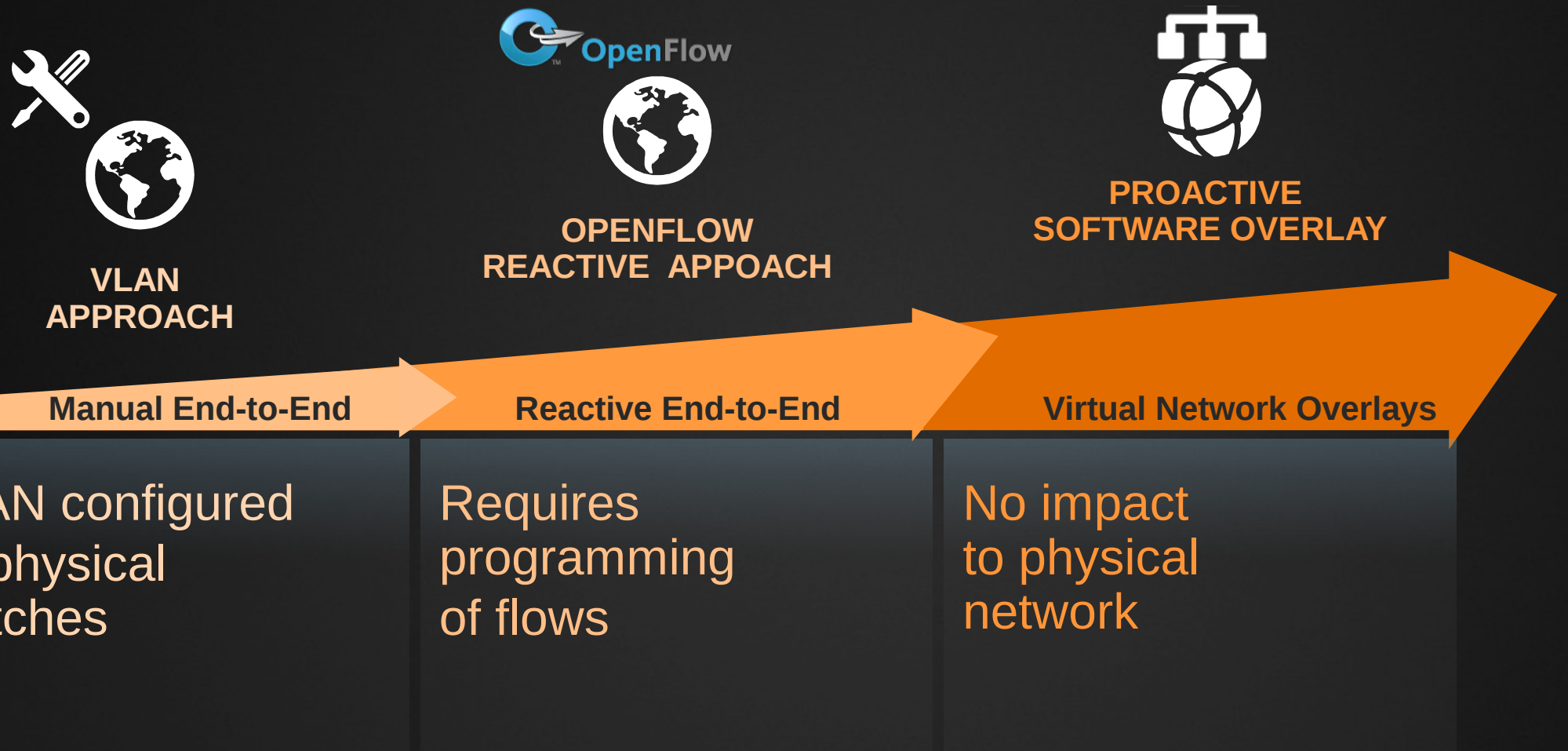
CONTRAIL 介绍

---Juniper overlay SDN 解决方案

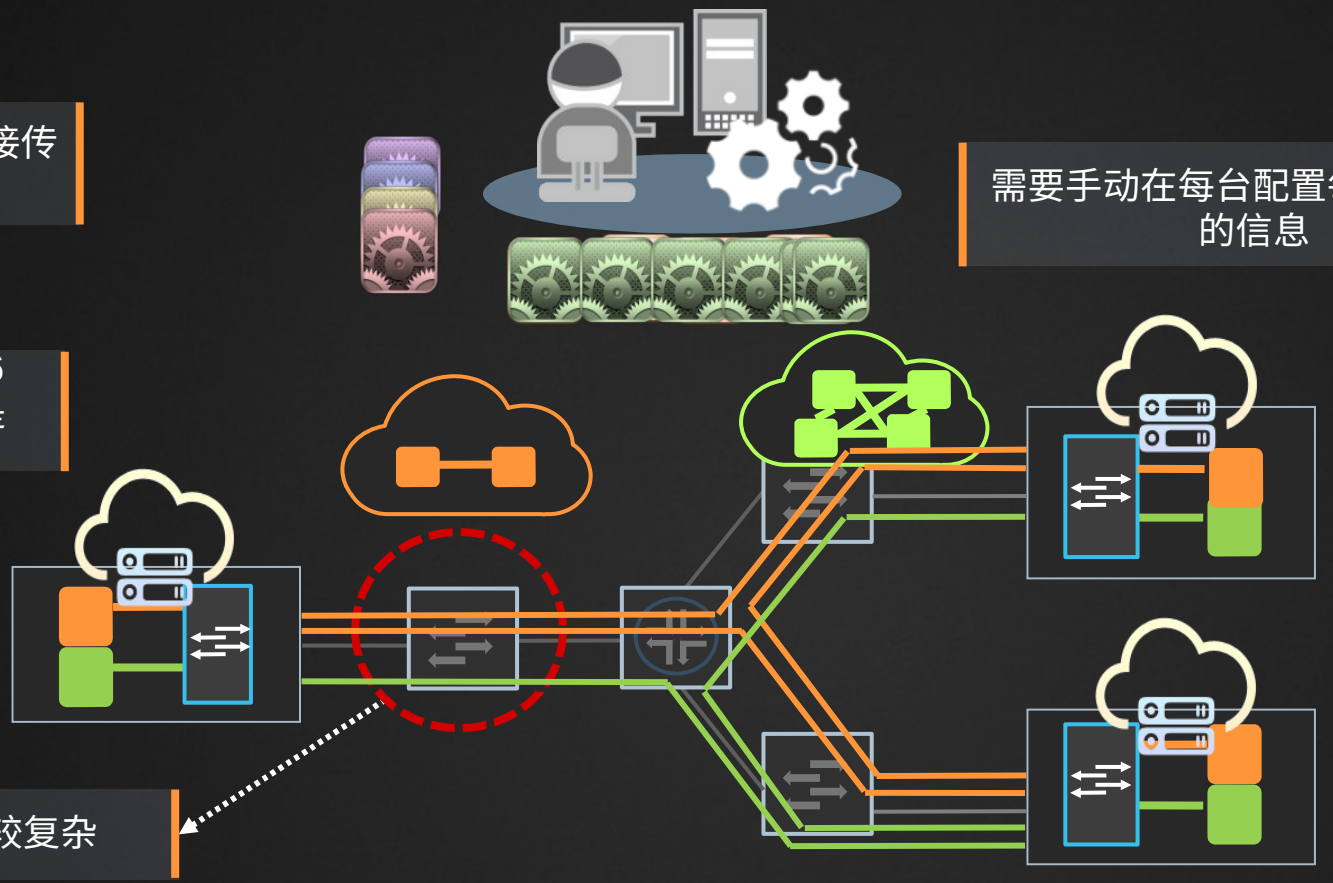
网络虚拟化演进方向


VLAN
APPROACH
OpenFlow
OPENFLOW
REACTIVE APPOACH
PROACTIVE
SOFTWARE OVERLAY
Manual End-to-End
Reactive End-to-End
Virtual Network Overlays
VLAN configured on physical switches
Requires programming of flows
No impact to physical network

网络虚拟化 ---VLAN

用户的流量在物理网络直接传输

VLAN ID 一共只有 4096 个，支持的用户数量也是 4096



需要手动在每台配置每隔 VLAN 的信息

插入服务相对比较复杂

手工控制．低效率．扩展性低．

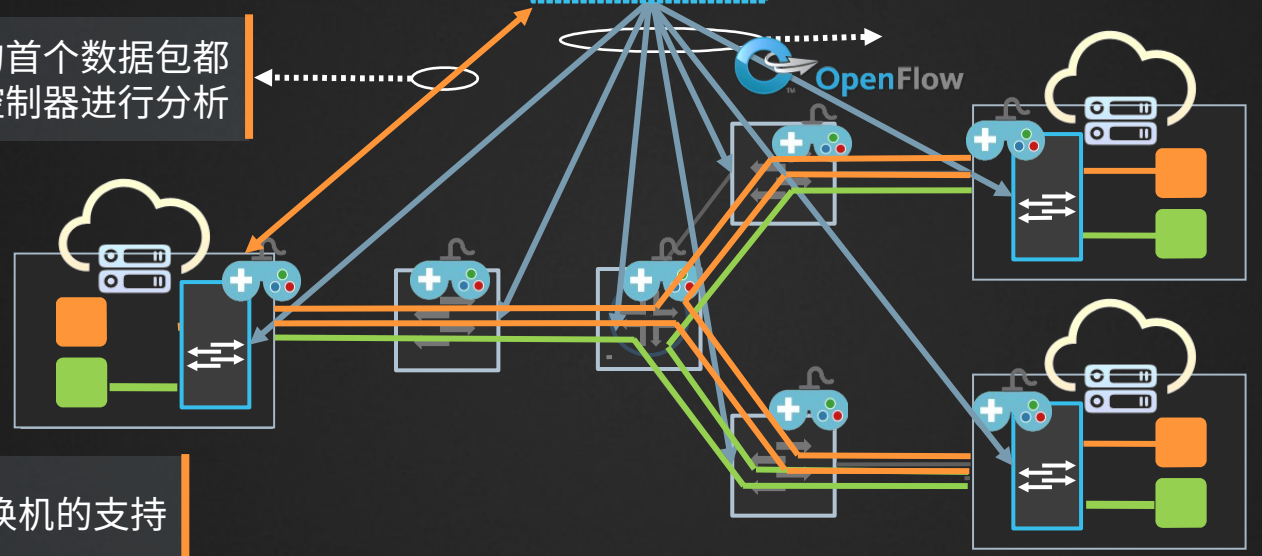
网络虚拟化 ----OPENFLOW

OpenFlow 需要编程

Openflow 控制器

每个用户的流量都要经过物理网络转发

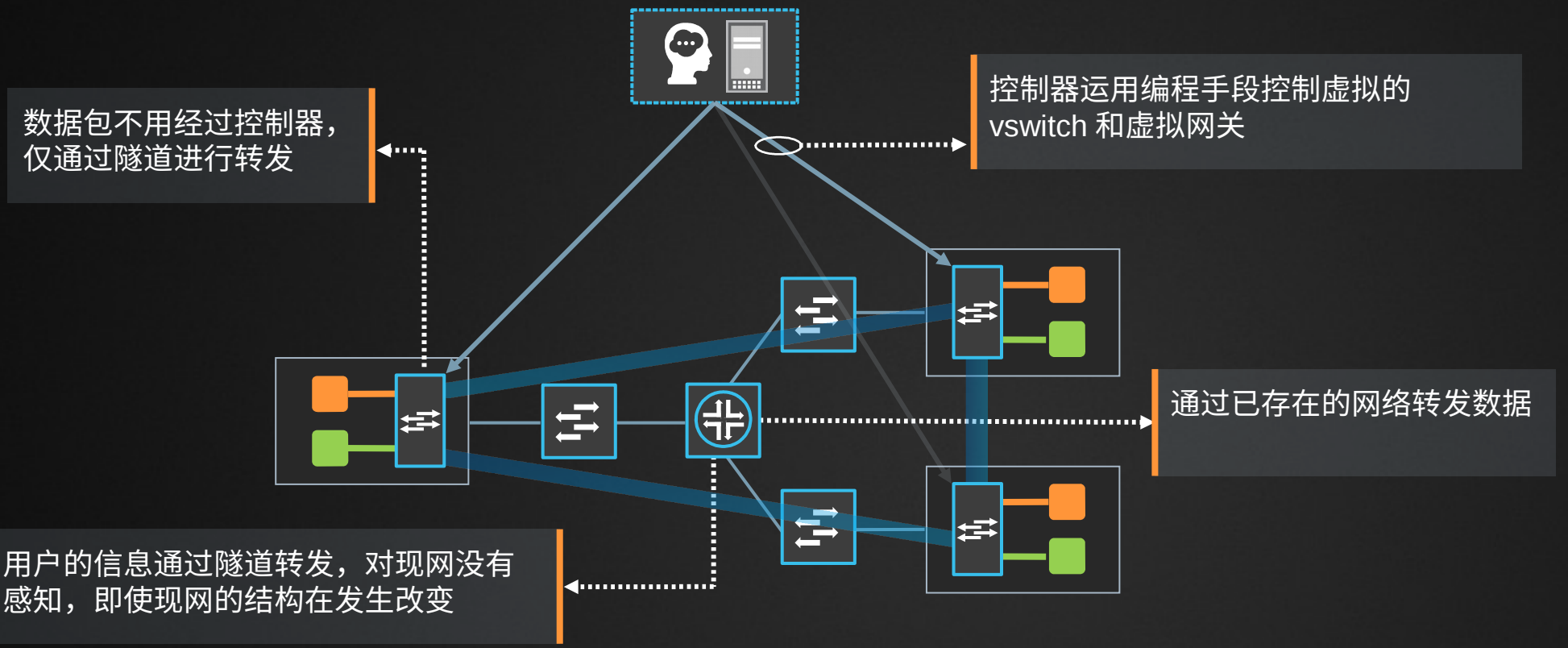
每个流量的首个数据包都要去到控制器进行分析



OpenFlow 需要底层交换机的支持

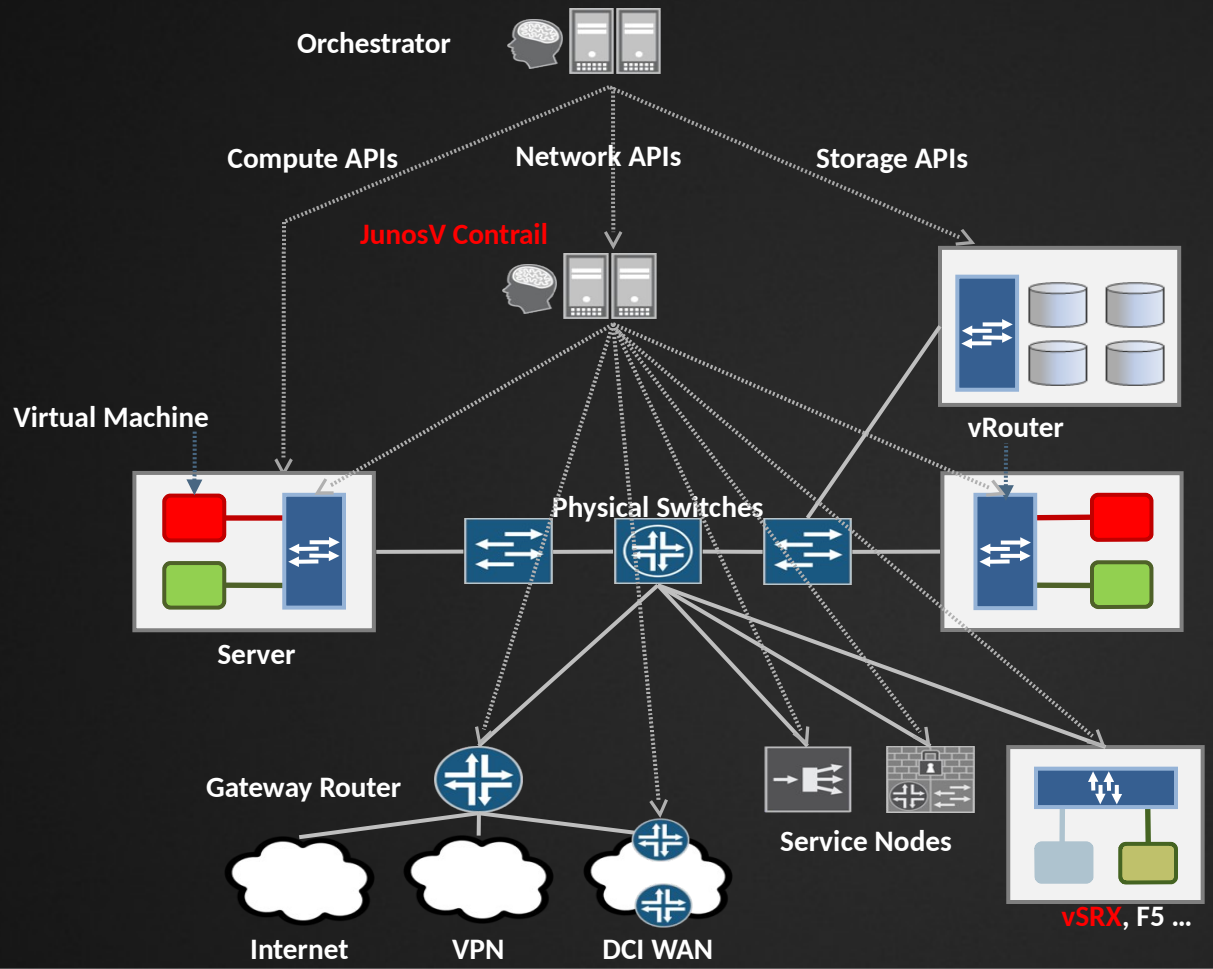
延迟较高． 扩展性低． 增加故障的考虑点． 可升级．

网络虚拟化 ----OVERLAY



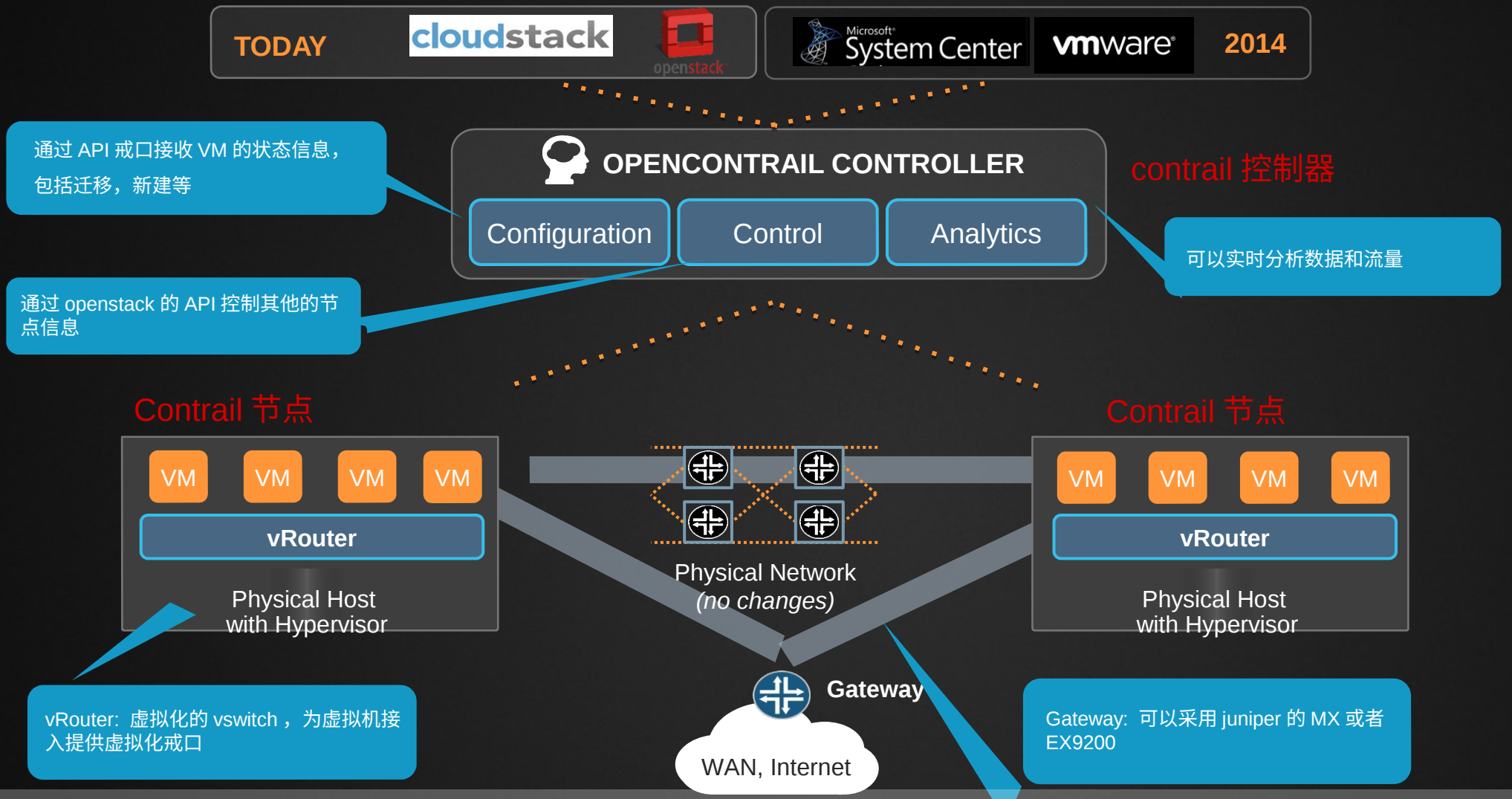
低延迟． 高扩展性． 自动恢复能力． 可以在任何网络上实现．

JUNIPER CONTRAIL 的角色与作用



JUNIPER 的 CONTRAIL 以 OPENSTACK 为基础，通过 API 调用 OPENSTACK 的组件

CONTRAIL 组件



JUNIPER 目前完成 OPENSTACK 的集成，后续还会支持 VMWARE 等更多的云平台系统

CONTRAIL 功能

Routing and Switching

IPAM, Virtual DNS

Load Balancing

Security

3rd Party Network Services

Gateway Services

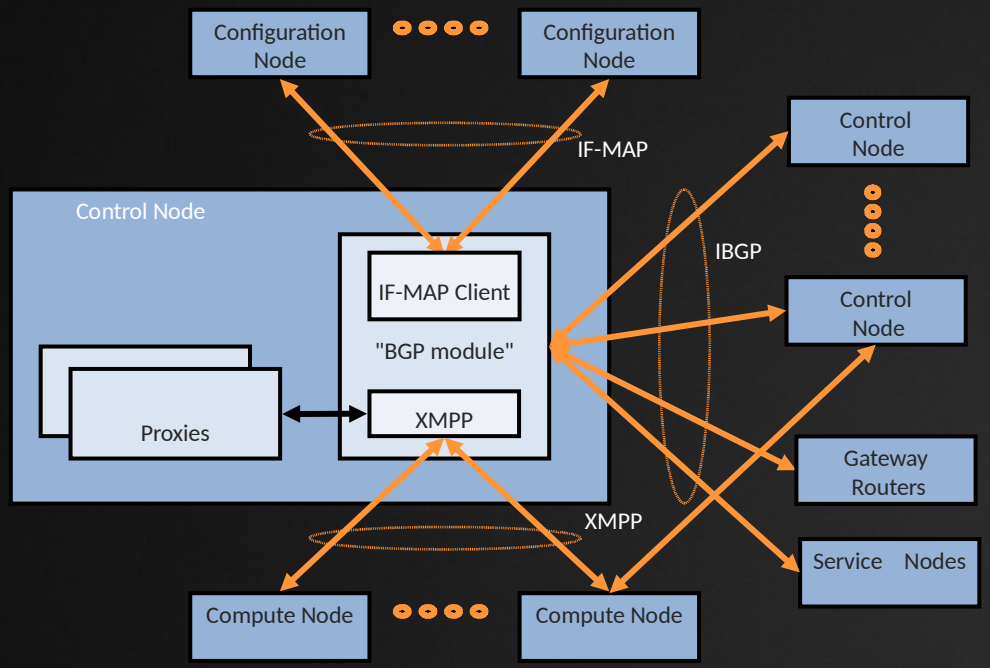
Rich Analytics

Service Chaining

High Availability

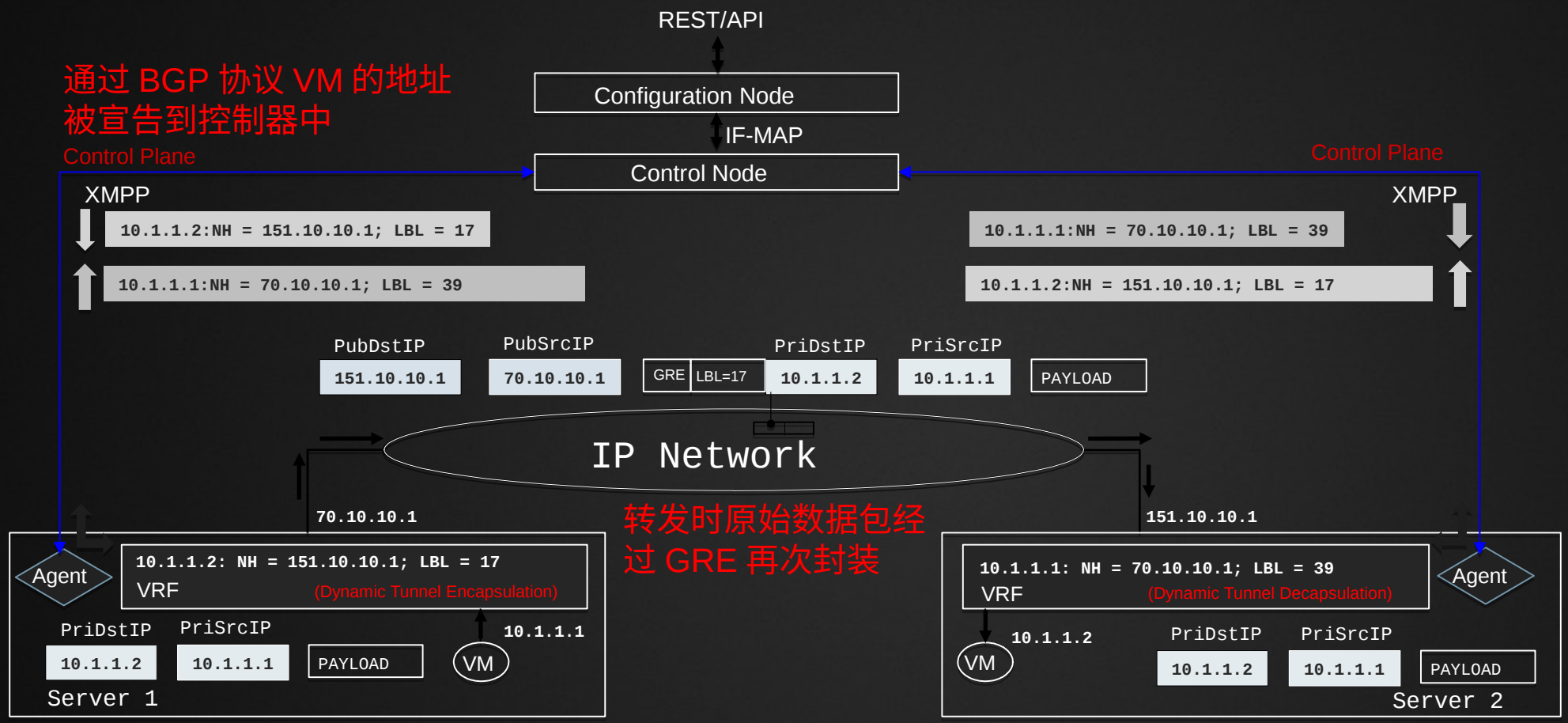
API Services

CONTRAIL – 控制器和节点



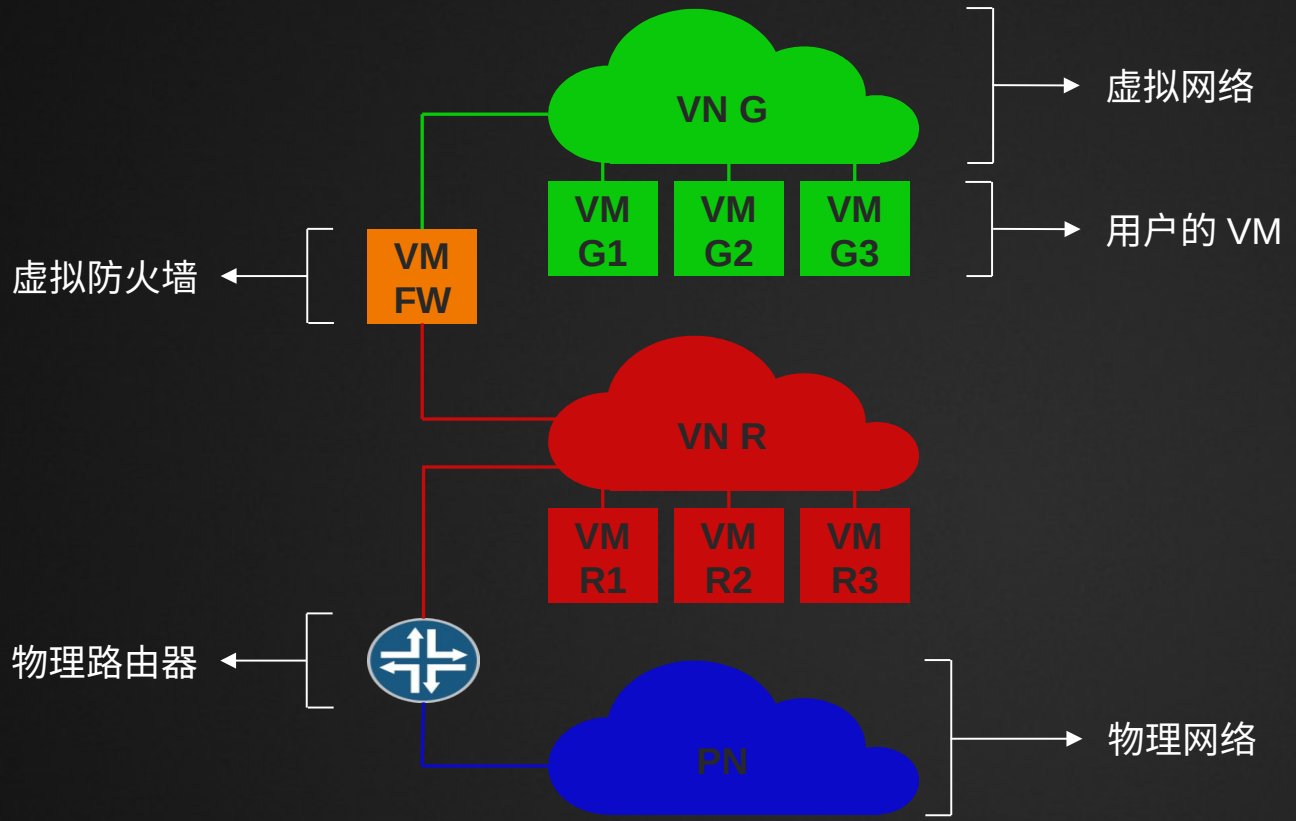
- 控制器和节点之间可以实现控制和转发分离
- 控制器可以控制多个节点，包括路由器和 compute node
- 控制层面通过 BGP 协议实现路由控制
- 转发点通过动态的 GRE 的隧道转发数据
- 物理拓扑和交换机对于用户是透明的

CONTROL PLANE – 路由发布方式

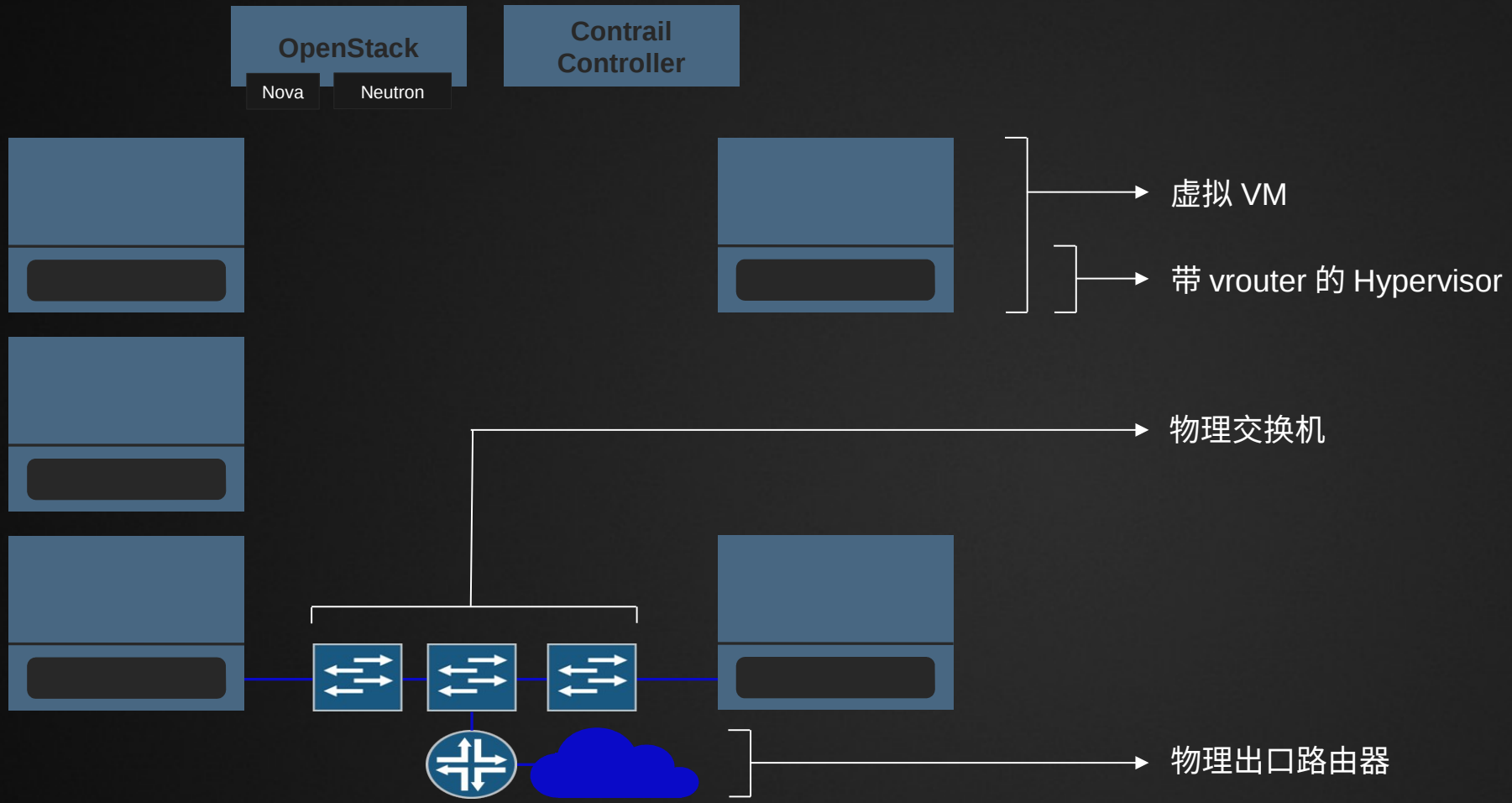


*Outer MAC header was left out intentionally to reduce clutter

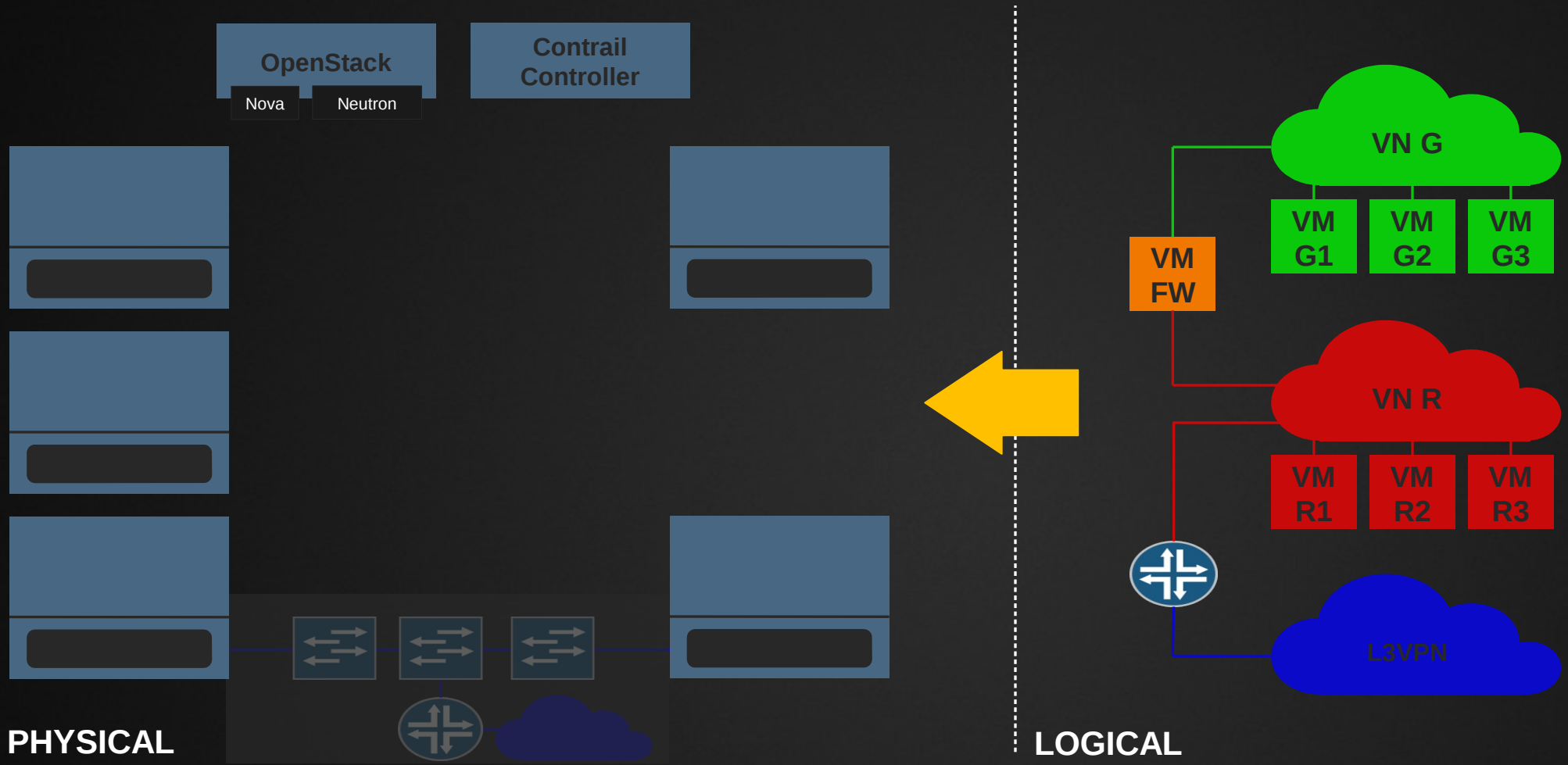
应用场景—逻辑拓扑


VN G
VM G1
VM G2
VM G3
虚拟网络
用户的 VM
虚拟防火墙
VM FW
VN R
VM R1
VM R2
VM R3
物理路由器
PN
物理网络

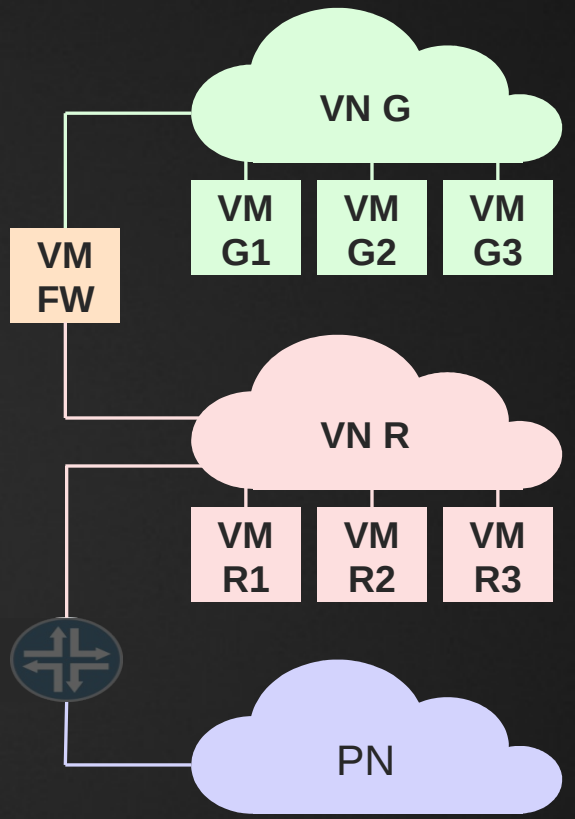
应用场景—物理拓扑


OpenStack
Nova
Neutron
Contrail Controller
虚拟 VM
带 vrouter 的 Hypervisor
物理交换机
物理出口路由器

逻辑与物理拓扑对应

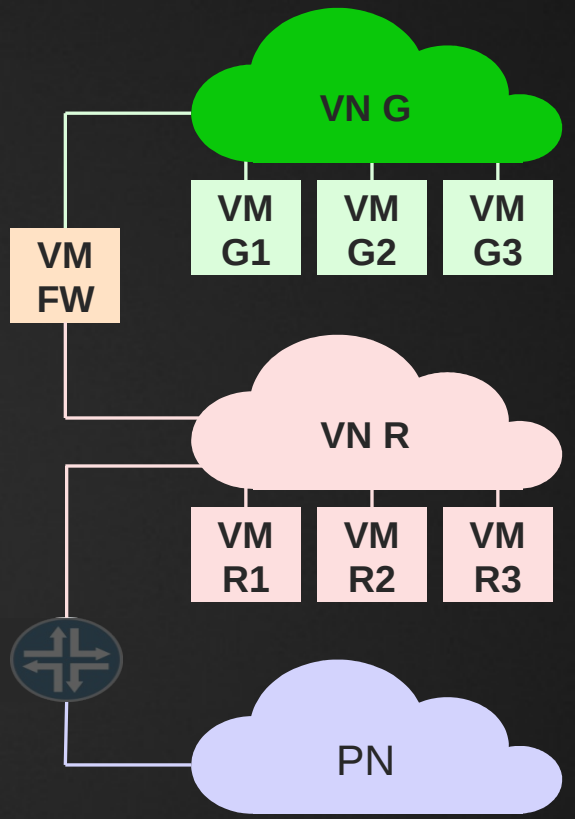

OpenStack
Nova
Neutron
Contrail Controller
VN G
VM G1
VM G2
VM G3
VM FW
VN R
VM R1
VM R2
VM R3
L3VPN
PHYSICAL
LOGICAL

初始化过程，网络还没有建立

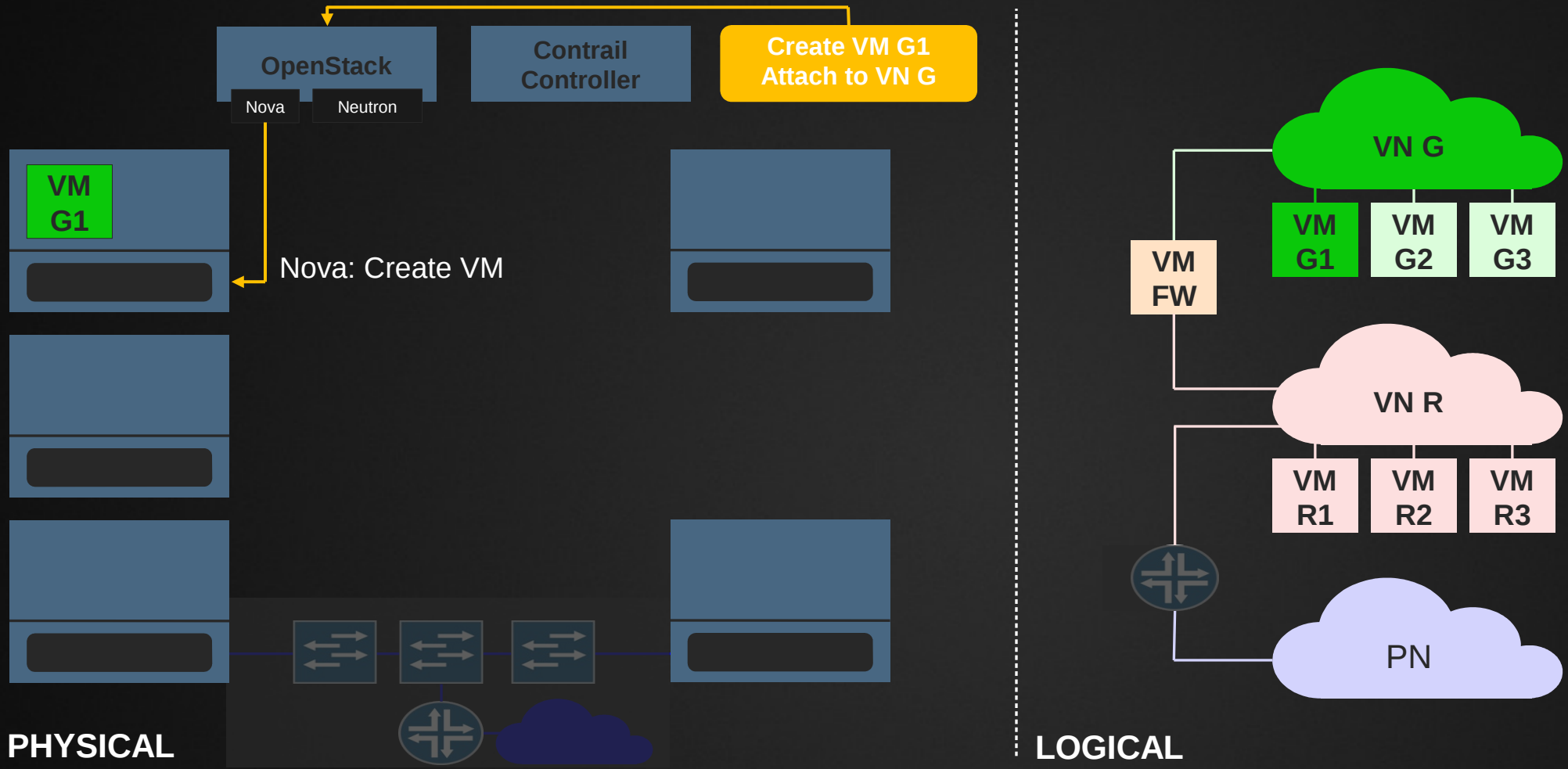


用户新建虚拟网络

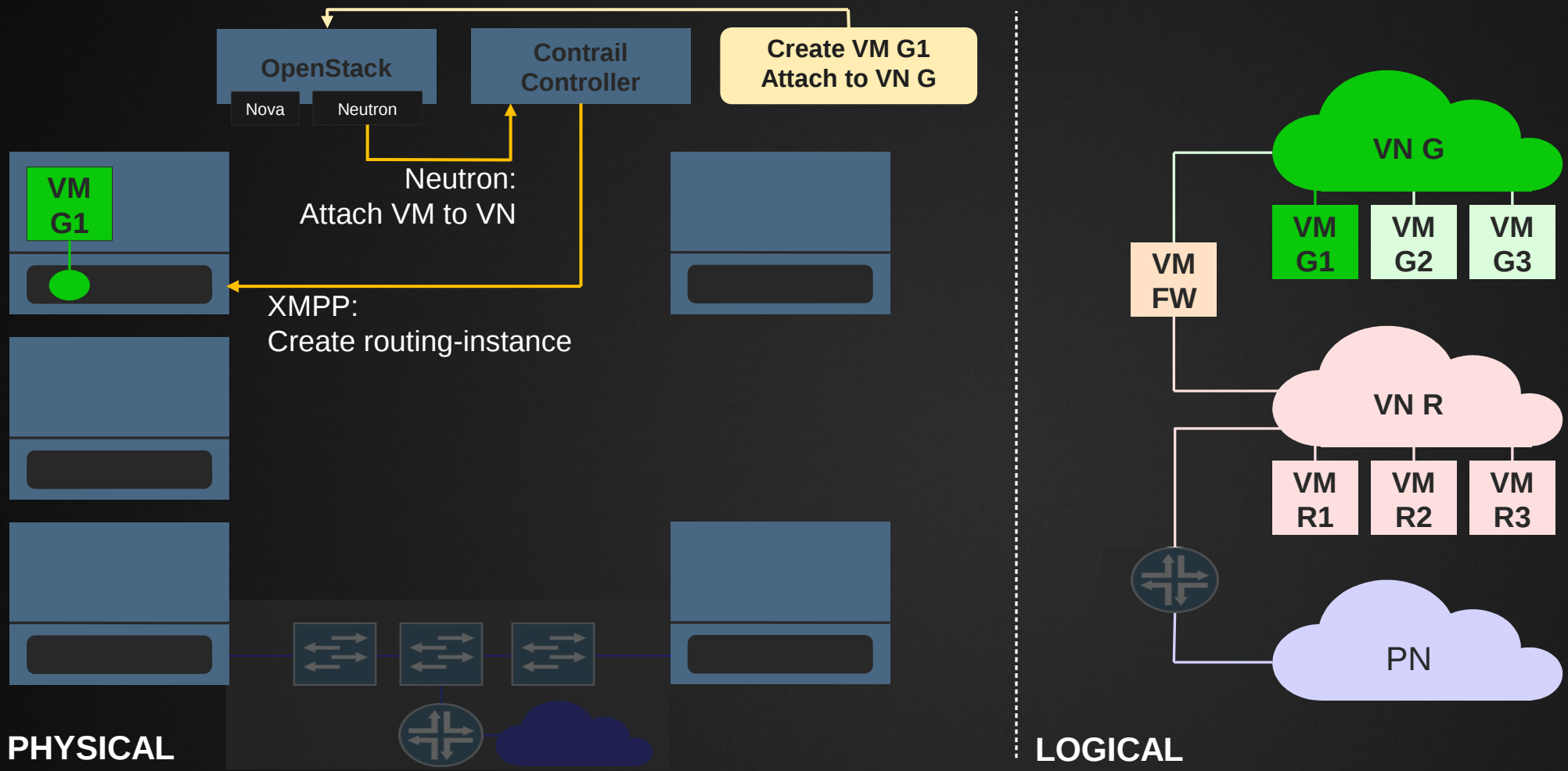

OpenStack
Nova
Neutron
Contrail Controller
Create VN G
PHYSICAL


VN G
VM G1
VM G2
VM G3
VM FW
VN R
VM R1
VM R2
VM R3
PN
LOGICAL

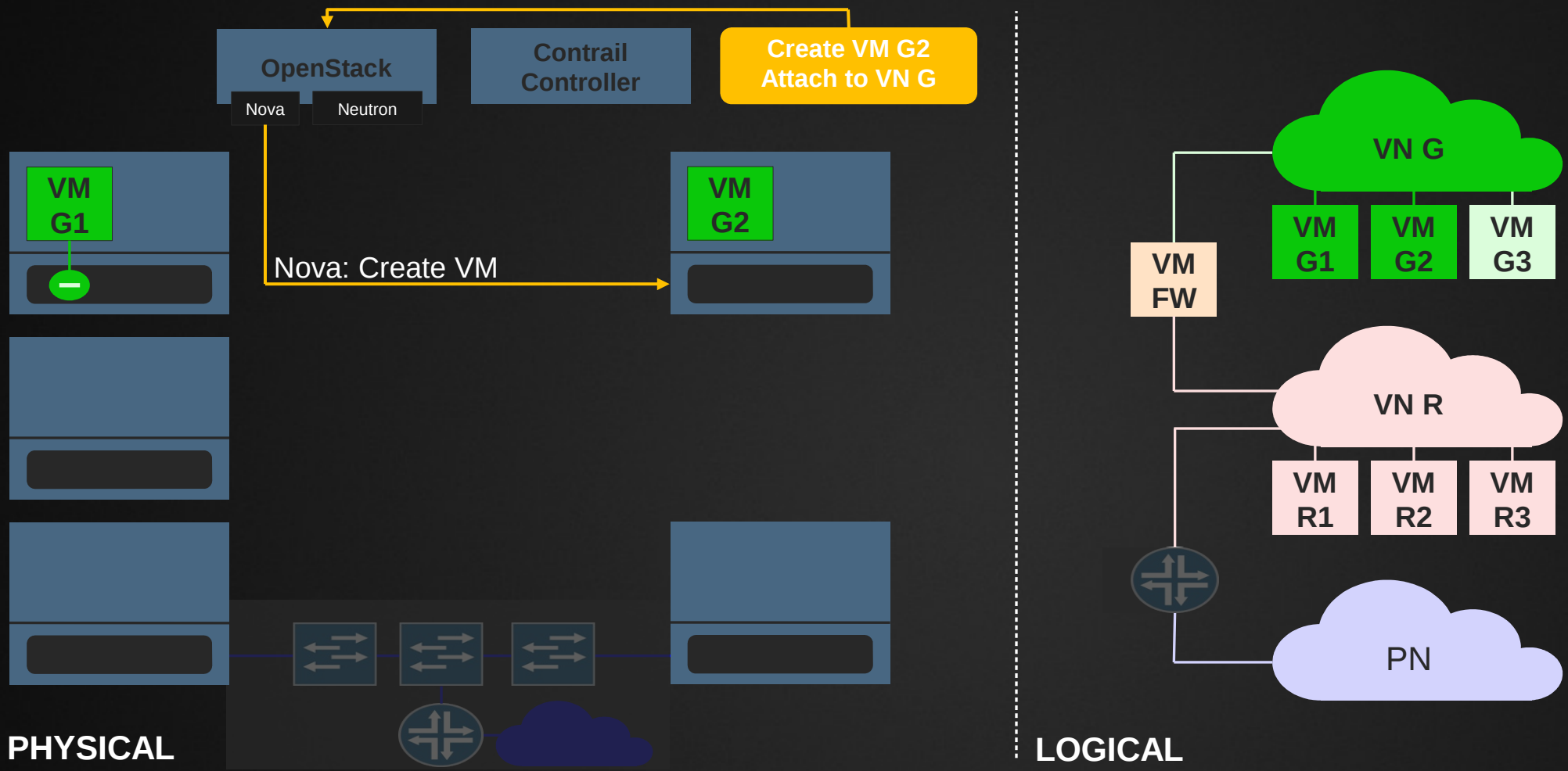
用户新建虚拟机 VM G1



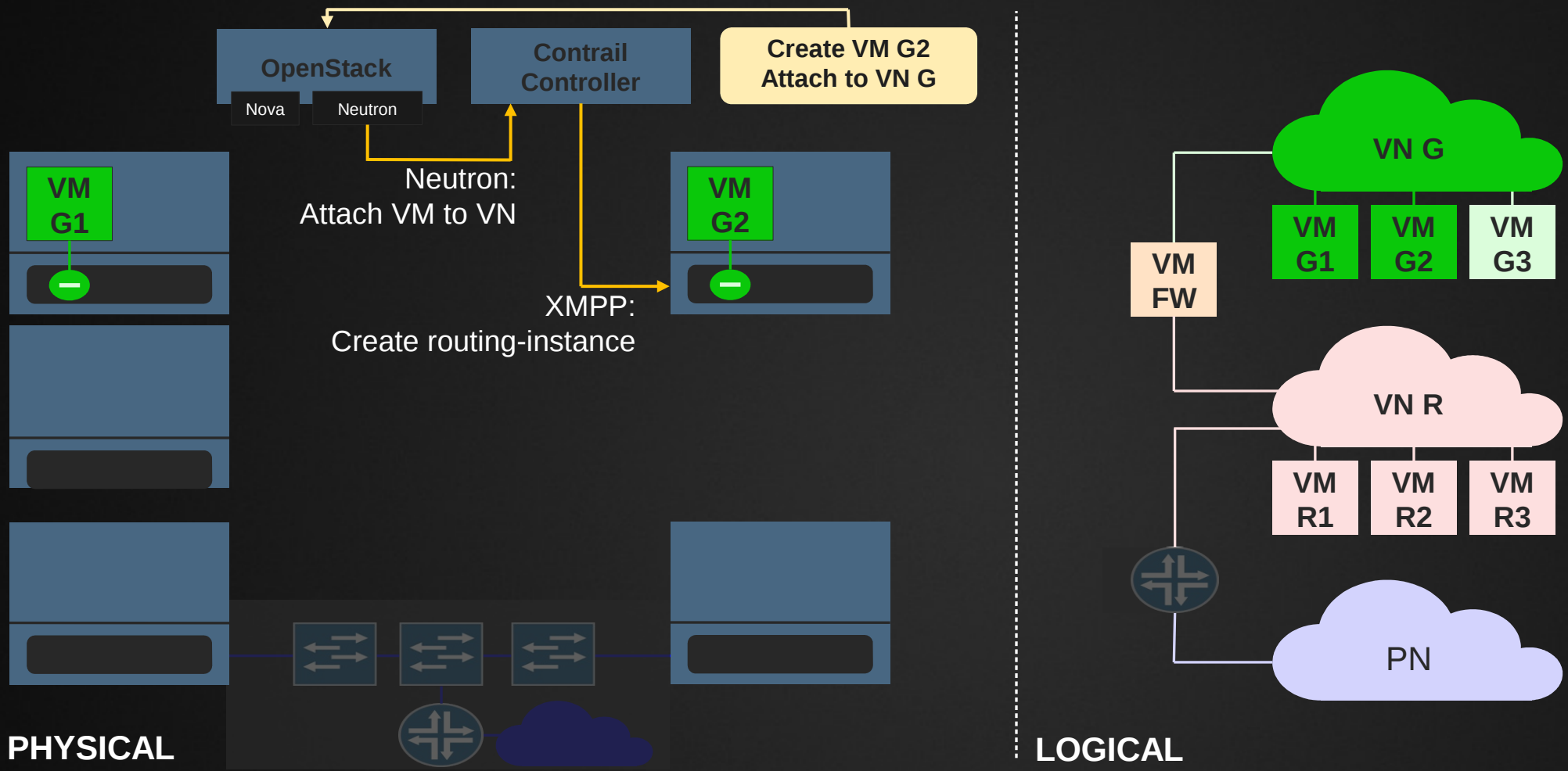
用户新建虚拟机 VM G1



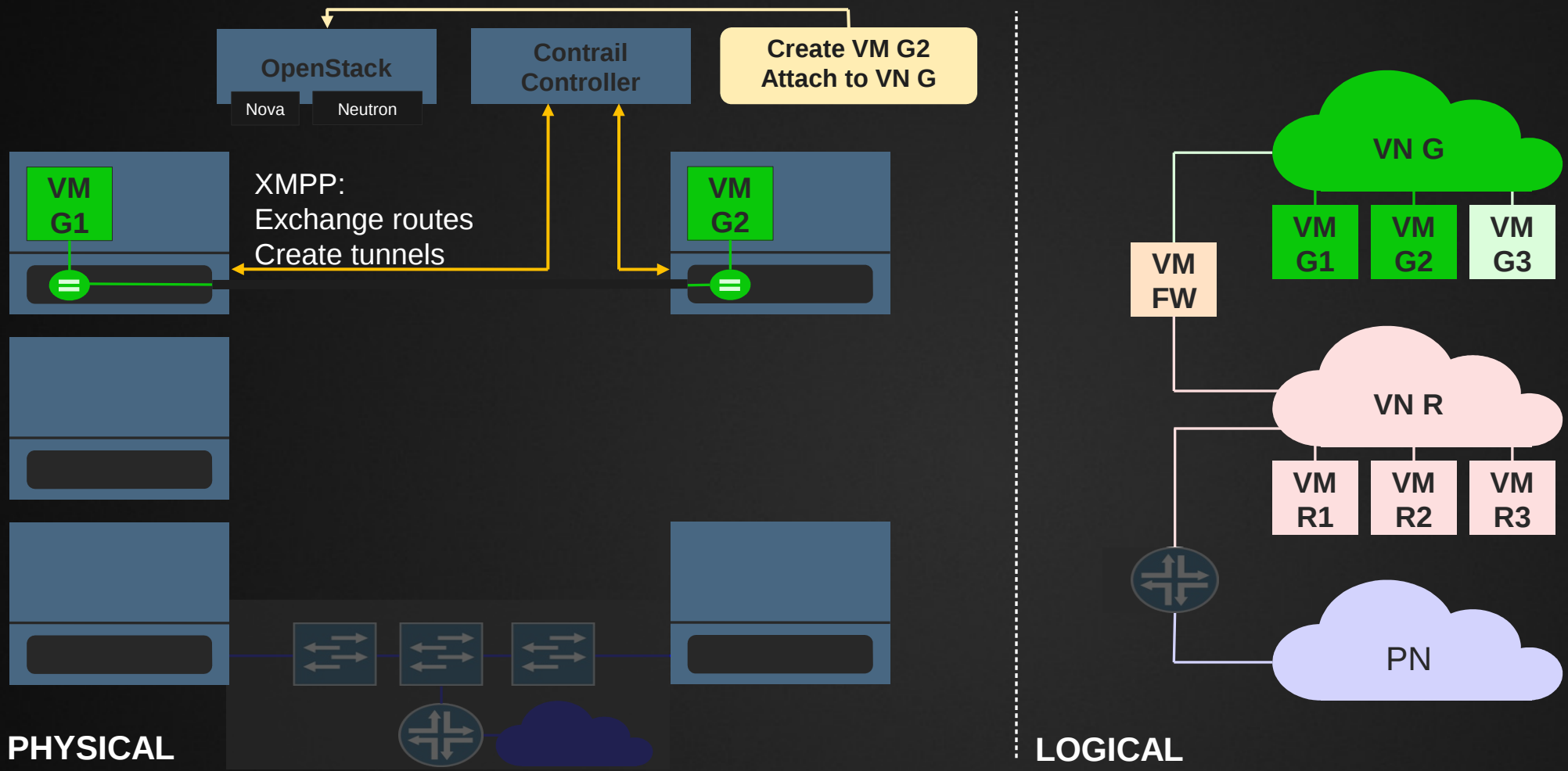
用户新建 VM G2


OpenStack
Nova
Neutron
Contrail Controller
Create VM G2 Attach to VN G
VM G1
VM G2
Nova: Create VM
PHYSICAL
VN G
VM G1
VM G2
VM G3
VM FW
VN R
VM R1
VM R2
VM R3
PN
LOGICAL

用户新建 VM G2

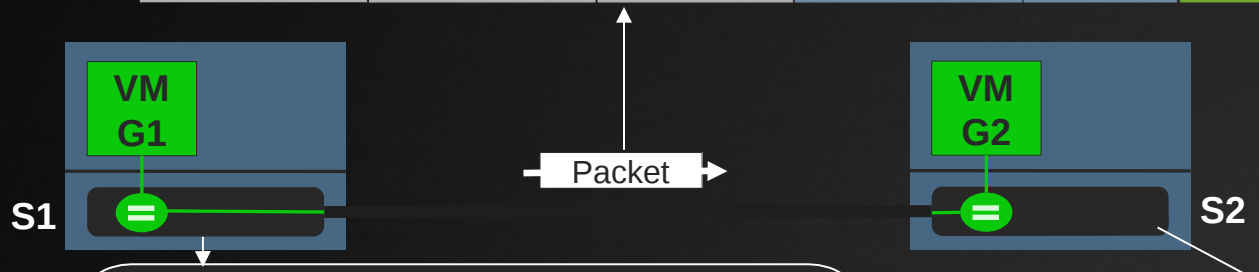

OpenStack
Nova
Neutron
Contrail Controller
Create VM G2
Attach to VN G
Neutron:
Attach VM to VN
XMPP:
Create routing-instance
VM G1
VM G2
PHYSICAL
VN G
VM G1
VM G2
VM G3
VM FW
VN R
VM R1
VM R2
VM R3
PN
LOGICAL

CONTRAIL 通过指令在两个服务器之间建立隧道



用户的数据包在隧道中转发的情况

Payload	Inner IP header		MPLS	GRE	Outer IP header		Ethernet	
	Source IP	Dest IP	Label		Source IP	Dest IP	Source MAC	Dest MAC
...	VM G1	VM G2	L2	...	Server S1	Server S2	Server S1	Server S2



Green routing-instance IP FIB

IP prefix	Nexthop
VM G1	Virtual ethernet port to VM G1
VM G2	Push label L2 + GRE encaps to server S2

Global MPLS FIB

MPLS label	Nexthop
L1	Pop + Green routing-instance

Global IP FIB

IP prefix	Nexthop
Server S2	Physical ethernet port

Green routing-instance IP FIB

IP prefix	Nexthop
VM G1	Push label L1 GRE encaps to server S1
VM G2	Virtual ethernet port to VM G2

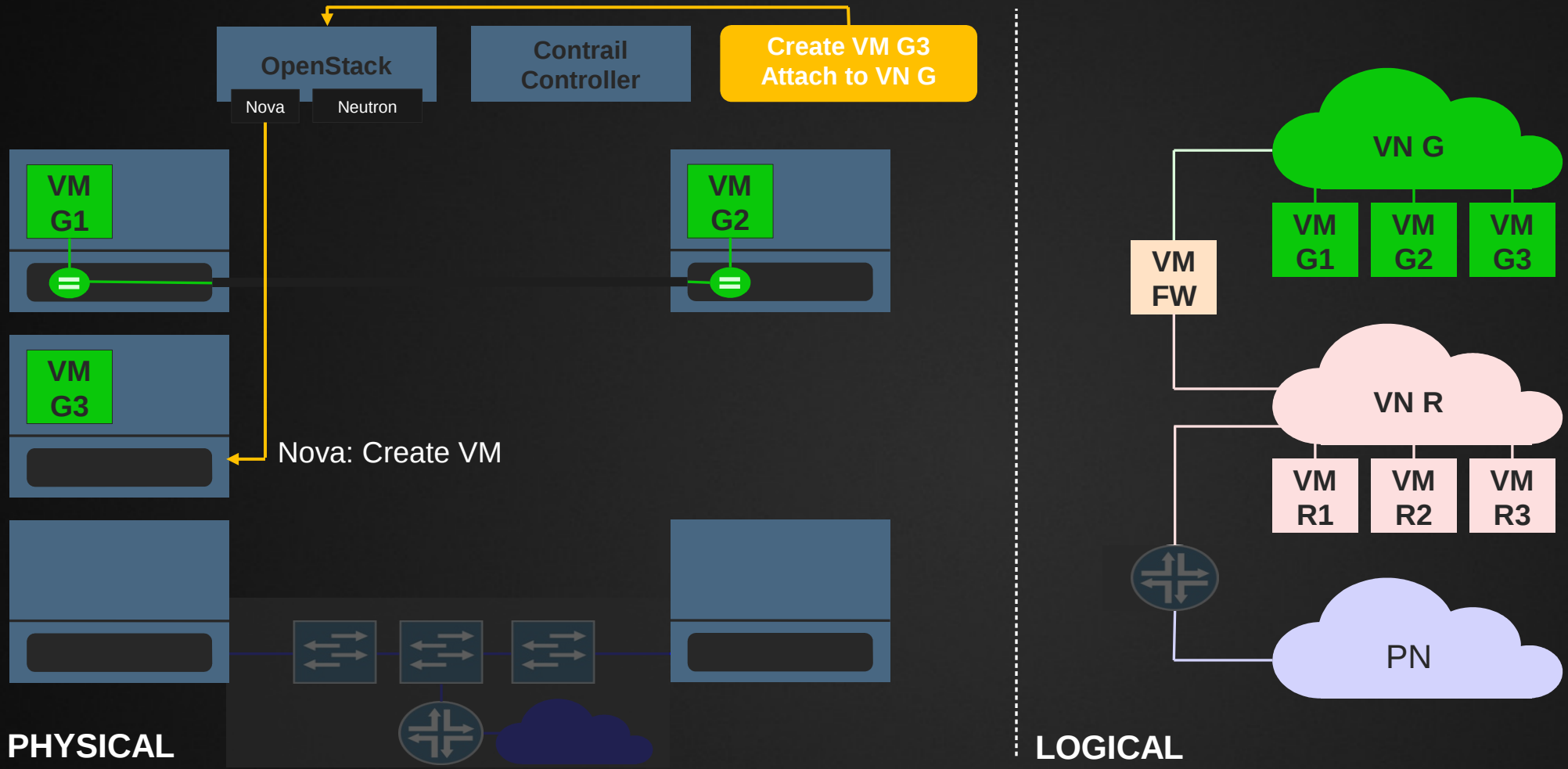
Global MPLS FIB

MPLS label	Nexthop
L2	Pop + Green routing-instance

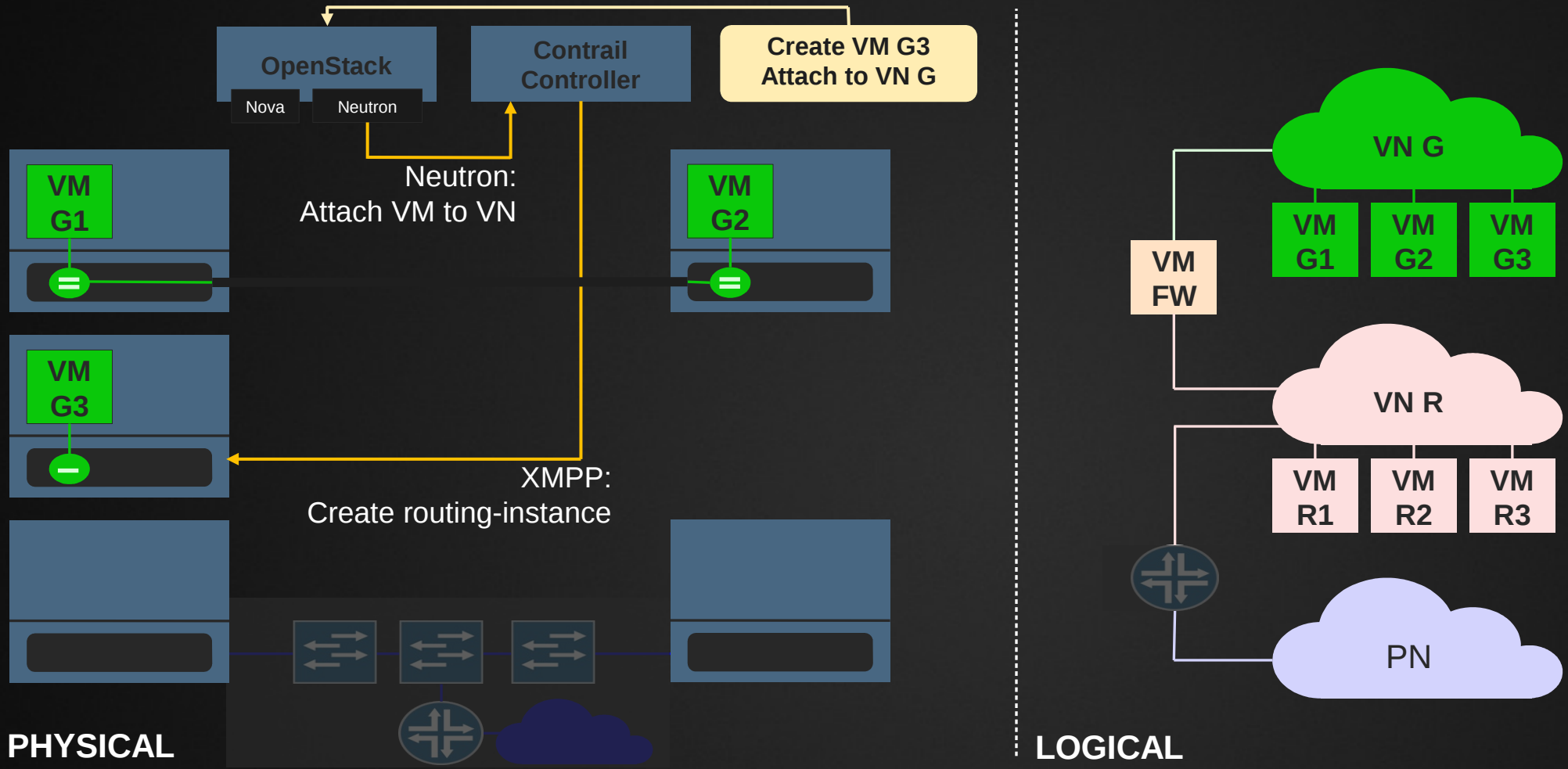
Global IP FIB

IP prefix	Nexthop
Server S1	Physical ethernet port

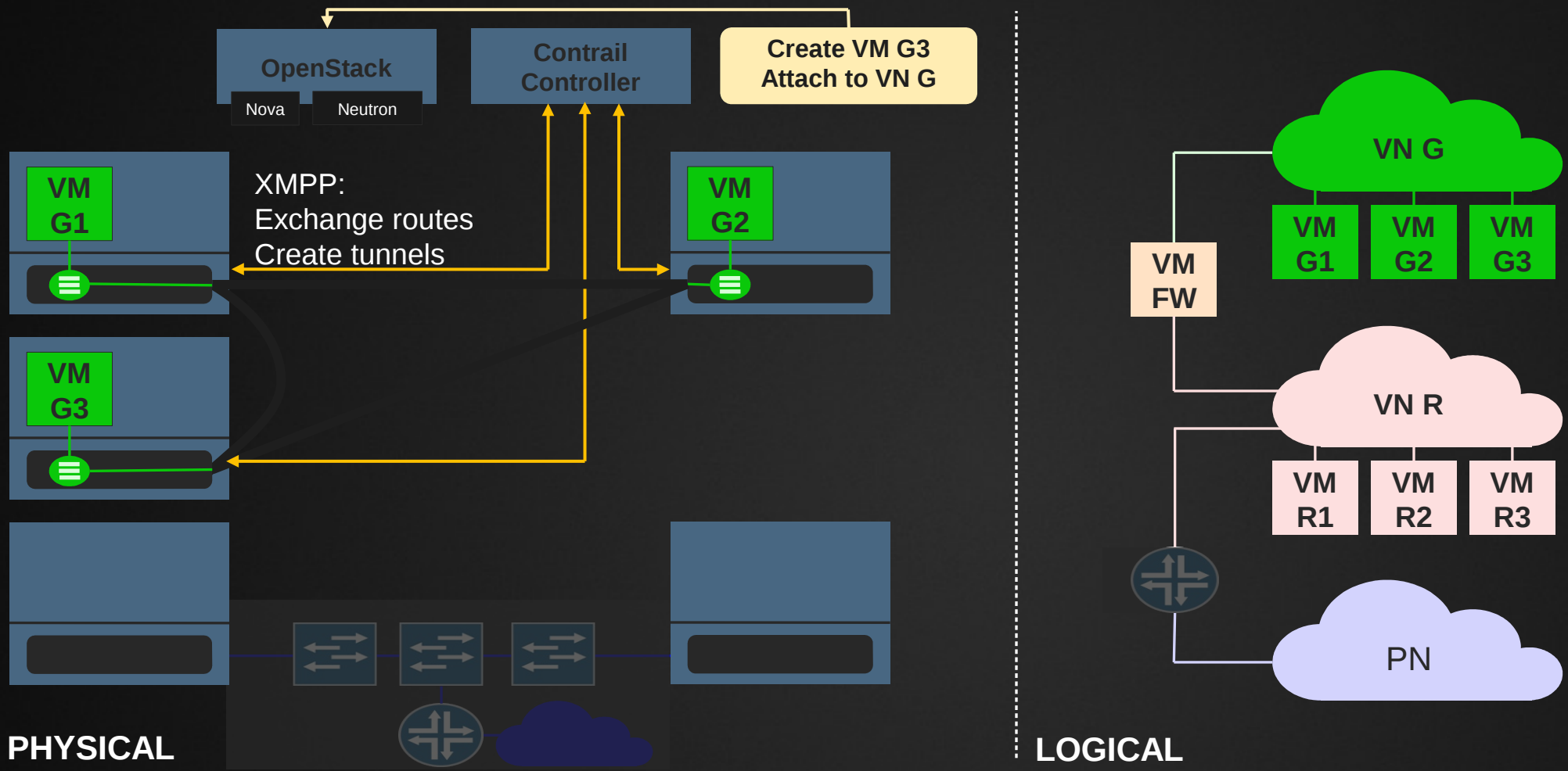
用户新建 VM G3


OpenStack
Nova
Neutron
Contrail Controller
Create VM G3 Attach to VN G
VM G1
VM G2
VM G3
Nova: Create VM
PHYSICAL
VN G
VM G1
VM G2
VM G3
VM FW
VN R
VM R1
VM R2
VM R3
PN
LOGICAL

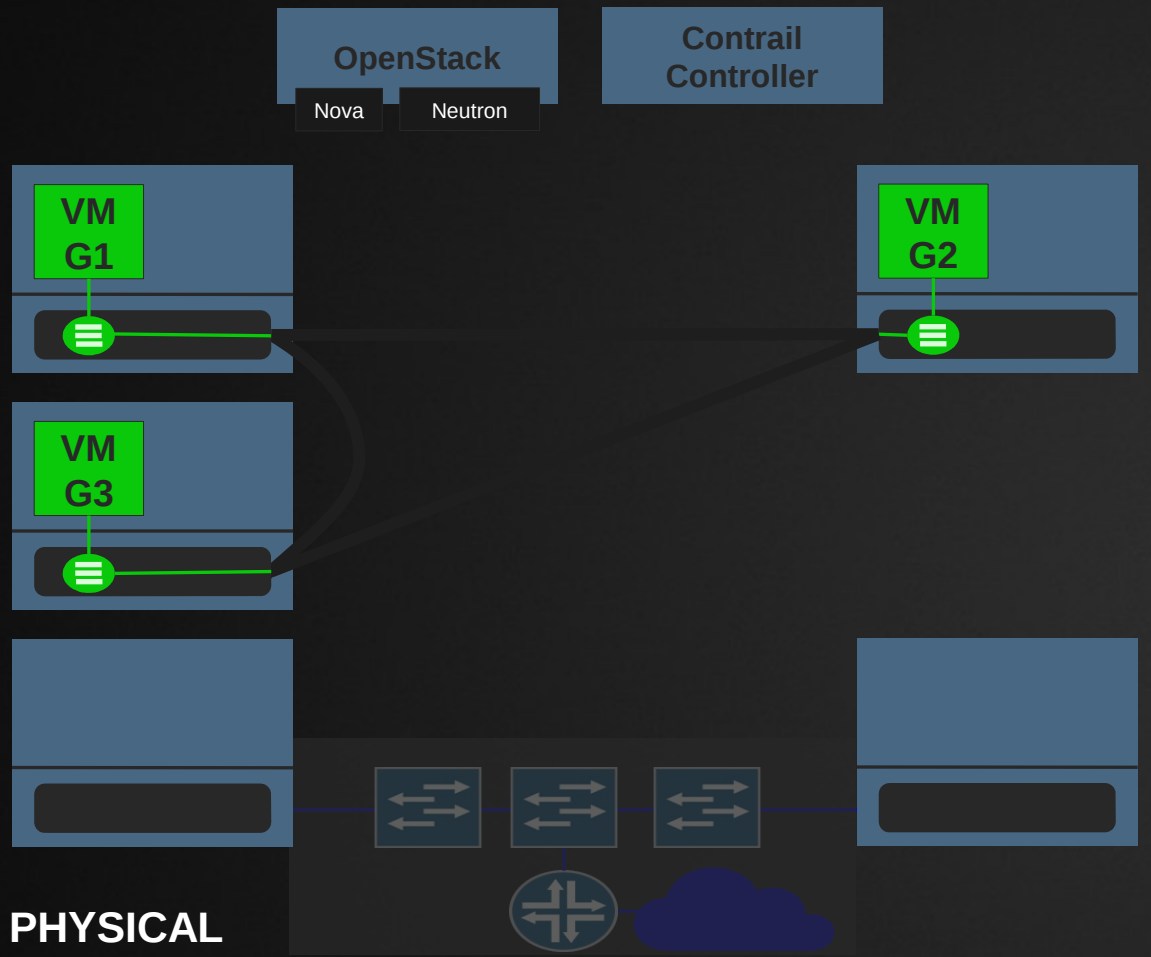
用户新建 VM G3

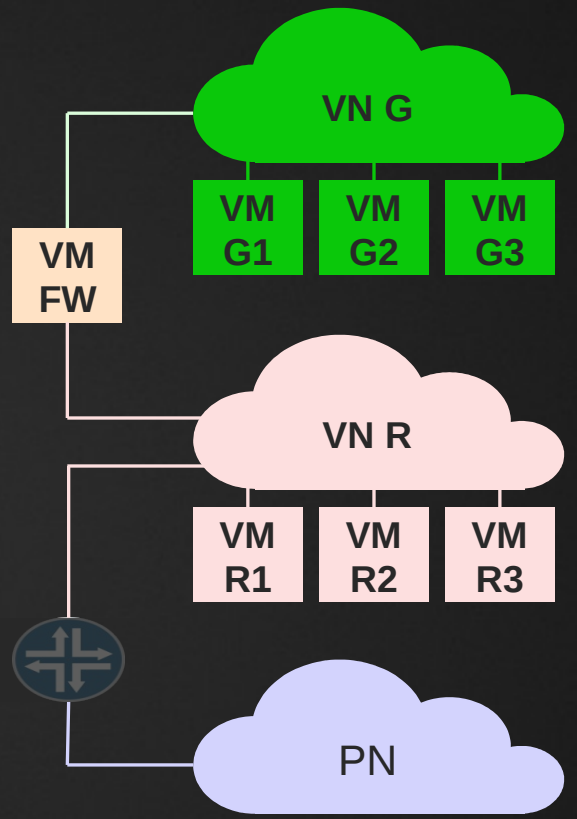

OpenStack
Nova
Neutron
Contrail Controller
Create VM G3
Attach to VN G
Neutron:
Attach VM to VN
VM G1
VM G2
VM G3
XMPP:
Create routing-instance
PHYSICAL
VN G
VM G1
VM G2
VM G3
VM FW
VN R
VM R1
VM R2
VM R3
PN
LOGICAL

CONTRAIL 在物理服务器之间再搭建两条隧道

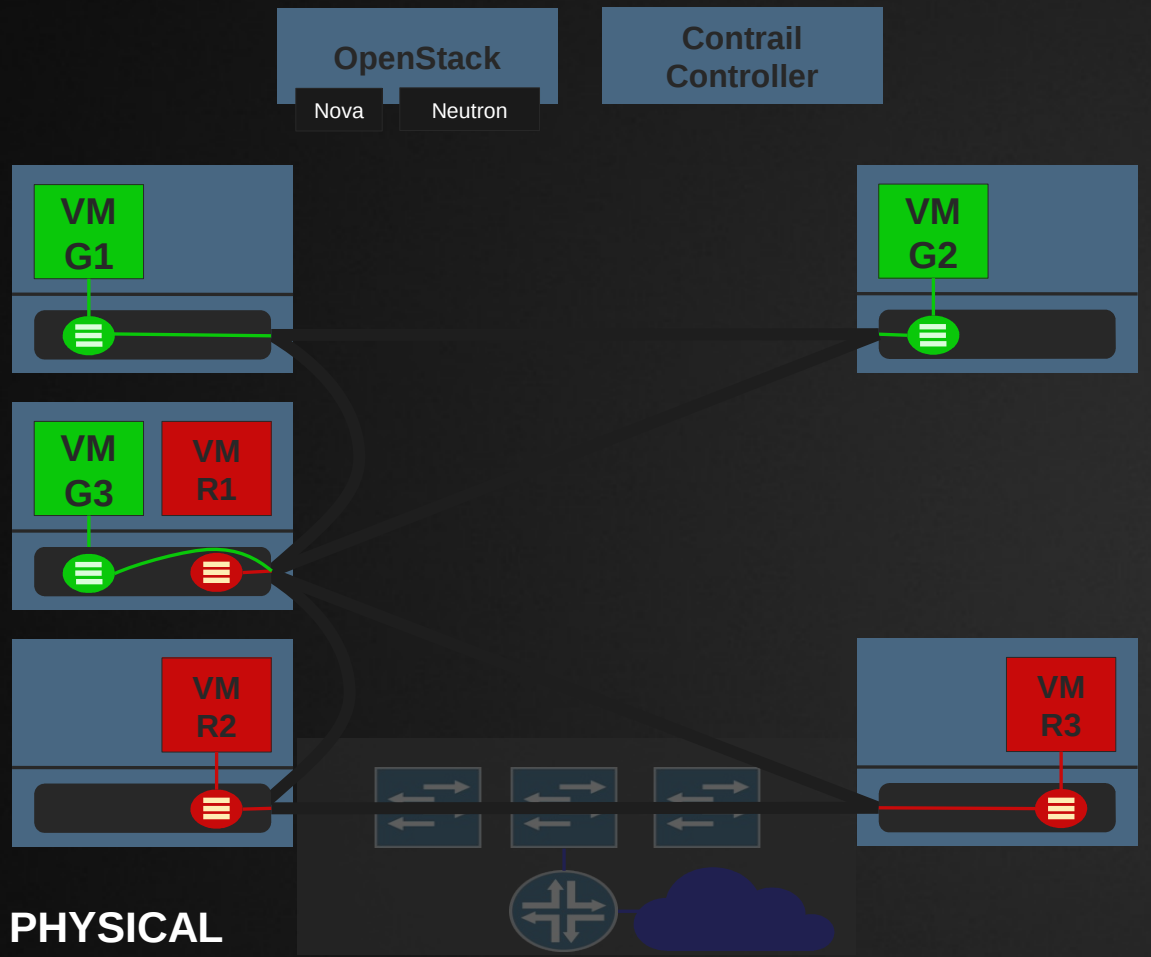


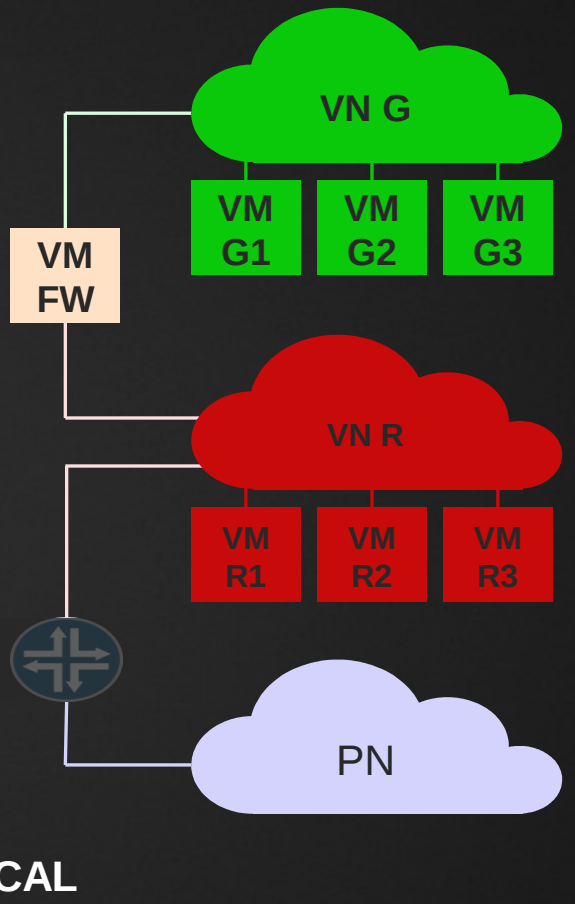
用户的最终状态


OpenStack
Nova
Neutron
Contrail Controller
VM G1
VM G2
VM G3
PHYSICAL

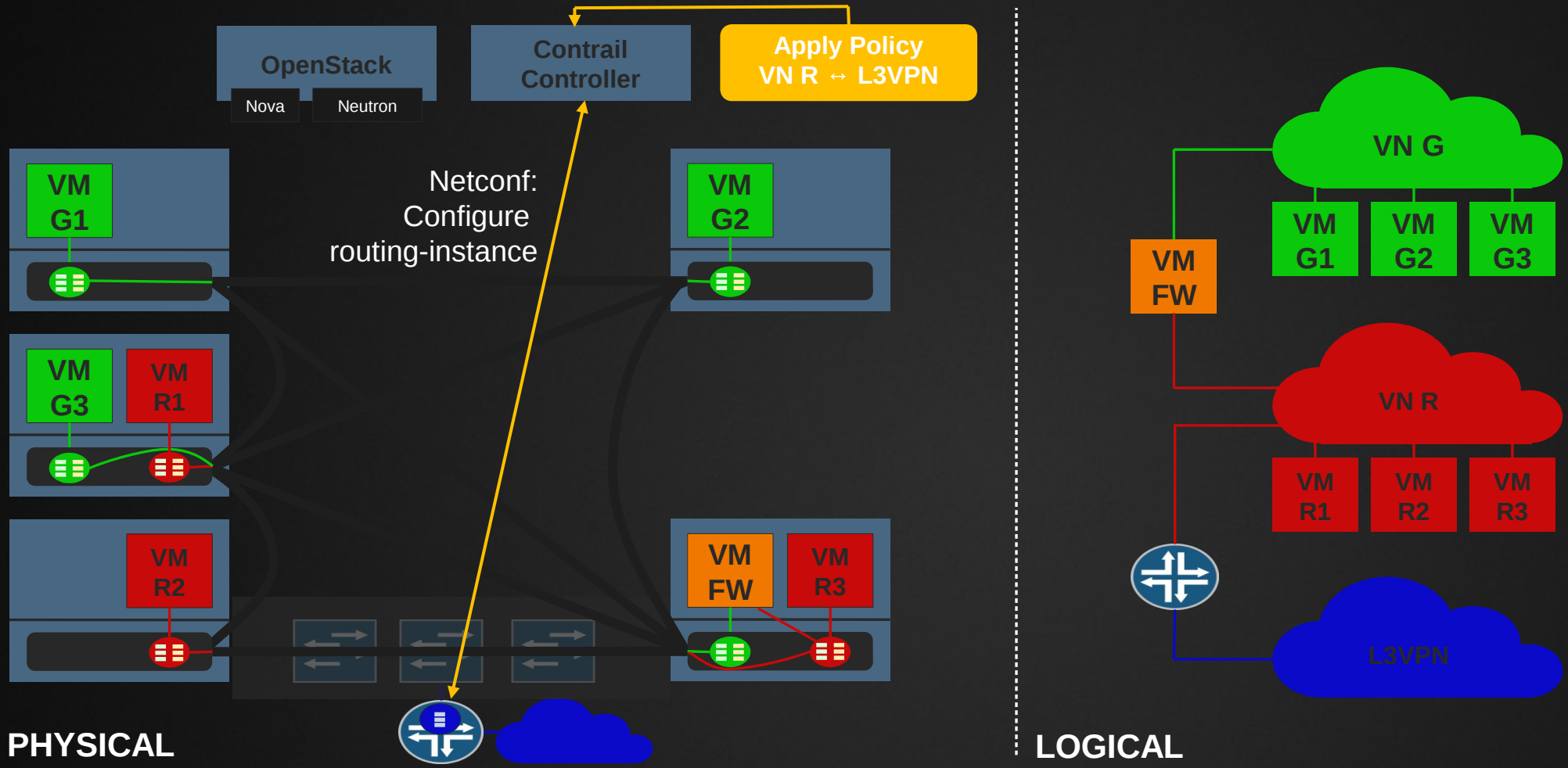

VN G
VM G1
VM G2
VM G3
VM FW
VN R
VM R1
VM R2
VM R3
PN
LOGICAL

两个不同用户均新建了 VM 以后

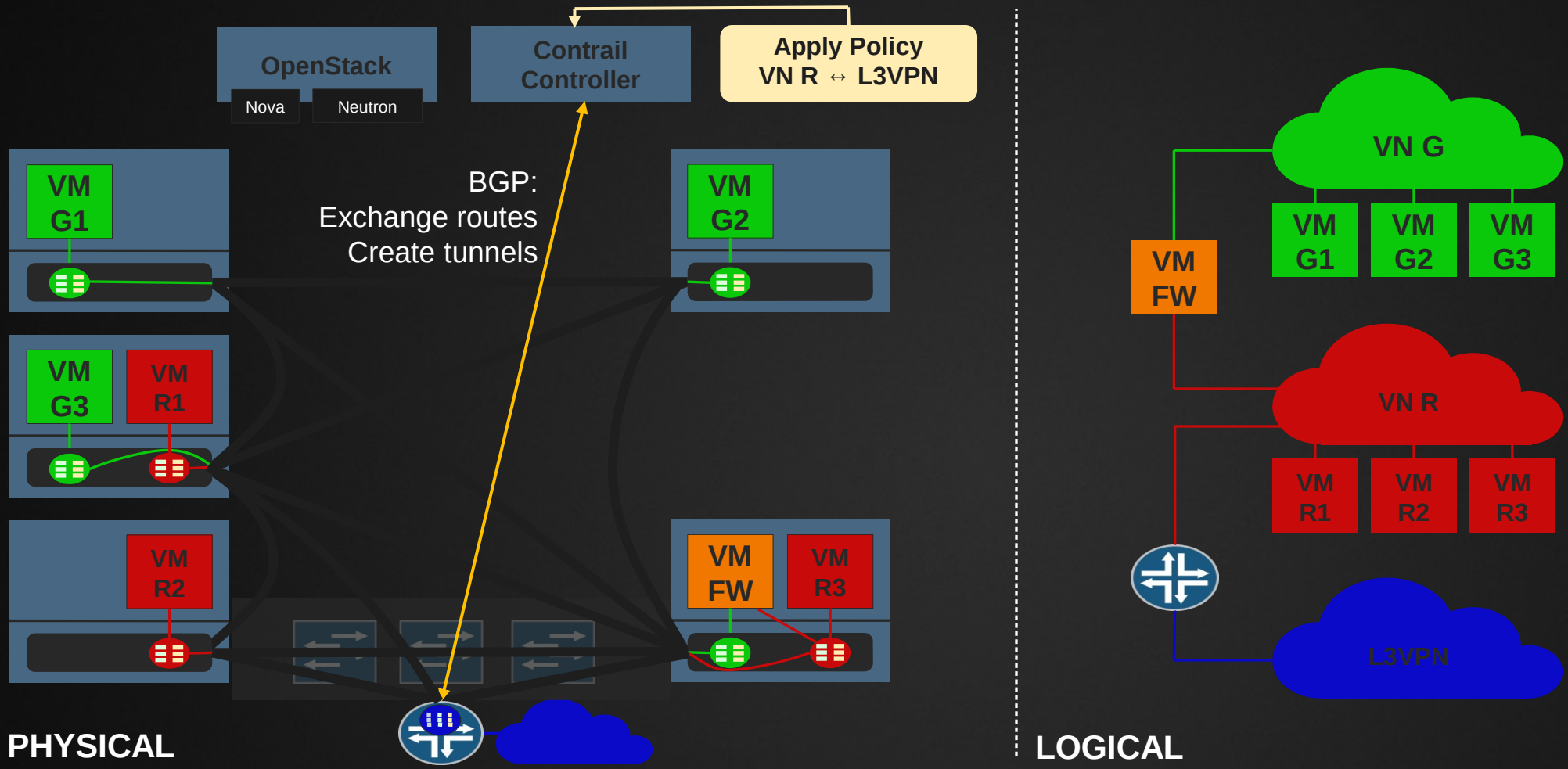




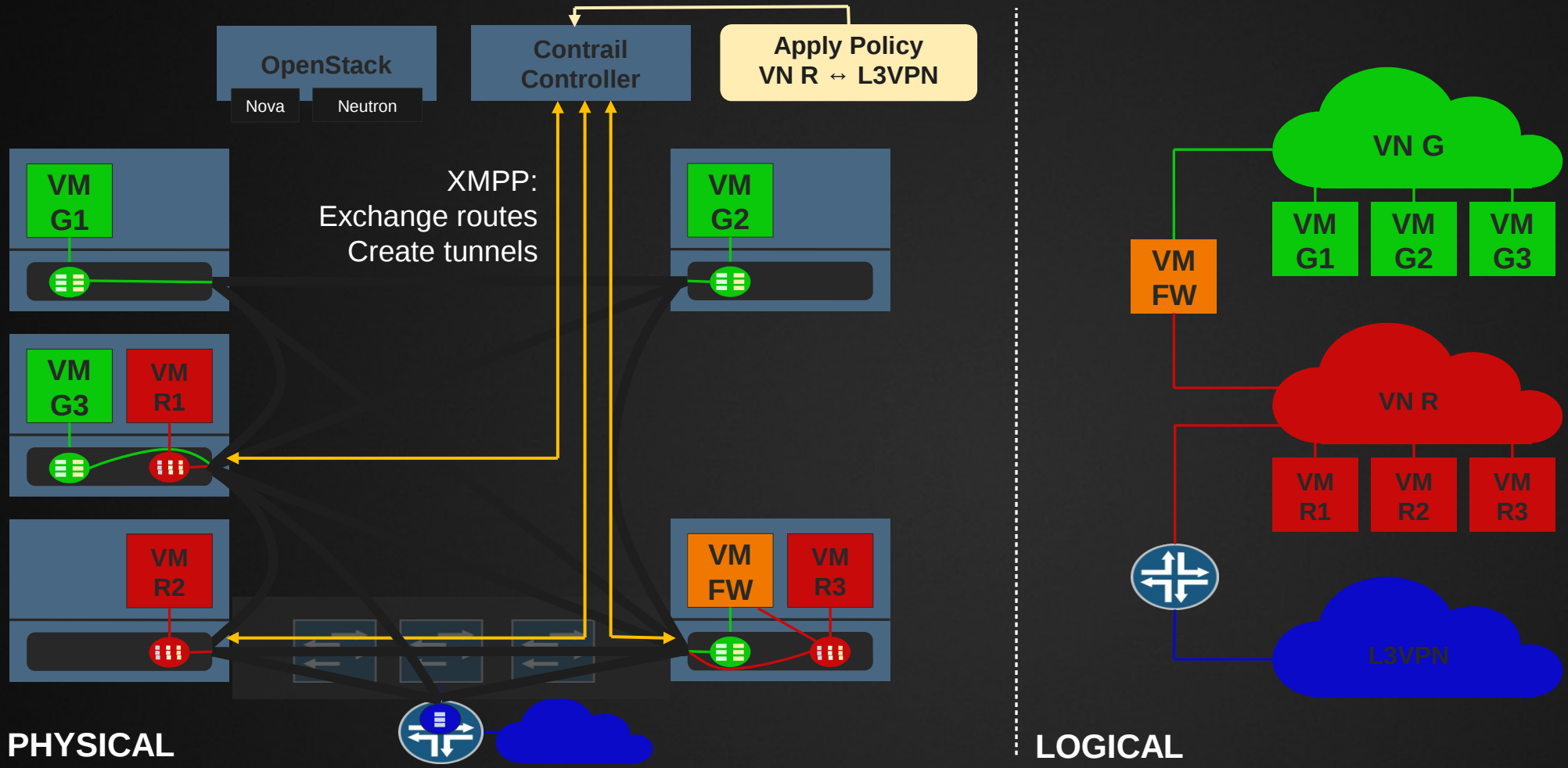
虚拟 VROUTER 和出口路由器之间建立隧道



虚拟 VROUTER 和出口路由器之间建立隧道



虚拟 VROUTER 和出口路由器之间建立隧道



可编程接口

API's 调用系统

- 所有的工作都是通过 API
  - 界面在调用 openstack 的 API
  - 所有的工作都是通过 API 完成
- 系统使用通过的编程语言
  - Python 和 Java libraries (others can be supported as there is interest)
  - 也可以是 curl
- 数据模型是公开的，有对应的文档

总结：CONTRAIL 是 SDN 解决方案的创新者

简单

- 可以与现有网络结合，节省升级和更换设备的成本
- SDN 的物理层面的架构简单化
- 简化网络中的组件，通过虚拟化实现

开放

- 开发那个的平台
- **所有的 Hypervisors 上运行的协议都是标准的**
- **可以云平台可以完好的结合**

智能化

- **自动更新云平台虚拟网络的结构**
- 可以与云平台进行结合使用
- 自带的分析系统可以分析流量

THANK YOU