

How to connect two Linux instances

Step1:

- Create two instances Server 1(master) And Server2(client)
- Create the ssh-keygen in the Server1

Step 2:

- Open that key cat /root/.ssh/id_rsa.pub
- Copy that key
- Go to server 2 and paste it into the Server2 vi /root/.ssh/authorized_keys
- Paste the key at top



```
root@ip-172-31-9-130:~  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDKb5TjX4skU5d1AjWoC6tj1hyEtEIgob6rQVI3arQ+k1BJh16MmXYFgG  
AYcPFN8uPLVi1/NcvP2W3S4DTot1DQabP0+wLi7EpN0u0cK4PQPGL8Q/wKz3onDcnFfOPqM9+Gka5sdwV4M1nW/nDtNTY  
NXR6Hsp6/R+jB1Eet0Fede1ZjmTzUk08y+1KyS6raBquPaJb5iw5WZFibQPygSYd1SRDm4Iw0FnppXB/50dCx0zruzWZz+  
Kj2G7ZT4uOs2yOfyyQrpYb3Rzh6eqHb1RKfH/KVBnRt2e2ksuLfX1JbNbnTn7ddh8wQa5xdChRPGLke5iVgoDpyJM2Q20t  
T52cFIVlGgaIqkJzoY+bjjSBXsLA/DYUJvkJTGfDRcmR0d0k6bvDv4CCdq5R731vMQXc1F+wog0rzTXWjuP5zxVis4shBv  
4CuqFge425ZzWL+D2RLGYxN5pTRj0L02BXfIEje12h3QAmfNzHcdb0SZ5g/J80fVBmozKuJIgFOQ1zA6M= root@ip-172  
-31-12-243.ap-south-1.compute.internal  
  
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the us  
er \"ec2-user\" rather than the user \"root\".';echo;sleep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2  
EAAAADAQABAAQgC+frax2mp4Ra8TfLj/R/LMikVz8qvVrGnhMOIzBI8kXDWPiohRxI145SimNwcM2G9abDhiM+jpT9AK  
SKtxgdptq19XjVh96j7Vi do0W1MZCrjMs4Cb4rHaKvTfcP2b94rEj8wcqt+iea2x8aCvYsz2Xx6W2gFbYA1WTtqmuhSpCM  
VX7ezi7ZKvrpB4y8Hp1c/FhT8UX0bJ3L1/76pv1/b6Qz7dZzwipZB0fB10H2nRsnvryI0b+It+8QuQqa4tVTFTFWFrYYTm  
Ojhz86k1KPuAbwPQko2uvHG2/A1X+uBjGAUj89TfiVXSJWB/kJDzdG60izdJdNCBwtihosqZeCw1 againsonar  
~  
~
```

Step 3:

- Now change the configuration setting of both Servers vi/etc/ssh/sshd_config
- Change only the Password authentication to yes
- Then restart the sshd in both Servers after the configuration
- Systemctl restart sshd

```
root@ip-172-31-5-194:/etc/ssh
#GSSAPIEnableK5Users no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via KbdInteractiveAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
# WARNING: 'UsePAM no' is not supported in RHEL and may cause several
# problems.
#UsePAM no

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
#X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# override default of no subsystems
Subsystem sftp /usr/libexec/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#    X11Forwarding no
#    AllowTcpForwarding no
#    PermitTTY no
#    ForceCommand cvs server
PasswordAuthentication yes
```

Step 4:

- Now write ssh root@public ip of client
- Now you can access the Server 2 From Server 1

Now we have to do rsync (Sync the content of Server1 into Server2):

The command for the rsync is

```
rsync [OPTION] -e ssh [SRC]... [USER@]HOST:DEST
```

```
- rsync -av -e ssh /root/amrit/* root@65.0.7.1:/root/aps/
```

The files from the source destination will be copied in the receiver destination

Rsync Option description

- -a, --archive (tells sync directories recursively, transfer special and block devices, preserve symbolic links, modification times, groups, ownership, and permissions.
- -z, --compress (used to compress the data if connection is slow)

- -P, --partial progress (tells progress bar during transfer, used when transferring large file over a slow and unstable network.
- -e (to use ssh)
- -v --verbose output (Displays the details of the transfer.

Automate rsync:

Now we have the automate the process of rsync so that it can sync automatically without writing commands again and again

Step 1:

- Firstly create a script file in which you have to define the rsync command
 - First create a directory in Server and and create a script in that directory
- ```
vi script
```
- `#!/bin/sh`
  - `rsync -azz -e ssh /root/aps1/* root@65.0.7.1:/root/aps/`  
[the rsync takes file from /root/aps1 and sync it into the another(destination) server's /root/aps/ directory]
  - save and exit



```
root@ip-172-31-15-196:~/scripts
#!/bin/sh
rsync -azz -e ssh /root/aps1/* root@65.0.7.1:/root/aps/
```

- `chmod +x script`

### Step 2:

Now create cronjob

- `crontab -e`

- a vi file will appear
- write a comment #backup everyminute
- 
- \* \* \* \* \* /root/scripts/script (give the address of your script)cd

```
[root@ip-172-31-15-196 scripts]# crontab -l
#photos backup everytime
* * * * * /root/scripts/script
```

MIN HOUR DOM MON DOW CMD

\* \* \* \* \* Stars mean after every minute it will sync the data

| Field | Description  | Allowed Value               |
|-------|--------------|-----------------------------|
| MIN   | Minute field | 0 to 59                     |
| HOUR  | Hour field   | 0 to 23                     |
| DOM   | Day of Month | 1-31                        |
| MON   | Month field  | 1-12                        |
| DOW   | Day Of Week  | 0-6                         |
| CMD   | Command      | Any command to be executed. |