Here’s a step-by-step guide to set up a connection between your Linux virtual machine (VM) and GitHub using MobaXterm, generate an SSH key for authentication, and push your repository to GitHub:

**Step 1: Install and Set Up MobaXterm**

1. **Download and Install MobaXterm**:  
   Download MobaXterm and install it on your Windows machine.
2. **Connect to the Linux VM**:
   * Launch MobaXterm and click on **Session > SSH**.
   * Enter the **IP address** of your Linux VM and click **OK**.
   * Log in with your Linux VM username and password.[osboxes, password]
   * You now have an SSH terminal to your Linux VM.

**Step 2: Generate an SSH Key in the Linux VM**

1. **Generate SSH Key Pair**:
   * In the MobaXterm terminal connected to the Linux VM, run the following command:

bash

-

ssh-keygen -t ed25519 -C "your\_email@example.com"

* + If ed25519 is not supported, use RSA:

bash

-

ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"

* + Follow the prompts to save the key (default location: ~/.ssh/id\_ed25519 or ~/.ssh/id\_rsa). Leave the passphrase empty for simplicity.

1. **Copy the Public Key**:
   * Display the public key:

bash

-

cat ~/.ssh/id\_ed25519.pub

* + Copy the output.

**Step 3: Add the SSH Key to GitHub**

1. **Log in to GitHub**:  
   Go to [GitHub](https://github.com) and log in to your account.
2. **Add the SSH Key**:
   * Navigate to **Settings > SSH and GPG Keys > New SSH Key**.
   * Title the key (e.g., "Linux VM via MobaXterm").
   * Paste the copied public key into the "Key" field and save.

**Step 4: Configure Git on the Linux VM**

1. **Set Global Git Configuration**:
   * Set your username and email:

bash

-

git config --global user.name "Your Name"

git config --global user.email "your\_email@example.com"

1. **Test SSH Connection**:  
   Verify GitHub connectivity using SSH:

bash

-

ssh -T git@github.com

If successful, you’ll see a message like:

vbnet

-

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

**Step 5: Clone or Initialize a Git Repository**

1. **Clone an Existing Repository**:
   * Replace <repository-url> with the SSH URL of your GitHub repository:

bash

-

git clone git@github.com:username/repository.git

1. **Initialize a New Repository**:
   * Create a new local repository:

bash

-

mkdir my-repo && cd my-repo

git init

* + Add files, commit, and link to GitHub:

bash

-

git add .

git commit -m "Initial commit"

git remote add origin git@github.com:username/repository.git

**Step 6: Push Code to GitHub**

1. **Push to GitHub**:  
   Push your changes to the remote repository:

bash

-

git branch -M main

git push -u origin main

1. **Verify on GitHub**:  
   Go to the GitHub repository page to confirm that the files have been uploaded.

**Step 7: Enable SSH Key Usage in MobaXterm**

1. **Load the SSH Key**:
   * In the MobaXterm terminal, start the SSH agent:

bash

-

eval "$(ssh-agent -s)"

* + Add the SSH private key:

bash

-

ssh-add ~/.ssh/id\_ed25519

1. **Automate SSH Key Addition (Optional)**:  
   Add the following lines to your ~/.bashrc file to load the SSH key automatically:

bash

-

eval "$(ssh-agent -s)"

ssh-add ~/.ssh/id\_ed25519

Following these steps ensures your Linux VM is connected to GitHub through MobaXterm using SSH authentication, and your code can be securely pushed to your GitHub repository.

|  |
| --- |
| To get the IP address of your Linux Virtual Machine (VM), you can use the following methods:  **1. Check Using ip Command**   1. Open the terminal on your Linux VM. 2. Run the following command to display all network interfaces and their details:   bash  -  ip addr   1. Look for the active network interface (e.g., eth0, ens33, wlan0). You will see a line like this:   -  inet 192.168.1.100/24   * + The IP address is 192.168.1.100 in this example.   **2. Check Using ifconfig Command *(Older Systems)***   1. If the ip command is unavailable, you can use ifconfig (part of the net-tools package). Run:   bash  -  ifconfig   1. Look for the active network interface and the line that starts with inet. Example:   -  inet 192.168.1.100 netmask 255.255.255.0 broadcast 192.168.1.255  **3. For VirtualBox VMs**   1. Open **VirtualBox Manager** on your host machine. 2. Go to the **Network Settings** of your VM:    * Select the VM, click **Settings > Network > Adapter 1 > Attached To**.    * If it's **NAT**, the internal VM IP might not be directly accessible. Use **Bridged Adapter** for direct access. 3. After setting **Bridged Adapter**, get the IP from the VM using the ip addr command.   **4. For VMs Using NAT and Port Forwarding**  If your VM uses **NAT**:   1. Open the terminal on the VM and find the internal IP (e.g., 10.0.2.15) using:   bash  -  ip addr   1. On your host machine, configure **port forwarding** in your VM settings to map the internal IP to a host port.   **5. Query Your Public IP (If Connected to the Internet)**  If your VM is connected to the internet and you need its public IP:   1. Use a service like curl to fetch your public IP:   bash  -  curl ifconfig.me   1. The output will be your VM's public IP address.   **6. Check VM’s IP in the Host Machine**  If you're running the VM on a host machine and know the virtual network range:   1. Use the following command on the host to find the VM's IP:   bash  -  arp -a   1. Match the MAC address of the VM's network interface with the corresponding IP.   **Notes:**   * Ensure your VM is properly connected to a network (e.g., via Bridged Adapter or NAT). * If you are using a cloud VM (e.g., AWS, Azure), the IP is displayed in the management console. |