

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

***Up SSH Key-Based Authentication Locally:*** *Generate an SSH key pair and configure it for passwordless login between two local machines or VMs*

**Name: VIJAYA NANDANA M Department: CSE**

A black and white logo

Description automatically generated

**Introduction**

Secure Shell (SSH) is a powerful protocol for securely accessing remote systems over a network. Typically, SSH uses a username and password for authentication, but SSH key-based authentication offers a more secure and convenient alternative. This Proof of Concept (PoC) focuses on setting up SSH key-based authentication between two local machines or VMs, enabling passwordless login for streamlined and secure remote access.

**Objectives**

Learn how to generate a 2048-bit RSA key pair for SSH key-based authentication.

* Gain hands-on experience with transferring the public key to the target machine and adding it to the authorized keys for passwordless login File.
* Understand the importance of setting correct permissions for the ~/.ssh directory and authorized keys file to secure SSH access.
* Verify that SSH key-based authentication works by testing passwordless login between the two machines.

**Importance**

1. **Security:** SSH key-based authentication is more secure than traditional password-based methods because it’s resistant to brute force attacks. The private key is never transmitted over the network, reducing the risk of interception.

2. **Automation:** For environments that require frequent or automated logins (such as scripts or DevOps pipelines), passwordless SSH login is essential for smooth operations without manual intervention.

3. **Streamlined Access:** Passwordless login simplifies the process of accessing remote machines, reducing the need to manually input passwords, and enabling seamless automation for system administration tasks.

4. **Best Practice:** Using SSH keys for authentication is widely considered a best practice in the industry, as it provides both security and convenience.

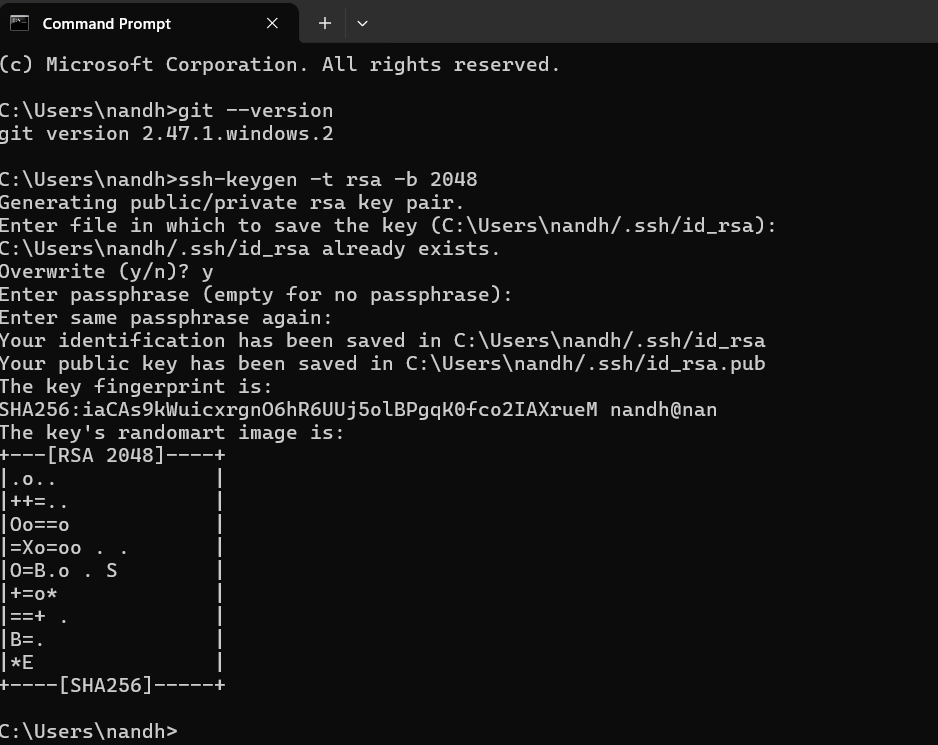
**Step by Step Overview**

**1. Generate SSH Key Pair**

* Open Git Bash or Terminal on the local machine from which you'll be accessing the target machine (VM or another local machine).
* Run the following command to generate an SSH key pair:

**ssh-keygen -t rsa -b 2048**

* This will create a 2048-bit RSA key pair. You will be prompted for a file to save the key. Press Enter to use the default path (~/.ssh/id\_rsa). Optionally, you can set a passphrase for extra security, but if you want passwordless login, leave it empty and press Enter.



**2. Copy Public Key to Target Machine**

* Now, you need to copy the public key (id\_rsa.pub) to the target machine (the one you're trying to log into).
* Use the ssh-copy-id command to copy the public key: s

*sh-copy-id username@target-machine-ip*

Replace username with the actual user on the target machine, and target-machine-ip with the target machine's IP address. If ssh-copy-id isn't available, you can manually copy the public key:

* On the source machine, display the public key:

*cat ~/.ssh/id\_rsa.pub*

* On the target machine, open the *~/.ssh/authorized\_keys* file (create it if it doesn't exist): *nano ~/.ssh/authorized\_keys*
* Paste the public key into the authorized\_keys file and save it.

### **3. Set Correct File Permissions**

### Ensure the correct file permissions are set for SSH to function properly: On the target machine, set the correct permissions for the ~/.ssh directory and authorized\_keys file:

### ***chmod 700 ~/.ssh chmod 600 ~/.ssh/authorized\_keys***

**4. Test Passwordless SSH Login**

Now that the key is copied and permissions are set, you can test passwordless SSH login from the source machine:

***ssh username@target-machine-ip***

You should be able to log in without entering a password. This sets up passwordless SSH login between the two machines

**Outcome:**

* Gain hands-on experience in generating and managing SSH key pairs using ssh-keygen, enhancing your understanding of public-key cryptography and its role in secure authentication.
* Successfully configure passwordless SSH login between two local machines or VMs, improving security and convenience by eliminating the need for password-based authentication.
* Learn to set correct file permissions for the ~/.ssh directory and authorized\_keys file, reinforcing security best practices for SSH access.
* Streamline remote access operations by enabling seamless, automated logins, which are essential for automation scripts, administrative tasks, and DevOps pipelines.
* Strengthen your foundational security skills by implementing a secure authentication mechanism that reduces exposure to brute-force attacks and credential theft.