



EC2에 CloudWatch Agent 구성하여 로그 모니터링하는 방법

1. EC2 에서 사용할 IAM Role을 생성합니다. Role 생성 시 trusted entity를 EC2로 선택하고 CloudWatchAgentServerPolicy, AmazonSSMFullAccess Policy를 추가합니다.

aws 서비스 user5 @ 5699-3439-7842 글로벌 지원

역할 > Log_monitoring

요약 역할 삭제

역할 ARN: arn:aws:iam::569934397842:role/Log_monitoring

역할 설명: Allows EC2 instances to call AWS services on your behalf. | 편집

인스턴스 프로파일 ARN: arn:aws:iam::569934397842:instance-profile/Log_monitoring

경로: /

생성 시간: 2021-05-20 14:47 UTC+0900

마지막 활동: 2021-05-20 17:49 UTC+0900 (오늘)

최대 세션 지속 시간: 1 시간 편집

권한: 신뢰 관계 | 태그 | 액세스 관리자 | 세션 취소

▼ Permissions policies (2 정책이 적용됨)

정책 연결 + 인라인 정책 추가

정책 이름	정책 유형	
CloudWatchAgentServerPolicy	AWS 관리형 정책	✕
AmazonSSMFullAccess	AWS 관리형 정책	✕

▶ Permissions boundary (not set)

▼ CloudTrail 이벤트를 기반으로 정책 생성

이 역할에 대한 액세스 활동을 기반으로 새 정책을 생성한 다음, 이를 사용자 지정하고 생성하여 이 역할에 연결할 수 있습니다. AWS는 CloudTrail 이벤트를 사용하여 사용된 서비스 및 작업을 식별하고 정책을 생성합니다. 자세히 알아보기

정책 생성

지난 7일 동안 정책 생성 요청이 없습니다.

의견 한국어 | 개인 정보 보호 정책 | 이용약관 | 쿠키 기본 설정

© 2008 - 2021, Amazon Web Services, Inc. 또는 계열사. All rights reserved.

2. EC2 (칼리) 를 선택하고 Attach/Replace IAM Role에서 방금 생성한 Role을 지정합니다.

aws 서비스 user5 @ 5699-3439-7842 서울 지원

인스턴스 (1/1) 정보

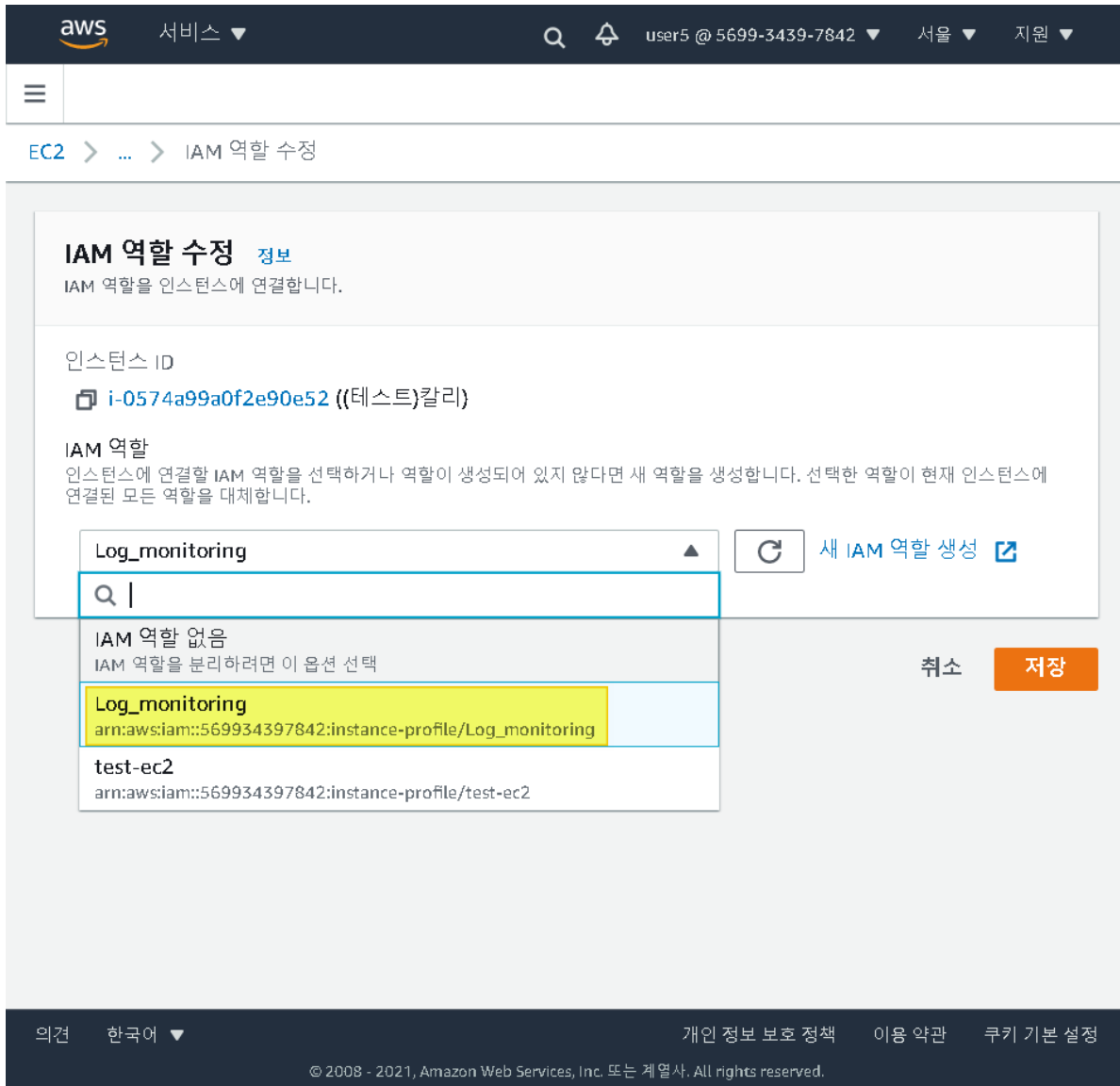
인스턴스 필터링

인스턴스 상태: running X

<input checked="" type="checkbox"/>	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사
<input checked="" type="checkbox"/>	(테스트)칼리	i-0574a99a0f2e90e52	실행 중	t2.micro	2/2

- 인스턴스 시작
- 템플릿으로 인스턴스 시작
- 연결
- 인스턴스 중지
- 인스턴스 시작
- 인스턴스 재부팅
- 인스턴스 최대 절전 모드
- 인스턴스 종료
- 인스턴스 설정 ▶
- 네트워킹 ▶
- 보안 ▶**
 - 보안 그룹 변경
 - Windows 암호 가져오기
 - IAM 역할 수정**
- 이미지 및 템플릿 ▶
- 모니터링 및 문제 해결 ▶

인스턴스: i-0574a99a0f2e90e52((테스트)칼리)



3. EC2에 ssh로 접속하여 CloudWatch Agent를 설치 후 구성합니다.

아키텍처	플랫폼	다운로드 링크	서명 파일 링크
		https://s3.regionAmazon Web Services-에이전트region/suse/amd64/latest/amazon-cloudwatch-agent.rpm	https://s3.region/suse/amd64/latest/amazon-cloudwatch-agent.rpm
AMD64	Debian	https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb	https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb
		https://s3.regionAmazon Web Services-에이전트region/debian/amd64/latest/amazon-cloudwatch-agent.deb	https://s3.region/debian/amd64/latest/amazon-cloudwatch-agent.deb
AMD64	Ubuntu	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb	https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb

```
wget https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb
//CloudWatch 에이전트를 다운로드합니다.
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb //Linux 서버에서 DEB 패키지를 다운로드한 경우 패키지가 있는 디렉터리로 변경하고 다음을 입력합니다.
```

4. 설치하고 나서 아래 명령을 수행하여 config.json을 생성해준다.

```
$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
=====
= Welcome to the AWS CloudWatch Agent Configuration Manager =
=====
On which OS are you planning to use the agent?
1. linux
2. windows
default choice: [1]:

Trying to fetch the default region based on ec2 metadata...
Are you using EC2 or On-Premises hosts?
1. EC2
2. On-Premises
default choice: [1]:

Which user are you planning to run the agent?
1. root
2. cwagent
3. others
default choice: [1]:

Do you want to turn on StatsD daemon?
1. yes
2. no
default choice: [1]:

Which port do you want StatsD daemon to listen to?
default choice: [8125]

What is the collect interval for StatsD daemon?
1. 10s
2. 30s
```

```

3. 60s
default choice: [1]:

What is the aggregation interval for metrics collected by StatsD daemon?
1. Do not aggregate
2. 10s
3. 30s
4. 60s
default choice: [4]:

Do you want to monitor metrics from CollectD?
1. yes
2. no
default choice: [1]:

Do you want to monitor any host metrics? e.g. CPU, memory, etc.
1. yes
2. no
default choice: [1]:

Do you want to monitor cpu metrics per core? Additional CloudWatch charges may apply.
1. yes
2. no
default choice: [1]:
2

Do you want to add ec2 dimensions (ImageId, InstanceId, InstanceType, AutoScalingGroupName) into all of your metrics if the info is available?
1. yes
2. no
default choice: [1]:

Would you like to collect your metrics at high resolution (sub-minute resolution)? This enables sub-minute resolution for all metrics, but at a higher cost.
1. 1s
2. 10s
3. 30s
4. 60s
default choice: [4]:

Which default metrics config do you want?
1. Basic
2. Standard
3. Advanced
4. None
default choice: [1]:
3
Current config as follows:
{
  "agent": {
    "metrics_collection_interval": 60,
    "run_as_user": "root"
  },
  "metrics": {
    "append_dimensions": {
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}"
    },
    "metrics_collected": {
      "collectd": {
        "metrics_aggregation_interval": 60
      },
      "cpu": {
        "measurement": [
          "cpu_usage_idle",
          "cpu_usage_iowait",
          "cpu_usage_user",
          "cpu_usage_system"
        ],
        "metrics_collection_interval": 60,
        "totalcpu": false
      },
      "disk": {
        "measurement": [
          "used_percent",
          "inodes_free"
        ],
        "metrics_collection_interval": 60,
        "resources": [
          "*"
        ]
      }
    }
  }
}

```

```

    },
    "diskio": {
      "measurement": [
        "io_time",
        "write_bytes",
        "read_bytes",
        "writes",
        "reads"
      ],
      "metrics_collection_interval": 60,
      "resources": [
        "*"
      ]
    },
    "mem": {
      "measurement": [
        "mem_used_percent"
      ],
      "metrics_collection_interval": 60
    },
    "netstat": {
      "measurement": [
        "tcp_established",
        "tcp_time_wait"
      ],
      "metrics_collection_interval": 60
    },
    "statsd": {
      "metrics_aggregation_interval": 60,
      "metrics_collection_interval": 10,
      "service_address": ":8125"
    },
    "swap": {
      "measurement": [
        "swap_used_percent"
      ],
      "metrics_collection_interval": 60
    }
  }
}

```

Are you satisfied with the above config? Note: it can be manually customized after the wizard completes to add additional items.

- 1. yes
- 2. no

default choice: [1]:

Do you have any existing CloudWatch Log Agent (<http://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AgentReference.html>) configuration f

- 1. yes
- 2. no

default choice: [2]:

Do you want to monitor any log files?

- 1. yes
- 2. no

default choice: [1]:

Log file path:

/var/log/messages

Log group name:

default choice: [messages]

Log stream name:

default choice: [{instance_id}]

Do you want to specify any additional log files to monitor?

- 1. yes
- 2. no

default choice: [1]:

Log file path:

/var/log/secure

Log group name:

default choice: [secure]

Log stream name:

default choice: [{instance_id}]

Do you want to specify any additional log files to monitor?

- 1. yes
- 2. no

default choice: [1]:

2

Saved config file to /opt/aws/amazon-cloudwatch-agent/bin/config.json successfully.
Current config as follows:

```
{
  "agent": {
    "metrics_collection_interval": 60,
    "run_as_user": "root"
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/var/log/messages",
            "log_group_name": "messages",
            "log_stream_name": "{instance_id}"
          },
          {
            "file_path": "/var/log/secure",
            "log_group_name": "secure",
            "log_stream_name": "{instance_id}"
          }
        ]
      }
    }
  },
  "metrics": {
    "append_dimensions": {
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}"
    },
    "metrics_collected": {
      "collectd": {
        "metrics_aggregation_interval": 60
      },
      "cpu": {
        "measurement": [
          "cpu_usage_idle",
          "cpu_usage_iowait",
          "cpu_usage_user",
          "cpu_usage_system"
        ],
        "metrics_collection_interval": 60,
        "totalcpu": false
      },
      "disk": {
        "measurement": [
          "used_percent",
          "inodes_free"
        ],
        "metrics_collection_interval": 60,
        "resources": [
          "*"
        ]
      },
      "diskio": {
        "measurement": [
          "io_time",
          "write_bytes",
          "read_bytes",
          "writes",
          "reads"
        ],
        "metrics_collection_interval": 60,
        "resources": [
          "*"
        ]
      },
      "mem": {
        "measurement": [
          "mem_used_percent"
        ],
        "metrics_collection_interval": 60
      },
      "netstat": {
        "measurement": [
          "tcp_established",
          "tcp_time_wait"
        ],
        "metrics_collection_interval": 60
      }
    }
  }
}
```



```

    },
    "statsd": {
      "metrics_aggregation_interval": 60,
      "metrics_collection_interval": 10,
      "service_address": ":8125"
    },
    "swap": {
      "measurement": [
        "swap_used_percent"
      ],
      "metrics_collection_interval": 60
    }
  }
}

```

Please check the above content of the config.
 The config file is also located at /opt/aws/amazon-cloudwatch-agent/bin/config.json.
 Edit it manually if needed.

Do you want to store the config in the SSM parameter store?

1. yes
2. no

default choice: [1]:

What parameter store name do you want to use to store your config? (Use 'AmazonCloudWatch-' prefix if you use our managed AWS policy)

default choice: [AmazonCloudWatch-linux]

Trying to fetch the default region based on ec2 metadata...

Which region do you want to store the config in the parameter store?

default choice: [ap-northeast-2]

Which AWS credential should be used to send json config to parameter store?

1. ASIAZYLTLN6RST6A4044(From SDK)
2. Other

default choice: [1]:

Successfully put config to parameter store AmazonCloudWatch-linux.

Program exits now.

[ec2-user@ip-10-10-1-16 ~]\$

5. 그 다음으로 생성한 config.json파일을 적용하여 cloudwatch agent를 실행합니다. 정확한 이유를 알 수 없으나 아래와 같은 에러가 발생합니다.

```

$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin
/opt/aws/amazon-cloudwatch-agent/bin/config-downloader --output-dir /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d --downlo
Successfully fetched the config and saved in /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp
Start configuration validation...
/opt/aws/amazon-cloudwatch-agent/bin/config-translator --input /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json --input-di
2019/11/17 06:27:37 Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp ...
Valid Json input schema.
I! Detecting runasuser...
No csm configuration found.
Configuration validation first phase succeeded
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent -schematest -config /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-age
Configuration validation second phase failed
===== Error Log =====
2019/11/17 06:27:37 I! AmazonCloudWatchAgent Version 1.231221.0.
2019/11/17 06:27:37 E! Error parsing /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml, open /usr/share/collectd/types.db:
[ec2-user@ip-10-10-1-16 ~]$

```

6. /usr/share/collectd/types.db 파일이 없다는 에러를 해결하기 위해 root로 접속하여 빈 types.db 파일을 만들어주면 해결이 된다.

```

$ mkdir /usr/share/collectd
$ cd /usr/share/collectd
$ touch types.db

```

7. types.db 파일을 만들고 다시 수행하면 정상적으로 실행이 됩니다.

```
$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin
/opt/aws/amazon-cloudwatch-agent/bin/config-downloader --output-dir /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d --downlo
Successfully fetched the config and saved in /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp
Start configuration validation...
/opt/aws/amazon-cloudwatch-agent/bin/config-translator --input /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json --input-di
2019/11/17 06:33:23 Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp ...
Valid Json input schema.
I! Detecting runasuser...
No csm configuration found.
Configuration validation first phase succeeded
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent -schematest -config /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-age
Configuration validation second phase succeeded
Configuration validation succeeded
Created symlink from /etc/systemd/system/multi-user.target.wants/amazon-cloudwatch-agent.service to /etc/systemd/system/amazon-cloudwatch-a
Redirecting to /bin/systemctl restart amazon-cloudwatch-agent.service
[ec2-user@ip-10-10-1-16 ~]$
```