RUNBOOK

Monitor Tomcat & Nginx on EC2 using CloudWatch Agent

OS: Amazon Linux 2023

Goal:

- Detect if **Tomcat** or **Nginx** stops
- Show status on a **CloudWatch Dashboard**
- Send **email alert** when a service is down

CORE CONCEPT (read once)

- EC2 **CPU/Network** metrics are automatic
- **Process/service status is NOT**
- We use **CloudWatch Agent + procstat**
- Metric logic:
  - `pid_count >= 1` → service running ✅
  - `pid_count = 0` → service stopped ❌ → ALARM

STEP 1 Create IAM Role (CRITICAL)

CloudWatch Agent **must** authenticate via **EC2 Metadata (IMDS)**.

Without this → nothing works.

Create role

```
IAM → Roles → Create role
```

Trusted entity

- AWS service
- EC2

Permissions

Attach:

`CloudWatchAgentServerPolicy`

Role name

`EC2-CloudWatch-Agent-Role`

STEP 2 Attach Role to EC2 Instance

```
EC2 → Instances → select instance
Actions → Security → Modify IAM role
```

Attach:

`EC2-CloudWatch-Agent-Role`

✅ Save

🔥 TROUBLESHOOT #1 (we hit this)

Symptom

```
NoCredentialProviders
EC2MetadataError: 404
```

Cause

- No IAM role attached to EC2

Fix

- Attach role

- Restart agent

STEP 3 Install CloudWatch Agent (Amazon Linux 2023)

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

STEP 4 Install & Start Services

Tomcat

```
sudo systemctl start tomcat
sudo systemctl enable tomcat
```

Nginx

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

Verify:

```
ps -ef | grep java | grep -v grep
ps -ef | grep nginx | grep -v grep
```

STEP 5 Create CloudWatch Agent Config

(Tomcat + Nginx together — ONE FILE)

📄 File:

```
/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json
```

☑ CORRECT CONFIG (this matters)

```
{
  "metrics": {
    "append_dimensions": {
      "InstanceId": "${aws:InstanceId}"
    },
    "metrics_collected": {
      "procstat": [
        {
          "pattern": "java",
          "measurement": ["pid_count"]
        },
        {
          "pattern": "nginx",
          "measurement": ["pid_count"]
        }
      ]
    }
  }
}
```

🔥 TROUBLESHOOT #2 (we hit this)

❌ WRONG

```
"measurement": ["lookup_pid_count"]
```

Error

```
measurement name lookup_pid_count is invalid
```

Explanation

- `pid_count` → **valid config keyword**

- `procstat_lookup_pid_count` → **CloudWatch UI metric name**
- Config schema ≠ UI metric name

STEP 6 LOAD CONFIG PROPERLY (IMPORTANT)

❌ **Do NOT rely on `systemctl restart` alone**

Always reload config like this:

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

Expected:

- No validation errors
- Agent starts cleanly

STEP 7 Verify Agent Health

```
sudo systemctl status amazon-cloudwatch-agent
```

Check logs:

```
sudo tail -20 /opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log
```

Expected:

```
Everything is ready. Begin running and processing data.
```

🔥 TROUBLESHOOT #3 (we hit this)

Symptom

- Config updated
- Only Tomcat metrics visible

Cause

- Config failed validation earlier
- Agent kept running **old config**

Fix

- Correct config
- Reload using `fetch-config`

STEP 8 Find the CORRECT Metric in CloudWatch

⚠️ This is where confusion happens.

❌ Ignore this (old/legacy)

```
procstat_lookup_pid_count
```

☑ Use this (correct)

```
procstat_pid_count
```

Navigate:

```
CloudWatch → Metrics → All metrics
CWAgent → procstat → procstat_pid_count
```

Now look at **Dimensions**, NOT the summary list.

You will see:

- `pattern = java` → Tomcat
- `pattern = nginx` → Nginx

🔥 TROUBLESHOOT #4 (we hit this)

“It still shows only tomcat”

Explanation

- CloudWatch **never deletes old metrics**
- Old `pattern=tomcat` will live forever
- New services appear under **new dimensions**

This is **expected behavior**.

STEP 9️⃣ Create Alarms

Tomcat Alarm

- Metric: `procstat_pid_count`
- Dimension:

```
pattern = java
```

- Condition:

```
Minimum < 1
```

- Period: `1 minute`
- Action: SNS Email

Name:

```
Tomcat-Service-Down
```

Nginx Alarm

- Metric: `procstat_pid_count`
- Dimension:

```
pattern = nginx
```

- Condition:

```
Minimum < 1
```

Name:

```
Nginx-Service-Down
```

STEP 🔟 Create Dashboard

```
CloudWatch → Dashboards → Create dashboard
```

Name:

```
EC2-Service-Monitoring
```

Widget 1 – Tomcat

- Metric: `procstat_pid_count`
- Filter: `pattern = java`
- Title:

```
Tomcat Status (0 = DOWN, >0 = UP)
```

Widget 2 – Nginx

- Metric: `procstat_pid_count`
- Filter: `pattern = nginx`
- Title:

```
Nginx Status (0 = DOWN, >0 = UP)
```

(Optional: add CPUUtilization)

STEP 1️⃣ Test End-to-End (MANDATORY)

```
sudo systemctl stop nginx
sudo systemctl stop tomcat
```

Within 1–2 minutes:

- Metrics → `0`
- Alarms → **ALARM**
- 📧 Email received

Restart:

```
sudo systemctl start nginx
sudo systemctl start tomcat
```

Alarms return to **OK**.

⌨️ FINAL STATE (What You Achieved)

✓ OS-level service monitoring

✓ One CloudWatch Agent

✓ One config file

✓ Multiple services tracked

✓ Dashboard visibility

✓ Email alerts on failure

✓ All real-world failure modes understood

⭐ INTERVIEW-READY TAKEAWAYS

- CloudWatch does **not** monitor processes by default
- IAM role is **mandatory**
- `pid_count` is the only correct procstat measurement
- Old metrics are **never deleted**
- UI metric names ≠ config keywords
- Always reload config using `fetch-config`