

Contents

RADIANT v4.18.0	1
Overview	1
Architecture	2
Requirements	2
Swift Deployer App	2
AWS Infrastructure	2
Quick Start	2
Option A: Use Swift Deployer (Recommended)	2
Option B: Manual Deployment	3
Accessing the Admin Dashboard	3
Project Structure	3
Infrastructure Tiers	4
Documentation	4
Development Phases	4
v4.18.0 New Features (PROMPT-33)	5
Admin Dashboard	5
Running the Dashboard	5
Database Migrations	5
Running Migrations	6
Seed Data	6
Lambda Handlers	6
Environment Variables	6
Testing	6
Error Handling	7
CI/CD	7
License	7

RADIANT v4.18.0

A **multi-tenant AWS SaaS platform** for AI model access and orchestration.

Overview

RADIANT consists of three components:

Component	Location	Purpose
Swift Deployer App	apps/swift-deployer/	macOS app for deploying AWS infrastructure
AWS Infrastructure	packages/infrastructure/	CDK stacks, Lambdas, databases (deployed to AWS)
Admin Dashboard	apps/admin-dashboard/	Next.js web admin interface (deployed to AWS)

Architecture

Your Mac

Swift Deployer App (RadiantDeployer)

- Runs locally on macOS
- Deploys infrastructure to AWS via CDK
- AI Assistant for deployment guidance

Deploys

AWS Cloud

Admin Dashboard (Next.js)

- Hosted on AWS (CloudFront + S3 or Amplify)
- Manage tenants, models, billing, compliance

AWS Infrastructure (CDK)

- API Gateway, Lambda, Aurora PostgreSQL
- Cognito, S3, SageMaker endpoints

Requirements

Swift Deployer App

- macOS 13.0 (Ventura) or later
- Xcode 15.0+ or Swift 5.9+

AWS Infrastructure

- Node.js 20.0+
- pnpm 8.0+
- AWS CDK 2.120.0+
- AWS Account with appropriate permissions

Quick Start

Option A: Use Swift Deployer (Recommended)

The Swift Deployer is a macOS app that guides you through deployment with an AI assistant.

```
cd apps/swift-deployer
swift build
swift run
```

This opens the RadiantDeployer app which will: 1. Validate your AWS credentials 2. Guide you through configuration 3. Deploy all infrastructure to AWS 4. Deploy the Admin Dashboard to AWS

Option B: Manual Deployment

1. Install dependencies

```
pnpm install
```

2. Build shared package

```
pnpm build:shared
```

3. Deploy infrastructure to AWS

```
pnpm deploy:dev
```

Accessing the Admin Dashboard

After deployment, the Admin Dashboard is accessible at your configured domain (e.g., admin.yourdomain.com). It is **not** run locally - it's deployed to AWS and accessed via browser.

Project Structure

```
radiant/
  package.json                # Root monorepo config
  pnpm-workspace.yaml         # pnpm workspace config
  tsconfig.base.json          # Base TypeScript config
  .github/workflows/ci.yml    # CI/CD pipeline
  .husky/pre-commit           # Pre-commit hooks

  apps/
    swift-deployer/           # RadiantDeployer macOS app (runs locally)
      Package.swift
      Sources/RadiantDeployer/
        Models/
        Services/
        Views/
      Tests/                  # Swift unit & E2E tests

    admin-dashboard/          # Next.js admin UI (deployed to AWS)
      app/(dashboard)/
      components/
      lib/
      e2e/                   # Playwright E2E tests
```

```

packages/
  shared/                # @radiant/shared - Types & constants
  src/
    types/
    constants/
    utils/
    errors/              # Standardized error codes
    testing/             # Test utilities

  infrastructure/        # @radiant/infrastructure - CDK stacks
    bin/radiant.ts
    lib/stacks/
    lambda/
      admin/__tests__/   # Lambda unit tests
      billing/__tests__/
      shared/__tests__/
    migrations/

docs/
  TESTING.md             # Testing guide
  ERROR_CODES.md         # Error codes reference
  sections/              # Full specification documents

```

Infrastructure Tiers

Tier	Name	Monthly Cost	Features
1	SEED	\$50-150	Dev/test, minimal resources
2	STARTUP	\$200-400	Small production, WAF
3	GROWTH	\$1,000-2,500	Self-hosted models, HIPAA
4	SCALE	\$4,000-8,000	Multi-region, global DB
5	ENTERPRISE	\$15,000-35,000	Enterprise-grade global

Documentation

- [Architecture Overview](#)
- [Full Specifications](#)

Development Phases

Phase	Description	Status
1-9	Core Platform (Sections 0-46)	Complete
10-24	PROMPT-32 Update	Complete
25-31	PROMPT-33 Update v3	Complete
32-33	Verification & Testing	Complete

v4.18.0 New Features (PROMPT-33)

Feature	Description
Unified Package System	Atomic component versioning with .pkg format
Cost Management	Real-time cost tracking with AI recommendations
Compliance Reports	SOC2, HIPAA, GDPR, ISO27001 report generation
Security Dashboard	Anomaly detection and intrusion alerts
A/B Testing	Hash-based experiments with statistical analysis
Deployment Settings	SSM-synced configuration with lock-step mode
AI Assistant	Claude-powered deployment guidance in Swift app

Admin Dashboard

The admin dashboard provides a complete management interface:

Page	Description
Dashboard	Overview metrics and activity
Models	AI model configuration
Providers	Provider management and credentials
Orchestration	Neural patterns and workflows
Administrators	User management and invitations
Billing	Credits, subscriptions, transactions
Storage	Storage usage monitoring
Localization	Translation management
Configuration	System settings
Migrations	Database migration approval
Audit Logs	Activity tracking
Notifications	System alerts
Settings	User preferences, deployment settings, timeouts
Analytics	Cost analytics with AI recommendations
Compliance	SOC2/HIPAA/GDPR/ISO27001 reports
Security	Anomaly detection and threat monitoring
Experiments	A/B testing with statistical analysis
Time Machine	Conversation history and versioning

Running the Dashboard

```
cd apps/admin-dashboard
pnpm dev
```

Database Migrations

44 migrations covering: - Core schema (tenants, users, admins) - AI models and providers - Orchestration patterns and workflows - Billing and credits - Storage management - Localization - Configuration management

Running Migrations

```
cd packages/infrastructure
pnpm migrate
```

Seed Data

Demo data is available for development:

```
psql -d radiant -f migrations/seed/001_demo_data.sql
```

Lambda Handlers

Handler	Purpose
router	Main API router
admin	Admin operations
feedback	Feedback learning
orchestration	Neural orchestration
proposals	Workflow proposals
localization	Translation API
configuration	Config management
billing	Credits and subscriptions
storage	Storage tracking
migration-approval	Dual-admin approval
cost-logger	Real-time cost tracking
compliance-reporter	SOC2/HIPAA/GDPR/ISO27001 reports
anomaly-detector	Security intrusion detection
experiment-tracker	A/B test assignment and analysis

Environment Variables

Copy the template and configure:

```
cp .env.example .env
```

Required variables: - AWS_REGION - AWS region - AWS_ACCOUNT_ID - AWS account ID - DB_CLUSTER_ARN - Aurora cluster ARN - DB_SECRET_ARN - Database secret ARN - COGNITO_USER_POOL_ID - Cognito user pool - COGNITO_CLIENT_ID - Cognito client ID

Testing

RADIANT includes comprehensive testing at multiple levels:

```
# Run all tests
```

```
pnpm test
```

```
# Run Lambda unit tests
```

```
cd packages/infrastructure && pnpm test
```

```
# Run E2E tests
```

```
cd apps/admin-dashboard && pnpm test:e2e
```

```
# Run Swift tests
```

```
cd apps/swift-deployer && swift test
```

See [Testing Guide](#) for detailed information.

Error Handling

RADIANT uses standardized error codes across all services. See [Error Codes Reference](#).

```
import { ErrorCodes, RadiantError } from '@radiant/shared';
```

```
throw new RadiantError(ErrorCodes.AUTH_INVALID_TOKEN);
```

CI/CD

GitHub Actions workflow includes: - Linting and type checking - Unit tests with coverage - E2E tests - CDK synthesis validation - Automated deployment (on merge to main)

License

MIT