DevOps Working Group

Thursday April 2, 2020
<table>
<thead>
<tr>
<th>Time</th>
<th>Topic</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Min</td>
<td>Geneva / DevOps Updates</td>
<td>James</td>
</tr>
<tr>
<td>20 Min</td>
<td>Open Horizons</td>
<td>Joe Pearson / David Booz / IBM</td>
</tr>
<tr>
<td>10 Min</td>
<td>Geneva Release Planning</td>
<td>Lisa</td>
</tr>
<tr>
<td>15 Min</td>
<td>Hanoi Release Pre-Planning</td>
<td>James / All</td>
</tr>
<tr>
<td>5 Min</td>
<td>AOB / Opens</td>
<td>All</td>
</tr>
</tbody>
</table>
Attendees
DevOps WG Update

• Jenkins Transformation to Pipelines
  • Work continues on the transformation to Jenkins Pipelines
    • Multiple stories related to release automation for Geneva - WIP
      • Create global function to handle docker images for the automated release
      • Add unit tests to global library – need to research
      • Add a snap stage job for edgex-go in cd management – WIP
      • New user story added to add edgex-global-pipeline version info to the pipeline log
      • Create global function for releasing git tags - WIP
      • Documentation
  • Optimizations of edgex-global-pipelines
    • Completed migration from Maven to Gradle unit test runner in order to easily leverage jacoco test reports for code coverage - DONE
    • go-mod-configuration Pipeline - Thank you Lenny !! - DONE
Open Horizon Project

• Started discussion with Open Horizon team
  • Shared documentation with David Booz/IBM
  • Meeting planned for 04/02/2020

• Linux Foundation Infrastructure
  • Access to the build infrastructure
    • Referred to Linux Foundation documentation
      https://docs.releng.linuxfoundation.org/en/latest/jenkins.html#

• Jenkins Pipelines
  • Current Build Process via Jenkins Pipelines documentation
    https://github.com/edgexfoundry/jenkins_pipeline_presentation
    https://docs.releng.linuxfoundation.org/en/latest/jenkins-sandbox.html

• OH source code
  • GitHub.com

• OH Build Infrastructure
  • Travis CI (Open Source)
  • Nexus / Docker Hub

• OH Artifacts
  • Docker images
  • Compiled binaries
  • Debian packages

• OH Resources
  • TBD: May – end of August

• Testing
  • Single VM

Timelines: 3-6 Months
Geneva Freeze and Release dates

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)
## Geneva release

<table>
<thead>
<tr>
<th></th>
<th>Completed</th>
<th>Work In Progress</th>
<th>Release Backlog</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>163</td>
<td>4</td>
<td>7</td>
</tr>
</tbody>
</table>

### Iteration Status

<table>
<thead>
<tr>
<th>Rank</th>
<th>ID</th>
<th>Name</th>
<th>Schedule State</th>
<th>Blocked</th>
<th>Points</th>
<th>Hours</th>
<th>Hours</th>
<th>Task Est</th>
<th>To Do</th>
<th>Owner</th>
<th>Defects</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>US6335</td>
<td>EdgeX DevOps: Integrate the edgeX/release Docker into the releaseKraken in cd-management</td>
<td>D</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>Lisa Rashid-Ranjbar</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>US6278</td>
<td>EdgeX DevOps: Need edgeX-global-pipeline version information in the pipeline log (implementation)</td>
<td>D</td>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>US6202</td>
<td>EdgeX DevOps: Documentation: edgeX-global-pipelines manual bump of stable and experimental tags</td>
<td>D</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>US6244</td>
<td>EdgeX DevOps: Integrate edgeXSnap function into edgeXBuildGoApp and edgeXBuildCApp</td>
<td>D</td>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>US6246</td>
<td>EdgeX DevOps: Integrate the snap release into the release kraken in cd-management</td>
<td>D</td>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>US5434</td>
<td>EdgeX DevOps: New build automation for device-opcua-c (BLOCKED)</td>
<td>D</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>US5925</td>
<td>EdgeX DevOps: Build automation for device-bluetooth-c (BLOCKED)</td>
<td>D</td>
<td></td>
<td>3</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>US5366</td>
<td>EdgeX DevOps: Create standard pipeline template for legacy Java repos - support:rules-engine (BLOCKED)</td>
<td>D</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Geneva Release Schedule
Geneva Dependency
Backlog Review
Meeting Minutes

No new update regarding open PR related to lftools / sigul – IT-19186

Officially closed the ticket related to Community Bridge – Advanced Snyk Reporting

Support for -race flag within CI - as discussed within the Architect’s Meeting this week (Hanoi pre-wire)
• It works for edgex-go because that Pipeline builds on the VM host, not within Docker like the other microservices
  •  edgex-go does not use the Alpine base-build image
• Action: Scope the work involved to change to Ubuntu / Debian based build images
  •  Note: There’s no commitment to include this scope for Hanoi – need to size the effort and report out at Hanoi planning meeting
  •  We build within Alpine base-build image because the microservices all use the same base for each EdgeX Foundry service
  •  Reason for using Alpine based image is not just because of performance, there’s also less work involved with patching / updating base build images when new CVEs are found

ADR to outline the release automation flow for EdgeX - https://github.com/edgexfoundry/edgex-docs/pull/110
• ADR0007 communicates the vision around continuous delivery and work involved to automate the release
• Action: Lisa will work to incorporate feedback from the reviewers
• The official approval process will require formal TSC approval
• Expect to close on this next week after additional review by the community

Removal of config-seed broke the Fuji jobs.
• Is there a plan for a Fuji dot release ? - NO
  •  DECISION: Jim White to cover with TSC next week – we will not fix the broken Fuji jobs
DevOps WG Update (Geneva)

Geneva (~Apr 2020) Focus:

• DevOps Jenkins Pipeline Transformation completed
  • 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
  • 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
Hanoi Planning

Scope Discussions
Hanoi - DevOps

• Performance Optimizations
  • Jenkins Pipeline optimizations for edgex-go
  • Explore options from LF for supporting Jenkins on K8s

• Performance of the Build Environment
  • Monitoring / Alerting optimizations (Continuous Improvement Opportunity)

• Technical Debt
  • Caching Dependencies – speed it up (upstream dependencies)

• Open Horizons Enablement
  • Shared Infra with Open Horizons
  • Build Automation for OH

• Stretch Goals
  • Code Coverage for Jenkins Global Libraries (codecov.io)
  • Snap improvements
  • Support for –race flag
Geneva Planning

Scope Discussions
Fuji Release

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

Phase 1
Work in Progress
Q3 2019
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</td>
<td>✔️</td>
</tr>
<tr>
<td>(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,</td>
<td>✔️</td>
</tr>
<tr>
<td>GuardRails, Synk, SonarQube</td>
<td></td>
</tr>
</tbody>
</table>

#### Accomplishments

- 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>
</tr>
</thead>
<tbody>
<tr>
<td>Size change to use Ubuntu / Debian base build images to support –race flag for Go Lang</td>
</tr>
<tr>
<td>Clair scan findings – Discussion developer community if we want to break the build when there’s findings</td>
</tr>
<tr>
<td>- Bring into Security WG for discussion</td>
</tr>
<tr>
<td>Open Horizons enablement</td>
</tr>
</tbody>
</table>
Attendees & Community Participation – ww14