



Infraestrutura II

Atividade obrigatória e individual

Ansible Exercício

O objetivo é criar um par de chaves associadas para conectar à instância; e, em seguida, uma instância EC2 com IP público associado às chaves criadas anteriormente.

Serviços

Por meio da lista de serviços fornecidos pela documentação do Ansible, encontre os módulos da AWS de que você precisa para cumprir o objetivo do seu exercício. [Link](#)

Na próxima página você encontrará a resolução. Continue o exercício como apenas autoavaliação.

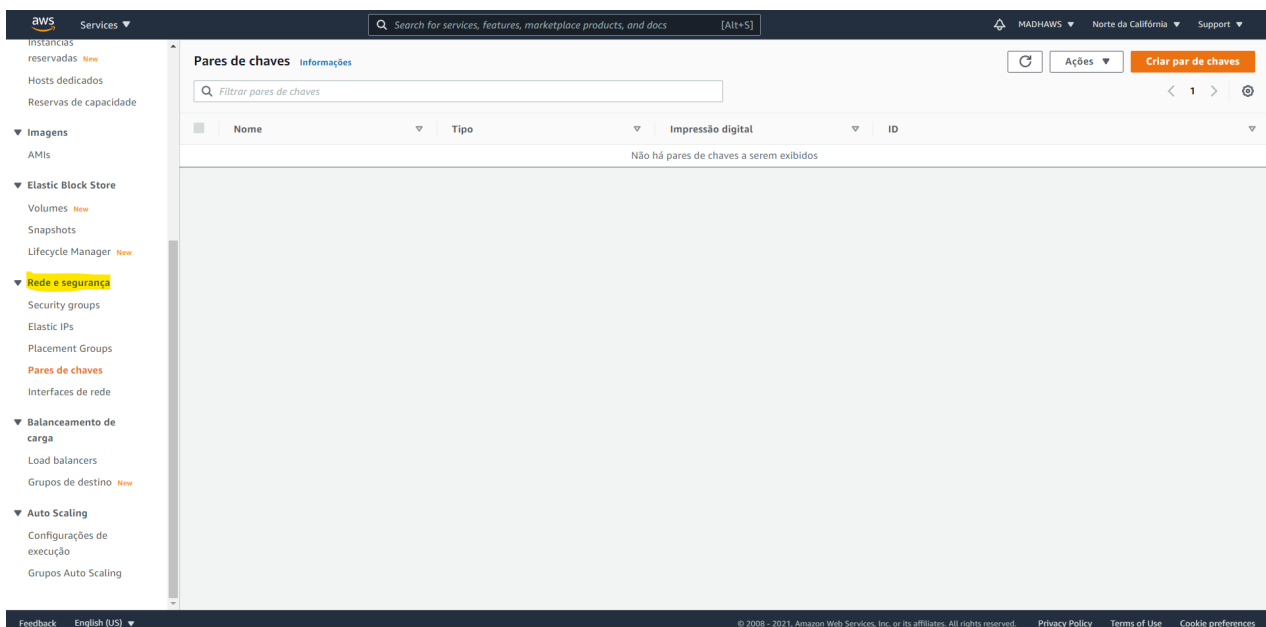
Resolução

Criando nossos Pares de chaves:

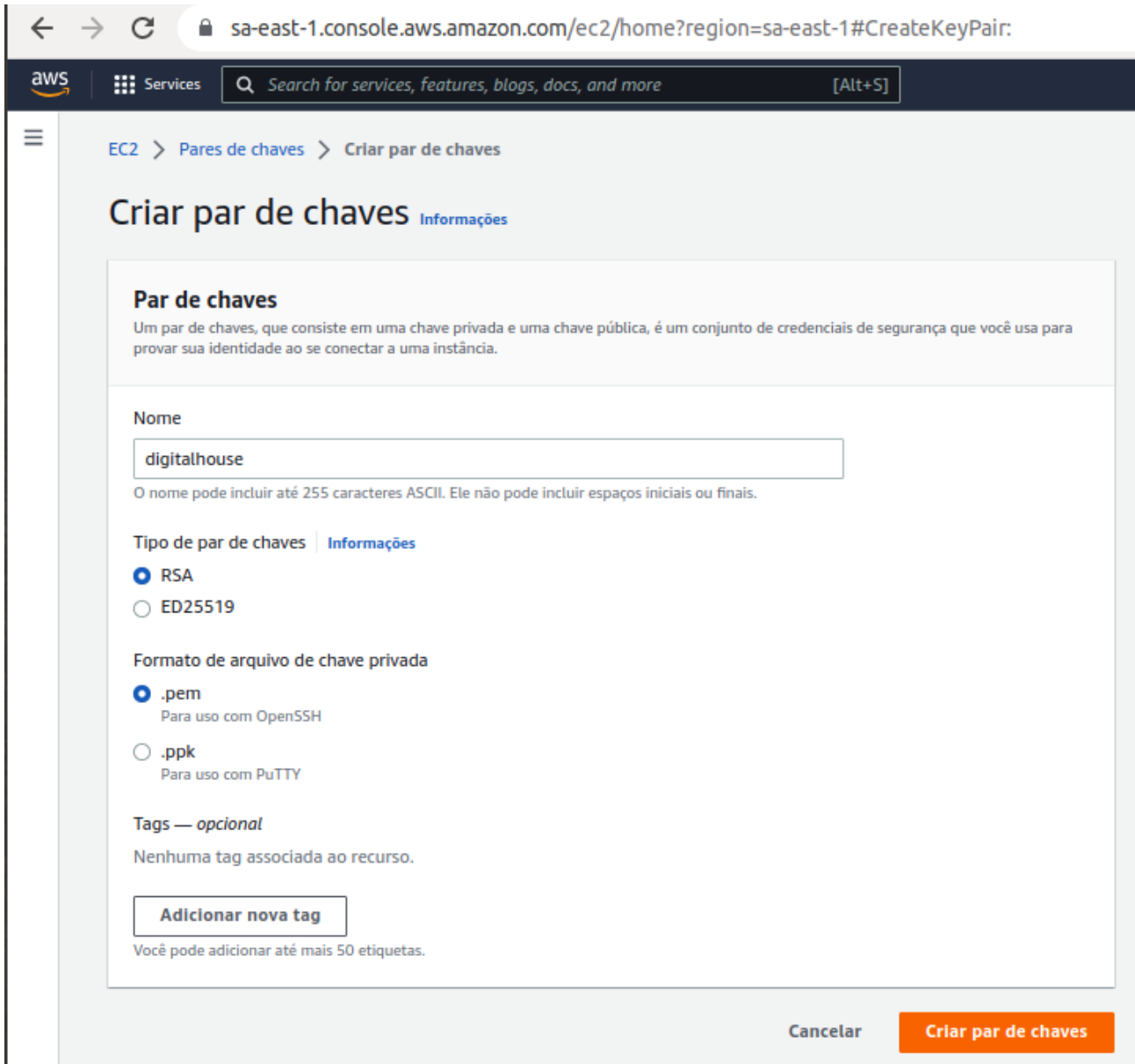
Antes de começar, crie um par de chaves na AWS para associar às suas instâncias, para que você possa se conectar posteriormente, uma vez que a execução de sua infraestrutura por código for concluída.

Você poderá criar o recurso necessário da seguinte maneira:

- Entrar no serviço EC2.
- Em seguida, vá para a seção "Rede e segurança" no menu à esquerda.
- Clique em "Pares de chaves".

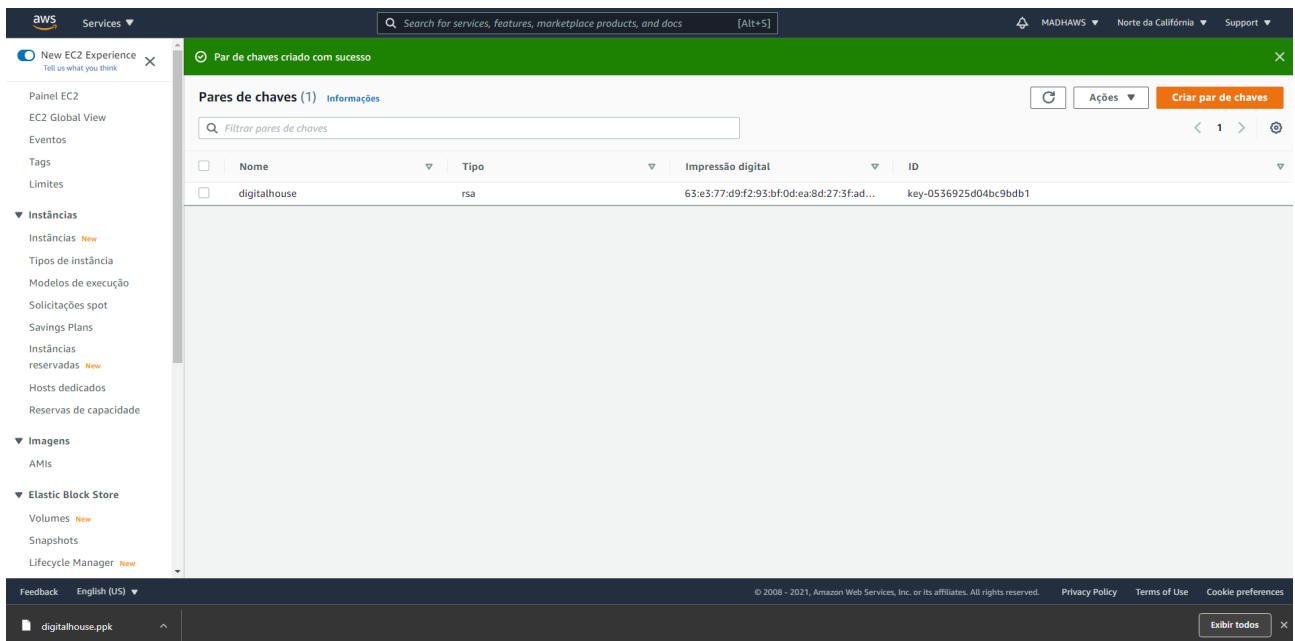


Em seguida, preencha as seguintes informações conforme aparecem na imagem a seguir.



The screenshot shows the AWS Management Console interface for creating a new key pair. The breadcrumb navigation indicates the path: EC2 > Pares de chaves > Criar par de chaves. The main heading is 'Criar par de chaves' with a link to 'Informações'. Below this, a section titled 'Par de chaves' explains that a key pair consists of a private key and a public key, used for authentication. The form includes a 'Nome' field with the value 'digitalhouse' and a note that the name can be up to 255 ASCII characters. The 'Tipo de par de chaves' section has two radio buttons: 'RSA' (selected) and 'ED25519'. The 'Formato de arquivo de chave privada' section has two radio buttons: '.pem' (selected, for OpenSSH) and '.ppk' (for PuTTY). There is a 'Tags' section with the note 'Nenhuma tag associada ao recurso.' and a button 'Adicionar nova tag' with a note that up to 50 tags can be added. At the bottom right, there are two buttons: 'Cancelar' and 'Criar par de chaves'.

Obs: Em nome, digite <seu-nome>-digitalhouse, ex: bgsouza-digitalhouse



Uma vez que essas opções são selecionadas, o arquivo "<seu-nome>-digitalhouse.pem" é baixado para o seu computador. Também teremos que obter o ID de nosso VPC padrão (ou criar um novo) dentro do serviço AWS "VPC".

Vamos agora gerar nossos tokens:

Vamos agora, gerar nosso Key/Secret que irá nos possibilitar conectar com a AWS.

Para isso vá no console da aws > IAM > selecione o seu usuário e vá na aba "Credenciais de Segurança"

aws Services [Alt+S]

Identity and Access Management (IAM)


Painel

- Gerenciamento de acesso
 - Grupos de usuários
 - Usuários**
 - Funções
 - Políticas
 - Provedores de identidade
 - Configurações de conta
- Relatórios de acesso
 - Analizador de acesso
 - Regras de arquivamento
 - Analísadores
 - Configurações
 - Relatório da credencial
 - Atividade da organização
 - Políticas de controle de serviço (SCPs)

ID da conta da AWS:
405378853534

Usuários > bruno

Resumo


ARN do usuário: `arn:aws:iam::405378853534:user/bruno` 

Caminho: `/`

Hora de criação: 2022-09-12 18:34 UTC-0300

Permissões Grupos (5) Tags **Credenciais de segurança** Consultor de acesso


Credenciais de login

	Resumo
Link de login do console	https://405378853534.signin.aws.amazon.com/console
Senha do console	Habilitado (último login em Hoje) Gerenciar
Dispositivo MFA atribuído	Não atribuído Gerenciar
Certificados de assinatura	Nenhum 

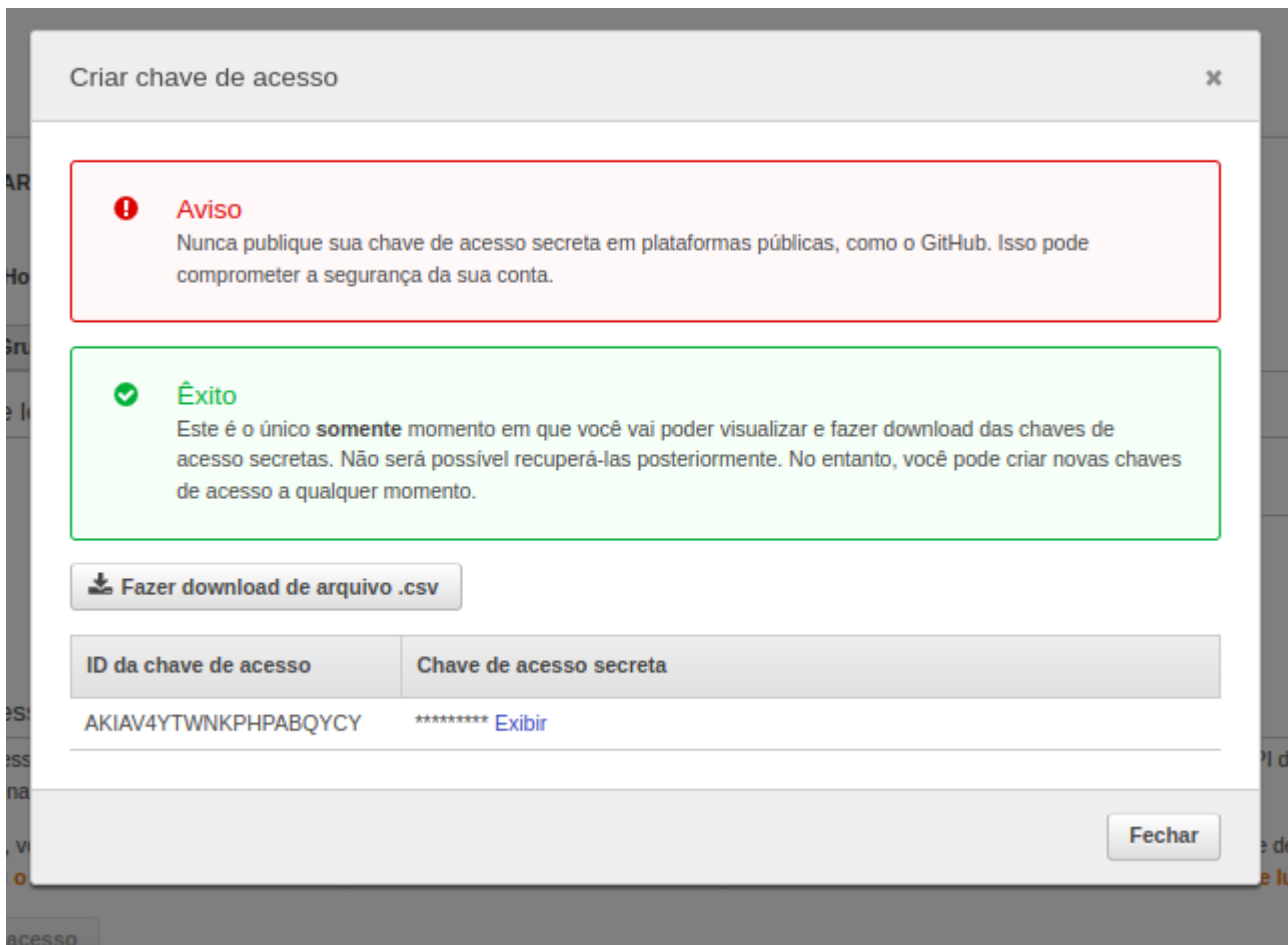
Chaves de acesso

Use chaves de acesso para fazer chamadas programáticas para a AWS a partir da CLI da AWS, Tools for PowerShell, SDKs da AWS ou chamadas de API (ativas ou inativas) por vez.

Para sua proteção, você nunca deve compartilhar suas chaves secretas com ninguém. Como prática recomendada, recomendamos alternar a visualização ou o download da chave secreta só podem ser feitos no momento da criação. Crie uma nova chave de acesso se você precisar.

 **Chave de acesso AKIAV4YTWNKPENHITN5L excluída**

Clique em “Criar chave de acesso”



Faça Download e guarde-o

Construindo nosso Playbook:

Agora, vamos aplicar nosso manual. O código por peça é o seguinte. Primeiro temos que apontar nossos hosts, no nosso caso, vamos apontar para "localhost", já que ele não se conecta a outro servidor para criar a instância.

```
---
- hosts: localhost
  gather_facts: yes
```

Criando as variáveis:

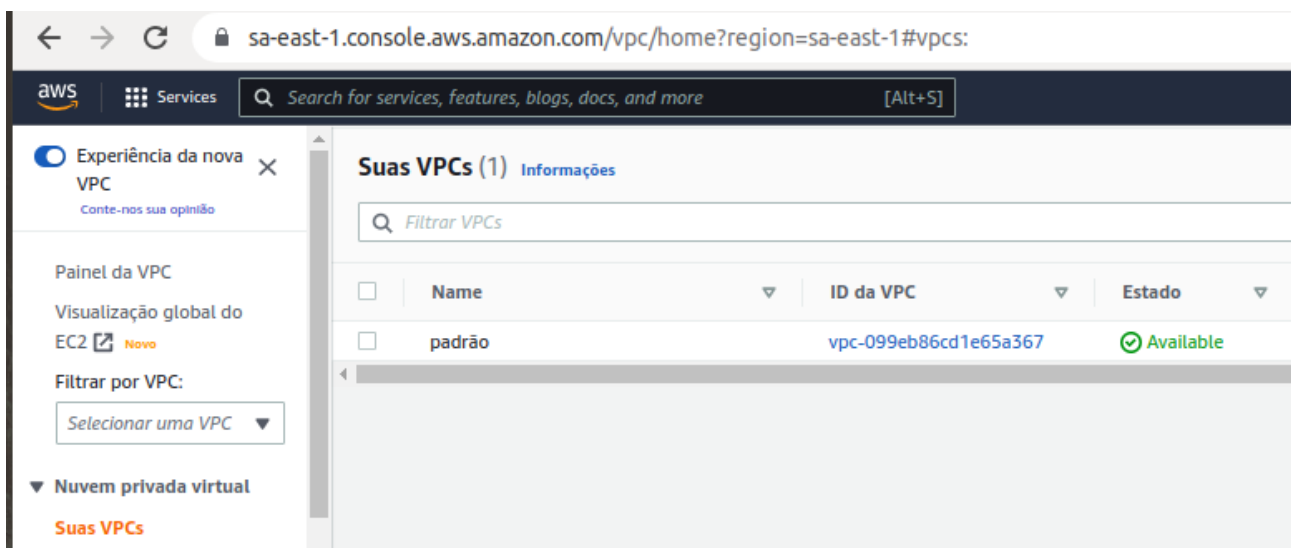
```
vars:
  keipair_name: <seu-nome>-digitalhouse
  my_vpc_id: <vpc-id>
  subnet_id: <subnet-id>
  my_ami_id: ami-04b3c23ec8efcc2d6
  ansible_python_interpreter: /usr/bin/python3
  AWS_access_key: "<esta-no-csv-baixado>"
  AWS_secret_key: "<esta-no-csv-baixado>"
```

DICA:

Para manter a segurança, não expor suas credenciais e ainda poder versionar de forma segura o seu playbook, de uma olhada no [Ansible Vault](#) e [também aqui](#)

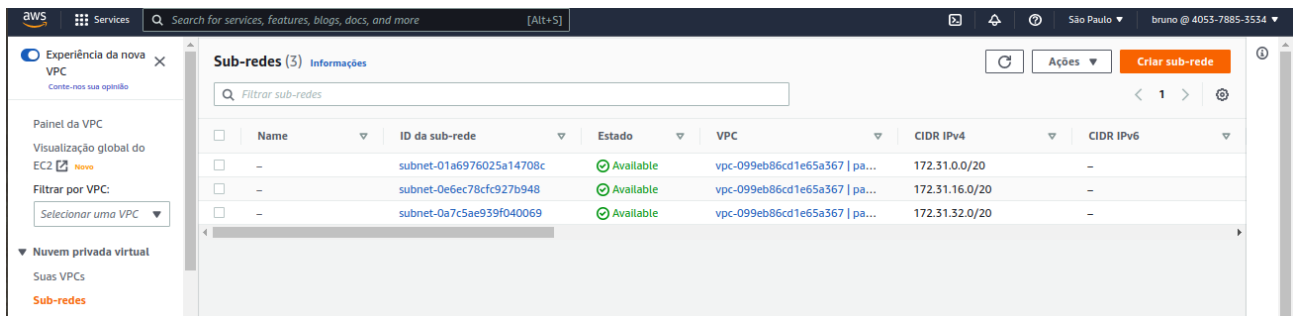
Obtendo o VPC ID

Console VPC > Suas VPCs



Obtendo o Subnet ID

Console VPC > Sub redes e escolha a primeira



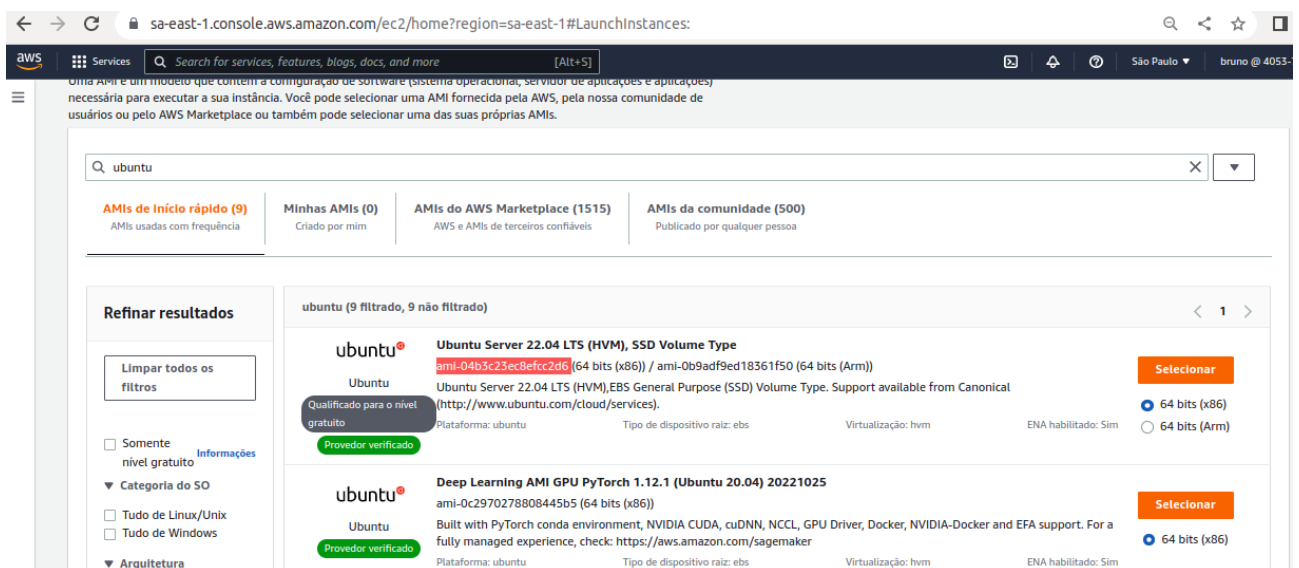
Obtendo o AMI ID

Acesse a url:

<https://sa-east-1.console.aws.amazon.com/ec2/home?region=sa-east-1#LaunchInstances>
es:

Filtre por: Ubuntu Server 22.04

Pegue o ID do x86



Definimos as tarefas:

```
tasks:
```


Vamos criar um grupo de segurança para poder acessar por SSH (porta 22) e se conectar com nosso par de chaves. A título de exemplo, também adicionamos as portas para HTTP e HTTPS, embora não façam parte do slogan:

```
- name: Criar grupo de segurança com HTTPS, HTTP e SSH
```

```
ec2_group:
  name: digitalhouse
  vpc_id: "{{ my_vpc_id }}"
  description: "sg com as regras"
  aws_access_key: "{{ AWS_access_key }}"
  aws_secret_key: "{{ AWS_secret_key }}"
  region: "sa-east-1"
  rules:
    - proto: tcp
      from_port: 22
      to_port: 22
      cidr_ip: 0.0.0.0/0
    - proto: tcp
      from_port: 80
      to_port: 80
      cidr_ip: 0.0.0.0/0
    - proto: tcp
      from_port: 443
      to_port: 443
      cidr_ip: 0.0.0.0/0
  rules_egress:
    - proto: all
      cidr_ip: 0.0.0.0/0
```

Criamos nossa tarefa, ou seja, a instância utilizando o módulo "ec2".

Em seguida, preenchemos as informações necessárias para criar a instância:

```
- name: Criamos nosso servidor
ec2:
    region: "sa-east-1"
    group: "digitalhouse"
    instance_type: "t2.micro"
    image: "{{ my_ami_id }}"
    aws_access_key: "{{ AWS_access_key }}"
    aws_secret_key: "{{ AWS_secret_key }}"
    wait: yes
    wait_timeout: 500
    volumes:
        - device_name: "/dev/xvda"
          volume_type: "gp2"
          volume_size: 8
    vpc_subnet_id: "{{ subnet_id }}"
    assign_public_ip: yes
    key_name: "{{ keipair_name }}"

register: info
```

Também registramos a saída em uma variável chamada "info". Em seguida, usamos o módulo "debug" para exibir essas informações no log de saída do Ansible:

```
- name: IP público do nosso servidor
  debug:
    var: info.instances[0].public_ip

- name: DNS público de nosso servidor
  debug:
    var: info.instances[0].public_dns_name
```

A instância foi criada com sucesso, isso pode ser evidenciado no console da AWS e nos logs do Ansible. Além disso, verificamos se podemos nos conectar por ssh corretamente de nossos computadores. Vamos notar como o que mostramos com "debug" o usamos

para nos conectar à nossa instância sem ter que entrar no AWS, só temos que saber o usuário padrão, que em um “Amazon Linux 2 AMI” é sempre “ec2-user”.

- Dependências antes de executar o playbook:
 - **pip3 install boto**

Execute o playbook

```
digitalhouse@user-infra $ ansible-playbook ec2.yml
```

Observe a saída

```
ansible-playbook ec2.yml

[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 [Criar grupo de segurança com HTTPS, HTTP e SSH]
*****

ok: [localhost]

TASK [Criamos nosso servidor]
*****
```



```
changed: [localhost]

TASK [IP público do nosso servidor]
*****

ok: [localhost] => {

    "info.instances[0].public_ip": "18.231.10.86"

}

TASK [DNS público de nosso servidor]
*****

ok: [localhost] => {

    "info.instances[0].public_dns_name": "ec2-18-231-10-86.sa-east-1.compute.amazonaws.com"

}

PLAY RECAP
*****
*****

localhost                : ok=5    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

Conecte na instância

```
digitalhouse@user-infra $ chmod 400 <seu-nome>-digitalhouse.pem

digitalhouse@user-infra $ ssh -i <seu-nome>-digitalhouse.pem
ubuntu@ec2-18-231-10-86.sa-east-1.compute.amazonaws.com

Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1019-aws x86_64)
```



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

System information as of Mon Oct 31 23:01:09 UTC 2022

System load:  0.0               Processes:            99
Usage of /:   19.7% of 7.57GB   Users logged in:     0
Memory usage: 20%              IPv4 address for eth0: 172.31.11.4
Swap usage:   0%

0 updates can be applied immediately.

The list of available updates is more than a week old.

To check for new updates run: sudo apt update

Last login: Mon Oct 31 22:56:55 2022 from 201.26.23.147

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-11-4:~$
$ ubuntu@ip-172-31-11-4: ~$
```

Observe na console AWS

ID de la instancia	Estado de la i... ▼	Tipo de inst... ▼	Comprobación ...	Estado de la ...	Zona de dispon... ▼	DNS de IPv4 pública
i-04dd886604a17385f	✓ En ejecución 🔍	t2.micro	✓ 2/2 comprobador	Sin alarmas +	us-west-1c	ec2-52-53-190-226.us

Gabarito

Nosso código completo é o seguinte:

```
---
- hosts: localhost
  gather_facts: yes

vars:
  keipair_name: <seu-nome>-digitalhouse
  my_vpc_id: vpc-099eb86cd1e65a367
  subnet_id: subnet-01a6976025a14708c
  my_ami_id: ami-04b3c23ec8efcc2d6
  ansible_python_interpreter: /usr/bin/python3
```



```
AWS_access_key: ""
AWS_secret_key: ""

tasks:
- name: Criar grupo de segurança com HTTPS, HTTP e SSH
  ec2_group:
    name: digitalhouse
    vpc_id: "{{ my_vpc_id }}"
    description: "sg com as regras"
    aws_access_key: "{{ AWS_access_key }}"
    aws_secret_key: "{{ AWS_secret_key }}"
    region: "sa-east-1"
    rules:
      - proto: tcp
        from_port: 22
        to_port: 22
        cidr_ip: 0.0.0.0/0
      - proto: tcp
        from_port: 80
        to_port: 80
        cidr_ip: 0.0.0.0/0
      - proto: tcp
        from_port: 443
        to_port: 443
        cidr_ip: 0.0.0.0/0
    rules_egress:
      - proto: all
        cidr_ip: 0.0.0.0/0

- name: Criamos nosso servidor
  ec2:
    region: "sa-east-1"
    group: "digitalhouse"
    instance_type: "t2.micro"
    image: "{{ my_ami_id }}"
```



```
aws_access_key: "{{ AWS_access_key }}"
aws_secret_key: "{{ AWS_secret_key }}"
wait: yes
wait_timeout: 500
volumes:
  - device_name: "/dev/xvda"
    volume_type: "gp2"
    volume_size: 8
vpc_subnet_id: "{{ subnet_id }}"
assign_public_ip: yes
key_name: "{{ keipair_name }}"
register: info

- name: IP público do nosso servidor
  debug:
    var: info.instances[0].public_ip

- name: DNS público de nosso servidor
  debug:
    var: info.instances[0].public_dns_name
```