Chef and Docker

CMPE 283 Spring 2015

Team 1:

Neeraja Kukday

Anusha Ravikumar

Harshad Kulkarni

Mahesh Dhamgunde

Project Group Members Contribution

Docker Demo	Harshad, Mahesh
Chef Demo	Neeraja, Anusha
Docker & Chef Research	Neeraja, Anusha, Harshad, Mahesh

Presentation Outline

* Business Problem
* Solution to Business Problem
* Docker
* Chef
* Demo Chef
* Demo Docker

Business Problems

- Scalability
- Service Isolation
- Configuration Management

Docker

- Similar to Linux Containers
- Solomon Hykes - Standard shipping containers that can be used to ship any product
- DevOps Engineers - Need to only know how much to ship and where to ship
- Use case: Ebay

CHEF

- Used for configuration management
- Write scripts for configuration, updation, deployment on the server
- Used for continuous integration. Automatically deploying new builds

Nightmares working on local dev env

- Too many manual tasks required in setup
- Performance issue/machine is too slow
- Inconsistency with production environment

How To solve?

- Environment on Demand
- Automated Configuration Management
- Faster Deployment cycle
- Better performance

Introduction to Docker

- An open source software used to perform various operations with Linux containers.
- Helps in creating environment on demand.
- Containers are lightweight(Typical laptops can run around 10-100 containers)
- Helps to isolate the processes.
- Does not require a separate OS, uses Kernel's functionality and uses separate namespaces.

Why Docker

- Rapid Application Deployment
- Portability across machines
- Version control and component reuse
- Sharing images with help of Docker Registry
- Simplified maintenance

Containers Vs VM

- **You can create or destroy containers quickly and easily.**
- **Containers are lightweight, therefore, more containers than virtual machines can run simultaneously on a host machine.**
- **Containers share resources efficiently. Virtual machines are isolated.**
- **Disadvantage: Virtual machines can be migrated while still executing, however containers cannot be migrated while executing and must be stopped before moving from host machine to host machine.**
- **Containers do not replace virtual machines for all use cases. Careful evaluation is still required to determine what is best for your application.**

Dockerfile Example: MongoDB

```
# Dockerizing MongoDB: Dockerfile for building MongoDB images
# Based on ubuntu:latest, installs MongoDB following the instructions from:
# http://docs.mongodb.org/manual/tutorial/install-mongodb-on-ubuntu/

FROM       ubuntu:latest
MAINTAINER 283Proj <mahesh.dhamgunde@sjsu.edu>

# Installation:
# Import MongoDB public GPG key AND create a MongoDB list file
RUN apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
RUN echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | tee /etc/apt/sources.list.d/10gen.list

# Update apt-get sources AND install MongoDB
RUN apt-get update && apt-get install -y mongodb-org

# Create the MongoDB data directory
RUN mkdir -p /data/db

# Expose port #27017 from the container to the host
EXPOSE 27017

# Set /usr/bin/mongod as the dockerized entry-point application
ENTRYPOINT ["/usr/bin/mongod"]
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                                          1,1           All
```

Dockerfile Example: Node.js

```
# Dockerizing Nodejs: Dockerfile for building Nodejs images with Express framework app
# Base image is phusion/passenger-nodejs
FROM phusion/passenger-nodejs:0.9.15

MAINTAINER Mahesh Dhamgunde <mail@mahesh.me>

# Set correct environment variables.
ENV HOME /root

# Use baseimage-docker's init system.
CMD ["/sbin/my_init"]

# Equivalent to: export DEBIAN_FRONTEND="noninteractive" which means no questions asked on apt-get install
ENV DEBIAN_FRONTEND noninteractive

# Bundle app source
COPY . /docker
# Install app dependencies
RUN cd /docker/DockerApp; npm install

EXPOSE  3000
CMD ["node", "/docker/DockerApp/app.js"]

# Clean up APT when done.
RUN apt-get clean && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                                    1,1           All
```

Docker Demo

- Docker Images
- Dockerfile and building and running Docker
- Isolation of services in Docker containers : MongoDB and Node.js

Introduction to Chef

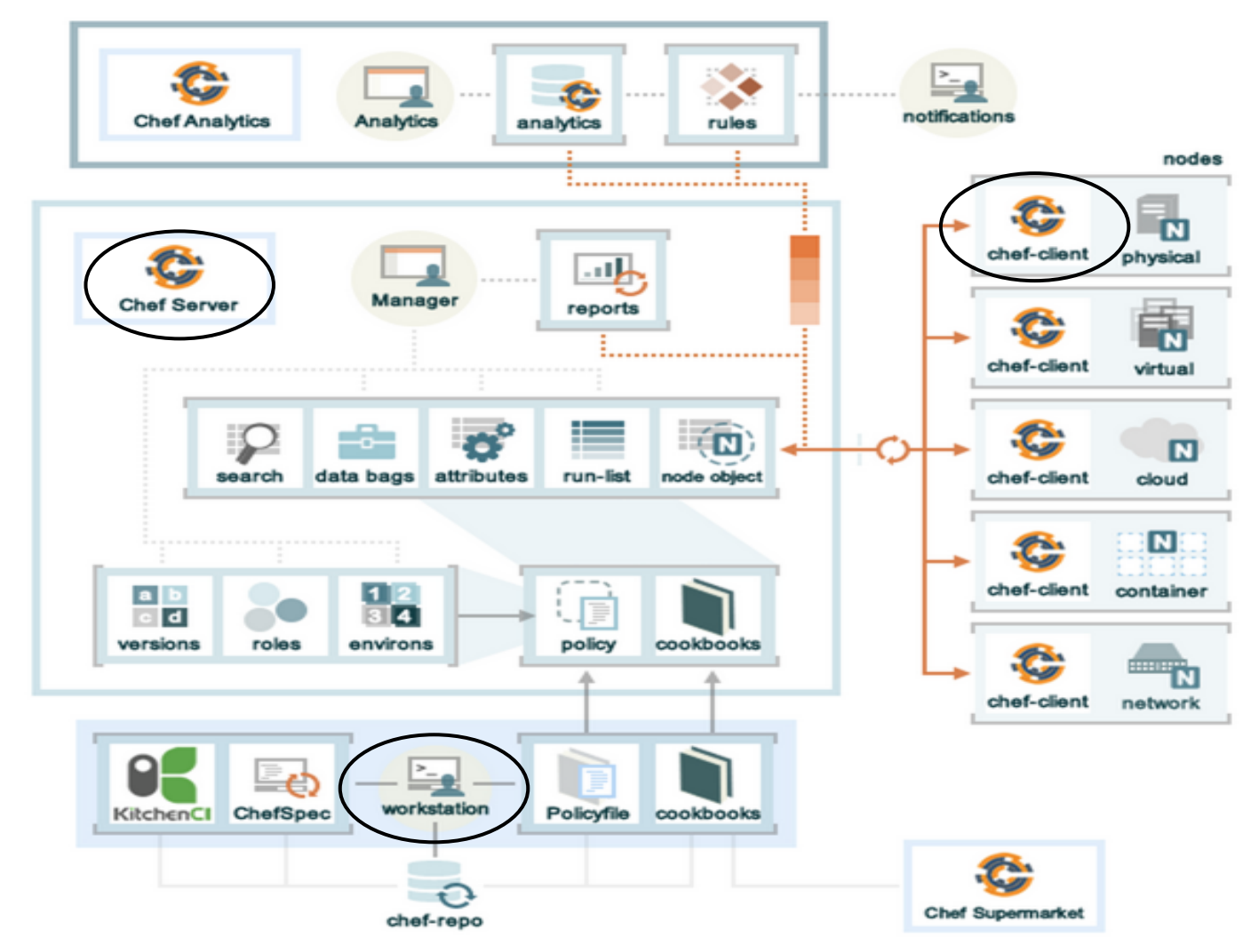
* Configuration Management tool (ruby)
* Environment management
* Turns infrastructure into code-->easy to deploy servers and applications to any physical, virtual, or cloud location
* Automate: configuration, deployment, and manage infrastructure.

Chef Terminologies

* **Resource:** piece of infrastructure and its desired state
* **Recipe**: collection of resources with desired configuration
* **Cookbooks**: collection of recipes; defining a scenario

* **Chef-client**: installed on every node that is under management by Chef. Executes recipes pulled from chef server.
* **Chef Server**: all infrastructure information, manages nodes
* **Workstation**: local dev env.; 1+ workstations are configured to allow users to create, modify, and maintain cookbooks.
  * Knife command line tool to interact with the server.
  * Cookbooks are uploaded to the Chef server from the workstation.

Components of Chef



Why Use Chef?

* Improves Efficiency:
  * Reduces manual tasks required in setup
  * Reduces inconsistency
    * Idempotent
  * Reduces documentation
  * Speed and automation

"Chef provided an automation solution flexible enough to bend to our scale dynamics without requiring us to change our workflow."

Phil Dibowitz Production Engineer, Facebook

facebook

"Chef's biggest advantage is the amount of time we save in setting up virtual servers and other tasks."

Leandro Reox, Senior Infrastructure Engineer and Cloud Architect

mercado Libre

Disadvantages of Chef

* Ties users to Ruby
* Doesn’t support push:
  * systems must check with the Chef server for configuration changes to take effect.
  * immediate implementation of changes is not possible
* Required dependencies have incompatible versions.

Chef Demo

1. Anusha- Chef basic: resources and recipes
2. Neeraja - Chef and docker

Thank you!

References

* https://www.chef.io/chef/
* http://docs.chef.io/chef_overview.html
* http://www.slideshare.net/sanjeev-sharma/chef-for-dev-ops-an-introduction
* http://www.slideshare.net/dstine4/jenkins-and-chef-infrastructure-ci-and-automated-deployment
* http://www.scriptrock.com/articles/salt-vs-chef