SpinKube Project Meeting

Note: This dedicated SpinKube meeting has been paused.
Please bring SpinKube agenda topics to the Spin project meeting: https://docs.google.com/document/d/1EG392gb8Eg-1ZEPDy18pgFZvMMrdAEybpCSufFXoe00

Meetings are held bi-weekly at 8:00 am PT/11:00 am ET/5:00 pm CET.
Come listen in, ask a question, or share an idea!

Wednesday, Oct 1st, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Skipped due to lack of quorum; will propose agenda item asynchronously in Slack

Attendees

- Vaughn Dice

Agenda (please add [yourname] to items when adding them)

- [vdice] Proposal to merge this SpinKube meeting into the Spin Project meeting

Action Items

-

Wednesday, Sep 17th, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamIvHjhF.1
Meeting ID: 834 7305 6051
Passcode: 880947

Skipped due to lack of agenda, see you at the next one!

Wednesday, Sep 3rd, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamIvHjhF.1
Meeting ID: 834 7305 6051
Passcode: 880947

Skipped due to lack of agenda, see you at the next one!

Wednesday, Aug 20th, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamIvHjhF.1
Meeting ID: 834 7305 6051
Passcode: 880947

Skipped due to lack of agenda, see you at the next one!

Wednesday, Aug 6th, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamIvHjhF.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Vaughn Dice
- Joshua Aminu

Agenda (please add [yourname] to items when adding them)

- Joshua and I (Vaughn) chatted about SpinKube and some of its high-level moving parts, as Josh is new to the project

Action Items

- [Vaughn] Label some issues in the spin-operator queue as ‘good first issue’ so that Josh can take a look as he familiarizes with SpinKube

Wednesday, July 23rd, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Skipped due to lack of agenda, see you at the next one!

Wednesday, July 9th, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Vaughn Dice
- MacKenzie Adam
- Joshua Aminu
- Jasmine Xu
- Matt Fisher

Agenda (please add [yourname] to items when adding them)

- Whose Zoom account are we using anyways?
- Discuss https://github.com/spinframework/spin-operator/pull/409 and 0.6.1 patch release

Notes

- Kenzie to look into which Zoom account we're on – migrate to CNCF?
- Need to cut 0.6.1 to get a new chart with a correct spin-operator image reference (spinframework instead of spinkube ghcr.io org)
    - Fisher: agreed, let's do that
- Kenzie: local service chaining issue? (https://github.com/spinframework/spin-operator/issues/357)
- Kenzie: Or more generally, can we go through the issue queue to maybe look at a theme for the next (0.7.0) release
- Fisher: Ah, that's a good callout. I'd be happy to start on this - though note I do go out on vacation starting tomorrow for week
- Fisher: I can send a message in Slack with a potential list
- Vaughn: Do we want to create a milestone or use some sort of process?
- Fisher: I think for now we just keep it high-level, like corralling a few issues as candidates
- Kenzie: What organization did we do in the past?
- Fisher: Basically just 'vibe releasing'

Wednesday, Jun 25th, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamIvHjhF.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Kate Goldenring
- Adam Reese

Agenda (please add [yourname] to items when adding them)

- Discuss releases

Notes

- Discussed weather to release the spin operator now that SKIP 6 is implemented
- Adam and KAte both agreed that might as well since last release was march
- Also should release shim since the spin version bumped

Wednesday, Jun 11th, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamIvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Caleb Schoepp
- Rajat Jindal
- Kate Goldenring
- Vaughn Dice

Agenda (please add [yourname] to items when adding them)

- Discuss SKIP to support per app invocation limits https://github.com/spinframework/skips/pull/18 - Kate

Notes

- Discuss SKIP
    - Kate: We need to chat about this SKIP to get clarity in a sync context and this is one of the steps to move forward with it. Let's take 5 to read the skip.
    - Rajat: I see we're namespacing keys for what executor limits. Is that really necessary b/c we can already infer the executor?
    - Kate: Initial idea for namespace was that limits that were not namespaced are ones that are validated. It's not expected that namespaced ones are validated. In reality an executor doesn't need to follow namespacing. This is more of a recommendation than something we would enforce.
    - Rajat: I think we should either not have the ns or we should enforce the ns
    - Kate: Enforcement it would require knowing the ns of the executor a priori

- Rajat: It can check the keys it doesn't know about and make sure they are namespaced. foo.com/exectime and foo.dev/exectime would be considered different values
- Kate: Interesting example. How do people feel about some keys having special behaviour if things aren't ns'd
- Rajat: I'm not strongly opposed to namespacing
- Kate: It is up to executor to validate their namespaced keys. Good point that the spin operator could make sure that the same keys does not have two different namespaces
- Kate: Foo.dev/exectime and bar.dev/exectime shouldn't be possible
- Rajat: Expands on why the above is hard to differentiate in spin operator
- Kate: I don't think we see real world usage of that. Multiple
- Caleb: Value of executor in SpinApp is to make it easy to swap between executors, so it feels weird to have validation that makes it hard to do that
- Rajat: How do you feel like namespace should be the same executor
- Kate: Wouldn't that prevent Caleb's story b/c you would have to edit it anyways
- Rajat: ..
- Kate: I want spin operator to not have knowledge of executor invocation limit namespace fields.
- Rajat: Executor name is rajat. Make sure that is in the namespace
- Kate: What comes after is impl that updates SpinApp CRD and update Spin Operator. I want no validation to happen for unknown fields
- Rajat: I'm onboard with no enforcement
- Kate: Validation is easy to add later
- Kate: I liked the structure of the skip process
- Kate: Meeting adjourned

Wednesday, May 14th, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Kate Goldenring
- Vaughn Dice

Agenda (please add [yourname] to items when adding them)

- CNCF misc.
- Runtime-class-manager
- KubeCon CFPs close Tuesday, May 27 at 11:59pm Eastern Daylight Time (UTC-4) [Kate]

Notes

- CNCF misc.
    - Kate: Is this Project Meeting on the CNCF Calendar?
    - Doesn't look like it, issuing a CNCF Service Desk request (Spin project meeting not seen either; slightly trickier with the alternating time)
    - Kate: Would be good to have a YouTube channel
    - Vaughn: I believe we do; Kenzie, Michelle and Radu should have access. We're also requesting a bump to the paid/Pro tier for the Zoom account for automated project meeting uploads to YT
    - Kate: On to Spin; sent
    - Kate: Any other tasks for Jira/Service Desk?
    - Vaughn: Not sure… there's some domain transfer, LF footer…
    - Kate: What about License scanning? Snyk or FOSSA. Want to work on this?
    - Vaughn: Sure
    - Kate: Which one? Hmmm well there's this: https://fossa.com/customers/cncf/ (seems to be the preferred tool for CNCF)
    - Kate: Going to look at other repos who use FOSSA for a pattern to follow (https://github.com/dapr/dapr/actions/runs/14956063058/workflow?pr=8710)
    - Kate: So, looks like this is another Service Desk request
    - Kate: Oh look at LFX Security, that looks interesting. (After researching, deciding to punt… maybe a larger endeavor to set up.)
    - Kate: So, FOSSA for now?
    - Vaughn: I think so.
    - Kate: How about the LF footer?
    - Vaughn: We'd want them on the Spin and SpinKube docs websites
    - Kate: Let's hop off and pair on this (Vaughn: Ok!)
- RCM:
    - Kate: We've added K8s-distro-specific restarters
    - Kate: I'm working on parity with the containerd-shim-spin re: containerd/systemd options. There's some questions on how the runtime options should be configured in the Shim CRD. There was feedback that we might need/want it to be more specific to containerd to distinguish between other runtimes eg cri-o.
    - Kate: I'll update from a generic 'RuntimeOptions' to a more specific 'ContainerdRuntimeOptions' but the data sent in via the CRD will still be generic, eg to support other containerd options down the road.
    - Vaughn: Want to note that this is the last (anticipated) issue before we close out RCM's first milestone

(https://github.com/spinframework/runtime-class-manager/milestone/1) - then we can start looking at finally using RCM in place of Kwasm-Operator in SpinKube!
    - Kate: Whoop!

Wednesday, April 30th, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Vaughn Dice
- Matt Fisher
- Caleb Schoepp
- Kate Goldenring
- Rajat Jindal
- Michelle Dhanani

Agenda (please add [yourname] to items when adding them)

- Runtime-class-manager [vdice]
    - Closed/completed nearly all of the first milestone issues! (thx Cristoph for helping and reviewing)
    - Final issue: node-installer#140

Notes

- Vaughn: Welcome everyone! One agenda item on runclass manager. Follow Up to last week. After the meeting last week we knocked out a bunch of issues from the first milestone. We just have one left, https://github.com/spinframework/runtime-class-manager/issues/140. I did some research and have a proposed path forward. Describes what issue describes. I would propose we overhaul the logic to be more like the shim node installer script. The shim one has bespoke things for each distro, whereas the runtime class manager does the same thing for each distro which gives us issues. Let's gather feedback, and hopefully we can agree on an approach to fix this then land runtime class manager.
- Matt: High level overview of why we are swapping in rcm?
- Vaughn: To replace kwasm
- Vaughn: Any other agenda items?

- Matt: It's been a while since I've contributed here. Are there any areas where we're looking for more help?
- Vaughn: The spinkube docs come to mind. Some good stuff that is easy in the docs issue queue
- Michelle (in chat): Just an announcement: if anyone has spare cycles, could use some help knocking out the last few CNCF onboarding items
- Matt: Who would I ask questions to?
- Kate: I can help. People are confused why we don't scale pods to 0. We want to have a good workflow with KEDA to enable scaling to 0. We might have to fight with the operator to get this working
- Matt: I was thinking about that autoscaling section of the docs already. Perhaps that would be a good area to expand on. We can help people through this unintuitive stuff. Docs seems like a good place.
- Matt: Is anyone else working on that docs stuff? Is there a BDFL?
- Vaughn: Not that I know of? You could ask in the spinkube channel.
- Kate: It'd be great to at some point revamp the installation guides. Have a baseline guide then have variations and caveats for each distro.
- Matt: Available providers? Azure, k3s…
- Kate: Azure is the only place with a marketplace offering, but spin-operator is just a helm chart so it is consistent across all k8s distributions.
- Vaughn: Scanning the docs, I will say rancher desktop has more of the under the hood support. It handles spinkube stuff behind the scenes.
- Vaughn: Any other topics?
- Rajat (in chat): I am planning on getting the 'actions' repo sorted out this week
- Rajat: Removing deploy to Fermyon Cloud actions out of the action repo. As part of the move we should make sure we don't break existing users, as we do not know what GitHub behavior will be with this partial move. ~~[Second part notetaker missed]~~
- Vaughn: Meeting adjourned 😀

Wednesday, April 16, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Kate Goldenring
- Vaughn Dice

Agenda (please add [yourname] to items when adding them)

- Runtime class manager finishing first milestone

Notes

- Kate: Is now a good time to make a push of RCM?
- Vaughn: there are a couple PRs and one PR needed to finish milestone 1. I can put a couple cycles in on that
- Kate: I will hold off on making the kwasm chart configurable and we can instead add the systemd containerd driver configuration to RCM instead
- Vaughn: RCM has Shim CRDs to define which shim should be installed, so should be fairly trivial to add support for more shims. Here is the url to fetch the binary and the strategy

Wednesday, April 2, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

Cancelled - KubeCon week

Wednesday, March 19, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Michelle Dhanani
- Radu Matei
- Kate Goldenring
- Vaughn Dice
- Rajat Jindal
- James Sturtevant

Agenda (please add [yourname] to items when adding them)

- [Michelle] Triage CNCF Onboarding board
    - SpinKube docs migration
    - Governance relationship between Spin <-> SpinKube_
    - KubeCon Prep

- [Radu] Meetings meetings
- [Joe] Requesting a release of the spin operator
- [Joe] Update - have been working on adding integration tests to node installers

Notes

- Michelle: CNCF Onboarding triage
    - We've moved a lot of items to Done! Bunch still in progress.
    - Thanks to Kate and Brian for the homebrew tap
    - Have moved almost all of the repos; enabled DCO (and required for PRs)
    - Spin Docs: Ronan offered two options, we picked option B, we've merged. Ivan had feedback (non-blocking) that we can visit as follow-ups. Still envision updates for/prior to Kubecon.
    - JS SDK: Major version release being worked on; Karthik and Till have a plan for npm migration
        - Radu: Right, new package in a spinframework npm org, due date around Kubecon
    - Transfer trademark and logo assets?
        - Radu: I think this is done
    - SpinKube Docs: Need to move over to the spinframework org, get the GH Pages deployment updated, update CNAME
        - Kate: I recall Ronan mentioned he's done this many times and should be able to advise on/confirm approach
        - Radu: We're moving the repo but keeping the same actual site/docs, right?
        - Michelle: Yes
        - Radu: And `spinkube.dev` is also being moved over to CNCF?
        - Michelle: Yes, both `spinkube.dev` and `spinframework.dev`. Likely next week. Would love to get the docs at least moved over.
    - We have a YouTube channel now
        - TODOs: automate Spin and SpinKube community meetings up to channel, setup shared access
        - Radu: Smash that Like & Subscribe button: https://www.youtube.com/@spinframework
        - Michelle: CNCF created this so they own it; all set there
    - Transferring domains
        - Michelle: As mentioned, probably a next week task
    - CNCF Code-of-conduct
        - Michelle: We should have our canonical version of the code-of-conduct in the governance repo and then link to this from the other repos
        - Radu: I can volunteer for that
    - Maintainer list and emails for onboarding
        - Vaughn: PR up, email sent!

- Import relevant issues from fermyon/developer
    - Michelle: Quyum is working on this
- Migrate Discord to CNCF Slack
    - Michelle: Any opinions on urgency for this?
    - Kate: So is there a certain date where we stop replying? Or just tell Discord users to move?
    - Michelle: Goal is to have one stream of comms for projects (and should be on CNCF Slack)
    - Kate: I'd say we don't need to delete our Spin channel on Discord (Fermyon's Discord), but would be nice to get more activity in the official CNCF channel
    - Radu: For maintainers and contribs to the project(s), def agree should move that to CNCF Slack - and seems to have already happened. Similar to Kate, I don't see a reason to delete the Discord channel - rather, just encourage aforementioned dialogue to happen in CNCF Slack. We can send a reminder as we get closer to KubeCon.
    - Radu: I can send this reminder ^^
- Linux Foundation footer on website (still TODO)
- Governance
    - Michelle: We don't really define the relationship between Spin and SpinKube in governance yet; they each have their own repos/docs. What should the relationship be? Last time we talked about it, SpinKube wanted the freedom to operate independently. So assuming this, we just need to add definition. The other Q is how is SpinKube involved in the top-level steering committee? We'd want representatives spanning spinframework org projects. Think we need a more formal structure here.
    - Kate: +1. We can start to think about what the process will be (eg elections, etc). Would this be in the form of a SIP (Spin Improvement Proposal)?
    - Michelle: Any objections to documenting the current state? (No.)
    - Michelle: Can we move the governance document to the spinframework/governance repo? (Yes.)
    - Vaughn: Does CNCF offer a blueprint for steering committee setup/workings?
        - Michelle: When I was on the ToC, we didn't offer a process for governance/steering committee
- OpenSSF badge
    - Kate: I'm familiar with this one if you want to assign me to it. Will start with Spin and then we can see what other repos we'd like it on (SpinKube-related?)
- Book time w/ CNCF to understand project benefits/resources
    - Michelle: Post-Kubecon
- Spin-related GH actions in fermyon/actions

        - Rajat: I can look into which ones are spinframework-specific and could move to this org (and which are Fermyon-specific and would stay in fermyon/actions)
    - SDKs are housed in spinframework package manager orgs
        - Radu: npm is the only registry where we have an org namespace (currently Fermyon). For crates.io (Rust) and py.pi (Python), we can just add spinframework bot for push access.
        - Vaughn: Bootstrapped the spinframework bot, can help here
    - Shared secret store
        - Vaughn: Blocked on maintainer email registrations to Service Desk; then we can request access to the secret store via Service Desk
- Radu: Meetings, meetings, meetings!
    - This may be more Spin-related, but will introduce anyways
    - We've been having a bunch of language-specific discussions (JS/TS, Rust, Python, etc). I think there's enough context that we might want meetings with this spinframework CNCF group
    - Michelle: Could these be additional topics in the existing Spin meeting?
    - Radu: Just hesitant to take over an entire meeting when/if there are lots of language-related details to go over. If we do have the time on the agenda, that would be great.
    - Radu: Also, this is a good opportunity to see if we do want to form new meeting(s) or whether we are fine just lumping into pre-existing meetings as agenda topics.
    - Kate: By the way, the CNCF calendar is working group-focused, so the meetings wouldn't go there…
    - Michelle: How about a community repo?
    - Radu: Where we can discuss something like this and document meeting(s) schedule(s)? (Michelle: Yes.)
    - Kate: Update: there are some Incubating proj meetings on the CNCF calendar
    - James: Right, Joe is trying to get runwasi on the calendar ^^
    - Radu: I'm happy to create a community repo if people are in favor
- Kate: next topic is from Joe: requesting a new release of the Spin Operator
    - Kate: We'll want a shim release as well, prior to Kubecon
    - Kate: Any reason we wouldn't release the Spin Operator? (No.)
    - Kate: I can take that on, haven't yet done a release, but can pull people in.
    - Michelle: Just an FYI, Spin 3.2 is due on Mar 24th
    - Kate: Ok, the shim can release after that
- Kate: another from Joe: He's been working on integration tests for the shim node-installer for all the K8s flavors
    - Kate: Part of the motivation is the runwasi update to configure the cgroup driver (to fix/support pod metrics!)
- Michelle: SpinKube docs
    - Michelle: Any changes we need/want pre-KubeCon? Eg note about CNCF Sandbox status. Call to action in the spinkube Slack channel.

Wednesday, March 5, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamIvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Michelle Dhanani
- Kate Goldenring
- Sven Pfennig
- Vaughn Dice
- Joe Zhou
- James Sturtevant

Agenda (please add [yourname] to items when adding them)

- [Michelle] CNCF Onboarding update and next steps

Notes

- Kate: only one agenda item so far, Michelle's CNCF Onboarding update, but then we can discuss ad hoc topics
- Michelle: Yes, just updated the project board: https://github.com/orgs/spinframework/projects/2/views/1. We were a bit blocked on the Contribution Document signing. Next steps: #1 priority to unblock CI (need larger runners). Brian did a lot of work around updating automation (Homebrew tap, CI, etc). Might be some more work around docs (talking with Ronan).
- Kate: Will give a bump to review Joe's Workload Identity SKIP https://github.com/spinframework/skips/pull/16. (He's also posed a Q in spinkube-dev channel)
- Kate: Runtime-class-manager?
- Sven: I had some time to update references (eg domain name to spinkube.dev, spinkube -> spinframework)
- Kate: So we'd be ready to switch out kwasm-op w/ rcm?
- Sven: Not 100% sure. We need to check in with milestone issues, test on clusters, etc.
- Sven: the spinkube -> spinframework change was not strictly necessary since GH does auto-redirect.
- Sven: We are working with spinkube.dev as the CRD domain. The spinframework change was just for eg URLs

- Michelle: KubeCon is coming up. We might think about other changes we might want in docs in anticipation. We're already doing a docs refresh for SpinKube, so maybe that's a good opportunity for a fresh set of eyes.
- Kate: Yes, especially the document around 'How to optimize SpinKube'
- Michelle: Did we decide if the SpinKube docs website would be the same? Merge sites or keep separate?
- Kate: I believe we decided to keep them separate.
- Joe: Do we have any talks about Spin/SpinKube at Kubecon?
- Kate: No, the project(s) weren't yet accepted into the Sandbox at time of talk deadline.
- Michelle: David Justice may have one; Thorsten has a Realtime AI inference talk w/ Spin.
- https://events.linuxfoundation.org/kubecon-cloudnativecon-europe/program/schedule/
- Joe: I have a talk about containerd deep-dive. Mainly talking about WASI, but could be an opportunity to call out SpinKube.
- Kate: We could have a blog post about where to find Spin maintainers at Kubecon, etc.
- Joe: We don't have a booth for Spin, though right?
- Michelle: No
- Kate: Ok, we'll have some amount of presence at KubeCon EU; I think we'll want to put all the resources we can leverage into KubeCon NA.
- Kate: Michelle mentioned a standalone Spin blog (not from Fermyon's blog). I would maybe expect it to live on the new/separate Spin Docs website, eg spinframework.dev. (Even though we do have a blog on spinkube.dev) I'll double-check the wireframes from Ronan to be sure there is a Blog there.
- James: Do we get spots on the CNCF webinars too?
- Kate: Yes!
- Joe: SKIP: Main goal was to add Workload Identity to the SpinApp CRDs. During my implementation, I realized the Spin Operator should be cloud-provider agnostic. Adding Azure-specific metadata should be handled by a different agent. So, I slimmed the SKIP to just propose adding one generic field (ServiceAccountName) to the SpinApp CRD. Should I just delete the SKIP and PR that change?
- Kate: I think the SKIP is still helpful, to provide context and vision, despite the change(s) being very small.
- Joe: Ok, that sounds good to me. I will also go ahead and implement that in the Spin Operator.
- Kate: In the SKIP or in the PR, can you add a section on how cloud-provider specific metadata is added?
- Joe: I am working on a 'spin azure' plugin to streamline eg creating an AKS cluster with everything in place for supporting Workload Identity.
- Kate: That sounds awesome.
- Joe: Also want another item to 'spin kube scaffold' to allow adding ServiceAccountName
- Kate: Agreed, anytime anything is added to the SpinApp CRD, an update to the plugin should follow.
- Michelle: (Different topic) There was a fix around enabling HPA (cgroups path patch)… timeline for that?

- Kate: I believe runwasi was released with that fix… Joe?
- Joe: Yes, patch is in but we haven't released yet. I can do it this week. There are some other changes I want to get in… but blocked on upstream rust-extensions fix. (in anticipation for a 1.0).
- Kate: How about 0.10 with the cgroups fix and then 1.0 with the other changes?
- Kate: We'll have to update the node-installer as well, right? Don't we need to update restart logic as well?
- Joe: systemctl restart works for k3s… but maybe?
- Kate: Ok, we'll just have to caveat that nothing is going to change until we update the node-installer.
- Joe: Kate you mentioned some autoscaling functionality will be unblocked?
- Kate: Right, we were unable to get accurate metrics from pods. So, HPA would fail (pod-level metric was never changing). With this change, we can use HPA and autoscalers in general. (With k3d we could do this in demos bc it didn't use systemd.)
- Kate: So it sounds like we'll have a release from runwasi this week, then we can line up a shim release, then hackathon with node-installer to check all K8s distros. So, optimistically end of this week, more realistically mid-next-week
- James: Going in and setting up/configuring systemd would be strange.
- Kate: We should be able to detect which cgroups driver is being used, right? And go from there?
- James: Yes. In image builder, we default everything to systemd. May just be a documentation issue? Logging statement to say 'You should turn on systemd, here's why'?
- Kate: Ok, so we can detect. Filesystem cgroups is the default. I'll look into this; will create an issue.
- Kate: Did we want to do some triage?
- Michelle: No, just linked one issue 🙂

Wednesday, Feb19th, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Joe Zhou
- Kate Goldenring
- Vaughn Dice

Agenda (please add [yourname] to items when adding them)

- Joe: merged SpinKube repos into spinframework
- Joe: Azure Workload Identity SKIP
- Joe: pod metrics issue in Runwasi
- Joe: customer feedback on SpinKube on scale-to-zero

Notes

- Kate: Welcome everyone!
- Joe: We've moved all of the SpinKube repos to spinframework. We have different maintainer lists (spin-operator, runtime-class-manager, containerd-shim), which I've copied over to spinframework. As far as I know the only item left is documentation.
- Kate: spinkube.dev is moved over already, right?
- Joe: Doesn't look like it.
- Kate: I don't think the spinkube docs are too complicated to move; however the Spin docs are a little more involved; Ronan is working on this.
- Joe: Do we still call SpinKube a thing to the outside world?
- Kate: Yes, it still is. B/c we have infra around SpinKube as a project, it will probably remain a thing. I don't feel a need to change the namespace.
- Joe: Another alternative: spinframework.
- Kate: Ok so everything Spin under this?
- Joe: Idea here is to unify the brand. After all it is just one project.
- Kate: That's a good Q: when we talk about Spin should we be using 'Spin Framework'. I thought we chose spinframework originally just per GH org availability. This could be a good thing to bring up in the broader Spin meeting.
- Joe: By the way, when is the Spin community meeting?
- Kate: Every week, though we alternate between EU-friendly and US-friendly. (Next week is EU-friendly.) If you look at the description of the #spin CNCF channel, you'll find dates/times/links. (We have the same for the #spinkube channel.)

- Joe: Azure Workload Identity SKIP (https://github.com/spinframework/skips/pull/16)
- Joe: Background: back in July, David Justice added SKIP 004 around Cloud Identity and Access (https://github.com/spinframework/skips/pull/9), covering all applicable clouds (Azure, Google, etc)
- Joe: In my proposal, I've scoped to Azure. The operator will create a service account w/ annotations, a pod with label(s). After talking with David, we then discussed just the pod label(s) and the operator should not create the SA (we want the operator to have very restricted access perms). So, we'd add a workload identity field to the SpinApp to contain metadata and then the operator will apply the corresponding label(s) to the Spin App pods, again specific to Azure.
- Kate: So are we moving away from David's draft SKIP? Should we close it?

- Joe: Yes, I think we should close it for now. Happy to add a 'Future Work' to mine referencing this.
- Joe: I have an idea for another SKIP around User-managed Identity and Federated credentials in the form of a spin (azure) plugin. Basically a wrapper of Terraform and/or Azure CLI.
- Joe: (Shares screen). I note in the SKIP that there are some prereqs that the user needs to create. Spin Operator will not create or manage any of these. So, the plugin might handle some or all of these. But out of scope of PR 16.
- Kate: Could some of that go into the az cli itself eg a cluster extension?
- Joe: Not sure. If you are already doing app building and deployment in the spin CLI, why not use spin.
- Kate: Right. Historically, spin plugins have been tied to using spin directly. Brainstorming, wonder if this would be a 'spin cluster' plugin eg 'spin cluster az', 'spin cluster k3s', etc. Not blocking, but just thinking ambitiously.
- Joe: Although I said this only Azure, when I designed the fields, I am considering the potential for adding other cloud providers. Eg first was 'providers' and under that is 'azure' and under that 'client id'... then changed to generic provider metadata key/values.
- Kate: So Spin Operator doesn't need to know about a specific provider?
- Joe: Ultimately, you want to create a pod with cloud-provider-specific label.
- Kate: Right, just thinking about trade-off between generic vs specific. (strongly-typed vs dynamic). If we have a map of generic data, we may also want a webhook to check. More troubleshooting.
- Joe: If we want SpinKube to be totally generic, maybe there would be an additional operator, in this case an azure operator, which would do the same.
- Kate: No, I think it is fine to be in Spin Operator, just want to make sure it is inviting to other cloud providers.
- Joe: Please take some time to review this SKIP. Fairly confident in the scope; will work on a PoC for this and attach it to the SKIP.

- Kate: Ok, it is the Joe Show today! Another one from Joe: Pod metrics in runwasi (https://github.com/containerd/runwasi/issues/821)
- Joe: Pod metrics issue was mainly investigated by you, Kate. Another contributor helped with investigation. Found the root cause was our use of a diff cgroup driver. The solution: we should parse the cgroup driver from the options field of the cri data. However, parsing isn't trivial. There is a workaround: string/regex matching. I believe a PR is already up with this. After this is merged, I expect to cut a release for the containerd-shim runwasi crate by the end of Feb. Then we can release for SpinKube. (After confirming it works locally.)
- Kate: I saw Jorge mention the grpc approach. So perhaps in the future we can go beyond string-matching. That allows us to parse more options. Implication of this is we can now use Horizontal Pod Autoscaling!
- Joe: By the way, the next runwasi release will probably be the last pre-1.0. Targeting 1.0 at/by end of March.

- Kate: runwasi stabilizing puts the shim in a good spot, which in turn will put SpinKube in a good (stable) spot.
- Joe: Right, the spin shim should be ready to be 1.0 afterwards.
- Kate: Yes but there are some triggers incorporated into the shim that may need more vetting.
- Kate: By the way, we should talk about dynamic loading of triggers. Essentially dynamically loading a host component, which is a big ask.

- Kate: Next topic: Customer feedback re: scaling-to-zero
- Joe: We talked about encouraging users to use KEDA last time. This is becoming a main pain point with customers. They expect SpinKube to somehow scale to zero.
- Kate: Last time I talked about writing a doc around optimizing this and help users better understand how Spin works. Eg if you create a Spin App with a lot of endpoints, you get a good level of scalability.
- Joe: Agree documentation would help.
- Joe: We've been talking with a financial services company where this came up. Scaling down to zero as well as persistent DB connections.
- Kate: For the latter, I recall that issue – but it should be resolved. (Looks) Ok, it was merged into Spin on Jan 13th, so it may not be in SpinKube.
- Joe: 3rd pain point: Want to scale down workloads combined with improving memory and CPU usage.
- Vaughn: related to Horizontal Pod Autoscaling mentioned in runwasi pod metrics issue?
- Joe: Not sure.
- Joe: So action items could be: docs and performance report

- Vaughn: Will try to reignite momentum for runtime-class-manager

- Joe: Azure Marketplace: There is a doc about how to use SpinKube on Azure. We should add Marketplace offering to this. Is Kenzie working on that? I will ping.
- Joe: With Workload Identity, we could have a checkbox for the Azure offering to automatically provision these resources. (And then, the aforementioned cli plugin for doing so outside of the Marketplace.)

-

Wednesday, Jan 22nd, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamIvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Michelle Dhanani
- Adam Reese
- Joe Zhou
- Sven Pfennig
- Kate Goldenring
- Vaughn Dice
- James Sturtevant

Agenda (please add [yourname] to items when adding them)

- Joe: creation of SpinKube-maintainers channel on CNCF Slack
- Joe: discuss customer feedback
- Michelle/Kate: CNCF onboarding
    - Maintainers should review “Review and understand other documents” sections
    - Spin and SpinKube governance
    - Migrating to a new GitHub org
        - new GitHub org membership invites for maintainers
- Joe: Docs review https://github.com/spinkube/documentation/pull/274
-

Notes

- Joe: Creation of SpinKube-maintainers (or -dev) channel on CNCF Slack
    - The current channel is more user-facing; I’m more comfortable posting dev/maintainer-related items in a channel for these topics
    - Michelle: We had two similar channels for Helm
    - Kate: containerd-dev is public
    - Kate: Sub-question: -maintainers or -dev. Maybe prefer -dev? -maintainers might signify private. (Actually, we are allowed to create private channels, so we could do this)
    - Joe: So what do we think about it? Any need for a private channel?

    - Michelle: Public seems right for now. Also, we get a maintainers mailing list through CNCF
- Joe: Customer feedback
    - Recently did some dashboard work to look into SpinKube on AKS (either/both Marketplace or directly via Helm chart)
    - Ralph reached out to customers:
        - “We got SpinKube working. But didn’t meet our expectations. Apps still run in Pods which still take up space…”
    - My take from the feedback: We should always scale to zero
    - Kate: Spin is unique in that we scale per request and this has been hard to communicate. The fact that a pod exists is our baseline, so yes it does take that minimal amount of memory. To scale pods down automatically, we’d need to integrate Keda, right? So now we have more pods.
    - Michelle: Perf/optimization could use some love in terms of how we document and communicate it. Should have a concerted effort here.
    - Kate: We should also document: The more components in a Spin app, the better return on efficiency/scale (all these would be in the one pod).
    - Joe: Back to integrating Keda, can you elaborate, Kate?
    - Kate: They would need to add pod autoscaler and keda-operator, so another dependency to run SpinKube.
    - Joe: Okay so this isn’t something we’d want spin-operator to implement?
    - Kate: Right, since these tools already exist
    - Michelle/Kate: Thank you for sharing this feedback, Joe.
    - Michelle: How do we address the documentation/communication needs?
    - Kate: This probably fits best in SpinKube docs; does that seem reasonable? (Yes.)
- Michelle: CNCF onboarding
    - There is an issue for this. [PROJECT ONBOARDING] Spin & SpinKube · Issue #330 · cncf/sandbox · GitHub
    - SpinKube has been accepted as part of the Spin project.
    - As one of the people who worked on Spin governance, we left it a bit open-ended as we thought we’d eventually have a steering committee…
        - Options:
            - SpinKube is a sub-project, they govern themselves. Have a # of reps in the top-level Spin governance body. A bit more loosely-defined
            - Concerted effort around restructuring the top-level Spin governance, eg creation of steering committee
    - Any thoughts?
    - Kate: I think defining a sub-project for SpinKube might be tricky, since we have a bunch of others (triggers, etc). A pattern in a diff. Bytecode Alliance example -- project is use of representatives and designating a sub-project as ‘graduated’.

    - Michelle: I don't know if we should adopt the term of 'graduated', but I agree with the representative approach. Wonder if we should have a joint decision with Spin maintainers. Any other thoughts here?
    - Michelle: Kate and Joe and I will form a sub-team to go through the onboarding issue. Let us know if you'd like to help
        - The issue has a list of docs that Spin(Kube) maintainers should review and read
        - Moving to a new GH org. One project on the GH
        - For spin we started thinking about naming already and got the GitHub org spinframework
        - If you are a maintainer, should have gotten an email invite
    - Adam: So if SpinKube is going to have to move over, are all the other repos in the SK org going to be moved over and are all of those subprojects.
    - Michelle: Idk if SK is the subproject and all of those projects are subprojects of spinkube. Spin-operator will be github.com/spinframework/spin-operator.
    - Joe: Where in the CNCF document says that one project = one org?
    - Michelle: Someone reached out to Bob Killan and he gave us that feedback. Maybe ask in the issue
    - **Recap for maintainers: review issue docs, check that have gotten invite to GH org, let us know if you have a better idea for org name, we will get more clarification on whether we have to move SK repos to that org**
    - D
    - Kate: When we talk about art and logos, is it just Spin that will be on the landscape?
    - Michelle: I think we'll be able to add SpinKube (a current sub-proj) to the landscape. Is kubebuilder part of the landscape?
    - Caleb: Will SpinKube tie its release cadence more tightly with Spin?
        - I don't think we need to answer this right away
        - Kate: I think we'll naturally be more aligned when we have both channels on the CNCF Slack. Agree there is room for an improvement on cutting a SpinKube release after Spin
    - Vaughn: Sven and I wanted to check on updating the domain name. Is spinkube.dev. I am assuming that this is not changing with the migration with the subproject
    - Michelle: We do need to start migrating domains to CNCF; I think this includes SpinKube. If people have knowledge around eg spinkube.dev (analytics, etc) we could use their help.
- Joe: I have a PR to add community call details open, please review: https://github.com/spinkube/documentation/pull/274
-

Action Items

- Kate: write up SpinKube docs on performance to address user comment:
    - "We did try Spinkube and got it working with our own small code example (webhook callback). We didn't see the benefits of it though, because it didn't work how we expected it would. We were

expecting the Spinkube Operator to dynamically create and manage things but it seemed to act like regular pods and so take up space, even when not being called."

- Michelle, Joe, and Kate: work on CNCF checklist
-

Wednesday, Jan 8th, 2025 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- David Justice
- Radu Matei
- Kate Goldenring
- Thorsten Hans
- Michelle Dhanani
- Vaughn Dice
- Sven Pfennig
- Caleb Schoepp

Agenda (please add [yourname] to items when adding them)

- CNCF Sandbox submission updates []
- [Michelle] docs update

Notes

- Kate: kick us off – CNCF Sandobx
- Radu: tl;dr - We had a wasm wg meeting yesterday and I think where we're at is the leads and tag runtime folks are working on spin and spinkube assessment which is happening together. Some misunderstanding recently in the wasm wg. Basically, they are currently working on the review. Want to bring up a conversation TOC has had and we've had recently based off Bob Killman's comments around making Spin and SpinKube a single project.

- Kate: would that play out as the CNCF accepting Spin and saying SpinKube is a part of ti?
- Michelle: from my understanding, yes; SpinKube a subproject of Spin
- David:that makes sense for some things; but kinda missing the point of the runtime class manager, which are not really tied to Spin. Also part of the misunderstanding of the original review; not tied entirely to Spin; having other runtime executors might help, but idk. The project is nascent, and there;s only so much time. I'll be adding to the feedback I have on the analysis when presenting to TAG runtime.
- Kate: Sven – for RCM and our milestones, do we have milestones to support other shims? Something we might reference. Maybe SpinKube isn't tied to Spin entirely. Something to consider.
- Sven: at least in the readme; MIlestone 1 to support Spin, milestone 2 to support all things KWasm did.
- Michelle: sucha hard conversation to have; yeah, the SpinKube project was so nascent, and if we're to support other shims and runtimes, that's something we haven't really had in suych depth; no one is opposed, but there's only so much time to prioritize the end users. I worry about locking the SPinKube project in a specific roadmap if we put it under the Spin project; but the name m,akes it somewhat confusing. But if we had to rename the project, … then there's this whole other note that Spin is a runtime, but since it's more modular, you could have runtime configurations that get put into the mix. Maybe it would be helpful to talk about what we want the scope of what we want the prouejct to be; or maybe it's young and don't want to dot hat yet.
- Kate: we may need to rename "Kube" anyway; not clear whether the project gets grandfathered; regardless, a rename may better fit our roadmap. Going back to Michelle question – is this the time we should be talking about the roadmap?
- Michelle: in the spirit of moving the discussion – maybe it's easier for the projects to be together for now under a single project. If we want to expand, then we breakout the projects later and submit; feels like so much admin work and it's hindering a bit the momentum. I really like the SK org, governance, structure, feels good; for the eng momentum to be hindered based on admin requirements, feels off.
- Kate: is there a scenario where we're happier SK being independent for longer rather than together? Would we rather wait?
- Radu: This boils down to why we wanted both as CNCF projects. It benefits the community and helps build momentum around Wasm. To me, the reason to submit both is very much in scope. Makes sure no single company can steam roll. Feels very relevant for both projects -- the CNCF is this neutral place. I feel like SpinKube belongs in the CNCF. I'll state my opinion: Bob mentioned the admin work being double if there are two projects. I still strongly believe that both belong, conceptually and spiritually. Sven and I had a similar conversation about kwasm in the past -- it belongs in the cncf whether a part of one project or another. CNCF sandbox is the place for projects to grow.
- Sven: It is Sandbox. It is not very different if there is one project or another. It's a coin flip.
- Kate: even when Spin and SK became the same project, we could probably keep governance of the projects differently as well – Spin as the org would

- David: containerd does a good job of that; runwasi is a containerd subproject, and have little project overhead admin tasks, but reap a lot of the benefits; I can see a similar setup
- Radu: The spirit of the SK governance from day one has been to have clear governance of projects and their release schedules. We already enable individual projects to have their own freedom
- Michelle: We have spin core and its core maintainers. Then we have SDKs, plugins, etc, and they have their own maintainer set. The core maintainers of Spin core serve as the top level governance body. Eventually we want a steering committee like K8s, so that any non-code level decisions can fall to the steering committee. It is consensus based -- checking for no objections. We have a path for conflict resolution . we want it to be very transparent about the processes and the path to become a maintainer.
- Caleb: Michelle raised a good point on not losing eng momentum – if we take it as a give that we want SK in the CNCF, it feels like a lot less work to piffyback. State that I think that;'s the best way IMO to keep the momentum.
- Joe: Question, if we do merge the projects, will this happen prior to or after the CNCF TOC votes?
- Michelle: this is a really good q; Kate mentioned that it's normal to move the projects and orgs AFTER being accepted. What does it mean to merge projects? Open question, don't expect a full answer. Docs, governance, logistics? I'd love for the SK org to remain its own; already confusing with lots of things, but personal, VERY minor nit.
- Kate: we can ask for that.
- Michelle: another thing I thought of – it'd be nice for SK to show up as a logo in the tools roadmap; "how do I run Wasm on Kubernetes" – discover easily. If housed under Spin…? Discoverability?
- David: same org says it's only for Spin; true today, but maybe not in the future. Or the intent. Is that reasonable? At least that's what I thought. If not, then the same org…
- Michelle: can we talk about – we're passionate about not locking SK into what it is today, making sure it has a future that's open to supporting multiple shims/runtimes? If we do a rename – from the TOC video, the name was the confusing bit; so making it more clear the intention is more broadly scoped, that's an option that the TOC can digest and might help us.
- Radu: if the recommendation is that they should be considered together, are we okay with that? I'd be sad to lose another cycle of TOC reviews. Are we in consensus that we are okay with them being submitted together?
- [Nods … no one vetos]
- Kate: Michelle – your point on discoverability – I've seen in other CNCF projects that it's kinda normal for the K8s part to be separate; example – k6s – people will get to the operator eventually. I think we can hit that; sure there are more examples.
- Michelle: agreed; because I helped the governance for SPin, we should talk about what the governance would look like; I can help draft how potentially the governance should look if subpeojcts and what the wording should be. Essentially, SK project governance to remain the same, but the top level steering committe to ensure it represents all the views. I can take that as an action item.

- Radu: I would invite anyone with thoughts about governance to speak up about it. Reading through Bob's comments, I think we are good on governance. I think for incubation it needs to be more buttoned up.

- Docs update
    - Michelle: Thorsten sent me a video of someone walking through our docs, and they looked confused; took a stab at redoing our front page; will submit a PR later today; summarizing some of the individual projects; you'll get pinged. If others have ideas, we should sync.

Wednesday, December 11th, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Adam Reese
- Kate Goldenring
- Caleb Schoepp
- Joe Zhou
- Sven Pfennig

Agenda (please add [yourname] to items when adding them)

- Reviews needed on Vaugh's runtime-class-manager PRs: https://github.com/spinkube/runtime-class-manager/milestone/1
- Skip next meeting because of holidays
-

Notes

- Open discussion
    - Shim PR to bump to containerd-shim 0.8.0 is blocked by dependency issue in Spin. 0.7.0 bump is ready for review
    - Blog about Azure marketplace offering is underway

Wednesday, November 27th, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamIvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Vaughn Dice
- Kate Goldenring
- Michelle Dhanani
- Sven Pfennig
- Radu Matei

Agenda (please add [yourname] to items when adding them)

- Runtime-class-manager updates [Vaughn]
- TAG Runtime meeting recap [Kate]
- SKIPs to merge/review? [Radu]
- Component dependencies support in the shim [Kate]

Notes

- Runtime class manager updates
    - Vaughn: while KubeCon, have been working on Runtime class manager trying to get to milestone 1, so we can swap RCM for KWasm in SpinKube. We now need reviews for the PRs – last one around CD/automation to publish the chart; we also have more test automation, more flavors to the k8s array, covering minikube, microk8s, k3d; more a call to action for the PR queue – https://github.com/spinkube/runtime-class-manager/pulls
    - Vaughn: helm chart CD and test automation are the outstanding milestone 1; if this is it, we should be good to swap KWasm operator.
    - Kate: is RCM using the updated domain for SpinKube?
    - Vaughn: created an issue for it, that's another good area for people to chime in https://github.com/spinkube/runtime-class-manager/issues/255 ; it would be preferable to get this in .
    - Sven: yeah, it just needs to be done
    - Vaughn: I can pick this up after the holiday.

    - Kate: for reviewing, anything else you want people to test out , in clusters? Side by side with the operator?
    - Vaughn: pending chart automation, you can install directly from the repo; PRs might be good since there could be follow ups from any of the PRs; waiting until the next SpinKube meeting for a call to action to test all the things out.
- TAG runtime meeting recap
    - Kate: last week Michelle and I presented to the runtime TAG a review of Spin+SpinKube; we are being considered for the CNCF sandbox att hte nxet TOC meeting [Michelle: coming up early Jan]; that presentation went well, we didn't have time for Q&A, but after the chair reached out to us to open the conversation and to support us going forward; the runtime TAG is excited and we need to get to the TOC hurdle for acceptance. One other consideration, for the next TOC review it's just Spin being reviewed, since they'll most likely consider a single project ,instead of two; we might later on submit as a separate project ,or subproject ,not sure how that'll unfold.
    - Kate: if there are any questions, you can put it in their channel, or Michelle or I can ask;
- https://github.com/spinkube/skips/pulls?q=sort%3Aupdated-desc+is%3Apr+is%3Aopen
    - Radu: There is one SKIP on how we want to merge SKIPS, please review that one. It is on the SKIP approval process. It has 4 approves, but it would be nice to have more. Please review SKIP #5 https://github.com/spinkube/skips/pull/13
- Component deps in the shim
    - Kate: we found out we don't support them; Brian has been working on it and we have an issue; it's related to precompilation, we expect each component to be precompiled as a component, whereas now multiple layers can be composed and then precompiled; we need to update the precompilation implementation in runwasi, then update the shim. It might be best in the SpinKube docs to say it's not supported.
    - Radu: can we turn off precompilation?
    - Kate: to turn off precompilation we need to pass that to runwasi;
    - Radu; executor setting?
    - Kate: so if multiple executors?
    - Kate: we can document workaround for container based.
    - Vaughn: following up on executor configuration – that's containerd config for your shim binary? Brainstorming that on RCM, that's a component that installs shims; would we see it in the future a config option there?
    - Kate: ideal place to do it IMO.
    - Kate: is there a world where we want to configure this in spin.toml? Because we can do it that way too; we get all the OCI layers, we can process the config layer. This might be hacky..
    - Radu: you can even read the lockfile right now and skip precompilation if any dependencies.
    - Kate: that would not be a bad idea.
    - Vaughn: temporary, while we add support for precompilation?

    - Radu: and if
    - Kate: how do we configure precompilation more generally, not just now?
    - Michelle: is that the only issue component deps has? I remember Brian saying something about garbage collection for OCI?
    - Kate: IIRC there's something about removing unused layers? That would be part of the updated spec James was talking about; that might speak about the complexity of the feature..
-

(Wednesday, November 13th meeting skipped per Wasm/KubeCon)

Wednesday, October 30th, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Radu Matei
- Caleb Schoepp
- Kate Goldenring
- Jiaxiao Zhou
- David Justice
- Rajat Jindal

Agenda (please add [yourname] to items when adding them)

- Containerd shim CHANGELOG workflow - Kate
- "SpinKube" release discussion revisit - Kate
- Sidecar container SKIP - Caleb
- Documentation for breaking changes in Spin Operator v0.4.0 - Caleb
- Cutting Spin Operator v0.4.0 - Caleb
- What kind of an approval process do we want for SKIPs - Caleb (oops I added another one)

Notes

- Containerd shim CHANGELOG workflow - Kate

    - Kate: Joe – thoughts on the changelog changes and best practices? Recently you commented to add things to the changelog, and wanted your thoughts.
    - Joe: there is a “keepchangelog.com” that suggests we add a new line to the changelog that describes the changes. Convention, at the top of the file is a comment suggesting the style. The intention is that we want a human readable list of changes, so at release time, parse the file, add them to the notes.
    - Kate: curious if this predates the tooling around GH that gives you this ability based on commits and PRs. Wondering if the tradeoff between every contributor adding this there compared to one person releasing spending time during the release.
    - Joe: good point. In my practice, people are less rigorous about PR typos and descriptions, automatic GH release notes are not accurate.
    - David: I think it comes down to how good the commit messages / PR descriptions.
    - Kate: only other concern is increasing churn for PRs, with people not remembering. Open to trying it out and see if we’re finding it’s causing churn or that a release is not made easier. Good to having the context, sounds good.
    - Joe; apprepaciate it, let’s try it out and get some feedback. We can always revert.
- “SpinKube” release discussion revisit - Kate
    - Kate: what does it mean to release revisited – doc matrix – pushed a draft PR and want us to put a user’s hat to see if that’d be sufficient. For marketplace offerings, they are bundled. Wonder if having releases/bundles would be helpful for marketplaces to know when to update.
    - Caleb: does Spin operator act as the entrypoint ? Could we use that as the marquee ?
    - Kate: are you saying everyone should be versioned as the operator?
    - Caleb: curious if there are other places where SpinKube is used where the operator is not the main interface.
    - Kate: several releases of SK happening without operator changes. Say we add functionality to the shim with no operator changes. A bundle needs to be mostly “wording” – people can see the exact versions of what’s inside.
    - Joe: IIRC, part of the concern was re: KubeCon, and if we wanted to release something for that. Let’s do a bundle release for KC; other times,I’d advocate for individual releases. The arch for the operator, shim, and RCM are decoupled. IF there are new features in one, not necessary to have in other. For events like KC, let’s doa bundle.
    - Radu: Still very much in favor of individual releases of projects and each following semver. Generalizing a bit on Joe’s point, most likely I will not care about the individual versions of each. I will care about a version that is installable that works. The bundle is that user installable stable release of that. We should have that for KubeCon and regularly. We can put bundles out which signals for users to upgrade. There will be a class of users that will update whenever features land. We should have a
    - Kate: action item writing a SKIP defining what a bundle is? Docs?

    - Radu: excellent point, a SKIP sounds like the great next step/
    - Kate: SKIP, or issue? I have lots of questions before. I might start with an issue to see where people are aiming, then we can move to a SKIP.
- Sidecar container SKIP - Caleb
    - Caleb: wanted space for this. Approval for SKIPS? For the domain change we decided to ask for approval ? I'm proposing we should merge this given the approvals and wanted to chat here.
    - Joe: SKIP is to add sidecar container support to the operator. They are already supported in the shim. But there is not way to specify that in the Spin app CRD, since it assumes it takes the form of an app + executor. This SKIP adds container syntax to the CRD so you can specify containers. In k8s, there is a specific syntax for sidecars – this extends the SKIP in supporting init containers in the CRD as well.
    - https://github.com/spinkube/skips/pull/11
    - Caleb: bringing it up as a chance for anyone who thinks we shouldn't move forward with this.
    - James: can you expand a bit on how sidecar relates to init containers?
    - Joe: there's docs about sidecar containers and the feature state in k8s 1.29 as beta. Restart policy is what defines whether
    - https://github.com/spinkube/skips/pull/11
    - James: didn't know about the sidecar; thought it was just something that was a regular pod with several containers. TIL.
    - Joe: that's what people practiced for a long time with the pattern. This is to me a new thing in k8s. We could talk about whether we
    - Radu: What i am hearing is that initContainer as a separate thing is obsolete due to the addition of side car containers. SKIP takes advantage of this new K8s concept. Is that a good summary?
    - Joe: Yes
    - James: what is the advantage of using the SPin app vs. a standard deployment? If we start to have to keep up with all the features, it feels like we're playing catchup. What's the clear value add we're getting here?
    - Caleb: native runtime config, swapping between executors; those we get natively out of the CRD compared to directly using deployments.
    - Radu: I would add the ability to have SpinApp CRDs be portable across environments and abstract away lots of k8s concepts.
    - Kate: We automatically support things like OTel or selective components in the SpinApp CRD
    - Radu: always a good thing to keep ourselves honest on whether we're adding value.
    - Kate: scope creep is always a good thing to keep in mind. Can anyone else anticipate future requests that may come out of this?
    - James: don't have one. This sidecar pattern, are we able to match the standard pattern that's been around? Or does this pattern meet all of the concerns so we

don't have to implement all. Can we put a wasm and another container next to each other, ? Or we'll never do the standard ¾ containers inside of a pod?
    - Joe: valid concern. We'd have to open the Spin app CRD to any containers?
    - James: not saying we should.
    - Kate: when we originally talked about opening for just containers, I got scared. Init containers sound great. IT defines why to use and how. If we have requests for people who wanted this, we could have a flag that takes advantage.
    - Radu: Having this in K8s vX doesn't seem like a huge limitation. Containerd is more of a limitation and this proposal seems to be the way to handle initContainers and sidecar containers in a unified way
    - Joe; I think this is intended to solve the long standing issues with init and sidecars.
    - James: the feature is on since 1.29.
    - Joe: both Dapr and Envoy are still using the old pattern.
    - Radu: always a good test, but are they doing that because it's not doing something, or they haven't gotten around to updating?
    - James: if it doesn't meet the need, we can adjust since it's our own CRD. Good point.
    - Caleb: 1] is this enough to go forward with this?
    - Joe: yes. I can go back to the SKIP and add more docs. I'd like to move that forward. I can take on the impl since I have a prototype. Mostly come after KubeCon.
    - Caleb: 2] we haven't ratified what the SKIP approval process is. We'll have this issue of fumbling around "are we happy to approve this". We need to write a SKIP about the approval process. Does that seem reasonable? I can take the action item
    - Kate: for the init containers, assuming approved, merging.Is there a world where we'd get this working before KubeCon? Is it worth discussing?
    - Joe: I have a prototype I can PR to the operator as soon as this week. Most it'll get in after KC.
    - David: since it seems somewhat controversial, wouldn't want to push it as quickly as possible. We don't want to rush it. I'd vote not trying to get it in.
    - Michelle: always voting the same as David. The only caveat is we're still not alpha status with the CRD, so that's the only reason it'd be nice to get it in if possible. No rush. At KC we get all this exposure, new feature..f
- Documentation for breaking changes in Spin Operator v0.4.0 - Caleb
    - Caleb: Adam and I worked on docs for the domain change breaking changes, PR up, wanted to share.
    - Caleb: heads up, planning to cut operator 0.4 ahead of KC. Last warning if you want to get something in.
- What kind of an approval process do we want for SKIPs - Caleb (oops I added another one)
    - Caleb: 2 or 3 maintainers have to approve unless it touches all parts, in which case all maintainers?

    - Kate: we have that at the top of the SKIP. Might be fair to say that it requires a maintainer from each project it touches.
    - Michelle: and maybe a minimum duration of being open, like 1 week?
    - James: lazy consensus deadline, so it doesn't sit there idle.
- Michelle: overcommunicating – we're planning on cutting an RC of the Spin project eow or early next week, then several releases of SDKs, want to make sure at a minimum update Rust ,JS, maybe Go. Then a list of plugins we want to make sure we cut releases of. Also use EOW to do that. Schedule is on the Spin project. If anything else, please let me know and I'll make sure the project gboard will get updated.
- Rajat: wasip2? Not released yet. One PR and I have further PRs, with the first being stuck.
- Michelle: I'll follow up with you separately on the Go SDK.

Wednesday, October 16th, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Kate
- Joe
- Adam
- Caleb
- David

Agenda (please add [yourname] to items when adding them)

- Shim version - continue to label prerelease?
- KubeCon SpinKube release
- How much do we want to ease the breaking changes around changing domain name
- Joe: https://github.com/spinkube/spin-operator/issues/325
-

Notes

- Shim version
    - Kate: Joe raised the question of why we are labeling this prerelease. Let's talk about it. I would like the following to call it stable

        - Expose pod level metrics: https://github.com/spinkube/containerd-shim-spin/issues/180
        - Shim processes are orphaned if you tear down your cluster: https://github.com/spinkube/containerd-shim-spin/issues/195hub
    - Kate: If we land those things I feel like that would be a reasonable gate
    - Adam: Rather than pre-release make it v1.0.0?
    - Kate: Lots of projects don't follow semver for prerelease stuff
    - Joe: We are bundling pre-release and 1.0 together. Pre-release is just a tag in the Github release version. Sub 1.0 releases can be non-pre-releaseases
    - Kate: I see pre-release as not production ready, Joe sees it as not an official release.
    - Caleb: We can just stop marking as pre-release and push 1.0 decision down the road
    - Kate: GitHub: "Optionally, to notify users that the release is not ready for production and may be unstable, select This is a pre-release." I think we should keep using pre-release flag until prod ready
    - Adam: I believe post 1.0 it changes how you filter for the latest release
    - David: I think pre-release is a good option for rcs and quick features we want to get out. Still, logging all the features we want for 1.0 would be good, even if we cannot get it out before KubeCon
    - Kate: After hearing it was made for 1.0 post distinctions, I don't feel the need to be clicking it.
    - Caleb: Any other points, or have we reached consensus? </silence>
- Kate: Do we have the idea of a "SpinKube Release", a collection of version which are a batch of SpinKube. One benefit is marketing. Another is that folks would know they work together at this version. Downside is it implies that folks need to have curated versions that work together.
    - Caleb: Mechanically, how would we want to do this. In OTel, they have a bunch of packages similar to us. They version them all together in lockstep. I could imagine us having a release in Spin operator that would bundle up components. Should we wait to 1.0 since the projects may not have reached the stability needed.
    - Kate: We may have a good marker coming, Spin v3. If we update the docs / blog post that tells folks that these 3 versions are Spin v3.0.
    - Caleb: Good point. I hadn't thought about locking into Spin's release cycle.
    - Kate: Might as well
    - Calab: Locking it all together is more work for us to curate the release to ensure compat.
    - Joe: Can we specify the version in the Spin executor CRD. Or, do we say this is the Spin shim executor.
    - Kate: No, not right now. The executor is not what provisions the shim. It just uses it. At some point if we have a more mature Runtime Class Manager, we should be able to do this.

    - Joe: At this point we don't have a lot of flexibility. If we could specify the min version, then we wouldn't need to build a bundle.
    - Caleb: If we could install multiple versions of shims, we'd be able to do this.
    - Kate: There is no format for documenting features alignment. It seems important to have something like this.
    - Adam: Might take some investigation, but is there a way to do this programmatically, like have something in Spin (operator?) to do this?
    - Kate: There might be a way with a validating webhook, but it is obscured by the runtime class.
    - Caleb: The compatibility matrix would help. A reasonable next step would be to add the shim to the matrix.
    - Kate: I think that would put us in a good place and help us to have a better conversation about this. Can anyone right now think of SpinKube / shim changes that would be noted?
    - Joe: Runtime class installer and shim share the same version. If you install kwasm, you can install it by version. Also, if folks want to use the k3d cluster, it already has the versions of the shims.
    - David: I think the tough part here is that we can be sure that runtime classes have naming conventions but we internally in AKS when deploying we applied versioning into the containerd config. Right now, the runtime class says it is a "spin" runtime class. So, it finds the binary on the path with that version baked into the file. That is not present in anything in the k8s level
    - Kate: Even there we are not giving the sh
    - im version. We are giving version v2.
    - Joe: That "v2" version is actually the containerd shim compatible version. In AKS we baked in a name like containerd-spin-shim-0-2-3-v2 for the file name.
    - Kate: This would take updates to the node installer. I think it would be worth it. Not sure if we add that to kwasm or the runtime class manager.
    - Caleb: Is this all for programmatic version determination.
    - Kate: How would this be helpful for targeting versions?
    - Joe: As a user if I install a k3d cluster from the release page. I have to do an extra step of installing the spin helm chart in my cluster. I have to choose what min version of the operator to ensure the shim works.
    - David: Naming scheme is helpful for handling an old app. Similar to factors, in that the app can target it but
    - Kate: I think we are good. I will work on a dependency matrix. I'm going to pass it off to how do we minimize the impact of changing domain names.
- Caleb: We landed a skip to **change the domain name of the CRDs in the SpinKube** projects. Changing the domain name for CRDs is a breaking change and a drastic one at that. Adam and I were doing some investigation. There is a spectrum of how breaking the change will be. One end is that folks will be fully responsible for replacing old CRDs with new CRDs. We could have scripts that properly migrate the SpinApps in your cluster. What if you are using Argo or other and that wipes out the change of the script or

operator. We could extend the operator to handle 2 domains and maybe get it in for KubeCon.
    - Adam: I am going to push against the operator having two CRDs. Going to introduce a lot of dead code. Could maybe have 2 operators running side by side.
    - David: This is going to happen once. It can NEVER HAPPEN AGAIN! Make the choice to break. The Argo point is a great point and I hadn't thought about it initially. The best we can do is provide some sed -- find and replace is not that hard. How many people have active workloads out there and how big of a pain is it? Kind of painful. But we could give good instructions -- run this script to remove these objects. Could have the operators running side by side, listening to different namespaces
    - Adam: I started going down that path and did run into a few issues
    - Caleb: Wasn't it that they were creating resources with the same name -- i guess you could put them in a different namespace.
    - David: Yes, great idea. Parallel namespaces. When you're ready, go delete the other things
    - Adam: One other issue that could come up is the dual version thing where we have two things running. There could be two of the same up running if you don't do a delete of the old domain before a fresh install of the new one. I'll keep investigating this path
    - Caleb: I think a question to ask is -- is anyone using this in production?
    - David: The only solution we have really endorsed is the WASI node pools. We look forward to having a supported solution of SpinKube. I don't know of anyone using it today, but i'll do some research
    - Adam: Since this is a decided thing, do we want to get it merged, then do the documentation work?
    - Kate: Since SpinKube does not have marketplace offerings right now, we should probably use this time to get it in. Do we also have a marketplace offering in Google.
    - Caleb: We do have a Civo marketplace offering. I'm going to go on a limb and say it will not be a problem there yet.
    - Kate: I was just countering my point thinking we are not in 1 marketplace, but we are actually in others too.
    - Caleb: It seems we should merge, and land in the middle of you are on your own and here is the best path. We will add the docs and explore 2 operators running side by side. Can we move forward due to having more topics.
- Joe: The issue is about supporting sidecar containers https://github.com/spinkube/spin-operator/issues/325. This is a proposal to add the sidecar functionality to a SpinApp. I did some prototyping in the Spin operator, but I don't believe it's ready for inclusion yet. The containers key would contain a list of Linux containers to run as sidecars. This is a non-breaking change since it's an additive field. I hope to get some comments / thought on it.
    - Caleb: I heard you say that you filter for only Linux images. Did I miss hear you? I don't see it in the code.

- Joe: I haven't tested if the image path is another Spin application. Since we are using the same executor, it should work.
- Caleb: What happens if I load a spin app listening to a port and another listening on same port?
- Joe: That is a good question. I think if both were running side by side, they will have a conflict on the port in use.
- Caleb: I like the idea conceptually. I also see use cases. My concern is what applies to the main image vs side car.
- Kate: I think we should consider what it would be like to have sidecar spin apps. It would be relatively easy to lift the port to the SpinApp structure. In here we are allowing an entire container type specified here, but there is some incongruent controls when Linux vs Spin app.
- Joe: I think it would be difficult to initially run to Spin apps side by side. If we narrow the scope to Linux apps, it would be easier. I also asked if we could allow folks to specify a port, but the response was negative. I had hoped to use this for running an Envoy sidecar. What is the usecase for 2 Spin apps side by side?
- Kate: It's not the best use of a Spin app, but a group may want to run 2 of them side by side due to working on 2 different apps. I would like to continue, but I need to go. We should continue the discussions.
- Caleb: I agree with Kate. Technically, it is easy. However, I believe Danielle's pushback was due to how we want to handle ports. I would also like to get a customer to discuss what and why they want sidecars.
- Joe: I will work on a SKIP about this.
-

Wednesday, October 2nd, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamIvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Kate Goldenring
- David Justice
- Adam Reese

Agenda (please add [yourname] to items when adding them)

- Fermyon SpinKube Azure Marketplace offering [Joe]
- Wasm + container workloads experience [Michelle]

Notes

- What is the KubeCon plan? Azure Marketplace offering status and Spinv3?
- Kate: we plan to release spin 3.0 by kubecon
- Joe: Azure Marketplace offering is currently in beta. Tried it out and it is working well. We are very close to being able to officially release it
- Joe: If there is an intention to release it for kubecon and deprecate node pools
- DAvid: as soon as the fermyon marketplace offering is availabl, we would point the documentation for WASI node pools into using SpinKube via the Fermyon marketplace offering
- David: We'd have a section on the ways to run Wasm on AKS. Thats where Thorsten and others docs would land
- Kate: We need to hear from Thorsten and Vaughn whether we could get that marketplace offering out of beta by kubecon
- David: When do we think Spin 3.0 will be released? And all of the components?
- Michelle: Lets check in with Caleb on timeline
- David: I just want to make sure that the docs experience is optimal
- Michelle: internally we've found documentation Fridays to be helpful. We will need doc changes for the domain updates (to spinkube.dev).
- Adam: The documentation on that should be ready to go. May want to add reinstall to that. PR is up and queued
- Joe: in addition to documentation, i want to set up telemetry and metrics are set up and that customer support have a great experience. When customers have an issues with marketplace, what is the process for forwarding customers to us
- Kate: If we document triggers somewhere, will want to document that MQTT was added
- Caleb: Spintainer should be documented as another executor
- Michelle: Also just an FYI - we're looking at mid October as the first rc cut of Spin 3.0

Sidecar container skip

- Michelle: There are a few usecases that I've heard of people wanting to run sidecars alongside Wasm. They'll have some stuff abstracted in the container and one container serves multiple Wasm workloads. Instead, in this case, they want to run one container for one wasm. Example use cases: (1)Service mesh land: people sometimes use sidecar containers. (2) pushing logs or metrics to a central location. (3) Want to use wasm but using a proprietary binary or some encryption work that cannot be put in Wasm. They want to run part of their workload as a sidecar container that has to start up before the WAsm workload and run for the entire lifetime of the Wasm workload. Currently the shim does support running containers alongside =your workload, so if you had something that injected it in your workload, you'd be good to go. So the question is, do we want to support this usecase by adding a way to configure SpinApps in the CR.

- Joe giving a talk around this: https://sched.co/1i7oH
- David: Customers may use Dapr to bridge the gap. Some people want to use Wasm as sidecars to their container workloads -- little side car microservices. More of the first. I am not sure if there will be less need for the container side cars as Wasm functionality increases
- Michelle: even though the wasm sidecar is a smaller use case, it could be a good way to onboard people onto Wasm. On the first point of integration of dataservices and Dapr, Zeiss is using one container for N spin apps.
- David: Definitely good to point people towards ambient Dapr rather than all sidecars. Same with service mesh -- want ambient mesh
- Michelle: We can support this, but what are the actual benefits? Onboarding is one of them
- Caleb: So networking -- communicating over localhost is one reason?
- David: Telemetry forwarding -- OTEL collector sidecar
- Caleb: I think if nodeaffinity and more pods solves it then hard to motivate
- Joe: I want to emphasize, if you are running a sidecar container, means you have the same cgroup namespaces. On the SpinApp CRDs, what happens if the image is pointing to a normal linux container?
- Kate: would run fine
- Joe: So all we need to be able to do is specify multiple images in the spinapp
- Michelle: not all executors will support running linux containers. We could add support for multiple images but want to make sure people have a consistent experience across executors so could add support for sidecar containers in the executor
- Kate: That means we would run the same sidecar for all spinapps of that executor?
- Caleb: I think we need to be more comfortable about having more executors, you could have 10 executors
- Michelle: This gives me a lot to go on. I'll drop the SKIP in our channel when it's ready
- Caleb: Are we diluting our messaging by saying you can put a linux image in the SpinApp CR
- David: I think this problem comes at us more if we talk about inspecting the component inside the artifact. We look at the interface it's importing and KV is one of them, we could then make sure to give it the appropriate identity/rights.
- Kate: we could detect that its a container and jump over all the cool steps we were doing with components
- Michelle: Back to containers, it would be nice to have an example or two around the service mesh and telemetry patterns.
- Caleb: if we support multiple containers in SpinApp CR, in the shim we'd have one main container and then sidecar which is fine. Does this get us to where i can put 5 spin apps in a SpinApp CR
- Kate: 5 spinapps could tell the executor to run all 5 in one process
- David: i think that makes sense in terms of the containerd sandbox API
- KAte: may want to make the SKIP generic and say, "these are the set of images i want in this SpinApp CR", executor go handle them"

Wednesday, September 18th, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamIvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947
Recording:
https://us06web.zoom.us/rec/share/mU3ory6GvbBIFlHVZv1HJNT8ftOw7gZIpHt5XNLw6Qgb8QedwdCJyrKgY5uTlCk._MZttxPRYYNsn_h4
Password: 6*VTb7st

Attendees

- Matt Fisher
- Kate Goldenring
- Ryan Levick
- Radu Matei
- Vaughn Dice
- MacKenzie Adam
- Caleb Schoepp

Agenda (please add [yourname] to items when adding them)

- Meeting invite ??? [Radu]
- Shim PR for Spin factors https://github.com/spinkube/containerd-shim-spin/pull/189 [Ryan]
- Spin Operator 0.3 blog post [Radu]
- Spintainer – what is it? [Caleb]
- Runtime-class-manager misc. [Vaughn]
- Wasm + container workloads experience [Michelle] (most likely postpone for next meeting)

Notes

- Meeting invite
    - Radu: Christoph asked for a calendar invite that he could add to his calendar for this
    - Matt: That should be fairly simple since it is fairly static -- just need ical. I can take this on and share it along
- Shim PR for Spin Factors
    - Ryan: starting high level, for those who are not aware, we have done a rewrite of the internals of Spin based on a concept called factors, which is bundling of

functionality. Can build a runtime but putting these factors into a collection. The thought behind this was a way to make building runtimes composable. Building a custom runtime should be easier now -- doesnt have to be the CLI runtime. This PR brings it over the SpinKube. Should be a no-op and not bring anything new to it. Not really true, because the shim is on 2.6 and there is a delta between 2.6 and 2.7. We are essentially targeting Spin main which is targeting 3.0. There are some relatively small and non trivial that change between 2.7 and 3.0
    - Vaughn: in draft because waiting for 3.0?
    - Ryan: In draft in part because all the triggers need to be updated. Most immediate thing is that Joe has a PR open reorganizing / refactoring the shim. I based my PR on that PR, so we probably want to merge that first or close that PR.
    - Radu: For SQS, redis, MQTT command trigger etc, is there a request for those maintainers to update those to factors too. So the ask is: “if you’re a maintainer of any trigger, go review”.
    - Radu: is there a 3.0 breaking change that we need to warn users of?
    - Ryan: There is an issue on the spin repo (https://github.com/fermyon/spin/issues/2815) which says to track, agree upon, and document all of the breaking changes. We don’t have all of those changes fully tracked. I started a list here but probably missing 1 or two. A lot of these changes have happened overtime from review (effectively 2 people decisions). Maintainers can look at this list and decide that we want to make these breaking changes and then document them.
    - Radu: where should people give feedback on this? On the issue 2815?
    - Ryan: Makes sense to me. First though, we need to make sure this is a complete list.
    - Radu: We cut a Spin operator 0.3 a couple weeks ago. We usually do a blog post. It contains tls, http outbound improvements. Do we want to have a blog post on the spinkube.dev site? Volunteers?
    - [Matt volunteered]
    - Kate: when Ryan was talking, we don’t have a release of the shim with MQTT, so that won’t make it .We can try and squeeze that in, but it might be messy. The release does not have MQTT.
    - Vaughn: Triggers release with shim and kwasm operator but we are still adding them to spin operator release note?
    - Radu: People care about features regardless of where it is -- SpinKube or Spin Operator
    - Matt: User is mostly concerned about the features not the technical projects that comprise SpinKube
    - Matt: This is going on `spinkube.dev` right? (yes)
- Spintainer
    - Caleb: We just merged Spintainer into SpinKube. New executor. Instead of running on shim, can run Spin app in a container. Enables using custom triggers and images. Should unblock specific use cases.

    - Matt: seems largely about getting a different version of Spin than the shim has
    - Caleb: I haven't done any perf testing but my assumption would be that the shim would be faster because you are not starting up an entire container
    - Caleb: Another reason is in the case you dont have perms to install the shim on the cluster
    - Radu: How do image pull secrets work in this case for fetching the spin application from a private repo
    - Caleb: That is a good issue to file and test
    - Radu: assumption is that the image pull secrets used would be for the spin container not the Spin apps.
    - Caleb: Will likely need to project the creds into the container
    - Matt: Is this ready for review/testing
    - Caleb: It's merged. We should document this but i wouldn't put it front and center
    - Matt: Yeah how to position this is interesting. Focus on the version story or should we hide this from public docs
    - Caleb: I don't think it should be hidden from pub docs. It just shouldn't be on the spinkube hot path docs, more in the misc
- Runtime class manager
    - Vaughn: I have been familiarizing myself with the runtime class manager. We have some milestones for the project. First is to swap in RCM for kwasm. That milestone revolves around the helm chart https://github.com/spinkube/runtime-class-manager/pull/204. RCM still needs some distro detection that kwasm does have -- don't want breaking changes. This is a pre-existing PR (84). We should put more eyes on that.
    - Matt: This is something we wanted to do in 0.2, so getting momentum behind the work is great! Do the maintainers need other assistance?
    - Kate: refresher: can you remind me what runtime class manager aims to be beyond kwasm? My understanding is we wanted to make more of an operator and grow into more. Could you refresh us?
    - Vaughn: one big improvement. Today, you have to apply the runtime class manually in SpinKube. RCM will handle that. It works on the idea of a "shim custom resource". In that resource, you can declare the label, shim binary, metadata for the runtime class. Once you apply that to the cluster, RCM will handle .RCM uses labels instead of annotations. This topic came up recently with the Azure marketplace; annotations don't persist, labels do. So a big reason we want to get the RCM MVP in. Other goals in the project that I'm not really up to date. Cristoph would be better to talk about them .KWasm feature parity is the first milestone.
    - Kate: you mentioned that labels persist more than annotations. Is that documented? That's a new realization for me.
    - Vaughn: Joe from MSFT just created an issue yesterday, will post the issue.

Action items

- Matt Fisher: Create calendar invite for SpinKube meetings
- Caleb: support private spinapps in spintainer

- Review RCM helm chart PR https://github.com/spinkube/runtime-class-manager/pull/204 and PR 84

SpinKube Project Meeting

Meetings are held bi-weekly at 8:00 am PT/11:00 am ET/5:00 pm CET. Come listen in, ask a question, or share an idea!

NEXT Meeting: Wednesday, September 4th, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Caleb Schoepp
- Adam Reese
- Michelle Dhanani
- James Sturtevant
- Vaughn Dice
- Radu Matei
- Kate Goldenring
- David Justice
- Matt Fisher

Agenda (please add [yourname] to items when adding them)

- Does the operator know about installed / available executors? [Radu]
- SpinKube Azure marketplace offering [Vaughn]
- Bonus: Containers alongside Wasm workloads [Michelle]

Notes

- Executor validation
    - Michelle: not sure about validation. Caleb?
    - Caleb: looking at the code now to refresh my memory. The constants just were never cleaned up. All we validate in the webhook is the presence of an executor.
    - Radu: So the handling of that is left to another controller.
    - Caleb: do we need to document this?
    - Radu: that would be ideal
    - Caleb: where would you expect the doc to live?

    - Radu: no strong opinion. I can take the action item to create a doc for this.
- Vaughn: update on the Azure marketplace. Versioning snafu. We published 1.0 of the marketplace bundle; we created a 0.1 bundle, but we're still getting the 1.0 as the latest. As a backup, sticking with 1.0 train then when breaking changes in SpinKube, just rev to the major change. The version itself is not presented on the marketplace offering page. Not headlined, but wanted to bring up the workaround.
    - David: the workaround you talk about is getting rid of the v1 artifacts?
    - Vaughn: as users we can't. On the backend maybe? Ideally, we'd version this in sync with SpinKube.
    - David: it doesn't show up in the UI. How does someone figure out the version they are running under? For compatibility reasons? Haven't tested yet, but want to create a TF plan with the extension enabled. I've been told internally that you cna upgrade. I'll try this and come back with feedback. I don't really know how breaking changes happen in Azure marketplace.
    - Vaughn: IIRC the concept of "release tracks" that follows semver. Not sure how that works with prereleases, but we would be in prerelease territory.
    - David: I imagine others are in the same situation. The docs are lacking. Good opportunity for feedback to the marketplace team. I'll share that with them as well.
    - Radu: it was an awesome experience creating a cluster with SpinKube enabled from the marketplace
- Michelle: talked to end user company recently; they are interested in containers and wasm in the same pod. Right now, you need to inject a container in the pod. Is this something we'd encourage / define in the Spin app manifest CRD?
    - Kate: clarifying – is it an init pod, or random sidecars?
    - Michelle: logic they can't do in Wasm; they want to use the same logical app.
    - Kate; no way to define that in the Spin app. You can with just the shim. In the CRD we only define one image. SKIP?
    - Michelle: I can volunteer to do that SKIP.
    - David: it gets interesting when you consider different executors. Different abilities to run container workloads vs. wasm containers. Is that something that gets validated?
    - Kate: maybe we need webhooks that validate/ reject based on executors. Already fields that some executors can't use.
    - Michelle: and perhaps adding a field for that.
    - Radu: noticing people interested in wasm + containers side by side.
    - Radu: putting this on the agenda?
    -

Meeting: Wednesday, Aug 21, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamIvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- Caleb Schoepp
- James Sturtevant
- Kate Goldenring
- Danielle Lancashire
- Vaughn Dice
- Michelle Dhanani
- Radu Matei
- Christoph Voigt
- David Justice

Agenda (please add [yourname] to items when adding them)

- V0.3.0 release for Spin Operator [Caleb]
    - CA secret annotation [Danielle]
    - Executor validation — https://github.com/spinkube/spin-operator/pull/308
    - OTel support
    - Domain rename (breaking change)
- Azure Marketplace PR — https://github.com/spinkube/azure/pull/4 [Vaughn]
- GKE SKIP — https://github.com/spinkube/skips/pull/8 [Vaughn]
- Community repo updates [Radu]
    - Add reference to project meeting, notes — https://github.com/spinkube/community/issues/1
- Runtime class manager milestone 1 — https://github.com/spinkube/runtime-class-manager/milestone/1 [Radu]
- MQTT trigger in the shim — https://github.com/spinkube/containerd-shim-spin/pull/175 [Kate]
- Spin and SpinKube CNCF Sandbox application status update [Michelle]
- Update on containerd pushing metrics to metrics-server? [Caleb]

Notes

- V0.3.0

    - Caleb: 2 things to discuss: do we want a minor release and anything we'd wait for? And domain change would be a breaking change, are we ok communicating in this minor
    - Danielle: semver says yes, since we're pre-1.0
    - Michelle: plan for if anyone else is adopting the new domain name? Maybe we adopt it on a roadmap?
    - Caleb: the understanding of the SKIP is that all projects would adopt it.
    - Radu: runtime class manager has a few CRDs and not sure where we are.
    - Cristoph: not sure, I need to look into this. I agree that we should have the same domain.
    - Caleb: other features?
    - Danielle: the redesign of the executor CRD to avoid 2 consecutive breaking changes. If we want to include it,I can prioritize a draft next week based on the async conversation.
    - Radu: link?
    - Danielle: we have an issue already, but not enough context. I can take the action item to add more context.
    - Michelle: I know back to back releases with breaking changes is not ideal; But I'd vote for releasing what we have. Don't want to rush the design.
    - Danielle: alternatively, we can punt this until that change is ready.
    - Caleb; PR in draft. Not a large amount of work, but I'd be ok with a breaking change that has both.
    - Radu: releasing often what we have and keeping the velocity up would be great.
    - Caleb: loose consensus for shipping what we have in 0.3, then break in 0.4?
    - <loose agreement>
- Azure marketplace PR
    - Vaughn: for those savvy in Azure, this adds the rest of the technical assets for the marketplace (arm templates, UI assets). We have a preview that is looking good. If you have time, David, feedback from you would be great.
    - David: if I were to give you an azure sub, would that work?
    - Vaughn: that would work
    - David: I can give it a try! That's awesome!
- GKE SKIP:
    - Vaughn: we have a SKIP Kedar worked on; reminder to review and move it forward; I've scaffolded out the initial chart for that, would be great to move along.
    - Radu: the change is minimal, so checking with Kedaar
- Community repo updates:
    - Radu: Realized we have a community repo that has nothing in it. This is me asking if there is anything I should wait for before opening a PR there. Is there any reason to not put it there? No one says anything Anything else we want to have in that README?
    - Kate: It might be nice to have a summary of SpinKube
    - Radu: I can do that. I'll try to keep content to a minimum to prevent it from getting out of date.

    - Christoph: Add a link to this google doc?
- Runtime class manager milestones
    - Radu: There are 3 milestones. I know we're working on milestone 1. I was curious if that milestone needed help. Maybe we can open good first issues or help in other ways?
    - Christoph: Project overall could use some help. We're currently in a state where we can do the majority of milestone 1 things (tested on Kind). Majority of work remaining is some automated tests. And someone stepping up to make a Helm chart.
    - Vaughn: I can volunteer to help test the draft helm chart PR
    - Christoph: Did I make one? Hahaha
    - Vaughn: We can model after spin-operator testing methodology and iterate to testing on specific providers
    - Christoph: I don't have super strong opinions here. It would be good to keep the tests similar to spin operator.
- MQTT
    - Kate: I have a PR up with the MQTT trigger added to the shim. Tests are not passing right now. It has to do with builds failing for static musl; So kind of stuck on getting tests to pass within the automated testing for the repo. I can try to hack on it a bit more ,but cross-builds on my machine working fine. I'll keep trying; But this will bring support for MQTT.
- Metrics:
    - Caleb: was talking to one of our interns who is blocked by the shim not sending metrics to the metrics server. Any updates here?
    - Kate: for starters, I've been wanting to create an issue, so will do that later today. There appears to be an issue with how we're getting pod metrics for apps running in the shim. Even if it's a container, we're not creating pod level metics. So kubectl top, HPA, all use pod level metrics. Weirdly, it works on k3d. Not on kind. I have this matrix of versions of containerd, no patterns yet. Maybe with yuki or rust-extensions on how we're reporting metrics.
    - James: I haven't done more than giving some advice on where to look so far. We fake out the sandbox container; just a stub inside runwasi; wondering if it has something to do with that.
    - Radu: One of the very early issues you hit was cgroups 1 vs cgroups 2 and containerd metrics. To add to that complex matrix could you try comparing the cgroup types (I think I mistook this ….)
    - Kate: Sounds good
- SpinKube, Spin, CNCF
    - Michelle: good news and good news! Activity on the SPinKube application, folks are asking questions, so fingers crossed! We also submitted the Spin CNCF application, thanks all for the patience here, we worked through the Spin governance first. The Spin CNCF application was a good exercise in making sure we have all pieces in place. As far as I can tell, the TOC is looking at both

applications. There was a question around whether SpinKube would be part of the Spin application, so wanted to bring this here.

- Michelle: my opinion – they are 2 separate projects. Spin itself has the core runtime and devex in scope, plus SDKs, plugins; pretty large surface area in itself. SpinKube has its own subprojects, SKIP, governance. The way I've been thinking about these two projects is akin to Istio and Envoy, for example. One is the core experience, with its own governance, and the other its own subproject and governance. The question was asked: will SpinKube be part of the Spin project. How is everyone else thinking about this?
- Kate:
- Radu: given the distinct user bases, and that Spin has users that would never
- Caleb:
- James: my initial reaction would be for them to go together; when I look at the Spin operator, they feel like tools that are specific for Spin. I know the runtime class can install any runtime class, but it seems that SpinKube would be a subproject. Your comparison to Istio and Envoy was interesting, havent' considered it that way.
- Michelle: I had the same exact thought process; thinking through it, would love to hear your thoughts once you have a chance to think through it. Helm and Kubernetes is another good example. Helm moved out of the Kubernetes project because it needed its own ecosystem.
- David: I don't have a strong opinion. The direction in general is sufficient to be satisfied; Envoy/Istio analogy is a good one. Another analogy is containerd and its subprojects. Related, wouldn't be able to exist without, and having an umbrella. Honestly though, I wouldn't be upset in whichever way it went. Whatever makes the least amount of friction makes sense.
- Cristoph: I agree with David on the best outcome. From the POV of runtime class manager, once we reach milestone 2 or 3 it's not dependent on Spin. So not that strong relation there, unless we were happy that we found this home. The *SPIN* organixation would not be the right one, since SpinKube. On the operator side, I was surprised it wasnt' in the Spin repo itself to begin with. Not sure it helps though.
- Radu: we'd be open for SpinKube to support other application formats.
- Caleb: my topic from earlier: on board with them as different projects. How about triggers and plugins? MQTT trigger – Spin or SpinKube? Have we thought about that?
- Michelle: triggers can live anywhere;
- Kate:
- Michelle: we're putting both of them in the CNCF because governance, neutral place to contribute. We want to answer: what is best for each project?
- Radu: projects having lives of their own.
- David: have the projects progress at whatever pace separately. From the naive observer looking in, it's not easy to tell that Spin really is targetted to things

        outside of Kubernetes. If you have k8s glasses on, you're locked into that mindset. Having them separate helps them have a life of its own.
    - James: after this, I land in the same spot. I wan't aware that the longer term goal of SpinKube was to potentially bring in other kinds of apps. If that's the case, maybe the name might be misleading? That might be something to consider.
    - Michelle: thank you. We pitched this, let's think about next steps.
- Issue

Wednesday, July 24, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- MacKenzie Adam
- Caleb Schoepp
- Michelle Dhanani
- Matt Fisher
- Andrew Steurer
- Adam Reese
- Kedaar Sridhar
- James Sturtevant
- Joe Zhou (mossaka)
- Vaughn Dice

Agenda (please add [yourname] to items when adding them)

- [Caleb] Skip to change CR domain
- [Caleb] Spintainers update
- [Matt Fisher] docs SKIP update
- [Kedaar] GCP repo Skip
- [Joe] seeking updates on azure docs PR
-

Notes

- [Caleb] Skip to change CR domain
    - [Caleb] Looks like I have a few approvals. I'd like to talk about two things ~ what do we need for consensus in SKIPs / does anyone have any thoughts on this?

    - [Michelle] Since SKIPs are org wide, does it make sense to get a LGTM from every project?
    - [Caleb] That seems reasonable to me. Thinking about this, we should add this to governance?
    - [Fisher] Good feedback, is there anything in SKIP 001 that would have information in there?
    - [Caleb] Ya, I don't recall anything in there about the decision making process in SKIPs. I'll hunt down approval for approval from each project and open an issue in governance about how we decide on SKIPs
- [Caleb] Spintainers update
    - I've been working on a new executor type called Spintainers. It's an alternative to running SpinApps with the shim. We have Spin running directly in container. Useful for scenarios where they can't install the shim or want to use plugins that the shim doesn't support. It's currently a draft, but it's ready for folks to experiment with. Also I'm going on vacation this Friday if someone wants to take this over. Vaughn, maybe now's a good time to chat about whether CI lives here or Spin. https://github.com/fermyon/spin/issues/2677
    - [Vaughn] If people here have strong opinions, check the issue above. It could live either place, but Spin seems like a natural place
    - [Caleb] The image we're talking about basically runs Spin… Go bikeshed and put opinions in that issue. That's all I had for Spintainers. We can move onto Fisher's item.
    - [Fisher] I have a question about Spintainer. At some point, do you think there would be detraction where we'd need to inject k8s or SpinKube specific functionality.
    - [Caleb] Can you give a specific example
    - [Fisher] As containerd evolves and more features are added.. IF there are features in containerd shim spin that aren't in the core runtime, would there be a separate docker image that's meant to keep parity. Or is it just a wrapper around Spin itself?
    - [Caleb] I don't know… I know we aren't necessarily trying to maintain parity between executors. I don't have an answer for you right now. I would imagine that the shim is more fully features, but I don't know if that's necessarily true. I don't know if that answers your question but I don't think there will always be parity between the two
    - [Fisher] Gives me high level understanding ont he difference between the two so that's context enough.
- [Matt FIsher] docs SKIP update: https://github.com/spinkube/skips/pull/7
    - [Fisher] I've been writing a SKIP based on my experience looking at the documentation and trying to show it as one holistic project (vs the combination of 4). One of the things that is coming out of this, I'm trying to think about how we want to talk about the project. This is the thing you install and work with the project. I'm trying to reconcile between those two methodologies which is partially part of the docs. Prior to the docs it was part of it. If you looked at the docs

before, it was high level but didn't give a good sense of everything that's in the architecture. WIth this SKIP, I'm trying to come up with a sense of guidelines on writing style, provide context around how we write the documentation. I'm also trying to find out how we'd like to shape the documentation going forward. Do we want to have instructions on how to compile each individual project or do we want to have a product facing take. I'm trying to get a sense of how to do that at a high level.
    - [Kenzie] excited docs are getting love, would consider adding personas to help think about this
    - [Fisher] Yes, I have three in mind! Developer who just wants to kick tires, contributor who is interested in contributing to the project, and then the operator who wants to maintain SpinKube itself and wants to tune/maintain the cluster. So, I didn't explicitly how each side should make to personas, but I did put a note as we're writing content for the documentation.. We should have one persona in mind so we can fine-tune the content to speak to them. But i haven't made a decision that quickstart should only be for developers, or tutorials should only be for operators. If people have thoughts or feelings on how that should be structured, that would be great to call out.
    - [Michelle] I love that you're thinking about it! One callout I say in Discord, a user wasn't able to understand what all was offered and what were the holistic featureset is. I think having more of a focus in the area would be helpful to this user and others. Even myself, I'll chime in too. I have to look at the containerdshim code base to understand the example - for example, what triggers are supported. IT would be nice to bubble that up to the documentation
    - [Fisher] That's a good point. There's an overview page, but it's super super high level. Here are the projects and how it operates. A better understanding how how my app maps to this and triggers. If we want to have more user-facing documentation inside of SpinKube, do we want it to live there? I think user orientation documentation should live in spinkube.dev as opposed to reference documentation or high level architecture. That might be reference documentation. But ya, i think that's great feedback and good places to consider where we should focus on our content going forward
- [Kedaar] GCP Repo Skip
    - [Kedaar] GCP PMs have been interested in creating a repo that includes initial helmchart and future gcp-oriented helm charts as well as testing. If anyone has thoughts or comments, feel free to pop it in there. Vaughn had a great comment that we have an umbrella on the central helm chart, or even a charts repo to have some of those top level charts. For now, I think this is a good next step for pushing further spinkube integrations along!
    - [Kenzie] It would be good to see if we could connect docs to these partner integrations where there's more support (GCP, Azure, SUSE).
- [Joe] seeking updates on azure docs PR
    - [Jpe] Syncing some updates from the Azure docs PR created by Thorsten. Want to get it moving

    - [Kenzie] He's on vacation, don't think he'd mind you taking it over. Happy to keep eyes on it as well and provide feedback
- [Joe] PSA that I'm on vacation for 3 weeks starting Aug 1st
- [Kedaar] Popping back to approval, who are the maintainers I need to ping?
    - [Caleb] There are teams on Github who shows who are maintainers for each team https://github.com/orgs/spinkube/teams
    - [Vaughn] TBD on what quorum is needed?
    - [Caleb] Right now loosely get approval from each team, but we'll follow up on the issue i created on maintenance.

Action item

- SpinKube maintainer team on GitHub
- Joe - azure docs PR

Wednesday, July 10, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us06web.zoom.us/j/83473056051?pwd=gDFNQCxUG1OMzdBqTNXIQamlvHjhFo.1
Meeting ID: 834 7305 6051
Passcode: 880947

Attendees

- MacKenzie Adam
- Caleb Schoepp
- Sven Pfenning
- Kate Goldenring
- Adam Reese
- Vaughn Dice
- Radu Matei
- James Stuartevant
- David Justice
- Jiaxiao (Joe) Zhou
- Michelle Dhanani

Agenda (please add [yourname] to items when adding them)

- [Caleb] Discuss changing CR domain to spinkube.dev
    -
- [Joe] Discuss OTEL difficulties: https://github.com/containerd/runwasi/pull/582#issuecomment-2218311212

    - Background on set_global_default subscriber: https://docs.rs/tracing/latest/tracing/subscriber/fn.set_global_default.html
        - "Note: Libraries should *NOT* call `set_global_default()`! That will cause conflicts when executables try to set them later."

Notes

- [Caleb] Discuss changing CR domain to spinkube.dev
    - Going forward, if other projects are creating custom resources, what domain would we want to house them under? We should have a discussion as to whether we want to make this change and whether it's too early for this big of a breaking change
    - [David] Doing something earlier is better since you'll likely impact the fewest amount of users
    - [Caleb] To Kate's point, what should other custom resources use?
    - [Kate] Sven, what do you think?
    - [Sven] Sooner rather than later is good. Not sure about the governance, but since the project is called SpinKube I think it's ok. As a contributor, we should also think about being so tightly coupled to the Spin runtime since we can support other runtimes as well.
    - [Radu] SpinKube governance, it's a domain we registered when starting the project early on and we're donating as part of the CNCF submission. We'll hand off credentials to wherever it's registered
    - [Sven] If there's no veto, i think it's good to change
    - [Caleb] Good step forward would be to open a PR for this so people have a chance to express any further opinions. If people feel strongly about it, I can write a small SKIP, have that approved and do a PR
    - [Radu] That would be nice
    - [Caleb] That's all we have for this topic
- [Joe]: background: working on adding OTEL to the shim; applying that PR to the Spin shim. The difficulty: the shim and the Spin runtime set a global default for the subscriber. As a library, the Spin library sets the global. The shim is also setting up a global (TTRPC server). That creates a conflict. Talking to Caleb, we came up with two approaches:
    * use what the host has
        - Enable shim telemetry and Spin runtime telemetry, use the shim telemetry and bypass what Spin sets.
    - [Joe]: if we have a single one, we could send all data to the same sink. David has another thought around setting separate implementations for the app and runtime, but I think it's difficult.
    - [Caleb]: if one of the project doesn't use Rust tracing, we could have both. Not sure it's a good solution though.
    - [David]: not sure it prevents us from having macros; it's a sticky issue in Rust generally; not a pretty solution around it. Maybe we find the best of the worst solutions.

    - [Joe]: we could modify the Spin telemetry to not use the default;
    - [Caleb]: tried to go down that route, but had to touch all async boundaries, which was not feasible. Using the OTEL SDK directly in the shim may be the most straightforward.
    - [Caleb]: David, could you elaborate on the separate collectors?
    - [David]: I don't have data to talk about this objectively; intuitively, if I set OTEL config on my running workload, I'd like it to be respected. If overridden or ignored, that would be off-putting and unexpected.
    - [Caleb]: that makes sense to me.
    - [David]: if Spin is ONLY intended to be used as an executable, setting a global default makes sense. If intended to use as a library, setting the global default is a no-no.
    - [Caleb]: I can't think of other scenarios.
    - … next steps
    - [David] We may want to consider in Spin, the impact of running this library in other projects as well
    - [Joe] Definitely, yesterday you mentioned we can add the describer (?) based on the namepsace and see if we can do that in the Rust SDK
    - [David] I was going to run into OTEL SDKs b/c I think there are named subscribers. You can use a named tracer that is then used and has it's own configuration to what subscriber it goes to. That might be useful as well. A named tracer and then we can have one named for the shim. Then user uses whatever tracer they want
    - [Caleb] I don't think we're going to run into issues with OTEL. The madness is coming from Rust subscriber. What I can tell, OTEL plays nicely with multiple things
    - [Joe] hopefully!
    - [Caleb] I will work offline next steps
- [Michelle] We've been working on milestone planning… Triaging issues
    - [Kenzie] if someone gets involved, what do next steps look like?
    - [Michelle] Kate and I started triaging yesterday, we'll come back to this meeting along with an update on what that process looks like. The CNCF SpinKube channel is the best place to throw in questions

Wednesday, June 26, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us02web.zoom.us/j/82108121812?pwd=ZkhuSHExTXBPek9SaFVrTEszTDZPZz09
Meeting ID: 892 3054 4590
Passcode: 644621
Google Calendar Event

Attendees

- Matthew Fisher (Fermyon)
- Adam Reese (Fermyon)
- James Sturtevant (Microsoft)
- Kate Goldenring (Fermyon)
- Michelle Dhanani (Fermyon)
- Vaughn Dice (Fermyon)
- MacKenzie Adam (Fermyon)
- Radu Matei (Fermyon)
- Kedaar Sridhar (Fermyon)
- David Justice (Microsoft)
- Ralph Squillace

Agenda (please add [yourname] to items when adding them)

- https://github.com/spinkube/containerd-shim-spin/issues/147
- [Michelle] spin governance update: PR here

Notes

- Michelle: SIP for Spin governance; if you have thoughts, we'd love feedback.
- #147:
    - Kate: updating conformance tests, and there are a few TCP tests; running them on the shim fails; I asked if we want to support that.
    - Radu: <... missing … to summarize, context on remaining comment thread after conformance testing caught gap in spin shim from local user experience >
    - Fisher: If a container didn't have ability to perform some sort of kernel level action, how is that done with k8s today?
    - Danielle: https://kyverno.io/ a
    - James: seccomp, pod security policies so you can block things between namespaces. Containers are by default everything and then you lock it down, WASI is the opposite direction and that's a core difference in the way things work. I only briefly scanned the issue but I think I generally agree with Ralph that we turn these things on when needed and then have everything locked down
    - Radu: On the notion of opening everything up by default, this only happens when the app declares in the app package that it should access a specific host. The settings in the allowed network host are granular to protocol, port, and hostname. The main question is in the tension between what the app developer is declaring in the package and what the runtime allows or dis-allows. If you declare in Spin

manifest or wastime command, then `spin up` should respect that. The outcome today is an app that runs locally and then fails when deployed to SpinKube. To use that's a bug. The main question there… do we want to find a way to enable or disable at the executor level or not?

- David: I agree with Radu's POV. If the app runs appropriately locally, we can't have it running differently on the shim. It should have the same behavior and the same kind of pod behaviors. If someone applies sec filters, we should respect and deal with those appropriately. To me, it's totally a bug if the app behaves differently on the shim when compared to local DX. It would be unexpected and really weird to the user. Two balancing points, we need to aspire to have the same capabilities when we're running this app and developing it as we are going to deploy it (these expectations exist for containers). There's this defense in depth idea here. You can say you don't have access to system level features, when a SpinApp has constrained access we need to respect that.
- Radu: Putting my other hat on that I run a platform, and regardless of what my app dev asks for, I don't want to enable TCP. I think that's a valid scenario that we should enable and document. Then there's the next conversation about defaults. The main question for people who are interested in that type of platform, is there interest in defining what that feature looks like and implementing it?
- David: I think this is actually implemented… We could do this at an executor level; however, we also have the secomp filters, where we say at the container level you don't get access to that kind of stuff. Unless we see a reason to not to use those filters, I'd say it's not necessarily needed to have this configuration right now. Can someone see why we'd want to do that in the executor? In particular for k8s environments?
- Danielle: You could potentially use executor to default modify the secomp to avoid user error / set someone up for success
- David: Do we want to wait until someone say they want this?
- Radu: The reason why we're having this convo, Ralph pointed out this behavior should be controllable or the default should be the opposite. We're trying to figure out how we should think about this for the shim but also when we're configuring the whole stack
- David: I think he's taking the POV that Wasm is deny-default and less about Spin in particular. So, I see Spin in this use case as defining precisely what it's asking for… In Wastime, you'd be using that from the command line. You're providing that info at a slightly different level. You're stating what you're asking for and declaring it. From the Wasmtime, you'd have some sort of invocation at the command line to describe that. I think the use cases are very different. I'd guess Ralph is taking perspective of using Wasmtime as opposed to Spin. I think this is more of a purity argument (this is how Wasm works, this is how we should do this). I agree with the POV that Wasm has great features for constraining this, and we're still using it, but we have enough information to make good choices about it

    - Radu: I want to accommodate the type of feature request and not push back immediately, so I want to make sure as maintainers of project and we take the request while being clear we're open to talk about this and for folks who have a real use case, we're open to supporting them (even if we close the issue for now)
    - James: I think I missed that this is defined in the Spin.toml… From that perspective, I agree with what we land on here. If they ask for it, we should enable it in the Spin.toml. To your point, there are environments where you might not want to enable it from a platform perspective. That makes sense, thanks.
    - David: To Radu's point, making sure we're hearing and acting on feedback/requests, would it be sufficient to advise that if someone want to do this right now with secomp, or do we want to have some other solution (e.g., build config for an executor), or say this is a possible path and if we want to pursue this we can
    - Radu: We can summarize this conversation in the issue, and see if that's what Ralph has in mind. Then we can figure out if there is a feature request. But summarizing where we're at and describing where we're at in the issue would be excellent.
    - David: I'll take a swing at it.
    - Radu: Thanks!
- Cloud platform identities and simplify auth with SpinKube - https://github.com/fermyon/spin/pull/2566
    - David: Making it so we don't have to have secrets packaged with Spin (just using CLoud platform identities to access sensitive data). There's Azure Workload Identity, which is a mutating webhook, when a podspec is decorated with a specific label, that goes to a secret account, then there's a trust between Azure Active Directory … This terraform file is the clearest example of this. It deploys out a Spin workload that uses cosmos DB as a KV store. This is cool, and it kind of stinks at the same time. Wait, rolling back… imagine every Spin app we create needs access to certain resources. Each one should have it's own identity, and that identity has access to whatever resources it needs behind it (it has it's own roles and rights). Probably one identity with particular rights for each app. This gets us into a situation where we want to build this for SpinKube, then we'll have to provision provider resources for every cloud (e.g., creating that identity, assigning rights to that identity). So right now, we don't really have a way to do that in the operator. We could start building out providers, but perhaps we can consider using some sort of cloud infra operator (Crossplane). However, that's a pretty heavy dependency and I'm not sure we want to go down that path. I imagine we want to get to the place where we apply the SpinApp, the operator creates the pod, decrorates with identity provider, service account, and the user account along with the rights associated with it. The user doesn't need to see any of this, they just get the identity with the rights that they need. If we could build this stuff in that way, you could take away the need for secrets or understanding of identity in these cloud and I think it would be a great UX for folks. Thoughts?

- Michelle: If I'm understanding this correctly, we're going 1 step further than service accounts for listing out the rights for a particular app. IF you have a service account that's configured a certain way, and then your app gets installed, then the user token gets generated and applied.
- David: The mutating webhook catches the pod as long as it has the label, then it add environment variables to the pod spec, one is the Azure federated token. I'll get these environment variables and document them out. They're then used in all the Azure SDKs to exchange for a JWT token we use in Azure. It might be the token itself,I need to go check… Secret gets applied secretly. One of the tough things we have to do (not super tough) is make sure those environment variable end up landing in the container so the Spin runtime can take advantage of them.
- James: I think the idea is great. You deploy the app and identity gets provisioned at the time. Will that slow down the provisioning of the application itself?
- David: That's a good question… Identity creation is actually pretty fast, and I'm not talking about microseconds fast. Fast in the glocal scheme of things, so there's going to be a slight delay for first image. Then scaling up and scaling down should be nothing. First creation of app, we'll have a small hit than nothing
- Kate: Could we add prefixes?
- David: We could if we created our own mutating webhook, but I don't think these need to be exposed to teh guest. I think they're just the host of the runtime. These are just within Spin, until we get to components where you have a KV component we need to feed that variable into. The running process has access to the environment variables, you're not actually pumping them into the app variables.
- Kate: I flipped the SKIP. Part of that, there are a lot of k8s env variables that expect them to exist. If we don't inject env variables, then a lot of the k8s client libraries aren't accessible to users. We probably don't want to inject the variables you're discussing, but we can turn those variables off (but it's on a per-service level). More discussions are needed but that adds another use case for us to consider
- David: There's going to be a lot of stuff we don't want to expose into the Wasm environment. Like you're saying, we need to figure out how to sort them (what goes into the Wasm environment and what gets into runtime for runtime configurations).
- James: It feels like in that scenario, environment variables feel similar to turning on IPs and various request from earlier conversations. We want to say, hey this component can access these environment variables. At a higher level, these containers should access these environment variables by default (since we know we're in that environment).
- Kate: We could make a quick adjustment to pass only the k8s ones?
- James: I know you're spending a lot of time on this
- Kate: It's a tricky one, but I'm happy to spend as much time back and forth as needed

- David: I think most of our fear / concern about environment variables is leaking into the Wasm guest itself, and everything having access there. Perhaps adding something to `spin.toml` that has an allow-list of environment variables. From the host side of it, I don't think we're worried about it. We're basically running a trusted application, but when we get into the untrusted space (guest app). I think right now the host side of the runtime has access to this stuff anyway, we're just trying to figure out what we want to get to the guest. You could say * in your spin.toml, but you can say hey I only want to provide these environment variables into the Spin app that's running.
- Ralph: I want to lean into what Spin can do to see how close you can map certain things that are using Spin. At the same time, I want to walk up the latter to full functionality auth can provide. That guest boundary, what does the module get and have access to? We should think really really hard about what we should do. That's why I was talking about higher level protocols like HTTP. No guest should really want socket access, but if it does need it then how do you turn it on. Especially if we're considering moving across environments. On the other hand, I appreciate the idea that Spin can do it all and do all these kind of custom things. That makes perfect sense to me. Deciding how to traverse that is tricky. It's easy to have a Spin app developer ask for TCP, but the host doesn't allow that, how do we surface that in a nice way? That's hard. I just wanted to make sure folks know where I'm coming from and the concerns between the guest and host boundary
- James: Is the app variables the allow list?
- Kate: At the component level, you list the environment variables the component will have access to. Downside is the app developer needs to know what the environment variables to plug them into their application manifest. It comes down to how limited vs how verbose of an experience. If we do app variables approach, we can always do a more permissive approach in the future. Or do something along the lines of what David was suggesting. I think the scenario where the Spin App not know the environment at runtime we won't support.
- David: You may also want to have explicit deny. Azure federated credential - you may want to explicitly deny that for every component running under X,Y,Z. That way you don't have to know all of them, but just the ones that are scary.
- David: If we could loop back around, do we want to spin up this infra (do we want to provision identities in Cloud providers? Do we want to go down that path?)
- Danielle: I think it makes sense, but figuring out how to expose it to users is hard. If someone has "make this pretty in k8s API suggestions" then great. But I couldn't think of a nice way to make that work. Every Cloud provider wants something _slightly_ different. Ideally I want that thing to be tied to the executor rather than specify on every SpinApp, but I'm not sure how
- David: Decorating onto executor means every app is going to participate in this identity. Or is it a single identity, or ?
- Danielle: You still want workload identity, but you have a couple of classes of executor (I want Azure permissions please). So you can abstract away from the

average developer what needs to be set for workload identity to happen. So whoever is configuring executor deployments can make that happen for you.
    - David: for workload identity, for a Spin app, “use cloud identity == true”, that should be the only information… Wait nevermind
    - Danielle: You need to annotate pods with labels and other stuff right?
    - David: in the operator that should be done, the label for pod identity is (string), then the service count name. Those are the two things that go into the pod spec. The other information we probably need is the tenant and the subscription. That information would be needed on SpinApp or Exectuor. I’d need to look at other clouds and how they do it to think deeply about API structure
    - Michelle: Super generically, if there was a way to have a custom mutating webhook be configured so when a pod is created, a certain dance happens. That is very generically non-Cloud provider specific and that’s the dance?
    - David: I don’t think we even need a mutating webhook. When we get a deployment of a SpinApp, then we’re responsible for creating a pod spec and we can see things like we need a reconciler (operator or a provider). Then we’re going to proceed with the pod spec before we actually apply that with the information it needs. We’re reconciling a few more objects during that entire operator/controller loop
    - Michelle: custom extension mechanism in the operator that understands when something happens, my pod needs to be decorated with certain labels. Trying to describe this in a non-specific-Cloud provider to understand what needs to be applied. It seems like this needs to happen at creation time
    - David: I think we have all the information we need when we create
    - Michelle: It makes a lot of sense to figure out how to customize cloud spec to environment needs, but for example we allow in the SpinApp to add custom annotations we need. It seems like there was a different extension mechanism on create. We could even get rid of those b/c it would be specific to what your environment would want. Thanks David for helping me work through that, but I might need more time to think through
    - Ralph: That’s to David and James for channeling me when I was out, doing what was best for Spin and SpinKube
    - David: I’ll throw together a google doc for a potential SKIP on this topic

Wednesday, June 12, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us02web.zoom.us/j/82108121812?pwd=ZkhuSHExTXBPek9SaFVrTEszTDZPZz09
Meeting ID: 892 3054 4590
Passcode: 644621
Google Calendar Event

Recording:
https://us02web.zoom.us/rec/share/324MAH6SYk2_c7--M_KvWpWpWT5xRQIImuISGd6takrgpiaDIBzJrJjIc3OcPBA.7pW8fvt3-BJA70YH
Passcode: OVEB6L%c

Attendees

- Melissa Klein (Fermyon)
- Adam Reese (Fermyon)
- MacKenzie Adam (Fermyon)
- James Sturtevant (Microsoft)
- Kate Goldenring (Fermyon)
- Kedaar Sridhar (Fermyon)
- Mikkel Mørk Hegnhøj (Fermyon)
- Vaughn Dice (Fermyon)
- Jiaxiao (Joe) Zhou (Microsoft)
- Michelle Dhanani (Fermyon)

Agenda (please add [yourname] to items when adding them)

- [Till/Michelle] Update on Spin Governance
- [Joe] Update on the azure helm chart
- [Kate] shim configuration SKIP: https://github.com/spinkube/skips/pull/4

Notes

- **Update on Spin Governance**

Michelle: We have a version we're feeling good about, waiting to hear from our legal team. Hoping this will be something the Spin community will like to see.

- **Azure Helm chart**

Joe: New repo: spinkube/azure. To hold everything Azure-related. In the future, may hold Azure-specific resources like ARM templates, bicep templates (all the things in that are bundled up in a Marketplace offering).
Vaughn: Working on a Marketplace-specific Helm chart (some requirements around image references)

- **Shim configuration SKIP**

Kate: We've been talking about how we'd like to support configuring options on the Spin runtime. I put up a SKIP tracking thoughts/concerns and a proposal, plus alternatives. Still need to look at comments. We could discuss here or could just keep as a PSA.
James: Took a look, some details to settle around runwasi but overall like the approach

Kate: Summary of approach: using container environment variables. These go to the Spin runtime, as opposed to the Spin App. We're going to really need to explain this to users.
Joe: Question about spin.toml. For configuring wasm components, do we use variables in the spin.toml?
Kate: If you want to add a variable to a spin.toml, would need to re-push artifact. If you'd like to update or supply a variable value, you'd change in the SpinApp custom resource. Then, the Spin operator constructs the SPIN_VARIABLE_<var> env var to pass to the app environment. Not sure on what it looks like when an Azure Key Vault or Hashicorp Vault provider is used.
James: We may want to detail that
Kate: Yes, I can add that in a background section
Joe: In runwasi, we are planning to implement the sandboxer api from containerd. That would mean, in shared mode, we'd only have one shim runtime per node. What are the implications of this?
Kate: Ah, yes, I was thinking about this. Also moving from unshared to shared mode. Perhaps in shared mode we'd ignore some/all of the runtime config options (like, listening port). In shared mode, maybe we'd utilize containerd config for that? Ok, let's lay out a section about that.
Joe: Not urgent, we haven't yet implemented it but something on the horizon.

Melissa: Anything else?
Group: No :)

Wednesday, May 29, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us02web.zoom.us/j/82108121812?pwd=ZkhuSHExTXBPek9SaFVrTEszTDZPZz09
Meeting ID: 892 3054 4590
Passcode: 644621
Google Calendar Event

Recording:
https://us02web.zoom.us/rec/share/XqNdXxUfWppJ3ZsllODUP73kYm_Au9S6Xgwmru8dgMsnYDUkY1F_w8sE6cpH5Ul5.fmUuX8hJNo3QwTpT?startTime=1716994774000
Passcode: 4.wEvPbv

Attendees

- Melissa Klein (Fermyon)
- Caleb Schoepp (Fermyon)
- Adam Reese (Fermyon)
- David Justice (Microsoft)
- MacKenzie Adam (Fermyon)

- Jasmine Alindayu (Fermyon)
- Kate Goldenring (Fermyon)
- Jiaxiao (Joe) Zhou (Microsoft)
- James Sturtevant (Microsoft)
- Radu Matei (Fermyon)
- Kedaar Sridhar (Fermyon)
- Danielle Lancashire (Fermyon)

Agenda (please add [yourname] to items when adding them)

- Passing configuration to the containerd-shim-spin https://github.com/spinkube/containerd-shim-spin/issues/115#issuecomment-2111868496
- [Joe] proposes to create a “spinkube/azure” repo under the SpinKube org that provides the best end-to-end experience for managing spinkub on Azure Kubernetes Service. Similar to https://github.com/Mossaka/spinkube-aks
- [Rajat] proposes to add a subcommand `doctor` to the `spin kube` plugin. As discussed in one of the previous meetings, a formal SKIP has been created for this.
- [Rajat] Discuss volume mounts issue and OCI Loader (https://github.com/fermyon/spin/issues/2532)
-

Notes

- Passing config to the containerd-shim-spin
    - Kate: We talked about it in the Runwasi meeting. I.e. how do we pass config to the shim. Has tried passing config with CRI. James mentioned another user was having an issue on a version of containerd.
    - James: We shouldn’t need containerd 2.0.
    - Radu: What’s the expectation in other shim impls that take config?
    - David: There are some env vars like OTEL_TRACING that are usually passed along. Config options are on the protobuf message for task create. It is a protobuf any field. It is up to the consumer to unmarshal it.
    - James: We use it a lot in the windows shim. Worthwhile looking at the runc shim impl.
    - Joe: Did some debugging last week and found we can set key value vars to the config toml but I was not able to see them in containerd
    - Radu: If I understand correctly the options field is passed through the proto message.
    - David: Slightly more meta point: There are different categories or layers of configuration. Here’s what we want to configure and where do they belong.
    - Radu: It would suck to use containerd config which requires a containerd restart which some users don’t have perms to do.
    - Kate: We need to break down our personas

    - Caleb: Personas are key. Do we have a clear answer on how config is passed through and how we need to categorize how they're passed through right now?
    - Radu: +1, if it's system wide config, then containerd plugin makes sense. Executor level, or application level? To Caleb's point, that's a prereq to splitting what config belongs where.
    - Kate: When we say, "do we know what we want to do for config" we need to figure out config options pass in containerd that's been established. Then for levels of info (e.g., SpinApp), I'd be curious to see if we want to use pod level environment variables. Or if there are things we'd want to set in runtime config. There are lots of scenarios we'd want to hit.
- Proposal to create spinkube/azure repo
    - Joe: Msft and fermyon has had convos about this. I made a repo with a helm chart to simplify the process of deploying spinkube to aks. Going to add more things like the ability to integrate with azure application insights. I want to move this repo to the spinkube org itself. I propose spinkube/azure
    - James: Is there anything specific to aks in the helm chart? Or is it just generic
    - Joe: At the moment there is nothing specific about aks but there are things specific about azure
    - David: Helm might not let us mutate everything we want to. We might need to use some bicep templates. E.g. creating a pki, not using cert-manager, setting workload identities. I would imagine that each provider will have some weirdness that could be nicely segregated into this repo for native behavior. It'd be nice to set up a really good experience with azure insights.
    - Danielle: It would be nice to have a skip for this. Sets a precedent e.g. if amazon were to come along.
    - Radu: Following skip having maintainers and a release process would be good too. This is my assumption I would like made explicit.
    - Adam: Are we trying to break this into a skip about making repos and a skip for the azure repo?
    - Danielle: No just a single skip for making the new azure repo
    - Joe: I'm happy to do it
- Proposal to add subcommand doctor to the spinkube plugin
    - Radu: Makes pun about don't skip the skip
    - Joe: Is this related to the spinkube project itself?
    - Danielle: We want broader feedback b/c it interacts directly with the clusters. Directionally the plugin is in a weird spot b/c do we want it to just be a scaffolding tool or a catch all tool. We haven't decided that yet. We don't want to reimplement kubectl poorly.
    - Joe: For people to run the doctor they have to install spin on their host machine
    - Radu: Plugins are just executables so there is nothing stopping you from just running that exe
    - Joe: Just confirming we don't need to install spinkube plugin on the cluster
    - David: What if spinkube doctor could auto-create an issue

    - Danielle: Best things we did in cocoapods was have a ruby exception handler that auto formatted markdown for a github issue with all the details you needed
- Discuss volume mounts issue and oci loader
    - Radu: Context – this is four issues in a trench coat. The most visible of which is issue 108. It appears that we correctly create the volume mount. But something in the spin loader is broken. Mostly a loader issue in how it fetches content from oci and builds the wasm filesystem. The other issue Rajat pointed out is that we do lots of redundant copies when setting up the filesystem and when fetching from oci. I have a draft PR that drops it to one redundant copy. Still struggling to inject files from outside the oci artifact
    - Adam: I looked into this and view it differently. [Notetaker didn't follow]
    - Radu: Files from oci artifact are copied inside. The loader needs to correctly handle injecting things from volume mount
    - David: I'm confused. Trying to mount foo that exists in the artifact. Trying to volume mount something from the podspec. Volume mount foo at /foo.
    - Radu: Unsure. The way the spin oci loader handles fetching artifact from the content store conflicts with trying to inject the volume in the…
    - David: Is expectation that what you volume mount is going to live in the artifact?
    - Radu: That is part of the question. In both scenarios it fails.
    - Danielle: Does it fail for use in host components too?
    - Radu: Idk. Mostly guest use was tested.
    - David: Opportunity with shim to document how filesystems are layed out. We should specify which side wins in conflicts, or should the app not run if there is conflicting state.
    - Radu: Semi-strong opinion that we should let the volume mount win and overwrite stuff as is the precedent in k8s
    - Adam/Danielle: [Notetaker did not follow]
    - Danielle: Potential for what you have in a volume to be overwritten
    - David: We probably have enough control over this. With enough work we can figure this out. We need to match k8s
    - Danielle: Does an oci loader for spinkube look the same as a spin cli tool oci loader
    - Radu: Not necessarily, I agree with david on being explicit about behavior. At least document the existing behavior. No overwriting happening b/c of the spin loader as of now
    - Danielle: Does this mean we could deadlock a kernel b/c mounting something above the hierarchy of an existing mount can get weird
    - Radu: You never copy files directly inside the host at the expected guest path. You mount from garbage directory in host to guest
    - Danielle: If in the guest you've already mounted to /blah, then the loader mounts to /blah/blah. Doing this can break things in weird ways.
    - Radu: We're talking about different guests and hosts. I'm talking about wasm guest. Wasmtime handles the mounting using temp directories.

    - James: One other thing to check out is that we have to fs. Container fs which gets mounts from k8s. Then we have a wasm fs which needs to be explicitly told to see those file mounts. In the wasmtime shim we explicitly map in the root directory. If we don't do that in the containerd shim spin then we are doing it right. We might not always want to map in the root directory.
- Joe: I merged in newest release of containerd-wasm-shim which greatly improved the startup time.
- Caleb: Circling back to config on next steps to understand how I can engage. Is that working with Kate?
    - Kate: I'm happy with that next step. We get options passed if its CRI, run-wasi isn't doing anything with them so we'll need to do some design around that and pair
    - Radu: Are we also clear on how we'd pass config per application
    - Kate: No, we need design around that too. We need to think about runtime config, environment variables… need more though, maybe a SKIP would be appropriate to lay out all the considerations
    - Caleb: I'm more interested in the SpinApp and SpinApp Executor so coming up with a proposal
    - David: I'll wait for SKIP to chat through it, wasn't sure if the artifact config media layer would come into play. IT comes down to where in the processing chain these need to land. Based on the use-cases in the SKIP will inform where they need to land and what we need to do to land them.
- Radu: Throw back topic!!! do we not see volume mount in just host, or guest and host? Runtime config and certs are injected by Spin Operator into applications. They're making their way into the container, but not visible in the guest. So they're there…. Just not where we're looking
- Joe: Any updates to applying SpinKube to CNCF?
    - Danielle: Haven't heard anything yet… Getting closer to what we want to do with Spin Governance, and assume that will help that process greatly. Till was going to come talk about the plan a few weeks ago, but holiday clash
    - Radu: let's put this on the agenda for next time to make sure we're all on the same page
    - Danielle: Michelle may be a good presenter for that topic

CANCELED - Wednesday, May 15, 2024 8:00 am PT/11:00 am ET/5:00 pm CET
https://us02web.zoom.us/j/82108121812?pwd=ZkhuSHExTXBPek9SaFVrTEszTDZPZz09
Meeting ID: 892 3054 4590
Passcode: 644621

Attendees

Agenda (please add [yourname] to items when adding them)

Notes

SpinKube Project Meeting

Wednesday, May 8, 2024 8:00 am PT/11:00 am ET/5:00 pm CET
https://us02web.zoom.us/j/82108121812?pwd=ZkhuSHExTXBPek9SaFVrTEszTDZPZz09
Meeting ID: 892 3054 4590
Passcode: 644621

Recording:
https://us02web.zoom.us/rec/share/NYsZp6-kr9bHbJq9kiUo9AwvvGanHJlSt65Mmi7sIQD4okfwPF1fcnlewVcLRutz.rRJ4blty09ipH79f
Passcode: MM.=L17E

Attendees

- Radu Matei (Fermyon)
- Kate Goldenring (Fermyon)
- Joe Zhou (Microsoft)
- Caleb Schoepp (Fermyon)
- Rajat Jindal (Fermyon)
- Michelle Dhanani (Fermyon)
- David Justice (Microsoft)
- Matt Fisher (Fermyon)

Agenda (please add [yourname] to items when adding them)

- [Joe] The observability work and start up performance
    - Observability in Wasm OCI Runtimes
    - Performance: https://github.com/containerd/runwasi/pull/577, https://github.com/containers/youki/pull/2780
    -
- [Joe] Better logging across the stack. (e.g. 1, 2, 3)
- [Joe] benchmarking (1, 2)

- [Kate] Using released versions of trigger crates (`spin-sqs-trigger` and `spin-command-trigger`)
- [Kate] Spin Security Advisory and shim patch (1)
- [Kate] SpinKube performance testing (related to the above)

Notes

- Observability and startup performance
    - Joe: used OTel to find a performance issue in the shims (including the Spin shim).
    - Joe: demo - local changes in runwasi, Rust extensions, Youki, and a local instance of containerd; running a local app with a wasm container with an Jaeger endpoint which has all the tracing across containerd and wasmtime;
    - Joe: used this dashboard with traces to find a startup performance issue; already fixed locally (on the create function), which is now fixed; the root cause for it was not being able to delete the shim socket address before closing the shim, causing all shim processes to crash at the end; this was because youki has chdir functionality (youki#2780), changing the shim root dir;
    - Joe: there is PR that shows a performance improvement, from 1.7s to 0.4s. This PR is replaced by more robust changes in youki; hoping to merge this PR; waiting for a second review for the PR.
    - Radu: this would need to propagate through runwasi and the Spin shim.
    - Joe: correct.
    - Joe: to get tracing the same you would need PR runwasi#582. Draft, requires more work.
    - Radu: this will make debugging shim issues so much easier, thank you for working on this!
    - Joe: Google doc for the design of observability of Wasm OCI shims. There are several projects that support OTel (logs, metrics, traces); containerd and CRI-O both have OTel support; we need to add the ability to propagate context from containerd into the shim and back, so we can correlate across containerd and shims; for lower level runtimes, started the work in runwasi; youki has a tracking issue, but no work there yet. Rust-extensions has an issue for a more customizable logger based on tracing; then we have the shim; Spin has experimental OTeL support in 2.4; Wasmtime doesn't have plans to support it at the moment, Wasmedge has a plan, but no work yet; Wasmcloud has support for OTel since 1.0
    - Joe: we want to broaden adoption of OTeL across Wasm projects, with end to end observability, ideally from kubelet -> OCI runtimes -> shims -> and the Wasm runtimes; perf benchmarks are needed to understand the runtime performance effect, and memory usage; documentation is also needed for developers to leverage observability in debugging.

    - Radu: what is the dependency needed to get application observability in SpinKube?
    - Caleb: the thing preventing here is the conflict between init for subscribers; I'm focusing on fixing the conflict there, but we need to get them to play nicely, since both are trying to instantiate a global logger; Joe, I'd love to spend some time learning more about this.
    - Joe: sounds good; two parts: OTeL tracing in runwasi; Caleb did a lot of work adding logging to the SPIn shim; those are orthogonal, but belong to the same stack;
    - Radu: distinction between operator level observability and user application observability.
    - David: is there a plan to do that? It seems there are two exporters aimed at two collectors; is there a plan on how to collect those differently?
    - Caleb: we need to discuss this; it probably makes sense to have two different exporters, since you might not want to send them to the same collector; we'll have to discuss this in the issue tracking this.
    - Caleb: do we think having infra vs. app observability separate? Or should they be grouped?
    - David: I like the separation of the two; I don't want to see the infra when working on the app; as an app dev I might not know or care about containerd;
    - Radu: keeping them separate logically makes sense to me; app devs should not care about infra level traces; that being said ,I don't have an opinion about whether they should be separate exporters.
    - David: what would you do with a pod normally? How would you debug a pod workload, where would you put traces? It's been a theme for the shim to be consistent with all other KUbernetes workloads;
    - Joe: Caleb and I will have a chat offline about this.
    - Joe: does anyone know of adding tracing to Wasmtime?
    - Caleb: I was chatting to Alex Crichton, there is some tracing emitting at a debug level, we can chat about this later.
- Security advisory
    - Kate: this morning PT we released a critical security advisory for Spin; network requests could escape the sandbox under specific scenarios (link to advisory). We patched this and released v2.4.3; Will update the shim and should have a patch release for the shim in the next couple of hours. Thoughts about communicating the advisory in Spin?
    - Kate: we got lucky with the shim on this patch; originally I thought we weren't going to have v0.14 done yet; it was going to be difficult if we had to retroactively patch the external triggers we have; We should require versioned dependencies rather than just revisions;
    - Kate: once I'm done with the patch I'll create releases for the SQS and command triggers. We can have it as part of the release process to update triggers;
    - Joe: is there any reason for the SQS trigger to be not in the Spin repo?
    - Kate: experimental;

    - Radu: also it allows us to have external maintainers;
    - Kate: do we want to have release branches for the shim? We currently only have tags; having branches would be helpful to point to and have checkpoints. We can keep going with the tag process.
    - Joe: that would make sense; David has done work in runwasi to simplify the release process;
    - Kate: is there a write-up that we can reuse in the shim?
    - Joe: yes.
    - David: I think it's up to date; https://github.com/containerd/runwasi/blob/main/RELEASE.md
    - Michelle: those release branches will be handy if you have multiple patches we need to apply on specific branches.
    - Kate: is the other way just checking out the latest patch?
    - Michelle: that would be fine too.
    - David: tags are great for the start of a release; but when patching, a branch is nice;
    - Michelle: in Spin, we have several things going into the release;
    - Radu: I'd like to put on the agenda in the future discussing the policy for patching previous versions.
    - Michelle: don't want to lose track of what you're saying, Radu. What would be a good place to write this up?
    - Radu:
    - Joe: we don't have a signing process for release binaries yet; we should add them. Also, should we remove the pre-release tag?
    - Radu: I'd liket to have a broader conversation about this.
    - Matt: when you mark something as pre-release, when you fetch releases it doesn't come up in the API;
- Performance testing
    - Kate: we're interested in tracking the performance of SpinKube in general (density, performance, no. of requests). I'm working on a performance testing suite alongside Vaughn, working on a set of k6s tests, pushing results to DataDog (for now, but can be Prometheus). Wanted to mention we're working on this if others are interested.
    - Kate: showing demo of the dashboard showing performance.
    - Joe: this is extremely useful, well done!
    - Kate: my main concern: this is all visual; I have to toggle time stamps to see previous runs; but I don't know how to export this page so we can see this over time.
    - Caleb: Grafana cloud, hosted Grafana?
    - Kate: maybe using the paid version would fix this.
    - Caleb: if you're using your own instance, you can push into Prometheus.

SpinKube Project Meeting

Wednesday, April 17th, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us02web.zoom.us/j/82108121812?pwd=ZkhuSHExTXBPek9SaFVrTEszTDZPZz09
Meeting ID: 892 3054 4590
Passcode: 644621
Recording:
https://us02web.zoom.us/rec/share/SX8cQWNmVT_ySKomZHAy73HeFrVp1qcgQ8IFDrhoQjGdr-Qc-aRFFVOQ5fp4axTa.bl4hHXDUQeA3KJCt?startTime=1713366008000
Passcode: *Rd?#w7s

Attendees

- Matt Fisher (Fermyon)
- Michelle Noorali (Fermyon)
- Danielle Lancashire (Fermyon)
- Vaughn Dice (Fermyon)
- Rajat Jindal (Fermyon)
- Radu Matei (Fermyon)
- Jiaxiao Zhou (Microsoft Azure)
- David Justice (Microsoft Azure)
- Caleb Schoepp (Fermyon)

Agenda (please add [yourname] to items when adding them)

- Discuss “spin registry push” test approach (details here) [rajat]
- Demo “spinkube-doctor” and collect feedback. [rajat]
- Talk about observability across the stack (rust-ext, runwasi, spin shim, spin, wasmtime, etc) [joe]
- Sandbox application and Spin governance [danielle]
    - Context: https://github.com/fermyon/spin/issues/2449
    - Further context: https://github.com/cncf/sandbox/issues/90

Notes

- Melissa leads the meeting. First agenda item – spin registry push testing

- Rajat: started looking at test cases with `spin registry push`. Today we're only testing the shim with container images, and we should test with the Spin registry push command as well. In the makefile, there is a part of the test setup; and that's where we do the container image build; for those apps we need to configure the spin command to push. I read in the SpinKube channel that have an effort to improve the GH actions; should I hold on to the changes for now?
- Joe: no one is working on the Dagger CI right now.
- Rajat: ok, so I will continue the work here.
- Joe: you can write some bash scripts (or GH actions) to reuse some of that logic. All of my changes are in the runwasi repo, so they are independent.
- Radu: +1 on testing this
- Rajat: once we start building apps on CI, they became very slow; do you see concerns with using prebuilt wasm files in the spin apps?
- Radu: why not use pre-pushed apps?
- David: perhaps ttl-sh?
- Joe: +1 on what David said. https://github.com/spinkube/containerd-shim-spin/issues/81 CI is running `make build` and `make test`. But the cache is not reused.. `Make-test` should be able to use the cache. Haven't had time to investigate the issue yet.
- Rajat: clarifying one thing: I think Radu was mentioning that instead of checking the wasm files, we can pull them from something like ghcr. With that clarification, do you think it's ok to fetch them during CI?
- David: SGTM
- Radu: For the record, I don't really have a strong preference to either approach.
- David: also good to consider apps that have been around that may have stopped working (historically)
- Danielle: Using Finicky Whiskers as a canary might be helpful
- Rajat: we have several test cases we've been using for Spin in the past.

- SpinKube Doctor. Rajat: trying to run Spin apps on my personal cluster and that was failing (old containerd version). I thought it would be good to have a tool that inspects the cluster and tells you what might be wrong. Standalone tool that we can add as a subcommand to the spin kube plugin; it runs against the cluster and gives you a report / suggestions on what you should do for the cluster to be ready to run Spin apps.
- Rajat: we can also have it easily create a bug report for issues; or use it to check version checks.
- Rajat: independent tool, or subcommand for the kube plugin? What do you think?
- Radu: https://developer.fermyon.com/spin/v2/cli-reference#spin-doctor
- Radu: how do we keep this in sync with the docs?
- Rajat: re: compatibility – keeping them up to date with Spin operator; specifying a version of the operator, and keep templates in place.
- Matt Fisher: how is that file distributed to users? Do we package that with the binary, or do we fetch that from a repo?
- Rajat: we could have a default version that we ship as part of the binary, but we can have release artifacts for the plugin. Based on the version, we can fetch the file.

- Matt Fisher: an unknown – if there are updates to the infrastructure of the SpinKube project (1.0 -> 2.0), how would you envision this working?
- Rajat: I did consider the version number being configurable.
- Caleb: there are dependencies in the order in which these should be done. The order in which we display these matters; And general feedback – I think this is really useful, interesting work.
- Rajat: the checks in the file are orders; `autofix` vs. `check` is another idea.
- Michelle: I think this is great, really excited to use it. What would the workflow be if I wanted to add a custom check?
- Rajat: you need to implement a function in `checks.go`.
- Caleb: I kind of like the idea of this being a plugin like `spin kube doctor` similar to `spin doctor`
- Radu: I think this brings up the point of an improvement proposal.
- Danielle: It think that would be really good, in particular for cases where there are some alternative designs; so documenting those in an improvement proposal would be good.

- Observability. Joe: across the stack: rust extensions, runwasi, the Spin shim, Spin operator. I started the work here about a month ago. I made some progress there recently, and Caleb raised the issue on OTel tracing to Spin, and there were conflicts with the underlying work. The idea is to change the logs to tracing logs, and we think that would solve the issue. This conversation spawned the idea of talking about OTel across the entire stack. If we're changing to tracing, this would impact multiple repos. Caleb, anything you'd like to add here?
- Caleb: I think that's an accurate description.
- Joe: overview of the current stack. Rust-extensions is used by runwasi to use the protobuf for containerd. This is the first repo we need to update. Next is runwasi. I spent most time here adding tracing. Next is the Spin shim, where Caleb has been adding OTel here. It uses runwasi and the rust extensions library as a transitive dep. This is the stack where we need to add observability. Runwasi also uses youki, which already did the work to conver their logs into tracing logs. Last, containerd itself has some ways to collect the logs from the emitting shims from tracing logs.
- Joe: this morning I pushed a PR to runwasi to instrument some of the code there. Every public function has the instrument macro now.
- Radu: is this getting us closer to being able to get shim logs in the application logs?
- Caleb: me and Joe had a good discussion last week; to attempt to answer your question: converting everything to tracing would allow us to set up multiple layers, so we could send these both to stdout (so they're considered app logs) and to contaienrd. There is a conversation about whether we *do* want to allow shim logs to users.
- David: The delineation of infra vs app logs is the big thing imo
- Joe: everything that happens in the sandbox that youki creates gets in the pod logs.
- Caleb: the logging internals; if we switch everything to tracing, who would set the subscriber?
- Joe: the binary's main function (shim main).
- Joe: my work is on the span, Caleb on the logging.

- Discussion on how to more easily debug applications; consensus on an improvement proposal that would allow setting a variable that would send all shim logs to pod logs.
- Rajat: discussion on Kubernetes events.
- Joe: create an issue for it.
- David:

- Danielle: I wanted to get a bit of context and temperature check on the SpinKube application, how it's going, and more clarity on what people would like from Spin governance to feel comfortable with the project.
- David: quickly, what I put in the issue. I think it would be helpful for adoption as well. My perspective is biased from working for MSFT; we already provide a service called AKS WASI node pools. SpinKube is a much better option, and I love where it's going. I want to see it grow and bloom everywhere. Part of it is wanting to make sure are all pieces are under community governance. That's where my comment is coming from.
- Danielle: i think we all agree on that.
- Joe: my naive question: it would be great ofr Spin to be in the CNCF or BA.
-

SpinKube Project Meeting

Wednesday, April 3rd, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us02web.zoom.us/j/82108121812?pwd=ZkhuSHExTXBPek9SaFVrTEszTDZPZz09
Meeting ID: 892 3054 4590
Passcode: 644621
Recording:
https://us02web.zoom.us/rec/share/z9FX1bGUCx4hUOApLoUFGU1uIHv-P1KUdJLcl3DaG5sosmYWv07UMcscJbuD5mr-.lFOrtGUMJddXTLMa?startTime=1712156354000
Passcode: @Es72G*i

Attendees

- Danielle Lancashire (Fermyon)
- Radu Matei (Fermyon)
- Flavio Castelli (SUSE)
- James Sturtevant (Microsoft Azure)
- Matt Fisher (Fermyon)
- Cristoph Voigt (Liquid Reply)
- Jiaxiao Zhou (Microsoft Azure)
- Kate Goldenring (Fermyon)

- Rajat Jindal (Fermyon)
- Caleb Schoepp (Fermyon)
- MacKenzie Adam (Fermyon)
- Sven Pfenning (Liquid Reply)

Agenda

Welcome/Introductions

Discussions

- Project updates
- WASI init containers and sidecars
- Issues
    - https://github.com/spinkube/containerd-shim-spin/issues/57 (Talos Linux)
    - https://github.com/spinkube/containerd-shim-spin/issues/62 (self HTTP calls)
    - https://github.com/spinkube/spin-operator/pull/184 (injecting CA bundles)
- Q&A

Notes

- CNCF application: reviews on the 9th, not sure if SpinKube will be reviewed this time around.

SpinKube Project Kickoff Meeting

Friday, March 8, 2024 8:00 am PT/11:00 am ET/5:00 pm CET

https://us02web.zoom.us/j/89230544590?pwd=bktlR0o3Mk1nTjlaQ1d2enJjK2hVZz09
Meeting ID: 892 3054 4590
Passcode: 644621

Attendees

X Danielle Lancashire
X Caleb Schoepp
X Michelle Dhanani

X Kate Goldenring
X Radu Matei
Jiaxiao Zhou
James Sturtevant
X David Justice
X Flavio Castelli

X Christoph Voigt
X Sven Pfenning

X Melissa Klein

Agenda

Welcome/Introductions

Discussions

- Timeline
    - Make SpinKube org and its projects open to the public - March 8
    - PR about SpinKube - March 13
    - Apply to be a CNCF Sandbox project - March 20
    - KubeCon: Announce intent to contribute SpinKube to CNCF - March 21
- Governance
    - SpinKube Organization
        - Security doc
        - Org admins
        - Contributor Ladder

    - Individual Projects
- Regular meeting schedule for organization
    - Organization/Project(i
    - Community
    - Maintainer
- Regular release schedules and Roadmap
- Documentation
    - maintainers

CNCF Sandbox Application Requirements

Application Draft: CNCF Application Draft

- Contributing Guide
- Maintainer Files
- Project Champions (CNCF)
- Roadmap

Notes

- Cristoph: what is the criteria for making things public? The project is not yet ready to do what it is supposed to
- Danielle: I'm pro doing things in the open
- Christoph: then making it public sounds good to me.

- Melissa: timeline
- Melissa: governance — org vs. project level
- Danielle: multi-vendor open source is hard; I want to make sure we have clear governance from the beginning, so there is no single source of failure; figuring the logistics of GitHub administration (and not having a single admin); add yourself to the list if you want to volunteer as an admin.
- Danielle: contributor ladder and security policy
- Michelle: (https://github.com/spinkube/governance/pull/12) the security policy is up, please add feedback and comments; the proposal is: each project would accept security advisories and have an email address; policy is to respond within 24 hours. Once / If we are accepted as a CNCF project ,we can have CNCF mailing lists. For the time being, a single email address for the maintainers is the proposal. If we have people relying on our software in production, we'd like to have a private mailing list where we'd discose information with our users before making it public; The policy is mostly adapted from containerd, because it looked great.
- Danielle: we have some credentials we need to share between maintainers: Twitter ,Google, Mastodon; we should set up;

- Radu: 1Password is the suggestion?
- Danielle: unless someone has a better suggestion; plus, it'll be free when we join the CNCF
- Caleb: contributor ladder; originally we didn't scope the roles to repos, but after feedback it does make more sense to do that; in terms of governance, we want to keep it simple; maintainers have a vote in governance; we can add more complexity later;
- Danielle: projects own their own versioing and release; I don't think we need to sync that;
- Radu: codeowners file – some repos have one, we should go over the repos and make sure they're accurateCNF
    - TODO: codeowners file for runtime-class-manager

- Melissa: meetings; in the beginning, I'd propose we meet every 2 weeks, since it feels like a good cadence to get things runnuing; we can then drop later if needed;
- David: are these going to be public meetings?
- Melissa: there are 3 types of topics;
- Danielle: I'd like to make these public after this one, unless we have a reason to have a private;
- Michelle: community or maintainers?
- Danielle: both;
- David: if we need private meetings, perhapd ad-hoc as we need them ,as opposed to standing meetings; My preference would be to handle as much as possible in public;
- <general agreement>
- Melissa: kick it off with a maintainer meeting open to public;
- Michelle: the week after KubeCon do no-meeting week; not sure if we want to follow that patter, so we'd do the week after;
- Cristoph, Danielle: vacation the week after KubeCon;
- Melissa: any other meetins?
- Radu: any reason not to record?
- <no> we will add a recording
- Kate: where will we host these? Not using CNCF hosting yet; where are we Slacking with each other?
- Michelle: can we get a Slack channel unless we're a CNCF project?
- Danielle: because we don't want to publicize until next week, we'll create one then.

- Melissa: regular releases, roadmap; looking at other CNCF applications, this does not have to be too extensive; but we need a roadmap showing the future of the proejct; A project board?
- Michelle: any way to add things in a milestone to a project board?
- Melissa: yes, I think there is;
- Radu: together with the project board, I am suggesting we have a paragraph for each project describing what the project will look like for the next few months;
- Melissa: that is actually part of the application;
- Danielle: https://github.com/orgs/spinkube/projects/3

- Radu: do we enable GH Discussions and use these?
- David: does anyone like discussions in GH?
- Danielle: I never know they exist
- Cristoph: not opposed to letting it open and seeing if community members interact with us?
- Caleb: I'd opt to just use discussions and see how it goes
- Michelle: I've seen a roadmap repos in other projects btw. Feels too early for us now but just throwing that out there. We could have a repo with just a README and that might be an entry point for folks who don't know where to submit an issue.
- Kate: https://github.com/orgs/Green-Software-Foundation/discussions
- Danielle: this means my vote would be no for now;
- Caleb: what is the norm in CNCF ?
- Danielle: Slack and GH issues;
- Conclusion: 4 no, 1 yes, 3 abstained
- Melissa: no GH discussions now ,we can turn them on later;
- Melissa: explanation of the roadmap somewhere;
- Michelle: roadmap repo, with a description would be great;
- Kate: docs repo?
- <ok using the docs repo for roadmap>

- Melissa: docs repo is out; volunteers to maintain?
- Danielle
- Radu: it would be good to have someone from each project maintain the docs
- Danielle: mainteiners team bing part of the codeowners for those files;
- <general agreement to have that>
- Caleb: not explicit maintainers, or in addition?
- Danielle: in addition
- Docs are live! https://www.spinkube.dev/

- Melissa: CNCF application; link above to take a lok; what we need: contributors guide (SpinKube, or project level?)
- Danielle: we have one for the operator; I assume it's good enough to open
- Michelle: volunteers to write the
- Danielle: current plan to file this during the keynote; more fun; contact email, one for each company? IP attribution;
- Michelle: spinkube maintainers email? Or just one person?
- Danielle: all recent had individual
- Melissa: maintainers.md files? SpinKube level or individual projects
- Radu: .github repo that shows all thigns across repos
- Melissa: we can use that repo
- Danielle: anyone opposed to renaming the governance repo?
- Michelle: i like the governance rrepo..
- Danielle: anyone wants to volunteer moving what should be in the .github repo?

- Caleb: I'll do it;

TODO: update the docs with a link to the CNCF slack channel
TODO: Caleb to move files to .github