

Networking Run Book

Information required:

DC/OS:

- DC/OS version
 - `/opt/mesosphere/bin/dcos-shell printenv DCOS_VERSION`
- OS version
 - [rhel] `cat /etc/system-release`
 - [coreos] `cat /etc/os-release`
- JournalD logs and SystemD statuses
 - [all] ``mesos-master`, `mesos-agent`, and `mesos-dns``
 - [1.9, 1.10] ``navstar`, `spartan``
 - [1.11] ``dcos-net``
- Error log files for Erlang applications:
 - [1.9, 1.10]
 - `/opt/mesosphere/active/spartan/spartan/error.log*`
 - `/opt/mesosphere/active/spartan/spartan/log/`
 - `/opt/mesosphere/active/spartan/spartan/erl_crash.dump`
 - `/opt/mesosphere/active/navstar/navstar/error.log*`
 - `/opt/mesosphere/active/navstar/navstar/log/`
 - `/opt/mesosphere/active/navstar/navstar/erl_crash.dump`
 - [1.11]
 - `/opt/mesosphere/active/dcos-net/dcos-net/log/`
 - `/opt/mesosphere/active/dcos-net/dcos-net/erl_crash.dump`
- DC/OS configuration
 - `/opt/mesosphere/etc/user.config.yaml`
- Marathon App definition:
 - Especially to understand the network they are trying launch the container on, and if its a UCR vs a Docker container.

dcos-l4lb (minuteman)

From agent on which service instance is running, and the client from which the connectivity was initiated.

- Try to resolve l4lb fqdn:
 - `dig viplabel.yourframework.l4lb.thisdcos.directory @198.51.100.1`
- Get real servers from ipvsadm and try to connect to your applications directly
 - `docker run --rm --net=host --privileged mesosphere/net-toolbox:latest ipvsadm -L -n -t $(dig +short viplabel.yourframework.l4lb.thisdcos.directory):vipport`
- Lashup state:
 - `[1.10, 1.11] curl http://localhost:62080/v1/vips | tee vips.txt`
 - `[1.9] /opt/mesosphere/active/navstar/navstar/bin/navstar-env eval 'lashup_kv:value([minuteman, vips2]).' | tee vips.txt`
- IP_VS state
 - `docker run --rm --net=host --privileged mesosphere/net-toolbox:latest ipvsadm -L -n | tee ipvsadm.txt`
- IP_VS timeouts
 - `docker run --rm --net=host --privileged mesosphere/net-toolbox:latest ipvsadm -L --timeout | tee ipvsadm-timeout.txt`
- Conntrack table state
 - `cp /proc/net/ip_vs_conn ip_vs_conn.txt`
- Kernel configuration
 - `(sysctl net.ipv4.vs; sysctl net.ipv4.ip_local_port_range) | tee sysctl.txt`
- Kernel version
 - `uname -a | tee uname.txt`
- Iptables configuration
 - `iptables-save | tee iptables-save.txt`
- Minuteman routing table
 - `ip route show table local dev minuteman scope host | tee routes.txt`
- Lashup hyparview membership
 - `[1.9, 1.10] /opt/mesosphere/active/navstar/navstar/bin/navstar-env eval 'lashup_gm:gm().' | tee lashup-membership.txt`
 - `[1.11] /opt/mesosphere/bin/dcos-net-env eval 'lashup_gm:gm().' | tee lashup-membership.txt`
- Mesos agent state
 - `[1.9, 1.10] /opt/mesosphere/active/navstar/navstar/bin/navstar-env eval 'mesos_state_client:poll(mesos_state:ip(), 5051).' | tee navstar-mesos-agent.txt`
 - `[1.11] /opt/mesosphere/bin/dcos-net-env eval 'false = dcos_dns:is_master(), dcos_net_mesos:poll("/state").' | tee dcos-net-mesos-agent.txt`
- Mesos master state
 - `open $(dcos config show core.dcos_url)/mesos/state`

dcos-overlay (navstar)

From agent on which service instance is running, and the client from which the connectivity was initiated.

- How many masters and agents are present.
- Networking configuration on the agents.
 - `ifconfig -a | tee ifconfig.txt`
 - `ip link | tee ip-link.txt`
 - `ip addr | tee ip-addr.txt`
 - `ip route | tee ip-route.txt`
 - `iptables-save | tee iptables.txt`
- Mesos overlay information
 - [master]
`http://$(/opt/mesosphere/bin/detect_ip):5050/overlay-master/state`
 - [agent]
`http://$(/opt/mesosphere/bin/detect_ip):5051/overlay-agent/overlay`

dcos-dns (spartan)

From agent on which service instance is running, and the client from which the connectivity was initiated.

- **Resolv.conf**
 - `cat /etc/resolv.conf`
- **Connectivity to spartan, mesos-dns and upstream dns servers:**
 - `dig ready.spartan`
 - `dig ready.spartan @198.51.100.1`
 - `dig dcos.io @198.51.100.1`
 - `dig leader.mesos @198.51.100.1`
 - `source /opt/mesosphere/etc/dns_config; for R in $(echo $RESOLVERS | tr ',' '\n'); do echo "=== Upstream DNS server: $R ==="; dig dcos.io @$R; done`
- **All DC/OS DNS records:**
 - `[1.9] /opt/mesosphere/active/spartan/spartan/bin/spartan-env eval 'Zones = erldns_zone_cache:zone_names_and_versions(), lists:flatten([begin {ok, Zone} = erldns_zone_cache:get_zone_with_records(Z), element(6, Zone) end || {Z, _} <- Zones]).' | tee records.txt`
 - `[1.10] curl http://localhost:63053/v1/records | tee records.txt`
 - `[1.11] curl http://localhost:62080/v1/records | tee records.txt`
- **DC/OS Net mesos master state:**
 - `[1.9, 1.10] /opt/mesosphere/active/navstar/navstar/bin/navstar-env eval 'mesos_state_client:poll("https://leader.mesos:5050/state").' | tee navstar-mesos-state.txt`
 - `[1.11] /opt/mesosphere/bin/dcos-net-env eval 'true = dcos_dns:is_master(), dcos_net_mesos:poll("/state").' | tee dcos-net-mesos-state.txt`
- **Mesos master state**
 - `open $(dcos config show core.dcos_url)/mesos/state`
- **Lashup hyparview membership**
 - `[1.9, 1.10] /opt/mesosphere/active/navstar/navstar/bin/navstar-env eval 'lashup_gm:gm().' | tee lashup_gm.txt`
 - `[1.11] /opt/mesosphere/bin/dcos-net-env eval 'lashup_gm:gm().' | tee lashup_gm.txt`

Mesos-dns

- Connectivity to spartan, mesos-dns and upstream dns servers:
 - `dig ready.spartan`
 - `dig ready.spartan @198.51.100.1`
 - `dig leader.mesos @198.51.100.1`
- Resolve DNS record directly on leader node
 - `dig <yourapp>.<yourframework>.mesos @127.0.0.1 -p 61053`
- Mesos-dns configuration
 - `/opt/mesosphere/etc/mesos-dns.json`
- Get all mesos-dns DNS records
 - `curl http://localhost:8123/v1/enumerate | tee mesos-dns.json`
- Mesos master state
 - `open $(dcos config show core.dcos_url)/mesos/state`

Marathon-LB

- Marathon-LB Version
- Marathon-LB configuration
 - `dcos marathon app show /marathon-lb`
- HAPROXY_* labels from application definition
 - `dcos marathon app show /<yourapp> | jq '.labels | with_entries(select(.key | startswith("HAPROXY_")))'`
- haproxy.cfg
 - `curl`
`http://marathon-lb.marathon.agentip.dcos.thisdcos.directory:9090/_haproxy_getconfig`
- Logs: stderr and stdout from /marathon-lb application
 - `dcos task log --lines=10240 marathon-lb stdout`
 - `dcos task log --lines=10240 marathon-lb stderr`

Edge-LB

- Edge-LB version
 - `dcos edgelb version`
- Edge-LB configuration
 - `dcos marathon app show /dcos-edgelb/api`
 - `dcos marathon app show /dcos-edgelb/pools/<pool-name>`
- Pool configuration JSON
 - [`< v1.0.0`] `dcos edgelb pool config <pool-name>`
 - [`>= v1.0.0`] `dcos edgelb show <pool-name>`
- haproxy.cfg
 - `curl http://<agent-ip>:9090/_haproxy_getconfig`
 - [`< v1.0.0`] `dcos edgelb pool artifact <pool-name> haproxy.cfg`
 - [`>= v1.0.0`] `dcos edgelb lb-config <pool-name>`
- Logs: stderr and stdout from these applications / tasks:
 - APIServer: `/dcos-edgelb/api`
 - `dcos task log --lines=10240 dcos-edgelb_api stdout`
 - `dcos task log --lines=10240 dcos-edgelb_api stderr`
 - Pool Scheduler: `/dcos-edgelb/pools/<poolname>`
 - `dcos task log --lines=10240 dcos-edgelb_pools_<poolname> stdout`
 - `dcos task log --lines=10240 dcos-edgelb_pools_<poolname> stderr`
 - LB Task: `edgelb-pool-0-server ... edgelb-pool-N-server` tasks
 - `dcos task log --lines=10240 edgelb-pool-<N>-server stdout`
 - `dcos task log --lines=10240 edgelb-pool-<N>-server stderr`
- Mesos state:
 - `open $(dcos config show core.dcos_url)/mesos/state`