Table of Contents

SSH keys

Downloads & Links

PDF version
ssh_test.py

A password-less SSH key pair is needed if you want to use the QuantumATK Job Manager for executing and managing QuantumATK jobs on a remote computing cluster. This tutorial shows you how to

1. create a pair of public and private SSH keys;
2. add the public key to the list of authorized SSH keys on the remote cluster;
3. check that the password-less SSH connection works properly.

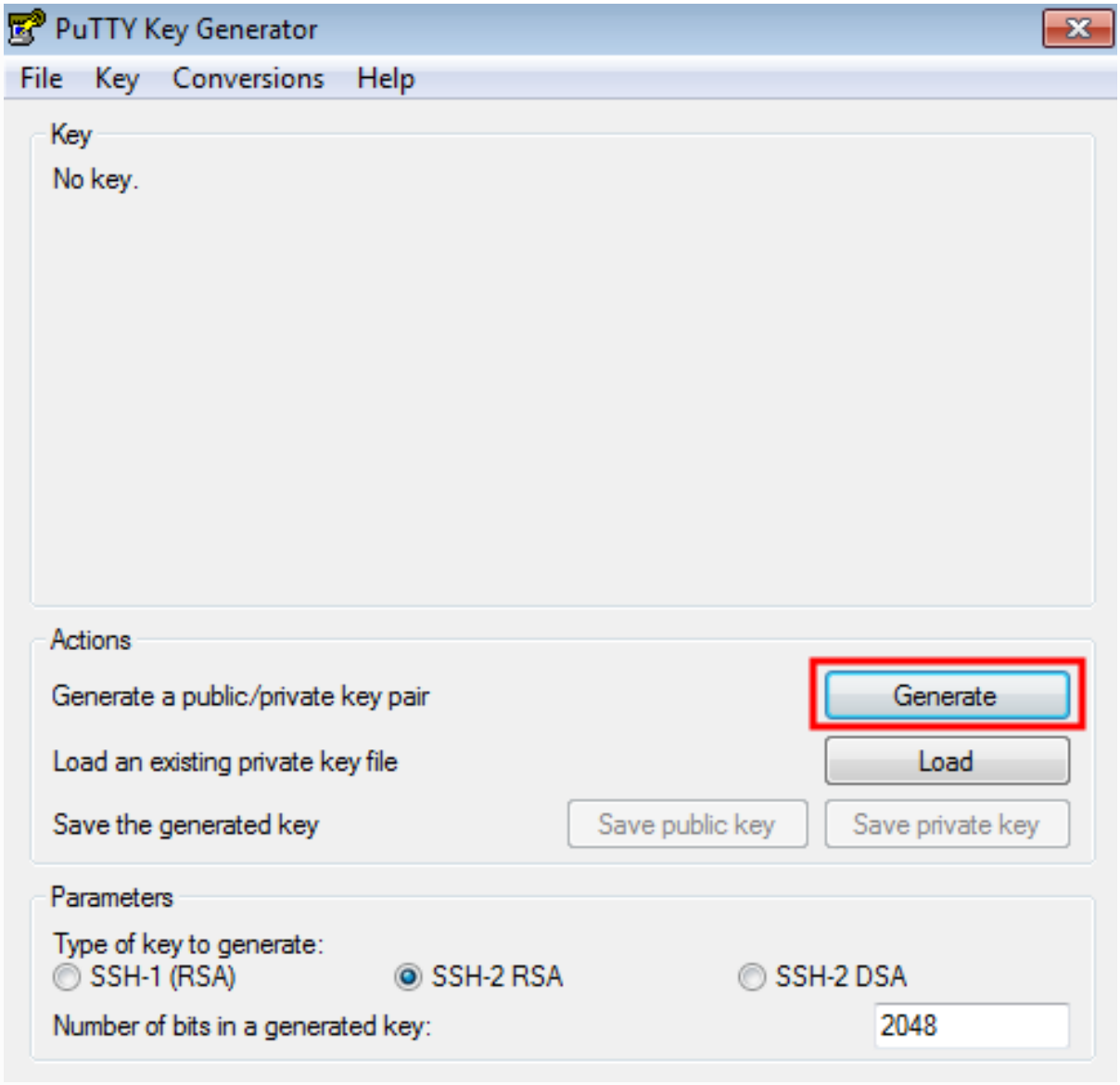
We provide detailed instructions for Windows, Linux, and OS X.

Create a pair of public and private SSH keys

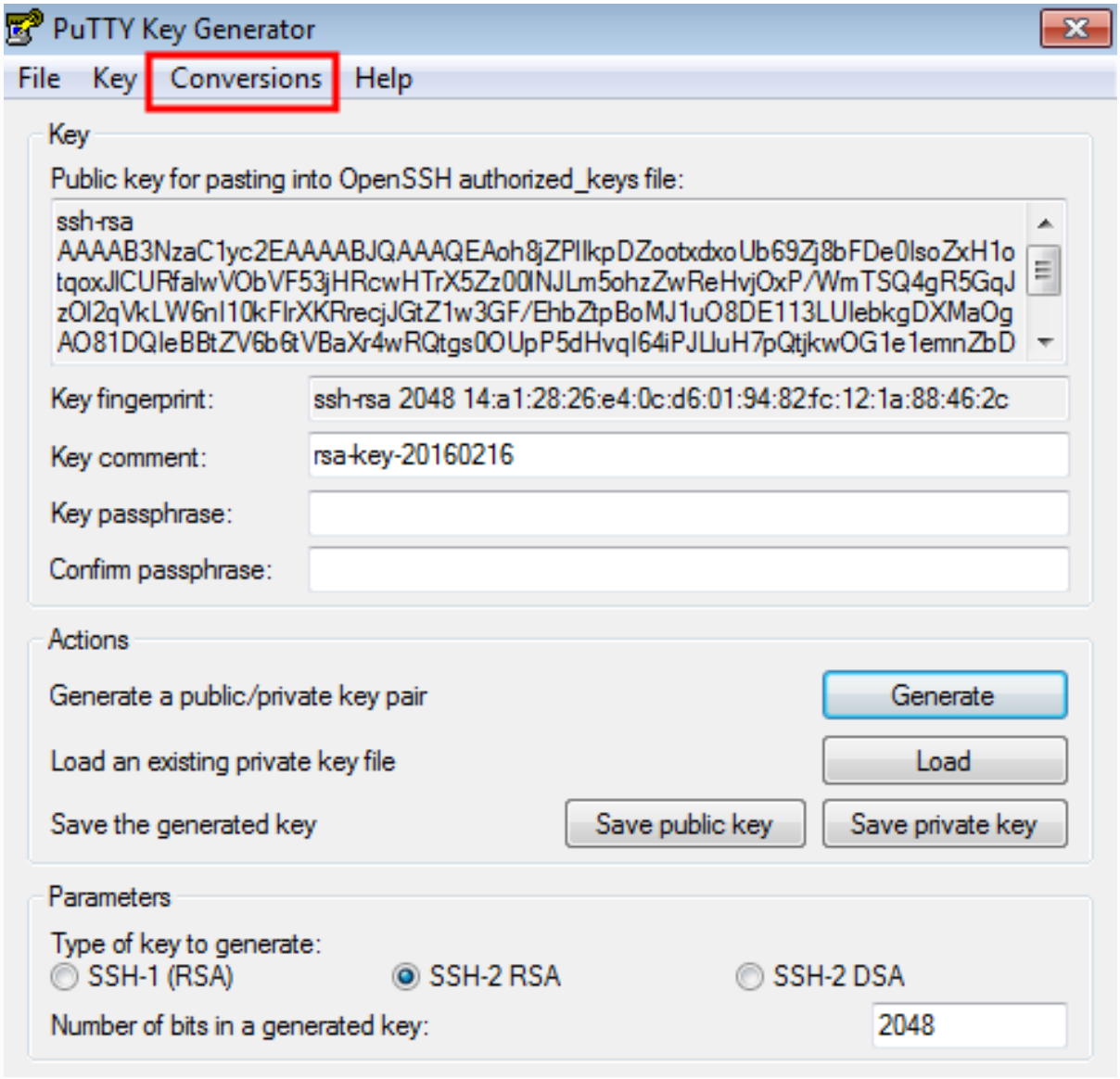
Windows

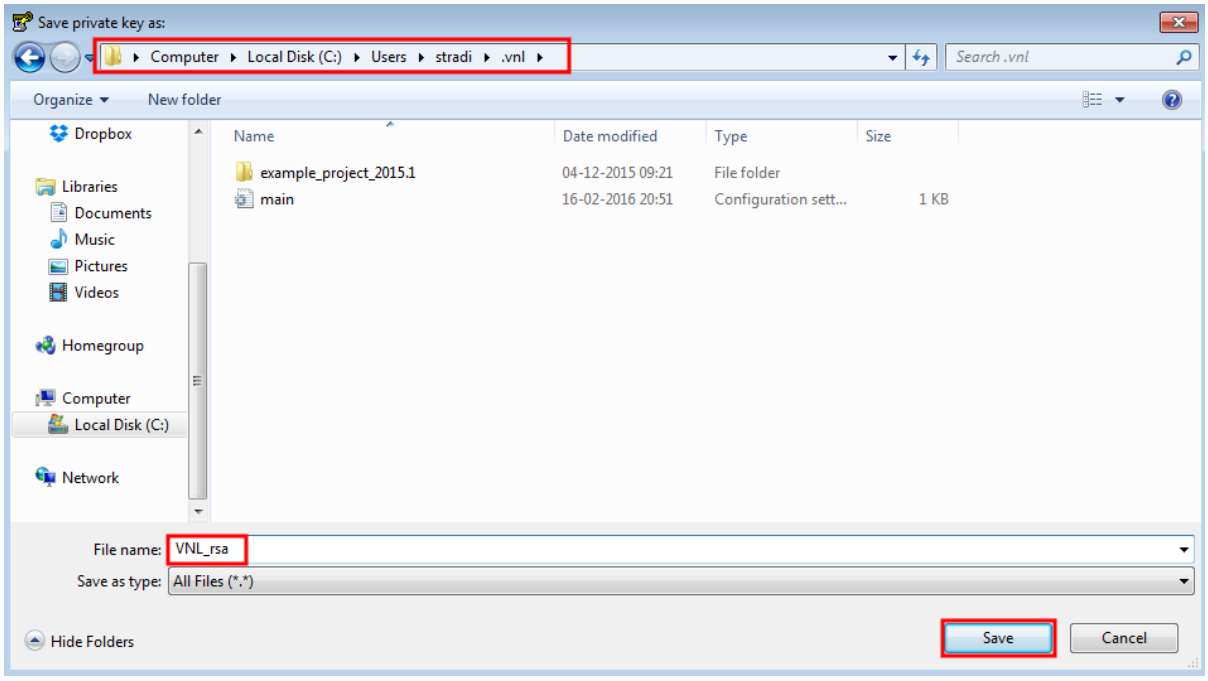
You need to install a bit of specialized software to generate SSH keys and establish SSH connections on Windows. PuTTY and PuTTYgen are popular choices. Download and install them both.

- Open **PuTTYgen** and click *Generate* to create a key pair. Do not use a key passphrase.

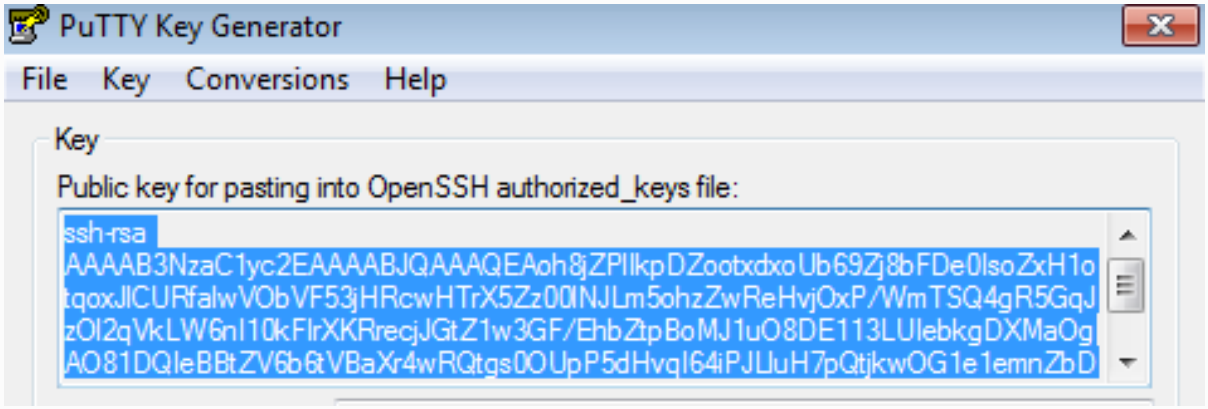


- Go to Conversions ▸ Export OpenSSH key to export and save the private key. Save it as `QuantumATK_rsa` in the folder `Computer/LocalDisk(C:)/Users/user/.quantumatk/` (adapt the path if it differs on your local machine. In particular, replace “user” with a proper *username*).





- Right-click the public key area and copy the text.



- Paste the public key into an editor (e.g. Notepad) and save it as `QuantumATK_rsa.pub` in the `.quantumatk/` folder.

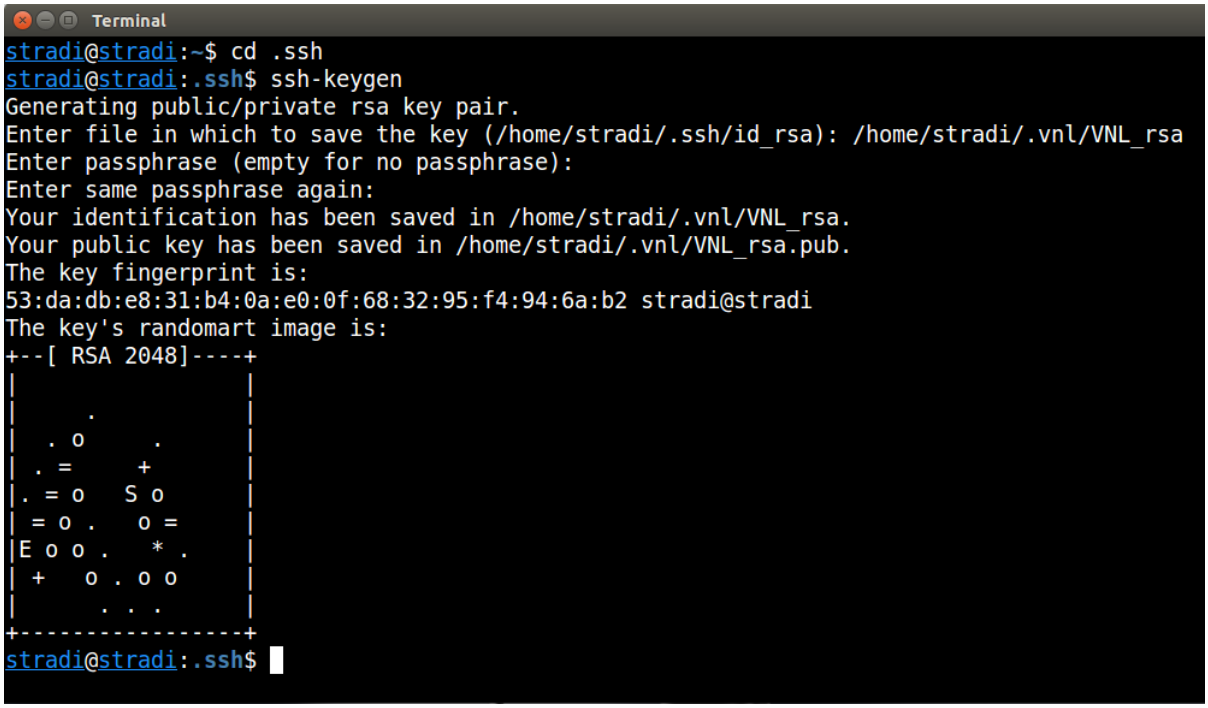
Linux and OS X

These operating systems should by default offer the software you need.

- Open a Terminal window, and navigate to the folder `/home/user/.ssh`. Replace “user” with a proper *username*.
- Check if a set of keys are already generated (`id_rsa` and `id_rsa.pub`).
  - If they are, you need to check if they are password-less: Check if the second line of `/home/user/.ssh/id_rsa` says “ENCRYPTED”.
  - If yes, the key pair is not password-less, and you need to generate a new pair.
  - If no, copy the key pair to the folder `/home/user/.quantumatk/` and give them the prefix “QuantumATK” (`/home/user/.quantumatk/QuantumATK_rsa` and `/home/user/.quantumatk/QuantumATK_rsa.pub`). Then proceed to the section Add the public key to authorized SSH keys on the remote cluster.
- Use the `ssh-keygen` command to generate a new SSH kay pair:

```
$ ssh-keygen
```

- Choose `/home/user/.quantumatk/QuantumATK_rsa` for the file name.
- Do not use a key passphrase! Simply hit enter when asked for the passphrase.
- You should now have public (`QuantumATK_rsa.pub`) and private (`QuantumATK_rsa`) SSH keys in the folder `/home/user/.quantumatk` .



Add the public key to authorized SSH keys on the remote cluster

Next, you need to append your public SSH key to the list of authorized keys on the remote cluster.

Note

You may need your cluster administrator to do this for you. Anyway, below we show how to do it in the case you have access to do it.

Linux and OS X

Use the `ssh-copy-id` command:

```
ssh-copy-id -i /home/user/.quantumatk/QuantumATK_rsa username@HOST.CLUSTER.EDU
```

and enter your password when prompted. Your public key should now be appended to the file `/home/user/.ssh/authorized_keys` on the cluster.

Windows

You need to log on to the cluster and manually add the key.

- Open **PuTTY** in your computer, enter the remote *hostname*, and click *Open*.

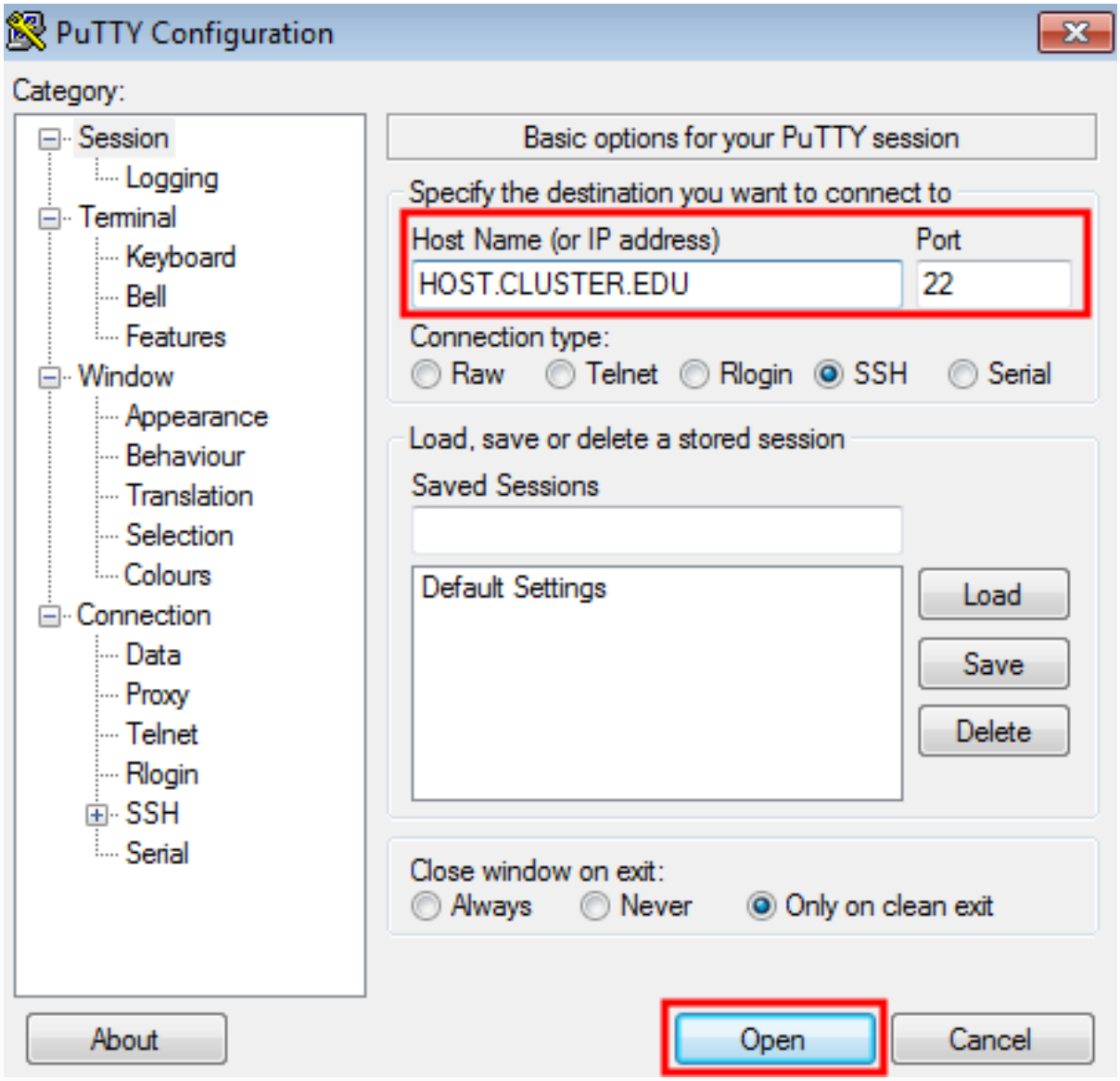


Fig. 32 Replace HOST.CLUSTER.EDU with the proper *hostname*.

- Append the contents of your public key (the one with `.pub` extension) to the file `/home/user/.ssh/authorized_keys` and save it.
- Exit the remote cluster and go to the next section to test the setup.

Test the password-less SSH connection

We here provide an QuantumATK script that attempts to establish a connection to the remote cluster, and reports back if the connection was successfully established.

- Download the script ssh_test.py and edit the connection settings in lines 7 to 9 (*key_dir*, *hostname*, and *username*).

```
from NL.ComputerScienceUtilities.SSHConnection import SSHConnection
import os

# ------------------------------------ #
# Edit only these 3 settings
# ------------------------------------ #
key_dir  = 'path_to_SSH_keys'
hostname = 'HOST.CLUSTER.EDU'
username = 'my_user_name'

# ------------------------------------ #
port = 22
ok = os.path.isdir(key_dir)
if ok:
    print("Sucessfully found local dir with SSH keys.")
    con = SSHConnection(hostname, port, username, key_dir)
    con.connect()
    ok = con.isConnected()

    if ok:
        print("Connection succesful.")
else:
    print("Error: Could not find local dir with SSH keys.")
# ------------------------------------ #
```

- Execute it from command line:

```
$ atkpython ssh_test.py
```

- If the test passes, the lines following lines are printed:

```
Successfully found local dir with SSH keys.
Connection successful.
```

Tip

You should now be ready to use the QuantumATK Job Manager for running your QuantumATK jobs on the remote cluster, see the tutorial Job Manager for remote execution of QuantumATK scripts.

Previous Next