# Netconf base on XOS

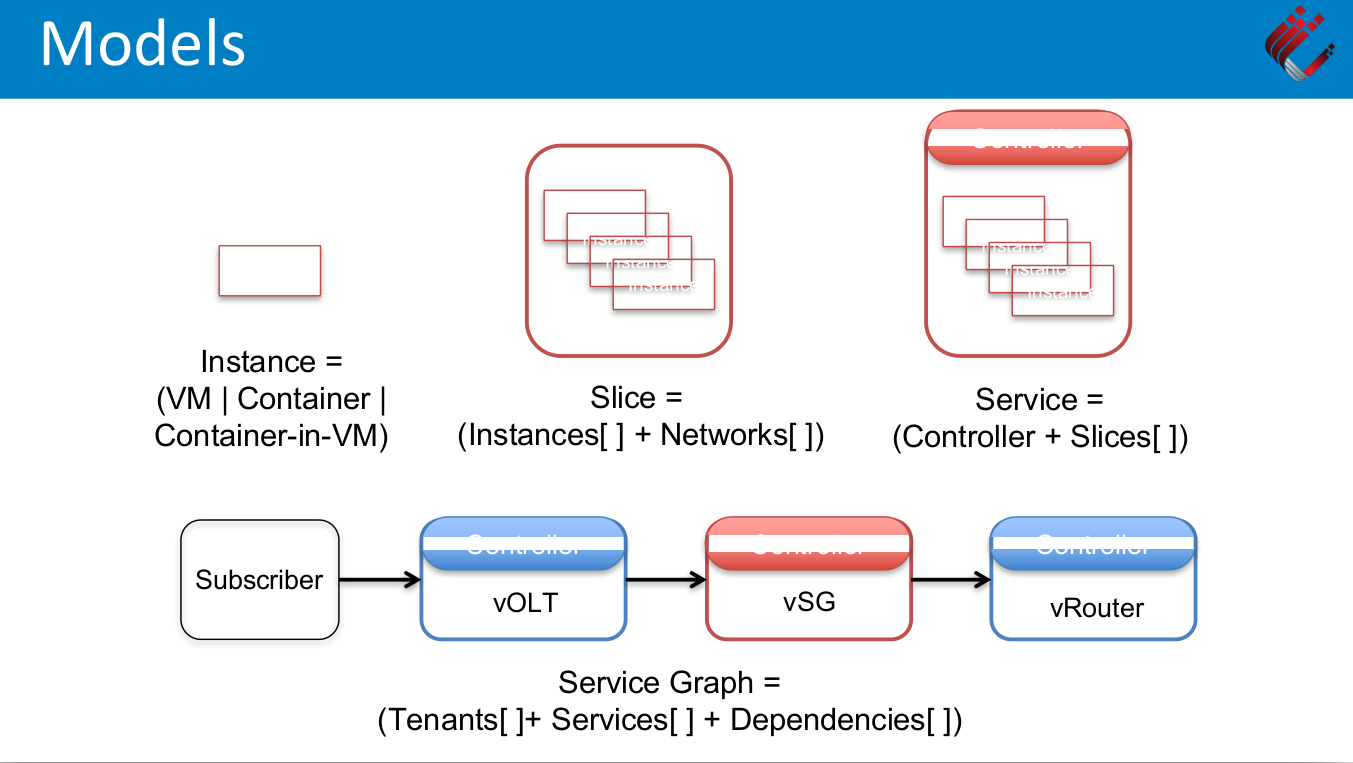
## 参考资料

XOS Guide: <https://guide.xosproject.org/>

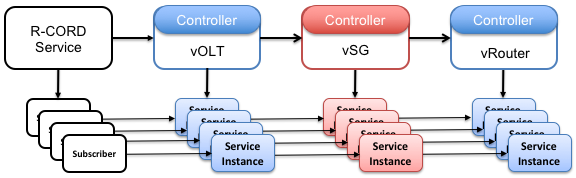
说明：这个是XOS官方的资料，重点要搞懂XOS的模型框架、同步框架、核心模型。其实netconf-service用到的模型很简单，就一个继承Service的NetconfService和继承XOSBase的NetconfApp。

核心模型中提到了Service/ServiceInstance、ServiceDependency/ServiceInstanceLink

我说下我的理解。Service就是XOS中的一个服务，也就是能提供一个完整功能的服务，而ServiceInstance设计出来是面向租户的，一个租户对应一个ServiceInstance，它是Service的一个实例（其实Netconf-Service没用到这个）。服务间可能有依赖，就是ServiceDenpendency描述的，看下图：

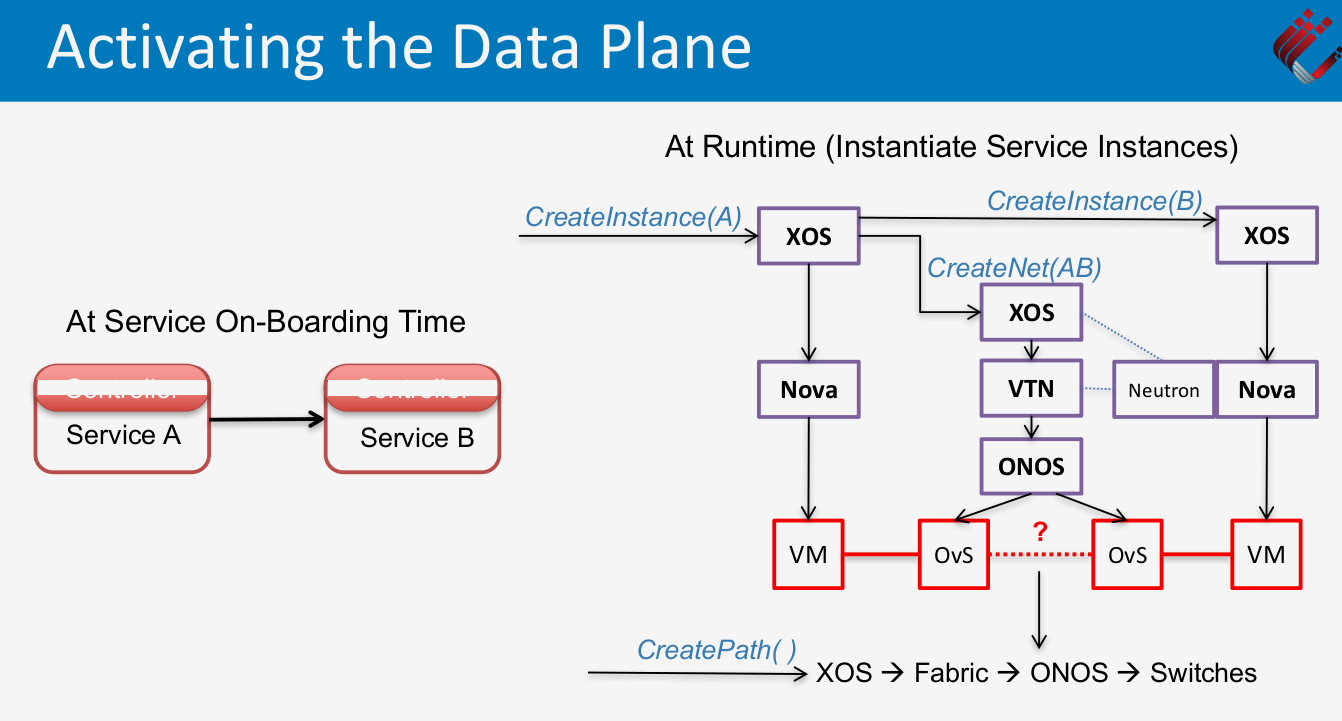


我的理解是vOLT+vSG+vRouter以及之间的依赖组成了一个服务。依赖就是图中的箭头线，XOS中把这个叫做边。其实租户真正使用是不是直接针对Service，而是新建一个ServiceInstance，如下图：



我的理解是这三个服务以及依赖组成了R-CORD这样一个服务。租户生成一个R-CORD Service的instance，也就是Subscriber，默认就会对应其他三个服务的ServiceInstance，他们之间就原生有依赖（ServiceInstanceLink）。Guide中提到这么做目的是用户之间的隔离，也提到Service和ServiceInstance并不一定同构。

看这个图：



这是基于OpenStack的实现，生成用户数据的方式就是创建ServiceInstace(A)和ServiceInstance(B)，这是因为ServiceA和ServiceB以及依赖组成某种服务。XOS接到这两个请求时，就下发给Nova创建虚拟机，以及下发给Neutron创建虚拟机之间的网络，而这个网络是基于ONOS的实现（这里的理解不敢保证全对），因此两个VM之间就有ONOS实现的网络。

说了这么多，其实我理解XOS就是一个开放架构的服务粘合剂，每个想使用XOS集成自己功能的服务，就得基于目前已经提供的XOS服务生成一个自己的服务。要这么做：

1. 首先，写一个数据模型，比如你继承哪个已有的服务，NetconfService继承核心模型提供的基础服务Service，然后定义自己的数据。
2. 我先插入一段我自己看到的XOS的代码，这个模型会被编译进xos.proto里，也就是你定义的服务直接进入了XOS的核心，这个可以在netconf-service pod的这个路径看到：

root@netconf-service-5f88c64f89-k24sj:/opt/xos/synchronizers/netconf# ls /tmp/xos\_grpc\_protos/xos.proto

我理解，XOS Core也会加载进我们自定义的模型或者说服务。默认情况下整个XOS都可见这个服务，但是有模型定义的授权策略，这个在核心模型中可以看到。Netconf-service并没有用到这个功能。

1. 定义自己的同步器。有四种同步机制，这个是数据发生变化（GUI/kafka/tosca等）后怎么处理数据的。Netconf-service只用到了SyncStep这种机制，这种机制主要用来跟后端服务通信的。也就是检测到数据变化时，netconf-service(此时它是客户端)和定义的netconf-server怎么交互数据。我们目前就是通过tosca，把netconf server信息配置到NetconfApp定义的数据中去，因为数据发生变化，所以SyncStep被启动，然后就根据Server信息连接真正的Netconf-server，查询数据，把yang数据写入定义好的数据项中。核心模型定义了几个时间戳，用来检测哪些步骤会启动，详细的看Guide中的说明。
2. 数据模型和同步器都定义好之后生成netconf-service的镜像

定义helm chart。目前chart包含拉起netconf-service的Deployment以及一个Job。这个Job就是tosca任务，tosca用来定义各个模型中的数据是什么。关于tosca可以参考这个链接：<http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.2/TOSCA-Simple-Profile-YAML-v1.2.html>

Tosca Job其实就是发送一个POST请求给xos-tosca。Tosca任务的数据就描述了数据以及数据间的关系。

## netconf-service代码

附件的netconf-0.0.1.tar.gz就是netconf-service，它用来生成镜像。编译命令如下：

docker build -t netconf-synchronizer:latest -f Dockerfile.synchronizer .

附件的netconf-service-0.0.1.tar.gz就是helm charts。部署命令如下：

helm install -n netconf-service netconf-service

需要强调的是必须部署cord-platform(其中包括的XOS)。

helm install -n cord-platform cord/cord-platform --version=6.1.0

## 参考示例

1. base-kubernets:

服务：<https://gerrit.opencord.org/kubernetes-service>

Helm charts: <https://charts.opencord.org/base-kubernetes-1.0.2-fix1.tgz>

1. Onos-service

服务：<https://gerrit.opencord.org/onos-service>

Helm charts: <https://charts.opencord.org/seba-1.0.0.tgz>

1. Simpleexampleservice

服务：<https://gerrit.opencord.org/simpleexampleservice>

Helm charts: https://charts.opencord.org/demo-simpleexampleservice-1.0.2.tgz

多看示例，代码并不多。