Why contrail with Kubernetes ?   
  
Now after we have seen the main concepts of Kubernetes in chapter 2 and 3 what could be the gain in adding Contrail to a standard Kubernetes deployment ?  
 in brief, Contrail offers common deployment for multiple environments (OpenStack, Kubernetes,..,etc) as well it enriches Kubernetes networking and security capabilities.

When it comes to deployment for multiple environments,   
Yes containers is the current trend to build applications but don’t expect everyone to migrate everything from VM to containers that fast (This is not to mention the nested approach where containers are hosted in VM ) and if we add to the picture workload fully or partially run in the public cloud, we end up feeling the misery for network and security administrators where Kubernetes becomes just one thing to manage   
Network and security administrator in many organization manage individual orchestrator / manager for each environment. OpenStack or VMware NSX for VM, Kubernetes or Mesos for Containers, AWS console.   
and here what contrail could put the network and security administrators out of their misery as it provides dynamic end-to-end networking policy and control for any cloud, any workload, and any deployment, from a single user interface   
contrail translates abstract workflows into specific policies, simplifying the orchestration of virtual overlay connectivity across all environments by building and securing virtual networks that connect BMS, VM and Containers located in private or public cloud.   
  
A very common way to deploy Kubernetes is to lunch its POD in VMs orchestrated by OpenStack and this one of the many use cases of contrail doing its magic   
in this book we won’t cover contrail integration with other environments as we focus only in Kubernetes but any feature that we explain in here could be extended for other environments   
  
  
Then what we mean by contrail enriching standard Kubernetes deployment?   
Kubernetes offers flat network connectivity with some security feature confined in a cluster

but Contrail could offer on top of that  
1- namespaces and services customized isolations for segmentations and multi-tenancy

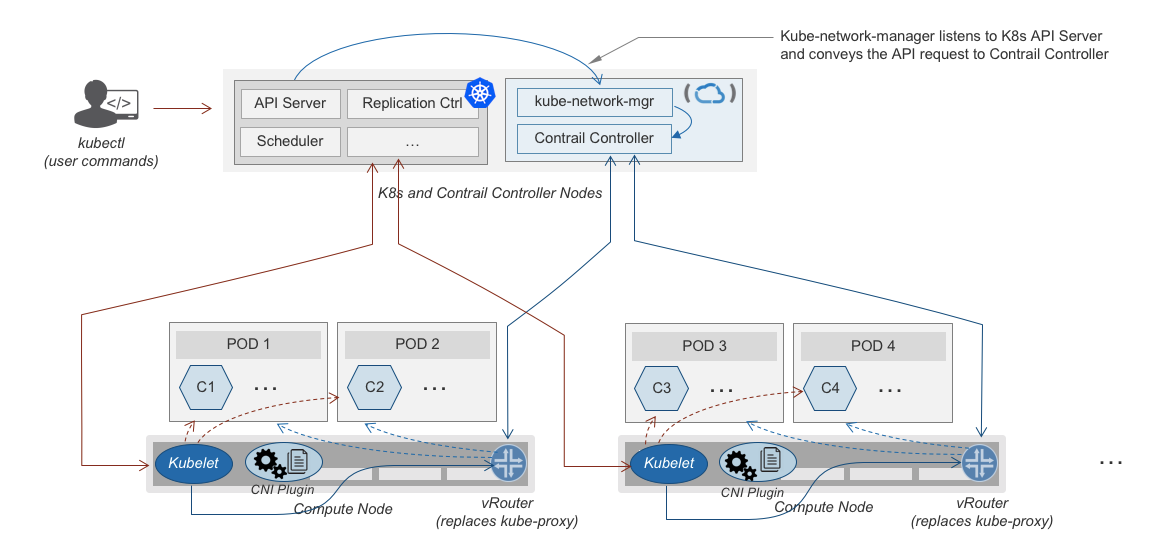
2- service chaining

3-distrubted LB and firewall with extensive centralized flow and logs insight

4- rich security policy using tags that can extend to other environment (OpenStack, VMWare, BMS, AWS ,..,etc)

In this chapter we will cover some of these aspects but first let’s talk about Kubernetes/contrail architecture and the object mapping

Contrail – Kubernetes architecture   
  
A new components of contrail has been add called Kube-network-manager which would listen to API request sent from the Kubernetes API server and translate that to Contrail controller and on the compute node the contrail Vrouter would replace Kube-proxy as shown in the diagram



So not much of change of the regular contrail that we have seen before and all of that is happening behind the scene.  
what we have to be aware of it before dealing with Kubernetes/contrail is the object mapping. because contrail is single interface managing multiple environments - as explained before – each environment has its own acronym and terms hence the need for this mapping

For example, Namespace in Kubernetes are intended for segmentation between multiple teams, or projects as if we are creating virtual cluster. In contrail the similar concept would be named as project so when you create a namespace in Kubernetes it will automatically create an equivalent project in contrail. more on that will come later on for now kindly make yourself familiar with this list of object mapping

Namespace

Single project OR Shared project

Pod

Service

Ingress

Network Policy

Virtual Machine

ECMP Loadbalancer (Native)

Haproxy Loadbalancer for URL

Contrail Security

**Kubernetes Objects**

**Contrail Objects**