SSH Key Generation

What is an SSH key?

- It's a **pair of cryptographic keys** (a private key and a public key).
- Used to authenticate you to servers (like GitHub) without a password.
- The **private key** stays **only on your computer**.
- The **public key** can be shared freely (added to GitHub, servers, etc.).
- Together, they provide a **secure, passwordless login**.

Step 1: Generate a key pair

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

Explanation:

- **`ssh-keygen`** → command to create SSH keys.
- **`-t ed25519`** → specifies the algorithm.
    - `ed25519` is modern, fast, and secure.
    - If your system doesn't support it, use **`-t rsa -b 4096`** (RSA, 4096-bit).
- **`-C "your_email@example.com"`** → a label so you know what this key is for (commonly your GitHub email).

You'll then see prompts:

```
Enter file in which to save the key (/home/you/.ssh/id_ed25519):
```

- Just press **Enter** for the default path.
- You can also give a custom filename if you want multiple keys (e.g., `id_github`, `id_gitlab`).

```
Enter passphrase (empty for no passphrase):
```

- **Empty** → no passphrase (faster, but less secure).
- **With passphrase** → adds an extra password protecting the private key (recommended if others have access to your machine).

Step 2: Files created

- **Private key**: `~/.ssh/id_ed25519`
    - Stays on your machine, never shared.

    - Used to "prove" your identity.
- **Public key**: `~/.ssh/id_ed25519.pub`
    - Can be copied to GitHub, servers, etc.
    - Anyone with the public key cannot guess your private key.

⚙️ SSH Agent in Detail

What is the SSH Agent?

- A **background program** that securely holds your **private keys** in memory.
- It prevents you from re-typing your passphrase every time you use the key.
- When a program (like Git) needs your key, it **asks the agent** instead of reading the file directly.

Step 1: Start the agent

```
eval "$(ssh-agent -s)"
```

- **`ssh-agent`** → starts the agent program.
- **`-s`** → outputs shell commands.
- **`eval "$(…)"`** → runs those commands in your shell, setting environment variables like `SSH_AUTH_SOCK` (where the agent listens).

After running it, you'll see something like:

```
Agent pid 12345
```

Step 2: Add your private key

```
ssh-add ~/.ssh/id_ed25519
```

- This loads the private key into the agent.
- If your key has a passphrase, you'll enter it **once** here.
- The agent will now use the unlocked key until you log out or restart.

Step 3: Verify loaded keys

```
ssh-add -l
```

Shows all the keys currently managed by the agent.

🛠️ Putting It Together (GitHub Example)

1. Generate keys → `ssh-keygen -t ed25519 -C "email@example.com"`
2. Start agent → `eval "$(ssh-agent -s)"`
3. Add key → `ssh-add ~/.ssh/id_ed25519`
4. Copy public key → `cat ~/.ssh/id_ed25519.pub`
5. Paste into GitHub → Settings → SSH Keys → New Key
6. Test → `ssh -T git@github.com`

🔐 **Summary**

- **SSH key pair** = digital identity (private key = secret, public key = ID card).
- **SSH agent** = a helper that remembers your unlocked private key so you don’t need to re-enter your passphrase every time.