Implementing Performance Engineering as a Self-Service

DevOps to NoOps in Action

dynatrace
Perform

Implementing Performance Engineering as a Self-Service

DevOps to NoOps in Action

dynatrace
Perform

Sumit Nagal
Principal Engineer in Quality
Intuit

Sonja Chevre
Technical Product Manager
Dynatrace

Autonomous Cloud Survey Results

6 to **1**

Dev : Ops Ratio

4 to **1**

Sprint to Releases

2 out of **10**

Business Impacting Releases

Let's hear from Sumit how they improve these values!

Fill out the survey: https://dynatrace.ai/acsurvey

About Sumit

work: intuit. turbotax quickbooks proconnect mint

play: Java CHEF GETCHEF.COM aws CloudBees The Enterprise Jenkins Company

node JS dynatrace OverOps splunk>

kubernetes docker argo

Who we serve

Consumers

Small Businesses

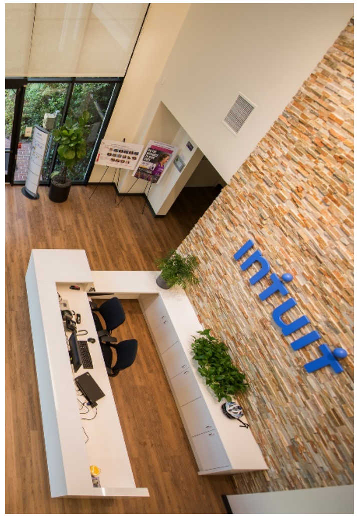
Self-Employed

Our mission

Powering prosperity around the world

Who we are

1983	1993	8,900	21	$6B	50M
Founded	IPO	Employees	Locations	Revenue	Customers



Our mission statement

Find Customer Issue – 1 Minute

FIX It – 10 Minutes

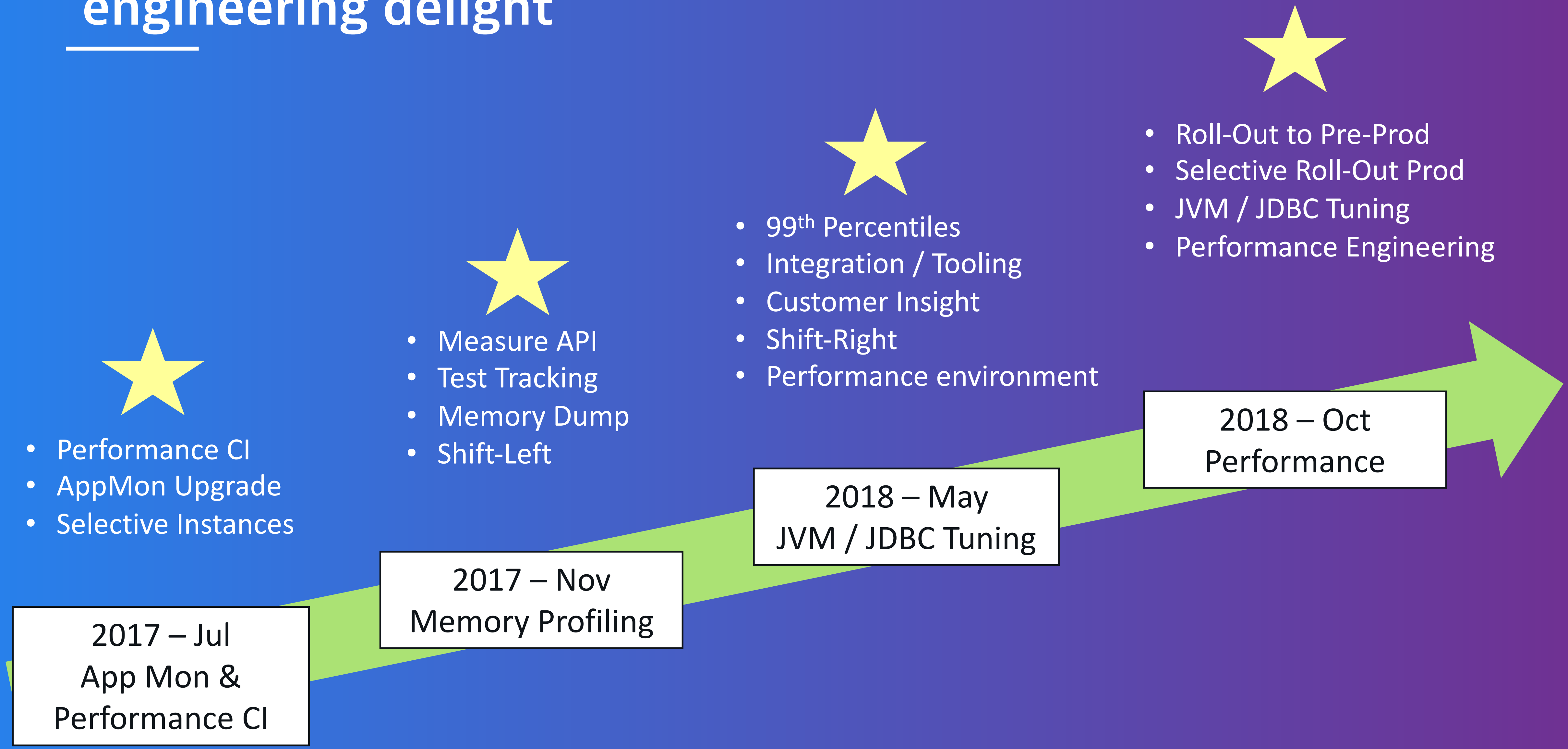
RELEASE 45 minutes

Improving customer 99th percentile perform 99.9th f top flows

Dev-Ops

operation transformation

Our journey: using Dynatrace for customer insight & engineering delight



2017 – Jul
App Mon & Performance CI

- Performance CI
- AppMon Upgrade
- Selective Instances

2017 – Nov
Memory Profiling

- Measure API
- Test Tracking
- Memory Dump
- Shift-Left

2018 – May
JVM / JDBC Tuning

- 99th Percentiles
- Integration / Tooling
- Customer Insight
- Shift-Right
- Performance environment

2018 – Oct
Performance

- Roll-Out to Pre-Prod
- Selective Roll-Out Prod
- JVM / JDBC Tuning
- Performance Engineering

Shift-Right: “Performance for everybody”

Dynatrace as Self-Service: We ALL ARE experts!

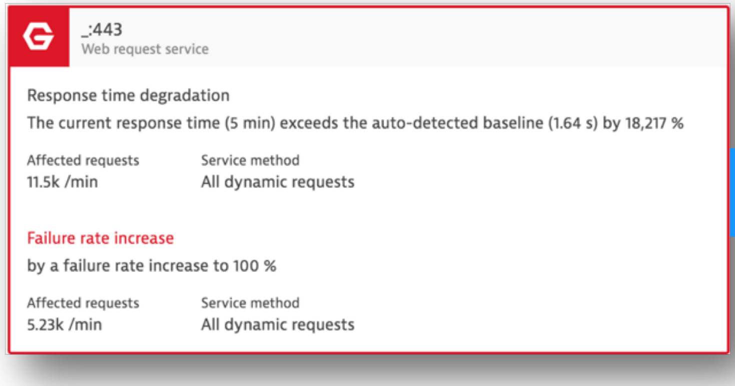
Benefit of self-service teams

10+ Scrum Team

5 Engineers

Example 1: API/DB degradation

#1 Impact



_:443
Web request service

Response time degradation

The current response time (5 min) exceeds the auto-detected baseline (1.64 s) by 18,217 %

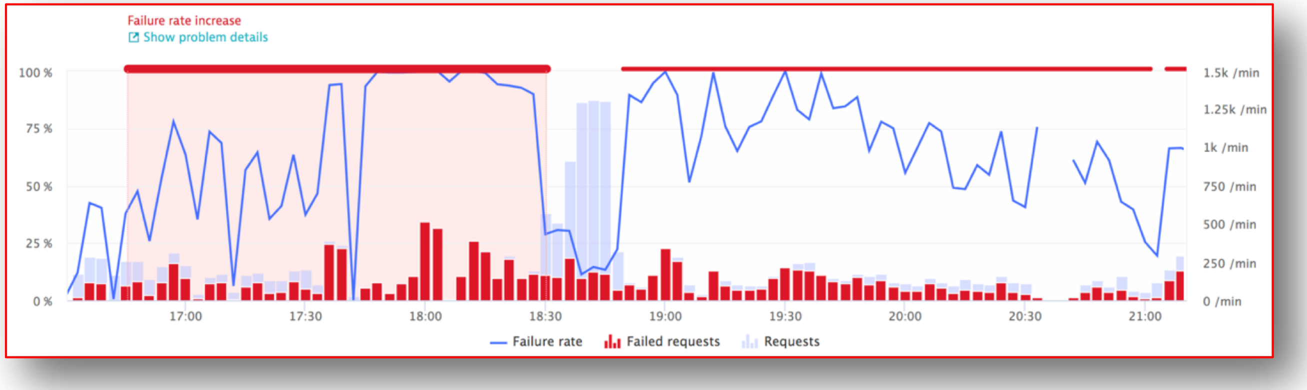
Affected requests	Service method
11.5k /min	All dynamic requests

Failure rate increase

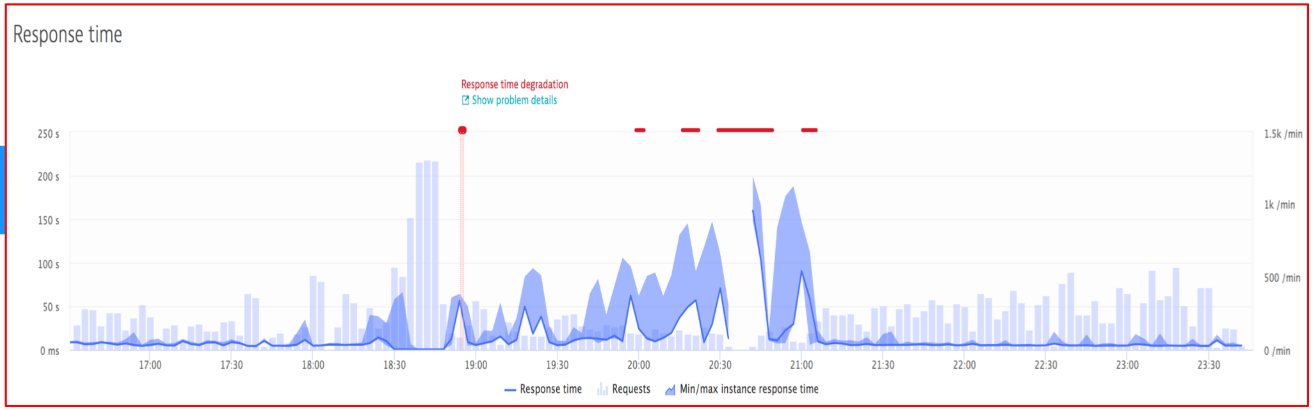
by a failure rate increase to 100 %

Affected requests	Service method
5.23k /min	All dynamic requests

#2 Impact Details

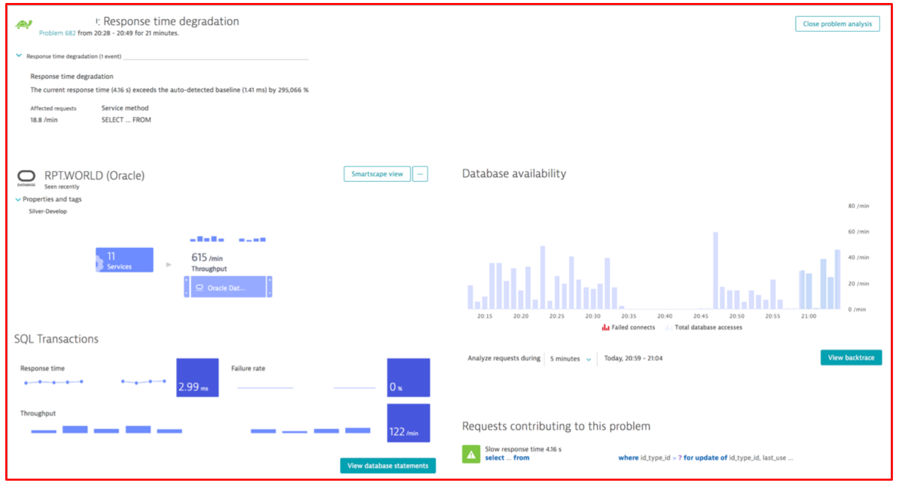


#2 Impact Details



Response time

#3 Root Cause



: Response time degradation
Problem 682 from 20:28 - 20:49 for 21 minutes.

Close problem analysis

Response time degradation (1 event)

Response time degradation

The current response time (4.16 s) exceeds the auto-detected baseline (1.41 ms) by 295,066 %

Affected requests	Service method
18.8 /min	SELECT ... FROM

RPT.WORLD (Oracle)
Seen recently

Smartscape view

Properties and tags
Silver-Develop

SQL Transactions

View database statements

Database availability

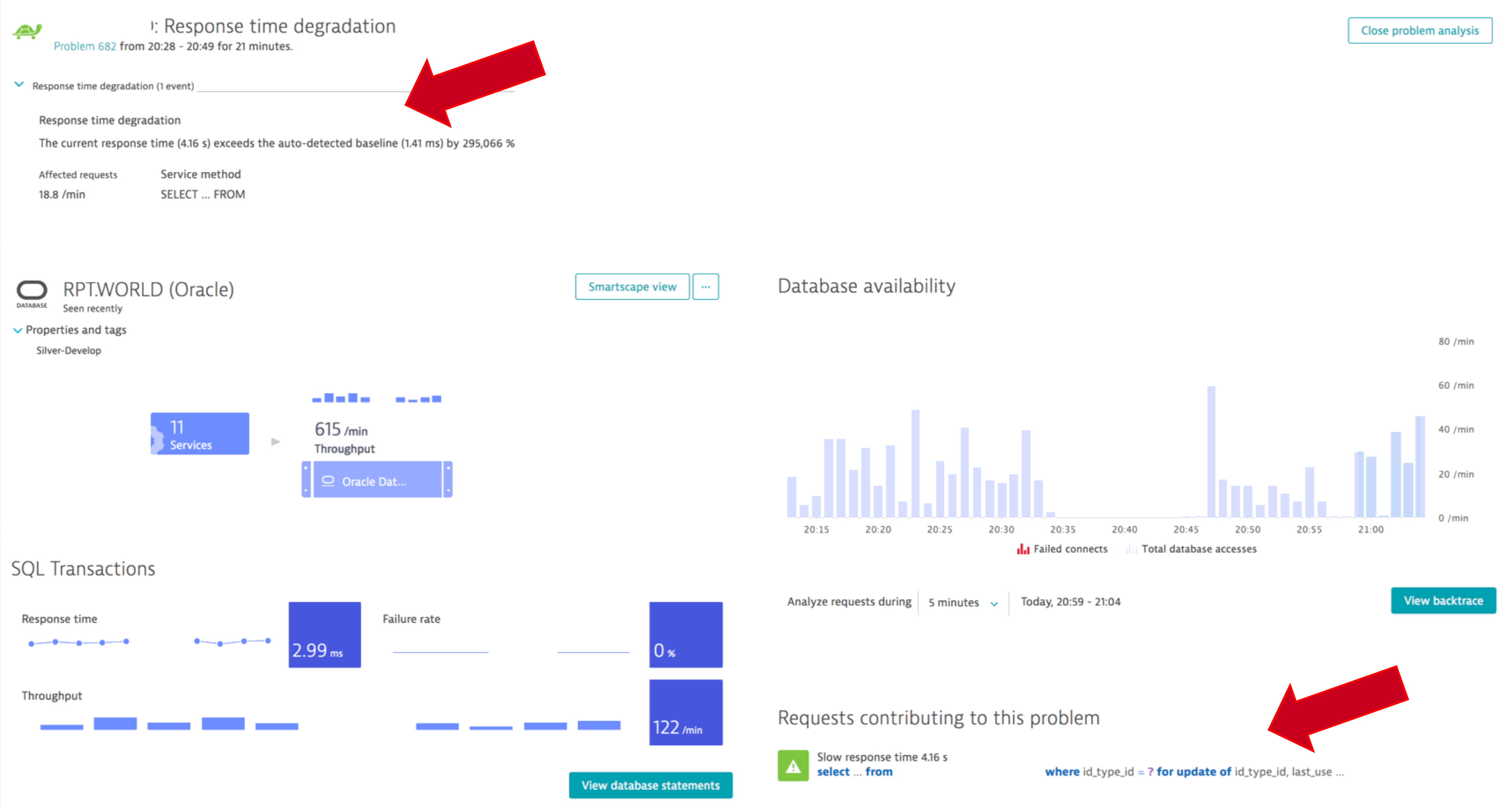
Analyze requests during 5 minutes Today, 20:59 - 21:04

View backtrace

Requests contributing to this problem

Slow response time 4.16 s
select ... from where id_type_id = ? for update of id_type_id, last_use ...

#1 Impact details: SQL degradation & database availability issue



: Response time degradation

Problem 682 from 20:28 - 20:49 for 21 minutes.

Close problem analysis

Response time degradation (1 event)

Response time degradation

The current response time (4.16 s) exceeds the auto-detected baseline (1.41 ms) by 295,066 %

Affected requests	Service method
18.8 /min	SELECT ... FROM

RPT.WORLD (Oracle)

Seen recently

Properties and tags

Silver-Develop

Smartscape view

Database availability

Analyze requests during 5 minutes Today, 20:59 - 21:04

View backtrace

SQL Transactions

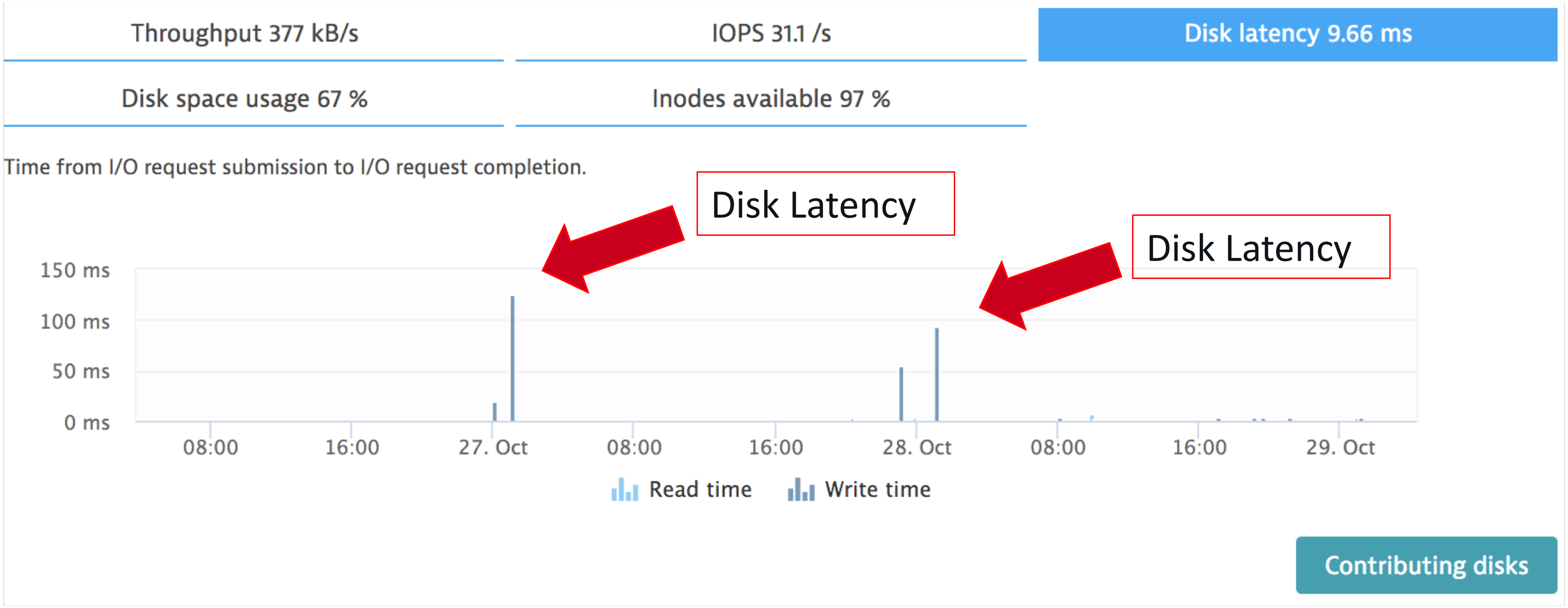
Requests contributing to this problem

Slow response time 4.16 s

select ... from where id_type_id = ? for update of id_type_id, last_use ...

View database statements

Example 2: Disk latency issues due to API



Throughput 377 kB/s	IOPS 31.1 /s	Disk latency 9.66 ms
Disk space usage 67 %	Inodes available 97 %	

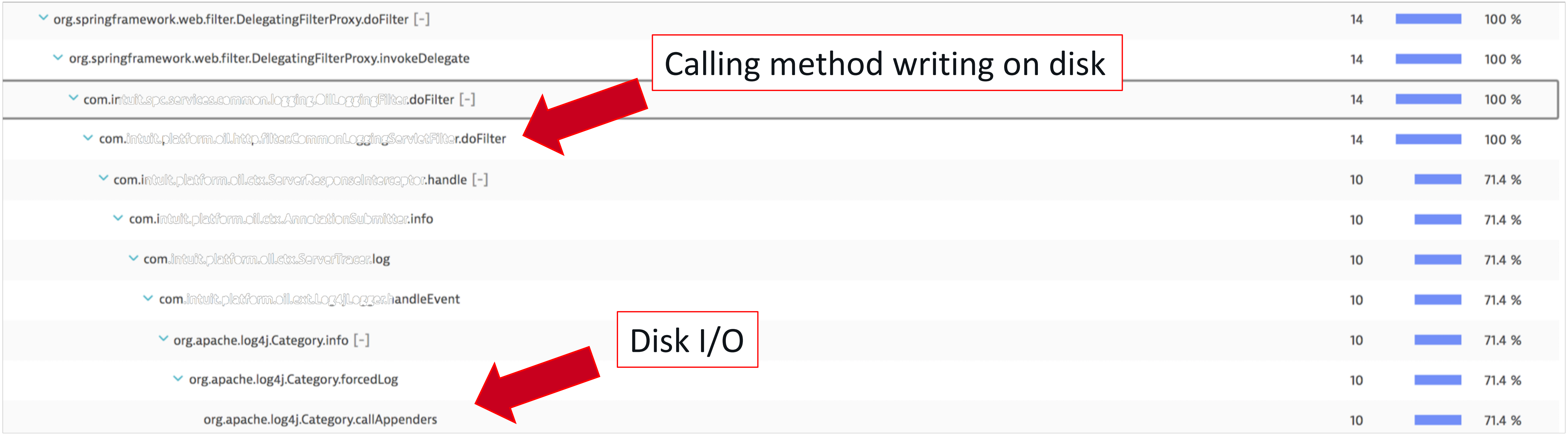
Time from I/O request submission to I/O request completion.

Disk Latency

Disk Latency

Contributing disks

#2 Root cause: Excessive application logging hitting disk

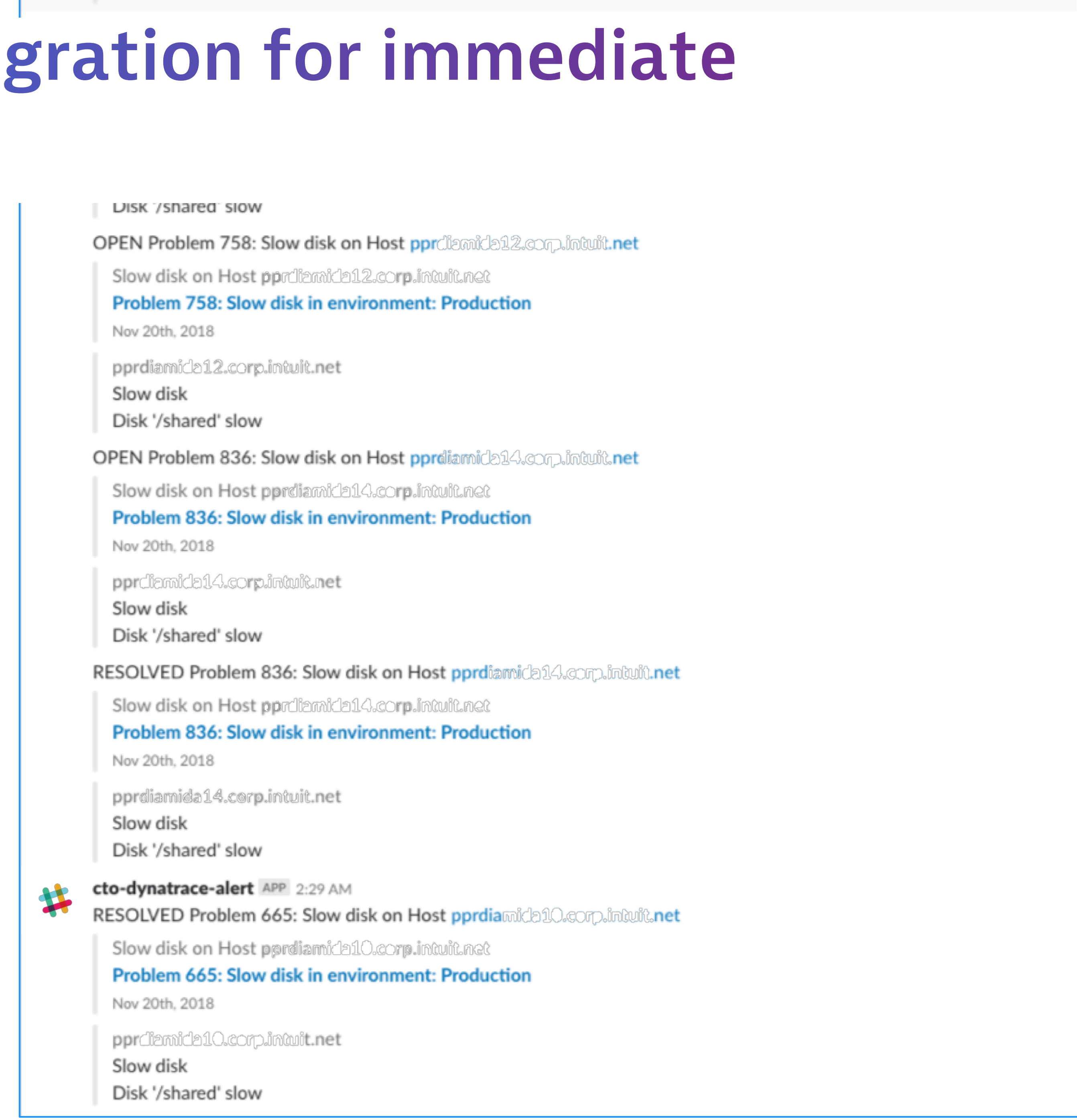


org.springframework.web.filter.DelegatingFilterProxy.doFilter [-]	14	▬	100 %
org.springframework.web.filter.DelegatingFilterProxy.invokeDelegate	14	▬	100 %
com.intuit.spc.services.common.logging.OilLoggingFilter.doFilter [-]	14	▬	100 %
com.intuit.platform.oil.http.filter.CommonLoggingServletFilter.doFilter	14	▬	100 %
com.intuit.platform.oil.ctx.ServerResponseInterceptor.handle [-]	10	▬	71.4 %
com.intuit.platform.oil.ctx.AnnotationSubmitter.info	10	▬	71.4 %
com.intuit.platform.oil.ctx.ServerTracer.log	10	▬	71.4 %
com.intuit.platform.oil.ext.Log4jLogger.handleEvent	10	▬	71.4 %
org.apache.log4j.Category.info [-]	10	▬	71.4 %
org.apache.log4j.Category.forcedLog	10	▬	71.4 %
org.apache.log4j.Category.callAppenders	10	▬	71.4 %

Calling method writing on disk

Disk I/O

Dynatrace Tip: Slack integration for immediate developers feedback



Disk '/shared' slow

OPEN Problem 758: Slow disk on Host pprdiamida12.corp.intuit.net

Slow disk on Host pprdiamida12.corp.intuit.net
Problem 758: Slow disk in environment: Production
Nov 20th, 2018

pprdiamida12.corp.intuit.net
Slow disk
Disk '/shared' slow

OPEN Problem 836: Slow disk on Host pprdiamida14.corp.intuit.net

Slow disk on Host pprdiamida14.corp.intuit.net
Problem 836: Slow disk in environment: Production
Nov 20th, 2018

pprdiamida14.corp.intuit.net
Slow disk
Disk '/shared' slow

RESOLVED Problem 836: Slow disk on Host pprdiamida14.corp.intuit.net

Slow disk on Host pprdiamida14.corp.intuit.net
Problem 836: Slow disk in environment: Production
Nov 20th, 2018

pprdiamida14.corp.intuit.net
Slow disk
Disk '/shared' slow

cto-dynatrace-alert APP 2:29 AM
RESOLVED Problem 665: Slow disk on Host pprdiamida10.corp.intuit.net

Slow disk on Host pprdiamida10.corp.intuit.net
Problem 665: Slow disk in environment: Production
Nov 20th, 2018

pprdiamida10.corp.intuit.net
Slow disk
Disk '/shared' slow

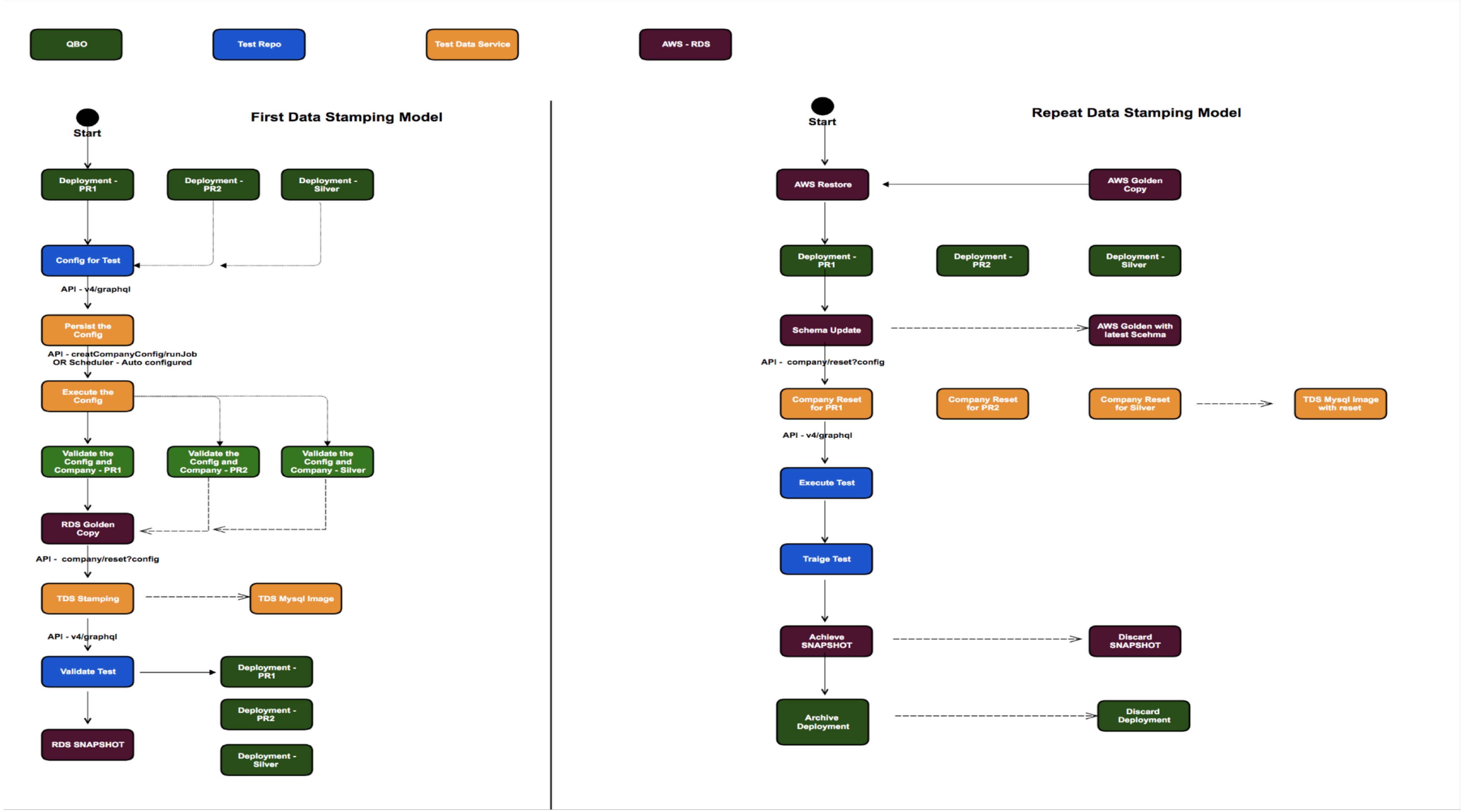
Shift-Right: Production feeds pre-prod

Data & behavior for “Production Twin”- testing

Fix production data related issues

- More performance issues captured in pre-prod
- Lesser performance issues in prod

Shift-Right: Data



QBO

Test Repo

Test Data Service

AWS - RDS

First Data Stamping Model

Start

Deployment - PR1

Deployment - PR2

Deployment - Silver

Config for Test

API - v4/graphql

Persist the Config

API - creatCompanyConfig/runJob OR Scheduler - Auto configured

Execute the Config

Validate the Config and Company - PR1

Validate the Config and Company - PR2

Validate the Config and Company - Silver

RDS Golden Copy

API - company/reset?config

TDS Stamping

TDS Mysql Image

API - v4/graphql

Validate Test

Deployment - PR1

Deployment - PR2

Deployment - Silver

RDS SNAPSHOT

Repeat Data Stamping Model

Start

AWS Restore

AWS Golden Copy

Deployment - PR1

Deployment - PR2

Deployment - Silver

Schema Update

AWS Golden with latest Scehma

API - company/reset?config

Company Reset for PR1

Company Reset for PR2

Company Reset for Silver

TDS Mysql Image with reset

API - v4/graphql

Execute Test

Traige Test

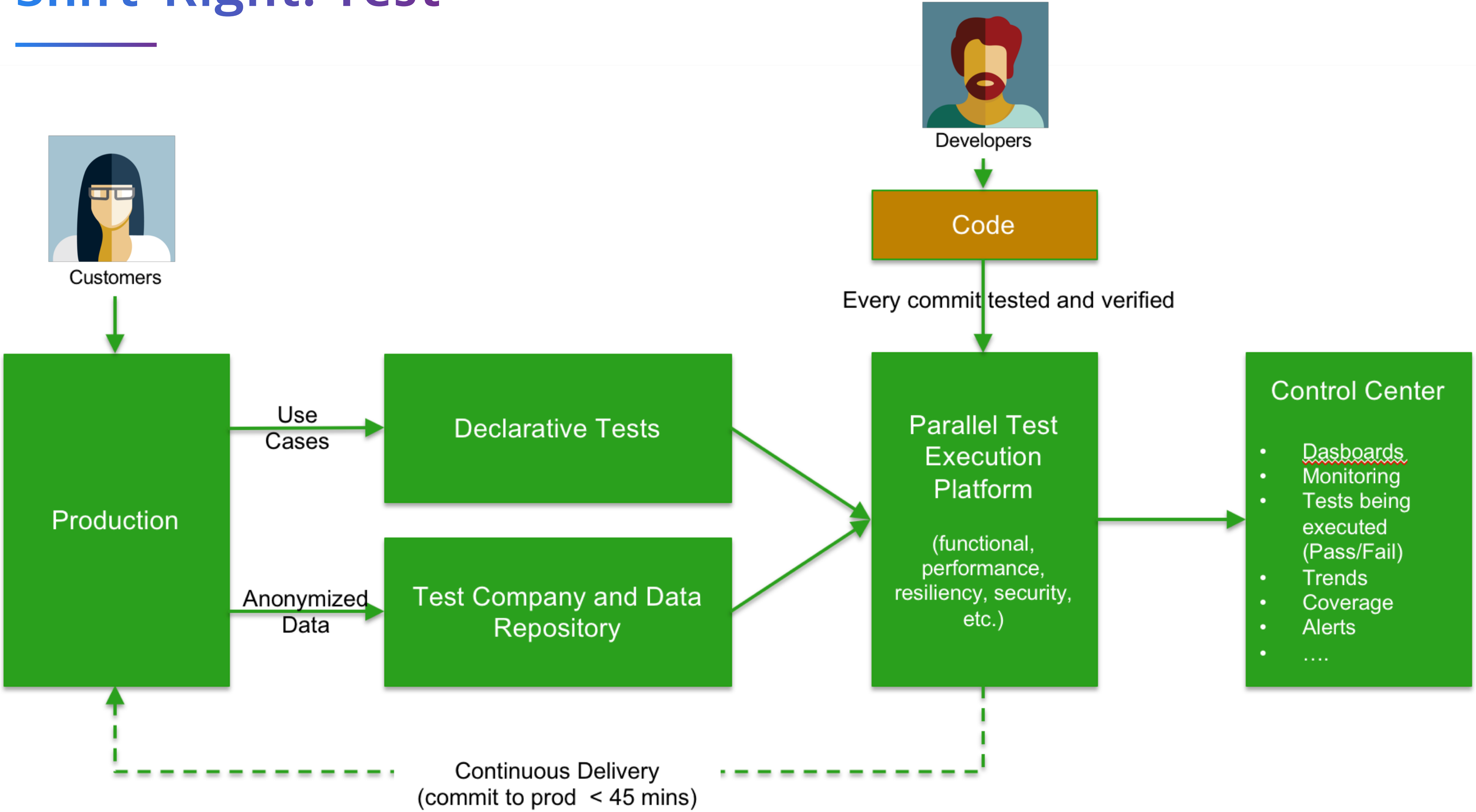
Achieve SNAPSHOT

Discard SNAPSHOT

Archive Deployment

Discard Deployment

Shift-Right: Test



Customers

Developers

Code

Every commit tested and verified

Production

Use Cases

Anonymized Data

Declarative Tests

Test Company and Data Repository

Parallel Test Execution Platform

(functional, performance, resiliency, security, etc.)

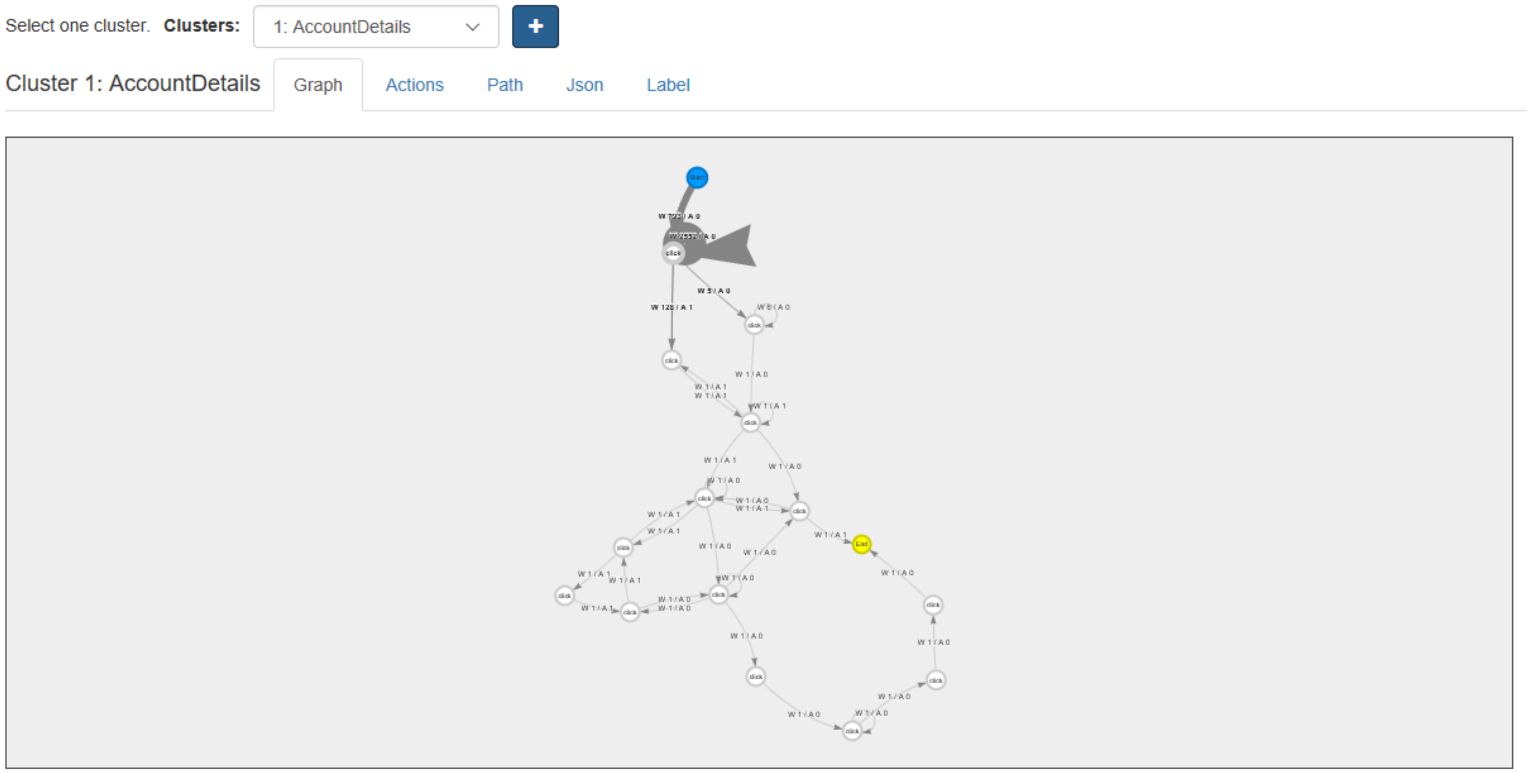
Control Center

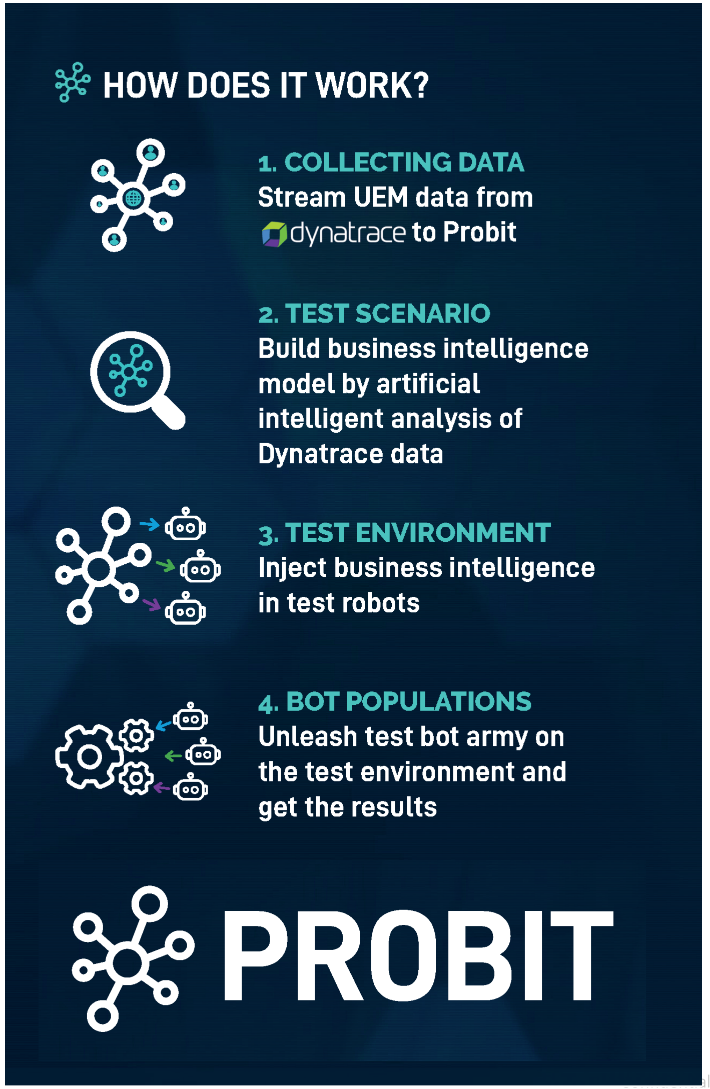
- Dasboards
- Monitoring
- Tests being executed (Pass/Fail)
- Trends
- Coverage
- Alerts
- ….

Continuous Delivery (commit to prod < 45 mins)

Dynatrace Tip: Export Dynatrace user-session data for testing

- User session export automates the streaming of real user monitoring data, including all user actions, high-level timings, and error data





HOW DOES IT WORK?

1. COLLECTING DATA
Stream UEM data from dynatrace to Probit

2. TEST SCENARIO
Build business intelligence model by artificial intelligent analysis of Dynatrace data

3. TEST ENVIRONMENT
Inject business intelligence in test robots

4. BOT POPULATIONS
Unleash test bot army on the test environment and get the results

PROBIT

Shift-Left: Automated feedback

Continuous performance feedback in your pipeline

Increased developer productivity

- Instant performance feedback on developer machines
- PR pipeline block code for check-in
- Regression detection on every deployment

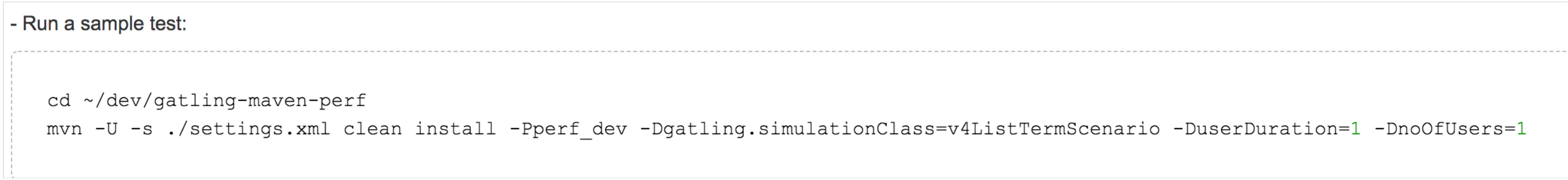
Shift-Left: Re-purpose tests in many stages

Write Test Once

https://github.com/intuit/karate

Performance feedback on the developer workstation

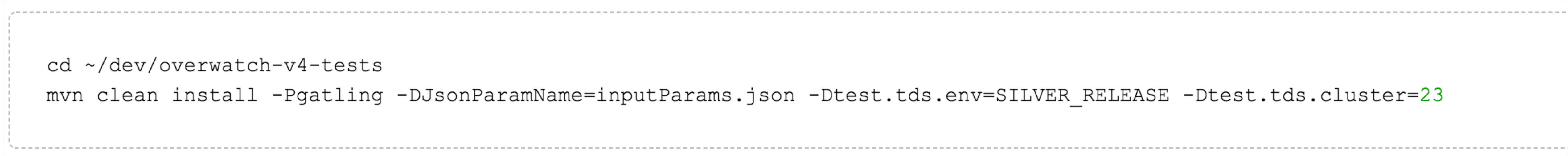
Perf Test - Gatling



- Run a sample test:

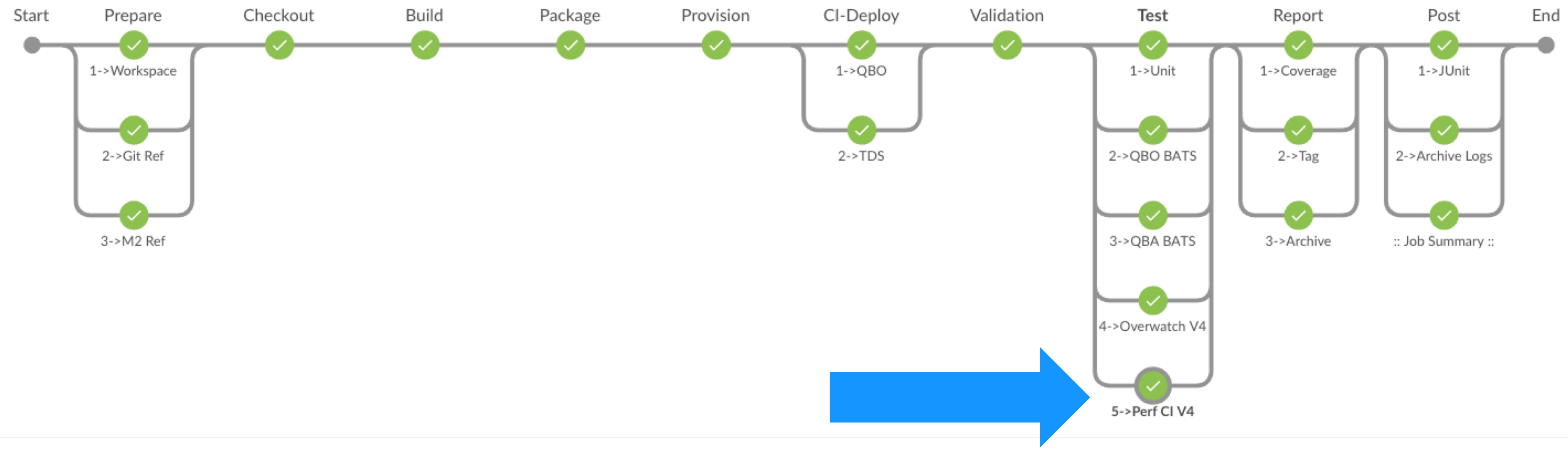
```
cd ~/dev/gatling-maven-perf
mvn -U -s ./settings.xml clean install -Pperf_dev -Dgatling.simulationClass=v4ListTermScenario -DuserDuration=1 -DnoOfUsers=1
```

Single Test – Perf / QA

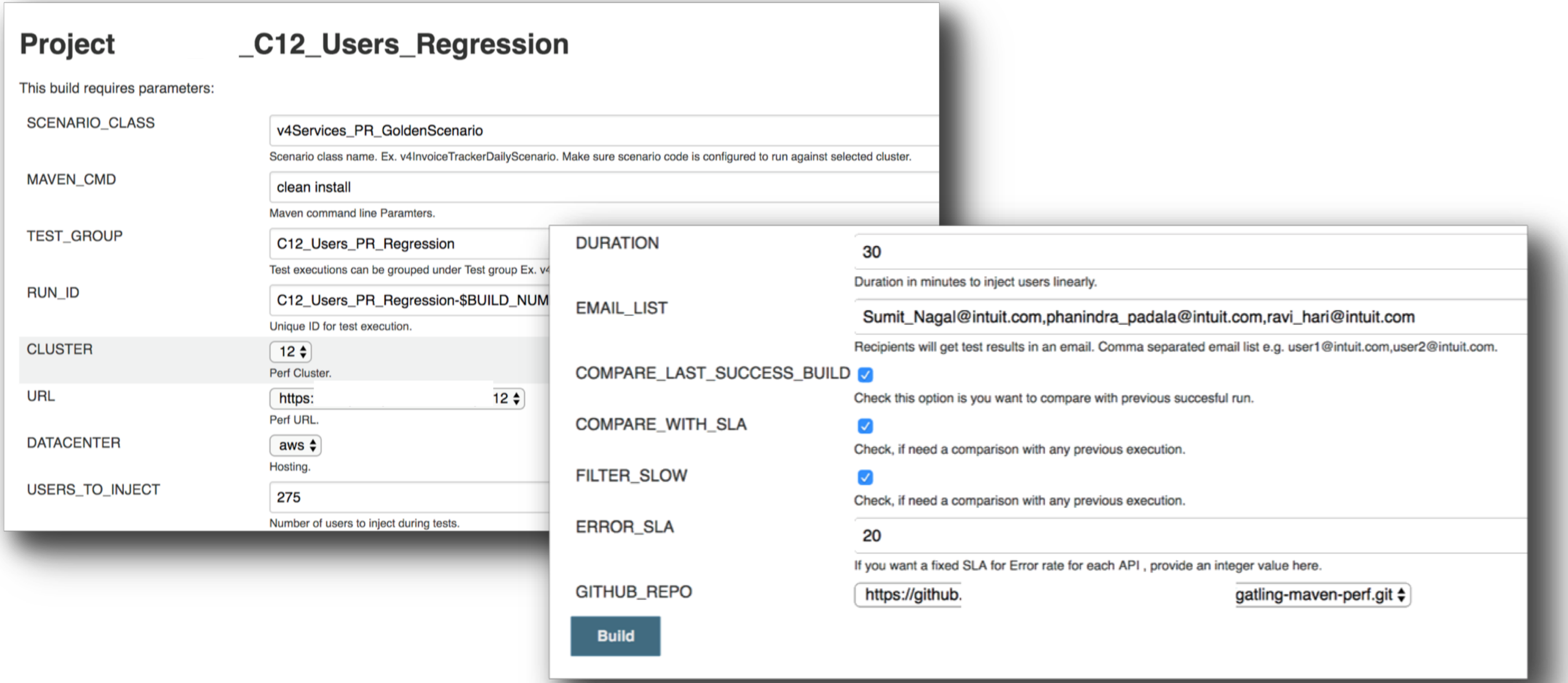


```
cd ~/dev/overwatch-v4-tests
mvn clean install -Pgatling -DJsonParamName=inputParams.json -Dtest.tds.env=SILVER_RELEASE -Dtest.tds.cluster=23
```

Performance feedback in the CI/CD pipeline



Easy to integrate into Jenkins pipeline



Project _C12_Users_Regression

This build requires parameters:

SCENARIO_CLASS: v4Services_PR_GoldenScenario
Scenario class name. Ex. v4InvoiceTrackerDailyScenario. Make sure scenario code is configured to run against selected cluster.

MAVEN_CMD: clean install
Maven command line Paramters.

TEST_GROUP: C12_Users_PR_Regression
Test executions can be grouped under Test group Ex. v4

RUN_ID: C12_Users_PR_Regression-$BUILD_NUM
Unique ID for test execution.

CLUSTER: 12
Perf Cluster.

URL: https: 12
Perf URL.

DATACENTER: aws
Hosting.

USERS_TO_INJECT: 275
Number of users to inject during tests.

DURATION: 30
Duration in minutes to inject users linearly.

EMAIL_LIST: Sumit_Nagal@intuit.com,phanindra_padala@intuit.com,ravi_hari@intuit.com
Recipients will get test results in an email. Comma separated email list e.g. user1@intuit.com,user2@intuit.com.

COMPARE_LAST_SUCCESS_BUILD ☑
Check this option is you want to compare with previous succesful run.

COMPARE_WITH_SLA ☑
Check, if need a comparison with any previous execution.

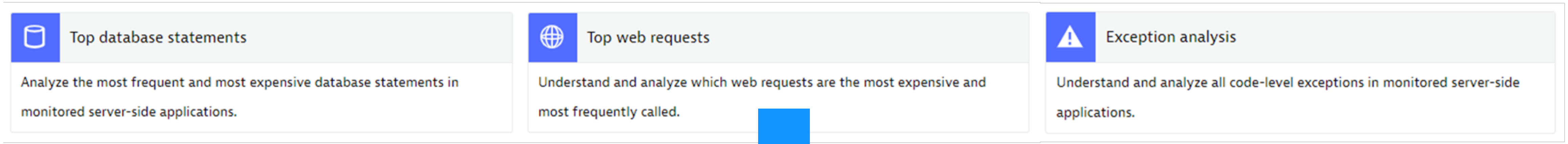
FILTER_SLOW ☑
Check, if need a comparison with any previous execution.

ERROR_SLA: 20
If you want a fixed SLA for Error rate for each API , provide an integer value here.

GITHUB_REPO: https://github. gatling-maven-perf.git

Dynatrace Tip: Load Testing Integration – automatic comparison & hotspot analysis

dynatrace Perform



Top database statements

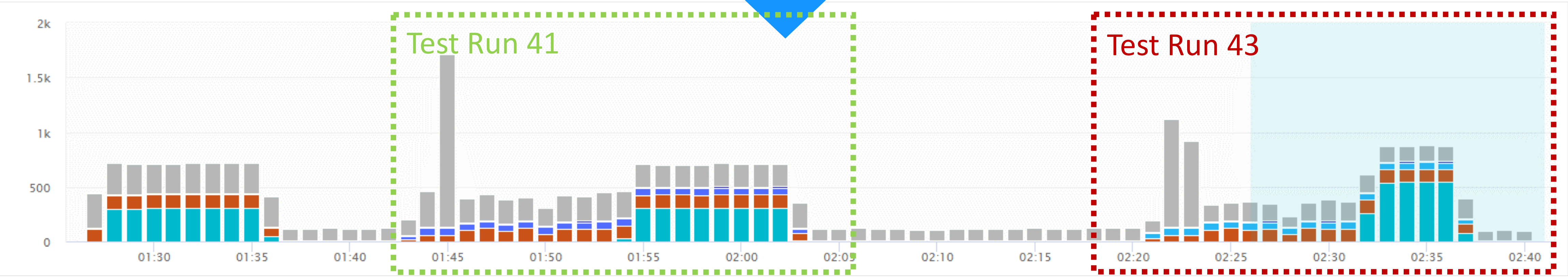
Analyze the most frequent and most expensive database statements in monitored server-side applications.

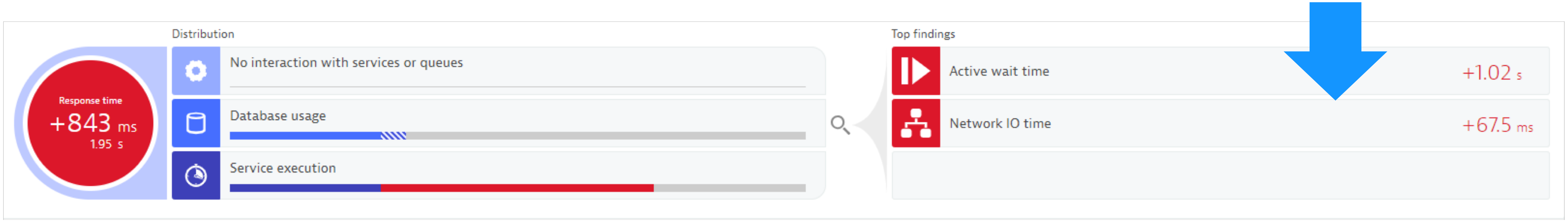
Top web requests

Understand and analyze which web requests are the most expensive and most frequently called.

Exception analysis

Understand and analyze all code-level exceptions in monitored server-side applications.





Response time +843 ms 1.95 s

Distribution

- No interaction with services or queues
- Database usage
- Service execution

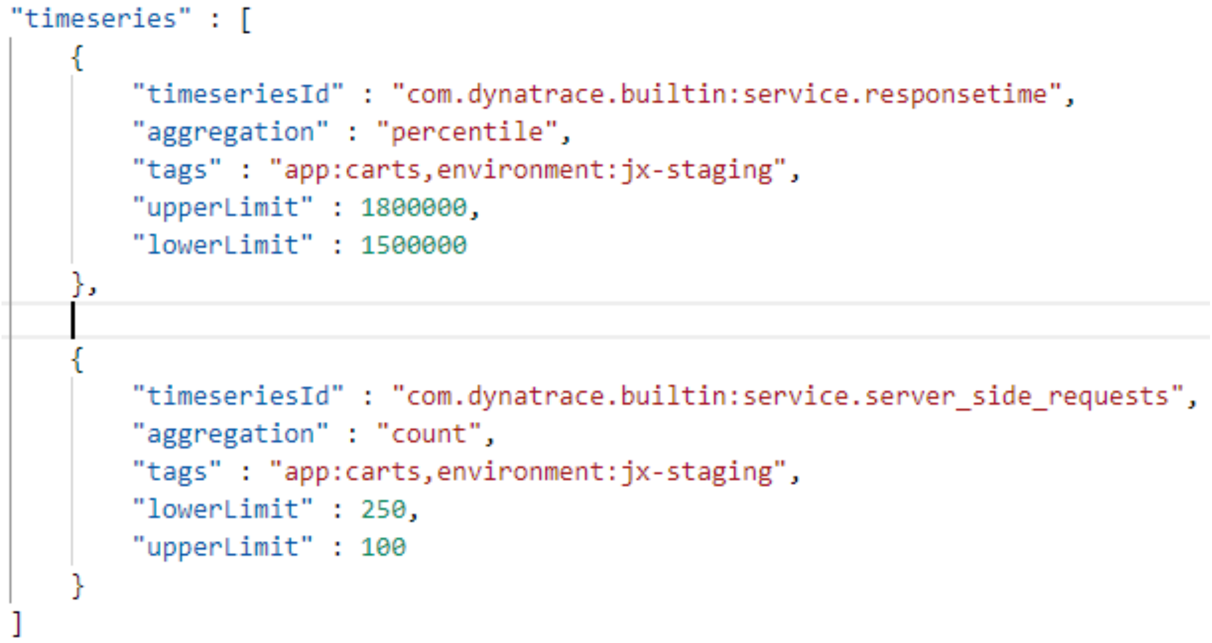
Top findings

Active wait time	+1.02 s
Network IO time	+67.5 ms

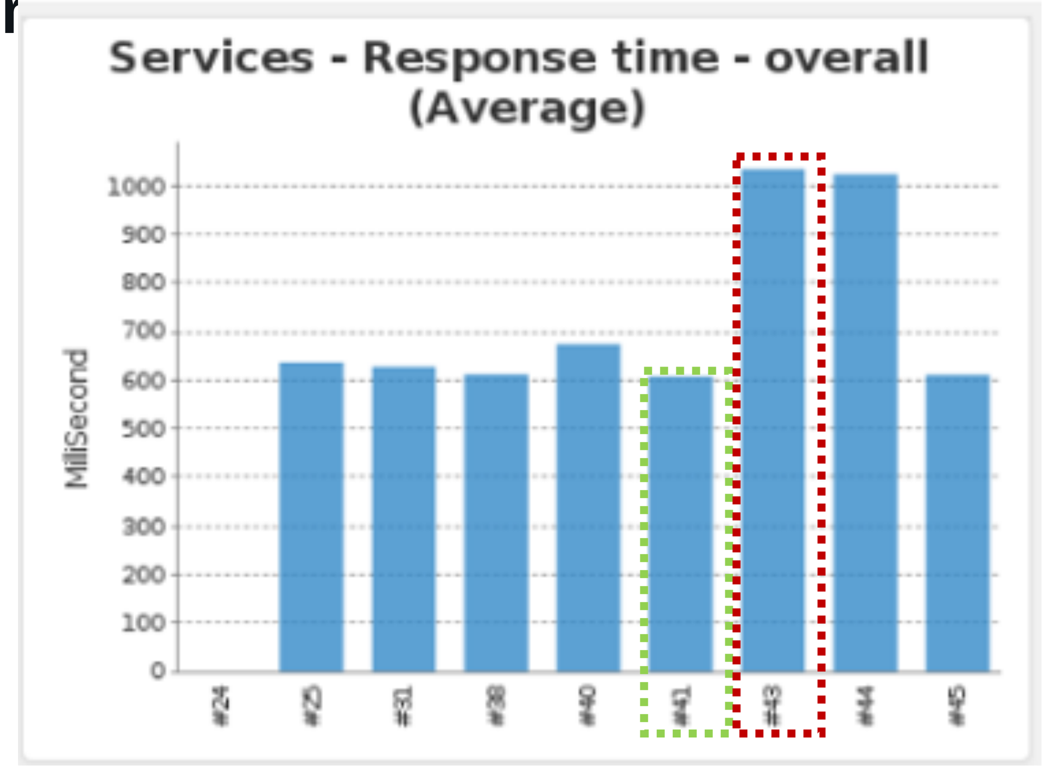
APACHE JMeter™ · Neotys · LOAD RUNNER · BlazeMeter High performance software. Delivered. · Gatling LOAD TESTING

Dynatrace Tip: Integrate performance signature

- Leverage “Monitoring as Code” (=Performance Signature) Approach
  - Automate Validation of Key Metr



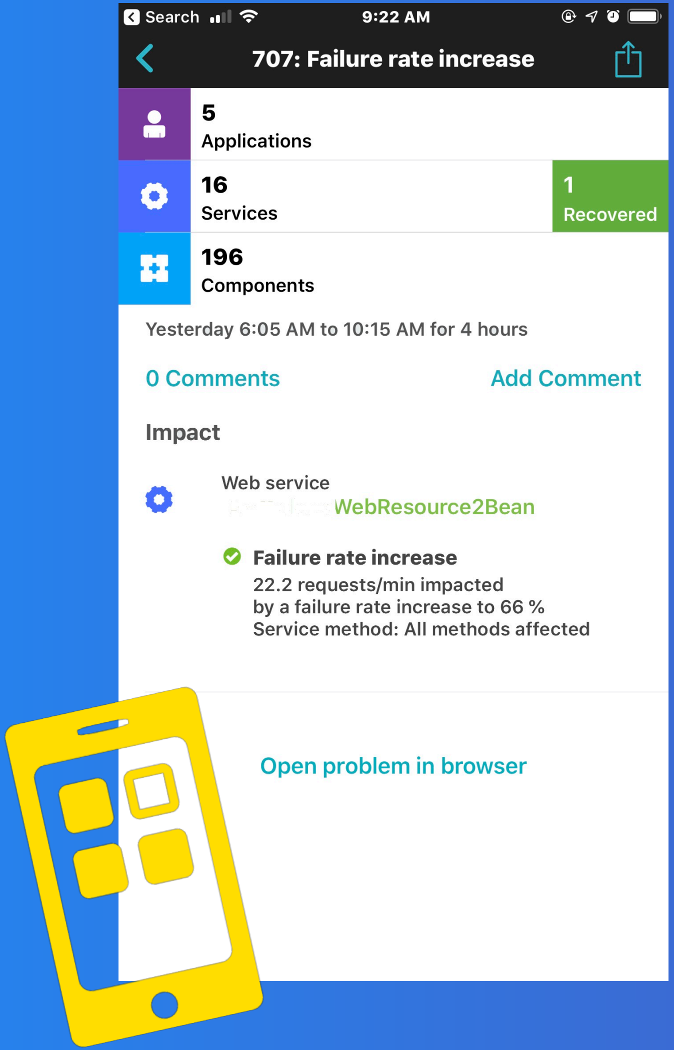
```
"timeseries" : [
    {
        "timeseriesId" : "com.dynatrace.builtin:service.responsetime",
        "aggregation" : "percentile",
        "tags" : "app:carts,environment:jx-staging",
        "upperLimit" : 1800000,
        "lowerLimit" : 1500000
    },

    {
        "timeseriesId" : "com.dynatrace.builtin:service.server_side_requests",
        "aggregation" : "count",
        "tags" : "app:carts,environment:jx-staging",
        "lowerLimit" : 250,
        "upperLimit" : 100
    }
]
```



Services - Response time - overall (Average)

Final remarks

Integrations we leverage



707: Failure rate increase

5 Applications

16 Services

1 Recovered

196 Components

Yesterday 6:05 AM to 10:15 AM for 4 hours

0 Comments   Add Comment

Impact

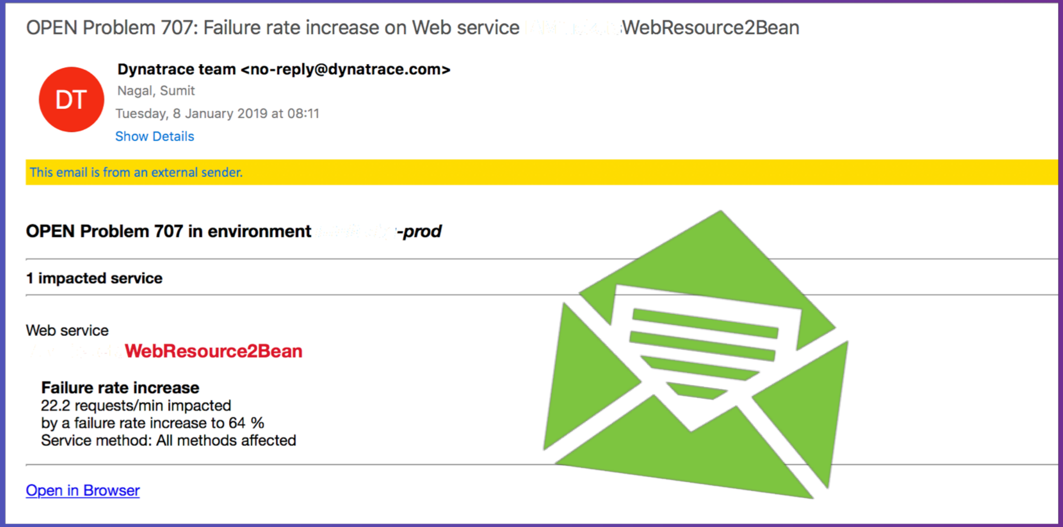
Web service
[redacted]WebResource2Bean

Failure rate increase
22.2 requests/min impacted
by a failure rate increase to 66 %
Service method: All methods affected

Open problem in browser

SSO
Single Sign On

Jira


OPEN Problem 707: Failure rate increase on Web service [redacted]WebResource2Bean

DT
Dynatrace team <no-reply@dynatrace.com>
Nagal, Sumit
Tuesday, 8 January 2019 at 08:11
Show Details

This email is from an external sender.

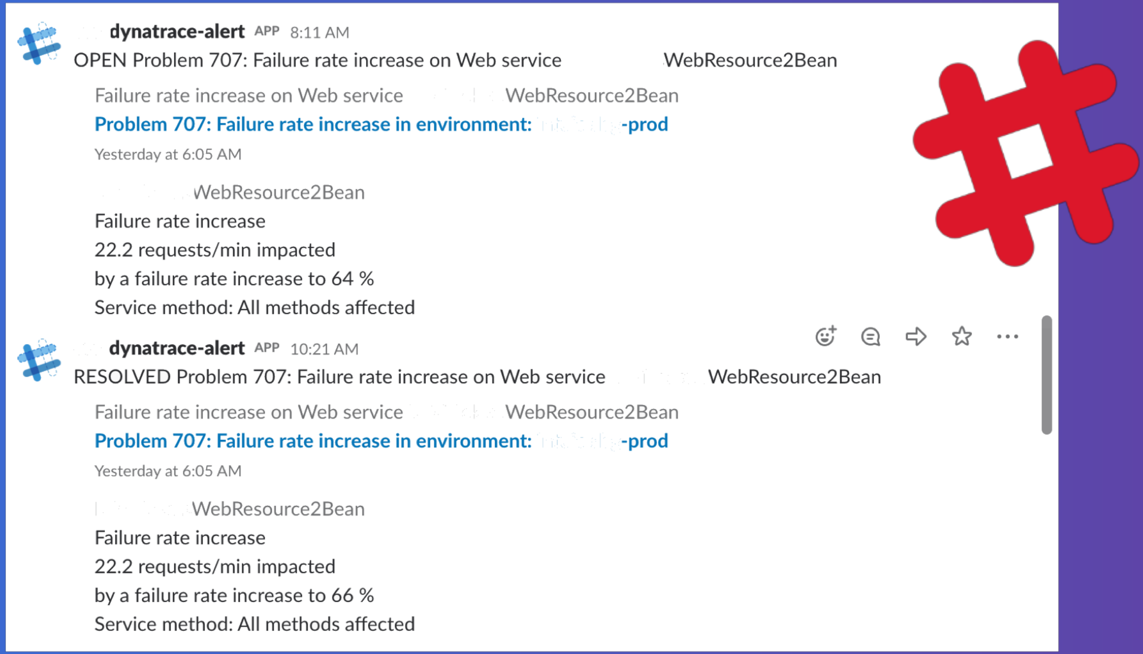
OPEN Problem 707 in environment [redacted]***-prod***

1 impacted service

Web service
[redacted]**WebResource2Bean**

Failure rate increase
22.2 requests/min impacted
by a failure rate increase to 64 %
Service method: All methods affected

Open in Browser


dynatrace-alert APP 8:11 AM
OPEN Problem 707: Failure rate increase on Web service [redacted] WebResource2Bean

Failure rate increase on Web service [redacted]WebResource2Bean
Problem 707: Failure rate increase in environment: [redacted]**-prod**
Yesterday at 6:05 AM

[redacted]WebResource2Bean
Failure rate increase
22.2 requests/min impacted
by a failure rate increase to 64 %
Service method: All methods affected

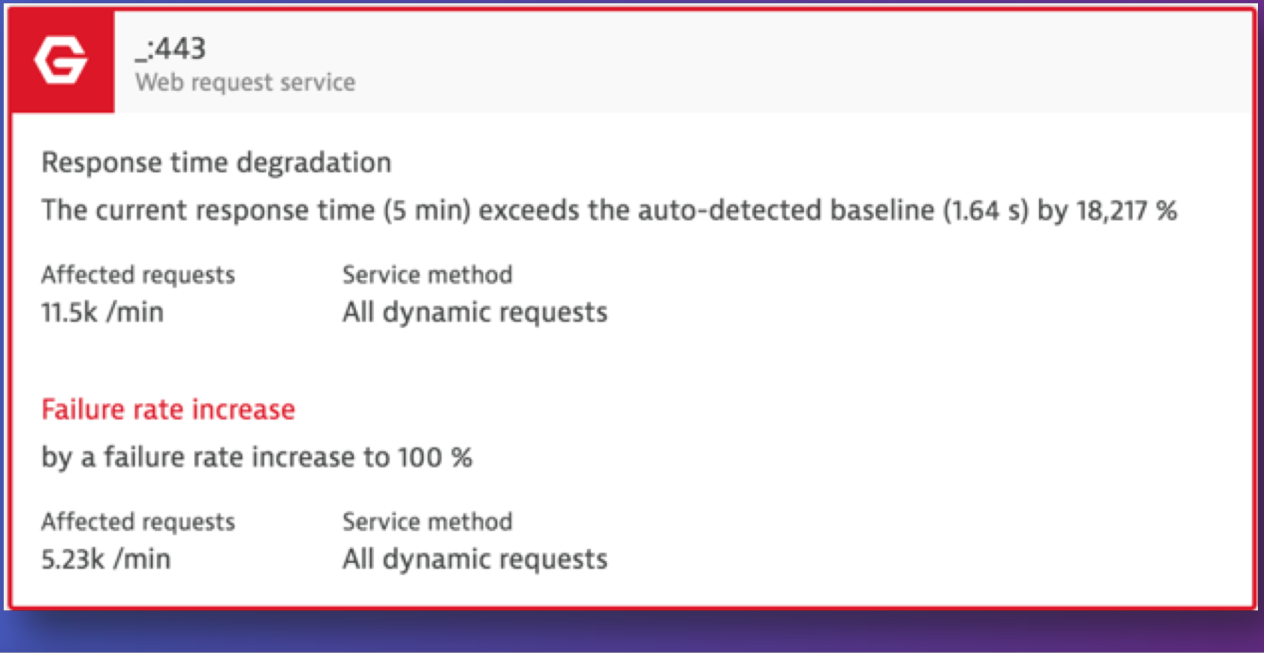
dynatrace-alert APP 10:21 AM
RESOLVED Problem 707: Failure rate increase on Web service [redacted] WebResource2Bean

Failure rate increase on Web service [redacted]WebResource2Bean
Problem 707: Failure rate increase in environment: [redacted]**-prod**
Yesterday at 6:05 AM

[redacted]WebResource2Bean
Failure rate increase
22.2 requests/min impacted
by a failure rate increase to 66 %
Service method: All methods affected

Benefits

- Self-Service Problem Analysis
- Comparison of Two Executions
- Tracing Single PurePath
- Tagging
- Management Zones
- Root Cause Analysis
- Memory Dumps
- AI / ML
- Docker / k8s …



_:443
Web request service

Response time degradation

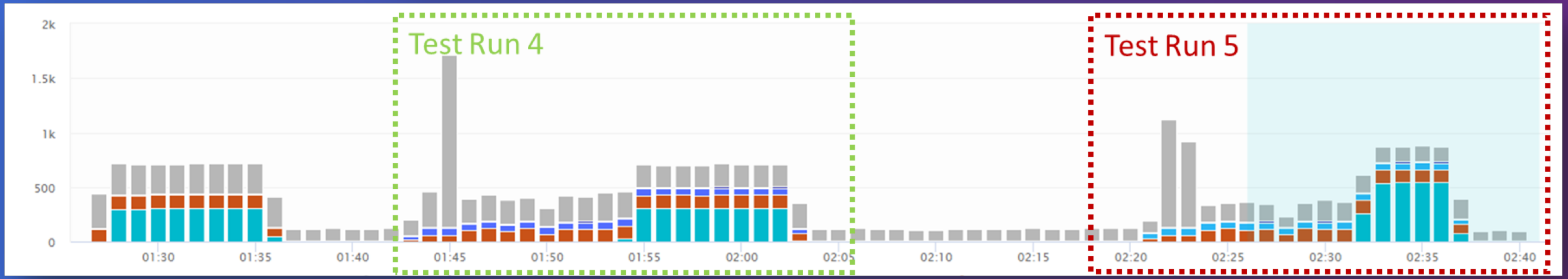
The current response time (5 min) exceeds the auto-detected baseline (1.64 s) by 18,217 %

Affected requests	Service method
11.5k /min	All dynamic requests

Failure rate increase

by a failure rate increase to 100 %

Affected requests	Service method
5.23k /min	All dynamic requests



Next step (opportunities)

- SSO for Saas
- Subdomain Support
- Tagging for Public and Monitoring End Point
- Control on Upgrade with Self Sign
- Memory Dump setup via Gateway (Size)
- Not Search Base on Request Attribute
- Anomaly Customization (Severity)
- Top Requests
- Alert Noise – Green / Blue

Call to action

- Performance Clinic: Advanced Diagnostics with Dynatrace
  - https://www.youtube.com/watch?v=tYuvpAw6ZYg
- Performance Clinic: Unbreakable Delivery Pipeline: Shift-Left, Shift-Right, Self-Healing
  - https://www.youtube.com/watch?v=r-0xunTqwQ0
- Dynatrace and load testing tools integration
  - https://www.dynatrace.com/support/help/integrations/third-party/test-automation/dynatrace-and-load-testing-tools-integration/
- Fill out the Autonomous Cloud Survey in the PERFORM app
  - https://dynatrace.ai/acsurvey

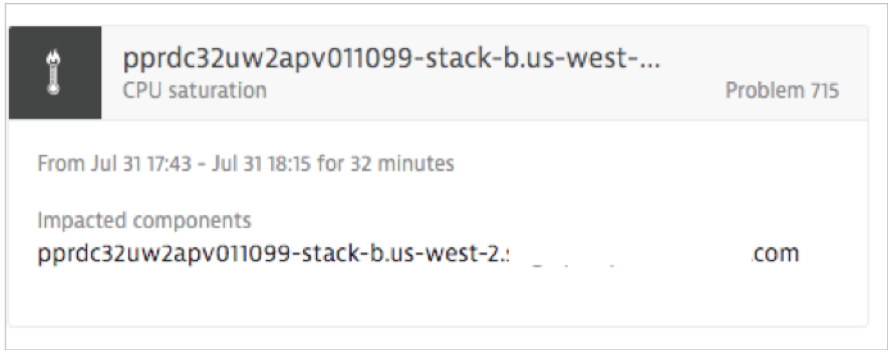
Thank you

dynatrace

Perform

Additional use cases

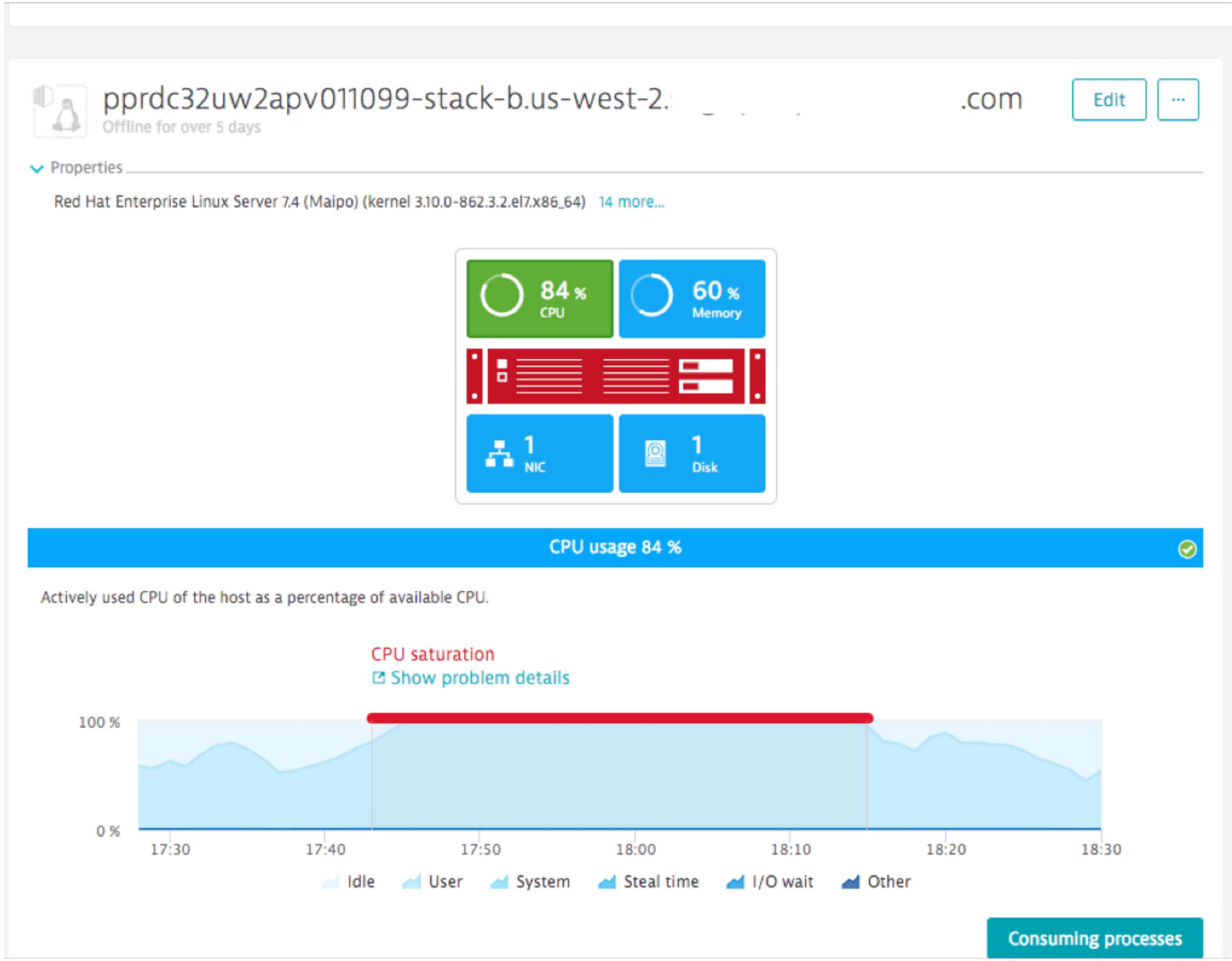
Performance – CPU issue



pprdc32uw2apv011099-stack-b.us-west-...
CPU saturation
Problem 715

From Jul 31 17:43 - Jul 31 18:15 for 32 minutes

Impacted components
pprdc32uw2apv011099-stack-b.us-west-2.: _ . . .com



pprdc32uw2apv011099-stack-b.us-west-2. _ . . .com
Offline for over 5 days

Properties

Red Hat Enterprise Linux Server 7.4 (Maipo) (kernel 3.10.0-862.3.2.el7.x86_64) 14 more...

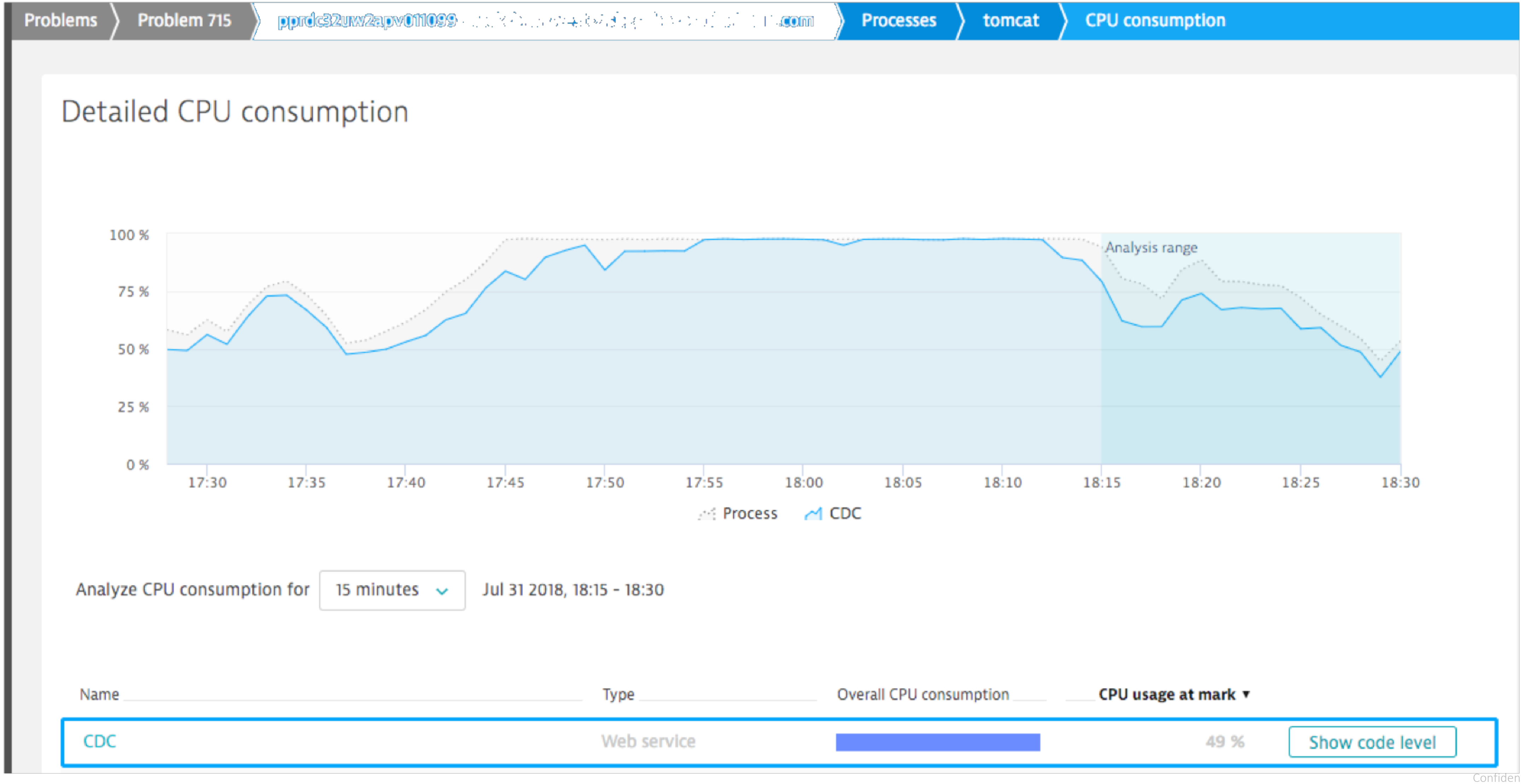
CPU usage 84 %

Actively used CPU of the host as a percentage of available CPU.

CPU saturation
Show problem details

dynatrace
Perform

Performance – CPU issue



Problems 〉 Problem 715 〉 pprdc32uw2apv011099[redacted].com 〉 Processes 〉 tomcat 〉 CPU consumption

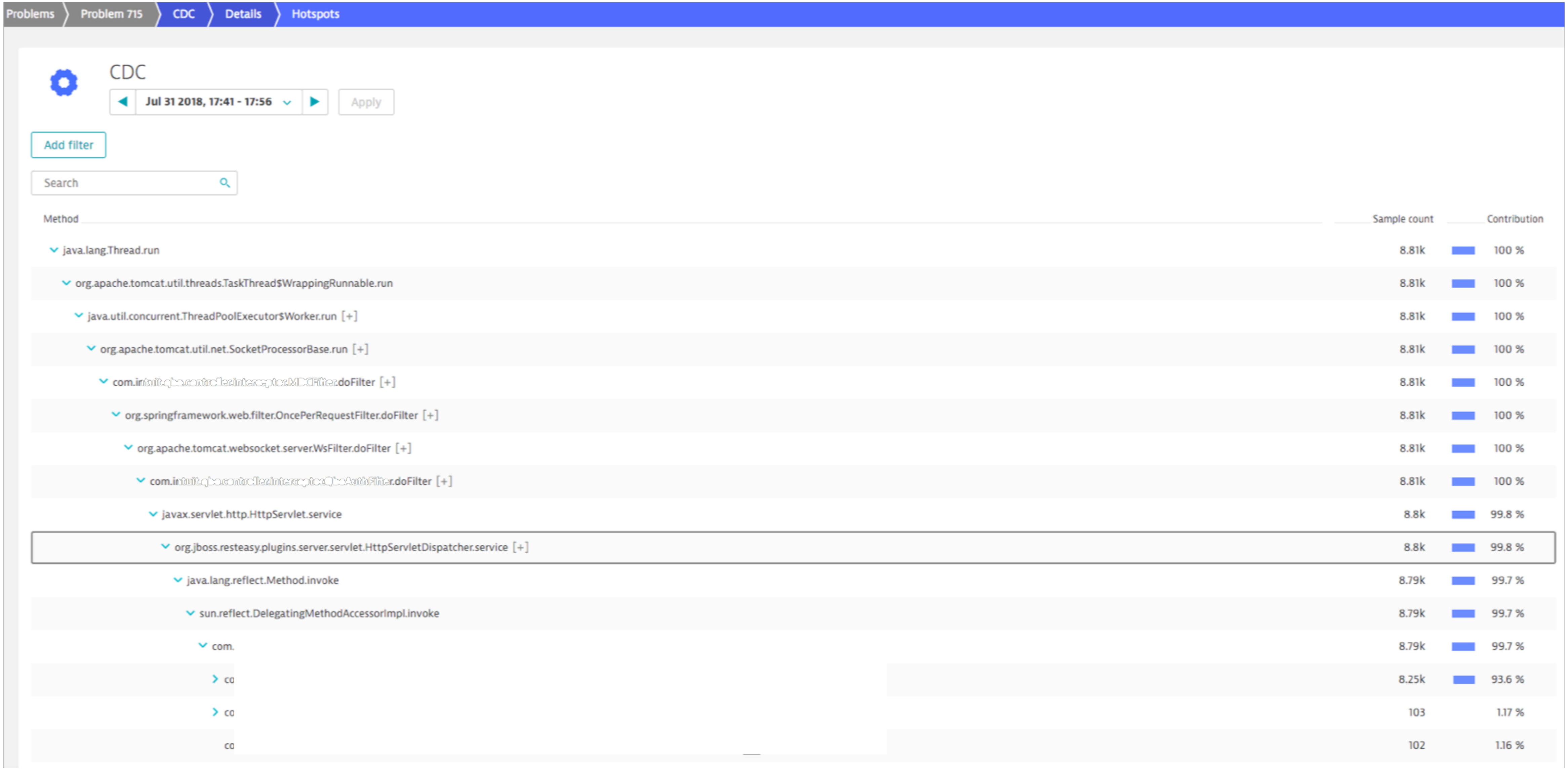
Detailed CPU consumption

Analyze CPU consumption for 15 minutes ⌄ Jul 31 2018, 18:15 - 18:30

Name	Type	Overall CPU consumption	CPU usage at mark ▾	
CDC	Web service	▬	49 %	Show code level

dynatrace
Perform

Performance – CPU issue



Problems 〉 Problem 715 〉 CDC 〉 Details 〉 Hotspots

CDC

◀ Jul 31 2018, 17:41 - 17:56 ⌄ ▶ Apply

Add filter

Search

Method	Sample count	Contribution
java.lang.Thread.run	8.81k	100 %
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run	8.81k	100 %
java.util.concurrent.ThreadPoolExecutor$Worker.run [+]	8.81k	100 %
org.apache.tomcat.util.net.SocketProcessorBase.run [+]	8.81k	100 %
com.in[redacted].doFilter [+]	8.81k	100 %
org.springframework.web.filter.OncePerRequestFilter.doFilter [+]	8.81k	100 %
org.apache.tomcat.websocket.server.WsFilter.doFilter [+]	8.81k	100 %
com.in[redacted].doFilter [+]	8.81k	100 %
javax.servlet.http.HttpServlet.service	8.8k	99.8 %
org.jboss.resteasy.plugins.server.servlet.HttpServletDispatcher.service [+]	8.8k	99.8 %
java.lang.reflect.Method.invoke	8.79k	99.7 %
sun.reflect.DelegatingMethodAccessorImpl.invoke	8.79k	99.7 %
com.[redacted]	8.79k	99.7 %
co[redacted]	8.25k	93.6 %
co[redacted]	103	1.17 %
co[redacted]	102	1.16 %