



# CURSO 525

## INFRAESTRUTURA ÁGIL COM PRÁTICAS DEVOPS



## Características do Ansible

## 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

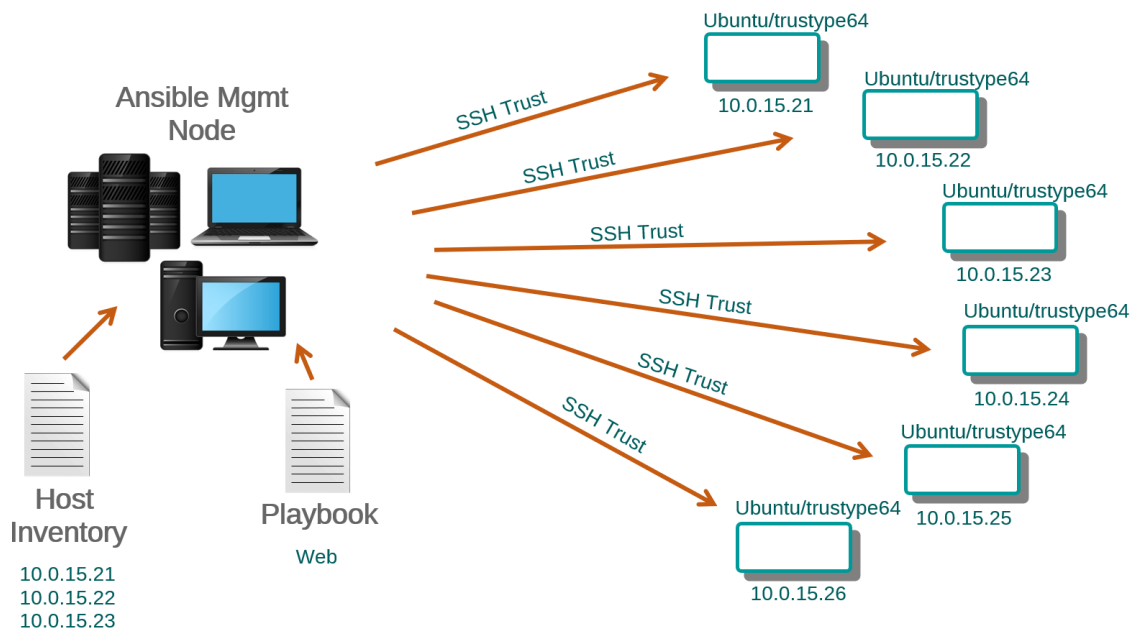


# ANSIBLE

## 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

Documento produzido pela 4Linux



O ansible possui uma arquitetura **MASTER TO NODES**, sendo necessário apenas que os nós possuam o Python instalado e uma conexão SSH com a máquina que deverá responder ao comando remoto, podendo ser realizada via usuário/senha ou até mesmo troca de chaves. Uma das vantagens de se utilizar Ansible em ambientes Linux, é que na maioria das distribuições o Python já vem pré-instalado, o que não faz necessário nenhuma outra configuração no nó de destino, se não o Servidor SSH.

automation.4labs.example

- ## Anotações

O arquivo de inventário do Ansible armazena uma lista de servidores gerenciados e acessados por ele. Por padrão, ele fica localizado em `/etc/ansible/hosts`.

Dentro do arquivo, podemos definir um host ou múltiplos, depois separá-los por grupos com o uso de “[ ]”.

1

Gere o arquivo hosts utilizando o modelo ansible-hosts da pasta Aula 5.1:

```
# cd /vagrant/4525/arquivos/Aula\ 5.1/
# sudo cp ansible-hosts /etc/ansible/hosts
# cat /etc/ansible/hosts
[automation]
automation.4labs.example
....
```

automation.4labs.example

No arquivo de inventário, podemos especificar uma máquina colocando o hostname ou endereço ip, ou especificar um grupo utilizando as chaves.

Também podemos especificar variáveis pelo inventário e passar essas variáveis de acordo com o node. No exemplo abaixo, estamos passando a variável `ansible_python_interpreter` para apontar onde está localizado o interpretador do python, utilizado pelo ansible em cada servidor.

[ubuntu]

srv01.4labs.example ansible\_python\_interpreter=/usr/bin/python

[rhel]

srv02.4labs.example ansible\_python\_interpreter=/usr/bin/python3

srv03.4labs.example ansible\_python\_interpreter=/usr/bin/python3



===Ansible Hosts ===

[automation]

automation.4labs.example ansible\_python\_interpreter=/usr/bin/python

[compliance]

compliance.4labs.example ansible\_python\_interpreter=/usr/bin/python3

[container]

container.4labs.example ansible\_python\_interpreter=/usr/bin/python

[scm]

scm.4labs.example ansible\_python\_interpreter=/usr/bin/python

[log]

log.4labs.example ansible\_python\_interpreter=/usr/bin/python3

No arquivo `ansible.cfg`, podemos definir os parâmetros de configuração do ansible.

1

Edite o arquivo de configuração do ansible:

```
# sudo vim /etc/ansible/ansible.cfg

roles_path = /etc/ansible/roles
host_key_checking = False
private_key_file = /root/.ssh/id_rsa
```

Podemos verificar agora a conexão, utilizando o módulo `ping` do ansible.

2

Execute o módulo `ping` do ansible para todo o inventário:

```
# sudo ansible -m ping all
```

automation.4labs.example

O arquivo `ansible.cfg` é responsável por toda a configuração do ansible, nele definimos qual será a nossa chave de acesso privada a ser utilizada nas conexões via SSH para os nós. Definimos também que não será feita a checagem do fingerprint.

O comando **ansible -m ping all** faz com que o programa (ansible) utilize o módulo (-m) `ping` nos hosts (all).

O módulo `ping` do ansible executa a conexão SSH e retorna, caso a conexão seja realizada com sucesso ou não.

- 1 O que é o Ansible
- 2 Características
- 3 Arquitetura
- 4 Instalação do Ansible

## 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.

- ## Anotações

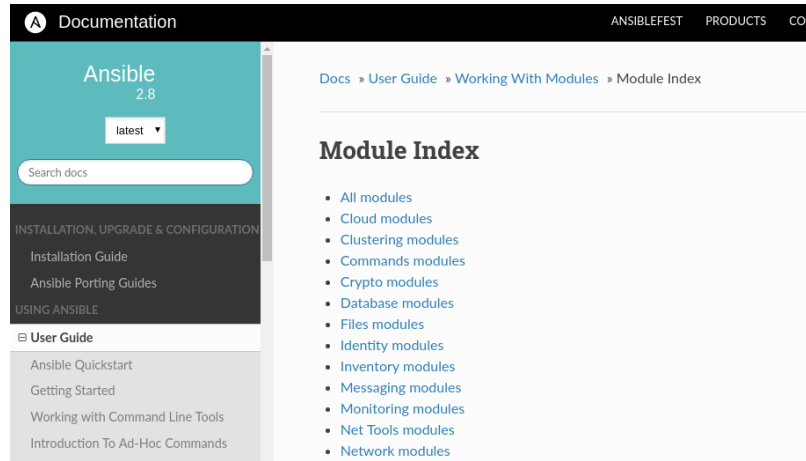
Módulos são utilizados para descrever as atividades que o Ansible executará na máquina de destino. Alguns desses módulos são:

Instalação	Arquivos	Execução	Download
✓ package;	✓ file;	✓ command;	✓ get_url;
✓ yum;	✓ unarchive;	✓ shell;	✓ git;
✓ apt;	✓ copy;	✓ raw;	✓ unarchive;
✓ pacman;	✓ template;		✓ fetch.

Ansible possui milhares de módulos para as mais variáveis atividades, desde gerenciar os pacotes que serão instalados em uma máquina virtual, até o gerenciamento de containers. Alguns dos módulos mais usados no Ansible são:

**apt:** Gerencia a instalação de pacotes, utilizando o APTITUDE (debian/ubuntu);  
**yum:** Gerencia a instalação de pacotes, utilizando o YUM (RedHat/CentOS/Fedora);  
**package:** Gerencia a instalação de pacotes, utilizando o APTITUDE/YUM;  
**command:** Executa um comando em node remoto;  
**shell:** Executa um shell script dentro da máquina, após realizar as transferências;  
**service:** Gerencia serviços em máquina remotas;  
**copy:** Copia arquivo na máquina local para o node remoto;  
**git:** Gerencia repositório do git;  
**unarchive:** Descompacta arquivos na máquina remota;  
**mount:** Gerencia os dispositivos montados na máquina;  
**template:** Gerencia templates no ansible.

A documentação de todos os módulos disponíveis do Ansible e seus parâmetros, podem ser visualizados em seu website <https://docs.ansible.com/>



A documentação do Ansible é completa, pesquisando por módulos temos uma lista de informações com todos seus parâmetros possíveis, bem como exemplos de playbooks (assunto que será visto posteriormente).

Podemos também utilizar o comando **ansible-doc** para visualizar a documentação.

1 **# ansible-doc --help**

2 Liste os módulos:  
**# ansible-doc -l**

3 Visualize parte da documentação de um módulo:  
**# ansible-doc -s <MODULO>**

4 Visualize a documentação completa de um módulo:  
**# ansible-doc <MODULO>**

automation.4labs.example

**ansible-doc** – Ferramenta de documentação de plugins do ansible.

**-l** → Lista os plugins disponíveis;

**-s** → Visualiza parte da documentação.



Para executar um comando no ansible, precisamos informar qual módulo será utilizado, quais são os parâmetros, se houver, e quais máquinas no inventário receberão o comando.

```
ansible -m <MODULO> -a <ARGUMENTOS> <HOSTS>  
ou  
ansible <HOSTS> -m <MODULO> -a <ARGUMENTOS>
```

Por padrão, o Ansible lê o arquivo de inventário localizado em `/etc/ansible/hosts`, porém é possível que seja informado outro arquivo de inventário, passando o parâmetro `-i` seguido do arquivo de inventário a ser utilizado.

Ex.:

```
ansible srv01 -m ping -i /files/inventario
```

Vamos realizar algumas tarefas rotineiras, utilizando o ansible.

```
1 # ansible --help
```

```
2 # sudo ansible compliance -m shell -a "cat /etc/hosts"
```

3 Transfira um arquivo para o servidor log:  
# sudo ansible log -m copy -a "src=/etc/hosts dest=/tmp"

automation.4labs.example

## Anotações

[illegible]

```
# sudo ansible compliance -m apt -a "name=cmatrix state=present
update_cache=yes"
```

```
# sudo ansible automation -m service -a "name=crond state=restarted
enabled=yes"
```

## Anotações

Através dele podemos coletar diversas informações importantes, tais como armazenamento, configurações de redes, configuração do sistema, versão do sistema operacional, dentre outras.

## Anotações

[illegible]

automation.4labs.example

- |   |   |
|---|---|
| 1 | Verifique os facts do servidor log:<br><code># sudo ansible log -m setup</code>                           |
| 2 | Colete os facts de todos os servidores:<br><code># sudo ansible all -m setup</code>                       |
| 3 | Filtre uma seleção de facts:<br><code># sudo ansible all -m setup -a "filter=ansible distribution"</code> |

## Anotações

- ## 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 **YAML (YAML Ain't Markup Language)** é um formato de serialização de dados legível por humanos para todas as linguagens de programação.

É amplamente utilizada em ambientes DevOps, devido a sua facilidade de interpretação.



YAML é uma linguagem simples de ler e escrever, o ansible utiliza o formato nos arquivos de configuração dos Playbooks e Roles.

Exemplo de arquivo em YAML:

```
---  
# Funcionário da Dexter  
  
- name: João Devops da Silva  
  job: Analista DevOps  
  skills:  
    - Administração de Sistema GNU/Linux  
    - Automação com Ansible  
    - Gerencia de Configuração com Puppet  
    - Testes de Infraestrutura com Inspec  
    - Versionamento com Git e Gogs
```

- ## Anotações

[illegible]

## Preparando o VIM

Podemos utilizar alguns recursos no VIM que nos ajudarão a escrever nossos arquivos no formato YAML, para isso vamos adicionar alguns parâmetros no arquivo **vimrc**:

1

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

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

2

Edite o arquivo vimrc e adicione o conteúdo no final do arquivo:

```
# sudo vim /etc/vimrc +$
" Recursos para o YAML
    set cursorline
    set cursorcolumn
    set number
```

automation.4labs.example

**vimrc** → Arquivo de parametrização do vim;

**cursorline** → Exibe uma linha, onde o cursor estiver posicionado;

**cursor column** → Exibe uma coluna, onde o cursor estiver posicionado;

**number** → Exibe o número das linhas.

## YAML Lint

```
1  --  
2  # Funcionário da Dexter  
3  
4  - name: João Devops da Silva  
5  job: Analista DevOps  
6  skills:  
7  - Administração de Sistema GNU/Linux  
8  - Automação com Ansible  
9  - Gerencia de Configuração com Puppet  
10 - Testes de Infraestrutura com Inspec  
11 - Versionamento com Git e Gogs  
12  
13
```

# YAML Validator

```
1  ---
2  # Funcionário da Dexter
3
4  - name: João Devops da Silva
5    job: Analista DevOps
6    skills:
7      - Administração de Sistema GNU/Linux
8      - Automação com Ansible
9      - Gerência de Configuração com Puppet
10     - Testes de Infraestrutura com Inspec
11     - Versionamento com Git e Gogs
```

<https://jsonformatter.org/yaml-validator>

## 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

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]

- ## Anotações

Em um nível mais básico, as Playbooks podem ser usadas para gerenciar as configurações e realizar deploys nas máquinas remotas.

## 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.



Altere para o root e crie a estrutura de diretórios para testar as playbooks do ansible:

2 Entre na pasta exemplos:

```
# cd /etc/ansible/exemplos
```

## Anotações

Gere o arquivo **resolv.conf** utilizando o modelo da pasta Aula 5.4:

```
# cat resolv.conf
```

automation.4labs.example

[illegible]

Gere o arquivo **primeiro.yml** utilizando o modelo da pasta Aula 5.4:

```
# cat primeiro.yml
```

## # Minha primeira Playbook

```
- name: Minha primeira playbook
```

```
hosts: all
```

```
tasks:
```

```
- name: Atualizar o resolv.conf
```

copy:

```
src: resolv.conf
```

```
dest: /etc/resolv.conf
```

automation.4labs.example

## Anotações

1

2

7

Ao criar uma variável, você está permitindo que o usuário gerencie qual será a informação daquele campo, permitindo melhor reusabilidade da playbook.

[illegible]

```
vars:
- server_name: servidor01
- username: analistadevops
- conf_file: /opt/app/app.conf
```

## 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

[illegible]

Agora, vamos executar nossa playbook com a variável.

## Execute a playbook:

```
# ansible-playbook users.yml
```

Utilizaremos o ansible ad-hoc para verificar a criação do usuário.

Utilize o módulo shell para verificar o usuário no passwd:

```
# ansible all -m shell -a "getent passwd sysadmin"
```

## Anotações

[illegible]



```
Execute a playbook:  
# ansible-playbook users.yml -e username=developer
```

Utilize o módulo shell para verificar o usuário no passwd:

```
# ansible all -m shell -a "getent passwd developer"
```

## 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.

```
vars_files:
- config.yml
```

## Anotações

Crie o arquivo:

```
# vim variaveis.yml
username: analista
```

```
# cp /vagrant/4525/arquivos/Aula\ 5.4/users2.yml users.yml
```

```
# cat users.yml
```

(...)

```
- name: Criacao de usuarios nos servidores
  hosts: all
  vars_files:
    - variaveis.yml
```

///

automation.4labs.example

## Anotações

[illegible]

```
Execute a playbook:
# ansible-playbook users.yml
```

Utilizaremos o ansible ad-hoc para verificar a criação do usuário.

Utilize o módulo shell para verificar o usuário no passwd:

```
# ansible all -m shell -a "getent passwd analista"
```

## Anotações

O Ansible permite que criemos laços de repetição, utilizando uma variável que receberá uma lista ou lista de **loop**, permitindo assim executar a mesma ação de acordo com a quantidade de itens que ela possui.

[illegible]

1

automation.4labs.example

[illegible]

1

2

## Anotações

Vamos criar uma segunda playbook, utilizando loops em variáveis:

1

Gere o arquivo **loop2.yml** utilizando o modelo da pasta Aula 5.4:

```
# cp /vagrant/4525/arquivos/Aula\ 5.4/loop2.yml .
# cat loop2.yml
- name: Instalando uma lista de pacotes nos servidores Ubuntu
  hosts: compliance log
  tasks:
    - name: Instalando pacotes
      apt:
        name: "{{ item }}"
        update_cache: yes
      loop:
        - sl
        - cmatrix
        - ncdu
```

automation.4labs.example

**loop:** Cria uma variável temporária chamada **item**, onde serão armazenados todos os valores que você passou para dentro do loop.

Vale lembrar que essa variável só existirá enquanto estiver executando aquele módulo. Após a conclusão, os valores serão perdidos. Caso necessite de uma lista reutilizável, use variáveis no lugar.

Em versões mais antigas do Ansible (anterior a 2.5 ), o parâmetro a ser utilizado era o **with\_items** ao invés do **loop**, porém o mesmo foi descontinuado.



```
Execute a playbook:
# ansible-playbook loop2.yml
```

```
Verifique se os pacotes foram instalados:
# ansible compliance -m shell -a "dpkg -l | egrep 'sl|cmatrix|ncdu'"
# ansible log -m shell -a "dpkg -l | egrep 'sl|cmatrix|ncdu'"
```

## Anotações

O Ansible utiliza o parâmetro **when** para efetuar esse teste.

## 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

```
# cp /vagrant/4525/arquivos/Aula\ 5.4/when.yml .
```

```
# cat when.yml
```

- name: Instalando uma lista de pacotes nos servidores Ubuntu

```
hosts: all
```

tasks:

```
- name: Instalando pacotes
```

apt:

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

Loop:

- cmatrix

- sl

- ncdu

```
when: ansible_distribution == "Ubuntu"
```

1

## Anotações

Agora, vamos executar nossa playbook condicional.

1

### Execute a playbook:

```
# ansible-playbook when.yml
```

Note que ao executar a `playbook`, é feita uma varredura no inventário e os `hosts` que não trazem a condicional tratada no **when** como verdadeira, não executam a `playbook` e apresentam como **skipping** (pulando) no terminal.

```
TASK [Instalando pacotes] *****
skipping: [scm.4labs.example] => (item=cmatrix)
skipping: [scm.4labs.example] => (item=sl)
skipping: [scm.4labs.example] => (item=ncdu)
skipping: [container.4labs.example] => (item=cmatrix)
skipping: [container.4labs.example] => (item=sl)
skipping: [container.4labs.example] => (item=ncdu)
skipping: [automation.4labs.example] => (item=cmatrix)
skipping: [automation.4labs.example] => (item=sl)
skipping: [automation.4labs.example] => (item=ncdu)
ok: [log.4labs.example] => (item=cmatrix)
ok: [compliance.4labs.example] => (item=cmatrix)
ok: [log.4labs.example] => (item=sl)
ok: [compliance.4labs.example] => (item=sl)
ok: [log.4labs.example] => (item=ncdu)
```

## 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.