Chef
Automation on the Cloud

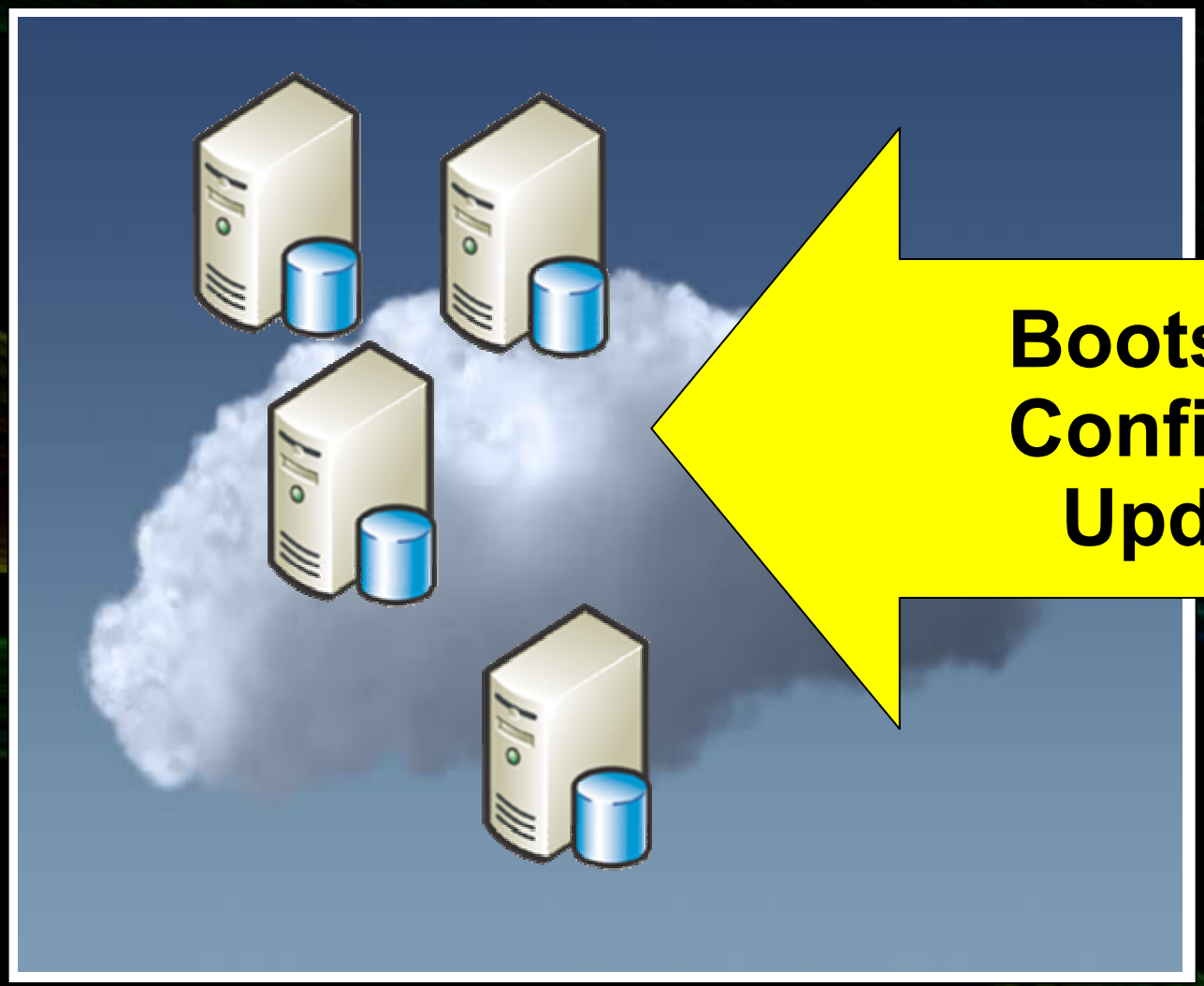
Mike Scherbakov

Saratov State University,

Grid Dynamics

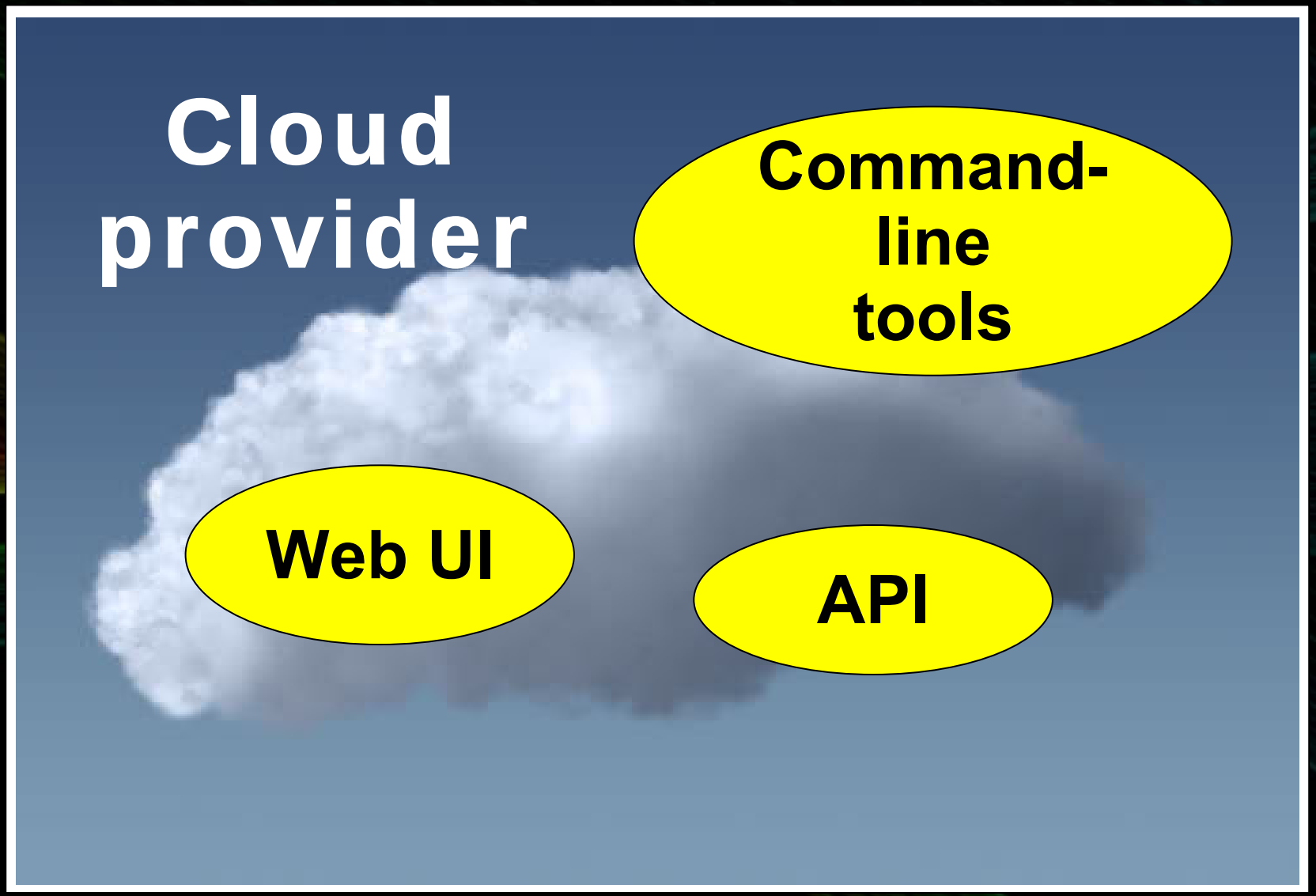
2010, April 15-16

Deployment



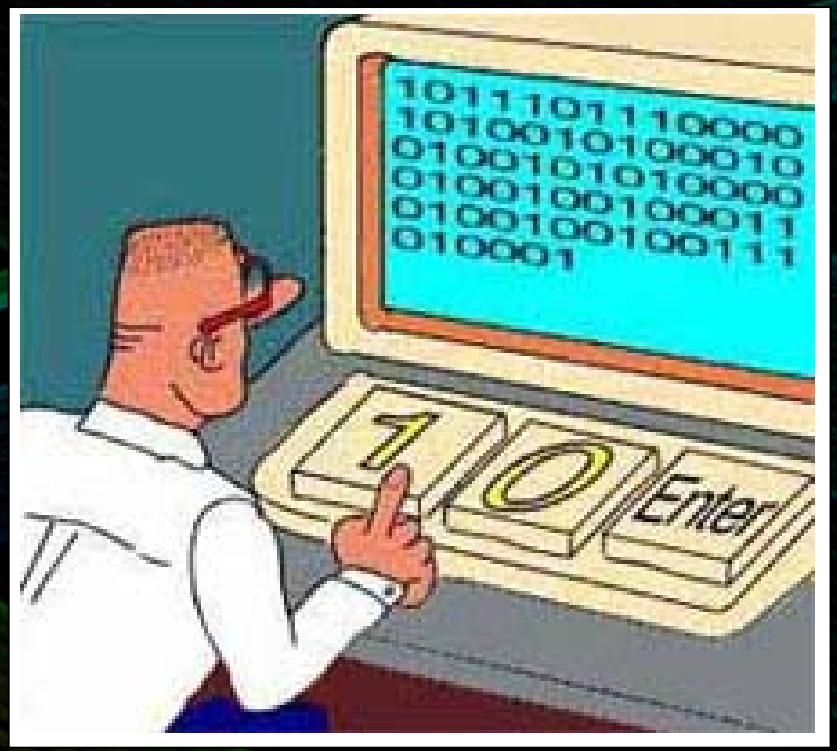
Bootstrap
Configure
Update

Bootstrap instance



Configure

Manually



by Scripts

- Shell
- Python
- Ruby
- Capistrano

by Scripts, but…

NOT scalable

NOT idempotent

DEPENDENT to *NIX distribution

Idempotency

Chef

Idempotent resources

Easy to collaborate

Configuration? Programming!

Chef operates with…

- Recipes
- Templates
- Attributes
- Files

Resources

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

Abstraction

Define required state

1+ providers

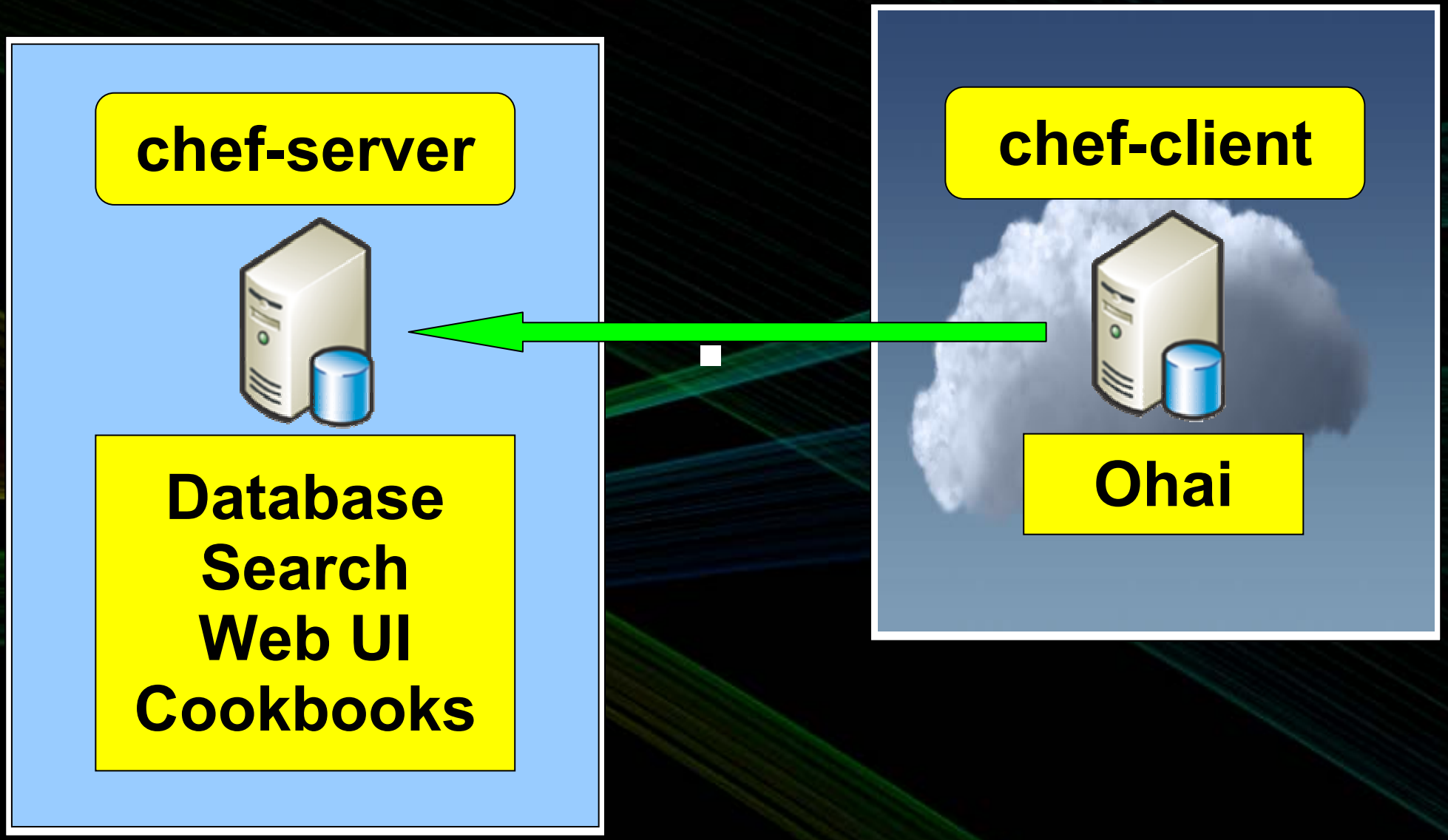
Attributes

```
apache[:listen_ports] = [ 8080 ]
```

Related to the node

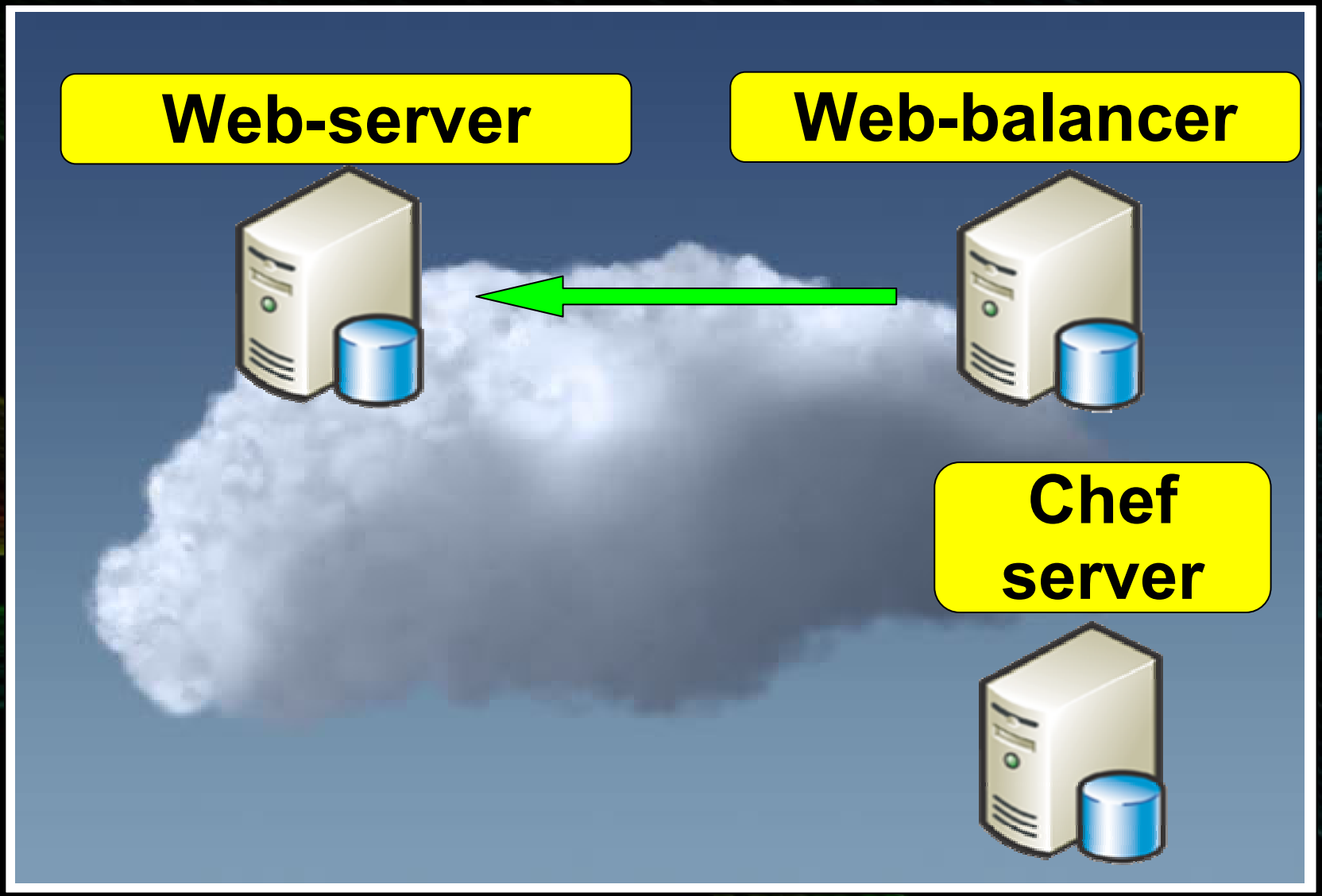
Indexing

Architecture #1

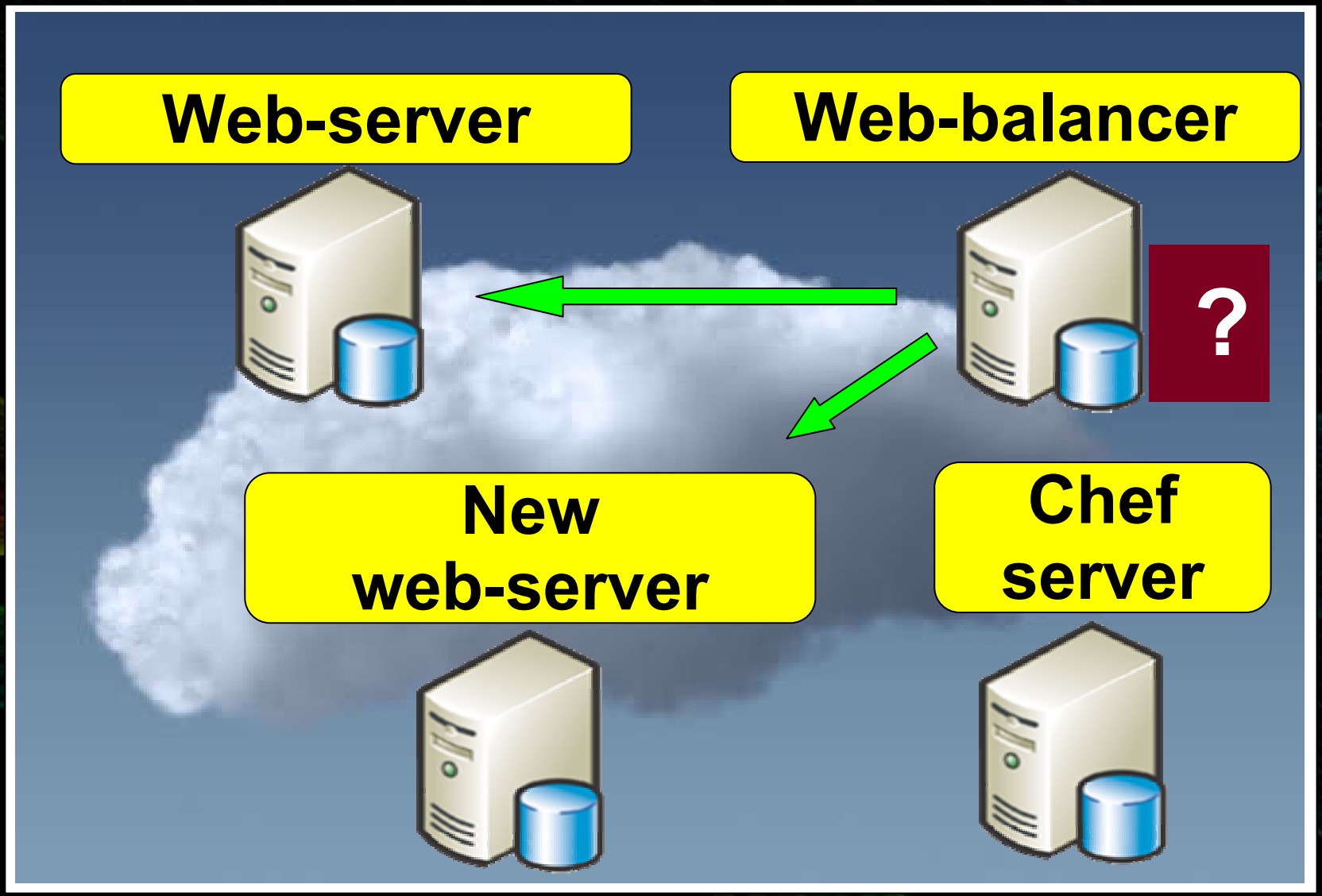


Architecture #2

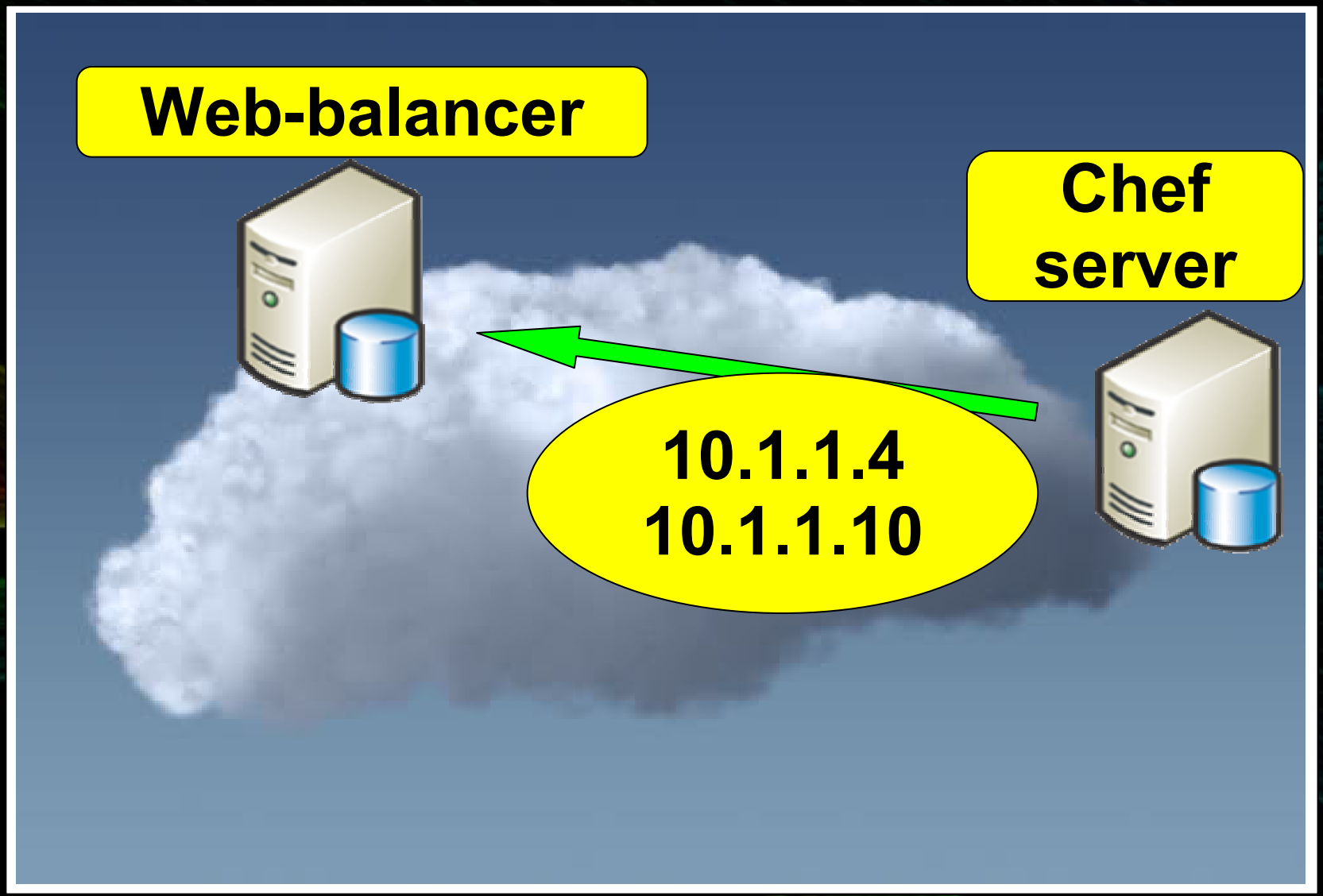
Example



Add new server



Update configuration



Recipes

```
service "haproxy" do
  action [:enable, :start]
end

webs = search(:node,"recipe:simple-httpd").map
{ |w| [ w["ipaddress"], w["fqdn"] ] }

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy.cfg.erb"
  variables(:webs => webs)
  notifies :restart, resources(:service =>
"haproxy")
end
```

Workflow #1

Client node

validate.pem

Chef server

Install chef-client

Workflow #2

Write recipe

Assign it to the node

Chef server

Client node:

Run chef-client

Alternatives

- SmartFrog
- Puppet
- CFEngine

Thank you for attention!

.

Mike Scherbakov
mscherbakov@griddynamics.com

Saratov State University,
Grid Dynamics

2010, April 15-16