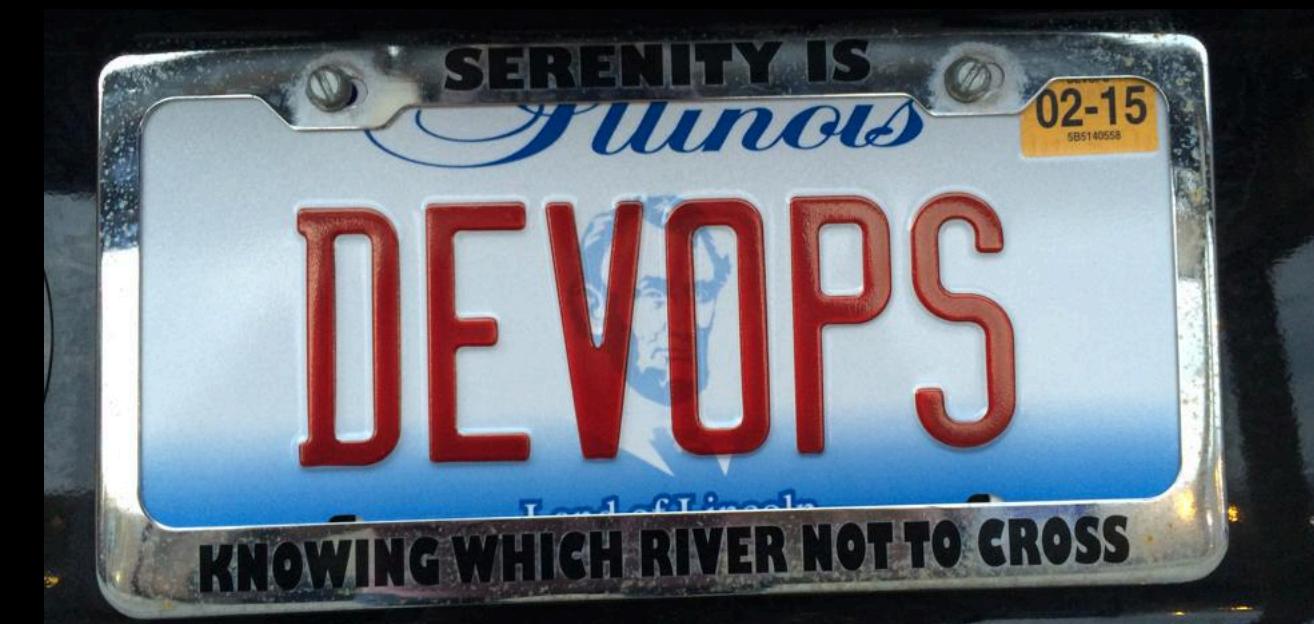


**MAKING INFRASTRUCTURE
S'MORES WITH CHEF**

WHOAMI



WHAT IS CHEF?

- ▶ Define reusable resources and infrastructure state as code
 - ▶ Manages deployment and on-going automation
 - ▶ Community content available for all common automation tasks

```
▼ package "httpd" do
  action :install
end
▲
▼ template "/etc/httpd/httpd.conf" do
  source "httpd.conf.erb"
  mode 0755
  owner "root"
  group "root"
end
▲
▼ service "httpd" do
  action [:enable, :start]
end
```

A colorful cartoon illustration of a king. He has a large, ornate golden crown with a blue jewel. He is wearing a red robe with a white collar and a gold chain with a circular pendant. He is holding a large, curved sword with a brown hilt and a silver blade. A small, white, winged figure with a crown is perched on his right shoulder. The background is a light blue.

ANYONE CAN DO ANYTHING?



OLD WAY
COMMUNICATE VIA TICKETS

**NEW WAY
COMMUNICATE VIA CODE**



I drive the new hotness.

DOMAIN EXPERTS

- ▶ Systems are complicated today
- ▶ Nobody can know everything about the stack
- ▶ Let your domain experts contribute their portion directly

CONFIGURATION DRIFT



DON'T DO THINGS BY HAND

Every time someone logs onto a system by hand,
they jeopardize everyone's understanding of the
system

– Mark Burgess

PEOPLE MAKE
MISTAKES
THIS DOESN'T SCALE

INFRASTRUCTURE AS CODE

Enable the reconstruction of the business from nothing but a source code repository, an application data backup, and [compute] resources

– Jesse Robins

**VERSIONED
MODULARIZED
TESTED**

**EXECUTABLE
DOCUMENTATION**



- I'VE MADE A HUGE MISTAKE.

HOW DO I MAKE SURE
NOBODY MESSES STUFF
UP?

TESTING IS ESSENTIAL

**COMMUNICATE
THROUGH CODE
(REDUX)**



**WHAT HAPPENS WHEN YOU
HAVE ONE GROUP WRITING
ALL THE AUTOMATION?**







HOW DO WE SOLVE THIS?

USE A PIPELINE

CHEF AUDIT MODE AS THE FINAL TEST

EXAMPLE OF AN AUDIT COOKBOOK

```
control '6.9 Ensure FTP Server is not enabled' do
  it 'is not running the vsftpd service' do
    expect(service('vsftpd')).to_not be_running
    expect(service('vsftpd')).to_not be_enabled
  end

  it 'is not listening on port 21' do
    expect(port(21)).to_not be_listening
  end
end
```

**ENCOURAGE LOCAL TESTING
WITH FOODCRITIC**

EXAMPLE FOODCRITIC CUSTOM RULE

```
rule 'COMP001', 'Do not allow recipes to mount disk volumes' do
  tags %w{recipe compliance}
  recipe do |ast|
    mountres = find_resources(ast, :type => 'mount').find_all do |cmd|
      cmd
    end
    execres = find_resources(ast, :type => 'execute').find_all do |cmd|
      cmd_str = (resource_attribute(cmd, 'command') || resource_name(cmd)).to_s
      cmd_str.include?('mount')
    end
    mountres.concat(execres).map{|cmd| match(cmd)}
  end
end
```

ERROR OUTPUT FROM FOODCRITIC

```
$ foodcritic -I /afs/getchef.com/foodcritic-rules/rules.rb .
COMP001: Do not allow recipes to mount disk volumes: ./recipes/default.rb:20
COMP001: Do not allow recipes to mount disk volumes: ./recipes/default.rb:26
```

QUESTIONS?



RESOURCES

- ▶ Sidney Dekker - Field Guide to Human Error
- ▶ foodcritic.io
- ▶ <https://github.com/chef-cookbooks/audit-cis>
- ▶ [http://jtimberman.housepub.org/blog/2015/04/03/chef-audit-
mode-introduction/](http://jtimberman.housepub.org/blog/2015/04/03/chef-audit-mode-introduction/)
- ▶ twitter.com/mattstratton