

linux面试问题

- 问题
- 问题
- linux命令
- mysql/mariadb
- docker
- Redis
- Nginx
- 网络
- SQL
- node
- dotnet
- consul

linux



```
#
```

```
whoami
```

```
#
```

```
groups
```

```
#
```

```
groups lingtest
```

```
#
```

```
usermod -a -G lingtest lingtest
```

```
#
```

```
gpsswd lingtest -d lingtest
```

```
#
```

```
cat /etc/passwd
```

```
cat /etc/group
```

```
#
```

```
/bin/bash
```

```
/sbin/nologin
```

```
#
```

```
sudo adduser lingtest
```

```
sudo passwd lingtest
```

```
#
```

```
/etc/passwd
```

```
/etc/group
```

```
sudo passwd lingtest
```



```
#root
```

```
sudo chown -R lingtest:lingtest /home/lingtest
```

```
#root
```

```
sudo chmod -R 775 /home/lingtest
```

```
#root nginxroot
```

```
chmod u+s /usr/local/nginx/sbin/nginx
```

```
#root --root -root -root
```

```
sudo visudo
```

```
echo "lingtest ALL=(ALL) NOPASSWD: /usr/local/nginx/sbin/nginx -s reload" >> /etc/sudoers
```

```
sudo visudo -c
```

```
#root
```

```
chmod +x test.bh
```

```
#root
```

```
su - lingtest
```

```
#root
```

```
exit
```

```
#root
```

```
groupdel lingtest
```

```
#root
```

```
userdel lingtest
```

```
#root
```

```
userdel -rf lingtest
```

```
#root
```

```
d      rwx      r-x      r--  25      hades      hades      kkk.txt
```

```
root      root      root -root      root      root      root      root
```




```
#!/bin/bash
```

ANSI

```
GREEN='\033[0;32m'
```

YELLOW='\033[1;33m'

```
RED='\033[0;31m'
```

NC='\033[0m' # No Color

```
REQUESTURL="http://192.168.0.100:7000/api/Cert/CreateCertNode"
```


--	--	--	--	--	--

```
if [ "$#" -ne 3 ]; then
```

```
echo -e "${RED}■■■■ : $0 <username> <password> <temptoken>${NC}"
```

```
exit 1
```

fi

--	--	--	--

USERNAME=\$1

PASSWORD=\$2

TEMPTOKEN=\$3

```
echo "-----"
```


--	--	--	--

```
echo -e "${YELLOW}■■■■ ■■■■■■ ■■■■■■■■■■ ;${NC}"
```

```
echo -e "${YELLOW} ██████████
```

;\$\{NC\}''

```
echo -e "${YELLOW}XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX${NC}"
```

```
echo -e "${YELLOW}XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;${NC}"
```

```
echo -e "${YELLOW}XXXXXXXXXXXXXXXXXXXXXXXXXXXX${NC}"
```

```
echo -e "${YELLOW}XXXXXXXXXXXXXXXXXXXX${NC}"
```

```
echo "-----"
```


--	--	--	--	--	--	--	--

```
echo -e "${GREEN}■■■■■:${NC}"
```

```
HOSTNAME=$(hostname)
```

```
ARCHITECTURE=$(uname -m)
```

```
echo -e "${GREEN}███ : $HOSTNAME${NC}"
```

```
echo -e "${GREEN}■■■■ : $ARCHITECTURE${NC}"
```

```
echo "-----"
```


--	--	--	--	--	--


```
echo -e "${GREEN}███ : $(whoami)${NC}"
```

```
echo "-----"
```

```
# ████
```

```
echo -e "${GREEN}███ : $(pwd)${NC}"
```

```
echo "-----"
```

```
# ██████████
```

```
echo -e "${GREEN}████ $USERNAME ███ ...${NC}"
```

```
sudo useradd -m $USERNAME
```

```
echo "$USERNAME:$PASSWORD" | sudo chpasswd
```

```
if [ $? -ne 0 ]; then
```

```
    echo -e "${RED}██ $USERNAME ██████████ ${NC}"
```

```
    exit 1
```

```
fi
```

```
echo -e "${GREEN}██ $USERNAME ██████████ ${NC}"
```

```
# ██████████
```

```
echo -e "${GREEN}███ /home/$USERNAME ███ ...${NC}"
```

```
sudo chown -R $USERNAME:$USERNAME /home/$USERNAME
```

```
if [ $? -ne 0 ]; then
```

```
    echo -e "${RED}██ /home/$USERNAME ████████ ${NC}"
```

```
    exit 1
```

```
fi
```

```
echo -e "${GREEN}/home/$USERNAME ████████ $USERNAME:$USERNAME ${NC}"
```

```
# ██████████
```

```
echo -e "${GREEN}███ /home/$USERNAME ███ ...${NC}"
```

```
sudo chmod -R 775 /home/$USERNAME
```

```
if [ $? -ne 0 ]; then
```

```
    echo -e "${RED}██ /home/$USERNAME ████████ ${NC}"
```

```
    exit 1
```

```
fi
```

```
echo -e "${GREEN}/home/$USERNAME ████████ 775 ${NC}"
```

```
echo "-----"
```

```
# ███ IP
```

```
PUBLIC_IP=""
```

```
for service in "icanhazip.com" "v4.ident.me" "inet-ip.info" "whatismyip.akamai.com" "ipecho.net/plain"; do
```

```
    echo -e "${GREEN}██ $service ███ IP...${NC}"
```



```

PUBLIC_IP=$(curl -k -s $service)

if [ -n "$PUBLIC_IP" ]; then
    echo -e "${GREEN}██████ IP: $PUBLIC_IP${NC}"
    break
else
    echo -e "${RED} $service  IP  ${NC}"
fi
done

if [ -z "$PUBLIC_IP" ]; then
    echo -e "${RED} IP  ${NC}"
    exit 1
fi

echo -e "${GREEN} IP: $PUBLIC_IP${NC}"
echo "-----"

#  nginx
echo -e "${GREEN}  Nginx  ...${NC}"
NGINX_PATH=$(which nginx)
if [ -z "$NGINX_PATH" ]; then
    echo -e "${RED}Nginx  Nginx  ${NC}"
    exit 1
fi
echo -e "${GREEN}Nginx  : $NGINX_PATH${NC}"
echo "-----"

#  nginx
echo -e "${GREEN}  $USERNAME  Nginx  ...${NC}"
sudo chmod u+s "$NGINX_PATH"
if [ $? -ne 0 ]; then
    echo -e "${RED}  $USERNAME  Nginx  ${NC}"
    exit 1
fi
echo -e "${GREEN}  $USERNAME  Nginx  ${NC}"
NGINX_VERSION=$(sudo $NGINX_PATH -v 2>&1 | grep -oP 'nginx/[0-9.]+')
#
echo "-----"
echo -e "${GREEN}  ${NC}"
echo -e "${GREEN}  ${NC}"
echo "-----"

```



```

echo -e "${GREEN}███ : $HOSTNAME${NC}"
echo -e "${GREEN}████ : $ARCHITECTURE${NC}"
echo -e "${GREEN}███ : $USERNAME${NC}"
echo -e "${GREEN}██ : $PASSWORD${NC}"
echo -e "${GREEN}Nginx ███ : $NGINX_PATH${NC}"
echo -e "${GREEN}Nginx ██ : $NGINX_VERSION${NC}"
echo "-----"

# ██████████
echo -e "${GREEN}████████████████████ ...${NC}"
# ██ API URL ██
API_URL="${REQUESTURL}"
DATA=$(cat <<EOF
{
    "HostName": "$HOSTNAME",
    "Architecture": "$ARCHITECTURE",
    "NginxPath": "$NGINX_PATH",
    "NginxVersion": "$NGINX_VERSION",
    "HostIP": "$PUBLIC_IP",
    "TempToken": "$TEMPTOKEN"
}
EOF
)

# ██ POST ██████
OUTPUT=$(curl -s -X POST -H "Content-Type: application/json" -d "$DATA" "$API_URL")
ERROR=$?

# ██████
if [ $ERROR -ne 0 ]; then
    echo "██████████ : $ERROR"
    exit 1
fi

# ████
echo "██████████ : $OUTPUT"
echo "-----"

# ████
echo -e "${GREEN}██████ ...${NC}"
rm -f "$0"

```



```
echo -e "${GREEN}■■■■■   ${NC}"
```

■■■■■

```
//■■■■■■■■■  
cd -  
tab ■■■■■■
```

■■■■■

```
#■■■  
https://mirrors.ustc.edu.cn/
```

■■■■■

```
crontab -e  
crontab -l  
crontab -r
```

■■■■■

```
ls -a ■■■■■■  
ls -l ■■■■■  
ls -al ■■■■  
ls -lh ■■■■■■  
  
cd ../ ■■■■■  
cd ./ ■■■■■  
cd / ■■■■■  
pwd ■■■■■  
  
mkdir ../ ■■■■■  
mkdir ./ ■■■■■  
mkdir / ■■■■■
```



```

mkdir -p [ ] [ ] [ ] [ ] [ ] [ ] [ ]

#ssh[ ] -[ ] [ ] [ ]
/etc/ssh/sshd.config
#[ ] [ ] [ ] [ ]
hostnamectl
#[ ] [ ] [ ] [ ]
uname -m
#[ ] [ ] [ ] [ ] [ ] [ ]
hostnamectl set-hostname [ ] [ ] [ ] [ ] [ ]

#[ ] [ ] centos [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
setenforce 0;[ ] [ ] [ ] [ ]
/etc/selinux/config;[ ] [ ] [ ] [ ]
selinux=disabled

cd /etc/sysconfig/network-scripts/
//[ ] [ ]
df -Th

```

yum[] []

```

cat > /etc/yum.repos.d/proxysql.repo << EOF
[proxysql]
name=ProxySQL YUM repository
baseurl=https://repo.proxysql.com/ProxySQL/proxysql-2.5.x/centos/\$releasever
gpgcheck=1
gpgkey=https://repo.proxysql.com/ProxySQL/proxysql-2.5.x/repo_pub_key
EOF

```



```

#[ ] [ ] [ ] [ ]
sudo firewall-cmd --zone=public --add-port=3306/tcp --permanent
sudo firewall-cmd --reload
#[ ] [ ] [ ] [ ] [ ] [ ]
sudo firewall-cmd --zone=public --list-ports
sudo firewall-cmd --list-all

```




```
#root
systemctl list-unit-files --type=service
#root
/etc/systemd/system
#root
/usr/lib/systemd/system
#root
sudo systemctl is-enabled mysql
```



```
#root
export PATH=/usr/local/nginx/sbin:$PATH
#root      =root
echo "export PATH=/usr/local/nginx/sbin:$PATH" >> ~/.bash_profile
#root      =root
echo "export PATH=/usr/local/nginx/sbin:$PATH" >> /etc/profile
```



```
#npmroot
npm install cnpm -g
#vue
npm run build
#tree
tree -L 1
```



Ip

```
curl icanhazip.com
```




```
tail -f [ ]
```

systemctl

```
// systemctl systemd
systemctl daemon-reload

// service enabled
sudo systemctl list-unit-files --type=service --state=enabled

// service disabled
sudo systemctl list-unit-files --type=service --state=disabled

// 
sudo systemctl list-units --type=service --state=running

// 
systemctl is-enabled dcloudcast

// 
systemctl enable dcloudcast

// 
/etc/systemd/system

// 
# 
[Unit]
# 
Description=dcloudcast.server
# After network.target
After=network.target
# 
[Service]
# 
WorkingDirectory=/usr/local/share/drawingcloudcasting
# dotnet dotnet
ExecStart=/usr/local/share/dotnet/dotnet8/dotnet
/usr/local/share/drawingcloudcasting/DrawingCloudCasting.RESTApi.dll
# 
Restart=always
# 1GB
# MemoryLimit=1G
#StandardOutput=file:/var/log/dcloudcast.log
#StandardError=file:/var/log/dcloudcast.err
[Install]
```


#

WantedBy=multi-user.target



```
// CPU
nproc

//
ulimit -n

//
ulimit -Hn

//
ulimit -n 16384

// - /etc/security/limits.conf
* soft nofile 16384
* hard nofile 524288

echo "* soft nofile 16384" >> /etc/security/limits.conf
echo "* hard nofile 524288" >> /etc/security/limits.conf

// /etc/systemd/system.conf
[Manager]
DefaultLimitNOFILE=16384
```

which whereis

	which	whereis
	PATH	 /usr/bin , /usr/src , /usr/man

mysql/mariadb



```
#  
mysql_secure_installation  
  
#  
mysqld_safe --skip-grant-tables
```



```
#  
mysqladmin -u root -p password  
  
#  
alter user 'root'@'localhost' identified by '12345678';  
  
#  
grep -i 'temporary password' /var/log/mysqld.log  
  
#  
validate_password.length = 8  
validate_password.mixed_case_count = 1  
validate_password.number_count = 1  
validate_password.special_char_count = 1  
  
# mysql  
SHOW VARIABLES LIKE 'validate_password%';  
  
# mysql  
SET GLOBAL validate_password.policy = LOW;
```



```
#  
create user 'root'@'%' identified by '12345678';  
  
#  
grant all privileges on *.* to 'root'@'%' with grant option;  
  
#  
flush privileges;
```



```
# 删除用户
drop user 'laoda'@'12345678';

# 查看权限
show grants for [用户名];
```

查看日志

```
# 清空日志
sudo truncate -s 0 /var/log/mysql.log

# 查看日志
cat /var/log/mysql.log | grep '2021-09-13'
tail -n500 /var/log/mysql.log|grep -E 'Warning|ERROR'

cat group|grep mysql
cat passwd|grep mysql
```

创建数据库

```
# 创建数据库
desc [数据库名];

# 查看索引
show index from [数据库名];

# 创建表
create table if not exists '[数据库名].[表名]' (
'id' int UNSIGNED AUTO_INCREMENT,
'[数据库名].[表名]' varchar(100) not null,
primary key ('id')
)engine=InnoDB default charset=utf8mb4;

# 删除表
drop table [数据库名].[表名];

# 清空表
truncate table [数据库名].[表名];

# 查看表结构
show create table [数据库名].[表名];

# 插入数据
insert into [数据库名].[表名]
([数据库名].[表名] 1,[数据库名].[表名] 2,[数据库名].[表名] 3)
values
('1','2',now()),
('1','2',now());

# 查看数据
```



```
alter table [table] alter [column] set default [value];

#test

alter table [table] alter [column] drop default;
```

test

```
#test

create database [database] default charset utf8mb4 collate utf8_general_ci;
```

test

test

```
#test

show variables like 'datadir';

#test

show variables like 'datadir';

tar -zcvf mysql.tar.gz mysql

#test

scp mysql.tar.gz root@[ip]:test

#test      Mysqltest

tar -zxvf mysql.tar.gz

#test      centostest
```

test

```
#test      .sqltest

mysqldump -uroot -p12345678 --all-databases > all-first-$(data +%f).sql

#test      .sql

mysql -uroot -p12345678 < all-first.sql

#test log

show variables LIKE '%log%';

#test

#test                                     !!!

log_bin = mysqlbin

# test                                     binlog!!!

sync_binlog = 1

# test      ID!!!

server-id = 1
```



```
# 开启 binlog 功能 ROW!!!
binlog_format = ROW
# 清空日志
flush logs;
# 开启 bin-log
show BINARY LOGS;!!
show MASTER STATUS;
# 查看日志
SHOW BINLOG EVENTS in 'binlog.000002';
# 关闭 bin-log
reset master;
# 清空日志
flush binary logs;
flush logs [binlog 文件] -> [binlog 文件 + 1]
purge binary logs to '1-8';
# 查看日志
mysqlbinlog --no-defaults /var/lib/mysql/[binlog 文件] --start-position=[位置] --stop-position=[位置] | mysql -
uroot -p12345678 [binlog 文件]
# 查看日志
mysqlbinlog /var/lib/mysql/[binlog 文件] --start-position=[位置] --stop-position=[位置] >export.sql
# 查看日志
mysql -uroot -p12345678 < all.first.sql
# 查看日志
source ./export.sql
```



```
# 设置 MySQL 的 ID
server-id=1
# 设置 MySQL 的日志文件路径
log-bin=mysql-bin
# 设置 MySQL 的日志文件大小
read-only=0
# 设置 MySQL 的日志保留时间
expire_logs_days=30
# 设置 MySQL 的日志文件大小
1GB
```



```

max_binlog_size=500M
#[ ]
binlog-ignore-db=test
#[ ] ,
binlog-do-db=atguigu_master_slave
binlog-do-db=
#[ ] binlog
binlog_format=STATEMENT

```



```

#
systemctl restart mysqld
# log_bin
show variables like '%log_bin%';
#
show master status;
#
slave

```



```

#[ ] ID
server-id=2
#[ ]
relay-log=mysql-relay
#
read_only = 1
#
master_info_repository=TABLE
#
relay_log_info_repository=TABLE

```



```

#
stop slave;
#
change master to
master_host='192.168.148.130',
master_user='slave',

```



```
master_password='12345678',
master_port=3306,
master_log_file='mysql-bin.000003',
master_log_pos=346;

#
start slave;

#
show slave status\G;
```



```
#
yum -y install perl-Time-HiRes
yum -y install perl-DBD-Mysql.x86_64
yum -y install libaio
yum -y install rsync
yum -y install lsof
yum -y install libev.so.4*
yum -y install perl-JSON.noarch
yum -y install perl.x86_64
yum -y install perl-devel.x86_64
yum -y install socat
yum -y install net-tools
yum -y install openssl-devel
#
yum -y install libboost_program_options.so*
rpm -ivh percona-xtrabackup*
```



mariadb mysql

Mysql-wsrep



galera-4


```
rpm -ivh galera-4-26.4.16-1.el7.x86_64.rpm
```

```
//
```

```
rpm -qa | grep -E 'galera|mysql|percona'
```



```
#
```

```
find / -name libgalera_smm.so
```

```
# my.cnf
```

```
[mysqld]
```

```
datadir=/var/lib/mysql
```

```
socket=/var/lib/mysql/mysql.sock
```

```
log-error=/var/log/mysqld.log
```

```
pid-file=/var/run/mysqld/mysqld.pid
```

```
net_read_timeout=3600
```

```
net_write_timeout=9000
```

```
max_allowed_packet=10000M
```

```
interactive_timeout=28800000
```

```
wait_timeout=28800000
```

```
max_connections=1000
```

```
wsrep_provider = /usr/lib64/galera-4/libgalera_smm.so
```

```
wsrep_cluster_name="mysql_galera_cluster"
```

```
wsrep_cluster_address= "gcomm://192.168.188.129,192.168.188.128"
```

```
#wsrep_sst_method=xt_rabackup
```

```
wsrep_sst_auth=laoda:12345678
```

```
wsrep_node_name=node1
```

```
wsrep_node_address="192.168.188.129"
```



```
#
```

```
/usr/bin/mysqld_bootstrap
```

```
//
```

```
#
```

```
grastate.dat
```

```
#
```

```
safe_to_bootstrap
```




```
show status like '%wsrep_cluster%';
show variables like 'wsrep_cluster_address';
show variables like 'wsrep_auto_increment_control';
show variables like '%max_allowed_pack%';
show variables like 'innodb_buffer_pool%';
show global status like 'wsrep_local_state_comment';
```



proxysql



```
cat > /etc/yum.repos.d/proxysql.repo << EOF
[proxysql]
name=ProxySQL YUM repository
baseurl=https://repo.proxysql.com/ProxySQL/proxysql-2.5.x/centos/\$releasever
gpgcheck=1
gpgkey=https://repo.proxysql.com/ProxySQL/proxysql-2.5.x/repo_pub_key
EOF
```



```
#
insert into mysql_replication_hostgroups ( writer_hostgroup, reader_hostgroup, comment) values (10,20,'proxy');
#
insert into mysql_servers(hostgroup_id,hostname,port,comment) values
(10,'192.168.148.130',3306,masterinstance);
insert into mysql_servers(hostgroup_id,hostname,port,comment) values
(20,'192.168.148.130',3306,slaveinstance);
#
set mysql-monitor_username='monitor';
set mysql-monitor_password='12345678';
#
insert into mysql_users (username,password,default_hostgroup) values ('proxysql','12345678',10);
#
insert into mysql_query_rules(rule_id,active,match_pattern,destination_hostgroup,apply) values
(1,1,'^select.*for update$',10,1);
insert into mysql_query_rules(rule_id,active,match_pattern,destination_hostgroup,apply) values
```



```
(2,1,'^select',20,1);
#
set global read_only=1;
set global read_only=0;
#
show global variables like 'read_only';
```



```
#
select * from monitor.mysql_server_connect_log;# connect
select * from mysql_server_ping_log limit 10; # ( ping )
select * from mysql_server_read_only_log limit 10; # read_only
#
select hostgroup,schemaname,username,digest_text,count_star from stats_mysql_query_digest;
SELECT hostgroup,schemaname,digest,digest_text,count_star,sum_time FROM stats_mysql_query_digest ORDER
BY sum_time DESC;
#
show tables from monitor;
```



```
#
load mysql users to runtime;
save mysql users to disk;
load mysql servers to runtime;
save mysql servers to disk;
load mysql query rules to runtime;
save mysql query rules to disk;
load mysql variables to runtime;
save mysql variables to disk;
load admin variables to runtime;
save admin variables to disk;
```


docker

docker

```
#  
# step 1:  
sudo yum install -y yum-utils device-mapper-persistent-data lvm2  
# Step 2:  
sudo yum-config-manager --add-repo https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo  
# Step 3  
sudo sed -i 's+download.docker.com+mirrors.aliyun.com/docker-ce+' /etc/yum.repos.d/docker-ce.repo  
# Step 4: Docker-CE  
sudo yum makecache fast  
sudo yum -y install docker-ce  
# Step 4: Docker  
sudo service docker start
```

docker- images

```
#  
# step 1:  
sudo yum install -y yum-utils device-mapper-persistent-data lvm2  
# Step 2:  
sudo yum-config-manager --add-repo https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo  
# Step 3  
sudo sed -i 's+download.docker.com+mirrors.aliyun.com/docker-ce+' /etc/yum.repos.d/docker-ce.repo  
# Step 4: Docker-CE  
sudo yum makecache fast  
sudo yum -y install docker-ce  
# Step 4: Docker  
sudo service docker start
```

docker- container


```

#容器  -容器
docker run -it --name [容器 ] [容器 ] /bin/bash
docker run -itd --name [容器 ] [容器 ] /bin/bash  !!!容器

#容器
docker rm [容器 /容器 id]
docker rm -f [容器 /容器 id] ||容器

#容器
docker exec -it [容器 id] /bin/bash

#容器
docker ps

#容器
docker ps -a

#容器 -容器 -容器 -容器
docker start/stop/restart [容器 id]

#容器
docker rm -f (docker ps|grep [容器 ] |awk '{print $1}')
```

docker容器

```

#
cd /etc/docker/daemon.json

#
{}

https://docker.nju.edu.cn/

{}

https://docker.mirrors.sjtug.sjtu.edu.cn/

{
  "registry-mirrors": ["https://docker.nju.edu.cn","https://docker.mirrors.sjtug.sjtu.edu.cn"]
}
```


Redis

redis-cli

```
#  
config set requirepass [ ]  
config rewrite  
#  
redis-cli  
#  
info  
#  
auth [ ]  
#  
set [key] [value]  
#  
get [key]  
#  
keys *  
#  
expire [key] [ ]  
#  
TTL [key]  
#  
select [0-15]
```

redis

```
#  
wget ....  
# gcc  
#  
make distclean  
make clean  
#  
cd deps
```



```
make hdr_histogram
```

```
wget http://downloads.sourceforge.net/tcl/tcl8.6.1-src.tar.gz
```

```
sudo tar xzvf tcl8.6.1-src.tar.gz -C /usr/local/
```

```
cd /usr/local/tcl8.6.1/unix/
```

```
sudo ./configure
```

```
sudo make
```

```
sudo make install
```

```
# redis
```

```
redis-server [ ]
```

```
#
```

```
redis-cli
```

```
#
```

```
redis-cli shutdown
```


Nginx



```
//安装依赖
yum install pre1
yum install pcre pcre-devel
sudo yum install libxml2 libxml2-devel libxslt libxslt-devel
sudo yum install gd gd-devel

//编译安装
nginx -V

//配置
make clean

//配置参数
./configure \
--prefix=/usr/share/nginx \
--sbin-path=/usr/sbin/nginx \
--modules-path=/usr/lib64/nginx/modules \
--conf-path=/etc/nginx/nginx.conf \
--error-log-path=/var/log/nginx/error.log \
--http-log-path=/var/log/nginx/access.log \
--http-client-body-temp-path=/var/lib/nginx/tmp/client_body \
--http-proxy-temp-path=/var/lib/nginx/tmp/proxy \
--http-fastcgi-temp-path=/var/lib/nginx/tmp/fastcgi \
--http-uwsgi-temp-path=/var/lib/nginx/tmp/uwsgi \
--http-scgi-temp-path=/var/lib/nginx/tmp/scgi \
--pid-path=/run/nginx.pid \
--lock-path=/run/lock/subsys/nginx \
--user=nginx \
--group=nginx \
--with-compat \
--with-debug \
--with-file-aio \
--with-http_addition_module \
--with-http_auth_request_module \
--with-http_dav_module \
```



```

--with-http_degradation_module \
--with-http_flv_module \
--with-http_gunzip_module \
--with-http_gzip_static_module \
--with-http_image_filter_module=dynamic \
--with-http_mp4_module \
--with-http_perl_module=dynamic \
--with-http_random_index_module \
--with-http_realip_module \
--with-http_secure_link_module \
--with-http_slice_module \
--with-http_ssl_module \
--with-http_stub_status_module \
--with-http_sub_module \
--with-http_v2_module \
--with-http_xslt_module=dynamic \
--with-mail=dynamic \
--with-mail_ssl_module \
--with-pcre \
--with-pcre-jit \
--with-stream_ssl_module \
--with-stream_ssl_preread_module \
--with-stream \
--with-threads \
--with-cc-opt='-O2 -flto=auto -ffat-lto-objects -fexceptions -g -grecord-gcc-switches -pipe -Wall -Werror=format-
security -Wp,-D_FORTIFY_SOURCE=2 -Wp,-D_GLIBCXX_ASSERTIONS -specs=/usr/lib/rpm/redhat/redhat-
hardened-cc1 -fstack-protector-strong -specs=/usr/lib/rpm/redhat/redhat-annobin-cc1 -m64 -march=x86-64-v2 -
mtune=generic -fasynchronous-unwind-tables -fstack-clash-protection -fcf-protection' \
--with-ld-opt='-Wl,-z,relro -Wl,--as-needed -Wl,-z,now -specs=/usr/lib/rpm/redhat/redhat-hardened-ld -
specs=/usr/lib/rpm/redhat/redhat-annobin-cc1 -Wl,-E' \
--add-module=/home/nginx-http-flv-module-master \
--add-module=/home/nginx-vod-module-master
//[] Makefile[]
make
//[]
make install

```

dhparam

nginx-http-flv-module

```
# http server
server {
    listen 443 ssl;
    listen [::]:443 ssl;
    http2 on;
    server_name rtmp.lingyanspace.com;

    location /live {
        flv_live on;
        # 'Transfer-Encoding: chunked'
        chunked_transfer_encoding on;
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Credentials' 'true';
    }

    location /vod {
        types {
            application/vnd.apple.mpegurl m3u8;
            video/mp2t ts;
        }
        alias /home/lingyanassets/vods;
        expires -1;
    }

    # control
    location /control {
        rtmp_control all;
    }

    location /stat {
        #
        rtmp_stat all;
        rtmp_stat_stylesheet stat.xml;
    }

    location /stat.xml {
```



```

        root /home/nginx-http-flv-module-master/;
    }
}
#rtmp server
server {
    listen 1935;
    chunk_size 4096;
    application live {
        #
        live on;
        #
        allow publish all;
        #
        allow play all;
        #
        record off;
        # HLS
        hls on;
        # HLS
        hls_path /home/lingyanassets/lives;
        # 10
        hls_fragment 10s;
        # 60
        hls_playlist_length 60m;
        #
        hls_continuous on;
        #
        hls_cleanup off;
    }
    # 'video'
    application vod {
        play /home/lingyanassets/vods; #
    }
}

```

nginx-vod-module-master


```
server {

    listen 443 ssl;
    listen [::]:443 ssl;
    http2 on;
    server_name hls.lingyanspace.com;


    # 512MB
    vod_metadata_cache metadata_cache 512m;
    # 256MB
    vod_response_cache response_cache 256m;
    # VOD
    vod_mode local;
    # 2
    vod_segment_duration 2000;
    #
    vod_align_segments_to_key_frames on;


    #

    location /LimitVideo/ {

        auth_request /test_auth;


        alias /home/lingyanspace/LimitVideo/;

        vod hls;

        add_header Access-Control-Allow-Headers '*';

        add_header Access-Control-Expose-Headers 'Server,range,Content-Length,Content-Range';

        add_header Access-Control-Allow-Methods 'GET, HEAD, OPTIONS';

        add_header Access-Control-Allow-Origin '*';
```



```

        expires 100d;

    }

    location =/test_auth {

        proxy_pass http://127.0.0.1:7777/LimitVideo;

        proxy_pass_request_body off;

        proxy_set_header X-Original-URI $request_uri;

        proxy_set_header X-Original-Method $request_method;

        proxy_set_header X-Forwarded-Proto $scheme;

    }

    access_log /var/log/nginx/hls.lingyanspace.com_access.log;

    error_log /var/log/nginx/hls.lingyanspace.com_error.log;

}

```

nginx

[Unit]

Description=The nginx HTTP and reverse proxy server

After=network-online.target remote-fs.target nss-lookup.target

Wants=network-online.target

[Service]

Type=forking

PIDFile=/run/nginx.pid

Nginx will fail to start if /run/nginx.pid already exists but has the wrong

SELinux context. This might happen when running `nginx -t` from the cmdline.

https://bugzilla.redhat.com/show_bug.cgi?id=1268621

ExecStartPre=/usr/bin/rm -f /run/nginx.pid


```
ExecStartPre=/usr/sbin/nginx -t
ExecStart=/usr/sbin/nginx
ExecReload=/usr/sbin/nginx -s reload
KillSignal=SIGQUIT
TimeoutStopSec=5
KillMode=mixed
PrivateTmp=true
```

[Install]

```
WantedBy=multi-user.target
```

nginx 信号

```
location ~* /(lingyanchat)$ {
    proxy_pass http://lingyanasp;

    # Host
    proxy_set_header Host $host:$server_port;
    # HTTP/1.1 ,
    proxy_http_version 1.1;
    # WebSocket HTTP/2 , HTTP WebSocket
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}
```

nginx

```
user nginx; # nginx

worker_processes auto;# worker ( 4 8G )
worker_cpu_affinity auto;#

error_log /var/log/nginx/error.log warn; # warn
pid /var/run/nginx.pid;

worker_rlimit_nofile 16384;# nginx -- worker_processes * worker_connections
```



```
events {
    use epoll;

    worker_connections 4096; # [redacted] worker_processes *
1024

    multi_accept on; # [redacted]

} # [redacted]

http {
    include mime.types; # [redacted] MIME [redacted]
    default_type application/octet-stream; # [redacted] MIME [redacted]

    charset utf-8; # [redacted]
    # [redacted]

    log_format main '{"@timestamp": "$time_iso8601",'
        '"host": "$server_addr",'
        '"clientip": "$remote_addr",'
        '"size": $body_bytes_sent,'
        '"responsetime": $request_time,'
        '"upstreamtime": "$upstream_response_time",'
        '"upstreamhost": "$upstream_addr",'
        '"http_host": "$host",'
        '"url": "$uri",'
        '"xff": "$http_x_forwarded_for",'
        '"referer": "$http_referer",'
        '"agent": "$http_user_agent",'
        '"status": "$status"}';

    access_log /var/log/nginx/access.log main;

    sendfile on; # [redacted] sendfile [redacted]
    tcp_nopush on; # [redacted] TCP_CORK [redacted]
    tcp_nodelay on; # [redacted] TCP_NODELAY [redacted] Nagle [redacted]

    server_names_hash_bucket_size 128; # [redacted] server_names [redacted] 128 [redacted]
    server_names_hash_max_size 512; # [redacted] server_names [redacted] 512 [redacted]
    keepalive_timeout 65; # [redacted] HTTP [redacted] 65 [redacted]
    client_header_timeout 15M; # [redacted] 15 [redacted]
    client_body_timeout 15M; # [redacted] 15 [redacted]
    send_timeout 60s; # [redacted] 60 [redacted]
```



```

limit_conn_zone $binary_remote_addr zone=perip:10m; # [ ] IP [ ]
10MB
limit_conn_zone $server_name zone=perserver:10m; # [ ] 10MB
limit_conn perip 2; # [ ] IP [ ] 2 [ ]
limit_conn perserver 20; # [ ] 20 [ ]
limit_rate 300k; # [ ] 300KB/s

proxy_cache_path /data/nginx-cache levels=1:2 keys_zone=nginx-cache:20m max_size=50g inactive=168h;
# [ ]

client_body_buffer_size 512k; # [ ] 512KB
client_header_buffer_size 4k; # [ ] 4KB
client_max_body_size 1G; # [ ] 512KB
large_client_header_buffers 2 8k; # [ ]
proxy_connect_timeout 5s; # [ ] 5 [ ]
proxy_send_timeout 120s; # [ ] 120 [ ]
proxy_read_timeout 120s; # [ ] 120 [ ]
proxy_buffer_size 16k; # [ ]
proxy_buffers 4 64k; # [ ]
proxy_busy_buffers_size 128k; # [ ]
proxy_temp_file_write_size 128k; # [ ]
proxy_next_upstream http_502 http_504 http_404 error timeout invalid_header; #
[ ]

gzip on; # [ ] gzip [ ]
gzip_min_length 1k; # [ ] 1KB
gzip_buffers 4 16k; # [ ] gzip [ ]
gzip_http_version 1.1; # [ ] HTTP/1.1 [ ] gzip [ ]
gzip_comp_level 4; # [ ] gzip [ ] 4[ ]
gzip_types text/plain text/css application/json application/x-javascript text/xml application/xml
application/xml+rss text/javascript; # [ ] gzip [ ] MIME [ ]
gzip_vary on; # [ ] Vary: Accept-Encoding[ ]

fastcgi_buffers 64 4K; # [ ] FastCGI [ ]
open_file_cache max=2000 inactive=10m; # [ ] 2000[ ] 10 [ ]
open_file_cache_valid 4m; # [ ] 4 [ ]
open_file_cache_min_uses 2; # [ ] 2[ ]
open_file_cache_errors on; # [ ]
aio on; # [ ] I/O

```



```
proxy_set_header Host $host; # 空 Host 空
proxy_set_header X-Real-IP $remote_addr; # 空 IP 空
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; # 空 IP 空
proxy_set_header X-Forwarded-Proto $scheme; # 空 HTTP 空 HTTPS空

# 空 *.lingyanspace.com空 HTTP空
server {
    listen 80;
    listen [::]:80;
    #空
    server_name *.lingyanspace.com;
    #空 http空 https
    return 301 https://$host$request_uri;
}

# 空 *.lingyanspace.com空 HTTPS空
server {

    listen 443 ssl;
    listen [::]:443 ssl;
    http2 on;
    server_name *.lingyanspace.com;

    # SSL空
    ssl_certificate /etc/nginx/ssl/lingyanspace/GeneralFullChainCert.pem;
    ssl_certificate_key /etc/nginx/ssl/lingyanspace/GeneralPrivateKey.pem;
    ssl_dhparam /etc/nginx/ssl/dhparam.pem;

    #空 SSL空
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
    ssl_prefer_server_ciphers on;

    #ocsp stapling空

    ssl_stapling on;
```



```
ssl_stapling_verify on;

resolver 223.5.5.5 223.6.6.6 119.29.29.29 119.28.28.28 180.76.76.76 1.2.4.8 114.114.114.114 1.1.1.1
1.0.0.1 valid=300s;#[] [] DNS[] [] cloudflare

resolver_timeout 5s;

}

#[] conf.d>[] .http.conf[]

include conf.d/*.http.conf;

}
```

[illegible]

```
ps aux | grep nginx | grep worker | awk '{print $2}' | xargs -I {} cat /proc/{}/limits | grep "Max open files"
```




windows

```
//shell- 1
#
$folderPath = "D:\      \      \
\webapi\LingYanSpace\LingYanSpace.Domain\UnauthorizedFolderHost\SysUserStaticResource"

#
$files = Get-ChildItem -Path $folderPath -File

#
$counter = 1

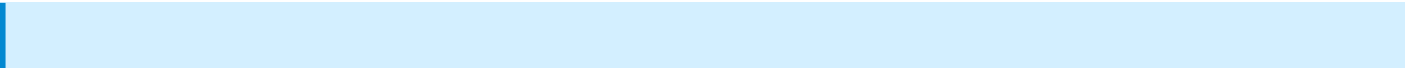
#
foreach ($file in $files) {
    #
    $newFileName = "$counter.jpg"
    $newFilePath = Join-Path -Path $folderPath -ChildPath $newFileName

    #
    Rename-Item -Path $file.FullName -NewName $newFileName

    #
    $counter++
}
```

linux

```
//
ls -l | awk '{print $9}' | sort -k1.1nr
```




```
@echo off
setlocal enabledelayedexpansion
```

```
rem [ ]
for /r %%a in (.) do (
    set "dir=%%a"
    rem [ ]
    dir "!dir!" /a -d /b >nul 2>&1
    if errorlevel 1 (
        rem [ ]      dir[ ]      errorlevel[ ] 1[ ]
        echo [ ]      !dir!
        rmdir "!dir!"
    )
)

endlocal
```


SQL



```
UPDATE CourseFiles
```

```
SET FilePath = REPLACE(FilePath, 'https://hls.lingyanspace.com/LimitVideo',  
'https://dynamicapi.lingyanspace.com/AuthorizedFolderHost/LimitVideo')
```

```
WHERE FilePath LIKE '%https://hls.lingyanspace.com/LimitVideo%';
```


node



//

```
wget https://npmmirror.com/mirrors/node/v20.18.0/node-v20.18.0-linux-x64.tar.xz
```

//

```
tar -xvf node-v20.18.0-linux-x64.tar.xz
```

//

```
export PATH=/usr/local/node/bin:$PATH
```

//

```
echo "export PATH=/usr/local/node/bin:$PATH" >> ~/.bash_profile
```

//

```
source ~/.bash_profile
```


dotnet

在 centos7.9 上

libstdc++.so

```
// 1. libstdc++.so
cd /usr/lib64
// 2. 创建软链接
ln libstd*
// 3. 创建软链接
rm -rf libstdc++.so.6
// 4. 创建软链接
ln -s libstdc++.so.6 /usr/lib64/libstdc++.so.6.0.26
// 5. 创建软链接
ldconfig
```

安装 glibc 2.18

```
$ wget http://ftp.gnu.org/gnu/glibc/glibc-2.18.tar.gz
$ tar -xvf glibc-2.18.tar.gz
$ cd glibc-2.18
$ mkdir build && cd build && ../configure --prefix=/usr && make -j4 && make install
$ strings /lib64/libc.so.6 |grep GLIBC_
```

安装 dotnet

```
// 1. 安装 dotnet
whereis dotnet
// 2. 安装 dotnet
PATH=PATH
```



```
echo $PATH
```

```
//[ ] [ ] [ ] [ ] dotnet[ ] [ ] [ ] [ ]
```

```
echo "export PATH=/usr/local/share/dotnet/dotnet8:/usr/local/share/dotnet/dotnet8/tools:$PATH" >>
~/.bash_profile
```



```
source ~/.bash_profile
```


--	--	--	--

```
dotnet publish -r linux-x64 /p:PublishSingleFile=true
```

//

--	--

```
./[ ] --urls http://0.0.0.0:[ ]
```

```
#nohup
```

```
nohup ./[ ] --urls http://0.0.0.0:[ ] &
```


--	--

```
tail -f nohup.out
```


consul

```
//节点1
```

```
consul agent -server -ui \  
-bootstrap-expect=3 \  
-data-dir=/tmp/consul \  
-node=consul-1 \  
-client=0.0.0.0 \  
-bind=192.168.80.100 \  
-datacenter=dc1
```

```
consul agent -server -ui \  
-bootstrap-expect=3 \  
-data-dir=/tmp/consul \  
-node=consul-2 \  
-client=0.0.0.0 \  
-bind=192.168.80.101 \  
-datacenter=dc1 \  
-join 192.168.80.100
```

```
//节点2
```

```
consul agent -dev -ui -datacenter=dc1
```

```
//节点3 server 节点
```

```
consul agent -server -ui \  
-bootstrap \  
-data-dir=/usr/local/consul/data \  
-node=consul-server \  
-client=0.0.0.0 \  
-bind=127.0.0.1 \  
-datacenter=dc1
```

```
//client节点
```

```
# 节点4 consul-client.hcl  
server = false
```



```
datacenter = "dc1"
node_name = "client-prod-01"
bind_addr = "192.168.148.131" # IP
client_addr = "127.0.0.1" #
data_dir = "/usr/local/consul/clientdata" #
```

```
retry_join = [
  "192.168.148.131",
  "192.168.148.130"
]
```

```
#
check {
  id = "health-check"
  name = "Client Health Status"
  interval = "10s"
  timeout = "1s"
  http = "http://localhost:8500/v1/agent/self"
}
```

```
//
consul members
```

```
// Raft
consul operator raft list-peers
```

```
#
consul catalog nodes -detailed
```

```
#
consul monitor
```



agent	Consul
-server	Raft
-ui	Web 8500
-bootstrap-expect=3	3
-data-dir=/tmp/consul	/tmp
-node=consul-1	consul-1
-client=0.0.0.0	API UI 0.0.0.0
-bind=192.168.80.100	IP
-datacenter=dc1	