

该文档，不包括fastdfs部署，fastdfs参考《FastDFS 单机版环境搭建》文档。

redis、mysql组件如果已经安装不需要重复安装，只需要根据文档做必要的配置。

但对于nginx的安装和配置也参考《FastDFS 单机版环境搭建》，需要支持fastdfs和upload模块。

1. MySQL

1.1 Ubuntu Linux安装MySQL

(1) 安装Mysql Server

```
apt-get install mysql-server
```

(2) 安装Mysql Client

```
apt-get install mysql-client
```

(3) 安装libmysqlclient

```
apt-get install libmysqlclient-dev
```

如果安装mysql-server时没有提示设置密码，得手动设置密码

步骤 1): 输入命令mysql -u root -p指定 root 用户登录 MySQL，输入后按回车键输入密码。如果没有配置环境变量，请在 MySQL 的 bin 目录下登录操作。

步骤 2): 修改密码（比如root密码修改为123456）

```
use mysql;
update user set authentication_string=PASSWORD("123456") where user='root';
update user set plugin="mysql_native_password";
flush privileges;
quit;
```

然后 /etc/init.d/mysql restart 重启mysql，再mysql -u root -p 登录测试密码是否正确。

1.2 Mysql启动/停止/重启

(1) Mysql启动

```
/etc/init.d/mysql start
```

(2) Mysql停止

```
/etc/init.d/mysql stop
```

(3) Mysql重启

```
/etc/init.d/mysql restart
```

1.3 创建用户

不一定要再创建用户，这里学习可以直接使用root。

1.3.1 创建用户

```
CREATE USER username@host IDENTIFIED BY password;
```

说明：

- username：你将创建的用户名
- host：指定该用户在哪个主机上可以登陆，如果是本地用户可用localhost，如果想让该用户可以从任意远程主机登陆，可以使用通配符%
- password：该用户的登陆密码，密码可以为空，如果为空则该用户可以不需要密码登陆服务器

范例：CREATE USER 'darren'@'%' IDENTIFIED BY '123456';

1.3.2 授权

```
GRANT privileges ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;
```

说明：

- privileges：用户的操作权限，如SELECT，INSERT，UPDATE等，如果要授予所有的权限则使用ALL
- databasename：数据库名
- tablename：表名，如果要授予该用户对所有数据库和表的相应操作权限则可用表示，如.*

#设置darren用户拥有0voice_tuchuang数据库所有权限

范例：

```
GRANT ALL PRIVILEGES ON 0voice_tuchuang.* TO 'darren'@'%';  
FLUSH PRIVILEGES;
```

1.4 设置远程连接

1.4.1 修改配置文件

```
vi /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
#注释blind-address
```

```
#bind-address=127.0.0.1(默认是没有注释的)
```

1.4.2 修改远程连接

#查看mysql所有用户

```
select user, host from mysql.user;  
+-----+-----+  
| user          | host          |  
+-----+-----+  
| root          | localhost    |
```

#update修改连接用户的host字段值为'%', 此处以root用户为例。
update user set host='%' where user='root';

1.4.3其他

防火墙原因(云服务器需要放开端口3306)

2. Redis

如果已经安装过redis不需要再重新安装。

2.1 Ubuntu Redis安装

```
#下载
git clone https://gitee.com/mirrors/redis.git
cd redis
git checkout 6.2.3
make
make install
```

编译 hiredis

```
cd ./deps/hiredis
make
make install
```

查看版本命令:

```
redis-server -v
```

显示: Redis server v=6.2.3 sha=00000000:0 malloc=jemalloc-5.1.0 bits=64
build=77053994c60ea3c2

3 网盘项目部署

解压 tuchuang.tar.gz

```
tar -zxvf tuchuang.tar.gz
```

服务器端和客户端程序都在0voice_tuchuang目录。

.

- ├─ 0voice_tuchuang_clear.sql
- ├─ 0voice_tuchuang.sql 需要导入的数据库
- ├─ tc-front **web客户端程序 (编译后的文件)**
- └─ tc-src 后台服务器程序源码 (课程重点要掌握的内容)

部署图床项目tuchuang之前, 确保mysql、redis是可用的。

3.1 客户端部署

3.1.1 安装nodejs

1. 下载nodejs

官方: <https://nodejs.org/en/download/>

如图:




Downloads

Latest LTS Version: **16.14.0** (includes npm 8.3.1)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS
Recommended For Most Users

Current
Latest Features

 Windows Installer node-v16.14.0-x64.msi	 macOS Installer node-v16.14.0.pkg	 Source Code node-v16.14.0.tar.gz
--	--	---

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v16.14.0.tar.gz	

复制下载链接后, 从命令行下载:

```
wget https://nodejs.org/dist/v16.14.0/node-v16.14.0-linux-x64.tar.xz
```

2. 解压node安装包

```
xz -d node-v16.14.0-linux-x64.tar.xz  
tar xf node-v16.14.0-linux-x64.tar
```

3. 创建符号链接, 供直接从命令行访问无需输入路径 (/root/0voice/node-v16.14.0-linux-x64路径是自己的路径, 不要照抄)

```
ln -s /root/0voice/node-v16.14.0-linux-x64/bin/node /usr/local/bin/node  
ln -s /root/0voice/node-v16.14.0-linux-x64/bin/npm /usr/local/bin/npm
```

打印版本:

```
~/0voice/node-v16.14.0-linux-x64/bin# node -v  
v16.14.0
```

```
~/0voice/node-v16.14.0-linux-x64/bin# npm -v  
8.3.1
```

3.1.2 获取web客户端的路径

```
root@iZbp1h2l856zgoegc8rvnhZ:~/tuchuang/0voice_tuchuang# cd tc-front/  
root@iZbp1h2l856zgoegc8rvnhZ:~/tuchuang/tuchuang/tc-front# pwd  
/root/tuchuang/tuchuang/tc-front
```

/root/tuchuang/tuchuang/tc-front 该路径配置到nginx.conf，在2.3.2节时统一配置。

3.2 Redis配置

根据redis课程的知识启动redis。确保redis可以正常访问，切记的是这里redis不要设置密码！

3.3 创建mysql

导入0voice_tuchuang.sql文件

```
root@iZbp1h2l856zgoegc8rvnhZ:~/0voice/tc/tuchuang$ mysql -uroot -p      #登录mysql
```

.....

```
mysql>
```

```
mysql> source /root/tuchuang/tuchuang/0voice_tuchuang.sql; #导入数据库，具体看自己存放的路径
```



查看数据库创建情况：

```
mysql> use 0voice_tuchuang;  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_0voice_tuchuang |  
+-----+  
| file_info                  |  
| share_file_list            |  
| share_picture_list         |  
| user_file_count            |  
| user_file_list             |  
| user_info                  |  
+-----+  
6 rows in set (0.01 sec)
```

3.4 服务端部署

3.4.1 配置tc-src/tc_http_server.conf

具体的ip地址根据自己机器进行配置。

修改配置 tc-src/tc_http_server.conf （具体的ip、port、账号等是自己的，不能照抄）

1. 114.215.169.66是自己服务器对外的地址；
2. mysql数据库 username和password根据自己实际情况；

3. redis一般用默认的即可（第一节的服务程序还是单线程模式，后续随着课程迭代到多线程模式）

HttpListenIP=0.0.0.0

HttpPort=**8081**

ThreadNum=4

dfs_path_client=/etc/fdfs/client.conf

web_server_ip=**114.215.169.66**

web_server_port=80

storage_web_server_ip=**114.215.169.66**

storage_web_server_port=80

#configure for mysql

DBInstances=tuchuang_master,tuchuang_slave

#tuchuang_master

tuchuang_master_host=**127.0.0.1**

tuchuang_master_port=3306

tuchuang_master_dbname=0voice_tuchuang

tuchuang_master_username=**root**

tuchuang_master_password=**123456**

tuchuang_master_maxconnct=4

#tuchuang_slave

tuchuang_slave_host=**127.0.0.1**

tuchuang_slave_port=3306

tuchuang_slave_dbname=0voice_tuchuang

tuchuang_slave_username=**root**

tuchuang_slave_password=**123456**

tuchuang_slave_maxconnct=4

#configure for token

CacheInstances=token,ranking_list

#未读消息计数器的redis

token_host=**127.0.0.1**

token_port=6379

token_db=0

token_maxconnct=4

#群组设置redis

ranking_list_host=127.0.0.1

ranking_list_port=6379

ranking_list_db=0

ranking_list_maxconnct=4

3.4.2 配置nginx

配置fastcgi程序，这里是修改nginx.conf，可以直接使用课程提供的nginx.conf(在tuchuang目录)覆盖自己本地的nginx.conf。

然后在此基础上做修改。

```
user root;
worker_processes 1;

#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;

#pid logs/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;

    #log_format main '$remote_addr - $remote_user [$time_local] "$request" '
    #                '$status $body_bytes_sent "$http_referer" '
    #                '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log logs/access.log main;

    sendfile on;
    #tcp_nopush on;

    #keepalive_timeout 0;
    keepalive_timeout 65;

    # file 10M
    client_max_body_size 10m;
    #gzip on;
    server {
        listen 80;
        server_name localhost;

        #charset koi8-r;

        #access_log logs/host.access.log main;

        index index.html index.htm default.htm default.html;
        #root /home/liaoqingfu/tc-front;
        root /root/tuchuang/0voice_tuchuang/tc-front;

        autoindex off;
```

```

#access_log logs/host.access.log main;
# 指定允许跨域的方法，*代表所有
add_header Access-Control-Allow-Methods *;

# 预检命令的缓存，如果不缓存每次会发送两次请求
add_header Access-Control-Max-Age 3600;
# 带cookie请求需要加上这个字段，并设置为true
add_header Access-Control-Allow-Credentials true;

# 表示允许这个域跨域调用（客户端发送请求的域名和端口）
# $http_origin动态获取请求客户端请求的域 不用*的原因是带cookie的请求不支持*号
add_header Access-Control-Allow-Origin $http_origin;

# 表示请求头的字段 动态获取
add_header Access-Control-Allow-Headers
$http_access_control_request_headers;

#charset koi8-r;

#access_log logs/host.access.log main;

location / {
    root /root/tuchuang/0voice_tuchuang/tc-front;
    index index.html index.htm;
    try_files $uri $uri/ /index.html;
}
location ~/[group([0-9])/M([0-9])([0-9]) {
    ngx_fastdfs_module;
}

location /api/login{
    proxy_pass http://127.0.0.1:8081;
}

location /api/reg{
    proxy_pass http://127.0.0.1:8081;
}

location /api2/upload{
    proxy_pass http://127.0.0.1:8081;
}

location /api/md5{
    proxy_pass http://127.0.0.1:8081;
}

location /api/myfiles{
    proxy_pass http://127.0.0.1:8081;
}

location /api/dealfile{
    proxy_pass http://127.0.0.1:8081;
}

```



```

location /api/sharefiles{
    proxy_pass http://127.0.0.1:8081;
}
location /api/dealsharefile{
    proxy_pass http://127.0.0.1:8081;
}

location /api/sharepic{
    proxy_pass http://127.0.0.1:8081;
}

#error_page 404                /404.html;

# redirect server error pages to the static page /50x.html
#
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root    html;
}

location /api/upload {
    # Pass altered request body to this location
    upload_pass    @api_upload;

    # Store files to this directory
    # The directory is hashed, subdirectories 0 1 2 3 4 5 6 7 8 9 should
exist
    upload_store /root/tmp 1;

    # Allow uploaded files to be read only by user
    upload_store_access user:r;

    # Set specified fields in request body
    upload_set_form_field "${upload_field_name}_name" $upload_file_name;
    upload_set_form_field "${upload_field_name}_content_type"
$upload_content_type;
    upload_set_form_field "${upload_field_name}_path" $upload_tmp_path;

    # Inform backend about hash and size of a file
    upload_aggregate_form_field "${upload_field_name}_md5"
$upload_file_md5;
    upload_aggregate_form_field "${upload_field_name}_size"
$upload_file_size;
    #upload_pass_form_field ".*";
    upload_pass_form_field "^user"; # 把user字段也传递给后端解析处理
    #upload_pass_form_field "^submit$|^description$";
}

# Pass altered request body to a backend
location @api_upload {
    proxy_pass    http://127.0.0.1:8081;
}

}
}

```

重点要修改:

root /root/tuchuang/0voice_tuchuang/tc-front; 对应的web前端页面的路径。

client_max_body_size 10m; 注意这里控制最大的文件上传，本项目主要是存储图片做分享，限制在10M的大小。

停止并启动nginx

```
/usr/local/nginx/sbin/nginx -s stop
```

```
/usr/local/nginx/sbin/nginx
```

<http://114.215.169.66>正常登录画面



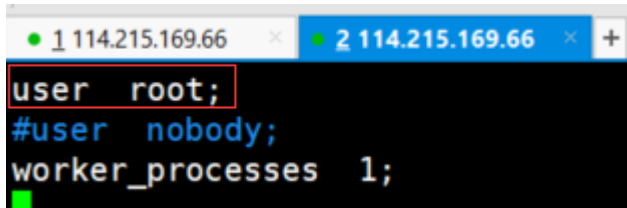
403 Forbidden

nginx/1.16.1

查看nginx日志: `tail -f /usr/local/nginx/logs/error.log`

查看到访问的时候报错: `tail -f /usr/local/nginx/logs/error.log`

原因: root权限的问题, 可以先在nginx.conf加入**user root;权限**。



```
1 114.215.169.66 x 2 114.215.169.66 x +
user root;
#user nobody;
worker_processes 1;
```

然后重新: `/usr/local/nginx/sbin/nginx -s reload`

3.4.3 部署服务器程序

进入tuchuang/tc-src 目录, 使用cmake的方式编译

```
mkdir build
cd build
cmake ..
make
```

编译完成后, 将tuchuang/tc-src 下的tc_http_server.conf拷贝到build目录

```
cp ../tc_http_server.conf .
```

目前先用前台运行的方式方便观察打印信息

```
./tc_http_server
```

如果报错, 比如

redisConnect failed: Connection refused

Init cache pool failed20221101 07:47:47.260652Z 12930 ERROR CacheManager init failed -
main.cpp:79

CacheManager init failed

则说明redis没有连接成功。