

# Contents

<b>RADIANT v4.18.0 - AI Build Agent Configuration</b>	<b>1</b>
What You Are Building . . . . .	1
Technology Stack . . . . .	1
Project Structure . . . . .	1
Critical Constants . . . . .	2
MANDATORY POLICY ENFORCEMENT . . . . .	2
MANDATORY DOCUMENTATION UPDATES . . . . .	2
Quick Reference - Always Update These: . . . . .	3
The Golden Rule . . . . .	3
Implementation Rules . . . . .	3
Phase Execution . . . . .	3
Batch 1: Core Platform (Phases 1-9) COMPLETE . . . . .	3
Batch 2: PROMPT-32 Update (Phases 10-24) COMPLETE . . . . .	4
Batch 3: PROMPT-33 Update v3 (Phases 25-31) COMPLETE . . . . .	4
Batch 4: Verification (Phases 32-33) COMPLETE . . . . .	4
Quick Start Commands . . . . .	4
Documentation References . . . . .	4
Verification Checklist . . . . .	5

## RADIANT v4.18.0 - AI Build Agent Configuration

This file provides persistent context for Windsurf/Claude Opus 4.5 Read this FIRST before any implementation work

### What You Are Building

RADIANT is a **multi-tenant AWS SaaS platform** for AI model access and orchestration. It has THREE components:

Component	Location	Purpose
<b>Swift Deployer App</b>	apps/swift-deployer/	macOS app that deploys infrastructure
<b>AWS Infrastructure</b>	packages/infrastructure/	CDK stacks, Lambdas, databases
<b>Admin Dashboard</b>	apps/admin-dashboard/	Next.js web admin interface

### Technology Stack

- **Swift App:** SwiftUI, macOS 13.0+, Swift 5.9+, Xcode 15+, SQLCipher
- **Infrastructure:** AWS CDK (TypeScript), Aurora PostgreSQL, Lambda, API Gateway
- **Dashboard:** Next.js 14, TypeScript, Tailwind CSS
- **AI Integration:** 106+ models (50 external + 56 self-hosted), LiteLLM

### Project Structure

radiant/



```

VERSION                                # Package version: "4.18.0"
RADIANT_VERSION                        # Radiant component: "4.18.0"
THINKTANK_VERSION                      # Think Tank: "3.2.0"
VERSION_HISTORY.json                  # All releases with hashes
tools/scripts/                        # Build & validation scripts

apps/
  swift-deployer/                      # Swift macOS deployer app
    Package.swift
    Sources/RadiantDeployer/
      Models/
      Services/                        # AIAssistantService, LocalStorageManager, TimeoutService
      Views/

  admin-dashboard/                    # Next.js admin UI
    app/(dashboard)/                  # Dashboard pages
    components/                       # React components

packages/
  shared/                             # @radiant/shared - Types & constants
  infrastructure/                     # @radiant/infrastructure - CDK stacks
    lib/stacks/                       # 14 CDK stacks
    lambda/                           # Lambda handlers
    migrations/                       # 44 database migrations

```

## Critical Constants

```

// Always use these - NEVER hardcode
const RADIANT_VERSION = "4.18.0";
const DOMAIN_PLACEHOLDER = "{{RADIANT_DOMAIN}}";

```

## MANDATORY POLICY ENFORCEMENT

**BEFORE** starting ANY significant task:

1. Scan **ALL** workflows in `/.windsurf/workflows/`
2. Read each **description** field in the frontmatter
3. For each **matching policy** → follow its requirements
4. New policies **auto-apply** - no manual update needed

DYNAMIC: New policies in `/.windsurf/workflows/` are **AUTOMATICALLY** enforced.

**NO EXCEPTIONS.** Forgetting policies creates technical debt and compliance risks.

## MANDATORY DOCUMENTATION UPDATES

**CRITICAL:** Every code change **MUST** include documentation updates

**Master Policy:** `/.windsurf/workflows/docs-update-all.md` **Documentation Manifest:** `/docs/DOCUMENTATION-MANIFEST.json`



## Quick Reference - Always Update These:

Change Type	Required Documentation
<b>ANY change</b>	CHANGELOG.md
<b>Think Tank feature</b>	THINKTANK-USER-GUIDE.md + THINKTANK-ADMIN-GUIDE.md + THINKTANK-ADMIN-GUIDE-V2.md
<b>Platform feature</b>	RADIANT-ADMIN-GUIDE.md
<b>Admin guide update</b>	STRATEGIC-VISION-MARKETING.md (auto-trigger)
<b>Technical/Architecture</b>	ENGINEERING-IMPLEMENTATION-VISION.md
<b>Database change</b>	sections/SECTION-07-DATABASE-SCHEMA.md + RADIANT-PLATFORM-ARCHITECTURE.md
<b>Competitive advantage</b>	THINKTANK-MOATS.md or RADIANT-MOATS.md
<b>Swift Deployer change</b>	SWIFT-DEPLOYER-USER-GUIDE.md

## The Golden Rule

IF you change code → THEN you MUST update ALL applicable documentation

**NEVER:** - Say “I’ll update docs later” - Update only CHANGELOG.md - Skip user guide for user-facing changes - Skip admin guide for admin-facing changes - Forget STRATEGIC-VISION-MARKETING.md when updating admin guides

**ALWAYS:** - Update documentation IN THE SAME TASK as code changes - Update version numbers in all touched documents - Check the trigger matrix in /docs/DOCUMENTATION-MANIFEST.json

## Implementation Rules

1. **File Creation Order:** Follow dependency graph in each section
2. **RLS Variables:** Always use `app.current_tenant_id` (not `current_tenant_id`)
3. **Sendable Conformance:** All types crossing actor boundaries must be Sendable
4. **No Hardcoded Versions:** Use `RADIANT_VERSION` constant
5. **Error Handling:** Use typed errors with helpful messages
6. **Policy Compliance:** Run `/policy-enforcement` pre-flight check for every task

## Phase Execution

Phases are in `docs/phases/`. Execute in order:

### Batch 1: Core Platform (Phases 1-9) COMPLETE

1. **Phase 1:** Foundation (Sections 0-2) - Shared types, Swift app, base CDK
2. **Phase 2:** Core Infrastructure (Sections 3-7) - AI stacks, Lambdas, DB schema
3. **Phase 3:** Admin & Deployment (Sections 8-9) - Dashboard, assembly guide
4. **Phase 4:** AI Features (Sections 10-17) - Visual AI, Brain, Analytics
5. **Phase 5:** Consumer Platform (Sections 18-28) - Think Tank, collaboration
6. **Phase 6:** Advanced Features (Sections 29-35) - Registry, Time Machine
7. **Phase 7:** Intelligence Layer (Sections 36-39) - Neural engine, workflows
8. **Phase 8:** Platform Hardening (Sections 40-42) - Isolation, i18n, config
9. **Phase 9:** Billing System (Sections 43-46) - Credits, storage, subscriptions



## Batch 2: PROMPT-32 Update (Phases 10-24) COMPLETE

Extension specs in: `radiant-cascade-extension-v4.18.0/docs/phases/`

10. **Phase 10:** Header & Shared Types (PROMPT-32 Part 01)
11. **Phase 11:** Swift App Enhancements (PROMPT-32 Part 02)
12. **Phase 12:** CDK Base Updates (PROMPT-32 Part 03)
13. **Phase 13:** CDK AI & API (PROMPT-32 Part 04)
14. **Phase 14:** Lambda Core (PROMPT-32 Part 05)
15. **Phase 15:** Lambda Admin (PROMPT-32 Part 06)
16. **Phase 16:** Self-Hosted Models (PROMPT-32 Part 07)
17. **Phase 17:** Database Schema (PROMPT-32 Part 08)
18. **Phase 18:** Admin Dashboard (PROMPT-32 Part 09)
19. **Phase 19:** Deployment Guide (PROMPT-32 Part 10)
20. **Phase 20:** AI Features (PROMPT-32 Part 11)
21. **Phase 21:** Think Tank (PROMPT-32 Part 12)
22. **Phase 22:** Time Machine (PROMPT-32 Part 13)
23. **Phase 23:** Orchestration (PROMPT-32 Part 14)
24. **Phase 24:** Billing (PROMPT-32 Part 15)

## Batch 3: PROMPT-33 Update v3 (Phases 25-31) COMPLETE

25. **Phase 25:** Package System (PROMPT-33 Part 01)
26. **Phase 26:** AI Assistant & Progress UI (PROMPT-33 Part 02)
27. **Phase 27:** Build System & Local Storage (PROMPT-33 Part 03)
28. **Phase 28:** Cost Management (PROMPT-33 Part 04)
29. **Phase 29:** Compliance & Security (PROMPT-33 Part 05)
30. **Phase 30:** A/B Testing & Settings (PROMPT-33 Part 06)
31. **Phase 31:** Database & Checklist (PROMPT-33 Part 07)

## Batch 4: Verification (Phases 32-33) COMPLETE

32. **Phase 32:** Implementation Guide Verification
33. **Phase 33:** Interrogation Testing (55 edge-case scenarios)

## Quick Start Commands

```
# Start Phase 1 implementation  
/implement-phase 1
```

```
# Continue to next phase  
/continue-phase
```

```
# Verify current phase  
/verify-phase
```

## Documentation References

- Full specs: `docs/sections/` (split by section number)
- Phase summaries: `docs/phases/`



- Original prompt: `docs/RADIANT-PROMPT-32-FULL.md`

### **Verification Checklist**

Before marking any phase complete: - ☐ All files created per section spec - ☐ Types match shared definitions in Section 0 - ☐ Imports resolve correctly - ☐ No TypeScript/Swift compilation errors  
- ☐ Database migrations are numbered sequentially