✅ Configuração AWS S3 - RecruitAI

📋 Resumo da Configuração

O AWS S3 foi configurado com sucesso para armazenar currículos de candidatos na plataforma RecruitAI.

🔑 Credenciais Configuradas

```
AWS_ACCESS_KEY_ID=AKIARJI3AIFWJPOTWNMN
AWS_SECRET_ACCESS_KEY=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
AWS_S3_REGION=us-east-2
AWS_S3_BUCKET_NAME=recruitai-resumes
AWS_S3_FOLDER_PREFIX=resumes/
```

🪣 Informações do Bucket

Propriedade	Valor
Nome do Bucket	`recruitai-resumes`
Região	`us-east-2` (Leste dos EUA - Ohio)
ARN	`arn:aws:s3:::recruitai-resumes`
Prefixo de Pasta	`resumes/`
URL de Login AWS	https://088640340332.signin.aws.amazon.com/console

---

✅ Testes Realizados

Teste 1: Verificação do Bucket

```
✅ Bucket encontrado em us-east-2
```

Teste 2: Upload e Download

```
✅ Upload realizado com sucesso!
✅ URL de download gerada com sucesso!
✅ Todos os testes foram bem-sucedidos!
```

📝 Estrutura de Armazenamento

Os currículos são armazenados na seguinte estrutura:

```
recruitai-resumes/
└── resumes/
    └── uploads/
        └── {timestamp}-{nome-original-do-arquivo}
```

Exemplo:

```
resumes/uploads/1764101550100-curriculo-joao-silva.pdf
```

🔐 Segurança

Permissões do Usuário IAM

O usuário IAM `recruitai-s3-user` tem as seguintes permissões:
- ✅ **AmazonS3FullAccess** - Acesso completo ao S3
- ✅ **Política de Bucket** - Acesso configurado via CORS

Configuração CORS

O bucket está configurado com CORS para permitir:
- `AllowedMethods`: GET, PUT, POST, DELETE
- `AllowedOrigins`: * (todos)
- `AllowedHeaders`: * (todos)
- `ExposeHeaders`: ETag

🔄 Como Funciona na Aplicação

1. Upload de Currículo

Quando um candidato faz upload do currículo:

1. O arquivo é enviado para `/api/candidates/upload-resume`
2. O backend valida o arquivo (tipo e tamanho)
3. O arquivo é convertido para Buffer
4. A função `uploadFile()` envia para o S3
5. A chave S3 (caminho) é salva no banco de dados

```
const resumeUrl = await uploadFile(buffer, fileName);
// Retorna: "resumes/uploads/1764101550100-curriculo.pdf"
```

2. Download de Currículo

Quando uma empresa quer baixar o currículo:

1. O sistema busca a chave S3 do banco de dados
2. A função `downloadFile()` gera uma URL assinada (válida por 1 hora)
3. O usuário é redirecionado para a URL

```
const downloadUrl = await downloadFile(resumeUrl);
// Retorna uma URL assinada válida por 3600 segundos
```

📦 Arquivos Modificados

1. `/nextjs_space/lib/aws-config.ts`

Configurado para:
- Usar novas variáveis de ambiente (`AWS_S3_*`)
- Priorizar credenciais explícitas sobre AWS_PROFILE
- Criar cliente S3 com região e credenciais corretas

2. `/nextjs_space/lib/s3.ts`

Modificado para:
- Criar cliente S3 de forma lazy (apenas quando necessário)
- Obter configurações dinamicamente
- Evitar conflitos de inicialização

3. `/nextjs_space/.env`

Adicionadas as variáveis:

```
AWS_ACCESS_KEY_ID=AKIARJI3AIFWJPOTWNMN
AWS_SECRET_ACCESS_KEY=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
AWS_S3_REGION=us-east-2
AWS_S3_BUCKET_NAME=recruitai-resumes
AWS_S3_FOLDER_PREFIX=resumes/
```

🚀 Deploy no Vercel

Quando fizer o deploy no Vercel, adicione as seguintes variáveis de ambiente:

Via Vercel Dashboard:

1. Acesse: https://vercel.com/seu-projeto/settings/environment-variables
2. Adicione cada variável:

Nome	Valor
`AWS_ACCESS_KEY_ID`	`AKIARJI3AIFWJPOTWNMN`
`AWS_SECRET_ACCESS_KEY`	`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`
`AWS_S3_REGION`	`us-east-2`
`AWS_S3_BUCKET_NAME`	`recruitai-resumes`
`AWS_S3_FOLDER_PREFIX`	`resumes/`

Via Vercel CLI:

```
vercel env add AWS_ACCESS_KEY_ID
vercel env add AWS_SECRET_ACCESS_KEY
vercel env add AWS_S3_REGION
vercel env add AWS_S3_BUCKET_NAME
vercel env add AWS_S3_FOLDER_PREFIX
```

💰 Custos Estimados

Free Tier (Primeiro Ano):

- ✅ 5 GB de armazenamento
- ✅ 20.000 solicitações GET
- ✅ 2.000 solicitações PUT

Após Free Tier:

Serviço	Custo
Armazenamento	$0.023/GB/mês
Upload (PUT)	$0.005 por 1.000 uploads
Download (GET)	$0.004 por 1.000 downloads

Exemplo com 1.000 Currículos:

Assumindo 2 MB por currículo:

```
Espaço: 2 GB = $0.046/mês
Uploads: 1.000 = $0.005
Downloads: 5.000 = $0.020
-----------------------------------
Total: ~$0.071/mês (R$ 0,35/mês)
```

🔧 Scripts de Teste

Teste Completo do S3:

```
cd /home/ubuntu/ats_platform/nextjs_space
yarn tsx scripts/test-s3.ts
```

Resultado Esperado:

```
🔧 Testando conexão com AWS S3...

📋 Variáveis de Ambiente:
  AWS_ACCESS_KEY_ID: ✅ Configurado
  AWS_SECRET_ACCESS_KEY: ✅ Configurado
  AWS_S3_REGION: us-east-2
  AWS_S3_BUCKET_NAME: recruitai-resumes
  AWS_S3_FOLDER_PREFIX: resumes/

🔌 Criando cliente S3...
✅ Cliente S3 criado com sucesso!

🪣 Configuração do Bucket:
  Bucket Name: recruitai-resumes
  Folder Prefix: resumes/

📤 Testando upload de arquivo...
✅ Upload realizado com sucesso!
  S3 Key: resumes/uploads/1764101550100-test-1764101550099.txt

📥 Testando download de arquivo...
✅ URL de download gerada com sucesso!

🎉 Todos os testes foram bem-sucedidos!

✅ Seu AWS S3 está configurado corretamente!
```

🆘 Troubleshooting

Erro: “Credentials are required”

Causa: Variáveis de ambiente não configuradas

Solução:

```
# Verifique se as variáveis estão no .env
cat .env | grep AWS_
```

Erro: “Access Denied”

Causa: Usuário IAM sem permissões adequadas

Solução:
1. Acesse AWS Console → IAM

2. Vá em “Usuários” → `recruitai-s3-user`
3. Verifique se tem a política `AmazonS3FullAccess`

Erro: “Bucket not found”

Causa: Nome do bucket incorreto ou região errada

Solução:

```
# Verifique o nome do bucket
echo $AWS_S3_BUCKET_NAME

# Verifique a região
echo $AWS_S3_REGION
```

Erro: “CORS policy”

Causa: Configuração CORS não aplicada

Solução:
1. Acesse AWS Console → S3
2. Selecione o bucket `recruitai-resumes`
3. Vá em “Permissões” → “CORS”
4. Adicione a configuração CORS conforme documentado

✅ Status Final

- ✅ AWS S3 configurado e testado
- ✅ Upload de arquivos funcionando
- ✅ Download de URLs assinadas funcionando
- ✅ Permissões IAM corretas
- ✅ Configuração CORS aplicada
- ✅ Integração com aplicação Next.js completa

📚 Próximos Passos

1. ✅ **Deploy no Vercel**
   - Adicionar variáveis de ambiente no Vercel
   - Fazer deploy da aplicação

2. ✅ **Testar em Produção**
   - Upload de currículo real
   - Download de currículo pela empresa
   - Verificar logs do AWS CloudWatch

3. ✅ **Monitoramento**
   - Configurar alarmes no CloudWatch
   - Monitorar uso e custos no AWS Cost Explorer

4. ✅ **Backup**
   - Configurar versionamento no bucket (opcional)
   - Configurar lifecycle policies para arquivos antigos

Documentação criada em: 25 de novembro de 2024
Versão: 1.0
Status: ✅ CONFIGURADO E TESTADO