2.AWS構築手順書

🟦 AWS構築手順書

案件名：中小企業向けコーポレートサイトAWS移行プロジェクト

作成者：ふうま

作成日：2025/11/22

バージョン：v1.0

🟩 1. 目的・概要

このドキュメントは、AWS上に構築した3層構成（Web / AP / DB）の構築手順をまとめたもの。

再現性と構成意図を明確にし、将来的なIaC化（Terraform対応）を見据えた設計・実装メモを兼ねる。

🟨 2. 構築環境情報

項目	内容
リージョン	ap-northeast-1（東京）
AZ構成	1a / 1c
OS	Amazon Linux 2023
DBエンジン	MySQL 8.0
AWSアカウント	個人検証用
構築期間	2025/11/11 〜 2025/11/17
構築目的	オンプレ環境の3層クラウド化検証

🟧 3. 手順概要（章立て一覧）

1️⃣ ネットワーク（VPC / Subnet / Routing / NATGW）

2️⃣ セキュリティ（IAM / SG / キーペア）

3️⃣ コンピュート（EC2構築・設定）

4️⃣ データベース（RDS構築・接続）

5️⃣ ストレージ（S3バケット作成・連携）

6️⃣ ロードバランサ（ALB設定・ターゲット登録）

7️⃣ DNS・証明書（Route53 / ACM設定）

8️⃣ 監視（CloudWatch / SNS通知）

9️⃣ 動作確認・テスト

🔟 構築後の課題・改善点

🟩 4. 手順詳細

🧩 4.1 ネットワーク構築

目的：

AWS上でPublic / Private / DB層を分離し、安全で拡張性の高いVPCを設計する。

手順：

1. VPC作成（10.10.0.0/16）
2. Public Subnet作成（10.10.0.0/24, 10.10.1.0/24）
3. Private-App Subnet作成（10.10.10.0/24, 10.10.11.0/24）
4. Private-DB Subnet作成（10.10.20.0/24, 10.10.21.0/24）
5. IGW作成・VPCにアタッチ
6. NATGWを1aに作成しElastic IPを割り当て
7. Route Table設定（Public＝IGW、Private＝NATGW）

確認：

- ✅ 設定パラメータが正しいことを確認

🧩 4.2 セキュリティ設定

目的：

最小権限・ネットワーク分離を実現する。

手順：

1. IAMユーザー ***** 作成（MFA有効）
2. IAMロール EC2InstanceRole-Web 作成
   → ポリシー: AmazonS3ReadOnlyAccess, CloudWatchAgentServerPolicy
3. セキュリティグループ作成：
   - ALB-SG: 80/443 from 0.0.0.0/0
   - Web-SG: 80 from ALB-SG, 22 from 管理端末
   - DB-SG: 3306 from Web-SG

確認：

- ✅ 設定パラメータが正しいことを確認

🧩 4.3 RDS構築（DB層）

目的：

Web層と分離されたRDSでMySQLを安定運用。

手順：

1. RDS(MySQL)作成（db.t3.micro, Multi-AZ有効）
2. サブネットグループにPrivate-DBサブネットを指定
3. マルチAZ配置を設定
4. 自動バックアップ7日間を設定

確認：

- ✅ RDS起動
- ✅ マルチAZ構成あり
- ✅ バックアップが自動で作成されている

🧩 4.4 S3設定（ストレージ）

目的：

静的ファイル・バックアップをS3に保存する。

手順：

1. バケット名：***-*****
2. バージョニング有効化
3. ライフサイクル：90日後にIAへ移行
4. EC2ロールにS3アクセス権付与

確認：

- ✅ ファイルアップロード・取得成功

🧩 4.5 ALB構築（ロードバランサ）

目的：

冗長化・HTTPS通信の実現。

手順：

1. ALB作成（Public Subnet）
2. ターゲットグループにEC2を登録
3. ヘルスチェック設定（/index.php）

確認：

- ✅ 両方のEC2がHealthy状態

🧩 4.6 EC2構築（Web / AP層）

目的：

Nginx＋PHP環境を構築し、Webアプリを稼働させる。

手順：

1. EC2起動（AMI: Amazon Linux 2023, t3.small）
2. ユーザーデータで基本設定：
   - Nginx / PHP インストール
   - index.php配置
3. Nginx設定（/etc/nginx/conf.d/default.conf）
4. PHP-FPM設定（/etc/php-fpm.d/www.conf）
5. systemctl enable nginx php-fpm

確認：

- ✅ EC2起動
- ✅ ALBターゲット登録後、ヘルスチェックOK
- ✅ Webページ表示成功
- ✅ EC2からDB接続成功

🧩 4.7 Route53 / ACM設定

目的：

独自ドメインをALBに紐づけ、HTTPS通信を確立。

手順：

1. Route53 Public Hosted Zone作成
2. Aレコード作成（ALB DNSへAlias設定）
3. ACMで *.example.com 証明書発行
4. ALBでHTTPSリスナー追加
5. 証明書をALBに関連付け

確認：

- ✅ `https://example.com` でアクセス成功
- ✅ 証明書が有効（鍵マーク表示）

🧩 4.8 CloudWatch / SNS（監視・通知）

目的：

監視・ログ・通知を一元管理する。

手順：

1. CloudWatchアラーム作成（CPU使用率 > 80%）
2. SNSトピック作成（メール通知設定）

確認：

- ✅ インスタンス負荷テストにてアラーム発報
- ✅ メール通知受信成功

🧩 4.9 テスト・検証結果

テスト項目	結果	備考
ALB経由アクセス	✅ OK	HTTPS正常表示
DB接続	✅ OK	MySQL接続成功
S3アクセス	✅ OK	ファイルアップロード成功
CloudWatch監視	✅ OK	通知メール受信確認

🧩 4.10 課題・改善点

項目	内容	改善方針
NATGW単一構成	AZ障害時に通信不可	コスト次第で2台構成化
パッチ適用	手動対応	SSM Patch Manager導入
デプロイ方式	手動アップロード	CodePipeline導入
監視項目	最小限	RDS / ALBのメトリクスも追加予定

🟦 5. 添付資料

- 📸 スクリーンショット（ALB表示、RDS接続、SNS通知）
- 🗺️ 構成図（draw.io PNG）

- 💰 コスト試算（AWS Pricing Calculator）

🧠 6. メモ（学び・注意点）

- NATGW経由のルーティングに最初詰まった
- IAMロールのポリシー適用タイミングに注意（EC2再起動必要）
- ALB→EC2のヘルスチェック設定は `/` ではなく `/index.php` の方が安定

🏁 7. まとめ

この構築を通じて、AWSの基本構成と冗長化・監視・セキュリティ設計の一連を理解。

今後はIaC化（Terraform）とCI/CD導入に進む予定。
