

# Exclude Virtual Interfaces in IP Command

Apr 2023

In Kubernetes deployments, there are always a number of virtual interfaces, this isn't an issue until you are trying to look at the details of physical interfaces, and the virtual ones scroll the physical ones off the page. You can use a grep command to hide the virtual interfaces to stop this this though.

Lets say I use an IP command to list the interfaces on a Linux instance.

```
ip -br -c a
```

I will see something similar to the following output.

lo	UNKNOWN	127.0.0.1/8 ::1/128
eth2	UP	11.0.200.2/16 fe80::76fe:48ff:fe45:e65e/64
eth3	DOWN	
eth4	UP	12.0.200.4/16 fe80::76fe:48ff:fe45:e660/64
eth5	DOWN	
eth0	UP	10.43.30.200/24 fe80::76fe:48ff:fe45:1eee/64
eth1	DOWN	
internalk3s0	UNKNOWN	192.168.255.1/24
kube-ipvs0	DOWN	10.1.0.1/32 10.1.58.55/32 10.1.139.197/32 10.1.224.155/32 10.1.15.231/32
10.1.234.150/32	10.1.224.182/32	10.1.82.169/32 10.1.88.39/32 10.1.113.246/32 10.1.0.10/32 10.1.92.105/32
10.1.242.227/32	10.1.136.136/32	10.1.147.213/32 10.1.105.61/32 10.1.14.221/32 10.1.62.150/32
192.168.255.1/32	10.1.142.50/32	10.1.227.44/32 10.1.178.20/32 10.1.37.6/32 10.1.220.215/32
10.1.46.190/32	10.1.188.12/32	10.1.27.143/32 10.1.75.75/32 10.1.45.171/32 10.1.242.58/32 10.1.146.97/32
10.1.119.162/32	10.1.133.120/32	10.1.193.167/32 10.1.44.38/32 10.1.226.76/32 10.1.60.158/32
flannel.1	UNKNOWN	10.0.0.0/32 fe80::c426:bfff:fe61:77b7/64
cnio	UP	10.0.0.1/24 fe80::98df:c5ff:fe4f:e664/64
veth5b959f30@if2	UP	fe80::483d:bcff:fe00:bf49/64
veth1b36bc0e@if2	UP	fe80::a479:9fff:feec:741b/64
vetheafccb8b@if2	UP	fe80::fc7a:eaff:fe49:ade4/64
veth6cab92f8@if2	UP	fe80::80a7:bfff:fe8f:504/64
vetha6ca6c5c@if2	UP	fe80::784a:afff:fe3e:511/64
vetha47d76c5@if2	UP	fe80::94e0:b3ff:fe8f:8b1b/64
veth78d7ae7b@if2	UP	fe80::1083:91ff:fe56:b7ad/64
vethbfbdc9e@if2	UP	fe80::d4a7:13ff:fe38:bb29/64
veth12caaf80@if2	UP	fe80::104e:19ff:fe18:c5bf/64
vethecaec700@if2	UP	fe80::5c86:38ff:fedb:5bfa/64
vethf759469f@if2	UP	fe80::a455:1eff:fed1:d90/64
veth5b8833a6@if2	UP	fe80::cad:7aff:fea3:d84a/64
veth18c2a62d@if2	UP	fe80::40d1:4fff:fe6e:d485/64
veth8e69b358@if2	UP	fe80::f0be:46ff:fee7:fb23/64
veth62243482@if2	UP	fe80::c52:58ff:fe6e:a289/64
veth4def4cbc@if2	UP	fe80::106e:beff:fe82:90d9/64
veth6583da97@if2	UP	fe80::e8dd:3fff:fe64:9071/64
veth30a60131@if2	UP	fe80::5097:44ff:fea9:9783/64
veth17bbb9b4@if2	UP	fe80::4067:f9ff:fe92:3663/64
vetha71f5d6e@if2	UP	fe80::6493:5cff:fef8:aadd/64
veth4171ad7f@if2	UP	fe80::a9:96ff:fe0e:311/64
vethffc2e81@if2	UP	fe80::302f:f9ff:fee5:8f50/64
vethce83eb59@if2	UP	fe80::c0f8:c5ff:fe07:f5e9/64
veth30897d87@if2	UP	fe80::c0ac:bfff:fe78:6080/64
vethe9389c32@if2	UP	fe80::d810:1fff:fe14:5285/64
vethde8e4fea@if2	UP	fe80::547d:b0ff:fe0b:1f1c/64
vethc3a4bead@if2	UP	fe80::9c59:dcff:fee7:6f8e/64
veth4f402974@if2	UP	fe80::f490:b3ff:fe57:44e6/64
veth0cd2a299@if2	UP	fe80::283b:a8ff:fe3e:a4de/64
vethb0dc4ed2@if2	UP	fe80::e07f:9aff:fea0:99ea/64
veth1c3166fa@if2	UP	fe80::8c40:cbff:fe20:ec8/64
veth2255cafe@if2	UP	fe80::8df:44ff:feb0:4386/64
vethb12cf176@if2	UP	fe80::48b8:e7ff:fe3e:c539/64
veth931f4dd1@if2	UP	fe80::9cad:31ff:fe6d:b6bf/64

vetha3c59775@if2	UP	fe80::70fe:feff:fe35:d5a7/64
veth2ceb7cf6@if2	UP	fe80::a008:eeff:feb6:4cba/64
veth2199c7d1@if2	UP	fe80::f4c1:e4ff:fefa:465c/64

Unfortunately, the vethxxxxxx interfaces will cause the physical interface details at the top of the list out of the view of the console, so what we really need is a way to filter this result. This can be done using grep and a filter to format our output.

```
ip -br -c a |grep -v veth
```

This will hide anything containing the letters veth.

lo	UNKNOWN	127.0.0.1/8 ::1/128
eth2	UP	11.0.200.2/16 fe80::76fe:48ff:fe45:e65e/64
eth3	DOWN	
eth4	UP	12.0.200.4/16 fe80::76fe:48ff:fe45:e660/64
eth5	DOWN	
eth0	UP	10.43.30.200/24 fe80::76fe:48ff:fe45:1eee/64
eth1	DOWN	
internalk3s0	UNKNOWN	192.168.255.1/24
kube-ipvs0	DOWN	10.1.0.1/32 10.1.58.55/32 10.1.139.197/32 10.1.224.155/32 10.1.15.231/32
10.1.234.150/32		10.1.224.182/32 10.1.82.169/32 10.1.88.39/32 10.1.113.246/32 10.1.0.10/32 10.1.92.105/32
10.1.242.227/32		10.1.136.136/32 10.1.147.213/32 10.1.105.61/32 10.1.14.221/32 10.1.62.150/32
192.168.255.1/32		10.1.142.50/32 10.1.227.44/32 10.1.178.20/32 10.1.37.6/32 10.1.220.215/32
10.1.46.190/32		10.1.188.12/32 10.1.27.143/32 10.1.75.75/32 10.1.45.171/32 10.1.242.58/32 10.1.146.97/32
10.1.119.162/32		10.1.133.120/32 10.1.193.167/32 10.1.44.38/32 10.1.226.76/32 10.1.60.158/32
flannel.1	UNKNOWN	10.0.0.0/32 fe80::c426:bfff:fe61:77b7/64
cni0	UP	10.0.0.1/24 fe80::98df:c5ff:fe4f:e664/64

No our list is much more readable.

We could refine this output with multiple exclusions using the -e switch:

```
ip -br -c a |grep -v -e "veth" -e "cni0" -e "flannel" -e "kube" -e "internal" -e"lo"
```

This give the following output.

eth2	UP	11.0.200.2/16 fe80::76fe:48ff:fe45:e65e/64
eth3	DOWN	
eth4	UP	12.0.200.4/16 fe80::76fe:48ff:fe45:e660/64
eth5	DOWN	
eth0	UP	10.43.30.200/24 fe80::76fe:48ff:fe45:1eee/64
eth1	DOWN	

This is a lot to type/remember, but you could make a script out of it.

From:  
<http://cameraangle.co.uk/> - WalkerWiki - [wiki.alanwalker.uk](http://wiki.alanwalker.uk)

Permanent link:  
[http://cameraangle.co.uk/doku.php?id=exclude\\_virtual\\_interfaces\\_in\\_ip\\_command](http://cameraangle.co.uk/doku.php?id=exclude_virtual_interfaces_in_ip_command)

Last update: 2023/04/07 17:01

