



EDGE X FOUNDRY™

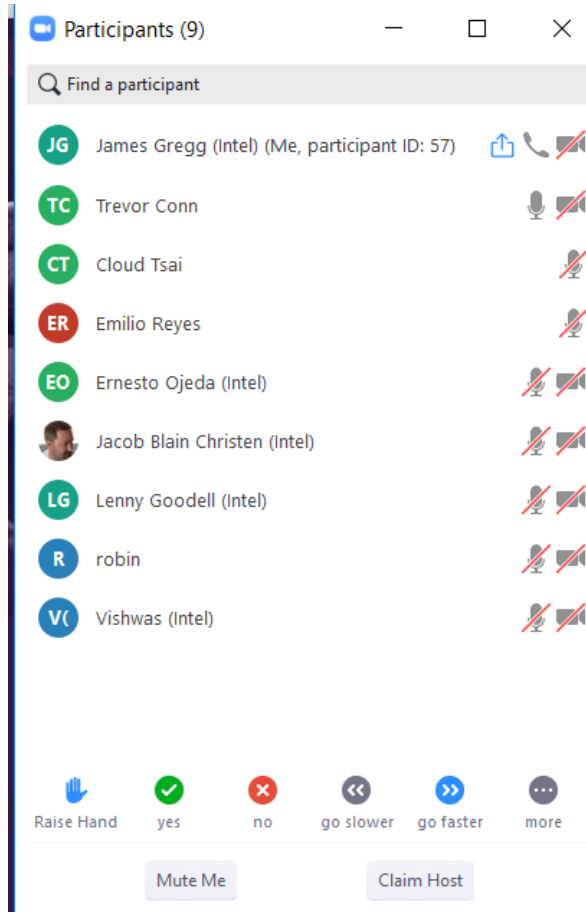
DevOps Working Group

Thursday July 18, 2019

Agenda

Time	Topic	Owner
10 min	Edinburgh : Dot Release Discussion / Planning 7/22	James Gregg / Eric Ball / Jim White
10 Min	Fuji Update	James Gregg
10 min	Performance Testing – LF Configuration EB IAC	Emilio
10 Min	Clair – Docker Image Scan Reports / Demo	Ernesto
	Opens – Nexus Questions	All

Attendees



A screenshot of a Zoom meeting participants list. The window title is "Participants (9)". At the top is a search bar labeled "Find a participant". Below it is a list of nine participants, each with a circular profile picture icon, their name, and their affiliation (if any). To the right of each name are icons for chat, video, and audio. At the bottom of the list are several control icons: a hand for "Raise Hand", a green checkmark for "yes", a red X for "no", a double left arrow for "go slower", a double right arrow for "go faster", and a three-dot menu for "more". Below these icons are two buttons: "Mute Me" and "Claim Host".

Initials	Name	Organization	Audio	Video	Chat
JG	James Gregg	(Intel)	On	Off	On
TC	Trevor Conn		Off	Off	On
CT	Cloud Tsai		Off	Off	On
ER	Emilio Reyes		Off	Off	On
EO	Ernesto Ojeda	(Intel)	Off	Off	On
	Jacob Blain Christen	(Intel)	Off	Off	On
LG	Lenny Goodell	(Intel)	Off	Off	On
R	robin		Off	Off	On
V	Vishwas	(Intel)	Off	Off	On



EdgeX DevOps WG Update (Edinburgh Dot Release)

Edinburgh Dot Release (1.0.1)

Decision: TSC Meeting voted and Approved on 7/17
Planned Release now scheduled for 7/22

- List of Artifacts impacted
 - go-mod-messaging (this is the) module that doesn't get released.

All other services need to be released (1.0.1)

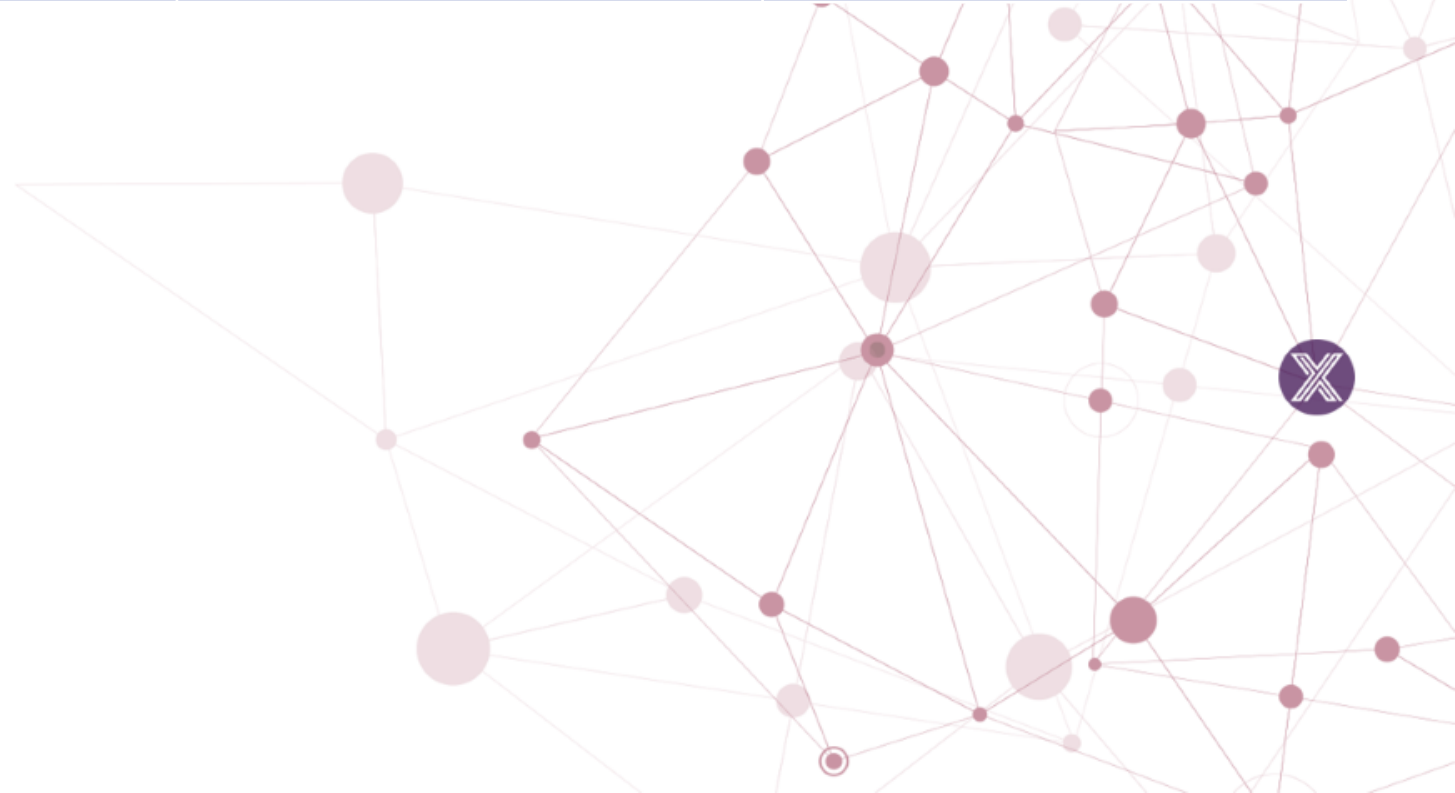
- 1.) We need to merge a fix into the master and Edinburgh branches of go-mod-messaging
- 2.) The Edinburgh branch of both edgex-go and app-functions-sdk will need to consume the Edinburgh branch of go-mod-messaging
- 3.) We will need to republish the artifacts for all services in edgex-go for the Edinburgh release as well as app-functions-sdk
- 4.) We will need to adjust the Edinburgh docker-compose files to utilize the version numbers applied to the new images by the above actions.

DevOps WG Update

- **Fuji Scope**
- Container Scanning (Clair Server landing request) has now been committed to by Linux Foundation with resources planned and funding approved for hosting on AWS.
 - Progress made on Clair reporting - Demo and Discussion in DevOps WG
- Codecov.io integration completed on all repos with tests
- Static Code Analysis Tools - decision to defer decision on SonarCloud
 - Plan to review analysis completed to date in the Security WG
- GitHub repo audit completed and actions taken to address empty repos
 - Next steps: audit Teams of Committers
- Review of the Linux Foundation Infrastructure as Code re: Performance Testing completed
- Release Czar Proposal to use GitHub Projects for all Issues as an aide for the release mgmt.

Work Review

Helpdesk Ticket #	Description	Details	Status
75648	Dedicated Clair server for EdgeX	Pending decision on strategy for K8s + cost / availability of resources with LF	WIP



Backlog Review

EdgeX Foundry Project

Repositories 88 Packages People 55 Teams 89 Projects 14

DevOps WG Updated 4 hours ago

Filter cards + Add cards Fullscreen

3 Icebox

- regex for isReleaseStream is overly broad
edgex-global-pipelines#26 opened by dweomer
- Snap builds should use unbuffer for better output logging
ci-management#369 opened by anonymouse64
- edgeXSemver leaves command to caller but doesn't allow specifying the version/image to use
edgex-global-pipelines#15 opened by dweomer
bug

8 New Issues

- Slack integration with Jenkins Pipeline for Clair reporting.
ci-build-images#54 opened by jamesgregg
enhancement
Geneva
- New build automation for device-bacnet-c
ci-management#451 opened by jamesgregg
enhancement **fuji**
Fuji
- Update snapcraft inside docker to use new setup from snapcraft
ci-management#449 opened by anonymouse64
enhancement
- New image for sonar scanner needed for use with SonarCloud
ci-build-images#45 opened by jamesgregg
enhancement **good first issue**
Fuji
- New Build Automation needed for application-service-configurable

Automated as To do Manage

3 Release Backlog

- Delete repo - jenkins_pipeline_presentation (FUJI)
edgex-global-pipelines#32 opened by jamesgregg
- <https://github.com/edgexfoundry-holding/go-mod-core-security>
ON-HOLD (FUJI)
May not be needed - Trevor will check and advise
Added by jamesgregg
- Placeholder for snap global function in edgex-global-pipelines (FUJI)
US4581
US4580
Added by jamesgregg

1 In Progress

- Clair Server landing - LF Infrastructure or AWS decision and architecture / \$\$ with TSC approval (@jamesgregg)
ci-management#439 opened by jamesgregg
enhancement **fuji**
Fuji

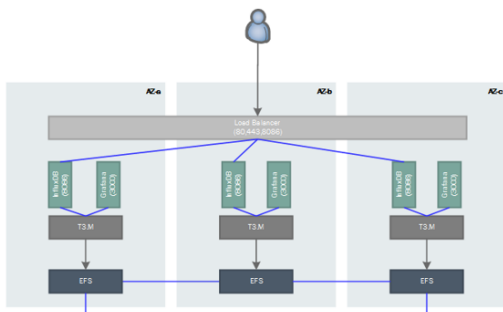
1 QA/Code Review

- WIP: Add sigul signing to go lang binaries
ci-management#318 opened by JPWKU
Changes requested

Automated as In progress Manage

Analysis / Review of LF TIG IAC

- QA-Test WG reporting missing data when querying Grafana
 - Issue is inconsistent between test runs
 - Analysis of logs reveal
 - File corruption
 - Out of memory
 - “Influxdb Starting” which suggests container crashed and was spinning back up
 - Emilio / Jacob reviewed the Terraform (infrastructure as code)
 - Requested input from LF as an opportunity to vet our assumptions
- Assumed AS-IS architecture



- Amazon Elastic Beanstalk
- Load-balancing, Autoscaling Environment
 - Instances: Min: 3, Max: 5
- Multiple Availability Zones
- Leveraging EFS storage for DB
- T3.Medium Instances, 1.5GB/Container

/efs/influxdb-data
/efs/grafana-storage

Recommendations

- Switch to single instance
 - InfluxDB OSS does not support clustering, HA for open-source requires custom layer to facilitate replication of data to all InfluxDB nodes
 - Switch to Elastic Beanstalk Single-Instance Environment
- Scale up single instance
 - InfluxDB sizing recommendations for AWS:
 - <https://docs.influxdata.com/influxdb/v1.7/introduction/installation/#hosting-influxdb-oss-on-aws>
 - Minimum of 8GB RAM
 - Use R4 or M4.large
 - Vendor recommends SSDs
- Leverage EBS volume for InfluxDB data storage instead of EFS
 - Locking on network mount problematic

Nice To Haves

- Read-only access to dashboards to monitor performance
 - Elastic Beanstalk environment health console
- Access to CloudWatch logs

Clair Docker Image Scan Reports



Use of Clair vulnerability scanning within the EdgeX pipelines.

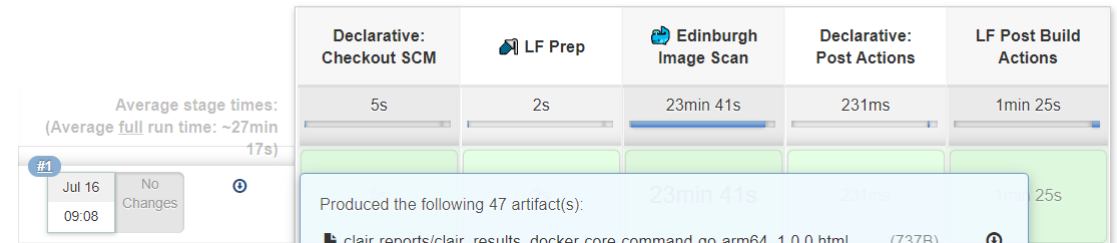
- Two approaches:
 - Embedded within pipelines. When images are pushed to nexus, a scan occurs and an HTML report is archived with the build.
 - Continuously scans via audit reports, part of the new ci-audit-pipelines repository.
- Results will be archived in Jenkins.
- [ASK] Should results be actionable or reporting only.

Geneva scope

- 🌟 [ASK] Potential Slack integration when vulnerabilities are discovered.

Demo & Discussion

Stage View



Permalinks

- [Last build \(#1\), 2 days 2 hr ago](#)
- [Last stable build \(#1\), 2 days 2 hr ago](#)
- [Last successful build \(#1\), 2 days 2 hr ago](#)
- [Last completed build \(#1\), 2 days 2 hr ago](#)

Scan Results for Docker Image [edgexfoundry/docker-device-grove-c-arm64:1.0.0]

Analysing 16 layers
 Got results from Clair API v1
 Found 2 vulnerabilities
 Medium: 2

SEVERITY	NAME	FEATURENAME	FEATUREVERSION	FIXEDBY	DESCRIPTION	LINK
Medium	CVE-2019-3836	gnutls	3.6.2-r0	3.6.7-r0		https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-3836
Medium	CVE-2019-3829	gnutls	3.6.2-r0	3.6.7-r0		https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-3829

Use of Nexus for Test Automation Framework

- Seeking input on use of Nexus as an artifact management solution for test reports
- As part of the QA/Test, we are working on to introduce TAF which is based on robot/python.
- The artifacts (reports, logs, snapshots etc) generated as part of the test execution has to be stored in some repository and to be published.
- Looking for a way to utilize the existing Nexus infrastructure
 - We may need to create a new repository
 - To begin with, we need a way to experiment in Sandbox environment.

Meeting Minutes

Analysis of LF IAC of EB implementation on AWS

- After review of the recommendation, Cloud suggested to try the recommendations on the current AWS implementation
 - Decision: Work with LF to review the assessment, collaborate, and try the recommendations if acceptable to the LF Infra team.
Will try to coordinate via existing JSD Ticket #

Clair Reporting

Decision: nothing actionable for now – just the reports

We will add a user story to the backlog for Slack integration (Geneva) but if possible pull it

Nexus for Test Automation Framework Reports

Need to submit a ticket so that the Nexus repo can be created and settings updated on Jenkins Sandbox

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

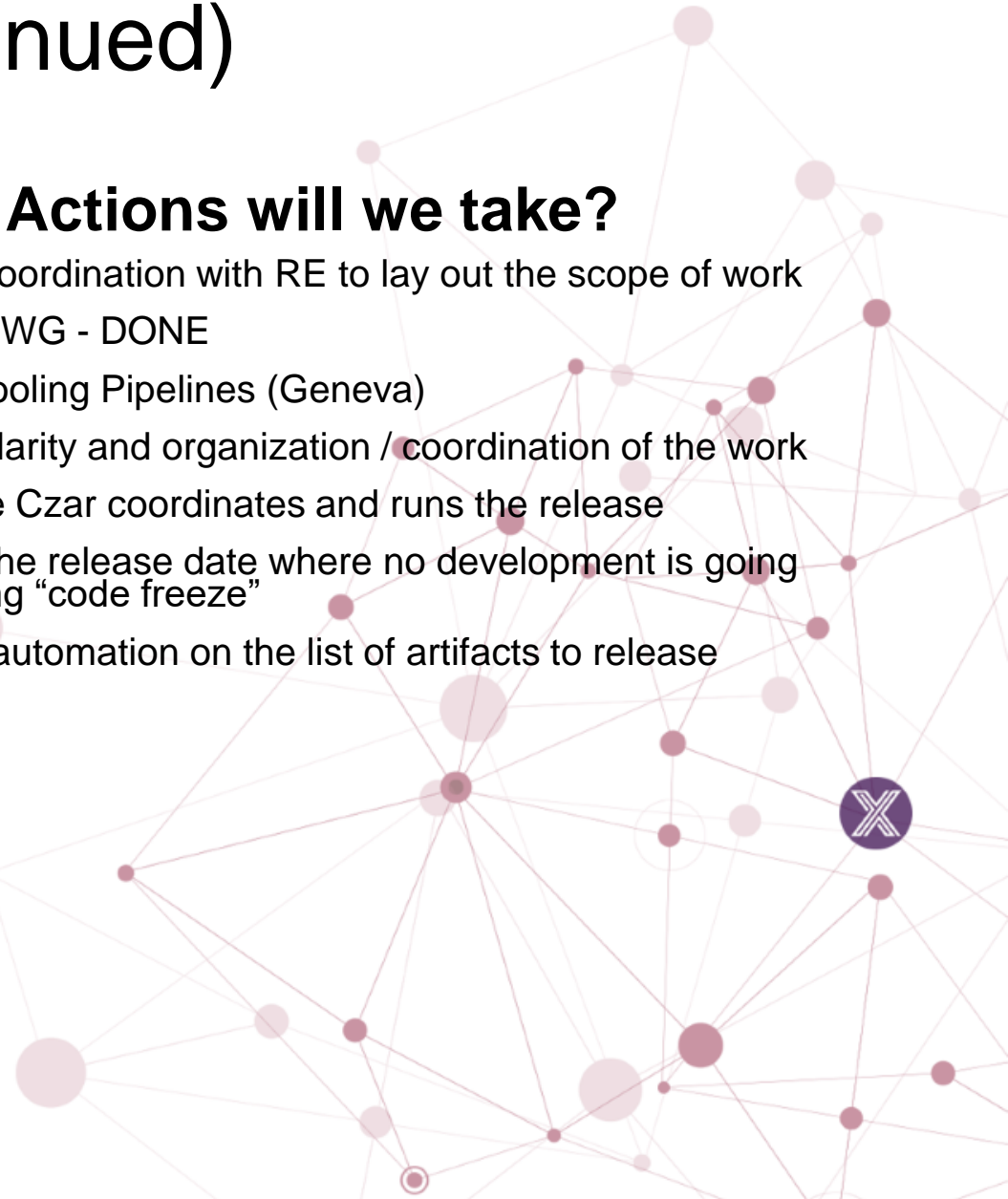
Edinburgh Retrospective (continued)

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





EDGE X FOUNDRY™

Fuji Planning

Scope Discussions

Fuji – DevOps

In

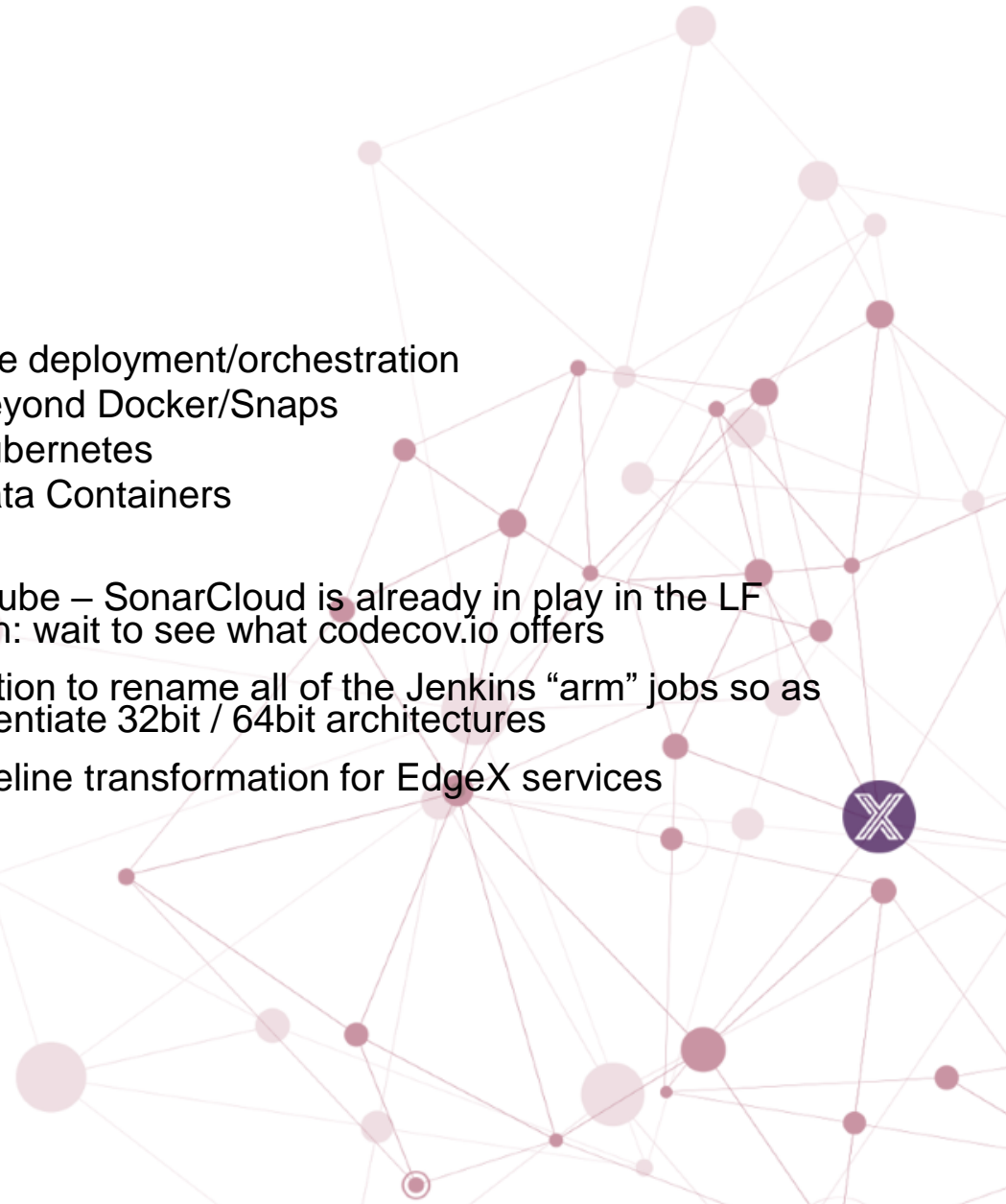
- Static code analysis tool identified and integrated into the EdgeX Jenkins Pipeline for Docker image scanning (Clair Server)

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	
Add static artifact analysis into the EdgeX Jenkins Pipeline (analysis of Docker /runtime artifacts, not the source code)	
Add code and artifact signing with semantic versioning	
Conduct build performance optimizations by: <ul style="list-style-type: none"> • 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 	
Explore static code analysis like Checkmarx, Coverity, GuardRails, Synk, SonarQube	

- Clair Server landing at **Risk** for Fuji
 - Work Around will be to use Intel hosted Clair server until decision is made by LF to support landing dedicated infrastructure
 - Work related to automating the scans as part of the build, will defer to Geneva scope
- 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



EDGE X FOUNDRY™

Edinburgh Release

Release Planning

Edinburgh Dates

- Freeze Date – May 28
- Release Date – June 20
- Press Release – July 11
- Dot Release – July 22





EDGE X FOUNDRY™

Past / Future Agenda Topics

WW27	No Meeting – US Holiday
WW28	
WW29	
WW30	
WW31	
WW32	
WW33	
WW34	
WW35	
	Athens Project – proxy server for go package dependencies
	Community Involvement