📚 Relatório de Configuração de Instância EC2 na AWS

Este relatório detalha o processo de implantação da aplicação ServeRest em uma instância EC2 da Amazon Web Services (AWS) pela Squad Level UP. O objetivo principal desta atividade foi praticar a configuração e o uso da AWS EC2, bem como a implantação de aplicações em ambientes de nuvem. Este documento aborda o passo a passo seguido, os comandos executados e os desafios superados durante o processo.

Membros da Squad Level UP

Função	Nome
Lider	@Isa Reginabs
Membro	@Eric Lima da Silva
Membro	@Ivo Sobral dos Santos Junior
Membro	@João Gabriel Oliveira Magalhães
Membro	@Maycon Douglas Da Silva

Ferramentas Utilizadas

Ferramenta	Descrição
AWS EC2	Serviço de computação em nuvem para provisionamento da instância.
Node.js	Ambiente de execução JavaScript para o ServeRest.
npm/npx	Gerenciador de pacotes Node.js para instalação e execução do ServeRest.
Curl	Ferramenta de linha de comando para transferência de dados.
Yum	Gerenciador de pacotes para sistemas Linux.
Git	Sistema de controle de versão para o repositório do projeto.

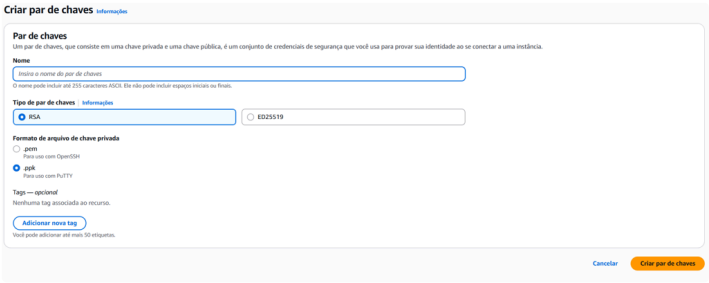
1. Criação e Organização da Pasta de Acesso

Antes de iniciar a configuração no Console AWS, foi criada **uma pasta local de fácil localização** para armazenar a chave de acesso (.pem) que será utilizada para autenticação na instância.

2. Criação do Par de Chaves

1. No **Console AWS**, acessou-se o campo de busca e digitou-se **EC2**, abrindo o **Dashboard EC2**.

2. No menu lateral, dentro do grupo **Rede e segurança**, foi selecionada a opção **Pares de chaves**.
3. Foi clicado em **Criar par de chaves**.



4. As configurações escolhidas para o par de chaves foram:
   - **Nome**: Definido de acordo com o projeto.
   - **Tipo de chave**: RSA
   - **Formato do arquivo**: .pem
5. Após a criação, o download da chave foi feito automaticamente.
6. O arquivo foi movido para a pasta criada anteriormente.

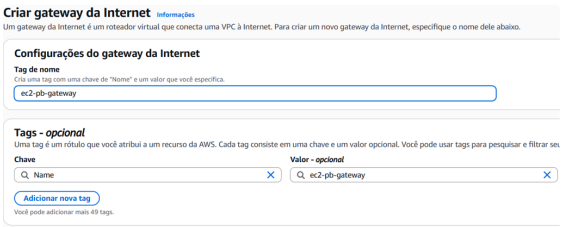
3. Configuração do Internet Gateway e da Rede

No console, foi utilizada a barra de pesquisa para localizar o recurso **Internet Gateway**.

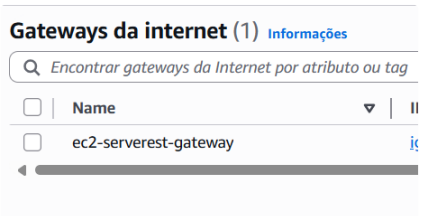
> **Observação:**
> *Antes de configurar o gateway, mudar o servidor para Norte da Virgínia us-east-1, para que futuramente não precise reconfigurar o gateway, já que ele fica vinculado ao servidor selecionado. No tutorial, no console já está definido us-east-1, mas só depois de configurado que é orientado mudar o servidor para us-east-1. Causando confusão e falha ao tentar conectar pelo terminal.*

3.1 Criação e Associação do Internet Gateway

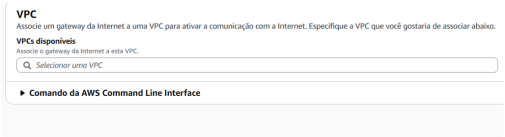
1. No menu lateral, foi acessada a opção **Gateways da Internet**.
2. Foi clicado em **Criar Gateway da Internet** e definido um nome para o recurso.



3. Após criado, o sistema redirecionou para a tela de detalhes do gateway.

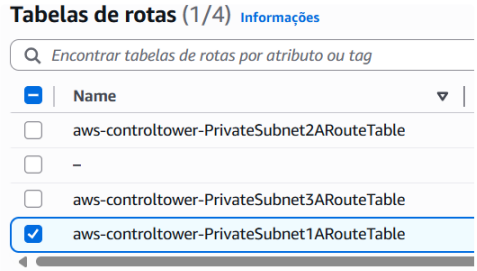


4. Clicou-se em **Ações > Associar à VPC**.
5. Na lista de VPCs disponíveis, foi selecionada a VPC já existente e clicado em **Associar Gateway da Internet**.



3.2 Configuração da Tabela de Rotas

1. Voltando ao menu lateral, foi selecionada **Tabelas de Rotas**.
2. Entre as tabelas disponíveis, foi escolhida a tabela **não marcada como principal** (a tabela principal já vem configurada por padrão).



3. Com a tabela selecionada, na aba **Rotas**, clicou-se em **Editar rotas**.
4. Foi adicionada uma nova rota:
   - **Destino:** 0.0.0.0/0
   - **Alvo:** Gateway da Internet criado anteriormente.
5. A configuração foi salva, liberando acesso à internet para a VPC associada.

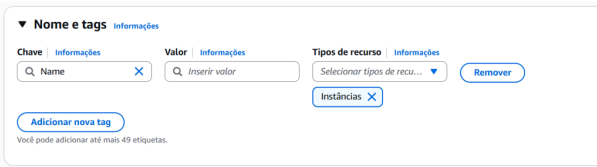
4. Criação da Instância EC2

Após a configuração de rede e segurança, iniciou-se a criação da instância EC2:

4.1. Nomeação e Tags

1. Voltando ao **Dashboard EC2**, clicou-se em **Executar Instâncias**.
2. Em **Nome e Tags**, foi adicionado:

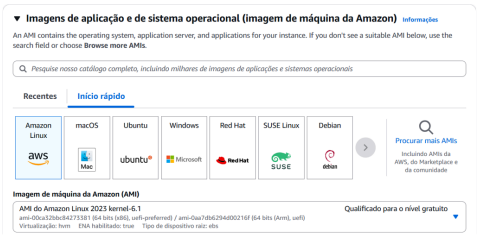
Chave	Valor	Tipo de
Name	Linux Serverest	Instâncias e Volumes
Project	Programa de Bolsas	Instâncias e Volumes
CostCenter	Quality Assurance	Instâncias e Volumes



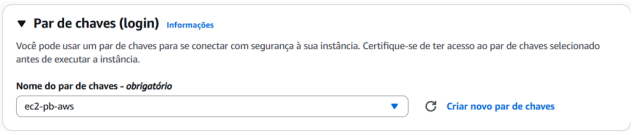
3. Apenas estas três tags foram criadas, conforme orientado pelo tutorial.

4.2. Configurações de Sistema Operacional e Instância

1. **Imagem do Sistema Operacional:** Amazon Linux (64 bits).

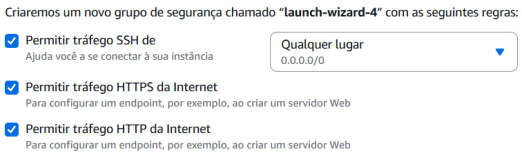


2. **Tipo de Instância:** t2.micro (padrão e elegível ao nível gratuito).
3. **Par de Chaves:** Selecionado o par de chaves criado anteriormente.



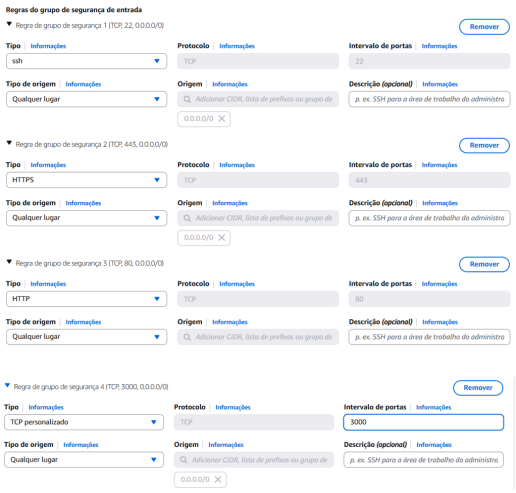
4.3. Configuração de Rede e Segurança

1. As regras de segurança foram configuradas para permitir:
   - **SSH:** Acesso de qualquer lugar.
   - **HTTPS:** Acesso da internet.
   - **HTTP:** Acesso da internet.



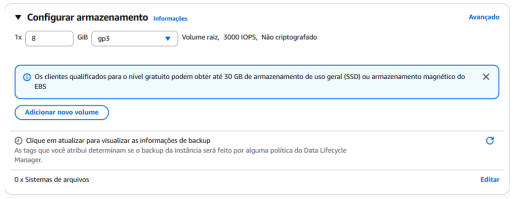
2. O **IP público** da instância foi habilitado (opção padrão é desativada).

3. Uma nova regra de grupo de segurança foi adicionada para permitir tráfego na porta **3000/TCP**, que será utilizada pelo serviço Serverest:
   - **Tipo:** TCP personalizado
   - **Protocolo:** TCP
   - **Intervalo de portas:** 3000
   - **Origem:** Qualquer lugar



4.4. Armazenamento

O armazenamento padrão foi mantido em **8 GiB, tipo gp3**, garantindo desempenho básico para a instância.



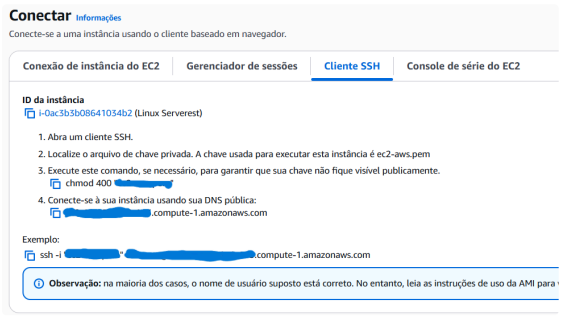
4.5. Execução da Instância

Após revisão das configurações, clicou-se em **Executar Instâncias**, concluindo com sucesso a criação da instância EC2.

5. Conexão à Instância EC2 via SSH

Após a configuração da instância, foi necessário acessar a máquina virtual na AWS via SSH para instalar dependências e configurar o ambiente.

1. **Acesso ao Painel de Conexão:**
   - No **Console da AWS**, selecionou-se a instância EC2 e clicou-se em **Conectar-se à instância**.
   - Na aba **Conexão de instância do EC2**, foi exibido o **IP público** da instância, que foi **anotado para uso posterior**.
   - Em seguida, acessou-se a aba **Cliente SSH**, onde são fornecidos os comandos de conexão.



2. **Acesso ao Diretório da Chave de Acesso:**
   - No terminal, acessou-se a pasta onde o arquivo `.pem` foi armazenado:

```
cd /home/usuario/pasta
```

   - Em seguida, listou-se o conteúdo da pasta para confirmar a presença da chave:

```
ls
```

A chave `ec2-pb-aws.pem` estava disponível no diretório.

3. **Ajuste de Permissões da Chave:**

   Para garantir a segurança do arquivo `.pem`, foi aplicado o comando sugerido:

```
chmod 400 ec2-pb-aws.pem
```

4. **Conexão via SSH:**

   Utilizando o IP público anotado, foi executado:

```
ssh -i "ec2-pb-aws.pem" ec2-user@<IP_PÚBLICO>
```

   Ao aparecer a mensagem:

   `Are you sure you want to continue connecting (yes/no)?`

   Foi digitado `yes` e pressionado `Enter`.

   A partir desse ponto, a sessão SSH com a instância EC2 foi estabelecida com sucesso.

6. Atualização e Instalação de Pacotes Essenciais

Com a conexão estabelecida, iniciou-se a configuração do ambiente:

1. **Atualização do Sistema:**

```
sudo yum update -y
```

   Esse comando garantiu que todos os pacotes da distribuição Amazon Linux estivessem atualizados.

2. **Instalação de Compiladores e Ferramentas:**

```
sudo yum install gcc-c++ make -y
```

3. **Verificação e Instalação do cURL:**
   - Verificação:

```
curl --version
```

   - Caso o `curl` não estivesse instalado, foi executado:

```
sudo yum install curl -y
```

7. Criação do Diretório de Projeto e Instalação do Node.js

1. **Criação da pasta para o Serverest:**

```
mkdir serverestApi
cd serverestApi
```

2. **Instalação do Node.js:**

   A instalação recomendada no tutorial com `curl` não funcionou. Utilizou-se a instalação direta com:

```
sudo yum install -y nodejs
```

8. Inicialização do Serverest

Para instalar e executar a API Serverest diretamente:

```
npx serverest@latest
```

Ao iniciar o Serverest, a API ficou disponível na **porta 3000** da instância EC2.

Com isso, foi possível acessar a interface Swagger do Serverest diretamente pelo navegador utilizando:

```
http://<IP_PÚBLICO>:3000
```

Substituindo `<IP_PÚBLICO>` pelo endereço anotado anteriormente.

A página do Swagger confirmou que o **Serverest** estava em execução e acessível pela internet.

9. Desafios:

- **Desafio:** Um dos desafios encontrado pelos integrantes do squad é que há mudanças, mesmo que sutis, entre a plataforma da AWS e as vídeos aulas, por exemplo:
  - Na vídeo aula, no momento da criação do gateway da internet, mostra que não há nenhuma VPC associado a outro gateway, mas na plataforma o VPC está associado a outro gateway automaticamente criado. Isso causou um pouco de confusão.

- **Desafio:** Outro desafio que deve ser citado é a ocorrência de erro, especificamente o erro timeout, que poderia ter sido evitado se na vídeo aula fosse abordado primeiramente que a região deveria ser alterada para Norte-Virgina us-east-1, pois essa informação foi falada em um momento que o gateway de internet já tinha sido criado e associado a uma vpc em outra região, causando assim o erro.

- **Desafio:** Um dos desafios aconteceu quando foi acessado o link disponibilizado e redirecionado para uma página relativamente diferente do tutorial. Enquanto o instrutor falava para clicarmos em ***management console***, essa opção não existia na plataforma atual.
  - **Resolução:** Foi realizado uma reunião com outro integrante e dessa forma descobrimos que a opção correta para seguir com os passo a passo era **AdministratorAccess**.

Imagem do tutorial

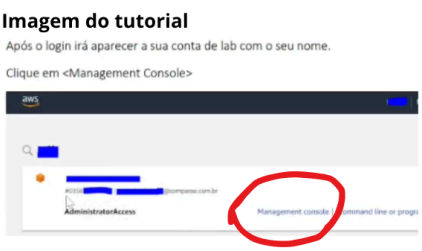
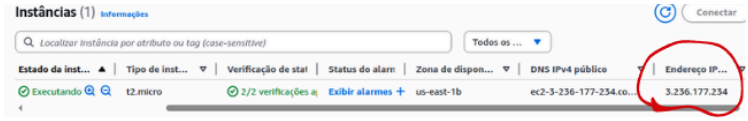
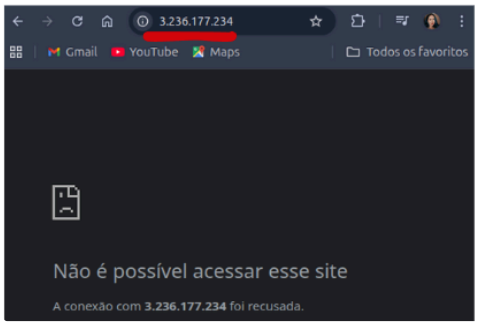
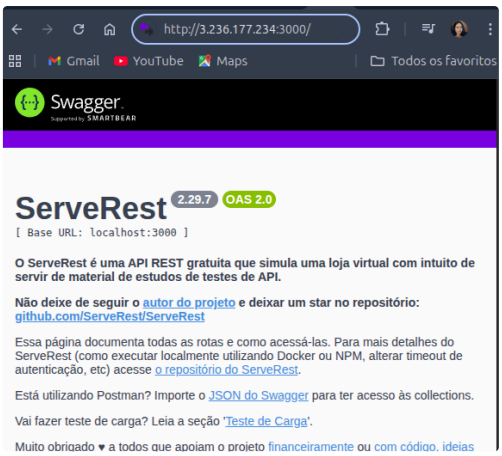


Imagem da plataforma atual

- **Desafio:** No momento de copiar o IP publica (conforme o instrutor nos ensinou) para a URL, gerou erro.
  - **Resolução**: Mesmo que essa instrução de copiar o IP publico tenha funcionado com outros colegas da squad, não foi o caso de outros. Desse modo, foi necessário especificar a rota assim: http://3.236.177.234:3000/


Instâncias (1) Informações
Conectar
Localizar instância por atributo ou tag (case-sensitive)
Todos os ...
Estado da inst...
Tipo de inst...
Verificação de stat
Status do alarm
Zona de dispon...
DNS IPv4 público
Endereço IP...
Executando
t2.micro
2/2 verificações a
Exibir alarmes
us-east-1b
ec2-3-236-177-234.co...
3.236.177.234


3.236.177.234
Gmail
YouTube
Maps
Todos os favoritos
Não é possível acessar esse site
A conexão com 3.236.177.234 foi recusada.


http://3.236.177.234:3000/
Gmail
YouTube
Maps
Todos os favoritos
Swagger
ServeRest
2.29.7
OAS 2.0
[Base URL: localhost:3000]
O ServeRest é uma API REST gratuita que simula uma loja virtual com intuito de servir de material de estudos de testes de API.
Não deixe de seguir o autor do projeto e deixar um star no repositório:
github.com/ServeRest/ServeRest
Essa página documenta todas as rotas e como acessá-las. Para mais detalhes do ServeRest (como executar localmente utilizando Docker ou NPM, alterar timeout de autenticação, etc) acesse o repositório do ServeRest.
Está utilizando Postman? Importe o JSON do Swagger para ter acesso às collections.
Vai fazer teste de carga? Leia a seção 'Teste de Carga'.
Muito obrigado ♥ a todos que apoiam o projeto financeiramente ou com código, ideias