



CURSO 525

INFRAESTRUTURA ÁGIL COM PRÁTICAS DEVOPS





Construindo a infraestrutura como código

Ansible Galaxy

Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

- ## Anotações



Uma role nada mais é do que um diretório contendo um conjunto de arquivos, variáveis e playbooks que deverão ser aplicados em seu ambiente.

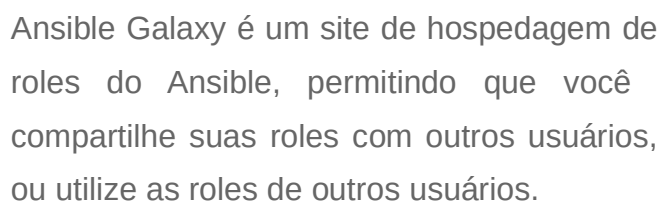
This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

default	Diretório, onde será informado os valores padrões das variáveis de sua role;
files	Diretório, onde são armazenados arquivos que deverão ser copiados para dentro do servidor;
handlers	Módulos que deverão ser executados, após um módulo ter sido executado – é feito em background.

[illegible]

tasks	Igual a playbook, deverá conter uma lista de módulos e a ordem em que cada um deverá executar e como executar;
templates	Template de arquivos que deverão ser construídos no servidor;
vars	Diretório, onde deve conter todas as variáveis internas de sua playbook.

[illegible]



No site existem roles das mais variadas atividades, desde instalação do Prometheus até de gerenciamento do Bind.

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Para trabalhar com roles, usamos o comando `ansible-galaxy` que permite criar, listar, baixar e remover roles, igual quando usamos o `apt` e o `yum`. Vamos tentar localizar a role “`geerlingguy.puppet`” para utilizarmos.

1 Na pasta `infraagil`, conecte-se a máquina `automation` para acessar o diretório de roles:

```
# vagrant ssh automation
# cd /etc/ansible/roles
```

2 Veja se consegue localizar a role `geerlingguy.puppet`:

```
# ansible-galaxy search puppet
```

3 Instale a role `geerlingguy.puppet`:

```
# sudo ansible-galaxy install geerlingguy.puppet
```

`automation.4labs.example`

Abaixo, segue alguns dos parâmetros que o `ansible-galaxy` pode receber:

search: Procura por playbooks, cujo o nome seja o informado;

list: Lista playbooks que estejam instaladas no servidor no momento;

install: Instala uma playbook no servidor;

install -f: Atualiza uma playbook para a ultima versão;

remove: Remove uma playbook;

init: Inicializa um diretório com a estrutura de uma playbook.

Gere o arquivo **compliance.yml** utilizando o modelo da pasta Aula 6.1:

1

2

automation.4labs.example

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Para criar uma role, usamos o parâmetro `init` do comando `ansible-galaxy`. Com ele, conseguimos criar a estrutura de uma role no diretório que estivermos. Vamos criar uma role de teste.

1

Crie a role teste:

```
# sudo ansible-galaxy init teste
```

2

Verifique a estrutura de diretórios da role:

```
# tree /etc/ansible/roles/teste
```

automation.4labs.example

Anotações

```
/etc/ansible/roles/teste
```

```
|-- README.md
```

```
|-- defaults
```

```
|  |-- main.yml
```

```
|-- files
```

```
|-- handlers
```

```
|  |-- main.yml
```

```
|-- meta
```

```
|  |-- main.yml
```

```
|-- tasks
```

```
|  |-- main.yml
```

```
|-- templates
```

```
|-- tests
```

```
|  |-- inventory
```

```
|  |-- test.yml
```

```
`-- vars
```

```
    |-- main.yml
```

- ## Anotações

[illegible]



Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

- ## Anotações



Vamos separar nossa instalação em blocos, devido aos diferentes sistemas operacionais utilizados: CentOS, Debian e Ubuntu.

[illegible]

- automation.4labs.example

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Criando a Role Puppet Agent

É importante saber que devemos criar algumas estruturas condicionais, utilizando o **when** para cada bloco, conforme o exemplo a seguir.

```
- name: Download e instalação do puppet Agent - RedHat
  block:
    - yum:
      name: 'https://yum.puppet.com/puppet6-release-el-7.noarch.rpm'
      state: present
    - yum:
      name: puppet-agent
      state: present
  when: ansible_distribution == 'CentOS'
```

automation.4labs.example

O parâmetro **when** está alinhado com o bloco, isso significa que esta condicional serve para qualquer task dentro do bloco declarado.

Para verificar o `ansible_distribution` de cada máquina, podemos executar o comando:

```
ansible all -m setup -a "filter=ansible_distribution"
```


Bloco: **Debian**

```
- get_url:
    url: https://apt.puppetlabs.com/puppet-release-buster.deb
    dest: /tmp/puppet6.deb
- apt:
    deb: /tmp/puppet6.deb
- apt:
    name: puppet-agent
    state: present
    update_cache: yes
```

Anotações

Bloco: **Ubuntu**

```
- get_url:
    url: https://apt.puppetlabs.com/puppet6-release-bionic.deb
    dest: /tmp/puppet6.deb
- apt:
    deb: /tmp/puppet6.deb
- apt:
    name: puppet-agent
    state: present
    update_cache: yes
```

Anotações

Bloco: **RedHat**

- ```
- yum:
 name: 'https://yum.puppet.com/puppet6-release-el-7.noarch.rpm'
 state: present
- yum:
 name: puppet-agent
 state: present
```

automation.4labs.example

## Anotações

[illegible]

1

```
lineinfile:
 path: '/etc/puppetlabs/puppet/puppet.conf'
 line: "{{ item }}"
loop:
 - '[main]'
 - 'runinterval = 1h'
 - '[agent]'
 - 'server = compliance.4labs.example'
 - 'certname = {{ ansible_nodename }}'
```

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

automation.4labs.example

```
- name: Executando o Puppet Agent nos servidores
 systemd:
 name: puppet
 state: started
 enabled: yes
```

automation.4labs.example

## Anotações

Primeiramente, vamos criar a estrutura da nossa role utilizando o Ansible Galaxy, e então, prosseguir com a criação da mesma.

```
1 # cd infraagil
vagrant ssh automation
```

2 Acesse o diretório `/etc/ansible/roles`:

```
cd /etc/ansible/roles
```

```
3 Crie a role puppet-agent:
sudo ansible-galaxy init puppet-agent
```

automation.4labs.example

## Anotações

[illegible]

```
Gere o arquivo main.yml utilizando o modelo da pasta Aula 6.2:
sudo cp /vagrant/4525/arquivos/Aula\ 6.2/main.yml
puppet-agent/tasks/
cat /etc/ansible/roles/puppet-agent/tasks/main.yml
```

automation.4labs.example

[illegible]

=== /etc/ansible/roles/puppet-agent/tasks/main.yml ===

---

# tasks file for puppet-agent

- name: Download e instalacao do Puppet Agent - Ubuntu

block:

- get\_url:

url: https://apt.puppetlabs.com/puppet6-release-bionic.deb

dest: /tmp/puppet6.deb

- apt:

deb: /tmp/puppet6.deb

- apt:

name: puppet-agent

state: present

update\_cache: yes

when: ansible\_distribution == 'Ubuntu'

- name: Download e instalacao do Puppet Agent - Debian

block:

- get\_url:

url: https://apt.puppetlabs.com/puppet-release-buster.deb

dest: /tmp/puppet6.deb

- apt:

deb: /tmp/puppet6.deb

- apt:

name: puppet-agent

state: present

update\_cache: yes

when: ansible\_distribution == 'Debian'

- name: Download e instalacao do puppet Agent - RedHat

block:

- yum:

name: 'https://yum.puppet.com/puppet6-release-el-7.noarch.rpm'

state: present

- yum:

name: puppet-agent

state: present

when: ansible\_distribution == 'CentOS'

- name: Adicionando o binario do Puppet ao PATH

block:

- file:

path: /root/.bashrc

state: touch

- lineinfile:

path: /root/.bashrc

line: 'PATH=/sbin:/bin:/usr/sbin:/usr/bin:/opt/puppetlabs/bin:/opt/puppetlabs/bin'

- name: Adicionando configurações do puppet

lineinfile:

path: '/etc/puppetlabs/puppet/puppet.conf'

line: "{{ item }}"

loop:

- '[main]'

- 'server = compliance.4labs.example'

- 'runinterval = 1h'

- name: Executando o Puppet Agent nos servidores

systemd:

name: puppet

state: restarted

enabled: yes



Agora, precisamos criar nossa playbook para executar a role puppet-agent.

```
Gere o arquivo puppet-agent.yml utilizando o modelo da pasta Aula 6.2:
sudo cp /vagrant/4525/arquivos/Aula\ 6.2/puppet-agent.yml
/etc/ansible/playbooks/
cat /etc/ansible/playbooks/puppet-agent.yml
- name: Instalando e configurando o Puppet Agent
 hosts: all
 roles:
 - puppet-agent
```

Execute a playbook:

```
sudo ansible-playbook /etc/ansible/playbooks/puppet-agent.yml
```

automation.4labs.example

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

- ## Anotações

[illegible]



## Construindo a Role nginx

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

- ## Anotações



O NGINX é um servidor web conhecido por sua alta performance, estabilidade, funcionalidades, simples configuração e baixo consumo de recurso.

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

sites-available

Diretório onde ficam localizados todos arquivos de configuração dos Virtual Hosts, disponíveis para serem habilitados.

sites-enabled

Diretório onde ficam os Virtual Hosts habilitados, normalmente são utilizados links simbólicos para arquivos localizados no diretório sites-available.

automation.4labs.example

## Anotações

[illegible]

```
server {
 listen 80;
 root /var/www/html;
 index index.html index.htm
 location / {
 try_files $uri $uri/ =404;
 }
}
```

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins or other markings on the paper.

Primeiramente, vamos criar a estrutura da nossa role utilizando o Ansible Galaxy, e então, prosseguir com a criação da mesma.

1 Acesse a pasta infraagil e conecte-se a máquina automation, utilizando o vagrant ssh:  
# cd infraagil  
# vagrant ssh automation

2 Acesse o diretório /etc/ansible/roles:  
# cd /etc/ansible/roles

```
3 Crie a role nginx:
sudo ansible-galaxy init nginx
```

automation.4labs.example

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



Crie o arquivo index.html:

1

```
sudo vim nginx/files/index.html
```

# Bem vindo a 4labs

2

Gere o arquivo **main.yml** utilizando o modelo da pasta Aula 6.3:

```
sudo cp /vagrant/4525/arquivos/Aula\ 6.3/main.yml nginx/tasks/
```

```
cat nginx/tasks/main.yml
```

automation.4labs.example

## Anotações

```
=== /etc/ansible/roles/nginx/tasks/main.yml ===
```

```

```

```
tasks file for nginx
```

```
- name: Instalando o NGINX
```

```
 package:
```

```
 name: nginx
```

```
 state: present
```

```
- name: Removendo website padrao
```

```
 file:
```

```
 path: "{{ item }}"
```

```
 state: absent
```

```
 loop:
```

```
 - /var/www/html/index.nginx-debian.html
```

```
 - /var/www/html/index.html
```

```
 - /var/www/html/index.html
```

```
- name: Copiando website 4labs
```

```
 copy:
```

```
 src: index.html
```

```
 dest: /var/www/html/index.html
```

```
- name: Habilitando e reiniciando o serviço do NGINX
```

```
 service:
```

```
 name: nginx
```

```
 state: restarted
```

```
 enabled: yes
```

Agora, vamos criar nossa playbook.

```
Gere o arquivo nginx.yml utilizando o modelo da pasta Aula 6.3:
sudo cp /vagrant/4525/arquivos/Aula\ 6.3/nginx.yml
/etc/ansible/playbooks/
cat /etc/ansible/playbooks/nginx.yml
- name: Aplicando as roles de log
 hosts: log
 roles:
 - nginx
```

Execute a playbook:

```
sudo ansible-playbook /etc/ansible/playbooks/nginx.yml
```

automation.4labs.example

## Anotações

automation.4labs.example



This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

- ## Anotações



# Construindo a infraestrutura como código

## Construindo a Role iSCSI

[illegible]

- 1 O que é o iSCSI
- 2 Estrutura
- 3 Criando a Role

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

## O que é iSCSI



**iSCSI** (Internet SCSI) é um protocolo de transporte para comandos SCSI (Small Computer System Interface) entre um computador anfitrião (initiator) e um dispositivo de destino (target). É tipicamente utilizado no contexto de SAN (Storage Area Network) e funciona sobre uma rede IP.



O iSCSI se difere dos outros modelos de compartilhamento de arquivos como CIFS e NFS, principalmente por entregar um disco diretamente sem particionamento.

O acesso aos dados de Storage iSCSI é um block level (nível de bloco), isso significa que o iSCSI Target serve um iSCSI Initiator (host). Tal característica é a mesma quando o S.O. lê o protocolo SCSI, o sistema operacional interpreta o protocolo e o identifica como um novo disco. Esse armazenamento compartilhado em blocos, vem sendo muito utilizado em pequenas empresas, pois sua configuração é mais fácil e os custos de implementação e manutenção são mais baratos, em comparação a um Storage SAN (Storage Area Network).

Vantagens do Storage iSCSI:

- ✓ \* Solução para armazenamento compartilhado para bancos de dados como SQL e Oracle, alteração ou instalação de sistemas de compartilhamento de arquivos, baseados em servidores;
- ✓ \* Otimização do espaço em disco: o usuário pode aumentar a capacidade de armazenamento, ou então, diminuí-lo conforme julgar necessário;
- ✓ \* Dispositivo de armazenamento semelhante ao sistema ZFS, dispositivo de arquivamento para o sistema operacional Linux;
- ✓ \* Backup com espelhamento, função que consiste na gravação dos discos rígidos;
- ✓ \* Função autorreparação que identifica e elimina falhas, sem comprometer a produtividade;
- ✓ \* Iniciadores e alvos são identificados pelo Protocolo CHAP (Challenge Handshake Authentication Protocol), autenticação que verifica a identidade do operador do sistema.



- automation.4labs.example

[illegible]

Após essa configuração, vamos também formatar partições para que o disco seja utilizável.

automation.4labs.example

## Anotações

[illegible]

Acesse a pasta infraagil e conecte-se a máquina automation, utilizando o vagrant ssh:

```
2 # cd /etc/ansible/roles
```

automation.4labs.example

## Anotações

Gere o arquivo **main.yml** utilizando o modelo da pasta Aula 6.4:

1

automation.4labs.example

## Anotações

[illegible]

```
=== /etc/ansible/roles/puppet-agent/tasks/main.yml ===

tasks file for iscsi
- name: Instalando o tgt e configurando o iscsi
 block:
 - package:
 name: scsi-target-utils
 state: present
 - blockinfile:
 create: yes
 path: /etc/tgt/conf.d/4labs_scsi.conf
 block: |
 <target iqn.2019-05.automation.4labs.example:lun>
 backing-store /dev/sdb
 </target>
 - service:
 name: tgtd
 state: restarted
 enabled: yes
 when: ansible_hostname == "automation"

- name: Instalando e configurando o openiscsi
 block:
 - package:
 name: ["open-iscsi", "parted", "xfsprogs", "btrfs-progs"]
 state: present
 - open_iscsi:
 show_nodes: yes
 discover: yes
 portal: 10.5.25.10
 register: iscsi
 - open_iscsi:
 login: yes
 target: "{{ iscsi.nodes[0] }}"
 - replace:
 path: "/etc/iscsi/nodes/{{ iscsi.nodes[0] }}/10.5.25.10,3260,1/default"
 regexp: "node.startup = manual"
 replace: "node.startup = automatic"
 - service:
 name: open-iscsi
 state: restarted
 enabled: yes
 when: ansible_hostname == "log"

(...)
```

```
=== /etc/ansible/roles/puppet-agent/tasks/main.yml ===
```

```
(...)
```

```
- name: Criando partições no Disco
```

```
 block:
```

```
 - parted:
```

```
 device: /dev/sdb
```

```
 number: 1
```

```
 state: present
```

```
 part_end: 50%
```

```
 - parted:
```

```
 device: /dev/sdb
```

```
 number: 2
```

```
 state: present
```

```
 part_start: 50%
```

```
 - filesystem:
```

```
 fstype: ext4
```

```
 dev: /dev/sdb1
```

```
 - filesystem:
```

```
 fstype: ext3
```

```
 dev: /dev/sdb2
```

```
 when: ansible_hostname == 'log'
```

```
Gere o arquivo iscsi.yml utilizando o modelo da pasta Aula 6.4:
sudo cp /vagrant/4525/arquivos/Aula\ 6.4/iscsi.yml
/etc/ansible/playbooks/
cat /etc/ansible/playbooks/iscsi.yml
- name: Instalando o iSCSI
 hosts: automation log
 roles:
 - iscsi
```

automation.4labs.example

## Anotações

automation.4labs.example

- 1 Verifique se o target foi mapeado na VM Log:  
`# sudo ansible log -m shell -a "iscsiadm -m session"`
  - 2 Verifique se o bloco sdc1 e sdc2 foram criados na VM Log:  
`# sudo ansible log -m shell -a "lsblk"`

## Anotações



- ## Anotações



## This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

- ## Anotações

O Rundeck é um software de código aberto para automatização de processos e rotinas de operação ampla, utilizado em ambientes DevOps de Entrega Contínua.

[illegible]

## framework.properties

## rundeck-config.properties

automation.4labs.example

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

## framework.properties

```
framework.server.hostname = automation.4labs.example
framework.server.name = 4labs
framework.server.port = 4440
framework.server.url = http://automation.4labs.example:4440
```

automation.4labs.example

## Anotações

[illegible]



Primeiramente, vamos criar a estrutura da nossa role utilizando o Ansible Galaxy, e então, prosseguir com a criação da mesma.

1 Acesse a pasta infraagil e conecte-se a máquina automation, utilizando o vagrant ssh:  
# cd infraagil  
# vagrant ssh automation

2 Acesse o diretório `/etc/ansible/roles`:

```
cd /etc/ansible/roles
```

3 Crie a role rundeck:

```
sudo ansible-galaxy init rundeck
```

automation.4labs.example

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



1

automation.4labs.example

## This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

```
=== /etc/ansible/roles/rundeck/tasks/main.yml ===
```

```

```

```
tasks file for rundeck
```

```
- name: Instalacao das Dependencias e Repositorio Rundeck
```

```
 yum:
```

```
 name: "{{ item }}"
```

```
 state: present
```

```
 loop:
```

```
 - vim
```

```
 - java-1.8.0
```

```
 - https://repo.rundeck.org/latest.rpm
```

```
- name: Instalacao do Rundeck
```

```
 yum:
```

```
 name: rundeck
```

```
 state: present
```

```
- name: Configurando Rundeck
```

```
 lineinfile:
```

```
 path: "{{ item.path }}"
```

```
 regexp: "{{ item.regexp }}"
```

```
 line: "{{ item.line }}"
```

```
 loop:
```

```
 - {path: '/etc/rundeck/framework.properties', regexp: '^framework.server.name', line: 'framework.server.name = 4labs'}
```

```
 - {path: '/etc/rundeck/framework.properties', regexp: '^framework.server.hostname', line: 'framework.server.hostname = automation.4labs.example'}
```

```
 - {path: '/etc/rundeck/framework.properties', regexp: '^framework.server.port', line: 'framework.server.port = 4440'}
```

```
 - {path: '/etc/rundeck/framework.properties', regexp: '^framework.server.url', line: 'framework.server.url = http://automation.4labs.example:4440'}
```

```
 - {path: '/etc/rundeck/rundeck-config.properties', regexp: '^grails.server', line: 'grails.serverURL=http://automation.4labs.example:4440'}
```

```
- name: Iniciando e Habilitando o serviço do rundeck
```

```
 systemd:
```

```
 name: rundeckd
```

```
 state: started
```

```
 enabled: yes
```

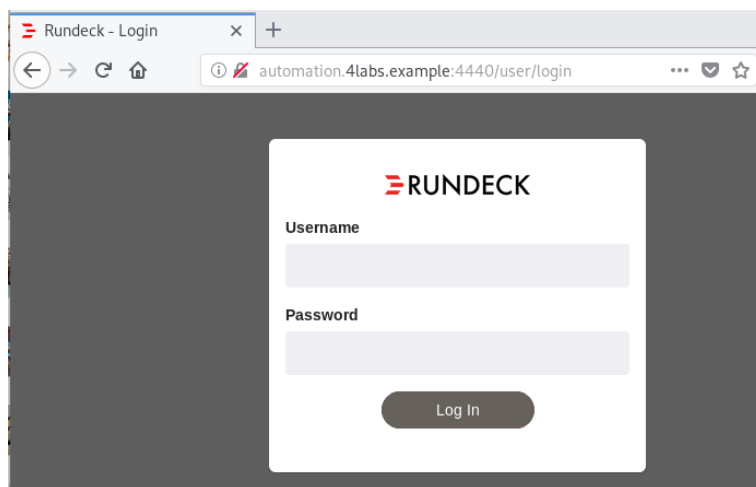
Agora, vamos criar nossa playbook.

- rundeck

automation.4labs.example

## This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Acesse pelo navegador o endereço `http://automation.4labs.example:4440`.



automation.4labs.example

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

- ## Anotações

[illegible]

- 1 O que é o Docker
- 2 Estrutura
- 3 Criando a Role Docker

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



- ✓ **docker-ce** → Docker Community Edition, é o daemon do Docker;

- ✓ **docker-ce-cli** → Docker Command Line Interface, é a interface de terminal para utilizar o docker;

- ✓ **containerd.io** → É um runtime de contêineres com ênfase em simplicidade, robustez e portabilidade;

✓ **Docker Machine Bash Completion** → São os recursos de auto completar do docker.

automation.4labs.example

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Primeiramente, vamos criar a estrutura da nossa role utilizando o Ansible Galaxy, e então, prosseguir com a criação da mesma.

1 Acesse a pasta infraagil e conecte-se a máquina automation, utilizando o vagrant ssh:

```
cd infraagil
vagrant ssh automation
```

2 Acesse o diretório `/etc/ansible/roles`:

```
cd /etc/ansible/roles
```

```
3 Crie a role docker:
sudo ansible-galaxy init docker
```

automation.4labs.example

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

```
Gere o arquivo main.yml utilizando o modelo da pasta Aula 6.6:
sudo cp /vagrant/4525/arquivos/Aula\ 6.6/main.yml
docker/tasks/
cat docker/tasks/main.yml

tasks file for docker
- name: Instalacao das Dependencias
 yum:
 name: "{{ item }}"
 state: present
(...)
```

automation.4labs.example

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

```
=== /etc/ansible/roles/docker/tasks/main.yml ===
```

```

```

```
tasks file for docker
```

```
- name: Instalacao das Dependencias
```

```
 yum:
```

```
 name: "{{ item }}"
```

```
 state: present
```

```
 loop:
```

```
 - epel-release
```

```
 - yum-utils
```

```
 - device-mapper-persistent-data
```

```
 - curl
```

```
 - lvm2
```

```
 - vim
```

```
 - bash-completion
```

```
- name: Adicionar repositório do Docker-Engine
```

```
 get_url:
```

```
 url: https://download.docker.com/linux/centos/docker-ce.repo
```

```
 dest: /etc/yum.repos.d/docer-ce.repo
```

```
- name: Instalar o Docker-Engine
```

```
 yum:
```

```
 name: "{{ item }}"
```

```
 state: present
```

```
 loop:
```

```
 - docker-ce
```

```
 - docker-ce-cli
```

```
 - containerd.io
```

```
- name: Instalar o Docker Machine bash completion
```

```
 get_url:
```

```
 url: https://raw.githubusercontent.com/docker/machine/v0.16.0/contrib/completion/bash/docker-machine.bash
```

```
 dest: /etc/bash_completion.d/docker-machine
```

```
- name: Habilitando e iniciando o serviço do Docker-ce
```

```
 systemd:
```

```
 name: docker
```

```
 state: started
```

```
 enabled: yes
```

```
- name: Instalando o docker-compose
```

```
 get_url:
```

```
 url: https://github.com/docker/compose/releases/download/1.24.0/docker-compose-Linux-x86_64
```

```
 dest: /usr/local/bin/docker-compose
```

```
 mode: 0755
```

Gere o arquivo **docker.yml** utilizando o modelo da pasta Aula 6.6:

```
sudo cp /vagrant/4525/arquivos/Aula\ 6.6/docker.yml
/etc/ansible/playbooks/
cat /etc/ansible/playbooks/docker.yml
- name: Instalação do Docker
 hosts: container
 roles:
 - docker
```

### Execute a playbook:

```
sudo ansible-playbook /etc/ansible/playbooks/docker.yml
```

## Anotações

1

```
sudo ansible container -m shell -a "docker --version"
sudo ansible container -m shell -a "docker system info"
```

automation.4labs.example

## Anotações

- 1 O que é o Docker
- 2 Estrutura
- 3 Criando a Role Docker

## Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.