

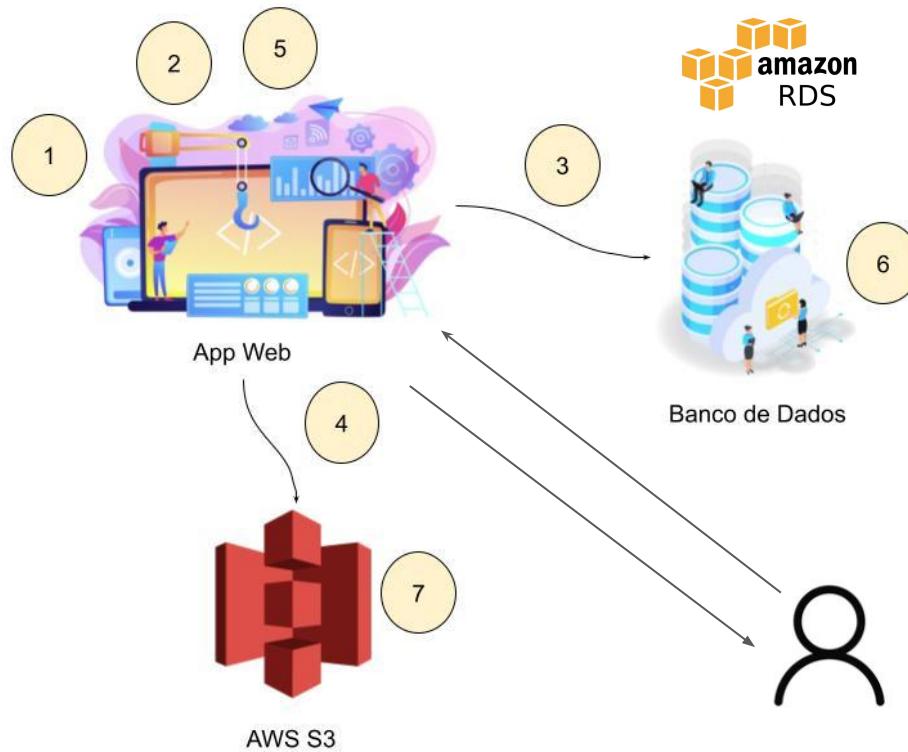


Tópicos em Engenharia de Software

**Aplicação Web integrada aos Serviços AWS S3, AWS RDS,
Login de usuários e Balanceamento de Carga com
Gerenciamento Manual
(Prova de conceito)**

Armando Soares Sousa
Departamento de Computação - UFPI
armando@ufpi.edu.br

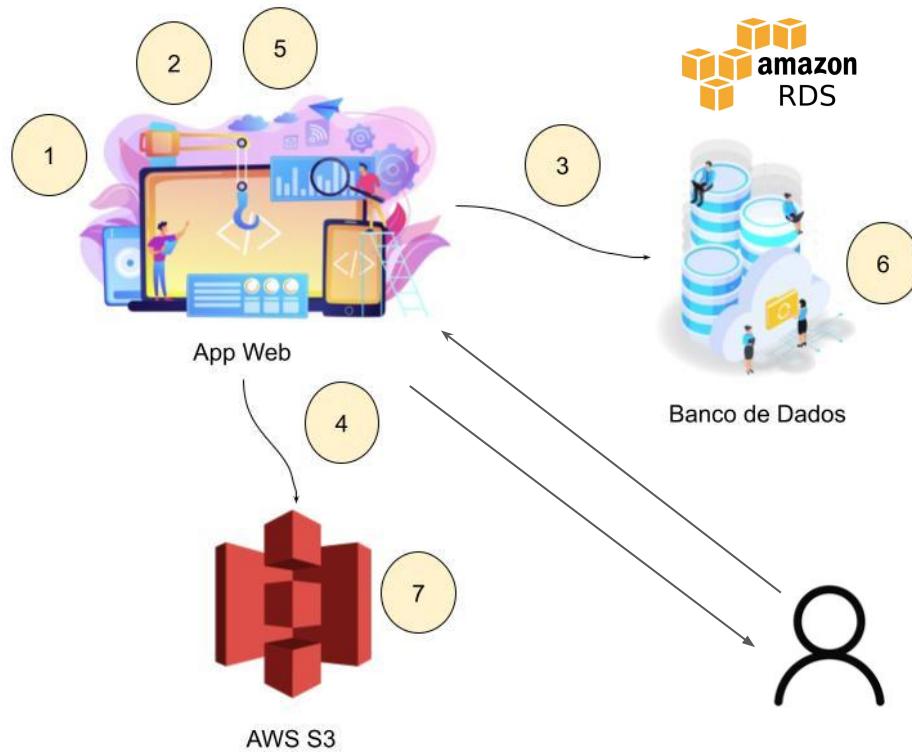
App Web com S3, RDS e controle de autenticação



Exemplo de aplicação web que faz upload e download de arquivos.

1. Estrutura da Aplicação Web
2. Credenciais de Acesso AWS
3. Conexão com o Banco de Dados
4. Conexão com o Serviço S3
5. Rotas da Aplicação Web
6. Estrutura do Banco de dados
7. Estrutura do Bucket S3

App Web com S3, RDS e controle de autenticação



Estrutura da Aplicação Web

```
✓ MYSIMPLEUPLO... ➔ docs  
    > instance  
    > templates  
        <> file_view.html  
        <> files_usuario.html  
        <> home.html  
        <> login.html  
        <> register.html  
        <> upload.html  
    > venv  
    ⚡ .gitignore  
    ✎ banco.py  
    ✎ forms.py  
    ✎ principal.py  
    ⓘ README.md  
    ⌂ requirements.txt  
    ✎ s3_handle.py  
    ✎ utilidades.py
```

Código da Aplicação Web (backend)

✚ principal.py > ⚭ show_image_by_bytes

```
1  from flask import Flask, render_template, request, redirect, url_for, flash, send_file
2  import os
3  import banco
4  import s3_handle
5  import uuid
6  import utilidades
7  import requests
8  import base64
9  import io
10 from forms import LoginForm, RegisterForm
11 from werkzeug.security import check_password_hash, generate_password_hash
12 from flask import session
13 from flask_login import logout_user, LoginManager
```

Código da Aplicação Web (backend)

```
15 # True para limpar a instancia banco de dados atual
16 # Obs: o valor deve ser True na 1a execucao da aplicacao
17 #      para criar um banco limpo a estrutura limpa das tabelas
18 DROP_DATA_BASE = True
19
20 # Carrega os valores das credenciais de acesso da AWS
21 ACCESS_KEY_ID = os.getenv('ACCESS_KEY_ID')
22 SECRET_ACCESS_KEY = os.getenv('SECRET_ACCESS_KEY')
23 DB_USER = os.getenv('DB_USER')
24 PASSWORD_DB_USER = os.getenv('PASSWORD_DB_USER')
25
26 # String de conexao para o RDS
27 SQLALCHEMY_DATABASE_URI_AWS_RDS = f'mysql+pymysql://{{DB_USER}}:{{PASSWORD_DB_USER}}@{{banco.INSTANCE_DB_AWS_RDS}}'
28 # String de conexao para banco local
29 SQLALCHEMY_DATABASE_URI_LOCAL = "sqlite:///users.db"
30
31 # Instancia principal da aplicação
32 app = Flask(__name__)
33 app.secret_key = 'thisismysecretkeyfrommywebapplication'
34 app.config['SQLALCHEMY_DATABASE_URI'] = SQLALCHEMY_DATABASE_URI_AWS_RDS
35 app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
36
37 # Inicializa a instância do banco de dados
38 banco.db.init_app(app)
39 banco.create_tables(app, DROP_DATA_BASE)
40
41 # Para controle de login do usuario
42 login_manager = LoginManager()
43 login_manager.init_app(app)
44 login_manager.login_view = "login"
--
```

Código da Aplicação Web (backend)

```
46 # Carrega o componente S3
47 print('Carregando as credenciais da AWS')
48 s3 = s3_handle.carrega_s3(ACCESS_KEY_ID, SECRET_ACCESS_KEY)
49
50 # Carrega os DAOs de usuarios e arquivos
51 userDao = banco.UserDAO(banco.db)
52 filesDAO = banco.FilesDAO(banco.db)
53
54 @login_manager.user_loader
55 > def load_user(user_id): ...
56
57 @login_manager.unauthorized_handler
58 > def unauthorized_callback(): ...
59
60
61
62 # Lista os recursos disponiveis da web app
63 @app.route('/recursos')
64 > def hello(): ...
```

Código da Aplicação Web (backend)

```
91 # Rota para a pagina de uploads
92 @app.route("/usuarios/<int:id>/upload", methods=["GET", "POST"])
93 def upload_page(id):
94     if 'username' not in session or id != session['user_id']:
95         return redirect(url_for('login'))
96
97     if request.method == "POST":
98         try:
99             uploaded_file = request.files["file-to-save"]
100            if not utilidades.allowed_file(uploaded_file.filename):
101                flash("Tipo de arquivo não permitido!")
102                return redirect(url_for('upload_page', id=id))
103
104            new_filename = uuid.uuid4().hex + '.' + uploaded_file.filename.rsplit('.', 1)[1].lower()
105            s3.upload_fileobj(uploaded_file, s3_handle.BUCKET_NAME, new_filename)
106            file = banco.File(original_filename=uploaded_file.filename, filename=new_filename,
107                               bucket=s3_handle.BUCKET_NAME, region=s3_handle.AWS_S3_REGION)
108            filesDAO.insert_file(file)
109            userDAO.link_to_file(user_id=id, file=file)
110        except Exception as ex:
111            flash(f"Erro no upload! {str(ex)}")
112            return redirect(url_for('upload_page', id=id))
113
114        return redirect(url_for("upload_page", id=id))
115
116    files = userDAO.list_all_files(id)
117    return render_template("upload.html", files=files, id=id)
```

Código da Aplicação Web (backend)

```
119     @app.route("/usuarios/<int:id>/files", methods=["GET"])
120     def lista_files_usuario(id):
121         if 'username' not in session or id != session['user_id']:
122             return redirect(url_for('login'))
123
124         files_usuario = userDAO.list_all_files(id)
125         nome_usuario=session["username"]
126         if not files_usuario:
127             flash('Nenhum arquivo cadastrado!')
128             return redirect(url_for('home_page', id=id, nome_usuario=nome_usuario))
129
130         return render_template("files_usuario.html", files=files_usuario, id=id)
131
132     # Recupera os bytes de um arquivo do bucket S3
133     def get_file_bytes(file_url):
134         response = requests.get(file_url)
135         response.raise_for_status() # Raise an exception for non-200 status codes
136         return response.content
```

Código da Aplicação Web (backend)

```
138 # Recupera os bytes da imagem e guarda em memoria
139 @app.route("/usuarios/<int:id>/myimagebytes/<nome>")
140 def show_image_by_bytes(nome):
141     if 'username' not in session or id != session['user_id']:
142         return redirect(url_for('login'))
143
144     image_url = s3_handle.BUCKET_PATH + "/" + nome
145     image_bytes = get_file_bytes(image_url)
146
147     extensao = utilidades.get_file_extension(nome)
148     my_mimetype = "image" + "/" + extensao
149
150     return send_file(io.BytesIO(image_bytes), mimetype=my_mimetype)
151
152 # Recupera os bytes do arquivo e guarda em memoria
153 @app.route("/usuarios/<int:id>/myfilebytes/<nome>")
154 def show_file_by_bytes(nome):
155     if 'username' not in session or id != session['user_id']:
156         return redirect(url_for('login'))
157
158     file_url = s3_handle.BUCKET_PATH + "/" + nome
159     file_bytes = get_file_bytes(file_url)
160
161     extensao = utilidades.get_file_extension(nome)
162     my_mimetype = utilidades.get_media_type(extensao)
163
164     return send_file(io.BytesIO(file_bytes), mimetype=my_mimetype)
165
```

Código da Aplicação Web (backend)

```
166 # Rota que recupera os bytes do arquivo e guarda em um formato base64
167 # para exibir o conteúdo na página file_view
168 @app.route("/usuarios/<int:id>/myfile/<nome>")
169 def show_file(id, nome):
170     if 'username' not in session or id != session['user_id']:
171         return redirect(url_for('login'))
172
173     file_url = s3_handle.BUCKET_PATH + "/" + nome
174     file_bytes = get_file_bytes(file_url)
175     extensao = utilidades.get_file_extension(nome)
176
177     encoded_bytes = base64.b64encode(file_bytes).decode('utf-8') # Encode as base64 and decode for URI
178     file_type = utilidades.get_media_type(extensao)
179     file_data_uri = f"data:{file_type};base64,{encoded_bytes}"
180
181     return render_template("file_view.html", file_data_uri=file_data_uri, file_type=file_type)
---
```

Código da Aplicação Web (backend)

```
183     @app.route('/login', methods=["GET", "POST"])
184     def login():
185         login_form = LoginForm()
186         if login_form.validate_on_submit():
187             username = login_form.username.data
188             password = login_form.password.data
189             user = userDAO.user_by_username(username)
190             if not user:
191                 flash("Usuário não existe, tente novamente.", category='danger')
192                 return redirect(url_for('login'))
193             elif not check_password_hash(user.password, password):
194                 flash('Senha incorreta, tente novamente.', category='danger')
195                 return redirect(url_for('login'))
196             msg = 'Usuário logado com sucesso!'
197             flash(msg, category='success')
198             session['username'] = username
199             session['user_id'] = user.id
200             return redirect(url_for('home_page', id=user.id, nome_usuario=username))
201
202             if login_form.errors != {}: #If there are not errors from the validations
203                 for err_msg in login_form.errors.values():
204                     flash(f'Existe um erro ao criar o usuário: {err_msg}', category='danger')
205             return render_template("login.html", form=login_form)
```

Código da Aplicação Web (backend)

```
207 # Página de registro
208 @app.route("/register", methods=['GET', 'POST'])
209 def register():
210     register_form = RegisterForm()
211     if register_form.validate_on_submit():
212         name = register_form.name.data
213         username = register_form.username.data
214         email = register_form.email.data
215         password = register_form.password.data
216         password2 = register_form.password2.data
217         resultado_busca = userDAO.user_by_username(username)
218
219         if (resultado_busca):
220             error = f"Usuário {username} já registrado."
221             flash(error, category='danger')
222             return redirect(url_for("register"))
223         if password != password2:
224             error = "O password não foi confirmado!"
225             flash(error, category='danger')
226             return redirect(url_for("register"))
227
228         # the name is available, store it in the database and go to the login page
229         hash_and_salted_password = generate_password_hash(password)
230         usuario = banco.User(name=name, username=username, email=email, password=hash_and_salted_password)
231         userDAO.create_user(user=usuario)
232         msg = 'Novo usuário criado com sucesso!'
233         flash(msg, category='success')
234         return redirect(url_for("login"))
235     if register_form.errors != {}:
236         for err_msg in register_form.errors.values():
237             flash(f'Existe um erro ao criar o usuário: {err_msg}', category='danger')
238             return redirect(url_for("register"))
239     return render_template("register.html", form=register_form)
241
242 @app.route('/logout')
243 def logout():
244     logout_user()
245     session.clear()
246     return redirect(url_for('login'))
247
248 if __name__=='__main__':
249     app.run(debug=True)
```

Código da Aplicação Web (backend)

✚ s3_handle.py > ⚙️ carrega_s3

```
1  import boto3
2  import sys
3
4  BUCKET_NAME = "my-app-files-bucket"
5  AWS_S3_REGION = "us-east-1"
6  BUCKET_PATH = "https://my-app-files-bucket.s3.amazonaws.com"
7
8  def carrega_s3(access_key_id, secret_access_key):
9      try:
10          if access_key_id is None or secret_access_key is None:
11              raise ValueError('Credenciais inválidas!')
12
13          # Componente para acessar o AWS S3
14          my_s3 = boto3.client(
15              's3',
16              aws_access_key_id=access_key_id,
17              aws_secret_access_key=secret_access_key
18          )
19          print("Componente de acesso ao S3 carregado com sucesso!")
20          return my_s3
21      except Exception as ex:
22          print(f"Erro ao carregar o componente do S3: {str(ex)}")
23          sys.exit(1)
```

Código da Aplicação Web (backend)

```
banco.py > ...
1  from flask_sqlalchemy import SQLAlchemy
2  import sys
3  from flask_login import UserMixin
4  from sqlalchemy import ForeignKey
5
6  INSTANCIA_DB_AWS_RDS = 'mydbfilesteste.cdwkkuakqpji.us-east-1.rds.amazonaws.com'
7  BANCO_AWS_RDS = 'mydbfiles'
8
9  # componente de banco de dados
10 db = SQLAlchemy()
11
12 user_files = db.Table('user_files',
13     db.Column('user_id', db.Integer(), ForeignKey('users.id')),
14     db.Column('file_id', db.Integer(), ForeignKey('files.id'))
15 )
16
17 > class User(db.Model, UserMixin):...
25
26  # Classe que representa os dados de um arquivo
27 > class File(db.Model):...
34
35 > class UserDAO:...
96
97 > class FilesDAO:...
126
127 # Cria as tabelas do banco
128 > def create_tables(app, drop_data_base):...
```

Código da Aplicação Web (backend)

```
-->
17 class User(db.Model, UserMixin):
18     __tablename__ = "users"
19     id = db.Column(db.Integer, primary_key=True)
20     name = db.Column(db.String(250), nullable=False)
21     username = db.Column(db.String(250), unique=True, nullable=False)
22     email = db.Column(db.String(250), unique=True, nullable=False)
23     password = db.Column(db.String(500), nullable=False)
24     my_files = db.relationship('File', secondary=user_files, backref='myfiles')
25
26 # Classe que representa os dados de um arquivo
27 class File(db.Model):
28     __tablename__ = "files"
29     id = db.Column(db.Integer, primary_key=True)
30     original_filename = db.Column(db.String(100))
31     filename = db.Column(db.String(100))
32     bucket = db.Column(db.String(100))
33     region = db.Column(db.String(100))
```

Código da Aplicação Web (backend)

```
- .
35 class UserDAO:
36     def __init__(self, db):
37         self.db = db
38
39     > def create_user(self, user): ...
42
43     > def user_by_id(self, user_id): ...
45
46     > def user_by_username(self, username): ...
48
49     > def update_user(self, user): ...
52
53     > def delete_user_by_id(self, user_id): ...
58
59     > def list_users(self): ...
61
62     > def link_to_file(self, user_id, file): ...
70
71     > def link_to_files(self, user_id, files): ...
80
81     > def unlink_file(self, user_id, file): ...
89
90     > def list_all_files(self, user_id): ...
93
94     > def get_file_by_user(self, name): ...
```

Código da Aplicação Web (backend)

```
class FilesDAO:  
    def __init__(self, db):  
        self.db = db  
  
    def insert_file(self, file):  
        try:  
            self.db.session.add(file)  
            self.db.session.commit()  
        except ValueError as ve:  
            raise ValueError(f'Error during insert file - {ve}')  
  
    def query_file_by_name(self, p_name):  
        file = File.query.filter_by(name=p_name).first()  
        return file  
  
    def query_file_by_id(self, p_id):  
        file = File.query.filter_by(id=p_id).first()  
        return file  
  
    def list_all_files(self):  
        return File.query.all()  
  
>    def delete_file(self, file):...
```

```
---  
127 # Cria as tabelas do banco  
128 def create_tables(app, drop_data_base):  
129     try:  
130         with app.app_context():  
131             print('Carrega as tabelas do banco')  
132             if drop_data_base:  
133                 db.drop_all()  
134                 db.create_all()  
135                 db.session.commit()  
136             print('Tabelas carregadas com sucesso!')  
137     except Exception as ex:  
138         print(f'Erro ao carregar o banco! {str(ex)}')  
139         sys.exit(1)
```

Código da Aplicação Web (backend)

forms.py > RegisterForm

```
1  from flask_wtf import FlaskForm
2  from wtforms import StringField, PasswordField, SubmitField, EmailField
3  from wtforms import validators
4
5  class LoginForm(FlaskForm):
6      username = StringField()
7      password = PasswordField()
8      submit = SubmitField('Logar')
9
10 class RegisterForm(FlaskForm):
11     name = StringField('Full name', [validators.DataRequired()])
12     username = StringField('Username', [validators.DataRequired()])
13     password = PasswordField('Password', [validators.DataRequired()])
14     password2 = PasswordField('Password2', [validators.DataRequired()])
15     email = EmailField('Email', [validators.DataRequired(), validators.Email()])
16     submit = SubmitField('Register')
```

Código da Aplicação Web (backend)

utilidades.py > get_media_type

```
1 ALLOWED_EXTENSIONS = {'jpeg', 'jpg', 'png', 'mp4', 'mp3'}
2
3 def allowed_file(filename):
4     return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
5
6 def get_file_extension(filename):
7     return filename.rsplit('.', 1)[1].lower()
8
9 def get_media_type(extension):
10    media_types = {
11        "jpeg": "image/jpeg",
12        "jpg": "image/jpeg",
13        "png": "image/png",
14        "mp4": "video/mp4",
15        "mp3": "audio/mp3"
16    }
17    return media_types.get(extension.lower())
```

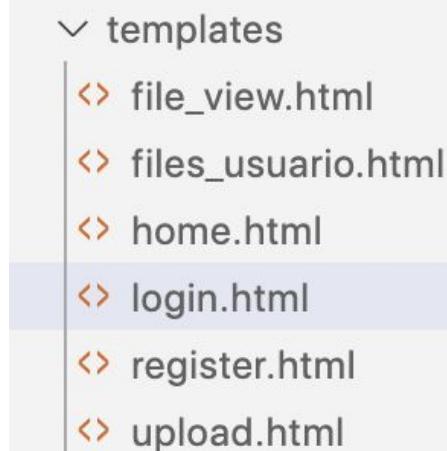
Código da Aplicação Web (backend)

☰ requirements.txt

```
1  Flask==3.0.3
2  Flask-Login==0.6.3
3  Flask-SQLAlchemy==3.1.1
4  Flask-WTF==1.2.1
5  WTForms==3.1.2
6  Jinja2==3.1.3
7  Werkzeug==3.0.2
8  PyMySQL==1.1.0
9  SQLAlchemy==2.0.29
10 requests==2.31.0
11 boto3==1.34.84
12 botocore==1.34.84
13 blinker==1.7.0
14 certifi==2024.2.2
15 charset-normalizer==3.3.2
16 click==8.1.7
17 dnspython==2.6.1
18 email_validator==2.1.1
19 greenlet==3.0.3
20 idna==3.7
21 itsdangerous==2.1.2
22 jmespath==1.0.1
23 MarkupSafe==2.1.5
24 python-dateutil==2.9.0.post0
25 s3transfer==0.10.1
26 six==1.16.0
27 typing_extensions==4.11.0
28 urllib3==2.2.1|
```

Código da Aplicação Web (frontend)

```
5   <body>
6     <h1>Login</h1>
7     <div>
8       {% with messages = get_flashed_messages() %}
9       {% if messages %}
10      <ul class="flashes">
11        {% for message in messages %}
12          <h2>{{ message }}</h2>
13        {% endfor %}
14      </ul>
15      {% endif %}
16      {% endwith %}
17    </div>
18
19    <div>
20      <form action="{{ url_for('login') }}" method="POST">
21        {{ form.hidden_tag() }}
22        <div>
23          {{ form.username(placeholder="User Name") }}
24        </div>
25        <div>
26          {{ form.password(placeholder="Password") }}
27        </div>
28        <div>
29          {{ form.submit() }}
30        </div>
31      </form>
32    </div>
33
34
35    <div>
36      <p>
37        <a href="{{ url_for('register') }}">Registrar uma nova conta</a>
38      </p>
39    </div>
```



Código da Aplicação Web (frontend)

```
10
11    {% if messages %}
12        <ul class="flashes">
13            {% for message in messages %}
14                <h2>{{ message }}</h2>
15            {% endfor %}
16        </ul>
17    {% endif %}
18    {% endwith %}
19</div>
20
21<div>
22    <form action="{{ url_for('register') }}" method="post">
23        {{ form.hidden_tag() }}
24        <div>
25            {{ form.name	placeholder="Full Name" }}
26        </div>
27        <div>
28            {{ form.username	placeholder="Username" }}
29        </div>
30        <div>
31            {{ form.email	placeholder="E-mail" }}
32        </div>
33        <div>
34            {{ form.password	placeholder="Passwod" }}
35        </div>
36        <div>
37            {{ form.password2	placeholder="Confirm Passwod" }}
38        </div>
39        <div>
40            {{ form.submit() }}
41        </div>
42    </form>
43</div>
44
45<div>
46    <a href="{{ url_for('login') }}">Eu já tenho uma conta</a>
47</div>
```

```
    <div>
        <ul class="list-group">
            <li><a href="#">Home</a></li>
            <li><a href="#">Register</a></li>
            <li><a href="#">Login</a></li>
            <li><a href="#">Logout</a></li>
        </ul>
    </div>

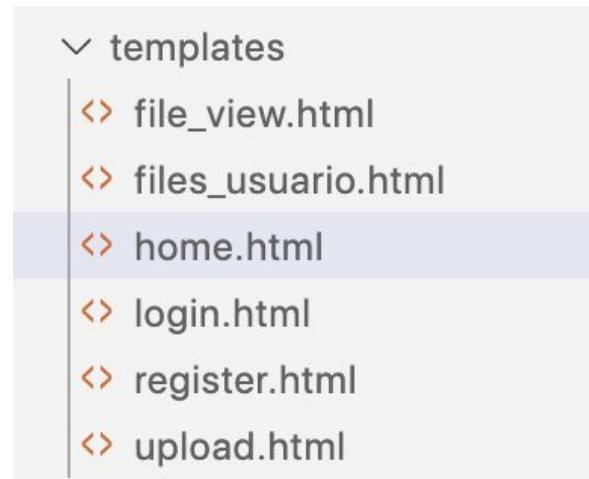
```

templates

- ↳ file_view.html
- ↳ files_usuario.html
- ↳ home.html
- ↳ login.html
- ↳ register.html
- ↳ upload.html

Código da Aplicação Web (frontend)

```
templates > <> home.html > html
1  <html>
2    <head>
3      <title>Home</title>
4    </head>
5    <body>
6      <h1>My Multimedia Files</h1>
7      <p>Upload, Downloads e Visualização de Arquivos (imagens, audios e videos) do AWS S3</p>
8      <p>Usuario: {{nome_usuario}}</p>
9
10     <div>
11       {% with messages = get_flashed_messages() %}
12       {% if messages %}
13         <ul class="flashes">
14           {% for message in messages %}
15             <h2>{{ message }}</h2>
16           {% endfor %}
17         </ul>
18       {% endif %}
19       {% endwith %}
20     </div>
21
22     <div>
23       <a href="{{url_for('upload_page', id=id)}}>Upload File</a>
24       <br>
25       <a href="{{url_for('lista_files_usuario', id=id)}}>Lista arquivos do usuario</a>
26     </div>
27     <div>
28       <br>
29       <a href="{{url_for('logout')}}>[Logout]</a>
30     </div>
31   </body>
32 </html>
```



Código da Aplicação Web (frontend)

```
7 <body>
8   <h3>Upload to S3</h3>
9
10  <div>
11    {% with messages = get_flashed_messages() %}
12    {% if messages %}
13      <ul class="flashes">
14        {% for message in messages %}
15          <h2>{{ message }}</h2>
16        {% endfor %}
17      </ul>
18    {% endif %}
19    {% endwith %}
20  </div>
21
22  <div>
23    <form method="POST" action="{{ url_for('upload_page', id=id)}}" enctype="multipart/form-data">
24      <input type="file" name="file-to-save" />
25      <button>Upload</button>
26    </form>
27  </div>
28  <br>
29  <div>
30    {% for file in files %}
31      <div>
32        {{ file.original_filename }} - <a href="https://{{ file.bucket }}.s3.{{ file.region }}.amazonaws.com/{{ file.key }}">View</a>
33      </div>
34    {% endfor %}
35    <br>
36  </div>
37  <a href="/">[Home]</a>
38
39 </body>
```

▼ templates

- ↳ file_view.html
- ↳ files_usuario.html
- ↳ home.html
- ↳ login.html
- ↳ register.html
- ↳ upload.html

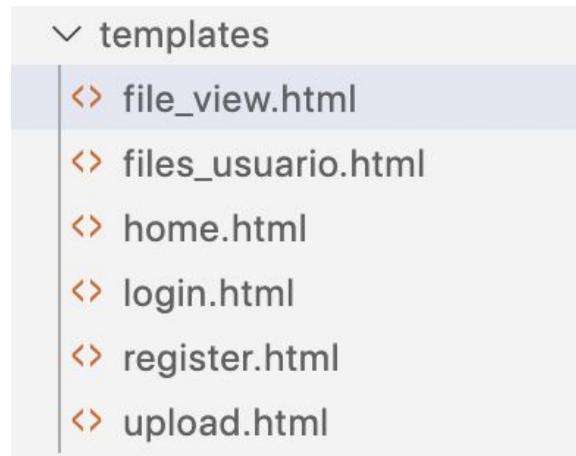
Código da Aplicação Web (frontend)

```
templates > files_usuario.html > html
1  !DOCTYPE html>
2  html lang="en">
3  head>
4      <meta charset="UTF-8">
5      <title>Downloads</title>
6  /head>
7  body>
8      <h3>Lista de arquivos do S3</h3>
9
10     <div>
11         {% with messages = get_flashed_messages() %}
12         {% if messages %}
13             <ul class="flashes">
14                 {% for message in messages %}
15                     <h2>{{ message }}</h2>
16                     {% endfor %}
17                 </ul>
18             {% endif %}
19             {% endwith %}
20     </div>
21
22     {% for file in files %}
23     <div>
24         {{ file.original_filename }} - <a href="https://{{ file.bucket }}.s3.{{ file.region }}.amazonaws.com/{{ f
25         - <a href="{{ url_for('show_file', id=id, nome=file.filename)}}>View</a>
26     </div>
27     {% endfor %}
28     <br>
29     <a href="/">[Home]</a>
30   </body>
31 </html>
```

```
▽ templates
  <> file_view.html
  <> files_usuario.html
  <> home.html
  <> login.html
  <> register.html
  <> upload.html
```

Código da Aplicação Web (frontend)

```
8   -->
9   <h3>Arquivo do S3</h3>
10
11  {% with messages = get_flashed_messages() %}
12  {% if messages %}
13    <ul class="flashes">
14      {% for message in messages %}
15        | <h2>{{ message }}</h2>
16        | {% endfor %}
17    </ul>
18  {% endif %}
19  {% endwith %}
20
21  <h1>File from S3 Byte Content</h1>
22  <!-- view image -->
23  {% if file_type=="image/jpeg" or file_type=="image/png" %}
24  
25  {% endif%}
26
27  {% if file_type=="audio/mp3" %}
28  <!-- play audio -->
29  <audio controls>
30    <source src="{{ file_data_uri }}" type="audio/mpeg">
31    Your browser does not support the audio tag.
32  </audio>
33  {% endif%}
34
35  {% if file_type=="video/mp4" %}
36  <!-- view video -->
37  <video width="320" height="240" controls>
38    <source src="{{ file_data_uri }}" type="video/mp4">
39    Your browser does not support the video tag.
40  </video>
41  {% endif%}
42  <br>
43  <a href="/">[Home]</a>
```



Código da Aplicação Web (fonte)

Disponível em <https://github.com/armandossrecife/mysimpleuploads3rdslogin>

The screenshot shows a GitHub repository page for the repository `mysimpleuploads3rdslogin`, which is public. The main navigation bar includes links for `main`, `2 Branches`, `2 Tags`, a search bar labeled `Go to file`, an `Add file` button, and a `Code` dropdown menu. Below the navigation is a list of commits from the `main` branch, ordered by date. The commits are as follows:

Author	Commit Message	Date
armandossrecife	Update README.md	f60b47a · 12 hours ago
	Create passos_lb_manual.md	12 hours ago
	Refatoracao para permitir manipulacao de imagens, audio...	2 days ago
	.gitignore	5 days ago
	Update README.md	12 hours ago
	1a versao draft	5 days ago
	banco.py	5 days ago
	forms.py	5 days ago
	principal.py	20 hours ago
	Update principal.py	5 days ago
	requirements.txt	5 days ago
	s3_handle.py	5 days ago
	1a versao draft	5 days ago
	utilidades.py	2 days ago
	Refatoracao para permitir manipulacao de imagens, audio...	2 days ago

Protótipos das Telas (Registro)

← → C ▲ Não seguro | http://ec2-54-226-136-220.compute-1.amazonaws.com:8000/register

Register

Full Name
Username
E-mail
Passwod
Confirm Passwod
Register

[Eu já tenho uma conta](#)

Protótipos das Telas (Login)

← → C A Não seguro | http://ec2-54-226-136-220.compute-1.amazonaws.com:8000/login

Login

[Registrar uma nova conta](#)

Protótipos das Telas (Dashboard)

← → C A Não seguro | http://ec2-54-226-136-220.compute-1.amazonaws.com:8000/?id=1&nome_usuario=armando

My Multimedia Files

Upload, Downloads e Visualização de Arquivos (imagens, audios e videos) do AWS S3

Usuario: armando

Usuário logado com sucesso!

[Upload File](#)

[Lista arquivos do usuario](#)

[\[Logout\]](#)

Protótipos das Telas (Upload)

← → C ▲ Não seguro | http://ec2-54-226-136-220.compute-1.amazonaws.com:8000/usuarios/1/upload

Upload to S3

teletubes.png

mypython.png - [306cee1ad8084099bc3497871f5c447b.png](#)

Adrienne.mp4 - [ed306c493610416288232ecbfbc869ae.mp4](#)

Kalimba.mp3 - [cale547135e04d61999428b78e3c0d8f.mp3](#)

An Honest Mistake Closed Captioned.mp4 - [562dab0a33164db491e772f460bc02a8.mp4](#)

15 Overkill.mp3 - [536994dead6c4dad8dd513a7cc653558.mp3](#)

screen_matrix.jpeg - [e0364ddde2e74ecdaa524b4d24ef8fbcc.jpeg](#)

[Home]

Protótipos das Telas (Upload)

← → C A Não seguro | http://ec2-54-226-136-220.compute-1.amazonaws.com:8000/usuarios/1/upload

Upload to S3

Nenhum arquivo escolhido

mypython.png - [306cee1ad8084099bc3497871f5c447b.png](#)

Adrienne.mp4 - [ed306c493610416288232ecbfbc869ae.mp4](#)

Kalimba.mp3 - [ca1e547135e04d61999428b78e3c0d8f.mp3](#)

An Honest Mistake Closed Captioned.mp4 - [562dab0a33164db491e772f460bc02a8.mp4](#)

15 Overkill.mp3 - [536994dead6c4dad8dd513a7cc653558.mp3](#)

screen_matrix.jpeg - [e0364ddde2e74ecdaa524b4d24ef8fbcc.jpeg](#)

teletubes.png - [80e41cd4685b42818cce1982b9f2e24b.png](#)

[\[Home\]](#)

Telas (Lista arquivos do usuário)

← → C A Não seguro | http://ec2-54-226-136-220.compute-1.amazonaws.com:8000/usuarios/1/files

Lista de arquivos do S3

mypython.png - [download](#) - [View](#)

Adrienne.mp4 - [download](#) - [View](#)

Kalimba.mp3 - [download](#) - [View](#)

An Honest Mistake Closed Captioned.mp4 - [download](#) - [View](#)

15 Overkill.mp3 - [download](#) - [View](#)

screen_matrix.jpeg - [download](#) - [View](#)

teletubes.png - [download](#) - [View](#)

[[Home](#)]

Telas (Visualiza Imagem Selecionada)

← → C Não seguro | http://ec2-54-226-136-220.compute-1.amazonaws.com:8000/usuarios/l/myfile/80e41

Arquivo do S3

File from S3 Byte Content



Telas (Lista arquivos do usuário)

← → C A Não seguro | http://ec2-54-226-136-220.compute-1.amazonaws.com:8000/usuarios/1/files

Lista de arquivos do S3

mypython.png - [download](#) - [View](#)

Adrienne.mp4 - [download](#) - [View](#)

Kalimba.mp3 - [download](#) - [View](#)

An Honest Mistake Closed Captioned.mp4 - [download](#) - [View](#)

15 Overkill.mp3 - [download](#) - [View](#)

screen_matrix.jpeg - [download](#) - [View](#)

teletubes.png - [download](#) - [View](#)

[[Home](#)]

Telas (Toca Áudio Selecionado)

← → C ▲ Não seguro | http://ec2-54-226-136-220.compute-1.amazonaws.com:80

Arquivo do S3

File from S3 Byte Content



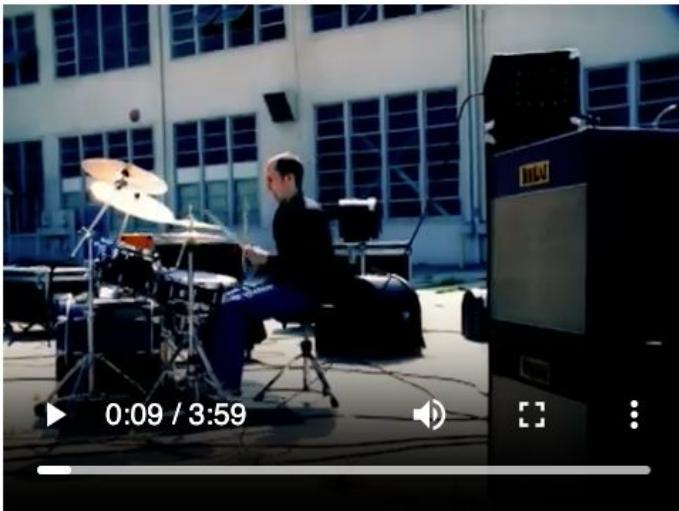
[Home]

Telas (Toca Vídeo Selecionado)

← → C ▲ Não seguro | http://ec2-54-226-136-220.compute-1.amazonaws.com:8000

Arquivo do S3

File from S3 Byte Content



[Home]

Console AWS S3 (Lista de buckets)

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with links to various AWS services like VPC, EC2, EFS, IAM, S3, RDS, DynamoDB, CloudWatch, Trusted Advisor, and Route 53. The S3 icon is highlighted. On the left, there's a sidebar with a 'Amazon S3' section. The main area has a heading 'Snapshot da conta' with a link to 'Todas as regiões da AWS'. Below it, a message says 'O Storage Lens fornece visibilidade sobre o uso e as tendências de atividades.' with a 'Saiba mais' link. There are two tabs: 'Buckets de uso geral' (selected) and 'Buckets de diretórios'. A large table lists two buckets: 'my-app-files-bucket' and 'my-app-image-bucket'. Each row includes columns for Nome, Região da AWS, IAM Access Analyzer, and Data de criação. Buttons for 'Copiar ARN', 'Vazio', 'Excluir', and 'Criar bucket' are also visible.

Nome	Região da AWS	IAM Access Analyzer	Data de criação
my-app-files-bucket	Leste dos EUA (Norte da Virgínia) us-east-1	Exibir analisador para us-east-1	7 Apr 2024 04:17:23 PM -03
my-app-image-bucket	Leste dos EUA (Norte da Virgínia) us-east-1	Exibir analisador para us-east-1	6 Apr 2024 11:28:27 AM -03

Console AWS S3 (Lista de objetos)

Objetos Propriedades Permissões Métricas Gerenciamento Pontos de acesso

Objetos (14) Informações

Copiar URI do S3 Copiar URL Fazer download Abrir Excluir Ações Criar pasta

Carregar

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o [inventário do Amazon S3](#) para obter uma lista de todos os objetos em seu bucket. Para outras pessoas acessarem seus objetos, você precisará conceder permissões explicitamente a eles. [Saiba mais](#)

Localizar objetos por prefixo < 1 >

<input type="checkbox"/>	Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento
<input type="checkbox"/>	173277000f0b4074accbe00d75091fab.jpeg	jpeg	8 Apr 2024 12:21:34 AM -03	339.9 KB	Padrão
<input type="checkbox"/>	2f7967f49c164d87a2e0bc3bae5f0414.jpeg	jpeg	10 Apr 2024 12:26:27 AM -03	433.7 KB	Padrão
<input type="checkbox"/>	40143cef3f9840c284919d96b0b42cab.png	png	9 Apr 2024 10:48:06 PM -03	102.3 KB	Padrão

Console AWS S3 (Política de acesso de um bucket)

Política do bucket

[Editar](#)[Excluir](#)

A política de bucket, escrita em JSON, fornece acesso aos objetos armazenados no bucket. As políticas de bucket não se aplicam a objetos de propriedade de outras contas. [Saiba mais](#) 

```
{  
  "Version": "2012-10-17",  
  "Id": "Policy1712415077164",  
  "Statement": [  
    {  
      "Sid": "Stmt1712415054743",  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::my-app-files-bucket/*"  
    }  
  ]  
}
```

[!\[\]\(a80fb6830acb1cf4906e86ddade0b6a0_img.jpg\) Copiar](#)

Console AWS S3 (Detalhes de um objeto)

ca2ec5697d0744699ea729fb51b1ffd7.png [Informações](#)

[Copiar URI do S3](#) [Fazer download](#) [Abrir](#) [Ações de objeto](#)

[Propriedades](#) [Permissões](#) [Versões](#)

Visão geral do objeto

Proprietário	URI do S3
armando.sousa	s3://my-app-files-bucket/ca2ec5697d0744699ea729fb51b1ffd7.png
Região da AWS	Nome de recurso da Amazon (ARN)
Leste dos EUA (Norte da Virgínia) us-east-1	arn:aws:s3:::my-app-files-bucket/ca2ec5697d0744699ea729fb51b1ffd7.png
Última modificação	Tag da entidade (Etag)
10 Apr 2024 12:17:51 AM -03	570ad5c57c674732937c16fd53e4f555
Tamanho	URL de objeto
4.5 MB	https://my-app-files-bucket.s3.amazonaws.com/ca2ec5697d0744699ea729fb51b1ffd7.png
Tipo	
png	
Chave	
ca2ec5697d0744699ea729fb51b1ffd7.png	

Console AWS S3 (Permissões de um objeto)

ca2ec5697d0744699ea729fb51b1ffd7.png [Informações](#)

[Copiar URI do S3](#) [Fazer download](#) [Abrir](#) [Ações de objeto ▾](#)

[Propriedades](#) [Permissões](#) [Versões](#)

Lista de controle de acesso (ACL) [Editar](#)

Conceda permissões básicas de leitura/gravação a contas da AWS. [Saiba mais](#)

Esse bucket tem a configuração imposto pelo proprietário do bucket aplicada à propriedade do objeto
Quando a configuração [imposto pelo proprietário do bucket](#) for aplicado, use políticas de bucket para controlar o acesso. [Saiba mais](#)

Beneficiário	Objeto	ACL do objeto
Proprietário do objeto (sua conta da AWS) ID canônico: da018f3776d306f41c7bb92ab6c465c8451759385bccb70c38082a8ab5860443	Leitura	Leitura, gravação
Todos (acesso público) Grupo: http://acs.amazonaws.com/groups/global/AllUsers	-	-
Grupo de usuários autenticados (qualquer pessoa com uma conta da AWS) Grupo: http://acs.amazonaws.com/groups/global/AuthenticatedUsers	-	-

Console AWS S3 (Download de um objeto)

<https://my-app-files-bucket.s3.amazonaws.com/ca2ec5697d0744699ea729fb51b1ffd7.png>



Console AWS RDS (Instância criada)

Banco de dados criado com êxito mydbfileteste

Você pode usar as configurações do mydbfileteste para simplificar a configuração de complementos do banco de dados sugeridos enquanto concluímos a criação do banco de dados para você.

[Visualizar detalhes da conexão](#)

RDS > Bancos de dados

Considerar criar uma implantação azul/verde para minimizar o tempo de inatividade durante as atualizações

Você pode considerar o uso de implantações azuis/verdes do Amazon RDS e minimizar seu tempo de inatividade durante as atualizações. Uma implantação azul/verde fornece um ambiente de preparação para alterações nos bancos de dados de produção. [Guia do usuário do RDS](#) [Guia do usuário do Aurora](#)

Bancos de dados (1)		<input checked="" type="radio"/> Recursos do grupo		Modificar	Ações ▾	Restaurar do S3	Criar banco de dados
<input type="text"/> Filtrar por bancos de dados		1					
Identificador de banco de dados	Status	Função	Mecanismo	Região e AZ	Tamanho	Recomendações	
mydbfileteste	Disponível	Instância	MariaDB	us-east-1d	db.t3.micro		

Console AWS RDS (Segurança e conexão)

RDS > Bancos de dados > mydbfilesteste

mydbfilesteste

Modificar Ações ▾

Resumo

Identificador de banco de dados

mydbfilesteste

CPU

3.35%

Status

Disponível

Função

Instância

Mecanismo

MariaDB

Classe

db.t3.micro

Atividade atual

0 Conexões

Recomendações

[Segurança e conexão](#)

[Monitoramento](#)

[Logs e eventos](#)

[Configuração](#)

[Manutenção e backups](#)

[Tags](#)

[Recomendações](#)

Segurança e conexão

Endpoint e porta

Endpoint

mydbfilesteste.cdwkkuakqj.us-east-1.rds.amazonaws.com

Porta

3306

Redes

Zona de disponibilidade

us-east-1d

VPC

vpc-0e1426c149056cec2

Grupo de sub-redes

default-vpc-0e1426c149056cec2

Sub-redes

subnet-04dbbf803db26da23

subnet-04b41e74b7ff36a19

subnet-02d79321c9d25cdc3

subnet-0861dd1952bfdfc90

subnet-0c163a828b14fba3d

subnet-06fb92bf4c5bbe16e

Segurança

Grupos de segurança da VPC

default (sg-00750ce86a5dd273e)

Ativo

Publicamente acessível

Sim

Autoridade de certificação [Informações](#)

rds-ca-rsa2048-g1

Data da autoridade de certificado

May 25, 2061, 20:34 (UTC-03:00)

Data de expiração do certificado da instância de banco de dados

April 13, 2025, 13:25 (UTC-03:00)

Console AWS RDS (Regras de segurança e replicação)

Regras de grupos de segurança (3)		
Grupo de segurança	Tipo	Regra
default (sg-00750ce86a5dd273e)	EC2 Security Group - Inbound	sg-00750ce86a5dd273e
default (sg-00750ce86a5dd273e)	CIDR/IP - Inbound	0.0.0.0/0
default (sg-00750ce86a5dd273e)	CIDR/IP - Outbound	0.0.0.0/0

Replicação (1)					
Identificador de banco de dados	Função	Região e AZ	Origem da replicação	Estado da replicação	Atraso
mydbfilesteste	Instância	us-east-1d	-	-	-

Console AWS RDS (Configuração)

mydbfileteste

Resumo

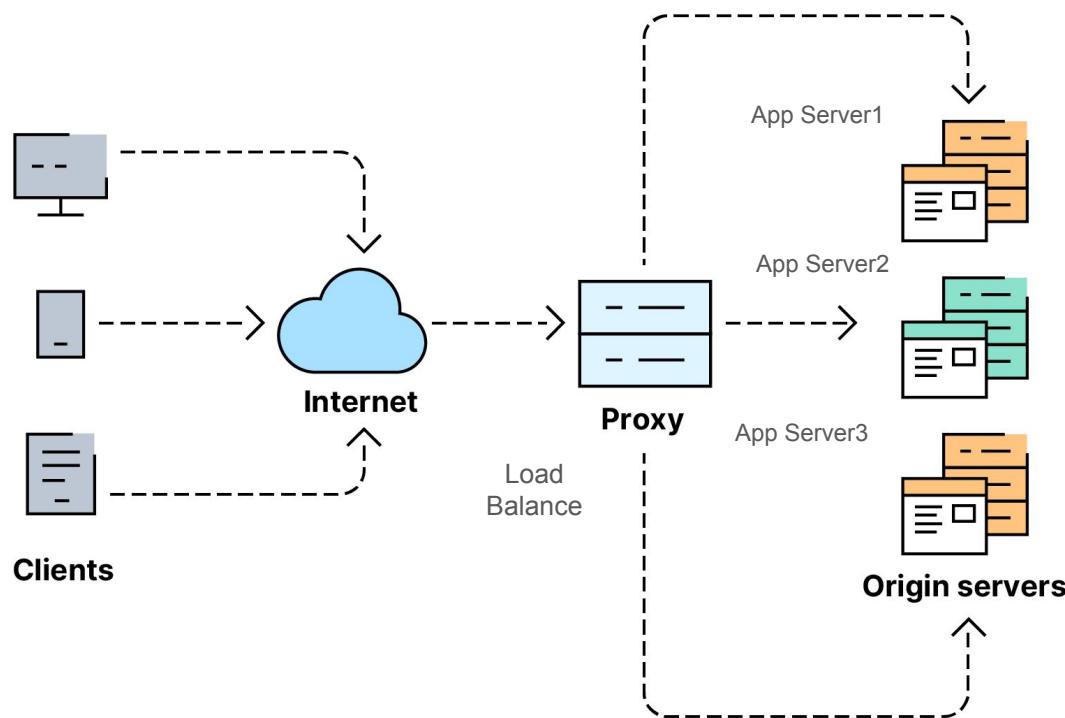
Identificador de banco de dados mydbfileteste	Status Disponível	Função Instância	Mecanismo MariaDB	Recomendações
CPU 3.55%	Classe db.t3.micro	Atividade atual 0 Conexões	Região e AZ us-east-1d	

Segurança e conexão | Monitoramento | Logs e eventos | **Configuração** | Manutenção e backups | Tags | Recomendações

Instância

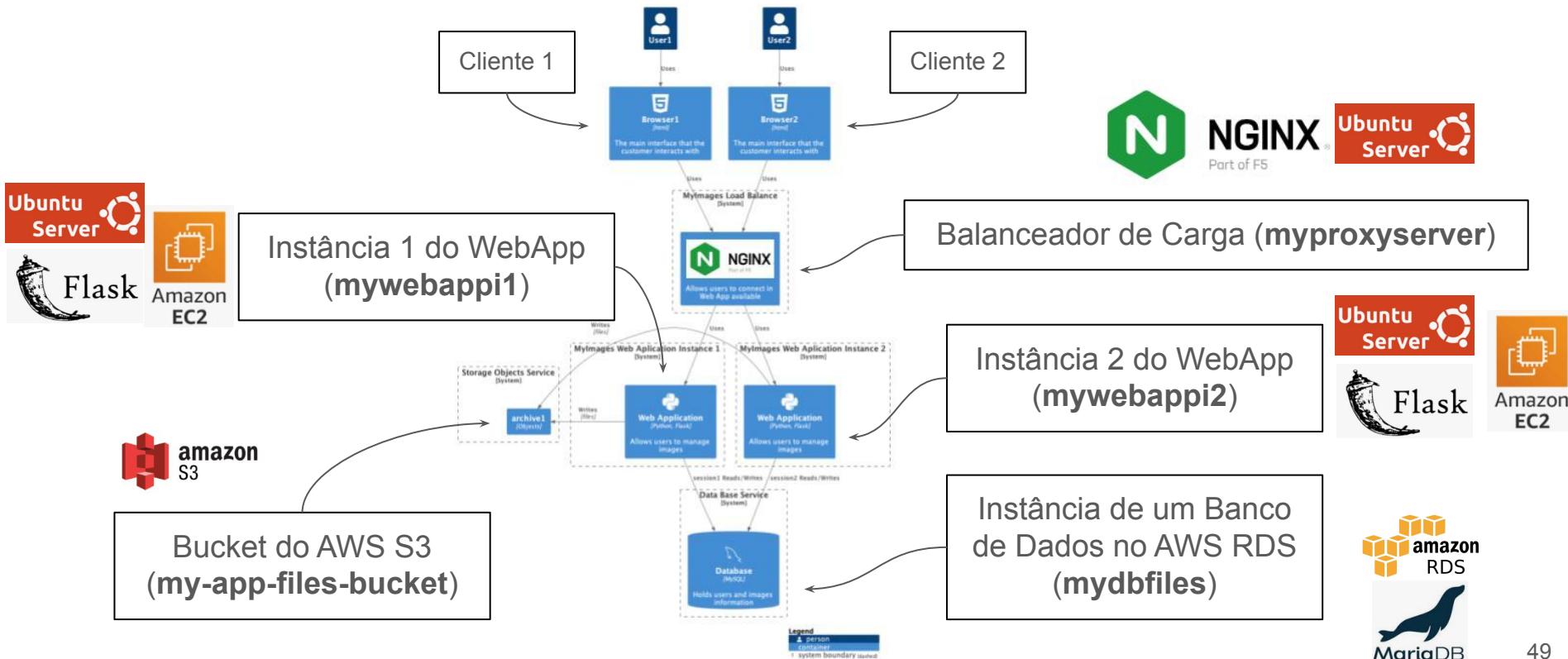
Configuração ID da instância de banco de dados mydbfileteste Versão do mecanismo 10.11.6 Nome do banco de dados mydbfiles Modelo de licença General Public License Grupos de opções default:mariadb-10-11 Em sincronia Nome de recurso da Amazon (ARN) arn:aws:rds:us-east-1:229020406899:db:mydbfileteste ID do recurso db-WD053HKB3AI4KBRAL553K3I2I	Classe de instância Classe de instância db.t3.micro vCPU 2 RAM 1 GB Disponibilidade Nome do usuário principal admin Senha principal ***** Autenticação do banco de dados do IAM Não habilitado Multi-AZ	Armazenamento Criptografia Não habilitado Tipo de armazenamento SSD de uso geral (gp2) Armazenamento 20 GiB IOPS provisionadas - Taxa de throughput - Escalabilidade automática do armazenamento Desabilitado Configuração do sistema de arquivos de armazenamento	Performance Insights Performance Insights habilitado Desativado
---	---	--	--

Balanceamento de Carga



- Os平衡adores de carga melhoram a disponibilidade e a capacidade de resposta dos aplicativos e **evitam a sobrecarga do servidor.**
- Cada balanceador de carga fica entre os dispositivos clientes e os servidores back-end, **recebendo e distribuindo as solicitações** recebidas para qualquer servidor disponível capaz de atendê-las.

Balanceamento de Carga (Gerenciamento Manual)



Console AWS Instâncias do EC2

The screenshot shows the AWS EC2 Instances console interface. At the top, there's a navigation bar with links for VPC, EC2, EC2 Image Builder, EFS, IAM, S3, RDS, DynamoDB, CloudWatch, Trusted Advisor, and Route 53. A search bar and a notifications section are also present. Below the navigation bar, a green banner displays a success message: "Iniciadas com sucesso i-0e509de7af5a8d0fa, i-0c28f1bddc183849e, i-05e720be43f40009e". The main area is titled "Instâncias (3) Informações" and contains a table with the following data:

	Name	ID da instância	Estado da instância	Tipo de inst...	Verificação de stat	Status do alarm	Zona de dispon...	DNS IPv4 públic
<input type="checkbox"/>	myproxyserver	i-0e509de7af5a8d0fa	Executando	t2.micro	Inicializando	Exibir alarmes +	us-east-1c	ec2-54-226-136-
<input type="checkbox"/>	mywebappi1	i-0c28f1bddc183849e	Executando	t2.micro	Inicializando	Exibir alarmes +	us-east-1c	ec2-174-129-10
<input type="checkbox"/>	mywebappi2	i-05e720be43f40009e	Executando	t2.micro	Inicializando	Exibir alarmes +	us-east-1c	ec2-54-145-9-15

Instâncias do EC2 (myproxyserver/detalhes)

EC2 > Instâncias > i-0e509de7af5a8d0fa

Resumo da instância para i-0e509de7af5a8d0fa (myproxyserver) [Informações](#)

Atualizado há less than a minute

ID da instância i-0e509de7af5a8d0fa (myproxyserver)	Endereço IPv4 público 54.226.136.220 [endereço aberto]	Endereços IPv4 privados 172.31.17.150
Endereço IPv6 -	Estado da instância Executando	DNS IPv4 público ec2-54-226-136-220.compute-1.amazonaws.com [endereço aberto]
Tipo de nome do host Nome do IP: ip-172-31-17-150.ec2.internal	Nome do DNS de IP privado (somente IPv4) ip-172-31-17-150.ec2.internal	Endereços IP elásticos -
Nome do DNS do recurso privado de resposta IPv4 (A)	Tipo de instância t2.micro	Descoberta do AWS Compute Optimizer Opte por participar do AWS Compute Optimizer para obter recomendações. Saiba mais
Endereço IP atribuído automaticamente 54.226.136.220 [IP público]	ID da VPC vpc-0e1426c149056cec2	Nome do Grupo do Auto Scaling -
Função do IAM -	ID da sub-rede subnet-04dbbf803db26da23	

Instâncias do EC2 (myproxyserver/segurança)

Detalhes | Status e alarmes [Novo](#) | Monitoramento | **Segurança** | Redes | Armazenamento | Tags

▼ Detalhes de segurança

Função do IAM

-

ID do proprietário

 229020406899

Data de lançamento

Thu Apr 18 2024 10:14:22 GMT-0300 (Horário Padrão de Brasília)

Grupos de segurança

 [sg-00750ce86a5dd273e \(default\)](#)

▼ Regras de entrada

Regras de filtro					
Nome	ID da regra do grupo de se...	Intervalo de po...	Protocolo	Origem	Grupos de segurança
-	sgr-085eb5470ffbaab42	22	TCP	177.193.20.153/32	default 
-	sgr-09d75c045d4e63a34	22	TCP	0.0.0.0/0	default 
-	sgr-05cbf81e5b8d36287	8000	TCP	0.0.0.0/0	default 
-	sgr-0a24b7294172a10a6	3306	TCP	0.0.0.0/0	default 
-	sgr-018aeb5ad314e51f4	80	TCP	0.0.0.0/0	default 
-	sgr-00732a74db38660c2	Todos	Todos	sg-00750ce86a5dd273e 	default 
-	sgr-0f1e3a08a9879e203	5000	TCP	0.0.0.0/0	default 

Instâncias do EC2 (myproxyserver/rede)

Detalhes | Status e alarmes [Novo](#) | Monitoramento | Segurança | **Redes** | Armazenamento | Tags

▼ Detalhes da rede [Informações](#)

Endereço IPv4 público 54.226.136.220 [endereço aberto]	Endereços IPv4 privados 172.31.17.150	ID da VPC vpc-0e1426c149056cec2
DNS IPv4 público ec2-54-226-136-220.compute-1.amazonaws.com [endereço aberto]	Nome do DNS de IP privado (somente IPv4) ip-172-31-17-150.ec2.internal	
ID da sub-rede subnet-04dbbf803db26da23	Endereços IPv6 -	Endereços IPv4 privados secundários -
Zona de disponibilidade us-east-1c	Endereços IP da operadora (temporários) -	ID do outpost -
Usar RBN como nome de host do sistema operacional convidado Desabilitado	Responder IPv4 do nome de host DNS de RBN Habilitado	

▼ Interfaces de rede (1) [Informações](#)

Instâncias do EC2 (myproxyserver/armazenamento)

Detalhes Status e alarmes [Novo](#) Monitoramento Segurança Redes [Armazenamento](#) Tags

▼ Detalhes de dispositivo raiz

Nome de dispositivo raiz



/dev/sda1

Tipo de dispositivo raiz

EBS

Otimização para o EBS
desativado

▼ Dispositivo de blocos

[Filtrar dispositivos de blocos](#)

ID de volume	Nome do dispositivo	Tamanho do volume	Status da associação	Hora da associação	Criptografia	ID da chave do KMS
vol-0540f71f2598231f6	/dev/sda1	25	Associado	2024/04/17 23:26 GMT-3	Não	-

Instâncias do EC2 (myproxyserv/cesso via SSH)

[EC2](#) > [Instâncias](#) > [i-0e509de7af5a8d0fa](#) > Conectar-se à instância

Conectar-se à instância Informações

Conecte-se à sua instância i-0e509de7af5a8d0fa (myproxyserv) usando qualquer uma destas opções

Conexão de instância do EC2

Gerenciador de sessões

Cliente SSH

Console de série do EC2

ID da instância

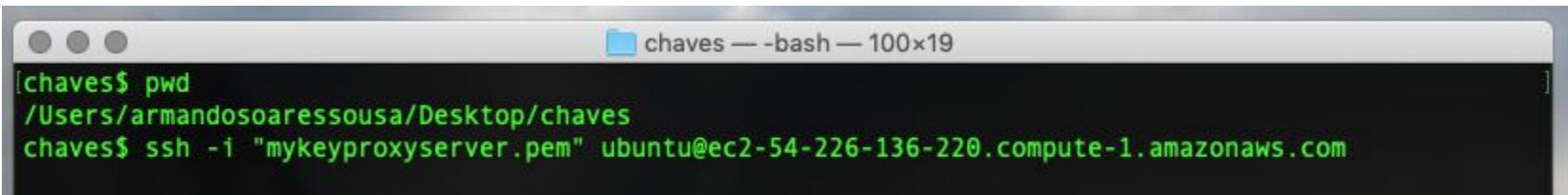
[i-0e509de7af5a8d0fa \(myproxyserv\)](#)

1. Abra um cliente SSH.
2. Localize o arquivo de chave privada. A chave usada para executar esta instância é mykeyproxyserv.pem
3. Execute este comando, se necessário, para garantir que sua chave não fique visível publicamente.
 chmod 400 "mykeyproxyserv.pem"
4. Conecte-se à sua instância usando sua DNS pública:
 ec2-54-226-136-220.compute-1.amazonaws.com

Exemplo:

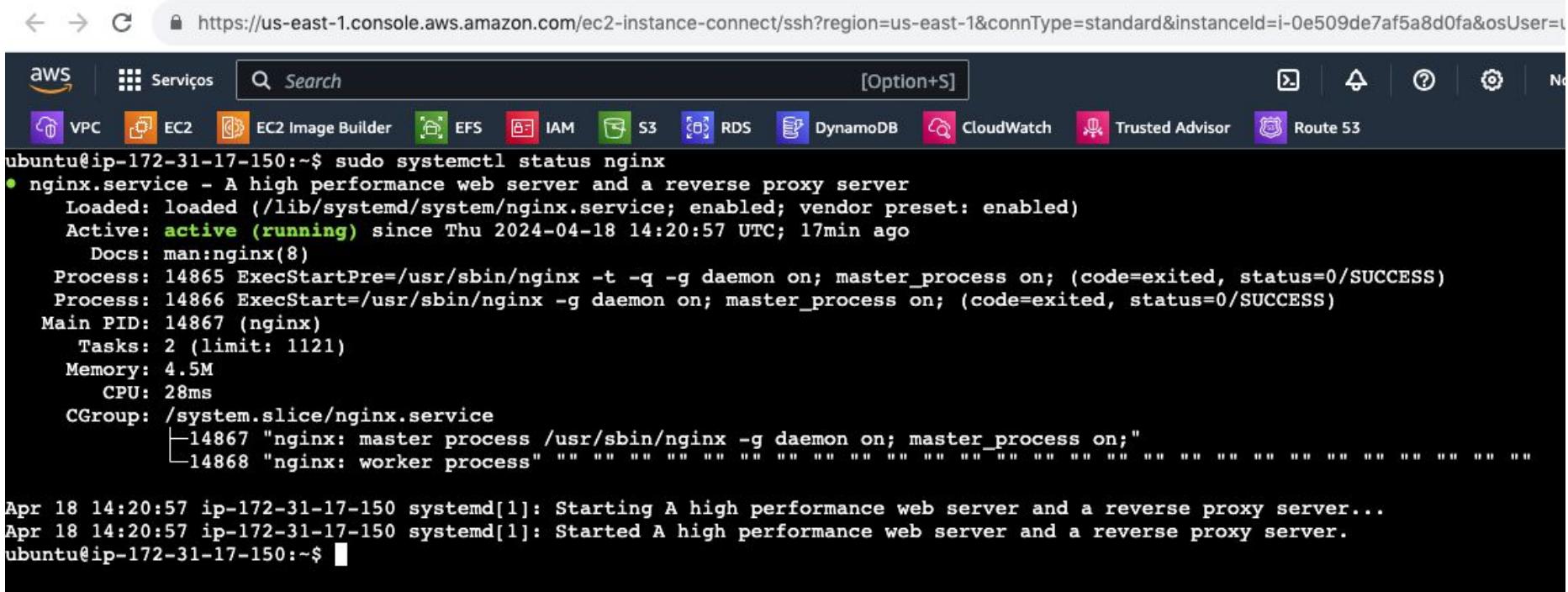
ssh -i "mykeyproxyserv.pem" ubuntu@ec2-54-226-136-220.compute-1.amazonaws.com

Instâncias do EC2 (myproxyserver/acesso via SSH)



```
chaves$ pwd  
/Users/armandosoressousa/Desktop/chaves  
chaves$ ssh -i "mykeyproxyserver.pem" ubuntu@ec2-54-226-136-220.compute-1.amazonaws.com
```

Instâncias do EC2 (myproxyservcer/serviço Nginx)



A screenshot of a AWS CloudWatch terminal window. The top navigation bar shows the URL: https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0e509de7af5a8d0fa&osUser=ubuntu@ip-172-31-17-150. The window title is "Search". Below the title bar are icons for VPC, EC2, EC2 Image Builder, EFS, IAM, S3, RDS, DynamoDB, CloudWatch, Trusted Advisor, and Route 53. A search bar contains the placeholder "[Option+S]". The main content area displays the output of the "sudo systemctl status nginx" command. The output shows the nginx service is active (running) since April 18, 2024, at 14:20:57 UTC. It details the process IDs (14865, 14866, 14867, 14868), memory usage (4.5M), CPU usage (28ms), and the master and worker processes. Log entries at the bottom show the service starting and being started.

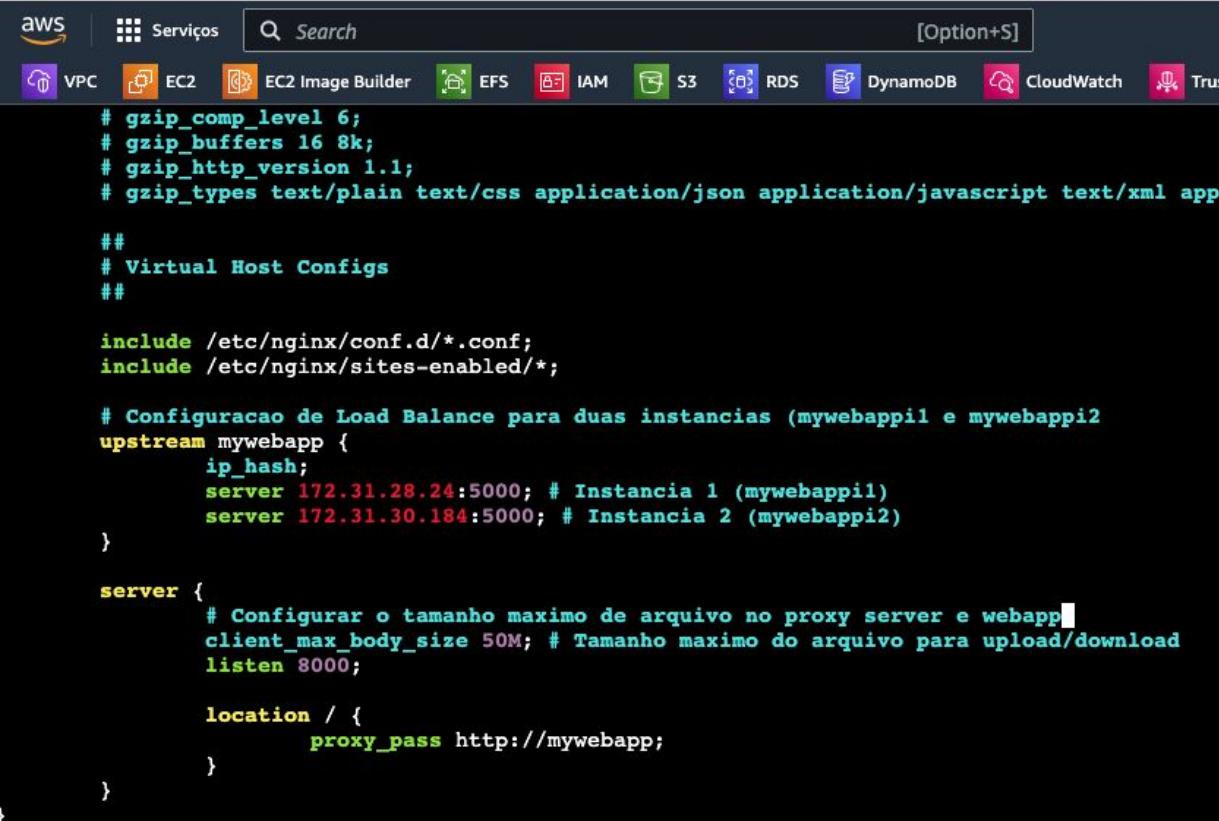
```
ubuntu@ip-172-31-17-150:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2024-04-18 14:20:57 UTC; 17min ago
    Docs: man:nginx(8)
 Process: 14865 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Process: 14866 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 14867 (nginx)
   Tasks: 2 (limit: 1121)
     Memory: 4.5M
       CPU: 28ms
      CGroup: /system.slice/nginx.service
          ├─14867 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
          └─14868 "nginx: worker process"

Apr 18 14:20:57 ip-172-31-17-150 systemd[1]: Starting A high performance web server and a reverse proxy server...
Apr 18 14:20:57 ip-172-31-17-150 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-17-150:~$ █
```

Nginx: <https://nginx.org/en/docs>

[Install and configure Nginx in Ubuntu Server](#)

myproxyserver/serviço Nginx (*Load Balance*)



The screenshot shows the AWS Management Console with the "Serviços" (Services) tab selected. A search bar at the top right contains the text "[Option+S]". Below the search bar, there are links to various AWS services: VPC, EC2, EC2 Image Builder, EFS, IAM, S3, RDS, DynamoDB, CloudWatch, and Trusted Advisor. The main content area displays a block of Nginx configuration code.

```
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

## Virtual Host Configs

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;

# Configuração de Load Balance para duas instâncias (mywebapp1 e mywebapp2)
upstream mywebapp {
    ip_hash;
    server 172.31.28.24:5000; # Instancia 1 (mywebapp1)
    server 172.31.30.184:5000; # Instancia 2 (mywebapp2)
}

server {
    # Configurar o tamanho maximo de arquivo no proxy server e webapp
    client_max_body_size 50M; # Tamanho maximo do arquivo para upload/download
    listen 8000;

    location / {
        proxy_pass http://mywebapp;
    }
}
```

O balanceamento de carga em diversas instâncias de aplicativos é uma técnica comumente usada para otimizar a utilização de recursos, maximizar o rendimento, reduzir a latência e garantir configurações tolerantes a falhas.

É possível usar o nginx como um平衡ador de carga HTTP muito eficiente para distribuir o tráfego para vários servidores de aplicativos e melhorar o desempenho, escalabilidade e confiabilidade de aplicações web com o nginx.

[Nginx as HTTP load balancer](#)

Instâncias do EC2 (mywebappi1/detalhes)

EC2 > Instâncias > i-0c28f1bddc183849e

Resumo da instância para i-0c28f1bddc183849e (mywebappi1) [Informações](#)

Atualizado há less than a minute

ID da instância i-0c28f1bddc183849e (mywebappi1)	Endereço IPv4 público 174.129.101.223 [endereço aberto]	Endereços IPv4 privados 172.31.28.240
Endereço IPv6 -	Estado da instância Executando	DNS IPv4 público ec2-174-129-101-223.compute-1.amazonaws.com [endereço aberto]
Tipo de nome do host Nome do IP: ip-172-31-28-240.ec2.internal	Nome do DNS de IP privado (somente IPv4) ip-172-31-28-240.ec2.internal	Endereços IP elásticos -
Nome do DNS do recurso privado de resposta IPv4 (A)	Tipo de instância t2.micro	Descoberta do AWS Compute Optimizer Opte por participar do AWS Compute Optimizer para obter recomendações. Saiba mais
Endereço IP atribuído automaticamente 174.129.101.223 [IP público]	ID da VPC vpc-0e1426c149056cec2	Nome do Grupo do Auto Scaling -
Função do IAM -	ID da sub-rede subnet-04dbbf803db26da23	

Instâncias do EC2 (mywebappi1/segurança)

Detalhes | Status e alarmes [Novo](#) | Monitoramento | **Segurança** | Redes | Armazenamento | Tags

▼ Detalhes de segurança

Função do IAM
-

ID do proprietário
 229020406899

Data de lançamento
Thu Apr 18 2024 10:14:22 GMT-0300 (Horário Padrão de Brasília)

Grupos de segurança
 [sg-00750ce86a5dd273e \(default\)](#)

▼ Regras de entrada

Regras de filtro						<	1	>
Nome	ID da regra do grupo de se...	Intervalo de po...	Protocolo	Origem	Grupos de segurança			
-	sgr-085eb5470ffbaab42	22	TCP	177.193.20.153/32	default ↗			
-	sgr-09d75c045d4e63a34	22	TCP	0.0.0.0/0	default ↗			
-	sgr-05cbf81e5b8d36287	8000	TCP	0.0.0.0/0	default ↗			
-	sgr-0a24b7294172a10a6	3306	TCP	0.0.0.0/0	default ↗			
-	sgr-018aeb5ad314e51f4	80	TCP	0.0.0.0/0	default ↗			
-	sgr-00732a74db38660c2	Todos	Todos	sg-00750ce86a5dd273e ↗	default ↗			
-	sgr-0f1e3a08a9879e203	5000	TCP	0.0.0.0/0	default ↗			

Instâncias do EC2 (mywebappi1/redes)

Detalhes	Status e alarmes Novo	Monitoramento	Segurança	Redes	Armazenamento	Tags
▼ Detalhes da rede Informações						
Endereço IPv4 público			Endereços IPv4 privados		ID da VPC	
<input type="checkbox"/> 174.129.101.223 endereço aberto			<input type="checkbox"/> 172.31.28.240		<input type="checkbox"/> vpc-0e1426c149056cec2	
DNS IPv4 público			Nome do DNS de IP privado (somente IPv4)		Endereços IPv4 privados secundários	
<input type="checkbox"/> ec2-174-129-101-223.compute-1.amazonaws.com endereço aberto			<input type="checkbox"/> ip-172-31-28-240.ec2.internal		<input type="checkbox"/>	
ID da sub-rede			Endereços IPv6		ID do outpost	
<input type="checkbox"/> subnet-04dbbf803db26da23			-		-	
Zona de disponibilidade			Endereços IP da operadora (temporários)		Responder IPv4 do nome de host DNS de RBN	
<input type="checkbox"/> us-east-1c			-		<input type="checkbox"/>	
Usar RBN como nome de host do sistema operacional convidado			Habilitado			
<input type="checkbox"/> Desabilitado						
▼ Interfaces de rede (1) Informações						

Instâncias do EC2 (mywebapp1/armazenamento)

Detalhes | Status e alarmes [Novo](#) | Monitoramento | Segurança | Redes | **Armazenamento** | Tags

▼ Detalhes de dispositivo raiz

Nome de dispositivo raiz

/dev/sda1

Tipo de dispositivo raiz

EBS

Otimização para o EBS

desativado

▼ Dispositivo de blocos

Filtrar dispositivos de blocos

ID de volume	Nome do dispositivo	Tamanho do volume	Status da associação	Hora da associação	Criptografia	ID da chave do KMS
vol-044cd19470ccc4131	/dev/sda1	25	Associado	2024/04/17 23:58 GMT-3	Não	-

Instâncias do EC2 (mywebappi2/detalhes)

EC2 > Instâncias > i-05e720be43f40009e

Resumo da instância para i-05e720be43f40009e (mywebappi2) [informações](#)

Atualizado há less than a minute

ID da instância	Endereço IPv4 público	Endereços IPv4 privados
i-05e720be43f40009e (mywebappi2)	54.145.9.153 [endereço aberto]	172.31.30.184
Endereço IPv6	Estado da instância	DNS IPv4 público
-	Executando	ec2-54-145-9-153.compute-1.amazonaws.com [endereço aberto]
Tipo de nome do host	Nome do DNS de IP privado (somente IPv4)	Endereços IP elásticos
Nome do IP: ip-172-31-30-184.ec2.internal	ip-172-31-30-184.ec2.internal	-
Nome do DNS do recurso privado de resposta	Tipo de instância	Descoberta do AWS Compute Optimizer
IPv4 (A)	t2.micro	Opte por participar do AWS Compute Optimizer para obter recomendações.
Endereço IP atribuído automaticamente	ID da VPC	Saiba mais
54.145.9.153 [IP público]	vpc-0e1426c149056cec2	
Função do IAM	ID da sub-rede	Nome do Grupo do Auto Scaling
-	subnet-04dbbf803db26da23	-

Instâncias do EC2 (mywebappi2/segurança)

Detalhes | Status e alarmes [Novo](#) | Monitoramento | **Segurança** | Redes | Armazenamento | Tags

▼ Detalhes de segurança

Função do IAM
-

ID do proprietário
[🔗](#)
229020406899

Data de lançamento
Thu Apr 18 2024 10:14:22 GMT-0300 (Horário Padrão de Brasília)

Grupos de segurança
[🔗](#)
sg-00750ce86a5dd273e (default)

▼ Regras de entrada

Regras de filtro					
Nome	ID da regra do grupo de se...	Intervalo de po...	Protocolo	Origem	Grupos de segurança
-	sgr-085eb5470ffbaab42	22	TCP	177.193.20.153/32	default ↗
-	sgr-09d75c045d4e63a34	22	TCP	0.0.0.0/0	default ↗
-	sgr-05cbf81e5b8d36287	8000	TCP	0.0.0.0/0	default ↗
-	sgr-0a24b7294172a10a6	3306	TCP	0.0.0.0/0	default ↗
-	sgr-018aeb5ad314e51f4	80	TCP	0.0.0.0/0	default ↗
-	sgr-00732a74db38660c2	Todos	Todos	sg-00750ce86a5dd273e ↗	default ↗
-	sgr-0f1e3a08a9879e203	5000	TCP	0.0.0.0/0	default ↗

Instâncias do EC2 (mywebappi2/redes)

Detalhes | Status e alarmes [Novo](#) | Monitoramento | Segurança | **Redes** | Armazenamento | Tags

▼ Detalhes da rede [Informações](#)

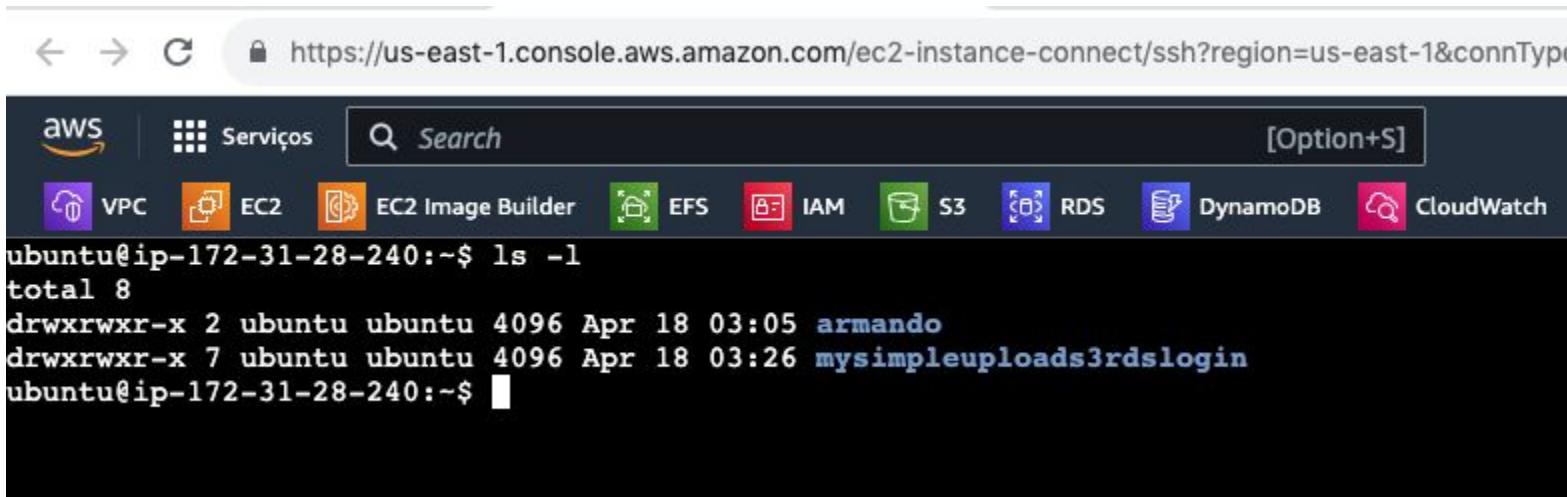
Endereço IPv4 público  54.145.9.153 endereço aberto 	Endereços IPv4 privados  172.31.30.184	ID da VPC  vpc-0e1426c149056cec2 
DNS IPv4 público  ec2-54-145-9-153.compute-1.amazonaws.com endereço aberto 	Nome do DNS de IP privado (somente IPv4)  ip-172-31-30-184.ec2.internal	
ID da sub-rede  subnet-04dbbf803db26da23 	Endereços IPv6 -	Endereços IPv4 privados secundários -
Zona de disponibilidade  us-east-1c	Endereços IP da operadora (temporários) -	ID do outpost -
Usar RBN como nome de host do sistema operacional convidado  Desabilitado	Responder IPv4 do nome de host DNS de RBN  Habilitado	

▼ Interfaces de rede (1) [Informações](#)

Instâncias do EC2 (mywebappi2/armazenamento)

Detalhes	Status e alarmes <small>Novo</small>	Monitoramento	Segurança	Redes	Armazenamento	Tags
▼ Detalhes de dispositivo raiz						
Nome de dispositivo raiz  /dev/sda1	Tipo de dispositivo raiz EBS	Otimização para o EBS desativado				
▼ Dispositivo de blocos						
<input type="text"/> Filtrar dispositivos de blocos						
ID de volume	Nome do dispositivo	Tamanho do volume	Status da associação	Hora da associação	Criptografia	ID da chave do KMS
vol-00dbecf261917b5ab	/dev/sda1	25	 Associado	2024/04/18 00:35 GMT-3	Não	-

Acesso à instância mywebapp1

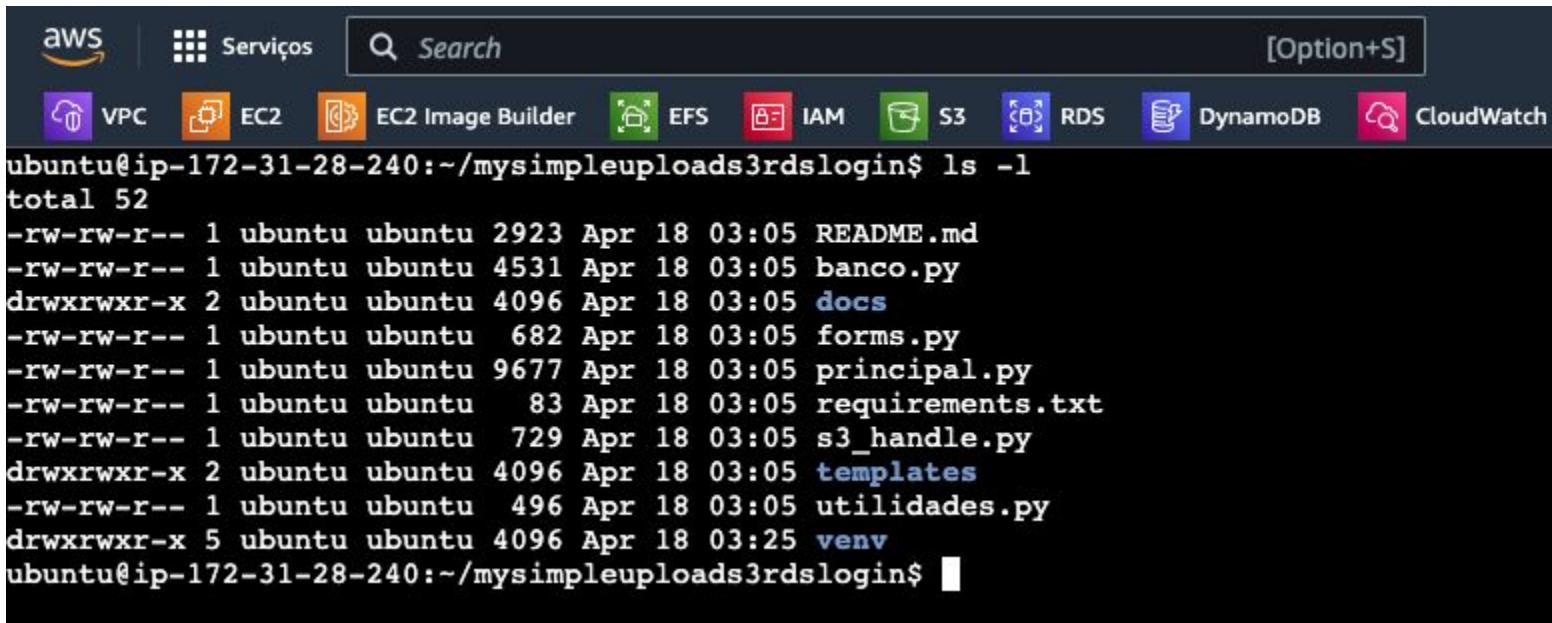


The screenshot shows a terminal window on an AWS EC2 instance. The URL in the browser bar is <https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=terminal>. The AWS navigation bar includes links for VPC, EC2, EC2 Image Builder, EFS, IAM, S3, RDS, DynamoDB, and CloudWatch. The terminal command history shows:

```
ubuntu@ip-172-31-28-240:~$ ls -l
total 8
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 18 03:05 armando
drwxrwxr-x 7 ubuntu ubuntu 4096 Apr 18 03:26 mysimpleuploads3rdslogin
ubuntu@ip-172-31-28-240:~$ █
```

<https://github.com/armandossrecife/mysimpleuploads3rdslogin>

Acesso à instância mywebapp1



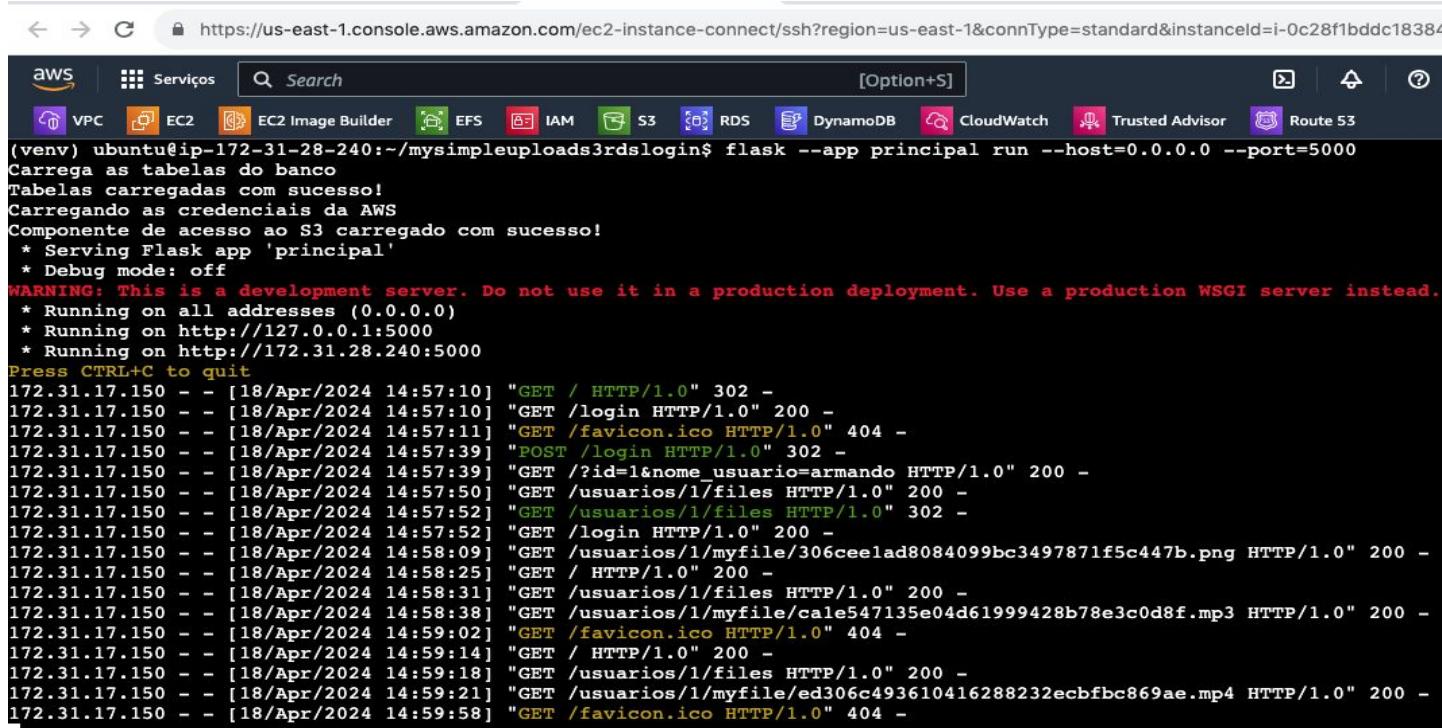
The screenshot shows the AWS Management Console interface with the EC2 service selected. A terminal window is open, displaying a Linux command-line session. The user is logged in as 'ubuntu' on an EC2 instance with IP 'ip-172-31-28-240'. The terminal shows the output of the 'ls -l' command, listing several files including 'README.md', 'banco.py', 'docs', 'forms.py', 'principal.py', 'requirements.txt', 's3_handle.py', 'templates', 'utilidades.py', and 'venv'. The session ends with the command 'exit'.

```
aws | Serviços | Search | [Option+S]
VPC EC2 EC2 Image Builder EFS IAM S3 RDS DynamoDB CloudWatch

ubuntu@ip-172-31-28-240:~/mysimpleuploads3rdslogin$ ls -l
total 52
-rw-rw-r-- 1 ubuntu ubuntu 2923 Apr 18 03:05 README.md
-rw-rw-r-- 1 ubuntu ubuntu 4531 Apr 18 03:05 banco.py
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 18 03:05 docs
-rw-rw-r-- 1 ubuntu ubuntu 682 Apr 18 03:05 forms.py
-rw-rw-r-- 1 ubuntu ubuntu 9677 Apr 18 03:05 principal.py
-rw-rw-r-- 1 ubuntu ubuntu 83 Apr 18 03:05 requirements.txt
-rw-rw-r-- 1 ubuntu ubuntu 729 Apr 18 03:05 s3_handle.py
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 18 03:05 templates
-rw-rw-r-- 1 ubuntu ubuntu 496 Apr 18 03:05 utilidades.py
drwxrwxr-x 5 ubuntu ubuntu 4096 Apr 18 03:25 venv
ubuntu@ip-172-31-28-240:~/mysimpleuploads3rdslogin$ exit
```

<https://github.com/armandossrecife/mysimpleuploads3rdslogin>

Executa a webapp na instância 1

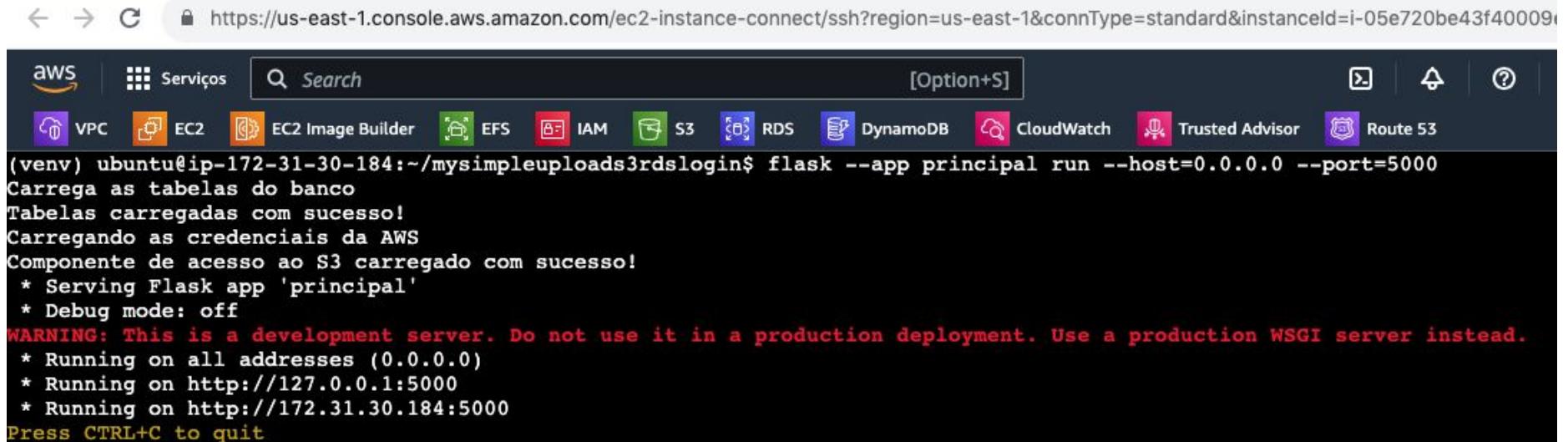


The screenshot shows a terminal window within the AWS CloudWatch interface. The URL is https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-0c28f1bddc18384. The terminal session is titled '(venv) ubuntu@ip-172-31-28-240:~/.mysimpleuploads3rdslogin\$'. The user runs the command 'flask --app principal run --host=0.0.0.0 --port=5000'. The output shows the application starting up, including database schema creation ('Carrega as tabelas do banco'), loading credentials ('Carregando as credenciais da AWS'), and connecting to S3 ('Componente de acesso ao S3 carregado com sucesso!'). It then serves the application ('* Serving Flask app 'principal'') and indicates debug mode is off ('* Debug mode: off'). A warning is displayed: 'WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.' The log concludes with a series of HTTP requests from the same IP address (172.31.17.150) for various files like favicon.ico and user profile pages.

```
(venv) ubuntu@ip-172-31-28-240:~/.mysimpleuploads3rdslogin$ flask --app principal run --host=0.0.0.0 --port=5000
Carrega as tabelas do banco
Tabelas carregadas com sucesso!
Carregando as credenciais da AWS
Componente de acesso ao S3 carregado com sucesso!
* Serving Flask app 'principal'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.31.28.240:5000
Press CTRL+C to quit
172.31.17.150 - - [18/Apr/2024 14:57:10] "GET / HTTP/1.0" 302 -
172.31.17.150 - - [18/Apr/2024 14:57:10] "GET /login HTTP/1.0" 200 -
172.31.17.150 - - [18/Apr/2024 14:57:11] "GET /favicon.ico HTTP/1.0" 404 -
172.31.17.150 - - [18/Apr/2024 14:57:39] "POST /login HTTP/1.0" 302 -
172.31.17.150 - - [18/Apr/2024 14:57:39] "GET /?id=1&name_usuario=armando HTTP/1.0" 200 -
172.31.17.150 - - [18/Apr/2024 14:57:50] "GET /usuarios/1/files HTTP/1.0" 200 -
172.31.17.150 - - [18/Apr/2024 14:57:52] "GET /usuarios/1/files HTTP/1.0" 302 -
172.31.17.150 - - [18/Apr/2024 14:57:52] "GET /login HTTP/1.0" 200 -
172.31.17.150 - - [18/Apr/2024 14:58:09] "GET /usuarios/1/myfile/306ceelad8084099bc3497871f5c447b.png HTTP/1.0" 200 -
172.31.17.150 - - [18/Apr/2024 14:58:25] "GET / HTTP/1.0" 200 -
172.31.17.150 - - [18/Apr/2024 14:58:31] "GET /usuarios/1/files HTTP/1.0" 200 -
172.31.17.150 - - [18/Apr/2024 14:58:38] "GET /usuarios/1/myfile/cales547135e04d61999428b78e3c0d8f.mp3 HTTP/1.0" 200 -
172.31.17.150 - - [18/Apr/2024 14:59:02] "GET /favicon.ico HTTP/1.0" 404 -
172.31.17.150 - - [18/Apr/2024 14:59:14] "GET / HTTP/1.0" 200 -
172.31.17.150 - - [18/Apr/2024 14:59:18] "GET /usuarios/1/files HTTP/1.0" 200 -
172.31.17.150 - - [18/Apr/2024 14:59:21] "GET /usuarios/1/myfile/ed306c493610416288232ecbfbc869ae.mp4 HTTP/1.0" 200 -
172.31.17.150 - - [18/Apr/2024 14:59:58] "GET /favicon.ico HTTP/1.0" 404 -
```

<https://github.com/armandossrecife/mysimpleuploads3rdslogin>

Executa a webapp na instância 2



A screenshot of an AWS CloudShell terminal window. The title bar shows the URL: https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-05e720be43f40009. The AWS logo is in the top-left, followed by 'Serviço' and a search bar with placeholder 'Search [Option+S]'. Below the search bar are icons for various AWS services: VPC, EC2, EC2 Image Builder, EFS, IAM, S3, RDS, DynamoDB, CloudWatch, Trusted Advisor, and Route 53. The main terminal area displays the following output:

```
(venv) ubuntu@ip-172-31-30-184:~/mysimpleuploads3rdslogin$ flask --app principal run --host=0.0.0.0 --port=5000
Carrega as tabelas do banco
Tabelas carregadas com sucesso!
Carregando as credenciais da AWS
Componente de acesso ao S3 carregado com sucesso!
 * Serving Flask app 'principal'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.31.30.184:5000
Press CTRL+C to quit
```

<https://github.com/armandossrecife/mysimpleuploads3rdslogin>