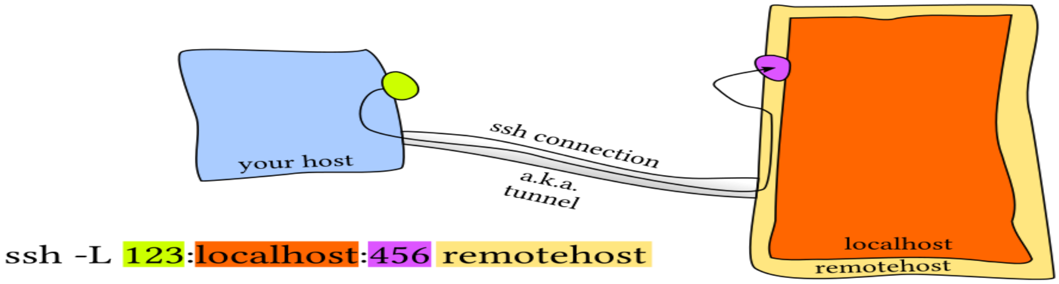
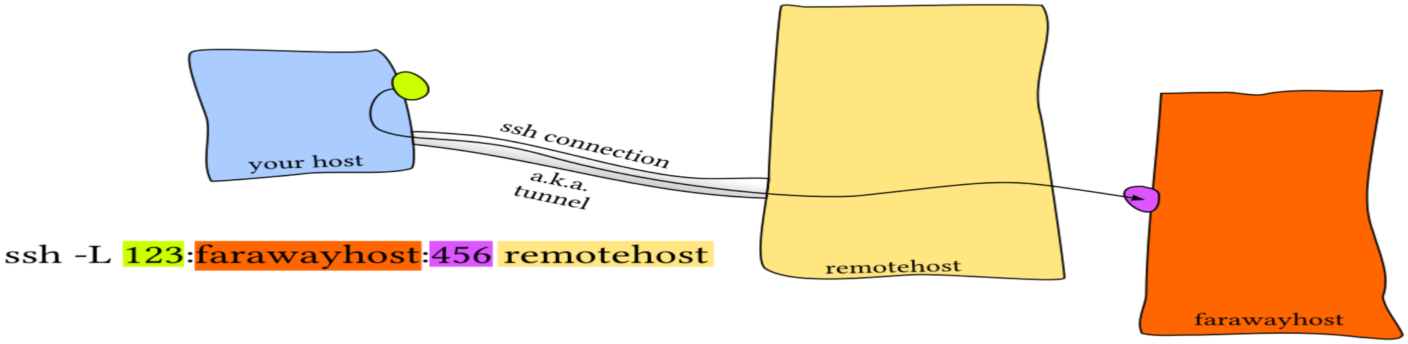
Reverse SSH Tunnel

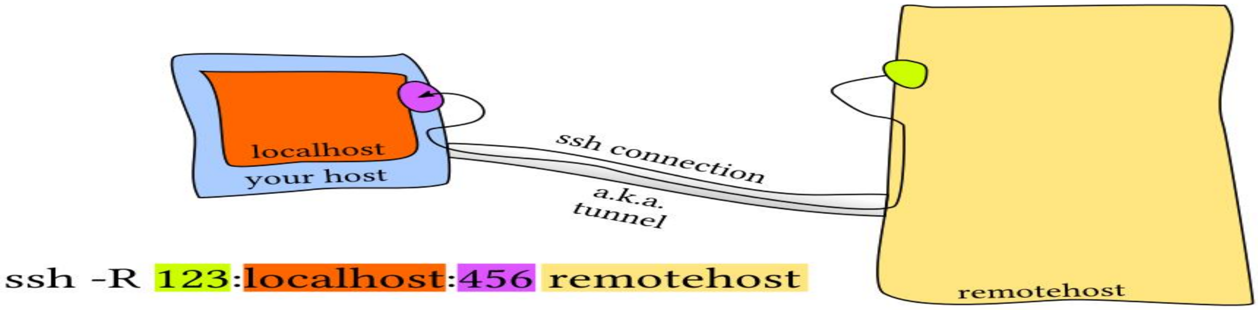
The machine, where the ssh tunnel command is typed is called **»your host«**.



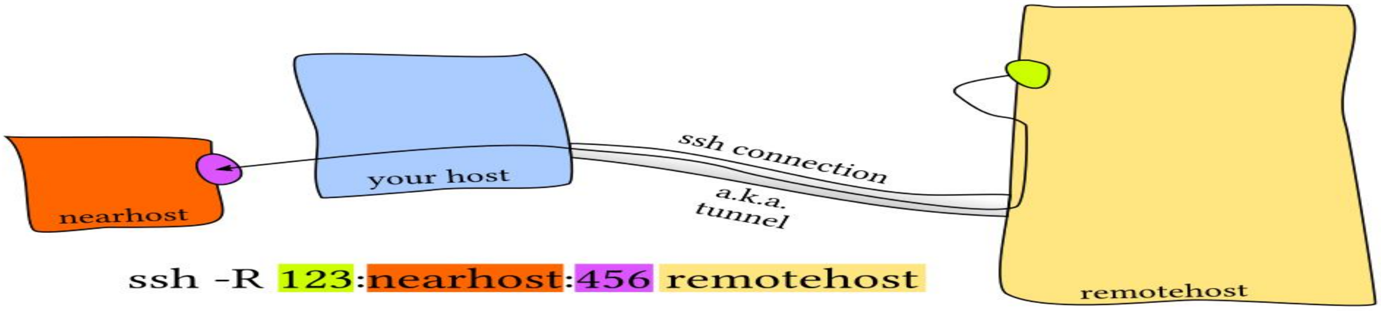
```
ssh -L 123:localhost:456 remotehost
```



```
ssh -L 123:farawayhost:456 remotehost
```



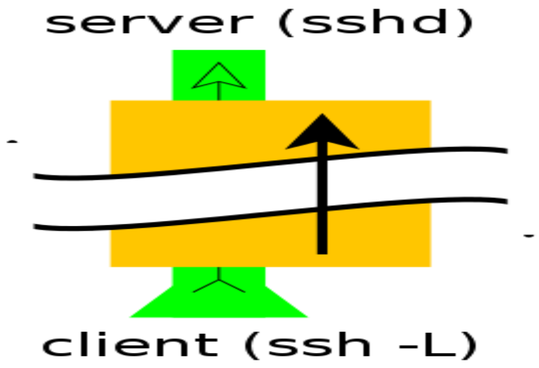
```
ssh -R 123:localhost:456 remotehost
```

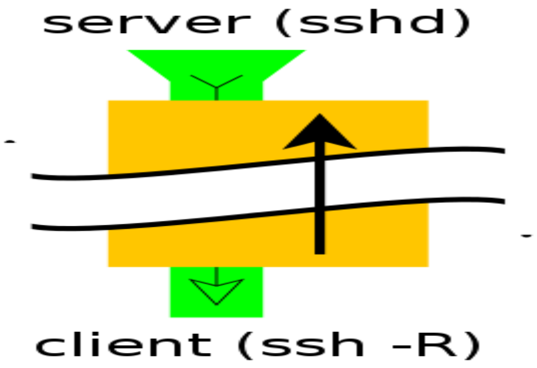


```
ssh -R 123:nearhost:456 remotehost
```

Introduction

1. local: `-L Specifies that the given port on the local (client) host is to be forwarded to the given host and port on the remote side.`
   `ssh -L sourcePort:forwardToHost:onPort connectToHost` means: connect with ssh to `connectToHost`, and forward all connection attempts to the **local** `sourcePort` to port `onPort` on the machine called `forwardToHost`, which can be reached from the `connectToHost` machine.
2. remote: `-R Specifies that the given port on the remote (server) host is to be forwarded to the given host and port on the local side.`
   `ssh -R sourcePort:forwardToHost:onPort connectToHost` means: connect with ssh to `connectToHost`, and forward all connection attempts to the **remote** `sourcePort` to port `onPort`on the machine called `forwardToHost`, which can be reached from your local machine.





Additional options

- `-f` tells ssh to background itself after it authenticates, so you don't have to sit around running something on the remote server for the tunnel to remain alive.
- `-N` says that you want an SSH connection, but you don't actually want to run any remote commands. If all you're creating is a tunnel, then including this option saves resources.
- `-T` disables pseudo-tty allocation, which is appropriate because you're not trying to create an interactive shell.

Your example

The third image represents this tunnel. **But** the blue computer called »your host« represents the computer where *someone* starts the ssh tunnel, in this case the firewalled machine.
So, ask *someone* to start a ssh tunnel connection to your machine. The command should basically look like

```
ssh -R 12345:localhost:22 YOURIP
```

Now the tunnel is opened. You can now connect via ssh to the firewalled machine through the tunnel with the command

```
ssh -p 12345 localhost
```

which will connect to your own `localhost` (your machine) on port `12345`, but port `12345` is forwarded through the tunnel to port 22 of the localhost of the firewalled computer (i.e. the firewalled computer itself).

http://unix.stackexchange.com/questions/46235/how-does-reverse-ssh-tunneling-work