# Contents

# RADIANT v4.17.0 - Troubleshooting Guide

## Common Issues and Solutions

### CDK Deployment Failures

| Issue | Likely Cause | Solution |
| --- | --- | --- |
| Bootstrap failed | Wrong account/region | Verify `aws sts get-caller-identity` |
| Stack timeout | Slow resource creation | Check CloudFormation events in AWS Console |
| Resource limit | Service quota exceeded | Request quota increase via AWS Service Quotas |
| IAM permission denied | Insufficient permissions | Ensure IAM user has AdministratorAccess |
| Circular dependency | Stack references | Check stack dependencies in CDK code |
| Asset upload failed | S3 bucket permissions | Verify CDK bootstrap bucket exists |

### Aurora Database Issues

| Issue | Likely Cause | Solution |
|---|---|---|
| Connection refused | Security group rules | Verify Lambda SG can reach Aurora SG on port 5432 |
| Authentication failed | Wrong credentials | Check Secrets Manager for correct credentials |
| Connection timeout | Missing VPC endpoints | Add RDS VPC endpoint to private subnets |
| Too many connections | Connection exhaustion | Use RDS Proxy or increase max_connections |
| Slow queries | Missing indexes | Run EXPLAIN ANALYZE and add appropriate indexes |
| RLS blocking access | Tenant ID not set | Ensure `app.current_tenant_id` is set in session |

**Lambda Function Errors**

| Issue | Likely Cause | Solution |
|---|---|---|
| Cold start > 10s | VPC attachment | Use provisioned concurrency for critical functions |
| Timeout | Slow downstream services | Increase timeout, check DB/API latency |
| Out of memory | Large payloads/responses | Increase memory allocation (also increases CPU) |
| Permission denied | IAM role misconfigured | Check Lambda execution role policies |
| Module not found | Missing dependency | Verify all dependencies in package.json |
| Handler not found | Incorrect handler path | Check function configuration in CDK |

**LiteLLM / ECS Issues**

| Issue | Likely Cause | Solution |
|---|---|---|
| 503 Service Unavailable | ECS task unhealthy | Check ECS service events and task logs |
| Provider timeout | Invalid API key | Verify provider secrets in Secrets Manager |
| Rate limited | Too many requests | Implement exponential backoff retry |
| Wrong model response | Model misconfigured | Check config.yaml model mappings |
| Container crashes | Memory exhaustion | Increase task memory in CDK |

| Issue | Likely Cause | Solution |
| --- | --- | --- |
| No healthy targets | Health check failing | Verify health check endpoint returns 200 |

**SageMaker Issues (Tier 3+)**

| Issue | Likely Cause | Solution |
| --- | --- | --- |
| Endpoint failed to create | Insufficient capacity | Try different instance type or region |
| InvocationError | Model loading failed | Check CloudWatch logs for model errors |
| Slow cold start | Large model size | Use warm pools or smaller model variant |
| Capacity error | Instance quota reached | Request SageMaker quota increase |
| Timeout | Long inference time | Increase endpoint timeout or optimize model |

**Cognito Authentication Issues**

| Issue | Likely Cause | Solution |
| --- | --- | --- |
| Invalid grant | Expired refresh token | Re-authenticate user |
| User not confirmed | Email not verified | Check email or manually confirm user |
| MFA required | MFA not set up | Complete MFA setup flow |
| Invalid client | Wrong client ID | Verify app client ID in configuration |
| Callback URL mismatch | URL not whitelisted | Add URL to allowed callbacks in Cognito |

**Admin Dashboard Issues**

| Issue | Likely Cause | Solution |
| --- | --- | --- |
| 403 Forbidden | CloudFront OAC issue | Verify S3 bucket policy allows CloudFront |
| API calls fail | CORS configuration | Check API Gateway CORS settings |
| Login redirect loop | Cookie domain mismatch | Verify cookie domain matches site domain |
| Blank page | Build error | Check `next build` output for errors |
| Slow load | Large bundle size | Enable code splitting and lazy loading |

## Log Locations

| Component | CloudWatch Log Group |
| --- | --- |
| API Gateway | /aws/api-gateway/radiant-{env}-api |
| Lambda Functions | /aws/lambda/Radiant-{env}-* |
| LiteLLM (ECS) | /ecs/radiant-{env}-litellm |
| SageMaker Endpoints | /aws/sagemaker/Endpoints/radiant-* |
| Aurora PostgreSQL | /aws/rds/cluster/radiant-{env}/postgresql |
| CloudFront | Standard CloudFront logs in S3 |

## Viewing Logs

```
# Tail Lambda logs in real-time
aws logs tail /aws/lambda/Radiant-dev-router --follow

# View ECS logs
aws logs tail /ecs/radiant-dev-litellm --follow

# Search logs for errors
aws logs filter-log-events \
  --log-group-name /aws/lambda/Radiant-dev-router \
  --filter-pattern "ERROR" \
  --start-time $(date -d '1 hour ago' +%s)000
```

## Health Check Endpoints

```
# Platform API health
curl https://api.YOUR_DOMAIN/health
# Expected: {"status":"healthy","version":"4.17.0"}

# LiteLLM health
curl https://api.YOUR_DOMAIN/v2/litellm/health
# Expected: {"status":"healthy"}

# Admin API health
curl https://admin-api.YOUR_DOMAIN/health

# Model registry status
curl https://api.YOUR_DOMAIN/v2/models/status
```

## Emergency Procedures

### Database Restore from Snapshot

```
# List available snapshots
aws rds describe-db-cluster-snapshots \
  --db-cluster-identifier radiant-prod-cluster \
  --query 'DBClusterSnapshots[*].[DBClusterSnapshotIdentifier,SnapshotCreateTime]' \
  --output table

# Restore from snapshot
aws rds restore-db-cluster-from-snapshot \
  --db-cluster-identifier radiant-prod-restored \
  --snapshot-identifier your-snapshot-id \
  --engine aurora-postgresql \
  --vpc-security-group-ids sg-xxx \
  --db-subnet-group-name radiant-prod-db-subnet
```

### Point-in-Time Recovery

```
aws rds restore-db-cluster-to-point-in-time \
  --source-db-cluster-identifier radiant-prod-cluster \
  --db-cluster-identifier radiant-prod-recovered \
  --restore-to-time "2024-12-20T10:00:00Z" \
  --vpc-security-group-ids sg-xxx \
  --db-subnet-group-name radiant-prod-db-subnet
```

### Rollback CDK Deployment

```
# Rollback to previous deployment
cd packages/infrastructure
npx cdk deploy Radiant-prod-API \
  --context environment=prod \
  --context tier=3 \
  --rollback
```

### Disable Problematic Model

```
-- Connect to Aurora and run:
UPDATE models
SET status = 'disabled',
    disabled_reason = 'Emergency disable due to errors',
    updated_at = NOW()
WHERE model_id = 'problematic-model-id';
```

### Force Scale Down SageMaker

```
# Scale endpoint to 0 instances
aws sagemaker update-endpoint-weights-and-capacities \
```

```
    --endpoint-name radiant-prod-model-endpoint \
    --desired-weights-and-capacities '[{"VariantName":"AllTraffic","DesiredInstanceCount":0}]'
```

---

## Performance Benchmarks

| Metric | Target | Acceptable | Action if Exceeded |
|---|---|---|---|
| API Gateway p50 latency | < 50ms | < 100ms | Check Lambda cold starts |
| API Gateway p99 latency | < 200ms | < 500ms | Enable provisioned concurrency |
| Chat streaming start | < 500ms | < 1s | Check LiteLLM/provider latency |
| Admin dashboard load | < 2s | < 3s | Optimize bundle, enable CDN caching |
| Model warm-up time | < 3 min | < 5 min | Use larger instance or warm pools |
| Aurora query latency | < 10ms | < 50ms | Add indexes, optimize queries |

---

## Support Checklist

When reporting issues, include:

1. **Environment**: dev/staging/prod
2. **Tier**: 1-5
3. **Error message**: Full error text
4. **Request ID**: From response headers
5. **Timestamp**: When the error occurred
6. **Steps to reproduce**: What actions led to the error
7. **Relevant logs**: CloudWatch log excerpts

---

## Useful AWS CLI Commands

```
# Check stack status
aws cloudformation describe-stacks --stack-name Radiant-dev-API \
  --query 'Stacks[0].StackStatus'

# List recent CloudFormation events
aws cloudformation describe-stack-events --stack-name Radiant-dev-API \
  --query 'StackEvents[0:10].[Timestamp,ResourceStatus,ResourceType,LogicalResourceId]' \
  --output table

# Check Lambda function configuration
```

```
aws lambda get-function-configuration --function-name Radiant-dev-router

# List ECS services
aws ecs list-services --cluster radiant-dev-cluster

# Describe ECS service
aws ecs describe-services --cluster radiant-dev-cluster \
  --services radiant-dev-litellm

# Check Secrets Manager secret
aws secretsmanager get-secret-value --secret-id radiant/dev/db-credentials \
  --query 'SecretString' --output text | jq .
```