Accessing Virtual Machine (SSH)

- Generate SSH Key
- Install Private Key (SSH)
- Access Virtual Machine (VM) using SSH Key (Windows / Macbook)

Generate SSH Key

Here are step-by-step instructions to generate an SSH key on both **Windows** and **Mac** systems:

For Windows (Windows 10/11):

1. **Ensure OpenSSH Client is Installed**
   - Open **Settings** > **Apps** > **Apps & Features** > **Optional Features**.
   - Check if **OpenSSH Client** is listed.
   - If not, click **Add a feature**, find **OpenSSH Client**, and install it.
   - Restart your computer if needed.
2. **Open Command Prompt or PowerShell**
   - Press `Windows + R`, type `cmd` or `powershell`, and press Enter.
3. **Generate SSH Key**
   - Type the command:

     ```
     ssh-keygen
     ```

   - Press Enter to accept the default save location (`C:\Users\<username>\.ssh\id_rsa`).
4. **Set a Passphrase (Optional)**
   - When prompted, enter a passphrase for extra security, or just press Enter twice to skip.
5. **Complete**
   - The keys are generated, usually two files: `id_rsa` (private key) and `id_rsa.pub` (public key) under `.ssh` folder in your user directory.

For Mac (macOS):

1. **Open Terminal**
   - Find Terminal via Spotlight Search or in **Applications > Utilities**.
2. **Generate SSH Key**
   - Run the command, replacing your email with your own:

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

   - If your macOS or server doesn't support Ed25519, use:

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

3. **Save Key Location**

- Press Enter to accept the default file location (`/Users/you/.ssh/id_ed25519` or `/Users/you/.ssh/id_rsa`).

4. **Set a Passphrase (Optional)**
   - Enter a secure passphrase or press Enter to leave it blank.
5. **Add SSH Key to ssh-agent**
   - Start the ssh-agent:

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

   - Add your private key:

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

   - If using RSA:

     ```
     ssh-add -K ~/.ssh/id_rsa
     ```

In both OS, your generated SSH public key is the one you copy and add to services like GitHub, servers, etc. The private key should remain secure and never shared.

Install Private Key (SSH)

Here are step-by-step instructions on how to **store an SSH private key securely** once generated on both **Windows** and **Mac** systems:

Windows

1. **Default Location**
   - By default, SSH private keys are saved in your user profile directory under:

     ```
     C:\Users\<your_username>\.ssh\
     ```

   - The private key file is usually named `id_rsa` or similar.
2. **Set File Permissions**
   - Ensure only your user account has access to the private key file:
     - Right-click the private key file > Properties > Security tab.
     - Remove access for any other users or groups except yourself.
3. **Use SSH-Agent for Secure Key Storage**
   - Open **PowerShell as Administrator**.
   - Enable and start the ssh-agent service:

     ```
     Get-Service ssh-agent | Set-Service -StartupType Automatic
     Start-Service ssh-agent
     ```

   - Add your private key to ssh-agent to avoid typing your passphrase every time:

     ```
     ssh-add $env:USERPROFILE\.ssh\id_rsa
     ```

   - The ssh-agent stores your private keys encrypted, protected by your Windows user credentials.
4. **Backup Private Key Securely**
   - Copy the private key file to an **encrypted external drive** or secure **password manager**.
   - Avoid storing private keys in shared or non-encrypted folders.
   - Do NOT share the private key.

Mac (macOS)

1. **Default Location**
   - Private keys are stored in the hidden `.ssh` folder in your home directory:

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

- Use `ls -la ~/.ssh` in Terminal to view.

2. **Set Proper File Permissions**
   - Ensure your private key file is readable only by you:

```
chmod 600 ~/.ssh/id_rsa
```

3. **Add SSH Key to ssh-agent**
   - Start the ssh-agent:

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

   - Add your private key with:

```
ssh-add -K ~/.ssh/id_rsa
```

   - This stores your key securely in the macOS keychain.
4. **Backup Private Key Safely**
   - Store a copy on an **encrypted USB drive** or a **password manager** like 1Password.
   - Never upload your private key unencrypted to cloud storage.
   - Keep backups separate from your main machine to avoid loss.

Security best practices:

- Always protect your private key with a **strong passphrase** when generating it.
- Keep private key files with strict permissions so only your user account can read them.
- Use **ssh-agent** to manage keys in memory instead of repeatedly entering passwords or exposing keys.
- Backup private keys securely and do not share your private keys with anyone.

Access Virtual Machine (VM) using SSH Key (Windows / Macbook)

Here are step-by-step instructions to access a Server or Virtual Machine using SSH on both **Windows** and **Mac**:

Windows (Using PowerShell or Command Prompt)

1. **Ensure OpenSSH Client is Installed**
   - Go to **Settings > Apps > Optional Features**.
   - Confirm **OpenSSH Client** is installed; if not, click **Add a feature**, find it, and install.
2. **Open PowerShell or Command Prompt**
   - Press `Windows + R`, type `powershell` or `cmd`, and hit Enter.
3. **Connect via SSH**
   - Use the command:

     ```
     ssh username@server_ip
     ```

   - Replace `username` with your remote server username.
   - Replace `server_ip` with the server's domain name or IP address.
   - If the server uses a non-standard port, add `-p` followed by the port number:

     ```
     ssh username@server_ip -p port_number
     ```

4. **Verify and Accept Host Key**
   - At first connection, you'll be asked to verify the server's fingerprint.
   - Type `yes` and press Enter to continue.
5. **Authenticate**
   - Enter your password when prompted (or if using key authentication, make sure your private key is loaded or specify it with `-i`).
6. **You are connected when your prompt changes to the remote server's shell.**

Mac (Using Terminal)

1. **Open Terminal**
   - Use **Spotlight** or find Terminal in **Applications > Utilities**.
2. **Connect via SSH**
   - Type the command:

     ```
     ssh username@server_ip
     ```

   - Substitute `username` and `server_ip` with appropriate remote account and server info.
   - For custom ports, add `-p` option:

     ```
     ssh username@server_ip -p port_number
     ```

3. **Accept Host Key**
   - For the first-time connection, review and accept the host key by typing `yes`.
4. **Authenticate**
   - Enter your password or ensure your SSH private key is loaded (use `ssh-add` if needed).
5. **You are connected once you see the shell prompt of the remote server.**

Additional Tips:

- To specify a custom private key file, use the `-i` flag:

  ```
  ssh -i /path/to/private_key username@server_ip
  ```

- To exit the SSH session, type:

  ```
  exit
  ```