[sig-arch] Code Organization Subproject

When do we meet?
Every 2 weeks on Thursday at 11 AM PT / 2pmET / 18:00 UTC
Zoom: https://zoom.us/j/845605479
Slack: #k8s-code-organization

Future Topics:

- pain points in managing multiple repos in http://github.com/kubernetes-csi/ - needs a representative from SIG Storage

Agenda

Month Day, Year (recording)

Host: TBD
Note Taker: TBD
Attendees:

- *Add your names here*

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- AI review

Open Discussion [timebox to N min]:

- Add your suggested topic here [firstname lastname, email or @slackname]

Sep 15, 2022 - 11 am PT / 2 pm ET

Host: dims
Note Taker: TBD
Attendees:

- *Add your names here*

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- AI review

Open Discussion [timebox to N min]:

-

Jan 20, 202github1 - 11 am PT / 2 pm ET

Host: dims
Note Taker: TBD
Attendees:

- *Add your names here*

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- AI review

Open Discussion [timebox to N min]:

- Do we need another github org for forks? What about more-but-smaller libs for internal use?
    - Example : https://github.com/kubernetes/utils/pull/232
- [Sergey] https://github.com/kubernetes/kubernetes/issues/107442 types files difference. Is it worth to clean it up?
- [Sergey] CRI API: code organization forcing us to add new methods to v1alpha2: https://github.com/kubernetes/kubernetes/pull/104620
    - Also going forward, is there a precedence to test against earlier versions of API (beyond version skew tests?). For https://github.com/kubernetes/kubernetes/issues/107190 need to ensure kubelet can work with any version of v1 implementation, not just current and v-1.
- Add your suggested topic here [firstname lastname, email or @slackname]

Jan 6, 2021 - 11 am PT / 2 pm ET

Host: dims
Note Taker: TBD
Attendees:

- *Add your names here*

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- AI review

Open Discussion [timebox to N min]:

- Add your suggested topic here [firstname lastname, email or @slackname]
- [Sergey] minor qq: diff between
    - pkg/apis/core/types.go
    - staging/src/k8s.io/api/core/v1/types.go
    - liggitt: intentional for internal/external files to be able to diverge. Docs on external are more important, intended for end users and used for generated API doc.

Internal may have more fields as it can bridge between multiple different external versions.
    - There are verification scripts that validate that conversion from one to another can happen
- CRI v1 and v1alpha2 API
    - https://github.com/kubernetes/kubernetes/pull/107192
    - probably more a question for node and API review (e.g. https://github.com/kubernetes/kubernetes/issues/107190)
        - We will need to think about policies for gRPC API based on what TimH comes back to us with.
        - Is scenario consuming grpc API as compiled version?
            - liggitt: we version the repo… so you can get k8s.io/cri-api@v0.22.0, etc
        -
    - Do we need to split CRI API out from staging so things like cadvisor can use it?
        - liggitt: seems plausible… very small, very self-contained, very slow-moving
        - sergey: csi api had some difficulties coordinating pulling in API updates
        - dims: cri API seems slower-moving than csi
        - sergey: there have been windows-related changes
        - liggitt: I think there are more things in the pipeline (windows changes, checkpoint changes), need to make sure there's a reasonable workflow for making changes in an external repo and pulling them back into k/k
- docker/distribution dep
    - tried replacing with distribution/distribution (https://github.com/kubernetes/kubernetes/pull/107211)
    - pulled in ~20 additional deps (a lot transitively)
    - our use of docker/distribution is *very* small (just image parsing), so prefer extract/fork that and drop the rest of the dep
    - Options: where we can fork part of what we use.
        - third_party/ in k/k (not possible because some of the code that needs this in staging/)
        - k8s.io/utils (previously we had inotify)
            - Dims will prototype
- GOPATH → go module work
    - go1.18 supports workspace mode (makes it easier to tell go that there are some related modules)
    - makefile, gengo, and code generators are the biggest remaining item that doesn't work in module mode
    - thockin looking at reworking gengo/generators on top of golang.org/x/tools/go/packages to work in module mode
    - umbrella issue: https://github.com/kubernetes/kubernetes/issues/82531
        - go1.18 adds workspace mode which helps multi-module dev

        - communicating with go team to get rough edges with workspace mode (like supporting vendor) ironed out
        - communicating with go team to identify blockers for dropping GOPATH
        - Tim surveying work needed to adjust generators on top of golang packages library
        - AI: Tim and Jordan to update this issue when they get a chance, identify code generator work that can be parallelized
- Golang
    - There is an existing PR for 1.18beta1 open. Policy question here, do we ship a version of k/k with a beta of the golang compiler?

September 2, 2021 - 11 am PT / 2 pm ET

Host: TBD
Note Taker: TBD
Attendees:

- *Add your names here*

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- AI review

Open Discussion [timebox to N min]:

- depstat alternate approach - https://kubernetes.slack.com/archives/CHGFYJVAN/p1629795118067100 [@arsh] https://github.com/RinkiyaKeDad/gomodgraph-sixteen/blob/master/cmd/root.go#L75-L140
- Moving components out of k/k [@arsh, @nabarun]
    - kubeadm out of k/k - https://github.com/kubernetes/enhancements/pull/1425
- Add your suggested topic here [firstname lastname, email or @slackname]

August 19th, 2021 - 11 am PT / 2 pm ET

Host: Jordan Liggitt
Note Taker: Nabarun Pal
Attendees:

- *Jordan Liggitt*
- *Nabarun Pal*
- *Lubomir I. Ivanov*

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- AI review

Open Discussion [timebox to N min]:

- Go 1.17 changes to $GOPATH

    - The changes were never made!
    - https://github.com/kubernetes/kubernetes/issues/82531#issuecomment-897152418
- Tracking issue for go module building (building with GO111MODULE=on) https://github.com/kubernetes/kubernetes/issues/82531
    - Generator fix to boilerplate path in module mode https://github.com/kubernetes/kubernetes/pull/103847
    - defaulter-gen fix https://github.com/kubernetes/kubernetes/pull/104330
    - WIP fixing module build issues - https://github.com/kubernetes/kubernetes/pull/99226
        - last two commits work around gengo use of go/build that is 100x slower
        - gengo change that adds warning to the non-module path https://github.com/kubernetes/gengo/pull/208/files#diff-59ef411aab527cfdf48a91aa8619324b2866a6d09b030d8370ba3380ef9880acR326
    -
- dependency stats presubmit in k/k uses `go mod graph`
    - go1.17 denormalizes all transitive dependencies into the root go.mod file
    - https://github.com/golang/go/issues/47648
    - issue to switch dependency stats to use go list (or external tool like https://beta.pkg.go.dev/github.com/loov/goda) instead of go mod graph
        - https://github.com/kubernetes/kubernetes/issues/104461
    - AI: after ^, switch our go.mod files to go 1.17 (can be done after starting to build with go 1.17)
- go 1.17
    - WIP https://github.com/kubernetes/kubernetes/pull/103692, landing soon
- Would k/k keep on vendoring after moving to go modules?
    - Yes. Dependencies can bring in a lot of different changes that may not be very apparent.
    - With a vendor dir, it makes it easier to vet those changes when they are being PR'ed.
    - Would need to answer the question "can we still build if an original dependency source becomes unavailable and the go module proxy does not have a cached copy any more"

~~Jun 10th, 2021 11 am PT / 2 pm ET~~

~~Host: dims~~
~~Attendees:~~

- ~~—~~

~~Agenda:~~

- ~~Open Agenda~~

May 27th, 2021 - 11 am PT / 2 pm ET

Host: dims
Attendees:

- Gautier Delorme
- Arsh Sharma
- Navid Shaikh
- Arsh Sharma
- Eddie Zaneski
- Dims
- Mike Brown
- Kirsten G

Agenda:

- Announcement: We now have k8s-infra-code-organization kubernetes.io group for managing secrets under google secrets manager, see https://github.com/kubernetes/k8s.io/pull/2085/files
- Separating kubectl from k/k tree
    - Different release cadence as that of k/k (does not infer/require order of release)
    - Vendor kubectl in k/k just like any other dep
    - Need a plan to write this down
- Forbidden dependencies
    - PR: https://github.com/kubernetes/kubernetes/pull/102309
    - [gautier] Have a way to specify we don't want specific dependency and we don't want them to come back
    - [gautier] Have a way to specify what deps we have today but we don't want them in the future
    - [dims] We shouldn't go back to the older version of something we are using latest version of
    - [dims] When new deps are coming up, prefer to have them pinned using tags instead of shas for maintenance overhead
- Are there new changes coming up from containerd
    - [mike] nope
- have pings opened up for updating Azure, GCP, vsphere, AWS sdk(s) for 1.22 in their respective channels

Apr 29th, 2021 11 am PT / 2 pm ET

Host: dims
Attendees:

- Nabarun Pal
- Dims
- Arsh Sharma

Agenda:

- Depstat scripts in k/k
-

Apr 1st, 2021 11 am PT / 2 pm ET

Host: dims
Attendees:

- Arsh (Slack @arsh)
- Dims
- Navid
- Nabarun
- Parris (Slack @Parris4DurhamNC)

Agenda:

- [arsh] - demo of the tool to evaluate dependency updates
    - Code repo: https://github.com/RinkiyaKeDad/dependency-analyzer-poc
    - --feedback on demo--
    - Use consistent wording for depstat
    - Check how kubectl help message is formatted and refer
    - Help for individual commands
        - Add examples section per command
        - Parris Lucas (@GrooveCS): Support adding the following
            - Auto-Completion
            - Documentation Review
            - Unit Testing Review
    - Fix the shorthand for --json flag
    - For ‘cycles’ command
        - Remove extra line in individual listing
        - Add --json flag
            - Key: ‘cycles’ and value: array
    - For ‘graph’
        - Add flag for overwrite
        - Add output file flag
    - For ‘List’ command
        - Sort the output
        - Add --json flag
    - For ‘Stats’
        - Sort the output
    - Wrap the output to 80 chars
    - Write down examples for most common use cases in command usage and repo README

    - Auto-completion
    - ----
    - Next steps:
        - Integration in k/k
            - Add depstats-file representing the current stats of dependencies
            - Add hack and verify scripts which manage depstats-file (examples: verify-gofmt , update-gofmt in hack/ , hack/update-vendor.sh)
            - Anytime the depstats-file is updated, flag it
        - PoC the update/verify scripts
        - Ask about what code repository to use

Mar 18th, 2021 11 am PT / 2 pm ET

Host: dims
Attendees:

- dims
- arsh
- bentheelder
- liggitt
- neolit123
- palnabarun

Agenda:

- What do we do for 1.22?
    - go build no longer depends on GOPATH (required by go1.17?)
        - https://github.com/kubernetes/kubernetes/issues/82531
        - Build scripts (sig-release? thockin?)
        - Code generation (sig-api-machinery? sttts?)
    - Protobuf / grpc versions (etcd and gogo-protobuf)
        - https://github.com/kubernetes/kubernetes/issues/93320
        - https://go-review.googlesource.com/c/protobuf/+/300869/ added minimal support for gogo-generated messages
            - github.com/golang/protobuf@v1.5.1+
            - google.golang.org/protobuf@v1.26.0+
            - go.etcd.io/client/v3@v3.5.0-alpha.0+
            - google.golang.org/grpc@v1.32.0+
            - WIP: https://github.com/kubernetes/kubernetes/pull/100488
    - Generate graphs for why something ended up in a go.sum file:
        - https://gist.github.com/dims/3a416e0d83741cf8f31943f99cf1c7ee
        - Example on how containerd pulls k8s.io/kubernetes:

        - https://www.evernote.com/l/AZy-c1wxNxJE94H1TZaGfL0pQwvVodn1HvM
        -
- LFX Internship - Arsh
    - Print the longest chain for each dependency as you visit them in the DFS call to debug.

Feb 18th, 2021 11 am PT / 2 pm ET

Host: dims
Attendees:

- dims
- Liggitt
- Arsh Sharma

Agenda:

- Go-1.16 is released. We can fix up client-go now.
    - https://github.com/kubernetes/kubernetes/issues/98267
- Issues Review
- PR Review
- Dependency analyzer POC - What do we need to report
    - Number of dependencies
    - Max depth of dependencies
    - Dependencies in the dependency tree that we don't vendor
- [lubomir] should we host the kubelet systemd specs and DEB / RPM package specs in k/k.

Jan 21th, 2021 11 am PT / 2 pm ET

Host: dims
Attendees:

- dims
- liggitt

Agenda:

- Client-go - consider retracting old pre-module versions
    - https://github.com/kubernetes/kubernetes/issues/98267
- protobuf updates
    - https://github.com/etcd-io/etcd/issues/12197#issuecomment-764400685
    - kubernetes side
        - verify performance improvements make it possible to switch off of gogo
        - prep reworking our protobuf generation code to no longer depend on gogo

        - compile list of which dependencies we need to chase to switch away from gogo
    - kubernetes uses gogo-specific generation rewriting, might be difficult to transition
    - assuming k8s can switch off of gogo, we *still* have to pick up new versions of all protobuf-containing dependencies generated with modern protobuf library versions. coordinating this will be *hard*
        - etcd
        - others?
-
- Review Issues and PRs with label “area/code-organization”
    -

July 23rd, 2020 11 am PT / 2 pm ET

Host:
Attendees:

- Jordan Liggitt
- Nikhita Raghunath
- Walter Fender
- Andrew Sy Kim
- Alessandro
- Shreyas Sreenivas

Agenda:

- Renaming the library-go package - https://github.com/kubernetes/kubernetes/pull/92507#issuecomment-660956984
    - commented in the issue, will discuss there
- Moving code from `pkg/kubelet/apis` to `k8s.io/kubelet/pkg/apis` - https://github.com/kubernetes/kubernetes/pull/92527, https://github.com/kubernetes/kubernetes/pull/92985, https://github.com/kubernetes/kubernetes/pull/92632
    - AI: liggitt to review
- [wfender] Moving features (public vs private)
    - https://github.com/kubernetes/kubernetes/pull/92871 (fixup moving proxy features out of apiserver)
    - https://github.com/kubernetes/kubernetes/pull/92201 (proposal to relocate CCM)
    - https://github.com/kubernetes/kubernetes/pull/93068 (move of feature names to API)
    - Defining features in multiple places
        - already did this for apiserver features + kube features
        - what is the primary owner for a given feature?
        - should the feature definition live in that component's exported repo?
        - how do we register those into the feature gate used in a given process?

- Handling registration errors
  - in the past: silent error (not registered == Enabled() would return false)
  - now: runtime error / panic
  - ideal: compile error
    - would probably require registration to shared global with init()
    - Already registering into shared global via init(): https://github.com/kubernetes/kubernetes/blob/master/pkg/features/kube_features.go#L650-L652
    - Defined in https://github.com/kubernetes/kubernetes/blob/master/staging/src/k8s.io/apiserver/pkg/util/feature/feature_gate.go#L28
  - If all features are defined in k8s.io/api/features, and all register on init(), all features pollute CLI help
    - Is there a pattern we can follow to init/register features in files where they are used?
    - **Proposal:**
      - **Easy path: RegisterAll() peer to feature definitions**
      - **clients that don't care about curating can just register all**
      - **clients that do care can curate**
  - Do we want to just move feature names referenced from staging repos to k8s.io/api, or move all?
    - k8s.io/apiserver
    - things referenced from k8s.io/cloud-controller-manager
    - things referenced from CSI controllers?
      - https://github.com/kubernetes-csi/external-provisioner/blob/master/go.mod#L20
    - **limiting the number of features exposed via k8s.io/api seems good: exposing default feature enablement values introduces skew issues when consumed by external libraries**
  - If we auto-register features on import in init(), can a particular cloud controller build adjust the default enablement state for a feature?
    - is this desirable? make sure consumers can do what they want changing/enabling features
  - Do we want to prevent client-go / cli-runtime / kubectl from referencing/using features
    - today it cannot reference them because they live in apiserver
    - we don't have mechanisms (config, cli) to control features in client libraries
    - **let's set up import guards to prevent accidental use from client-go / cli-runtime / kubectl until we mean to and have a good story for users configuring it**

June 25th, 2020 11 am PT / 2 pm ET

Host: dims
Attendees:

- Jordan Liggitt
- Davanum Srinivas
- Mo Khan
- Walter Fender

Agenda:

- [dims] Initialize k8s.io/library-go staging repository
    - example guidelines for utils repo: https://github.com/kubernetes/utils/#criteria-for-adding-code-here
    - outstanding questions:
        - name (library-go, confused with client-go? maybe api-helpers?)
        - structure (probably avoid sig-named packages, tend toward component-oriented packages?)
            - scheduling
            - authentication
            - authorization/rbac
            - etc
        - need clear requirements/guidelines for code living here
            - e.g. depend only apimachinery/api/client-go and transitive deps via those repos
            - accompanied by import guards to make sure we don't accidentally pick up additional deps
- [walter] module dependency graph helper script
    - https://github.com/kubernetes/kubernetes/pull/92445
- [walter] CCM making steady progress to staging
    - CSI is part of cloud provider extraction, but not specifically part of CCM extraction
- [dims] Walk through open issues/PRs?
    - https://github.com/kubernetes/kubernetes/pulls?q=is%3Aopen+is%3Apr++label%3Aarea%2Fcode-organization+
    - https://github.com/kubernetes/kubernetes/issues?q=is%3Aopen+label%3Aarea%2Fcode-organization+sort%3Aupdated-desc
-

June 11th, 2020 11 am PT / 2 pm ET

Host: dims
Attendees:

- Andrew Sy Kim
- Dims

- Jordan Liggitt
- David Eads
- Benjamin Elder

Agenda:

- klog/v2 + cadvisor bug + runc dev branch intersection
    a. what happened?
        - cadvisor regressed cpu detection
        - klog/v2 switch started updating master of many dependencies, then propagated into kubernetes/kubernetes
        - kubernetes/kubernetes got stuck with a bad version of cadvisor, and could not pick up a fix before completing the klog/v2 migration (all systems with SMT disabled and ARM were broken on kubernetes/kubernetes master)
    b. where are we now?
        - cannot roll back to good version of cadvisor because of klog/v2
        - rolling forward to currently fixed version of cadvisor requires
            - picking up an untagged version of runc (which we've encountered problems with before)
            - picking up new grpc
            - picking up new etcd
        - recommendation: rollback to last known good cadvisor
    c. implications for code-organization
        - klog/v2 rollout had a lot of ripples, required moving 5-6 deps in sync
        - possible alternatives:
            - move deps to interface that can connect to klog/v1 and klog/v2
            - move deps to std go logging, which we wire to klog
        - AI: give sig-node information about options for rolling cadvisor back to last known good version and temporarily shimming klog/v1 to klog/v2
    d. implications for sig-node / sig-release
        - no presubmit or release-blocking CI jobs that caught this
            - things we want to work should have CI
            - what is the project stance for architectures we release without CI?
            - some architectures have been rejected
        - cadvisor:
            - switch to new CPU detection without fallback seems unbaked
                - AI: dashpole: can we fall back to the previous detection method if we get clearly nonsense results from the new detection method
            - time-bound fix forward
- AI: dims to follow-up on cadvisor issue for shimming klog v1/v2 so rollback is possible.
- AI: follow-up with SIG Arch on whether all the “supported” architectures should pass conformance
- AI: follow-up with SIG release on criteria for supporting specific CPU architectures.

May 28th, 2020 11 am PT / 2 pm ET

Host: dims
Attendees: Riaan Kleinhans (riaan@ii.coop)

-

Agenda:

- Gogo protobuf
- Blocked PRs
    a. https://github.com/kubernetes/kubernetes/pull/90976
    b.

Apr 16th, 2020, 11 am PT / 2 pm ET

Host: Andrew Sy Kim
Note Taker:
Attendees:

- Andrew Sy Kim
- Jorge
- Walter Fender
- Dims
- Jordan Liggitt

Agenda:

- [andrewsykim/jorge] Staging e2e framework KEP (WIP): https://docs.google.com/document/d/1QgYEmdrdIirRCGBi2j55yb88WCZOGBqOJw7xj3s05PI/edit#
- [dims] Dockerless KEP is closer to getting merged
- [dims] klog v2 PR to support structured logging is hairy:
    - https://github.com/kubernetes/kubernetes/pull/90183
    - [Jordan] Is scalability issues w/ klog + distroless a blocker?
    - [dims] can't root cause scalability issues
    - [dims] there are performance benchmarks in klog but not related to performance issues when using distroless
    - [liggitt] do the performance benchmarks cover all the different ways klog can be configured?
    - AI (dims): track down where new dependencies are introduced

Apr 2nd, 2020, 11 am PT / 2 pm ET

Host: TBD
Note Taker:
Attendees:

- Andrew Sy Kim
- Jordan Liggitt
- Mo Khan
- Aaron Crickenberger
- Dims
- Lubomir

Agenda:

- [andrewsykim] https://github.com/kubernetes/kubernetes/issues/74352 - moving test/e2e/framework to staging is p0 for testing-commons subproject (sig-testing)
    - k8s.io/e2e-framework ?
    - Ben's KEP for builds without in-tree cloud providers https://github.com/kubernetes/enhancements/blob/master/keps/sig-cloud-provider/20190729-building-without-in-tree-providers.md
- [dims] hcsshim update - https://github.com/kubernetes/kubernetes/pull/89710
- [dims] cri-api split - https://github.com/kubernetes/kubernetes/pull/87493
    - Related - "KEP proposing to move streaming library to a new home" https://github.com/kubernetes/enhancements/pull/937
- [lubomir] kubeadm split - https://github.com/kubernetes/enhancements/pull/1425
    - sync up w/ SIG Release

Mar 19th, 2020, 11 am PT / 2 pm ET

Host: Andrew Sy Kim
Note Taker:
Attendees:

- Andrew Sy Kim
- Jordan Liggitt

Agenda:

- needs review:
    - https://github.com/kubernetes/kubernetes/pull/88867
    - https://github.com/kubernetes/kubernetes/pull/88631
- <new topics>
- cutting links to docker/docker and docker/libnetwork dependencies - dims
    - in progress:
        - jsonlog: https://github.com/kubernetes/kubernetes/pull/89013
        - runtime/cpu: https://github.com/kubernetes/kubernetes/pull/89182

        - term: https://github.com/kubernetes/kubernetes/pull/89159
        - migrating to moby/ipvs: https://github.com/kubernetes/kubernetes/pull/89116
            - AI for testing/network: set up ipvs CI job
    - 1.19:
        - isolate remaining dependencies to dockershim
            - import guards, "dockerless" build tag
            - POC: https://github.com/mattjmcnaughton/kubernetes/commits/mattjmcnaughton/poc-compiling-kubelet-wo-docker
        - update CI jobs using cri/containerd to build dockerless
    - deprecate/remove dockershim
- fyi: cadvisor slicing, isolating storage/cri integrations in progress - liggitt
    - isolated binary dependencies in https://github.com/google/cadvisor/pull/2386
    - writeup at https://docs.google.com/document/d/1NTEIdBln3U9xKaFUarEnfAxT8LUF2Y-ATuWqA1npovw/edit

Mar 5th, 2020, 11 am PT / 2 pm ET

- **Cancelled due to code freeze**

Host:
Note Taker:
Attendees:

- <add your name here>

Agenda:

- [andrewsykim] Need to break dependency to GenericControllerManagerConfig in cloud-controller-manager. Options:
    1) Make a duplicate and move to cmd/cloud-controller-manager
    2) Move GenericControllerManagerConfig out of k/k, but where? Maybe k8s.io/component-base/controllermanager/config or k8s.io/controller-manager/config?
    3) ???
- [andrewsykim] where can we move generically useful controllers util methods in pkg/controller?
    - required as part of https://github.com/kubernetes/kubernetes/issues/74352 and https://github.com/kubernetes/kubernetes/issues/81172
    - some examples:
        - AddOrUpdateTaintOnNode
        - ReplicaSetsByCreationTimestamp
        - FilterActivePods

    - so far we've been duplicating these methods, some functions may get duplicated up to 3 times, is that acceptable? Should we find a better home for these methods?

Feb 20, 2020, 11 am PT / 2 pm ET - **CANCELLED**

- Cancelled due to lack of agenda items

Feb 6, 2020, 11 am PT / 2 pm ET

Host: Dims
Note Taker:
Attendees:

- Mo Khan, VMware, @enj
- ...

Agenda:

- stats for dependency updates - @liggitt
    - it would be really helpful to know things like the following for each direct dependency k/k has:
        - max depth of transitive chain via that dep
        - total number of transitive deps via that dep
        - unique transitive deps via that dep (what is it pulling in we wouldn't otherwise have)
        - number/degree of shared transitive deps (lots of things expressing version opinions about the same diamond dependency)
        - number of dependency cycles via that dep
        - AI: Jordan to write up description of consuming graph output, computing each metric, rationale for why that metric is impactful for k/k maintainers
    - two tools available
        - go mod graph
            - very fast
            - only outputs module-level dependencies
            - only works for things that declare go.mod
        - go list -deps
            - works at the package level
            - slightly slower
            - allows understanding dependencies via components that don't have go.mod
    - Use "vndr" / vendor.conf ?

Jan 23rd 2020, 11 am PT / 2 pm ET

Host: Andrew Sy Kim
Note Taker:
Attendees:

- <add your name here>
- Dims
- Lubomir I. Ivanov
- Jordan Liggitt
- Andrew Sy Kim
- David Ashpole (dashpole)
- Elijah Oyekunle (eloyekunle)

Agenda:

- [dims] CAdvisor dependency challenges
    - https://github.com/kubernetes/kubernetes/pull/86975
    - https://github.com/google/cadvisor/pull/2386
    - https://docs.google.com/document/d/1NTEIdBln3U9xKaFUarEnfAxT8LUF2Y-ATuWqA1npovw/edit
- CRI-API use in hcsshim
    - https://github.com/kubernetes/kubernetes/issues/87420
    - AI (dims): follow up with SIG node/windows
- Import-boss - reverse rules [elijah,sttts]
    - https://github.com/kubernetes/kubernetes/pull/83526
    - tl/dr: pkg/foo/.import_restrictions: pkg/foo may only be imported by these prefixes (enforced only within k/k obviously)
    - Example: https://github.com/kubernetes/kubernetes/pull/83526/commits/07fb6b4727f7b85215fd220a7839743f00ecf5b2
    - Example verification failure: https://prow.k8s.io/view/gcs/kubernetes-jenkins/pr-logs/pull/83526/pull-kubernetes-verify/1219307820348870657
    - gengo PR: https://github.com/kubernetes/gengo/pull/116
    - AI (elijah): Update devel documentation with the new feature
- [lubomir] cloning external repositories (kubeadm) as part of the k/k release tooling.
    - defer to SIG release since they will run into similar issues with kubectl

~~Jan 9th 2020, 11 am PT / 2 pm ET~~

~~Host: Dims~~

~~Note Taker:~~
~~**Attendees:**~~
- ~~<add your name here>~~
- ~~—~~

~~**Agenda:**~~
- ~~—~~

Nov 14th 2019, 11 am PT / 2 pm ET

Host: Dims
Note Taker: Andrew Sy Kim
Attendees:
- <add your name here>
-

Agenda:
- [dims] great progress on migrating mount package to k/utils, @codenrhoden doing most of the work
    - had to merge PRs in k/utils w/ CLA override
- [dims] removal openstack cloud provider is still pending CSI migration
    - remove duplicated mount package once above item is done
- [lubomir] kubeadm no longer depends on k/k
    - Jordan updated import boss to restrict new imports to k/k
    - pending discussions w/ SIG release
    - AI: open issue in k/release highlighting pending tasks
        - link?
        - release process needs to be able to pull in external repos given a SHA
        - [lubomir] was hoping kubectl was going to trailblaze the release process
        - [Jordan] kubectl is not as decoupled to kubernetes as kubeadm
    - two options for release
        - vendor it into k/k
            - [Jordan] can't do this, still depends on things in staging
        - have scripts point to a kubeadm SHA?
            - [Lubomir] how do we handle patch releases? should we bump this every release?
            - [dims] depends on release cadence
            - [Lubomir] ideally kubeadm shadows kubernetes release
        - separate release for kubeadm
            - release scripts in k/k would just reference a kubeadm release
            - [jordan] similar to addons that we bundle.
                - Example: cluster autoscaler will build/push a new release and then do a manifest bump before a release. Requires coordination but in practice has not been problematic.

                - [Lubomir] can we have a bot that automates coordination of tags between repos?
                - [Jordan] as this becomes more common, should we formalize this process?
        - [Lubomir] can we get rid of bazel in kubeadm repo if we're still releasing from kubernetes?
- Next steps for https://github.com/kubernetes/kubernetes/issues/85005 ?
    - AI: explore moving ipvs package to third_party/forked
- <add items here>

Oct 31st 2019, 11 am PT / 2 pm ET

Host: Dims
Note Taker: Andrew Sy Kim
Attendees:

- <add your name here>
- Stephen Augustus
- Andrew Sy Kim
- Tim Pepper
- Jordan Liggitt
- Yakov Zaytsev
- Dims
- Muhammad (novice)
- Quinton Hoole

Agenda:

- [augustus] hyperkube out-of-tree migration (Will they? Won't they?)
    - k/org: https://github.com/kubernetes/org/issues/1365
    - k/k: https://github.com/kubernetes/kubernetes/issues/81760
    - Some concerns after discussions on slack:
        - importing k8s.io/kubernetes - anti-pattern we are warning against
        - Jordan worked on a PoC which could be better
        - Jordan Q: What are people using hyperkube for?
            - Single image for packaging all binaries? This can be done with a docker file
            - Use a shim that routes to the correct binary?
        - Why are we dropping this at all?
            - because of dependency magnet issues
            - moving the dependency mess to another repo doesn't improve the situation
        - [Dims] Background

        - initial proposal: just get rid of it -- folks in community said they are still using it production, not sure who else is still using it.
        - external repo was a way to get folks in the community involved
- [Jordan] we need more clarity on what we are doing with hyperkube
    - if the sole purpose is to continue supporting specific use-cases while people transition off of it, we need to understand the key use-cases
        - single binary? single image?
        - some use-cases can be done without creating a separate repo
- [Stephen] put hyperkube in the release repo?
- [Dims] ideal end goal is we don't need it and we can get rid of hyperkube
- [Jordan] are usage metrics recorded anywhere?
    - [Stephen] check mailing list
        - Microsoft
            - AI (stephen): one more follow-up on hyperkube usage
        - Talos
        - ???
- [Jordan] no project should depend on k8s.io/kubernetes
    - not a supported use of the project
    - confusing/misleading if we create a new project that imports k8s.io/kubernetes
    - projects importing k8s.io/kubernetes will likely not be sustainable and not setup for success
- [Dims] let's say we don't setup the hyperkube repo, when will we get rid of it?
    - [Jordan] depends on what the people using it can tolerate
    - [Stephen] figure out usage metrics
        - sample size
        - need response from users using hyperkube
        - announcing a deprecation is likely to get more people yelling
    - [Jordan] update hyperkube to actually include each individual binary, this removes dependencies at the cost of a larger image.
        - doesn't break compatibility and gives incentives to users to stop using it

[Stephen] Ooof! We already removed it:

https://github.com/kubernetes/kubernetes/pull/83454

        - [Jordan] Can we get rid of the binary but keep the image?
            - Seems like most users just care about the all-in-one image
        - [Dims] some projects depend on the command line plumbing from hyperkube
            - [Jordan] not convincing argument since hyperkube is a single go file
        - [Jordan] recommendation is to build an image w/ same functionality by stitching together existing binaries instead of a single binary
        - [Dims] so can we deprecate hyperkube binary?
            - [Stephen] should be do-able, but can we do it this cycle?
        - AI: Dims to add hyperkube **image** back into release artifacts
        - AI: Dims to start deprecation of hyperkube binary
        - [Stephen] our deprecation policy doesn't cover changes in process/release, similar situation with CNI
            - AI: ????
- pain points in using feature branches - needs someone with experience using feature branches in k8s.io/kubernetes (server-side apply?)
    - A Slack discussion (w.r.t. Ingress v1: https://github.com/kubernetes/kubernetes/issues/84026)
        - https://kubernetes.slack.com/archives/C2C40FMNF/p1572477299499000
    - Stakeholders? : SIG Arch/Release/PM/Testing/GH Admin subproject
    - [Stephen] Folks working on Ingress V1 want to land it in one-piece.
        - Note to unblock work: https://kubernetes.slack.com/archives/C2C40FMNF/p1572630581112700?thread_ts=1572482976.040000&cid=C2C40FMNF
    - [Stephen] Issue is that feature branch management requires elevated permissions to k8s.io/kubernetes
        - not something that is given out willy nilly
    - [Stephen] Considering "feature forks"
        - initial response from the community has been good
        - same or different github org?
        - Using kubernetes-sigs
            - can give groups admin access to a fork
        - [Dims] what is the expected lifecycle of feature forks?
            - [Stephen] same as any other repo in kubernetes-sigs
        - [Dims] What can we learn from the security org?
            - used a private fork of Kubernetes to address security issues
            - [Jordan] private fork is always stale and CI is broken because people working on CI can't see the private fork
            - [Jordan] need to rebase often and force push
            - Can we use branch fast forwarding?
                - master branch of feature fork mirrors master of k/k, feature development then happens on a branch of the feature fork

        - [Jordan] why can't someone just create a throw away fork and add collaborators?
            - you get the benefits of CI as long as a PR is open against k/k
            - is it important to preserve comments/PRs/issues?
                - this would be lost with personal forks
            - Stephen to take point
-
- <add items here>

Oct 17th 2019, 11 am PT / 2 pm ET

Host: Andrew Sy Kim
Note Taker:
Attendees:

- <add your name here>
- Sean Sullivan
- Jorge Alarcon
- Andrew Sy Kim
- Lubomir I. Ivanov
- Wojtek Tyczynski

Agenda:

- <add items here>
- kubeadm staging progress update?
    - kubeadm won't use staging, hard transition to external repo
    - questions for sig release
        - still release kubeadm as part of tarballs or new process?
        - our tools at the moment for releases don't support choosing builds from external repos, needs to be addressed prior to migration.
            - kubectl won't have this problem because it's still being staged
            - kubectl will run into this issue if removed from staging in the future
    - kubeadm will follow Kubernetes core release exactly
    - v1.18 was proposed in last release
        - Can we push this date back due to discussions above with SIG release
        - AI: andrewsykim to follow-up with Dims, Tim St Clair and Jordan
            - Dims okay to postpone by 1 release if it means we're skipping staging
    - Kubeadm / sig-release - release process discussion:
        - https://github.com/kubernetes/kubernetes/issues/83127
        - help wanted for ^
- kubectl staging
    - work in progress to stage kubectl
    - requires follow-up with SIG release on changes to release process after kubectl is moved to staging

        - semver for kubectl will be very similar to Kubernetes core
            - major and minor will track Kubernetes core
            - patch releases may not match patch releases in Kubernetes core
            - need to identify which versions of kubectls work with which versions of apiserver, hence tracking major/minor versions
        - Tentative ETA: v1.19
        - blocking items
            - kubectl get
            - kubectl convert
            - kubectl auth reconcile
- Issue Triage
- PR Triage
-

Oct 3rd 2019, 11 am PT / 2 pm ET

Host: Andrew Sy Kim
Note Taker:
Attendees:

- Andrew Sy Kim
- Dims
- Clayton Coleman
- Quinton Hoole
- David Eads
- Sean Sullivan
- Jordan Liggitt

Agenda:

- Go through Laundry List prioritization / Owners again?
- Laundry List prioritization / owners:
    - in progress
        - cloud providers - remove in-tree cloud providers
            - AI (andrewsykim): ask openstack if in-tree provider can be removed in v1.17
            - AI (andrewsykim): remove duplicated mount package in in-tree open stack provider
        - Kubectl - move to staging/ (90-95% already done, PRs in flight), need repo for plugin, last bit is dependency on rbac
            - https://github.com/kubernetes/kubectl/issues/725

                - Should internal APIs be in staging?
                - Jordan: should there be a stand-alone command for converting APIs?
                - AI: Clayton and Jordan to follow-up
    - not in progress, requested
        - device plugin API - move to staging/ just like CRI API
            - We should do this ASAP!
            - AI for dims
                - https://groups.google.com/forum/#!topic/kubernetes-sig-node/uFq-yi5HFCY
                - https://github.com/kubernetes/kubernetes/issues/82437
        - credential providers - make them pluggable
            - AI: Andrew to add OWNERS from SIG Auth or Cloud Provider
    - Kubeadm - move to k/kubeadm repo
        - owner: @neolit123
    - Test framework - move test framework to staging/
        - owner: @andrewsykim
        - stage the actual e2e tests as well
    - hyperkube - move to another repository
        - owner: @dims
        - 1st step: stop releasing hyperkube
        - Currently gauging interest on this work from community
        - Some images still depend on hyperkube, e.g conformance test images
            - Seems do-able for v1.17
    - cloud-controller-manager - move to staging? Move controllers as well not just binary
        - move cloud controllers -> k8s.io/cloud-provider/controllers
        - move cmd/cloud-controller-manager -> k8s.io/cloud-provider/cmd/cloud-controller-manager
    - kube-apiserver - move to staging/ (config only? config + binary?)
        - @dead2sk
    - kube-proxy - move to staging/ (config only? config + binary?)
        -
    - Dockershim - move to separate repository
        - We need to stabilize CRI API
        - What about the streaming/ package?
    - Client-go
        - build alternative for generated clients that doesn't require k8s.io/{api,apimachinery}? more stable release-to-release
        - @deads2k proposed in sig-api-machinery
    - kube-scheduler - move Policy to staging/*/kube-scheduler/config/v1 (Wei Huang for 1.17)
        - in progress, waiting for Jordan's review.

- Issue Triage?
- PR Triage?
-

Sept 19th 2019, 11 am PT / 2 pm ET

Host: Dims
Note Taker:
Attendees:

- Sean Sullivan - Google
- Wojtek Tyczynski
- Jordan Liggitt

Agenda:

- Laundry List prioritization / owners:
    - in progress
        - cloud providers - remove in-tree cloud providers
        - Kubectl - move to staging/ (90-95% already done, PRs in flight), need repo for plugin, last bit is dependency on rbac
    - not in progress, requested
        - device plugin API - move to staging/ just like CRI API
            - We should do this ASAP!
        - credential providers - make them pluggable
    - Kubeadm - move to k/kubeadm repo
    - Test framework - move test framework to staging/
        - e2e as well
    - hyperkube - move to another repository
    - cloud-controller-manager - move to staging? Move controllers as well not just binary
    - kube-apiserver - move to staging/ (config only? config + binary?)
        - @dead2sk
    - kube-proxy - move to staging/ (config only? config + binary?)
    - Dockershim - move to separate repository
        - We need to stabilize CRI API
        - What about the streaming/ package?
    - Client-go
        - build alternative for generated clients that doesn't require k8s.io/{api,apimachinery}? more stable release-to-release
        - @deads2k proposed in sig-api-machinery
    - kube-scheduler - move Policy to staging/*/kube-scheduler/config/v1 (Wei Huang for 1.17)
- Do we need a deprecation policy for Go API?
    - Let's make verbiage we have more visible (cc @liggitt)
- Issue Triage

- PR Triage
- Update to Go 1.13 for k/k
    - Need to wait for perf improvement

August 22th 2019, 11 am PT / 2 pm ET

Host: Dims
Note Taker:
Attendees:

- Jordan Liggitt
- Jay Pipes
- Wojtek
- Brahmaroutu
- Sean Sullivan
- Lubomir I. Ivanov
- Jorge Alarcon

Agenda:

- Update on kubectl utils moving out
- Import-boss issues
    - https://github.com/kubernetes/kubernetes/pull/81752
    - Followup tracked here:
        - https://github.com/kubernetes/kubernetes/issues/81799
- 5k job for distroless PR
    - We should run dims's PR (copied from yuwen's) this weekend.
    - https://github.com/kubernetes/kubernetes/pull/81752

August 8th 2019, 11 am PT / 2 pm ET

Host: Dims
Note Taker: jaypipes
Attendees:

- Dims
- Jay Pipes
- Jorge Alarcon
- Jordan Liggitt
- Wojtek Tyczynski
- Arturo Tarin

Agenda:

- (jaypipes): e2e testing! let's separate this out of k/k! :) Patrick Ohly started the refactoring, but seems kind of stalled?

    - Jay to check in with Patrick and see how to help
    - Jay and others, continuing trying using e2e vendored in external projects and provide feedback on continued pain there
- Mount code too difficult to reuse from CSI packages
    - add mount package from k/k/pkg/util/mount
    - openstack cloud provider moving to staging
    - Dims to talk to Travis to see how to speed things up (keep in k/utils)
- kubeadm: kustomize core
    - See PR. Need feedback from some sig-cli folks
- import boss ignores some files
    - reproducible using bash script
    - Dims to look into it
- moving cloud controllers to k8s.io/cloud-provider/controllers
    - ML thread on this
    - just need a skeleton/cookiecutter thing in k/k to use for creating new external cloud controllers
    - Jordan expects this will look like the cloud provider extraction
- update to latest klog
    - revert of revert of reapply…
    - we want to get to distroless
        - which requires a single log file
    - but… our binaries by default use klog
        - which creates a dir for logs
    - Wojtek to run 5k CI job against klog and the distroless PRs over the next few days
        - verify duplicate log file issue is resolved
        - verify performance regression doesn't resurface

July 25th 2019, 2019 11 am PT / 2 pm ET

Meeting Zoom: https://zoom.us/j/845605479
Host:
Note Taker:
Attendees:

- Dims
- Quinton Hoole
- Lubomir Ivanov
- Arturo Tarin
- Jay Pipes
- Jorge Alarcon
- Wojtek Tyczynski

Future Topics:

- pain points in managing multiple repos in http://github.com/kubernetes-csi/ - needs a representative from SIG Storage
- pain points in using feature branches - needs someone with experience using feature branches in k8s.io/kubernetes (server-side apply?

Agenda:

- <add agenda items here>
- (jaypipes) CredentialsProvider stuff… I have some questions about it (low priority discussion/can be done at end)
- (jaypipes) Including (any) code in vendor/ directory. Is this a practice the project wishes to continue? Or is there a plan to move towards getting rid of storing the vendor/ code directly in the source repositories and using dependency tracking manifest to manage vendor deps?

July 11th 2019, 2019 11 am PT / 2 pm ET

Meeting Zoom: https://zoom.us/j/845605479
Host: Andrew Sy Kim
Note Taker:
Attendees:

-

Future Topics:

- pain points in managing multiple repos in http://github.com/kubernetes-csi/ - needs a representative from SIG Storage
- pain points in using feature branches - needs someone with experience using feature branches in k8s.io/kubernetes (server-side apply?

Agenda:

- v1.16 planning
    - https://github.com/kubernetes/kubernetes/issues?q=is%3Aopen+is%3Aissue+label%3Aarea%2Fcode-organization

June 27th 2019, 2019 11 am PT / 2 pm ET - **CANCELLED**

Meeting Zoom: https://zoom.us/j/845605479
Host: Andrew Sy Kim
Note Taker:
Attendees:

-

Future Topics:

- pain points in managing multiple repos in http://github.com/kubernetes-csi/ - needs a representative from SIG Storage
- pain points in using feature branches - needs someone with experience using feature branches in k8s.io/kubernetes (server-side apply?

Agenda:

- <add agenda items here>

June 13th 2019, 2019 11 am PT / 2 pm ET (<add recording>)

Meeting Zoom: https://zoom.us/j/845605479
Host: Andrew Sy Kim
Note Taker: spiffxp

Attendees:

- Lubomir I. Ivanov - VMware
- Andrew Sy Kim - VMware
- Aaron Crickenberger - Google (@spiffxp)
- Jorge Alarcon
- Yassine Tijani - VMware
- Adel Zaalouk - SAP
- Matt Farina - Samsung SDS

Future Topics:

- pain points in managing multiple repos in http://github.com/kubernetes-csi/ - needs a representative from SIG Storage
- pain points in using feature branches - needs someone with experience using feature branches in k8s.io/kubernetes (server-side apply?)

Agenda:

- Staging kubeadm. Priority? What's blocking? Timelines?
    - timothysc - What are the goals?
        - Liggitt:
            - Goals of moving kubeadm out of tree:
                - Enable different release cadence of kubeadm (would have helped get critical kubeadm-only setup fixes out in previous releases)
                - Reduce dependency gridlock in k/k (e.g. kubeadm > docker)

        - Enable reuse of pieces of kubeadm? (sounds like only the kubeadm config types are desired)
        - Make it easier to prevent import of things that aren't intended to be imported outside of k/k (can be partially replicated with import-boss)
    - Benefits of intermediate move to staging
        - Allows work to start on CI jobs using an external kubeadm
        - Allows people to import kubeadm types outside k/k immediately
        - Prevents all new dependencies on k/k from kubeadm
    - Costs of intermediate move to staging
        - ?
- Timothysc - Staging as representation of what we intend to be publicly exported
- Only API we care about though is the command line, and serialized config types
- Dims - if kubeadm has its own release cadence, that would be valuable (but staging doesn't enable that (liggitt))... put in staging? Take it out completely?
- Timothysc - Reason we are so tightly coupled is because there is no contract or guarantees that we uphold…(think these are guarantees of k/k)
- We tend to change behavior in some subtle way and then eat it on the backside. Thus usually we're usually the ones to catch things first
- My hope is if we create APIs around this thing, there would be some level of rigor around the behavior of these things
- Liggitt - The API objects hit the same things flags hit, nothing is stopping us from writing better tests against flag based changes
- How are people configuring all of these things today? With flags, so it seems like gating on config files for stability is beside the point
- Coupling kubeadm to the kubernetes release lifecycle seems problematic, thus a desire to move out of staging, since staging still couples us
- Timothysc - from a development perspective it's not a high priority for us, because we get so many benefits from living in k/k
- Liggitt - something got lost along the way, kubeadm was never meant to be in tree, it was just put there for expedience… as a setup tool it is distinctly different from all of the other components that are built and live there
- The people bearing the cost and impact of this are typically the ones trying to wrangle the kubernetes dependency tree
- Timothysc - there are parts of our community that want this now (eg: api types)
- Chuckha - as a dev importing kubeadm types, not burdensome
- Liggitt - not who I'm referring to, eg: would be cloud provider trying to get a bug fix and bumping into docker dep version conflict
- Responses in the past have been "wait on componentconfig" and that doesn't seem like it should be a blocker
- Timothysc - reasonable expectation to have guarantees

        - Liggitt - which doesn't block on API, those guarantees can exist today based on flags
        - Agree that what is impossible is to combinatorially test all config variations
        - Aaron - can kubeadm override flags today to allow us to test these things? (yes)
        - Liggitt - several things going on: what works comes first? If reusing go types for kubeadm not a goal, then moving to staging not a goal, and moving to staging doesn't really help the "dependency gridlock" problem.
        - Given that maybe moving to distinct repo should be prioritized
        - Isolating dependencies to make the hop possible, there are many deps in kubeadm that shouldn't be there
        - Second problem is testing and test apparatus
        - Timothysc - there are things we get bitten by, which we want the code org project to handle
        - What about forcing cluster provisioning tools to be out of tree
        - What about nuking cluster entirely
        - Aaron - feel like the path forward is deciding how to get increased test coverage to meet parity with cluster… kubeadm at present only really exercises the set of tests that kind supports
        - Liggitt - if getting its own repo is a long term goal, then getting to staging lets you start developing the "test this using that over there" sort of workflows
        - [lubomir] questions for staging and the publishing bot
            - can the target repository contain authored content and can the bot only push/maintain a single folder in the repo, for example, put the current kuberentes/cmd/kubeadm under a sub folder under the kubernetes/kubeadm repo without stomping the existing authored content.
        - [lubomir] kubeadm API to staging:
            - question about the cluster-bootstrap repo:
                - https://github.com/kubernetes/cluster-bootstrap
        - List of current deps for kubeadm:
            - https://docs.google.com/document/d/1oDp7fXhGHA66cpaKE7Sm4eF3w4G4uZrq8EhtSWBxfus/edit
        - Tracking issues:
            - Import-boss:
                - https://github.com/kubernetes/kubeadm/issues/1600
            - kubeadm's package layout:
                - https://github.com/kubernetes/kubeadm/issues/1205
        - AI: Lubomir to follow-up with meeting logistics for kubeadm planning for v1.16
- Import Aliasing in k8s.io/kubernetes
    - https://github.com/kubernetes/kubernetes/pull/76863
    - https://github.com/kubernetes/kubernetes/pull/78780
    - https://github.com/kubernetes/kubernetes/pull/78836
    - Conflicting import alias across the repo and PRs

        - Updating import aliases is acceptable but increases technical debt since there's a lack of consistency for import aliases.
        - Maybe using go-imports to enforce this is okay? How does this slow down development workflow?
        - Dims suggestion:
            - ./hack/import-aliases - holds the desired import aliases
            - ./hack/verify-import-aliases.sh - accepts a file (./hack/import-aliases) and an include path then uses AST to verify if import aliases matches. Optionally force rewrite to file.
    - AI: push forward with dims PR, base spiffxp's "fixup" PR on that, blacklist import alias verify from CI
    - AI: revisit results of "ease of conformance review" and how fiddly this is in a month
- Updating vendored deps into tagged releases https://github.com/kubernetes/kubernetes/issues/78434
    - List of current deps requiring updates: https://github.com/kubernetes/kubernetes/issues/78434
    - SIGs are not going through dependency updates to see if the version updates are valid.
    - Thought was once we get this in, what more could we do and how could we get others to help? To the point where these are smaller PRs, log these as "good first issue" things etc.
    - How can we split this work up?
    - Liggitt - when we're reviewing some of these things, some of them can be a little more impactful and require things like specific review from specific sigs… we're not blindly merging these
    - Andrew - would it be helpful to group these commits to specify which sig should look at which lib?
    - Liggitt - would be good to update the description to check these off as they get reviewed, and probably mention which sigs would be interested in them
    - Sigs might be intimately familiar with direct dependencies (or you could derive from imports and code paths - ed) but transitives are much harder
    - Should we have verify scripts that doesn't allow commit sha dependencies and force tagged releases?
    - Timothysc - there are very specific conditions where you drag in one dependency, but they pin their deps, and we use a different dep… this is a valiant effort and a good idea but the go community doesn't often adhere to standards
    - Dims - The other thing that would be helpful: once we get things to a sane level, eg: if there's a CVE we'll be able to have the kind of conversation about what's different between versions which is harder to have with random SHAs

Open Discussion [timebox to N min]:

- Add any discussion topic you want here

May 2nd, 2019 11 am PT / 2 pm ET (<add recording>)

Meeting Zoom: https://zoom.us/j/845605479
Host : Andrew Sy Kim
Note Taker: Jordan Liggitt
Attendees:

- Andrew Sy Kim
- dims
- Jordan Liggitt
- Jorge Alarcon
- Guinevere Saenger
- Lubomir I. Ivanov
- Alok Kumar Singh
- Quinton Hoole

Future Topics:

- pain points in managing multiple repos in http://github.com/kubernetes-csi/
- pain points in using feature branches

Recurring Topics [timebox to N min]:

- Welcome new members/attendees!
- AI review:
  - @liggitt, @alok to put together a script to gather specific metrics for dependency graph, update doc for reviewing vendor updates to evaluate impact before/after a proposed update to see if it is moving in the right direction
    - in progress at https://github.com/alok87/kubernetes/commit/79d238d50751ce3ff33961a35c0b8caf838ae26f
    - Gathering package-level, module-level, incoming/outgoing
    - Plan to check in counts in a text file along with vendor which we can review (if we can do so in a way that doesn't become a conflict/rebase issue)
    - Automate some statistics that can be in one of the CI jobs to spit out some dependency data for reviewers

Open Discussion [timebox to N min]:

- Add your suggested topic here [firstname lastname, email or @slackname]
-
- [andrewsykim] What are the current priorities / areas of focus? Should we pick 1-2 top priorities and focus our efforts there?
  - external dependency management
  - sweep dependency staleness

        - how far from master / latest tagged release are we?
            - script to gather info about stale versions (`go list -u` shows available updates, etc)
        - bugs? security updates?
        - should SIGs explicitly owned dependencies?
            - custom tooling to link a dep to a SIG?
            - who pays attention to bugfixes/security updates for a given dep?
            - transitive dependencies are very hard
            - AI: Alok to investigate tooling for discovering owning SIG
    - reducing binary sizes
        - good tool for identifying big packages we barely depend on (e.g. ugorji)
        - several providers we're ready to drop in 1.16 (photon, ...)
    - general code cleanliness, package structure, naming, etc
        - rename packages/methds before staging as consumability increases
    - consumability of Kubernetes code - what needs to be staged?
        - subtopic/prereq - go API compatibility goals/plans?
        - most easily separable commands:
            - kubectl (KEP)
            - kubeadm
                - blocker: config API still in v1beta1 - is a v1 API necessary for staging?
                - AI: gather more folks in SIG Cluster Lifecycle to discuss this further
            - cloud-controller-manager
                - not a leaf node, should we prioritize?
        - component config direction/endgame?
            - config types (e.g. kubelet config)
            - application libraries (e.g. pkg/kubelet)
            - application commands (e.g. cmd/kubelet)

April 18th, 2019 (recording)

Meeting Zoom: https://zoom.us/j/845605479
Host : Dims
Note Taker: TBD
Attendees:

- *Jaice*
- *Dims*
- *Andrew Sy Kim*
- *Maciej Szulik*
- *Alok Kumar Singh*

- *Tim Hockin*
- *Matt Farina*
- *Ryan Bezdicek*
- *Jordan Liggitt*
- *Quinton Hoole*
- *Damini*
- *Brendan Burns*
- *David Eads*
- *Justin Santa Barbara*
- *Vallery Lancey*
- *Clayton Colman*
- *Tim Allclair*
- *Mayank Kumar*

Recurring Topics [timebox to N min]:

- Welcome any new members or attendees
- AI review

Open Discussion [timebox to N min]:

- Add your suggested topic here [firstname lastname, email or @slackname]
- [andrewsykim] Updates to Cloud Provider Efforts
    - snipping cloudprovider → k/k dependencies
        - and also storage plugins
        - and credential providers (sig-node? sig-auth?)
            - some
    - relocating cloud providers to staging repo
        - PR to migrate first provider needs review https://github.com/kubernetes/kubernetes/pull/75910
- Recent successful efforts:
    - made cadvisor container integrations opt-in, let us trim mesos and rkt dependencies from kube (cadvisor #2209, #76291)
    - tooling-only vendoring (#76466)
    - follow-up dropping things outside their deprecation period (e.g. long-deprecated kube-up providers #76711)
- Recent problematic issues:
    - proto/grpc library versions are very impactful and our dependency graph has many consumers at wildly different expected levels (#76656)
    - [clayton] Inability to extract e2e tests from the core Kube codebase
        - Anyone have an update on “testing-commons” effort?
            - https://github.com/kubernetes/kubernetes/issues/75601
            - https://github.com/kubernetes/kubernetes/issues/75604
        - ACTION: Document what staging/ rules apply
    - A KEP was opened to discuss the movement of streaming library out of k/k. Tim has a suggestion around it (#937)
- Approach to prioritizing work:

- metrics:
    - outgoing transitive dependency links (either direct or full downstream tree)
        - Age / out-of-dateness?
    - compatibility (hard to quantify, but very impactful when trying to select an appropriate common version, and some often used libraries are far better than others)
    - code size
- Tools
-
    - go mod graph (module-level) (top-level is not workable)
    - @tallclair has a package-level tool - https://github.com/google/godepq
    - stale dependencies script (needs go-module update)
        - 2018 results: https://docs.google.com/presentation/d/1i3Qnt2yQOfqVLXXkyiaQ_6HOj5ZqkeWQ2wRmT1U2aBs/edit#slide=id.g499934ca8f_0_72
    - Red hat has an effort to do open source cve/security/dependency analysis of golang source trees, Clayton will have them expect to assist where possible here
    - Snky? https://snyk.io/
    - Does The Linux Foundation have any tools/efforts to offer? [quinton] A quick search suggests “no”.
    - Investigate use of static analysis
        - https://github.com/avelino/awesome-go#code-analysis
        - https://github.com/mre/awesome-static-analysis#go
    - AI: @liggitt, @alok to put together a script to gather specific metrics listed above, update doc for reviewing vendor updates to evaluate impact before/after a proposed update to see if it is moving in the right direction
- [clayton] Questions about scope
    - Do we address binary size issues related to code organization?
        - SUGGESTIONS: Delegate to other sigs and efforts rather than being too focused
        -
    - Do we address process complications caused by changes we suggest (moving things out, splitting things out)? What philosophies guide our decisions?
        - SUGGESTIONS: Look for things that make maintainability easier (or even possible)
        - Have clear reasons/benefits for each thing being separated/split, along with enumeration of costs/risks (for developers, deployers, users)
        - Focus on catching them on the way in (good metrics for vendor reviewers)
        - Look for reducing fragility of high impact security / maintenance issues (we can’t bump because of X)
        - Create explicit priorities around what cleanup is most valuable

Current Status

Efforts are around - “How do we avoid folks having to vendor k/k”. Examples are:

- SIG cloud-provider driving changes, so external cloud providers do not need to vendor k/k
- ~~CRI can be split out into staging/~~
- SIG cluster-lifecycle doing work for better configuration componentization
- Running E2E tests externally

Mission statement

Kubernetes prefers a distributed, mostly decentralized code model, and relies on automation over process to help support developers. By better organizing, subdividing, and maintaining our code we can improve outcomes for our developers, our ecosystem, and our users.

End Goals

- Better developer experience for folks contributing to kubernetes
    - Better subdivision of complexity (don’t need to know about kubelet to do X)
- Better user experience for folks using our code in their own projects*ƒ*
    - Vendoring an important Kube component should be easy
    - SIGs should have a good balance between delivering new code changes and preserving API compatibility for consumers
    - Not all release cycles of components exactly match Kube releases
- Better control of dependencies between subprojects
    - E.g. e2e tests depends on all code in the kube project just to get 5-10 utility classes and packages
    - Internal-like packages (though Golang’s internal packages do not work for us)
- More clearly define ownership of code within k/k for approvers / reviewers / sigs
    - E.g. staging/src/k8s.io/kubelet/* has a clear ownership
- Encourage mental models that lead to ecosystem health
    - Plugin models and making Kube code “first of many” encourages community flexibility - cloud providers, virtual kubelet, CRI, CSI
    - Clear API boundaries give us testable rules for conformance
- Smaller kubernetes/kubernetes repository once the staging/ repos are moved out
    - How do we better coordinate work across repos when we do that?
    - How do we reduce drift between levels of kube repos? Vendoring is difficult and drift between disparate components makes it even harder.
- Reduce the number of external dependencies we use
    - Survey minimally used libs and try to eliminate them?

    - Reducing external dependencies doc
- Ensure we pick up Security fixes in dependencies?

How do we do it?

- Continue to refine existing tools and patterns
    - staging/src as a way of creating artificial walls
    - Bots as a way of controlling access
    - Import protection as a way of hiding dependencies
- ~~Better support for Go modules?~~
    - ~~There's a KEP for that…~~
    - ~~https://github.com/kubernetes/enhancements/blob/master/keps/sig-architecture/2019-03-19-go-modules.md~~
    - DONE!
- Better support for folks to use Feature branches?
- More feature rich publishing-bot?
    - What's missing here?

Existing issues

Keywords: "staging", "godep", "vendoring", "go modules", "multiple repos", "feature branches"

- https://github.com/kubernetes/kubernetes/issues/24343
    - https://github.com/kubernetes/kubernetes/issues/41629
- https://github.com/kubernetes/kubernetes/issues/44873
- https://github.com/kubernetes/kubernetes/issues/48209
- https://github.com/kubernetes/kubernetes/issues/63607
- https://github.com/kubernetes/kubernetes/issues/68201
- https://github.com/kubernetes/kubernetes/issues/68577
- https://github.com/kubernetes/kubernetes/issues/69585
- https://github.com/kubernetes/kubernetes/pull/70081
- https://github.com/kubernetes/kubernetes/pull/70292
- https://github.com/kubernetes/kubernetes/issues/72638
- https://github.com/kubernetes/kubernetes/issues/73504
- https://github.com/kubernetes/kubernetes/pull/74021
- https://github.com/kubernetes/kubernetes/pull/74087
- https://github.com/kubernetes/kubernetes/issues/74352
- https://github.com/kubernetes/community/issues/566
- https://github.com/kubernetes/kubernetes/issues/75526
- https://github.com/kubernetes/kubernetes/pull/74691

Problems faced by folks

- https://github.com/kubernetes/kubernetes/issues/75338
- e2e-tests require importing all of k/k, they should be a client and only depend on public apis and some reused utility code as a discipline / protection against regression / cleanliness thing
- Node team has discussed wanting to create better separation of kubelet from core code base (to be client-only) and moving to staging/src was desirable
- Kube-proxy has been discussed moving to staging/src so that network vendors can more easily use / reuse it since they often need to tweak its implementation
- SIGs Cloud provider, API machinery, CLI, and component-base are moving significant amounts of code

TODO

- ~~We need a label for issues~~
    - https://github.com/kubernetes/test-infra/pull/12170
- ~~We need a slack channel~~
    - Please join #k8s-code-organization
- We need a project board to coordinate work
- Find people interested and setup a regular meeting
- Create roles and post to role board

Who is interested?

- Clayton Coleman
- Andrew Sy Kim (to help SIG Cloud Provider efforts move along)
- Vallery Lancey (@vllry)
- David Eads (@deads2k)
- Alok Kumar Singh (@alok87)
- Hippie Hacker (@hh)
- Jordan Liggitt (@liggitt)
- Stefan Schimanski (@sttts)
- Rohit Sardesai (@rohitsardesai83)
- Harsh Vardhan (@vharsh)
- Damini Satya (@daminisatya)
- Rahulkrishnan R A (@rahulkrishnanra)
- Tim Allclair (@tallclair)
- Lalatendu Mohanty(@LalatenduMohanty)
- Quinton Hoole
- Wojtek Tyczynski (@wojtek-t)

- Jorge Alarcon (@alejandrox1)

Maybe relevant:

- Recent issue / question raised by Aaron: https://github.com/kubernetes/kubernetes/pull/75339
- Contrib summit session notes: https://docs.google.com/document/d/163JvzDIuBa4CzttxxG8tjYYPACQnD1DboW3BDG8VCUA/edit#
- 2017 notes: https://docs.google.com/document/d/1_3hyDE4n-HLz5Z3YfGFVW1X1rkx0g5wF_JgtzjMeWpQ/edit?ts=592e0b4c#
- Prototype of an inverse import boss tool: “declare who (in k/k) is able to import this package” – flexible variant which can enforce internal-like package behaviour https://github.com/kubernetes/gengo/pull/116