

github使用ssh直接用git自动登陆github账号

SSH密钥与身份验证

为了简化登录过程，GitHub支持使用SSH密钥进行身份验证。通过配置SSH密钥，你可以在不输入用户名和密码的情况下与GitHub进行交互。

配置SSH密钥的步骤：

1. 生成SSH密钥对：在终端中运行以下命令生成SSH密钥对：

bash复制

```
ssh-keygen -t rsa -b 4096 -C "你的邮箱"
```

按提示操作，可以选择默认路径保存密钥文件。

2. 将SSH公钥添加到GitHub：

- 打开生成的公钥文件（通常在`~/.ssh/id_rsa.pub`）。
- 复制公钥内容。
- 登录GitHub，进入“Settings” > “SSH and GPG keys”。
- 点击“New SSH key”，粘贴公钥内容并保存。

3. 使用SSH链接克隆仓库：使用SSH链接克隆仓库时，Git会自动使用配置的SSH密钥进行身份验证，无需手动输入用户名和密码。

4. 命令解析：

bash复制

```
ssh-keygen -t rsa -b 4096 -C "moshuai1013@outlook.com"
```

- `-t rsa`：指定密钥类型为RSA。
- `-b 4096`：指定密钥长度为4096位（更安全）。
- `-C "moshuai1013@outlook.com"`：添加一个注释（通常是你的邮箱），用于标识密钥。

5. 保存路径：

- 默认情况下，SSH密钥会保存在用户主目录的`.ssh`文件夹中（例如`C:\Users\莫帅\.ssh\id_rsa`）。
- 然后要做一下步骤

1. 输入Passphrase（密码短语）

Passphrase 是一个额外的安全层，用于保护你的SSH私钥。如果你的私钥被他人获取，没有正确的passphrase，他们也无法使用它。

- 输入**Passphrase**：输入一个安全的密码短语（可以包含字母、数字和特殊字符）。这个密码只有你自己知道。
- 确认**Passphrase**：系统会要求你再次输入相同的密码短语以确认。

如果你不想设置passphrase，可以直接按回车键跳过（不推荐，因为passphrase可以显著提高安全性）。

2. 完成密钥生成

完成以上步骤后，SSH密钥对将被生成：

- 私钥：保存在 D:\bilibili\ssh，文件名为 ssh。
 - 公钥：保存在 D:\bilibili\ssh.pub，文件名为 ssh.pub。
-

3. 查看生成的密钥文件

生成完成后，你可以检查生成的密钥文件：

- 打开文件资源管理器，导航到 D:\bilibili 文件夹。
 - 你会看到两个文件：
 - ssh：这是你的私钥文件，不要泄露。
 - ssh.pub：这是你的公钥文件，用于添加到GitHub。
-

4. 将公钥添加到GitHub

为了使用SSH密钥与GitHub交互，你需要将公钥添加到你的GitHub账户中：

1. 打开 D:\bilibili\ssh.pub 文件（可以用记事本打开）。
 2. 复制文件中的所有内容（从 ssh-rsa 开始，到结尾的邮箱地址）。
 3. 登录GitHub，进入 **Settings > SSH and GPG keys**。
 4. 点击 **New SSH key** 按钮。
 5. 在 **Title** 中输入一个描述（例如 Windows SSH Key）。
 6. 在 **Key** 中粘贴你的公钥内容。
 7. 点击 **Add SSH key** 按钮完成。
-

5. 测试SSH连接

为了确保SSH密钥配置成功，你可以测试与GitHub的连接：

1. 打开命令行工具（例如Git Bash）。
2. 输入以下命令：

bash复制

```
ssh -T git@github.com
```

3. 如果提示 The authenticity of host 'github.com' can't be established, 输入 yes。
4. 如果配置成功，你会看到类似以下的提示：

```
Hi username! You've successfully authenticated, but GitHub
does not provide shell access.
```

这表示你的SSH密钥已经成功配置。

```
C:\Users\莫帅>ssh-keygen -t rsa -b 4096 -C "moshuai1013@outlook.com"
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\莫帅\.ssh/id_rsa): D:\bilibili
D:\bilibili already exists.
Overwrite (y/n)? n

C:\Users\莫帅>ssh-keygen -t rsa -b 4096 -C "moshuai1013@outlook.com"
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\莫帅\.ssh/id_rsa): D:\bilibili\ssh
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in D:\bilibili\ssh
Your public key has been saved in D:\bilibili\ssh.pub
The key fingerprint is:
SHA256:9trq4s/xaIoraFdSfW2CdBUKgHRpp6rNLz6ddSP8JY moshuai1013@outlook.com
The key's randomart image is:
+---[RSA 4096]-----+
|      o+. .+..      |
|      +... o        |
|      =.o..         |
|      o ooo o        |
|      o .oS.o        |
|      .. +..o.+      |
|      . +o...E..     |
|      ..o.+oo.o*     |
|      ..+=+oo*B.o    |
+---[SHA256]-----+

C:\Users\莫帅>
```

扩展知识：

在生成SSH密钥对时，ssh-keygen工具会提供一个选项来显示密钥的“随机艺术图像”（RandomArt image）。这是一个可视化的表示，用于帮助用户识别和验证SSH密钥。

随机艺术图像的作用

随机艺术图像是一种将密钥的指纹（fingerprint）转换为图形的方式。它的主要目的是：

1. 增强可读性：密钥指纹通常是一串复杂的十六进制字符，很难记忆和验证。随机艺术图像通过图形化的方式，让用户更容易识别和比较密钥。
2. 辅助验证：在某些场景下，用户可以通过比较随机艺术图像来验证密钥是否正确，尤其是在手动检查密钥时。

随机艺术图像的生成

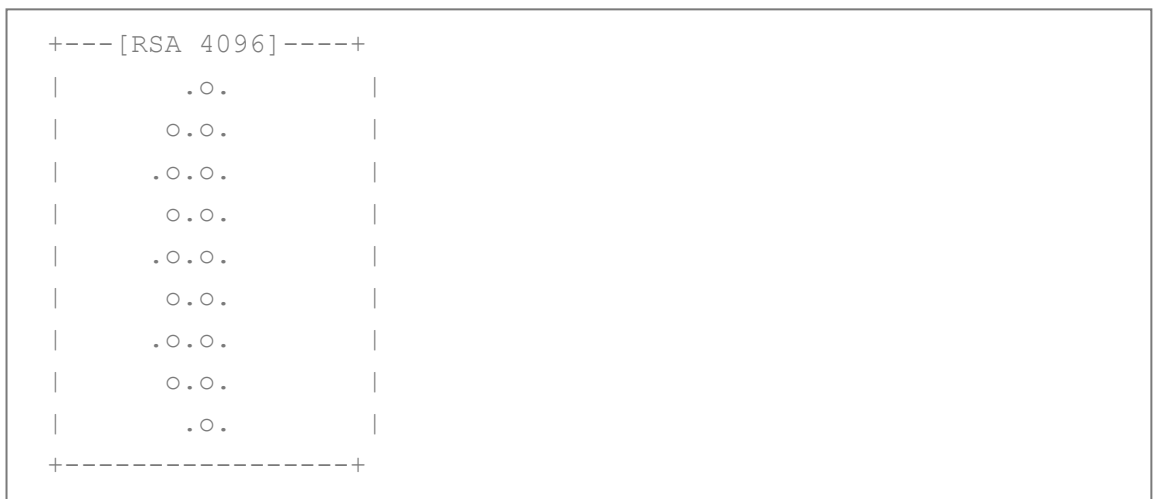
当你运行ssh-keygen命令时，如果你指定了-v（verbose，详细信息）选项，它会生成随机艺术图像。例如：

bash复制

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com" -v
```

运行这个命令后，除了生成密钥对，还会输出类似以下的随机艺术图像：

复制



如何理解随机艺术图像

随机艺术图像看起来像一个由点和线组成的图案，每个图案都是根据密钥的指纹生成的。虽然它看起来像是随机的，但实际上它是基于密钥的唯一性生成的。如果你对同一个密钥多次运行`ssh-keygen -v`，生成的图像会是相同的。

实际用途

在实际使用中，随机艺术图像主要用于以下场景：

1. 密钥验证：当你需要验证一个密钥是否是你所期望的密钥时，可以比较随机艺术图像。如果图像相同，说明密钥是相同的。
2. 密钥管理：在管理多个密钥时，随机艺术图像可以帮助你区分不同的密钥。

总结

随机艺术图像是SSH密钥的一个可视化表示，主要用于增强密钥的可读性和验证性。虽然它看起来像是随机的，但实际上它是根据密钥的指纹生成的，具有唯一性。

如果你在生成SSH密钥时没有看到随机艺术图像，可以尝试加上`-v`选项重新运行命令。