

Beginner Chef Antipatterns

Julian C. Dunn
Senior Consultant
Opscode, Inc.





Oops, there was a problem.
Our engineers have been notified and we will fix it.

I learned Chef the hard way



ASCII Addison Wesley Programming Series

Fascicle 4

Generating All Trees
History of Combinatorial Generation

The Art of Computer Programming 4

DONALD E. KNUTH 著

有澤 誠／和田 英一 監訳
見一彦／小出 洋 訳

Chef can have a steep learning curve

... which we try to mitigate

- learnchef.com
- docs.opscode.com
- Opscode Public/Private training classes
- Podcasts (Food Fight Show, etc.)
- Local user groups
- ChefConf! (and the hallway track)



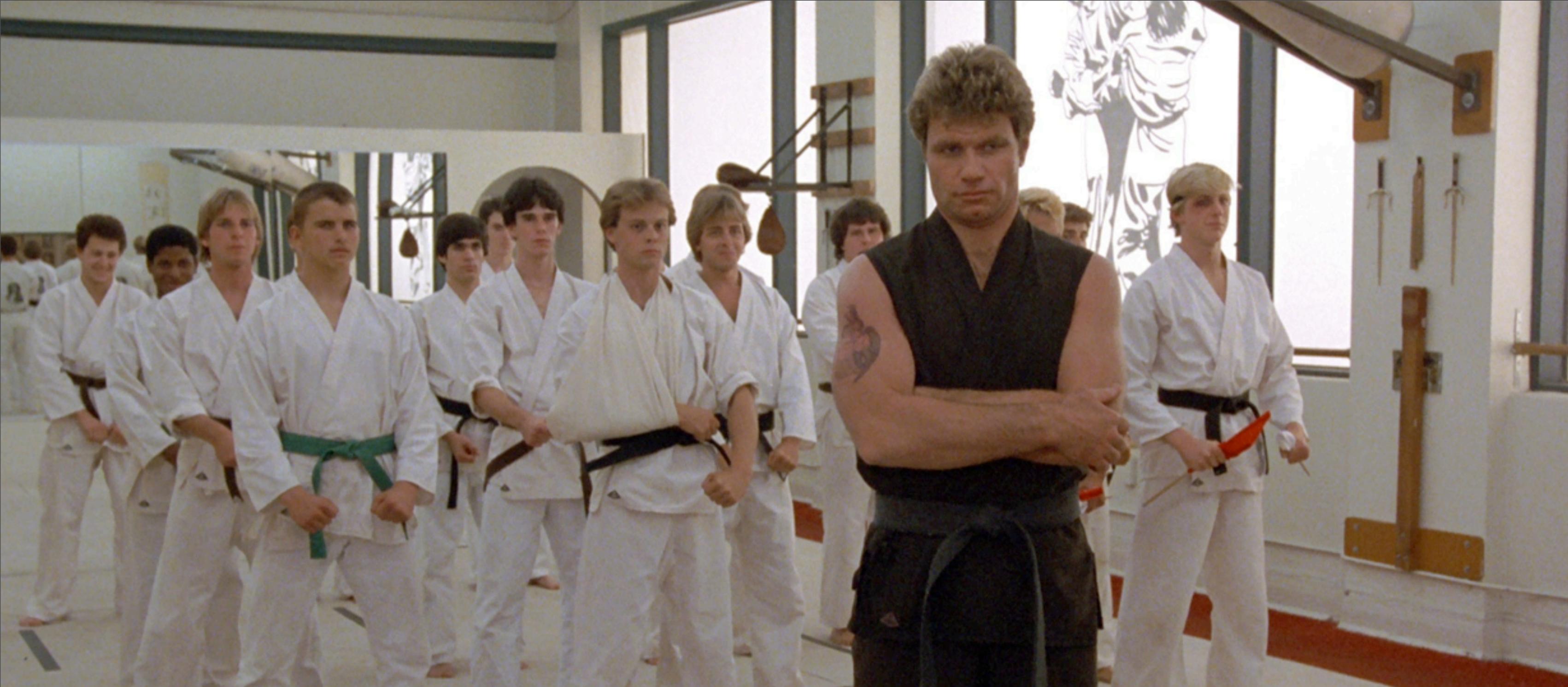
Still, it's hard to know
when you're doing
things right.



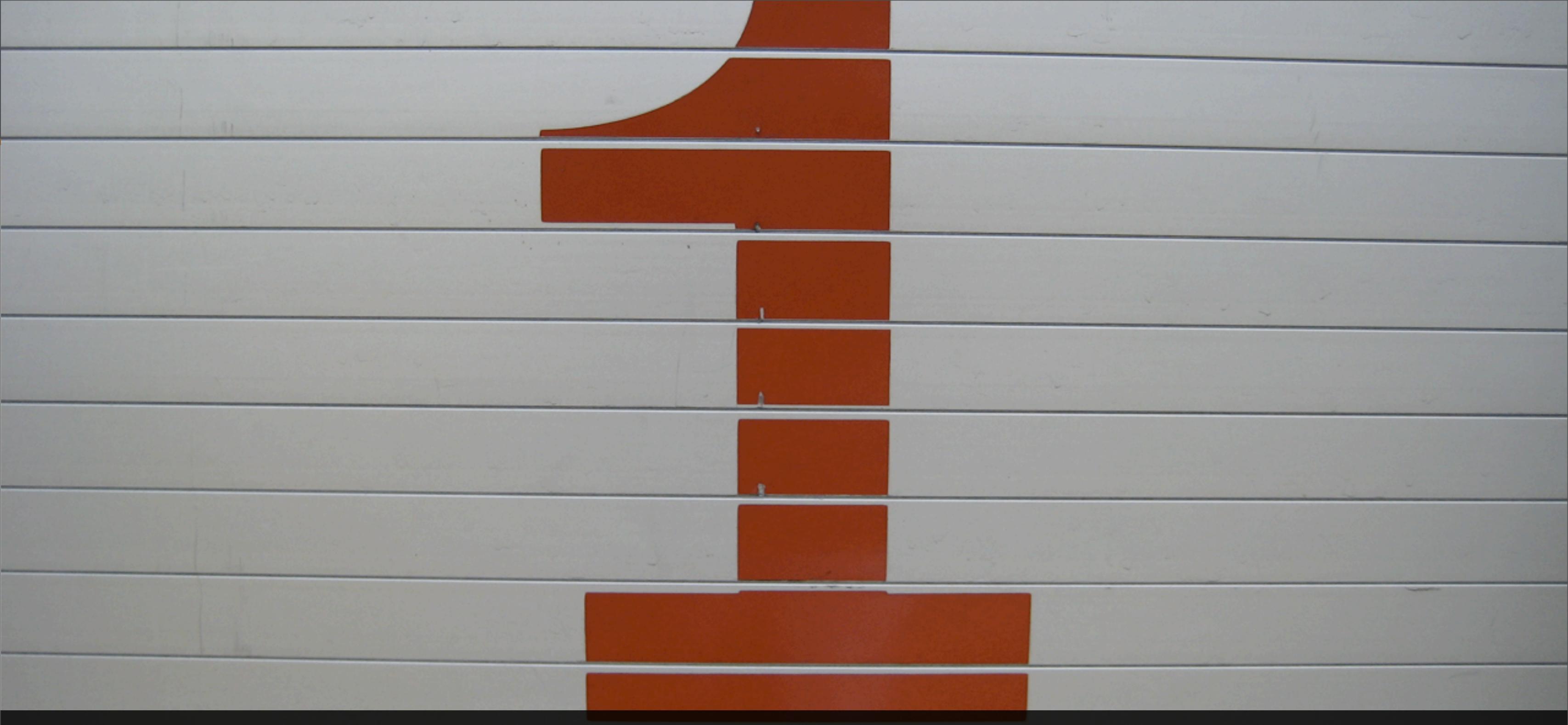
**Even harder to know when you're doing
something wrong.**

**“Best practices” in the community
are evolving all the time.**

- “I would have liked to see more about best practices ... [o]ur instructor had to go ‘off topic’ to explain some common pitfalls.”
- *feedback from Chef 2-Day Fundamentals*



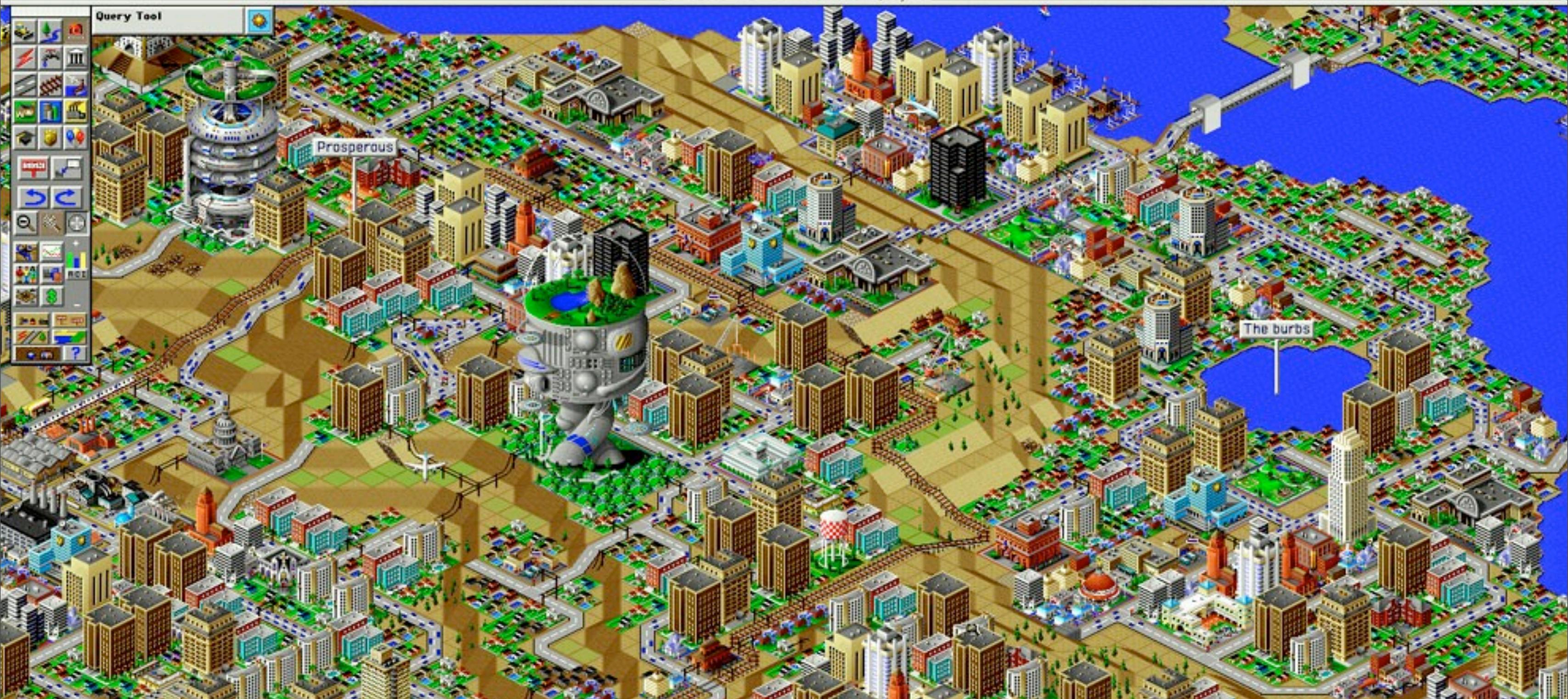
**This talk will give you some best practices to make
you a Master Chef quickly.**



Number one rule

Jun 2389 <New Bournemouth> \$95,513





Nothing replaces good advance planning

Advance planning



Advance planning

- Plan in advance:



Advance planning

- Plan in advance:
 - What cookbooks you're going to have



Advance planning

- Plan in advance:
 - What cookbooks you're going to have
 - What recipes



Advance planning

- Plan in advance:
 - What cookbooks you're going to have
 - What recipes
 - Roles and their names



Advance planning

- Plan in advance:
 - What cookbooks you're going to have
 - What recipes
 - Roles and their names
 - How many environments



Advance planning

- Plan in advance:
 - What cookbooks you're going to have
 - What recipes
 - Roles and their names
 - How many environments
 - Clusters within those environments



Advance planning

- Plan in advance:
 - What cookbooks you're going to have
 - What recipes
 - Roles and their names
 - How many environments
 - Clusters within those environments
 - Data bag hierarchy & naming, data bag item structure



Advance planning

- Plan in advance:
 - What cookbooks you're going to have
 - What recipes
 - Roles and their names
 - How many environments
 - Clusters within those environments
 - Data bag hierarchy & naming, data bag item structure
 - BTW, if you want to go blind: www.textfiles.com/underconstruction/



The Top Ten List of Antipatterns

10. The Giant Git repo for your chef-repo

```
git://github.com/yourcompany/chef-repo.git
```

10. The Giant Git repo for your chef-repo

```
git://github.com/yourcompany/chef-repo.git
```

Why is this bad?

10. The Giant Git repo for your chef-repo

- Mixing temporal data (environments, roles) with versioned data (cookbooks)
- Git philosophy: One Git repo for each thing you're versioning independently
- Don't be afraid of more Git repos!

10. The Giant Git repo for your chef-repo

Better:

10. The Giant Git repo for your chef-repo

Better:

```
git://github.com/yourcompany/chef-data.git
```

10. The Giant Git repo for your chef-repo

Better:

```
git://github.com/yourcompany/chef-data.git
```

```
git://github.com/yourcompany-cookbooks/foo.git
```

10. The Giant Git repo for your chef-repo

More reasons to do this:

10. The Giant Git repo for your chef-repo

More reasons to do this:

```
git remote add upstream git://github.com/whatevs/  
upstream.git  
git fetch upstream  
git merge upstream/master
```

10. The Giant Git repo for your chef-repo

More reasons to do this:

```
git remote add upstream git://github.com/whatevs/  
upstream.git  
git fetch upstream  
git merge upstream/master
```

**Also, easy to open-source your
cookbooks just by tweaking ACL**

9. The one giant cookbook for your company

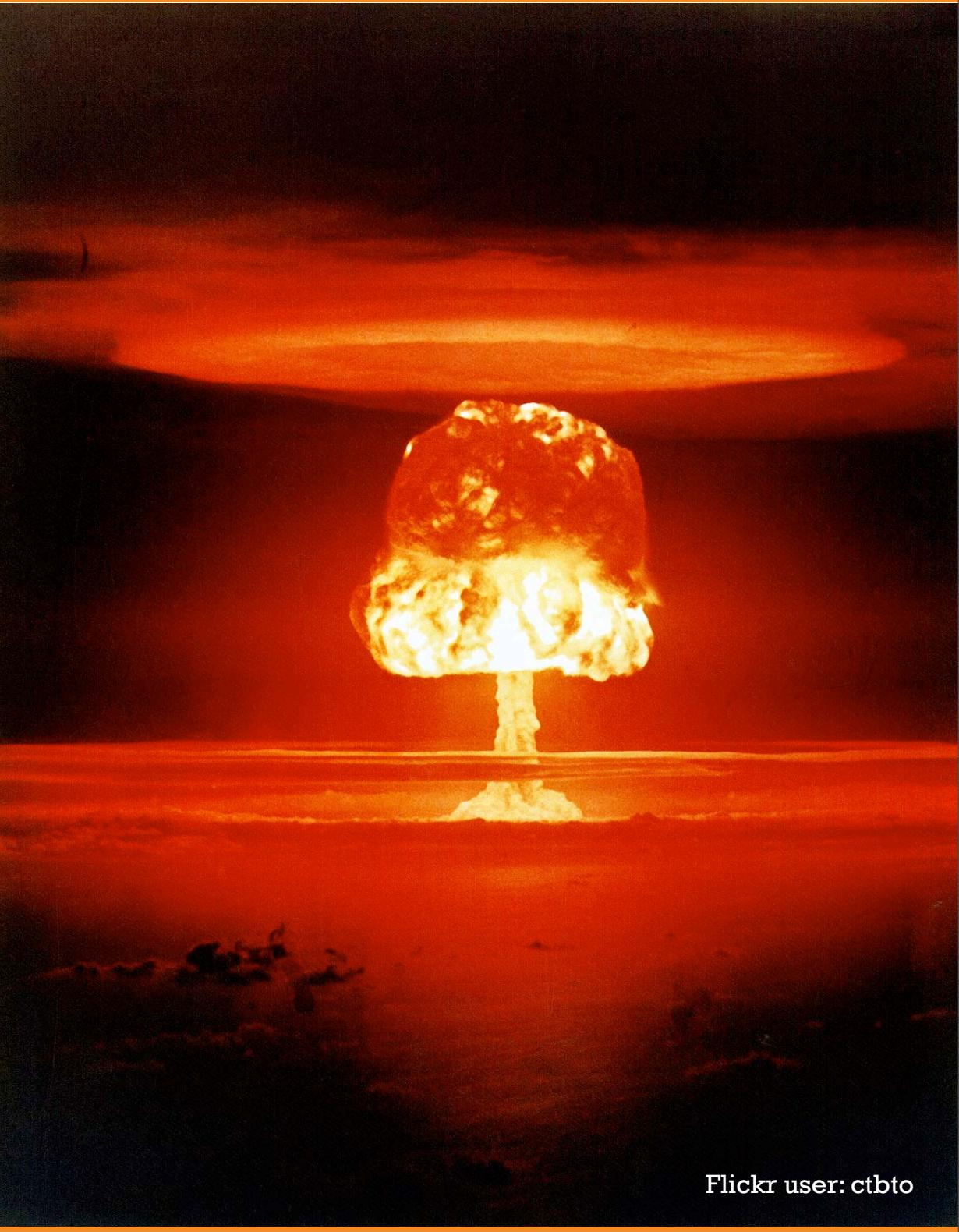
```
git://github.com/yourcompany-cookbooks/yourco.git
```

9. The one giant cookbook for your company

```
git://github.com/yourcompany-cookbooks/yourco.git
```

Why is this bad?

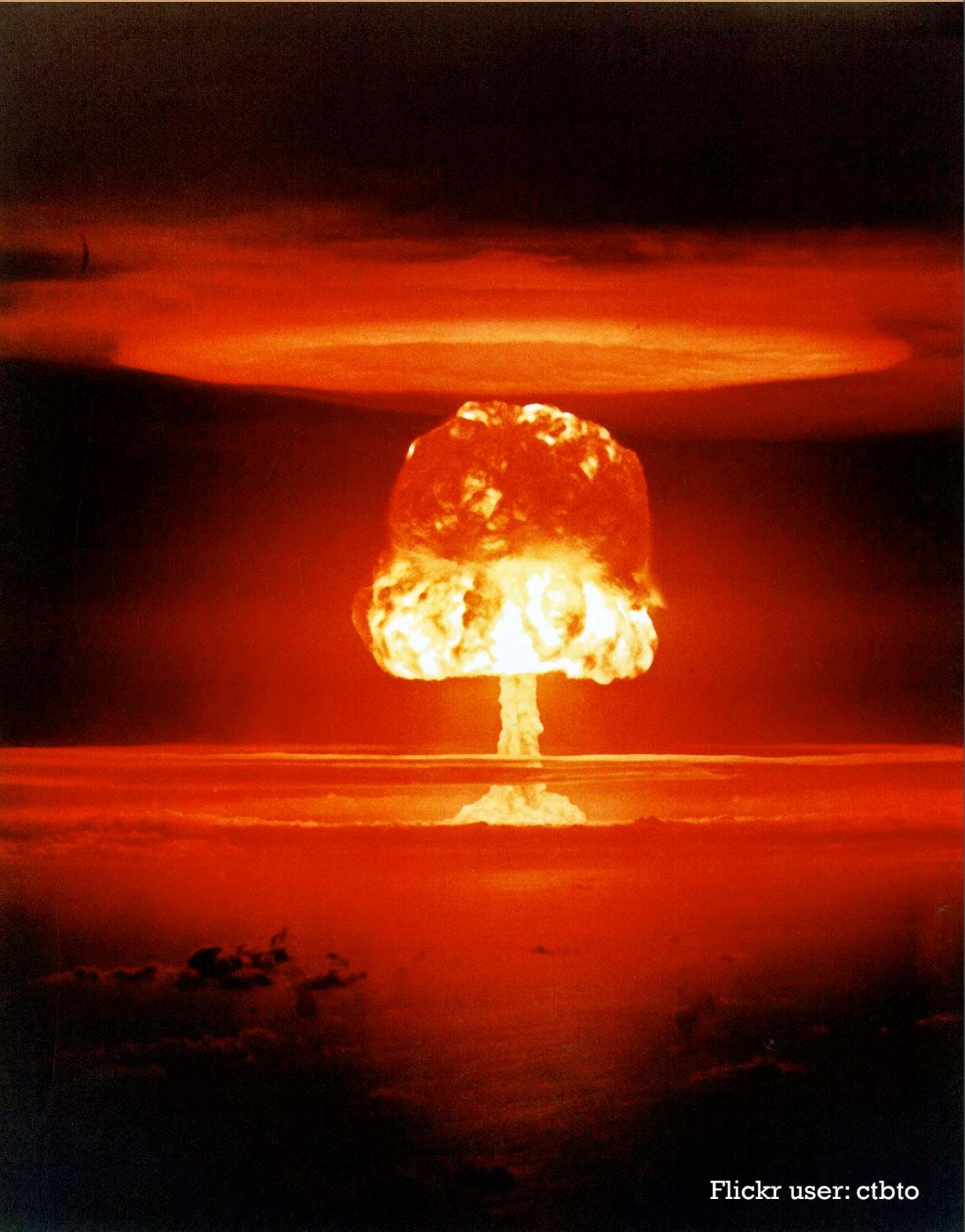
9. The one giant cookbook for your company



Flickr user: ctbt0

9. The one giant cookbook for your company

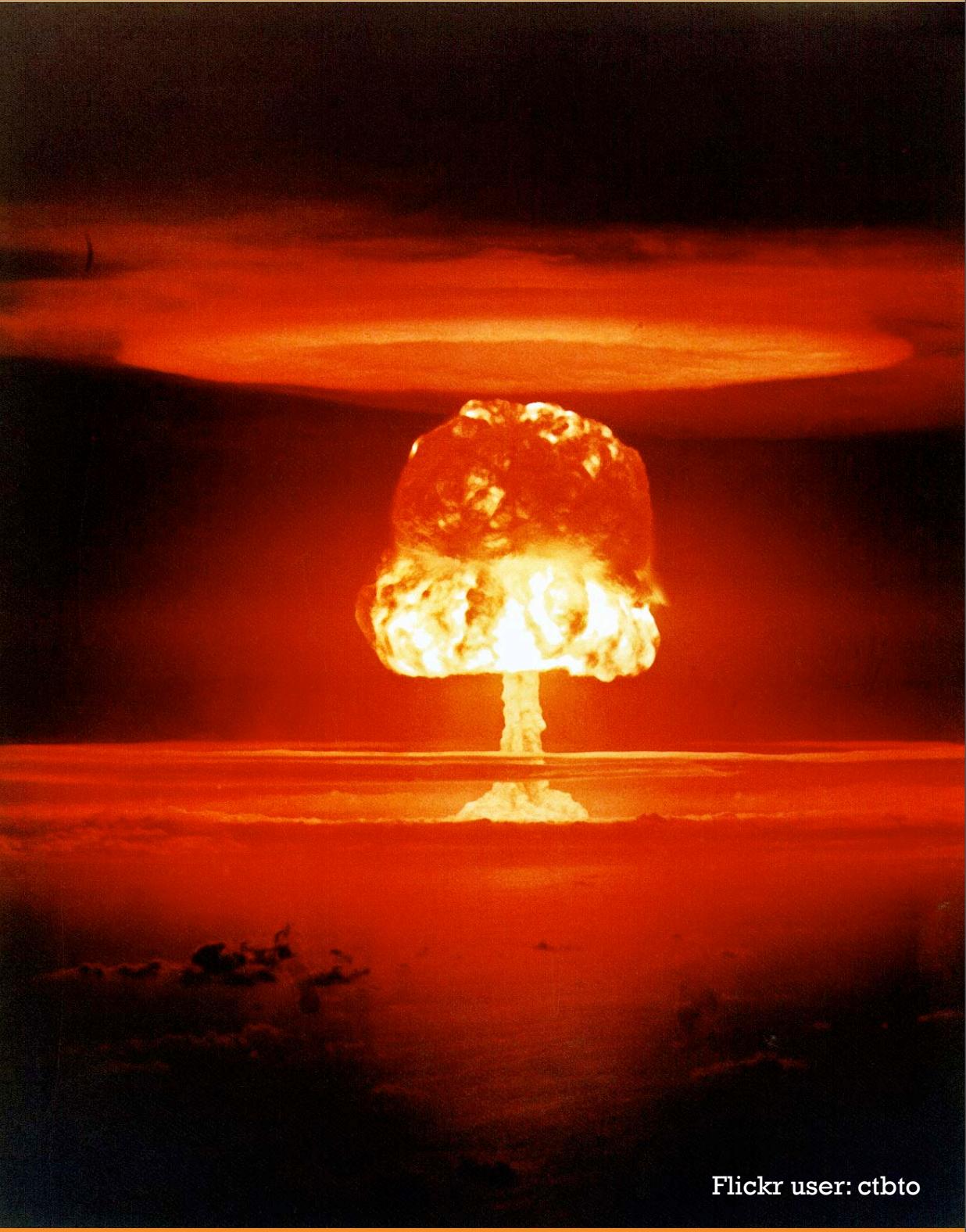
- Chef cookbooks configure a top-level service



Flickr user: ctbt0

9. The one giant cookbook for your company

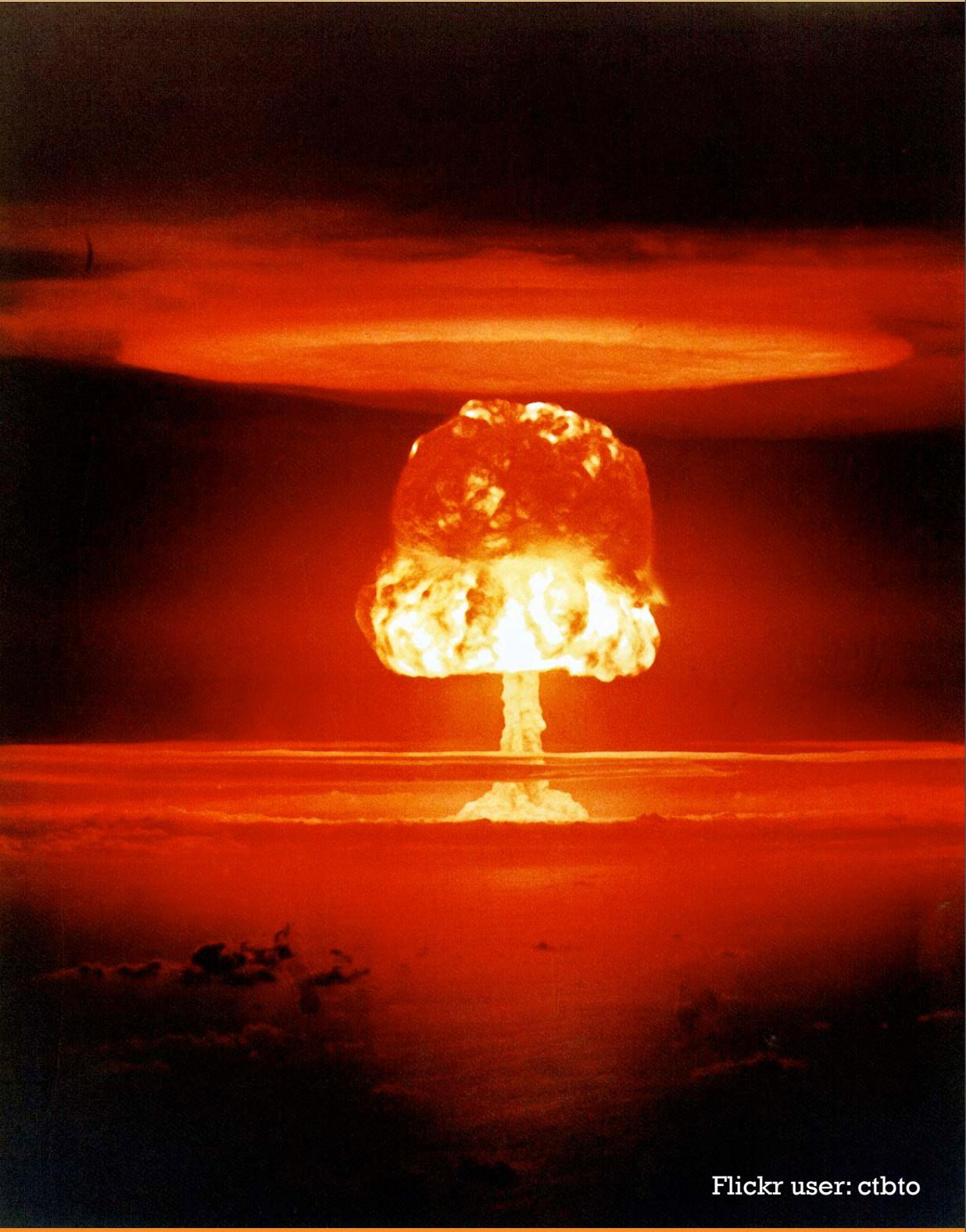
- Chef cookbooks configure a top-level service
- The Giant Cookbook mixes & matches things that don't go with one another



Flickr user: ctbt0

9. The one giant cookbook for your company

- Chef cookbooks configure a top-level service
- The Giant Cookbook mixes & matches things that don't go with one another
- Big blast radius on changes to recipes: leads to accidents



Flickr user: ctbt0

9. The one giant cookbook for your company

Rather than:

- + cookbooks
- + yourcompany
- + recipes
- |
- +- mainsite-apache-virtualhost.rb
- +- anothersite-apache-virtualhost.rb
- +- spring-properties.rb

9. The one giant cookbook for your company

This:

```
+ cookbooks
  + mainsite
    |   + recipes
    |     +- apache-virtualhost.rb
  |
  + anothersite
    |   + recipes
    |     +- apache-virtualhost.rb
  |
  + springproperties
    + recipes
      +- properties.rb
```

8. Using Chef Environments for more than just logical environment

8. Using Chef Environments for more than just logical environment

- Environments are a logical concept, mapping to your actual environments

8. Using Chef Environments for more than just logical environment

- Environments are a logical concept, mapping to your actual environments
- Don't be tempted to overload them as “cluster name” or “data center name” though!

8. Using Chef Environments for more than just logical environment

```
mongos = search(:node, "role:mongodb AND  
chef_environment:#{node.chef_environment}")
```

8. Using Chef Environments for more than just logical environment

```
mongos = search(:node, "role:mongodb AND  
chef_environment:#{node.chef_environment}")
```

**Might not be enough if you have more
than one MongoDB cluster in the
“production” environment**

8. Using Chef Environments for more than just logical environment

Better:

8. Using Chef Environments for more than just logical environment

Better:

```
node.set['mongodb']['cluster_name'] =  
  'mongocluster1'  
  
mongos = search(:node, "role:mongodb AND  
chef_environment:#{node.chef_environment}  
} AND  
mongodb.cluster_name=#{node['mongodb'][  
  'cluster_name']}")
```

8. Using Chef Environments for more than just logical environment

Even Better:

8. Using Chef Environments for more than just logical environment

Even Better:

```
node.set['globals']['data_center'] = 'portlandia'  
node.set['mongodb']['cluster_name'] =  
  'mongocluster1'
```

```
mongos = search(:node, "role:mongodb AND  
chef_environment:#{{node.chef_environment}} AND  
mongodb.cluster_name=#{node['mongodb']  
  ['cluster_name']} AND  
globals.data_center=#{node['globals']  
  ['data_center']}}")
```

7. Forking community cookbooks



7. Forking community cookbooks

- Opscode maintains ~130 cookbooks



7. Forking community cookbooks

- Opscode maintains ~130 cookbooks
- Others out there are also really great & well-maintained (Redis, MongoDB)



7. Forking community cookbooks

- Opscode maintains ~130 cookbooks
- Others out there are also really great & well-maintained (Redis, MongoDB)
- **Resist the temptation to fork cookbooks!**



7. Forking community cookbooks

- Opscode maintains ~130 cookbooks
- Others out there are also really great & well-maintained (Redis, MongoDB)
- **Resist the temptation to fork cookbooks!**
- You won't get the benefit of upstream bugfixes & enhancements



7. Forking community cookbooks

- Rather, use application/library cookbook pattern to overlay your changes (thanks, Bryan Berry)
- Example: SecondMarket's “wrapper” PostgreSQL cookbook

7. Forking community cookbooks

- Rather, use application/library cookbook pattern to overlay your changes (thanks, Bryan Berry)
- Example: SecondMarket’s “wrapper” PostgreSQL cookbook

smpostgresql/recipes/server.rb:

```
21 # Ordering here is very important.
22 # 1. pgdg is to set up the attributes to overlay Opscode's postgresql cookbook
23 #     with the version of PostgreSQL we want
24 # 2. then invoke the platform_postgresql and platform_pg_hba recipes to
25 #     override Opscode attributes with results from our searches
26 # 3. once all attrs are set, then call Opscode's recipe
27
28 include_recipe "smpostgresql::pgdg"
29 include_recipe "smpostgresql::platform_pg_hba"
30 include_recipe "smpostgresql::platform_postgresql_conf"
31 include_recipe "postgresql::server"
32
33 # Jeff Levitt uses some functions imported from tablefunc.sql so it's important to have this
34 # on the system
35 include_recipe "postgresql::contrib"
36
```

See: github.com/secondmarket-cookbooks/smpostgresql.git

6. Run list in roles

6. Run list in roles

- **Controversial, I know!**

6. Run list in roles

- **Controversial, I know!**
- Opscode's own training material says to put run lists in roles

6. Run list in roles

- **Controversial, I know!**
- Opscode's own training material says to put run lists in roles
- But... roles aren't versioned. Anyway, they are temporal data.

6. Run list in roles

- **Controversial, I know!**
- Opscode's own training material says to put run lists in roles
- But... roles aren't versioned. Anyway, they are temporal data.
- Hard to deploy `run_list` changes in a role across environments without the “nuclear” option

6. Run list in roles

Instead of:

6. Run list in roles

Instead of:

```
"run_list": [  
    "recipe[selinux::permissive]",  
    "recipe[rsyslog]",  
    "recipe[chef-client::config]",  
    "recipe[chef-client::service]",  
    "recipe[chef-client::delete_validation]",  
    "recipe[openssh::iptables]"  
]
```

6. Run list in roles

Instead of:

```
"run_list": [  
    "recipe[selinux::permissive]",  
    "recipe[rsyslog]",  
    "recipe[chef-client::config]",  
    "recipe[chef-client::service]",  
    "recipe[chef-client::delete_validation]",  
    "recipe[openssh::iptables]"  
]
```

Do:

6. Run list in roles

Instead of:

```
"run_list": [
    "recipe[selinux::permissive]",
    "recipe[rsyslog]",
    "recipe[chef-client::config]",
    "recipe[chef-client::service]",
    "recipe[chef-client::delete_validation]",
    "recipe[openssh::iptables]"
]
```

Do:

```
% knife cookbook create roles
% vi roles/base.rb
```

```
“run_list”: [ “recipe[roles::base]” ]
```

6. Run list in roles

roles/recipes/base.rb:

6. Run list in roles

roles/recipes/base.rb:

```
include_recipe "selinux::permissive"
include_recipe "rsyslog"
include_recipe "chef-client::config"
include_recipe "chef-client::service"
include_recipe "chef-client::delete_validation"
include_recipe "openssh::iptables"
```

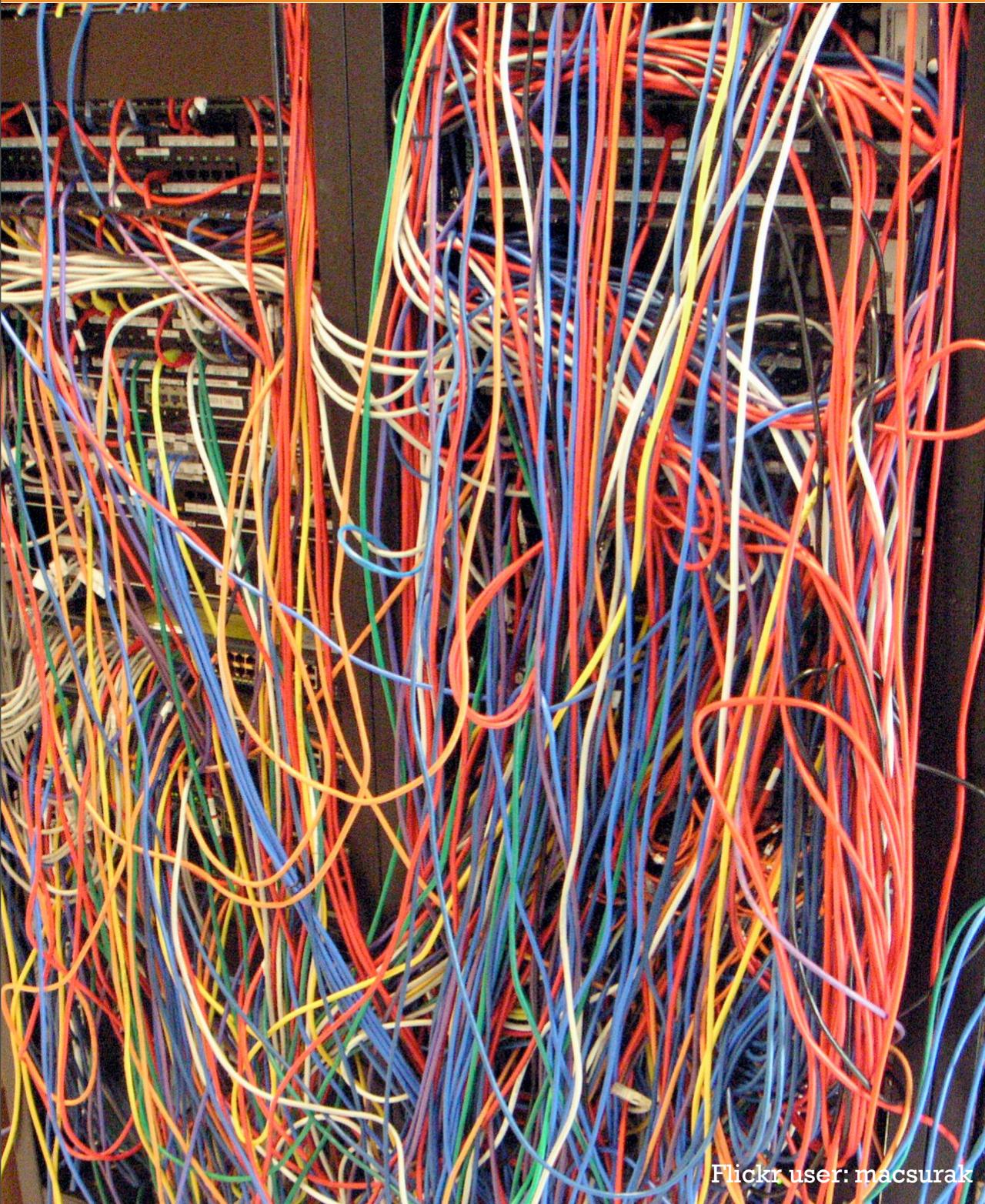
6. Run list in roles

roles/recipes/base.rb:

```
include_recipe "selinux::permissive"
include_recipe "rsyslog"
include_recipe "chef-client::config"
include_recipe "chef-client::service"
include_recipe "chef-client::delete_validation"
include_recipe "openssh::iptables"
```

- Write conditionals around these too if you want
- Or set role attributes in the recipe

5. Disorganized data bags



- Remember what I said about pre-planning?

5. Disorganized data bags

5. Disorganized data bags

5. Disorganized data bags

- Only have two-levels (data bag, and then data bag item) to work with, so plan ahead!

5. Disorganized data bags

- Only have two-levels (data bag, and then data bag item) to work with, so plan ahead!
- Avoid making data bag items enormous JSON hashes - keep them small for performance

5. Disorganized data bags

- Only have two-levels (data bag, and then data bag item) to work with, so plan ahead!
- Avoid making data bag items enormous JSON hashes - keep them small for performance
- 8 KB JSON x 4 Chef runs/h x 1000 nodes = **5.38 GB/week!**

4. Not knowing about or using the chef-shell



Flickr user: blueridgekitties

4. Not knowing about or using the chef-shell

- Chef-Shell (formerly Shef): One of the most under-utilized tools!
- IRB (Interactive Ruby) + Chef primitives
- Cookbook development
- Production debugging



Flickr user: blueridgekitties

4. Not knowing about or using the chef-shell



4. Not knowing about or using the chef-shell

Chef Failure on [REDACTED] Chef::Mixin::Template::TemplateError: undefined method `[]' for nil:NilClass



4. Not knowing about or using the chef-shell

```
29: <% @members.each do |member| -%>
30: <%= member['hostname'] %>      IN      CNAME
<%= member['ec2']['public_hostname'] %>.
31: <% end -%>
```

4. Not knowing about or using the chef-shell

```
[jdunn@dns1 ~]$ chef-shell -z
loading configuration: /etc/chef/client.rb
Session type: client
.
.
.
chef > echo off
chef > members = search('node', "domain:epicfail.com")
chef > members.each do |m|
chef >   pp "#{m['hostname']}, #{m['ec2']['public_hostname']}"
chef ?> end
"host1, ec2-50-17-43-13.compute-1.amazonaws.com"
"host37, ec2-23-23-145-243.compute-1.amazonaws.com"
"host3, "
NoMethodError: undefined method `[]' for nil:NilClass
```

4. Not knowing about or using the chef-shell

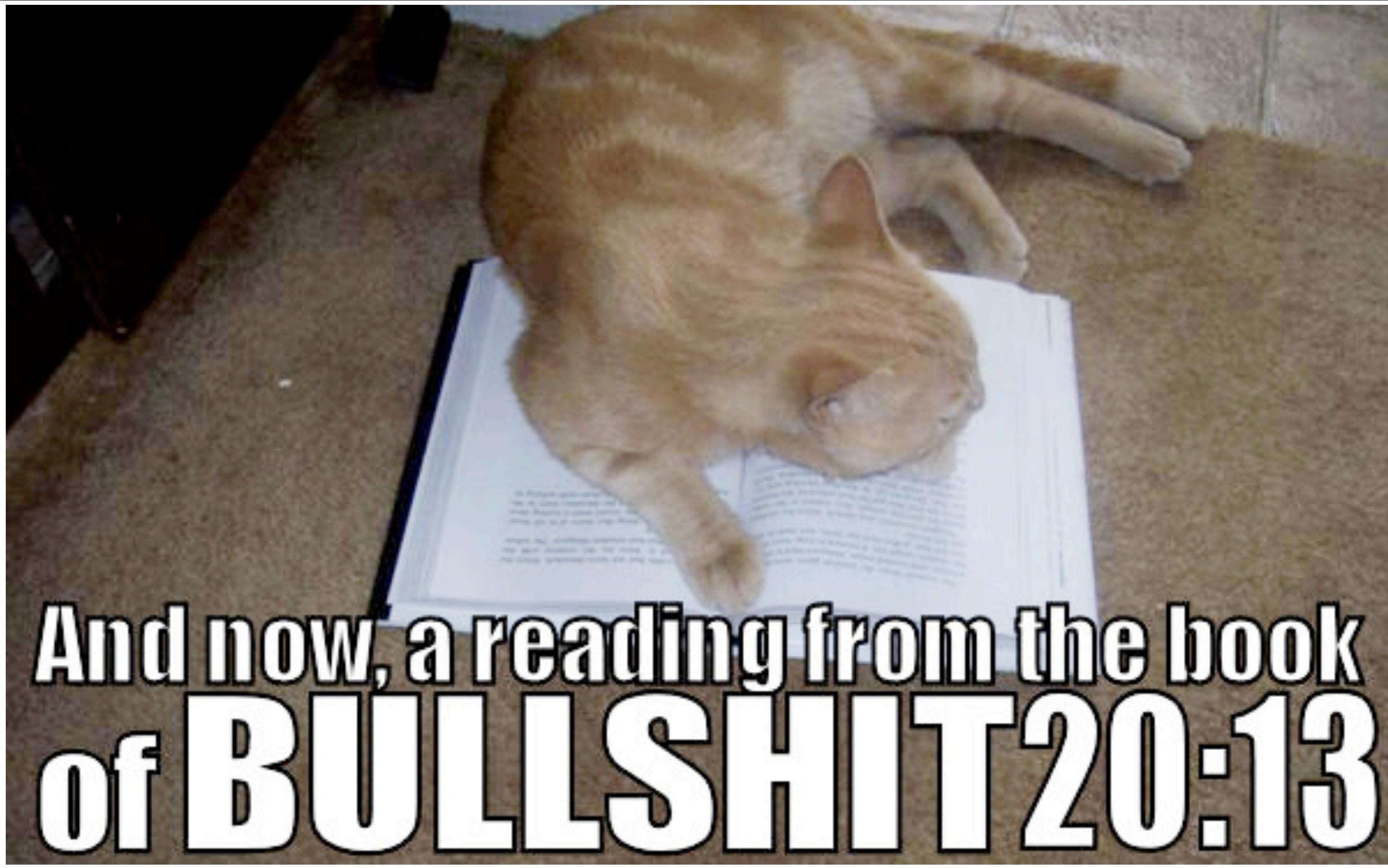
- Way more stuff than this
- Check out my Slideshare deck: [slideshare.net/J JulianDunn/an-introduction-to-shef-the-chef-shell](https://www.slideshare.net/J JulianDunn/an-introduction-to-shef-the-chef-shell)
- **Chef Shell will save you time, guaranteed!**

3. Who's Afraid of the Big Bad LWRP

- **Myth:** LWRPs are hard to write!
You need to know Ruby!



Flickr user: edenpictures



And now, a reading from the book
of BULLSHIT 20:13

catmacros.wordpress.com

3. Who's Afraid of the Big Bad LWRP

- Use inline resources
- Basic Ruby classes and methods go a long way
(Array, Hash, String, etc.)
- The LWRP framework is ... lightweight and does a lot for you

3. Who's Afraid of the Big Bad LWRP

```
cookbooks/mouse/recipes/default.rb
```

```
mouse "Itchy" do
  says "Ow, Scratchy cut off my tail"
  tail false
  action :say
end
```

3. Who's Afraid of the Big Bad LWRP

cookbooks/mouse/resources/default.rb

```
actions :say

attribute :given_name, :name_attribute => true
attribute :phrase, :default => "squeak"
attribute :tail, :default => true, :kind_of => [TrueClass, FalseClass]
```

cookbooks/mouse/providers/default.rb

```
action :say do
  log "My name is #{new_resource.given_name}"
  log new_resource.phrase unless new_resource.phrase =~ /^squeak$/
  log "I #{new_resource.tail ? 'do' : 'do not'} have a tail"
end
```



See, it's that easy!

2. “Not Invented Here” Syndrome

- Bias against using other people’s code/libraries/cookbooks
- Temptation to write your own bespoke cookbook
- Instead, do your research, find the best one, and use it in a library/application cookbook pattern
- Contribute improvements/changes back

#1 Chef Antipattern...



Being the only Chef in your shop

1. The Lone Wolf Chef

- Bus/truck factor of 1
- Chef configures applications
 - Developers know applications better than you
 - Get them involved in writing & maintaining cookbooks
- **Then, everyone is responsible for production-readiness!**

Recap: Top Ten List of Antipatterns

- The one giant Git repo for all Chef data
- The one giant cookbook named after your company
- Using Chef Environments for more than just logical environment
- Forking community cookbooks
- Maintaining the run list in your role
- Disorganized data bags
- Not knowing about or using the chef-shell
- Being afraid of LWRPs
- Not Invented Here Syndrome
- The Lone Wolf Chef

Thanks!

@julian_dunn
github.com/juliandunn
jdunn@opscode.com

We're hiring like gangbusters! opscode.com/careers