Lighthouse

A tool to create and manage Docker instances across multiple cloud platforms

What is Docker?

Distributed applications pose a large number of challenges; two of the most important being consistency across environments and scalability. Docker provides a thin application layer on top of server operating systems, isolating application execution and allowing new applications (containers) to start quickly from uniform file-system images. Docker removes the need for a full virtual machine, allowing applications to share system resources, providing near-native runtime speed.

Why Lighthouse?

Problem

Docker is great, but I want to:

- Manage my application as a whole, not as individual Docker instances
- Utilize cloud providers like GCE or AWS to run my application
- Control who can change my application
- Do all of this through a user interface

Solution

Lighthouse centralizes the management of multiple Docker instances, regardless of their hosted platform or geographical location.

Design Requirements

Functional Requirements

- Docker core functionality
  - Container/image management
- Lighthouse custom functionality
  - Application level control and versioning
  - Cloud provider interfacing
- Application analysis
  - Logs, history, usages, etc
- User management

Non-Functional Requirements

- Security - specifically from:
  - Cross-site scripting, session hijacking, data exposures
  - Unauthenticated and unauthorized requests
- Extensibility and documentation
  - Additional functionality should be straightforward
  - Users should be able to create their own frontend
- Low latency
  - Early error detection in the pipeline

Use Cases

Release Manager

- Log in/Authenticate
- View currently deployed containers
- Deploy and start an application
- Rollback an application
- View application logs

IT Administrator

- Log in/Authenticate
- Add cloud providers (beacons)
- Manage users
- View application status

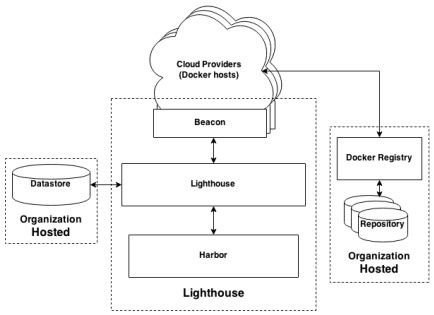
Technology

Lighthouse: docker, Go, PostgreSQL

Beacon: DigitalOcean

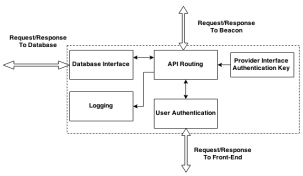
Harbor: AngularJS

Design - System



Design - Components

Lighthouse (server-side data management)



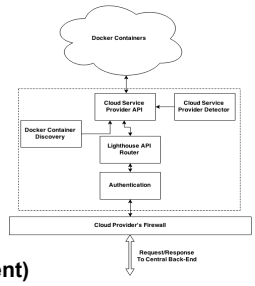
Provides a central access point to Docker instances

- Handles user authentication
- Manages application deployments and history
- Routes requests to Docker instances

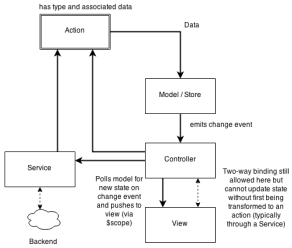
Beacon (provider interfacing)

Create software "drivers" for detecting cloud environments

- Probe Providers for existing/owned VMs with Docker
- Establish basic API for Lighthouse
- Create pluggable interface for future cloud providers



Harbor (web client)



Provides a user interface for managing individual Docker instances and performing large deployments.

- Unidirectional data-flow provides predictable state changes
- Support for streaming APIs
- Client-side template rendering trades initial download time for responsive UI updates

Testing

Plan:

- Test-Driven Development
- Continuous Integration
- Code Review each Pull Request

Services:

- GitHub (repository hosting)
- Travis CI (tests)
- Coveralls (coverage)

May15-17

Caleb Brose, CprE
Chris Fogerty, CprE
Nick Miller, CprE
Rob Sheehy, SE
Zach Taylor, CprE

may1517.ece.iastate.edu
github.com/lighthouse

Client: Workiva

Advisor: Dr. Simanta Mitra