

nathenharv...	***** MEETING STARTS *****	
	Today's agenda can be found here - https://github.com/chef/chef-community-irc-meet...	12:01 pm
	we'll get started with the various updates and then move on to the open PRs we have against the RFC repository	12:03 pm
	And I'll start with the community update	12:03 pm
	Nominations for the CBGB are all in but we're a bit behind on pulling together the voting process. We are currently on track to have voting open tomorrow	12:03 pm
	Watch for an email to the mailing list with a link to the voting form. The message will be posted in the chef category on discourse - https://discourse.chef.io/c/chef	12:04 pm
	The Chef Community Summit was held in London last week. Thanks to everyone who participated. The format was a bit different than the Seattle Summit	12:05 pm
	We had a full day of presentations followed by a full day of open spaces.	12:05 pm
	The presentations were recorded and we'll be making those recordings available soon	12:06 pm
	The final community update is that the mailing list migration seems to be settling down. One of our last hurdles was fixing up the subscription settings for migrated users. I think we're in a good place now but encourage everyone to login to http://discourse.chef.io and make sure your preferences are set to your liking	12:07 pm
	That's the community update for this week.	12:08 pm
	I'll turn it over to adamedx for an update next	12:08 pm
adamedx	Thanks nathenharvey	12:08 pm

adamedx	Pretty brief: chef-dk 0.10.0 was released last week, so please grab it	12:09 pm
	it restores the creation of Berksfiles in generators that were removed in 0.9.0, though you can still tell the generators to emit policyfiles if you've already moved on to that workflow	12:10 pm
btm	*monday	12:10 pm
adamedx	also contains knife-windows 1.1.1, which mainstreams the improvements in security and usability for bootstrapping windows nodes via winrm	12:10 pm

	We are working on a Chef Server release, though I'm not ready to give an ETA	12:11 pm
	a big feature of that release is native support for PolicyFiles rather than an "emulation"	12:11 pm
	the native capability allows full management of policies (e.g. enumeration, deletion in addition to being able to use them in deployment)	12:12 pm
	so that server release signals the point at which everyone should be able to fully jump on policyfiles without special flags, etc.	12:12 pm
	that's it for me nathenharvey	12:12 pm
kallistec	To be clear, the new feature in Chef 12.3 is that node objects have policy_name and policy_group, so you can search on them and so on. This also enables bootstrapping nodes in policyfile mode via the -j JSON	12:12 pm
	and changing a node's policy_name/policy_group via `knife node edit` and so on	12:13 pm
nathenharv...	Thanks, adamedx and kallistec!	12:13 pm
adamedx	good clarifications kallistec	12:13 pm
nathenharv...	Let's move on to our first PR - Cookbook Segment Deprecation - https://github.com/chef/chef-rfc/pull/161	12:14 pm
kallistec	Chef client 12.5 already has the changes you need for those feature	12:14 pm
	*s	12:14 pm
nathenharv...	lots of discussion in the PR but it doesn't look like lamont is around this morning...	12:15 pm
coderanger	+1	12:16 pm
nathenharv...	I suggest we skip ahead to PR-163, Update versioning for a CI/CD future (and present) - https://github.com/chef/chef-rfc/pull/163	12:16 pm
kallistec	ssd7 wrote "I think for new API responses, the RFC should include example payloads and example returns." I think this is a good idea if we know enough about what the implementation would look like	12:16 pm
nathenharv...	btm: want to give us a quick update on 163?	12:16 pm
btm	we have CI/CD, we want to release faster, we need to automate versioning to do that. it'll make releases have significantly less ceremony internally and easier to do.	12:18 pm

	<i>NOTICE: [13chef-rfc] 15nathenharvey pushed 1 new commit to 06discourse_is_live: 02http://git.io/v8pRn</i>	12:19 pm
	<i>NOTICE: 13chef-rfc/06discourse_is_live 1458ba4b1 15Nathen Harvey: add a note about the date of the cutover</i>	12:19 pm
	pretty much everything about major/minor stays the same, but patch becomes build, and moves fast. which might scare some people, who we will then hug as defined in the hug RFC.	12:19 pm
agentmeer...	is there really a hug rfc?	12:20 pm
coderanger	Yes	12:20 pm
btm	agentmeerkat: there is a proposal https://github.com/chef/chef-rfc/pull/164	12:20 pm
someara	*sigh*	12:20 pm
agentmeer...	That's freaking awesome	12:21 pm
kallistec	I'm +1 on 163 and I don't really care about when the build number gets reset	12:21 pm
adamedx	yeah +1	12:21 pm
stevenmur...	I'm of the same view as kallistec there..	12:21 pm
ssd7	+1, also don't care about build number.	12:21 pm
stevenmur...	+1 and don't care about build number reset	12:22 pm
martinisoft	+1 from me as well. Would love to see faster releases with the help of this	12:22 pm
btm	yeah, lost in the weeds is the discussion where we just wrote to reset build on major/minor bump because we didn't really care otherwise.	12:22 pm
ssd7	I think martinb3's unease about our ability to signal risky upgrades is merited; however, I think the way forward there is to focus on how reduce the risk of big changes like that.	12:22 pm
coderanger	From a user support POV, this is going to confuse a lot of people.	12:23 pm
	See also why Ruby stopped doing this	12:23 pm
	Having smaller numbers makes them more understandable to people	12:24 pm
martinb3	I love the idea of major.minor (+ some build id), for the record. I just want to be sure it's as actionable as possible for the people using it/reading it.	12:24 pm
btm	we could just not talk about BUILD most of the time, we've kind of been doing that lately anyway, e.g. Chef 12.5.	12:24 pm

martinb3	That's what I'd expect would happen ^	12:25 pm
coderanger	Yeah, if we do a lot more feature releases that would be fine	12:25 pm
adamedx	do we need an update to the rfc then btm?	12:25 pm
martinb3	Something more like https://github.com/DamonOehlman/slimver-spec	12:26 pm
	Is what I was aiming at.	12:26 pm
btm	for the time being, we're just doing feature releases because there's a lot of ceremony to make them, and we do patch releases to fix the regressions in those. I see that continuing for a while.	12:26 pm
nathenharv...	oh, I forgot to mention, holoway is not around today neither are jonlives or thom so adamedx is playing the role of decider	12:27 pm
coderanger	btm: Then this would be a bad change IMO	12:27 pm
btm	adamedx: I'd be interested if people are into going as far as having "chef --version" print "12.5 build 42"	12:27 pm
coderanger	btm: Would have to get in the habit of doing more fine-grained feature releases	12:27 pm
ssd7	Yes, I think part of the point of many of these changes is to work as a forcing function towards that as well	12:27 pm
adamedx	so there are a lot of +1's here, do you want to take additional feedback into account then btm?	12:27 pm
martinb3	We could also make a more thorough recommendation or blog post for users on the intent of the versioning scheme. To encourage understanding of what each (major, minor, build or patch) actually means in terms of risk. I think that would mitigate a lot of what my concerns about communicating risk better.	12:28 pm
btm	yeah, i'll commit to making sure we get a blog post before a crazy build version goes out.	12:28 pm
coderanger	Basically I can summarize my opposition to this as "what even was 1.9.3-p327?"	12:29 pm
ssd7	Best ruby ever?	12:29 pm
btm	adamedx: I think I'd like to roll ahead with the +1s, then open up another PR against it for discussion about how we communicate build version in the project. this way, we can drastically clean up the release process and get more feature releases regularly out like coderanger referred to.	12:30 pm

adamedx	Seems ok btm	12:30 pm
martinb3	My concern (generally, not to the RFC specifically) was 99% around people pinning to major and allowing minor to float, and then being surprised at breakage. But if they stay at patch, they don't even get basic/simple/trivial bugfixes. I'd love to make it where they can choose flavors of Chef like (may break on upgrade, 99% sure it won't break on upgrade).	12:30 pm
	I'll stop injecting asides :)	12:30 pm
kallistec	I don't think that analogy is truly accurate because at the time, major, minor, and patch were all part of "major" from a breaking change perspective	12:30 pm
adamedx	all right, this one is approved based on +1's, with btm's actions above noted for communication amendments	12:31 pm
coderanger	kallistec: If features releases only happen every 4 months, that's not substantively different :)	12:31 pm
kallistec	martinb3: that's what the channels are for. Current == latest successful build	12:31 pm
adamedx	Good discussino for btm's upcoming pr	12:31 pm
	nathenharvey, I think we can move to 166 now	12:32 pm
martinb3	btm: I'd volunteer to also blog or help blog where needed to help folks better understand the intent of the scheme and the risks :)	12:32 pm
adamedx	awesome martinb3	12:32 pm
kallistec	martinb3: and stable == thing that's had more substantive testing	12:32 pm
adamedx	https://github.com/chef/chef-rfc/pull/166	12:32 pm
nathenharv...	OK...moving on to 166...	12:32 pm
adamedx	this one is yours, right nathenharvey?	12:33 pm
nathenharv...	I wanted to indicate that the mailing list migration has been completed. instead of bike shedding around a new status for the RFC, I'm taking Adam Leff's suggestion and simply adding a line to the RFC	12:33 pm
ssd7	I think this is mostly just up to rfc-editors as to what to do?	12:33 pm
	oh, nvm, the "Final" change was removed since I read it :D	12:34 pm
	+1	12:34 pm
nathenharv...	seems like a very straight forward change now that I think we just need some +1s on and we can move on	12:35 pm

adamedx	seems administrative at this point folks, more +1's?	12:35 pm
martinisoft	+1	12:35 pm
btm	nathenharvey: so I'm fine with leaving status as accepted, but, reading that line as a new person you'd be like "okay, so they moved the mailing list from lists.opscode.com to discourse.chef.io, and now what are they proposing?"	12:35 pm
	nathenharvey: can we hash that out later? like moving it up and changing a couple words?	12:36 pm
nathenharv...	btm: sure, I'll make one more update that makes it clear to the reader that it's done	12:36 pm
btm	nathenharvey: as is, it reads much more "setting the stage" than "closing the curtains"	12:36 pm
	+1	12:36 pm
adamedx	assuming deciderator can still vote, I'm +1	12:37 pm
	so this is accepted, btm and nathenharvey will tweak	12:37 pm
nathenharv...	OK. I'll make one more change to address btm's concern and then we'll accept	12:37 pm
	:)	12:37 pm
martinb3	+1 from me, I'm sure nathenharvey can make it clear	12:37 pm
kallistec	same	12:37 pm
	+1	12:37 pm
adamedx	cool, thanks all	12:38 pm
nathenharv...	lamont_oc has arrived so let's circle back to those PRs	12:38 pm
	Cookbook Segment Deprecation - https://github.com/chef/chef-rfc/pull/161	12:38 pm
adamedx	https://github.com/chef/chef-rfc/pull/161	12:38 pm
nathenharv...	lamont_oc: give us the latest?	12:38 pm
lamont_oc	well where are we at with merging the ohai plugin one?	12:38 pm
	ah its been merged	12:39 pm
	catching up on the segments one	12:39 pm
martinisoft	Yup it was merged	12:39 pm
lamont_oc	so for the segments one, i'm unclear if we can just leverage server API	12:42 pm

	versioning and repurpose the top level key "files:" mean all-the-things, or if we have to do the "all_files:" dance where the server responds with both for back-compat still.	
coderanger	The latter seems safer	12:42 pm
lamont_oc	it does	12:42 pm
	steven seemed to be arguing a bit that we could the former, although he didn't take it all the way	12:43 pm
	*do	12:43 pm
coderanger	It's a minor level of semantic grossness on an internal data structure, given the scale of the change I suspect we all have bigger fish to fry :)	12:44 pm
lamont_oc	since i haven't playing with the server API versioning, i'm just unclear what words to write where	12:44 pm
ssd7	Right, version NEXT responses can be whatever we want	12:44 pm
	they could be completely different as they will only be served to clients expecting them	12:44 pm
lamont_oc	how does that negotiation happen?	12:44 pm
ssd7	When the client makes a request, it includes a header with the version it wants	12:44 pm
	if no header is included, it gets v0	12:45 pm
nathenharv...	5 minute warning	12:45 pm
ssd7	if a header is included but we don't know what that version is, it gets a 4XX	12:45 pm
	<i>NOTICE: [13omnibus-chef] 15chefsalm pushed 1 new commit to 06salam/fastmsi-redux: 02http://git.io/v8prN</i>	12:45 pm
	<i>NOTICE: 13omnibus-chef/06salam/fastmsi-redux 142a2a5d4 15Salim Alam: Update Gemfile.lock for omnibus 5.0.0</i>	12:45 pm
adamedx	lamont_oc do you need time ot consider use of api version?	12:45 pm
lamont_oc	what happens when a client that supports NEXT is first making a request to a server?	12:45 pm
kallistec	the problem for us right now is that NEW chef client can send a request for api version NEXT to an OLD chef server and the versioning stuff will just be ignored	12:45 pm
	where OLD == 11.x up to whenever we added versioning (12.1??)	12:46 pm

ssd7	kallistec: Right, I think the only way out of that is to do some initial call that just gets you the API version of the server	12:46 pm
	if you get a 404, you know it is old	12:46 pm
martinb3	^ this is also what I was thinking. some sort of endpoint just for negotiation	12:46 pm
	old versions wouldn't have it.	12:47 pm
adamedx	lamont_oc seems like there is more to consider here	12:47 pm
	<i>martinb3 only chimed in since someone said the same thing I was thinking.</i>	12:47 pm
lamont_oc	that mostly works because i think we upload cookbook files to sandboxes first and can see the server responses	12:47 pm
ssd7	Another thing we need to figure out is what casting to an older API version looks like	12:47 pm
adamedx	do you want to make a more definitive call on the mechanism in an rfc update?	12:47 pm
kallistec	that'll be unneeded once everything supports versioning though, and theny your're left with inefficient logic	12:48 pm
lamont_oc	but its pretty ugly to add an asterix to the POST method for uploading cookbook metadata that the client MUST hit another endpoint first to determine the server version	12:48 pm
ssd7	Well, right now that post would get a 400	12:48 pm
	so we could just retry it when we get a 400 on an old server	12:48 pm
	retry it at a lower api version that is	12:48 pm
kallistec	especially for stuff like knife, you really want to reduce round trips	12:49 pm
lamont_oc	if we rename all_files to files it'll be a perfectly valid, but semantically totally uncorrect request on an old server, it won't 400	12:49 pm
ssd7	Yeah, I think we should try to solve this generally if it doesn't add a ton of work to this feature.	12:49 pm
	This API versioning stuff has been a bit rocky	12:49 pm
nathenharv...	OK, we have 1 minute left and it feels like this one requires a bit more thought so I'll move it to next week's agenda.	12:49 pm
ssd7	but we really want to use it more in the future	12:49 pm

	so we should sort it	12:49 pm
ranjibd	Yup	12:49 pm
lamont_oc	so really i think we leave the RFC as-written	12:50 pm
kallistec	ssd7: I think we did fix it correctly, just we can't rely on api versioning until we can kill off 11.x	12:50 pm
ranjibd	It will also help prototyping in goiardi	12:50 pm
kallistec	at that point we'll be golden	12:50 pm
lamont_oc	i put a lot of verbiage in like "MUST serve cookbooks with both <styles>, unless it can determine that the client prefers one or the other"	12:50 pm
ssd7	lamont_oc: Well, even if we solve this, the issue of what to do with things that can't be cast to lower API versions. i.e uploaded with version NEXT but requested with version 0	12:50 pm
kallistec	ssd7: return the files you can	12:51 pm
lamont_oc	that last clause would be satisfied by API versioning, but i was deliberately vague about mechanism and implementation.	12:51 pm
kallistec	functionality would be the same	12:51 pm
ssd7	sure, just think that should be spelled out	12:51 pm
coderanger	(we're at time in theory)	12:51 pm
nathenharv...	OK, we're at time	12:51 pm
	***** MEETING ENDS *****	12:51 pm