

从Docker到Kubernetes 第12周

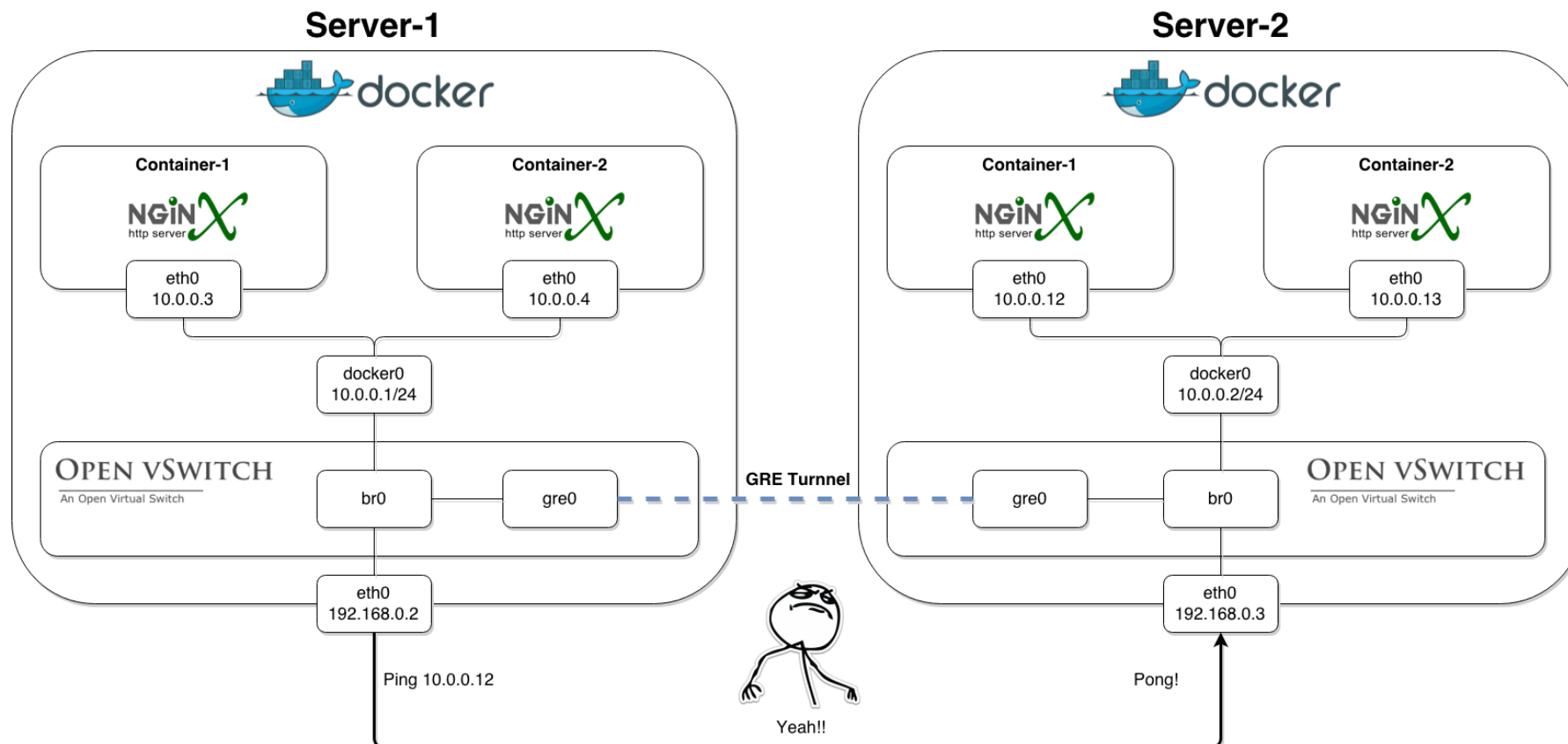
DATAGURU专业数据分析社区

【声明】 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散布，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>

- 可选的几种网络方案
- Flannel网络方案



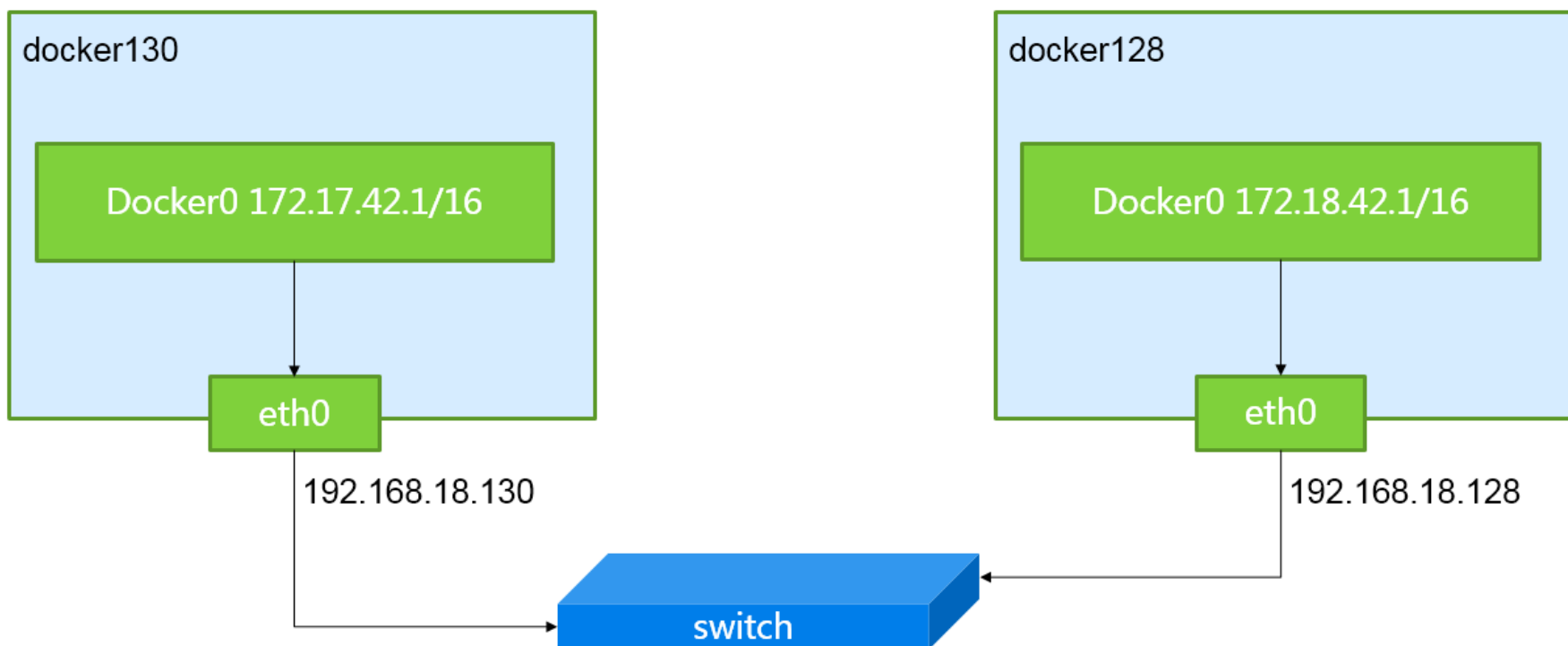
可选的几种网络方案 openvswitch

是一种主流的虚拟化大二层技术

灵活
对现有物理网络没要求
业界主流

软件封装导致性能低
复杂度比较高
Trouble Shooting难

Linux路由机制打通网络

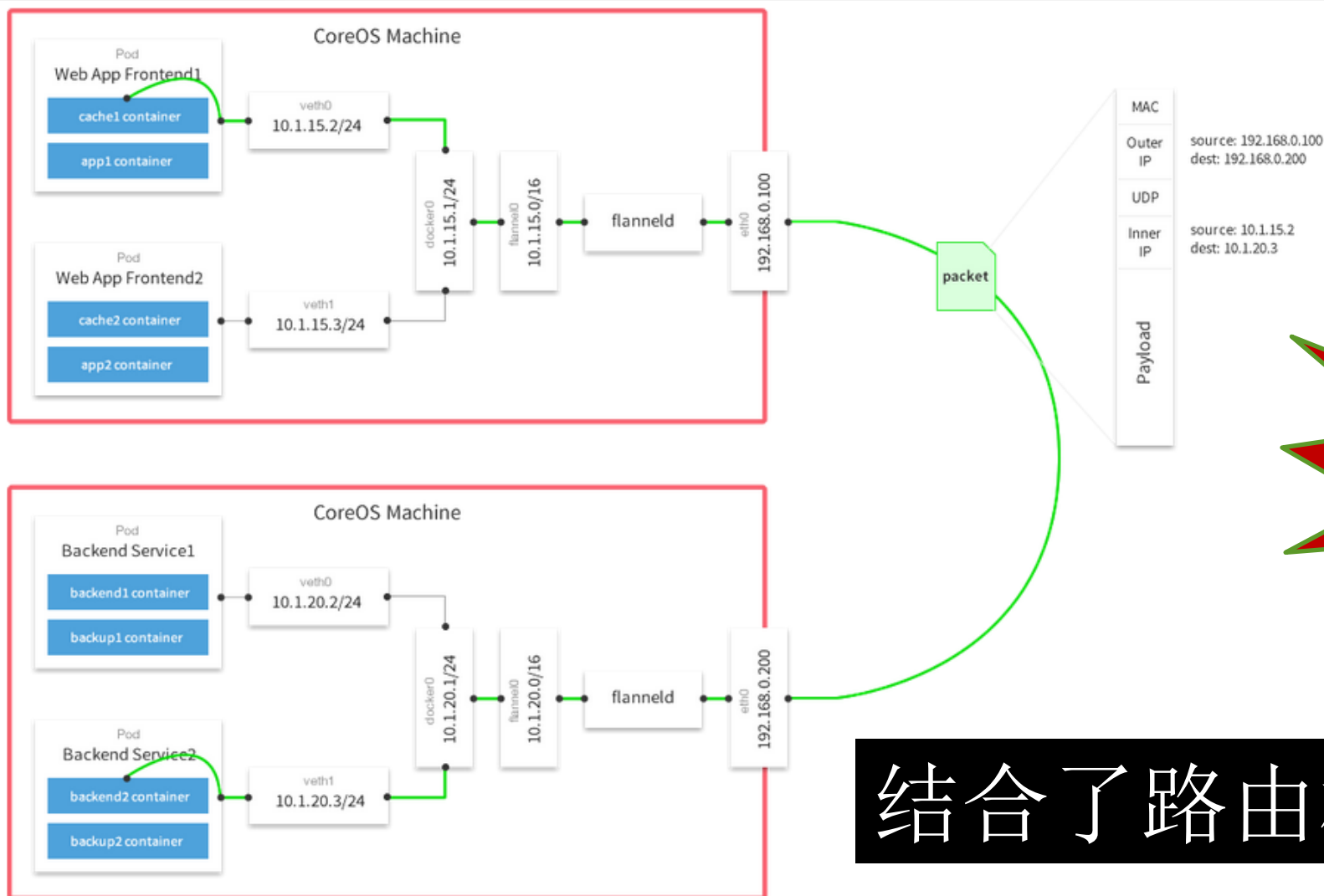


可选的几种网络方案 路由技术

一种结合了Linux路由功能的“正统”网络技术

常规路由技术
使用传统网络技术
简单
高性能

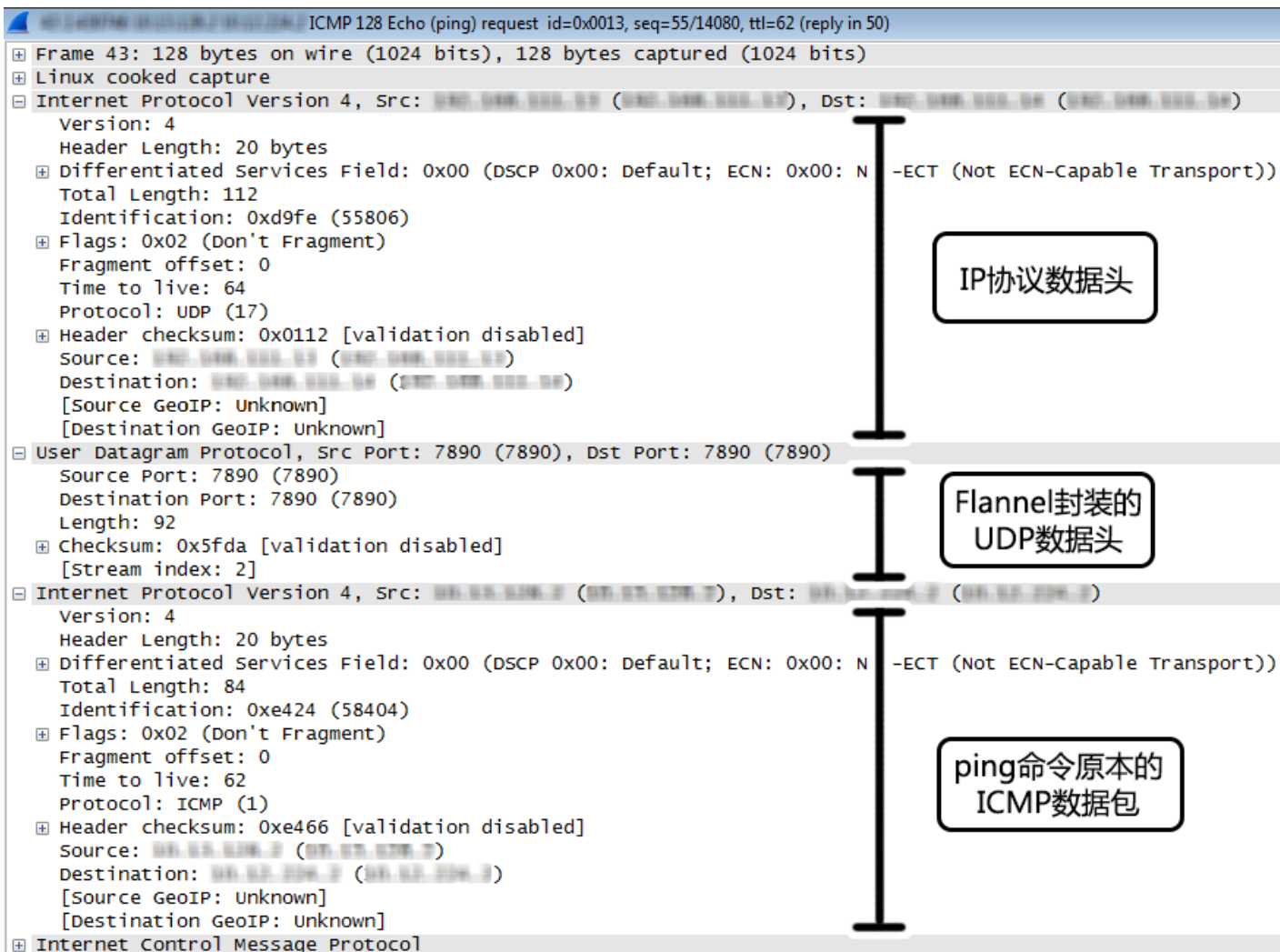
与现有网络融为一体
灵活性低



设计巧妙
使用简单

结合了路由机制与隧道技术

原始数据是在起始节点的Flannel服务上进行UDP封装的，投递到目的节点后就被另一端的Flannel服务还原成了原始的数据包，两边的Docker服务都感觉不到这个过程的存在。



用Etcd存储子网拓扑与路由表

怎么确保每个Node上的Docker引擎用不同的IP地址段，这个事情看起来很诡异，但真相十分简单。其实只是单纯的因为Flannel通过Etcd分配了每个节点可用的IP地址段后，偷偷的修改了Docker的启动参数，见下图。

```
core@ip-172-31-14-97 / $ ps aux | grep '172.17.18.1' | grep docker
root      21740  0.0  1.7 215388 17604 ?        Ssl  00:40   0:00 docker --daemon --host=fd:// --bip=172.17.18.1/24 --mtu=8973 --ip-masq=false
core@ip-172-31-14-97 / $
```

这个IP范围是由Flannel自动分配的，由Flannel通过保存在Etcd服务中的记录确保它们不会重复。

```
core@ip-172-31-14-97 / $ etcdctl ls /coreos.com/network/subnets
/coreos.com/network/subnets/172.17.18.0-24
/coreos.com/network/subnets/172.17.46.0-24
/coreos.com/network/subnets/172.17.61.0-24
core@ip-172-31-14-97 / $ etcdctl get /coreos.com/network/subnets/172.17.18.0-24
{"PublicIP":"172.31.14.97"}
core@ip-172-31-14-97 / $ etcdctl get /coreos.com/network/subnets/172.17.46.0-24
{"PublicIP":"172.31.14.96"}
core@ip-172-31-14-97 / $ etcdctl get /coreos.com/network/subnets/172.17.61.0-24
{"PublicIP":"172.31.30.200"}
```

需要在每台Node上都安装flannel。flannel软件的下载地址为<https://github.com/coreos/flannel/releases>。将下载的压缩包flannel-<version>-linux-amd64.tar.gz解压，把二进制文件flanneld和mk-docker-opts.sh复制到/usr/bin（或其他PATH环境变量中的目录），即可完成对flannel的安装。

Flannel网络方案详解

使用systemd系统为例对flanneld服务进行配置。

编辑服务配置文件/usr/lib/systemd/system/flanneld.service:

```
[Unit]
Description=Flanneld overlay address etcd agent
After=network.target
Before=docker.service
[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/flanneld
EnvironmentFile=-/etc/sysconfig/docker-network
ExecStart=/usr/bin/flanneld -etcd-endpoints=${FLANNEL_ETCD} $FLANNEL_OPTIONS
[Install]
RequiredBy=docker.service
WantedBy=multi-user.target
```

编辑配置文件/etc/sysconfig/flanneld，设置etcd的URL地址（Master上安装的etcd节点地址）

```
FLANNEL_ETCD= " http://192.168.1.128:4001 "
```

在启动flannel之前，需要在etcd中添加一条网络配置记录，这个配置将用于flannel分配给每个Docker的虚拟IP地址段。

```
# etcdctl set /coreos.com/network/config '{ " Network " : " 10.1.0.0/16 " }'
```

由于flannel将覆盖docker0网桥，所以如果Docker服务已启动，则停止Docker服务，然后systemctl restart flanneld

在每个节点上执行：

```
sudo mk-docker-opts.sh -i  
source /run/flannel/subnet.env  
sudo rm /var/run/docker.pid  
sudo ifconfig docker0 ${FLANNEL_SUBNET}
```

重新启动一次Docker，这样配置就完成了。

Flannel很新颖很不错

实际在可控的网络里，路由的方案则性能最好，适合生产用，配合Quagga 来做到自动路由更新

Thanks

FAQ时间