

# 고성능 서비스를 위한 Redis 활용 및 아키텍처 강의

# 강의 소개

## 1 Orientation

## Redis 강의 소개와 학습 목표

# 1.

## 강의 소개

### 학습 목차

#### Chap1 | Redis 소개와 특징

Redis 처음 만나기

#### Chap2 | NoSQL과 Redis

DB의 분류와 NoSQL DB로서의 Redis 특성 파악하기

#### Chap3 | Redis 설치와 실습

Redis를 설치해보고 간단히 조작해보기

#### Chap4 | Redis Data Type의 이해

Redis가 제공하는 다양한 데이터 타입들을 알아보기

#### Chap5 | Redis 개발 실습

Java를 이용해 Redis 프로그래밍해보기

#### Chap6 | 분산 환경에서의 세션 스토어 만들기

여러 대의 서버에서 Redis를 사용해 세션을 공유해보기

#### Chap7 | 서비스 속도를 높이는 캐시 레이어 만들기

Redis를 이용한 서비스 성능 개선하기 실습

#### Chap8 | 게임 리더 보드 만들기

Redis를 이용해 쉽게 구현하는 랭킹 관리

#### Chap9 | Pub/Sub을 이용해 손쉽게 채팅방 기능

구현하기

Redis가 제공하는 기능을 이용해 채팅 서비스를 만들어보기

#### Chap10 | 안정성과 가용성을 위한 클러스터

서비스 안정성을 떠받치는 분산 환경과 클러스터 이해하기

#### Chap11 | 클러스터 구성 실습

실무에 꼭 필요한 클러스터를 구성 실습

#### Chap13 | Redis 성능 튜닝

Redis 효율적으로 사용하기

#### Chap14 | Redis Streams를 이용한 Event-Driven

아키텍처

MSA에서 사용되는 Event-Driven 아키텍처 맛보기

#### Chap12 | Redis의 백업과 장애 복구

장애에도 문제없는 Redis 백업 구성

#### Chap15 | 글로벌 서비스를 위한 Active-Active

아키텍처

분산 환경에서도 속도 빠른 서비스 만들기

## Redis 강의 소개와 학습 목표

# 1.

## 강의 소개

### 왜 Redis를 공부해야 하는가?

- 가장 손쉽게 사용할 수 있는 In-memory 저장소
- 높은 성능
- 다양한 활용성
- 현대적인 서버 구조에서 세션 관리나 캐시는 빠질 수 없는 구성 요소

## Redis 강의 소개와 학습 목표

1.

강의 소개

### 학습 목표

- **Redis**가 무엇인지 특성 중심으로 설명할 수 있게 된다.
- 세션 스토어, 캐시와 같은 **Redis**의 주요한 활용 방법을 익히고 구현할 수 있게 된다.
- 클러스터 구성, 백업과 장애 복구 등 실제 **Redis** 운영에 필요한 지식을 갖춘다.
- **Redis**를 활용하는 다양한 어플리케이션 아키텍처들을 익힌다.

# Redis 소개와 특징

## 1 In-memory DB로서의 Redis

## Redis 강의 소개와 학습 목표

1.

In-memory  
DB로서의 Redis

### Redis의 정의

- Remote Dictionary Server
- Storage: 데이터 저장소 (데이터 관점)
- Database: 전통적인 DBMS의 역할을 수행 (영속성 관점)
- Middleware: 어플리케이션이 이용할 수 있는 유용한 기능을 제공하는 소프트웨어

### Redis로 할 수 있는 것?

- 아주 빠른 데이터 저장소로 활용
- 분산된 서버들간의 커뮤니케이션(동기화, 작업 분할 등)
- 내장된 자료구조를 활용한 기능 구현

## In-memory DB로서의 Redis

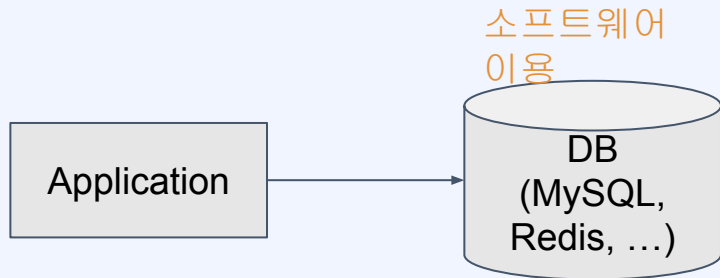
1.

In-memory  
DB로서의 Redis

## DB, Database, DBMS?

- 데이터를 읽고 쓸 수 있는 기능을 제공하는 소프트웨어
- 어플리케이션이 데이터 저장을 간단히 처리할 수 있도록 해줌
- 관심사의 분리, 계층화  
직접 파일에 쓰기

주소록.txt	
이름	주소
-----	
김철철	경기
박선선	부산





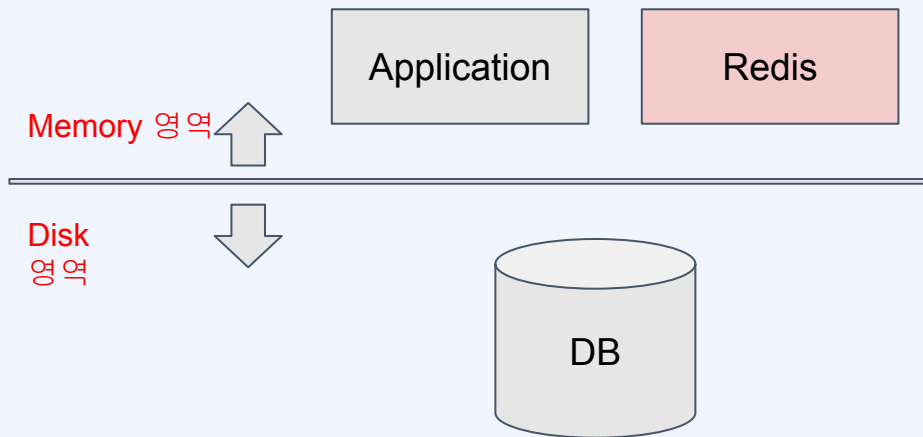
## In-memory DB로서의 Redis

1.

In-memory  
DB로서의 Redis

### In-memory DB?

- 데이터를 디스크에 저장하지 않음.
- 휘발성인 **RAM**에 저장.
- 빠른 속도



## In-memory DB로서의 Redis

1.

In-memory  
DB로서의 RedisMemory와 Disk의 속도  
차이

	Read	Write	비교
HDD	130 MB/s	120 MB/s	
SSD	500 MB/s	450 MB/s	HDD보다 3~4배 빠름
RAM	20,000 MB/s	20,000 MB/s	SSD보다 40배 빠름

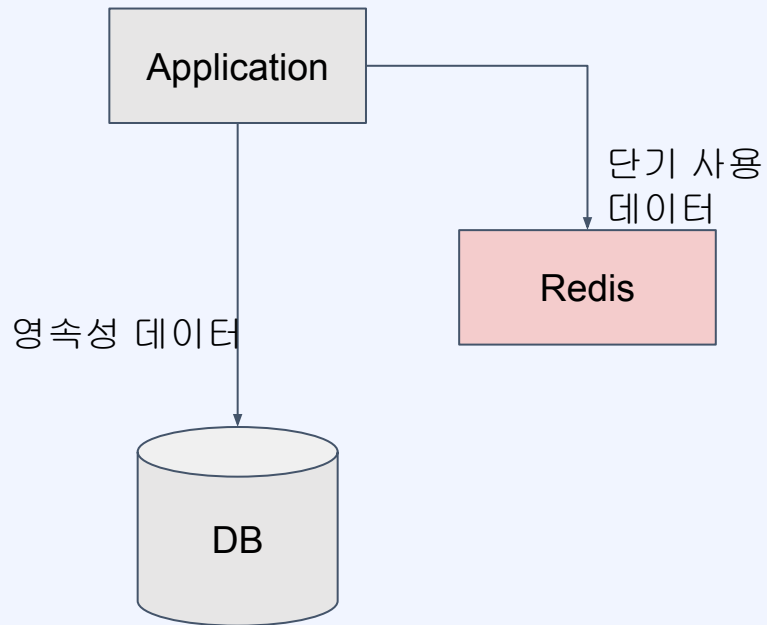
## In-memory DB로서의 Redis

1.

In-memory  
DB로서의 Redis

빠른 속도와 휘발성의 절충

- 용도에 맞게 DB와 Redis를 사용
- 혼합해서 사용(Cache)
- Redis의 영속성 확보(백업 등)



# Redis 소개와 특징

## 2 Key-value store로서의 Redis

## Key-value store로서의 Redis

2.

Key-value  
store로서의 Redis

### 데이터 저장소의 구조

- 프로그램 언어에서의 데이터 구조(Array, List, Map, ...)
- DB의 데이터 모델 관점에서의 구조  
(네트워크 모델, 계층형 모델(Tree), 관계형 모델, ...)

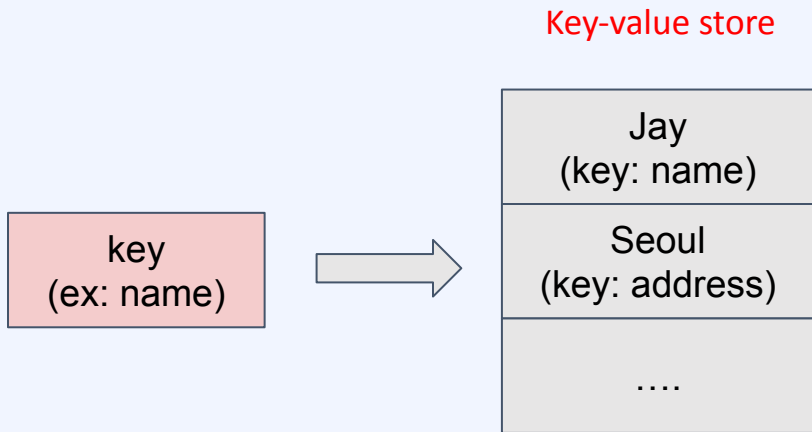
## Key-value store로서의 Redis

2.

Key-value  
store로서의 Redis

### Key-value store?

- 특정 값을 **key**로 해서 그와 연관된 데이터를 **value**로 저장(Map과 같음)
- 가장 단순한 데이터 저장 방식
- 단순한 만큼 빠르고 성능이 좋음



## Key-value store로서의 Redis

2.

Key-value  
store로서의 Redis

### Key-value 구조의 장점

- 단순성에서 오는 쉬운 구현과 사용성
- Hash를 이용해 값을 바로 읽으므로 속도가 빠름(추가 연산이 필요 없음)
- 분산 환경에서의 수평적 확장성

### Key-value 구조의 단점

- Key를 통해서만 값을 읽을 수 있음
- 범위 검색 등의 복잡한 질의가 불가능

## Key-value store로서의 Redis

2.

Key-value  
store로서의 Redis

### Key-value 스토어의 활용

- 언어의 자료구조(Java의 HashMap 등)
- NoSQL DB(Redis, Riak, AWS DynamoDB)
- 단순한 구조의 데이터로 높은 성능과 확장성이 필요할 때 사용



# Redis 소개와 특징

## 3 Redis의 등장과 활용

## Redis의 등장과 활용

### 3.

#### Redis의 등장과 활용

### Redis 히스토리

- 2009년 살바토레 산필리포(Salvatore Sanfilippo)가 개발
- 오픈소스화 이후 캐시 기술로서 대중적인 인기를 얻음
- BSD 라이선스(개작/배포 시 저작권자 이름과 **BSD** 라이선스 표시)
- 2015년부터 **Redis Labs**가 지원(엔터프라이즈 모듈은 별도 라이선스로 판매)
- 다양한 언어에서 **Redis**를 지원하고 많은 라이브러리가 존재  
(ActionScript, C, C++, C#, Clojure, Dart, Erlang, Go, Haskell, Java, Node.js, Lua, Objective-C, PHP, Perl, Python, Ruby, Scala 등)

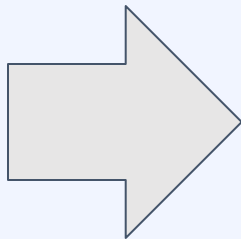
## Redis의 등장과 활용

3.

Redis의 등장과  
활용

### Redis 활용

- 서버를 개발하며 **Redis**를 사용하지 않는 기업을 찾기가  
어려울 정도
- Session Store
- Cache
- Limit Rater
- Job Queue



별도로 구현해야 했던 공통적인  
기능들을 **Redis**를 사용하면  
쉽게 구현 가능하다.