## Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Topic</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Min</td>
<td>DevOps WG Update (Fuji)</td>
<td>James Gregg</td>
</tr>
<tr>
<td>10 Min</td>
<td>Snyk Demo Recap and Discussion</td>
<td>All</td>
</tr>
<tr>
<td>10 min</td>
<td>Other Business: Multiple Topics</td>
<td></td>
</tr>
<tr>
<td>10 Min</td>
<td>Opens</td>
<td>All</td>
</tr>
</tbody>
</table>
Attendees
DevOps WG Update

- **Container Scanning**
  - Clair server has now been landed by Linux Foundation on AWS.
  - Troubleshooting with Daniel Nunez to debug issues with the endpoint (503 Errors)

- **Static Code Analysis Tools**
  - Checkmarx, SonarCloud and Snyk output from egex-go scan shared with Security WG
  - Coverity scan and output still WIP

- **Audit of GitHub repos / teams** of main edgexfoundry org completed with handoff to Lisa
  - Review and identify repo owners
  - Discuss in next DevOps WG
    - Changes to add repo owners will require TSC approval

- **Build Performance Optimizations** (new scope)
  - Log archiving optimization made to the Jenkins freestyle jobs - Thank you Emilio

- **Proposal for Shared Infrastructure** under LF Edge umbrella introduced by Andrew Grimberg
  - Need clarifications for the list of questions that were not addressed in the DevOps WG

- **Jenkins Jobs**
  - Blackbox test ran for 5 days
Snyk Demo / Discussion

Feedback from Demo

An overview of your organization's vulnerability status

Security issues

<table>
<thead>
<tr>
<th>Issues</th>
<th>Projects</th>
<th>Severity</th>
<th>Issue type</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td></td>
<td>HIGH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td>MEDIUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>LOW</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

License issues

<table>
<thead>
<tr>
<th>Issues</th>
<th>Projects</th>
<th>Severity</th>
<th>Issue type</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>HIGH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>MEDIUM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>LOW</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Exposure window

Issues over time

- 11 HIGH
- 27 MEDIUM
- 6 LOW

To remove this message in the future, please run `snyk config set disabledSuggestions=true`
LFtools log publishing optimization

- Leverage existing docker image that includes LFtools to push logs to Nexus
  - Eliminates need of installing LFtools for every build
    - Saves time and cleans up logs by eliminating installation messages

- Build time reduction
  - ARM: ~4-5 minutes
  - X64: ~1-2 minutes

- Bug - SysInfo log missing SAR system information
API Endpoints (Lisa)
How do we want to report versioning long term based off a tag?

• Issue: We need a way for services to identify their version at runtime.
  • Right now we are injecting this at build time using the VERSION file.
  • We have seen that we often forget to update the VERSION file manually so this is problematic.

Options:
Previously discussed “git describe”
  • The downside to “git describe” is that without pushing a tag it will return the commit SHA.

• Recommendation
  • Use “git-semver” utility
How do we want to report versioning long term based off a tag? (continued)

In Edinburgh we implemented git-semver a tool to manage the version in github repos to address these small gaps. Git itself does not infer what the next version should be based off semantic versioning but git-semver does.

We can use git semver utility to inject the version into the build in the Makefile.

Right now app-service-configurable has a Makefile set up where we can pop in the git semver command.

https://github.com/edgexfoundry-holding/app-service-configurable/blob/master/makefile

Simply change the "cat ./VERSION" command to "git semver" this will return the version of the build.

We can pilot this with app-service-configurable when we set up the pipeline to build the service.
## Work Review

<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>
<tr>
<td>IT-17020</td>
<td>Status Update Requested: Clair Server</td>
<td></td>
<td>WIP</td>
</tr>
<tr>
<td>IT-17019</td>
<td>Permissions request for SonarCloud</td>
<td>Need to change Quality Profile / Quality Gate</td>
<td>WIP</td>
</tr>
</tbody>
</table>
Backlog Review
New Work for DevOps User Stories

- App Service Configurable – build automation (Jenkins)
  - version endpoint on SDK injected during the build

Device Services - current/coming

Open Source connectors (Current):
- Modbus (TCP/RTU)
- Virtual Device
- Random
- Grove C - ARM only
- SNMP
- MQTT

Open Source connectors (Future):
- BACnet (IP & MSTP) (Aug)
- OPC UA (Aug)
- Bluetooth (Aug)
- ONVIF Cameras (Oct)

Commercial connectors (Current):
- MQTT
- Modbus (TCP/RTU)
- BACnet (IP & MSTP)
- OPC UA
- GPS
- MEMS

Commercial connectors (Future):
- CAN SocketCAN, CANopen (Sept)
- EtherNet/IP (Sept)
- EtherCat (Sept)
- PROFINET (Sept)
- Siemens s5/s7 (Sept)
Meeting Minutes

• There were some optimizations being made to the BB testing but the job should have had a timeout set after 72 hours, so the fact that it ran for 5 days was an anomaly. Robin will check with the team and get back on the setting to ensure it has a timeout set up for the job going forward.

• Snyk demo was impressive from Tom / Rob yesterday in Security WG meeting
  • +1 from James, Jim, Trevor
  • Need to look into cost for Enterprise version with reporting and ensure that what was saw in the demo was included in the OSS version
  • Look into enabling the Jenkins Pipeline as a POC in Sandbox for additional evaluation
    • James to follow up with Rob / Snyk

• API Versioning discussion
  • We will set up the new app service configurable as a Pipeline job using git-semver as a POC
    • Lisa - Bring back a flow diagram which illustrates the way things work so we can visualize the flow
    • Set up the new Pipeline job for POC

• Lftools optimization / enhancement
  • Emilio shared the work he completed to optimize the Jenkins freestyle jobs
    • Introduction of a minor bug re: SAR should be fixed by EOW

• New device services expected to land by end of Aug
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>Complete</td>
</tr>
<tr>
<td>Add code and artifact signing with semantic versioning</td>
<td>Complete</td>
</tr>
</tbody>
</table>
| Conduct build performance optimizations by:  
  - Adding Pipelines for EdgeX Foundry base build images  
  - Allow base build images to be managed locally within Nexus  
  - Leverage PyPi Proxy for local pip dependencies | Complete |
| Explore static code analysis like Checkmarx, Coverity, GuardRails, Synk, SonarQube | Complete |

- 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
Edinburgh Release

Release Planning
Edinburgh Dates

• Freeze Date – May 28
• Release Date – June 20
• Press Release – July 11
• Dot Release – July 22
## Past / Future Agenda Topics

<table>
<thead>
<tr>
<th>WW27</th>
<th>No Meeting – US Holiday</th>
</tr>
</thead>
<tbody>
<tr>
<td>WW28</td>
<td></td>
</tr>
<tr>
<td>WW29</td>
<td></td>
</tr>
<tr>
<td>WW30</td>
<td></td>
</tr>
<tr>
<td>WW31</td>
<td>Shared Infrastructure LF Edge Umbrella Projects</td>
</tr>
<tr>
<td>WW32</td>
<td>Fuji Update / Performance Optimization lftools</td>
</tr>
<tr>
<td>WW33</td>
<td></td>
</tr>
<tr>
<td>WW34</td>
<td></td>
</tr>
<tr>
<td>WW35</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Athens Project – proxy server for go package dependencies</td>
</tr>
<tr>
<td></td>
<td>Community Involvement</td>
</tr>
</tbody>
</table>