

创建普通用户家目录



ubuntu如何创建普通用户和家目录



在Ubuntu中，创建一个普通用户并为其创建家目录，可以使用以下步骤：

1. 使用 `useradd` 命令创建用户

`useradd` 命令用于创建新用户，并可以指定创建家目录。默认情况下，`useradd` 会在 `/home/` 下为用户创建一个同名的家目录。

创建用户并自动创建家目录：

```
bash

sudo useradd -m <用户名>
```

- `-m` 选项表示同时创建该用户的家目录（即 `/home/<用户名>`）。

例如，创建一个名为 `newuser` 的用户：

```
bash

sudo useradd -m newuser
```

2. 设置用户密码

创建完用户后，需要为其设置密码。使用 `passwd` 命令来设置密码：

```
bash

sudo passwd newuser
```

系统会提示你输入并确认密码。

3. 验证用户及家目录

创建用户和家目录后，可以使用 `ls` 命令查看 `/home/` 目录来确认家目录是否成功创建：

```
bash

ls /home/
```

你应该能看到新用户的家目录 `/home/newuser`。

4. (可选) 添加用户到用户组

默认情况下，用户会被添加到与用户名相同的组中。如果你想将用户添加到其他组，可以使用 `usermod` 命令：

```
bash

sudo usermod -aG <组名> newuser
```

例如，将用户 `newuser` 添加到 `sudo` 组中，使其拥有管理权限：

```
bash

sudo usermod -aG sudo newuser
```

总结

完整的操作步骤如下：

1. 创建用户并自动创建家目录：

```
bash

sudo useradd -m newuser
```

2. 设置密码：

```
bash

sudo passwd newuser
```

3. (可选) 将用户添加到某个组：

```
bash

sudo usermod -aG sudo newuser
```

通过以上步骤，你就创建了一个普通用户并为其设置了家目录。



如何给普通用户添加权限



在Ubuntu中，可以通过将普通用户添加到特定的用户组或者通过修改文件和目录的权限来赋予普通用户权限。以下是一些常见的方法，具体取决于你希望授予用户什么样的权限。

1. 将普通用户添加到特定组

Ubuntu中有很多组定义了不同的权限，通过将用户添加到合适的组，你可以授予他们相应的权限。

常见的用户组：

- **sudo**: 授予用户管理员权限（可以使用 `sudo` 执行管理命令）。
- **www-data**: 授予用户Web服务器的访问权限（常用于管理Web应用）。
- **docker**: 授予用户使用Docker命令的权限。
- **audio**: 授予用户访问音频设备的权限。

添加用户到组：

使用 `usermod` 命令将用户添加到一个组中：

```
bash

sudo usermod -aG <组名> <用户名>
```

例如，想要将用户 `newuser` 添加到 `sudo` 组，使其能够使用 `sudo` 执行管理员操作：

```
bash

sudo usermod -aG sudo newuser
```

注意: `-aG` 参数意味着将用户追加到指定的组，而不会移除他原本属于的组。

检查用户所属的组：

你可以使用 `groups` 命令查看某个用户所属的所有组：

```
bash

groups newuser
```

2. 设置文件和目录权限

通过更改文件和目录的权限，允许用户对特定文件进行读、写或执行操作。你可以使用 `chmod` 和 `chown` 命令来设置这些权限。

`chmod` 命令：修改文件权限

`chmod` 命令用于设置文件和目录的权限。权限分为三种：读（r）、写（w）和执行（x）。

- 读权限（r）：用户可以查看文件内容。
- 写权限（w）：用户可以修改文件内容。
- 执行权限（x）：用户可以执行文件（如果是脚本或程序）。

权限用数字表示，例如：

- `7` = rwx（读、写、执行）
- `6` = rw-（读、写）
- `5` = r-x（读、执行）
- `4` = r--（只读）

使用 `chmod` 修改权限：

```
bash

sudo chmod <权限数字> <文件或目录路径>
```

例如，给文件 `file.txt` 添加读写权限：

```
bash

sudo chmod 660 file.txt
```

这个命令授予文件拥有者和文件所属组的用户读写权限，而其他用户没有权限。

`chown` 命令：修改文件或目录的拥有者

`chown` 命令用于更改文件或目录的所有者和所属组：

```
bash

sudo chown <用户名>:<组名> <文件或目录路径>
```

例如，将文件 `file.txt` 的所有者设置为 `newuser`，所属组设置为 `newuser`：

```
bash

sudo chown newuser:newuser file.txt
```

3. 使用 `sudo` 权限控制（`sudoers` 文件）

如果你想给普通用户更细粒度的权限（例如，可以运行某些特定命令，但不能做其他事情），你可以编辑 `sudoers` 文件，来授予用户执行特定命令的权限。

编辑 `sudoers` 文件：

使用 `visudo` 命令来安全编辑 `sudoers` 文件：

```
bash

sudo visudo
```

在文件中添加一行，允许用户执行特定命令。例如，允许 `newuser` 以管理员身份执行 `/bin/ls` 和 `/bin/cat`：

```
bash

newuser ALL=(ALL) NOPASSWD: /bin/ls, /bin/cat
```

这将允许 `newuser` 无需输入密码即可执行这些命令。

4. 给予用户对某些目录的访问权限

如果你想让普通用户访问或管理特定目录（例如，`/var/www` 目录下的文件），可以更改目录的访问权限或所有权。

更改目录的所有者：

```
bash

sudo chown -R newuser:newuser /var/www/
```

这样 `newuser` 就可以管理该目录下的文件了。

给予用户对目录的写权限：

```
bash

sudo chmod -R u+w /var/www/
```

总结

1. **将用户添加到组：** 使用 `usermod -aG <组名> <用户名>`，例如将用户 `newuser` 添加到 `sudo` 组中。
2. **修改文件或目录权限：** 使用 `chmod` 设置文件或目录的权限，使用 `chown` 修改文件或目录的所有者。

3. **细粒度权限控制（`sudoers` 文件）**：使用 `visudo` 编辑 `sudoers` 文件，赋予用户对特定命令的权限。

4. **特定目录权限**：使用 `chown` 和 `chmod` 命令修改目录权限，使用户可以访问或管理特定的目录。

通过这些方法，你可以灵活地管理普通用户的权限。



在/etc/sudoers如何设置普通用户可以用sudo su - 输入root密码进行切换root用户



要在 `/etc/sudoers` 文件中设置普通用户可以使用 `sudo su -` 命令切换到 `root` 用户并输入 `root` 密码（而不是普通用户的密码），你需要做一些调整。

不过，请注意，这种做法通常不建议使用，因为它涉及绕过正常的 `sudo` 密码认证方式，但可以根据需求来配置。

步骤：

1. 编辑 `/etc/sudoers` 文件

使用 `visudo` 命令来安全地编辑 `/etc/sudoers` 文件。直接编辑此文件容易出错，`visudo` 会在保存时进行语法检查，防止错误。

```
bash

sudo visudo
```

2. 添加一条规则，允许普通用户使用 `sudo su -` 切换到 `root`

在 `sudoers` 文件中找到 `User privilege specification` 部分，通常这个部分会在文件的底部。

你可以添加以下规则来允许特定用户（例如 `newuser`）使用 `sudo su -` 命令切换到 `root`，并且要求输入 `root` 密码而不是 `newuser` 的密码。

```
bash

newuser ALL=(ALL) NOPASSWD: /bin/su
```

解释：

- `newuser`：你希望授权的普通用户名。
- `ALL`：允许在任何主机上执行命令。
- `(ALL)`：允许以任何用户身份执行命令（包括 `root`）。
- `NOPASSWD`：表示不需要输入 `newuser` 的密码。
- `/bin/su`：表示可以运行 `su` 命令，通常是用来切换到 `root` 用户。

这条规则的意思是，用户 `newuser` 可以在不输入自己密码的情况下运行 `/bin/su` 命令。接下来，在切换到 `root` 用户时，会要求输入 `root` 的密码。

3. 保存并退出

完成编辑后，保存文件并退出。对于 `visudo`，按下 `Ctrl + X`，然后按 `Y` 确认保存，最后按 `Enter` 键。

4. 测试

现在你可以通过执行 `sudo su -` 命令切换到 `root` 用户，系统会要求你输入 `root` 用户的密码，而不是普通用户的密码。

```
bash

sudo su -
```

你应该会被要求输入 `root` 密码。

其他注意事项

- **安全性:** 允许用户通过 `sudo su -` 切换到 `root` 用户并使用 `root` 密码，可能会带来安全风险，因为这会使普通用户更容易以 `root` 身份执行命令。在大多数情况下，使用 `sudo` 直接执行特定命令是更安全的做法。
- **完全权限:** 如果你希望普通用户完全获得 `root` 权限，并且输入 `root` 密码，你可以将他们添加到 `sudo` 组，或者允许他们执行所有 `sudo` 命令（而不要求输入密码）：

```
bash

newuser ALL=(ALL) NOPASSWD: ALL
```

这会允许 `newuser` 运行任何命令（包括 `su`）而不需要输入密码。请谨慎使用此方法。

通过以上步骤，普通用户将能够通过 `sudo su -` 切换到 `root` 用户，并输入 `root` 密码进行身份验证。



ubuntu怎么更改普通用户密码



在Ubuntu中，修改普通用户的密码，可以使用 `passwd` 命令。以下是更改普通用户密码的详细步骤：

1. 更改自己密码（当前用户）

如果你想要更改当前登录用户的密码，直接使用 `passwd` 命令即可。

```
bash

passwd
```

系统会提示你输入当前密码，然后再输入新密码并确认新密码。

2. 更改其他用户的密码

如果你是管理员（具有 `sudo` 权限），可以更改其他普通用户的密码。使用 `sudo passwd` 命令，并指定用户名。

```
bash

sudo passwd <用户名>
```

例如，如果你想要更改名为 `newuser` 的用户密码，执行以下命令：

```
bash

sudo passwd newuser
```

系统会要求你输入两次新密码。输入并确认新密码后，用户 `newuser` 的密码就会被更新。

3. 注意事项

- **密码强度**：Ubuntu 默认会对密码强度做一些要求，例如密码长度、字符种类等。如果你输入的密码不符合系统的要求，系统会提示你重新输入一个符合要求的密码。
- **`sudo` 权限**：如果你没有管理员权限，无法更改其他用户的密码。你必须先获取管理员权限或联系系统管理员。

总结

- 更改当前用户的密码：`passwd`
- 更改其他用户的密码（需要管理员权限）：`sudo passwd <用户名>`

通过这两个命令，你可以方便地更改用户的密码。



这是什么问题？E45: 'readonly' option is set (add ! to override)



`E45: 'readonly' option is set (add ! to override)` 是在使用 **Vim** 编辑器时，遇到的一种错误提示。

这个错误的含义：

Vim 给出的 `E45` 错误信息意味着，你正在尝试编辑一个 **只读** 文件（或者在一个只读的缓冲区中），而 Vim 阻止你进行保存或修改操作。

- **只读模式**：文件或者缓冲区的权限设置为只读，防止你意外修改文件内容。
- 你正在尝试保存修改，或者退出时保存文件，但由于文件处于只读模式，Vim 阻止了这个操作。

解决方法：

你可以使用以下几种方法来解决这个问题，具体方法取决于你想要执行的操作。

1. 强制保存文件

如果你确定要保存文件并覆盖它，可以使用 `:w!` 命令来强制写入文件，忽略只读设置：

```
vim

:w!
```

这会强制保存文件并覆盖文件，即使它是只读的。

2. 退出时保存文件

如果你想保存文件并退出 Vim，可以使用 `:wq!` 命令（表示强制保存并退出）：

```
vim

:wq!
```

3. 以超级用户权限编辑文件

如果文件是只读的并且你没有足够的权限进行修改，可能需要使用 **超级用户权限**。你可以在 Vim 中通过以下命令来以管理员权限重新打开文件：

```
vim

:sudoedit <文件名>
```

这会要求你输入管理员（`sudo`）密码，并以超级用户权限打开文件，允许你保存修改。

4. 修改文件权限（如果有权限）

如果你有文件的访问权限，可以修改文件的读写权限，以便直接修改它。使用 `chmod` 命令更改文件权限：

```
bash

sudo chmod +w <文件名>
```

这会给文件添加写权限。然后你可以在 Vim 中直接保存文件。

总结

- `:w!` 强制保存文件，忽略只读设置。
- `:wq!` 强制保存并退出。
- `:sudoedit <文件名>` 使用超级用户权限打开文件并保存修改。
- `chmod +w` 更改文件的写权限，使其可编辑。

通过这些方法，你可以解决 `E45` 错误并成功保存你的修改。



ubuntu如何禁止root用户远程登录呢？只允许普通用户sudo su -



在 Ubuntu 中，禁止 `root` 用户远程登录（通过 SSH 或其他远程服务），并且只允许普通用户通过 `sudo su -` 方式切换到 `root` 用户，你可以按以下步骤操作：

1. 禁止 `root` 用户通过 SSH 登录

Ubuntu 使用 `OpenSSH` 作为远程登录工具，你可以通过配置 SSH 服务来禁止 `root` 用户直接远程登录。

步骤：

1. 打开 SSH 配置文件 `/etc/ssh/sshd_config`：

```
bash

sudo nano /etc/ssh/sshd_config
```

2. 找到并修改以下行：

```
bash

PermitRootLogin yes
```

修改为：

```
bash

PermitRootLogin no
```

这将禁止 `root` 用户通过 SSH 进行远程登录。

3. 保存文件并退出（在 `nano` 中按 `Ctrl + O` 保存，按 `Ctrl + X` 退出）。
4. 重新启动 SSH 服务以使更改生效：

```
bash

sudo systemctl restart ssh
```

或者使用以下命令：

```
bash

sudo service ssh restart
```

这样，`root` 用户将无法通过 SSH 进行远程登录。

2. 允许普通用户使用 `sudo su -` 切换到 `root`

普通用户已经可以通过 `sudo su -` 切换到 `root` 用户，前提是该用户已被授予 `sudo` 权限。

如果你尚未为某个用户配置 `sudo` 权限，可以按以下步骤操作：

1. 将普通用户添加到 `sudo` 组

使用 `usermod` 命令将用户添加到 `sudo` 组（`sudo` 组的用户默认具有执行 `sudo` 命令的权限）：

```
bash

sudo usermod -aG sudo <用户名>
```

例如，将 `newuser` 添加到 `sudo` 组：

```
bash

sudo usermod -aG sudo newuser
```

2. 验证普通用户是否具有 `sudo` 权限

切换到该用户并运行 `sudo` 命令来测试是否具有管理员权限：

```
bash

su - newuser
sudo ls /root
```

如果成功执行，则表示该用户具有 `sudo` 权限。

3. 确保可以使用 `sudo su -` 切换到 `root`

普通用户可以使用 `sudo su -` 切换到 `root`：

```
bash

sudo su -
```

系统会要求用户输入自己（而不是 `root`）的密码。之后用户会进入 `root` 用户的 shell。

3. 确保普通用户无法直接以 `root` 身份登录

你已经禁用了 `root` 用户的远程登录，并且将 `sudo` 权限授予了普通用户。通过 `sudo su -` 方式可以切换到 `root`，但 `root` 账户本身仍然不能直接登录。

为了进一步确保安全，可以检查一下以下设置：

- **禁止 `root` 用户本地登录（例如通过控制台）：**

在 `/etc/securetty` 文件中，确保没有列出允许 `root` 登录的终端。你可以通过以下命令查看该文件：

```
bash

cat /etc/securetty
```

如果文件中列出了任何终端（如 `tty1`、`tty2` 等），可以注释掉或删除这些行来进一步限制 `root` 的本地登录。

总结

- **禁止 `root` 用户远程登录：**通过修改 SSH 配置文件 `/etc/ssh/sshd_config`，将 `PermitRootLogin` 设置为 `no`。
- **允许普通用户使用 `sudo su -` 切换到 `root`：**确保普通用户在 `sudo` 组中，并通过 `sudo su -` 进行切换。
- **确保 `root` 无法直接登录：**禁用 `root` 账户的 SSH 登录权限，并检查 `/etc/securetty` 以限制本地登录。

通过这些设置，你可以确保 `root` 用户无法远程登录，而普通用户可以使用 `sudo` 切换到 `root` 用户，既保证了系统的安全性，又保留了必要的管理权限。



我的配置文件有啥问题？为什么按照你说的配置，root用户还是能远程登录呢？

```
Include /etc/ssh/sshd_config.d/*.conf
```

```
Port 1737
```

```
PermitRootLogin no
StrictModes yes
```

```
KbdInteractiveAuthentication no

UsePAM yes

AcceptEnv LANG LC_*

Subsystem sftp /usr/lib/openssh/sftp-server
```



根据你提供的 SSH 配置文件片段，禁用 `root` 用户远程登录的配置应该是有效的，尤其是 `PermitRootLogin no` 这一行。出现问题的原因可能有几个，你可以检查以下几个方面：

1. `sshd_config` 配置文件的加载顺序

你提到配置文件中有一行：

```
bash

Include /etc/ssh/sshd_config.d/*.conf
```

这表示主配置文件 `/etc/ssh/sshd_config` 可能会包含其他的配置文件，例如 `/etc/ssh/sshd_config.d/` 目录下的所有 `.conf` 文件。如果这些文件中有涉及 `PermitRootLogin` 的配置，并且该配置允许 `root` 登录（例如 `PermitRootLogin yes`），它将覆盖主配置文件中的设置。

解决方案：

检查 `/etc/ssh/sshd_config.d/` 目录中的所有 `.conf` 文件，确认其中没有允许 `root` 登录的设置。你可以使用以下命令列出所有 `.conf` 文件并查看它们的内容：

```
bash

cat /etc/ssh/sshd_config.d/*.conf
```

确保没有重复的 `PermitRootLogin` 设置。

2. 检查 SSH 服务的重启是否生效

配置文件修改后，需要确保重新启动了 SSH 服务。如果没有正确重启服务，配置更改可能没有生效。

解决方案：

确保你已经使用以下命令重新启动了 SSH 服务：

```
bash

sudo systemctl restart ssh
```

或者：

```
bash

sudo service ssh restart
```

3. 验证配置是否正确加载

你可以通过 `sshd -T` 命令来检查当前 SSH 服务的所有有效配置，这样可以验证是否加载了正确的配置，并确保 `PermitRootLogin` 设置正确。

```
bash

sudo sshd -T | grep permitrootlogin
```

此命令的输出应该是：

```
bash

permitrootlogin no
```

如果返回 `no`，说明配置已经生效。如果返回 `yes` 或者其他值，说明配置没有正确加载或者被覆盖。

4. 检查 SSH 连接时的实际行为

有时候，即使配置正确，可能是你连接的方式导致 `root` 仍然能登录。确保你在连接时没有以 `root` 用户身份尝试登录（即直接使用 `root@hostname`）。

可以通过以下命令尝试远程连接：

```
bash

ssh newuser@hostname
```

然后在连接后使用 `sudo su -` 切换到 `root` 用户。

5. `AllowUsers` 或 `AllowGroups` 配置

如果你在配置文件中使用了 `AllowUsers` 或 `AllowGroups` 指令来限制哪些用户可以登录 SSH，确保没有误将 `root` 用户列在其中。例如：

```
bash

AllowUsers newuser
```

这样就会只允许 `newuser` 登录，而禁用 `root` 用户。

6. 其他配置冲突（如 PAM）

在某些情况下，PAM（Pluggable Authentication Modules）配置可能会影响 SSH 登录行为。配置中的 `UsePAM yes` 允许 PAM 进行身份验证，某些 PAM 模块可能影响 `root` 登录行为。通常情况下，`UsePAM yes` 是启用的，但可以检查 `/etc/pam.d/` 中是否有任何相关的文件（比如 `/etc/pam.d/sshd`）干扰 SSH 的登录限制。

总结步骤：

1. 检查 `/etc/ssh/sshd_config.d/` 目录中的所有 `.conf` 文件，确认没有任何配置覆盖 `PermitRootLogin no`。
2. 确保重新启动了 SSH 服务：`sudo systemctl restart ssh`。
3. 使用 `sshd -T | grep permitrootlogin` 检查当前 SSH 配置是否生效。
4. 确保你没有在 SSH 连接时直接使用 `root` 登录。
5. 检查是否有 `AllowUsers` 或 `AllowGroups` 设置允许了 `root` 用户登录。
6. 检查 `/etc/pam.d/sshd` 配置文件是否存在干扰。

按照这些步骤排查问题，应该能找出为什么 `root` 用户仍然能够远程登录的问题。



为什么我的普通用户 `sudo su -` 不用输入密码 `sudo -i` 需要输入 root 密码呢，能否让 `sudo -i` 不需要输入密码



问题分析：

在 Ubuntu 中，`sudo su -` 和 `sudo -i` 都是用来切换到 `root` 用户的命令，但它们的行为略有不同，特别是在密码提示方面。

1. `sudo su -`：
 - `sudo su -` 通常会要求你输入当前普通用户的密码，而不是 `root` 用户的密码。因为你是通过 `sudo` 提升到 `root` 权限，然后切换到 `root` 用户的 shell（`su -`）。
2. `sudo -i`：
 - `sudo -i` 启动一个 **登录 shell**，并且会在新的 shell 环境中执行 `root` 用户的命令。它默认要求你输入 **root 用户的密码**，而不是普通用户的密码。这是因为它启动的是一个与 `root` 用户相关的环境。

解决方案：让 `sudo -i` 也不需要输入密码

要让 `sudo -i` 不需要输入密码，可以通过修改 `sudo` 的配置文件 `/etc/sudoers` 来实现。你可以为特定的用户配置，使得他们在运行 `sudo -i` 或其他 `sudo` 命令时，不需要输入密码。

1. 修改 `/etc/sudoers` 文件

你可以通过 `visudo` 命令安全地编辑 `/etc/sudoers` 文件。`visudo` 会检查文件的语法错误，防止由于语法错误导致的系统问题。

```
bash

sudo visudo
```

2. 添加配置来禁用密码

在打开的文件中，你可以为特定的用户或用户组配置 `NOPASSWD` 选项，从而让 `sudo` 命令不再要求输入密码。假设你想要允许用户 `newuser` 在使用 `sudo -i` 时不输入密码。你可以在文件中添加如下配置：

```
bash

newuser ALL=(ALL) NOPASSWD: /usr/bin/sudo
```

这表示用户 `newuser` 可以在任何主机上运行 `sudo` 命令而无需输入密码。

如果你希望所有在 `sudo` 组的用户都能无密码使用 `sudo`，可以添加：

```
bash

%sudo ALL=(ALL) NOPASSWD: ALL
```

这会允许 `sudo` 组的所有成员在运行任何命令时都不需要输入密码。

3. 保存并退出

编辑完成后，按下 `Ctrl + X` 保存并退出（如果你使用的是 `nano` 编辑器）。如果你在其他编辑器中，可以根据相应编辑器的操作保存退出。

4. 验证配置是否生效

完成配置后，可以通过以下命令来验证：

```
bash

sudo -i
```


这时，如果配置正确，`sudo -i` 应该不会再要求你输入密码。

总结：

- `sudo su -` 会要求输入当前普通用户的密码。
- `sudo -i` 默认会要求输入 `root` 用户的密码。
- 要让 `sudo -i` 不要求输入密码，你可以在 `/etc/sudoers` 文件中添加 `NOPASSWD` 配置，允许特定用户或者组在执行 `sudo` 命令时不输入密码。

通过这些步骤，你可以让 `sudo -i` 命令与 `sudo su -` 一样，不再要求输入密码。