**Topcoder - Create CronJob For Populating Changed Challenges To Elasticsearch version 1.0**

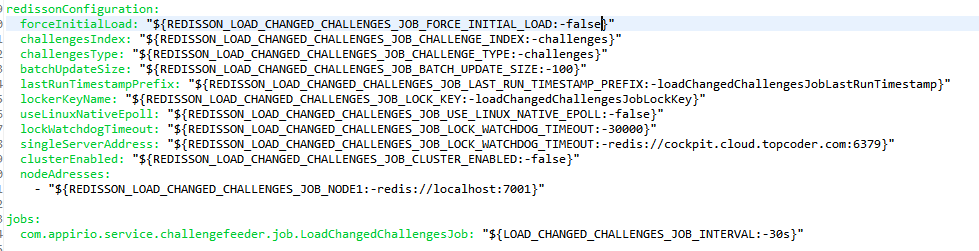
**Verification Guide**

# 

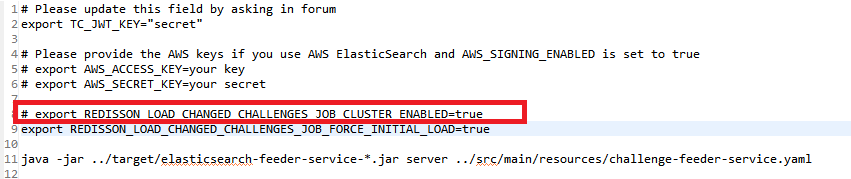
# Setup

For the challenge feeder service, run 'mvn clean package' to compile the code and then cd local folder and run './run.sh' to start the services. Before starting the services, you need to first setup the envrionment as mentioned in the Verification\_Topcoder - Elasticsearch Feeder Service For Challenges version 1.0.docx

In src/main/resources/challenge-feeder-service.yaml, all the config values can be overridden by the environment.



In local/run.sh, if the REDISSON\_LOAD\_CHANGED\_CHALLENGES\_JOB\_CLUSTER\_ENABLED is commented out, the service will be run in the standalone model. By default the service will be run against the standalove model.

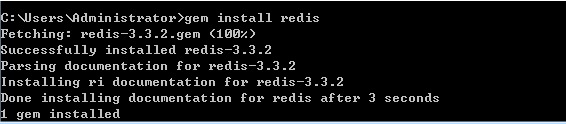


If REDISSON\_LOAD\_CHANGED\_CHALLENGES\_JOB\_CLUSTER\_ENABLED is set to true,

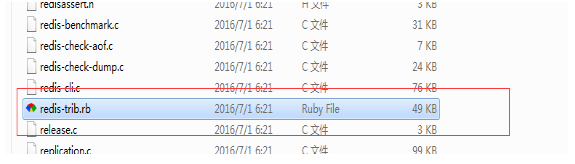
the redis clusters for the redisson should be setup.

Following is the steps for windows. For linux, you can refer to the document in https://redis.io

1. download [Ruby 2.3.3 (x64)](https://dl.bintray.com/oneclick/rubyinstaller/rubyinstaller-2.3.3-x64.exe)  http://rubyinstaller.org/downloads/ and install it. In the cmd, run 'gem install redis'

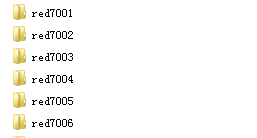


2. clone the https://github.com/MSOpenTech/redis, and find the redis-trib.rb file:



3. download the redis for windows: https://github.com/MSOpenTech/redis/releases/download/win-3.2.100/Redis-x64-3.2.100.zip

extrac the fiels to the following folders(6 copies)



In each redxxx file, redis-trib.rb should be copied from the step 2, and the redis.windows.conf will be:

daemonize yes

port 7001

cluster-enabled yes

cluster-config-file nodes.conf

cluster-node-timeout 15000

appendonly yes



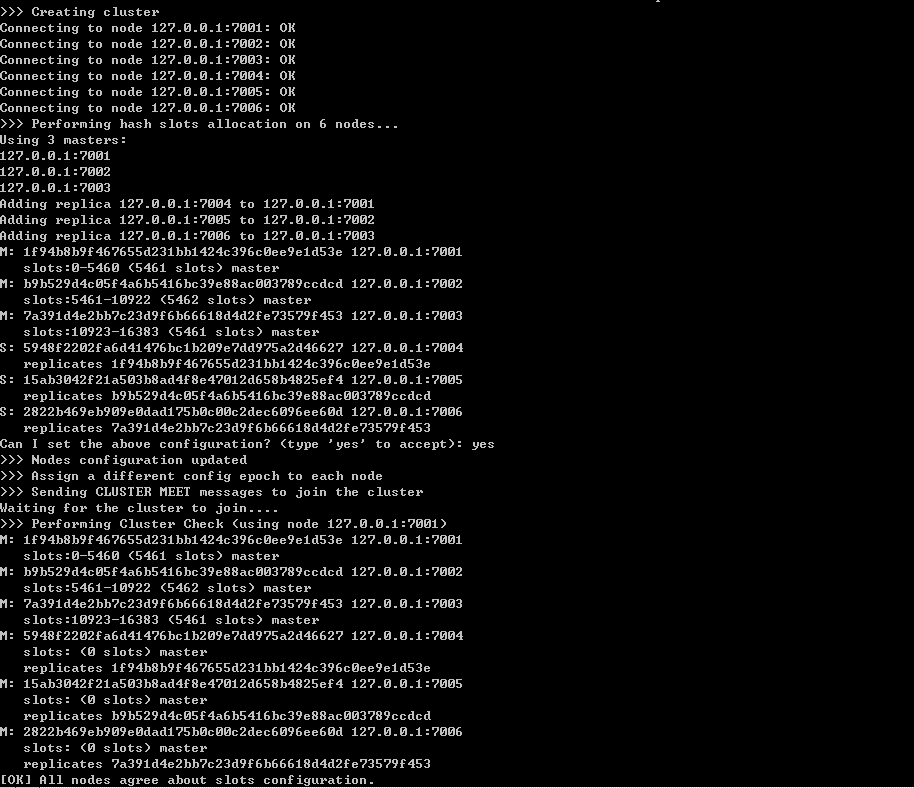
Then in each folder, run 'redis-server.exe redis.windows.conf' to start the node

4. open another cmd, and run the following:

redis-trib.rb create --replicas 1 127.0.0.1:7001 127.0.0.1:7002 127.0.0.1:7003 127.0.0.1:7004 127.0.0.1:7005 127.0.0.1:7006

redis-trib.rb can be set in the envrionment path.

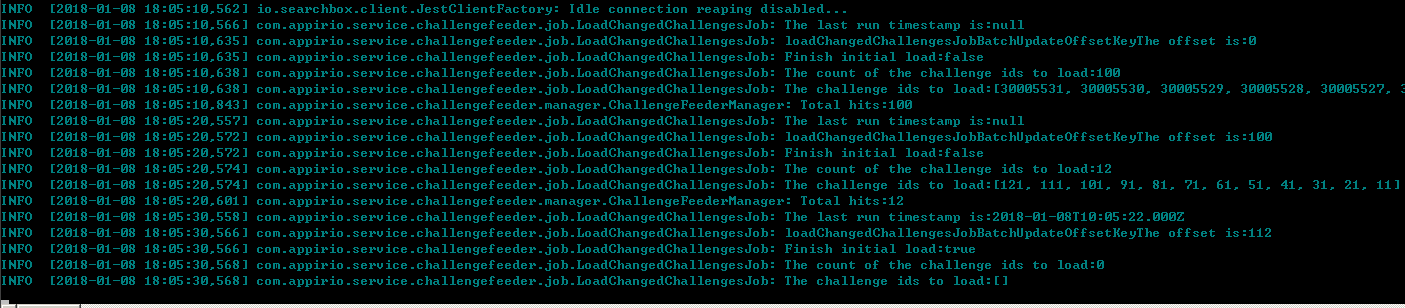
The output will be:



# Verify

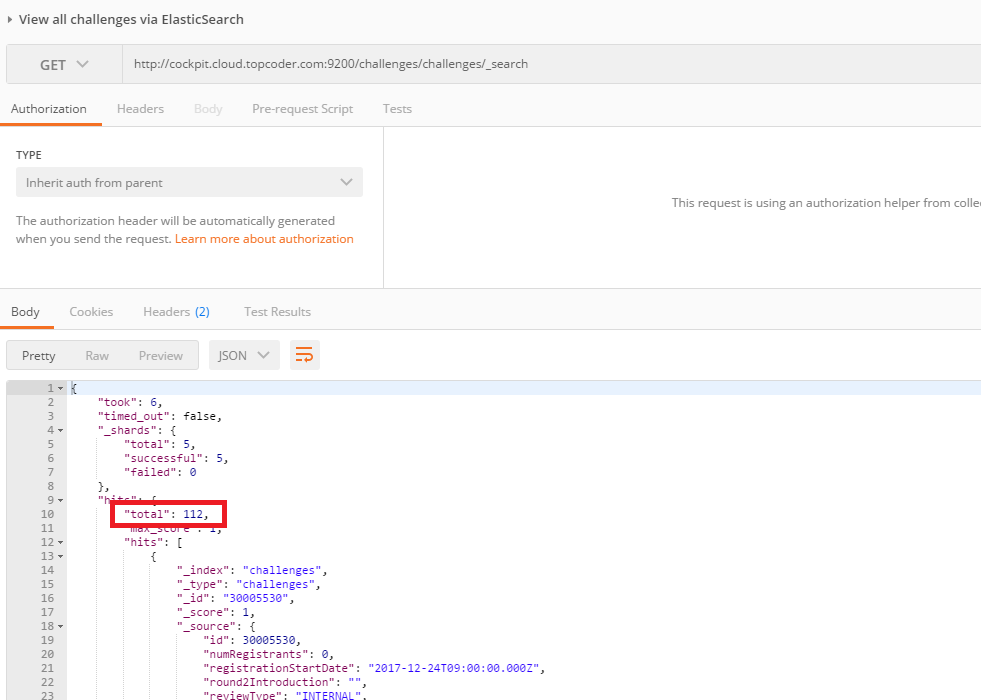
Run local/run.sh

The console will be:



Check the console, there ae total 112 challenges in the database. For the initial load, it will first load the 100 challenges, and then 12 challenges.

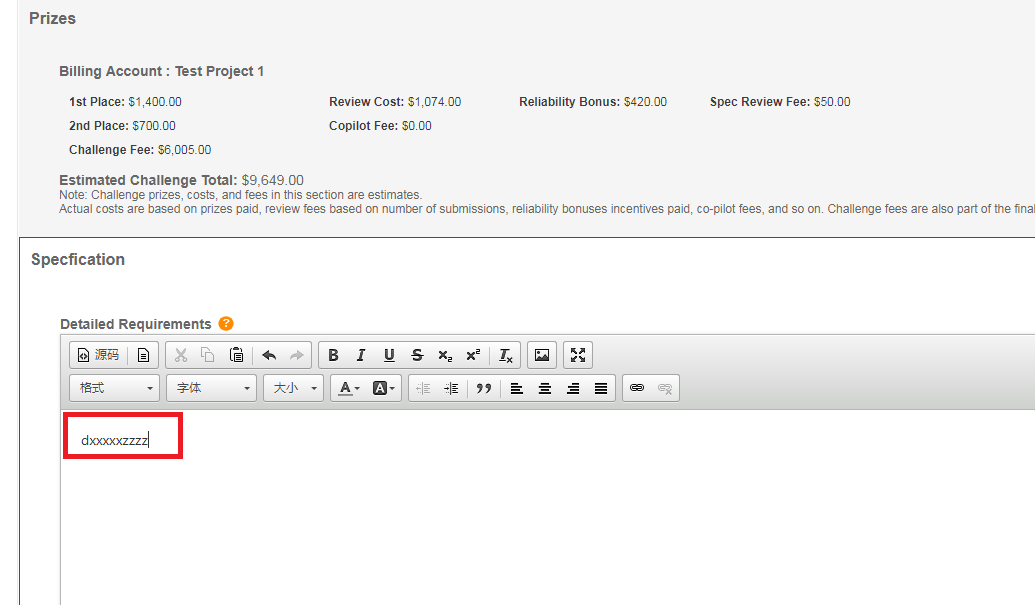
Check ' View all challenges via ElasticSearch' in the postman, totally 112 challenges are showed:



Then we try to modify the challenge with id 30005522 in the direct app:

https://cockpit.cloud.topcoder.com/direct/contest/detail.action?projectId=30005522

Edit the spec for it and save the changes:

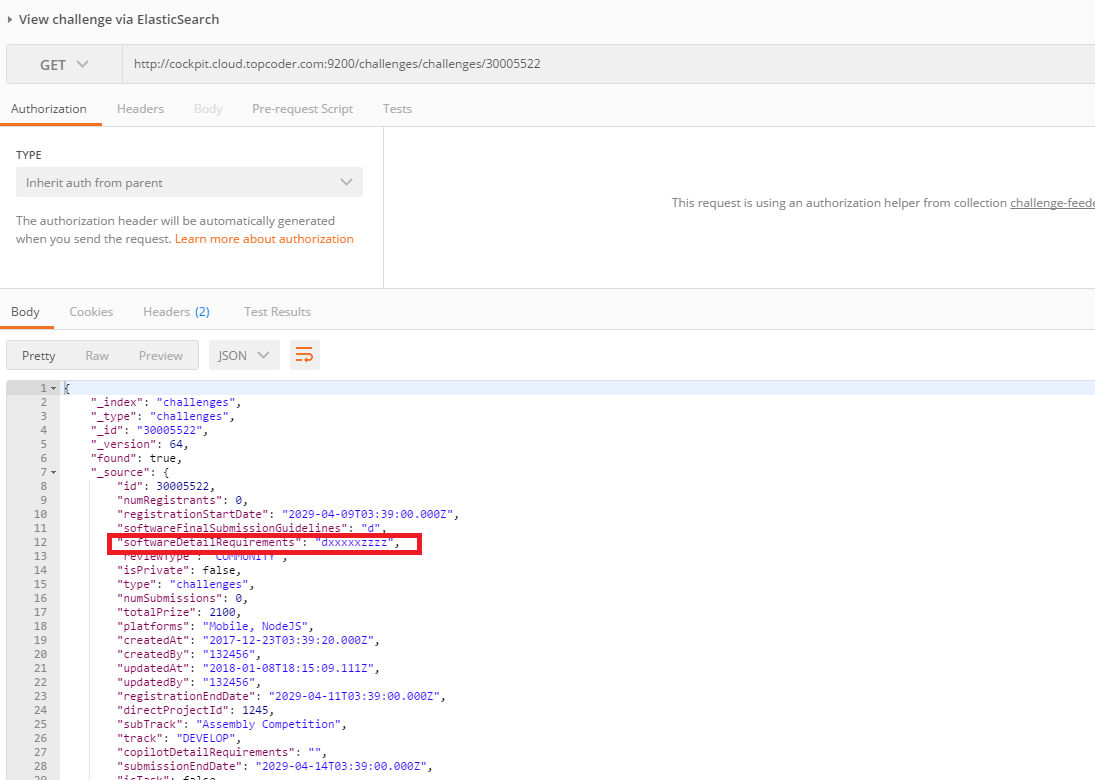


Wait for sometime and then check the console, the challenge 30005522 is updated:

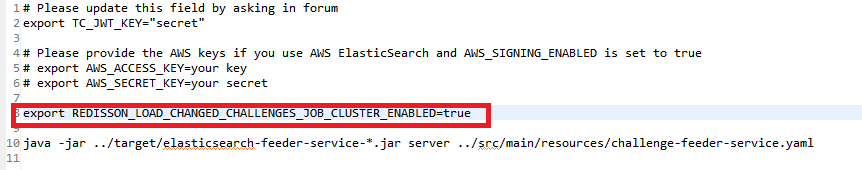


Then check the ' View challenge via ElasticSearch' in thepostman

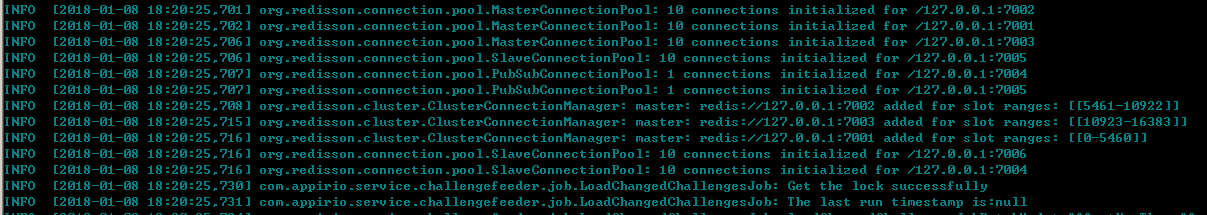
The challenge should be updated.



In local/run.sh, REDISSON\_LOAD\_CHANGED\_CHALLENGES\_JOB\_CLUSTER\_ENABLED is set to true:



Stop the service and run local/run.sh again, check the console:



The app has got the lock from the cluster sucessfully with the log message:

INFO [2018-01-08 18:20:45,096] com.appirio.service.challengefeeder.job.LoadChangedChallengesJob: Get the lock successfully

You can also deploy the feeder service in another matchine to check only one job will run in the same time.