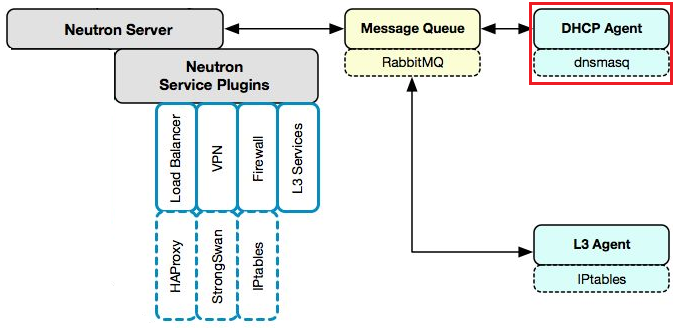


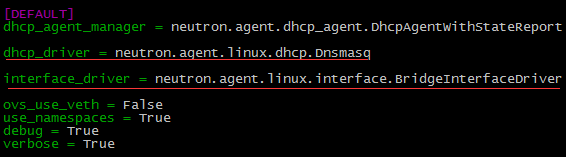
前面章节我们看到 instance 在启动过程中能够从 Neutron 的 DHCP 服务获得 IP，本节将详细讨论其内部实现机制。

Neutron 提供 DHCP 服务的组件是 DHCP agent。 DHCP agent 在网络节点运行上，默认通过 dnsmasq 实现 DHCP 功能。



**配置 DHCP agent**

DHCP agent 的配置文件位于 /etc/neutron/dhcp\_agent.ini。



**dhcp\_driver**使用 dnsmasq 实现 DHCP。

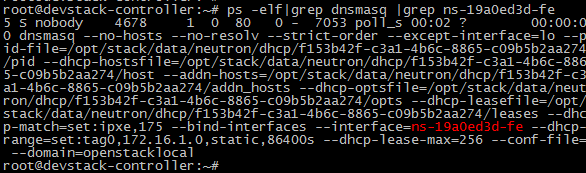
**interface\_driver**使用 linux bridge 连接 DHCP namespace interface。

当创建 network 并在 subnet 上 enable DHCP 时，网络节点上的 DHCP agent 会启动一个 dnsmasq 进程为该 network 提供 DHCP 服务。

dnsmasq 是一个提供 DHCP 和 DNS 服务的开源软件。

dnsmasq 与 network 是一对一关系，一个 dnsmasq 进程可以为同一 netowrk 中所有 enable 了 DHCP 的 subnet 提供服务。

回到我们的实验环境，之前创建了 flat\_net，并且在 subnet 上启用了 DHCP，执行 ps 查看 dnsmasq 进程，如下图所示：



DHCP agent 会为每个 network 创建一个目录 /opt/stack/data/neutron/dhcp/，用于存放该 network 的 dnsmasq 配置文件。

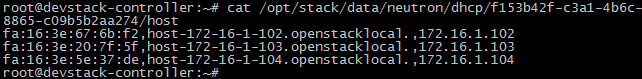
下面讨论 dnsmasq 重要的启动参数：

**--dhcp-hostsfile**存放 DHCP host 信息的文件，这里的 host 在我们这里实际上就是 instance。

dnsmasq 从该文件获取 host 的 IP 与 MAC 的对应关系。

每个 host 对应一个条目，信息来源于 Neutron 数据库。

对于 flat\_net，hostsfile 是 /opt/stack/data/neutron/dhcp/f153b42f-c3a1-4b6c-8865-c09b5b2aa274/host，记录了 DHCP，cirros-vm1 和 cirros-vm2 的 interface 信息。

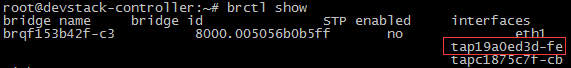


**--interface**指定提供 DHCP 服务的 interface。

dnsmasq 会在该 interface 上监听 instance 的 DHCP 请求。

对于 flat\_net，interface 是 ns-19a0ed3d-fe。

或许大家还记得，之前我们看到的 DHCP interface 叫 tap19a0ed3d-fe（如下图所示），并非 ns-19a0ed3d-fe。



从名称上看，ns-19a0ed3d-fe 和 tap19a0ed3d-fe 应该存在某种联系，但那是什么呢？

要回答这个问题，需要先搞懂一个概念：**Linux Network Namespace**，我们下一节详细讨论。