Unix & Linux Stack Exchange is a question and
answer site for users of Linux, FreeBSD and
other Un*x-like operating systems. It only takes
a minute to sign up.

Anybody can ask a question

✕

Anybody can answer

Sign up to join this community

The best answers are voted up and
rise to the top

UNIX & LINUX

How to dd a remote disk using SSH on local machine and save to a local disk

Asked 9 years, 8 months ago   Modified 1 year, 6 months ago   Viewed 241k times

▲

125

▼

🔖

🕓

How can I create a backup of a remote disk using SSH on my local machine and save it to
a local disk?

I've tried the following:

```
ssh hostname@my.ip.address "sudo dd if=/dev/sdX " | \
    dd of=/home/username/Documents/filename.image`
```

However, I receive the following error:

asked May 31, 2014 at 4:30

Qu0rk
1,391   2   10   9

By clicking "Accept all cookies", you agree Stack Exchange
can store cookies on your device and disclose information in
accordance with our Cookie Policy.

Accept all cookies   Necessary cookies only

orted by: Highest score (default)   ⬍

Customize settings

If your intent is to backup a remote computer's **HDD A** via SSH to a single file that's on your local computer's HDD, you could do one of the following.

Examples

run from remote computer

```
$ dd if=/dev/sda | gzip -1 - | ssh user@local dd of=image.gz
```

run from local computer

```
$ ssh user@remote "dd if=/dev/sda | gzip -1 -" | dd of=image.gz
```

Live example

```
$ ssh skinner "dd if=/dev/sda5 | gzip -1 -" | dd of=image.gz
208782+0 records in
208782+0 records out
106896384 bytes (107 MB) copied, 22.7608 seconds, 4.7 MB/s
116749+1 records in
116749+1 records out
59775805 bytes (60 MB) copied, 23.9154 s, 2.5 MB/s

$ ll | grep image.gz
-rw-rw-r--.  1 saml saml  59775805 May 31 01:03 image.gz
```

Methods for monitoring?

1. Login via `ssh` in another terminal and `ls -l` the file to see what it's size is.

2. You can use `pv` to monitor the progress of a large dd operation, for instance, for the

@local dd of=image.gz

ts. Something like:

inally left via comments by @Ryan

to make them more obvious.

Share Improve this answer

Follow

edited Jan 21, 2022 at 8:40

Stephen Kitt
430k   55   1114   1205

answered May 31, 2014 at 5:05

slm ♦
368k   121   774   878

I tried this for a similar problem (backing up remote computer `ONE`'s hard drive to remote computer `TWO`'s storage), like this:
[root@ONE]# dd if=/dev/sda1 | ssh root@TWO `dd` of=/root/Public/ONE/sda1.img
409600+0 records in 409600+0 records out 209715200 bytes (210 MB) copied, 0.894929 s, 234 MB/s Pseudo-terminal will not be allocated because stdin is not a terminal root@TWO's password: The problem is that the `/root/Public/ONE/sda1.img` file was created on `ONE`, not on `TWO` as expected. How do I fix this? – Urhixidur Jul 15, 2015 at 13:59 ✎

1   Can someone give us an example of using `pv` from the local machine? Thank you. – TCB13 Dec 16, 2015 at 15:08

it is better to use lbzip2 or pigz insted of gzip – Shimon Doodkin Mar 10, 2016 at 0:15

12   Regarding monitoring: Newer dd versions also have the `status=progress` option. Thus, you do not need to run `pkill -USR1 dd` from another terminal anymore. – Till Schäfer Jan 2, 2018 at 15:23

2   restore: gunzip -c xxx.gz | pv | dd of=/dev/sda1 – nobjta_9x_tq Apr 11, 2020 at 6:12

▲

15

▼

The error you receive is due to the use of sudo remotely (you'd be asked for the password but you don't have a tty to enter it). On the other hand, as a normal user you can't normally use simply `dd` as suggested in other answer (you don't have the required permissions to the device). You can solve the problem by giving yourself rights to execute `dd` as sudo without a password. You can do this by editing the sudoers file (on the remote!):

```
sudo visudo
```

add the following line:

```
userfoo ALL=(ALL) NOPASSWD: /bin/dd if=/dev/sdX
```

`=test.dd`

y in sudoers, allowing you to run permissions to the minimum you

Share | Improve this answer | Follow
Or, you can just run 'sudo ls' at the applicable end (and type in password), just prior to running the 'sudo dd ...' command – MikeW Feb 19, 2019 at 11:28

answered Jan 4, 2015 at 15:30

elfantin

1 With newer versions of dd adding status=progress even allows monitoring of copy progress. **151** 1 2
– sensslen May 21, 2019 at 9:16

This is the correct answer regarding the error message. But you could add these alternatives: use `ssh -t` to execute the command with a `tty` allowing for interactive password input / use `sudo -S` to read password from `stdin` even without a `tty`. – acran Nov 1, 2020 at 14:34

3 @MikeW note that this might not work since by default `sudo` only remembers the authentication for the same console session, see `timeout_type` in `man 5 sudoers`. – acran Nov 1, 2020 at 14:35

▲

8

▼

🔖

↺

I've just used a modified version on a virtual Linux Mint box to clone a physical hard drive on an aging RedHat server.

I ran the following as root on the virtual box:

```
ssh root@192.168.1.5 "dd if=/dev/cciss/c0d0" | dd of=/dev/sdb
```

- 192.168.1.5 is the physical Red Hat Enterprise Linux 2.1 box.

- /dev/sdb is a new virtual disk, which when finished cloning will be the basis of a virtual version of the old dying physical box.

Share | Improve this answer

Follow

edited Apr 25, 2016 at 1:01

slm ♦
368k 121   774   878

answered May 20, 2015 at 4:47

Fred   Fred Nerk
Nerk': **81**   1   1

Combining these answers allowed me to clone to another Linux host and avoid the 'permission denied' error:

- RPI4 SD card (`/dev/mmcblk0` - this clones the entire card)
- avoid permission issue by prompting for elevated privilege (`sudo -S`)
- see progress as it copies (`status=progress`)
- compressing the copy (`gzip`)
- specify buffer size (`bs=64k`) for improved speed. You can specify higher values. please read this

```
ssh user@1.1.1.1 "sudo -S dd if=/dev/mmcblk0 bs=64k status=progress | gzip -1 -"
| dd of=rpiimg.gz bs=64k
```

Share Improve this answer

Follow

edited Nov 1, 2020 at 0:18

Hummus_Ful
3   3

answered Nov 28, 2019 at 3:38

Paul Sturm
171   1   3

1   great - works like a charm. – opinion_no9 Mar 9, 2023 at 20:12

I was in a similar situation recently. The difference for me was, that

- due to security measures `root` `ssh` access was disabled
- `sudo` was not installed (deliberately)

So what I did was:

1. examine device file permissions (grep for appropriate block device naming sheme (`hd` , `sd` , `nvme` , etc.))

```
user@machine:~$ ls -la /dev/ |grep sd
brw-rw----  1 root disk      8,   0 Nov 22 13:36 sda
brw-rw----  1 root disk      8,   1 Nov 22 13:36 sda1
brw-rw----  1 root disk      8,   2 Nov 22 13:36 sda2
brw-rw----  1 root disk      8,   5 Nov 22 13:36 sda5
```

2. add the user to the `disk` group

```
root@machine:~$ usermod -aG user disk
```

3. then run the `dd` the same way you already tried (but w/o `sudo`)

```
ssh user@my.ip.address "dd if=/dev/sdX " |  dd of=/home/username/Documents/
filename.image
```

(or also chain-in the `gzip` as others suggested)

Share Improve this answer Follow

answered Nov 23, 2020 at 10:03

helvete
151   4

ciation bonus) in order to answer this
om spam and non-answer activity.

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our Cookie Policy.

Accept all cookies   Necessary cookies only

Customize settings