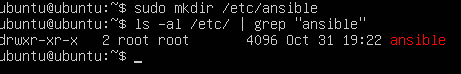
1. **PRIMEIRAS CONFIGURAÇÕES DO ANSIBLE**

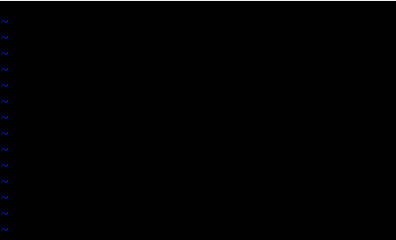
**1 - Agora vamos adicionar uma máquina CLIENT no ANSIBLE, para isso, precisamos criar um diretório chamado ansible, dentro da etc:**

execute: **mkdir /etc/ansible  
**

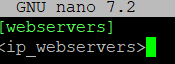
**OBS: Esse diretório é padrão do Ansible, é onde vamos configurar os nossos grupos de servidores, apontando IP, chave ssh.**

**Toda vez que o Ansible for fazer alguma instalação, ele vai acessar esse hosts ou o grupo de servidores para realizar a configuração.**

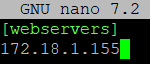
**2 - Precisamos de um arquivo chamado hosts para configurar os o IPs dos WebServes:**

execute: **nano /etc/ansible/hosts**

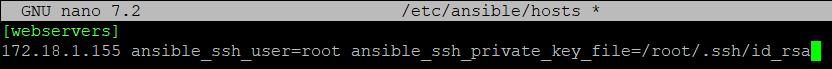
**3 - Dentro do arquivo hosts, configure da seguinte maneira:**

****

**OBS: Substitua *<ip\_do\_webserver>* pelo IP que você copiou anteriormente**

****

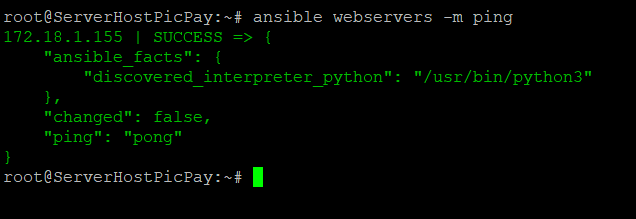
**Vamos configurar acesso SSH (RECOMENDADO):**

Na frente do IP que você copiou anteriormente, digite: **ansible\_ssh\_user=root ansible\_ssh\_private\_key\_file=/root/.ssh/id\_rsa**

**OBS: Após isso, salvar o arquivo**

**4 - Primeiro teste com comando AD-HOC, vamos validar se o Ansible consegue alcançar a nossa máquina:**

execute: **ansible webservers -m ping**



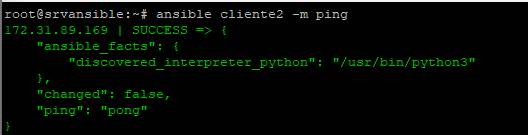
**OBS:** O parâmetro -m é o comando que será executado, neste caso ping

**5 - Cliente 2 com Senha   
  
Execute: nano /etc/ansible/hosts**

**Dentro do arquivo hosts, configure da seguinte maneira**

****

Na frente do IP que você copiou anteriormente, digite: **ansible\_ssh\_user=root ansible\_ssh\_pass=Senha  
**

**Execute: apt install sshpass -y   
  
Faça o teste de ping   
Execute: ansible cliente2 -m ping  
**

1. **TRABALHAR COM VÁRIOS HOSTS**
2. **CRIE UMA SEGUNDA MÁQUINA VIRTUAL (NÃO EXCEDA LIMITES DE CPU, MEMÓRIA RAM E ARMAZENAMENTO)**

**OBS: Coloque a placa de ambas máquinas virtuais em Bridge**

**1 - Podemos colocar mais de um IP, funcionará como se fosse uma lista.**

**Assim podemos configurar mais de um grupo de hosts, exemplo:**

**No /etc/ansible/hosts, adicionar grupos de servidores ou tipos de equipamentos:**

**[meu\_primeiro\_servidor]**

**PRIMEIRO\_IP\_JÁ\_CRIADO**

**[webserverpicpay]**

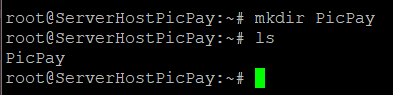
**<ip\_router\_01>**

**<ip\_router\_02>**

1. **Configurar nosso primeiro playbook, na máquina do Ansible, mas antes, configure um diretório**

**3.1 - Criar Diretório para organização do Playbook**

execute: **mkdir PicPay**



No Ansible, uma das melhores maneiras de organizar seus playbooks é por diretórios, por padrão, quando você executa um playbook.

O Ansible vai procurar o primeiro **playbook.yml** que estiver disponível, ou seja, varrendo a pasta, caso não encontre, poderá retornar algum erro ou aplicar as atualizações/downloads no grupo de máquinas erradas.

**3.2 - Crie um arquivo chamado playbook.yml**

execute: **touch PicPay/installdocker.yml**

**3.3 - Agora, preste atenção na indentação do código, o Ansible é extremamente rígido nesse quesito, podendo dar diversos erros e muitas vezes não muito claros (Copie o conteúdo abaixo)**

**Execute o comando abaixo direto no shell:**

*cat <<EOF> PicPay/installdocker.yml*

*---*

*- name: Instalar Docker e Docker Compose na VM*

*hosts: webservers*

*become: yes*

*tasks:*

*- name: Atualizar lista de pacotes*

*apt:*

*update\_cache: yes*

*- name: Instalar Docker*

*apt:*

*name: docker.io*

*state: present*

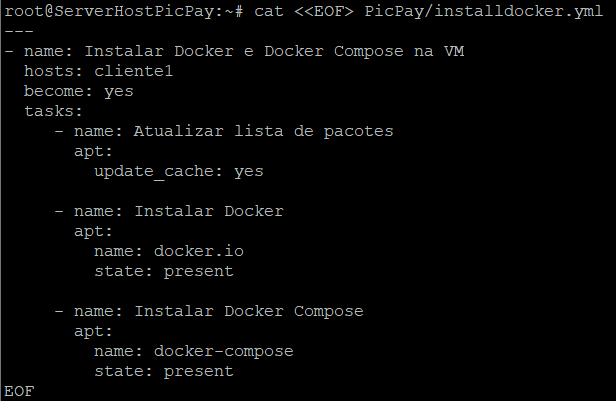
*- name: Instalar Docker Compose*

*apt:*

*name: docker-compose*

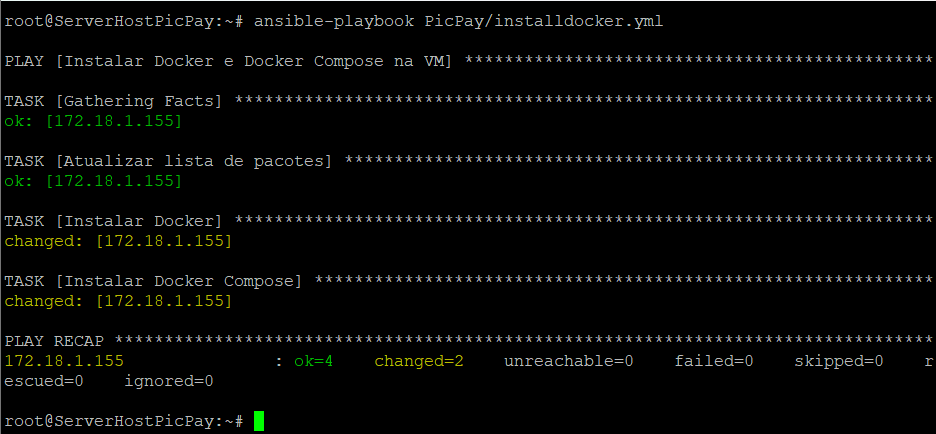
*state: present*

*EOF*

****

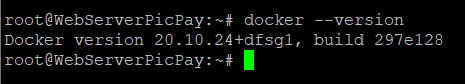
**3.4 - Agora vamos executar nosso playbook, para isso, se a instalação do ansible foi bem sucedida, teremos a linha de comando *ansible-playbook*:**

execute: **ansible-playbook PicPay/installdocker.yml**

****

**2.5 -Testar o comando Git na máquina WebserverPicPay**

execute: **docker --version**

****

**3 - Erros na execução do playbook, na máquina do Ansible**

Se houver um problema na hora de iniciarmos o nosso playbook, isso ocorreu por conta que no nosso playbook nós definimos um campo chamado

**become = true**, esse campo indica que as ações a seguir (tasks) deverão ser executadas com o usuário root ou usuário com sudo.

Há diversas maneiras de resolver isso:  
  
1 - Acessando a máquina secundária e tirando a senha do usuário root (NÃO RECOMENDADO)

2 - Permitindo que o usuário default da imagem (aluno) consiga baixar pacotes (NÃO RECOMENDADO)

3 - Criando um usuário do ansible com a permissão para baixar pacotes e fazer determinadas ações (RECOMENDADO)  
  
4 - Adicionar o parâmetro ***--ask-become-pass*** para que, quando rodarmos o comando novamente, o ansible nos pergunte sobre a senha do usuário root (NÃO RECOMENDADO EM AMBIENTES DE PRODUÇÃO)

Por questões de agilidade e tutorial, vamos utilizar a opção **4** portanto, execute o novo comando: **ansible-playbook PicPay/installdocker.yml --ask-become-pass**

****

**Digite a senha do seu usuário ROOT**

**Aguarde as execuções das tarefas e Resultado esperado acima**

Nosso apache foi instalado com sucesso e a página de testes parece funcionar. Vamos voltar para a máquina Ansible Host e ajustar o site, siga os próximos passos.