



Inventory - Linux

Alternative authentication with private key or Sudo user

TITLE	Inventory - Linux
AUTHOR	Docusnap Consulting
DATE	10 September 2026
VERSION	4.2 valid from September 10, 2026

This document contains proprietary information. The reproduction and distribution of this document as a whole or in part as well as the utilization and disclosure of its contents to third parties without the express authorization by itelio GmbH are prohibited. Offenders will be held liable for the payment of indemnification. All rights reserved.

TABLE OF CONTENTS

1.	INTRODUCTION	4
2.	PRIVATE KEYS IN DOCUSNAP	5
2.1	CREATE AND USE PRIVATE KEYS IN DOCUSNAP	5
2.2	IMPORT OF AN EXISTING PRIVATE KEY	6
2.3	DEPOSIT KEY ON LINUX SYSTEM	6
2.4	USING THE PRIVATE KEY FOR THE INVENTORY	10
3.	USE OF A SUDO USER	11
3.1	PERFORM SUDO CONFIGURATION	11
3.2	ACTIVATE SUDO FOR INVENTORY	15

1. INTRODUCTION

For the inventory of Linux systems, Docusnap offers three different variants regarding authentication:

- Root
- Sudo
- Private Keys

The alternatives for authentication (sudo and Private Keys) on the Linux systems to be inventoried are helpful if the root user is not available or access by root users via SSH is blocked.

Furthermore, the script-based inventory for Linux systems is also available. Further information can be found in the provided [HowTo: Docusnap Script Linux](#).

This [HowTo](#) describes the use of one or more Private Keys and the necessary configuration to use a sudo user.

2. PRIVATE KEYS IN DOCUSNAP

2.1 CREATE AND USE PRIVATE KEYS IN DOCUSNAP

Docusnap offers you the possibility to create or import RSA keys, in OpenSSH format, for Linux inventory. The following encryption algorithms can be used for this purpose:

- ecdsa-sha2-nistp256
- ecdsa-sha2-nistp384
- ecdsa-sha2-nistp521
- ssh-ed25519 (can be created by Docusnap)
- ssh-rsa

Private keys can be created and managed in Docusnap administration.

- Administration - Inventory – Private Key Management

Click the **New** button to create a private key. Enter a **name** for this and choose **Create Key**. The key pair is encrypted using the ssh-ed25519 method. The key used is then encrypted again and stored in the database. A passphrase is not created.

If you want to increase security and additionally store a passphrase, you can create the private key with a third-party product (e.g. PuTTY Key Gen or ssh-keygen).

When creation is complete, you can preview the key - this is useful for better identification when using different keys.

Now select **Save** and the key has been successfully created.

You can repeat the above steps as often as you like, for example to create keys for the different clients in your Docusnap environment and then use them.

With the button **Export PublicKey** you can export the public key and store it on the Linux systems - see [chapter 2.3](#).

2.2 IMPORT OF AN EXISTING PRIVATE KEY

An existing private key can be imported to Docusnap as follows.

- Docusnap - Management - Inventory - Private Key Management New

In the next step, assign a **Name** to the key and select the **Import Private Key** button - select your existing private key.

If a passphrase is used for the key, you will be asked for it. The key is then stored in Docusnap.

With the button **Export PublicKey** you can export the public key and store it on the Linux systems - see [chapter 2.3](#).

2.3 DEPOSIT KEY ON LINUX SYSTEM

The described steps might differ between the Linux distributions. Please inform yourself in advance in which directory and which file the public key for your distribution is to be entered. The following application example is performed on a Ubuntu system (16.04.2 64-bit).

In this HowTo the software WinSCP is used, so that the public key is deposited on the Linux system.

Open WinSCP and establish the connection to the Linux system.

If the server is not yet known to the client, a security message is displayed. Click Yes to add the host key to the list of trusted machines.

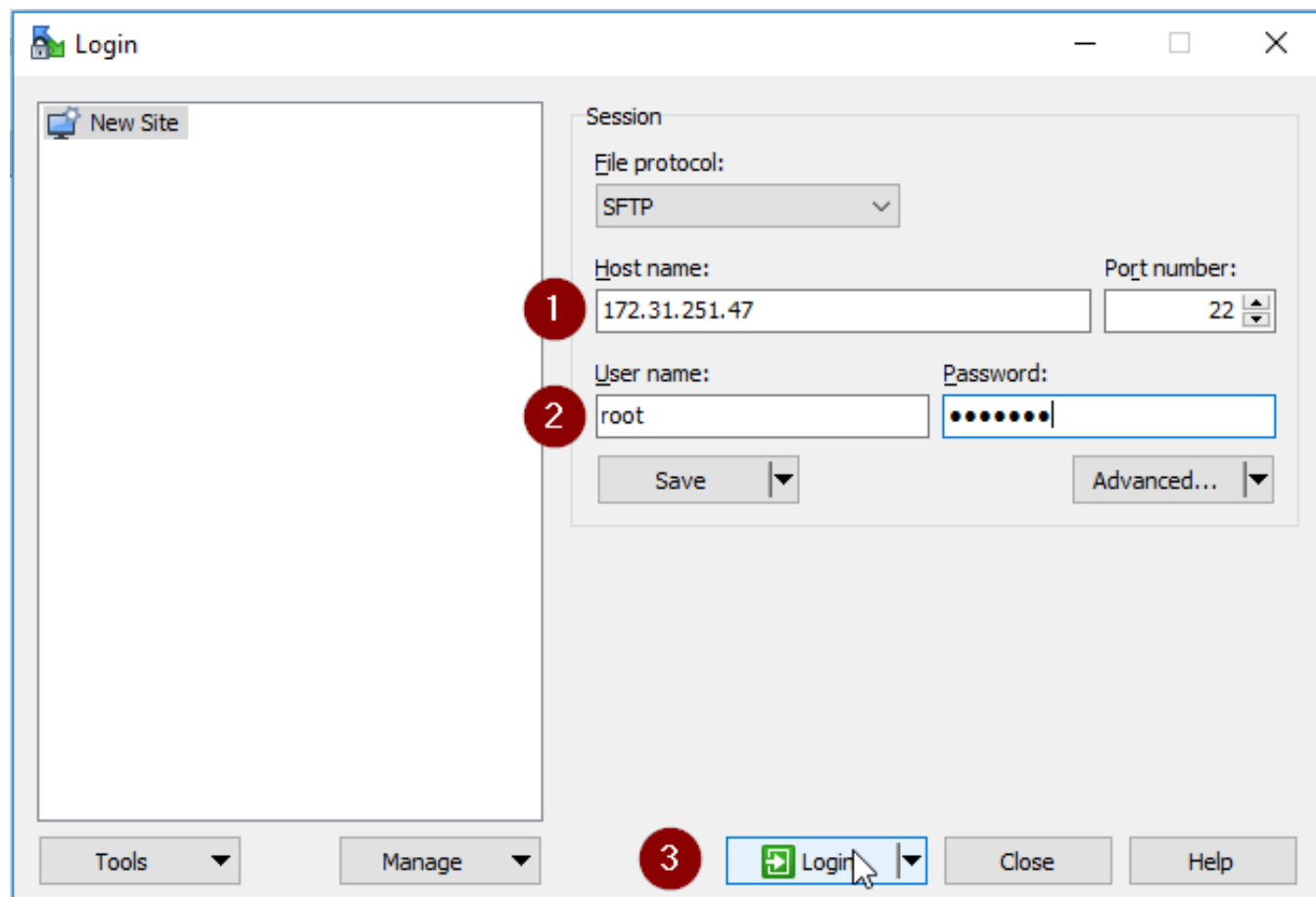


Fig. 1 - Establishing a WinSCP Connection

Step 1

After login, WinSCP changes to the home directory of the logged in user. If this is not the user you connect to via SSH in the future, change to the corresponding home directory.

Step 2

If hidden files and folders are not displayed, please click on the label that shows the number of hidden files.

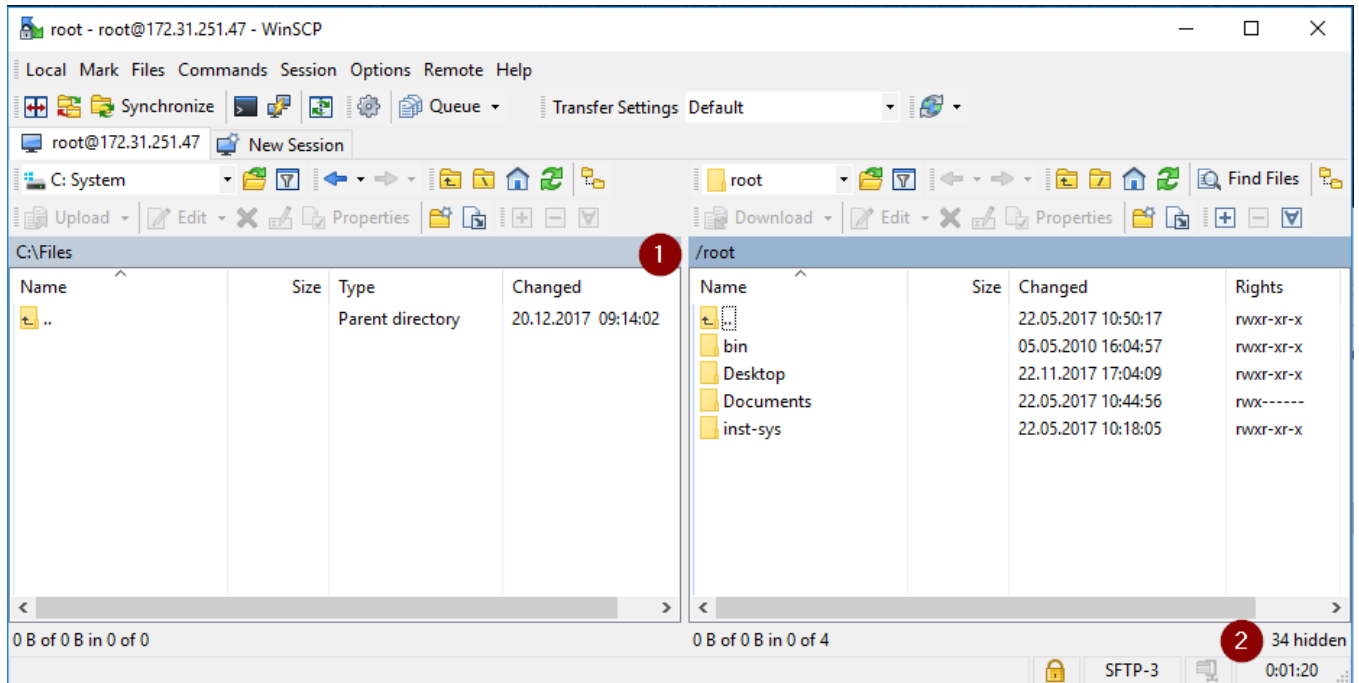


Fig. 2 - Connection to the Target System Established

Change to the directory `.ssh` and edit the file `authorized_keys` there.

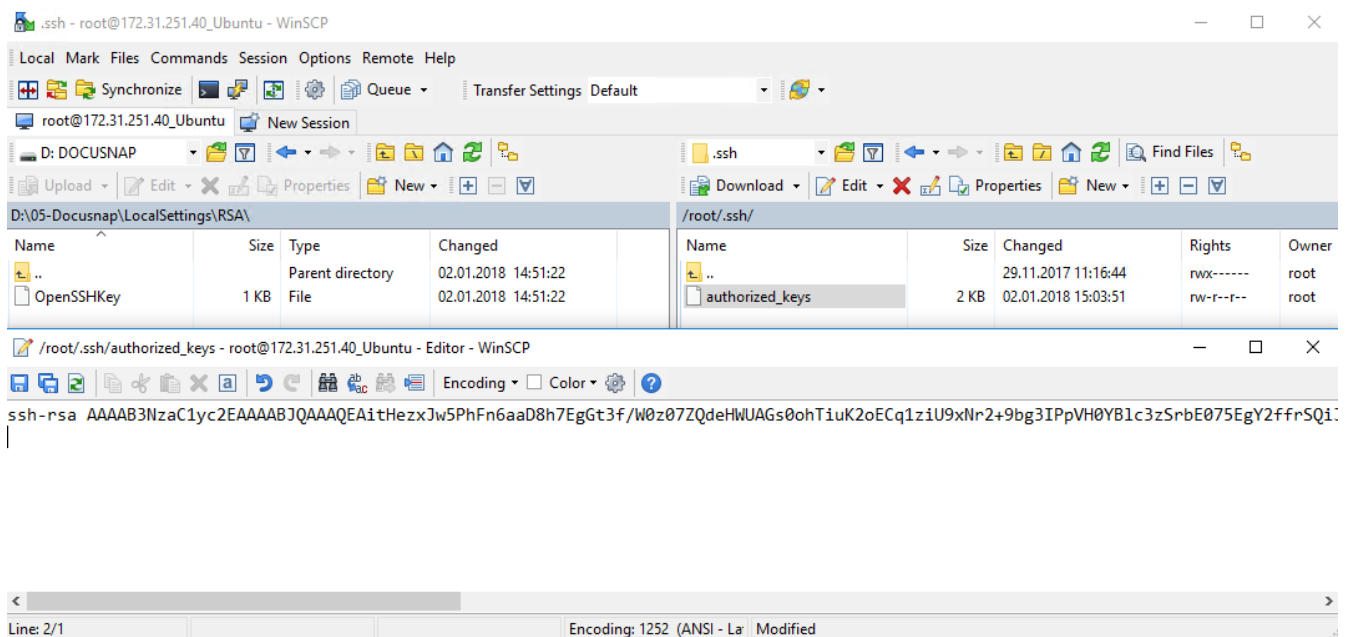


Fig. 3 - Editing the `authorized_keys` File

To store the previously created key, an export of the PublicKey from Docusnap is required. Open the Docusnap - Management - Inventory – Private Key Management and select the button Export PublicKey. Save the file. Open the file with a text editor and copy the PublicKey to the clipboard.

Switch back to WinSCP and insert the PublicKey in a new line. Save the file.

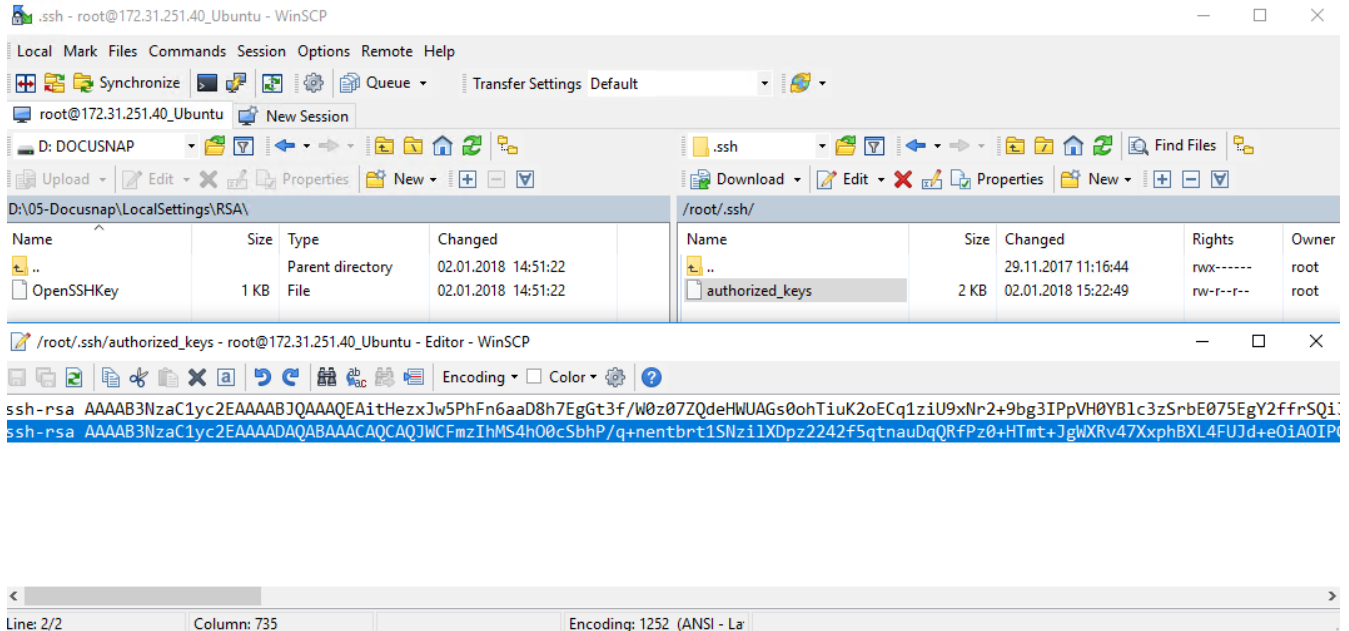


Fig. 4 - Store private key

The PublicKey is now stored on the target system. The inventory can now be carried out. You only need to specify the user name in the wizard.

2.4 USING THE PRIVATE KEY FOR THE INVENTORY

After the public key has been deposited on the Linux systems, the inventory can be carried out with it. Open the Linux Inventory Wizard.

- Discovery – All Wizards – Linux
- Inventar – All Wizards – Linux
- Alle Aufträge – All Wizards – Linux

In **step 3** you have the choice of which authentication you want to use.

You can select private keys for entire IP ranges and also for individual systems. The preselection from the IP ranges can be overwritten for individual systems.

If you do not use a private key, a password must be deposited. However, you can also use both authentication options - private key and password. Both variants are checked, the first one that is successful in the registration is used.

3. USE OF A SUDO USER

3.1 PERFORM SUDO CONFIGURATION

Before you can perform the Linux inventory with a user and the sudo command, you must perform a sudo configuration on the Linux systems - this is described below.

Please note that the possibly newly created user needs an assigned login shell. If this is not the case, the Linux inventory will be incomplete even if the SUDO configuration is correct. Therefore we recommend to create the user with the command **Adduser**. This way the login shell will be assigned automatically.

For the configuration you can use the Docusnap program directory `%Docusnap-Installationfolder%\Tools\scripts`. In this script you will find all commands to which the sudo user is authorized.

Copy the script to the Linux system. In this HowTo the software WinSCP was used.

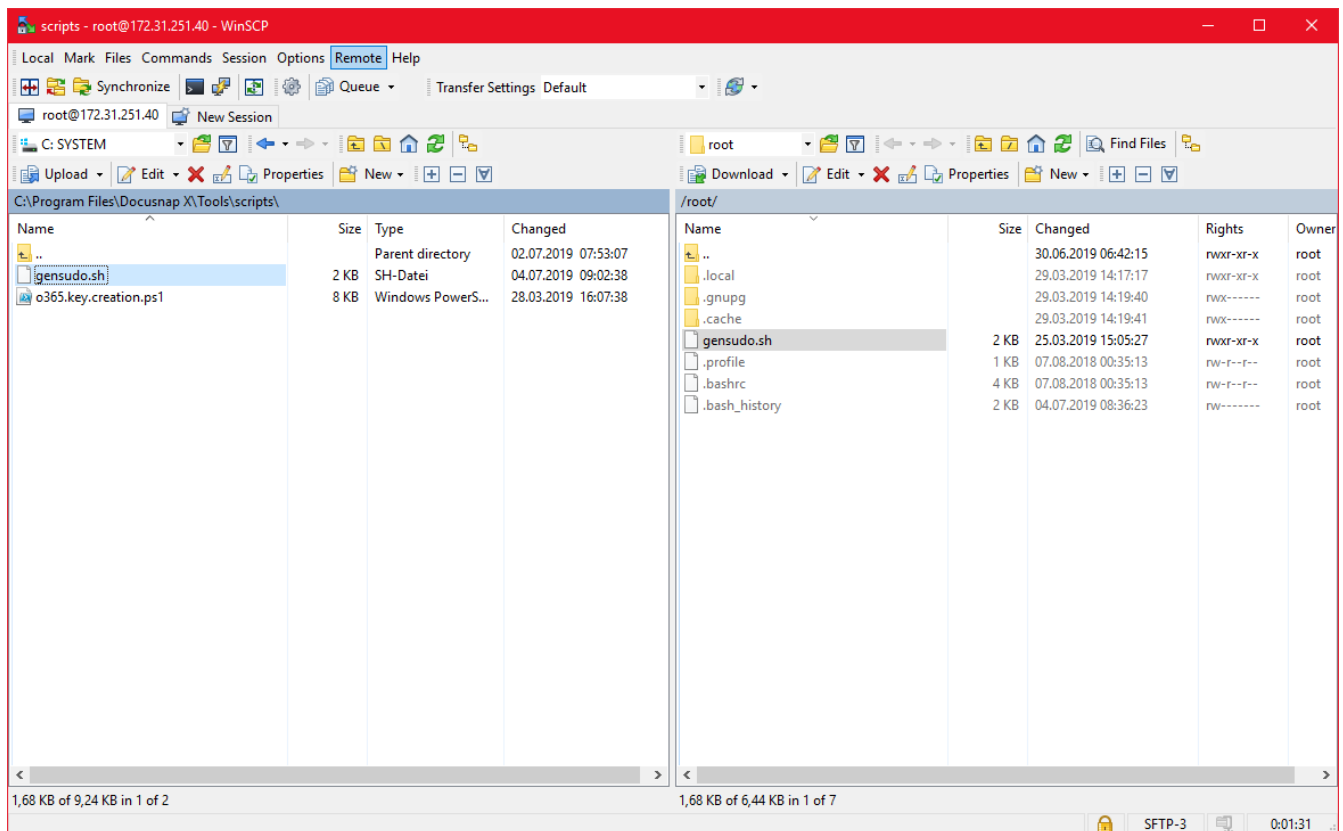
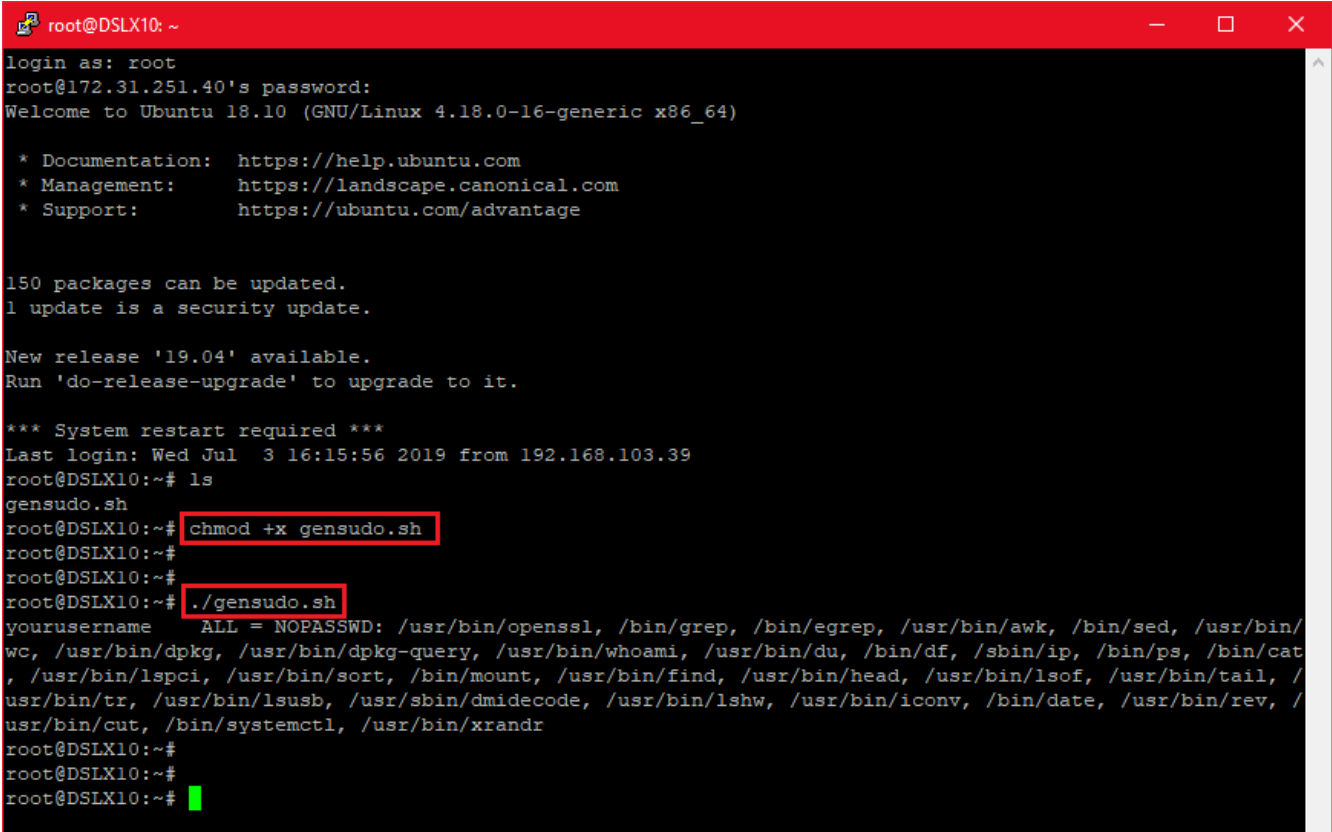


Fig. 5 - Copying the Script

Then connect e.g. with Putty to the console of the Linux system, edit the script to make it executable. Run it afterwards.

```
chmod +x Gensudo.sh
./gensudo.sh
```



```
root@DSLX10: ~
login as: root
root@172.31.251.40's password:
Welcome to Ubuntu 18.10 (GNU/Linux 4.18.0-16-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

150 packages can be updated.
1 update is a security update.

New release '19.04' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Wed Jul  3 16:15:56 2019 from 192.168.103.39
root@DSLX10:~# ls
gensudo.sh
root@DSLX10:~# chmod +x gensudo.sh
root@DSLX10:~#
root@DSLX10:~# ./gensudo.sh
yourusername ALL = NOPASSWD: /usr/bin/openssl, /bin/grep, /bin/egrep, /usr/bin/awk, /bin/sed, /usr/bin/wc, /usr/bin/dpkg, /usr/bin/dpkg-query, /usr/bin/whoami, /usr/bin/du, /bin/df, /sbin/ip, /bin/ps, /bin/cat, /usr/bin/lspci, /usr/bin/sort, /bin/mount, /usr/bin/find, /usr/bin/head, /usr/bin/lsof, /usr/bin/tail, /usr/bin/tr, /usr/bin/lshw, /usr/sbin/dmidecode, /usr/bin/lshw, /usr/bin/iconv, /bin/date, /usr/bin/rev, /usr/bin/cut, /bin/systemctl, /usr/bin/xrandr
root@DSLX10:~#
root@DSLX10:~#
root@DSLX10:~#
```

Fig. 6 - Making a Script Executable and Executing it

Copy the output and paste it into a text editor.

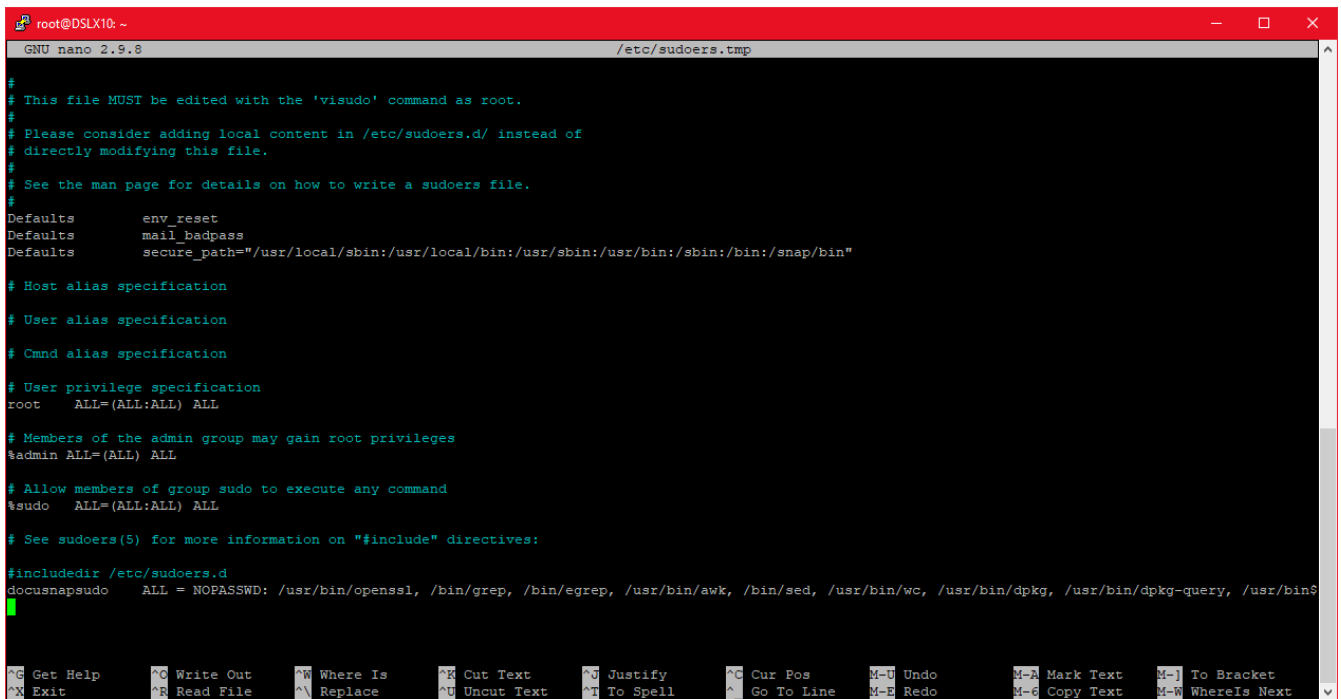
At the beginning of the output you have to change the following: Change YourUserName with the name of the sudo user. After completion of the configuration, the specified user has the permissions to execute the specified commands as root.

```
yourusername ALL = NOPASSWD: /usr/bin/systemctl -a --no-pager -q -l --no-legend --no-block, /usr/sbin/dmidecode "", /usr/sbin/dmidecode -h, /usr/sbin/dmidecode --type 17, /usr/sbin/dmidecode --type 6, /usr/sbin/ip addr show, /usr/sbin/ip addr show *, /usr/sbin/ip address show dev *, /usr/sbin/ip route, /usr/bin/ps -ef, /usr/bin/ps --ppid 1 -f, /usr/bin/ps --no-headers -p *, /usr/bin/dpkg-query --show *, /usr/bin/df -P, /usr/bin/lspci "", /usr/bin/lspci -vv, /usr/bin/lspci --version, /usr/bin/lspci -vvv -m -d *, /usr/bin/lspci -v -m -s *, /usr/bin/mount -l, /usr/bin/mount -t efivarfs efivarfs /sys/firmware/efi/efivars, /usr/bin/ss -apt *, /usr/bin/ss -apu *, /usr/bin/lsof -P -b +c0 -n -i tcp, /usr/bin/lsof -P -b +c0 -i tcp, /usr/bin/lsof -P -b +c0 -n -i udp, /usr/bin/lsof -P -b +c0 -n +M -i tcp, /usr/bin/lsof -P -b +c0 +M -i tcp, /usr/bin/lsof -P -b +c0 +M -n -i tcp, /usr/bin/lshw "", /usr/bin/lshw -xml -C disk, /usr/bin/lshw -class display -xml, /usr/bin/lshw -class multimedia -xml, /usr/sbin/pvdisplay -c, /usr/sbin/vgdisplay -c, /usr/sbin/lvdisplay -c, /usr/bin/locale "", /usr/bin/locale -a, /usr/bin/file /etc/passwd, /usr/sbin/blkid -p *, /usr/bin/od -An -t u1 -j 4 -N 1 -v *, /usr/bin/od -An -tx1 -j 4 -v *, /usr/bin/find /proc/ide/ -name media, /usr/bin/find /var/spool/cron/ -type f, /usr/bin/find /usr/local/sbin/ -
```

```
type f, /usr/bin/find /usr/local/bin/ -type f, /usr/bin/cat, /usr/bin/grep, /usr/bin/stat, /usr/bin/test, /usr/bin/iconv
```

Please note that the previous version of the script was as of 10 September 2026. Changes could have taken place in the meantime.

Copy the custom output and switch back to Putty. Type visudo in Putty and go to the end of the file and paste the clipboard (right mouse button).



```

root@DSLX10: ~
GNU nano 2.9.8 /etc/sudoers.tmp

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include_dir /etc/sudoers.d

docuSnapsudo    ALL = NOPASSWD: /usr/bin/openssl, /bin/grep, /bin/egrep, /usr/bin/awk, /bin/sed, /usr/bin/wc, /usr/bin/dpkg, /usr/bin/dpkg-query, /usr/bin/

```

Fig. 7 - Inserted Script with Custom Username

Exit (Ctrl + X) and save (Y) the file with the existing filename (Enter).

You can use the `cat /etc/sudoers` command to check whether the changes have been applied.

```
root@DSLX10: ~  
root@DSLX10:~# cat /etc/sudoers  
#  
# This file MUST be edited with the 'visudo' command as root.  
#  
# Please consider adding local content in /etc/sudoers.d/ instead of  
# directly modifying this file.  
#  
# See the man page for details on how to write a sudoers file.  
#  
Defaults        env_reset  
Defaults        mail_badpass  
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"  
  
# Host alias specification  
  
# User alias specification  
  
# Cmnd alias specification  
  
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
  
# Members of the admin group may gain root privileges  
%admin   ALL=(ALL) ALL  
  
# Allow members of group sudo to execute any command  
%sudo    ALL=(ALL:ALL) ALL  
  
# See sudoers(5) for more information on "#include" directives:  
  
#includedir /etc/sudoers.d  
docusnapsudo    ALL = NOPASSWD: /usr/bin/openssl, /bin/grep, /bin/egrep, /usr/bin/awk, /bin/sed, /usr/bin/wc, /usr/bin/whoami, /usr/bin/du, /bin/df, /sbin/ip, /bin/ps, /bin/cat, /usr/bin/lspci, /usr/bin/sort, /bin/mount, /usr/bin/find, /tail, /usr/bin/tr, /usr/bin/lsusb, /usr/sbin/dmidecode, /usr/bin/lshw, /usr/bin/iconv, /bin/date, /usr/bin/rev, /usr/ndr  
root@DSLX10:~#
```

Fig. 8 - Reviewing the Change

3.2 ACTIVATE SUDO FOR INVENTORY

The inventory via the sudo user can then be activated in the Linux inventory wizard. Enter an IP address range, the user, his password and activate the option **Use Sudo**.

LIST OF FIGURES

FIG. 1 - ESTABLISHING A WINSCP CONNECTION	7
FIG. 2 - CONNECTION TO THE TARGET SYSTEM ESTABLISHED	8
FIG. 3 - EDITING THE AUTHORIZED_KEYS FILE	8
FIG. 4 - STORE PRIVATE KEY	9
FIG. 5 - COPYING THE SCRIPT	11
FIG. 6 - MAKING A SCRIPT EXECUTABLE AND EXECUTING IT	12
FIG. 7 - INSERTED SCRIPT WITH CUSTOM USERNAME	13
FIG. 8 - REVIEWING THE CHANGE	14

VERSION HISTORY

date	description
January 11, 2018	Version 1.0 created
October 24, 2018	Changed Screenshots
April 24, 2020	Version 3.0 - Revision of the HowTo for Docusnap 11
January 5, 2023	Version 4.0 – Revision of the HowTo for Docusnap 12
November 29, 2024	Version 4.1 - Customisation of the key that can be created by Docusnap



Docusnap[®]

-=www.docusnap.com/support=- proudly presents
© itelio GmbH - www.itelio.com