

Guided Lab: Installing and configuring CloudWatch Agent

Description

CloudWatch Agent is a software package that you can install on your EC2 instances or on-premises servers for monitoring. It collects detailed system-level metrics such as CPU, memory, disk, and network usage. The agent also lets us publish custom metrics to monitor specific application needs, like tracking unique business or operational Key Performance Indicators (KPIs).

Prerequisites

This lab assumes you have experience creating EC2 instances and are familiar with its basic components.

If you find any gaps in your knowledge, consider taking the following labs:

- Creating an Amazon EC2 instance (Linux)
- Setting up a Web server on an EC2 instance

Objectives

In this lab, we'll set up an EC2 instance to demonstrate how to install and configure CloudWatch Agent.

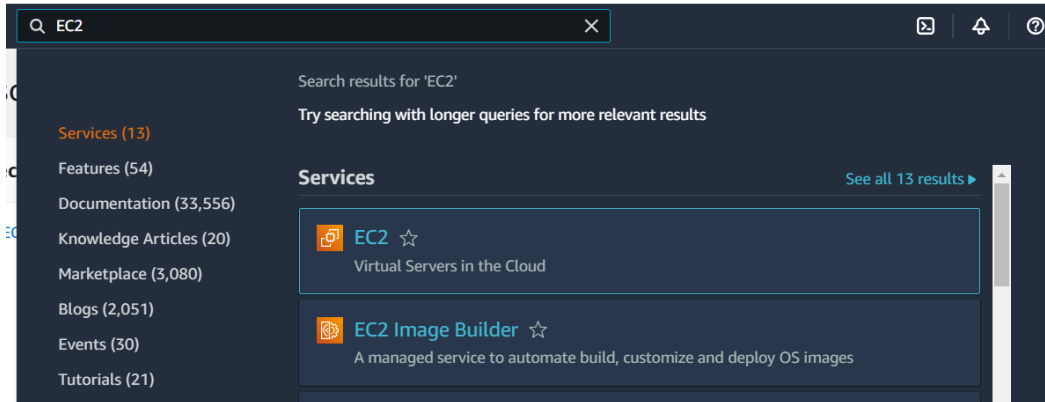
Subscribe to access AWS

PlayCloud Labs

Lab Steps

Connecting to your EC2 Instance via EC2 Instance Connect

1. Search “**ec2**” in the AWS Management Console search bar. Click **EC2** on the search results.



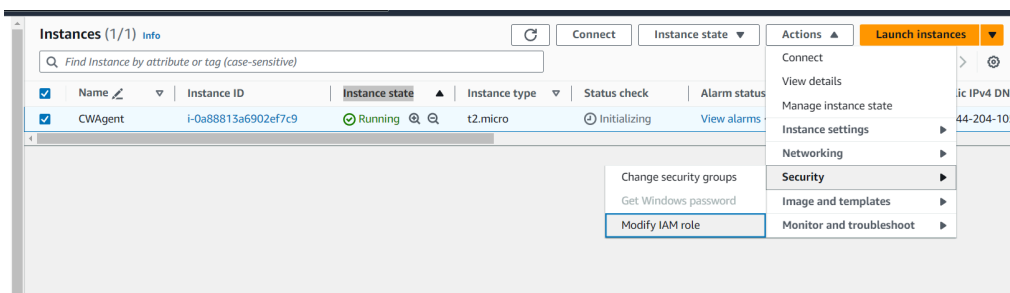
2. Launch EC2 instance with the following configurations.

- Name: CWAgent
- AMI: Ubuntu
- Instance type: t2.micro
- Key pair: If you don't already have a key pair, please create a new one.
 - Key pair name: CWAgent-ssh
 - Key pair type: RSA
 - Private key file format: pem
- Network settings:
 - Allow SSH traffic from: My IP

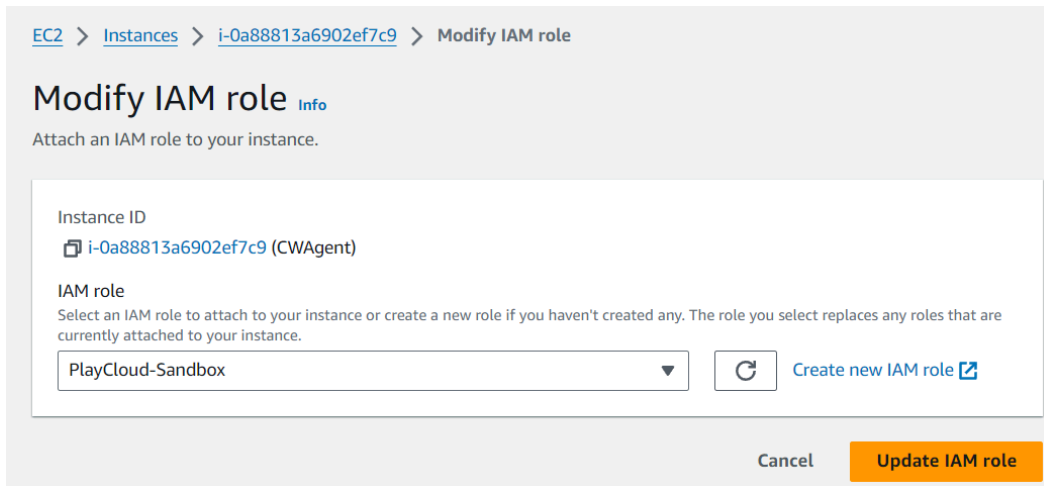
3. **Create an IAM Role:** Before CloudWatch will work with an EC2 instance, it must have an IAM role. For this lab, we will be using the provided IAM role.

4. Before you proceed, make sure that the **Instance state** is Running.

5. **Attach the IAM Role to the EC2 instance:** Navigate to the EC2 console and select your instance. Click on **Actions -> Security -> Modify IAM role**.



6. Choose PlayCloud-Sandbox.



7. Click Update IAM role.

Installing the CloudWatch Agent

1. **Download the CloudWatch Agent Package:** SSH into your instance and run the command below:

```
sudo wget https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb
```

This command simply downloads the agent from an S3 bucket to the current directory.

```

$ ssh -i CWAgent-ssh.pem ubuntu@ec2-44-204-105-59.compute-1.amazonaws.com
The authenticity of host 'ec2-44-204-105-59.compute-1.amazonaws.com (44.205.59)' can't be established.
ED25519 key fingerprint is SHA256:Pi/hNcaIi7Vx7VqPvz7Wa+7idRKXnGaSws8jgm88kxs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-204-105-59.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Jan 18 07:52:37 UTC 2024

System load: 0.0          Processes: 95
Usage of /: 20.5% of 7.57GB Users logged in: 0
Memory usage: 20%        IPv4 address for eth0: 192.168.5.43
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-192-168-5-43:~$ sudo wget https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb
--2024-01-18 07:53:25-- https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb
Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.217.140.0, 54.231.130.112, 54.231.136.32, ...
Connecting to s3.amazonaws.com (s3.amazonaws.com)|52.217.140.0|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 91736066 (87M) [application/octet-stream]
Saving to: 'amazon-cloudwatch-agent.deb'

amazon-cloudwatch-agent.d 100%[=====] 87.49M 97.5MB/s in 0.9s

2024-01-18 07:53:26 (97.5 MB/s) - 'amazon-cloudwatch-agent.deb' saved [91736066/91736066]

```

2. Install the CloudWatch Agent: Run the command below:

```
sudo dpkg -i ./amazon-cloudwatch-agent.deb
```

This command installs the agent using the DEB package manager.

```

ubuntu@ip-192-168-5-43:~$ sudo dpkg -i ./amazon-cloudwatch-agent.deb
Selecting previously unselected package amazon-cloudwatch-agent.
(Reading database ... 64799 files and directories currently installed.)
Preparing to unpack ./amazon-cloudwatch-agent.deb ...
create group cwagent, result: 0
create user cwagent, result: 0
Unpacking amazon-cloudwatch-agent (1.300032.3b392-1) ...
Setting up amazon-cloudwatch-agent (1.300032.3b392-1) ...
ubuntu@ip-192-168-5-43:~$

```

Configuring the CloudWatch agent to send custom metrics to CloudWatch

1. Edit the agent's configuration file. Run `sudo vi`

```
/opt/aws/amazon-cloudwatch-agent/bin/config.json
```

- Copy-paste the following:

```
{
  "agent": {
    "metrics_collection_interval": 60,
    "run_as_user": "root"
  },
  "metrics": {
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}"
    },
    "metrics_collected": {
      "disk": {
        "measurement": [
          "used_percent"
        ],
        "metrics_collection_interval": 60,
        "resources": [
          "/"
        ]
      },
      "mem": {
        "measurement": [
          "mem_used_percent"
        ],
        "metrics_collection_interval": 60
      }
    }
  }
}
```

- Enter `ESC :wq!` to save the file.

2. Stop and Start the CloudWatch Agent:

- Stop the CloudWatch Agent using the command line.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-
cloudwatch-agent-ctl -m ec2 -a stop
```

- Start the CloudWatch agent using the command line.

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

3. Validate Agent is Active: Start the agent, check the agent logs, and review the console for server log data and metrics. Run the following commands.

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

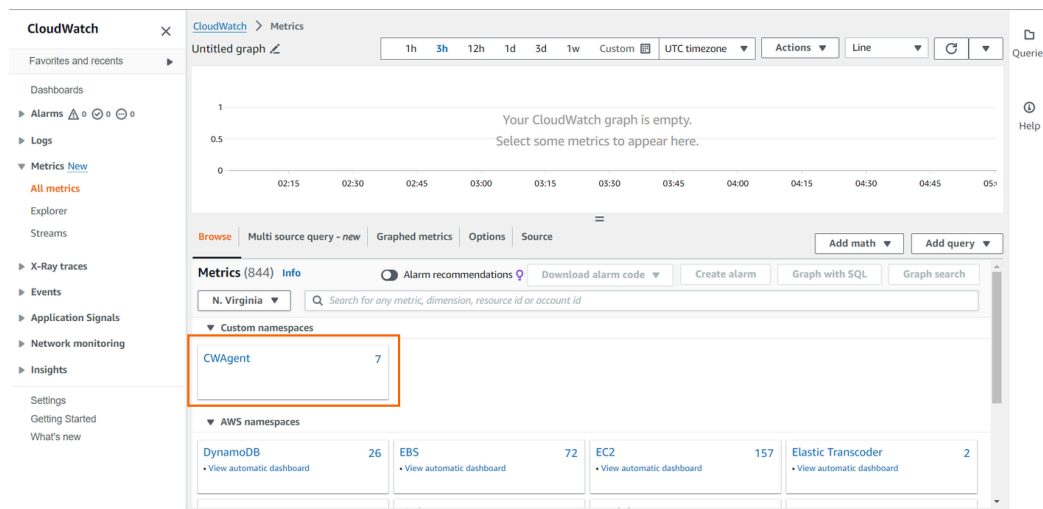
```
ubuntu@ip-192-168-5-40:~$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a status  
{  
  "status": "running",  
  "starttime": "2024-01-24T04:57:56+00:00",  
  "configstatus": "configured",  
  "version": "1.300032.3b392"  
}
```

```
cat /opt/aws/amazon-cloudwatch-  
agent/logs/configuration-validation.log
```

```
ubuntu@ip-192-168-5-40:~$ cat /opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log  
CWAgent/1.300032.3b392 (go1.21.5; linux; amd64)  
The given config: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml is valid
```

4. Confirm if the custom metrics are being sent to CloudWatch.

- Go to CloudWatch → Metrics → All Metrics → CWAgent



5. Now, let's break down each metric in your configuration:

- `metrics_collection_interval` : This is the frequency at which the metrics are collected. It's set to 60 seconds, which means the CloudWatch agent will collect metrics every minute.
- `run_as_user` : This is the user that the CloudWatch agent runs as. It's set to "root", which means the agent has full permissions on the system.
- `append_dimensions` : These are the default dimensions that are assigned to all collected metrics. They include:
 - `ImageId` : The ID of the Amazon Machine Image (AMI) used to launch the instance.

- `InstanceId` : The ID of the instance.
- `InstanceType` : The type of the instance.

The screenshot shows the AWS CloudWatch Metrics console on the left and a terminal window on the right displaying a JSON configuration file named `cat config.json`. In the console, a search for `ImageId, InstanceId, InstanceType, device, fstype, path` has been performed, resulting in 3 items. The terminal window shows the following JSON configuration:

```
cat config.json
{
  "agent": {
    "metrics_collection_interval": 60,
    "run_as_user": "root"
  },
  "metrics": {
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}"
    },
    "metrics_collected": {
      "disk": {
        "measurement": [
          "used_percent", "free", "total"
        ],
        "metrics_collection_interval": 60,
        "resources": [
          "/"
        ]
      },
      "mem": {
        "measurement": [
          "mem_used_percent", "cached", "total", "free"
        ],
        "metrics_collection_interval": 60
      }
    }
  }
}
```

- `metrics_collected` : This is where you specify the metrics that you want to collect. You're collecting disk and memory metrics:
 - `disk` : This collects metrics about disk usage. The metrics include:
 - `used_percent` : The percentage of disk space used.
 - `free` : The amount of free disk space.
 - `total` : The total amount of disk space.
 - `metrics_collection_interval` : The frequency at which the disk metrics are collected. It's set to 60 seconds.
 - `resources` : The file paths that the disk metrics are collected from. It's set to `"/"`, which represents the root directory.

The screenshot shows the AWS CloudWatch Metrics console on the left and a terminal window on the right displaying a JSON configuration file named `cat config.json`. In the console, a search for `ImageId, InstanceId, InstanceType, device, fstype, path` has been performed, resulting in 3 items. The terminal window shows the following JSON configuration:

```
cat config.json
{
  "agent": {
    "metrics_collection_interval": 60,
    "run_as_user": "root"
  },
  "metrics": {
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}"
    },
    "metrics_collected": {
      "disk": {
        "measurement": [
          "used_percent", "free", "total"
        ],
        "metrics_collection_interval": 60,
        "resources": [
          "/"
        ]
      },
      "mem": {
        "measurement": [
          "mem_used_percent", "cached", "total", "free"
        ],
        "metrics_collection_interval": 60
      }
    }
  }
}
```

- `mem` : This collects metrics about memory usage. The metrics include:
 - `mem_used_percent` : The percentage of memory used.
 - `cached` : The amount of cached memory.

- `total` : The total amount of memory.
- `free` : The amount of free memory.
- `metrics_collection_interval` : The frequency at which the memory metrics are collected. It's set to 60 seconds.

The screenshot shows the AWS CloudWatch Metrics console on the left and a terminal window on the right displaying the contents of `cat config.json`. The console shows a table of metrics for instances in N. Virginia, filtered by CWAgent, ImageId, InstanceId, and InstanceType. The metrics listed are `mem_used_percent`, `mem_total`, `mem_free`, and `mem_cached`. An orange box highlights these metrics, and an arrow points from this box to the `mem` section in the JSON configuration file on the right. The JSON file shows the configuration for the CloudWatch agent, including the `metrics_collection_interval` set to 60, and the `mem` section which lists the metrics to be collected: `mem_used_percent`, `cached`, `total`, and `free`.

```
cat config.json
{
  "agent": {
    "metrics_collection_interval": 60,
    "run_as_user": "root"
  },
  "metrics": {
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}"
    },
    "metrics_collected": {
      "disk": {
        "measurement": [
          "used_percent", "free", "total"
        ],
        "metrics_collection_interval": 60,
        "resources": [
          "/"
        ]
      },
      "mem": {
        "measurement": [
          "mem_used_percent", "cached", "total", "free"
        ],
        "metrics_collection_interval": 60
      }
    }
  }
}
```

Remember, you can customize this configuration to fit your specific needs by adding or removing metrics, changing the collection intervals, or modifying the dimensions.