

一、 服务器配置

1. Tomcat 服务配置 (安装后主目录必须是/opt/tomcat/)

一：安装 JDK

(截止编写, 参考自 <http://topspeedsnail.com/ubuntu16-install-java-jdk/>)、
如何安装 openjdk：

更新软件包列表：

```
$ sudo apt-get update
```

安装 openjdk-8-jdk：

```
$ sudo apt-get install openjdk-8-jdk
```

查看 java 版本：

```
$ java -version
```

编辑/etc/profile

```
$ sudo vim /etc/profile
```

在文件尾添加 java 环境变量：

```
export JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre/bin"
```

二：安装 eclipse (暂时我还没有用到 ide 可以先不装)

三：安装 Apache 服务

(参考自 <https://cnbin.github.io/blog/2015/07/03/ubuntu-14-dot-04-an-zhuang-apache/>)

```
sudo apt-get install apache2
```

四：安装 Tomcat 服务

(参 考 自 <https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-8-on-ubuntu-16-04>)

首先，更新 apt：

```
$sudo apt-get update
```

安装 jdk:(可跳过,如果已有 jdk)

```
$sudo apt-get install default-jdk
```

新建 tomcat 用户组:

\$sudo groupadd tomcat

新建 tomcat 用户[假定家目录是/opt/tomcat/ tomcat 就安装在这里 以后都工作基本都在这里]

\$sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat

转移当前目录为 tem:

\$cd /tmp

从 t o m c a t 官网下载最新的 tomcat

新建安装目录:

\$sudo mkdir /opt/tomcat

解压安装文件:

\$sudo tar xzvf apache-tomcat-8.tar.gz -C /opt/tomcat --strip-components=1

切换目录:

\$cd /opt/tomcat

让 tomcat 组内成员获得对整个安装目录的拥有权

\$sudo chgrp -R tomcat /opt/tomcat

让 tomcat 组内成员对 conf 目录拥有读和执行权限

\$sudo chmod -R g+r conf

\$sudo chmod g+x conf

让 tomcat 用户拥有以下文件目录:

\$sudo chown -R tomcat webapps/ work/ temp/ logs/

tomcat 运行基于 java,因此需要我们指定 java 的安装目录给它,便于 tomcat 调用 jre

使用以下命令查看 java 版本:

sudo update-java-alternatives -l

输出可能是这样的:

```
java-1.8.0-openjdk-amd64          1081          /usr/lib/jvm/java-1.8.0-openjdk-  
amd64
```

你的 JAVA_HOME= 输出+"/jre" 比如上述输出对应的家目录是

JAVA_HOME= /usr/lib/jvm/java-1.8.0-openjdk-amd64/jre

利用以上信息修改配置文件:

\$sudo nano /etc/systemd/system/tomcat.service

复制以下信息到该文件中:(如果和你自己的java 位置不同,记得修改其中的JAVA_HOME)

[Unit]

Description=Apache Tomcat Web Application Container

After=network.target

[Service]

Type=forking

Environment=JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre

Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid

Environment=CATALINA_HOME=/opt/tomcat

Environment=CATALINA_BASE=/opt/tomcat

Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'

Environment='JAVA_OPTS=-Djava.awt.headless=true

Djava.security.egd=file:/dev/./urandom'

ExecStart=/opt/tomcat/bin/startup.sh

ExecStop=/opt/tomcat/bin/shutdown.sh

User=tomcat

Group=tomcat

UMask=0007

RestartSec=10

Restart=always

[Install]

WantedBy=multi-user.target

重新载入配置:

\$sudo systemctl daemon-reload

启动 tomcat:

\$sudo systemctl start tomcat

再次确认没有错误发生:

\$sudo systemctl status tomcat

启用 8080 端口:

\$sudo ufw allow 8080

设置开机启动 tomcat:

sudo systemctl enable tomcat

设置 tomcat 网页管理接口(略过 可以不做)

访问 tomcat 默认主页:

打开浏览器,输入 localhost:8080 或者 127.0.0.1:8080 就可以看见 tomcat 默认主页

2. 数据库配置

需要新建数据库“DFS”，密码自设。

建立下表：

```
CREATE
TABLE
`DEVICE`
(
    `ID` int NOT NULL AUTO_INCREMENT,
    `IP` char(20) NOT NULL DEFAULT '',
    `PORT` int NOT NULL DEFAULT 0,
    `ISONLINE` boolean NOT NULL,
    `RS` int NULL DEFAULT 0 ,
    PRIMARY KEY (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `FRAGMENT` (
    `ID` int NOT NULL,
    `PATH` char(20) NOT NULL DEFAULT '',
    PRIMARY KEY (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `FILE` (
    `ID` int NOT NULL AUTO_INCREMENT,
    `NAME` char(20) NOT NULL DEFAULT '',
    `PATH` char(60) NOT NULL DEFAULT '',
    `ATTRIBUTE` char(10) NOT NULL DEFAULT '',
    `TIME` char(10) NOT NULL DEFAULT '',
    `NOA` int NOT NULL DEFAULT 1,
    `ISFOLDER` boolean NOT NULL DEFAULT false,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `REQUEST` (
```

```

`ID` int NOT NULL AUTO_INCREMENT,
`TYPE` int NOT NULL DEFAULT 0,
`FRAGMENTID` int NOT NULL DEFAULT 0,
`DEVICEID` int NOT NULL DEFAULT 0,
PRIMARY KEY (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `USER` (
  `ID` int NOT NULL AUTO_INCREMENT,
  `NAME` char(20) NOT NULL UNIQUE DEFAULT '',
  `PASSWD` char(20) NOT NULL DEFAULT '',
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE UNIQUE INDEX `idx_FILE_PATH_NAME` ON `DFS`.`FILE` (PATH, NAME)
COMMENT '' ALGORITHM DEFAULT LOCK DEFAULT;

```

```

CREATE UNIQUE INDEX `idx_USER_NAME` ON `DFS`.`USER` (NAME) COMMENT ''
ALGORITHM DEFAULT LOCK DEFAULT;

```

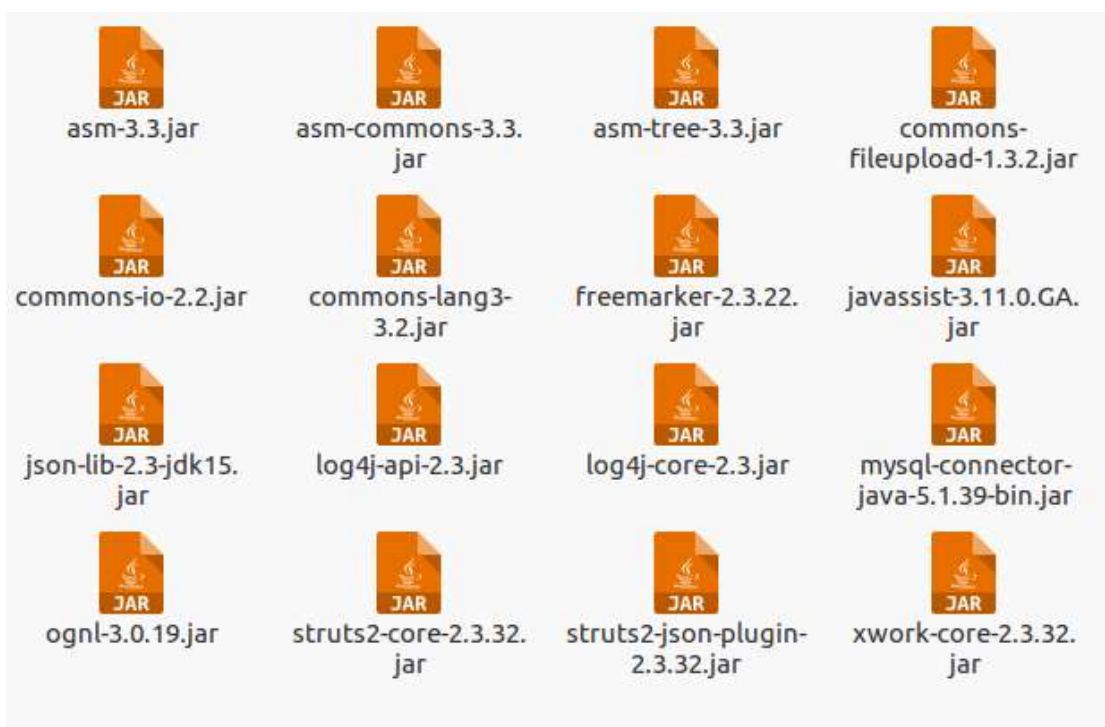
3. 服务器 Java 程序配置

直接运行服务器端 java 程序。

4. 网页文件夹配置

将工程直接 publish 到 tomcat 的 webapp/DFS/目录下

5. Eclipse 编译服务器端基于 struts 框架的 java 程序需要的额外 lib
lib/中增加了的 lib 文件有：



1. struts 的 blank 模板的所有 lib
2. jdbc 相关 lib 1 个
3. json-plugin + json-lib

二、 客户端配置

1. 客户端文件夹配置

文件目录遵循如下格式：

```
|
|-----clinet.jar
|-----setup.ini
|-----CloudDriveClient
|
|---tem/
|---fragment/
|---XXX/
|---XXX
|---
```

其中 XXX 文件夹数目和名称根据你的 setup.ini 而定

2. 客户端配置文件格式

客户端的目前使用其同一文件夹下的 setup.ini 文件进行配置

setup.ini 文件的结构为

{服务器 IP}

{服务器控制链接端口}

{服务器数据链接端口}

{客户端 ID}

{碎片文件夹路径（用于保存服务器分配来的碎片）}

{临时文件夹路径（用于在上传过程中保存本地文件的碎片，上传完成后将被清空）}

{需要监控的上传文件夹数量}

{上传文件夹 1 路径}

{上传文件夹 1 中的文件在分布式文件系统上的 path}

{上传文件夹 2 路径}

{上传文件夹 2 中的文件在分布式文件系统上的 path}

...

三、 特殊提醒 代码针对性修改

1. `struts` 动态方法调用默认关闭 需要在 `struts.xml` 文件中添加：
`<constant name="struts.enable.DynamicMethodInvocation" value="true"></constant>`
开启
2. 客户端连接服务器采用一条控制链接和数据连接，端口号需按照实际情况修改（`setup.ini` 配置文件中）
3. 服务器数据库访问的包中写死了数据库密码，需要根据实际的数据库密码进行修改