QUNAR Redis运维实践

赵应钢
@nettedfish

关于我

□DBA@去哪儿网

□MySQL、Redis、Hbase

□DevOps

Agenda

- Redis简介
- Redis运维自动化
- Redis高可用
- Our patches for Redis

Redis简介

- KV NoSQL
- In-memory can persistent
- Open source
- Born on 2009. @antirez
- 2.8.6 now avaliable
- 3.0.0 support cluster

sponsored by:

vmware®

> “Everything runs from memory in Web 2.0”
>
> Evan Weaver, Twitter, March 2009

Redis简介

- Storage & Cache
  - string、hash、list、set、sorted set
  - persistence: AOF vs RDB
  - high performance
  - PUB/SUB

- High Avaliability
  - replication
  - redis-sentinel
  - redis cluster

Redis运维自动化

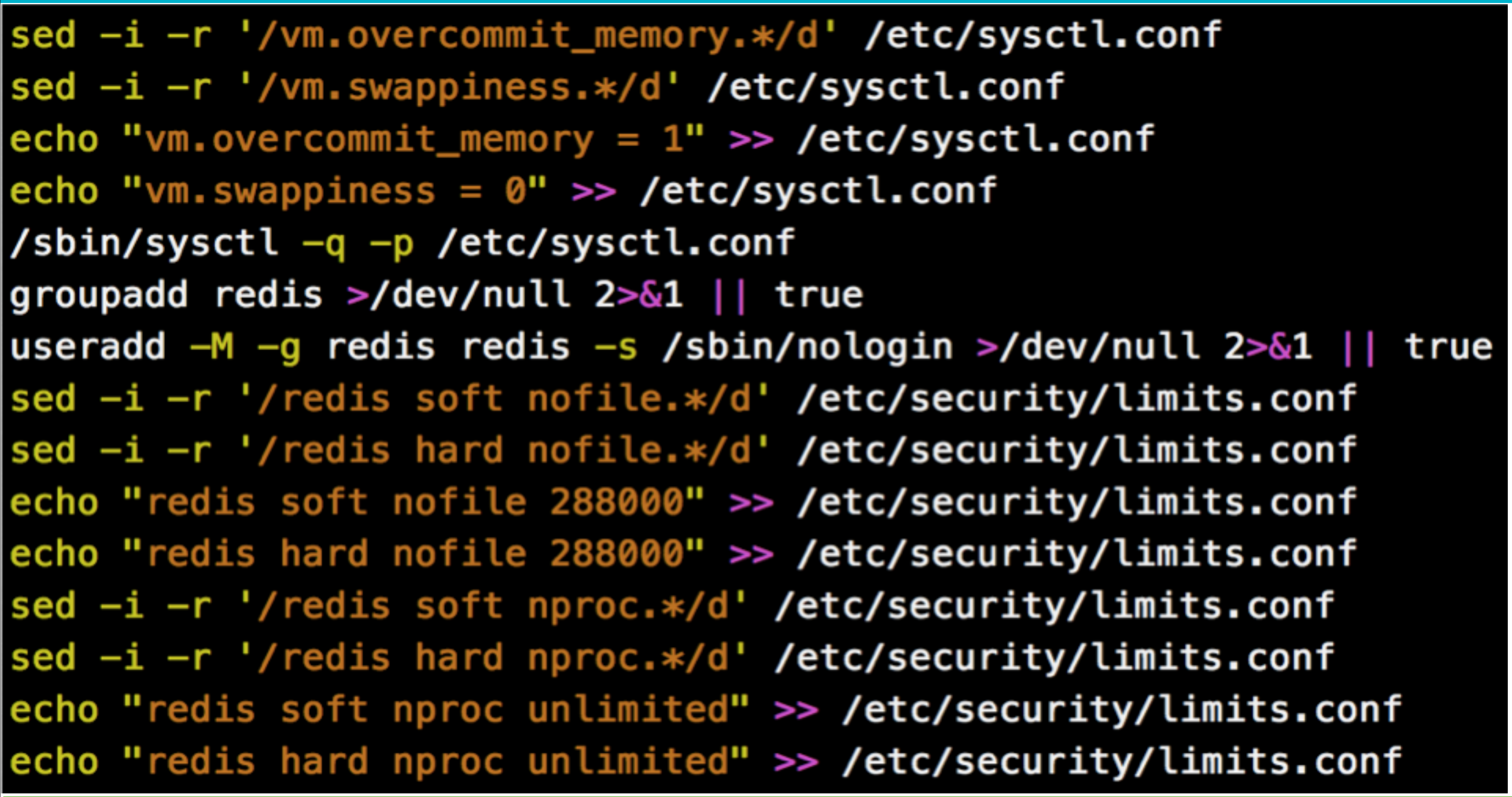
- 面临的问题：
  - 单机多实例的部署和管理成本增大
  - 不得不面对单机多版本的兼容问题
  - storage vs cache，需求变化多样
  - 需要工具和流程来减少DBA误操作
  - 遇到故障需要能快速恢复服务

规范化

统一化

自动化

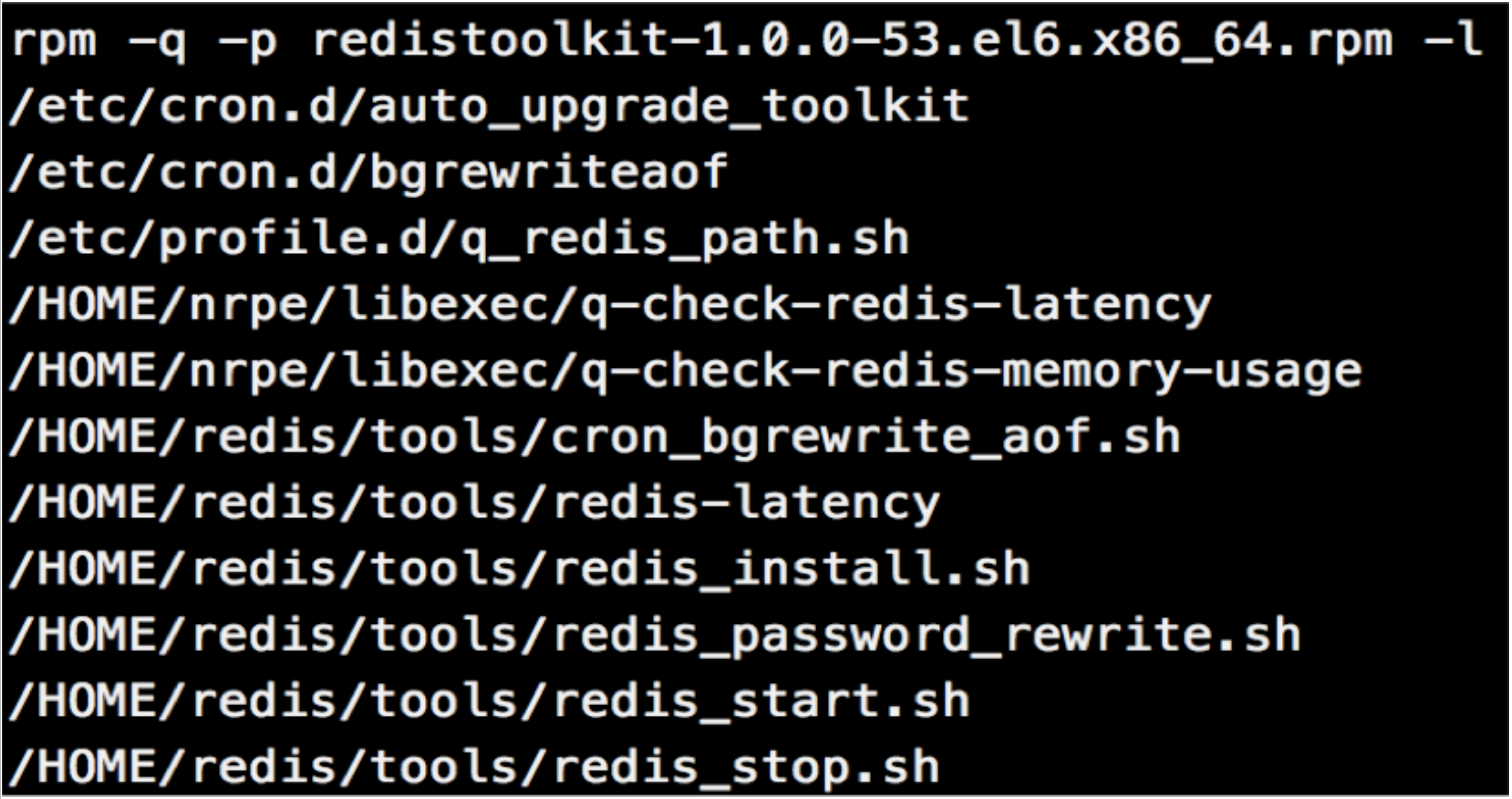
[1]初始化系统环境



```
sed -i -r '/vm.overcommit_memory.*/d' /etc/sysctl.conf
sed -i -r '/vm.swappiness.*/d' /etc/sysctl.conf
echo "vm.overcommit_memory = 1" >> /etc/sysctl.conf
echo "vm.swappiness = 0" >> /etc/sysctl.conf
/sbin/sysctl -q -p /etc/sysctl.conf
groupadd redis >/dev/null 2>&1 || true
useradd -M -g redis redis -s /sbin/nologin >/dev/null 2>&1 || true
sed -i -r '/redis soft nofile.*/d' /etc/security/limits.conf
sed -i -r '/redis hard nofile.*/d' /etc/security/limits.conf
echo "redis soft nofile 288000" >> /etc/security/limits.conf
echo "redis hard nofile 288000" >> /etc/security/limits.conf
sed -i -r '/redis soft nproc.*/d' /etc/security/limits.conf
sed -i -r '/redis hard nproc.*/d' /etc/security/limits.conf
echo "redis soft nproc unlimited" >> /etc/security/limits.conf
echo "redis hard nproc unlimited" >> /etc/security/limits.conf
```

Just add these lines to your rpm spec file !

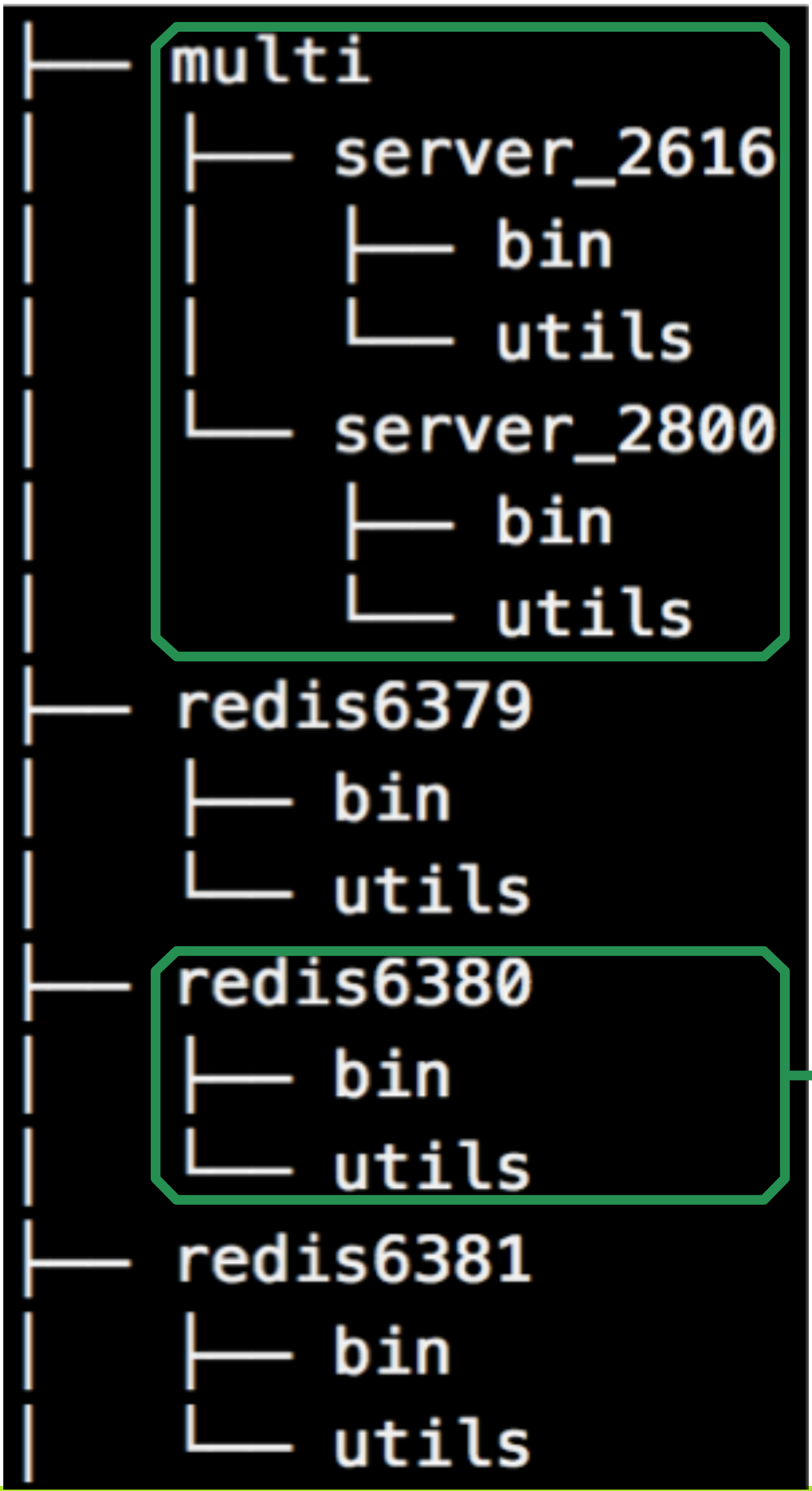
[2]统一管理工具



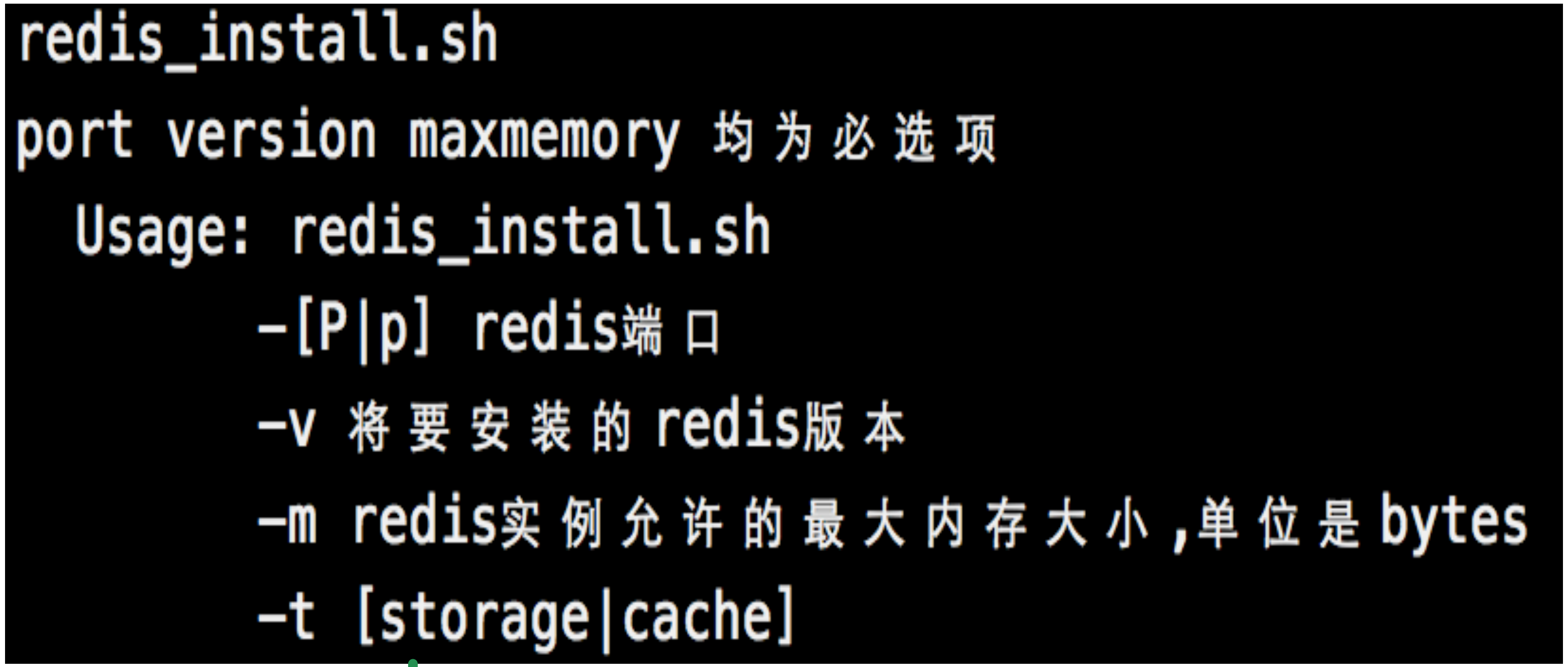
```
rpm -q -p redistoolkit-1.0.0-53.el6.x86_64.rpm -l
/etc/cron.d/auto_upgrade_toolkit
/etc/cron.d/bgrewriteaof
/etc/profile.d/q_redis_path.sh
/HOME/nrpe/libexec/q-check-redis-latency
/HOME/nrpe/libexec/q-check-redis-memory-usage
/HOME/redis/tools/cron_bgrewrite_aof.sh
/HOME/redis/tools/redis-latency
/HOME/redis/tools/redis_install.sh
/HOME/redis/tools/redis_password_rewrite.sh
/HOME/redis/tools/redis_start.sh
/HOME/redis/tools/redis_stop.sh
```

yum install toolkit， enjoy！

[3]单机多实例多版本

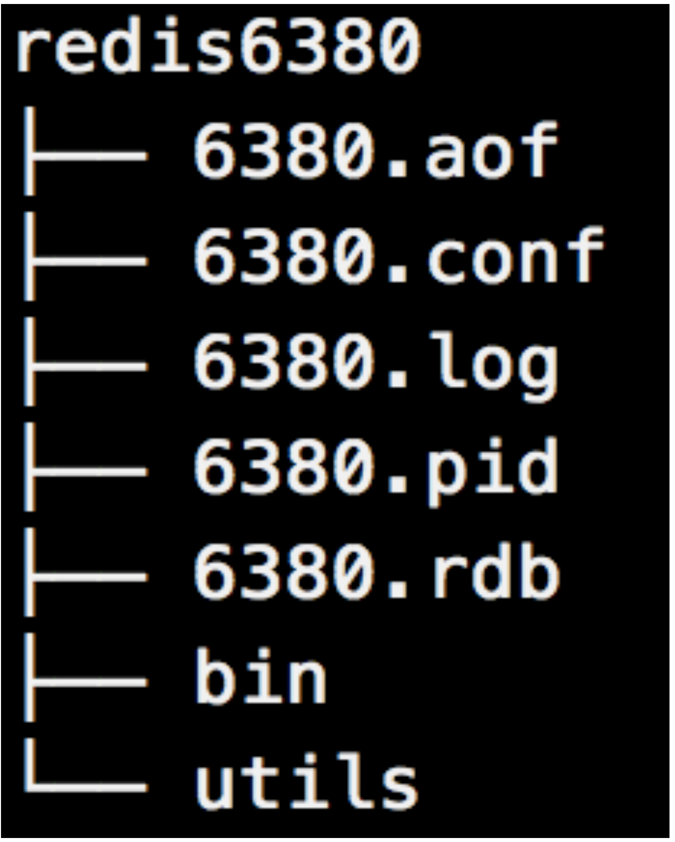


```
├── multi
│   ├── server_2616
│   │   ├── bin
│   │   └── utils
│   └── server_2800
│       ├── bin
│       └── utils
├── redis6379
│   ├── bin
│   └── utils
├── redis6380
│   ├── bin
│   └── utils
├── redis6381
│   ├── bin
│   └── utils
```



```
redis_install.sh
port version maxmemory 均为必选项
  Usage: redis_install.sh
        -[P|p] redis端口
        -v 将要安装的redis版本
        -m redis实例允许的最大内存大小,单位是bytes
        -t [storage|cache]
```

安装脚本提供选项和配置文件模版，
自定义安装不同版本



```
redis6380
├── 6380.aof
├── 6380.conf
├── 6380.log
├── 6380.pid
├── 6380.rdb
├── bin
└── utils
```

Redis高可用

- 面临的问题
  - 网络瞬断导致全同步
  - 主从没有同步点，recovery只能依赖AOF
  - slaveof、bgsave对IO冲击大
  - 持久化引起swap
  - 单实例受到QPS、容量限制

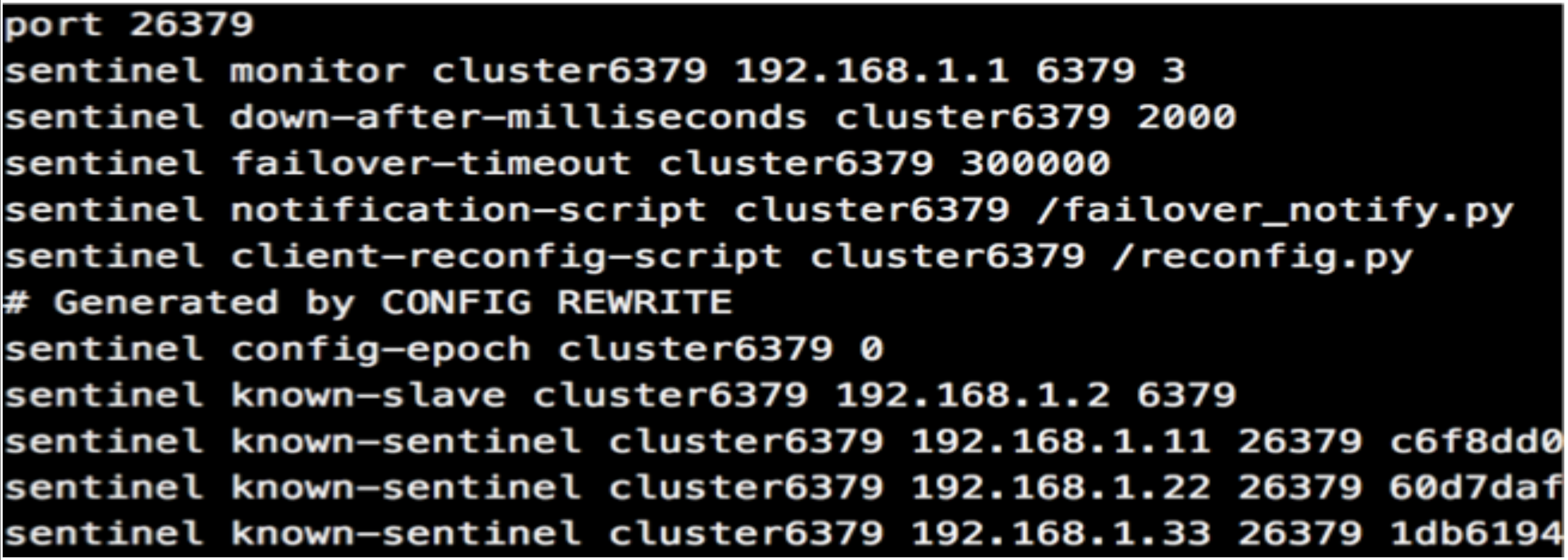
自动错开AOF重写

单实例容量20G以下，引入中间层突破容量限制

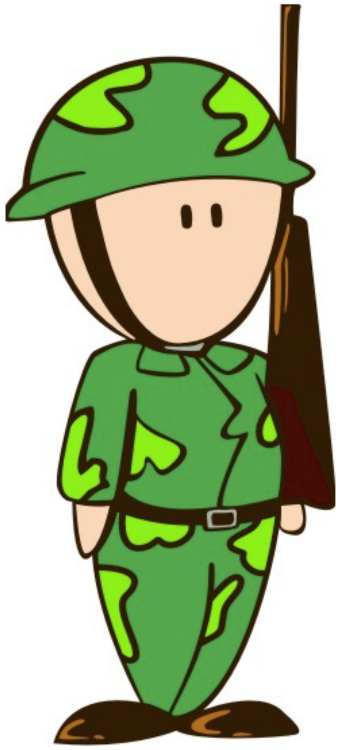
哨兵保障主库高可用

Redis哨兵

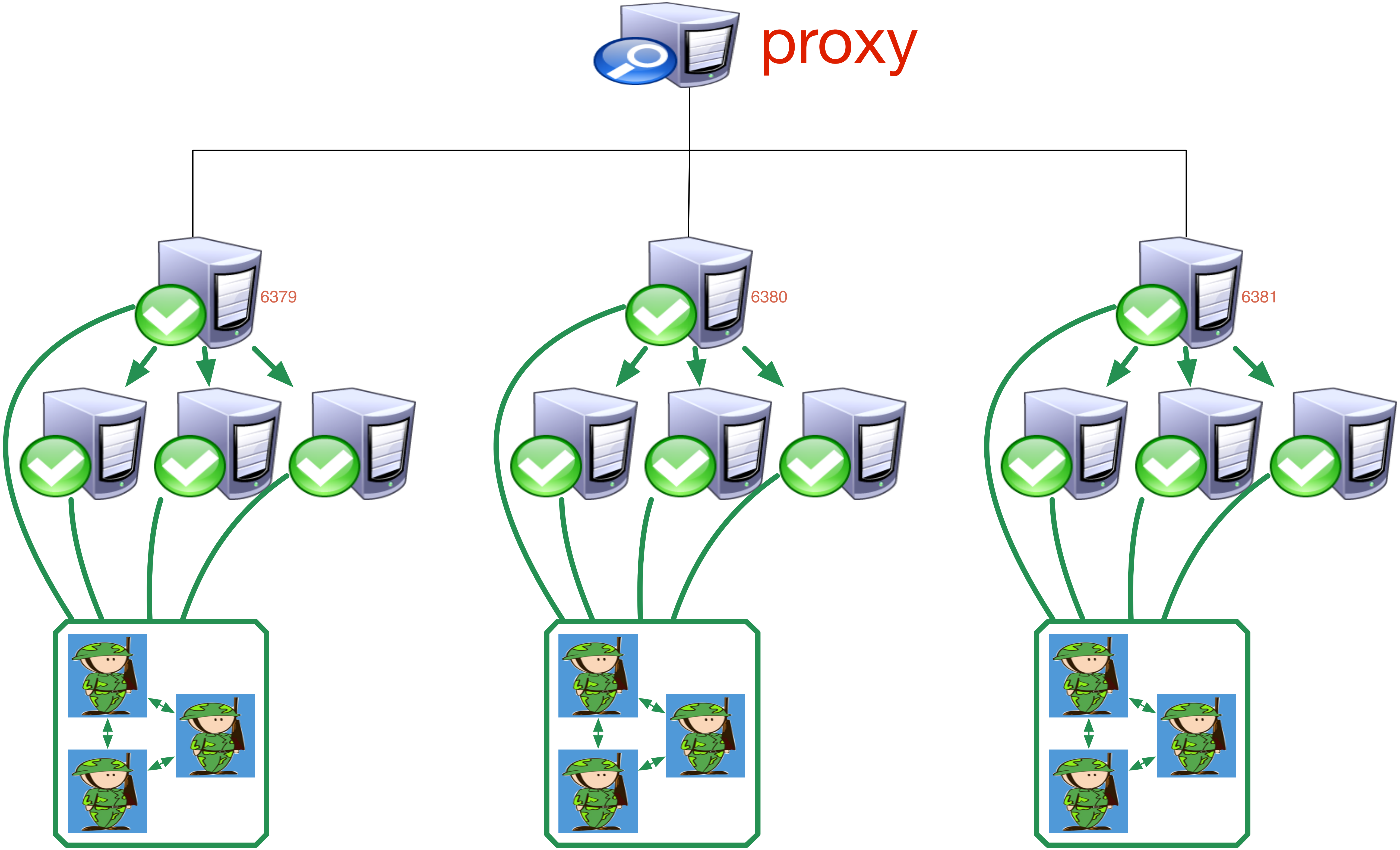
- 哨兵简介
  - redis-server的独特运行方式
  - 通过PUB/SUB来交流
  - 分布式
  - 切换时先选举Leader，再选举新主库



```
port 26379
sentinel monitor cluster6379 192.168.1.1 6379 3
sentinel down-after-milliseconds cluster6379 2000
sentinel failover-timeout cluster6379 300000
sentinel notification-script cluster6379 /failover_notify.py
sentinel client-reconfig-script cluster6379 /reconfig.py
# Generated by CONFIG REWRITE
sentinel config-epoch cluster6379 0
sentinel known-slave cluster6379 192.168.1.2 6379
sentinel known-sentinel cluster6379 192.168.1.11 26379 c6f8dd0
sentinel known-sentinel cluster6379 192.168.1.22 26379 60d7daf
sentinel known-sentinel cluster6379 192.168.1.33 26379 1db6194
```



哨兵与redis集群



twemproxy中间层

- twemproxy简介
  - open source，@twitter
  - 兼容mc与redis
  - 多种一致性hash策略
  - 长连接，减少连接数
  - pipeline，批量处理
  - 两种使用方式：
    - cache：动态映射
    - storage：用从库顶替

性能

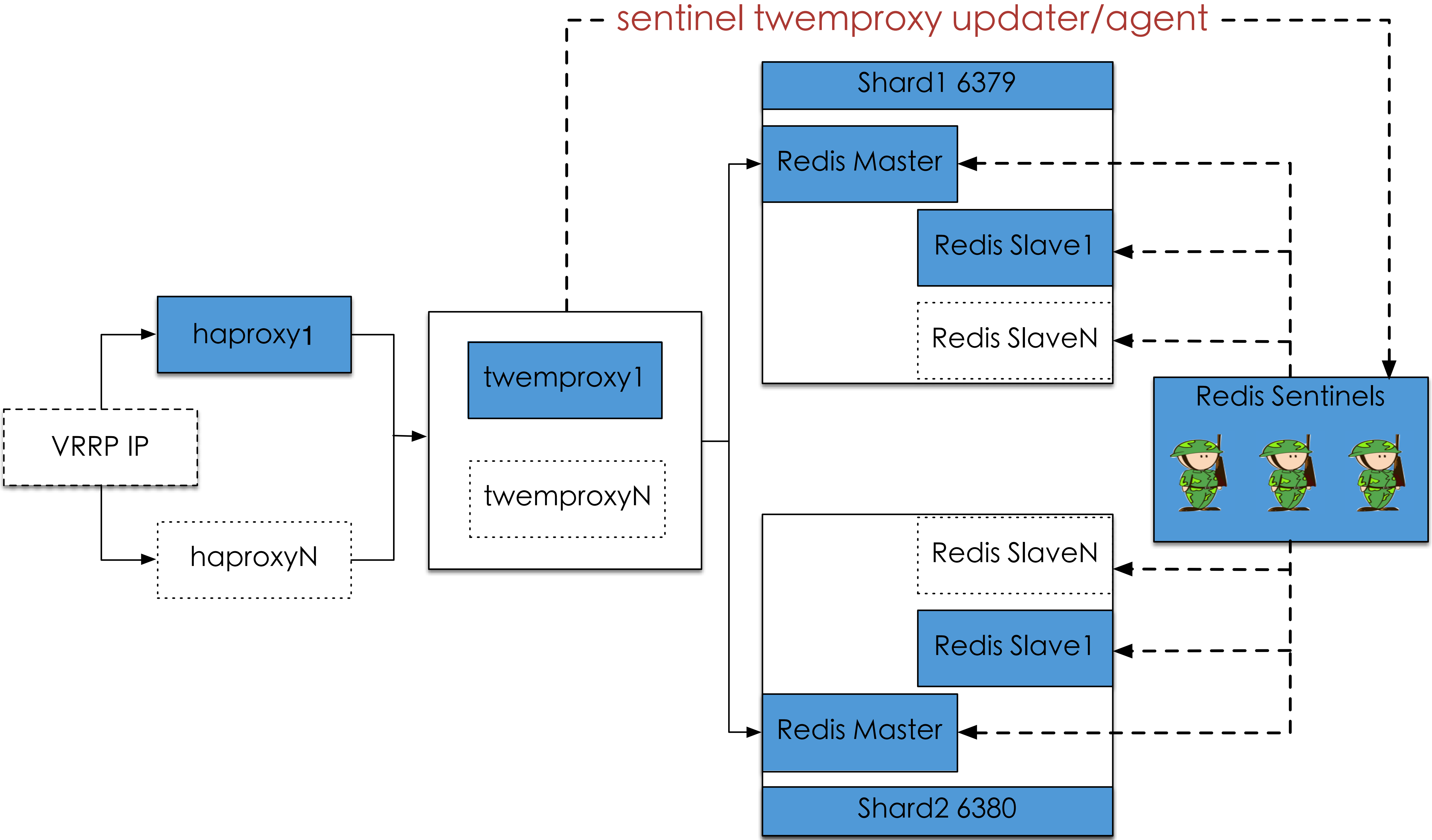
multi key

管理命令

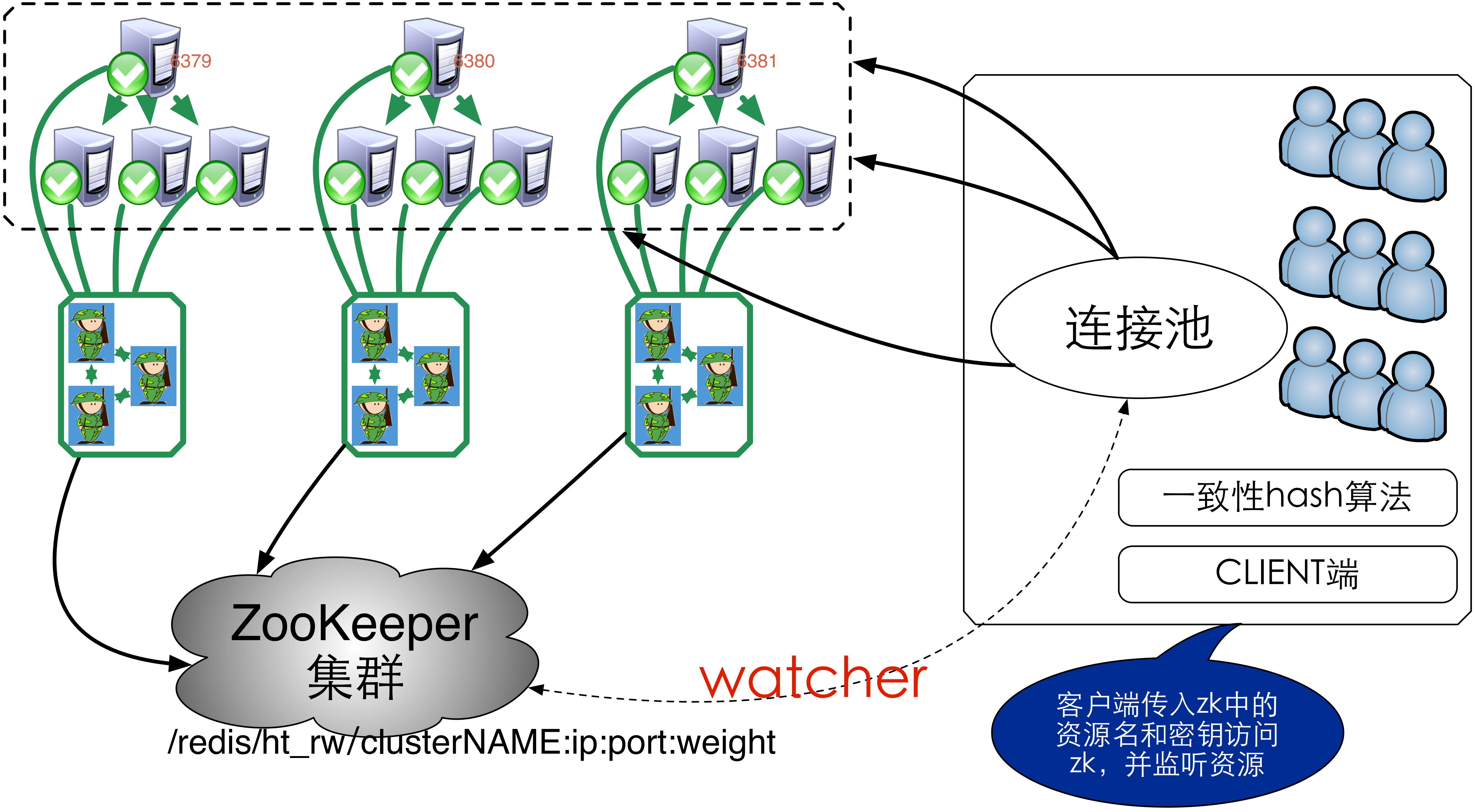
事务

高可用

Redis HA：哨兵+中间层



客户端+ZooKeeper实现高可用



Our patches for redis

- redis 安全patch
  - 屏蔽远程危险命令
  - 应用密码+白名单
- redis启动即注册为zookeeper临时节点
  - zookeeper
  - zookeeper-hosts
  - zookeeper-path
- redis响应时间监控

```
redis-latency -p 6379 --latency-history
avg:0.08:== 单位：毫秒 (199 samples) -- 2.01 seconds range
```

redis故障诊断

- 排查redis错误日志
- 排查cacti监控图像-[percona redis 监控模版]
- 排查哨兵日志
- 排查响应时间统计
- 排查系统资源监控

Q&A