DevOps Working Group

Thursday May 21, 2020
<table>
<thead>
<tr>
<th>Time</th>
<th>Topic</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 Min</td>
<td>Geneva Retrospective</td>
<td>All</td>
</tr>
<tr>
<td>25 Min</td>
<td>Hanoi / DevOps Updates</td>
<td>James</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All</td>
</tr>
<tr>
<td>5 Min</td>
<td>AOB / Opens</td>
<td>All</td>
</tr>
</tbody>
</table>
Attendees

- **JG** James Gregg (Intel) (Me)
- **TE** tony espy
- **BM** Bill Mahoney (Intel)
- **ER** Emilio Reyes (Intel)
- **EB** Eric Ball
- **EO** Ernesto Ojeda (Intel)
- **JW** Jim White
DevOps WG Recap (Geneva)

Geneva (May 2020):

- DevOps Jenkins Pipeline Transformation
  - Introduced new Jenkins Global Libraries for build automation
    - Includes test framework for Groovy code
    - Explore underway to look into code coverage of Groovy code using Codecov.io
  - Semantic Versioning using Intel contributed utility (git-semver) enhanced to include test framework
  - Continuous Delivery via "release-kraken"
  - Developer Enablement – GitHub Project Tracker, GitHub Issue label creation automated, gitcommit linter implemented *
  - New ci-build images and global libraries developed to support Jenkins Pipelines
  - New life cycle policies implemented on Linux Foundation Nexus repositories
  - Automation of the labels across the project
  - GitHub Tracker (Kanban board) – utilized weekly with built in workflow
  - Developer Documentation created for new Jenkins Pipelines
  - Improved performance of all builds to include collaboration with Linux Foundation to drive performance improvements for ARM builds (~15 mins build performance improvements using a new flavor of LF build nodes)
    - X86 build nodes (VM) uses 4cpu – 2gb
    - Arm64 build nodes (VM) now uses 4 cpu – 16gb

DevSecOps scope includes:

- Snyk Advanced Reporting via Community Bridge - $8K savings on licensing for developer licenses
- Snyk Docker Hub image scans with weekly reports of new vulnerabilities
- Snyk CLI of Go integrated into scan stage of Jenkins Pipelines
- Clair image scans within scan stage of Jenkins Pipelines
- DevOps contributed code fixes to address CVEs found in images based on Snyk reporting
- Lftools updated to use latest version – code signing, git tag signing, Docker image signing
Geneva Freeze and Release

TSC approved

- Freeze: 12pm GMT, April 22 (Wed, week before planning meeting)
- Release: 12pm GMT, May 13 (Wed two weeks after planning meeting)

See Geneva release notes for details (on Slack)

**REMINDER:**
We will NOT be branching off master for the Geneva release.

Includes EVERYTHING

Will not be versioning go modules

Do we need blackbox tests to be an “artifact” of a release?

- QA/ Test WG doesn’t require signed tags, but since release kraken can be used to automate the creation of the tag, it would be a signed tag
- If there’s a need to patch Geneva, the tagged blackbox tests would be used
- Since blackbox tests wasn’t previously considered a “release artifact” does it get tagged? – YES it does

Decision: We now need to consider blackbox tests as a formal artifact. Tag would be generated at the time of the formal release
Geneva Release Schedule

New scope – consider blackbox tests as artifact of the release
- should have been considered within review of ADR007

Green light decision to release
- TSC meeting late in the day
- Multiple issues worked throughout the day

support-rules-engine

Timeline to be reviewed for Geneva Retrospective

Snap label / promotion issue identified

Start Date: 4/22/20
Geneva Retrospective

What went right?

- Smoother release – no branching at code freeze equated to efficiency
- Whole DevOps team was responsive
- Developers embraced the opportunity to create the Jenkinsfiles themselves
- Great collaboration and cross pollination of the information
- Linux Foundation was very helpful and responsive in the release – easier and supported well
- Use of JIRA tickets helped with response times on support / help needed from LF release engineer
- Andrew Grimberg came into the DevOps wg for a roadmap discussion
- Automation of the release went well – good coordination
- Phased approach of the work helped align to sprint cadence
- ADR practice helped with communications across the project
- Ernesto recognized for work on the snaps
- Lisa recognized the good communication / teamwork with Emilio / Ernesto
- Tony / Ian helped with review of the snap automation code – THANK YOU!!
- Risk acceptance / Risk taken - It worked!!
- Dry Run on release automation functionality
- Tony / Ian were responsive wrt Snap store issues – THANK YOU!!

What could be improved?

- Communication gaps
  - support-rules-engine issue related to a change in plan
    - DEPRECATION (Define process needed ??)
- Snap release process could be better understood
  - Need full path to production for snap release process
  - release to beta candidate channel >> stable
  - Time crunch in the end could be root cause for the snap release issue
  - Might need TAF testing for snaps
  - No real hw testing (Akraino community lab – University of New Hampshire)
    - Canonical presentation on how they do hw testing with snaps
  - Need functional testing for snap automation
  - Need to figure out an example service (sample-service)
  - Inability to properly test in a sandbox, test environment
    - Help needed from LF to support ability
  - Release Kraken Improvements (re-lable / tagging)
    - Idempotency
    - Need to specify a commit (might be an edge use case but better)
    - Set up of the snap YAML
  - Manual release of documentation needs fixed
  - Snap store issues (503 error) – length of time to build snaps
DevOps WG Update

Hanoi

• **Performance Optimizations**
  - Build Optimizations for edgex-go
    - Explore completed and demo’d by Ernesto Ojeda with observable performance improvements in the build time
    - Internal team demo of Docker image promotion techniques
      - Need edgex-go to use git-semver
      - Additional unity / functional testing
      - Documentation
  - Other Sonatype tools could be complimentary if used within the developer workflow - OSS Index, DepShield, Nancy
    - Add a badge to the README to see issues with dependencies
  - Looked at GitHub Marketplace for a Dependency Management bot
    - Dependabot – adds a “dependency” label to the PR – good visualization option for the PR Reviewer
    - One option might be to monitor go.mod and IF CHANGED – add the label
    - Already using Snyk (without go.sum)

• **DevSecOps**
  - Continued explore of options for addressing Issue #1947 - vetting of 3rd party components (OSS dependencies)
    - Crawl – manual data collection
    - Walk - Automation of the Paper Study + Community Bridge + Clair + Snyk
      - Python script / Docker image “ghmetrics” - contribution from from Intel (Thank you Delay Bitner / Nick Haunschild / Ramu Bachala)
    - Jog – Enhanced Community Bridge – Advanced Snyk Reporting
    - Run - Nexus IQ offered by Linux Foundation
      - EdgeX Foundry would be the first to use it for Go but requires go.sum + go.mod
      - Previously ran into problems in the builds when were still on Jenkins Freestyle jobs with Verify stage
  - Other LFTools / Sigul latest version that supports Python 3.x
    - Update Eric Ball working on a sigul fork but running into some issues - IT-19186

• **Community Bridge Feature Requests**
  - Transitive dependencies for Go modules - findings don’t match other tools like Sonatype
    - SUPPORT-1311 - CommunityBridge findings for Go modules
  - Other LFTools / Sigul latest version that supports Python 3.x

• Community Bridge Feature Requests
Hanoi Planning

Scope Discussions
DevOps Scope of Work - Hanoi

- Performance Optimizations
  - Jenkins Pipeline optimizations for edgex-go
  - Explore options from LF for supporting Jenkins on K8s – completed roadmap review within Geneva
  - Explore alternatives to containerization within the builds
    - Explore use of BuildKit to simplify creation of x86/ARM build images so they share a single manifest when published to Docker Hub / Nexus
    - Explore use of Kaniko
      - Explore Complete – **Will not Work**
        - Requires use of K8s persistent volumes and dedicated build agents which are long lived
- Performance of the Build Environment
  - Monitoring / Alerting optimizations (Continuous Improvement Opportunity)
- Technical Debt
  - Caching Dependencies – speed it up (upstream dependencies)
    - Reference Linux Foundation roadmap
- Open Horizons Enablement
  - Shared Infra with Open Horizons
  - Build Automation for OH
- Stretch Goals
  - Code Coverage for Jenkins Global Libraries (codecov.io)
  - Snap improvements – build optimizations
  - Support for `–race` flag with goals to address with Go 1.15 …*but there are options*
Geneva Planning

Scope Discussions
### Fuji Release

- **Freeze:** Oct 23rd (Wednesday)
- **Release:** Nov 15th (Friday)

---

<table>
<thead>
<tr>
<th>Start Date: 10/23/19 (with extension)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th></th>
<th>week 1</th>
<th>week 2</th>
<th>week 3</th>
<th>week 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Assigned to:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Developers</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cut Fuji Branches</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update Documentation, Compose Files and Bug Fixes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>WG Chairs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GitHub Issues: Close / Mark for Geneva</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DevOps</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Fuji Jobs For Existing Repos</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DevOps</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clair Scan of EdgeX Images</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Release Tsar</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Release Tsar</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Code Freeze**
- **EdgeX F2F in Phoenix**
- **Open Tickets with LF for release on 11/15/19**
- **Finalize Release Notes**
Geneva – DevOps

In
• Full Pipeline transformation for EdgeX services
  • Convert Jenkins JJB Freestyle jobs to Jenkins Pipelines
• Introduce GitHub Org Plugin
• Simplified Jenkinsfile
• Global Libraries to support Jenkins Pipeline transformation
• Add Unit testing to global-libraries (uncommitted) **
• Snyk integration for edgex services
  • As part of Jenkins Pipeline conversion
• Slack integration with Jenkins pipelines
• Nexus Cleanup / Lifecycle Policy

Out
• Alternate deployment/orchestration
  • Beyond Docker/Snaps
  • Kubernetes
  • Kata Containers
  • …
• Integration Test Pipelines
• Code signing / Artifact signing **
Geneva Transformation: Architecture
How long does it take? Is this all Geneva scope?

Geneva Transformation

Phase 1
- Research Spikes
- Plugin Setup and Configuration
  - Jenkinsfile
  - Jenkinsfile.sandbox

Phase 2
- Jenkinsfile templates
- Implementation details get solidified
- Refactor existing pipelines to use new templates

Phase 3
- Existing Job Migration

Full Transformation by Geneva Release - April 2020
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)

<table>
<thead>
<tr>
<th>Scope of Work</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>✔️</td>
</tr>
<tr>
<td>Add code and artifact signing with semantic versioning</td>
<td>✔️</td>
</tr>
<tr>
<td>Conduct build performance optimizations by:</td>
<td>✔️</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>✔️</td>
</tr>
</tbody>
</table>

- Clair Server landing no longer at risk for Fuji
  - LF committed to implement on AWS and fund with expected completion next week
- 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
## Past / Future Agenda Topics

<table>
<thead>
<tr>
<th>Topic</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size change to use Ubuntu / Debian base build images to support –race flag for Go Lang</td>
<td></td>
</tr>
<tr>
<td>Clair scan findings – Discussion developer community if we want to break the build when there’s findings - Bring into Security WG for discussion</td>
<td></td>
</tr>
<tr>
<td>Open Horizons enablement</td>
<td></td>
</tr>
<tr>
<td>Alignment to new LF roadmap self-service offerings – EdgeX use case for handling holding repositories</td>
<td></td>
</tr>
<tr>
<td>Release automation - key learnings and sharing with LF</td>
<td></td>
</tr>
<tr>
<td>Explore use of Buildkit</td>
<td></td>
</tr>
<tr>
<td>Explore use of Kanico</td>
<td></td>
</tr>
<tr>
<td>Snyk Dashboard Review</td>
<td></td>
</tr>
</tbody>
</table>
Attendees & Community Participation – ww14

Community Participation

- Intel
- IoTech
- Dell
- VMWare
- ARM
- Canonical
- LF
- Kong
Attendees & Community Participation – ww15
Attendees & Community Participation – ww16

Attendees

Participants (5/6)
- James Gregg (Intel) (Me)
- Ernesto Ojeda (Intel)
- tony espy
- Emilio Reyes (Intel)
- Jim White
- Lisa Rashidi-Ranjar

Community Participation

- Intel
- IoTech
- Dell
- VMware
- ARM
- Canonical
- LF
- Kong
Attendees & Community Participation – ww17