Relatório de Configuração AWS e ServeRest

Introdução

Este documento apresenta um passo a passo detalhado para a criação, configuração e execução da aplicação ServeRest em uma instância EC2 na AWS, utilizando o ambiente da Academy Compass. O objetivo é demonstrar o processo completo, desde o acesso ao console da AWS, configuração de chaves e recursos de rede, até a instalação do ambiente Node.js e a execução da aplicação para validação via navegador e ferramentas como o Postman.

As instruções são voltadas para usuários com acesso corporativo ao ambiente da AWS Academy Compass, e seguem práticas recomendadas de segurança e organização, como uso de chaves temporárias, configuração adequada de permissões de rede e separação de arquivos em diretórios locais específicos.

---

1. Acesso ao Console AWS

Acesso via Academy Compass

O acesso ao ambiente da AWS é realizado utilizando as credenciais fornecidas com o domínio `@compasso.com.br`, conforme o passo a passo abaixo:

1. Acesse a URL:
   https://academy-compass.awsapps.com/start#/
2. Será solicitado o seu usuário corporativo (AD):
   `seu_nome@compasso.com.br`
3. Após o login, você verá a sua **conta de lab** identificada com seu nome.
4. Clique em **<Management Console>** para ser redirecionado ao Console de Gerenciamento da AWS.

Acesso Programático (via CLI)

Para utilizar o acesso programático (linha de comando ou via SDK/API), **não devemos criar usuários IAM** manualmente. Em vez disso:

1. Na mesma tela onde está sua conta de lab, clique em:
   <Command line or programmatic access>
2. Siga as instruções exibidas para obter suas **Access Key ID** e **Secret Access Key** temporárias.

Essas chaves podem ser configuradas com o comando:

```
aws configure
```

Insira os dados quando solicitado:

- Access Key ID
- Secret Access Key
- Região (ex: `us-east-1`)
- Formato de saída (ex: `json`)

> ❗ *Importante:* Estas chaves são temporárias e devem ser utilizadas com responsabilidade apenas nos ambientes permitidos.

2. Criação da Pasta e Par de Chaves

- Criamos localmente uma pasta chamada `EC2-AWS` para organização dos arquivos relacionados.
- No Console AWS:
  - Iremos até o serviço **EC2** (caso não estivesse visível, buscamos por “EC2”).
  - No menu lateral esquerdo, clicamos em **"Rede e segurança" > "Pares de chaves"**.
  - Clicamos em **"Criar par de chaves"**.
  - Nome: `ec2.pb-aws`
  - Formato: **PEM**
  - Tipo de chave: **RSA**
  - Após a criação, o arquivo `.pem` foi baixado automaticamente e movido para a pasta `EC2-AWS`.

---

3. Criação e Associação de Internet Gateway

- Buscamos por **“Internet Gateway”** na barra de busca.
- Antes de criar, navegamos para **"Suas VPCs" > "Gateways da Internet"**.
- Clicamos em **"Criar Gateway da Internet"** e definimos:
  - Nome: `ec2-serverest-gateway`
- Após a confirmação da criação, clicamos no pop-up para **associar à VPC**.
- Selecionamos a VPC disponível e clicamos em **"Associar"**.
- Em seguida, navegamos até **"Tabelas de rotas"**.
- Selecionamos uma das tabelas disponíveis, clicamos em **"Editar rotas"**.
- Adicionamos uma nova rota:
  - Destino: `0.0.0.0/0`
  - Destino (Gateway): Selecionamos o ID do Internet Gateway recém-criado.
- Salvamos as alterações.
- Repetimos o processo para outras tabelas de rotas disponíveis, se necessário.

---

4. Criação da Instância EC2

- Voltamos ao painel principal do EC2 e clicamos em **"Executar Instância"**.
- Configuramos a instância com os seguintes parâmetros:

Nome e Tags

- Nome da instância: `Linux Serverest`
- Tags obrigatórias:
  - `Name`: `Linux Serverest`
  - `Project`: `Programa de Bolsas`
  - `CostCenter`: `quality assurance`
- Tipo de recurso das tags: **Instâncias e Volumes**

Imagem, Tipo e Par de Chaves

- AMI: **Amazon Linux 2**
- Arquitetura: **x86_64 (64 bits)**
- Tipo de instância: **t2.micro**
- Par de Chaves: **Selecionamos** `ec2.pb-aws` **previamente criado**

Configurações de Rede e Segurança

- Permissões de acesso:

    - ✔️ SSH (porta 22)
    - ✔️ HTTP (porta 80)
    - ✔️ HTTPS (porta 443)
- Adicionamos uma nova regra:
    - Tipo: **Personalizado TCP**
    - Porta: **3000**
    - Origem: **Qualquer lugar (0.0.0.0/0)**

Armazenamento

- Volume padrão: **8 GB gp3**

Após revisar tudo, clicamos em **"Executar instância"**.

A instância foi criada com sucesso e iniciada automaticamente

---

5. Conexão com a Instância EC2 via SSH

Após o lançamento da instância EC2, é necessário conectar-se a ela via SSH para continuar com a instalação da aplicação.

Conectando ao AWS EC2:

1. No Console da AWS, vá até **Instâncias** no painel do EC2.
2. Selecione sua instância e clique no botão **“Conectar”**.
3. Você será redirecionado para uma nova página com as instruções de conexão.
4. Copie o **Endereço IP público** exibido — será usado para conectar via SSH.

Acesso via Cliente SSH:

1. Clique na aba **Cliente SSH** no topo da página.
2. Certifique-se de que você esteja no diretório onde salvou o arquivo da chave privada `ec2-pb-aws.pem`.
   Recomendação: salve-o dentro da pasta `EC2-AWS`.
3. No terminal, execute o seguinte comando para definir a permissão correta da chave:

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

4. Em seguida, copie o **comando de exemplo** exibido na página de conexão da AWS. Geralmente terá o seguinte formato:

```
ssh -i "ec2-pb-aws.pem" ec2-user@<IP-PÚBLICO-DA-EC2>
```

5. Cole o comando no terminal e pressione **Enter**.
6. Quando solicitado, digite `yes` para confirmar a primeira conexão com o host.
7. Após isso, a conexão SSH com sua instância será estabelecida com sucesso.

---

6. Preparando o Ambiente na Instância EC2

Atualização de Pacotes

Execute o comando para atualizar os pacotes da instância:

```
sudo yum update -y
```

Instalação de utilitários

Instale pacotes necessários para o processo:

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

Verificação do `curl`

Verifique se o `curl` já está instalado:

```
curl --version
```

Se não estiver, instale com:

```
sudo yum install curl
```

---

7. Instalação do Node.js

1. Crie uma pasta para a aplicação:

```
mkdir serverestApi
cd serverestApi
```

2. Execute o script para preparar o ambiente com a fonte de instalação do Node.js:

```
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash - && sudo yum install -y nodejs
```

3. Caso o comando acima retorne erro relacionado ao `apt-get`, execute este comando isolado:

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

4. Verifique se a instalação foi bem-sucedida:

```
node -v
npm -v
```

---

8. Executando o ServeRest na EC2

1. Execute o seguinte comando para instalar e iniciar o ServeRest:

```
npx serverest@latest
```

2. Aguarde o carregamento da aplicação. No final, será exibida a mensagem:

```
Teste o funcionamento acessando http://localhost:3000/usuarios
```

Atenção: como estamos executando em uma máquina remota, você deve substituir `localhost` pelo IP público da instância EC2.

---

9. Validação do Funcionamento da API

1. Acesse em seu navegador o endereço:

```
http://<IP-PÚBLICO-DA-EC2>:3000
```

Você verá a interface Swagger da ServeRest.

2. Para validar as rotas, utilize o Postman e envie requisições para endpoints como:

```
GET http://<IP-PÚBLICO-DA-EC2>:3000/usuarios
```

3. Verifique a resposta para garantir que a aplicação esteja rodando corretamente.

A partir desse momento, o ServeRest está funcionando com sucesso em sua instância EC2 na AWS.

Conclusão

Durante a execução deste processo, foi possível configurar com sucesso uma instância EC2 na AWS utilizando o ambiente disponibilizado pela Academy Compass. Todas as etapas foram realizadas com base em boas práticas, desde o acesso programático com chaves temporárias, configuração de rede com Internet Gateway, permissões via grupos de segurança, até a instalação do ambiente Node.js e execução da aplicação ServeRest.

A aplicação foi devidamente testada e validada por meio de interface gráfica (Swagger) e testes de endpoints via Postman. O resultado final confirma que o ambiente está funcional, acessível externamente e pronto para utilização em treinamentos, testes e simulações de APIs RESTful.

Este relatório pode ser utilizado como referência prática para futuros projetos similares envolvendo instâncias EC2 e aplicações em Node.js na AWS.