

服务器登录

登录工具

- `xshell`
- `MobaXterm`

登录方式

准备好服务器公网IP地址、远程登陆开放端口号、远程登录账号、登录密码

1. 命令行

```
ssh -P [端口号 默认22] [用户名]@[IP]  
回车，输入登录密码（输入登录密码一般是隐藏的）  
eg:  
ssh -P 22 ubuntu@192.168.1.111
```

2. 服务配置

管理工具配置服务：
新建服务/Session
填入以上准备好的服务器登录信息，点击测试登录通过，保存。之后可直接点击打开

3. 可安装openssh服务的电脑

如Mac系统、Ubuntu桌面系统等

打开命令行终端Terminal

1. 执行登录命令，参见第一种方式。
2. 配置文件统一配置

```
# 1. 打开配置文件，命令行下执行以下命令  
>vim ~/.ssh/config // 一般在用户工作目录下  
  
# 2. 复制以下内容到打开的配置文件，用需要登录的远程服务器IP替换“IP地址”，用登录账号替换“用户名”  
  
# 服务器1，Host为自定义名称  
Host server1  
HostName IP地址  
Port 22  
User 用户名  
# 服务器2  
Host server2  
HostName IP地址  
Port 22  
User 用户名
```

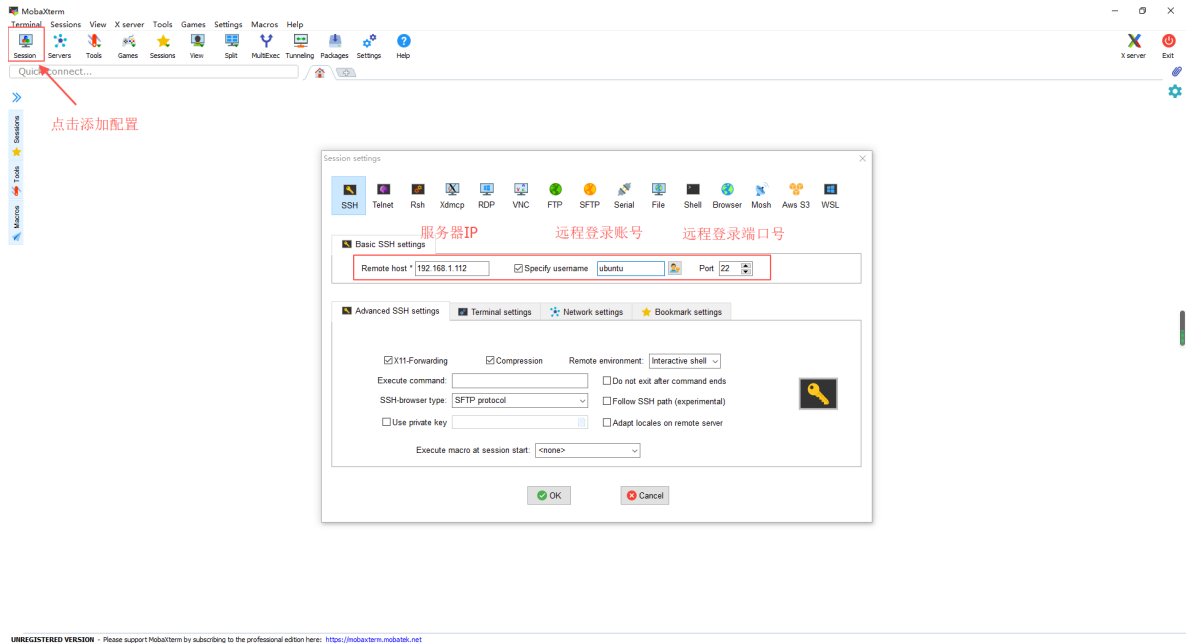
3. 按下ESC退出编辑模式,同时按下Shift+":",当屏幕左下角出现":"时,输入"wq"保存配置并退出,回到命令行。

4. 在终端输入以下命令,按回车,会提示输入登录密码;输入登录密码再次按下回车即可。

```
>ssh server1
```

输入登录密码:

以下是mobaXterm登录方式



点击"ok"按钮确定,就可以直接进入远程登录界面,第一次添加会要求输入登录密码,登录配置也会被保存起来,下次可以直接点击进入。

服务安装与配置

php 7.3.16

ubuntu 系统默认会提供一个低版本的PHP运行环境.如需升级版本,依次执行以下步骤:

1. 首先更新资源

```
sudo apt-get update && sudo apt-get upgrade
```

2. 安装数据库管理工具

```
sudo apt-get install software-properties-common
```

3. 安装最新的PHP数据包

```
sudo add-apt-repository ppa:ondrej/php
```

4. 更新新添加的数据源

```
sudo apt-get update
```

5. 安装指定版本的PHP,及必要的扩展

```
sudo apt-get install php7.3
sudo apt-get install php-pear php7.3-fpm php7.3-curl php7.3-dev php7.3-gd
php7.3-mbstring php7.3-zip php7.3-mysql php7.3-xml
```

以下是Laravel框架需要的PHP环境

```
PHP >= 7.2.5
BCMath PHP Extension
Ctype PHP Extension
Fileinfo PHP extension
JSON PHP Extension
Mbstring PHP Extension
OpenSSL PHP Extension
PDO PHP Extension
Tokenizer PHP Extension
XML PHP Extension
```

composer PHP包管理工具

1. 安装

复制以下代码到服务器终端

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'e5325b19b381bfd88ce90a5ddb7823406b2a38cff6bb704b0acc289a09c8128d4a8ce2bbafc
d1fcbdc38666422fe2806') { echo 'Installer verified'; } else { echo
'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

2. 拷贝命令到执行目录,并赋予执行权限

```
sudo cp ~/.composer.phar /usr/local/bin/composer
sudo chmod +x /usr/local/bin/composer
```

3. 更换数据源

官方默认数据源为国外数据源,访问速度很慢,需要切换到中国镜像.

可以选择以下其中一个

```
https://mirrors.aliyun.com/composer/ # 阿里云数据源,如果使用的服务器是阿里云就使用
阿里云
https://mirrors.cloud.tencent.com/composer/ # 腾讯云数据源,如果使用的云服务器是腾
讯云,则数据源相应的换为腾讯云的数据源

https://packagist.phpcomposer.com # 中国全量镜像
```

配置方式:

```
composer config -g repos.packagist composer  
https://mirrors.aliyun.com/composer/
```

mysql 5.7.29

1. 更新数据源

```
sudo apt-get update
```

2. 安装MySQL服务

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

3. 初始化

```
sudo mysql_secure_installation utility
```

4. 设置MySQL账号root密码

```
# 执行以下命令,进入MySQL查询界面  
sudo mysql  
  
# 输入以下命令,回车  
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY  
'YourPassword';  
  
# 输入"exit;"退出MySQL控制台
```

5. 验证配置是否生效

```
# 输入以下命令,回车  
mysql -uroot -p  
# 终端会提示输入密码,输入密码之后,按下回车,直接进入MySQL控制台,则配置成功.
```

6. 全局配置

打开MySQL配置文件

```
sudo vim /etc/mysql/conf.d/mysql.cnf
```

复制以下脚本粘贴到配置文件,以下配置解决utf8无法存储emoji等特殊字符

```
[mysql]

default-character-set = utf8mb4

[mysqld]
character-set-client-handshake = FALSE
character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci

[client]
default-character-set = utf8mb4
```

按下ESC, Shift+";输入wq保存退出

7. 重启配置生效

```
sudo service mysql restart
```

nginx

nginx版本相对稳定,Ubuntu系统初始化默认已安装,不用重复安装.

配置文件

```
/etc/nginx/nginx.conf # 主配置文件
/etc/nginx/sites-available # 可用服务配置
/etc/nginx/sites-enabled # 生效服务配置,nginx服务一旦启动,该目录下所有配置生效
```

主配置

```
...

user www-data; # 配置服务用户权限
worker_processes auto; # 分配进程数,最大为系统CPU核心数 auto为系统自动分配
pid /run/nginx.pid; # 主进程id
include /etc/nginx/sites-enabled/*.conf; # 引入自定义web服务
events {
    worker_connections 768; # 单进程允许同时建立最大连接数,受操作系统及内存限制,最大可调整数65535
}
http { # web服务主配置
    sendfile on;

    client_max_body_size 20M; # 客户端最大上传数据,如上传文件等

    # 请求超时设置
    fastcgi_connect_timeout 600;
    fastcgi_send_timeout 1200;
    fastcgi_read_timeout 1200;

    # 支持请求数据类型
    include /etc/nginx/mime.types;
    # 默认请求类型
    default_type application/octet-stream;

    # 日志文件配置
```

```

access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

# 开启压缩
gzip on;

# 虚拟机(即web服务)配置
include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
}
...

```

Web服务配置

配置文件格式

```

server {
    listen 80; # 监听端口
    server_name localhost www.example.com; # 绑定域名

    root /var/www/html; # 项目根地址

    index index.php index.html index.htm; # web服务文件查找index配置

    location / { # 路径匹配
        try_files $uri $uri/ /index.php?$query_string;
        #try_files $uri $uri/ =404; # 404页面跳转
    }

    location /api { # nginx反向代理
        proxy_pass http://api.com;

        proxy_set_header Host $host;

        proxy_set_header X-Real-IP $remote_addr;

        proxy_set_header X-Real-Port $remote_port;

        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    location ~ /\.php { # PHP重定向
        fastcgi_pass unix:/run/php/php7.3-fpm.sock; # PHP cgi 服务
        #fastcgi_pass 127.0.0.1:9000; # php cgi 默认监听端口9000
        fastcgi_connect_timeout 60;
        fastcgi_read_timeout 180;
        fastcgi_send_timeout 180;
        fastcgi_buffers 8 128k;
        fastcgi_index index.php;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_param PATH_INFO $fastcgi_path_info;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}

```

1. 新建虚拟机配置

```
sudo touch /etc/nginx/sites-available/website.conf
```

2. 打开刚新建的虚拟机配置文件

```
sudo vim /etc/nginx/sites-available/website.conf
```

3. 切换文件到编辑模式, 按下 `shift+i`, 左下角看到"insert"字样, 将以上server配置拷贝到当前文件, 修改root为需要配置的项目地址

4. 编辑完成, 按下 `ESC`, 再按下 `shift+:`, 左下角看到":", 输入 `wq` 保存退出

5. 将以上设置启用

```
cd /etc/nginx/sites-enabled  
sudo ln -s /etc/nginx/sites-available/website.conf website.conf
```

6. 检查配置文件语法错误, 如见到"successfully"字样, 则配置成功

```
nginx -t
```

7. 重新启动nginx服务

```
sudo service nginx reload
```

8. 查看服务启动状态, 如未见到"error"字样, 则启动成功.

```
sudo service nginx status
```

9. 打开浏览器地址栏输入以上配置域名"www.example.com", 验证配置.