

Git

- [Git Hub Using SSH](#)

Git Hub Using SSH

1. Use Personal Access Tokens (PAT)

You can create a Personal Access Token in GitHub and use it instead of your password. Here's how:

1. **Generate a Personal Access Token:**

- Go to your GitHub account settings.
- Navigate to **Developer settings** > **Personal access tokens** > **Tokens (classic)**.
- Click on **Generate new token**.
- Select the scopes/permissions you need (for repository access, select repo).
- Click **Generate token** and make sure to copy the token (you won't be able to see it again).

2. **Use the Token for Authentication:** When prompted for a username and password in the terminal:

- Use your GitHub username as the username.
- Use the generated token as the password.

2. Use SSH Authentication

Alternatively, you can use SSH keys for authentication. Here's how to set it up:

1. **Generate an SSH Key (if you don't have one):**

bash

```
ssh-keygen -t rsa -b 4096 -C "cto@issentialsolutions.com"
```

Follow the prompts to save the key (usually in ~/.ssh/id_rsa).

2. **Add the SSH Key to your GitHub Account:**

- Copy the SSH key to your clipboard:

bash

```
cat ~/.ssh/id_rsa.pub
```

- Go to GitHub settings > **SSH and GPG keys** > **New SSH key**.
- Paste the key and give it a title.

3. **Clone the Repository Using SSH:** Change the repository URL to use SSH:

bash

```
git clone git@github.com:krishnaaka-online/is-odoo.git
```

To See the private Key

```
cat ~/.ssh/id_rsa
```

Summary

Choose either Personal Access Tokens or SSH authentication to resolve the authentication issue. If you frequently use GitHub, SSH is often more convenient, while PATs are simpler for one-off uses.

Steps to Configure Host Key Verification in Jenkins

- 1. Access Jenkins Configuration:**
 - Open your Jenkins dashboard in a web browser.
 - Go to **Manage Jenkins**.
- 2. Configure Git Host Key Verification:**
 - In the **Manage Jenkins** section, look for **Security** or **Git Host Key Verification Configuration**.
 - If you don't see this option, you might need to install the **Git Plugin** or check the plugin settings.
- 3. Add GitHub to Known Hosts:**
 - You need to add the GitHub host key to the known_hosts file. You can do this manually or through Jenkins:
 - **Manually:**
 - Open a terminal on the Jenkins server and run:
bash

```
ssh-keyscan -t ed25519 github.com >> ~/.ssh/known_hosts
```
 - Ensure that the file exists and is correctly populated.
 - **Through Jenkins:**
 - If the Jenkins interface allows, you can add the host key directly there.

4. **Verify the Known Hosts File Exists:**

- Ensure that the `known_hosts` file exists at the expected location. If the file does not exist, create it:

bash

```
touch ~/.ssh/known_hosts
```

- Then, add the GitHub host key as mentioned above.

5. **Test SSH Connection:**

- On the Jenkins server, test the SSH connection to GitHub:

bash

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

- You should see a message indicating successful authentication.

6. **Check Permissions:**

- Ensure that the user running Jenkins has permission to read the `~/.ssh/known_hosts` file and that the file has the correct permissions set (typically 600).

Summary

By configuring the host key verification in Jenkins and ensuring that the `known_hosts` file is properly set up, you should be able to resolve the host key verification issue and allow Jenkins to clone the GitHub repository successfully. If you continue to encounter issues, double-check the SSH keys and permissions.