

# DevOps, microservices and stress-free incidents: How to have your cake and eat it too

Peter Holditch  
Principal PreSales Engineer

 [pholditch@appdynamics.com](mailto:pholditch@appdynamics.com)

 [@pholditch](https://twitter.com/pholditch)

# Have your agenda and eat it too

- How did we get here?
  - Why DevOps
  - Why Microservices
- Where DevOps and microservices collide
- Joint DevOps & microservices management requirements
- Delivering these requirements: buy vs build
- Business case for buying a solution

# Why DevOps?

- The rate of change of systems has grown exponentially
  - 2 releases a year, delivered complete with a big operations manual is not a recipe for high velocity delivery
- Keeping systems reliable in the face of this rate of change requires
  - Automation of the software release (manufacturing) process
  - Collaboration across the software lifecycle
    - “traditional ops” run platforms, delivery pipelines
    - Developers (formally) take responsibility for application troubleshooting in production

# Why microservices?

- Loose coupling enables agility
  - Choose best language for the job by service
  - Different rates of change per service
  - Different NFRs per service
  - Easier to coordinate activity within smaller development teams
- Focused domain expertise
- Better accountability to business stakeholders

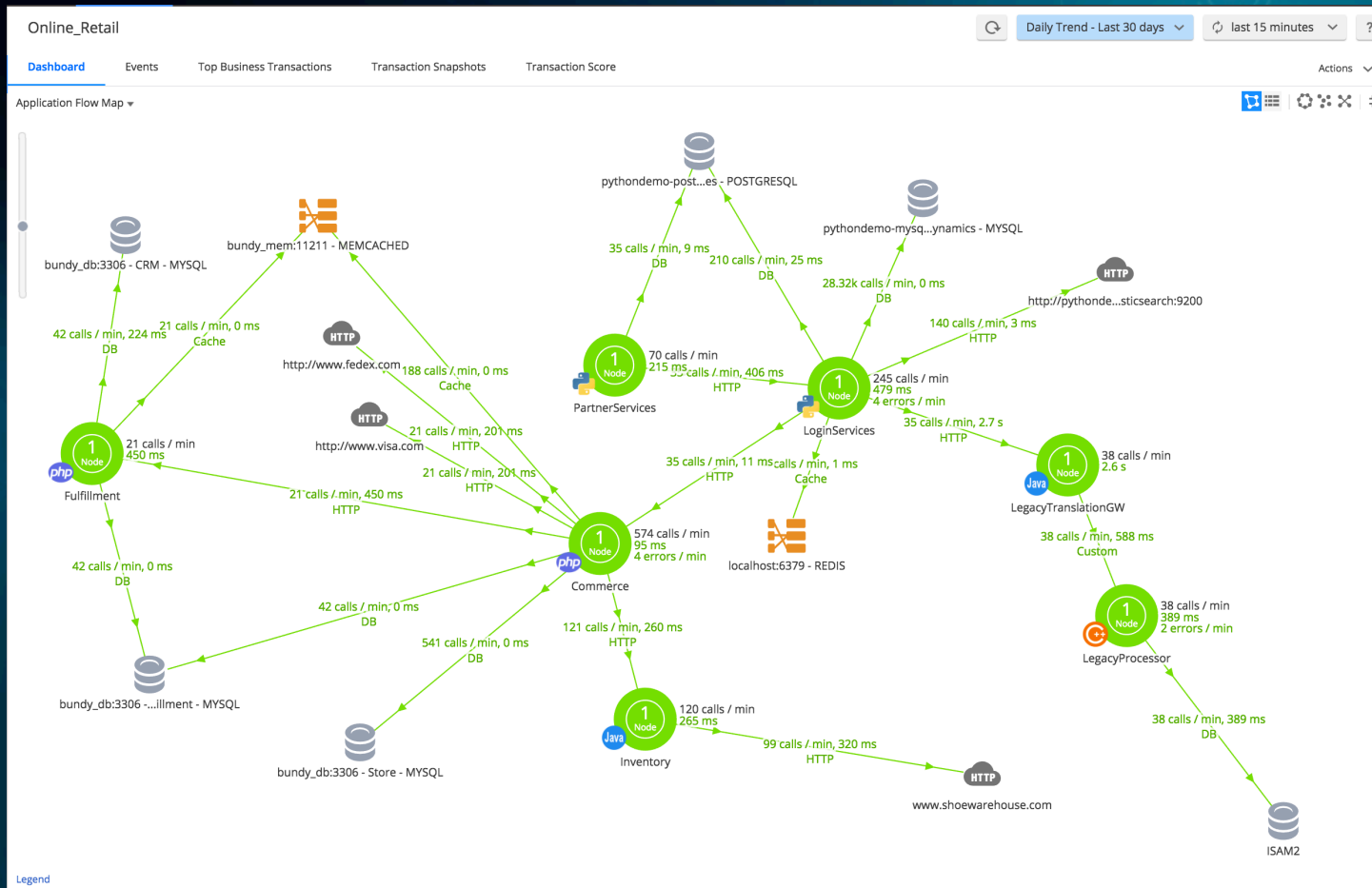
# Where DevOps and microservices collide

- Trade off dev time simplicity vs operational complexity
- Troubleshooting : how many technology “throats to choke”?
  - Triage vs analysis
  - Is each piece managed in the same way?
  - Dev time from each team to support => the dreaded war room
- Maturity: once a service is stable and the team move on...
  - Who owns the service now?
  - Who can troubleshoot it?

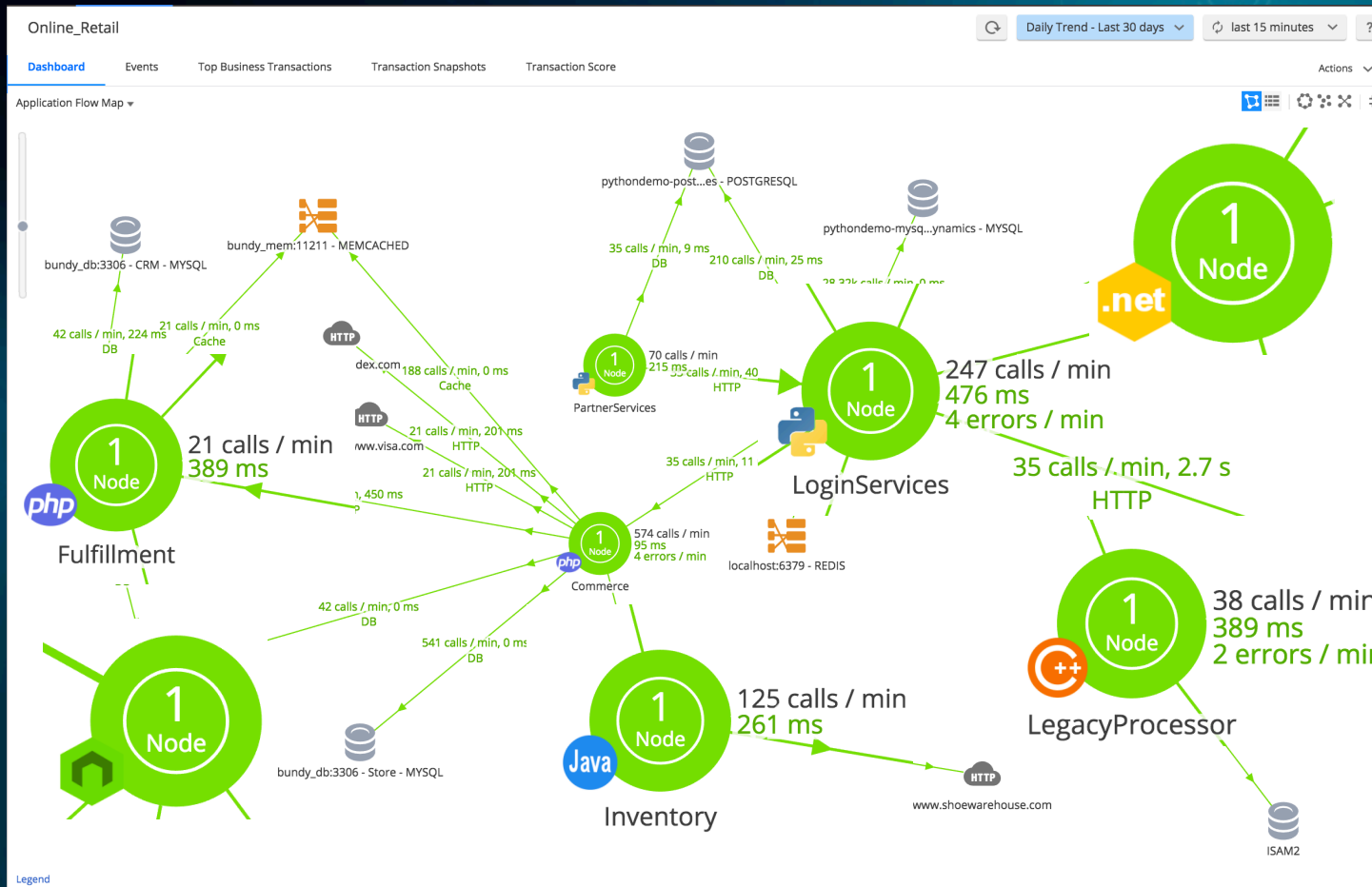
# Requirements for unified monitoring of microservices

- For developers
  - Language support
  - Low touch
    - In implementation and production handover
    - Provide detailed code level visibility when necessary
- For Ops
  - Ease of use; simplify troubleshooting (“shift left”)
    - Consistency
    - Comprehensive end to end visibility, fault domain isolation
- For DevOps
  - Foster dev/ops collaboration
  - Provide feedback throughout delivery pipeline

# Automatically trace application traffic end to end

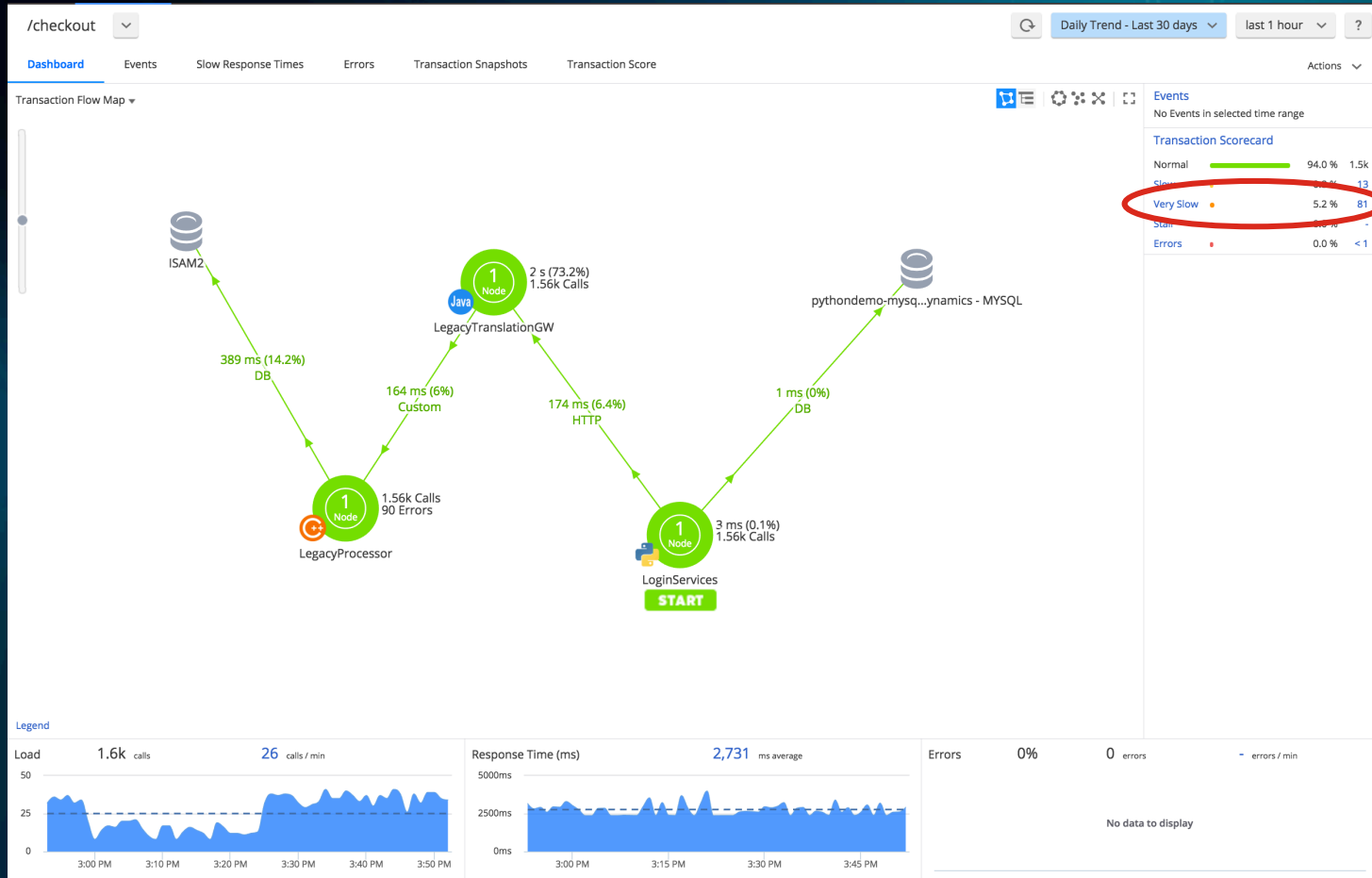


# Support polyglot applications

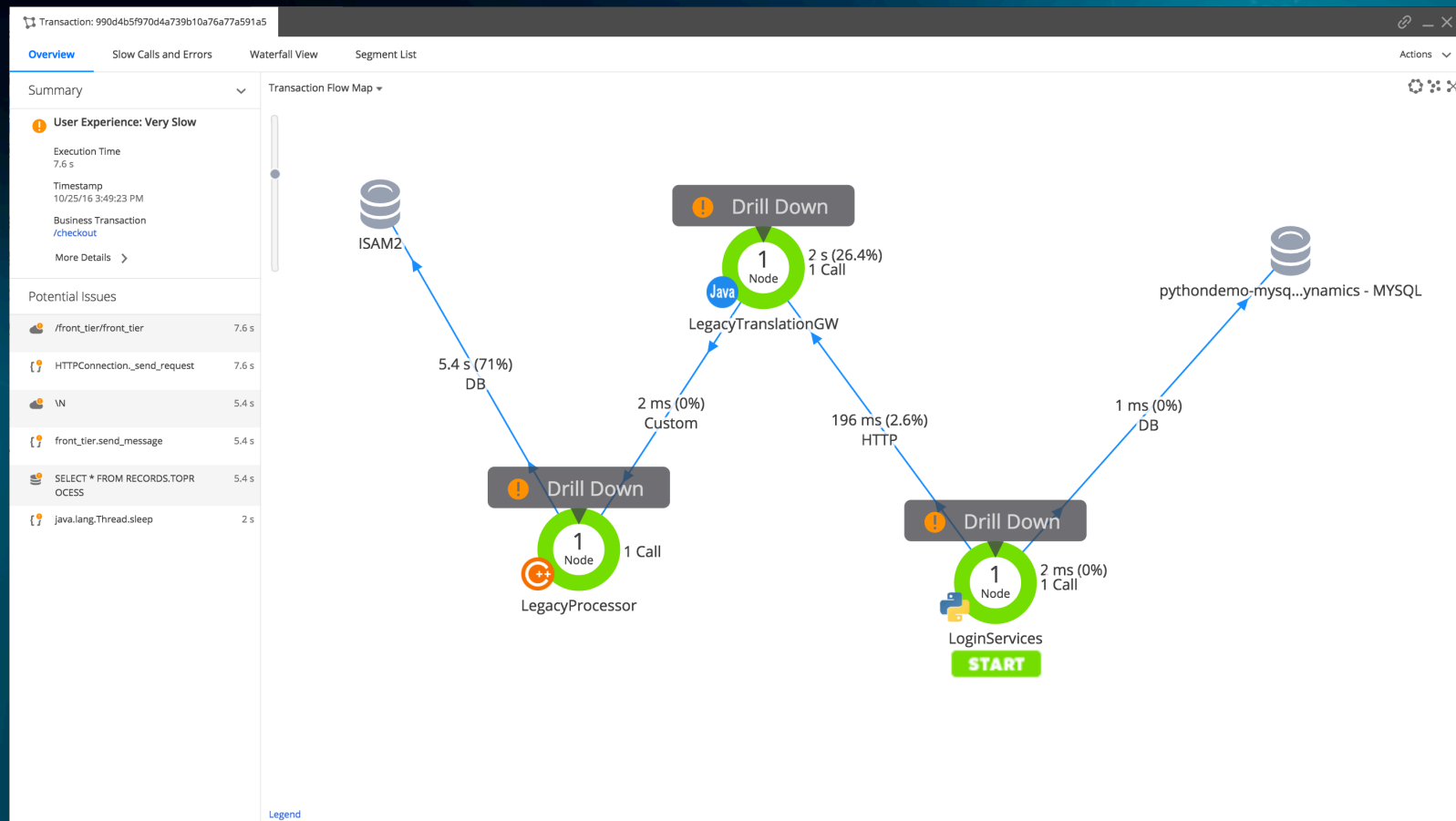




# Identify flow per transaction



# Troubleshoot individual transactions...



# ...down to code level

Transaction: 990d4b5f970d4a739b10a76a77a591a5 | 7,582 ms - LoginServices

Overview | **Call Graph** | Slow Calls & Errors | DB & Remote Service Calls | Server | Data Collectors | More

Execution Time: 7.6 s | Node: Login-Node | Timestamp: 10/25/16 3:49:23 PM

Details | Filters | Set Root | Reset Root | Export

Name	Time (ms)	Percent %	Exit Calls / Threads *
(request) -	3 ms (self)	0%	DB
Thread::_bootstrap - python2.7/threading.py:769	0 ms (self)	0%	
Thread::_bootstrap_inner - python2.7/threading.py:792	0 ms (self)	0%	
Thread::run - python2.7/threading.py:752	0 ms (self)	0%	
ThreadedWSGIServer::process_request_thread - python2.7/SocketServer.py:586	0 ms (self)	0%	
ThreadedWSGIServer::finish_request - python2.7/SocketServer.py:332	0 ms (self)	0%	
WSGIRequestHandler::__init__ - python2.7/SocketServer.py:643	0 ms (self)	0%	
WSGIRequestHandler:handle - werkzeug/serving.py:212	0 ms (self)	0%	
WSGIRequestHandler:handle - python2.7/BaseHTTPServer.py:336	0 ms (self)	0%	
WSGIRequestHandler:handle_one_request - werkzeug/serving.py:245	0 ms (self)	0%	
WSGIRequestHandler:run_wsgi - werkzeug/serving.py:133	0 ms (self)	0%	
<b>execute - werkzeug/serving.py:180</b>	<b>7,579 ms (to...)</b>	<b>100%</b>	<b>HTTP</b>

**execute:180**

Name: execute - werkzeug/serving.py  
Type: PY  
Class: Class  
Method: execute  
Line Number: 180

Self Time: 0ms (0%)  
Total Time: 7,579ms (100%)

Type	Details	Count	Time (ms)	% Time	From	To
HTTP	/front_tier/front_tier	1	7579	100%	LoginServic...	LegacyTran...

**HTTP Exit Call** | Drill Down into Downstream Call

Time: 7,579ms  
From: LoginServices  
To: LegacyTranslationGW  
Details: /front\_tier/front\_tier

**Properties**

HOST: 52.39.227.146  
PORT: 8080  
CorrelationHeader: appId=40\*acctguid=9cd60207-9acc-4877-bc95-ac6f95c1f115\*ctrlguid=1471639717\*btid=470\*snapenable=true\*guid=9029ea24faa5466c847b7dc58d3894e\*unresolvedexitid=619\*exitguid=2\*cidfrom=144\*cidto=171\*esypeorde=HTTP\*esubtype=HTTP

# I can build that myself with ELK / zipkin / ...

- Sure, it's only software... Let's get on with it!
  - Stand up Elastic Search and suck in your logs. Job done?
  - Now, standardise your log messages so they're consistent
  - Now, flow a transaction identifier end to end and add it to the messages, or use zipkin to trace transactions
- That's several man months of work already...
  - ... and you still have no code level visibility...
  - ... nor any easy to use UI to allow that essential "shift left"
- Now imagine you acquire a product or a company 😞

## Bonus: architectural benefits

- Architectural governance
  - Per service dependency analysis
  - Manage Conway's law effects
- Evolve and manage risk in the system over time
- Data for capacity planning
- Achieve maturity with less risk of “hitting a wall”
- Evolution toward #noops?

# The business case for unified monitoring

- Save developer time in production troubleshooting typically between 7-20% of developer time
- Save developer time in performance testing
- Save developer time implementing custom monitoring
- Save developer time in production handoff
- Ease staff on-boarding

Free 1 dev  
Free .5 dev  
Free .75 dev  
Free .25 dev

“1 month to 1 day”  UBS

- [SaaS] avoid running costs of monitoring back-ends

# Summary

- Microservices shift complexity from development to production
- DevOps is not a panacea for production complexity: developers have day jobs!
- Production complexity needs to be managed to allow efficient (stress-free) operation
- AppDynamics offers an excellent, cost-justifiable, solution to these issues

## Questions?

# Register for a free trial!

[www.appdynamics.com/free-trial](http://www.appdynamics.com/free-trial)

The screenshot shows the AppDynamics website landing page. At the top left is the 'APPDYNAMICS' logo, and at the top right is an 'Account' dropdown menu. The main heading is 'Get started quickly with AppDynamics'. Below this, a sub-heading reads 'Start your free 15-day trial now, easily, with no credit card required.' To the right of this text is a dark grey box containing the text 'Sign Up for Free' and a prominent blue button labeled 'CREATE A FREE ACCOUNT'. Below the main heading, a paragraph states: 'Designed for production and pre-production environments, AppDynamics gives you visibility into your entire application topology from a single pane of glass. Instrument your own application or use our sample app.' The page features four feature cards, each with an icon and a title: 1. 'Application Performance Management' with an 'APP' icon, describing monitoring and managing end-to-end performance of complex distributed applications. 2. 'End User Monitoring' with a mobile phone icon, describing delivering a better user experience with Mobile and Browser Real-User Monitoring and Browser Synthetic Monitoring. 3. 'Infrastructure Visibility' with a server rack icon, describing unlocking deeper insights by correlating server and database performance with application performance. 4. 'Real-time Business Monitoring' with a dashboard icon, describing getting real-time business awareness into IT operations, customer experience, and business outcomes with Transaction, Log, Browser, and Mobile Analytics. The footer contains copyright information: 'Copyright © 2009 - 2017 AppDynamics. All Rights Reserved.', a language selector for 'English (USA)', and links for 'Privacy Policy', 'Cookie Policy', 'Terms of Use', and 'Security'. Social media icons for Facebook, Twitter, LinkedIn, and YouTube are also present.

APPDYNAMICS Account ▾

## Get started quickly with AppDynamics

Start your free 15-day trial now, easily, with no credit card required.

Sign Up for Free

[CREATE A FREE ACCOUNT](#)

Designed for production and pre-production environments, AppDynamics gives you visibility into your entire application topology from a single pane of glass. Instrument your own application or use our sample app.

- Application Performance Management**  
Monitor and manage end-to-end performance of complex distributed applications.
- End User Monitoring**  
Deliver a better user experience with Mobile and Browser Real-User Monitoring and Browser Synthetic Monitoring.
- Infrastructure Visibility**  
Unlock deeper insights by correlating server and database performance with application performance.
- Real-time Business Monitoring**  
Get real-time business awareness into IT operations, customer experience, and business outcomes with Transaction, Log, Browser, and Mobile Analytics.

Copyright © 2009 - 2017 AppDynamics. All Rights Reserved. English (USA) Privacy Policy Cookie Policy Terms of Use Security

f t in