** 1. 网络环境搭建**

** 2. 建立VPN隧道：客户端到内网主机，内网主机到客户端**

** 加密隧道**

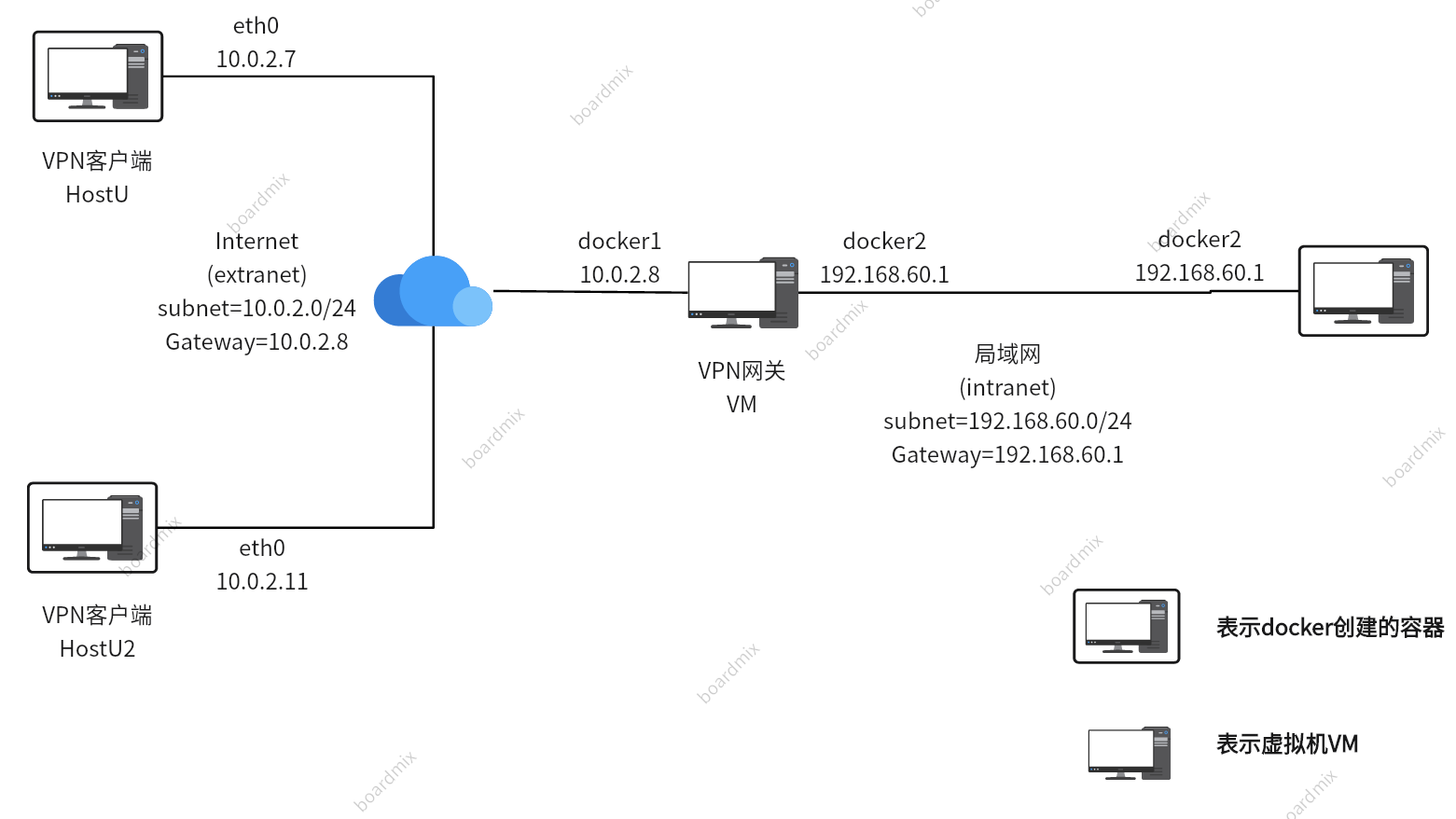
** 身份认证**

** 多客户端支持**

**检查单**

|  |  |  |
| --- | --- | --- |
| **认证VPN服务器** | **用openssl检查VPN服务器证书信息** | **证书主题包含个人信息** |
| **修改VPN客户端主机时间到VPN服务器证书有效期之后再登录VPN服务器** | **VPN客户端提示证书过期** |
| **认证VPN客户端** | **VPN客户端以错误的用户名或口令登录VPN服务器** | **提示错误无法登录** |
| **VPN客户端以正确的用户名口令登录VPN服务器** | **能正确登录** |
| **加密隧道通信** | **VPN客户端ping内网主机，wireshark在VPN服务器外口截包检查** | **能通信** |
| **经隧道封装** |
| **隧道为TLS** |
| **支持多客户端** | **开启2个以上VPN客户端容器，同时登录VPN服务器，分别测试telnet通信** | **各自正常登录通信** |
| **断开其中一个VPN客户端，测试另外一个的隧道通信** | **隧道保持，通信不受影响** |
| **易用性和稳定性** | **VPN客户端虚拟IP获取** | **手动分配还是VPN服务器分配？** |
| **VPN客户端虚拟IP配置** | **手动配置还是程序自动添加？** |
| **VPN客户端内网路由配置** | **手动配置还是程序自动添加？** |
| **正常使用时的稳定性** | **运行稳定** |

网络拓扑

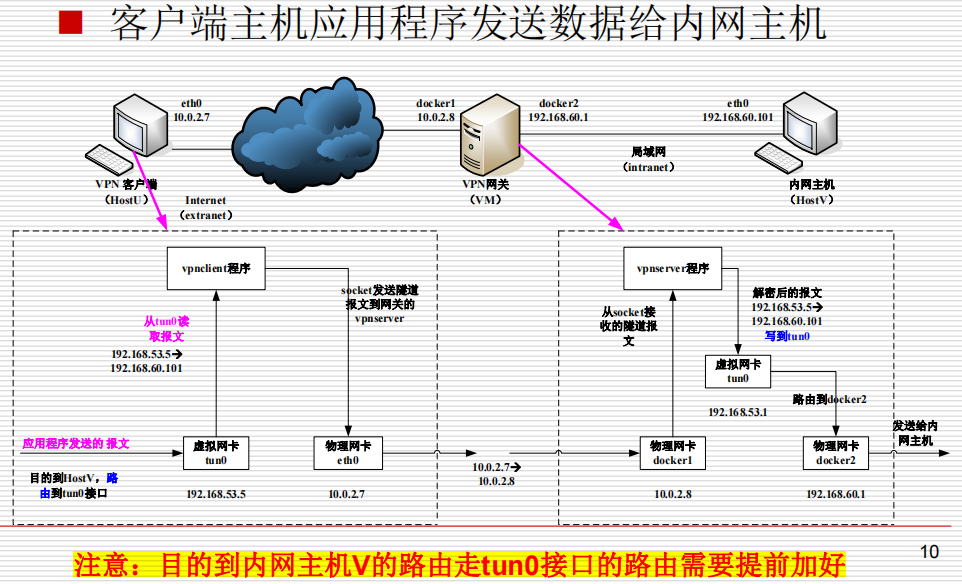


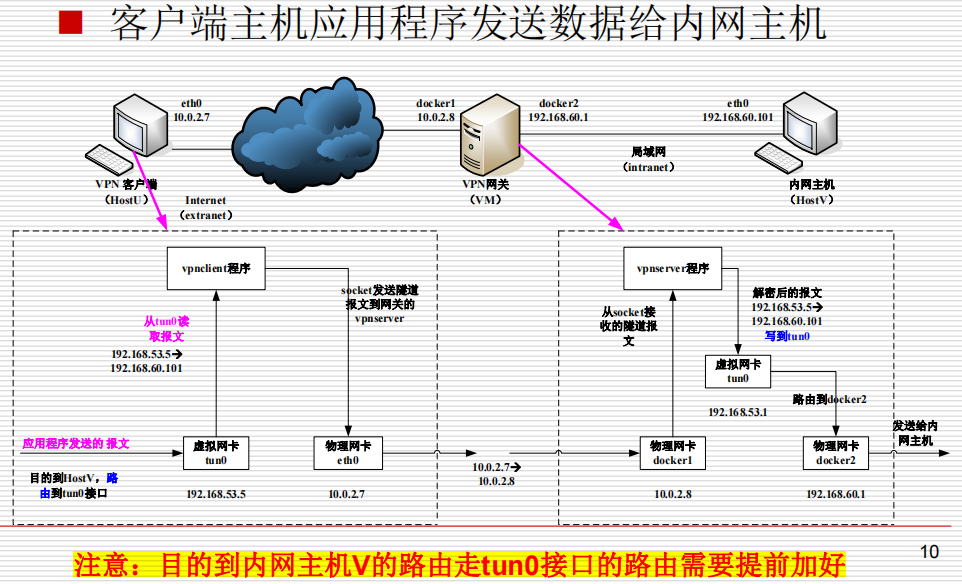
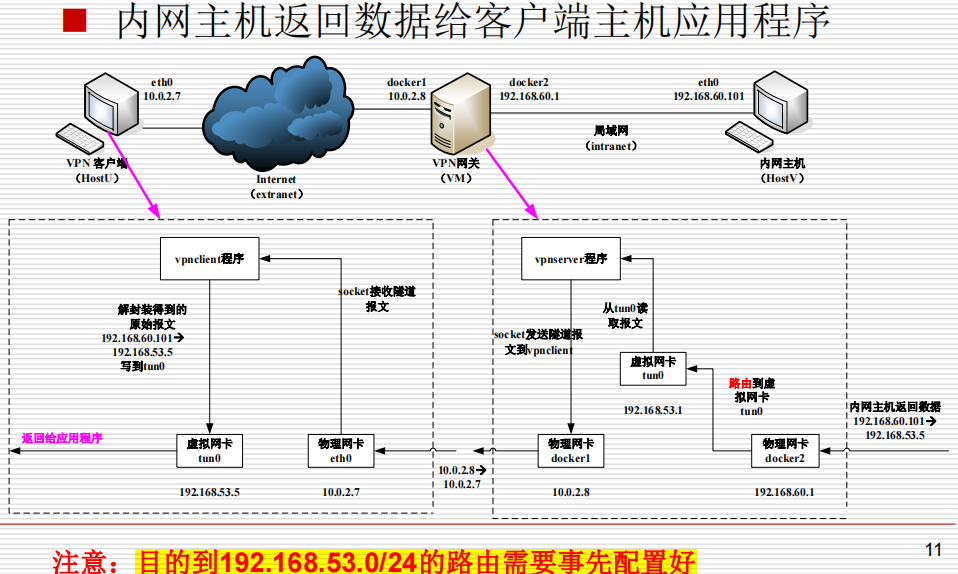
docker1是extranet，docker2是intranet

使用虚拟机本身作为VPN服务器网关（VM）

TLS/SSL VPN中使用了TUN/TAP技术，TUN和TAP是虚拟网络内核驱动程序，linux直接支

持 ◼ TAP模拟以太网设备，处理的是以太网帧等二层 数据包；TUN模拟网络层设备，处理的是IP等三层数据包 ◼ 我们可以用TAP/TUN创建虚拟网络接口。





隧道协议由UDP->TCP 在客户端和服务器之间的TCP连接上建立 TLS/SSL会话