github.com/opencontainers/specs
Image Format

*Application Container Image* (.aci)
tarball of rootfs + manifest
uniquely identified by ImageID (hash)
Image Discovery

Resolves app name → artefact (.aci)
example.com/http-server
coreos.com/etcd

DNS + HTTPS + HTML meta tags
Crypto Verification

Take an ACI, public key and signature. Verify()
Pods

grouping of multiple applications
(templated or deterministic)

shared execution context
(namespaces, volumes)
Executor

runtime environment
isolators, networking, lifecycle
metadata service
OCI - Open Containers Initiative

- Announced June 2015 (as OCP)
- Lightweight, open governance project
- Linux Foundation
- Container runtime format
  - configuration on disk, execution environment
- Runtime implementation (runc)
appc vs OCI

appc
- image format
- runtime environment
- pods
- image discovery

OCI
- runtime format
- runtime environment
appc vs OCI

appc runtime
- environment variables
- Linux device files
- hooks
- etc...
- multiple apps

OCI runtime
- environment variables
- Linux device files
- hooks
- etc...
- single app (process)
Container Network Interface

github.com/appc/cni

Brandon Philips
@brandonphilips
Application containers are awesome

- Application containers provide
  - isolation
  - packaging

- Networking isolation
  - its own port space
  - its own IP
Network Namespace

- Can every container have a "real" IP?
- How should network be virtualized?
- Is network virtualization part of "container runtime"? e.g. rkt, docker, etc
$ sudo unshare -n /bin/bash

$ ip addr
1: lo: <LOOPBACK> mtu 65536 ...
   link/loopback 00:00:00:00:00:00 brd ...
$ ip link set lo up

$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 ...
 link/loopback 00:00:00:00:00:00 brd ...
 inet 127.0.0.1/8 scope host lo
   valid_lft forever preferred_lft forever
 inet6 ::1/128 scope host
   valid_lft forever preferred_lft forever

New net ns
New net ns

$ ping 8.8.8.8
connect: Network is unreachable

$ ip route show
$
$
veth

10.0.1.4  10.0.1.5/31

10.0.1.6  10.0.1.7/31
veth

10.0.1.1/24

10.0.1.5/24
10.0.1.7/24
Virtualizing the NIC and Network

- veth pair (plus linux-bridge)
- macvlan
- ipvlan
- OVS
- vlan
- vxlan
IP Address Management

- Host
- Cluster
- Global
Which one?

No right answer!
Need pluggable network strategy
Container Runtime (e.g. rkt)

- veth
- macvlan
- ipvlan
- OVS
Container Runtime (e.g. rkt)

- veth
- macvlan
- ipvlan
- OVS
Container Runtime (e.g. rkt)

Container Networking Interface (CNI)

veth  macvlan  ipvlan  OVS
CNI

- Container can join multiple networks
- Network described by JSON config
- Plugin supports two commands
  - Add container to the network
  - Remove container from the network
User configures a network

```bash
$ cat /etc/rkt/net.d/10-mynet.conf
{
    "name": "mynet",
    "type": "bridge",
    "ipam": {
        "type": "host-local",
        "subnet": "10.10.0.0/16"
    }
}
```
CNI: Step 1

Container runtime creates network namespace and gives it a named handle

$ cd /run
$ touch myns
$ unshare -n mount --bind /proc/self/ns/ns/net myns
CNI: Step 2

Container runtime invokes the CNI plugin

$ export CNI_COMMAND=ADD
$ export CNI_NETNS=/run/myns
$ export CNI_CONTAINERID=5248e9f8-3c91-11e5-...
$ export CNI_IFNAME=eth0

$ $CNI_PATH/bridge </etc/rkt/net.d/10-mynet.conf
CNI: Step 3

Inside the bridge plugin (1):

$ brctl addbr mynet
$ ip link add veth123 type veth peer name $CNI_IFNAME
$ brctl addif mynet veth123
$ ip link set $CNI_IFNAME netns $CNI_IFNAME
$ ip link set veth123 up
CNI: Step 3

Inside the bridge plugin (2):

```
$ IPAM_PLUGIN=host-local  # from network conf
$ echo $IPAM_PLUGIN
{
  "ip4": {
    "ip": "10.10.5.9/16",
    "gateway": "10.10.0.1"
  }
}
```
CNI: Step 3

Inside the bridge plugin (3):

# switch to container namespace

$ ip addr add 10.0.5.9/16 dev $CNI_IFNAME

# Finally, print IPAM result JSON to stdout
<table>
<thead>
<tr>
<th>Top level</th>
<th>IPAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ptp</td>
<td>host-local</td>
</tr>
<tr>
<td>bridge</td>
<td>dhcp</td>
</tr>
<tr>
<td>macvlan</td>
<td></td>
</tr>
<tr>
<td>ipvlan</td>
<td></td>
</tr>
</tbody>
</table>
Questions

github.com/appc/cni