<b>10</b>
<b>5. INSTALAÇÃO DO DOCKER .....</b>	<b>11</b>
<b>6. O PROJETO .....</b>	<b>13</b>
6.1. VISÃO GERAL DA ESTRUTURA DE PASTAS .....	13
6.2. INVENTORY .....	14
6.3. ANSIBLE.CFG .....	14
6.4. PLAYBOOKS:.....	14
6.5. ROLES .....	15
6.6. CONFIG-BASIC:.....	16
6.7. DOCKER.....	17
<b>7. CONSIDERAÇÕES FINAIS .....</b>	<b>20</b>
<b>8. REFERENCIAS TÉCNICAS .....</b>	<b>20</b>

## 1. INTRODUÇÃO

Esse TP visa a desenvolver a instalação e automatização de aplicações, através do Playbook Ansible, onde irei apresentar o passo a passo da configuração, execução do processo e resultado. Irei a apresentar a criação e configuração de containers Docker.

Irei apresentar e disponibilizar em um repositório GitHub, com a URL para o repositório.

## 2. SOBRE PLAYBOOK ANSIBLE



Ansible é uma ferramenta de automação em processos de instalação remota em vários servidores, a partir de um único ponto.

É possível realizar a instalação de uma ou várias aplicações customizadas.

Dentro do processo, existem as Tasks, que são as tarefas no processo, como por exemplo “Instalar o .net 4.8” ou “Adicionar o user *joao.araujo* no grupo de Administradores Local do servidor”, etc.

Para cada conjunto de execuções dessas tarefas (Tasks) temos o Play, que é composta por essas atividades.



O Playbook é o conjunto de Plays para concluir a automatização do processo.

### 3. INSTALAÇÃO DO SISTEMA OPERATIVO

Para preparação do ambiente, irei instalar o Ubuntu na versão 18.04, sendo implementado dois servidores no ambiente virtual, com a ferramenta Oracle VM Virtual Box.

#### 3.1. SOBRE O UBUNTU 18.04

O Ubuntu é um sistema operacional de código aberto que usa o Kernel Linux, baseado no Debian. O Ubuntu foi desenvolvido por uma empresa sul-africana e hoje é um dos sistemas operacionais mais utilizados no mundo.



A versão mais recente do sistema operacional, é a versão 18.04. Dentre as principais novidades dessa versão, destacamos;

- Versão Mini, com as instalações mínimas para o funcionamento do Sistema Operacional;
- Melhorias na instalação de patches no kernel do SO;
- Kernel mais atualizado e estável;
- Melhoria e aprimoramento na interface gráfica;
- Melhoria nos relatórios de utilização do Servidor

#### 3.2. SOBRE O Oracle VM Virtual Box

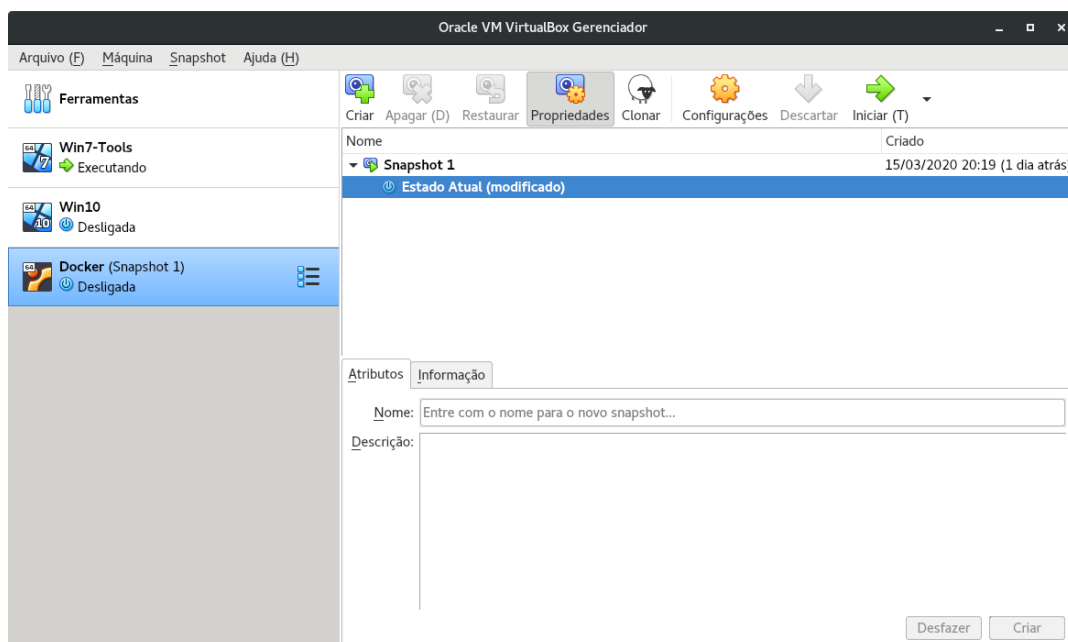
O **Oracle VM Virtual Box** é um software de virtualização, desenvolvido inicialmente pela Innotek, hoje a SUN é detentora dos direitos, e a empresa responsável por seu desenvolvimento. A aplicação vista criar ambiente com máquinas, rede, datastores virtuais.



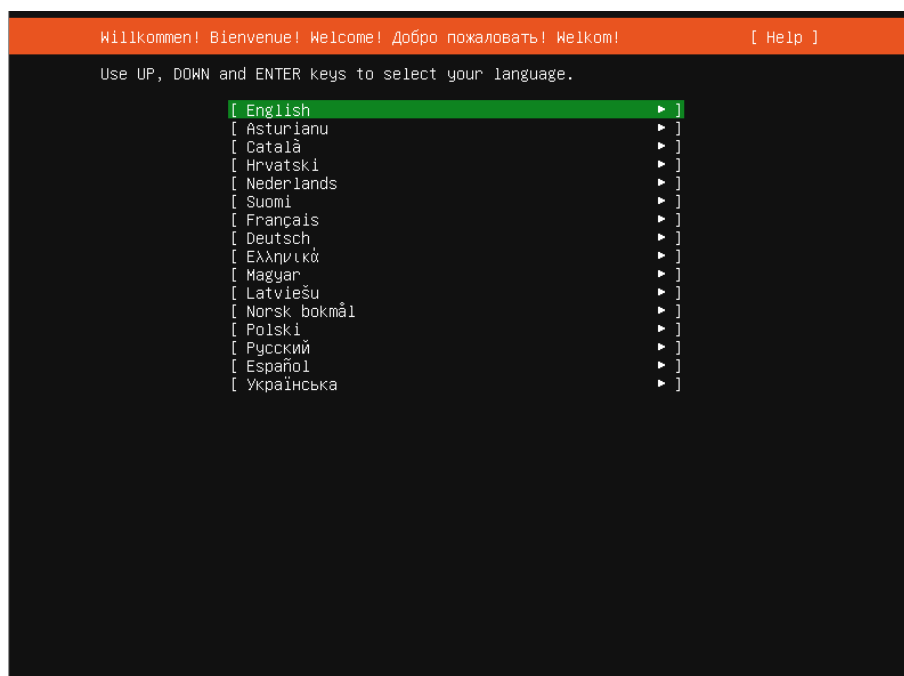
O **Oracle VM Virtual Box** será o software que irá ser usado na criação do ambiente de LAB para o gerenciamento das Máquinas virtuais.

### 3.3. INSTALAÇÃO DO UBUNTU

Segue abaixo a instalação do servidor Linux Ubuntu versão 18.04.



A instalação foi realizada no idioma inglês.



Foi selecionado o teclado em inglês.

```
Keyboard configuration [ Help ]

Please select your keyboard layout below, or select "Identify keyboard" to
detect your layout automatically.

Layout: [ English (US) ▼ ]

Variant: [ English (US) ▼ ]

[ Identify keyboard ]

[ Done ]
[ Back ]
```

Foi configurada a rede.

```
Network connections [ Help ]

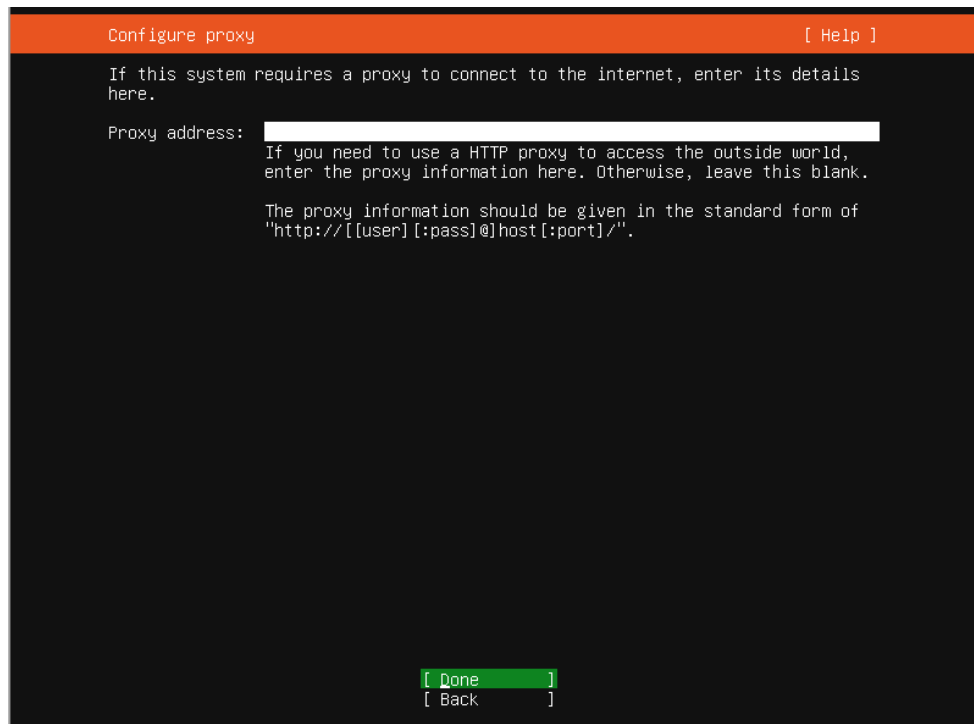
Configure at least one interface this server can use to talk to other machines,
and which preferably provides sufficient access for updates.

NAME  TYPE  NOTES
[ enp0s3 eth - ► ]
DHCPv4 10.0.2.15/24
08:00:27:fe:94:40 / Intel Corporation / 82540EM Gigabit Ethernet Controller
(PRO/1000 MT Desktop Adapter)

[ Create bond ► ]

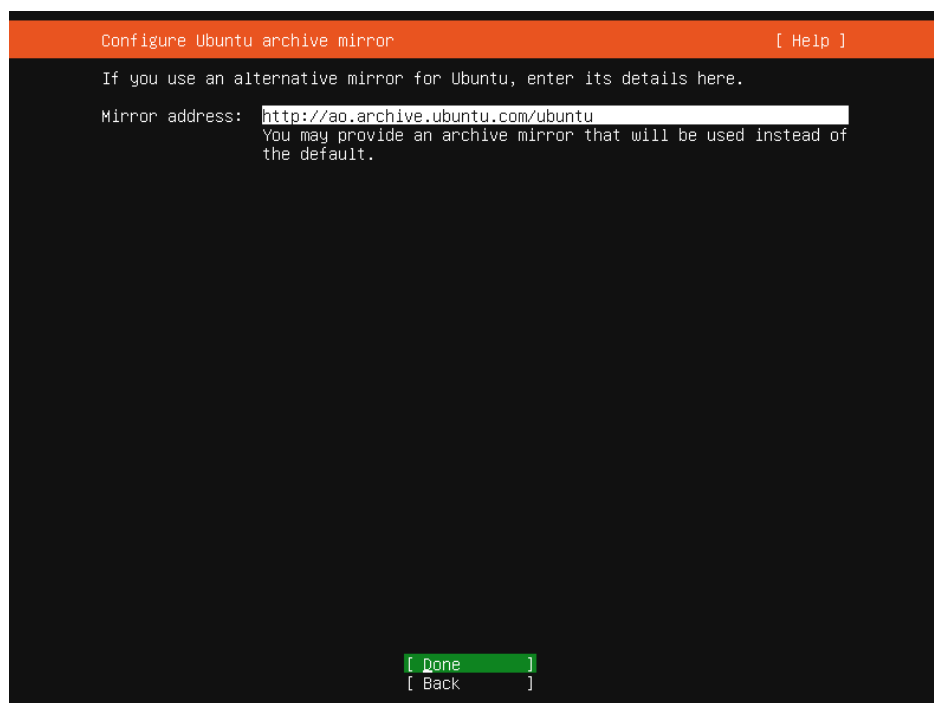
[ Done ]
[ Back ]
```

Foi configurada o proxy.



The screenshot shows a terminal window with a dark background. At the top, there is an orange header bar with the text "Configure proxy" on the left and "[ Help ]" on the right. Below the header, the text "If this system requires a proxy to connect to the internet, enter its details here." is displayed. Underneath, the label "Proxy address:" is followed by a white text input field. To the right of the input field, there is explanatory text: "If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank." and "The proxy information should be given in the standard form of 'http://[user][:pass]@host[:port]/'". At the bottom of the dialog, there are two green buttons: "[ Done ]" and "[ Back ]".

Foi configurado o Mirror Address.



The screenshot shows a terminal window with a dark background. At the top, there is an orange header bar with the text "Configure Ubuntu archive mirror" on the left and "[ Help ]" on the right. Below the header, the text "If you use an alternative mirror for Ubuntu, enter its details here." is displayed. Underneath, the label "Mirror address:" is followed by a white text input field containing the URL "http://ao.archive.ubuntu.com/ubuntu". To the right of the input field, there is explanatory text: "You may provide an archive mirror that will be used instead of the default." At the bottom of the dialog, there are two green buttons: "[ Done ]" and "[ Back ]".

Foi configurado o disco.

```
Filesystem setup [ Help ]

The installer can guide you through partitioning an entire disk either directly
or using LVM, or, if you prefer, you can do it manually.

If you choose to partition an entire disk you will still have a chance to
review and modify the results.

[ Use An Entire Disk ]
[ Use An Entire Disk And Set Up LVM ]
[ Manual ]
[ Back ]
```

```
Filesystem setup [ Help ]

The selected guided partitioning scheme creates the required bootloader
partition on the chosen disk and then creates a single partition covering the
rest of the disk, formatted as ext4 and mounted at '/'.

Choose the disk to install to:

[ VBOX_HARDDISK_VB7cd056cd-536aac00 local disk 10.000G ▶ ]
unused
```

```
Filesystem setup [ Help ]

FILE SYSTEM SUMMARY

MOUNT POINT    SIZE    TYPE    DEVICE TYPE
[ /            9.997G  new ext4  new partition of local disk ▶ ]

AVAILABLE DEVICES

No available devices

[ Create software RAID (md) ▶ ]
[ Create volume group (LVM) ▶ ]

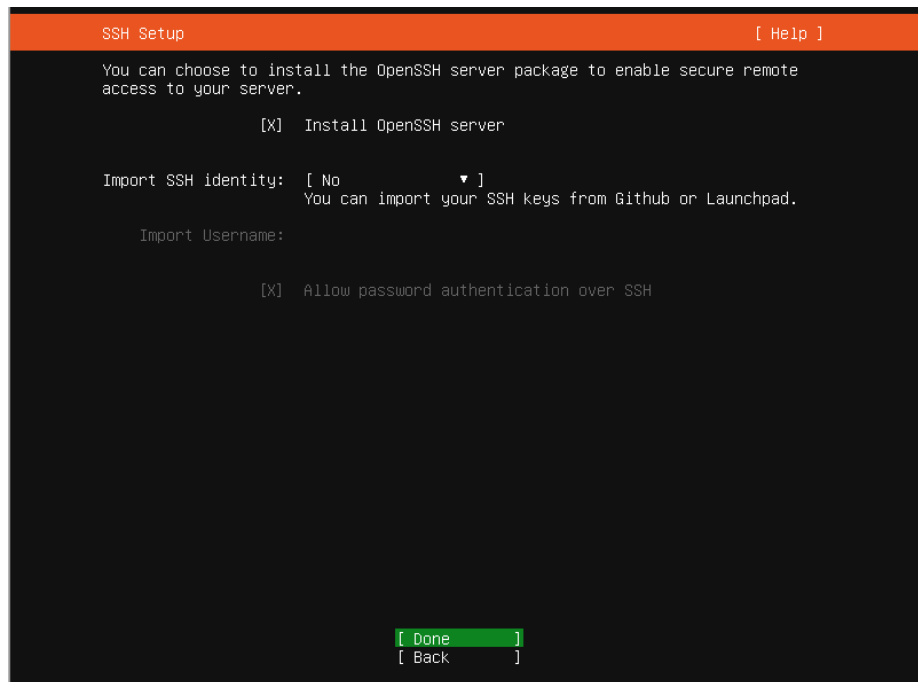
USED DEVICES

DEVICE                                TYPE    SIZE
[ VBOX_HARDDISK_VB7cd056cd-536aac00  local disk 10.000G ▶ ]
partition 1  new, bios_grub              1.000M ▶
partition 2  new, to be formatted as ext4, mounted at / 9.997G ▶

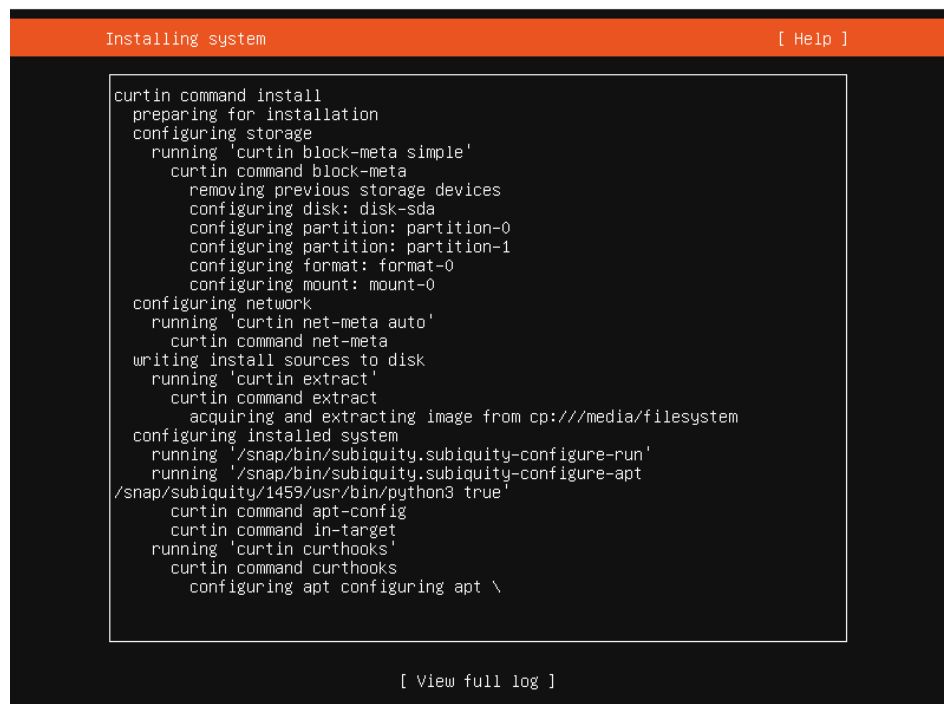
[ Done ]
[ Reset ]
[ Back ]
```



Foi instalado e configurar o SSH.



Foi instalado com sucesso o servidor Ubuntu.



## 4. INSTALAÇÃO DO ANSIBLE

O Ansible foi configurado na Máquina Ubuntu e para a sua instalação, foram utilizados os seguintes comandos:

Para incluir no PPA (arquivo de pacotes pessoais), foi executado o seguinte comando:

```
sudo apt-add-repository ppa:ansible/ansible
```

Após inclusão, foi executado o comando para atualizar o índice de pacotes do sistema:

```
sudo apt update
```

Após atualizar, foi instalado o Ansible com o seguinte comando:

```
sudo apt install ansible
```

Após instalação, podemos verificar que o servidor está com Ansible em funcionamento.

```
root@docker:/etc/ansible# ansible docker -m ping
10.0.2.15 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
root@docker:/etc/ansible#
```

Após instalação, foi realizada as configurações básicas do Ansible, seguindo o que foi descrito na TP4:

```
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

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

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

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

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

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com
[docker]
10.0.2.15

[docker:vars]
ansible_python_interpreter=/usr/bin/python3

-- INSERT --
```

46,1

Bot

## 5. INSTALAÇÃO DO DOCKER

O Docker foi configurado na Máquina Ubuntu e para a sua instalação, foram utilizados os seguintes comandos:

Inicialmente, foi atualizado a lista de Pacotes com o comando:

```
sudo apt update
```

Após atualização, foi configurado para que os pacotes necessários para instalação, façam download:

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

Foi adicionada a chave GPG para o repositório do Docker:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Foi adicionado o repositório do Docker às fontes do APT:

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
```

Após configuração, foi atualizado o banco de dados de pacotes com o Docker:

```
sudo apt update
```

Após isso, foi instalado o Docker:

```
sudo apt install docker-ce
```

Após instalação, foi verificado que o Docker estava funcionando, através do seguinte comando:

```
sudo systemctl status docker
```

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

System information as of Mon Mar 16 17:44:53 UTC 2020

System load:  1.4               Processes:    105
Usage of /:   62.6% of 3.87GB   Users logged in: 0
Memory usage: 5%               IP address for enp0s3: 10.0.2.15
Swap usage:   0%               IP address for docker0: 172.17.0.1

14 packages can be updated.
0 updates are security updates.

root@docker:~# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-03-16 17:44:46 UTC; 2min 41s ago
     Docs: https://docs.docker.com
   Main PID: 1050 (dockerd)
    Tasks: 11
   CGroup: /system.slice/docker.service
           └─1050 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Mar 16 17:44:46 docker dockerd[1050]: time="2020-03-16T17:44:46.270711066Z" level=warning msg="Your
Mar 16 17:44:46 docker dockerd[1050]: time="2020-03-16T17:44:46.270749525Z" level=warning msg="Your
Mar 16 17:44:46 docker dockerd[1050]: time="2020-03-16T17:44:46.270758316Z" level=warning msg="Your
Mar 16 17:44:46 docker dockerd[1050]: time="2020-03-16T17:44:46.271137115Z" level=info msg="Loading
Mar 16 17:44:46 docker dockerd[1050]: time="2020-03-16T17:44:46.697973221Z" level=info msg="Default
Mar 16 17:44:46 docker dockerd[1050]: time="2020-03-16T17:44:46.837426123Z" level=info msg="Loading
Mar 16 17:44:46 docker dockerd[1050]: time="2020-03-16T17:44:46.941505376Z" level=info msg="Docker d
Mar 16 17:44:46 docker dockerd[1050]: time="2020-03-16T17:44:46.942528512Z" level=info msg="Daemon h
Mar 16 17:44:46 docker systemd[1]: Started Docker Application Container Engine.
Mar 16 17:44:46 docker dockerd[1050]: time="2020-03-16T17:44:46.980810058Z" level=info msg="API list
lines 1-19/19 (END)
```

Foi verificada a versão do Docker

```
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
 127.0.0.0/8
Live Restore Enabled: false

WARNING: No swap limit support
root@docker:~# docker version
Client: Docker Engine - Community
 Version:           19.03.8
 API version:       1.40
 Go version:        go1.12.17
 Git commit:        afacb8b7f0
 Built:             Wed Mar 11 01:25:46 2020
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.8
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.12.17
  Git commit:       afacb8b7f0
  Built:            Wed Mar 11 01:24:19 2020
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.2.13
  GitCommit:        7ad184331fa3e55e52b890ea95e65ba581ae3429
 runc:
  Version:          1.0.0-rc10
  GitCommit:        dc9208a3303feef5b3839f4323d9beb36df0a9dd
 docker-init:
  Version:          0.18.0
  GitCommit:        fec3683
root@docker:~#
```

## 6. O PROJETO

Para mostrar como funciona o Ansible, criaremos um blog na plataforma Ghost e esse será executado em um container Docker utilizando a imagem Ghost.

### 6.1. VISÃO GERAL DA ESTRUTURA DE PASTAS

```
root@docker:/etc/ansible# tree .
.
├── ansible.cfg
├── blog-server.yml
├── hosts
│   └── blog-server
├── hosts.old
└── roles
    ├── config-basic
    │   ├── defaults
    │   │   └── main.yml
    │   ├── files
    │   ├── handlers
    │   │   └── main.yml
    │   ├── meta
    │   │   └── main.yml
    │   ├── README.md
    │   ├── tasks
    │   │   └── main.yml
    │   ├── templates
    │   ├── tests
    │   │   ├── inventory
    │   │   └── test.yml
    │   └── vars
    │       └── main.yml
    ├── docker
    │   ├── defaults
    │   │   └── main.yml
    │   ├── files
    │   │   └── blog
    │   │       └── docker-compose.yml
    │   ├── handlers
    │   │   └── main.yml
    │   ├── meta
    │   │   └── main.yml
    │   ├── README.md
    │   ├── tasks
    │   │   ├── add-docker-compose.yml
    │   │   ├── docker-compose.yml
    │   │   ├── docker.yml
    │   │   └── main.yml
    │   ├── templates
    │   ├── tests
    │   │   ├── inventory
    │   │   └── test.yml
    │   └── vars
    │       └── main.yml
    └── start-services
        ├── defaults
        │   └── main.yml
        ├── files
        ├── handlers
        │   └── main.yml
        ├── meta
        │   └── main.yml
        ├── README.md
        ├── tasks
        │   └── main.yml
        ├── templates
        ├── tests
        │   ├── inventory
        │   └── test.yml
        └── vars
            └── main.yml

30 directories, 32 files
```

## 6.2. INVENTORY

Na pasta `/etc/ansible/hosts` definimos o inventário das nossas máquinas que serão executados os playbooks (já veremos o que isso significa), no trecho abaixo, temos um inventário chamado de `blog-server`, que contém apenas uma máquina inventariada.

```
[blog-erver]
```

```
server-1 ansible_ssh_host=10.2.0.134 ansible_ssh_user=root
```

## 6.3. ANSIBLE.CFG

No arquivo `/etc/ansible/ansible.cfg` definimos qual será o inventário padrão e ignoramos a pergunta relacionada ao desejo de adicionar o IP da máquina ao arquivo `known_hosts`.

```
[defaults]
```

```
inventory=hosts/blog-server
```

```
record_host_keys=false
```

```
host_key_checking=false
```

## 6.4. PLAYBOOKS:

Podemos dizer que um **playbook** é uma receita, onde é definido o que será provisionado nos nossos inventários. Fazendo uma analogia, seria um passo a passo para uma receita de bolo.

```
- sudo: yes
```

```
roles:
```

```
- config-basic
```

```
- Docker
```

```
- start-services
```

Nesse exemplo temos 3 passos a serem executadas, `config-basic`, `docker`, `start-services`. É importante dizer que esses passos são executados sequencialmente. Abaixo veremos o que cada um deles faz.

## 6.5. ROLES

Na pasta **roles** temos o seguinte formato:

*Roles*

*config-basic*

*files*

*tasks*

*docker*

*files*

*tasks*

*start-services*

*files*

*tasks*

Dentro de cada role, existe a pasta **files**, onde ficam os arquivos que serão utilizados para configurar algo no provisionamento do ambiente. Dentro da pasta **tasks** ficam as tarefas que serão executadas para provisionar o ambiente desejado. Por padrão o Ansible sempre procura dentro da pasta tasks um arquivo **main.yml**.

## 6.6. CONFIG-BASIC:

Abaixo temos o nosso arquivo main.yml com tarefas para configurar: timezone, language, criação de um usuário, adicionar o usuário no grupo de sudo e instala o unzip na máquina provisionada.

Arquivo: /etc/ansible/roles/config-basic/tasks/main.yml

- name: Set the timezone

```
shell: echo "America/Manaus" | tee /etc/timezone && dpkg-reconfigure --frontend noninteractive tzdata
```

name: Set locale

```
shell: locale-gen pt_BR.UTF-8
```

- name: Define locale

```
shell: echo 'LC_ALL="pt_BR.UTF-8"' > /etc/default/locale
```

#Password 123456

- name: Add user ghost

```
user:                                name=ghost                                group=root
password=$6$rounds=656000$YikcDgDgNFSqpWaa$htE7Z0WFUQkH31FApl/NbEE2rj.WmjQc.
mM1LsVxyrBfVkOWMwupDGGwUce1wOog/bHSrLTBQ0TZNFhn40An1
```

- name: Add ghost user to sudoers

lineinfile:

dest=/etc/sudoers

regexp="ghost} ALL"

line="ghost ALL=(ALL) ALL"

state=presente

- name: Install latest unzip

```
apt: name=unzip state=latest update_cache=yes
```



## 6.7. DOCKER

Dentro da pasta files temos um arquivo do docker-compose.yml para subir o blog que desejamos na máquina provisionada.

Arquivo: /etc/ansible/roles/docker/tasks/docker-compose.yml

blog:

**hostname: blog**

**image: ghost**

**container\_name: ghost-blog**

**ports:**

**- 80:2368**

volumes:

**- ../var/lib/ghost**

Dentro da pasta tasks temos uma tarefa para adicionar o arquivo docker-compose.yml para dentro da home do usuário.

Arquivo: /etc/ansible/roles/docker/tasks/add-docker-compose.yml

*- name: Add docker files blog*

*copy: src=roles/docker/files/{{ item }} dest=/home/ghost/*

*with\_items:*

*- blog*

Agora temos a tarefa para instalar o Docker na máquina que irá ser provisionada.

Arquivo: /etc/ansible/roles/docker/docker.yml

*- name: Install Docker*

*shell: wget -qO- https://get.docker.com/ | sh*

*- name: Add ghost user group Docker*

*shell: usermod -aG docker ghost*

Próxima tarefa é para instalar o docker-compose na máquina provisionada.

Arquivo: `/etc/ansible/roles/docker/tasks/docker-compose.yml`

*- name: Install pip*

*apt: name=python-pip pkg= update\_cache=yes*

*- name: Install docker-compose*

*shell: pip install -U docker-compose*

Agora o arquivo main.yml tem apenas os imports das tarefas que estão na mesma role, dessa forma, fica mais organizado do que adicionar todas as tarefas no mesmo arquivo.

Arquivo: `/etc/ansible/roles/docker/main.yml`

*- include: docker.yml*

*- include: docker-compose.yml*

*- include: add-docker-compose.yml*

Start-services:

E por último, vamos executar o comando para "startar" o serviço com o docker-compose.

Arquivo: `/etc/ansible/roles/start-services/tasks/main.yml`

*- name: Start services ghost*

blog

*shell: docker-compose -f /home/ghost/blog/docker-compose.yml up -d*

Para provisionar uma máquina e rodar o blog com o Ghost, basta executar o comando:

```
Arquivo Editar Ver Pesquisar Terminal Ajuda
TASK [config-basic : Define locale] *****
changed: [server-1]

TASK [config-basic : Add user blog] *****
ok: [server-1]

TASK [config-basic : Add blog user to sudoers] *****
changed: [server-1]

TASK [config-basic : Install latest unzip] *****
ok: [server-1]

TASK [docker : Install docker] *****
changed: [server-1]
[WARNING]: Consider using get_url or uri module rather than running wget

TASK [docker : Add blog user group docker] *****
changed: [server-1]

TASK [docker : Install pip] *****
ok: [server-1]

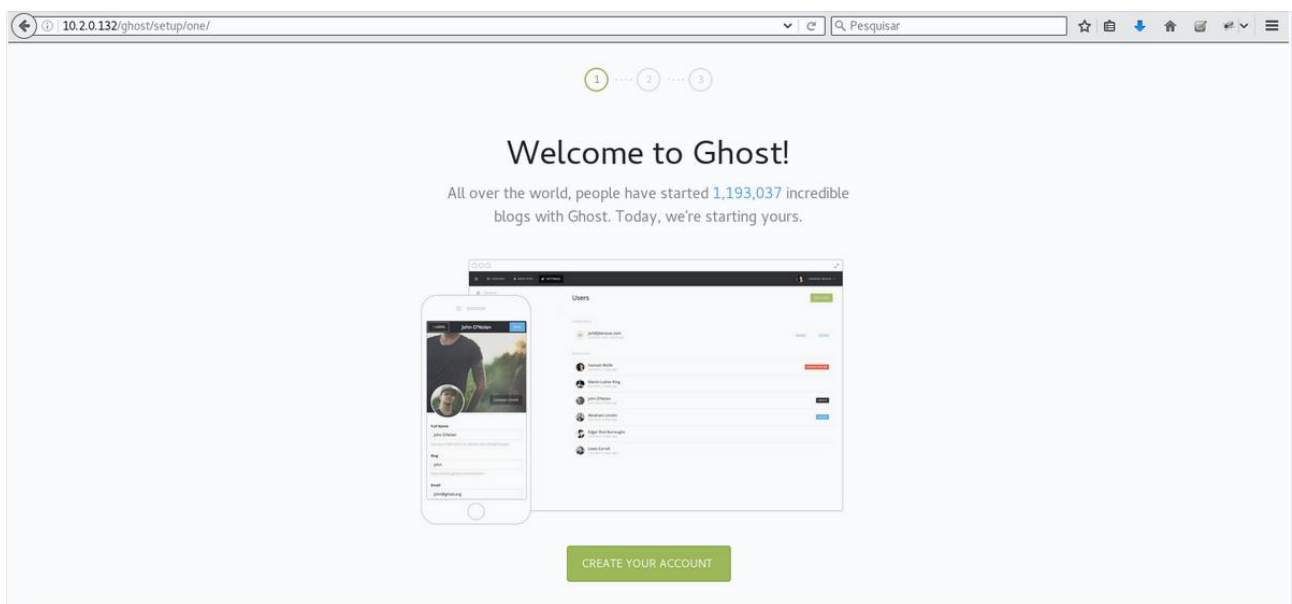
TASK [docker : Install docker-compose] *****
changed: [server-1]

TASK [docker : Add docker files portal-selo] *****
ok: [server-1] => (item=blog)

TASK [start-services : Start services ghost blog] *****
changed: [server-1]

PLAY RECAP *****
server-1 : ok=12  changed=8  unreachable=0  failed=0
```

Ao acessar a máquina que provisionamos, podemos ver o nosso blog em execução, pronto para configurarmos.



## **7. CONSIDERAÇÕES FINAIS**

A TP foi realizada dentro da minha experiência, apresentando ideias e conceitos aprendidos ao longo da minha carreira profissional.

A realização do projeto contou com o apoio de alguns profissionais que trabalham com Linux, exaustivas pesquisas na internet sobre o sistema operativo, além do material disponibilizado pelo professor.

Algumas informações, como a configuração do Ansible passo a passo, foram omitidas para não tornar a TP grande, visto que esse tema foi abordado e descrito na TP anterior.

## **8. REFERENCIAS TÉCNICAS**

Para prepara o documento, foram usadas as seguintes fontes de pesquisa:

[https://docs.ansible.com/ansible/latest/user\\_guide/playbooks.html](https://docs.ansible.com/ansible/latest/user_guide/playbooks.html)

<https://www.digitalocean.com/community/tutorials/como-instalar-e-configurar-o-ansible-no-ubuntu-18-04-pt>

<https://www.youtube.com/watch?v=FTxBa7i8VMM>