Fault tolerant frameworks: Making use of CNI without Docker

Aaron Wood
Principal Software Engineer
Verizon Labs

Tim Hansen
Senior Software Engineer
Verizon Labs

September 15, 2017

verizon

Making the framework: V0 or V1?

V0

Requires C++ or bindings

- Java/Scala and Python bindings provided by Mesos
- Not the recommended path for development

V1

Streaming HTTP

- Allows for compression

No bindings required

- Use any language you want

JSON or Protobuf payloads between scheduler and Mesos

Obvious choice was V1 with compressed Protobuf payloads

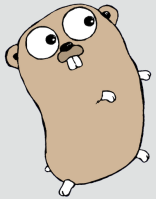
verizon

Making the framework: Language

Go as scheduler, executor and accompanying SDK

Why Go?

- Increased developer speed
- Fast and light on memory
- Excellent concurrency primitives
- Single statically-linked binary for the scheduler and executor



Language of choice for Verizon Labs greenfield projects

Note: Gopher image created by Renee French taken from https://golang.org/doc/gopher/

Making the framework: SDK

All common functionality moved to a separate SDK

Reduced boilerplate

Required code that any scheduler can handle

Task lifecycle management

Resource (Offer) lifecycle management

RecordIO event decoder

Streaming HTTP

- Scheduler and executor calls
- Client with leader detection

Persistence (storage backend)

Scheduler, executor and common protobufs

verizon

Containers are containers.

There’s no real need for an extra daemon and client when Mesos can containerize tasks.

verizon

Isolation: Containers without Docker

Pros

Docker + dependencies don’t need to be installed, managed and patched/upgraded across the cluster

No more blocking if the Docker daemon gets stuck

Reduced attack surface

No need to manage “secrets” (aka encoded JSON) or setup an external credentials store

Broader support for container image specifications

Isolation: Containers without Docker

Cons

User namespaces currently not supported

Seccomp currently not supported

Showstopper bugs if using Mesos < 1.2.x

Image backends

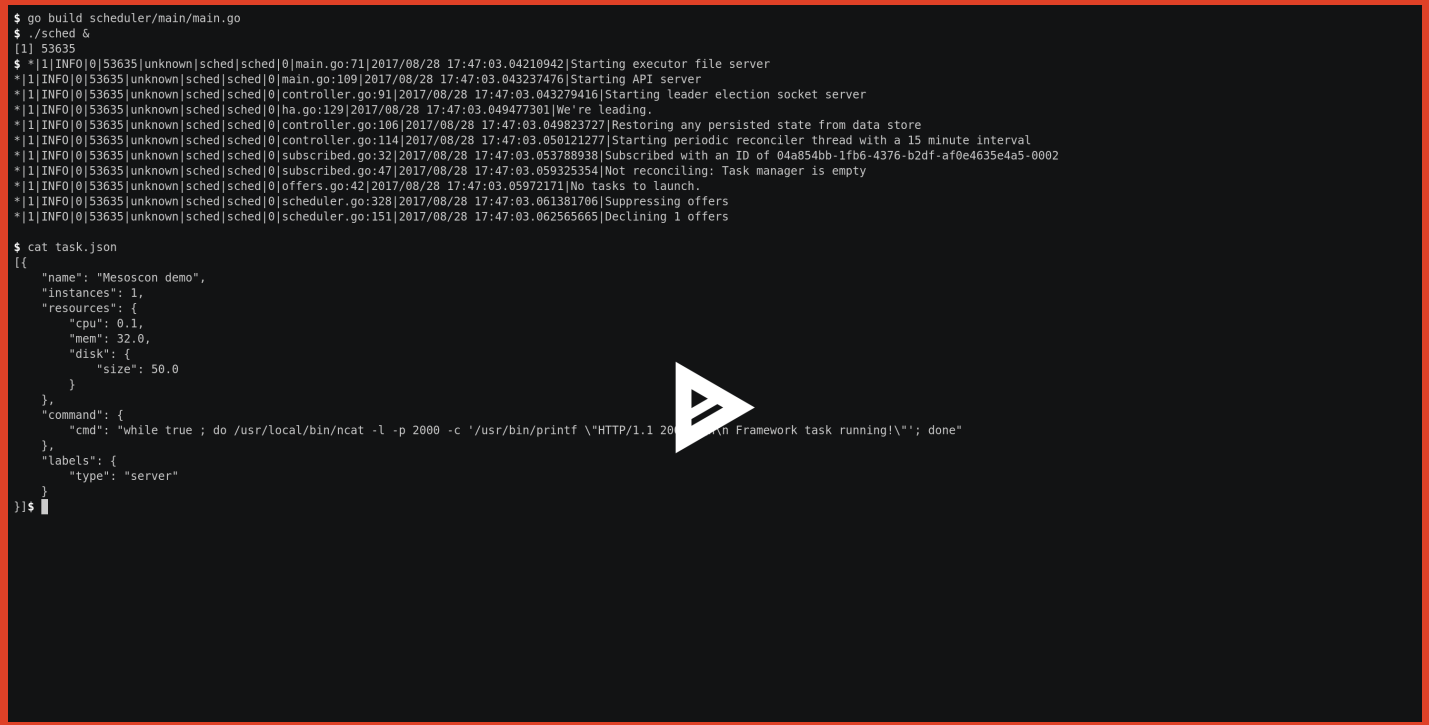
- MESOS-6875
- MESOS-5028
- MESOS-6327
- MESOS-7280

Whiteout files

- MESOS-6002
- MESOS-6360

verizon

DEMO



Container network interface: Overview

Standardized way of creating networks for containers.

CNI allows us to create a network at runtime for a container on any host.

Supported Types of Plugins

Network	IPAM	Other
bridge	DHCP	flannel
ipvlan	**host-local**	tuning
loopback		portmap
macvlan		
ptp		
vlan		

Container network interface: Mesos

How does it interface with Mesos?

CNI configurations are placed on each node; the default is /etc/cni/net.d/<config>.conf

Configuration describes the CNI version, name, network type and other options according to the network type

The NetworkInfo protobuf attached to the task has the name set to CNI network to which it wishes to attach

Once the task is launched onto a node in the cluster, Mesos sends the information to CNI

CNI looks at its configuration and sees if network has a configuration

Network interface(s) are then created in the namespace for the container

Container network interface: Benefits

Automates network configuration and management for containers

Standardized with no vendor lock-in

Can support multiple networks and plug-ins per container

Isolation from other services for multi-tenant environments

End user visibility; looks like a private L2/L3 network

Implementation is decoupled from the interface to allow flexibility, i.e., change to an overlay mechanism like VXLAN or NVGRE without the end user noticing

IPv4 IPAM address management

Container network interface: Example

```
 21:59:33 node-50-11 mesos-agent[8738]: I0828 21:59:33.869058  8757 slave.cpp:4264] Handling status update TASK_FINISHED (UUID: adb67f3f-1bc1-4dfd-9bf3-542e85661d92) for task test-4 of framework a2b793ec-72
19-e5fddf1383ff-0003 from executor(1)@11.50.11.2:33577
 21:59:26 node-50-11 mesos-agent[8738]: I0828 21:59:26.396481  8765 slave.cpp:5715] Current disk usage 7.18%. Max allowed age: 5.797728782231215days
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.775009  8755 status_update_manager.cpp:832] Checkpointing ACK for status update TASK_RUNNING (UUID: fed48bc7-b84a-43c0-be1f-97852d3aa8a0) for task test
ework a2b793ec-728c-4a4e-bd19-e5fddf1383ff-0003
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.774948  8755 status_update_manager.cpp:395] Received status update acknowledgement (UUID: fed48bc7-b84a-43c0-be1f-97852d3aa8a0) for task test-4 of fram
93ec-728c-4a4e-bd19-e5fddf1383ff-0003
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.771245  8772 slave.cpp:4614] Sending acknowledgement for status update TASK_RUNNING (UUID: fed48bc7-b84a-43c0-be1f-97852d3aa8a0) for task test-4 of fra
793ec-728c-4a4e-bd19-e5fddf1383ff-0003 to executor(1)@11.50.11.2:33577
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.771157  8772 slave.cpp:4704] Forwarding the update TASK_RUNNING (UUID: fed48bc7-b84a-43c0-be1f-97852d3aa8a0) for task test-4 of framework a2b793ec-728c
e5fddf1383ff-0003 to master@172.17.50.47:5050
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.771013  8762 status_update_manager.cpp:832] Checkpointing UPDATE for status update TASK_RUNNING (UUID: fed48bc7-b84a-43c0-be1f-97852d3aa8a0) for task t
ramework a2b793ec-728c-4a4e-bd19-e5fddf1383ff-0003
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.770807  8762 status_update_manager.cpp:323] Received status update TASK_RUNNING (UUID: fed48bc7-b84a-43c0-be1f-97852d3aa8a0) for task test-4 of framewo
c-728c-4a4e-bd19-e5fddf1383ff-0003
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.770056  8751 slave.cpp:4264] Handling status update TASK_RUNNING (UUID: fed48bc7-b84a-43c0-be1f-97852d3aa8a0) for task test-4 of framework a2b793ec-728
9-e5fddf1383ff-0003 from executor(1)@11.50.11.2:33577
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.765527  8751 slave.cpp:2529] Sending queued task 'test-4' to executor 'test-4' of framework a2b793ec-728c-4a4e-bd19-e5fddf1383ff-0003 at executor(1)@11
3577
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.765141  8758 cpu.cpp:101] Updated 'cpu.shares' to 204 (cpus 0.2) for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.764077  8758 memory.cpp:199] Updated 'memory.soft_limit_in_bytes' to 64MB for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.763311  8771 disk.cpp:208] Updating the disk resources for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8 to cpus(*)(allocated: *):0.2; mem(*)(allocate
isk(*)(allocated: *):1024
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.762318  8739 slave.cpp:3790] Got registration for executor 'test-4' of framework a2b793ec-728c-4a4e-bd19-e5fddf1383ff-0003 from executor(1)@11.50.11.2:
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.563856  8768 cni.cpp:1010] Unable to find DNS nameservers for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8, using host '/etc/resolv.conf'
 21:58:43 node-50-11 mesos-agent[8738]: I0828 21:58:43.563606  8759 cni.cpp:1301] Got assigned IPv4 address '11.50.11.2/24' from CNI network 'dev-net' for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8
 21:58:42 node-50-11 mesos-agent[8738]: I0828 21:58:42.558835  8748 cni.cpp:1301] Got assigned IPv4 address '10.50.11.12/24' from CNI network 'data-net' for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.397740  8765 cni.cpp:888] Bind mounted '/proc/9783/ns/net' to '/run/mesos/isolators/network/cni/f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8/ns' for container
ba0-41ca-8fa8-ea908ae3d4e8
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.396611  8750 containerizer.cpp:1630] Checkpointing container's forked pid 9783 to '/var/lib/mesos/meta/slaves/a2b793ec-728c-4a4e-bd19-e5fddf1383ff-S0/f
a2b793ec-728c-4a4e-bd19-e5fddf1383ff-0003/executors/test-4/runs/f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8/pids/forked.pid'
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.394263  8753 systemd.cpp:96] Assigned child process '9783' to 'mesos_executors.slice'
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.390377  8753 linux_launcher.cpp:429] Launching container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8 and cloning with namespaces CLONE_NEWNS | CLONE_NEWUTS |
D | CLONE_NEWNET
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.387537  8766 cpu.cpp:101] Updated 'cpu.shares' to 204 (cpus 0.2) for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.386939  8766 memory.cpp:228] Updated 'memory.limit_in_bytes' to 64MB for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.385855  8766 memory.cpp:199] Updated 'memory.soft_limit_in_bytes' to 64MB for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.384769  8766 memory.cpp:590] Started listening on 'critical' memory pressure events for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.384028  8766 memory.cpp:590] Started listening on 'medium' memory pressure events for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.383303  8766 memory.cpp:590] Started listening on 'low' memory pressure events for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.382562  8766 memory.cpp:479] Started listening for OOM events for container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.378604  8744 slave.cpp:2316] Queued task 'test-4' for executor 'test-4' of framework a2b793ec-728c-4a4e-bd19-e5fddf1383ff-0003
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.378422  8765 containerizer.cpp:1001] Starting container f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8 for executor 'test-4' of framework a2b793ec-728c-4a4e-bd19
3ff-0003
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.378113  8744 slave.cpp:7038] Launching executor 'test-4' of framework a2b793ec-728c-4a4e-bd19-e5fddf1383ff-0003 with resources cpus(*)(allocated: *):0.
llocated: *):32 in work directory '/var/lib/mesos/slaves/a2b793ec-728c-4a4e-bd19-e5fddf1383ff-S0/frameworks/a2b793ec-728c-4a4e-bd19-e5fddf1383ff-0003/executors/test-4/runs/f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.377841  8744 paths.cpp:573] Trying to chown '/var/lib/mesos/slaves/a2b793ec-728c-4a4e-bd19-e5fddf1383ff-S0/frameworks/a2b793ec-728c-4a4e-bd19-e5fddf138
xecutors/test-4/runs/f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8' to user 'root'
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.377154  8744 slave.cpp:2087] Launching task 'test-4' for framework a2b793ec-728c-4a4e-bd19-e5fddf1383ff-0003
 21:58:40 node-50-11 mesos-agent[8738]: I0828 21:58:40.376957  8744 slave.cpp:1900] Authorizing task 'test-4' for framework a2b793ec-728c-4a4e-bd19-e5fddf1383ff-0003
```

Container network interface: Example

```
core@node-50-11 ~ $ export PID=9783
core@node-50-11 ~ $ sudo nsenter --target $PID --mount --uts --ipc --net --pid
Update Strategy: No Reboots
Failed Units: 1
  update-engine-stub.service
f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8 / # ip addr sho
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    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
3: eth1@if22: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 0a:58:0b:32:0b:02 brd ff:ff:ff:ff:ff:ff
    inet 11.50.11.2/24 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::3c4b:9aff:fe3c:2bef/64 scope link
       valid_lft forever preferred_lft forever
5: eth0@if23: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 0a:58:0a:32:0b:0c brd ff:ff:ff:ff:ff:ff
    inet 10.50.11.12/24 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::fc26:7bff:fee5:ada9/64 scope link
       valid_lft forever preferred_lft forever
f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8 / # ip route sho
default via 11.50.11.1 dev eth1
10.50.11.0/24 dev eth0  proto kernel  scope link  src 10.50.11.12
11.50.11.0/24 dev eth1  proto kernel  scope link  src 11.50.11.2
f6cc7a81-4ba0-41ca-8fa8-ea908ae3d4e8 / # logout
core@node-50-11 ~ $
```

verizon

Container network interface: Improvements

What's lacking?

IPv6 support

Support for dynamic traffic policy filtering

Support for dynamic updates to existing network configurations

verizon

Questions?

verizon

Thank you.

verizon