Setting Up Public Key & SSH Passthrough

Follow the steps below to set up public key authentication.

Before You Begin: We highly recommend that you create a ~/.ssh/config file on your local system as described in Setting Up SSH Passthrough before you proceed with these steps.

Step 1: Create an SSH Public/Private Key Pair

To use public key authentication with the Secure Front Ends (SFEs), you need to have an SSH public/private key pair. If you do not, you can create a SSH public/private key pair by typing the following command and following the prompts:

your_local_system% ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/*username*/.ssh/id_rsa): **ENTER**
Enter passphrase (empty for no passphrase):**enter a passphrase**
Enter same passphrase again: **type your passphrase again**
Your identification has been saved in /Users/*username*/.ssh/id_rsa.
Your public key has been saved in /Users/*username*/.ssh/id_rsa.pub.

Your passphrase for the private key must meet NASA password requirements.

If you are using a non-Unix system, please consult your SSH documentation.

Step 2: Initialize the SSH Public Key on the SFEs

Copy the public key file to your ~/.ssh2 directory on each SFE you want to use, as follows (where *sfeX* represents sfe[*6-8*]):

your_local_system% ssh username@sfeX.nas.nasa.gov ssh-key-init --add < ~/.ssh/id_rsa.pub

Windows users: The above command with the redirection (i.e., < ~/.ssh/id_rsa.pub) will hang on Windows systems. You can try the following alternative method:

1. On your local Windows system, copy the contents of the ~/.ssh/id_rsa.pub file onto the clipboard (e.g., using the notepad program on Windows).
2. From your Windows system, ssh into an SFE: *your_local_system*% ssh sfe
3. From the SFE, run the command: sfe% ssh-key-init --add
4. Paste the contents of the clipboard, then type Ctl-D to complete.

Note: Public keys with embedded forced commands or other constraints will be ignored.

To view your existing public keys on sfe*X*:

ssh *username*@sfeX.nas.nasa.gov ssh-key-init --get

To completely replace all existing public keys on sfe*X*:

ssh *username*@sfeX.nas.nasa.gov ssh-key-init --set < ~/.ssh/id_rsa.pub

To delete one or more public keys on sfe*X* that contain the string denoted by STRING:

ssh *username*@sfeX.nas.nasa.gov ssh-key-init --remove=STRING

For example, the following would remove the public key(s) containing AAAAB3NzaC1yc2EAAAADAQABAAABAQCw2qCC:

ssh-key-init --remove=AAAAB3NzaC1yc2EAAAADAQABAAABAQCw2qCC

Notes:

- If you don't specify the full key string, you must ensure that any shorter string you use does not match other keys you do not want to remove.
- The *username@* is not required when your local username matches your NAS username, or if you have already configured your username in the configuration file (~/.ssh/config). You can download a NAS template for the configuration file here: Setting Up SSH Passthrough

You can also invoke the ssh-key-init command when you are already logged into an SFE. In this case, omit the ssh *username*@sfe*X*.nas.nasa.gov portion and begin the command line with ssh-key-init. Note that when you include the --add and --set options directly on an SFE, you must paste the keys into the terminal window and then terminate by typing Control-d. Keep in mind that public keys are not currently synchronized between SFEs, so each ssh-key-init invocation is only relevant to the particular sfe*X* it is run on.

To test your public/private key pair:

your_local_system% ssh -i ~/.ssh/id_rsa *username@sfeX*.nas.nasa.gov

You will be prompted for both your SecurID passcode and your private key passphrase.

TIP: Complete these steps for all four SFEs so you can set up SSH passthrough on each one. This enables you to easily switch from one to another if necessary. For example, if you have any issues connecting to one, you can try connecting to a different one.

NAS supercomputers are served by several front-end systems within the secure enclave. These include:

- **Athena:** athfe[*01-04*]
- **Cabeus:** cfe[*01-02*], cghfe[*01-02*]
- **Aitken Milan:** afe[*01-02*]
- **Aitken Cascade Lake & Rome; Electra; Endeavour:** pfe[*20-22,25,26*]
- **Lou:** lfe[*5-8*]

By investing some time to set up SSH passthrough, you can make your future NAS logins and inbound file transfers to these systems easier and faster. The SSH passthrough feature allows you to log into any of the front-end systems by typing just one SSH command.

- ***Without*** setting up SSH passthrough, you must first log into a secure front end (SFE), and then log into a front-end system.
- ***With*** SSH passthrough, you can "pass through" an SFE directly to a front-end system in the secure enclave, where you will do most of your work.

In other words, if you set up SSH passthrough, you can SSH directly from your local system to a NAS system as if you were directly connected, even though all connections are transparently passing through the SFEs.

SSH Connection Timeouts: SSH connections to front ends inside the enclave are disconnected after 15 minutes of inactivity. For more information, including best practices for minimizing the impact to your interactive sessions, see Using the Screen or tmux Commands.

How To Set Up SSH Passthrough

At this point, you are just three steps away from streamlining all of your future logins and inbound file transfers. With SSH passthrough, you will be able to log in quickly to any host in the secure enclave, and you will be able to copy files from your local system to any NAS host.

Before You Begin: If you have not already done so, complete the steps for first-time login to NAS systems.

Step 1: Create/Modify the .ssh/config File on Your Local System

You must have a ~/.ssh/config file on your local system to enable SSH passthrough. The file should include entries for the hosts you want to access inside the NAS enclave.

TIP: Although you can create a customized version of the ~/.ssh/config file, we recommend downloading and using the **ssh_config.txt** template attached at the end of this page.

This template should work for Mac, Linux, or Windows systems; for Windows, it will work only through the use of Windows Subsystem for Linux (WSL) or MobaXterm. If you are not using WSL or MobaXterm, you will need to modify this template by replacing ssh with ssh.exe in all of the ProxyCommand lines.

Move the file to the .ssh directory on your local system, and uncomment and replace <nas_login_name> with your NAS username. For example, if your username is zsmith, replace <nas_login_name> with zsmith.

If you already have an existing file named config, rename it to a different name. Then, *choose one* of the following two approaches to use this template:

- Create a new file called config which contains just this line: Include ~/.ssh/ssh_config.txt.
- Rename the file ssh_config.txt to config.

The contents of the current **ssh_config.txt** template are shown below. In this template, sfe6 is used; in the event that sfe6 is unavailable, or if you want to use a different SFE for SSH passthrough, you can switch to sfe[*7-8*].

```
#Updated 20260218

Host sfe
   HostName              sfe6.nas.nasa.gov

Host sfe sfe?.nas.nasa.gov
   # Uncomment to allow multiplexing of single connections
   #ControlMaster         auto
   #ControlPersist        1
   #ControlPath           ~/.ssh/master-%r@%h:%p
   ForwardX11             yes
   ServerAliveInterval    5m

Host sfe sfe?.nas.nasa.gov sup*.nas.nasa.gov
   LogLevel               info
   ProxyCommand           none

Host pfe pfe-last pfe.nas.nasa.gov pfe-last.nas.nasa.gov
   ProxyCommand           ssh -ax -oCompression=no sfe ssh-balance %h

Host lfe lfe-last lfe.nas.nasa.gov lfe-last.nas.nasa.gov
```

```
    ProxyCommand          ssh -ax -oCompression=no sfe ssh-balance %h

Host lou lou-last lou.nas.nasa.gov lou-last.nas.nasa.gov
    ProxyCommand          ssh -ax -oCompression=no sfe ssh-balance %h

# Add additional hosts to the list below as needed
Host *.nas.nasa.gov lou lou? ?fe *fe* *-last pbspl?
    HostbasedAuthentication no
    ProxyCommand          ssh -ax -oCompression=no sfe ssh-proxy %h
    ServerAliveInterval    5m
    # Uncomment the line below to enable X11 Forwarding
    # ForwardX11            yes
    # Uncomment out EnableEscapeCommandLine yes if you use OpenSSH 9.2
    # or newer and you are not able to escape into the SSH subshell/tunnel
    # with the use of SHIFT-<TILDA>-C keys when setting up VNC
    # EnableEscapeCommandLine    yes
    # Uncomment and replace <NAS_login_name> with your NAS username
    # if different than local username
    # User                <NAS_login_name>
```

WARNING: Your .ssh/config file should be set with no group/others write permission. Otherwise, you will get this error message when you connect:
Bad owner or permissions on /u/*your_local_username*/.ssh/config.

Once you have created your config file, test to confirm it's working by logging into a PFE as follows:

your_local_system% ssh pfe

If the file is configured properly, you will be prompted for your NAS password once, and for your RSA SecurID passcode twice.

You can continue to streamline your logins and file transfers further by completing steps 2 and 3. You will then be able to log in with just your RSA passcode.

Step 2: Create and Copy OpenSSH Public Key to Hosts Inside the Enclave

Before You Begin: Follow the steps in Setting Up Public Key Authentication to create an SSH public/private key pair and to initialize the SSH public key on the SFEs.

Then, complete these steps to copy your public key to the hosts inside the enclave and place the key in your .ssh/authorized_keys file. The examples in these steps use a PFE.

Note: This must be done for both a PFE and an LFE inside the enclave to which you want to connect using SSH passthrough.

a. Copy Your OpenSSH Public Key

   Ensure that you have a .ssh directory on the NAS host (in this example, a PFE) before issuing the scp command below. Otherwise, the command will copy the file id_rsa.pub to the PFE with the filename ".ssh." To create the directory, log into the PFE and run:

   pfe% mkdir .ssh

   On your local system, run:

   your_local_system% scp ~/.ssh/id_rsa.pub pfe:.ssh

b. Add Your OpenSSH Public Key to Your .ssh/authorized_keys File on the Enclave Host

   On your local system or on an SFE, run:

   your_local_system or *sfeX*% ssh pfe

   On the PFE, run:

   pfe% cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

   If you get the error message "/u/*username*/.ssh/authorized_keys: No such file or directory " after issuing the above command, you likely have set noclobber, which prevents you from overwriting files. You can use the command unset noclobber first to avoid this problem.

   WARNING: The permission for the authorized_keys file must be set to 600. To set this permission, run the command chmod 600 authorized_keys. Group/others write permissions on /u/*username* and /u/*username*/.ssh are not allowed for public key authentication.

   Repeat this step for the other NAS hosts you plan to use, such as an LFE.

   Note: If your local ~/.ssh/config file is not based on the NAS **ssh_config.txt** template, you may need to change the scp and ssh commands. To change the scp command:

   your_local_system% scp -oProxyCommand='ssh *username*@sfeX.nas.nasa.gov
   ssh-proxy %h' ~/.ssh/id_rsa.pub *username*@pfe:.ssh

where *sfeX* is sfe[*6-9*].

Note: Because of a formatting issue, this command is shown broken into two lines. It should be on only one line (with a space before ssh-proxy.).

To change the ssh command:

sfe*X*% ssh *username*@pfe

Substitute your NAS username for *username*. (If your local host username and your NAS username are the same, you can omit *username@* from the command line.)

Step 3: Set Up SSH Agent

The ssh-agent program holds and manages the private key on your local system and responds to key challenges from remote hosts. The private key is not initially stored in the agent and is added through the ssh-add program.

Typically, ssh-agent is started at the beginning of an X session or a login session, and you provide your passphrase to unlock your private key for this originating session. For any SSH connection to a remote host (for example, sfe*X*) made from this original session, the ssh-agent remembers your private key and will respond to challenges automatically without prompting you to type in your passphrase again.

Run one of the following command lines to launch ssh-agent:

- **For csh or tcsh:**

  your_local_system% eval `ssh-agent -c`

- **For sh or bash:**

  your_local_system% eval `ssh-agent -s`

  If the bash command line shown above results in error, try running this one instead:

  your_local_system% eval "$(ssh-agent -s)"

To add your private key to ssh-agent, run:

your_local_system% ssh-add *private_key*

Example:

your_local_system% ssh-add ~/.ssh/id_rsa
Enter passphrase for /Users/*username*/.ssh/id_rsa: **type your passphrase**
Identity added: /Users/*username*/.ssh/id_rsa (*username*@....)

Once you've completed these steps, you can now SSH from your local system to a NAS system inside the secure enclave, and be prompted for only your RSA SecurID passcode.

Note: If you get the following error when starting ssh-agent from a terminal on Windows 10, the most likely reason is that thessh-agent service is set to disabled state:

unable to start ssh-agent service, error :1058

To resolve this error, set the service to start manually in the Services GUI or run the following command in admin mode:

% Get-Service -Name ssh-agent | Set-Service -StartupType Manual

If you have set up public key authentication for one or more of the secure front ends (SFEs), you are ready to do this two-step connection from your local system to a system inside the secure enclave, such as a PFE or LFE.

Before You Begin: If you have not yet set up public key authentication, complete the steps in Setting Up Public Key Authentication.

Step 1: Accessing the SFEs

Access one of the SFEs from your local system by running SSH as follows:

your_local_system% ssh *username@sfeX*.nas.nasa.gov

where *username* is your NAS username and *sfeX* represents sfe[*6-8*].

Notes:

- You can omit *username@* from the command line if you have the same username for your local system and the NAS systems.
- If you have any issues connecting to one of the four SFEs, try logging into a different one.

After typing the command, you will be connected to the SFE and prompted for your private key passphrase followed by your RSA SecurID passcode.

Step 2: Accessing NAS Systems

Access a front-end system inside the enclave (for example, an LFE) from the SFE by running SSH as follows

sfeX% ssh lfe*X*

where *sfeX* represents sfe[*6-8*] and lfe*X* represents lfe[*5-8*]. You will be prompted for your RSA SecurID passcode.

If you want to avoid a two-step login for future sessions, you can set upSSH Passthrough.

SSH Connection Timeouts: SSH connections to front ends inside the enclave are disconnected after 15 minutes of inactivity. For more information, including best practices for minimizing the impact to your interactive sessions, see Using the Screen or tmux Commands.

One-Step Connection Using Public Key and Passthrough

This method requires that you first set up public key authentication and SSH passthrough. Once those steps are completed, you can connect from your local system to a system inside the NAS enclave with one single use of SSH, as follows:

your_local_system% ssh *your_username@nas_hostname*

where *your_username* is your NAS username, and *nas_hostname* is one of the hosts you listed in your ~/.ssh/config file—for example, pfe[*20-22,25,26*] or lfe[*5-8*].

Note: If you have the same username for your local system and the NAS systems, you can omit "*your_username*@" from the command line.

Example:

your_local_system% ssh pfe*25*
Enter PASSCODE: *type in your passcode*
pfe*25%*

SSH Connection Timeouts: SSH connections to front ends inside the enclave are disconnected after 15 minutes of inactivity. For more information, including best practices for minimizing the impact to your interactive sessions, see Using the Screen or tmux Commands.

You can use MobaXterm to connect from your local Windows system to NAS systems via SSH, as an alternative to Cygwin/OpenSSH. MobaXterm includes a built-in X11 client, and has the ability to forward X11 from NAS systems back to your local system. You can also use it to create VNC sessions.

Note: MobaXterm is only useful if you are using SSH to connect to NAS systems. It cannot be used for remote transfers utilizing the Secure Unattended Proxy (SUP) with Shift or other commands.

Downloading and Installing MobaXterm

MobaXterm Home Edition is available for free, and should be sufficient for most users. If you have admin privileges on your local system, you can install a full version (Installer Edition) with Start menu support. If a full software installation is not possible, you can use the standalone version (Portable Edition).

A paid version (MobaXterm Professional) is also available.

You can download them here: https://mobaxterm.mobatek.net/download.html

Connecting to NAS Systems

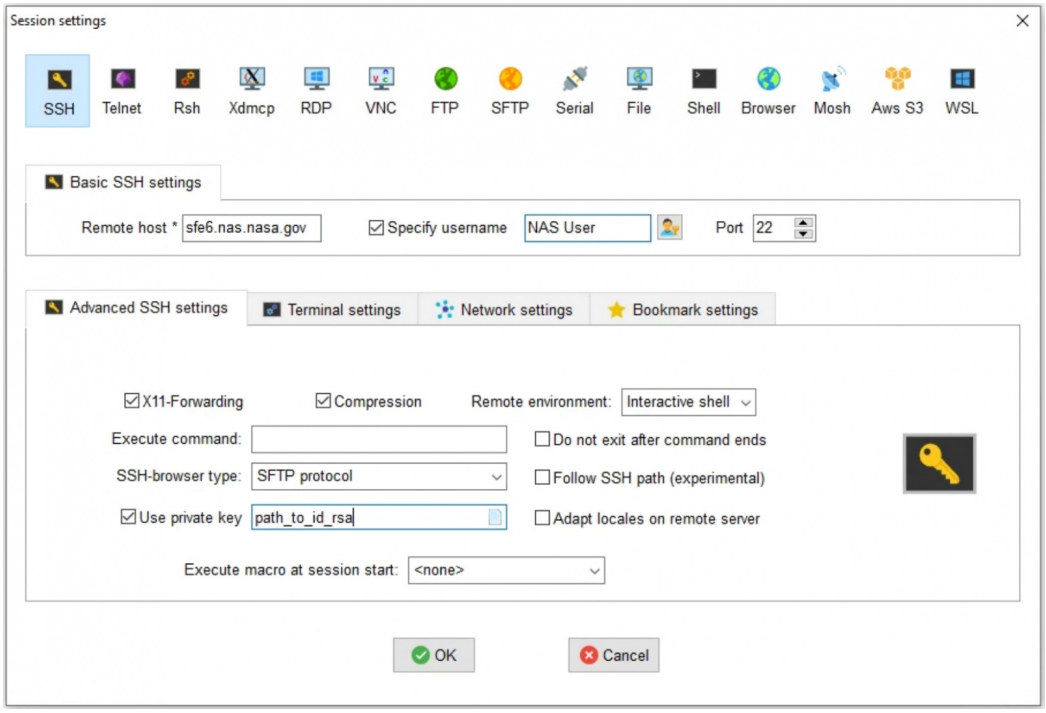
If you do not have SSH Passthrough set up, use Method 1 to first connect to a secure front end (SFE), then log into a PFE or LFE.

If you have SSH Passthrough set up, you can connect directly to a PFE or LFE using Method 2.

Method 1: Creating a New SSH Session to Connect to an SFE

Complete these steps to create a new SSH session to connect to a secure front end (SFE).

1. Click the **Session** icon in the top left corner of the initial window (or use **Ctrl-Shift-N**).
2. In the **Session settings** window (shown below), select the SSH tab by clicking the **SSH** icon in the top menu.
3. In the **Remote host** field, enter the full hostname of an SFE. In the **Specify username** field, enter your NAS username.
4. Click **OK**, and enter your NAS password and your RSA SecurID passcode when prompted.
   Option: If you have set up public key authentication between your local Windows system and the SFEs, before you click **OK** you can provide the path to your public key file (id_rsa) in the **Advanced SSH settings** tab, as shown below. You will be prompted for only the RSA SecurID passcode.



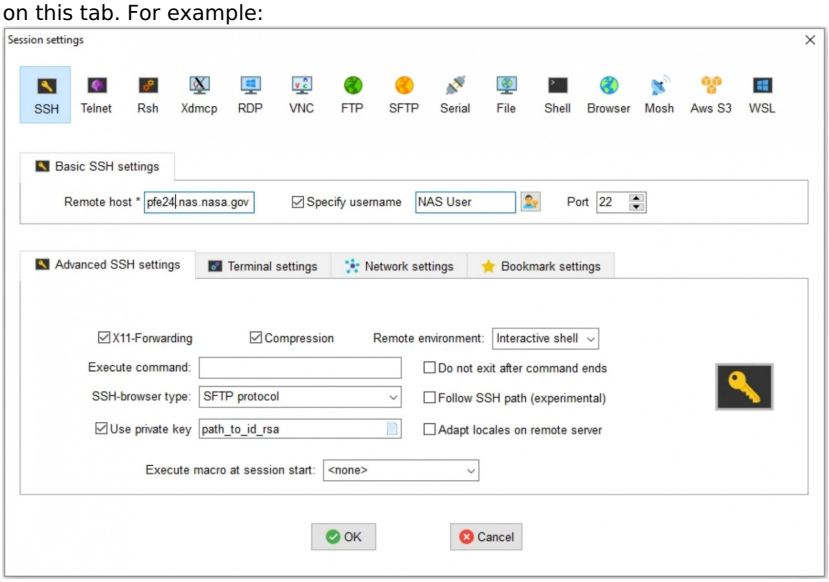
Once you are connected to the SFE, you can use the Linux command line to ssh into a PFE or LFE. Enter your NAS password when prompted.

Note: After you change your NAS password, you might have to remove your old password stored in MobaXterm before you can enter the new password when prompted. To remove your old NAS password, go to the the **MobaXterm Settings** dropdown menu and select **Configuration;** then select **MobaXterm passwords management;** then select the password that was changed, and click **Delete**.

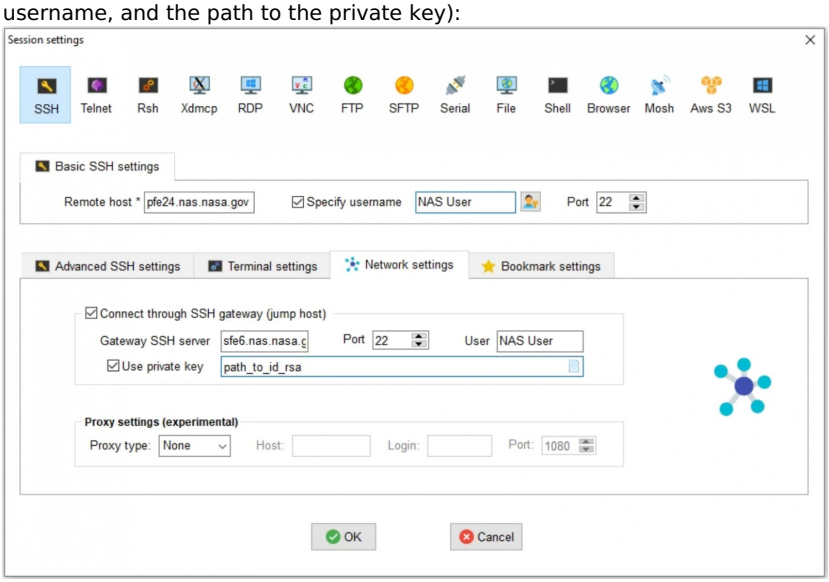
Method 2: Creating a New SSH Session to Connect Directly to a PFE or an LFE

If you have SSH Passthrough set up, complete these steps to connect directly from your Windows system to a PFE or LFE:

1. Click the **Session** icon in the top left corner of the MobaXterm client window (or use **Ctrl-Shift-N**).
2. In the **Session settings** window, click the **SSH** icon in the top menu to select the **SSH** tab.
3. In the **Remote host** field, enter the full hostname of a PFE or LFE.
4. Point MobaXterm to your public key file (rsa_id) in two places:
   - On the **Advanced SSH settings** tab, enter the path to your public key. If you like, you can also enable X11 forwarding on this tab. For example:

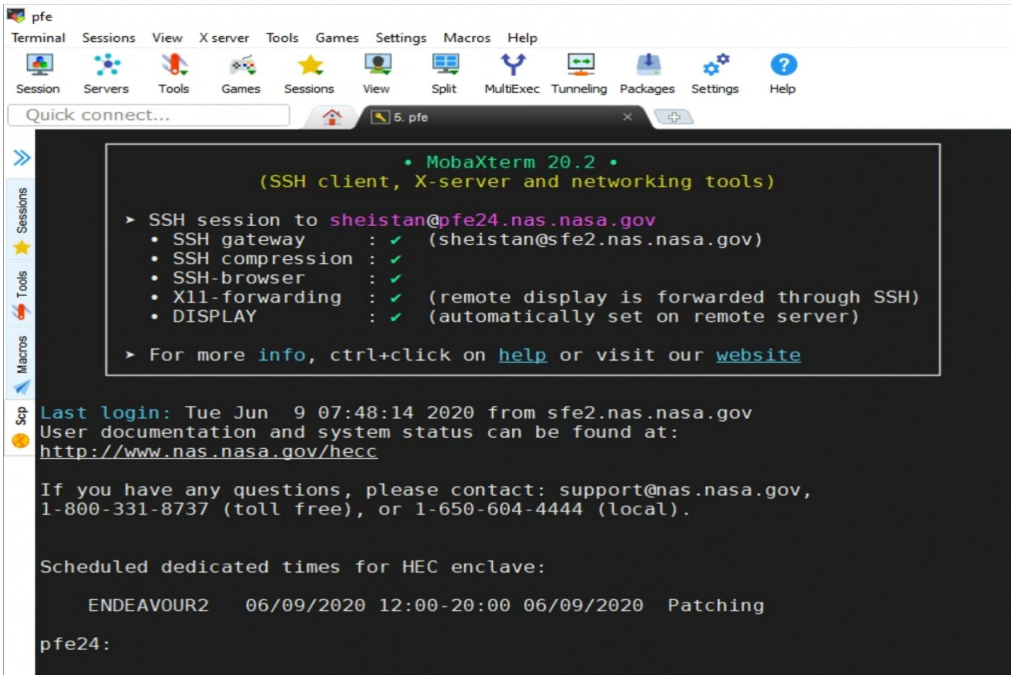


   - On the **Network settings** tab, enter the information used to go through the SFE (the gateway SSH server, your NAS username, and the path to the private key):



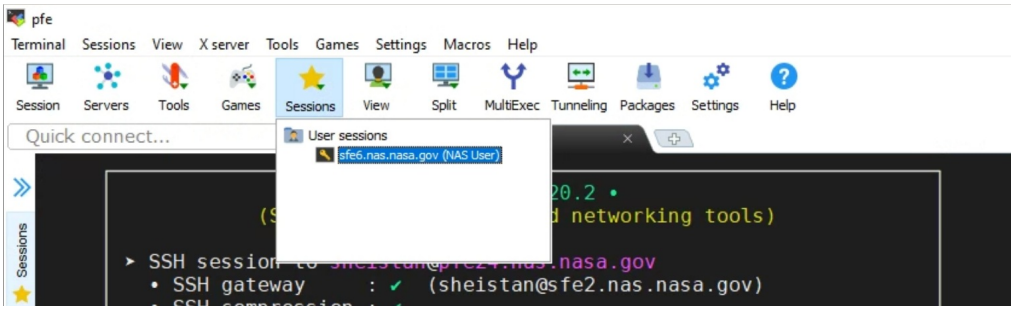
5. (Optional) In the **Bookmark settings** tab, name this session for future use.
6. Click **OK** to start the login process.
   Note: From now on, the session will be displayed in the **Recently used** section when you start MobaXterm, and you can start the login process by double-clicking it there.
7. You will be prompted to enter your SSH key passphrase (if set), and then your RSA SecurID passcode.

MobaXterm will log into the NAS system that you set in Step 3 (in this example, pfe24, which has since been decommissioned; please use one of pfe[20-22,25,26] in your own session):

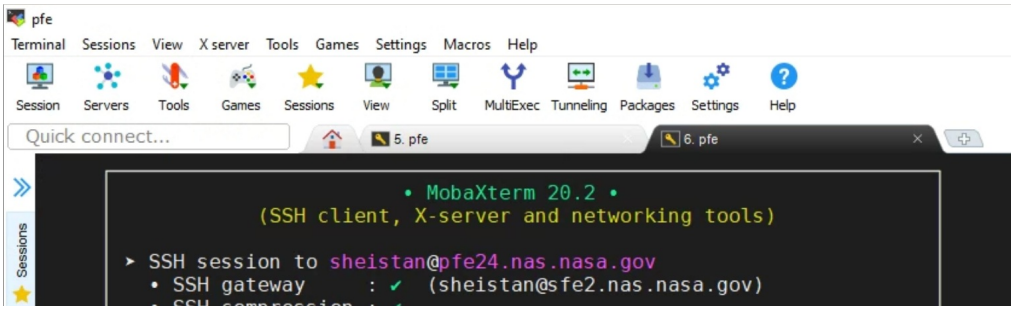


Duplicating a Login Session

If the original login session is still running, you can duplicate it to create new login sessions. Click the **Sessions** section, and select the session you want to duplicate:



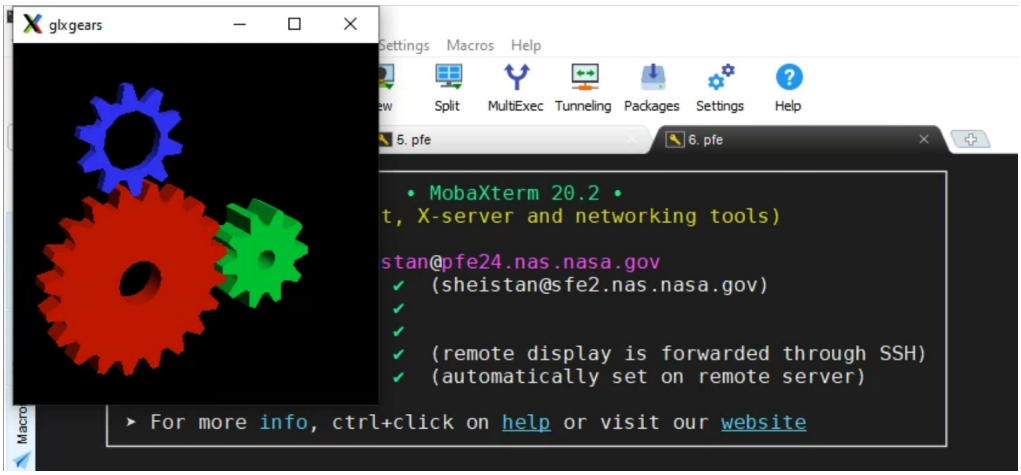
A new xterm tab will be created and logged in. No passphrase or passcode will be needed:



Note: To connect to other NAS systems (such as a different PFE), you must start a new connection. You will be prompted for your passphrase and RSA SecurID passcode.

Running a Graphical Application with VNC

If you enabled X11-forwarding in the **Advanced SSH settings** tab when you initiated your SSH session, you can run X and OpenGL applications. For example:



However, we highly recommend using Virtual Network Computing (VNC) for graphical work on NAS back to your local machine, as it is generally faster than X11 forwarding.

Using an X11 alternative like VNC in MobaXterm requires two tunnels through the NAS SFEs. To run VNC, you will need to do the following:

- Start a VNC server on a PFE and get a port established for the connection (for example, 5901), as described in Step 2 of the VNC article (Run the vncserver Command on pfe24).
- Create the tunnels from your local workstation to the VNC server running on the PFE, as described in the next section.

Sample Setup for Using VNC on MobaXterm

In this example, forwarding is enabled between port 5901 of the local machine and port 5901 on pfe26, where a VNC server is running with a tunnel that passes through the SFE on port 6901.

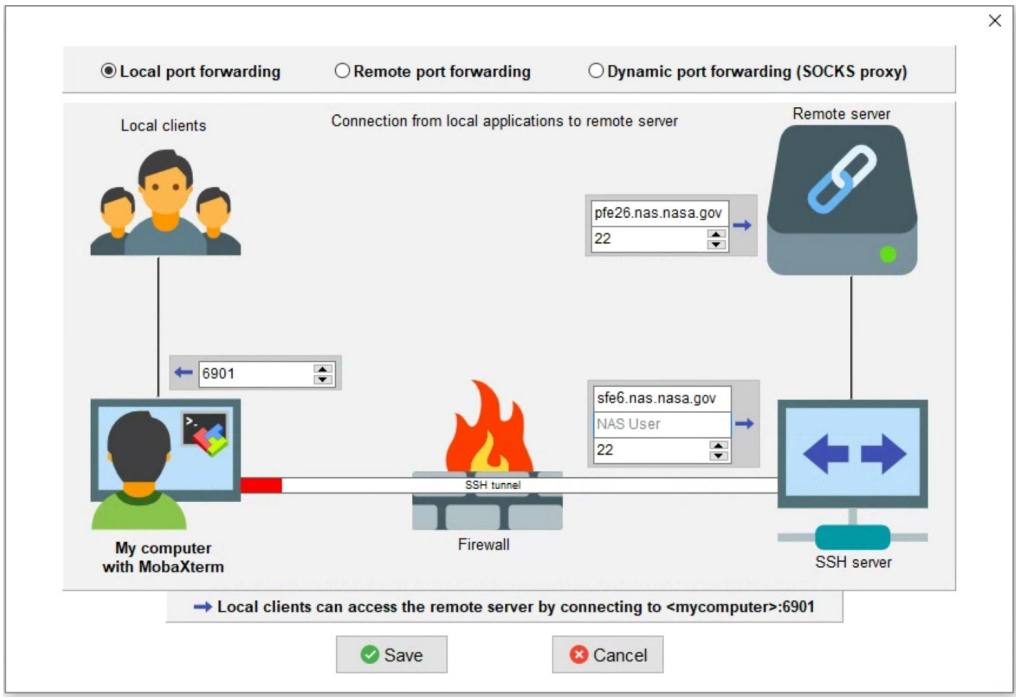
Note: These port numbers will be different in your case, depending on which ports VNC is listening on and which PFE you are running on.



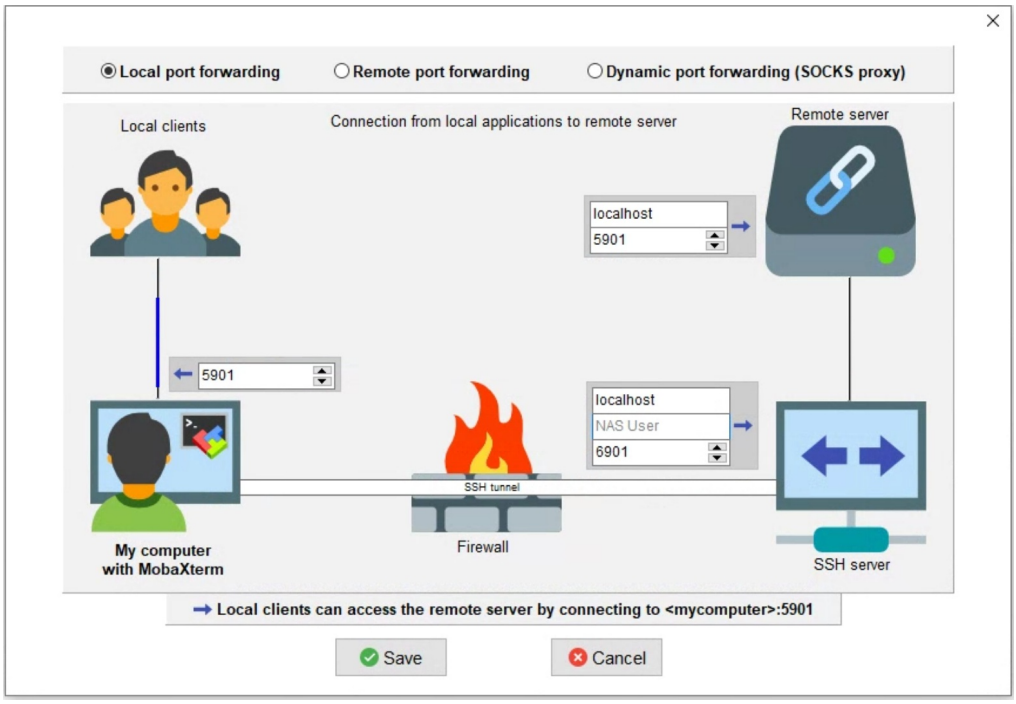
This example was created as follows:

In the main MobaXterm client window, click the **Tunneling** icon to open the MobaSSHTunnel window. Then, click **New SSH tunnel** and create **Tunnel 1** and **Tunnel 2**.

To create **Tunnel 1**:



To create **Tunnel 2**:



Once **Tunnel 1** is started, then start **Tunnel 2**. Both will prompt for the RSA SecurID passcode and your NAS password, as MobaXterm doesn't allow for SSH keys when tunneling.

Now, you can start a local VNC viewer on your local system, pointing at port 5901.