UCSC

SSH - Secure Shell

Introduction to SSH

SSH (Secure Shell) is a cryptographic network protocol that allows secure remote login and command execution on remote machines. It replaces older protocols like Telnet, which transmit data in plain text.

Key Features:

- Encrypts all data, including passwords.
- Secure file transfer with scp(Secure copy Protocol) and sftp(Secure File Transfer Protocol).
- Enables secure remote administration.

1. Installing SSH

Install OpenSSH Server

To accept SSH connections, the remote machine must have the SSH server installed.

```
sudo apt update
sudo apt install openssh-server
```

Install SSH Client

Most Linux systems come with the client pre-installed. You can check with:

```
ssh -V
```

If not installed:

```
sudo apt install openssh-client
```

2. Starting and Managing the SSH Service

Start SSH Server

```
sudo systemctl start ssh
```

Enable SSH to run on boot

```
sudo systemctl enable ssh
```

Check SSH Status

```
sudo systemctl status ssh
```

3. Basic SSH Connection

Syntax:

```
ssh username@hostname_or_ip
```

Example:

```
ssh alice@192.168.1.100
```

If it's the first connection, it will ask for confirmation to trust the host. Then it will ask for the user's password.

4. Running Remote Commands via SSH

Instead of starting a full session, you can run one command on the remote server:

```
ssh alice@192.168.1.100 "uptime"
```

5. SSH Key-Based Authentication

Generate SSH Keys:

```
ssh-keygen -t rsa -b 4096
```

By default, this creates:
`~/.ssh/id_rsa` (private key)
`~/.ssh/id_rsa.pub` (public key)

Copy Public Key to Remote Machine:

```
ssh-copy-id alice@192.168.1.100
```

Now you can log in without a password.

6. Secure File Transfers with SCP

Copy file from local to remote:

```
scp file.txt alice@192.168.1.100:/home/alice/
```

Copy file from remote to local:

```
scp alice@192.168.1.100:/home/alice/file.txt .
```

7. Secure File Transfers with SFTP

```
sftp alice@192.168.1.100
```

Once connected:

ls, cd, put, get, bye can be used like in FTP.

8. SSH Configuration Files

Client Config (~/.ssh/config)

```
Host myserver
    HostName 192.168.1.100
    User alice
    Port 22
```

Then connect with:

```
ssh myserver
```

9. Useful SSH Options

- -p - Specify port number. ssh -p 2222 user@host
- -v - Verbose output for debugging.
- -X - Enable X11 forwarding (GUI apps).
- -N - Do not execute remote command (useful for port forwarding).

10. SSH Hardening Tips

Change the default port in /etc/ssh/sshd_config:

```
Port 2222
```

Disable root login:

```
PermitRootLogin no
```

Use only key-based login:

```
PasswordAuthentication no
```

After changes:

```
sudo systemctl restart ssh
```

Activities to try

Activity 1: Enable and Test SSH Locally

1. Install openssh-server.
2. Start and enable the SSH service.
3. Run ip a to get your local IP address.
4. SSH into your own machine: ssh yourusername@127.0.0.1
5. Exit with exit.

Activity 2: Key-Based Login Setup

1. On your main machine, generate SSH keys:
   ```
   ssh-keygen
   ```
2. Copy the public key to the target machine:
   ```
   ssh-copy-id username@target_machine_ip
   ```
3. Test logging in without a password.

Activity 3: Secure File Transfer Practice

1. Create a test file:
   ```
   echo "Hello SSH" > hello.txt
   ```
2. Copy it to the remote machine using scp.
3. Retrieve it back with scp.

Activity 4: Remote Command Execution

1. Run the following via SSH:
   ```
   ssh user@remote_ip "df -h"
   ```
2. Try this:
   ```
   ssh user@remote_ip "cat /etc/os-release"
   ```

Secure Shell (SSH) with PuTTY

Introduction to PuTTY

PuTTY is a free and open-source SSH client for Windows that provides terminal emulation to connect to Linux systems securely.

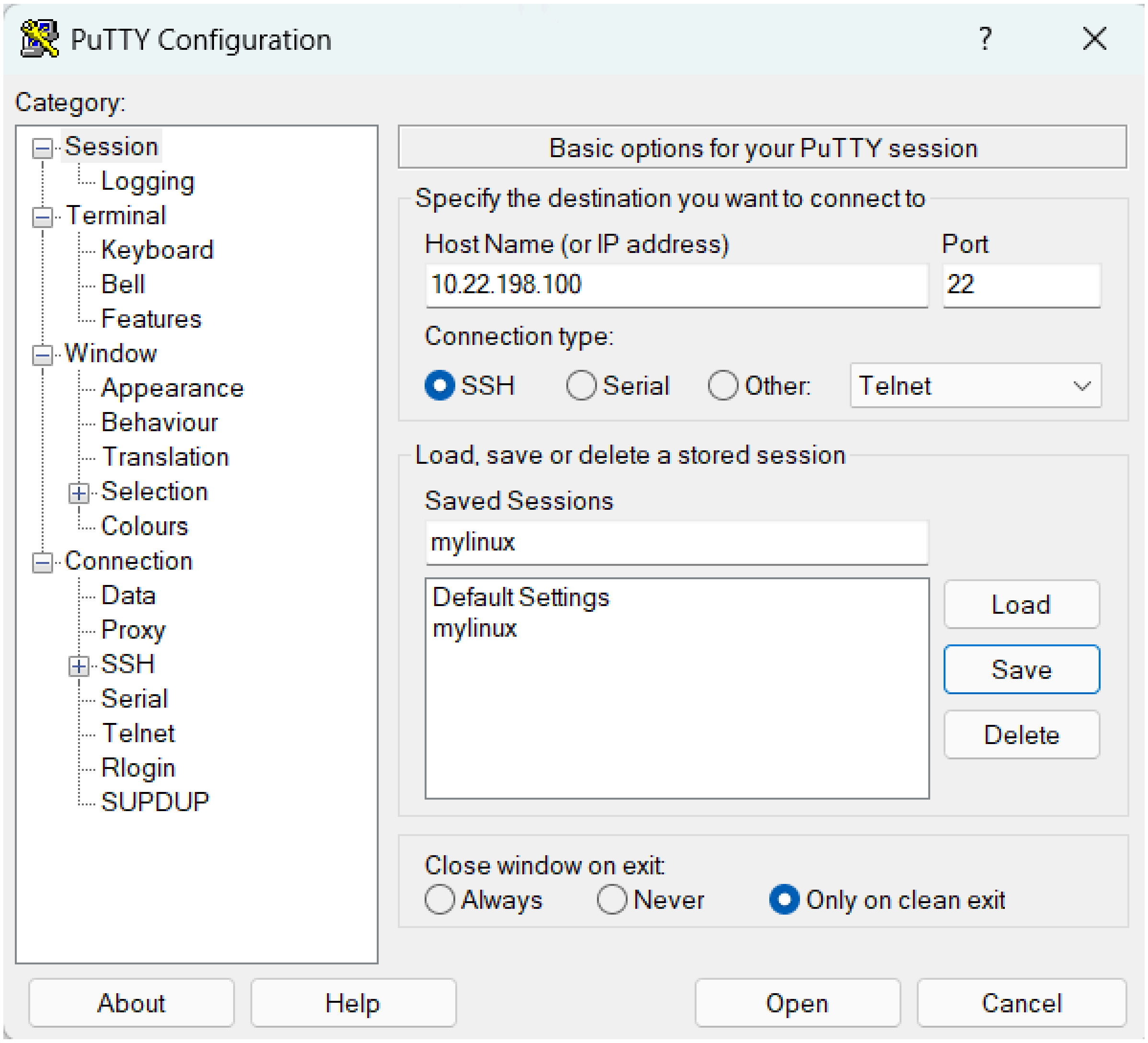
1. Downloading and Installing PuTTY

- Visit the official website: https://www.putty.org
- Download the Windows Installer (putty-64bit-<version>.msi).
- Install using default settings.
- Launch PuTTY from the Start menu.

2. Connecting to a Linux Server Using SSH

- Open PuTTY.
- In the Host Name (or IP address) field, enter the IP or domain of your Linux machine.
- Set Port to 22.
- Ensure Connection type is set to SSH.
- (Optional) Save the session:
  - Under Saved Sessions, type a name like MyLinuxVM
- Click Save.

- Click Open to start the session.
- When prompted, enter your username and password.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 10.22.198.100

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions: mylinux

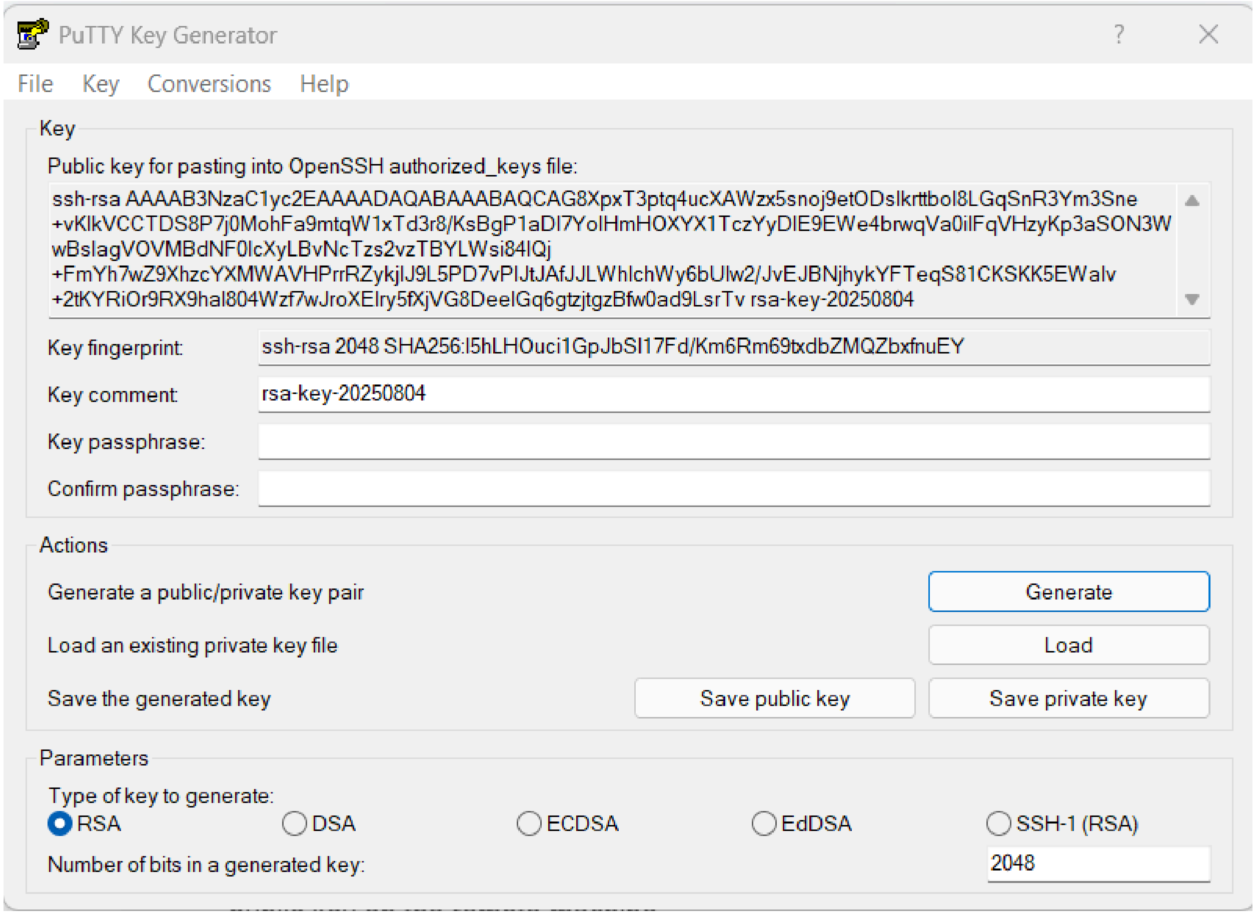
Default Settings
mylinux

Close window on exit: ○ Always ○ Never ● Only on clean exit

3. Using SSH Keys with PuTTY

Step 1: Generate Keys with PuTTYgen

- Open PuTTYgen from the Start menu.
- Choose RSA, click Generate, and move your mouse around.
- Save:
  - Private key (.ppk) – save this for PuTTY.
  - Public key – copy and paste this into your Linux server’s ~/.ssh/authorized_keys.



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCAG8XpxT3ptq4ucXAWzx5snoj9etODslkrttbol8LGqSnR3Ym3Sne
+vKlkVCCTDS8P7j0MohFa9mtqW1xTd3r8/KsBgP1aDI7YolHmHOXYX1TczYyDIE9EWe4brwqVa0ilFqVHzyKp3aSON3W
wBslagVOVMBdNF0lcXyLBvNcTzs2vzTBYLWsi84IQj
+FmYh7wZ9XhzcYXMWAVHPrrRZykjlJ9L5PD7vPIJtJAfJJLWhlchWy6bUlw2/JvEJBNjhykYFTeqS81CKSKK5EWalv
+2tKYRiOr9RX9hal804Wzf7wJroXElry5fXjVG8DeelGq6gtzjtgzBfw0ad9LsrTv rsa-key-20250804
```

Key fingerprint: ssh-rsa 2048 SHA256:I5hLHOuci1GpJbSI17Fd/Km6Rm69txdbZMQZbxfnuEY

Key comment: rsa-key-20250804

Key passphrase:

Confirm passphrase:

Generate a public/private key pair

Load an existing private key file

Save the generated key

Type of key to generate:

● RSA ○ DSA ○ ECDSA ○ EdDSA ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

Step 2: Upload the Public Key to the Server

- Use another method (like scp, SFTP via FileZilla, or cloud console) to place the public key on the remote machine.

- **Example:**

```
mkdir -p ~/.ssh
nano ~/.ssh/authorized_keys
# Paste the public key and save
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

Step 3: Use Key in PuTTY

- In PuTTY: Go to Connection > SSH > Auth
- Click Browse, and select the .ppk file.
- Go back to Session, select your saved session, click Save.
- Click Open to connect with key-based authentication.

4. Transferring Files via PuTTY (Using PSCP or FileZilla)

PuTTY doesn’t have built-in file transfer GUI.
Use:

PSCP (PuTTY Secure Copy)
Download pscp.exe from the PuTTY site and place it in C:\Windows\System32.
Example:
Copy file from Windows to Linux:

```
pscp C:\Users\user\Documents\hello.txt username@192.168.1.100:/home/username/
```

From Linux to Windows:

```
pscp username@192.168.1.100:/home/username/report.txt C:\Users\user\Downloads\
```

GUI Alternative: FileZilla (with SFTP)
Install FileZilla.
Use SFTP protocol, enter:

Host: sftp://<ip>
Username
Password
Port: 22

Activities to try

Activity 1: First SSH Connection with PuTTY

Launch PuTTY and connect to your Linux system using username and password.
Run the following commands:

- Hostname
- Uptime
- Whoami
- ls -al ~

Activity 2: Create and Manipulate Files

From your PuTTY session:

- Create a folder putty_practice
- Inside it, create a file intro.txt
- Add content: echo "Welcome to SSH with PuTTY" > intro.txt
- Display the content: cat intro.txt
- Copy intro.txt to welcome.txt
- Rename welcome.txt to greeting.txt
- Delete intro.txt
- Show all file operations using ls -l

Activity 3: SSH Key Login

- Generate an SSH key pair using PuTTYgen.
- Add the public key to your Linux server.
- Configure PuTTY to use the .ppk file.
- Login without password using the key.
- Run whoami to verify.

Activity 4: Secure File Transfer Using PSCP

- Create a file on your Windows desktop: message.txt
- Use pscp to transfer it to /home/yourusername/ on the server.
- SSH into the server with PuTTY and verify using ls.

How to SSH into a Virtual Machine (VM) from PuTTY

Prerequisites:

- A Linux-based virtual machine (on the cloud or local via VirtualBox/VMware).
- The public IP address or domain name of the VM.
- SSH port (default is 22).
- A username (usually ubuntu, ec2-user, root, or custom).
- Putty installed in your windows machine
- Either:
  - The password of the user, or
  - The private SSH key (.ppk file) if key-based login is required.

Step 1: Get VM SSH Connection Details

- IP address - e.g., 192.168.1.101 or public cloud IP
- Port - usually 22
- Username - e.g., ubuntu, centos, root, admin
- Authentication Password or SSH private key

Step 2: Connect Using PuTTY (with Password)

- Open PuTTY.
- In the Host Name (or IP address) field, enter the VM’s IP address.
- Leave the Port as 22.
- Under Connection type, choose SSH.
- (Optional) Go to Session > Saved Sessions, enter a name like MyVM, and click Save.
- Click Open.
- If prompted with a security alert, click Yes to trust the host.
- When asked for the username, enter it and press Enter.
- Enter the password when prompted (it won’t show while typing) and press Enter.

- You are now connected to your VM via SSH using PuTTY!

Optional Configurations

Keep Session Alive

Go to Connection tab.
Set “Seconds between keepalives” to 30 to avoid timeouts.

Auto-login Username

Go to Connection > Data
Set Auto-login username to ubuntu (or relevant user)

Example Activities After Connecting

Once connected, try these commands to explore the VM:

Whoami - Shows current user
uname -a - Shows system information
df -h - Disk usage
uptime - Show system uptime
sudo apt update - Update package list (Debian/Ubuntu-based)

Common Issues & Fixes

- **Connection refused - Check SSH service on VM and IP address. Ensure port 22 is open**.
    - How to Check if SSH is installed and running
      ```
      sudo systemctl status ssh
      ```
    - If it's active and running, you’ll see something like:
      ```
      ssh.service - OpenBSD Secure Shell server
      Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
      Active: active (running)
      ```
    - If it’s not running, start it:
      ```
      sudo systemctl start ssh
      ```
    - Enable it to start on boot:
      ```
      sudo systemctl enable ssh
      ```
    - Check if Port 22 is Open and Listening
      Run:
      ```
      sudo ss -tuln | grep :22
      ```
      or
      ```
      sudo netstat -tuln | grep :22
      ```

    - Output should show something like:
      ```
      tcp   LISTEN  0  128  0.0.0.0:22  0.0.0.0:*
      ```
      If you see this, SSH is listening on port 22.

- **Timeout - Verify VM is running. Check firewall/security group rules.**

    - How to Check Firewall Rules on the VM
      For UFW (Ubuntu)
      ```
      sudo ufw status
      ```
    - Make sure there's a rule like:
      ```
      22/tcp                    ALLOW       Anywhere
      ```
    - If not, allow SSH:
      ```
      sudo ufw allow 22
      sudo ufw reload
      ```

- Permission denied - Check username and SSH key. Make sure key is attached to the correct user.
- Host key alert - Click “Yes” if connecting for the first time.

Activities to Try

Activity 1: Create and Manage a Directory Tree

- Create a directory structure:
  ```
  mkdir -p ~/project/reports/2025
  ```
- Navigate between levels using relative paths.
- Use pushd and popd to track and return to previous directories.
- Delete one folder (rmdir or rm -r) and explain what happens.

Activity 2: File Operations and Permissions

- Create three files: log.txt, data.csv, and config.ini.
  ```
  touch log.txt data.csv config.ini
  ```
- Add some text to each file using echo or nano.
- Use ls -l to view permissions.
- Change permissions:
  ```
  chmod 600 config.ini
  chmod 755 log.txt
  ```
- Try accessing files with another user if possible.