# **SSH KEY GENEREATE**

If you have followed all the steps and still encounter the Permission denied (publickey) error, let's delve deeper into the problem. Here are some advanced troubleshooting steps:

**Advanced Troubleshooting Steps**

**1. Verify the SSH Key Pair**

Ensure that the SSH key pair is correctly generated and not corrupted.

On the Jenkins server, check the key format:

sh

Copy code

cat ~/.ssh/id\_rsa.pub

cat ~/.ssh/id\_rsa

Ensure they have the correct format:

* Public key (id\_rsa.pub):

sql

Copy code

ssh-rsa AAAAB3... user@host

* Private key (id\_rsa):

vbnet

Copy code

-----BEGIN OPENSSH PRIVATE KEY-----

...

-----END OPENSSH PRIVATE KEY-----

**2. Verify User and Path**

Make sure you are copying the key to the correct user and path on the remote server.

1. **On Jenkins server:**

sh

Copy code

ssh user@13.234.87.245

If you used root previously, ensure you are using the correct username (e.g., root).

1. **On the remote server:**

sh

Copy code

whoami

Confirm you are using the same user as you did when setting up the authorized keys.

**3. Append the Key Manually with Correct Permissions**

Make sure the key is correctly appended to authorized\_keys and permissions are set properly.

1. **On the remote server:**

sh

Copy code

mkdir -p ~/.ssh

chmod 700 ~/.ssh

nano ~/.ssh/authorized\_keys

Ensure the public key is correctly pasted in one line. Save and exit.

sh

Copy code

chmod 600 ~/.ssh/authorized\_keys

**4. Check SSHD Configuration**

Ensure the SSH daemon configuration on the remote server is correct.

1. **On the remote server:**

sh

Copy code

sudo nano /etc/ssh/sshd\_config

Ensure the following settings are enabled:

sh

Copy code

PubkeyAuthentication yes

AuthorizedKeysFile .ssh/authorized\_keys

Save and exit.

1. **Restart SSH Service:**

sh

Copy code

sudo systemctl restart sshd

**5. Debugging with Verbose Mode**

Use verbose mode to get detailed output about the SSH connection attempt:

1. **On Jenkins server:**

sh

Copy code

ssh -vvv -i ~/.ssh/id\_rsa user@13.234.87.245

Analyze the output for any clues on why the connection is being denied.

**6. Verify Key Permissions on Jenkins Server**

Ensure the private key on the Jenkins server has the correct permissions.

1. **On Jenkins server:**

sh

Copy code

chmod 600 ~/.ssh/id\_rsa

**7. Ensure Correct Ownership**

Ensure that the .ssh directory and the files within have the correct ownership on both the Jenkins server and the remote server.

1. **On Jenkins server:**

sh

Copy code

chown $(whoami):$(whoami) ~/.ssh/id\_rsa

chown $(whoami):$(whoami) ~/.ssh/id\_rsa.pub

1. **On remote server:**

sh

Copy code

chown $(whoami):$(whoami) ~/.ssh/authorized\_keys

**8. SSH Agent Forwarding (Optional)**

If you're using SSH agent forwarding, ensure it's set up correctly.

1. **On Jenkins server:**

sh

Copy code

ssh-add ~/.ssh/id\_rsa

ssh -A user@13.234.87.245

**Example Recap**

1. **Generate and verify SSH keys on Jenkins server:**

sh

Copy code

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

cat ~/.ssh/id\_rsa.pub

1. **On remote server:**

sh

Copy code

ssh another\_user@13.234.87.245

mkdir -p ~/.ssh

chmod 700 ~/.ssh

nano ~/.ssh/authorized\_keys

# Paste the public key

chmod 600 ~/.ssh/authorized\_keys

sudo nano /etc/ssh/sshd\_config

# Ensure PubkeyAuthentication yes and AuthorizedKeysFile .ssh/authorized\_keys

sudo systemctl restart sshd

1. **Test connection with verbose mode:**

sh

Copy code

ssh -vvv -i ~/.ssh/id\_rsa user@13.234.87.245

By meticulously following these advanced steps, you should be able to identify and resolve the Permission denied (publickey) error. The verbose mode output is particularly useful for diagnosing the exact point of failure in the SSH authentication process.