

December 05, 2018:

No one joined, no agenda.

March 15, 2016

Agenda

- 1) Kubernetes node team 1.3 roadmap:

https://docs.google.com/document/d/1Uw0NsTFIhI3EPiIHxNuWIXqOV0Siz__1G2IreEPJdKQ/edit

@Dawn: Priority Define

[P0] Blocker issue.

[P1] Nice to have.

[P2] Help wanted or extra engineering resource.

@resouer: What kind of logging handling?

@Dawn: System log and container log.

@Random-Liu: What is the difference between runtime conformance test and docker validation?

@Dawn: Mostly runtime conformance test + docker specific test.

@feiskyer: Client/Server runtime interface?

@Dawn: Good for modulizing. But most concern - overhead.

@resouer: The overhead is important, there is already an extra layer.

@Dawn: Start/stop latency should be acceptable, but for continuous maintenance latency is unacceptable. **Benchmark data is needed.**

@Dawn: Who's reporting container status in hyper?

@feiskyer: hyperd -> hyperstart (Monitoring containers in VM)

@chenye: 1.3 Deadline?

@Dawn: 3 - 4 months each release.

@chenye: Does disk management include volume management?

@Dawn: Sure. Already support - Volume + Overlayfs + Aufs.

March 1, 2016

Agenda

- 1) hyper integration
- 2) huawei docker conformance test
- 3) kubernetes 1.3 release brain storming

Signode Meeting Note:

<https://docs.google.com/document/d/1Ne57gvidMEWXR70OxxnRkYquAoMpt56o75oZtg-OeBg/edit>

1.2 Status Update (@dchen)

For the whole team, some issues for deployment object and HPA scale. Some AWS issues for k8s 1.2.

For node side all block features are finished. K8s 1.2 will go with docker 1.9.1.

Current problem for docker 1.10: Most forward compatibility issues are solved. The most important issue left - docker image format is changed.

Docker 1.10 Validation issue: <https://github.com/kubernetes/kubernetes/issues/19720>

Huawei Container Runtime Conformance Test (@chenye)

- 1) First PR is merged.
- 2) Next: Image pull/push test.
- 3) Problems: How to test rkt? Test different environment.
- 4) Parallelise rkt integration and runtime conformance test.
- 5) To be continued?

Hyper Integration (@feiskyer)

- 1) Kind of blocked by gce account

1.3 Release Brain Storm

- 1) @hongchao: Over-commit? -> In 1.3 roadmap today, but not finalized. In fact we can do over-commit today, but reliability and QoS should be guaranteed. (@dchen)
 - a) Detect out of resource
 - b) Decide which pod should be killed
 - c) Well-defined eviction policy is needed.

Q&A

@dchen: Etcd v3.0 alpha version in April; stable version in May or June; Try to ship with kubernetes 1.3.

@xiang:

- New node metric api merged? Yeah.
- Is the new metric api added for better scheduling? No, necessary information for scheduling is available on kubelet side from v1.0.
- Hardware level metrics?

@feiskyer: Pod level data in cadvisor? At cgroup level, we could do that. But at API level, it is still in discussion. Internally shareable resource between containers is very powerful but quite complex. If customers don't ask for that now, we may want to defer it. @dchen: How to define extra override of rkt and hyper in api, should it be visible for user?

Related issue: <https://github.com/coreos/rkt/issues/1788>

Feb 16, 2016

Feb 2, 2016

Agenda

proposed by @dchen:

- 1) Introduction - 10mins
- 2) Test suite of docker as container runtime - 10mins
- 3) rktnetes outstanding issues - 20mins
- 4) hyper integration - 20mins

any other suggestion?

Notes

Self introduction:

Huawei: Hardy Simpson(CCE, Scheduler), Chenye Liang (Docker Conformance Test, Security), Victor - Hu Ruifeng (Scheduler), jiangyaoguo (Scheduler, same team with Victor).
CoreOS: yifan(rktnetes), hongchao-deng(Scheduler)
Hyper: feiskyer(Hypernetes), Laijs - Laijiangshan

Hongchao: Usage-based scheduling -> Dawn: The information is enough, maybe isolation is needed.

- 1) Docker container runtime conformance test. @chenye

@chenye: Mainly himself does this now. Status update with Lantao and Philip and track the progress with wiki <https://github.com/liangchenye/kubernetes/wiki/E2E-test>

- Decided to add 2 kinds of test:
 - Black box test: Has done, wait for reviewing.
<https://github.com/kubernetes/kubernetes/pull/20415>
 - Runtime interface test => Still in process.
- Rkt integration with K8s. @Yifan is driving, needs help for e2e test, rkt container runtime test, network related issue etc.

@feiskyer: What's the difference between e2e test and node conformance test?

- K8s as a black box (e2e) vs. Kubelet or node as a black box (node conformance test).
- In the long run: node conformance test, system and prerequisites version etc.
 - Help customer figure out problem and build up private cluster.
 - Node has the most work for debugging => For experience of Google, it's tough even in google with plenty of tools. => Node conformance test may help.
- @feiskyer: Will it be part of CI? => Yes, we have that already in internal jenkins.
- @Hardy Simpson: Run after node failure to diagnose the node? => That's future vision. Now it's only part of e2e integration test.

2) Rktnetes outstanding issues. @yifan

- Doc:
https://docs.google.com/document/d/132fMB60poMCcejf8l82Vxpynff8_BRycZeOn24jn8pc/edit?ts=56b248a1
- DNS problem => Done, the master node is still using docker.
<https://github.com/kubernetes/kubernetes/pull/20302>
- CoreOS master reboot automatically, or can't ssh login. => Only on GCE. May be base image problem.
- CAdvisor integration. => May need Huawei folks' help after rkt 1.0
- E2e test against the cluster. Run and fix test.
- Kind of blocked by CNI PR.
- @feiskyer: All container runtimes need to support CNI? => Hopefully. => @feiskyer: Mentioned on sig-network, policy-based network may change current design.
- At least, rkt could cooperate with Huawei on runtime conformance test now.

3) Hyper integration @feiskyer

- Integration plan with K8s:
 - Keep a separate branch for K8s 1.2 + Hyper.
 - Try to push the runtime integration to upstream, related PR will be reviewed by @k8s-sig-node.
- Current status, e2e test fail a lot. Main issues:
 - Portforward needs the container runtime to be a namespace, which hyper doesn't support now.
 - CAdvisor integration is hard, because it is hard to run CAdvisor in a VM and get information from the host.
 - CPU and memory must be set when starting hyper, which is different from docker.
 - Maybe use limit range? It's hard to force the user set the resource requirement now.
 - May force the user to set resource limit if using hyper as runtime. Reject on apiserver: Change the api validation to force the resource requirement. Or reject on node.
 - API version parse failure.
- Runtime interface?: @dchen: Hyper support is very important.
 - => Container technology is new and popular but still has issues like security issue
 - => Hyper as an extra layer of security is very important.
 - => However, rkt is also a very important use case, so rkt integration should not be blocked now.
 - => Docker design is not align with kubernetes, while rkt is align from the beginning.
 - => For example, All docker operations go through daemon: if daemon died, container died; resource limit of container is hard, for example container keeps sending request to docker daemon.

- @feiskyer: OCI?
 - Hyper is based on runV (<https://github.com/hyperhq/runv>) => @Laijs is following this. => @Vish is the OCI representative on google side.
 - Our attitude on OCI: Follow the status and it is what we expect now => Keep exchanging requirement. => Check result.
 - Will kubernetes supports runC => systemd+runC or containerd+runC => systemd+runC is not much different from rkt? runC has separated the image management, rkt can do this too.
 - @yifan: What's the meaning of separate image management with runtime. => jar file, tar file, aci image, docker image or something => container is just a sandbox (docker, kvm, cgroup+namespace) =>
 - @feiskyer: Like different images to OCI format and pass to the runtime? => For hyper, pull image is in hyper like docker, can be separated, but image format should be fixed. => This is just the future vision and moving target.

@Victor: One requirement - Kubelet starts pod by running docker api. Separately pod Create and Start? => A new pod state created and scheduled but not running. => @dchen: This is API discussing. Briant strongly against. Node team prefers. => @Victor: Our use case, parallel deploy and sequentially start up containers. Init container issue: <https://github.com/kubernetes/kubernetes/issues/1589>

Note: Record next time!!!!