SRE Dashboards with AWS EC2 & CloudWatch

4-Hour Hands-On Workshop (Updated with IAM Permissions)

Workshop Overview

Duration: 4 hours

Level: Intermediate

Prerequisites: AWS account with EC2 and CloudWatch access

Learning Objectives:

- Configure CloudWatch monitoring for EC2 instances
- Create SRE dashboards using CloudWatch console
- Set up alarms and notifications
- Implement SRE metrics (SLIs, SLOs)

Part 1: SRE Concepts Overview

Key Metrics to Monitor:

- **Latency:** Response time of requests
- **Traffic:** Number of requests per second
- **Errors:** Failed request rate
- **Saturation:** CPU, Memory, Disk usage

SRE Targets:

- **SLI:** Availability percentage (e.g., 99.9%)
- **SLO:** Target uptime (e.g., 99.9% = 43 minutes downtime/month)
- **Error Budget:** Remaining allowable downtime

Part 2: Create IAM Role for CloudWatch

⚠️ CRITICAL: Do this BEFORE launching EC2 instances!

Step 1: Navigate to IAM Console

1. Open AWS Console

2. Search for "IAM" in top search bar
3. Click IAM service

Step 2: Create IAM Role

1. Click **Roles** in left navigation menu
2. Click **Create role** button
3. Select trusted entity type: **AWS service**
4. Use case: Select **EC2**
5. Click **Next**

Step 3: Attach Policies

1. In the search box, type: `CloudWatchAgentServerPolicy`
2. Check the box next to **CloudWatchAgentServerPolicy**
3. (Optional) Also add: `AmazonSSMManagedInstanceCore` for better management
4. Click **Next**

Step 4: Name and Create Role

1. Role name: `SRE-Workshop-CloudWatch-Role`
2. Description: `Allows EC2 instances to send metrics to CloudWatch`
3. Click **Create role**

✅ **You now have an IAM role ready to attach to EC2 instances!**

Part 3: Launch EC2 Instances

Step 1: Navigate to EC2 Console

1. Open AWS Console
2. Search for "EC2" in top search bar
3. Click EC2 service

Step 2: Launch Instance

1. Click **Launch Instance** button (orange)
2. Name: `SRE-Workshop-Web-01`
3. Select **Amazon Linux 2023 AMI**
4. Instance type: `t3.micro`
5. Key pair: Select existing or create new

6. Network settings: Allow SSH (port 22) and HTTP (port 80)
7. ⚠️ **IMPORTANT:** Expand **Advanced details** section
8. **IAM instance profile:** Select `SRE-Workshop-CloudWatch-Role`
9. Click **Launch Instance**
10. Repeat for second instance: `SRE-Workshop-Web-02`

✅ **Both instances now have permission to send metrics to CloudWatch!**

Step 3: Add Tags

1. Select instance
2. Click **Tags** tab
3. Click **Manage tags**
4. Add tags:
   - Key: `Environment`, Value: `Workshop`
   - Key: `Application`, Value: `WebServer`
   - Key: `Team`, Value: `SRE`
5. Click **Save**

Step 4: Connect to Instance

1. Select instance
2. Click **Connect** button
3. Choose **EC2 Instance Connect** tab
4. Click **Connect** button
5. Browser terminal opens

Step 5: Install CloudWatch Agent

Execute these commands in terminal:

```bash
sudo dnf update -y
sudo dnf install amazon-cloudwatch-agent -y
```

Part 4: Create CloudWatch Agent Configuration

Step 1: Create Configuration File

```
bash

sudo nano /opt/aws/amazon-cloudwatch-agent/etc/config.json
```

Step 2: Paste Configuration

Copy and paste this configuration:

```
json

{
  "agent": {
    "metrics_collection_interval": 60
  },
  "metrics": {
    "namespace": "SREWorkshop",
    "metrics_collected": {
      "cpu": {
        "measurement": [
          "cpu_usage_idle",
          "cpu_usage_iowait"
        ],
        "totalcpu": false
      },
      "disk": {
        "measurement": [
          "used_percent"
        ],
        "resources": [
          "*"
        ]
      },
      "mem": {
        "measurement": [
          "mem_used_percent"
        ]
      }
    }
  }
}
```

Press Ctrl+X, then Y, then Enter to save

Step 3: Start CloudWatch Agent

```
bash
```

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c file:/opt/aws/amazon-
```

Expected output: `Configuration validation succeeded`

Step 4: Verify Agent Status

```bash
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a status -c default
```

Expected output:

```json
{
  "status": "running",
  "starttime": "2025-10-05T11:34:53+00:00",
  "configstatus": "configured",
  "version": "1.300057.2"
}
```

Also verify with systemd:

```bash
systemctl status amazon-cloudwatch-agent
```

Expected output: Should show `active (running)` in green

Step 5: Wait for Metrics to Appear

⏱️ **IMPORTANT:** Wait 3-5 minutes for metrics to appear in CloudWatch!

- The agent collects metrics every 60 seconds
- It takes 2-3 collection cycles for data to appear in CloudWatch console
- The `SREWorkshop` namespace won't be visible until the first batch of metrics is sent

---

Part 5: Install Sample Application

Step 1: Install Web Server

```bash
```

```
sudo dnf install nginx -y
sudo systemctl start nginx
sudo systemctl enable nginx
```

Step 2: Verify Installation

```bash
curl localhost
```

Step 3: Generate Some Traffic

```bash
for i in {1..100}; do curl localhost; done
```

Step 4: Check Nginx Status

```bash
sudo systemctl status nginx
```

Step 5: View Nginx Logs

```bash
sudo tail -f /var/log/nginx/access.log
```

Press Ctrl+C to stop viewing logs

Part 6: Navigate to CloudWatch

1. Open AWS Console (new browser tab)
2. Search for "CloudWatch" in top search bar
3. Click CloudWatch service
4. Click **Dashboards** in left navigation menu

Part 7: Create New Dashboard

Step 1: Create Dashboard

1. Click **Create dashboard** button

2. Dashboard name: `SRE-Production-Dashboard`
3. Click **Create dashboard**

Step 2: Add Widget - CPU Utilization

1. Select **Line** widget type
2. Click **Next**
3. Click **Metrics** tab
4. Click **EC2 → Per-Instance Metrics**
5. Search for: `CPUUtilization`
6. Check boxes for both instances
7. Click **Create widget** button

Step 3: Add Widget - Memory Usage

⚠️ **IMPORTANT:** If `SREWorkshop` namespace is not visible, wait 2-3 more minutes and refresh the page!

1. Click **Add widget** (plus icon)
2. Select **Line** widget
3. Click **Next**
4. Click **Browse** tab
5. Select **SREWorkshop** namespace (under Custom namespaces)
6. Click **Metrics with no dimensions**
7. Select `mem_used_percent`
8. Click **Options** tab
9. Title: `Memory Utilization %`
10. Left Y-axis: Min `0`, Max `100`
11. Click **Create widget**

Step 4: Add Widget - Disk Usage

1. Click **Add widget**
2. Select **Number** widget type
3. Click **Next**
4. Click **Browse** tab
5. Navigate: **SREWorkshop → Metrics with no dimensions**
6. Select `disk_used_percent`
7. Click **Options** tab

8. Title: `Disk Usage %`
9. Click **Create widget**

Step 5: Add Widget - Network Traffic

1. Click **Add widget**
2. Select **Line** widget
3. Click **Next**
4. Click **Browse → EC2 → Per-Instance Metrics**
5. Search: `NetworkIn`
6. Select both instances
7. Search: `NetworkOut`
8. Select both instances
9. Click **Options** tab
10. Title: `Network Traffic (Bytes)`
11. Click **Create widget**

Step 6: Add Widget - Status Check

1. Click **Add widget**
2. Select **Number** widget
3. Click **Next**
4. Click **Browse → EC2 → Per-Instance Metrics**
5. Search: `StatusCheckFailed`
6. Select both instances
7. Click **Options** tab
8. Title: `Status Check Failures`
9. Click **Create widget**

Step 7: Arrange Dashboard

1. Drag widgets to arrange layout
2. Resize widgets by dragging corners
3. Suggested layout:
   - **Top row:** CPU and Memory (side by side)
   - **Middle row:** Network Traffic (full width)
   - **Bottom row:** Disk Usage and Status Check (side by side)

Step 8: Save Dashboard

1. Click **Save dashboard** button (top right)
2. Confirmation message appears

Part 8: Configure Time Range & Auto-Refresh

1. Click time range dropdown (top right)
2. Select **1h** (1 hour)
3. Click **Auto refresh** dropdown
4. Select **1m** (1 minute auto-refresh)

Part 9: Create CloudWatch Alarms

Step 1: Navigate to Alarms

1. In CloudWatch console
2. Click **Alarms** in left menu
3. Click **All alarms**
4. Click **Create alarm** button

Step 2: Create CPU Alarm

1. Click **Select metric**
2. Navigate: **EC2 → Per-Instance Metrics**
3. Search: `CPUUtilization`
4. Select one instance
5. Click **Select metric**

Step 3: Configure Alarm Conditions

1. Statistic: `Average`
2. Period: `5 minutes`
3. Threshold type: `Static`
4. Condition: `Greater than 80`
5. Click **Next**

Step 4: Configure Notifications

1. Select: **Create new topic**
2. Topic name: `SRE-Alerts`
3. Email: Enter your email address
4. Click **Create topic**
5. Click **Next**

Step 5: Name Alarm

1. Alarm name: `High-CPU-Web-01`
2. Description: `CPU exceeds 80% threshold`
3. Click **Next**
4. Review settings
5. Click **Create alarm**

Step 6: Confirm Email Subscription

1. Check your email
2. Click confirmation link in AWS notification email
3. Return to CloudWatch console

Step 7: Create Additional Alarms

Repeat steps 1-5 for:

- **High Memory Usage:** `mem_used_percent > 85`
- **Disk Space Low:** `disk_used_percent > 90`
- **Status Check Failed:** `StatusCheckFailed > 0`

Part 10: Add Alarms to Dashboard

Step 1: Open Dashboard

1. Click **Dashboards** in left menu
2. Click **SRE-Production-Dashboard**

Step 2: Add Alarm Widget

1. Click **Add widget**
2. Select **Alarm status** widget
3. Click **Next**
4. Click **Configure**

5. Select all alarms created
6. Click **Create widget**

Step 3: Position Widget

1. Drag alarm widget to top of dashboard
2. Resize to full width
3. Click **Save dashboard**

Part 11: Create SLO Tracking

Step 1: Calculate Availability SLO

1. Navigate to **Metrics → All metrics**
2. Click **Browse** tab
3. Select **EC2 → Per-Instance Metrics**
4. Select: `StatusCheckFailed`

Step 2: Create Math Expression

1. Click **Add math → Start with empty expression**
2. Enter formula: `100 - (m1 * 100)`
3. Label: `Availability %`
4. ID: `e1`

Step 3: Add to Dashboard

1. Click **Actions → Add to dashboard**
2. Select: `SRE-Production-Dashboard`
3. Widget type: **Number**
4. Click **Add to dashboard**
5. Click **Save dashboard**

Part 12: Test & Validate

Step 1: Generate Load

Return to EC2 instance terminal:

```bash
```

```
sudo yum install stress -y
stress --cpu 2 --timeout 300s
```

Step 2: Monitor Dashboard

1. Return to CloudWatch dashboard
2. Watch metrics update in real-time
3. Observe CPU spike
4. Wait for alarm notification email

Step 3: Stop Load Test

In terminal press: `Ctrl+C`

Troubleshooting Guide

Issue: SREWorkshop Namespace Not Appearing

Symptoms: Custom metrics don't show up in CloudWatch after 5+ minutes

Solutions:

1. **Check IAM Role:**

```bash
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/
```

- If empty, the instance doesn't have an IAM role attached
- Go to EC2 Console → Select instance → Actions → Security → Modify IAM role
- Attach `SRE-Workshop-CloudWatch-Role`
- Restart CloudWatch agent

2. **Verify Agent Status:**

```bash
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a status -c default
```

- Should show `"status": "running"`

3. **Check Agent Logs:**

```bash
```

```
sudo tail -f /opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log
```

- Look for permission errors or configuration issues

4. **Restart Agent:**

```bash
sudo systemctl restart amazon-cloudwatch-agent
```

Workshop Summary & Best Practices

Key Takeaways

Dashboard Organization:

- Group related metrics together
- Use consistent time ranges
- Enable auto-refresh for production monitoring
- Color-code critical vs warning states

Alarm Strategy:

- Set thresholds based on historical data
- Use multiple evaluation periods to avoid false positives
- Create escalation paths (warning → critical)
- Test alarms regularly

SRE Metrics Priority:

1. **Availability** (uptime)
2. **Latency** (response time)
3. **Error rate** (failed requests)
4. **Saturation** (resource usage)

Next Steps

1. Add Application Metrics:

- Custom application logs
- Business metrics

- User experience metrics

2. Create Multiple Dashboards:

- Executive summary dashboard
- Detailed technical dashboard
- Team-specific dashboards

3. Implement Automation:

- Auto-scaling based on metrics
- Automated remediation
- Incident response workflows

4. Set Up Log Analysis:

- CloudWatch Logs Insights
- Log aggregation
- Error pattern detection

Cleanup Instructions

1. Delete Alarms:

CloudWatch → Alarms → Select all → Actions → Delete

2. Delete Dashboard:

CloudWatch → Dashboards → Select → Delete

3. Terminate EC2 Instances:

EC2 → Instances → Select → Instance state → Terminate

4. Delete SNS Topics:

SNS → Topics → Select → Delete

5. Delete IAM Role:

IAM → Roles → Select `SRE-Workshop-CloudWatch-Role` → Delete

Additional Resources

AWS Documentation:

- CloudWatch User Guide: https://docs.aws.amazon.com/cloudwatch/
- EC2 Monitoring Guide: https://docs.aws.amazon.com/ec2/monitoring.html
- IAM Roles for EC2: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

SRE Resources:

- Google SRE Book: https://sre.google/books/
- SLO Workshop: https://sre.google/workbook/slo-document/

End of Workshop