**Resolved Issues When My Local is Windows and the Server is Ubuntu**

**Scenario: Ubuntu Server SSH Connection Issue**

When your PEM key is not working after multiple attempts and troubleshooting, and you're encountering the following error:

ubuntu@ip.compute-1.amazonaws.com: Permission denied (publickey)

**Step 1: Check the Public Key on the Server**

If you have access to the instance via another session or user, follow these steps:

1. **Log in to the server using an alternate working method** (e.g., through another user or session).
2. **Verify the authorized\_keys file**:
   * Check that your public key is present in the ~/.ssh/authorized\_keys file:

bash

Copy code

cat ~/.ssh/authorized\_keys

* + If the key is missing, add the public key extracted from your .pem file using the following command:

ssh-keygen -y -f /path/to/your-key.pem >>~/.ssh/authorized\_keys

**Step 2: Ensure Correct Permissions for the .ssh Directory and authorized\_keys File**

Make sure that the permissions for the .ssh directory and the authorized\_keys file are correct to allow SSH access:

sudo chmod 700 /home/ubuntu/.ssh

sudo chmod 600 /home/ubuntu/.ssh/authorized\_keys

sudo chown -R ubuntu:ubuntu /home/ubuntu/.ssh

**Step 3: Resolving Permission Issues and Key Management on Windows**

If you're still facing issues, follow these steps as they worked for me.

**Generate a Public/Private Key Pair on Windows**

1. **Create the SSH Key Pair at a specific location on your local Windows machine**:

ssh-keygen -t rsa -b 4096 -f C:\path\to\directory\custom-key-name

This will create the file my\_ssh\_key.pub (the public key).

You can also generate the public key from an existing private key using:

bash

Copy code

ssh-keygen -y -f /path/to/your-key.pem > /path/to/your-key.pub

* + Replace C:\path\to\directory\custom-key-name with your desired directory and key name.

**Set Correct Permissions on the Private Key**

On Windows, SSH expects strict permissions for the private key. To fix this:

1. **Open the private key’s properties**:
   * Right-click the private key file (e.g., my\_ssh\_key), then go to Properties → Security tab.
2. **Grant read permissions only to the user**:
   * Make sure only your user has full access, and remove access for other users (e.g., "Everyone").

Alternatively, use the following commands to remove inheritance and grant full permissions to your user:

bash

Copy code

icacls "C:\Users\Lenovo\Downloads\key\my\_ssh\_key" /inheritance:r

icacls "C:\Users\Lenovo\Downloads\key\my\_ssh\_key" /grant:r Lenovo:F

**Copy the Public Key to the Server**

**Important:** Make sure to back up the existing authorized\_keys file on the server before making any changes.

1. **Copy the public key from the generated key pair to the authorized\_keys file on the server**. **Do not modify the existing keys** on the server.
   * Open the public key file (my\_ssh\_key.pub) in Notepad and copy its content.
2. **Paste the public key on the server**:
   * Log in to the server (with an alternate method if necessary) and open the authorized\_keys file:

bash

Copy code

nano ~/.ssh/authorized\_keys

* + Paste the public key content from Notepad.

**Step 4: SSH into the Server Using the Private Key**

Now, you should be able to use the private key to connect to the server:

bash

Copy code

ssh -i .\my\_ssh\_key ubuntu@ip

This command will allow you to log into the Ubuntu server using the private key you generated on your Windows machine.

**Additional Notes:**

* **Permission Issues**: It's critical to ensure that the permissions on both the private key file (on your Windows machine) and the authorized\_keys file (on your Ubuntu server) are correct. Otherwise, SSH will fail to authenticate.
* **Public/Private Key Management**: Make sure you're using the **private key** (my\_ssh\_key) for the -i flag in the SSH command. The **public key** (my\_ssh\_key.pub) is used on the server and should never be used in the command to connect.