**Openstack容器化部署研究之：Kolla离线制作Openstack服务的Docker容器镜像**

Kolla项目提供了kolla-build命令进行Docker容器镜像编译，用户从Github上(https://github.com/openstack/kolla/)下载安装之后即可使用kolla-build命令进行镜像编译。正常情况下，kolla-build命令默认编译全部已被Kolla项目组容器化的Openstack项目，但是就目前国内的网络环境而言，对于普通用户，使用kolla-build命令几乎不可能以在线方式正常编译全部Openstack容器镜像。关于为何需要采取离线方式自己制作Openstack容器镜像，个人认为有以下几个原因：

1、理清Kolla制作镜像的机制原理，这是熟悉Kolla项目的极佳过程；

2、灵活按需自定义Docker镜像，按照自己需求修改Dockerfile文件以自定义Openstack的Docker容器镜像；

3、比起使用他人制作的镜像，自己定义的镜像更具安全感和成就感；

4、最关键也是最重要一点，国内以在线方式使用Kolla几乎没法进行镜像编译。

理论上讲，通过Kolla项目来制作Openstack的Docker镜像其实非常简单，如果不是因为国内特殊的网络环境，只需kolla-build一个不带任何参数的命令即可搞定。反过来想，应该感谢GFW，不然也不会花心思去研究kolla-build背后的秘密。要通过Kolla项目进行离线Docker容器镜像的制作，笔者认为应该具备如下几方面的知识：

1、本地yum源的制作，以及远程yum仓库的本地同步原理和配置。可参考如下博文：http://blog.csdn.net/madmanvswarrior/article/details/49952245。

2、jinja2模板的基本语法，Kolla大量采用jinja2模板进行变量传递，因此需要掌握基本的jinja2相关知识，例如模板的定义和渲染等基本知识，以及如何在jinja2模板中引用变量和编写Python命令行等。参考：http://docs.jinkan.org/docs/jinja2/

3、Docker Registry/Repository知识，提到Docker镜像，Docker仓库必不可少，因此如何制作Docker私有仓库，以及如何与私有仓库进行pull/push操作是必须掌握的。可参考如下博文：http://blog.csdn.net/wangtaoking1/article/details/44180901

4、Dockerfile是必须掌握的，Dockerfile中最基本的命令，如RUN、COPY、CMD等等，Kolla项目的Dockerfile遵循Docker官方的Dockerfile最佳实践指南。参考：https://docs.docker.com/engine/userguide/eng-image/dockerfile\_best-practices/#run

5、RHEL/Centos系统，则最好了解Docker在Centos下的默认存储驱动DeviceMapper，Docker为了保证开箱即用，在Centos中的DeviceMapper默认采用的是loop-lvm模式，对于生产系统，官方推荐采用direct-lvm模式，因此不管是否生产系统都都建议手动配置为direct-lvm模式。参考：  http://blog.csdn.net/qq\_26923057/article/details/52351731

6、基本的Python语言知识。要想更深层次的了解Kolla，你需要一定的Python知识，Kolla项目的kolla/kolla/image/build.py是Kolla编译镜像的关键，熟读此脚本是理解Kolla的关键。脚本在此：https://github.com/openstack/kolla/blob/stable/ocata/kolla/image/build.py