

**TRIGGER WARNING:
TRISKAIDEKAPHOBIA**

GET READY FOR

CHEF 13

WHAT'S GOING ON HERE?

- ▶ Why Am I Here?
- ▶ Why Chef 13?
- ▶ Why Should You Care?
- ▶ Isn't Upgrading Going To Suck?
- ▶ Writing Modern Cookbooks
- ▶ Questions?

WHY AM I HERE?

- ▶ Chef User/Contributor since 2009
- ▶ Chef Community Engineering Lead since 2015
- ▶ @thommay on github/twitter

WHY CHEF 13?

**LET'S GO BACK IN
TIME**

Not too far back

2013: CHEF 11

Almost exactly 4 years ago. The Chef 10 - 11 upgrade `_sucked_`

2014: CHEF 12

December of 2014, so almost 2 years later. This did not suck anywhere near as hard as 10-11 for users, but was a total death march internally.

OCT 2015: 12.5

Really we should've called this Chef 13.

DEC 2015: 12.6

We **definitely** should've called this 13

PRACTICE WHAT WE PREACH

We're all about continuous delivery, but we sucked at getting our own products out the door. When I joined I was asked to do a Chef 11 build. It took a week to even get the pipeline running, and then another week getting a build to pass tests.

2016: MONTHLY!

Took us 14 months to do 7 releases. Since then we've done a release a month, every month

**HOW DO WE DO
BETTER?**

**APRIL 2017 –
CHEF 13**

APRIL 2018 – CHEF 14

We're putting it all on the line - in 14 months time, we'll do Chef 14. And in 26 months time - well, we hopefully won't be putting out chef 26. But we will put out chef 15.

MORE TOOLS

foodcritic moves to cookstyle
rubocop style auto corrections in cookstyle

WHY SHOULD YOU CARE?

predictability
not bundling up everything in one huge lump

**1 NEW RESOURCE
PER RELEASE OF 12**

On average, we've added just under 1 new resource every time we've made a release of chef 12

NEW RESOURCES

- ▶ [bff_package](#)
- ▶ [homebrew_package](#)
- ▶ [reboot](#)
- ▶ [windows_service](#)
- ▶ [openbsd_package](#)
- ▶ [paludis_package](#)
- ▶ [dsc_resource](#)
- ▶ [ksh](#)
- ▶ [chocolatey_package](#)
- ▶ [apt_repository](#)
- ▶ [osx_profile](#)
- ▶ [apt_update](#)
- ▶ [launchd](#)
- ▶ [systemd_unit](#)
- ▶ [yum_repository](#)
- ▶ [cab_package](#)
- ▶ [msu_package](#)
- ▶ [dnf_package](#)
- ▶ [powershell_package](#)
- ▶ [zypper_package](#)

PROBLEM?

Well, yes.

A SOLID BASELINE

Which Chef 12 release can I use which resource in?

FEWER COOKBOOK DEPS

By the time 13 is released, we aim to have all the resources from the yum, apt and windows cookbooks in core chef. So your dependency chains get shorter.

**ISN'T UPGRADING
GOING TO SUCK?**

MAYBE

**BUT PROBABLY
NOT**

**GET STARTED
TODAY**

DEPRECATIONS

Deprecations now tell you about themselves, and give you a link to read more

DEPRECATIONS AS ERRORS

you can make deprecation warnings errors in test-kitchen or in chef-client config

RESOURCE CLONING

Resource Cloning, aka CHEF-3694, is finally getting turned off. Kinda. If you have two resources that are more or less identical, that's generally fine. It's just when it becomes unsafe to merge them you'll see errors.

Like deprecations, you can make chef error now if you get resource cloning warnings.

KITCHEN.YML

provisioner:

client_rb:

treat_deprecation_warnings_as_errors: true

resource_cloning: false

WRITING MODERN COOKBOOKS

Tim's gonna talk about Community Cookbooks. But I have some tips and thoughts on general cookbook writing.

CUSTOM RESOURCES

We don't talk about HWRPs or LWRPs any more. I would make a 1984 joke, but it's a bit too topical right now.

BUILD RESOURCES

COMPOSE THEM IN RECIPES

So called wrapper cookbooks. But really, just cookbooks. We're programmers - most of our life is mashing together libraries into working programs. No different here.

EDIT_RESOURCE
DELETE_RESOURCE

If you are still wrapping cookbooks and using `chef_rewind`, use this instead

ACCUMULATORS

METADATA GEM DEPENDENCIES

QUESTIONS?