

How to create a Docker Cloud with Brooklyn, jclouds and Clocker.io

Andrea Turli

About me

APACHE CON
NORTH AMERICA

Andrea Turli

Software Engineer

Open Source passionate

github.com/andreaturli



Open Source Application Management Specialists

@turlinux

Agenda

APACHE CON
NORTH AMERICA

An aerial photograph of a city grid, likely San Francisco, showing a central circular plaza with a star-shaped monument. The monument has the letters 'E', 'S', 'T', and 'A' on its points. The surrounding city blocks are arranged in a regular grid pattern.

Clocker Introduction
What is a Docker Cloud?
Demos
Advanced features

@turlinux



What's Clocker.io?

What does it do?

Clocker supports **single-click deployment** and **runtime management** of multi-node applications that run on *Docker containers* that can be distributed across multiple **Docker engines**

Open Source

Apache 2.0 Licensed

java

[@clockercentral](http://clocker.io)

Status

Started by @grkvlt and @turlinux

~500 total commits

~10 contributors

+37 forks on GitHub

On the shoulder of the giants

APACHE CON
NORTH AMERICA

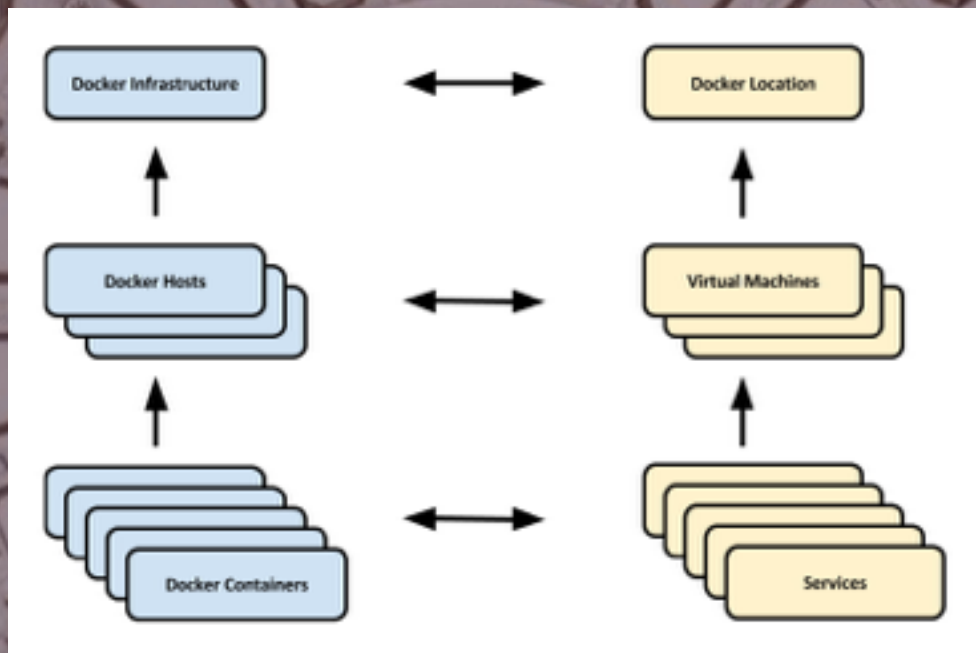
apache brooklyn
jclouds



@turlinux

Clocker and Apache Brooklyn

Brooklyn Blueprint



Brooklyn Location

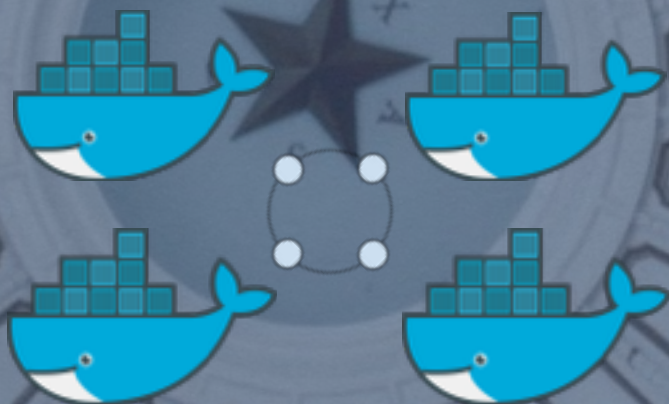


Demo: Building a Docker Cloud

On the shoulder of the giants

APACHE CON
NORTH AMERICA

apache brooklyn
jclouds



@turlinux

Application Management Platform

Apache 2.0 Licensed

Donated to ASF - Incubator Project

<http://brooklyn.io> @brooklyncentral

Status

Founded by Alex Heneveld and Aled Sage

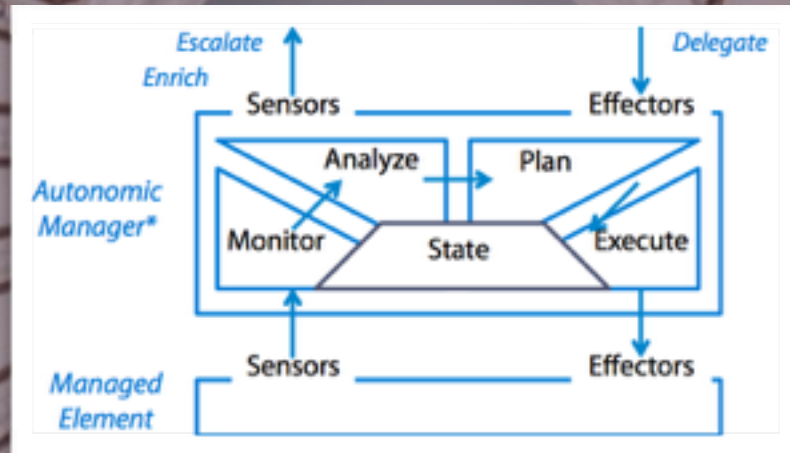
~11K total commits

~40 contributors

+50 forks on GitHub

Hierarchical autonomic management

MAPE-K control loop for self-adaptive systems



Deploy, Manage and Monitor **autonomic Blueprints**

Blueprint

Capture an application's initial *topology* plus *policies*

Topology

Component wiring, groupings and management hierarchy

Policy

Governs an app's behavior (Cluster Mgmt, Failover ...)

Location

Target environment for blueprint instantiation

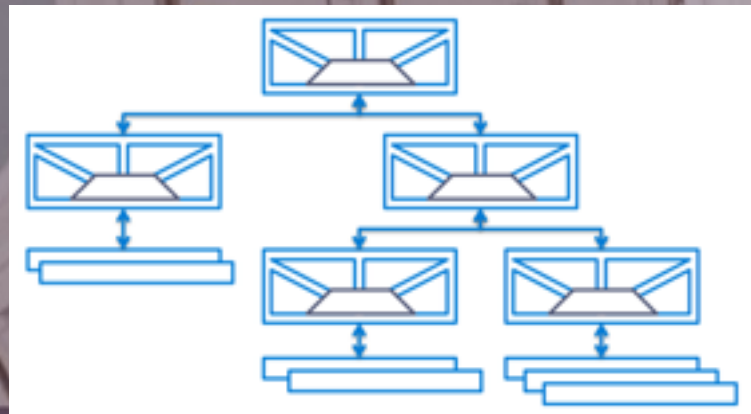
App Topologies

Policies

+ Service Blueprints + Locations

⇒ Deployed Applications
with
Dynamic Management

```
name: My Web Cluster
location: localhost
services:
- type: brooklyn.entity.webapp.ControlledDynamicWebAppCluster
  name: My Web
  brooklyn.config:
    wars.root: https://bit.ly/brooklyn-example-hello-world-sql-webapp-0.6.0.war
    java.sysprops:
      brooklyn.example.db.url: >
        $brooklyn:formatString("jdbc:{{user}}@{{password}}",
          component("db").attributeWhenReady("datastore.url"),
          "visitors",
          "brooklyn",
          "brooklin")
- type: brooklyn.entity.database.mysql.MySqlNode
  id: db
  name: My DB
  brooklyn.config:
    creationScriptUrl: https://bit.ly/brooklyn-visitors-creation-script
```



A multi-cloud toolkit for the Java platform

Apache 2.0 Licensed - Top Level Project

<http://jclouds.org>

Status

Started on April 2009 by Adrian Cole

+12K total commits

~200 contributors (65 last year)

+200 forks on GitHub

latest stable release 1.9.0

jclouds

APACHE CON
NORTH AMERICA



Portable abstractions: Compute, BlobStore, LoadBalancer

jclouds-docker @turlinux

<https://github.com/jclouds/jclouds-labs> in 1.9.0

This initial work kicked off Clocker

```
// get a context with docker that offers the portable ComputeService api
ComputeServiceContext context = ContextBuilder.newBuilder("docker")
    .credentials(email, password)
    .modules(ImmutableSet.<Module> of(new Log4jLoggingModule(),
                                     new SshjSshClientModule()))
    .buildView(ComputeServiceContext.class);
ComputeService client = context.getComputeService();

String sshableImageId = "your-sshable-image-id"; // this can be obtained using `docker images --no-trunc` command
Template template = client.templateBuilder().imageId(sshableImageId).build();

// run a couple nodes accessible via group container
Set<? extends NodeMetadata> nodes = client.runNodesInGroup("container", 2, template);

// release resources
context.close();
```

Docker is a cloud provider which spins up containers up instead of VMs



Docker REST API

APACHE CON
NORTH AMERICA

Docker REST API v1.17

Docker Remote API <- supported by jclouds-docker

ContainerAPI

ImageAPI

MiscAPI

Docker Hub API

Docker Registry API

@turlinux



A platform to pack, ship and run any application as a lightweight container.

Apache 2.0 Licensed
<https://www.docker.com/>

Status

Started on March 2015
+14K total commits
~840 contributors
+4300 forks on GitHub



Docker Networking 101

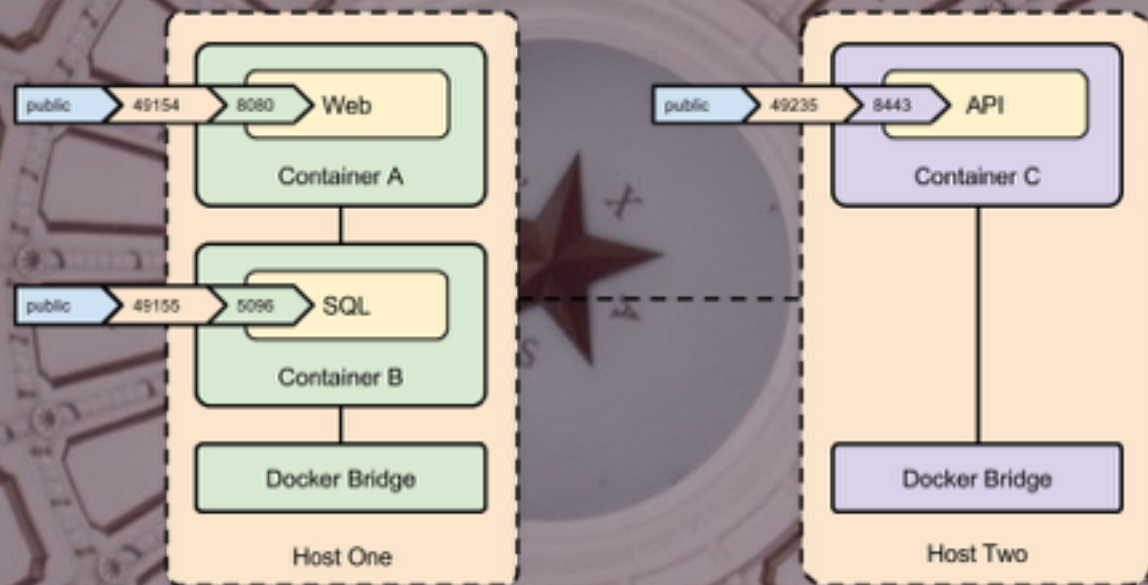
Natively, each container runs isolated

A bridged network is provided by Docker

Exposing ports

Port Forwarding (mapping port) to “external interface”

what if I have a cluster of nodes wants to use the same port?



Software Defined Networking

APACHE CON
NORTH AMERICA

Host to Host Communication

Same LAN Segment

No Port Forwarding

Natural Application Configuration

Initial Driver was Erlang Distributed Protocol and EPMD based Applications

Uses Available Provider

Currently Weave, Metaswitch Calico

Ethernet Switch

User Space

Weave agent is a Docker Container

Routes TCP Traffic

Forwards over TCP



Demo: Deploying an application on a Docker Cloud

Orchestrated Docker 1.5.0 deployment with Weave

Automated application deployment over containers provisioned on multiple Docker Hosts

Automated attachment of containers to multiple dynamic networks created on-the-fly by the SDN provider

Deep dive

APACHE CON
NORTH AMERICA

Container Management
Container Placement
Provisioning
Headroom
Networking

@turlinux

Deep dive: Container Mgmt

APACHE CON
NORTH AMERICA

Sources

Docker Image Definition

Docker Hub

Dockerfile

Brooklyn Entity Definition

Create Image Automatically

```
type: brooklyn.entity.proxy.haproxy.HAProxyController
brooklyn.config:
  docker.image.name: haproxy
  docker.image.tag: 1.5.9
  install.dir: /usr/local/sbin/
  run.dir: /usr/local/etc/haproxy/
```

Placement Strategies and Headroom

Random, Depth or Breadth First
CPU or Memory Usage
Memory, CPU or Container Limits
Geographic Constraints

```
docker.container.strategies:  
- $brooklyn:object:  
  type: brooklyn.location.docker.strategy.MaxContainersPlacementStrategy  
  brooklyn.config:  
    maxContainers: 12  
- $brooklyn:object:  
  type: brooklyn.location.docker.strategy.BreadthFirstPlacementStrategy
```

Deep dive: Networking

APACHE CON
NORTH AMERICA

Pluggable Providers

Weave, Metaswitch Calico

Multiple Networks provisioning

Single Application or Shared, Private Addresses,
Segmented by CIDR

```
- type: brooklyn.networking.VirtualNetwork
  networkId: my-application
  cidr: 192.168.12.0/24
  gateway: 192.168.12.1
  dnsServers:
  - 8.8.8.8
  brooklyn.config:
    sdn.example.securityGroup: "my-security-group"
```

Summary

APACHE CON
NORTH AMERICA

Clocker.io

Brooklyn + jclouds + Docker + Weave / Calico

solves

Composite Application Management

Docker Cloud Networking

Container Placement and Provisioning

@turlinux

Where to find it

Releases

0.8.0 Developer Preview <http://git.io/jGhs>

0.8.0 final - to be announced at Docker Meetup

Where to find us

@clockercentral

#brooklyncentral on IRC freenode



Questions?