



New issue



Set memory.min or memory.low, without memory.high #131077

Open

Labels

needs-triage

sig/node



acurtiz opened 6 hours ago



Problem

In kubernetes, the container memory requests aren't propagated to the container runtime. This makes sense because they're not *directly* meaningful to any runtime setting (unlike e.g. container memory limits which translate to `memory.max` on cgroupv2).

[KEP-2570: Memory QoS](#) aims to change this by setting `memory.min` on relevant cgroups based on the corresponding k8s memory requests. This KEP has stalled out due to also tweaking `memory.high`, which has undesirable effects.

Question

Can we start passing container memory requests to the container runtime?

I see two potential paths:

- Path 1: Begin setting `memory.min` on relevant cgroups based on memory requests. This is a part of KEP-2570. Critically, we wouldn't set `memory.high` ever. This alone appears to achieve the goals of the KEP. It's unclear to me if or why we must set `memory.high` together with `memory.min`. Must we?
- Path 2: Begin setting `memory.low` on relevant cgroups based on memory requests. Don't set `memory.low` or `memory.high` at all. Less guarantees than with `memory.min`, but also should be less disruptive from today's behavior, and still a behavior improvement. I believe this should be completely backwards compatible to when/if we support `memory.min` in the future.

Is there any appetite for this? Am I missing some nuance that makes these paths non-starters? KEP-2570 goes far, and it feels like we can get partial benefit more easily by reducing the ambition.

Why?

1. Achieve partial or even full benefits that KEP-2570 aims to achieve by de-coupling some parts of the KEP from others, and tweaking others.
2. I have an on-node plugin which tracks CPU and memory requests of containers on the node, which it uses as input to actuate changes elsewhere on-node. This plugin primarily uses a combination of NRI and directly querying CRI to achieve this. However, it needs to know memory requests, which it has to fetch either from kube-api (external to node), or the local authenticated /pods endpoint (unreliable, security implications). Would be great to just get all needed info from NRI/CRI directly.



k8s-ci-robot added **needs-sig** **needs-triage** 6 hours ago



k8s-ci-robot 6 hours ago

Contributor



This issue is currently awaiting triage.

If a SIG or subproject determines this is a relevant issue, they will accept it by applying the `triage/accepted` label and provide further guidance.

The `triage/accepted` label can be added by org members by writing `/triage accepted` in a comment.

► Details



tallclair 4 hours ago

Member



Related: [kubernetes/enhancements#4113](https://github.com/kubernetes/enhancements/issues/4113)

/sig node

This looks like a good topic for the sig-node weekly discussion, if you're able to join:

<https://github.com/kubernetes/community/tree/master/sig-node#meetings>



k8s-ci-robot added **sig/node** and removed **needs-sig** 4 hours ago



Add a comment

H B I | ≡ <> 🔗 | ≡ ½≡ 📋 | @ ↗ ↶ 📌

Write

Preview


Use Markdown to format your comment

 Paste, drop, or click to add files

 Close issue

▼

Comment

 Remember, contributions to this repository should follow its [contributing guidelines](#), [security policy](#) and [code of conduct](#).

Metadata

Assignees

No one assigned

Labels

needs-triage sig/node

Type

No type

Projects

No projects


Milestone

No milestone


Relationships

None yet

Development

 Open in Workspace

No branches or pull requests

 **Subscribe**

You're not receiving notifications from this thread.

Participants

