

# Chef

```
cd ~/scripts && find . -type f -name "*.sh" -exec rm -rf {} \;
```

**You've got a cloud.  
Now what ?**

# Problems

**What lots of  
people do now**

# Manual

**Often error prone**

**Slow**

**Time Consuming**

**Process Heavy**

**I know, I write  
bash scripts**

```
#!/bin/bash
NODES="webserver.example.com database.example.com"
for n in $NODES
do
    ssh $n uptime
done
```

**SSH for loops**  
**Tests, what tests ?**

**What if we could version control  
our servers and network, the  
way we do with code?**



# Infrastructure as Code

**Desired state**

**Transformations to achieve it**

**As text files**

**In version control**

**Testable and reproducible**

# Chef is

**a ruby-based system integration framework  
designed to bring the benefits of configuration  
management to your entire infrastructure**



Huh?

a **Configuration management** system

a **System integration** platform

an **API** for your entire infrastructure

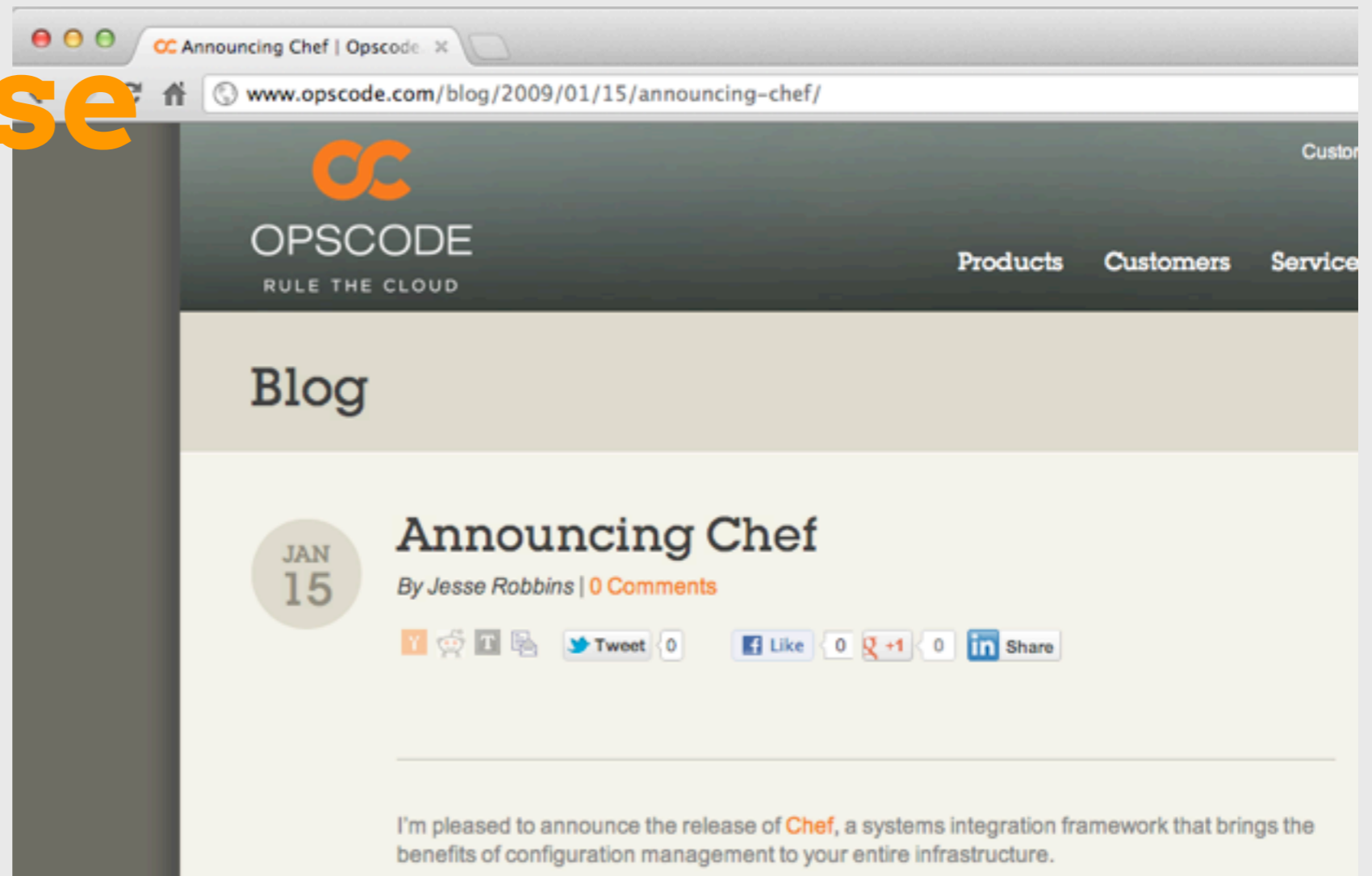
# Initial commit

Mar 05, 2008

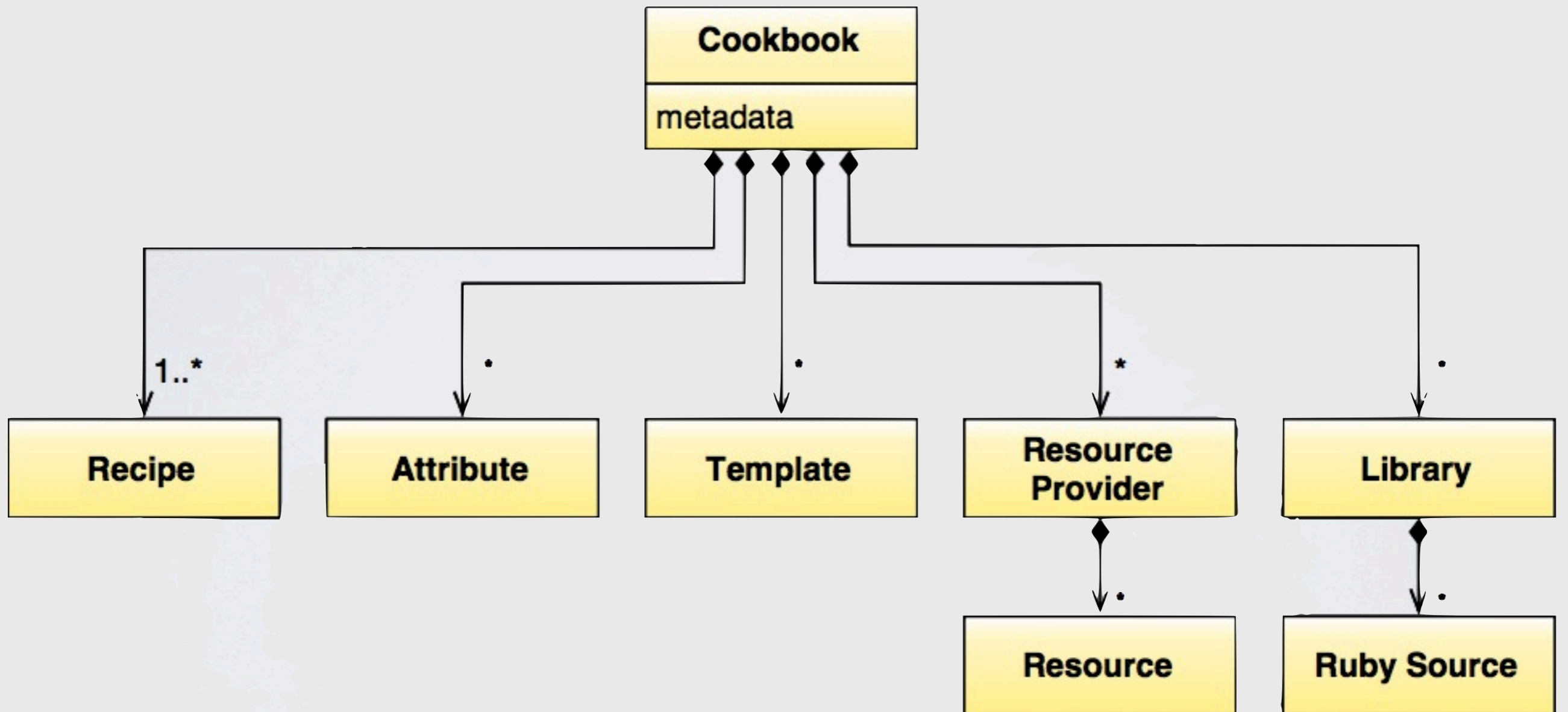


# 1st release

Jan 15, 2009



# Cookbooks



# Git cookbook

```
▪
|-- README.md
|-- attributes
|  `-- default.rb
|-- metadata.rb
|-- recipes
|  |-- default.rb
|  |-- server.rb
|  `-- source.rb
`-- templates
    `-- default
        |-- sv-git-daemon-log-run.erb
        `-- sv-git-daemon-run.erb
```

4 directories, 11 files

# metadata.rb

```
name "git"
maintainer "Opscode, Inc."
maintainer_email "cookbooks@opscode.com"
license "Apache 2.0"
description "Installs git and/or sets up a Git server daemon"
long_description IO.read(File.join(File.dirname(__FILE__), 'README.md'))
version "1.0.2"
recipe "git", "Installs git"
recipe "git::server", "Sets up a runit_service for git daemon"
recipe "git::source", "Installs git from source"

%w{ amazon arch centos debian fedora redhat scientific ubuntu windows }.each do |os|
  supports os
end

supports "mac_os|x", ">= 10.6.0"

%w{ build-essential dmg runit yum }.each do |cb|
  depends cb
end
```



# recipes/default.rb

```
case node['platform_family']  
when "debian", "ubuntu"  
  package "git-core"  
else  
  package "git"  
end
```

# recipes/server.rb

```
include_recipe "git"

directory "/srv/git" do
  owner "root"
  group "root"
  mode 00755
end

case node['platform_family']
when "debian"
  include_recipe "runit"
  runit_service "git-daemon"
else
  log "Platform requires setting up a git daemon service script."
  log "Hint: /usr/bin/git daemon --export-all --user=nobody --group=daemon --base-path=/srv/git"
end
```

# MongoDB cookbook

```
.  
|-- README.md  
|-- attributes  
| `-- default.rb  
|-- definitions  
| `-- mongodb.rb  
|-- files  
| `-- default  
|   `-- mongodb.init  
|-- libraries  
| `-- mongodb.rb  
|-- metadata.json  
|-- metadata.rb  
|-- recipes  
| |-- 10gen_repo.rb  
| |-- configserver.rb  
| |-- default.rb  
| |-- mongos.rb  
| |-- replicaset.rb  
| `-- shard.rb  
`-- templates  
    `-- default  
        `-- mongodb.default.erb
```

8 directories, 14 files

# templates/default/mongodb.default.erb

```
DAEMON="<> @daemon_path <>"
NAME="<> @name <>"

DAEMON_OPTS=""

<> @port ? "DAEMON_OPTS=\"\$DAEMON_OPTS --port #{@port}\" : "" <>
<> @dbpath ? "DAEMON_OPTS=\"\$DAEMON_OPTS --dbpath #{@dbpath}\" : "" <>
<> @logpath ? "DAEMON_OPTS=\"\$DAEMON_OPTS --logpath #{@logpath}\" : "" <>

<> @config ? "DAEMON_OPTS=\"\$DAEMON_OPTS --config #{@config}\" : "" <>
<> @configdb ? "DAEMON_OPTS=\"\$DAEMON_OPTS --configdb #{@configdb}\" : "" <>

<> @configsvr ? "DAEMON_OPTS=\"\$DAEMON_OPTS --configsvr\" : "" <>
<> @shardsvr ? "DAEMON_OPTS=\"\$DAEMON_OPTS --shardsvr\" : "" <>

<> if @replicaset_name <>
DAEMON_OPTS="\$DAEMON_OPTS --replSet <> @replicaset_name <>"
<> end <>
```

**A lot of layers in Chef  
have **attributes** :**

**Environments,  
Nodes,  
Roles  
and Cookbooks.**

**Chef groups Nodes  
into **Environments**.  
You can have as many environments  
as you like.**

**Nodes have one or more **Roles**  
applied to them.**

**A Role is comprised of one or more  
**Cookbooks**.**

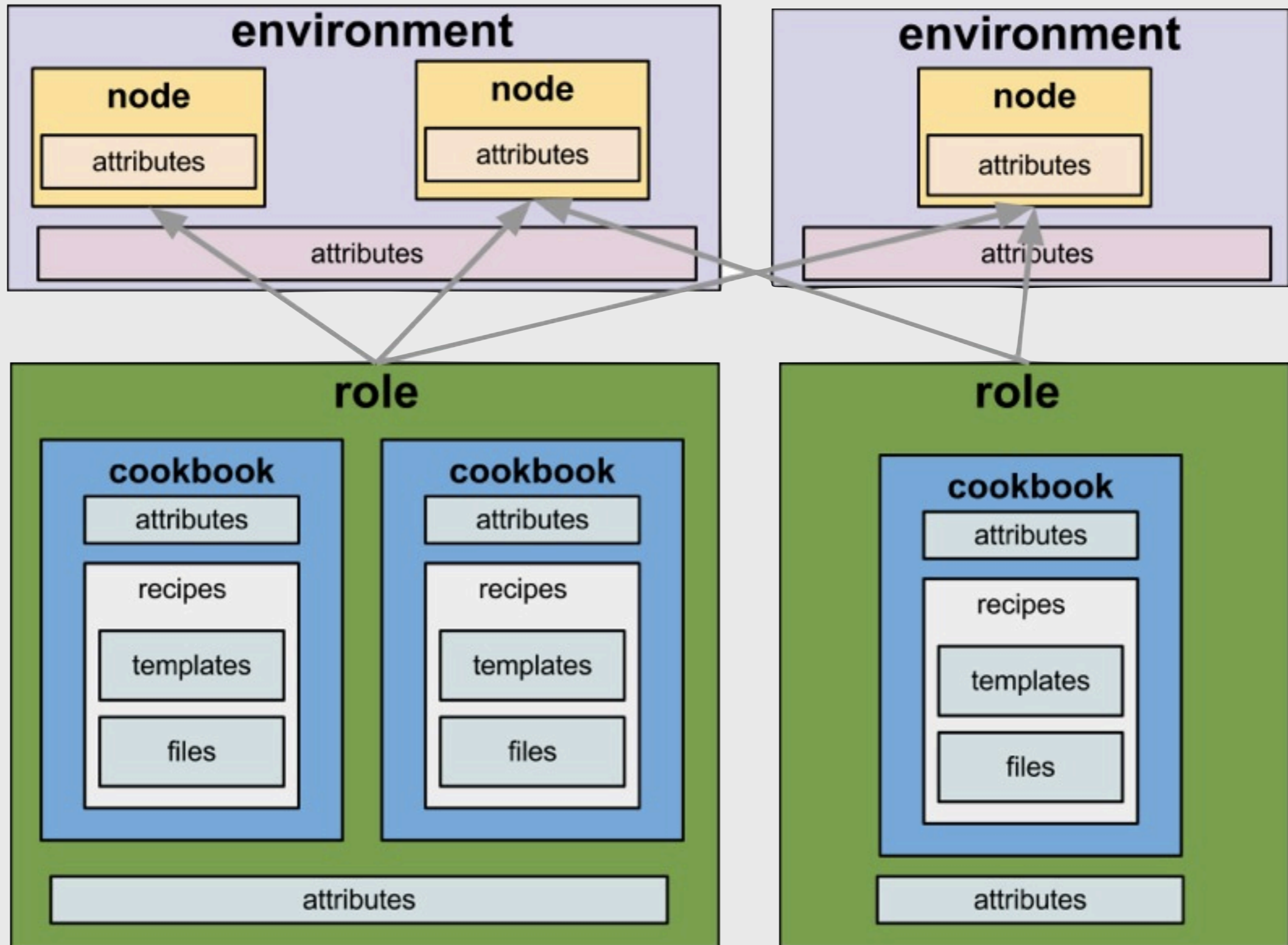
# Cookbooks

are comprised of one or more

**Recipes, Templates,**

**Files** and more.

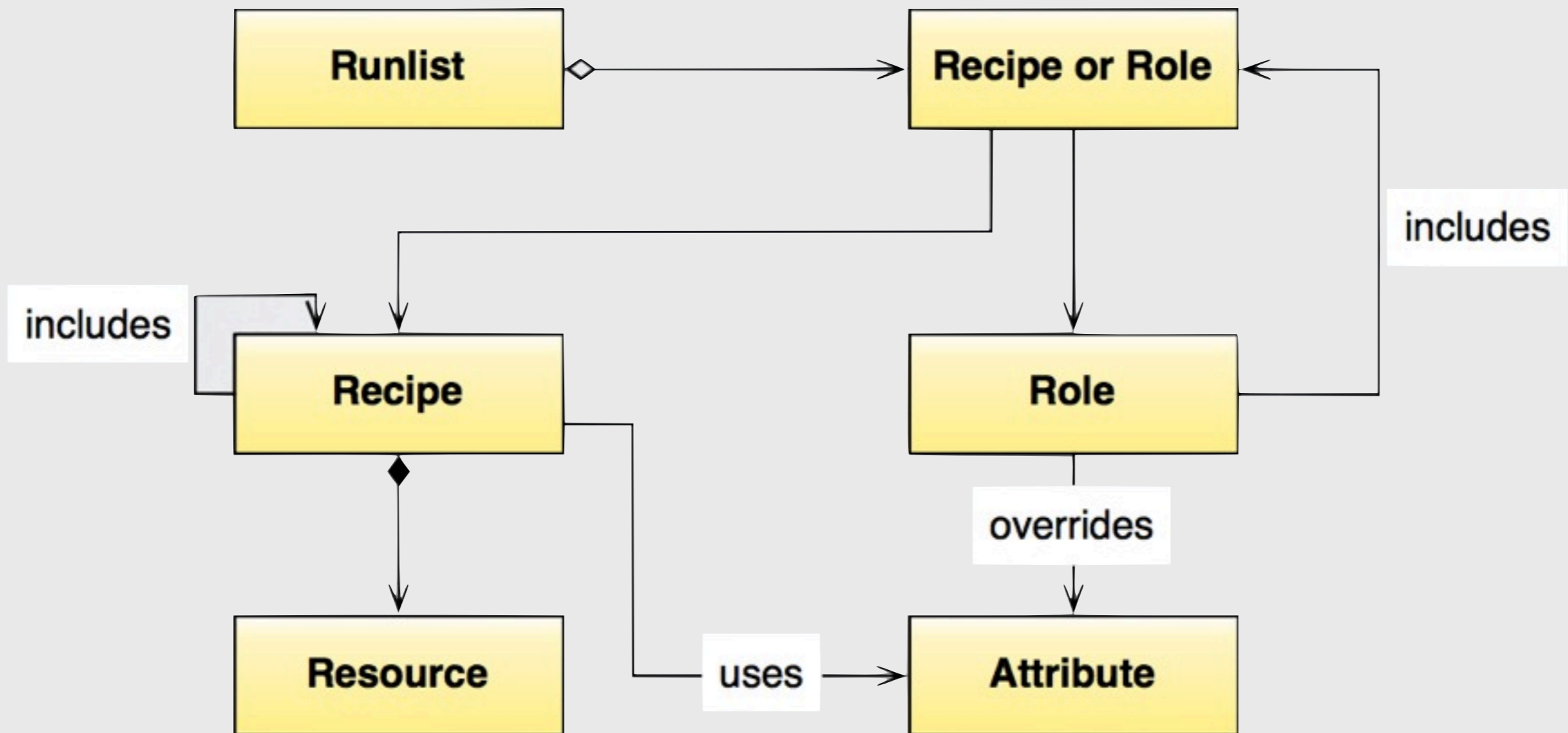
# Tadaaaa!





**How do cookbooks  
get applied to  
servers?**

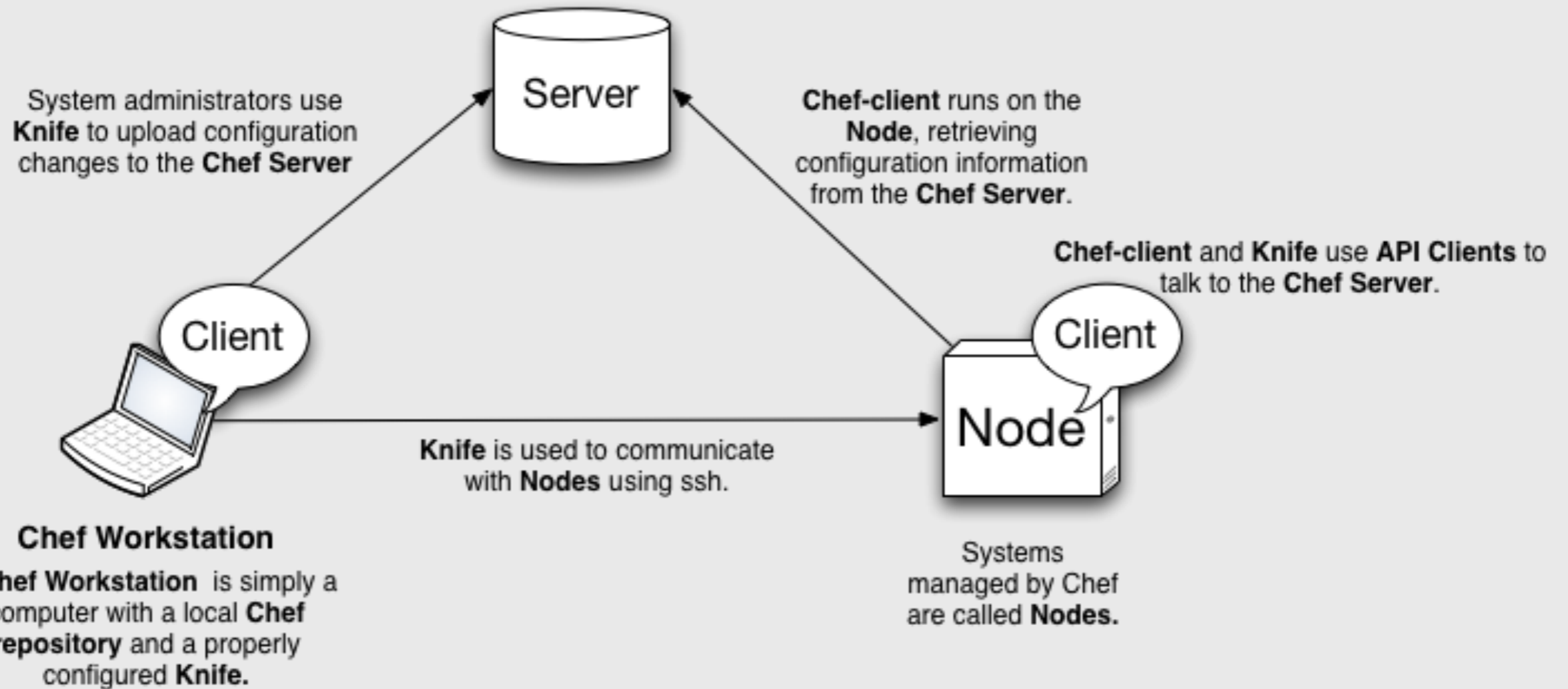
# Runlists executes on nodes



## **In plain words**

- 1. Clients talk to a Chef server, and ask for their configuration.**
- 2. Clients run ruby code on themselves, to converge to the aimed configuration**

The **Chef Server** is the centralized store of your infrastructure's configuration.



source: <http://wiki.opscode.com/display/chef/Architecture+Introduction>

**Cool stuffs**

# Data bags

```
{  
  "id": "mehlah",  
  "name": "Mehdi Lahmam B.",  
  "uid": "2001",  
  "group": "www-data",  
  "home": "/home/mehlah",  
  "roles": [  
    "unix",  
    "sudo"  
  ],  
  "ssh_keys": [  
    "akeystring"  
  ]  
}
```

# Search

**through Data Bags, Roles, Nodes and more...**

```
search(:node, "*:*") { |n| nodes << n }
```

## Examples

```
knife search role "name:production*"
```

```
knife search node 'platform:ubuntu'
```

```
knife search admins 'id:christophe'
```

```
search(:node, 'dmi_system_manufacturer:Dell*').each do |node|  
  puts node[:dmi][:system][:serial_number] + "\t" + node[:fqdn]  
end
```

```
XJS1NF1 www1.example.org
```

```
XJS1NF2 www2.example.org
```

```
XJS1NF3 www3.example.org
```

# Knife

```
$ gem install knife-ec2
Fetching: excon-0.6.6.gem (100%)
...
Successfully installed knife-ec2-0.5.10
6 gems installed
$ knife ec2 server create 'role[webserver]' -I ami-7000f019 -f m1.small
Instance ID: i-9e7ef1fe
Flavor: m1.small
Image: ami-7000f019
Region: us-east-1
Availability Zone: us-east-1c
Security Groups: default
SSH Key: relevance_aws
Waiting for server...
```



**Not so cool stuffs**

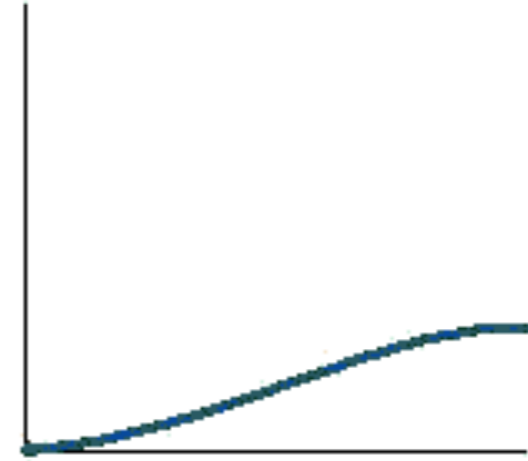
Classical learning curves for some common editors

11-17-04

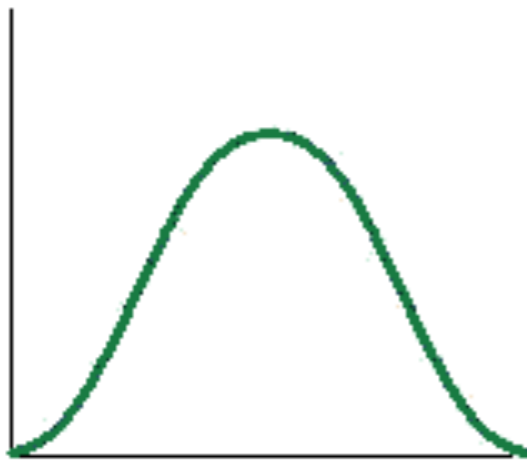
**Notepad**



**Pico**



**Visual Studio**



**vi**



**emacs**



# Local development and Vagrant

**Builds virtual machines from a text file and  
some Chef cookbooks**

# Vagrantfile

```
Vagrant::Config.run do |config|
  config.vm.box = "precise32"
  config.vm.network :hostonly, "33.33.33.10"

  config.vm.provision :chef_solo do |chef|
    chef.cookbooks_path = ["cookbooks", "cookbooks-src"]

    chef.add_recipe "build-essential"
    chef.add_recipe "apt"
    chef.add_recipe "ruby_build"
    chef.add_recipe "rbenv::system"
    chef.add_recipe "rbenv::vagrant"
    chef.add_recipe "nginx"
    chef.add_recipe "unicorn"
    chef.add_recipe "rails-lastmile"

    chef.json = {
      'rvm' => {
        'default_ruby' => 'ruby-1.9.3-p194',
        'gem_package' => {
          'rvm_string' => 'ruby-1.9.3-p194'
        }
      }
    }
  end
end
```

# Benefits

**Version control dev environment**

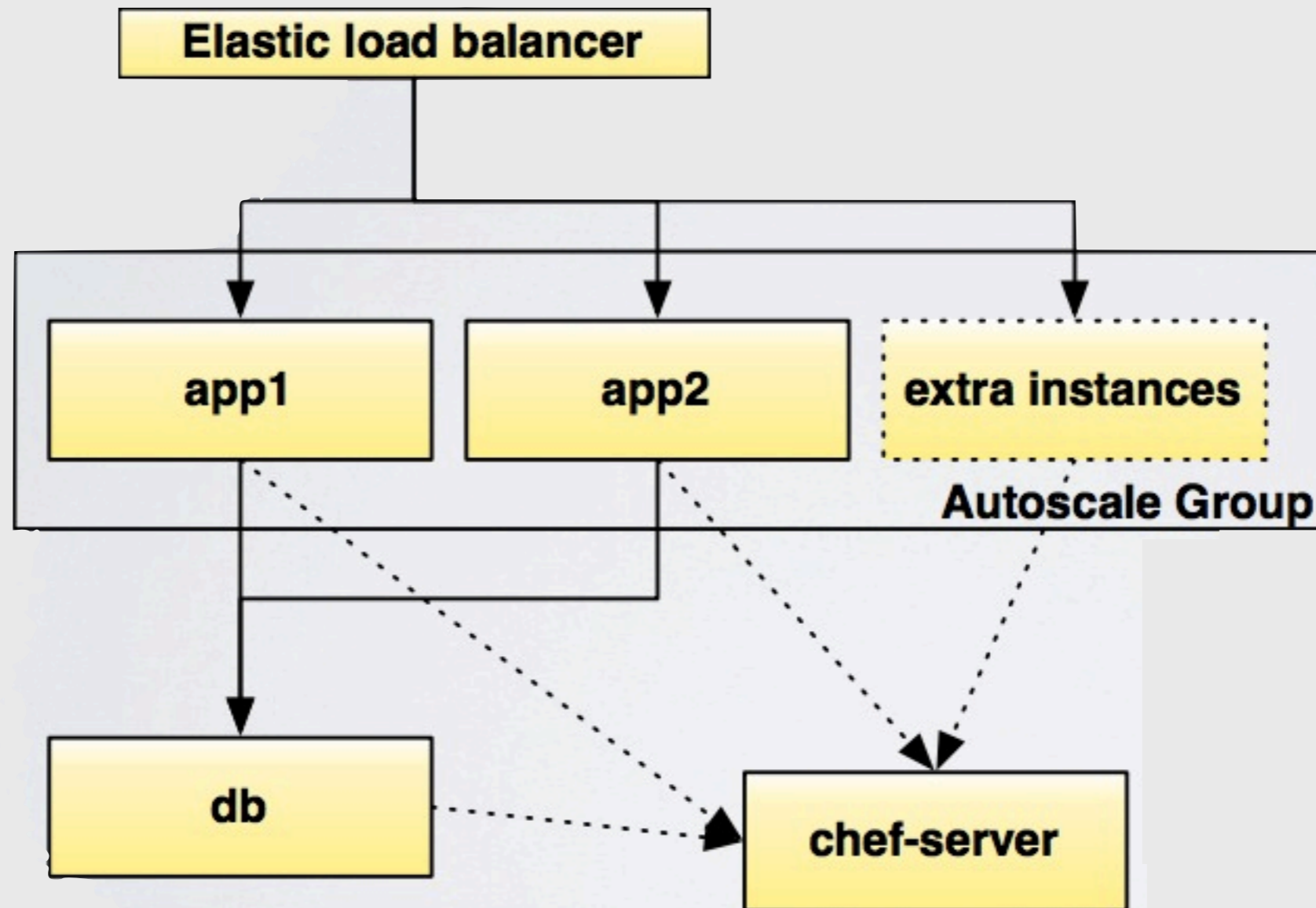
**Share configs across team**

**Recreate project or client setup**

**Keep host environment clean**

**1D Touch**

# 1D Touch + Rackspace Cloud Servers




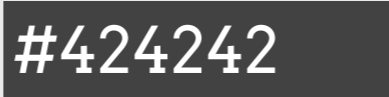

**Thanks!**

@mehlah



# Notes on design

Typeface: Arvo

Colors:  #FF9900  #424242  #3299BB #E9E9E9