SHELL SCRIPTING

Shell Scripting is an open-source computer program designed to be run by the Unix/Linux shell. Shell Scripting is a program to write a series of commands for the shell to execute. It can combine lengthy and repetitive sequences of commands into a single and simple script that can be stored and executed anytime, which reduces programming efforts.

In this project, we would onboard 20 new Linux users onto a server using a shell script.

The script will read the file containing the names of the users, create each user on the server, and add them to an existing group called developers. This script will first check for the existence of the user on the system, before it will attempt to create that user. This script will also ensure that each user that is being created also has a default home folder.

Each user will have a *.ssh* folder within its HOME directory and for each user’s SSH configuration, we will create an *authorized\_keys* file and add a public key for connecting to the server.

Prerequisites

* Ubuntu Server 20.04 LTS (this project is executed on a Microsoft Azure Ubuntu VM)
* Putty (for Windows OS)
* Puttygen, a key generator for Putty to convert private *.pem* SSH key to *.ppk*

Step 1 - Create the users list, .*ssh* folder and *authorized\_keys* file

Create the group ‘developers’

*$ sudo addgroup developers*

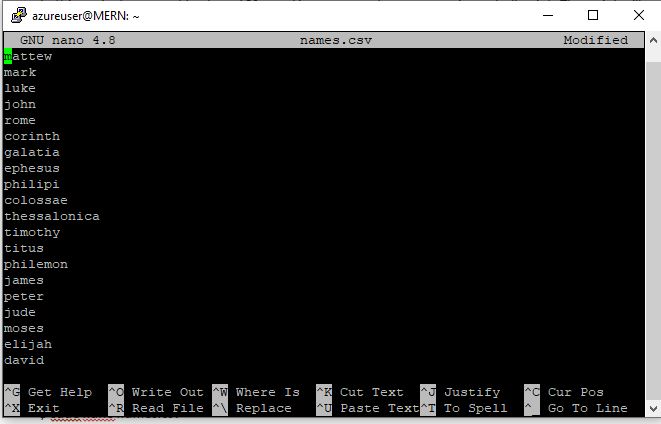


To view the list of groups in a server, run the command

*$ cat /etc/group*

Create a csv file *names.csv* with any text editor of choice and add twenty names in it, one on each line.

*$ sudo nano names.csv*



Next action is to create a folder (with any name of choice) in which a folder named *.ssh* will be created. In the *.ssh* folder, a file called *authorized\_keys* will be created. This file will contain the public key to be used by the 20 users to SSH into the Linux server.

Create a folder named *publickey* and cd into it

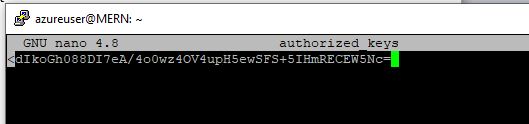
*$ sudo mkdir publickey && cd publickey*

Create a folder named *.ssh* and cd into it

*$ sudo mkdir .ssh && cd .ssh*

Create an *authorized\_keys* file and add the below public key in it

ssh-rsa 



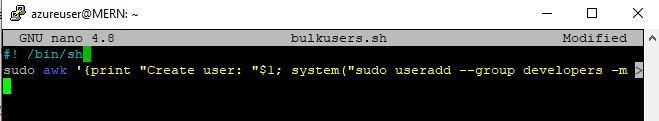
Step 2 – Create Shell Script

Go back to the home directory

*$ cd ../..*

Create and open a file named *bulkusers.sh* (with extension *.sh*). Start the script with *#! /bin/sh* then copy and paste the commands in the next line:

*sudo awk '{print "Create user: "$1; system("sudo useradd --group testgroup -m -k <file\_path\_to\_folder\_containing\_.ssh\_folder> "$1)}' $1*



Explanation

*#! /bin/sh* – This directs the script to the bourne-shell.

*awk* - this is the interpreter for the AWK Programming Language. The AWK language is useful for manipulation of data files, text retrieval and processing.

*print "Create user: "$1;* - This prints out an output showing that a user is being created, verifying that names have been processed.

*system* – This executes the command *sudo useradd* which creates the users.

*--group developers* adds the users to the already-existing *developers* group.

*-m* creates the home directory for each user if it does not exist.

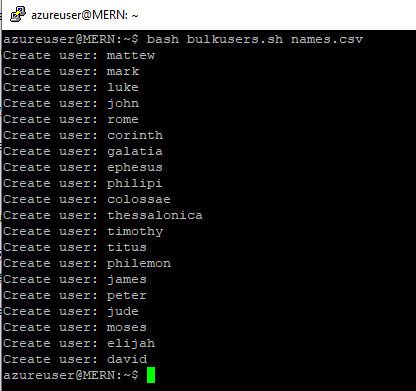
*-k* copies the contents of a folder into each user’s home directory.

Replace *<file\_path\_to\_folder\_containing\_.ssh\_folder>* with the location of the folder containing the *.ssh* folder i.e. */home/azureuser/publickey*

The *$1* at the end of the script refers to the argument that is to be passed into the script from the command line i.e. the file containing the names of the users.

To create the users, run the command

*$ bash bulkusers.sh names.csv*

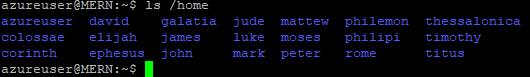


To confirm that the users were created, run the command

*$ nano /etc/passwd*

To confirm that the directories for each user were created, run the command

*$ ls /home*



Step 3 – SSH with a new user into the server

Below is the private key needed for the newly-created users to SSH into the server:

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

b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn

NhAAAAAwEAAQAAAYEAsymconB8SJJJUqzZcbCcsk63CHQSLfOGdHEG90eJNIx+d5MZGk+Y

T/LX1k3uti14u71fOn2Pkl+0wTTO+lxgm8P+r6zBnoJuidcdotQIizcKuZ6k4zhp9ACMJL

uCJQgiwAe7XX0clbgmq5gZcJ8t5c0csmyFAFl0fw+r6TdPTLAyC2ynYWsvkP6fTlATEiz6

l3WP0CIfznt61b9ZcGNwKITX6u1d7TvPMyrMHwOYonsTYkmF9Zsk9Dq1/HARO+9Y/arqDW

YUIGiyNZiDLQL5x0Aoa5/ZNp1GjvwTAQkBqJeV8xr9WnaO4YKFT6JbiFA/hRjaP6R+M3Jg

yyAh+na12PwvTBEp5DFDyxcQVyjvzorQKWO9b0WqPZwIQMKVTLMuhaojm+bmsIb4aB2CB0

1f3+BLWoFWoETD9j8WDGEeo8pCFeDDyU5ApvpRoU8jJUbvqkUI3S5/AALR3+GhFfHSJKBo

dPPAyO3gP+KNMM+DleLqR+XsEhUvuSB5kRAhFuTXAAAFgIuJ0uiLidLoAAAAB3NzaC1yc2

EAAAGBALMpnKJwfEiSSVKs2XGwnLJOtwh0Ei3zhnRxBvdHiTSMfneTGRpPmE/y19ZN7rYt

eLu9Xzp9j5JftME0zvpcYJvD/q+swZ6CbonXHaLUCIs3CrmepOM4afQAjCS7giUIIsAHu1

19HJW4JquYGXCfLeXNHLJshQBZdH8Pq+k3T0ywMgtsp2FrL5D+n05QExIs+pd1j9AiH857

etW/WXBjcCiE1+rtXe07zzMqzB8DmKJ7E2JJhfWbJPQ6tfxwETvvWP2q6g1mFCBosjWYgy

0C+cdAKGuf2TadRo78EwEJAaiXlfMa/Vp2juGChU+iW4hQP4UY2j+kfjNyYMsgIfp2tdj8

L0wRKeQxQ8sXEFco786K0CljvW9Fqj2cCEDClUyzLoWqI5vm5rCG+GgdggdNX9/gS1qBVq

BEw/Y/FgxhHqPKQhXgw8lOQKb6UaFPIyVG76pFCN0ufwAC0d/hoRXx0iSgaHTzwMjt4D/i

jTDPg5Xi6kfl7BIVL7kgeZEQIRbk1wAAAAMBAAEAAAGAPf8KOpOeDibAxKEXZWXt8y2V3J

D9sXTxc92gwXS5n7t2D76REy+zzwaDdZ7mGZhGjQCMsVq9kbMYgzrY3H2W2I/L09J99XHA

+mW71Zp1kmbriSvCdvYQg+SkmhlggZv9GmISjdk7SPu+Nead9wC+CyUc5wjyRRqvW0B7Bm

qjQDBAQP/KM8W5Yf0Z9ylyT/nMhRijOSx1wSeta8WZF3DxYLQHWz3kILFvk48dryW5bZAV

Nw+mEUUsVm7yhnXpIMpDdl7wlDlqAWcuEQKJ7WJ7swuZM/FTQW4rFMmpDO8Q8PgijqOFDQ

jl8XfCPCkOhI9JOFTbmImTxfbRZ/NYYF09cFcqhKyvEi/Egx2oUZq4M81EGpP+EZnWgZtG

/PHqrSqIW166fixe/47eGCSt+AlyeR8SZCA1jjMRf7WB1RjANUHgC59tNTMQiFg+T5c2Yj

ORmPT0PpzEtQ+WMSMI5hGoklmqXuS5iiyJx7HyLOnK7wNloj7oqboz91wcCYnYWCORAAAA

wQDUbuGf0dAtJ4Qr2vdHiIi4dHAlMQMMsw/12CmpuSoqeEIWHVpAEBpqzx67qDZ+AMpBDV

BU9KbXe7IIzzfwUvxl1WCycg/pJM0OMjyigvz4XziuSVmSuy10HNvECvpxI3Qx8iF/HgAP

eyYe369FHEBsNZ5M5KhZ4oHI/XgZB5OGOaxErJd3wXhGASHnsWcmIswIjat7hH9WlAeWAk

/aeMz92iSDnYBOr+gAycsBm/skEDrN7dD45ilSvLZ6DQ2hbKAAAADBAOhLy9Tiki1IM2Gg

ma8KkUiLrqqx8IexPd580n7KsL32U2iu6Y88+skC8pkZQmIVG2UQhjiVLpNBgrzKKDJciK

/lyen21npQjuYaJPUgVUG0sjMtTpgGwbN/IVyHO28KSOogB6MclRBW7Z2SJggSAJaQmO9g

u7kieXbtf+5A7gUSb7icD629OiYCEJMTKTpVS/Pk7NDx/ZXQVzGrkJMKdPFU8nDoOjFLSP

jdbbddYe6zuB/HwabV3Lpaxl38tNG78wAAAMEAxXHS2IRABAvX7+OmZO2JU7+9Gxh/gudJ

eXf76c10kKvUztoe8Mskw79yVq6LtYd0JGOVx0oNgMeZJHmwUc2qVPKaFGEhSG6MuFn3J2

O5+Kt+KfU5M9uAN7tob3+yG18ZJt9FY+5FTK1TV5LmF5OTGBN9XyehT2Miqa8sSu80rwpN

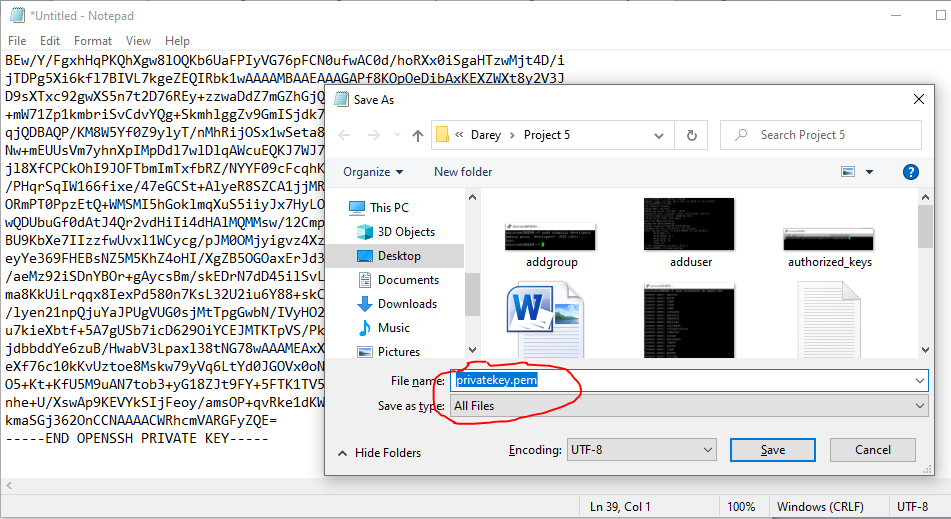
nhe+U/XswAp9KEVYkSIjFeoy/amsOP+qvRke1dKWBsU12IbhnMgjDHVggkYV52l7d9S2bx

kmaSGj362OnCCNAAAACWRhcmVARGFyZQE=

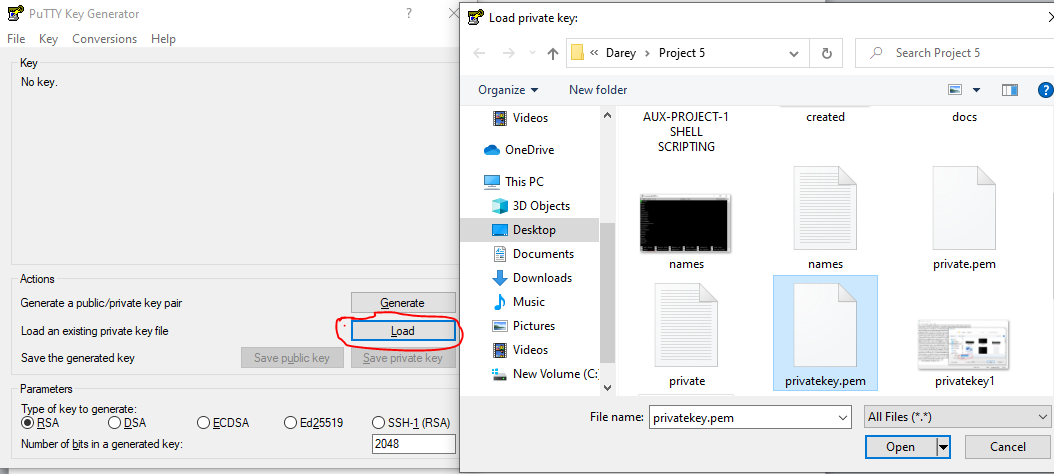
-----END OPENSSH PRIVATE KEY-----

The SSH client used is Putty, which requires a private key with *.ppk* extension. The private key needs to be saved with a .*pem* extension, then converted to *.ppk* using Puttygen.

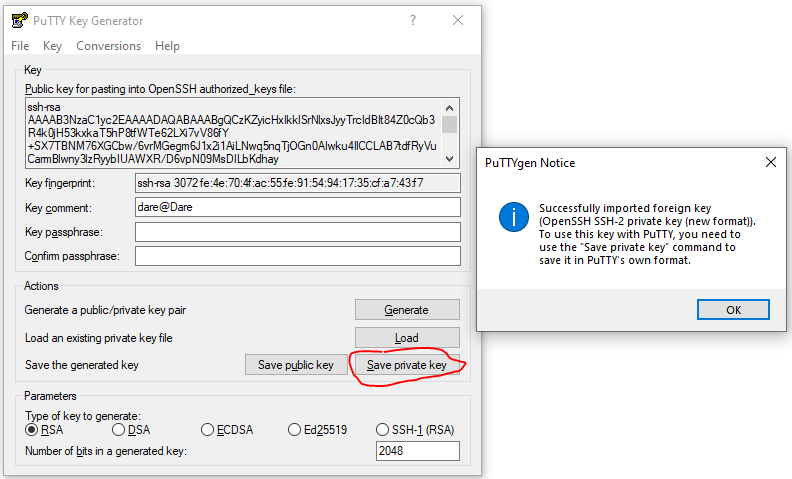
Copy and paste the private key into a blank notepad, then save the file with the extension *.pem*



Open Puttygen and click on ‘Load’. Select the *.pem* key and click ‘Open’

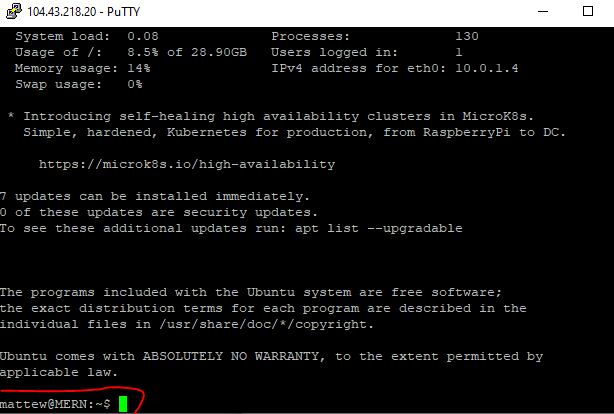


A notification that the key was successfully imported will show up. Click ‘Ok’ and click ‘Save private key’



A warning asking if you want to save the key without a passphrase will pop up. Click ‘Yes’, choose the location for the *.ppk* to be created and save.

With the *.ppk* private key, any of the new users can SSH into the server via the Putty SSH client.



Conclusion

Shell scripting in Linux can help you create complex programs containing conditional statements, loops, and functions

Basic Shell Scripting Commands in Linux are *cat, more, less, head, tail, mkdir, cp, mv, rm, touch, grep, sort, wc, cut* and more.

Credits

<https://darey.io>

[Shell Scripting Tutorial: How to Create Shell Script in Linux/Unix (guru99.com)](https://www.guru99.com/introduction-to-shell-scripting.html)

[Creating basic shell scripts (compciv.org)](http://www.compciv.org/recipes/cli/basic-shell-scripts/)

[Import users from CSV file - Ask Ubuntu](https://askubuntu.com/questions/633695/import-users-from-csv-file)