

JanusGraph单机测试

JanusGraph自带了一个可以开箱使用的JanusServer，里面集成了简化的cassandra和ES，不能作为生产环境，但可以用来学习使用JanusServer。该Janus Server是基于Websocket的，使用步骤如下：

- 下载janusgraph-\$version.zip文件
- <https://github.com/JanusGraph/janusgraph/releases/download/v0.3.1/janusgraph-0.3.1-hadoop2.zip>
- 解压
- 运行bin/janusserver.sh start，该步骤将会启动Gremlin Server及Cassandra/ES到一个单独的进程。

启动之后即支持gremlin客户端通过websocket连接server，最常用的即为gremlin.sh shell。

由于root用户启动不es，换一个用户

- user add janusgraph
- chown -R janusgraph:janusgraph janusgraph-0.3.1-hadoop2

```
$ bin/janusgraph.sh start
ForkingCassandra...
Running`nodetool statusthrift`.. OK (returned exit status 0 and
printed string "running").
ForkingElasticsearch...
Connecting to Elasticsearch(127.0.0.1:9300)... OK (connected to
127.0.0.1:9300).
ForkingGremlin-Server...
Connecting to Gremlin-Server(127.0.0.1:8182)... OK (connected to
127.0.0.1:8182).
Run gremlin.sh to connect.
```

Connecting to Gremlin Server

当运行了janusserver之后，Gremlin Server已经做好准备接受socket连接，最简单的方式是使用gremlin console:、

使用bin/gremlin.sh 启动Gremlin Console，使用:remote命令连接远程服务器；使用:> 提交 gremlin语句。

```
$ bin/gremlin.sh
\,,,/
(o o)
-----o00o-(3)-o00o-----
plugin activated: tinkertop.server
plugin activated: tinkertop.hadoop
plugin activated: tinkertop.utilities
plugin activated: janusgraph.imports
plugin activated: tinkertop.tinkergraph
gremlin>:remote connect tinkertop.server conf/remote.yaml
==>Connected- localhost/127.0.0.1:8182
gremlin>:> graph.addVertex("name", "stephen")
==>v[256]
gremlin>:> g.V().values('name')
==>stephen
```

注意 :> 有一个空格

conf/remote.yaml是配置文件，指明了要连接的地址。