Using Apache Brooklyn and Docker to Simulate your Production Environments in the Cloud

Andrew Kennedy
grkvt@apache.org
Simulating Production

Using Apache Brooklyn and Clocker to Simulate Production Environments in the Cloud

ApacheCon Austin, TX; April 2015
Andrew Kennedy @grkvlt
Introduction

• Andrew Kennedy
  – Clocker Project Founder and Lead Engineer
  – Open Source and Distributed Systems
  – Apache Committer for Brooklyn and Qpid
  – github.com/grkvlt

• Cloudsoft Corporation
  – Open Source Application Management Specialists

@grkvlt
Agenda

1. Clocker Introduction
2. What is a Docker Cloud?
3. Demonstration
4. Clocker Applications
5. Simulating Production?
6. Questions

@grkvlt
Clocker Introduction
What does it do?

1. Spins up and Manages Docker Clouds
2. Serves up Containers on Demand
3. Manages Composite Application Deployments on Docker

@grkvlt
What does it provide?

• Multi Host and Multi Container Applications
• Seamless Networking
  – Communication Between Services
• Orchestration and Clustering
  – Control of Containers
  – Container Management

@grkvlt
Who is using it?

• Testing and Proof of Concept Stage
  – Financial Services
  – Insurance
• Production
  – Multi-tenant Application Trial
  – Container per service
  – Ideally suited to the Clocker model

@grkvlt
Where can I find it?

- Open Source on GitHub
  - Apache 2.0 Licensed
  - [http://clocker.io](http://clocker.io)
- Status
  - **0.8.0 Developer Preview** available now
  - **0.8.0 Release** at Docker Meetup this week!

@grkvlt
What is a Docker Cloud?
Docker Cloud

1. On-demand
2. Multi-Tenant
3. Hardware Independent
4. Application Driven
Clocker and Brooklyn

• What is it?
  – Brooklyn Application and Location
  – Uses jclouds for Docker access

• What does it provide?
  – First Class Docker Support in Brooklyn
  – Optimized Brooklyn Blueprints for Docker

@grkvlt
Apache Brooklyn

• Application Management Platform
• Deploy, Manage and Monitor Blueprints
• Provisioning, Installation and Customization
• Management
  – AutoScaling, Resilience, Performance, Security
Apache jclouds

• Java Cloud Library
• API Agnostic
• Create Virtual Machines
• Docker Driver by @turlinux
• Virtual Container
Docker

- Popular
- Containers
  - Isolation
  - Performance
  - Composable
  - Complex
  - The Future...

@grkvlt
Software-defined Networking

• Pluggable providers
  • Weave
  • Project Calico
    • New in 0.8.0
  • DOVE
• Write your own!

@grkvlt
Clocker Orchestration

- Clocker
- Cloud
- Virtual Machine
- Docker Engine
- Container
- SDN
- Network Segment
Demonstration
Docker Cloud Infrastructure

Status: STARTING
Service Up: false
URL: /clocker
Type: brooklyn.entity.basic.BasicApplication
ID: ezq2RlyY

Blueprint

Config
Features

• Orchestrated Docker 1.5.0 deployment with SDN integration
• Automated attachment of containers to multiple dynamic networks
• Brooklyn application blueprints with network topology
• Docker images as Brooklyn entity source
Clocker Applications
Clocker Features

• Application Deployment
  – Oasis CAMP YAML Blueprint
  – TOSCA in Development
  – Docker Compose
  – Core Brooklyn

• Mixed Destinations
  – Some Virtual Machines
  – Some Bare Metal
  – Some Containers

@grkvlt
Clocker Features

• Docker Extensions to Brooklyn
  – Dockerfile or Image Specification for Installation
  – Placement Strategies for Containers
  – Create Docker Images and Networks

• Manages Docker Engine
  – Deployment and Management
  – Installation and Configuration
  – Software-Defined Networking
Brooklyn Blueprints

• Describe Applications
• OASIS CAMP Standard
• List of Services
• Tree Structure
• Sensors, Effectors and Policies

@grkvlt
Blueprint Example

```json
name: appserver-w-policy
services:
  - type: brooklyn.entity.webapp.ControlledDynamicWebAppCluster
    initialSize: 1
    memberSpec:
      $brooklyn:entitySpec:
        type: brooklyn.entity.webapp.jboss.JBoss7Server
        brooklyn.config:
          wars.root:
            http://search.maven.org/remotecontent?filepath=io/brooklyn/example/brooklyn-example-hello-world-sql-webapp/0.6.0/brooklyn-example-hello-world-sql-webapp-0.6.0.war
          http.port: 8080+
          java.sysprops:
            brooklyn.example.db.url: $brooklyn:formatString("jdbc:%s%s?user=%s\&password=%s", component("db").attributeWhenReady("datastore.url"), "visitors", "brooklyn", "br00k11n")
    brooklyn.policies:
      - policyType: brooklyn.policy.autoscaling.AutoScalerPolicy
        brooklyn.config:
          metric: $brooklyn:sensor("brooklyn.entity.webapp.DynamicWebAppCluster", "webapp.reqs.perSec.windowed.perNode")
          metricLowerBound: 10
          metricUpperBound: 100
          minPoolSize: 1
          maxPoolSize: 5
  - type: brooklyn.entity.database.mysql.MySqlNode
    id: db
    name: DB HelloWorld Visitors
    brooklyn.config:
      datastore.creation.script.url:
```
Application Components

• Services
  • Catalog Entries
  • Defined by Brooklyn Code
• Policies
• Sensors
• Enrichers

@grkvlt
Services

• Brooklyn Entities
  • Installed by running SSH commands
  • Add packages or extract archive files
  • Run arbitrary commands

• Clocker commits image after installation

@grkvlt
Container Definition

• Sources
  – Brooklyn Entity Definition
  – Chef Recipe
  – Docker Image Definition
  – Dockerfile

• Create Image Automatically
  – Commit or Push for Reuse

@grkvlt
Container Definition

id: docker-haproxy
name: "Docker Hub HAProxy Load Balancer"
origin: "https://registry.hub.docker.com/_/haproxy/
locations:
  - my-docker-cloud
services:
  - type: brooklyn.entity.proxy.haproxy.HAProxyController
    id: haproxy
    brooklyn.config:
      docker.image.name: haproxy
      docker.image.tag: 1.5.9
      install.dir: /usr/local/sbin/
      run.dir: /usr/local/etc/haproxy/
      network.list:
      - dmz
Container Definition

id: dockerfile-mysql
name: "Docker Hub MySQL Database"
origin: "https://registry.hub.docker.com/_/mysql/"
locations:
  - my-docker-cloud
services:
  - type: brooklyn.entity.container.docker.application.DockerfileApplication
    id: mysql
    name: "MySQL"
    brooklyn.config:
      docker.dockerfile.url:
        "https://s3-eu-west-1.amazonaws.com/brooklyn-clocker/mysql-5.6.tgz"
      docker.container.environment:
        MYSQL_ROOT_PASSWORD: "s3cr3t"
Container Definition

id: dockerfile-mysql
name: "Docker Hub LAMP Stack"
locations:
  - my-docker-cloud
services:
  - type: docker:mysql:5.7.5
    id: mysql
    env:
      MYSQL_ROOT_PASSWORD: "s3cr3t"
  - type: docker:grkvlt/myapp:latest
    id: application
    env:
      MYSQL_HOST:
        $brooklyn:component("mysql").attributeWhenReady("host.hostname")
Container Placement

• Where do we want the service to run?
• Supply and Demand
  – Here's the locations you can use...
  – I want a very specific location...
• Docker Swarm
  – Possible future integration point...
Container Placement

• Demand
  – Adding an Application
  – Scaling existing Application

• Requirements
  – Host Location
  – Service Resources
  – CPU, Memory

@grkvlt
Container Placement

• Supply
  – Choose a Host from available
  – Create new Host if required

• Start Container there
  – Set CPU and Memory
  – Attach to Network
Container Placement

• Placement Strategies
  – Random, Depth or Breadth First
  – CPU or Memory Usage
  – Memory, CPU or Container Limits
  – Geographic Constraints

• User Defined
  – Java Predicate
Placement Strategy

• Deterministic
• Simple

  – Predicate and Comparator

```python
docker.container.strategies:
  - $brooklyn:object:
    type: "brooklyn.location.docker.strategy.MaxContainersPlacementStrategy"
    brooklyn.config:
      maxContainers: 16
  - $brooklyn:object:
    type: "brooklyn.location.docker.strategy.CpuUsagePlacementStrategy"
    brooklyn.config:
      maxCpu: 0.75
```

@grkvlt
Autonomics

• Brooklyn Policies
• Attached to Entities in Application
  – Nothing Docker Specific
• Elastic Scaling
  – Cluster Resizing
  – Sensor Driven

@grkvlt
Application Resilience

• Service Resilience and Replacement
  – Restart Service and Container
  – Application Level, Not Infrastructure
  – Same as Cloud

• Snapshot Running Container for Restart

@grkvlt
Headroom

- Ensure resources available
- Based on MaxContainers strategy limit
  - Or Percentage Utilization
  - Or CPU and RAM allocation
- Scale Docker Host Cluster Automatically
  - Add new Docker hosts
  - Remove empty Docker hosts

@grkvlt
Software-Defined Networking

• Needed for Seamless Provisioning
• Host to Host Communication
  – Same LAN Segment
  – No Port Forwarding
  – Natural Application Configuration
• Initial Driver was EPMD Applications

@grkvlt
Networking Providers

• Implementation Agnostic
  – L2 over L3 etc.
  – Similar to Hypervisor in Clouds

• Generic Interfaces
  – Host Component
  – Service Component (or Endpoint)
Clocker Networking

Internet \(\rightarrow\) SDN Gateway \(\rightarrow\) Host \(\rightarrow\) Container \(\rightarrow\) Container

Internet \(\rightarrow\) SDN Gateway \(\rightarrow\) SDN Bridge

@grkvlt
Networking Capabilities

• Provide Multiple Networks
  – Single Application or Shared
  – Private Addresses
  – Segmented by CIDR

• Docker Port Forwarding Access
  – Debug Mechanism
Simulating Production?
Application Development Cycle

1. Development
2. Continuous Integration
3. UAT or Testing
4. Staging
5. Production
Dev Cycle Reality

- My Laptop
- Jenkins Server
- Bob's Laptop
- Some spare VMs we found...
- The best we can afford
  - until next year's budget...?

@grkvlt
Application Development

• Same Application
• Different Infrastructures
  – Very Different
• So ends up...
  – Different Application
Application Development

• Which means
  – We aren't testing the right things
  – Production is probably broken under load or scale
  – Ops are unhappy ;(
Different Application

• Very Different!
  – No resilient pairs
  – No failover
  – No load balancer
  – No Clustering
  – Single network, namespace, domain, etc.

@grkvlt
How can we fix this?

• Perfect worlds
• All environments identical
• Staging *is* an *exact* copy of production
  – Ready for App and Infra cut-over
• UAT *is* Staging, with anonymized data
• And so on, rolling through environments

@grkvlt
How can we fix this?

- CI becomes CD
  - Dedicated production-level environment for builds
  - Successful builds promoted to UAT
- Test and Dev...?
  - You have to make some sacrifices ;)
  - But will try and test the HA mechanism and so on in isolation
  - But at least Ops are happy
How can we fix this?

- Imperfect world
- Or, the DevOps way
  - We don't have enough money for six copies of our architecture
  - Particularly at scale or with large data sets
  - So we fake it!
- The *important* thing is our architecture
  - It defines the application completely
  - We describe this *once* in a blueprint
  - And then deploy to our various environments

@grkvlt
Clocker and DevOps

- Application blueprint
- Describes
  - Services
    - Configuration
  - Policies
  - Networks
  - Hierarchy
    - Connections
- Can be deployed to any Brooklyn Location
Clocker and DevOps

• Locations include
  – Vagrant or other VMs on my laptop
  – Apache jclouds supported providers
    • On premise OpenStack CI cloud
    • Public SoftLayer environment
    • ... choose your favourite
  – Docker Clouds using Clocker
    • Containers instead of VMs
    • Automatically
    • No input from developer required
There's More Than One Way...

• Could use Clocker and Docker everywhere
  – Generate Docker image during build process
  – Size underlying VMs appropriately
  – Allocate different CPU/memory to containers
  – Deploy images to Clocker everywhere

• Many enterprises not yet ready for this...

@grkvlt
Thank You!
Questions?
Web Resources

http://clocker.io/
http://brooklyn.io/
http://docker.io/
http://weave.works/
http://projectcalico.org/
http://abstractvisitorpattern.co.uk/

@grkvlt