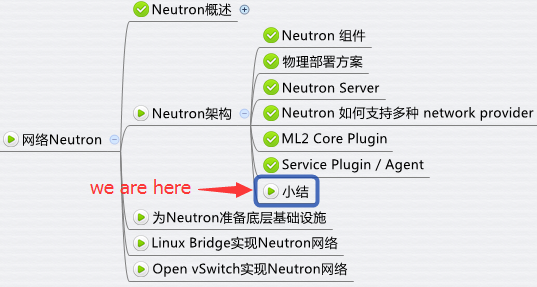
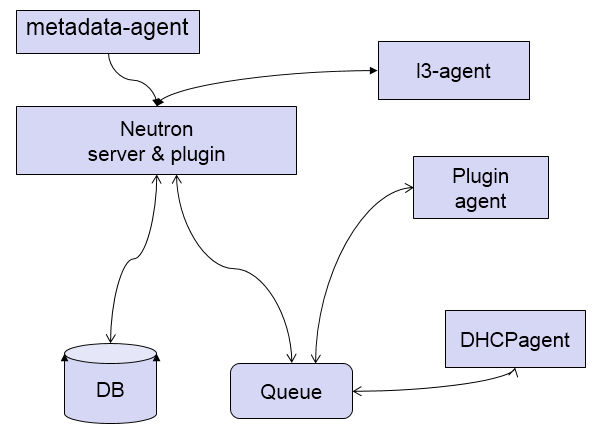
****

前面我们详细讨论了 Neutron 架构，包括 Neutron Server，Core 和 Service Agent。现在用两张图做个总结。  
先看第一张：



与 OpenStack 其他服务一样，Neutron 采用的是分布式架构，包括 Neutorn Server、各种 plugin/agent、database 和 message queue。

1. Neutron server 接收 api 请求。
2. plugin/agent 实现请求。
3. database 保存 neutron 网络状态。
4. message queue 实现组件之间通信。

metadata-agent 之前没有讲到，这里做个补充：

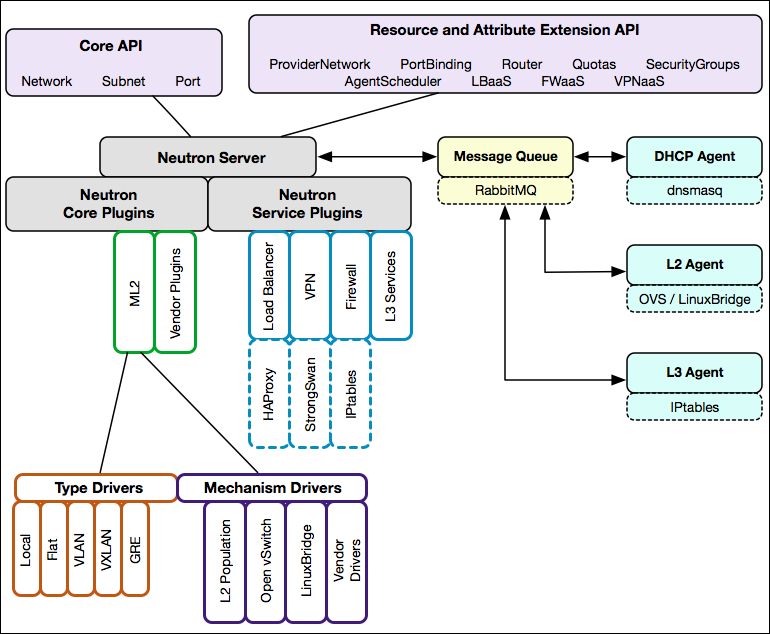
instance 在启动时需要访问 nova-metadata-api 服务获取 metadata 和 userdata，这些 data 是该 instance 的定制化信息，比如 hostname, ip， public key 等。

但 instance 启动时并没有 ip，如何能够通过网络访问到 nova-metadata-api 服务呢？

答案就是 neutron-metadata-agent

该 agent 让 instance 能够通过 dhcp-agent 或者 l3-agent 与 nova-metadata-api 通信

如果我们将 Neutron 架构展开，则会得到下面第二张图：



1. Neutron 通过 plugin 和 agent 提供的网络服务。
2. plugin 位于 Neutron server，包括 core plugin 和 service plugin。
3. agent 位于各个节点，负责实现网络服务。
4. core plugin 提供 L2 功能，ML2 是推荐的 plugin。
5. 使用最广泛的 L2 agent 是 linux bridage 和 open vswitch。
6. service plugin 和 agent 提供扩展功能，包括 dhcp, routing, load balance, firewall, vpn 等。

至此，Neutron 架构已经讨论完，希望大家已经理解。  
下节开始，将通过实验来实践架构中的各个组件。  
第一步将是准备实验用的物理环境。