# AWS Monitoring Setup Guide

Generated on: 9/3/2025

## AWS Monitoring and Alerting Setup Guide

### Table of Contents



### CloudWatch Dashboard Configuration

To create a comprehensive CloudWatch dashboard for your multi-tier web application, follow these steps:

1. Open the CloudWatch console
2. Click on "Dashboards" in the left navigation pane
3. Click "Create dashboard"
4. Name your dashboard (e.g., "Production-Web-App-Dashboard")
5. Add the following widgets to your dashboard:

   - Line graphs for key metrics (listed in the next section)
   - Text widgets for important information and links
   - Alarm status widgets for critical alarms
   - Log insights widgets for application logs

6. Arrange the widgets in a logical order, grouping related metrics together

### Key Metrics and Alert Thresholds

#### Route 53

Metric: HealthCheckStatus

Threshold: < 1 for 1 minute

Action: Critical alert

### CloudFront

Metric: 5xxErrorRate

Threshold: > 1% for 5 minutes

Action: Critical alert

### Elastic Load Balancing

Metric: UnHealthyHostCount

Threshold: > 0 for 5 minutes

Action: Warning alert

Metric: TargetResponseTime

Threshold: > 1 second for 15 minutes

Action: Warning alert

### EC2

Metric: CPUUtilization

Threshold: > 80% for 15 minutes

Action: Warning alert

Metric: StatusCheckFailed

Threshold: > 0 for 5 minutes

Action: Critical alert

### Auto Scaling

Metric: GroupInServiceInstances

Threshold: < Desired Capacity for 10 minutes

Action: Warning alert

### S3

Metric: 4xxErrors

Threshold: > 100 for 5 minutes

Action: Warning alert

### DynamoDB

Metric: ReadThrottleEvents or WriteThrottleEvents

Threshold: > 0 for 5 minutes

Action: Warning alert

### ElastiCache

Metric: CPUUtilization

Threshold: > 90% for 15 minutes

Action: Warning alert

### RDS

Metric: CPUUtilization

Threshold: > 80% for 15 minutes

Action: Warning alert

Metric: FreeableMemory

Threshold: < 10% for 15 minutes

Action: Critical alert

### SES

Metric: Reputation.BounceRate

Threshold: > 5% for 1 hour

Action: Warning alert

## Escalation Procedures

1. Warning Alerts:
   - Notify on-call engineer via SNS
   - If not acknowledged within 15 minutes, escalate to senior engineer

2. Critical Alerts:
   - Notify on-call engineer and senior engineer via SNS and phone call
   - If not acknowledged within 5 minutes, escalate to engineering manager

3. Incident Response:
   - On-call engineer investigates and attempts to resolve the issue
   - If unable to resolve within 30 minutes, engage additional team members
   - For prolonged outages (> 1 hour), notify stakeholders and consider public status page update

## Performance Optimization Recommendations

1. EC2:
   - Use EC2 Auto Scaling to automatically adjust capacity based on demand
   - Implement application-level caching to reduce database load

2. RDS:
   - Enable and tune Performance Insights for database query optimization
   - Consider using read replicas for read-heavy workloads

3. DynamoDB:
   - Use DAX (DynamoDB Accelerator) for frequently accessed data
   - Implement efficient partition key design to avoid hot partitions

4. ElastiCache:
   - Implement proper cache invalidation strategies
   - Monitor cache hit ratio and adjust cache size as needed

5. CloudFront:
   - Enable caching for static content

- Use Origin Shield to reduce load on your origin

6. S3:
   - Enable transfer acceleration for faster uploads from distant locations
   - Use appropriate storage classes for infrequently accessed data

## Cost Monitoring Setup

1. Enable AWS Cost Explorer:
   - Go to the AWS Billing Console
   - Navigate to Cost Explorer and enable it

2. Set up AWS Budgets:
   - Create a budget for overall monthly spend
   - Set alerts at 50%, 80%, and 100% of the budget

3. Configure Cost Anomaly Detection:
   - Enable AWS Cost Anomaly Detection
   - Set up notifications for detected anomalies

4. Create a Cost & Usage Report:
   - Go to the AWS Cost Management Console
   - Create a detailed Cost & Usage Report delivered to an S3 bucket

5. Add cost widgets to your CloudWatch dashboard:
   - Total monthly spend
   - Spend by service
   - Forecast vs. actual spend

6. Set up custom CloudWatch metrics for cost-related data:
   - EC2 Reserved Instance coverage
   - S3 storage by storage class
   - RDS instance utilization

7. Implement tagging strategy:

   - Enforce tagging policies for all resources

   - Create tag-based views in Cost Explorer

8. Schedule regular cost review meetings:

   - Weekly review of Cost Explorer data

   - Monthly deep dive into Cost & Usage Reports

   - Quarterly optimization planning sessions

By implementing this comprehensive monitoring and alerting setup, you'll have a robust system in place to ensure the health, performance, and cost-effectiveness of your AWS infrastructure. Regular reviews and adjustments of these metrics and thresholds will help maintain an optimized environment as your application evolves.