



# shoutd

让Docker容器也可以叽叽喳喳

宋晓寒

[chef@dark.kitchen](mailto:chef@dark.kitchen)



# What

shoutd (读作shouted) 是一个Docker的网络功能插件, 可以为Docker容器提供**跨主机**的多播 (multicast) 能力.

# Why

多播技术在服务发现, 集群高可用和弹性分布式系统等领域有广泛的应用.

- \* Elasticsearch的自组网集群

- \* 智能家居和物联网领域使用的设备发现协议 (mDNS / DSSP)

<http://arduino.local>

- \* 高可用集群中用于消除单点故障的浮动IP技术 (VRRP / CARP)

[keepalived](#)

# Why

Docker官方的overlay网络只能支持同一主机上的容器之间的多播通信, 不支持跨主机的情况.[\[1\]](https://github.com/docker/libnetwork/issues/552)

在基础设施层, 绝大多数的云主机厂商不提供原生的多播能力, 包括AWS VPC.

[\[1\] https://github.com/docker/libnetwork/issues/552](https://github.com/docker/libnetwork/issues/552)

## Multicast in Overlay driver #552



**nicklaslof** opened this issue on Sep 21, 2015 · 14 comments

# Why not...

**Calico** 工作在网络层 (Layer 3), 非协议透明, 官方声明只支持TCP, UDP, ICMP 和 ICMPv6.

**Weave** 和 **Flannel** 工作在用户态, 大负载时性能较差, 实现复杂.

**Docker overlay network** 仅提供部分支持.

# Why not...

**Calico** 工作在网络层 (Layer 3), 非协议透明, 官方声明只支持TCP, UDP, ICMP 和 ICMPv6.

**Weave** 和 **Flannel** 工作在用户态, 大负载时性能较差, 实现复杂.

**Docker overlay network** 仅提供部分支持.

**Fun!**

# How

Virtual EThernet device (VETH) 管道, 连接容器与外部世界

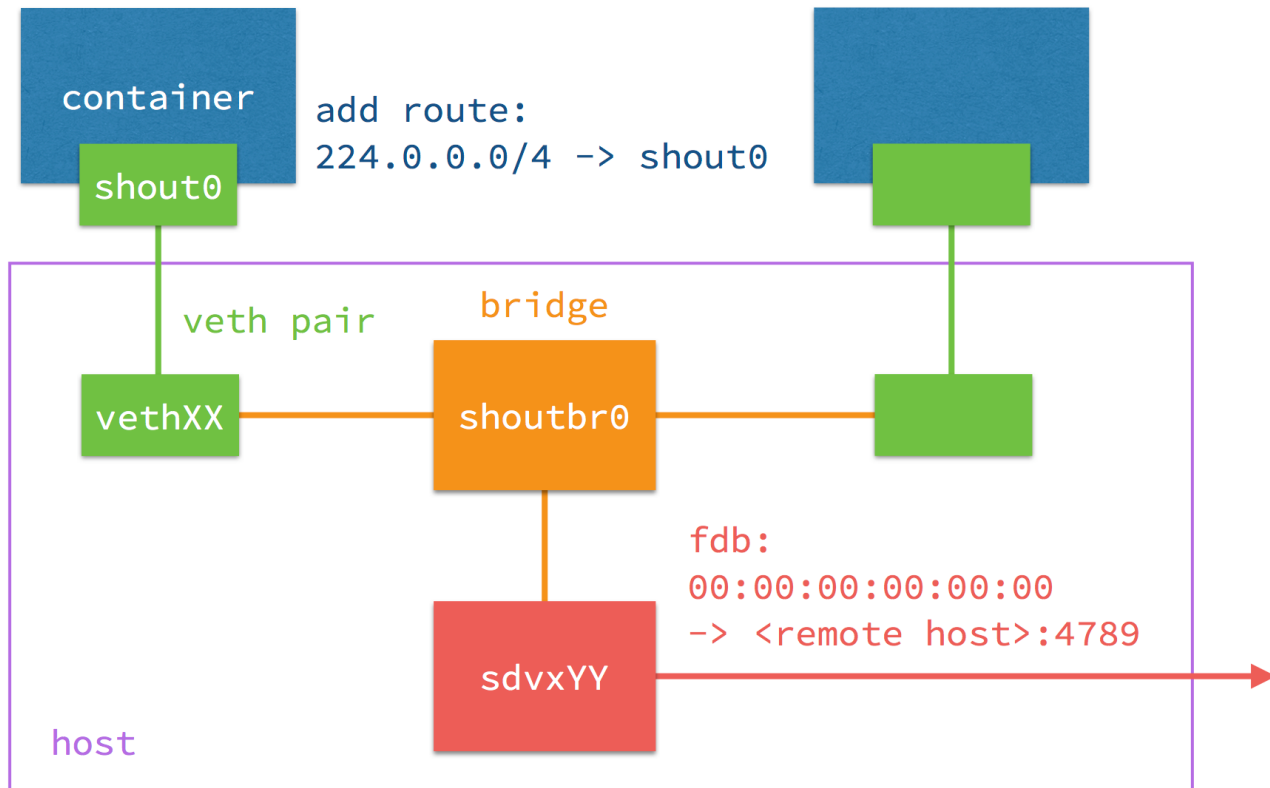
Linux network namespace 避免互相干扰

Linux software bridge 连接, 交换

Virtual eXtensible Local Area Network (VXLAN) 封装, 传输

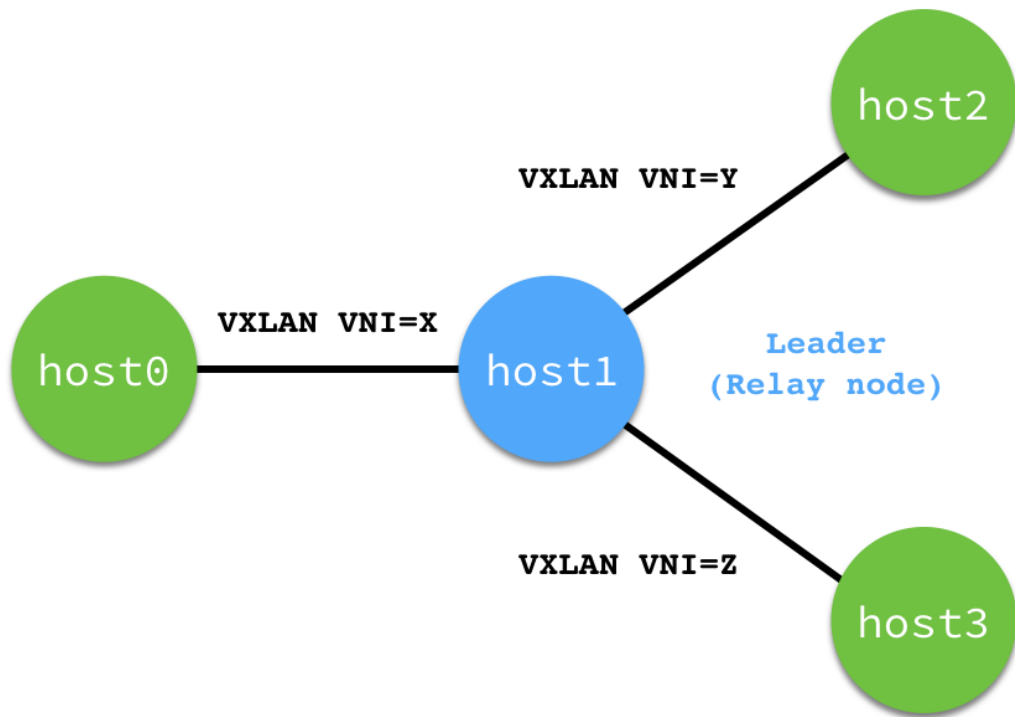
Raft Consensus Algorithm 谁才是头儿

# How





# How



# How to...

安装shoutd

```
pip3 install shoutd
```

在每个Swarm工作节点上启动shoutd

```
shoutdctl start <advertise_address> [--peer-addresses=<...>]
```

创建shoutd网络

```
docker network create --driver shoutd foo
```

加入网络开始叽叽喳喳

```
docker run -it --rm --network=foo ubuntu bash
```

# Demo

```
ubuntu@node2-172-31-16-148:~$ docker ps -a
```

| CONTAINER ID | IMAGE              | COMMAND                | CREATED           | STATUS           |
|--------------|--------------------|------------------------|-------------------|------------------|
| 81742a61c81e | lab                | "/bin/bash"            | About an hour ago | Up About an hour |
| e3d5e4df5d93 | darkkitchen/shoutd | "shoutd 172.31.16.148" | 2 hours ago       | Up 2 hours       |
| 27b0f1cd5611 | swarm              | "/swarm join --advert" | 3 hours ago       | Up 3 hours       |
| 914f53c5fb18 | swarm              | "/swarm manage -H :44" | 3 hours ago       | Up 3 hours       |

# Demo

```
root@81742a61c81e:/# iperf -u -c 224.1.1.1 -p 5000 -t 8 -T 2 -i 1
-----
Client connecting to 224.1.1.1, UDP port 5000
Sending 1470 byte datagrams
Setting multicast TTL to 2
UDP buffer size: 208 KByte (default)
-----
[  3] local 10.0.0.4 port 57202 connected with 224.1.1.1 port 5000
[ ID] Interval           Transfer     Bandwidth
[  3] 0.0- 1.0 sec      129 KBytes  1.06 Mbits/sec
[  3] 1.0- 2.0 sec      128 KBytes  1.05 Mbits/sec
[  3] 2.0- 3.0 sec      128 KBytes  1.05 Mbits/sec
[  3] 3.0- 4.0 sec      128 KBytes  1.05 Mbits/sec
[  3] 4.0- 5.0 sec      128 KBytes  1.05 Mbits/sec
[  3] 5.0- 6.0 sec      128 KBytes  1.05 Mbits/sec
[  3] 6.0- 7.0 sec      129 KBytes  1.06 Mbits/sec
[  3] 7.0- 8.0 sec      128 KBytes  1.05 Mbits/sec
[  3] 0.0- 8.0 sec    1.00 MBytes  1.05 Mbits/sec
[  3] Sent 715 datagrams
root@81742a61c81e:/# _
```

# Demo

```
root@3e87c2daac1b:/# iperf -u -s -B 224.1.1.1 -p 5000 -i 1
```

```
-----  
Server listening on UDP port 5000  
Binding to local address 224.1.1.1  
Joining multicast group 224.1.1.1  
Receiving 1470 byte datagrams  
UDP buffer size: 208 KByte (default)  
-----
```

```
[ 3] local 224.1.1.1 port 5000 connected with 10.0.0.4 port 57202
```

| [ ID] | Interval     | Transfer    | Bandwidth      | Jitter   | Lost/Total | Datagrams |
|-------|--------------|-------------|----------------|----------|------------|-----------|
| [ 3]  | 0.0- 1.0 sec | 128 KBytes  | 1.05 Mbits/sec | 0.059 ms | 0/ 89      | (0%)      |
| [ 3]  | 1.0- 2.0 sec | 128 KBytes  | 1.05 Mbits/sec | 0.073 ms | 0/ 89      | (0%)      |
| [ 3]  | 2.0- 3.0 sec | 128 KBytes  | 1.05 Mbits/sec | 0.030 ms | 0/ 89      | (0%)      |
| [ 3]  | 3.0- 4.0 sec | 128 KBytes  | 1.05 Mbits/sec | 0.046 ms | 0/ 89      | (0%)      |
| [ 3]  | 4.0- 5.0 sec | 128 KBytes  | 1.05 Mbits/sec | 0.073 ms | 0/ 89      | (0%)      |
| [ 3]  | 5.0- 6.0 sec | 129 KBytes  | 1.06 Mbits/sec | 0.047 ms | 0/ 90      | (0%)      |
| [ 3]  | 6.0- 7.0 sec | 128 KBytes  | 1.05 Mbits/sec | 0.071 ms | 0/ 89      | (0%)      |
| [ 3]  | 7.0- 8.0 sec | 128 KBytes  | 1.05 Mbits/sec | 0.089 ms | 0/ 89      | (0%)      |
| [ 3]  | 0.0- 8.0 sec | 1.00 MBytes | 1.05 Mbits/sec | 0.086 ms | 0/ 715     | (0%)      |

```
—
```

# Links

<https://github.com/xhs/shoutd>

<https://hub.docker.com/r/darkkitchen/shoutd/>

AWS

S U M M I T

beijing

Q&A