



专题11-配置LNMP/LAMP环境

配置LNMP或LAMP环境，并安装Web应用程序，如开源博客系统WordPress，开源学习平台Moodle等。

介绍

LNMP 和 LAMP 都是常见的 Web 服务器架构，它们在组成、应用场景等方面有一定的区别。

组成架构不同

- LAMP: Linux+Apache+MySQL+PHP（或Perl、Python）。Apache 是一款历史悠久、功能强大的 Web 服务器，具有丰富的模块和配置选项，能支持多种不同的应用场景和需求。
- LNMP: Linux+Nginx+MySQL+PHP（或Perl、Python）。其中 Nginx 是一款轻量级的高性能 Web 服务器及反向代理服务器，具有占用内存少、并发能力强等特点。

性能特点不同

- LAMP: Apache 相对来说在处理动态请求时可能会消耗更多系统资源，在高并发场景下性能可能不如 LNMP，但它对各种 PHP 应用的 **兼容性** 非常好，在一些传统的 PHP 项目中表现稳定。
- LNMP: Nginx 在处理 **静态文件** 和 **高并发连接** 方面性能卓越，能轻松应对大量并发请求，静态文件处理速度比 Apache 快。在高并发场景下，LNMP 架构资源占用相对较少，系统负载较低。

配置复杂程度不同

- LAMP: Apache 的配置文件相对复杂一些，有大量的指令和选项可供配置。对于初学者来说，可能需要花费更多的时间来熟悉和掌握其配置方法。
- LNMP: Nginx 的配置文件相对简洁明了，易于理解和管理。但由于其模块和功能的丰富性，在进行一些高级配置时可能需要对 Nginx 有更深入的了解。

应用场景侧重不同

- LAMP: 更适合于一些对兼容性和稳定性要求较高的传统 PHP 应用，以及对 Apache 的丰富功能有特定需求的项目，如一些企业级的内部管理系统等。

- LNMP：适用于对性能要求较高的大型网站和 Web 应用，特别是在处理高并发请求和静态文件服务方面表现出色，如大型电商网站、新闻资讯平台等。****

安装PHP环境

首先添加PHP源

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

安装PHP8.3及相关的常用包

```
sudo apt install php8.3 php8.3-cli php8.3-common php8.3-curl php8.3-mbstring php8.3-mysql php8.3-x
```

- `php8.3`：这是 PHP 8.3 的主程序包，包含了 PHP 的核心功能和基本模块，是运行 PHP 脚本所必需的。
- `php8.3-cli`：用于在命令行中运行 PHP 脚本的工具包。安装这个包后，可以在终端中直接使用 `php` 命令来执行 PHP 脚本，方便进行一些命令行下的 PHP 脚本测试和开发。
- `php8.3-common`：包含了 PHP 8.3 的一些公共文件和库，这些文件和库被其他 PHP 相关的软件包所共享和依赖，确保 PHP 的各个组件能够正常协同工作。
- `php8.3-curl`：提供了对 `curl` 库的支持，使得 PHP 能够通过 `curl` 函数进行各种网络请求，如获取网页内容、与 API 进行交互等，丰富了 PHP 的网络编程能力。
- `php8.3-mbstring`：用于处理多字节字符串的扩展包。在处理中文等多字节字符编码的文本时非常有用，提供了一系列的函数来进行字符串的转换、编码、解码等操作。
- `php8.3-mysql`：是 PHP 连接和操作 MySQL 数据库的扩展包。安装这个包后，PHP 脚本可以方便地与 MySQL 数据库进行交互，实现数据的存储、查询、更新等操作。
- `php8.3-xml`：提供了对 XML 文档的处理支持，使得 PHP 能够解析、生成和操作 XML 格式的数据，方便与其他系统进行数据交换和集成。

`-y`：这个参数是 `apt` 命令的一个选项，表示在安装过程中自动回答“是”（Yes），即对于安装

过程中可能出现的提示信息，如确认安装、覆盖文件等，都自动选择“是”，从而实现无人值守的自动安装，避免了手动输入确认信息的麻烦。

查看完成安装后的PHP版本

```
php -v
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ php -v
PHP 8.3.14 (cli) (built: Nov 25 2024 18:07:43) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.14, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.14, Copyright (c), by Zend Technologies
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

让PHP 8.3与Nginx一起使用

1.安装Nginx：下载Nginx之前的教程我已经有讲述，这里不着重说了。

2.安装PHP 8.3-FPM（FastCGI Process Manager）：

```
sudo apt install php8.3-fpm -y
```

3.启动PHP-FPM服务并设置为开机自启：

```
sudo systemctl start php8.3-fpm
sudo systemctl enable php8.3-fpm
```

4.配置Nginx以使用PHP-FPM。编辑Nginx站点 **配置文件**（通常在 `etc/nginx/sites-available/` 的目录下）：

```
sudo vim /etc/nginx/sites-available/caicloudcat.publicvm.com
```

在配置文件中，添加以下内容：

```

server {
    listen 80;
    server_name caicloudcat.publicvm.com www.caicloudcat.publicvm.com;

    access_log /var/log/nginx/caicloudcat-publicvm.log;
    error_log /var/log/nginx/caicloudcat-publicvm.log;

    return 301 https:$server_name$request_uri;
}

server {
    listen 443 ssl;
    server_name caicloudcat.publicvm.com www.caicloudcat.publicvm.com;
    root /var/www/caicloudcat.publicvm.com;

    ssl_certificate /certs/caicloudcat.publicvm.com/zerossll_combined.crt;
    ssl_certificate_key /certs/caicloudcat.publicvm.com/private.key;

    index index.html index.php;

    location ~ /\.php$ {
        root /var/www/caicloudcat.publicvm.com;
        fastcgi_pass unix:/var/run/php/php8.3-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}

```

最后检查以及重启服务，我这里不赘述了。

接下来测试验证配置：

```
echo "<?php phpinfo(); ?>" | sudo tee /var/www/caicloudcat.publicvm.com/index.php
```

在Linux中访问：`https://www.caicloudcat.publicvm.com/index.php` 可以看到PHP信息页面，说明配置成功。

安装开源Web应用-WordPress

在LAMP环境中安装开源Web应用，如开源博客系统WordPress，开源学习平台Moodle等。

这里我以WordPress为例。

创建用户管理以及数据库

首先创建数据库，创建一个新的数据库以及对应的用户用于WordPress，这里我的MySQL的版本为8.4.3，为mysql8版本。

进入mysql控制台：

```
sudo mysql -u root -p
```

创建一个新的数据库：

```
create database wordpress_db;
```

创建一个新的用户并分配权限：

```
grant all on wordpress_db.* to 'wordpress_user'@'localhost' identified by 'root';
```

发现会报错：

```
mysql> GRANT ALL ON wordpress_db.* TO 'wordpress_user'@'localhost' IDENTIFIED BY 'root';
1064 - You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'IDENTIFIED BY 'root'' at line 1
```

这是因为在MySQL 8中，对于创建用户和授权的操作进行了一定的改变。

在MySQL8之前的版本中，可以使用GRANT语句一次性创建用户并授权其权限。但是在MySQL8中，创建用户和授权权限是分开操作。

要解决这个问题，可以按照下面的步骤进行：

1.首先，使用CREATE USER语句创建用户。例如，创建一个名为"wordpress_user"的用户，其密码为 root：

```
CREATE USER 'wordpress_user'@'localhost' IDENTIFIED BY 'root';  
# 因为mysql8加密方式和Navicat不同, 修改加密方式  
alter user wordpress_user identified with mysql_native_password by 'root';  
flush privileges;
```

2.然后, 设置用户的对应的 host :

```
use mysql;  
update user set host='%' where user='wordpress_user';  
flush privileges;
```

3.然后, 使用GRANT语句为用户授予所需的权限。例如, 授予用户"wordpress_user"在数据库"wordpress_db"上的所有权限:

```
GRANT all ON wordpress_db.To 'wordpress_user'@'%';  
flush privileges;  
exit;
```

4.最后用 `wordpress_user` 用户进入mysql控制台:

```
mysql -u wordpress_user -p
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ mysql -u wordpress_user -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 26  
Server version: 8.4.3 MySQL Community Server - GPL  
  
Copyright (c) 2000, 2024, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> █
```

参考资料: [解决mysql8报错: ERROR 1410 \(42000\): You are not allowed to create a user with GRANT](#)、[解决MySQL8报错: ERROR 1410 \(42000\): You are not allowed to create a user with GRANT](#)

配置WordPress

下载WordPress：在网站的根目录下（通常在 `/var/www/html/` ）下载并解压WordPress，如果配置了虚拟主机，也可以配置在虚拟主机的根目录下，这里我配置在我的虚拟主机下。

```
cd /var/www/caicloudcat.publicvm.com/  
sudo wget -c http://wordpress.org/latest.tar.gz  
sudo tar -xzf latest.tar.gz  
sudo rm latest.tar.gz
```

配置WordPress：将默认配置文件复制为新的配置文件，然后设置WordPress配置。

```
cd /var/www/caicloudcat.publicvm.com/wordpress  
sudo cp wp-config-sample.php wp-config.php  
sudo vim wp-config.php
```

在编辑器中修改以下行，将数据库名称、用户名和密码替换为之前创建的数据库和用户信息。

```
define('DB_NAME', 'wordpress_db');  
define('DB_USER', 'wordpress_user');  
define('DB_PASSWORD', 'host');  
define('DB_HOST', 'localhost')
```

设置文件权限：运行以下命令设置WordPress文件和目录的权限。

```
cd /var/www/caicloudcat.publicvm.com/  
sudo chown -R www-data:www-data wordpress  
sudo chmod -R 755 wordpress
```

配置相关的Nginx的配置文件：

首先找到相关的配置文件目录（通常在 `/etc/nginx/sites-available/` ）中：

```
cd /etc/nginx/sites-available/
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ cd /etc/nginx/sites-available/  
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/nginx/sites-available$ ls -l  
total 8  
-rw-r--r-- 1 root root 871 Dec 14 00:31 caicloudcat.publicvm.com  
-rw-r--r-- 1 root root 3164 Dec 12 23:24 default  
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/nginx/sites-available$
```

编辑 caicloudcat.publicvm.com :

```
sudo vim caicloudcat.publicvm.com
```

修改相关配置:


```

server {
    listen 80;
    server_name caicloudcat.publicvm.com www.caicloudcat.publicvm.com;
    # root /var/www/caicloudcat.publicvm.com;

    access_log /var/log/nginx/caicloudcat-publicvm.log;
    error_log /var/log/nginx/caicloudcat-publicvm.log;
    # 当有客户端通过 HTTP（80 端口）访问对应的域名时，
    # 给客户端返回一个 301 状态码（表示永久重定向），并将请求的 URL 重定向到对应的 HTTPS 版本
    return 301 https:$server_name$request_uri;
}

server {
    listen 443 ssl;
    server_name caicloudcat.publicvm.com www.caicloudcat.publicvm.com;
    root /var/www/caicloudcat.publicvm.com/wordpress;

    ssl_certificate /certs/caicloudcat.publicvm.com/zeross1_combined.crt;
    ssl_certificate_key /certs/caicloudcat.publicvm.com/private.key;

    # `index` 指令用于指定当客户端请求的是一个目录时，服务器默认查找并返回的文件。
    # 可以设置多个文件，按照顺序查找，找到第一个存在的文件就返回给客户端。
    # index index.html index.php;
    index index.php;

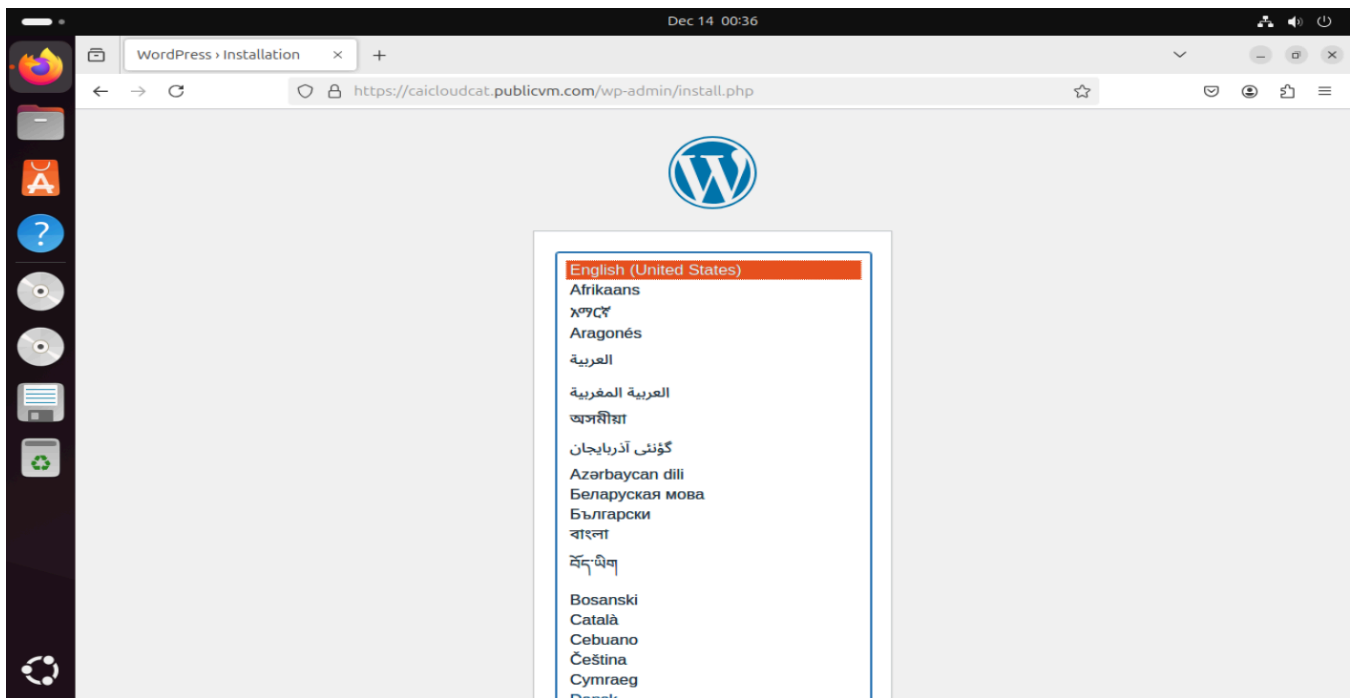
    location ~ \.php$ {
        root /var/www/caicloudcat.publicvm.com/wordpress;
        fastcgi_pass unix:/var/run/php/php8.3-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}

```

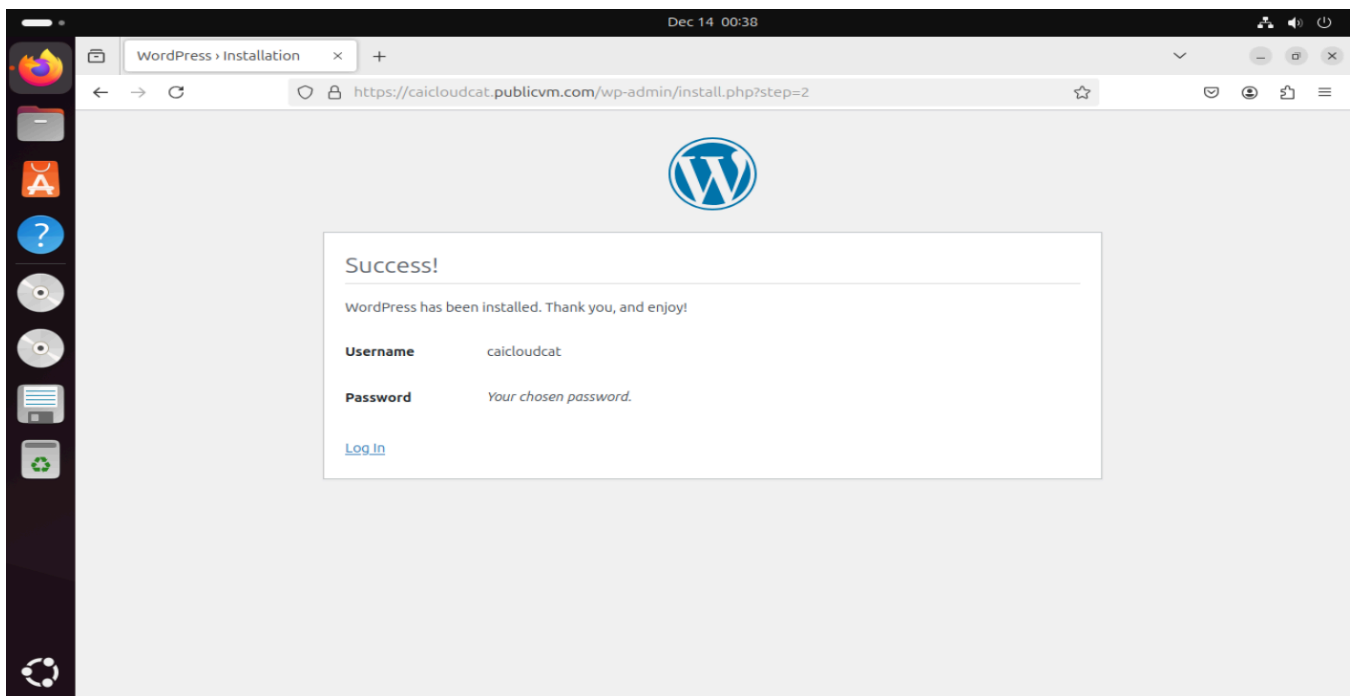
最后就是检查是否有错误 `sudo nginx -t` 以及重启服务 `sudo systemctl restart nginx`。

接下来就是进行WordPress的安装了：

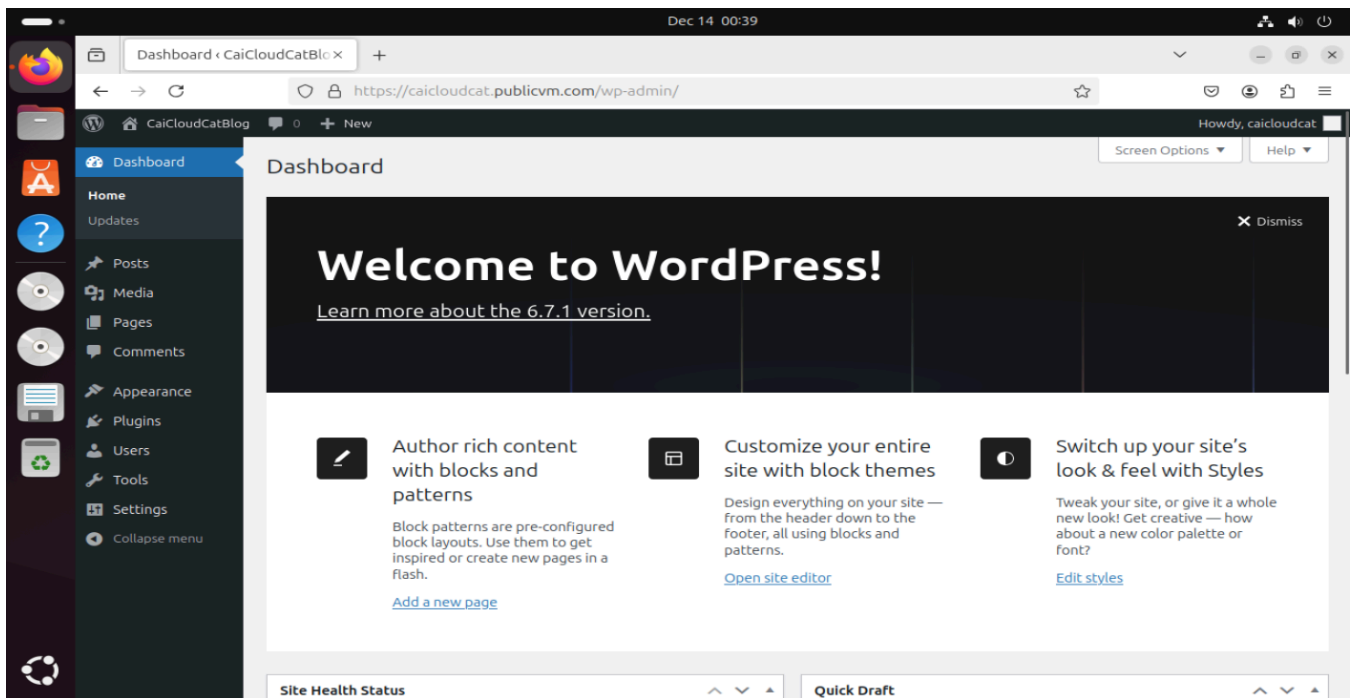
选择语言：



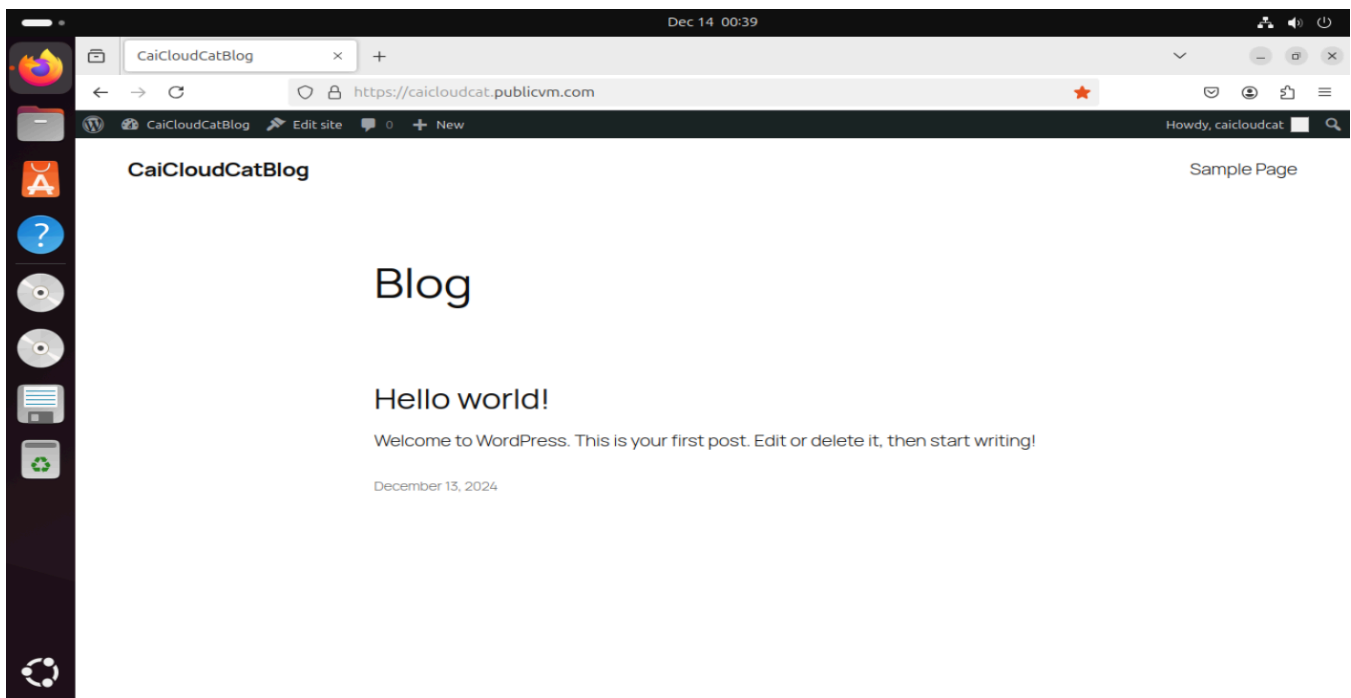
进行基础信息的填写，最后成功：



接下来就到了管理者的后台界面：



以及主页面：



参考资料

在Ubuntu24.04安装PHP8.3加Nginx，创建WordPress开发环境

ubuntu下如何安装配置php nginx Mysql环境

[安装部署wordpress\(Ubuntu\)](#)

[ubuntu系统服务器安装WordPress教程](#)

[安装 WordPress – 如何在 Ubuntu 上安装 WordPress](#)

[在ubuntu系统上安装nginx以及php的部署](#)

[在 Ubuntu 22.04 中配置 LNMP 并安装 WordPress \(1\)](#)

[Ubuntu上使用Nginx和PHP搭建Web服务器](#)

[Ubuntu24手动部署LNMP环境](#)