Runwasi + Containerd-Wasm-Shims

📆 **Fortnightly**, Tuesdays at 9:00am PT

Join the Zoom Meeting

Past Meetings

Runwasi Group Slack

Runwasi Repo

Containerd-Wasm-Shims Repo

Active Runwasi Maintainers and Reviewers

Name	**GitHub**	**Slack**	**E-mail**	**Company**
Brain Goff	@cpuguy83	cpuguy83	brgoff@microsoft.com	Microsoft
Danilo Chiarlone	@danbugs	Dan Chiarlone	dchiarlone@microsoft.com	Microsoft
David Justice	@devigned	justice	david.justice@microsoft.com	Microsoft
Jiaxiao (Joe) Zhou	@mossaka	Mossaka	jiazho@microsoft.com	Microsoft
Ismo Puustinen	@ipuustin	Ismo Puustinen	ismo.puustinen@intel.com	Intel
Djordje Lukic	@rumpl	Djordje Lukic	djordje.lukic@docker.com	Docker
James Sturtevant	@jsturtevant	jsturtevant	jstur@microsoft.com	Microsoft
Jorge Prendes	@jprendes	jprendes	jorge.prendes@gmail.com	Microsoft
Toru Komatsu	@utam0k	Toru Komatsu	k0ma@utam0k.jp	Preferred Networks
Tomasz Andrzejak	@andreiltd	Tomasz Andrzejak	andreiltd@gmail.com	Microsoft

How This Meeting Works

- This document is open for comment. We encourage suggestions for upcoming meeting topics. If there are none 24 hours before the meeting, we may propose canceling on our slack channel and mailing list.
- The meeting is recorded and posted to our Youtube Playlist before the end of the week. A host moderates the meeting, and we encourage hand-raising instead of interrupting.

Instructions for facilitators

- Get the host key from one of the maintainers
- Wait a few minutes for people to arrive (9:02 PT)
- The meeting will start recording automatically
- Introduce the meeting "This is the Runwasi + Containerd-Wasm-Shims Communicate Call of [date]." Notify participants that we are abiding by the CNCF Code of Conduct.
- Ask participants to use the "raise hands" feature, and keep the Participants window open (or docked against the zoom window).
- Share your screen with the agenda, so it's available for the recording.

Agenda

[Template] Month Day, Year (recording)

Host: TBD

Note Taker: TBD

Attendees:

- Comment your names here

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]

April 22nd, 2025 (recording)

Host: Joe Zhou
Note Taker: TBD
Attendees:

- Joe Zhou (MSFT)
- Jorge Prendes (MSFT)
- Kate Goldenring (Fermyon)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Announcing v1.0 Release 👏
    - At kubecon we announced 1.0 prerelease
    - We introduced and tested all the breaking changes
    - We now have a stable API, with minimal surface, and won't have breaking changes any time soon
    - Split the instance trait to another trait called shimkit, where we can continue introducing breaking changes
- Planning for post-v1.0 tasks
    - Priorities
        - [Joe] Telemetry in the container process
            - Support metrics and logging to otel
        - [Jorge] Benchmark:
            - Current: we are benchmarking how many workloads we can run in one pod
            - For SpinKube, it's mostly one pod, one container. The startup time is more important than how many containers we can run inside a pod. We need to measure the time that takes it from the pod creation to the start of the container.
        - [Jorge] Use async for the main shim (https://github.com/containerd/runwasi/issues/801)
            - ttrpc server we are running is still using sync, should help resource consumptions of the shim.
        - MacOS support
            - https://github.com/darwin-containers/homebrew-formula
            - [Joe] https://github.com/lima-vm/lima

        - A well-defined issue for external contributors
    - [Joe] A landing page for https://runwasi.dev/
    - “Scale to zero”
        - https://knative.dev/docs/
        - [Kate] Tutorials on using Keda and Knative on spinkube.dev
        - Should spinkube install keda?
    - Sandbox API (https://github.com/containerd/runwasi/issues/385)
    - Windows support
        - support for a minimal version of the shim
        - A well-defined issue for external contributors
- Directions

March 25th, 2025 (recording)

Host: Joe Zhou
Note Taker: TBD
Attendees:

- Joe Zhou (Microsoft)
- Jorge Prendes (Microsoft)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Jorge: Changes to the Engine trait for 1.0
    - The Engine trait is now renamed to the Shim trait.
    - The Shim trait encapsulate two different processes: the shim process and the container process
    - The idea is to split the behaviour running in the container process and that in the shim
        - A “Sandbox” trait
        - A “Shim” trait
    - Another trait to split is precompilation - related to the Engine trait. It is optional and has two functions. The `Shim` trait will expose a `compiler()` method.
    - The main disadvantage is that before you only need to define one struct for your engine, but now you need to define 2-3 structs to implement 2-3 traits. It makes

        adopters write more code - but this change is more clear and it might be difficult to do after 1.0.
    - Previously, you had to use LazyLock to implement the precompiler but now you don't need it because we take care of that in the shim.
    - Action for Joe: to implement the Spin Shim on this PR to do a sanity test.
    - Follow up: simplify the `shim_main` function signature. The `name()` function and the name argument can be merged, the same goes to the "v1" argument to the `shim_main()`.
- Jorge: What's left before 1.0?
    - https://github.com/containerd/runwasi/issues/934
    -

March 11, 2025 (recording)

Host: Joe Zhou
Note Taker: Jorge Prendes
Attendees:

- Jorge Prendes (MSFT)
- Joe Zhou (MSFT)
- Brian Goff (MSFT)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Joe: Runwasi v0.10.0 was released! Let's celebrate it and talk about its new features, changes and what do we plan for the upcoming stable release.
    - Release of the core crate only. Shim binaries are only released on demand. Beneficial to do scheduled release for the core crate.
    - Some breaking changes, and new features
    - Support for systemd cgroups, which fixes a long standing bug. Now you can see pod level metrics.
    - Added structured logging macros to the crate, which log
- Jorge: What's left before 1.0?

Feb 25, 2025 (cancelled)

Host: TBD
Note Taker: TBD
Attendees:

Feb 11th, 2025 (recording)

Host: Jorge Prendes
Note Taker: Joe Zhou
Attendees:

- Jorge Prendes (MSFT)
- Joe Zhou (MSFT)
- Kate Goldenring (Fermyon)
- David Justice (MSFT)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Kate: Discuss next steps for pod level metrics issue: https://github.com/containerd/runwasi/issues/821#issuecomment-2649544315
    - Recent progress in the issue thanks to input from youki contributors. Youki has a setting to whether to use systemd cgroups, and we are setting that to false.
    - Jorge: youki is a runc replacement, there is a runc shim that calls runc binary which has a CLI interface.
    - Jorge: Youki has a command line arguments to configure systemd for cgroups
    - Jorge: if this is standard, we should use the same mechanism. If this is not standard, add a heuristic to decide if we should enable systemd
    - David: Will this be a config in the shim?
    - David: default to whatever is working instead of a heuristic - if we are missing information this is the best to have in the first place. If not, default to systemd and allow someone to rewrite that.
    - Kate: Need to check k3d systemd setting
    - **Action**: Quick check on whether there is a correct way of doing it, and flip it to be true for the time being. Follow up with a release
- Joe: Adding outbound HTTP tests to the wasmtime shim
    - User asking in Slack about outbound http connection, like -Shttp flag in wasmtime.
    - Kate: Should we add a test for something we don't support? A test that always fails?
    - Joe: I think we should support it.
    - Kate: I think it's a door we shouldn't add by default.
    - Joe: We could have a config for that
    - Kate: Are you referring to wasmtime config file?
    - Jorge: the issue is that the image might be locked into a specific wasm runtime.
    - David: having it as part of the workload assuming that the workload has authority to see what is being used and what is not being used. "No I don't want you to be able to use this"

    - Jorge: two tier access - images say they want to have access to something and the pod config says the images are not allowed to access to something.
    - David: shouldn't the runtime reject an interface that it doesn't support?
    - **Action**: make sure outbound http works across the shim
    - **Action**: create an issue to discuss possible configuration on app capabilities.
    - David: Does this apply to the wasmtime shim only?
    - Jorge: Yes, because all the other runtimes do not support component yet so it's out of scope.
- Joe: Issues triage

Jan 28th, 2025 (recording)

Host: Jorge Prendes
Note Taker: Joe Zhou
Attendees:

- Joe Zhou (Microsoft)
- Jorge Prendes (Microsoft)
- Kate Goldenring (Fermyon)
- Brian Hardock (Fermyon)
- James Sturtevant (Microsoft)
- David Justice (Microsoft)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Joe: containerd new release cadence and talk about runwasi's release cadence post-1.0
    - Recent change in c8d, time based release, every 6 months, pending review.
    - https://github.com/containerd/containerd/pull/11294
    - Once we reach runwasi 1.0 (targeting end of March), should we follow 6 month cadence of c8d?
    - Runwasi is a smaller project, and we move faster than c8d, so maybe a faster cadence.
    - Kate: What is the benefit of syncing with c8d?
    - Joe: in case of breaking changes of c8d.
    - Kate: prefer more frequent releases for runwasi as a consumer of runwasi.
    - Joe: I agree

    - Jorge: I like the idea of time based releases. 6 month is too long for runwasi as we are less mature than containerd. Perhaps a new version every month, and we can do ad-hoc things in between.
    - **Action:** Joe to document the release cadence in the repo.
- Brian H: General runwasi questions
    - Related to the DAG PR
    - https://github.com/containerd/runwasi/pull/792
    - Roadblock on proposed solution labeling parents with links to the childs, need to access the original layer in addition to precompiled layers for spin.
    - Have a potential solution, to be discussed with spin maintainers, would leave runwasi untouched. Idea is to enable Spin producing OCI artifacts with composed components, removing component dependencies https://github.com/fermyon/spin/issues/2988
    - Another solution: Spin shim to bypass precompile. To receive a c8d client, and handle precompilation itself.
    - How locked are we in storing the precompilation in c8d?
    - Joe: trying to understand issues and solutions. Can you elaborate more on the issue roadblock?
    - Brian: runwasi gets the image from the image and pass them to the precompile method. Shim composes the components while tracking the layers used for composition. Returns two new layers X(lock spin app) and Y(c wasm) with links to the parents.
    - Kate: go back to something else
    - James: images mutable is the problem. We place the digest at the lower layer than the image layer. People are gonna push precompiled apps to the registry due to speed optimization, and this might be an answer. Brian initially wrote a his own content store and James changed it to containerd's content store.
    - James: intuitively it seems wrong for shim to manage all these
    - James: We can come up with a scheme for labeling.
    - Kate: this is a question that can be fundamental which we can discuss in wasm-wg. Do we expect the OCI artifact to compose components?
    - James: we designed it to have multiple components in the artifact. Instead of doing this in the shim layer, there should be a plugin in containerd so that containerd can precompile and hand off the layers to the shim. Need to go to containerd team to propose the whole plugin system
    - David: do we mean all the interfaces are satisfied in composing components?
    - Kate: enough for the runtime to handle
    - David: how do we allow runtime to say "we can support this"
    - David: we don't want to compose multiple OCI artifacts
    - David: try not to be prescriptive and see what solutions what people build.
    - James: spins interesting take on this is that we have not just one app but many different apps in the same artifact
    - Brian: One of the downsides for composed components is dependencies like JS runtime.

  - Brian: as a short term solution to this for apps that have dependencies - we can go this route and move towards a goal to support universally in runwasi.
  - Brain: secondary to this solution is the precompile function exposed by runwasi is getting complicated
  - Jorge:
  - Jorge: the consensus is that we do want to handle this in runwasi in some way but we don't know how
  - Brian: There is no labeling scheme out there to enable this
  - Jorge: **action** is to understand constraints we have. The image is mutable. What does that imply? Try to come up with a strategy to work around this issue.
  - Kate: share the documents in containerd that are relevant to the discussion would be helpful for the future
  - Jorge: diagrams of communication for storing content between runwasi and containerd.
  - Brian: we need to extend GC awareness into the shim
  - Kate: Does this affect 1.0 roadmap?
  - Joe: Yes
  - Kate: Can we also try to get the pod cpu and mem issue resolved before 1.0 since it may affect cgroup paths/naming
  - Joe: yes that's in our roadmap
- Jorge: Issues triage
    -

Jan 14th, 2025 (Recording)

Host: mossaka
Note Taker: James Sturtevant
Attendees:

- Joe Zhou (Microsoft)
- Jorge Prendes (Microsoft)
- Brian Goff (Microsoft)
- James Sturtevant (Microsoft)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Joe: go through the project's dashboard and set up a goal for the major release
    - Want to set up a 1.0 version, need stable public api, docs, benchmarks, guidance usage, contributions

        - We have dashboard, but haven't followup all that much
        - https://github.com/orgs/containerd/projects/8
        - We have a few 1.0 required, and a few other important but no absolutely needed
        - Propose we meet regularly to go over the dashboard
        - Brian: meetings like this is perfect for this type of thing, maybe don't need a second meeting
        - James: Having a regular time to review is important otherwise it doesn't matter
        - Jorge: Agree, lets use this time period
        - Agreement to start using this missing to regularly going throught it.
        - Stable API:
            - We haven't really had a major
            - Joe is going kubecon that would be a nice time
            - James: we want to make sure we give enough time for Jorge's work and precompilation API changes.
            - Jorge:
                - Wants to introduce a few changes,
                - There are two public apis: instance and engine api
                    - We should reduce the scope of this
                    - Propose instance api isn't public since its not being used
            - Joe: Hyperlight is using instance api since it isn't tied to youki
            - Jorge: Do we see hyperlight as a consumer or a different back end such as Windows or linux? For instance swapping hyperlight with youki
            - Joe: it doesn't really need the youki isolation since it provides it only
            - Jorge: Can it be a backend so it can use private api's
            - Brian: could we version api's separately in a different crate?
            - Jorge: What's the advantage of the instance api?
                - Is there anything specific?
            - Maybe a little but not much.
            - Brian: we could work it in runwasi and upstream
            - Joe: If we can split it out since it is a crate, its valuable since we have someone working on it and using it.
            - Jorge: in the Quasar repo, they made it more generic. Which is kind of nice and also maybe not, we should look at it and try to learn
        - James: Are we still pursing Sandbox API?
            - It is an implementation api and not something that will affect the public api
        - Jorge: Moving things to async, we are seeing more and more of async leaking in
            - Its a breaking change so should be 1.0
        - Given all this is the timeline for kubecon still appropriate?
        - James: we should give some time for SpinKube to adopt the changes. Seeing stable CI for a few weeks should be expected.
        - Jorge: let's try to make it happen, but not force it
    - Joe: discussion on improving issue tracking and prioritization
        - How do we make it effective?
        - Jorge: making sure we have a dedicated time

    - James: keep it lightweight
- Jorge: Splitting Precompilation from Engine / runtime.
    - https://github.com/containerd/runwasi/issues/799
    - Pre-compilation Happens on lifetime of the shim,
    - This is relevant since we have to different processes, we won't be cloning the engine. This means engine we run is not the same as the one running it, which is a current assumption
    - We create the engine and then clone it which causes a bit of confusion
    - Prose we split trait into two: one for engine runtime, and one for prepossoing
    - James: How do we insure the config of the engine matches across these two
    - Jorge: Probably not a big issue, via testing
    - James: generally agree, wondering if there are subtle bugs we might run into
    - Jorge: there are some settings that aren't needed today, shouldn't be really be an issue
    - James: sounds good!
    - Joe: What is the issue and advantage?
    - Jorge: It breaks the lifetimes, and as an implementer you are encourage to not try to share too much
    - Jorge: Also ask the contribor that is working on the new changes if there are changes that would make it more usable in practice
- Joe: discussion on deprecating the containerd-wasm-shims repo
    - There is a slit between this repo and spin kube, last work done was a 7 months ago
    - There are at least two shims that are no longer active
    - Not sure about lunatic, but this hasn't change in several months
    - Jorge: Lunatic has no actively for 10 months
    - James: lets through an issue on the repo and see if anyone speaks up.
    - Jorge: do we archive it?
    - Yes!

Dec 17rd, 2024 (Recording)

Host: David Justice
Note Taker: TBD
Attendees:

- David Justice
- Jorge Prendes

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- (Jorge) Zygote process update

    - https://github.com/containerd/runwasi/issues/357
    - https://github.com/containerd/runwasi/issues/755
    - https://github.com/jprendes/zygote
    - Notes:
        - Youki is running many things a single thread.
        - New repo is the https://github.com/jprendes/zygote which demonstrates the starting of a process and using to be the basis of fork early on to not establish a significant amount of state.
        - The change in Youki to make sibling processes rather than children has landed, but has not been released yet.
        - Next steps will be updating Youki, merging zygote changes to update runwasi, and working with the maintainers.

Dec 3rd, 2024 (Recording)

Host: Jorge Prendes
Note Taker: TBD
Attendees:

- James Sturtevant
- Jorge Prendes
- Tomasz Andrzejak

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Jorge: Issues with signals
    - Graceful shutdown isn't working
    - Route cause is the Undefined Behaviour since we clone with no fork
    - Tokio signal feature, uses self pipe trick, when the signal fires it write a byte in to the pipe as signal and then handles them all
    - When we clone to the process we have more than one process listening to the signal
    - Tests only spawn one container so we don't see it, opened new pr to add test that demonstrates failure.
    - Not sure solution, tried one possible where create a pipe for other processes but this isn't currently working
    - Use a different signal handling solution?
    - (james) Does the async solution help?
    - (jorge) No, in fact we might have this problem even more
    - (James) other async runtimes?
    - (Jorge) Maybe, but containerd and
    - (Tomasz) Maybe bring up with Tokio discussions board?

    - (Jorge) sure could bring use case,
    - (Tomasz) Could we do something with the signal crate and clear the registry manually?
    - (Jorge) maybe? Some issues with the
    - (James) Is there another way to use clone without exec? Should we figure out how to do Exec?
    - (Jorge) main problem is we don't have an executable to call.
    - Trying a few different things here but haven't figured it out yet.
    - (Tomasz) Do we have the same issue with Musl?
    - (Jorge) Main take away, there is a PR up that demonstrates this and we should try to pass this test.
- Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]

Nov 19th, 2024 (recording)

Host: Joe Zhou
Note Taker: Jorge Prendes
Attendees:

- Jorge Prendes (MSFT)
- Joe Zhou (MSFT)
- James Sturtevant (MSFT)
- Tomasz Andrzejak (MSFT)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- New reviewer, Tomasz!
- Jorge: update on stress-testing (now in a PR! (#734))
    - Presented stress-testing in the last community meeting and is looking for a review. It is not currently running in the CI. There is an issue (#615) about running many containers in parallel in order to benchmark the memory and CPU usage.
- Jorge: fixed in CI's exit code
    - Wasmtime test was failing to compile and Tomasz helped to debug this issue. The CI was not failing on the tests and was not detecting it. We use cross to build the tests and execute the binaries outside of the cross container environment due to the need for admin permissions and syscalls. The Makefile is now using bash with pipe failure.
    - James: it was also failing on my dev machine
    - Joe: there was an open issue regarding returning status code 0 even if the test failed

- Tomasz: bumping the Wasmer version
    - Wasmer is depending on an older version of a crate. It has a very strict version of the wat crate. Tomasz relaxed the version requirement and helped us to move to the new rust version
    - James: this is a 2rd or the 3rd time we stuck at the wat crate, isn't this only used in testing?
    - Tomasz: both Wasmtime and WasmEdge depend on that wat crate. Maybe we can propose to relax the version?
    - Jorge: consider splitting the shims to their own workspace?
    - Joe: anyone is maintaining the wasmer shim?
    - Joe: should we trigger the wasmer shim deprecation according to the document in the repo?
    - Jorge: Wasmer is now depending on git dependency. The wamr also didn't publish to the crates.io. We may want to switch back to the published version if available.
    - Tomasz: Wasmer is depending on a commit.
    - Joe: we can create an issue to track that
- Tomasz: http/proxy in shim vs wasmtime serve
    - Comparing the WASI http impl in runwasi and wasmtime serve command. We used wasmtime serve as a reference for out implementation.
    - However, wasmtime serve is 10x faster than our implementation.
    - Benchmarked using hey, using 10 parallel workers. Could this be that we are not running the tasks in parallel? Even if we reduce the number of concurrent workers to 1, the wasmtime is still 3x faster, so maybe not.
    - Tomorrow planing to hack with Jorge on runwasi to run outside a container to rule out issues related to the shim being inside a container.
    - https://github.com/containerd/runwasi/issues/729
- Tomasz: documentation fixes and added documentation tests
    - Added step in CI to check for typos, and also run rust doc tests.
- Jorge: async shim, shimkit and trapeze
    - As part of the stress testing the shim frequently failed. When we try to create a container, we clone the process with only the calling thread and all other threads disappear. We are running the full Wasm runtime in that process.
    - A few things we can do:
        - run Wasm runtime in-process. No clone
        - Shim process is single threaded
    - In the future: bring Wasm runtime to in-process due to concurrency and performance issues to be able to run more containers in the same pod. It has implications on security and compliant on OCI spec. The better short term solution is to have the shim running in the same process and the approach is to make it async.
    - Jorge prototyped the async implementation.
    - Jorge: [Sharing screen]

    - Jorge: the issue with containerd-shim is that even if they use async, they still spawn new processes, and there is no support for Windows
    - Jorge: shimkit fully asynchronous, based on trapeze, an implementation of ttRPC with Windows support. It is narrowed scoped: no logging and collecting metrics.
    - James: we've discussed two issues - async and shimkit. No strong opinion regarding the async part, but what's the story with shimkit, do we want to donate it to the CNCF?
    - Jorge: expressed interests to have trapeze in ttrpc-rust.
    - James: My concern is that we are taking some extra stuff. I intend to want to contribute back to the upstream and am wondering what's the tradeoff here in your perspective.
    - Jorge: It felt like fixing the ttrpc-rust is a rewrite from scratch. Happy to donate trapeze to CNCF.
    - Brian: can we upstream trapeze to ttrpc-rust as 2.0? I know rust-extension is hard to modify.
    - James: I don't think they are at 1.0 yet
    - James: there is no tie with sync and async APIs in rust-extension. There is a room to cut off sync implementation completely if we move to async.
    - James: happy to introduce the folks there in ttrpc-rust
    - Joe: what's the next steps?
    - Jorge: starting with ttrpc-rust crate to propose that ttrpc crate to drop the sync API and focuses on the async API. protobuf crate to prost.
    - James: Windows is still on the table but a low priority, not a blocking to adopt async APIs.
    -
- Joe: release of the Runwasi shims
- James:
    - https://github.com/containerd/runwasi/pull/723#discussion_r1844564852
- [if we still have time] Joe: Review "In Progress" tasks for Runwasi 1.0
- Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]

Nov 5th, 2024 (recording)

Host: Joe Zhou
Note Taker: David Justice
Attendees:

- Comment your names here

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Joe: wamr shim
    - Notes;
        - The first implementation of Wamr was a created a few months ago. I now have a working example of the Wamr shim.
        - Todo: need to expose exit codes from Wamr SDK
        - (Jorge) We do have the test for exit code. Those are nice to have. I believe we should still be able to merge things without having the exit code passing. We just can't get the specific code.
        - (Joe) I would like to wait till we have the exit code. It is in the C API, but not exposed to Rust.
        - (Jorge) If it's in the pipeline already, it makes sense.
- Jorge: stress test on runwasi
    - Notes
        - (Jorge) Recently there's been a PR (https://github.com/containerd/runwasi/issues/704) comparing Runwasi with native containers. The premise is that a Runwasi wasm container is not faster than a native container.
        - I've created some stress tests to measure startup time.
        - https://github.com/jprendes/runwasi/tree/stress-test
        - (Jorge) </sharing screen> I created a new crate called stress-test. It takes the place of containerd. We talk ttrpc to the shim with all the things we need. The shim publishes event to containerd and we needed to replicate that behavior. Once started, the shim can be connected to over ttrpc.
        - You can think of a task as a process inside of container. With runwasi you can think of a task as a Wasm run.
        - When you create a task, it builds the structure, but start is when the task is started… additional lifecycle events…
        - Was hitting issues when running task concurrently. We don't often start many tasks together, like 2000 tasks.
        - Jorge runs with 10 tasks, and it works, then in parallel and after a few tries as parallel, it eventually deadlocks.
        - We are experiencing undefined behavior when accessing a mutex across threads in forked processes.
        - I think this is interesting b/c this is where the UB issue becomes real rather than just theoretical. We can observe these behaviors and fix them.
        - When we run them serially, usually it works.
        - When I time those containers, I can see that it took about 16ms per container. That is far off what we saw in the bug report. I also run it with a 1000 containers so more of the setup cost is amortized. I can get down to about 10ms on this machine.
        - The task I create is not very realistic. There is no setup for networking. If there is cost because of setup, then we are not measusing it.

- Also, we are not using containerd, so there might be slowness in the interaction. In this case, not sure what to do. Perhaps, discuss with upstream containerd folks.
- (Joe) Are you using libcontainer in here?
- (Jorge) I'm using the shim we create.
- (James) Is it the call to log that's causing it?
- (Jorge) Sometimes instead of lock, it will get an error from the shim trying to bind to a socket, getting the current working directory, sometimes it hangs due to the mutex. And there are some that I hook the debugger to it.
- (James) Perhaps, we should not use the fifo logger. Once we fork, we should write to the pipe containerd gives us directly. I'm pretty sure we are doing something weird with logging.
- (Jorge) Shows the process is waiting on a mutex.
- (James) We also use the Rust logger in that forked process. If we were to remove the extra logging there, we may avoid some deadlocks.
- (Jorge) This example is being called from libcgroup and is used for tracing. This is all before the Wasm runtime has been called. There are things happening between when we fork and when we are in the forked side.
- We do have an issue with passing the logger to the Wasm runtime where we'd get crainlift output in the logger.
- If we were to run each module in its own process, we could avoid the UB.
- (Joe) We had discussed using lower level apis rather than using container builder.
- (Jorge) There is also the question of how much performance benefit do we get from running outside a container. If we do that we lose some defense in depth. If there was significant perf impact, I could definitely see a justification for doing it. The other solution would be to make the shim process single threaded using async.
- (Joe) I think besides perf, I think it's more important to deal with the UB demonstrated by the stress test. Perhaps, we could remove the logging and see if we still hit this.
- (James) I'm wondering if the runc impl has the same issue. I'm wondering how they avoid the situation.
- (Jorge) For each execution they exec runc. However, it does pay the cost of a new process each time.
- (James) b/c they delegate to runc, they don't have the same concurrency issues?
- (Jorge) https://github.com/youki-dev/youki/pull/2961
- Next step: schedule a meeting with Brian to deep dive into this issue

Oct 22nd, 2024 (recording)

Host: Joe Zhou
Note Taker: Kate Goldenring
Attendees:

- Comment your names here
- Kate Goldenring

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- [Tomasz] alternatives for fixing precompilation of components https://github.com/containerd/runwasi/pull/700
    - Working on WASI HTTP the past week and encountered an error when trying to run any Wasm component. Error was related to using Wasmtime compile module instead of compile component. Pushed a quick fix. But brought up that we may need more refactoring before applying the fix. Not sure what the best next step is. (PR #700).
    - Joe: James left a comment about how the spin shim has non wasm layers.
    - James: I think this change is fine just as it is. It is a simple API for a small subcase, so i lean towards letting it be as it is
    - Jorge: Alternative would be to have a free function that takes bytes and tells you if it is a component or module
    - James: Isn't that what WasmBinaryType::from_bytes does?
    - Jorge: Doesn't work with WAT
    - Jorge: When Wasmtime fails to precompile, we log and error - strange because it is not fatal. Maybe you can change that to a warning.
    - Tomasz: I can downgrade the error
    - Jorge: So we move forward as is?
    - [consensus]
- [Jorge] Sandbox API
    - I've been trying to get the sandbox API working but there is not much docs on it and that there are 2 aPIs: sandbox API and sandboxer API, which are related. We have to build 2 binaries, one for sandboxer API and one for the shim -- but realized it is something specific to Kuasar. Turns out can create one shim with the Sandbox API. Can implement on the same server the Sandbox API. Today we do `Task/v2` -- now it is `Sandbox/v2`. Sand API is used for cases in k8s where you are going to run many containers that have something in common. Currently in the shim we create a special pause container. With sandbox api , do not need to consider the pause container. Trying to get containerd to call the `CreateSandbox` api. Created something called `shimkit` to test this all out and get Sandbox API working (github.com/jprendes/shimkit).

    - Have to change systemd to contain env var and update the containerd config.
    - Next step is to go beyond the CreateSandbox call to the rest of the API
    - Also figured out how to debug shims. When containerd calls the shim, the shim process has to finish immediately and print address to unix socket. The shim works as a daemon manager. Launches itself in the background. That is one of the problems with debugging the shim -- we don't launch the shim process. Hard because not launching with a debugger. Instead have to get the PID after the process was created. You can start the shim itself with the correct argos so that you can get the address of teh socket. I am going to add these debugging steps to runwasi.
    - James: Do you think we can accomplish what we are trying to accomplish from the sandbox api? Do we need to use the sandboxer?
    - Jorge: Sandboxer APi provides a plugin version to the sandbox api -- so same logic could be implemented as a plugin.
    - Joe: For context, the sandboxer api is not upstreamed to contaienrd yet. Kwasr implemented it in a fork of containerd. Daemon binary starts another binary that implements the task api. Similar to what Brian implemented with shared mode. I don't want to say it is not useful -- but until upstreamed, the sandbox one is the way to go
    - Jorge: ctr sandboxer cli seems to use some of the sandboxer api -- it doesnt use the sandbox api.
    - Joe: when jorge and i went to the community meeting and asked about the sandboxer api, it seemed like the maintainers didn't know much about it. We need to reach out to the kwasr maintainers and see their plans for it. Their motivation is much like the snapshotter API -- running the shim inside the virtual machine monitor VMM.
    - Jorge: I think hiday (??) is his handle. He is also involved with WasmEdge shim so maybe he has some information
    - Joe: What version of containerd are you using in the shimkit?
    - Jorge: whatever is shipped with archlinux (1.7.22)
    - Joe: so you still need to enable CRI sandboxes. Will this be enabled by default in 2.0?
    - Jorge: I don't know.
    - Brian and James not sure
    - Brian: I don't see any references to that env var in 2.0 -- only a reference in a test
    - Jorge: but in 1.7.22 i found a lot of references to it in the codebase -- so maybe changing
    - Joe: Yeah i was hoping the user would not have to make this change for 2.0
- [Joe] what do we do with containerd-wasm-shims?
    - Joe: The idea is that slight / spider lightning was deprecated, spin shim was moved to spinkube org, wasmworkers seems inactive for the past couple months, and i don't know if anyone is using the lunatic shim. Do we want to archive the repo? Or there is something valuable about having a collection of shims for ones we don't want to but in the runwasi repo.

  - Jorge: I for archiving the repo -- doesn't mean it goes away. Just says nothing else is happening there. Are we going to have another tier of runtimes than runwasi. How do we decide what goes in runwasi or not? We can put an archiving notice and direct people to runwasi or spinkube depending on where they want to go
  - Kate: had the same question about the distinction about what goes in runwasi
  - Joe: was based on whether is a wasm runtime or a platform; however, wasmedge is a bit of both. Also do we have a maintainer that can actively maintain that shim in the repo. We know who to reach out to. We have this for wasmedge and wasmer.
  - Joe: Brian do you have thoughts on this -- you've said you don't want runwasi to be bloated
  - Brian: still have that concern. Something you should be able to build off of. Wasmtime is fairly open while Spin is more specific
  - Joe: https://github.com/containerd/runwasi/pull/642
  - Brian: When we added WasmEdge, it complicated the build significantly.
  - Jorge: Adding Wamr would be similar since it has some FFI and linking with libraries. Both are c++.
  - Joe: Thats a good point
  - Brian: As long as things don't become a release blocker
  - Joe: What is the impact to the downstreams such as docker desktop
  - Jorge: Docker desktop is pinned to a commit. For spin it is already pinned to the spinkube repo. We can reach out to Georgi (??)
  - Joe: I vote for archiving the repo
  - [vote taking place in chat] 3 votes in favor, 2 abstained
  - Joe: I'll take on the task of archiving
- [Joe] More benchmarks https://github.com/containerd/runwasi/issues/704
  - Joe: wonder if we should compare perf with runc
  - Jorge: if you have an ubuntu based image with rust vs a distroless container with a binary. We will never be faster than a distroless container. Need to be careful about giving wrong information. She divides it into pull time vs time to start
  - Joe: if it is slower than 1 sec, then my fix was involved
  - James: pulling time can be messy because based on network
  - Jorge: I think having benchmarks and hooking up a profiler would be great
  - Kate: Testing with precompilation would be great
  - Joe: tracing would be helpful too
- [Tomasz] wasi:http/proxy work https://github.com/containerd/runwasi/pull/691
  - Tomasz: for wasmtime shim. Works similarly to wasmtime serve command. Shim will detect wasi http api and if that is the case, will spin up the server and pass the requests to the components. I can prepare some real use cases for the next meeting
- [Kate]: look at debugging pod level resources. Tried setting up the debugger but k3s' embedded containerd made it difficult.

- Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]

Oct 8th, 2024 (recording)

Host: Joe Zhou
Note Taker: James
Attendees:
- James Sturtevant
- Joe Zhou
- David Justice
- Jorge Prendes
- Brian Goff

Recurring Topics [timebox to N min]:
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:
- [Joe]: v0.7.0 released!
    - There is a breaking change in the library, please be aware!
    - https://github.com/containerd/runwasi/releases/tag/containerd-shim-wasm%2Fv0.7.0
    - 4 new contributors!
    - Shared mode deprecated
- David does intro to the meeting.
- [Jorge / Joe]: Sandbox API
    - Currently the shim manages the containers in a pod.This requires a pause container and we have a special logic to handle this container.
    - The sandbox api would replace the sandbox container
    - Provides an API that will allow you group containers
    - This api is new, so not as wide spread
    - https://github.com/kuasar-io/kuasar is a project that uses, this with a few changes that need to be upstreamed to rust-extensions
    - Looking for documentation of sandbox, Mostly in pr's issues.
    - Brian suggests this is something where we could improve the docs.
    - Can look at the CRI implementation for how it interacts with the sandboxer.
    - Joe: there is a community meeting we can join, Thursday this week
    - Jorge: should reach out to Kuasar and work with them to upstream changes
- Joe: Question about threading model
    - David: Probably need to look into this and validate our assumptions around which one
    - Jorge: Do we need to do threads vs process?

    - Brian: process vs threads, the main issue is resource monitoring/limits. The issue is cgroups really needs the process model. We want to make sure our overhead is minimal.
    - David: How do we start multiple wasm's with a single process. How do we group modules into a single sandboxed runtime?
    - Joe: https://docs.google.com/presentation/d/12yGT6_6guKgIZVqt18-o8tU11RtjtlyyBnxQIoj3miY/edit?usp=sharing How do we want to isolate, the sandboxer api gives us some options on different approaches.
    - David: need to do so
    - Brian, we don't need to have multiple wasm engines per pod but we need to make sure the pod is the boundary
    - Jorge, if we use microvms as part of isolation, sharing wasm engines will be challenging.
- Action Item:
    - reach out to Kuasar and work with them to upstream changes (Joe)

Sept 10th, 2024 (recording)

Host: Joe Zhou
Note Taker: TBD
Attendees:

- Jorge Prendes (MSFT)
- David Justice (MSFT)

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- https://github.com/containerd/runwasi/pull/668

Aug 27th, 2024 (recording) (Canceled)

Host: David Justice

Aug 13th, 2024 (recording)

Host: David Justice
Note Taker: James Sturtevant
Attendees:

- Comment your names here
- David Justice (MSFT)
- James Sturtevant (MSFT)

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Add your discussion item here
- WASM OCI Artifact - https://github.com/containerd/runwasi/pull/661
- Issues in ci: https://github.com/containerd/runwasi/issues/662

July 30th, 2024 (recording)

Host: David Justice
Note Taker: TBD
Attendees:

- Comment your names here
- Brian Goff (MSFT)
- David Justice (MSFT)
- James Sturtevant (MSFT)

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- (Justice) There were no discussion items covered during the meeting. We did discuss that there will be some holiday time coming up for multiple of the maintainers during August, so it is likely that we will not have quorum during the next meeting. We will follow up via Slack to ensure folks are kept informed on the status of upcoming meetings.

July 16, 2024 (recording)

Host: David Justice
Note Taker: TBD
Attendees:

- David Justice (MSFT)
- Jorge Prendes (Docker)
- Brian Goff (MSFT)
- Joe Zhou (MSFT)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- (Jorge) OTEL conflict between shim and spin
    - Notes:
        - Shim and runtime sets up set_global_ in both which is a problem
        - The discussed solution was to use the OTel lib, but that may need a lot of changes in underlying libs
        - Another option is to set_default per thread
        - Spin folks recognized they should not set_global since Spin is being used as a library rather than a top level executable. Unfortunately, that means they are going to need to set_default for each thread.
        - We don't have to use the tracing crate. OTel has its own SDK which allows for creating span without the use of the Rust tracing crate.
        - We could also change TTRPC Rust to offer a hook for setting up a new trace context for each thread.
        - If Spin wants to stop using the global tracing provider, they would need to do the same thing.
        - This problem is not only for the tracing crate. This is also a problem for the global logger.
        - If we don't go with the global tracing provider, it is going to be viral. Each thread is going to need to set this up, correct.
        - (Joe) correct.
        - (Joe) The reason both set_global is since we have 2 configured endpoints, containerd and the pod configuration. Both shim and runtime are multithreaded.
        - (Jorge) I wonder if we should have 1 endpoint and have it dispatch to the 2 endpoints.
        - (Joe) We do need to support 2 endpoints.
        - (Jorge) If we have 1 provider, we could dispatch to the correct endpoint.
        - (Joe) That is doable theoretically. I've had difficulty navigating that.

- (Joe) We are using the tracing subscriber, which allows for multiple layers. It allows us to configure lower layers with different trace levels. I think we could do something similar to dispatch to different providers. I think we could setup 2 subscribers that listen to mutually exclusive paths.
- (Jorge) Tracing is from Tokio, right. So, you don't own thread creation. Tracing usually uses thread local storage to store trace context. I wonder if we could use something from Tokio to help us do this. I'm wondering how they do that in Tokio or if there is some special relationship between them.
- (Joe) Speaking of Tokio, I wanted to discuss your concerns with creating a Tokio runtime in the main process.
- (Jorge) The way that we are forking the process in Youki, we are using clone to start the thread. The reason we have not hit any major issues with that is since we have a lot of dormant threads in TTRPC. Libc has some synchronous constructs to deal with this. If we have a Tokio runtime running, when we clone, it may run into more undefined behavior. We may get lucky. If you haven't experienced issues while testing OTel, it's probably fine. If we want to fix this, it goes to the next discussion point.
- (Joe) thus far we are not running into problems. You are saying that we may b/c tokio is starting threads?
- (Jorge) If we are starting a process when we hit a sync point, like allocating memory in the clone, we may hit that error.
- (Jorge) If you call clone, you will get a new process with a copy of the thread that made the clone syscall. The main problem arises when you need sync across threads. Perhaps, before calling clone from a thread that has a lock, afterward, you will never be able to unlock that lock b/c it is no longer there.
- (Joe) That may explain the situation with the missing traces that David and I observed / investigated. This happens after we clone and there are no traces omitted any longer.
- (Jorge) Yes, you may be sending messages to the void.
- (Joe) If I try to setup another pipeline for the tracing provider.
- (Jorge) That may be due to it being a global with a lock. The other thread keeps the lock. When you clone you get a copy of that memory, and the cloned thread will not be able to access the lock.
- (Joe) Then the suggestion is to be single threaded?
- (Jorge) No… I'm not sure that tracing requires multiple threads. If we can do it single threaded, it could be a solution, but that depends on tracing.
- (Joe) sounds like the best solution now would be to get away from the global tracing. To be clear to remove set_global from the shim will not be enough.
- (Jorge) Not seeing the traces inside of Youki smells of a multithreaded issue.
- (Joe) Thank you so much for all of the comments!

- (Jorge) Process forking UB (linux solutions and approach for windows)
    - Notes
        - (Jorge) We discussed the problems of seeing multithreaded issues. There's different options we could have. I would also like to discuss an approach for Windows while James was here. My only 2 ideas are either doing the shim process single threaded, or when we fork in Youki, we do a fork exec, however, fork exec wouldn't work since we don't have the bin to exec in the container. The other option is single thread / maybe Youki is doing too much for us. Perhaps, we should use lower level APIs from libcontainer. The builder from libcontainer was handling a lot of the heavy lifting for us.
        - (Joe) It would be an interesting conversation to figure out what is necessary.
        - (Jorge) It is good to have the security in depth.
        - (Brian) I think it comes down to what works. We are not going to have user namespaces since threading tricks are not going to work out. Perhaps, seccomp filter are not going to work out and the runtime knows what to limit.
        - (Jorge) If we drop some of the sandboxing, we may gain in performance.
        - (Brian) There is a ton of synchronization involved in parent / child processes.
        - (Jorge) There is overhead on the first write (copied memory). I think one of the problems is going to be pid namespace.
        - (Brian) There's time and something else as well. Time and PID.
        - (David) Would the absence of seccomp filters be incongruent to a Kubernetes user?
        - (Jorge) an option would be to allow a pause container or something else handle the initial setup. We could do this at the shim level. Each instance would not be as isolated, but the pods themselves would be as isolated as they are at the moment.
        - (Brian) There's already things that don't make sense in the pod security context, like run as user; unless we intend on running the runtime as a user. There's app armor capabilities, readonly rootfs (probably could do something there).
        - (Jorge) It's not clear to me that there's no value there. There may be value there. POSIX requires per process user, but Linux kernel is able to do this per thread.
        - (Brian) I think this is going to be tricky.
        - (Jorge) I think the Youki builder is doing a lot of heavy lifting.
        - (Joe) When you say using low-level APIs from Youki, is it lower level than the init process, intermediate process level.
        - (Jorge) Yes. We could use libcontainer to build the rootfs and many of the other setups without going straight into the builder, which parses the OCI spec and everything.

        - (Joe) To get started on that, Brian had a non-Youki implementation which setup the cgroups. We could bring that back and use it.
        - (Jorge) In a way the shim is already doing something similar when the shim bin executes itself. There are many approaches we could use (argv 0).
- (Jorge) Wasmedge 0.14 update
    - Notes
        - (Jorge) Wasmedge 0.14 is a BIG API change at all levels. When the Rust SDK was released, some of the bins were incorrectly specified. It is not clear to me how we should be using the Wasmedge Rust API. There was no clear migration. I've talked with some of the Wasmedge folks. They will be updating samples to provide migration guidance.
- https://kuasar.io/

July 2, 2024 (recording)

Host: Joe Zhou
Note Taker: David Justice
Attendees:

- David Justice (Microsoft)
- James Sturtevant (Microsoft)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- **(David)** Telemetry PR https://github.com/containerd/runwasi/pull/582
    - **Notes:**
        - Asking for a quick update on the PR
        - (Joe) Still testing with the Spin shim and not seeing any log / trace spans even though I've changed the rust log level. There is also a comment from Brian that I need to add an env standard var. I haven't done that yet, but it should be a pretty easy add.
        - (James) I think I was just waiting on the Spin shim results, and how the env var was being used.
        - (Joe) have you ever seen the log messages from the Spin shim?
        - (James) yes, I believe so, but it's been a while.
        - (Joe) I will follow up with you, b/c I was unable to see them.

        - (Joe) Any additional feedback on this PR
        - (Brian) No. The containerd PR to pass down the trace context is about to land (https://github.com/containerd/containerd/pull/10186)
- (James) I opened a PR to update the OCI spec version and generates the correct format using the library that generates the updated format.
- (Joe) The OCI tar builder, we should release the bins
- (James) Once things stabilize more in the bytecode alliance repo, we could use the code there to do the OCI tar builder.
- (Brian) I think we can make buildkit output the right media types.
- (James) I think I talked with someone from Docker that was interested in doing that. I think there are some cool things you could do with that which could take a file / filesystem to build a WASI-virt static FS layer that would be available to the Wasm application (would be pretty cool!).
-

June 18, 2024 ([recording]())

Host: Joe Zhou
Note Taker: David Justice
Attendees:

- Kate Goldenring (Fermyon)
- David Justice (Microsoft)
- Brian Goff (Microsoft)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Joe: Design of the Wasm Sandboxer API [Support V3 containerd task service and Sandboxers · Issue #385]()
    - Notes
        - Prepared Sandbox API
        - Containerd 2.0 is going to release soon. In 1.7 introduced the sandboxer api.
        - There is a long thread on the challenges of integrating K8s and firecracker.
            - Impedance of containers vs. VMs.
            - Containerd doesn't natively support the concept of grouping containers as K8s does with pods. In Runwasi we implement a

        fake pause container, then we move on to creating Wasm containers.
    - The resource specification in the CRI is very different from what is needed in Firecracker.
- You can think of a sandbox as a group of containers.
    - Acts as a parent entity for containers.
    - Acquires resources needed to run child containers. It will act as a pause container to acquire the namespace.
- There is a sandbox controller - there are 2 implementations, pod sandbox controller and shim sandbox controller. Both registered as container plugins.
- {Describing the workflow diagram of containerd}
    - First we have a CRI service plugin, which can specify the pod sandboxer or the shim sandbox controller.
    - The pod sandbox controller follows a similar path as what exists today.
    - If you choose the shim controller, it will have a task and sandbox service remotely (over ttrpc?)
- {showing code of the containerd client side implementation}
    - You can use the sandbox to create a new container. You can also start a container given sandbox id. Then, you can run tasks.
    - The sandbox apis are invoked prior to creating a container.
- {showing code for the Sandbox service}
    - life cycle management of the sandbox
- {showing diagram of Runwasi shared mode}
    - Currently, containerd asks the shim to start a task. The shim is then talking with a daemon running on the node over a Unix socket to delegate that call. Containerd can talk to the task server in the “sandbox”.
    - Shared mode has one instance of wasmtime shared on the node.
- {showing Runwasi sandbox API}
    - Proposal to have 1 bin that implements both the sandbox api and the task api. Sandbox api can simplify our project by having 1 process that can both sandbox and tasks.
    - (q: Kate) If we create a component that handles http, will they both have their own ip bind?
    - (Joe) they are related. A task is a container. That container will start a task.
    - (Kate) Just want to clarify that we will have a single listener per application.
    - (Joe) Yes, nothing changes here (in the app).
    - (Kate) wanted to clarify since we have been discussing listener address configuration.

        - This is what I propose to replace the shared mode. We can get rid of the separation of the shim and the daemon.
        - (Brian) We could treat the whole thing as a sandbox, so we wouldn't need to deal with the pause container. We would need to do the setup related to the pause container, but not actually use the pause container.
        - (Joe) question about threading vs process
        - (Brian) The main difference between shared and shim. We are giving 1 socket per.
        - (Joe) in standard mode every container creates a new process for each container.
    - What about the sandboxer api?
        - The quaza? team has introduced a new api for the sandboxer. They have responded that the api they have envisioned has not landed yet in containerd and probably won't for some time.
    - {showing reference slide}
- James: Container vs wasm host(etc, spin) vs wasm instance boundries https://github.com/containerd/runwasi/issues/619
    - Notes
        - We've been having a few different conversations on GH on this. We have the container that is wrapping around the host of the wasm runtime. We also have the wasm instance. When you inject env vars through CRI, they are being set on the container. We then plumb them into the Wasm instance. Spin shim doesn't do it quite that way. One of the things we do is that when we start the container, the env vars are set from the container config. Afterward, we do some additional setup ending with firing off the instance. There is some configuration that might be considered part of the shim, but it is something the host is doing to setup the runtime. Say you want to set the RUST_LOG for the wasm instance, it would override the short time when the Wasm instance is being constructed for run. I think the way Spin is doing it is the right way. I don't think we should directly map the env vars directly from the container into the Spin instance.
            - (Brian) I think that the way it is now is an unintended artifact of using Youki. When we started, we were directly adding these env vars to the Wasm runtime.
            - (James) When I was looking at it, Wasm has a host and instance. I believe the host should be running inside the container. The Spin engine is running in the host and access to the network namespace. The instance itself is another layer and that's where it gets fuzzy.
            - (Brian) Probably even more when we start to focus on components.

- (James) in the Wasmtime shim, we are injecting all the env vars. I don't think we have a way to represent what vars are expect to be relayed to the Wasm application.
- (Kate) The access of Spin to have access to the env vars was a nice coincidence. When you are talking about Spin application, you are not talking about a single application. It can be composed via multiple applications. That is why we use application vars in Spin. Env vars is not the most native way to use vars. This get's back to the question of should this be a shim by shim decision. I keep flipping on myself. It doesn't seem like there is a clear right answer. Am I missing something in containerd about how we should configure this.
- (Brian) I think this comes down to the wasm engine part of the workflow. When we try to map OCI stuff into the Wasm world, do they really apply? In containerd land you can pass container specific configuration that the shim could process and change as needed. When CRI passes that down, it's just passing down toml, raw toml that we could parse. I think it's going to depend on each implementation to determine if this is part of the workload or the shim.
- (Kate) just to clarify that toml config. Do you mean the runtime options?
- (Brian) yes
- (Kate) say we want to set the listen address.
- (David) I think this is an application configuration
- (Brian) if this is at the host it's shim, if it is namespace then application
- (Joe) I think 1 IP per sandbox makes a lot of sense. Does this mean we will run spin up in each sandbox?
- (Kate) I think this is an important discussion about what we will expose to users. 1 OTEL exporter, 1 log listener and configure them in the containerd config. It flips the direction of where we put this information. We are probably months away from an impl. If we allow configuration now and then later have to change, we will have a hard time telling that story.
- (Kate) I think the experience for community consistency having 1 IP per instance is what we should do.
- (Brian) maybe we can flip how you configure the listener is by prefixing the env var to be able to pull it from bag of env vars
- (Kate) that is in the proposal.
- (David) How do we handle things like identity env vars?
- (Kate) I don't see an issue with the spin runtime getting access to env vars. It doesn't use environment vars called spin foo.
- (Joe) Brian can you elaborate on allow list idea

        - (Brian) The generic wasi thing that we have today. I feel like it's a special case of things. Other things like Spin, does it really need access to a ton of env vars. Let the shim say what access to env vars rather than all env vars. I think the component model makes this interesting too. I don't really want to put the impl that deals with my workload identity solution. I'd probably want some trusted component to deal with that which would need an env var.
        - (Kate) Why doesn't the Spin runtime understand what env vars a component wants. We have kv stores allowed and other things, but not what env vars are allowed. We could say these env vars allowed. That is something we could add to Spin and it is responsible for allowing / disallowing.
        - (Brian) I was also thinking about how Spin could configure itself. Perhaps, you have a Wasm component that could be used. At least have a component that deals with env vars.
        - (Kate) I agree that the use of env vars and components may not be a use case. Use Spin prefixed vars to configure the runtime. Otherwise use application variables to configure the app.
-

June 04, 2024 (recording)

Host: Joe Zhou
Note Taker: Kate Goldenring
Attendees:

- Comment your names here
- Kate Goldenring (Fermyon)
- James Sturtevant (Microsoft)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Joe: Runwasi roadmap to 1.0
    - Notes:
        - Roadmap is a bunch of issues that are important.
        - Joe's target for runwasi 1.0 is end of the year -- robust and production ready
        - Issues:

- Add OTEL to runwasi
- Startup performance
- Shim cannot connect to runtime daemon. Daemon mode is the weakest point of runwasi because we don't have tests and use cases right now. Support V3 containerd task service and sandboxers issue makes the old implementation of daemon mode obsolete. Instead of allocating resources to maintain daemon mode, we should implement the sandboxers API
    - Have we assigned priority to any of these issues? Supporting v3 containerd is a nice to have
- Windows support. James started working on this in a branch in May. Focusing on new OCI format from Wasm WG and getting it upstreamed into containerd. The way we got the Spin OCI to work was essentially a hack. Containerd doesn't handle artifacts today, so we need to teach it how to handle those artifact types. Containerd will fetch the artifacts and store in the content store. Right now, it is a container image media type with custom layers. Are you adding Windows support to Youki? Youki is very specific to linux. We are not implementing windows containers in this case. We are currently only creating a job process. The job object gives you some level of isolation and we are relying on wasmtime's sandboxing to provide the sandboxing mechanism
    - This is not blocking for 1.0
- Pre-open directories and other capabilities: How are we going to pre-open directories for WASI and what permissions we will assign. Related to the issue in the spin shim of not seeing the volume mounts #108. Do we map in all of the volumeMounts? Some may be for the host system and some for the container. Each shim is responsible for this
- Not seeing metrics for cpuacct.usage: Joe is looking at it. P0 because related to metrics
- Add support for structured logging. James added the ability to do some structured logging with timestamps. Intention was to follow it up with a macro that makes it easy to add the container instance id automatically. Then you can filter by the instance and see everything that happened for that container. Shouldn't be too hard to add.
- Unidentified Behavior occurring
- Why is the shim running on its own network ns? 4 out of 20 times the integration tests were failing. The Youki maintainers closed the linked Youki PR saying they were doing the right thing for it. Still need to keep looking at this
- Create an API for pre-processing layers: This is a braindump on the feedback gotten during the precompilation work to make the

API more generic rather than focusing on pre-compilation. This can be removed from 1.0 as it is a nice to have

  - Extract CRI runtime configs to separate files: When we change config.toml, we don’t want to have to restart containerd. Each shim would have it’s own config. At the moment, all the runtimes are configured in config.toml.
  - Adding containerd config options for the shim: What are the scenarios of things we’d want to set in the containerd config that we wouldn’t want to set in args that we’d want to set in the config. Do we want to go the args route for now or pipe through containerd config options because we can?
    - Joe: containerd options generates a lot of friction because you have to restart. But if there is a fixed set of options that they want for all the workloads on that runtime.
    - Original reason was to turn off precompilation. The general idea of being able to able to toggle features in important
  - Benchmarking: How does it compare to other runtimes
    - https://github.com/fermyon/spinkube-performance
    - Runc vs crun vs runwasi startup time with wasmtime
    - Run per PR
      - For startup time, use https://github.com/Mossaka/runwasi-perf
      - CPU and Memory
    - Run before releases: For density, run spinkube-performance suite
-

May 21, 2024 (recording)

Host: Joe Zhou
Note Taker: David Justice
Attendees:

- Comment your names here
- Kate Goldenring (Fermyon)
- David Justice (MSFT)
- James Sturtevant (microsoft)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees

- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]
- Joe: updates on observability
    - Notes:
        - Joe has made tracing optional. This PR (<pr>) will make the tracing dependency option using a Rust feature. If the feature is on, then we use the tracing attribute. The macro will make sure the tracing attribute is not included when the feature is disabled.
        - Joe also has a PR (<pr>) on Rust extensions repo that will also enable tracing including the shim crate that runwasi depends upon.
        - The tracing PR is ready to review. Please give it a look.
        - **Call to action**: Please review the PRs when you have time.
- Kate: Shim configuration with options https://github.com/containerd/runwasi/issues/573
    - Notes:
        - There’s been a few issues that have been discussed about how to pass options to the shim. We want to pass the otel endpoint for example. We could set these in the containerd config, and have not been able to pipe those options through. I wanted to get a sense if we want to have a set of options that we would like to have and how would we like to pass down the config options.
        - (Joe) I looked into this issue by debugging containerd when it sends the TaskCreate to the shim. I looked into what’s in the configuration, and I wasn’t able to see the config “foo=bar” and was unable to see that. Not sure if ctr is able to do that. Correct me if I’m wrong. I’m new to this too.
        - (Kate) let’s proceed to testing as Joe suggested. Should we be opinionated about the structure of what config options?
        - (Brian) all containerd config will be done through env vars in the future. We should ensure that shim is passing these along especially if the shim is running as a daemon. I don’t think things should be opinionated.
        - (Kate) Just clarify. Are you saying that options are not going to be used and everything is using env vars, or is going to be easier using otel.
        - (Brian) only env for otel
        - (James) I think this is not just otel, but config options like runc options.
        - (Brian) ctr should be passing some options, but I don’t know what it looks like off the top of my head. But, yes, cri does pass runc options. If it’s not the runc options. I think it marshals the toml directly
        - (David): task_create struct has an options of any type field, not sure how much of that is in runwasi, probably want each shim to describe this struct or extend from the base struct what the options are.
        - (Brian) I was just going to say that 100% correct. Sometimes containerd will marshal the raw toml and the implementation needs to handle it.

- (James) I think coming back to what Kate mentioned, runwasi doesn't really know what to do with the options. We need to hand that off to a shim. One of the things that came up was how to configure the underlying runtime. We probably want to be able to handle key/values too so that we can pass those down to the runtime since there are a lot of flags.
- (Kate) Keeping runwasi non-opinionated and passing to the shim would work
- (James) the precompilation feature is handled by runwasi. There may be a few features across the shims we'd want to turn on / off
- (Kate) do we want to determine a naming convention to route to runwasi or to shim
- (David) If the naming convention has a structure, `runtime` section with key-values, and `shim` specific section for that configuration. First step is to how to pass in the options, and from here, what is the best way to structure this to be kind to operators / devs working on the project
- (Joe) in today's options field they already have reserved keywords (binname). Perhaps, we can have some reserved runwasi names that we document as well.
- (Kate) for a user experience, in SpinKube we have the runtime-class-manager (rcm). We'd expect to be able to update the configuration of the rcm resource, then we could do an update cycle.
- (Caleb) would this update the env vars?
- (Kate) I think all updates should be owned by the rcm
- (Caleb) would this be 1 containerd / node and not per spin app?
- (James) you could have another runtime class with different configuration
- (Kate) clarifying that you could have a spin1 and spin2 runtime class with different runtime classes.
- (James) if you wanted 1 with pre-compilation on and one off, you could do that
- (David) where does dynamic configuration come to play here? For OTEL endpoint, that makes sense to be in configuration of containerd or the shim itself. For OTEL for the app level, it should be in runtime / dynamic config in the wasm component provided by the pod spec or applied at that level instead of in configuration of containerd?
- (Caleb) That may be right, but I'm not sure we are going to be able to support 2 tracing providers in Rust. Not sure if something coming in on the app env might be too late. As a Spin dev, I don't think I want to think about containerd config.
- (Kate) Is it wasi:config or Spin runtime config? I think of about spin runtime config as at start up / configuration. I think of wasi:config as something that comes after the application starts.
- (Caleb) Can we get pod env vars into the application?
- (David) Yes we can.
- (Kate) does the shim have access to the env vars too?

- (Joe) yes, the shim does have access to these env vars from the pod
- (James) I think we have multiple layers of config. We need to identify them and document. We then should look at where each one fits and how it should work. I'm just making examples, but it might make sense. Perhaps, we should, in the issue, outline them and flesh that out.
    - Joe +1
- (Kate)
    - Action recap
    - 1. Double check if we use CRI are these options passed
    - 2. Start a thread on the variables we want to set and where they fit
- (Joe) I found it is extremely helpful to setup the debugger on containerd to figure out what is being set / sent. Ping me for help.
- (Kate) I'll reach b/c I haven't set that up yet.
- (Joe) Do we want to raise an issue on runwasi to document the env vars? Is that a good first step?
    - Runwasi layer (otel endpoint, disable pooling allocator, pre-compilation)
    - Runtime specific (any runtime flags to `spin up` or `wasmtime serve`)
    - Application specific (Spin runtime config)
- (Joe) WASI has its own configurations like directories you can read or write. Wasmtime is very different from WasmEdge. Do we want to bake in specific shim configurations so runwasi doesn't need to care about configurations
- (David) Do we have any concerns over how often we capture traces? There are a lot of things we can do here, but we may want to be lazy about it waiting until people have specific needs.
- (Brian) More application specific vs shim layer. (sry… I missed some of this)
- (Caleb) Waiting for a specific need is good, but I'm coming with a specific need for getting a solution for Spin users to configure. I think we are moving the right direction.
- (Joe) Frequency of capture - I think we've concluded we would like to wait for specific need.

May 7, 2024 (recording)

Host: Joe Zhou
Note Taker: Jorge Prendes
Attendees:

- Comment your names here
- Jorge Prendes
- Brian Goff (MSFT)

- David Justice (MSFT)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Joe: Observability updates
    - Observability in Wasm OCI Runtimes
    -

April 23, 2024 (recording)

Host: Joe Zhou
Note Taker:
Attendees:

- James

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- Please add your name to the attendee list above

Open Discussion [timebox to N min]:

- Trapeze, an async ttrpc library by Jorge
- Joe: Observability updates
- Joe: Release pipeline updates
-

April 9, 2024 (recording)

Host: Joe Zhou
Note Taker: Kate Goldenring
Attendees:

- Kate Goldenring - Fermyon
- Brian Goff - MSFT

- David Justice - MSFT

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- Please add your name to the attendee list above

Open Discussion [timebox to N min]:

- Joe: release pipeline https://github.com/containerd/runwasi/pull/549
    - Joe's work inspired by BA wit-bindgen repo. Use the workflow trigger to start a release. It creates a PR to bump the version which then triggers a final workflow to create the release that publishes the artifacts, crates, etc. See PR #549.
    - Brian: how do you get the token to create a PR? Usually you cannot use the actions token
    - David: You couldn't push directly to main but you could push a tag
    - Joe: I used `gh pr create` which will use whatever permissions are on the GHJ action
    - Brian: last i saw, ambient identity cannot create another action
    - Kate: We do something similar in Spin, creating a PR in another repo in our release workflow in order to update our homebrew formula: https://github.com/fermyon/spin/pull/2428
    - David: Can only maintainers create tagged releases? How do make sure we lock down who can kick off this workflow
    - Joe: We could defer that to the review mechanism of PRs. We can also check if the commit was made by the bot
    - General consensus that want this automation
- Joe: An update for observability
    - Joe: troubles figuring out span with series of process calls. When you want to propagate the context from one process to another, serialize that context to a json stream and feed it to another process to continue the span. Unblocks the work I started 5 weeks ago. Mosaka/otel-condvar repo. Ideally the context can be sent back to containerd and then it is sent to the shim
    - Kate: runwasi main will create the span. What responsibility does the shim have?
    - David: how does this pair with the Spin OTEL work?
    - Kate: Add Otel work to spin shim: https://github.com/spinkube/containerd-shim-spin/issues/61
    - Kate: create a best practice around this is important
    - Brian: don't want the shim spans to propegate to the application right?
    - Brian: Yes! Separation between infrastructure and application. Don't want the tracing span from shim and containerd to got to the application but you may want to export them to the same place.
    - David: we need a new root span at the beginning of the application context
    - David: could we have named subscribers? How do we support multiple OTEL exporter endpoints?
    - **Joe will reach out to Caleb and continue looking into adding traces to Runwasi and rust extension**

- Joe: Remove spin shim from deislabs/containerd-wasm-shims: https://github.com/deislabs/containerd-wasm-shims/pull/217
    - Docker put up a request to remove it from the deislabs shims. Joe has a PR in to remove it. **Call to action to review it.** *-12k lines*
- Joe: containerd 2.0 and sandboxer APIs : https://github.com/containerd/containerd/issues/9431
    - Joe: asked a maintainer of containerd for an update on the progress of the sandbox API. Looks like it will be in the 0.2 release. Joe is aiming to get involved. Need to get it implemented in rust-extensions, then runwasi.
    - Brian: we should be able to start implementing in rust-extensions since a lot of the implementation is already in containerd. Ideally do it before the 0.2 release so we can determine if any changes needed before the release
- Joe: Runwasi roadmap to v1.0
    - Joe: also presented this in the SpinKube developper meeting.
    - GIST: @joe please add
        - Asyncify runwasi: Jorge mentioned this could be advantageous for hardening because there are currently some places that are susceptible to deadlocks
    - Runwasi 1.0 project board - we need to **make it public**
    - 1.0 means production ready so hardening performance is the most important part
    - Joe: Want to see if it makes sense for containerd to collect the information from the spans
    - Brian: There is support for gRPC in containerd and support for otel in ttrpc

March 26, 2024 (recording)

Host: David Justice
Note Taker: TBD
Attendees:

- Brian Goff (MSFT)

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- Please add your name to the attendee list above

Open Discussion [timebox to N min]:

- Check out the new release!! There is support for precompiling Wasm modules and components.
    - How does it work? What does it do?
        - In the past we packaged an image and put your module in it – it's not optimized for where it runs. Now when we start up, we pre-compile the module at startup time by inspecting the layers, then store it in containerd. Upon subsequent runs, the runtime can reuse the pre-compiled content.

    We see some startup time savings. The longer the compile time, the better your results.
    - Zeiss + SpinKube at KCEU using precompilation: Keynote Panel Discussion: Revolutionizing Cloud Native Architectur…
- Releases unstable
  - We are reworking the runwasi release flow and looking for some feedback. Moving from a PR with a crate version update followed by a tag to start the release workflow to a workflow_dispatch trigger that will take as input the crate and version, create the commit updating the crate version, and apply the tag upon successfully pushing artifacts. This should make it easier to interact with the workflow_dispatch trigger and reduce the opportunity for error.

March 12, 2024 (recording)

Host: Joe Zhou
Note Taker: TBD
Attendees:

- Kate Goldenring (Fermyon)
- Brian Goff (MSFT)

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- Please add your name to the attendee list above

Open Discussion [timebox to N min]:

- Mossaka: Setting up the mailing list
  - Maybe we should put runwasi community meeting on the CNCF calendar https://www.cncf.io/about/contact/
- Kate: termination process for runwasi and shims.
  - Sometimes the processes don’t get terminated in the shims.
  - Brian: doubt it’s in the containerd. It might be in rust-extension or runwasi.
- Joe: otel update: https://github.com/containerd/runwasi/issues/10
- Kate: should add tracing to the shim.
- Brian: If logging is causing perf issues, there might be some issues with the use of locks in logging.
- Rajat: `ctr task list` returns a list of unknown status with pid 0 tasks.
  - https://github.com/containerd/runwasi/issues/418#issuecomment-1991073616
  - Use ttrpc as a client
- Kate: kind consistently has the issue with the latest. K3d never. K3s sometimes. GKE cluster has.
- Rajat: if the pod started for 5 - 10 seconds and then deleted, it’d stuck.
-

Jan 16, 2024 (recording)

Host: Joe Zhou
Note Taker: David Justice
Attendees:

- Angel M Miguel (VMware / Broadcom)
- Brian Goff (Microsoft)
- David Justice (Microsoft)

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- Please add your name to the attendee list above
- Congratulation to James and Jorge promoting to active maintainers of runwasi

Open Discussion [timebox to N min]:

- Mossaka: Follow up on https://github.com/containerd/runwasi/issues/418
    - Notes
        - Joe: Status update Joe would like to give is that he has done some testing on the main branch and added integration tests to verify the pods are terminating. Any questions on this issue?
        - David: any environmental factors?
        - David: someone is using spin shim in AKS wasi nodepool, and used Kwasm to install the newest bits and saw this problem go away. It signaled that the bug has been resolved, but there are remaining concerns.
        - Joe: We are not sure what has changed that may have solved this issue.
        - Angel: Perhaps, replicate using a small env
        - Brian: We need to ensure we are sending the signal back to the wait API. Comment was posted on the issue.
        - David: found it difficult to debug at this level. How do you go about it, Brian? How could we add more tracing and observability?
        - Brian: There is support for Otel in containerd. We could wire that up. There is no support for Otel in ttrpc. You'd also need to turn on debug logging. We have talked about adding Otel ttrpc support, but it has been challenging. On the rust side having Otel may be easier than on the Go side b/c of code dependencies.
        - Joe: Brian when you say there's no support for Otel in ttrpc, I thought about the API version upgrade from v2 to v3. Do you think that will help to add Otel support.
        - Brian: If we are going to use gRPC, then we need to think about what that is going to do to our shim (memory, etc). We need to measure the impact. On the rust side might work on go side, it is larger.
        - Joe: Showing architecture image describing containerd runtime API v2 for shims. Where in this is the signal to terminate.

        - Brian: We are sending sigterm to the process over an RPC call. We are not expecting anything back other than the response it was performed. What we are expecting to see after that is a container exit event; for CRI specifically, we are waiting for wait() to exit. Then we need to call task delete and clean up after it.
        - David: the ttrpc connection was disconnected before sending the signal back.
        - Brian: The disconnect shouldn't be a problem, but CRI might have a bug. Check by sigkill'ing the shim and see what happens (could add to our CI pipelines).
        -
- Justice: https://cloud-native.slack.com/archives/C04LTPB6Z0V/p1705059273058979
    - Notes
    - Related issue in spin shim:
        - Joe brings up the slack discussion on perf comparison between crun and runwasi. There is a significant difference (~1 order of magnitude) between the executions.
        - David: all point to debugging and observability
        - Brian: We may be able to export custom metrics and also tracing would be greatly helpful.
        - Sven: it takes up to half a minute for a wasm container to be available.
- (last if we have time) Mossaka: A build failure on containerd-wasm-shims
    - Notes
        - Joe: In the containerd wasm shims repo I recently ran into a build failure. This is for any shims. The build error shows for Lunatic, but same for other shims too. There's no change in env or other aspects, but when we compile to musl, we see something like relocation against … cannot be used when making a PIE object. Does this mean we need to recompile musl with -fPIE.
        -

Jan 2, 2024 (recording)

Host: Joe
Note Taker: TBD
Attendees:

- Comment your names here

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Discussed runwasi open issues

Dec 5, 2023 (recording)

Host: Joe
Note Taker: TBD
Attendees:

- Comment your names here
- Sven Pfennig (LiquidReply)
- David Justice (Microsoft)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Component model in Wasmtime https://github.com/containerd/runwasi/issues/384
- Shim V3 support https://github.com/containerd/runwasi/issues/385
- Merging containerd-wasm-shims repo discussion https://cloud-native.slack.com/archives/C04LTPB6Z0V/p1699387939555679
- Add guidance for adding and removing shims: https://github.com/containerd/runwasi/pull/408

Nov 21, 2023 (canceled)

Host: TBD
Note Taker: TBD
Attendees:

- Comment your names here

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:
(moved agenda items to Dec 5th)

- Component model in Wasmtime https://github.com/containerd/runwasi/issues/384
- Shim V3 support https://github.com/containerd/runwasi/issues/385
    - Explain the APIs to the meeting

- Merging containerd-wasm-shims repo discussion https://cloud-native.slack.com/archives/C04LTPB6Z0V/p1699387939555679
-

Nov 7, 2023 (canceled)

Oct 24, 2023 (recording)

Host: James
Note Taker: TBD
Attendees:

- Comment your names here
- James Sturtevant
- David Justice (Microsoft)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]
- Running runwasi shims with a “runc-like” CLI interface - @Sven Pfennig
    - The question is: Can I do this with Cri-o?
    - wasmtime/and wasmedge work in runc/yoki
    - How do we run the containerd-wasm-shims(spin/slight/etc) with runc?
    - Demo time!
    - Works with crio/openshift
    - Is there a need for this? What do folks think?
        - Looking for customers/users using cri-o
        - Folks using podman and not containerd, its not easy
    -

Oct 10th, 2023 (canceled)

Sept 26th, 2023 (recording)

Host: James Sturtevant
Note Taker: TBD
Attendees:

- Angel M Miguel (VMware)
- Jorge Prendes (Docker)
- Sven Pfennig (Liquid Reply)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- [James] OCI artifact update: https://github.com/containerd/runwasi/pull/147 [James] process hang https://github.com/containerd/runwasi/issues/326

Aug 29th, 2023 (recording)

Host: Jiaxiao (Joe) Zhou
Note Taker: TBD
Attendees:

- Brian Goff (Microsoft)
- James Sturtevant (Microsoft)
- Jorge Prendes (Docker)
- Sven Pfennig (Liquid Reply)
- David Justice (Microsoft)
- Angel M Miguel (VMware)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- [Jorge] Add simplified libcontainer instance API #250
- [Sven] Backward compatibility issues with 0.9

Aug 15th, 2023 (Cancelled)

Host: Jiaxiao (Joe) Zhou
Note Taker: TBD
Attendees:

- Comment your names here

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]

Aug 1st, 2023 (recording)

Host: Jiaxiao (Joe) Zhou (Microsoft)
Note Taker: TBD
Attendees:

- Djordje Lukic (Docker)
- Angel M Miguel (VMware)
- Ismo Puustinen (Intel)
- Jorge Prendes (Docker)
- James Sturtevant (Microsoft)
- SvenPfennig (LiquidReply)
- Hung-Ying Tai (Second State)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- **Joe: release pipeline updates**
- **Jorge: release on alpinze?**
- **James: OCI updates**
    - https://github.com/containers/youki/issues/2220
    - Warg + runwasi: https://www.youtube.com/watch?v=SEwBcEFgXUM&feature=youtu.be
-

July 18th, 2023 (recording)

Host: Jiaxiao (Joe) Zhou
Note Taker: David Justice
Attendees:

- David Justice (msft)
- Jorge Prendes (Docker)
- Djordje Lukic (Docker)
- Jiaxiao (Joe) Zhou (msft)
- Brian Goff (msft)
- Sven Pfennig

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]
- (Jorge) Discussion: release process
    - Notes:
    - Currently, for Docker Desktop, we have a separate repo where we build artifacts. We have been doing it for a couple of shims. To do it for all the shims, it would take a lot of effort. We are downloading artifacts from Deislabs. It would be great to do the same thing with the runwasi bins. I think it would be great to fix the release process and go into a more standardized process for releasing. For example when youki support was added, there was no release. Also working with wasmedge to ship a statically linked bin. Once those things are in, I'd like to help move forward the release process.-
    - (joe): 1. Fix the release pipeline 2. New release 3. Stable release process
    - (jorge): not sure if the release pipeline is broken. It was for a while, but might be ok. I haven't tried the GH action workflow.
    - (joe): We haven't had a scheduled for containerd-wasm-shims. It has been ad hoc based on features.
    - (brian) I believe the release process is working. There was a bug with empty artifacts. We should be able to push a tag to kick off a release.
    - (djordje): Are you saying that the release pipeline works. I'm seeing small artifacts.
    - (brian): Yes, that bug should be fixed.
    - (djordje): Where can we find a release with artifacts?
    - (brian): We have not done one yet since we broke the projects out into separate crates.

- (djordje): We have been broken and it would be great if we could gather the release artifacts to be able to bundle them together for Docker Desktop.
- (brian): We just need to run a release.
- (djordje): When are we going to have a release?
- (brian): We can release possibly even today. Just push a new tag.
- (jorge): I would like to release in about a week's time. So we can get wasmedge with static linking. We could have 2 release too.
- (joe): Has some changes that would like to get in, but won't block the release. How often do we want to release? Do we need a cadence?
- (jorge): I think we want to have releases as we have new features.
- (djordje): We would like to have releases as we have releases for Docker Desktop.
- (joe): We could do that. Open an issue or something.
- (brian): We could introduce some automation to kick off a release.
- (joe): j & d I know you have mentioned the release in the containerd-wasm-shims repo. Do you need anything?
- (jorge): The Wasm Worker Server release was recently published. I think releases with human trigger makes sense. It would make sense to use the same system in both. I'm happy with fully automated release if there are effective end to end tests. I would like to have tests to make sure it is running with Docker.
- (joe): Brian would you like to kick off the release?
- (brian): You can do it. We just need to ensure the tag names match the crates. We need to have a tag for each crate we are doing a release for. The release tag is namespaced by the crate name.
- (justice): We should do the same in containerd-wasm-shims
- (sven): It is hard to determine what version is installed on a system. It would be great if we could figure out what version the shim is.
- (justice): The code to add args is down in the rust-ext crate. We need to add the arg parsing there.
- (joe): https://github.com/containerd/rust-extensions/issues/120
-

June 20th, 2023 (recording)

Host: David Justice (Microsoft)
Note Taker: TBD
Attendees:

- Jiaxiao (Joe) Zhou (Microsoft)
- Saiyam Pathak (Civo)
- Radu Matei (Fermyon)
- Wilson Wang (ByteDance)
- Brian Goff (Microsoft)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]
- [jsturtevant] Update on OCI artifacts
  - https://github.com/containerd/runwasi/pull/147
  - https://github.com/containerd/containerd/pull/8699
- [joe] Update on wasmtime youki integration - https://github.com/containerd/runwasi/pull/142
-

June 6th, 2023 (recording)

Host: Jiaxiao (Joe) Zhou
Note Taker: TBD
Attendees:

- Comment your names here
- Angel M Miguel (VMware)
- Ismo Puustinen (Intel)
- James Sturtevant (Microsoft)
- Jorge Prendes (Docker)
- Djordje Lukic (Docker)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- **Update on Wasm OCI Artifacts in Containerd: https://docs.google.com/document/d/11shgC3l6gplBjWF1VJCWvN_9do51otscAm0hBDGSSAc/**
- **Ismo: if there's time, discuss the runwasi shared mode design**

Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]

May 23rd, 2023 (Recording)

Host: Jiaxiao (Joe) Zhou
Note Taker: TBD
Attendees:

- Angel M Miguel (VMware)
- Jiaxiao (Joe) Zhou (Microsoft)
- Brian Goff / @cpuguy83 (Microsoft)
- James Sturtevant h(Microsoft)
- Djordje Lukic (Docker)
- Jorge Prendes (Docker)
- Sven Pfennig (Liquid Reply)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above.
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Jsturtevant - Update and on Distribute (and Consume) using OCI artifacts https://github.com/containerd/runwasi/issues/108 and https://docs.google.com/document/d/11shgC3l6gplBjWF1VJCWvN_9do51otscAm0hBDGSSAc/edit
- Joe zhou - how can we add preview2 support in runwasi?
- Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]

May 9th, 2023 (Recording)

Host: David Justice (MSFT)
Note Taker: TBD
Attendees:

- Comment your names here
- Angel M Miguel (VMware)
- Brian Goff (MSFT)
- Djordje Lukic (Docker)
- Sven Pfennig (Liquid Reply)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above.
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Wasm Workers Server shim demo
- Distribute using OCI artifacts https://github.com/containerd/runwasi/issues/108
- Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]

April 25th, 2023 (recording)

Host: Brain Goff
Note Taker: Joe Zhou
Attendees:

- Brian Goff (MSFT)
- Angel M Miguel (VMware)
- Svren Pfennig (Liquid Reply)
- David Justice (MSFT)
- Ismo Puustinen (Intel)
- Jiaxiao Zhou (Microsoft)
- James Sturtevant (Microsoft)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please add your name to the attendee list above. Join the sig-cluster-lifecycle group for edit access.
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- Mixed container pods (https://github.com/containerd/runwasi/issues/64) - Sven Pfennig, 0xe282b0
- (DJ) Wasm apps packaged as OCI Artifacts
    - BCA SIG-Registry OCI support: https://github.com/bytecodealliance/registry/pull/87
    - SIG-Registry meetings Wednesdays 2pm ET: https://us02web.zoom.us/j/86461363574?pwd=WXkvSDRzQm1qemRmRWNmWDhPWUhyQT09&from=addon
- Add your suggested topic here and move this to the line below [firstname lastname, email or @slackname]

Notes

Sven Introduces himself, short talk about KWasm.sh to install runwasi shims on Kubernetes.
Sven: There are some use cases for mixed containerd pods. Wasmedge has Crun integration. WasmEdge is the only shim using Youki.
Angel: What is Youki?
Sven: Youki is a OCI compliant container runtime. The project is written in Rust and enables you to start a Linux container. WasmEdge shim enables libcontainer in the shim to make it OCI compliant.
Brian: detect command to exec is linux binary and run linux runtime instead of wasi runtime. Another option is to have different runtime classes and configs passing down to the runtime. Crun will pass down a runc config rather than a specific runtime config.
Ismo: containerd plugin model
Sven: It's hard to debug if I don't know what container started.
Brian: can get something to work with ctr.
David: Start a proposal of what we really want to do.
Ismo: This could in principle solve in containerd layer
David: Initially we can iterate through it.
Sven: we will get a feature flag to work, and slowly move forward to a proposal and standardization.
Joe: Volunteered to work with Sven on this feature work.
Wasm apps packaged as OCI Artifacts
David: With the upcoming wasm component work, spin shim can pull the artifacts and unwrap everything. What do others think?
James: opened an issue in containerd shims https://github.com/deislabs/containerd-wasm-shims/issues/89 having runtime unpacking things seems strange
David: images should be OS agnostic. There shouldn't be a multi-arch image for wasm.
Brian: One thing to consider is that containerd is looking at supporting VM with confidential computers with certain requirements of needing a sandbox to handle fetching the content of the images. The current thinking to put forward is to allow a shim to implement **poll**. Containerd can tell the shim to poll inside the sandbox. Will retrieve an issue from containerd where they talked about this.
Sven: Solo.io has a wasm spec https://github.com/solo-io/wasm/blob/master/spec/README.md
David: https://github.com/bytecodealliance/registry/pull/87 is a proposal to OCI v1.1 content source in Warg. Like to standardize this across the ecosystem.
James: not clear how these components will get pulled.
Sven: when is the sig-registry meeting?
David: Wed 2pm ET
Ismo: What's the next step after youki libcontainer gets merged?
Brian: OCI spec -> wasm runtime. If we have an abstraction to get rid of the boilerplate is important. Need to look at if we get too much from libcontainer at the moment. Add more testing for the docker scenario.
David: we have wasm cluster-api. This could also be a use case for kwasm.
Joe: Do we have any action items from previous artifacts conversation?

David: 1. Feedback for proposal 2. Open an issue in containerd for wasm artifacts to clarify what we need to do. Get some movement there.

March 28th, 2023 (recording)

Host: Joe Zhou (mossaka)
Note Taker: Danilo Chiarlone
Attendees:

- Ismo Puustinen (Intel)
- Daniel Lopez (Wasm Labs/VMware)
- Angel M Miguel (Wasm Labs / VMware)
- Dan Chiarlone (Microsoft)
- Joe Zhou (Microsoft)
- Wilson Wang (Bytedance)
- James Sturtevant (Microsoft)
- Djordje Lukic (Docker)
- Brian Goff (Microsoft)
- David Justice (Microsoft)

Recurring Topics [timebox to N min]:

- (if you have any… here are some suggestions)
- Welcome any new members or attendees
- Please comment your name to the attendee list above.
- (At the end) Milestone review
- (At the end) Add agenda for next office hours

Open Discussion [timebox to N min]:

- [Ismo Puustinen] Roadmap:
    - Should we plan for some release content? What is needed for "1.0" release?
    - Should we plan for API stability? How does that correspond to the releases?
    - Any other external users of the library than Spin and Slight shims?
- [Ismo Puustinen] Project:
    - Create a collection of test Wasm modules?
    - Try out various OCI features, which ones are supported in libcontainer?
    - What can we test and how much "common" support from runtimes should we expect?
    - Later, when the WASI proposals are accepted, can we make the workloads depend on WASI proposals rather than runtimes?
- [Ismo Puustinen] Document a security process (see https://github.com/containerd/project/blob/main/SECURITY.md)
    - Benchmarking:
    - Test workloads for "real-world" benchmarks?
    - Comparison with WAGI / Knative?
- [Jiaxiao Zhou] Add new maintainers to runwasi
-

Notes

Ismo Puustinen: What is runwasi's current roadmap?
David Justice: There isn't much of a roadmap right now. More than anything, there is tons of operational work to be done – i.e., logging, security, etc.. It's work that goes beyond features. Also, with the fact that we are working on a platform that is still moving (i.e., WASI's fast-dev. model), it's tough to keep up. In my opinion, I don't think we should rush to get a 1.0 out; instead, let's let the environment around us mature, and then we can have that conversation.
Brian Goff: I agree w/ everything that David said. Also, we've got four crates, and improving their maintainability is a must – Like, there are tons of duplicate code that we can clean up. In the meantime, we can just let WASI mature, and, once they are ready, we'll be ready. Other than WASI, we also have to care about Wasmtime, and ensuring isolation there.
Joe Zhou: Ismo – What are your expectations for a 1.0?
Ismo Puustinen: I think we just need to ensure transparency, and openness to ensure security. Other than that, I wanted to ask about the Spin, and Slight shims – Is anyone using them for production?
Joe Zhou: Yes, both of those shims are currently available in AKS, and they were just added to Docker too.
Angel M De Miguel: We, at VMware, have an offering like Slight, and Spin. How can we make that available in AKS?
David Justice: I think that's definitely possible – but we have been very selective about the shims that we make available, so let's chat offline about that.
Brian Goff: I just wanted to add that, in the topic of additional shims, it would be nice if WASI didn't force us to have all these different shims. I'm not sure if that will work out yet, but it's something to keep an eye out for. Like, it would be nice if WASI offered a way for all shims to look the same.
Joe Zhou: I'd also like to add that some runtimes currently have their current hacks for sockets, but that work is currently being worked on upstream, and that could aid w/ shim uniqueness.
Ismo Puustinen: What can we test and how much "common" support from runtimes should we expect?
Joe Zhou: I definitely agree that we need more tests in runwasi, but we're currently blocked by some other PRs that we are waiting to merge in.
Brian Goff: However, there are some things in the OCI spec that we can't really deal w/ (e.g., putting device files in rootfs) or just won't make much sense – at least for now.
Djordje Lukic: Wouldn't wasi-nn need devices?
Wilson Wang: I was in a meeting that discussed that and they don't need access to the physical device, but they do have system dependencies (i.e., libs).
David Justice: Exactly – Here's a link that talks more about it, if you are interested: https://github.com/deislabs/wasi-nn-onnx
Joe Zhou: Cool – Let's move on to documenting the security process.
Ismo Puustinen: I think we should use containerd's security manifest: https://github.com/containerd/project/blob/main/SECURITY.md
Brian Goff: Yes, let's just link that. Plus, we already got security processes routed in when we moved to the containerd org..

Ismo Puustinen: Lastly, I wanted to talk about benchmarking – Like, there are projects in the same space. How are we doing in relation to them? I think this is very important too because it is one one of the main selling points of Wasm vs. normal containers, right?
Daniel Lopez Ridruejo: "I'm a big fan of "Make it work, make it correct, make it fast", I believe this project is still in the "make it work" phase so focus on benchmarks is going to be tricky."
Brian Goff: I don't want to deal w/ benchmarking WASI as a whole, but it could be interesting to benchmark booting up something like Fibonacci w/ runwasi, and runc. Also, that would be good to show how speed is increasing from 0.2 to 0.3, and so forth.
David Justice: I think we should shoot to have benchmarks for things that are meaningful to people, like… Density! How much can I run better w/ the same amount of computational power? That said, I want to ask… Does anyone here have opinions on something that would be valuable to them?
Angel M De Miguel: It would be hard to have a benchmark that correctly targets WebAssembly vs. the actual shim implementation, which might not be too valuable.
Wilson Wang: I have a question of density. It's true that w/ Wasm we can pack more instances w/ the same amount of memory, but there are constraints on the Kubernetes side too – Like, only supporting something like 2000 pods. Are we taking that into account in our design?
David Justice: I am so happy you brought that up. That's such a great problem to have. I'm not sure, though – If anyone has any ideas on how we can make things run better without taking so much time from kubelet, I'd like to hear more about it.
Joe Zhou: I will create an issue on runwasi for it.
Brian Goff: The long-running shim would really help with this – long-running shim = sharing the hypervisor across all pods, which should drastically reduce overhead.
Angel M De Miguel: Is the long-running shim the 'shared mode' mentioned in the docs?
Brian Goff: Yes.
Joe Zhou: There are other optimizations we can make too. With that, let's move on! The last item in our agenda is promoting contributors to maintainers.
Dan Chiarlone: I think it's important to ask if folks want to be maintainers too.
Ismo Puustinen: I think it would be good to get folks from Youki to come here and talk about their project too, because it obviously relates a lot to runwasi's work, but it might be tough because they are in Japan.
Daniel Lopez Ridruejo: I think we could maybe alternate the meetings.
Joe Zhou: We could have meetings in the afternoon too for people in Asia. Ok – Let's talk about the agenda for the next office hours. Do we want to maybe talk about the Youki PR?
Brian Goff: I hope not – I want us to merge that ASAP.
Dan Chiarlone: I might give benchmarking runwasi vs. runc a try, could bring my findings for next time.