



Reducing risk by testing every change
where it matters, in production



Agenda

- Why test in production
- Specific strategies we've used
- Deep Dive: Kinesis to Kafka Migration
- Deep Dive: S3 to Mongo Migration
- Starting/accelerating the journey

Why test in production? (pre-production testing is not enough)



Staging != Production

Breadth and depth

It's not set up the same way

- ▶ Smaller clusters
- ▶ Less services
- ▶ Different configurations

User behaviors

Its users behave differently

- ▶ Less users
- ▶ Less scenarios
- ▶ Less complex interactions

Monitoring

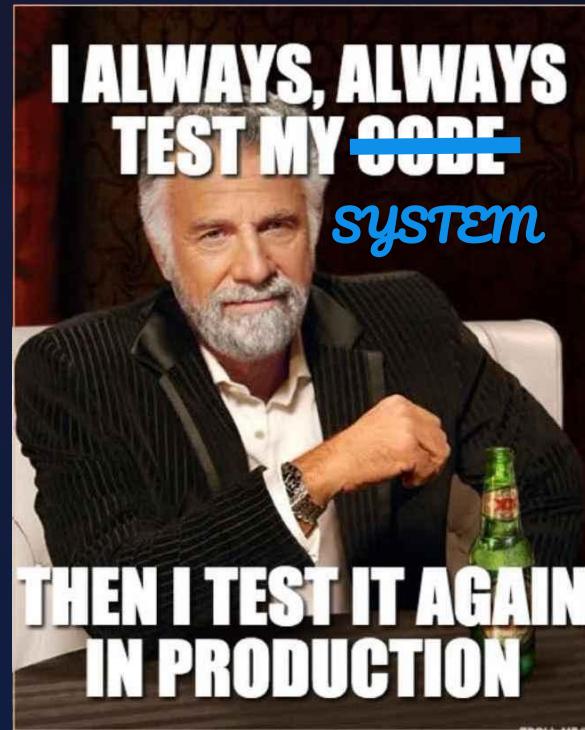
Signals differ

- ▶ Thresholds are different
- ▶ Certain signals never emitted

“None of this is meant to suggest that maintaining a staging environment is *completely* useless . . . Only that, as often as not, it’s relied upon to a much greater degree than necessarily needed, to the point where at many organizations it remains the *only* form of testing that happens before a full production rollout.”

– Cindy Sridharan, [Testing in Production, the safe way](#)

What is Testing in Production?



Three distinct stages of “Production”



Deployment

- ▶ Integration Tests
- ▶ Tap Compare
- ▶ Load Tests
- ▶ Shadowing
- ▶ Configuration Tests
- ▶ Dark Launches

Release

- ▶ Canarying
- ▶ Monitoring
- ▶ Traffic Shaping
- ▶ Feature-Flagging
- ▶ Exception Tracking

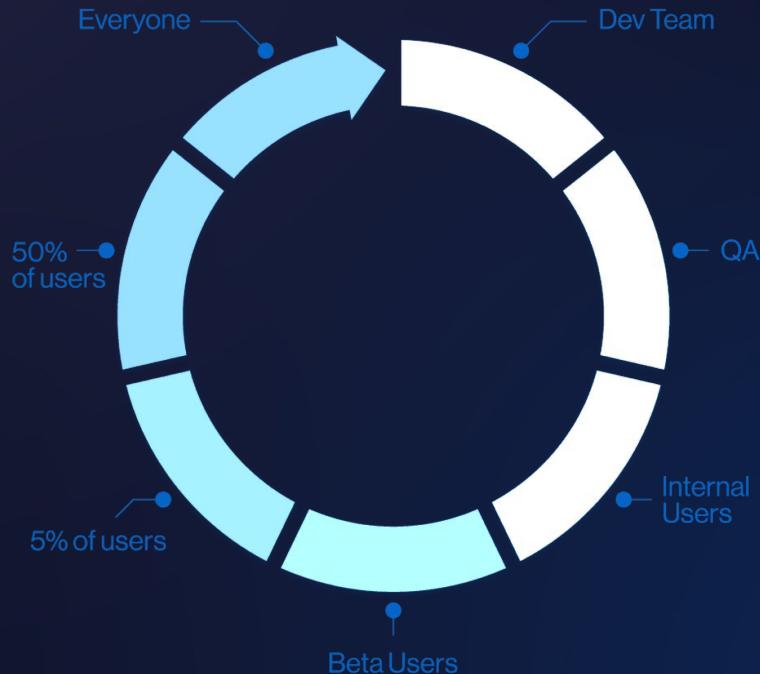
Post-Release

- ▶ Teeing
- ▶ Profiling
- ▶ Logs/Events
- ▶ Chaos Testing
- ▶ Monitoring
- ▶ A/B/n Testing
- ▶ Tracing
- ▶ Dynamic Exploration
- ▶ Real-User Monitoring
- ▶ Synthetic User Testing
- ▶ Auditing

Specific strategies we've used

Progressive Delivery

Rolling out to an increasingly larger impact radius



Dark Launches

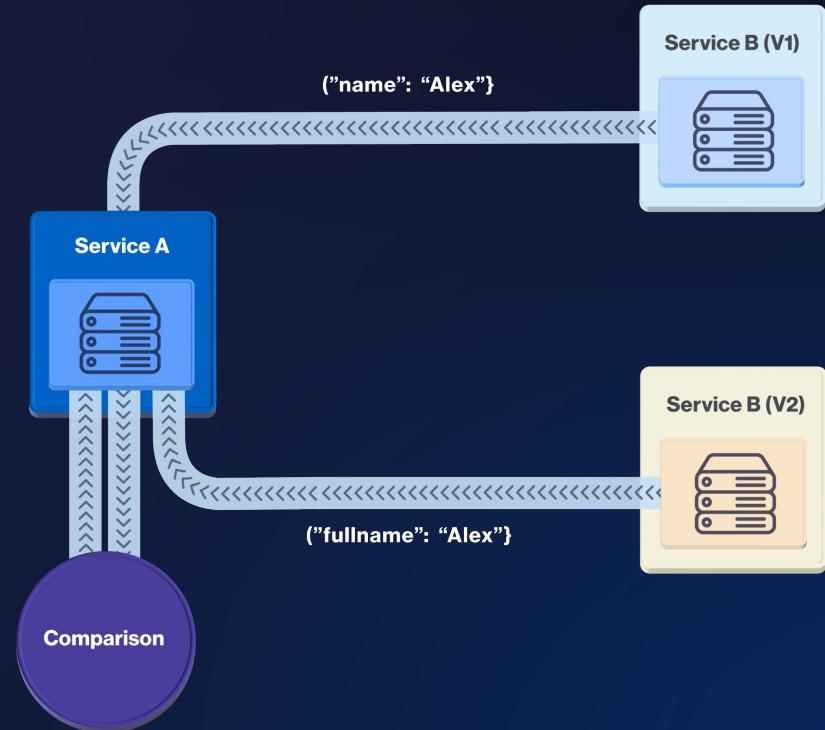
Deploying new features without enabling them for customers



Tap Compare Testing

“Tap compare” is a testing technique that allows you to test the behavior and performance of the new service by comparing its results against the old service.

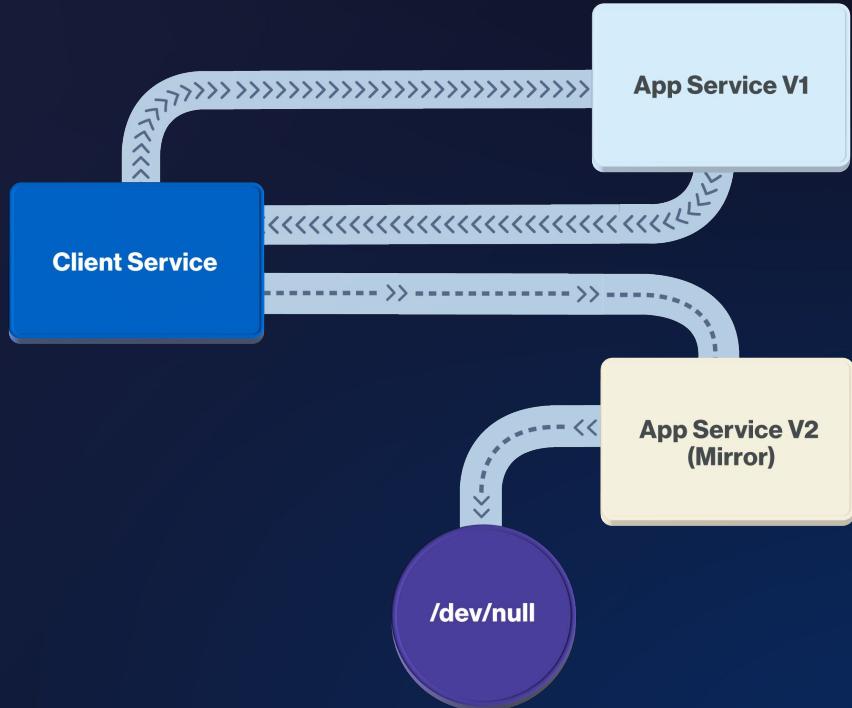
- [Tap Compare Testing with Diferencia and Java Microservices, InfoQ](#)



Traffic Mirroring

“Traffic mirroring” is when you deploy a new dark version of an application and copy traffic to it that the current live version is receiving. The traffic is sent in a “fire and forget” fashion, with the responses from the mirrored instance being ignored.

You can then observe and monitor the dark version for errors without impacting the live version.



Canary Releases

Rolling features out to a subset of users to assess the reaction of the overall system



Configuration Testing

Testing with multiple combinations of software and hardware in order to evaluate the functional requirements and find out optimal configurations

Format

Treatment

on | video_carousel

videos_displayed	=	5	<input type="button" value="–"/>
tile_size large	=	large	<input type="button" value="–"/>

Format

Treatment

on

```
1 {  
2   "box_color": "red",  
3   "box_border_color": "red",  
4   "box_border_width": "thin",  
5   "text_shadow": "1px 1px 0 black",  
6   "font_size": "medium"  
7 }
```

Alpha Testing

Leverage your internal testers!



Serve

To Users

To Segments "/>
Start typing a segment

Beta Program

Allow customers to opt-in



How to get the app before the crowd.

You can get **EARLY** and **FREE** access to our products by joining the beta programme. Get to know the team whilst your opinion and feedback help to shape the app for others... here will be perks along the way. Request an below.

I want in to the beta programme!

Name

Preferred Platform ▾

Enter your email address

Request Invite

```
curl --request PUT \
--url
'https://api.split.io/internal/api/v2/segments/environment_id
/<segment_name>/upload?replace=false' \
--header 'accept: application/json' \
--header 'content-type:
application/x-www-form-urlencoded' \
--data file=file
```



Deep-dives:

The role of feature flags and data for testing in production





Deep Dive: Kinesis to Kafka Migration

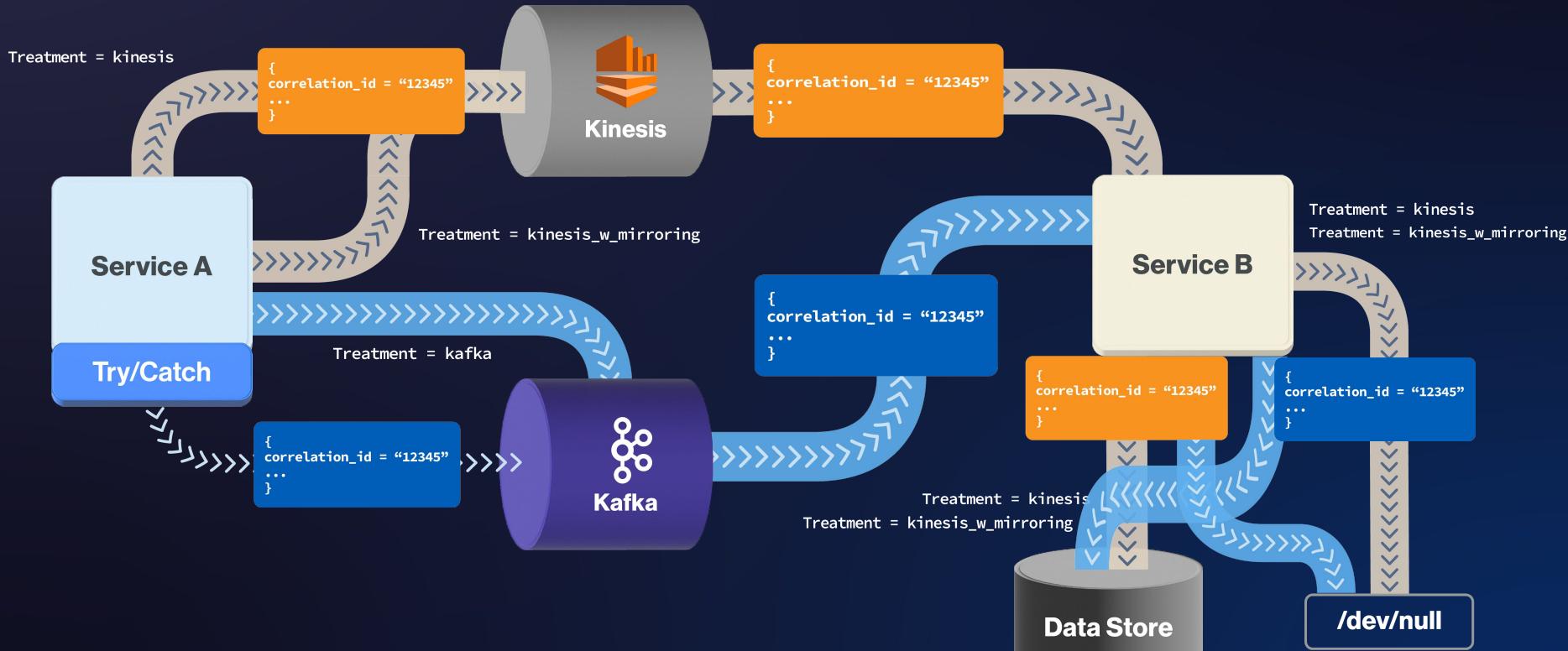
Goals & Strategy

- ▶ Why: Reduce overall costs for our data pipelines
- ▶ Constraints:
 - ▷ Zero lost events
 - ▷ No increase in latency
 - ▷ No decrease in throughput

Strategy: Traffic Mirroring

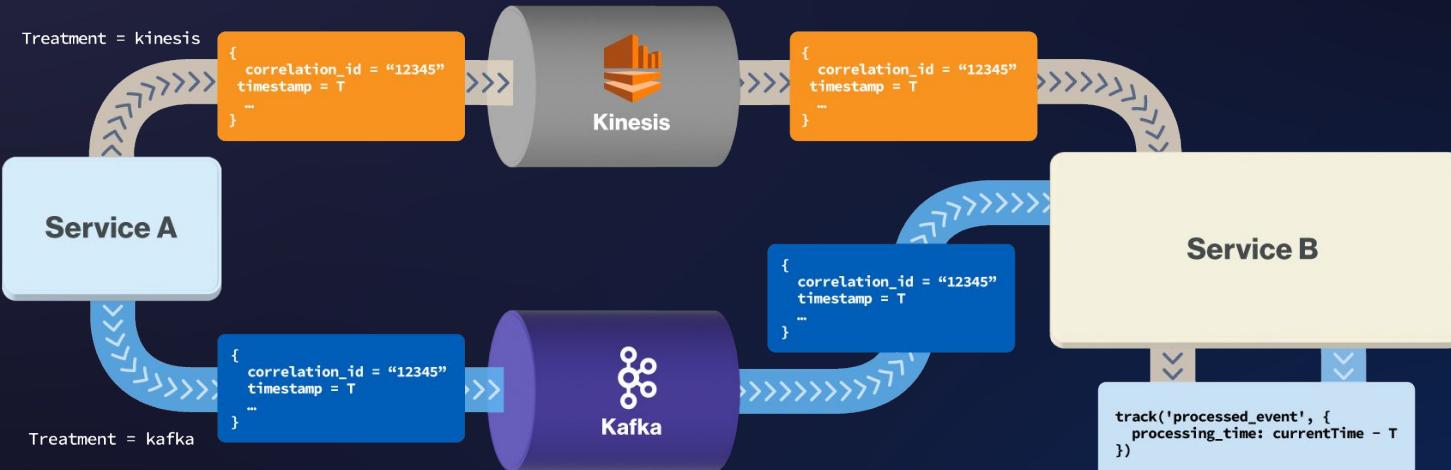
Kinesis to Kafka Migration

Traffic mirroring with Feature Flags



Kinesis to Kafka Migration

The power of Feature Flags + Data



Alert fired	Name	Relative impact	Absolute impact
08/24/2022	event_latency_ms	5.10% (0.95%)	2.51 (± 0.48)
08/24/2022	events_per_sec	13.44% (2.89%)	0.35 (± 0.08)





Deep Dive: S3 to Mongo Migration

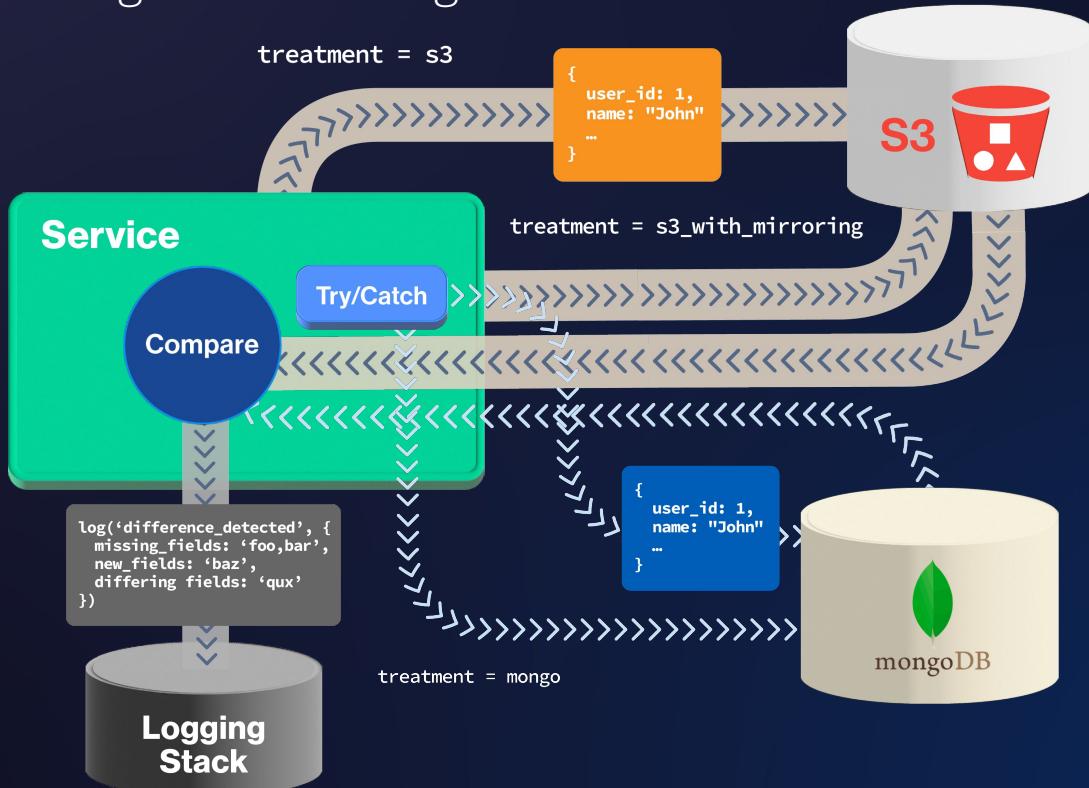
Goals & Strategy

- ▶ Why: Support more complex queries
- ▶ Constraints:
 - ▷ Zero data loss
 - ▷ Reduce latency
 - ▷ Increase write throughput

Strategy: Tap Compare via Traffic Mirroring

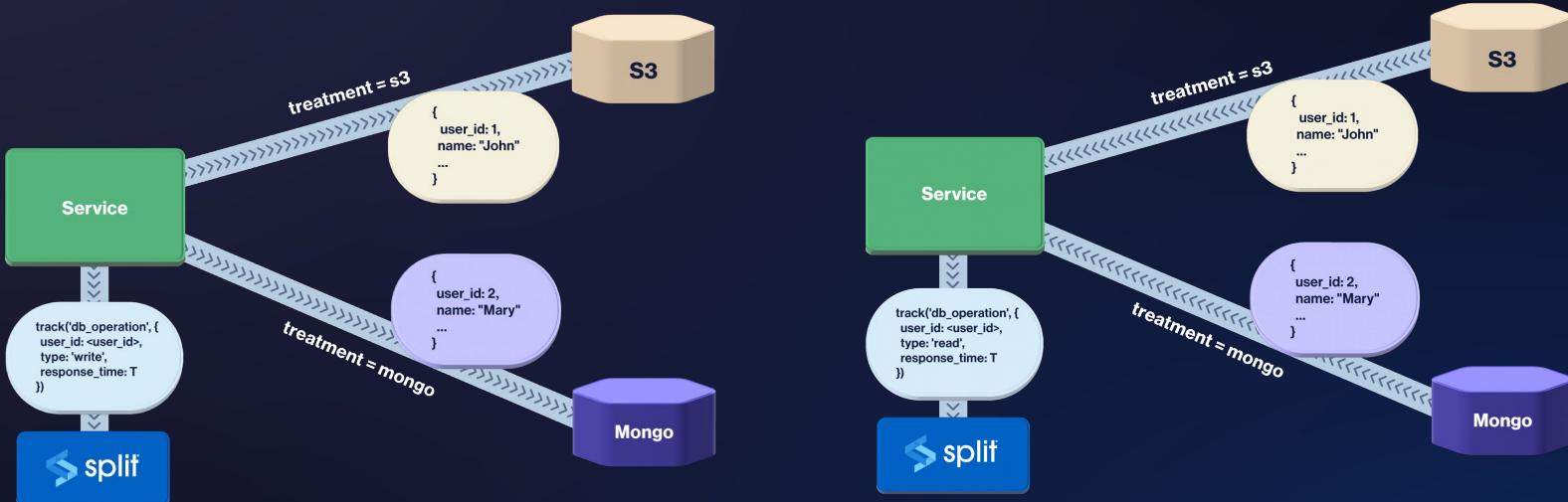
S3 to Mongo Migration

Tap compare testing with mirroring



S3 to Mongo Migration

Measuring impact on latency and throughput



DB Operations per minute

DESIRED

↑ 20.27%

Impact lies between 15.86% and 24.32%

Read latency

UNDESIRED

↑ 19.74%

Impact lies between 16.36% and -23.12%

Write latency

DESIRED

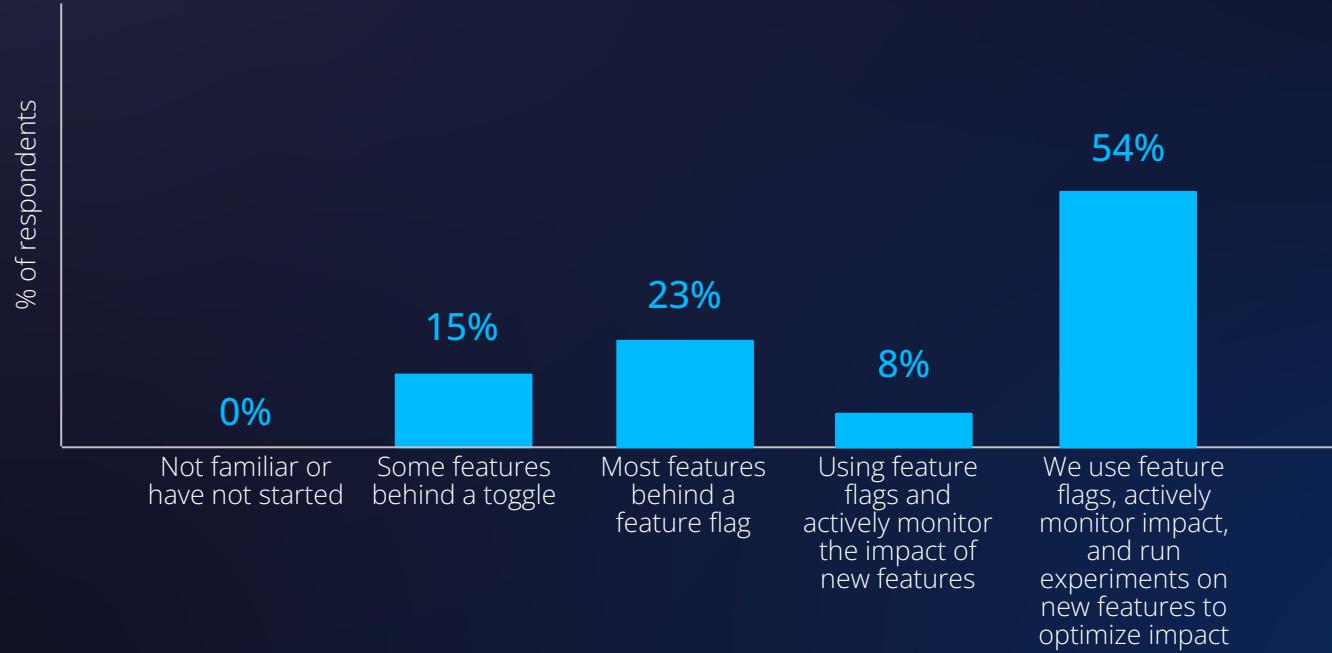
↓ 95.19%

Impact lies between -100.00% and -24.19%

Starting on/accelerating the journey

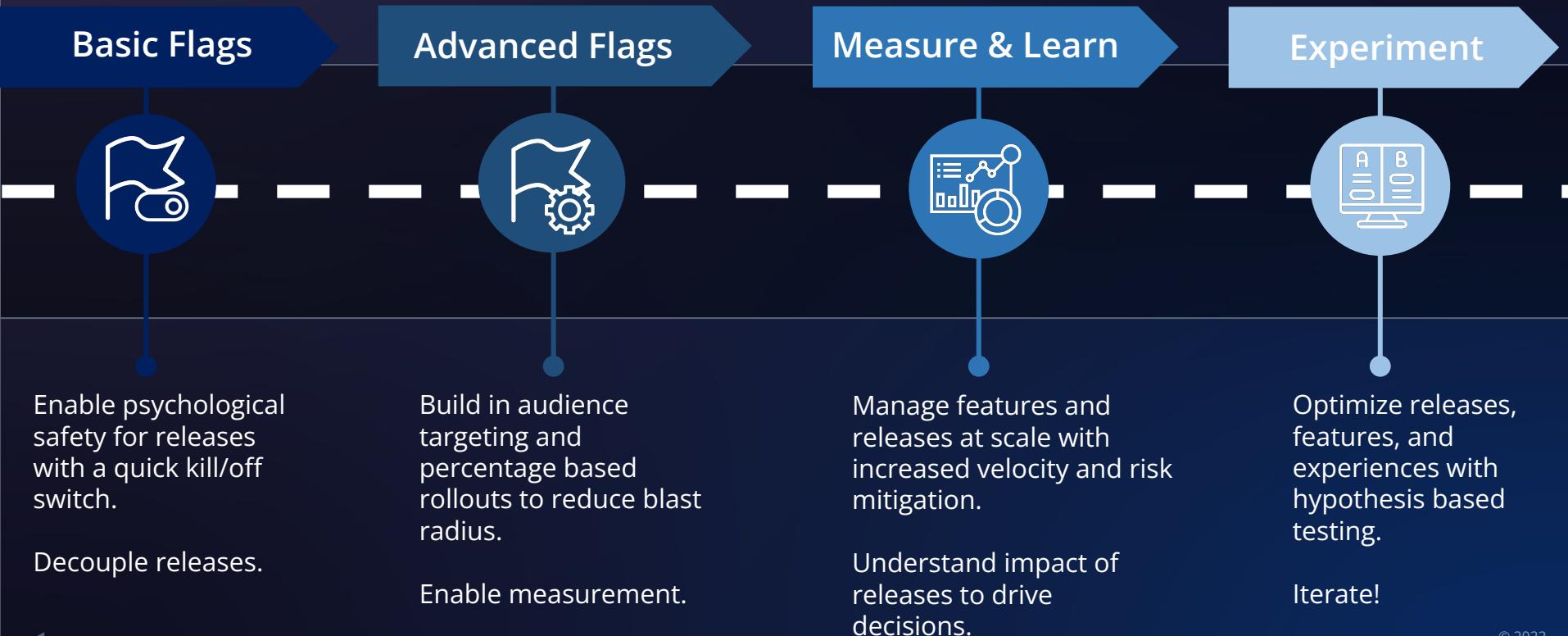
Each of us are at a different stage

Important to understand and explore best practices and use cases to move between stages



We're all on the feature flagging journey

Making informed decisions every step of the way



Moving towards an experimentation mindset

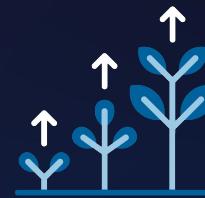
Embracing the challenge



Be curious and ask questions



Understand impact of releases



Start small and keep going



Stay humble and embrace mistakes



Make data-led decisions



Collaborate and get feedback

Focus on the journey itself, not the technology

Success happens through people, aligned and empowered to get things done



Customer Success Managers

Ensure your business objectives are fulfilled with a proactive, main point of contact.



Integration Advisors

Our experts guide you on how to implement Split into your DevOps practice.



Experimentation Advisors

Get one-on-one guidance on prioritizing experiments and making data-driven decisions.



Persona-based **maturity assessments**



Self-guided **learning pathways**



Integrated **training & tutorials**



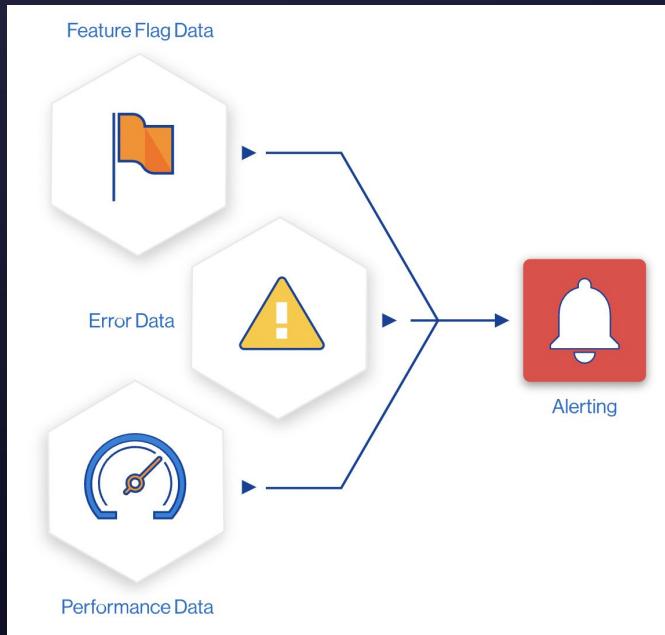
Step-level **certifications**



LinkedIn **badges**

Monitoring + Alerting with Split

Split continuously compares your feature flags and performance data to find anomalies caused by new features

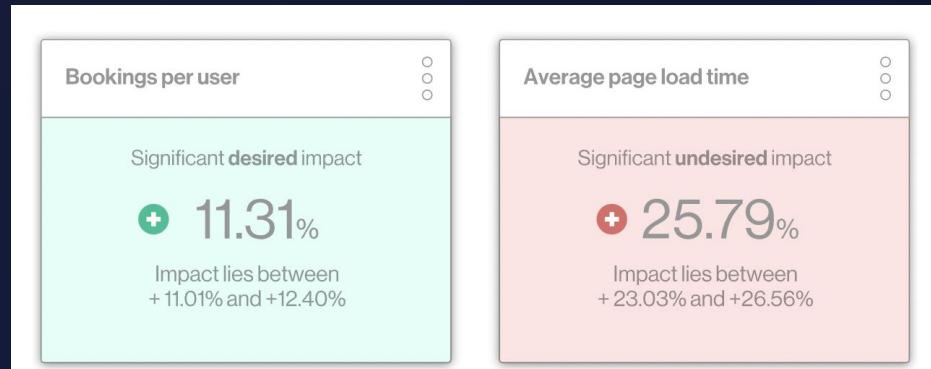
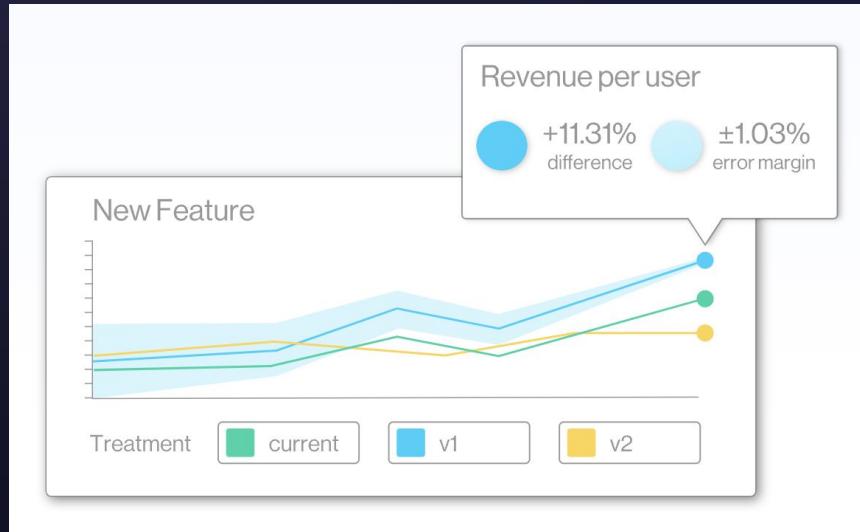


The screenshot shows the Split software's alerting interface. At the top, a red banner displays the message "ALERT! This rollout has caused 1 alert in the last 5 minutes." with a "Kill" button on the right. Below this, a section titled "Define Treatments" shows a "Treatment" (1-step) and a "Description" (register on same page). A table below lists three alerts that have fired on 08/24/2020:

Alert fired	Name	Relative impact	Absolute impact
08/24/2020	errors_per_user	5.10% (0.95%)	2.51 (± 0.48)
08/24/2020	page_load_time	13.44% (2.89%)	0.35 (± 0.08)
08/24/2020	avg_cart_size	-8.31% (4.22%)	-1.55 (± 0.80)

Experimentation

Measuring the impact of every change



Experimentation

A

Account Average Time to ODM Interactive Averag...

Desired ↑1,000.58%-1,800.58%

Account Average Time to ODM Interactive long...

Inconclusive ⚡

Account Average Time to ODM Interactive long...

Undesired ↑1,000.58%-1,800.58%

B

Accounts who view key metrics cards on click

DESIRED

↓ 95.19 %

Impact lies between -100.00% and -24.19%

Account Average Time to ODM Interactive

INCONCLUSIVE

0.27%

Impact lies between -0.86% and 0.32%

Account Average Time to ODM Interactive

UNDESIRABLE

↓ 19.74%

Impact lies between 16.36% and 23.12%

The importance of guardrails



Business Metrics

Depends on your business

- ▶ DAU/WAU/MAU
- ▶ ARPU
- ▶ AOV/Avg. Cart Size
- ▶ 7/14/Monthly Retention

Operational Metrics

Good base to start from: RED

- ▶ Rate
- ▶ Errors
- ▶ Duration



Thank You
ariel.perez@split.io