



Gaming on AWS 2017 Hands-on Labs

Gaming Data Lake on AWS

24-Oct-2017

Table of Contents

Table of Contents

Table of Contents	2
랩 소개	5
Data Lake 의 장점	5
준비 사항	5
1 단계: AWS 에 가입	6
2 단계: IAM 사용자 생성	6
3 단계: EC2 Instance 생성 및 접근	7
Lab 1. 로그 데이터 생성 및 Kinesis 로 저장	8
AWS Command Line Tool 설정	8
실습 코드 다운로드	8
AWS S3 버킷 생성	9
Amazon Kinesis Stream 서비스 생성	10
로그 데이터 생성 및 Kinesis 로 전송	12
Kinesis stream 에 입력된 데이터 확인하기	13
Lab 2. Amazon Redshift 로 Dataware housing 구성	15
Kinesis stream 에 저장된 데이터 Amazon S3 에 저장	15
Amazon Redshift 생성하기	17

Redshift table 에 데이터 적재	22
Lab 3. Amazon QuickSight 를 이용한 BI 시각화	25
Lab 4. Amazon Elasticsearch 와 Lambda 를 통한 실시간 검색엔진	32
Amazon Elasticsearch 생성	32
1 단계: Define domain	32
2 단계: Configure cluster	33
3 단계: Set up access	33
4 단계: Review	35
실시간 데이터 로딩을 위한 Lambda 함수 생성	36
Lambda 함수 생성	36
Lambda 함수 코드 수정 및 업로드	37
Lambda 함수 트리거 설정	39
Kibana 로 데이터 확인	41
Lab 5. Amazon EMR (Elastic MapReduce) 에서 S3 데이터 직접 쿼리	45
데이터 파일에 직접 Table 생성	45
Amazon EMR 클러스터 생성	45
Hive 를 이용하여 S3 에 직접 쿼리하기	47
Lab 6. Amazon Athena 를 이용하여 직접 S3 데이터 쿼리	52
JSON 에서 Parquet 형 변환	52
EMR Cluster Hive 를 활용 형 변환	52

Amazon Athena 생성	54
데이터 포맷에 따른 성능 비교	57
Lab 8. Amazon Kinesis Analytics 를 이용한 실시간 데이터 처리	59
Kinesis Analytics application 생성하기	59
실시간 쿼리 실행	61

랩 소개

Gaming Data Lake on AWS 랩은 다양한 AWS 서비스를 이용하여 데이터 분석을 다양한 예제를 통해 학습할 수 있도록 합니다. 데이터 생성, 저장, 처리, 그리고 시각화 하는 과정을 AWS 서비스들을 이용하여 쉽게 구성하며 빠르게 데이터 분석을 위한 시스템을 디자인 할 수 있도록 합니다.

Data Lake 의 장점

다양한 데이터 소스가 분리되어 생성되고 생성된 데이터가 흩어져 효율적인 접근이 어려운 경우, 하나의 단일 저장소에 데이터를 모아 쉽게 접근하고 저장할 필요가 있습니다. AWS Simple storage service 는 안전하고 확장성 있는 스토리지를 제공하며, 접근을 제어할 포함하여 다양한 AWS 서비스에서 데이터에 접근하여 각기 다른 목표의 분석이 가능합니다. Data Lake 개념으로 데이터 저장을 단일화 단순화 할 수 있도록 합니다.

만약 다양한 데이터 소스가 존재한다면 각각의 스키마가 다르며, 접근 방식이 다를 수 있어, 데이터를 가져오는데 (Ingest) 어려움이 있습니다. Data Lake 개념은 미리 정의된 스키마없이 쉽게 데이터를 가져올 수 있도록 합니다.

데이터의 양이 빠르게 증가함에 따라 확장성에 대한 고려가 필요합니다. 데이터 저장에 대한 스토리지와 연산을 위한 컴퓨팅을 분리함으로써 데이터 저장과 처리를 진행하며 동시에 확장 가능한 아키텍처를 쉽게 구현할 수 있도록 합니다. Data Lake 개념은 그러한 목적에 부합할 수 있는 확장성을 지원합니다.

비즈니스의 요건이 다양해지면서 같은 데이터 소스에 대해서는 보고자 하는 방법이나 분석 데이터 결과가 다를 수 있습니다. 데이터 분석 시스템은 이러한 요건을 쉽게 부합할 수 있는 시스템이 필요합니다. 또한 분석 역시 상시적이지 않고 필요에 의해서 잠시 사용되는 (Ad-hoc) 요건이 많아지게 됩니다. 동일 데이터 소스라도 손쉽게 Schema 를 적용하여 원하는 데이터를 추출하여 활용할 수 있는 부분도 Data Lake 개념의 장점입니다.

준비 사항

기본적으로 이번 랩은 AWS 사용 경험이 있음을 가정합니다. 또한 AWS 에 계정 (Account) 을 생성하고 바로 AWS 서비스를 사용할 수 있어야 합니다. AWS EC2 인스턴스에 접근하여 Bash shell 에서 간단한 작업을 수행할 수 있어야 합니다. 만약의 경우 아직 AWS 에 가입을 하지 않았거나, EC2 에 대한 접근에 대하여 가이드가 필요한 경우는 아래 내용을 참조하여 주십시오.

1 단계: AWS 에 가입

Amazon Web Services(AWS)에 가입하면 AWS 의 모든 서비스에 AWS 계정이 자동으로 등록됩니다. 사용한 서비스에 대해서만 청구됩니다.

AWS 는 사용한 리소스에 대해서만 비용을 지불합니다. AWS 를 처음 사용하는 고객인 경우 주요 서비스들을 무료로 시작할 수 있습니다. 자세한 내용은 [AWS 프리 티어](#)를 참조하십시오.

이미 AWS 계정이 있다면 다음 단계로 건너뛰십시오. AWS 계정이 없는 경우에는 다음 절차의 단계를 수행하여 계정을 만듭니다.

To create an AWS account

1. <https://aws.amazon.com/>을 열고 [Create an AWS Account]를 선택합니다.
2. 온라인 지시 사항을 따릅니다.

등록 절차 중 전화를 받고 전화 키패드를 사용하여 PIN 을 입력하는 과정이 있습니다.

다음 단계에 필요하므로 AWS 계정 ID 를 기록합니다.

2 단계: IAM 사용자 생성

AWS 서비스에 액세스하려면 해당 서비스가 소유한 리소스에 대한 액세스 권한이 있는지 확인하기 위해 자격 증명을 제공해야 합니다. 콘솔은 암호를 요구합니다. AWS 계정에 대한 액세스 키를 생성하면 AWS CLI 또는 API 에 액세스할 수 있습니다. 그러나 AWS 계정용 자격 증명을 사용하여 AWS 에 액세스하지 않는 것이 좋습니다. 대신 AWS Identity and Access Management(IAM)을 사용하는 것이 좋습니다. 관리자 권한을 사용하여 IAM 사용자를 생성하고 IAM 그룹에 추가한 다음 생성한 IAM 사용자에게 관리자 권한을 부여합니다. 그러면 특정 URL 과 해당 IAM 사용자의 자격 증명을 사용하여 AWS 에 액세스할 수 있습니다.

AWS 에 가입했지만 IAM 사용자를 만들지 않은 경우, IAM 콘솔을 사용하여 사용자를 만들 수 있습니다.

이 가이드는 관리자 권한이 있는 사용자(adminuser)가 있다고 가정합니다. 절차에 따라 계정에서 adminuser 를 만듭니다.

To create an administrator user and sign in to the console

1. Create an administrator user called adminuser in your AWS account. For instructions, see [Creating Your First IAM User and Administrators Group](#) in the *IAM 사용 설명서*.

2. A user can sign in to the AWS Management Console using a special URL. For more information, [How Users Sign In to Your Account](#) in the *IAM 사용 설명서*.

사용자 생성 후 Credential 정보 (Access Key ID, Secret access Key) 를 다운로드하여 **저장**합니다. (CSV file)

Add user



Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://806506827877.signin.aws.amazon.com/console>

Download .csv

	User	Access key ID	Secret access key	Email login instructions
▶	adminuser	AKIAI44QH8DHBVS7GAL3	***** Show	Send email ↗

중요

관리자 자격증명을 사용하는 프로덕션 애플리케이션을 구축하고 테스트할 때 제한된 권한이 있는 서비스 관련 관리자를 생성하는 것이 좋습니다.

IAM 에 대한 자세한 내용은 다음을 참조하십시오.

- [Identity and Access Management\(IAM\)](#)
- [시작하기](#)
- [IAM 사용 설명서](#)

3 단계: EC2 Instance 생성 및 접근

EC2 인스턴스 생성 및 접근 가이드는 아래 링크의 내용을 참조하여 주십시오. 사용하는 운영 체제에 따라서 가이드가 다르므로, 해당하는 가이드를 참고하여 주십시오.

아래 링크를 활용하여 Oregon Region 에 t2.micro 인스턴스를 생성합니다.

Amazon EC2 Linux 인스턴스 시작하기

http://docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/EC2_GetStarted.html

Amazon EC2 Windows 인스턴스 시작하기

http://docs.aws.amazon.com/ko_kr/AWSEC2/latest/WindowsGuide/EC2_GetStarted.html

Lab 1. 로그 데이터 생성 및 Kinesis 로 저장

위 준비 사항에서 생성한 EC2 instance 에 SSH 를 이용하여 접속합니다. SSH 접속 방법은 준비 사항의 인스턴스 시작하기를 참고합니다.

MAC 에서는 아래와 같이 접속 합니다.

```
$ ssh -i {다운로드한 key pair pem 파일} ec2-user@{EC2 인스턴스의 Public IP Address}
```

AWS Command Line Tool 설정

AWS 에서는 AWS CLI (Command Line Interface)를 제공하고 있습니다. Shell 에서 AWS 의 대부분의 리소스를 관리할 수 있습니다. Amazon Linux 에는 이미 AWS CLI 가 설치되어 있습니다. 준비 과정에서 만든 사용자인 adminuser 의 Credential 정보를 입력하면 해당 권한을 가지고 API 서비스를 관리할 수 있게 됩니다.

Shell 에서 아래 명령으로 Credential 설정이 가능합니다. IAM 에서 Adminuser 를 만들 때 저장한 Credential.csv 에 있는 키 값을 아래와 같이 입력합니다.

```
$ aws configure
AWS Access Key ID [None]: {Access Key ID}
AWS Secret Access Key [None]: {Secret access key}
Default region name [None]: us-west-2
```

```
$ aws s3 ls
```

설정이 잘 되었다면 위와 같이 실행하였을 때, 별 다른 에러 메시지가 발생하지 않아야 합니다.

실습 코드 다운로드

실습에서 사용할 코드를 다운로드하여 압축을 풀어 준비합니다. 코드는 Python 으로 작성되어 있으며, AWS S3 에 올려져 공유되어 있습니다.

```
$ mkdir lab;cd lab
```



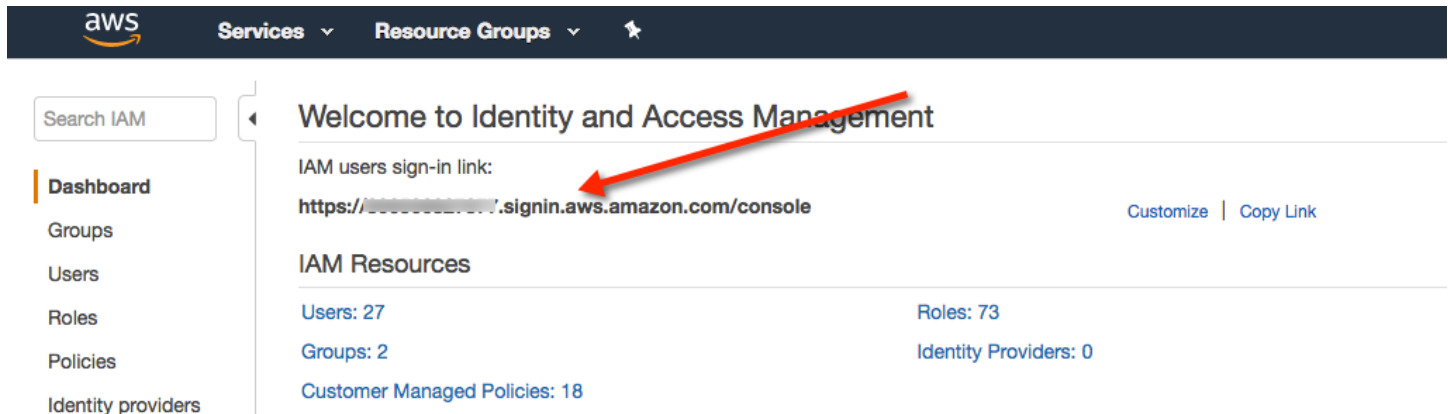
```
$ wget https://s3-ap-northeast-1.amazonaws.com/www.aws-korea.com/data_lake_labs.zip
```

```
$ unzip data_lake_labs.zip
```

AWS S3 버킷 생성

Data Lake 는 안전하고 확장성 있는 데이터 스토리지가 중심이 됩니다. 다양한 데이터 소스를 한 곳에 저장할 수 있으며, 데이터 양이 계속 늘어가거나 다양한 데이터 형식의 제약이 없는 안정적인 스토리지로 AWS S3 를 사용합니다. S3 bucket (저장소를 구분하기 위한 최상위 Prefix 이름) 을 생성하기 위해서 Console 에 접속합니다.

AWS Management console 에 adminuser 로 접속을 합니다. IAM 에서 생성한 사용자로 로그인할 경우는 IAM user sign-in URL 를 사용합니다. IAM user 로 로그인하기 위한 주소는 IAM 메뉴에서 확인할 수 있습니다. 아래 예를 참조하여 주십시오. (또는 앞에 저장한 Credential.csv 에서도 IAM 사용자의 로그인을 위한 주소를 확인 할 수 있습니다.)



위 주소에서 로그인에 성공하면 adminuser 는 모든 서비스의 권한을 가지고 있으므로 console 에서 모든 서비스를 관리할 수 있습니다.

서비스 중 AWS S3 로 이동하여 Create bucket 을 선택합니다. 유일한 이름으로 bucket 이름을 생성해야만 합니다. Region 은 Oregon 을 선택합니다. 원하는 bucket name 을 입력합니다. 아래 예는 labs-game-log 로 bucket 이름을 선택하여 생성합니다. 후에 다른 설정은 기본 설정으로 Next 를 선택합니다. 마지막으로 Create bucket 을 눌러 최종 Bucket 을 생성합니다.

Name and region

Bucket name ⓘ

labs-game-log

Region

US West (Oregon)

Copy settings from an existing bucket

 Amazon S3

+ Create bucket

Delete bucket

Empty bucket

Amazon Kinesis Stream 서비스 생성

Amazon Kinesis 는 스트리밍 데이터를 쉽게 수집할 수 있는 서비스입니다. 대용량의 스트리밍 데이터를 안전하게 저장할 수 있으며, 데이터 분석 플랫폼에서 데이터를 수집하는 핵심 기능을 담당할 수 있습니다. 위에 생성한 Amazon S3 Bucket 에 데이터를 저장하기 전에 Amazon Kinesis 에 데이터를 수집하고 다양한 서비스를 백엔드에 구성하는 것이 가능합니다. 이 랩에서는 Amazon Kinesis 에 데이터를 저장하고 저장된 데이터를 Amazon S3 에 저장하는 방식을 배워봅니다.

웹 콘솔에서 Amazon Kinesis 로 이동합니다. Streams, Firehose, Analytics 의 세 가지 서비스가 있으나, 우선은 Streams 를 선택합니다.

Create Kinesis stream 을 선택하여 사용할 Kinesis stream 을 생성합니다. Kinesis stream name 에 원하는 Kinesis stream 이름을 입력합니다. 아래 예에서는 labs-game-stream 을 이름으로 선택하였습니다.

aws

Services

Resource Groups

Amazon Kinesis

Streams

Firehose

Analytics

Create Kinesis stream

Kinesis stream name*

Shards

A shard is a unit of throughput capacity. Each shard ingests up to 1MB/sec and 1000 records/sec, and emits up to 2MB/sec. To a for higher or lower throughput, the number of shards can be modified after the Kinesis stream is created using the API. [Learn mor](#)

```

graph LR
    subgraph Producers
        P1[ ]
        P2[ ]
        P3[ ]
    end
    subgraph Kinesis_stream [Kinesis stream]
        subgraph Shards
            S1[Shard]
            S2[Shard]
        end
    end
    subgraph Consumers
        C1[ ]
        C2[ ]
        C3[ ]
    end
    Producers --> Kinesis_stream
    Kinesis_stream --> Consumers
  
```

Number of shards 는 랩이므로 우선 1 개로 선택합니다. Create Kinesis stream 버튼을 눌러 생성을 마무리 합니다.

aws

Services

Resource Groups

Amazon Kinesis

Streams

Firehose

Analytics

Kinesis streams

A Kinesis stream is an ordered sequence of data records. To add data to a Kinesis stream, configure producers using the Streams PU

Total shards in use: 2 Total shards remaining: 498 ⓘ

Create Kinesis stream

Connect to Firehose ⓘ

Actions ▾

	Kinesis stream name	Number of shards
<input checked="" type="checkbox"/>	labs-game-stream	1

로그 데이터 생성 및 Kinesis 로 전송

실습 코드 중에 simulator.py 파일은 random 으로 게임 관련 로그를 생성하여 Kinesis stream 으로 전송하는 예제 코드입니다. 앞에 생성된 Kinesis stream 정보를 입력해야 어떠한 Kinesis stream 으로 데이터를 보낼 지 알 수 있으므로, config.json 파일을 수정합니다.

Vi 와 같은 편집기를 열어 아래 파일의 Kinesis 항목의 속성 중 stream_name 을 앞에서 생성한 Kinesis stream 이름을 입력하고 저장합니다.

config.json 파일 내용

```
{
  "aws": {
    "access_key": "YOUR_ACCESS_KEY",
    "secret_key": "YOUR_SECRET_KEY"
  },
  "kinesis": {
    "stream_name": "YOUR_KINESIS_STREAM_NAME",
    "region": "us-west-2"
  },
  "s3": {
    "bucket_name": "YOUR_S3_BUCKET_NAME",
    "region": "us-west-2"
  },
  "redshift": {
    "host": "YOUR_REDSHIFT_HOST",
    "port": 5439,
    "dbname": "YOUR_DB_NAME",
    "user": "YOUR_DB_USERNAME",
    "password": "YOUR_DB_PASSWORD"
  },
  "log": {
    "dir": "./logs/",
    "prefix": "gamelog_",
    "extension": ".dat"
  },
  "db": {
    "log_table": "log",
    "sum_table": "summary"
  }
}
```

simulator.py 를 실행하면 kills, deaths, is_winner, assists, char_class, char_name 의 정보를 임의로 생성하여 Kinesis stream 으로 데이터를 계속 전송하게 됩니다.

```
$ python simulator.py
```

```
.....
```

위에서 점이 지속적으로 출력된다면 데이터가 정상적으로 입력되고 있는 표시입니다.

Kinesis stream 에 입력된 데이터 확인하기

데이터가 실제 Kinesis stream 에 지속적으로 입력이 되고 있습니다만, 직접 입력된 데이터를 확인하기 위해서는 데이터를 직접 Kinesis stream 에서 읽어와야 합니다. Github 에는 kinesis-poster-worker 라는 간단한 툴을 제공하고 있습니다. 위의 툴을 가져와서 쉽게 데이터를 읽어 볼 수 있습니다.

Kinesis-poster-worker 위치: <https://github.com/awslabs/kinesis-poster-worker>

작업을 편하게 하기 위해 새로 SSH 터미널을 하나 더 연결합니다 (가능하시다면 screen 을 써도 좋습니다). Git 명령을 통해서 툴을 다운로드 받습니다. 도움말을 확인하고 간단히 아래와 같이 명령어를 실행하면 Kinesis stream 에 입력된 데이터를 읽어와 출력하게 됩니다. Simulator.py 가 실행 중인 동안에 실행해야 데이터를 확인할 수 있습니다.

```
$ git clone https://github.com/awslabs/kinesis-poster-worker.git
```

```
$ cd kinesis-poster-worker
```

```
$ python worker.py --help
```

```
$ python worker.py --region us-west-2 --echo labs-game-stream
```

```
.....
# Shard Count: 1
#-> shardId: shardId-0000000000000
#-> starting: shard_worker:0
+ KinesisWorker: shard_worker:0
+> working with iterator: LATEST
+> getting next records using iterator:
AAAAAAAAAAAHy6+oZfmdEAS/KLB6JyEqOrJunKEQDqBD4dM/m7sJxPxKW2NEusz3zZOA3n2Ga5Dsk19XZ6cXk
NzZ9XwNYwFjx4b4to3J5D3Aa0G0DyABCLdm4W7rQvLVVQULSBm1/J37bO1Tea6jK4AYueQTX666H6oZYt83Q7
pa1lf4TylkqpL4FkDDtgbJ1QNP+Q6eD2Xekzkw6AZ9jbo32LZsqKb/p
.....
+> shard_worker:0 Got 2 Worker Records
+--> echo record:
{"kills": 22, "deaths": 6, "is_winner": false, "assists": 16, "char_class": "Assassin", "char_name": "Roggtul"}
+--> echo record:
{"kills": 9, "deaths": 5, "is_winner": true, "assists": 10, "char_class": "Damager", "char_name": "Redthok"}

+> shard_worker:0 Got 1 Worker Records
+--> echo record:
{"kills": 17, "deaths": 26, "is_winner": false, "assists": 22, "char_class": "Wziard", "char_name": "Thehilda"}
..
+> shard_worker:0 Got 1 Worker Records
+--> echo record:
```

```
{ "kills": 15, "deaths": 18, "is_winner": true, "assists": 0, "char_class": "Wziard", "char_name": "Thehilda" }  
...  
+-> shard_worker:0 Got 1 Worker Records  
+--> echo record:  
{ "kills": 6, "deaths": 9, "is_winner": true, "assists": 9, "char_class": "Damager", "char_name": "Redthok" }
```

데이터가 입력된 것을 확인합니다.

Lab 2. Amazon Redshift 로 Dataware housing 구성

Lab 1 에서 게이밍 로그 데이터를 생성하여 안전하게 Amazon Kinesis stream 에 저장하였습니다. Amazon Kinesis 에 저장된 데이터는 일정 기간 동안 데이터가 저장되게 됩니다. 따라서, Kinesis stream 은 데이터 수집 역할이므로 데이터를 영속성 있는 스토리지가 데이터 베이스에 저장할 필요가 있습니다. Lab 2 에서는 Kinesis stream 에 저장된 데이터를 읽어와 앞에 생성한 Amazon S3 에 파일로 데이터를 모아 저장하고 Amazon 의 DW 서비스인 Redshift 로 데이터를 저장하는 실습을 합니다.

Kinesis stream 에 저장된 데이터 Amazon S3 에 저장

config.json 파일에 앞에서 생성한 S3 bucket 이름을 설정합니다. 파일을 열어 아래 bucket_name 에 앞에서 생성한 S3 bucket 이름을 입력하고 저장합니다. consumer.py 는 Kinesis stream 에서 데이터를 가져와 S3 에 업로드하는 역할을 합니다.

```
"s3": {
  "bucket_name": "YOUR_S3_BUCKET_NAME",
  "region": "us-west-2"
},
```

github 에 Kinesis client library 로 Python 을 쉽고 안전하게 실행할 수 있도록 도움을 주는 코드가 있어 활용합니다. 설정이 필요해 미리 해당 툴을 압축하여 준비하였습니다. 아래처럼 다운로드하여 압축을 풉니다.

참조: <https://github.com/aws-labs/amazon-kinesis-client-python>

```
$ wget https://s3-ap-northeast-1.amazonaws.com/www.aws-korea.com/amazon\_kcl.zip
```

```
$ unzip amazon_kcl.zip
```

실행을 위한 설정 파일을 수정합니다. consumer.properties 파일을 열어 아래 내용을 구성에 맞게 수정합니다. Consumer.py 위치를 입력하고, 생성한 Kinesis stream 이름을 입력합니다. 아래 예를 참고하여 본인의 구성에 맞게 설정합니다.

```
$ vi consumer.properties
```

```
executableName = /home/ec2-user/lab/consumer.py
```

```
streamName = labs-game-stream
```

```
regionName = us-west-2
```

...

Simulator.py 가 실행 되는 동안 지속적으로 데이터가 Kinesis stream 으로 저장되고 있습니다. 앞에서 simulator.py 실행을 중지하셨다면, 다시 해당 터미널에서 실행합니다. 실행의 편의를 위해 다른 터미널을 열어서 아래 명령으로 앞에 수정한 Kinesis client 를 실행합니다. 편의를 위해 run_consumer.sh 란 스크립트로 실행 명령을 수행합니다.

```
$ ./run_consumer.sh
```

```
[ec2-user@ip-172-16-0-124 lab]$ ./run_consumer.sh
```

```
Oct 11, 2017 1:34:42 AM com.amazonaws.services.kinesis.clientlibrary.config.KinesisClientLibConfigurator
getConfiguration
INFO: Value of workerId is not provided in the properties. WorkerId is automatically assigned as: c9db9d56-9885-4da0-
96ce-60555b406a68
Oct 11, 2017 1:34:42 AM com.amazonaws.services.kinesis.clientlibrary.config.KinesisClientLibConfigurator
withProperty
INFO: Successfully set property initialPositionInStream with value TRIM_HORIZON
Oct 11, 2017 1:34:43 AM com.amazonaws.services.kinesis.clientlibrary.config.KinesisClientLibConfigurator
withProperty
INFO: Successfully set property regionName with value us-west-2
Oct 11, 2017 1:34:43 AM com.amazonaws.services.kinesis.multilang.MultiLangDaemonConfig buildExecutorService
Oct 11, 2017 1:37:03 AM com.amazonaws.services.kinesis.multilang.MessageWriter call
INFO: Message size == 402 bytes for shard shardId-000000000000
Oct 11, 2017 1:37:03 AM com.amazonaws.services.kinesis.multilang.LineReaderTask call
INFO: Starting: Reading next message from STDIN for shardId-000000000000
Oct 11, 2017 1:37:03 AM com.amazonaws.services.kinesis.multilang.MultiLangProtocol validateStatusMessage
INFO: Received response {"action":"status","responseFor":"processRecords"} from subprocess while waiting for
processRecords while processing shard shardId-000000000000
Oct 11, 2017 1:37:04 AM com.amazonaws.services.kinesis.multilang.MessageWriter writeMessage
INFO: Writing ProcessRecordsMessage to child process for shard shardId-000000000000
Oct 11, 2017 1:37:04 AM com.amazonaws.services.kinesis.multilang.MessageWriter call
INFO: Message size == 403 bytes for shard shardId-000000000000
Oct 11, 2017 1:37:04 AM com.amazonaws.services.kinesis.multilang.LineReaderTask call
INFO: Starting: Reading next message from STDIN for shardId-000000000000
Oct 11, 2017 1:37:04 AM com.amazonaws.services.kinesis.multilang.MultiLangProtocol validateStatusMessage
INFO: Received response {"action":"status","responseFor":"processRecords"} from subprocess while waiting for
processRecords while processing shard shardId-000000000000
Oct 11, 2017 1:37:05 AM com.amazonaws.services.kinesis.multilang.MessageWriter writeMessage
INFO: Writing ProcessRecordsMessage to child process for shard shardId-000000000000
.....
```

수 분이 지나고 aws s3 ls 명령으로 아래와 같이 S3 bucket 에 데이터 파일이 저장되고 있는 것을 확인할 수 있습니다.

```
$ aws s3 ls s3://{YOUR_S3_BUCKET_NAME}
```

```
$ aws s3 ls s3://labs-game-log/
```



```

2017-10-11 01:37:00 41247 gamelog_20171011-0135.dat
2017-10-11 01:37:01 211 gamelog_20171011-0136.dat
2017-10-11 01:38:02 11703 gamelog_20171011-0137.dat
2017-10-11 01:39:02 11839 gamelog_20171011-0138.dat
2017-10-11 01:40:01 12286 gamelog_20171011-0139.dat
2017-10-11 01:41:02 11315 gamelog_20171011-0140.dat

```

Amazon Redshift 생성하기

Amazon Redshift 는 대용량의 데이터를 처리할 수 있는 Dataware housing 서비스입니다. Redshift 는 S3 에 저장된 데이터를 직접 DW 내의 Table 에 바로 Loading 할 수 있는 기능을 지원합니다. Inserter.py 코드는 Redshift 에 테이블을 만들고 S3 에서 파일을 다운로드하는 기능을 합니다.

AWS management console 에서 Redshift 서비스를 선택합니다. Launch Cluster 를 눌러 아래와 같이 입력하고 Redshift cluster 를 생성합니다. 이름과 패스워드는 원하는 대로 설정이 가능합니다. Database name, admin id, password 는 config.json 파일에 업데이트 해야 하므로 정확히 기록해 둡니다.

The screenshot shows the 'Cluster Details' step in the AWS Redshift console. The left sidebar lists navigation options: Redshift dashboard, Clusters, Snapshots, Security, Parameter groups, Workload management, Reserved nodes, Events, and Connect client. The main content area has a progress bar with four steps: CLUSTER DETAILS (active), NODE CONFIGURATION, ADDITIONAL CONFIGURATION, and REVIEW. Below the progress bar, a note states: 'Provide the details of your cluster. Fields marked with * are required.'

The form contains the following fields and descriptions:

- Cluster identifier***: labs-dw-instance. This is the unique key that identifies a cluster. This parameter is stored as a lowercase string. (e.g. my-dw-instance)
- Database name**: mydb. Optional. A default database named dev is created for the cluster. Optionally, specify a custom database name (e.g. mydb) to create an additional database.
- Database port***: 5439. Port number on which the database accepts connections.
- Master user name***: admin. Name of master user for your cluster. (e.g. awsuser)
- Master user password***: Password must contain 8 to 64 printable ASCII characters excluding: /, *, ', \, and @. It must contain 1 uppercase letter, 1 lowercase letter, and 1 number.
- Confirm password***: Confirm master user password

At the bottom, there are 'Cancel' and 'Continue' buttons. The 'Continue' button is highlighted in blue.

Continue 를 선택합니다.

CLUSTER DETAILS

NODE CONFIGURATION

ADDITIONAL CONFIGURATION

REVIEW

Choose a number of nodes and node type below. Number of Compute Nodes is required for multi-node clusters.

...

The ds2 node types replace the deprecated ds1 node types. The newer ds2 node types provide higher performance than ds1 at no extra cost. [Learn more.](#)

Node type

dc1.large

CPU

7 EC2 Compute Units (2 virtual cores) per node

Memory

15 GiB per node

Storage

160GB SSD storage per node

I/O performance

Moderate

Cluster type

Single Node

Number of compute nodes*

1

Maximum

1

Minimum

1

Specifies the compute, memory, storage, and I/O capacity of the cluster's nodes.

Single Node clusters consist of a single node which performs both leader and compute functions.

Cancel

Previous

Continue

기본 설정 그대로 진행합니다. Continue 를 누릅니다.

Provide the optional additional configuration details below.

Cluster parameter group default.redshift-1.0 ▾ Parameter group to associate with this cluster.

Encrypt database ☒ None ☐ KMS ☐ HSM [Learn more about database encryption](#)

Configure networking options:

Choose a VPC Default VPC (vpc-34d6375d) ▾ The identifier of the VPC in which you want to create your cluster

Cluster subnet group default ▾ Selected Cluster Subnet Group may limit the choice of Availability Zones

Publicly accessible ☒ Yes ☐ No Select Yes if you want the cluster to be accessible from the public internet. Select No if you want it to be accessible only from within your private VPC network

Choose a public IP address ☐ Yes ☒ No Select Yes if you want the cluster to have a public IP address that can be accessed from the public Internet, select No if you want the cluster to have a private IP addressed that can only be accessed from within the VPC.

Enhanced VPC Routing ☐ Yes ☒ No Select Yes if you want to enable Enhanced VPC Routing. [Learn more](#)

Availability zone No Preference ▾ The EC2 Availability Zone that the cluster will be created in.

Associate your cluster with one or more security groups.

VPC security groups default (sg-debf53b7) List of VPC security groups to associate with this cluster.

Optionally, create a basic alarm for this cluster.

Create CloudWatch Alarm ☐ Yes ☒ No Create a CloudWatch alarm to monitor the disk usage of your cluster.

기본 설정으로 두고 Continue 를 진행합니다. Launch cluster 를 눌러 생성을 진행합니다. Cluster 가 생성되는데 수 분이 소요됩니다. Cluster 가 active 상태로 완전히 생성된 것을 확인합니다.

클러스터의 상세 정보를 눌러 URL 을 확인합니다.

Clusters

Launch Cluster Manage Tags Manage IAM roles

Cluster	Cluster Status	DB Health	In Maintenance	Recent Events
labs-dw-instance	available	healthy	no	3

Endpoint labs-dw-instance.c91bwulbunqt.us-west-2.redshift.amazonaws.com:5439 (authorized) ⓘ

Cluster Properties		Cluster Status	
Cluster Name	labs-dw-instance	Cluster Status	available
Node Type	dc1.large	Database Health	healthy
Nodes	1	In Maintenance Mode	no
Zone	us-west-2a	Parameter Group Apply Status	in-sync
Cluster Parameter Group	default.redshift-1.0 (in-sync)	Pending Modified Values	None
Enhanced VPC Routing	No		

Config.json 파일을 열어 Redshift cluster(Endpoint = host, port, dbname, user id, password) 정보, 그리고 AWS credential (Access key ID, Secret access key) 를 입력합니다.

\$ vi config.json

```
"aws": {
  "access_key": "YOUR_ACCESS_KEY",
  "secret_key": "YOUR_SECRET_KEY"
},
"kinesis": {
  "stream_name": "labs-game-stream",
  "region": "us-west-2"
},
"s3": {
  "bucket_name": "labs-game-log",
  "region": "us-west-2"
},
"redshift": {
  "host": "YOUR_REDSHIFT_HOST",
  "port": 5439,
  "dbname": "YOUR_DB_NAME",
  "user": "YOUR_DB_USERNAME",
  "password": "YOUR_DB_PASSWORD"
},
```

Redshift cluster 에 정상적으로 접근이 가능한지 PostgreSQL Client tool 로 접속을 확인합니다.

참고: <http://docs.aws.amazon.com/redshift/latest/mgmt/connecting-from-psql.html>

툴을 직접 리눅스에 설치하기 위해서 아래와 같이 설치를 진행합니다.

```
$ sudo yum install postgresql8 -y
```

Cluster 에 접속하기 전에 Cluster 에 설정된 VPC security group 에서 방화벽 규칙에서 Redshift cluster 가 사용하는 Port 를 열어줘야만 합니다.

Cluster: labs-dw-instance

Cluster ▾ Database ▾ Backup ▾

Endpoint [labs-dw-instance.c91bwulbunqt.us-west-2.redshift.amazonaws.com:5439](#) (**authorized**) ⓘ

Cluster Properties

Cluster Name	labs-dw-instance
Cluster Type	Single Node
Node Type	dc1.large
Nodes	1
Zone	us-west-2a
Created Time	October 11, 2017 at 2:16:48 PM UTC+9
Cluster Version	1.0.1459
VPC ID	vpc-5c4ead39 (View VPCs)
Cluster Subnet Group	redshift-subnet
VPC security groups	default (sg-e103c584) (active)
Cluster Parameter Group	default.redshift-1.0 (in-sync)
Enhanced VPC Routing	No

Cluster Status

Cluster Status	available
Database Health	healthy
In Maintenance Mode	no
Parameter Group Apply Status	in-sync
Pending Modified Values	None

위 그림에서의 VPC Security group 을 선택하여 inbound 에서 Edit 를 눌러 아래와 같이 규칙을 추가합니다.
Source 는 편의를 위해 Anyway 를 선택합니다.

Security Group: sg-e103c584

Description Inbound Outbound Tags

Edit

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
Redshift	TCP	5439	0.0.0.0/0	
Redshift	TCP	5439	:::0	

다시 터미널에서 아래와 같이 수행하여 접근이 되는 것을 확인합니다.

```
$ psql -h <endpoint> -U <userid> -d <databasename> -p <port>
```

예를 들어 아래와 같이 입력하면 접속이 된다면 Password 를 물어봅니다. Password 를 정확히 입력하면 접속이 완료됩니다. (툴에서 나오는 법은 Ctrl+d)

```
$ psql -h labs-dw-instance.c91bwulbunqt.us-west-2.redshift.amazonaws.com -p 5439 -U admin -d mydb
```

Password for user admin:

psql (8.4.20, server 8.0.2)

WARNING: psql version 8.4, server version 8.0.

Some psql features might not work.

SSL connection (cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256)

Type "help" for help.

```
mydb=#
```

Redshift table 에 데이터 적재

Insertter.py 는 데이터를 적재하는 일을 합니다. psycopg2 - Python-PostgreSQL Database Adapter 가 필요하므로 아래와 같이 패키지를 설치합니다.

```
$ sudo pip install psycopg2
```

insertter.py 를 실행합니다. simulator.py 로 계속 데이터를 생성하고 ./run_consumer 를 실행하여 지속적으로 S3 에 파일이 쌓이는 것을 insertter.py 에서 Redshift cluster 의 log table 로 계속 적재하게됩니다. Web management console 에서 적재 명령이 잘 실행되는 것을 확인할 수 있습니다.

Redshift dashboard

Cluster: **labs-dw-instance** | Configuration | Status | Performance | Queries | **Loads** | Table restore

Terminate Load

Filter: Last 24 Hours | Search...

	Load	Run time	Start time	Status	Completion	Table	User	SQL
<input type="checkbox"/>	264	207ms	October 11, 2017 at 3:24:02 PM UTC+9	COMPLETED	100%	log	admin	copy log from 's3://labs-game-log/game-log_20171011-0623.dat' credentials '
<input type="checkbox"/>	259	202ms	October 11, 2017 at 3:23:02 PM UTC+9	COMPLETED	100%	log	admin	copy log from 's3://labs-game-log/game-log_20171011-0622.dat' credentials '
<input type="checkbox"/>	253	201ms	October 11, 2017 at 3:22:01 PM UTC+9	COMPLETED	100%	log	admin	copy log from 's3://labs-game-log/game-log_20171011-0621.dat' credentials '
<input type="checkbox"/>	249	178ms	October 11, 2017 at 3:21:01 PM UTC+9	COMPLETED	100%	log	admin	copy log from 's3://labs-game-log/game-log_20171011-0620.dat' credentials '
<input type="checkbox"/>	241	2.79s	October 11, 2017 at 3:20:23 PM UTC+9	COMPLETED	100%	log	admin	copy log from 's3://labs-game-log/game-log_20171011-0619.dat' credentials '

또는 psql 로 직접 cluster 에 접속하여 아래와 같이 확인할 수 있습니다.

```
$ psql -h labs-dw-instance.c91bwulbunqt.us-west-2.redshift.amazonaws.com -p 5439 -U admin -d mydb
```

Password for user admin:

psql (8.4.20, server 8.0.2)

WARNING: psql version 8.4, server version 8.0.

Some psql features might not work.

SSL connection (cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256)

Type "help" for help.

```
mydb=#
```

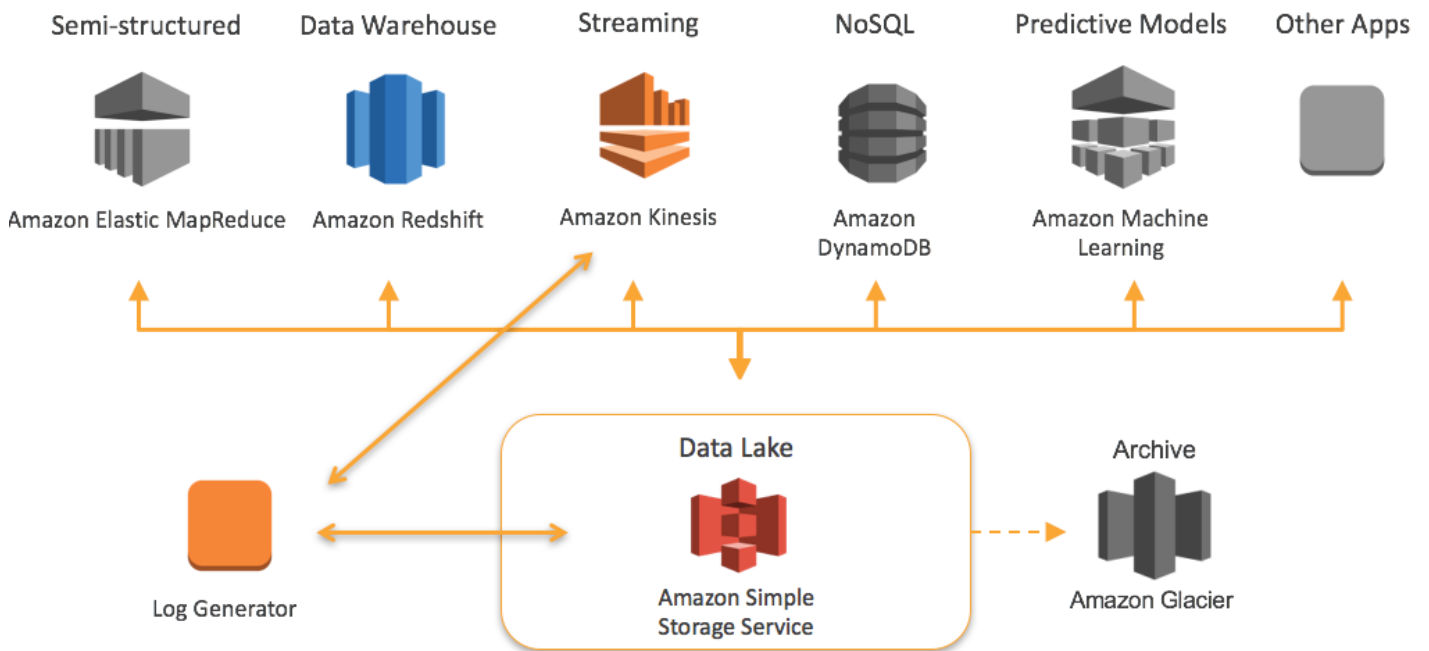
```
mydb=#
```

```
mydb=# select * from log limit 10;
```

```
char_class | char_name | kills | deaths | assists | is_winner
```

```
-----+-----+-----+-----+-----+-----
Fighter | Carle | 4 | 4 | 11 | t
Damager | Redthok | 2 | 4 | 3 | f
Tank | Cemorilthorn | 2 | 2 | 5 | f
Healer | Gelocks Brace | -1 | -1 | -1 | t
Wizard | Erilalta | 6 | 8 | 14 | t
Damager | Redthok | 9 | 22 | 10 | t
Fighter | Leofugrad | 0 | 2 | 2 | t
Fighter | Leofugrad | 16 | 3 | 18 | f
Assassin | Roggtul | 0 | 0 | 0 | f
Damager | Redthok | 17 | 15 | 6 | f
(10 rows)
```

다이어그램으로 보면 아래와 같이 현재 구성하였습니다.



Lab 3. Amazon QuickSight 를 이용한 BI 시각화

앞에서 simulator.py, run_consumer, inserter.py 를 통해서 데이터 생성, 저장, DW 적재를 수행하였습니다. DW 에 저장된 데이터를 Amazon QuickSight Business Intelligent 서비스를 통해 원하는 대로 데이터를 시각화하여 볼 수 있습니다.

참조: <https://quicksight.aws/>

Redshift cluster 에 적재된 데이터를 이용하여 summary 란 통계 테이블을 하나 생성합니다. Summarizer.py 란 파일을 수행하면 log table 에 적재된 데이터를 바탕으로 summary 란 통계 테이블을 Redshift cluster 에 생성합니다.

```
$ python summarizer.py
```

```
> dropping table...
```

```
> creating table...
```

```
done
```

Redshift cluster 에 접속하여 summary 테이블 내용을 확인합니다.

```
$ psql -h labs-dw-instance.c91bwulbunqt.us-west-2.redshift.amazonaws.com -p 5439 -U admin -d mydb
```

Password for user admin:

psql (8.4.20, server 8.0.2)

WARNING: psql version 8.4, server version 8.0.

Some psql features might not work.

SSL connection (cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256)

Type "help" for help.

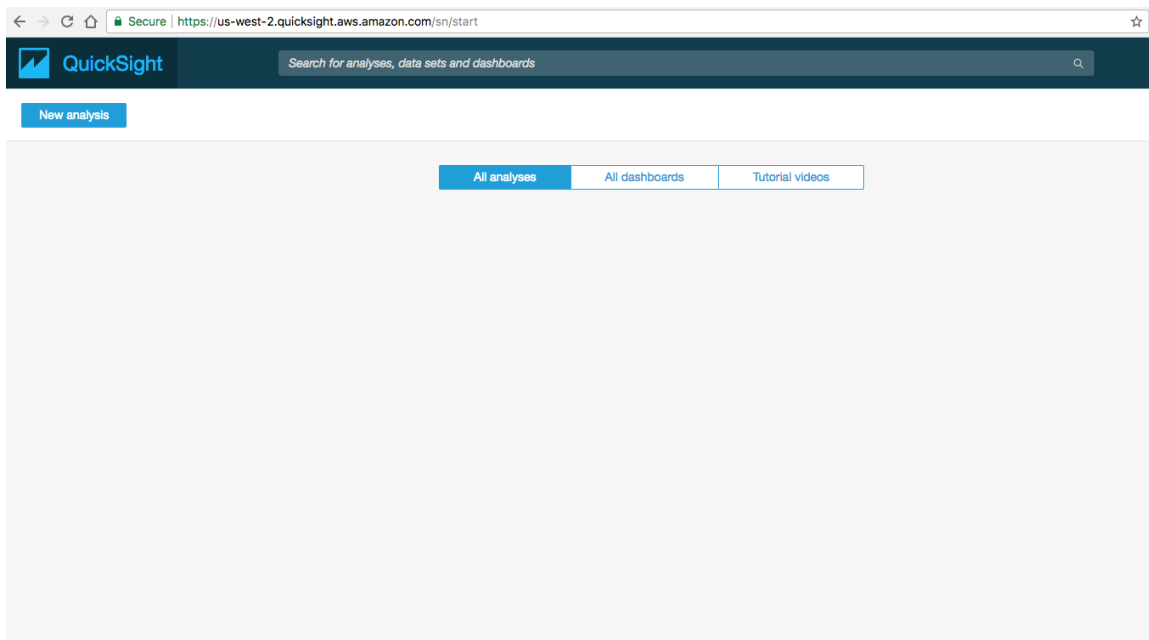
```
mydb=# select * from summary
```

```
mydb=# ;
```

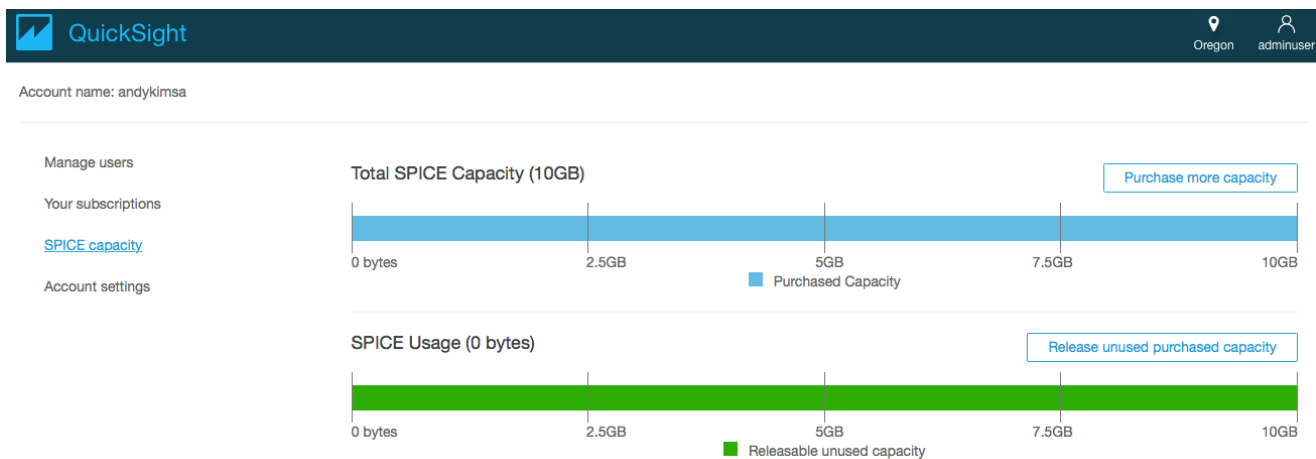
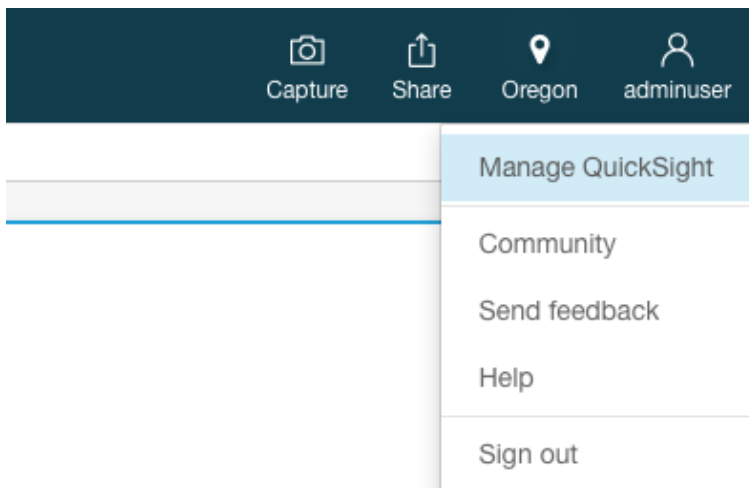
char_name	kills	deaths	assists	matches	win	lose	avg_kills	avg_deaths	avg_assists	win_loss_ratio
Lukthos	2525	2523	2354	357	185	172	7	7	7	0.51821
Leofugrad	2232	2224	2183	347	181	166	6	6	6	0.52161
Gelocks Brace	1562	1417	1464	217	118	99	7	7	7	0.54378
Roggtul	2371	2132	2331	370	186	184	6	6	6	0.5027
Cemorilthorn	2402	2258	2457	350	165	185	7	6	7	0.47143
Leedhel	2465	2369	2387	364	177	187	7	7	7	0.48626
Ilthi	2631	2566	2759	383	187	196	7	7	7	0.48825
Gaorz	3035	2838	2847	404	203	201	8	7	7	0.50248
Ralmen Long	2440	2677	2829	394	206	188	6	7	7	0.52284
Redthok	2497	2511	2394	370	168	202	7	7	6	0.45405
Thehilda	2444	2392	2360	365	181	184	7	7	6	0.49589
Brasta	1090	1159	1105	181	86	95	6	6	6	0.47514
Erilalta	2373	2399	2461	370	195	175	6	6	7	0.52703
Shamen	2679	2675	2839	386	185	201	7	7	7	0.47927
Thorwulf	2357	2428	2484	359	180	179	7	7	7	0.50139
Carle	2258	2472	2359	328	167	161	7	8	7	0.50915
Wulfcon	2554	2456	2313	369	183	186	7	7	6	0.49593
Holda Sack	2530	2560	2535	375	186	189	7	7	7	0.496

```
(18 rows)
```

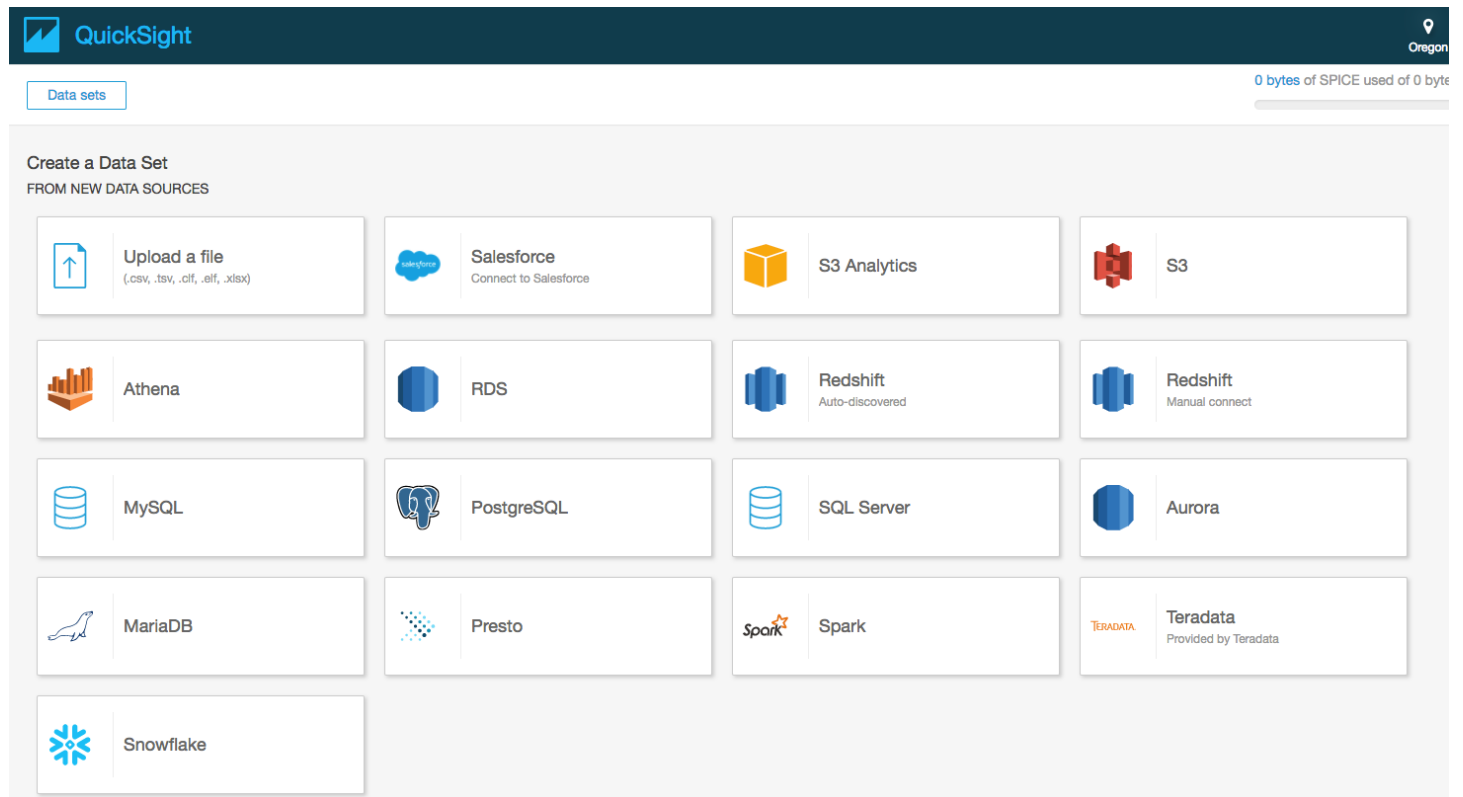
Web management console 에서 QuickSight 서비스로 이동합니다.



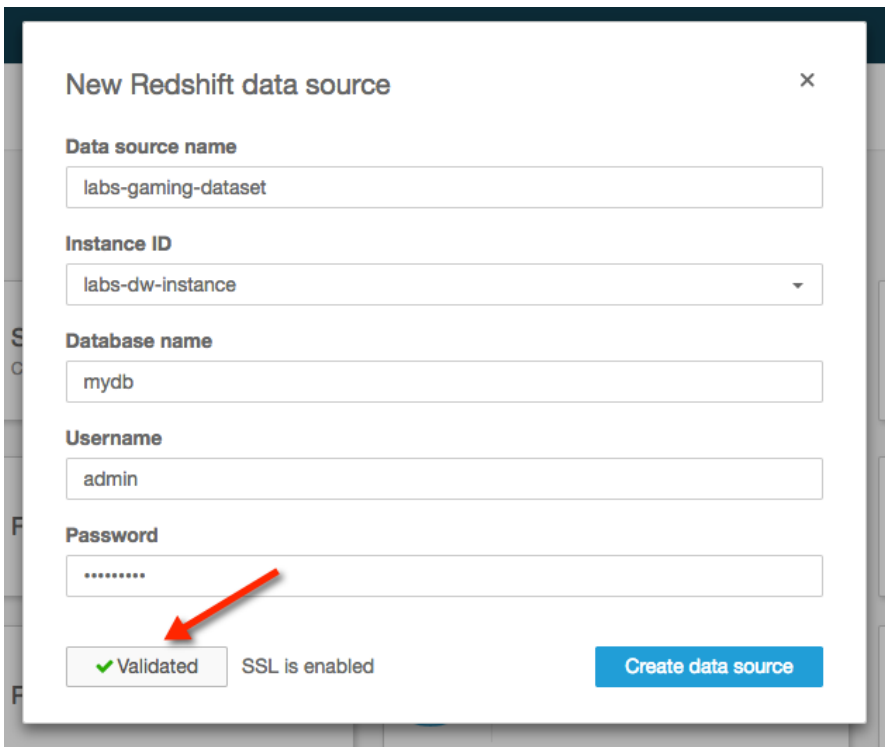
Oregon region 은 현재 QuickSight 가 SPICE (데이터를 메모리에 적재하여 성능향상) 사용할 공간이 없으므로, 아래와 같이 Manage QuickSight 에서 10GB 정도를 구매합니다.



New analysis 를 선택하고 New Data Set 을 선택합니다. QuickSight 에서 직접 데이터 소스를 참조할 수 있는 다양한 데이터 소스를 확인할 수 있습니다. 현재는 S3 와 Redshift 에 데이터가 적재되어 있으므로 두 가지를 데이터 소스로 지정할 수 있습니다.



동일한 Region 을 활용하고 있으므로 Redshift (Auto-discovered)를 선택하면 앞에서 생성된 Redshift cluster 를 바로 선택할 수 있습니다. Data Source Name 을 입력하고 아래 내용들을 모두 입력합니다. 모두 입력 후 Validate connection 을 해서 문제가 없이 접근이 가능하다면 Validated 로 바뀝니다. 아래 이미지를 참조하여 주십시오. 마지막으로 Create Data source 를 선택합니다.



New Redshift data source ×

Data source name
labs-gaming-dataset

Instance ID
labs-dw-instance ▼

Database name
mydb

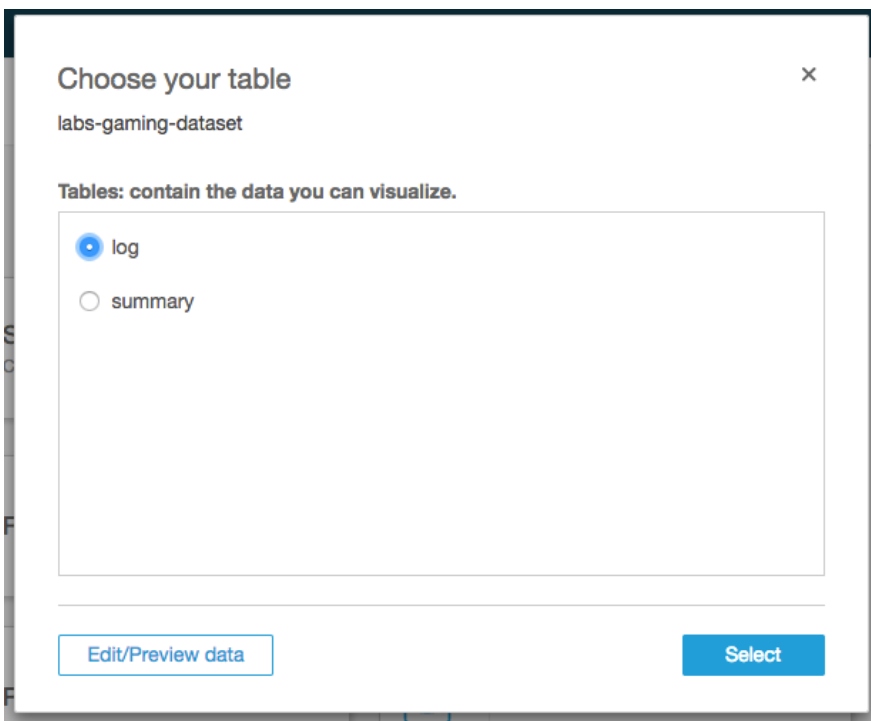
Username
admin

Password
.....

✓ Validated SSL is enabled

Create data source

앞에서 생성한 log, summary 테이블을 선택할 수 있습니다.



Choose your table ×

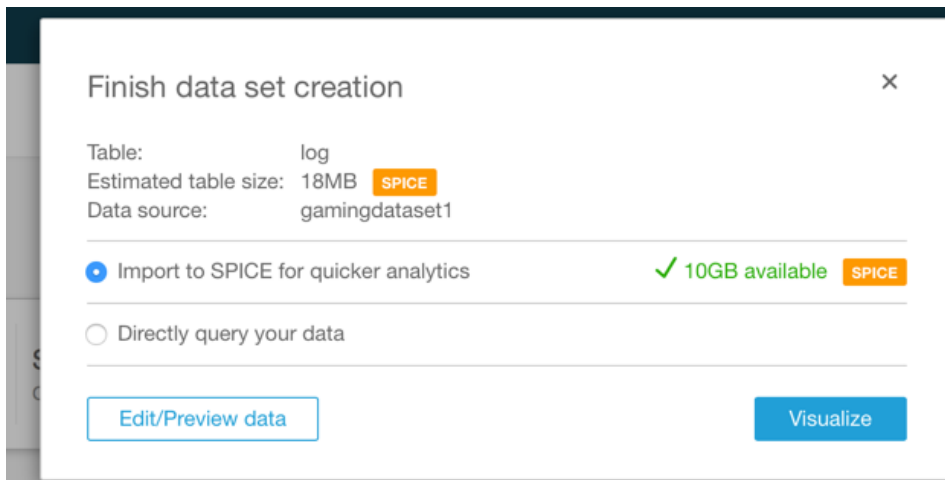
labs-gaming-dataset

Tables: contain the data you can visualize.

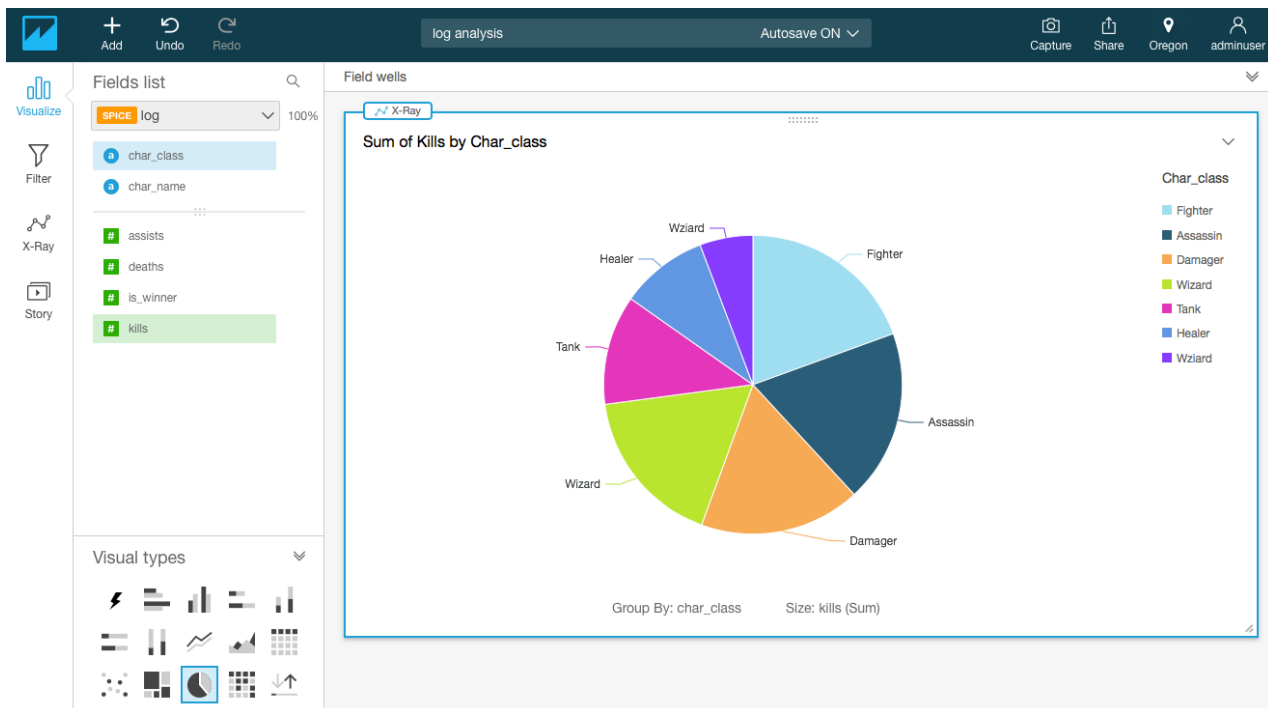
☒ log

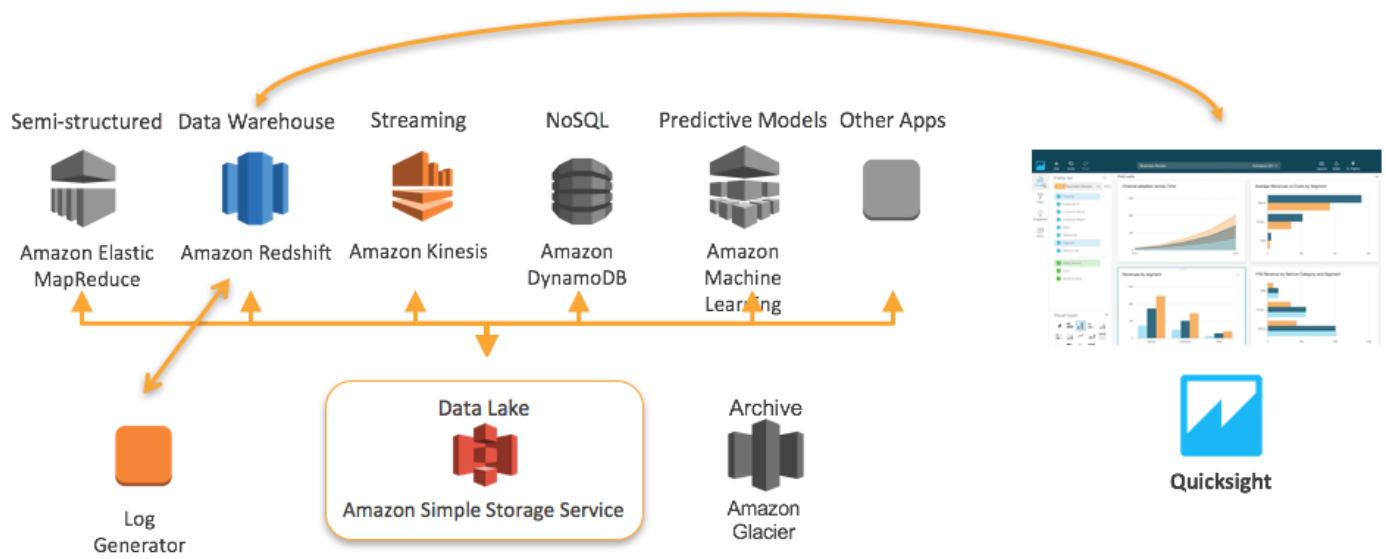
☐ summary

Edit/Preview data Select



Visualize 를 선택하면 기본적으로 아래와 같이 다양한 그래픽으로 데이터 분석이 가능한 화면으로 이동합니다.





현재까지 구성한 아키텍처를 보면 위와 같이 설명이 될 수 있습니다.

Lab 4. Amazon Elasticsearch 와 Lambda 를 통한 실시간 검색엔진

AWS 는 잘 알려진 Elasticsearch 를 관리형 서비스로 제공하고 있습니다. 데이터를 저장하고 바로 원하는 데이터를 쉽게 검색할 수 있으며, Elasticsearch 서비스의 운영 및 관리는 AWS 에서 지원을 합니다. 바로 생성하고 데이터를 입력하면 사용이 가능하게 됩니다. 이번 랩은 앞에서 만들어서 생성되고 있는 게이밍 로그 데이터를 Lambda 를 이용하여 바로 Elasticsearch 에 입력하고 사용하는 것을 구현합니다. Amazon 의 서비스는 서비스들 간의 Integration 이 잘 되어 있는 점입니다. Elasticsearch 역시 다양한 서비스를 데이터 소스로 연결할 수 있습니다.

참고: 다양한 소스로부터 Elasticsearch 로의 데이터 입력 지원

- [Loading Streaming Data into Amazon ES from Amazon S3](#)
- [Loading Streaming Data into Amazon ES from Amazon Kinesis](#)
- [Loading Streaming Data into Amazon ES from Amazon Kinesis Firehose](#)
- [Loading Streaming Data into Amazon ES from Amazon DynamoDB](#)
- [Loading Streaming Data into Amazon ES from Amazon CloudWatch](#)

Amazon Elasticsearch 생성

Web management console 에서 Elasticsearch 서비스 메뉴로 이동합니다. Create a new domain 버튼을 선택합니다.

1 단계: Define domain

Domain name 을 설정합니다. 아래 예를 gaminglabs 로 지정하였습니다. 입력 후 Next 로 이동합니다.

Create Elasticsearch domain

step 1: Define domain

step 2: Configure cluster

step 3: Set up access

step 4: Review

Define domain

A domain is a collection of all the resources needed to run your Elasticsearch cluster.

Domain Name

Enter a name for your Elasticsearch domain. The domain name will be part of your domain endpoint.

Elasticsearch domain name

The name must start with a lowercase letter and must be between 3 and 28 characters long (lowercase only, 0-9, and - (hyphen)).

Version

Select the version of the Elasticsearch engine for your domain.

Elasticsearch version

2 단계: Configure cluster

Elasticsearch cluster 구성 설정입니다. 기본 구성 그대로 진행합니다. Next 버튼을 눌러 다음으로 진행합니다.

3 단계: Set up access

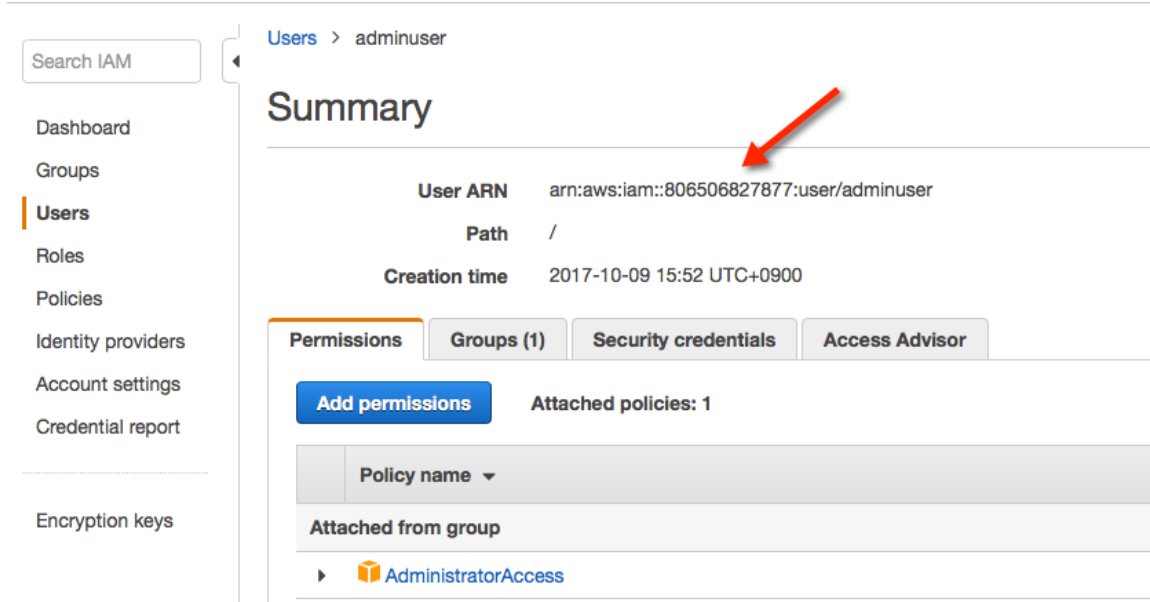
Network configuration 에서 Public access 를 설정합니다.

Network configuration

Choose internet or VPC access. To enable VPC access, we will use private IP addresses from your VPC, which provides security by default. You control network access within your VPC using security groups. You can optionally add an additional layer of security by applying a restrictive access policy. Internet endpoints are publicly accessible. If you select public access, you should secure your domain with an access policy that only allows specific users or IP addresses to access the domain.

- ☒ Public access
☐ VPC access

Elasticsearch domain 이 접근할 수 있는 접근 권한을 설정합니다. 앞에서 생성한 adminuser 사용자만 접근가능하도록 설정합니다. 먼저 다른 브라우저 창으로 IAM 메뉴에서 adminuser 의 ARN 을 확인하고 복사합니다.



Search IAM

Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report

Encryption keys

Users > adminuser

Summary

User ARN arn:aws:iam::806506827877:user/adminuser
Path /
Creation time 2017-10-09 15:52 UTC+0900

Permissions **Groups (1)** **Security credentials** **Access Advisor**

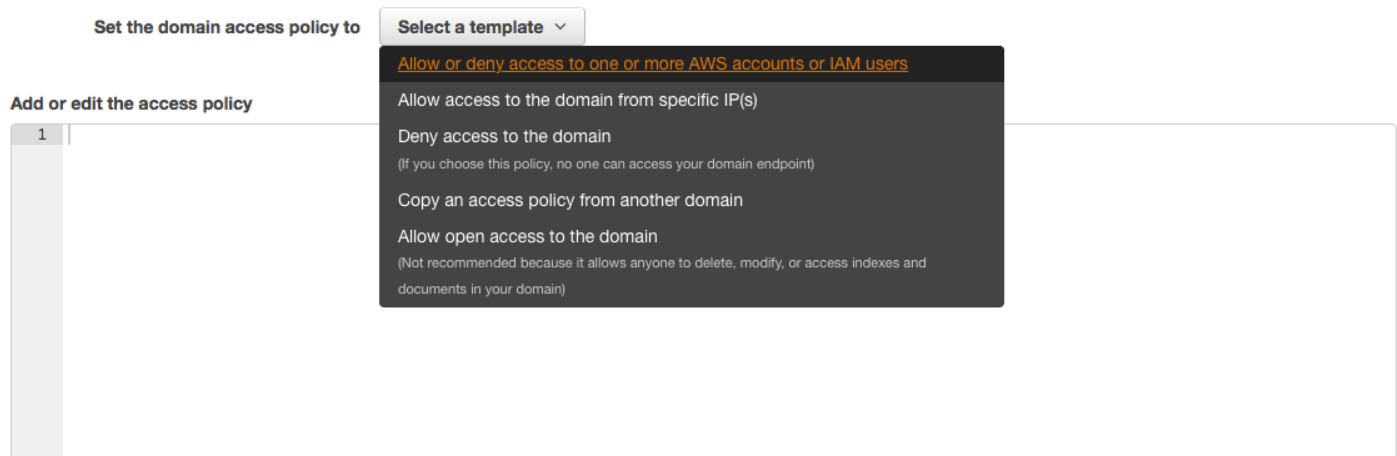
Add permissions **Attached policies: 1**

Policy name
Attached from group
AdministratorAccess

Elasticsearch 구성에서 이어서 아래와 같이 Set the domain access policy to 에서 "Allow or deny access to one or more AWS accounts or IAM users"를 선택합니다.

Access policy

To allow or block access to the domain, select a policy template from the template selector or add one or more Identity and Access Management (IAM) policy statements in the **Edit the access policy** box.



Set the domain access policy to

Select a template

- Allow or deny access to one or more AWS accounts or IAM users
- Allow access to the domain from specific IP(s)
- Deny access to the domain
(If you choose this policy, no one can access your domain endpoint)
- Copy an access policy from another domain
- Allow open access to the domain
(Not recommended because it allows anyone to delete, modify, or access indexes and documents in your domain)

Add or edit the access policy

1

adminuser 의 ARN 정보를 아래와 같이 Account ID or ARN* 에 입력합니다.

User access

Type a comma-separated list of valid AWS account IDs, AWS account ARNs, or IAM user ARNs.

Examples:
AWS account ID 806506827877
AWS account ARN arn:aws:iam::AWS-account-ID:root ⓘ
IAM user ARN arn:aws:iam::AWS-account-ID:user/user-name-1 ⓘ

Effect

Allow

Account ID or ARN*

iam::[redacted]:user/adminuser|

Cancel

OK

OK 버튼을 누른 후, Next 버튼을 눌러 진행합니다.

4 단계: Review

내용 확인 후 Confirm 버튼을 선택합니다. 수 분 후 Elasticsearch cluster 가 생성됩니다.

실시간 데이터 로딩을 위한 Lambda 함수 생성

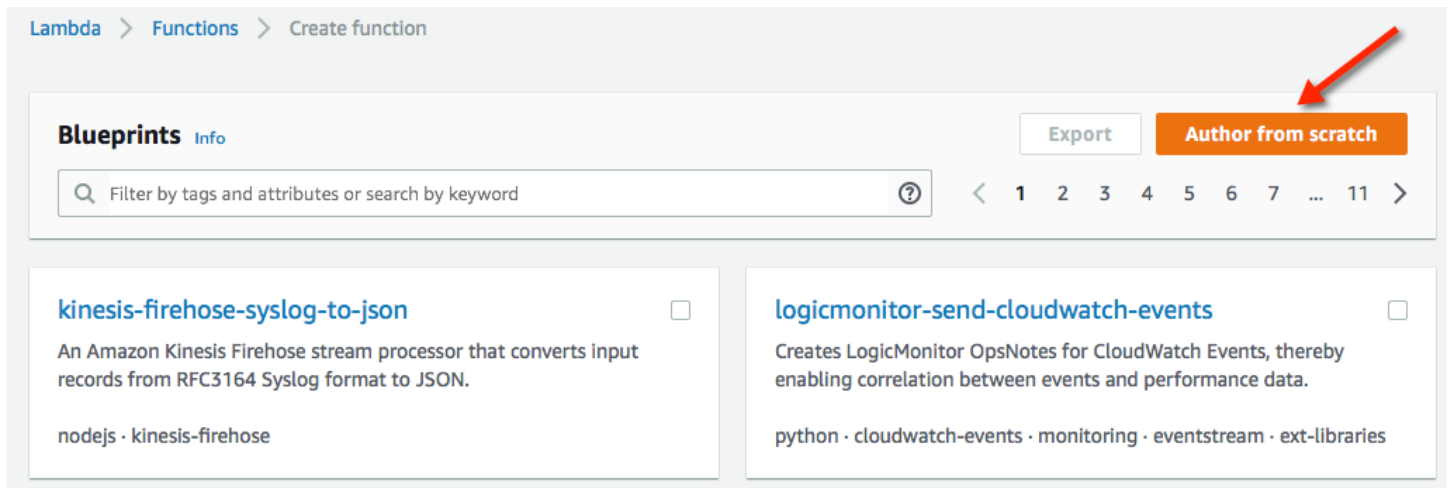
우선, Nodejs 를 사용하는 기본 Lambda 함수를 하나 생성하고, 코드는 후에 작업하여 업데이트 하도록 합니다.

Lambda 함수 생성

AWS management console 에서 Lambda 서비스로 이동합니다.

Create function 을 선택합니다.

빈 함수로 시작하려면 Author from scratch 를 선택합니다.



함수 이름을 원하는 이름으로 입력한 후 Role 은 lambda_basic_execution 을 아래 그림과 같이 선택합니다.

Lambda > Functions > Create function > Author from scratch

Basic information Info

Name*

Role*

Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.

Choose an existing role ▼

Existing role*

You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.

lambda_basic_execution ▼

* These fields are required.

Cancel Previous **Create function**

Create function 을 선택하면, 함수가 생성됩니다. Default 언어가 Nodejs 로 선택됩니다. 화면 아래 Basic settings 에서 Timeout 시간을 3 초에서 10 초로 변경해 줍니다. 변경 후 위 쪽에 Save 버튼을 눌러 저장합니다.

▼ Execution role

Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.

Choose an existing role ▼

Existing role

You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.

lambda_basic_execution ▼

▼ Basic settings

Memory (MB) Info

Your function is allocated CPU proportional to the memory configured.

128 MB

Timeout Info

0 min 10 sec

Description

Lambda 함수 코드 수정 및 업로드

실제 동작할 코드를 Instance 에서 수정하여 압축하고 방금 생성의 Lambda 함수에 업로드 합니다.

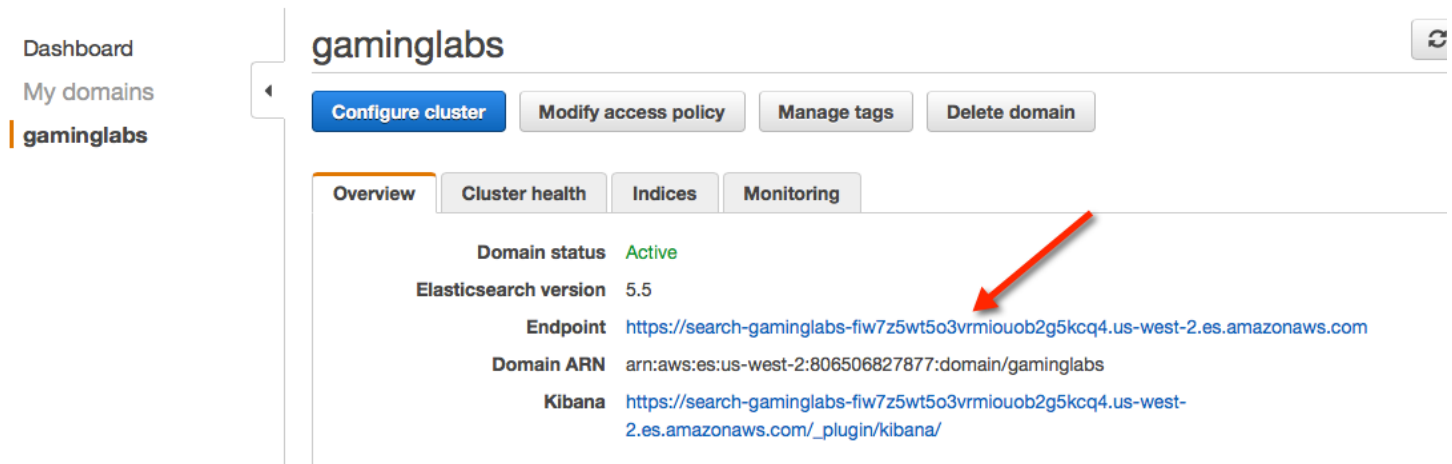
미리 코드를 생성하여 준비해 놓았습니다. 코드를 다운로드할 폴더를 생성하고 코드와 필요한 패키지를 미리 저장해 놓은 파일을 다운로드 합니다.

```
$ mkdir eslambdacd eslambdac
```

```
$ wget https://s3-ap-northeast-1.amazonaws.com/www.aws-korea.com/eslambdac.zip
```

```
$ unzip eslambdac.zip
```

index.js 파일을 열어 아래 코드에서 Elasticsearch 의 endpoint 정보를 수정합니다. Elasticsearch 의 Endpoint 는 앞에서 생성한 Elasticsearch cluster 를 선택하면 확인이 가능합니다. 아래 예제 그림을 참조하여 Endpoint 를 확인하십시오.



코드에서는 아래 부분만을 수정하고 저장합니다.

```
var esDomain = {
  region: 'us-west-2',
  endpoint: '[INPUT_YOUR_ELASTICSEARCH_ENDPOINT]',
  index: 'myindex',
  doctype: 'mytype'
};
```

이제 수정된 코드와 패키지를 다시 ZIP 파일로 묶습니다. 아래 예는 export.zip 이란 이름으로 묶었습니다.

```
$ zip -r export.zip *
```

AWS 커맨드 라인 툴은 Lambda 에 대한 기능도 지원합니다. 따라서 코드를 AWS CLI 로 바로 업로드 할 수 있습니다.

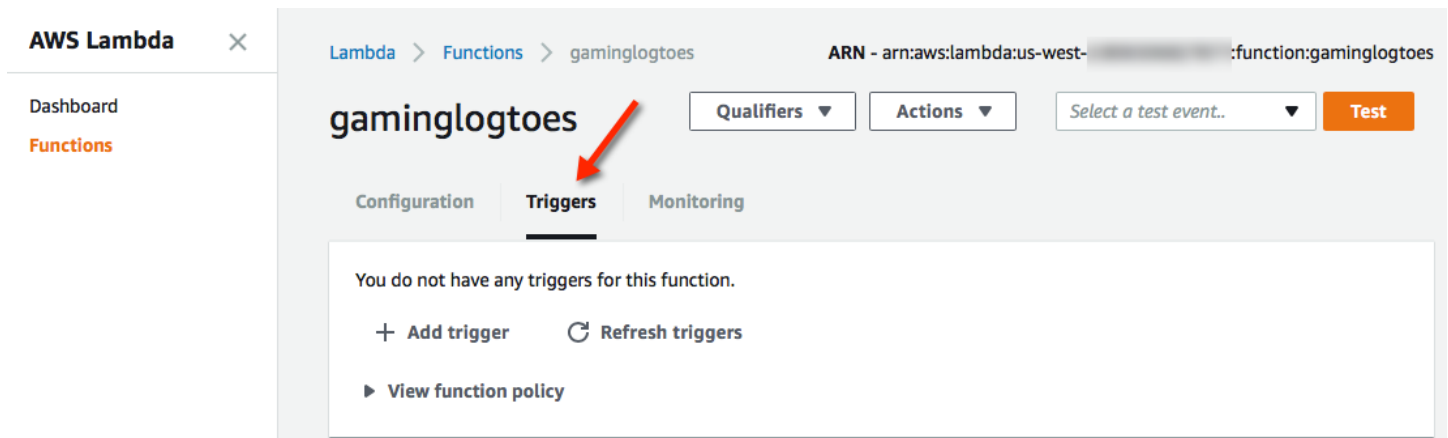
```
$ aws lambda update-function-code --function-name [YOUR_LAMBDA_FUNCTION_NAME] --zip-file
fileb://export.zip --region us-west-2
```

위와 같이 실행하면, 앞에서 만든 Lambda 함수에 방금 수정된 코드와 패키지가 업로드 되어 동작하게 됩니다.

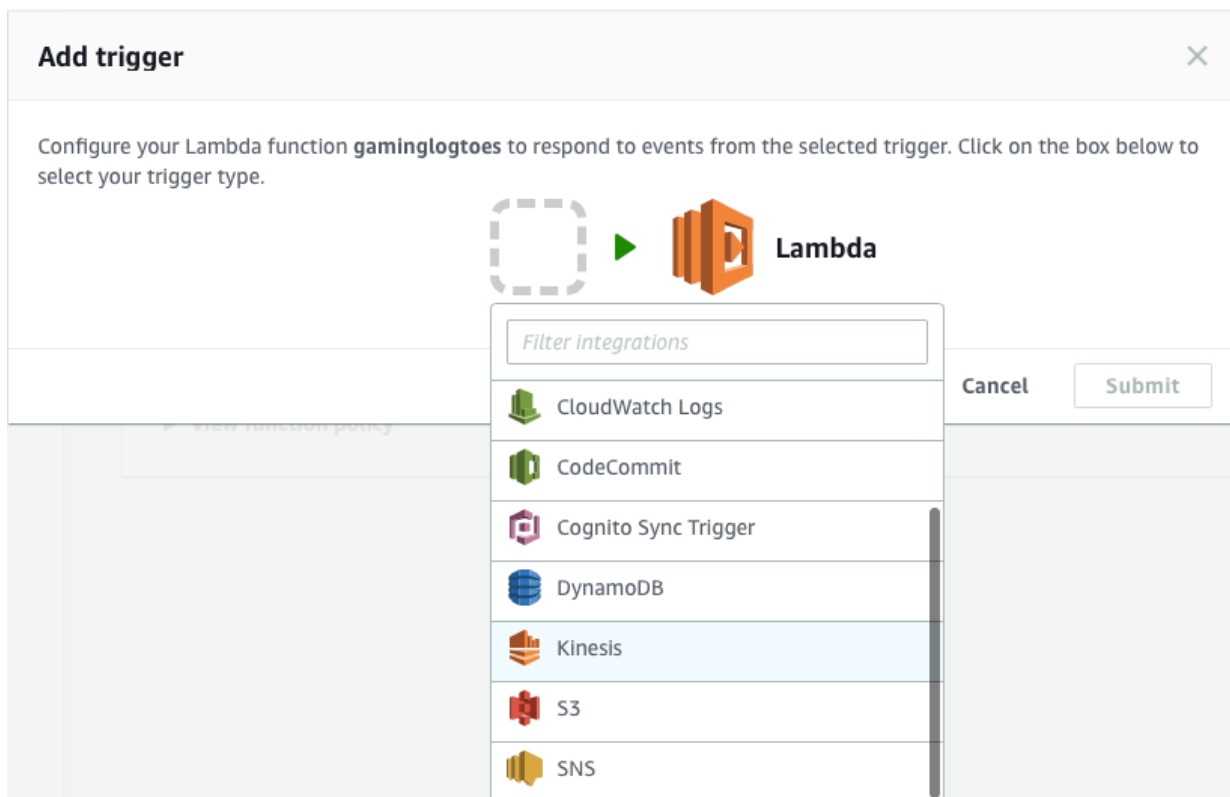
Lambda 함수 트리거 설정

Lambda 는 특정한 이벤트가 발생할 때 마다 실행될 수 있는 기능을 가지고 있습니다. Kinesis stream 을 이벤트 트리거로 설정할 수 있습니다.

앞에 생성한 Lambda 함수를 선택합니다.



"+ Add trigger"를 선택하고, Kinesis 를 선택합니다.



Kinesis 를 선택하면 어떠한 Kinesis stream 을 사용할 지 선택해야 합니다. 앞에서 생성한 Kinesis stream 을 선택합니다. Batch size 는 한 번에 가져올 레코드 개수로 10 개로 합니다. Starting position 은 Latest 로 합니다.

Configure your Lambda function **gaminglogtoes** to respond to events from the selected trigger. Click on the box below to select your trigger type.



Kinesis stream
Please select a Kinesis stream. The Lambda function will be invoked whenever this stream is updated.

lab-game-stream ▼

Batch size
The largest number of records that AWS Lambda will retrieve from your stream at the time of invoking your function. Your function receives an event with all the retrieved records.

10

Starting position
The position in the stream where AWS Lambda should start reading. For more information, see [ShardIteratorType](#) in the Amazon Kinesis API Reference.


Latest ▼

저장 완료하면 아래와 같이 Lambda 함수 트리거가 설정됩니다.

Configuration

Triggers

Monitoring



Kinesis: labs-game-stream

arn:aws:kinesis:us-west-2:806506827877:stream/labs-game-stream

Last processing result: **No records processed** Batch size: **10**

Disable

Delete

+ Add trigger

↻ Refresh triggers

▶ View function policy

데이터 생성 및 Elasticsearch cluster 저장 확인

모든 구성이 완료되었으므로, 게이밍 로그가 `$ python simulator.py` 로 계속 수행 중이라면 이제부터 Kinesis stream 으로 입력된 데이터를 자동으로 Lambda 함수를 통해 Elasticsearch cluster 로 저장되게 됩니다. 앞에서 `simulator.py` 수행을 멈췄다면 다시 수행합니다.

```
$ python simulator.py
```

Elasticsearch cluster 로 이동하여, Indices 에서 myindex 가 생성되고 데이터 사이즈가 증가하는 것을 볼 수 있습니다.

The screenshot shows the AWS Elasticsearch console for a domain named 'gaminglabs'. On the left sidebar, 'Dashboard' and 'My domains' are visible, with 'gaminglabs' selected. The main panel has tabs for 'Overview', 'Cluster health', 'Indices', and 'Monitoring', with 'Indices' currently active. Under the 'Indices' tab, there are two indices listed: '.kibana' and 'myindex'. The 'myindex' index is expanded, displaying its details: 'Count' is 1760, 'Size in bytes' is 314.03 kB, 'Query total' is 0, and 'Mappings' for 'mytype' are listed with fields: 'assists' (long), 'char_class' (text), 'char_name' (text), 'deaths' (long), 'is_winner' (boolean), and 'kills' (long).

Kibana 로 데이터 확인

Amazon Elasticsearch 는 Kibana 툴을 기본으로 제공해 줍니다. 단, 당연히 제한된 접근으로만 접근이 가능해야 하므로, Elasticsearch 에서 Access policy 를 수정해 줘야만 합니다.

Elasticsearch cluster 로 이동하여, Modify access policy 를 선택합니다.

Dashboard
My domains
gaminglabs

gaminglabs

Configure cluster Modify access policy Manage tags Delete d

Overview Cluster health Indices Monitoring

Domain status **Active**

Elasticsearch version 5.5

Endpoint <https://search-gaminglabs-fiw7z5wt5o3vrmiouc>

선택후 Set the domain access policy to 에서 "Allow access to the domain from specific IP(s)"를 선택하여 현재 본인이 접속하고 있는 Public IP Address 를 입력하고 저장합니다.

IP 확인: <https://www.whatismyip.com>

Dashboard
My domains
gaminglabs

Modify the access policy for gaminglabs

To allow or block access to the domain, select a policy template from the template selector or add one or more Identity and Access Management (IAM) the access policy box.

Status **Active**

Set the domain access policy to **Select a template**

- Allow or deny access to one or more AWS accounts or IAM users
- Allow access to the domain from specific IP(s)**
- Deny access to the domain (If you choose this policy, no one can access your domain endpoint)
- Copy an access policy from another domain
- Allow open access to the domain (Not recommended because it allows anyone to delete, modify, or access indexes and documents in your domain)

Add or edit the access policy

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "AWS": "*"
8       },
9       "Action": "es:*",
10      "Resource": "arn:aws:es:us-west-2:domain/gaminglabs/*",
11      "Condition": {
12        "IpAddress": {
13          "aws:SourceIp": "205.251.233.180"
14        }
15      }
16    }
17  ]
18 }
```

변경 후 저장하면 아래와 유사한 Policy 로 저장이 됩니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```

"Effect": "Allow",
"Principal": {
  "AWS": "*"
},
"Action": "es:*",
"Resource": "arn:aws:es:us-west-2:1234567890:domain/gaminglabs/*",
"Condition": {
  "IpAddress": {
    "aws:SourceIp": "205.251.233.180"
  }
}
}
}
}
}
}

```

Status 가 Processing 으로 변경되었다가, 수 분 후 Active 로 바뀌면 Kibana URL 을 통해 접근하여 저장된 데이터를 확인합니다.

The screenshot shows the Kibana web interface. At the top, it indicates 1,762 hits. A search bar contains the query "Search... (e.g. status:200 AND extension:PHP)". The left sidebar has a menu with options: Discover, Visualize, Dashboard, Timelion, Dev Tools, and Management. The main content area shows a list of search results. The first result is a configuration object for Kibana. The subsequent results are game-related records, each containing fields like kills, deaths, is_winner, assists, char_class, char_name, and _id.

현재 구성한 내용을 아래와 같은 아키텍처로 간단히 요약해 볼 수 있습니다.



Lab 5. Amazon EMR (Elastic MapReduce) 에서 S3 데이터 직접 쿼리

Amazon EMR 은 관리형 하둡 프레임워크로서 동적으로 확장 가능한 Amazon EC2 인스턴스 전체에서 대량의 데이터를 쉽고 빠르게 비용 효율적으로 처리할 수 있습니다. 또한, Amazon EMR 에서 [Apache Spark](#), [HBase](#), [Presto](#) 및 [Flink](#) 와 같이 널리 사용되는 분산 프레임워크를 실행하고, Amazon S3 및 Amazon DynamoDB 와 같은 다른 AWS 데이터 스토어의 데이터와 상호 작용할 수 있는 서비스입니다. 데이터 분석에서 Amazon EMR 을 활용할 수 있는 부분은 매우 다양합니다. 단순한 ETL 부터 데이터 분석, 실시간 분석 등 제한이 없습니다.

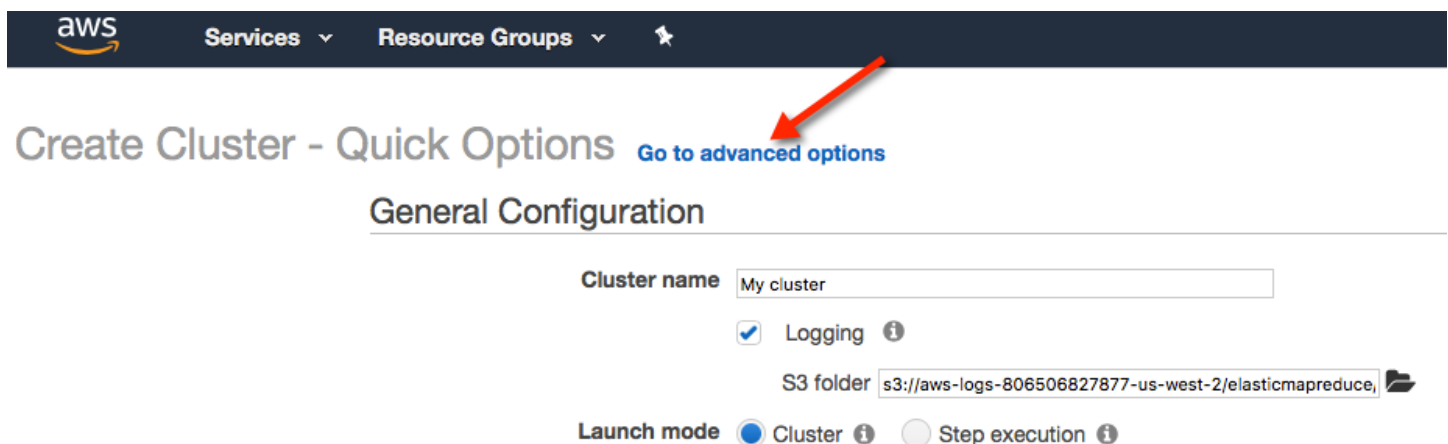
데이터 파일에 직접 Table 생성

EMR Hive 를 이용하면 S3 에 저장된 파일을 Local HDFS 로 가져오지 않고 직접 Table 을 생성하여 접근할 수 있습니다. Hive external table 을 활용하며 바로 스키마를 구성하여 테이블을 생성하고 쿼리를 실행할 수 있으며, Metastore 를 분리할 경우, 클러스터를 없애고 새로 생성할 경우에도 테이블 상태 그대로 다시 접근할 수 있습니다. Data Lake 의 기본이 되는 중요한 기능입니다.

Amazon EMR 클러스터 생성

Web management console 에서 Amazon EMR 서비스로 이동합니다.

Create cluster 를 선택합니다. 아래 그림과 같이 Go to advanced options 이라는 옵션을 선택합니다.



최신 Release 를 선택하고 Spark 을 추가로 체크합니다. Next 를 눌러 진행합니다.

Software Configuration

Release

<input checked="" type="checkbox"/> Hadoop 2.7.3	<input type="checkbox"/> Zeppelin 0.7.2	<input type="checkbox"/> Livy 0.4.0
<input type="checkbox"/> Tez 0.8.4	<input type="checkbox"/> Flink 1.3.2	<input type="checkbox"/> Ganglia 3.7.2
<input type="checkbox"/> HBase 1.3.1	<input checked="" type="checkbox"/> Pig 0.17.0	<input checked="" type="checkbox"/> Hive 2.3.0
<input type="checkbox"/> Presto 0.184	<input type="checkbox"/> ZooKeeper 3.4.10	<input type="checkbox"/> Sqoop 1.4.6
<input type="checkbox"/> Mahout 0.13.0	<input checked="" type="checkbox"/> Hue 4.0.1	<input type="checkbox"/> Phoenix 4.11.0
<input type="checkbox"/> Oozie 4.3.0	<input checked="" type="checkbox"/> Spark 2.2.0	<input type="checkbox"/> HCatalog 2.3.0

다음으로 진행하여, Network 에서 기본 현재 사용 중인 vpc-xxxxxx 를 선택합니다. Core node instance 노드는 2 개에서 1 개로 변경합니다.

Hardware Configuration

If you need more than 20 EC2 instances, [see this topic](#).

- Instance group configuration**
- ☒ **Uniform instance groups**
Specify a single instance type and purchasing option for each node type.
 - ☐ **Instance fleets**
Specify target capacity and how Amazon EMR fulfills it for each node type. Mix instance types and purchasing options. [Learn more](#)

Network [Create a VPC](#)

EC2 Subnet

Root device EBS volume size GiB

Node type	Instance type	Instance count	Purchasing option	Auto Scaling
Master Master - 1	m3.xlarge 8 vCPU, 15 GiB memory, 80 SSD GB storage EBS Storage: none	1 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Maximum bid price: \$	Not available fo
Core Core - 2	m3.xlarge 8 vCPU, 15 GiB memory, 80 SSD GB storage EBS Storage: none	<input type="text" value="1"/> Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Maximum bid price: \$	Not enabled
Task Task - 3	m3.xlarge 8 vCPU, 15 GiB memory, 80 SSD GB storage EBS Storage: none	<input type="text" value="0"/> Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Maximum bid price: \$	Not enabled

Next 를 눌러 다음으로 이동합니다. General Cluster Settings 는 변경 없이 다음으로 이동합니다.

Security 에서는 EC2 key pair 에 앞에서 EC2 instance 생성에서 만든 EC2 key pair 를 선택합니다.

Create Cluster - Advanced Options [Go to quick options](#)

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

Security Options

EC2 key pair  

☒ Cluster visible to all IAM users in account 

Permissions 

☒ Default ☐ Custom

Create cluster 버튼을 눌러 EMR 클러스터를 생성합니다. 수 분 후 EMR 클러스터가 생성 완료되면, Cluster 의 Status 가 Waiting 으로 변경됩니다.

Hive 를 이용하여 S3 에 직접 쿼리하기

EMR 은 S3 에 저장된 데이터 파일에 대해서 직접 테이블을 생성하고 쿼리를 이용할 수 있는 기능을 제공합니다. 이는 Data Lake 개념을 쉽게 적용할 수 있는 편리한 기능으로, EMR 은 S3 외에도 Kinesis, DynamoDB 와 같은 서비스에도 직접 접근이 가능한 기능을 가지고 있습니다.

EMR Cluster master 노드에 접속하기 위해서는 EMR Cluster master 노드의 Security group 에서 SSH 접속을 허가해 주어야만 합니다. EMR 에서 생성된 클러스터를 선택하여 상세 메뉴로 이동합니다. 아래 그림과 같이 Security group 을 찾아 선택하면 해당 Security group 설정 화면으로 이동합니다.

Clone Terminate **AWS CLI export**

Cluster: My cluster **Waiting** Cluster ready after last step completed.

Summary Application history Monitoring Hardware Events Steps Configurations Bootstrap actions

Connections: [Enable Web Connection](#) – Hue, Spark History Server, Resource Manager ... (View All)

Master public DNS: 35.164.158.175 [SSH](#)

Tags: -- [View All / Edit](#)

Summary

ID: j-2JYT4HDSISLCF
 Creation date: 2017-10-16 13:51 (UTC+9)
 Elapsed time: 8 hours, 18 minutes
 Auto-terminate: No
 Termination On [Change](#)
 protection:

Configuration details

Release label: emr-5.9.0
 Hadoop distribution: Amazon 2.7.3
 Applications: Hive 2.3.0, Pig 0.17.0, Hue 4.0.1, Spark 2.2.0
 Log URI: s3://aws-logs-806506827877-us-west-2/elasticmapreduce/
 EMRFS consistent view: Disabled
 Custom AMI ID: --

Network and hardware

Availability zone: us-west-2c
 Subnet ID: [subnet-373e0a71](#)
 Master: **Running** 1 m3.xlarge
 Core: **Running** 1 m3.xlarge
 Task: --

Security and access

Key name: ilho_oregon
 EC2 instance profile: EMR_EC2_DefaultRole
 EMR role: EMR_DefaultRole
 Auto Scaling role: EMR_AutoScaling_DefaultRole
 Visible to all users: All [Change](#)
 Security groups for [sg-377dee4a](#) (ElasticMapReduce-Master: master)
 Security groups for [sg-a563f0d8](#) (ElasticMapReduce-Core & Task: slave)

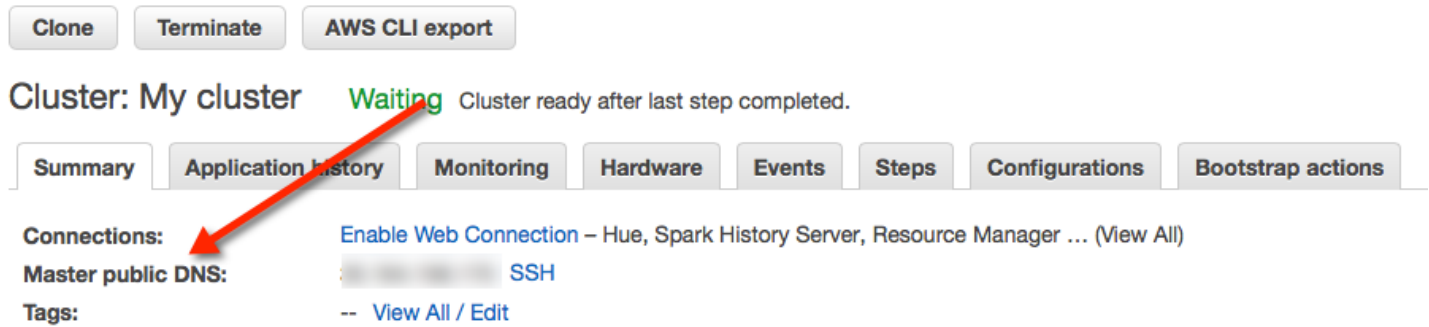
ElasticMapReduce-master Security group 을 선택하고 Inbound 에서 SSH 룰을 허가해 줍니다. Source 에서 My IP 를 선택하여 본인의 Public IP 만 허가하도록 합니다. Save 를 눌러 저장합니다.

Edit inbound rules

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
All TCP	TCP	0 - 65535	Custom sg-377dee4a
All TCP	TCP	0 - 65535	Custom sg-a563f0d8
SSH	TCP	22	Custom 205.251.233.180/32
Custom TCP Ru	TCP	8443	Custom 205.251.233.48/29

해당 클러스터를 선택하면 클러스터 상세 메뉴로 이동합니다. Master public DNS의 IP 주소를 참고하여 EC2 instance에 접속하는 것과 같이 SSH 접속을 클러스터 Master 노드에 합니다. 단, User ID는 ec2-user가 아니라 hadoop인 부분이 다릅니다.

```
$ ssh -i [YOUR_EC2_KEY_PAIR] hadoop@[Master public DNS]
```



S3 bucket에 저장된 파일을 AWS CLI 명령으로 확인합니다.

```
$ aws s3 ls s3://[YOUR_S3_BUCKET_NAME]
```

아래와 같이 파일로 저장된 로그 파일을 확인할 수 있습니다.

```
$ aws s3 ls s3://[YOUR_S3_BUCKET_NAME]/
2017-10-11 01:37:00 41247 gamelog_20171011-0135.dat
2017-10-11 01:37:01 211 gamelog_20171011-0136.dat
2017-10-11 01:38:02 11703 gamelog_20171011-0137.dat
2017-10-11 01:39:02 11839 gamelog_20171011-0138.dat
2017-10-11 01:40:01 12286 gamelog_20171011-0139.dat
2017-10-11 01:41:02 11315 gamelog_20171011-0140.dat
2017-10-11 01:42:02 12495 gamelog_20171011-0141.dat
```

.....

아래 명령을 통해 input 폴더를 bucket 밑에 만들어서 파일들을 복제합니다.

```
$ aws s3 cp s3://[YOUR_S3_BUCKET_NAME] s3://[YOUR_S3_BUCKET_NAME]/input/ --recursive
```

hive를 실행합니다.

```
$ hive
```

```
hive>
```

hive external table 을 생성합니다. 아래 코드를 S3 bucket 이름만 바꾸고 hive shell 에서 그대로 실행합니다. 테이블 이름은 gaming 으로 합니다.

```
ADD JAR /usr/lib/hive-hcatalog/share/hcatalog/hive-hcatalog-core-2.3.0-amzn-0.jar;
CREATE EXTERNAL TABLE gaming (
kills int,
deaths int,
is_winner boolean,
assists int,
char_class string,
char_name string)
ROW FORMAT serde 'org.apache.hive.hcatalog.data.JsonSerDe'
with serdeproperties ( 'paths'='kills, deaths, is_winner, assists, char_class, char_name' )
LOCATION 's3://[YOUR_S3_BUCKET_NAME]/input';
```

실제 select query 를 통해서 데이터를 읽어봅니다.

```
hive> select * from gaming limit 10;
OK
0      15      false   4      Wziard  Thehilda
11     6       true    2      Damager   Wulfcon
5      6       true    1      Fighter  Leofugrad
4      26      false   23     Tank     Cemorilthorn
12     1       true    10     Wizard   Erilalta
13     14      true    9      Damager   Thorwulf
5      6       true    5      Wizard   Ralmen Long
2      2       true    0      Fighter  Carle
7      9       false   6      Healer   Gelocks Brace
0      4       true    11     Wizard   Holda Sack
Time taken: 0.095 seconds, Fetched: 10 row(s)
hive>
```

Hadoop 2 에서는 기본 엔진이 Tez 로 되어 있습니다. 기존에는 MR 이 사용되었으나, 성능에서 Spark, Tez 가 우세하므로 변경되어 있습니다.

```
hive> set hive.execution.engine;
hive.execution.engine=tez
```

몇 개의 Record 인지 아래 쿼리를 실행해 봅니다.

```
hive> select count(*) from gaming;
```

```
Query ID = hadoop_20171016152646_84b5c04a-6715-48d9-9c8e-3d7eef2978c2
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1508162208449_0010)
```

```
-----
VERTICES   MODE      STATUS TOTAL COMPLETED RUNNING PENDING FAILED KILLED
```

```

-----
Map 1 ..... container  SUCCEEDED  1      1      0      0      0      0
Reducer 2 ..... container  SUCCEEDED  1      1      0      0      0      0
-----
VERTICES: 02/02 [=====>>] 100% ELAPSED TIME: 17.35 s
-----
OK
10629
Time taken: 21.971 seconds, Fetched: 1 row(s)

```

이번에는 engine 을 기존 MR 로 변경하여 실행해 봅니다.

```
hive> set hive.execution.engine=mr;
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution
engine (i.e. spark, tez) or using Hive 1.X releases.
```

```

hive> select count(*) from gaming;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a
different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = hadoop_20171016153017_c4142b9a-7ea9-44eb-b703-7a7e6e300653
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1508162208449_0011, Tracking URL = http://ip-172-16-0-113.us-west-
2.compute.internal:20888/proxy/application_1508162208449_0011/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1508162208449_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-10-16 15:30:28,122 Stage-1 map = 0%, reduce = 0%
2017-10-16 15:30:39,652 Stage-1 map = 16%, reduce = 0%, Cumulative CPU 9.96 sec
2017-10-16 15:30:42,776 Stage-1 map = 53%, reduce = 0%, Cumulative CPU 12.29 sec
2017-10-16 15:30:43,815 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 12.73 sec
2017-10-16 15:30:50,065 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 15.4 sec
MapReduce Total cumulative CPU time: 15 seconds 400 msec
Ended Job = job_1508162208449_0011
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 15.4 sec HDFS Read: 6435 HDFS Write: 105 SUCCESS
Total MapReduce CPU Time Spent: 15 seconds 400 msec
OK
10629
Time taken: 34.102 seconds, Fetched: 1 row(s)

```

참고로 수행 시간 차이가 발생하는 것을 볼 수 있습니다.

Lab 6. Amazon Athena 를 이용하여 직접 S3 데이터 쿼리

앞에서 EMR cluster 를 생성하여 S3 데이터를 External table 을 정의하여 접근하는 것을 알아보았습니다. EMR Cluster 를 활용하면 매우 다양한 활용 용도로 사용할 수 있습니다. 만약 클러스터를 운용하지 않고 바로바로 Ad-hoc 쿼리를 수행할 수 있다면 그 부분 역시 매우 편리할 것입니다.

그러한 요건에 따라 Amazon Athena 는 표준 SQL 을 사용해 Amazon S3 에 저장된 데이터를 간편하게 분석할 수 있는 대화식 쿼리 서비스입니다. Athena 는 서버리스 서비스이므로 관리할 인프라가 없으며 실행한 쿼리에 대해서만 비용을 지불하면 됩니다.

Athena 는 사용이 쉽습니다. Amazon S3 에 저장된 데이터를 지정하고 스키마를 정의한 후 표준 SQL 을 사용하여 쿼리를 시작하기만 하면 됩니다. 그러면 대부분 결과가 수 초 만에 제공됩니다. Athena 에서는 데이터 분석을 준비하기 위한 복잡한 ETL 작업이 필요 없습니다. 따라서 SQL 을 다룰 수 있는 사람은 누구나 신속하게 대규모 데이터 세트를 분석할 수 있습니다.

JSON 에서 Parquet 형 변환

대용량의 데이터를 분석할 경우 다양한 데이터 포맷을 사용할 수 있습니다. 기본적인 CSV 부터 Avro, Parquet, ORC 등 다양합니다. 중요한 것은 Parquet 과 같은 데이터로 형을 변환하는 것 만으로도 데이터에 대한 접근 및 분석 성능을 매우 높아질 수 있습니다. 앞에서 S3 에 저장된 게이밍 로드 데이터를 JSON 에서 Parquet 포맷으로 변경하여 데이터 분석에서 보다 효과적으로 데이터를 활용할 수 있게 됩니다. 특히 Amazon Athena 의 경우는 Parquet 과 같은 데이터 포맷 사용 시 일반 CSV 와 같은 파일과 비교할 때 성능이 보다 좋습니다.

EMR Cluster Hive 를 활용 형 변환

앞에서 구성한 EMR Cluster 를 이용하면 Hive 를 통해서 바로 JSON 파일이나 다른 파일 등을 Parquet 파일 형식으로 변환할 수 있습니다.

Hive shell 을 실행합니다.

```
$ hive
```

아래 쿼리를 S3 bucket 이름만 변경하여 직접 수행합니다. Parquet 데이터가 생성될 폴더는 myParquet 으로 합니다.

```
CREATE EXTERNAL TABLE parquet_hive (  
  kills int,
```

```

deaths int,
is_winner boolean,
assists int,
char_class string,
char_name string)
STORED AS PARQUET
LOCATION 's3://[YOUR_S3_BUCKET_NAME/myParquet/';

```

INSERT OVERWRITE 를 수행하여 Parquet 형식으로 데이터를 생성합니다.

```

INSERT OVERWRITE TABLE parquet_hive
SELECT
kills,
deaths,
is_winner,
assists,
char_class,
char_name FROM gaming;

```

수행 결과 화면을 아래와 같이 확인할 수 있습니다.

```

hive> INSERT OVERWRITE TABLE parquet_hive
> SELECT
> kills,
> deaths,
> is_winner,
> assists,
> char_class,
> char_name FROM gaming;
Query ID = hadoop_20171016154055_c7141895-b276-426a-8de8-bbc6ffca2d0b
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1508162208449_0012)

```

```

-----
VERTICES   MODE      STATUS TOTAL COMPLETED RUNNING PENDING FAILED KILLED
-----
Map 1 ..... container  SUCCEEDED   1      1      0      0      0      0
-----
VERTICES: 01/01 [=====>>] 100% ELAPSED TIME: 16.42 s
-----
Loading data to table default.parquet_hive

```

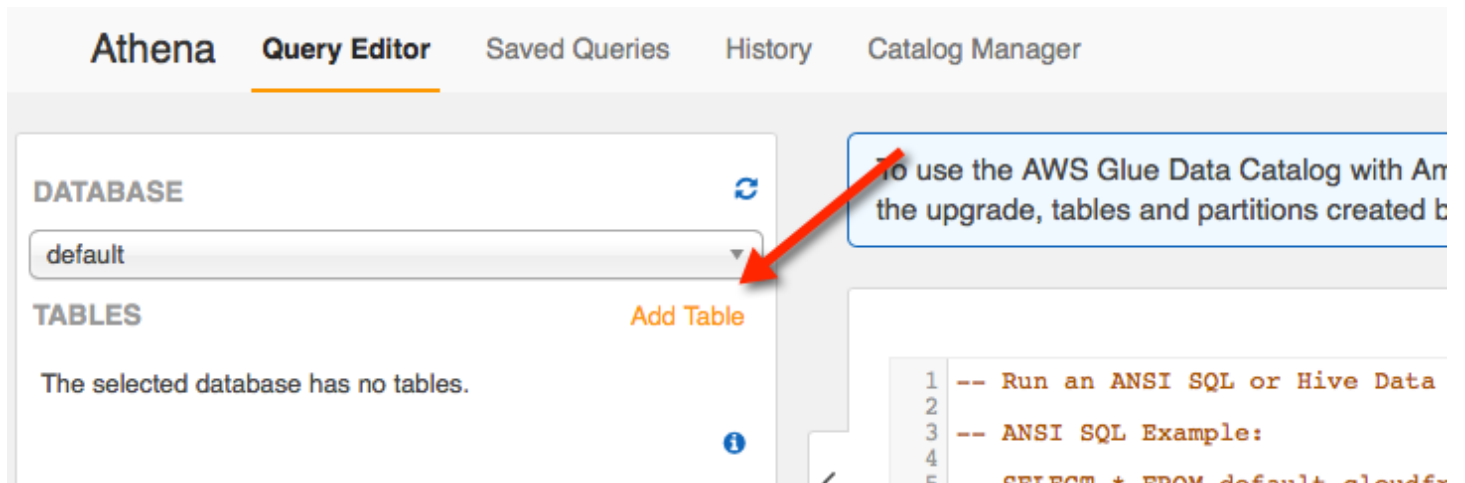
Bash shell 에서 아래 명령을 실행하면 새로운 파일이 실제 생성된 것을 확인할 수 있습니다.

```
$ aws s3 ls s3://[YOUR_S3_BUCKET_NAME/myParquet/
```

Amazon Athena 생성

Web management console 에서 Athena 서비스로 이동합니다. Get started 누릅니다.

Add Table 을 선택합니다.



Database 에 새로운 데이터 베이스 이름을 입력합니다.

Table Name 은 원하는 테이블 이름을 입력합니다.

Location of Input Data Set 이 바로 실제 데이터 파일이 저장되어 있는 S3 위치를 입력합니다. 앞에서 생성했던 Parquet 파일을 이용하기로 합니다. 아래를 참조하여 입력합니다.

s3://[YOUR_S3_BUCKET_NAME]/myParquet/

Databases > Add table

Step 1: Name & Location Step 2: Data Format Step 3: Columns Step 4: Partitions

Database

Choose an existing database or create a new one by selecting "Create new database".

Name of the new database

Table Name

Name of the new table. Table names must be globally unique. Table names tend to corres

Location of Input Data Set

☐ Encrypt

Input the path to the data set you want to process on Amazon S3. For example if your dat
your data is already partitioned, e.g. s3://input-data-set/logs/year=2004/month=12/day=11

Next 를 선택하여 다음으로 진행합니다. Data Format 은 Parquet 을 선택합니다. 그 외에도 다양한 Data Format 을 지원하는 것을 확인할 수 있습니다.

Databases > Add table

Step 1: Name & Location **Step 2: Data Format** Step 3: Columns Step 4: Partitions

Data Format

- ☐ Apache Web Logs
- ☐ CSV
- ☐ TSV
- ☐ Text File with Custom Delimiters
- ☐ JSON
- ☒ Parquet
- ☐ ORC

Next 를 선택하여 다음으로 진행합니다. Columns 이름과 데이터 타입에 대한 입력을 합니다. 한 번에 쉽게 입력하기 위해 아래 Bulk add columns 를 선택하고 아래에 스키마 내용을 입력합니다.

Databases > Add table

Step 1: Name & Location

Step 2: Data Format

Step 3: Columns

Step 4: Partitions

Column Name
Column name must be single words that start with a letter or a digit.

Column type
Type for this column. Certain advanced types (namely, structs) are not exposed in this interface.

Add a column **Bulk add columns**

kills int,
deaths int,
is_winner boolean,
assists int,
char_class string,
char_name string

입력 후 Add 를 눌러 진행하면 입력한 각 컬럼의 정보가 표시되며 잘 못 된 게 없다면, Next 를 눌러 다음으로 진행합니다.

마지막으로 Create table 을 선택합니다.

Query Editor 에서 데이터를 쿼리해 봅니다.

```
select count(*) from gaming_log;
```

```
select * from gaming_log where kills < 5;
```


DATABASE
 mydb

TABLES
 Add Table
 Filter Tables...

- gaming_log
 - kills (int)
 - deaths (int)
 - is_winner (boolean)
 - assists (int)
 - char_class (string)
 - char_name (string)

To use the AWS Glue Data Catalog with Amazon Athena and Amazon Redshift Spectrum, you must upgrade your Athena Data Catalog to the AWS Glue Data Catalog. Without the upgrade, tables and partitions created by AWS Glue cannot be queried with Amazon Athena or Redshift Spectrum. Click [here](#) to upgrade.

```

1 select count(*) from gaming_log;
2 select * from gaming_log where kills < 5;
  
```

Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

Run Query Save As Format Query New Query (Run time: 0.86 seconds, Data scanned: 38.92KB)

Results

	kills	deaths	is_winner	assists	char_class	char_name
1	0	15	false	4	Wziard	Thehilda
2	4	26	false	23	Tank	Cemorilthorn
3	2	2	true	0	Fighter	Carle
4	0	4	true	11	Wizard	Holda Sack
5	2	0	true	1	Wizard	Erilalta
6	4	10	false	8	Wizard	Erilalta

데이터 포맷에 따른 성능 비교

원본 데이터인 JSON 형식의 데이터를 소스로 하여 Add table 을 통해 다른 이름으로 테이블을 추가로 생성합니다. 동일한 내용이 테이블에 생성되나, 성능이 Parquet 이 상대적으로 빠르며, 중요한 것은 Athena 의 사용 비용이 Data scan 을 기반으로 하므로 차이가 크게 됩니다.

위에서 한 것과 동일하게 진행하되 데이터 포맷만 JSON 으로 선택하여 다른 이름의 테이블을 생성합니다.

Databases > Add table

Step 1: Name & Location

Step 2: Data Format

Step 3: Columns

Step 4: Partitions

Database

mydb

Choose an existing database or create a new one by selecting "Create new database".

Table Name

gaming_log_JSON

Name of the new table. Table names must be globally unique. Table names tend to correspond to the directory v stored.

Location of Input Data Set

s3://labs-game-log/input/

☐ Encrypted data set ⓘ

Input the path to the data set you want to process on Amazon S3. For example if your data is stored at s3://input please enter s3://input-data-set/logs/. If your data is already partitioned, e.g. s3://input-data-set/logs/year=2004/i input the base path s3://input-data-set/logs/

External



테이블 생성이 완료되면 두 개의 테이블을 생성된 것을 확인할 수 있으며, 여러 쿼리를 통하여 쿼리 성능과 Data scanned 양을 보시면 Columnar 방식의 Parquet 의 장점을 알 수 있습니다.

Athena

Query Editor

Saved Queries

History

Catalog Manager

DATABASE

mydb

TABLES

Add Table

Filter Tables...

gaming_log

gaming_log_json

To use the AWS Glue Data Catalog with Amazon Athena and Amazon Redshift Spectrum, you must upgrade to the latest version of the AWS Glue Data Catalog. Without the upgrade, tables and partitions created by AWS Glue cannot be queried by Amazon Athena or Amazon Redshift Spectrum. Click [here](#) to upgrade.

```

1 CREATE EXTERNAL TABLE IF NOT EXISTS mydb.gaming_log_JSON (
2   `kills` int,
3   `deaths` int,
4   `is_winner` boolean,
5   `assists` int,
6   `char_class` string,
7   `char_name` string
8 )
9 ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
10 WITH SERDEPROPERTIES (
11   'serialization.format' = '1'
12 ) LOCATION 's3://labs-game-log/input/'
13 TBLPROPERTIES ('has_encrypted_data'='false');
14
15 select count(*) from gaming_log_json;
16 select count(*) from gaming_log;

```

Run Query

Save As

Format Query

New Query

(Run time: 0.42 seconds, Data scanned: 0KB)

Lab 8. Amazon Kinesis Analytics 를 이용한 실시간 데이터 처리

앞의 랩을 통해 Kinesis stream 을 활용하고 Kinesis 에 저장된 데이터를 다시 Database 나 파일에 저장하여 쿼리를 할 수 있는 부분을 진행해 보았습니다. 실시간 데이터를 바로 처리할 수 있는 요건도 중요한 Data Lake 의 요건 중 하나입니다.

그러한 요건을 위해서 Amazon Kinesis Analytics 는 새로운 프로그래밍 언어 또는 처리 프레임워크를 배울 필요 없이 표준 SQL 을 통해 실시간으로 스트리밍 데이터를 처리할 수 있는 가장 쉬운 방법입니다. Amazon Kinesis Analytics 를 사용하면 SQL 을 사용하여 스트리밍 데이터를 쿼리하거나 전체 스트리밍 애플리케이션을 구축할 수 있으므로 수행해야 할 작업을 정확하게 파악하고 비즈니스 및 고객 요구 사항에 적절하게 대응할 수 있습니다. Amazon Kinesis Analytics 는 쿼리를 지속적으로 실행하는 데 필요한 모든 작업을 처리하며 수신 데이터의 볼륨과 처리량 속도에 맞춰 자동으로 확장됩니다. Amazon Kinesis Analytics 에서는 쿼리가 사용한 리소스에 대한 비용만 지불하면 됩니다.

Kinesis Analytics application 생성하기

Web management console 에서 Kinesis 로 이동하여 Analytics 를 선택합니다.

Create application 을 선택하고, Application name 을 입력합니다.

Create application 을 선택합니다.

Create application

Application name*

Description

To enable interactivity with your data during configuration of your streaming application you will be prompted to run your application. **Usage based charges apply.** [See Kinesis Analytics pricing.](#)

* Required

[Cancel](#)

[Create application](#)

다음으로 Source 를 지정합니다. Connect to a source 를 선택합니다.

✓ Successfully created Application **gaming_log_realtime**
Next, choose **Connect to a source**.

gaming_log_realtime

Application ARN: arn:aws:kinesisanalytics:us-west-1:123456789012:application/gaming_log_realtime

Application version ID: 1 ⓘ



Source

Connect to a Kinesis stream, a Firehose delivery stream, or create and connect to a demo stream. [Learn more](#)

[Connect to a source](#)

이미 만들어진 Kinesis stream 가 보여집니다. 앞에서 생성한 Kinesis stream 을 선택합니다.

Connect to source

Choose from your Kinesis streams and Firehose delivery streams, or quickly configure a demo Kinesis stream that can be used to explore Kinesis Analytics.

Select a stream (5)

[Configure a new stream](#)

labs

Stream name	Stream type
labs-game-stream	Kinesis stream

Discovery schema 를 선택하면 자동으로 Schema 를 생성하여 보여줍니다.

Schema discovery can generate a schema using recent records from the source. Schema column names are the same as in the source, but they contain special characters, repeated column names, or reserved keywords. [Learn more](#)

[Edit schema](#) [Retry schema discovery](#)

Save and continue 를 선택하여 진행합니다. 잠시 후 Kinesis Analytics application 이 생성 완료 됩니다.

Go to SQL editor 를 선택합니다.

gaming_log_realtime

Application status: READY

Application ARN: arn:aws:kinesisanalytics:us-west-2:806506827877:application/gaming_log_realtime

Application version ID: 2 ⓘ



Source

	Source	In-application stream	ID ⓘ	Record pre-processing ⓘ
	Kinesis stream labs-game-stream	SOURCE_SQL_STREAM_001	2.1	Disabled



Real-time analytics

Author your own SQL queries or add SQL from templates to easily analyze your source data. [Learn more](#)[Go to SQL editor](#)

매우 다양한 Query template 을 지원하고 있습니다. Kinesis Analytics 는 SQL 을 활용하여 원하는 실시간 분석 쿼리를 작성할 수 있습니다. Table 대신 Stream 이란 개념을 지원하고 있어 문서의 참조가 필요합니다.

<http://docs.aws.amazon.com/kinesisanalytics/latest/dev/streaming-sql-concepts.html>

아래 예제는 계속 simulator.py 가 실행되면서 Kinesis stream 에 데이터가 계속 쌓이는 과정에서 10 초마다 쌓이는 데이터를 읽어와서 평균 kill 수를 구하는 예제를 만든 쿼리 입니다.

Amazon Kinesis

Streams

Firehose

Analytics

Kinesis Analytics applications > gaming > SQL editor

Real-time analytics

Add and run SQL queries to continuously analyze source data in real-time. Then, optionally, connect the in-ap

[Add SQL from templates](#)

```

1 CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" (
2   "avg_kills" INTEGER
3 );
4
5 CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"
6 SELECT STREAM "avg_kills"
7 FROM (
8   SELECT STREAM
9     "char_name",
10    AVG("kills") OVER W1 as "avg_kills"
11   FROM "SOURCE_SQL_STREAM_001"
12   WINDOW W1 AS (PARTITION BY "char_name" RANGE INTERVAL '10' SECOND PRECEDING)
13 )
14 WHERE ABS("avg_kills") > 10;
```

쿼리

```
CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" (
  "avg_kills" INTEGER
);
```

```
CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"
SELECT STREAM "avg_kills"
FROM (
  SELECT STREAM
    "char_name",
    AVG("kills") OVER W1 as "avg_kills"
  FROM "SOURCE_SQL_STREAM_001"
  WINDOW W1 AS (PARTITION BY "char_name" RANGE INTERVAL '10' SECOND PRECEDING)
)
WHERE ABS("avg_kills") > 10;
```

실행하면 쿼리에 문제가 없으면, 지속적으로 쿼리가 실행되면서 아래와 같이 결과를 보여줍니다.

Source data
Real-time analytics
Destination

Application status: RUNNIN

In-application streams:

DESTINATION_SQL_STREAM
error_stream

Pause results
New results are added every 2-10 seconds. The results below are sampled. ⓘ
☐ Scroll to bottom when new results arrive.

ROWTIME	avg_kills
2017-10-17 13:03:50.982	13
2017-10-17 13:04:01.035	22
2017-10-17 13:04:08.006	11
2017-10-17 13:04:12.012	13
2017-10-17 13:04:12.012	17
2017-10-17 13:04:15.016	16
2017-10-17 13:04:23.068	17
2017-10-17 13:04:25.031	17
2017-10-17 13:04:25.031	11
2017-10-17 13:04:26.032	12