Testing in Production

Aja Hammerly
Aja Hammerly
http://github.com/thagomizer
@the_thagomizer
http://www.thagomizer.com
Google 🖤 Ruby
Victory Conditions
Comfort
Already Doing It
Intentionality
Definitions
Testing
Verifying
Traditional Tests
Blackbox Tests
Real Env $\Rightarrow$ Real Bugs
Real Users ⇒ Real Load
Integrations
Seams
HesienBugs
Is It New?
GET OFF MY LAWN
How
Deployment Testing

User Focused Testing

Reusing Tests

Implicit Testing
Deployment Techniques
Canary
Phased Rollout
\subset Users or \subset Servers
Monitor Vigorously
If 👍

Expand Canary Group
Choosing Canary Group
Internal Users
Random
Geographic
% of Servers
Demographic
Users That Sign Up
Pick As Many As You Like
Blue/Green
Live

Idle

Deploy New Code Here
Idle
Old Code

Live
New Code
Easy Rollback
Disaster Recovery
Database ⇒ Pain
Snapshots & Replication
Non-Relational
Variation
Auto Rollback
Scripted
Data[base] Migrations
Session Affinity
Separate Data Migrations
User Focused Tests
A/B Testing
Basic Experimental Design
Both Live at Once
Betas & EAPs
EAP = Early Access Program
Stability & Usability
Give Users Time
Make Expectations Clear
Reusing Tests
Smoke Tests Against Prod
Smoke Test
Subset of Existing Tests
Scheduled Runs
3rd Party Integrations
Leave No Trace
Controlled Breakage
Response & Recovery
Stay in Scope
Simian Army
DiRT
Penetration Testing
Ethical Breakage
Disaster Recovery Verification
For Real
Move Traffic
Move Data Centers
Restore Database
Testing Scripts
Testing People
assert people.detect_issues
Implicit Testing
Monitoring
Alerts?
Alerts
Tell Us Something Is Wrong
Tell Us The System Isn’t Meeting Expectations
Expectations?
Testing
Alert if latency > 500 ms
assert latency < 500 ms
Proactive
assert 5xx/4xx < 0.05
assert (Disk % Full) < 80%
MoM / YoY
Overview

Insights
No insights to report for the past 7 days.

Recent traces

<table>
<thead>
<tr>
<th>LATENCY</th>
<th>URI</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>56 ms</td>
<td>/material/edit</td>
<td>9...</td>
</tr>
<tr>
<td>64 ms</td>
<td>/api/v1/item/...</td>
<td>10...</td>
</tr>
<tr>
<td>11 ms</td>
<td>/api/v1/user</td>
<td>3...</td>
</tr>
<tr>
<td>61 ms</td>
<td>/api/v1/event/...</td>
<td>10...</td>
</tr>
<tr>
<td>95 ms</td>
<td>/api/v1/item/...</td>
<td>6...</td>
</tr>
<tr>
<td>9 ms</td>
<td>/api/v1/event/...</td>
<td>10...</td>
</tr>
</tbody>
</table>

See more ...

Most frequent URIs

<table>
<thead>
<tr>
<th>LATENCY</th>
<th>URI</th>
<th>REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>59 ms</td>
<td>/api/v1/item/...</td>
<td>v...</td>
</tr>
<tr>
<td>596 ms</td>
<td>/api/v1/item/...</td>
<td></td>
</tr>
<tr>
<td>524 ms</td>
<td>/api/v1/item/...</td>
<td></td>
</tr>
<tr>
<td>531 ms</td>
<td>/api/v1/item/...</td>
<td></td>
</tr>
<tr>
<td>550 ms</td>
<td>/api/v1/item/...</td>
<td></td>
</tr>
</tbody>
</table>

Daily analysis reports

/api/v1/item/tag/title/podcast
Overall latency for requests that make remote procedure calls

% of total requests

Latency in ms

Start 2018-04-10 (20:00:00)  End 2018-04-11 (20:00:00)
Start 2018-04-17 (20:00:00)  End 2018-04-18 (20:00:00)
assert_equal error_rate_old, error_rate_new
assert error_rate_new < error_rate_old
Dos and Don’ts
Do Have Clear Goals
Don’t DDOS Yourself
Do Test Your Seams
Don’t Mess With User Data
Do Clean Up After Yourself
Make Alerts Actionable
Do Verify Your Integrations
Do Act Methodically
Have Clear Goals
Test Your Seams
Verify Your Integrations
Clean Up After Yourself
Don’t DDOS Yourself
Leave User Data Alone
Keep Alerts Actionable
Thank You
Get Off My Lawn
Questions?