Contents

RADIANT Scaling Runbook

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

---

1. Auto-Scaling Configuration

1.1 Lambda Functions

Function	Min	Max	Scaling Trigger
API Handler	10	1000	Concurrent requests
Brain Router	5	500	Queue depth
Webhook Handler	2	100	Event count

1.2 Aurora PostgreSQL

Setting	Value	Notes
Min ACUs	2	Development
Max ACUs	64	Production
Scale-out cooldown	5 minutes	
Scale-in cooldown	15 minutes	

1.3 SageMaker Endpoints

Model Category	Min	Max	Scale Trigger
Vision Models	0	5	Invocations/min
LLM Models	0	3	Queue depth
Audio Models	0	2	Invocations/min

---

2. Manual Scaling Procedures

2.1 Pre-Event Scaling

Before expected high traffic:

```
# Scale Aurora to maximum
aws rds modify-db-cluster \
  --db-cluster-identifier radiant-cluster \
  --serverless-v2-scaling-configuration MinCapacity=16,MaxCapacity=64

# Pre-warm Lambda functions
for i in {1..100}; do
  aws lambda invoke \
    --function-name radiant-api \
    --invocation-type Event \
    --payload '{"warmup": true}' \
    /dev/null &
done
wait

# Scale SageMaker endpoints
aws sagemaker update-endpoint-weights-and-capacities \
  --endpoint-name radiant-vision \
  --desired-weights-and-capacities '[{"VariantName":"AllTraffic","DesiredInstanceCount":3}]'
```

2.2 Emergency Scaling

During unexpected traffic spike:

```
# Increase Lambda concurrency limit
aws lambda put-function-concurrency \
  --function-name radiant-api \
  --reserved-concurrent-executions 2000

# Scale Aurora immediately
aws rds modify-db-cluster \
  --db-cluster-identifier radiant-cluster \
  --serverless-v2-scaling-configuration MinCapacity=32,MaxCapacity=128 \
  --apply-immediately
```

---

3. Monitoring Scaling Events

3.1 Key Metrics

Metric	Warning	Critical
Lambda Concurrent Executions	70% of limit	90% of limit
Aurora ACU Utilization	80%	95%

Metric	Warning	Critical
API Gateway 5xx Rate	1%	5%

3.2 CloudWatch Alarms

```
# List scaling alarms
aws cloudwatch describe-alarms \
  --alarm-name-prefix "radiant-scaling"

# Check alarm history
aws cloudwatch describe-alarm-history \
  --alarm-name "radiant-api-high-concurrency" \
  --history-item-type StateUpdate
```

---

4. Cost Considerations

Resource	Cost Factor	Optimization
Lambda	Duration × Memory	Right-size memory
Aurora	ACU-hours	Scale down off-peak
SageMaker	Instance-hours	Use spot instances
API Gateway	Request count	Enable caching

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