上一节最后问了大家一个问题：如果 subnet 没有开 DHCP，会是怎样一个情况？

在其他条件不变的情况下，cloud-init 依然会完成那 3 个步骤，也就是说网卡还是会被配置成 dhcp 模式，只是最后网卡没办法获得 IP 而已。

不开 DHCP 也是一个常见的场景，为了让 instance 的网卡在这种情况下也能够被正确配置，我们需要借助 config drive，下面开始实践。

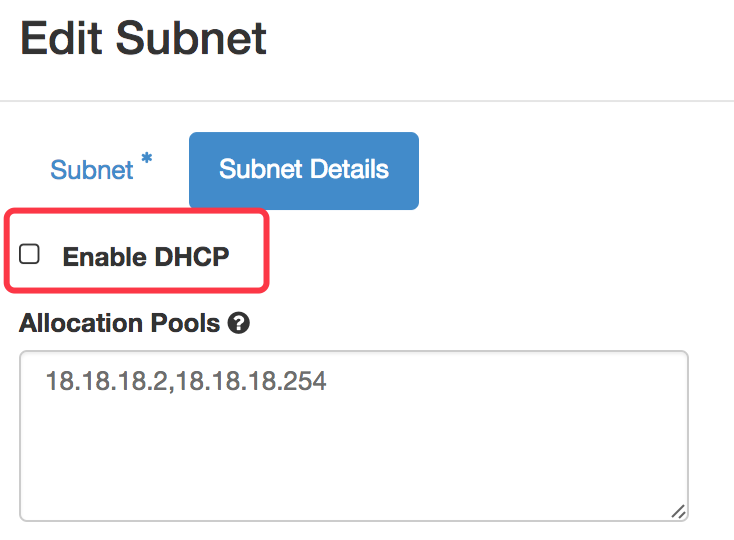
在计算节点 /etc/nova/nova.conf 中需要添加一个配置，然后重启 nova-compute 服务。

[DEFAULT]

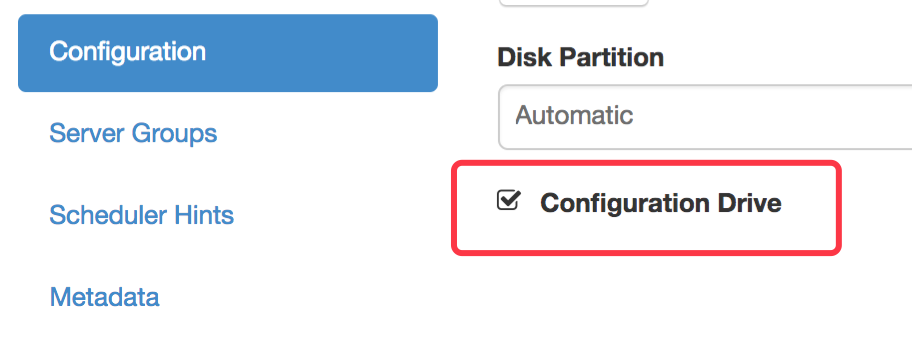
flat\_injected = True

flat\_injected 的作用是让 config drive 能够在 instance 启动时将网络配置信息动态注入到操作系统中。

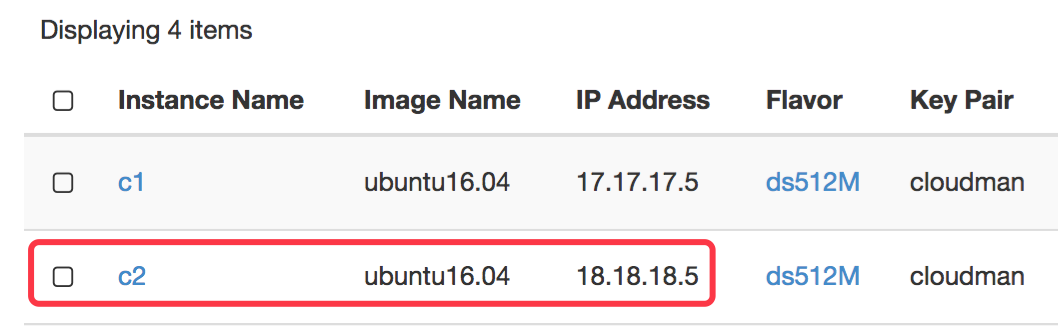
当前网络的 DHCP 已经关闭。



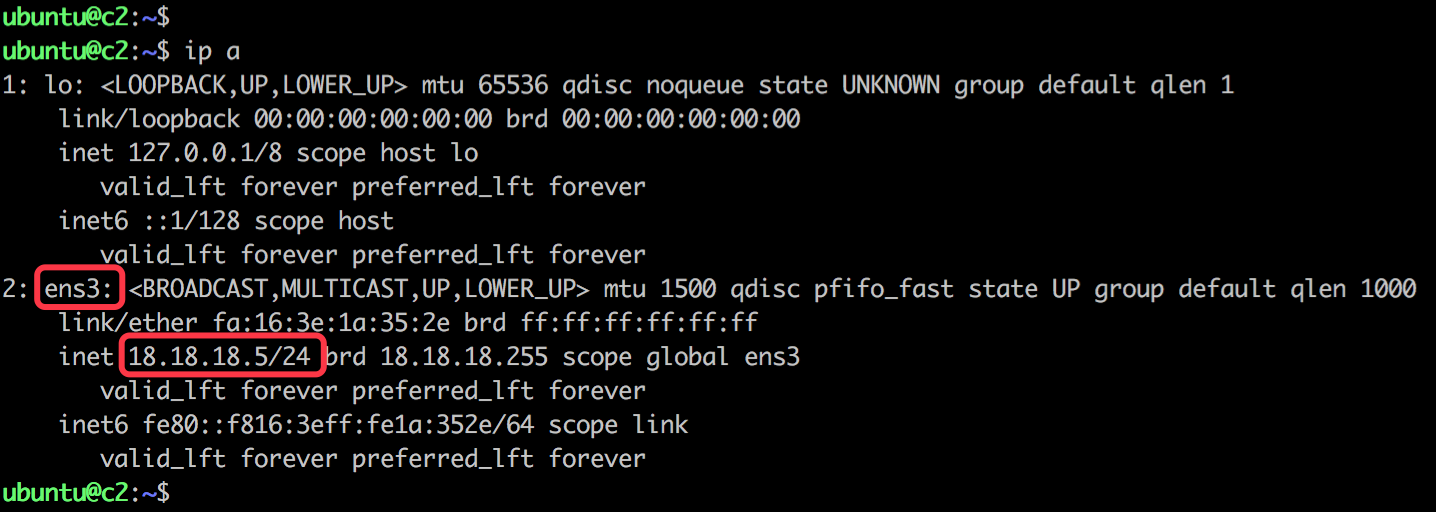
instance 部署时指定使用 config drive。



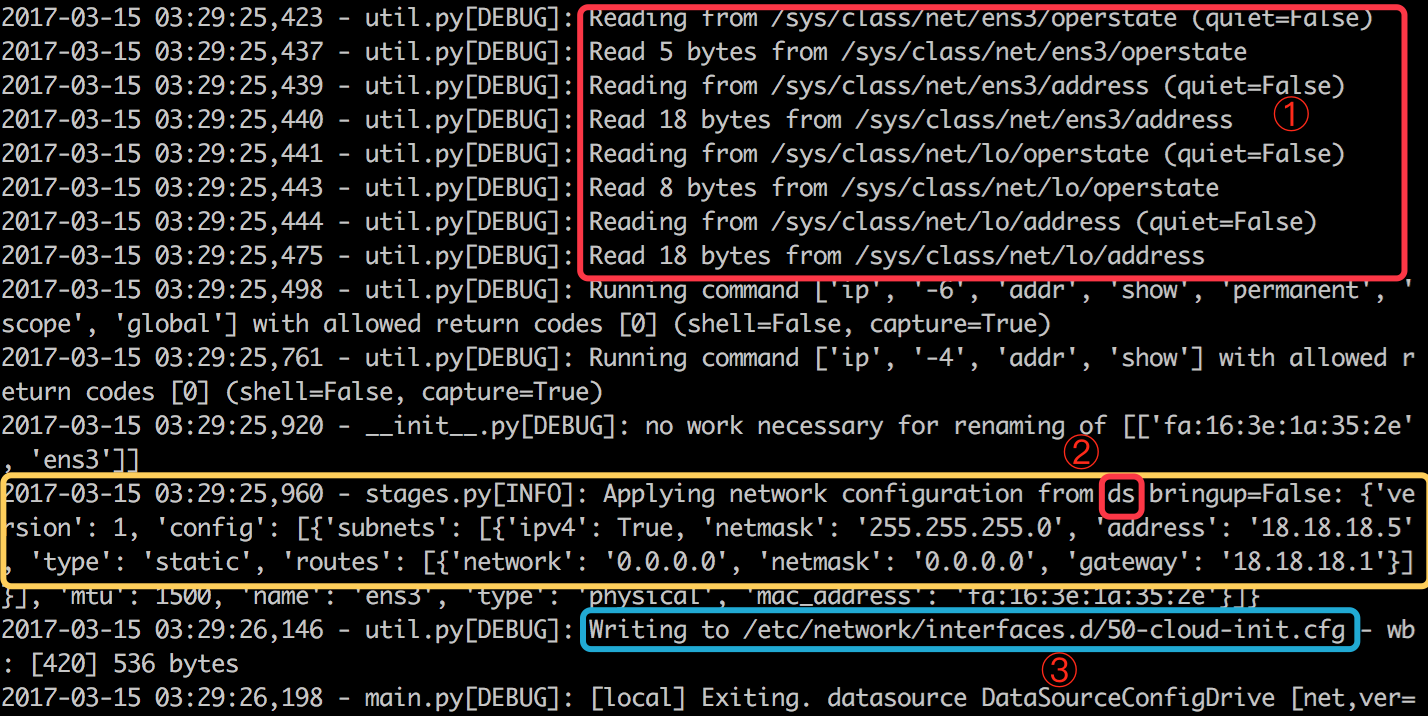
Neutron 为 instance 分配的 IP 为 18.18.18.5。



instance 启动后登录系统，ip a 验证 IP 已经成功配置，说明 config drive 起作用了。

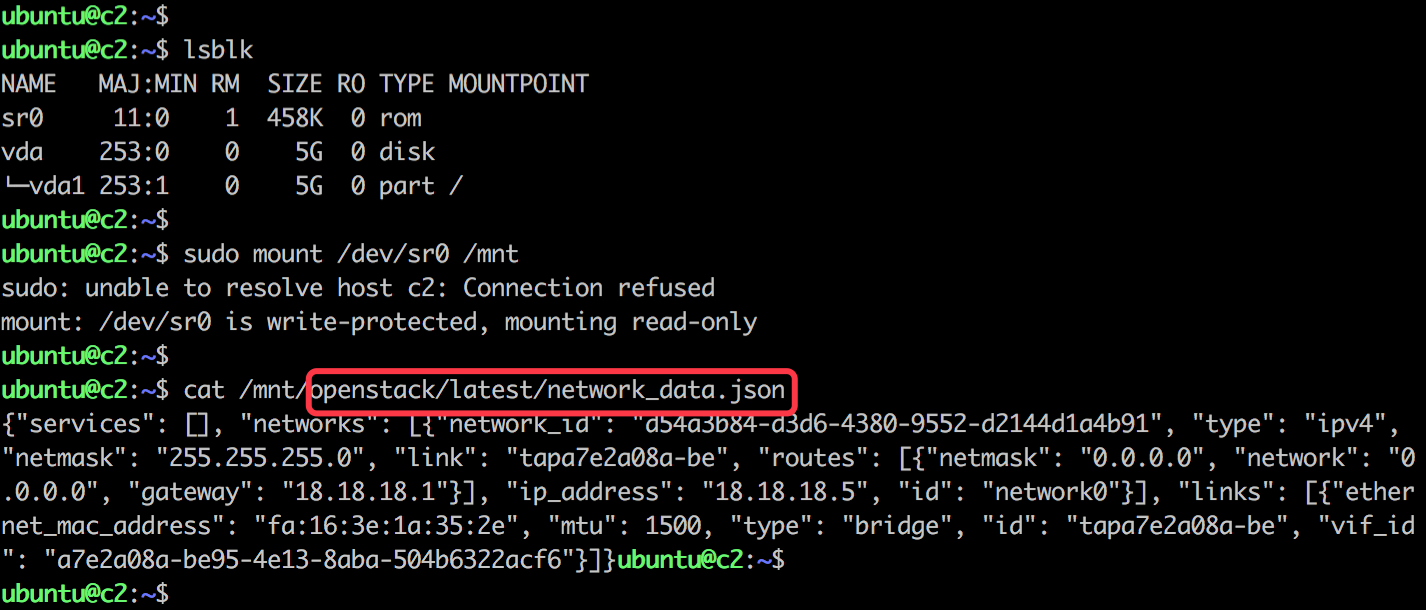


重要的是弄懂 18.18.18.5 这个 IP 是如何配置上去的。打开 /var/log/cloud-init.log，分析如下：

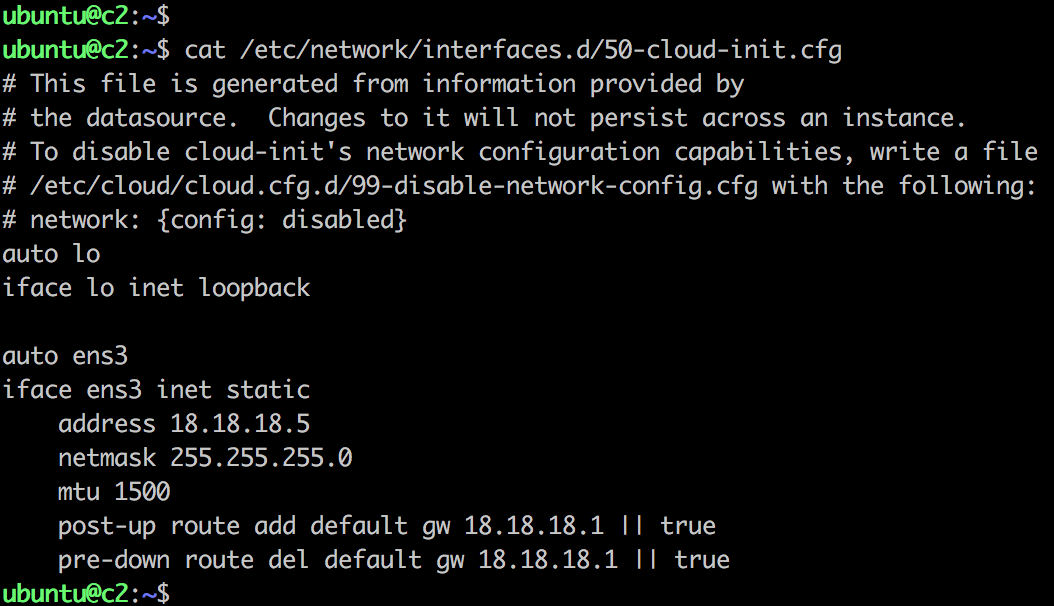


① 扫描出 instance 中的所有网卡，这一步与不使用 config drive 的情况完全一样。

② 获取该网卡的配置信息。 日志显示配置信息是从 ds 获取。ds 是 datasource 的缩写，在这里指的就是 config drive。在不使用 config drive 的情况下采用的是 fallback 配置。网卡配置信息记录在 config drive openstack/latest/network\_data.json 文件里，内容如下：



③ 将配置信息写入 /etc/network/interfaces.d/50-cloud-init.cfg，内容为：



可以看到 IP 以 static 方式配置。

总结一下：

1. 在没有使用 config drive 的情况下，cloud-init 只会配置第一块网卡，且设置为 dhcp 模式，所以：

① 如果 instance 只有一块网卡，且启用了 DHCP，网卡能够被正常拉起。

② 如果 instance 有多块网卡，第一块会尝试以 dhcp 方式拉起，其他网卡不作处理。

2. 使用 config drive 的情况下，无论是否启用 DHCP，所有网卡都能被正确配置且成功拉起（如果 dhcp 网卡 >= 2，CentOS 还是有问题，可能跟目前所用的 cloud-init 版本较低有关）。

3. 如果可能，尽量使用 config drive。  
  
讨论完网络配置这个最重要的主题，下一节我们来看看 cloud-init 的其他典型应用：设置 hostanme，设置用户初始密码，安装软件等。