Contents

RADIANT Incident Response Runbook

Version: {{RADIANT_VERSION}} **Last Updated**: {{BUILD_DATE}}

---

1. Incident Classification

Severity	Definition	Response Time	Examples
P1 - Critical	Platform down, all users affected	15 minutes	Database failure, Auth down
P2 - High	Major feature broken, many users affected	1 hour	Payment processing failed
P3 - Medium	Feature degraded, some users affected	4 hours	Slow response times
P4 - Low	Minor issue, workaround available	24 hours	UI bug, typo

---

2. Incident Response Process

2.1 Detection

1. **Automated Alerts**: CloudWatch, PagerDuty
2. **User Reports**: Support tickets, status page
3. **Monitoring**: Dashboard anomalies

2.2 Triage

```
       Incident Detected




   Assess Impact & Severity
- Users affected?
- Revenue impact?
- Data at risk?




   Assign Severity (P1-P4)




   Notify Stakeholders
```

2.3 Response Actions

P1 - Critical 1. Page on-call engineer immediately 2. Create incident channel (#incident-YYYYMMDD) 3. Update status page to “Investigating” 4. Assemble incident team 5. Begin investigation

P2 - High 1. Notify on-call engineer 2. Create incident ticket 3. Update status page if customer-facing 4. Begin investigation within 1 hour

---

3. Common Incidents

3.1 Database Connection Failure

Symptoms: - 500 errors on API requests - “Connection refused” in logs

Investigation:

```
# Check Aurora cluster status
aws rds describe-db-clusters --db-cluster-identifier radiant-cluster

# Check security group rules
aws ec2 describe-security-groups --group-ids sg-xxx

# Verify secrets
aws secretsmanager get-secret-value --secret-id radiant/db-credentials
```

Resolution: 1. Check if cluster is available 2. Verify security group allows Lambda access 3. Check if credentials rotated 4. Restart affected Lambda functions

3.2 High Latency

Symptoms: - Response times > 5 seconds - Timeout errors

Investigation:

```
# Check Lambda duration metrics
aws cloudwatch get-metric-statistics \
  --namespace AWS/Lambda \
  --metric-name Duration \
  --dimensions Name=FunctionName,Value=radiant-api \
  --start-time $(date -u -d '1 hour ago' +%Y-%m-%dT%H:%M:%SZ) \
  --end-time $(date -u +%Y-%m-%dT%H:%M:%SZ) \
  --period 300 \
  --statistics Average,Maximum

# Check external provider health
curl -w "@curl-format.txt" https://api.openai.com/v1/models
```

Resolution: 1. Identify slow component (DB, provider, processing) 2. Scale resources if needed 3. Enable caching if appropriate 4. Contact provider if external issue

3.3 Provider Outage

Symptoms: - Errors from specific AI provider - Brain Router selecting alternatives

Investigation:

```
# Check provider health dashboard
# Review error rates by provider in CloudWatch

aws cloudwatch get-metric-statistics \
  --namespace RADIANT/Providers \
  --metric-name ErrorRate \
  --dimensions Name=Provider,Value=openai \
  --start-time $(date -u -d '1 hour ago' +%Y-%m-%dT%H:%M:%SZ) \
  --end-time $(date -u +%Y-%m-%dT%H:%M:%SZ) \
  --period 60 \
  --statistics Average
```

Resolution: 1. Verify outage on provider status page 2. Brain Router should auto-failover 3. Update internal status page 4. Monitor for resolution 5. Post-incident: review failover effectiveness

---

4. Communication Templates

4.1 Status Page - Investigating

```
Investigating Increased Error Rates
```

```
We are currently investigating reports of increased error rates
affecting [service]. Our team is actively working to identify
and resolve the issue.

We will provide updates every 30 minutes or as we have new information.

Posted: [TIME] UTC
```

4.2 Status Page - Identified

```
Issue Identified - [Brief Description]

We have identified the cause of [issue]. The problem is related to
[root cause summary]. Our team is implementing a fix.

Estimated resolution: [TIME] UTC

Posted: [TIME] UTC
```

4.3 Status Page - Resolved

```
Resolved - [Brief Description]

The issue affecting [service] has been resolved.
[Brief explanation of fix].

Total duration: [X] hours [Y] minutes
Impact: [description of impact]

We apologize for any inconvenience caused.

Posted: [TIME] UTC
```

---

5. Post-Incident

5.1 Post-Mortem Template

```markdown
# Incident Post-Mortem: [Title]

**Date**: [Date]
**Duration**: [Start] - [End] ([Duration])
**Severity**: P[X]
**Author**: [Name]

## Summary
[1-2 sentence summary]
```

```
## Impact
- Users affected: [number]
- Revenue impact: [amount]
- SLA impact: [yes/no]

## Timeline
| Time (UTC) | Event |
|------------|-------|
| HH:MM | [Event] |

## Root Cause
[Detailed explanation]

## Resolution
[How it was fixed]

## Action Items
| Item | Owner | Due Date | Status |
|------|-------|----------|--------|
| [Action] | [Name] | [Date] | Open |

## Lessons Learned
- [Lesson 1]
- [Lesson 2]
```

5.2 Review Meeting

Schedule within 48 hours of resolution: - Review timeline - Identify root cause - Assign action items - Update runbooks if needed

---

6. Contacts

Role	Contact
On-Call Engineer	PagerDuty
Platform Lead	[email]
Security Team	security@radiant.ai
Customer Success	support@radiant.ai

This runbook is part of the RADIANT v{{RADIANT_VERSION}} documentation.