# How to put a bird on a docker container on arista




**#FRnOG33 - 2019-09-13**

#Docker #Arista #BIRD #route-server #NFV #SDN #bingo!

Arnaud Fenioux - **HOP***US*.net

*@afenioux*

# And voila!

# What is HOP*US*?

Born with the internet exchange principles we focus on providing quality routes to major networks that usually don't connect IXs.

We are trying to establish a virtuous model in which content can reach eyeballs with the best possible quality at a fair price.

Eyeballs networks receiving some payback on the traffic they carry and all members being enlisted to offer guarantees on their networks.

next-hop us!

# How **HOP*US works*?**

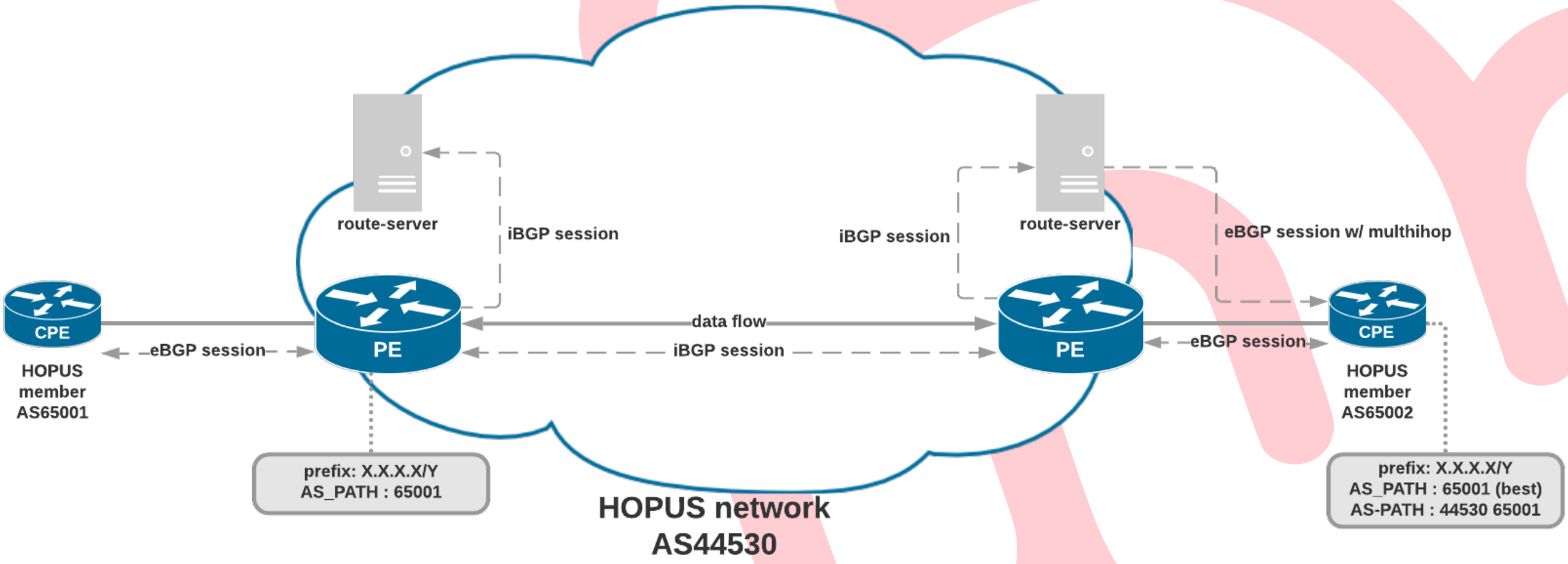
Routed IP network

Each member announce (all) his routes to all members

Each member can, on demand, bypass HOPUS ASN 44530 allowing them to shorten their path to destination (use of route-servers)

Wherever is your access node on HOPUS, you access all connected networks

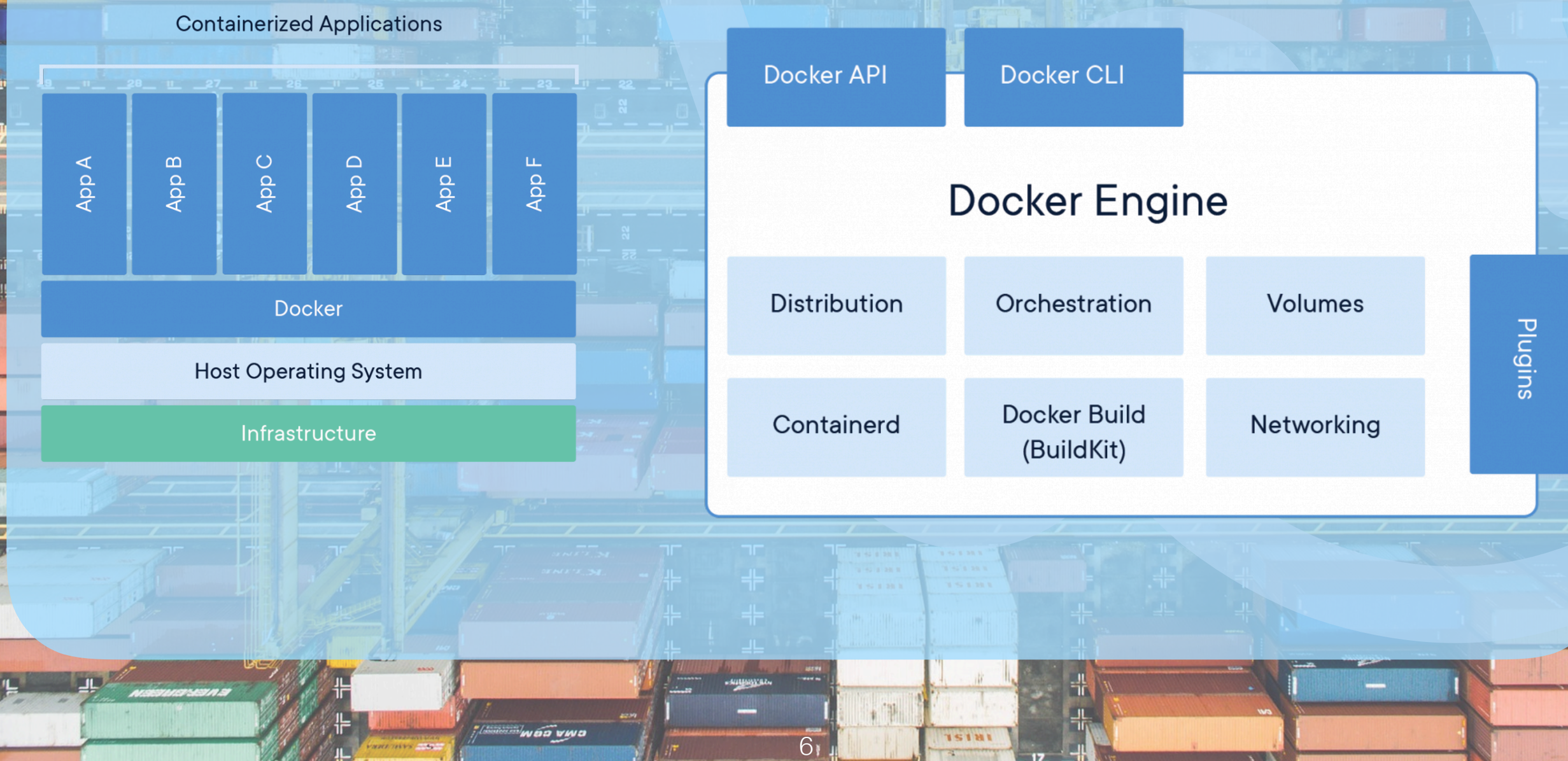
# How **HOP*US*** route-servers work?




- BGP session with the HOPUS routeur (PE) is mandatory
- BGP session with HOPUS route-server is optional
- The route-server does not learn routes from the eBGP sessions with the HOPUS members
- The route-server announce all the HOPUS routes
- BGP is not reciprocal, therefore each HOPUS member should establish a peering session with their local route-server if he does not want to see AS44530 (HOPUS ASN) in the AS_PATH

# How Docker works?

## 30 000ft overview



# Run the docker image

- Pull the image and run it right here, right now!
- https://hub.docker.com/r/afenioux/bird
- Docker is supported on Arista EOS from 4.20.1F and onwards

```
bash sudo su
service docker start
  Starting docker (via systemctl):       [  OK  ]

docker run -d -p 179:179 -v /mnt/flash/docker/bird:/etc/bird:rw --name rs \
--memory 512m --memory-swap 512m --cpus 1.1 afenioux/bird
  Unable to find image 'afenioux/bird:latest' locally
  latest: Pulling from afenioux/bird
  176ce20f5fa13658bc8b8068d599ee3044a2e3b0f0959603bdc2cf4b5baa8349

docker ps
  CONTAINER ID    IMAGE            COMMAND                  CREATED           STATUS          PORTS          NAMES
  176ce20f5fa1    afenioux/bird    "/bin/sh -c 'bird -d'"   20 seconds ago    Up 19 second    179            rs

docker logs rs
  2019-07-12 15:27:37 <INFO> Started

docker exec -it rs birdc show protocols
BIRD 2.0.4 ready.
Name       Proto      Table      State  Since         Info
device1    Device     ---        up     2019-07-12 15:27:43
iBGP       BGP        ---        up     2019-07-12 15:27:56  Established
iBGP_v6    BGP        ---        up     2019-07-12 15:28:43  Established
```

# DONE!

- But wait a minute…

I IZ SERIUS ADMNIM
THIZ IZ SERIUS BIZNIS

# Build and deploy an image yourself

- Multi-stage image based on debian:stable : 132MB
  https://hub.docker.com/r/afenioux/bird/dockerfile

```
docker build -t registry.hopus.net/bird:2.0.4 --build-arg BIRDV=2.0.4 .
  Sending build context to Docker daemon  3.584kB
  Step 1/19 : FROM debian:stable AS compiler
        ---> 22f4c938cdc8
  ...
  Successfully built 3bf87da26f2c
  Successfully tagged registry.hopus.net/bird:2.0.4

docker images
      REPOSITORY                TAG                 IMAGE ID            CREATED              SIZE
      registry.hopus.net/bird   2.0.4               3bf87da26f2c        28 seconds ago       132MB

docker history registry.hopus.net/bird:2.0.4
  IMAGE               CREATED             CREATED BY                                      SIZE
  3bf87da26f2c        About a minute ago  /bin/sh -c #(nop)  CMD ["/bin/sh" "-c" "bird…   0B
  8183c946e908        About a minute ago  |1 BIRDV=2.0.4 /bin/sh -c apt-get install -y…   6.96MB
  df609b80e3a7        About a minute ago  |1 BIRDV=2.0.4 /bin/sh -c echo $TZ > /etc/ti…   1.46MB
  05ecdc4530b2        About a minute ago  /bin/sh -c #(nop)  ENV TZ=Europe/Paris          0B
  b1cbdcc1aa18        About a minute ago  |1 BIRDV=2.0.4 /bin/sh -c mkdir /etc/bird /r…   0B
  539cf16c5236        About a minute ago  /bin/sh -c #(nop) COPY multi:dc6b1122ade7326…   4.48MB
  dd51f8b7b6fd        2 months ago        /bin/sh -c #(nop)  ARG BIRDV=2.0.4              0B
  c3a3916906fa        2 months ago        /bin/sh -c apt-get update && apt-get install…   18.1MB
  22f4c938cdc8        3 months ago        /bin/sh -c #(nop)  CMD ["bash"]                 0B
  <missing>           3 months ago        /bin/sh -c #(nop) ADD file:d891105338b1a0ab1…   101MB

docker login registry.hopus.net
docker push registry.hopus.net/bird:2.0.4
  2.0.4: digest: sha256:001b1b05a66e815cc90f369fa43647bff5b9a450f09da1ba10c4020413e4bf72 size: 1580
```

# Create a registry (and enable IPv6…)

- On the registry VM (using an LXC is a bad idea and is dangerous)

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
echo "deb https://download.docker.com/linux/debian stretch stable" > /etc/apt/sources.list.d/docker.list
apt-get update
apt-get install docker-ce docker-ce-cli containerd.io

docker run --entrypoint htpasswd registry:2 -Bbn the_user the_password >> /etc/docker/auth/htpasswd

docker run -d -p 80 --restart=always --name registry -v /etc/docker/auth:/auth \
  -e REGISTRY_HTTP_ADDR=0.0.0.0:80 -e “REGISTRY_AUTH=htpasswd" \
  -e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
  -e REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd \
  registry:2
```

- On the client (Docker host)

```
/etc/docker/daemon.json
  {
     "insecure-registries":["registry.hopus.net"],
     "ipv6": true,
     "fixed-cidr-v6": "2001:db8:1::/64"
  }

docker login registry.hopus.net
```

# What if you need to put a loopback IP into the container?

- This tricks is a courtesy from ENIX,
  btw they do excellent trainings :

  https://intro-2019-04.container.training

```
$CONTAINTER_ID=$(docker run -d -p 179 -v /mnt/flash/docker/bird:/etc/bird registry.hopus.net/bird)

NETNS_ID=$(docker inspect --format '{{.NetworkSettings.SandboxKey}}' $CONTAINTER_ID \
| awk -F '/' '{print $NF'})


echo "Adding loopback IP to container... "
ln -s /var/run/docker/netns/$NETNS_ID /var/run/netns

ip netns exec $NETNS_ID ip    address add $IP_RS  dev eth0
ip netns exec $NETNS_ID ip -6 address add $IP6_RS dev eth0
ip    route add $IP_RS  via 172.17.0.2
ip -6 route add $IP6_RS via 2001:db8:1::2
```

# Real case scenario

- Port 179 is already used by Rib on EOS

```
lab-arista# bash sudo netstat -lntup | grep :179
tcp        0      0 0.0.0.0:179             0.0.0.0:*               LISTEN      2512/Rib
tcp6       0      0 :::179                  :::*                    LISTEN      2512/Rib
```

- Instead of publishing the port (docker run **-p 179**) we will use iptable to do Destination NAT (as Docker does anyway) and set the IP of the container manually

```
interface loopback 1
  description IP_RS for Docker
  ip   address ${IP_RS}/32
  ipv6 address ${IP6_RS}/128

docker network create --subnet=172.16.0.0/24 --ipv6 --subnet=fd00:172:16::/64 hopusnet

docker run -d -v /mnt/flash/docker/bird:/etc/bird --name rs --hostname rs --memory 512m \
--cpus 1.1 --net hopusnet --ip 172.16.0.2 --ip6 fd00:172:16::2 registry.hopus.net/bird:$BIRD_VER

iptables  -t nat -I PREROUTING --dst ${IP_RS}/32   -p tcp --dport 179 -j DNAT --to 172.16.0.2
ip6tables -t nat -I PREROUTING --dst ${IP6_RS}/128 -p tcp --dport 179 -j DNAT --to fd00:172:16::2
```

- The Rib BGP daemon from EOS is local, so it can’t reach $IP_RS, but you can locally peer with the IP 172.16.0.2 even if port 179 isn’t published (same for IPv6).

# Wanna SDN?

Scheduling an event-handler is king of SDN, right?

```
conf session rs

schedule refresh-rs-config at 11:00:00 interval 1440 timeout 60 max-
log-files 7 command bash /mnt/flash/docker/refresh-rs-config.sh

event-handler docker-start
   trigger on-boot
   action bash /mnt/flash/docker/docker-start.sh
   delay 480
   asynchronous
   timeout 120
exit

commit
```

# Remove df warning at login

Apply this handmade fix at your own risk!

Keep a terminal logged in, and try to log-in in another terminal.

Failing to do so would result in no way to connect to the router and you would have to reboot it.


```
Arnauds-MBP:~ afenioux$ ssh lab-arista
 _      _    _       _  _   _   __     _   _   __                  _
 | |    / \  |  \    | || |/ _ \| _ \ | | | |/ __|    _ _   __   | |_
 | |_  | - | | <     | __ | (_) |  _/ | |_| |\__ \   | -  \/ -_)|  _|
 |_ _| |_|_| | _/    |_||_|\___/|_|    \___/ |___/ (_)_||_|\___| \__|

df: '/var/lib/docker/containers/
46bd000b263eec3919ef1852b432ed74722dbbb7bf2b7a30ee3689c9a1317148/shm':
Permission denied


lab-arista# bash sudo sed s/stdout=TacUtils.CAPTURE/
stdout=TacUtils.CAPTURE,\ stderr=TacUtils.DISCARD/ -i /usr/lib/python2.7/
site-packages/RunCli.py
```

# Sources

- ENIX training :
  https://intro-2019-04.container.training

- Docker basics :
  https://docker-curriculum.com/

- Docker on Arista :
  https://eos.arista.com/docker-containers-on-arista-eos/

- Docker networking :
  https://docs.docker.com/network/network-tutorial-macvlan/
  http://hicu.be/docker-networking-macvlan-bridge-mode-configuration
  https://docs.docker.com/v17.09/engine/userguide/networking/default_network/ipv6/

- Create a registry :
  https://docs.docker.com/registry/deploying/

  https://docs.docker.com/install/linux/docker-ce/debian/
  https://docs.docker.com/registry/insecure/
  https://docs.docker.com/v17.12/edge/engine/reference/commandline/dockerd/#insecure-registries

- Notes for transition from BIRD 1.6 to BIRD 2.0
  https://gitlab.labs.nic.cz/labs/bird/wikis/transition-notes-to-bird-2


MEET NETWORKS THAT MATTER