Scaling DevOps Deployments with AWS CodePipeline and Dynatrace AppMon

Followed by Live Q&A

- Part of Dynatrace Performance Clinic Series
- Get Dynatrace AppMon: http://bit.ly/dtpersonal
- Ask questions using the ***Questions Tab*** – or post on ***answers.dynatrace.com***

Rob Jahn

Sr. Consultant @ CGI

@RobJahn45

Joe Sicree

Sr. Consultant @ CGI

@JoeSicree

Daniel Freij

Sol. Engineer @ Apica

@DanielFreij

Andi Grabner

DevOps @ Dynatrace

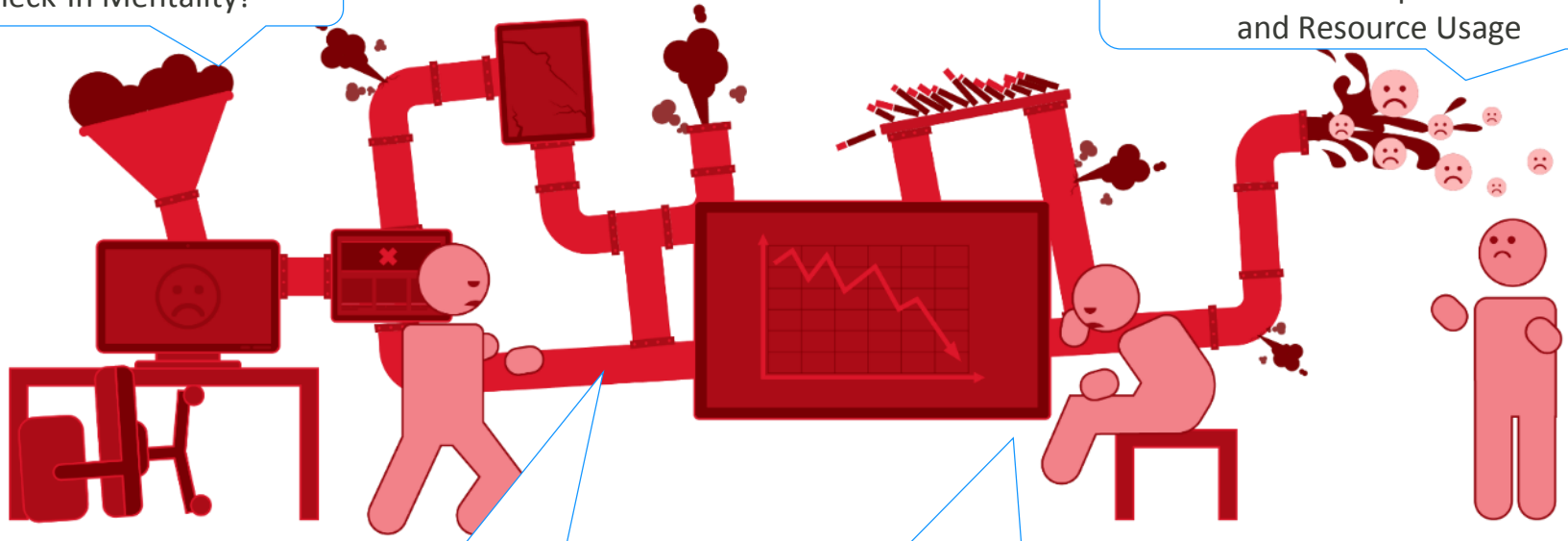
@grabnerandi

dynatrace

AWS does NOT PREVENT your teams from pushing *bad code* through the pipeline *more frequently*!

Dev&Test: Trial & Error Check-In Mentality!

Ops/Biz: Bad Code Changes make it to Production. Impact End Users and Resource Usage

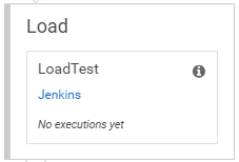


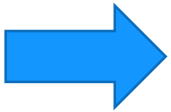
Test / CI: Basic Tests that only test high-level functionality!

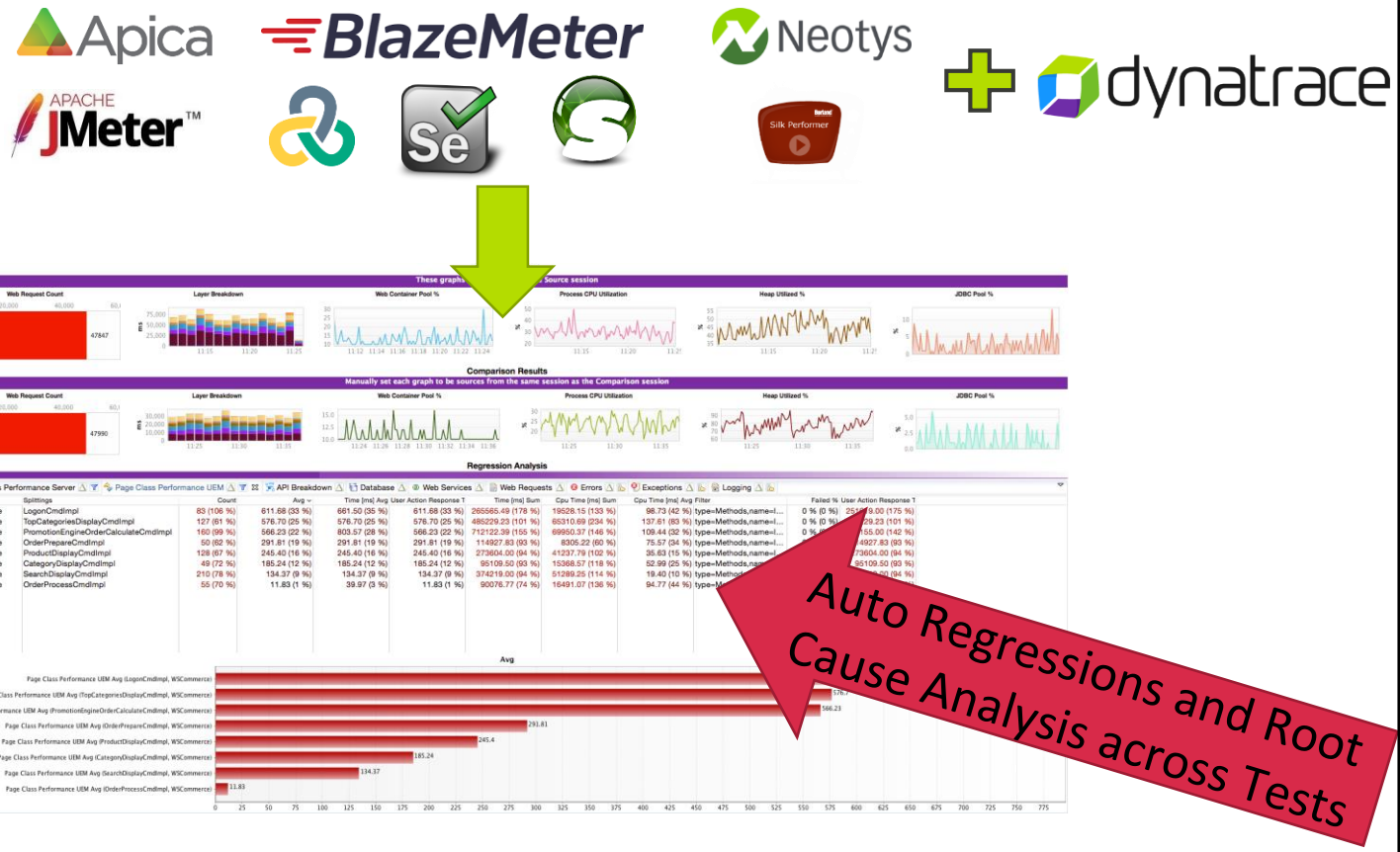
Performance: Too many code commits leads to skipping performance testing to save time

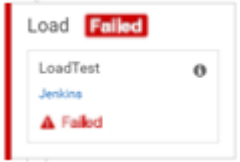
dynatrace

Dynatrace: Automatic Performance Analysis across Load Tests







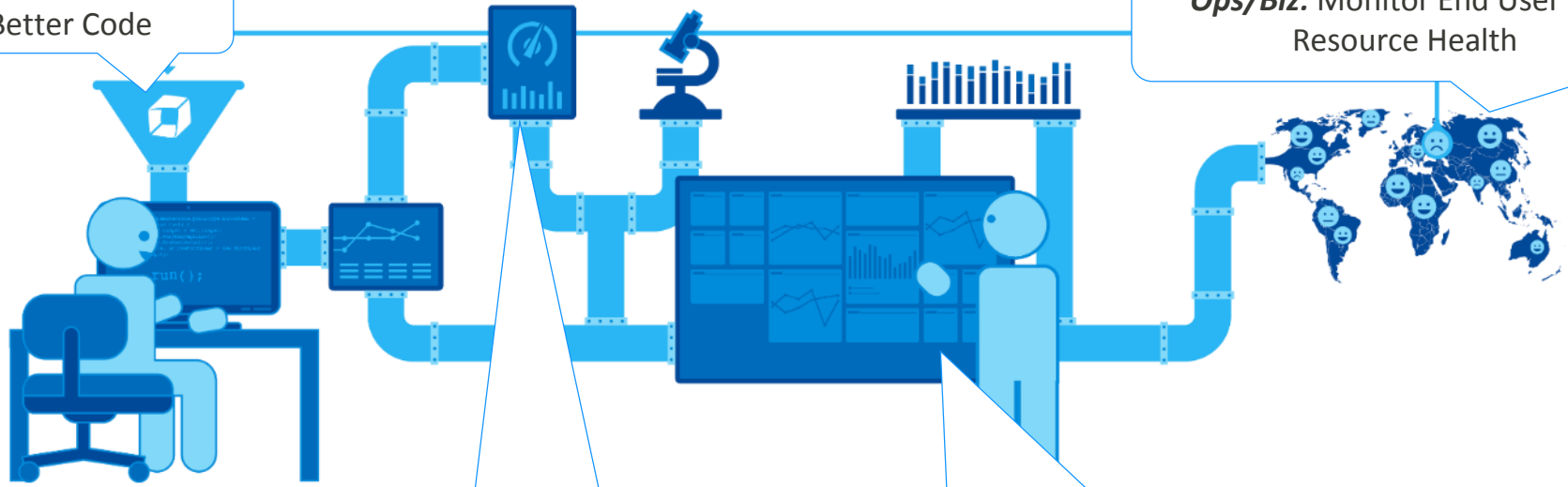


Scaling Continuouos Innovation with Dynatrace on AWS:

Shift-Left Quality, Reduce Lead Time & Increase Flow

Dev&Test: Check-In Better Code

Ops/Biz: Monitor End User and Resource Health

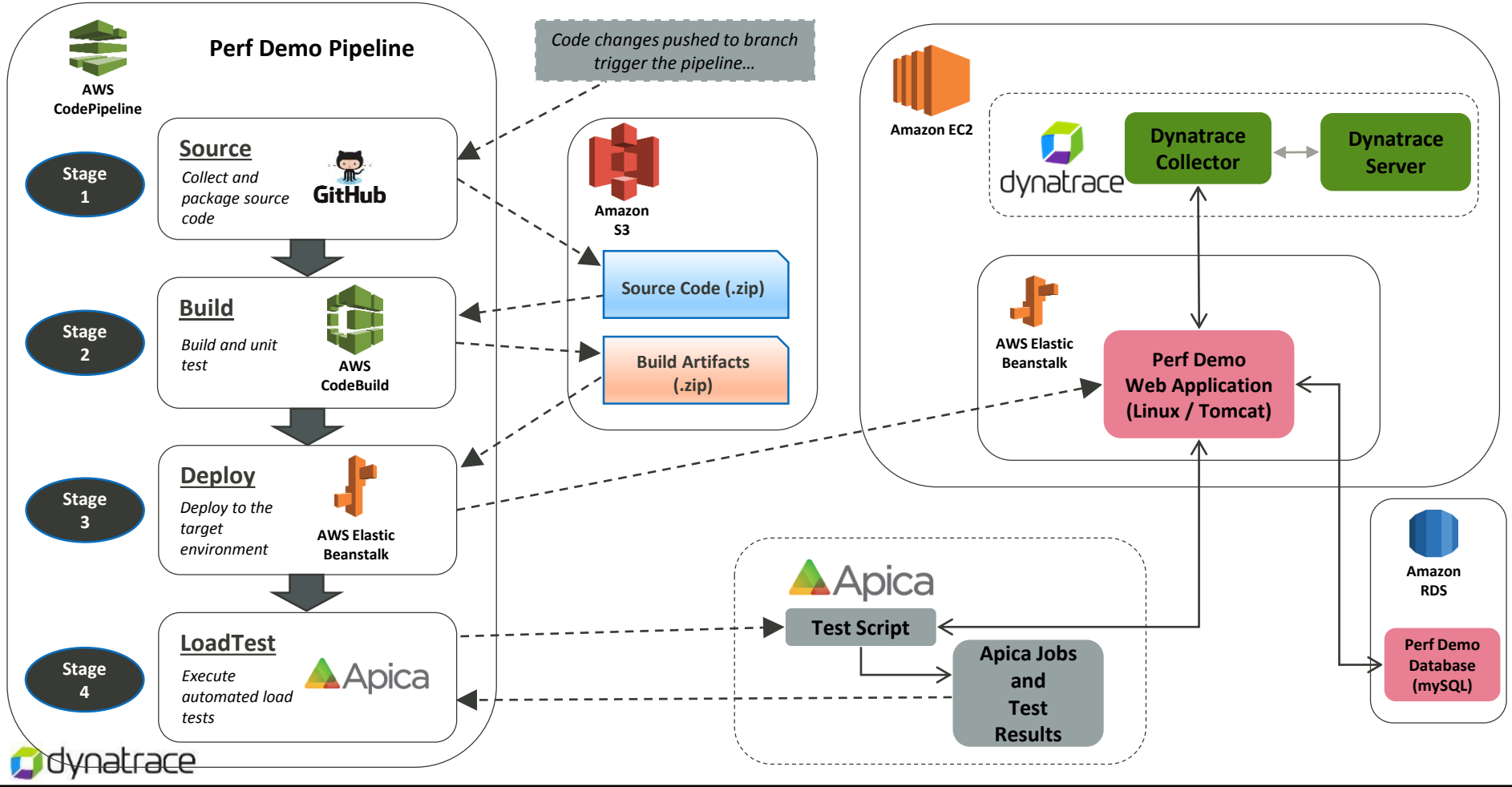


Test / CI: Stop 80% of Bad Builds Early

Performance: Production Ready Checks on Good Builds Only!

dynatrace

Today's Goal: Full AWS CodePipeline Demo!



Agenda

- Introduction of Speakers
- What is AWS CodePipeline?
- How does Apica LoadTest and Dynatrace AppMon fit in?
- Live Demo: 3 Performance Engineering Use Cases
- Live Q&A

Apica

”Helping companies with their end-user experience focusing on availability and performance”

Apica Synthetic Monitoring

Apica LoadTest

SaaS

Hybrid

On-Premise

Global Offices
Stockholm
London
New York
Santa Monica

Apica

Founded in 2005

500+ Customers

technology fast 50™ companies-to-watch | Deloitte.

dynatrace

CGI is a global information technology and business process services leader

World's 5th largest independent IT and BPS firm

High-end business and IT consulting

Client proximity model complemented by unique global delivery network

Serving **5000+ clients** from over **400 offices** around the world

End-to-end IT and business process services

100+ leading IP-based solutions

71,000 professionals, **40** offices

Focused industry and domain **expertise**

36 years of successfully partnering with our clients

Revenue:
C$10.4B *
Backlog:
C$17.7B

dynatrace

CGI Performance Engineering Services

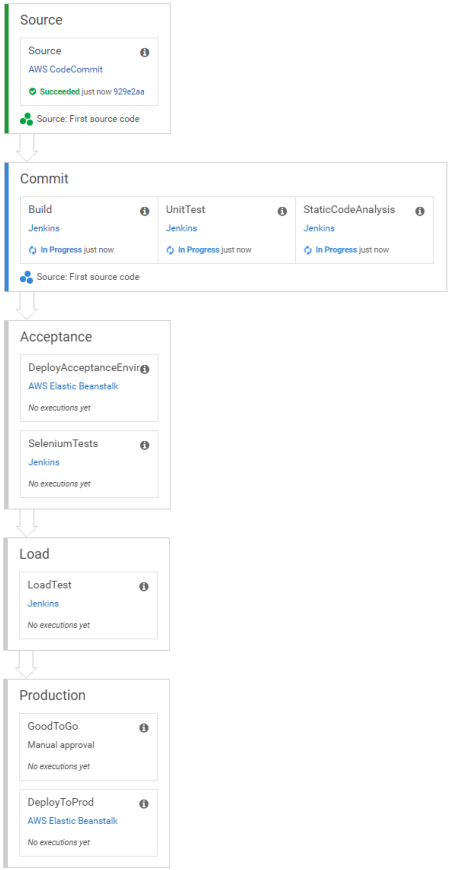
Software Performance Engineering

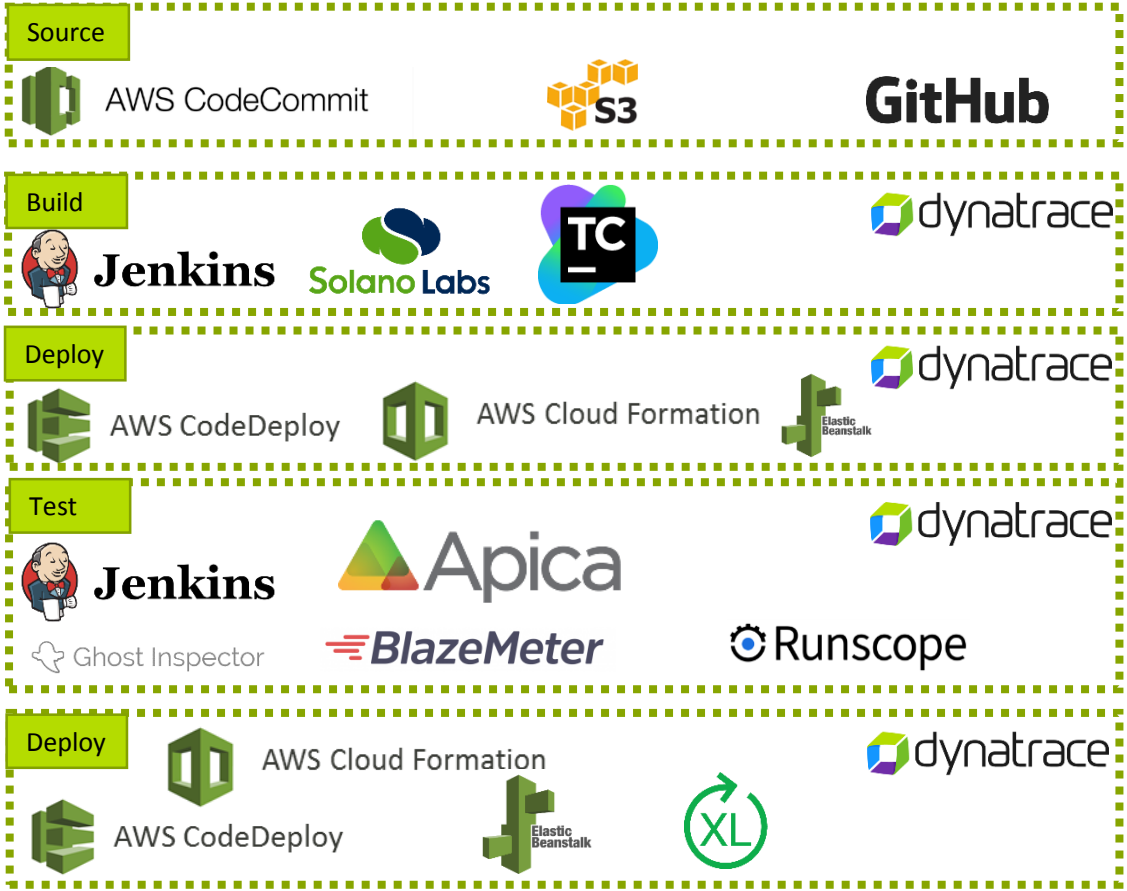
CGI's performance engineering services drive improvements in information technology and ensure scalability and stability for existing and planned systems, including improving overall quality, reliability, capacity, security, and performance.

Advisory Services	• Is my current team delivering the right services across the SDLC? • Organizational and operational planning, strategy and design services for transforming internal capabilities. • Establishing PE Center of Excellence (COE).Deliver for the enterprise or for a critical program.
Application Readiness Assessment and Performance Testing	• Is my application ready for increased volumes? • Technically assess the application to understand the risks to performance, scalability and availability. • Combination of architecture and code review to identify performance issues and risks along with comprehensive performance testing using CGI's proven PT methodology. End result is identification of issues and their remediation
Application Performance Management	• Do you know your end user's experience? • End to end proactive monitoring of critical business processes. Define and implement an APM strategy. • Identify key operation use cases including key business transactions and business operational metrics. • Build process for measurement, trending, and capacity management. • Establish best practices via an APM reference architecture.
Production Performance Rescue	• Do you have a critical production issue resulting in repeated downtime and lost business? • A proven approach to production performance and stability issues, reviewing the end to end transaction. Minimizing disruption and quickly restoring functionality and capability.

dynatrace

What is AWS CodePipeline? How does Apica Load and Dynatrace fit in?

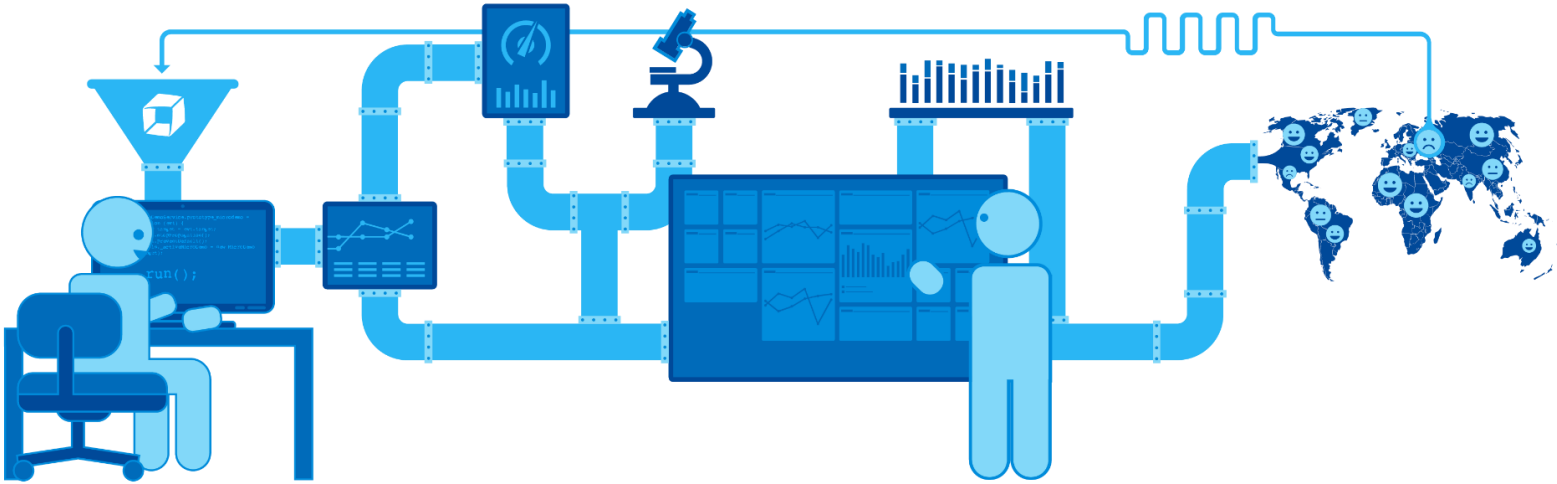

Source
Source
AWS CodeCommit
Succeeded just now 929e2aa
Source: First source code
Commit
Build
Jenkins
In Progress just now
UnitTest
Jenkins
In Progress just now
StaticCodeAnalysis
Jenkins
In Progress just now
Source: First source code
Acceptance
DeployAcceptanceEnvir
AWS Elastic Beanstalk
No executions yet
SeleniumTests
Jenkins
No executions yet
Load
LoadTest
Jenkins
No executions yet
Production
GoodToGo
Manual approval
No executions yet
DeployToProd
AWS Elastic Beanstalk
No executions yet


Source
AWS CodeCommit
S3
GitHub
Build
Jenkins
Solano Labs
TC
dynatrace
Deploy
AWS CodeDeploy
AWS Cloud Formation
Elastic Beanstalk
dynatrace
Test
Jenkins
Apica
dynatrace
Ghost Inspector
BlazeMeter
Runscope
Deploy
AWS Cloud Formation
AWS CodeDeploy
Elastic Beanstalk
XL
dynatrace

dynatrace

Make yourself familiar with …

- AWS CodePipeline: https://aws.amazon.com/codepipeline
- Apica SaaS and Zebra Tester: https://www.apicasystem.com/load-testing/
- Dynatrace AppMon & UEM: http://bit.ly/dttutorials



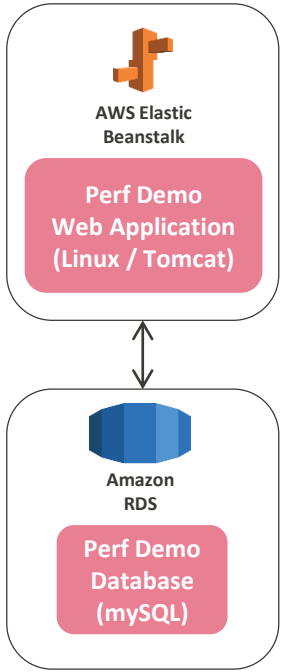
DEMO TIME!

Demo Overview

- The following demo will walk us through a simple pipeline for a web application called Perf-Demo.
- The demo will focus on the following use cases:
  - Missed Service Level Agreements (SLA) for a mix of transactions
  - Memory leaks
  - Poor performance when accessing a database
- The demo will use a simple web application to demonstrate these use cases.

Demo Application

- Java Spring web application that provides a number of REST web services for use in performance testing.
- The web application provides the following services:
  - **calculateCompoundInterest** – Calculates compound interest for one or more accounts over a specified time period.
  - **addDataToMap** – Populates a hash map with data using a simple key
  - **getFinancialRecordsByDate** – Fetches financial data for one or more stocks on a specified date.



Get the code at https://github.com/jsicree/perf-demo

Demo AWS CodePipeline

AWS CodePipeline

Perf Demo Pipeline

Code changes pushed to branch trigger the pipeline…

Stage 1

Source

Collect and package source code

GitHub

Stage 2

Build

Build and unit test

AWS CodeBuild

Stage 3

Deploy

Deploy to the target environment

AWS Elastic Beanstalk

Stage 4

LoadTest

Execute automated load tests

Apica

Amazon S3

Source Code (.zip)

Build Artifacts (.zip)

Amazon EC2

dynatrace

Dynatrace Collector

Dynatrace Server

AWS Elastic Beanstalk

Perf Demo Web Application (Linux / Tomcat)

Amazon RDS

Perf Demo Database (mySQL)

Apica

Test Script

Apica Jobs and Test Results

dynatrace

Demo #1 – Slow Services

The Service

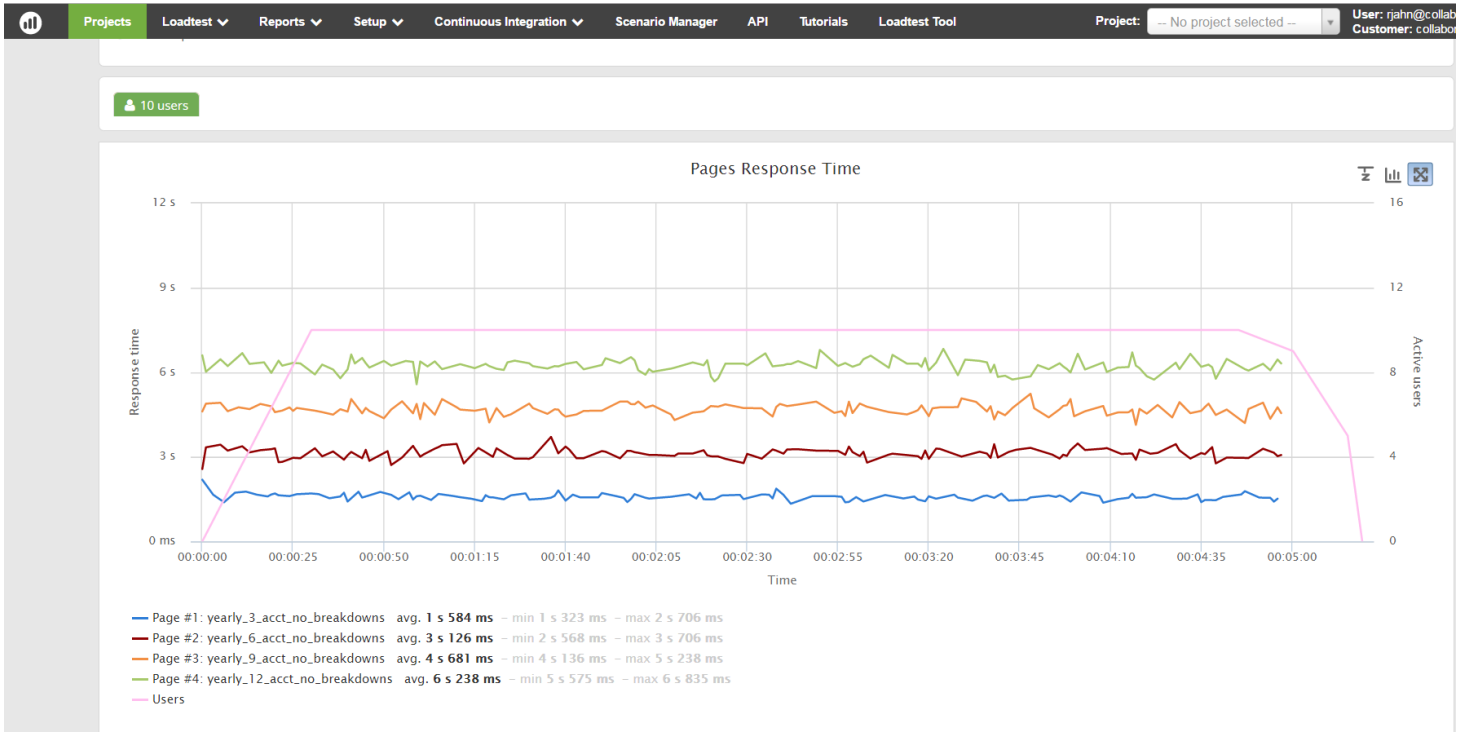
The computeCompoundInterest service computes the compound interest for a collection of accounts for a requested number of monthly or yearly intervals.

The Problem

The initial load test for the service using a mix of 3, 6, 9 and 12 accounts showed that the average response time was above the expected SLA for the requests with more accounts.

Demo #1 – Slow Services (V1 Test)

Exceeding SLA for Response Time



dynatrace

Demo #1 – Slow Services (cont.)

The Analysis

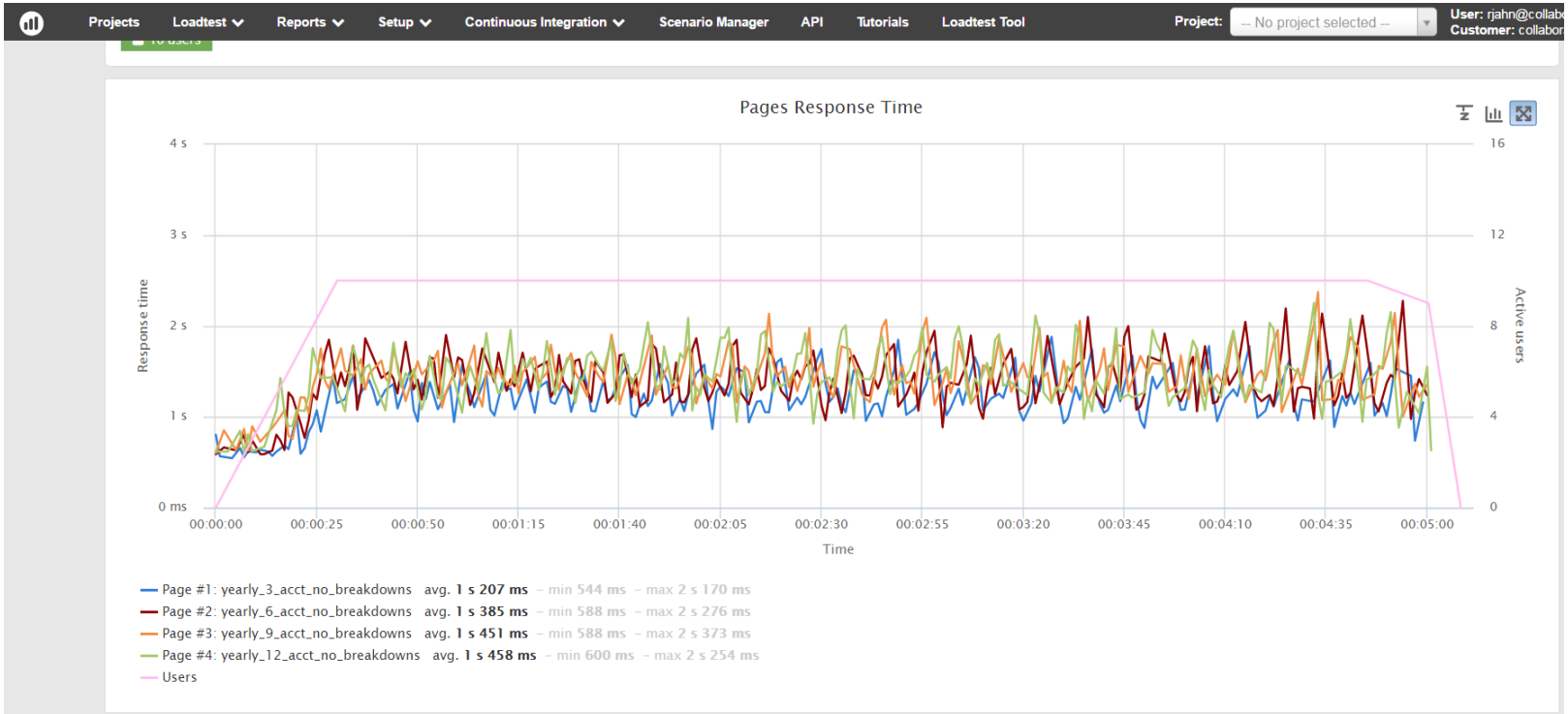
The underlying service was implemented synchronously. The more accounts included in the request, the longer the web service call took.

The Solution

The service was refactored to run asynchronously using threads.

Demo #1 – Slow Services (v2 Test)

All calls are under the 3 second SLA



dynatrace

Demo #2 – Memory Leak

The Service

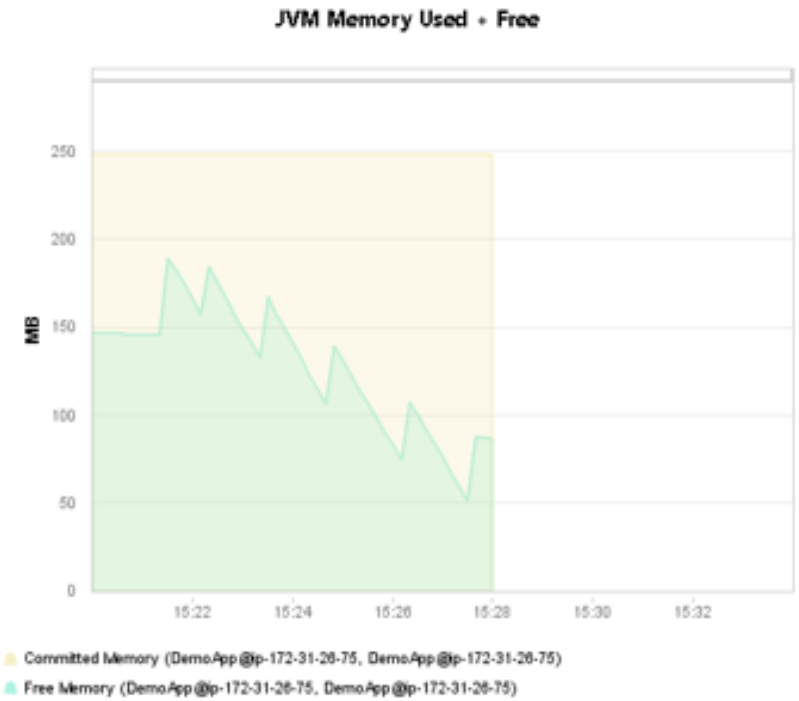
The service addDataToMap adds a specified block of bytes to a hash map using a simple key.

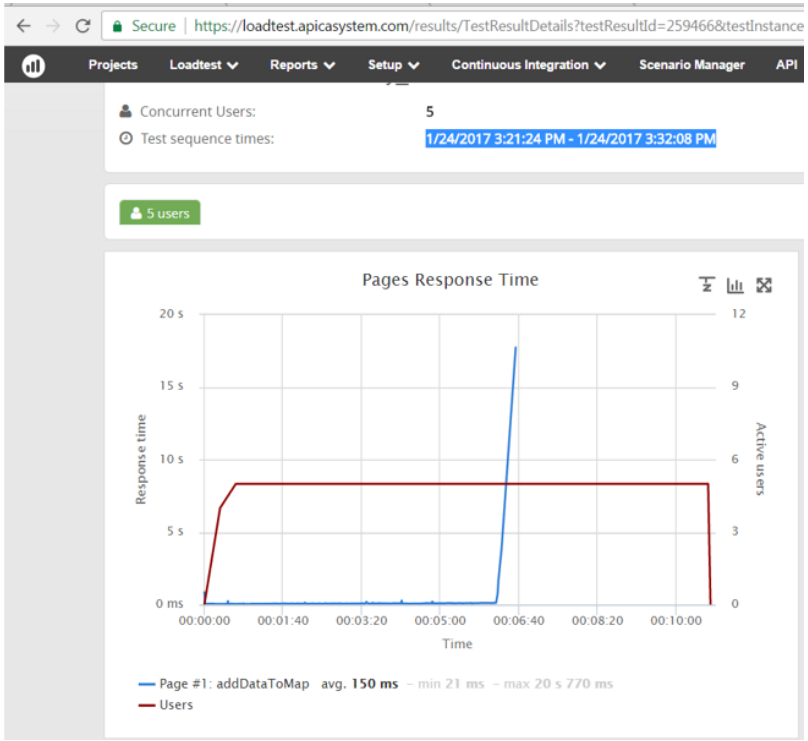
The Problem

The initial load test for the service showed that memory usage continuously increased, eventually causing an OutOfMemory error.

Demo #2 – Memory Leak (V1 Test)

Out of Memory caused server to crash. Timeout of request as a result





dynatrace

Demo #2 – Memory Leak (Cont.)

The Analysis

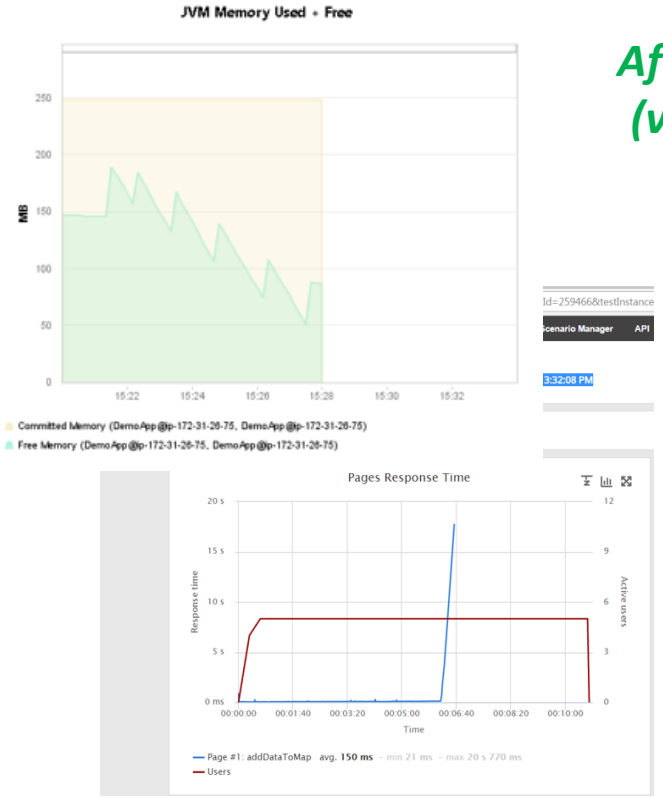
Service was implemented with a key class that did not implement an equals() method. As data was added to the map by each service call, values in the hash map were not being overwritten as intended. Instead, the new values were being appended to the map. This resulted in a memory leak that eventually led to OutOfMemory errors.

The Solution

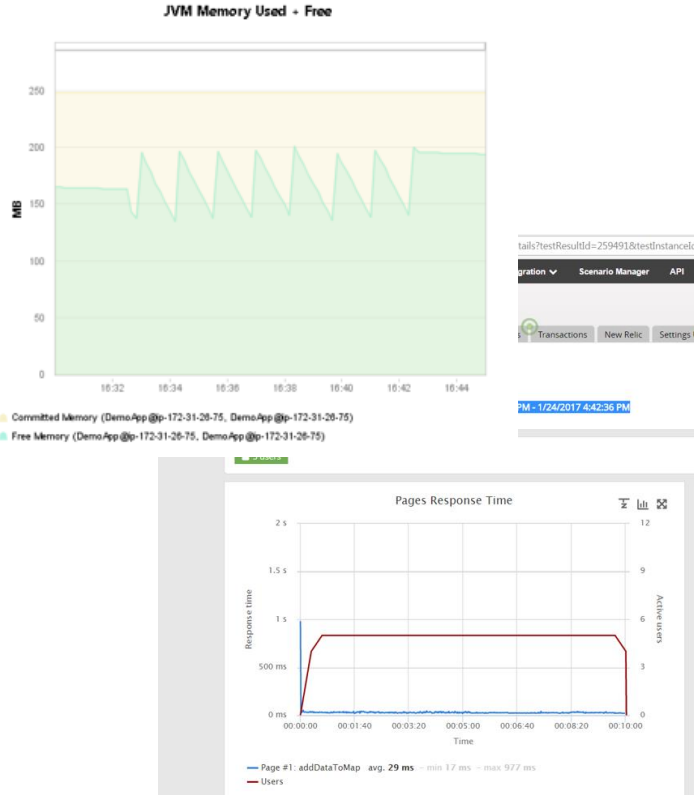
The bad key class was refactored.

Demo #2 – Memory Leak (V2 Test)

Before (v1)



After (v2)



dynatrace

Demo #3 – Poor DB Performance

The Service

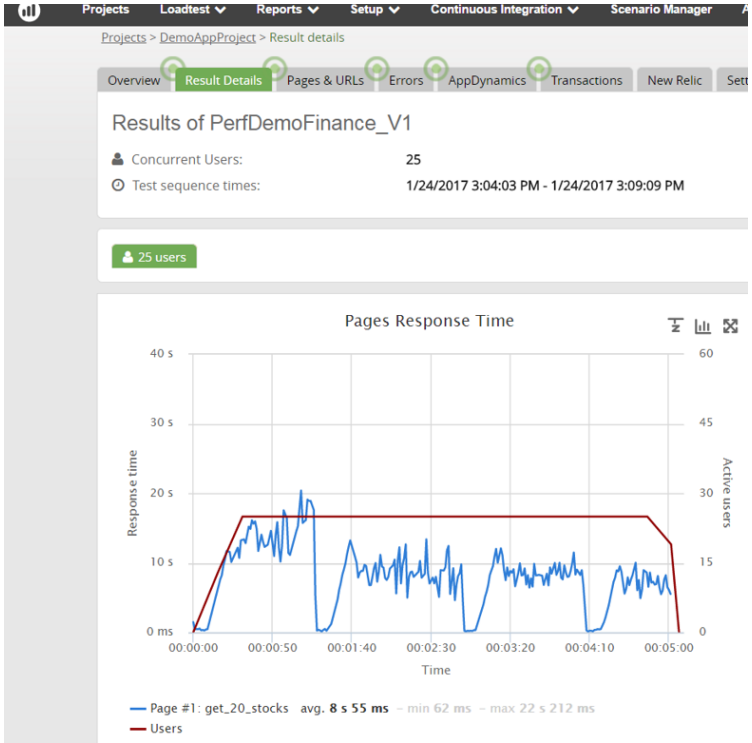
The service getFinancialRecordsByDate fetches financial data for a specified list of stock symbols for a specified date from database tables

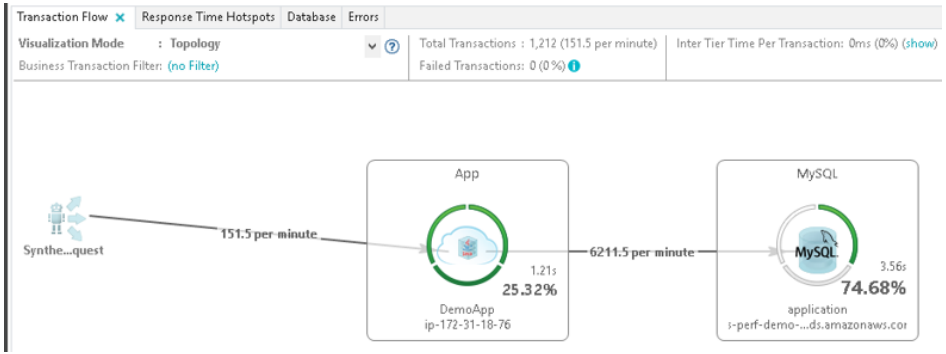
The Problem

The initial load test for the service showed that there were a large number of database calls for each service call.

Demo #3 – Poor DB Performance (V1 Test)

Poor Response times and much of the time in the database





dynatrace

Demo #3 – Poor DB Performance (V1 Test)

41 SQL calls Per Service call.

get_20_stocks	3967.26	io (100.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	6110.20	sync io (73.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	5637.34	sync io (85.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	7066.20	io (85.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	6123.61	sync io (63.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	8150.27	io (98.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	6726.56	io (84.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	6841.03	io (93.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	6837.42	io (98.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	6238.74	io (98.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	6715.28	io (94.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	9000.72	io (95.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	6114.60	io (96.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	7320.33	io (100.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy
get_20_stocks	7804.00	io (88.0%)	7	DemoApp@ip-172-31...	perf-demo-fin...	Asy

PurePaths | Contributors | Errors

PurePath Tree (showing only relevant nodes)

Method	Argument	Exec Total [ms]
doFilter(ServletRequest request, ServletResponse response, FilterChain chain)	/finance/getFinan...	3967.26
service(HttpServletRequest request, HttpServletResponse response)		3967.16
runGetFinancialRecordsByDate(FinancialRecordRequest request, ServiceVersion version)	VERSION_1	3727.00
Database Summary		3248.28
21 × Connection Acquisition @ jrs-perf-demo-db.ce0zdvouniy.us-east-1.rds.amazonaws.com	application	1329.46
20 × select financiali0_.id as id1_0_0_, financiali0_.EXCHANGE as EXCHANGE2_0_0_, financiali0_.Sy	application	959.13
20 × select financialr0_.id as id1_1_, financialr0_.CLOSE_PRICE as CLOSE_PR2_1_, financialr0_.DATE	application	957.00
1 × select financiali0_.id as id1_0_, financiali0_.EXCHANGE as EXCHANGE2_0_, financiali0_.SYMBO	application	2.70

Demo #3 – Poor DB Performance (cont)

The Analysis

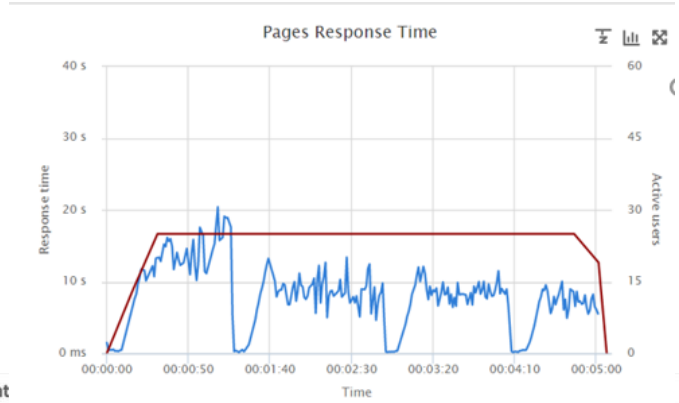
Service suffered from an N+1 query problem where to fetch the data for N stocks, N+1 database queries were made.

The Solution

Refactored to use a SQL join clause to reduce the number of database queries to 1 per service call

Demo #3 – Poor DB Performance (V2 Test)

Before – 10 sec

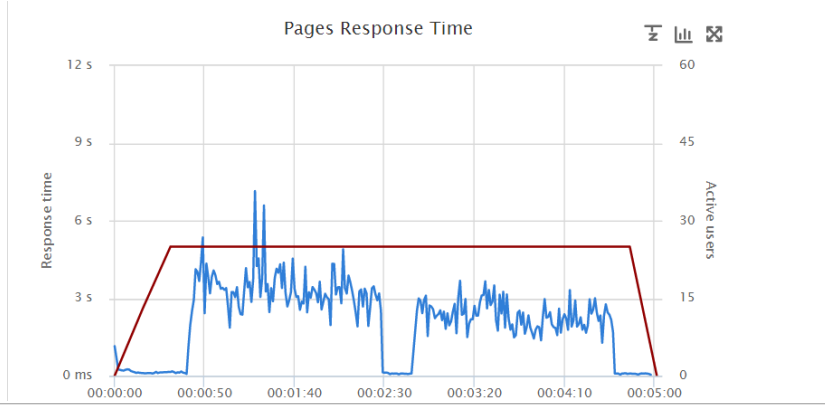


PurePat

Method

- doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
  - service(HttpServletRequest request, HttpServletResponse response)
    - runGetFinancialRecordsByDate(FinancialRecordRequest request, ServiceVersion version)
      - *Database Summary*
        - 21 × Connection Acquisition @ jrs-perf-demo-db.ce0zdvouniy.us-east-1.rds.amazonaws.com
        - 20 × select financiali0_.id as id1_0_0_, financiali0_.EXCHANGE as EXCHANGE2_0_0_, financiali0_.SY
        - 20 × select financialr0_.id as id1_1_, financialr0_.CLOSE_PRICE as CLOSE_PR2_1_, financialr0_.DATE
        - 1 × select financiali0_.id as id1_0_, financiali0_.EXCHANGE as EXCHANGE2_0_, financiali0_.SYMBO

After – 3 sec – 20 less SQL calls



PurePath Tree (showing only relevant nodes)

Method	Argument	Exec Total [ms]
doFilter(ServletRequest request, ServletResponse response, FilterChain chain)	/finance/getFinan...	3113.14
service(HttpServletRequest request, HttpServletResponse response)		3113.04
getFinancialRecordsByDate(FinancialRecordRequest)		*2875.71*
runGetFinancialRecordsByDate(FinancialRecordRequest request, ServiceVersion ve	VERSION_2	2875.71
Database Summary		2872.68
1 × Connection Acquisition @ jrs-perf-demo-db.ce0zdvouniy.us-east-1.rc	application	1797.81
20 × select financiali0_.id as id1_0_0_, financiali0_.EXCHANGE as EXCHANG	application	1072.89
1 × select financialr0_.id as id1_1_, financialr0_.CLOSE_PRICE as CLOSE_PR	application	1.97

dynatrace

Try it yourself!

Follow these simple steps

1. Setup AWS account for use of EBS, EC2, CodePipeline, CodeBuild: https://aws.amazon.com
2. Setup Apica SaaS Trial Account: https://www.apicasystem.com/
3. Setup a Dynatrace AppMon Personal Licence: http://bit.ly/dtpersonal
4. Fork Demo App Code as to perform Builds and Run it – *Readme files have a lot of details* - https://github.com/jsicree

Contact reminders …

- CGI – http://www.cgi.com
- Apica – https://www.apicasystem.com
- Dynatrace - http://www.dynatrace.com

Rob Jahn

Sr. Consultant @ CGI

@RobJahn45

Joe Sicree

Sr. Consultant @ CGI

@JoeSicree

Daniel Freij

Sol. Engineer @ Apica

@DanielFreij

Andi Grabner

DevOps @ Dynatrace

@grabnerandi

Event Reminders

- Dynatrace PERFORM – Feb 7th to 9th
  - http://www.cgi.com/perform

- YouTube Channel: http://bit.ly/dttutorials
- Education & Forums: http://apmu.dynatrace.com & http://answers.dynatrace.com
- Dynatrace AppMon Personal License: http://bit.ly/dtpersonal
- Share Your PurePath: http://bit.ly/sharepurepath
- PurePerformance Podcast: http://bit.ly/pureperf

Open Q&A

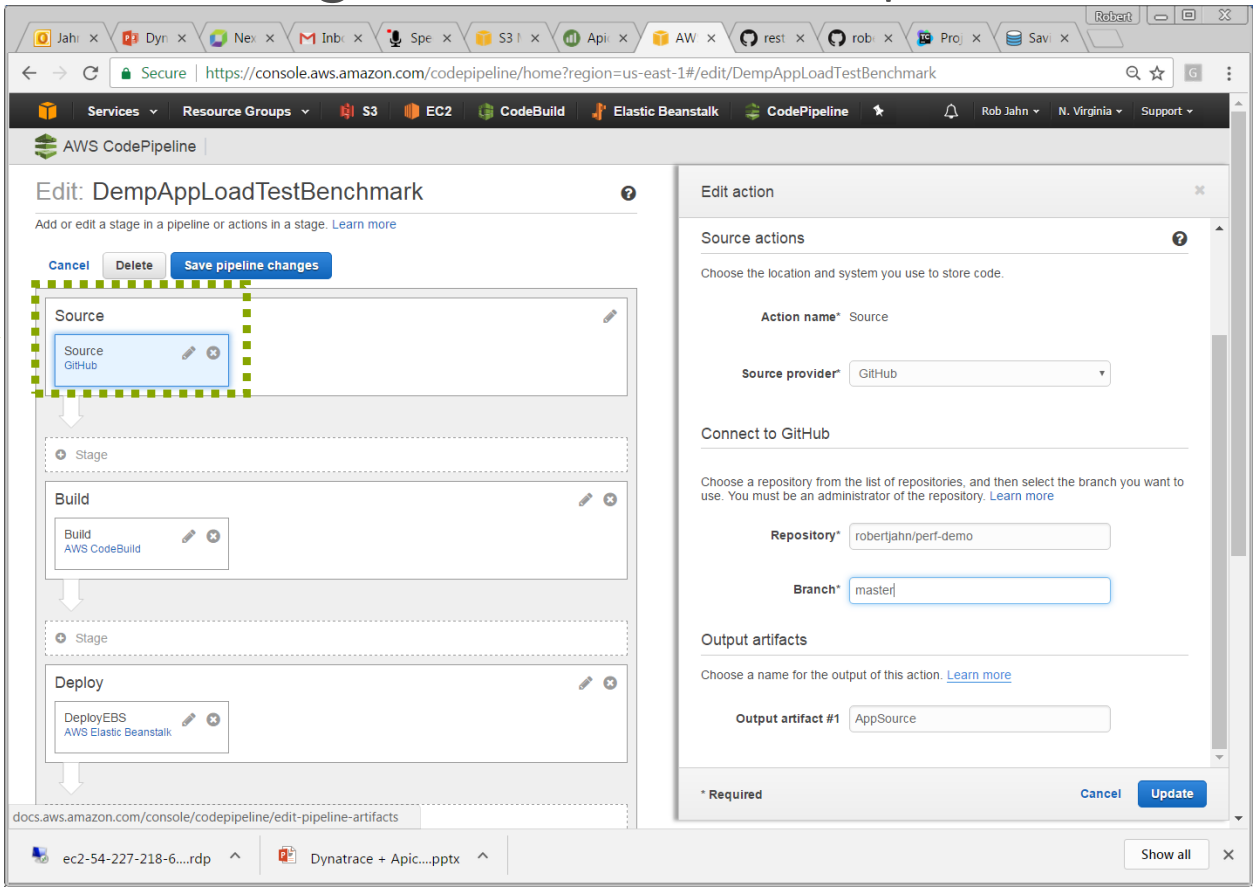
Use the Question Feature in GoTo Webinar
or post it later to
http://answers.dynatrace.com


dynatrace

Configuration Details

Source Stage in AWS CodePipeline

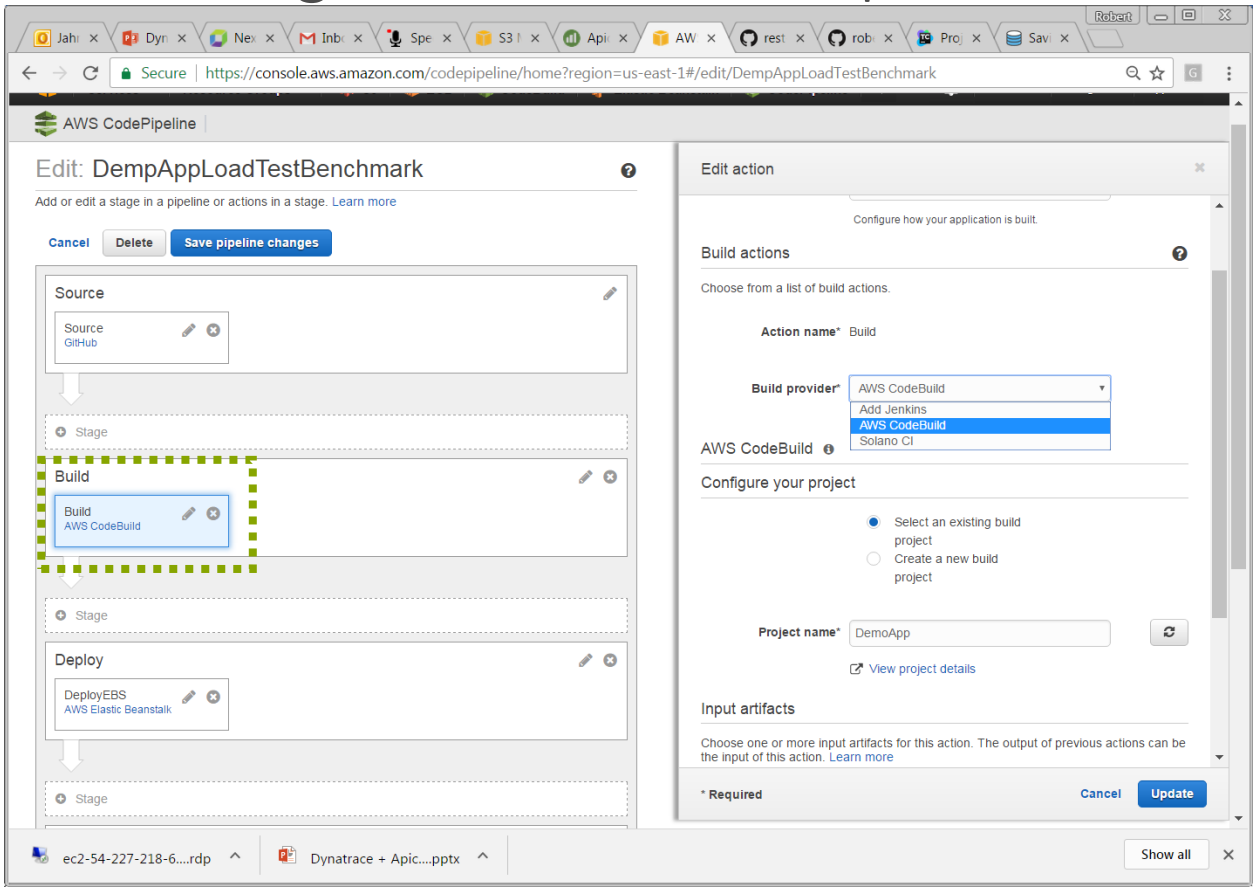
Commit Triggers the Pipeline!



dynatrace

Build Stage in AWS CodePipeline

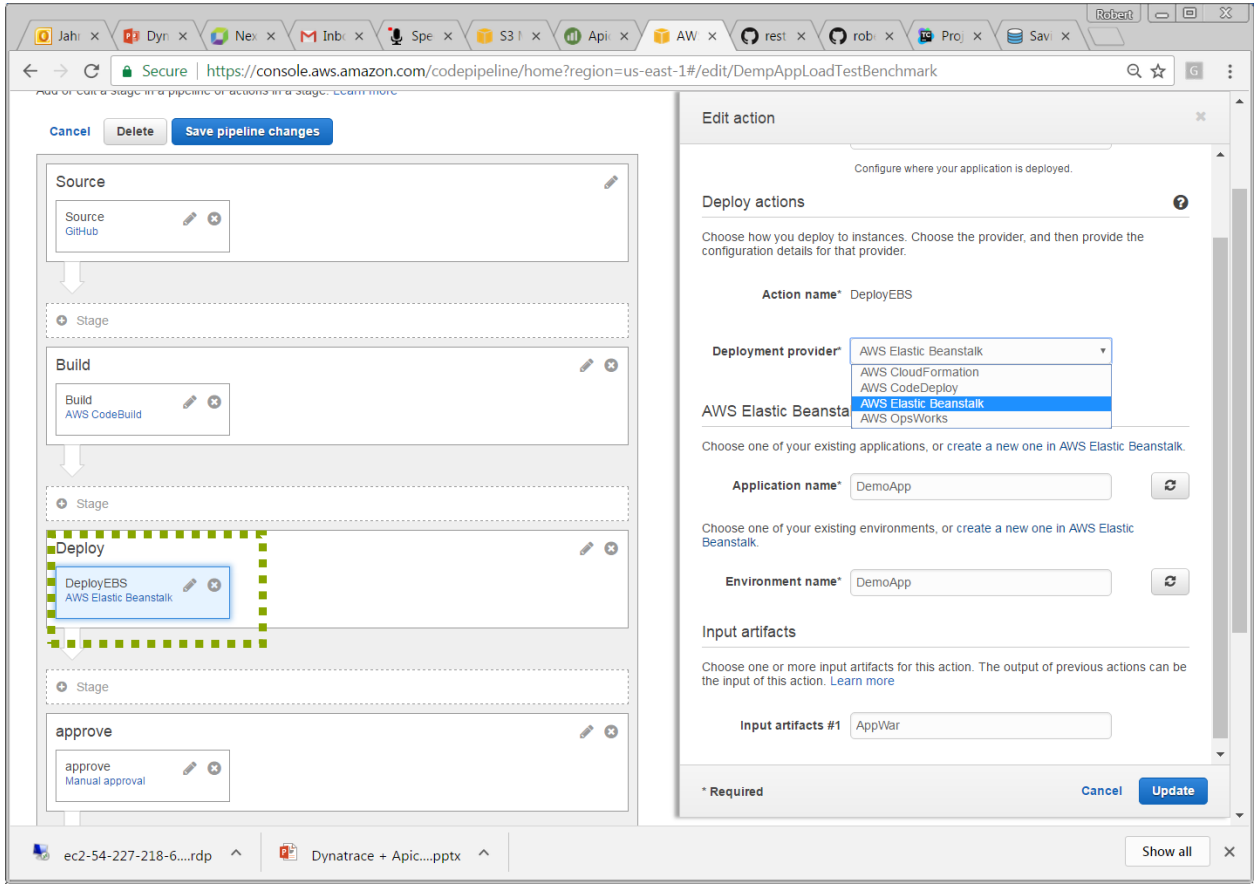
Trigger AWS CodeBuild



dynatrace

Deploy Stage in AWS CodePipeline

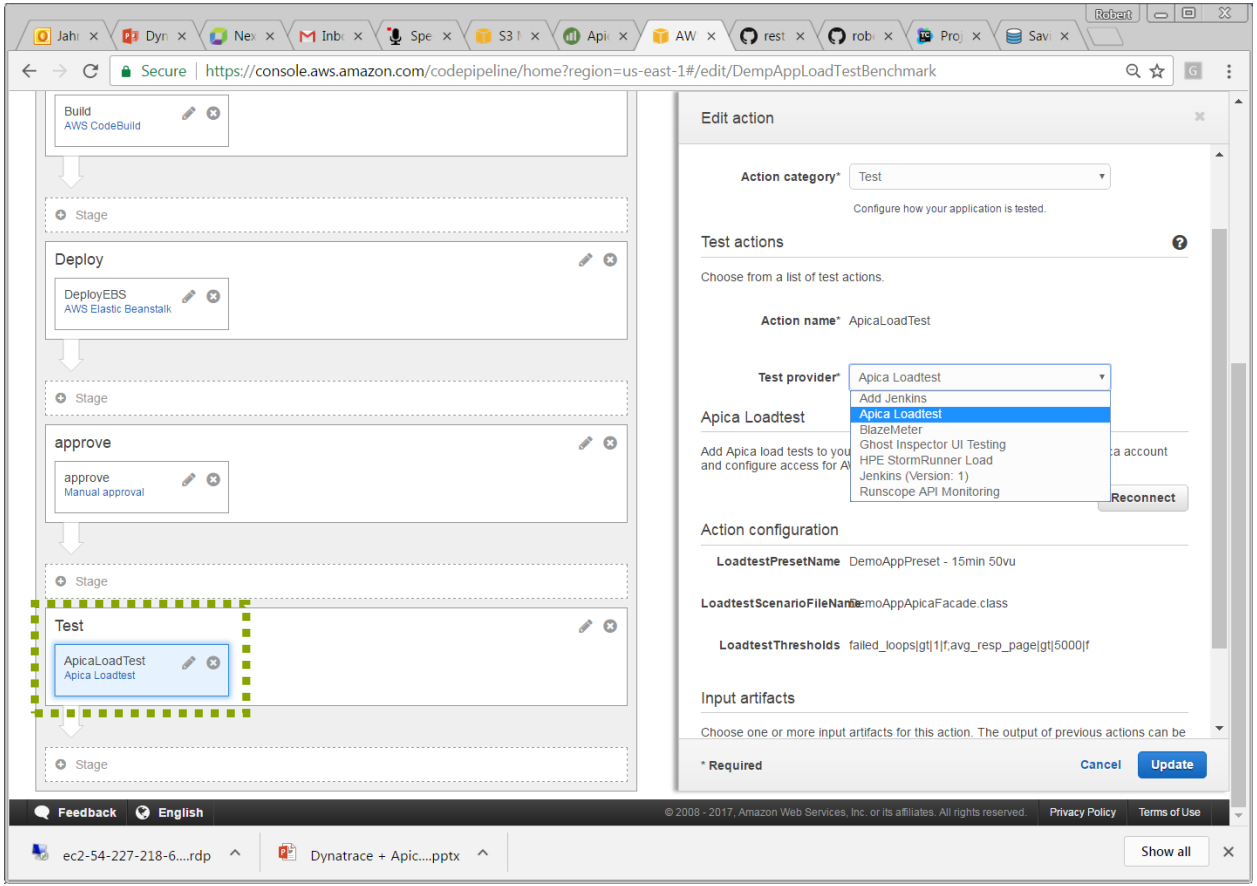
Trigger AWS EBS Deployment



dynatrace

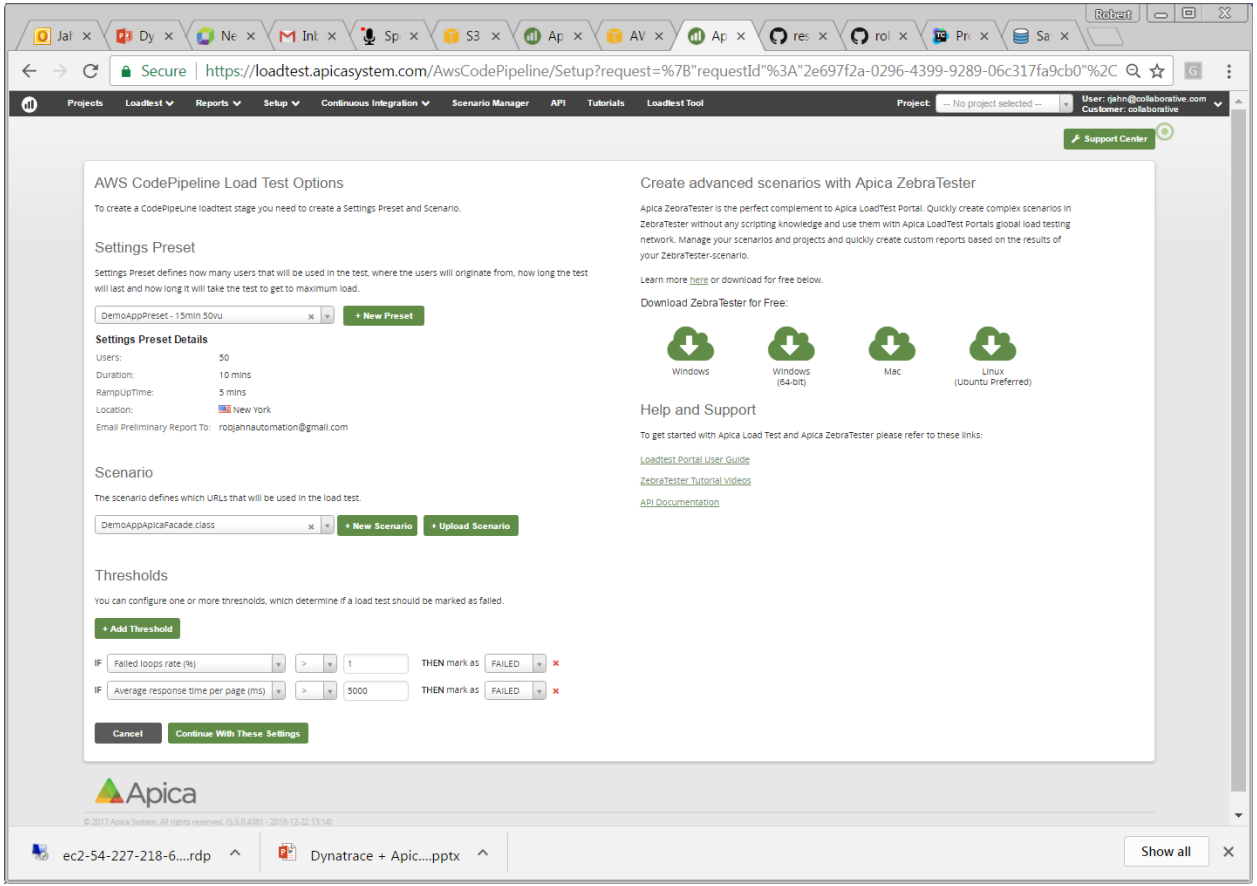
Test Stage in AWS CodePipeline

Trigger Apica Load Test



dynatrace

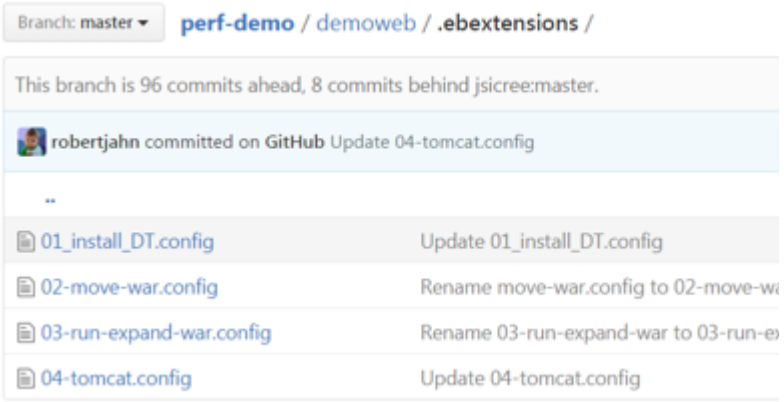
Apica Setup for Test Stage in AWS CodePipeline



dynatrace

EBS Configuration

- To get DT agent on EBS, need to use EB extensions as part of the deploy to install agent and to configure the JVM arguments.



https://github.com/robertjahn/perf-demo/blob/master/demoweb/.ebextensions/01_install_DT.config

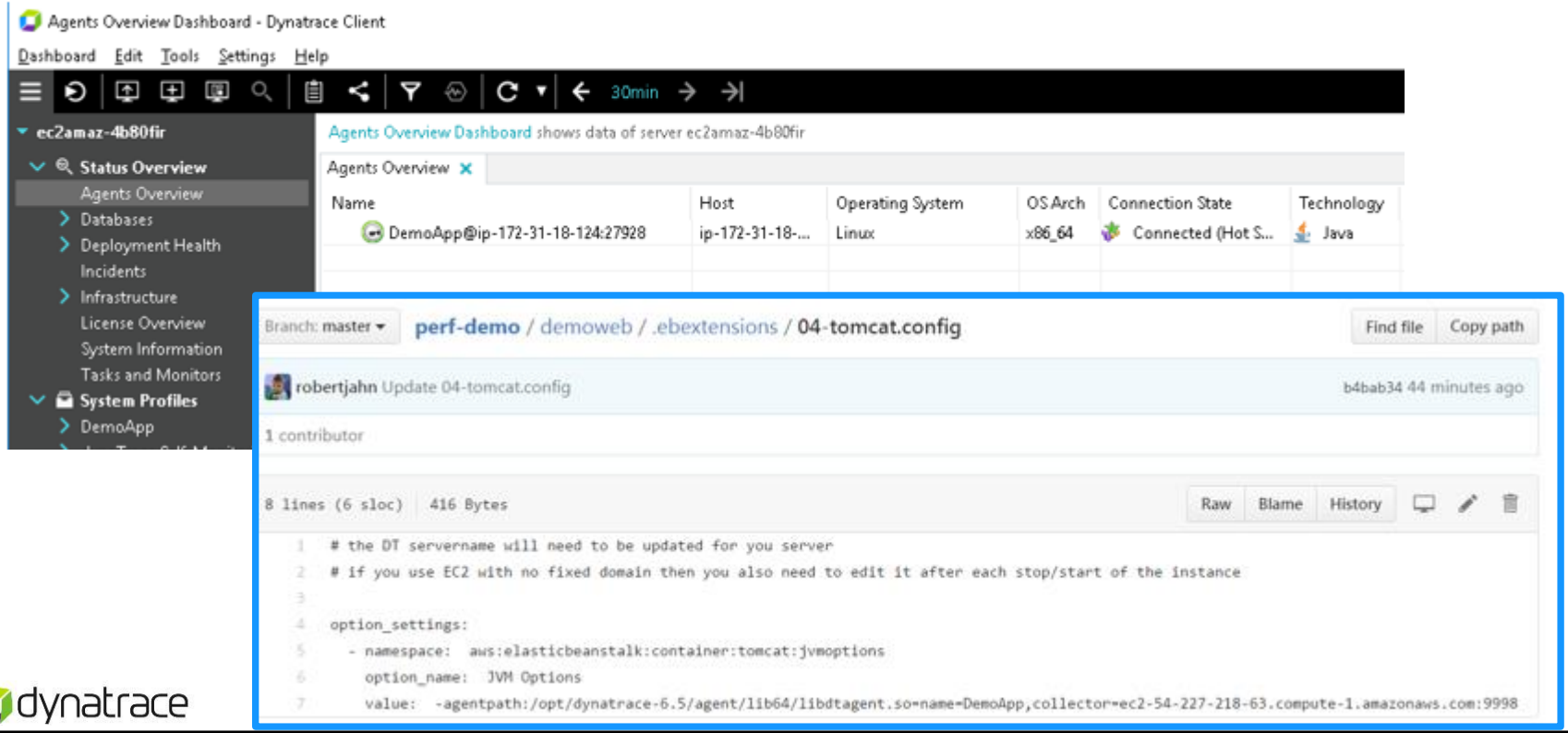
28 lines (25 sloc) | 863 Bytes

```
files:
   "/tmp/dynatrace-agent-6.5.0.1289-unix.zip":
      mode: "000755"
      owner: root
      group: root
      source: https://s3.amazonaws.com/rjahn-temp-s3/dynatrace-agent-6.5.0.1289-unix.zi

   "/opt/elasticbeanstalk/hooks/appdeploy/pre/99_install_dynatrace.sh":
      mode: "000755"
      owner: root
      group: root
      content: |
            #!/bin/bash
            if [ ! -d /opt/dynatrace-6.5 ]
            then
            #unzip_dynatrace
            sudo unzip -o /tmp/dynatrace-agent-6.5.0.1289-unix.zip -d /tmp
            #install_dynatrace in silent mode
            cd /opt
            sudo java -jar /tmp/dynatrace-agent-6.5.0.1289-unix.jar -y
            fi

commands:
    install_dynatrace:
      command: /opt/elasticbeanstalk/hooks/appdeploy/pre/99_install_dynatrace.sh
      cwd: /opt
      test: "[ -d /opt/dynatrace-6.5 ]"
```

dynatrace

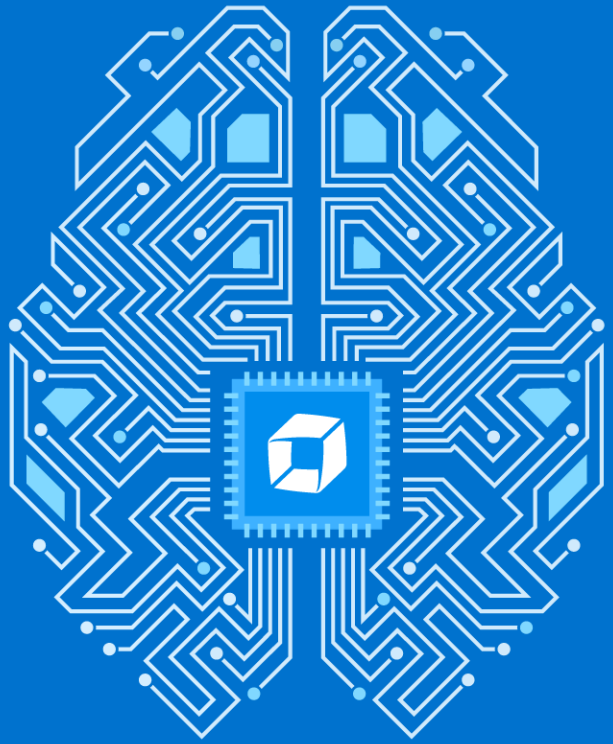
EBS Configuration

DT agent coming in from EBS



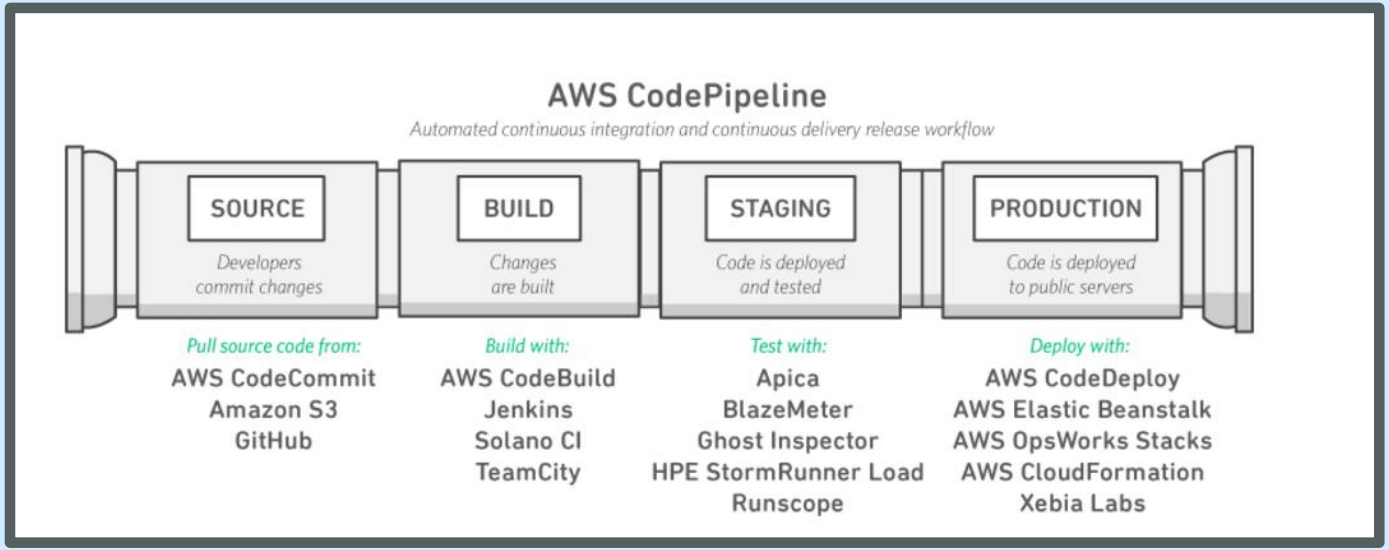
dynatrace

Backup



What is the AWS CodePipeline?

AWS CodePipeline is a continuous integration and continuous delivery service for fast and reliable application and infrastructure updates. CodePipeline builds, tests, and deploys your code every time there is a code change, based on the release process models you define.



3 Minute Explainer Video on https://aws.amazon.com/codepipeline/

Apica Saas and Zebra Tester

Self Service - SaaS Model

Test like a pro with the most powerful SaaS platform, tools and expertise currently available.

Perform high capacity load testing and compare results in a pay-as-you-go model using the Apica LoadTest Portal.

Full API support and scheduling services provide an effective, streamlined platform for continuous testing.

- Know your limits; validate performance
- Scheduled tests/API support
- Unlimited testing for a fixed price
- Capacity-on-demand

Learn More

Apica ZebraTester: Powerful Load Testing Tool

Apica ZebraTester is a powerful load testing and script management software tool available inside Apica's LoadTest Portal.

Pinpoint exactly where and when poor performance starts using ZebraTester's highly accurate load test capabilities. No programming required.

- Powerful Testing tool
- Advanced scripting
- Unlimited testing for a fixed price
- Cloud scalability

Learn More

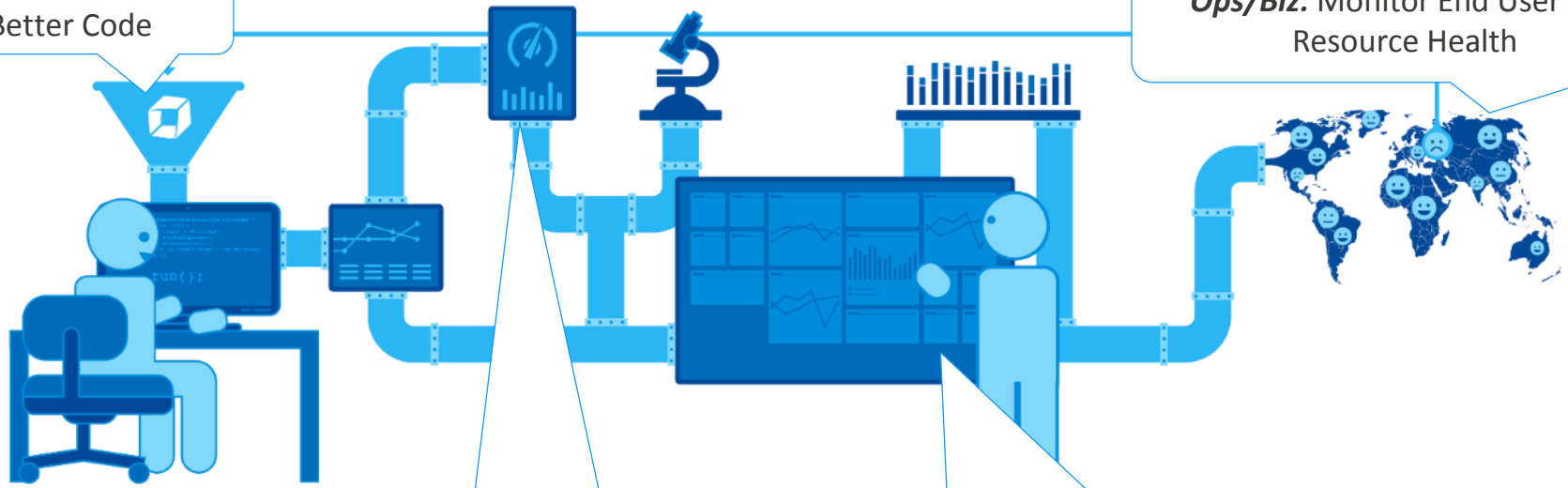
Learn more at https://www.apicasystem.com/load-testing/

Scaling Continuous Innovation with Dynatrace on AWS:

Shift-Left Quality, Reduce Lead Time & Increase Flow

Dev&Test: Check-In Better Code

Ops/Biz: Monitor End User and Resource Health



Test / CI: Stop 80% of Bad Builds Early

Performance: Production Ready Checks on Good Builds Only!

dynatrace

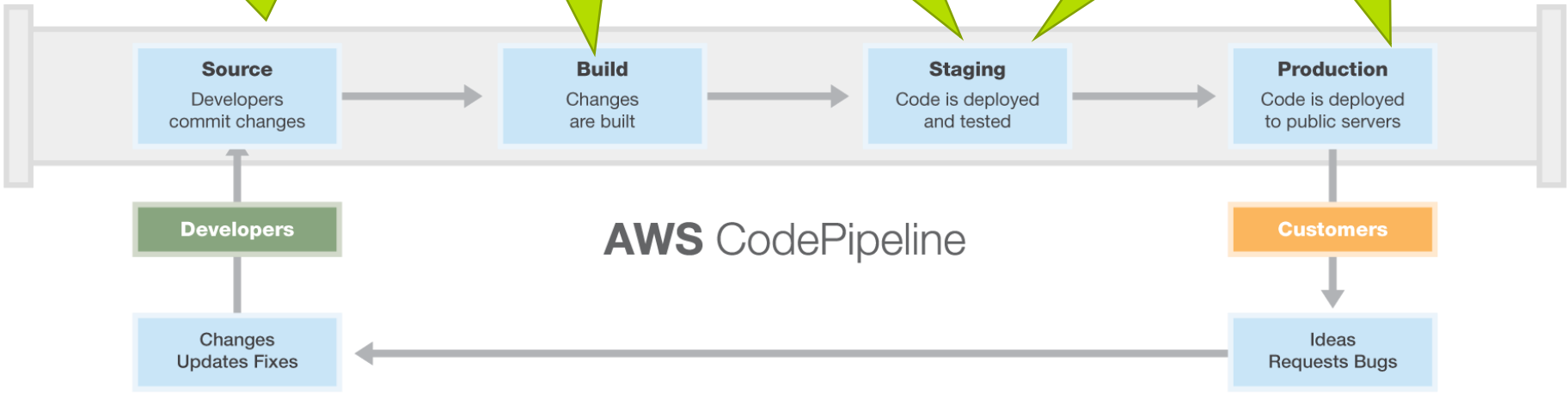
What is the AWS CodePipeline?

#1: Dev Checks in Code!
„A Commit“ to *AWS CodeCommit*, *S3* or *GitHub* triggers the Pipeline. The commited code is the „Output Artifact“ for the next phase!

#2: Building!
Execute a Job in *Jenkins* or *Solano CI*. Result is another „Output Artifact“

#3: Testing
Execute Jenkins, Apica, Ghost Inspector, Runscope API to test your „Output Artifact“

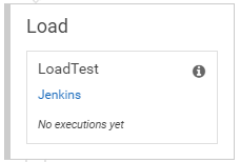
#4: Deploy in Staging or Production
Deploy any Output Artifact using AWS Cloud Formation, AWS CodeDeploy, AWS Elastics Beanstalk



dynatrace

3 Minute Explainer Video on https://aws.amazon.com/codepipeline/

Dynatrace: Automatic *Performance Analysis Across Load Tests*



Apica

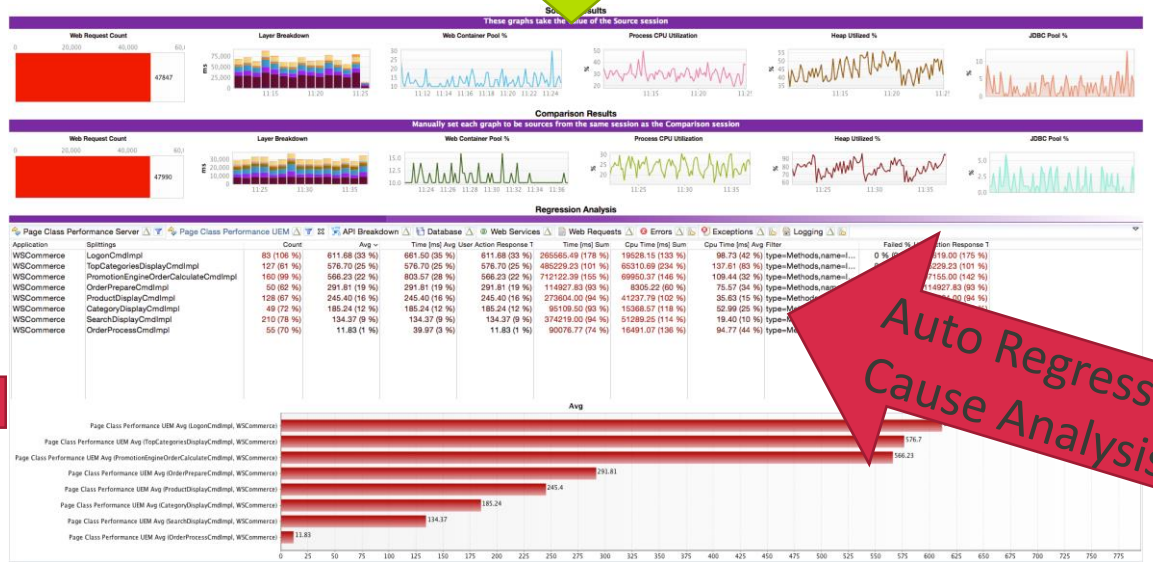
BlazeMeter

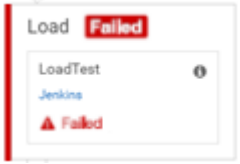
Neotys

APACHE JMeter™

Silk Performer

dynatrace





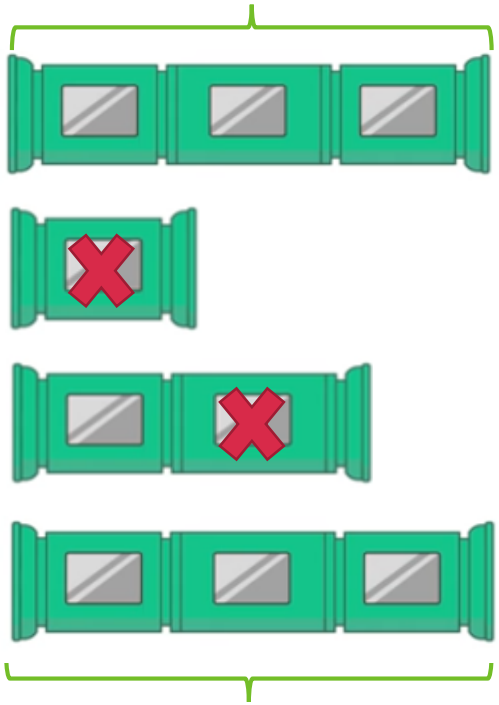
Auto Regressions and Root Cause Analysis across Tests

dynatrace

Learn more @ https://www.dynatrace.com/capabilities/load-testing/

AWS + Dynatrace => *Faster Lead Times, Better Quality*

Optimal Lead Time with 1 Build in Pipeline



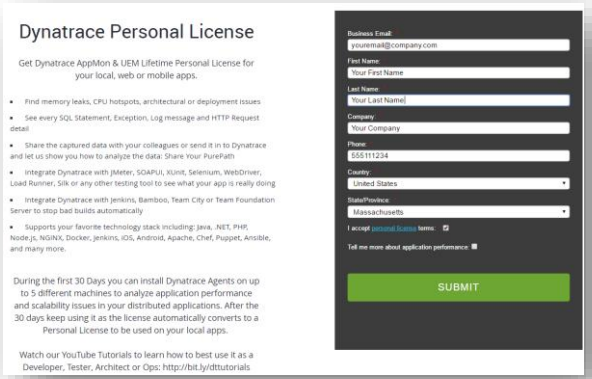
Happy Users

Keeping Optimal Lead Times with Many Builds

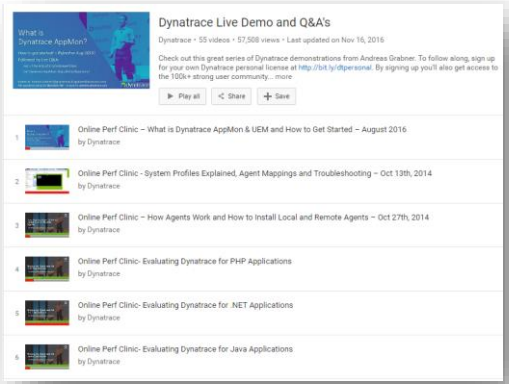
dynatrace

Get Started with 3 Simple Steps

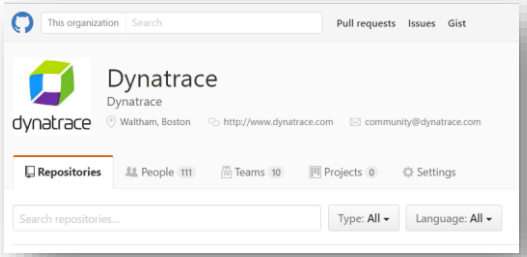
Step #1: License
http://bit.ly/dtpersonal



Step #2: Learn
http://bit.ly/dttutorials

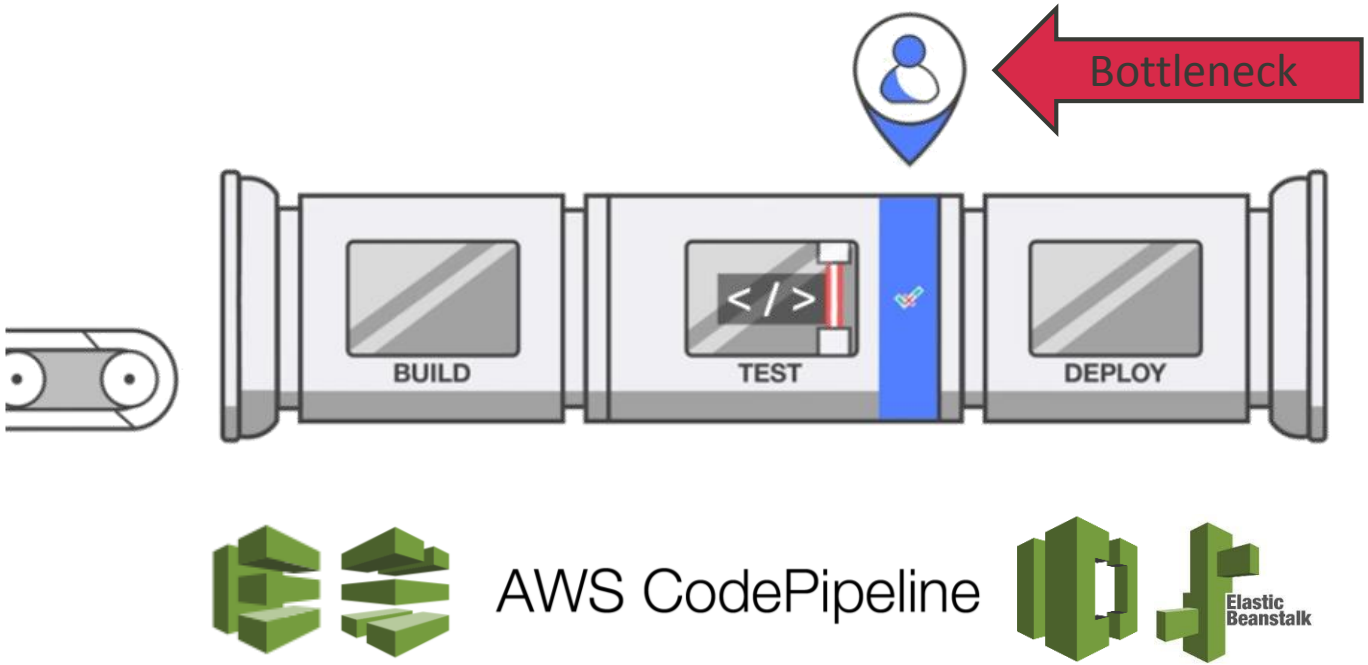


Step #3: Expand
https://github.com/Dynatrace

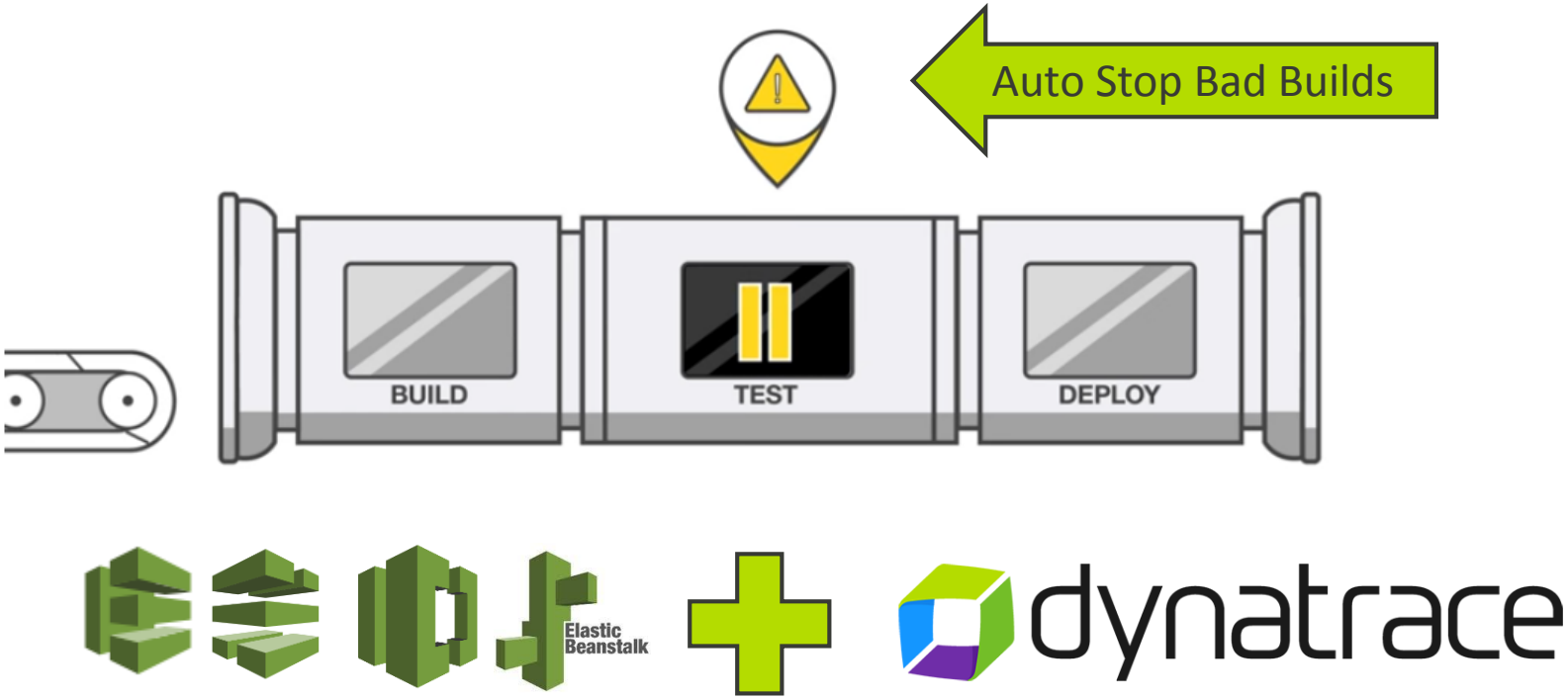


dynatrace

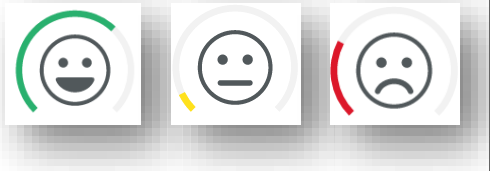
AWS DevOps Pipeline: Manual Approval and Regression Detection *doesn't scale*

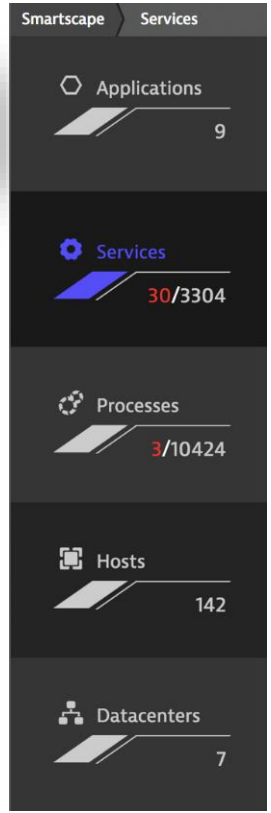


AWS + Dynatrace: Automatically Detect Architectural, Performance and Scalability Regressions



AWS + Dynatrace: Closing the Feedback Loop by automatically monitoring your Deployment





dynatrace


dynatrace