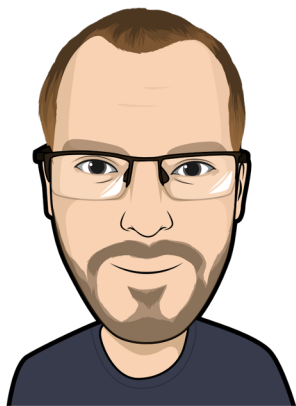




Machine Learning and Artificial Intelligence for Modern Software Delivery

Nik Papageorgiou

Solutions Engineer, Harness



- ML / Neuroscience focused studies
- Started career in software development
- Spent time in helping companies enable their digital transformation
- Focused on modernising the software delivery process under DevOps context

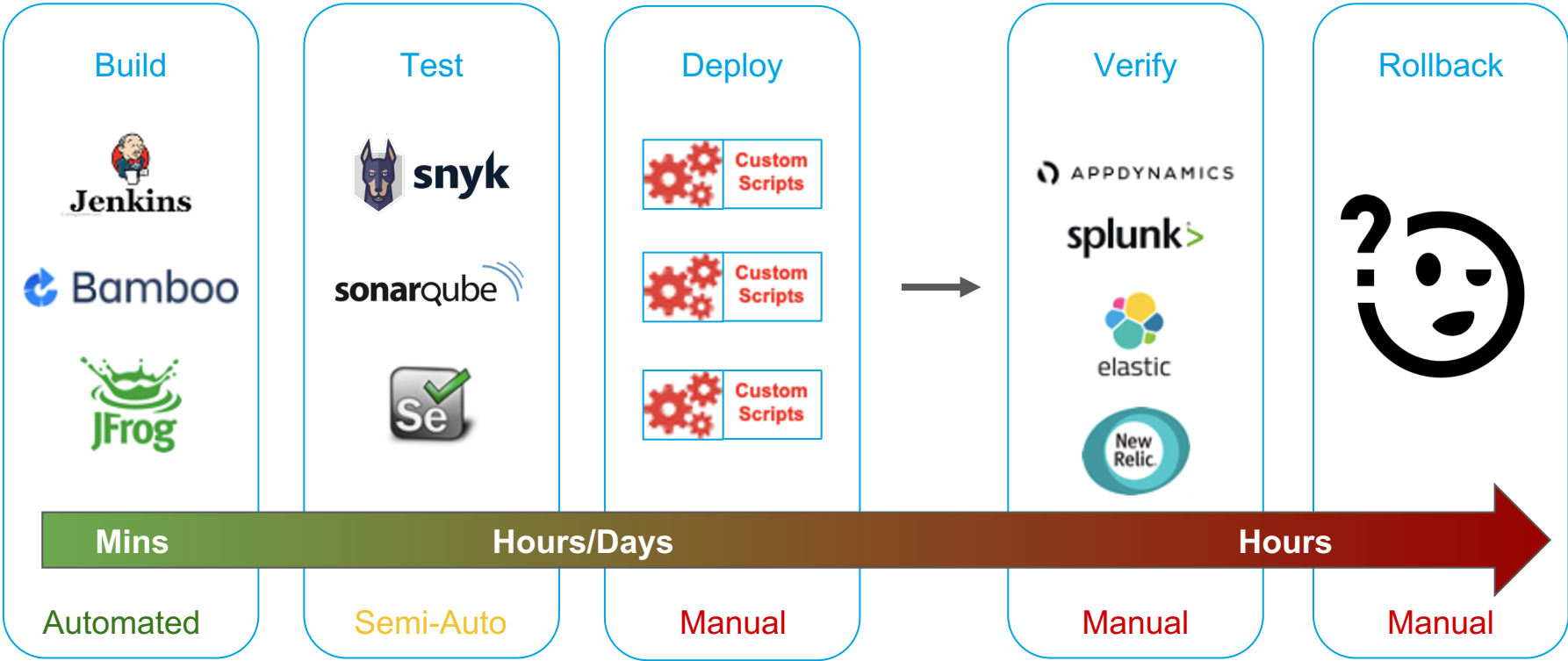
payment - service



reCAPTCHA



Software Deployment Pipeline



Areas to improve using AI & ML

Build

Waiting for build tests to run

When your code compiles after 253 failed attempts



Extra coffee breaks



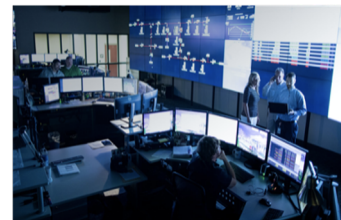
Deploy



Free Pizza

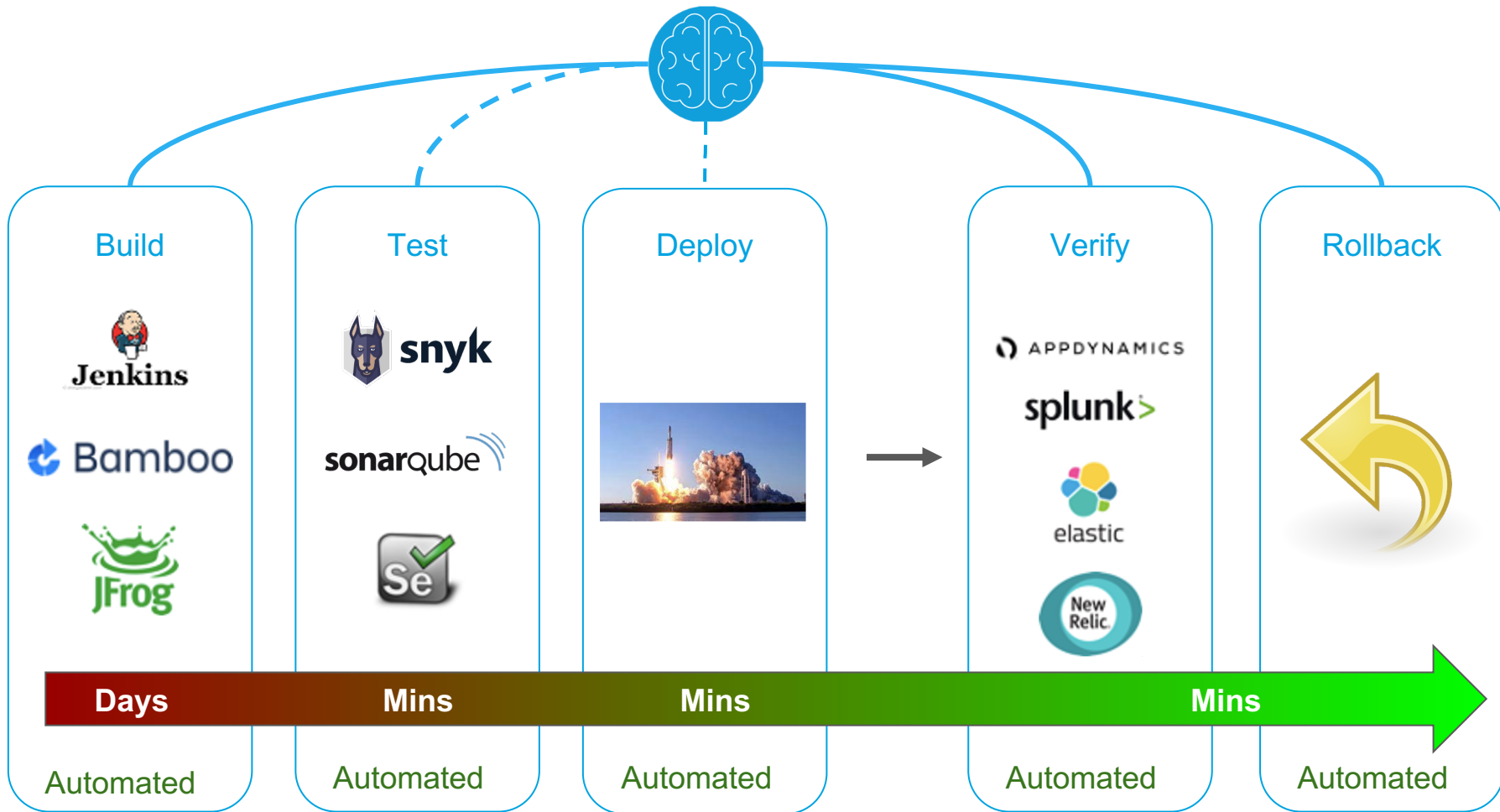


Feature Flags / DB toggles



More Free Pizza





Machine Learning 101

Supervised Learning

- Classification
- Regression

Where used?

- Statistical analysis where datasets exist (House Prices. Mortgage predictions)

VS

Unsupervised Learning

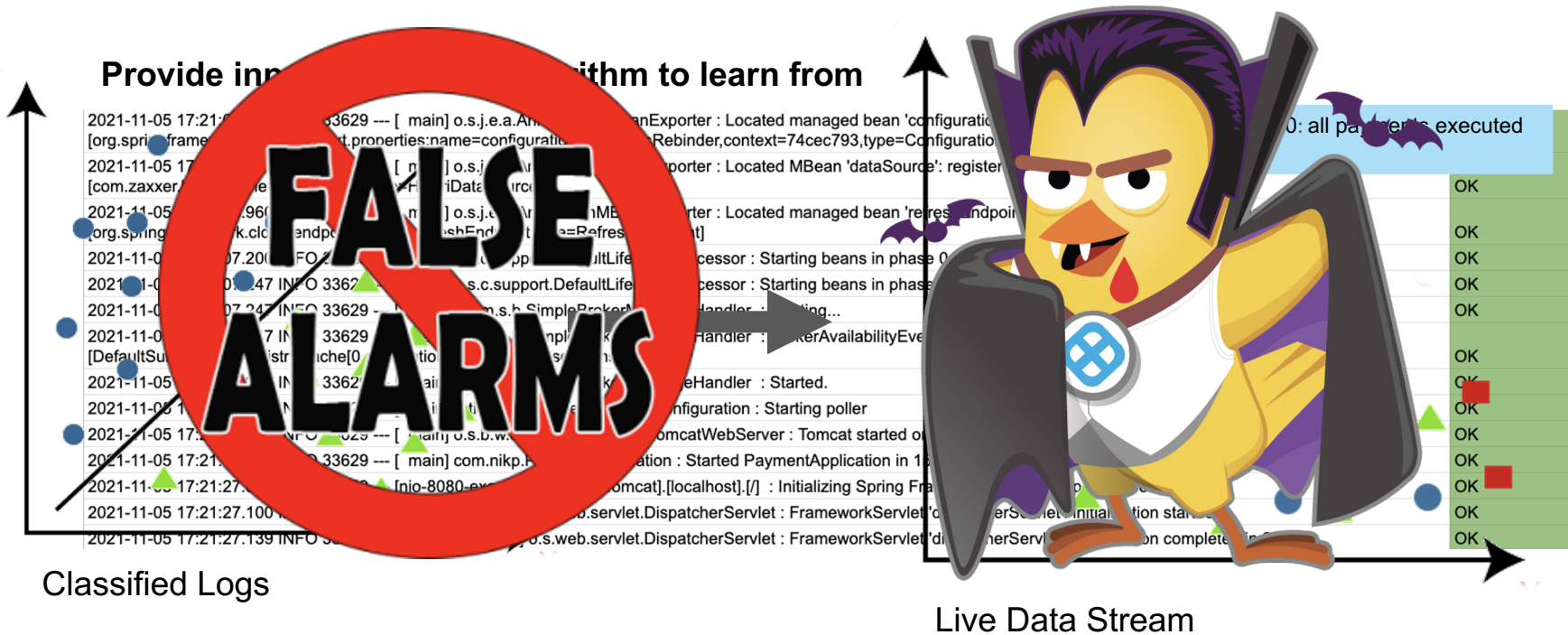
- Clustering
- Dimensionality reduction

Where used?

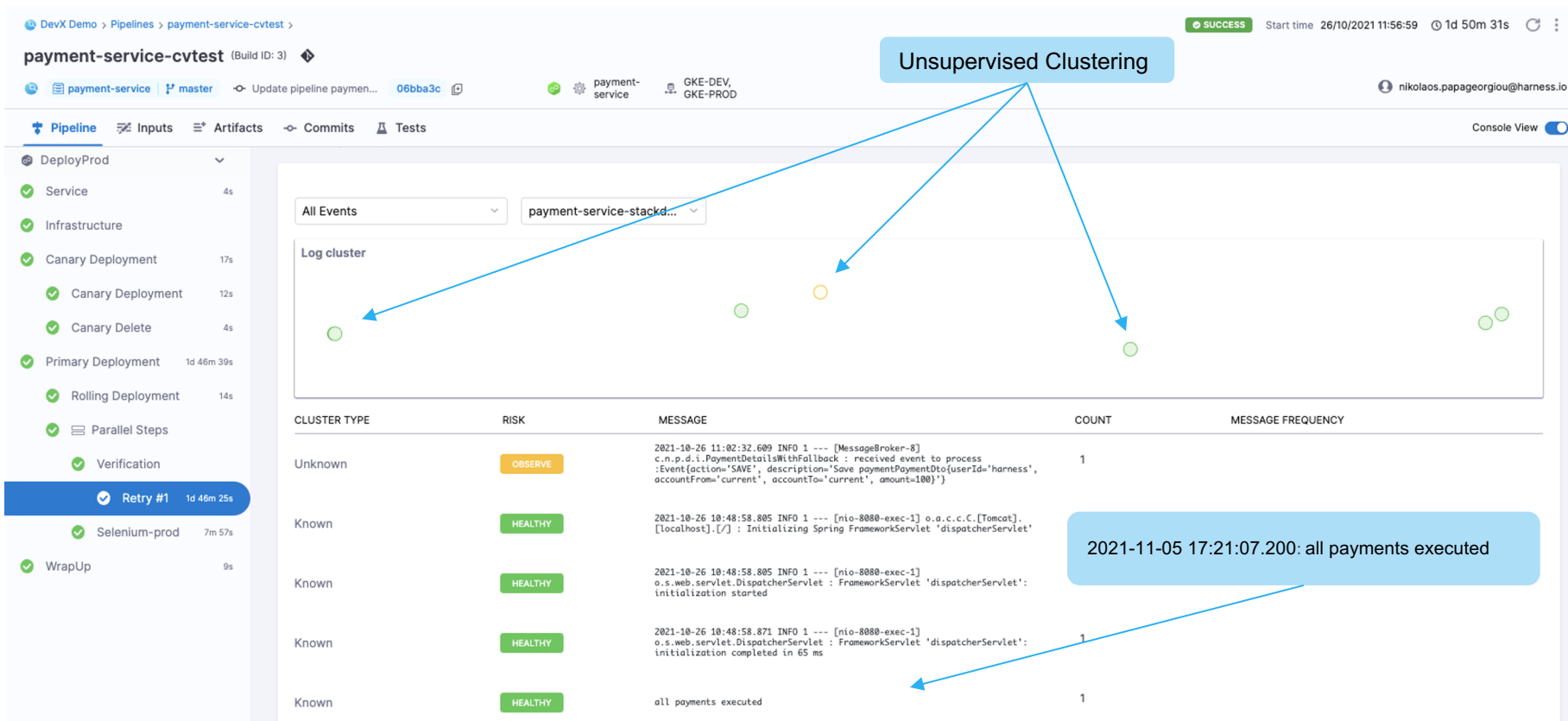
- Natural Language Processing, Computer Vision, Autonomous Technologies



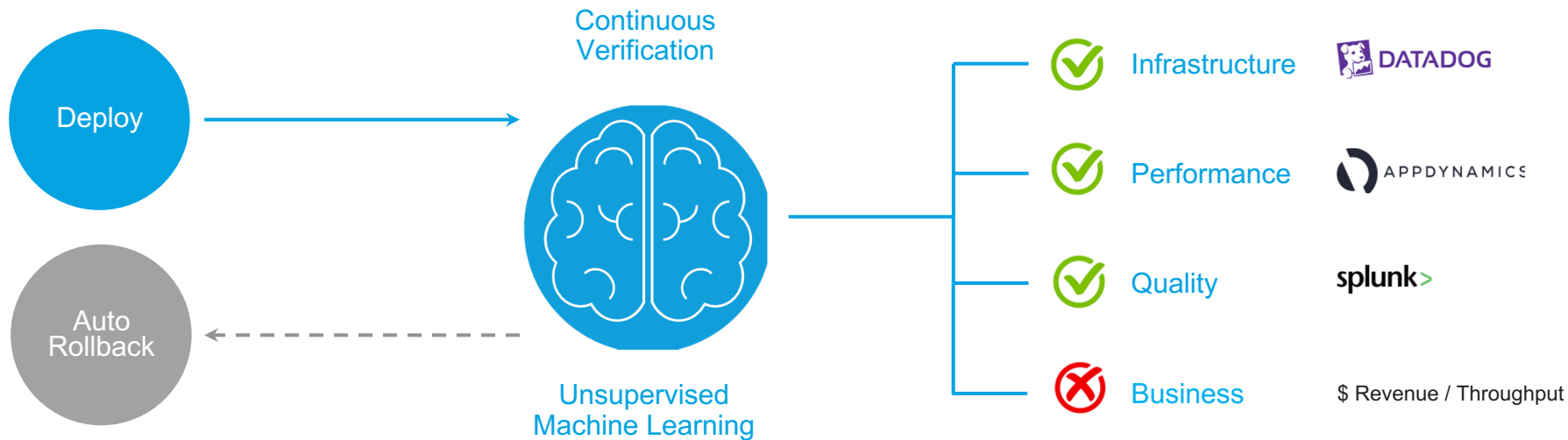
Supervised Learning Algorithms for Logs



Unsupervised Learning Algorithms for Logs



Harness - Applying Machine Learning to CD



Project

DevX Demo >>

Overview

Builds

Pipelines

PROJECT SETUP

Feature Flags

Cloud Costs

Change Intelligence

CONTINUOUS Integration

DevX Demo > Pipelines > payment-service-cvtest >

payment-service-cvtest (Build ID: 10)

payment-service | master Update pipeline payment... 42783a5

payment-service GKE-DEV, GKE-PROD

nikolaos.papageorgiou@harness.io

Pipeline

Inputs

Artifacts

Commits

Tests

Console View

Build

DeployDev

DeployProd

DeployProd

Started at: 02/11/2021, 10:33:43

Duration: 44m 21s

Service(s)
payment-service

Environment(s)
GKE-PROD

FAILED Error Summary

Canary Deployment

Canary Delete

Rolling Deployment

Verification

rollback

WrapUp

Rollback

Selenium-prod

Step: rollback

Details

Input

Output

Started at 02/11/2021, 11:17:51

Ended at 02/11/2021, 11:17:56

Duration 5s

Timeout 10m

Delegate devx-demo-gke for K8s Task: Rolling Deployment Rollback (Delegate selection logs)

Step Logs

Initialize

1 INFO 02/11/2021 11:17:52 Initializing..

2 INFO 02/11/2021 11:17:52 Release Name: [release-1]

3 INFO 02/11/2021 11:17:53 Found following Managed Workloads:

Kind Name

Versioned

Deployment payment-service-nikpapag-d

deployment false

4 INFO 02/11/2021 11:17:53 Done.

Rollback

1 INFO 02/11/2021 11:17:53 Previous eligible Release is 206 with status Succeeded

2 INFO 02/11/2021 11:17:53 Rolling back to release 206



"It literally took me **1 hour**
to onboard my first
service"

Migrating to Microservices

Kubernetes & AWS ECS



"We achieved **more in
one day** than we did in
two months with
Spinnaker"

Migrating to Microservices

Kubernetes & AWS



Deployed first pipeline
In **4 hours**

Modernize & Scale CI/CD

Kubernetes & AWS



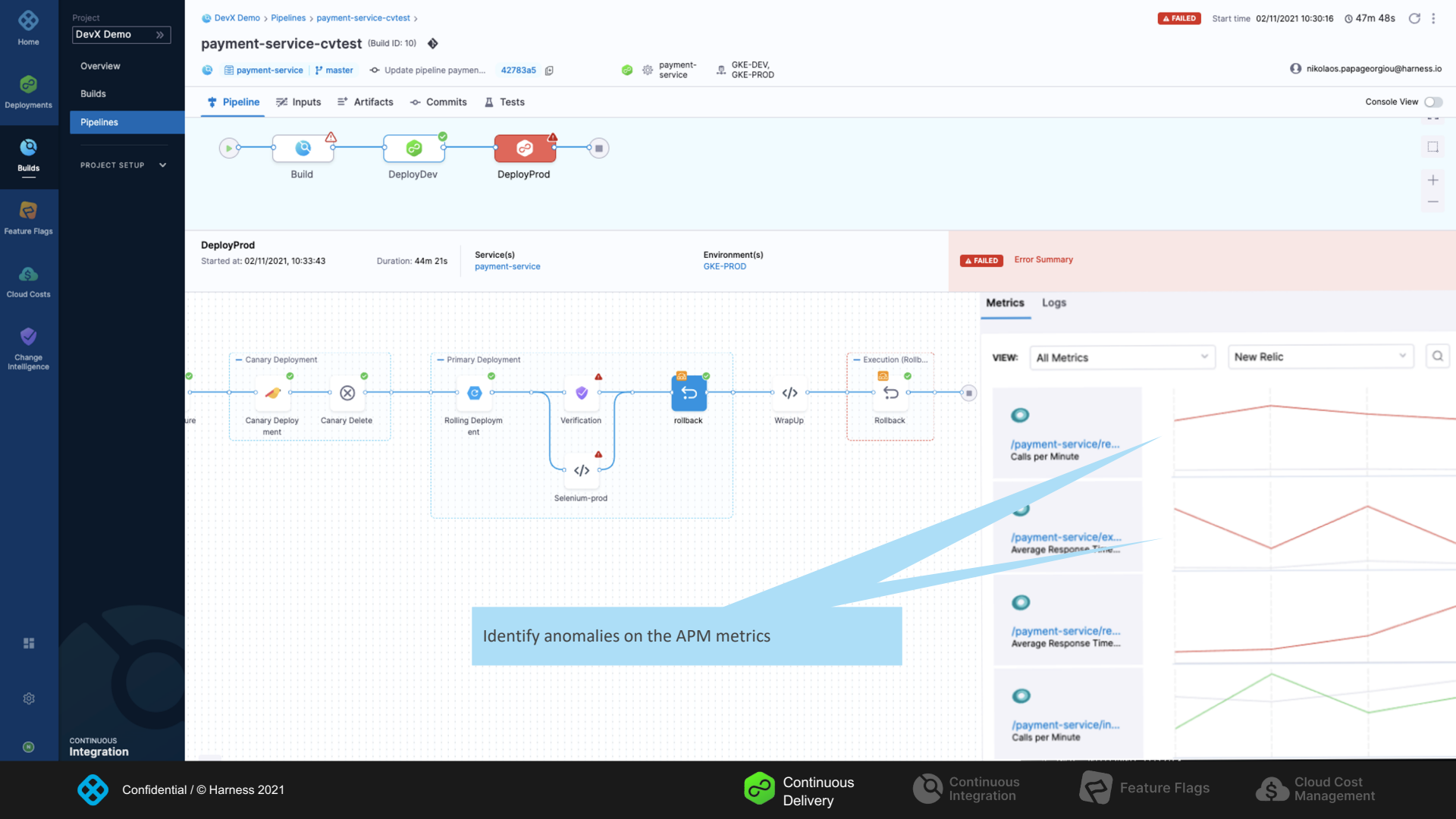
"Deployed in production
within a week"

Self-Service
Continuous Delivery

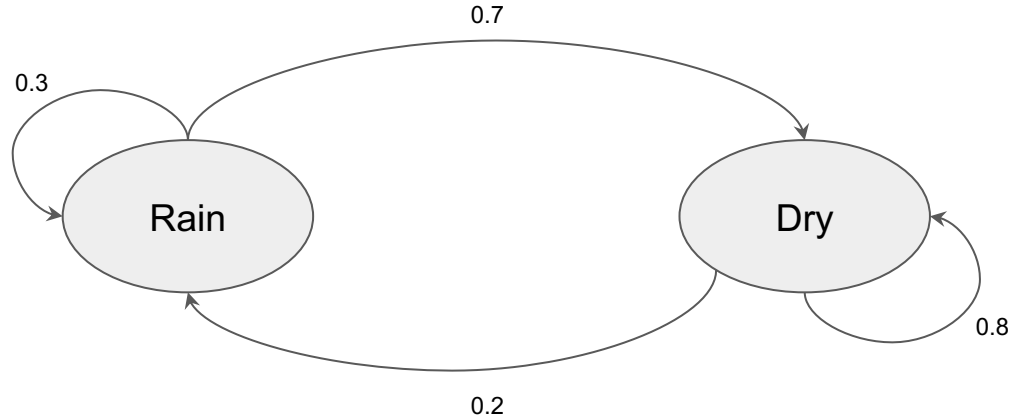
AWS EC2, ECS & Lambda

Application Performance Metrics (Anomalies)





Markov Model

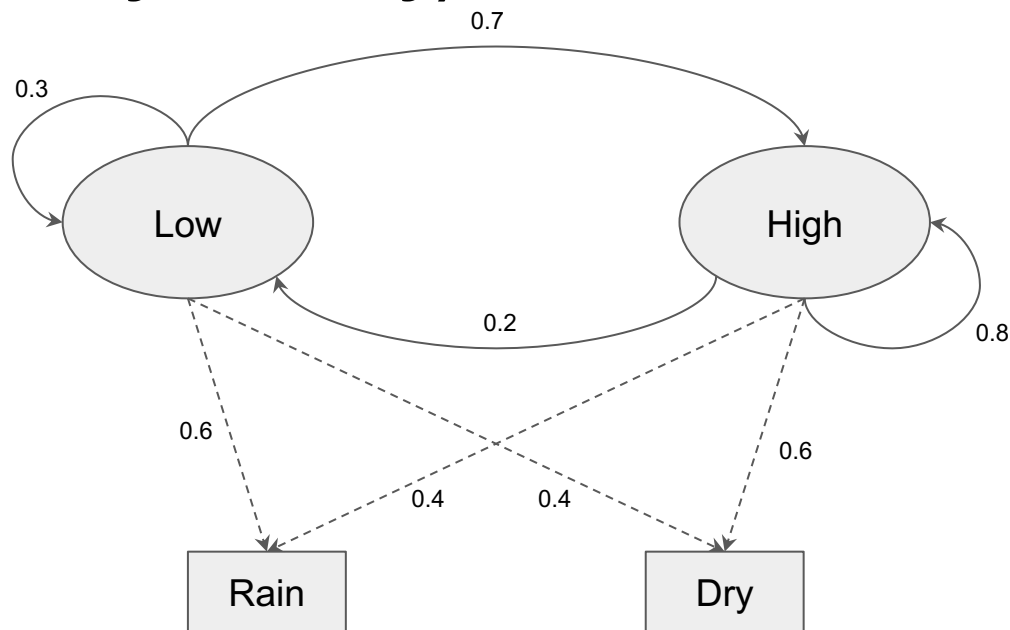


Predict the weather ? Easy !!! Use Markov's Model to predict the weather based on previous state

Very low accuracy Why ?



Hidden Markov Model (Weather is not a result of what weather we had yesterday)

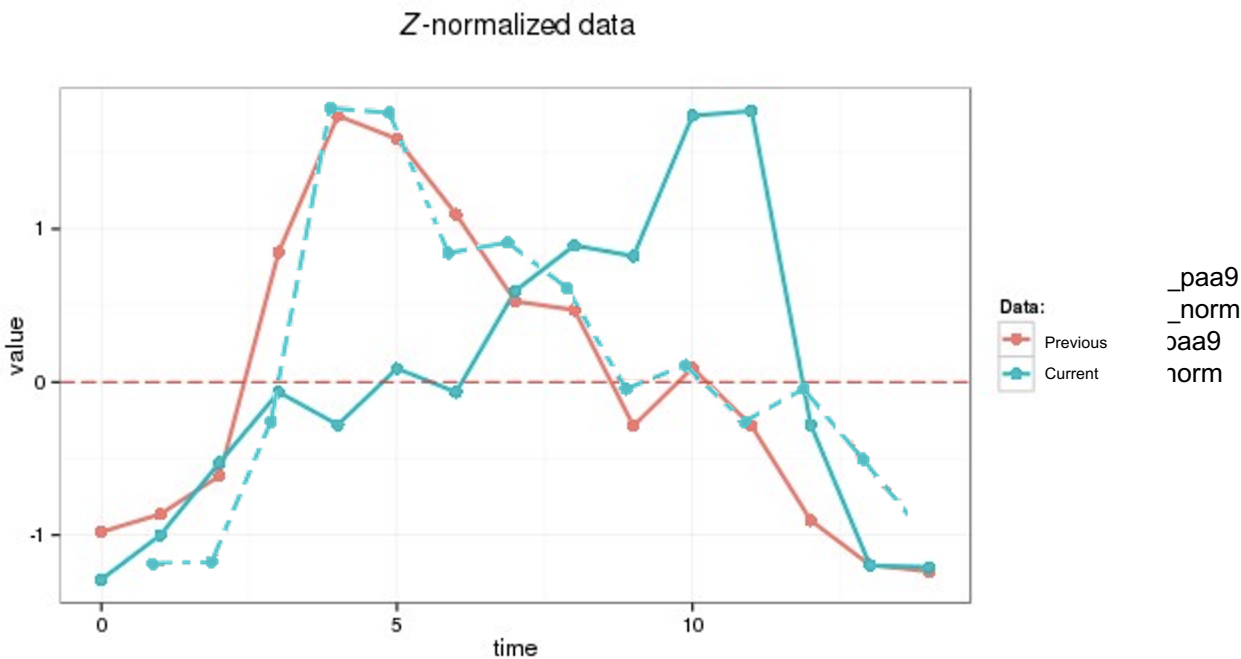


Atmospheric Pressure can be our additional telemetry



Symbolic aggregate expression (SAX)

Previous Piecewise Aggregate Approximation Time Series Series



https://jmotif.github.io/sax-vsm_site/morea/algorithm/SAX.html



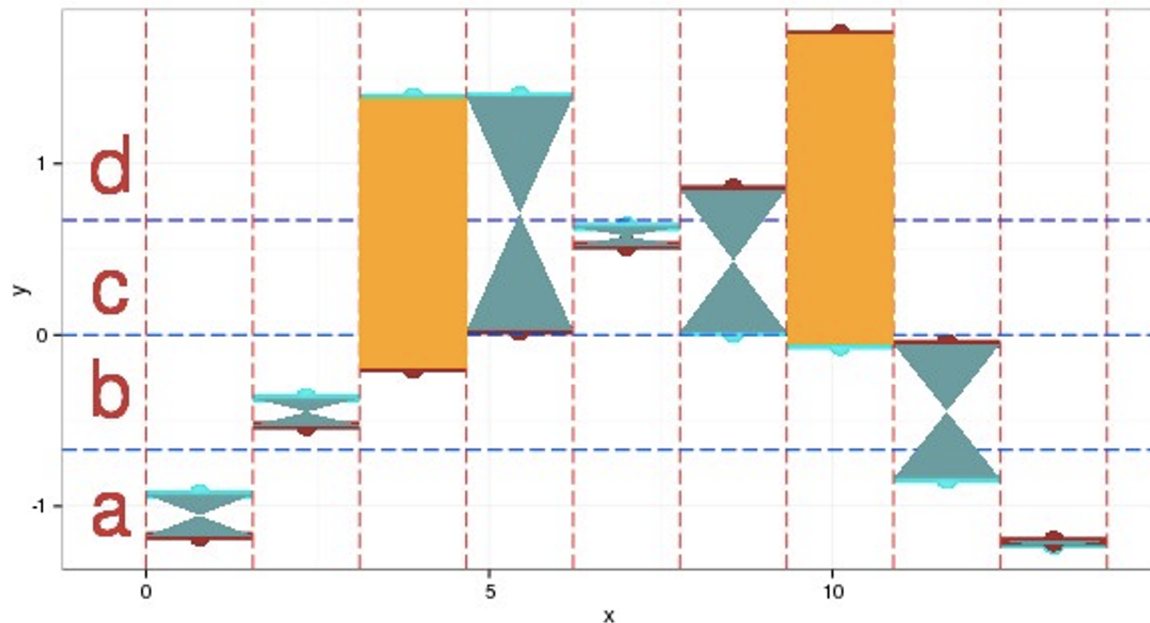
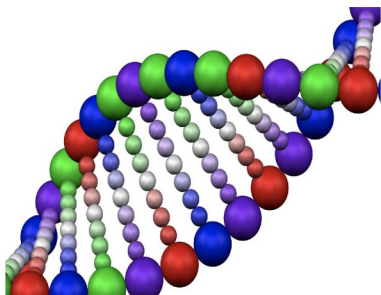
Symbolic aggregate expression (SAX)

Time Series becomes a text sequence that we can compare.

Previous : abddccbaa

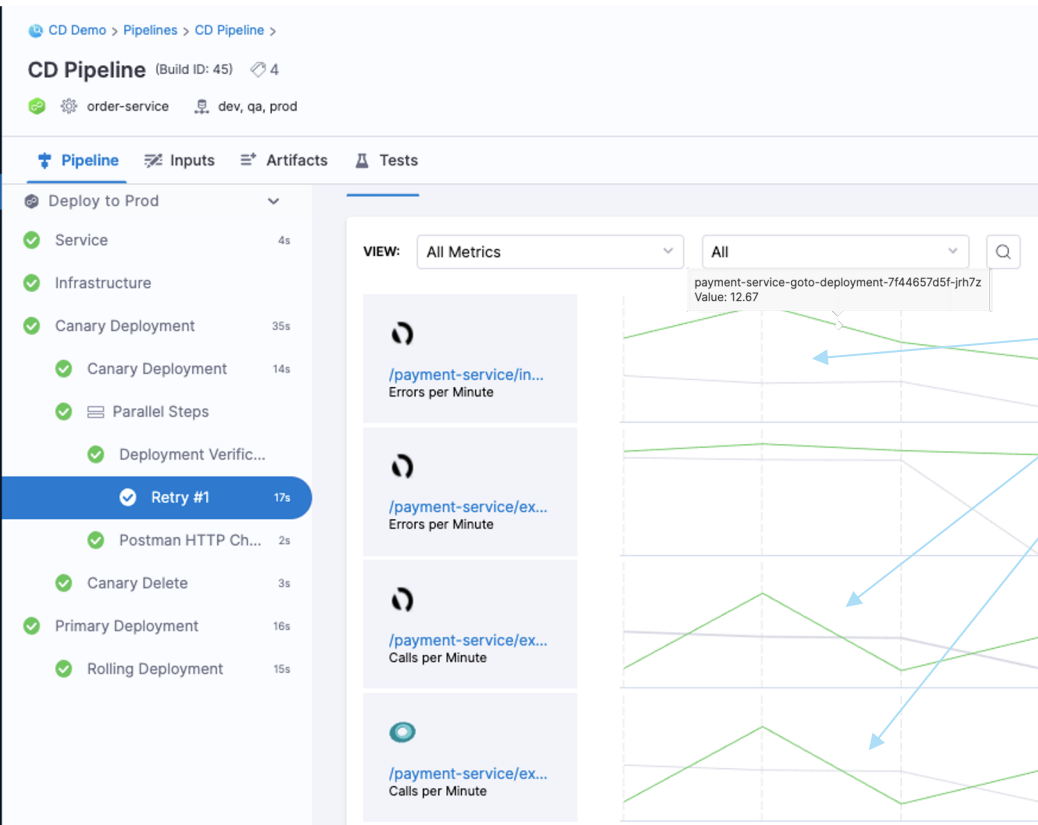
Current : abbccddba

Does this sound familiar ?



https://jmotif.github.io/sax-vsm_site/morea/algorithm/SAX.html





Peaks are not impacting the overall understanding of the Application's performance

Application Performance Logs

splunk>

 **logz.io**

 **bugsnag**

 **elastic**

 **sumologic**

 **Stackdriver**

Do we need all of them?

splunk> App: Search & Re... Messages Settings Activity Find Jason Skowronski Search & Reporting

Search Datasets Reports Alerts Dashboards

New Search

host="ip-172-31-3-221"

3,814,823 of 3,827,406 events matched No Event Sampling

Events (1,528,696) Patterns Statistics

Format Timeline Zoom Out Zoom to Selection

List

< Hide Fields All Fields

Selected Fields

a host 1

a source 1

a sourcetype 1

Interesting Fields

linecount 1

a splunk_server 1

Dashboard [Filebeat System] Syslog dashboard ECS

Syslog events by hostname [Filebeat System] ECS

Syslog hostnames and processes [Filebeat System] ECS

Syslog logs [Filebeat System] ECS

Count

@timestamp per 10 minutes

Dans-MacBook-Pro

Dans-MacBook-Pro...

log -bash 108x30

total 77656

drwxr-xr-x 3 root wheel 96 Jan 21 15:20 Bluetooth

-rw-r--r-- 1 root wheel 12 Jan 21 15:22 CDIS.custom

drwxr-xr-x 2 root wheel 64 Aug 21 2018 CoreData

drwxr-xr-x 33 root admin 1056 Apr 25 01:23 DiagnosticMessages

drwxr-xr-x 4 root wheel 128 Jan 29 05:16 apache2

drwxr-xr-x 31 root wheel 992 Apr 25 09:25 asl

drwxr-xr-x 2 root wheel 64 Aug 21 2018 com.apple.xpc.launchd

-rw-r--r-- 1 root wheel 15855878 Apr 25 09:56 corecaptured.log

drwxr-xr-x 3 root wheel 96 Feb 1 19:55 cups

-rw-r--r-- 1 root wheel 255497 Apr 24 28:38 daily.out

drwxr-xr-x 5 _displaypolicy _displaypolicy 160 Oct 29 19:51 displaypolicy

-rw-r--r-- 1 root wheel 134 Oct 29 19:50 displaypolicy.stdout.log

drwxr-xr-x 3 root wheel 96 Aug 18 2018 emond

-rw-r--r-- 1 root wheel 46922 Apr 24 23:57 fsck_apfs.log

drwxr-xr-x 1 root wheel 156 Apr 24 23:57 fsck_apfs_error.log

-rw-r--r-- 1 root wheel 55977 Apr 25 09:23 fsck_hfs.log

-rw-r--r-- 1 root admin 4270497 Apr 25 08:52 install.log

-rw-r--r-- 1 root wheel 1178 Apr 11 19:30 monthly.out

drwxr-xr-x 18 root admin 576 Apr 25 08:05 powermanagement

drwxr-xr-x 2 root wheel 64 Aug 23 2018 ppp

-rw-r--r-- 1 root wheel 0 Oct 29 19:51 shutdown_monitor.log

-rw-r--r-- 1 root admin 134625 Apr 25 18:29 system.log

-rw-r--r-- 1 root admin 27510 Apr 25 08:05 system.log.0.gz

-rw-r--r-- 1 root admin 109057 Apr 24 08:05 system.log.1.gz

-rw-r--r-- 1 root admin 62887 Apr 23 08:05 system.log.2.gz

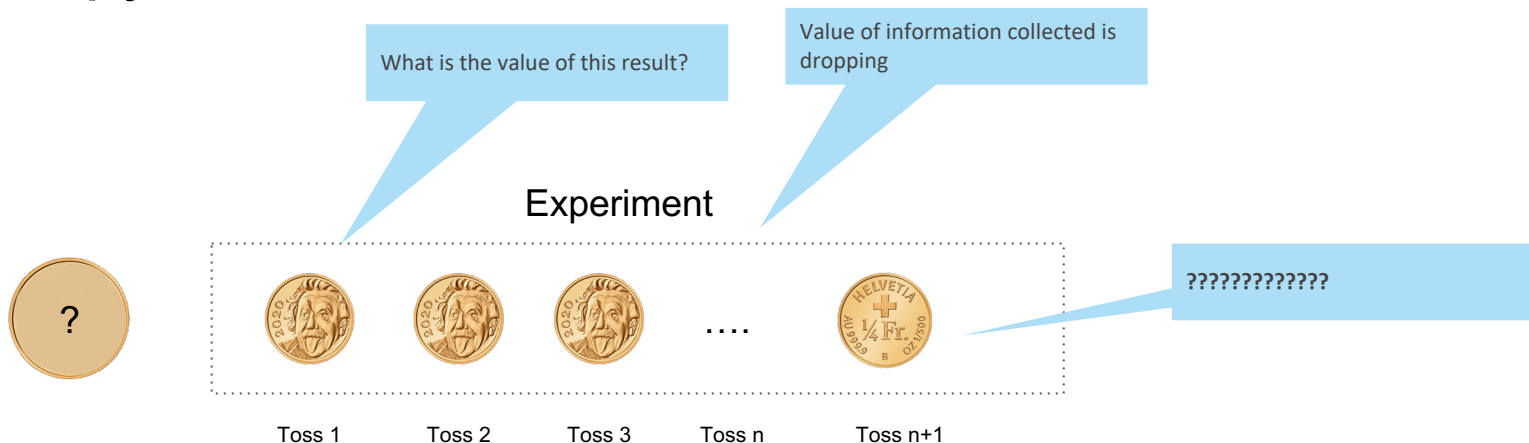
-rw-r--r-- 1 root admin 17042 Apr 22 08:05 system.log.3.gz

-rw-r--r-- 1 root admin 86761 Apr 21 08:05 system.log.4.gz

-rw-r--r-- 1 root admin 24738 Apr 20 08:05 system.log.5.gz

-rw-r--r-- 1 root admin 30312 Apr 19 08:05 system.log.6.gz

Entropy



Shannon entropy calculator

Real example how to calculate and interpret information entropy

Your string is: 1111111011111

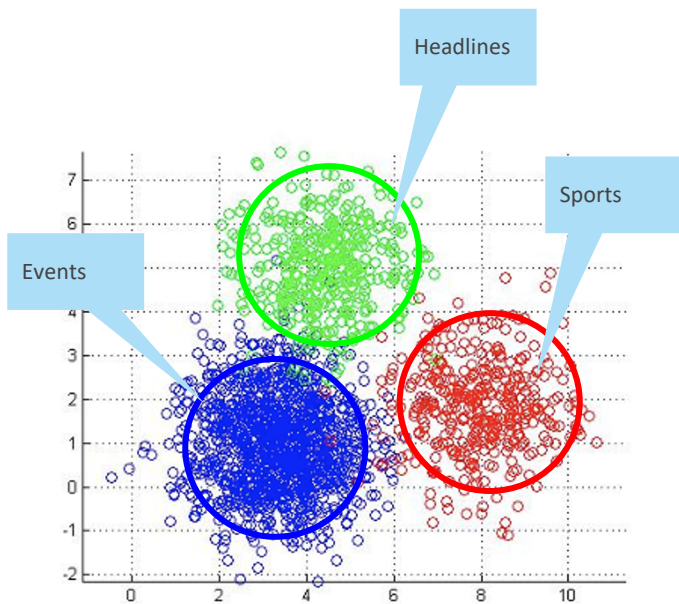
Alphabet of symbols in the string: 0 1

Frequencies of alphabet symbols:

- 0.077 -> 0
- 0.923 -> 1



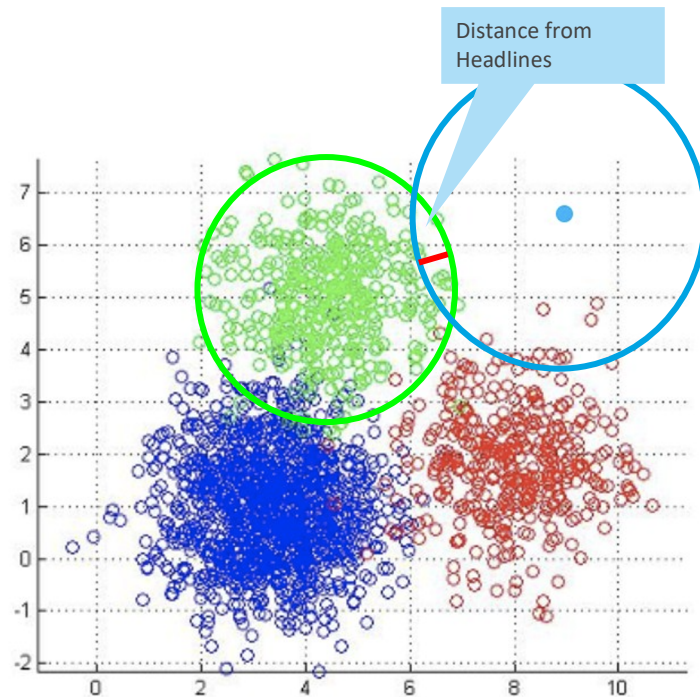
K-Means with Jaccard Distance



Trained pre-existing clusters
(Assuming all seeds are
exhausted)



New tweet
needs to
be
assigned
to a cluster



Twitter has predefined tweet categories (K Clusters)

BusinessException



ClassNotFoundException



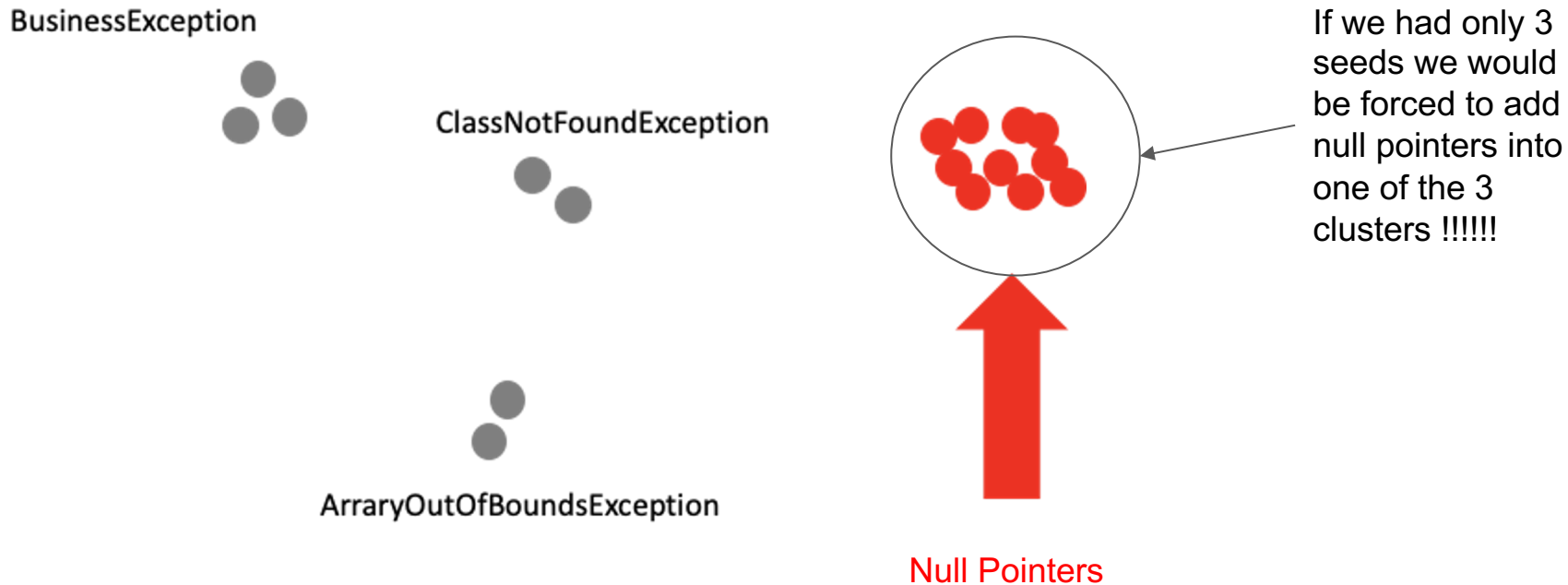
ArrayOutOfBoundsException



Three Clusters
resulted in good
allocation of data



Twitter has predefined tweet categories (K Clusters)



Why not increase the number of Cluster to a bigger number?

{ Noise }

```
2017-11-11 05:57:55, 239, 996 [32m[pool-10-thread-6]][0;39m
[31mWARN [0;39m
[36msoftware.wings.service.impl.AlertServiceImpl[0;39m -
Alert opened:
Alert(accountId=UMn5VYHwSYuzO9EPzFTVhQ,
type=NoActiveDelegates, status=Open, title=No delegates
are available, category=Setup, severity=Error,
alertData=NoActiveDelegatesAlert(accountId=UMn5VYHwSYu
zO9EPzFTVhQ), closedAt=0)
```

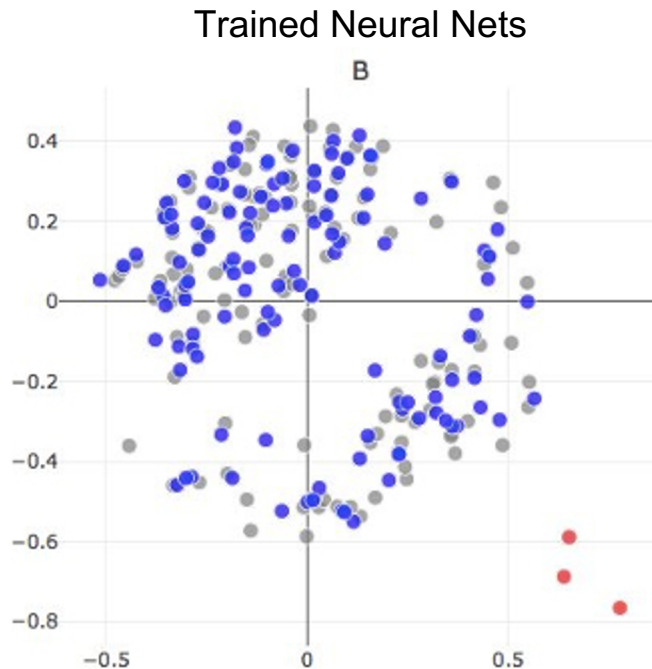
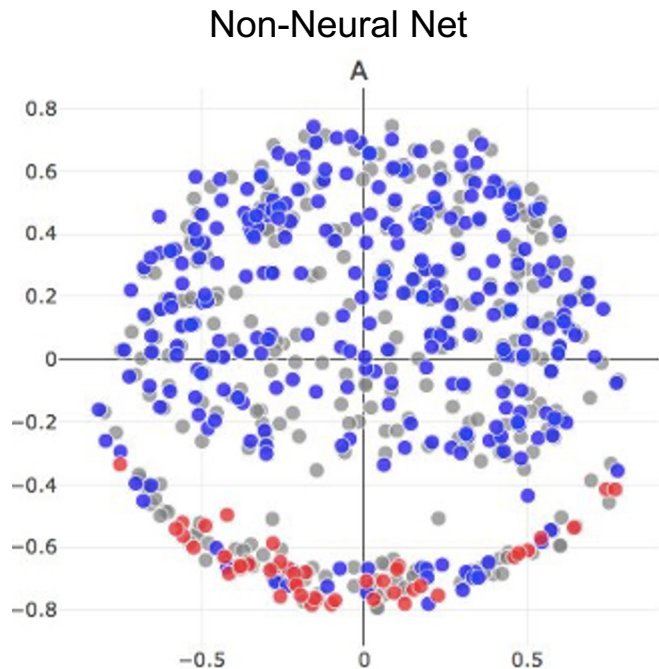
```
2017-11-10 20:33:26, 996 [32m[pool-10-thread-6]][0;39m
[31mWARN [0;39m
[36msoftware.wings.service.impl.AlertServiceImpl[0;39m -
Alert opened: Alert(accountId=axNvTFUnSr2Fi4hMRZdteA,
type=NoActiveDelegates, status=Open, title=No delegates are
available, category=Setup, severity=Error,
alertData=NoActiveDelegatesAlert(accountId=axNvTFUnSr2Fi
4hMRZdteA), closedAt=0)
```

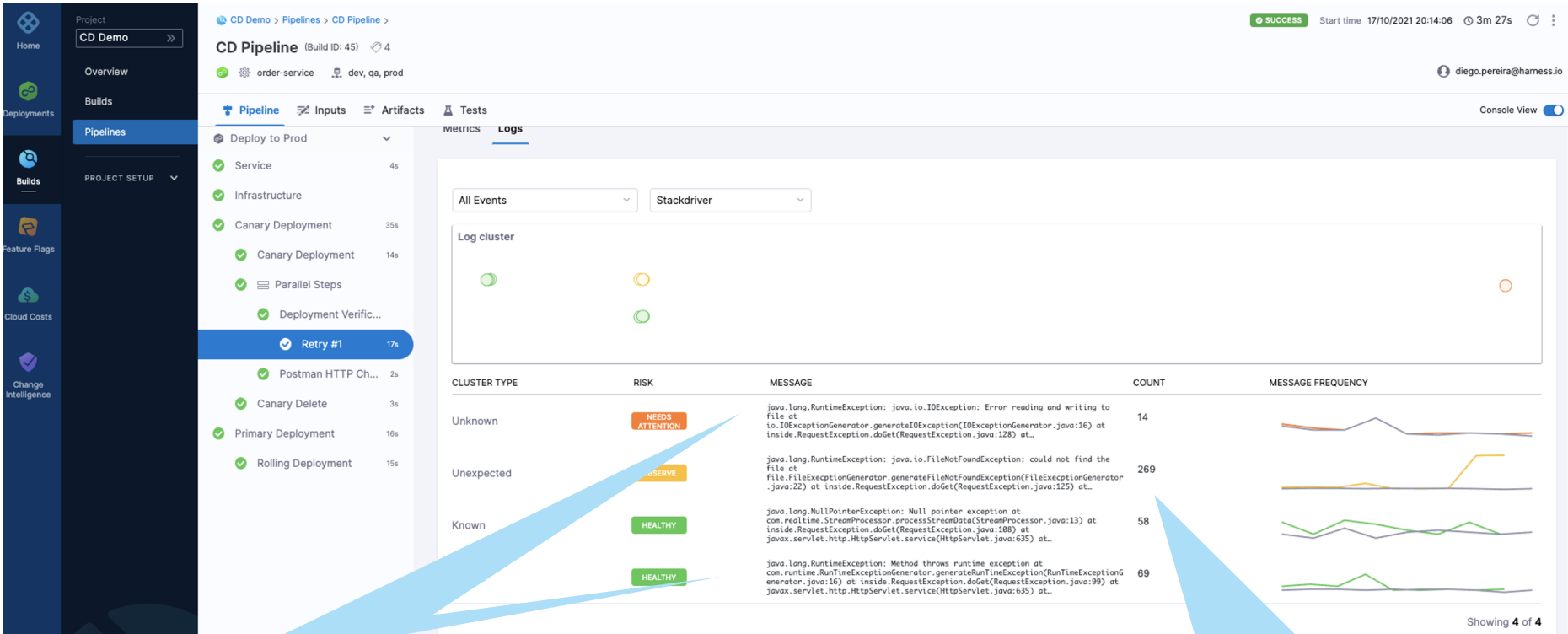


Same Log error different cluster, why?



Noise? Use neural nets to define the K





RuntimeExceptions split into different clusters based on the message/context

Neural Nets removing the noise of unique data points (IDs)





Compelling Event:

Migrating to AWS Public Cloud.

Time-To-Value:

Onboarded 10 app teams (~100 services) in less than 2 months. Most teams onboarded on day 1.



Martin Reynolds
Head of DevOps
Advanced

Before Harness

After Harness

DevOps deploy	→	Developers deploy
2 days to deploy	→	2 hours to deploy
2 days to onboard dev teams	→	2 hours to onboard dev teams
Manual Verification & Rollback	→	Automated Verification & Rollback

10X ROI in 2 Months



Deploy to Prod

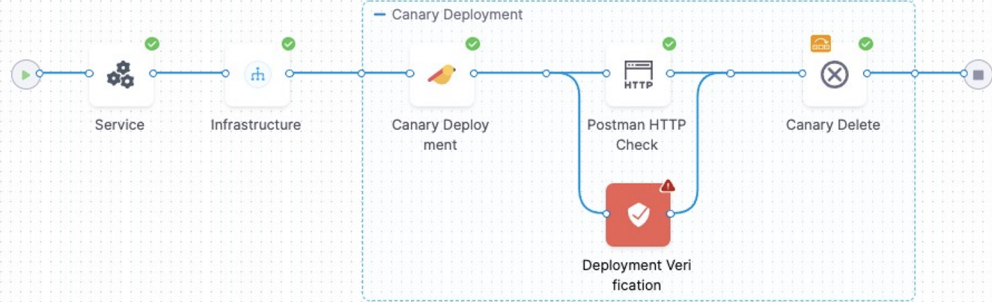
Started at: 22/10/2021, 11:08:20

Duration: 2h 34s

Service(s)

Environment(s)
prod

FAILED Error Summary



Step: Deployment Verification

Started On: Oct 22, 2021 10:54 AM
Duration: 5 minutes

FAILED

PRIMARY

CANARY



SUMMARY

8 OUT OF 24
Metrics in violation
[View Details →](#)

2 OUT OF 4
Log Clusters in violation
[View Details →](#)



Compelling Event:

Existing CI/CD process was not able to scale

Time-To-Value:

DevOps Engineers deploying on-demand in a matter of weeks.



Tommy Bø
DevOps Engineer
Entur

Before Harness

After Harness

Custom scripts containing
1000s of lines of XML



0 XML configurations

Weekly deployments



Deploy on-demand

20% change failure rate



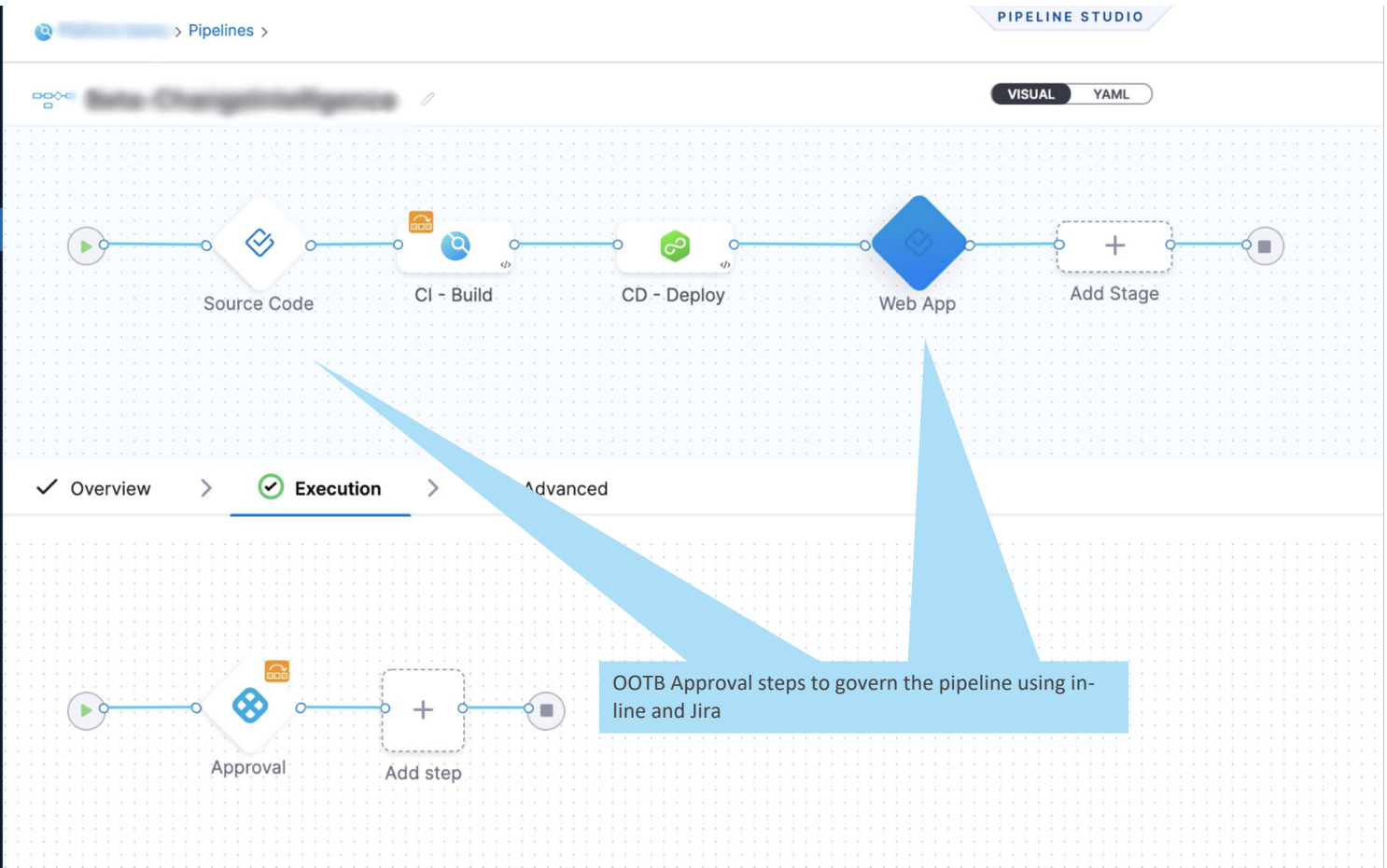
5% change failure rate

Deployment failure required
a 2 hour manual roll-back



Automated Verification and
Rollback

75% Reduction in Downtime

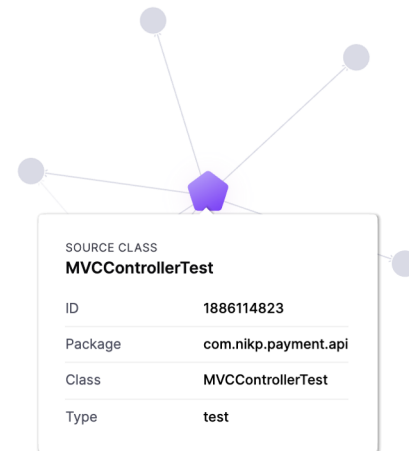


Harness - Applying Machine Learning to CI



Call Graph: `com.nikp.payment.api.MVCTestControllerTest`

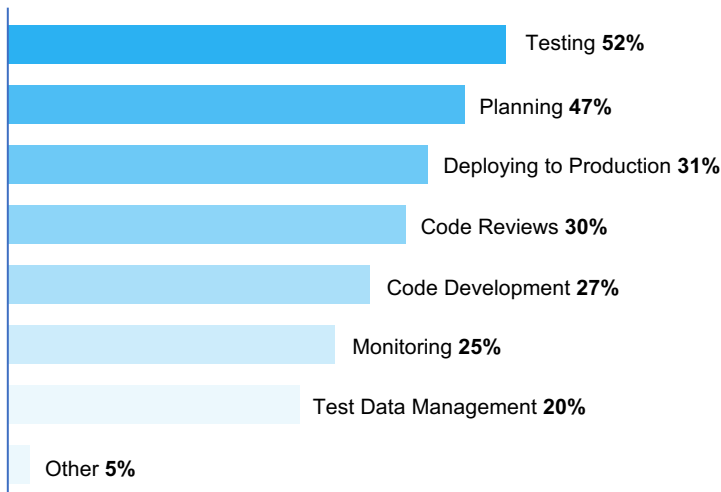
- several metrics
- dependencies
- touch mode code
- likely to fail
- overtesting



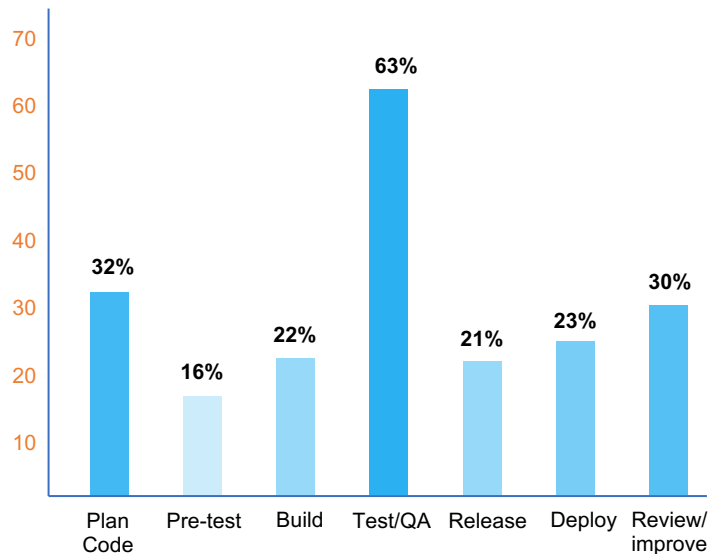
Testing is the Biggest Bottleneck!

A recent GitLab survey that targeted developers and engineers found that testing is responsible for more delays than any other part of the development process.

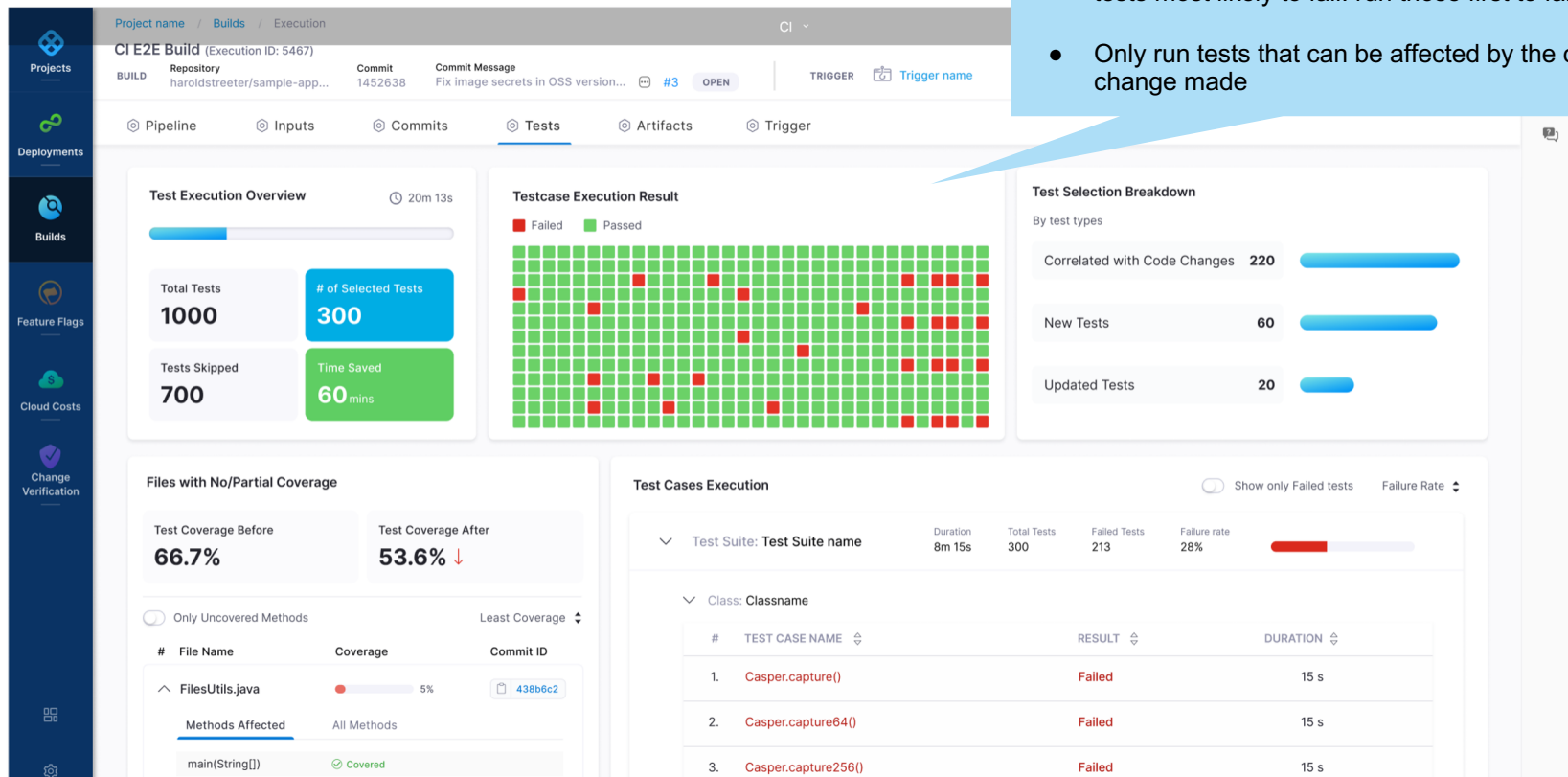
Where in the development process do you encounter the most delays?



Where are the Main Hold-ups in the Software Production Process?



Test Intelligence



- Machine Learning used to determined which tests most likely to fail: run those first to fail fast
- Only run tests that can be affected by the code change made

Test Intelligence Results

Test Intelligence Applied to Open-Source Projects

Project Name	Avg Build Execution Time Without TI	Avg Build Execution Time With TI	% Time Savings
Portal (Harness)	43 mins	32 mins	25%
Incubator Pinot	338 mins	228 mins	32%
Hudi	58 mins	43 mins	25%
RocketMQ	4.6 mins	3.1 mins	32%
Spring Cloud Alibaba	0.744 mins	0.59 mins	20%
Incubator Shenyu	1.16 min	0.4 min	65%
Sentinel	1.90 min	1 min	47%

35%

Average Time Saved



Thank you!