

StaticsServer 安装配置手册

作者: jamiesun

概述

本文档主要是为使用 StaticsServer 的开发人员提供一个安装配置使用的指导，以便于开发人员能正确的使用 StaticsServer 来进行文件存储。本文档主要描述了在 linux 下安装配置过程。

安装前准备

StaticsServer 版本:

当前版本 **staticsV1.0.0.1**

linux 服务器系统

这里采用的是 ubuntu 8.10 x64

系统环境配置

配置图形处理环境

安装 ImageMagick

`sudo apt-get install ImageMagick`

安装 jmagick

`sudo apt-get install jmagick`

`sudo cp /usr/lib/jni/libJMagick.so /usr/lib/`

以上三步是配置服务器的图形处理环境

安装应用系统

在当前用户目录下创建应用目录

```
mkdir statics_node
```

上传 staticsV1.0.0.1.zip 至 statics_node 目录并解压

```
tar -xvf staticsV1.0.0.1.zip  
cd staticsV1.0.0.1
```

修改配置文件 config.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">  
<properties>  
  
    <!-- 存储路径配置，默认是在当前目录下的 statics 目录，第一次会自动创建，  
         也可以配置到其他目录，如系统路径 /opt/staticsdb -->  
    <entry key="store.path">statics</entry>  
  
    <!-- tcp 服务端口，提供客户端连接的端口-->  
    <entry key="tcp.port">8007</entry>  
  
    <!-- http 服务端口，提供 http 读取图片的端口-->  
    <entry key="http.port">9007</entry>  
  
    <!-- http 最大并发连接数，当前服务节点允许的最大连接数 -->  
    <entry key="http.max">1024</entry>  
  
    <!-- 缓存最大天数，http 缓存策略 -->  
    <entry key="cache.expireDay">30</entry>  
  
    <!-- 当前节点名称，多节点时同步用，组内唯一-->  
    <entry key="node.name">ReplicaNode_wjt</entry>  
  
    <!-- 当前节点群组，由多台节点组成-->  
    <entry key="node.group">RepicaGroup</entry>  
  
    <!-- 当前节点相连的节点，当前节点将和这些节点同步数据 -->  
    <!-- 格式说明： host:port,host:port... -->
```

```
<entry key="node.helpers">172.16.0.200:8100</entry>
</properties>
```

同步端口说明，

为了配置文件的易用，对配置文件的同步端口制定了一些规则。

tcp.port 是提供给客户端连接的，http.port 是提供通过 http 访问文件用的

而系统之间的同步监听端口则是 tcp.port+100，如果 tcp.port 是 8000，则同步端口为 8100

在 node.helpers 里就要写 host : 同步端口，

例如，节点 1 tcp.port 为 8000 节点 2 tcp.port 为 9000，节点 2 要与节点 1 同步，则要配置 node.helpers 为：节点 IP : 8100 而不是：节点 IP : 8000

修改日志配置文件 log4j.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<!DOCTYPE log4j:configuration PUBLIC "-//ZLAB //DTD Log4j Configuration 1.0.1//EN"
"http://www.ly-bns.net/dtd/log4j.dtd">
```

```
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
```

```
  <appender name="console.log" class="org.apache.log4j.ConsoleAppender">
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d [%t] %p - %m%n" />
    </layout>
  </appender>
```

```
<appender name="error.log" class="org.apache.log4j.DailyRollingFileAppender">
  <param name="File" value="logs/error.log" />
  <param name="Append" value="true" />
  <param name="DatePattern" value="'.yyyy-MM-dd'"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d [%t] %p - %m%n" />
  </layout>
  <filter class="org.apache.log4j.varia.LevelRangeFilter">
    <param name="LevelMin" value="WARN" />
    <param name="LevelMax" value="FATAL" />
  </filter>
</appender>
```

```

        </filter>
    </appender>

    <appender name="info.log" class="org.apache.log4j.DailyRollingFileAppender">
        <param name="File" value="logs/info.log" />
        <param name="Append" value="true" />
        <param name="DatePattern" value="'yyyy-MM-dd"/>
        <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d [%t] %p - %m%n" />
        </layout>
        <filter class="org.apache.log4j.varia.LevelRangeFilter">
            <param name="LevelMin" value="DEBUG" />
            <param name="LevelMax" value="INFO" />
        </filter>
    </appender>

    <root>
        <level value="info" />
        <appender-ref ref="console.log" />
        <appender-ref ref="error.log" />
        <appender-ref ref="info.log" />
    </root>

</log4j:configuration>

```

服务命令

启动

[./startup.sh](#)

查看状态

[./status.sh](#)

```

usns@usnsserver:~/statics/staticsV1.0.0.1$ ./status.sh
usns      2362      1  5 17:29 pts/0    00:00:40 ../jdk1.5.0_14/bin/statics -server
-Xms64m                                -Xmx1024m                                -classpath
../jdk1.5.0_14/lib/dt.jar:../jdk1.5.0_14/lib/tools.jar:./lib/statics.jar:./lib/xSocket.jar:./lib
/log4j.jar:./lib/commons-
logging.jar:/usr/share/java/jmagick6.jar:./lib/xlightweb.jar:./lib/picocontainer-
1.3.jar:./lib/gson-1.5.jar:./lib/je-4.0.103.jar:./lib/xSocket-multiplexed-2.1.7.jar

```

com.ly.statics.server.Main

存储库 shell 工具命令

./StoreShell

```
usns@usnsserver:~/statics/staticsV1.0.0.1$ ./StoreShell.sh
StoreShell>
```

有用的几个命令

StoreShell>status

```
NodeAddressMap:
ReplicaNode_200=>172.16.0.200/172.16.0.200:8000
ReplicaNode_254=>172.16.0.254/172.16.0.254:8102
ReplicaNode_254Monitor=>/172.16.0.254:8202
```

```
ActiveServerAddresses:
172.16.0.200/172.16.0.200:8000
/172.16.0.254:8002
master:ReplicaNode_200=>/172.16.0.200:8000
```

上面显示的是当前服务器群组信息

NodeAddressMap 是当前所有的节点

ReplicaNode_* 为存储节点

ReplicaNode_*Monitor 为监控节点

这些节点信息可能包含已经停止的节点信息，只做参考

ActiveServerAddresses 是当前在线的节点

master 为当前主节点

StoreShell>size

101

查看当前存储的总记录数

StoreShell>list

```
StaticFile[sid=01d3df97-5372-4800-bc6d-
5b6922a08167;type=image/jpg;meta={uuCode=110168};small=1283bc30-93b2-
4478-99ed-f4f6e0680d55;medium=1ef887a8-a711-4d74-999a-
```

```
c768c5c702d4;original=af977ed5-a6b5-4936-8715-c196a497dc04;]  
StaticFile[sid=01d95360-591f-41a6-9e74-  
023c03d13e6a;type=image/jpg;meta={uuCode=110168};small=a4fea5f2-8694-4eeb-  
8f10-f9cf289aac72;medium=ed3a4e36-4e97-4eea-a5ce-  
59b095648cd7;original=5825b178-4d83-480b-9a15-6287ba86641f;]  
...
```

查看最新的 10 条记录

list 100 查看 100 条记录，依次类推

StoreShell>get 01d3df97-5372-4800-bc6d-5b6922a08167

```
StaticFile[sid=01d3df97-5372-4800-bc6d-  
5b6922a08167;type=image/jpg;meta={uuCode=110168};small=1283bc30-93b2-  
4478-99ed-f4f6e0680d55;medium=1ef887a8-a711-4d74-999a-  
c768c5c702d4;original=af977ed5-a6b5-4936-8715-c196a497dc04;]
```

通过 sid 查询记录

StoreShell>del 01d3df97-5372-4800-bc6d-5b6922a08167

true

通过 sid 删除，小心使用

StoreShell>quit

退出终端

停止服务

[./shutdown](#)

访问文件

文件 url 规则为：

<http://nodeAddress:httpPort/{sid}> //原始文件

<http://nodeAddress:httpPort/{sid}/m> //中等质量(640*480)文件（图片有效）

<http://nodeAddress:httpPort/{sid}/m> //小文件(96*96)（图片有效）

一般对于多个服务节点，最好在前端配置一台 apache 或 nginx 作为反向代理，