

jbellone^work is now known as jbellone 5:58 pm

pwm_away is now known as pwm 10:35 pm

January 8th, 2015

zz_kvanderw is now known as kvanderw 7:03 AM

martinisoft_ is now known as martinisoft 8:36 AM

coderanger Ten minute warning 8:51 AM

martinisoft grabs coffee 8:52 AM

c_t whoop whoop whoop 8:52 AM

nathenharvey Hello and happy new year everyone! 8:56 AM

drag00n nathenharvey: same to you! 8:57 AM

ssd7 Ohai, Chefs! 8:57 AM

drag00n o/ 8:57 AM

j^2 :D 8:58 AM

c_t ohai, all 8:58 AM

zts :) 8:58 AM

c_t pity that on IRC there can't be a snack table
to munch at while waiting for everything to
start 8:58 AM

ortho_stice <hugs> 8:59 AM

ssd7 As a heads up, A lot of Chef employees are
in company meetings today, so it might be a
slow meeting 9:00 AM

holoway__ WHATS UP 9:01 AM

thom happy new year, folks 9:01 AM

	<i>holoway__ is now known as holoway</i>	9:01 AM
holoway	how are you healing up, thom?	9:02 AM
nathenharvey	Hello, all!	9:02 AM
	Let's get this meeting started	9:02 AM
	*** MEETUNG STARTS ***	9:02 AM
	s/U/I ;)	9:02 AM
	<i>j_b_d is now known as jdossey</i>	9:02 AM
holoway	I think Meetung is more metal	9:02 AM
	fwiw	9:02 AM
nathenharvey	I've just pushed an update to today's agenda - https://github.com/opscode/chef-community-irc-m...	9:02 AM
c_t	Substitution replacement not terminated at -e line 1.	9:03 AM
nathenharvey	MEETUNG it shall be	9:03 AM
jonlives	i'm in a colossal meeting too so will be absent unless anybody mentions me :)	9:03 AM
thom	holoway: living with one working leg is somewhat amusing but it's going well, thanks :)	9:03 AM
holoway	jonlives has great hair	9:03 AM
jonlives	:p	9:03 AM
holoway	thom: did they rebuild you, make you a stronger, better, and faster? :)	9:03 AM
thom	developing mad hopping skills	9:03 AM
nathenharvey	As ssd7 mentioned, most of the people that work at CHEF are in Seattle this week for lots of beginning of the year kick-off and planning meetings	9:04 AM
holoway	so things are a bit sparse	9:04 AM

	but we have some RFCs to review	9:04 AM
	I can give some org updates about whats up in Chef Inc	9:05 AM
	so lets talk RFCs, eh?	9:05 AM
	you around, @coderanger ?	9:05 AM
coderanger	Yep	9:05 AM
holoway	number 1 on the agenda is dialects	9:05 AM
	https://github.com/opscode/chef-rfc/pull/71	9:05 AM
	I think the thread came to good conclusions	9:06 AM
nathenharvey	Big thanks to coderanger for continuing to spread the word about this meeting on the mailing list and twitter while I've been tied up in said company meetings	11:06 am
holoway	that we're all basically +1 on the code cleanup, but super duper worried about introducing even more confusion into the ecosystem	11:07 am
	meanwhile, POSHChef rocks a complete powershell dialect, making Noah look prescient	11:07 am
coderanger	Okay, I tried to clarify the issues from last time with leaking more features in to core	11:07 am
holoway	https://github.com/POSHChef/POSHChef/	11:07 am
coderanger	The hooks will be added to core objects, but no new support on any of them except that it could be used to unify loading behavior on chef_fs	11:08 am
	but I think thats desired anyway	11:08 am
holoway	yep - I think we should add your comment clarification to the RFC	11:09 am
coderanger	As for community complexity, any cookbook written in a dialect will have to depend on the relevant dialect, so will always work "out	11:09 am

	of the box" if you use a recursive downloader	
holoway	and then unless the core is against it, I say we accept the RFC and look forward to our dialect driven life	11:09 am
	is there anyone against?	11:09 am
coderanger	holoway: I can add, its not really part of the spec though, thats mostly just an enumeration of the current features of Chef :)	11:09 am
jdosssey	to be clear, you're saying dialects yes, but just for chef_fs?	11:10 am
coderanger	holoway: Maybe add it as an acceptance criteria section :)	11:10 am
	jdosssey: No, would be added to all object loading	11:10 am
holoway	jdosssey: yeah - we are saying add the subsystem, but only have the current dialects added	11:10 am
coderanger	holoway: But it wouldn't just be for chef_fs	11:10 am
jdosssey	I dread the day when community cookbooks are written in FORTH	11:10 am
holoway	tl;dr - we don't add any new variants to core chef, but we open up the api	11:10 am
	coderanger: right	11:10 am
	chef_fs would gain dialects too	11:10 am
	as you point out	11:11 am
Tyrael	maybe not the appropriate place and time but any chance that the opscience/opscience-webui2 repo will be made public so people can make Pull Requests?	11:11 am
coderanger	That is just notable because it should be the only place where externally visible changes happen	11:11 am
holoway	ya	11:11 am

coderanger	holoway: Opinions on templates and/or node dialects?	11:12 am
holoway	Tyrael: not super likely, considering it's proprietary code. we've toyed with the idea of a license that allows that (and keeps it proprietary) but haven't actually moved on it	11:12 am
coderanger	holoway: I think templates are a v2, and node objects just not at all	11:12 am
holoway	coderanger: I think it would be a mistake to not add templates and node dialects	11:12 am
	but you can progressively render	11:12 am
	ie: add them later	11:12 am
Tyrael	holoway: do you have anything in the repo which doesn't already shipped with the opscore-manage package?	11:12 am
coderanger	holoway: Yeah, initial bits will just be porting up the old patches	11:12 am
	Tyrael: Yeah, unminified JS code	11:13 am
Tyrael	do you do some kind of obfuscation or just simple minify? because if the latter, there are js unminifiers/beutifiers, so reverting that isn't an issue, so there are no real gains from that	11:14 am
ranjib	chef_fs dialects for storage abstraction?	11:14 am
Tyrael	and btw. props for the new UI, really nice.	11:14 am
nathenharvey	I think we can/should further discuss the opscore-webui2 repo but should get it on the agenda.	11:16 am
holoway	ok - nathenharvey, lets make a note that the core LTs have until the next meeting to mount an arugment, or if they all consent, I'm just merging this RFC	11:16 am
coderanger	holoway: Okay, added new section	11:16 am
nathenharvey	holoway: got it!	11:16 am

jdosssey	Sounds like we're all in agreement on the dialects thing	11:16 am
coderanger	https://github.com/coderanger/chef-rfc/blob/dia...	11:16 am
holoway	I posted to github	11:16 am
	coderanger: YDM	11:16 am
Tyrael	(sorry for the distraction)	11:16 am
holoway	also, coderanger, in what world are you not a core maintainer?	11:16 am
	(I mean, make your choices or whatever, but seriously)	11:16 am
coderanger	holoway: Haven't gotten to it :P	11:17 am
holoway	coderanger: ok good	11:17 am
	lets move on	11:17 am
	https://github.com/opscode/chef-rfc/pull/75	11:17 am
	robust attribute tracing	11:17 am
	<i>ortho_stice waves</i>	11:17 am
	I have a feeling this will move quickly into https://github.com/opscode/chef-rfc/pull/77	11:17 am
ortho_stice	yup, 75 is pretty dependent on 77	11:17 am
lamont_oc	yeah #77 needs to get sorted out first	11:18 am
coderanger	All hail our glorious attr2.0 overlords	11:18 am
ortho_stice	What's next for #75? do we put it on hold until 77 matures?	11:20 am
coderanger	holoway: Whats next?	11:20 am
lamont_oc	yeah it'd be more productive to talk about 77	11:20 am
ortho_stice	nod	11:20 am

ranjib	+1, should wait till 77 is merged,	11:20 am
coderanger	I think we should probably just close out 75, there is a section in 77 about tracing support?	11:20 am
	Any required features should be documented there	11:21 am
holoway	lets talk 77	11:21 am
coderanger	can't really do any impl planning since thats too far afield right now anyway	11:21 am
holoway	I say we put 75 on ice	11:21 am
	and make sure its part of anything we build on 77 explicitly	11:21 am
	I have to catch up on the actual thread on 77	11:21 am
lamont_oc	75 should fall out of 77 if we do 77 right	11:21 am
holoway	my \$0.02	11:21 am
	1) I want the new attribute API to be method calls, and not use the [] syntax at all	11:22 am
	(and if it does, it should be because it returned you an actual hash	11:22 am
lamont_oc	to be arguments to methods?	11:22 am
holoway	ya	11:22 am
lamont_oc	yeah	11:22 am
holoway	2) desired and current state should be seperated	11:22 am
	3) we still need per-node persistent storage, but I don't care how	11:23 am
	4) per-node consistent storage may just be slang for "desired state", but probably isn't all of it	11:23 am
lamont_oc	dan decided that per-node storage was out	11:23 am

of scope for 77

holoway

lamont_oc: you need the equivalent of a node normal attribute

11:23 am

(without them, you don't support servers that are pets - and while I know we all want cattle, sometimes...)

11:24 am

coderanger

I think the plan was to leave normal as-is

11:24 am

so will still be supported the old way

11:24 am

lamont_oc

yeah, i thought we were just going to leave it the old way

11:24 am

coderanger

new APIs for it are out of scope

11:24 am

there was some talk of pulling the persistent storage out of attributes and in to its own thing

11:24 am

but thats a bigger change

11:25 am

would be nice to not accidentally poison attrs but I'm not sure how workable that idea is

11:25 am

holoway

my point is, the new api isn't complete until it has that functionality

11:25 am

coderanger

holoway: Okay, I don't think there is any disagreement on that front, it will have it :)

11:25 am

holoway

skipping it is a cop out because we're being opinionated about the value of nodes with persistent attributes

11:25 am

so put it in the RFC, and we can implement it in stages

11:26 am

but lets declare our intent completely

11:26 am

coderanger

Can I just say for the record that I really hate the idea of setters using method syntax :-(

11:26 am

holoway

coderanger: example?

11:26 am

coderanger	getters via methods looks nice, but set('foo').to('bar')	11:26 am
	is just grrrr	11:26 am
	Or whichever of the chaining proposals you pick	11:27 am
	I like my equals signs	11:27 am
holoway	ahh	11:27 am
	yeah, I can see an ergonomic argument for set('foo') = 'bar'	11:28 am
coderanger	They lend to quick visual scanning way better, we're all deeply training to grok assignment syntax	11:28 am
	holoway: Yeah, unfortunately ruby doesn't allow that	11:28 am
lamont_oc	right	11:28 am
ranjib	wont it be better to separate out the persistence requirements into separate RFC, just build consensus on the API bit now get it merged, then raise a dedicated RFC for attribute persistence overhaul,	11:28 am
lamont_oc	i think dan's ideas on persistence is that in addition to attr() and sys() we have store() or something like that	11:29 am
holoway	ranjib: maybe - I think the attributes API shape is pretty impacted by what we think needs to happen to the node object	11:29 am
ranjib	i like all the styles that does not involve method missing	11:29 am
	k	11:29 am
coderanger	The #[] and #[]= syntax is something familiar to a lot of people, I really don't like this idea of building a new API from scratch	11:29 am

	I agree it shouldn't be deeply nested hashes	11:29 am
holoway	ranjib: just so we're clear, kallistec totally got rid of method missing like 5 years ago, and I sat on his patch like a punk	11:30 am
coderanger	but I strongly think we should stick to those two interfaces for access/sets	11:30 am
holoway	ranjib: so you can punch me in the shoulder when you see me next :)	11:30 am
troyready	I like the idea of the vision of persistent attributes being included in the api2 RFC. Regardless of the implementation	11:30 am
holoway	coderanger: so much of the confusion is precisely because of the things we have to do to make it work hashy	11:30 am
coderanger	holoway: Sure, I would rather see it be more like a flat list of attrs instead of fully hashy	11:30 am
	holoway: Like node['foo.bar']	11:31 am
holoway	I really think the [] syntax is a mistake	11:31 am
	we're going to be breaking compat deeply anyway, and have to keep the old api kicking (and know how to move between the two) for a long time	11:31 am
ranjib	:-) . also, not the default['parent', 'child'] style declaration is bit odd, as we now have search filters that takes the same style, but returns flat hash,	11:31 am
lamont_oc	yeah after writing both method-argument syntax and hash-chaining syntax for the chef 12 attrs stuff, i hate hash method chaining and the #[] and #[]= operators because unless you hit a #[]= you never know you're done or if you're reading or writing. it sucks.	11:31 am
coderanger	holoway: So we don't need to support the weird magic with complex sets, its still a	11:31 am

	single method call if you do <code>node['foo.bar'] = 'baz'</code>	
holoway	coderanger: but if we do that, just put () around it	11:32 am
coderanger	lamont_oc: No, if you declare it to be single-layer you don't need to do the hash magic	11:32 am
holoway	and be done	11:32 am
ranjib	yeah.. thats a pain [x.y] style..	11:32 am
coderanger	holoway: <code>node('foo.bar') = 'baz'</code> is not valid ruby	11:32 am
	holoway: If you could do that, I would be all for it, but thats only valid with []	11:32 am
lamont_oc	the dot syntax has broken edge cases	11:32 am
ranjib	:-) lvar, i miss perl	11:32 am
holoway	ranjib: RIGHT?!?!	11:32 am
zts	+1	11:33 am
ortho_stice	hah, yup	11:33 am
holoway	ranjib: a little <code>Class::MOP</code> and we could break the AST	11:33 am
	ranjib: and this shit would be solved	11:33 am
	we could use unicode dot	11:33 am
coderanger	lamont_oc: Yeah, thats the hard part, you can fix the edge cases, but it gets ugly	11:33 am
	<code>node[['foo', 'bar']] = 'baz'</code>	11:33 am
lamont_oc	i really don't want to inflict the edge cases of the dot syntax on users, phil is opposed, serdar and dan don't like it either. we looked at that for the chef-12 attr changes and it was hella ugly and we just ignored it.	11:33 am
holoway	I'm going to use my decider powers and just say I'm killing [] from api 2	11:33 am

ortho_stice	please yes unicode dot	11:33 am
lamont_oc	holoway +1	11:33 am
coderanger	or node['foo', 'bar'] = 'baz'	11:34 am
holoway	if we're going to event contemplate inflicting this on users, we owe it to ourselves (and them) to make it actually compelling and awesome	11:34 am
coderanger	holoway: Just keep in mind that also kills =	11:34 am
holoway	coderanger: I'm totally fine with that	11:34 am
coderanger	holoway: And I don't think you can make this awesome without -	11:34 am
holoway	I'm not sure it kills = neccessarily	11:35 am
coderanger	Unless there are deep Ruby tricks I don't know, you can't mix = and method calls	11:35 am
	only []= and attr_writter	11:35 am
holoway	I think you are right, but there might be Deep Ruby Tricks We Don't Know	11:36 am
	but even so, .to is survivable if the behavior is clear and the ergonomics are okay	11:36 am
	I'm the guy who brought you	11:36 am
coderanger	holoway: Thats literally telling everyone they need to relearn how to set a variable :-/	11:36 am
holoway	package "foo" do; version '1.2.3'; end	11:36 am
	coderanger: they aren't setting a variable, and that's my point right there	11:37 am
	coderanger: that shit is not a variable	11:37 am
coderanger	What are they?	11:37 am
holoway	coderanger: it's a bit of contextual data hung off the node object, but it's not a	11:37 am

	ruby builtin	
	a-la 'foo = ":bar"'	11:37 am
coderanger	holoway: I don't think that fairly reflects how node attrs are used today	11:37 am
holoway	coderanger: I agree, and i'm pretty sure its why we need to break the API	11:38 am
	because it keeps not behaving like one	11:38 am
coderanger	They are used _extensively_ as variables to communicate between wrapper and wrapped cookbooks	11:38 am
holoway	I worked hard to make it act like a real builtin	11:38 am
	and it just isn't	11:38 am
	sure, and that will still work just fine	11:38 am
coderanger	It will work, but it sounds like you want to move away from those semantics	11:38 am
	and I'm not sure you can get away with that :)	11:38 am
holoway	(for example, you don't use = to set parameters on resources)	11:39 am
coderanger	Thats going to turn this from a new API, in to "switch to new semantics"	11:39 am
holoway	(because.. its not a variable)	11:39 am
coderanger	holoway: Sure, but those aren't used to move data around in the same way by most people	11:39 am
holoway	coderanger: splitting desired and current state is going to force a switch to new semantics	11:39 am
	my core point stands - 99% of the confusion is precisely because we encourage people to think about them as if they are the standard builtins	11:40 am

	and it just isn't at all	11:40 am
	not even close	11:40 am
cmalek	holoway: what confusion have people expressed?	11:40 am
holoway	as soon as you see the {} syntax, you think "oh, a hash - I know what to do with this"	11:40 am
coderanger	holoway: If there is an operation "set X to Y", I think that should look like a similar operation in almost every other lang	11:40 am
	holoway: And I disagree that {} means the current weirdo nested hash setup	11:41 am
	holoway: I think it could be as a single-layer the way we want to with methods	11:41 am
holoway	coderanger: that's precisely what it means to everyone else	11:41 am
cmalek	holoway: ah	11:41 am
coderanger	holoway: I think you infer too much, if the return value of node['foo'] is a string, you don't assume it is a nested hash	11:41 am
holoway	no, I assume 'node' is a hash	11:42 am
	which it isn't at all	11:42 am
coderanger	holoway: node['foo'] == nil, node['foo', 'bar'] == 'baz'	11:42 am
	or something like that	11:42 am
holoway	that the hash gave me a string, good times	11:42 am
ortho_stice	Well, many people assign entire hash literals	11:42 am
holoway	while the node['foo', 'bar'] syntax can be forced to work, that's also ergonomically bizarre	11:42 am
	and not idiomatic ruby at all	11:42 am

ortho_stice	Or even more problematic, try to append to an array	11:42 am
coderanger	holoway: Thats basically the same syntax as the other proposals with the method name changed :)	11:43 am
holoway	right	11:43 am
	and I'm saying just stop forcing the use of '=', which in ruby (tends to) mean assignment to a variable (local, class, instance)	11:43 am
	which isn't what you are doing when you set these attributes	11:44 am
	it's okay to break it	11:44 am
coderanger	holoway: I still don't see how to teach node atts except as "these are values, you can set them"	11:44 am
lamont_oc	from a code perspicive i can make node['foo', 'bar'] or node('foo', 'bar') work because i can see where it ends. node['foo']['bar'] makes	11:44 am

[Sign In](#)

#chef-hacking / January 8th 2015

Search



Hide joins/parts

	it has keys and values	2013 1:44 am
	there is literally no way around that	2014 1:44 am
holoway	you can set them	2015 1:44 am
	the fact that the API to set it is a meth call and not an assignment with = is n big deal	January am Last Week
coderanger	the way people think of key value stor in Hash (or dict, hashmap, whatever) : almost every language uses [] and []= for that	This Week Current
holoway	AND THIS IS MY POINT	11:45 am

	it's not a hash	11:45 am
coderanger	holoway: Its a key value store, that makes it a hash	11:45 am
holoway	and if we keep the facade that it is, we are going to keep driving users straight off a cliff	11:45 am
	bullshit	11:45 am
	that it's a hash is an implementation detail	11:45 am
coderanger	What else would you call a thing with keys and values?	11:45 am
holoway	just because it stores keys and values doesn't make it a hash :)	11:46 am
	in the built-in sense	11:46 am
lamont_oc	its about 15 hash maps which are joined together with bizzare deep merge rules with autovivification on writes	11:46 am
coderanger	holoway: In the built-in sense yes, in the "I'm trying to explain what this is to a new user" sense I'm not sure what I would say	11:46 am
nathenharvey	we have 15 minutes left and should probably table this discussion for now	11:46 am
holoway	nah, this is the most important discussion we could have	11:46 am
	:)	11:47 am
	you would say "this is how you set a node attribute"	11:47 am
	etc etc	11:47 am
	and then show the syntax	11:47 am
	and then complain about that time when holoway was ridiculously intransigent	11:47 am
	:)	11:47 am

coderanger	I agree the current behavior is untenable, I'm just trying to keep this grounded in things people know about programming in general, to keep chef relatable to other tools and systems	11:47 am
	this is the same reason we recommend 'foo' instead of :foo	11:47 am
thom	we should not frame this conversation about programming, i think	11:47 am
ortho_stice	Just so long as the assignment syntax allows us to easily determine the attribute "path" and assignment context, I'm happy re: #75	11:47 am
holoway	I'm saying the way to do that is stop pretending it s what they think it is	11:47 am
thom	because that's the point at which novice sysadmins leave and go hurt themselves with salt	11:48 am
holoway	thom: I agree - if it's about the meaning of '=', I work in languages all the time where that means "pattern match"	11:48 am
zts	I'd rather see something that makes sense in the context of chef, than something which kinda resembles one paradigm that some set of people are familiar with	11:48 am
thom	zts: +MAX_INT	11:48 am
holoway	I think https://gist.github.com/danielsdeleo/fe88a8b397...	11:49 am
	is pretty darn clear	11:49 am
lamont_oc	i think that given how hard it is to get #[],# []= to work correctly on attributes that what we have is clearly not a simple hash map -- the code is telling us that holoway is correct	11:49 am
holoway	even though I would rather have .to be =	11:49 am
	but if I can't have my cake and eat it to, I'll take '.to'	11:49 am

	<i>holoway nods</i>	11:50 am
coderanger	holoway: I can accept that statement, I just don't like it :)	11:50 am
holoway	coderanger: I can live with that	11:50 am
	:)	11:50 am
coderanger	holoway: If those were the only two options, I would side with you	11:50 am
holoway	ok - I say the next action here	11:50 am
coderanger	holoway: I don't yet accept that [] is disallowed just because it is associated with the current mess	11:50 am
holoway	is that kallistec updates his RFC with syntax, more examples, and rouds things out	11:50 am
	meanwhile	11:50 am
	coderanger comes up with an alternate proposal with [] and []=	11:51 am
	so we aren't aruging about tiny ones in IRC	11:51 am
	and we can evaluate both with open minds and hearts	11:51 am
	(also, fuck a []! I kid.)	11:51 am
	and lets talk about those updates in the next meeting	11:52 am
ranjib	: -)	11:52 am
jtimberman	seems like a lot of the [] turns into .to(), and thats an easy regexp replacement :)	11:52 am
thom	(doesn't the lifetime of the current syntax imply that [] is disallowed in any new one, lest plentiful people cry?)	11:52 am
ranjib	jtimberman, +1	11:52 am
holoway	thom: I suspect that is true	11:53 am

jdosssey	block syntax for set() could also make for pretty code, eh? set('upstart') { u u.set('runlevels').to('2345') } ?	11:53 am
holoway	thom: but we could feature flag it	11:53 am
thom	holoway: true.	11:53 am
jtimmerman	so i missed some of the conversation but does this deprecate #[], #[]= ?	11:54 am
holoway	okay, I say we table talking about https://github.com/opscode/chef-rfc/pull/79/files	11:54 am
	jtimmerman: no	11:54 am
	jtimmerman: at least, not yet	11:54 am
jtimmerman	ok whew	11:54 am
holoway	(but it probably will in the ned)	11:54 am
	ok	11:54 am
	I have a few things	11:54 am
lamont_oc	yeah, eventually, but node.default['foo'] = will have to still continue to work	11:54 am
nathenharvey	I'll move https://github.com/opscode/chef-rfc/pull/79 to next agenda	11:54 am
jtimmerman	i should probably have paid more attention sooner, but i'll go through #77 again and comment.	11:54 am
martinisoft	It's quite expressive with '.to', I like it. Similar to how chef-sugar gives you the alternative block syntax. *shrug*	11:54 am
coderanger	I think 79 is thumbsup overall, and dialects support gives it a path to be implemented	11:54 am
lamont_oc	i would like to start throwing deprecation warnings on node.foo sooner rather than later	11:54 am
holoway	one is that we're doing a bunch of kick off	11:54 am

	stuff	
	coderanger: me too	11:55 am
	I want to see us put more effort behind the maintenance guidelines	11:55 am
nathenharvey	we have 5 minutes	11:55 am
holoway	in particular, having the various component maintainers take over PR merging and vetting	11:55 am
Tech356	Is there a timeline for the 12.0.4 chef client release?	11:55 am
holoway	we're going to have nathenharvey, lamont_oc and mark anderson help guide that	11:56 am
	so if you want to be a maintainer, get your PR on :)	11:56 am
	with that, anyone have an answer for Tech356	11:56 am
lamont_oc	no, its in flux	11:57 am
holoway	I have the long term answer, which is that by the end of Q1 I want to move to having publicly available nightlies and steady stable release builds moving	11:57 am
	nathenharvey - can you take a note to get an RFC written re: that	11:57 am
martinisoft	Speaking of maintainer stuff, my PR is still waiting: https://github.com/opscode/chef/pull/2592	11:57 am
holoway	martinisoft: not any more	11:58 am
jtimmerman	martinisoft: it's not anymore!	11:58 am
holoway	https://github.com/opscode/chef/pull/2592	11:58 am
lamont_oc	we also wound up with kind of a mess where we need to release 11.18.0 so that we can release chefdk 0.3.6 so that we can release	11:58 am

	<p>chefdk 0.4.0 with chef 12, which is a huge yak shave, and there's broken ci tests and all kinds of fun and nobody is quite certain who owns it all now. discussions are ongoing.</p> <p><i>martinisoft gets a case of whiplash from the merge button getting hit so fast</i></p>	11:58 am
Tech356	holoway: are you looking to go to a timed schedule like every 6 weeks or something?	11:58 am
holoway	although - martinisoft - don't you just want to be the LT?	11:58 am
	Tech356: yeah, but I would rather that be like every 2 weeks	11:58 am
jtimberman	lamont_oc: yeah, let's get that fixed.	11:58 am
holoway	martinisoft: you don't want me making decisions about FreeBSD, I promise you that	11:59 am
	:)	11:59 am
lamont_oc	we did arrive at the decision yesterday that nothing is blocking 11.18.0 so someone just needs to hit the button	11:59 am
jtimberman	"lgtm!"	11:59 am
ranjib	looking at the 12.x bugs, i was thinking of raising an RFC on CI server for chef	11:59 am
martinisoft	holoway: Hah, fair point. LT it is then. Separate PR?	11:59 am
lamont_oc	(although its not that simple)	11:59 am
holoway	martinisoft: yes	11:59 am
martinisoft	holoway: Done. Sending over shortly.	12:00 pm
holoway	ranjib: that would be cool	12:00 pm
nathenharvey	I've got a todo to write up the RFC about the nightlies	12:00 pm
holoway	cool	12:00 pm

	we are at a time - nathenharvey, have you things to say?	12:00 pm
nathenharvey	I do have a list but for the sake of time...	12:01 pm
holoway	ChefConf RFP is probably #1, eh?	12:01 pm
nathenharvey	ChefConf CFP officially closes tomorrow, please make with the submitting of talks	12:01 pm
Tech356	Would it be a good idea to move the announcements to the beginning of the meeting? then talk about RFCs after that?	12:02 pm
nathenharvey	and encourage your friends to do the same. As usual, we're struggling a bit with the diversity of the proposed speakers so all help would be appreciated there	12:02 pm
	un-officially, but I'll tell you now, the CFP will be extended for a week	12:02 pm
	Contributor discount code for ChefConf is CHEFSTAR	12:03 pm
jdossy	If there are topics not listed on the page that you'd especially like to see proposals for, do say	12:03 pm
c_t	will CHEFSUPERNOVA result in an even bigger discount?	12:03 pm
holoway	c_t: no, it just burns you alive with the heat of how awesome ChefConf will be	12:03 pm
	;)	12:03 pm
c_t	ooh, if I go I could bring my little telescope too	12:04 pm
jonlives	nathenharvey: speaker confirmations still gonna be by the 21st?	12:04 pm
nathenharvey	jonlives: yes	12:04 pm
jonlives	gravy, thanks	12:04 pm
jtimmerman	jonlives: you should come over anyway	12:05 pm

	;)	12:05 pm
jonlives	jtimberman: i most likely will tbh :p	12:05 pm
jtimberman	jonlives: excellent	12:05 pm
holoway	okay - I'll see you crazy cats next tie	12:05 pm
	er time	12:05 pm
nathenharvey	jdossey: there's some guidance on the CFP page, but I really would prefer you propose something you really care about and want to share	12:05 pm
	thanks all! See you in two weeks	12:05 pm
	*** MEETING ENDS ***	12:05 pm
jtimberman	woot	12:05 pm
jdossey	sigh, I wish I could stay on this channel but the security overlords at this location block IRC. see yall in a couple of weeks.	12:07 pm
thom	i think https://vine.co/v/MQZmmTnB2iw neatly sums up the last hour	12:08 pm
coderanger	If anyone wants help with a ChefConf proposal, both myself and Jennifer Davis have offered to review and help edit!	12:09 pm
c_t	that's oddly hypnotizing	12:09 pm
martinisoft	coderanger: I'd love help if you are offering	12:13 pm
	Have two I want to send in today	12:13 pm
coderanger	martinisoft: Surely :)	12:13 pm
	martinisoft: email? noah@coderanger.net	12:13 pm
martinisoft	Sure thing, thank you coderanger	12:13 pm
	holoway: Done	12:14 pm
	https://github.com/opscode/chef/pull/2732	
	<i>pwm_away is now known as pwm</i>	12:34 pm

pwm is now known as pwm_away

12:35 pm

Find Us

 [#lincolnloop](#)

 [@BotBotme](#)

 beep@botbot.me

 [Gratipay](#)

© 2015 [Lincoln Loop](#) [Terms](#) [Privacy](#)