Laboratório JWT - Criando seu próprio *Token JWT* e validando-o.

1.

Crie uma nova instância na Ubuntu na AWS

1.1 No Console AWS -> EC2 -> *Launch Instance*.
1.2 Escolha **Ubuntu 22.04 LTS AMI**.
1.3 Dê um nome qualquer. (*exemplo: lab-jwt*)
1.4 Deixe como estão as demais configurações da Instância.

2.

Após criada, configure o Security Group Rules

2.1 Allow SSH (TCP 22). (Source: 0.0.0.0/0).
2.2 Allow HTTP (TCP 80). (Source: 0.0.0.0/0).
2.3 Allow Custom TCP (TCP 3000). (Source: 0.0.0.0/0).

3.

Conecta à sua Instância.

No terminal da sua máquina, digite:
ssh -i chave.pem ubuntu@ip_da_instancia

4.

Atualizar e instalar dependências na sua Instância.

No terminal da sua instância, digite (um de cada vez):
sudo apt update && sudo apt upgrade -y
sudo apt install -y curl ca-certificates gnupg
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt-get install -y nodejs
sudo apt install -y jq

Após isso, verifique se está tudo correto com:
node -v
npm -v

5.

Criar o projeto Node.js (Express + jsonwebtoken)

No terminal da sua instância, digite:
mkdir ~/lab-jwt && cd ~/lab-jwt
npm init -y
npm install express jsonwebtoken dotenv

6.

Criando arquivo index.js

No terminal da sua instância, digite:

nano index.js

Dentro desse arquivo, cole isso e depois salve:

```
require('dotenv').config();
const express = require('express');
const jwt = require('jsonwebtoken');

const app = express();
app.use(express.json());

const PORT = process.env.PORT || 3000;
const SECRET = process.env.JWT_SECRET || 'troque_esse_seguro';

app.post('/login', (req, res) => {
  const { username } = req.body;

  // Usuário fictício para demo
  const user = {
    id: 1,
    username: username || 'aluno',
    nome: 'Seu nome',
    email: 'seuemail@gmail.com',
    role: 'admin',
    curso: 'Sistemas de Informação'
  };

  // PAYLOAD -> tudo isso vai dentro do token (VISÍVEL no jwt.io)
  const payload = {
    sub: user.id,                // subject (padrão JWT)
    username: user.username,
    nome: user.nome,
    email: user.email,
    role: user.role,
    curso: user.curso
  };

  const token = jwt.sign(payload, SECRET, {
    expiresIn: '5m',             // validade do token
    algorithm: 'HS256',
    issuer: 'jwt-demo-aws',      // quem emitiu
    audience: 'frontend-app'     // quem pode usar
  });
```

```
  res.json({
    mensagem: 'Token gerado com sucesso',
    token
  });
});

function authenticate(req, res, next) {
  const authHeader = req.headers.authorization || '';
  const token = authHeader.replace(/^Bearer\s+/i, '');

  if (!token) {
    return res.status(401).json({ error: 'Token ausente' });
  }

  try {
    const decoded = jwt.verify(token, SECRET, {
      issuer: 'jwt-demo-aws',
      audience: 'frontend-app'
    });

    req.user = decoded; // dados do token agora disponíveis na requisição
    next();
  } catch (err) {
    return res.status(401).json({
      error: 'Token inválido ou expirado',
      detalhes: err.message
    });
  }
}

app.get('/dashboard', authenticate, (req, res) => {
  res.json({
    mensagem: 'Acesso autorizado',
    dados_do_token: req.user
  });
});

app.listen(PORT, () => {
  console.log(`Server rodando na porta ${PORT}`);
});
```

7.

Criando arquivo .env (assinatura do token):

No terminal da sua instância, digite:

nano .env

Dentro do arquivo, cole isso e depois salve:

JWT_SECRET="uma_senha_qualquer_bem_grande"
PORT=3000

8.

Iniciando o servidor Node.js

No terminal da sua instância, digite:

node index.js

9.

Gerando seu primeiro Token JWT:

No terminal da sua máquina, digite:

```
curl -s -X POST http://ip_da_instancia:3000/login \
    -H "Content-Type: application/json" \
    -d '{"username":"nome_qualquer"}' | jq
```

Isso gerará um Token JWT, copie-o.

10.

Acessando rota protegida com Token JWT

No terminal da sua máquina, digite:

```
curl -s http://ip_da_instancia:3000/dashboard \
    -H "Authorization: Bearer TOKEN" | jq
```

OBS: Substitua a palavra TOKEN, no código acima, pelo seu Token que você acabou de criar e copiou. (sem aspas)

BÔNUS.

Parabéns!!! Você acaba de criar seu primeiro **Token JWT** e acessou uma **rota web** com ele**.** Perceba que, caso você tente acessar a mesma rota (repetir o passo 10) com outro **Token JWT**, acontecerá um erro, pois **somente o Token que você gerou pode fazer isso.**

No site https://www.jwt.io/ **você pode colar seu Token e a assinatura dele (que você definiu no arquivo** .env**) para ver o conteúdo dele.** *(OBS: Por ser um exemplo didático, o JWT é apenas assinado, não criptografado. Qualquer pessoa pode ler o conteúdo, mas não pode alterá-lo sem a chave secreta.)*