

# O&M Lab Notes: Monitoring and Logging for Our Cloud App - Master Guide

## **Mental Model: Your Cloud Spaceship**

Imagine your cloud application as a **tiny, important spaceship** 🚀. To keep it running smoothly and fix problems fast, you need two main tools:

**Cloud Eye** 🎮 The ship's **dashboard and control panel**

Real-time monitoring and alerting

**LTS (Log Tank Service)** 📖 The ship's **flight recorder and captain's log**

Historical log collection and analysis

---

# Phase 1: Building Our Spaceship (Application Infrastructure)

## Infrastructure Blueprint

CLOUD APPLICATION LAYERS	
Layer 4: Application	<div>Discuz! Forum Software</div> <ul style="list-style-type: none"><li>• Web interface</li><li>• User authentication</li></ul>
Layer 3: Compute	<div>ECS (Elastic Cloud Server)</div> <ul style="list-style-type: none"><li>• Virtual machine</li><li>• Runs application code</li></ul>
Layer 2: Database	<div>RDS (Relational DB Service)</div> <ul style="list-style-type: none"><li>• Managed MySQL database</li><li>• Stores user data</li></ul>
Layer 1: Networking	<div>VPC + Subnets + Security</div> <ul style="list-style-type: none"><li>• Private network isolation</li><li>• Traffic control</li></ul>

# Step-by-Step Implementation

## 1. VPC & Network Architecture

### What We Built:

```
VPC: vpc-primary
├── Subnet-web (192.168.1.0/24) ← Web Server Network
│   └── ECS Instance
└── Subnet-db (192.168.2.0/24) ← Database Network (NEW!)
    └── RDS Instance
```

### Key Concept: *Network Segmentation*

- **Security Benefit:** Separates web and database traffic
- **Performance Benefit:** Reduces broadcast domains
- **Operational Benefit:** Independent scaling of tiers




## 2. Security Configuration

### Security Group: `sg-db` - The Digital Bouncer

Inbound TCP 3306 ECS Private IP Allow database access

Inbound ALL ALL Deny All Default deny rule

### Why This Matters:

-  **Zero Trust Model:** Only explicit permissions are allowed
-  **Least Privilege:** ECS can only connect on MySQL port
-  **Audit Trail:** Every connection attempt is logged

### Finding the Private IP:

# In ECS Console:

1. Navigate to "Elastic Cloud Server"
2. Click on your instance (ecs-web)
3. Find "Private IP" in "Network Information"

## 3. Database Deployment

### RDS Configuration Summary:

**Instance Name** `rds-web` Clear naming for operations

**Engine** MySQL 5.7 Proven, stable version **Type** Primary/Standby High availability

**Primary AZ** AZ5 Spread risk across zones **Standby AZ** AZ3 Automatic failover

**Security Group** `sg-db` Enforces network policy

**Password** `Huawei@123#$` Strong credentials

**Pro Tip:** Always use different passwords for different services and store them securely!

## 4. Application Installation

**Discuz! Configuration Flow:**

```
http://ECS_Public_IP:80
↓
License Agreement (Click "I Agree")
↓
System Check (Verify permissions)
↓
Installation Type (Choose "New Installation")
↓
Database Configuration:
  • Server: RDS Private IP
  • Password: RDS Root Password
↓
Admin Account Setup
↓
Installation Complete!
```

**Verification:** Access `http://ECS_Public_IP:80` again to see your forum!

---

## Phase 2: Cloud Eye - The Monitoring Dashboard

### A. Metric Monitoring: The Gauges and Dials

#### What Are Metrics?

- **CPU Usage:** How hard your computer is working (0-100%)
- **Memory Usage:** How much RAM is being used
- **Disk I/O:** Reading/writing speed to storage
- **Network Traffic:** Data flowing in/out

## Viewing Metrics in Cloud Eye:

Navigation: Service List → Management & Deployment → Cloud Eye

↓

Select "Elastic Cloud Server" from Resource Type

↓

Choose your ECS instance (ecs-web)

↓

View Dashboard with 6 Default Charts:

1. CPU Usage (%)
2. Memory Usage (MB)
3. Disk Read Rate (KB/s)
4. Disk Write Rate (KB/s)
5. Network Inbound (bit/s)
6. Network Outbound (bit/s)

## Customization Options:

- Add/Remove specific charts

- Change time range (1h, 3h, 12h, 24h, 7d)
- Export data for analysis
- Set comparison periods

## B. Event Monitoring & Alarms: Red Lights & Sirens

### SMN (Simple Message Notification): The Messenger Service

#### Conceptual Model:

Event → Alarm Rule → SMN Topic → Subscriber → Notification

#### Step 1: Create SMN Topic (The Radio Station)

1. Service List → Application Services → Simple Message Notification
2. Topics → Create Topic
3. Name: "test"
4. Display Name: "Test Topic"
5. Create

#### Step 2: Add Subscription (Tuning In)

1. Click on "test" topic
2. Subscriptions → Add Subscription
3. Protocol: Email
4. Endpoint: your\_email@example.com
5. Confirm subscription via email

### Step 3: Create Alarm Rule (The Trigger)

#### Rule Configuration: `alarm-reboot`

Condition: IF (Event Source = ECS) AND (Event Name = "ECS restarted")  
Action: THEN (Send to SMN Topic = "test")  
Severity: Critical  
Enabled: Yes

### Anatomy of an Event:

```
{  
  "event_name": "ECS restarted",  
  "event_source": "ECS",  
  "time": "2023-01-01T12:00:00Z",  
  "resource_id": "ecs-web-id",  
  "resource_name": "ecs-web",  
  "resource_type": "ecs",  
  "severity": "critical"  
}
```

## Step 4: Testing the Alarm

### Manual Test Procedure:

1. ECS Console → Find ecs-web → More → Restart
2. Wait 1-2 minutes for event detection
3. Check your email for alarm notification

### Expected Email Content:

Subject: [CES Alarm][Critical] alarm-reboot

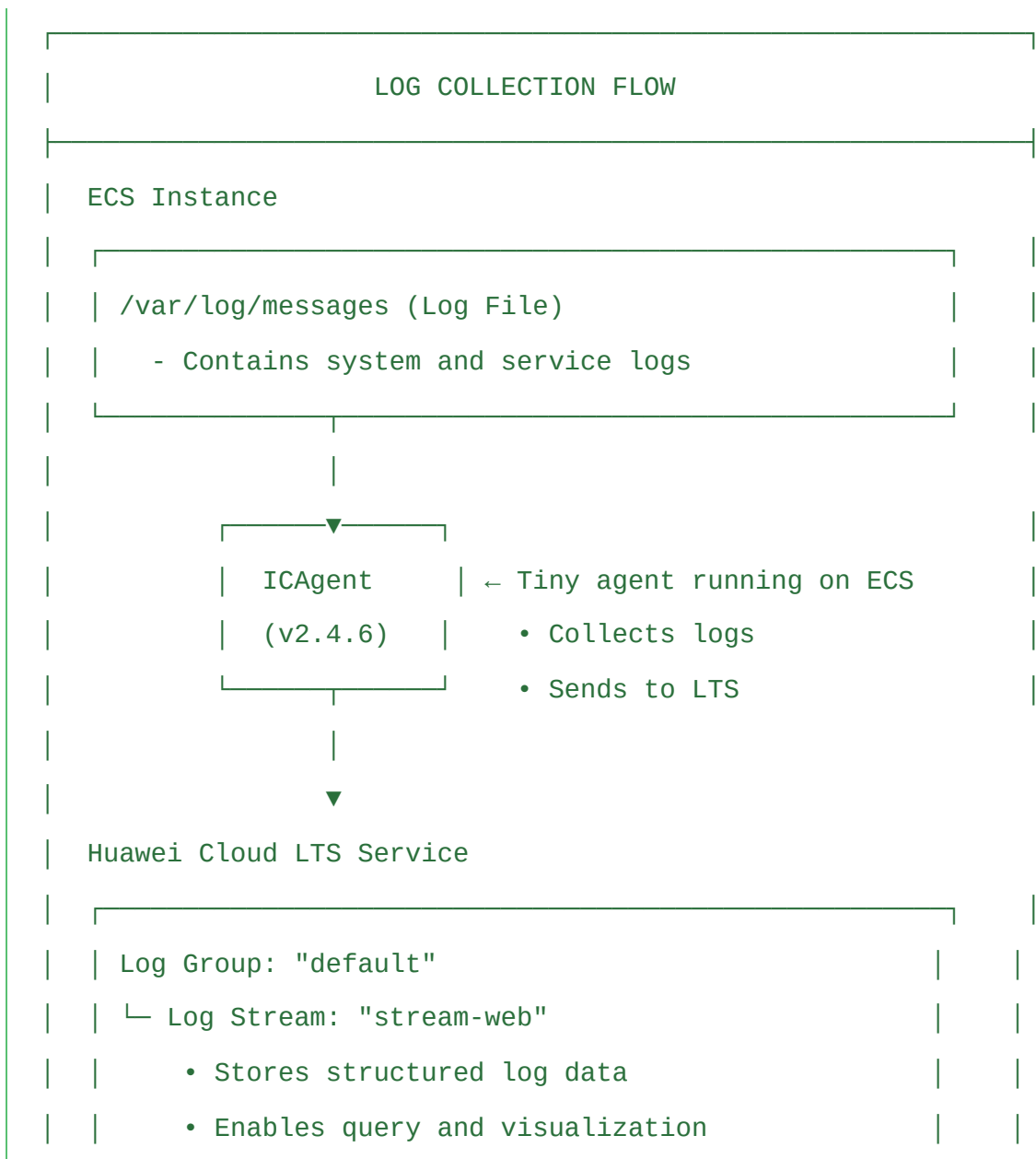
Body:

- Alarm Status: OK → ALARM
  - Resource Name: ecs-web
  - Event: ECS restarted
  - Time: 2023-01-01 12:00:00 GMT+08:00
  - Alarm Policy: alarm-reboot
-



## Phase 3: LTS (Log Tank Service) - The Flight Recorder

### A. Log Collection Architecture



## B. Step-by-Step LTS Configuration

### Step 1: Create Log Structure

Log Group (Container of containers) → Log Stream (Individual container)

#### 1. Create Log Group:

Service List → Management & Deployment → Log Tank Service

↓

Log Groups → Create Log Group

↓

Name: "default" (or custom name)

Retention Period: 7 days (configurable)

Create

#### 2. Create Log Stream:

Click on "default" Log Group

↓

Log Streams → Create Log Stream

↓

Name: "stream-web"

Create

## Step 2: Install and Configure ICAgent

**ICAgent = Intelligent Collection Agent**

### Prerequisites Checklist:

- ☒ ECS is running
- ☒ Log Group/Stream created
- ☒ Have valid AK/SK credentials
- ☒ Network connectivity to LTS

### Installation Process:

1. LTS Console → Log Collection → ICAgent Management
2. Click "Install ICAgent"
3. Select Host Type: Linux
4. Get installation command

## Critical Configuration: AK/SK (Access Key/Secret Key)

### What Are AK/SK?

- **AK (Access Key):** Like a username (public)
- **SK (Secret Key):** Like a password (PRIVATE!)

### Where to Find AK/SK:

Huawei Cloud Console → Top Right (Username) → My Credentials

↓

Access Keys → Create Access Key

↓

Download credentials.csv (ONCE! Never share!)

### Agent Status Troubleshooting:

✅ **Normal** Agent working correctly None needed

⚠️ **Abnormal** Communication issue Check AK/SK, network

❌ **Not Installed** Agent not present Install ICAgent

↻ **Upgrading** Agent updating Wait 2-3 minutes

### Step 3: Configure Host Group & Log Collection

#### Host Group Concept:

- Logical grouping of servers with similar logging needs
- Example: `hccdp` group for all web servers

#### Configuration Steps:

1. LTS → Log Collection → Host Groups
2. Create Host Group: "hccdp"
3. Add ECS to group
4. Configure Log Collection:
  - Log Path: /var/log/messages
  - Log Stream: stream-web
  - Extraction Mode: Full Text

## Intelligent Extraction Explained:

Raw Log: Jan 1 12:00:00 localhost systemd: Stopped httpd.service

↓

Extraction Configuration:

- Separator: Colon (:)
- Fields: [Timestamp], [Host], [Service], [Message]

↓

Structured Log:

```
{
  "timestamp": "Jan 1 12:00:00",
  "host": "localhost",
  "service": "systemd",
  "message": "Stopped httpd.service"
}
```

## C. Log Analysis & Visualization

### Viewing Logs:

LTS Console → Log Query

↓

Select: Log Group = "default", Log Stream = "stream-web"

↓

Time Range: Last 15 minutes (adjustable)

↓

Click "Search"

### Visualization Types:

1. **Table View:** Raw log entries with timestamps
2. **Bar Chart:** Log frequency over time
3. **Pie Chart:** Distribution by service/severity
4. **Line Chart:** Trend analysis

### Sample Query Results:

Timestamp	Host	Service	Message
Jan 1 12:00:00	localhost	systemd	Stopped httpd.service
Jan 1 12:00:01	localhost	crond	Started periodic cmd
Jan 1 12:00:02	localhost	sshd	Accepted password

## D. Log-Based Alarms

### Creating a Log Alarm Rule:

**Goal:** Detect when any service stops unexpectedly

### Configuration:

Rule Name: "log-alarm-stopped"

Log Group: default

Log Stream: stream-web

Query Statement: service = "systemd" AND message like "%Stopped%"

Check Interval: 1 minute

Trigger Condition: Number of logs >= 1

Action: Send to SMN Topic "test"

### Keyword Statistics Concept:

- Count occurrences of specific words/patterns
- Example: Count "Stopped", "Error", "Failed" occurrences
- Trigger alarm when threshold exceeded

## Testing the Log Alarm

### Simulating a Service Stop:

```
# SSH into ECS (using EIP and password)
ssh root@<ECS_Public_IP>

# Stop a service to generate log
systemctl stop httpd
# OR
systemctl Stopped httpd

# Check if logged
tail -f /var/log/messages
```

### Expected Alert Flow:

1. Service stopped → Log entry created
  2. ICAgent collects log (within 1 minute)
  3. LTS processes and analyzes log
  4. Alarm rule triggers (if "Stopped" found)
  5. SMN sends email notification
  6. You receive alert within 2-3 minutes
- 

## Comprehensive Troubleshooting Guide

### Common Issues & Solutions

#### 1. ICAgent Status: "Abnormal"

##### Root Causes:

- Invalid AK/SK credentials
- Network connectivity issues
- Agent process crashed

##### Diagnosis Steps:

```
# Check agent status on ECS
systemctl status icagent

# Check agent logs
tail -f /opt/icagent/icagent.log

# Verify connectivity
telnet lts.ap-southeast-3.myhuaweicloud.com 443
```

**Solution:** Reinstall ICAgent with correct AK/SK

## 2. No Logs Appearing in LTS

### Checklist:

- ☒ ICAgent status = Normal
- ☒ Log path configured correctly
- ☒ Host added to host group
- ☒ Log file exists and has permissions
- ☒ Time range in query is correct

### Debug Command:

```
# Check if logs are being generated
ls -la /var/log/messages

# Check file permissions (should be readable)
cat /var/log/messages | head -5

# Test ICAgent collection manually
/opt/icagent/bin/icagent -t
```

### 3. No Alarm Notifications

#### Investigation Path:

1. Check Alarm Rule Status (Enabled/Disabled)
2. Verify SMN Topic exists
3. Confirm email subscription is "Confirmed"
4. Check spam/junk folder
5. Verify event actually occurred
6. Check Cloud Eye event logs

### 4. Performance Issues

#### Monitoring Metrics to Watch:

- **ICAgent CPU/Memory:** Should be < 10%
- **Log Volume:** Adjust collection if too high

- **Network Traffic:** Monitor LTS egress costs
-



# Key Terminology Reference

**Cloud Eye** Monitoring service that watches cloud resources

Car dashboard showing speed, fuel, engine lights

**LTS** Service that collects and analyzes log files Airplane black box recorder

**SMN** Notification service that sends alerts Emergency broadcast system

**Topic** Channel for sending messages Radio station frequency

**Subscription** Endpoint receiving topic messages Radio tuned to a station

**ICAgent** Software that collects logs from servers

Security camera recording footage

**AK/SK** Credentials for API access Username and password for bank account

**Host Group** Logical group of servers Department in a company

**Log Stream** Container for related log data Chapter in a book

**Metric** Numerical measurement of resource Speedometer reading

**Event** Discrete occurrence in the system Check engine light turning on

---

## **Best Practices Summary**

### **Monitoring Best Practices:**

#### **1. Set Meaningful Thresholds:**

- CPU: Warning at 70%, Critical at 90%
- Memory: Warning at 80%, Critical at 95%
- Disk: Warning at 85%, Critical at 95%

#### **2. Create Escalation Policies:**

- Level 1: Email to team
- Level 2: SMS to on-call (repeated alerts)
- Level 3: Phone call (critical, unacknowledged)

#### **3. Regular Review:**

- Weekly: Review false positives
- Monthly: Tune thresholds
- Quarterly: Test alerting chain

# Logging Best Practices:

## 1. Log Structure:

- Use consistent timestamp format
- Include severity levels (INFO, WARN, ERROR)
- Add correlation IDs for tracing

## 2. Retention Strategy:

- 7 days: Hot storage (frequent queries)
- 30 days: Warm storage (occasional queries)
- 1 year: Cold storage (compliance/audit)

## 3. Security Considerations:

- Never log passwords or secrets
  - Mask PII (Personally Identifiable Information)
  - Encrypt sensitive log data
-



# Advanced Scenarios for Future Learning

## 1. Custom Metrics

```
# Push custom application metrics to Cloud Eye
curl -X POST "https://ces.myhuaweicloud.com/V1.0/..."
-d '{"metric_name":"user_logins","value":42}'
```

## 2. Log Correlation

- Combine Cloud Eye metrics with LTS logs
- Example: High CPU + “Out of memory” logs = Memory leak
- Use time synchronization for accurate correlation

## 3. Automated Remediation

Alarm Triggered → Runbook Automation → Auto-fix

Example:

1. Alarm: Disk > 90%
2. Action: Run cleanup script
3. Verify: Disk < 70%
4. Notify: Cleanup complete

## 4. Distributed Tracing

User Request → Load Balancer → Web Server → Database

↓

↓

↓

↓

Trace ID: abc123   abc123        abc123        abc123

Correlate across all logs and metrics!

## Quick Reference Commands

### ECS Monitoring Commands:

# Check CPU usage

```
top -b -n 1 | grep "Cpu(s)"
```

# Check memory

```
free -h
```

# Check disk

```
df -h
```

# Check running processes

```
ps aux --sort=-%cpu | head -10
```

## Log Investigation Commands:

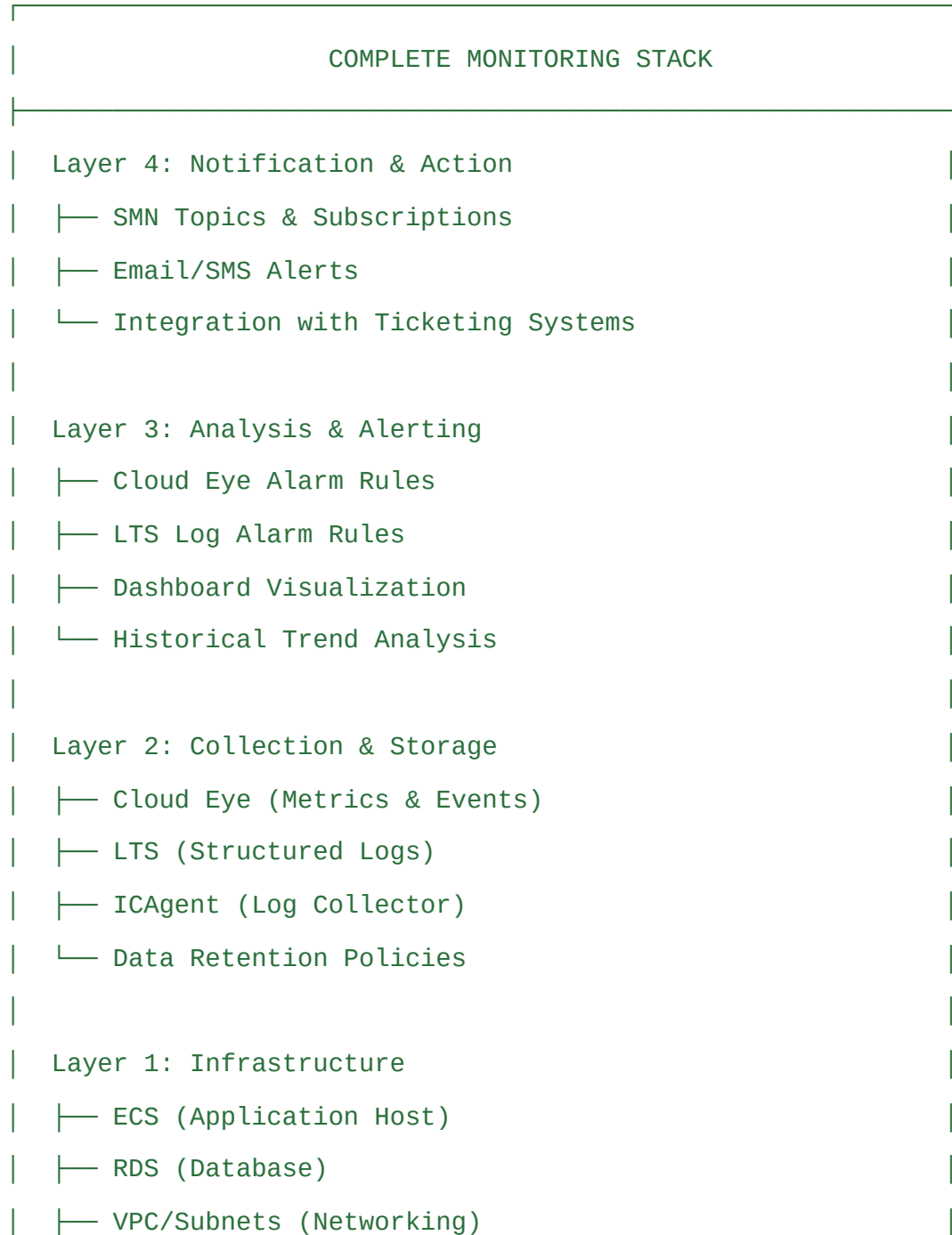
```
# Tail logs in real-time  
tail -f /var/log/messages  
  
# Search for errors  
grep -i "error\|failed\|stopped" /var/log/messages  
  
# Count occurrences  
grep -c "Stopped" /var/log/messages  
  
# View last 50 lines with timestamps  
tail -50 /var/log/messages | grep -E "^[A-Z][a-z]{2}"
```

## Service Management:

```
# Check service status  
systemctl status httpd  
  
# Start service  
systemctl start httpd  
  
# Stop service (for testing)  
systemctl stop httpd  
  
# Restart service  
systemctl restart httpd
```

## **Final Architecture Review**

Your complete monitoring stack:



| — Security Groups (Access Control) |

---



## Success Metrics for Your O&M Setup

**MTTD** (Mean Time to Detect) < 5 minutes Time from issue to alert

**MTTR** (Mean Time to Resolve) < 30 minutes Time from alert to fix




**Alert Accuracy** > 95% (Valid alerts)/(Total alerts)




**Log Coverage** 100% of critical systems Systems with ICAgent

**Notification Success** > 99% Alerts delivered successfully

---

**Congratulations!** 🎉 You've now mastered the complete O&M lifecycle:

-  **Infrastructure Setup** (Building the ship)
-  **Metric Monitoring** (Watching the gauges)
-  **Event Alerting** (Setting up alarms)

-  **Log Collection** (Installing the flight recorder)
-  **Log Analysis** (Reading the logs)
-  **Proactive Monitoring** (Anticipating issues)

Remember: **Good monitoring tells you what's happening. Great monitoring tells you what's about to happen.** You're now equipped to build the latter!

*"May your alerts be few, your uptime be high, and your coffee be strong!"* 