



## How do you collect custom metrics from an EC2 instance using CloudWatch?

By default, EC2 automatically sends **basic system-level metrics** to CloudWatch — things like:

- CPU utilization
- Network in/out
- Disk read/write

These are collected every **5 minutes by default** (can be **1-minute** with detailed monitoring).

But if I want more **custom metrics** like:

- Memory usage
- Disk space used
- App-specific metrics (e.g., request count)

Then I do the following:

- Install the **CloudWatch Agent on the EC2 instance**.
- Configure a cloudwatch-agent.json file specifying what to collect.
- Start the agent, and it begins sending those metrics to CloudWatch.

Real-world example:

I once set this up for a web server running on EC2 where the customer wanted alerts for memory crossing 85%. We used the CloudWatch agent and created an alarm based on that.

### 1. Basic metrics (already there by default)

- From the moment your EC2 starts, CloudWatch collects:
  - CPU utilization (%)
  - Network In/Out (bytes)
  - Disk Read/Write (bytes)
- These arrive in CloudWatch every **5 minutes** (or **1 minute** with detailed monitoring).

### 2. Custom metric requirement

- Problem: Memory usage is **not collected by default**.
- Solution: Install the **CloudWatch Agent**.

### 3. Install & Configure CloudWatch Agent

- SSH into the EC2 and run:

```
bash
```

```
-----
```

```
sudo yum install amazon-cloudwatch-agent
```

- Create a **cloudwatch-agent.json** file, for example:

```
json
```

```
-----
```

```
{
  "metrics": {
    "metrics_collected": {
      "mem": {
        "measurement": [
          "mem_used_percent"
        ],
        "metrics_collection_interval": 60
      }
    }
  }
}
```

This tells CloudWatch Agent to send memory usage % every **1 minute**.

#### 4. Start the CloudWatch Agent

```
bash
```

```
-----
```

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

#### 5. Create a CloudWatch Alarm

- In the AWS console, go to **CloudWatch → Alarms → Create alarm**.
- Select the `mem_used_percent` metric for your EC2.
- Set the threshold: **GreaterThanThreshold 85%** for 2 consecutive minutes.
- Action: Send notification to an **SNS topic** (email/SMS) when triggered.

---

#### Final Result:

The EC2 now sends memory usage data to CloudWatch.

If memory crosses **85%**, CloudWatch triggers the alarm → SNS sends you an email → you can take action before the server crashes.

#### 🔗 Example in Python — tracking “ActiveUsers”:

```
python
```

```
CopyEdit
```

```
import boto3
```

```
import datetime
```

```
cloudwatch = boto3.client('cloudwatch')
```

```
def publish_active_users(count):
```

```
    cloudwatch.put_metric_data(
```

```
        Namespace='MyAppMetrics', # Your custom namespace
```

```

MetricData=[
  {
    'MetricName': 'ActiveUsers',
    'Timestamp': datetime.datetime.utcnow(),
    'Value': count,
    'Unit': 'Count'
  },
]
)

```

# Example usage

active\_users = 57 # You got this from your app logic

publish\_active\_users(active\_users)

❓

### 1. Metrics CloudWatch Already Knows How to Collect

- These are **default metrics** AWS services automatically send to CloudWatch without you doing anything.
- For EC2, that's **CPU utilization, disk read/write, network in/out**.
- They **don't** need the CloudWatch Agent.
- Frequency:
  - **5 min** (basic monitoring)
  - **1 min** (detailed monitoring)

---

### 2. Metrics the OS Knows but CloudWatch Doesn't Collect by Default

*(Still system-level, but not part of EC2's default set)*

- Examples: **Memory usage, disk space used, swap usage, processes count**.
- These require **CloudWatch Agent** because AWS can't get them from the hypervisor — they live inside the OS.
- These are **custom metrics** in AWS terminology (even though they're still system metrics).

---

### 3. Metrics Only the Application Knows

- Examples: **Active users, request count, cache hit ratio, error rates**.
- The application itself must publish these.
- Two ways to do it:
  1. **CloudWatch Agent + StatsD/collectd** integration
    - Agent reads from your app and pushes metrics.
  2. **CloudWatch PutMetricData API** from your app code.
    - No agent needed, but you write code to push metrics.
- These are also **custom metrics** in AWS terminology.

---

**What is the default metric collection frequency in CloudWatch?**

Answer:

By default, CloudWatch collects metrics from AWS services like EC2 every 5 minutes. But if I enable detailed monitoring on EC2, then metrics are collected every 1 minute.


👉 Example:

For most critical production instances, I enable 1-minute monitoring so we can detect spikes faster — like sudden CPU usage surge that would be missed in a 5-minute average.

---

**Which EC2 metrics are collected by CloudWatch by default?**


- A. Memory usage and disk space
- B. CPU utilization, network in/out, disk read/write
- C. Request count and error rate
- D. Application logs

B. CPU utilization, network in/out, disk read/write 

---

**How often does CloudWatch collect EC2 metrics by default?**

- A. Every 1 minute
- B. Every 3 minutes
- C. Every 5 minutes
- D. Every 10 minutes

C. Every 5 minutes 

---

**What is required to collect custom EC2 metrics like memory usage?**

- A. Enable Detailed Monitoring
- B. Install CloudWatch Agent
- C. Enable VPC Flow Logs
- D. Create IAM Role

B. Install CloudWatch Agent 

---

**What is the frequency of EC2 metrics with detailed monitoring enabled?**

- A. 30 seconds
- B. 1 minute
- C. 3 minutes
- D. 5 minutes

B. 1 minute 

---