

Contents

RADIANT CDK Stack Dependencies	1
Stack Dependency Graph	1
Stack Definitions	2
Layer 1: Foundation	2
Layer 2: Core Infrastructure	2
Layer 3: Authentication	3
Layer 4: Application Services	3
Layer 5: AI Services	3
Layer 6: Specialized Services	3
CDK Implementation	3
Explicit Dependencies	3
Cross-Stack References	5
Deployment Order	5
Fresh Install	5
Update (with dependencies)	5
Selective Deployment	5
Rollback Considerations	5
Validation Script	6
Related Documentation	6

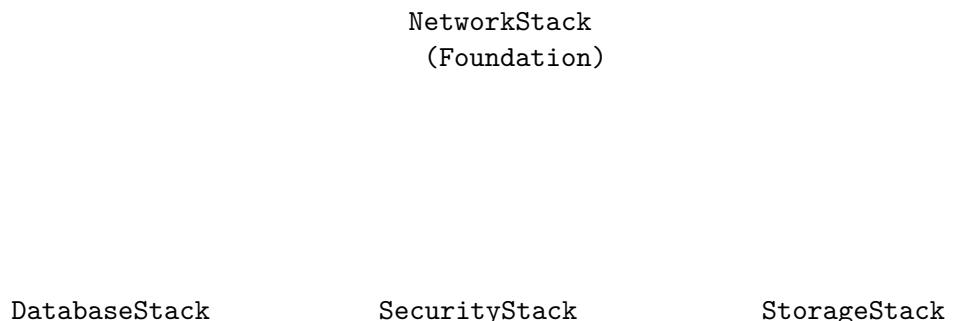
RADIANT CDK Stack Dependencies

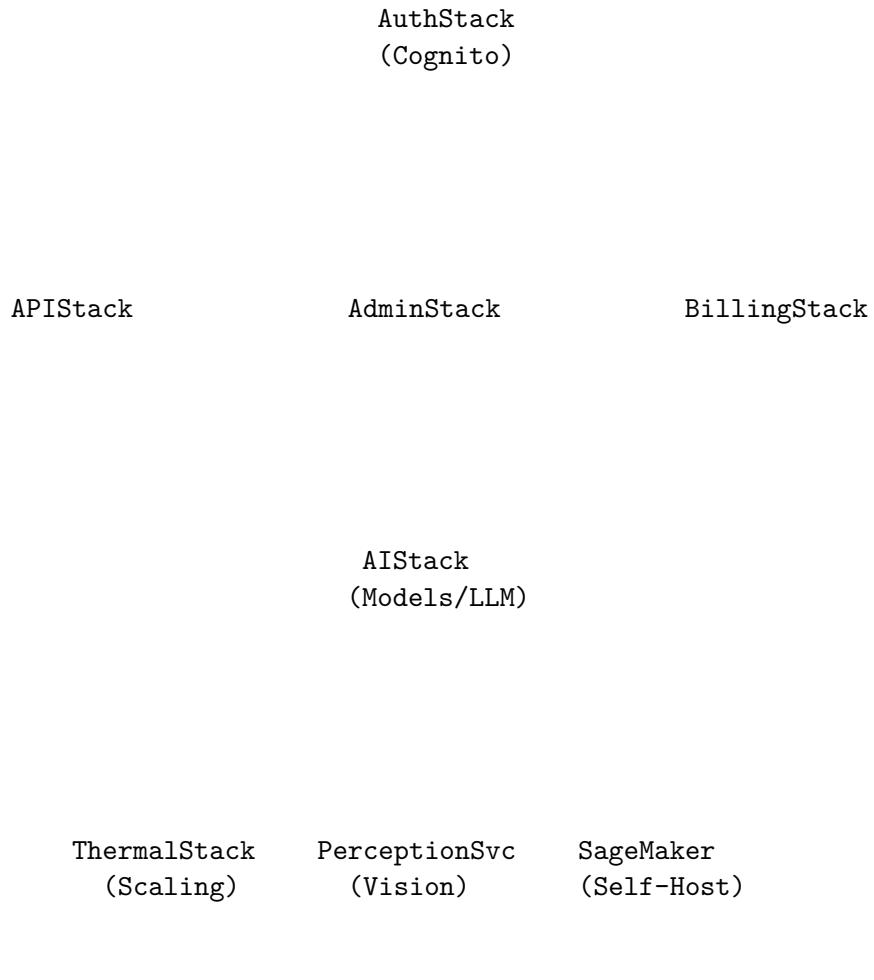
Technical Reference

Version: 4.18.1 | Last Updated: December 2024

This document defines the explicit dependency graph for RADIANT CDK stacks to ensure correct deployment ordering.

Stack Dependency Graph





Stack Definitions

Layer 1: Foundation

Stack	Purpose	Exports
NetworkStack	VPC, Subnets, NAT Gateways	VPC ID, Subnet IDs, Security Group IDs

Layer 2: Core Infrastructure

Stack	Purpose	Dependencies	Exports
DatabaseStack	Aurora PostgreSQL, RDS Proxy	NetworkStack	Cluster ARN, Secret ARN, Proxy Endpoint
SecurityStack	KMS Keys, WAF, GuardDuty	NetworkStack	Key ARNs, WAF ACL ARN

Stack	Purpose	Dependencies	Exports
StorageStack	S3 Buckets, CloudFront	NetworkStack	Bucket ARNs, Distribution ID

Layer 3: Authentication

Stack	Purpose	Dependencies	Exports
AuthStack	Cognito User/Identity Pools	Database, Security, Storage	Pool IDs, Client IDs

Layer 4: Application Services

Stack	Purpose	Dependencies	Exports
APIStack	API Gateway, Lambda handlers	Auth, Database, Security	API Endpoint, Lambda ARNs
AdminStack	Admin API, Dashboard hosting	Auth, Database, Security	Admin API Endpoint
BillingStack	Stripe integration, Usage tracking	Auth, Database	Billing API Endpoint

Layer 5: AI Services

Stack	Purpose	Dependencies	Exports
AISocket	LiteLLM, Model routing	API, Database, Security	AI Gateway Endpoint

Layer 6: Specialized Services

Stack	Purpose	Dependencies	Exports
ThermalStack	Model scaling, State management	AI, Database	Thermal API Endpoint
PerceptionStack	Computer vision pipeline	AI, Storage	Perception API Endpoint
SageMakerStack	Self-hosted model endpoints	AI, Network	Endpoint ARNs

CDK Implementation

Explicit Dependencies

```
// packages/infrastructure/lib/main.ts
```

```

import { App } from 'aws-cdk-lib';

const app = new App();

// Layer 1
const networkStack = new NetworkStack(app, 'Network', { env });

// Layer 2
const databaseStack = new DatabaseStack(app, 'Database', {
  env,
  vpc: networkStack.vpc,
});
databaseStack.addDependency(networkStack);

const securityStack = new SecurityStack(app, 'Security', {
  env,
  vpc: networkStack.vpc,
});
securityStack.addDependency(networkStack);

const storageStack = new StorageStack(app, 'Storage', {
  env,
  vpc: networkStack.vpc,
});
storageStack.addDependency(networkStack);

// Layer 3
const authStack = new AuthStack(app, 'Auth', {
  env,
  database: databaseStack,
  security: securityStack,
  storage: storageStack,
});
authStack.addDependency(databaseStack);
authStack.addDependency(securityStack);
authStack.addDependency(storageStack);

// Layer 4
const apiStack = new APIStack(app, 'API', {
  env,
  auth: authStack,
  database: databaseStack,
  security: securityStack,
});
apiStack.addDependency(authStack);

// ... continue for remaining stacks

```

Cross-Stack References

```
// Example: APIStack referencing DatabaseStack exports

export class APIStack extends Stack {
  constructor(scope: Construct, id: string, props: APIStackProps) {
    super(scope, id, props);

    // Use exports from DatabaseStack
    const clusterArn = props.database.clusterArn;
    const secretArn = props.database.secretArn;

    // Create Lambda with database access
    const handler = new Function(this, 'ApiHandler', {
      environment: {
        AURORA_CLUSTER_ARN: clusterArn,
        AURORA_SECRET_ARN: secretArn,
      },
    });
  }
}
```

Deployment Order

Fresh Install

```
# Deploy in dependency order
cdk deploy NetworkStack
cdk deploy DatabaseStack SecurityStack StorageStack --parallel
cdk deploy AuthStack
cdk deploy APIStack AdminStack BillingStack --parallel
cdk deploy AIStack
cdk deploy ThermalStack PerceptionStack SageMakerStack --parallel
```

Update (with dependencies)

```
# CDK handles ordering automatically when using addDependency
cdk deploy --all
```

Selective Deployment

```
# Deploy specific stack and its dependencies
cdk deploy AIStack --require-approval never
```

Rollback Considerations

Stack	Rollback Safe	Notes
NetworkStack	Caution	May affect all dependent stacks
DatabaseStack	Caution	Requires DB snapshot for data preservation
SecurityStack	Safe	KMS keys have deletion protection
StorageStack	Caution	S3 buckets may have data
AuthStack	Safe	Cognito pools preserved
APIStack	Safe	Stateless Lambda functions
AdminStack	Safe	Stateless
BillingStack	Caution	May have pending transactions
AIStack	Safe	Stateless routing
ThermalStack	Safe	State in database
SageMakerStack	Caution	May have running endpoints

Validation Script

```

#!/bin/bash
# tools/scripts/validate-stack-deps.sh

echo "Validating CDK stack dependencies..."

# Check for circular dependencies
cdk synth --quiet 2>&1 | grep -i "circular" && {
    echo " Circular dependency detected!"
    exit 1
}

# Verify deployment order
cdk diff --all 2>&1 | head -50

echo " Stack dependencies validated"

```

Related Documentation

- [Deployment Guide](#) - Full deployment procedures
- [Deployer Architecture](#) - Package and deployment flow