



EDGE X FOUNDRY™

# DevOps Working Group

Thursday December 19, 2019

# Agenda

Time	Topic	Owner
5 Min	DevOps Updates	James
10 Min	Jenkins Production Stability Issues	Eric Ball
20 Min	Unit Testing Jenkins Pipeline Code (Groovy)	Emilio
10 Min	DevOps Year in Review Recap	James
10 Min	Opens	

# Attendees

Participants (13)

Find a participant

JG	James Gregg (Intel) (Me)		
ER	Emilio Reyes		
	Brandon Forster		
EB	Eric Ball		
E	Ernesto		
JW	Jim White		
LG	Lenny Goodell (Intel)		
LR	Lisa Rashidi-Ranjbar		
	Michael Estrin		
MJ	Mike Johanson		
TC	Trevor Conn		
VS	Vishwas Sp		
W	Walt (Intel)		



# 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

**VULNERABILITY DETECTION**

Projects

**EdgeCheck**

Vulnerability Report

0 HIGH SEVERITY	0 MEDIUM SEVERITY	0 LOW SEVERITY	0 TOTAL ISSUES
0 Fixable	0 Fixable	0 Fixable	0 Fixable

Last updated December 13, 2019

**EdgeCheckRepo**

Vulnerability Report

0 HIGH SEVERITY	0 MEDIUM SEVERITY	0 LOW SEVERITY	0 TOTAL ISSUES
0 Fixable	0 Fixable	0 Fixable	0 Fixable

Last updated December 13, 2019

**MacEdge**

Vulnerability Report

0 HIGH SEVERITY	0 MEDIUM SEVERITY	0 LOW SEVERITY	0 TOTAL ISSUES
0 Fixable	0 Fixable	0 Fixable	0 Fixable

Last updated December 13, 2019

**EdgeX Foundry**

Vulnerability Report

217 HIGH SEVERITY	91 MEDIUM SEVERITY	16 LOW SEVERITY	324 TOTAL ISSUES
211 Fixable	59 Fixable	16 Fixable	286 Fixable

Last updated December 13, 2019

**VULNERABILITY DETECTION**

Dependency tree

support-rulesengine

- com.fastenxm1.jackson.core.jackson-annotations@2.6.7 Apache-2.0
- log4j:log4j@1.2.17 Apache-2.0
- org.edgexfoundry.core-command-client@0.2.0 Unknown
- org.edgexfoundry.core-domain@0.5.0 Unknown
  - org.apache.commons:commons-lang3@3.5 Apache-2.0
  - org.edgexfoundry.core-exception@0.5.0 Unknown
  - org.edgexfoundry.export-domain@0.5.0 Unknown
  - org.edgexfoundry.support-logging-client@0.5.0 Unknown
- org.freemarker:freemarker@2.3.25-incubating Apache-2.0
- org.jboss.resteasy:resteasy-client@3.0.13.Final Apache-2.0
  - org.jboss.logging:jboss-logging@3.3.0.Final Apache-2.0
  - org.jboss.resteasy:resteasy-jaxrs@3.0.13.Final MEDIUM Apache-2.0
    - commons-io:commons-io@2.4 Apache-2.0
    - javax.activation:activation@1.1.1 CDDL-1.1
    - net.jpjpc:jcip-annotations@1.0 Unknown
    - org.jboss.spec.javax.annotation:jboss-annotations-api\_1.2\_spec@1.0.0.Final CDDL-1.1, GPL-2.0
    - org.jboss.spec.javax.ws.rs:jboss-jaxrs-api\_2.0\_spec@1.0.0.Final CDDL-1.1, GPL-3.0
- org.jboss.resteasy:resteasy-jackson-provider@3.0.13.Final Apache-2.0
  - org.codehaus.jackson:jackson-core-asl@1.9.12 Apache-2.0
  - org.codehaus.jackson:jackson-jaxrs@1.9.12 Apache-2.0, LGPL-2.1
  - org.codehaus.jackson:jackson-mapper-asl@1.9.12 MEDIUM Apache-2.0

# DevOps WG Update

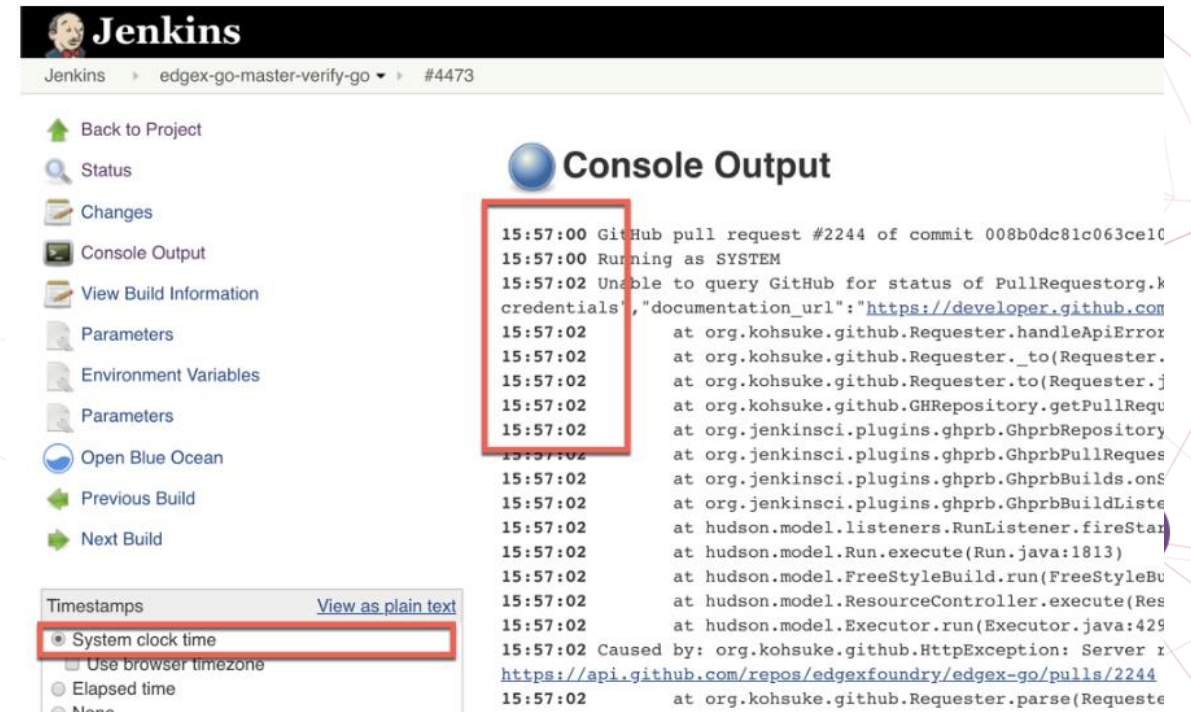
- Git commit message enforcement
  - Coming out of the EdgeX Geneva F2F and Architects meetings, DevOps agreed to take on a SPIKE to look at tooling / automation that would be helpful for the project in terms of Commit Messages (enforce standard format that aligns to conventional commit standard or agreed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

## Examples:

- <https://itnext.io/enforcing-commit-templates-8cf3dbfe2510>
  - Example of pre-commit bash scripts.
- <https://git-scm.com/book/en/v2/Customizing-Git-An-Example-Git-Enforced-Policy>
  - Examples of client-side and server-side ruby scripts
- <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>
  - Commit message lint  
[https://github.com/SystangoTechnologies/commit-message-lint/wiki/Commit-Message-Lint---2.0?installation\\_id=5828561&setup\\_action=install](https://github.com/SystangoTechnologies/commit-message-lint/wiki/Commit-Message-Lint---2.0?installation_id=5828561&setup_action=install)
- <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



The screenshot shows the Jenkins interface for a build named 'edgex-go-master-verify-go' with ID '#4473'. The 'Console Output' tab is selected, displaying a log of system messages. A red box highlights a specific error message: 'Unable to query GitHub for status of PullRequestorg.k credentials', which is a 401 Unauthorized error. The error message includes the GitHub URL 'https://developer.github.com' and the GitHub API URL 'https://api.github.com/repos/edgexfoundry/edgex-go/pulls/2244'. The error is caused by the 'org.kohsuke.github.Requester.parse' method.

```
15:57:00 GitHub pull request #2244 of commit 008b0dc81c063ce10
15:57:00 Running as SYSTEM
15:57:02 Unable to query GitHub for status of PullRequestorg.k
credentials", "documentation_url": "https://developer.github.com
15:57:02 at org.kohsuke.github.Requester.handleApiError
15:57:02 at org.kohsuke.github.Requester._to(Requester.
15:57:02 at org.kohsuke.github.Requester.to(Requester.j
15:57:02 at org.kohsuke.github.GHRepository.getPullRequ
15:57:02 at org.jenkinsci.plugins.ghprb.GhprbRepository
15:57:02 at org.jenkinsci.plugins.ghprb.GhprbPullReques
15:57:02 at org.jenkinsci.plugins.ghprb.GhprbBuilds.on$
15:57:02 at org.jenkinsci.plugins.ghprb.GhprbBuildListe
15:57:02 at hudson.model.listeners.RunListener.fireStar
15:57:02 at hudson.model.Run.execute(Run.java:1813)
15:57:02 at hudson.model.FreeStyleBuild.run(FreeStyleBu
15:57:02 at hudson.model.ResourceController.execute(Res
15:57:02 at hudson.model.Executor.run(Executor.java:429
15:57:02 Caused by: org.kohsuke.github.HttpException: Server r
https://api.github.com/repos/edgexfoundry/edgex-go/pulls/2244
15:57:02 at org.kohsuke.github.Requester.parse(Requeste
```

\*\*\* 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>
- 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

```
[WARNING] Tests run: 87, Failures: 0, Errors: 0, Skipped: 2
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:37 min
```

# 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

The screenshot displays a Jira backlog for the 'EdgeX Foundry Project'. The interface includes a top navigation bar with 'Repositories 96', 'Packages', 'People 68', 'Teams 97', and 'Projects 12'. Below this, a 'DevOps WG' filter is active, updated 2 days ago. The backlog is organized into five columns:

- New Issues (6 items):** Includes tasks like 'New build automation for device-rest-go', 'New build automation for device-bluetooth-c', and 'New build automation for device-opcu-c'. Statuses include 'enhancement', 'geneva', 'blocked', and 'phase 1'.
- Release Backlog (19 items):** Features 'Spike - Identify a tool / plugin (bot) to check PR / Issue status', 'Spike - Identify a tool / plugin to enforce use of "conventional commit" messages', and 'Build Automation for Open Horizons'. Statuses include 'geneva', 'spike', and 'blocked'.
- In Progress (9 items):** Contains 'New build automation for go-mod-bootstrap', 'Bot Explore for Pull Requests, Issues, Commit Message Enforcements', and 'Github Org/Branch source plugin configuration'. Statuses include 'geneva', 'phase 1', and 'phase 2'.
- QA/Code Review (4 items):** Lists 'New build automation for device-bacnet-c', 'Update Jenkinsfile to use edgeXBuildDocker function', and 'Update Jenkinsfile to use edgeXBuildDocker() function (fuji)'. Statuses include 'enhancement', 'fuji', 'geneva', and 'hold'.
- Done (2 items):** Shows 'Validate and fix any incompatibilities with the Github branch source plugin' and 'New global pipelines for GoLang and Docker builds'. Statuses include 'geneva', 'phase 1', and 'Changes approved'.

Each item in the backlog provides details such as the issue ID, description, assignee, and status. The interface also includes a search bar for filtering cards and a 'Manage' button at the bottom of each column.

# 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 ww51 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



# EDGE X FOUNDRY™

# Past / Future Agenda Topics