# DevOps WG Meeting 08/13/18

**08 August 2018 / 9:00 AM CT**

## ATTENDEES

Jeremy Phelps, Steve Osselton, Trevor Conn, Andrew Foster, Tony Espy, Jim White, Ian Johnson, Michael Hall, Brandon Forster

## AGENDA

**Last Meeting Follow-up (Last week was cancelled.)**

1. California dot release still in the works.

2. Branch Cut Delhi on Oct 8th, formal release pushed out to November. This gives us some cushion for issues and travel to F2F. (TSC)

3. Manage go deps across branches. (Trevor) This is done?  Did we pin versions?  I also noticed Drasko has a PR for migrating to dep.

    a. Was discussed that we were not migrating to dep.  Jim closing PR.

    b. We will migrate dep tools once the official becomes ready (Vigo).

    c. Pin deps to specific Version/git tag in glide.yaml.  Issue for Delhi (JIM for now).

4. Jeremy to port pom.xml enhancements for java services to master (In the works)

5. Separation of deployment and code build.  This is already done in separate jobs for the java services.  Work is being done on core-config-seed-go, egdex-go, and security-api-gateway to do this same (On Hold).  The way we have it in go services work currently, and this is an optimization we can make.  Need to get other things sorted first.

6. LF infra ARM hardware still in the works.  I am able to launch arm servers on our provider now.  I'm refactoring our ansible bits to account for multi-arch.

      a. 16.04 vs 18.04 affects snap builds.

      b. Will need to use containers to build snap on 18.04 (LXD or Docker). (JEREMY)

      <span style="color:red">c. Ansible templates help</span>

7. Repos and nexus bits created for device-sdk-(go|c).  Still getting jenkins jobs outlined. (Looking for volunteers).  This is still outstanding.

      <span style="color:red">a. Volunteer needed.</span>

8. Refactor of some java staging builds.  For dependencies during the staging build we will only pull official releases. (device-(virtual|modbus)) completed.  Others in progress...volunteers?

9.

1. Tony Espy and team had successful snap build on cali.  Currently these build java service snapshots, which is not ideal;  Have decided to get it working and then refactor the jars that are used.
      a. Note here that SNAPS are building latest on cali.
      <span style="color:red">b. Need integration testing before publishing stable.</span>

2.  Core-domain brought into california as it's a dependency that is still needed and Steve Osselton found a bug to be fixed.

3.  Finish out jenkins jobs for device-sdk-(go|c)

4. Edgex-UI ...create repo (TODO) (Delhi release item; flesh out as we go)
      a. No updates yet.
      b. Bring it over soon (JIM will contact)
      <span style="color:red">c. Will need Jenkins jobs (JEREMY?)</span>
      d. Create nexus bits (JEREMY)

5.  Port java staging Docker file refactors to master branch (only makes sense to do if we are going to release java services in Delhi).

6.  Deprecation and archiving of java services.  POST DELHI
      a. device-sdk-(go|c) will be key here.
      b. Jim follow up about Samsung.
      c. Get a general idea about where we are at here.
      d. I am moving forward with Java services CI refactors.
      <span style="color:red">e. Jim and Keith actively discussing with Samsung</span>

7.  Make sure all generated containers have the git-sha and VERSION interpolated into the build as a LABEL.  This is currently done on edgex-go and core-config-seed-go containers.  Need to do in java service repos and security* repos. (JEREMY)  Done, but there is one issue still work work out for Java services.

8. Publish daily docker images to Docker Hub. (IN PROGRESS) (Will follow up with Jim) (Look at how it affects black box tests).  Not done yet.

9. **Discuss regularly releasing artifacts to release repos.  This is done very regularly across projects without fanfare.  A named release is then largely administrative in nature and points to a set of versions.**  We do in fact need to do this; I received some guidance from Thanh Ha that our staging job artifacts should only have deps on RELEASED artifacts.  I'm talking with him today in more detail about this.
    a. Mono-repo presents a challenge here.
    b. Vgo may help out.

10. Regarding point 9.  This will mean that our individual service versions will drift from each other… this is totally ok.

11. Michael, There are some docker-compose files don't work (references to latest tag).  Jim is checking on his PR's and also checking the compose file in edgex-go. Jeremy still has cleanup to do, delete all latest tags from registries.

12. TREVOR.  Issue identified with docker-compose file versions.  We have v2 and v3 which are drastically different.  ANDY F to check on this.

13. TONY and ANDY to talk about snap testing.  IOTech has interest.

## Documentation

1. Sphinx docs published to nexus atm.
    a. Want a docs.edgexfoundry.org
    b. Michael suggests pointing at california docs for now.
    c. Jeremy to ping INFRA team to set up redirect.
    d. Readthedocs tabled until future.

2. Go Docs
    a. We have this now but not being built/published.
    b. Use device-sdk-go docs as reference.
    c. Discuss at next meeting.

3. CI docs.
    a. Some extra help needed in this area.

# CI-WOES

  There seems to be a general misunderstanding about CI.  I hope to clarify
some points and in the process get consensus.  CI, like any of the projects,
is essential in EdgeX longevity and success.  EdgeX, being an open source
project, will succeed on the efforts of the community and not a single person.

We have verify jobs that run on any PR action (opening, update, comment of
"recheck").  The verify jobs build code and run unit tests.  This is the first
gate to pass in order to merge into a branch.  A merge event triggers a
"merge" job that builds code, runs unit tests, and produces a SNAPSHOT
artifact.  The next "level" up would be the daily jobs, which do similar
things but in turn produce a staging artifact (A Release Candidate).  This
artifact runs through our black box testing (having been through unit testing
several times already).  At any time a project can opt to do a formal release
of their service from a staging environment.  IMO this should be early and
often.  Other projects that depend on this service are responsible for knowing
and coordinating when to fetch the new release.  We have had at least 2
instances I know of now because we were not pinned to released dependencies.

Further, dealing with dependencies is generally difficult not matter what
language you work with.  We will have the same difficulty with golang that we
had with java.

Here are some points that I'd like to discuss in detail:

  1. Projects are responsible for managing their own dependencies properly.
  2. CI should be "brittle".  That is, we do want it to fail at the first
     hint of an issue.  In fact, this is especially beneficial in a
     micro-service environment.
  3. Projects should do official releases early and often.
  4. A projects staging artifact (release candidate), should only rely on
     dependencies that have been officially released.
  5. Should SNAPSHOT artifacts only depend on officially released? (many
     projects are at this level).
  6. We should separate packaging from code build/test as much as possible.
  7. Projects should do more and more CI implementation details.

8. Having a service implemented in more than one language makes integration testing especially difficult.  (Think a matrix of black box testing scenarios).  We need a plan for this or we will get swamped.)
9. Can we remove all of the barcelona Jenkins jobs and lock that branch?
10. Can we remove all jobs for services that have been migrated to go? (core-command et. al.)

## NOTES

## ACTION ITEMS
1. ...

## NEXT WEEK'S AGENDA

...