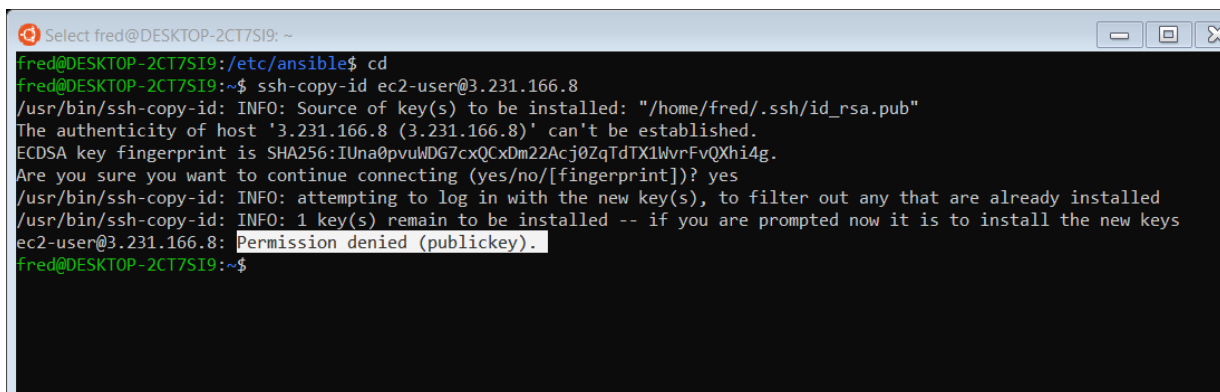


COMMON ERRORS THAT OCCUR WHEN WORKING WITH ANSIBLE

A: when trying to copy the public key to the node.

Main error: **Permission_denied (publickey)**

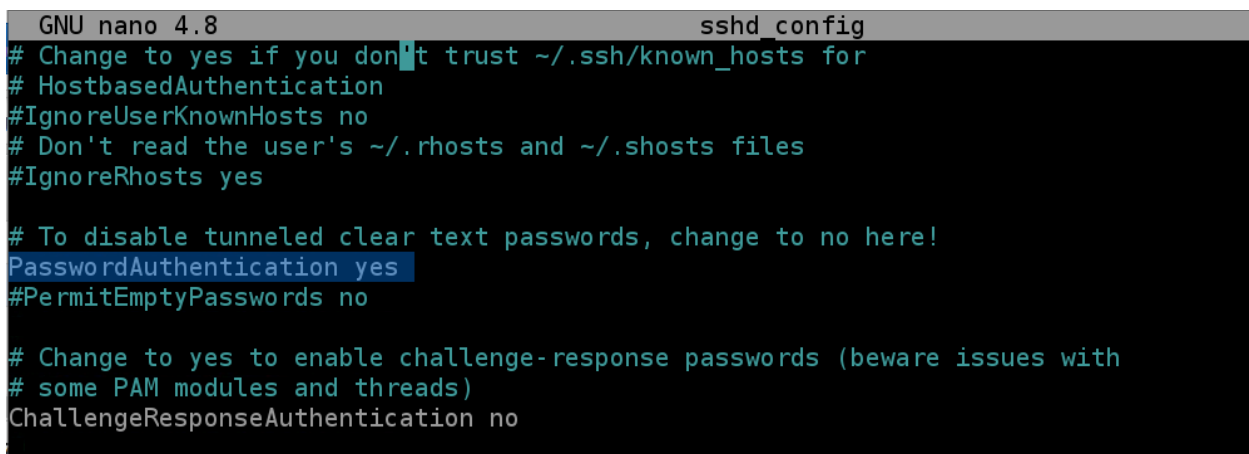


```
Select fred@DESKTOP-2CT7SI9: ~
fred@DESKTOP-2CT7SI9:/etc/ansible$ cd
fred@DESKTOP-2CT7SI9:~$ ssh-copy-id ec2-user@3.231.166.8
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/fred/.ssh/id_rsa.pub"
The authenticity of host '3.231.166.8 (3.231.166.8)' can't be established.
ECDSA key fingerprint is SHA256:IUa0pvuWdG7cxQCxDm22Acj0ZqTdTX1WvrFvQXhi4g.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ec2-user@3.231.166.8: Permission denied (publickey).
fred@DESKTOP-2CT7SI9:~$
```

SOLUTION

Step 1: log into the node

- Type `sudo su` to switch privilege to root.
- Move to `ssh` directory by typing `cd /etc/ssh`
- While in `ssh` directory nano into `sshd_config`
- Under `#PermissionAuthentication`
- Change `PasswordAuthentication no` to `PasswordAuthentication yes`



```
GNU nano 4.8                                sshd_config
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no
```

Now to refresh ssh directory depends on the Linux distro you are using

- For ubuntu type -----→ `service ssh restart`

OR

```
# PasswordAuthentication. Depending on your PAM configuration you may want to
# Change this to "password" for password-based authentication.
# See the man page of the PAM module you are using for more details.

root@ip-172-16-0-105:/etc/ssh# service ssh restart
root@ip-172-16-0-105:/etc/ssh#
```

- For any other Linux distro, type -----→ `service sshd restart`

```
[root@ip-172-16-0-66 ssh]# service sshd restart
Stopping sshd: [ OK ]
Starting sshd: [ OK ]
[root@ip-172-16-0-66 ssh]#
```

Step 3: Go back and run the `ssh-copy-id user@target` command again.

```
fred@DESKTOP-2CT7S19:~$ ssh-copy-id ubuntu@75.101.222.125
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/fred/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ubuntu@75.101.222.125's password:
```

Everything looks great, but password is needed to authenticate the user.

Step 3: Create the `user's` password

Go back to the note and type `sudo passwd user_name`.

Enter your preferred `password`.

```
ubuntu@ip-172-16-0-105:~$ sudo passwd ubuntu
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ip-172-16-0-105:~$
```

Now, go back and enter the **password** you just created to authenticate the user.

```
fred@DESKTOP-2CT7SI9:~$ ssh-copy-id ubuntu@75.101.222.125
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/fred/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ubuntu@75.101.222.125's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ubuntu@75.101.222.125'"
and check to make sure that only the key(s) you wanted were added.
```

Result:

- Number of key(s) added: 1
- Now try logging into the machine, with "**ssh 'ubuntu@75.101.222.125'**" and check to make sure that only the key(s) you wanted were added

Ok, Let's try logging into the machine, with: "**ssh 'ubuntu@75.101.222.125'**"

```
fred@DESKTOP-2CT7SI9:~$ ssh ubuntu@75.101.222.125
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1029-aws x86_64)

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

System information as of Fri Jan  8 19:39:12 UTC 2021

System load:  0.0           Processes:            111
Usage of /:   16.9% of 7.69GB Users logged in:             1
Memory usage: 22%          IPv4 address for eth0: 172.16.0.105
Swap usage:   0%

1 update can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

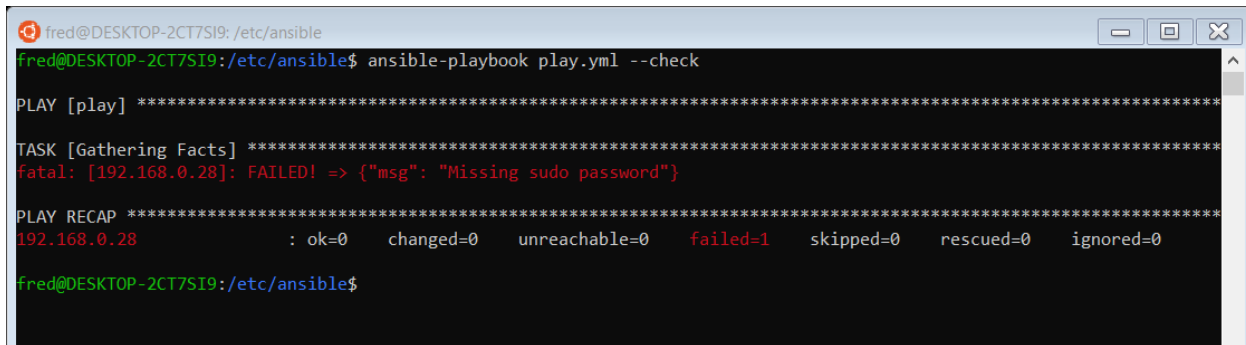
Last login: Fri Jan  8 19:32:14 2021 from 68.38.168.62
ubuntu@ip-172-16-0-105:~$
```

Now you've successfully authenticated **ssh** connection between the two machines.

Congratulations!!

B: When running playbook.

Main error: **fatal: [target_name]: FAILED! => {"msg": "Incorrect sudo password"}**

A terminal window titled 'fred@DESKTOP-2CT7SI9: /etc/ansible' shows the command 'ansible-playbook play.yml --check' being executed. The output indicates a failure on host 192.168.0.28 with the message 'Missing sudo password'.

```
fred@DESKTOP-2CT7SI9: /etc/ansible$ ansible-playbook play.yml --check

PLAY [play] *****

TASK [Gathering Facts] *****
fatal: [192.168.0.28]: FAILED! => {"msg": "Missing sudo password"}

PLAY RECAP *****
192.168.0.28      : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

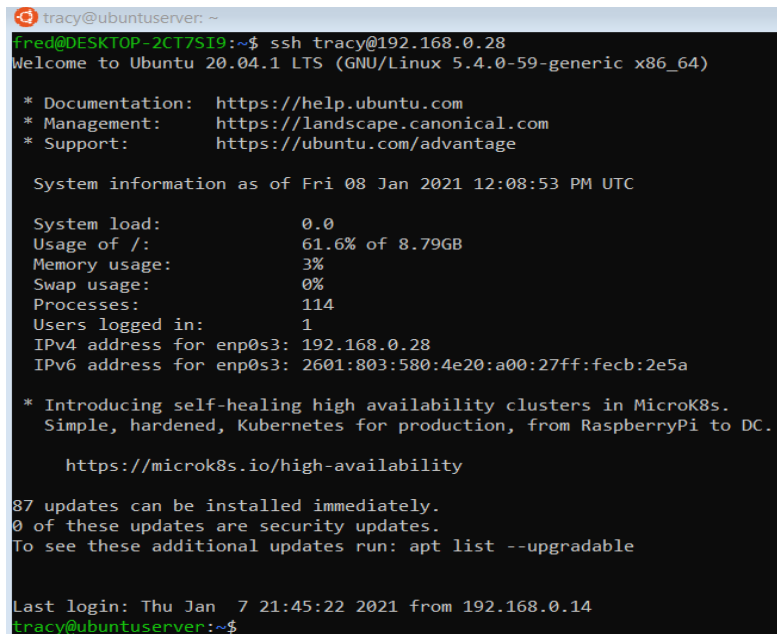
fred@DESKTOP-2CT7SI9: /etc/ansible$
```

SOLUTION

This error appears when **user** on the **node (server)** you are trying to deploy your **code** does not have **root** privilege.

In this case

Step 1. **ssh** into the **node** by **ssh user_name@target**

A terminal window shows a successful SSH login from 'fred@DESKTOP-2CT7SI9' to 'tracy@ubuntuserver'. The output displays system information for Ubuntu 20.04.1 LTS, including system load, memory usage, and network details.

```
tracy@ubuntuserver: ~$
fred@DESKTOP-2CT7SI9:~$ ssh tracy@192.168.0.28
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-59-generic x86_64)

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

System information as of Fri 08 Jan 2021 12:08:53 PM UTC

System load:          0.0
Usage of /:            61.6% of 8.79GB
Memory usage:         3%
Swap usage:           0%
Processes:            114
Users logged in:      1
IPv4 address for enp0s3: 192.168.0.28
IPv6 address for enp0s3: 2601:803:580:4e20:a00:27ff:feeb:2e5a

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

   https://microk8s.io/high-availability

87 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Last login: Thu Jan  7 21:45:22 2021 from 192.168.0.14
tracy@ubuntuserver:~$
```

Great! you are in the node

Step 2. While in the **node**, type **sudo su** to gain **root** privilege.

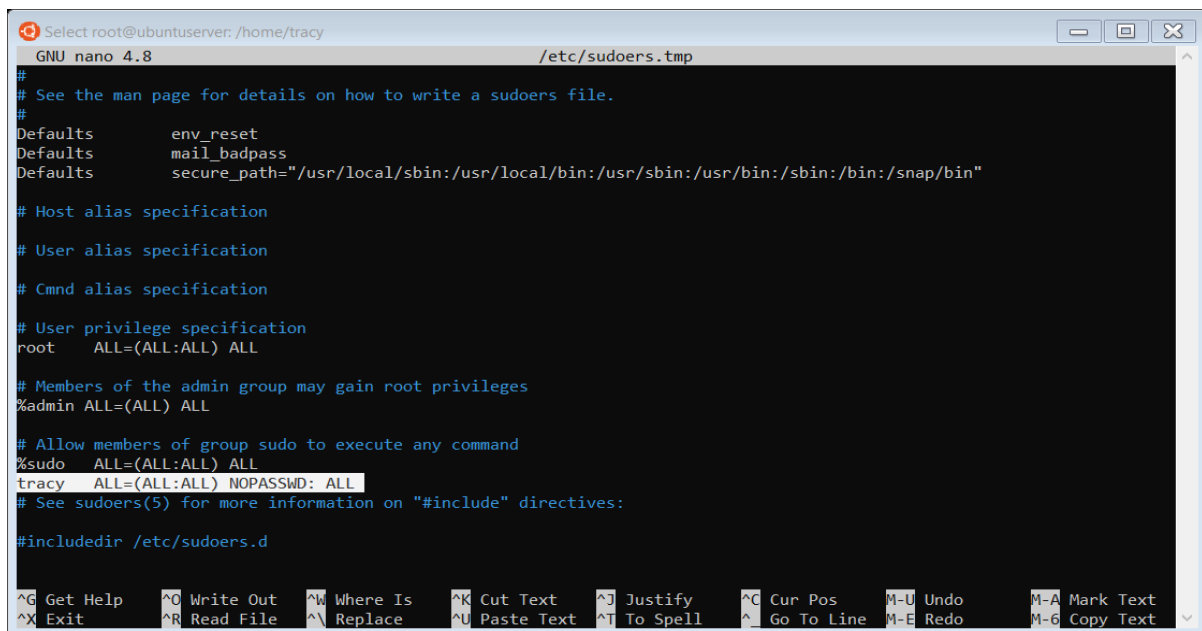
```
root@ubuntuuserver: /home/tracy
tracy@ubuntuuserver:~$ sudo su
root@ubuntuuserver: /home/tracy#
```

If “[sudo] password for tracy:” prompted, put password for user “tracy”

Step 3. Type ”visudo” to add tracy to root privilege.

Under #Allow members of group sudo to execute any command

type “user_name ALL=(ALL:ALL) NOPASSWD: ALL” to give root privilege.



```
GNU nano 4.8 /etc/sudoers.tmp
# 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

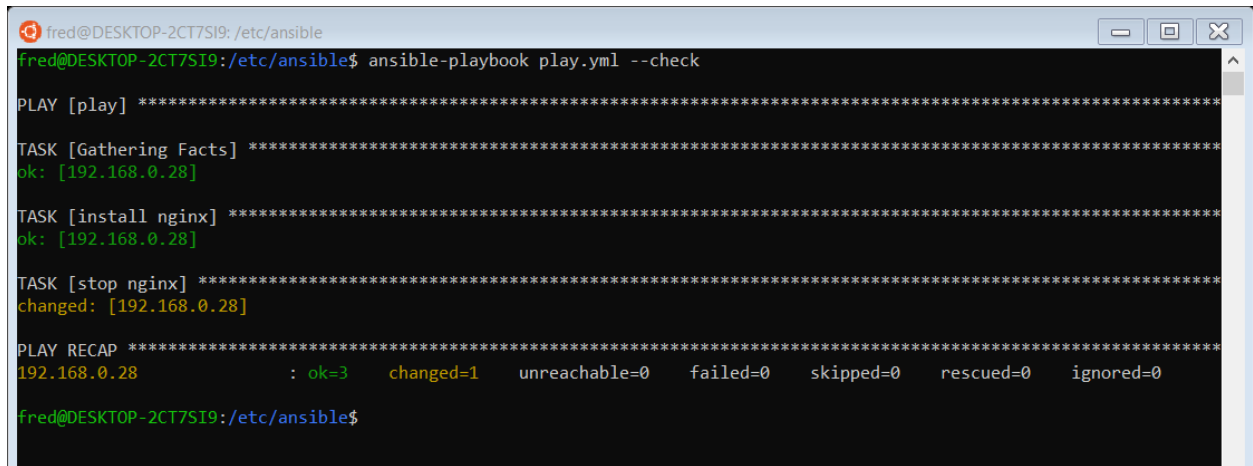
# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
tracy    ALL=(ALL:ALL) NOPASSWD: ALL
# See sudoers(5) for more information on "#include" directives:

#include_dir /etc/sudoers.d

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      M-U Undo        M-A Mark Text
^X Exit          ^R Read File    ^N Replace      ^U Paste Text   ^T To Spell     ^_ Go To Line    M-E Redo        M-G Copy Text
```

Step 4. Let's try to deploy the code again

A terminal window titled 'fred@DESKTOP-2CT7SI9: /etc/ansible' showing the output of the command 'ansible-playbook play.yml --check'. The output displays the results of a dry-run for a playbook named 'play'. It shows three tasks: 'Gathering Facts' (ok), 'install nginx' (ok), and 'stop nginx' (changed). A 'PLAY RECAP' section at the bottom summarizes the results for host '192.168.0.28': 3 ok, 1 changed, and 0 for all other status categories.

```
fred@DESKTOP-2CT7SI9: /etc/ansible$ ansible-playbook play.yml --check

PLAY [play] *****

TASK [Gathering Facts] *****
ok: [192.168.0.28]

TASK [install nginx] *****
ok: [192.168.0.28]

TASK [stop nginx] *****
changed: [192.168.0.28]

PLAY RECAP *****
192.168.0.28 : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

fred@DESKTOP-2CT7SI9: /etc/ansible$
```

Now everything looks great

Congratulations!!!