

Docker@OVH with Mesos/Marathon

June 28th 2016

@brouberol

Devops / Python charmer

@devatoria

Devops / Golang artist

The Docker ecosystem



- **Docker Engine:** run containerized processes (aka: containers)
- **Docker Compose:** run multi-container applications on a single host
- **Docker Machine:** provision and manage multiple remote Docker hosts
- **Docker Swarm:** orchestrate multi-containers app on multi hosts, manage networks, health-checks, etc

Docker orchestrators



Orchestrator: lets you automate the creation, monitoring, and deployment of resources in your environment.

- **Docker Swarm** (Docker Inc.)
- **Mesos/Marathon** (Mesosphere)
- **Kubernetes** (Google)

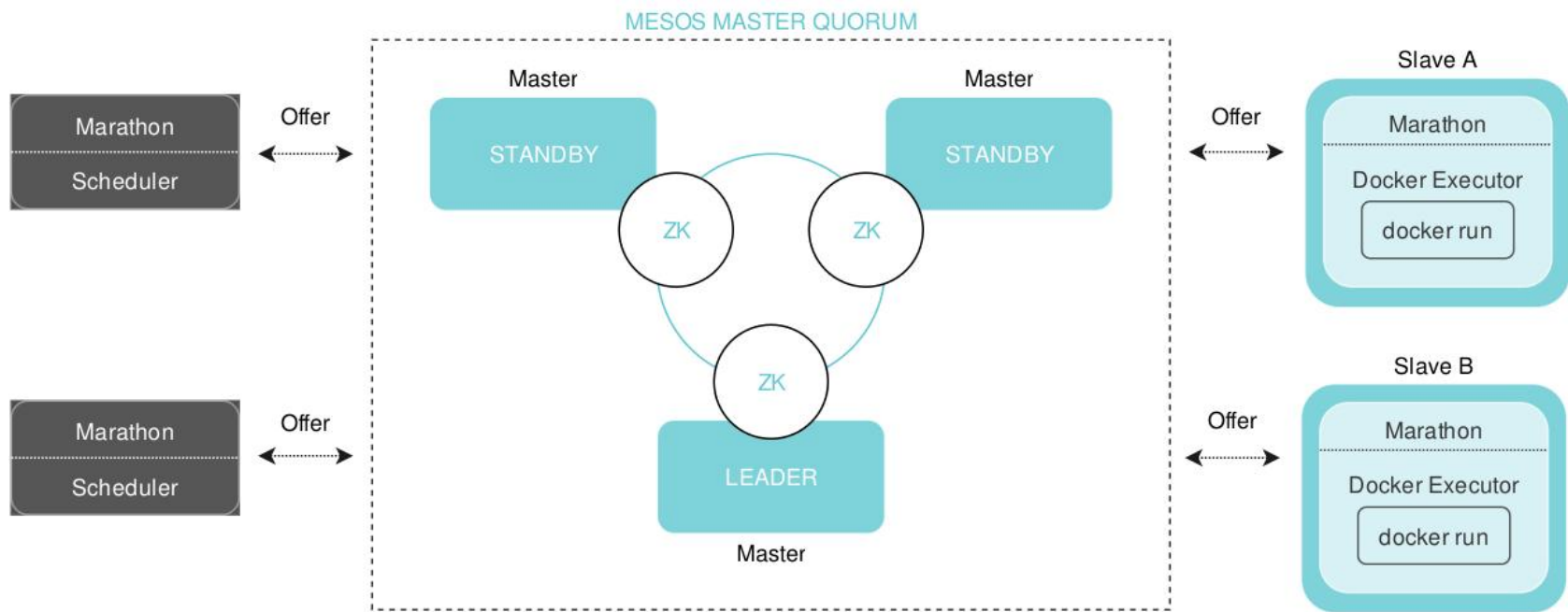
Multiple orchestrators: all of each are now **standards!**

DockerAAS @ OVH



- You build your image, we run your containers
- Based on Mesos/Marathon
- Will support all orchestrators in the future
- Stateless/stateful containers
- Run on any OVH infrastructure (cloud, dedicated, etc)
- runabove.com/docker-with-mesos-marathon

Mesos/Marathon



Marathon UI

The screenshot displays the Marathon UI interface. At the top, there are navigation tabs for 'Applications' and 'Deployments', with a notification badge on 'Deployments'. The 'Applications' tab is active, showing a list of applications under the 'monitoring' group. The sidebar on the left provides filters for 'STATUS' (Running: 117, Deploying, Suspended, Delayed, Waiting: 2) and 'HEALTH' (Healthy: 115, Unhealthy, Unknown: 4). A 'LABEL' dropdown is set to 'Select'. The main table lists 14 applications, all with a status of 'Running' and a health of 'Healthy' (indicated by green progress bars). Each application row includes a name, a list of labels, CPU usage (0.1), memory usage (512 MIB), status, running instances (1 of 1), and a health indicator.

Name	CPU	Memory	Status	Running Instances	Health
monitoring	0.1	256 MIB		1 of 1	Healthy
c001-55c85514092747000a90df00	0.1	512 MIB	Running	1 of 1	Healthy
c001-55c89990092747000a90df29	0.1	512 MIB	Running	1 of 1	Healthy
c001-55c8bd08092747000a90df3e	0.1	512 MIB	Running	1 of 1	Healthy
c001-55c8bd49092747000a90df41	0.1	512 MIB	Running	1 of 1	Healthy
c001-55c8bda0092747000a90df44	0.1	512 MIB	Running	1 of 1	Healthy
c001-55c8bde1092747000a90df47	0.1	512 MIB	Running	1 of 1	Healthy
c001-55c8be3c092747000a90df4a	0.1	512 MIB	Running	1 of 1	Healthy
c001-55c8be63092747000a90df4d	0.1	512 MIB	Running	1 of 1	Healthy
c001-55c8be9e092747000a90df50	0.1	512 MIB	Running	1 of 1	Healthy
c001-55c8bec6092747000a90df51	0.1	512 MIB	Running	1 of 1	Healthy
c001-55c8bf03092747000a90df54	0.1	512 MIB	Running	1 of 1	Healthy
c001-55c8bf67092747000a90df59	0.1	512 MIB	Running	1 of 1	Healthy
c001-55d30c6144e17000a4e3a08	0.1	512 MIB	Running	1 of 1	Healthy
c001-55e018a7dc38670008a182a1	0.1	512 MIB	Running	1 of 1	Healthy

Marathon API

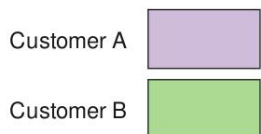
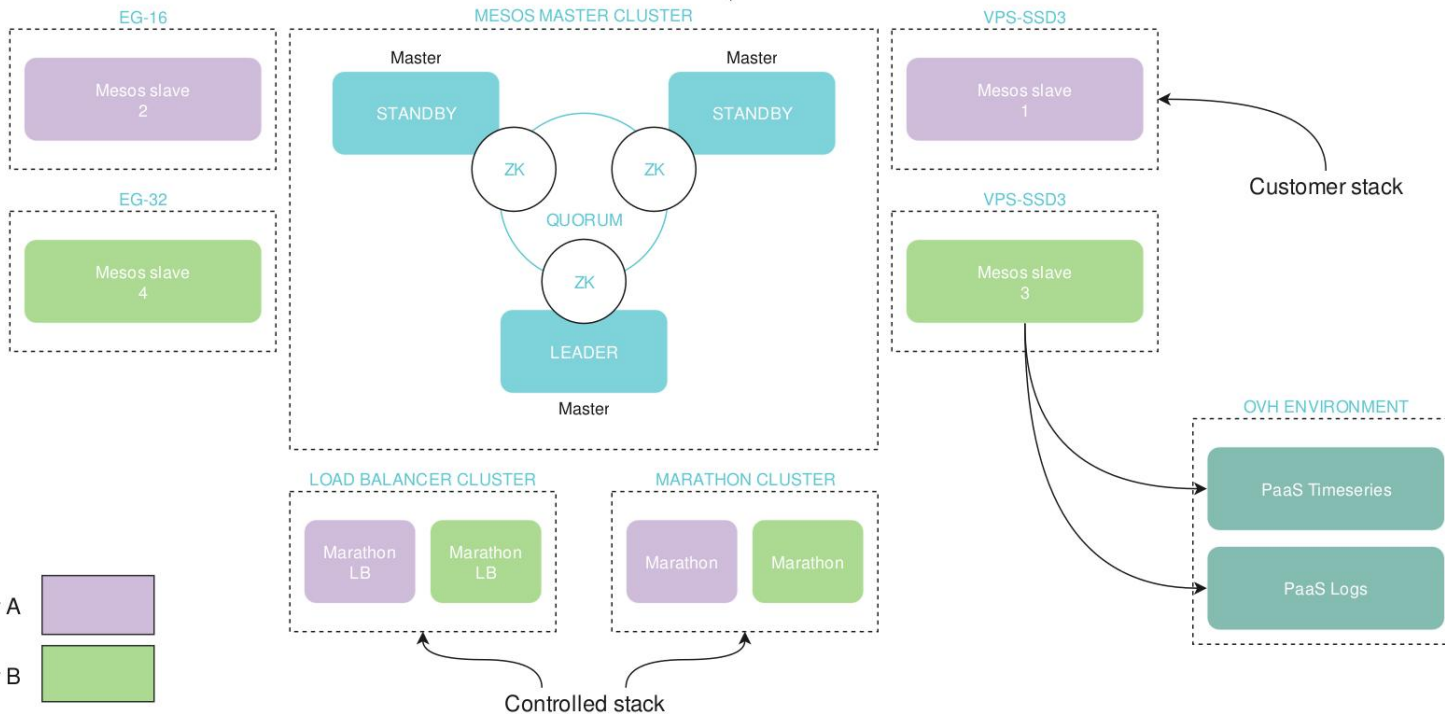


- Start/pause/stop/redeploy applications
- Used from our CI environnement (Bamboo/CDS) to create or update application after a CI build
- Monitor app health status
- Fetch marathon event stream
- etc

Mesos/Marathon @ OVH



Managed stack

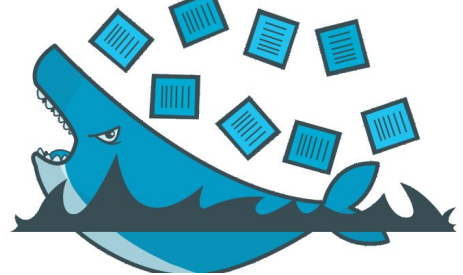


(Big) Data



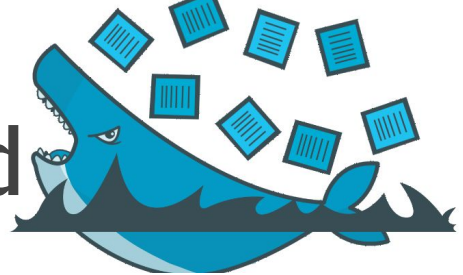
- Docker 1.11.1
- 3000 containers (stateful/stateless) running at all times
- 2000 (v)CPUs
- 9TB RAM
- 6 clusters (4 prod, 2 beta)
- 500 hosts (VMs/dedicated)
- 5 people

Docker in stormy weather: storage drivers



-
- Devicemapper (default) only suitable for development
 - OverlayFS: way faster but lots of inode consumed
 - ZFS: to be tested, not easily available on Linux
 - Messy documentation
 - Some (dated) benchmarks

Docker in stormy weather: docker under load



- Some deadlocks: `docker ps/info/run` stuck
- Goroutine leaks: kernel ran out of threads

Thanks!
Questions?