Confidential（秘密情報）

動画コンテンツシステム

インフラ構築手順書

第1.1版　2025年7月1日

株式会社amoibe

改定履歴

|  |  |  |  |
| --- | --- | --- | --- |
| 版数 | 日付 | 改定内容 | 項番・ページなど |
| 1.0 | 2025年7月1日 | 初版作成 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目次

内容

[1. 業務要件 3](#_Toc203135562)

[1-1. システム化の背景と目的 3](#_Toc203135563)

[1-2. システムの全体像 今回構築するシステムの全体構成図は以下である。 3](#_Toc203135564)

[1-3. システム化の範囲 3](#_Toc203135565)

[2. 機能・非機能要件 3](#_Toc203135566)

[2-1. 機能要件 4](#_Toc203135567)

[2-2. 非機能要件 5](#_Toc203135568)

[3. インフラ設計 6](#_Toc203135569)

[3-1. アーキテクチャ 6](#_Toc203135570)

[3-2. テクノロジースタック 6](#_Toc203135571)

[3-3. ソーステクノロジスタック 6](#_Toc203135572)

[3-4. 命名規則 6](#_Toc203135573)

[4. インフラ構築手順(APP) 8](#_Toc203135574)

[4-1. 前提条件 8](#_Toc203135575)

[4-2. 作業完了条件 8](#_Toc203135576)

[4-3. 作業者 8](#_Toc203135577)

[4-4. 作業場所 8](#_Toc203135578)

[4-5. 作業端末 8](#_Toc203135579)

[4-6. 制約事項・備考 8](#_Toc203135580)

[4-7. 作業実績 8](#_Toc203135581)

[4-8. 手順(APP) 9](#_Toc203135582)

[4-8-1. VPC 9](#_Toc203135583)

[4-8-2. EC2 9](#_Toc203135584)

[4-8-3. RDS 10](#_Toc203135585)

[4-8-4. ALB 10](#_Toc203135586)

[4-8-5. S3 10](#_Toc203135587)

[4-8-6. CloudFront 10](#_Toc203135588)

# 1. 業務要件

## 1-1. システム化の背景と目的

　現行のサービスは、Gigaスクール構想やDX化に伴う急速なコンテンツ拡充に対応できないことが課題となっている。

また、講師ごとに異なるスキルや経験に対応するカスタマイズ機能や、学習進捗の可視化機能の不足も指摘されている。

これを解決するため、クライアント企業は新たな動画コンテンツ配信システムの開発を決定した。

## 1-2. システムの全体像 今回構築するシステムの全体構成図は以下である。

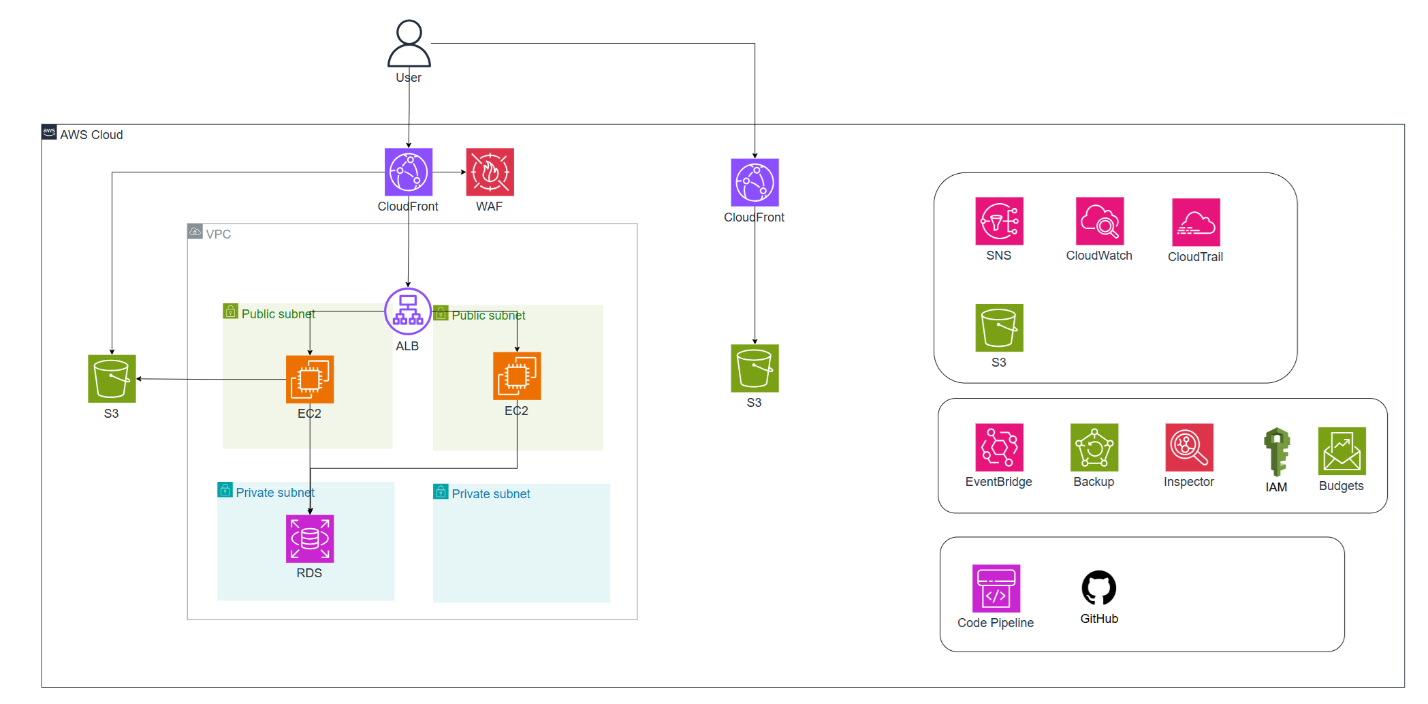


Fig.1 全体構成図

## 1-3. システム化の範囲

本システムの開発範囲は、動画コンテンツ全体のインフラ基盤を開発する。

# 2. 機能・非機能要件

動画コンテンツシステム新規開発プロジェクトの機能要件と非機能要件である。

## 2-1. 機能要件

1. 環境

* 開発環境のみ構築
* 開発環境内の構成
  + App（動画プラットフォーム用システム）
  + LP（動画プラットフォーム用の静的サイト）

2. サーバー（App）

* EC2 (Amazon Linux 2023)
* Javaバージョン: JDK 17
* フレームワーク: Spring Boot
* 必要なソフトウェア
  + Git
  + Maven
  + Gradle
  + Timezone 設定
  + PostgreSQL client
* IPアドレスを固定

3. データベース

* Amazon RDS (PostgreSQL 16)
* 初期データベース名: movie
* Timezone 設定

4. ストレージ

* S3 バケット
  + App: 動画保存用
  + LP: 静的ホスティング用（静的Webサイト）

5. 負荷分散

* ALB による負荷分散
* パスベースルーティング
  + /contents/\* → S3 へリダイレクト
  + その他のパス → ALB へ

6. 監視

* Appサービスの監視
  + エラー発生時にメール通知
  + 監視結果をダッシュボード化

7. CI/CD

* LP に対する CI/CD 環境の構築

8. IAM管理

* 開発メンバー以外の IAM 設定
  + 経理部門: コスト関連のサービスのみ許可
  + 経営管理部門: 管理者権限 (Admin)
  + IT部門: Cost と IAM 以外の権限

9. 監査

* ISO27001取得予定に対応
* AWSユーザー操作ログを S3 に保存

## 2-2. 非機能要件

1. 可用性

* 本番環境では負荷分散を想定
* 本番環境ではEC2 / RDS の AZ 冗長化
  + 開発環境ではシングル構成
* 動画コンテンツのキャッシュ化

2. セキュリティ

* ストレージの暗号化
* 脆弱性診断
  + EC2 → AWS Inspector
  + LP → OWASP ZAP

3. バックアップ

* EC2 / RDS のバックアップを 7 世代保存
* メンテナンスタイム: 0時 ~ 4時
  + 0時 ~ 2時は別システム連携があるため要考慮

4. コスト管理

* EC2 は平日日中のみ起動
* ランニングコストを 7 万円 / 月 以下に抑える
* コストが 7 万円 / 月 を超えた場合は通知

5. ドメイン

* ドメイン関連はヒアリング中のため、保留

6. AWS リソースの運用ルール

* 命名規則の設定・運用
* タグの命名規則の設定・運用

7. 監視

* 一般的な監視項目の設定
* 監視結果をメール通知
* 監視ダッシュボードの作成
* メールアドレスは未定のためインフラ担当者のメールを設定

# 3. インフラ設計

## 3-1. アーキテクチャ

AWSシステム構成図はFig.1に掲載しているため省略

## 3-2. テクノロジースタック

テクノロジースタックは以下である。

* インフラ: AWS
* アプリケーション: Java 17 + Spring Boot
* CI/CD: AWS CodePipeline
* 監視・運用: CloudWatch, Inspector, OWASP ZAP
* セキュリティ: IAM, 暗号化
* コスト管理: スケジューリング、予算超過通知

## 3-3. ソーステクノロジスタック

ソーステクノロジスタックは以下である。

* 言語: Java 17
* フレームワーク: Spring Boot
* データベースクライアント: PostgreSQL Client
* バージョン管理: Git
* Java 11
* Spring 2.7.15

## 3-4. 命名規則

リソースの命名規則は以下の通りです。

movie-{env}-{system}-{AWS-Service}-{optional-1}-{optional-2}

（例）

- EC2 Instance: `movie-dev-app-ec2`

- RDS: ` movie -dev-app-rds`

- Elastic IP: ` movie -dev-app-ec2-eip`

タグの命名規則は以下の通りです。

Table.1 タグ命名規則

|  |  |
| --- | --- |
| Key | Value |
| Cost | movie\_{name} |
| Name | Resource Name |
| Project | movie |
| env | dev |
| createdBy | {name} |

# 4. インフラ構築手順(APP)

## 4-1. 前提条件

* 作業者のAWSアカウントが発行されていること
* 適切なロールが適用されていること

## 4-2. 作業完了条件

* 全てのリソースの作成が完了していること
* アプリケーションが正しく動くこと

## 4-3. 作業者

* 〇〇 〇〇 (xxxxx@dotlife.co.jp)

## 4-4. 作業場所

* 〒153-0063 東京都目黒区目黒1-4-16

## 4-5. 作業端末

* Macbook Pro, 2020

## 4-6. 制約事項・備考

* 特になし

## 4-7. 作業実績

2025年4月20日 ～ 2025年7月5日

開発環境におけるすべてのリソースを構築完了。

## 4-8. 手順(APP)

### 4-8-1. VPC

1. AWS マネジメントコンソール にログイン
2. VPC サービス を開く
3. 左メニューの 「VPC を作成」 をクリック
4. VPCなどを選択



1. 設定の入力（詳細な設定はパラメータシートを参照）
2. 「VPCを作成」をクリック

正常性確認

* VPCなど必要なネットワーク設定ができていること

### 4-8-2. EC2

1. AWS マネジメントコンソール にログイン
2. EC2 サービス を開く
3. 「インスタンスの作成」ボタンをクリック
4. 設定の入力（詳細な設定はパラメータシートを参照）
5. 「起動」ボタンを押す
6. SSHにログインし、内部設定を実施（詳細な設定はパラメータシートを参照）

正常性確認

* EC2のステータスチェックが2/2であること
* EC2にSSHログイン可能であること
* 必要なソフトウェアがインストールされていること

### 4-8-3. RDS

1. AWS マネジメントコンソール にログイン
2. RDS サービス を開く
3. 「データベースの作成」ボタンをクリック
4. 設定の入力（詳細な設定はパラメータシートを参照）
5. 「データベースの作成」ボタンをクリック
6. 作成完了まで待機（15分程度）

正常性確認

* RDSが正常に起動していること
* EC2からRDSへログインできること

### 4-8-4. ALB

1. AWS マネジメントコンソール にログイン
2. EC2 サービス を開く
3. 左メニューの 「ロードバランサー」 を開き、「ロードバランサーを作成」ボタンをクリック
4. 設定の入力（詳細な設定はパラメータシートを参照）

正常性確認

* TGがhealthyであること
* ALBのDNS名でEC2の内容が表示されていること

### 4-8-5. S3

1. AWS マネジメントコンソール にログイン
2. S3 サービス を開く
3. 「バケットを作成」ボタンをクリック
4. 設定の入力（詳細な設定はパラメータシートを参照）
5. 「バケットを作成」ボタンをクリック

正常性確認

* バケットが正しく作成されていること

### 4-8-6. CloudFront

1. AWS マネジメントコンソール にログイン
2. CloudFront サービス を開く
3. 「ディストリビューションを作成」ボタンをクリック
4. 設定の入力（詳細な設定はパラメータシートを参照）
5. 「ディストリビューションを作成」ボタンをクリック

正常性確認

* ディストリビューション名にてEC2の表示内容が確認できること

以上