FundifyHub — Cost-first AWS Deployment Guide

Multiple approaches, step-by-step console guidance, pros/cons, and cost estimates for 100/500/1000 monthly users

Generated: 7/11/2025, 10:22:44 pm

Overview

This document provides cost-first deployment approaches for FundifyHub on AWS. It lists four approaches: A) Single EC2 + RDS (lowest cost), B) ECS Fargate + RDS (managed containers), C) Serverless (Lambda + Aurora Serverless), and D) Hybrid (ECS + EC2 worker). Each approach contains pros/cons, a step-by-step console-focused walkthrough, pictorial mock panels highlighting important choices, and rough cost estimates for 100, 500 and 1000 monthly users.

Approach A — Single EC2 + RDS (Lowest ongoing cost)

Pros / Cons

Pros

- Lowest ongoing cost
- Simple to manage initially
- Easy to debug (SSH access)

Cons

- Single point of failure
- Must manage OS and Redis
- Manual scaling overhead

Step-by-step (AWS Console)

Below are explicit instructions and the exact AWS Console choices to use. Use the RDS console to create the Postgres instance first, then launch an EC2 instance in the same VPC, install Docker, and deploy the services with docker-compose.

1) Provision RDS — Create PostgreSQL

AWS Console !’ RDS !’ Create database

Choose: Engine=PostgreSQL, DB instance class=db.t4g.small (or db.t3.small), Multi-AZ=No, Storage=gp3 20GB, Public accessibility=No, VPC=select same VPC as EC2.

RDS Console — Key choices

Engine	PostgreSQL
DB instance class	db.t4g.small
Multi-AZ	No (single-AZ)
Storage	gp3, 20 GB

2) Provision

EC2 — Launch instance

AWS Console !’ EC2 !’ Launch instance. Choose AMI: Ubuntu 22.04 LTS or Amazon Linux 2. Instance type: t4g.medium or t3.medium. Configure storage: 20GB gp3. Security group: allow SSH from your IP, allow 80/443 from 0.0.0.0/0, allow 5432 from EC2 to RDS (via SG).

EC2 Launch Wizard — Key choices

4) Migrate and seed — Prisma

Run migration and seed commands from the `packages/prisma` folder with DATABASE_URL pointing to RDS.

Prisma commands

Migrate	pnpm --filter @fundifyhub/prisma run db:migrate
Seed	pnpm --filter @fundifyhub/prisma run db:seed

5) Monitor & scale

Create CloudWatch alarms for CPU > 70% and memory (via CloudWatch agent) and SNS notifications. When alarms trigger, consider scaling by separating services or moving to ECS.

Estimated monthly cost — Approach A

Scenario	Monthly cost (low)	Monthly co
100 monthly users (~10k requests)	$45	$95
500 monthly users (~50k requests)	$60	$150
1000 monthly users (~100k requests)	$100	$250

Approach B — ECS Fargate + RDS (Managed containers)

Pros / Cons

Pros

- Managed orchestration, easy CI/CD
- No EC2 maintenance

Cons

- Higher base cost for always-on tasks
- Slightly more complex initial setup

Steps

1) Build Docker images and push to ECR. 2) Create RDS as in Approach A. 3) Create ECS cluster (Fargate) and Task Definitions for frontend/api and worker. 4) Create ALB and configure listener/target groups. 5) CI/CD: GitHub Actions to push and update services.

ECS / Fargate Key choices

Task size	0.5 vCPU, 1 GB (start small)
Service	Frontend/API, Worker separate
ALB	HTTPS via ACM

Estimated monthly cost — Approach B

Scenario

100 monthly users

500 monthly users

1000 monthly users

Approach C — Serverless (Lambda + Aurora Serverless)

Pros / Cons

Pros

- Auto-scaling, low ops overhead
- Pay per use for spiky traffic

Cons

- Can be expensive for steady high traffic
- Need DB pooling/proxy

Steps

1) Convert Next.js and backend to serverless (serverless-next.js) or run as container-based Lambdas. 2) Use Aurora Serverless v2 or RDS Proxy to handle connections. 3) Use SQS for background jobs or lightweight Lambdas triggered by SQS. 4) CI/CD to deploy functions.

Serverless Key choices

Function memory	128-512 MB (optimize duration)
DB	Aurora Serverless v2 or RDS + Proxy
Queue	SQS (recommended) or ElastiCache

Estimated monthly cost — Approach C

Scenario

100 monthly users

500 monthly users

1000 monthly users

Approach D — Hybrid (ECS + EC2 worker)

Run frontend/API on ECS Fargate and run long-running workers on a small EC2 to reduce Fargate costs for long-running processes. Use RDS Multi-AZ for higher availability when ready.

Hybrid Key choices

Frontend	ECS Fargate
Worker	EC2 t4g.small
DB	RDS Multi-AZ (optional)

Estimated monthly cost — Approach D

Scenario	Monthly cost (low)	Monthly co
100 monthly users	$80	$180
500 monthly users	$120	$350
1000 monthly users	$220	$600

Runbook checklist & next steps

- Create IAM roles & limit permissions.
- Configure VPC with private subnets for RDS.
- Store secrets in AWS Secrets Manager or environment files with tight permissions.
- Set CloudWatch alarms (CPU, Memory, Error rates) and SNS for notifications.
- Keep a migration/backfill plan for existing production data before any destructive changes.