

Cookie-cutter distributed apps with Chef & GCE

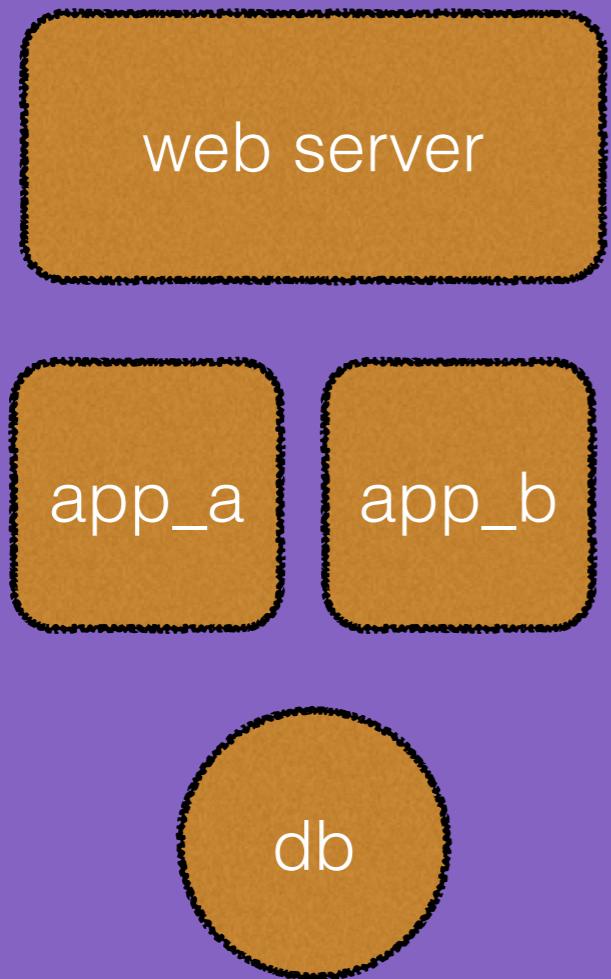
**DevFestDC
21Nov2014**

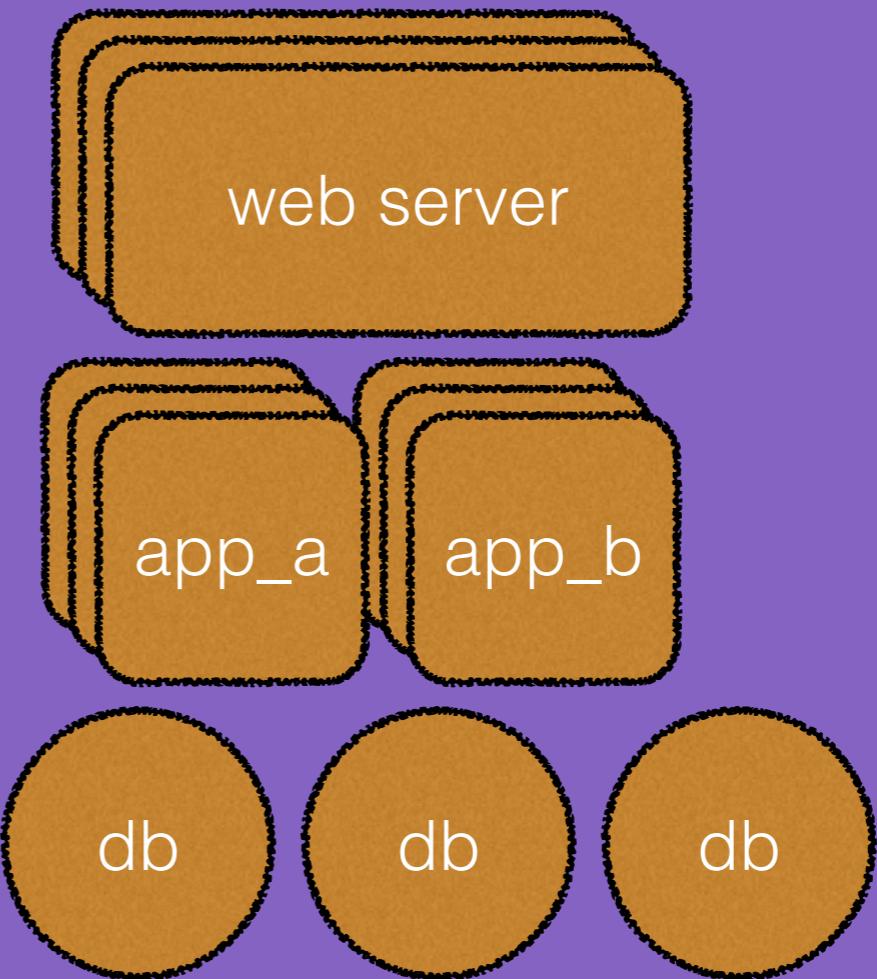
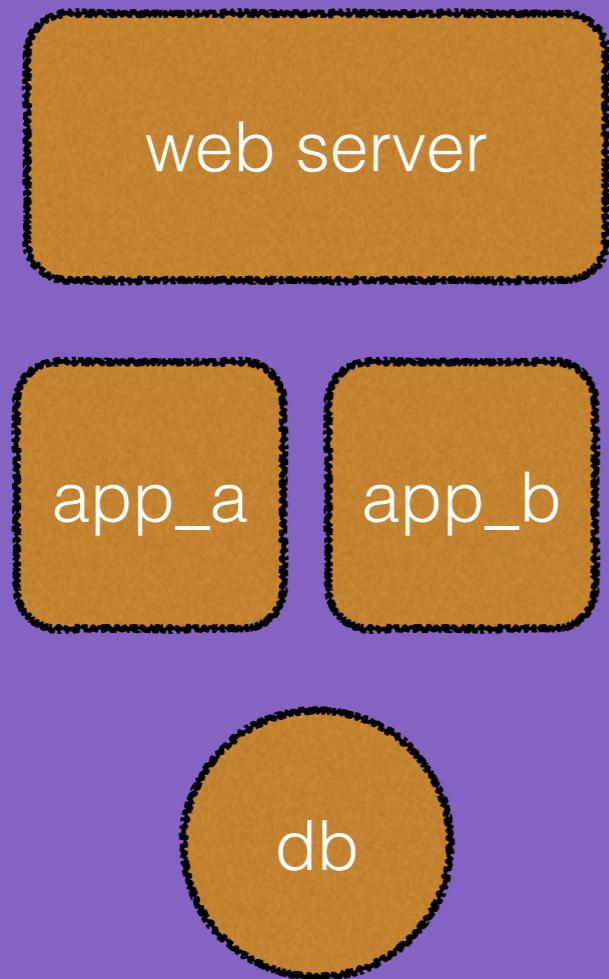
Peter Burkholder
pburkholder@getchef.com
@pburkholder

whoami

whoareu

Distributed Applications





web

ap ap



web

ap ap



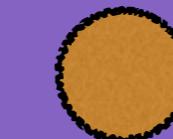
web

ap ap ap ap



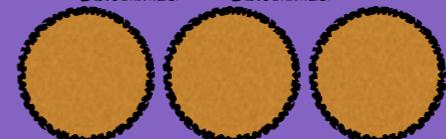
web

ap ap



web

ap ap



web

ap ap



$$\$ + \text{hourglass} = \text{poop}$$



devops

Chef

Chef is a Language

describes *resources*
implemented by *provisioners*

Chef is a Service

serves Chef code, as *cookbooks*
discovery of Chef resources on *nodes*

Chef is a Client

consumes *code* and *attributes*
compiles code into resource object
converges the node by acting on the resource objects

Chef & Cloud

Chef Provisioning

node resources:

user, package, service, file,

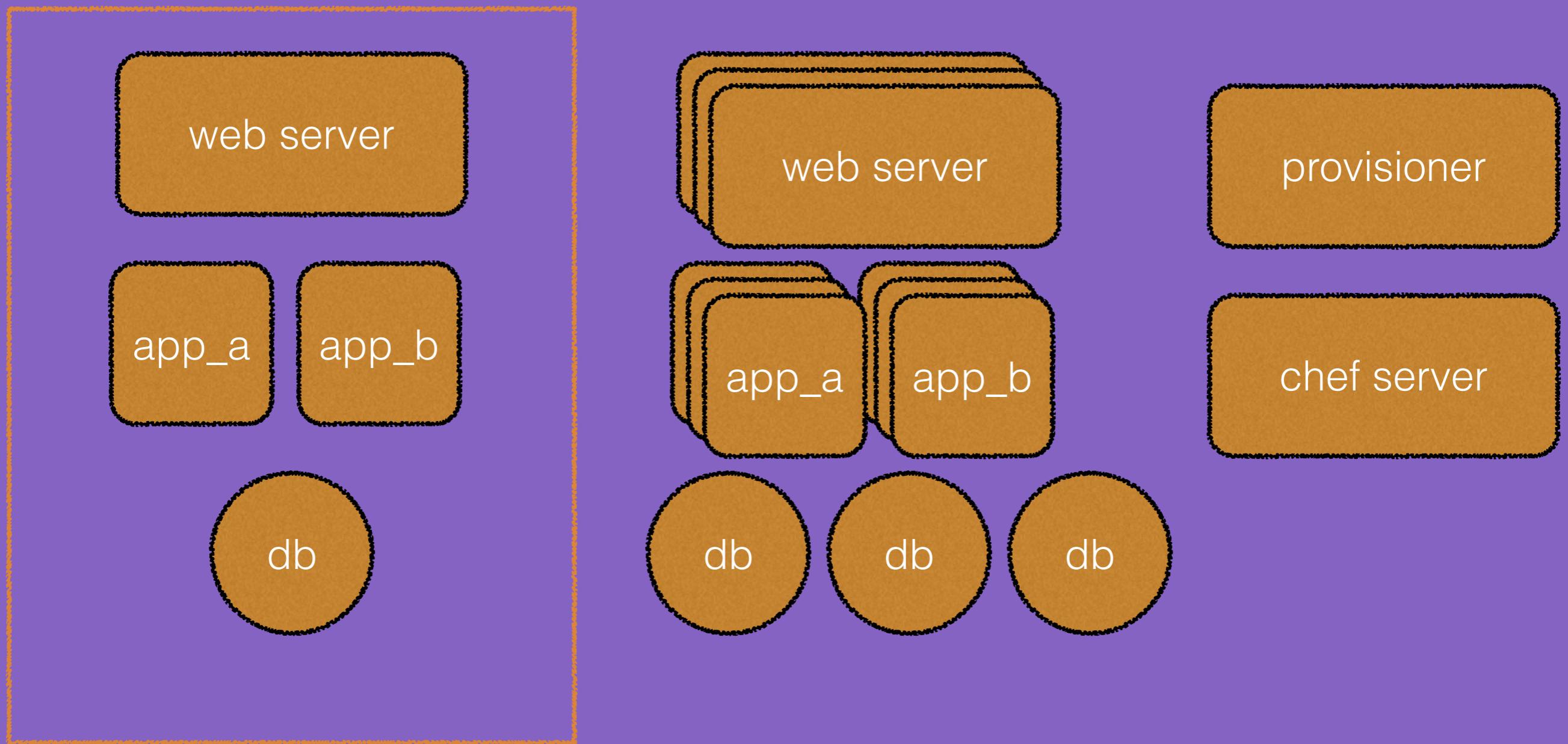
provisioning resources:

machine, machine_batch,

gce resources:

disk, instance, network, firewall, ...

runs from a *provisioning node*



```
gce = data_bag_item("gce", "service_account")
client = client_rb(env)
validation = validation_pem()

gce_instance "webserver" do
  boot_disk_image node['dc_devfest']['boot_disk_image']
  tags ["http-server", "webserver", "myapp-webserver"]
  first_boot_json "{\"run_list\": [\"role[debian_base]\", \"recipe[my-app::balancer]\"]}"
  # boilerplate:
  client_rb client
  validation_pem validation
  client_email gce['client_email']
  key_location gce['key_location']
  project_id gce['project_id']
  machine_type "n1-standard-1"
  zone_name "us-central1-a"
  service_account_scopes [
    "compute", "userinfo.email", "devstorage.full_control"
  ]
  auto_restart true
  on_host_maintenance "MIGRATE"
  action :create
end
```

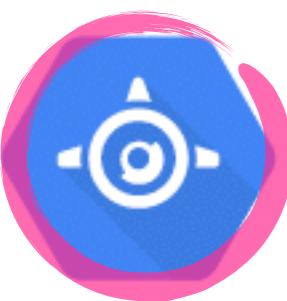
```
include_recipe 'dc_devfest::database'
include_recipe 'dc_devfest::myapp-a'
include_recipe 'dc_devfest::myapp-b'
include_recipe 'dc_devfest::webserver'
```

```
$ chef-client -z -r 'dc_devfest::default'
```



Google Cloud Platform

Compute



App Engine



Compute Engine

Storage



Cloud Storage

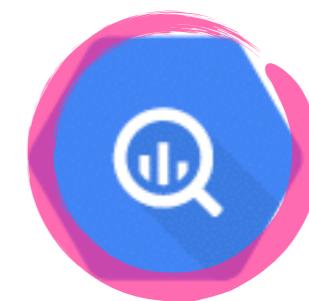


Cloud Datastore



Cloud SQL

Big Data/Analysis



BigQuery



Cloud Endpoints

GCE

Google Cloud Platform ⇒ Google Compute Engine

- AutoMigration
- AutoRestart
- Consistent internal DNS
- Usable internal DNS
- 10m spin up billing with 1m increments
- Long-term discounts
- Mutable metadata
- Google OAuth integration
- APIs and CLIs
- Autoscaling (beta)
- Load-balancing
- Image snapshots
- Image templates
- Container Engine
- VPC

GCE

Google Cloud Platform ⇒ Google Compute Engine



Services ▾

IAM

EC2

VPC

S3

CloudFront

Edit ▾

Tho

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Spot Requests

Reserved Instances

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Load Balancers

Key Pairs

Network Interfaces

AUTO SCALING

Launch Configurations

Auto Scaling Groups

Actions ▾

Filter: All resource types ▾

All event types ▾

All statuses ▾

Search Events

X

Resource Name ▾

Resource Type ▾

Resource Id ▾

Availability Zon ▾

Event Type ▾

Event Status ▾

Event Descripti ▾

Start Ti

There are no events of the selected type currently in progress in this region.

Click This

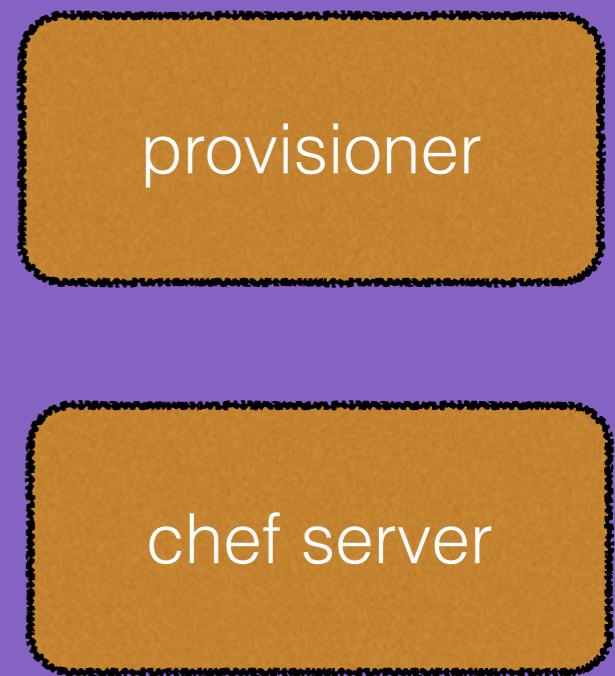
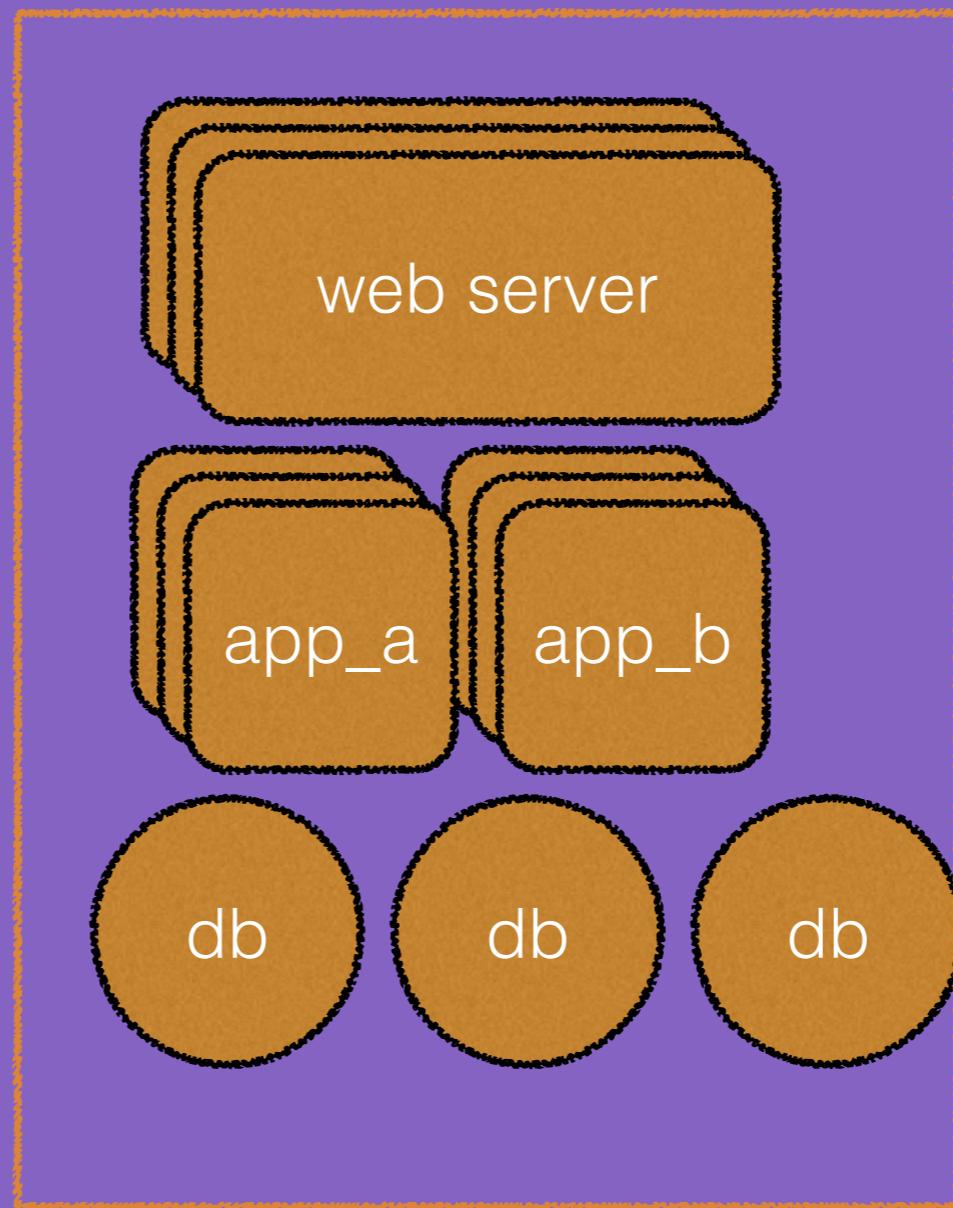
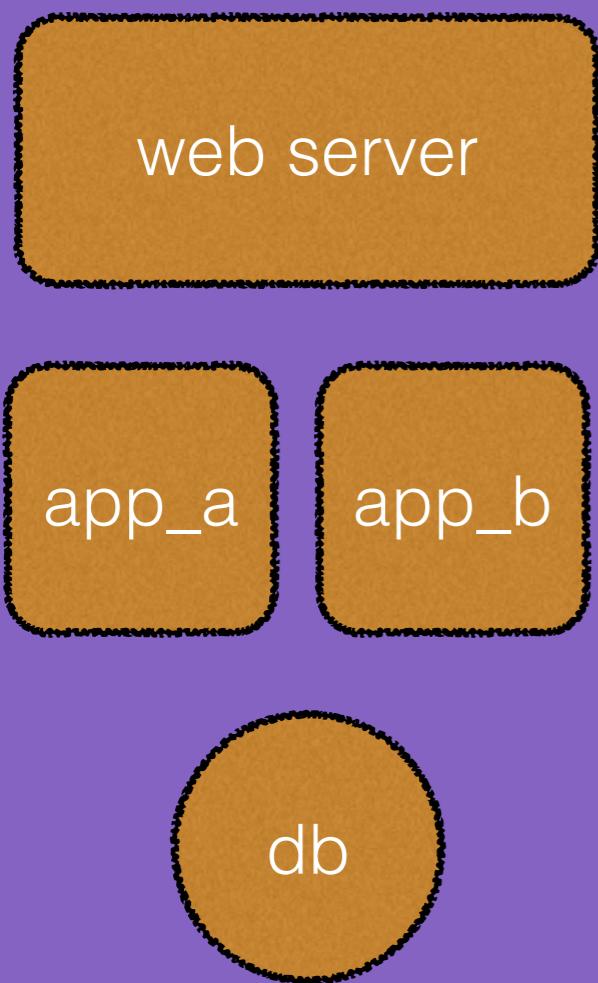
**Instances Scheduled To Reboot
Will Be Listed Here**

Select an event above

...

	OnDemand	Long Term	Details
AWS	0.070	0.027	1cpu, 3.75G
GCE	0.063	0.025	1cpu, 3.75G
Azure	0.154	?	1cpu, 1.75G
Dig. Ocean	0.060		2cpu, 4G

	OnDemand	Long Term	Details
AWS	\$50.40	\$19.44	1cpu, 3.75G
GCE	\$45.36	\$18	1cpu, 3.75G
Azure	\$110.88	?	1cpu, 1.75G
Dig. Ocean	\$40.00		2cpu, 4G



```
gce = data_bag_item("gce", "service_account")

env = node.environment
client = client_rb(env)
validation = validation_pem()

node['dc_devfest']['webserver_instances'].times do |instance_index|
  gce_instance "#{env}-webserver-#{instance_index}" do
    boot_disk_image node['dc_devfest']['boot_disk_image']
    tags ["http-server", "webserver", "myapp-webserver"]
    first_boot_json "{\"run_list\":[\"role[debian_base]\",\"recipe[my-app::balancer]\"]}"
  end
end
```

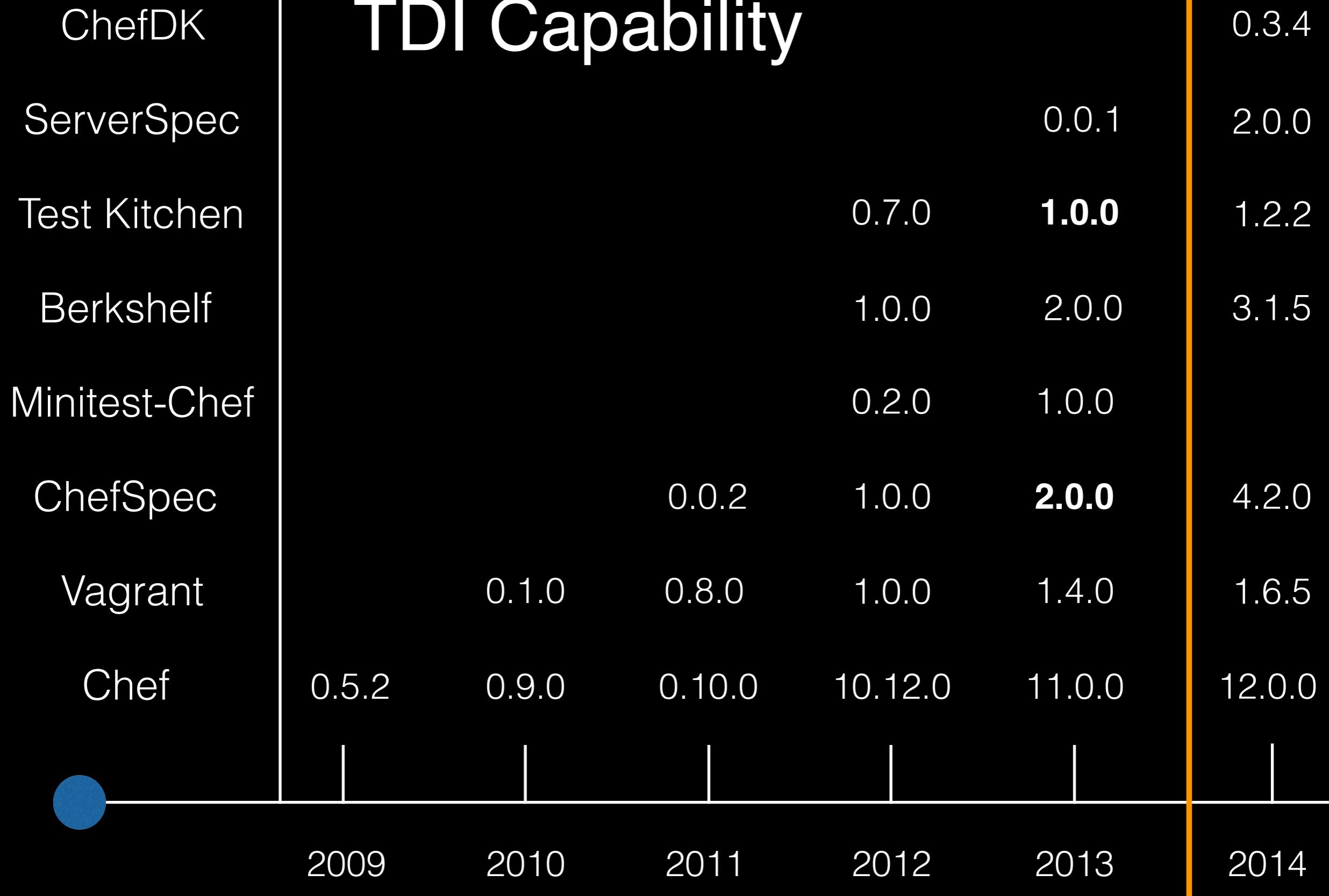
```
{
  "name": "prod",
  "default_attributes": {
    "dc_devfest": {
      "database_instances": 3,
      "myapp-a_instances": 2,
      "myapp-b_instances": 6
    }
  }
}
```

```
chef-client -z -r 'dc_devfest' \  
-c .chef/client.rb -E prod
```

Let's talk about testing

devops

TDI Capability



code:
user
package
service

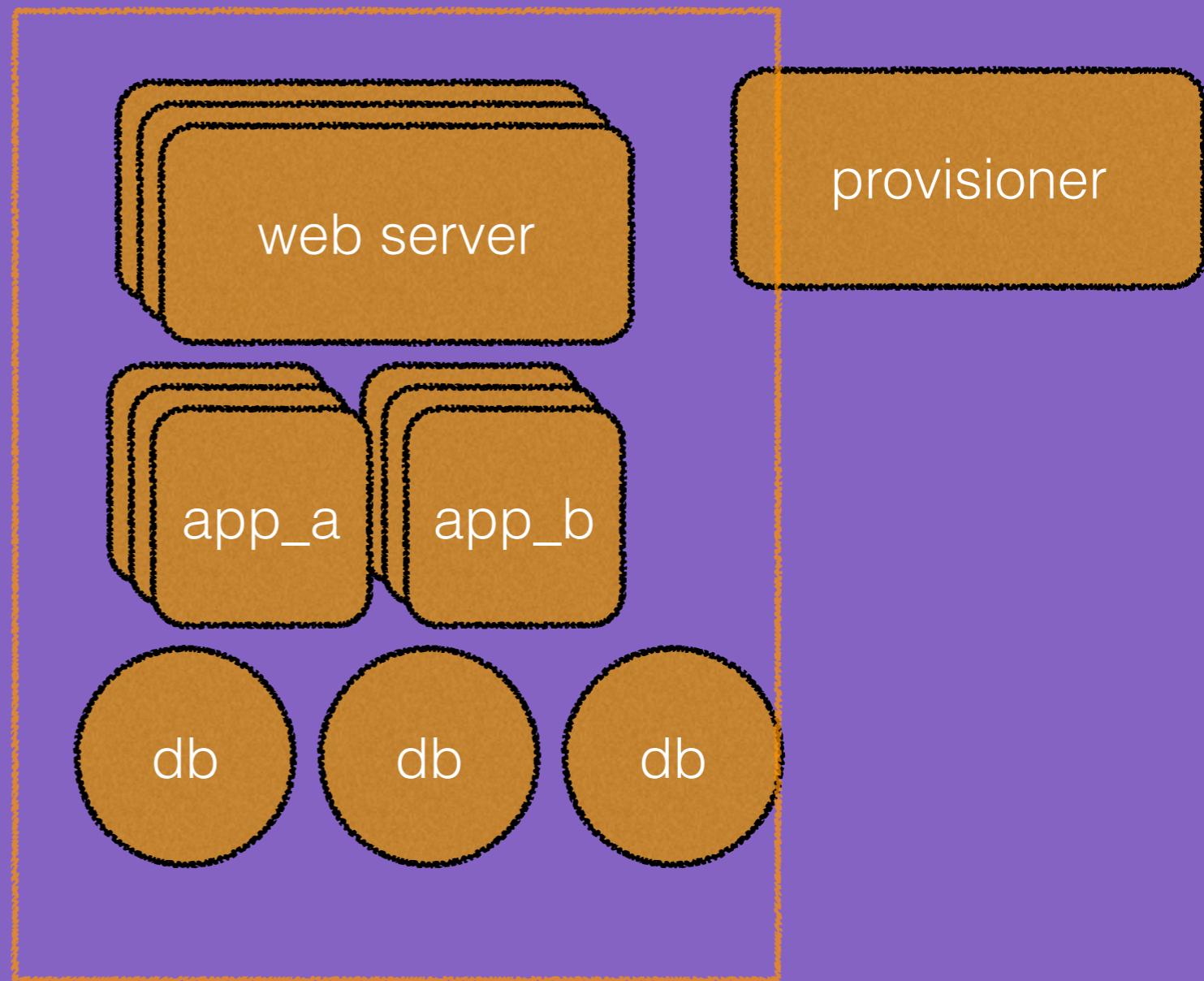
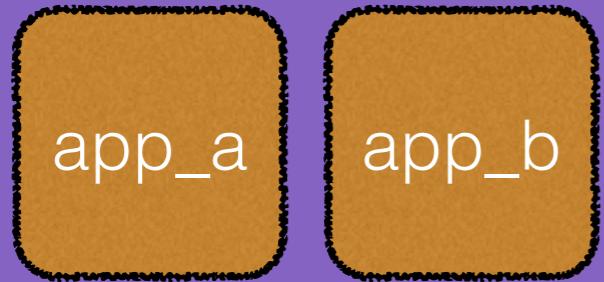
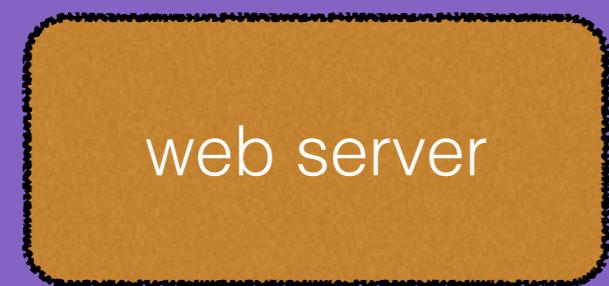
client pulls code and
relevant *attributes*
from server

combines with
discovery of node
data

**rubocop and
foodcritic**

ChefSpec
compile into objects:
user(name, passwd)
package(version)
service(enabled)

KitchenCI
converge via
test-and-repair:
user /home/peterb
package nginx 1.7.1
nginx start



Summarize

GCE IAAS platform (v. AWS & Digital Ocean)

+ Chef = IAC (infrastructure as code) that is

- testable
- repeatable
- scalable

paradise?



please don't hurt me

file a bug report or send a PR

@pburkholder or pburkholder@getchef.com

http://getchef.com http://learn.getchef.com

**http://cloud.google.com/starterpack
chef14-con promo code \$500**

github.com/pburkholder/gce_preso