Adding SSH Keys to GitHub: Complete Guide

> SSH keys allow secure interaction with GitHub without re-entering credentials. This guide covers checking for keys, generation, and GitHub account configuration.

1. Check for Existing SSH Keys

Open your terminal and check the `.ssh` directory:

```
ls -al ~/.ssh
```

Look for files named `id_rsa.pub`, `id_ecdsa.pub`, or `id_ed25519.pub`.

2. Generate a New SSH Key

Run the following command, substituting your GitHub email:

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

When prompted to "Enter a file in which to save the key," press **Enter** to use the default location. Next, enter a secure passphrase for added security.

3. Add the SSH Key to ssh-agent

Ensure the SSH agent is running and add your private key:

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519
```

4. Add SSH Key to GitHub Account

Copy your public key content to the clipboard:

```
cat ~/.ssh/id_ed25519.pub
```

Follow these steps on the GitHub website:

1. Go to **Settings** via your profile icon.
2. Select **SSH and GPG keys** from the sidebar.
3. Click **New SSH key**.
4. Provide a title (e.g., "Work Laptop") and paste your key into the text box.
5. Click **Add SSH key**.

5. Verify Communication

Test the connection to ensure authentication is successful:

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

You should receive a success message: *"Hi username! You've successfully authenticated..."*

Best Practices

- **Passphrase:** Always use a passphrase to protect your private key on local storage.
- **Multiple Keys:** Use different SSH keys for separate accounts (Work vs. Personal) using an SSH config file.
- **Key Rotation:** Periodically rotate your SSH keys for better security posture.

Guide compiled from GeeksforGeeks and GitHub Technical Docs.