

AWS Monitoring and Alerting Setup Guide

Table of Contents

1. [CloudWatch Dashboard Configuration](#cloudwatch-dashboard-configuration)
2. [Key Metrics to Monitor](#key-metrics-to-monitor)
3. [Alert Thresholds and Escalation Procedures](#alert-thresholds-and-escalation-procedures)
4. [Performance Optimization Recommendations](#performance-optimization-recommendations)
5. [Cost Monitoring Setup](#cost-monitoring-setup)

CloudWatch Dashboard Configuration

To set up a comprehensive CloudWatch dashboard for your multi-tier web application:

1. Log in to the AWS Management Console and navigate to CloudWatch.
2. Click on "Dashboards" in the left sidebar and then "Create dashboard".
3. Name your dashboard (e.g., "Multi-Tier-Web-App-Dashboard").
4. Add the following widgets to your dashboard:
 - a. EC2 Instances:
 - CPU Utilization
 - Network In/Out
 - Status Check Failed
 - b. ECS:
 - CPU Utilization
 - Memory Utilization
 - Running Task Count
 - c. Application Load Balancer:

- Request Count
- HTTP 4xx/5xx Errors
- Latency

d. RDS:

- CPU Utilization
- Free Storage Space
- Database Connections

e. ElastiCache:

- CPU Utilization
- Evictions
- Cache Hit Rate

f. CloudFront:

- Requests
- Error Rate
- Total Bytes Downloaded

g. S3:

- BucketSizeBytes
- NumberOfObjects
- 4xxErrors and 5xxErrors

h. WAF:

- CountedRequests
- BlockedRequests

5. Arrange the widgets in a logical order for easy monitoring.

Key Metrics to Monitor

Amazon Route 53

HealthCheckStatus

DNSQueries

S3

BucketSizeBytes

NumberOfObjects

4xxErrors and 5xxErrors

AWS WAF

CountedRequests

BlockedRequests

PassedRequests

Amazon CloudFront

Requests

ErrorRate

TotalBytesDownloaded

OriginLatency

Application Load Balancer

RequestCount

HTTPCode_ELB_4XX_Count

HTTPCode_ELB_5XX_Count

TargetResponseTime

ECS

CPUUtilization

MemoryUtilization

RunningTaskCount

EC2

CPUUtilization

NetworkIn/NetworkOut

StatusCheckFailed

Auto Scaling Group

GroupDesiredCapacity

GroupInServiceInstances

GroupPendingInstances

Amazon RDS

CPUUtilization

FreeStorageSpace

DatabaseConnections

ReadIOPS/WriteIOPS

Amazon ElastiCache

CPUUtilization

CacheHits/CacheMisses

Evictions

CurrConnections

SNS

NumberOfMessagesPublished

NumberOfNotificationsDelivered

NumberOfNotificationsFailed

Alert Thresholds and Escalation Procedures

Set up the following CloudWatch Alarms with associated SNS topics for notifications:

1. EC2 CPU Utilization > 80% for 5 minutes
 - Action: Send to SNS topic "EC2-High-CPU"

- Escalation: If persists for 30 minutes, send to "Ops-Team" SNS topic

2. RDS CPU Utilization > 80% for 5 minutes

- Action: Send to SNS topic "RDS-High-CPU"
- Escalation: If persists for 30 minutes, send to "DBA-Team" SNS topic

3. ECS Service CPU Utilization > 80% for 5 minutes

- Action: Send to SNS topic "ECS-High-CPU"
- Escalation: If persists for 30 minutes, trigger Auto Scaling policy

4. Application Load Balancer 5XX errors > 10 per minute

- Action: Send to SNS topic "ALB-High-5XX"
- Escalation: If persists for 15 minutes, send to "Dev-Team" SNS topic

5. ElastiCache Evictions > 100 per minute

- Action: Send to SNS topic "Cache-Evictions"
- Escalation: If persists for 15 minutes, send to "Ops-Team" SNS topic

6. S3 4XX Errors > 100 per minute

- Action: Send to SNS topic "S3-Access-Issues"
- Escalation: If persists for 30 minutes, send to "Security-Team" SNS topic

7. WAF BlockedRequests > 100 per minute

- Action: Send to SNS topic "WAF-Blocked-Requests"
- Escalation: If persists for 15 minutes, send to "Security-Team" SNS topic

Ensure that each SNS topic has appropriate subscribers (email, SMS, or integrated with incident management tools like PagerDuty).

Performance Optimization Recommendations

1. Implement Multi-AZ deployment for RDS to improve high availability.
2. Use AWS Fargate for serverless container management to reduce operational overhead.
3. Implement AWS Shield for enhanced DDoS protection.

4. Utilize AWS Secrets Manager for secure management of database credentials.
5. Implement AWS Config for configuration management and compliance tracking.
6. Consider migrating to Amazon Aurora for improved database performance and scalability.
7. Use ElastiCache to offload read requests from RDS and improve application response times.
8. Implement CloudFront caching strategies to reduce origin requests and improve content delivery speed.
9. Use S3 Transfer Acceleration for faster uploads to your S3 bucket from distant locations.
10. Implement Auto Scaling for EC2 and ECS to handle varying loads efficiently.

Cost Monitoring Setup

1. Enable AWS Cost Explorer:
 - Go to AWS Cost Management console
 - Navigate to Cost Explorer and enable it
2. Set up AWS Budgets:
 - Create a budget for overall monthly spend
 - Set up service-specific budgets for high-cost services (e.g., EC2, RDS)
 - Configure alert thresholds at 50%, 80%, and 100% of budgeted amount
3. Use Cost Allocation Tags:
 - Implement a consistent tagging strategy across all resources
 - Activate relevant tags for cost allocation in the Billing and Cost Management console
4. Set up a Cost and Usage Report:
 - Go to AWS Cost Management console
 - Navigate to Cost & Usage Reports
 - Create a new report that delivers to an S3 bucket
5. Create a CloudWatch Dashboard for Cost Monitoring:
 - Add widgets for:
 - Total monthly spend
 - Spend by service

- Spend by tagged resources

6. Implement AWS Trusted Advisor:

- Enable Trusted Advisor checks
- Set up notifications for cost optimization recommendations

7. Use AWS Compute Optimizer:

- Enable Compute Optimizer
- Regularly review recommendations for EC2 instance right-sizing

8. Set up anomaly detection:

- Use CloudWatch Anomaly Detection on your cost metrics
- Create alarms based on detected anomalies

By following this guide, you'll have a comprehensive monitoring and alerting setup that covers performance, security, and cost aspects of your AWS infrastructure. Regularly review and adjust these configurations as your application evolves and your understanding of its behavior improves.