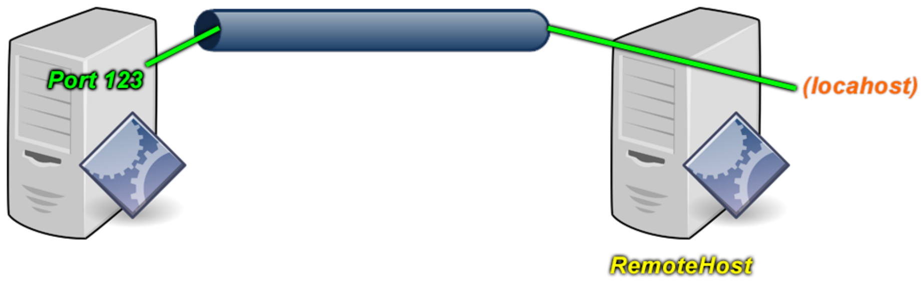
SSH LOCAL TUNNEL/ PORT FORWARD

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

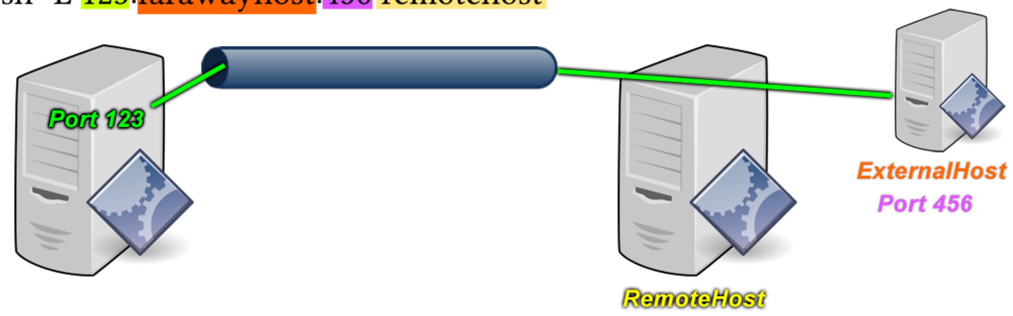


local: -L ---- given port on local (client) host to be forwarded to host, port on other side.

ssh -L sourcePort:forwardToHost:onPort connectToHost

Connect ssh to connectToHost, and forward all connection attempts to the local sourcePort to port onPort on forwardToHost, which can be reached from the connectToHost machine.

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

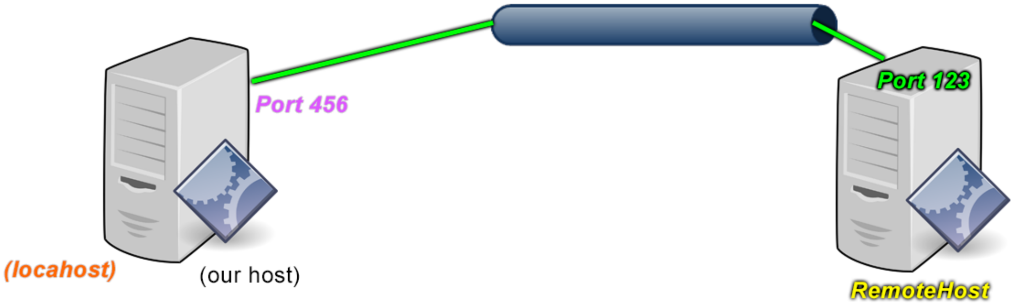


ssh -L 80:localhost:80 SUPERSERVER
You specify that a connection made to the local port 80 is to be forwarded to port 80 on SUPERSERVER. That means if someone connects to your computer with a webbrowser, he gets the response of the webserver running on SUPERSERVER. You, on your local machine, have no webserver running.

ssh -L 8080:127.0.0.1:80 user@webserver
Then in your browser on local use URL http://localhost:8080/
it will connect to local machines port 8080, which ssh will forward on to remote ssh, and it will then make a request to 127.0.0.1:80. Note 127.0.0.1 is actually the remote server's localhost, but it could have been a host/IP available at the remote machine's network.

SSH REMOTE/REVERSE TUNNEL/ PORT FORWARD

ssh -R 123:localhost:456 remotehost

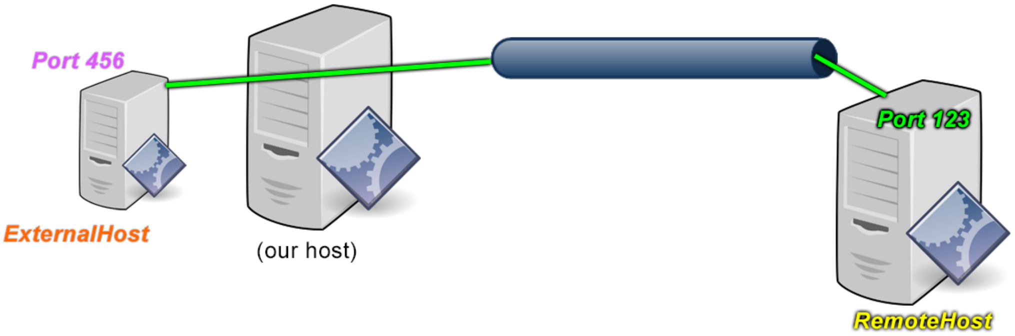


remote: -R - Port on remote host is to be forwarded to the given host and port on the local side.

ssh -R sourcePort:forwardToHost:onPort connectToHost

Connect with ssh to connectToHost, forward all connection attempts to remote sourcePort to port onPort on the machine forwardToHost, which can be reached from your local machine.

ssh -R 123:nearhost:456 remotehost



ssh -R 80:localhost:80 tinyserver

You specify that a connection made to the port 80 of tinyserver is to be forwarded to port 80 on your local machine. That means if someone connects to the small and slow server with a webbrowser, he gets the response of the webserver running on your local machine. The tinyserver, which has not enough diskspace for the big website, has no webserver running. But people connecting to tinyserver think so.

Other things could be: The powerful machine has five webservers running on five different ports. If a user connects to one of the five tinyservers at port 80 with his webbrowser, the request is redirected to the corresponding webserver running on the powerful machine. That would be

```
ssh -R 80:localhost:30180 tinyserver1
ssh -R 80:localhost:30280 tinyserver2
etc.
```

Or maybe your machine is only the connection between the powerful and the small servers. Then it would be (for one of the tinyservers that play to have their own webservers):

```
ssh -R 80:SUPERSERVER:30180 tinyserver1
ssh -R 80:SUPERSERVER:30280 tinyserver2
etc
```

ssh -R 10123:127.0.0.1:123 user@webserver

Asks ssh to create a listening port on the remote machine which it will forward back (reverse) to the local ssh to forward on. So, after ssh connects to webserver, the remote ssh creates and listens on a port 10123. A process on webserver connecting to 10123, ssh will pick it up and send it back to the local machine's ssh, which sends it on to 127.0.01:123 port.