SSH端口转发(port forwarding)基础知识

原创 胡小白的数据科学之路 于 2021-07-08 12:31:28 发布 2190 收藏 4
分类专栏： SSH 文章标签： ssh shell

1.介绍

我们知道 SSH 的一个重要作用就是连接远程机器(比如一台服务器)来实现对远程机器的操作。

但是除此之外，SSH还可以用于构建机器之间的**通信中介**。

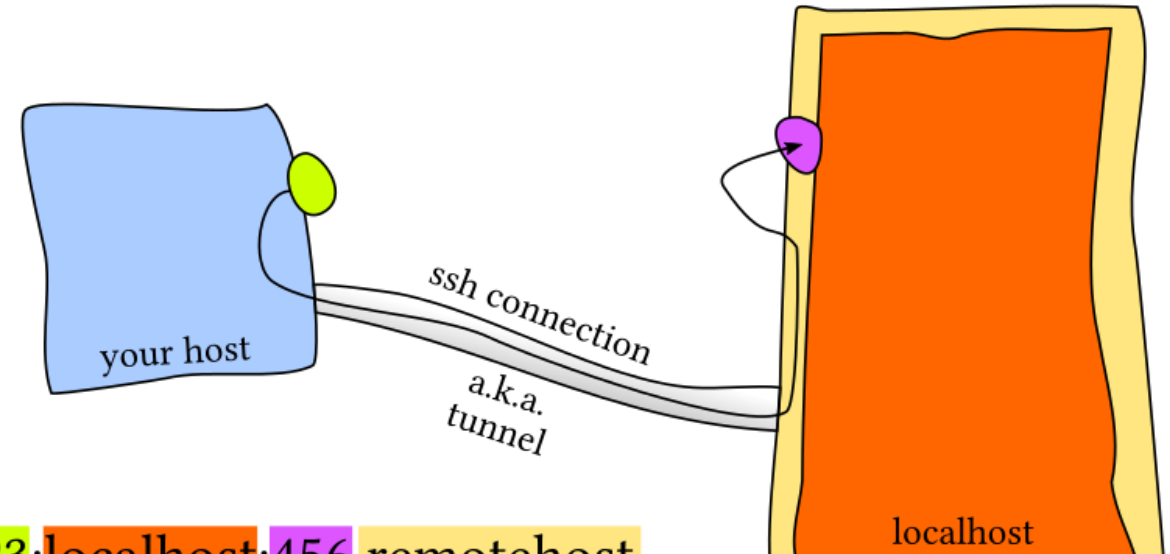
举个简单的例子，我们有时会在本地(localhost)运行某些程序或应用，而每个程序或者应用都会运行在一个指定的端口下(port), 我们可以直接通过 监听 (listen)该端口来查看其他用户向该应用发出的请求(request)并做出响应(response)。但是如果我们希望对本地端口的请求，由远程机器来进行响应，换句话说，**我们希望能将发送给本地端口的请求转发给远程机器**，又该怎么做呢？一种做法就是使用**SSH端口转发**(也叫做**SSH隧道**)。

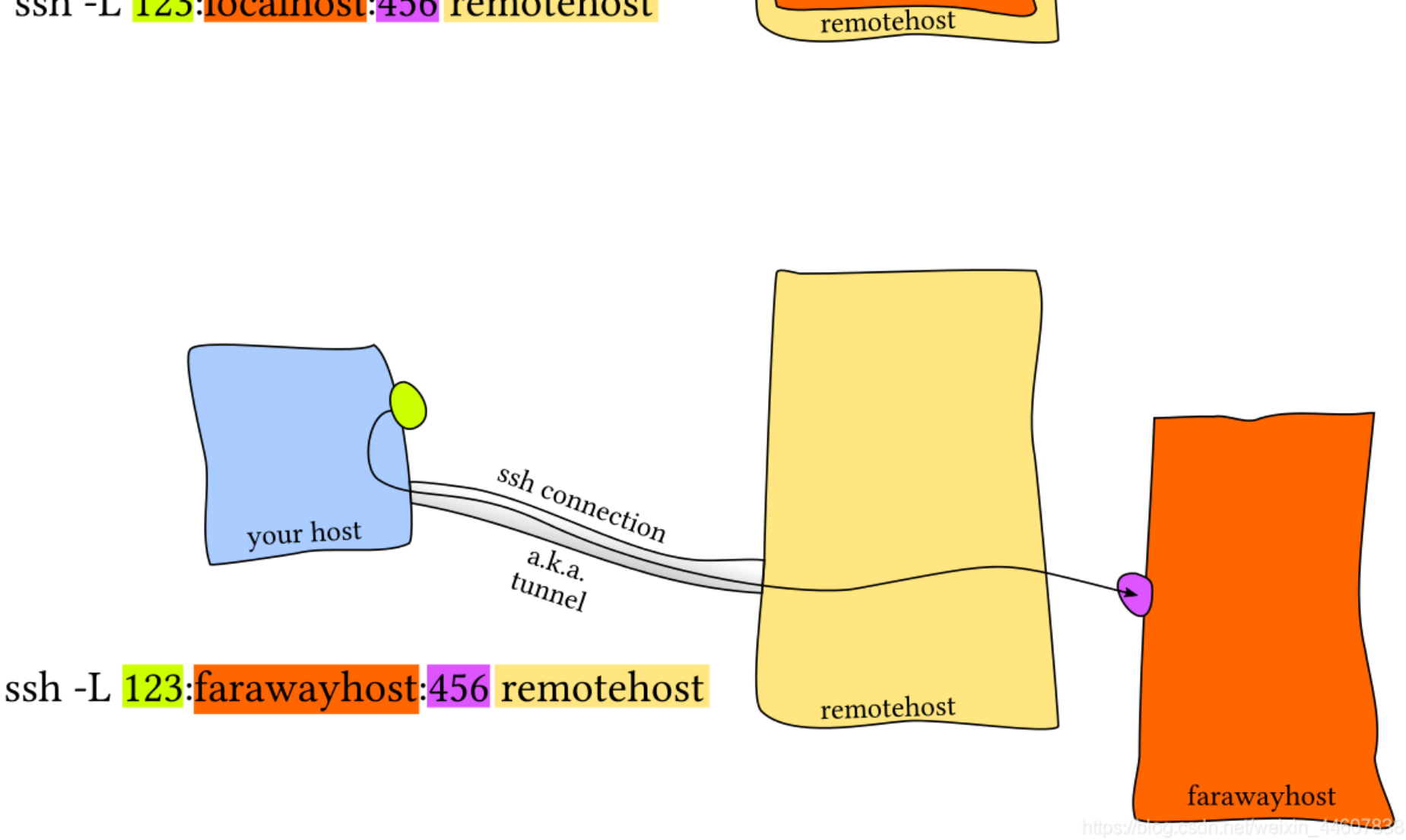
事实上上面我描述的这种 端口转发 是**本地转发(local port forwarding)**，也就是说我希望发送给我本地端口的请求转发到远程机器上。那么反过来，另外一种转发方式就是**(remote port forwarding)**，也就是说远程机器监听请求并将请求转发给本地机器来做出响应。下面我来具体介绍一下这两种转发方式。

2.本地转发

本地转发(local port forwarding)概念

首先先放一张图，此图来源是StackOverflow的某个帖子,觉得很棒就直接搬运过来了。





在本地转发的情况下，本地机器(上图中的your host)是**ssh的客户端**，同时也是**应用的客户端**(监听请求),而隧道另一头的远程机器既是**ssh的服务端**,也是**应用的服务端**(上图的情况1)或**跳板机**(情况2)。

关于如何具体建立本地转发的终端指令，上图中也非常清楚的展示了，这里再做一个简单说明。

1.如果目标机器(希望获得响应的host)就是我们SSH隧道的终点机器的话，就对应着上图中的第一部分。

```
$ ssh -L 123:localhost:456 remote-host
```

这里的L代表 Local的形式，而123代表本机的123端口(监听请求), localhost:456代表的是远程机器的456端口，这里要注意的是**localhost并不是指我们的本地主机，而是远程机器的localhost**. 最后一部分就是登录远程机器的信息。

2.如果目标机器(希望获得响应的host)并不是我们SSH隧道的终点，我们建立SSH隧道的终点只是一个SSH跳板机时，就对应着上图中的第二部分。

```
$ ssh -L 123:target-host:456 remote-host
```

本地转发(local port forwarding)应用

相对来说本地转发是比较常见的一种情况，比如我们在做一些机器学习或者深度学习任务时，当我们想要获取更快的训练速度时，会希望使用性能更好的远程服务器而不是本地主机来运行代码。比如我们使用jupyter notebook编写代码时，就可以将Jupyter notebook运行在服务器上，然后本地来进行操作。

这个时候，我们就可以使用local forwarding来实现了，假设我们已经在服务器上开启了Jupyter Notebook应用，端口号为8888，这个时候我们希望从本地主机的9999端口来建立隧道。就可以使用如下命令。

```
$ ssh -L 9999:localhost:8888 user@remote_server
```

当然，最后一部分的user@remote_server就是指定服务器的账户。接下来我们就可以在本机上进入localhost:9999来进行代码的编写了，而运行任务会转发给远程服务器。

除此之外，另一个简单的应用就是vpn,我们可以借助本地转发来实现一个穷人版的vpn假如我们想访问某个内网的服务器，端口号为80，443，但是内网的服务器无法从外网直接访问，但是如果网络管理者为我们开放了一个bastion host提供ssh服务，那么我们就可以借助它作为我们的跳板机访问内部服务器。

```
$ ssh -L 8080:internal-server:80 -L 8443:internal-server:443 bastion-host -N
```

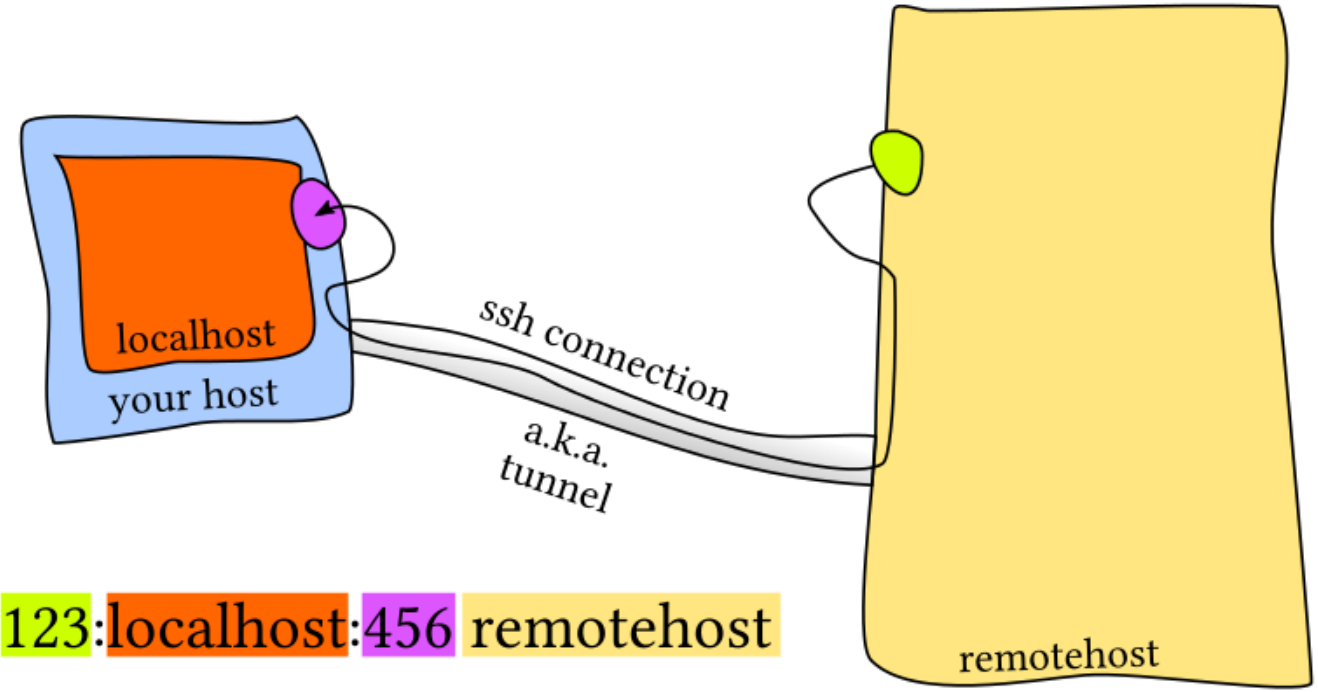
其中8080和8443分别为本机端口。

OK，接下来我们来进入远程转发的介绍。

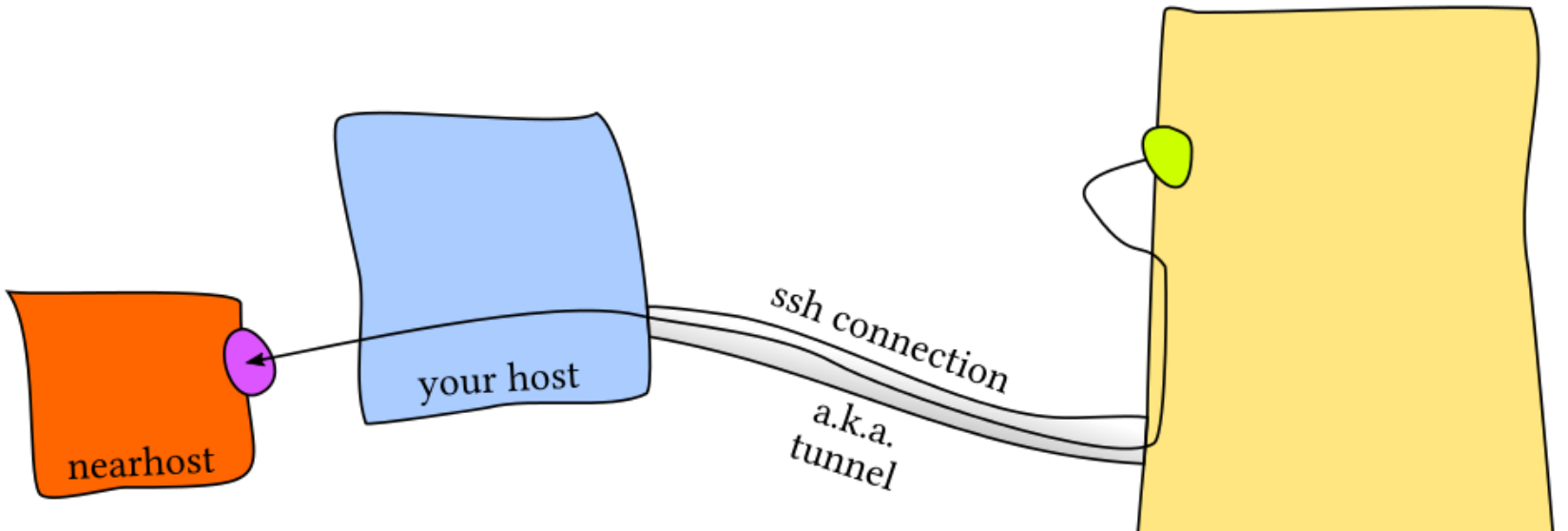
3.远程转发

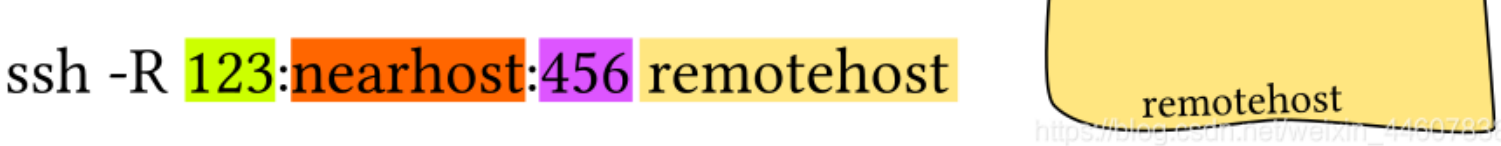
远程转发(remote port forwarding)概念

同样的，先上图。




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





在远程转发的情况下，本地主机(your host)同样作为ssh的客户端，远程主机(remotehost)同样是ssh的服务端，**但是与本地转发不同的是，这时远程主机才是应用的监听端，将请求转发给本地主机，本地主机是应用的服务端。**

关于具体的终端指令，上图中有清晰的说明，同时逻辑与本地转发十分类似，这里就不再重复说明了。这里主要来介绍一下远程转发的的应用例子。

远程转发(remote port forwarding)应用

事实上，远程转发主要针对的是内网的情况。通常情况下，当我们的本地主机处于外网，而目标机器或者目标机器与SSH的跳板机均位于内网中。那么我们是无法简单通过本地转发连接到内网中的目标主机的。**这时我们只能反过来通过SSH跳板机来发起隧道，即让远端的机器（我们的本地主机）来进行端口转发。**假设我的本地主机端口为2080，内网中的服务端口为80，此时在**SSH跳板机**应该输入如下指令

```
$ ssh -R 2080:target-server:80 local-host
```

此时，本地机器lcoal-host作为ssh的服务端，必须要安装SSH服务器，才能接受跳板机的远程登录。

4.小总结与额外补充

事实上，本地转发与远程转发的核心区别就在于发起转发请求的host究竟需要扮演什么角色。对于本地转发而言，发起转发请求的host就是服务的监听端，本身却不是服务端响应端，而对于远程转发而言，发起转发请求的host就是真正的服务响应端(除非如上面的例子，作为SSH跳板机)。

同时我们在进行转发命令时，通常会有一些基本的选项，比如-N和-f.

-N:代表这个 SSH 连接只进行端口转发，不登录远程 Shell，不能执行远程命令，只能充当隧道。

-f:后台运行SSH隧道，即使我们关闭了创建隧道时所使用的SSH会话，对应的SSH隧道也不会消失。

OK,以上就是这篇文章的全部内容，谢谢大家的阅读。

参考:

1.SSH端口转发

2.An Illustrated Guide to SSH Tunnels

3.ssh端口转发：ssh隧道

4.MIT-Missing Semester-Command_line_environment