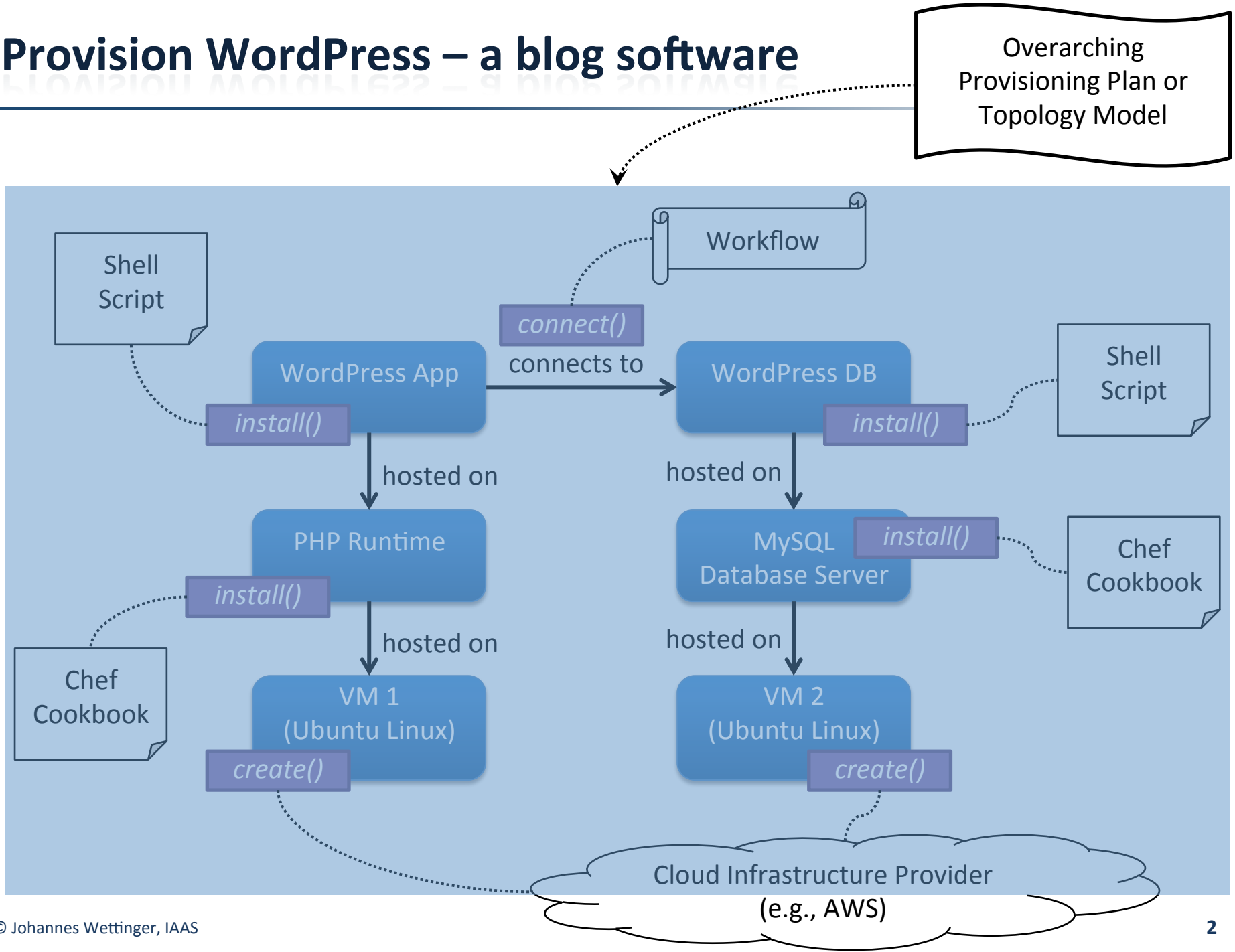
Provisioning Techniques
for Cloud Applications

University of Stuttgart
Universitätsstr. 38
70569 Stuttgart
Germany

Johannes Wettinger
Institute of Architecture of Application Systems
wettinger@iaas.uni-stuttgart.de

Phone +49 711 685 88 460
Fax +49 711 685 88 472

Provision WordPress – a blog software



Why Provisioning Automation?

DevOps

Architects & **Dev**elopers

App Arch.

App Design

App Code

Fast and frequent releases
require
Efficient and tight collaboration
requires
holistic automation!

Plethora of provisioning automation possibilities

→ **Explosion** of tools, artifacts, techniques, providers, etc. … and **combinations!**

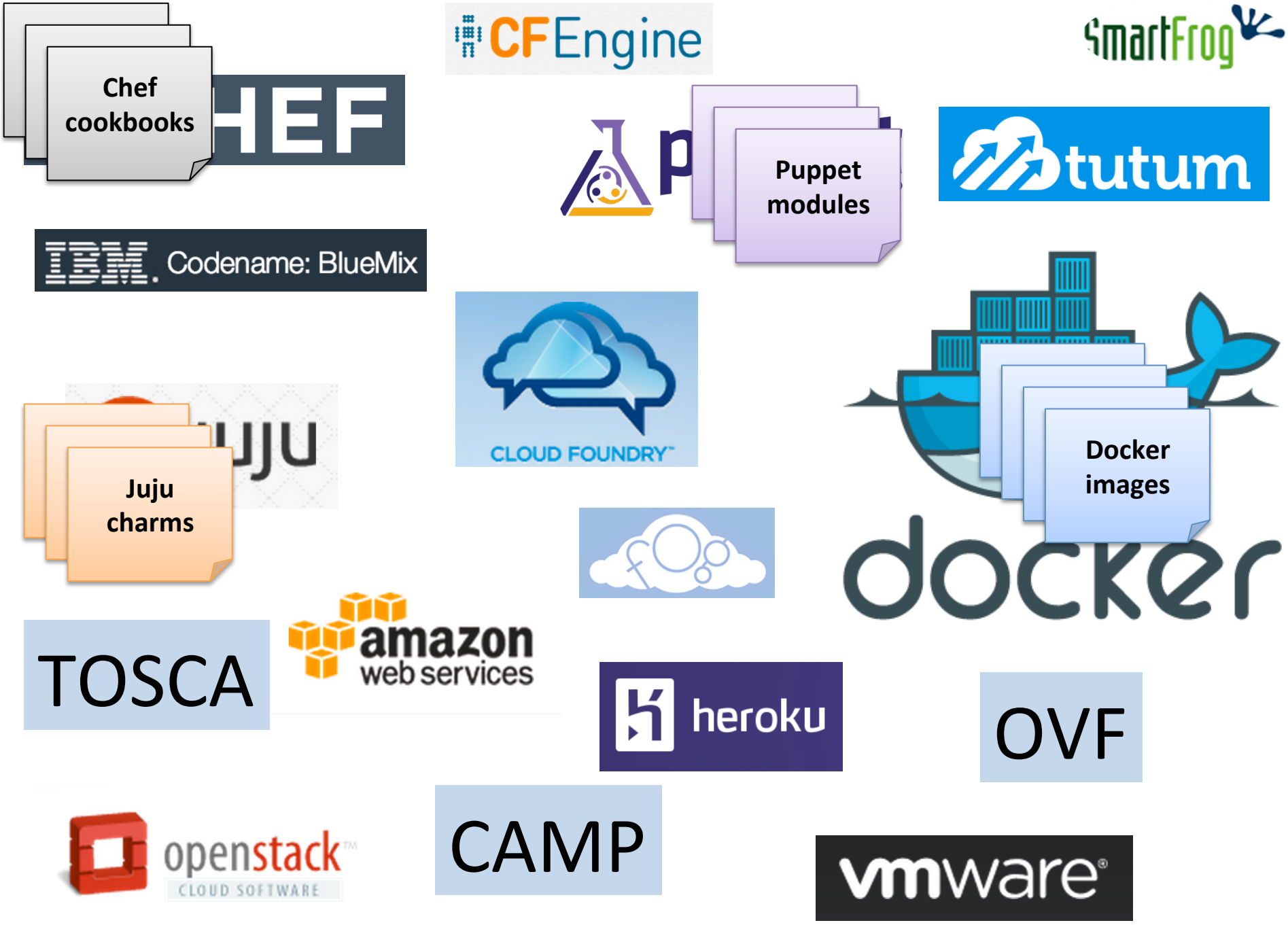
Ops Personnel / Cloud Service Creator

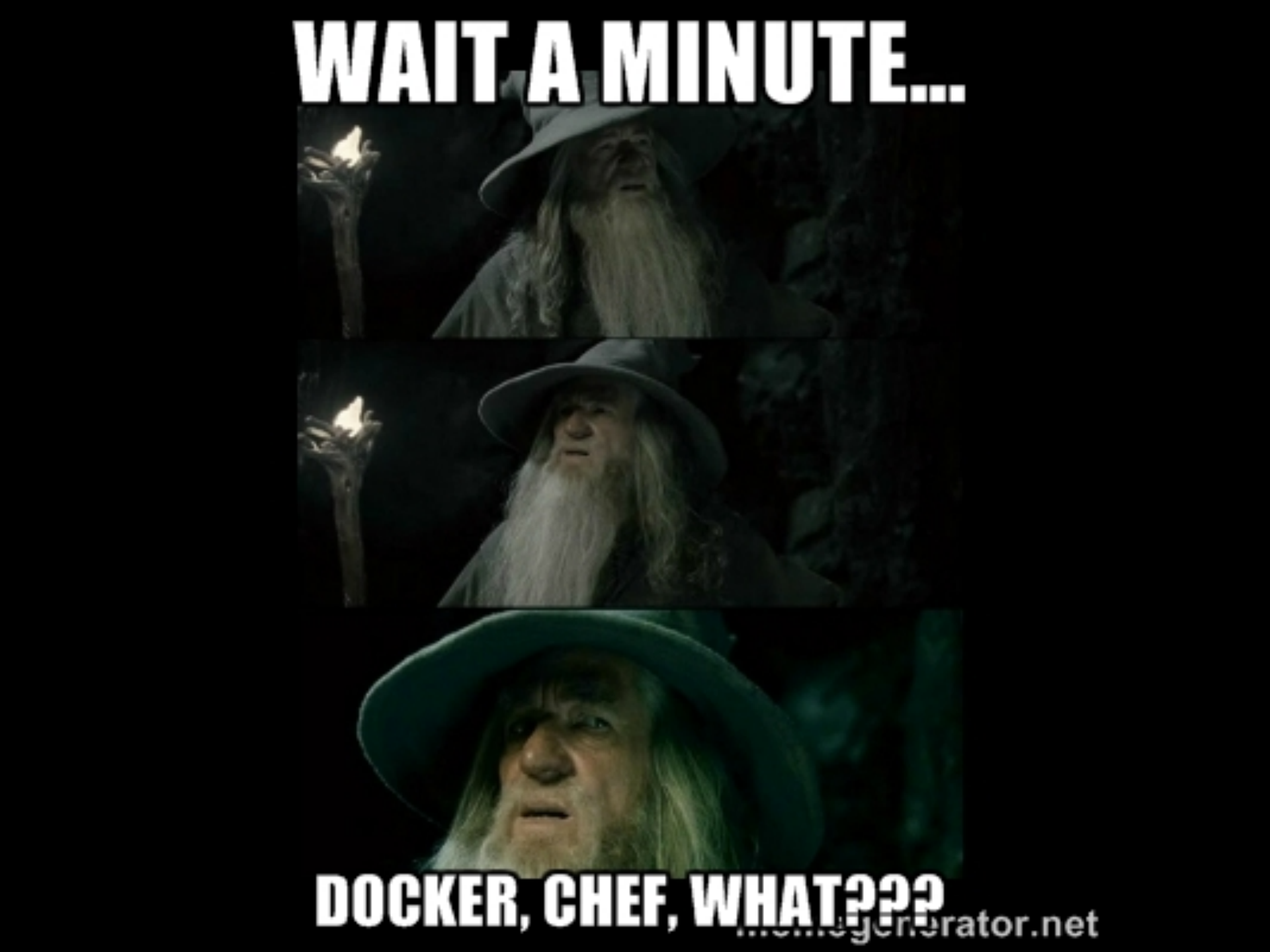
Deployable App Model

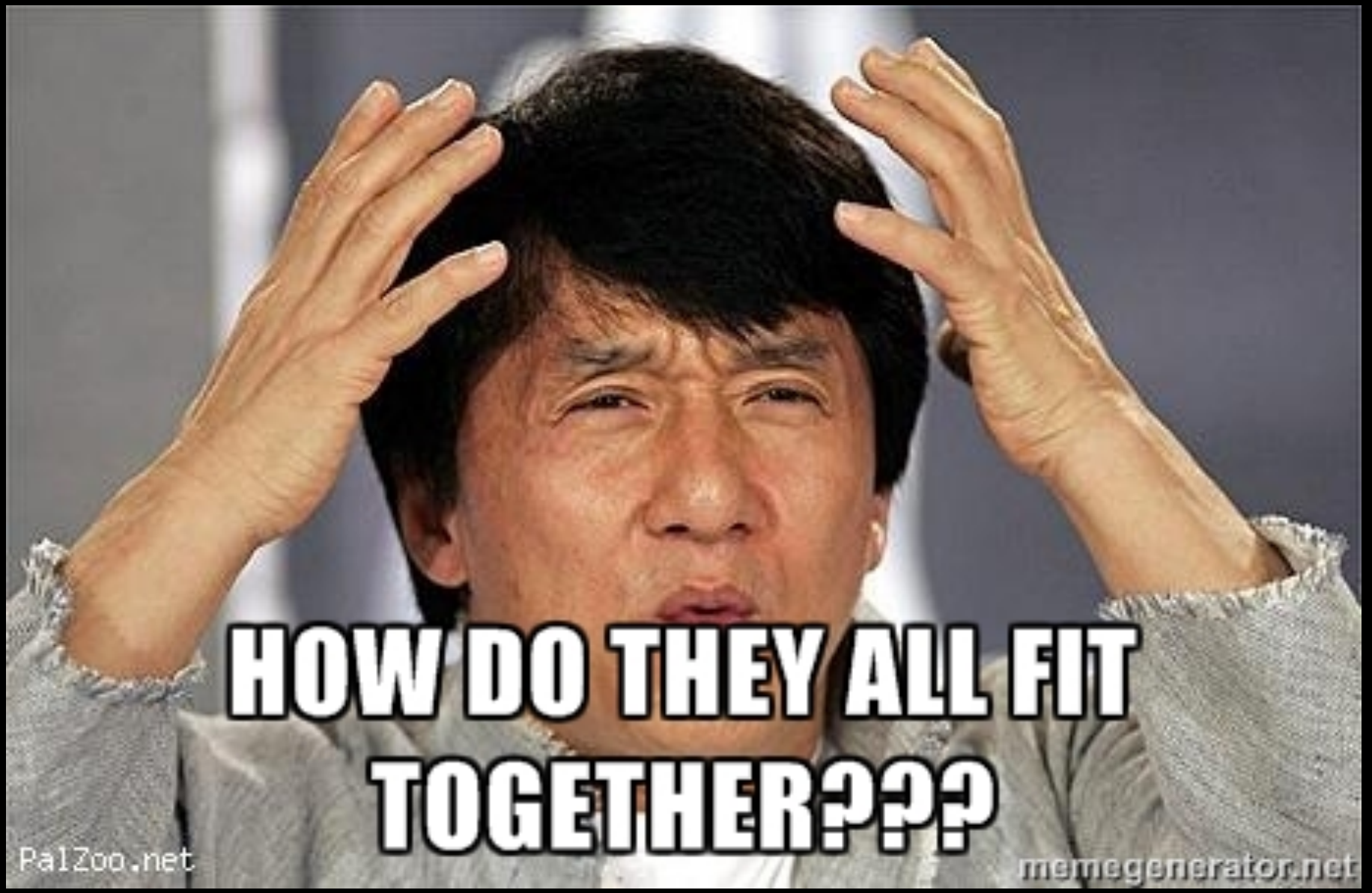
2

App Instances

Dev., Test, Prod., …


Chef cookbooks
CHEF
CFEngine
SmartFrog
Puppet modules
tutum
IBM Codename: BlueMix
CLOUD FOUNDRY
Juju charms
Docker images
docker
TOSCA
amazon web services
heroku
OVF
openstack
CLOUD SOFTWARE
CAMP
vmware


WAIT A MINUTE...
DOCKER, CHEF, WHAT???
memegenerator.net


HOW DO THEY ALL FIT TOGETHER???
PalZoo.net
memegenerator.net

What we need:

Systematic classification of provisioning automation approaches

Enables decision support

… leading to informed decisions ☺

One Dimension: Level of Dependencies

Provider-dependent:

Approach depends on a particular **provider** in terms of service offerings, APIs, etc.
It may further depend on **provider-specific tooling** such as SDKs, CLIs, etc.

Tooling-dependent:

Approach depends on **certain tools,** but may be used in conjunction with different providers.

Standards-based:

Approach is based on standards and thus does not depend on particular tools or providers.

amazon web services

docker

TOSCA

heroku

CLOUD FOUNDRY™

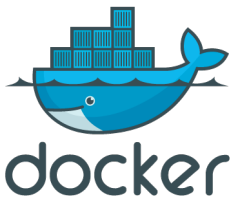
OVF

+ Level of Dependencies -

Another Dimension: Level of Virtualization

Middleware Virtualization:

PaaS: middleware provided as a service (DBaaS, Messaging-as-a-Service, …)



OS Virtualization:

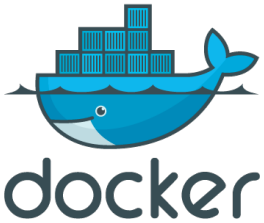
Containers: process isolation, filesystem isolation, …

OVF

Hardware Virtualization:

Hypervisor/VMM (para- vs. full virtualization)

Yet Another Dimension: Code vs. Image



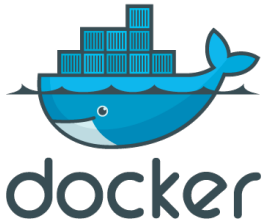
OVF

TOSCA

And Yet Another Dimension: Infrastructure- vs. App-centric

TOSCA

App-centric

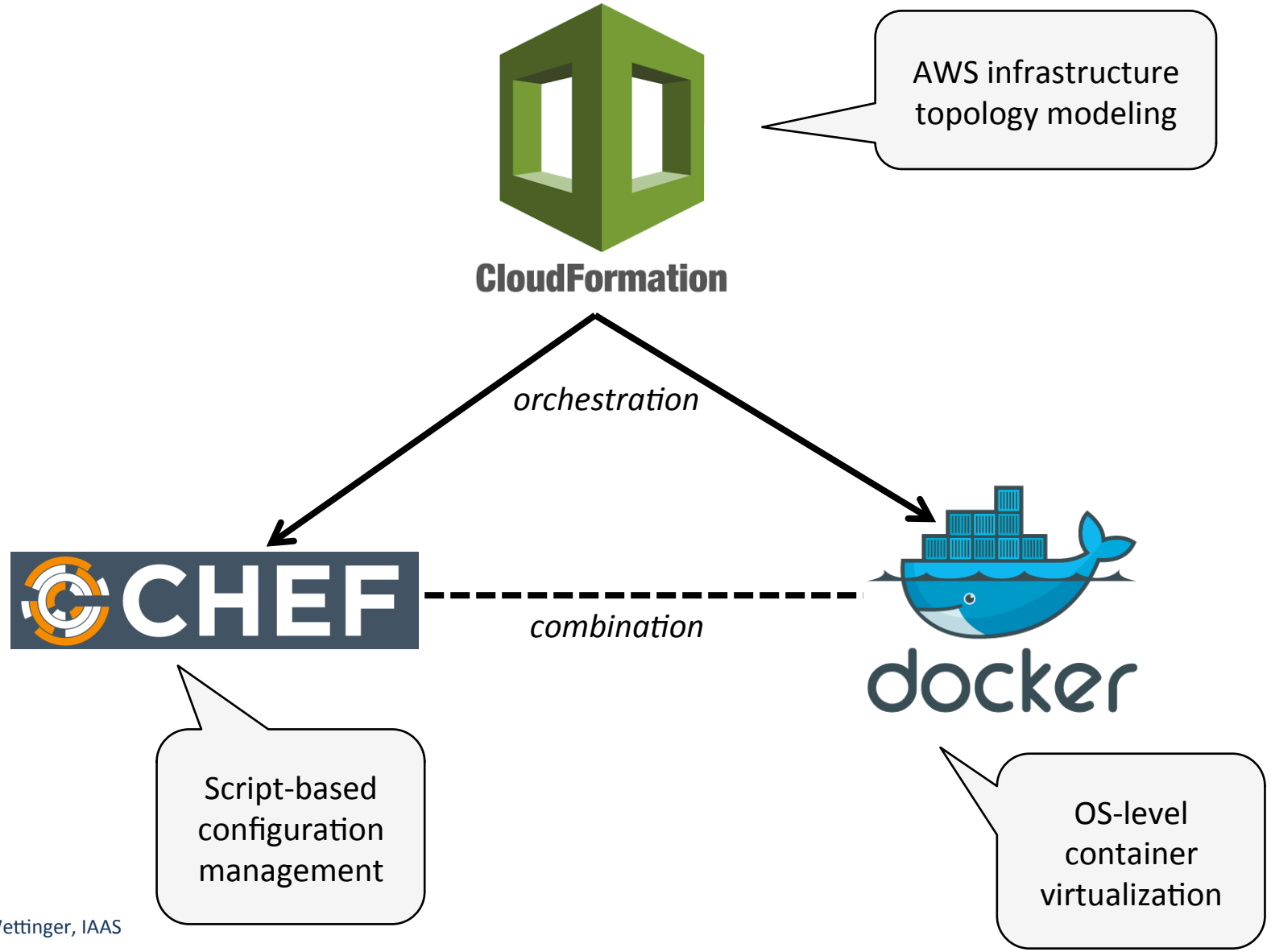


Infrastructure-centric

Further dimensions may consider …

- Infrastructure-centric vs. Application-centric
  - E.g., CloudFormation templates vs. Juju bundles
- Declarative vs. Imperative
  - E.g., CloudFormation templates vs. TOSCA plans
- …

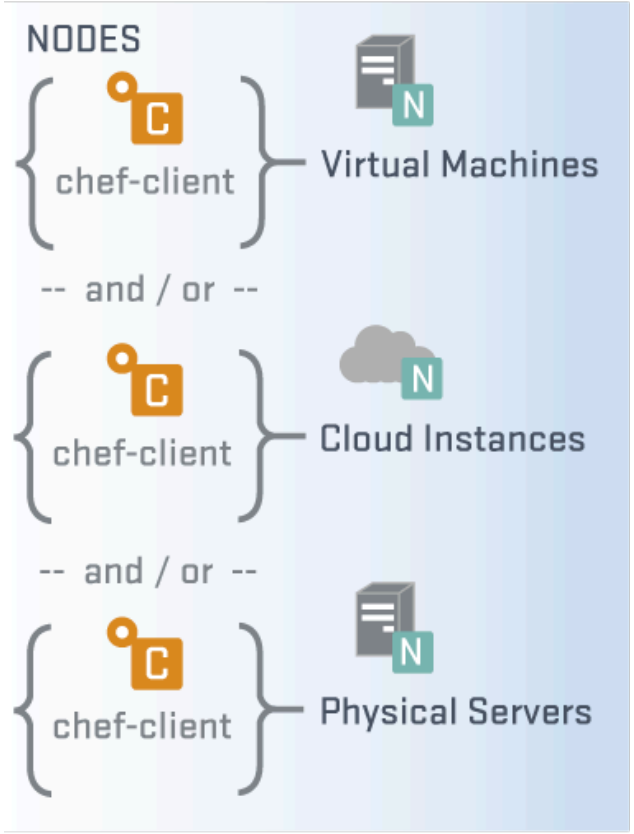
Deep Dives



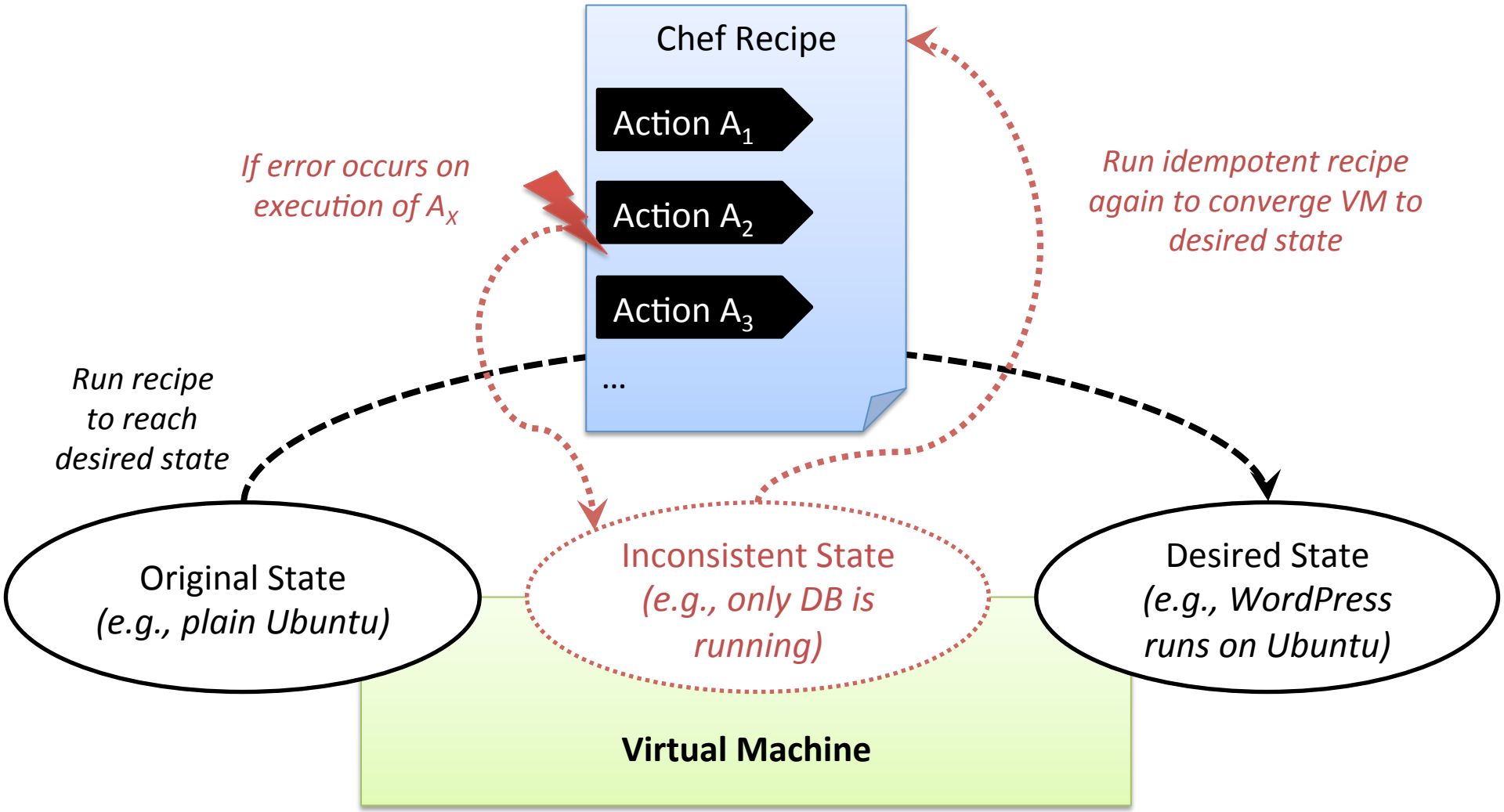

CHEF

Chef in a Nutshell

- Configuration of resources captured defined by **Recipes** using a Ruby DSL
  - Declarative: *ensure that MySQL server is installed*
  - Imperative: `apt-get install mysql-server`
- Recipes are bundled in **Cookbooks**
- Chef (server & client) is free and open-source!
- Hosted Chef (paid offering) = “Chef-as-a-service”


NODES
chef-client
Virtual Machines
-- and / or --
chef-client
Cloud Instances
-- and / or --
chef-client
Physical Servers

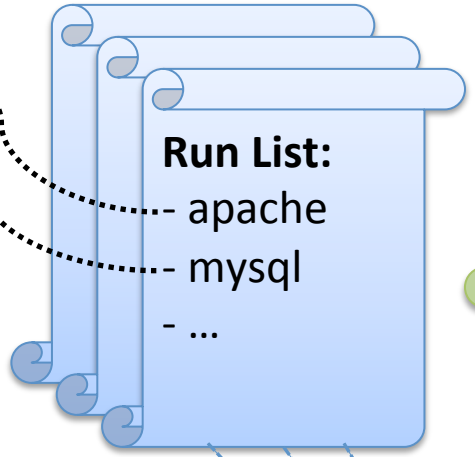
Convergent Provisioning Automation

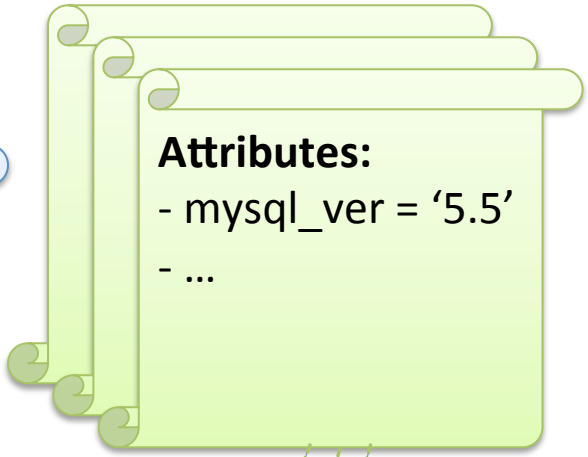


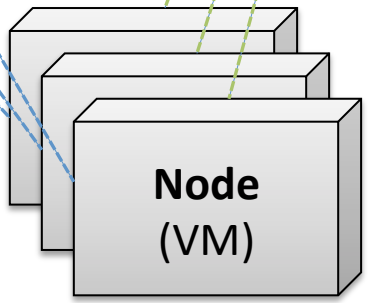
Chef Recipes, Nodes, Run Lists, and Attributes

Recipes:

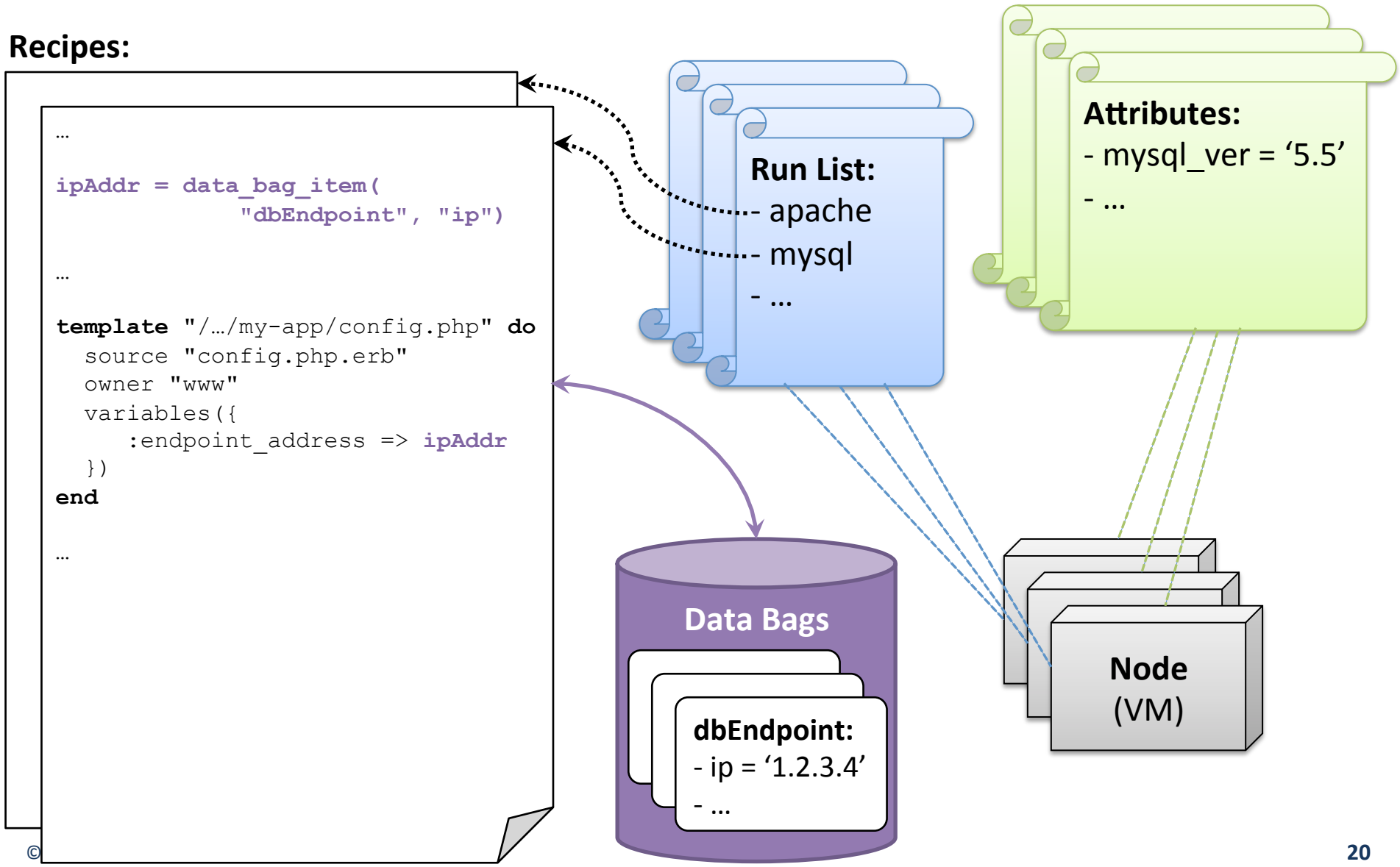
```
…

directory "/tmp/folder" do
  owner "root"
  group "root"
  mode 0755
  action :create
end

package "mysql" do
  version node["mysql_ver"]
end

http_request "delete_resource" do
  url "http://…/some_resource"
  action :delete
end

Bash "mysqltuner" do
  user "root"
  code <<-EOH
  apt-get install mysqltuner
  EOH
end

…
```







Chef Data Bags (to wire components etc.)



Community:
~ 1500
cookbooks

Databases

- mysql ★★★★☆ 24903 downloads
- postgresql ★★★☆☆ 9759 downloads
- database ★★★★★ 6839 downloads

See all

Web Servers

- apache2 ★★★★★ 21182 downloads
- nginx ★★★★☆ 16497 downloads
- passenger_apache2 ★★★☆☆ 3670 downloads

See all

Process Management

- runit ★★★★★ 8739 downloads
- monit_bin ★★★★★ 2499 downloads
- supervisor ☆☆☆☆☆ 1181 downloads

See all

Monitoring & Trending

- zabbix ★★★★★ 10575 downloads
- nagios ★★★★★ 8618 downloads
- newrelic ★★★★★ 7332 downloads

See all

Programming Languages

- java ★★★★★ 16593 downloads
- php ★★★★☆ 8317 downloads
- python ★★★☆☆ 6731 downloads

See all

Package Management

- apt ★★★☆☆ 16734 downloads
- yum ★★★★☆ 8469 downloads
- yumrepo ★★★★☆ 7108 downloads

See all

Applications

- application ★★★★☆ 13189 downloads
- nodejs ★★★★★ 7239 downloads
- rabbitmq ★★★★☆ 4952 downloads

See all

Networking

- ntp ★★★☆☆ 5232 downloads
- haproxy ★★★★☆ 4721 downloads
- arpwatch ☆☆☆☆☆ 4171 downloads

See all

Operating Systems & Virtualization

- windows ★★★★★ 8127 downloads
- aws ★★★★☆ 3967 downloads
- pxe_dust ★★★★★ 2717 downloads

See all

Utilities

- build-essential ★★★★★ 9028 downloads
- git ★★★★★ 7587 downloads
- chef-client ★★★★★ 7043 downloads

See all

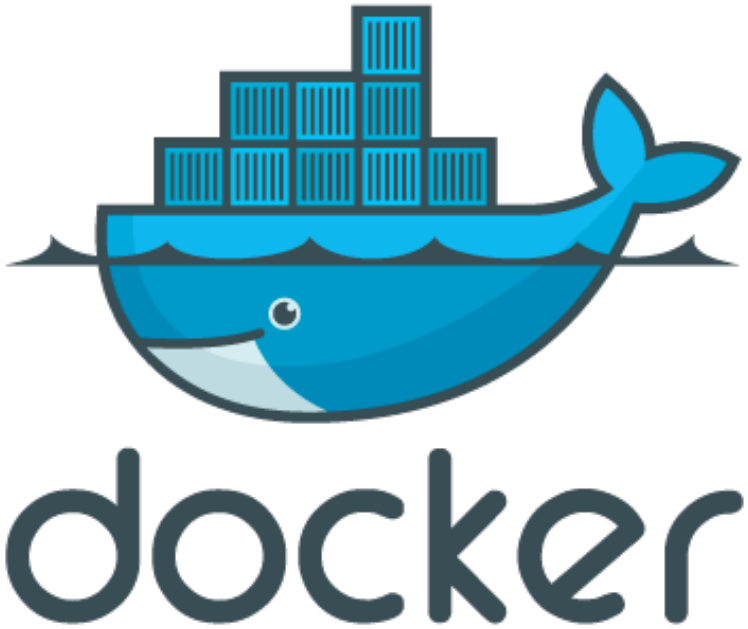
Other

- users ★★★★☆ 4390 downloads
- memcached ★★★☆☆ 4361 downloads
- getting-started ★★★★★ 2056 downloads

See all

Who's using Chef?

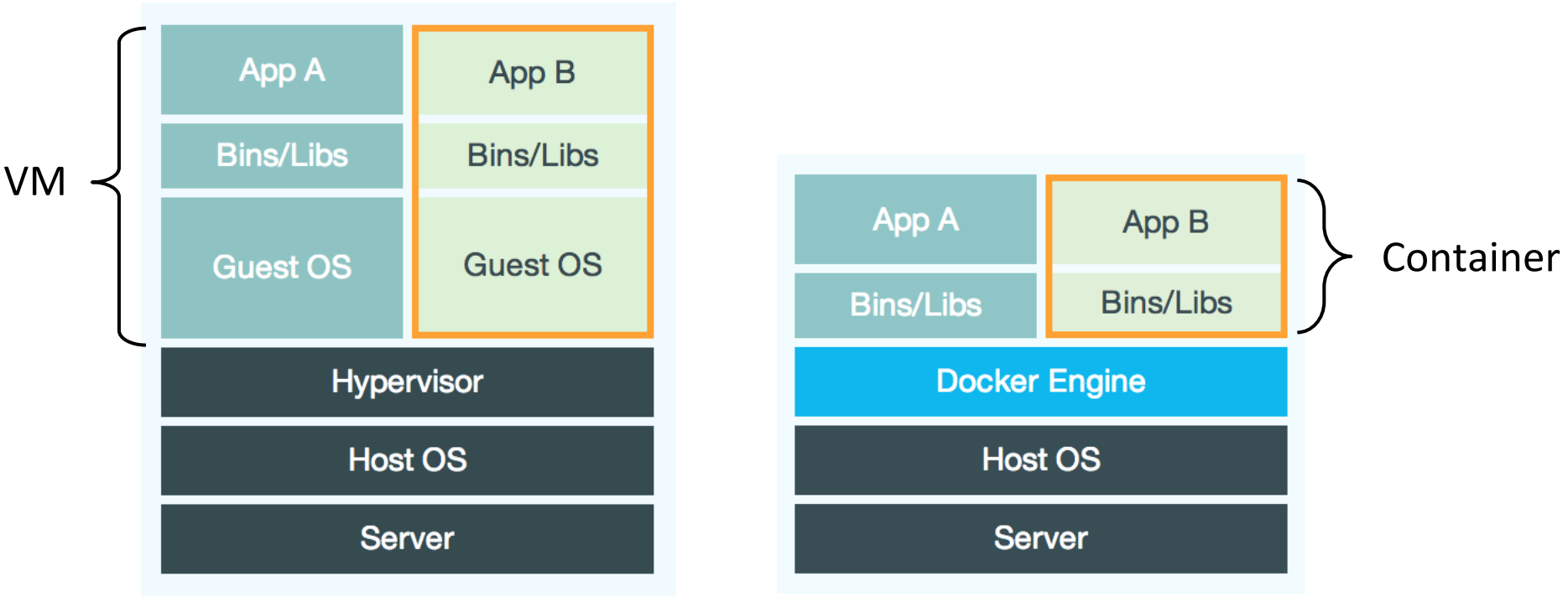
More: http://www.getchef.com/customers


docker

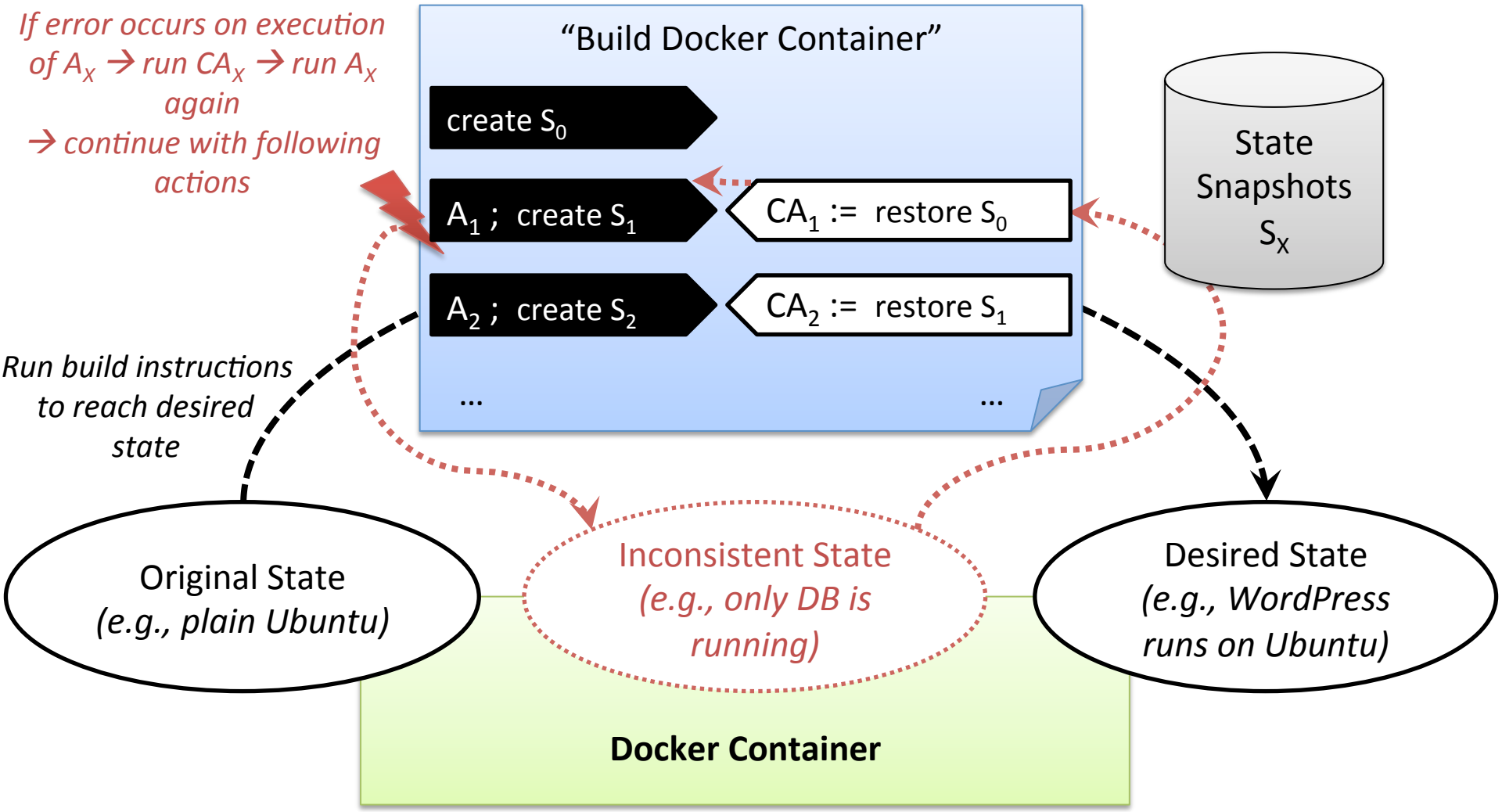
Docker in a Nutshell

- Virtualization on OS level
  - *No hypervisor, no separate guest OS*
- Host (VM or physical machine) runs:
  - Docker daemon (service with RESTful API)
  - Docker containers
    - *Isolation in terms of processes, filesystem, networking*
- **Dockerfile** = imperative build plan for containers
- Fine-grained container snapshots

Virtualization: VMs vs. Containers



Snapshot-based Provisioning Automation



Dockerfile to Deploy WordPress

```
FROM tutum/lamp:latest
MAINTAINER ...

# Download latest version of Wordpress into /app
RUN git clone https://github.com/WordPress/WordPress.git /app

# Configure WordPress to connect to local DB
ADD wp-config.php /app/wp-config.php

# Modify permissions to allow plugin upload
RUN chmod -R 777 /app/wp-content

# ...

EXPOSE 80
CMD /run.sh
```

Docker Container Links – via TCP/IP ports

Docker Container
MySQL

Exposed Port:
3306

Link alias = **DB**

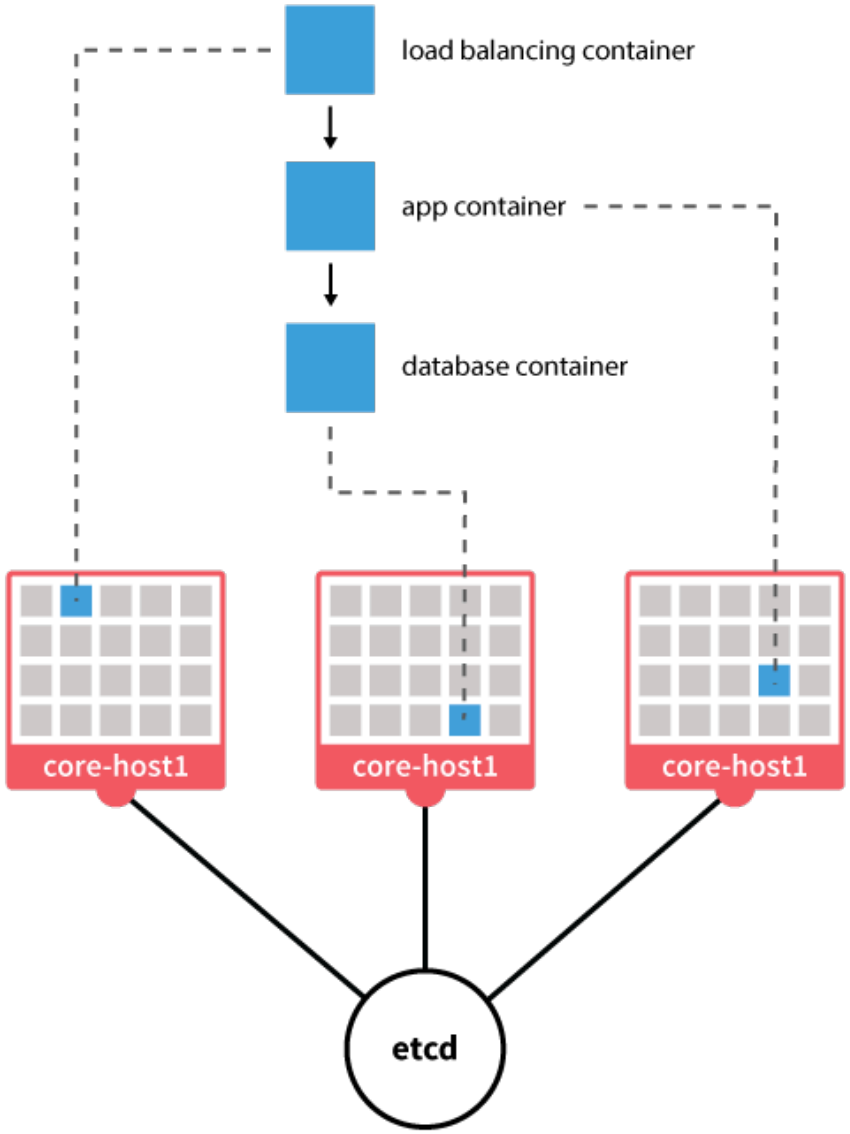
Docker Container
WordPress

`$DB_PORT_3306_TCP_PORT`
`$DB_PORT_3306_TCP_ADDR`

Exposed Port:
80

Docker Cluster Management at Scale

- CoreOS
- Kubernetes (Google)
- Decking
- Flynn
- Docker + OpenStack
- …



redis

ubuntu

The Official Ubuntu base image

WordPress

WordPress is a free and open source blogging tool and a content management system

MySQL

Popular open-source relational database management system

mongoDB

Document-oriented NoSQL database

CentOS

Official CentOS base image

NGINX

High performance reverse proxy server

Relational database management system

node.js

Node.js is a platform for scalable server-side and networking applications

Top Contributors

Community:
~ 14000
"Dockerized"
apps

clue 158

~Aachen, Germa

cpuguy83 153

Florida

Popular Repositories

ubuntu 244

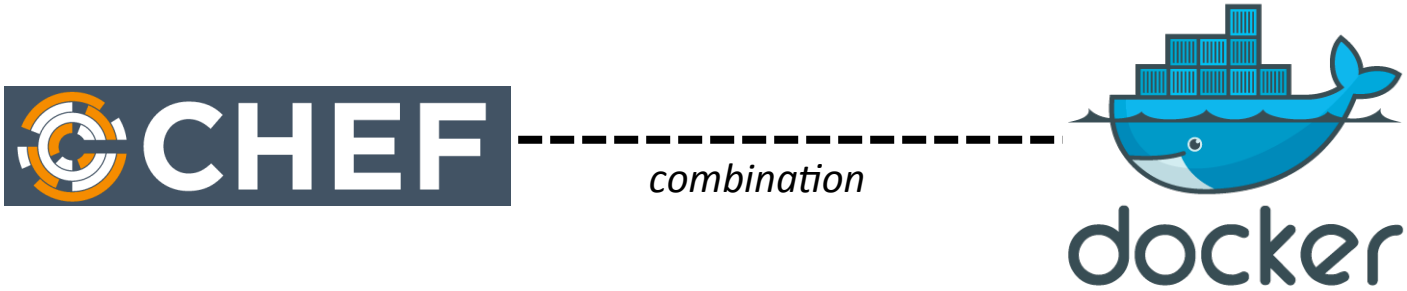
Official Ubuntu base image

stackbrew

Who's using Docker?

More: http://www.docker.com/resources/usecases

Lunch Break ☺


CHEF
combination
docker

Possible Combination Scenario

- Chef recipes “usually” run directly on VMs
- However, recipes may also run inside containers ☺
- E.g., invoke recipe in Dockerfile:

```
# Install mysql server
RUN chef-solo -c ./chef-config.rb -j ./chef-runlist.json
```

```
{
  “run_list”: [
    “recipe[mysql::server]”
  ]
}
```

Further Combination Scenarios

- Use Chef to deploy Docker engine
- Use Chef server to orchestrate Docker containers
- Integrate Docker with Chef
  to enable snapshot-based provisioning for Chef
  - Recipes do not have to be idempotent ☺ ☺


CloudFormation

CloudFormation in a Nutshell

- Orchestration of Amazon Cloud resources
- **Template** (Model)
  - Declarative infrastructure topology
  - JSON-based domain-specific language
- **Stack** (Instance of Template)
- CloudFormation API to manage stacks

Orchestration of Amazon Cloud Resources

- EC2
- Auto Scaling
- CloudFront
- CloudWatch
- DynamoDB
- ElastiCache
- Elastic Beanstalk
- Elastic Load Balancing
- Identity and Access Management

- …

CloudFormation Template Structure

```
{
    "Description" : "A text description for the template usage",

    "Parameters" : {
        // A set of inputs used to customize the template per deployment
    },

    "Resources" : {
        // The set of AWS resources and relationships between them
    },

    "Outputs" : {
        // A set of values to be made visible to the stack creator
    },

    "AWSTemplateFormatVersion" : "2010-09-09"
}
```

“Hello World” Template

```
{
    "Description" : "Create an EC2 instance running the Amazon Linux 32 bit AMI."
        "Parameters" : {
        "KeyPair" : {
            "Description" : "The EC2 Key Pair to allow SSH access", "Type" : "String"
        }
    },
    "Resources" : {
        "Ec2Instance" : {
            "Type" : "AWS::EC2::Instance",
            "Properties" : {
                "KeyName" : { "Ref" : "KeyPair" },
                "ImageId" : "ami-3b355a52"
            }
        }
    },
    "Outputs" : {
        "InstanceId" : {
            "Description" : "The InstanceId of the newly created EC2 instance",
            "Value" : { "Ref" : "Ec2Instance" }
        }
    },
    "AWSTemplateFormatVersion" : "2010-09-09"
}
```

Template for WordPress

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "WordPress is ...",

  "Parameters" : {

    "DBRootPassword": {
      "NoEcho": "true",
      "Description" : "Root password for MySQL",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "41",
      "AllowedPattern" : "[a-zA-Z0-9]*",
      "ConstraintDescription" : "must contain only alphanumeric characters."
    },

    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",
      "Default" : "m1.small",
      "AllowedValues" : [ "t1.micro", "m1.small", "m1.medium", "m1.large", ... ],
      "ConstraintDescription" : "must be a valid EC2 instance type."
    }

    ...
  },
```

Template for WordPress

```
"Mappings" : {
   "AWSInstanceType2Arch" : {
     "m1.small" : { "Arch" : "64" },
     ...
   },

   "AWSRegionArch2AMI" : {
     "us-east-1" : { "32" : "ami-31814f58", "64" : "ami-1b814f72" },
     ...
   }
 },

 "Resources" : {

   "WebServer": {
     "Type": "AWS::EC2::Instance",
     "Metadata" : {
       "AWS::CloudFormation::Init" : {
         "config" : {
           "packages" : {
             "yum" : {
               "httpd" : [],
               "php" : [],
               ...
             }
           },
```

Template for WordPress

```
"sources" : {
  "/var/www/html" : "http://wordpress.org/latest.tar.gz"
},

"files" : {
  "/tmp/setup.mysql" : {
    "content" : { "Fn::Join" : ["", [
      "CREATE DATABASE ", { "Ref" : "DBName" }, ";\n",
      "CREATE USER '", { "Ref" : "DBUsername" }, "'@'localhost' IDENTIFIED BY '", { "Ref" :
                                                                    "DBPassword" }, "';\n",
      "GRANT ALL ON ", { "Ref" : "DBName" }, ".* TO '", { "Ref" : "DBUsername" }, "'@'localhost';\n",
      "FLUSH PRIVILEGES;\n"
      ]]},
    "mode"  : "000644",
    "owner" : "root",
    "group" : "root"
  },

  "/var/www/html/wordpress/wp-config.php" : {
    "content" : { "Fn::Join" : ["", [
      "<?php\n",
      "define('DB_NAME',          '", {"Ref" : "DBName"}, "');\n",
      "define('DB_USER',          '", {"Ref" : "DBUsername"}, "');\n",
      "define('DB_PASSWORD',      '", {"Ref" : "DBPassword" }, "');\n",
      "define('DB_HOST',          'localhost');\n"
    ]] }
  }
},
```

Template for WordPress

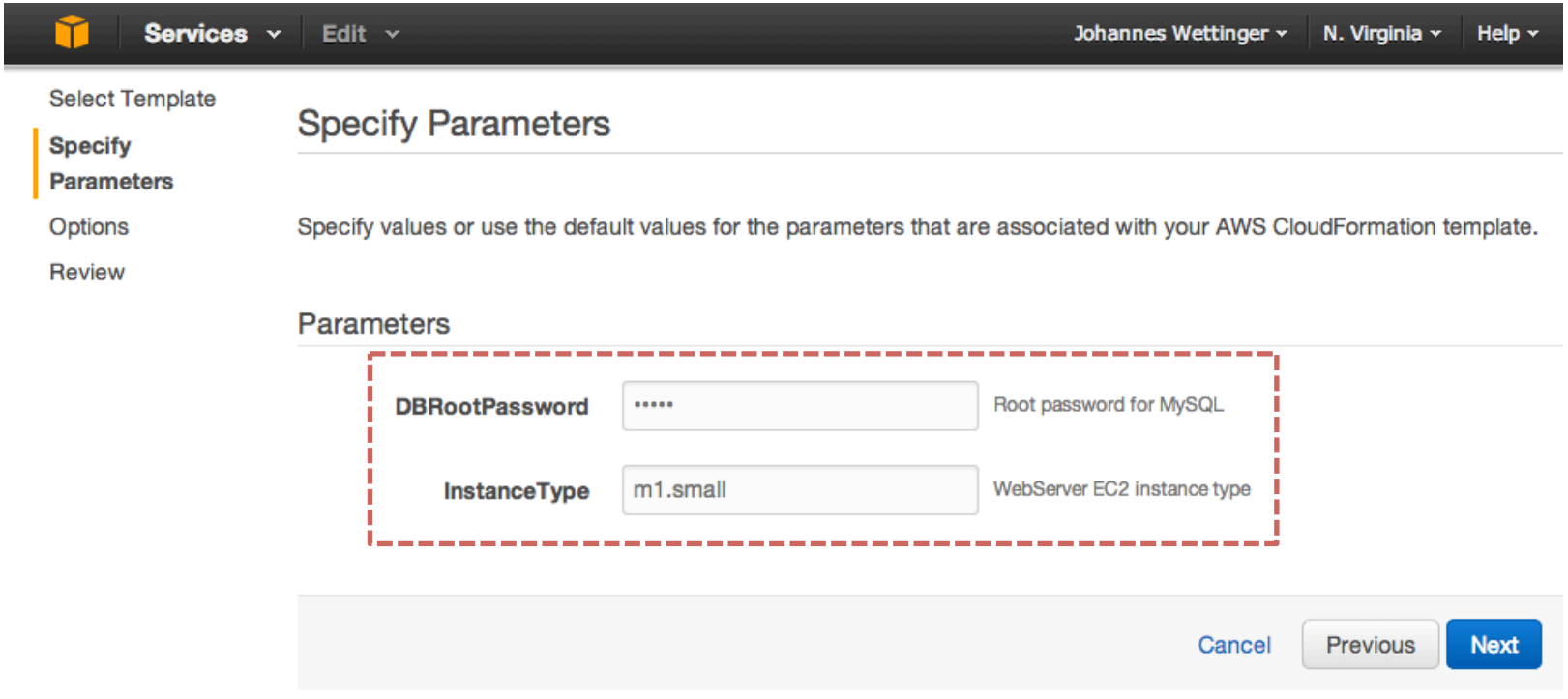
```
"Properties": {
    "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" },
                  { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" : "InstanceType" }, "Arch" ] } ] },

    "InstanceType"   : { "Ref" : "InstanceType" },
    "SecurityGroups" : [ {"Ref" : "WebServerSecurityGroup"} ],
    "KeyName"        : { "Ref" : "KeyName" },

    "UserData"       : { ... [
      "#!/bin/bash -v\n",

      "# Setup MySQL root password and create a user\n",
      "mysqladmin -u root password '", { "Ref" : "DBRootPassword" }, "' || error_exit 'Failure …'\n",
      "mysql -u root --password='", { "Ref" : "DBRootPassword" }, "' < /tmp/setup.mysql ||
                                                                  error_exit 'Failure …'\n",

      "# Setup correct file ownership\n",
      "chown -R apache:apache /var/www/html/wordpress\n",

      ...

      "# All is well so signal success\n",
      "/opt/aws/bin/cfn-signal ..."

    ]]}}
  }
},
```

Template for WordPress

```
    "WebServerSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80 and SSH access",
        "SecurityGroupIngress" : [
          {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp" : "0.0.0.0/0"},
          {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" : { "Ref" : "SSHLocation"}}
        ]
      }
    }
  },

  "Outputs" : {
    "WebsiteURL" : {
      "Value" : { "Fn::Join" : ["", ["http://", { "Fn::GetAtt" : [ "WebServer", "PublicDnsName" ]} ]] },
      "Description" : "WordPress Website"
    }
  }
}
```

Create a WordPress Stack

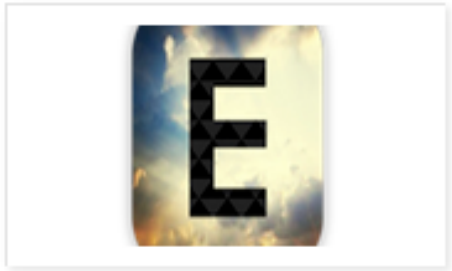
```
"Parameters" : {

   "DBRootPassword": {
     "NoEcho": "true",
     "Description" : "Root password for MySQL",
     "Type": "String",
     ...
   },

   "InstanceType" : { ... },
   ...
```



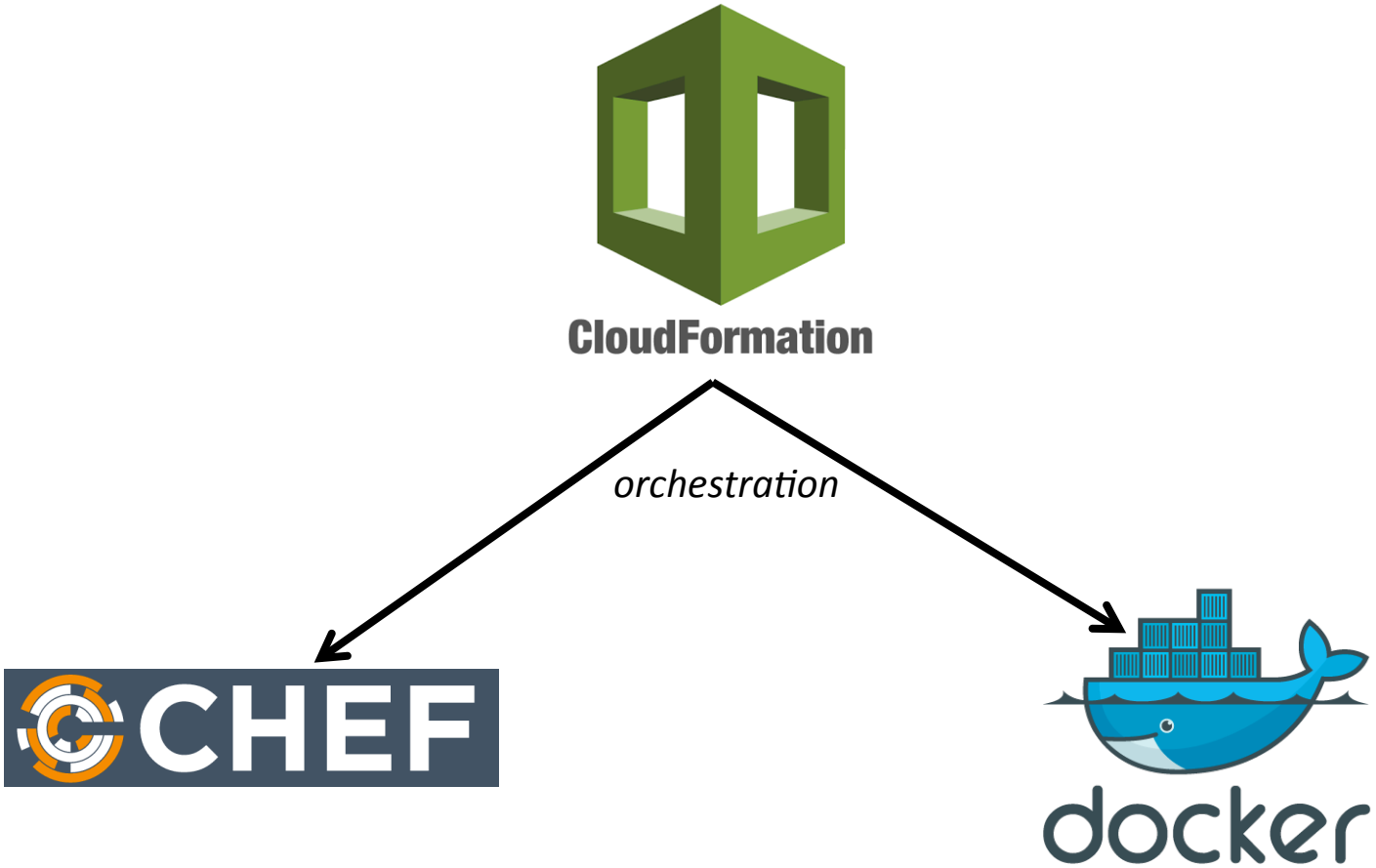
Drupal is an open source content management platform powering millions of websites and applications. For more details click here.	Single EC2 Instance with local MySQL database	**Launch Stack**
	Single EC2 Instance web server with Amazon RDS database instance	**Launch Stack**
	Highly Available Web Server with Multi-AZ Amazon RDS database instance and using S3 for storing file content	**Launch Stack**
Gollum is a simple wiki system built on top of Git that powers GitHub Wikis. For more details click here.	Single EC2 Instance with data stored on ephemeral drive	**Launch Stack**
	Single EC2 Instance with data stored on an EBS volume	**Launch Stack**
Insoshi is an open source social networking platform in Ruby on Rails. For more details click here.	Single EC2 Instance with local MySQL database	**Launch Stack**
	Single EC2 Instance web server with Amazon RDS database instance	**Launch Stack**
	Highly Available Web Server with Multi-AZ Amazon RDS database instance and files stored on S3	**Launch Stack**

Amazon:
~ 130
sample templates

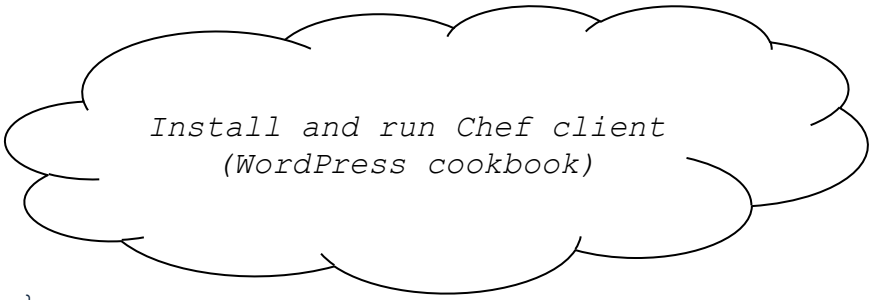
Who's using CloudFormation?



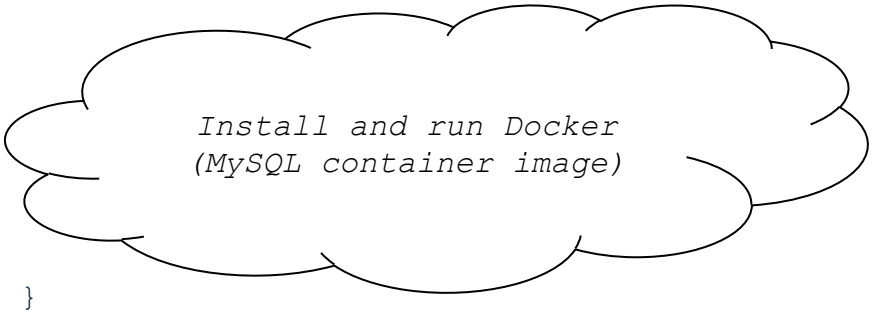
More: http://aws.amazon.com/cloudformation


CloudFormation
orchestration
CHEF
docker

```
{
    ...
    "Resources" : {
        "WordPress" : {
            "Type" : "AWS::EC2::Instance",
```



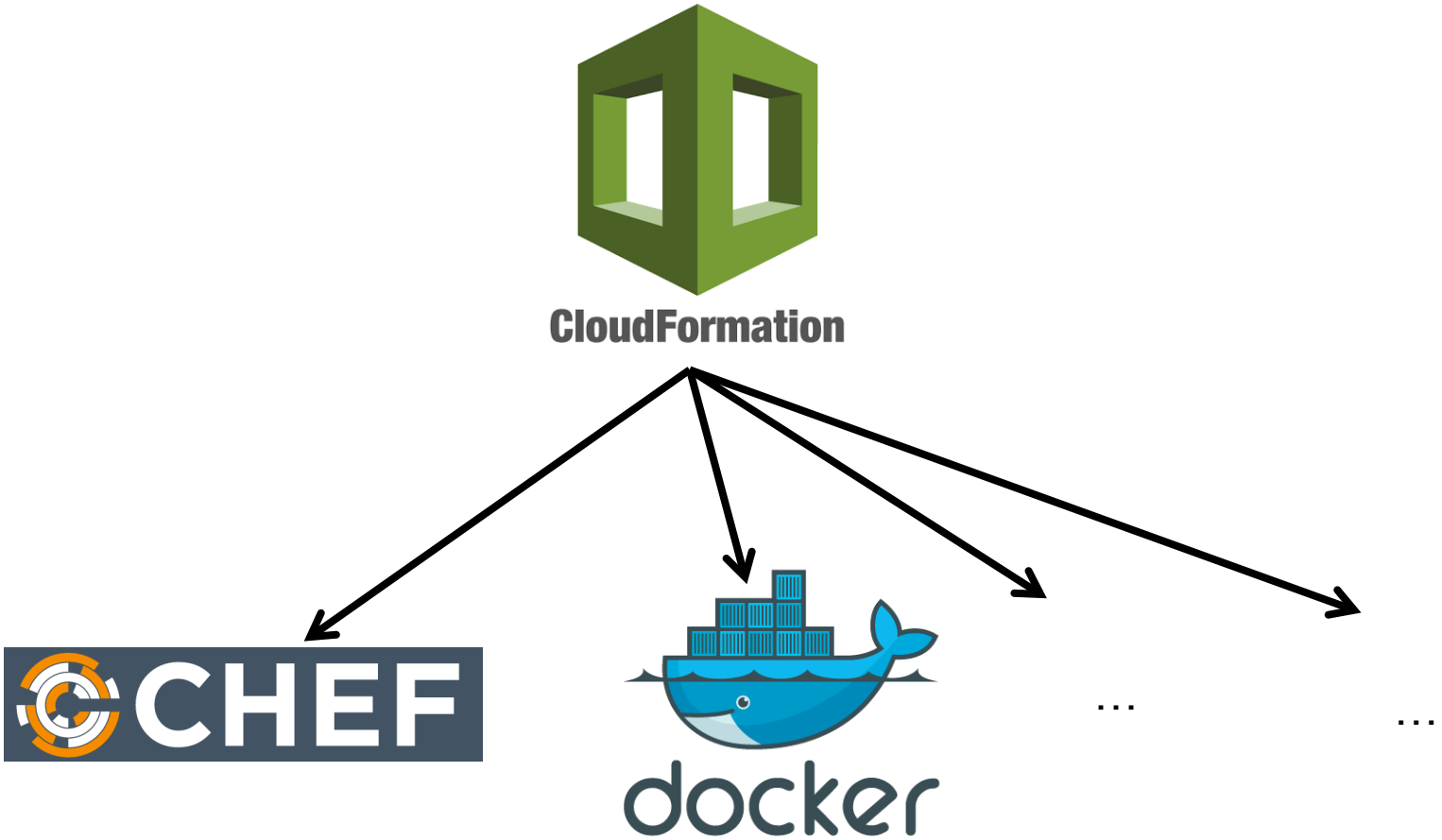

```
        },
        "Database" : {
            "Type" : "AWS::EC2::Instance",
```




```
        }
    },
    ...
}
```

Technical details: http://goo.gl/pOsU0

So, everything is perfect? ☺



Not at all! ☹

Infrastructure-centric ☹ ☺
What about application-centric modeling? → **Juju bundles**

Bound to AWS (public cloud) ☹
How to do orchestration in other clouds, esp. in private clouds?
→ **OpenStack Heat**

Declarative topology models ☹
What about custom runtime management plans?

What about hybrid cloud management?
E.g., OpenStack + Amazon


Research