Contents

Incident Response Runbook

Overview

This runbook provides procedures for responding to incidents in the RADIANT platform.

Severity Levels

Level	Description	Response Time	Examples
SEV1	Critical - Service down	15 minutes	Complete outage, data loss
SEV2	Major - Significant degradation	30 minutes	Partial outage, high error rates
SEV3	Minor - Limited impact	2 hours	Single feature broken
SEV4	Low - Minimal impact	Next business day	Cosmetic issues

Initial Response

1. Acknowledge the Alert

```
# Check CloudWatch dashboard
aws cloudwatch get-dashboard --dashboard-name radiant-production-dashboard

# Check recent alarms
aws cloudwatch describe-alarms --state-value ALARM
```

2. Assess Impact

- ☐ How many users are affected?
- ☐ Which services are impacted?
- ☐ Is data at risk?
- ☐ What's the business impact?

3. Communicate

- SEV1/SEV2: Immediately notify on-call and stakeholders
- Update status page if available
- Create incident channel

Common Incidents

API Gateway 5XX Errors

Symptoms: - High 5XX error rate in CloudWatch - Users reporting API failures

Investigation:

```
# Check Lambda logs
aws logs filter-log-events \
  --log-group-name /aws/lambda/radiant-production-router \
  --filter-pattern "ERROR" \
  --start-time $(date -d '1 hour ago' +%s000)

# Check for throttling
aws cloudwatch get-metric-statistics \
  --namespace AWS/Lambda \
  --metric-name Throttles \
  --dimensions Name=FunctionName,Value=radiant-production-router \
  --start-time $(date -d '1 hour ago' -Iseconds) \
  --end-time $(date -Iseconds) \
  --period 60 \
  --statistics Sum
```

Resolution: 1. Check if Lambda is hitting memory/timeout limits 2. Check database connectivity 3. Review recent deployments 4. Scale up if needed

Database Connection Issues

Symptoms: - Connection timeout errors in Lambda logs - High database connection count

Investigation:

```
# Check connection count
aws cloudwatch get-metric-statistics \
  --namespace AWS/RDS \
  --metric-name DatabaseConnections \
  --dimensions Name=DBClusterIdentifier,Value=radiant-production \
  --start-time $(date -d '1 hour ago' -Iseconds) \
  --end-time $(date -Iseconds) \
  --period 60 \
  --statistics Average

# Check for long-running queries (via admin dashboard)
```

Resolution: 1. Check for connection leaks in Lambda 2. Increase connection pool size 3. Scale database if CPU is high 4. Kill long-running queries if necessary

High Latency

Symptoms: - API p99 latency > 5 seconds - User complaints about slowness

Investigation:

```
# Check Lambda duration
aws cloudwatch get-metric-statistics \
  --namespace AWS/Lambda \
  --metric-name Duration \
  --dimensions Name=FunctionName,Value=radiant-production-router \
  --start-time $(date -d '1 hour ago' -Iseconds) \
  --end-time $(date -Iseconds) \
  --period 60 \
  --statistics p99

# Check database latency
aws cloudwatch get-metric-statistics \
  --namespace AWS/RDS \
  --metric-name ReadLatency \
  --dimensions Name=DBClusterIdentifier,Value=radiant-production \
  --start-time $(date -d '1 hour ago' -Iseconds) \
  --end-time $(date -Iseconds) \
  --period 60 \
  --statistics Average
```

Resolution: 1. Check for slow database queries 2. Review cold start times 3. Check for external API latency (AI providers) 4. Scale up Lambda memory if CPU-bound

Authentication Failures

Symptoms: - Users cannot log in - Token validation failures

Investigation:

```
# Check Cognito metrics
aws cloudwatch get-metric-statistics \
  --namespace AWS/Cognito \
  --metric-name SignInSuccesses \
  --dimensions Name=UserPool,Value=radiant-production-users \
  --start-time $(date -d '1 hour ago' -Iseconds) \
  --end-time $(date -Iseconds) \
  --period 300 \
  --statistics Sum
```

Resolution: 1. Check Cognito service health 2. Verify JWT token configuration 3. Check for clock skew issues 4. Review recent Cognito changes

Storage Quota Exceeded

Symptoms: - Upload failures - Storage billing alerts

Investigation:

```
# Check S3 bucket size
aws s3 ls s3://radiant-storage-production --summarize --recursive | tail -2

# Check storage usage in database
# Use admin dashboard Storage page
```

Resolution: 1. Identify large files/tenants 2. Contact affected tenants 3. Clean up orphaned files 4. Increase quota if appropriate

Rollback Procedures

Lambda Rollback

```
# List versions
aws lambda list-versions-by-function \
  --function-name radiant-production-router

# Rollback to previous version
aws lambda update-alias \
  --function-name radiant-production-router \
  --name live \
  --function-version <previous-version>
```

Database Rollback

WARNING: Database rollbacks are dangerous. Follow dual-admin approval.

1. Create a new migration to undo changes

2. Get dual-admin approval via Migration Approval page
3. Execute the migration
4. Verify data integrity

CDK Rollback

```
# Check CloudFormation events
aws cloudformation describe-stack-events \
  --stack-name RadiantProductionApi

# Rollback to previous state
aws cloudformation continue-update-rollback \
  --stack-name RadiantProductionApi
```

Post-Incident

1. Resolve Alert

```
# Set alarm to OK (if manually resolved)
aws cloudwatch set-alarm-state \
  --alarm-name radiant-production-api-5xx-errors \
  --state-value OK \
  --state-reason "Manually resolved"
```

2. Document

Create an incident report: - Timeline of events - Root cause analysis - Actions taken - Lessons learned - Follow-up items

3. Review

- Schedule post-mortem for SEV1/SEV2
- Update runbooks with new learnings
- Create tickets for improvements

Contacts

Role	Contact	Escalation
On-Call Engineer	PagerDuty	Auto-escalate after 15 min
Engineering Lead	@engineering-lead	SEV1/SEV2
Security	@security-team	Security incidents
Product	@product-team	Business impact

Useful Links

- CloudWatch Dashboard
- Admin Dashboard
- Status Page
- Incident Slack Channel