DC/OS Components, System Components, and Basic Troubleshooting

Component	Service Name	Layer	Dependencies	Subcomponent	Description	Impact to Environment	Where to begin
AdminRouter	`dcos-adminrouter.service`	DC/OS	dcos-master; leader.mesos:5050		Control plane proxy for DC/OS admin and components	Loss of the adminrouter component will prevent control plane access (management) to the cluster. Loss of adminrouter on the agents will prevent management via the UI/CLI to agents and will need to be recitified on the local agent itself. Loss of agent from management UI.	Check status of the service: `systemctl status dcos-adminrouter.service`; Check logs of adminrouter from boot to see if there are issues: `journalctl -u dcos-adminrouter -b` (looking for exit codes); restart admin router: `systemctl restart dcos-adminrouter`; Inspect logs for running adminrouter: `journalctl -flu dcos-adminrouter`
Exhibitor	`dcos-exhibitor.service`	DC/OS	Zookeeper		Zookeeper supervisor	Cluster master node communication and quorum voting is conducted via Zookeeper. Monitoring and watching of Zookeeper is accomplished via Exhibitor. Losing Exhibitor is tantamount to losing quality and reliability of the entire cluster.	Verify that Exhibitor is up and operational at: `http://<master-node-ip-addr>:8181/exhibitor`; on a master node, check the Exhibitor logs: `journalctl -flu dcos-exhibitor`; Check that all masters are in the cluster and serving and one is elected leader- from a master: `curl -fsSL http://localhost:8181/exhibitor/v1/cluster/status |jq .`
Navstar	`dcos-navstar.service`	DC/OS	Mesos master; inet_gethost; erlang port mapping daemon (epmd)		Networking overlay for DC/OS; manages IPv4 Layer 4 load balancing	Loss of Navstar will impact the internal networking of agents/masters and components within the DC/OS solution/cluster.	Ensure that filesystems on the node with Navstar issues is not full. Investigate the logs of navstar: `journalctl -flu dcos-navstar`
CockRoachDB	`dcos-cockroach.service`	DC/OS	-		Distributed SQL database with strongly consistent key-value store.	Losing or issues with cockroachdb will affect IAM (bouncer) and thusly will create cascade events within the cluster. Logging in and authentication are obvious side-effects.	`journalctl -flu dcos-cockroach`
Bouncer	`dcos-bouncer.service`	DC/OS	mesos.master		IAM access control to components and services within DC/OS. Can also be integrated with external services like LDAP	Authenication and control issues within the environment. Mesos and Marathon may also be integrated with bouncer and when bouncer is lost access to these services will not be possible. Requests and interaction with adminrouter will also be affected.	`journalctl -flu dcos-bouncer`
Marathon	`dcos-marathon.service`	DC/OS		Service Marathons	Marathon is the main container orchestration component (engine) for DC/OS. It handles management of long running tasks in the cluster. Marathon can also manage child-Marathon tasks in the environment, frequently called Service Marathons.	As a meta-framework in DC/OS, any issue adversly affecting Marathon could potentially affect all tasks (sometimes more than 1000s) running on the cluster. Without Marathon there can be substantial degradation in the cluster and loss of service.	Monitor Marathon from the DC/OS UI: click "Services" tab and select Marathon; inspect logs for Marathon from boot time: `journalctl -u dcos-marathon -b`
Master	`dcos-master.service`	DC/OS	Zookeeper	cluster agents, tasks, communication, overall cluster	DC/OS distributed systems kernel.	Losing masters in the DC/OS cluster is a critical event and can potentially impact the entire cluster and all tasks running on the cluster. Most production clusters have a minimum of 3 master nodes which can sustain the failure of one node.	View the Mesos Master interface directly at: http://<master-ip-addr>/mesos; gather logs from the start of Mesos Master processes: `journalctl -u mesos-master -b`
Slave	`dcos-mesos-slave.service`	DC/OS	Tasks running on agents		DC/OS distributed systems kernel agent.	Loss of an agent within DC/OS may cause the agent to become unregistered with the DC/OS masters. It may also negatively affect the tasks running on the agent but existing tasks, executed by an executor should continue running.	Check to make sure the slave process is running on the agent node. Connect to the mesos-agent UI: http://<agent-ip-addr>:5051/slave; investigate logs for the agent service: `journalctl -u dcos-mesos-slave -b`
Slave Public	`dcos-mesos-slave-public.s`	DC/OS	Tasks running on public agents		DC/OS distributed systems kernel agent on public nodes within a cluster	Loss of an agent within DC/OS may cause the agent to become unregistered with the DC/OS masters. It may also negatively affect the tasks running on the agent but existing tasks, executed by an executor should continue running.	Check to make sure the slave process is running on the agent node. Connect to the mesos-agent UI: `http://<agent-ip-addr>:5051/slave`; investigate logs for the agent service: `journalctl -u dcos-mesos-slave -b`
Mesos-DNS	`dcos-mesos-dns.service`	DC/OS	mesos.master; /etc/resolv.conf; detect_ip		Mesos-DNS provides domain name based service discovery for tasks running with a DC/OS environment.	Issues with Mesos-DNS will affect service discovery and the ability for apps to resolve names correctly within the cluster. This may manifest itself as service/app outages.	You can verify if name resolution is working within the cluster by performing nslooksups and/or dig @ commands from a master or agent node in the cluster, looking for service names. Additionally, investigate the logs for mesos-dns: `journalctl -u dcos-mesos-dns -b`

DC/OS Components, System Components, and Basic Troubleshooting

Component	Service Name	Layer	Dependencies	Subcomponent	Description	Impact to Environment	Where to begin
Spartan	`dcos-spartan.service`	DC/OS	mesos-dns		Spartan forwards DNS requests to multiple DNS servers.	Name resolution issues will arise with loss of Spartan.	Investigate the logs for Spartan: `journalctl -u dcos-spartan -b`
NTP	`ntpd.service`	Linux	Linux kernel; ntpd; time servers		Keeps time constant on hosts within an environment.	Time drift is a major disruptor or incident causing issue with DC/OS clusters. Time must be synchronized across the cluster in order for components, masters, and slaves to work correctly.	`journalctl -u ntpd.service; ntptime; timedatectl`
Kernel Parameters		Linux	assorted /etc configuration files		Kernel parameters drive how the host operating system runs, responds to needs, and ultimately performs to manage workloads on the system.	Improperly configured Kernel(s) on hosts within a DC/OS cluster can have many negative impacts on running, capacity, and performance.	Systools.