

Managing Redis

Redis Administration

Redis setup tips

- 주로 Linux에서 실행, 배포, 테스트 되고 있다
- `maxmemory` 옵션을 설정 할 때 여유를 갖고 설정하자
 - Reids의 오버헤드
 - 메모리 단편화 발생
 - EX) Memory: 10GB, maxmemory: 8GB, 9GB
- Replication
 - `persistence` 비활성화 하더라도 Replication을 사용한다면, RDB 파일을 저장한다.
 - Replication을 사용한다면, master의 `persistence` 를 활성화 하거나, master가 crash 되었을 때 자동으로 재시작 하지 않게 해야한다.
 - master가 빈 데이터로 재시작 할 경우, replicas도 비어진다. (모두 master를 따라가려함)
- Security
 - redis 기본적으로 인증을 사용하지 않고, 모든 네트워크를 수신한다. → 외부 인터넷에 노출되지 않게 하자

Running Redis on EC2

- PV 기반 인스턴스 대신 HVM 인스턴스 사용
 - PV (Para Virtualization): 반가상화
 - HVM (Hardward Virtual Machine): 완전가상화

Upgrading or restarting a Redis instance without downtime

- Redis는 `long-running process` 으로 설계되었다.

- `CONFIG SET` 명령어를 통해 재시작 필요 없이 config 변경 가능
<https://redis.io/commands/config-set>
 - Redis 프로세스 업그레이드 등 COFIG 명령으로 지원하지 않을 경우 재시작 필요
 - 새로운 Redis 인스턴스를 현재 Redis의 replica로 설정
 - master, slave의 키가 같은지 `INFO` 로 확인
 - `CONFIG SET slave-read-only no` 로 replica에 쓰기 허용
 - 모든 클라이언트가 replica를 바라 보게 한다. (`CLIENT PAUSE` 사용)
 - master가 더 이상 쿼리를 수신하지 않으면, replica를 master로 승격한 후, 기존 master 종료
-

Security

Security model

- 기본적으로 Redis는 Client를 신뢰
 - 외부 인터넷에 노출하는것은 좋지 않다.

Network Security

- Redis를 사용하는 Server에서만 Access 가능해야한다.
 - 단일 컴퓨터에 Redis가 존재할 경우, 루프백을 사용해 Access 가능

Protected mode

- 이미 많은 Redis가 외부 인터넷에 노출되어 있다.
 - 3.2.0 부터는 default config를 Password 없이 설정하였을 경우 Protected mode 가 실행된다.
- Protected mode일 경우 루프백으로만 쿼리 실행 가능, 나머지는 오류를 내려줌

Authentication

- 기존에는 redis.config 파일을 통해 Password 설정 가능

- Password가 활성화 되었을 경우, 인증되지 않은 모든 클라이언트 쿼리 거부
- Redis 6.0 이후에는 ACL 지원
 - ACL(Access Control List): 사용자 마다 실행 할 수 있는 명령을 지정

TLS support

- TLS는 선택적으로 지원한다고 합니다 π
-

High availability with Redis Sentinel

Redis Sentinel을 통해 Cluster를 사용하지 않을 때도, 고가용성을 제공할 수 있다.

Sentinel 기능

- Monitoring
 - master / slave 모니터링
- Notification
 - 인스턴스의 문제가 있을 경우 외부에 알림
- Automatic failover
 - master 장애 발생 시 replica → master 승격 진행
- Configuration provider
 - client에게 알려줄 master의 주소를 관리

Sentinel as a distributed system

Sentinel은 자체적으로 여러 Sentinel 프로세스의 협력을 고려하여 설계되었습니다.

- 여러 Sentinel 함께 실패 감지 → 오탐지 확률 낮아짐
- Sentinel을 단일 실패 지점으로 만들지 않을 수 있다

Sentinel quick start

- Sentinel 실행 (config 파일 필수)
 1. `redis-sentinel /path/to/sentinel.conf`
 2. `redis-server /path/to/sentinel.conf --sentinel`
- 최소 3개 이상의 Sentinel 인스턴스 필요
- Sentinel monitor
 - `sentinel monitor <master-group-name> <ip> <port> <quorum>`
 - quorum: master가 실패 시, 복구 절차 시작을 위해 동의해야하는 최소의 Sentinel 의 수
 - failover 감지 이후, Leader Sentinel이 선출되어 권한을 받아 장애 조치 실행

Other Sentinel options

- `down-after-milliseconds` : 인스턴스가 Down 되었다고 판단하는 시간
- `parallel-syncs` : 동시에 재구성 할 수 있는 master의 수

그 외 실습

- 현재 마스터 주소: `SENTINEL get-master-addr-by-name mymaster`
- 장애 조치 테스트: `redis-cli -p 6379 DEBUG sleep 30`
- Runtime 환경에서 Sentinel 설정 변경
 - `SENTINEL SET objects-cache-master down-after-milliseconds 1000`
 - `SENTINEL SET objects-cache-master quorum 5`
- Sentinel 추가, 제거
 - 추가: Sentinel 추가 시, 모든 Sentinel과 연결 되고, master - replica list도 알게 된다.
 - 제거: 다른 Sentinel 들은 제거되는 것을 모르기 때문에 다른 모든 Sentinel 초기화 작업 필요 (`SENTINEL RESET *`)

Replication

- 기본 구성: Leader - Follower (master - replica)
- master의 data에 변화를 주는 모든 커맨드는 replica로 전달
 - 연결이 끊어 질 경우, 끊어진 동안 발생한 command만 전달하여 부분 동기화 시도
 - 부분 동기화 불가능 시. 전체 동기화 실행
- asynchronous replication 사용
 - Client에서 `WAIT` Command를 통해 synchronous replication 요청 가능
→ 지정된 수의 복사본이 존재하는지만 알 수 있음 == 강한 일관성을 가진 CP 시스템으로

Important facts about Redis replication

- replica는 다른 replica에 연결 될 수 있다.
 - 계층형 구조가 가능함
- replication은 non-blocking으로 동작한다.
 - replica 동기화 과정에도 master는 쿼리 처리 가능

Safety of replication when master has persistence turned off

- master의 persistence 세팅을 키거나, master의 자동 재시작을 막아야 한다.

How Redis replication works

- master는 replication ID (dataset을 위한 일종의 random string), replication stream의 바이트마다 증가하는 offset을 가지고 있다.
- replication ID와 offset을 쌍으로 dataset의 버전을 식별한다.
 - 동기화 할 때, master에게 replication ID와 지금까지 처리한 offset을 전달
 - master가 처리한 offset 이후의 command 전달
 - master가 buffer하고 있는 log가 부족하거나 replication ID가 사라졌을 경우, 전체 동기화

Replication ID explained

- replication ID는 dataset의 history를 의미
- replica가 master로 승격 될 경우, 새로운 replication ID 생성
 - 동일한 dataset의 경우 동일한 replication ID 갖는다.

Read-only replica

- replica는 기본적으로 read only로 설정된다.
 - read only 상태일 경우, write 커맨드 거부