

# Overview of Chef

What is this thing again?



- Understand how Chef thinks about Infrastructure Automation
- Understand the following concepts:
  - Node
  - Resource
  - Recipe
  - Cookbook
  - Run List
  - Roles
  - Search

# Evolving towards Configuration Management

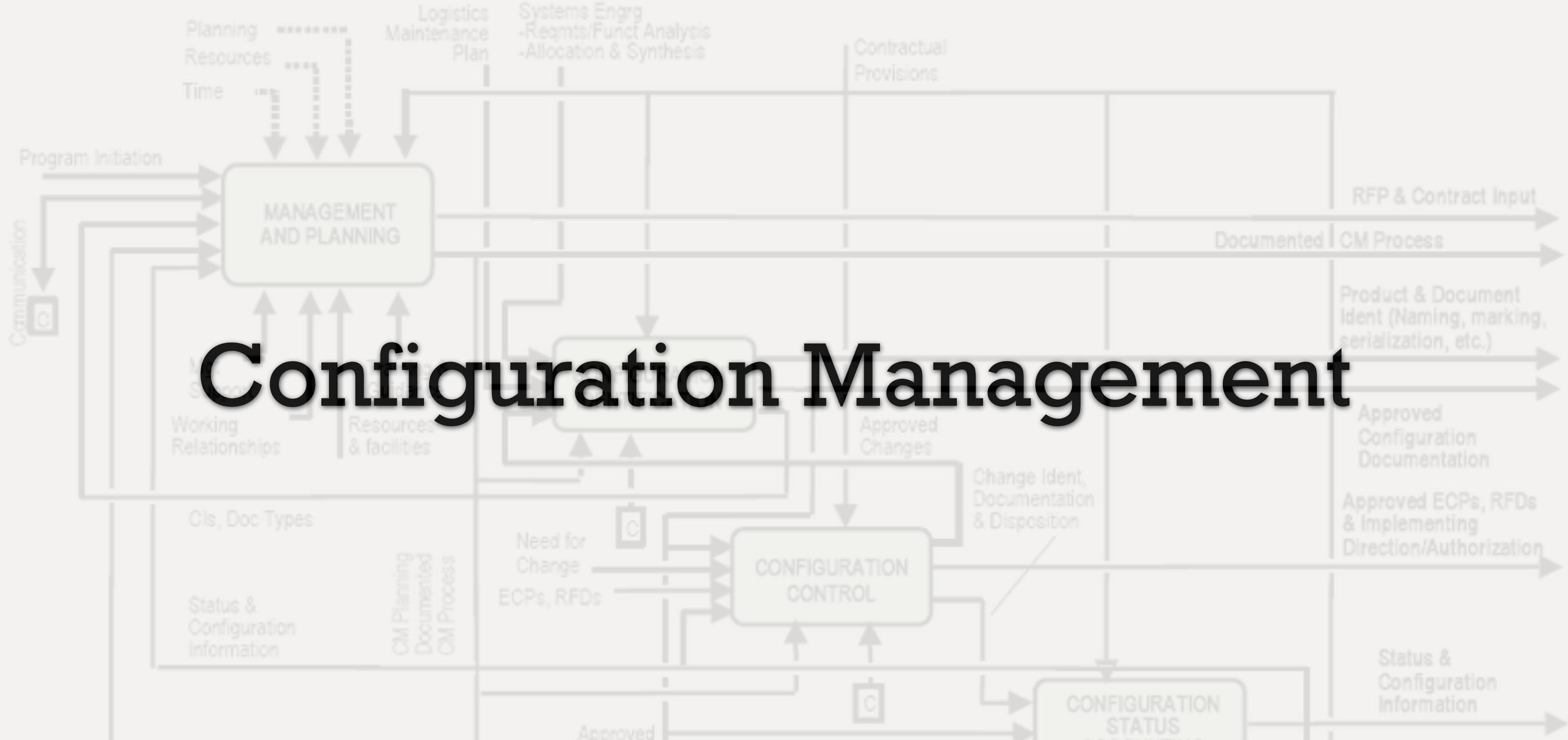
- Just build it
- Keep notes in server.txt
- Move notes to the wiki
- Custom scripts (in scm?!)
- Snapshot & Clone

RULE THE CLOUD

# Part of the Software Development Lifecycle

- Dev and Test infrastructure should match Production as closely as possible
- Applications are complex!

RULE THE CLOUD



# Configuration Management

# System Integration

<http://www.flickr.com/photos/opalsson/3773629074/>

# Applications



<http://www.flickr.com/photos/steffenz/337700069/>  
<http://www.flickr.com/photos/kky/704056791/>

# Infrastructure



RULE THE CLO

<http://www.flickr.com/photos/sbh/462754460/>

# Collection of Resources

- Nodes
- Networking
- Files
- Directories
- Symlinks
- Mounts
- Routes
- Users
- Groups
- Tasks
- Packages
- Software
- Services
- Configurations
- Stuff



<http://www.flickr.com/photos/stevekeys/3123167585/>

# Acting in concert



RULE THE

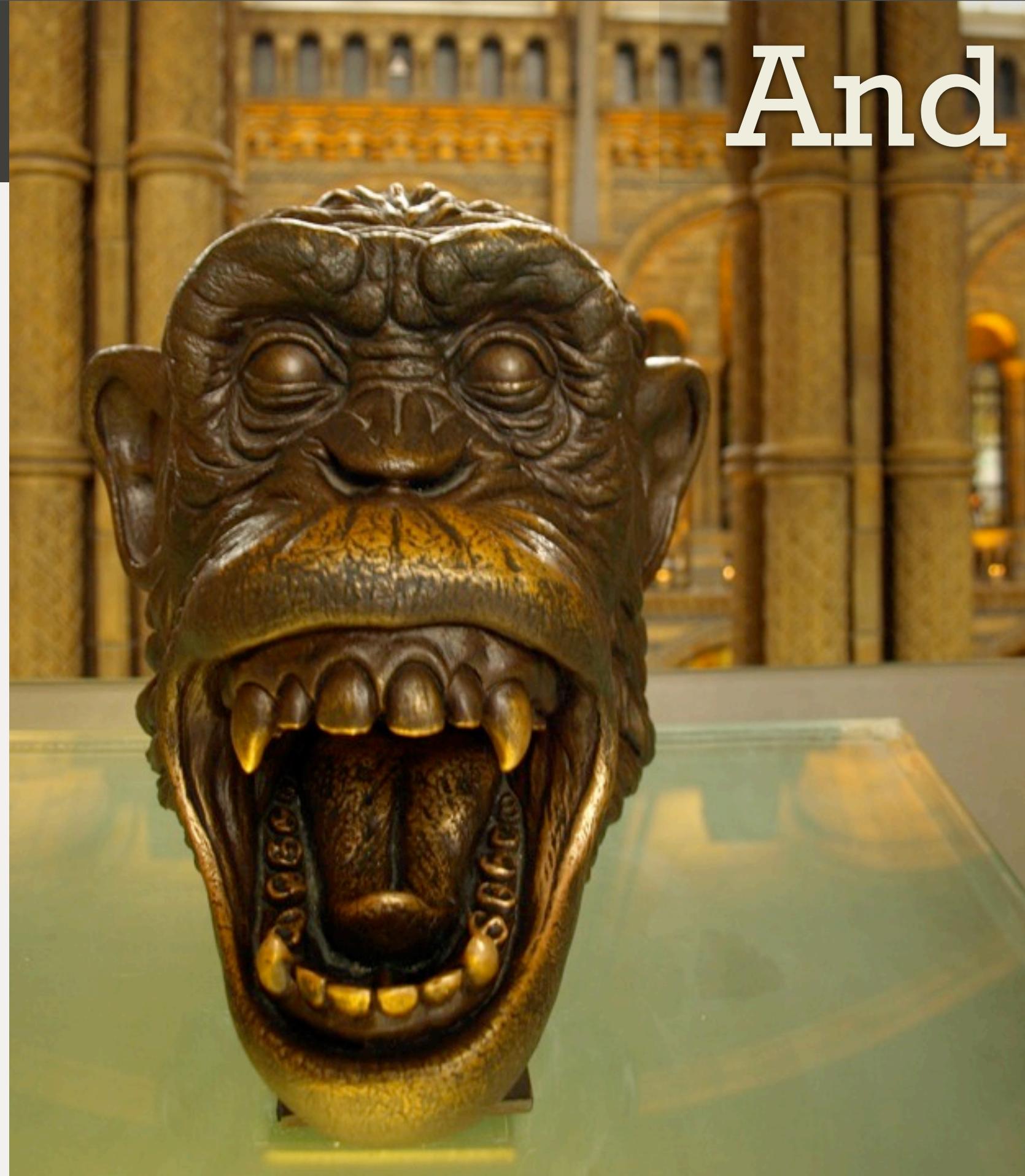
<http://www.flickr.com/photos/glowjangles/4081048126/>

# To provide a Service



<http://www.flickr.com/photos/28309157@N08/3743455858/>

# And it evolves

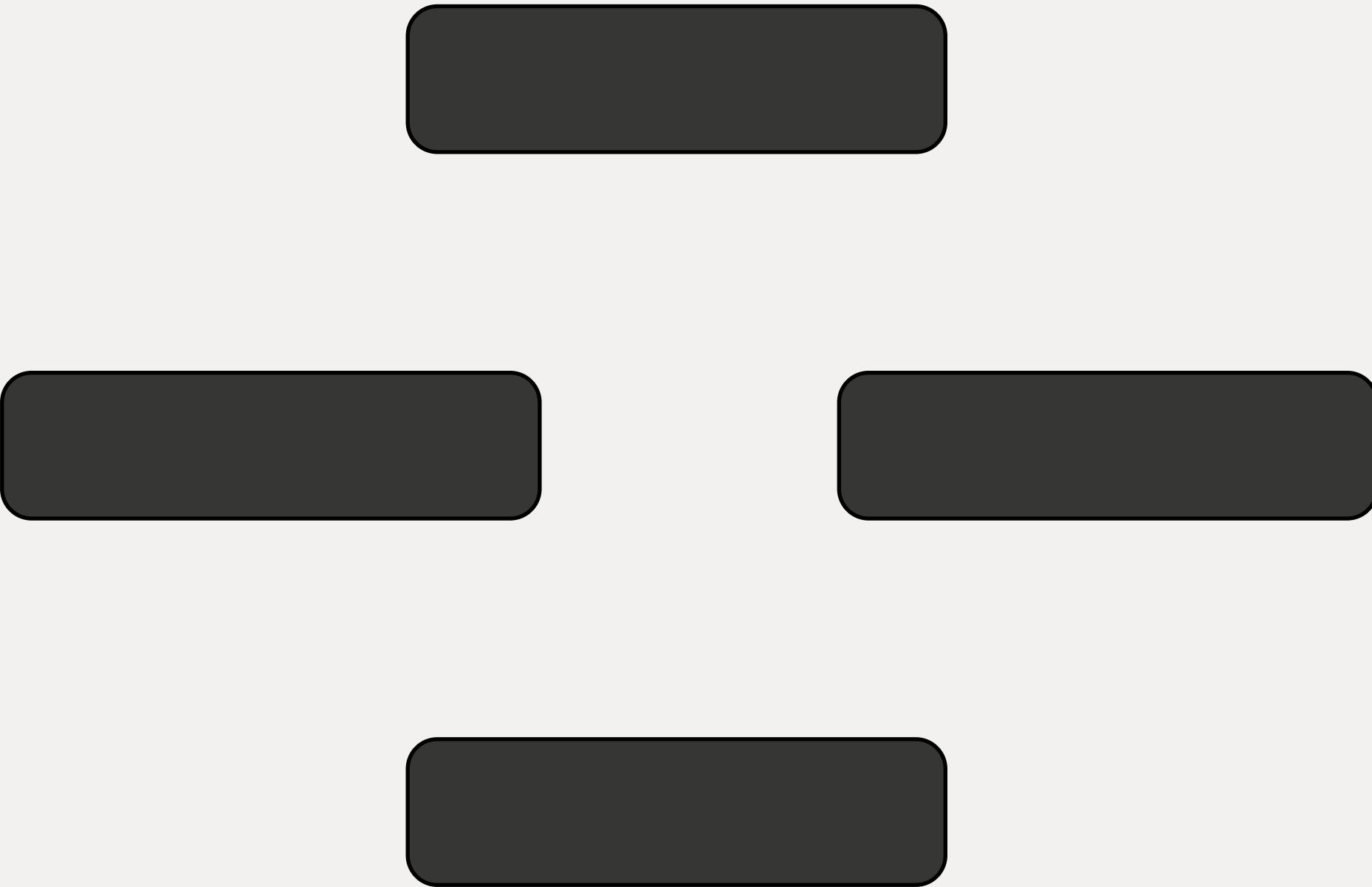


<http://www.flickr.com/photos/16339684@N00/2681435235/>

# n-Tier Infrastructure



# n-Tier Infrastructure



- Provision

# n-Tier Infrastructure

Load Balancer

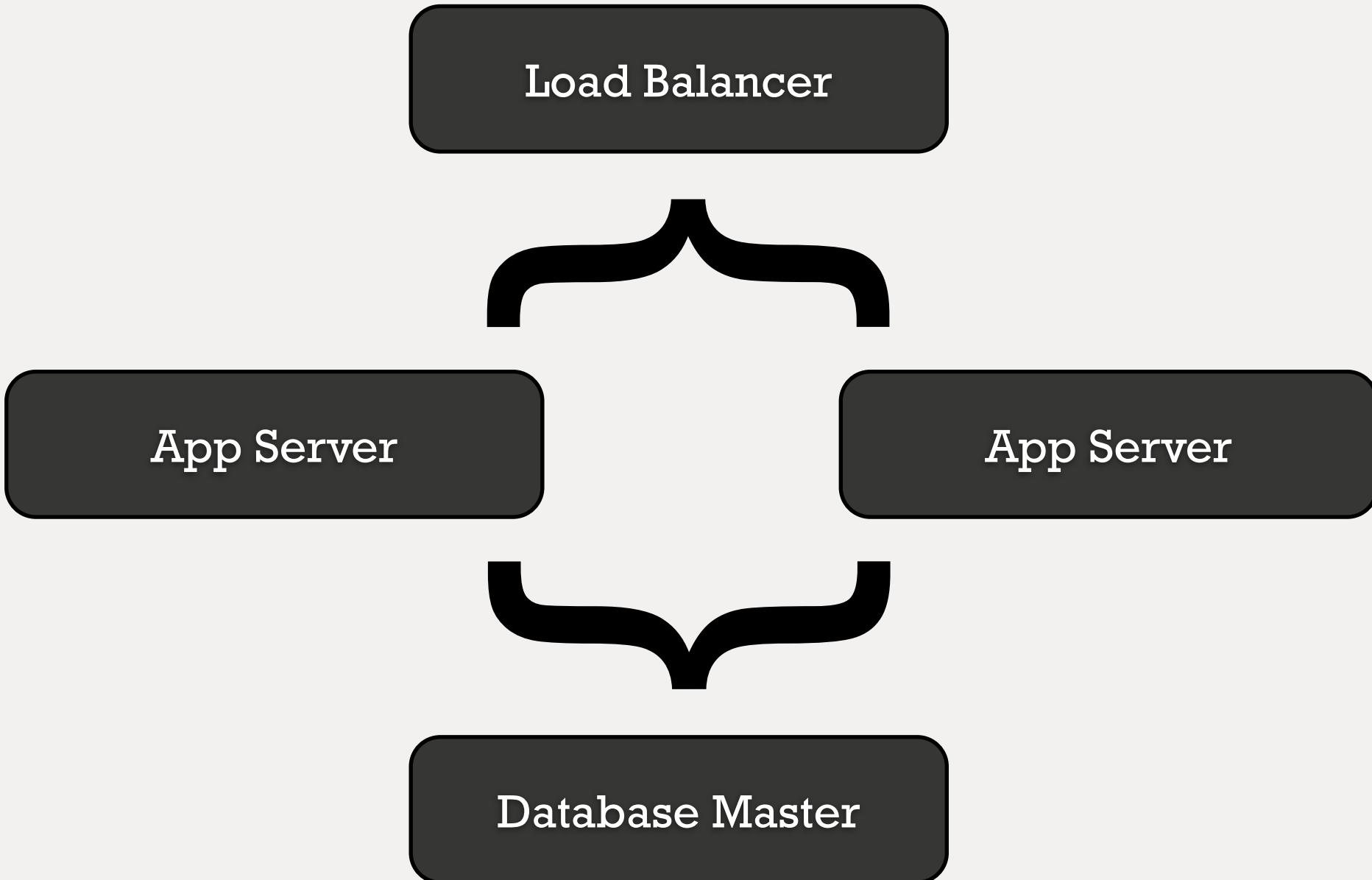
App Server

App Server

Database Master

- Provision
- Configure

# n-Tier Infrastructure

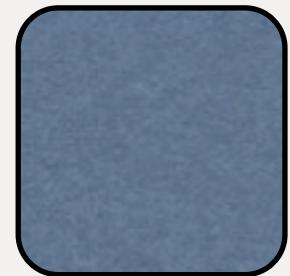


- Provision
- Configure
- Integrate

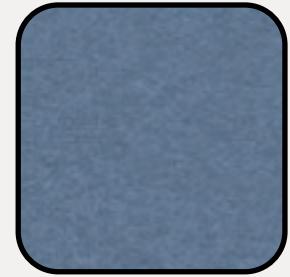


Application

# See Nodes

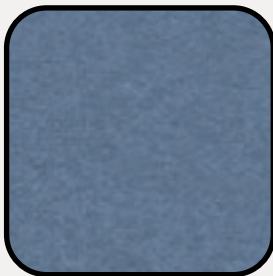


Application

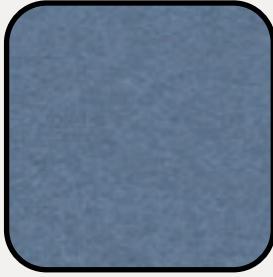
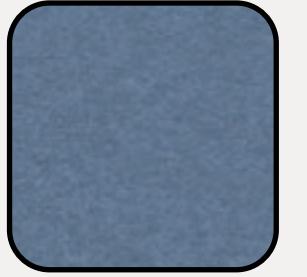


Application Database

# See Nodes Grow

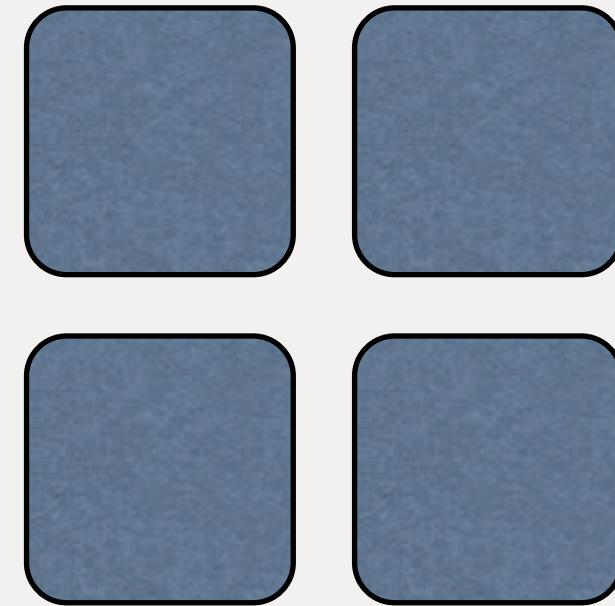


Application



App Databases

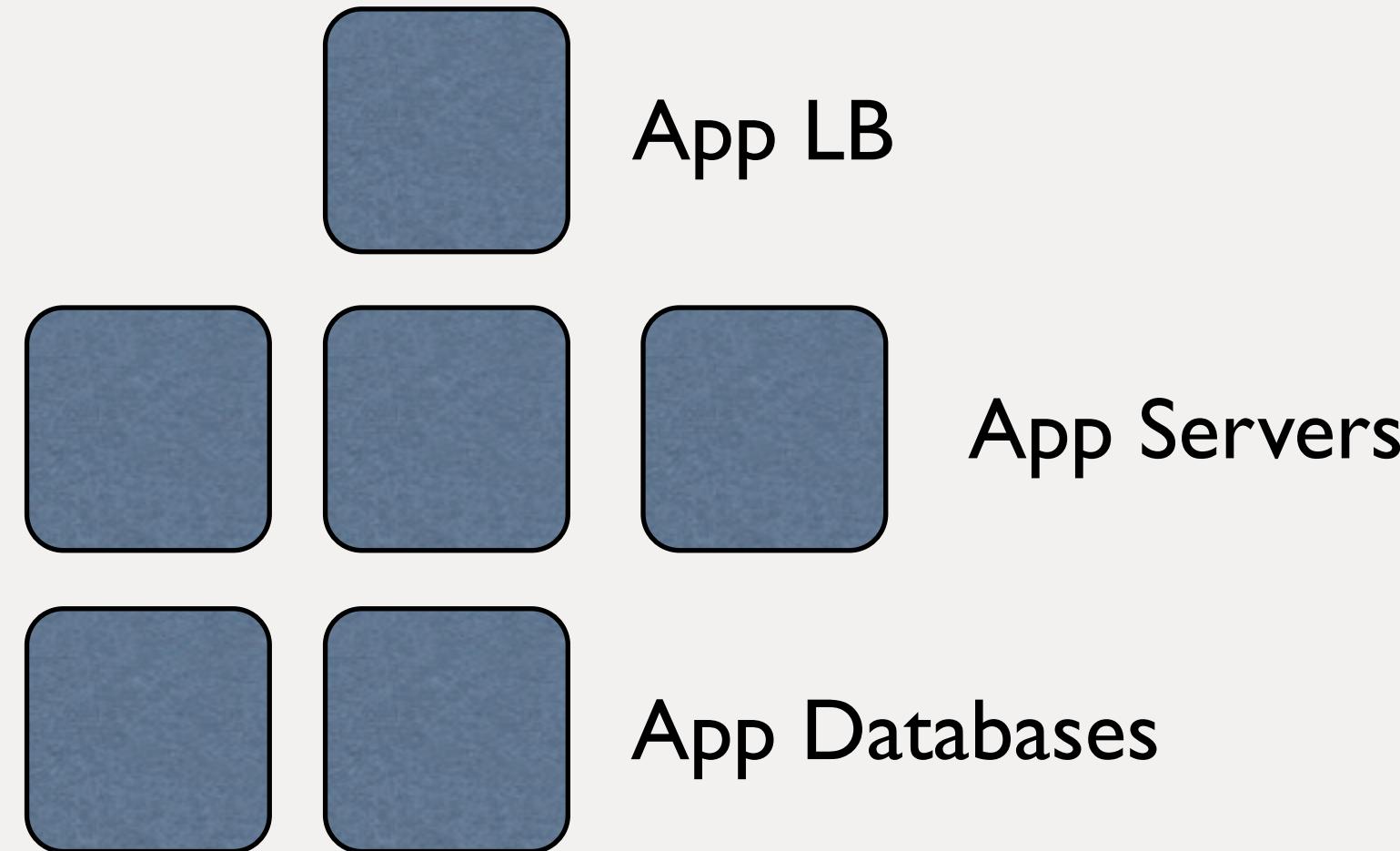
# See Nodes Grow



App Servers

App Databases

# See Nodes Grow

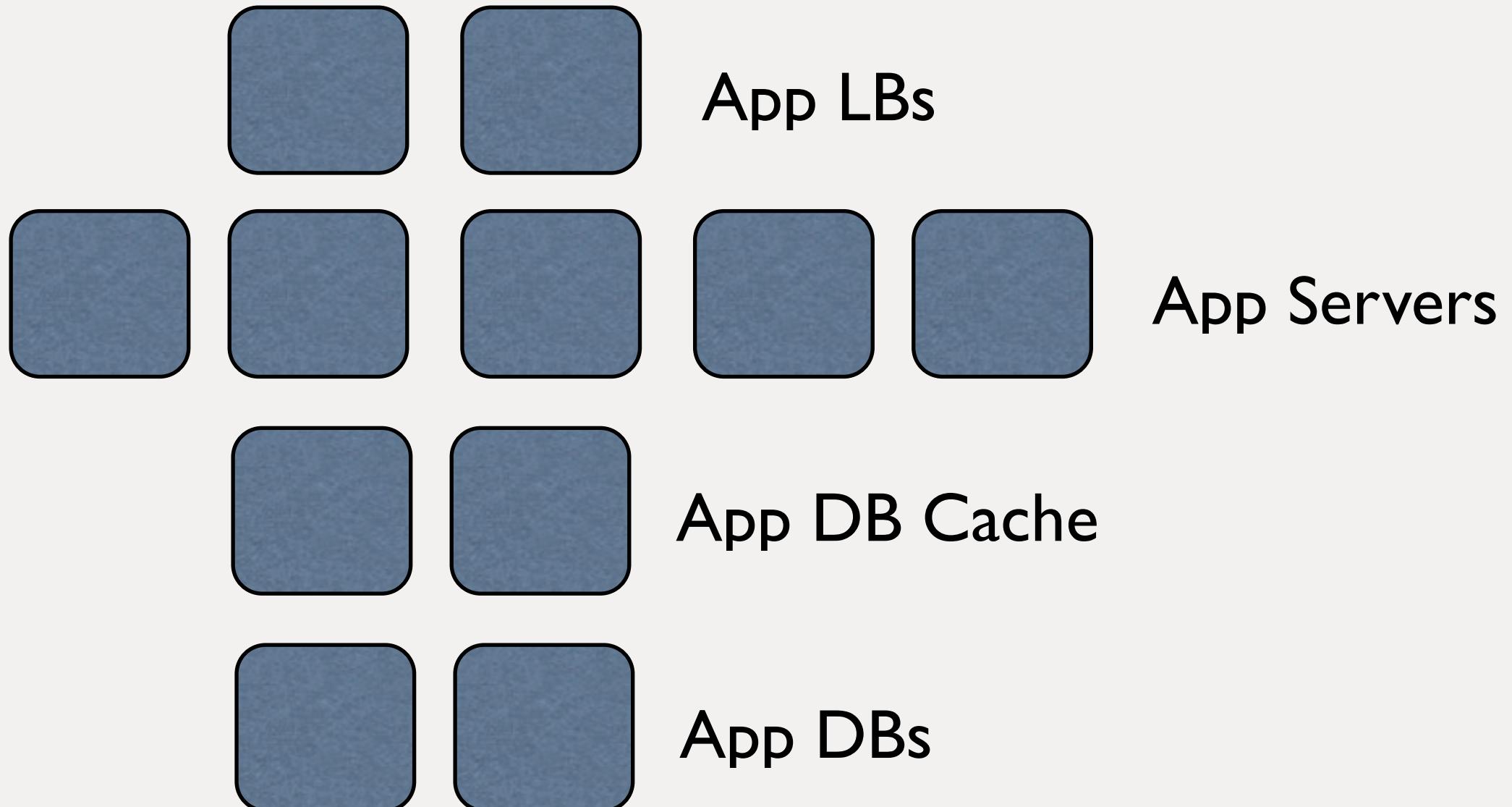


# See Nodes Grow



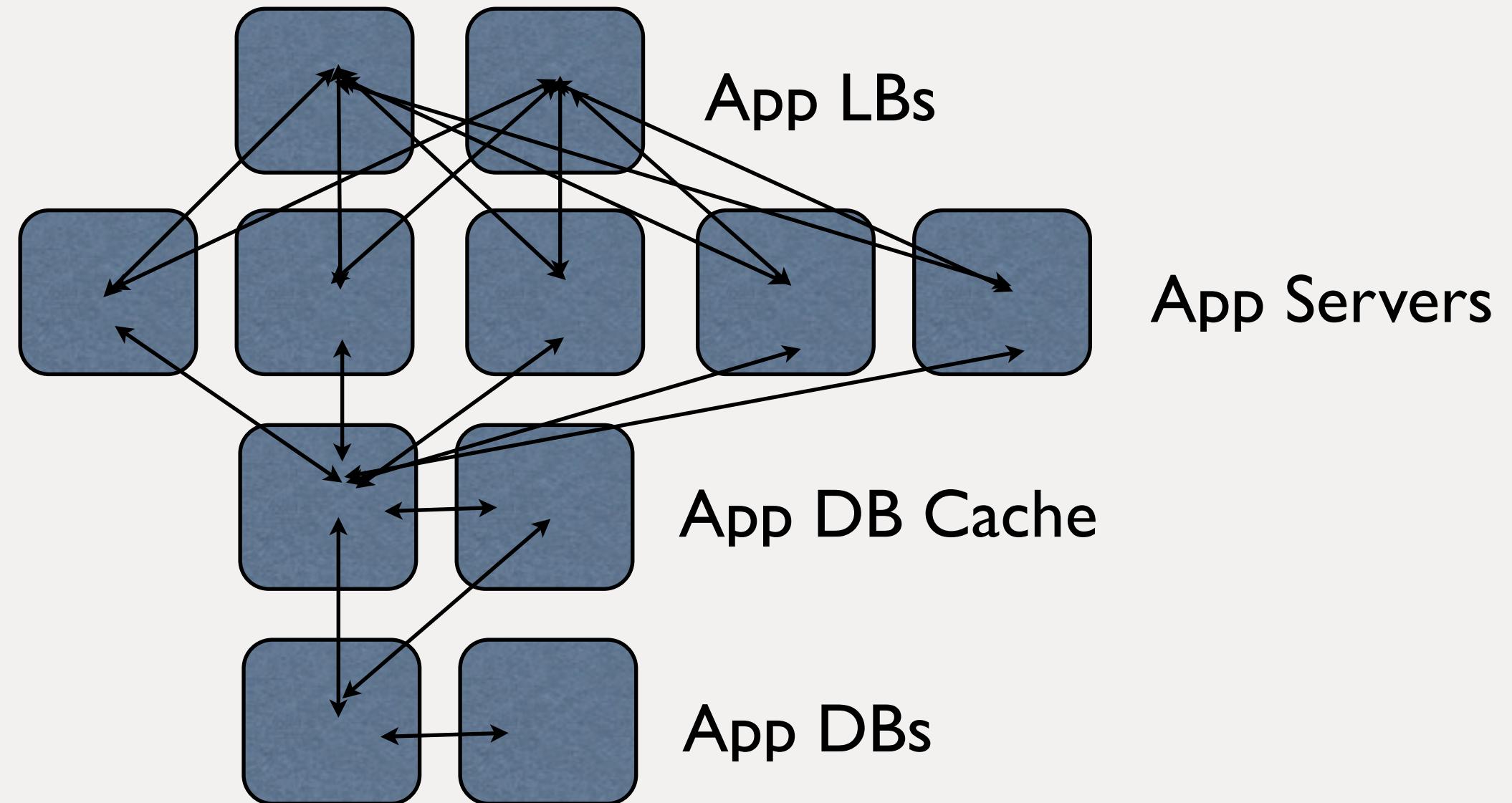
RULE THE CLOUD

# See Nodes Grow

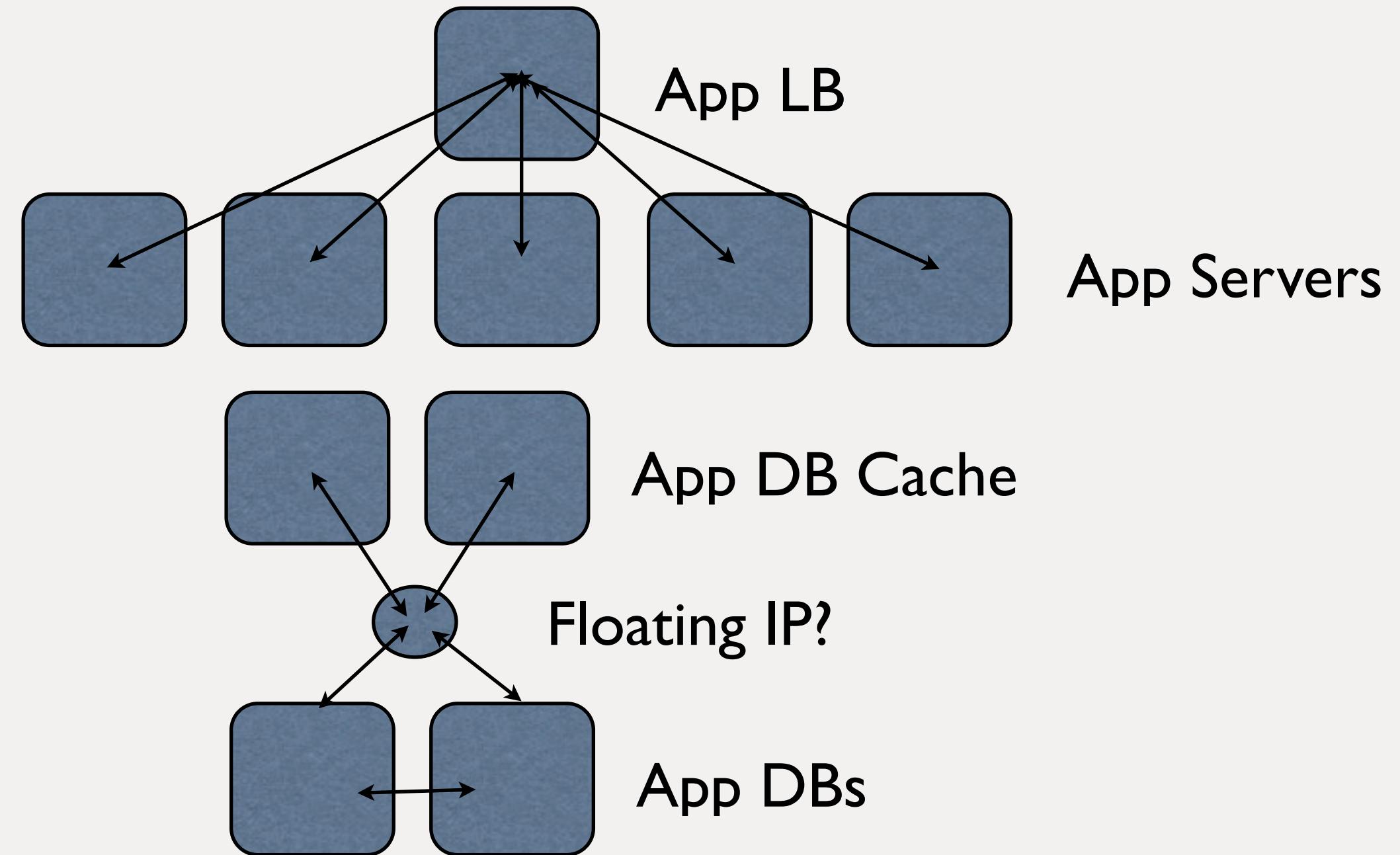


RULE THE CLOUD

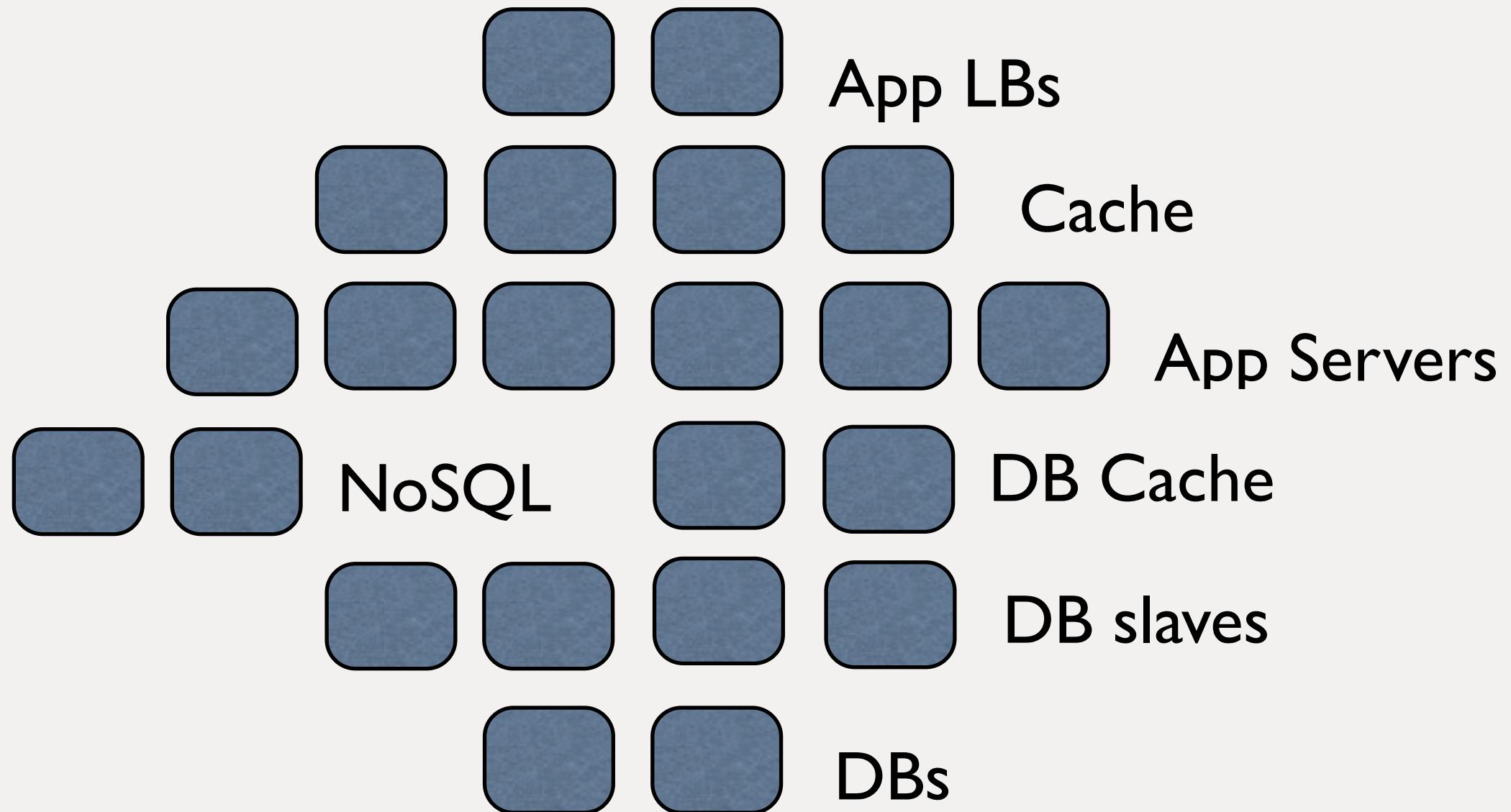
# Stitched together with configs



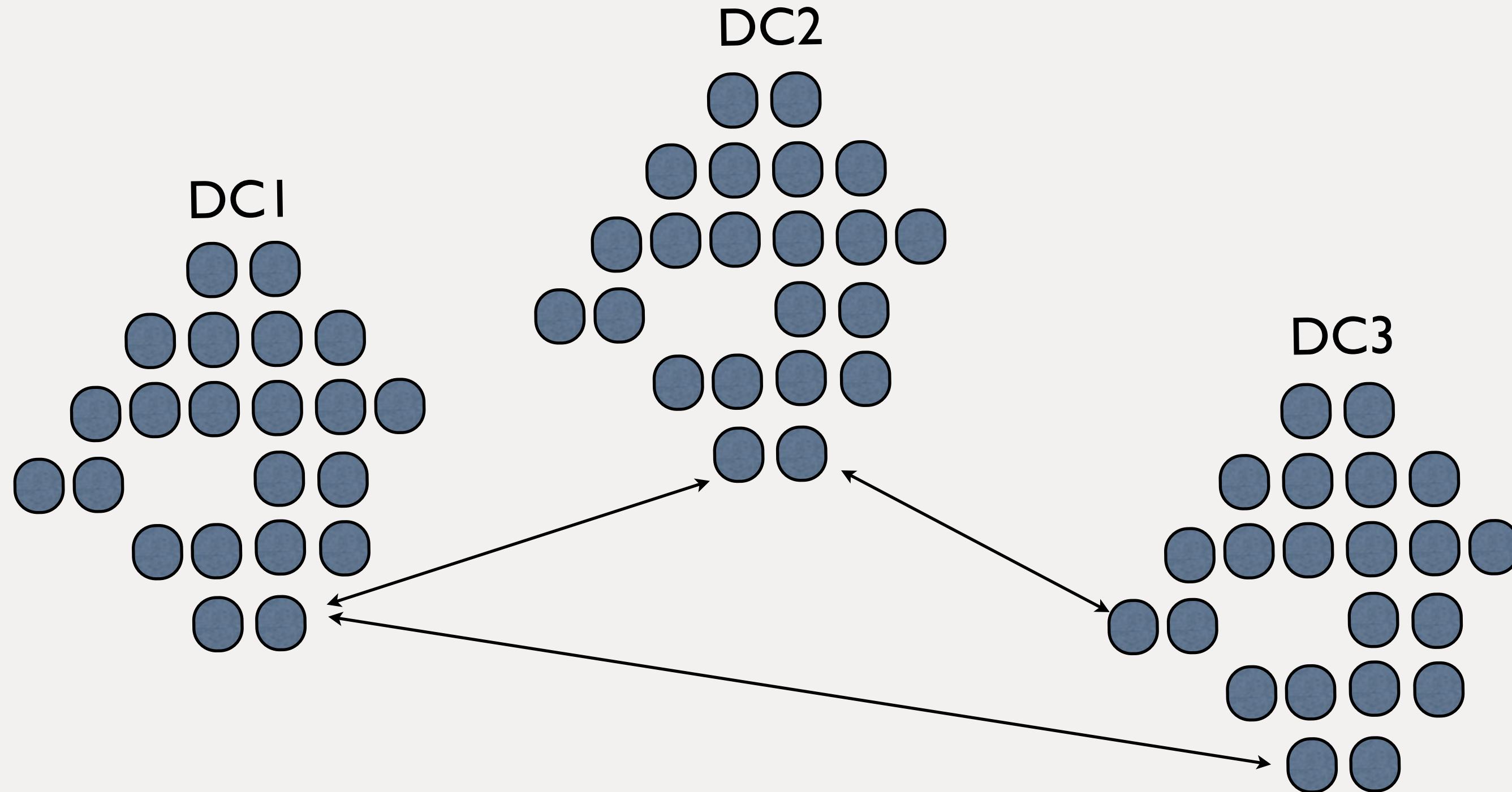
# Your Infrastructure is a snow flake



# Complexity increases quickly



# Complexity increases very quickly





# Programs!

- Generate configurations directly on nodes
- Reduce management complexity
- Version control the programs

<http://www.flickr.com/photos/ssoosay/5126146763/>

# Declarative Interface to Resources

- Define policy
- Say what, not how
- Pull not Push



# Recipes and Cookbooks

- Recipes are collections of Resources
- Cookbooks contain recipes, templates, files, custom resources, etc
- Code re-use and modularity



<http://www.flickr.com/photos/shutterhacks/4474421855/>

# Resources

# Declarative interface to system resources

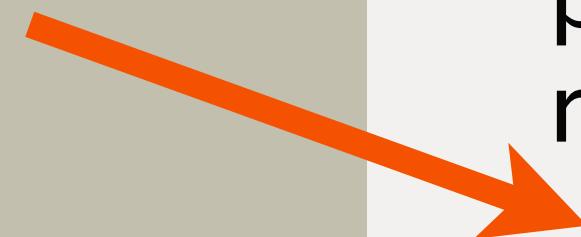
```
user "django_app" do
  shell "/bin/false"
  comment "Django App User"
  system true
  action :create
end

package "python" do
  action :install
end

python_pip "gunicorn" do
  action :install
end
```

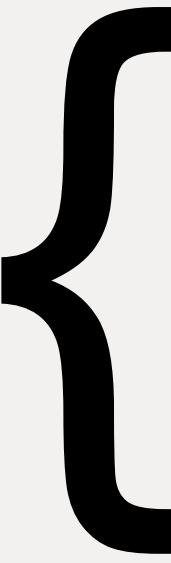
**Describe \*what\*.**

**Not how.**



```
def install_package(name, version)
  package_name = "#{name}=#{version}"
  package_name = name if @is_virtual_package
  run_command_with_systems_locale(
    :command => "apt-get -q -y
      #{expand_options(@new_resource.options)}
      install #{package_name}",
    :environment => {
      "DEBIAN_FRONTEND" => "noninteractive"
    }
  )
end
```

**package “python”**



```
yum install python  
apt-get install python  
pacman sync python  
pkg_add -r python
```

# Recipes

# Ruby Internal Ruby DSL Ruby

```
user "django_app" do
  shell "/bin/false"
  comment "Django App"
  system true
end

package "python"

python_pip "gunicorn" do
  action :install
end
```

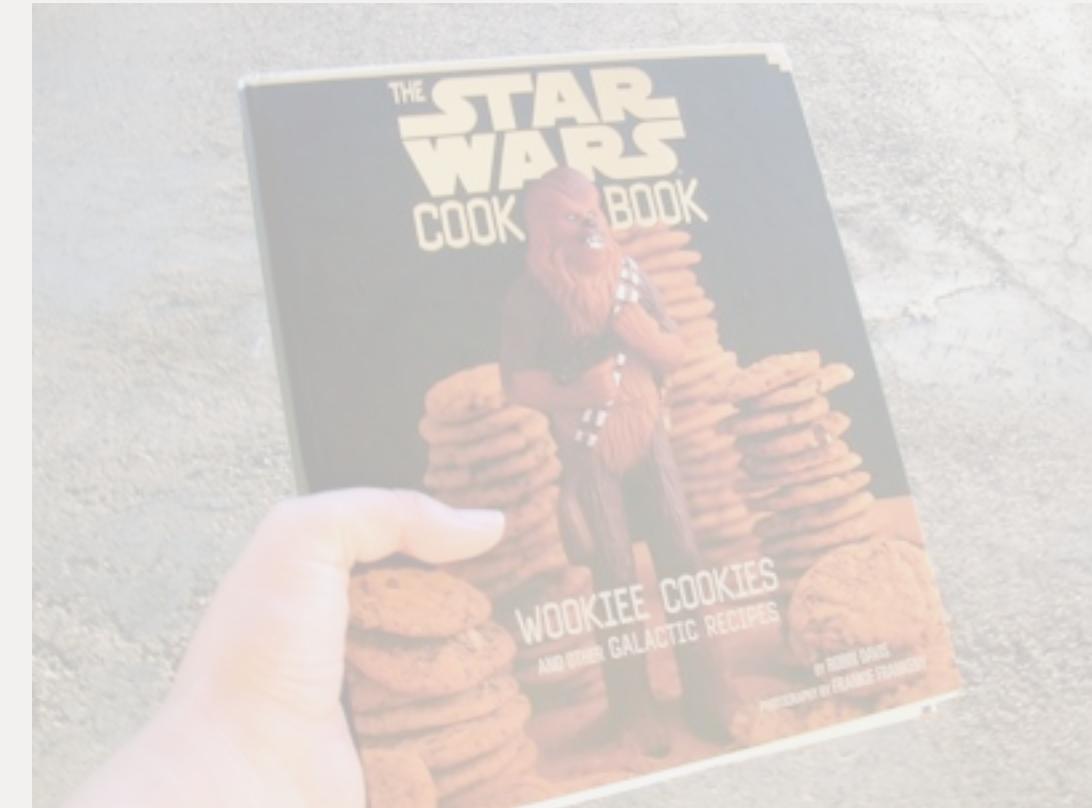
# Cookbooks



OPSCODE

RULE THE CLOUD

```
opscode/cookbooks/python
├── README.md
├── attributes
│   └── default.rb
├── metadata.rb
├── providers
│   ├── pip.rb
│   └── virtualenv.rb
├── recipes
│   ├── default.rb
│   ├── package.rb
│   ├── pip.rb
│   ├── source.rb
│   └── virtualenv.rb
└── resources
    ├── pip.rb
    └── virtualenv.rb
```



# Roles

# Roles describe nodes.

```
name "django_cms"
description "django app app
server"
run_list(
  "recipe[mysql::client]",
  "recipe[application]"
)
```

# Roles contain recipes

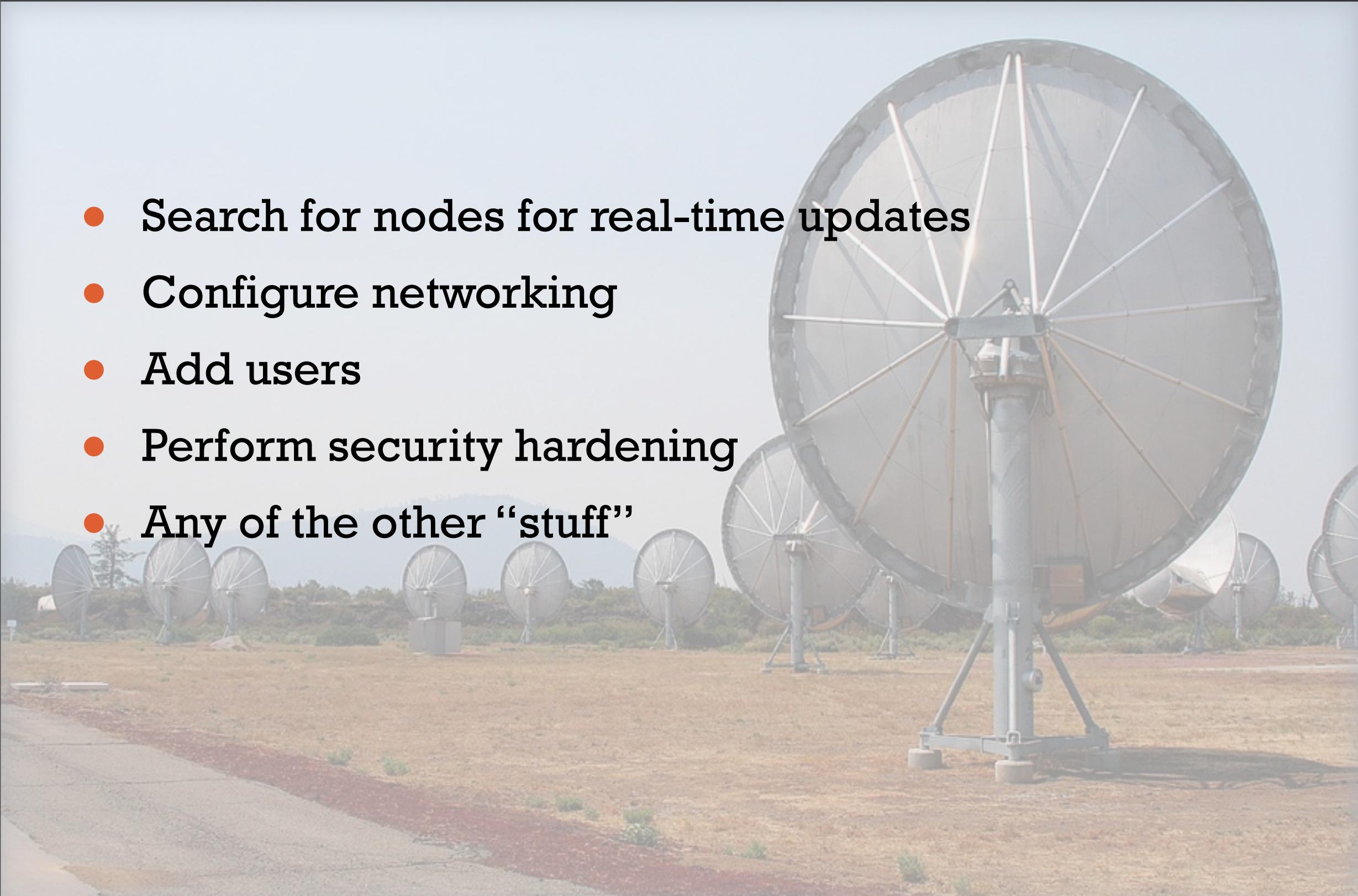
```
name "base"
description "All nodes have the base role"
run_list(
  "recipe[zsh]",
  "recipe[sudo]",
  "recipe[apt]",
  "recipe[git]",
  "recipe[build-essential]"
)
override_attributes(
  :authorization => {
    :sudo => {
      :users => ["ubuntu"],
      :passwordless => true
    }
  }
)
```

# Application Deployment

```
git "/srv/django_app" do
  repository "git://github.com/me/django_app.git"
  reference "master"
  action :sync
end
```

```
deploy_revision "/srv/django_app" do
  revision "2.0.17"
  repository "git://github.com/me/django_app.git"
  user "django_app"
  group "www-data"
  before_migrate do
    requirements_file = "#{release_path}/
requirements.txt"
    execute "pip install -r #{requirements_file}"
  end
  action :deploy
end
```

- Search for nodes for real-time updates
- Configure networking
- Add users
- Perform security hardening
- Any of the other “stuff”



```
file_cache_path "/var/chef-solo"  
cookbook_path "/home/ubuntu/cookbooks"
```

```
{  
  "run_list": ["recipe[hackademy]"]  
}
```

chef-solo: cookbooks/hackademy/

```
$ mkdir -p ~/cookbooks/hackademy/recipes
```



OPSCODE

## chef-solo: cookbooks/hackademy/recipes/default.rb

```
directory "/usr/local/myapp" do
end
git "/usr/local/myapp" do
  repository "git://github.com/YOU/REPO.git"
  reference "master"
  action :sync
end
```

RULE THE CLOUD

## chef-solo: running chef-solo

```
$ sudo chef-solo -c ~/solo.rb -j ~/node.json
```

RULE THE CLOUD