

# Configuring Nagios with Chef

Bryan McLellan

Technical Program Manager, Open Source

[btm@opscode.com](mailto:btm@opscode.com) / [@btmspox](https://twitter.com/btmspox)



**Nagios**<sup>®</sup>  
**World Conference**  
**North America**



OPSCODE

RULE THE CLOUD

- Who am I?
- Why automation
- Introduction to Chef
- Nagios Demo
- Questions

- Chef  
Early developer, user, pundit
- 10+ years in Systems Administration  
Computer repair, ISPs, Corporate IT,  
Web operations
- Event Logistics Volunteer  
Traffic Control, Parking,  
Communications, Networking,  
Emergency Management
- Hacker-Operator  
Trucks, Bikes, Radios, Tractors



Bare Metal Deployment

## Bare Metal Deployment

- Purchasing

## Bare Metal Deployment

- Purchasing
- Vendor build

## Bare Metal Deployment

- Purchasing
- Vendor build
- Delivery

## Bare Metal Deployment

- Purchasing
- Vendor build
- Delivery
- Installation



## Bare Metal Deployment

- Purchasing
- Vendor build
- Delivery
- Installation
- OS deployment

## Bare Metal Deployment

- Purchasing
- Vendor build
- Delivery
- Installation
- OS deployment
- Application deployment

## Bare Metal Deployment

- Purchasing
- Vendor build
- Delivery
- Installation
- OS deployment
- Application deployment

Weeks?

## Cloud Deployment

- ~~• Purchasing~~
- ~~• Vendor build~~
- ~~• Delivery~~
- ~~• Installation~~
- ~~• OS deployment~~
- Application deployment

**Nearly immediate**

## Cloud Deployment

- ~~• Purchasing~~
- ~~• Vendor build~~
- ~~• Delivery~~
- ~~• Installation~~
- ~~• OS deployment~~
- Application deployment

Nearly immediate

**Must be fast**

## Good Reasons:

- More agility and faster scalability
- Improved infrastructure documentation
- Better disaster recovery

## Good Reasons:

- More agility and faster scalability
- Improved infrastructure documentation
- Better disaster recovery

## Really Good Reasons:

- Spend less time on monotonous tasks
- Spend more time solving interesting problems





- Idempotent



- Idempotent
- Reasonable



- Idempotent
- Reasonable
- Primitives





- Idempotent
- Reasonable
- Primitives
- Scalable



- Idempotent
- Reasonable
- Primitives
- Scalable
- Hackable



- Idempotent
- Reasonable
- Primitives
- Scalable
- Hackable
- Shareable





- Configuration management language
- Systems integration framework
- API for your infrastructure

<http://www.flickr.com/photos/morville/3220961040/>

## The Meatcloud Manifesto

Give me an API or give me death.

-- Andrew Clay Shafer (@littleidea)



Chef manages **Nodes**

Chef manages **Nodes**

**Nodes** have Attributes

Chef manages **Nodes**

**Nodes** have Attributes

**Users** and **Nodes** authenticate as **Clients**

Chef manages **Nodes**

**Nodes** have Attributes

**Users** and **Nodes** authenticate as **Clients**

**Cookbooks** contain **Recipes**

Chef manages **Nodes**

**Nodes** have Attributes

**Users** and **Nodes** authenticate as **Clients**

**Cookbooks** contain **Recipes**

Each node has a **Run List**

Chef manages **Nodes**

**Nodes** have Attributes

**Users** and **Nodes** authenticate as **Clients**

**Cookbooks** contain **Recipes**

Each node has a **Run List**

A **Run List** is a list of **Recipes** to run

Chef manages **Nodes**

**Nodes** have Attributes

**Users** and **Nodes** authenticate as **Clients**

**Cookbooks** contain **Recipes**

Each node has a **Run List**

A **Run List** is a list of **Recipes** to run

**Recipes** can be grouped in a **Role**

Chef manages **Nodes**

**Nodes** have Attributes

**Users** and **Nodes** authenticate as **Clients**

**Cookbooks** contain **Recipes**

Each node has a **Run List**

A **Run List** is a list of **Recipes** to run

**Recipes** can be grouped in a **Role**

**Roles** can also be added to a **Run List**



Chef manages **Nodes**

**Nodes** have Attributes

**Users** and **Nodes** authenticate as **Clients**

**Cookbooks** contain **Recipes**

Each node has a **Run List**

A **Run List** is a list of **Recipes** to run

**Recipes** can be grouped in a **Role**

**Roles** can also be added to a **Run List**

**Nodes** can be in **Environments**

Chef manages **Nodes**

**Nodes** have Attributes

**Users** and **Nodes** authenticate as **Clients**

**Cookbooks** contain **Recipes**

Each node has a **Run List**

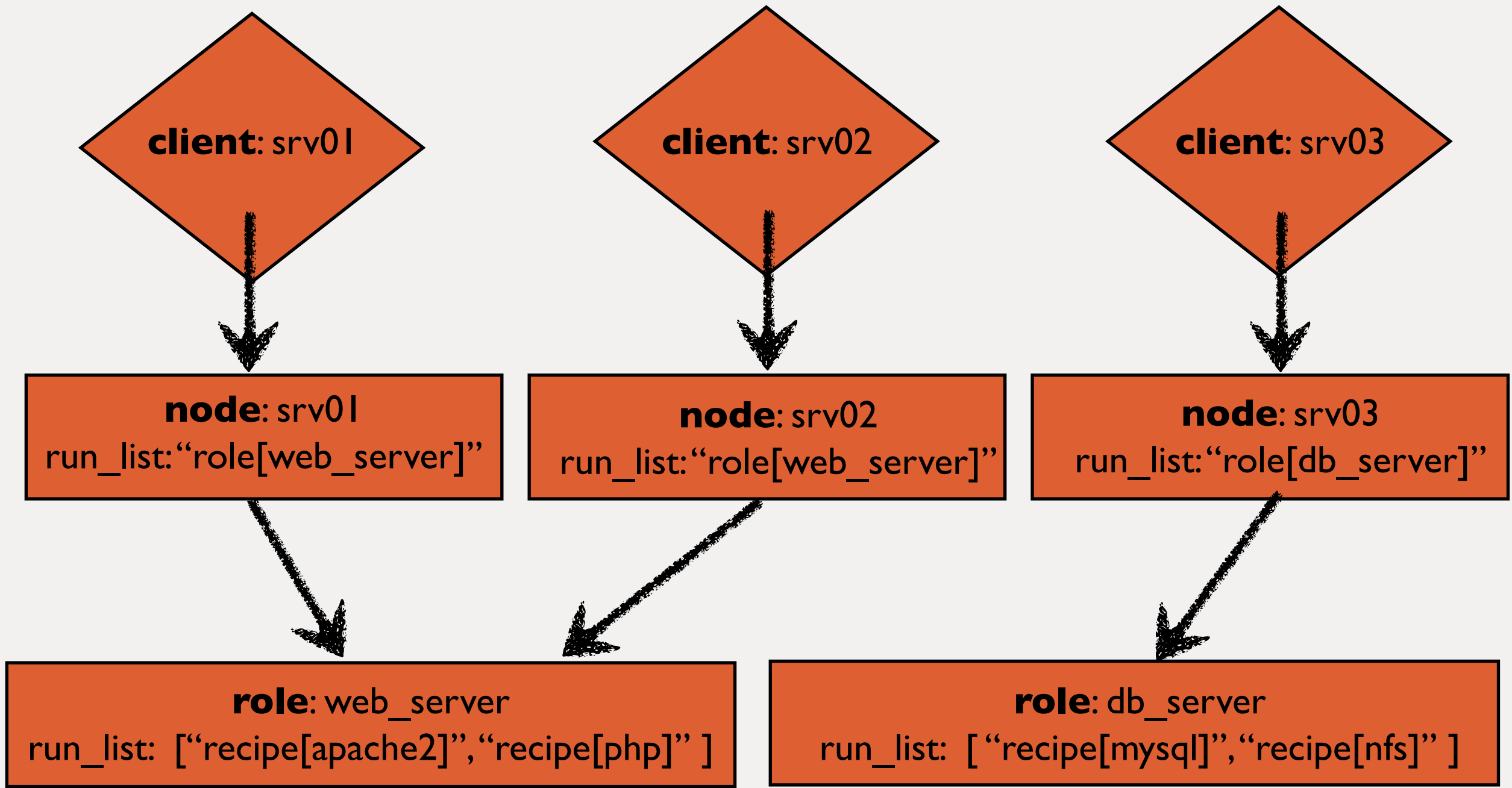
A **Run List** is a list of **Recipes** to run

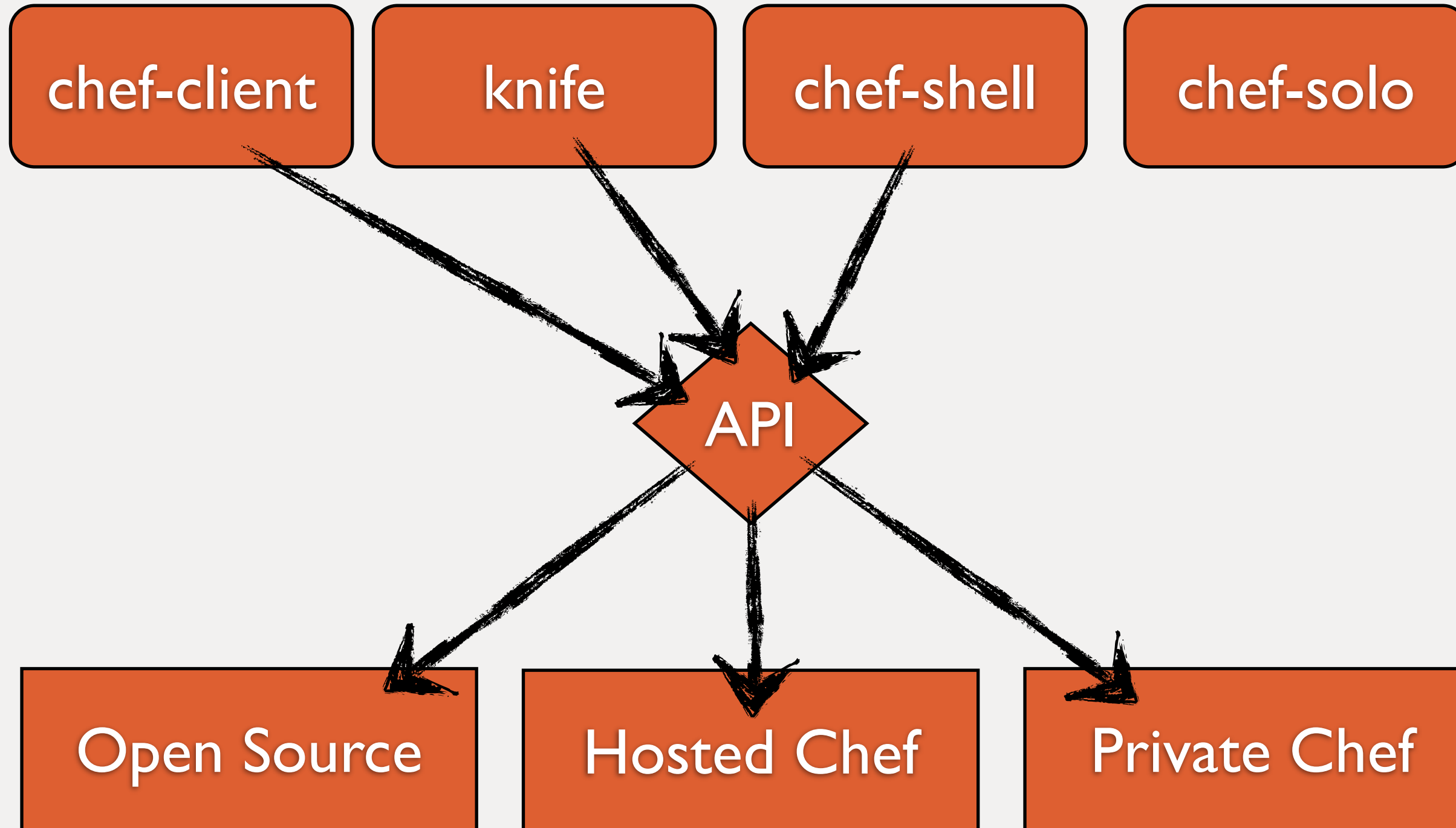
**Recipes** can be grouped in a **Role**

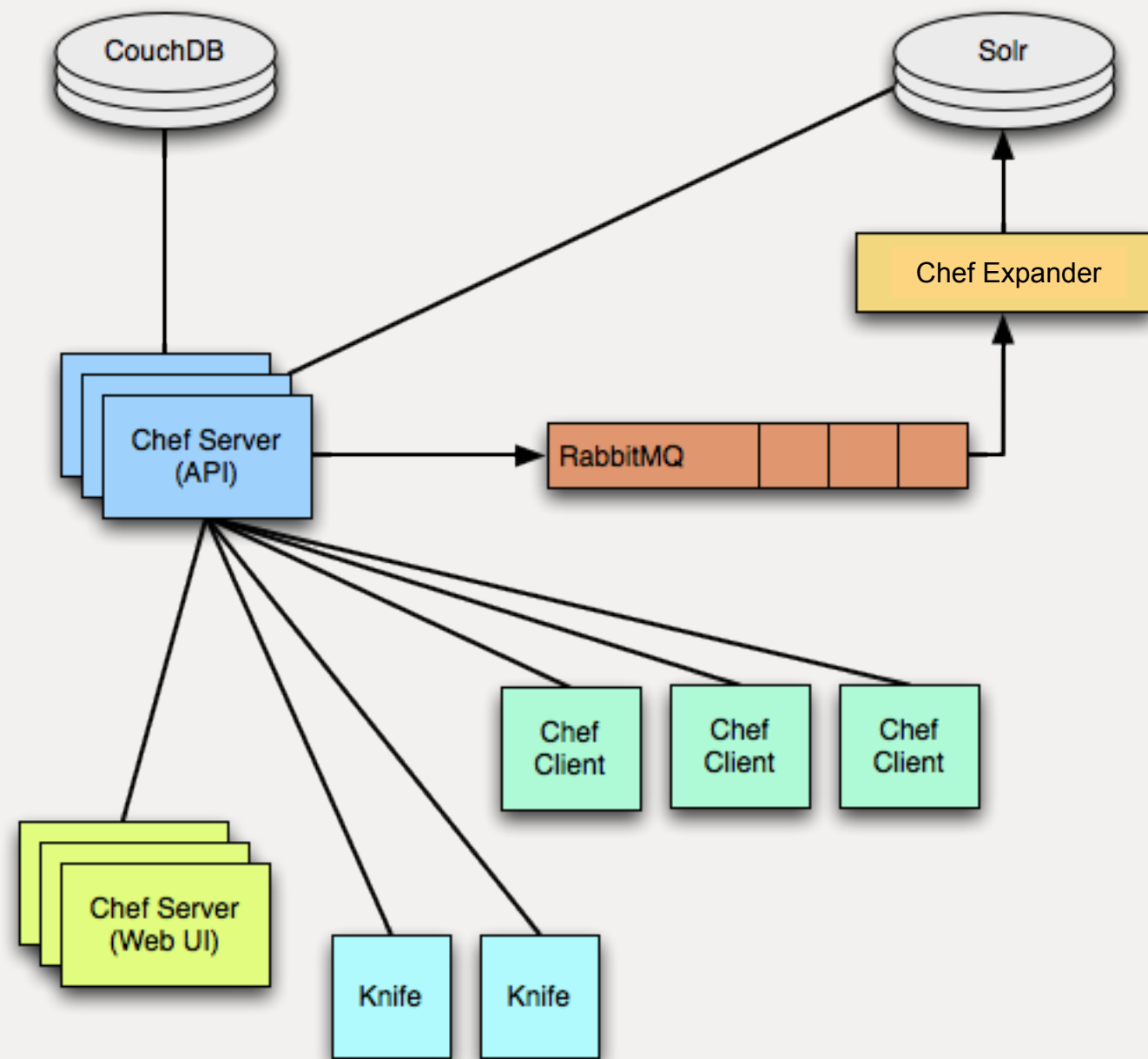
**Roles** can also be added to a **Run List**

**Nodes** can be in **Environments**

**Data bags** are... bags of data.







- A **Resource** is something you manage  
service, package, file, user, execute, git

- A **Resource** is something you manage  
service, package, file, user, execute, git
- Resources have **actions**  
start, install, create, deploy
- Resources can **notify** of other resources

- A **Resource** is something you manage  
service, package, file, user, execute, git
- Resources have **actions**  
start, install, create, deploy
- Resources can **notify** of other resources

```
cookbook_file "/etc/apache2/apache2.conf" do
  source "apache2.conf"
  owner "root"
  group "root"
  mode 0644
  notifies :restart, "service[apache2]"
end
```



- A **Provider** performs the actions specified by the resource

- A **Provider** performs the actions specified by the resource
- Each Resource can have multiple providers  
package: apt, yum, macports...  
service: upstart, windows, systemd...

- A **Provider** performs the actions specified by the resource
- Each Resource can have multiple providers  
package: apt, yum, macports...  
service: upstart, windows, systemd...
- Each platform (OS) has default Providers that can be overridden

```
package "sudo" do
  provider Chef::Provider::Package::Yum
  action :install
end
```

```
package "apache2" do
  action :install
end
```

```
package "apache2" do
  action :install
end

service "apache2" do
  action :enable
end
```

```
package "apache2" do
  action :install
end

service "apache2" do
  action :enable
end

cookbook_file "/etc/apache2/apache2.conf" do
  source "apache2.conf"
  owner "root"
  group "root"
  mode 0644
end
```

```
package "apache2" do
  action :install
end

service "apache2" do
  action :enable
end

cookbook_file "/etc/apache2/apache2.conf" do
  source "apache2.conf"
  owner "root"
  group "root"
  mode 0644
end

service "apache2" do
  action :start
end
```

```
package "apache2" do
  action :install
end

service "apache2" do
  action :enable
  supports [ :restart, :reload, :status ]
end

cookbook_file "/etc/apache2/apache2.conf" do
  source "apache2.conf"
  owner "root"
  group "root"
  mode 0644
  notifies :restart, "service[apache2]"
end

service "apache2" do
  action :start
end
```







# Nagios Demo

- Download server cookbooks
- Install server
- Create fake nodes
- Update server



OPSCODE

# Questions?

There is lots more to  
learn about Chef at  
<http://wiki.opscode.com>