Contents

RADIANT Deployer - Administrator Guide

> **Complete guide for deploying and managing RADIANT infrastructure using the Swift Deployer App**
>
> Version: 4.18.1 | Last Updated: December 2024

Table of Contents

---

1. Introduction

1.1 What is RADIANT Deployer?

RADIANT Deployer is a native macOS application that provides a complete deployment management solution for the RADIANT platform. It offers:

- **One-Click Deployments**: Deploy entire infrastructure stacks with a single click
- **AI-Powered Assistance**: Claude-powered assistant for deployment guidance
- **Snapshot Management**: Create and restore deployment snapshots
- **Multi-Region Support**: Deploy across multiple AWS regions
- **Health Monitoring**: Real-time health checks and status monitoring
- **Secure Credential Storage**: Keychain-integrated credential management

1.2 Architecture

```
                    RADIANT Deployer App


      Dashboard           Deployments          Settings
        View                  View               View



                         Services Layer

         AWS              CDK              AI Assistant
        Service          Service             Service


        Snapshot         Health             Local Storage
        Service          Check               Manager




                       Storage Layer

         Keychain        SQLCipher          File System
        (Secrets)        (Local DB)         (Snapshots)
```

1.3 Key Features

Feature	Description
Deployment Wizard	Step-by-step guided deployment process
Lock-Step Mode	Ensures component version consistency
Automatic Rollback	Reverts failed deployments automatically
Offline Mode	Core functionality works without internet
Audit Logging	Complete deployment history tracking

Deployment Wizard Explained The Deployment Wizard breaks down the complex AWS infrastructure deployment into manageable steps. Instead of manually running CDK commands and configuring services, the wizard:

1. **Validates Prerequisites**: Checks that all required software (Node.js, AWS CLI, CDK) is installed and properly configured before starting
2. **Guides Configuration**: Walks you through selecting environment (dev/staging/prod), tier (1-5), and region settings with explanations for each option
3. **Shows Progress**: Displays real-time progress as each CloudFormation stack deploys, with estimated time remaining
4. **Handles Errors**: If any step fails, provides clear error messages and suggested remediation steps

Lock-Step Mode Explained Lock-Step Mode prevents version mismatches between RADIANT components that could cause compatibility issues:

- **What it does**: Ensures the Admin Dashboard, Lambda functions, and database schema are all on compatible versions
- **Why it matters**: A v4.18 Lambda function might expect database columns that don't exist in a v4.17 schema, causing runtime errors
- **How it works**: Before deployment, the system checks version numbers across all components and blocks deployment if drift exceeds the configured maximum (default: 1 minor version)
- **When to disable**: Only disable during development when testing individual component changes

Automatic Rollback Explained Automatic Rollback protects your production environment by reverting failed deployments:

- **Trigger conditions**: Activates when any deployment phase fails (CDK deploy error, health check failure, migration error)
- **Rollback process**: Restores the pre-deployment snapshot, which includes database state, Lambda code, and configuration
- **Recovery time**: Typically 5-10 minutes depending on the size of changes being reverted
- **Notification**: Sends alerts via configured channels (email, Slack) when rollback occurs

Offline Mode Explained Offline Mode allows essential operations when internet connectivity is unavailable:

- **Available offline**: Viewing deployment history, browsing local snapshots, reviewing configuration, accessing cached documentation
- **Requires internet**: Deploying to AWS, health checks, AI assistant queries, credential validation
- **Data sync**: When connectivity returns, local changes sync automatically with the cloud state

Audit Logging Explained Every action in the Deployer is logged for compliance and troubleshooting:

- **What's logged**: User identity, timestamp, action type, parameters, outcome, duration
- **Storage**: Logs stored locally in SQLCipher database and optionally synced to CloudWatch
- **Retention**: Local logs kept for 90 days by default (configurable)
- **Export**: Logs can be exported to CSV/JSON for compliance audits

---

2. System Requirements

2.1 Hardware Requirements

Component	Minimum	Recommended
macOS Version	13.0 (Ventura)	14.0+ (Sonoma)
Processor	Apple Silicon or Intel	Apple Silicon M1+
Memory	8 GB RAM	16 GB RAM
Storage	2 GB free	10 GB free
Display	1280x800	1440x900+

Why These Requirements? **macOS 13.0+**: Required for Swift 5.9 runtime and modern SwiftUI features. Older versions lack required system APIs for secure keychain access and modern networking.

Apple Silicon Recommended: While Intel Macs are supported, Apple Silicon provides 2-3x faster CDK synthesis and compilation times. The app is built as a universal binary supporting both architectures.

8 GB RAM Minimum: CDK synthesis loads the entire infrastructure definition into memory. Complex deployments (Tier 3+) with many resources may require more memory. With 8 GB, you may experience slowdowns during synthesis.

16 GB RAM Recommended: Allows comfortable multitasking while deployments run in the background. Essential if you're also running Docker, IDEs, or other development tools.

2 GB Storage Minimum: Covers the application itself (~200 MB), local database (~50 MB), and several snapshots. However, snapshots can grow large over time.

10 GB Storage Recommended: Provides room for multiple deployment snapshots, comprehensive logs, and CDK cache. Each full snapshot can be 100-500 MB depending on your deployment size.

2.2 Software Requirements

Software	Version	Purpose	Installation
Xcode	15.0+	Swift runtime (Command Line Tools sufficient)	`xcode-select --install`
AWS CLI	2.x	AWS operations	`brew install awscli`
Node.js	20.x LTS	CDK operations	`brew install node@20`
AWS CDK	2.120+	Infrastructure deployment	`npm install -g aws-cdk`
pnpm	8.x+	Package management	`npm install -g pnpm`

Software Explained **Xcode Command Line Tools**: Provides the Swift runtime and compiler. You don't need the full Xcode IDE - just the command line tools (4 GB vs 12 GB download). The Deployer uses Swift for its native performance and seamless macOS integration.

AWS CLI v2: The official AWS command-line interface. Used internally by the Deployer to execute AWS operations, assume IAM roles, and query service status. Version 2 is required for SSO support and improved credential handling.

Node.js 20 LTS: Required to run the AWS CDK, which is written in TypeScript. LTS (Long Term Support) versions receive security updates for 30 months. The Deployer manages Node.js processes internally during CDK operations.

AWS CDK 2.120+: The Cloud Development Kit that defines RADIANT's infrastructure as TypeScript code. Version 2.120+ includes critical bug fixes for Aurora PostgreSQL and Lambda layer handling. The CDK synthesizes your infrastructure into CloudFormation templates.

pnpm 8.x: A fast, disk-efficient package manager used to install CDK dependencies. Chosen over npm for its superior handling of monorepo workspaces and 2-3x faster installation times.

2.3 AWS Requirements

Requirement	Details	How to Verify
AWS Account	Active account with billing enabled	Check AWS Console billing page
IAM User	AdministratorAccess or equivalent	`aws sts get-caller-identity`
Regions	Access to us-east-1 (required) + additional regions	`aws ec2 describe-regions`
Service Quotas	Default quotas sufficient for Tier 1-2	AWS Service Quotas console

AWS Requirements Explained **Active AWS Account with Billing**: RADIANT deploys real AWS resources that incur costs. Billing must be enabled and a valid payment method on file. New accounts have a $5 pending charge verification. Free tier covers some resources for 12 months but won't cover all RADIANT components.

IAM User with AdministratorAccess: The Deployer needs broad permissions to create and manage resources across many AWS services. For production, you can use a scoped-down policy (see Appendix A), but AdministratorAccess is recommended for initial setup to avoid permission errors.

Required Permissions Include: - CloudFormation (stack operations) - Lambda (function management) - API Gateway (REST API setup) - RDS/Aurora (database provisioning) - Cognito (user pool management) - S3 (storage buckets) - IAM (role creation) - SSM (parameter storage) - Secrets Manager (credential storage) - CloudWatch (logging and monitoring)

us-east-1 Required: This region is required even if you deploy to other regions because: - ACM certificates for CloudFront must be in us-east-1 - Some global services (IAM, Route 53) operate from us-east-1 - CDK bootstrap resources are region-specific

Service Quotas: Default AWS quotas are sufficient for Tier 1-2 deployments. Tier 3+ may require quota increases for: - Lambda concurrent executions (default: 1,000) - RDS instances (default: 40) - VPC Elastic IPs (default: 5)

To request quota increases: AWS Console > Service Quotas > Select service > Request increase

---

3. Installation

3.1 Download and Install

Option A: Pre-built Application (Recommended)

1. Download the latest release from GitHub Releases
2. Drag `RadiantDeployer.app` to `/Applications`
3. Right-click and select "Open" (first launch only)
4. Grant necessary permissions when prompted

Option B: Build from Source

```
# Clone the repository
git clone https://github.com/your-org/radiant.git
cd radiant/apps/swift-deployer

# Build the application
swift build -c release

# Run the application
swift run RadiantDeployer
```

3.2 Initial Permissions

The app requires the following permissions:

Permission	Purpose	How to Grant
Keychain Access	Store AWS credentials securely	Approve on first credential save
Network Access	Connect to AWS and AI services	Approve in System Settings
File Access	Save snapshots and logs	Approve when prompted

3.3 Verify Installation

1. Launch RadiantDeployer
2. Navigate to **Settings → About**
3. Verify version shows `4.18.1`
4. Check all services show green status

---

4. First-Time Setup

4.1 Setup Wizard

On first launch, the Setup Wizard guides you through:

```
                    Setup Wizard


  Step 1: Welcome                       Complete
  Step 2: AWS Credentials               In Progress
  Step 3: Environment Configuration     Pending
  Step 4: AI Assistant Setup            Pending
  Step 5: Verification                  Pending
```

4.2 AWS Credentials Setup

1. Click **"Add AWS Credentials"**
2. Enter your credentials:
    - **Name**: Descriptive name (e.g., "Production Account")
    - **Access Key ID**: `AKIA...` (20 characters)
    - **Secret Access Key**: Your secret key (40 characters)
    - **Region**: Primary region (e.g., `us-east-1`)
3. Click **"Validate"** to test connectivity
4. Click **"Save"** to store securely in Keychain

4.3 Environment Configuration

Configure your deployment environment:

Setting	Description	Default
Environment	`dev`, `staging`, or `prod`	`dev`
Tier	Infrastructure tier (1-5)	`1`
Domain	Your domain name	Required for Tier 2+
Stack Prefix	CDK stack name prefix	`radiant`

Environment Types Explained **Development (dev)**: For active development and testing. Features relaxed security settings, smaller instance sizes, and no deletion protection. Data can be reset freely. Cost: ~$50-150/month for Tier 1.

Staging (staging): Pre-production environment that mirrors production configuration. Use this to test deployments before going live. Same security as production but can use smaller instances. Cost: ~60% of production.

Production (prod): Live environment serving real users. Includes deletion protection, multi-AZ deployments, automated backups, and enhanced monitoring. Never deploy untested changes directly to production.

Infrastructure Tiers Explained

Tier	Name	Monthly Cost	Use Case	Resources
1	SEED	$50-150	Development, POC	Single-AZ, t3.small instances, 20GB storage
2	STARTUP	$200-400	Small production	Multi-AZ database, WAF, basic monitoring
3	GROWTH	$1,000-2,500	Medium production	Self-hosted models, HIPAA compliance, enhanced security
4	SCALE	$4,000-8,000	Large production	Multi-region, global database, dedicated instances
5	ENTERPRISE	$15,000-35,000	Enterprise	Custom SLAs, dedicated support, on-premise options

Choosing the Right Tier: Start with Tier 1 for development. Move to Tier 2 when you have paying customers. Tier 3+ is for organizations with compliance requirements or high traffic.

Domain Configuration For Tier 2+, you need a custom domain:

1. **Register a domain** in Route 53 or transfer an existing domain
2. **Create a hosted zone** in Route 53 for your domain
3. **Request an ACM certificate** in us-east-1 for `*.yourdomain.com`
4. **Validate the certificate** via DNS (automatic if using Route 53)

The Deployer will configure: - `api.yourdomain.com` - API Gateway endpoint - `admin.yourdomain.com` - Admin Dashboard - `app.yourdomain.com` - Think Tank application

4.4 AI Assistant Setup (Optional)

Enable the Claude-powered AI assistant:

1. Navigate to **Settings → AI Assistant**
2. Enter your Anthropic API key
3. Select response style:
    - **Concise**: Brief, action-focused responses
    - **Detailed**: In-depth explanations
    - **Tutorial**: Step-by-step guidance
4. Test the connection with a sample query

---

5. AWS Credentials Management

5.1 Credential Sets

Manage multiple AWS accounts:

Field	Description	Example
Name	Friendly identifier	"Production"
Access Key ID	AWS access key	`AKIAIOSFODNN7EXAMPLE`
Secret Access Key	AWS secret	(stored encrypted)
Region	Default region	`us-east-1`
Role ARN	Optional assume role	`arn:aws:iam::123:role/deploy`

5.2 Adding Credentials

1. Navigate to **Credentials** tab
2. Click **"+ Add Credential Set"**
3. Fill in the required fields
4. Click **"Validate"** to test
5. Click **"Save"**

5.3 Credential Validation

The app validates:

- Access key format (AKIA prefix, 20 chars)
- Secret key length (40+ chars)
- Region validity
- AWS connectivity (STS GetCallerIdentity)
- Required permissions

5.4 Security Best Practices

Practice	Recommendation
Rotate Keys	Every 90 days
Least Privilege	Use scoped IAM policies
MFA	Enable on AWS account
Audit	Review access logs regularly
Backup	Export credentials securely

5.5 Importing from AWS CLI

```
# The app can import from ~/.aws/credentials
# Navigate to Credentials → Import from AWS CLI
```

---

6. Deployment Operations

6.1 Deployment Dashboard

The main dashboard shows:

```
Environment: dev          Status:   Healthy
Version: 4.18.1           Last Deploy: 2024-12-25 10:30:00



   Deploy           Rollback          Settings
   [Button]         [Button]          [Button]


Recent Deployments:

  2024-12-25 10:30   v4.18.1   prod    Success   4m 32s
  2024-12-24 15:45   v4.18.0   prod    Success   5m 12s
  2024-12-24 09:00   v4.17.0   dev     Success   3m 45s
```

6.2 Starting a Deployment

1. Select target **Environment** (dev/staging/prod)
2. Select **Tier** (1-5)
3. Review deployment plan
4. Click **"Start Deployment"**
5. Monitor progress in real-time

6.3 Deployment Phases

Phase	Duration	Description
1. Validation	~30s	Credential and configuration check
2. Snapshot	~1m	Create pre-deployment backup
3. CDK Synth	~1m	Generate CloudFormation templates
4. CDK Deploy	~10-20m	Deploy infrastructure
5. Migration	~2m	Run database migrations
6. Health Check	~1m	Verify all services
7. Cleanup	~30s	Remove temporary resources

Phase Details **Phase 1 - Validation (30 seconds)**

Before any changes are made, the system validates: - AWS credentials are valid and not expired - IAM permissions are sufficient for all required operations - Target environment exists or can be created - No conflicting deployments are in progress (deployment lock check) - Required software versions are installed (Node.js, CDK, etc.) - Network connectivity to AWS services

If validation fails, you'll see specific error messages explaining what needs to be fixed.

Phase 2 - Snapshot (1 minute)

A pre-deployment snapshot captures the current state so you can rollback if needed: - Database schema and critical data (not full data backup) - Current Lambda function code versions - SSM Parameter Store values - Current CloudFormation stack states - Configuration files

Snapshots are stored locally and can be managed in the Snapshots tab.

Phase 3 - CDK Synth (1 minute)

The CDK synthesizes TypeScript infrastructure code into CloudFormation templates: - Reads infrastructure definitions from `packages/infrastructure/` - Resolves all construct dependencies - Generates CloudFormation JSON/YAML templates - Calculates resource changes (what will be created/updated/deleted) - Validates templates against AWS CloudFormation rules

You can review the generated templates before proceeding.

Phase 4 - CDK Deploy (10-20 minutes)

The actual AWS resource deployment happens in this phase: - CloudFormation stacks are created or updated in dependency order - Resources are provisioned (databases, Lambda functions, API Gateway, etc.) - IAM roles and policies are configured - Networking (VPC, subnets, security groups) is set up - DNS records are created/updated

This is the longest phase. Progress shows which stack is currently deploying.

Phase 5 - Migration (2 minutes)

Database migrations ensure your schema matches the deployed code: - Connects to Aurora PostgreSQL using Data API - Runs pending migration files in sequence - Creates new tables, columns, indexes as needed - Updates RLS (Row-Level Security) policies - Verifies migration success with integrity checks

Migrations are idempotent - running them twice won't cause issues.

Phase 6 - Health Check (1 minute)

Verifies all deployed services are functioning: - API Gateway responds to health endpoint - Lambda functions can be invoked - Database connections succeed - Cognito user pools are accessible - S3 buckets are reachable - CloudFront distributions are deployed

If health checks fail, automatic rollback is triggered (if enabled).

Phase 7 - Cleanup (30 seconds)

Final cleanup tasks: - Removes temporary files created during deployment - Clears CDK staging buckets of old assets - Updates deployment history in local database - Releases deployment lock - Sends completion notification

6.4 Deployment Progress

```
Deploying to Production


[                ] 65%

Current Phase: CDK Deploy
Stack: Radiant-prod-API (5 of 9)
Elapsed: 8m 23s | Estimated: 4m remaining

  Validation complete
  Snapshot created: snap-20241225-103000
  CDK synthesis complete
  Deploying stacks...
    Radiant-prod-Foundation
    Radiant-prod-Networking
    Radiant-prod-Security
    Radiant-prod-Data
    Radiant-prod-API
    Radiant-prod-Auth
    Radiant-prod-AI
    Radiant-prod-Admin
    Radiant-prod-Monitoring

[Cancel Deployment]
```

6.5 Deployment Settings

Configure deployment behavior:

Setting	Description	Default
Auto-Rollback	Rollback on failure	Enabled
Lock-Step Mode	Require version consistency	Enabled
Max Version Drift	Maximum version difference	1
Approval Required	Require confirmation for prod	Enabled
Notification	Send completion notifications	Enabled

6.6 Operation Timeouts

Operation	Default Timeout	Configurable
CDK Deploy	30 minutes	Yes
Health Check	5 minutes	Yes
Migration	10 minutes	Yes
Snapshot	5 minutes	Yes
Rollback	15 minutes	Yes

---

7. Multi-Region Deployments

7.1 Overview

Deploy RADIANT across multiple AWS regions for:

- **High Availability**: Survive regional outages
- **Low Latency**: Serve users from nearest region
- **Compliance**: Data residency requirements

7.2 Supported Regions

Region	Code	Primary Use
US East (N. Virginia)	`us-east-1`	Primary (required)
US West (Oregon)	`us-west-2`	West coast users
EU (Ireland)	`eu-west-1`	European users
EU (Frankfurt)	`eu-central-1`	GDPR compliance
Asia Pacific (Singapore)	`ap-southeast-1`	APAC users
Asia Pacific (Tokyo)	`ap-northeast-1`	Japanese users

7.3 Adding a Region

1. Navigate to **Multi-Region** tab
2. Click **"Add Region"**
3. Configure:
    - **Region**: Select from available regions
    - **Is Primary**: Set primary region flag
    - **Stack Prefix**: Region-specific prefix

   - **Endpoint**: Custom domain for region
4. Click **“Deploy to Region”**

7.4 Region Consistency

Monitor version consistency across regions:

```
Multi-Region Status                         Consistency:   100%


Region            Version    Status    Last Deploy

us-east-1 (P)     4.18.1      Healthy  2024-12-25 10:30
eu-west-1         4.18.1      Healthy  2024-12-25 10:35
ap-southeast-1    4.18.1      Healthy  2024-12-25 10:40

[Deploy All]  [Check Consistency]  [Sync Versions]
```

---

8. Snapshots & Rollbacks

8.1 Snapshot Types

Type	Description	Retention
Pre-Deploy	Automatic before each deployment	30 days
Manual	User-initiated backup	Until deleted
Scheduled	Periodic backups	Configurable

8.2 Creating a Snapshot

1. Navigate to **Snapshots** tab
2. Click **“Create Snapshot”**
3. Enter description (optional)
4. Select components to include:
   - Database state
   - Configuration
   - Lambda code
   - Infrastructure state
5. Click **“Create”**

8.3 Snapshot Contents

```
Snapshot: snap-20241225-103000
  metadata.json          # Snapshot info
  database/
```

```
    schema.sql          # Database schema
    data.sql            # Critical data
config/
    ssm-parameters.json # SSM parameters
    secrets.json        # Secret references
lambda/
    functions.zip       # Lambda code packages
infrastructure/
     state.json          # CDK state
```

8.4 Restoring from Snapshot

1. Navigate to **Snapshots** tab
2. Select snapshot to restore
3. Click **"Restore"**
4. Confirm restoration scope:
    - Full restoration
    - Database only
    - Configuration only
5. Click **"Confirm Restore"**

8.5 Automatic Rollback

When enabled, failed deployments automatically:

1. Stop deployment process
2. Identify last known good state
3. Restore from pre-deploy snapshot
4. Verify system health
5. Send notification

---

9. AI Assistant

9.1 Overview

The Claude-powered AI Assistant provides:

- **Deployment Guidance**: Step-by-step help
- **Error Translation**: User-friendly error explanations
- **Troubleshooting**: Problem diagnosis
- **Best Practices**: Recommendations

9.2 Enabling AI Assistant

1. Navigate to **Settings → AI Assistant**
2. Enter Anthropic API key
3. Toggle **"Enable AI Assistant"**
4. Select response style

9.3 Using the Assistant

Access the assistant via:

- **Chat Panel**: Click AI icon in toolbar
- **Contextual Help**: Click “?” on any screen
- **Error Dialogs**: Click “Explain” on errors

9.4 Sample Interactions

```
AI Assistant


You: What does error "AccessDenied: User is not authorized
     to perform dynamodb:CreateTable" mean?

Assistant: This error indicates your AWS credentials don't
have permission to create DynamoDB tables. To fix this:

1. Check your IAM user/role has the required permissions
2. Add the "AmazonDynamoDBFullAccess" managed policy, or
3. Add specific permissions:
   - dynamodb:CreateTable
   - dynamodb:DescribeTable
   - dynamodb:UpdateTable

Would you like me to show the exact IAM policy needed?

[Yes, show policy]  [No, I'll figure it out]
```

9.5 Offline Mode

When offline, the assistant provides:

- Pre-cached common error explanations
- Local troubleshooting guides
- Fallback recommendations

---

10. Package Management

10.1 Package System

RADIANT uses atomic packages for deployment:

```
radiant-4.18.1.pkg
  manifest.json          # Package manifest
  checksums.sha256       # Component checksums
```

```
radiant/                # Radiant components
   infrastructure/
   lambda/
   dashboard/
thinktank/              # Think Tank components
    api/
    frontend/
```

10.2 Viewing Packages

Navigate to **Packages** tab to see:

- Installed packages
- Available updates
- Package history
- Component versions

10.3 Version Management

Version Type	Format	Example
Radiant	Major.Minor.Patch	4.18.1
Think Tank	Major.Minor.Patch	3.2.0
Package	Combined	4.18.1+3.2.0

10.4 Lock-Step Mode

When enabled:

- All components must have same minor version
- Maximum version drift configurable
- Automatic sync available

---

11. Monitoring & Health Checks

11.1 Health Dashboard

```
System Health                           Overall:  Healthy


Service              Status    Latency    Last Check

API Gateway           Up      45ms       10s ago
Lambda (Router)       Up      120ms      10s ago
Aurora PostgreSQL     Up      12ms       10s ago
DynamoDB              Up      8ms        10s ago
Cognito               Up      85ms       10s ago
```

```
  S3 Storage             Up      35ms       10s ago
  CloudFront             Up      22ms       10s ago
  SageMaker (if T3+)     Up      250ms      10s ago

  [Refresh]  [Run Full Check]  [View History]
```

11.2 Health Check Types

Check	Frequency	Timeout
Quick	Every 60s	5s
Standard	Every 5m	30s
Deep	Manual/Deploy	2m

11.3 Alerts

Configure alerts for:

- Service degradation
- High latency
- Error rate spikes
- Failed deployments

---

12. Security Features

12.1 Credential Security

Feature	Implementation
Storage	macOS Keychain (encrypted)
Memory	Cleared after use
Transport	TLS 1.3 only
Validation	Format + connectivity check

12.2 Deployment Locks

Prevent concurrent deployments:

```
Deployment Lock: Active
  Acquired: 2024-12-25 10:30:00
  Owner: deployer@example.com
  Environment: production
  Expires: 2024-12-25 11:30:00
```

12.3 Audit Logging

All operations are logged:

```json
{
  "timestamp": "2024-12-25T10:30:00Z",
  "operation": "deployment.start",
  "user": "admin@example.com",
  "environment": "production",
  "version": "4.18.1",
  "status": "success",
  "duration_ms": 272000
}
```

12.4 Secret Detection

Pre-commit checks scan for:

- AWS access keys
- API keys
- Passwords
- Private keys

---

13. Troubleshooting

13.1 Common Issues

Deployment Fails at CDK Synth **Symptoms**: Deployment stops at synthesis phase

Solutions: 1. Check Node.js version: `node --version` (need 20.x) 2. Clear CDK cache: `rm -rf cdk.out` 3. Update CDK: `npm update -g aws-cdk` 4. Check TypeScript errors in `packages/infrastructure`

AWS Credentials Invalid **Symptoms**: "Invalid credentials" error

Solutions: 1. Verify access key format (starts with AKIA) 2. Check secret key hasn't expired 3. Verify IAM user is active 4. Test with AWS CLI: `aws sts get-caller-identity`

Health Check Timeout **Symptoms**: Services show unhealthy after deployment

Solutions: 1. Wait 2-3 minutes for cold start 2. Check CloudWatch logs for errors 3. Verify security group rules 4. Check VPC endpoint configuration

13.2 Log Locations

Log Type	Location
App Logs	`~/Library/Logs/RadiantDeployer/`
Deployment Logs	`~/Library/Application Support/RadiantDeployer/deployments/`

Log Type	Location
AWS Logs	CloudWatch Log Groups

13.3 Getting Help

1. **AI Assistant**: Built-in help
2. **Documentation**: This guide + online docs
3. **Support**: support@radiant.example.com

---

14. Reference

14.1 Keyboard Shortcuts

Shortcut	Action
+ D	Start deployment
+ R	Refresh status
+ S	Create snapshot
+ ,	Open settings
+ ?	Open AI assistant
+ L	View logs

14.2 CLI Commands

```
# Build and run from source
cd apps/swift-deployer
swift build -c release
swift run RadiantDeployer

# Run with specific config
swift run RadiantDeployer --environment prod --tier 3

# Headless deployment
swift run RadiantDeployer deploy --non-interactive
```

14.3 Environment Variables

Variable	Description
`RADIANT_ENV`	Override environment
`RADIANT_TIER`	Override tier
`RADIANT_DEBUG`	Enable debug logging
`RADIANT_AI_KEY`	Anthropic API key

14.4 File Locations

File	Location
Configuration	`~/Library/Application Support/RadiantDeployer/config.json`
Snapshots	`~/Library/Application Support/RadiantDeployer/snapshots/`
Logs	`~/Library/Logs/RadiantDeployer/`
Database	`~/Library/Application Support/RadiantDeployer/local.db`

Appendix A: IAM Policy Requirements

Minimum IAM permissions for deployment:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "s3:*",
        "lambda:*",
        "apigateway:*",
        "cognito-idp:*",
        "rds:*",
        "dynamodb:*",
        "sqs:*",
        "sns:*",
        "events:*",
        "logs:*",
        "iam:PassRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "ssm:*",
        "secretsmanager:*",
        "ecr:*",
        "ecs:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Appendix B: Glossary

Term	Definition
CDK	AWS Cloud Development Kit
Stack	CloudFormation stack deployed by CDK
Snapshot	Point-in-time backup of deployment
Lock-Step	Version consistency enforcement
Tier	Infrastructure sizing level (1-5)

Document Version: 4.18.1 Last Updated: December 2024