

# Building our own custom Code Insight tool at Form3

**Adelina Simion & Ross McFarlane**

Conf42 DevSecOps



## Introductions

# Our Customers and Partners



# Our People

260 employees  
51 nationalities  
22 countries  
100% remote

# Our Investors

\$220m in  
investment so far



---

**01** Engineering at Form3

---

**02** Code Insight requirements

---

**03** Code Insight architecture

---

**04** Driving Adoption

---

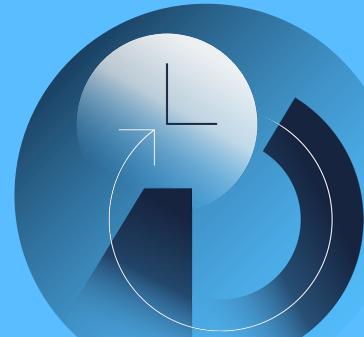
**05** Insights from Code Insight

# Engineering at Form3

# Delivering code at scale 🚀

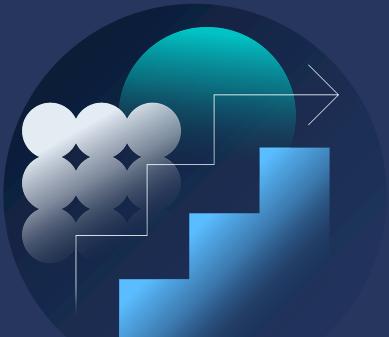
### Large number of repos

We have over 500 repositories in different languages. Some are not actively maintained, while some are under development



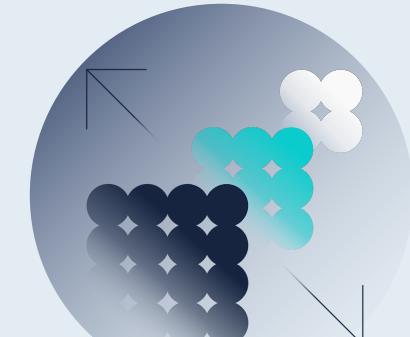
### Growing engineering teams

Our teams are in hypergrowth. We have new teams and new engineers contributing to the codebase at rapid pace

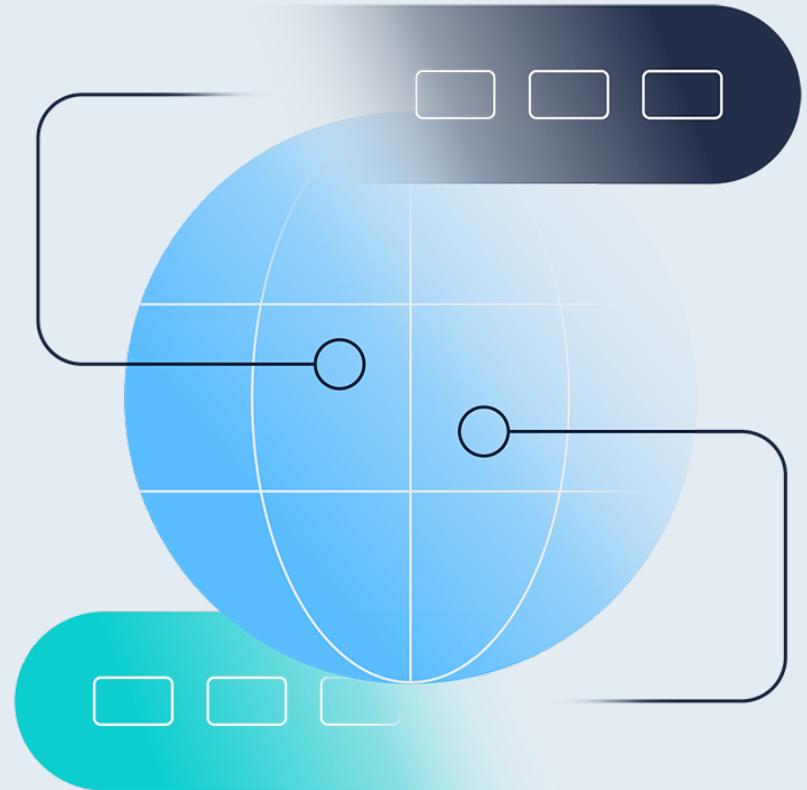


### Highest security standards

Our platform is compliant with the highest standards of security. All our repositories should remain free of vulnerabilities



The DevSecOps mindset is crucial at Form3. Our teams own every part of the delivery flow, including the security of their repositories and services



# Code analysis at Form3

- Used <https://github.com/coinbase/salus> for static analysis of our GitHub repos
- Scans ran in one Docker container
- Relatively heavyweight containers of 1GB+



# First solution - SecScan

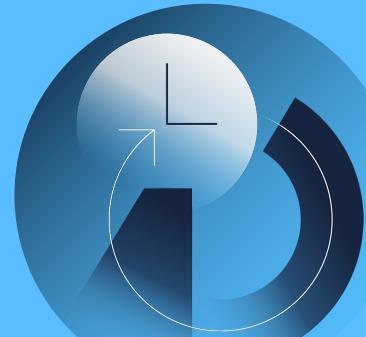
- Our first custom solution to provide standardised scanning across our GitHub repos
- Scans ran in a SecScan docker container
- Tokens for service scanning for each repo
- Teams configured their scans in a Makefile or Travis YAML file on each repository



# Problems with SecScan

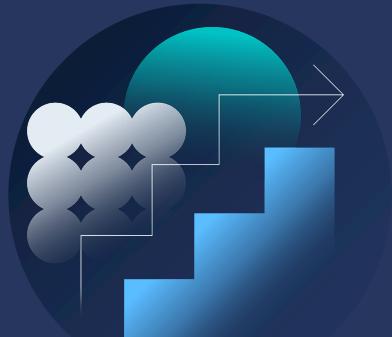
## No enforcement

Difficult to enforce rules and code standards across our repositories as setup on new repositories is not mandatory



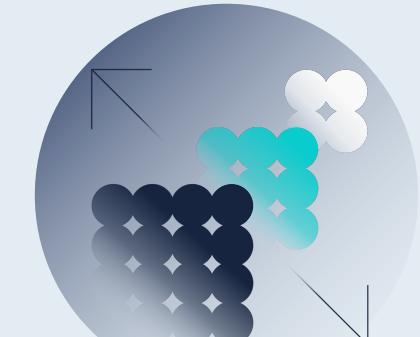
## No visibility

Repositories marking their own homework means attackers could bypass the scan.  
Scans run on code commits means possible undetected issues



## Lots of maintenance

Initial configuration and maintenance required on each repository when rolling out updates



# Code Insight requirements

## Code Insight requirements

# A better solution was needed!



The InfoSec team identified some key requirements for a new solution

Automatic enforcement

Automatically chosen scans

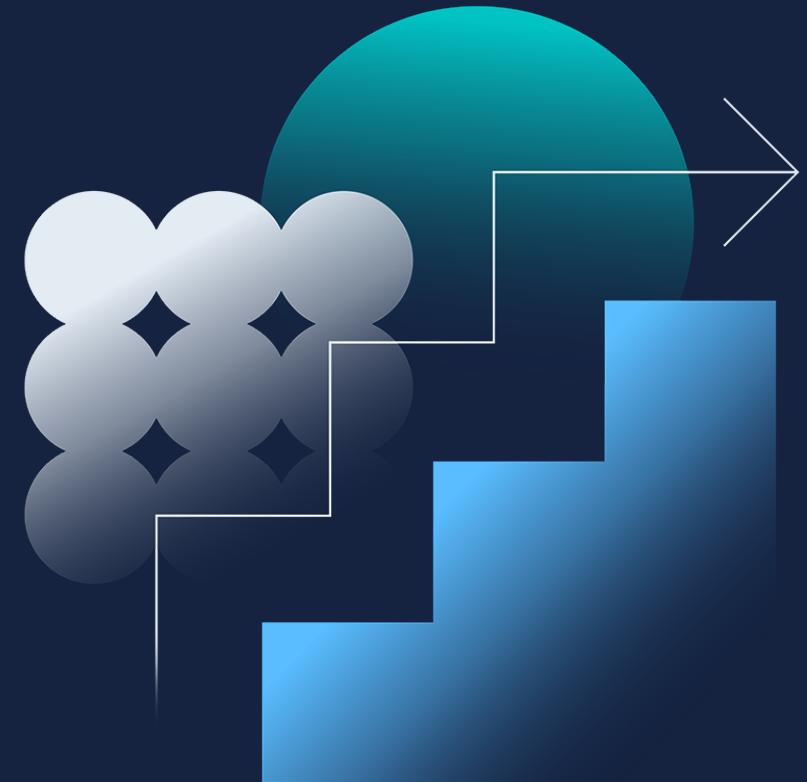
Centralised config

Report on vulnerabilities

Scan repositories regularly

Easily build into development pipeline

The Code Insight project was kicked off to implement these requirements and address some of the issues with SecScan 😊

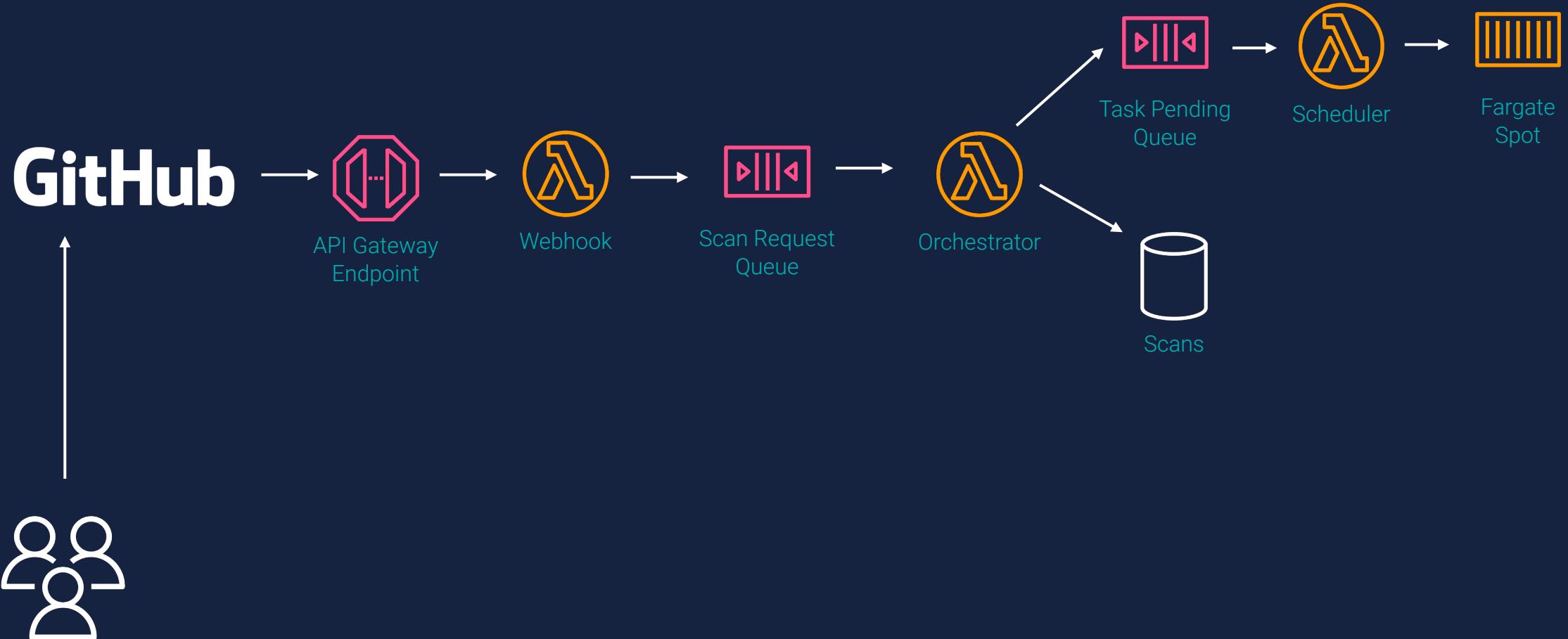


# Teamwork makes the dream work!

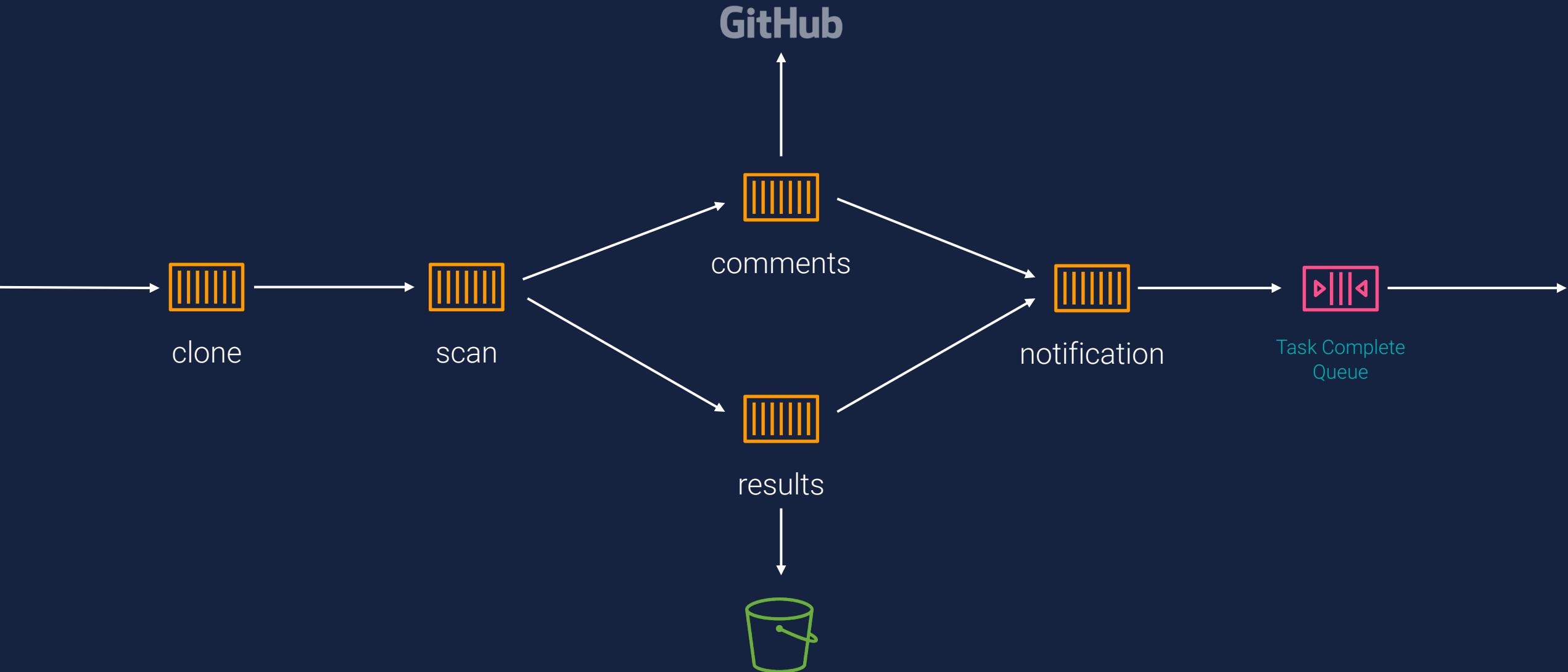


# Code Insight architecture

# Architecture



# Architecture



# Architecture



Cloudwatch Event



API Gateway  
Endpoint

Webhook

Scan Request  
Queue

Notification  
Queue

Notifier

Task Complete  
Queue

Orchestrator

Task Pending  
Queue

Scans

Suites



Scheduler



Fargate  
Spot



Results Bucket

# Inspecting your checks

> WIP

✓ Form3 Code Insight

■ Code Insight

**Form3 Code Insight / Code Insight**  
completed on 18 Aug in 2m 10s

**Passed (failures allowed)**

⚠ Scanning has detected problems, but these were ignored and will not block the PR from being merged.

Code-Insight is the Form3 centralised source code scanning solution.

Your code is inspected to determine the languages used and run the relevant checks.

If you encounter problems with this check, please contact #infosec-engineering-team on Slack, including the link to this page in your message.

DETAILS

**Tasks**

	SCAN	NAME	STATE
✓	6e261dc8-c24b-4546-bfaf-e0ec593c4c00	Squealer (All)	complete
⚠	7796e1b1-41ad-48d4-90e8-cf03c971738e	Hadolint (Dockerfile)	soft-failure
⚠	efbdbdc7-6320-4501-bb50-14b2a1f525d2	Snyk (Dockerfile)	soft-failure
⚠	592d8ecd-8417-4c43-ad8b-98050dbbce19	Lint (Golang)	soft-failure
⚠	918f676f-0abc-44b4-8bb9-b8a93356e441	Snyk (Golang)	soft-failure

# Inspecting your checks



## Code Insight



form3tech/k8s-tenant-s3sync

Global > Suite c248d992-9f8c-4614-a931-50a13f355e13

Leaderboard

✓ Scanning has completed with no issues.

Branch: master

Commit: dbbbfb87f1addc2c58a4e8113243a3281380421a8

Retry

# Suite ID: c248d992-9f8c-4614-a931-50a13f355e13

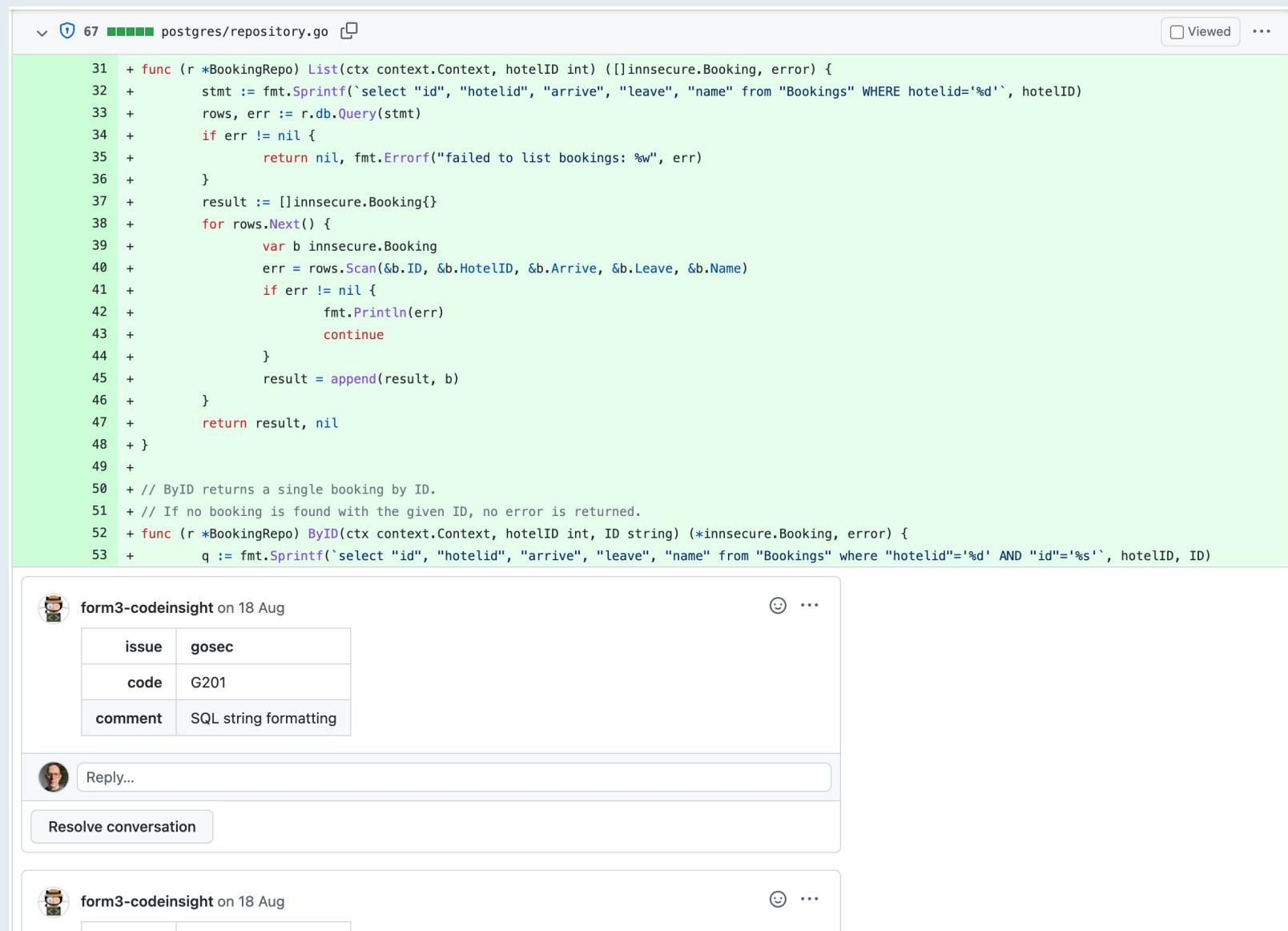
Duration: 16 hour(s)

✓ Passed   # 729a373c   >_ golang-snyk	<a href="#">View Logs</a>
✓ Passed   # 27e4ddd3   >_ shell-supplychain	<a href="#">View Logs</a>
✓ Passed   # 0505581a   >_ all-squealer	<a href="#">View Logs</a>
✓ Passed   # 19f512b2   >_ all-trivy	<a href="#">View Logs</a>
✓ Passed   # 444321e5   >_ dockerfile-hadolint	<a href="#">View Logs</a>
✓ Passed   # 49b9b30f   >_ golang-lint	<a href="#">View Logs</a>

Problems? Report a bug.

# Comments on PR

PR comments help our engineers easily identify and fix issues! 🚀



The screenshot shows a GitHub pull request interface. At the top, there's a navigation bar with a shield icon, the number '67', the repository name 'postgres/repository.go', and a 'Viewed' button.

The main area displays a diff of a Go file. Lines 31 through 53 are shown, with several additions (prefixed with '+') and one deletion (prefixed with '-'). The code implements a database query to list bookings by hotel ID.

Below the code, a comment from a user named 'form3-codeinsight' is visible, dated 'on 18 Aug'. The comment includes a table with three rows: 'issue', 'gosec', 'code', 'G201', and 'comment', 'SQL string formatting'. There are reply and resolve buttons below the comment.

At the bottom, another identical comment from the same user is shown, indicating a follow-up or a reply to the previous comment.

```

31 + func (r *BookingRepo) List(ctx context.Context, hotelID int) ([]innsecure.Booking, error) {
32 +     stmt := fmt.Sprintf(`select "id", "hotelid", "arrive", "leave", "name" from "Bookings" WHERE hotelid=%d`, hotelID)
33 +     rows, err := r.db.Query(stmt)
34 +     if err != nil {
35 +         return nil, fmt.Errorf("failed to list bookings: %w", err)
36 +     }
37 +     result := []innsecure.Booking{}
38 +     for rows.Next() {
39 +         var b innsecure.Booking
40 +         err = rows.Scan(&b.ID, &b.HotelID, &b.Arrive, &b.Leave, &b.Name)
41 +         if err != nil {
42 +             fmt.Println(err)
43 +             continue
44 +         }
45 +         result = append(result, b)
46 +     }
47 +     return result, nil
48 +
49+
50+ // ByID returns a single booking by ID.
51+ // If no booking is found with the given ID, no error is returned.
52+ func (r *BookingRepo) ByID(ctx context.Context, hotelID int, ID string) (*innsecure.Booking, error) {
53+     q := fmt.Sprintf(`select "id", "hotelid", "arrive", "leave", "name" from "Bookings" where "hotelid"=%d AND "id"=%s`, hotelID, ID)

```

## Code Insight architecture

# Benefits

Centralised config makes maintenance a lot easier

New scans are easy to write

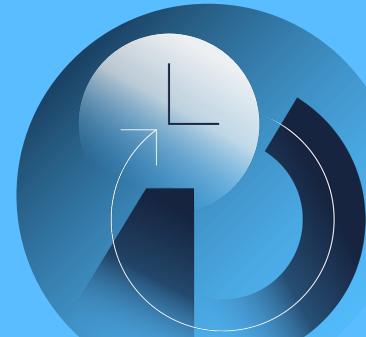
☀️ On-demand infrastructure, responds to diurnal pattern of use🌙

## Driving adoption

# Introducing Code Insight at Form3

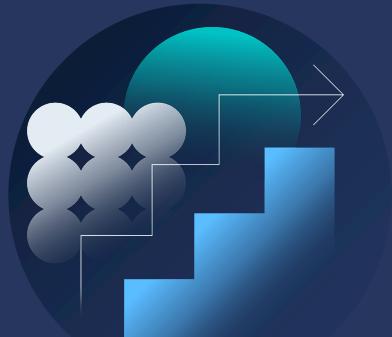
### Existing repositories

Soft Failures at first, with  
'ratcheting' approach, and  
support to raise coverage



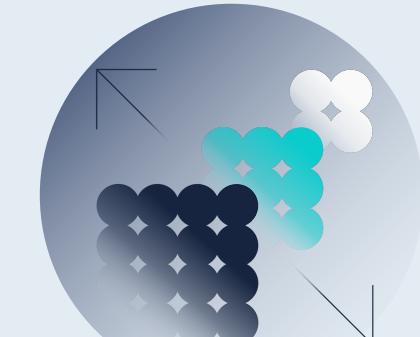
### New Repositories

Automatically enforced for all  
new repositories



### Metrics

We gather metrics from scan  
results to help us assess  
vulnerabilities and Code  
Insight's performance



## Driving Adoption

# Driving Adoption

Batch PRs

Mob sessions  
to improve coverage

Some gamification with  
Team Leaderboard

# Gamification

## Code Insight



---

### 🏆 Leaderboard

Global > Leaderboard

	Ranking	Team	% Pass	# Pass	# Fail
	1	product			
	2	international			
	3	infosec			
	4	fps-gateway			
	5	core-ui			
	6	platform			
	7	europe			
	8	tooling			
	9	core-payments			
	10	systems			

# Insights from Code Insight

## Insights from Code Insights

# Stuff that didn't go so well

A flaky scan can put the brakes on the engineering team

Large repositories are tricky to reform

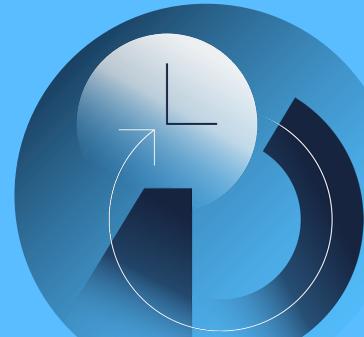
New vulnerabilities can break a build when you least expect it

## Insights from Code Insight

# Upcoming features/improvements

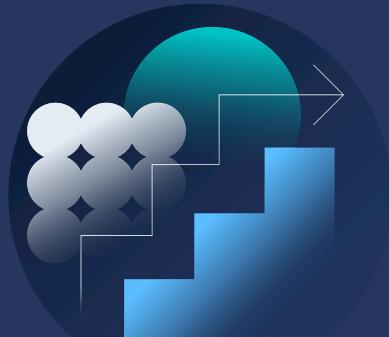
### Work in progress

We want to improve our detection to distinguish between old and newly introduced errors



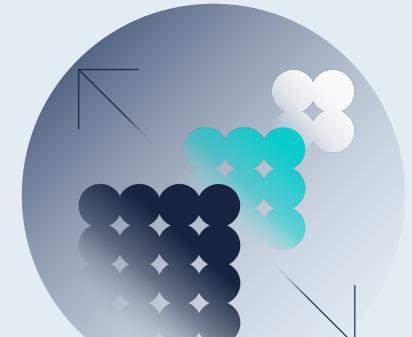
### Insight metrics monitoring

We have more work to do on the monitoring of Code Insight metrics



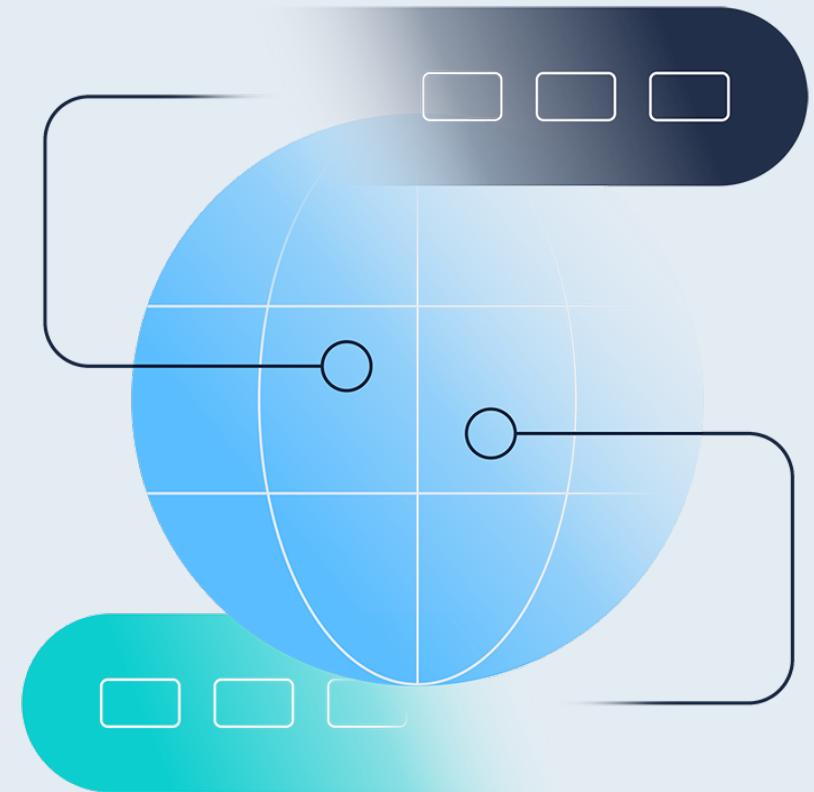
### Policy adjustments

We have found Code Insight to be too sensitive to failures at times. We should incorporate our policies of age & severity in the failures.

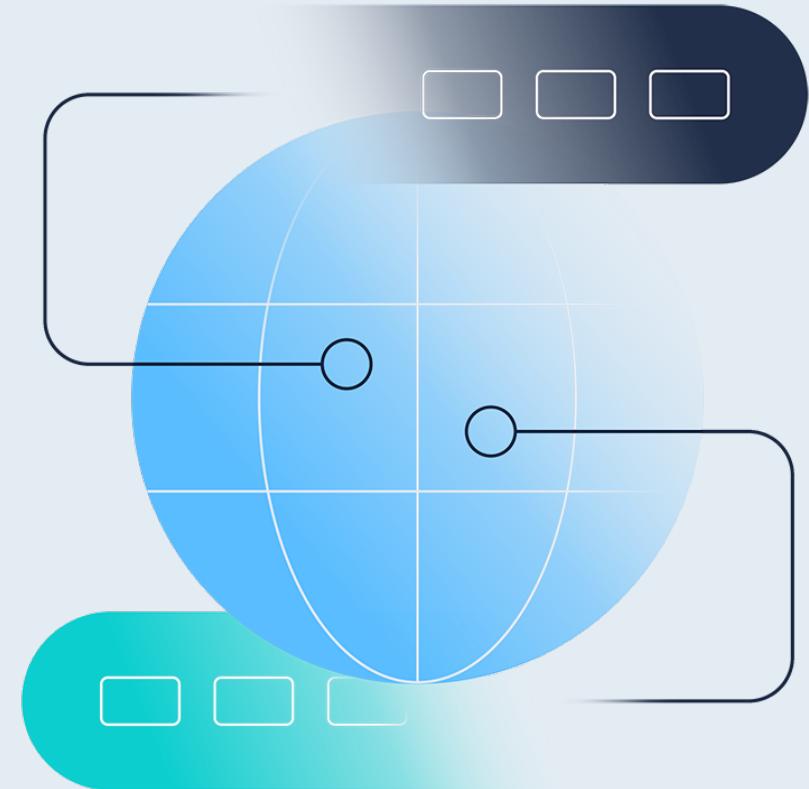


# Conclusions

Code Insight allowed Form3 to streamline development work.

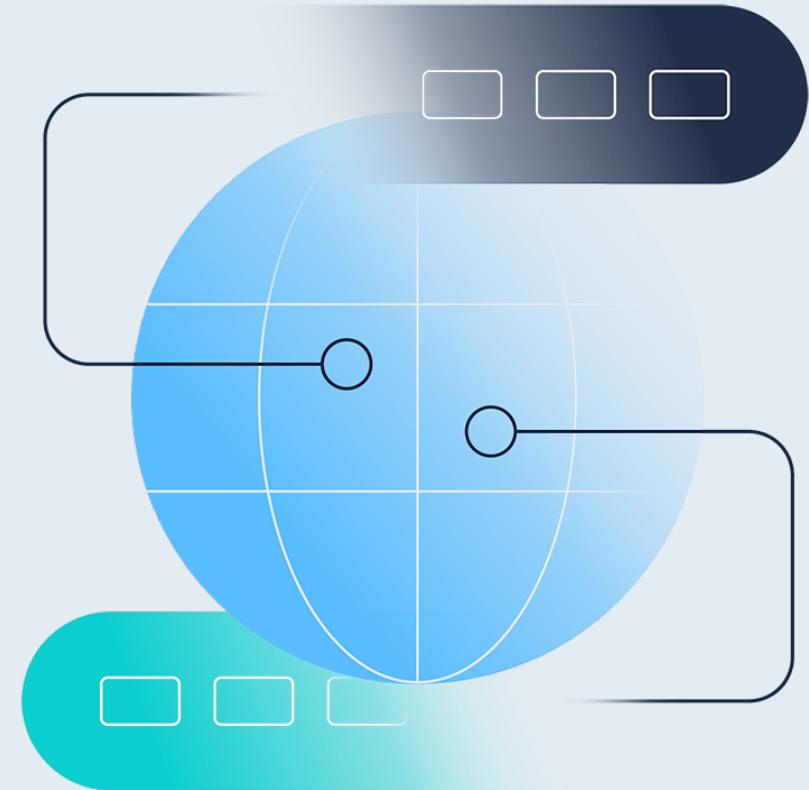


Nightly builds, alongside PR builds, ensure that we have an up-to-date view of our vulnerabilities.

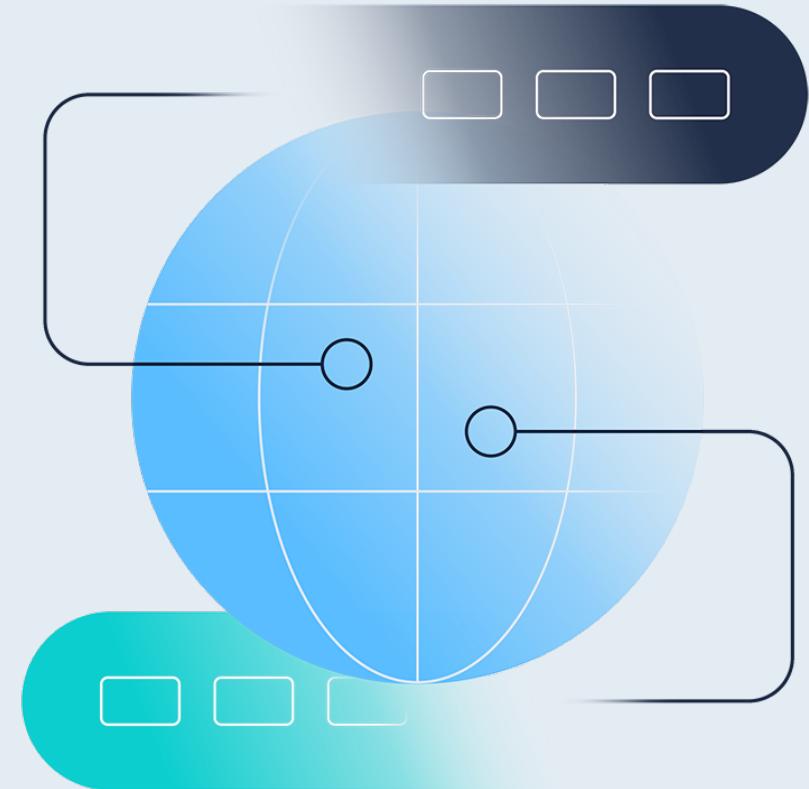


GitHub integration and PR comments were helpful for our engineers to fix issues.

Gamification was not meaningful to this project.



No extra config and easy maintenance were big improvements to our previous code scanning solution.



Thanks for listening! 🎤👊⬇️



Adelina Simion



addetz



classic\_addetz



Ross McFarlane



rossmcf



rossmcf

Check out our careers site,  
podcast and  
our engineering site!

[form3.tech/careers](https://form3.tech/careers)

[techpodcast.form3.tech](https://techpodcast.form3.tech)

[engineering.form3.tech](https://engineering.form3.tech)