DevOps Design — UI, API, Database & CI/CD

This document contains an AWS-focused architecture design for a web application.
It includes frontend hosting (S3 + CloudFront), backend services (ECS/Fargate behind ALB),
a relational database design (RDS PostgreSQL Multi-AZ + read replicas), and a CI/CD workflow using GitHub

Files included in the GitHub-ready package:
- README.md (this file)
- DevOps_Design_Diagram_and_Explanation.pdf (this document)

UI Architecture

Diagram (simplified)

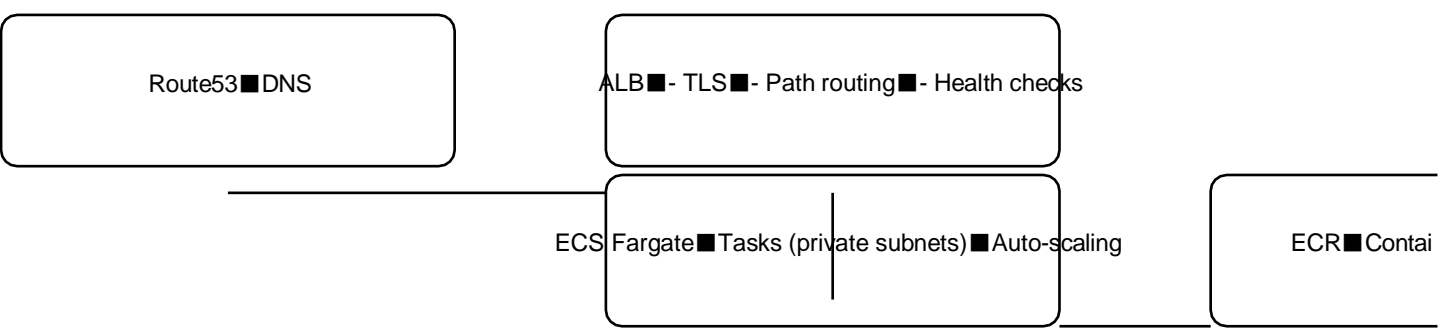| CloudFront (CDN)■- TLS■- WAF | S3 Origin■- Static assets■- Versioned builds |
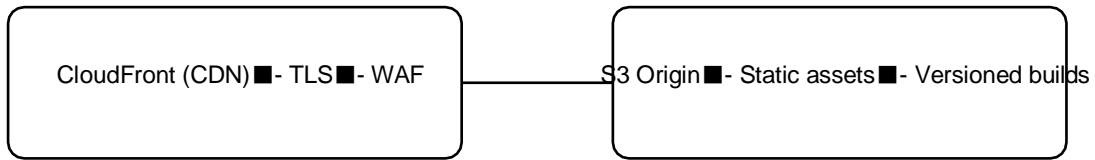
Edge cache -> Origin fetch

Notes:
- CloudFront provides global caching and TLS termination.
- Use Cache-Control, ETag, and versioned assets to optimize cache hit ratio.
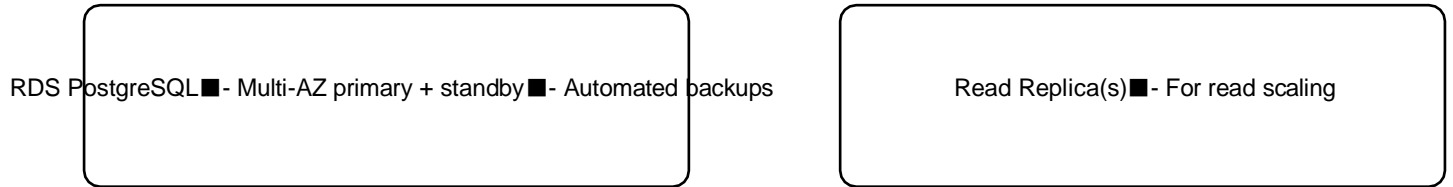- Use Lambda@Edge for small customizations (redirects, security headers).

API Architecture

Diagram (simplified)

Route53■DNS

ALB■- TLS■- Path routing■- Health checks

ECS Fargate■Tasks (private subnets)■Auto-scaling

ECR■Contai

Notes:
- API is containerized and runs in private subnets; ALB in public subnets routes traffic.
- Use Task Roles and Secrets Manager for credentials.
- Auto-scaling based on target CPU/request latency; use CloudWatch metrics.

Database Architecture

RDS PostgreSQL■- Multi-AZ primary + standby■- Automated backups

Read Replica(s)■- For read scaling

- Backups: automated snapshots + PITR enabled
- Migrations: use versioned migration tool (Flyway/Liquibase); prefer backward-compatible changes
- Scale vertically by resizing instance; horizontally with read replicas or sharding if necessary

CI/CD Design & Operational Notes

GitHub Actions — Conceptual Workflow

1) Triggers: PRs (test), push to main (deploy to staging), protected merges to main (deploy to prod with approval).
2) Build: compile frontend; build backend container image; run linters.
3) Test: unit tests, integration/smoke tests against ephemeral or staging environment.
4) Deploy: upload frontend to S3 (invalidate CloudFront); push container to ECR and update ECS Service (blue/green or
5) Verify: health checks, synthetic tests, monitoring alarms for rollback triggers.

Operational considerations:
- Monitoring: CloudWatch dashboards, alarms, X-Ray tracing.
- Security: IAM least privilege, Secrets Manager, WAF, GuardDuty.
- DR: Cross-region snapshots for critical data; documented RTO/RPO.

Appendix — Deployment Checklist

[ ] Create VPC with public/private subnets across 2+ AZs
[ ] Configure S3 bucket for frontend + CloudFront distribution
[ ] Configure ALB, target groups, and ECS cluster (Fargate)
[ ] Provision RDS PostgreSQL with Multi-AZ and automated backups
[ ] Setup Secrets Manager and IAM roles for tasks
[ ] Create GitHub Actions workflows for build/test/deploy
[ ] Setup CloudWatch dashboards and alerts; enable X-Ray
[ ] Configure WAF rules and GuardDuty