



“Will This Never End?”

Marco Coulter  
Technical Evangelist  
AppDynamics  
@marcoccoulter  
marco.coulter@appdynamics.com

## Notice

This presentation may contain proprietary or product roadmap information of AppDynamics, Inc. ("AppDynamics" or the "Company"). AppDynamics reserves the right to change any product roadmap information at any time, for any reason and without notice. This information is intended to outline AppDynamics' general product direction, it is not a guarantee of future product features, and it should not be relied on in making a purchasing decision. The development, release, and timing of any features or functionality described for AppDynamics' products remains at AppDynamics' sole discretion. AppDynamics reserves the right to change any planned features at any time before making them generally available as well as never making them generally available.

All third-party trademarks, including names, logos and brands, referenced by AppDynamics in this presentation are property of their respective owners. All references to third-party trademarks are for identification purposes only and shall be considered nominative fair use under trademark law.

# Agenda

---

- Will this never end?
- You want me to do what?
- Will this thing work?
- Oops



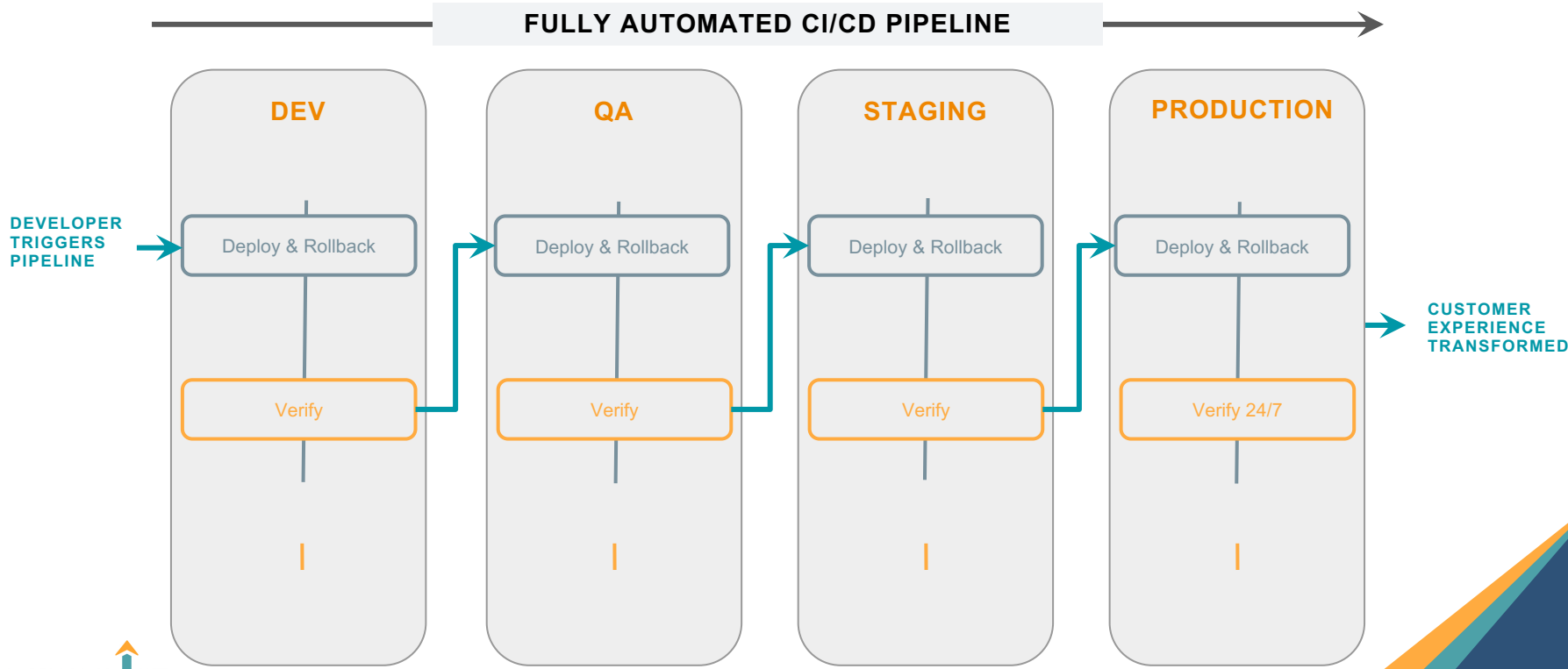
## Marco Coulter

Technical Evangelist

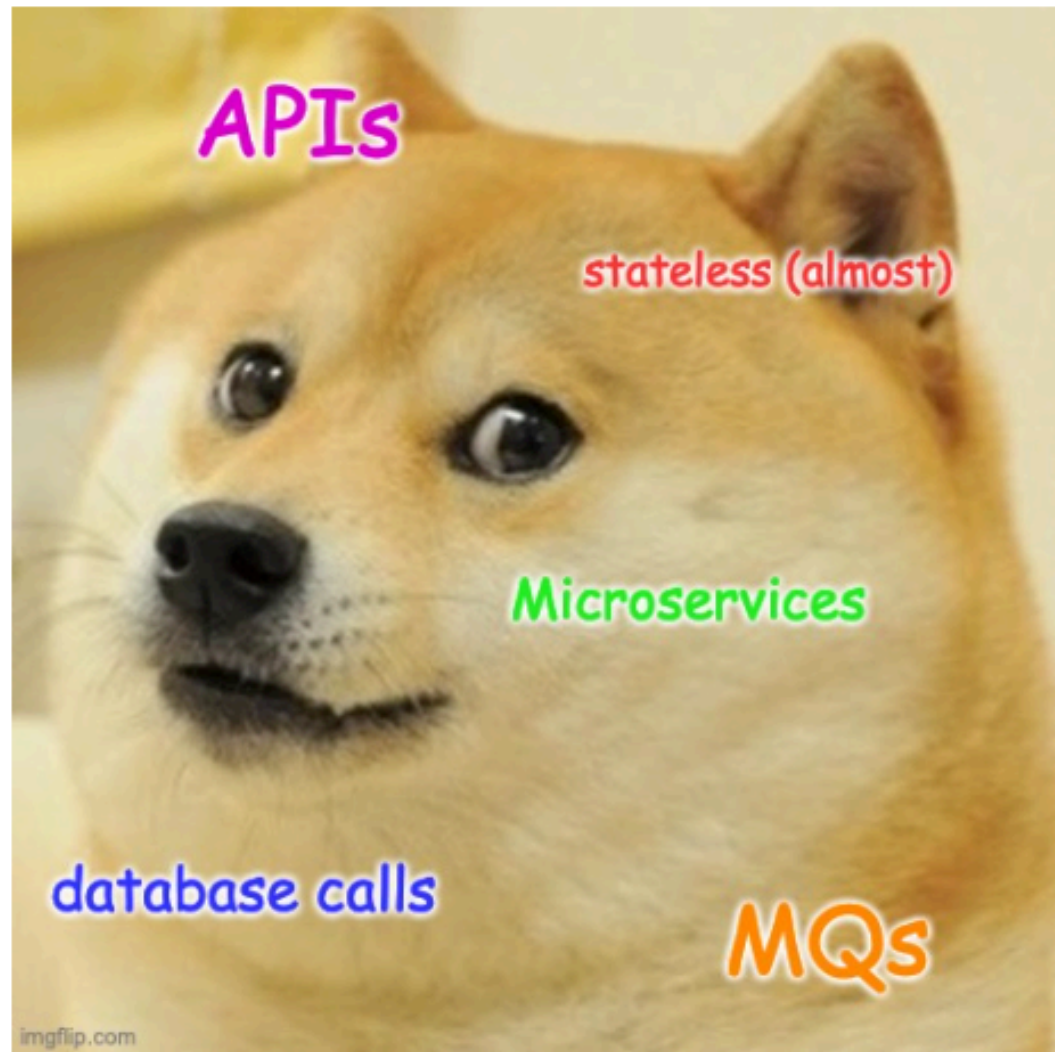
[@marcocoulter](#)

Marco Coulter is passionate about the experience that humans have when interacting with technology. A former startup CTO, Marco led the data science organization at 451 Research; managed global engineering and operations organizations at several major banks and hospitals. He earned him the nickname "tech-whisperer" for his skills in translating business drivers for techies and technical concepts for business leaders. When taking the rare break from technology, Marco can be found harvesting fresh vegetables from his apartment garden.

# Continuous Delivery & Verification



“Will this never  
end?”



# The Example App

ACME APPDYNAMICS Home Applications User Experience Databases Servers Analytics Dashboards & Reports Alert & Respond

ACME ACME Baseline... last 5 minutes

Dashboard Network Dashboard Events Top Business Transactions Transaction Snapshots Transaction Score Actions

Application Flow Map

```

    graph LR
      NodeBackend((1 Node  
5 calls / min  
4 ms  
NodeBackend)) -- "5 calls / min, 15 ms  
HTTP" --> WebTier((1 Node  
11 calls / min  
12 ms  
WebTier))
  
```

Events  
No Events in selected time range

Business Transaction Health  
0 critical, 0 warning, 11 normal

Node Health  
0 critical, 0 warning, 2 normal

Server Health

Transaction Scorecard

Normal	97.9 %	47
Slow	2.1 %	1
Very Slow	0.0 %	-
Stall	0.0 %	-
Errors	0.0 %	-

Exceptions

Exceptions	-total	- / min
HTTP Error Codes	-total	- / min
Error Page Redirects	-total	- / min

Legend

Load 48 calls 16 calls / min

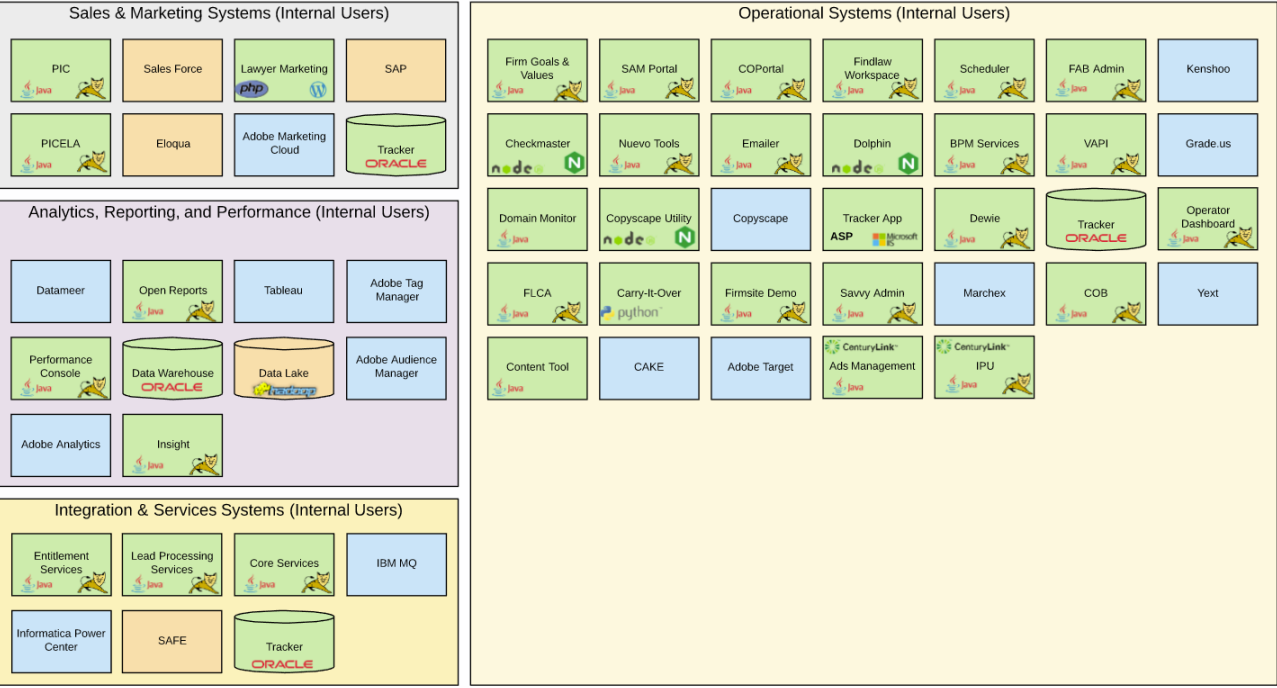
Response Time (ms) 78 ms average

Errors - % - errors - errors / min

No data to display



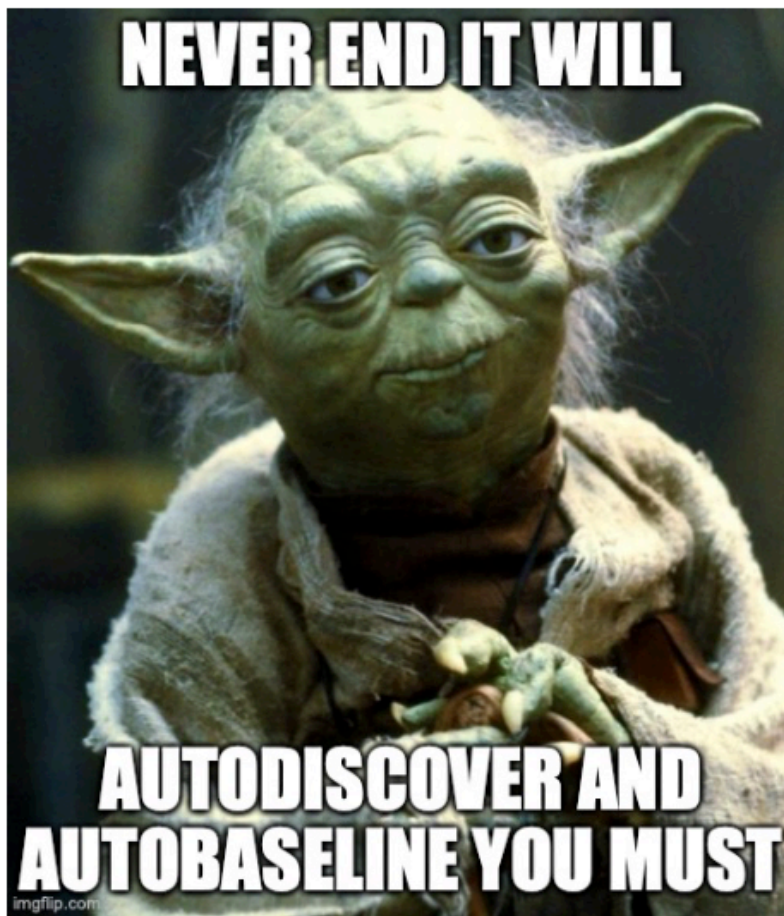
# What we think the code is doing



Findlaw Supported
  TR Supported
  Vendor Supported

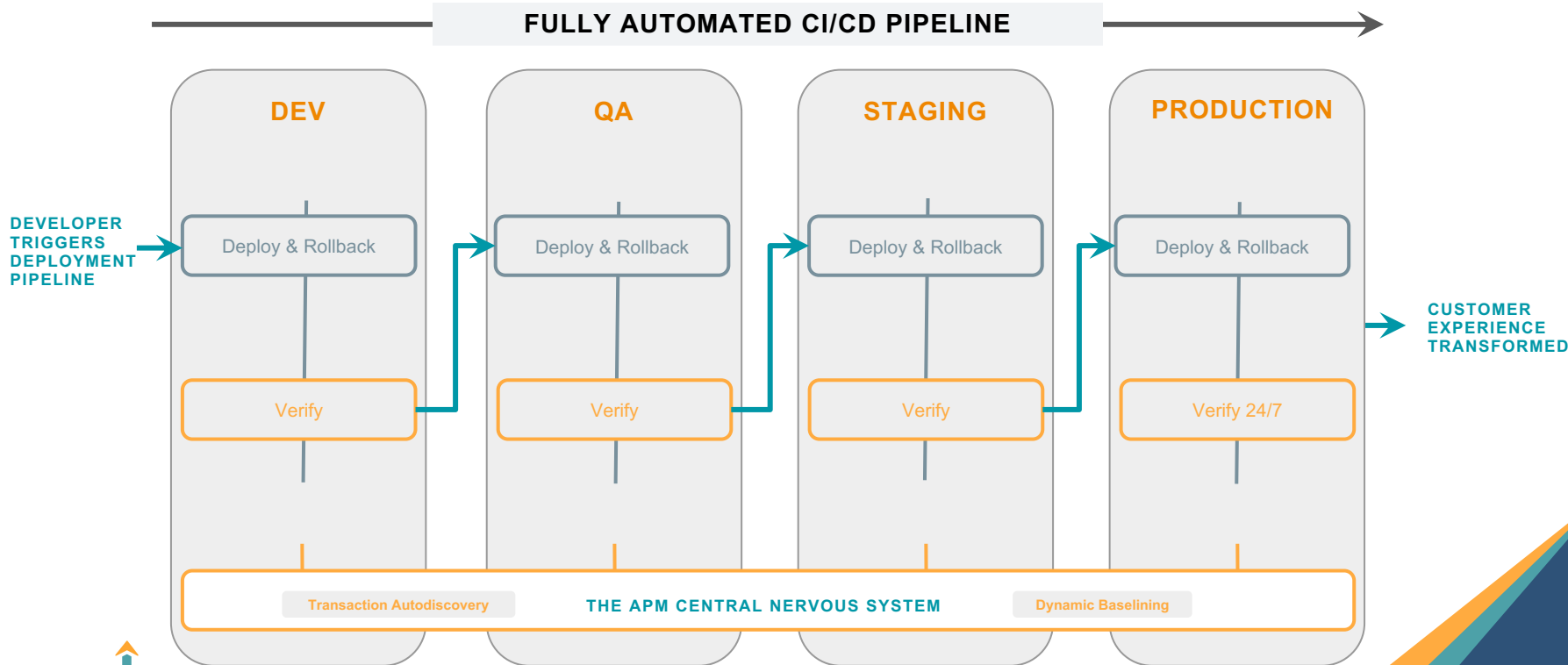






## Lesson One

# Continuous Delivery & Verification



“You want me  
to do what?”



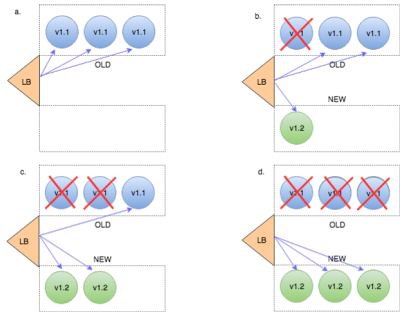
**AFTER SEEING THE BUILD SCRIPTS**

imgflip.com

# Deployment Options

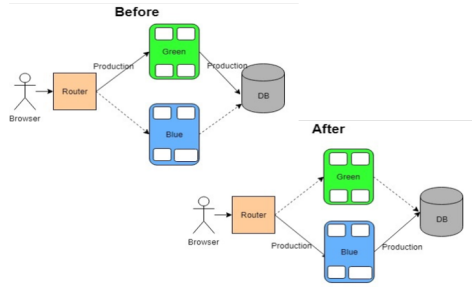
## ROLLING UPGRADE

For many PaaS's this is the default deployment. Slowly scale up the new version while simultaneously slowly scaling down the old version.



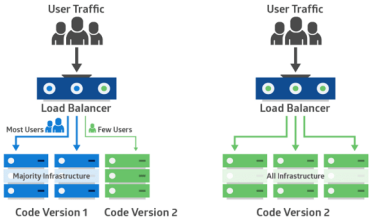
## BLUE/GREEN DEPLOYMENT

Stand up a complete deployment of the new version (blue) while leaving a complete copy of the old version (green) up and running.



## CANARY DEPLOYMENT

A variation of Blue/Green. Stand up a small deployment of the new version (the canary) and direct a small portion of incoming traffic to the canary

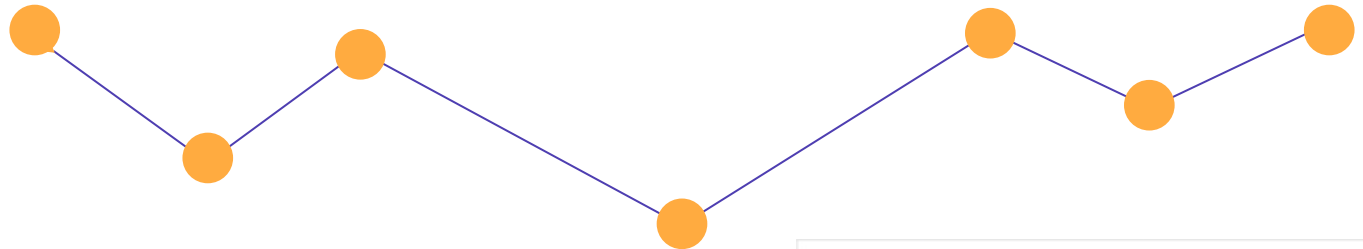


# Continuous Deploy and Rollback

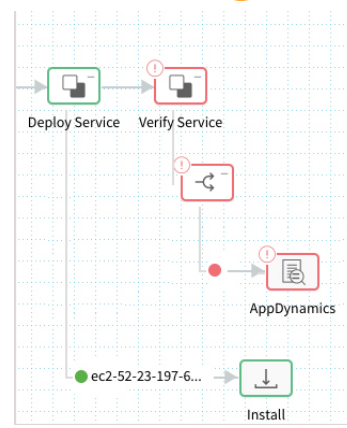
VISIBILITY

INSIGHT

ACTION



- Identify slow application response time
- Measure Business Impact of Performance Degradation



**APPDYNAMICS**

Business Transaction Analysis

Overall Risk Level ◆ HIGH

3 high risk transactions found.

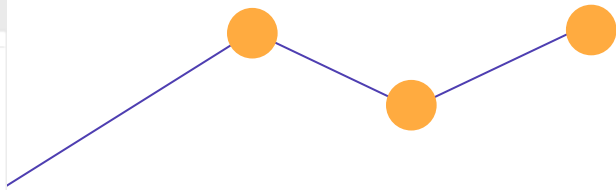
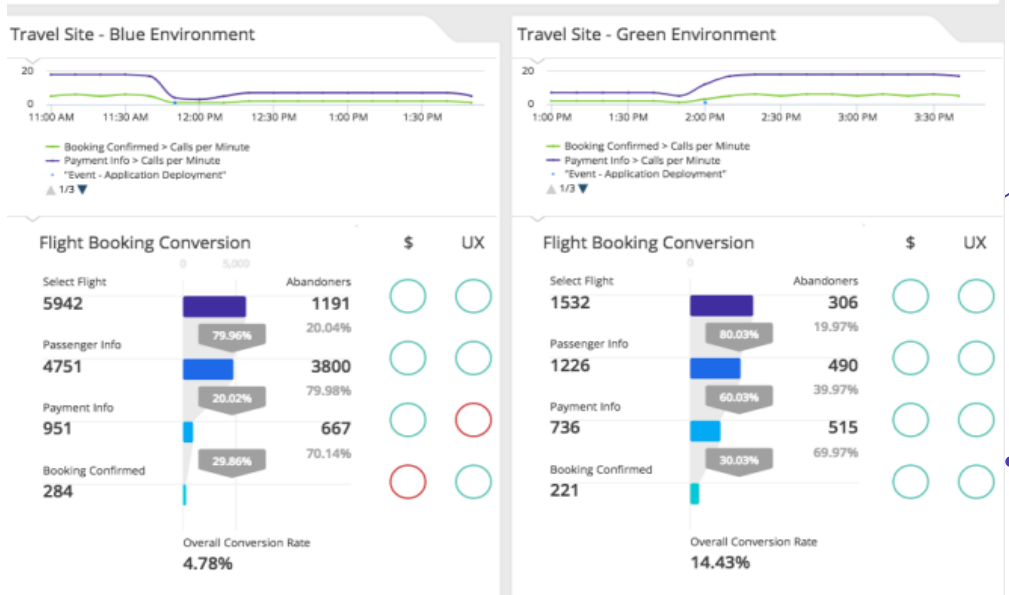
Stalls	Response Time(95%)	Slow	Revenue	Business Transaction
●	●	●	●	Web Transaction: Checkout
●	●	●	●	Revenue Value: \$3,200/min Previous Value: \$14,100/min
●	●	●	●	Risk: <span style="color: red;">◆ HIGH</span>
●	●	●	●	Checkout

# Continuous Deploy and Rollback

VISIBILITY

INSIGHT

ACTION



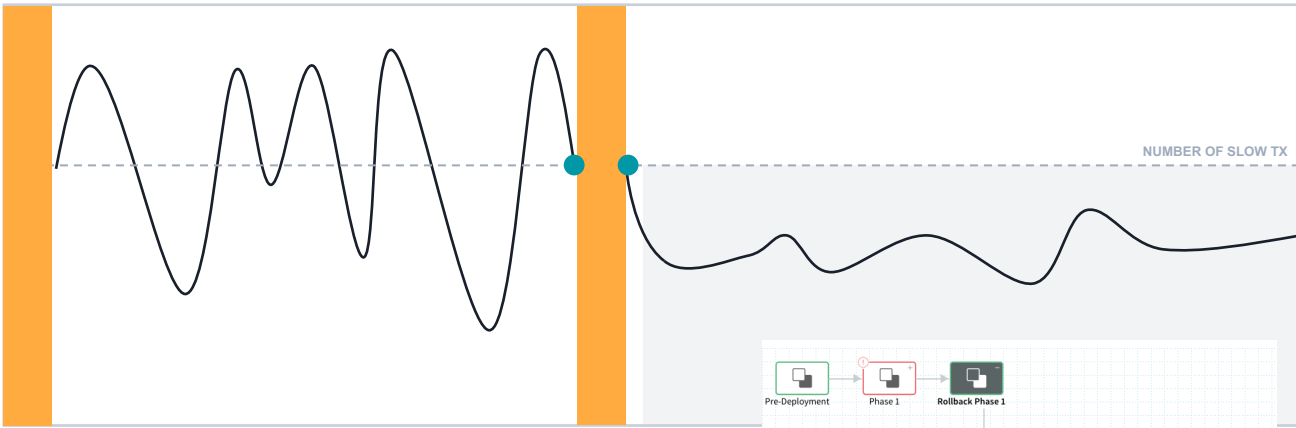
Know the business & UX impact  
Make a data-driven decision

# Continuous Deploy and Rollback

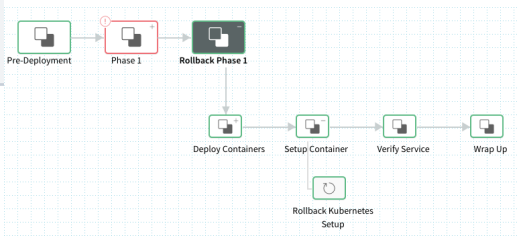
VISIBILITY

INSIGHT

ACTION



Canary/Blue Green Releases are stopped when performance issues are identified



**AUTOMATE**

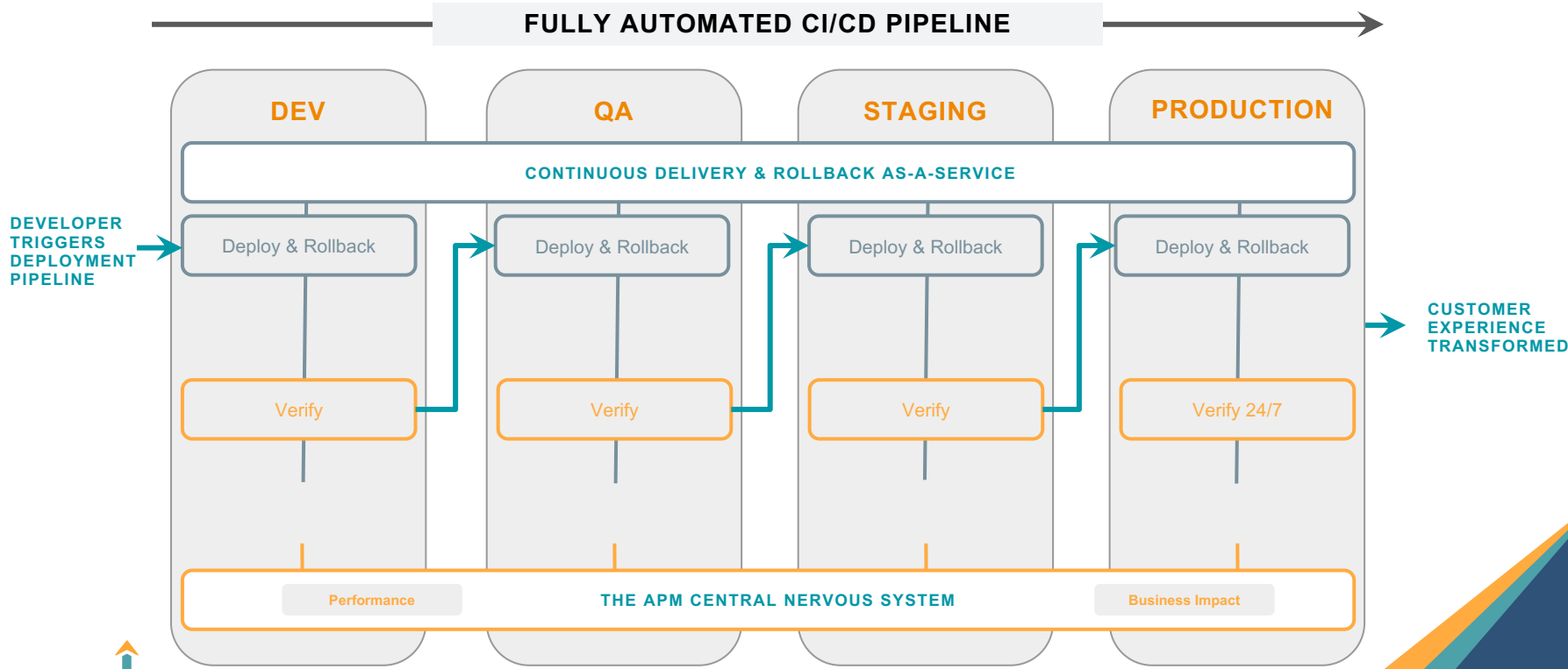
**ALL DEPLOYS AND ROLLBACKS**

imgflip.com

## Lesson Two



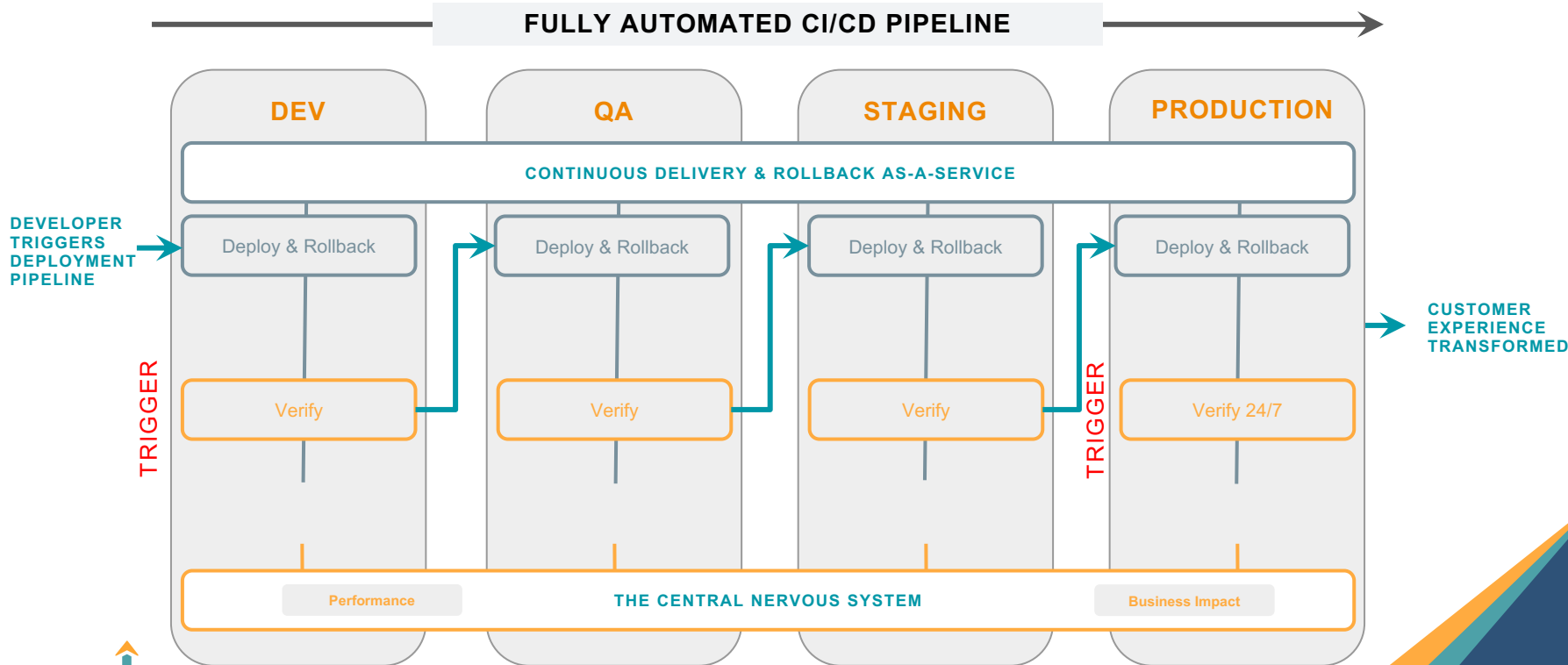
# Continuous Delivery & Verification





## Lesson Three

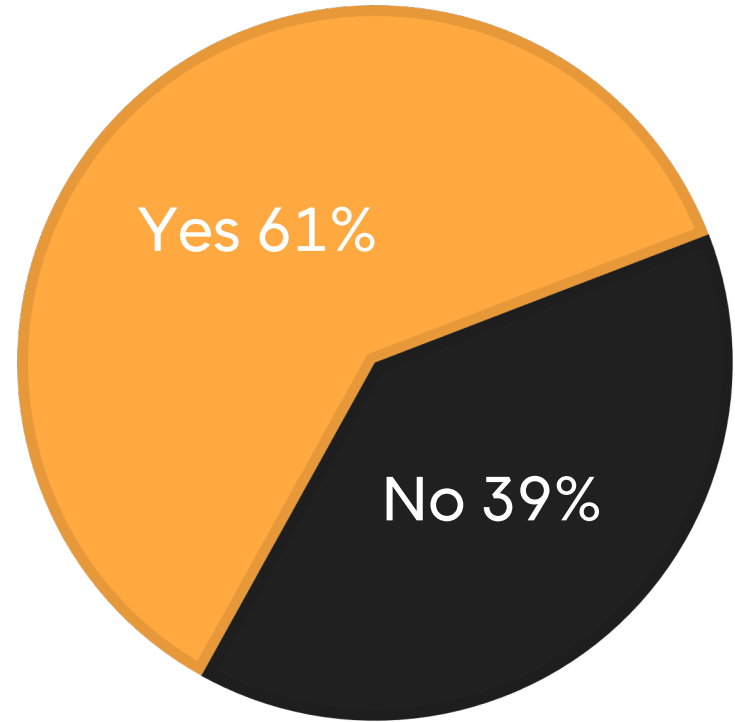
# Continuous Delivery & Verification



“Will this thing  
work?”



Q: Suffers from  
release anxiety?



Source: DevOps.com webinar, June 2020

# This is what verification looks like

Create Scripts >

ZebraTester

Zebra IDE

Zebra YAML



Setup Monitoring >

Apica

APPDYNAMICS  
part of Cisco

Build >

Jenkins

Microsoft  
Visual Studio  
Team Foundation Server

TC

Bamboo

Run Test >

Global Cloud



amazon  
WEB SERVICES

rockspace

Microsoft Azure

Apica

On premises



Analyze >



Report >

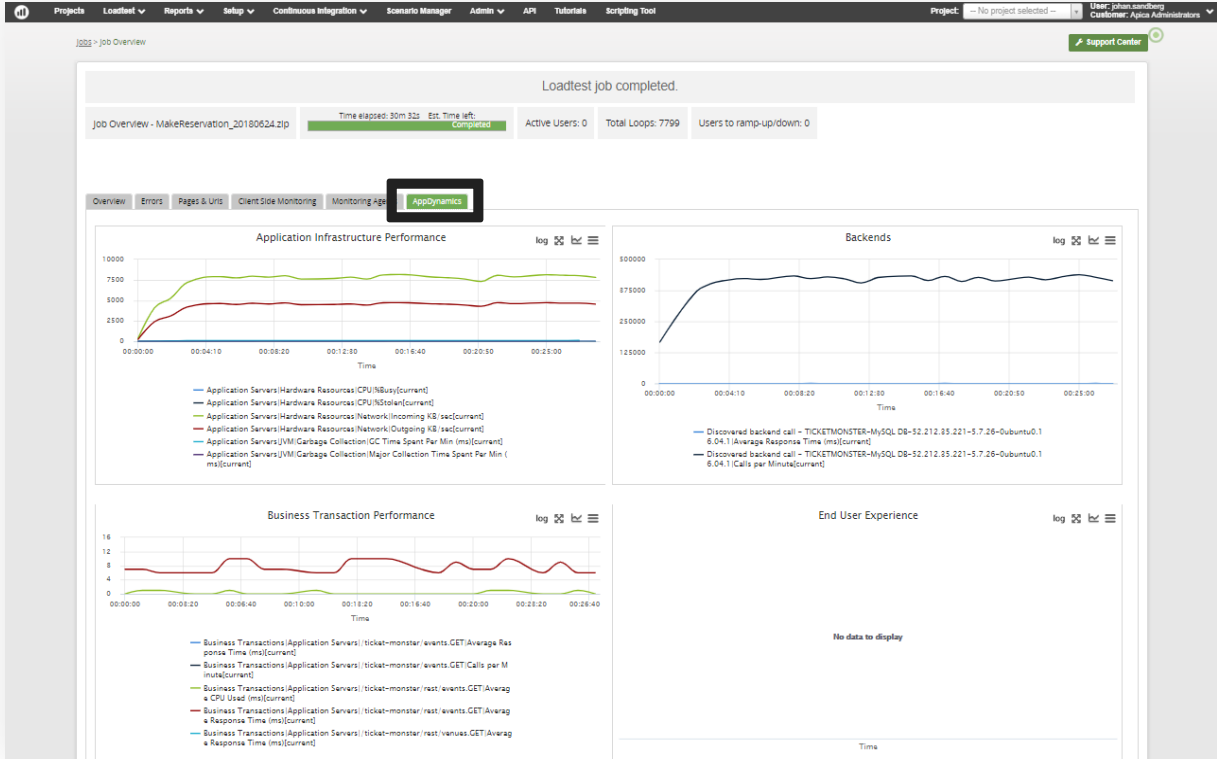


Ensure scripts match skills

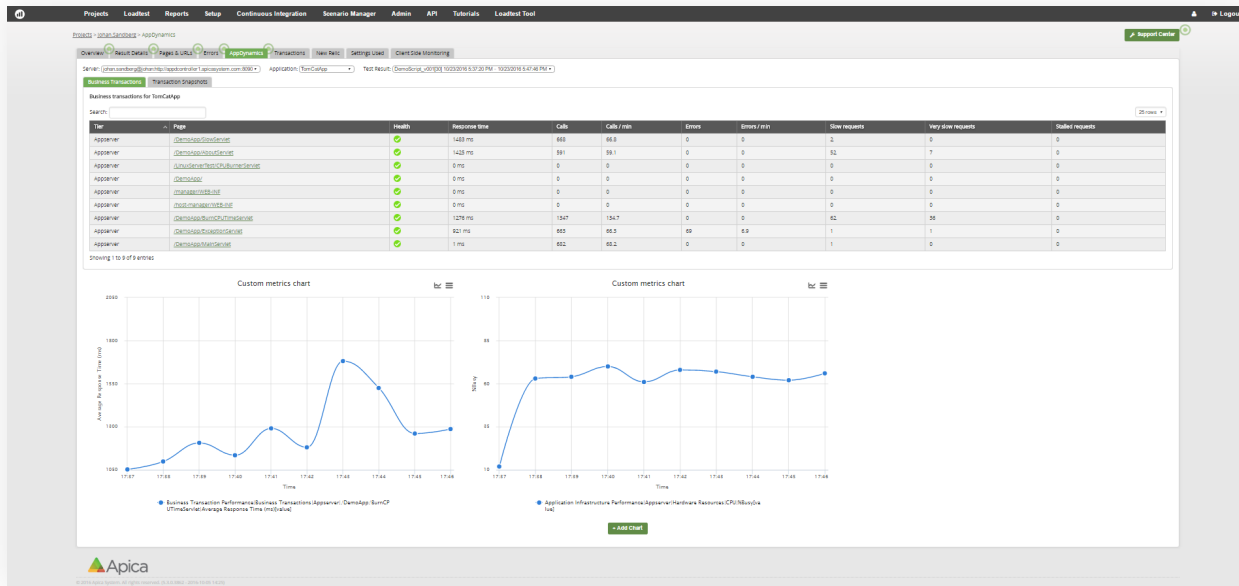
Save metrics selections as part  
of test config

Pull APM metrics into the test  
analysis

# This is what integrated verification looks like

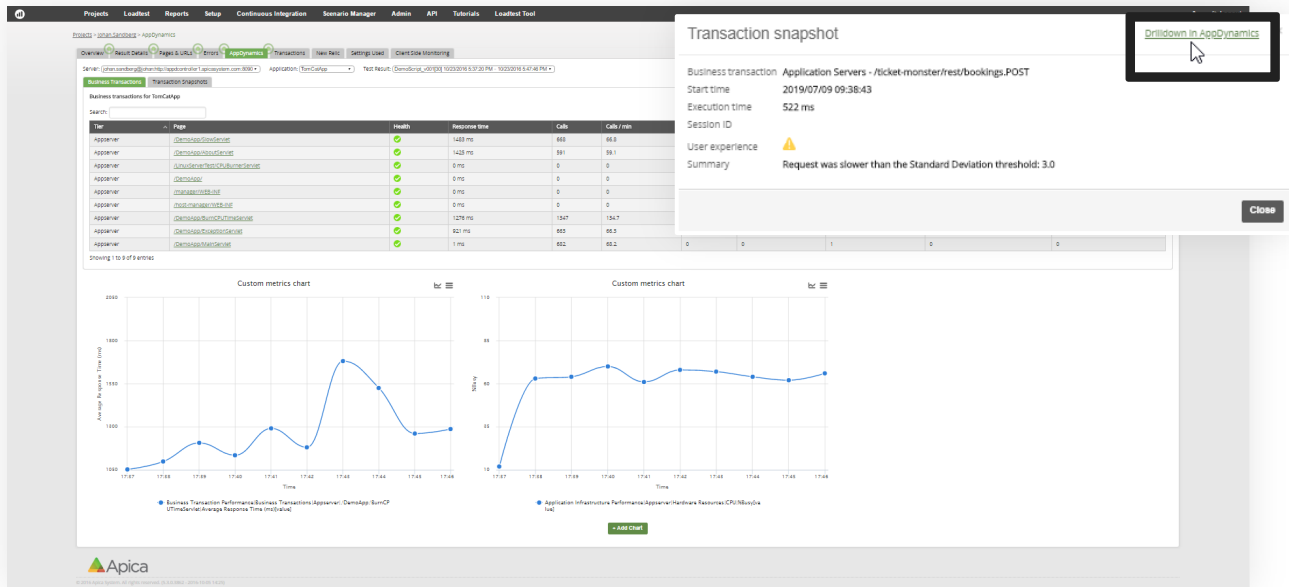


# Integration is the key to success





# Integration is the key to success





# Integration is the key to success

The screenshot displays the Apica monitoring interface. The main dashboard shows a table of business transactions for 'TicketCatApp' with columns for Page, Health, Response time, Calls, and Calls/min. Below the table are two 'Custom metrics chart' graphs showing average response time over time. A 'Transaction snapshot' window is open, showing details for a slow transaction (ID: 6b4e6bb8-54d9-4c8f-6724-568644bc213) with an execution time of 522 ms. A 'User Experience: Slow' warning is present. A 'Potential Issues' list includes 'Thread Contention - Blocked on java.lang.Object@2b949120'. A 'Drill Down into Call Graph' window is also open, showing a call graph for the transaction with a 'Thread Contention' node highlighted, indicating a 27 ms delay.

Page	Health	Response time	Calls	Calls/min
AppServer - /tickets/monsters	OK	1427 ms	168	16.8
AppServer - /tickets/monsters/get	OK	1428 ms	161	16.1
AppServer - /tickets/monsters/create	OK	0 ms	0	0
AppServer - /tickets/monsters/delete	OK	0 ms	0	0
AppServer - /tickets/monsters/edit	OK	0 ms	0	0
AppServer - /tickets/monsters/list	OK	0 ms	0	0
AppServer - /tickets/monsters/show	OK	1276 ms	147	14.7
AppServer - /tickets/monsters/show/get	OK	82 ms	102	10.2
AppServer - /tickets/monsters/show/post	OK	1 ms	62	6.2

# Integration is the key to success

The image displays the Apica monitoring interface, which provides a comprehensive view of system performance and transaction details.

**Transaction Snapshot:** A summary of a business transaction for the endpoint `Application Servers - /ticket-monster/rest/bookings.POST`. The transaction started at `2019/07/09 09:38:43` and took `522 ms` to complete. A warning indicates that the request was slower than the standard deviation threshold of `3.0`.

**Performance Metrics:** A table lists the health and performance of various application servers. The table includes columns for `App`, `Page`, `Health`, `Response time`, `Calls`, and `Calls / min`.

App	Page	Health	Response time	Calls	Calls / min
AppServer	Dashboard/index.html	OK	1427 ms	468	98.8
AppServer	Dashboard/submitOrder	OK	1428 ms	391	99.1
AppServer	Dashboard/reserveClubMember	OK	0 ms	0	0
AppServer	Dashboard/	OK	0 ms	0	0
AppServer	Dashboard/submitOrder	OK	0 ms	0	0
AppServer	Dashboard/submitOrder	OK	0 ms	0	0
AppServer	Dashboard/submitOrder	OK	1276 ms	1427	142.7
AppServer	Dashboard/submitOrder	OK	821 ms	602	60.2
AppServer	Dashboard/submitOrder	OK	1 ms	622	62.2

**Custom Metrics Chart:** Two line graphs show performance trends over time. The left chart displays `Average Response Time (ms)` from `17:37` to `17:45`, with a peak around `17:42`. The right chart shows `Application Infrastructure Performance (Average Hardware Resources CPU/Memory)` over the same period.

**Transaction Details:** A detailed view of a specific transaction (`Transaction: ab4e8b8b-54a9-4d8f-6724-56864a4bc213`) shows a `522 ms` execution time. A `Drill Down` button is visible, leading to a call graph.

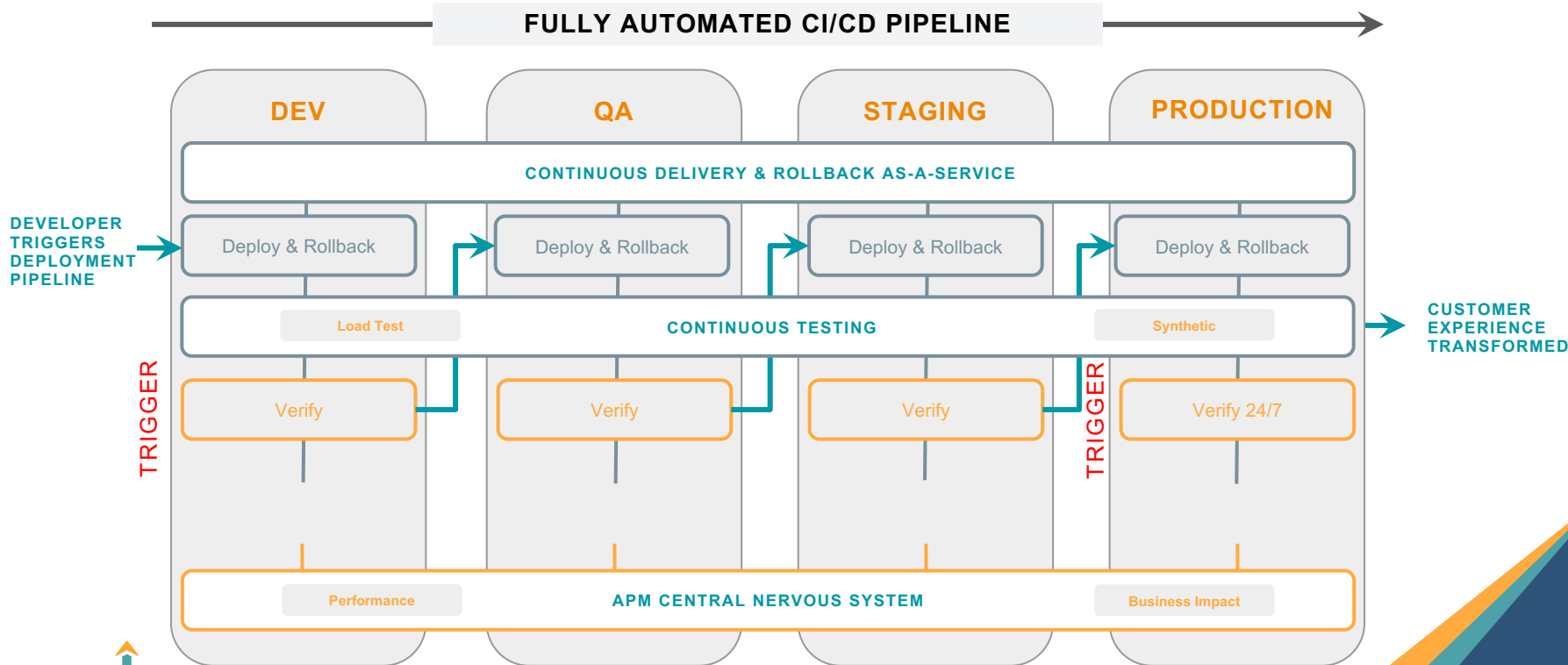
**Call Graph:** A detailed call graph for the transaction shows the execution flow through various service handlers. The `HTTPServiceHandler` is highlighted in red, indicating it is the slowest component in the call chain, taking `27 ms` (self) and `100%` of the total time.

**Legend:** A legend at the bottom of the call graph identifies the components, including `HTTPServiceHandler` and `BookingService$View42$createBooking`.

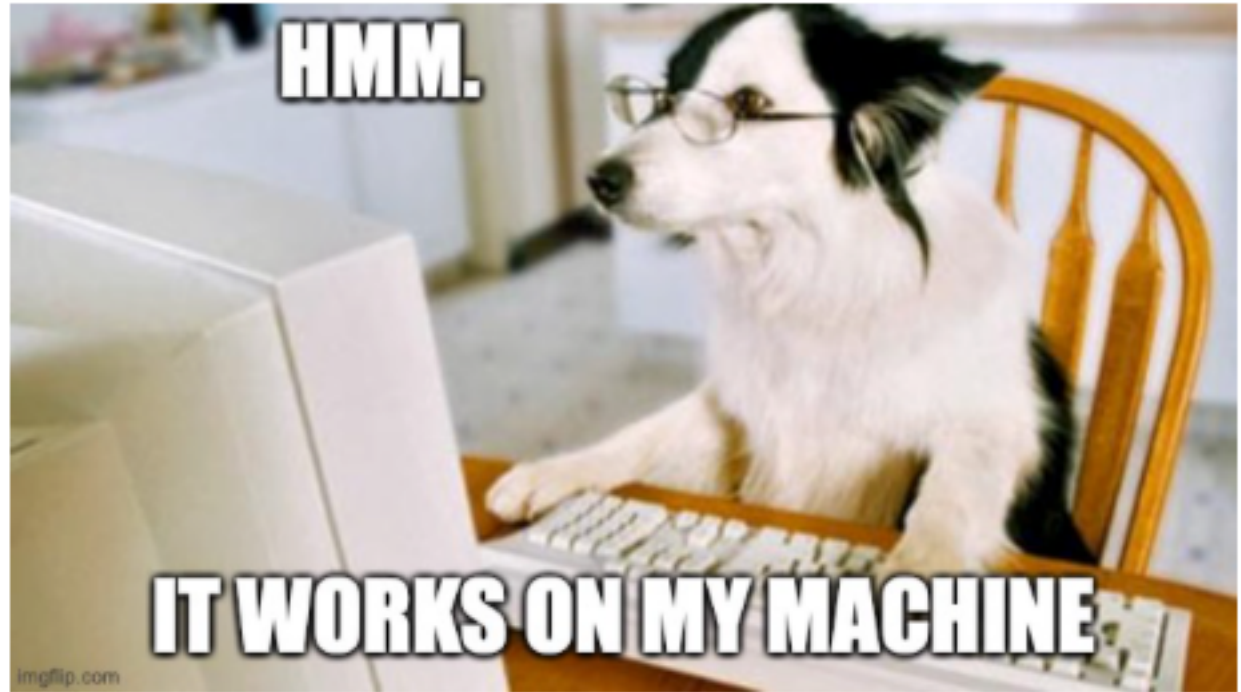


## Lesson Four

# Continuous Delivery & Verification



“Oops”



# What does it take for Dev to collect more data?

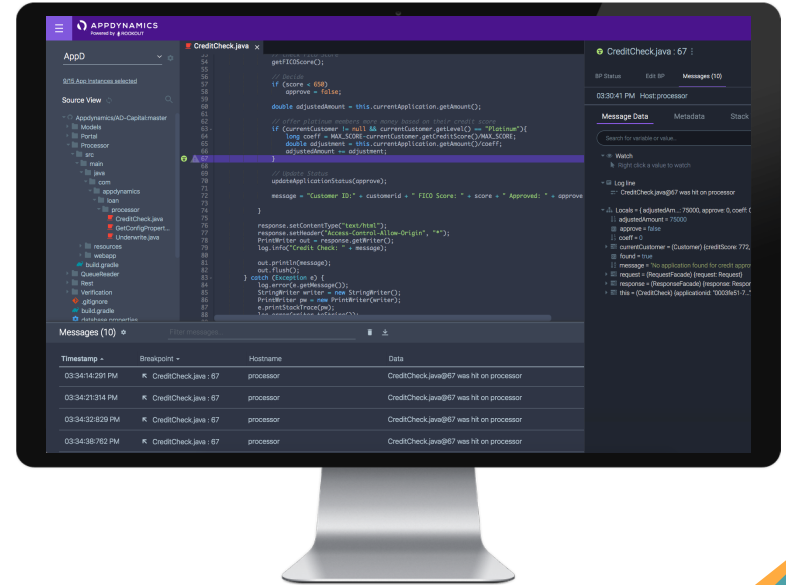
- Typical Debugging Process





# What are Non-Breaking Breakpoints?

- May be applied to a live instance of an application without changing code or restarting the app (in development, staging or production)
- N-B Breakpoint triggers internal logic to fetch local variables and send debugging data to DCI



# Deep Code Insights

Powered by Rookout

The screenshot displays the AppDynamics IDE interface. On the left, a source view shows the project structure for 'AppDynamics/AD-Capital:master', with 'processor/CreditCheck.java' selected. The main editor shows the Java code for 'CreditCheck.java' with a breakpoint set at line 67. The code includes logic for calculating adjusted amounts based on credit scores and customer levels. The right sidebar shows the 'Messages (10)' panel, which lists four messages indicating that the breakpoint was hit on the 'processor' host at various timestamps.

```
54 getFICOScore();
55
56
57 // Decide
58 if (score < 650)
59     approve = false;
60
61
62 double adjustedAmount = this.currentApplication.getAmount();
63
64 // offer platinum members more money based on their credit score
65 if (currentCustomer != null && currentCustomer.getLevel() == "Platinum"){
66     long coeff = MAX_SCORE - currentCustomer.getCreditScore()/MAX_SCORE;
67     double adjustment = this.currentApplication.getAmount()/coeff;
68     adjustedAmount += adjustment;
69
70
71 // Update Status
72 updateApplicationStatus(approve);
73
74 message = "Customer ID:" + customerid + " FICO Score: " + score + " Approved: " + approve
75 }
76
77 response.setContentType("text/html");
78 response.setHeader("Access-Control-Allow-Origin", "*");
79 PrintWriter out = response.getWriter();
80 log.info("Credit Check: " + message);
81
82 out.println(message);
83 out.flush();
84 } catch (Exception e) {
85     log.error(e.getMessage());
86     StringWriter writer = new StringWriter();
87     PrintWriter pw = new PrintWriter(writer);
88     e.printStackTrace(pw);
89     log.error(writer.toString());

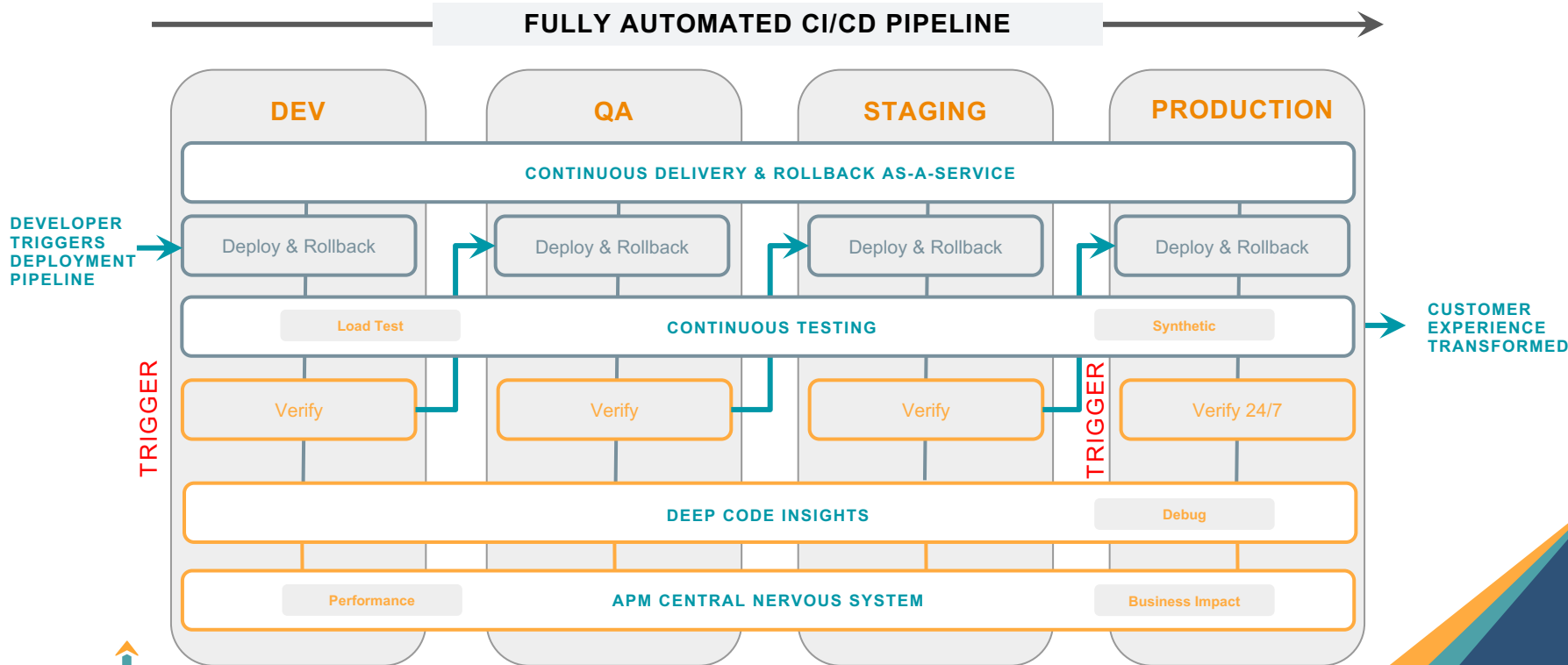
```

Timestamp	Breakpoint	Hostname	Data
03:34:14:291 PM	⏏ CreditCheck.java : 67	processor	CreditCheck.java@67 was hit on processor
03:34:21:314 PM	⏏ CreditCheck.java : 67	processor	CreditCheck.java@67 was hit on processor
03:34:32:829 PM	⏏ CreditCheck.java : 67	processor	CreditCheck.java@67 was hit on processor
03:34:38:762 PM	⏏ CreditCheck.java : 67	processor	CreditCheck.java@67 was hit on processor

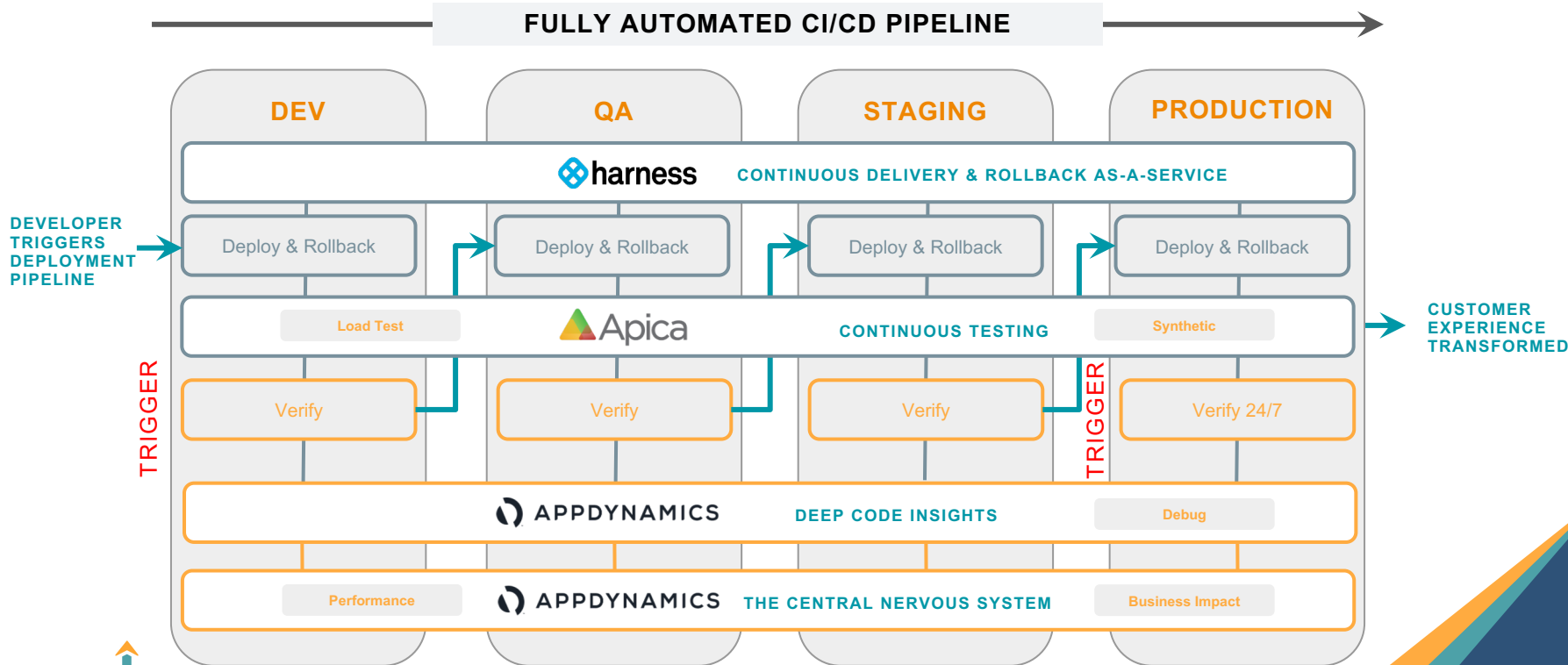


## Lesson Five

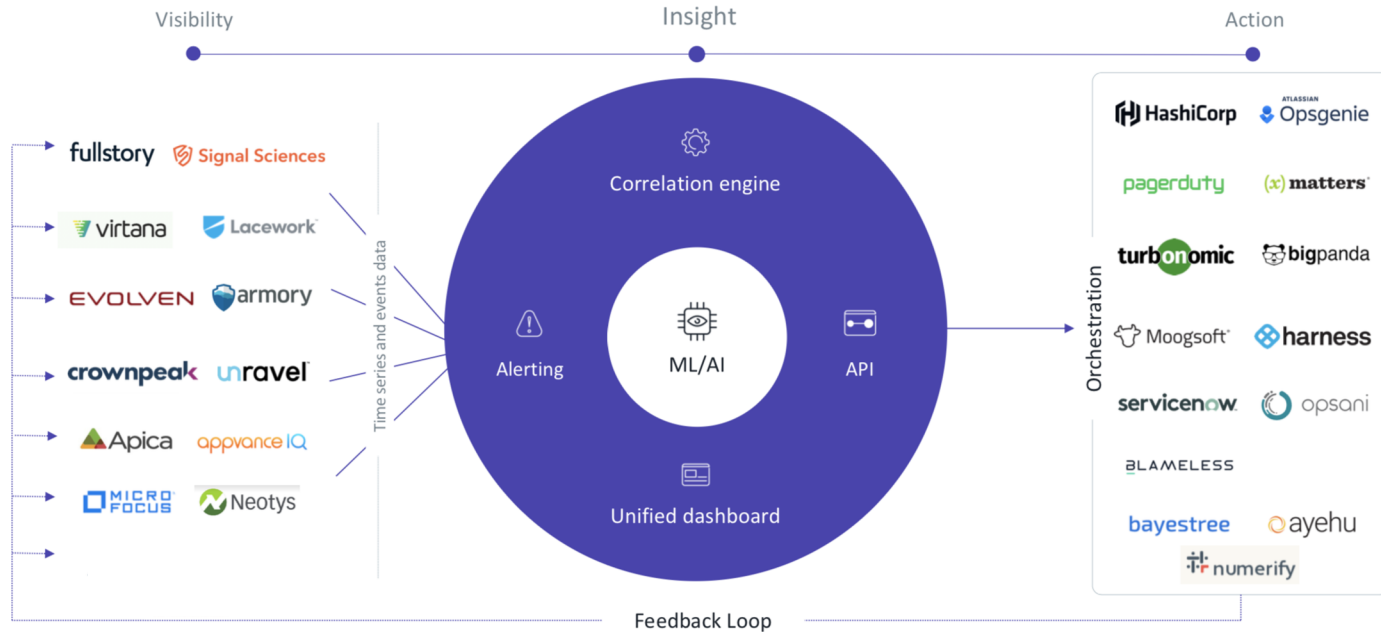
# Continuous Delivery & Verification



# Continuous Delivery & Verification



# Integrated with AppDynamics



“experience with performance tuning and monitoring has moved ... to a 39% must-have skill ranking.”

Source: DevOps Institute | 2020 Upskilling: Enterprise DevOps Skills Report

## Lessons

- APM first gives autodiscovery & dynamic baselines
- Automate deploys & rollbacks
- Schedule releases on engineering AND business needs
- APM & Test tool integration = relaxed releases
- Sometimes, it can only be diagnosed live



# Links



- Join a session specific ongoing conversation at [https://eurl.io/#B4zglZQ\\_E](https://eurl.io/#B4zglZQ_E)  
(WebEx Teams - will ask for your email)
- Continuous Delivery <https://www.appdynamics.com/solutions/continuous-delivery>
- Central Nervous System <https://www.appdynamics.com/central-nervous-system/>
- AppDynamics & Harness - [https://bit.ly/Appd\\_Harness](https://bit.ly/Appd_Harness)
- AppDynamics & Apica - [https://bit.ly/AppD\\_Apica](https://bit.ly/AppD_Apica)
- Accelerating Digital Transformation and Application Performance In Turbulent Times - <https://bit.ly/accelturbulent>
- Successfully Deploying AIOps - <https://bit.ly/deployaiops>
- My chosen charity for your consideration: Doctors Without Borders [https://bit.ly/msf\\_dwb](https://bit.ly/msf_dwb)

# THANK YOU!

Meet me in the Network  
Chat Lounge for questions

