

# Stop Crawling and Start Walking

*Transitioning from Continuous  
Integration to Continuous Delivery*

Lisa Ranjbar (@Iranjbar)

**DEVOPS  
WORLD**  
by CloudBees

# About EdgeX Foundry

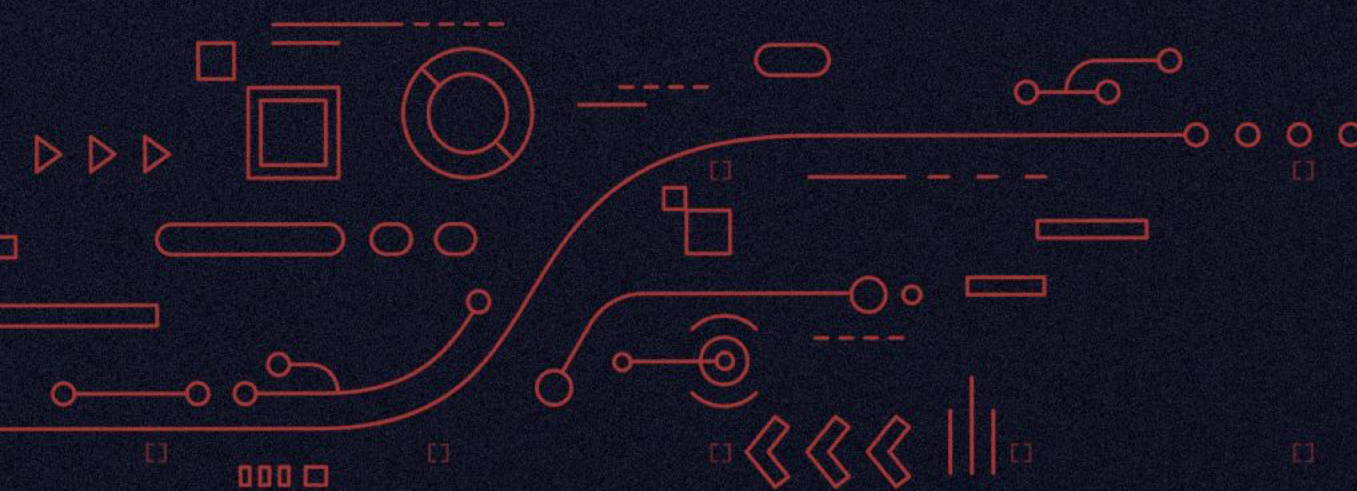
---

- An open source, vendor neutral project (and ecosystem)
- A microservice, loosely coupled software framework for IoT edge computing
- Hardware and OS agnostic
- Started in 2017

EDGE X FOUNDRY™

OLFEDGE

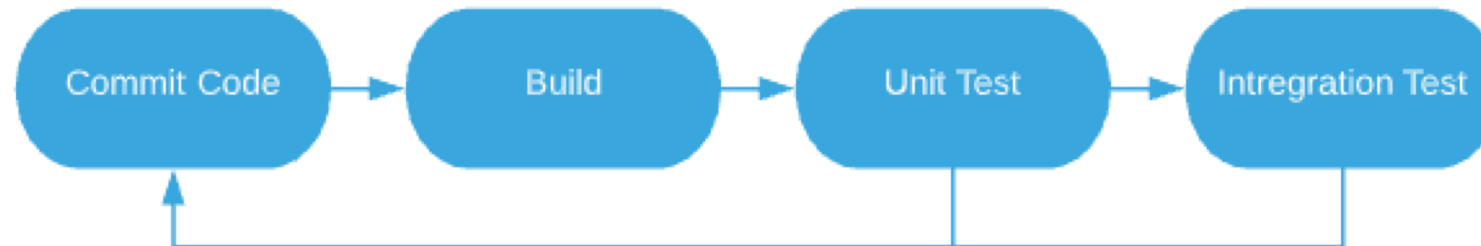
# What is Continuous Delivery Anyway?



# Continuous Integration

---

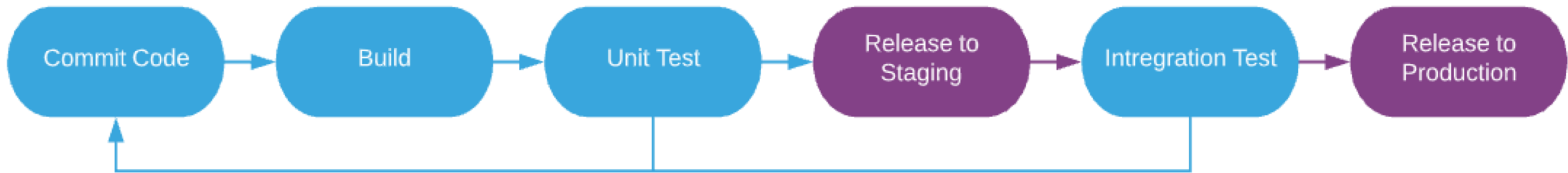
- CI System and SCM Setup
- Automated Build and Tests
- Limited Artifact Staging
- Manual Version Tracking
- Manual Artifact Delivery
- Manual Release Management



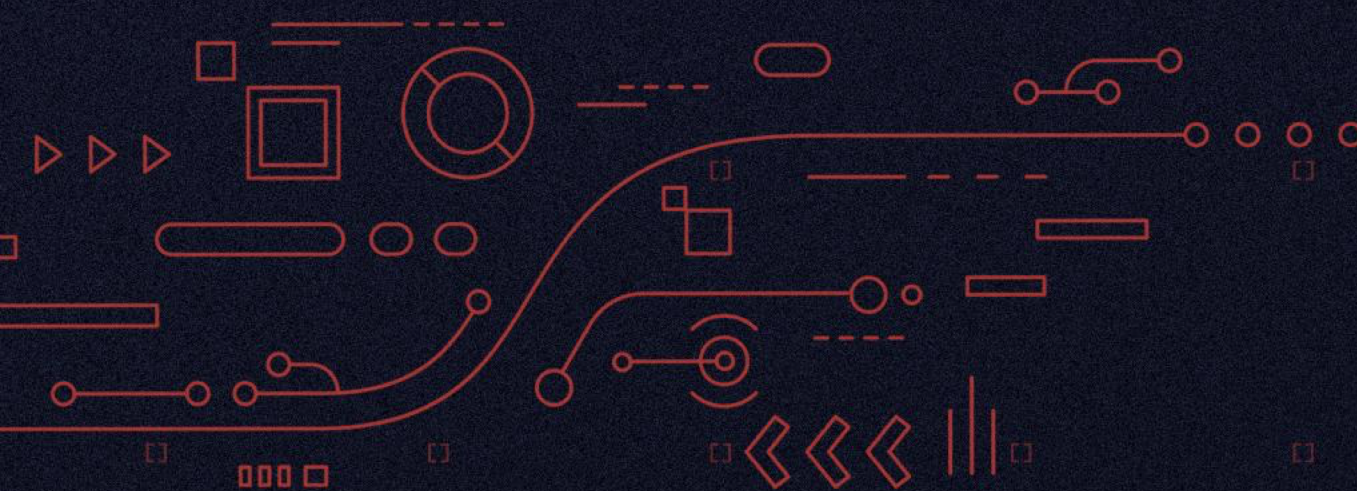
# Continuous Delivery

---

- Continuous Integration
- Automated Artifact Staging
- Automated Version Tracking
- Automated Artifact Delivery
- Automated Release “Button”
- Semi-automated Release Management



# No One Size Fits All Solution



**DEVOPS  
WORLD**  
*by CloudBees*

# Releasing EdgeX Foundry

---

- A microservice framework
- 34 active repositories
- Hardware and OS agnostic
- Over 100 images to release
- Not everything is released at the same time

EDGE X FOUNDRY™

ALFEDGE

# The Problems

---

- Test results lagged behind the code changes up to 48 hours
- Too many jobs per repository (20~30)
- Making new jobs was pretty labor intensive
- Test coverage wasn't tracked
- Built images were tagged with incorrect versions
- Errors were hard to trace to the commits and code changes



# The Path to Continuous Delivery

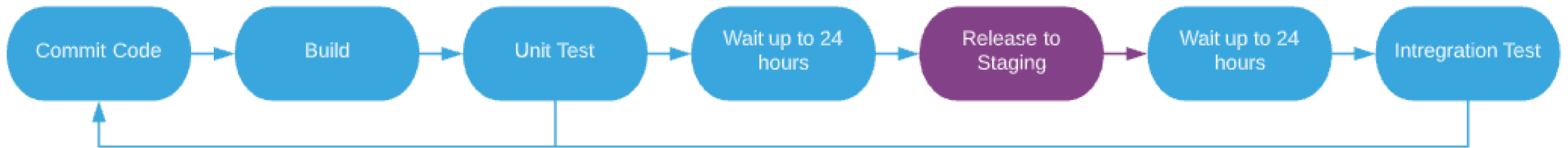
## Part 1: Migration to Jenkins Pipelines



# Migration to Jenkins Pipelines

- Jenkins Freestyle jobs were schedule driven
- Each repository had several freestyle jobs
- Changes had to be made in several places to add new features

S	W	Name ↓	Last Success	Last Failure	Last Duration
🔴	☁️	<a href="#">device-grove-c-snap-master-stage-snap</a>	9 mo 13 days - <a href="#">#127</a>	3 hr 20 min - <a href="#">#407</a>	7 min 4 sec
🔴	☁️	<a href="#">device-grove-c-snap-master-stage-snap-arm</a>	9 mo 13 days - <a href="#">#127</a>	10 hr - <a href="#">#407</a>	13 min
🔵	☀️	<a href="#">device-grove-c-snap-master-verify-snap</a>	1 yr 0 mo - <a href="#">#4</a>	N/A	6 min 2 sec
🔵	☀️	<a href="#">device-grove-c-snap-master-verify-snap-arm</a>	1 yr 0 mo - <a href="#">#4</a>	N/A	16 min
🔵	☀️	<a href="#">device-grove-c-snap-release-snap</a>	1 yr 1 mo - <a href="#">#7</a>	N/A	3 min 16 sec
🔴	☁️	<a href="#">edgex-go-snap-master-verify-snap</a>	1 mo 4 days - <a href="#">#529</a>	11 days - <a href="#">#540</a>	1 hr 0 min
🔴	☁️	<a href="#">edgex-go-snap-master-verify-snap-arm</a>	1 mo 4 days - <a href="#">#531</a>	11 days - <a href="#">#542</a>	1 hr 9 min
🔵	☀️	<a href="#">edgex-go-snap-release-snap</a>	15 days - <a href="#">#60</a>	N/A	3 min 16 sec
🔴	☁️	<a href="#">edgex-ui-go-snap-master-stage-snap</a>	N/A	17 hr - <a href="#">#109</a>	3 min 1 sec

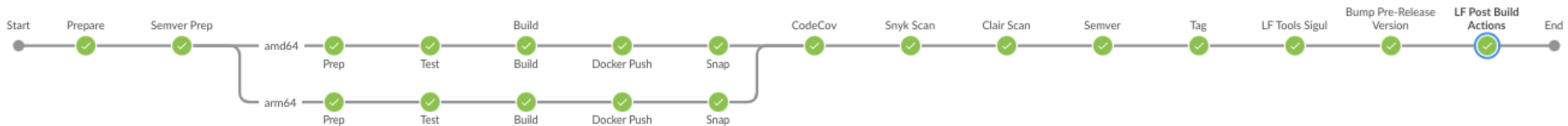


# Migration to Jenkins Pipelines

- Created a Jenkins Shared Library for common functionality
- Encapsulated common build functionality into functions
- Use Jenkinsfiles to define jobs
- Jenkins jobs are triggered on Github events

Example Jenkinsfile

```
edgeXBuildGoApp (  
    project: 'device-modbus-go',  
    goVersion: '1.13',  
    buildSnap: true  
)
```



# Migration to Jenkins Pipelines

















- Jenkins Github Branch Source Plugin
- Jenkins jobs automatically created from Jenkinsfile in the repository
- Give developers the power to create jobs

<https://github.com/jenkinsci/github-branch-source-plugin>

## EdgeX Foundry Project

Folder name: edgexfoundry

### Repositories (34)

S	W	Name ↓	Description
		<a href="#">app-functions-sdk-go</a>	Owner: Applications WG
		<a href="#">app-service-configurable</a>	Owner: Applications WG
		<a href="#">cd-management</a>	Owner: DevOps WG
		<a href="#">ci-build-images</a>	Owner: DevOps WG
		<a href="#">device-bacnet-c</a>	Owner: Device WG
		<a href="#">device-camera-go</a>	Owner: Device WG
		<a href="#">device-grove-c</a>	Owner: Device WG
		<a href="#">device-modbus-go</a>	Owner: Device WG

# Migration to Jenkins Pipelines



## device-modbus-go

Owner: Device WG

Branches (2)

Pull Requests (2)

S	W	Name ↓	Last Success	Last Failure	Last Duration
		<a href="#">edgex-modbus-simulator</a>	3 mo 25 days - <a href="#">#3</a>	3 mo 25 days - <a href="#">#2</a>	15 min
		<a href="#">master</a>	18 days - <a href="#">#16</a>	1 mo 19 days - <a href="#">#14</a>	13 min



## device-modbus-go

Owner: Device WG

Branches (2)

Pull Requests (2)

S	W	Name ↓	Last Success	Last Failure	Last Duration
		<a href="#">PR-160</a>	5 days 13 hr - <a href="#">#1</a>	N/A	15 min
		<a href="#">PR-94</a>	N/A	6 mo 18 days - <a href="#">#1</a>	1.4 sec

# The Path to Continuous Delivery

---

- Simple way for developers to make new jobs
- The amount of Jenkins jobs is reduced
- Test results no longer lag behind code changes



# The Problems

---

- ~~Test results lagged behind the code changes up to 48 hours~~
- ~~Too many jobs per repository (20~30)~~
- ~~Making new jobs was pretty labor intensive~~
- Test coverage wasn't tracked
- Built images were tagged with incorrect versions
- Errors were hard to trace to the commits and code changes

# The Path to Continuous Delivery

## Part 2: Adding Code Coverage

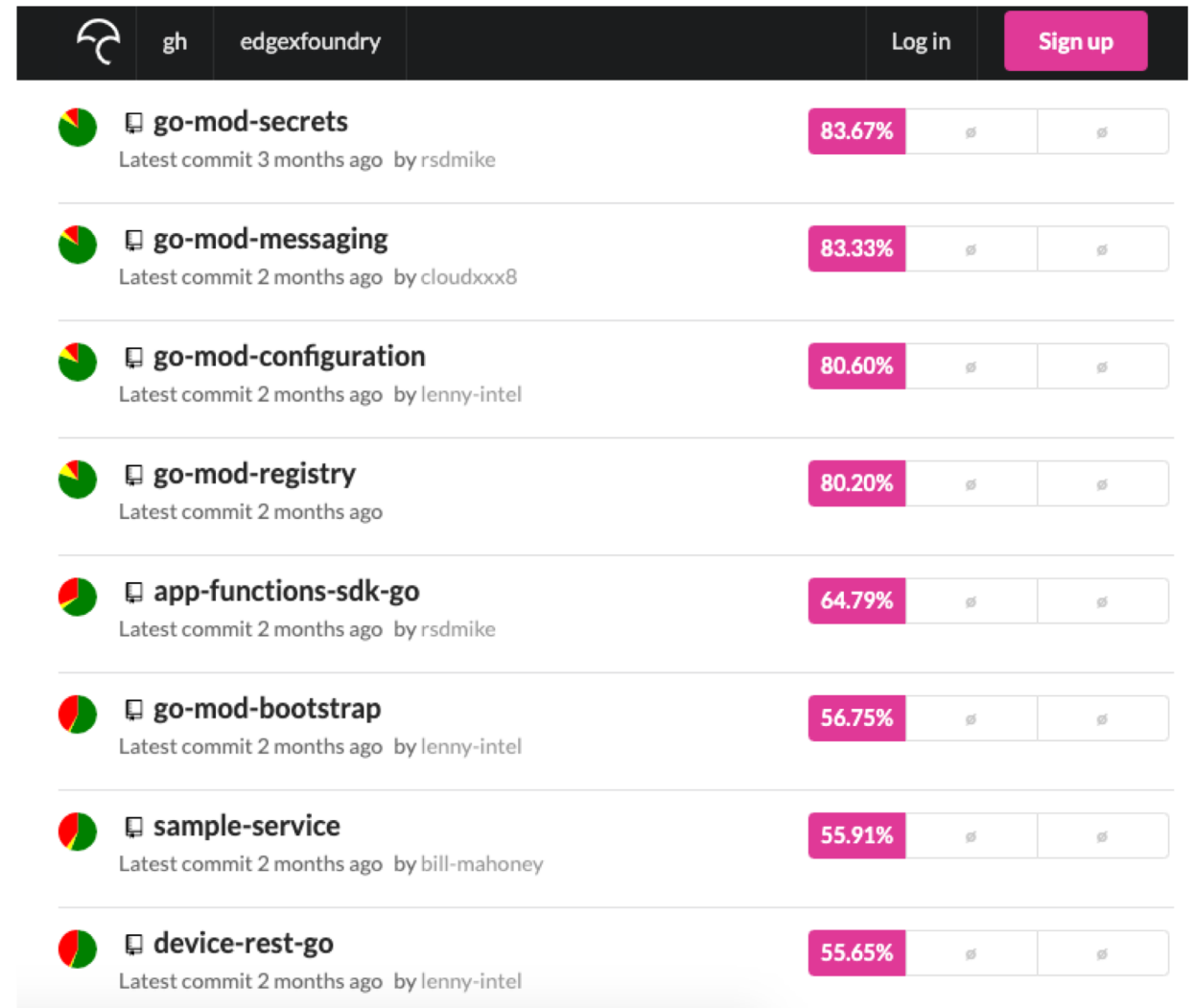




# Adding Code Coverage

- Minimal test coverage in the project
- Needed a tool that would work for both Go and C
- Decided to use Codecov.io
- Integrates with Github

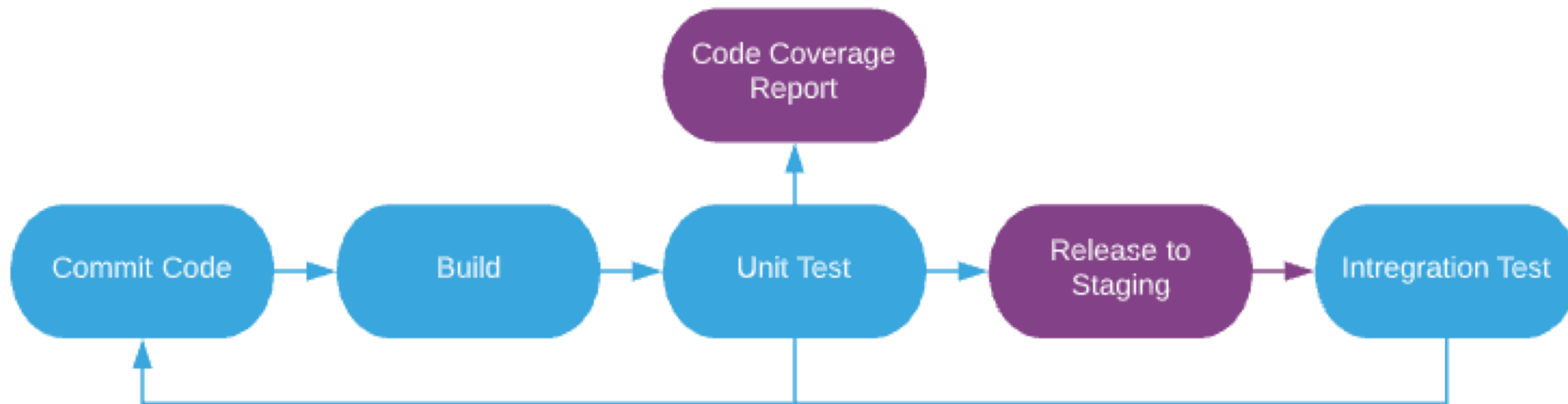
<https://codecov.io/gh/edgexfoundry>



# The Path to Continuous Delivery

---

- With code coverage and other testing reports we gain more confidence



# The Problems

---

- ~~Test results lagged behind the code changes up to 48 hours~~
- ~~Too many jobs per repository (20~30)~~
- ~~Making new jobs was pretty labor intensive~~
- ~~Test coverage wasn't tracked~~
- Built images were tagged with incorrect versions
- Errors were hard to trace to the commits and code changes

# The Path to Continuous Delivery

## Part 3: Fixing Traceability



# Git Semver

- Track semantic versions outside of the main commit stream
- Next semantic version is stored locally in `.semver` and saved in a detached semver branch.
- Supports tracking versions for multiple branches

<https://github.com/edgexfoundry/git-semver>

```
lranjbar@Lisas-MacBook-Air ~/g/e/device-modbus-go master
> git semver
1.2.2-dev.3

lranjbar@Lisas-MacBook-Air ~/g/e/device-modbus-go master
> git semver bump final
1.2.2
```

edgexfoundry / device-modbus-go

<> Code Issues 5 Pull requests 2 Actions

semver 7 branches 8 tags

This branch is 29 commits ahead, 140 commits behind master.

edgex-jenkins	semver(master): 1.2.2-dev.3
fuji	semver(fuji): 1.1.2-dev.1
master	semver(master): 1.2.2-dev.3

# Git Semver

- Used to keep Github tags and docker image versions in sync

edgexfoundry / device-modbus-go

<> Code    ! Issues 5    🔗 Pull requests 2

Releases    **Tags**

Tags

**v1.2.2-dev.2** ...

🕒 19 days ago    🔗 ede6161    📦 zip    📦 tar.gz

**v1.2.2-dev.1** ...

🕒 26 days ago    🔗 0d9dcf3    📦 zip    📦 tar.gz

v2/docker-device-modbus-go/manifests/1.2.2-dev.2

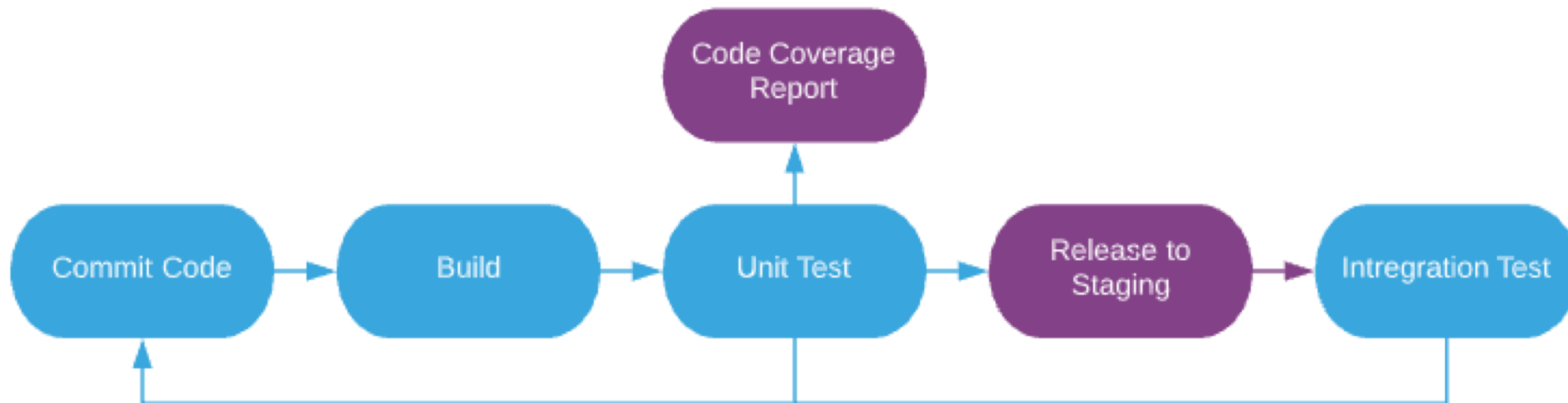
Delete asset

Summary	
Repository	docker.staging
Format	docker
Component Name	docker-device-modbus-go
Component Version	1.2.2-dev.2
Path	<a href="#">v2/docker-device-modbus-go/manifests/1.2.2-dev.2</a>
Content type	application/vnd.docker.distribution.manifest.v2+json
File size	944 bytes
Blob created	Wed Jul 22 2020 00:12:25 GMT-0700 (Mountain Standard Time)

# The Path to Continuous Delivery

---

- We have the ability to check if our integration tests are testing correct versions



# The Problems

---

- ~~Test results lagged behind the code changes up to 48 hours~~
- ~~Too many jobs per repository (20~30)~~
- ~~Making new jobs was pretty labor intensive~~
- ~~Test coverage wasn't tracked~~
- ~~Built images were tagged with incorrect versions~~
- ~~Errors were hard to trace to the commits and code changes~~



# The Path to Continuous Delivery

## Part 3: Adding the Control Plane

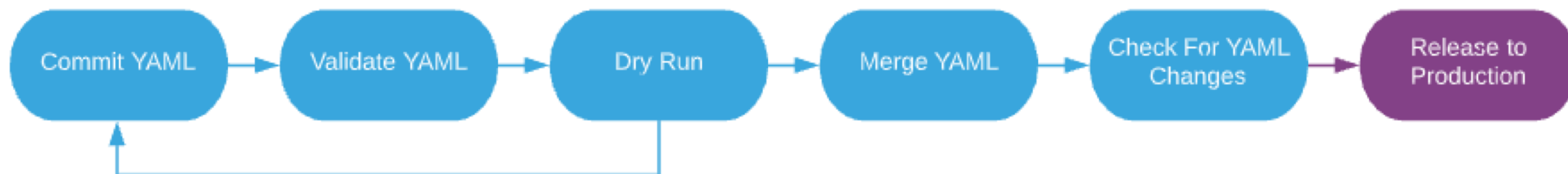


# Enter cd-management

---

- Extended our shared library to have release functions
- Uses YAML files to track metadata about the release
- Trigger the release actions on merge into this branch
- GitOps Model

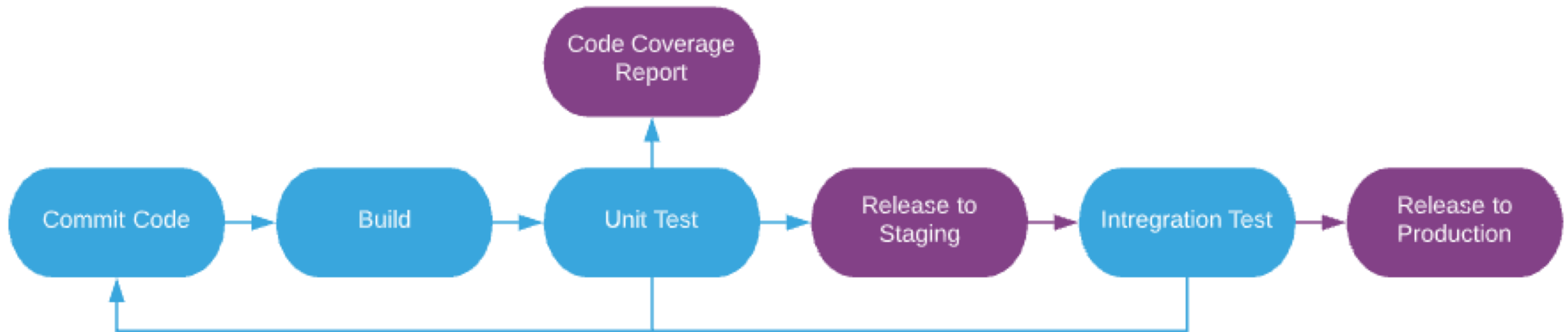
<https://github.com/edgexfoundry/cd-management/tree/release>



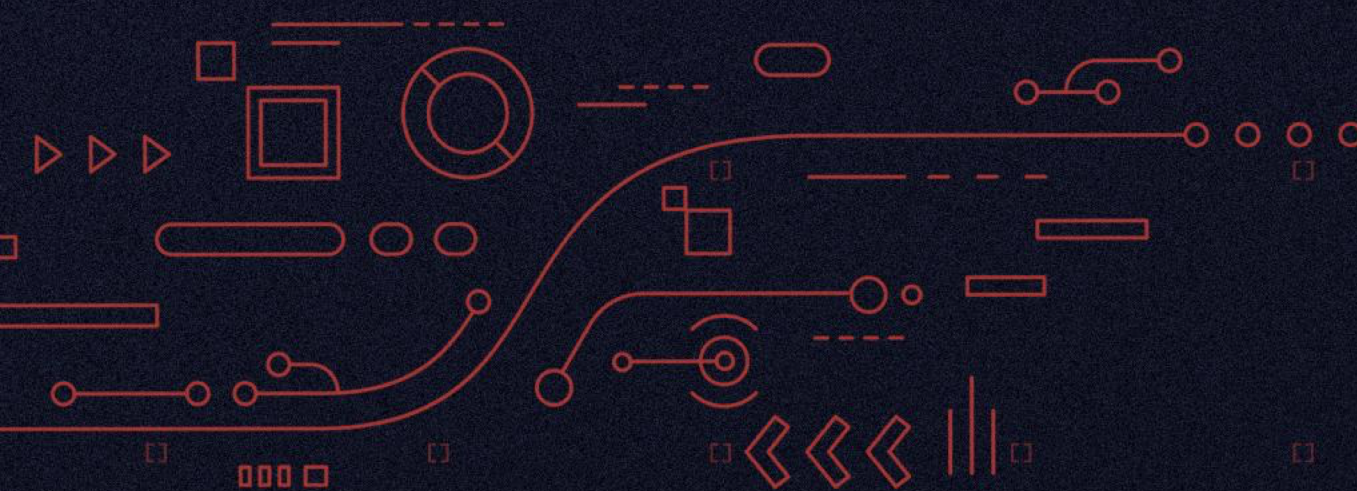
# The Path to Continuous Delivery

---

- With the addition of cd-management we have our “automated release” button



# Recap



# Recap

---

- Continuous Integration has a manual release process
- Continuous Delivery has a semi-automated release process
- Understand the problems with your CI implementation
- Improve your CI implementation to gain confidence
- Add your control plane for CD



**DEVOPS  
WORLD**  
by CloudBees