**SSH Key Generation (passwordless non-interactive SSH)**

**Examples for both Linux and Windows in both 7-Mode and Cluster-Mode**

**7-mode**

\*\* Using “vol0” as root volume name and “root” as the user for unix and “administrator” as the user for windows

**Cluster-mode**

\*\* Using “admin” as the username

**ONTAP**

**7-Mode (ssh setup by default ONTAP 8.x)**

ontap> **secureadmin setup ssh # go through prompts default**

ontap> **secureadmin enable ssh**

**Cluster-Mode**

Enable **Public/Private SSH Keys** for passwordless access

**node::> security login create -username admin -application ssh -authmethod publickey -profile admin**

**\*\*\*\* Linux \*\*\*\***

<https://kb.netapp.com/support/index?page=content&id=1010841>

Linux> **ssh-keygen -t rsa**

When asked for a 'passphrase', do not enter one. Just press Enter twice (or 3 times)

**7-Mode**

Linux> **ssh root@filer vol status** # it will ask for a password, but confirm ssh connectivity

Linux> **mount filer:/vol/vol0 /mnt/filer**

Linux> **mkdir -p /mnt/etc/sshd/.ssh**

Linux> **cat ~/.ssh/id\_rsa.pub >> /mnt/etc/sshd/root/.ssh/authorized\_keys**

Linux> **chmod 700 /mnt/etc/sshd/root/.ssh**

Linux> **chmod 600 /mnt/etc/sshd/root/.ssh/authorized\_keys**

Linux> **ssh root@filer vol status** # test command out passwordless

**Cluster-Mode**

Linux> **ssh admin@clustername security login show** # it will ask for a password, but confirm for ssh connectivity

Linux> **cat ~/.ssh/id\_rsa.pub**

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAxVJkDvfW7QuvORb+iKgRWUj3rbaeywtT2/aAzCH+aof7+wjhiBSsybQ7TtRo69m+2pWLGW1/wJEaLm/XPU6PRhgWeLiuv2wLnPKAp7h4b8zV85p201dsLIGtlzxTPcxExDnAScoGRYQc1lNYeRkD/nhhULSGc6Fjmx/Y0Zk1tyK+8YA8MrDleQ9xsYj9+L9dza5U69ygNXmDlYu8nrQWciH5GIbRTgHapt+H5ndlROBPzgZ240zhflbfNks9KrJa9VNFxFlTiOQTWofd860obLaIit3WuB55tg/nJfnWD0c9uK+Yvs7yT0tMRqFeAeL0RJC3nGW2/FNkTRM0ftzKEw== netapp@localhost.localdomain

**node::> security login publickey load-from-uri -username root -uri** [**file://localhost/mroot/id\_rsa.pub**](file:///\\localhost\mroot\id_rsa.pub) **or** [**http://ip/path/id\_rsa.pub**](http://ip/path/id_rsa.pub) **[-overwrite false]**

* + - **for** [**file://**](NULL) **scp the file to /mroot one node**
    - **OR - create key for user (copy/paste - using uri method can be easier)**

**node::> security login publickey create -username admin -index 1 -publickey "ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAxVJkDvfW7QuvORb+iKgRWUj3rbaeywtT2/aAzCH+aof7+wjhiBSsybQ7TtRo69m+2pWLGW1/wJEaLm/XPU6PRhgWeLiuv2wLnPKAp7h4b8zV85p201dsLIGtlzxTPcxExDnAScoGRYQc1lNYeRkD/nhhULSGc6Fjmx/Y0Zk1tyK+8YA8MrDleQ9xsYj9+L9dza5U69ygNXmDlYu8nrQWciH5GIbRTgHapt+H5ndlROBPzgZ240zhflbfNks9KrJa9VNFxFlTiOQTWofd860obLaIit3WuB55tg/nJfnWD0c9uK+Yvs7yT0tMRqFeAeL0RJC3nGW2/FNkTRM0ftzKEw== netapp@localhost.localdomain"**

* + Confirm user and key
    - **node::> security login publickey show -username admin**

Linux> **ssh admin@clustername security login show** # test command out passwordless

**\*\*\*\* Windows (using PuTTy) \*\*\*\***

<https://kb.netapp.com/support/index?page=content&id=1010841>

<https://kb.netapp.com/support/index?page=content&id=1011670>

**Generate keys for this, use puttygen.exe.**

* Open **puttygen.exe**.
* Select the **SSH-2RSA** option. (default)
* **Accept the default number of bits for the key size (1024).** The key size on the host does not have to match that of the storage system but it does have to be larger.
* Click **Generate**. You will be prompted to move the mouse in the key area.
* **DO NOT enter a passphrase when generating the keys.**
* Once the keys have been generated, save them to the plink.exe directory.
* Click “Save public key” **rsa\_pub\_clientplink\_key**
* Click “Save private key” **rsa\_priv\_plink\_key.ppk**

**Create an authorized\_keys file**. As a general rule, the authorized\_keys file does not take any line breaks. Therefore, do not edit this file with notepad, use **wordpad** or textpad.

The public key file generated by puttygen will look like the following:

Open **rsa\_pub\_clientplink\_key** with WordPad

---- BEGIN SSH2 PUBLIC KEY ----

Comment: "rsa-key-20121010"

AAAAB3NzaC1yc2EAAAABJQAAAIEAw1n6SjBBzDeferczSOYVblHq9aQlV7nq+zy/

Zao7ZXK9M34Qg08u65eep1KxzrUMzwRnbYAPoClb4BHhwz0HnsJSF9MkEP38VeOr

Gn/BZIayxc1GYE0FHfOiwFYAQJkZ1yz6h5VF0TXkefl8Ebo4AyjC5oCtNd2oncPX

Gdo1HF8=

---- END SSH2 PUBLIC KEY ----

Strip all line breaks and extra text from this file. – go to each line and remove the break on EVERY line…must be exact or it won’t work

Note: After 'ssh-rsa' there should be a space and then the key. There should not be any line breaks.

ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAIEAw1n6SjBBzDeferczSOYVblHq9aQlV7nq+zy/Zao7ZXK9M34Qg08u65eep1KxzrUMzwRnbYAPoClb4BHhwz0HnsJSF9MkEP38VeOrGn/BZIayxc1GYE0FHfOiwFYAQJkZ1yz6h5VF0TXkefl8Ebo4AyjC5oCtNd2oncPXGdo1HF8=

Notice that ssh-rsa is prepended to the line. Save this file as **authorized\_keys** and do not overwrite the original public key.

**7-Mode**

**plink.exe root@filer vol status** # it will ask for a password, but confirm ssh connectivity

**Create the directory structure on the storage system:**

C:\Users\test> **mkdir \\filer\etc$\sshd\administrator\.ssh**

-- Note: An error may be generated if this path already exists. This can be safely ignored.

Append the C:\plink\authorized\_keys file to the Storage Controller's \\filer\etc$\sshd\<username>\.ssh\authorized\_keys file:

C:\Users\test> **type** **C:\plink\authorized\_keys >> \\filer\etc$\sshd\<username>\.ssh\authorized\_keys**

Run the following command to add the private key to pageant:

**pageant C:\plink\rsa\_priv\_plink\_key.ppk**

where, rsa\_priv\_plink\_key.ppk is the file name for the private key file generated earlier assuming that the key is in the C:\plink Directory.

Run the plink.exe file on the host.

c:\plink>**plink.exe -v -i c:\plink\rsa\_ri\_plink\_key.ppk user@filer vol status** # test command out passwordless

The -v flag above is used to give a verbose output regarding the connection negotiation. It is a useful flag when trying to troubleshoot the connection.

**plink.exe root@filer vol status**

**Cluster-Mode**

**plink.exe admin@clustername security login show** # it will ask for a password, but confirm for ssh connectivity

**node::> security login publickey load-from-uri -username root -uri** [**file://localhost/mroot/id\_rsa.pub**](file:///\\localhost\mroot\id_rsa.pub) **or** [**http://ip/path/id\_rsa.pub**](http://ip/path/id_rsa.pub) **[-overwrite false]**

* + - **for** [**file://**](NULL) **scp the file to /mroot one node**
    - **OR - create key for user (copy/paste - using uri method is easier)**

**node::> security login publickey create -username admin -index 1 -publickey "** **ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAIEAw1n6SjBBzDeferczSOYVblHq9aQlV7nq+zy/Zao7ZXK9M34Qg08u65eep1KxzrUMzwRnbYAPoClb4BHhwz0HnsJSF9MkEP38VeOrGn/BZIayxc1GYE0FHfOiwFYAQJkZ1yz6h5VF0TXkefl8Ebo4AyjC5oCtNd2oncPXGdo1HF8="**

* + Confirm user and key
    - **node::> security login publickey show -username admin**

**plink.exe admin@clustername security login show** # test command passwordless