



PROJETO VIDEO ENCODING

Implantação de Arquitetura de Encoding e Streaming de Vídeo na AWS

Fabio Hara
fabioh@microsoft.com

Índice

Documentação Técnica: Implantação de Arquitetura de Encoding e Streaming de Vídeo na AWS.....	3
TL;DR	3
1. Visão Geral da Arquitetura	3
Características Principais	3
2. Topologia de Rede e VPC	3
2.1 Configuração da VPC.....	3
2.2 Arquitetura de Sub-redes	4
2.3 Componentes de Conectividade.....	4
2.4 Tabelas de Rotas	4
3. Componentes de Computação (EC2)	5
3.1 Application Load Balancer (ALB)	5
3.2 Proxies (Camada de Proxy Reverso)	5
3.3 Application Servers	6
3.4 HTTP Workers (Encoding/Transcoding)	8
4. Serviços Gerenciados AWS	10
4.1 Amazon S3 (Armazenamento de Objetos)	10
4.2 Amazon DynamoDB (Banco NoSQL)	11
4.3 Amazon CloudFront (CDN).....	12
4.4 Amazon API Gateway.....	13
4.5 Sistema de Arquivos (Amazon EFS - Opcional)	15
5. Fluxos de Comunicação.....	16
5.1 Fluxo de Upload	16
5.2 Fluxo de Encoding	16
5.3 Fluxo de Content Delivery.....	17
6. Segurança.....	18
6.1 Security Groups - Resumo.....	18
6.2 Network ACLs (NACLs)	18
6.3 Encryption	18
6.4 Monitoramento e Auditoria.....	18

7. Alta Disponibilidade e Escalabilidade	19
7.1 Auto Scaling Groups.....	19
7.2 Multi-AZ Deployment.....	20
7.3 Disaster Recovery.....	20
8. Estimativas de Custo (sa-east-1).....	21
8.1 Custos Mensais Estimados.....	21
9. Implementação e Deployment	22
9.1 Ordem de Implantação Recomendada	22
9.2 Ferramentas de Infrastructure as Code.....	22
9.3 Checklist de Deployment	23
10. Otimizações e Boas Práticas	24
10.1 Performance	24
10.2 Custo	24
10.3 Segurança Avançada	25
10.4 Monitoramento Avançado.....	25
11. Alternativas e Considerações.....	26
11.1 AWS Elemental MediaConvert	26
11.2 Arquitetura Serverless Completa.....	27
12. Conclusão	27

Documentação Técnica: Implantação de Arquitetura de Encoding e Streaming de Vídeo na AWS

TL;DR

1. Visão Geral da Arquitetura

Esta documentação descreve a implantação de uma **arquitetura completa de processamento e distribuição de vídeo** na Amazon Web Services (AWS), utilizando a região **sa-east-1 (Brazil South)**. O sistema é projetado para receber uploads de arquivos de vídeo, processá-los através de pipelines de encoding/transcoding, armazenar os resultados e distribuí-los globalmente através de CDN.

Características Principais

- Alta Disponibilidade:** Implantação multi-AZ com redundância em todos os componentes críticos
 - Escalabilidade:** Auto Scaling em camadas de aplicação e workers de encoding
 - Segurança:** Segregação de redes, credenciais temporárias, IAM roles com menor privilégio
 - Performance:** Uso de instâncias otimizadas por workload e CDN global
-

2. Topologia de Rede e VPC

2.1 Configuração da VPC

Especificações da VPC Principal:

Parâmetro	Valor Recomendado
CIDR Block	10.0.0.0/16
Região	sa-east-1 (South America - São Paulo)

Availability Zones	sa-east-1a, sa-east-1b, sa-east-1c
DNS Hostnames	Habilitado
DNS Resolution	Habilitado

2.2 Arquitetura de Sub-redes

Distribuição de Sub-redes por Availability Zone:

Tipo	AZ	CIDR	Propósito
Public Subnet 1	sa-east-1a	10.0.1.0/24	ALB, NAT Gateway AZ-A
Public Subnet 2	sa-east-1b	10.0.2.0/24	ALB, NAT Gateway AZ-B
Public Subnet 3	sa-east-1c	10.0.3.0/24	ALB, NAT Gateway AZ-C
Private Subnet 1	sa-east-1a	10.0.11.0/24	Proxies, App Servers AZ-A
Private Subnet 2	sa-east-1b	10.0.12.0/24	Proxies, App Servers AZ-B
Private Subnet 3	sa-east-1c	10.0.13.0/24	Proxies, App Servers AZ-C
Private Subnet 4	sa-east-1a	10.0.21.0/24	HTTP Workers AZ-A
Private Subnet 5	sa-east-1b	10.0.22.0/24	HTTP Workers AZ-B
Private Subnet 6	sa-east-1c	10.0.23.0/24	HTTP Workers AZ-C

2.3 Componentes de Conectividade

Recomendação de NAT Gateway vs NAT Instance:

Utilize **NAT Gateways gerenciados pela AWS** em vez de NAT Instances, pois oferecem alta disponibilidade automática, escalabilidade até 100 Gbps e manutenção zero¹.

2.4 Tabelas de Rotas

Tabela de Rotas - Sub-redes Públicas:

Destino	Target	Descrição
10.0.0.0/16	local	Tráfego interno da VPC
0.0.0.0/0	igw-xxxxx	Tráfego para internet via Internet Gateway

Tabela de Rotas - Sub-redes Privadas (Aplicação):

Destino	Target	Descrição
10.0.0.0/16	local	Tráfego interno da VPC
0.0.0.0/0	nat-xxxxx	Tráfego para internet via NAT Gateway da mesma AZ

Tabela de Rotas - Sub-redes Privadas (Workers):

Destino	Target	Descrição
10.0.0.0/16	local	Tráfego interno da VPC
0.0.0.0/0	nat-xxxxx	Tráfego para internet via NAT Gateway
s3.sa-east-1.amazonaws.com	vpce-xxxxx	Acesso direto ao S3 via VPC Endpoint
dynamodb.sa-east-1.amazonaws.com	vpce-yyyyy	Acesso direto ao DynamoDB via VPC Endpoint

¹<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-comparison.html>

3. Componentes de Computação (EC2)

3.1 Application Load Balancer (ALB)

Configuração:

- **Tipo:** Application Load Balancer (Layer 7)
- **Scheme:** Internet-facing
- **Sub-redes:** Todas as sub-redes públicas (sa-east-1a, 1b, 1c)
- **Listeners:**
 - HTTP (porta 80) → Redirect para HTTPS
 - HTTPS (porta 443) → Target Group dos Proxies

Security Group do ALB (sg-alb):

Tipo	Protocolo	Porta	Origem	Descrição
Inbound	TCP	80	0.0.0.0/0	HTTP público
Inbound	TCP	443	0.0.0.0/0	HTTPS público
Outbound	TCP	8080	sg-proxy	Tráfego para proxies

3.2 Proxies (Camada de Proxy Reverso)

Especificações das Instâncias:

- **Família EC2:** T3 ou M7i-flex
- **Tipo Recomendado:** t3.medium ou m7i-flex.large (para workloads com tráfego moderado)
- **Quantidade:** Mínimo 2 instâncias (1 por AZ ativa), máximo conforme Auto Scaling
- **AMI:** Amazon Linux 2023
- **Software:** Nginx ou HAProxy como proxy reverso

Distribuição:

Instância	AZ	Sub-rede	Propósito
Proxy-1	sa-east-1a	Private Subnet 1 (10.0.11.0/24)	Proxy principal AZ-A
Proxy-2	sa-east-1b	Private Subnet 2 (10.0.12.0/24)	Proxy principal AZ-B

Security Group dos Proxies (sg-proxy):

Tipo	Protocolo	Porta	Origem	Descrição
Inbound	TCP	8080	sg-alb	Tráfego do ALB
Outbound	TCP	8080	sg-app-server	Tráfego para Application Servers
Outbound	TCP	443	0.0.0.0/0	HTTPS outbound

Configuração do Nginx (exemplo):

```
1 upstream app_servers {
2     server 10.0.11.10:8080;
```

```

3      server 10.0.12.10:8080;
4          keepalive 32;
5      }
6
7  server {
8      listen 8080;
9      location / {
10         proxy_pass http://app_servers;
11         proxy_set_header Host $host;
12         proxy_set_header X-Real-IP $remote_addr;
13         proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
14     }
15 }
```

3.3 Application Servers

Especificações das Instâncias:

- Família EC2:** M7i (General Purpose com desempenho balanceado)
- Tipo Recomendado:** m7i.xlarge ou m7i.2xlarge
- vCPUs/Memória:** 4 vCPUs / 16 GiB RAM (m7i.xlarge)
- Quantidade:** Mínimo 2 instâncias (1 por AZ), máximo via Auto Scaling
- AMI:** Amazon Linux 2023 ou Ubuntu 22.04 LTS
- Storage:** 100 GB EBS gp3

Distribuição:

Instância	AZ	Sub-rede	IP Privado (exemplo)
AppServer-1	sa-east-1a	Private Subnet 1	10.0.11.10
AppServer-2	sa-east-1b	Private Subnet 2	10.0.12.10

Responsabilidades:

- Receber requisições dos proxies
- Gerar requisições de encoding (encode requests) para HTTP Workers
- Processar notificações de encoding (encode notifications) dos Workers
- Atualizar metadados no DynamoDB
- Comunicar-se com serviços de gerenciamento
- Orquestrar workflows de processamento de vídeo

Security Group dos Application Servers (sg-app):

Tipo	Protocolo	Porta	Origem	Descrição
Inbound	TCP	8080	sg-proxy	Tráfego dos proxies
Inbound	TCP	9090	sg-workers	Notificações dos workers

Outbound	TCP	443	0.0.0.0/0	HTTPS outbound (APIs, S3, etc.)
Outbound	TCP	8080	sg-workers	Requisições para workers

IAM Role para Application Servers:

```

1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": [
7                  "s3:GetObject",
8                  "s3:PutObject",
9                  "s3>ListBucket"
10             ],
11            "Resource": [
12                "arn:aws:s3::::video-upload-bucket-sa-east-1/*",
13                "arn:aws:s3::::video-output-bucket-sa-east-1/*"
14            ]
15        },
16        {
17            "Effect": "Allow",
18            "Action": [
19                "dynamodb:GetItem",
20                "dynamodb:PutItem",
21                "dynamodb:UpdateItem",
22                "dynamodb:Query"
23            ],
24            "Resource": "arn:aws:dynamodb:sa-east-1:*:table/video-
jobs-metadata"
25        },
26        {
27            "Effect": "Allow",
28            "Action": [
29                "sts:AssumeRole"
30            ],
31            "Resource": "arn:aws:iam::*:role/video-upload-
credentials-role"
32        }
33    ]
34 }
```

Este IAM Role segue o princípio de **menor privilégio**, concedendo apenas as permissões necessárias para as operações específicas.

3.4 HTTP Workers (Encoding/Transcoding)

Especificações das Instâncias:

- **Família EC2:** C7i (Compute Optimized para encoding intensivo)²
- **Tipo Recomendado:** c7i.4xlarge ou c7i.8xlarge^{3 4}
- **vCPUs/Memória:** 16 vCPUs / 32 GiB RAM (c7i.4xlarge) ou 32 vCPUs / 64 GiB RAM (c7i.8xlarge)
- **Justificativa:** Instâncias compute-optimized são ideais para workloads de transcoding/encoding de vídeo, oferecendo maior frequência de CPU e melhor custo-benefício para tarefas CPU-intensive^{5 6}
- **Storage:** 500 GB EBS gp3 (ou instâncias com NVMe local para melhor I/O)
- **Quantidade:** Auto Scaling baseado em filas (mínimo 2, máximo conforme demanda)

Distribuição:

Instância	AZ	Sub-rede	Propósito
Worker-1	sa-east-1a	Private Subnet 4	Encoding worker AZ-A
Worker-2	sa-east-1b	Private Subnet 5	Encoding worker AZ-B
Worker-N	sa-east-1c	Private Subnet 6	Workers adicionais conforme escala

Responsabilidades:

- Receber requisições de encoding dos Application Servers
- Baixar arquivos de vídeo source do S3
- Executar transcodificação/encoding (H.264, H.265, múltiplas resoluções)
- Fazer upload dos vídeos processados para S3 (bucket de output)
- Enviar notificações de conclusão para Application Servers
- Atualizar status no DynamoDB

Security Group dos Workers (sg-workers):

Tipo	Protocolo	Porta	Origem	Descrição
Inbound	TCP	8080	sg-app	Requisições de encoding dos App Servers
Outbound	TCP	443	0.0.0.0/0	HTTPS para S3/APIs

²<https://aws.amazon.com/ec2/instance-types/compute-optimized/>

³<https://aws.amazon.com/ec2/instance-types/compute-optimized/>

⁴<https://www.ittiam.com/making-optimal-deployment-choices-encoding-system-aws/>

⁵<https://aws.amazon.com/ec2/instance-types/compute-optimized/>

⁶<https://www.ittiam.com/making-optimal-deployment-choices-encoding-system-aws/>

Outbound	TCP	9090	sg-app	Notificações para App Servers
-----------------	-----	------	--------	-------------------------------

IAM Role para Workers:

```

1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": [
7                  "s3:GetObject"
8              ],
9              "Resource": "arn:aws:s3:::video-upload-bucket-sa-east-
1/*"
10         },
11         {
12             "Effect": "Allow",
13             "Action": [
14                 "s3:PutObject",
15                 "s3:PutObjectAcl"
16             ],
17             "Resource": "arn:aws:s3:::video-output-bucket-sa-east-
1/*"
18         },
19         {
20             "Effect": "Allow",
21             "Action": [
22                 "dynamodb:UpdateItem",
23                 "dynamodb:PutItem"
24             ],
25             "Resource": "arn:aws:dynamodb:sa-east-1:*:table/video-
jobs-metadata"
26         }
27     ]
28 }
```

Considerações de Encoding:

Para workloads de encoding de vídeo, a escolha entre instâncias C7i e uso de **AWS Elemental MediaConvert** deve ser considerada:

- **EC2 C7i:** Maior controle, customização de codecs, menor custo em larga escala⁷

⁷<https://www.ittiam.com/making-optimal-deployment-choices-encoding-system-aws/>

- **MediaConvert**: Serviço gerenciado, sem infraestrutura, encoding acelerado até 25x, suporte nativo a formatos HLS/DASH/CMAF⁸⁹

Para esta arquitetura, mantemos EC2 C7i para flexibilidade e controle total do pipeline de encoding.

4. Serviços Gerenciados AWS

4.1 Amazon S3 (Armazenamento de Objetos)

Bucket 1: video-upload-bucket-sa-east-1

Propriedade	Configuração
Região	sa-east-1
Versionamento	Habilitado
Encryption	SSE-S3 (AES-256)
Lifecycle Policy	Mover para S3 Glacier após 30 dias, deletar após 90 dias
Public Access	Bloqueado (uploads via credenciais temporárias)

Bucket Policy (Upload Bucket):

```

1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Principal": {
7                  "AWS": "arn:aws:iam::ACCOUNT_ID:role/video-upload-
credentials-role"
8              },
9              "Action": [
10                 "s3:PutObject",
11                 "s3:PutObjectAcl"
12             ],
13             "Resource": "arn:aws:s3:::video-upload-bucket-sa-east-
1/*"
14         }
15     ]

```

⁸<https://docs.aws.amazon.com/solutions/latest/video-on-demand-on-aws/encoding-options.html>

⁹<https://dev.to/sudoconsultants/how-to-leverage-aws-elemental-mediaconvert-for-scalable-video-processing-in-the-cloud-222k>

```
16 }
```

Bucket 2: video-output-bucket-sa-east-1

Propriedade	Configuração
Região	sa-east-1
Versionamento	Habilitado
Encryption	SSE-S3 (AES-256)
Lifecycle Policy	Mover para S3 Intelligent-Tiering após 90 dias
Public Access	Bloqueado (acesso via CloudFront OAI)
CORS	Habilitado para domínios autorizados

CORS Configuration (Output Bucket):

```
1 [
2   {
3     "AllowedHeaders": ["*"],
4     "AllowedMethods": ["GET", "HEAD"],
5     "AllowedOrigins": ["https://cdn.example.com",
6                        "https://app.example.com"],
7     "ExposeHeaders": ["ETag"],
8     "MaxAgeSeconds": 3600
9   }
10 ]
```

4.2 Amazon DynamoDB (Banco NoSQL)

Tabela: video-jobs-metadata

Atributo	Tipo	Descrição
jobId (PK)	String	ID único do job de encoding
uploadTimestamp	Number	Timestamp do upload (Sort Key)
status	String	NEW, PROCESSING, COMPLETED, FAILED
sourceFileKey	String	Chave S3 do arquivo fonte
outputFileKeys	List	Lista de chaves S3 dos arquivos processados
userId	String	ID do usuário que iniciou o upload
videoMetadata	Map	Resolução, codec, duração, bitrate
createdAt	Number	Timestamp de criação
updatedAt	Number	Timestamp de última atualização
errorMessage	String	Mensagem de erro (se aplicável)

Configuração:

- **Região:** sa-east-1
- **Billing Mode:** On-Demand (auto scaling automático)
- **Encryption:** AWS Managed Keys (KMS)
- **Point-in-Time Recovery:** Habilitado
- **Global Secondary Index (GSI):**
 - **GSI-1:** userId (PK) + uploadTimestamp (SK) → Para consultas por usuário

Exemplo de Item:

```
1  {
2      "jobId": "job-2026-01-28-abc123",
3      "uploadTimestamp": 1738051200,
4      "status": "COMPLETED",
5      "sourceFileKey": "uploads/user123/video-source.mp4",
6      "outputFileKeys": [
7          "output/user123/video-720p.mp4",
8          "output/user123/video-1080p.mp4",
9          "output/user123/video-hls/playlist.m3u8"
10     ],
11     "userId": "user123",
12     "videoMetadata": {
13         "resolution": "1920x1080",
14         "codec": "h264",
15         "duration": 125,
16         "bitrate": 5000000
17     },
18     "createdAt": 1738050000,
19     "updatedAt": 1738051200
20 }
```

4.3 Amazon CloudFront (CDN)

Distribuição CloudFront

Propriedade	Configuração
Origin	video-output-bucket-sa-east-1.s3.sa-east-1.amazonaws.com
Price Class	Use All Edge Locations (melhor performance global)
Alternate Domain Names	cdn.example.com
SSL/TLS Certificate	AWS Certificate Manager (ACM) certificate
Viewer Protocol Policy	Redirect HTTP to HTTPS
Allowed HTTP Methods	GET, HEAD, OPTIONS
Cache Policy	CachingOptimized (cache baseado em query strings)

Origin Access Identity (OAI):

Configure um OAI para que o CloudFront acesse o bucket S3 privado:

```
1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
```

```

6      "Principal": {
7          "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin
Access Identity XXXXXX"
8      },
9      "Action": "s3:GetObject",
10     "Resource": "arn:aws:s3:::video-output-bucket-sa-east-
11     /*"
12 }
13 }
```

Behaviors:

Path Pattern	Origin	Cache Behavior
*.mp4	S3 Output Bucket	Cache por 24h, compress automaticamente
*.m3u8	S3 Output Bucket	Cache por 5 minutos (playlists HLS)
*.ts	S3 Output Bucket	Cache por 24h (segmentos HLS)

4.4 Amazon API Gateway

API: Video Upload API

- **Tipo:** REST API (regional endpoint em sa-east-1)
- **Autenticação:** AWS IAM ou Amazon Cognito
- **Endpoints:**

Método	Path	Integração	Descrição
POST	/upload/request-credentials	Lambda Function	Gera credenciais temporárias para upload
GET	/jobs/{jobId}	Lambda → DynamoDB	Consulta status de job
GET	/jobs/user/{userId}	Lambda → DynamoDB	Lista jobs de um usuário

Endpoint: POST /upload/request-credentials

Request Body:

```

1  {
2      "userId": "user123",
3      "fileName": "my-video.mp4",
4      "fileSize": 524288000,
5      "contentType": "video/mp4"
6  }
```

Response:

```
1  {
```

```

2   "uploadUrl": "https://video-upload-bucket-sa-east-1.s3.sa-
east-1.amazonaws.com/",
3   "credentials": {
4     "accessKeyId": "ASIA...",
5     "secretAccessKey": "...",
6     "sessionToken": "...",
7     "expiration": "2026-01-28T10:11:47Z"
8   },
9   "fields": {
10    "key": "uploads/user123/my-video.mp4",
11    "bucket": "video-upload-bucket-sa-east-1"
12  },
13  "jobId": "job-2026-01-28-xyz789"
14 }

```

Lambda Function (Upload Credentials Issuer):

Esta função Lambda gera credenciais temporárias usando AWS STS (Security Token Service):

```

1 import boto3
2 import json
3 from datetime import datetime, timedelta
4
5 sts_client = boto3.client('sts')
6 dynamodb = boto3.resource('dynamodb')
7
8 def lambda_handler(event, context):
9     body = json.loads(event['body'])
10    user_id = body['userId']
11    file_name = body['fileName']
12
13    # Gerar credenciais temporárias com STS
14    response = sts_clientassume_role(
15        RoleArn='arn:aws:iam::ACCOUNT_ID:role/video-upload-
credentials-role',
16        RoleSessionName=f'upload-session-{user_id}',
17        DurationSeconds=3600,
18        Policy=json.dumps({
19            "Version": "2012-10-17",
20            "Statement": [
21                {"Effect": "Allow",
22                 "Action": ["s3:PutObject"]},
23             ]
24         })
25
26    return {
27        "statusCode": 200,
28        "body": response['Credentials'],
29        "headers": {}
30    }

```

```

23             "Resource": f"arn:aws:s3::::video-upload-
bucket-sa-east-1/uploads/{user_id}/*"
24         }]
25     })
26   )
27
28   # Criar entrada no DynamoDB
29   job_id = f"job-{datetime.now().strftime('%Y-%m-%d')}-
{user_id[:8]}"
30   table = dynamodb.Table('video-jobs-metadata')
31   table.put_item(Item={
32       'jobId': job_id,
33       'userId': user_id,
34       'status': 'NEW',
35       'sourceFileKey': f'uploads/{user_id}/{file_name}',
36       'createdAt': int(datetime.now().timestamp()),
37       'uploadTimestamp': int(datetime.now().timestamp())
38   })
39
40   return {
41       'statusCode': 200,
42       'body': json.dumps({
43           'uploadUrl': 'https://video-upload-bucket-sa-east-
1.s3.sa-east-1.amazonaws.com/',
44           'credentials': response['Credentials'],
45           'fields': {
46               'key': f'uploads/{user_id}/{file_name}',
47               'bucket': 'video-upload-bucket-sa-east-1'
48           },
49           'jobId': job_id
50       })
51   }

```

4.5 Sistema de Arquivos (Amazon EFS - Opcional)

Quando usar EFS:

- Compartilhamento de arquivos temporários entre workers
- Armazenamento de arquivos intermediários de encoding
- Cache de assets comuns (watermarks, overlays, templates)

Configuração:

Propriedade	Valor
Performance Mode	General Purpose
Throughput Mode	Bursting (ou Provisioned para alta demanda)
Encryption	Em repouso (KMS) e em trânsito (TLS)
Mount Targets	Um em cada AZ (sa-east-1a, 1b, 1c)

Mount nos Workers:

```
1 sudo mount -t efs -o tls fs-xxxxxx:/ /mnt/efs
```

5. Fluxos de Comunicação

5.1 Fluxo de Upload

Diagrama do Fluxo:

```

1 Cliente → API Gateway → Lambda (Credentials Issuer)
2                               ↓
3                               STS (AssumeRole)
4                               ↓
5           ← Retorna Credenciais Temporárias ←
6 Cliente → S3 Upload Bucket (upload direto)
7                               ↓
8           S3 Event Notification
9                               ↓
10      Lambda/SQS → Application Server

```

5.2 Fluxo de Encoding

Exemplo de Encode Request (JSON):

```

1  {
2      "jobId": "job-2026-01-28-abc123",
3      "sourceKey": "uploads/user123/video-source.mp4",
4      "outputBucket": "video-output-bucket-sa-east-1",
5      "profiles": [
6          {
7              "name": "720p",
8              "resolution": "1280x720",
9              "bitrate": 3000000,
10             "codec": "h264",
11             "outputKey": "output/user123/video-720p.mp4"
12         },
13         {
14             "name": "1080p",

```

```

15     "resolution": "1920x1080",
16     "bitrate": 5000000,
17     "codec": "h264",
18     "outputKey": "output/user123/video-1080p.mp4"
19   }
20 ],
21   "callbackUrl": "https://app-
server.internal:9090/encoding/callback"
22 }
```

Exemplo de Encode Notification (JSON):

```

1  {
2    "jobId": "job-2026-01-28-abc123",
3    "status": "COMPLETED",
4    "duration": 180,
5    "outputFiles": [
6      {
7        "profile": "720p",
8        "key": "output/user123/video-720p.mp4",
9        "size": 67108864,
10       "duration": 125
11     },
12     {
13       "profile": "1080p",
14       "key": "output/user123/video-1080p.mp4",
15       "size": 134217728,
16       "duration": 125
17     }
18   ],
19   "completedAt": "2026-01-28T09:15:30Z"
20 }
```

5.3 Fluxo de Content Delivery

Diagrama do Fluxo:

```

1  Usuário Final → CloudFront Edge (200+ PoPs globais)
2                      ↓ (cache miss)
3                      S3 Output Bucket (sa-east-1)
4                      ↓ (via OAI)
5                      CloudFront Edge (cacheia e serve)
6                      ↓
7                      Usuário Final (streaming)
```

6. Segurança

6.1 Security Groups - Resumo

6.2 Network ACLs (NACLs)

NACL para Sub-redes Públicas:

Regra	Tipo	Protocolo	Porta	Origem/Destino	Ação
100	Inbound	TCP	80	0.0.0.0/0	ALLOW
110	Inbound	TCP	443	0.0.0.0/0	ALLOW
120	Inbound	TCP	1024-65535	0.0.0.0/0	ALLOW (return traffic)
*	Inbound	ALL	ALL	0.0.0.0/0	DENY
100	Outbound	ALL	ALL	0.0.0.0/0	ALLOW

NACL para Sub-redes Privadas:

Regra	Tipo	Protocolo	Porta	Origem/Destino	Ação
100	Inbound	TCP	ALL	10.0.0.0/16	ALLOW
110	Inbound	TCP	1024-65535	0.0.0.0/0	ALLOW (return traffic)
*	Inbound	ALL	ALL	0.0.0.0/0	DENY
100	Outbound	ALL	ALL	0.0.0.0/0	ALLOW

6.3 Encryption

Em Trânsito:

- ALB → TLS 1.2+ com certificados ACM
- CloudFront → TLS 1.3 nas edges
- VPC → TLS para comunicação com APIs AWS
- EFS → TLS habilitado para montagens

Em Repouso:

- S3 → SSE-S3 (AES-256) em todos os buckets
- DynamoDB → AWS Managed KMS Keys
- EBS → Volumes encriptados com AWS KMS
- EFS → Encryption at rest com KMS

6.4 Monitoramento e Auditoria

AWS CloudWatch:

- Logs de Application Servers, Workers, Proxies
- Métricas customizadas: jobs/min, success rate, encoding time
- Alarmes: CPU > 80%, disk usage > 85%, failed jobs > 10%

AWS CloudTrail:

- Auditoria de todas as chamadas de API
- Logs de acesso aos buckets S3
- Mudanças em IAM roles e policies

VPC Flow Logs:

- Monitoramento de tráfego entre sub-redes
 - Detecção de anomalias de rede
 - Análise de padrões de comunicação
-

7. Alta Disponibilidade e Escalabilidade

7.1 Auto Scaling Groups

Auto Scaling Group - Proxies:

```
1  {
2    "AutoScalingGroupName": "proxy-asg",
3    "MinSize": 2,
4    "MaxSize": 10,
5    "DesiredCapacity": 2,
6    "HealthCheckType": "ELB",
7    "HealthCheckGracePeriod": 300,
8    "VPCZoneIdentifier": "subnet-private-1a,subnet-private-
1b,subnet-private-1c",
9    "TargetGroupARNs": ["arn:aws:elasticloadbalancing:sa-east-
1:..."],
10   "ScalingPolicies": [
11     {
12       "PolicyName": "scale-up-cpu",
13       "MetricName": "CPUUtilization",
14       "Threshold": 70,
15       "ScalingAdjustment": 2
16     }
17   ]
18 }
```

Auto Scaling Group - HTTP Workers:

```
1  {
2    "AutoScalingGroupName": "workers-asg",
3    "MinSize": 2,
4    "MaxSize": 50,
```

```

5   "DesiredCapacity": 4,
6   "HealthCheckType": "EC2",
7   "HealthCheckGracePeriod": 600,
8   "VPCZoneIdentifier": "subnet-private-workers-1a,subnet-
private-workers-1b,subnet-private-workers-1c",
9   "ScalingPolicies": [
10    {
11      "PolicyName": "scale-up-queue-depth",
12      "MetricName": "ApproximateNumberOfMessagesVisible",
13      "Namespace": "AWS/SQS",
14      "Threshold": 10,
15      "ComparisonOperator": "GreaterThanOrEqualToThreshold",
16      "ScalingAdjustment": 5
17    },
18    {
19      "PolicyName": "scale-down-queue-depth",
20      "MetricName": "ApproximateNumberOfMessagesVisible",
21      "Namespace": "AWS/SQS",
22      "Threshold": 2,
23      "ComparisonOperator": "LessThanThreshold",
24      "ScalingAdjustment": -2
25    }
26  ]
27 }

```

7.2 Multi-AZ Deployment

Benefícios:

- Resiliência a Falhas:** Falha de uma AZ não afeta disponibilidade
- Latência Reduzida:** Distribuição geográfica dentro da região
- Manutenção Zero-Downtime:** Rolling updates sem interrupção

Distribuição por AZ:

Componente	sa-east-1a	sa-east-1b	sa-east-1c
NAT Gateway	✓	✓	✓
ALB Nodes	✓	✓	✓
Proxies	✓	✓	✓
App Servers	✓	✓	✓
Workers	✓	✓	✓

7.3 Disaster Recovery

Backup Strategy:

Componente	Frequência	Retenção	Método
DynamoDB	Contínuo	35 dias	Point-in-Time Recovery
S3 Buckets	Contínuo	Versionamento habilitado	S3 Versioning + Lifecycle
AMIs	Semanal	4 semanas	AWS Backup ou snapshots manuais
EBS Volumes	Diário	7 dias	EBS Snapshots automatizados

RTO/RPO:

- RTO (Recovery Time Objective): < 1 hora
 - RPO (Recovery Point Objective): < 5 minutos (via DynamoDB PITR e S3 versioning)
-

8. Estimativas de Custo (sa-east-1)

8.1 Custos Mensais Estimados

Detalhamento - Computação:

Instância	Tipo	Quantidade	Horas/mês	Custo/hora (sa-east-1)	Subtotal
Proxies	t3.medium	2	730	\$0.0416	\$60.74
App Servers	m7i.xlarge	2	730	\$0.192	\$280.32
Workers (base)	c7i.4xlarge	2	730	\$0.714	\$1,042.44
Workers (burst)	c7i.4xlarge	5 (média)	365 (50% do tempo)	\$0.714	\$1,303.05
Total EC2					\$2,686.55

Detalhamento - Armazenamento S3:

Item	Volume	Custo Unitário	Subtotal
S3 Standard (Upload)	2 TB	\$0.0245/GB	\$49.00
S3 Standard (Output)	8 TB	\$0.0245/GB	\$196.00
S3 PUT requests	100k	\$0.005/1k	\$0.50
S3 GET requests	10M	\$0.0004/1k	\$4.00
Total S3			\$249.50

Detalhamento - Rede:

Item	Volume	Custo Unitário	Subtotal
CloudFront (primeiros 10 TB)	10 TB	\$0.120/GB	\$1,200.00
NAT Gateway (3x)	3 gateways	\$0.045/hora	\$98.55
NAT Gateway data processing	5 TB	\$0.045/GB	\$225.00
ALB	730 horas	\$0.0225/hora	\$16.43
Total Rede			\$1,539.98

Detalhamento - Outros:

Serviço	Custo Mensal
---------	--------------

DynamoDB On-Demand	\$50.00
API Gateway (1M requests)	\$3.50
CloudWatch Logs	\$20.00
Route 53 (Hosted Zone)	\$0.50
Total Outros	\$74.00

Custo Total Estimado: \$4,550.03/mês

Nota: Valores aproximados com base em preços de 2026 para sa-east-1. Custos reais variam conforme uso.

9. Implementação e Deployment

9.1 Ordem de Implantação Recomendada

9.2 Ferramentas de Infrastructure as Code

Recomendações:

1. **AWS CloudFormation**: Nativo AWS, templates YAML/JSON, integração total
2. **Terraform (recomendado)**: Multi-cloud, state management, módulos reutilizáveis
3. **AWS CDK**: TypeScript/Python, abstrações de alto nível, síntese para CloudFormation

Exemplo Terraform - VPC Module:

```

1 module "vpc" {
2   source  = "terraform-aws-modules/vpc/aws"
3   version = "~> 5.0"
4
5   name    = "video-processing-vpc"
6   cidr   = "10.0.0.0/16"
7
8   azs      = ["sa-east-1a", "sa-east-1b", "sa-east-1c"]
9   public_subnets = ["10.0.1.0/24", "10.0.2.0/24",
"10.0.3.0/24"]
10  private_subnets = [
11    "10.0.11.0/24", "10.0.12.0/24", "10.0.13.0/24",
12    "10.0.21.0/24", "10.0.22.0/24", "10.0.23.0/24"
13  ]
14
15  enable_nat_gateway  = true
16  single_nat_gateway = false
17  one_nat_gateway_per_az = true
18

```

```

19   enable_dns_hostnames = true
20   enable_dns_support   = true
21
22   enable_s3_endpoint      = true
23   enable_dynamodb_endpoint = true
24
25   tags = {
26     Environment = "production"
27     Project     = "video-encoding"
28   }
29 }
```

9.3 Checklist de Deployment

✓ Pré-Deployment:

- Conta AWS configurada em sa-east-1
- Service quotas verificados (EC2, VPC, S3)
- Domínios registrados e certificados SSL solicitados (ACM)
- Repositório Git para IaC criado
- Ambientes definidos (dev, staging, production)

✓ Durante Deployment:

- VPC e networking criados e validados
- Security groups e IAM roles aplicados
- S3 buckets e DynamoDB tables provisionados
- EC2 Launch Templates e AMIs customizadas
- Auto Scaling Groups com health checks
- ALB com target groups e listeners
- API Gateway e Lambda functions deployadas
- CloudFront distribution configurada

✓ Pós-Deployment:

- Testes de ponta-a-ponta (upload → encoding → delivery)
 - Monitoramento e alertas configurados
 - Logs agregados no CloudWatch
 - Documentação técnica atualizada
 - Runbooks para incidentes criados
 - Treinamento da equipe de operações
-

10. Otimizações e Boas Práticas

10.1 Performance

10

10.2 Custo

Estratégias de Redução:

1. **Spot Instances para Workers:** Redução de até 90% em custos de encoding¹¹
2. **S3 Intelligent-Tiering:** Move automaticamente objetos entre tiers baseado em acesso
3. **Savings Plans:** Compromisso de 1-3 anos para EC2 (até 72% de desconto)
4. **Reserved Capacity:** Para DynamoDB em produção com tráfego previsível
5. **CloudFront Reserved Capacity:** Para volumes de tráfego consistentes

Exemplo Spot Fleet para Workers:

```
1  {
2    "SpotFleetRequestConfig": {
3      "IamFleetRole": "arn:aws:iam::ACCOUNT_ID:role/spot-fleet-
4      role",
5      "AllocationStrategy": "lowestPrice",
6      "TargetCapacity": 10,
7      "LaunchTemplateConfigs": [
8        {
9          "LaunchTemplateSpecification": {
10            "LaunchTemplateId": "lt-worker-encoding",
11            "Version": "$Latest"
12          },
13          "Overrides": [
14            { "InstanceType": "c7i.4xlarge", "SubnetId": "
15            "subnet-workers-1a", "WeightedCapacity": 1 },
16            { "InstanceType": "c7i.8xlarge", "SubnetId": "
17            "subnet-workers-1b", "WeightedCapacity": 2 },
18            { "InstanceType": "c6i.4xlarge", "SubnetId": "
19            "subnet-workers-1c", "WeightedCapacity": 1 }
20          ]
21        }
22      }
23    }
```

¹⁰<https://www.ittiam.com/making-optimal-deployment-choices-encoding-system-aws/>

¹¹<https://www.ittiam.com/making-optimal-deployment-choices-encoding-system-aws/>

```
18      ]
19    }
20 }
```

10.3 Segurança Avançada

10.4 Monitoramento Avançado

CloudWatch Custom Metrics:

```
1 import boto3
2 from datetime import datetime
3
4 cloudwatch = boto3.client('cloudwatch')
5
6 # Métrica customizada: tempo de encoding
7 cloudwatch.put_metric_data(
8     Namespace='VideoProcessing',
9     MetricData=[
10         {
11             'MetricName': 'EncodingDuration',
12             'Dimensions': [
13                 {'Name': 'JobType', 'Value': '1080p'},
14                 {'Name': 'WorkerInstance', 'Value': 'i-
0abcd1234'}
15             ],
16             'Value': 180.5,
17             'Unit': 'Seconds',
18             'Timestamp': datetime.utcnow()
19         },
20         {
21             'MetricName': 'EncodingSuccessRate',
22             'Value': 98.5,
23             'Unit': 'Percent',
24             'Timestamp': datetime.utcnow()
25         }
26     ]
27 )
```

CloudWatch Alarms Essenciais:

Métrica	Threshold	Ação
ALB 5XX Errors	> 10 em 5 min	SNS notification para ops team

EC2 CPU Utilization	> 85% por 10 min	Scale up Auto Scaling Group
DynamoDB Throttled Requests	> 5 em 1 min	Increase provisioned capacity
S3 4XX Errors	> 50 em 5 min	SNS notification + Lambda investigation
Encoding Job Failures	> 5% em 15 min	Page on-call engineer

11. Alternativas e Considerações

11.1 AWS Elemental MediaConvert

Quando Considerar:

- Equipe pequena sem expertise em encoding
- Necessidade de formatos avançados (HLS, DASH, CMAF) out-of-the-box¹²
- Encoding acelerado (até 25x mais rápido)¹³
- Redução de complexidade operacional

Arquitetura Alternativa com MediaConvert:

```

1 Cliente → API Gateway → Lambda (generate presigned URL)
2 ↓
3 S3 Upload
4 ↓
5 S3 Event → Lambda → MediaConvert Job
6 ↓
7 S3 Output
8 ↓
9 CloudFront

```

Comparação EC2 C7i vs MediaConvert:

Aspecto	EC2 C7i Workers	AWS MediaConvert
Controle	Alto - codecs customizados	Médio - templates pré-definidos
Custo (alta escala)	Menor (Spot Instances)	Maior (pay-per-minute)
Manutenção	Alta - gerenciar infra	Baixa - fully managed
Time to Market	Mais longo	Rápido - horas para deploy

¹²<https://docs.aws.amazon.com/solutions/latest/video-on-demand-on-aws/encoding-options.html>

¹³<https://docs.aws.amazon.com/solutions/latest/video-on-demand-on-aws/encoding-options.html>

Performance	Dependente de instância ¹⁴	Aceleração automática ¹⁵
--------------------	---------------------------------------	-------------------------------------

11.2 Arquitetura Serverless Completa

Stack Serverless:

- **Compute:** Lambda (processar metadados), Fargate (workers containerizados)
- **Storage:** S3, DynamoDB
- **Encoding:** MediaConvert
- **API:** API Gateway + Lambda
- **CDN:** CloudFront

Benefícios:

- Zero gestão de servidores
- Pay-per-use verdadeiro
- Escala automática infinita
- Menor overhead operacional

Trade-offs:

- Menor controle sobre recursos
 - Cold start latency em Lambda
 - Custos podem ser maiores em cargas constantes
-

12. Conclusão

Esta documentação apresentou uma **arquitetura completa e production-ready** para processamento e distribuição de vídeo na AWS região Brazil South (sa-east-1). A solução combina:

- Alta Disponibilidade** através de deployment multi-AZ
- Escalabilidade** com Auto Scaling Groups e serviços gerenciados
- Segurança** com defense-in-depth, encryption everywhere e least privilege
- Performance** com instâncias otimizadas (C7i, M7i), VPC Endpoints e CloudFront
- Custo-Efetividade** através de Spot Instances, S3 Lifecycle e Reserved Capacity

¹⁴<https://www.ittiam.com/making-optimal-deployment-choices-encoding-system-aws/>

¹⁵<https://docs.aws.amazon.com/solutions/latest/video-on-demand-on-aws/encoding-options.html>

A arquitetura está pronta para suportar desde workloads iniciais até milhões de vídeos processados mensalmente, com capacidade de escalar horizontal e verticalmente conforme a demanda.

Próximos Passos Recomendados:

1. Implementar ambiente de desenvolvimento/staging para testes
 2. Criar pipelines CI/CD para deployment automatizado
 3. Desenvolver testes de carga e stress
 4. Estabelecer SLAs e SLIs para monitoramento
 5. Documentar runbooks para cenários de incident response
-

Documentação elaborada por: Arquiteto de Soluções Cloud

Data: 28 de Janeiro de 2026

Região: AWS sa-east-1 (South America - São Paulo)

Versão: 1.0