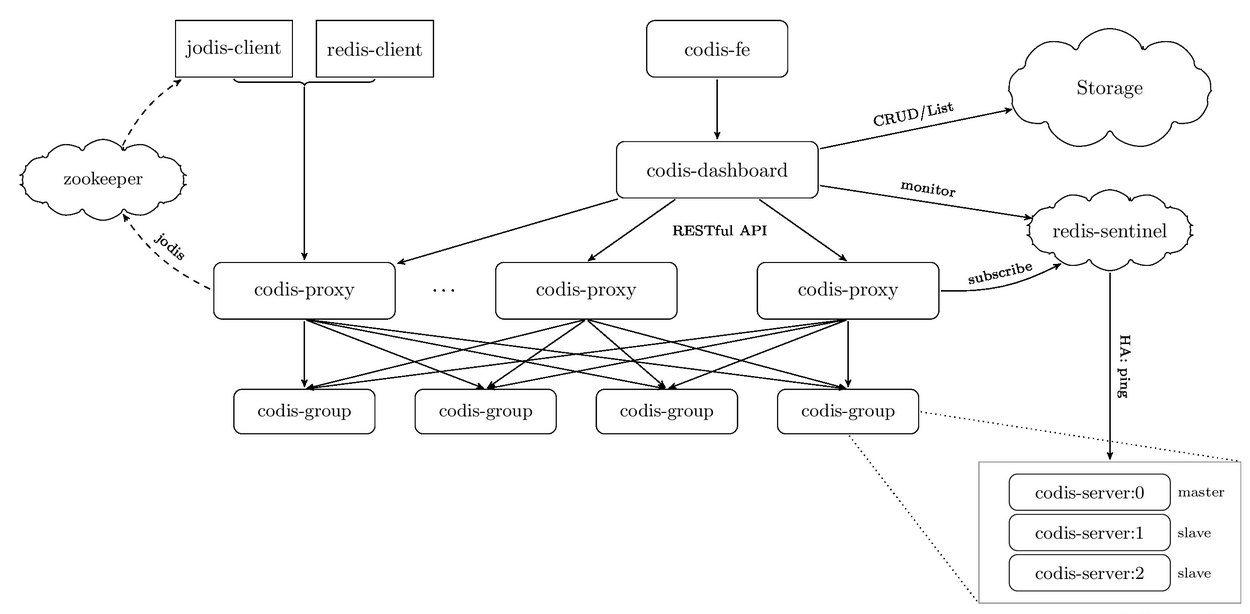
**Codis部署配置**

**一、环境介绍**



Codis3.x集群组件简介（官方）

1.Codis Server：基于 redis-3.2.8 分支开发。增加了额外的数据结构，以支持 slot 有关的操作以及数据迁移指令。具体的修改可以参考文档 redis 的修改。

2.Codis Proxy：客户端连接的 Redis 代理服务, 实现了Redis 协议。 除部分命令不支持以外(不支持的命令列表)，表现的和原生的 Redis 没有区别（就像 Twemproxy）。   
- 对于同一个业务集群而言，可以同时部署多个 codis-proxy 实例；   
- 不同 codis-proxy 之间由 codis-dashboard 保证状态同步。

3.Redis sentinel：Redis官方推荐的高可用性(HA)解决方案。它可以实现对Redis的监控、通知、自动故障转移。如果Master不能工作，则会自动启动故障转移进程，将其中的一个Slave提升为Master，其他的Slave重新设置新的Master服务。

4.Codis Dashboard：集群管理工具，支持 codis-proxy、codis-server 的添加、删除，以及据迁移等操作。在集群状态发生改变时，codis-dashboard 维护集群下所有 codis-proxy 的状态的一致性。   
- 对于同一个业务集群而言，同一个时刻 codis-dashboard 只能有 0个或者1个；   
- 所有对集群的修改都必须通过 codis-dashboard 完成。

5.Codis Admin：集群管理的命令行工具。   
- 可用于控制 codis-proxy、codis-dashboard 状态以及访问外部存储。

6.Codis FE：集群管理界面。   
- 多个集群实例共享可以共享同一个前端展示页面；   
- 通过配置文件管理后端codis-dashboard列表，配置文件可自动更新。

7.Storage：为集群状态提供外部存储。   
- 提供namespace概念，不同集群的会按照不同product name进行组织；   
- 目前仅提供了zookeeper、etcd、filesystem三种实现，但是提供了抽象的 interface 可自行扩展。

**zookeeper集群部署参考另一篇文档**

|  |  |  |  |
| --- | --- | --- | --- |
| 主机名 | Codis-01 | Codis-02 | Codis-03 |
| IP地址 | 192.168.1.210 | 192.168.1.211 | 192.168.1.212 |
| 服务 | zookeeper | zookeeper | zookeeper |
| codis-dashboard:18080 | codis-proxy:19000 | codis-proxy:19000 |
| codis-fe:18090 | redis-sentinel:26379 | redis-sentinel:26379 |
| codis-server:(6379&6380) | codis-server:(6379&6380) | codis-server:(6379&6380) |

**二、Codis安装**

1. 配置go语言环境（Codis基于go语言开发）

下载安装包

#salt 'Codis-\*' cmd.run 'wget [https://storage.googleapis.com/golang/go1.7.3.linux-amd64.tar.gz'](https://storage.googleapis.com/golang/go1.4.1.linux-amd64.tar.gz')

解压到相应目录

#salt 'Codis-\*' cmd.run 'tar -zxf go1.7.3.linux-amd64.tar.gz -C /data/application'

创建所需的目录（与go无关，为了整个部署过程的环境变量）

#salt 'Codis-\*' cmd.run 'mkdir /data/application/codis'

编辑环境变量配置文件

#vi /srv/salt/prod/codis/files/conf/go.sh

#!/bin/bash

export GOROOT=/data/application/go

export GOPATH=/data/application/codis

export PATH=$PATH:$GOROOT/bin:$GOPATH/bin

export CODIS\_HOME=$GOPATH/src/github.com/CodisLabs/codis

export GO15VENDOREXPERIMENT=0

将配置文件推送到各个节点

#salt-cp 'Codis-\*' /srv/salt/prod/codis/files/conf/go.sh /etc/profile.d/go.sh

2.Codis安装

创建Codis项目下载路径

#salt 'Codis-\*' cmd.run 'mkdir -p /data/application/codis/src/github.com/CodisLabs'

从github下载codis编译安装包

#salt 'Codis-\*' cmd.run 'cd /data/application/codis/src/github.com/CodisLabs && git clone https://github.com/CodisLabs/codis.git -b release3.2'

编译安装codis

#salt 'Codis-\*' cmd.run 'cd /data/application/codis/src/github.com/CodisLabs/codis && make'

**三、Codis-Server配置**

复制启动相关文件到codis目录

#salt 'Codis-\*' cmd.run 'cp -fr /data/application/codis/src/github.com/CodisLabs/codis/bin /data/application/codis/'

复制默认配置文件到codis目录

#salt 'Codis-\*' cmd.run 'cp -fr /data/application/codis/src/github.com/CodisLabs/codis/config /data/application/codis/'

创建日志目录

#salt 'Codis-\*' cmd.run 'mkdir -p /data/application/codis/log'

创建主节点数据目录

#salt 'Codis-\*' cmd.run 'mkdir -p /data/application/codis/data/redis-6379'

创建从节点数据目录

#salt 'Codis-\*' cmd.run 'mkdir -p /data/application/codis/data/redis-6380'

编辑主节点配置文件

#vi /srv/salt/prod/codis/files/conf/redis-6379.conf

daemonize yes   
 pidfile /data/application/codis/proc/redis-6379.pid  
 port 6379   
 timeout 86400   
 tcp-keepalive 60   
 loglevel notice   
 logfile /data/application/codis/log/redis-6379.log  
 databases 16   
 save “”   
 save 900 1   
 save 300 10   
 save 60 10000   
 stop-writes-on-bgsave-error no   
 rdbcompression yes   
 dbfilename dump-6379.rdb  
 dir /data/application/codis/data/redis\_data\_6379  
 masterauth "redis\_sync"  
 slave-serve-stale-data yes   
 repl-disable-tcp-nodelay no   
 slave-priority 100   
 requirepass "haomai123"   
 maxmemory 1gb   
 maxmemory-policy allkeys-lru   
 appendonly no   
 appendfsync everysec   
 no-appendfsync-on-rewrite yes   
 auto-aof-rewrite-percentage 100   
 auto-aof-rewrite-min-size 64mb

lua-time-limit 5000   
 slowlog-log-slower-than 10000   
 slowlog-max-len 128   
 hash-max-ziplist-entries 512   
 hash-max-ziplist-value 64   
 list-max-ziplist-entries 512   
 list-max-ziplist-value 64   
 set-max-intset-entries 512   
 zset-max-ziplist-entries 128   
 zset-max-ziplist-value 64   
 client-output-buffer-limit normal 0 0 0   
 client-output-buffer-limit slave 0 0 0   
 client-output-buffer-limit pubsub 0 0 0   
 hz 10   
 aof-rewrite-incremental-fsync yes   
 repl-backlog-size 33554432

编辑从节点配置文件

#vi /srv/salt/prod/codis/files/conf/redis-6380.conf

daemonize yes

pidfile /data/application/codis/proc/redis-6380.pid

port 6380

timeout 86400

tcp-keepalive 60

loglevel notice

logfile /data/application/codis/log/redis-6380.log

databases 16

save “”

save 900 1

save 300 10

save 60 10000

stop-writes-on-bgsave-error no

rdbcompression yes

dbfilename dump-6380.rdb

dir /data/application/codis/data/redis\_data\_6380

masterauth "redis\_sync"

slave-serve-stale-data yes

repl-disable-tcp-nodelay no

slave-priority 100

requirepass "haomai123"

maxmemory 1gb

maxmemory-policy allkeys-lru

appendonly no

appendfsync everysec

no-appendfsync-on-rewrite yes

auto-aof-rewrite-percentage 100

auto-aof-rewrite-min-size 64mb

lua-time-limit 5000

slowlog-log-slower-than 10000

slowlog-max-len 128

hash-max-ziplist-entries 512

hash-max-ziplist-value 64

list-max-ziplist-entries 512

list-max-ziplist-value 64

set-max-intset-entries 512

zset-max-ziplist-entries 128

zset-max-ziplist-value 64

client-output-buffer-limit normal 0 0 0

client-output-buffer-limit slave 0 0 0

client-output-buffer-limit pubsub 0 0 0

hz 10

aof-rewrite-incremental-fsync yes

repl-backlog-size 33554432

主配置文件推送

#salt-cp 'Codis-\*' /srv/salt/prod/codis/files/conf/redis-6379.conf /data/application/codis/config/redis-6379.conf

从配置文件推送

#salt-cp 'Codis-\*' /srv/salt/prod/codis/files/conf/redis-6380.conf /data/application/codis/config/redis-6380.conf

启动主节点

#salt 'Codis-\*' cmd.run '/data/application/codis/bin/codis-server /data/application/codis/config/redis-6379.conf'

启动从节点

#salt 'Codis-\*' cmd.run '/data/application/codis/bin/codis-server /data/application/codis/config/redis-6380.conf'

**四、Codis-dashboard配置**

#vi /srv/salt/prod/codis/files/conf/dashboard.conf

##################################################

# #

# Codis-Dashboard #

# #

##################################################

# Set Coordinator, only accept "zookeeper" & "etcd" & "filesystem".

# for zookeeper/etcd, coorinator\_auth accept "user:password"

# Quick Start

coordinator\_name = "zookeeper"

coordinator\_addr = "192.168.1.210:2181,192.168.1.211:2181,192.168.1.212:2181"

# Set Codis Product Name/Auth.

product\_name = "haomai-codis"

product\_auth = "haomai123"

# Set bind address for admin(rpc), tcp only.

admin\_addr = "192.168.1.210:18080"

# Set arguments for data migration (only accept 'sync' & 'semi-async').

migration\_method = "semi-async"

migration\_parallel\_slots = 100

migration\_async\_maxbulks = 200

migration\_async\_maxbytes = "32mb"

migration\_async\_numkeys = 500

migration\_timeout = "30s"

# Set configs for redis sentinel.

sentinel\_client\_timeout = "10s"

sentinel\_quorum = 2

sentinel\_parallel\_syncs = 1

sentinel\_down\_after = "30s"

sentinel\_failover\_timeout = "5m"

sentinel\_notification\_script = ""

sentinel\_client\_reconfig\_script = ""

|  |  |  |
| --- | --- | --- |
| 参数 | 说明 | 备注 |
| coordinator\_name | 存储类型 | 备注中以说明支持的三种类型。  Zookeeper,etcd,filesystem |
| coordinator\_addr | 存储列表 | 逗号分隔，IP:port |
| product\_name | 项目名称 |  |
| product\_auth | 集群密码 | 与redis-server内配置的requirepass一致 |
| admin\_addr | 监听地址与端口 | IP:port |

仅向规划的那台主机推送dashboard配置文件

#salt-cp 'Codis-01' /srv/salt/prod/codis/files/conf/dashboard.conf /data/application/codis/config/dashboard.conf

启动dashboard

#salt 'Codis-01' cmd.run 'nohup /data/application/codis/bin/codis-dashboard --ncpu=2 --config=/data/application/codis/config/dashboard.conf --log=/data/application/codis/log/dashboard.log --log-level=WARN &'

(如不加nohup与&，也能正常启动。因任务未执行完毕，JID仍存在于salt-master故此看似一切正常，当salt-master发生故障或停止时，则业务进程消失，风险极高)

|  |  |  |
| --- | --- | --- |
| 参数 | 描述 | 备注 |
| --ncpu | 最大使用CPU个数 |  |
| --config | 指定配置路径和文件 | 使用绝对路径 |
| --log | 指定日志输出路径和文件 | 使用绝对路径 |
| --log-level | 指定日志级别 | 取值：INFO、WARN、DEBUG、ERROR，推荐使用WRAN. |

停止dashboard

#salt 'Codis-01' cmd.run '/data/application/codis/bin/codis-admin --dashboard=192.168.1.210:18080 --shutdown'

1. **Codis-Proxy配置**

#vi /srv/salt/prod/codis/files/conf/proxy.conf

##################################################

# #

# Codis-Proxy #

# #

##################################################

product\_name = "haomai-codis"

product\_auth = "haomai123"

session\_auth = "HaoMai123#"

admin\_addr = "0.0.0.0:11080"

proto\_type = "tcp4"

proxy\_addr = "0.0.0.0:19000"

jodis\_name = "zookeeper"

jodis\_addr = "192.168.1.210:2181,192.168.1.211:2181,192.168.1.212:2181"

jodis\_auth = ""

jodis\_timeout = "20s"

jodis\_compatible = false

....(删除了注释及未做修改的部分配置信息)

|  |  |  |
| --- | --- | --- |
| 参数 | 描述 | 备注 |
| product\_name | 项目名称 |  |
| product\_auth | 设置登录dashboard密码 | 与redis配置的requirepass一致 |
| Session\_auth | 设置客户端登录密码 |  |
| admin\_addr | 管理员地址与端口 | 未修改。可结合实际情况修改监听地址 |
| proxy\_addr | 客户端连接的地址与端口 | 未修改。可结合实际情况修改监听地址 |
| jodis\_name | 存储类型 | 备注中以说明支持的两种类型。  Zookeeper,etcd |
| jodis\_addr | 存储列表 | 逗号分隔，IP:port |

推送proxy配置文件

#salt-cp 'Codis-0[2-3]' /srv/salt/prod/codis/files/conf/proxy.conf /data/application/codis/config/proxy.conf

启动proxy

#salt 'Codis-0[2-3]' cmd.run '/data/application/codis/bin/codis-proxy --ncpu=2 --config=/data/application/codis/config/proxy.conf --log=/data/application/codis/log/proxy.log --log-level=WARN &'

1. **Redis-sentinel配置**

复制sentinel程序

#salt 'Codis-0[2-3]' cmd.run 'cp -fr /data/application/codis/src/github.com/CodisLabs/codis/extern/redis-3.2.11/src/redis-sentinel /data/application/codis/bin/'

#vi /srv/salt/prod/codis/files/conf/sentinel.conf

bind 0.0.0.0

protected-mode no

port 26379

dir /data/application/codis/data

推送sentinel配置文件

#salt-cp 'Codis-0[2-3]' /srv/salt/prod/codis/files/conf/sentinel.conf /data/application/codis/config/sentinel.conf

启动sentinel

#salt 'Codis-0[2-3]' cmd.run 'nohup /data/application/codis/bin/redis-sentinel /data/application/codis/config/sentinel.conf &'

1. **codis-fe配置**

生成fe配置文件

#salt 'Codis-01' cmd.run '/data/application/codis/bin/codis-admin --dashboard-list --zookeeper=192.168.1.210:2181 | tee /data/application/codis/config/codis.json'

配置文件内容

[

{

"name": "haomai-codis",

"dashboard": "192.168.1.210:18080"

}

]

启动fe

#salt 'Codis-01' cmd.run 'nohup /data/application/codis/bin/codis-fe --ncpu=2 --log=/data/application/codis/log/fe.log --log-level=WARN --dashboard-list=/data/application/codis/config/codis.json --listen=0.0.0.0:18090 &'

1. WEB管理
2. 访问

本例访问URL

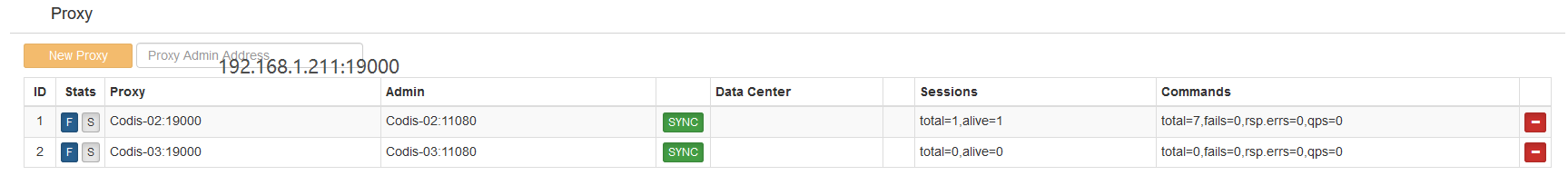
<http://192.168.1.210:18090>



这部分可以看到一些项目信息

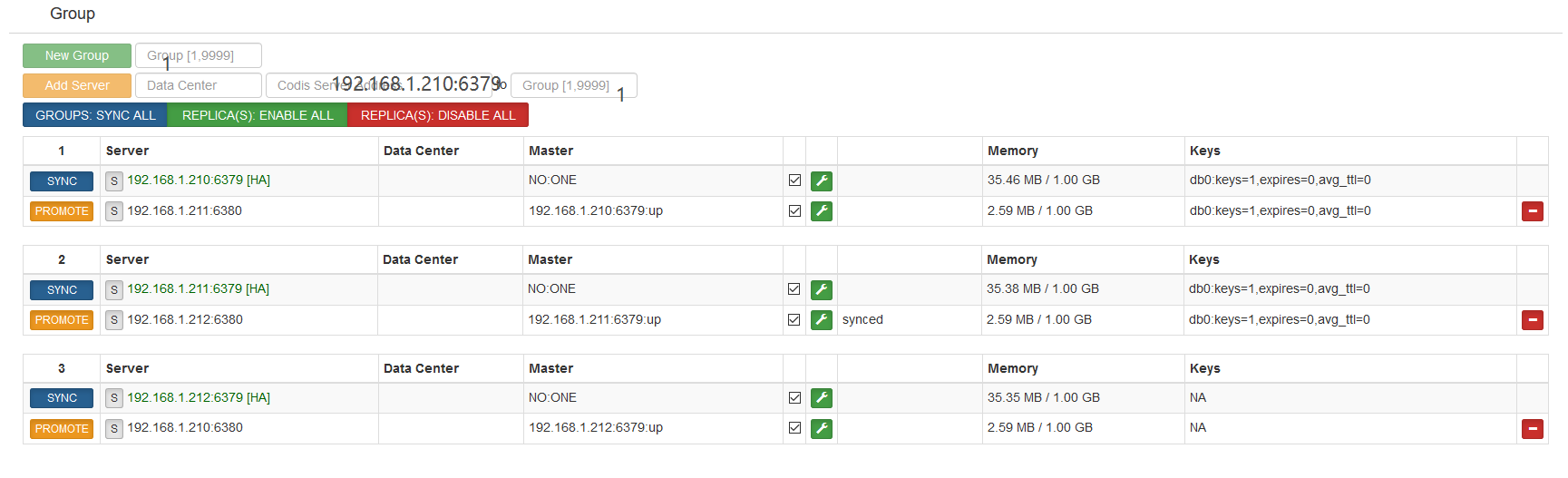
|  |  |
| --- | --- |
| Product Name | 项目名称 |
| [Zookeeper] | Zookeeper集群列表 |
| Codis QPS | 查询率 |
| Sessions | 会话数 |
| Redis Memory | Redis已用空间 |
| Redis Keys | Redis keys数 |
| Dashboard | Dashboard地址端口 |

1. 添加proxy



依次将192.168.1.211:1900和192.168.1.212:19000添加到此处

3.添加Codis-Server



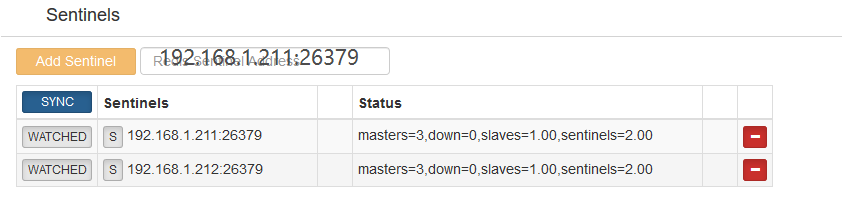
①NewGroup1组

②添加192.168.1.210:6379和192.168.1.211:6380到1组

依次创建2组、3组，将其它节点加入

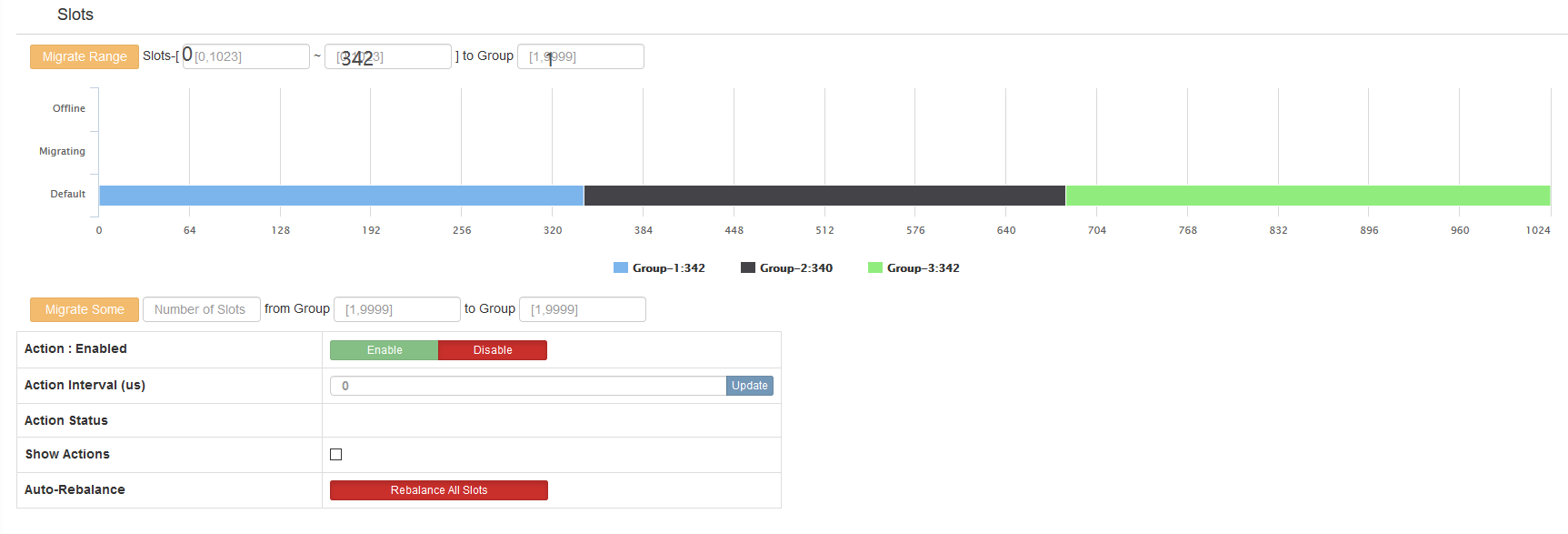
使用不同服务器上的Codis-Server组成HA组

1. 添加sentinel



依次将192.168.1.211:26379和192.168.1.212:26379加入到sentinel

1. 划分slots



依次将1、2、3组划分slots

起slots为0，最大为1024

尽可能平均分为3组进行划分

除了创建slots还有歉意功能，此处还没研究。待更新...

1. **故障处理**

1.dashboard日志报错

2018/03/13 09:39:04 main.go:171: [WARN] [0xc420154120] dashboard online failed [13]

2018/03/13 09:39:06 topom.go:189: [ERROR] store: acquire lock of haomai-codis failed

[error]: zk: node already exists

/data/application/codis/bin/codis-admin --remove-lock --product=haomai-codis --zookeeper=192.168.1.210:2181

重新启动dashboard