

Projeto de Bloco

TP9

Aluno: Hugo Florencio

Professor: Fábio Campos



OBSERVAÇÃO: Fábio só queria antes de iniciar o trabalho pedir desculpa pelo excesso de mensagem nos últimos dias ai, eu tava tentando descobrir um jeito de fazer o projeto sem usar o aws console, mas acabei aceitando que era o melhor a se fazer, porém tomei muito cuidado e não quis ficar subindo várias instâncias ou muitos componentes, por isso o trabalho pode não estar o mais completo possível. Um último detalhe, nos processos da etapa 9, que são a criação da instância eu tive um problema com o ssh_pubkey, por que eu não tinha essa chave, até pesquisei a criação dela, mas me retornava como se fosse uma imagem e não a chave por escrito, mas estou mandando o trabalho para que você possa ver pelo menos a forma dele e como ele está ficando e queria uma ajuda nessa parte da chave na aula segunda se possível para finalizar ele e ficar com tudo certinho e bonito para apresentação.

Sumário

1. Introdução	4
2. Objetivo	4
3. Desenvolvimento	5
3.1 Aplicações a serem desenvolvidas	5
3.2 - Identificação da função realizada por cada serviço	5
3.3 - Diagrama dos serviços envolvidos na implementação da aplicação	5
3.4 - A especificação da pilha de software adotada	6
3.5 - Estimativa de custo de investimento	6
4. Comparação entre essa implantação e outras abordagens, usando outras ferramentas.	8
5. planejamento da instalação	8
6. A instalação do projeto via Ansible:	9
6.1. Preparando o ambiente ansible para gerenciar a nossa nuvem:	9
6.2. O nosso ambiente na nuvem:	14
7. Execução de playbooks para executar a nossa aplicação dentro da nossa nuvem:	17

8. Meu repositório no github: 25

9. Bibliografia: 25

1. Introdução:

A empresa farmacêutica Florencios Farma pretende atuar em âmbito nacional, com filiais em todos os estados brasileiros, para isso deseja montar um sistema onde todos os medicamentos produzidos estejam catalogados com as informações de para quais estados foram e para quais lojas eles foram encaminhados, de forma que ela tenha controle sobre todos os produtos produzidos. E também é de desejo da empresa montar um site onde os clientes possam acessar para saber se o remédio que ele deseja tem no estoque na loja perto de sua casa.

2. Objetivo:

Montar um sistema em nuvem para que a empresa possa ter controle sobre seus produtos e que seja possível acessar em qualquer parte do país, de forma que a produção possa ser controlada e entendida mesmo que a fábrica seja em Manaus e o seu dono esteja no Rio de Janeiro, para isso é necessário a montagem de um banco de dados, e para que essas informações sejam disponibilizadas para os

clientes um site com acesso a esse banco de dados. Criar um ambiente em Ansible para realizar o controle dos componentes da nuvem, para que além do controle oferecido pela AWS a empresa tenha uma outra forma de realizar esse controle e segurança.

3. Desenvolvimento:

3.1. As aplicações escolhidas para desenvolvimento do nosso projeto são:

O wordpress para formação do site que ficará a disposição dos clientes para informações e também a montagem de um banco de dados MySQL para que as informações do site fiquem armazenadas e também as informações das produções dos remédios.

3.2. identificação da função realizada por cada serviço:

Antes de identificar as funções é importante ressaltar que os serviços a seguir serão provisionados pela AWS (Amazon Web Services) um serviço de nuvem, e que como o funcionamento da empresa a princípio é apenas em âmbito nacional, a região da AWS escolhida está localizada em São Paulo e seu código é “sa-east-1”.

- EC2: Também conhecido como Elastic Compute Cloud(EC2), este serviço da AWS será utilizado por nós para implantar a nossa aplicação, que será um site wordpress, o EC2 foi escolhido pois é um serviço que provê capacidade de computação flexível, redimensionável e segura sob demanda.

As instâncias EC2 para uma prática mais segura e redundante estariam distribuídas entre duas zonas de disponibilidade, de forma que 2 instâncias estariam em uma zona de disponibilidade, vamos chamar de zona A, e mais duas instâncias estariam disponíveis numa segunda zona de disponibilidade , chamada de zona B.

- S3: Simple Storage Service (S3) será utilizado pela empresa para armazenar os dados carregados no EC2, visto que o S3 é um serviço de armazenamento altamente confiável, tolerante a falhas e redundante, garantindo uma proteção dos dados.
- RDS: Relational Database Service (RDS) será o serviço utilizado para montar o banco de dados da empresa com todos os produtos produzidos e sua localização, a escolha desse serviço se deu a sua escalabilidade e alta performance, a plataforma escolhida pela

empresa para implementar o banco de dados foi o MySQL, e toda a manutenção do banco será feita pela AWS.

Assim com as instâncias EC2, o RDS também estaria distribuído em duas zonas de disponibilidade, cada zona com 1 RDS, seriam a zona A e a zona B.

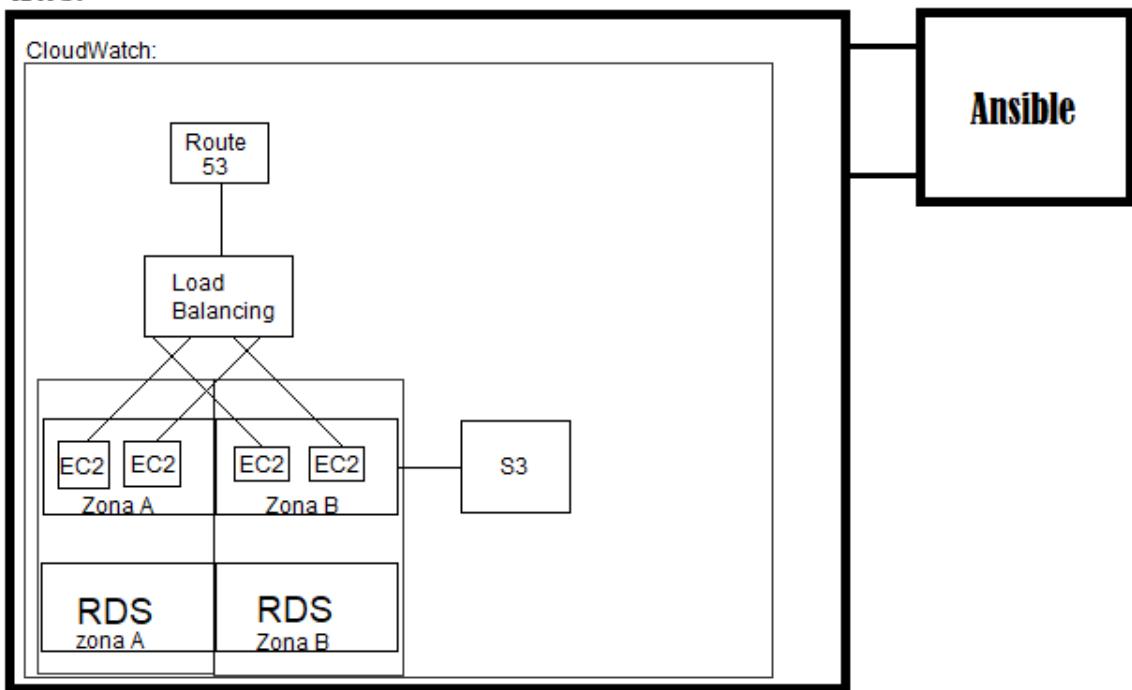
- VPC: Virtual Private Cloud, porém não provisiona uma nuvem privada e apenas a rede onde sua nuvem pode ficar, no caso da empresa será usado para que a aplicação fique na mesma rede do banco de dados, com a diferença de que a rede da aplicação terá acesso público à internet, enquanto a rede do banco de dados por motivos de segurança será uma rede privada.

A importância dos componentes estarem na mesma rede é justamente a de poderem ter uma “conversa” entre eles, ou seja, o ambiente em que está a aplicação consegue “ver” o ambiente do banco de dados, dessa forma toda nossa infraestrutura se torna mais simples e de melhor controle, desde que medidas de segurança sejam adotadas como o que já foi citado anteriormente de que o banco de dados fique privado sem acesso público.

- Amazon CloudWatch: Este serviço será utilizado para monitorar todo o ambiente da empresa que está no AWS.
- Elastic Load Balancing: Distribui automaticamente o tráfego de entrada de aplicações entre diversos destinos, como instâncias do Amazon EC2, contêineres, endereços IP, funções do Lambda e dispositivos virtuais.
- Route 53: Um web service Domain Name System (DNS) na nuvem altamente disponível e escalável. Ele foi projetado para oferecer aos desenvolvedores e empresas uma maneira altamente confiável e econômica de direcionar os usuários finais aos aplicativos de Internet, convertendo nomes como `www.example.com` para endereços IP numéricos como `192.0.2.1`, usados pelos computadores para se conectarem entre si.

3.3. Diagrama dos serviços envolvidos na implementação da aplicação:

AWS:



3.4 A especificação da pilha de software adotada:

Para o funcionamento dos serviços adotados usaremos o SO linux, o linux será usado tanto na máquina física da empresa para acesso do ambiente em nuvem da AWS assim como no próprio ambiente de trabalho na nuvem, e o banco de dados escolhido pela a empresa foi o MySQL também instalado em um ambiente linux.

3.5 estimativa de custo de investimento:

	A	B	C	D	E	F	G	H	I	J
2	Service Type,Components,Region,Component Price,Service Price									
3	Amazon EC2 Service (South America (Sao Paulo)),,,,,\$904.02									
4	,Compute:,South America (Sao Paulo),\$904.02									
5	Amazon S3 Service (South America (Sao Paulo)),,,,,\$5.85									
6	,S3 Standard Storage:,South America (Sao Paulo),\$0.65									
7	,S3 Standard Put Requests:,South America (Sao Paulo),\$1.4									
8	,S3 Standard Other Requests:,South America (Sao Paulo),\$0.12									
9	,S3 Standard Select Data Returned:,South America (Sao Paulo),\$0.03									
10	,S3 Standard Select Data Scanned:,South America (Sao Paulo),\$0.07									
11	,Cross Region Replication - Storage:,South America (Sao Paulo),\$0.37									
12	,Cross Region Replication - Requests:,South America (Sao Paulo),\$1									
13	,Cross Region Replication - Inter Region Data Transfer:,South America (Sao Paulo),\$2.21									
14	Amazon RDS Service (South America (Sao Paulo)),,,,,\$114.03									
15	,DB Instances:,South America (Sao Paulo),\$106.14									
16	,Storage:,South America (Sao Paulo),\$7.89									
17	Amazon CloudWatch Service (South America (Sao Paulo)),,,,,\$10.8									
18	,Standard Alarms:,South America (Sao Paulo),\$10.8									
19	AWS Support (Business),,,,,\$103.01									
20	,Support for all AWS services:,,,\$103.01									
21	,,Free Tier Discount:,,,\$-4.63									
22	,,Total Monthly Payment:,,,\$1133.08									

4. Comparação entre essa implantação e outras abordagens, usando outras ferramentas:

A escolha pela AWS foi feita pois pensando no propósito da empresa, ficou evidente que entre as opções apresentadas, que eram com vSphere, AWS ou OpenStack, a AWS seria aquela que precisaria de uma menor supervisão, visto que as outras infraestruturas contêm mais componentes e precisam de atenção em diversos outros detalhes e aspectos, sendo que o propósito da empresa não é sobre ter total controle sobre arquitetura mas sim ter a disponibilidade e escalabilidade para aumentar ou diminuir se necessário e apresentá-la para o público, portanto redes mais encorpadas com hardwares mais “pesados” e caros não são de extrema importância, e ainda com a AWS há a possibilidade de se conversar com a Amazon, conseguir certos descontos por tempo de uso e poder usar os datacenters deles visto que é uma opção em nuvem, por estes motivos explicitados a escolha foi pela AWS.

5. planejamento da instalação:

A instalação deve começar construindo a rede em que todos os componentes ficarão alocaados, para isso usamos o VPC da aws, ao criar a

rede podemos criar as subredes, criaremos duas subredes nessa etapa do projeto, cada uma em uma zona de disponibilidade diferente, após a criação das subredes já poderemos criar as instâncias, assim como as subredes cada uma terá que estar em uma zona de disponibilidade diferente. Agora que temos a base do nosso projeto iremos criar nosso banco de dados (RDS), ao criarmos iremos escolher a opção Multi AZ, assim nosso banco será duplicado e podemos por um em uma região diferente para termos mais segurança. Assim que acabamos a criação do banco e botamos o seu clone em uma outra região, vamos mudar de região dentro da aws e espelhar o que criamos na zona A para essa outra região, zona B. Após essas próximas criações pegamos o endereço público de nossas VMs que estão com a aplicação e jogaremos esses endereços para dentro do load balance que será criado agora, o load balance transformará esses endereços e os transformará em um só, dessa forma podemos usar esse endereço único e utilizar o Route 5 para formar um dns.

6. A instalação do projeto via Ansible:

6.1. Preparando o ambiente ansible para gerenciar a nossa nuvem:

Através do Ansible criamos um ambiente capaz de gerenciar e comandar a nossa nuvem AWS. Para isso foi necessário criar o ambiente propício, ou seja, baixar as versões do python correspondentes com nosso ambiente e utilizar as chaves de acesso fornecidas pela AWS para poder fazer seu gerenciamento. Encontramos algumas dificuldades nessa etapa, tais quais, a primeira versão do python que baixamos para utilizar estava incorreta, portanto os comando não funcionaram, quando percebemos o problema voltamos a VM para o snapshot que havíamos tirado antes de iniciarmos a modelagem do ambiente, isso nos permitiu baixar a versão correta do python e poder realizar os comando novamente (os mesmos comandos foram dados, por isso as telas de evidências dos procedimentos são da versão original (a primeira) com o python errado. A diferença foi que os primeiros comandos foram “\$ apt install python3-pip” e “\$ pip3 install boto” e depois esses comandos foram alterados para “\$ apt install python” e “\$ pip install boto”. Após essa mudança todo o resto executado foi exatamente igual.

```
root@ubuntu:/home/hugoflorencio
File Edit View Search Terminal Help
hugoflorencio@ubuntu:~$ export aws_secret_access_key='nhwFikZ506SwaBhpOA/wQBBewvyGxFoRMnRDAJ50'
hugoflorencio@ubuntu:~$ sudo pip install Boto
sudo: pip: command not found
hugoflorencio@ubuntu:~$ apt install python3-pip
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are you root?
hugoflorencio@ubuntu:~$ sudo su
root@ubuntu:/home/hugoflorencio# apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (9.0.1-2.3~ubuntu1.18.04.4).
The following packages were automatically installed and are no longer required:
  apache2-data apache2-utils efibootmgr gir1.2-geocodeglib-1.0 libegl1-mesa
  libfwupd1 libllvm8 libwayland-egl1-mesa linux-headers-5.0.0-23
  linux-headers-5.0.0-23-generic linux-image-5.0.0-23-generic
  linux-modules-5.0.0-23-generic linux-modules-extra-5.0.0-23-generic
  ubuntu-web-launchers
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
root@ubuntu:/home/hugoflorencio#
```

```
root@ubuntu:/home/hugoflorencio
File Edit View Search Terminal Help
ou root?
hugoflorencio@ubuntu:~$ sudo su
root@ubuntu:/home/hugoflorencio# apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (9.0.1-2.3~ubuntu1.18.04.4).
The following packages were automatically installed and are no longer required:
  apache2-data apache2-utils efibootmgr gir1.2-geocodeglib-1.0 libegl1-mesa
  libfwupd1 libllvm8 libwayland-egl1-mesa linux-headers-5.0.0-23
  linux-headers-5.0.0-23-generic linux-image-5.0.0-23-generic
  linux-modules-5.0.0-23-generic linux-modules-extra-5.0.0-23-generic
  ubuntu-web-launchers
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
root@ubuntu:/home/hugoflorencio# pip3 install boto
Collecting boto
  Downloading https://files.pythonhosted.org/packages/23/10/c0b78c27298029e4454a472a1919bde20cb182dab1662cec7f2ca1dcc523/boto-2.49.0-py2.py3-none-any.whl (1.4MB)
)    100% |██████████| 1.4MB 1.1MB/s
Installing collected packages: boto
Successfully installed boto-2.49.0
root@ubuntu:/home/hugoflorencio#
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
  command 'dtc' from deb device-tree-compiler
  command 'ddtc' from deb ddtc

See 'snap info <snapname>' for additional versions.

root@ubuntu:/home/hugoflorencio#
root@ubuntu:/home/hugoflorencio# cd /etc/ansible/
root@ubuntu:/etc/ansible# wget https://raw.githubusercontent.com/ansible/ansible/stable-1.9/plugins/inventory/ec2.py
--2021-03-16 10:47:54-- https://raw.githubusercontent.com/ansible/ansible/stable-1.9/plugins/inventory/ec2.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.1
33, 185.199.111.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 30905 (30K) [text/plain]
Saving to: 'ec2.py'

ec2.py          100%[=====] 30.18K  ---KB/s   in 0.001s

2021-03-16 10:47:54 (21.6 MB/s) - 'ec2.py' saved [30905/30905]

root@ubuntu:/etc/ansible#
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
See 'snap info <snapname>' for additional versions.

root@ubuntu:/home/hugoflorencio#
root@ubuntu:/home/hugoflorencio# cd /etc/ansible/
root@ubuntu:/etc/ansible# wget https://raw.githubusercontent.com/ansible/ansible/stable-1.9/plugins/inventory/ec2.py
--2021-03-16 10:47:54-- https://raw.githubusercontent.com/ansible/ansible/stable-1.9/plugins/inventory/ec2.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.1
33, 185.199.111.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 30905 (30K) [text/plain]
Saving to: 'ec2.py'

ec2.py          100%[=====] 30.18K  ---KB/s   in 0.001s

2021-03-16 10:47:54 (21.6 MB/s) - 'ec2.py' saved [30905/30905]

root@ubuntu:/etc/ansible# ls
ansible.cfg  ec2.py  hosts  hosts.save  roles
root@ubuntu:/etc/ansible#
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
ec2.py          100%[=====] 30.18K ---KB/s   in 0.001s
2021-03-16 10:47:54 (21.6 MB/s) - 'ec2.py' saved [30905/30905]

root@ubuntu:/etc/ansible# ls
ansible.cfg  ec2.py  hosts  hosts.save  roles
root@ubuntu:/etc/ansible# wget https://raw.githubusercontent.com/ansible/ansible/stable-1.9/plugins/inventory/ec2.ini
--2021-03-16 10:50:14-- https://raw.githubusercontent.com/ansible/ansible/stable-1.9/plugins/inventory/ec2.ini
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.1
33, 185.199.110.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4875 (4.8K) [text/plain]
Saving to: 'ec2.ini'

ec2.ini          100%[=====] 4.76K ---KB/s   in 0s
2021-03-16 10:50:14 (20.5 MB/s) - 'ec2.ini' saved [4875/4875]
root@ubuntu:/etc/ansible#
```

Tue 10:56

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
root@ubuntu:/etc/ansible# ls
ansible.cfg  ec2.py  hosts  hosts.save  roles
root@ubuntu:/etc/ansible# wget https://raw.githubusercontent.com/ansible/ansible/stable-1.9/plugins/inventory/ec2.ini
--2021-03-16 10:50:14-- https://raw.githubusercontent.com/ansible/ansible/stable-1.9/plugins/inventory/ec2.ini
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.1
33, 185.199.110.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4875 (4.8K) [text/plain]
Saving to: 'ec2.ini'

ec2.ini          100%[=====] 4.76K ---KB/s   in 0s
2021-03-16 10:50:14 (20.5 MB/s) - 'ec2.ini' saved [4875/4875]

root@ubuntu:/etc/ansible# nano ec2.py
root@ubuntu:/etc/ansible# chmod +x ec2.py
root@ubuntu:/etc/ansible# ls
ansible.cfg  ec2.ini  ec2.py  hosts  hosts.save  roles
root@ubuntu:/etc/ansible#
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4875 (4.8K) [text/plain]
Saving to: 'ec2.ini'

ec2.ini      100%[=====] 4.76K --.-KB/s   in 0s

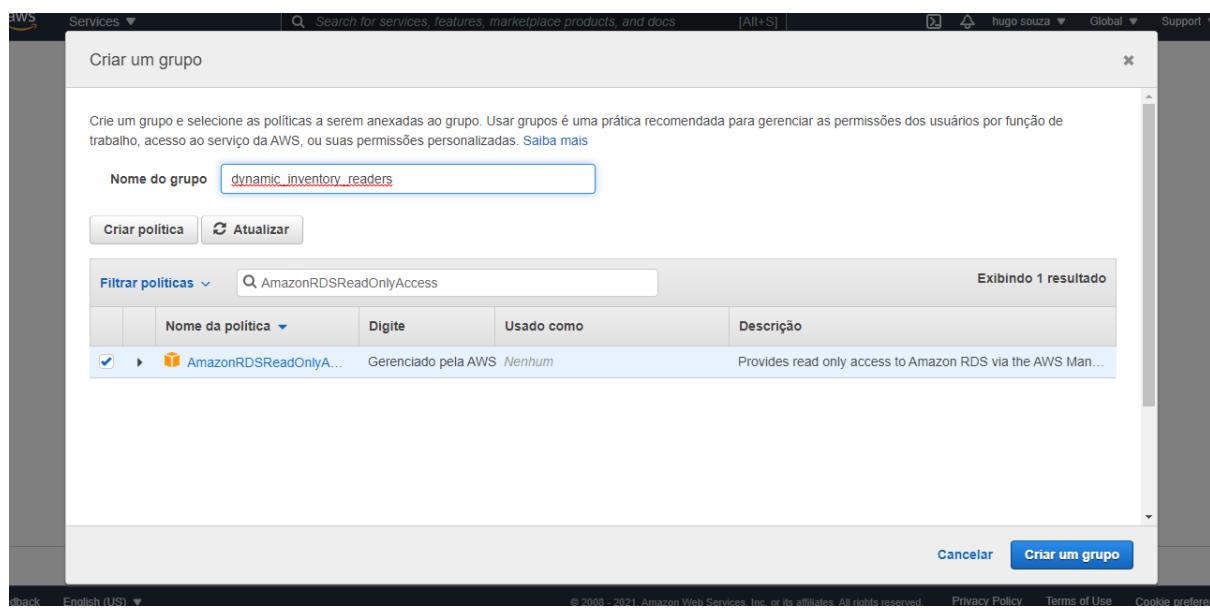
2021-03-19 07:33:43 (14.8 MB/s) - 'ec2.ini' saved [4875/4875]

root@ubuntu:/etc/ansible# ls
ansible.cfg  ec2.ini  ec2.py  hosts  hosts.save  roles
root@ubuntu:/etc/ansible# chmod +x ec2.py
root@ubuntu:/etc/ansible# ls
ansible.cfg  ec2.ini  ec2.py  hosts  hosts.save  roles
root@ubuntu:/etc/ansible# export AWS_ACCESS_KEY_ID='AKIAYT6EIMRF2CVOY2UB'
root@ubuntu:/etc/ansible# export AWS_SECRET_ACCESS_KEY='0a8DHth7D0lJWjKtwY+owxKN
0w7A241181foMso0'
root@ubuntu:/etc/ansible# export ANSIBLE_HOSTS=/etc/ansible/ec2.py
root@ubuntu:/etc/ansible# export EC2_INI_PATH=/etc/ansible/ec2.ini
root@ubuntu:/etc/ansible# nano ansible.cfg
root@ubuntu:/etc/ansible# nano ansible.cfg
root@ubuntu:/etc/ansible# ./ec2.py --list
```

```
OpenStack Fri 07:49
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
root@ubuntu:/etc/ansible# ./ec2.py --list
{
  "_meta": {
    "hostvars": {
      "100.25.31.2": {
        "ec2__in_monitoring_element": false,
        "ec2_ami_launch_index": "0",
        "ec2_architecture": "x86_64",
        "ec2_client_token": "",
        "ec2_dns_name": "ec2-100-25-31-2.compute-1.amazonaws.com",
        "ec2_ebs_optimized": false,
        "ec2_eventsSet": "",
        "ec2_group_name": "",
        "ec2_hypervisor": "xen",
        "ec2_id": "i-062aa0e9ccf511976",
        "ec2_image_id": "ami-038f1ca1bd58a5790",
        "ec2_instance_profile": "",
        "ec2_instance_type": "t2.micro",
        "ec2_ip_address": "100.25.31.2",
        "ec2_item": "",
        "ec2_kernel": "",
        "ec2_key_name": "hugoflorencio",
        "ec2_launch_time": "2021-03-19T14:26:16.000Z",
        "ec2_monitored": false
      }
    }
  }
}
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
100.25.31.2
],
"ec2": [
    "100.25.31.2"
],
"i-062aa0e9ccf511976": [
    "100.25.31.2"
],
"key_hugoflorencio": [
    "100.25.31.2"
],
"security_group_tp8": [
    "100.25.31.2"
],
>tag_Group__dynamic_inventory_readers": [
    "100.25.31.2"
],
>tag_Name_TP8_instancia": [
    "100.25.31.2"
],
"type_t2_micro": [
    "100.25.31.2"
],
"us-east-1": [
    "100.25.31.2"
]
```

6.2. O nosso ambiente na nuvem:



Éxito

Você criou com êxito os usuários mostrados abaixo. Você pode visualizar e fazer download das credenciais de segurança do usuário. Você também pode enviar um e-mail aos usuários com as instruções para fazer login no Console de Gerenciamento da AWS. Esta é a última vez que essas credenciais estarão disponíveis para download. No entanto, você pode criar novas credenciais a qualquer momento.

Os usuários com acesso ao Console de Gerenciamento da AWS podem fazer login em:
<https://592580207691.signin.aws.amazon.com/console>

[Fazer download .csv](#)

	Usuário	ID da chave de acesso	Chave de acesso secreta
▶	<input checked="" type="checkbox"/> inventory_reader	AKIAYT6EIMRF2COY2UB	Oa8Dhth7DOljWjKtwY+owx KNOW7A241181foMs00 Ocultar

[Fechar](#)

1. Selecionar a AMI 2. Escolher tipo de instância 3. Configurar instância 4. Adicionar armazenamento 5. Adicionar Tags 6. Configure o security group 7. Análise

Etapa 1: Selecione uma Imagem de máquina da Amazon (AMI)

Uma AMI é um modelo que contém a configuração do software (sistema operacional, servidor de aplicativos e aplicativos) necessária para executar a instância. Você pode selecionar uma AMI fornecida pela AWS, por nossa comunidade de usuários ou no AWS Marketplace, ou pode selecionar uma das suas próprias AMIs.

Procure uma AMI digitando um termo de pesquisa; por exemplo, "Windows" X

[Pesquisar por parâmetro do Systems Manager](#)

[Cancelar e sair](#)

Inicio rápido

Minhas AMIs	Amazon Linux qualificado para o	Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-038f1ca1bd58a5790 (64 bits x86) / ami-01437b695ea9225cd (64 bits Arm)	Selecionar
AWS Marketplace		Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is approaching end of life on December 31, 2020 and has been removed from this wizard.	<input checked="" type="radio"/> 64 bits (x86) <input type="radio"/> 64 bits (Arm)
AMIs da comunidade		Tipo de dispositivo raiz: ebs Tipo de virtualização: hvm ENA habilitado: Sim	
Somente nível gratuito			

[macOS Big Sur 11.2.1](#) - ami-0af92ed5f8a74cda3 [Selecionar](#)

64 bits (Mac)

Etapa 2: Escolha um tipo de instância

O Amazon EC2 oferece uma ampla seleção de tipos de instâncias otimizadas para se adequarem a casos de uso diferentes. Instâncias são servidores virtuais que podem executar aplicativos. Possuem várias combinações de CPU, memória, armazenamento e capacidade de rede e oferecem flexibilidade de escolha da composição adequada de recursos para os seus aplicativos. [Saiba mais](#) sobre tipos de instância e como podem atender às suas necessidades de computação.

Filtrar por: [Todas as famílias de instâncias](#) [Geração atual](#) [Mostrar/ocultar colunas](#)

Selecionada atualmente: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memória, Somente EBS)

	Família	Tipo	vCPUs	Memória (GiB)	Armazenamento da instância (GiB)	Disponível otimizado para EBS	Desempenho de rede	Compatibilidade com IPv6
<input type="checkbox"/>	t2	t2.nano	1	0.5	Somente EBS	-	Baixo a moderado	Sim
<input checked="" type="checkbox"/>	t2	t2.micro qualificado para o nível gratuito	1	1	Somente EBS	-	Baixo a moderado	Sim
<input type="checkbox"/>	t2	t2.small	1	2	Somente EBS	-	Baixo a moderado	Sim
<input type="checkbox"/>	t2	t2.medium	2	4	Somente EBS	-	Baixo a moderado	Sim
<input type="checkbox"/>	t2	t2.large	2	8	Somente EBS	-	Baixo a moderado	Sim

[Cancelar](#) [Anterior](#) [Verificar e ativar](#) [Próximo: Configure os detalhes da instância](#)

Etapa 6: Configure o security group

Um grupo de segurança é um conjunto de regras de firewall que controla o tráfego da sua instância. Nesta página, você pode adicionar regras para permitir que tráfegos específicos cheguem até a sua instância. Por exemplo, se você quiser configurar um servidor Web e permitir que tráfego da Internet chegue até a sua instância, adicione regras que permitam acesso irrestrito às portas HTTP e HTTPS. Você pode criar um novo grupo de segurança ou selecionar um dos existentes abaixo. Saiba mais sobre grupo de segurança do Amazon EC2.

Atribuir um grupo de segurança: Criar um grupo de segurança **novo** Selecionar um grupo de segurança **existente**

Nome do grupo de segurança: Acesso_inventario_dinamico

Descrição: launch-wizard-4 created 2021-03-16T14:23:51.196-03:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Anywhere	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere	e.g. SSH for Admin Desktop
LDAP	TCP	389	Anywhere	e.g. SSH for Admin Desktop

Add Rule

⚠️ Aviso

Cancelar **Anterior** **Verificar e ativar**

Selecionar um par de chaves existente ou crie um novo par de chaves

Um par de chaves é uma **chave pública** armazenada pela AWS, e um **arquivo de chave privada** que é armazenado por você. Juntos, eles permitem que você se conecte à sua instância com segurança. Para AMIs do Windows, para obter a senha usada para fazer login na sua instância, é necessário o arquivo de chave privada. Para AMIs do Linux, o arquivo de chave privada permite fazer SSH com segurança na sua instância.

Observação: O par de chaves selecionado será adicionado ao conjunto de chaves autorizado para essa instância. Saiba mais sobre [Como remover pares de chaves existentes de uma AMI pública](#).

Escolher um par de chaves existente

Selecionar um par de chaves

hugo281099

Eu reconheço que posso acesso ao arquivo de chave privada selecionado (hugo281099.pem) e que, sem esse arquivo, não poderei fazer login na minha instância.

Cancelar **Executar instâncias**

Instâncias (1/1) Informações

Name	ID de instância	Estado da inst...	Tipo de inst...	Verificação de s...	Status do al...	Zona de dispon...
TP8_instancia	i-062aa0e9ccf511976	Executando	t2.micro	2/2 verificações a	Sem alar...	us-east-1e

Instância: i-062aa0e9ccf511976 (TP8_instancia)

- Detalhes**
- Segurança
- Redes
- Armazenamento
- Verificações de status
- Monitoramento
- Tags

Resumo da instância

ID de instância i-062aa0e9ccf511976 (TP8_instancia)	Endereço IPv4 público 100.25.31.2 endereço aberto	Endereços IPv4 privados 172.31.50.244
Estado da instância Executando	DNS IPv4 público ec2-100-25-31-2.compute-1.amazonaws.com endereço aberto	DNS IPv4 privado ip-172-31-50-244.ec2.internal
Tipo de instância t2.micro	Endereços IP elásticos -	ID da VPC vpc-414cf13c

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#) [Cookie preferences](#)

7. Execução de playbooks para executar a nossa aplicação dentro da nossa nuvem:

Agora com a ajuda de um pequeno playbook, iremos apagar a instância criada anteriormente e depois com ajuda de outros playbooks e inventários dinâmicos subiremos a instância onde ficará nossa aplicação.

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
],
"tag_Name_TP8_instancia": [
    "100.25.31.2"
],
"type_t2_micro": [
    "100.25.31.2"
],
"us-east-1": [
    "100.25.31.2"
],
"us-east-1e": [
    "100.25.31.2"
],
"vpc_id_vpc-414cf13c": [
    "100.25.31.2"
]
}
root@ubuntu:/etc/ansible#
root@ubuntu:/etc/ansible#
root@ubuntu:/etc/ansible#
root@ubuntu:/etc/ansible# ls
ansible.cfg ec2.ini ec2.py hosts hosts.save roles
root@ubuntu:/etc/ansible# nano exclui-hosts.yml
root@ubuntu:/etc/ansible#
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
GNU nano 2.9.3          exclui-hosts.yml          Modified
---
# Exemplo basico de exclusao de hosts
- name: Exclui instancias EC2 de desenvolvimento
  hosts: tag_group_Dynamic_Inventory_Demo
  connection: local
  gather_facts: False
  tasks:
    - name: Exclui a instancia {{ ec2_id }} da regiao {{ aws_region }}
      local_action:
        module: ec2
        state: 'absent'
        region: '{{ ec2_region }}'
        instance_ids: '{{ ec2_id }}'■

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text^T To Spell ^L Go To Line
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
PLAY RECAP
100.25.31.2 : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0

root@ubuntu:/etc/ansible# nano exclui-hosts.yml
root@ubuntu:/etc/ansible# ansible-playbook -i ec2.py exclui-hosts.yml
[WARNING]: Invalid characters were found in group names but not replaced, use
-vvvv to see details

PLAY [Exclui instancias EC2 de desenvolvimento] *****

TASK [Exclui a instancia i-062aa0e9ccf511976 da regiao us-east-1] *****
fatal: [100.25.31.2]: FAILED! => {"changed": false, "msg": "Unable to terminate instance i-062aa0e9ccf511976, error: EC2ResponseError: 403 Forbidden\n<?xml version='1.0' encoding='UTF-8'?>\n<Response><Errors><Error><Code>UnauthorizedOperation</Code><Message>You are not authorized to perform this operation. Encoded authorization failure message: YutPMLWE0B59AJ8pht9380W0zwE34vp5rN3v9-ZZ3KE-3A8JhEQazSKY1ywohv-bUBF7nQoy0000dka_daCXBqSPVe1e0EUFBqQGR8dqa3Io08mKRA9kKlrLvd6GLy2u69g4AY48DiW4JKWL0Mw7XYXctjYcPt6A3-5g0MLD4Y0ChqJlQx4MY_pNAz59W4qPxvUl_nWUaCydEKwGGMvgqiF0u2cbDcJ07ogr9KQ_-8b60PBI0yBx-y0pRrBToF9hld8CP4sSX2EzyjCFT0pcIp4cC547N0rYCUCqUaVmtK6wtlhe5MtQRRrLiI40HDyYbdZ0ABuMzM50xr1kguGqINzusiJ0lHuHLMrKAbmX4NDt_3z_AllhzF5fcWZXaD2qa-eAA08XEY00m0_5rr8DAxMl6a6B98pqrflVgiDhx806ayAzncT7Ml-_JJxa9v_Cd_Glzhodhm8GI4XE3v0_yEeah2z4iHBoLIF2Rb3tDS9QyJIN6PEKrzY9cRfzHgY0eMCN5qWsw2WXji2WF5T65shrekLxkup_tnXPtiBnMZ5fIGbiQHQV3yTiAYqswJbHcYIb7YUT0HFGDDfUTC4E2Lkhff82EmBboNgR7020RXMrjhE57ox9kjx33SI3A3iDsBmIPos-nh9v9MiwjchrLCMwqZpGdn39Sopv3_pfWN3cA4IgxFwKPHrcy-vyBw5qb4ZKZZm3ucHoMs1Fid04RJMv3bv0IpLfRT6BPnNI8x5D6Pkzbio9chksW0xLxyIHe_vIu-yiRycqsWKAM3vRPZImZfM7tdCXu2JCHbkTR84C-DB</Message></Error><RequestID>d7fbfa5e62-f226-4251-a3ab-c2bbbf975603</RequestID></Response>"}

PLAY RECAP *****
100.25.31.2 : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
TASK [Exclui a instancia i-062aa0e9ccf511976 da regiao us-east-1] *****
fatal: [100.25.31.2]: FAILED! => {"changed": false, "msg": "Unable to terminate instance i-062aa0e9ccf511976, error: EC2ResponseError: 403 Forbidden\n<?xml version='1.0' encoding='UTF-8'?>\n<Response><Errors><Error><Code>UnauthorizedOperation</Code><Message>You are not authorized to perform this operation. Encoded authorization failure message: YutPMLWE0B59AJ8pht9380W0zwE34vp5rN3v9-ZZ3KE-3A8JhEQazSKY1ywohv-bUBF7nQoy0000dka_daCXBqSPVe1e0EUFBqQGR8dqa3Io08mKRA9kKlrLvd6GLy2u69g4AY48DiW4JKWL0Mw7XYXctjYcPt6A3-5g0MLD4Y0ChqJlQx4MY_pNAz59W4qPxvUl_nWUaCydEKwGGMvgqiF0u2cbDcJ07ogr9KQ_-8b60PBI0yBx-y0pRrBToF9hld8CP4sSX2EzyjCFT0pcIp4cC547N0rYCUCqUaVmtK6wtlhe5MtQRRrLiI40HDyYbdZ0ABuMzM50xr1kguGqINzusiJ0lHuHLMrKAbmX4NDt_3z_AllhzF5fcWZXaD2qa-eAA08XEY00m0_5rr8DAxMl6a6B98pqrflVgiDhx806ayAzncT7Ml-_JJxa9v_Cd_Glzhodhm8GI4XE3v0_yEeah2z4iHBoLIF2Rb3tDS9QyJIN6PEKrzY9cRfzHgY0eMCN5qWsw2WXji2WF5T65shrekLxkup_tnXPtiBnMZ5fIGbiQHQV3yTiAYqswJbHcYIb7YUT0HFGDDfUTC4E2Lkhff82EmBboNgR7020RXMrjhE57ox9kjx33SI3A3iDsBmIPos-nh9v9MiwjchrLCMwqZpGdn39Sopv3_pfWN3cA4IgxFwKPHrcy-vyBw5qb4ZKZZm3ucHoMs1Fid04RJMv3bv0IpLfRT6BPnNI8x5D6Pkzbio9chksW0xLxyIHe_vIu-yiRycqsWKAM3vRPZImZfM7tdCXu2JCHbkTR84C-DB</Message></Error><RequestID>d7fbfa5e62-f226-4251-a3ab-c2bbbf975603</RequestID></Response>"}

PLAY RECAP *****
100.25.31.2 : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0

root@ubuntu:/etc/ansible#
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
],
"tag_Name_TP8_instancia": [
    "100.25.31.2"
],
"type_t2_micro": [
    "100.25.31.2"
],
"us-east-1": [
    "100.25.31.2"
],
"us-east-1e": [
    "100.25.31.2"
],
"vpc_id_vpc-414cf13c": [
    "100.25.31.2"
]
}
root@ubuntu:/etc/ansible#
root@ubuntu:/etc/ansible#
root@ubuntu:/etc/ansible#
root@ubuntu:/etc/ansible# ls
ansible.cfg ec2.ini ec2.py hosts hosts.save roles
root@ubuntu:/etc/ansible# nano exclui-hosts.yml
root@ubuntu:/etc/ansible#
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
GNU nano 2.9.3          exclui-hosts.yml          Modified
---
# Exemplo basico de exclusao de hosts
- name: Exclui instancias EC2 de desenvolvimento
  hosts: tag_group_Dynamic_Inventory_Demo
  connection: local
  gather_facts: False
  tasks:
    - name: Exclui a instancia {{ ec2_id }} da regiao {{ aws_region }}
      local_action:
        module: ec2
        state: 'absent'
        region: '{{ ec2_region }}'
        instance_ids: '{{ ec2_id }}'
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^L Go To Line

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
PLAY RECAP
100.25.31.2 : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0

root@ubuntu:/etc/ansible# nano exclui-hosts.yml
root@ubuntu:/etc/ansible# ansible-playbook -i ec2.py exclui-hosts.yml
[WARNING]: Invalid characters were found in group names but not replaced, use
-vvvv to see details

PLAY [Exclui instancias EC2 de desenvolvimento] *****

TASK [Exclui a instancia i-062aa0e9ccf511976 da regiao us-east-1] *****
fatal: [100.25.31.2]: FAILED! => {"changed": false, "msg": "Unable to terminate instance i-062aa0e9ccf511976, error: EC2ResponseError: 403 Forbidden\n<?xml version='1.0' encoding='UTF-8'?>\n<Response><Errors><Error><Code>UnauthorizedOperation</Code><Message>You are not authorized to perform this operation. Encoded authorization failure message: YutPMlWE0B59AJ8pht9380W0zwE34vp5rN3v9-ZZ3KE-3A8JhEQazSKY1ywohv-bUBF7nQoy0000dka_daCXBqSPVe1e0EUFBqQGR8dqa3Io08mKRA9kKlrLvd6GLy2u69g4AY48DiW4JKWL0Mw7XYXctjYcPt6A3-5gOMPLD4Y0ChqJlQx4MY_pNAz59W4qPxvUl_nWUaCydEKwGGMvgqiFOu2cbDcJ07ogr9KQ_-8b60PBI0yBx-y0pRrBToF9hld8CP4sSX2EzyjCFT0pcIp4cC547N0rYCUCqUaVmtK6wtlhe5MtQRRrLiI40HDyYbdZ0ABuMzM50xr1kguGqINzusiJ0lHuHLMrKAbmX4NDt_3z_AllhzF5fcWZXaD2qa-eAA08XEY00m0_5rr8DAxMl6a6B98pqrflVgiDhxB06ayAzncT7Ml-_JJxa9v_Cd_Glzhodh8GI4XE3v0_yEeah2z4iHBoLif2Rb3tDS9QyJIN6PEKrZy9cRfzHgY0eMCN5qWsw2WXji2WF5T65shreVklxkup_tnXPtiBnMZ5fIGbiQHQV3yTiAYqswJbHcYIb7YUT0HFGDDfUTC4E2Lkhff82Em8boNgR7020RXMrjhE57ox9kjx33SI3A3iDsBmIPos-nh9v9MiwjchrLCMwqZpGdn39Sopv3_pfWN3cA4IxgxFwKPhrcy-vyBw5qb4ZKZZm3ucHoMs1FlD04RJMv3bv0IpLfrT6BPnNI8x5D6Pkzbio9chksW0xLxyIhe_vIu-iyRycqsWKAM3vRPZImZFm7tdCxu2JCHbkTR84C-DB</Message></Error><Errors><RequestID>dfba5e62-f226-4251-a3ab-c2bbbf975603</RequestID></Response>"}

PLAY RECAP *****
100.25.31.2 : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
TASK [Exclui a instancia i-062aa0e9ccf511976 da regiao us-east-1] *****
fatal: [100.25.31.2]: FAILED! => {"changed": false, "msg": "Unable to terminate instance i-062aa0e9ccf511976, error: EC2ResponseError: 403 Forbidden\n<?xml version='1.0' encoding='UTF-8'?>\n<Response><Errors><Error><Code>UnauthorizedOperation</Code><Message>You are not authorized to perform this operation. Encoded authorization failure message: YutPMlWE0B59AJ8pht9380W0zwE34vp5rN3v9-ZZ3KE-3A8JhEQazSKY1ywohv-bUBF7nQoy0000dka_daCXBqSPVe1e0EUFBqQGR8dqa3Io08mKRA9kKlrLvd6GLy2u69g4AY48DiW4JKWL0Mw7XYXctjYcPt6A3-5gOMPLD4Y0ChqJlQx4MY_pNAz59W4qPxvUl_nWUaCydEKwGGMvgqiFOu2cbDcJ07ogr9KQ_-8b60PBI0yBx-y0pRrBToF9hld8CP4sSX2EzyjCFT0pcIp4cC547N0rYCUCqUaVmtK6wtlhe5MtQRRrLiI40HDyYbdZ0ABuMzM50xr1kguGqINzusiJ0lHuHLMrKAbmX4NDt_3z_AllhzF5fcWZXaD2qa-eAA08XEY00m0_5rr8DAxMl6a6B98pqrflVgiDhxB06ayAzncT7Ml-_JJxa9v_Cd_Glzhodh8GI4XE3v0_yEeah2z4iHBoLif2Rb3tDS9QyJIN6PEKrZy9cRfzHgY0eMCN5qWsw2WXji2WF5T65shreVklxkup_tnXPtiBnMZ5fIGbiQHQV3yTiAYqswJbHcYIb7YUT0HFGDDfUTC4E2Lkhff82Em8boNgR7020RXMrjhE57ox9kjx33SI3A3iDsBmIPos-nh9v9MiwjchrLCMwqZpGdn39Sopv3_pfWN3cA4IxgxFwKPhrcy-vyBw5qb4ZKZZm3ucHoMs1FlD04RJMv3bv0IpLfrT6BPnNI8x5D6Pkzbio9chksW0xLxyIhe_vIu-iyRycqsWKAM3vRPZImZFm7tdCxu2JCHbkTR84C-DB</Message></Error><Errors><RequestID>dfba5e62-f226-4251-a3ab-c2bbbf975603</RequestID></Response>"}

PLAY RECAP *****
100.25.31.2 : ok=0    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0

root@ubuntu:/etc/ansible#
```

Após a exclusão da instância que tínhamos na nossa conta do AWS agora criaremos um playbook para criar uma instância, e dentro dela já por a nossa aplicação dentro de um dockers.

```
root@ubuntu: /etc/ansible/wordpress-etapa9
File Edit View Search Terminal Help
hugoflorencio@ubuntu:~$ sudo su
root@ubuntu:/home/hugoflorencio# ls
Desktop  ec2.py      nano.save  Public    wordpress-ansible
Documents examples.desktop  Pictures  Templates  wordpress-ansible-docker
Downloads Music      projeto    Videos
root@ubuntu:/home/hugoflorencio# cd /etc/ansible
root@ubuntu:/etc/ansible# ls
ansible.cfg  ec2.ini  ec2.py  exclui-hosts.yml  hosts  hosts.save  roles
root@ubuntu:/etc/ansible# git clone https://github.com/fcamposchaves/wordpress-e
tapa9.git
Cloning into 'wordpress-etapa9'...
remote: Enumerating objects: 79, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 79 (delta 9), reused 71 (delta 4), pack-reused 0
Unpacking objects: 100% (79/79), done.
root@ubuntu:/etc/ansible# ls
ansible.cfg  ec2.py      hosts      roles
ec2.ini      exclui-hosts.yml  hosts.save  wordpress-etapa9
root@ubuntu:/etc/ansible# cd wordpress-etapa9
root@ubuntu:/etc/ansible/wordpress-etapa9# ls
ec2.ini  ec2.py  exclui-hosts.yml  group_vars  id_rsa  playbook.yml  roles
root@ubuntu:/etc/ansible/wordpress-etapa9#
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
tapa9.git
Cloning into 'wordpress-etapa9'...
remote: Enumerating objects: 79, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 79 (delta 9), reused 71 (delta 4), pack-reused 0
Unpacking objects: 100% (79/79), done.
root@ubuntu:/etc/ansible# ls
ansible.cfg  ec2.py      hosts      roles
ec2.ini      exclui-hosts.yml  hosts.save  wordpress-etapa9
root@ubuntu:/etc/ansible# cd wordpress-etapa9
root@ubuntu:/etc/ansible/wordpress-etapa9# ls
ec2.ini  ec2.py  exclui-hosts.yml  group_vars  id_rsa  playbook.yml  roles
root@ubuntu:/etc/ansible/wordpress-etapa9# rm -rf ec2.* exclui-hosts.yml
root@ubuntu:/etc/ansible/wordpress-etapa9# ls
group_vars  id_rsa  playbook.yml  roles
root@ubuntu:/etc/ansible/wordpress-etapa9# mv * ../
root@ubuntu:/etc/ansible/wordpress-etapa9# ls
root@ubuntu:/etc/ansible/wordpress-etapa9# cd ..
root@ubuntu:/etc/ansible# rm -rf wordpress-etapa9/
root@ubuntu:/etc/ansible# ls
ansible.cfg  ec2.py      group_vars  hosts.save  playbook.yml
ec2.ini      exclui-hosts.yml  hosts      id_rsa      roles
root@ubuntu:/etc/ansible#
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
GNU nano 2.9.3           ansible.cfg           Modified
# config file for ansible -- https://ansible.com/
# =====

# nearly all parameters can be overridden in ansible-playbook
# or with command line flags. ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[defaults]

# some basic default values...
inventory      = inventory/
#inventory     = /etc/ansible/hosts
library        = library/
#library       = /usr/share/my_modules/
#module_utils  = /usr/share/my_module_utils/
#remote_tmp    = ~/.ansible/tmp
#local_tmp     = ~/.ansible/tmp

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Spell  ^  Go To Line
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
GNU nano 2.9.3           ansible.cfg           Modified
# What flags to pass to sudo
# WARNING: leaving out the defaults might create unexpected behaviours
#sudo_flags = -H -S -n

# SSH timeout
#timeout = 10

# default user to use for playbooks if user is not specified
# (/usr/bin/ansible will use current user as default)
remote_user = ubuntu
#remote_user = root

# logging is off by default unless this path is defined
# if so defined, consider logrotate
log_path = ~/ansible.log
#log_path = /var/log/ansible.log

# default module name for /usr/bin/ansible
#module_name = command

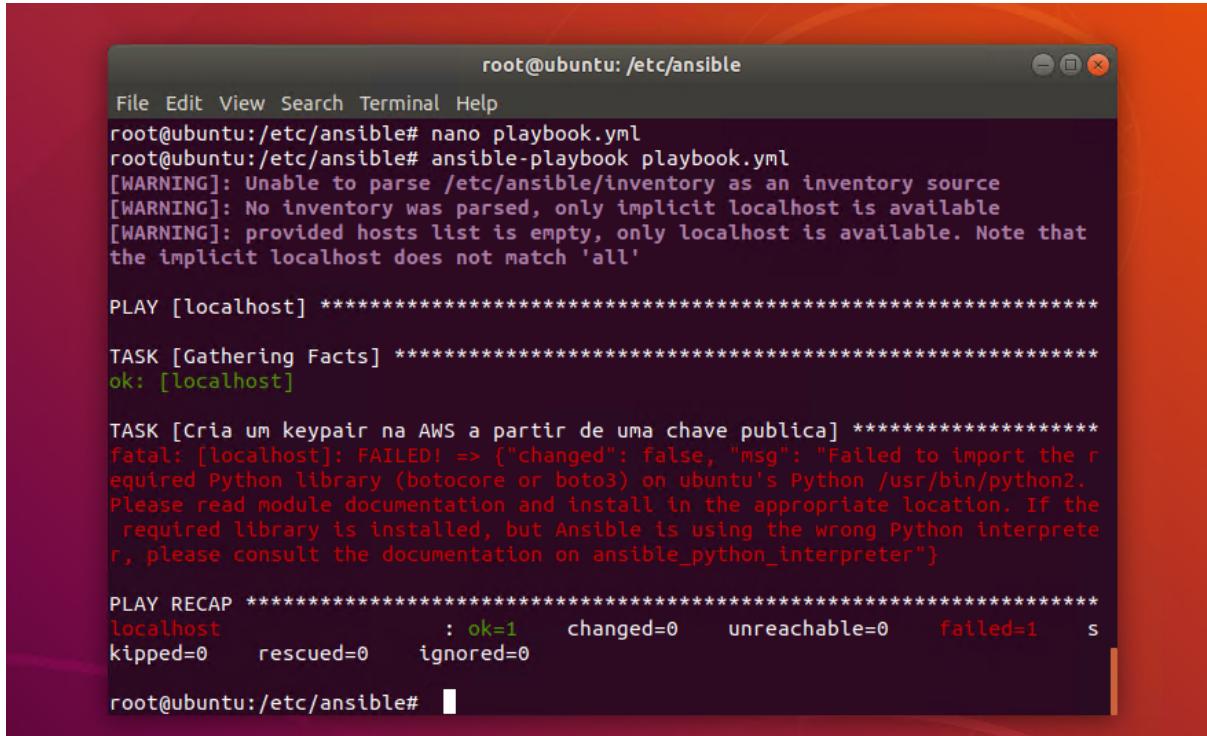
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Spell  ^  Go To Line
```

```
root@ubuntu: /etc/ansible/group_vars
File Edit View Search Terminal Help
GNU nano 2.9.3                               all                               Modified
---
aws_access_key: "AKIAYT6EIMRF2CVOY2UB"
aws_secret_key: "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"

application: "wordpress"
target_env: "prod"
aws_region: "us-east-1"
aws_region_short: "ue1"
aws_region_az: "c"
vpc_name: "vpc-{{ aws_region_short }}-{{ target_env }}-{{ application }}"
vpc_cidr_block: "172.27.0.0/16"
public_subnet_1_cidr: "172.27.0.0/18"
public_subnet_2_cidr: "172.27.64.0/18"
private_subnet_1_cidr: "172.27.128.0/18"
private_subnet_2_cidr: "172.27.192.0/18"
keypair_name: hugoflorencio-key

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell ^L Go To Line
```

```
root@ubuntu: /etc/ansible
File Edit View Search Terminal Help
bash: cd: /root/.ssh/id_rsa.pub: No such file or directory
root@ubuntu:/etc/ansible/group_vars# cd ..
root@ubuntu:/etc/ansible# cd ~/ssh/id_rsa.pub
bash: cd: /root/.ssh/id_rsa.pub: No such file or directory
root@ubuntu:/etc/ansible# ls
ansible.cfg  ec2.py      group_vars  hosts.save  playbook.yml
ec2.ini      exclui-hosts.yml  hosts      id_rsa      roles
root@ubuntu:/etc/ansible# mv id_rsa id_rsa.old
root@ubuntu:/etc/ansible# cp /home/hugoflorencio/.ssh/
cp: missing destination file operand after '/home/hugoflorencio/.ssh/'
Try 'cp --help' for more information.
root@ubuntu:/etc/ansible# cp /home/hugoflorencio/.ssh
cp: missing destination file operand after '/home/hugoflorencio/.ssh'
Try 'cp --help' for more information.
root@ubuntu:/etc/ansible# cp /home/hugoflorencio/.ssh/
cp: missing destination file operand after '/home/hugoflorencio/.ssh'
Try 'cp --help' for more information.
root@ubuntu:/etc/ansible# nano all
root@ubuntu:/etc/ansible# cd group_vars
root@ubuntu:/etc/ansible/group_vars# nano all
root@ubuntu:/etc/ansible/group_vars# cd ..
root@ubuntu:/etc/ansible# cd ~/ssh
bash: cd: /root/.ssh: No such file or directory
root@ubuntu:/etc/ansible# ssh-keygen -o
```



The screenshot shows a terminal window titled "root@ubuntu: /etc/ansible". The terminal output is as follows:

```
File Edit View Search Terminal Help
root@ubuntu:/etc/ansible# nano playbook.yml
root@ubuntu:/etc/ansible# ansible-playbook playbook.yml
[WARNING]: Unable to parse /etc/ansible/inventory as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that
the implicit localhost does not match 'all'

PLAY [localhost] ****
TASK [Gathering Facts] ****
ok: [localhost]

TASK [Cria um keypair na AWS a partir de uma chave publica] ****
fatal: [localhost]: FAILED! => {"changed": false, "msg": "Failed to import the required Python library (botocore or boto3) on ubuntu's Python /usr/bin/python2. Please read module documentation and install in the appropriate location. If the required library is installed, but Ansible is using the wrong Python interpreter, please consult the documentation on ansible_python_interpreter"}
```

PLAY RECAP ****

localhost	ok=1	changed=0	unreachable=0	failed=1	skipped=0	rescued=0	ignored=0

```
root@ubuntu:/etc/ansible#
```

8. Meu repositório no github:

<https://github.com/florenciohugo/woordpress-AWS.git>

9. Bibliografia :

<https://calculator.s3.amazonaws.com/>
<https://lms.infnet.edu.br/moodle>
<https://aws.amazon.com/pt/ec2/faqs/>
<https://aws.amazon.com/pt/route53/>
<https://github.com/fabiocchavesinfnet>
<https://github.com/fcamposchaves>

