DevOps Working Group

Thursday July 11, 2019
# Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Topic</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 min</td>
<td>Edinburgh: Dot Release Discussion / Planning</td>
<td>James Gregg / Eric Ball / Jim White</td>
</tr>
<tr>
<td>10 min</td>
<td>Performance Testing – LF Configuration EB IAC</td>
<td>Robin</td>
</tr>
<tr>
<td>10 Min</td>
<td>Fuji Update</td>
<td>James Gregg</td>
</tr>
<tr>
<td>10 Min</td>
<td>GitHub Projects for Issue Tracking</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td>Opens</td>
<td>All</td>
</tr>
</tbody>
</table>
Attendees
EdgeX DevOps WG Update (Edinburgh Dot Release)

- Edinburgh Dot Release (1.0.1)
- An race condition issue was discovered in go-mod-messaging
  - It occurred when core-data sent data to export services (or appl services)
  - It was intermittent, but was verified as an issue by several people
  - It appears most often on Mac OS machines
  - If this error occurs, nothing upstream will get data from EdgeX
- There may be another issue that needs addressed
  - Issue on low power devices??

- List of Artifacts impacted
  - go-mod-messaging (this is the module that doesn’t get released).
    All other services need to be released (1.0.1)
  - 1.) We need to merge a fix into the master and Edinburgh branches of go-mod-messaging
  - 2.) The Edinburgh branch of both edgex-go and app-functions-sdk will need to consume the Edinburgh branch of go-mod-messaging
  - 3.) We will need to republish the artifacts for all services in edgex-go for the Edinburgh release as well as app-functions-sdk
  - 4.) We will need to adjust the Edinburgh docker-compose files to utilize the version numbers applied to the new images by the above actions.

Decision: Not ready to set a date yet... May be other issues to pull into the dot release as well. Jim White will send email out next Tues with discussion and plan in TSC call on Wed.
DevOps WG Update (recap from TSC call)

- Container Scanning (Clair Server landing request)
  - Risk for Fuji if the landing doesn’t happen
    - Mitigations in place as a interim solution
- Codecov.io integration completed on all repos with tests
- Static Code Analysis Tools - Enabled SonarCloud for EdgeX Foundry - will need to update all JJB templates to enable use of the tool if this is the direction we plan to go.
  - Discussion continued in next slide
- Performance Testing – Robin / Cloud
  - Received configuration of what LF implemented on AWS for the TIG stack on EB

- Edinburgh Docs
  - edgex-go docs – Documentation migrated out of edgex-go
  - Major refactor by Ernesto completed – THANK YOU!!!
  - Status Check - Michael Hall
- Docker Hub Nightly Push (dev)
  - Old ticket since 08/2018
  - Why not build the images if need latest and greatest
  - Not seeing the demand for this and we will defer this for now
  - **Decision**: Close the Ticket and Mark as “Will Not Work”
- GitHub Projects
  - Release Czar Proposal to use GitHub Projects for all Issues as an aide for the release and overview of what’s needed for Fuji release.
- Empty GitHub repos
  - Audit completed identified many empty repos in both main org and holding
  - Send audit results to tsc / core / device services mail lists
  - User account and team membership cleanup
<table>
<thead>
<tr>
<th>Helpdesk Ticket #</th>
<th>Description</th>
<th>Details</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>75648</td>
<td>Dedicated Clair server for EdgeX</td>
<td>Pending decision on strategy for K8s + cost / availability of resources with LF</td>
<td>WIP</td>
</tr>
</tbody>
</table>
SonarCloud Discussion (SAST Explore continued)

- **SonarCloud (Code Quality + SAST)**
  - Very Limited Security Checks for Go
  - Other Languages include (mostly web oriented)
    - OWASP Top 10
    - Sans Top 25
    - CWE

- **Rules**
  - [https://sonarcloud.io/organizations/edgexfoundry/rules](https://sonarcloud.io/organizations/edgexfoundry/rules)

- **Recommendation** is to bring this forward to the Security WG
  - Action Item: James will bring to Security WG for review and continuation of the explore
Meeting Minutes

• Jim will send out message next Tues - make the call on TSC call on timing for the dot release
  1.) We need to merge a fix into the master and Edinburgh branches of go-mod-messaging
  2.) The Edinburgh branch of both edgex-go and app-functions-sdk will need to consume the Edinburgh branch of go-mod-messaging
  3.) We will need to republish the artifacts for all services in edgex-go for the Edinburgh release as well as app-functions-sdk
  4.) We will need to adjust the Edinburgh docker-compose files to utilize the version numbers applied to the new images by the above actions.

• New Work Items added to Backlog
  • App functions functions - git semver not working
    • Need to create an Issue
  • Device-gps in holding – slated to move out after TSC approval / will need build automation
Edinburgh Retrospective

What went right?
• Communication of when the release was scheduled, was very clear.
• LF and DevOps team cooperating together seemed to work well but with pressure added on top of everyone
• Most artifacts were ready to release in the beginning. It didn’t seem like every repo was affected with issues (no extra work)
• Edinburgh Staging view was very helpful
• Great communication and collaboration between Intel DevOps team members and great prioritization
• Prioritization and Organization of the work (assignments and splitting up the work early on in the code release) was helpful

What went wrong?
• EdgeX-UI repos were late code drop
• Lack of a UI WG
• Communication around details was lacking from WG leads
• Need clear definition and understanding from WG leads that have different / independent release cycles
• JSD was introduced in the middle of the release and introduced issues that impacted communications with LF RE
• Availability and Competing priorities of Eric Ball / RE impacted release work
• Branches cut early caused extra work for both developers and DevOps
Edinburgh Retrospective (continued)

What Ideas would help next time?

• Jenkins Pipelines branch defined in Jenkinsfile
• Do Not cut the branch early
• Release from master
• Don’t allow PRs to master during code freeze (unless bug fix)
• Release Czar manages the release and acts as coordinator
• Need better visibility as to WIP during release. What does DONE look like?
• Set clear expectations for FUJI ahead of time
• Have a solid and well defined scope for executing the release.
• Shorten the release timeframe
• Actually Freeze Code
• Don’t pull in late code drops

What Actions will we take?

• Better coordination with RE to lay out the scope of work
• New UI WG - DONE
• Better tooling Pipelines (Geneva)
• Better clarity and organization / coordination of the work
• Release Czar coordinates and runs the release
• Shrink the release date where no development is going on during “code freeze”
• Create automation on the list of artifacts to release
Fuji Planning

Scope Discussions
Fuji – DevOps

**In**

- Static code analysis tool identified and integrated into the EdgeX Jenkins Pipeline for Docker image scanning (Clair Server)
- Explore SAST for true static code analysis to include additional tooling such as Fortify / Coverity
- Code and artifact signing with semantic versioning
- Fix Documentation – edgex-go
  - Create a new repo for edgex-docs
- Build Performance Optimizations
  - Pipelines for EdgeX Foundry base build images
  - Basebuild images managed locally within Nexus
  - Leverage PyPi Proxy for local pip dependencies
  - ARM builds – optimization leveraging different high CPU build nodes / OS (ARM Team)

**Out**

- Alternate deployment/orchestration
  - Beyond Docker/Snaps
  - Kubernetes
  - Kata Containers
  - …
- SonarQube – SonarCloud is already in play in the LF
  - Decision: wait to see what codecov.io offers
- Suggestion to rename all of the Jenkins “arm” jobs so as to differentiate 32bit / 64bit architectures
- Full Pipeline transformation for EdgeX services
## EdgeX DevOps Commitments (Fuji)

### Scope of Work

<table>
<thead>
<tr>
<th>Activity</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add static artifact analysis into the EdgeX Jenkins Pipeline (analysis of Docker /runtime artifacts, not the source code)</td>
<td>![Traffic Light]</td>
</tr>
<tr>
<td>Add code and artifact signing with semantic versioning</td>
<td>![Traffic Light]</td>
</tr>
<tr>
<td>Conduct build performance optimizations by:</td>
<td>![Traffic Light]</td>
</tr>
<tr>
<td>• Adding Pipelines for EdgeX Foundry base build images</td>
<td></td>
</tr>
<tr>
<td>• Allow base build images to be managed locally within Nexus</td>
<td></td>
</tr>
<tr>
<td>• Leverage PyPi Proxy for local pip dependencies</td>
<td></td>
</tr>
<tr>
<td>Explore static code analysis like Checkmarx, Coverity, GuardRails, Synk, SonarQube</td>
<td>![Traffic Light]</td>
</tr>
</tbody>
</table>

- **Clair Server landing at Risk** for Fuji
  - Work Around will be to use Intel hosted Clair server until decision is made by LF to support landing dedicated infrastructure
  - Work related to automating the scans as part of the build, will defer to Geneva scope

- `gitsemver` along with `lftools` used for artifact signing and semantic versioning

- Jenkins build performance optimizations for base build images completed

- All base build images will now be stored in Nexus (Snapshot):10003

- PyPi enabled as part of Edinburgh scope

- Initial review of GuardRails showed that the product was identifying issues which were not applicable for microservices architecture
Edinburgh Release

Release Planning
Edinburgh Dates

• Freeze Date – May 28
• Release Date – June 20
• Press Release – July 11
• Dot Release – July ?
## Past / Future Agenda Topics

| WW27 | No Meeting – US Holiday |
| WW28 |  |
| WW29 |  |
| WW30 |  |
| WW31 |  |
| WW32 |  |
| WW33 |  |
| WW34 |  |
| WW35 | Athens Project – proxy server for go package dependencies |
|      | Community Involvement |