Blueprints for a Service Ecosystem on Mesos/Marathon

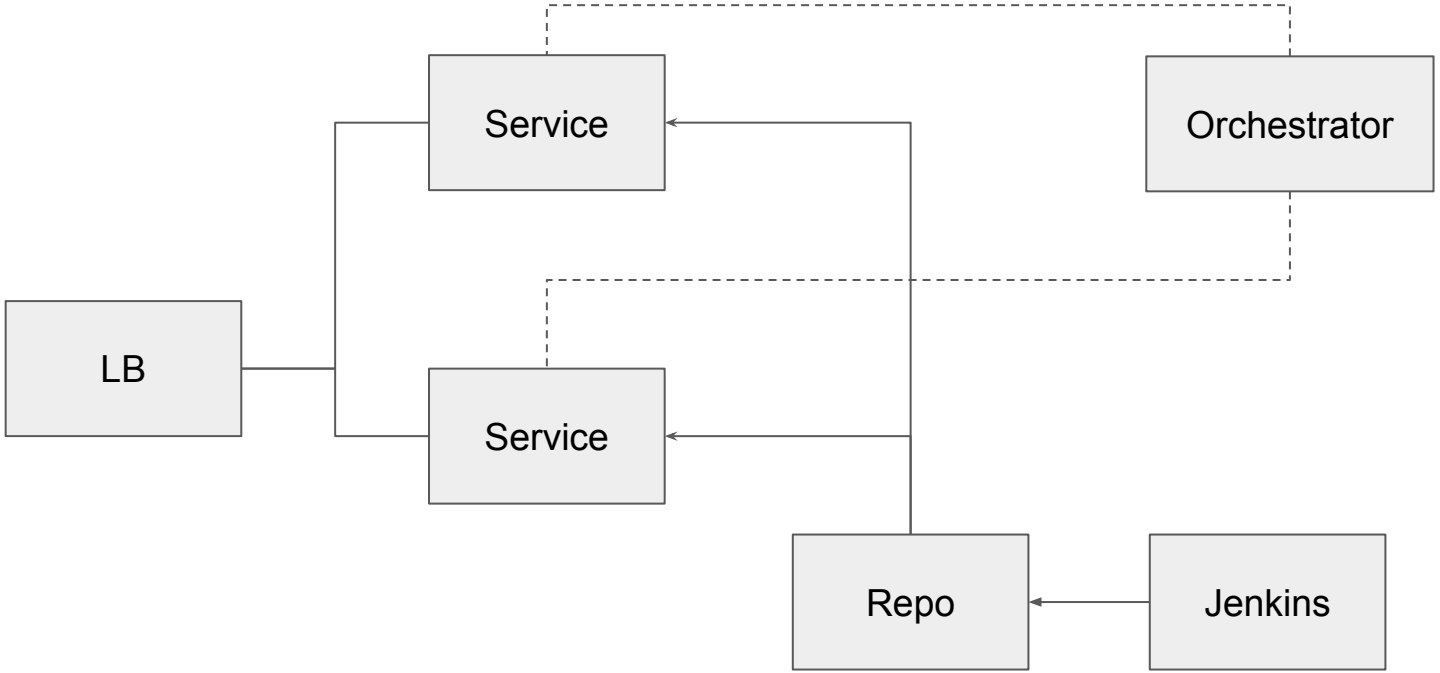
Santanu Sinha

Topic of the Day

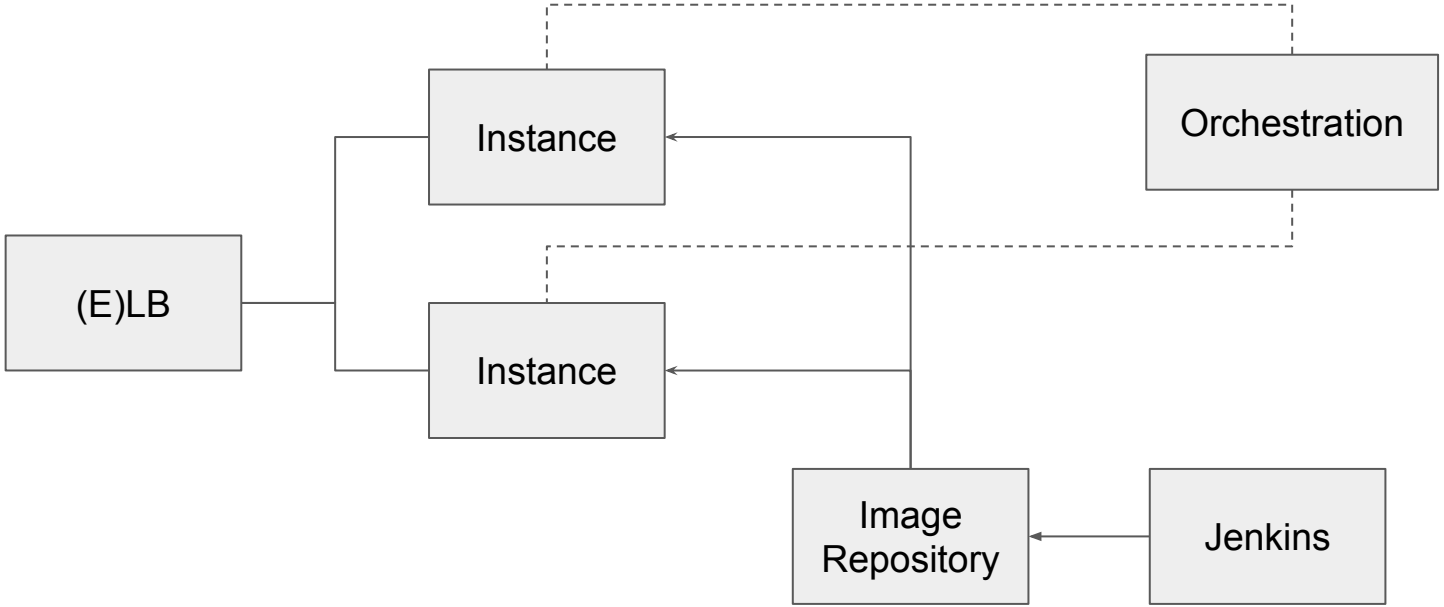
Infrastructure for data platform

- Resource utilization
  - High realtime workloads during day
  - High batch workloads at night
- Management of resources
- Quickly scale resources
- Reducing coupling with business systems
  - High volume data ingestion through endpoints
- Isolation between systems inside the platform
- Self-serve multi-tenant platform

Traditional Setup



Cloud Setup


Instance
Orchestration
(E)LB
Instance
Image
Repository
Jenkins

Things to be fixed...

- Resource Utilization
- Scaling
- Deployment
- Management and log access
- Service Discovery
- Devops friendliness
- Tool development

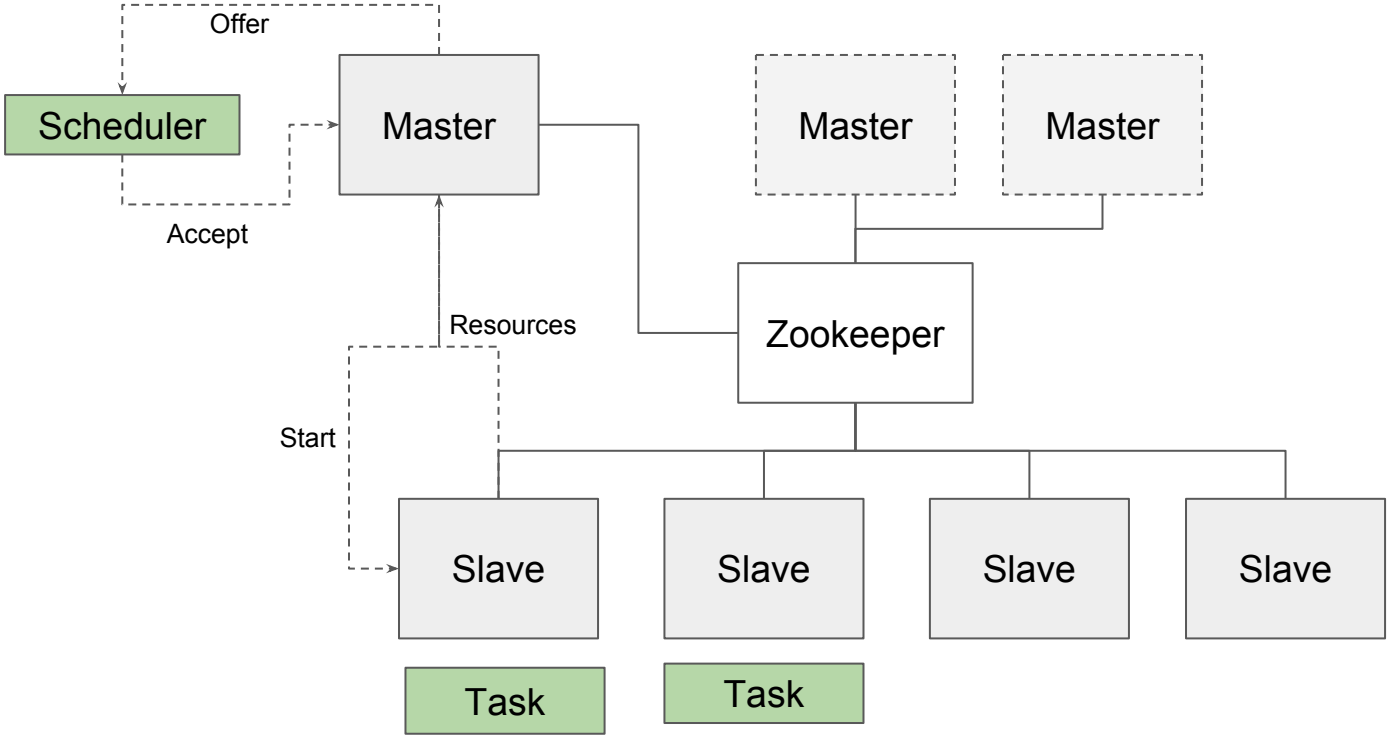
Apache Mesos

A Distributed System Kernel

- Cluster level resource manager
- Abstracts CPU/Memory/Storage and other cluster resources
- Provides fault-tolerance for frameworks
- Fault tolerance for itself using ZooKeeper
- Enables easy scaling

Reference: http://mesos.apache.org/

Mesos Cluster



Need of the hour: packaging/isolation

A way to package services to be easily run on mesos.

- Mesos frameworks for all services is not viable
- Popular rest frameworks that everybody is comfortable with
  - Dropwizard
- Must support any language (java/node/ruby/…)
- Must provide isolation
- Easy to build, deploy and can be easily integrated with CI tools like Jenkins

Cgroups

Linux kernel feature to control the isolation of processes.

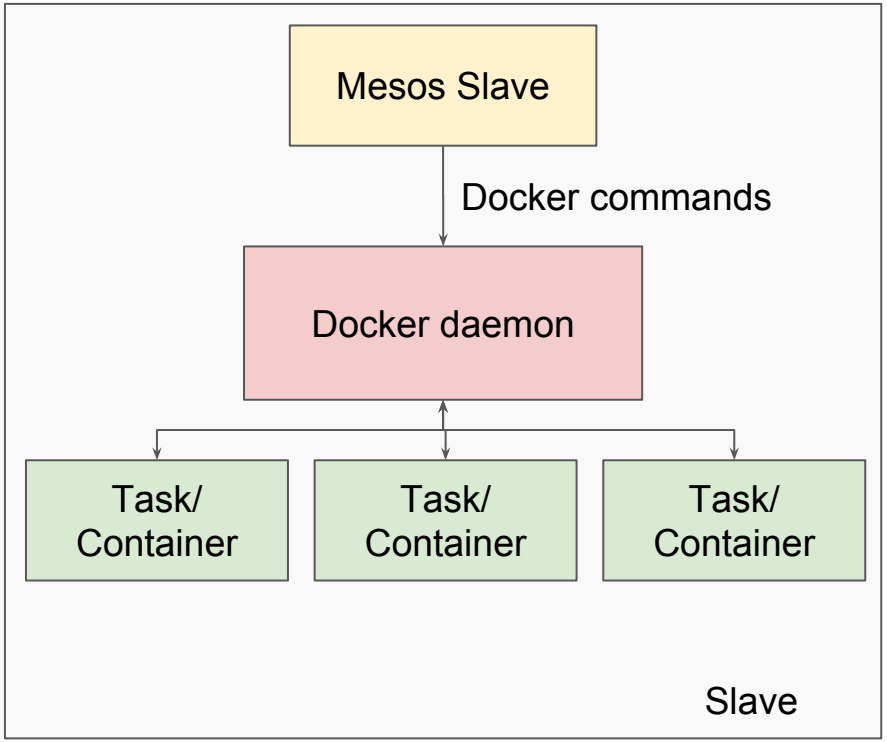
- CPU
- Memory
- Network
- Block device I/O
- Freeze

Reference: RHEL documentation

Docker

- Abstraction layer over isolation facilities built in the kernel
  - Cgroups
  - Namespaces
  - Union Filesystems (aufs, OverlayFS)
- Written in go
- Huge community support
- Rich ecosystem of pre-existing images for popular software
- Rich tooling
  - Maven plugin from spotify and others to build docker images
  - Easy to setup repositories
  - Works on Linux and others using docker-toolbox
- Reference: https://www.docker.com/

Mesos with docker



Mesos Containerizer

- Task isolation
- Force tasks to run using allocated resources only
- Allow pre-packaged images to be run, providing optimal environments for tasks
- Supported containerizers:
  - Docker - uses docker commands to run images in tasks
  - Mesos- uses isolators provided by mesos:
    - PID
    - Disk
    - Network

Reference: http://mesos.apache.org/documentation/latest/containerizer/

Need of the hour: Orchestrator

A framework/scheduler that can take container location etc from user and spawn them on slave nodes of mesos cluster

- Should take configurable resource limits
- Should provide fault tolerance
- Should provide dynamic scaling capabilities
- Should provide deployment capabilities
  - Rolling deployments
  - Easy upgrades
  - Constraints based deployment targeting
- Health monitoring

Marathon

Container orchestration framework on top of mesos

- Meets all the requirements
- Highly available
- REST APIs
- Functional web console
- Event bus for integration
- Metrics
- Stateful app support (beta)
- IP per container support (experimental)
- REST and Web requests are proxied to the master

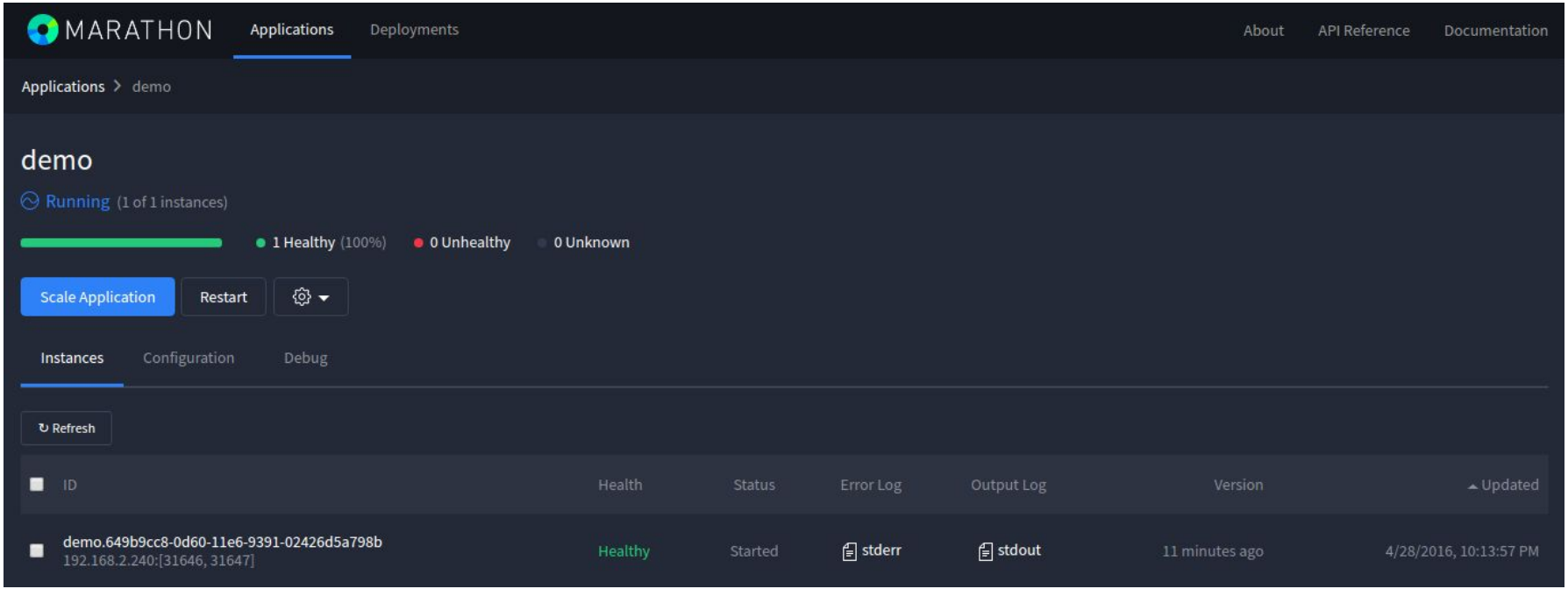
Reference: https://mesosphere.github.io/marathon/

Anatomy of a Marathon Deployment Descriptor

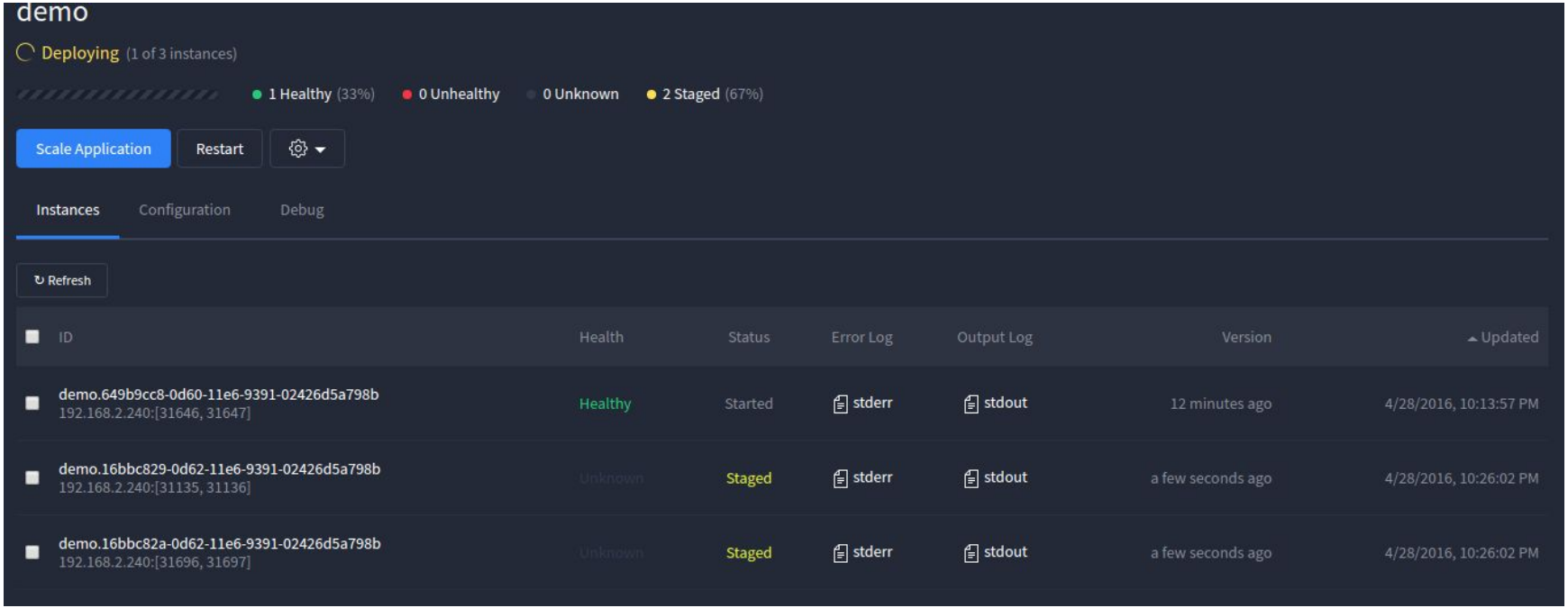
```
{
  "id": "/demo",
  "cpus": 0.5,
  "mem": 512,
  "instances": 1,
  "container": {
    "type": "DOCKER",
    "docker": {
      "image": "santanusinha/docker-demo-service",
      "network": "BRIDGE",
      "portMappings": [{
          "containerPort": 8080,
          "hostPort": 0,
          "protocol": "tcp"},
        {"containerPort": 8081,
          "hostPort": 0,
          "protocol": "tcp"}]
    }
  },
```

```
  "env": {
      "ZK_CONNECTION_STRING" : "zk-server1"
    },
    "labels": {
      "traefik.portIndex": "0",
      "traefik.enable": "true"
    },
    "healthChecks": [
      {
        "protocol": "HTTP",
        "path": "/healthcheck",
        "portIndex": 1
      }
    ]
}
```

Marathon console



Marathon scale up/down



Chronos

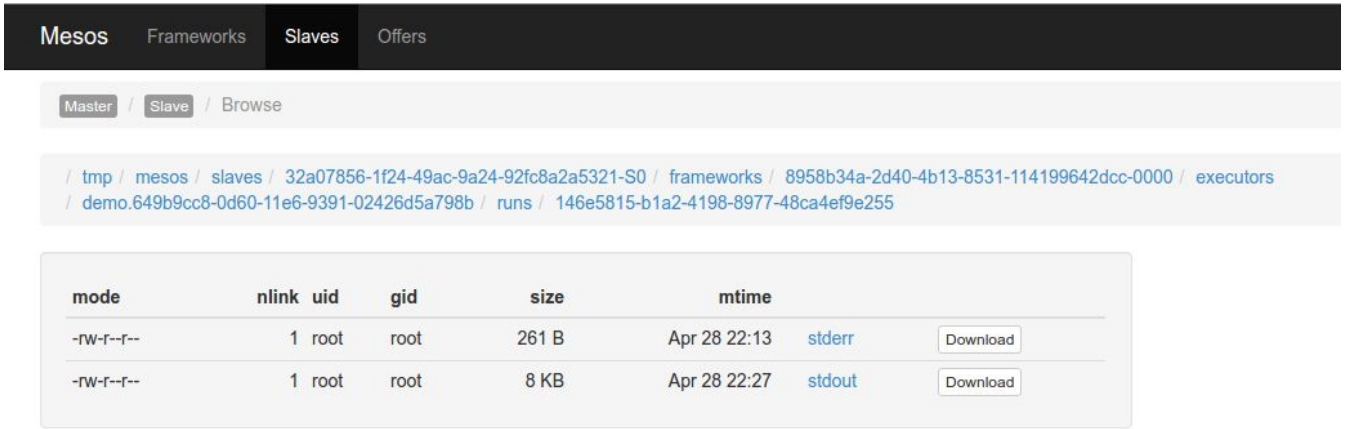
Distributed and fault tolerant replacement for *cron*

- Supports both shell commands and docker jobs
- Supports arbitrarily long dependency chains between jobs
- Provides scripts to submit jobs to remote Hadoop clusters and receive async notification when they complete

Reference: https://mesos.github.io/chronos/

Service Logs

- All service logs available directly from mesos console for download
- Supports tail like functionality
- Host level fluentd etc can be deployed to aggregate docker logs and push them to central store (ES/HDFS/Kafka etc)
- No need for devs to login to individual containers/hosts

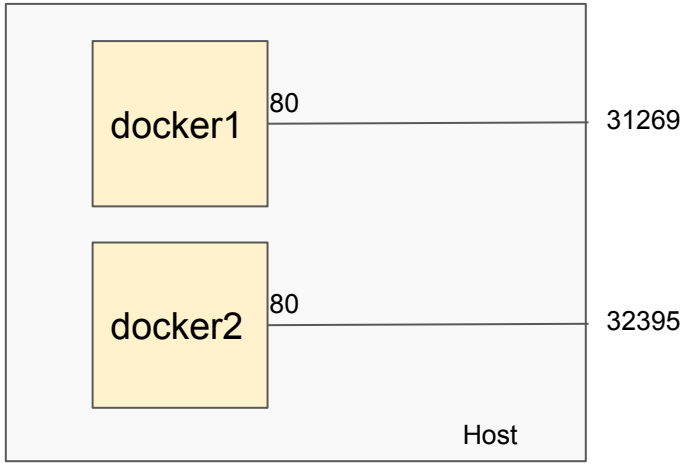


Metrics

- Framework level metrics available from marathon
  - Direct push to graphite
  - Available as REST API for integration with other systems
- Service metrics can flow into central metrics system as usual
- Event bus
  - Provides callback for events on marathon
    - Deployments
    - Restarts
    - Scale up/down

Need of the hour: Addressing

- Services deployed on containers inside machine
- Bridged IP visible on local machine only
- Host level port binding is not practical
  - What if two services run on same port?



Mesos DNS

- Provides records for every app deployed on marathon
  - A: <app>.marathon.mesos
  - SRV: _<app>._tcp.marathon.mesos
- Provides other meta entries like:
  - mesos.master
- Clients need to be able to contact the mesos dns servers
  - Needs host level resolver config
- Heavy dependency on ZK
- No way to take services OOR
- No sharding support

Reference: http://mesosphere.github.io/mesos-dns/

Service Discovery with Ranger

- Java library
- Zookeeper based with local caching
  - Continues to work even without zookeeper as long as the topology does not change
- Healthchecks
- Sharding based on custom data
- Customizable client side node and shard selection
- Dropwizard bundle available
  - https://github.com/santanusinha/dropwizard-service-discovery
- Can be used by smart clients to call services inside mesos from outside
  - Netflix Feign with Ranger discovery (https://github.com/phaneesh/feign-ranger)

Reference: https://github.com/flipkart-incubator/ranger

Need of the hour: Expose services

A way to expose services to outside the cluster

- Should be able to dynamically reconfigure itself
  - Update configs based on topology
- Expose and bring down endpoints based on app lifecycle on Marathon

Marathon LB

Python script to update HAProxy configs

- Can poll marathon or use eventbus and SSE
- Generates new HAProxy config and reloads
- Highly scalable due to HAProxy
- Provides docker build

Reference: https://github.com/mesosphere/marathon-lb

Traefik

Standalone reverse proxy and load balancer

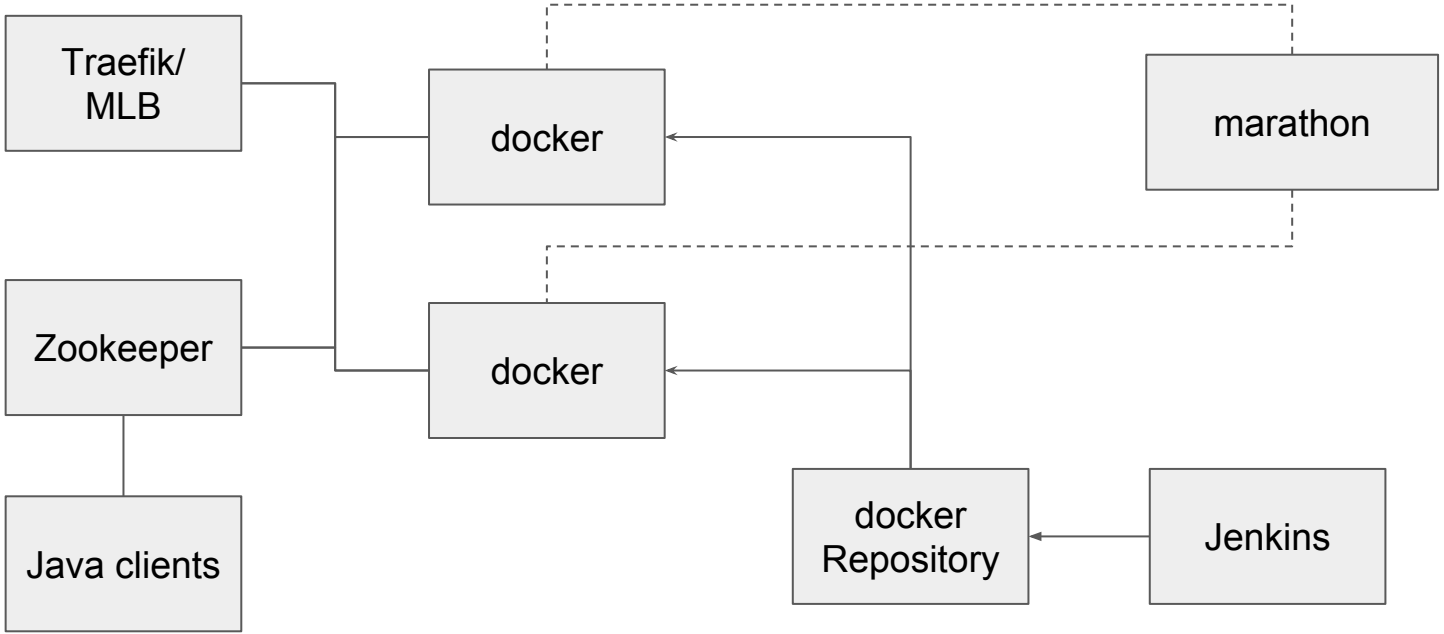
- Supports multiple backends
  - Docker, mesos/marathon, consul, zookeeper etc
- Single binary written in go
- Simple configuration
- Hot config reload
- Backend circuit breakers
- Websockets, SSL (frontend and backend), HTTP 2
- Integrates with event bus

Reference: https://traefik.io

Mesos ecosystem

- Resource management using Mesos
- Docker as deployment and execution container
- Container orchestration using Marathon/Aurora
- Scheduled execution using Chronos
- Service Discovery using Ranger/Mesos DNS
- Load balancing using Traefik/Marathon LB

Bringing it all together



Caveat emptor

- Configuration management needs to be planned properly
- Marathon injects multiple environment variables into docker
  - Can be used to find the ip of the machine where it has been deployed ($HOST)
  - Can be used to find the actual host port to which the docker port has been bridged to
    - If exposed port is 80, variable will be called $PORT_80
- Service discovery and integration points with outside clients needs to be planned properly
  - Stable IP per container support will make life much simpler for everyone
- Traefik will only expose apps that have traefik.enable: true label set
- Health check intervals etc are configurable
- Deployment strategies can be managed using variables in marathon config
- Constrains should be used to make sure service achieves HA

Miscellany...

- Stateful features of marathon can be used to run databases
- Mesos supports authentication for framework registrations, slave registrations etc
- Battle tested in large deployments
- Many projects like spark support mesos as resource manager
- Auto-scaling can be done using Relay
- DCOS provides enterprise support and well as easy to run cloudformation templates for free for AWS setups
  - Provides packaging support on top of marathon
  - Provides a command line client
  - Nice web UI
- Frameworks: http://mesos.apache.org/documentation/latest/frameworks/

Questions

Thank You

Twitter: @santanu_sinha

Demo code available at:
https://github.com/santanusinha/dropwizard-marathon-stub