# Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Topic</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 min</td>
<td>Work Review (GitHub Project)</td>
<td>James Gregg</td>
</tr>
<tr>
<td>10 min</td>
<td>Backlog Review</td>
<td>James Gregg</td>
</tr>
<tr>
<td>10-15 min</td>
<td>Config Options Discussion</td>
<td>Trevor Conn</td>
</tr>
<tr>
<td>10 min</td>
<td>Explore Update: Clair / Klar Docker Image Scanning</td>
<td>James Gregg / Pradeep</td>
</tr>
<tr>
<td></td>
<td>Opens</td>
<td>All</td>
</tr>
</tbody>
</table>
Attendees
EdgeX DevOps WG Update

• Jenkins Pipelines built for go-mod-core, go-mod-messaging, go-registry-core - PR#95  PR#10  PR#15
  • Note: Pipelines were created without LFTOOLS / Sigul
• Catalog of Basebuild Docker images now being built using Jenkins Pipelines with images successfully pushed to Nexus
  • Includes Kong-ARM64 basebuild image for blackbox security testing
• New automation built to take the human factor out of needing LF infrastructure team’s help for manually creating Jenkins Pipelines webhooks
• Issues:
  • LFTOOLS / Sigul defects not addressed by Linux Foundation
  • Roadmap for addressing technical debt not understood by Linux Foundation for build dependencies (python version)
<table>
<thead>
<tr>
<th>Helpdesk Ticket #</th>
<th>Description</th>
<th>Details</th>
<th>Status</th>
</tr>
</thead>
</table>
| 69830             | codecov.io config needed for edgex-go repo | Eric Ball has implemented codecov.io for edgex-go repo as of WW20  
Additional repos identified to include SDK repos – hold off on additional repos for now until we have it working with edgex-go | WIP (Eric) |
| 68377             | failed job related to timeout waiting for SSH | Eric Ball follow up with the team that owns VEXHOST – no progress Issue is with building new arm images – doesn’t affect edgex builds decision to leave it open for now. Circle back with VEXHOST team | WIP (Eric)  
Possible look at using a different OS for the builds Proposal to use Ubuntu instead of Centos ARM image |
| 71119             | need to extend sigul to include additional functionality – bugs identified with lftools / sigul | Eric Ball follow up with the team that owns VEXHOST – no progress Issue is with building new arm images – doesn’t affect edgex builds decision to leave it open for now. Circle back with VEXHOST team  
– sigul enhancements included in release last Thursday LFTOOLS release v0.23.1  
Lisa identified some defects Sigul enhancements being worked by LF resources with PR in Gerrit WW20 | WIP (Eric) |
| 71025             | TIG Performance Issues identified during testing on Sandbox | Tests are working in Sandbox but introduced OMM problems with AWS hosted TIG stack Redeployed the solution LF Infrastructure team engaged to resize the instances. | Resolved (Jordan)  
THANK YOU JORDAN CONWAY !! Outstanding PR for ci-management #338 |
| 71084             | Enable PyPI for EdgeX Jenkins caching of python dependencies | SPIKE to see what it would take to enable the PyPI Proxy for caching dependencies on Nexus | Resolved  
THANK YOU ERIC BALL !! Pulled in Fuji scope!! |
Backlog Review
• Expectation of enabling Redis
  • Propose to use the Docker Compose override to avoid having to create a new image
Architecture diagram of Clair hosted on AWS

- Clair uses PostgreSQL, so use Aurora PostgreSQL to host the Clair database. You deploy Clair as an ECS service with the Fargate launch type behind an Application Load Balancer.

- The Clair container is deployed in a private subnet behind the Application Load Balancer that is hosted in the public subnets. The private subnets must have a route to the internet using the NAT gateway, as Clair fetches the latest vulnerability information from multiple online sources.
Options to scan docker images for CVE detection

• Option #1: Clair hosted on AWS
  Clair server and Postgres DB is hosted on AWS & Klar runs within the Jenkins pipeline. Ideally the Jenkins job would be configured to include a state which scans the Docker images published in the EdgeX Nexus repo. A report of Critical CVEs is generated with detailed information within the Jenkins build log.

Option #2: Clair hosted on LF infrastructure
Similar solution to Option #1 but the Clair Server and DB are hosted on dedicated Linux Foundation infrastructure.

Option #3: Clair hosted on existing AWS TIG infrastructure

Need cost associated with options if hosted on AWS
Jenkins Pipeline job with Clair Image Scan Stage

Clair Docker Image Scan - 43s

- set +x & & aws ecr get-login --no-include-email --region us-west-2 — Shell Script
- Downloading Clair client Klar-2.4.0 — Print Message
- wget https://github.com/optiopay/klar/releases/download/v2.4.0/klar-2.4.0-linux-amd64 — Shell Script
- mv ./klar-2.4.0-linux-amd64 ./klar — Shell Script
- chmod u+x klar — Shell Script
- Running Clair scan on the Docker Image — Print Message
- ./klar $DOCKER_IMAGE — Shell Script
Clair Output – Jenkins log

The Clair package provides a collection of binary utilities for the manipulation of object code in various object file formats. It includes the ar, as, gmp, ld, nm, objcopy, objdump, ranlib, ranlib, dll, string, strip, and nm2line utilities. Security fix(2) * bjinutil: integer underflow or overflow via an ELF file with a crafted ELF or a block in a library library (CVE-2018-7560) * bjinutil: integer underflow or overflow via an ELF file with a crafted ELF or a block in a library library (CVE-2018-7560) * bjinutil: malloc pointer dereference in write_and_read_in_function in write.m (resulting in crash (CVE-2018-7560) * bjinutil: crash in elf.c (ElfSectionFromGref) with crafted executable (CVE-2018-7560) * bjinutil: heap-based buffer over-read in dwarf.c (process_cross_index) allows for denial of service via crafted file (CVE-2018-18772) * bjinutil: NULL pointer dereference in elf.c (ElfSectionFromGref) allows for denial of service via crafted file (CVE-2018-18772) * bjinutil: out of bounds access in get_gref (CVE-2018-18772) * bjinutil: Uncontrolled Resource Consumption in execution of m (CVE-2018-18772) For more details about the security issue(s), including the impact, a CVSS score, and other related information, refer to the CVE page(s) listed in the References section. Additional Changes: For detailed information on changes in this release, see the Red Hat Enterprise Linux 7.6 Release Notes linked from the References section.

https://access.redhat.com/errata/RHEL-2018-13032

The system packages contain a system and service manager for Linux, compatible with the SysV and LSB init scripts. It provides aggressive parallelization capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of services, and uses a track of processes using Linux groups. In addition, it supports snapshotting and restoring of the system state, handles mount and mount options, and implements an elaborate transactional dependency-based service control logic. It can also work as a drop-in replacement for SysVinit. Security Fix(es): * system: memory leak in JournaliseServer.c introduced by fix for CVE-2018-10104 (CVE-2018-5833) For more details about the security issue(s), including the impact, a CVSS score, and other related information, refer to the CVE page(s) listed in the References section.

https://access.redhat.com/errata/RHEL-2018-02811

The system packages contain a system and service manager for Linux, compatible with the SysV and LSB init scripts. It provides aggressive parallelization capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of services, and uses a track of processes using Linux groups. In addition, it supports snapshotting and restoring of the system state, handles mount and mount options, and implements an elaborate transactional dependency-based service control logic. It can also work as a drop-in replacement for SysVinit. Security Fix(es): * system: memory leak in JournaliseServer.c introduced by fix for CVE-2018-10104 (CVE-2018-5833) For more details about the security issue(s), including the impact, a CVSS score, and other related information, refer to the CVE page(s) listed in the References section.
Meeting Minutes

- Pradeep presented the material related to Clair / Klar explore.
  - Next Steps:
    - Need Cost associated with proposed Options
    - Need to review best option with LF Infrastructure team
    - Need to take the proposed solution forward to the TSC for a vote if there are additional costs / overhead to implement
Fuji Planning

Scope Discussions
Fuji – DevOps

In

• Static code analysis tool identified and integrated into the EdgeX Jenkins Pipeline for Docker image scanning
  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
Edinburgh Release

Release Planning
Edinburgh Dates

• Freeze Date – May 28
• Release Date – June 20
# Future Agenda Topics

<table>
<thead>
<tr>
<th>WW14</th>
<th>Documentation migration – edgex-go user documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>WW14</td>
<td>Topics for Fuji F2F</td>
</tr>
<tr>
<td></td>
<td>Jenkins Pipelines for EdgeX services</td>
</tr>
<tr>
<td>WW15</td>
<td>Review Aqua Microscanner – Image scanning tool for Vulnerabilities</td>
</tr>
<tr>
<td>WW16</td>
<td>NVIDIA – Security tooling within CodePipeline (Trevor request) 4/18/19</td>
</tr>
<tr>
<td>WW17</td>
<td></td>
</tr>
<tr>
<td>WW18</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Demo Clair / Klar</td>
</tr>
<tr>
<td></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>