

CLOUD SECURITY & MANAGEMENT
LAB

Name: Harsh Ranjan

SAP ID: 500097019

Roll no: R2142211262

SUBMISSION TO:

Ms. Avita Katal

Experiment 1 : To create key based authentication and login virtual machine from the host machine

Step 1: Generate RSA Key Pair on Ubuntu

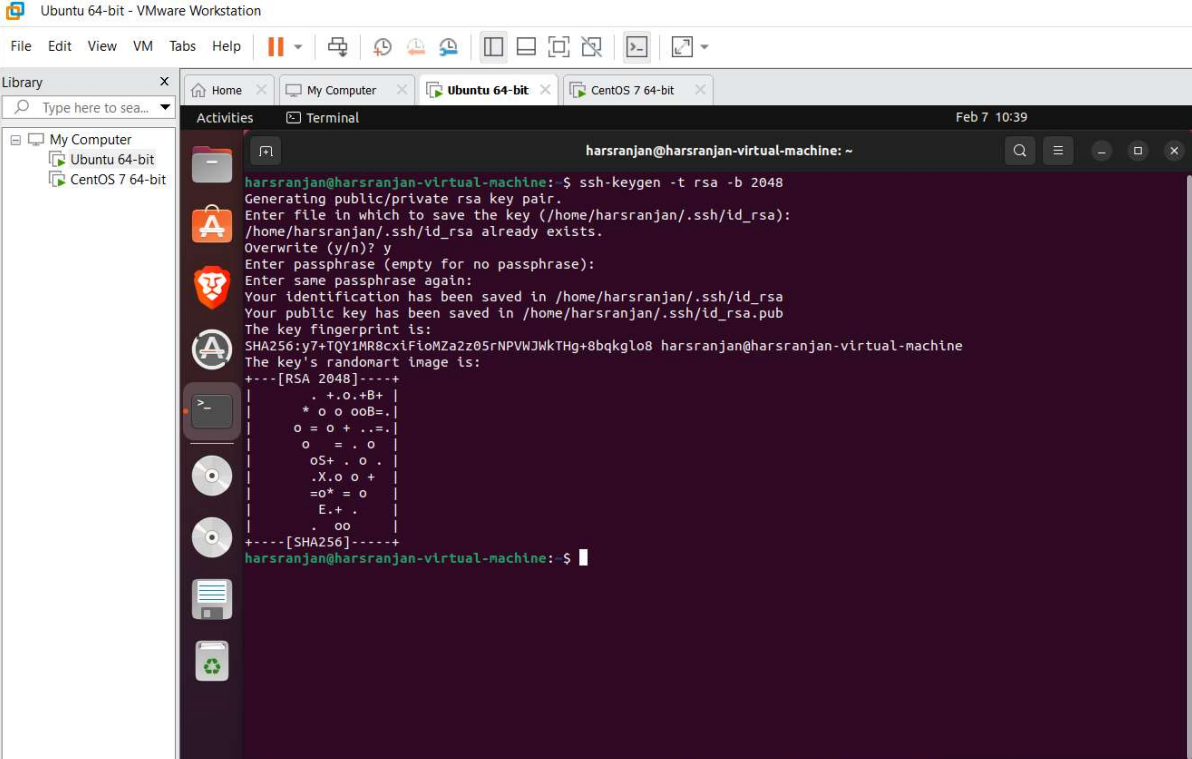
Open a terminal on your Ubuntu machine:

Generate RSA Key Pair on Ubuntu:

Open a terminal on your Ubuntu machine and run the following command:

: `ssh-keygen -t rsa -b 2048`

You will be prompted to enter a file path to save the keys and an optional passphrase. Press Enter to accept the default file path (usually /home/your_username/.ssh/id_rsa) and enter a passphrase if desired.



```
harsranjan@harsranjan-virtual-machine: ~  
harsranjan@harsranjan-virtual-machine: $ ssh-keygen -t rsa -b 2048  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/harsranjan/.ssh/id_rsa):  
/home/harsranjan/.ssh/id_rsa already exists.  
Overwrite (y/n)? y  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/harsranjan/.ssh/id_rsa  
Your public key has been saved in /home/harsranjan/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:y7+TQY1MR8cx1F1oMZa2z05rNPVWJWkTHg+8bqkgl08 harsranjan@harsranjan-virtual-machine  
The key's randomart image is:  
+---[RSA 2048]-----+  
  .+..+B+  
  * o o o o B+  
  o = o + . .+  
  o = . o  
  o S+ . o .  
  .X. o o +  
  =o* = o  
  E.+ .  
  . oo  
+---[SHA256]-----+  
harsranjan@harsranjan-virtual-machine: $
```

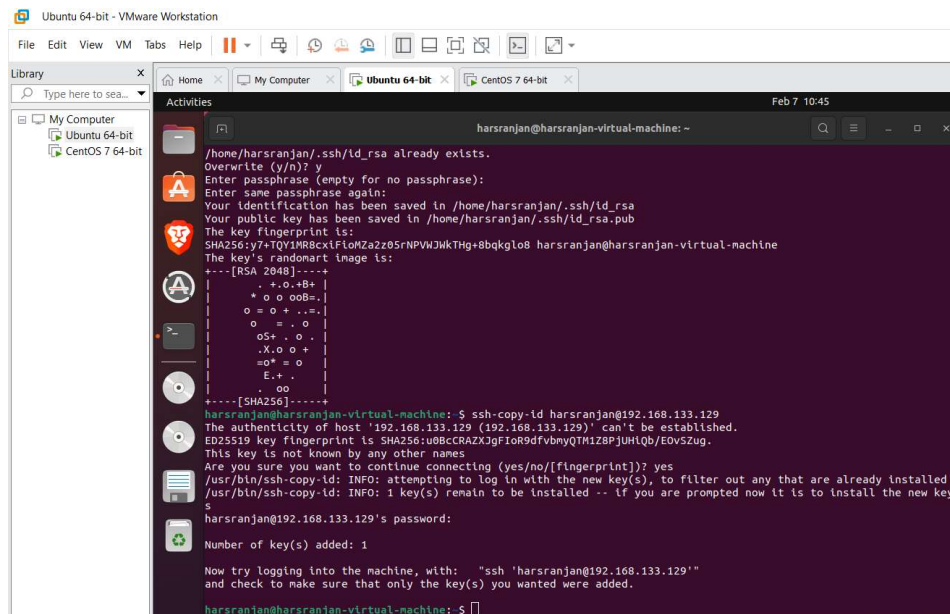
STEP 2: Copy Public Key to CentOS VM:

Using ssh-copy-id:

If your Ubuntu machine has the ssh-copy-id command installed, you can use it to copy the public key to the CentOS VM. Run the following command:

```
: ssh-copy-id username@centos_vm_ip_address
```

Replace username with your CentOS VM username and centos_vm_ip_address with the IP address of your CentOS VM.

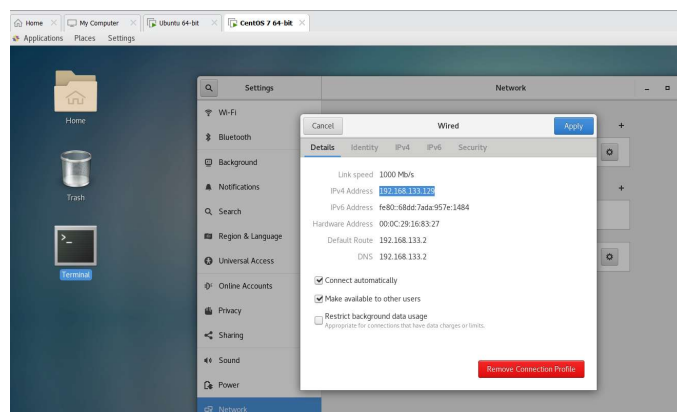


```
harsranjan@harsranjan-virtual-machine: ~
/home/harsranjan/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/harsranjan/.ssh/id_rsa
Your public key has been saved in /home/harsranjan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:y7+TQY1MR8cxIFi0M2a2z05rNPVMJkTHg+8bqkglo8 harsranjan@harsranjan-virtual-machine
The key's randomart image is:
+--[RSA 2048]-----+
  .+.o.+B+
  * o o oob+
  o = o + ..+
  o = . o
  oS+ . o .
  .X.o o +
  =o* = o
  E.+ .
  . oo
+-----[SHA256]-----+
harsranjan@harsranjan-virtual-machine: $ ssh-copy-id harsranjan@192.168.133.129
The authenticity of host '192.168.133.129 (192.168.133.129)' can't be established.
ED25519 key fingerprint is SHA256:u0BcCRAZk3gF1oR9dFvbnYQTH1ZBPJUHlQb/EOvsZug.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new key
$ harsranjan@192.168.133.129's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'harsranjan@192.168.133.129'"
and check to make sure that only the key(s) you wanted were added.
harsranjan@harsranjan-virtual-machine: $
```

Step 4: Test SSH Connection:

Check the IP address of the CentOS VM to connect with it.

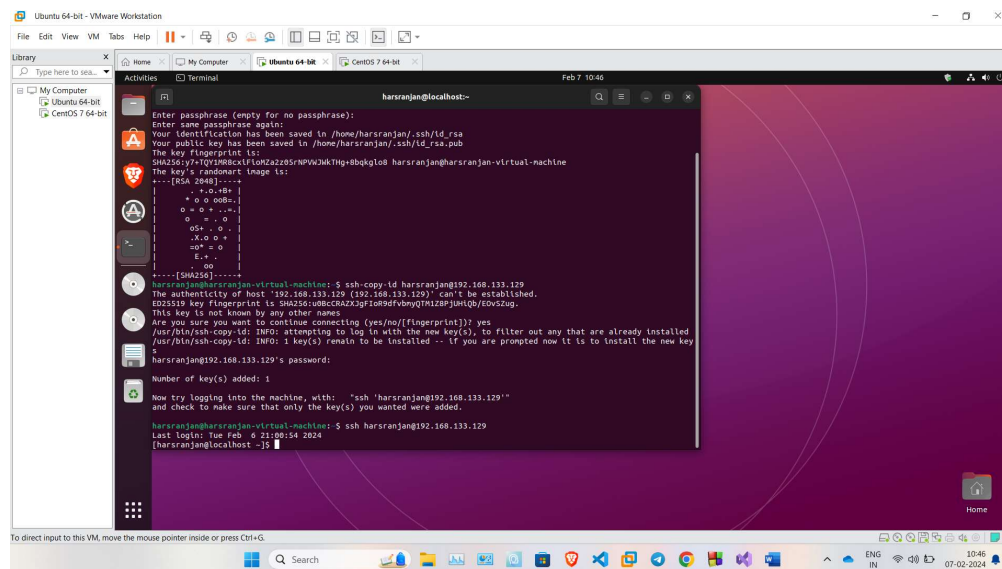


Step 4: Test SSH Connection:

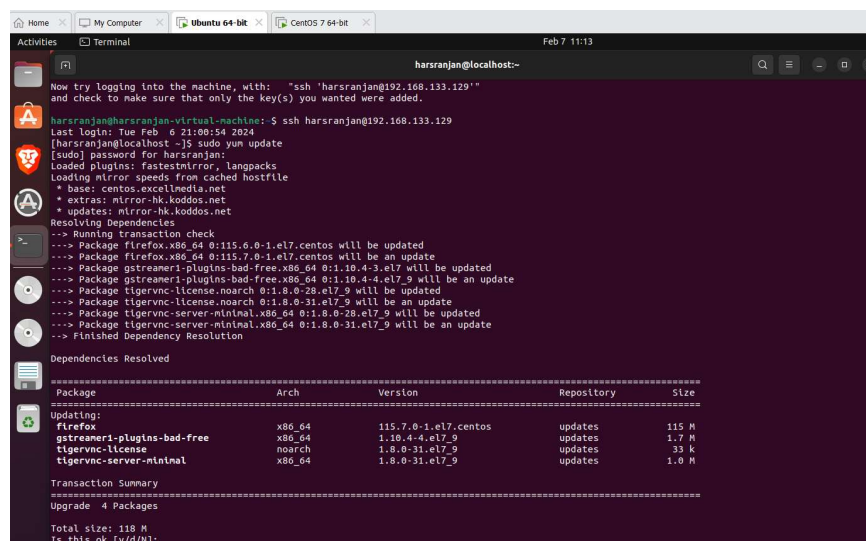
Once the public key is copied, try to SSH into your CentOS VM from the Ubuntu machine:

: `ssh username@centos_vm_ip_address`

You should now be able to log in without entering a password if everything was set up correctly.



STEP 4: Now you can access CentOS using Ubuntu



Q1. Define key-based authentication and explain how it differs from traditional password-based authentication.

ANSWER:

Key-based authentication involves using cryptographic keys to authenticate users instead of passwords. In this method, a pair of cryptographic keys (a public key and a private key) is used for authentication purposes. The public key is shared with the server while the private key is kept secure on the client-side. Traditional password-based authentication, on the other hand, relies on users providing a secret password to authenticate their identity.

Q2. What are the advantages of using key-based authentication over password-based authentication in terms of security?

ANSWER:

Key-based authentication offers several advantages over password-based authentication in terms of security:

- Stronger Authentication: Cryptographic keys are much harder to guess or crack compared to passwords, especially if they are long and complex.
- No Passwords: Since there are no passwords involved, there's no risk of password-based attacks such as brute force or dictionary attacks.
- No Password Sharing: Users don't need to remember or share passwords, reducing the risk of password compromise.
- Protection against Phishing: Key-based authentication is not susceptible to phishing attacks since there are no passwords to steal.

Q3. Describe the process of generating SSH keys. What are the components of an SSH key pair?

ANSWER:

The process of generating SSH (Secure Shell) keys involves the following steps:

- Use a key generation tool like ssh-keygen.
- Choose the type of key (RSA, DSA, ECDSA, or Ed25519).
- Specify the key size (typically 2048 or 4096 bits).
- Optionally provide a passphrase for added security.

- The key pair consists of a public key (which is shared with servers) and a private key (which is kept securely on the client).

Q4. Explain how SSH keys are used for authentication during the login process.

ANSWER:

During the SSH login process, the client sends its public key to the server. The server then checks whether the corresponding private key is authorized for access. If the private key matches an entry in the server's list of authorized keys, the client is granted access.

Q5. What is the role of the `authorized_keys` file in SSH key-based authentication?

ANSWER: The `authorized_keys` file on the server contains a list of public keys that are authorized to access the account. Each line in this file typically represents one public key. When a client attempts to connect, SSH checks this file to determine if the provided public key matches any of the authorized keys.

Q6. How do you securely transfer the public SSH key from the host machine to the virtual machine?

ANSWER:

To securely transfer the public SSH key from the host machine to the virtual machine, you can use methods like:

- Manually copying and pasting the public key.
- Using SSH to transfer the public key securely.
- Utilizing tools like `ssh-copy-id` for automated key transfer.

Q7. Describe the steps involved in configuring the SSH server on the virtual machine to allow key-based authentication.

ANSWER:

Configuring the SSH server on the virtual machine to allow key-based authentication involves:

- Editing the SSH server configuration file (typically `/etc/ssh/sshd_config`).

- Ensuring that public key authentication is enabled (PubkeyAuthentication yes).
- Optionally specifying the location of the authorized_keys file (AuthorizedKeysFile .ssh/authorized_keys).
- Restarting the SSH service for changes to take effect.

Q8. What security measures should be taken to protect the private SSH key on the host machine?

ANSWER:

To protect the private SSH key on the host machine, you should:

- Set appropriate permissions on the private key file (e.g., `chmod 600 ~/.ssh/id_rsa`).
- Use passphrase encryption for additional security.
- Regularly update and rotate keys.
- Avoid storing keys in publicly accessible locations.

Q9. How do you troubleshoot common issues with SSH key-based authentication, such as permission problems or incorrect key formats?

ANSWER:

To troubleshoot common issues with SSH key-based authentication:

- Ensure correct file permissions for the private key (`chmod 600 ~/.ssh/id_rsa`).
- Verify that the public key is correctly added to the authorized_keys file on the server.
- Check for formatting errors in the public key file.
- Examine SSH server logs for any error messages or warnings.
- Test connectivity using verbose SSH (`ssh -v`) for detailed output.