Five Lesser Known Useful SSH Features

- By Rahul Amaram

1. SSH CONFIG

SSH Common Options

- Some Common Options passed via ssh command line
  - Passing the identity file location
  - Passing the username

- Example
  - ssh -i ~/keys/gsg-keypair ubuntu@ec2-54-73-36-106.eu-west-1.compute.amazonaws.com

~/.ssh/config file

- Sepcify Options in ~/.ssh/config file

```
Host aerospike-testing
    HostName ec2-54-73-36-106.....
    User ubuntu
    IdentityFile ~/keys/gsg-keypair
```

- ssh **aerospike-testing**

Wildcards & More options

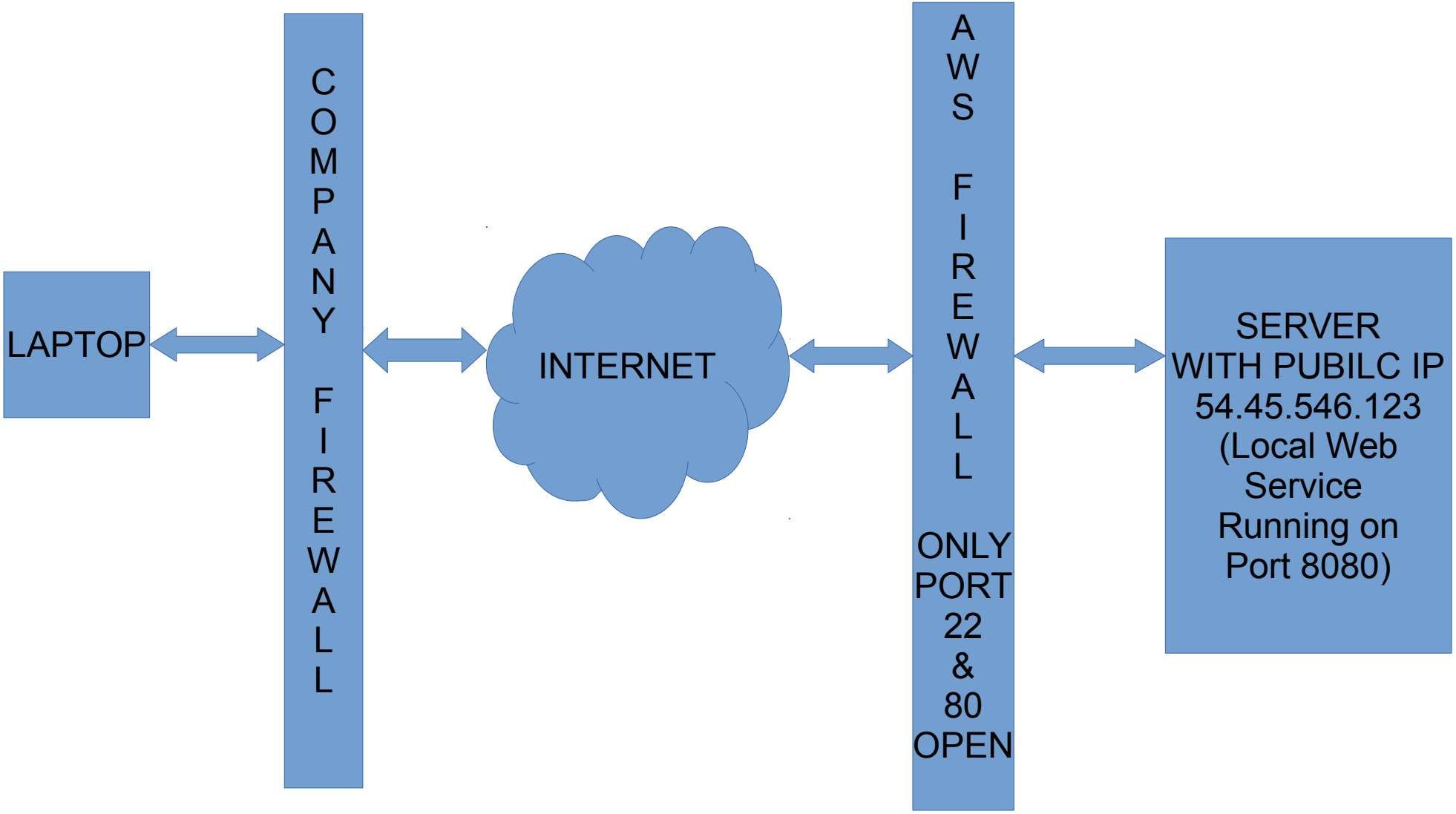
- Can also use wild cards. Example:

```
Host *.amazonaws.com
    User ubuntu
```

- List of all options - **man ssh_config**

2. Port Forwarding

Scenario 1.



How do you access the local web service?

General Method of Accessing

- Open Port in Firewall

- Limitations
  - Might not be possible to always have the port opened
  - Data between the laptop and the remote server travels unencrypted

Local Port Forwarding

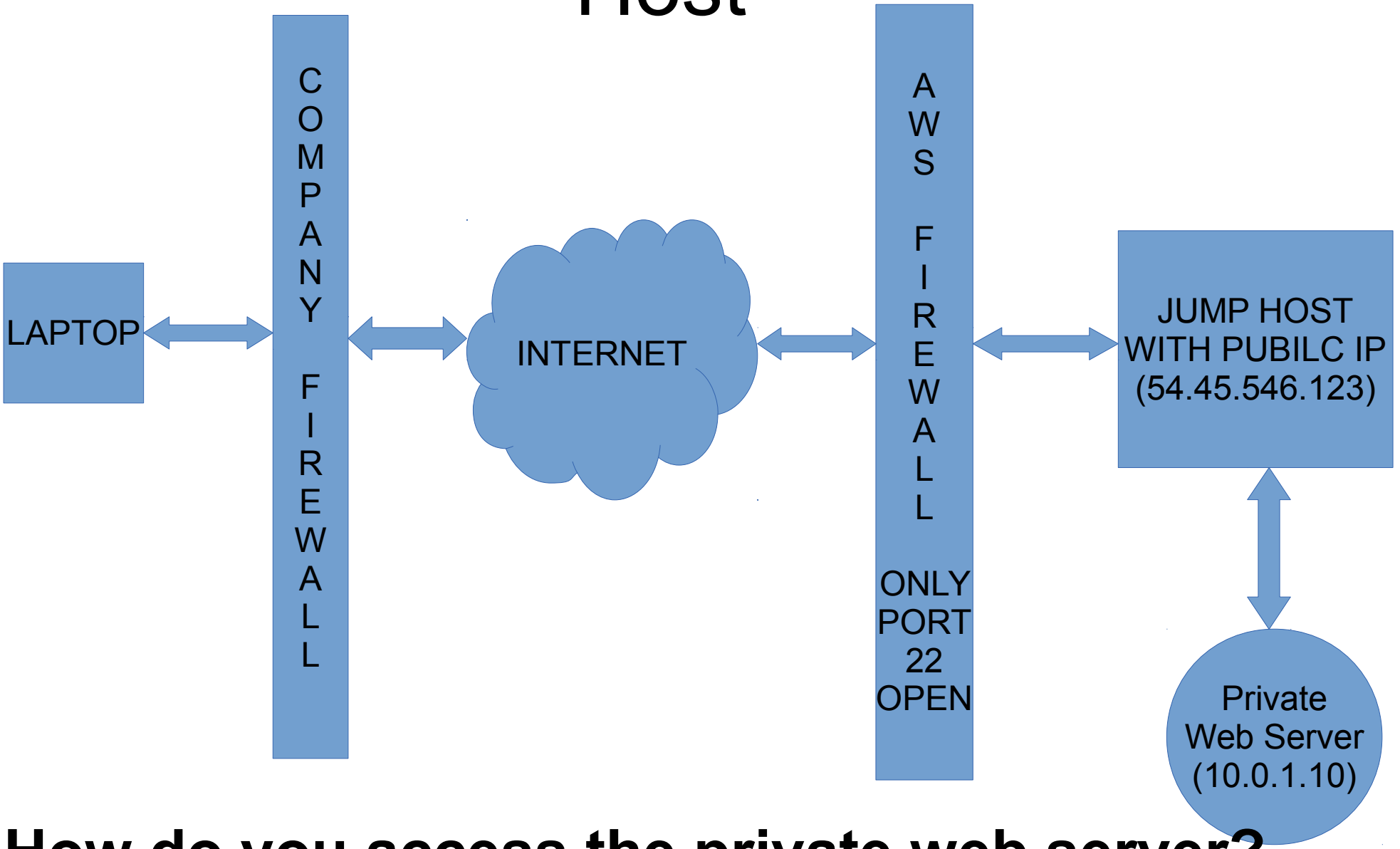
- ssh **-L <local-port>:<remote-dest-host>:<remote-dest-port>** <server-public-ip>

- Example:

  ssh **-L 8080:127.0.0.1:8080** ubuntu@54.45.546.123

- Open your browser, and acces http://localhost:8080/ . This connects to the server's port 8080.

Accessing Private Server from Jump Host



How do you access the private web server?

Local Port Forwarding to Private Server

- ssh **-L <local-port>:<private-server-ip>:<private-server-port>** <server-public-ip>

- Run from your laptop

  ssh **-L 8080:10.0.1.10:80** ubuntu@54.45.546.123

- Open your browser, and acces http://localhost:8080/. This connects to host 10.0.1.10 port 80.

Extends to any service

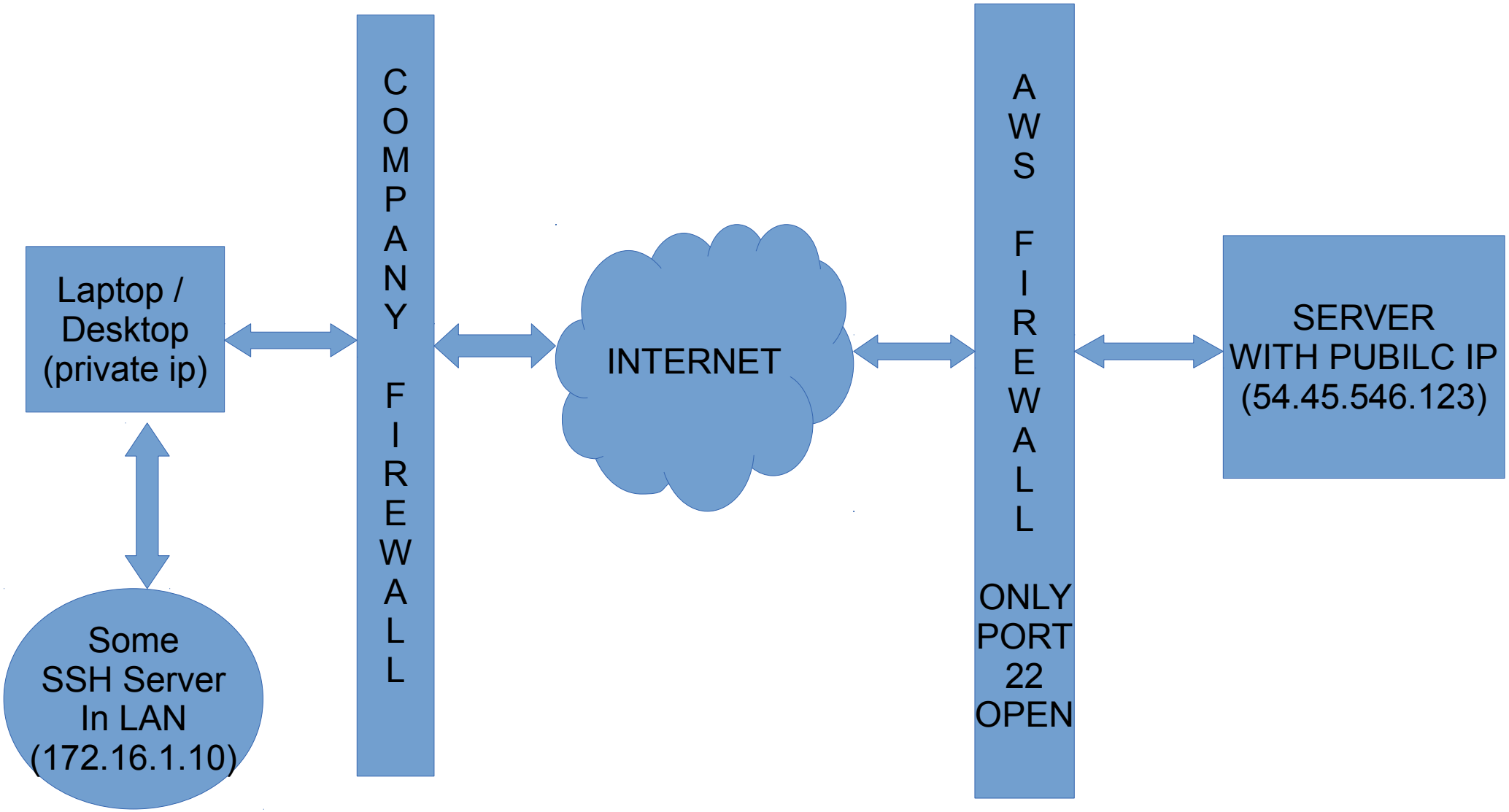
- ssh **-L 11211:10.0.1.10:11211** ubuntu@54.45.546.123

  telent **localhost 11211** -----> connects to 10.0.1.0 port 11211

- ssh **-L 2222:10.0.1.10:22** ubuntu@54.45.546.123

  ssh **-p 2222 localhost** ----> connects to 10.0.1.10 ssh service

Scenario 2.



How do you connect to SSH server from home?

Remote Port Forwarding

- ssh **-R <remote-port>:<private-machine-ip>:<private-machine-port>** <server-public-ip>

- Run from your laptop

  ssh **-R 2222:172.16.1.10:22** ubuntu@54.45.546.123

- From anywhere in the world, SSH to ubuntu@54.45.546.123 and then run:

  ssh **-p 2222 localhost** ---> This will connect to 172.16.1.10 ssh service

Scenario 3.

- Local Port Forwarding & Remote Port Forwarding Works On A port-to-port basis
- What if you don't know the ports beforehand? For example, while connecting to JMX service, the registry port can be defined, but it also opens additional ports which need to be accessed remotely.
- Or you want to access entire remote network via Jump Host

Dynamic Port Forwarding

- SSH Dynamic Port Forwarding setup up a local SOCKS proxy.

  ssh **-D 1080** ubuntu@54.45.546.123

- Open browser, go to Proxy settings and set it to **SOCKS Proxy 127.0.0.1:1080**.

- All websites visited by the browser are routed through the remote server ubuntu@54.45.546.123

3. SSH Agent

SSH Agent

- Recommended practice is to encrypt private keys
- Pain to decrypt them everytime
- Ssh Agent helps to store the decrypted private key in memory, thereby avoiding entering the password everytime.

SSH Agent Contd ...

- Can store multiple private keys, so that all of them are tried while trying to connect to a host.
- Command for adding the key - **ssh-add**
- Can also be added to GNOME Password Manager so that if your private key is encrypted with your login password, it is automatically decrypted when you login.

4. authorized_keys file

authorized_keys file

- Can be used to restrict the command which the user can run when authenticating with a particular private key

- Example entry in ~/.ssh/authorized_keys which allows user to only perform rsync in /home/ubuntu directory and enforces IP based restriction

  command="/usr/local/bin/rrsync -ro /home/ubuntu",from="192.168.1.10,172.16.0.5" ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAwzOUHRYyfW6KofxcXSVmxUUDMIDgFl4TlFV8LA2fboUERt+by1DekQMmsdHcaBtnf47Y0XLlh7MUTT9z2Y8dAVSt20grJgTUNAWJa2328RSCJ+8caxuvZtnf4kvArCcZFP7VM1FXlWY7OxcgSzOjMeGVXA3lQHeIKd2OxQn+BbPgK8ZTqT/8LB/8BUGYM3pFr9NPXLFz2X8jlmfbwZngqzfQ2ui2Wv2rHVKgz6rXrY+LdKKyfA2kXPNsYXW/Lo2vH1zTj3moS+Ga21ZMX5dCVF/uFlmND0c1biLv2FPzvPDWSMsrSLj6SpLToXCTBDIA22R1GioUK69Ykh4zYhmikw== backup

- More on authorized_keys options - **man authorized_keys**

5. X11 Forwarding

Running remote GUI Programs

- Using the X11 forwarding option of ssh, it is possible to run remote GUI programs on local machines.

- For example, for launching eclipse on remote machine but displaying it on your local machine, you can run:

  ssh **-X** user@host eclipse