Docker LibNetwork Plugins

Explorer’s Tale
Why am I here?

- I read a code …
- I re-read the code …
- I realized that the code is in GO!
- I re-re-read the code …
- Finally, I fixed the code …
- Now, I can tell a story about it!
What are we going to do?

- Brainstorm
- Ask ourselves questions
- Reflect on our own experiences
What is a Driver?

“In computing, a device driver is a computer program that operates or controls a particular type of device that is attached to a computer.”

Wikipedia …
What is a Plugin?

“Plugins are ways to extend and add to the functionality that already exists in Wordpress Docker.”

Wordpress Website...
CNI Drivers Plugins

- dhcp
- host-local
- bridge
- host-device
- ipvlan
- loopback
- macvlan
- ptp
- vlan

containernetworking / cni
Branch: master

plugins / plugins / ipam /
Branch: master

plugins / plugins / main /
Container Networking Models

CNM  CNI

Container Network Model

- Docker is the only container implementation
- Drinks with Docker Sales Team next Tuesday?
- Are you a “hands-off manager”?
- You don’t care about Enterprise vs. Community licensing?

Container Network Interface

- Multiple container runtimes
- You have friends at CoreOS
- You have relatives at Mesosphere
- You worked at Google
- Are you a “micro-manager” or a LOTR fan?

It is a “model” when it is yours, and it is an “interface” when it is someone else’s
CNI

Container Networking Interface
Key Facts about CNI

- It is a specification. However, it comes with tools and plugins (i.e. drivers)
- Each driver is a binary
- Plugin is synonymous to a driver
- Network definitions are stored in JSON files
- Network definitions are passed to the drivers through standard input, `stdin`
- Drivers learn about arguments, e.g. namespace and container id, via environment variables (or configuration file)
- Drivers create container network namespaces and connect them to host networking
Container Network Interface

Orchestrator
- K8s Master
- Nomad Server
- Mesos Master

Orchestration Agent

Host
Docker Engine

Host
Docker

Host
Docker

Host
Docker

Host
Docker
Container Network Interface

Deploy [x] with IP 10.10.10.10 on “mynet”

$ docker run ... --net=none ... --ip 10.10.10.10
Container Network Interface

Questions:

• How does Docker know about “mynet”?
• What is “mynet”?
• Is it “10.10.10.10/24” or “10.10.10.10/16”?
• Who creates it?
• Why the Master asks for 10.10.10.10?
$ docker run -d -t --net=none  
  --name=delta1  centos /bin/bash  

$ docker inspect  
  -f '{{ .NetworkSettings.SandboxKey }}'  

$ cat /etc/cni/net.d/10-mynet.conf |  
  CNI_COMMAND=ADD \  
  CNI_CONTAINERID=<Container ID> \  
  CNI_NETNS=<Sandbox Key>  
  CNI_IFNAME=eth0 \  
  CNI_PATH=/usr/local/sbin/cni/ /usr/local/sbin/cni/bridge
Container Network Interface

• No need to synchronize network information across Docker-enabled hosts
• “Orchestrator” is in charge
• Orchestrator performs IPAM/DHCP-like functions, i.e. cleanup/release
• Workers use CNI network configuration files and binaries
CNM
Container Networking Model
Container Network Model

- Docker Engine
- Host
- Local KV Store: libkv
- Key-Value Store:
  - Etcd
  - Consul
  - Zookeeper

Diagram shows Docker Engine connected to Hosts with Local KV Store connected to Docker.
Key-Value Store & Docker Engine

• Must sync network information across Docker-enabled hosts via Key-Value Store
• No “Orchestrator”
• No “Orchestration Agent”
• Must perform IPAM/DHCP-like functions
Container Network Model

Docker Container Runtime

Docker LibNetwork Drivers

Remote Driver

Plugins

NetworkDriver

IpamDriver
Unix Domain Sockets

$ curl --unix-socket /var/run/docker.sock -H "Content-Type: application/json" \\
-d '{"Image": "alpine", "Cmd": ["echo", "hello world"]}' \\
-X POST http://v1.24/containers/create \\
{"Id":"1c6594faf5","Warnings":null}

$ curl --unix-socket /var/run/docker.sock -X POST http://v1.24/containers/1c6594faf5/start

$ curl --unix-socket /var/run/docker.sock -X POST http://v1.24/containers/1c6594faf5/wait \\
{"StatusCode":0}

$ curl --unix-socket /var/run/docker.sock "http://v1.24/containers/1c6594faf5/logs?stdout=1"
Unix Domain Sockets & Docker Engine

echo -e "GET /images/json HTTP/1.0\r\n" | nc -U /var/run/docker.sock

echo -e "GET /version HTTP/1.0\r\n" | nc -U /var/run/docker.sock

echo -e "GET /plugins HTTP/1.0\r\n" | nc -U /var/run/docker.sock
Driver vs. Plugin

Plugin is the extension of LibNetwork “remote” Driver

Diagram:
- Virtual Switch
- Network Driver Plugin
- IPAM Plugin
- Docker Engine
- Management Plane
- Key-Value Store (Etcd, Consul, Zookeeper)
<table>
<thead>
<tr>
<th>NETWORK ID</th>
<th>NAME</th>
<th>DRIVER</th>
<th>SCOPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>52c55293741f</td>
<td>bridge</td>
<td>bridge</td>
<td>local</td>
</tr>
<tr>
<td>9f9c1211dc9e</td>
<td>host</td>
<td>host</td>
<td>local</td>
</tr>
<tr>
<td>d620dd8aa2be</td>
<td>none</td>
<td>null</td>
<td>local</td>
</tr>
</tbody>
</table>
$ docker network create -d overlay --ipam-driver ipamx --subnet 10.4.4.0/24 mynet

Error response from daemon: datastore for scope "global" is not initialized
Docker LibNetwork Remote IPAM Plugin

• It is a **web server** listening on Unix Domain Socket

• The socket location: `/run/docker/plugins/ipamx.sock`

• Plugin type: **IpamDriver**
How IPAM Driver works?

$ docker network create -d bridge --ipam-driver ipamx --subnet 10.4.4.0/24 mynet
received request to activate the plugin
method: POST
path: /Plugin.Activate
header: user-agent => Go-http-client/1.1
header: content-length => 0
header: accept => application/vnd.docker.plugins.v1.2+json
received request to provide capabilities
the advertised capabilities are:
  RequiresMACAddress=false,
  RequiresRequestReplay=false
method: POST
path: /IpamDriver.GetCapabilities
header: user-agent => Go-http-client/1.1
header: content-length => 0
header: accept => application/vnd.docker.plugins.v1.2+json
received request to provide default IP address spaces
the default IP address spaces are:
  global=ipamx-global,
  local=ipamx-local
method: POST
path: /IpamDriver.GetDefaultAddressSpaces
header: user-agent => Go-http-client/1.1
header: content-length => 0
header: accept => application/vnd.docker.plugins.v1.2+json
received request to reserve an IP address range
the '10.4.4.0/24' pool is in local address space
the IPv4 address range is 10.4.4.0/24 in the 'ipamx-local'
address space the pool identifier is 'ipamx-local/10.4.4.0/24'
method: POST
path: /IpamDriver.RequestPool
header: content-length => 89
header: accept => application/vnd.docker.plugins.v1.2+json
header: user-agent => Go-http-client/1.1
body: {
    "AddressSpace":"ipamx-local",
    "Pool":"10.4.4.0/24",
    "SubPool":"
    "Options":{},
    "V6":false
}
received request for an IP address
this is default gateway discovery request
(com.docker.network.gateway) for PoolID
'ipamx-local/10.4.4.0/24'
method: POST
path: /IpamDriver.RequestAddress
header: user-agent => Go-http-client/1.1
header: content-length => 112
header: accept => application/vnd.docker.plugins.v1.2+json
body: {
    "PoolID": "ipamx-local/10.4.4.0/24",
    "Address": "",
    "Options": {
        "RequestAddressType": "com.docker.network.gateway"
    }
}
How IPAM Driver works?

$ docker run -d -t --net=mynet --name=delta1 --ip 10.4.4.21 centos /bin/bash
received request to release an IP address
method: POST
path: /IpamDriver.ReleaseAddress
header: user-agent => Go-http-client/1.1
header: content-length => 59
header: accept => application/vnd.docker.plugins.v1.2+json
body: {
    "PoolID":"ipamx-local/10.4.4.0/24",
    "Address":"10.4.4.21"
}
How IPAM Driver works?

$ docker stop delta1
received request to release an IP address
method: POST
path: /IpamDriver.ReleaseAddress
header: user-agent => Go-http-client/1.1
header: content-length => 59
header: accept => application/vnd.docker.plugins.v1.2+json
body: {
   "PoolID":"ipamx-local/10.4.4.0/24",
   "Address":"10.4.4.21"
}
How IPAM Driver works?

$ docker network rm mynet
received request to release an IP address
method: POST
path: /IpamDriver.ReleaseAddress
header: user-agent => Go-http-client/1.1
header: content-length => 58
header: accept => application/vnd.docker.plugins.v1.2+json
body: {
    "PoolID":"ipamx-local/10.4.4.0/24",
    "Address":"10.4.4.1"
}
received request to release an IP address pool
the IP address pool identifier is 'ipamx-local/10.4.4.0/24'
method: POST
path: /IpamDriver.ReleasePool
header: user-agent => Go-http-client/1.1
header: content-length => 37
header: accept => application/vnd.docker.plugins.v1.2+json
body: {
    "PoolID": "ipamx-local/10.4.4.0/24"
}
Common Issues with IPAM in CN(MI)

- Static IP addressing
- Scaling
- Lack of Endpoints
  - /IpamDriver.Metrics
  - /IpamDriver.Version
  - /IpamDriver.Shutdown
  - /IpamDriver.Networks
  - /IpamDriver.Leases
  - /IpamDriver.Configuration
Writing Drivers/Plugins
Thank you! Engage with Community

- Github & Slack
  - docker/libnetwork
  - containernetworking/cni
- Meetups:
  - NANOG
  - NYNOG
- Ping me github.com/greenpau