

	hello everyone!	
jonlives	hey nathenharvey :)	12:01
		pm
thom	good afternoon!	12:01
		pm
sdelano	good morning thom!	12:02
		pm
jonlives	uk massive represent :p	12:02
		pm
thom	oh, does someone fancy doing some ChanServ magic for me seeing as I have additional permission bits now?	12:02
		pm
jtimberman	thom: yeah i can	12:02
		pm
nathenharv...	we can get started while holoway hugs his family goodbye	12:02
		pm
	agenda is here: https://github.com/chef/chef-community-irc-meet...	12:02
		pm
	<i>martinisoft waves to the crowd while sipping coffee</i>	12:03
		pm
thom	jtimberman: cheers	12:03
		pm
nathenharv...	first order of business is to congratulate and welcome our new Lts and Maintainers! Hugs and high fives for jonlives, thom, and jtimberman	12:03
		pm
cmluciano	o/	12:03
		pm
jtimberman	:D	12:03
		pm
holoway	hello	12:03
		pm
thom	^5s all round	12:03
		pm
martinisoft	Cheers! :)	12:04

		pm
holoway	quick agenda overview	12:04
		pm
	some new maintainers and LTs	12:04
		pm
	new business has a bunch of RFCs	12:04
		pm
	a couple action items from our last meeting	12:05
		pm
	and the sad update on the CBGB is there is no update, unless Nathen has one :)	12:05
		pm
	so - lets get started on maintainers, shall we?	12:05
		pm
	Thom May is now the LT for Core	12:05
		pm
nathenharv...	holoway: we already did maintainers but additional hugs are rarely a problem	12:05
		pm
ranjib	w00t	12:05
		pm
jonlives	#hugops	12:06
		pm
holoway	Jon Cowie rules Enterprise Linux with an iron fist	12:06
		pm
jonlives	\m/	12:06
		pm
holoway	Joshua Timberman is LT of OS X	12:06
		pm
ranjib	nice	12:06
		pm
holoway	and Thom is going to do double duty and deciderate for Ubuntu	12:06
		pm
	one thing I wanted to re-iterate, that got said on a PR	12:07
		pm

	is that the LTs all "report" to me, essentially - so while Thom runs Core, what that actually means is he's the catch-all for decisions that don't land in someone else's lap	12:07 pm
	but he doesn't have a veto over the per-subsystem LTs	12:07 pm
	anything I missed?	12:08 pm
jonlives	that about covers it	12:08 pm
holoway	otherwise, let's move on to https://github.com/chef/chef-rfc/pull/86	12:08 pm
	ranjib, you want to review?	12:09 pm
ranjib	im still working on btm's comment	12:09 pm
	i have to update the rfc text as well as the PR	12:09 pm
thom	i like btm's suggestion, fwiw	12:09 pm
ranjib	to reflect the proposed flags, instead of --github. but i like btm's suggestion	12:09 pm
jonlives	yeah	12:10 pm
holoway	one thing I want to say, Im against any kind of "freeze" of solo	12:10 pm
	it makes our lives easier to get to a place where we can have only local mode filling all the requirements, but until then, I'm not going to condemn that use case to purgatory	12:10 pm
adamedx	interesting	12:10 pm
___ranjib	+1 localmode is buggy	12:10 pm
kallistec	the overall behavior requested is totally reasonable, but the way all the tarball	12:10

	stuff happens in solo now is pretty bad from an implementation standpoint. would it be bad to move fetch-code-from-random-places to some other program and have ppl do fetch-my-code && chef-solo ?	pm
holoway	and, just so we're clear, when we get to a place where local mode can replace the guts of solo, we're keeping a solo binary anyway, because we're not going to break the world	12:11 pm
__ranjib	Ya, moving the fetch code is a good idea	12:12 pm
kallistec	for example, the code download stuff all happens before we get our error handling framework initialized so the UX is bad for failures that happen there	12:12 pm
mivok	kallistec: I like that idea - it would allow for alternate fetch code too such as git pull	12:12 pm
__ranjib	How u deploy the fetch code, if its outside solo?	12:12 pm
holoway	it would have to be in chef	12:12 pm
	chef-fetch	12:12 pm
	or whatever	12:12 pm
kallistec	yeah, it would ship with chef, whether it's a different git repo I don't care so much about	12:13 pm
holoway	it shouldn't be a different git repo	12:13 pm
jonlives	as long as we don't start needing a Fetchfile :P	12:13 pm
holoway	I'm not sure I think it isn't just a feature of the chef-client	12:13 pm
thom	that seems like it would break people almost as hard as losing a solo binary	12:13 pm
holoway	right?	12:13 pm
	that's good for us, I'm not sure it's good for solo users	12:13

		pm
kallistec	thom: moving code fetch stuff outside?	12:14
		pm
thom	yeah	12:14
		pm
holoway	yeah	12:14
		pm
mivok	would an option such as 'chef-solo --fetch-command' work?	12:14
		pm
holoway	it's already inside	12:14
		pm
	you can't remove it in this iteration	12:14
		pm
thom	if we can remove it for local-mode, then great	12:14
		pm
	but i think solo has to keep it	12:14
		pm
kallistec	right, don't remove the tarball stuff that's there without at least long deprecation cycle	12:14
		pm
holoway	I really think btm's suggestion on the PR is right, andw e should just fix the UX stuff, and make it work for solo + local	12:15
		pm
mivok	you could implement backwards compatibility by making the tarball fetch option expand to --fetch-command='chef-fetch URL' or similar	12:15
		pm
holoway	but lets take this to the PR	12:15
		pm
kallistec	right, the old tarball thing could delegate to the new stuff and give us a big improvement, then new features require you to use chef-fetch	12:15
		pm
holoway	I'm not sure I like the ux of chef-fetch	12:16
		pm
	nathenharvey: can you splat this conversation into the PR, and lets take it up async there	12:16
		pm
	https://github.com/chef/chef-rfc/pull/89	12:16
		pm

nathenharv...	holoway: will do!	12:17 pm
jonlives	I punga withnale, who wrote that PR, to see if he's around	12:17 pm
	but tl;dr is that it's gonna be hard to do well.	12:18 pm
holoway	my memory might be bad, but it feels like this only worked the way he's describing in puppet if you are populating the directory remotely	12:19 pm
jtimberman	I recall the feature from Puppet but I never used it	12:19 pm
jonlives	what's puppet? :p	12:19 pm
holoway	ie: if the resource owns all the data, it will purge the shit that it doesn't recognize	12:19 pm
jtimberman	jonlives: a tool referenced in the RFC :)	12:19 pm
holoway	to be specific, I think saying remote_directory gets a purge resource is fine	12:20 pm
	s/resource/parameter/	12:20 pm
lamont_oc	we've got that already as a flag to remote_directory	12:20 pm
holoway	ya	12:20 pm
jonlives	right	12:20 pm
holoway	but the magic here otherwise will be super rough	12:20 pm
thom	and will result in user's files going missing in exciting ways	12:21 pm
holoway	I think we say no to this RFC, point to the flag in remote_directory for the most common use case here, and refer to the zap cookbook for the more aggressive (and probably yucky) version	12:21 pm

lamont_oc	what this rfc asks for is simply tough to implement with LWRP sub-collections and i'm not sure what happens to sub-resources that some providers use	12:21 pm
mivok	I like how the zap cookbook does things - you can basically tell it exactly what resource types manage files in a given directory (e.g. I could implement it for sensu_check and friends by telling it only the various sensu classes can create files in this dir).	12:21 pm
jonlives	+1 to holoway's suggestion	12:21 pm
thom	holoway: +1, yeah	12:22 pm
holoway	mivok: I do too, but I think if you want that, you can rock the zap cookbook	12:22 pm
mivok	agreed	12:22 pm
jtimberman	+1 to that, and +1 to the zap cookbook	12:23 pm
btm	However we say no to this RFC, I'd like to request that whoever writes the words do so with some extra care. I feel like we treated withnale a bit like he stepped into a developers hornets nest and I'd like to be more welcoming than that.	12:23 pm
lamont_oc	+1	12:23 pm
holoway	ok, closed	12:24 pm
kallistec	lamont_oc's comment about a "resource log" is basically what we already do for reporting, we'd just need to make that accessible and public	12:24 pm
github-bawt	[chef-rfc] adamhjk closed pull request #89: Allow directories to be managed idempotently (master...idempotent_directories) http://git.io/bWMb	12:24 pm
martinisoft	Agreed. -1 on that RFC	12:24 pm
holoway	btm: if I wasn't gentle enough, please to gentle it more	12:25 pm
	https://github.com/chef/chef-rfc/pull/91	12:25

		pm
	kallistec?	12:25
		pm
kallistec	yep, this is the HTTP api for storing and fetching policyfile lock documents	12:25
		pm
lamont_oc	kallistec do we catch the resources which are created in providers in core chef?	12:25
		pm
kallistec	lamont_oc: I'll catch up on that later, though we -1'd so not sure how relevant it is	12:26
		pm
	anyway, back to PR91	12:26
		pm
	most contentious issue so far is whether the server should be very strict about checksumming policyfile lock content	12:27
		pm
	and also cookbook_artifact content	12:27
		pm
	https://github.com/chef/chef-rfc/blob/master/rf...	12:27
		pm
	So far I've not attempted to write the feature in such a way that it provides strong assertions about the content of these things	12:28
		pm
holoway	boy, I sure do like the idea of having those IDs be a stronger guarantee	12:28
		pm
kallistec	the client side implementation makes IDs of the various objects by checksumming canonical content	12:29
		pm
jonlives	without having thought too much about impl, I like the idea of the checksums being pretty strict	12:29
		pm
kallistec	but the user can override this to build an arbitrary workflow	12:29
		pm
	and the server doesn't care	12:29
		pm
holoway	to get that, tho, you would need to move from MD5 at the bottom to something like SHA2 all the way up?	12:29
		pm
	right	12:29
		pm

	the server just accepts the checksums the client gives it?	12:29 pm
jonlives	would an optional strict_mode be viable / necessary?	12:30 pm
kallistec	yes, current code is the server only has mild restrictions on length and allowed chars	12:30 pm
holoway	why would we ever want to not be strict?	12:30 pm
jonlives	holoway: in case people were overriding the IDs and didn't want the server to care, per kallistec's comment above	12:30 pm
	not saying we shouldn't be strict (I think we should), merely mooted the possibility	12:31 pm
holoway	yeah	12:31 pm
kallistec	the idea is that being super strict may proscribe some elements of your workflow and you'd disagree with that	12:31 pm
holoway	I suppose it means you really, really can't mutate something	12:31 pm
	which - yeah, um, don't mutate it	12:31 pm
kallistec	but you still want the server side benefits of not using the server-side dep solving	12:32 pm
adamedx	benefits are a good carrot though :)	12:32 pm
jonlives	I'd say at least defaulting to strict would be a sane idea	12:32 pm
adamedx	yeah	12:32 pm
holoway	my \$0.02 is this - lets declare that we want strict, and crypto secure would be even better	12:32 pm
kallistec	in my mind this is just an unknown unknown kind of thing	12:32 pm

holoway	don't block shipping v0	12:32 pm
	stick it in the design, but get a usable one out the door that makes it clear it doesn't have this functionality	12:33 pm
	triangle shaped slices and all that	12:33 pm
	thom?	12:33 pm
kallistec	there's the one semi-ugly compromise that I came up with, which is that IDs ending in ! get strict mode	12:33 pm
jonlives	i still vote strict by default	12:33 pm
holoway	I'm saying strict-only	12:33 pm
markander...	I think there's benefit to strict w/o crypto secure; protects against accidents but not overt attack.	12:33 pm
jonlives	yeah	12:33 pm
thom	yeah, i agree	12:34 pm
holoway	ok - kallistec or nathenharvey, can you add this to the PR thread?	12:34 pm
	strict only, we're okay with shipping early implementations without it, but lets get tehre	12:34 pm
kallistec	alright, so v0 will be in violation of that spec and we'll get there	12:34 pm
thom	and as kallistec says, there're unknown unknowns that we can nail down after the fact	12:34 pm
holoway	yep	12:34 pm
kallistec	yeah?	12:34 pm

thom	+1	12:34	pm
jonlives	+1	12:34	pm
holoway	+1	12:34	pm
ssd7	+1	12:35	pm
holoway	next up https://github.com/chef/chef-rfc/pull/92	12:35	pm
	https://github.com/chef/chef-rfc/pull/92 for better clicking	12:35	pm
kallistec	also, I'm in the process of implementing all that right now so if anyone hates any part of it, please to be commenting	12:35	pm
holoway	mparadise - you around?	12:35	pm
mparadise	Yep, I'm here	12:35	pm
holoway	give us the tl;dr	12:35	pm
mparadise	So a bit of background on this one - we're in the middle of adding keys rotation, and this will require changing the behavior of a couple of existing endpoints.	12:36	pm
	We don't want to break various clients of the server, and so RFC-92 was born - providing a way to version server features in a default-conservative means	12:36	pm
	by having the client sending version in a request header, and if no version is specified the oldest supported behavior for the endpoint is used.	12:36	pm
	</tl;dr>	12:37	pm
jonlives	my only concern there is that "oldest supported version" might be deprecated long after the release where new behavior was added	12:38	pm
	ie, your api client might break when the version it defaults to changes	12:38	pm

	could be a bit surprising	12:38 pm
jkeiser	Yeah, I'm concerned about "stable" -- are we planning to leave "oldest supported version" stable across releases?	12:38 pm
	If we're not, stable isn't really stable	12:38 pm
sdelano	jonlives: since the API has been pretty static, we could take the current implementation as version 0 (or -1) and decrement that when appropriate	12:39 pm
nlloyds	Any thoughts around including a list of capabilities rather than relying on version numbers?	12:39 pm
jonlives	I like jkeiser's idea of having stable stay stable for an entire release	12:39 pm
jkeiser	It feels like we might want to force people to just give a number if they want a particular version, that way they don't get surprised when it changes	12:39 pm
mparadise	This isn't specifically tied to product version (12.x, etc) for that reason - those behaviors and requirements can be defined separately	12:39 pm
	as part of the support policy.	12:40 pm
jkeiser	Actually, my concern was more that if you upgrade from 12 to 13, clients which have been expecting to continue to work because they specify "stable" will stop working in an unclear way	12:40 pm
holoway	yeah - I think in practice, though, you don't really ever want to use the 'stable'	12:40 pm
thom	i'm always a little wary of versioned server APIs, because it's incredibly hard to get people to upgrade without simply breaking them and waiting till they fix	12:40 pm
kallistec	it'll be very simple for the chef-client side stuff to add a version number and that's how I'd do it	12:40 pm
holoway	we're going to hard code the version in the client to a number	12:40 pm
jkeiser	If we shouldn't use stable, should we remove it? Does it serve a purpose?	12:40 pm
holoway	preciesly because of what jonlives said - stable can move, and you won't know the new endpoint, and blammo	12:40 pm

kallistec	also, on the client side we could add support for a your-version-will-die-soon header	12:41 pm
jonlives	I'm also thinking of third party API clients	12:41 pm
jkeiser	oh! It's a client thing? I'm OK with that I guess, though, who is supposed to use it? knife plugins would have issues	12:41 pm
holoway	jonlives: me too	12:41 pm
mparadise	jkeiser: that makes sense to me in this context. If there's no baseline for 'stable' then it's not stable.	12:41 pm
holoway	jonlives: they'll totally write to the 'stable' api, and then we'll move it, and then they'll break :)	12:41 pm
mparadise	Though we could define what 'stable' is for a given server product version?	12:41 pm
jonlives	I was more thinking that a given stable will remain supported for an entire release, eg 12.x.	12:41 pm
mparadise	^	12:41 pm
jonlives	and the expectation will be that stable will change in 13.x	12:42 pm
holoway	I'm good with that	12:42 pm
	the issue though	12:42 pm
	is that you really want the integer for 12	12:42 pm
jkeiser	OK, then how about stable-12 and stable-13 or something	12:42 pm
mparadise	Though again, I'd like to separate that out from what this RFC does - which is talk about the versioning mechanism. I agree that stable should be supported for a release,	12:42 pm
holoway	ie: if you have client code that will break on 13	12:42 pm

mparadise	but think that belongs in the support RFC	12:42 pm
jonlives	holoway: good point	12:42 pm
jkeiser	Or rename stable to something that doesn't imply that it's, you know, stable	12:42 pm
sdelano	if stable stays static across an entire release, then you're never going to get new features	12:42 pm
	I don't like the word "stable" for that reason	12:43 pm
jonlives	sdelano: i mean stable will be a guaranteed baseline	12:43 pm
thom	having a numerical version and a deprecation schedule is probably most easily communicated	12:43 pm
jonlives	maybe call it "compat" or something	12:43 pm
mparadise	sdelano: stable is really minimum or compatibility. So - s/stable/baseline?	12:43 pm
jkeiser	For some definition of "guaranteed"	12:43 pm
holoway	well, new endpoints are safe to add	12:43 pm
jonlives	having a deprecation schedule is tough, because people won't find it and stuff will break in surprising ways.	12:43 pm
jkeiser	Yeah, I like the number. I can't imagine specifying "stable" and really meaning "I'm OK with this changing across releases"	12:43 pm
jonlives	if at least it's tied to a release, that's a bit more expected I'd have thought	12:44 pm
	I can live with numbers too	12:44 pm
holoway	right - and as maintainers, we can basically say	12:44 pm

thom	jonlives: indeed. but otherwise you end up with 30 different API versions and then you die	12:44 pm
sdelano	holoway: but for something like key rotation you need to change the users and clients APIs to make it all make sense (because those currently return one key)	12:44 pm
jkeiser	Like, if I specify a number, my intent is to specify the number I want, whose responses are the same no matter what. And I want it to break the day the server stops supporting my number.	12:44 pm
sdelano	in this world, that doesn't become stable until 13?	12:44 pm
holoway	ahhhh	12:44 pm
	does the header apply per-endpoint?	12:44 pm
sdelano	or 12.next?	12:44 pm
thom	jkeiser: exactly	12:44 pm
holoway	or is it universal?	12:44 pm
mparadise	It's per-endpoint holoway	12:44 pm
	Well - to clarify:	12:45 pm
jonlives	we're going to end up with a semver-api-versioning discussion, i can see it coming :p	12:45 pm
mparadise	the version value is universal. Each endpoint can implement support for it at current version level as the minimum.	12:45 pm
jkeiser	I'm just confused as to the use case. If we can figure that out, we can decide whether "keep," "rename" or "remove" is the right thing with stable, which would let us move on and talk about the rest of this proposal, which is super useful	12:45 pm
sdelano	jonlives yay! calling schisamo!!!	12:45

		pm
mparadise	jkeiser: specifics for our first use case: the clients endpoint needs to return a list of keys instead of a single key.	12:45 pm
	But existing clients will not like that.	12:46 pm
	*chef-clients/web-ui, etc	12:46 pm
mmzyk	I struggle with this RFC b/c I feel it adds a lot of complexity that I'm not convinced is going to give us anything in return. I'm used to entire APIs being versioned, but this version at the endpoint level. It becomes a level of granularity that I think will introduce confusion.	12:46 pm
nathenharv...	15 minute warning	12:46 pm
mparadise	So we only return the list if the client says it is ready for it (eg, server api version 1)	12:46 pm
jkeiser	I'm curious why we don't design the endpoint for backcompat, but that's a different issue (let's chat after)	12:46 pm
sdelano	and what mparadise is saying is that they'll need an easy way to opt into these new features (or get them by default when they upgrade)	12:46 pm
	jkeiser: this conversation is how we add back-compat into the endpoint	12:47 pm
	if the server gets no version header, behave the old way, otherwise behave the new way	12:47 pm
	we've done this in the past by using the chef-client version header as a key	12:47 pm
jonlives	yeah my concern was more around what determines the old way and when it changes	12:47 pm
jkeiser	Yeah. I'll chat with you after about what I mean, it's irrelevant to this discussion, which is useful on its own	12:47 pm
kallistec	jkeiser: at some point you really will need to change the document format that you return from GET /clients/foo	12:47 pm
holoway	sdelano: for that use case, wouldn't feature flags be better than versions?	12:47 pm

sdelano	but there are so many tools outside of chef-client now that that way no longer makes sense	12:47 pm
kallistec	otherwise you're just moving the versioning around	12:48 pm
jkeiser	kallistec: yep. I think we have to now, too. Just not in as jarring a way as implied	12:48 pm
	But, irrelevant	12:48 pm
nathenharv...	holoway: I move that additional RFCs on our agenda move to the next meeting so that we have time for previous business and updates.	12:48 pm
holoway	agreed	12:48 pm
jonlives	here's a quick thought	12:48 pm
holoway	lets take this to the PR	12:48 pm
ssd7	holoway: By feature flags, what do you mean? If they are server-side flags, how would support a mix of client versions?	12:48 pm
holoway	kallistec/nathenharvey - please paste our convo into the pr	12:48 pm
jkeiser	So, what we're saying is "by default, clients will not get Chef Server 13+ behavior, unless they opt in with a flag, and we think the 'stable' version is the right way to specify that default state"	12:48 pm
	Yes?	12:48 pm
	ok	12:49 pm
	taking it to pr	12:49 pm
holoway	ssd7: I mean rather than saying I support version X, say I support feature Y	12:49 pm
adamedx	ah	12:49 pm

	like html5 recommendations	12:49 pm
holoway	"I can deal with multiple clients"	12:49 pm
ssd7	holoway: Yeah, I think a neat alternative would be a header that specified some set of descriptive behavior	12:49 pm
	right	12:49 pm
sdelano	holoway: and those become headers?	12:49 pm
holoway	yes	12:49 pm
mparadise	holoway: wouldn't that get cumbersome as the quantity of changed and new APIs grows? Too, if it's for a new API, there's no reason for a client to say it's supported - they're already asking for it.	9:49 AM
holoway	have N X-Ops-Feature headers	9:49 AM
ssd7	X-Features: multi_key; awesome_new_thing; wombats	9:49 AM
kallistec	holoway && ssd7 that would basically be moving versioning into content negotiation	9:50 AM
holoway	anyway - lets take it to the PR	9:50 AM
	kallistec: precisely	9:50 AM
jonlives	I like the idea of stable + feature headers, then next version becomes all those feature headers as stable	9:50 AM
sdelano	yes, let's take this all to the PR	9:50 AM
holoway	as the good lord Roy intended it	9:50 AM
	:)	9:50 AM

nathenharv...	10 minute warning	9:50 AM
sdelano	jonlives: PR :) (because I like that too)	9:50 AM
nathenharv...	I'll copy this discussion to the PR	9:50 AM
jonlives	ack	9:50 AM
holoway	are we any closer to https://github.com/chef/chef-rfc/pull/82	9:51 AM
	moving on to old bidness	9:51 AM
	I think we should follow btm and thom's suggestion, and get really clear about the partials for the bootstrap	9:52 AM
	and then let you override them easily	9:52 AM
thom	i'd like to see the steps stuff in the rfc (and I don't mean the pop band)	9:52 AM
holoway	which is the actually awesome thing	9:52 AM
	ranjib: do you concur?	9:52 AM
btm	yeah, I'd purchase beers for those steps.	9:52 AM
holoway	cool	9:53 AM
	lets do it	9:53 AM
	https://github.com/chef/chef-rfc/pull/82	9:53 AM
	what I mean is	9:53 AM
	https://github.com/chef/chef-rfc/pull/83	9:53

		AM
	sounds like it's now all +1	9:53
		AM
	30 seconds of "oh no don't do this", and then I'm stamping it as gtg	9:53
		AM
thom	do it do it	9:53
		AM
kallistec	yes, I +1'd	9:54
		AM
lamont_oc	+1	9:54
		AM
holoway	ship stamp done, take it away rfc-editors	9:54
		AM
	https://github.com/chef/chef-rfc/pull/77	9:55
		AM
nathenharv...	5 minutes	9:55
		AM
holoway	I think we just need the RFC to be updated to reflect all the conversation, and bring it back up	9:55
		AM
	lets do status real fast	9:55
		AM
	as I said at the beginning, I still have CBGB work to do	9:55
		AM
kallistec	yes, no movement on that one due to being busy w/ other stuff	9:55
		AM
	^^ pr77	9:55
		AM
holoway	sorry for not having an update	9:55
		AM
	nathenharvey - status?	9:56
		AM
nathenharv...	I hit your inbox twice already today with this message so I may as well repeat it here: we want you at ChefConf. Contributor discount code is CHEFSTAR. If	9:56
		AM

	you can't get there but want to, reach out and we'll do what we can to help.	
	Last week we had two new people start on my team Thom May who's helping with all things open source engineering and Nell Shamrell who's working on the Supermarket.	9:56 AM
	that's my update	9:56 AM
holoway	sdelano/adamedx?	9:56 AM
adamedx	We released chef-client 12.1.0 on Tuesday with all its multipackage and audit mode goodness	9:56 AM
	But we broke some providers by removing a public interface	9:56 AM
	https://github.com/chef/chef/pull/3012	9:56 AM
sdelano	holoway: I have a wall of text :D	9:56 AM
adamedx	new release hopefully this week to fix that regression	9:56 AM
	sdelano has a lot to say	9:57 AM
sdelano	Better sent to the mailing list this late in the meeting?	9:57 AM
holoway	sure	9:57 AM
adamedx	maybe still talk about in next meeting tho	9:57 AM
holoway	lets get it early on the agenda	9:57 AM
sdelano	Sure thing	9:57 AM
holoway	we may have to move to doing this weekly if we want to keep up	9:57 AM
jtimberman	from the agenda today i have two brief things	9:57

		AM
	chef-server cookbook: version 3.0 was released last week. This cookbook is as it was in it's previous iterations, a "standalone open source" implementation. For clustered implementations with add-ons, see chef-server-cluster. For background see https://github.com/chef-cookbooks/chef-server/p... - commit 0f2d123ad9ebb40ac18fdabdeeee2d66735604bbe	9:58 AM
	chef-client cookbook: I posted on the mailing list about refactoring the cookbook. That issue is https://github.com/opscode-cookbooks/chef-clien...	9:58 AM
jonlives	holoway: yeah the agenda is ever growing :p	9:58 AM
nathenharv...	sdelano: I'll put you first on next agenda	9:58 AM
holoway	thanks jtimberman	9:59 AM
thom	jtimberman: the client refactor sounds awesome	9:59 AM
holoway	what does the room think of moving to weekly? one thing that may happen is I won't be able to make it every time	9:59 AM
jonlives	same here, but otherwise in favor	9:59 AM
holoway	but we could put a list of rotating LTs to take over in my absence	9:59 AM
jonlives	that sounds reasonable	9:59 AM
thom	we seem to rush to get this done every time, so i'm mostly for it	9:59 AM
jkeiser	It would be nice to be able to move quicker on RFCs, we seem to have a lot	9:59 AM
lamont_oc	+1 weekly	10:00 AM
btm	+1	10:00 AM
thom	and by mostly i mean +1	10:00

		AM
jkeiser	so yeah +1	10:00
		AM
kallistec	+1	10:00
		AM
jtimberman	+1 wweekly	10:00
		AM
jkeiser	Also, it's a ton easier for me to remember something that happens every single week, especially in the morning	10:00
		AM
sdelano	+1	10:00
		AM
adamedx	+1	10:00
		AM
martinisoft	+1 weekly too	10:00
		AM
nathenharv...	ok, next meeting will be 12 March	10:00
		AM
adamedx	same thing for me jkeiser	10:00
		AM
lamont_oc	jkeiser: +1	10:01
		AM
holoway	ok	10:01
		AM
nathenharv...	weekly also helps when we need to skip a meeting b/c of things like ChefConf	10:01
		AM
holoway	nathenharvey: lets move it to weekly and set up jonlives and thom as possible alternate meeting runners	10:01
		AM
nathenharv...	holoway: will do	10:01
		AM
sdelano	also, +1 to nathenharvey sending out reminders to everyone (heart)	10:01
		AM
	thanks	10:01
		AM

nathenharv...

*** MEETING ENDS ***

10:01

AM