Contents

Cato Scaling Runbook

Overview

This runbook covers scaling Cato infrastructure to handle increased load.

Scaling Triggers

Automatic Scaling

The following components auto-scale based on metrics:

Component	Metric	Scale Up	Scale Down
Shadow Self (SageMaker)	GPU Utilization > 80%	+10 instances	-10 instances when < 40%
Ray Serve (EKS)	Requests/replica > 10	+5 replicas	-5 replicas when < 3

Component	Metric	Scale Up	Scale Down
NLI Model (SageMaker MME)	Latency p99 > 100ms	+2 instances	-2 when < 50ms
ElastiCache	Memory > 80%	Vertical scale	N/A

Manual Scaling Triggers

Scale manually when: - Daily traffic increases by 50%+ - Monthly costs exceed budget by 20%+ - Latency SLOs are at risk

Scaling Procedures

1. Shadow Self (SageMaker)

Scale Up:

```
aws sagemaker update-endpoint-weights-and-capacities \
  --endpoint-name cato-shadow-self \
  --desired-weights-and-capacities \
    VariantName=primary,DesiredInstanceCount=50
```

Check Status:

```
aws sagemaker describe-endpoint --endpoint-name cato-shadow-self
```

Timing: Instance provisioning takes ~5-10 minutes.

2. Ray Serve (EKS)

Scale Up:

```
kubectl scale deployment cato-orchestrator -n cato --replicas=100
```

Check Status:

```
kubectl get pods -n cato -l app=cato-orchestrator
```

Update Auto-Scaling:

```
kubectl patch hpa cato-orchestrator-hpa -n cato \
  --patch '{"spec":{"maxReplicas":200}}'
```

3. ElastiCache (Valkey)

Vertical Scale (downtime required):

```
aws elasticache modify-replication-group \
  --replication-group-id cato-cache-production \
  --cache-node-type cache.r7g.2xlarge \
  --apply-immediately
```

Add Read Replicas:

```
aws elasticache increase-replica-count \
  --replication-group-id cato-cache-production \
  --new-replica-count 5 \
  --apply-immediately
```

4. DynamoDB

DynamoDB with on-demand billing auto-scales. For provisioned capacity:

Update Capacity:

```
aws dynamodb update-table \
  --table-name cato-semantic-memory-production \
  --provisioned-throughput ReadCapacityUnits=10000,WriteCapacityUnits=5000
```

5. OpenSearch Serverless

OpenSearch Serverless auto-scales. To increase max capacity:

```
aws opensearchserverless update-collection \
  --id <collection-id> \
  --description "Increased capacity"
```

Scaling by User Scale

100K → 1M Users

1. **Shadow Self**: 5 → 50 instances
2. **Ray Serve**: 10 → 50 replicas
3. **ElastiCache**: cache.r7g.xlarge → cache.r7g.2xlarge
4. **Budget**: $500 → $2,000/month

```
# Terraform variable update
cd infrastructure/terraform/environments/production
terraform apply -var="shadow_self_min_instances=50" -var="cache_node_type=cache.r7g.2xlarge"
```

1M → 10M Users

1. **Shadow Self**: 50 → 250 instances
2. **Ray Serve**: 50 → 100 replicas
3. **ElastiCache**: Add cluster mode, 6 shards
4. **Multi-region**: Enable replicas in eu-west-1, ap-northeast-1
5. **Budget**: $2,000 → $100,000/month

```
# Enable global tables
terraform apply -var="enable_global_tables=true" -var="replica_regions=[\"eu-west-1\",\"ap-nor
```

Cost Optimization During Scaling

Use Savings Plans

At 10M users, SageMaker is ~$275K/month. With 3-year Savings Plan: - **Savings**: 64% (~$100K/month) - **Commitment**: 3-year term

```
# Create Savings Plan via AWS Console or API
aws savingsplans create-savings-plan \
  --savings-plan-offering-id <offering-id> \
  --commitment 100000 \
  --savings-plan-type Compute
```

Use Spot Instances for Batch

Night-mode curiosity can use Spot instances: - **Savings**: 70% on compute - **Risk**: Interruption (acceptable for batch)

Right-Size Instances

Monitor and right-size monthly:

```
aws compute-optimizer get-ec2-instance-recommendations \
  --filters name=Finding,values=OPTIMIZED
```

Monitoring During Scale

Key Metrics to Watch

Metric	Warning	Critical
Shadow Self Latency p99	$>$ 300ms	$>$ 500ms
Cache Hit Rate	$<$ 75%	$<$ 60%
Error Rate	$>$ 0.5%	$>$ 1%
Budget Burn Rate	$>$ 120%	$>$ 150%

Dashboards

- **Cato Overview**: All key metrics
- **Cost Explorer**: Real-time spend

Alerts

Configure PagerDuty/Slack alerts for critical thresholds.

Rollback Procedures

If scaling causes issues:

Rollback SageMaker

```
aws sagemaker update-endpoint-weights-and-capacities \
  --endpoint-name cato-shadow-self \
  --desired-weights-and-capacities VariantName=primary,DesiredInstanceCount=10
```

Rollback EKS

```
kubectl rollout undo deployment/cato-orchestrator -n cato
```

Rollback Terraform

```
cd infrastructure/terraform/environments/production
git checkout HEAD~1 -- *.tfvars
terraform apply
```

Contact

- **On-call**: #cato-oncall
- **Escalation**: consciousness-team@thinktank.ai