Contents

Cato Deployment Runbook

Prerequisites

Before deploying Cato, ensure:

1. **AWS Account** with sufficient limits:

   - SageMaker ml.g5.2xlarge: 300 instances
   - EKS node groups: 100 nodes
   - DynamoDB on-demand capacity

2. **Terraform** v1.5+ installed

3. **kubectl** configured for EKS

4. **Docker** for building custom containers

5. **AWS CLI** configured with appropriate credentials

Deployment Phases

Phase 1: Infrastructure (Terraform)

```
# Navigate to Cato infrastructure
cd infrastructure/terraform/environments/production

# Initialize Terraform
terraform init

# Plan deployment
terraform plan -out=plan.tfplan

# Apply infrastructure
terraform apply plan.tfplan
```

This creates: - VPC with public/private subnets - EKS cluster for Ray Serve - SageMaker endpoints (Shadow Self, NLI) - DynamoDB Global Tables - OpenSearch Serverless collections - Neptune cluster - ElastiCache clusters - Kinesis streams - EventBridge rules - Step Functions workflows

Phase 2: Container Images

```
# Build Shadow Self container
cd infrastructure/docker/shadow-self
docker build -t cato-shadow-self:latest .

# Push to ECR
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin $E
docker tag cato-shadow-self:latest $ECR_REPO/cato-shadow-self:latest
docker push $ECR_REPO/cato-shadow-self:latest

# Build NLI container
cd ../nli-model
docker build -t cato-nli:latest .
docker push $ECR_REPO/cato-nli:latest

# Build Ray Serve orchestrator
cd ../orchestrator
docker build -t cato-orchestrator:latest .
docker push $ECR_REPO/cato-orchestrator:latest
```

Phase 3: Model Deployment

```
# Download Llama-3-8B model
python scripts/download_model.py --model meta-llama/Meta-Llama-3-8B-Instruct --output s3://cato

# Download DeBERTa-MNLI model
python scripts/download_model.py --model microsoft/deberta-large-mnli --output s3://cato-models
```

```
# Deploy SageMaker endpoints
python scripts/deploy_sagemaker.py --environment production
```

Phase 4: Ray Serve Deployment

```
# Configure kubectl for EKS
aws eks update-kubeconfig --name cato-eks --region us-east-1

# Deploy Ray Serve
kubectl apply -f k8s/ray-serve/

# Verify deployment
kubectl get pods -n cato
kubectl get svc -n cato
```

Phase 5: Database Initialization

```
# Initialize DynamoDB tables
python scripts/init_dynamodb.py --environment production

# Initialize OpenSearch indices
python scripts/init_opensearch.py --environment production

# Initialize Neptune graph
python scripts/init_neptune.py --environment production

# Seed domain knowledge (800+ domains)
python scripts/seed_knowledge.py --environment production
```

Phase 6: Verification

```
# Health check
curl https://api.cato.thinktank.ai/health

# Test dialogue
curl -X POST https://api.cato.thinktank.ai/v1/dialogue \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"message": "Hello, Cato!"}'

# Check metrics
aws cloudwatch get-metric-data --cli-input-json file://scripts/health_check_metrics.json
```

Configuration

Environment Variables

Variable	Description	Example
`CATO_ENV`	Environment name	`production`
`AWS_REGION`	Primary region	`us-east-1`
`SHADOW_SELF_ENDPOINT`	SageMaker endpoint	`cato-shadow-self`
`NLI_ENDPOINT`	NLI SageMaker endpoint	`cato-nli-mme`
`CACHE_HOST`	ElastiCache host	`cato-cache.xxx.use1.cache.amazonaws`
`DYNAMODB_TABLE_SEMANTIC`	Semantic memory table	`cato-semantic-memory`
`OPENSEARCH_ENDPOINT`	OpenSearch endpoint	`https://cato-episodic.xxx.us-east-1`
`NEPTUNE_ENDPOINT`	Neptune endpoint	`cato-graph.xxx.us-east-1.neptune.a`

Budget Configuration

Set via Radiant Admin Dashboard or directly in DynamoDB:

```
aws dynamodb put-item \
  --table-name cato-config \
  --item '{
    "pk": {"S": "CONFIG"},
    "sk": {"S": "BUDGET"},
    "monthlyLimit": {"N": "500"},
    "dailyExplorationLimit": {"N": "15"},
    "nightStartHour": {"N": "2"},
    "nightEndHour": {"N": "6"}
  }'
```

Rollback Procedures

Rollback SageMaker Endpoint

```
# List endpoint versions
aws sagemaker list-endpoint-configs --name-contains cato-shadow-self

# Update endpoint to previous config
aws sagemaker update-endpoint \
  --endpoint-name cato-shadow-self \
  --endpoint-config-name cato-shadow-self-config-v1
```

Rollback Ray Serve

```
# Rollback to previous deployment
kubectl rollout undo deployment/cato-orchestrator -n cato

# Verify rollback
kubectl rollout status deployment/cato-orchestrator -n cato
```

Rollback DynamoDB

DynamoDB Global Tables support point-in-time recovery:

```
aws dynamodb restore-table-to-point-in-time \
  --source-table-name cato-semantic-memory \
  --target-table-name cato-semantic-memory-restored \
  --restore-date-time 2024-01-15T00:00:00Z
```

Monitoring

Key Dashboards

1. **Cato Overview** - CloudWatch dashboard with all key metrics
2. **Inference Latency** - SageMaker endpoint latencies
3. **Cache Performance** - ElastiCache hit rates
4. **Memory Usage** - DynamoDB/OpenSearch capacity
5. **Budget Tracking** - Daily/monthly spend

Alerts

Alert	Threshold	Action
High Latency	p99 > 2s	Scale up SageMaker
Low Cache Hit Rate	< 70%	Investigate query patterns
Budget Exceeded	> 90% monthly	Enter emergency mode
Error Rate	> 1%	Investigate logs
Shadow Self Unhealthy	> 3 failures	Restart endpoint

Troubleshooting

Shadow Self Not Responding

1. Check endpoint status:

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

2. Check CloudWatch logs:

   ```
   aws logs tail /aws/sagemaker/Endpoints/cato-shadow-self --follow
   ```

3. Restart endpoint if needed:

   ```
   aws sagemaker update-endpoint --endpoint-name cato-shadow-self --endpoint-config-name cato
   ```

High Latency

1. Check cache hit rate - low rate means more LLM calls
2. Check SageMaker instance utilization
3. Check for Bedrock throttling
4. Scale up instances if needed

Memory Issues

1. Check DynamoDB consumed capacity
2. Check OpenSearch shard health

3. Verify DAX cluster status
4. Check for hot partitions

Maintenance Windows

- **Nightly**: 2-6 AM UTC (night mode, lower traffic)
- **Weekly**: Sunday 4 AM UTC (major updates)
- **Monthly**: First Sunday of month (infrastructure updates)

Contact

- **On-call**: #cato-oncall Slack channel
- **Escalation**: consciousness-team@thinktank.ai
- **AWS Support**: Enterprise Support case