



# 로그 수집을 위한 설정 방법

소 속	BoB 9 기 디지털포렌식트랙
팀 명	Cloud?Kloud!
작성 일자	2020.11.05

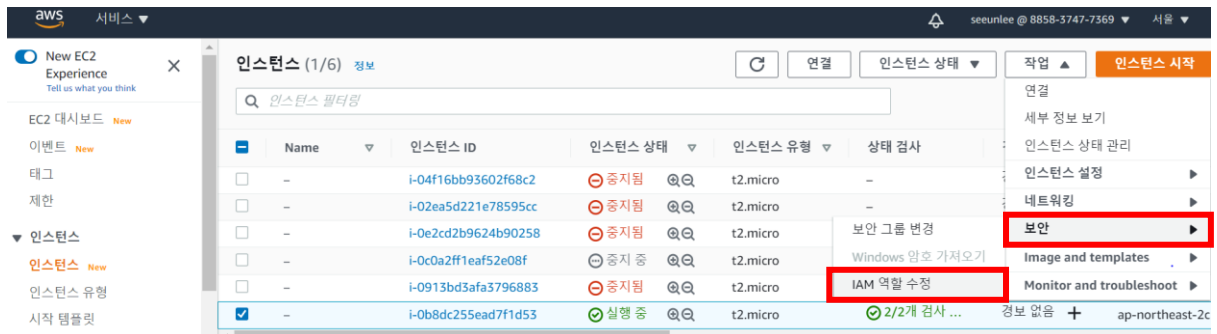
## 목 차

1. EC2 Linux 인스턴스 로그 수집 .....	3
2. EC2 Windows 인스턴스 로그 수집 .....	11
3. RDS 데이터베이스 로그 수집 .....	22
4. S3 로그 수집.....	27

## 1. EC2 Linux 인스턴스 로그 수집

### 1.1. Linux 인스턴스에서 IAM 역할 생성 및 지정

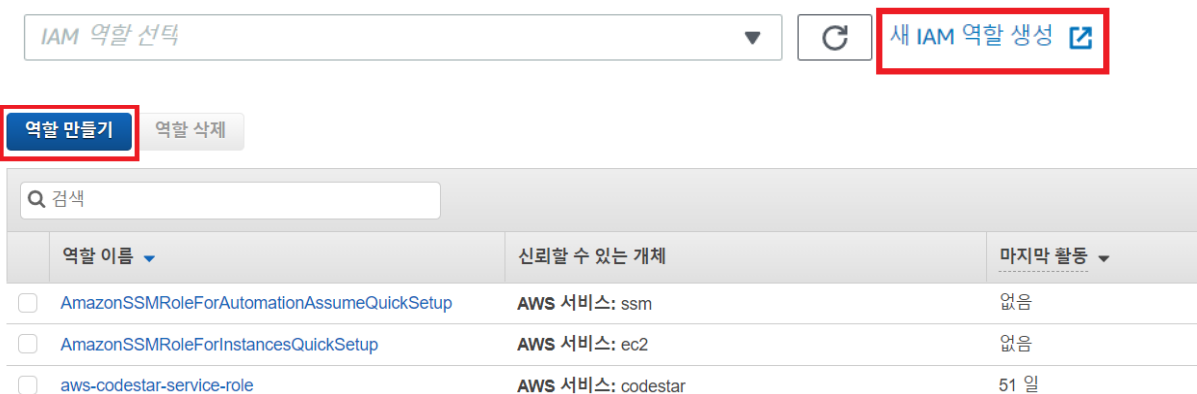
- CloudWatch를 통해 로그를 추출할 인스턴스를 선택한 후, "작업" 탭 - 보안 - "IAM 역할 수정"을 클릭한다.



- "새 IAM 역할 생성" 버튼을 눌러서 IAM 역할 페이지로 이동하여, IAM 역할을 생성한다.

#### IAM 역할

인스턴스에 연결할 IAM 역할을 선택하거나 역할이 생성되어 있지 않다면 새 역할을 생성합니다. 선택한 역할이 현재 인스턴스에 연결된 모든 역할을 대체합니다.



- 사용 사례 선택 부분에서는 "EC2"를 생성하고, 권한 정책 연결 부분에서는 "CloudWatchAgentServerPolicy"와 "AmazonSSMFullAccess"정책을 검색하여 체크한다.

## 역할 만들기

1 2 3 4

신뢰할 수 있는 유형의 개체 선택

**AWS 서비스**  
EC2, Lambda 및 기타

**다른 AWS 계정**  
귀하 또는 타사 소유

**웹 ID**  
Cognito 또는 OpenID 공급자

**SAML 2.0 연동**  
귀사 디렉터리

AWS 서비스가 사용자를 대신하여 작업을 수행하도록 허용합니다. [자세히 알아보기](#)

## 사용 사례 선택

일반 사용 사례

### EC2

Allows EC2 instances to call AWS services on your behalf.

### Lambda


Allows Lambda functions to call AWS services on your behalf.

## 역할 만들기

1 2 3 4

### ▼ 권한 정책 연결

새로운 역할에 연결할 정책을 1개 이상 선택하십시오.

정책 생성 

정책 필터 ▼

CloudWatchAgentServerPolicy

1 결과 표시

	정책 이름 ▼	사용 용도
<input checked="" type="checkbox"/>	CloudWatchAgentServerPolicy	Permissions policy (5)

정책 필터 ▼

AmazonSSMFullAccess

1 결과 표시

	정책 이름 ▼	사용 용도
<input checked="" type="checkbox"/>	AmazonSSMFullAccess	Permissions policy (2)

### ▶ 권한 경계 설정

\* 필수

[취소](#)

[이전](#)

[다음: 태그](#)

- 역할 이름을 설정한 뒤, "역할 만들기" 버튼을 클릭하여 역할을 생성한다.

## 검토

생성하기 전에 아래에 필요한 정보를 입력하고 이 역할을 검토하십시오.

역할 이름\* CloudWatchAgent  
영숫자 및 '+, @, \_' 문자를 사용합니다. 최대 64자입니다.

역할 설명 Allows EC2 instances to call AWS services on your behalf.  
최대 1000자입니다. 영숫자 및 '+, @, \_' 문자를 사용합니다.

신뢰할 수 있는 개체 AWS 서비스: ec2.amazonaws.com

정책 CloudWatchAgentServerPolicy  
AmazonSSMFullAccess

권한 경계 권한 경계가 설정되지 않았습니다

태그가 추가되지 않았습니다.

\* 필수

취소

이전

역할 만들기

- 다시 Linux 인스턴스의 IAM 역할 수정으로 돌아가, 생성된 IAM 역할을 선택한 후 "저장" 버튼을 클릭하여 설정을 완료한다.

**IAM 역할 수정** 정보  
IAM 역할을 인스턴스에 연결합니다.

인스턴스 ID  
i-0b8dc25ead7f1d53

IAM 역할  
인스턴스에 연결할 IAM 역할을 선택하거나 역할이 생성되어 있지 않다면 새 역할을 생성합니다. 선택한 역할이 현재 인스턴스에 연결된 모든 역할을 대체합니다.

CloudWatchAgent ▼ 새 IAM 역할 생성

취소 저장

## 1.2. Linux 인스턴스에 CloudWatch Agent 설치

- Linux 인스턴스에 접속하여, 아래 명령어를 통해 CloudWatch Agent 설치를 진행한다.

```
13.125.249.139 - PuTTY
[ec2-user@ip-172-31-40-46 ~]$ wget https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm
--2020-11-04 14:49:04-- https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm
Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.217.86.86
Connecting to s3.amazonaws.com (s3.amazonaws.com)|52.217.86.86|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38749838 (37M) [application/octet-stream]
Saving to: 'amazon-cloudwatch-agent.rpm'

100%[=====>] 38,749,838  7.74MB/s  in 5.9s

2020-11-04 14:49:11 (6.24 MB/s) - 'amazon-cloudwatch-agent.rpm' saved [38749838/38749838]

[ec2-user@ip-172-31-40-46 ~]$ sudo rpm -U ./amazon-cloudwatch-agent.rpm
[ec2-user@ip-172-31-40-46 ~]$
```

```
$ wget https://s3.amazonaws.com/amazoncloudwatch-agent/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm
```

```
$ sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

## 1.3. CloudWatch Agent의 config 파일 설정

- 아래 명령어를 통해 CloudWatch Agent에 대한 설정을 진행한다.

```
[ec2-user@ip-172-31-40-46 ~]$ 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]:
```

```
$sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
```

- 프로그램이 실행되면 아래와 같이 설정해준다. (나머지는 모두 default(엔터입력)로 설정)  
Linux 인스턴스에서는 시스템 메시지(/var/log/message), SSH 접속 로그(/var/log/secure), 로그인 레코드(/var/log/lastlog), YUM 패키지 로그(/var/log/yum.log)를 수집할 것이다. 따라서 다음과 같이 Log file path에 입력한다.
- message 로그를 통해 시스템의 전반적인 로그를, secure 로그를 통해 SSH 접속 기록 및 시도를 확인할 수 있다. 또한, lastlog를 통해 로그인 기록을, yum.log를 통해서 패키지 매니저에서 무엇을 수행(설치/삭제)했는지 확인할 수 있다.

```
Do you want to monitor any log files?
1. yes
2. no
default choice: [1]:
1
Log file path:
/var/log/messages ← /var/log/messages 입력
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 ← /var/log/secure 입력
Log group name:
default choice: [secure]

Log stream name:
default choice: [{instance_id}]
```

```
Log file path:
/var/log/lastlog ← /var/log/lastlog 입력
Log group name:
default choice: [lastlog]

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]:

Do you want to specify any additional log files to monitor?
1. yes
2. no
default choice: [1]:

Log file path:
/var/log/yum.log ← /var/log/yum.log 입력
Log group name:
default choice: [yum.log]

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.
```

마지막으로 AWS 보안증명(Credential)을 설정하면(default 값) config.json 파일 생성을 완료한다.

```
Which AWS credential should be used to send json config to parameter store?
1. ASIA44QAHGX4XB2EHBOT(From SDK)
2. Other
default choice: [1]:

Successfully put config to parameter store AmazonCloudWatch-linux.
Program exits now.
[ec2-user@ip-172-31-40-46 ~]$
```

- 생성한 config.json 파일을 적용시키기 위해 아래 명령어를 입력한다. 명령어를 실행하면, /usr/share/collectd/types.db가 존재하지 않는다는 에러가 발생한다.

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

```
[ec2-user@ip-172-31-40-46 ~]$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json -s
/opt/aws/amazon-cloudwatch-agent/bin/config-downloader --output-dir /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d --download-source file:/opt/aws/amazon-cloudwatch-agent/bin/config.json --mode ec2 --config /opt/aws/amazon-cloudwatch-agent/etc/common-config.toml --multi-config default
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-dir /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d --output /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml --mode ec2 --config /opt/aws/amazon-cloudwatch-agent/etc/common-config.toml --multi-config default
2020/11/04 15:06:02 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-agent.toml
Configuration validation second phase failed
===== Error Log =====
2020-11-04T15:06:02Z E! [telegraf] Error running agent: Error parsing /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml, open /usr/share/collectd/types.db: no such file or directory
[ec2-user@ip-172-31-40-46 ~]$
```

이 경우에는 직접 /usr/share/collectd 위치로 이동하여 types.db 파일을 생성해주어야 한다. 아래 명령어를 통해 파일을 생성하면 된다.

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

```
[ec2-user@ip-172-31-40-46 ~]$ sudo mkdir /usr/share/collectd
[ec2-user@ip-172-31-40-46 ~]$ cd /usr/share/collectd
[ec2-user@ip-172-31-40-46 collectd]$ sudo touch types.db
[ec2-user@ip-172-31-40-46 collectd]$ ls
types.db
[ec2-user@ip-172-31-40-46 collectd]$
```



types.db 파일을 생성한 후, 이전 명령어를 다시 입력하면 정상적으로 설정이 완료되며 CloudWatch Agent 가 동작할 수 있다.

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

```
[ec2-user@ip-172-31-40-46 collectd]$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json -s
/opt/aws/amazon-cloudwatch-agent/bin/config-downloader --output-dir /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d --download-source file:/opt/aws/amazon-cloudwatch-agent/bin/config.json --mode ec2 --config /opt/aws/amazon-cloudwatch-agent/etc/common-config.toml --multi-config default
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-dir /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d --output /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml --mode ec2 --config /opt/aws/amazon-cloudwatch-agent/etc/common-config.toml --multi-config default
2020/11/04 15:09:05 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-agent.toml
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-agent.service.
Redirecting to /bin/systemctl restart amazon-cloudwatch-agent.service
[ec2-user@ip-172-31-40-46 collectd]$
```

## 1.4. CloudWatch 로그 확인

- CloudWatch 서비스의 "로그 그룹"에서 생성된 로그들을 확인할 수 있다.

로그			
로그 그룹	○ /aws/rds/instance/database-2/slowquery	둘러보기	만기 없음
인사이트	○ Application	둘러보기	만기 없음
지표	○ Security	둘러보기	만기 없음
Explorer 베타	○ System	둘러보기	만기 없음
이벤트	○ amazon-ssm-agent.log	둘러보기	만기 없음
규칙	○ aws-cloudtrail-logs-885837477369-60089017	둘러보기	만기 없음
이벤트 버스	○ aws-cloudtrail-logs-885837477369-7f6e04f7	둘러보기	만기 없음
ServiceLens	○ aws-cloudtrail-logs-885837477369-test	둘러보기	만기 없음
서비스 맵	○ bmp	둘러보기	만기 없음
기록	○ filesystem	둘러보기	만기 없음
Container Insights 새로운	○ lastlog	둘러보기	만기 없음
Resources	○ messages	둘러보기	만기 없음
Performance monitoring	○ secure	둘러보기	만기 없음
Lambda Insights 새로운	○ syslog	둘러보기	만기 없음
Multi-function	○ win_ec2_test_lks	둘러보기	만기 없음
	○ yum.log	둘러보기	만기 없음

이벤트 필터링		모두 2020-
시간(UTC +00:00)	메시지	
2020-11-04		
지금은 발견된 이전 이벤트가 없습니다. 재시도.		
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 journal: Runtime journal is using 6.1M (max allowed 49.1M, trying to leave 73.7M free of 485.5M available → current	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: Linux version 4.14.193-149.317.amzn2.x86_64 (mockbuild@ip-10-0-1-32) (gcc version 7.3.1 20180712 (Red	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-4.14.193-149.317.amzn2.x86_64 root=UUID=b24eb1ea-ab1c-	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: e820: BIOS-provided physical RAM map:	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: BIOS-e820: [mem 0x0000000000000000-0x00000000000009dfff] usable	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: BIOS-e820: [mem 0x00000000000009e000-0x00000000000009ffff] reserved	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: BIOS-e820: [mem 0x0000000000000e0000-0x0000000000000fffff] reserved	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: BIOS-e820: [mem 0x000000000000100000-0x0000000000003fffff] usable	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: BIOS-e820: [mem 0x000000000000c000000-0x000000000000ffffff] reserved	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: NX (Execute Disable) protection: active	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: SMBIOS 2.7 present.	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: DMI: Xen HVM domU, BIOS 4.2.amazon 08/24/2006	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: Hypervisor detected: Xen HVM	
▶ 15:09:06	Nov 2 08:10:34 ip-172-31-40-46 kernel: Xen version 4.2.	

이벤트 필터링		모두 2020-11-03 (15:34:36) ▾
시간(UTC +00:00)	메시지	
2020-11-04		
지금은 발견된 이전 이벤트가 없습니다. 재시도.		
▶ 15:09:06	Nov 2 08:10:44 ip-172-31-40-46 useradd[3205]: new group: name=ec2-user, GID=1000	
▶ 15:09:06	Nov 2 08:10:44 ip-172-31-40-46 useradd[3205]: new user: name=ec2-user, UID=1000, GID=1000, home=/home/ec2-user, shell=/bin/bash	
▶ 15:09:06	Nov 2 08:10:44 ip-172-31-40-46 useradd[3205]: add 'ec2-user' to group 'adm'	
▶ 15:09:06	Nov 2 08:10:44 ip-172-31-40-46 useradd[3205]: add 'ec2-user' to group 'wheel'	
▶ 15:09:06	Nov 2 08:10:44 ip-172-31-40-46 useradd[3205]: add 'ec2-user' to group 'systemd-journal'	
▶ 15:09:06	Nov 2 08:10:44 ip-172-31-40-46 useradd[3205]: add 'ec2-user' to shadow group 'adm'	
▶ 15:09:06	Nov 2 08:10:44 ip-172-31-40-46 useradd[3205]: add 'ec2-user' to shadow group 'wheel'	
▶ 15:09:06	Nov 2 08:10:44 ip-172-31-40-46 useradd[3205]: add 'ec2-user' to shadow group 'systemd-journal'	
▶ 15:09:06	Nov 2 08:10:45 ip-172-31-40-46 sshd[3388]: Server listening on 0.0.0.0 port 22.	
▶ 15:09:06	Nov 2 08:10:45 ip-172-31-40-46 sshd[3388]: Server listening on :: port 22.	
▶ 15:09:06	Nov 2 08:15:42 ip-172-31-40-46 sshd[3470]: error: AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys ec2-user SHA256:iaQ6S3Lh/pgoUYKhHD1bn6FziC/	
▶ 15:09:06	Nov 2 08:15:42 ip-172-31-40-46 sshd[3470]: error: AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys ec2-user SHA256:iaQ6S3Lh/pgoUYKhHD1bn6FziC/	
▶ 15:09:06	Nov 2 08:15:42 ip-172-31-40-46 sshd[3470]: Accepted publickey for ec2-user from 112.146.53.117 port 4546 ssh2: RSA SHA256:iaQ6S3Lh/pgoUYKhHD1bn6FziCWUk+	
▶ 15:09:06	Nov 2 08:15:42 ip-172-31-40-46 sshd[3470]: pam_unix(sshd:session): session opened for user ec2-user by (uid=0)	
▶ 15:09:06	Nov 2 08:19:25 ip-172-31-40-46 sudo: ec2-user : TTY=pts/0 ; PWD=/home/ec2-user ; USER=root ; COMMAND=/bin/yum update	
▶ 15:09:06	Nov 2 08:19:25 ip-172-31-40-46 sudo: pam_unix(sudo:session): session opened for user root by ec2-user(uid=0)	
▶ 15:09:06	Nov 2 08:20:09 ip-172-31-40-46 sudo: pam_unix(sudo:session): session closed for user root	
▶ 15:09:06	Nov 2 08:20:39 ip-172-31-40-46 sshd[3470]: pam_unix(sshd:session): session closed for user ec2-user	
▶ 15:09:06	Nov 2 08:21:04 ip-172-31-40-46 sshd[12469]: error: AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys ec2-user SHA256:iaQ6S3Lh/pgoUYKhHD1bn6FziC/	

## 2. EC2 Window 인스턴스 로그 수집

### 2.1. Windows 인스턴스 연결

- EC2 Window 인스턴스에 연결하기 위해 RDP 방식을 사용하였다.

EC2 Window 인스턴스 - "연결" 탭 - "RDP 클라이언트" - "원격 데스크톱 파일 다운로드 "

The screenshot shows the AWS Management Console interface for an EC2 instance. The breadcrumb navigation is "EC2 > 인스턴스 > i-0c0a2ff1eaf52e08f". The instance details are displayed in a table:

i-0c0a2ff1eaf52e08f에 대한 인스턴스 요약 정보		
인스턴스 ID	퍼블릭 IPv4 주소	프라이빗 IPv4 주소
i-0c0a2ff1eaf52e08f	54.180.104.130   <a href="#">개방 주소법</a>	172.31.38.124
인스턴스 상태	퍼블릭 IPv4 DNS	프라이빗 IPv4 DNS
실행 중	ec2-54-180-104-130.ap-northeast-2.compute.amazonaws.com   <a href="#">개방 주소법</a>	ip-172-31-38-124.ap-northeast-2.compute.internal
인스턴스 유형	탄력적 IP 주소	VPC ID
t2.micro	-	vpc-e9f3780
IAM 역할	서브넷 ID	
-	subnet-5cae410	

The "연결" (Connect) button is highlighted with a red box.

The screenshot shows the "인스턴스에 연결" (Connect to Instance) page for the same EC2 instance. The breadcrumb navigation is "EC2 > 인스턴스 > i-0c0a2ff1eaf52e08f > 인스턴스에 연결". The page title is "인스턴스에 연결 정보".

다음 옵션 중 하나를 사용하여 인스턴스 i-0c0a2ff1eaf52e08f에 연결

**Session Manager** **RDP 클라이언트**

선택한 원격 데스크톱 클라이언트를 사용하고 아래의 RDP 바로 가기 파일을 다운로드하여 실행하면 Windows 인스턴스에 연결할 수 있습니다.

**원격 데스크톱 파일 다운로드**

메시지가 표시되면 다음 세부 정보를 사용하여 인스턴스에 연결합니다.

Public DNS	사용자 이름
ec2-54-180-104-130.ap-northeast-2.compute.amazonaws.com	Administrator

암호 [암호 가져오기](#)

인스턴스를 디렉터리에 조인한 경우 디렉터리 자격 증명을 사용하여 인스턴스에 연결할 수 있습니다.

- 다운로드 된 파일을 실행시키면 해당 인스턴스와 연결하는 창이 나오게 되고, 자격증명을 위해 비밀번호를 입력하도록 나온다.

비밀번호는 "Windows 암호 가져오기"를 이용하여 이전에 따로 저장한 키페어(.pem)를 가져와 해독한다. 여기서 추출된 비밀번호를 RDP 자격증명의 비밀번호에 입력해준다.

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the user 'seeunlee @ 8858-3747-7369' and region '서울'. Below this, the '인스턴스 (1/5) 정보' (Instances (1/5) Info) page is visible, showing a table of instances. The instance 'i-0c0a2ff1eaf52e08f' is selected, and its context menu is open, highlighting 'Windows 암호 가져오기' (Import Windows password) and '보안' (Security).

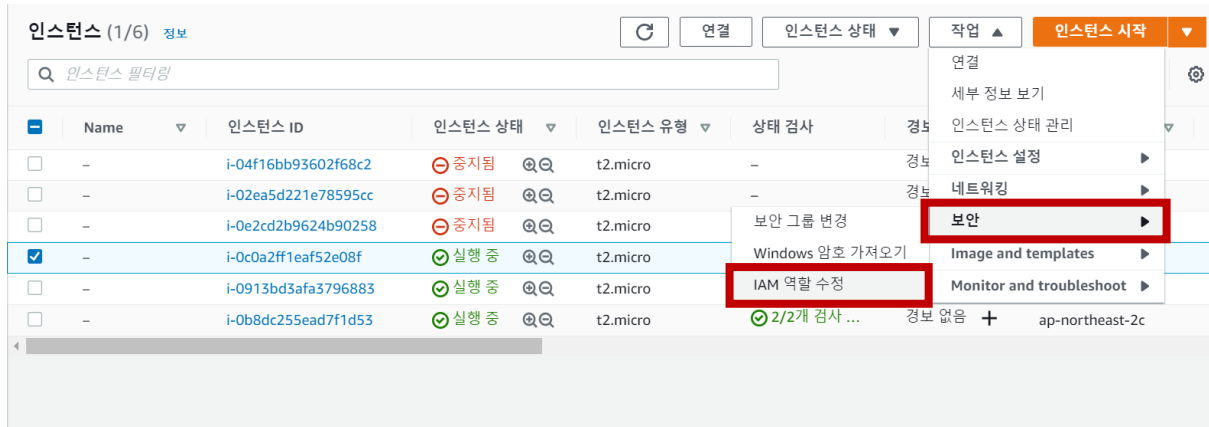
The 'Windows 암호 가져오기' (Import Windows password) page is shown below. It contains the following information:

- Windows 암호 가져오기 정보** (Import Windows password info): Retrieve and decrypt the initial Windows administrator password for this instance.
- To decrypt the password, you will need your key pair for this instance.**
- Key pair associated with this instance**: seeunlee\_win
- Browse to your key pair:** A 'Browse' button is highlighted with a red box.
- Or copy and paste the contents of the key pair below:** A large text area for pasting the key pair content.
- Buttons**: '취소' (Cancel) and '암호 해독' (Decrypt password) buttons are at the bottom right, with the latter highlighted by a red box.

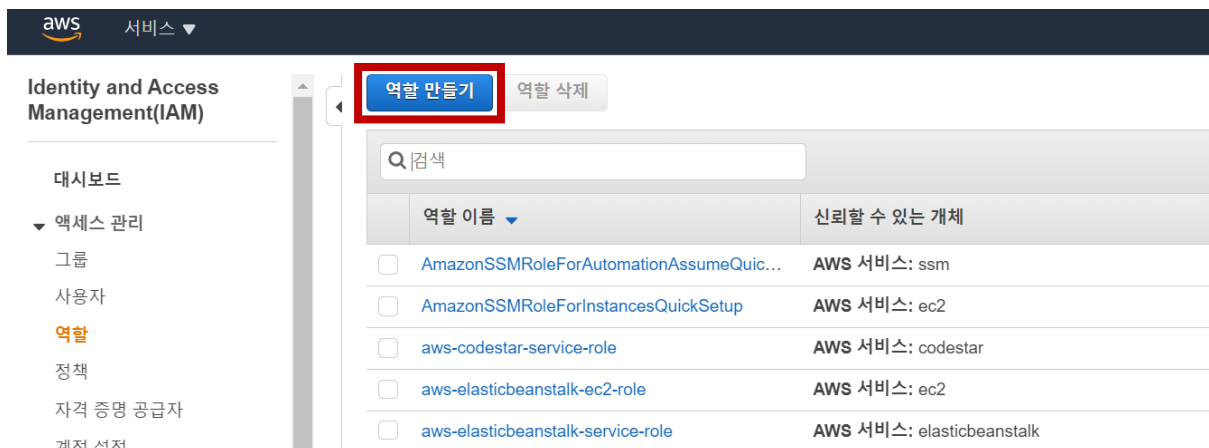
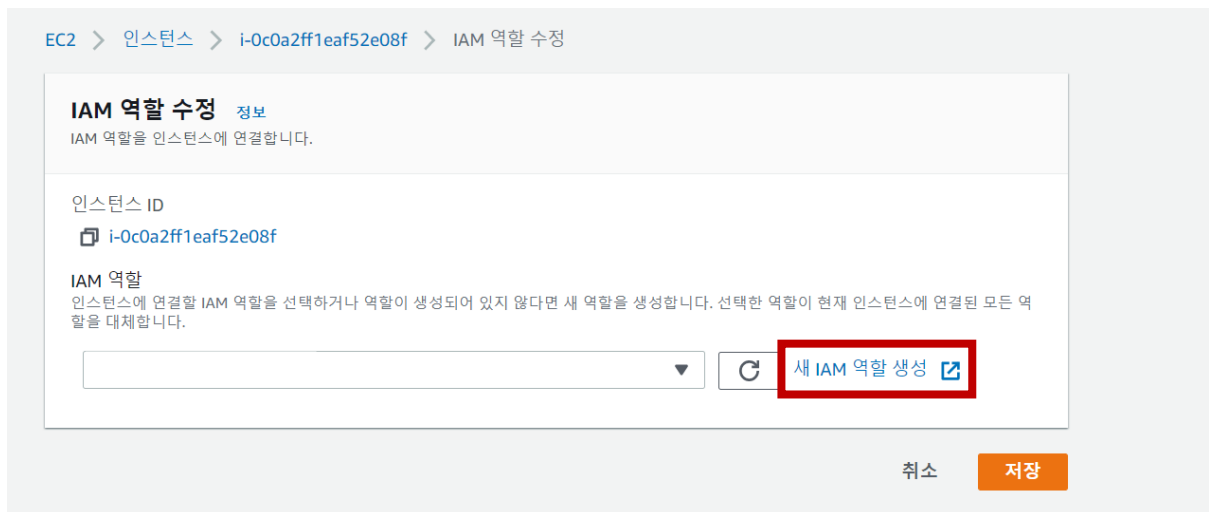
## 2.2. Windows 인스턴스에 IAM 역할 설정

- 로그를 추출할 인스턴스에 IAM 역할을 부여한다.

Windows 인스턴스 - "작업" 탭 - 보안 - "IAM 역할 수정"



- "새 IAM 역할 생성" 버튼을 눌러서 IAM 역할 페이지로 이동하여, IAM 역할을 생성한다.



- 사용 사례 선택 부분에서는 "EC2"를 생성하고, 권한 정책 연결 부분에서는 "CloudWatchAgentServerPolicy" 정책을 검색하여 체크한다.

## 역할 만들기

1 2 3 4

신뢰할 수 있는 유형의 개체 선택

**AWS 서비스**  
 EC2, Lambda 및 기타

**다른 AWS 계정**  
 귀하 또는 타사 소유

**웹 ID**  
 Cognito 또는 OpenID 공급자

**SAML 2.0 연동**  
 귀사 디렉터리

AWS 서비스가 사용자를 대신하여 작업을 수행하도록 허용합니다. [자세히 알아보기](#)

## 사용 사례 선택

일반 사용 사례

**EC2**  
Allows EC2 instances to call AWS services on your behalf.

**Lambda**  
Allows Lambda functions to call AWS services on your behalf.

또는 서비스를 선택하여 해당 서비스의 사용 사례 확인

[API Gateway](#)
[CloudWatch Events](#)
[EKS](#)
[KMS](#)
[Rekognition](#)  
[AWS Backup](#)
[CodeBuild](#)
[EMR](#)
[Kinesis](#)
[RoboMaker](#)

\* 필수

취소

다음: 권한

## 역할 만들기

1 2 3 4

### ▼ 권한 정책 연결

새로운 역할에 연결할 정책을 1개 이상 선택하십시오.

정책 생성

정책 필터
 
 1 결과 표시

정책 이름	사용 용도
<input checked="" type="checkbox"/> CloudWatchAgentServerPolicy	Permissions policy (4)

\* 필수

취소

이전

다음: 태그

## 역할 만들기

1 2 3 4

### 태그 추가(선택 사항)

IAM 태그는 사용자 역할에 추가할 수 있는 키-값 페어입니다. 태그는 이메일 주소와 같은 사용자 정보를 포함하거나 정책과 같은 내용일 수 있습니다. 태그를 사용하여 이 역할에 대한 액세스를 구성, 추적 또는 제어할 수 있습니다. [자세히 알아보기](#)

키	값(선택 사항)	제거
<input type="text" value="새 키 추가"/>	<input type="text"/>	

50 태그를 더 추가할 수 있습니다.

취소

이전

다음: 검토

- 역할 이름을 설정한 뒤, 역할을 생성한다.

## 역할 만들기

1 2 3 4

### 검토

생성하기 전에 아래에 필요한 정보를 입력하고 이 역할을 검토하십시오.

역할 이름\* cloudwatchagentserverpolicy

영숫자 및 '+', '@', '-' 문자를 사용합니다. 최대 64자입니다.

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

최대 1000자입니다. 영숫자 및 '+', '@', '-' 문자를 사용합니다.

신뢰할 수 있는 개체 AWS 서비스: ec2.amazonaws.com

정책  CloudWatchAgentServerPolicy [↗](#)

권한 경계 권한 경계가 설정되지 않았습니다

취소

이전

역할 만들기

### Identity and Access Management(IAM)

대시보드

▼ 액세스 관리

그룹

사용자

역할

정책

자격 증명 공급자

✓ 역할 cloudwatchagentserverpolicy 이(가) 생성되었습니다.

역할 만들기

역할 삭제

Q 검색

역할 이름 ▼	신뢰할 수 있는 개체	마지막 활동 ▼
<input type="checkbox"/> AmazonSSMRoleForAutomationAssumeQuic...	AWS 서비스: ssm	없음
<input type="checkbox"/> AmazonSSMRoleForInstancesQuickSetup	AWS 서비스: ec2	없음

- 다시 IAM 역할 수정 페이지로 돌아와, 방금 생성한 IAM 역할(cloudwatchagentserverpolicy)을 지정해준다.

EC2 > 인스턴스 > i-0c0a2ff1eaf52e08f > IAM 역할 수정

**IAM 역할 수정** 정보

IAM 역할을 인스턴스에 연결합니다.

인스턴스 ID  
i-0c0a2ff1eaf52e08f

**IAM 역할**  
인스턴스에 연결할 IAM 역할을 선택하거나 역할이 생성되어 있지 않다면 새 역할을 생성합니다. 선택한 역할이 현재 인스턴스에 연결된 모든 역할을 대체합니다.

cloudwatchagentserverpolicy

새 IAM 역할 생성

취소 **저장**

☑ Successfully attached cloudwatchagentserverpolicy to instance i-0c0a2ff1eaf52e08f

인스턴스 (1/6) 정보 연결 인스턴스 상태 ▼ 작업 ▼ 인스턴스 시작 ▼

🔍 인스턴스 필터링

	Name ▼	인스턴스 ID	인스턴스 상태 ▼	인스턴스 유형 ▼	상태 검사	경보 상태	가용 영역 ▼
<input type="checkbox"/>	-	i-04f16bb93602f68c2	⊖ 중지됨	t2.micro	-	경보 없음 +	ap-northeast-2a
<input type="checkbox"/>	-	i-02ea5d221e78595cc	⊖ 중지됨	t2.micro	-	경보 없음 +	ap-northeast-2c
<input type="checkbox"/>	-	i-0e2cd2b9624b90258	⊖ 중지됨	t2.micro	-	경보 없음 +	ap-northeast-2c
<input checked="" type="checkbox"/>	-	i-0c0a2ff1eaf52e08f	⊕ 실행 중	t2.micro	⊕ 2/2개 검사 ...	경보 없음 +	ap-northeast-2c
<input type="checkbox"/>	-	i-0913bd3afa3796883	⊕ 실행 중	t2.micro	⊕ 2/2개 검사 ...	경보 없음 +	ap-northeast-2c
<input type="checkbox"/>	-	i-0b8dc255ead7f1d53	⊕ 실행 중	t2.micro	⊕ 2/2개 검사 ...	경보 없음 +	ap-northeast-2c

## 2.3. Windows 인스턴스에 CloudWatch agent 설치

- RDP 클라이언트를 통해 Windows 인스턴스로 접속하여, 아래 링크로 파일을 다운로드한다.

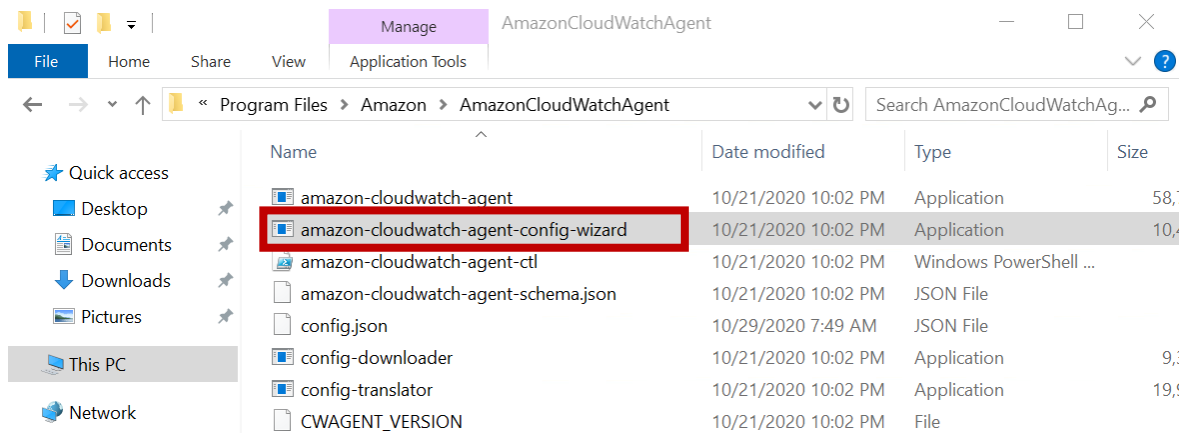
<https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi>

다운로드가 완료되면 실행하여 에이전트를 설치해준다. (설치가 완료되면 프로그램이 자동으로 종료된다)

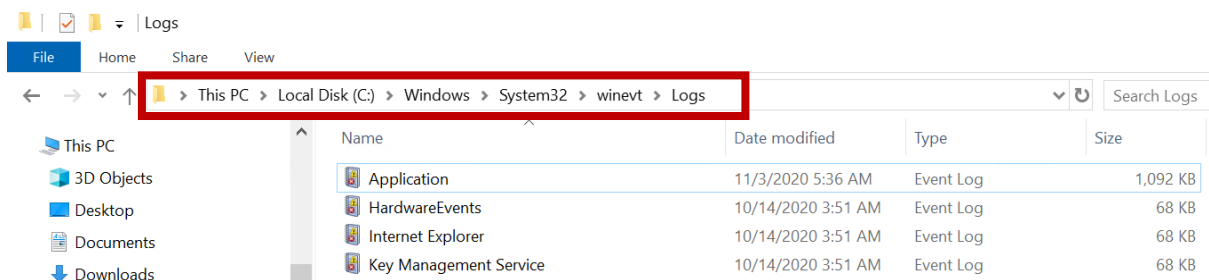


## 2.4. CloudWatch agent의 config 파일 설정

- 아래와 같은 경로로 들어가 "amazon-cloudwatch-agent-config-wizard"를 실행시킨다.



- 프로그램이 실행되면 아래와 같이 설정해준다. (나머지는 모두 default(엔터입력)로 설정)  
Windows 인스턴스에서는 이벤트 로그(System, Application, Security)를 수집할 것이다. 따라서 다음과 같이 Log file path에 입력한다.



### A. Windows의 이벤트 로그가 저장되는 경로

```
Do you want to monitor any customized log files?
1. yes
2. no
default choice: [1]:

Log file path:
C:\Windows\System32\winevt\Logs\System ← 경로 + System 입력
Log group name:
default choice: [System]

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:
C:\Windows\System32\winevt\Logs\Application ← 경로 + Application 입력
Log group name:
default choice: [Application]

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:
C:\Windows\System32\winevt\Logs\Security ← 경로 + Security 입력
Log group name:
default choice: [Security]

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
```

```
Do you want to monitor any Windows event log?
1. yes
2. no
default choice: [1]:

Windows event log name:
default choice: [System] ← System (default)
```

```
Log group name:
default choice: [System]

Log stream name:
default choice: [{instance_id}]

In which format do you want to store windows event to CloudWatch Logs?
1. XML: XML format in Windows Event Viewer
2. Plain Text: Legacy CloudWatch Windows Agent (SSM Plugin) Format
default choice: [1]:

Do you want to specify any additional Windows event log to monitor?
1. yes
2. no
default choice: [1]:
```

```

Windows event log name:
default choice: [System]
Application ← Application 입력
Do you want to monitor VERBOSE level events for Windows event log Application ?
1. yes
2. no VERBOSE, INFORMATION, WARNING, ERROR, CRITICAL level 모두 수집 O
default choice: [1]:

Do you want to monitor INFORMATION level events for Windows event log Application ?
1. yes
2. no
default choice: [1]:

```

여기서 VERBOSE, INFORMATION, WARNING, ERROR, CRITICAL 은 이벤트 로그의 위험도를 의미한다.

```

Log group name:
default choice: [Application]

Log stream name:
default choice: [{instance_id}]

In which format do you want to store windows event to CloudWatch Logs?
1. XML: XML format in Windows Event Viewer
2. Plain Text: Legacy CloudWatch Windows Agent (SSM Plugin) Format
default choice: [1]:

Do you want to specify any additional Windows event log to monitor?
1. yes
2. no
default choice: [1]:

```

```

Windows event log name:
default choice: [System]
Security ← Security 입력
Do you want to monitor VERBOSE level events for Windows event log Security ?
1. yes
2. no
default choice: [1]:

Do you want to monitor INFORMATION level events for Windows event log Security ?
1. yes
2. no
default choice: [1]:

Do you want to specify any additional Windows event log to monitor?
1. yes
2. no
default choice: [1]:
2
Saved config file to config.json successfully.
Current config as follows:

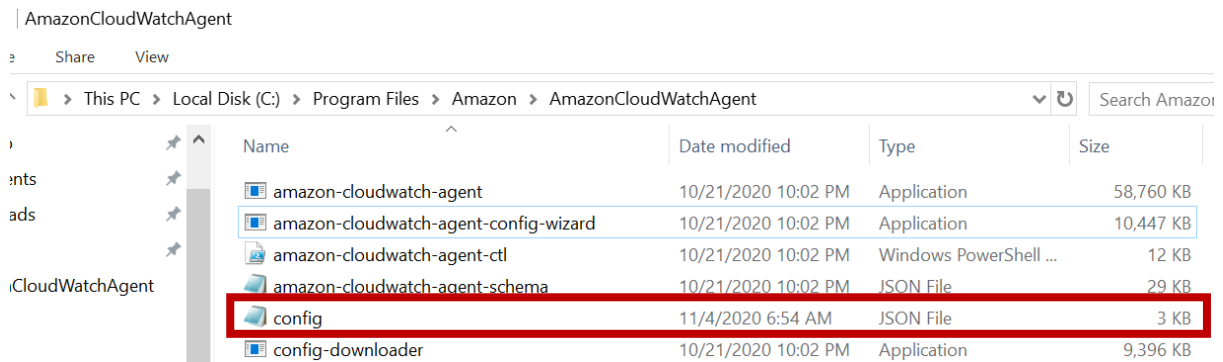
```

```

Please check the above content of the config.
The config file is also located at 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]:
2
Please press Enter to exit...

```

cmd 창을 종료하면 아래와 같이 config.json 파일이 생성된 것을 볼 수 있다.



## 2.5. Powershell을 이용한 CloudWatch agent 실행

- Powershell을 이용하여, AmazonCloudWatchAgent의 경로에서 다음 명령어를 입력해준다.

```
.\amazon-cloudwatch-agent-ctl.ps1 -a fetch-config -m ec2 file:.\config.json -s
```

위 명령어는 config 파일의 내용을 반영한 후 바로 동작을 수행하는 명령이다.

```

PS C:\Program Files\Amazon\AmazonCloudWatchAgent> .\amazon-cloudwatch-agent-ctl.ps1 -a fetch-config -m ec2 file:.\config.json -s
Successfully fetched the config and saved in C:\ProgramData\Amazon\AmazonCloudWatchAgent\Configs\file_config.json.tmp
Start configuration validation...
2020/11/02 07:45:22 Reading json config file path: C:\ProgramData\Amazon\AmazonCloudWatchAgent\Configs\file_config.json.tmp
Valid json input schema.
No csm configuration found.
Configuration validation first phase succeeded
Configuration validation second phase succeeded
Configuration validation succeeded

```

- 이후 현재 정상적으로 동작 중인지, 멈추어 있는지 상태를 확인하기 위해서는 다음 명령어를 입력한다.

```
.\amazon-cloudwatch-agent-ctl.ps1 -m ec2 -a status
```

```

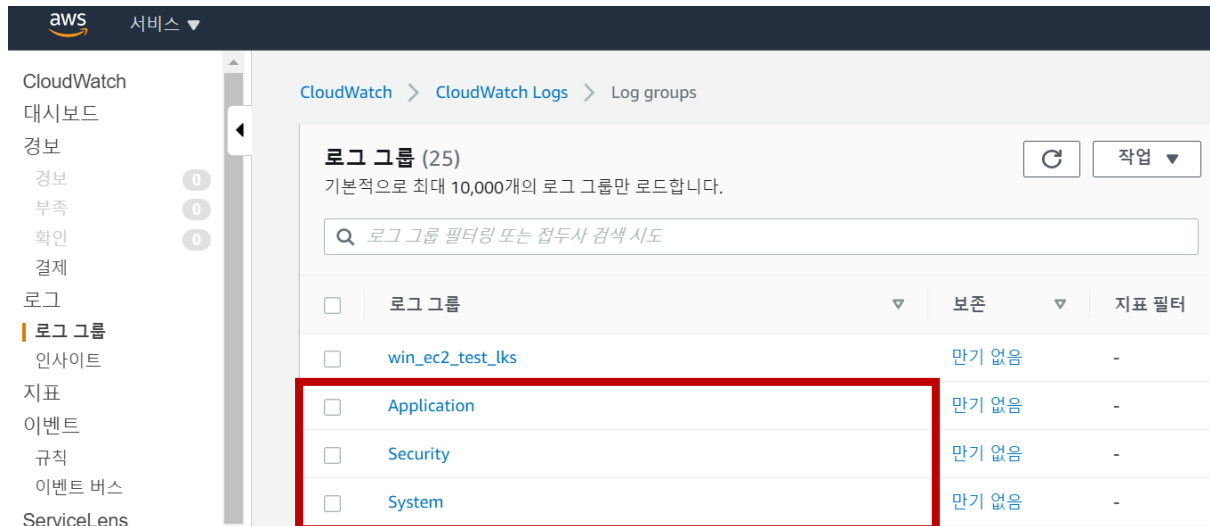
PS C:\Program Files\Amazon\AmazonCloudWatchAgent> .\amazon-cloudwatch-agent-ctl.ps1 -m ec2 -a status
{
  "status": "running",
  "starttime": "2020-11-02T07:45:23",
  "version": "1.247346.0b249609"
}

```

config 파일 생성과 Powershell 설정 이후부터 로그가 기록된다.

## 2.6. CloudWatch 로그 확인

- CloudWatch 서비스의 "로그 그룹"으로 이동하면, Windows 인스턴스 에이전트에서 생성한 로그 그룹들을 확인할 수 있다.



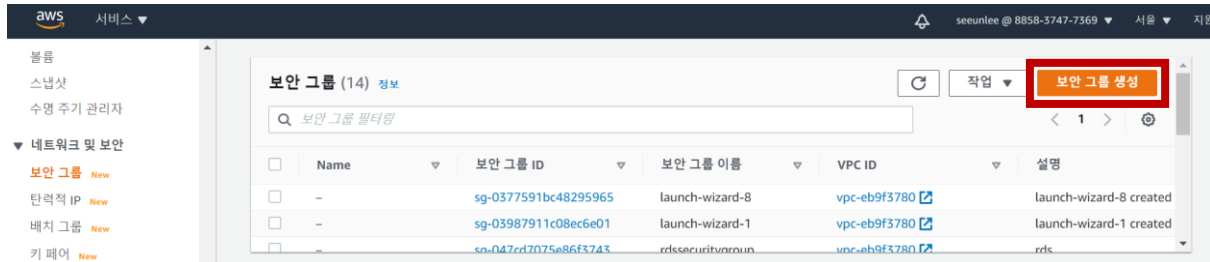
### 3. RDS 데이터베이스 로그 수집

#### 3.1. 보안 그룹 생성 및 RDS 데이터베이스 설정 수정

- 보안 그룹 생성

RDS 데이터베이스에 새로운 보안 그룹을 설정해주어야 한다.

EC2 - 네트워크 및 보안 - 보안 그룹으로 이동하여 “보안 그룹 생성” 버튼을 클릭한다.



- 인바운드 규칙으로 MySQL/Aurora, TCP 프로토콜, 3306 포트, 소스 유형을 설정한다.

EC2 > 보안 그룹 > 보안 그룹 생성

### 보안 그룹 생성 정보

보안 그룹은 인바운드 및 아웃바운드 트래픽을 관리하는 인스턴스의 가상 방화벽 역할을 합니다. 새 보안 그룹을 생성하려면 아래의 필드를 작성하십시오.

#### 기본 세부 정보

**보안 그룹 이름** 정보

newSecurityGroup

생성 후에는 이름을 편집할 수 없습니다.

**설명** 정보

for RDS logs

**VPC** 정보

vpc-eb9f3780

#### 인바운드 규칙 정보

**유형** 정보

MySQL/Aurora

**프로토콜** 정보

TCP

**포트 범위** 정보

3306

**소스** 정보

위치 무관

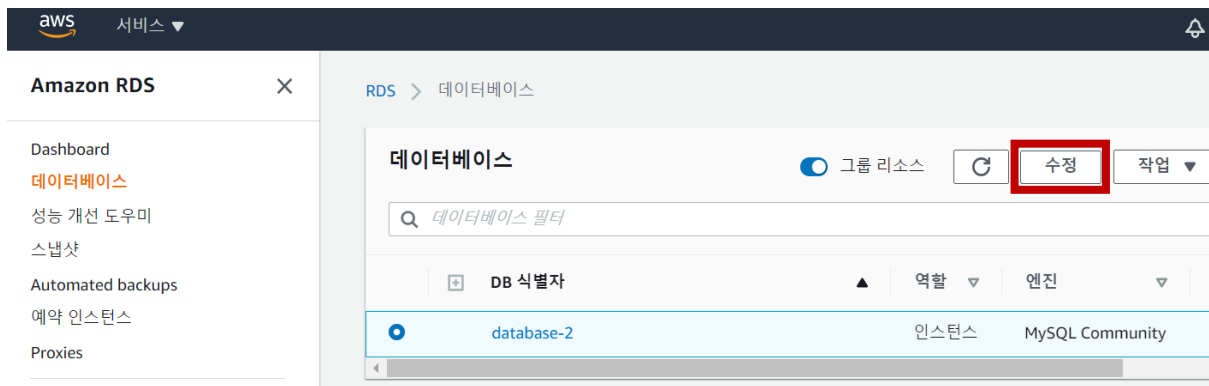
0.0.0.0/0

::/0

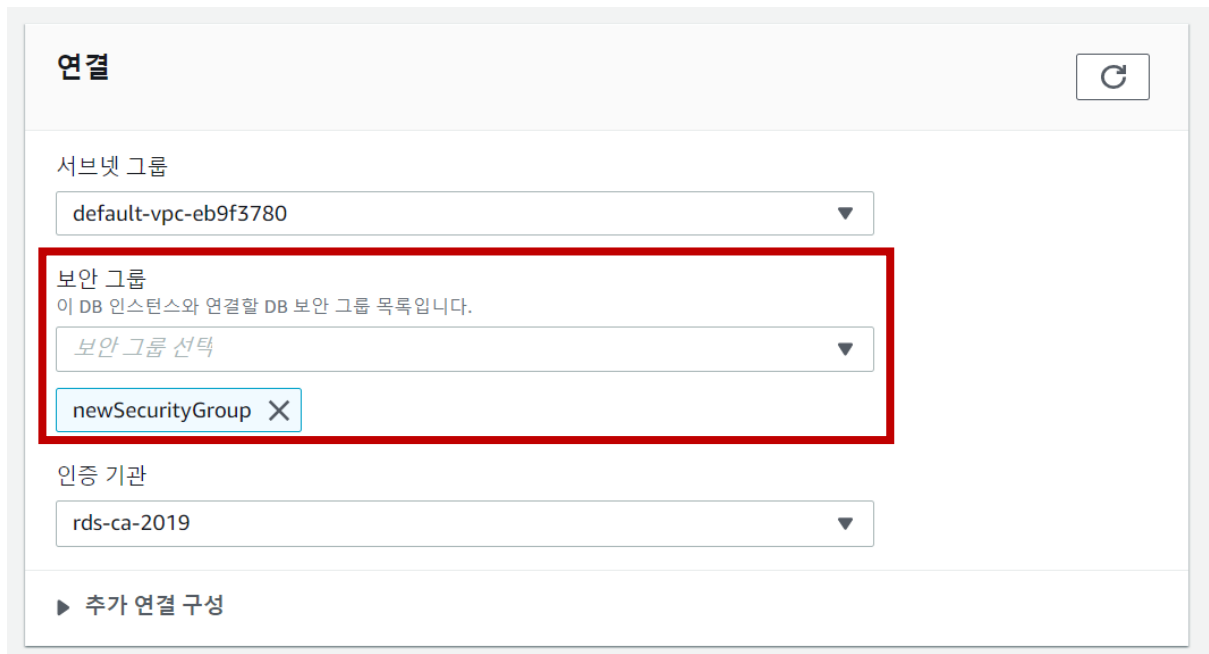
규칙 추가

그리고나서 보안 그룹을 생성을 마친다.

다시 RDS 데이터베이스로 돌아가 해당 데이터베이스를 선택한 후 "수정" 버튼을 클릭한다.



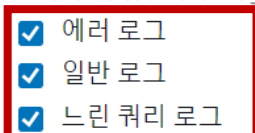
연결 - 보안 그룹 부분에서 생성한 보안 그룹을 선택한다.



추가 구성 - 로그 내보내기 부분에서는 에러 로그, 일반 로그, 느린 쿼리 로그 모두를 선택한 후, 수정을 마친다.

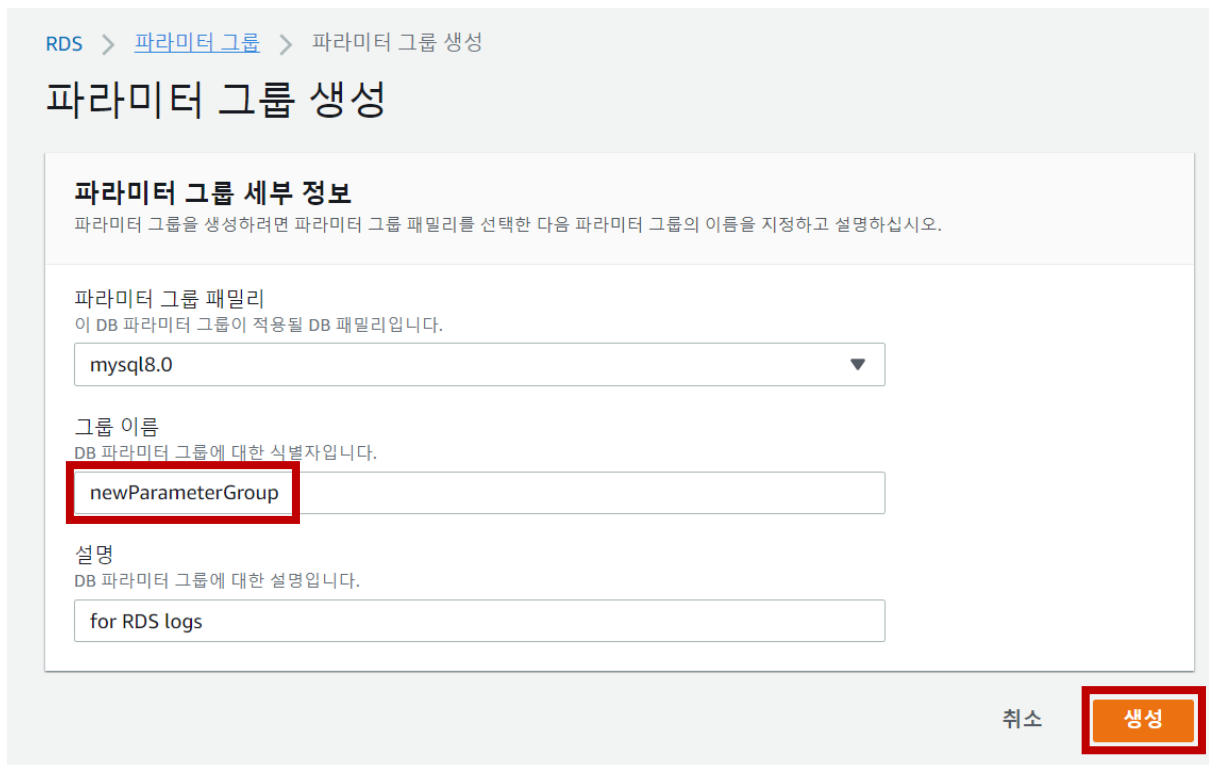
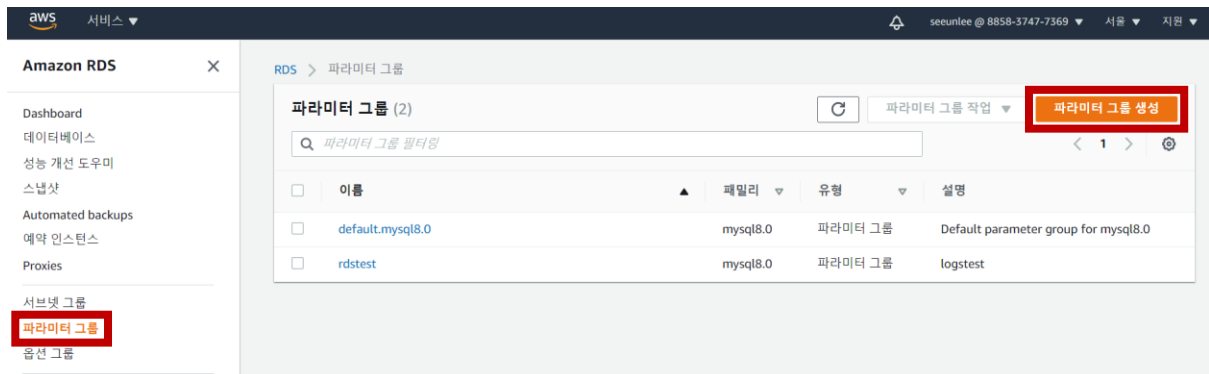
### 로그 내보내기

Amazon CloudWatch Logs로 게시할 로그 유형 선택

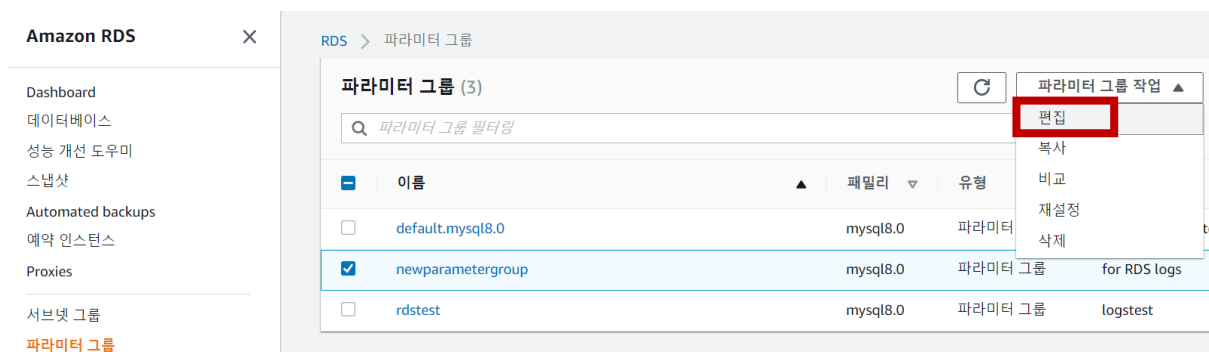


### 3.2. 파라미터 그룹 생성 및 RDS 데이터베이스 설정 수정

- RDS - 파라미터 그룹으로 이동하여 "파라미터 그룹 생성" 버튼을 클릭한다.

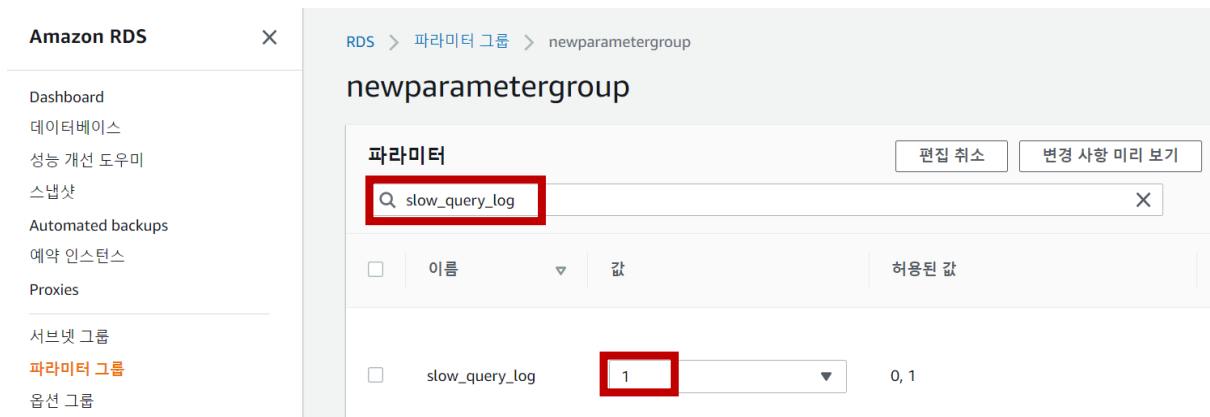
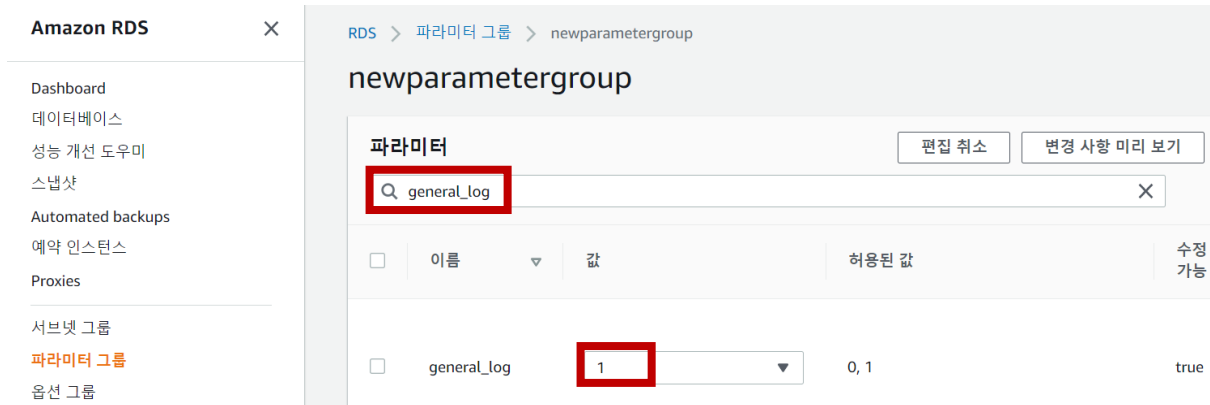


파라미터 그룹을 생성한 뒤, 해당 파라미터 그룹을 선택하여 파라미터 그룹 작업 중 "편집" 버튼을 클릭한다.

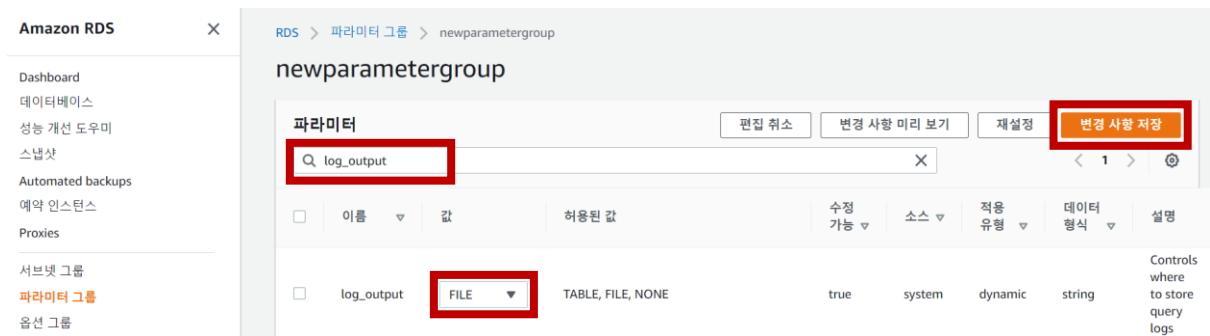




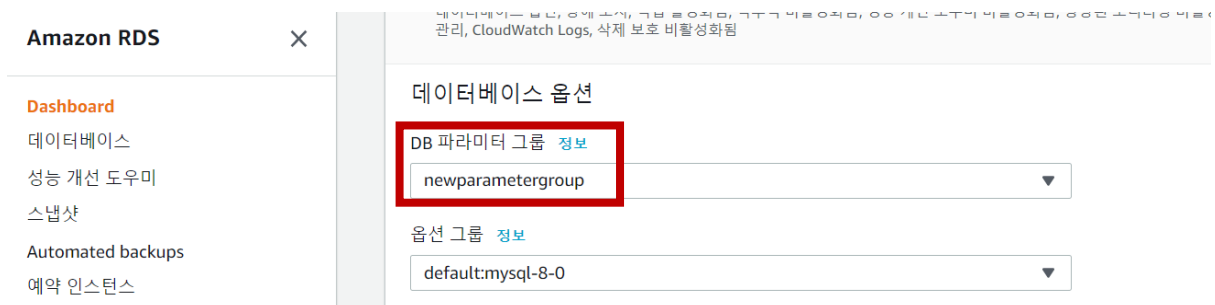
검색란에 general\_log 를 입력하여 값을 1 로 설정한다. slow\_query\_log 의 값도 1 로 설정한다.



마지막으로 log\_output 의 값을 FILE 로 설정한 후, "변경 사항 저장" 버튼을 클릭한다.

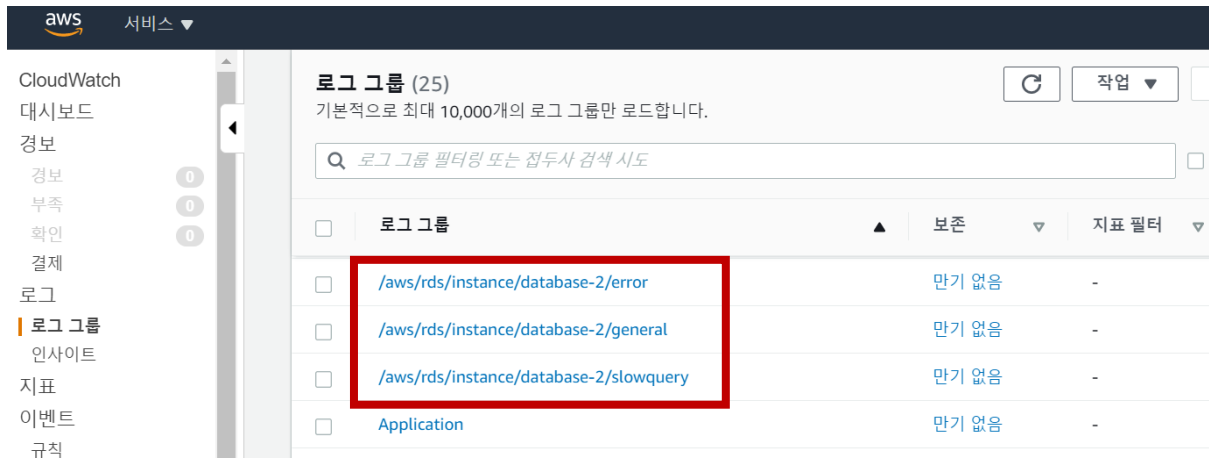


다시 RDS 데이터베이스로 이동하여 DB 파라미터 그룹을 해당 파라미터 그룹으로 설정한다.



### 3.3. CloudWatch 로그 확인

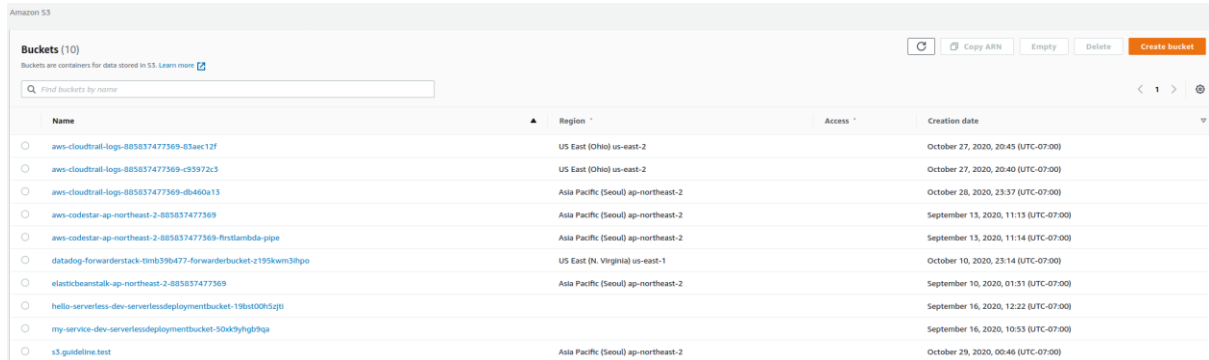
- CloudWatch의 로그 그룹 페이지에서 RDS 데이터베이스의 error, general, slowquery 로그를 확인할 수 있다.



## 4. S3 로그 수집

### 4.1. 로그를 수집하고자 하는 S3 버킷 선택

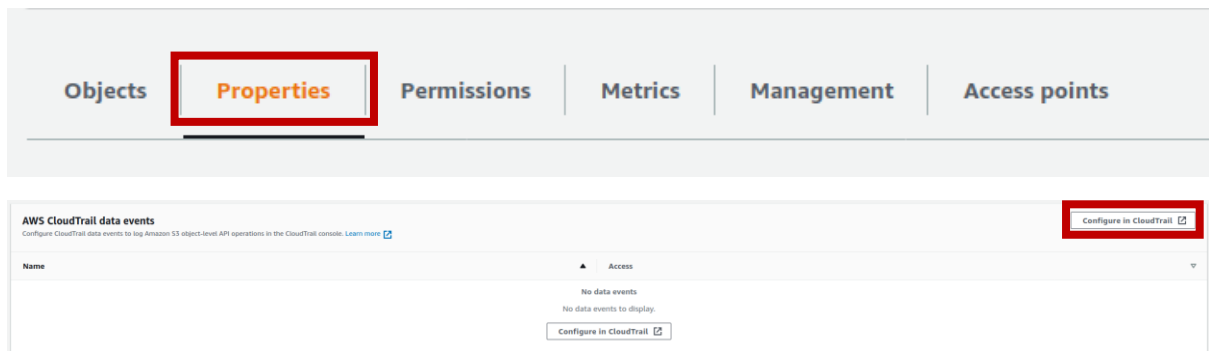
- 로그를 수집하고자 하는 버킷을 선택한다.



The screenshot shows the Amazon S3 console with a list of 10 buckets. The 'Properties' tab is selected, and the 'Configure in CloudTrail' button is highlighted in the top right corner.

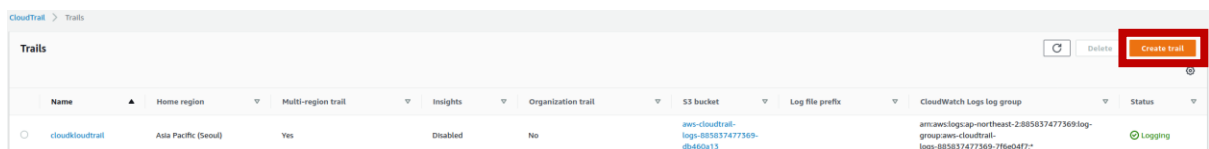
Name	Region	Access	Creation date
aws-cloudtrail-log-885837477369-83aec12f	US East (Ohio) us-east-2		October 27, 2020, 20:45 (UTC-07:00)
aws-cloudtrail-log-885837477369-c93972c3	US East (Ohio) us-east-2		October 27, 2020, 20:40 (UTC-07:00)
aws-cloudtrail-log-885837477369-d8460a13	Asia Pacific (Seoul) ap-northeast-2		October 26, 2020, 23:37 (UTC-07:00)
aws-codestar-ap-northeast-2-885837477369	Asia Pacific (Seoul) ap-northeast-2		September 13, 2020, 11:13 (UTC-07:00)
aws-codestar-ap-northeast-2-885837477369-firstlambda-pipe	Asia Pacific (Seoul) ap-northeast-2		September 13, 2020, 11:14 (UTC-07:00)
datadog-forwarderstack-timb39b477-forwarderbucket-z195kwn3lhp	US East (N. Virginia) us-east-1		October 10, 2020, 23:14 (UTC-07:00)
elasticbeanstalk-ap-northeast-2-885837477369	Asia Pacific (Seoul) ap-northeast-2		September 10, 2020, 01:51 (UTC-07:00)
hello-serverless-dev-serverlessdeploymentbucket-19bst00h5jti			September 16, 2020, 12:22 (UTC-07:00)
my-service-dev-serverlessdeploymentbucket-50dk9yhg8qga			September 16, 2020, 10:53 (UTC-07:00)
s3.guideline.test	Asia Pacific (Seoul) ap-northeast-2		October 29, 2020, 00:46 (UTC-07:00)

- 버킷을 선택한 후, 해당 버킷의 Properties 확인 시 AWS CloudTrail data events 탭을 확인할 수 있다. 여기서 "Configure in CloudTrail"을 클릭한다.



### 4.2. CloudTrail 생성

- CloudTrail을 생성하기 위해 "Create trail" 버튼을 클릭한다.



The screenshot shows the AWS CloudTrail console with a list of trails. The 'Create trail' button is highlighted in the top right corner.

Name	Home region	Multi-region trail	Insights	Organization trail	S3 bucket	Log file prefix	CloudWatch Logs log group	Status
cloudcloudtrail	Asia Pacific (Seoul)	Yes	Disabled	No	aws-cloudtrail-log-885837477369-d8460a13		arn:aws:logs:ap-northeast-2:885837477369:log-group:aws-cloudtrail-log-885837477369-7f0e047f*	Logging

S3 로그를 저장할 버킷을 새로 생성하여 저장하거나, 기존의 버킷에 저장할 수 있다. "추적 로그 버킷 및 폴더"의 값이 S3 버킷으로 생성되어 로그가 저장된다.

스토리지 위치 [Info](#)

☒ 새 S3 버킷 생성  
추적에 대한 로그를 저장할 버킷을 생성합니다.

☐ 기존 S3 버킷 사용  
이 추적에 대한 로그를 저장할 기존 버킷을 선택합니다.

추적 로그 버킷 및 폴더

로그를 저장할 새 S3 버킷 이름 및 폴더(접두사)를 입력합니다. 버킷 이름은 전역적으로 고유해야 합니다.

aws-cloudtrail-logs-885837477369-4329d002

로그는 aws-cloudtrail-logs-885837477369-4329d002/AWSLogs/885837477369

- CloudWatch Logs - optional 부분에서 아래와 같이 "CloudWatch Logs" 항목에 "Enabled"로 설정해준다.
- Log group은 새로 생성하거나 기존 그룹을 선택하여 사용할 수 있다. CloudWatch에서 해당 로그 그룹으로 들어가면 S3 로그를 확인할 수 있다.

### CloudWatch Logs - optional

Configure CloudWatch Logs to monitor your trail logs and notify you when specific activity occurs. Standard CloudWatch and CloudWatch Logs charges apply. [Learn more](#)

CloudWatch Logs [Info](#)

☒ Enabled

Log group [Info](#)

☒ New  
☐ Existing

Log group name

aws-cloudtrail-logs-885837477369-60089017

1-512 characters. Only letters, numbers, dashes, underscores, forward slashes, and periods are allowed.

- 로그 이벤트 선택

로그 이벤트로는 Management, Data, Insights 를 선택할 수 있으며, S3 에서는 세 가지의 로그를 모두 수집할 것이다.

CloudTrail > Trails > Create trail

Step 1  
[Choose trail attributes](#)

Step 2  
**Choose log events**

Step 3  
[Review and create](#)

## Choose log events

Events [Info](#)

Record API activity for individual resources, or for all current and future resources in AWS account. [Additional charges apply](#)

Event type

Choose the type of events that you want to log.

☒ Management events

Capture management operations performed on your AWS resources.

☒ Data events


Log the resource operations performed on or within a resource.

☒ Insights events

Identify unusual activity, errors, or user behavior in your account.

## Management events [Info](#)

Management events show information about management operations performed on resources in your AWS account.

 Charges apply to log management events on this trail because you are logging at least one other copy of management events in your account.


### API activity

Choose the activities you want to log.

☒ Read ☒ Write

☐ Exclude AWS KMS events

## Data events [Info](#)

**Additional charges apply**  Data events show information about the resource operations performed on or within a resource.

### Data event: S3 [Info](#)

[Remove](#)

#### Data event source

Select source of data events to log

S3 ▼

#### S3 bucket

You can choose to log read and/or write events for all buckets. You can also choose individual buckets.

All current and future S3 buckets

☒ Read

☒ Write

#### Individual bucket selection

Choose [Browse](#) to select multiple buckets, then choose to log Read, Write or both event types on all selected buckets.

 bucket/prefix

[Browse](#)

☒ Read


☒ Write

[×](#)

[Add bucket](#)

[Add data event type](#)

## Insights events [Info](#)

**Additional charges apply**  Identify unusual activity, errors, or user behavior in your account.

 Insights enabled

Usage anomalies are logged

그리고 나서 CloudTrail 생성을 마친다.

### 4.3. CloudWatch 로그 확인

- 생성된 Trail은 CloudWatch Logs Enabled에서 설정했던 로그 그룹 이름으로, CloudWatch의 로그 그룹에서 확인할 수 있다.

The screenshot shows the AWS CloudWatch console. On the left is a navigation menu with options like '대시보드', '경보', '로그', and '로그 그룹'. The main area is titled '로그 그룹 (27)' and shows a list of log groups. The log group 'aws-cloudtrail-logs-885837477369-60089017' is highlighted with a red box.

로그 그룹	보존	지표 필터
aws-cloudtrail-logs-885837477369-test	만기 없음	-
aws-cloudtrail-logs-885837477369-7f6e04f7	만기 없음	-
aws-cloudtrail-logs-885837477369-60089017	만기 없음	-

The screenshot shows the AWS CloudWatch console 'Log events' page. The page displays a list of log events with timestamps and JSON messages. The first event is highlighted.

타임스탬프	메시지
2020-11-05T15:14:28.741+09:00	{\"eventVersion\":\"1.07\",\"userIdentity\":{\"type\":\"AWSService\",\"invokedBy\":\"cloudtrail.amazonaws.com\"},\"eventTime\":\"202...
2020-11-05T15:14:28.741+09:00	{\"eventVersion\":\"1.07\",\"userIdentity\":{\"type\":\"AWSService\",\"invokedBy\":\"cloudtrail.amazonaws.com\"},\"eventTime\":\"202...
2020-11-05T15:17:59.515+09:00	{\"eventVersion\":\"1.05\",\"userIdentity\":{\"type\":\"AWSService\",\"invokedBy\":\"cloudtrail.amazonaws.com\"},\"eventTime\":\"202...
2020-11-05T15:17:59.515+09:00	{\"eventVersion\":\"1.05\",\"userIdentity\":{\"type\":\"AWSService\",\"invokedBy\":\"cloudtrail.amazonaws.com\"},\"eventTime\":\"202...
2020-11-05T15:17:59.515+09:00	{\"eventVersion\":\"1.05\",\"userIdentity\":{\"type\":\"AWSService\",\"invokedBy\":\"cloudtrail.amazonaws.com\"},\"eventTime\":\"202...
2020-11-05T15:18:53.763+09:00	{\"eventVersion\":\"1.07\",\"userIdentity\":{\"type\":\"AWSService\",\"invokedBy\":\"cloudtrail.amazonaws.com\"},\"eventTime\":\"202...
2020-11-05T15:18:53.763+09:00	{\"eventVersion\":\"1.07\",\"userIdentity\":{\"type\":\"AWSService\",\"invokedBy\":\"cloudtrail.amazonaws.com\"},\"eventTime\":\"202...
2020-11-05T15:18:53.763+09:00	{\"eventVersion\":\"1.07\",\"userIdentity\":{\"type\":\"AWSService\",\"invokedBy\":\"cloudtrail.amazonaws.com\"},\"eventTime\":\"202...
2020-11-05T15:18:53.763+09:00	{\"eventVersion\":\"1.07\",\"userIdentity\":{\"type\":\"AWSService\",\"invokedBy\":\"cloudtrail.amazonaws.com\"},\"eventTime\":\"202...