Search

DCOS Architecture

The Mesosphere DCOS architecture is comprised of:

- Marathon
- Mesos
- Mesosphere repository
- Security
- Service discovery
- ZooKeeper

Marathon

Marathon (/services/marathon/) is the “init system” for DCOS. It starts and monitors applications and services, automatically healing any failures.

Mesosphere repository

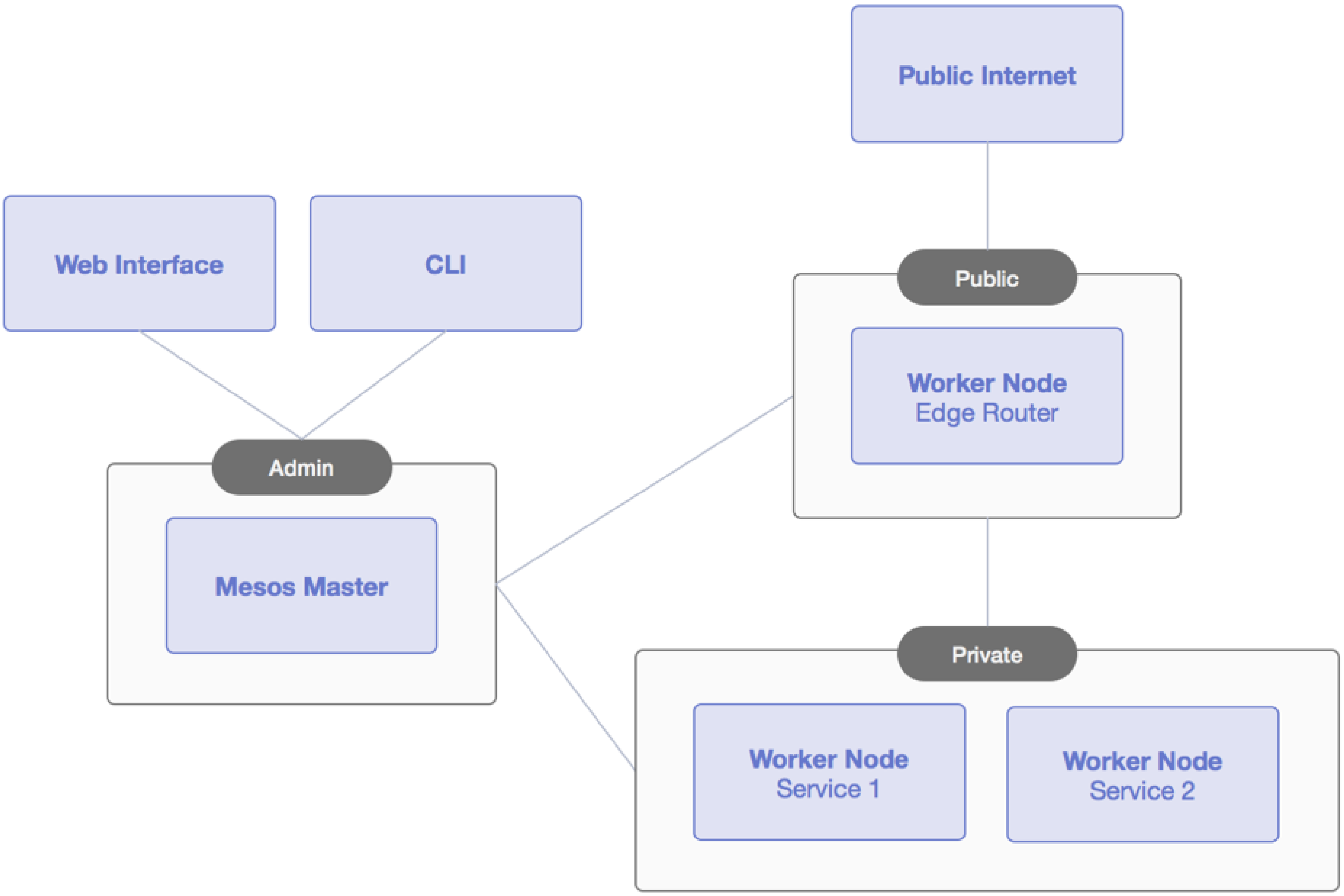
The Mesosphere repository contains supported DCOS services software packages. With the DCOS CLI installed, you can use the `dcos package command` to install and manage software packages from this public repository.

Mesos

Designed at UC Berkeley and hardened at Twitter, Mesos (http://mesos.apache.org/) is the distributed systems kernel at the core of Mesosphere DCOS. You can access the Mesos web interface at `<hostname>/mesos` .

Security

The DCOS offers the admin, private, and public security zones.



Admin

The admin zone is only accessible through an HTTP connection and provides access to your entire cluster through URL routing. The Mesos masters are located in the admin network and must be only accessible from internal networks. By default there are either 1 or 3 masters created by the DCOS template, depending on your specifications.

Private

This private zone is a non-routable network that is only accessible from the admin and public security zones (i.e. not accessible from the outside world). Deployed apps and services are run in the private zone. This zone is where most of your Mesos worker nodes are run. By default five private worker nodes are created by the DCOS template.

Public

The public zone is where publicly accessible applications are run. By default a single worker node is created by the DCOS template. Generally, only a small number of worker nodes are run in this zone. The edge router forwards traffic to applications running in the private zone.

The worker nodes in the public zone are labeled with a special role so that only specific tasks can be scheduled here. These worker nodes have both public and private IP addresses and only specific ports open in the firewall.

A simple app router is provided in the public zone to route HTTP requests to apps running securely in Marathon. Marathon is started with `--mesos_role` set to `slave_public` so that it gets offers from the public worker nodes. Marathon also sets `--default_accepted_resource_roles` to `*` so that by default apps use unreserved resources and do not launch on the public worker nodes. If you want to configure apps to use resources from the public worker nodes, you must set `acceptedResourceRoles` to `slave_public` or `,slave_public`, depending on whether you want to also use unreserved resources. For more information, see the tutorial, Deploying a Web App with DCOS (/tutorials/deploywebapp/), where an app is deployed in the public zone to route HTTP requests.

Implementation Details

- This version of DCOS has no authentication.
- The DCOS CLI and web interface do not currently use an encrypted channel for communication. However, you can upload your own SSL certificate to the masters and change your CLI and web interface configuration to use HTTPS instead of HTTP.
- You must secure your cluster by using security rules. It is strongly recommended that you only allow internal traffic.
- If there is sensitive data in your cluster, follow standard cloud policies for accessing that data. Either set up a point to point VPN between your secure networks or run a VPN server inside your DCOS cluster.

Service discovery

The DCOS uses Mesos-DNS (https://github.com/mesosphere/mesos-dns) for service discovery. Mesos-DNS allows applications and services that are running on Mesos to find each other by using the domain name system (DNS), similar to how services discover each other throughout the Internet.

ZooKeeper

The DCOS uses ZooKeeper, a high-performance coordination service to manage the installed DCOS services. To coordinate and manage the ZooKeeper system, we've integrated Exhibitor for ZooKeeper (https://github.com/Netflix/exhibitor).

@mesosphere (https://twitter.com/mesosphere)