## Agenda

<table>
<thead>
<tr>
<th>Time</th>
<th>Topic</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Min</td>
<td>DevOps Updates</td>
<td>James</td>
</tr>
<tr>
<td>10 Min</td>
<td>Jenkins Production Stability Issues</td>
<td>Eric Ball</td>
</tr>
<tr>
<td>20 Min</td>
<td>Unit Testing Jenkins Pipeline Code (Groovy)</td>
<td>Emilio</td>
</tr>
<tr>
<td>10 Min</td>
<td>DevOps Year in Review Recap</td>
<td>James</td>
</tr>
<tr>
<td>10 Min</td>
<td>Opens</td>
<td></td>
</tr>
</tbody>
</table>
DevOps WG Update

Geneva

• Jenkins Transformation to Pipelines
  • GitHub Org plugin
    • Plugin was introduced / tested in Sandbox – no issues
    • Plugin was introduced in Production – started seeing build issues
• New Pipeline for go-mod-bootstrap with git-semver
• Nexus Life Cycle Policy TSC approved and JSD ticket submitted
• Jenkins Pipeline Training published to GitHub
• CommunityBridge - Advanced Snyk Reporting
  • New Product Owner from LF identified - Meeting last Friday
    https://jira.linuxfoundation.org/servicedesk/customer/portal/4/SUPPORT-715
Vulnerability Detection – Advanced Snyk Reporting
DevOps WG Update

• Git commit message enforcement
  • Coming out of the EdgeX Geneva F2F and Architects meetings, DevOps agreed to take on a SPIKE to look at tooling / automation that would be helpful for the project in terms of Commit Messages (enforce standard format that aligns to conventional commit standard or agreed upon format as per EdgeX written guidance)

Examples:
  • [https://itnext.io/enforcing-commit-templates-8cf3dbfe2510](https://itnext.io/enforcing-commit-templates-8cf3dbfe2510)
    • Example of pre-commit bash scripts.
    • Examples of client-side and server-side ruby scripts
  • [https://dev.to/prahladyeri/how-to-enforce-conventional-commit-messages-using-git-hooks-2bmk](https://dev.to/prahladyeri/how-to-enforce-conventional-commit-messages-using-git-hooks-2bmk)
    • Pre-commit custom python script placed in the .git/hooks directory and committed to the repo
  • [https://github.com/marketplace/commit-message-lint](https://github.com/marketplace/commit-message-lint)
    • Commit message lint
  • [https://probot.github.io/apps/](https://probot.github.io/apps/)
    • Commitlint - [https://probot.github.io/apps/commitlint/](https://probot.github.io/apps/commitlint/)
Root Cause Analysis: Jenkins Stability Issues

- PAT (Jenkins-GitHub credentials) was regenerated Tuesday as part of the work involved to enable GitHub Org plugin
- Issues not immediately observed, but after ~15 mins afterwards, logs indicate credential failures (401)
- SSH key hasn’t been updated
- “Test Connections” – GitHub and Global Pipelines are successful
- Sandbox isn't showing the same issues because it isn’t actually connected to the Pull Requests
  - Not the exactly the same as Production

*** RECHECK ***
Corrective Action Plan

• Focused Tiger Team to troubleshoot the problem
• Clear communications in both Slack and Jira
• GitHub Plugins Version and Last Update checked
• Change Management Review
• Verification of Credentials / Secrets

Continuous Improvement Plan
1. ELK – AI look for “string look up for 401”
2. Use filebeat to stream to ELK (logstash)
3. Jenkins SPLUNKlike plugin
   1. Explore Jenkins-SPLUNKlike integration
4. Process: Open a JSD ticket (backlog issue needed)
5. Training: Build issues infrastructure related debug / log details to look for before reporting
Unit Testing of Jenkins Pipeline Code

**edgex-global-pipelines** Unit Tests

- Leverage **Jenkins-Spock** - library to help with testing Jenkins pipeline scripts and functions written in Groovy.
  - Based off of Spock Unit Testing Framework
  - Leverages Maven build automation tool
    - Requires JDK (1.8+) and Maven (3.3+)
- Mocking Jenkins dependencies
  - Every Pipeline step, and variable that exists are setup as a Spock Mock objects
  - Only pipeline steps (plugins) that are added as dependencies will get mocked
    - Add dependencies to Project Object Model (pom.xml)

**edgex-global-pipelines** Unit Tests

- Pull Request
  - [https://github.com/edgexfoundry/edgex-global-pipelines/pull/69](https://github.com/edgexfoundry/edgex-global-pipelines/pull/69)
- Unit Test Summary
  - ~89% Coverage for Groovy code
- Two types of Unit Tests
  - Data-Driven Tests
    - Exercise the same test code multiple times, with varying inputs and expected results
  - Behavior Driven Tests
    - Test behavior instead of implementation

![Test Results](image-url)
Unit Testing of Jenkins Pipeline Code (continued)

Gaps

- **Jenkins-Spock** currently lacks test coverage reporting
- Does not support Declarative Pipeline
  - Will require a leveraging a different capability to test this
    - JenkinsPipelineUnit testing framework
- No code linting capability included
- Library includes basic documentation
  - Initial implementation relied on trial/error

Next Steps

- Create CI build image to facilitate testing
  - Initial compilation can take several minutes to complete
    - Pull down all required dependencies (pom.xml)
- Add build automation integration
  - edgex-global-pipelines git commit status checks
- Refactor source code to make testing easier
  - Implement consistent environment variable access across all libraries
  - Consolidate and reuse shared methods (DRY)
  - Implement smaller functions where possible (keep the code simple)
  - Return and error conditions should be checked early (avoid deep nesting)
EdgeX DevOps Year in Review

- Dedicated Clair Server implemented on AWS
- TIG stack on AWS resized to address performance issues
- Adoption of Coverity for Code Coverage for all Go Code
- Introduction of SonarCloud for Additional Code Coverage, Code Smells, Quality Gates
- Semantic Versioning (git-semver)
- Maturing Release EdgeX Foundry Release Process
  - Project Tracker using GitHub Projects
- Jenkins Pipeline Transformation
  - Base build images
  - Global Libraries
  - Jenkinsfile
  - GitHub Org Plugin
- Introduction of Snyk
  - Advanced Snyk Reporting Explore with Cost Savings via ReUse Opportunity through Community Bridge
  - Automated Snyk Reports to SIR Team via Slack / Email
Backlog Review
Meeting Minutes

Discussion
Decision to continue forward with the configuration of the Snyk Advanced Reporting. We want to see the data so we know what to fix and can prioritize the work.

There’s a concern that the Snyk notifications aren’t getting to the SIR Team email distribution list. UPDATE: Determined that the Snyk portal is configured to send emails to the invited Administrators associated with the edgex-Jenkins project.

Jenkins stability issues wwi were resolved – Thanks Eric Ball!!
Good discussion with some ideas for CI

Excellent work and presentation of the Unit Testing for Jenkins Pipeline code (Groovy) – Thank you Emilio !!!

Opens
AR: James – take a look at the Snyk Slack integration to ensure that all of the Snyk alerts with findings are getting into the #snyk-reports channel.
UPDATE – The Slack integration is only used for updates since last scan of the repo, not weekly reports. It only sends new information based on new CVE findings.

AR: James – add CI options to backlog for consideration / prioritization
# Past / Future Agenda Topics

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>