Contents

RADIANT v4.18.0 - Deployment Guide

Overview

This guide covers deploying the RADIANT platform from development to production. The platform consists of:

1. **AWS Infrastructure** - CDK stacks for all cloud resources
2. **Admin Dashboard** - Next.js admin interface
3. **Swift Deployer App** - macOS deployment tool with AI assistant

What's New in v4.18.0

- **Unified Package System** - Deploy with atomic component versioning
- **AI Assistant** - Claude-powered deployment guidance in Swift app
- **Configurable Timeouts** - SSM-synced operation timeouts
- **Cost Management** - Real-time cost tracking and alerts
- **Compliance Reports** - SOC2, HIPAA, GDPR, ISO27001 reporting

Prerequisites

AWS Account Setup

Requirement	Action	Verification
AWS Account	Create or use existing	Account ID available
IAM User	Create with AdministratorAccess	Access keys configured
AWS CLI	Install and configure	`aws sts get-caller-identity`
Route 53 Domain (optional)	Register domain	Domain in hosted zone
ACM Certificate (optional)	Request in us-east-1	Certificate validated

Development Environment

Requirement	Version	Verification
Node.js	20.x LTS	`node --version`
pnpm	8.x+	`pnpm --version`
AWS CDK CLI	2.x	`cdk --version`
Xcode	15.x+	`xcode-select -p`
Swift	5.9+	`swift --version`

Quick Start

Automated Deployment

Use the deployment script for a streamlined deployment:

```
# Deploy to dev environment
./scripts/deploy.sh --environment dev

# Deploy to staging
./scripts/deploy.sh --environment staging

# Deploy to production
./scripts/deploy.sh --environment prod
```

> **Note:** RADIANT uses a unified deployment model. All features are available in every deployment. Licensing restrictions are handled at the application level, not infrastructure level.

Verify Deployment

After deployment, verify all resources:

```
./scripts/verify-deployment.sh --environment dev
```

Manual Deployment

1. Install Dependencies

```
cd radiant
npx pnpm install
```

2. Build Shared Package

```
cd packages/shared
npm run build
```

3. Build Lambda Functions

```
cd packages/infrastructure/lambda
npm install --legacy-peer-deps
npm run build
```

4. Build Admin Dashboard

```
cd apps/admin-dashboard
npm install
npm run build
```

CDK Deployment

Bootstrap CDK (One-time per account/region)

```
cd packages/infrastructure
npx cdk bootstrap aws://ACCOUNT_ID/us-east-1 --qualifier radiant
```

Deploy All Stacks

```
# Deploy in order with dependencies
npx cdk deploy --all \
  --context environment=dev \
  --context tier=1 \
  --require-approval never
```

Deploy Individual Stacks

```
# Phase 1: Foundation
npx cdk deploy Radiant-dev-Foundation --context environment=dev --context tier=1
npx cdk deploy Radiant-dev-Networking --context environment=dev --context tier=1

# Phase 2: Security & Data
npx cdk deploy Radiant-dev-Security --context environment=dev --context tier=1
npx cdk deploy Radiant-dev-Data --context environment=dev --context tier=1
npx cdk deploy Radiant-dev-Storage --context environment=dev --context tier=1

# Phase 3: Auth & AI
npx cdk deploy Radiant-dev-Auth --context environment=dev --context tier=1
npx cdk deploy Radiant-dev-AI --context environment=dev --context tier=1

# Phase 4: API & Admin
npx cdk deploy Radiant-dev-API --context environment=dev --context tier=1
npx cdk deploy Radiant-dev-Admin --context environment=dev --context tier=1
```

Database Migrations

After infrastructure is deployed, run database migrations:

```
# Connect to Aurora and run migrations
cd packages/infrastructure/migrations
./run-migrations.sh --environment dev
```

Migration files are applied in order (44 total): 1. `001_initial_schema.sql` - Base tables 2. `002_tenant_isolation.sql` - RLS policies 3. `003_ai_models.sql` - Providers and models 4. `004_usage_billing.sql` - Usage tracking 5. `005_admin_approval.sql` - Audit logs 6. `006_self_hosted_models.sql` - SageMaker config 7. `007_external_providers.sql` - Provider settings 8. ... (see `migrations/` for full list) 44. `044_cost_experiments_security.sql` - Cost tracking, A/B testing, security

Post-Deployment Configuration

Create First Super Admin

```
aws cognito-idp admin-create-user \
  --user-pool-id YOUR_ADMIN_POOL_ID \
  --username admin@example.com \
  --user-attributes Name=email,Value=admin@example.com \
  --temporary-password TempPass123! \
  --message-action SUPPRESS

aws cognito-idp admin-add-user-to-group \
  --user-pool-id YOUR_ADMIN_POOL_ID \
  --username admin@example.com \
  --group-name super_admin
```

Configure AI Providers

1. Navigate to Admin Dashboard → Providers
2. Add API keys for external providers:
   - OpenAI
   - Anthropic
   - Google AI
   - xAI (Grok)
   - DeepSeek
3. Verify connectivity with test requests

Environment Configuration

Tiers

Tier	Name	Use Case
1	SEED	Development, testing
2	STARTUP	Small production
3	GROWTH	Medium production
4	SCALE	Large production
5	ENTERPRISE	Enterprise with compliance

Environment Variables

Required environment variables for deployment:

```
export AWS_REGION=us-east-1
export AWS_ACCOUNT_ID=123456789012
export RADIANT_ENVIRONMENT=dev  # dev, staging, prod
export RADIANT_TIER=1           # 1-5
export RADIANT_DOMAIN=example.com
```

Verification

Health Checks

```
# API Health
curl https://YOUR_API_ENDPOINT/health

# Expected response:
# {"status":"healthy","version":"4.18.0"}
```

Smoke Tests

```
# Test chat completions
curl -X POST https://YOUR_API_ENDPOINT/v1/chat/completions \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"model":"gpt-4o-mini","messages":[{"role":"user","content":"Hello"}]}'
```

Troubleshooting

Common Issues

Issue	Cause	Solution
CDK bootstrap fails	Missing permissions	Ensure IAM user has AdministratorAccess
Aurora connection timeout	Security group	Check VPC endpoint and security group rules
Lambda cold starts	Function size	Enable provisioned concurrency for critical functions
Cognito auth fails	Pool configuration	Verify callback URLs and client settings

Logs

```
# View Lambda logs
aws logs tail /aws/lambda/Radiant-dev-router --follow

# View ECS logs (LiteLLM)
aws logs tail /ecs/radiant-dev-litellm --follow
```

Cleanup

To destroy all resources:

```
cd packages/infrastructure
npx cdk destroy --all --context environment=dev --context tier=1
```

Warning: This will delete all data including databases. Export data before destroying.

CI/CD Pipeline

RADIANT includes a GitHub Actions CI/CD pipeline:

Workflow Stages

Stage	Trigger	Actions
Lint	All PRs	ESLint, TypeScript check
Build	All PRs	Build shared, infrastructure, dashboard
Test	All PRs	Unit tests, coverage report
CDK Synth	All PRs	Validate CDK templates
Deploy Dev	Merge to develop	Auto-deploy to dev
Deploy Prod	Merge to main	Deploy with approval

Pre-commit Hooks

Pre-commit hooks run automatically via Husky:

```
# Hooks run on every commit:
- lint-staged (ESLint, Prettier)
- Secret detection
- Version bump enforcement
- Discrete validation
```

To bypass (not recommended):

```
git commit --no-verify -m "message"
```

Manual Deployment from CI

```
# Trigger deployment workflow
gh workflow run deploy.yml -f environment=staging -f tier=2
```

Testing Before Deployment

Always run tests before deploying:

```
# Run all tests
pnpm test

# Run E2E tests
cd apps/admin-dashboard && pnpm test:e2e

# Run CDK synthesis (validates templates)
cd packages/infrastructure && npx cdk synth
```

See Testing Guide for comprehensive testing information.

Support

For issues, check: 1. CloudWatch Logs for error details 2. CDK diff to verify expected changes 3. AWS Console for resource status 4. Troubleshooting Guide 5. Error Codes Reference