**SSH 端口转发**

ssh 端口转发需要配置 /etc/ssh/sshd\_config 中tcp转发选项为开启：

AllowTcpForwarding yes

默认是开启被注释掉的，可以去掉注释保存之后重启 sshd 服务。service  sshd restart （systemctl restart sshd）.

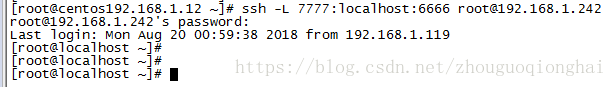
否则，来连接时，会出现 **channel 3: open failed: administratively prohibited: open failed** 错误。

**一、本地端口转发。**

本地端口转发命令： ssh -L 本地地址：本地端口：目的地址：目的端口 跳转机地址。

本地地址可以省略，这时本地侦听的是 127.0.0.1本地环回地址，只有本机上的进程能够连接该本地端口。

本机 192.168.1.12 上执行：ssh -L 7777:**localhost**:6666 root@192.168.1.242，将本机的 7777 端口转发到目标机器 192.168.1.242 的6666上。



这样会从 1.12 本机上登录到 1.242 目标机器上，保持登录状态。

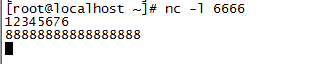
1.12 本机上查看 7777 端口的状态：

[root@centos192.168.1.12 ~]# ss -apn | grep 7777

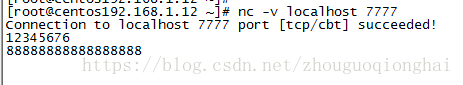
https://img-blog.csdn.net/20180820132307500?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

可以看到，7777 端口被ssh侦听。

在目标机 1.242 器上使用 netcat 侦听 6666 端口：

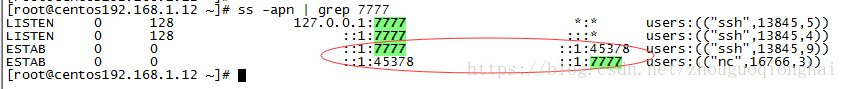


 1.12 本机上连接本机 7777 端口，连接之后发送文本消息。



发往1.12本机的7777端口的消息被 1.242 目标机器的 6666 端口接收到， 即就是完成了端口的转发。

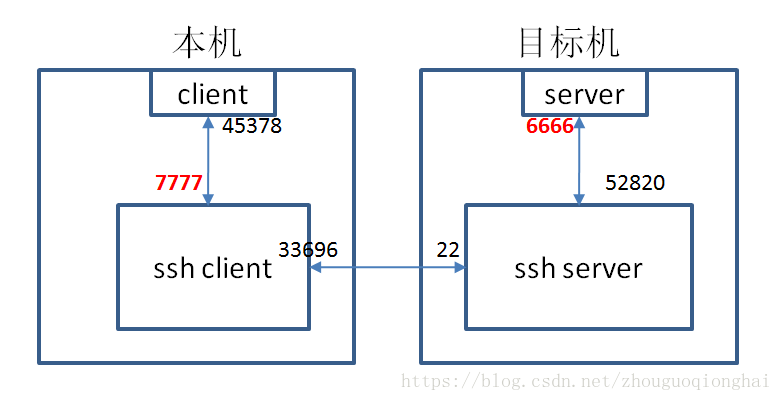
此时，查看1.12 本机的 7777 端口状态：



同时，在 1.242 目标主机上查看 6666 端口的状态：

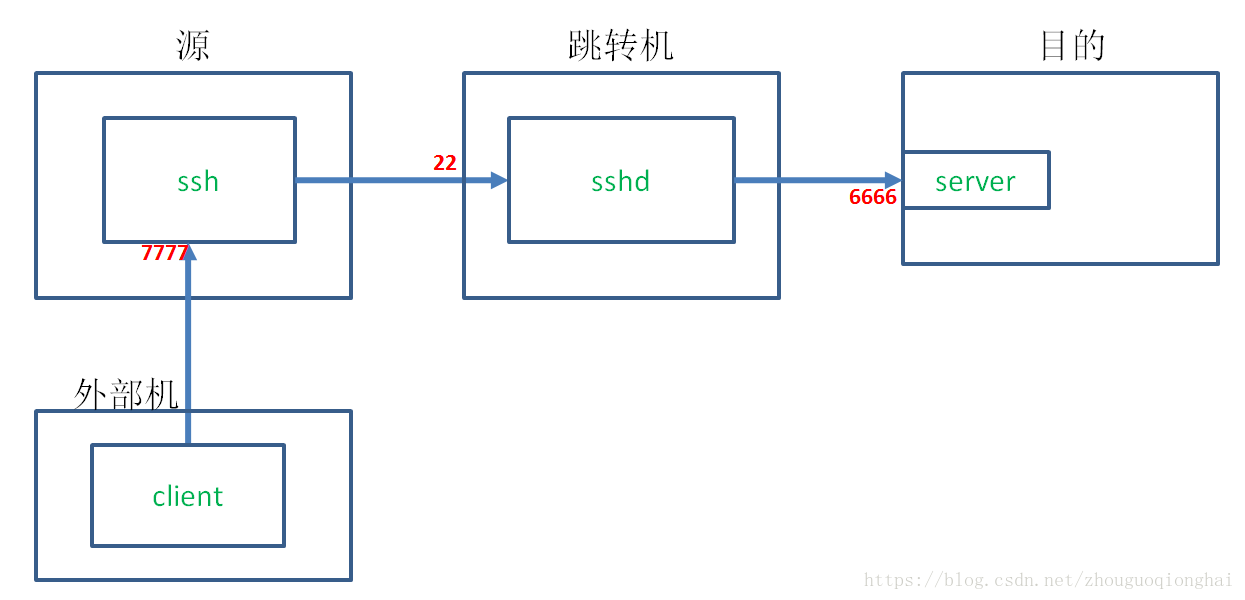
https://img-blog.csdn.net/20180820132821638?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

 可以看到，无论是在本机还是目标机器上，都建立了一条 ssh 与转发前的端口（本机上）和一条与转发后的端口之间的 TCP 连接（准确的说是在目标机上 sshd 与转发后的端口之间）。



注意：本地地址是相对于本机，目标地址是相对于提供ssh服务器的机器，所以上边的localhost其实是对 192.168.1.242 机器而言。

**本地转发更为一般的例子，将 ssh 服务机作为跳转机，使得原本不能被本机访问的目标机通过跳转机从而能够被访问。实现源到目的的转发。**



图中箭头方向仅仅表示连接的建立方向。

本机（源）：192.168.1.237                  提供转发前的原始端口

跳转机：192.168.1.242               ssh 服务器提供机

目标机：192.168.1.236                提供转发后的目的端口

外部机器 192.168.1.12                 访问原始端口

**1、**在本机上执行端口转发 ssh -L 0.0.0.0:7777:192.168.1.236:6666 root@192.168.1.242

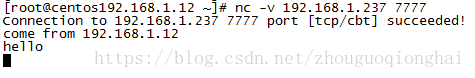
https://img-blog.csdn.net/20180821190145742?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

地址中使用了 0，表示任意的地址，这样其他机器也能连上 1.237 机器上被转发的端口。

**2、**在 1.236 目标机上侦听 6666 端口（转发之后的端口）。

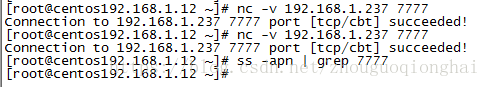
https://img-blog.csdn.net/20180821190316731?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

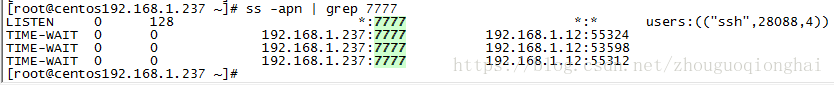
**3、**使用外部机器 1.12 连接 1.237 本机的 7777 端口，并发送内容：



成功建立连接，内容被正常发往目标机 1.236

假设 1.236 目标机没有侦听端口：





 使用外部 1.12 机器能连上 1.237 本机的 7777 端口，但是该连接马上被本机断开。从 1.237 本机的 TIME\_WAIT 状态看出是本机主动断开连接。同时，登录的 ssh 上会出现连接被拒。（ssh服务机尝试连接目标机的失败提示）

https://img-blog.csdn.net/20180821191200117?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

 4、假设在目标机 1.236 的防火墙上开启对跳转机 1.242 的屏蔽：

4.1、在目标机1.236 的 iptables 中删除对跳转机 1.242 的允许。

https://img-blog.csdn.net/20180821191355598?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

4.2、此时在外部机器 1.12 上，尝试连接 1.237 本机原始端口，连接将一直保持。

https://img-blog.csdn.net/20180821191418714?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

4.3、1.237 本地机器上查看与外部机 1.12 之间的连接状态。

https://img-blog.csdn.net/20180821213337297?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

4.4、在 1.242 跳转机上查看连接目标机的状态为建立连接的第一步 SYN-SENT，直到超时而终。

 ：https://img-blog.csdn.net/20180821191455180?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

二 、远端端口转发

如果源不能被跳转机访问，但是跳转机能够被源访问，这个时候就可以使用远端转发，将源转发到目的。

远端转发命令：

ssh -R 源端口：目的地址：目的端口 源地址（在跳转机上执行）。

源地址  192.168.1.242

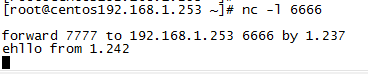
跳转地址 192.168.1.237

目的地址 192.168.1.253

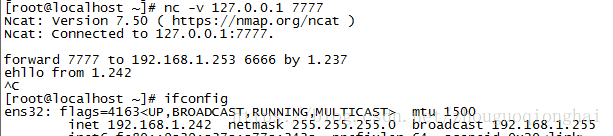
1、跳转机 1.237 上执行远端转发命令：

https://img-blog.csdn.net/20180822114547118?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

2、 目标主机侦听目标端口：

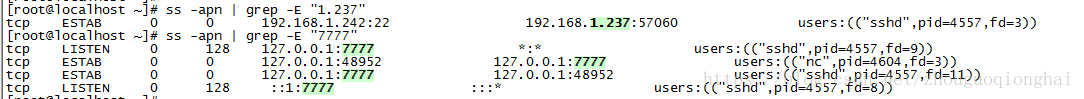


3、源主机上连接源端口，并发送消息。



 消息在目标机上被接收显示。

4、源机的连接信息。



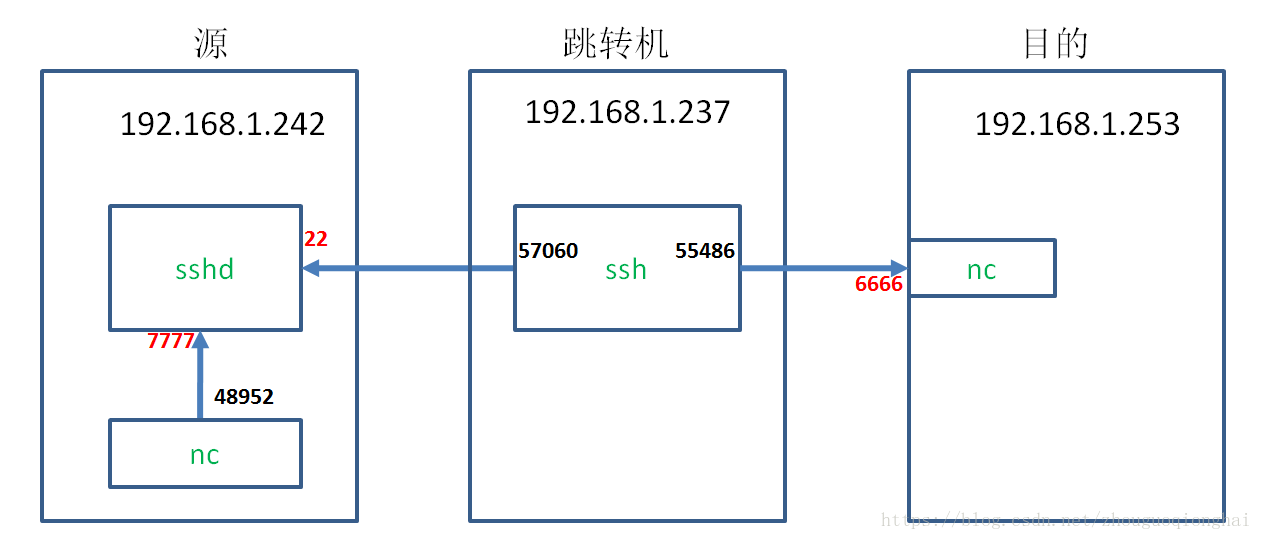
5、跳转转机的连接信息：

https://img-blog.csdn.net/20180822114915268?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

6、目的机的连接信息：

https://img-blog.csdn.net/20180822114942886?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

整个操作的示意图：

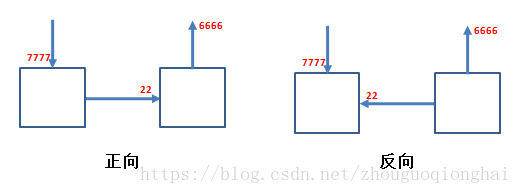


图中**箭头方向表示连接的建立方向**， TCP 是双工的，一旦建立连接当然可以双向的发送信息。

**三、总结**

1、本地端口转发在于ssh建立连接的方向与转发的方向一致，远端端口转发在于建立连接的方向与转发的方向相反，

看到网上有人分别叫正向转发与反向转发。



2、目的地址是相对于跳转机而言的。目的机和跳转机可以为同一台机，这是目的地址写 localhost 的原由。

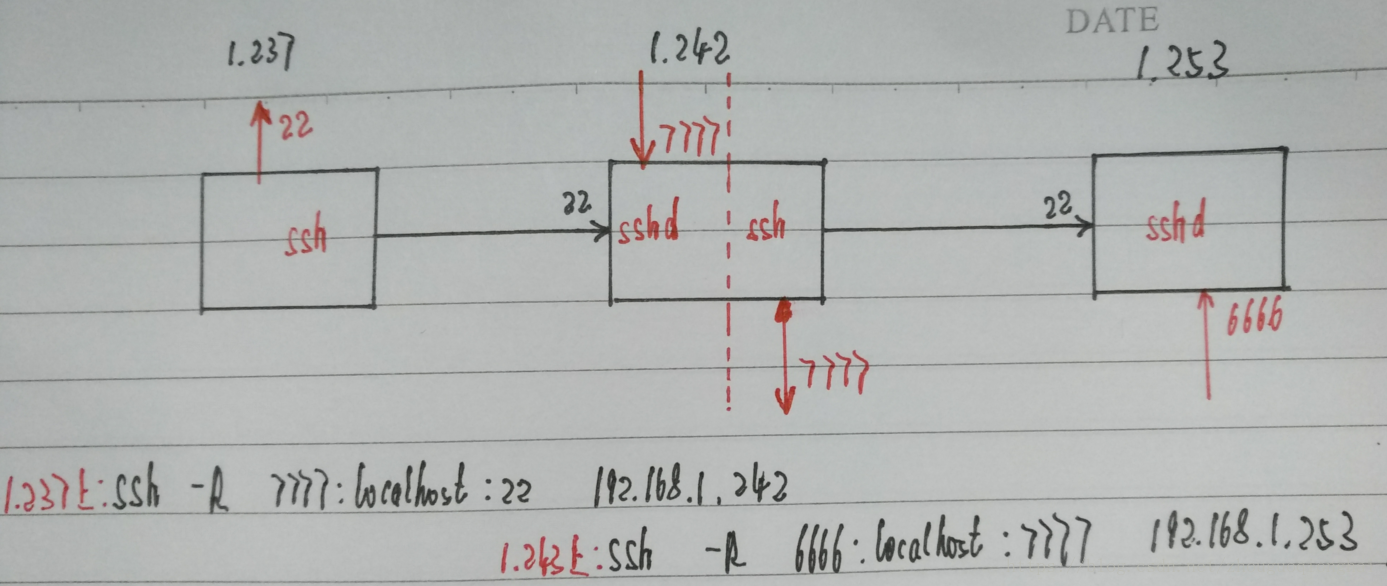
3、由于 ssh 和 sshd 的转发介入，原来的 1 条连接变成 3 连接。首和尾的连接发起方向必然是一致的，就是转发的方向，中间连接的发起方向相同，则是本地端口转发；不同，则是远端端口转发。

4、假如 A 能 ssh 到 B 上，反向却不行。可借助远端转发，将 B 的某个端口转发到 A 的 22 号端口上，从而实现 B 上间接 ssh 登录 A。

在 A 上执行： ssh -R 6666:localhost:22 root@B，之后在 B 上便可 ssh -p 6666 127.0.0.1 来登录 A机器。

假如 1.237 能通过 ssh 访问 1.242，并且1.242 能通过ssh访问 1.253。不同机器之间其他的端口全部被防火墙禁止。如何实现在1.253 上 ssh 登录1.237 呢？

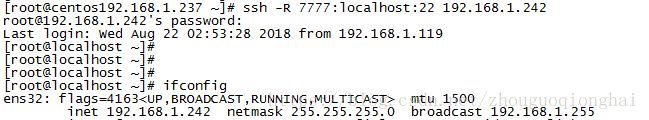
问题转化为将 1.253 上的 6666 端口转发到 1.237 的 22 端口上，在图上标注转发的方向。（1.253 6666端口入，1.237 22端口出）



 a、在 1.237 上，进行远端端口转发，将 1.242 的 7777 端口转发到 1.237 的 22 端口上。（1.242 7777 端口入，1.237 22端口出）

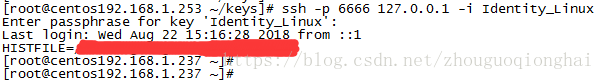
https://img-blog.csdn.net/20180822155639775?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3pob3VndW9xaW9uZ2hhaQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

 b、在 1.242 上，进行远端端口转发，将 1.253 的 6666 端口转发到 1.242 的 7777 端口上。（1.253 6666 端口入，1.242 777端口出）



 结合在一起，刚好 1.253 6666 端口入，1.242 7777 端口出再从 7777 端口入，最终从 1.237 22端口出。

在 1.253 上，尝试 ssh 登录，登录 1.237 成功。



 在1.237上执行 ssh -R 7777:localhost:22 192.168.1.242 远端端口转发之后，192.168.1.242 机器便开始侦听 7777 端口。一旦 7777 端口建立连接之后， 便通知 1.237 尝试去连接本地的22端口。在 1.253 上 ssh 登录 6666 端口时，由于端口转发，会1.242 会去尝试连接本机的 7777 端口。而本机的 7777 端口早处于侦听状态，整个链路便联通了。源端口有连接发生时，ssh/sshd 才会去连接转发之后的目的端口。