

실리콘밸리에서 날아온 데이터베이스

2. MySQL 소개/설치와 예제 데이터 소개

keeyonghan@hotmail.com

한기용

Harmonize, Inc

Contents

1. MySQL 소개
2. 클라우드/AWS 소개
3. MySQL 설치 - Docker
4. MySQL 설치 - AWS RDS
5. MySQL DDL과 예제 테이블 소개



MySQL 소개

가장 인기있는 관계형 데이터베이스인 MySQL 소개

◆ MySQL 역사

- ❖ 1995년 스웨덴 회사였던 MySQL AB에 의해 개발된 관계형 데이터베이스
 - 오픈소스로 시작됨
 - My는 개발자 중 한 사람의 딸의 이름이었음
- ❖ 2008년 썬 마이크로시스템스가 MySQL AB를 \$1B을 주고 인수
- ❖ 2009년 오라클이 썬을 인수하면서 MySQL이 유료화 여부가 쟁점이 됨
- ❖ 2010년 MySQL의 처음 개발자였던 Monty가 MySQL과 호환이 되는 MariaDB라는 오픈소스 개발



◆ MySQL 종류와 버전

❖ MariaDB

- 오픈 소스로 무료
- MySQL 5.5에 기반해서 개발됨
- MySQL과 인터페이스는 동일하나 성능은 더 좋음

❖ MySQL

- 두 가지 종류가 존재
 - MySQL Community Server: 오픈소스로 무료. 우리가 사용해볼 버전
 - MySQL Enterprise Server: 유료 버전으로 다양한 플러그인 제공
- 최신 버전은 8.0

◆ MySQL 특징 (1)

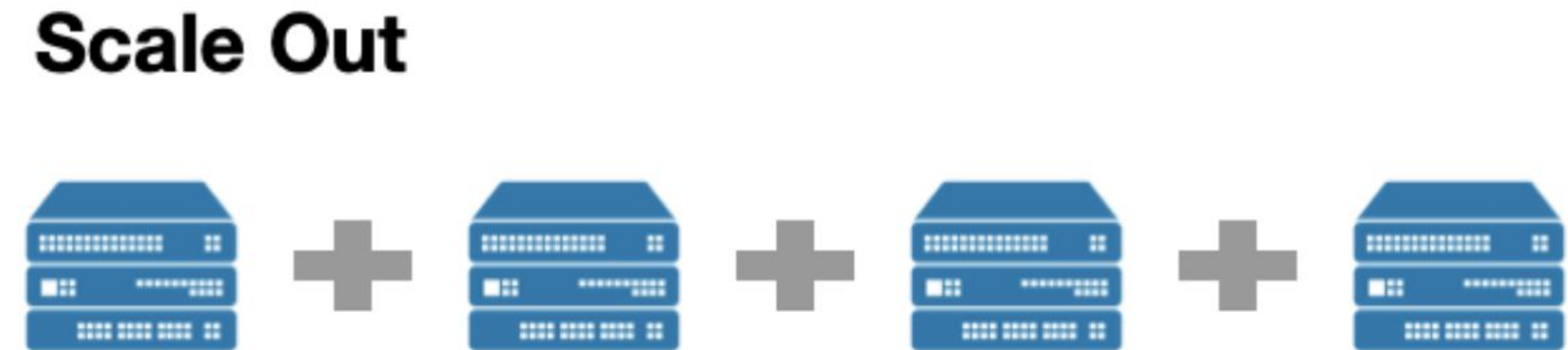
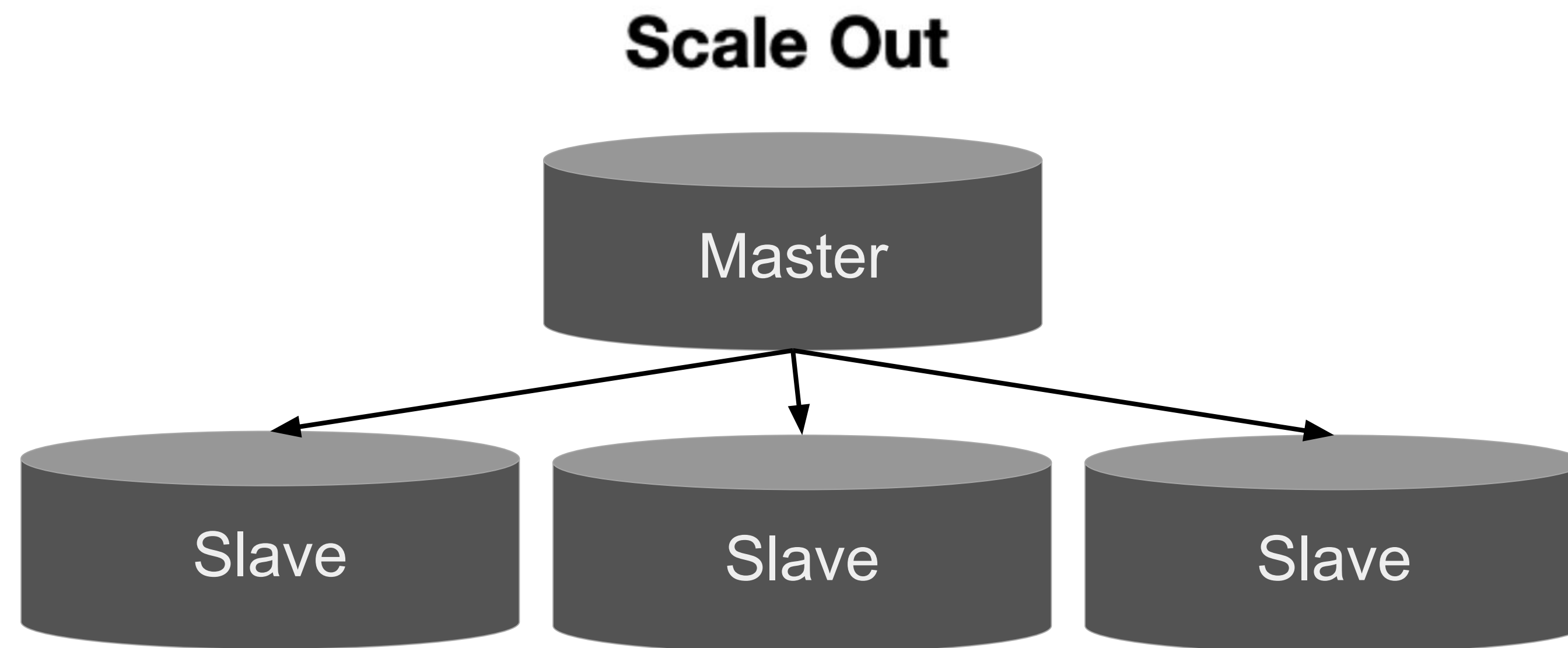
- ❖ 한동안 웹개발 표준 기술 스택 중의 하나였음
 - LAMP: Linux, Apache, MySQL, PHP
- ❖ 지금도 Postgres와 함께 가장 널리 쓰이는 프로덕션용 관계형 데이터베이스
- ❖ 용량 증대 (Scaling) 방식 1
 - Scale-Up: 서버에 CPU와 Memory 추가



◆ MySQL 특징 (2)

❖ 용량 증대 (Scaling) 방식 2

- Scale-Out: Master-Slave 구성
- 일반적으로 클러스터 구성을 이야기하나 MySQL은 이를 지원하지 못함





클라우드/AWS 소개

클라우드 컴퓨팅 소개
가장 큰 클라우드 업체인 AWS 소개

◆ 클라우드의 정의

- ❖ 컴퓨팅 자원(하드웨어, 소프트웨어 등등)을 네트워크를 통해 서비스 형태로 제공
- ❖ 키워드:
 - “No Provisioning”
 - “Pay As You Go”
- ❖ 자원(예를 들면 서버)을 필요한만큼 (거의) 실시간으로 할당하여 사용한만큼 지불
 - 탄력적으로 필요한만큼의 자원을 유지하는 것이 중요

◆ 클라우드 컴퓨팅이 없었다면?

- ❖ 서버/네트워크/스토리지 구매와 설정등을 직접 수행해야 함
- ❖ 데이터센터 공간을 직접 확보 (Co-location)
 - 확장이 필요한 경우 공간을 먼저 더 확보해야함
- ❖ 그 공간에 서버를 구매하여 설치하고 네트워크 설정
 - 보통 서버를 구매해서 설치하는데 적어도 두세달은 걸림
- ❖ 또한 **Peak time**을 기준으로 **Capacity planning**을 해야 함!
 - 놓고 있는 자원들이 높게 되는 현상 발생
- ❖ 직접 운영비용 vs. 클라우드 비용
 - 기회비용!

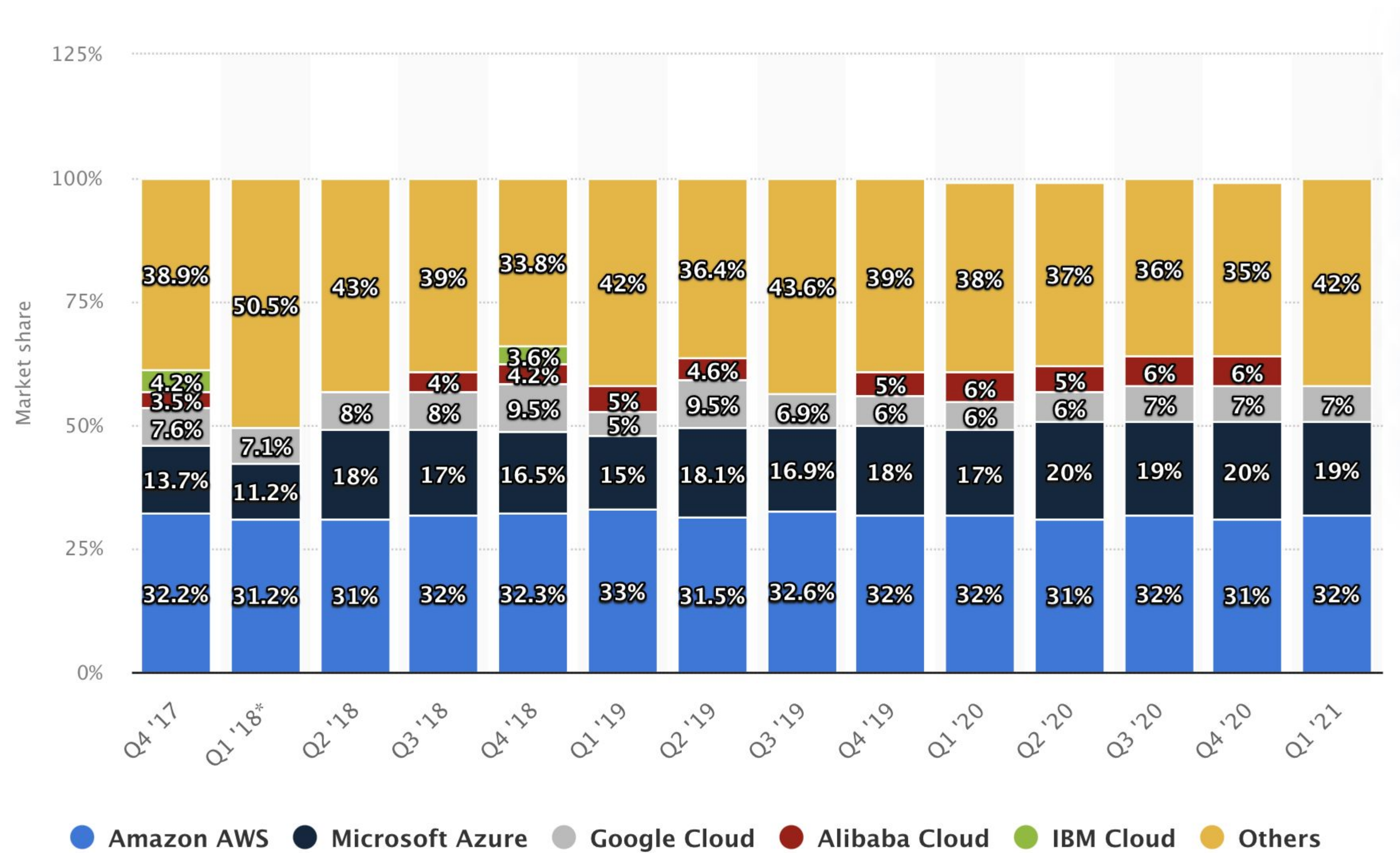
◆ 클라우드 컴퓨팅의 장점

- ❖ 초기 투자 비용이 크게 줄어듦
 - CAPEX (Capital Expenditure) vs. OPEX (Operating Expense)
- ❖ 리소스 준비를 위한 대기시간 대폭 감소
 - Shorter Time to Market
- ❖ 노는 리소스 제거로 비용 감소
- ❖ 글로벌 확장 용이
- ❖ 소프트웨어 개발 시간 단축
 - Managed Service (SaaS) 이용

◆ AWS 소개

- ❖ 가장 큰 클라우드 컴퓨팅 서비스 업체.
- ❖ 2002년 아마존의 상품데이터를 **API**로 제공하면서 시작
 - 현재 100여개의 서비스를 전세계 15개의 지역에서 제공.
 - 대부분의 서비스들이 오픈소스 프로젝트들을 기반으로 함.
 - 최근 들어 **ML/AI** 관련 서비스들도 내놓기 시작
- ❖ 사용고객
 - 다수의 상장업체들과 많은 국내 업체들도 사용시작 (서울 리전)
- ❖ 다양한 종류의 소프트웨어/플랫폼 서비스를 제공.
 - **AWS**의 서비스만으로 쉽게 온라인서비스 생성.
 - 뒤에서 일부 서비스를 따로 설명.

◆글로벌 클라우드 시장 점유율



◆ AWS의 매출/이익 추이



Q1 2021 매출액: \$13.1B. 아마존 전체 매출의 12%


◆ AWS Regions


Regions	Name
US East (Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northwest-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
China (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1
EU (Frankfurt)	eu-central-1
EU (Ireland)	eu-west-1
EU (Paris)	eu-west-3
EU (Stockholm)	eu-north-1
EU (London)	eu-west-2
South America (Sao Paulo)	sa-east-1
AWS GovCloud (US, US-East)	us-gov-west-1, us-gov-east-1


◆ AWS 서비스들

Search services


Group

 **Compute**


EC2
EC2 Container Service
Lightsail 
Elastic Beanstalk
Lambda
Batch

 **Storage**


S3
EFS
Glacier
Storage Gateway

 **Database**


RDS
DynamoDB
ElastiCache
Redshift

 **Networking & Content Delivery**


VPC
CloudFront
Direct Connect
Route 53

 **Developer Tools**


CodeCommit
CodeBuild
CodeDeploy
CodePipeline


 **Management Tools**


CloudWatch
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Trusted Advisor
Managed Services
Application Discovery Service

 **Security, Identity & Compliance**


IAM
Inspector
Certificate Manager
Directory Service
WAF & Shield
Compliance Reports

 **Analytics**


Athena
EMR
CloudSearch
Elasticsearch Service
Kinesis
Data Pipeline
QuickSight 

 **Artificial Intelligence**


Lex
Polly
Rekognition
Machine Learning

 **Internet Of Things**


AWS IoT

 **Game Development**


GameLift

 **Mobile Services**


Mobile Hub
Cognito
Device Farm


 **Application Services**


Step Functions
SWF
API Gateway
Elastic Transcoder

 **Messaging**

SQS
SNS
SES

 **Business Productivity**

WorkDocs
WorkMail
Amazon Chime 

 **Desktop & App Streaming**

WorkSpaces
AppStream 2.0

◆ EC2 - Elastic Cloud Compute (1)

❖ AWS의 서버 호스팅 서비스.

- 리눅스 혹은 윈도우 서버를 론치하고 로그인 가능 (구글앱엔진과의 가장 큰 차이점).
- 가상 서버들이라 전용서버에 비해 성능이 떨어짐.
- Bare-metal 서버도 제공하기 시작

❖ 다양한 종류의 서버 타입 제공

- <http://aws.amazon.com/ec2/>
- 예를 들어 미국 동부에서 스몰타입(t2.small)의 무료 리눅스 서버를 하나 할당시
 - 시간당 2.3 센트의 비용지불.
 - 2GB 메모리, 1 가상코어, 160GB 하드디스크
 - 2012년에는 8.5 센트였음
 - 타입별 지역별 가격을 알고 싶다면 여기를 방문

Incoming network bandwidth는 고평이지만 outgoing은 오히려

◆ EC2 - Elastic Cloud Compute (1)

❖ 세 종류의 구매 옵션

구매 옵션	설명
On-Demand	시간당 비용을 지불되며 가장 흔히 사용하는 옵션
Reserved	1년이나 3년간 사용을 보장하고 1/3 정도에서 40% 디스카운트를 받는 옵션
Spot Instance	일종의 경매방식으로 놓고 있는 리소스들을 보다 싼 비용으로 사용할 수 있는 옵션

◆ S3 – Simple Storage Service (1)

- ❖ <http://aws.amazon.com/s3/>
- ❖ 아마존이 제공하는 대용량 클라우드 스토리지 서비스
- ❖ S3는 데이터 저장관리를 위해 계층적 구조를 제공
- ❖ 글로벌 네임스페이스를 제공하기 때문에 톱레벨 디렉토리 이름 선정에 주의.
- ❖ S3에서는 디렉토리를 버킷(Bucket)이라고 부름
- ❖ 버킷이나 파일별로 액세스 컨트롤 가능

◆ S3 – Simple Storage Service (2)

- ❖ <https://aws.amazon.com/ko/s3/pricing/>
- ❖ Low cost. 1TB per month:
 - Standard storage: \$23
 - Infrequent Access storage: \$12.5
 - SLA가 다름
 - Glacier storage: \$4

◆ Database Services

- ❖ RDS (Relational Database Service)
 - **MySQL/MariaDB**, PostgreSQL, Aurora
 - Oracle, MS SQL Server
- ❖ DynamoDB
- ❖ Redshift
- ❖ ElastiCache
- ❖ Neptune (Graph database)
- ❖ ElasticSearch
- ❖ MongoDB

◆ AI & ML Services

❖ SageMaker

- Deep Learning and Machine Learning end-to-end framework

❖ Lex

- Conversational Interface (Chatbot service)

❖ Polly

- Text to Speech Engine

❖ Rekognition

- Image Recognition Service

◆ 기타 중요 서비스

❖ Amazon Alexa

- Amazon's Voice Bot Platform

❖ Amazon Connect

- Amazon's Contact Center Solution
- 콜센터 구현이 아주 쉬워짐

❖ Lambda

- Event-driven, serverless computing engine
- 서비스 구현을 위해서 EC2를 론치할 필요가 없음
- Google Cloud에는 Cloud Function이란 이름으로 존재
- Azure에는 Azure Function이란 이름으로 존재



MySQL 설치 - Docker

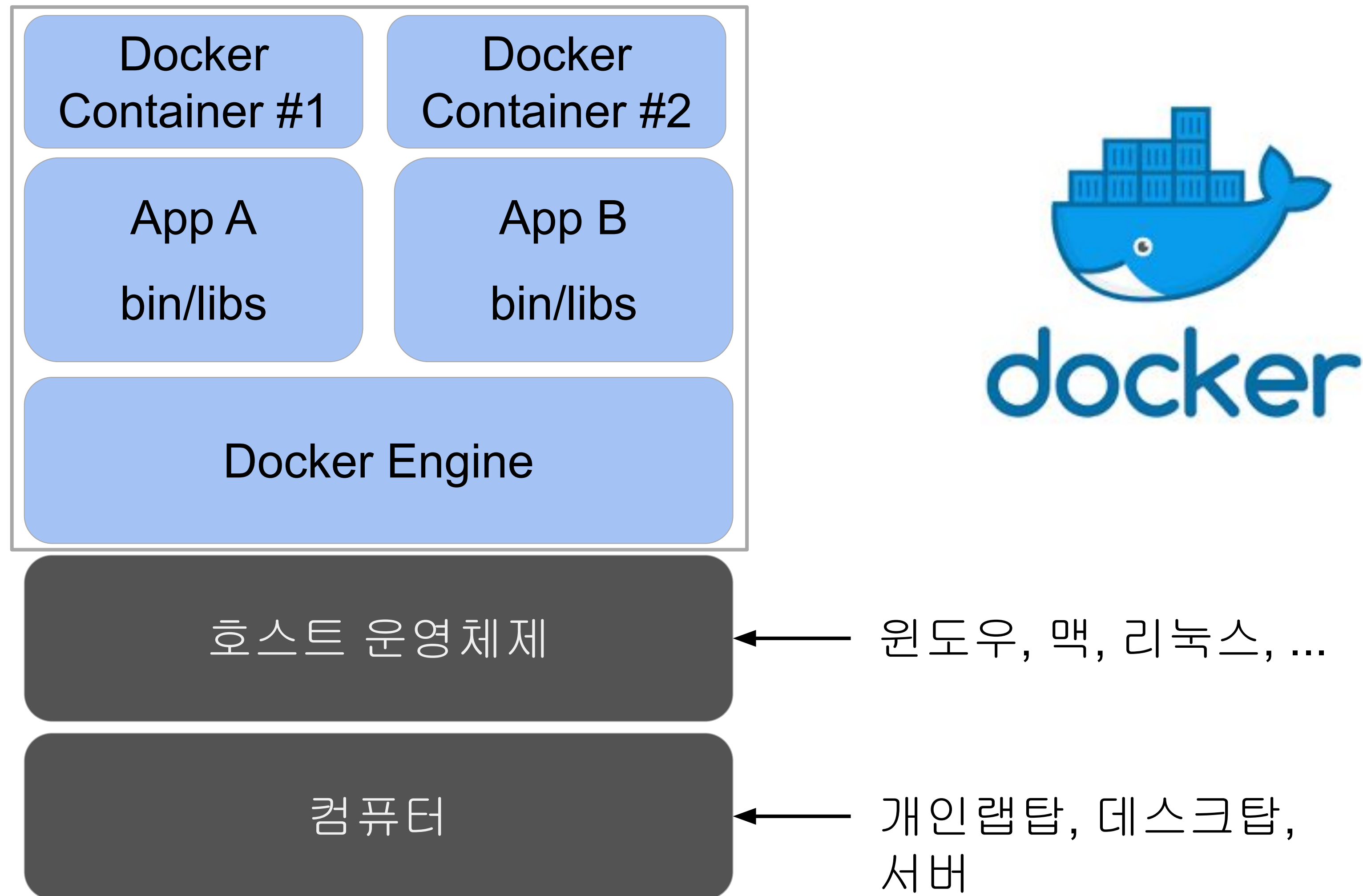
초간단 Docker 소개
Docker로 MySQL 설치방법 소개

◆ Docker란 무엇인가?

- ❖ 예를 들어 **MySQL**을 다른 **OS**에서 설치하려면 다양한 변수가 존재
 - 즉 설치 과정이 **OS**와 **OS**의 버전에 따라 달라지게 됨
 - 다양한 다수의 다른 소프트웨어들의 설치가 동반되는 것이 일반적임
- ❖ **Docker**는 특정 프로그램과 (그 프로그램을 실행하는데) 필요한 기타 소프트웨어들을 하나의 패키지로 만듬으로써 해당 프로그램의 개발과 사용을 도와주는 오픈소스 플랫폼
 - 이 패키지를 먼저 파일 시스템 형태로 만드는데 이를 **Docker Image**라고 함
 - 이 Image는 다른 이들과 공유가능
 - Docker Image 공유소를 Docker Registry(Docker Hub)라고 부름
 - Docker Image를 실행시킨 것을 Docker Container라고 부르며 응용프로그램에 해당

◆ Docker 구조

❖ Virtualization vs. Containerization



Docker Registry
(Docker Hub)

- 다양한 Docker Image들의 집합소 (이 역시 오픈소스)
- Docker Image의 인스턴스가 결국 Docker Container
 - docker images
 - docker ps

◆ Docker 기반으로 MySQL 설치해보기

- ❖ 먼저 해당 컴퓨터에 Docker Engine을 설치
 - <https://docs.docker.com/desktop/>
- ❖ 그 다음에 MySQL Docker Image를 다운로드
 - <https://dev.mysql.com/doc/mysql-installation-excerpt/8.0/en/docker-mysql-getting-started.html>
- ❖ 다운로드된 MySQL Docker Container를 실행
- ❖ Mac 기반 데모

◆ 맥 기반 Docker Engine 설치 (1)

❖ <https://docs.docker.com/desktop/>

Download and install

Docker Desktop is available for Mac and Windows. For download information, system requirements, and installation instructions, see:

- [Install Docker Desktop on Mac](#)
- [Install Docker Desktop on Windows](#)

Install Docker Desktop on Mac

Estimated reading time: 5 minutes

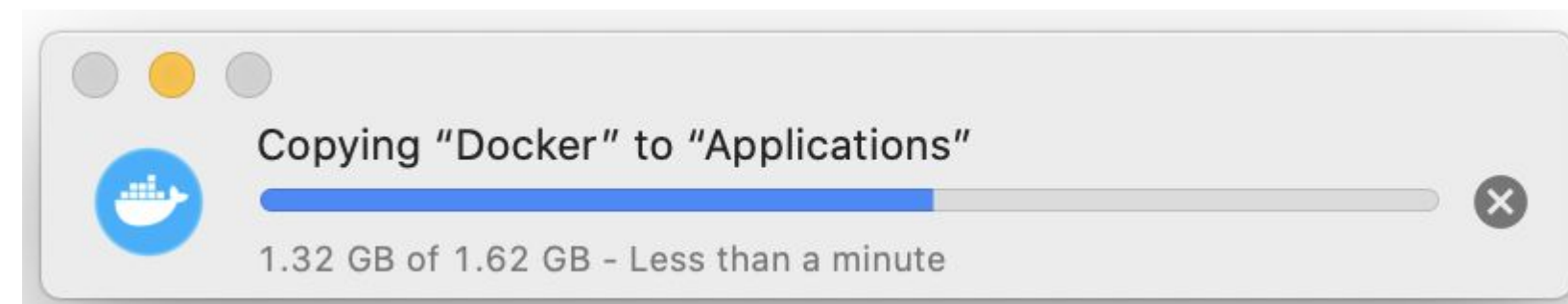
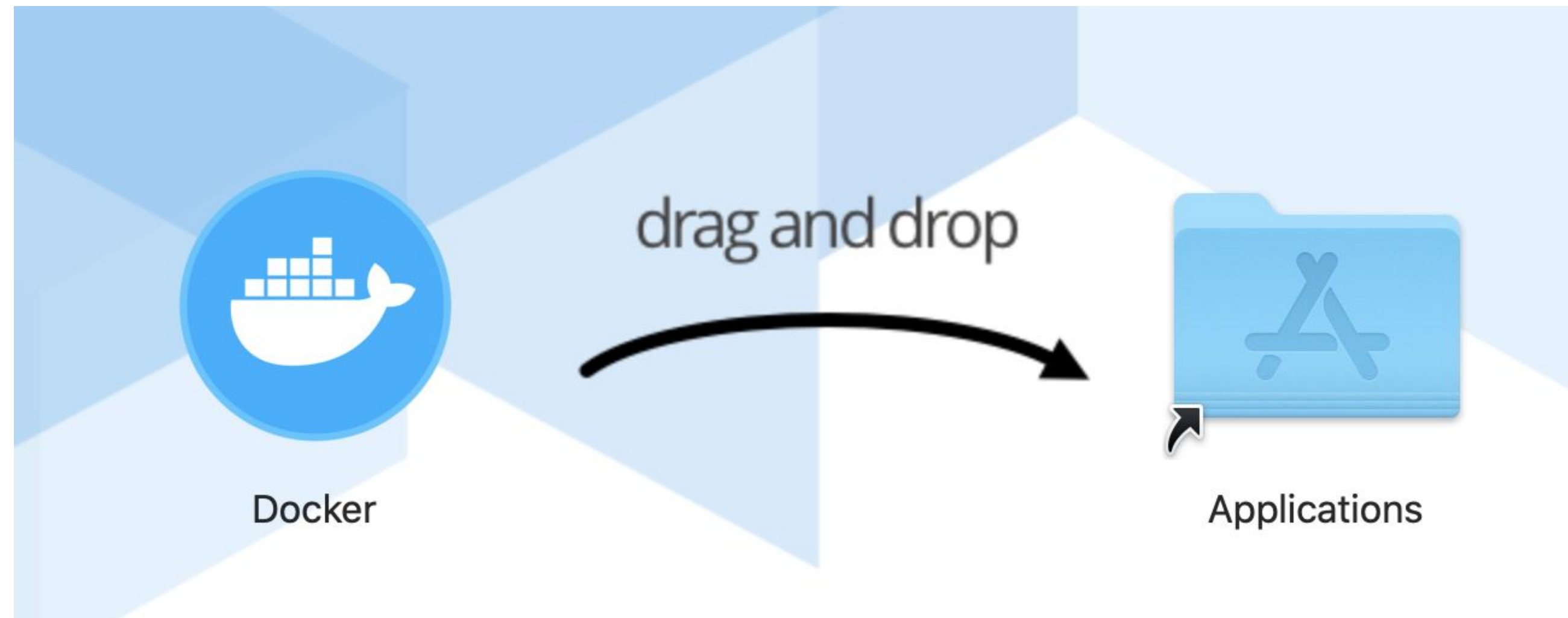
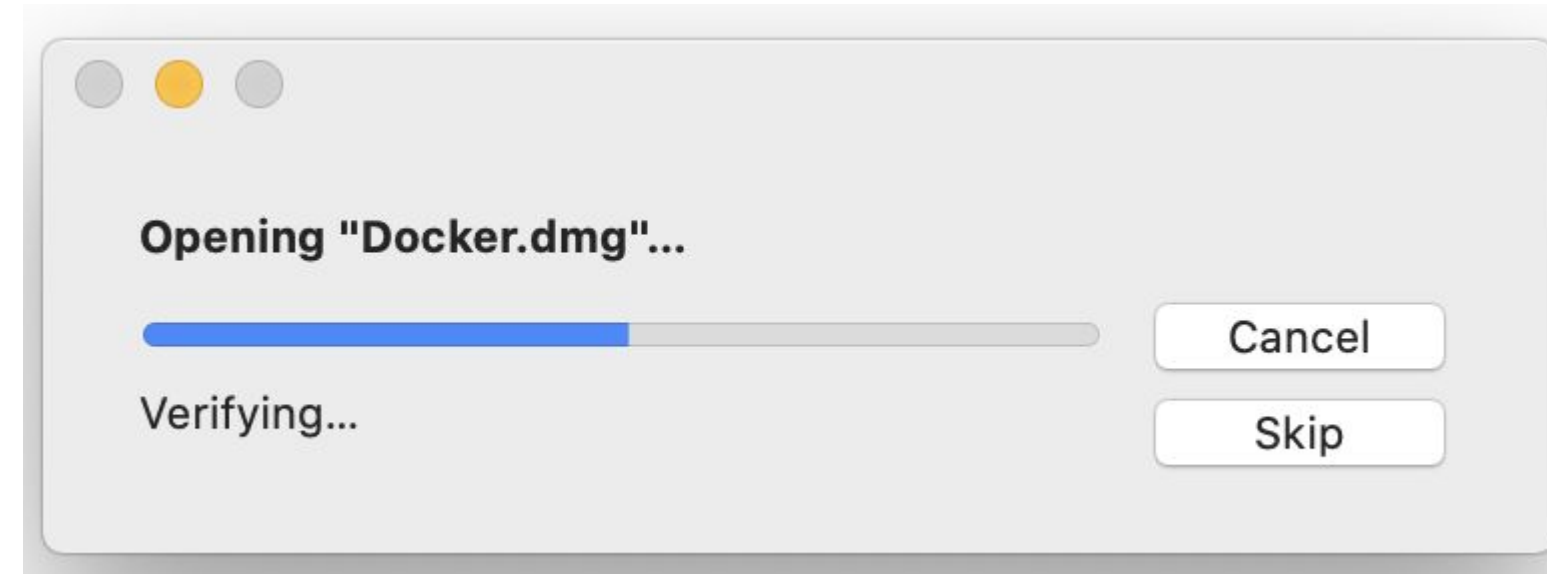
Welcome to Docker Desktop for Mac. This page contains information about Docker Desktop for Mac system requirements, download URLs, installation instructions, and automatic updates.

Download Docker Desktop for Mac:

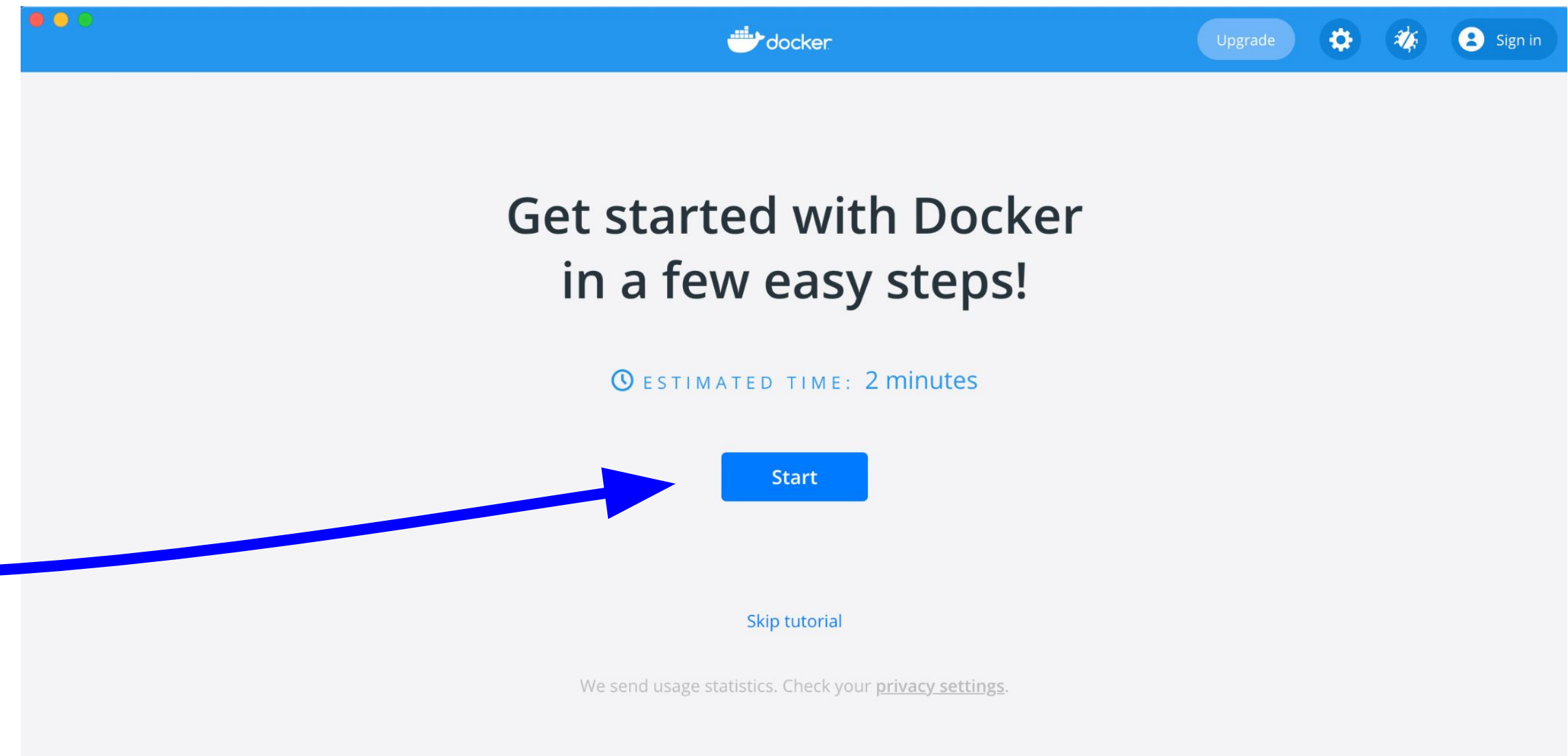
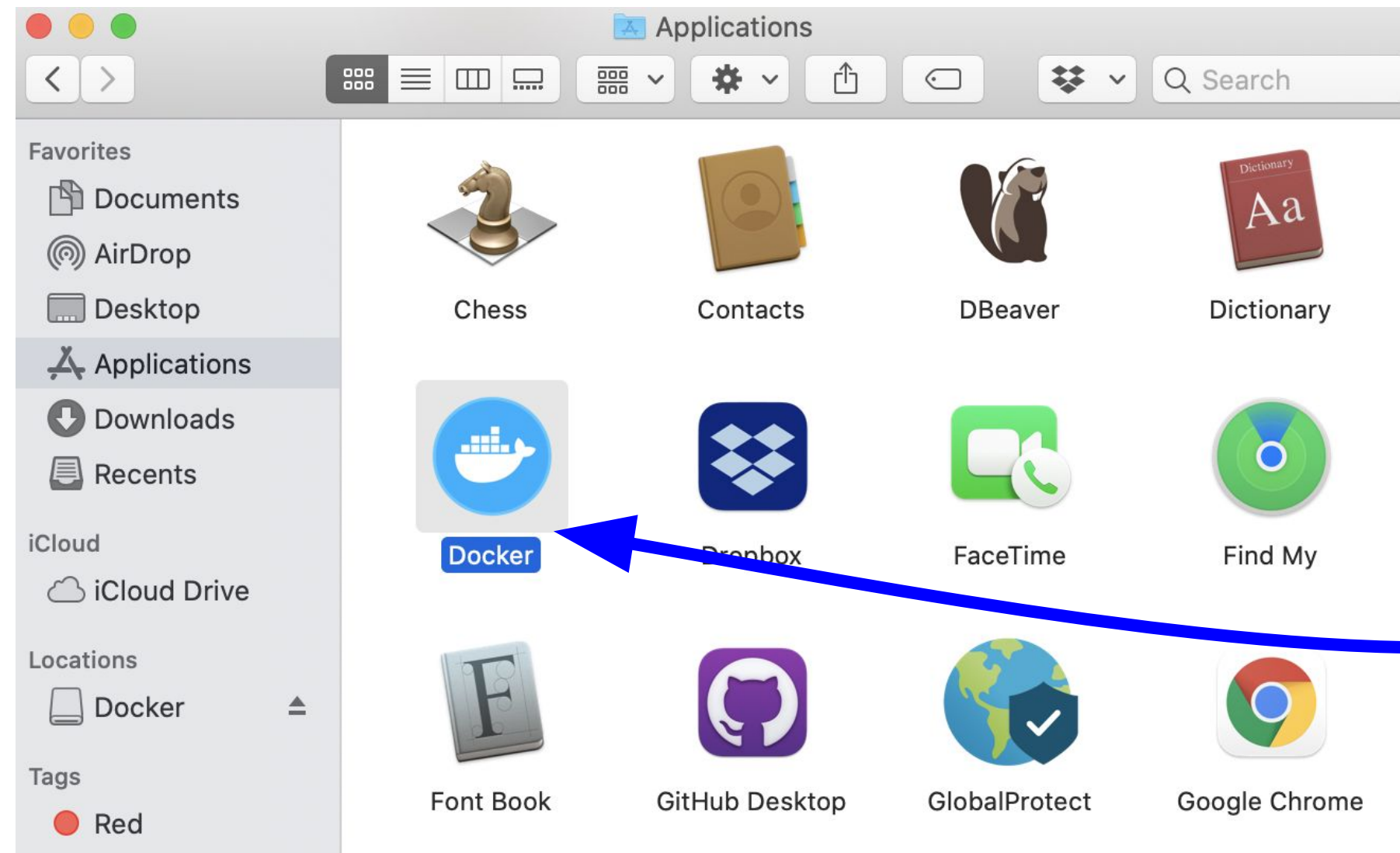
Mac with Intel chip

Mac with Apple chip

◆ 맥 기반 Docker Engine 설치 (2)



◆ 맥 기반 Docker Engine 설치 (3)



◆ 맥 기반 MySQL Docker 실행 (1)

- ❖ 먼저 terminal 프로그램 실행
- ❖ MySQL docker image를 다운로드
 - `docker pull mysql/mysql-server:8.0`
- ❖ 다운로드받은 이미지로 Docker container 실행
 - `docker run --name=mysql_container --restart on-failure -d mysql/mysql-server:8.0`
- ❖ MySQL root 계정의 패스워드 찾기
 - `docker logs mysql_container 2>&1 | grep GENERATED`
- ❖ 마지막으로 MySQL shell 실행하기
 - `docker exec -it mysql_container mysql -uroot -p`

◆ 맥 기반 MySQL Docker 실행 (2)

```
$ docker pull mysql/mysql-server:8.0
```

```
8.0: Pulling from mysql/mysql-server
```

```
...
```

```
38b3da6a86f7: Pull complete
```

```
Digest: sha256:5241f7de0483a70f5856da995fea98904cfce8f1c51734b7f3836c1663eead17
```

```
Status: Downloaded newer image for mysql/mysql-server:8.0
```

```
docker.io/mysql/mysql-server:8.0
```

```
$ docker run --name=mysql_container --restart on-failure -d mysql/mysql-server:8.0
```

```
6137dc6ea2fe283dd007b2e55a1cc2c37b0c89dc45eaa3b58085bae7daa799ed
```

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6137dc6ea2fe	mysql/mysql-server:8.0	"/entrypoint.sh my..."	8 secs ago	Up 7 secs (health: starting)	3306/tcp, 33060-33061/tcp	mysql_container

```
$ docker logs mysql_container 2>&1 | grep GENERATED
```

```
[Entrypoint] GENERATED ROOT PASSWORD: [PASSWORD]
```

```
$ docker exec -it mysql_container mysql -uroot -p
```

```
Enter password:
```


◆ 맥 기반 MySQL Docker 실행 (3)

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 10

Server version: 8.0.26

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>



MySQL 설치 - AWS RDS

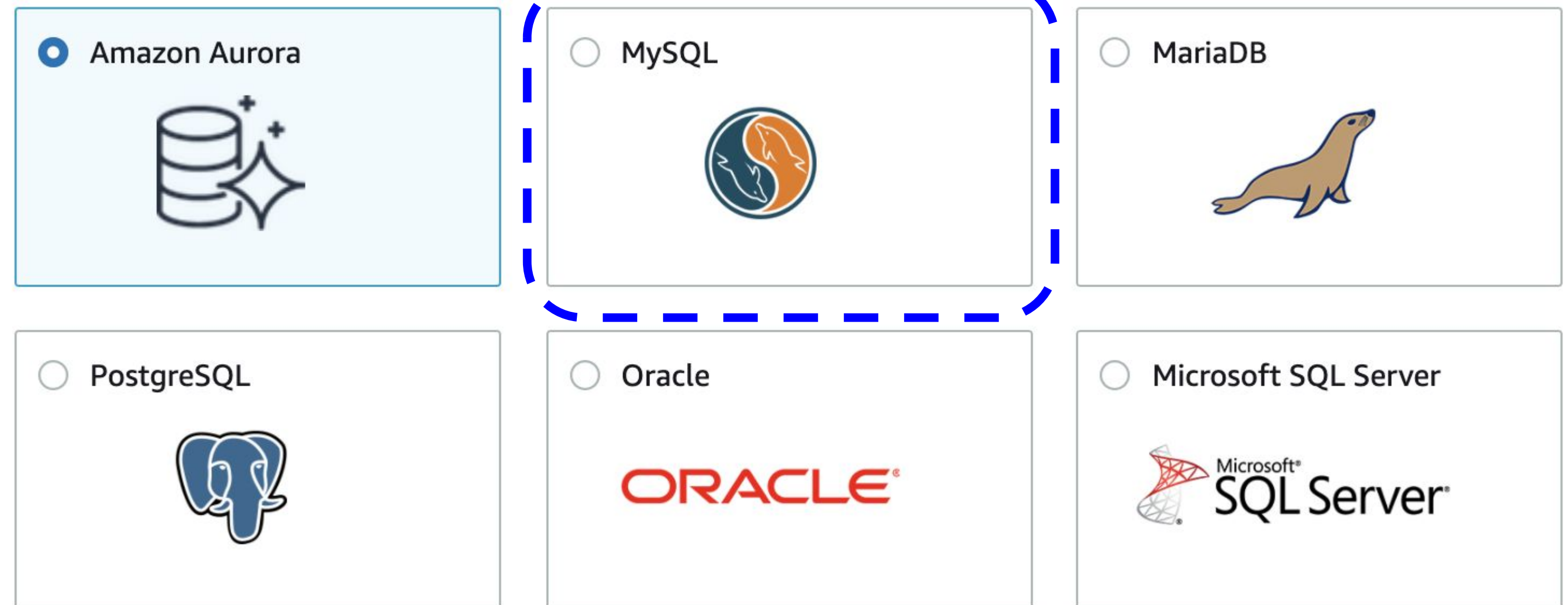
AWS RDS 소개
AWS RDS로 MySQL 설치 데모

◆ AWS RDS란?

❖ AWS가 제공하는 다양한 관계형 데이터베이스 서비스

- MySQL/MariaDB
- PostgreSQL
- Oracle
- Amazon Aurora
- Microsoft SQL Server

Engine type [Info](#)



◆ AWS RDS를 이용해 MySQL 설치

❖ 설치 데모

❖ 뒤 실습에서는 여기서 만든 AWS RDS를 사용할 예정

- 공용 계정을 사용할 예정이며 계정 정보는 내일 실습시 공유

MySQL DDL과 예제 데이터 소개

SQL DDL (Data Definition Language) 소개
앞으로 실습에서 사용할 예제 데이터 소개

◆ 관계형 데이터베이스 예제 - 웹서비스 사용자/세션 정보 (1)

- ❖ 사용자 ID: 보통 웹서비스에서는 등록된 사용자마다 부여하는 유일한 ID
- ❖ 세션 ID: 세션마다 부여되는 ID
 - 세션: 사용자의 방문을 논리적인 단위로 나눈 것
 - 사용자가 외부 링크(보통 광고)를 타고 오거나 직접 방문해서 올 경우 세션을 생성
 - 사용자가 방문 후 30분간 **interaction**이 없다가 뭔가를 하는 경우 새로 세션을 생성
 - 즉 하나의 사용자는 여러 개의 세션을 가질 수 있음
 - 보통 세션의 경우 세션을 만들어낸 접점(경유지)를 채널이란 이름으로 기록해둠
 - 마케팅 관련 기여도 분석을 위함
 - 또한 세션이 생긴 시간도 기록

◆ 관계형 데이터베이스 예제 - 웹서비스 사용자/세션 정보 (2)

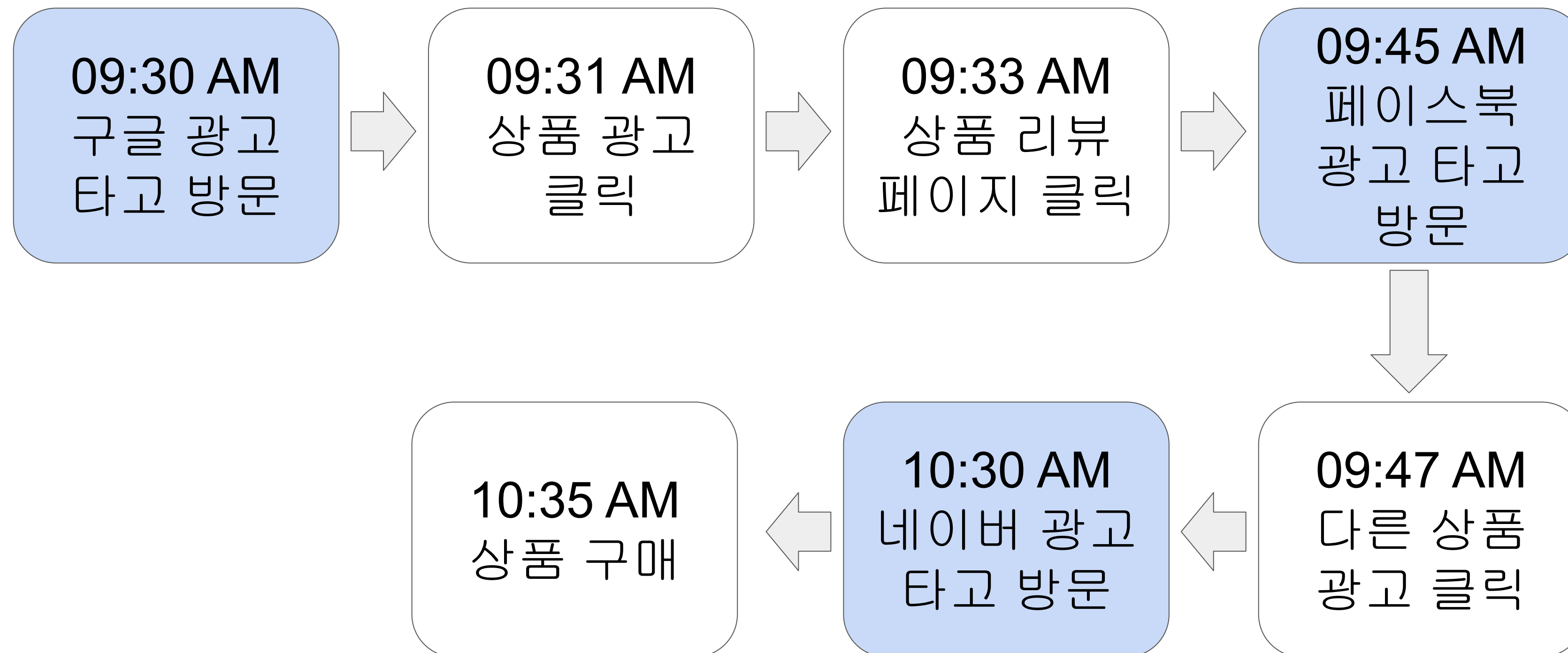
❖ 이 정보를 기반으로 다양한 데이터 분석과 지표 설정이 가능

- 마케팅 관련, 사용자 트래픽 관련
- DAU, WAU, MAU 등의 일주월별 Active User 차트
- Marketing Channel Attribution 분석
 - 어느 채널에 광고를 하는 것이 가장 효과적인가?

◆ 관계형 데이터베이스 예제 - 웹서비스 사용자/세션 정보 (3)

❖ 사용자 ID 100번: 총 3개의 세션(파란 배경)을 갖는 예제

- 세션 1: 구글 키워드 광고로 시작한 세션
- 세션 2: 페이스북 광고를 통해 생긴 세션
- 세션 3: 네이버 광고를 통해 생긴 세션



◆ 관계형 데이터베이스 예제: 우리가 저장하고 싶은 데이터

컬럼	타입	부가 설명	예
id	int	세션 ID	101
user_id	int	사용자 ID	201
created	timestamp	세션 생성시간	2021-05-01 11:32:59
channel	varchar(64)	채널 (접점) 이름	Facebook

session 정보

◆ 관계형 데이터베이스 예제: 데이터베이스와 테이블 요약 (1)

prod 데이터베이스

컬럼	타입
id	int
user_id	int
created	timestamp
channel_id	int

session 테이블

컬럼	타입
id	int
channel	varchar(32)

channel 테이블

◆ 테이블 필드의 중요 속성 (1)

❖ PRIMARY KEY

- 테이블에서 레코드의 유일성을 정의하는 필드: 그것이 **primary key**가 되어야 함
 - 예를 들면 사용자 테이블에서 “이메일” 혹은 주민등록번호
- **Composite primary key**: **primary key**가 두개 혹은 그 이상의 필드로 정의되는 경우
- **Primary key**로 지정된 필드가 있는 경우 데이터베이스단에서 중복된 값을 갖는 레코드가 생기는 것을 방지함: **primary key uniqueness constraint**

❖ Foreign key

- 테이블의 특정 필드가 다른 테이블의 필드에서 오는 값을 갖는 경우

◆ 테이블 필드의 중요 속성 (2)

❖ NOT NULL

- 필드의 값이 항상 존재해야하는 경우

❖ DEFAULT value

- 필드에 값이 주어지지 않은 경우 기본값을 정의해줌
- timestamp 타입: **CURRENT_TIMESTAMP**를 사용하면 현재 시간으로 설정됨

◆ 관계형 데이터베이스 예제: 데이터베이스와 테이블 요약 (2)

prod 데이터베이스

컬럼	타입	속성
id	int	NOT NULL AUTO_INCREMENT PRIMARY KEY
user_id	int	NOT NULL
created	timestamp	NOT NULL DEFAULT CURRENT_TIMESTAMP
channel_id	int	NOT NULL FOREIGN KEY

session 테이블

컬럼	타입	속성
id	int	NOT NULL AUTO_INCREMENT PRIMARY KEY
channel	varchar(32)	NOT NULL

channel 테이블

◆ 관계형 데이터베이스 예제: channel 테이블 내용

id	channel
1	Instagram
2	Naver
3	Youtube
4	Google
5	Facebook
6	Tiktok
7	Unknown

channel 테이블

◆ 관계형 데이터베이스 예제: channel table DDL

```
CREATE TABLE channel (  
    id int not null auto_increment primary key,  
    channel varchar(32) not null  
);
```

```
CREATE TABLE channel (  
    id int not null auto_increment,  
    channel varchar(32) not null,  
    primary key(id)  
);
```

id	channel
1	Instagram
2	Naver
3	Youtube
4	Google
5	Facebook
6	Tiktok
7	Unknown

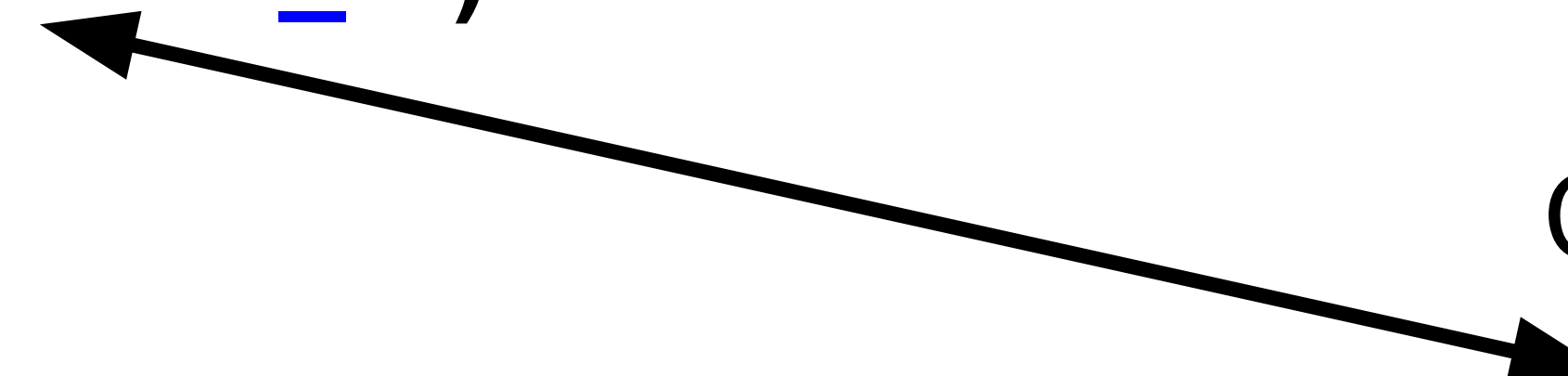
◆ 관계형 데이터베이스 예제: session 테이블 내용

id	user_id	created	channel_id
1	779	2019-05-01 00:36:00	5
2	230	2019-05-01 02:53:49	3
3	369	2019-05-01 12:18:27	6
4	248	2019-05-01 13:41:29	5
5	676	2019-05-01 14:17:54	2
6	40	2019-05-01 14:42:50	3
7	468	2019-05-01 15:08:16	1
8	69	2019-05-01 15:20:27	4

session 테이블

◆ 관계형 데이터베이스 예제: session DDL

```
CREATE TABLE session (  
  id int not null auto_increment primary key,  
  user_id int not null,  
  created timestamp not null default current_timestamp,  
  channel_id int not null,  
  foreign key(channel_id) references channel(id)  
);
```



```
CREATE TABLE channel (  
  id int auto_increment primary  
  key,  
  channel varchar(32) not null  
);
```

강의 요약

MySQL

클라우드/AWS

MySQL 설치 - Docker & AWS RDS

MySQL DDL과 예제 테이블

