

Dynatrace Managed

Entwickler Schulung

Patrick Hofmann

Agenda Mittwoch – 22.01.20

- | Überblick Dynatrace
- | Architektur / OneAgent
- | UI Basics
- | Dynatrace Topology Model and Smartscape
- | User Experience / Frontend Dashboards
- | Model Details / Transactions and Services / Backend Dashboards
- | Cloud / Virtualization Monitoring
- | Synthetic Monitors

Agenda Donnerstag – 23.01.20

- | Tagging
- | Management Zones
- | Monitoring Settings
- | RUM Settings
- | Service Settings
- | Erweiterung: Dynatrace API & Plugins

Dynatrace Überblick



Application Performance
Management



Browser



Mobile*



3rd parties



Application- & Webserver



Services



Containers, Processes,
Logs



OS, Disks



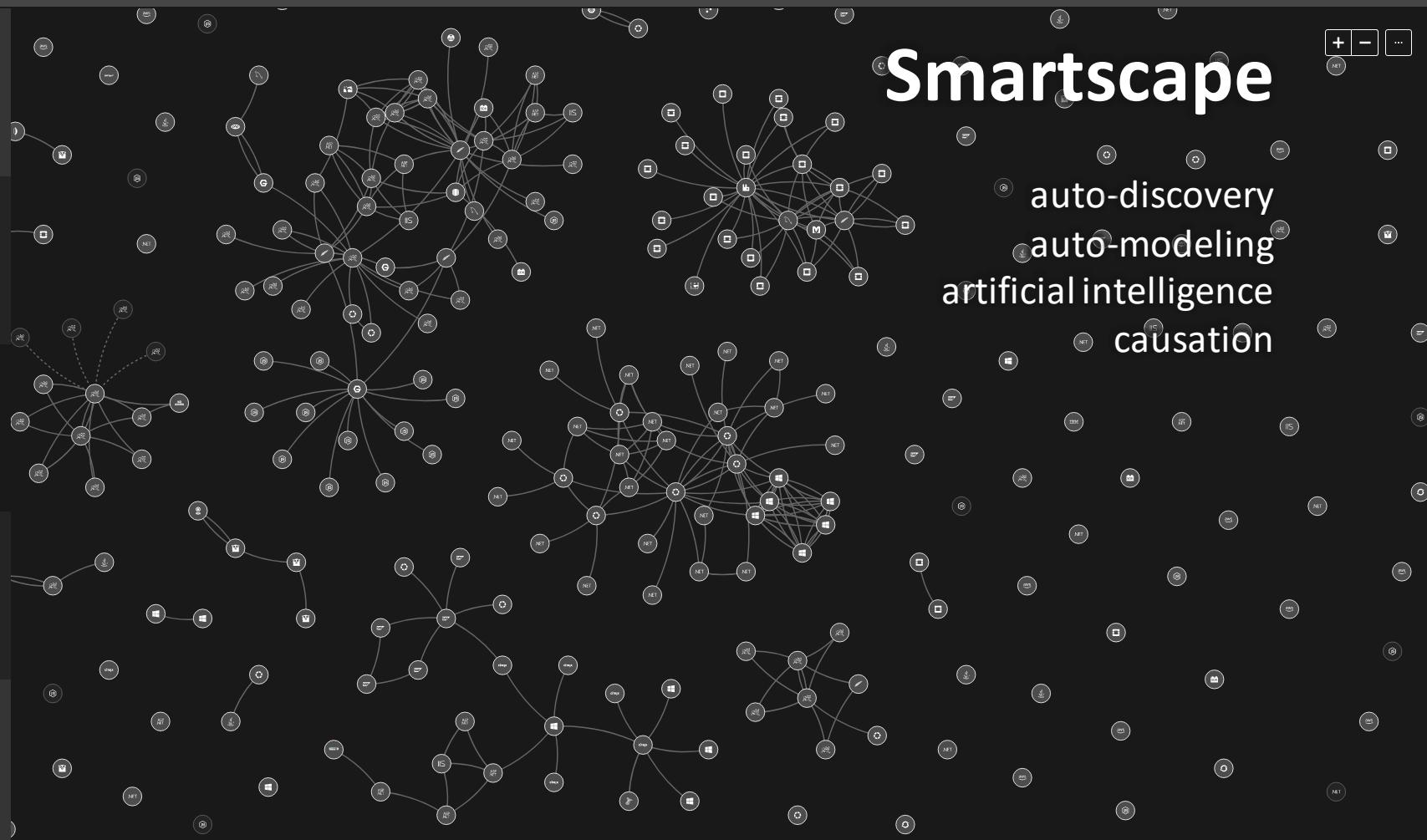
Cloud



Network



FULL STACK

Applications
20 Services
279 Processes
477 Hosts
105 Data centers
15

Smartscape

auto-discovery
auto-modeling
artificial intelligence
causation



Problems tend to become complex

3 applications: User action duration degradation
Problem 809 detected at 07:59 - 08:54 (was open for 55 minutes). This problem affects real users.

Affected applications 3 Affected services 17 Affected infrastructure 3 924,973,350 Dependencies analyzed

Business impact analysis
An analysis of all affected service calls and impacted real users during the first 10 minutes of the problem shows the following potential impact.

178 Impacted users (show first 100) 528k Affected service calls Show more

3 impacted applications
539 User actions per minute impacted

www.angular.easytravel.com Application

User action duration degradation
The current response time (24.7 s) exceeds the auto-detected baseline (401 ms) by 6,072 %

Affected user actions	User action
196 /min	2 User actions
Browser All	Geolocation All
	OS All

Root cause
Based on our dependency analysis all incidents have the same root cause

BB1-apache-tomcatjms-iis Host

CPU saturation
100 % CPU usage Analyze logs

CheckDestination Custom service

Response time degradation
The current response time (5.33 s) exceeds the auto-detected baseline (131 ms) by 3,980 %

Affected requests	Service method
636 /min	2 Service methods

Problems tend to become complex

3 applications: User action duration degradation
Problem 809 detected at 07:59 - 08:54 (was open for 55 minutes). This problem affects real users.

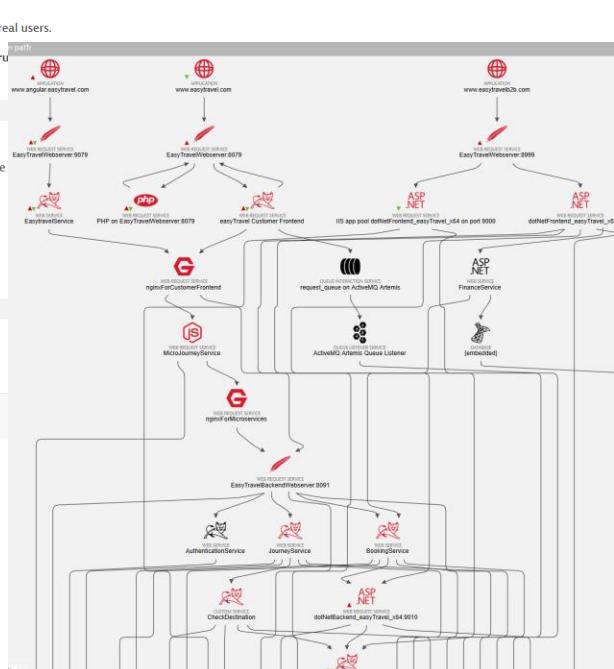
Affected applications: 3 | Affected services: 17 | Affected infrastructure: 3

Business impact analysis
An analysis of all affected service calls and impacted real users during the first 10 minutes of the impact.

178 Impacted users (show first 100) | 528k Affected service calls | Show more

3 impacted applications
539 User actions per minute impacted

www.angular.easytravel.com Application
User action duration degradation
The current response time (24.7 s) exceeds the auto-detected baseline (401 ms) by 6,072 %
Affected user actions: 196 / min | User action: 2 User actions
Browser: All | Geolocation: All | OS: All



924,973,350 Dependencies analyzed

Problem evolution
166 original events in 20 components
Head: 2019-08-08 08:19 + 30

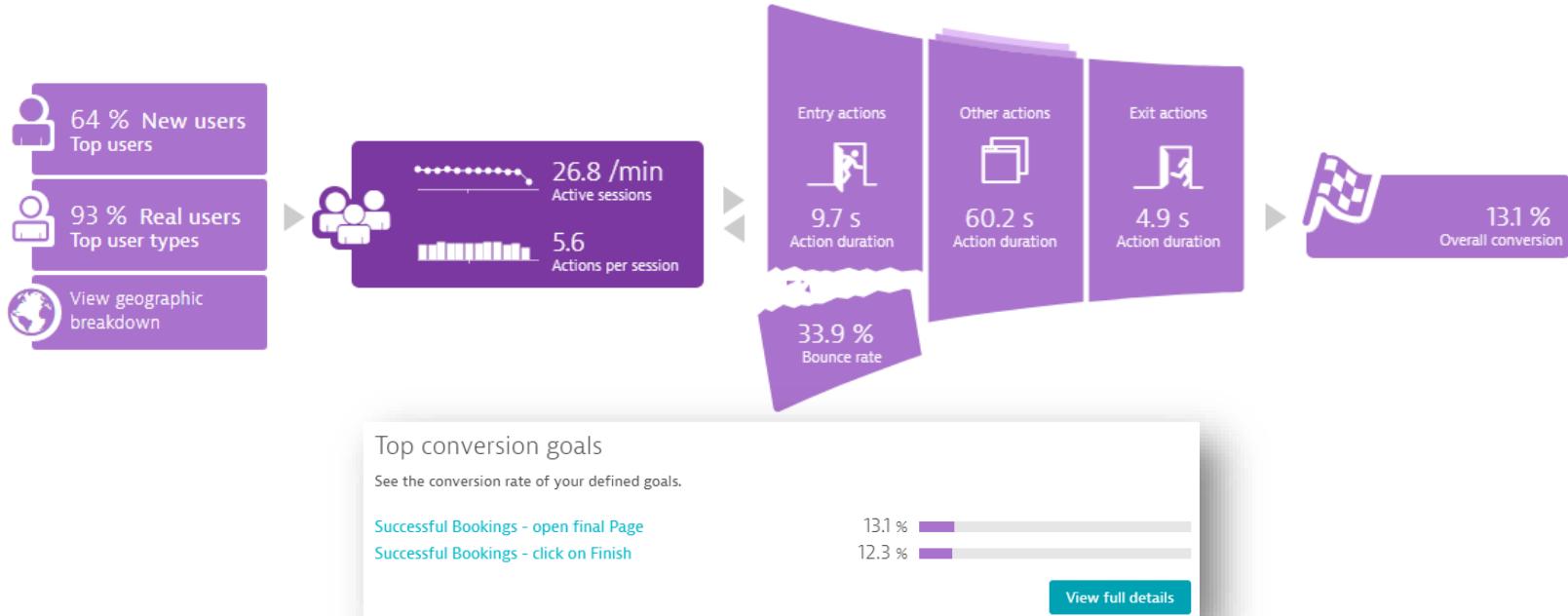
- www.angular.easytravel.com
 - User action duration degradation (45)
 - User action duration degradation (64)
- www.easytravel.com
 - User action duration degradation (64)
 - User action duration degradation (64)
- www.easytravel20.com
 - User action duration degradation (62)
- CustomerFrontend
 - Service method JourneyService.checkDestination slow down
 - Service method CheckDestination slow down
- CustomerBackend
 - All dynamic requests slow
 - Service method JourneyService/CheckDestination slow down
- CustomerBackend.easytravel_1d4 9000
 - All dynamic requests slow
 - Service method Booking slow down
 - Service method Images slow down
 - Service method LogOn slow down
 - All requests slow
 - All requests slow
 - All requests slow
 - Service method Report slow down
 - Service method Journey slow down
- CustomerFrontend.easytravel_1d4 9010
 - Service method JourneyService/Journey slow down
 - Service method JourneyService/Journey slow down
 - Service method JourneyService/Journey slow down

Highlights Dynatrace SaaS / Managed Infographics for Performance



Highlights Dynatrace SaaS / Managed

Infographics for User Behaviour



User Behaviour Analytics

Session Replay

Problems > Problem 604 > User sessions > Ashley McMullen

Impacted users by problem: 604 Filtered by

Ashley McMullen

Session details related to this user profile are outlined below.

Close problem analysis

1 Application accessed

performance-wear.perform-2018.dynalabs.io ...

London, United Kingdom

1 Browser and device type used

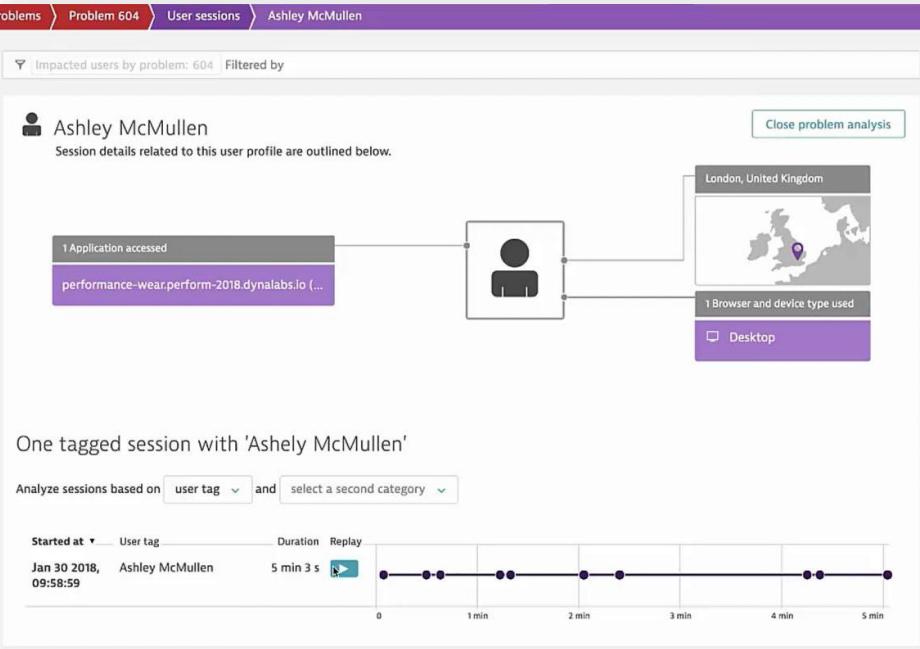
Desktop

One tagged session with 'Ashely McMullen'

Analyze sessions based on **user tag** and **select a second category**

Started at * User tag Duration Replay
Jan 30 2018, Ashley McMullen 5 min 3 s

0 1 min 2 min 3 min 4 min 5 min



Checkout

Camden Street 12A

City * London

State/Province

Zip/Postal Code AB12

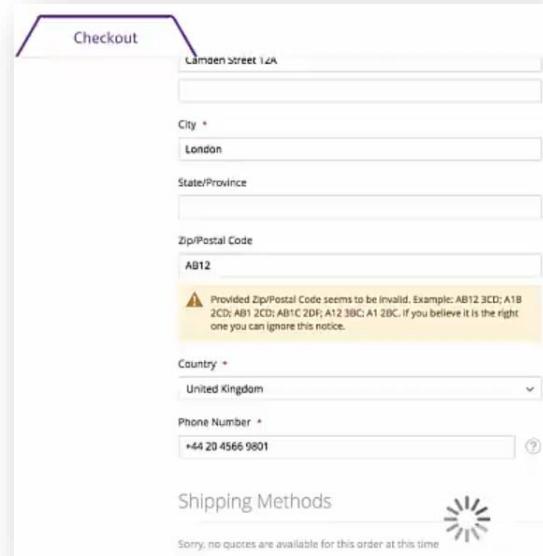
Provided Zip/Postal Code seems to be invalid. Example: AB12 3CD; A1B 2CD; AB1 2DF; A12 3BC; A1 2BC. If you believe it is the right one you can ignore this notice.

Country * United Kingdom

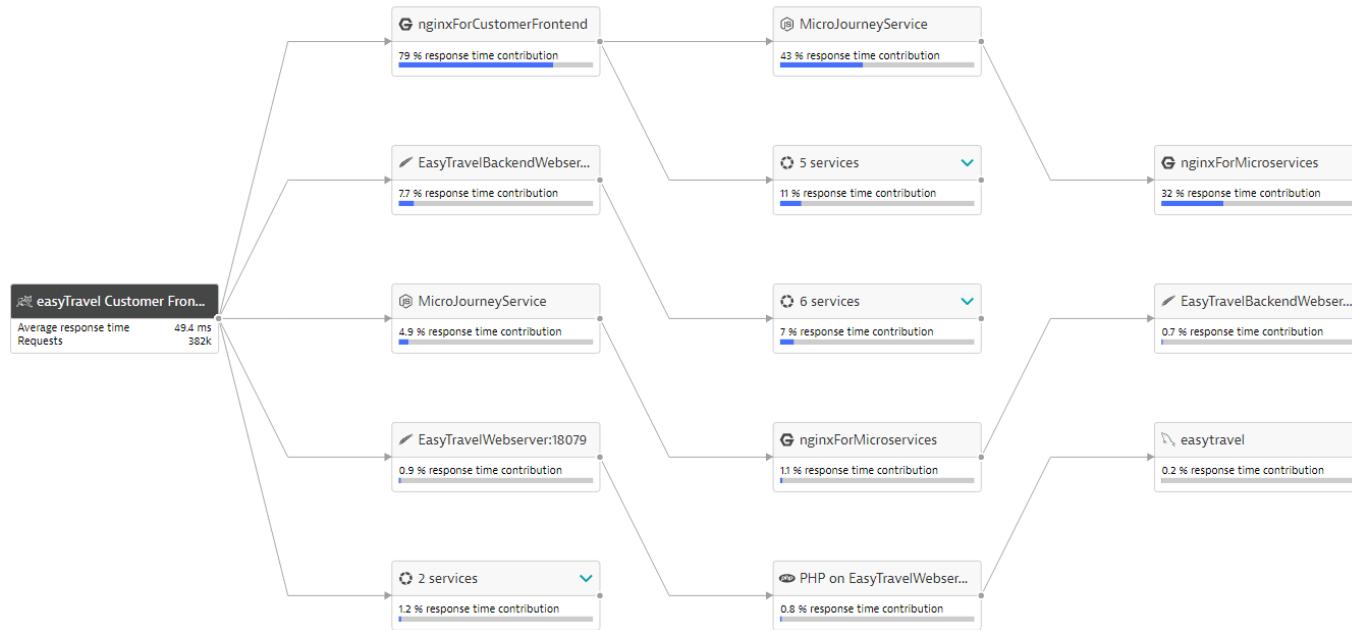
Phone Number * +44 20 4566 9801

Shipping Methods

Sorry, no quotes are available for this order at this time

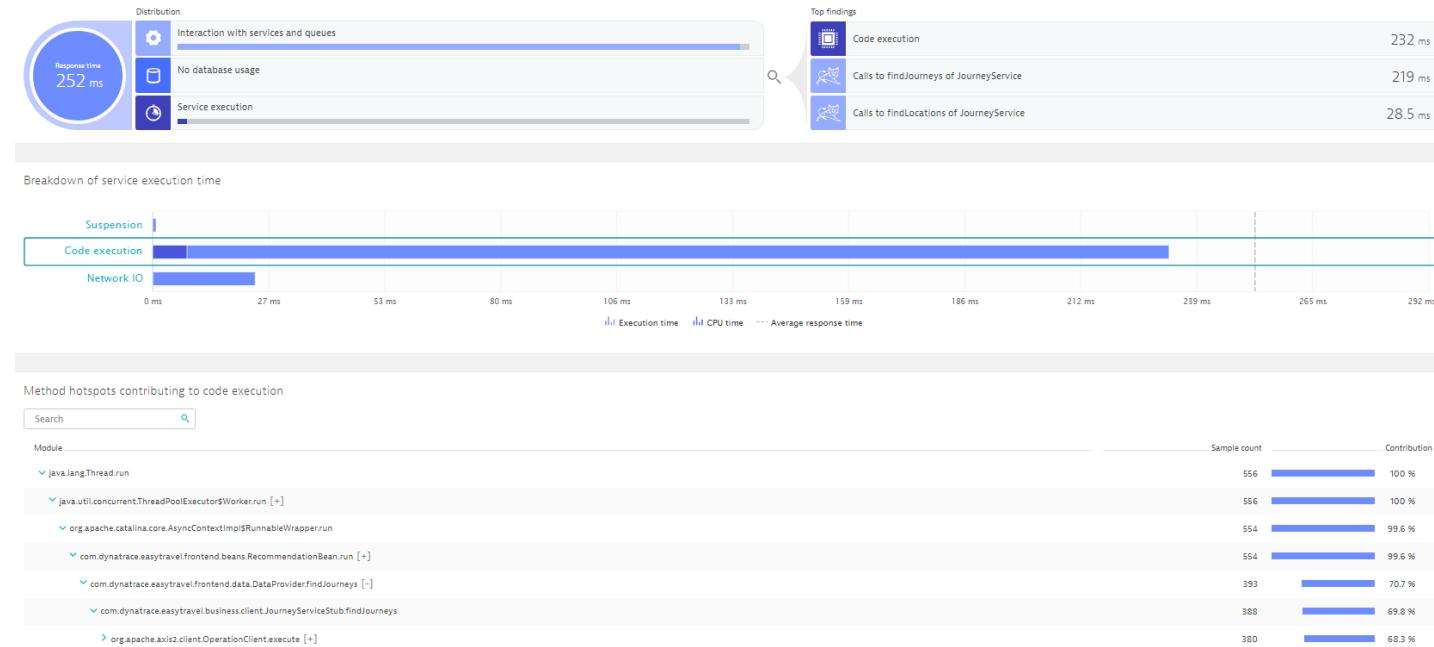


Highlights Dynatrace SaaS / Managed Understanding Dynamic Service Configurations

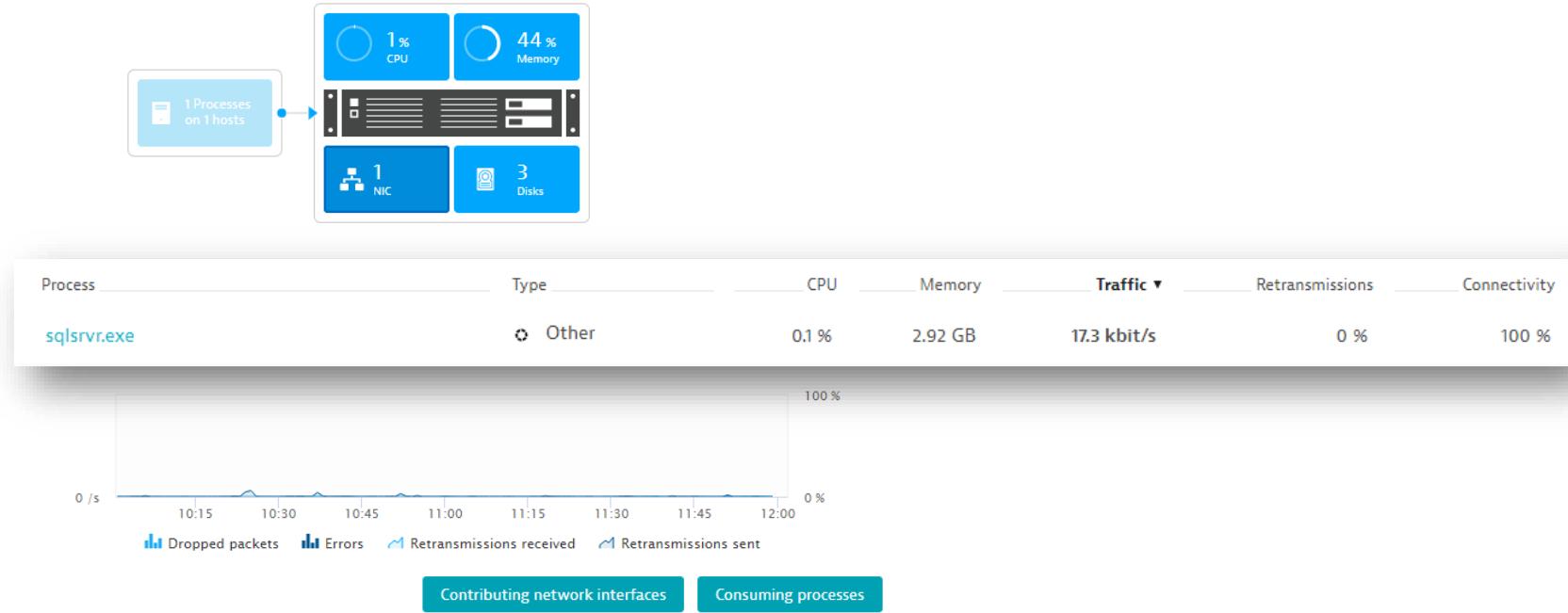


Highlights Dynatrace SaaS / Managed

Detailed Method-level Analysis



Highlights Dynatrace SaaS / Managed Infrastructure Monitoring including integrated Network Communication



Integrated Synthetic Monitoring

Synthetic User Experience / Service Monitoring

easyTravel demo booking
Clickpath web check running from 5 locations every 15 minutes

Pin to dashboard

22.50 % Availability 220.51 s Duration

Availability

22.50 % Availability	5 / 5 Locations	31 Total outages	2 h Total downtime
----------------------	-----------------	------------------	--------------------

Location	10:02	10:26	10:50	11:14	11:38	12:02	Average	Details
Singapore							25.00 %	
São Paulo							12.50 %	
Seattle							25.00 %	

9 Clickpath actions in the selected time frame

View action for All locations

Action	Action duration	Details
	Loading of "http://ec2-54-210-22-3.compute-1.amazonaws.com:8079/"	7.88 s
	click on "Book Now"	3.32 s
	click on "Next"	1.98 min
	click on "iceform:bookReviewNext"	21.2 s
	click on "iceform:fillMock"	35.4 s

< 1 2 >

Dynatrace Davis

Voice and Chat Ops for Production Environments

The screenshot shows the Dynatrace Davis Assistant interface with two cards displayed:

9:37: Global Outage for Synthetic Web Checks

Time Frame: Apr 8 at 8:34 AM - Apr 8 at 8:43 AM
Affected Application: easyTravel demo booking (Synthetic monitor)
Affected Synthetic Actions: Loading of "http://ec2-54-84-118-202.compute-1.amazonaws.com:8079/"
Please select an item, or say 'next page'.
First Second Third

8:09: Degradation in User Action Response Time Root Cause

Time Frame: Apr 8 at 7:59 AM - Apr 8 at 8:54 AM
Affected Applications: www.easytravelb2b.com, www.easytravel.com, www.angular.easytravel.com
Response Time (p50): 6 seconds **Response Time (p90):** 36 seconds
User Action: Xhr **Browser:** Chrome 56
Operating System: All **Geolocation:** All
Root Causes:
B82-Apache-Tomcatjms-1is
- Saturated CPU
CheckDestination
- Decrease in Service Response Time
B81-Apache-Tomcatjms-1is
- Saturated CPU
Is there anything else I can help you with?
Back Add Comment

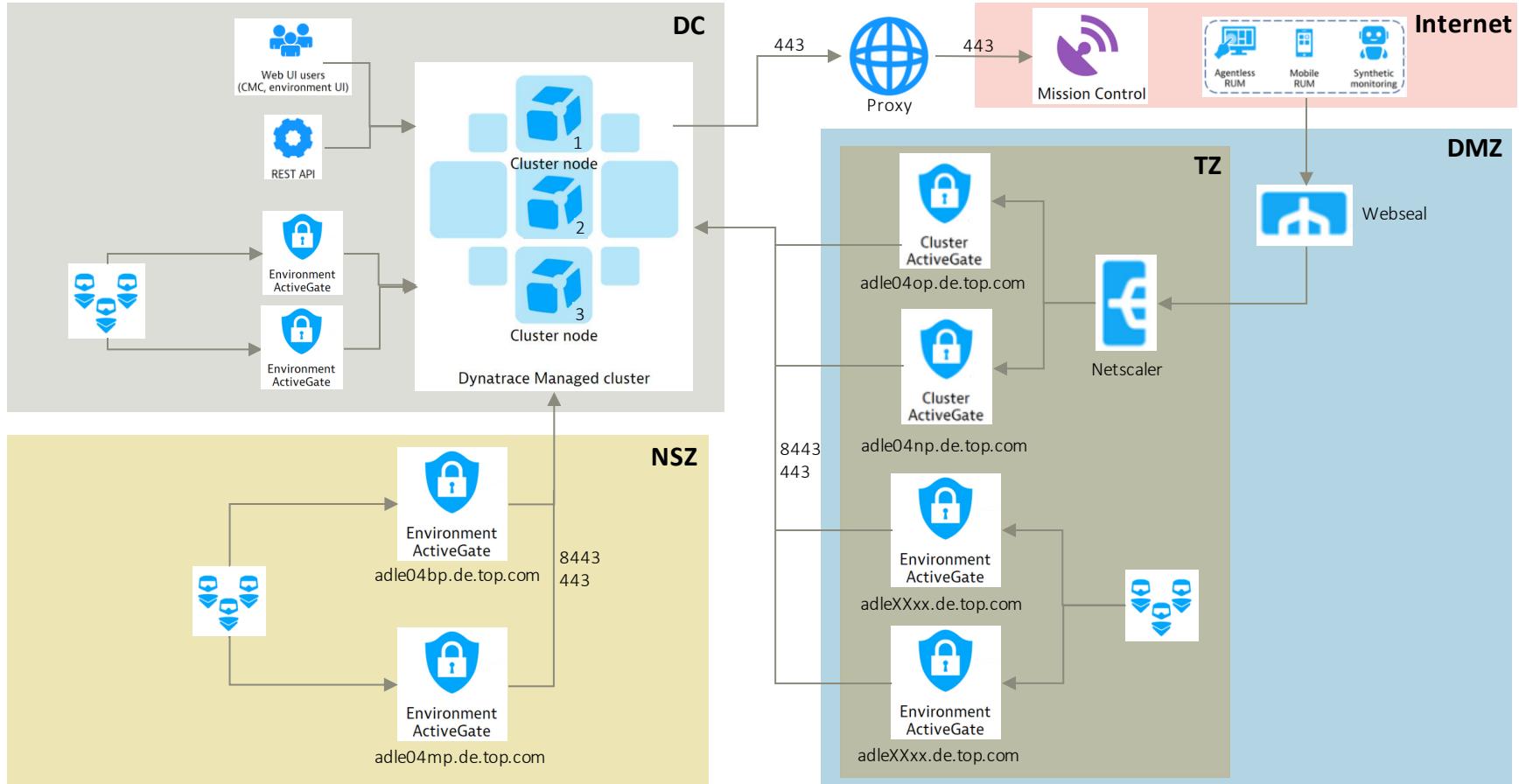
Interaction examples

In there any open problems? Try

Architecture



Application Performance
Management



Storage and Retention

- **Session Storage → file based**
 - Request Level – 35 days
 - Method Level – 10 Days
 - Depends on disk, quota can be defined for each environment
- **Time Series → Cassandra**
 - 1 minute intervals – 14 days
 - 5 minute intervals – 28 days
 - 1 hour intervals – 440 days
 - 1 day intervals – unlimited
- **Visits → Elasticsearch**
 - 35 days, can be extended
 - 10 days, detail-data like JS-Errors, Waterfall etc.

OneAgent



Application Performance
Management

One Agent to monitor them all



OneAgent

What's Included?

- Application Monitoring – Java, .NET, PHP, Node.js, Go
- Web Server Monitoring – Apache, IIS, Nginx
- Host Monitoring – CPU, Memory, Disk Utilization
- Process Monitoring
- Network Monitoring
- Log File Monitoring
- Plugin Execution
- Database Monitoring
- Third Party and Cloud Content
- Cloud and Container
- Real User Monitoring



UI Basics

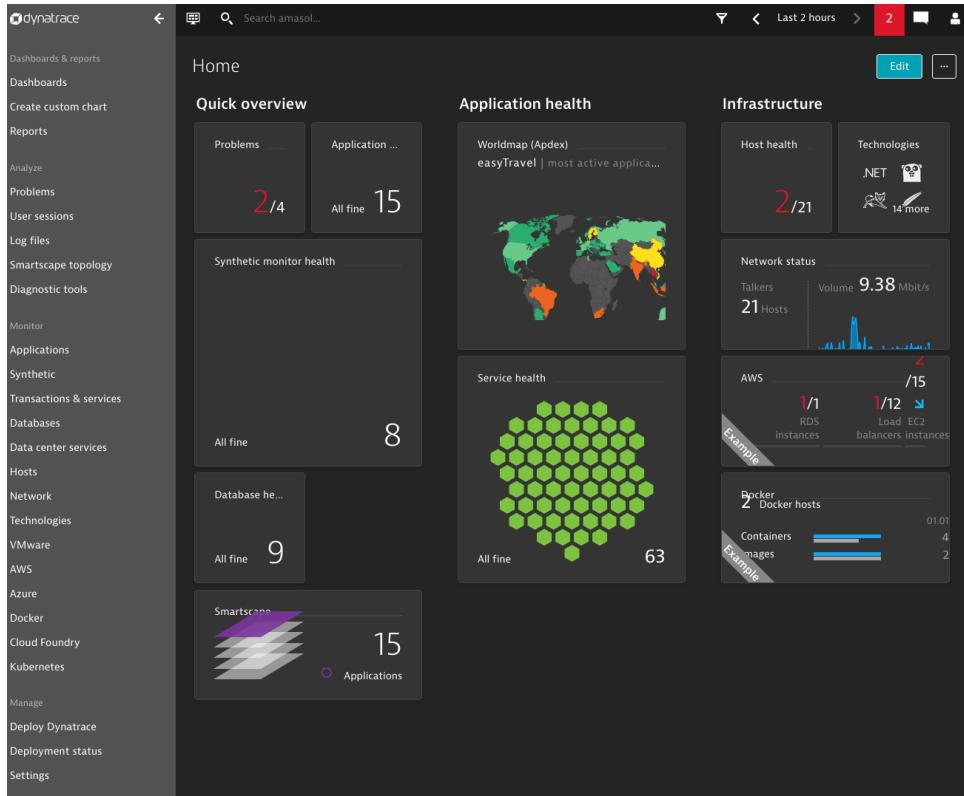


Application Performance
Management

UI Basics

Basic Navigation

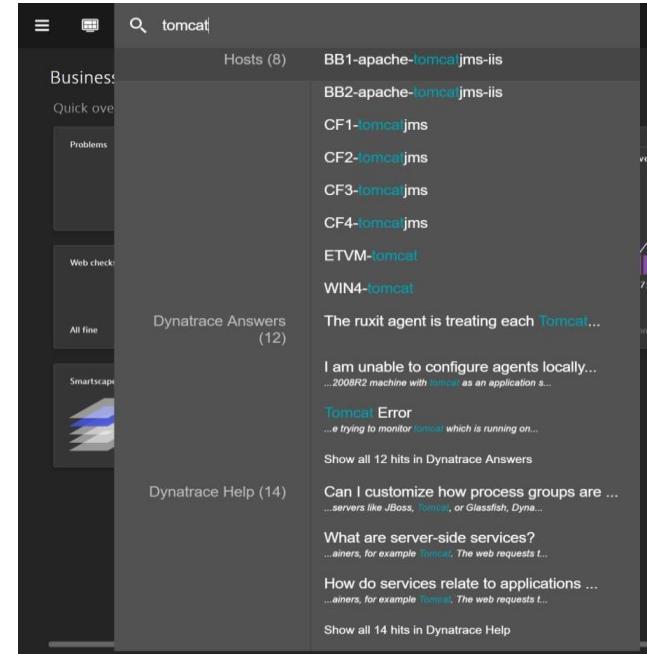
- **The UI consists of a side menu, a top menu, and the main body**
 - The body initially displays the last opened
- **The Time frame selector enables you to select a specific analysis time frame for each dashboard**
- **Out of the box dashboards can be accessed using the side menu or by drilling down through the Home Dashboard**
- **Custom dashboards can be created, accessed and shared using the Dashboard menu item from the Side**



UI Basics

Search Bar

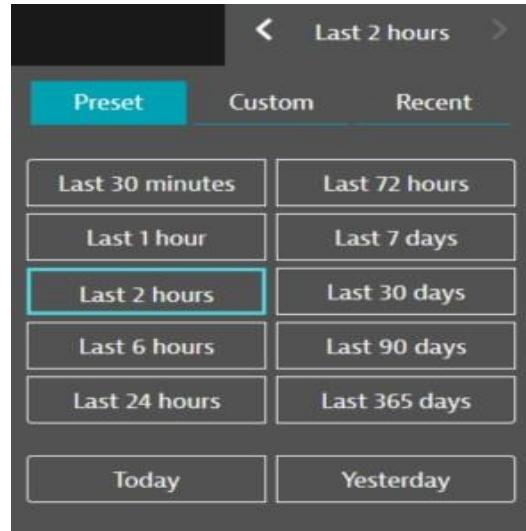
- The search bar works across your entire application environment
- You can search for virtual and physical hosts, applications, services, processes, and other entities
- You also get a list of all related discussion threads in our Answers forum and help topics in Dynatrace Help.



UI Basics

Time Frame Selector

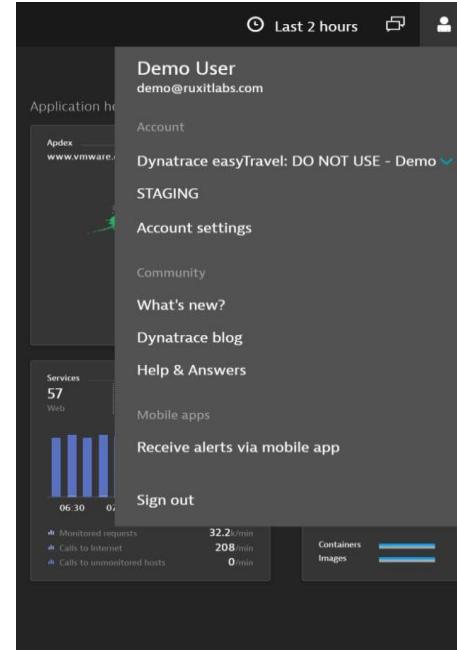
- Quick preset time settings are available for selection
- Set a custom time frame
- Recently used Custom Time Frames are available on the Recent tab



UI Basics

User Menu

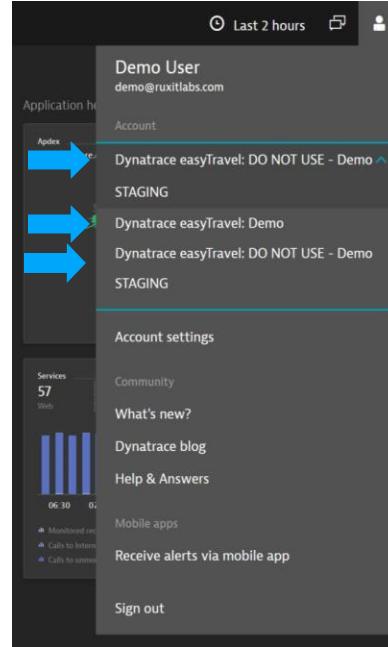
- **Use the “Person Icon” to access the User Menu**
- **Here you can:**
 - View details about the currently logged in user and account information
 - Toggle between different environments
 - Link to the Account settings (SaaS)
 - Access the Dynatrace Product News, Blog, and Help & Answers
 - Access to APIs
 - Sign Out



UI Basics

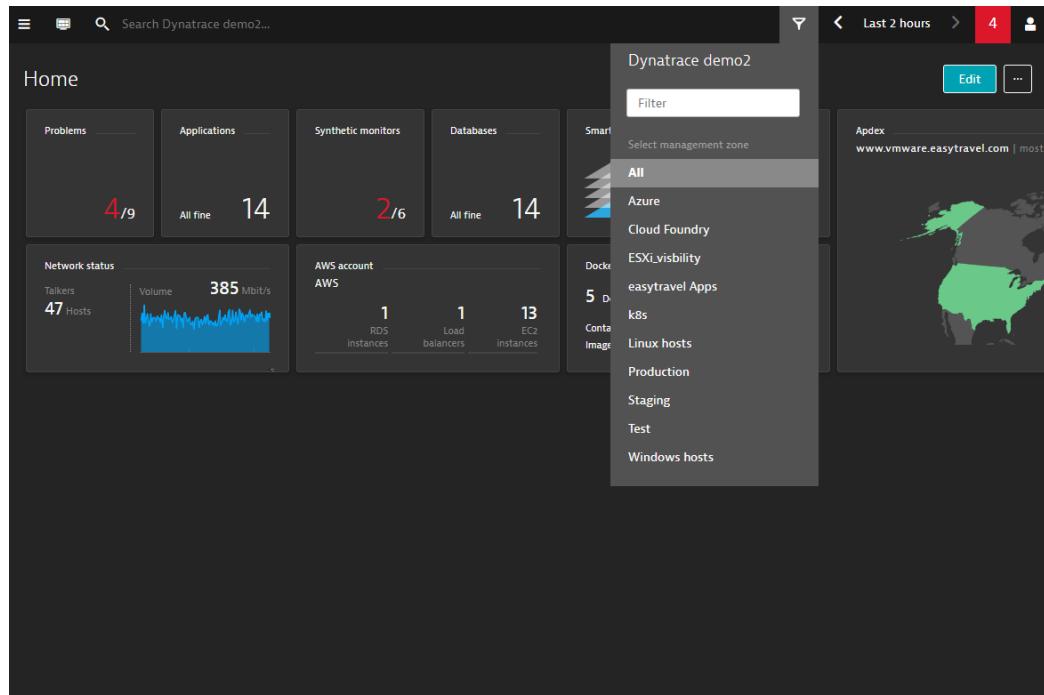
Monitor Multiple Environments

- If you have multiple environments, select the down arrow to show them all and select another, as needed



Management Zones

- Set of entities in a environment
- Granular permission management



Hands On: Vertrautmachen mit der Oberfläche

Ziele:

- Login auf <https://dynatracemanaged.generali-gruppe.de/> mit dem V-Key
 - Eine Suche durchführen
 - Das Zeitfenster ändern
 - Eine Management Zone auswählen und wieder abwählen
 - Öffnen einiger der Out-of-the-box-Dashboards im Side Menu

Topology Model and Smartscape



Application Performance
Management

Speaking the same language

Datacenters

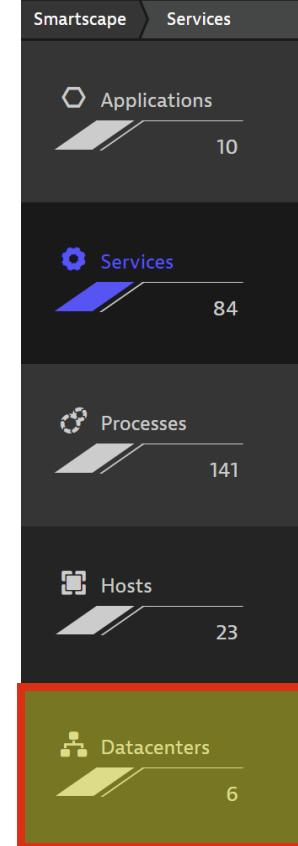
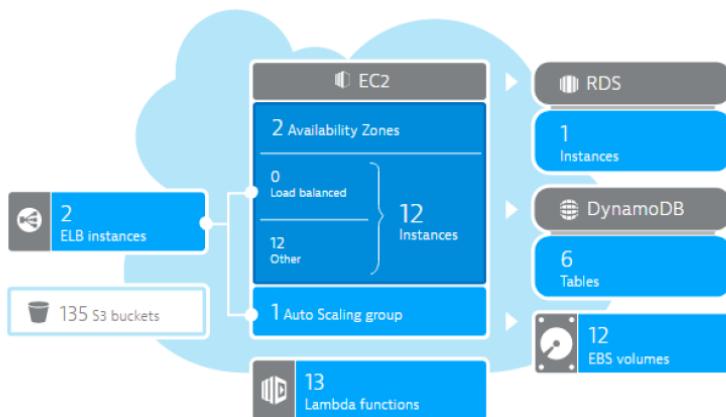
datacenter 

or **data center**

[dē-tuh-sen-ter]

1. a facility equipped with or connected to one or more computers, used for processing or transmitting data

Related terms: vCenter, zone, Virtual, AWS, Geo Region



Speaking the same language

Hosts

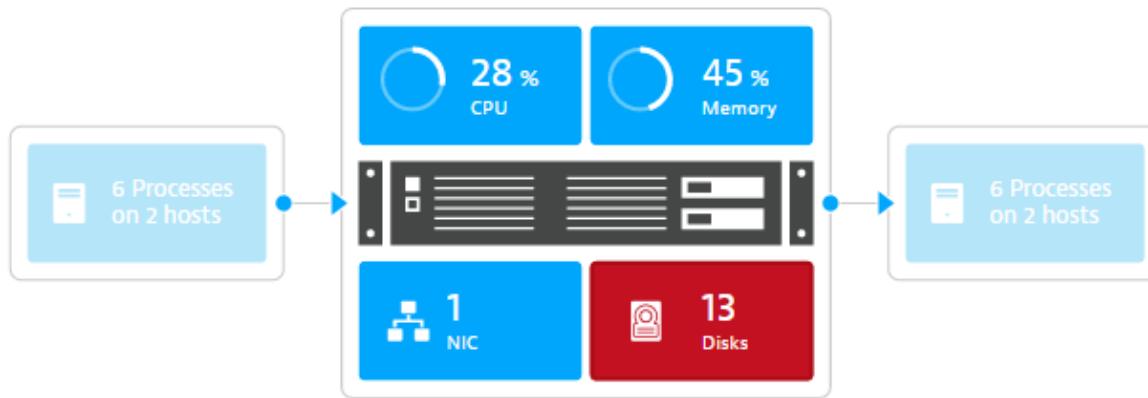
host



1. A physical or virtualized operating system
2. The source of compute, memory, and storage resources

[hohst]

Related terms: CPU, Interface, Virtual, AWS



Speaking the same language

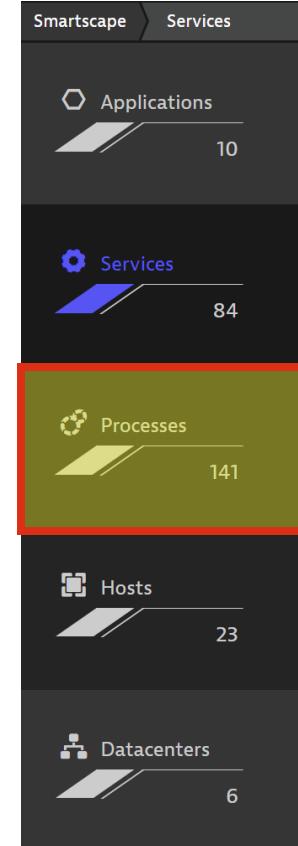
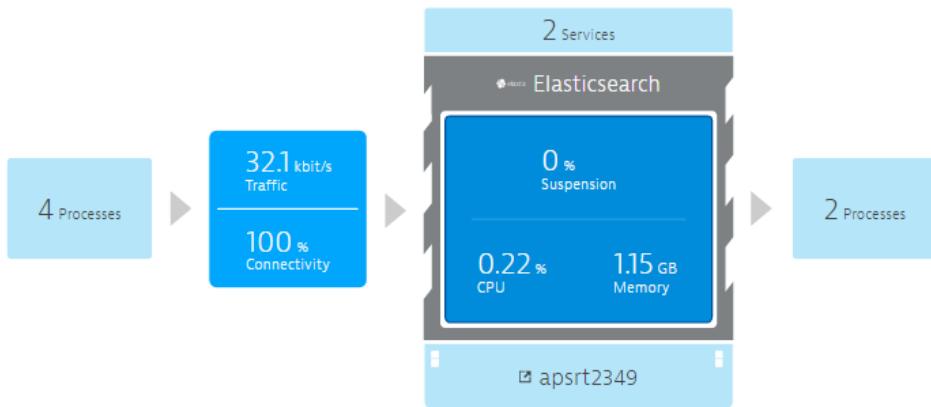
Processes

process 

[pros-es; especially British proh-ses]

1. A currently executing computer program
2. A means for code to request computing resources

Related terms: Requests, Utilization, PID, Memory



Speaking the same language

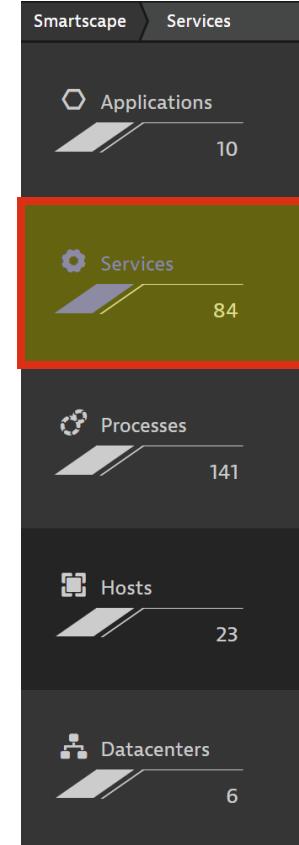
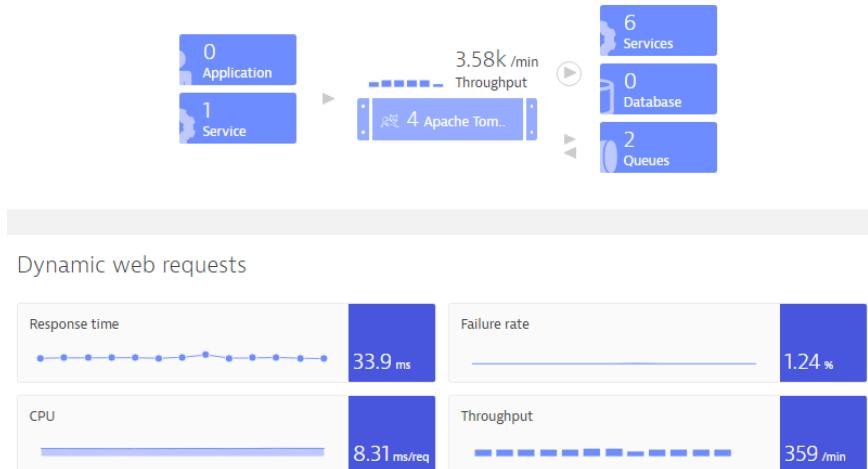
Services

service 

[sur-vis]

1. A set of code that accepts requests and returns results
2. The result of instrumenting a process
3. The “code layer” which requires “deep dive”

- Related terms:
Instrument, Web
Request, Breakdown,
Service Flow, Service
Backtrace, Hotspot,
PurePath



Speaking the same language

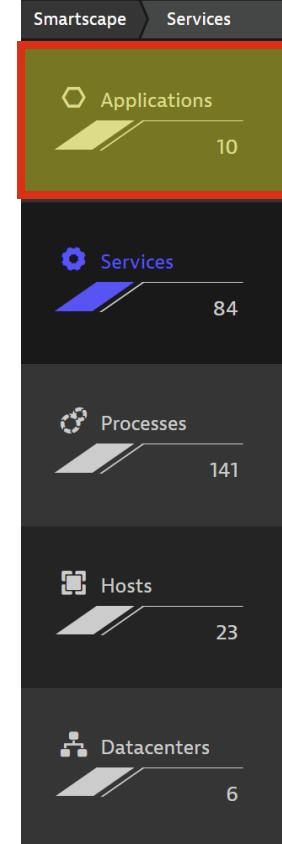
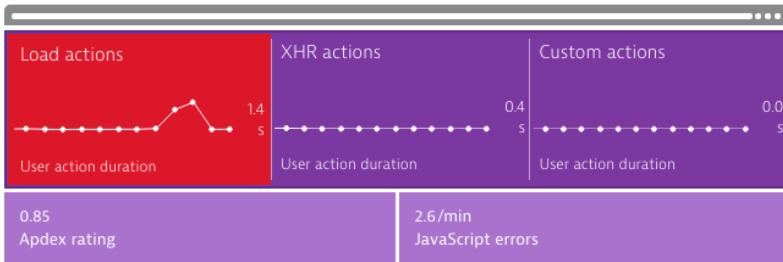
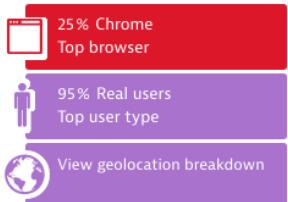
Applications

application 

[ap-li-keɪ-shuh n]

1. User experience as measured at the endpoint, such as a browser or mobile device
2. How software is presented to the end user

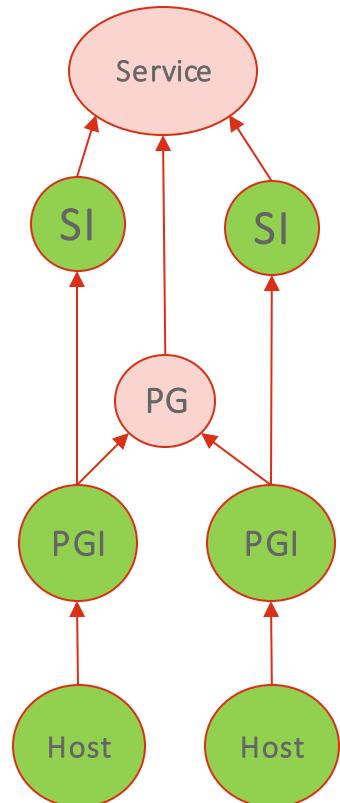
Related terms: Browser, User Action, Session, JavaScript, Waterfall



Dynatrace Topology Model

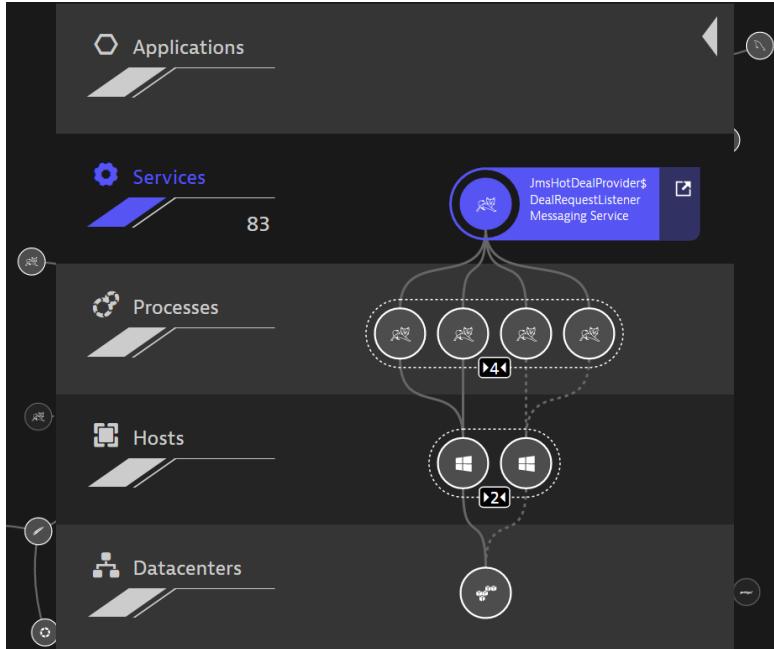
- Dynatrace is based on fully automated discovery of monitored entities and relations
Represented in SmartScape – real time
- SmartScape represents the actual view on how the systems horizontal and vertical relations
- Many different types of entities and relations, but the most important ones are
 - Host
 - PGI (Process Group Instance), in principle a continuous representation of process on a host
 - PG (Process Group), a logical group PGIs that belong to the same family, e.g. 5 Tomcats forming a PG
 - Service Instance (SI), one service discovered and running on a PGI
 - Service, logic group of Service Instances that serve one Service as a cluster on distributed PGIs
 - Application (not shown in diagram)

 = tangible  = logical



SmartScape

Horizontal view shows tier dependencies



- Automatically discovers your entire application stack
- Maps dependencies of found entities
- Presents the findings in the form of an interactive map

SmartScape Demo



Application Performance
Management

Hands On

Ziele:

- Finden des **Nginx Services (VPROD)**, der für **Cosmos Direkt** ausliefert
 - Prüfen der Verbindungen des Services
 - Finden eines Hosts, der diesen Service bereitstellt
 - Welche anderen Prozesse/Services laufen auf diesem Host?
 - Absprung in die Host-Ansicht bei diesem Host

Kaffee Pause



Application Performance
Management

RUM

Real User Monitoring



Application Performance
Management

RUM Overview

- Real User Monitoring (RUM) collects metrics from the users' web browsers and correlates the browser data with server side information gathered by the OneAgent
- Web browser data is collected by a JavaScript tag that is placed inside the applications' HTML pages

Architecture

- (modern) Browser rendering HTML (at least one – Single Page Apps)
- Dynatrace JavaScript code
- Data channel to send back from the browser to Dynatrace
 - called beacon or monitor signal

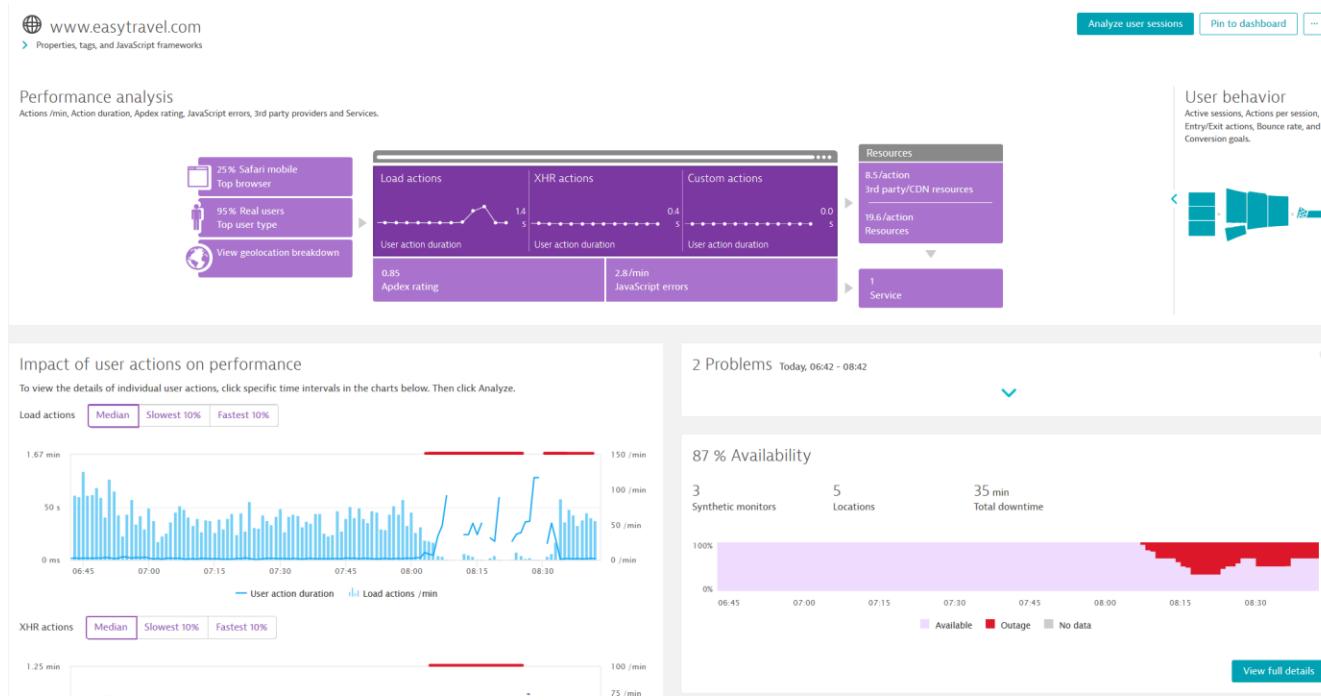
Web Applications Performance analysis



Application Performance
Management

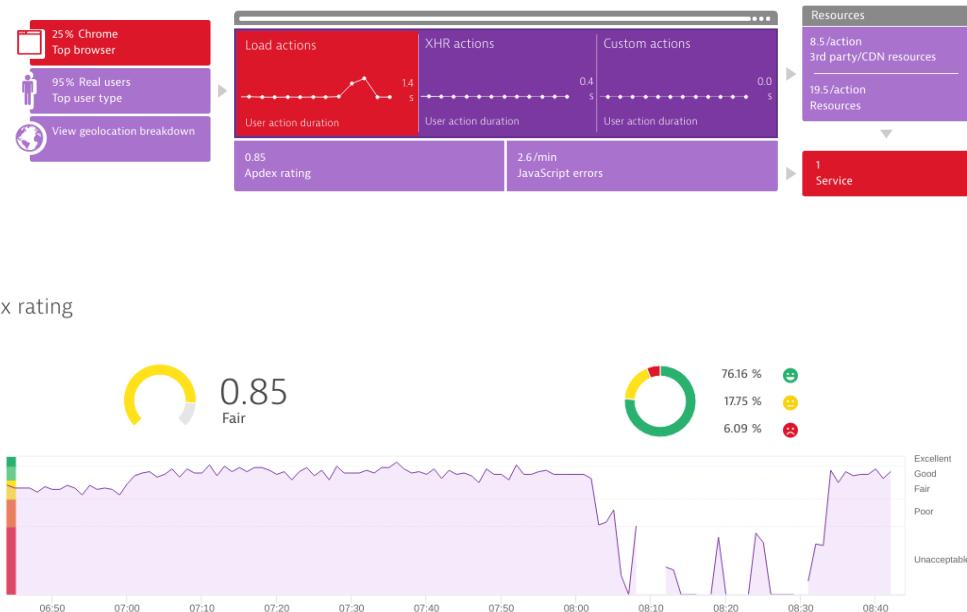
Performance analysis

Details in demo



What is Apdex Rating

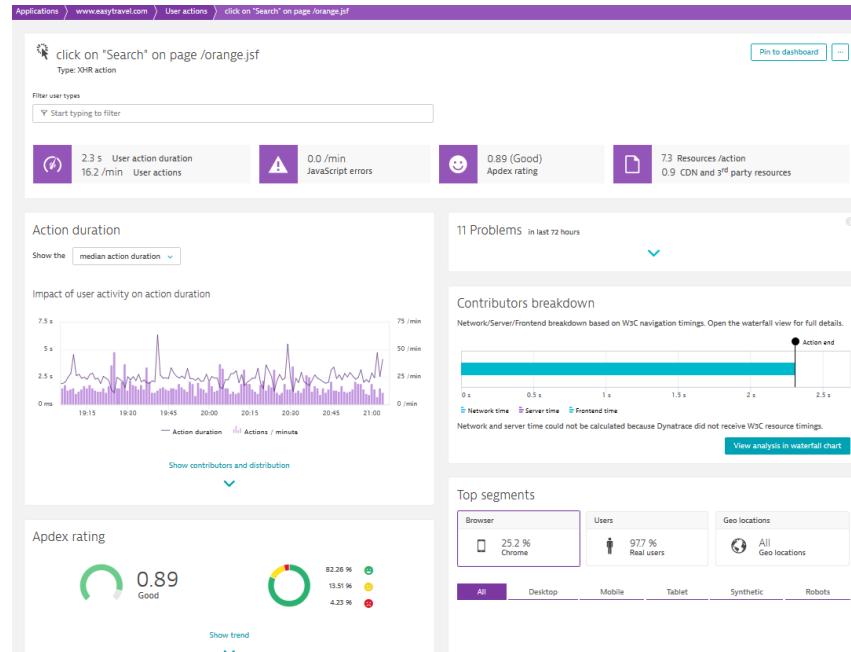
- Universal standard used to measures user satisfaction with application performance
- Based on Performance Metric and Errors
 - Default threshold is 3 seconds
- Custom thresholds possible
- Hot it works
 - Value = 1 Perfect!
 - Value < 0.5 Poor 😞
- Benchmarking and comparison
- User Session Index calculated for every user from the Apdex



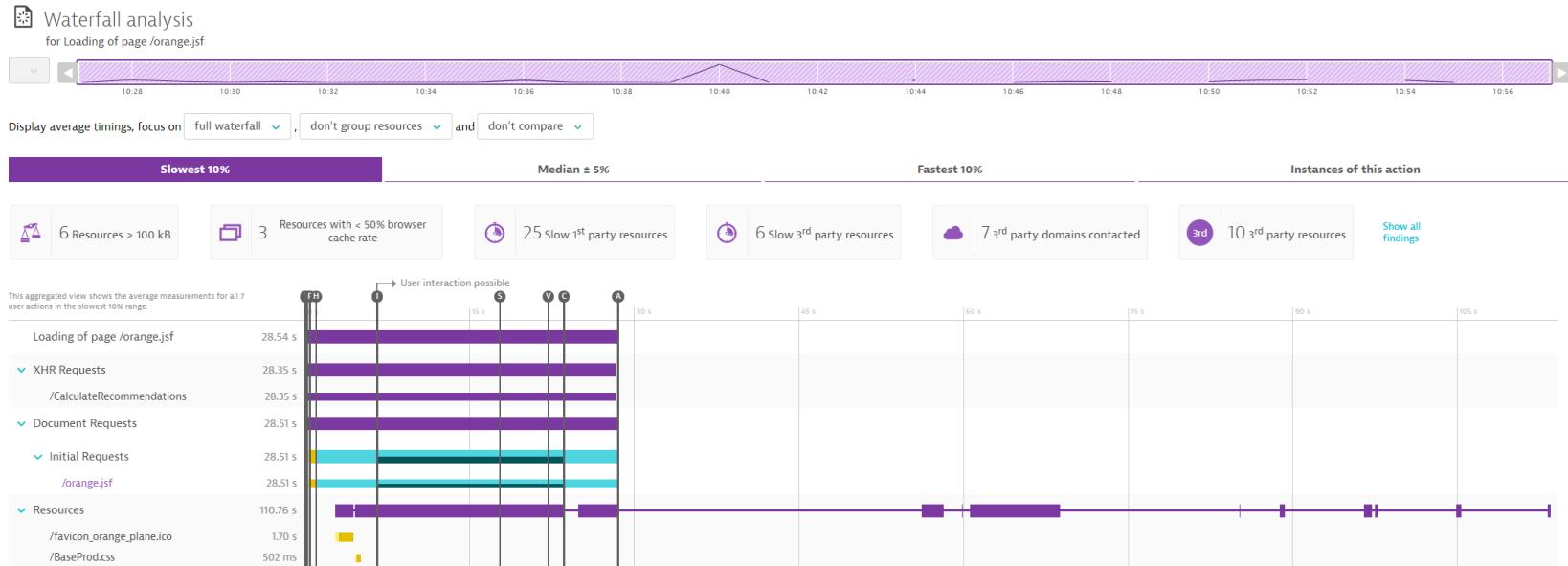
User Action

Performance Analysis Overview

- Action duration vs load
- Contributor breakdown & waterfall analysis
- Resources
- Top web request contributors
- Java Script impacry & analysis



Drill down to RUM Details for all users



View JavaScript errors.. and dynamically filter to fit your need

The screenshot shows a web-based application interface for managing JavaScript errors. At the top, a navigation bar includes links for 'Applications', 'www.easytravel.com', 'Top JavaScript errors', and 'Unexpected end of input'. A purple header bar above the main content area displays the URL 'www.easytravel.com' and a message: 'Have a detailed look under which conditions the JavaScript error has occurred by clicking on the occurrence statistics findings.' On the right side of this bar is a button labeled 'Ignore this JavaScript error'.

The main content area is titled 'Error details for: Unexpected end of input - jquery-110.2.min.js'. It features a summary box containing a warning icon, the error message 'Message: Unexpected end of input', the count 'occurred 127 times', and the file path '//javascript/pkg/vendors/jquery-110.2.min.js'. To the right of this box is a 'Occurrence statistics' section with five colored cards: a purple card for 'amazon.com' (100%), a grey card for 'Multiple browsers' (20%), a blue card for 'Multiple user actions', a yellow card for 'United States', and a green card for 'Multiple operating systems'.

Below this is a large white panel titled 'Unexpected end of input' with the subtitle 'For all browsers'. It contains a table with columns for 'Script file', 'Browser', 'User actions', 'Domains', 'Count', 'Message', and 'Stacktrace'. The 'Script file' column lists 'jquery-110.2.min.js'. The 'Browser' column shows 'Chrome' selected. The 'User actions' column lists 'Loading of page /orange-booking-review.jsf', 'Loading of page /orange.jsf', and 'Loading of page /orange-booking-payment.jsf'. The 'Domains' column lists 'ec2-54-174-224-100.compute-1.amazonaws.com'. The 'Count' column shows '20'. The 'Message' column lists 'SyntaxError: Unexpected end of input'. The 'Stacktrace' column displays a detailed call stack:

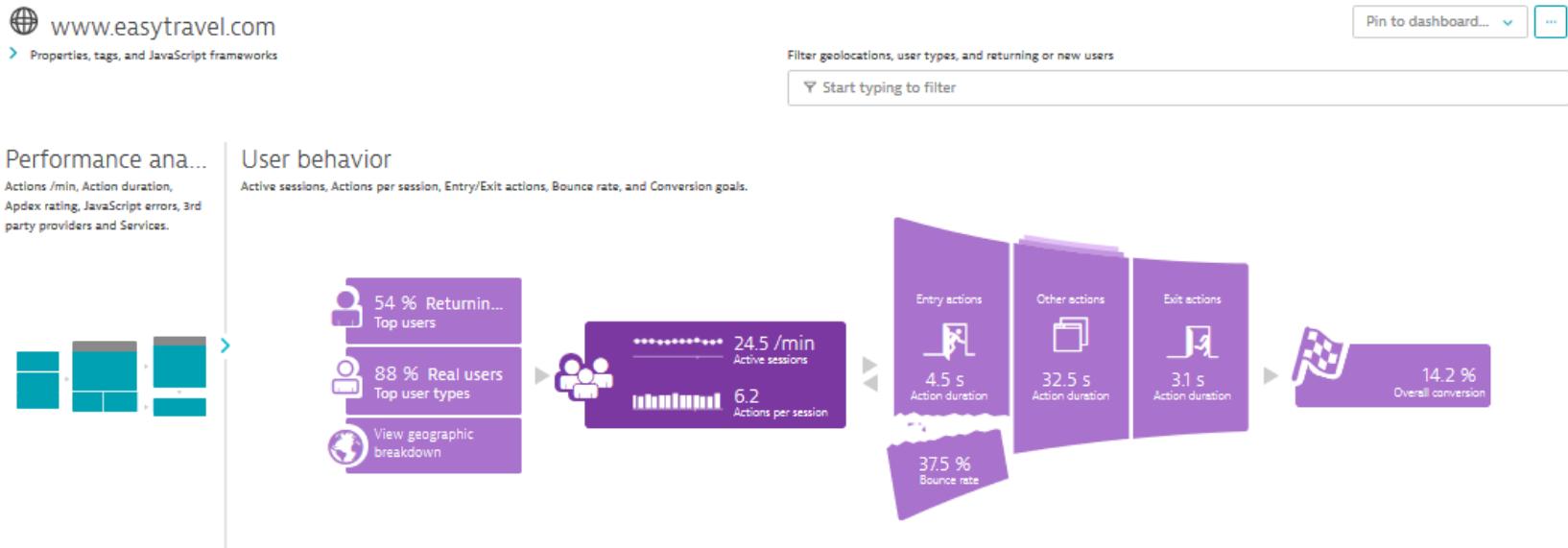
```
at Function ct.extend.parseJSON in http://ec2-54-174-224-100.compute-1.amazonaws.com/javascript/pkg/vendors/jquery-1.10.2.min.js:15166
at require callback in http://ec2-54-174-224-100.compute-1.amazonaws.com/Store/cart/cart.jsp?rdr=action=aditemtocart&url_catalog_ref_id=BOM0020-TFRGR-XL&url_product_id=BOM0020&url_quantity=1:22
at Object.b.aexecCb in http://ec2-54-174-224-100.compute-1.amazonaws.com/javascript/pkg/common/bootstrap.js:28792
at Object.b.check in http://ec2-54-174-224-100.compute-1.amazonaws.com/javascript/pkg/common/bootstrap.js:22388
at Object.b.anonymous in http://ec2-54-174-224-100.compute-1.amazonaws.com/javascript/pkg/common/bootstrap.js:25253
at <unknown> in http://ec2-54-174-224-100.compute-1.amazonaws.com/javascript/pkg/common/bootstrap.js:17248
at <unknown> in http://ec2-54-174-224-100.compute-1.amazonaws.com/javascript/pkg/common/bootstrap.js:25637
at each in http://ec2-54-174-224-100.compute-1.amazonaws.com/javascript/pkg/common/bootstrap.js:1674
at Object.x.emit in http://ec2-54-174-224-100.compute-1.amazonaws.com/javascript/pkg/common/bootstrap.js:25605
```

Web Applications User Behavior



Application Performance
Management

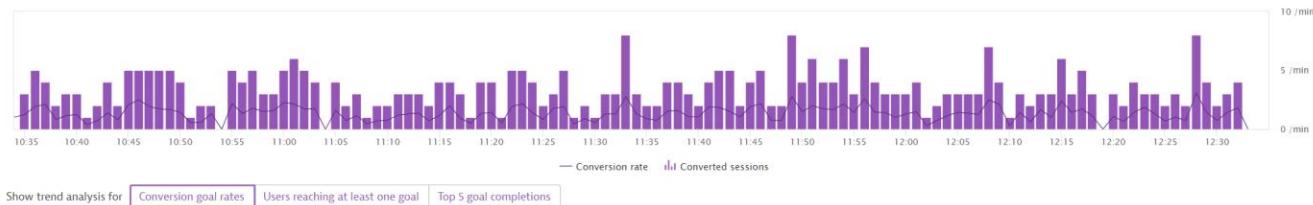
User Behavior



Conversion goals

Conversion trends of www.easytravel.com

Compare trends of individual goals to see how well your application meets your business objectives.



Conversion goals

Name

Successful Bookings - click on Finish

Type of goal

User action

Conversion rate ▾

13.36 %

Completions

403

Users

216

Details

Successful Bookings - open final Page

Destination

0.27 %

8

8

Details

User Sessions



Application Performance
Management

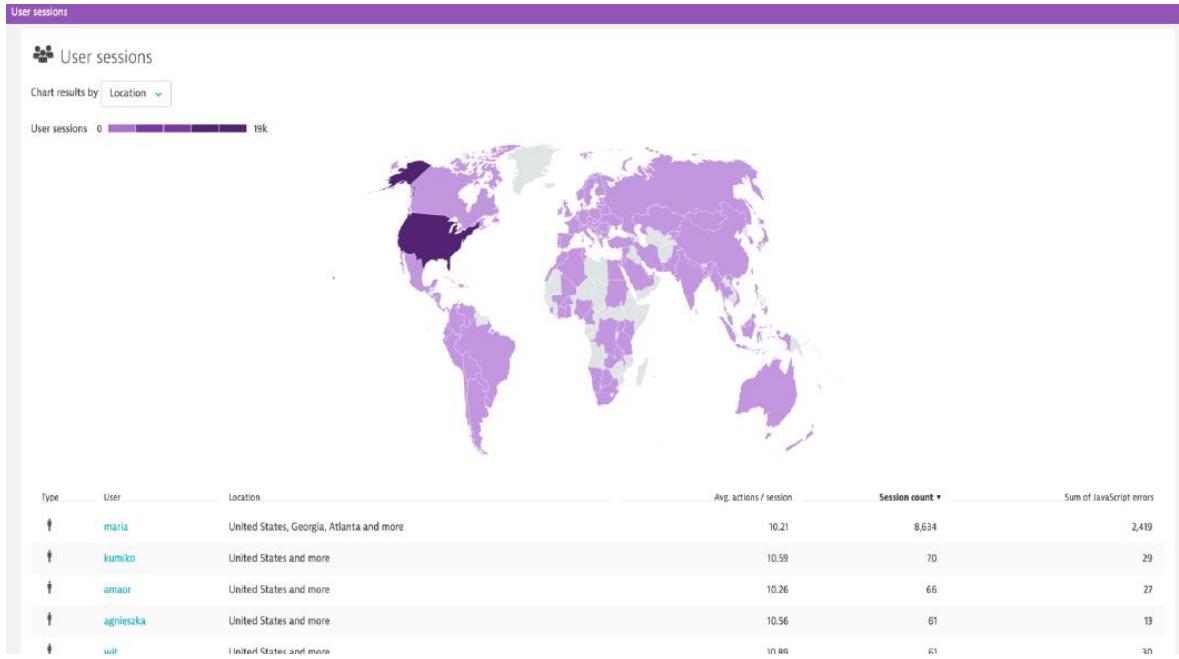
User Sessions

- What is a user session?
- New users
- Identify specific users
- Visit origin
- Device type

User sessions

Chart results by **Browser**

- Application
- Browser**
- Bounce
- Converted
- Conversion goals
- Browser type
- JavaScript error count
- Location
- ISP



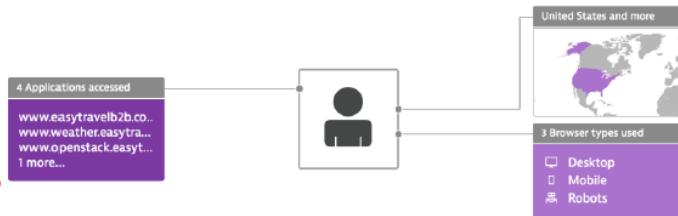
User Sessions

Cross application behavior analysis



wit

Session details related to this user profile are outlined below.



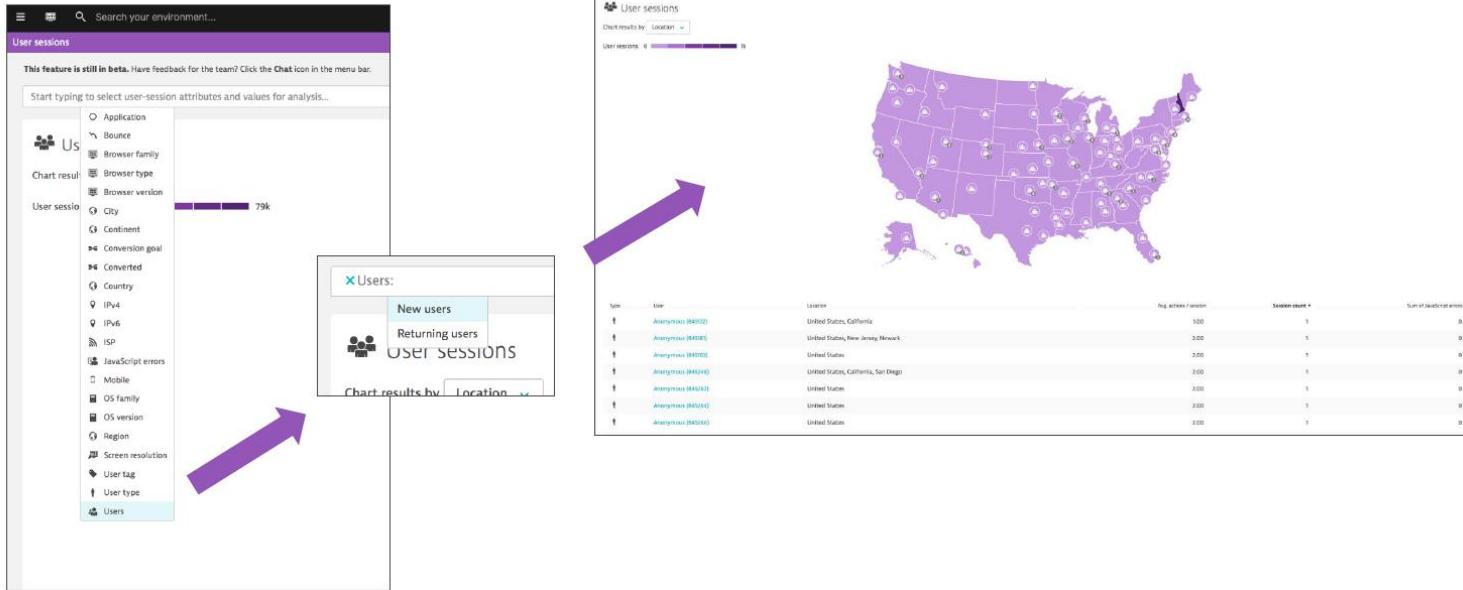
4 Applications

Accessed by "wit" during the selected time frame.

Application	Last access	Avg. actions / session	Session count	JavaScript errors
www.openstack.easytravel.com	37 min	9.20	46	8
www.easytravel.com	42 min	10.18	38	26
www.weather.easytravel.com	42 min	11.91	32	0
www.easytravelb2b.com	2 h 22 min	6.00	1	0

User Sessions

Advanced Filtering



User Sessions

Analysis

User sessions / maria

This feature is still in beta. Have feedback for the team? Click the Chat icon in the menu bar.

maria

Session details related to this user profile are outlined below.

2 Applications accessed

- easyTravel Customer ...
- Node.js Weather

United States and more

2 Browser types used

- Desktop
- Mobile

2 Browser types

Used by "maria" during the selected time frame.

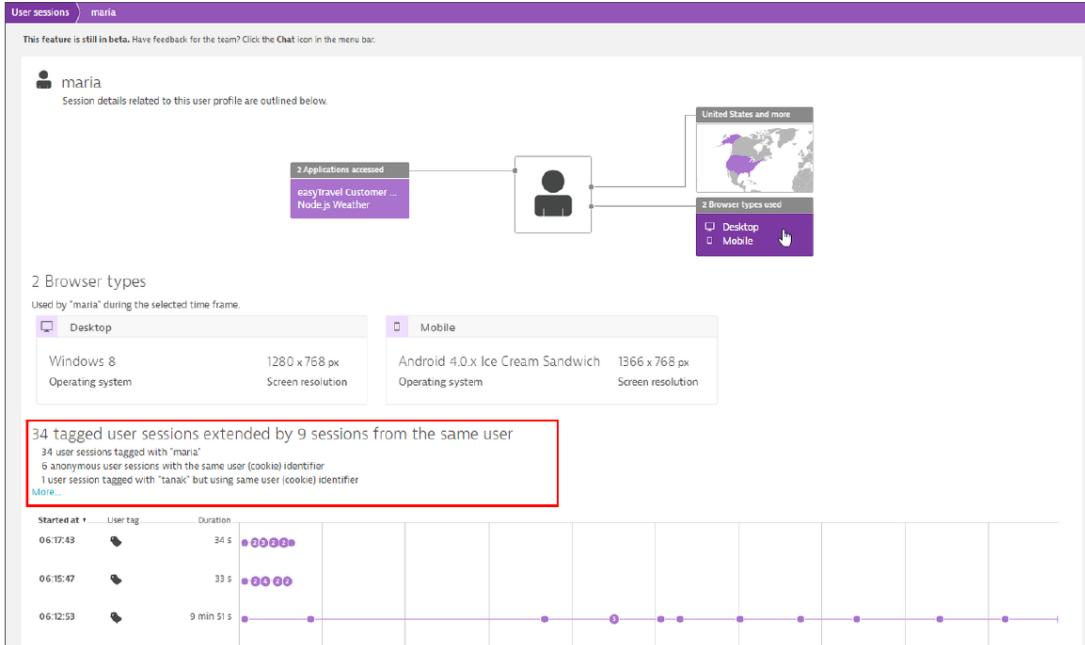
Desktop	Mobile
Windows 8	Android 4.0.x Ice Cream Sandwich
Operating system	Screen resolution

34 tagged user sessions extended by 9 sessions from the same user

34 user sessions tagged with "maria"
6 anonymous user sessions with the same user (cookie) identifier
1 user session tagged with "tanak" but using same user (cookie) identifier
[More...](#)

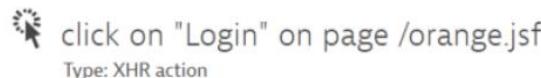
Started at User tag Duration

Started at	User tag	Duration
06:17:43	...	34 s
06:15:47	...	33 s
06:12:53	...	9 min 51 s



User Actions

- What are user actions?
- User action types
 - Initial Page Load
 - XHR calls
- Key User Actions



Applications > www.easytravel.com > User actions > Loading of page /privacy-orange-mobile.jsf

Loading of page /privacy-orange-mobile.jsf
Type: Load action, domain: ec2-54-84-39-252.compute-1.amazonaws.com

Data for non key user actions is available back to 13 February 2017 00:55 AM. For longer historical trends, setting a user action specific Apdex threshold or adding a user action tile to your dashboard, mark this user action as **key user action** by using the context button.

Filter user types
Start typing to filter... ▾

0.4 s User action duration 2.2 /min User actions	0.1 /min JavaScript errors	0.99 (Excellent) Apdex rating	14.0 Resources /action 3.0 CDN and 3 rd party resources
---	----------------------------	-------------------------------	---

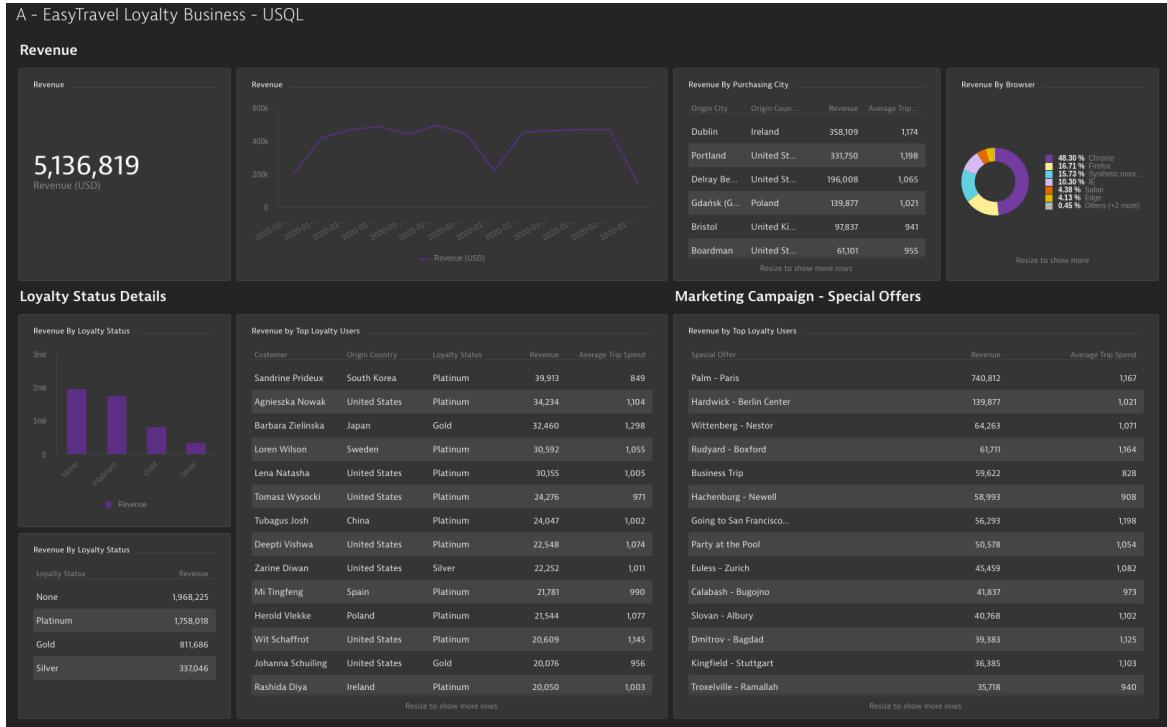
Action duration
Show the median action duration ▾

9 Problems in last 72 hours ▾

watched

User Session Querry Language (USQL)

Slice and Dice RUM Data



RUM Demo



Hands On

Ziele:

- Aufrufen der Applikation **Cosmos Direkt**
 - Welche sind die 3 meistbenutzten Browser?
 - Wie verlief der Apdex in den letzten 30 Tagen?
 - Benutzt die Applikation CDNs?
 - Was war der häufigste JavaScript Fehler in den letzten 30 Tagen?
 - Auf welcher Seite trat er am häufigsten auf?
 - Welche User Action hat gestern die meiste Ladezeit verursacht?
 - Wie verteilt sich die Zeit auf Browser, Netzwerk und Server?
 - Betrachten des Wasserfalls für diese User Action

Backend Dashboards



Application Performance
Management

In depth look at the Dynatrace model

Hosts

Process
Groups

Services

Hosts



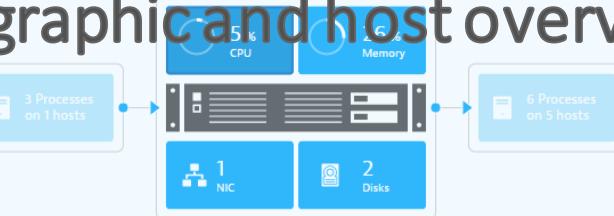
 CF1-tomcatjms
Uptime: over 31 days

Analyze process connections ...

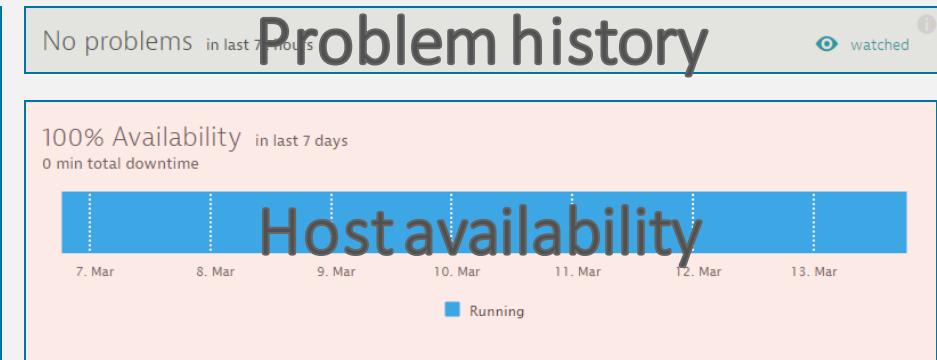
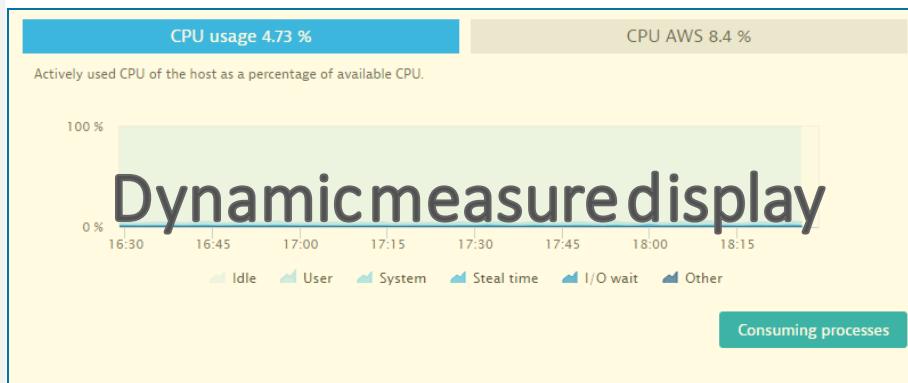
Properties and tags

Ubuntu 14.04.5 LTS, Trusty Tahr (kernel 3.13.0-108-generic) [23 more...](#)

Infographic and host overview



3 Processes on 1 hosts 5 CPU 6 Memory
1 NIC 2 Disks 6 Processes on 5 hosts



Host Demo

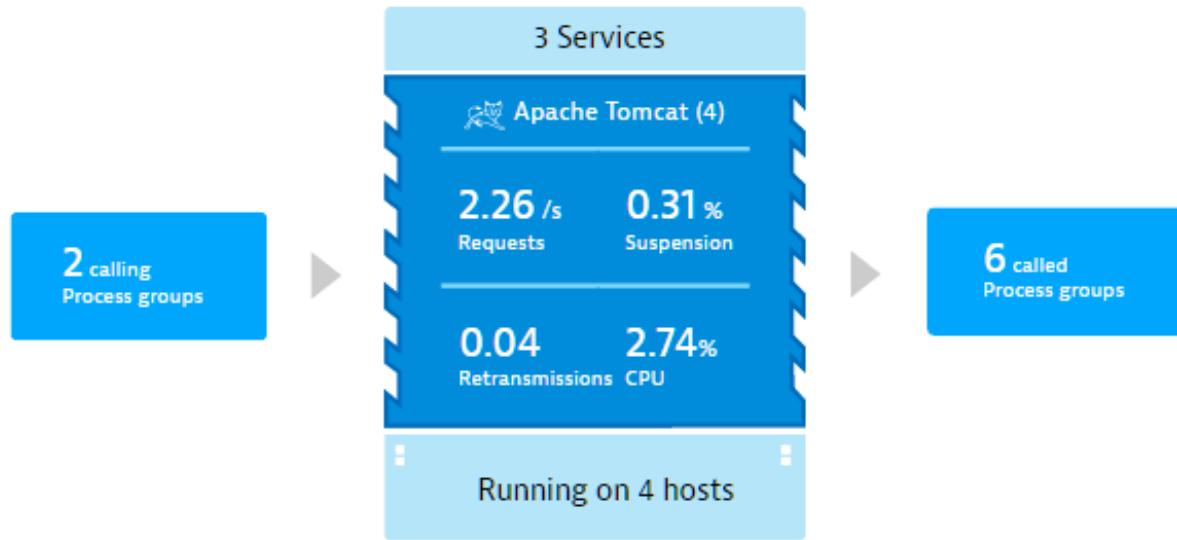


Process groups



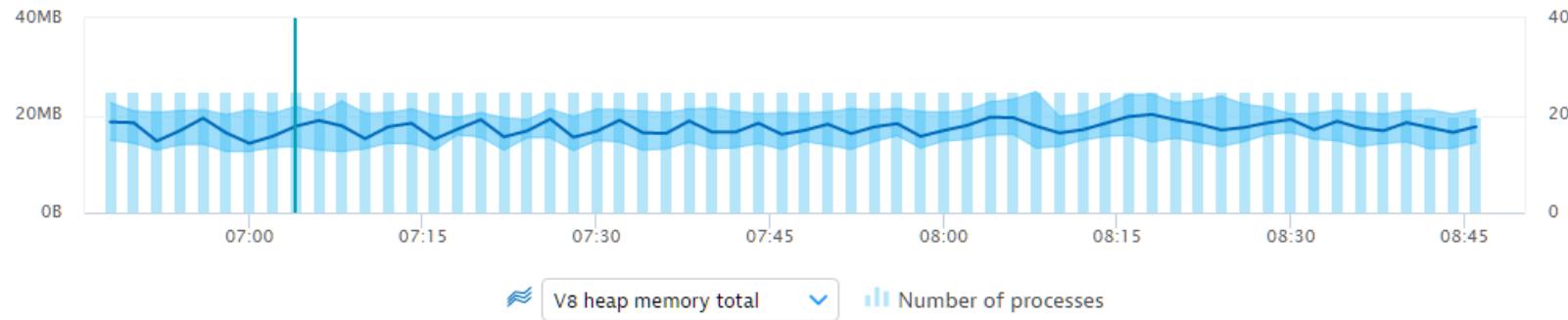
What are process groups

- Cluster of processes belonging together
- Tomcat cluster, Jboss cluster, WebSphere cluster
- Run the same software
- Service boundaries



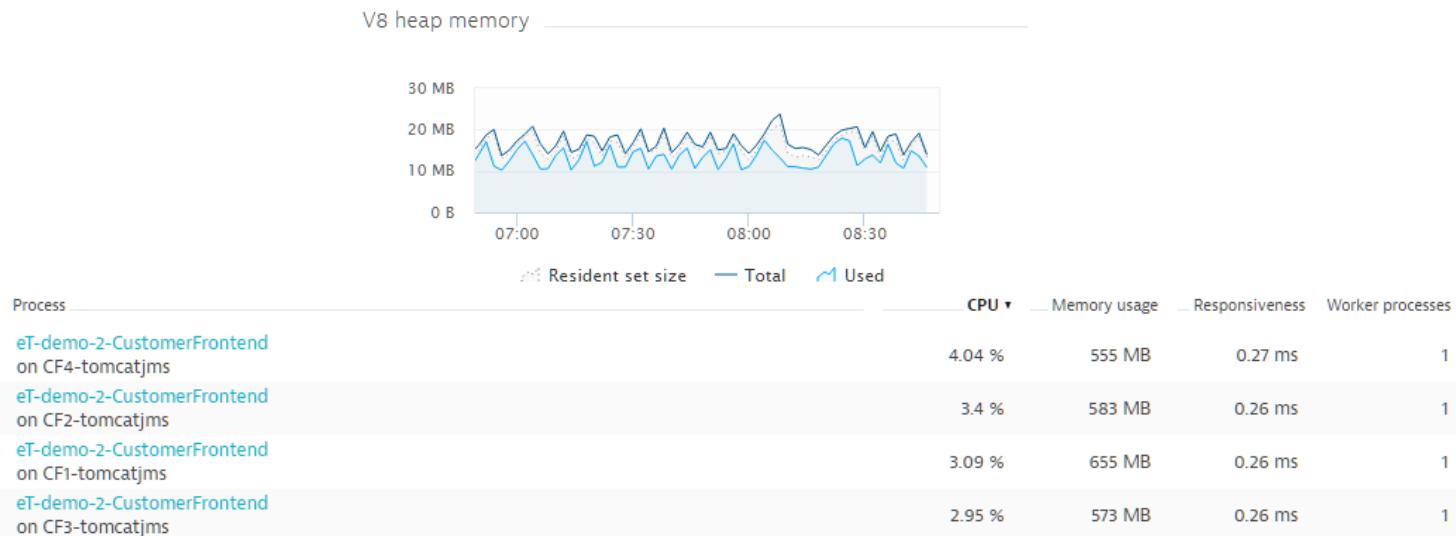
Why process groups

- Should be stable! (Deployment, version upgrade)
- Points of configuration
- Process, Plugin metrics continuity



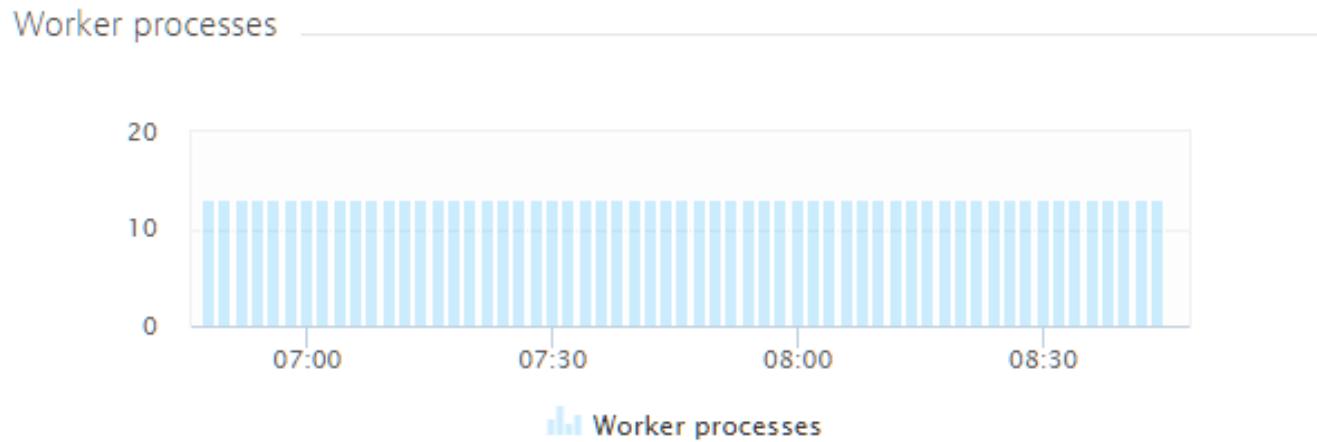
Processes vs Process group instance

- A process group on a host is a process group instance
- Normally one process with chart continuity (restart, crash, redeploy)



Processes vs Process group instance

- A process group instance can have many processes...
- Apache HTTP server has many worker processes (instead of threads), so does Node.js
- An Oracle DB consists of many processes...



Process group detection

Intention: Stable, meaningful and meta data

- **Lots of built-in rules**
 - Java
 - Jboss
 - WebSphere
 - Node
 - Apache
 - Databases
 - ...
- **Many special rules**
 - Docker
 - Azure
 - CloudFoundry
 - OpenShift
 -

Process group detection

Metadata

Customize and why?

- Not perfect for unknown technologies and frameworks
- Everything looks the same
- Customer specific naming needed

Process Group demo



Services



Application Performance
Management

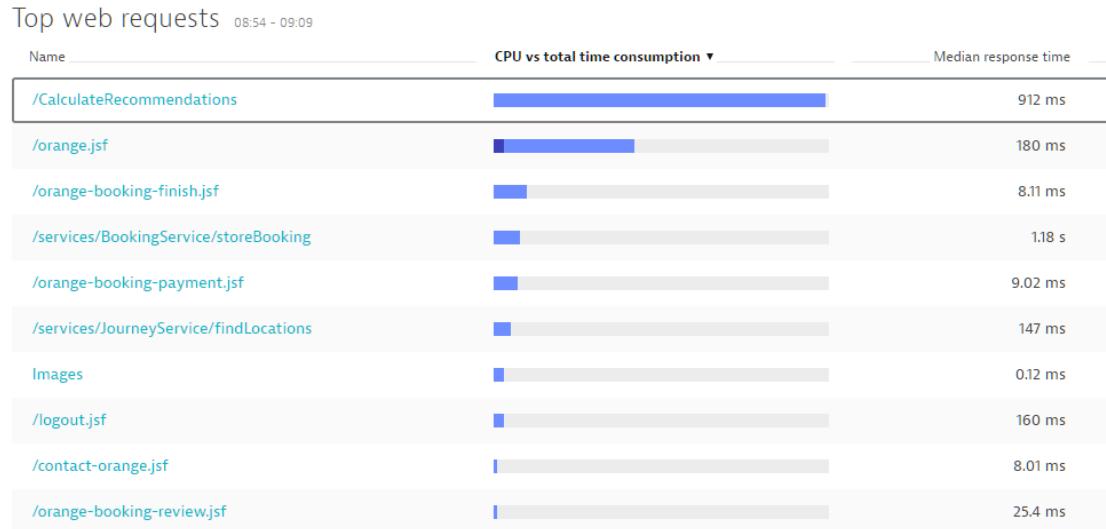
What are services

- Server-side code executed within a process group, e.g a Tomcat server
- Web containers, Web services or custom code that is deployed



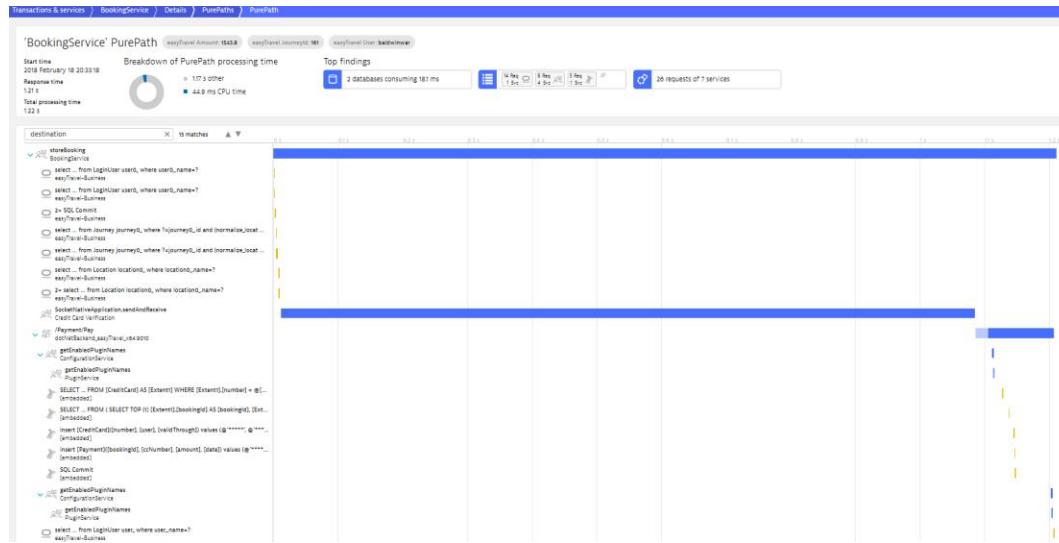
What are services

- All server side requests are monitored via services
- All code level information is collected for each request
- Services are aggregated data of underlying PurePaths



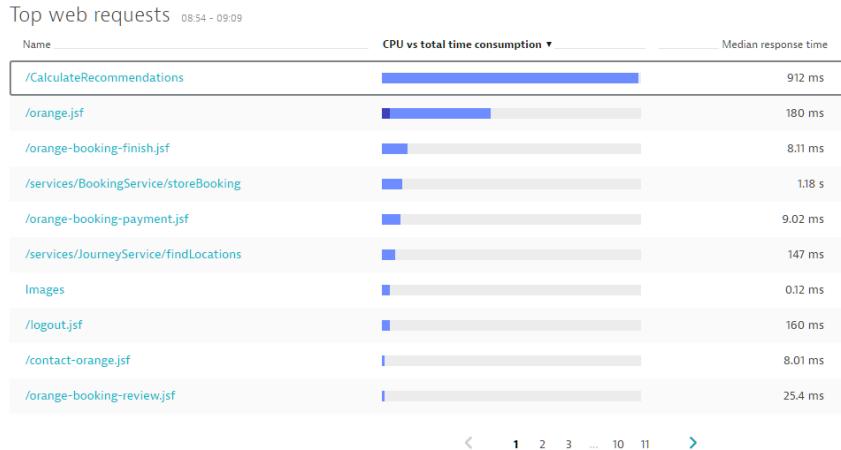
What is a PurePath?

- Timing and code level data context
- More information later



Why Services?

- Dynatrace does this automatically
 - Need to baseline at every tier!
 - Baseline and fault domain isolation work on services and service requests, “not” on PurePath



Database statements		08:55 - 09:10	Sort by response time
select location0_name as name1,_ from Location location0,_ where (lower(location0_name) like '%' ? %) _ and exists (select journey_id from Journey journey0 where journey0_destination_name=location0_name) _ and location0_name like ? _ is not null;		47.9 / min	22.6 ms
(call verify_location(?))		44.9 / min	20.5 ms
select location2_name as col_0,_ count(booking0_id) as col_1,_ location2_name as name1,_ from Booking booking0,_ inner join Journey journey0 on booking0_journey_id=journey0_id and (journey0_tenant_name=?)_ inner join Location location0 on location0_name=location2_name green by location0_name order by name1		1.6 / min	16.9 ms
select location2_name as col_0,_ count(booking0_id) as col_1,_ location2_name as name1,_ from Booking booking0,_ inner join Journey journey0 on booking0_journey_id=journey0_id and (journey0_tenant_name=?)_ inner join Location location0 on location0_name=location2_name green by location0_name order by name1		1.6 / min	16.5 ms
select booking0_id as id1,_ booking0_booking_id as booking0_id, booking0_journey_id as journey0_id,_ booking0_tenant_name as user_name, booking0_fromDate as fromBooking booking0_inner join Journey journey0 on bookings_journey_id=journey0_id and (tenant_name=? order by location0_name) desc fetch select journey0_id as id1,_ journey0_amount as amount1,_ journey0_description as description1,_ journey0_destination_name as destination_name1,_ journey0_fromDate as fromDate1,_ journey0_name as name1,_ fromBooking booking0_inner join Journey journey0 on bookings_journey_id=journey0_id and (tenant_name=? order by location0_name) desc fetch select journey0_id as id1,_ journey0_amount as amount1,_ journey0_description as description1,_ journey0_destination_name as destination_name1,_ journey0_fromDate as fromDate1,_ journey0_name as name1,_ fromBooking booking0_inner join Journey journey0 on bookings_journey_id=journey0_id and (tenant_name=? order by location0_name) desc fetch select count(booking0_id) as col_0,_ from Booking booking0_inner join Journey journey0 on booking0_journey_id=journey0_id and (journey0_tenant_name=?)		4.8 / min	15.2 ms
select sum(journey0_amount) as col_0,_ from Booking booking0_inner join Journey journey0 on booking0_journey_id=journey0_id and (journey0_tenant_name=?)		1.71 / min	14.1 ms
select journey0_id as id1,_ journey0_amount as amount1,_ journey0_description as description1,_ journey0_destination_name as destination_name1,_ journey0_fromDate as fromDate1,_ journey0_name as name1,_ fromBooking booking0_inner join Journey journey0 on bookings_journey_id=journey0_id and (tenant_name=? order by location0_name) desc fetch select journey0_id as id1,_ journey0_amount as amount1,_ journey0_description as description1,_ journey0_destination_name as destination_name1,_ journey0_fromDate as fromDate1,_ journey0_name as name1,_ fromBooking booking0_inner join Journey journey0 on bookings_journey_id=journey0_id and (tenant_name=? order by location0_name) desc fetch select count(booking0_id) as col_0,_ from Booking booking0_inner join Journey journey0 on booking0_journey_id=journey0_id and (journey0_tenant_name=?)		5 / min	11.4 ms
select count(booking0_id) as col_0,_ from Booking booking0_inner join Journey journey0 on booking0_journey_id=journey0_id and (journey0_tenant_name=?)		6.2 / min	7.06 ms
select sum(journey0_amount) as col_0,_ from Booking booking0_inner join Journey journey0 on booking0_journey_id=journey0_id and (journey0_tenant_name=?)		1.67 / min	5.77 ms
select journey0_id as id1,_ journey0_amount as amount1,_ journey0_description as description1,_ journey0_destination_name as destination_name1,_ journey0_fromDate as fromDate1,_ journey0_name as name1,_ fromBooking booking0_inner join Journey journey0 on bookings_journey_id=journey0_id and (tenant_name=? order by location0_name) desc fetch select journey0_id as id1,_ journey0_amount as amount1,_ journey0_description as description1,_ journey0_destination_name as destination_name1,_ journey0_fromDate as fromDate1,_ journey0_name as name1,_ fromBooking booking0_inner join Journey journey0 on bookings_journey_id=journey0_id and (tenant_name=? order by location0_name) desc fetch select count(booking0_id) as col_0,_ from Booking booking0_inner join Journey journey0 on booking0_journey_id=journey0_id and (journey0_tenant_name=?)		269 / min	4.26 ms

Services demo





Service Monitoring



Application Performance
Management

Service Analysis

Various views for analysis

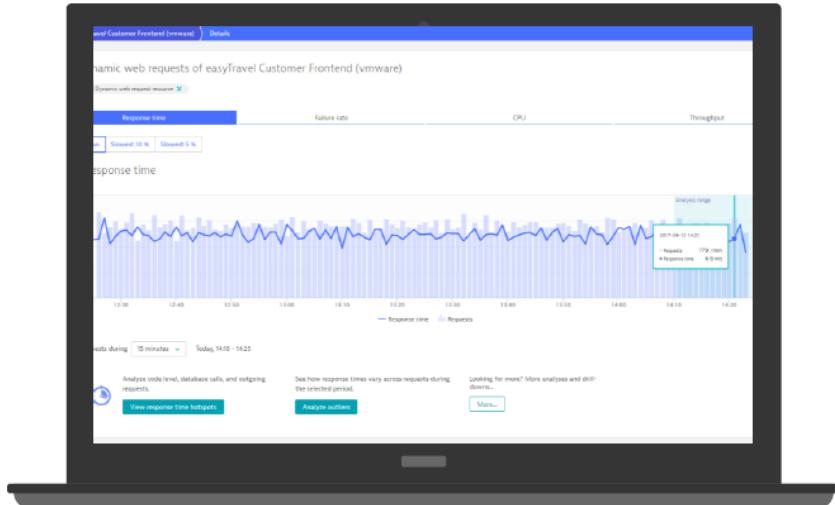
- Service Details
- Service Flow
- Service Backtrace
- Response Time Distribution
- Response Time Hotspots
- Failure Analysis
- PurePath
- Compare

Able to dynamically filter every view

Service Analysis

Performance Details

- **What is it?**
 - Detailed overview of a service's performance
 - Starting point for further analysis
- **When should I use it?**
 - Understand overall performance over time
 - Beginning manual hotspot and failure analysis
 - Landing view for several problem root causes



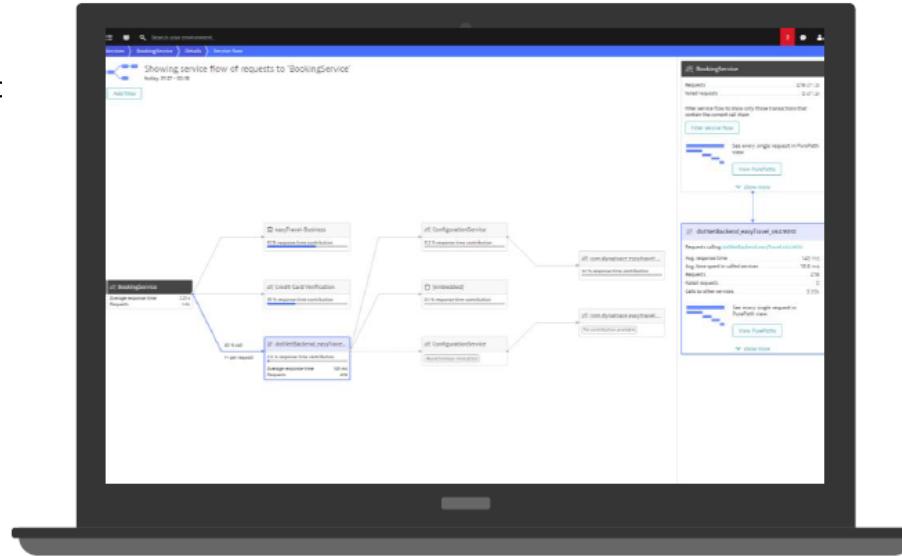
Service Details Demo



Service Analysis

Service Flow

- **What is it?**
 - Overview of all services and queues that a selected service makes requests to and the time spent within those services
- **When should I use it?**
 - Understand the call chain sequence of a service
 - View all the response time contributors for a service



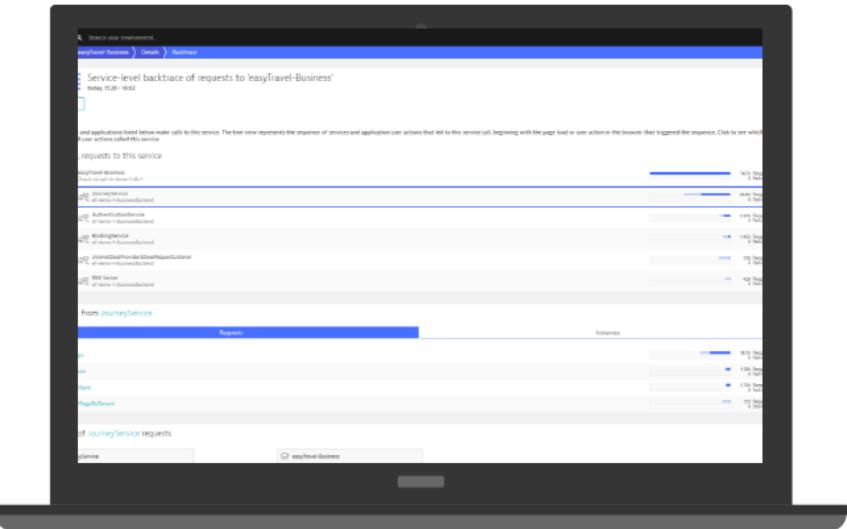
Service Flow Demo



Service Analysis

Service Backtrace

- **What is it?**
 - A view that shows information about who makes calls to a particular service
- **When should I use it?**
 - Understand what services call the selected service
 - Analyze the performance of a service from the perspective of the calling clients



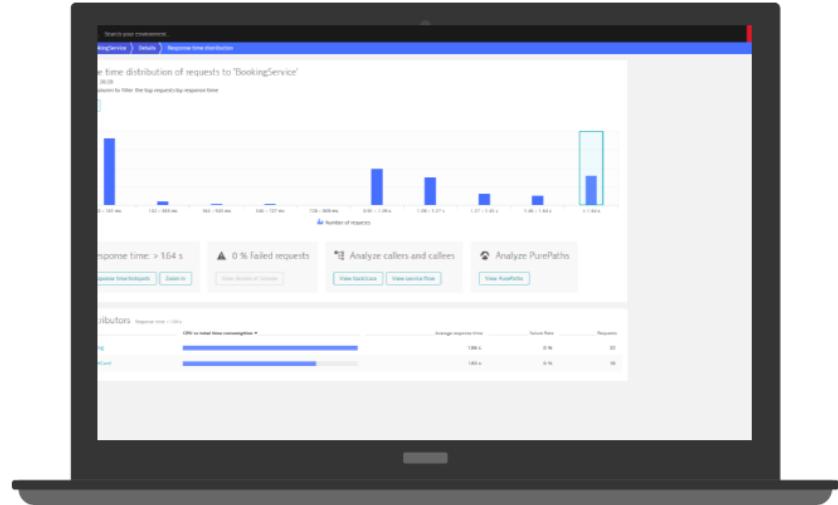
Service Backtrace Demo



Service Analysis

Response Time Distribution and outlier analysis

- **What is it?**
 - A feature that allows you to quickly view the variance in request duration
- **When should I use it?**
 - Easily view performance outliers and pick them for deeper analysis



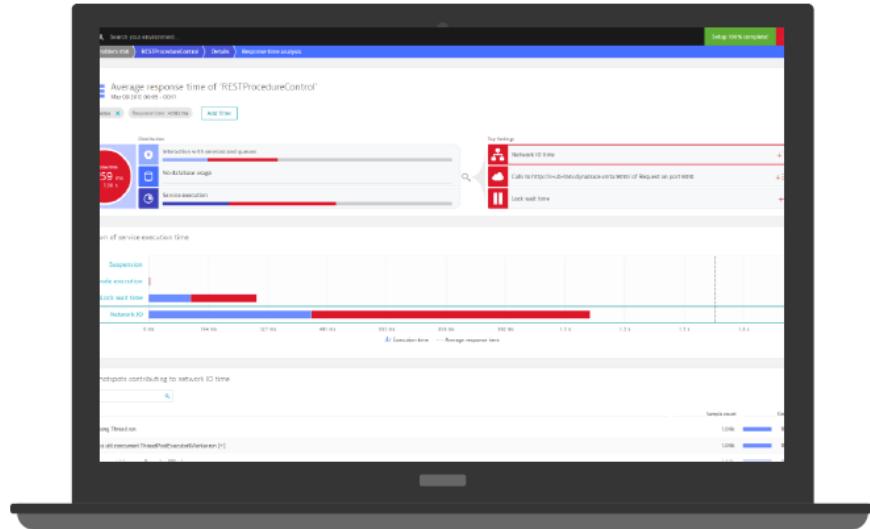
Response Time Distribution Demo



Service Analysis

Response Time Hotspots

- **What is it?**
 - A feature that allows you to breakdown time spent in any service or even individual requests
- **When should I use it?**
 - Performance analysis of any instrumented service
 - Understand total impact of
 - Code
 - DB queries
 - Calls to other services
 - Analyzing performance degradation during problems related to a service or request



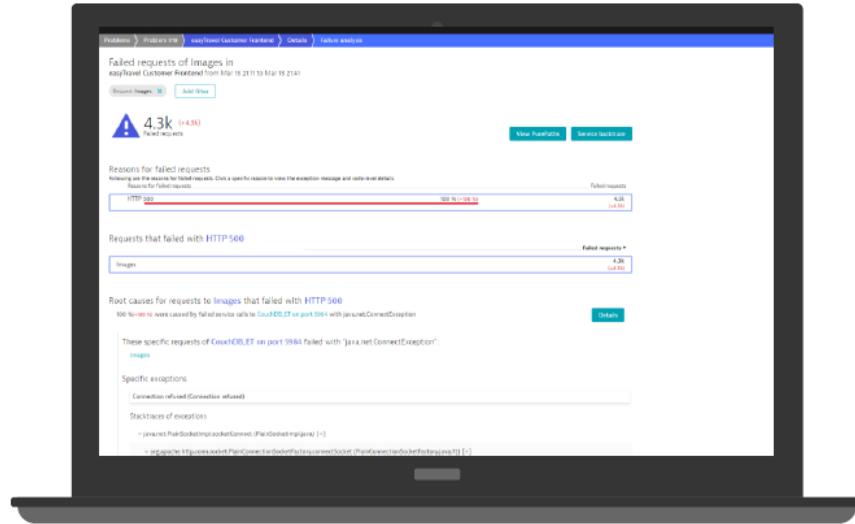
Response Time Hotspots Demo



Service Analysis

Failure Analyzer

- **What is it?**
 - View the details of failures occurring at any instrumented tier
- **When should I use it?**
 - Ad hoc failure analysis of any instrumented service
 - Analyzing failure rate increase during problems



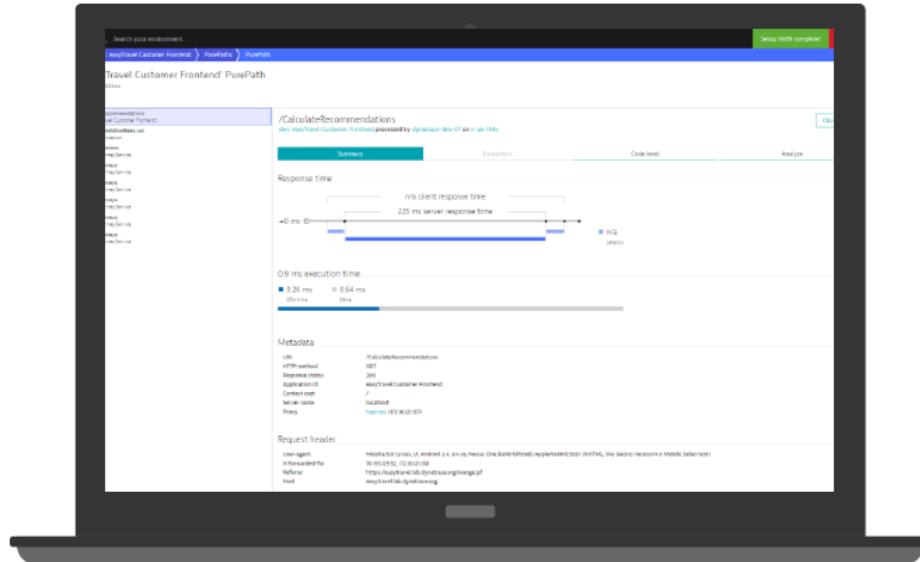
Failure Analyzer Demo



Service Analysis

PurePath

- **What is it?**
 - Deep dive breakdown of a single transaction
 - Waterfall breakdown of where time is spent



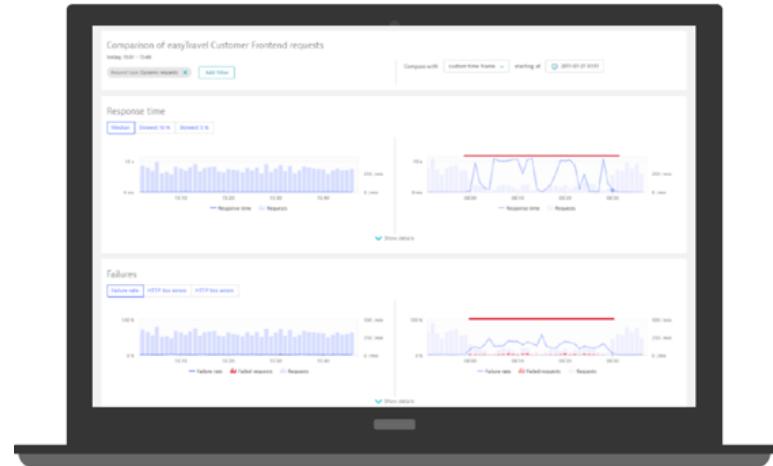
Pure Path Demo



Compare

What is it?

- Compare service-request metrics across two time-frames



Compare Demo

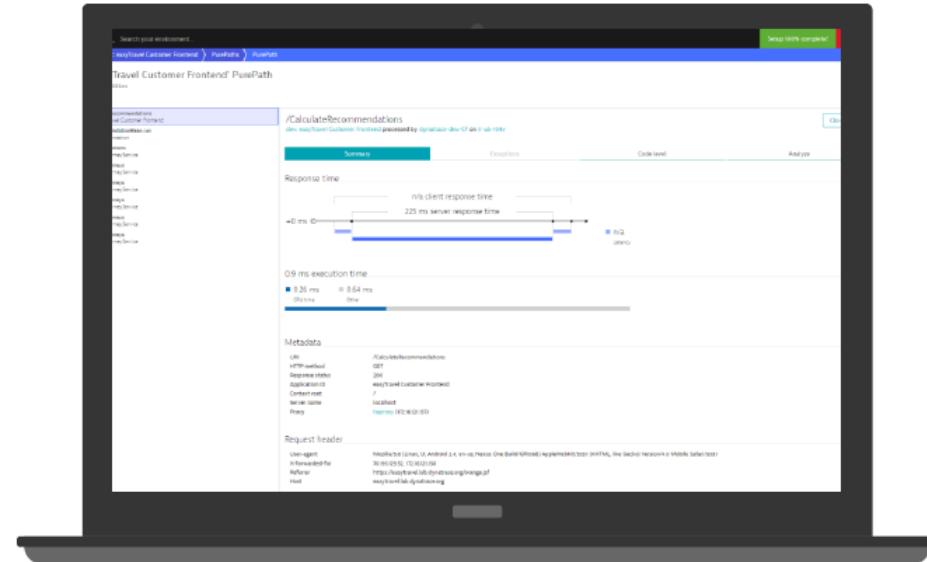


Application Performance
Management

Service Analysis

Multi-dimensional Analysis

- **What is it?**
 - Charting service and custom metrics
 - Splitting by
 - HTTP response codes
 - Requests
 - Request types
 - Service instances
 - Ability to save the diagram to the service



Multi-dimensional Analysis Demo



Hands On

Ziel:

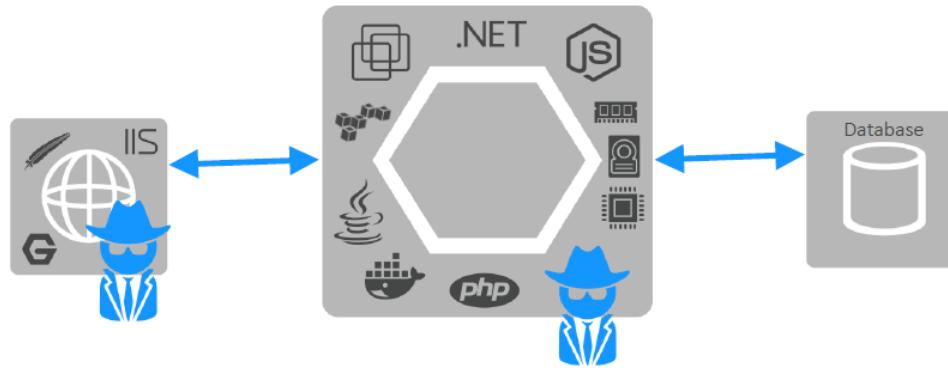
- Aufruf der Applikation **Cosmos Direkt**
 - Welche Services stellen diese Applikation bereit?
 - (Timeframe 30 Tage) Drilldown auf den Service mit den meisten Requests
 - Drilldown in die Performance Details der dynamischen Requests
 - Welche anderen Services hat der Service gerufen?
 - Von wo wurde der Service aufgerufen?
 - Auf welche URL wurde der langsamste Request in diesem Zeitraum abgesetzt und wie lange hat er gedauert?
- ➔ Drilldown zum PurePath

Databases



Database Monitoring

Dynatrace treats database as external services



Note on Database statement baselining

Anomalies detected based on

- Response time of query commands
- Failure rate of query commands

Not every single query is baselined

- Types of queries are baselined – commits, queries and modifications
- If a problem occurs, Dynatrace looks back in time at the performance of all the queries for comparison

Database Demo



Intelligent Correlation and Causation (aka. Dynatrace AI)



Application Performance
Management

Intelligent anomaly detection



Your environment is healthy No problems within the selected timeframe

Dynatrace's artificial intelligence engine continuously monitors every part of your application, learns its typical behavior and informs about problems. Every problem shows impact, root cause and provides replay for rapid resolution.

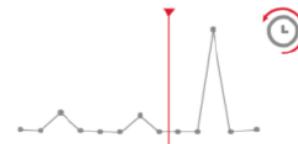
Receive problem notifications on your smartphone with [Dynatrace mobile apps](#).



Dynatrace automatically starts to detect anomalies based on the typical baseline of your applications' performance, traffic and failures.



Detected problems show their impact and root cause, based on automatically discovered dependencies within your system.



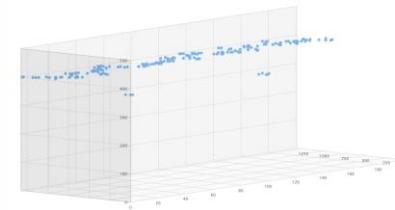
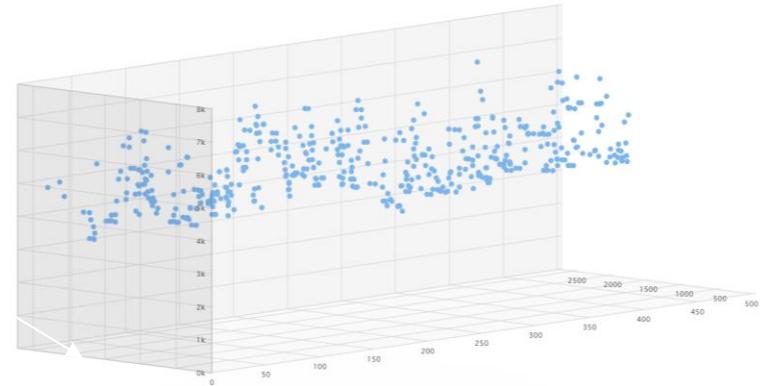
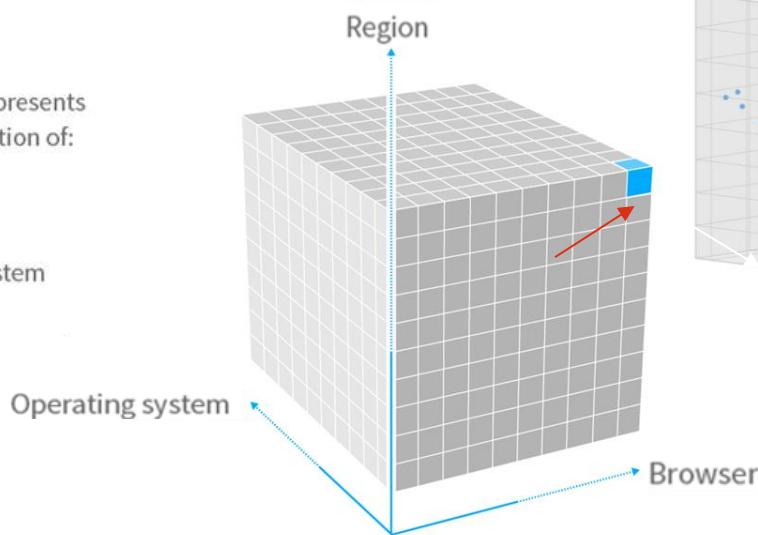
Replay complex anomalies to rapidly triage and resolve problems.

Fully automatic multidimensional baselining - Applications

Multidimensional baselining

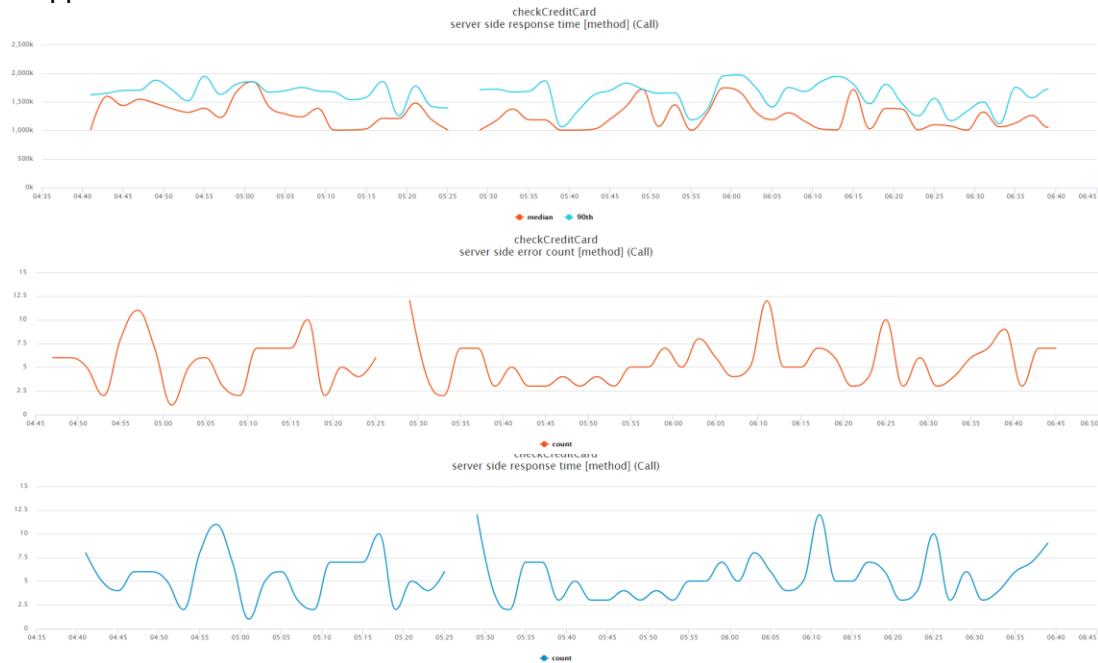
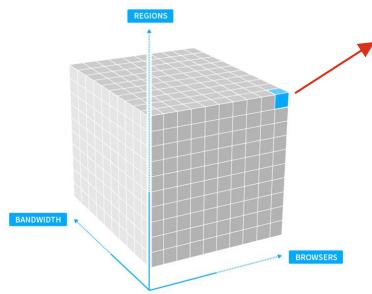
Each sub-cell represents
a unique correlation of:

- Region
- Browser
- Operating system
- User action

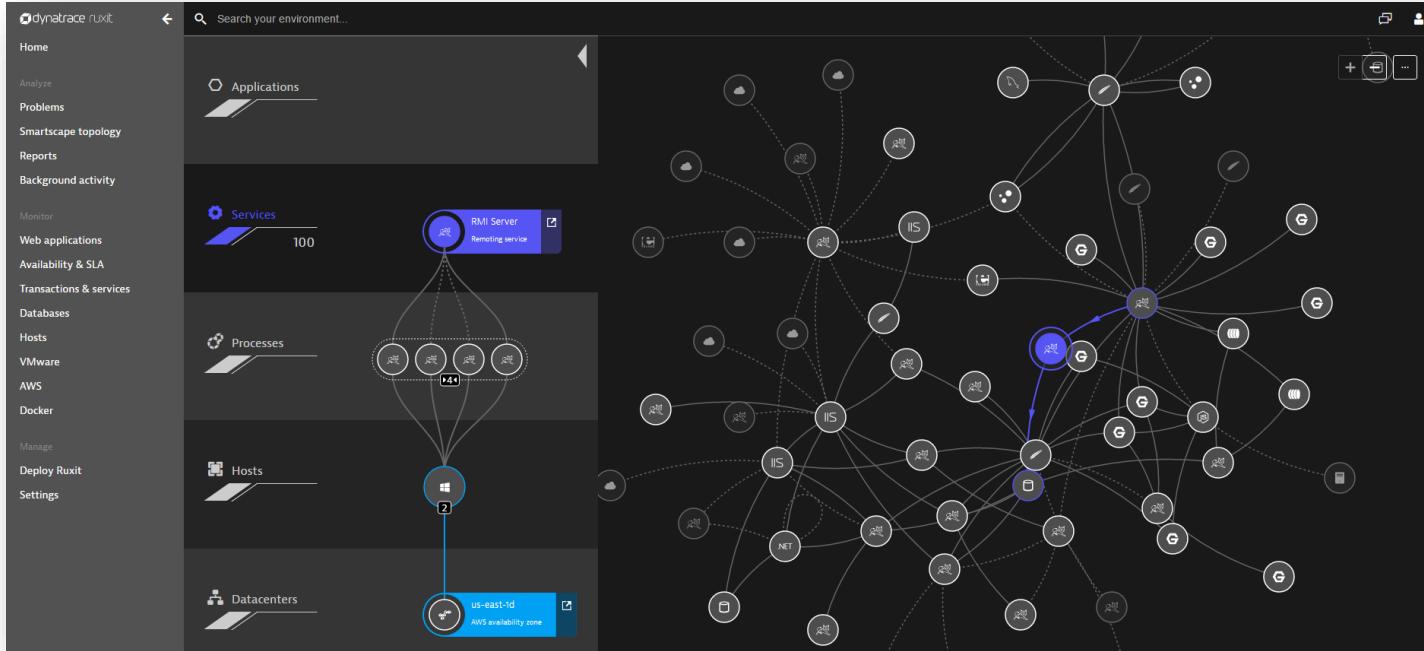


Fully automatic multidimensional baselining - Services

- Response times, error rates and application traffic

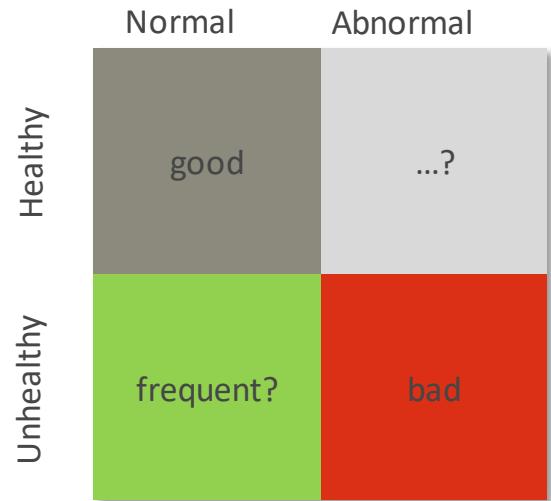


Automatic detection of topology and communication relation



Frequent issue detection

- Hypothesis: There are unhealthy situations that are normal for ops
 - Unimportant disk is full since several weeks
 - Regular backup process triggers CPU spikes
 - ...
- Dynatrace detects such regular events
 - On basis of a daily and weekly moving window
 - Once discovered only notify user if severity increases
- Frequent issue detection uses one week moving window
 - One week without frequent issue means reset to start



Problem pattern catalog

1. Applications

- 1.1 Unexpected high traffic
- 1.2 Unexpected low traffic
- 1.3 User action duration degradation
- 1.4 Javascript error rate increase
- 1.5 Multiple application problems
- 1.6 Application web check performance threshold violation
- 1.7 Application availability error detected by web check

2. Services

- 2.1 Response time degradation
- 2.2 Failure rate increase
- 2.3 Multiple service problems

3. Synthetic only

- 3.1 Web check performance threshold violation
- 3.2 Web check availability error

4. Infrastructure

- 4.1 CPU saturation
- 4.2 Memory saturation
- 4.3 Slow storage
- 4.3 Insufficient queue depth
- 4.4 Overloaded storage
- 4.5 Network congestion
- 4.6 Host or monitoring unavailable
- 4.7 High network utilization
- 4.8 Multiple infrastructure problems

Dynatrace AI 2.0

Automatic metric change point detection for causation

- Metric and event-based detection of abnormal component state
- Seamless integration of custom metrics within the Dynatrace AI process
- Third-party event ingests
- Availability root-cause
- Grouped root-cause

Root cause
Based on our dependency analysis all incidents have the same root cause.

 CheckDestination Service Instance	
1 Deployment CheckDestination v123.321	
Entity Time Approver Build Number CI Git commit Owner Project Remediation Source Version	CheckDestination today, 09:00 - 09:00 (-) Alice McBright (alice.mcbright@easytravel.com) 1.223.23432 http://a-tower.local/job/DeployJob/38/artifact/build/Deployment-v1.23.321.zip e5a6baac7eb Jason Miller (jason.miller@easytravel.com) CheckDestination http://a-tower.local/job/RemediateJob/38/ ServiceNow 1.23.321

Metric anomalies detected
Review the metrics below which show abnormal or outlying behaviour in the cause of the actual problem.

Response time + 56.9 ms Now  Before 	Failure rate + 0.49 % Now  Before 
---	---

Analyze findings



Problem Demo



Maintenance Windows



Application Performance
Management

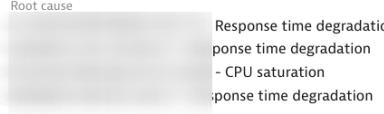
Maintenance Windows

Manage planned downtimes

- **Two types of maintenance windows**
 - Planned
 - Added in advance
 - Unplanned
 - Added retroactively to notify Dynatrace of unexpected downtimes that should not be included into overall performance and availability metrics
- Once a maintenance windows is defined, Dynatrace excludes the time period from its baseline calculation
- Can be defined manually or via the Dynatrace API

Maintenance Windows

3 problems matching selected filter from Mar 31 22:33 - Apr 7 22:33

 3 services Response time degradation <small>Problem 599</small>	 tomcat-nwi-ms on port 9489 Response time degradation <small>Problem 673</small>	 Unexpected low traffic <small>Problem 647</small>
<p>From 05:59 - 06:10 for 11 minutes</p> <p>Maintenance window</p> <p>⌚ Problem occurred during maintenance window</p> <p>Impacted services</p>  <p>Impact</p> <p>1.25k requests/min affected</p> <p>Root cause</p>  <p>Response time degradation</p> <p>ponse time degradation</p> <p>- CPU saturation</p> <p>:ponse time degradation</p>	<p>From Apr 2 06:09 - Apr 2 06:33 for 24 minutes</p> <p>Maintenance window</p> <p>⌚ Problem occurred during maintenance window</p> <p>Impacted services</p> <p>on port 9489</p> <p>Impact</p> <p>3.22k requests/min affected</p> <p>all dynamic requests</p> <p>Root cause</p> <p>on port 9489 - Response time degradation</p>	<p>From Apr 1 05:11 - Apr 1 05:19 for 8 minutes</p> <p>Maintenance window</p> <p>⌚ Problem occurred during maintenance window</p> <p>Impacted applications</p> <p>Impact</p> <p>17.2 user actions/min affected</p> <p>user action Xhr</p> <p>Root cause</p> <p>- Unexpected low traffic</p>

Kaffee Pause



Application Performance
Management

Dashboards

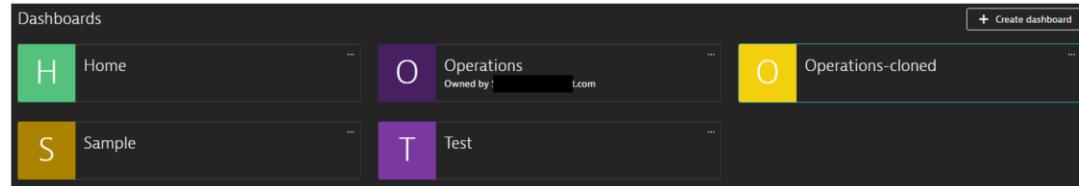


Application Performance
Management

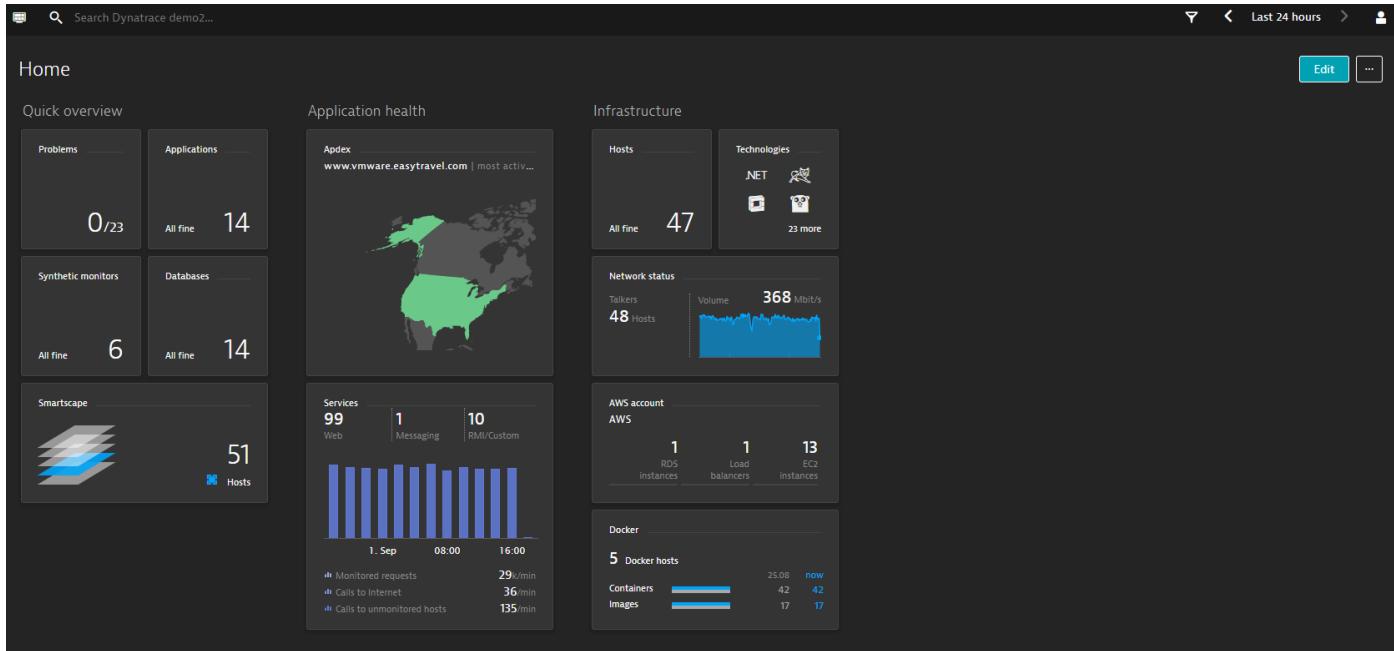
Dashboards

Possibilities

- View dashboards
- Create a custom dashboard
 - Tiles
 - Custom charts
 - Pin to dashboard
- Sharing and Cloning



Standard Dashboard



Dashboarding Demo



Dynatrace Entwickler Schulung

Ende Tag 1 – Q & A

Patrick Hofmann

Cloud / Virtualization Monitoring



Application Performance
Management

DEMO





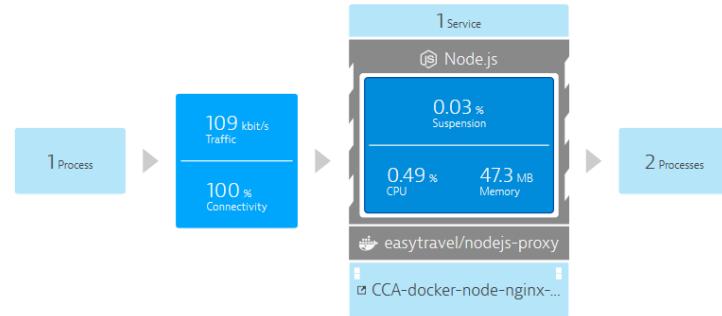
Can I monitor Docker?

- **Full-featured Docker monitoring**
 - The same monitoring functionality for Dockerized applications that is available for non-containerized applications.

JS MicroJourneyService (running in mesos-d00c7bcf-fee2-474c-a...)

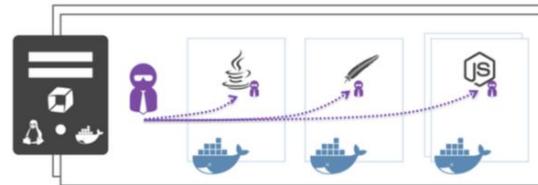
Properties

Type	Node.js (4.2.4)
Process group	MicroJourneyService
Ports	31793
Bitness	64-bit
Application version	4.2.4
Docker container	mesos-d00c7bcf-fee2-474c-ae97-9be9639027b4-50.013a0bbc-b7b5-4bc1-ae2d-2bf...
Docker image	easytravel/nodejs-proxy
Node name	running in mesos-d00c7bcf-fee2-474c-ae97-9be9639027b4-50.013a0bbc-b7b5-4bc...
Container	mesos-d00c7bcf-fee2-474c-ae97-9be9639027b4-50.013a0bbc-b7b5-4bc1-ae2d-2bf...
Node.js application	MicroJourneyService



Get started with Docker

- Seamlessly integrates with Docker environments and automatically monitors your containerized applications and services
- Hooks into containers and provides code for injecting the agent into containerized processes
- No need to modify Docker images, modify run commands, or create additional containers to enable Docker monitoring
- Simply install Dynatrace OneAgent on your hosts that serve containerized applications and services
- Dynatrace automatically detects the creation and termination of containers and monitors the applications and services contained within those containers



Docker Dashboard Demo





Deploying OneAgents



Application Performance
Management

Option 1

Install OneAgent on cluster nodes



OPENSSHIFT[®]
by Red Hat[®]

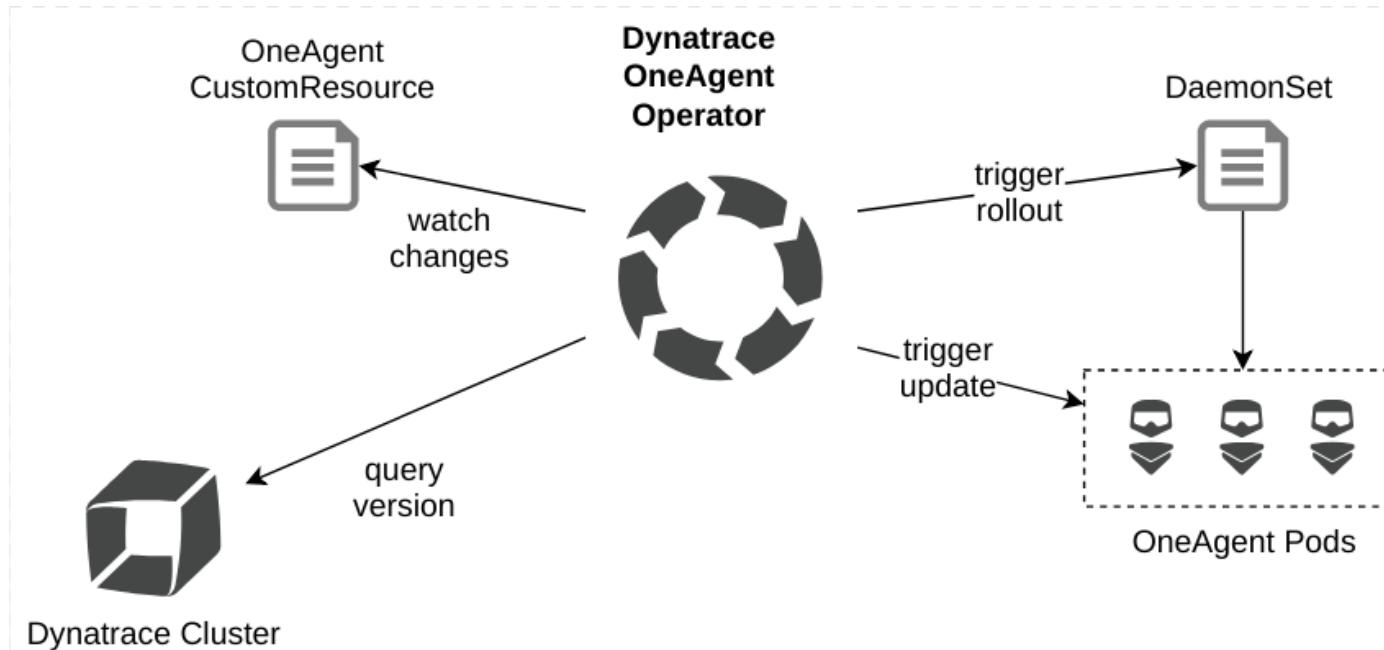


kubernetes
by Google[®]



Option 3

Dynatrace OneAgent Operator



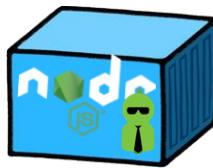
Deploying OneAgents for PaaS monitoring



Application Performance
Management

PaaS Monitoring

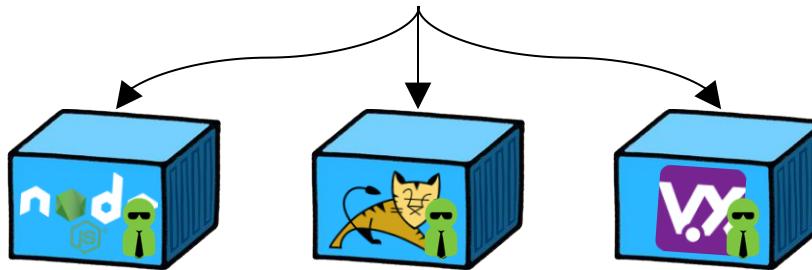
Install OneAgent inside containers



PaaS Monitoring

Install OneAgent inside containers

oc|s2i|Dockerfile



K8s Dashboard Demo



Synthetic Monitors



Application Performance
Management

Simple Availability & Complex Clickpath Checks

Synthetic > Add monitor

Create a new synthetic monitor

Synthetic monitors provide 24x7 visibility into the performance and availability of your web application.

URL of site you want to monitor
For example, <http://www.mysite.com>

Name this monitor
For example, My Web Site

Device profile

Device: Desktop | Screen size (pixels): 1024 x 768 px | Mobile device:

Bandwidth: No throttling | User agent: Default Dynatrace user agent

What type of synthetic monitor do you want to create?

 Availability

A browser monitor is a simulated page view with a real browser. Browser monitors tell you when your web application is inaccessible or when performance has degraded below baseline performance.

 Clickpath

Browser clickpath monitors are simulated user sessions that monitor your web applications business critical workflows. Use the Dynatrace Recorder to record an exact sequence of clicks and user input that you are interested in monitoring for availability and performance. [More...](#)

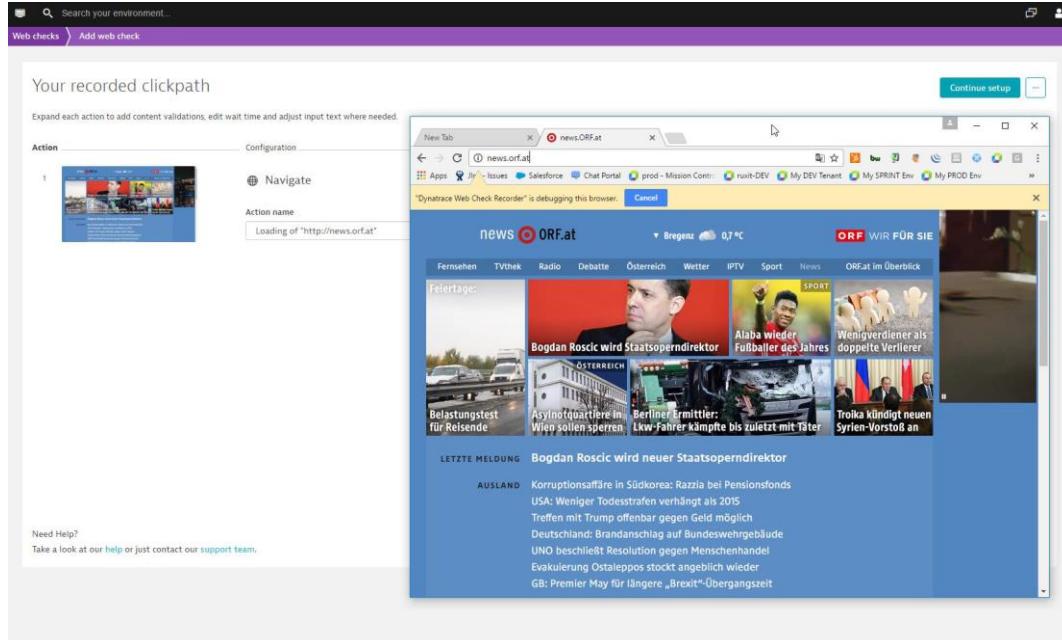
[Set up browser monitor](#) [Set up browser clickpath](#)

Execution in real browser

- Chrome browser used
 - Constantly updated to current version
- Empty browser caches on execution



Recorder-Plugin



Unique locations

59 locations selected

Filter by cloud provider, continent, country, city or by IP address

Hide inactive locations

<input checked="" type="checkbox"/> Location name	Type	Country	Continent	IP address	Cloud
<input checked="" type="checkbox"/> Johannesburg (southafricanorth) Beta	Public	South Africa	Africa	102.333.162.78, 102.333.67.158	Azure
<input checked="" type="checkbox"/> Beijing (cn-beijing) Beta	Public	China	Asia	39.105.201.46, 39.105.174.59	Alibaba
<input checked="" type="checkbox"/> Hohhot (cn-huhuaeast) Beta	Public	China	Asia	39.104.176.252, 39.104.115.180	Alibaba
<input checked="" type="checkbox"/> Zhangjiaokou (cn-zhangjiaokou) Beta	Public	China	Asia	39.98.228.53, 39.98.228.106	Alibaba
<input checked="" type="checkbox"/> Busan (koreasouth)	Public	South Korea	Asia	52.231.207.31, 52.231.200.125	Azure
<input checked="" type="checkbox"/> Hangzhou (cn-hangzhou) Beta	Public	China	Asia	47.99.90.105, 120.272.08.148	Alibaba
<input checked="" type="checkbox"/> Hong Kong (cn-hongkong) Beta	Public	China	Asia	47.75.85.239, 47.52.140.200	Alibaba
<input checked="" type="checkbox"/> Hong Kong (eastasia)	Public	Hong Kong	Asia	52.229.171.146, 52.229.169.134	Azure
<input checked="" type="checkbox"/> Osaka (ap-northeast)	Public	Japan	Asia	104.215.201.07, 104.215.13.77	Azure
<input checked="" type="checkbox"/> Seoul (ap-northeast-2)	Public	South Korea	Asia	13.125.78.12, 13.124.179.192	Amazon EC2
<input checked="" type="checkbox"/> Seoul (koreacentral)	Public	South Korea	Asia	52.231.79.75, 52.231.79.70	Azure
<input checked="" type="checkbox"/> Shanghai (cn-shanghai) Beta	Public	China	Asia	47.100.180.148, 47.100.300.59	Alibaba
<input checked="" type="checkbox"/> Shenzhen (cn-shenzhen) Beta	Public	China	Asia	47.106.127.87, 119.231.106.61	Alibaba
<input checked="" type="checkbox"/> Tokyo (ap-northeast-1)	Public	Japan	Asia	54.95.30.59, 13.112.243.168	Amazon EC2
<input checked="" type="checkbox"/> Tokyo (japaneast)	Public	Japan	Asia	40.115.156.100, 40.115.136.49	Azure
<input checked="" type="checkbox"/> Kuala Lumpur (ap-southeast-3) Beta	Public	Malaysia	Asia	47.254.216.179, 47.254.196.172	Alibaba
<input checked="" type="checkbox"/> Singapore (southeastasia)	Public	Singapore	Asia	13.635.225, 104.215.190.123	Azure
<input checked="" type="checkbox"/> Chennai (southindia)	Public	India	Asia	52.172.44.155, 52.172.24.388	Azure
<input checked="" type="checkbox"/> Mumbai (ap-south-1)	Public	India	Asia	13.233.243.185, 13.232.358.216	Amazon EC2
<input checked="" type="checkbox"/> Mumbai (westindia)	Public	India	Asia	104.211.187.32, 104.211.188.3186	Azure
<input checked="" type="checkbox"/> Pune (centralindia)	Public	India	Asia	52.172.202.148, 52.172.200.59	Azure
<input checked="" type="checkbox"/> Dubai (me-east-1) Beta	Public	United Arab Emirates	Asia	47.913.104.102	Alibaba
<input checked="" type="checkbox"/> Singapore (ap-southeast-1) Beta	Public	Singapore	Asia	47.88.175.243, 47.24.15.34	Alibaba
<input checked="" type="checkbox"/> Singapore (ap-southeast-1)	Public	Singapore	Asia	52.721.14.186, 13.350.162.25	Amazon EC2
<input checked="" type="checkbox"/> Cardiff (ukwest)	Public	United Kingdom	Europe	51.141.36.199, 51.140.224.57	Azure
<input checked="" type="checkbox"/> Dublin (eu-west-1)	Public	Ireland	Europe	34.243.169.142, 34.242.37.100	Amazon EC2

Waterfall-Analysis

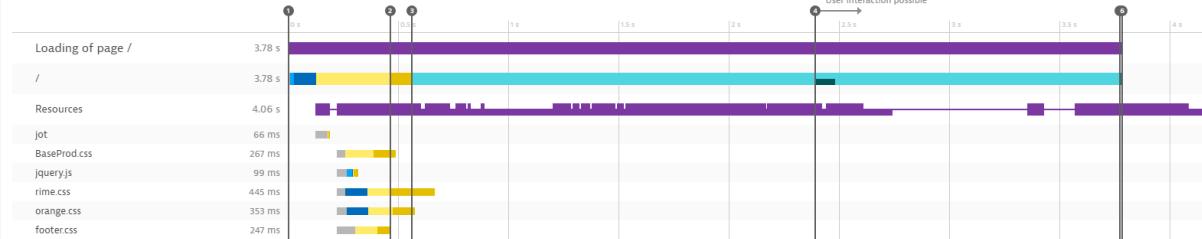
Action: Loading of "http://ec2-54-174-224-100.compute-1.amazonaws.com:8079/"

3.78 s action duration



Detailed breakdown
for Loading of page /

Display full waterfall ▾



Screenshot on error

Analyze a specific web check run

Select web check run for detailed analysis | Dec 21, 08:22:00

Clickpath action failure on click on "Book Now"

Select action for detailed breakdown:

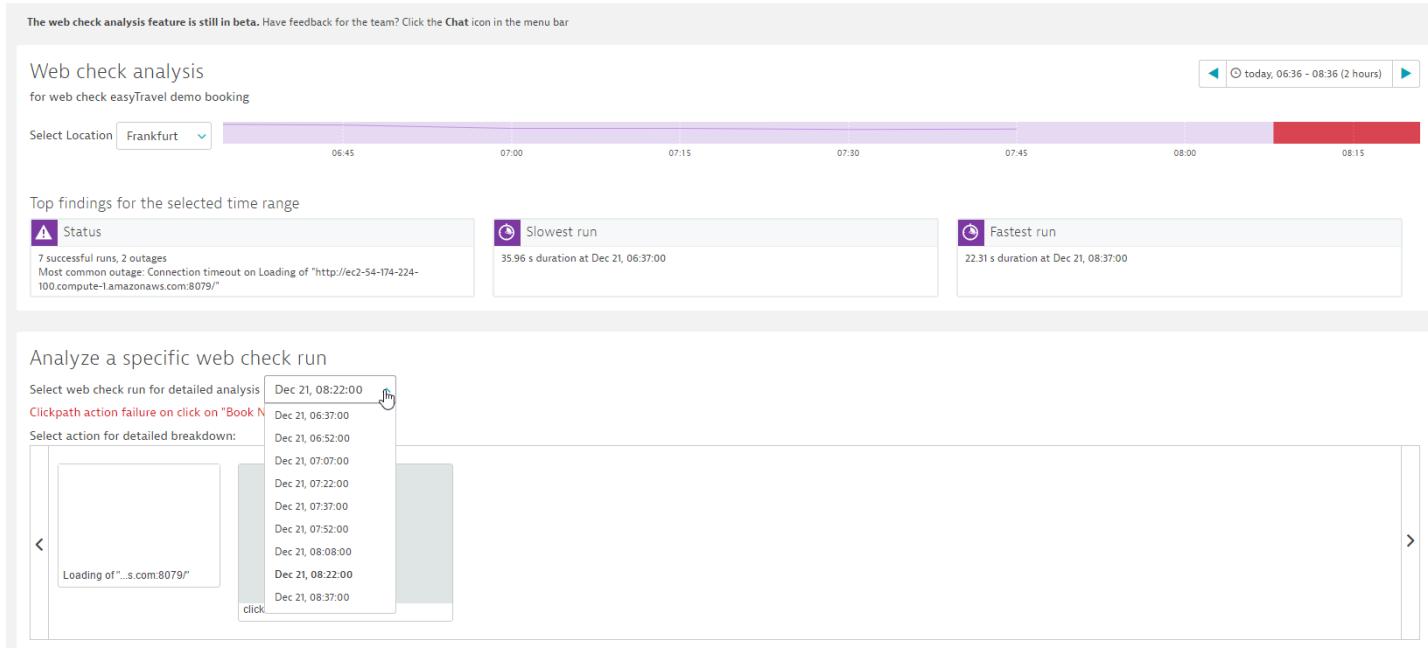
The screenshot displays a user interface for analyzing a web check run. At the top, a dropdown menu shows the selected date and time: 'Dec 21, 08:22:00'. Below it, a red error message reads 'Clickpath action failure on click on "Book Now"'. A section titled 'Select action for detailed breakdown:' contains two small screenshots. The left one, labeled 'Loading of "...s.com:8079/"', shows a white page with a progress bar. The right one, labeled 'click on "Book Now"', shows a gray placeholder box with the text 'No image available'. Navigation arrows are present on both sides of these boxes.

Action: click on "Book Now"

-- action duration

Below the analysis section, there is a comparison between a 'Failure' screenshot and an 'Expected result'. The 'Failure' screenshot shows a large gray area with the text 'No image available' and a 'Failure' label at the bottom. The 'Expected result' screenshot shows a travel booking website with a login form and a scenic landscape image of a road through a valley. A label 'Expected result' is at the bottom of this screenshot.

Run History



Synthetic Demo



Real User Monitoring Settings



Application Performance
Management

Agenda

- Application Rules
- Content Resources
- Geographic Regions
- Source maps & symbol files
- Data Privacy
- Custom User Actions
- User Session Export

RUM Settings



RUM settings

Overview

The screenshot shows the 'Application rules' section of the Dynatrace interface. The left sidebar has a tree view with 'Monitoring' expanded, showing 'Web and mobile monitoring' selected. Under 'Web and mobile monitoring', 'Application rules' is also selected. Other items include 'Source maps & symbol files', 'Geographic regions', 'Map IP addresses to locations', 'IP determination for auto-injected ...', 'Content resources', 'Provider breakdown', 'Resource URL cleanup rules', 'Resource types', 'Real user monitoring', 'Application rules' (selected), 'Data privacy', and 'Custom user actions'. Below this is a section for 'Cloud and virtualization' with 'Connect vCenter, OpenStack or AWS'.

Applications

The main content area is titled 'Applications'. It contains a descriptive text about customizing domain names and sub-domains for monitoring, mentioning the 'My web application' placeholder. It also discusses 'auto-injected applications' and how to configure them.

Create custom grouping rules

This section allows defining rules for grouping URLs into distinct applications. A button 'Create custom grouping rule' is available. Several rules are listed:

- www.vmware.easytravel.com : All URLs starting with <http://192.168.238.156:8079>
- image gallery : All URLs starting with http://ec2-54-173-46-56.compute-1.amazonaws.com/image_gallery
- image gallery : All URLs starting with http://54.173.46.56/image_gallery
- Madison Island : All URLs starting with <http://ec2-54-173-46-56.compute-1.amazonaws.com:8079>

RUM settings

Application Rules

- Default configuration
- Grouping rules
- Domain detection

Applications

The settings on this page enable you to customize how the domain names and sub-domains that Dynatrace detects in your environment should be recognized and grouped into distinct applications for monitoring. You can even define applications based on URL patterns by adding custom grouping rules. By default, Dynatrace associates all detected user actions with a placeholder application called 'My web application'.

Settings for auto-injected applications

These settings only apply to applications detected by the OneAgent. You need to [deploy the OneAgent](#) and [enable real user monitoring](#).

Configure my default web application

Change the configuration settings for the default application here: [My web application](#)

Create custom grouping rules

Define your own rules for grouping URLs into distinct applications for monitoring. [More...](#)

Application names have to be unique. You can check here for your already defined [agentless applications](#).

[+ Create custom grouping rule](#)

[www.vmware.easytravel.com](#) : All URLs starting with <http://192.168.238.136:8079>



[image gallery](#) : All URLs starting with http://ec2-54-173-46-56.compute-1.amazonaws.com/image_gallery



[image gallery](#) : All URLs starting with http://54.173.46.56/image_gallery



RUM settings

Grouping Rule

Create custom grouping rule

Choose an application input:

Provide a name for the application that this rule detects:

Enter new application name 

Select existing application

Enter new application name

Global ID for this application. You can create multiple grouping rules for the same application by c

Define which web requests are part of this application:

If the domain (host) 

matches... 

Example, mybookshop.com

Example:

If the domain matches **mybookshop.com** then the following URLs will be grouped into this web application: <http://www.mybookshop.com/forum/index.html>

Save

Cancel

RUM settings

Domain detection

Identify domains using HTTP request headers

Domains are trying to be detected automatically. Because web servers commonly operate behind firewalls, fully qualified domain names are often not displayed in browser address fields. [More...](#)

 Create HTTP header

X-Forwarded-Host (default)



X-Host (default)



Host (default)



 Restore default headers

Web Applications



Applications Settings

The screenshot shows the Dynatrace Monitor interface. A large red arrow points from the 'Applications' section in the top navigation bar down to the list of applications below. The list is titled '10 Applications' and includes the following items:

- www.easytravel.com
- www.openstack.easytravel.com
- Madison Island
- www.weather.easytravel.com

Each item has a small globe icon next to it. To the right of the application names is a purple sidebar with three buttons: 'Edit', 'Show top findings', and 'Smartscape view'. Below the sidebar, there is a note about active sessions and conversion goals.

The screenshot shows the 'Application settings' page for 'www.easytravel.com'. The URL 'www.easytravel.com' is in the address bar. The page title is 'Application settings' and the sub-section is 'General'. There are several configuration sections:

- General**: See your application setup.
- User actions**: Generate consistent user action names.
- Conversion goals**: Measure success against business goals.
- User tags**: Use metadata to analyze user sessions.
- Async requests and single page apps**: Select your frameworks.
- Content capture**: Define resource monitoring.

Below these sections is a section titled 'Apdex user satisfaction threshold settings' with a note: 'Apdex is a universal standard for measuring customer satisfaction with application performance.' It features a circular gauge for setting thresholds, with values 6 and 8 marked. The gauge has three segments: green (good), yellow (tolerable), and red (frustrating). Below the gauge, there are three performance ranges:

- < 6.8 s: Satisfactory performance
- 6.8 s - 27.2 s: Tolerable performance
- > 27.2 s: Frustrating performance

At the bottom, there is a note about JavaScript errors: 'Consider JavaScript errors in Apdex calculations' and 'Treat user actions with JavaScript errors as erroneous and rate their performance as "Frustrating". Turn off this setting if you're sure that the JavaScript errors don't affect your customers.'

Applications Settings

General

- Apdex

Applications > www.easytravel.com > Settings > General

Application settings
www.easytravel.com

General
See your application setup

User actions
Generate consistent user action names

Conversion goals
Measure success against business goals

User tags
Use metadata to analyze user sessions

Async requests and single page apps
Select your frameworks

Content capture
Define resource monitoring

Anomaly detection
Fine-tune the sensitivity and thresholds

Advanced setup
Fine-tune your real user monitoring

Detected by rule: URL starts with http://ec2-54-84-118-207.compute-1.amazonaws.com
URL starts with http://54.84.118.207:8079
URL starts with http://ec2-54-174-83-27.compute-1.amazonaws.com

[Change detection rules](#)

Real user monitoring
With real user monitoring enabled, Dynatrace gathers details about load times and page behavior that your customers experience with your application. Only applications with injected JavaScript tags can be monitored.

Apdex user satisfaction threshold settings
Apdex is a universal standard for measuring customer satisfaction with application performance. [More...](#)

Set thresholds for Apdex user satisfaction ranges



Performance Level	Threshold Range
	< 6.8 s
	6.8 s ~ 27.2 s
	> 27.2 s

Satisfactory performance Tolerable performance Frustrating performance

Consider JavaScript errors in Apdex calculations
Treat user actions with JavaScript errors as erroneous and rate their performance as 'Frustrating.' Turn off this setting if you're sure that the JavaScript errors don't affect your customers.

Applications Settings

User Action Names

- **By default, load action names are based on underlying HTML page name**
 - *Loading of page index.html*
- **XHR action names are based on the URL**
 - *http://example.com/api/v1/login*
- **Alternatively they can be based on user input + element caption**
 - click on “search”

Default user action names

Define how load and XHR user actions should be named. [More...](#)

Default name for load actions: 

Default name for XHR actions: 

Applications Settings

User Action Name - cleanup

- **URL cleanup rules**
 - Remove parameters via regex
- **Naming rules**
 - Custom names for certain patterns via regex
- **Extraction rules**
 - Naming by extracting parts from URL, Page title, Action name, Page oath or Content of meta tag

Add user action naming rule

Without consistent naming, user actions can be difficult to track. Here you can set up detection rules that instruct Dynatrace on how to ...
 Enable rule

Rule applies to:

Load actions 

Resulting user action name:

For example, Product detail page

This field cannot be empty.

Rule:

Action name  begins with 

Action name

Page title

Page URL

Page path

CSS selector

JavaScript variable

Content of meta tag

rule

Cancel

The user actions that match this rule are listed below. Affected user actions are displayed at the top of the list.

Filter for...

Type Action name

Resulting user action

Loading of page /

Loading of page /

Not matched by any rule. [More...](#)

Loading of page /logout.jsf

Loading of page /logout.jsf

Applications Settings

Conversion goals

- **Multiple Types**
 - Destination
 - User Action name/URL
 - Session Duration
 - Number of user actions

Conversion goals

Conversion goals are a versatile way of measuring how well your application fulfills your company's objectives. You can measure session length, or number of actions per session. You can then compare the conversion rates of various goals.

Name:

Type of goal:

Rule applies to:

Rule:

For example, click on

Case sensitive

Successful Bookings - open final Page : Destination begins with /orange-booking-finish.jsf

Applications Settings

User tags – Identify users

- Meta data
 - CSS selector
 - JavaScript variable
 - Meta tag
 - Cookie value
- Manual
 - JavaScript API

User tag

Set up rules to automatically tag the users of this application with metadata pulled from cookies, JavaScript variables, or other elements to gain deeper insight into the individual journeys of each of your customers. Tags for individual users – which may be comprised of user names, user IDs, or other metadata captured via a browser or CSS selector – can dramatically enrich the user experience insights you gain from [User session analysis](#).

Alternatively, you can use the JavaScript API call `dtrum.identifyUser()` to tag each user session with user-specific metadata such as a username or JavaScript variable. For details, see [JavaScript API](#).

Note: When multiple tags match a single user, only the first tag is used. Subsequent tags are ignored. Sort your tags to increase the order.

Add tag (identifier) rule

Select the expression type that includes the metadata you want to use for this user tag rule. [More...](#)

Expression type to capture: CSS selector

CSS selector 

CSS selector

JavaScript variable

Meta tag

Cookie value

Server side request attribute 

 Apply cleanup rule 

Add tag (identifier) rule

Cancel

Expression type name

Expression type

Move up/down

Delete Edit

No data

Applications Settings

XHR Detection

Async requests and single page apps

Dynatrace creates user actions when your app uses XMLHttpRequest or the Fetch API.

- Capture fetch() requests
- Support for XMLHttpRequest

⚠ Fetch API support requires Dynatrace OneAgent version 1.119 or later!

JavaScript framework support

Some JavaScript frameworks provide high-level abstractions for sending web requests that are more intuitive and accurate.

- AngularJS and Angular
- Dojo
- ExtJS, Sencha Touch
- ICEfaces
- jQuery, Backbone.js
- MooTools
- Prototype
- ActiveXObject detection



www.easytravel.com

Properties, tags, and JavaScript frameworks

+ Add tag

Ajax frameworks in the last 12 hours



ICEfaces



jQuery



Prototype

Framework settings

Session properties

Enrich User Session with additional Metadata e.g.

- User Groups
- Promotions
- A/B test group
- Customer Status
- Order Sums
-

User Action Properties

- Same on User Action Level
- Coming soon

Both are included in RUM export

Session properties

Session properties can be used to enrich monitored user sessions with additional details that facilitate fine-grain user-session and application-performance analysis. You might create key/value pairs for

- purchases above a certain amount (for example, `ordersum = 100`),
- datacenter location (for example, `datacenter = south1`), or for
- a segment of users receiving a new promotion (for example, `abtesting = c`)

Once defined, such properties can be used for filtering your monitoring results across many Dynatrace analysis views. Such defined properties are also included with user sessions upon export.

Session property usage quotas

You can specify up to **10** string value properties, and **10** properties of the datatypes Date, Long and Double.

Quota amounts used:

- **0** of **10** string properties
- **0** of **10** numeric data type properties (Long, Double or Date)

Defined session property rules

Add session property rule

Key ▲	Data type	Expression type
		No data

Applications Settings

Advanced – Content resource settings

Content resource settings

These settings influence the monitoring data you receive for 3rd party, CDN and 1st party resources.

- Send W3C resource timing metrics for each resource file

By default, Dynatrace sends metrics for each individual resource, including each image, CSS and JavaScript file.

Disable this setting to calculate only aggregate metrics for each resource type.

- ▼ Configure thresholds for top findings

Warn about uncompressed resources larger than bytes.

Warn about resources that are larger than kB.

Warn about resources with a lower browser cache rate than %.

Warn about slow first party resources that have a response time larger than ms.

Warn about slow third party resources that have a response time larger than ms.

Warn about slow CDN resources that have a response time larger than ms.

Applications Settings

Advanced – Exclusion

- IP addresses
- XHR requests
- Browsers

Exclude/Include IP addresses from monitoring

Select the checkbox below if the IP addresses are to be included. Leave the checkbox unchecked if they are to be excluded.

Exclude the following IP addresses:

Type individual IP addresses or ranges of IP addresses separated by a dash (-) to exclude them from monitoring.

These are the only IP addresses that should be monitored.

Examples:

84.112.10.5, 84.112.10.8/24
127.0.0.1-127.0.0.255
fe80::10at:c6b2:5f68:785d

Exclude XHR requests from monitoring

Specify a regular expression to match all URLs that should be excluded from becoming XHR actions.

Enter a regular expression.

Exclude/Include browsers from monitoring

If you want to exclude certain outdated browser types from your list of monitored browsers, create browser exclusion rules for the browsers that are to be excluded.

[+ Add browser exclusion rule](#)

No browser rules defined!

Applications Settings

Advanced – JavaScript library

- Library source path
- Beacon path

JavaScript library

Specify the source path for placement of your application's custom JavaScript library file. By default, this path is set to the root directory of your web server. A custom source path may be necessary if your server operates behind a firewall.

Specify location for JavaScript library file:

For example, /serverpath/

Specify path where JavaScript tag should send monitoring data:

Specify either a relative or an absolute URL. If you enter an absolute URL, data will be sent using CORS.

For example, /beacon

Custom configuration properties

Here you can set additional JavaScript tag properties that are specific to your application. To do this, type key-value pairs defined using (=) and separated using a () symbol.

For example, cux=1

Request path ID settings

Type the path that is to be used to identify the server's request ID. [More...](#)

For example, somepath

Applications Settings

Advanced – JavaScript library

- Library source path
- Beacon path

JavaScript library

Specify the source path for placement of your application's custom JavaScript library file. By default, this path is set to the root directory of your web server. A custom source path may be necessary if your server operates behind a firewall.

Specify location for JavaScript library file:

For example, /serverpath/

Specify path where JavaScript tag should send monitoring data:

Specify either a relative or an absolute URL. If you enter an absolute URL, data will be sent using CORS.

For example, /beacon

Custom configuration properties

Here you can set additional JavaScript tag properties that are specific to your application. To do this, type key-value pairs defined using (=) and separated using a (|) symbol.

For example, cux=1

Request path ID settings

Type the path that is to be used to identify the server's request ID. [More...](#)

For example, somepath

Agentless RUM



Agentless RUM

How it works

- **Create a custom JavaScript tag through the Web Interface**
- **Copy the tag information**
- **Manually add the tag to each web page of the Application**
 - Script should be the first JavaScript file of the page
 - Script tag must be manually updated if RUM settings are changed

Mobile Applications



Mobile Applications

Android

Gradle

Select your platform to begin:



Google Android



Apple iOS

Select method for instrumentation:

Gradle

Command line

Modify your `build.gradle` script

Update your module's `build.gradle` script to include the `jcenter()` repository and apply the Dynatrace plugin (requires Android Plugin for Gradle version 1.5 or higher). The generated `applicationId` is unique for this mobile application.

```
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.dynatrace.tools.android:'  
    }  
}  
  
apply plugin: 'com.dynatrace.tools.android'  
dynatrace {  
    defaultConfig {  
        applicationId "2f21b47-0605-4193-a715-066628b95700"  
        environment "55db5012-0533-4b77-8fd9-e87fcde53"  
        cluster "https://SGW1.demoamasol.de:9999"  
        agentProperties 'DTXManagedCluster': 'true'  
    }  
}
```

Copy

You can now build and run your app using your IDE.

Android Studio 2.0 users:

Please disable the Instant Run setting. To do this, open the `Settings` and navigate to `Build, Execution, Deployment > Instant Run`. Then deselect the `Enable Instant Run` checkbox.

CLI

Select your platform to begin:



Google Android



Apple iOS

Select method for instrumentation:

Gradle

Command line

Add Dynatrace OneAgent for Android:

Download the [OneAgent](#) for Android.

Start the Auto-Instrumentor for your Android app:

Create a `Dynatrace.properties` file with the contents below:

```
DTXApplicationID=2f21b47-d605-4193-a715-066628b95700  
DTXAgentEnvironment=55db5012-0533-4b77-8fd9-e87fcde53  
DTXClusterURL=https://SGW1.demoamasol.de:9999  
DTXManagedCluster=true
```

Call the following command to instrument your app on Linux systems:

`instrument.sh apk=<AppName>.apk prop=Dynatrace.properties`

Call the following command to instrument your app on Windows systems:

`instrument.cmd apk=<AppName>.apk prop=Dynatrace.properties`

You can now install and run your app using your IDE.

Instrumented APK: <PathToAPK>/<AppName>/dist/<AppName>.final.apk

Sign and bitalign your app

Before your instrumented APK package can be uploaded to the app store it must be signed and signed with your private key.

Mobile Applications

iOS

Cocoapods

Select your platform to begin:



Google Android



Apple iOS

Select method of dependency management:

Cocoapods

Developer

Modify your Podfile

Add the Dynatrace OneAgent as a dependency within your Podfile.

pod 'Dynatrace'

Add application identification keys to your Info.plist file:

```
<key>DTXAgentEnvironment</key>
<string>b5db5012-0533-4b77-a8fd-d9e87fcde53</string>
<key>DTXApplicationID</key>
<string>2f21b47-d605-4193-a715-0b6628b95700</string>
<key>DTXClusterURL</key>
<string>https://SGW1.demo.amasol.de:9999</string>
<key>DTXManagedCluster</key>
</true>
```

You can now build and run your app using your IDE.

Xcode

Select your platform to begin:



Google Android



Apple iOS

Select method of dependency management:

Cocoapods

Developer

Add Dynatrace OneAgent for iOS:

Following [download](#), unzip the package and drag the framework or library into your iOS Xcode project. Select the Build Phases tab and add the framework or library to the Link Binary with Libraries section. Select the Build Settings tab and add the framework or library to the Framework Search Paths section. Finally, select the Build Phases tab and add the framework or library to the Link Binary with Libraries section.

Add required libraries to your Xcode linker settings:

1. Open the Xcode project navigator.
2. Select your project directory.
3. Select the Build Phases tab.
4. Expand Link Binary with Libraries.
5. Click the + button.
6. Select the following libraries:

CoreLocation.framework
CoreTelephony.framework
MessageUI.framework
Security.framework
SystemConfiguration.framework
WebKit.framework
libc++_tbd
libsqlite3_tbd
libz_tbd

Mobile Applications

Settings

- Name
- Reporting interval

Application name:

Report captured events every minutes.

Keep in mind that short reporting intervals (less than 2 minutes) may negatively impact your customers' device power consumption.

- User action
 - Cleanup
 - Naming
 - extraction

User actions

Dynatrace can distinguish between two user action types:

- User actions — The actions that users perform within your application. Each user action contains at least one web request.
- Web requests — These occur when a mobile app loads content through a HTTP request.

Cleanup rules are executed first, followed by naming rules. Extraction rules are executed last.

➤ 0 URL cleanup rules _____

➤ 0 naming rules _____

➤ 0 extraction rules _____

Preview your rule configuration

See how your current application rules will change upcoming user actions. Affected user actions are displayed at the top of the list.

RUM Settings



RUM settings

Provider Breakdown

- Change 1st, 3rd part and CDN resource mappings
- Add not automatically detected providers

Provider breakdown

Set up rules that define how your applications' downloaded content resources (images, CSS, 3rd party widgets, and more) are displayed and categorized for analysis. Dynatrace uses the provider host names of downloaded resources to categorize content resources into either 3rd party resources, CDN resources, or 1st party resources.

Dynatrace auto-detects over 1,000 content providers out-of-the-box, including Google, Amazon, Facebook, and many more. There's nothing you need to do to set up detection of resources. If you can't find your provider in the list below, you can add it manually.

Manually added providers



Auto detected providers



Filter for providers

Name	Domain name patterns	Resource type
123greetings.com	.123greetings.com	3 rd party
1800flowers.com	.1800flowers.com	3 rd party
1822direkt.com	.1822direkt.com	3 rd party

RUM settings

Geographic regions

- Custom geo locations definitions
 - Internal applications
 - Smartscape Datacenter layer
- Custom client IP header

IP determination for auto injected applications

These settings only apply to applications where the Dynatrace OneAgent automatically injects code.

Identify client IP addresses

Client IP addresses are automatically determined based on HTTP request header. If your cl

▲ Note that the format of the "Forwarded" header specified by RFC 7239 requires Dynatrac

[+ Create custom client IP header](#)

rproxy_remote_address (default)

True-Client-IP (default)

X-Client-Ip (default)

X-Cluster-Client-Ip (default)

X-Forwarded-For (default)

X-Http-Client-Ip (default)

CF-Connecting-IP (default)

Map IP addresses to locations

If you don't see performance data for some of your customers on the world map, it may be because those customers have private IP ad

[Set geographic region for IP](#)

[Import geographic regions from CSV file](#)

192.168.42.1 - 192.168.42.254 (Germany, Bayern, Munich)

192.168.46.1 - 192.168.46.254 (Germany, Bavaria, Munich)

192.168.53.91 - 192.168.53.92 (Germany, Bavaria, Munich)

1 unresolved IP address

Show unresolved IP addresses only

[Filter for IP address, country...](#)

IP address

80.239.139.130

Country

Unresolved (WAN)

77.188.40.61

Germany

192.168.53.91

Germany

RUM settings

Source Map & Symbol files

- In production JavaScript is often “minified” or transformed in some way to improve performance
- Transformed JavaScript sources are not human-readable, thus difficult to debug
- Source maps can be used to transform JavaScript files back to their original source
 - Dynatrace attempts to fetch source maps automatically
 - If not found, a source map can be uploaded manually

Stacktrace

Below you can see the stacktrace for this error. If you open a stackframe we will try to detect any files that can be used to better visualize and help you in analyzing this specific frame. In addition you will have the ability to upload your source maps and JavaScript files so that we can use these instead of any detected files.

Stackframe Status Analyze ↗

at Function.g.a.error (jquery-3.1.1.js : 1586/8) Mapped

We were able to use your uploaded source map and can show you the error location in the original file. If you want see a code snippet of the error location you can upload your original file or a zipped package of your source.

Minified file

Minified file http://... jquery-3.1.1.min.js
Location of JavaScript error in minified file 2/13470

Source map

Uploaded source map file jquery-3.1.1.min.map
Original file name jquery-3.1.1.js
Location of JavaScript error in original file 1586/8



RUM settings

Data Privacy

Protect your end users' data privacy

While Dynatrace provides comprehensive [data privacy & security controls](#) for the protection of your organization's end users, you may want to restrict data-capture even further for this application using the settings below.

Note: The default settings enable you to receive the full breadth and depth of Dynatrace Real User Monitoring analytics. For full details on available data-privacy settings, see [data privacy settings](#).

Global settings

Global data-protection settings defined for your environment are listed below. To adjust system-wide data-protection settings, see [global data-privacy settings](#).

- Mask end-user IP addresses & GPS coordinates
- Mask personal data in URLs
- Mask user actions (web applications only)

User tracking

- Use persistent cookies for user tracking

When enabled, Dynatrace places a persistent tracking cookie on all end-user devices to identify returning users.

Data-collection & cookie opt-in mode

To provide your end users with the ability to decide for themselves if their activities should be tracked to measure application performance and usage, enable cookie opt-in mode.

- Data-collection & cookie opt-in mode
- ▲ Requires Dynatrace OneAgent version 1.129 or higher

With **Data-collection & cookie opt-in mode** enabled, Real User Monitoring data (user session and user action data) isn't captured until `dtrum.enable()` is called for specific user sessions. With this approach, you can comply with your ability to decide if they want to have their data captured.

'Do Not Track' end-user browser settings

Most modern web browsers have a privacy feature called "Do Not Track" that individual users may have enabled on their devices. Customize how Dynatrace should behave when it encounters this setting.

- Comply with 'Do Not Track' browser settings
- Capture anonymous user-session data for "Do Not Track" enabled browsers
- Disable Real User Monitoring for "Do Not Track" enabled browsers

RUM settings

Custom User Actions

- **Add custom user actions via JavaScript API**
 - Add additional user actions (e.g. JavaScript only) or fine grained timings
- **No custom library required**
- **User action name is defined manually**
- **Disabling RUM could lead to JS errors**
 - Be sure to check if RUM agent is there

RUM settings

User Session Export

- **Export stream to leverage dynatrace RUM data within 3rd party**
 - Big-data analysis systems
 - BI warehouses
 - Other custom analysis
- **Stream includes RUM data including user actions, high-level timings and error data**
- **Can also be streamed directly to Elasticsearch**

RUM settings

User Session Export

- HTTP/HTTPS web hook
 - Build JSON payload
 - Either PUT or POST
- Elasticsearch
 - Name of Elasticsearch index where data is to be sent
 - Type of documents in the Elasticsearch index

The screenshot shows the Dynatrace Settings interface with the 'User session export' section highlighted. The left sidebar lists various monitoring and integration modules. The 'Integration' module is expanded, and its 'User session export' sub-module is selected and highlighted with a red box.

User session export

User session export enables you to stream real user monitoring data from Dynatrace.

Enable user session export

Endpoint URL
eg. http://example.com:1234/mypath

Use POST method (instead of PUT)
 Use new export format

Basic authentication settings

User name

Password

Dynatrace can automatically send bulk data to Elasticsearch. You can use an SSI.

Send data directly to Elasticsearch

Elasticsearch settings

Name of the Index where data is sent to
default: sessions

Type of documents in the Elasticsearch index
default: session

Sample Data

By using the link below you can request some sample data based on your user :

[Download sample snapshot from last week](#)

Workshop

RUM Settings

Application Rules

- Slice and dice Applications via URL patterns

Geographic Regions

- Define Locations for private IP address ranges

Workshop

Application Settings

XHR Detection

- Define utilized JavaScript Frameworks for XHR Detection

User Action Names

- Apply rules for better user action naming
 - Cleanup rules
 - Naming rules
 - Extraction rules

Conversion Goals

Workshop

Application Settings

Key User Actions

Apdex

Custom User Actions Example

Check Defaults

- Resource settings
- CDN Providers
- Real Client IP

Kaffee Pause



Application Performance
Management

Tagging



Application Performance
Management

Tagging

Add metadata to hosts, process groups, services and application

Use cases

- Filters
- Dashboards
- Alert notifications
- Management Zones

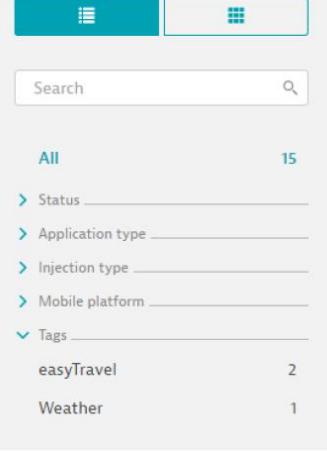
Can be done

- Manually
- In bulk
- Automatic
- Via Dynatrace API

Tagging

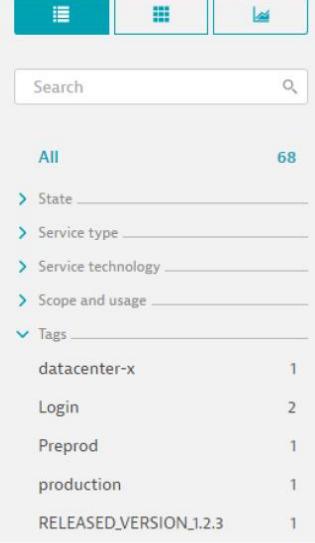
Filter for every scenario

Applications



Filter	Count
All	15
Status	_____
Application type	_____
Injection type	_____
Mobile platform	_____
Tags	_____
easyTravel	2
Weather	1

Services



Filter	Count
All	68
State	_____
Service type	_____
Service technology	_____
Scope and usage	_____
Tags	_____
datacenter-x	1
Login	2
Preprod	1
production	1
RELEASED_VERSION_1.2.3	1

Technology overview Process group eT-demo-2-BusinessBackend

eT-demo-2-BusinessBackend

Properties and tags

easyTravel X + Add tag

Type	_____
Ports	_____
Bitness	_____
JVM vendor	_____
JVM version	_____
Main class	_____
Node name	_____

Tagging

Filter notifications

Set up email integration

To:

CC

Subject

{State} Problem {ProblemID}: {ImpactedEntity}

Alerting profile

Default 

Select an [alerting profile](#) to filter the problem feed.

[Send test email](#)

[Save](#)

[Cancel](#)

Define who should receive the alert

Tags are built-in to Alerting Profiles

Rule

Availability alert (Immediate; All entities)

Problem severity level

Availability

Send a notification if a problem remains open longer than 0 minutes.

Filter problems by tag

Include all entities

Select which tags are included in the Alerting Profile

Tagging Settings

Demo

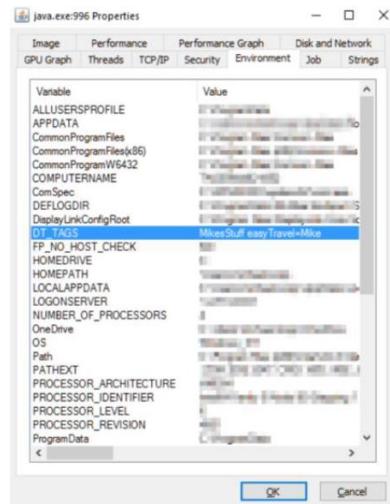


Application Performance
Management

More ways to tag

Environment variables

- Variable DT_TAGS at process or host level
 - Containing simple string or key/value pairs
 - Spaces are used to separate multiple tags



com.dynatrace.easytravel.launcher.jar easyTravel (x*)

Properties and tags

[Environment]MikesStuff [Environment]easyTravel: Mike

Type	Apache Tomcat
Technologies	Java (1.8.0_102-b14) and Java (Oracle HotSpot 1.8.0_102)
Process group	com.dynatrace.easytravel.launcher.jar easyTravel (x*)
Ports	1697
Bitness	64-bit
JVM vendor	Oracle HotSpot
JVM version	1.8.0_102-b14
Main class	com.dynatrace.easytravel.launcher.Launcher
Catalina base	C:/Program Files/dynaTrace/easyTravel (x*)
Executable	java.exe
Executable path	C:/Program Files/dynaTrace/easyTravel (x*)/jre/bin/java.exe
Jar file	com.dynatrace.easytravel.launcher.jar

More ways to tag

OpenShift

- Derive tags from OpenShift environment
- Automatically organize and filter all monitored OpenShift application components
- Specify labels in the Pod object definition or update labels of OpenShift resources with the command `oc label`
- Dynatrace automatically detects all labels attached to pods at application deployment time

The screenshot shows a Dynatrace process card for a pod named 'tomcat backend-23'. The card includes the following details:

- Name:** tomcat backend-23 (backend-23-106tm)
- Actions:** Analyze process connections, ...
- Properties and tags:**
 - [Openshift]OS: Atomic
 - [Openshift]app: backend
 - [Openshift]deployment: backend-23
 - [Openshift]deploymentconfig: backend
 - [Openshift]openshift_flavor: origin
- Type:** Apache Tomcat (8.5.15.0)
- Technologies:** Java (Oracle HotSpot 1.8.0_131), MongoDB Client (driver-core-only 3.2.2), MongoDB...
- Process group:** tomcat backend-23
- Ports:** 8080, 8009, and 8005
- Bitness:** 64-bit

More ways to tag API

Sample call to tag specific entity

```
https://{{id}}.live.dynatrace.com/api/v1/entity/services/{{YOUR_SERVICE_ID}}
```

with HTTP POST payload:

```
{  
  "tags": ["myTagthroughAPI", "secondTag"]  
}
```

```
{  
entityId: "HOST-D70EC6885E79D6C4",  
displayName: "gdn-rx-ub12-ci04v (maintained by PHP agent team)",  
customizedName: "gdn-rx-ub12-ci04v (maintained by PHP agent team)",  
tags: [  
{  
context: "USER",  
key: "opsTeamBoston"  
}]
```

Also able to see tags when pulling
information on any entity

Workshop

Tagging

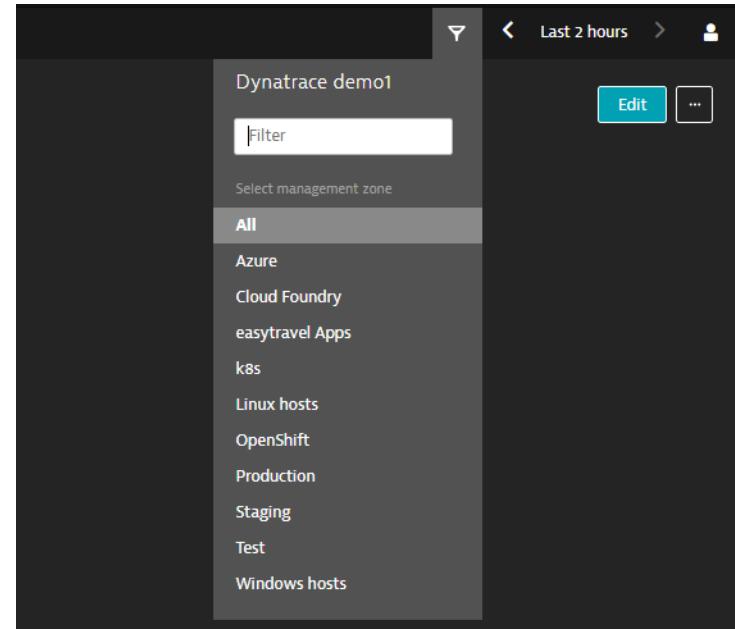
Ideensammlung

Konfiguration von Tags

Management Zones

What's a Management Zone?

- information-partitioning mechanism for
 - Applications
 - Services
 - Process Groups
 - Hosts
 - Synthetic Monitors
 - Technologies
 - ...
- **Management Zones may overlap**
- Usage: for focused analysis or granular permission management



Management Zones Configuration

- Bases on Tagging engine
- Selection of
 - Process Group
 - Host Group
 - Service Type
 - Technology
 - Service Topology
- Conditions for fine-tuning
- Applying to underlying or provided entities

Create management zone

Management zone name
myManagementZone

Add new rule

Active	Rule	Delete	Edit
	Services where Service type equals 'Web service' and Service technology equals 'Apache HTTP Server' and Service name contains 'amasol'		

Rule applies to Services of the following process group, service type, technology, and topology

Rule applies to entities matching the following properties

All process groups	All host groups
Web service	Apache HTTP Server
All service topologies	

Conditions

Service name	contains	<input checked="" type="checkbox"/> Case sensitive
amasol		

Add condition

Apply to underlying hosts of matching services Apply to underlying process groups of matching services

Preview

Matching entities

No matches detected

Management Zones

Permissions

- Ability to set different permissions for different management zones

Permissions

Select the permissions for this group.

Access environment	Environment::Management zone	Change monitoring settings	Download & install OneAgent	View logs	View sensitive request data	Configure request capture data
<input checked="" type="checkbox"/>	amasol	<input checked="" type="checkbox"/>				
<input type="checkbox"/>	amasol::amasol.de	<input type="checkbox"/>				
<input type="checkbox"/>	amasol::Demo	<input type="checkbox"/>				
<input type="checkbox"/>	amasol::easyTravel	<input type="checkbox"/>				
<input type="checkbox"/>	amasol::Production	<input type="checkbox"/>				
<input type="checkbox"/>	amasol::Test	<input type="checkbox"/>				

Management Zones vs. Environments

Management Zones

- Problem Detection and AI for the entire IT landscape
- may overlap
- Impossible to set quotas

Environments

- Problem Detection and AI divided for every environment; separate data retention
- can't overlap (OneAgent can only send data to one defined Environment)
- Possibility to set quotas for storage, host units & DEM units

Workshop

Management Zones

Ideensammlung
Konfiguration von Management Zones

Monitoring Settings



Application Performance
Management

Agenda

- **Monitoring overview**
 - Hosts
 - Process groups
 - Applications
- **Process Group Detection**
- **Monitored Technologies**

Monitoring Settings

Monitoring Overview

- Provides install links for
 - Agents
 - Security Gateways
 - RUM / Mobile
- Configure Monitoring on
 - Host level
 - Process group level
 - Application level

Settings

Monitoring

Setup and overview

Monitoring overview

Process group detection

Monitored technologies

Web and mobile monitoring

Global settings and configuration

Cloud and virtualization

Connect vCenter, OpenStack or AWS

Server-side service monitoring

Manage & customize service monitoring

Log analytics

Monitoring overview

The tabs below list the hosts, process groups and applications detected in your environment. To enable monitoring for any individual component, set the element's **Monitoring** switch to the **On** position.

Monitor another host | Install Security Gateway | Set up agentless or AMP monitoring | Monitor a mobile app

Hosts (63) | Process groups (129) | Applications (10)

Host	Summary	Host units	Monitoring: Off / On	Edit
angular-magento-fpm-mysql-nginx-docker	Monitored	0.25	<input checked="" type="checkbox"/>	<input type="button" value="edit"/>
BB1-apache-tomcatjms-iis	Monitored	0.5	<input checked="" type="checkbox"/>	<input type="button" value="edit"/>
BB2-apache-tomcatjms-iis	Monitored	0.5	<input checked="" type="checkbox"/>	<input type="button" value="edit"/>

Hosts



Hosts

- List of all hosts with OneAgent installed and running
- Turn monitoring on/off on host level
- Drill down to Host settings

Hosts (63)		Process groups (129)		Applications (10)	
Host	Summary	Host units	Monitoring: Off / On	Edit	
angular-magento-fpm-mysql-nginx-docker	UPDATE Monitored	0.25	<input checked="" type="checkbox"/>		
BB1-apache-tomcatjms-iis	UPDATE Monitored	0.5	<input checked="" type="checkbox"/>		
BB2-apache-tomcatjms-iis	UPDATE Monitored	0.5	<input checked="" type="checkbox"/>		
CCA-docker-node-nginx-mesos	UPDATE Monitored	0.25	<input checked="" type="checkbox"/>		
CCA-mysql-nginx-mesos-marathon	UPDATE Monitored	0.25	<input checked="" type="checkbox"/>		
CF1-tomcatjms	UPDATE Monitored	0.25	<input checked="" type="checkbox"/>		
CF2-tomcatjms	UPDATE Monitored	0.25	<input checked="" type="checkbox"/>		

Host

General

- Edit the name of the host
- Override global settings for Monitored Technologies

Hosts > LB-apache-php-varnish > Settings > General

Host settings
LB-apache-php-varnish

General

Host name: LB-apache-php-varnish
Auto detected name: et-demo-1-lnx1

Monitoring mode: Turn on monitoring to gain visibility into this host, its processes, services, and applications.

Detected processes: IIS module insights (Beta release. Requires OneAgent version 1115 or higher)
Note that all monitored IIS processes must be restarted to enable this setting.

Anomaly detection

OneAgent updates

Monitored technologies

The technologies CouchDB, Docker containers, Elasticsearch, HAProxy, Memcached, MongoDB, MS SQL, MySQL, Openstack Horizon, Openstack Keystone, Openstack libvirt, PHP-FPM, PostgreSQL, RabbitMQ, Redis, ActiveMQ JMX, Apache Hadoop HDFS, Apache Hadoop YARN, Apache Spark, AppServer JMX, Cassandra JMX, HornetQ JMX, Jetty JMX, Netflix OSS JMX, Solr JMX, WebSphere PMI Appserver, WebSphere PMI Connection Pools, Plugin manager, Log analytics, PHP, Java, Nginx, Node.js, Docker, Apache HTTP Server, and Network traffic are [enabled globally](#).

Technology	Type	Monitoring	Edit
Configure Couchbase	BETA Dynatrace plugin	<input checked="" type="button"/>	<input type="button"/>
Configure OpenStack notifications	BETA Dynatrace plugin	<input checked="" type="button"/>	<input type="button"/>
.NET Core	Service insights	<input checked="" type="button"/>	<input type="button"/>
FastCGI PHP processes launched by Apache HTTP Server	Service insights	<input checked="" type="button"/>	<input type="button"/>
Varnish Cache	BETA Service insights	<input checked="" type="button"/>	<input type="button"/>

Host

Detected processes

- View detected processes on that host
- Turn on/off service monitoring for each process

The screenshot shows the 'Host settings' page for the host 'LB-apache-php-varnish'. The left sidebar has tabs for 'General', 'Monitoring mode', 'Detected processes' (which is selected), and 'Anomaly detection' and 'OneAgent updates'. The main area displays a table of detected processes:

Process	Service monitoring: Off / On
api-remotecontrol.jar	<input type="button" value="Off"/>
com.dynatrace.easytravel.weblauncher.jar easytravel-*-**	<input type="button" value="On"/>
eT-demo-1-Frontend-LoadBalancer	<input checked="" type="button" value="On"/>
Varnish Cache	<input style="background-color: #f0f0f0; border: none; width: 100px; height: 20px;" type="button" value=" - "/>

Host

Anomaly Detection

- Turn on/off host connection problems
- Override global settings for Anomaly detection

The screenshot shows the 'Host settings' section for the 'Gitlab' host. The 'Anomaly detection' tab is selected. On the left, there are sections for General, Monitoring mode, Detected processes, OneAgent updates, Network, and Disk. The 'Anomaly detection' section is expanded, showing various detection options with checkboxes and dropdown menus for configuration.

Host anomaly detection settings	
<input checked="" type="checkbox"/>	Detect host or monitoring connection lost problems
<input checked="" type="checkbox"/>	Use global anomaly detection settings
<input type="checkbox"/>	Detect CPU saturation on host automatically
<input type="checkbox"/>	Detect high memory event on host automatically
<input type="checkbox"/>	Detect high GC activity automatically
<input type="checkbox"/>	Detect Java out of memory problem automatically
<input type="checkbox"/>	Detect high number of dropped packets automatically
<input type="checkbox"/>	Detect high network utilization automatically
<input type="checkbox"/>	Detect TCP connectivity problems for process automatically
<input type="checkbox"/>	Detect high retransmission rate automatically
<input checked="" type="checkbox"/>	Set specific thresholds per detected disk
<input type="checkbox"/>	Detect low disk space automatically
<input type="checkbox"/>	Detect slow-running disks automatically
<input type="checkbox"/>	Detect low inodes number available automatically

Process groups



Process groups

- List of all detected process groups
- Turn on/off Service monitoring on process group level
- Drill into process group settings

The screenshot shows a monitoring dashboard with three main tabs at the top: 'Hosts (63)', 'Process groups (129)', and 'Applications (10)'. The 'Process groups (129)' tab is selected and highlighted in blue. Below the tabs, there is a message: 'Looking for runtime information? Go to Deployment Status'. The main area displays a list of process groups, each with a small icon, the name, an 'Edit' button, and a toggle switch for 'Enable / disable deep monitoring'. The names of the process groups listed are: Apache Web Server apache*, Apache Web Server apache*, apl-remotecontrol.jar, AWS Lambda demo1WeatherBackend, and CollectGuestLogs.exe.

Process group	Enable / disable deep monitoring	Edit
Apache Web Server apache*	<input checked="" type="checkbox"/>	
Apache Web Server apache*	<input checked="" type="checkbox"/>	
apl-remotecontrol.jar	<input checked="" type="checkbox"/>	
AWS Lambda demo1WeatherBackend	<input checked="" type="checkbox"/>	
CollectGuestLogs.exe	<input type="checkbox"/>	

Process group

General

- Edit the name
- Give a description
- Turn on/off process monitoring
- Turn on/off RUM injection
- Turn on/off monitoring for a specific host

The screenshot shows the 'General' tab of the 'Process group settings' page. At the top, the breadcrumb navigation shows: Technology overview > Process group Apache Web Server apache* > Settings > General. The main title is 'Process group settings' with the subtitle 'Apache Web Server apache*'. On the left, there's a sidebar with 'General' selected and other options like 'Log monitoring' and 'Availability monitoring'. The right side contains several configuration sections:

- Process group name:** Apache Web Server apache* (Auto detected name: Apache Web Server apache*)
- Process group description:** A large text area labeled 'Description'.
- Automatically monitor newly found processes:** A toggle switch that is turned on (blue). A tooltip says: 'When enabled, all newly detected processes that belong to this group are monitored.'
- Automatically inject real user monitoring JavaScript tag:** A toggle switch that is turned on (blue). A tooltip says: 'Enable to inject JavaScript tag into each HTML request processed by this process group.'
- Enable process monitoring for specific hosts:** A section with a 'Host' input field containing 'OpenStackControllerNode' and a 'Monitoring' switch set to 'On' (blue).

A callout box on the right explains what a 'process group' is: 'A "process group" is a set of processes that perform the same function across multiple hosts. For example, you might have a cluster of servers with each server running the same process in support of multiple hosts.'

Process group

Log Monitoring

- Manually specify logs to be monitored
- Turn on/off automatically detected logs
- Turn on/off log monitoring for specific hosts

Technology overview > Process group Apache Web Server apache* > Settings > Log monitoring

Process group settings
Apache Web Server apache*

General

Add new log file for monitoring No custom log paths defined

Log monitoring

Auto-detected process log files

Availability monitoring

Process log name _____ Monitoring: Off / On

	/var/log/apache2/nova_placement_error.log.#	<input checked="" type="checkbox"/>
	/var/log/apache2/error.log.#	<input checked="" type="checkbox"/>
	/var/log/apache2/keystone_access.log.1	<input checked="" type="checkbox"/>
	/var/log/apache2/access.log.#	<input checked="" type="checkbox"/>
	/var/log/apache2/keystone_access.log	<input checked="" type="checkbox"/>

Process group

Anomaly detection

- Define how availability problems are detected
 - Impacted service requests
 - Process unavailable

The screenshot shows the Dynatrace interface for managing process group settings. The top navigation bar includes links for Technology overview, Process group Apache Web Server apache*, Settings, and Availability monitoring. The main content area has a sidebar titled "Process group settings" for "Apache Web Server apache*". The sidebar includes tabs for General and Log monitoring, with the "Availability monitoring" tab currently selected. The main panel displays the "Process group availability monitoring" configuration. It states that Dynatrace continuously monitors the availability of the process group and will open a new problem if any process becomes unavailable. A dropdown menu under "Open a new problem" is set to "if service requests are impacted". Below this, a note explains that Dynatrace will open a new problem if the shutdown or crash of any process impacts the performance of any service request.

Technology overview

Process group Apache Web Server apache*

Settings

Availability monitoring

Process group availability monitoring

Dynatrace continuously monitors the availability of this process group. If any process in this group becomes unavailable, a problem will be opened. Use the settings below to define the approach that Dynatrace should use for monitoring the availability of this process group.

Open a new problem

if service requests are impacted

Dynatrace will open a new problem if the shutdown or crash of any process in this group impacts the performance of any service request (for example, if a request fails because it can't reach an unavailable process)

General

Log monitoring

Availability monitoring

Applications



Applications

- View all configured applications
- Turn monitoring on/off on application level
- Drill down to application settings
- Details in next section

Hosts (63)	Process groups (129)	Applications (10)
Application	Summary	Monitoring: Off/On
 www.vmware.easytravel.com	634 sessions / h	<input checked="" type="checkbox"/> 
This application is detected by the following grouping rules:		
domain matches 192.168.238.132		
URL starts with http://213.186.79.232:8080/		
		Remove Edit application
 easyTravel Mobile	604 sessions / h	<input checked="" type="checkbox"/> 
 www.easytravel.com	547 sessions / h	<input checked="" type="checkbox"/> 
 www.openstack.easytravel.com	514 sessions / h	<input checked="" type="checkbox"/> 

Process group detection



Process group detection

- Dynatrace automatically detects process group
- Automatically recognizes if processes belong to the same group
- Works fine in most cases
- Not perfect though -> customization is possible

Process group detection

Settings > Monitoring > Process group detection

Process group detection

Dynatrace detects process groups using a default set of detection rules. These rules enable Dynatrace to know which processes should be considered part of the same process group (i.e., cluster). Detection rules also determine the default names for each process group and the associated process instances (each process instance represents a node in a process group cluster). The settings on this page enable you to adjust the default process-group detection logic.

Ignore versions, builds, dates, and GUIDs in process directory names

To determine the unique identity of each detected process, and to generate a unique name for each detected process, Dynatrace evaluates the name of the directory that each process binary is contained within. [More...](#)

Use CATALINA_BASE to identify Tomcat cluster nodes

By default, Tomcat clusters are identified and named based on the CATALINA_HOME directory name. This setting results in the use of the CATALINA_BASE directory name to identify multiple Tomcat nodes within each Tomcat cluster. [More...](#)

Use Docker container name to distinguish multiple containers

By default, Dynatrace uses image names as identifiers for individual process groups, with one process-group instance per host. [More...](#)

Automatically detect Cassandra clusters

Requires Dynatrace OneAgent version 1.129 or later. Enabling this flag will detect separate Cassandra process groups based on the configured Cassandra cluster name.

Process group detection rules

Custom detection rules provide flexibility and enable you to override the default set of detection rules for specific processes. [More about custom process-group detection](#)

Note: Process-group detection rules are only effective when OneAgent is deployed on your host. For this reason, PaaS integrations don't support process-group detection rules.

[Add detection rule](#)

Detection Rules for deep monitored process groups	Move up/down	Off/On	Delete	Edit
Advanced detection Rules for deep monitored process groups	Move up/down	Off/On	Delete	Edit

Custom process group detection rules

With Dynatrace it is possible to detect and monitor custom process groups. To start monitoring of a custom process group you have to specify executable's name and path. Add command line parameters to limit monitored process groups even more.

[Add detection rule](#)

Detection rule	Delete	Edit
----------------	--------	------

Process group detection

- **Ignore versions, builds, dates, and HUIDs in process directory names**
 - Dynatrace evaluates the directory name where the binary of a process is contained
 - This way the unique identity and name of a process are determined
- **Use CATALINA_BASE to identify Tomcat cluster nodes**
 - By default CATALINA_HOME directory name is used
- **Use docker container name to distinguish containers**
 - By default images names are used
- **Manual rules for detection and grouping**
 - Environment variable
 - Java system property
 - Process property

Process group detection rules

Custom detection rules provide flexibility and enable you to override the default set of detection rules for specific processes [More about custom process-group detection](#)

Note: Process-group detection rules are only effective when OneAgent is deployed on your host. For this reason, PaaS integrations don't support process-group detection rules.

[Add detection rule](#)

Detection Rules for deep monitored process groups

Move up/down _____ Off/On _____ Delete _____ Edit

Advanced detection Rules for deep monitored process groups

Move up/down _____ Off/On _____ Delete _____ Edit

Process group detection

Java system property

Java system property

Use a **Java system property** as the identifier for java process groups

Note: This feature only works for system properties defined at startup via the -D command line parameter

If Dynatrace detects the following Java system property at startup of a process it will use its value to identify a process group:

Use a Java system property to identify cluster nodes within a process group (optional; leave empty if you aren't sure):

Restrict this rule to processes that can be detected as

specific process type at startup.

Note: Only use if you are an expert; not all types can be detected at startup.

Save

Cancel

Process property

Use a **process property** as the identifier for process groups

If Dynatrace detects the property **select attribute** at startup

and it contains (case sensitive)

then extract the identifier for the process group from **select attribute**

in between **From delimiter** and **End delimiter** Ignore versions, hex, dates and build numbers

optionally extract node identifier from **select property**

in between **From delimiter** and **End delimiter** Ignore versions, hex, dates and build numbers

This feature requires OneAgent version 1.123 or higher

Restrict this rule to processes that can be detected as **specific process type** at startup.

Note: Only use if you are an expert; not all types can be detected at startup.

Save

Cancel

Process group detection

Custom detection rules

- **Dynatrace only monitors groups considered important**
- **Valuable process group types Dynatrace reports by default include:**
 - Java application server (e.g. Tomcat, WebSphere, WebLogic, Glassfish, Jboss)
 - All other java applications
 - .NET applications
 - Databases (e.g. MSSQL, Oracle, MYSQL, Cassandra)
 - Web servers (e.g. Apache and IIS)
 - Additional Processes (e.g. Node.js and PHP)
- **Other processes appear if**
 - They have an open TCP listening port or
 - CPU/memory consumption exceeds 5% within 3 samples taken within 5 minutes

Process group detection

Custom detection rules

- Add custom process by providing
 - Executable name
 - Executable path
 - Option: Command line parameter for further filtering

Detection rule _____ Delete Edit

Custom process type

Executable name Executable path

Command line parameters

Save Cancel

Monitored Technologies



Monitored technologies

- **Built-in plugins**
 - Dynatrace plugin
 - JMX monitoring
 - PMI monitoring
 - Infrastructure insights
 - Service insights

Monitored technologies

Specify which technologies Dynatrace should automatically monitor on your hosts. If you only want to enable technology-specific monitoring for individual hosts, disable global monitoring settings on this page and enable them for individual hosts using [host settings](#).

Supported technologies		Custom plugins - beta	Monitoring: Off/On... Edit
Technology	Type		
 Plugin manager	Dynatrace plugin	<input checked="" type="checkbox"/>	
 Configure Couchbase	BETA Dynatrace plugin	<input checked="" type="checkbox"/>	
 Configure CouchDB	BETA Dynatrace plugin	<input checked="" type="checkbox"/>	
 Docker containers <small>Requires Dynatrace OneAgent version 1.87 or later</small>	Dynatrace plugin	<input checked="" type="checkbox"/>	
 Configure Elasticsearch	Dynatrace plugin	<input checked="" type="checkbox"/>	
 Configure HAProxy <small>Requires Dynatrace OneAgent version 1.19 or later</small>	Dynatrace plugin	<input checked="" type="checkbox"/>	
 Memcached	Dynatrace plugin	<input checked="" type="checkbox"/>	
 Configure MongoDB <small>Requires Dynatrace OneAgent version 1.88 or later</small>	Dynatrace plugin	<input checked="" type="checkbox"/>	
 MS SQL <small>Requires Dynatrace OneAgent version 1.85 or later</small>	Dynatrace plugin	<input checked="" type="checkbox"/>	

Monitored technologies

Built-in plugins

- **Dynatrace plugin**
 - Turn on/off monitoring of detected technologies
 - Input additional environment configurations (database credentials, etc.)
- **JMX monitoring**
- **Infrastructure insights**
 - Log analytics and Network traffic monitoring
- **Service insights**
 - Code level visibility on Java, .NET, etc.

Monitored technologies

Custom plugins

- **Python plugins can be written for**
 - Hosts (e.g. additional system metrics)
 - Process groups (e.g. apps, DBs, load balancers)
 - SDK documentation [available on github.](#)
- **Custom metrics are displayed alongside OneAgent metrics**
- **Custom alerts are correlated and included in root cause analysis**

Workshop

Configuration

Process Groups

- Availability
- Process Group Detection

Service Detection Settings



Application Performance
Management

Agenda

- Custom services
- Merged services
- Service naming rules
- Request attributes

Custom Services



Custom services

- Server-side services are automatically detected
- Custom service configuration for non standard frameworks
 - Instruct Dynatrace which method, class or interface should be used
 - Currently Java, .Net and PHP

Custom service detection

Dynatrace automatically detects and monitors most server-side services in your environment with no configuration required. If your application doesn't rely on standard frameworks, you can set up custom services.

With a custom service you can instruct Dynatrace which method, class, or interface it should use to gain access to each of your application's custom server-side services.

Enable this setting to have changes to your custom services applied to Java and PHP processes in real-time with no required restart. [More...](#)

Enable real-time updates to Java and PHP services

Java services

.NET services

PHP services

Define Java service

Active

Service name

Entry points

Move up/down

Delete

Edit

No services

Custom services

- Provide a name for your custom service
- Click *Find entry point*

Define custom Java service

Service must have at least one entry point

Name your custom service

Credit Card

[Find entry point](#)

[Define entry point manually](#)

Custom services

Defining a custom service

- Provide a name for your custom service
- Click *Find entry point*

Define custom Java service

Service must have at least one entry point

Name your custom service

[Find entry point](#) [Define entry point manually](#)

- Select process group containing the entry point

Select the process group that contains your entry point.

[Cancel](#) [Continue](#)

eT-demo-2-BusinessBackend

Custom services

Defining a custom service

- Search for the class
- Optionally use super class or interface

Select process group Find class or interface Select scope Select methods

Select process group Find class or interface Select scope Select methods

Search for loaded classes with names that match a search string.

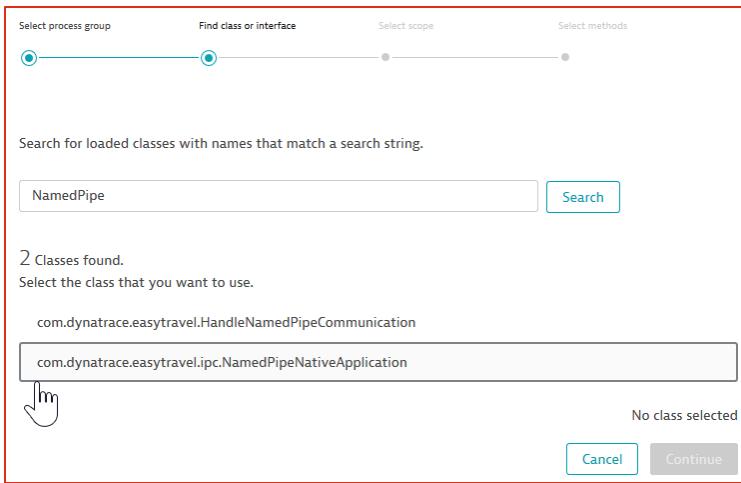
NamedPipe

2 Classes found.
Select the class that you want to use.

com.dynatrace.easytravel.HandleNamedPipeCommunication
 com.dynatrace.easytravel.ipc.NamedPipeNativeApplication



No class selected

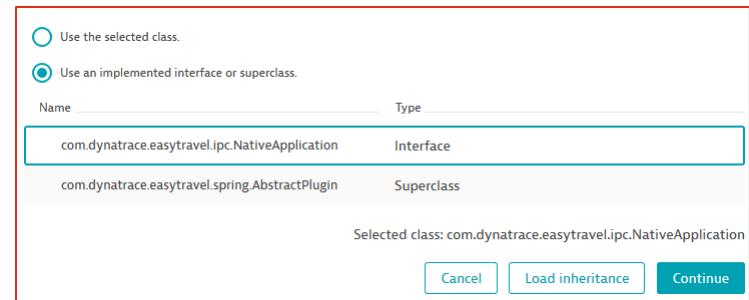


- Optionally use super class or interface

Use the selected class.
 Use an implemented interface or superclass.

Name	Type
com.dynatrace.easytravel.ipc.NativeApplication	Interface
com.dynatrace.easytravel.spring.AbstractPlugin	Superclass

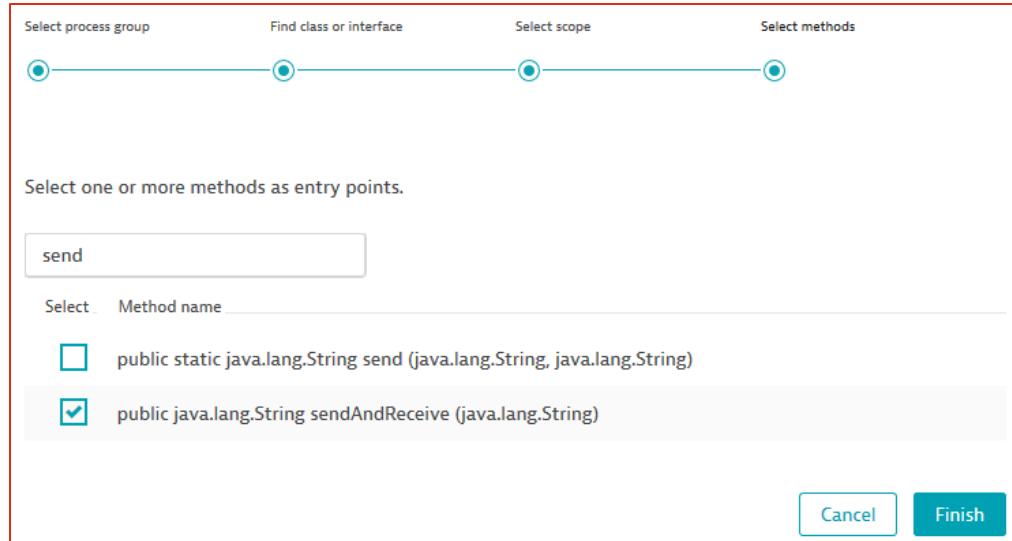
Selected class: com.dynatrace.easytravel.ipc.NativeApplication



Custom services

Defining a custom service

- Select methods that should act as entry point
- Click **Finish**
- Java only
 - With *real-time updates to Java services* changes take effect without process restart

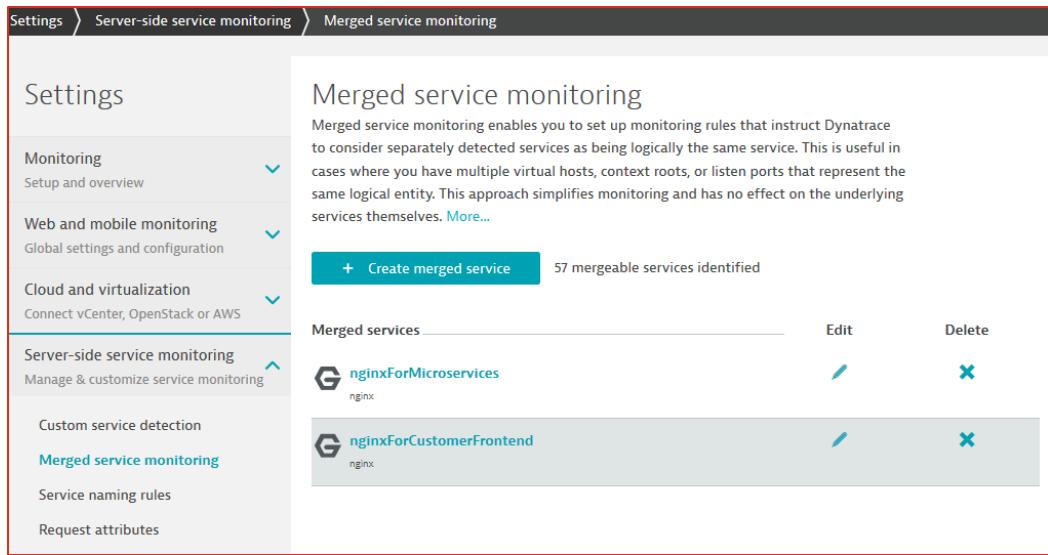


Merged Services



Merged services

- Merge multiple services into one
- For example if there are multiple virtual hosts, content roots or listen ports that represent the same logical entity
- Simplifies monitoring without impact on the services



The screenshot shows the Dynatrace interface for managing merged services. The left sidebar has a 'Settings' menu with several options: Monitoring, Web and mobile monitoring, Cloud and virtualization, Server-side service monitoring (which is selected and highlighted in blue), Custom service detection, Merged service monitoring (which is also highlighted in blue), Service naming rules, and Request attributes. The main content area is titled 'Merged service monitoring' and contains a brief description: 'Merged service monitoring enables you to set up monitoring rules that instruct Dynatrace to consider separately detected services as being logically the same service. This is useful in cases where you have multiple virtual hosts, context roots, or listen ports that represent the same logical entity. This approach simplifies monitoring and has no effect on the underlying services themselves.' Below this is a button labeled '+ Create merged service' and a message '57 mergeable services identified'. A table lists two merged services: 'nginxForMicroservices' (with icon) and 'nginxForCustomerFrontend' (with icon). Each row has 'Edit' and 'Delete' buttons.

Merged services

Example

- **For example an Apache web server with multiple virtual hosts for**
 - amasol.com
 - amasol.de
 - amasol.eu
- **From Apache / Dynatrace perspective these are independent virtual hosts / services**
- **For easier monitoring however, you might want to view the services as one logical service called *amasol web page***
- **Select services to merge and give them a name**

Create merged service

Add at least two services from the Mergeable services list.

Name of merged service

 Create Cancel

Merged services	Remove
 Request on port 8021 (No requests for 2 d 44 min)	
 Request on port 8021 Detecting service...	

Mergeable services	Add
 Request on port 8021 Detecting service...	

Service naming rules



Service naming rules

- Server-side services are detected based on basic properties of application deployment and configuration
- Properties and resulting service names should be intuitive because they reflect the service landscape
- In some cases default naming might not be to your liking or reflect the landscape though
- Process or service meta data can be added to improve usability
- Naming rules enable to customize and enhance default service naming

Service naming rules

- Every property from the *Properties & tags* section of the service page can be used
- Properties from the underlying process can be used
- Detected service and process names can be used

The screenshot shows the configuration interface for a service naming rule. It includes fields for 'Rule name' (empty), 'Service name format' ('New Name {Service:DatabaseVendor}'), and dropdowns for 'Rule applies to entities matching the following properties': 'All process groups', 'All service types', 'All technologies', and 'All service topologies'. Below these are 'Conditions' for 'Database host name' set to 'begins with' (unchecked) and 'Case sensitive' (checked). A text input field 'Type value to be evaluated by rule' is also present. At the bottom are 'Add condition', 'Save', 'Preview', and 'Cancel' buttons. To the right, a box lists 'Available placeholders' such as {AwsAutoScalingGroup:Name}, {AwsAvailabilityZone:Name}, etc., with a 'More...' link.

Rule name

Service name format

New Name {Service:DatabaseVendor}

Rule applies to entities matching the following properties

All process groups

All service types

All technologies

All service topologies

Conditions

Database host name begins with Case sensitive

Type value to be evaluated by rule

Add condition

Save Preview Cancel

Available placeholders

- {AwsAutoScalingGroup:Name}
- {AwsAvailabilityZone:Name}
- {AwsElasticLoadBalancer:Name}
- {AwsRelationalDatabaseService:Endpoint}
- {AwsRelationalDatabaseService:InstanceClass}
- {AwsRelationalDatabaseService:Name}
- {AzureRegion:Name}
- {AzureScaleSet:Name}
- {AzureVm:Name}
- {CustomDevice:DetectedName}
- {CustomDevice:DnsName}
- {CustomDevice:IpAddress}
- {CustomDevice:Port}
- {DockerContainerGroupInstance:ContainerName}
- {DockerContainerGroupInstance:FullImageName}
- {DockerContainerGroupInstance:ImageVersion}

More...

Service naming rules

Example – Add environment

Matching components

Name	New name
 AuthenticationService  dynatrace-dev-BB	dev: AuthenticationService
 BookingService  dynatrace-dev-BB	dev: BookingService
 ConfigurationService  dynatrace-dev-BB	dev: ConfigurationService
 easyTravel Business Backend  dynatrace-dev-BB	dev: easyTravel Business Backend
 easyTravel Customer Frontend  dynatrace-dev-CF	dev: easyTravel Customer Frontend

Service naming rules

Built-in rules

- Built-in rules are shown for documentation but can't be modified

Built-in rules	
Name	Show
PHP web request service (user defined application ID)	▼
PHP web request service	▼
Web request service (with web application ID)	▼
Web request service (without web application ID)	▼
Web request service (on a process that isn't deeply monitored)	▼
Web request service (on an unknown process of a monitored host)	▼
Explicitly monitored AWS S3 service	▼
Explicitly monitored third party web request	▼
Thrift service	▼
Web service	▼
Thrift service (on a process that isn't deeply monitored)	▼
Thrift service (on an unknown process on a monitored host)	▼
Web service (on a process that isn't deeply monitored)	▼
Web service (on an unknown process on a monitored host)	▼
Explicitly monitored 3rd party thrift service	▼

Request Attributes



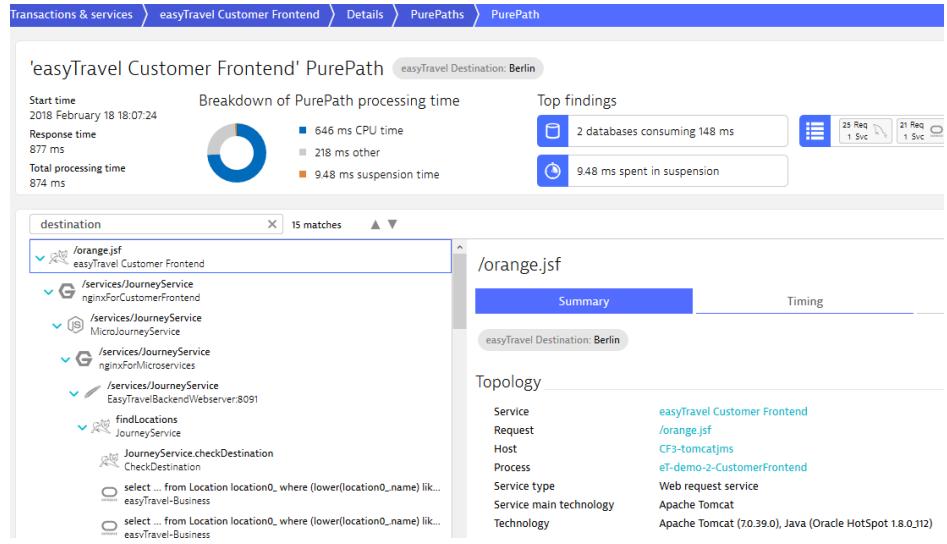
Request attributes

- Dynatrace tracks all requests end-to-end and monitors the services that underlie each transaction
- Performance and attributes of each requests can be analyzed in detail
- Call sequences can be analyzed from multiple angles
- Enables to slices and dice through request to find the *needle in the haystack*

Request attributes

Example

- **Custom request attributes can be defined to improve filtering**
 - Essentially key/value pairs associated with a service request
 - For example a travel website can track the destinations of bookings
 - Setup a destination attribute, will be populated on all service calls where it appears



Request attribute example

- ***Create new request attribute***
- **Provide a unique Request attribute name**
- **Request attributes can have one or more rules**
 - Rules define how attribute values are fetched

The screenshot shows the Dynatrace Settings interface. The navigation path is: Settings > Server-side service monitoring > Request attributes. On the left, there is a sidebar with the following items:

- Monitoring
- Web and mobile monitoring
- Cloud and virtualization
- Server-side service monitoring
- Custom service detection
- Merged service monitoring
- Service naming rules
- Request attributes** (this item is highlighted in blue)

The main content area has a title "Rule based request attributes" followed by a detailed description: "Dynatrace can define request attributes based on captured data from requests to facilitate advanced filtering. Based on custom data sources that you define, request attributes can serve as key/value attributes that are filterable across all Dynatrace service and PurePath views. Request attribute values can be extracted from Web request URLs, HTTP request headers, or other request metadata." Below this description is a blue button labeled "Create new request attribute". At the bottom of the main content area, there is a link to "Request attributes" and another link to "Confluence Space".

Request attribute example

- **Data type**
 - Text, Integer, Double
- **Aggregation**
 - First/Last occurrence
 - Set of distinct values
 - Min/Max/Avg/Sum of captured values
 - Count occurrences/distinct
- **Normalize text**
 - Original / to upper / to lower

Request attributes

Dynatrace can define request attributes based on captured data from requests to facilitate advanced filtering. Based on custom data sources that you define, request attributes can serve as key/value attributes that are filterable across all Dynatrace service and PurePath views. Request attribute values can be extracted from Web request URLs, HTTP request headers, or other request metadata.

Request attribute name

Data type (immutable after setup)

Aggregation on request

Normalize text

Request attribute contains confidential data

Request attributes

Choose data source

- **Data Sources**
 - HTTP Request/Response header
 - Web request URL
 - Web request URL query parameter
 - HTTP POST parameter
 - Java method
 - .NET method

Data sources

Rule applies to requests of the following process group, technology, and process-group tag

All process groups

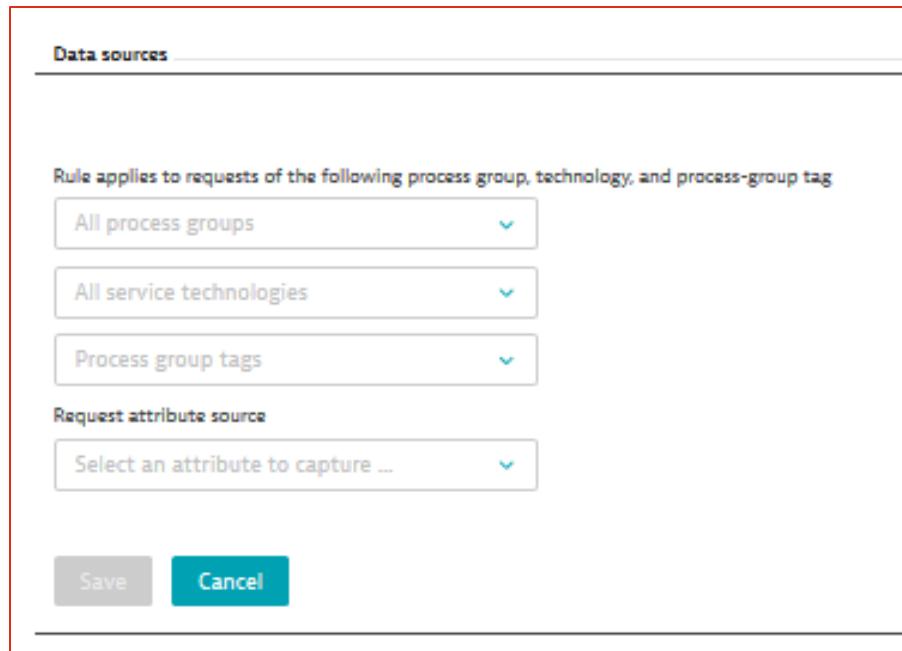
All service technologies

Process group tags

Request attribute source

Select an attribute to capture ...

Save **Cancel**



Request attribute example

Select class / interface

- Select the process group containing the class or interface
- Search for the class that includes the relevant method

Select process group Find class or interface Select scope Select methods

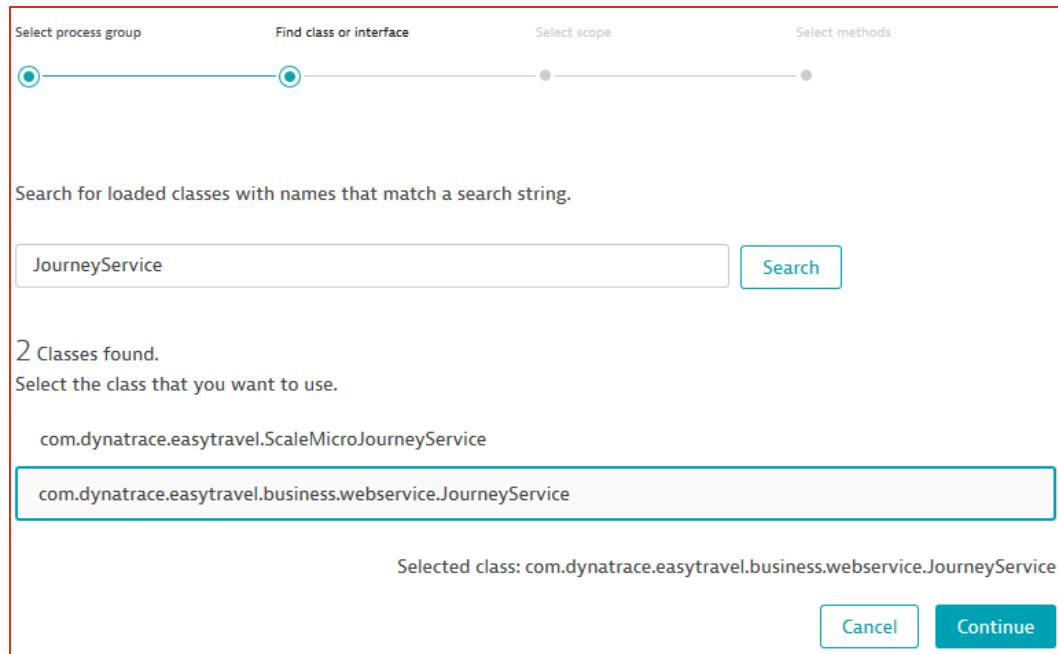
Search for loaded classes with names that match a search string.

JourneyService

2 Classes found.
Select the class that you want to use.

com.dynatrace.easytravel.ScaleMicroJourneyService
com.dynatrace.easytravel.business.webservice.JourneyService

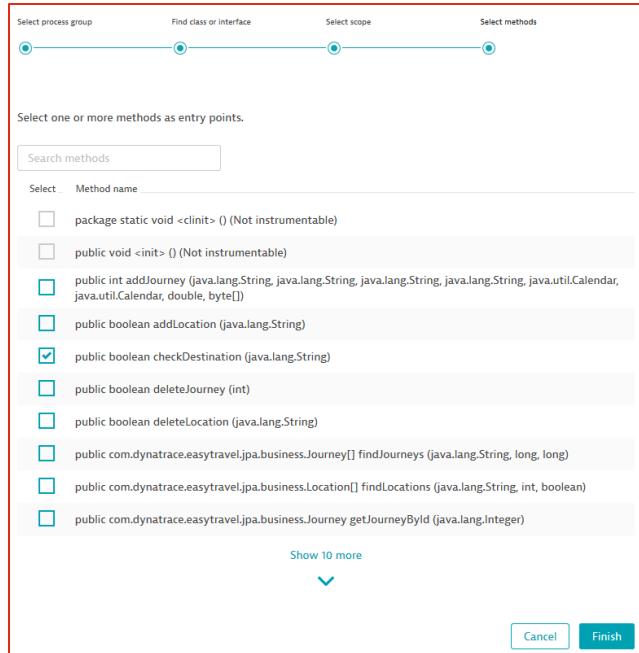
Selected class: com.dynatrace.easytravel.business.webservice.JourneyService



Request attribute example

Select method

- Pick one or more methods that you want to capture parameters from



Request attribute example

Select captured value

- Select which argument or return value you want to capture
- Either restart the process or ensure that real-time updates are enabled (Java only)

Rule applies to requests of the following process group, technology, and process-group tag

All process groups

All service technologies

Process group tags

Request attribute source

Java method parameter(s)

Class

com.dynatrace.easytravel.business.webservice.JourneyService

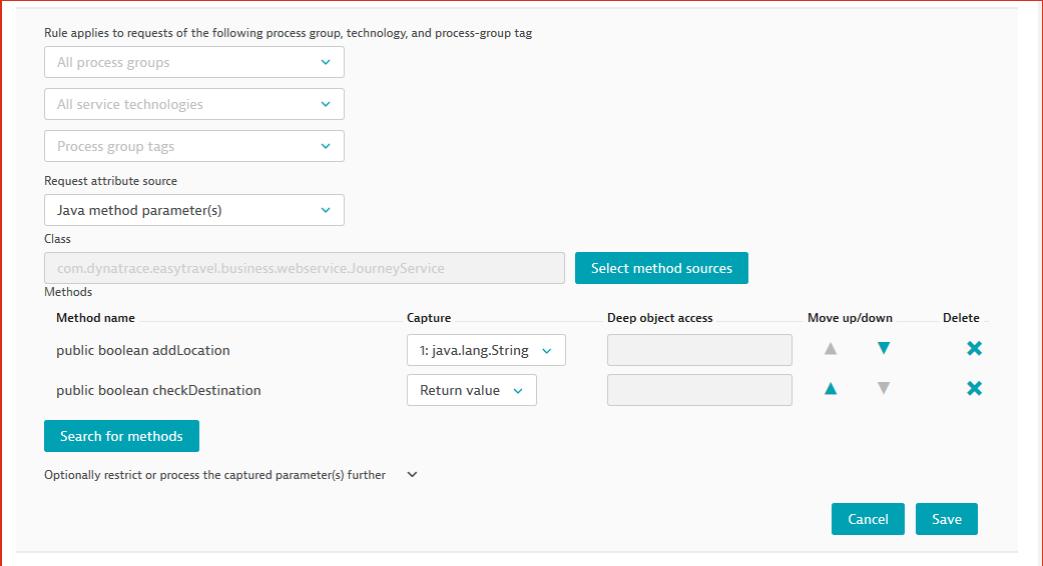
Select method sources

Method name	Capture	Deep object access	Move up/down	Delete
public boolean addLocation	1: java.lang.String		▲ ▼	✖
public boolean checkDestination	Return value		▲ ▼	✖

Search for methods

Optionally restrict or process the captured parameter(s) further

Cancel Save



Request attribute example

Deep object access

- Whenever an captured attribute is a complex object, a method (chain) for deep object access can be defined

Rule applies to requests of the following process group, technology, and process-group tag

All process groups ▾
All service technologies ▾
Process group tags ▾

Request attribute source

Java method parameter(s) ▾

Class

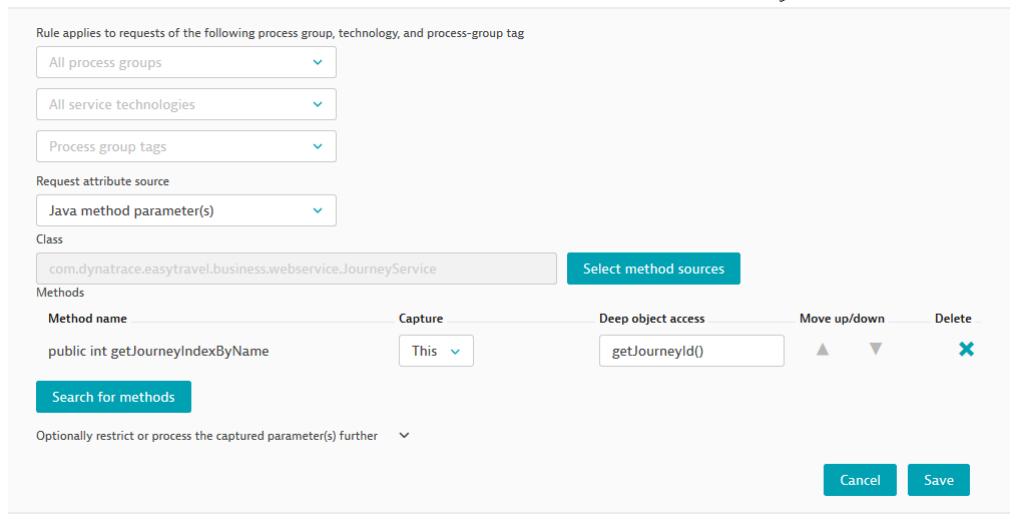
com.dynatrace.easytravel.business.webservice.JourneyService Select method sources

Method name	Capture	Deep object access	Move up/down	Delete
public int getJourneyIndexByName	This ▾	getJourneyId()	▲ ▼	X

Search for methods

Optionally restrict or process the captured parameter(s) further ▾

Cancel Save

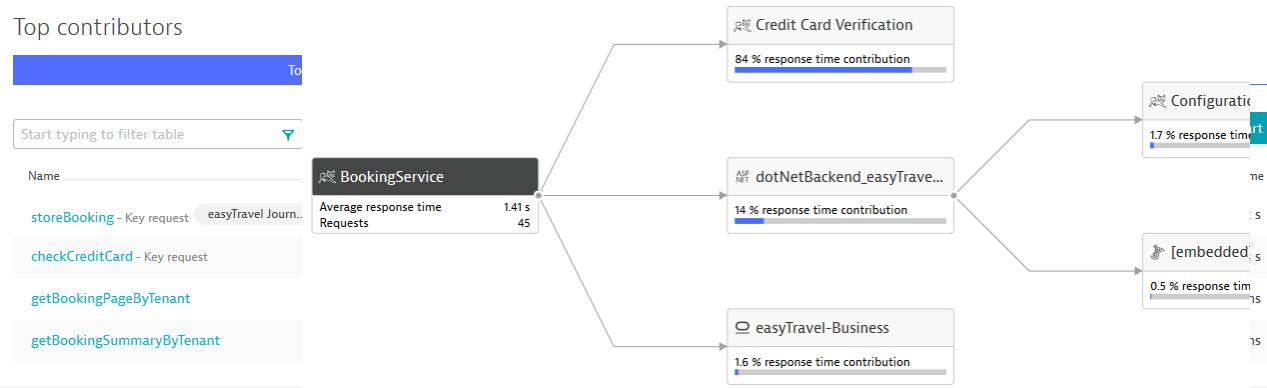


Viewing request attributes



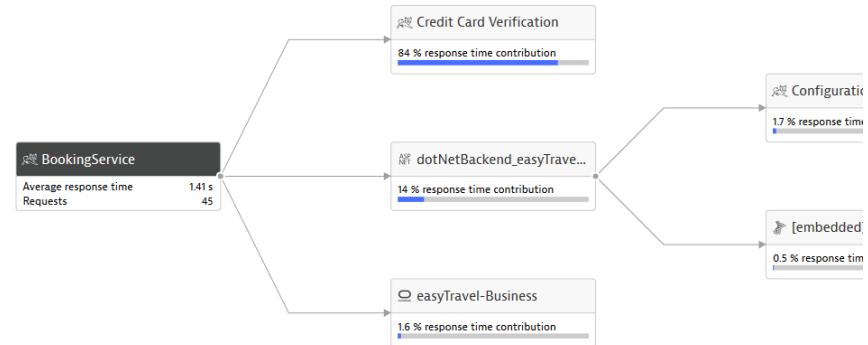
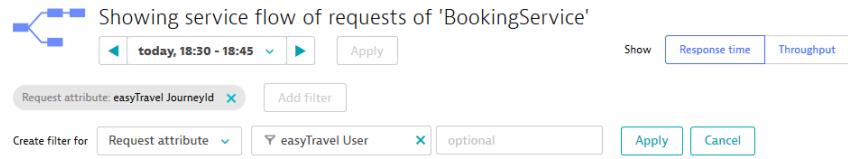
- Once the attributes are selected, look at the Top contributors.
- Look at the Top contributors for the selected attribute.
- The requests now filter to the selected attribute.
- Click any request attribute to view its details.

Top contributors



Viewing request attributes

- Request attributes can be leveraged across all service analysis views
- For example the service flow can be filtered by a request attribute



Workshop

Configuration

Service Settings

- Service Naming Rules
- Merged Services
- Request Attributes

Dynatace API



Application Performance
Management

Open APIs

- Dynatrace SaaS/Managed emphasizes powerful APIs
- Large scale deployments need APIs to cope with automated tasks
- DevOps love APIs to build their automated tasks around!
- Offer API openness in order to integrate with any other service or platform
- Allows Dynatrace to position itself as a monitoring Hub within existing software ecosystems
- Simplify integration with legacy systems

Dynatrace API

Api Documentation

The screenshot shows a navigation sidebar on the left and a main content area on the right. The sidebar lists various API categories: Get started, Data security, Technology support, Monitoring insights, Problem resolution, Diagnostics, User experience, Server-side services, Infrastructure, Monitoring plugins, Dynatrace Managed, Dynatrace API (which is highlighted in red), and External integrations. The main content area has several sections: Introduction, Authentication, Timeseries, Problems, Events, Topology & Smartscape, Real user monitoring & JavaScript, Custom network devices & metrics, and Configuration. Each section contains a list of questions or links related to that topic.

Dynatrace API

Introduction

[What do I need to know to get started?](#)

Authentication

[How do I set up authentication to use the Dynatrace API?](#)

Timeseries

[How do I fetch the metrics of monitored entities?](#)

Problems

[What does the Dynatrace Problems API provide?](#)

[How do I fetch the number of open problems?](#)

[How do I fetch the complete list of problems?](#)

[How do I fetch the full details of a problem?](#)

[How do I push or add comments to problems?](#)

Events

[What does the Events API provide?](#)

[How do API consumers read the events feed?](#)

[How do I push custom events from 3rd party systems?](#)

Topology & Smartscape

[What does the Topology and Smartscape API provide?](#)

[How do I fetch the list of monitored applications?](#)

[How do I fetch the list of monitored services?](#)

[How do I fetch the list of monitored hosts?](#)

[How do I fetch the list of monitored process groups?](#)

[How do I assign a tag to a monitored entity?](#)

Real user monitoring & JavaScript

[How do I fetch the latest JavaScript code for injection?](#)

[Can I customize real user monitoring using the JavaScript API?](#)

Custom network devices & metrics

[What does the Custom network devices and metrics API provide?](#)

Configuration

[What benefits does the Configuration API provide?](#)

[How do I create a maintenance window?](#)

[How do I delete a maintenance window?](#)

[How do I view all configured maintenance windows?](#)

[How do I create or update a plugin or custom event threshold?](#)

[How do I read all configured plugin and custom event thresholds?](#)

[How do I delete a configured plugin or custom event threshold?](#)

[API Documentation](#)

URL

- HTTPS
 - Secured communication channel over HTTPS is mandatory
- Dynatrace SaaS
 - *https://{id}.live.dynatrace.com/api/v1*
- Dynatrace Managed URL Prefix
 - *https://owndomainname/e/{id}/api/v1*

- Authentication is achieved via a user-generated access key (available in your Dynatrace environment settings)
- Log into your Dynatrace environment and go to Settings > Integration > Dynatrace API
- Generate a new access token by typing a unique string into the Key label field, then click the Generate key button.

Dynatrace API Token

The screenshot shows the Dynatrace API Token generation interface. The left sidebar has a 'Settings' section with a 'Integration' category expanded, showing 'Problem notifications' and 'Dynatrace API' selected. The main content area is titled 'Dynatrace API' and contains instructions about using the API to export monitoring data. It features a 'My Dynatrace API tokens' tab (selected) and an 'Other Dynatrace API tokens' tab. A form for generating a token includes a 'Type a token name' input field and several access scope switches. The switches are:

- Access problem and event feed, metrics, topology and RUM JavaScript tag management
- Access logs
- Configure maintenance windows
- User session query language
- Anonymize user session data for [data privacy](#) reasons

Below the switches are 'Generate' and 'Cancel' buttons. The 'API tokens' section shows a table with one row: 'Token name' (empty), 'Owner' (empty), and a row of buttons for 'Disable/enable', 'Delete', and 'Edit'. A note says 'No tokens available.'

API Token Header

- Passing your API token within an authorization header
 - *Authorization: Api-Token <Generated token>*
- Passing your API token within a query parameter
 - <https://{{id}}.live.dynatrace.com/api/v1/timeseries?Api-Token=cw88t44BRk2KcJkdM419T>

Timeseries API

- <https://{{id}}.live.dynatrace.com/api/v1/timeseries>
- The timeseries endpoint delivers metrics that Dynatrace collects from the different monitored entities over time
- This endpoint is used to read metrics, such as CPU usage, for selected entities over a given timeframe
- By passing additional parameters this endpoint allows you to filter the selected timeseries for entity types and to specify what type of result aggregation the result should contain.

Problems API

- <https://{{id}}.live.dynatrace.com/api/v1/problem/>
- This family of endpoints delivers metrics and details about problems that Dynatrace detects within a given environment
- The returned list of problems is identical to that shown in the Dynatrace Web UI
- A single problem typically contains summary information, impact analysis, and a list of any events that are correlated with the problem

Events API

- <https://{{id}}.live.dynatrace.com/api/v1/events/>
- This family of endpoints delivers details about all uncorrelated events that Dynatrace collects within a given environment
- Information returned for each event includes attributes about the event source, the entity where the event was collected, and other event-specific details
- This family of endpoints allows API consumers to receive a global feed of all uncorrelated events in an environment and to push external events to a Dynatrace environment
- The events REST endpoint enables 3rd party integrations to push custom events to one or more monitored entities via the API

Topology and Smartscape API

- <https://{{id}}.live.dynatrace.com/api/v1/entity/>
- This family of endpoints delivers details about the applications, services, and infrastructure entities that Dynatrace automatically detects and monitors within a given environment
- The returned information contains important attributes about the monitored entities as well as outgoing and incoming relationships
- This family of endpoints is organized along the three major environment layers:
 - Applications
 - Services
 - Infrastructure

Some examples

- Fetching problem feed and details
 - https://demo.live.dynatrace.com/api/v1/problem/feed/?relativeTime=week&Api-Token=<YOUR_TOKEN>
 - https://demo.live.dynatrace.com/api/v1/problem/details/5853544736714377710/?relativeTime=week&Api-Token=<YOUR_TOKEN>
- Fetching a metric/list of metrics
 - https://demo.live.dynatrace.com/api/v1/timeseries/?Api-Token=<YOUR_TOKEN>
 - https://demo.live.dynatrace.com/api/v1/timeseries/?filter=plugin&Api-Token=<YOUR_TOKEN>
 - https://demo.live.dynatrace.com/api/v1/timeseries/?relativeTime=hour&aggregationType=AVG×eriesId=com.dynatrace.builtin:host.cpu.user&entity=HOST-6D01CC4CCDA915AA&Api-Token=<YOUR_TOKEN>
 - <https://demo.live.dynatrace.com/api/v1/timeseries/?relativeTime=hour&aggregationType=AVG×eriesId=com.dynatrace.builtin:host.cpu.user&Api-Token=<Api-Token>>
- Predict a metric
 - https://demo.live.dynatrace.com/api/v1/timeseries/?relativeTime=10mins&aggregationType=AVG×eriesId=com.dynatrace.builtin:host.cpu.user&entity=HOST-6D01CC4CCDA915AA&predict=true&Api-Token=<YOUR_TOKEN>

Lifecycle Integration: First Steps

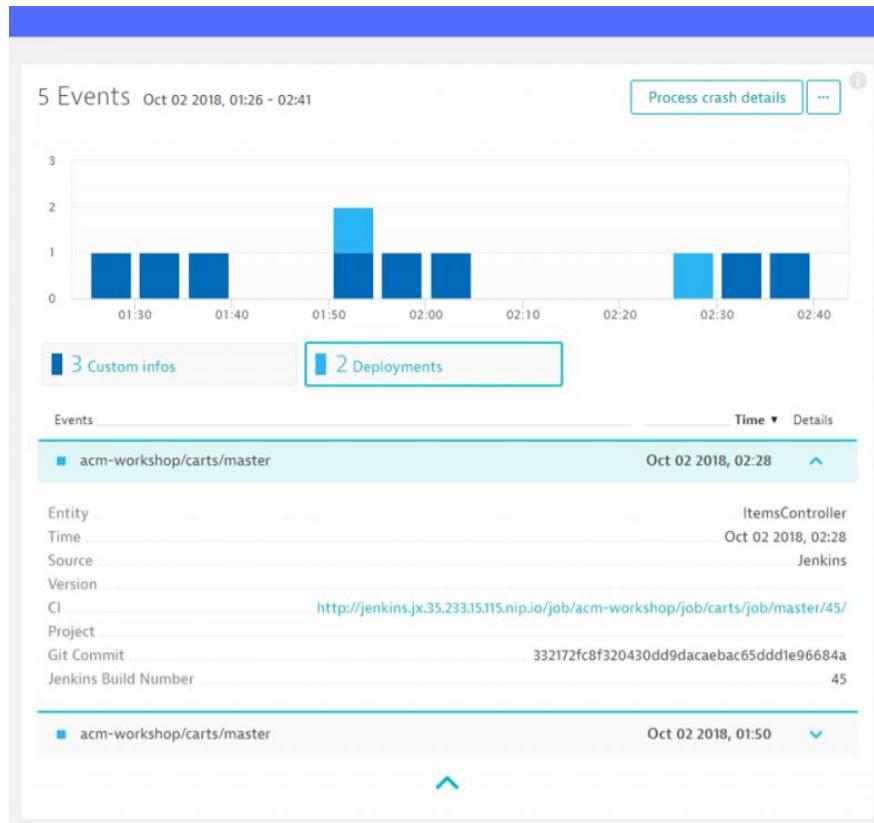


Application Performance
Management

Dynatrace CI Integration

Deployment Event

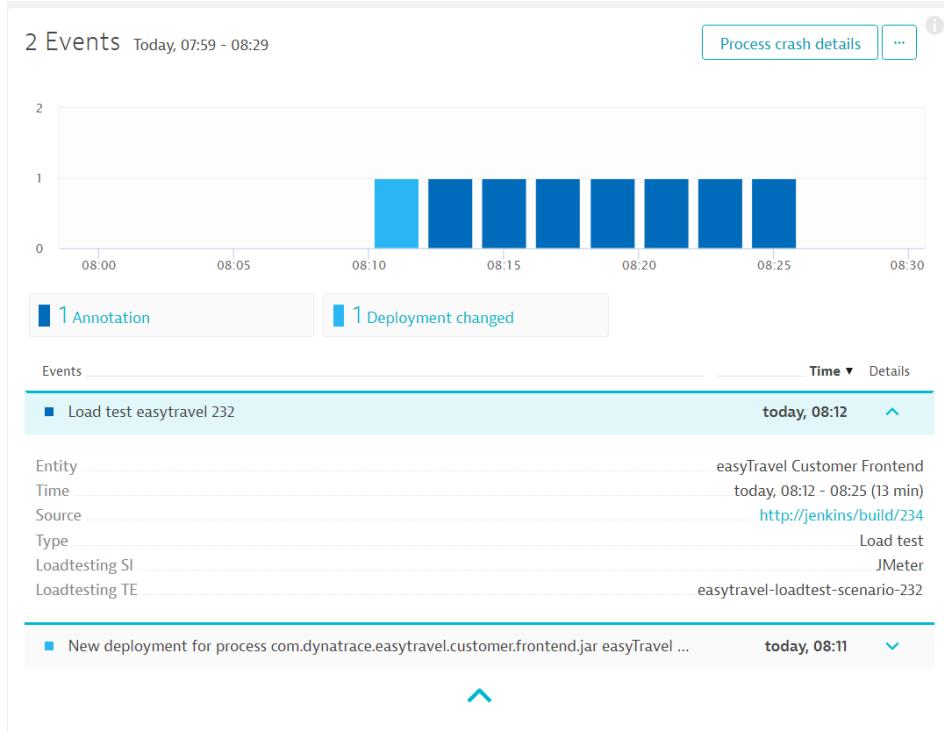
- Deployment Informationen an Dynatrace pushen
 - Bei jedem Pipeline Durchlauf gibt es eine Verbindung von Dyntrace zu Jenkins
 - Informationen zu Deployments sind jederzeit Zusammen mit den Monitoring Daten sichtbar
 - Werden in Root-Cause Analyse berücksichtigt



Dynatrace CI Integration

Tests

- Integration, API, Load or Performance Tests ausführen
 - Tagging zur leichteren Segemtierung Analog zu AppMon
 - Test Event an Dynatrace pushen
 - Weitere Möglichkeiten
 - Zusätzliche Metriken aus Test Tool an Dynatrace pushen
 - Environment Readiness Check via Smartscape API



Dynatrace CI Integration

Performance Signature

- Performance Signature erstellen
 - Evaluierung ausgewählter Metriken die über JSON Perf Spec Files definiert werden

Performance Signature Definition

```
"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  
    }  
]
```

Timeseries Evaluation

98th Percentile Response Time



Build Result

Throughput (Request-Count)



Pitometer (part of @keptnProject): Metrics-based grading of a Deployment!

Metric Source &
Query

Grading Details
& Metric Score

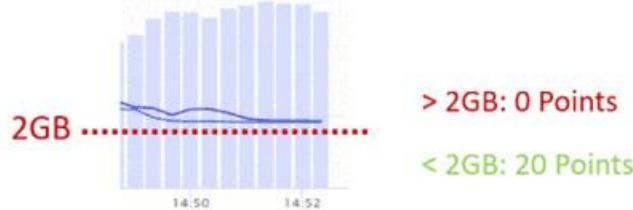
Total Scoring
Objectives

Pitometer Specfile

```
{
  "spec_version": "1.0",
  "indicators": [
    {
      "id": "go_gcstats_alloc_bytes",
      "source": "Prometheus",
      "query": "avg(go_gc_duration_seconds{instance='localhost:9090'})",
      "output": "total",
      "grading": {
        "type": "Threshold",
        "thresholds": {
          "upperSevere": 21882344
        },
        "metricScore": 20
      }
    },
    {
      "id": "ConversionRate",
      "source": "Dynatrace",
      "query": {
        "timeseriesId": "com.dynatrace.builtin:app.conversionrate",
        "aggregation": "avg",
        "tags": [
          "SockShop-Blue"
        ]
      },
      "output": "series",
      "grading": {
        "type": "Threshold",
        "thresholds": {
          "lowerSevere": 2,
          "lowerWarning": 5
        },
        "metricScore": 20
      }
    }
  ],
  "objectives": {
    "pass": 30,
    "warning": 15
  }
}
```

Source

Allocated Bytes (from Prometheus)



Grader

If value: 3GB

Score: **0**

Conversion Rate (Dynatrace)



If value: 3.9%

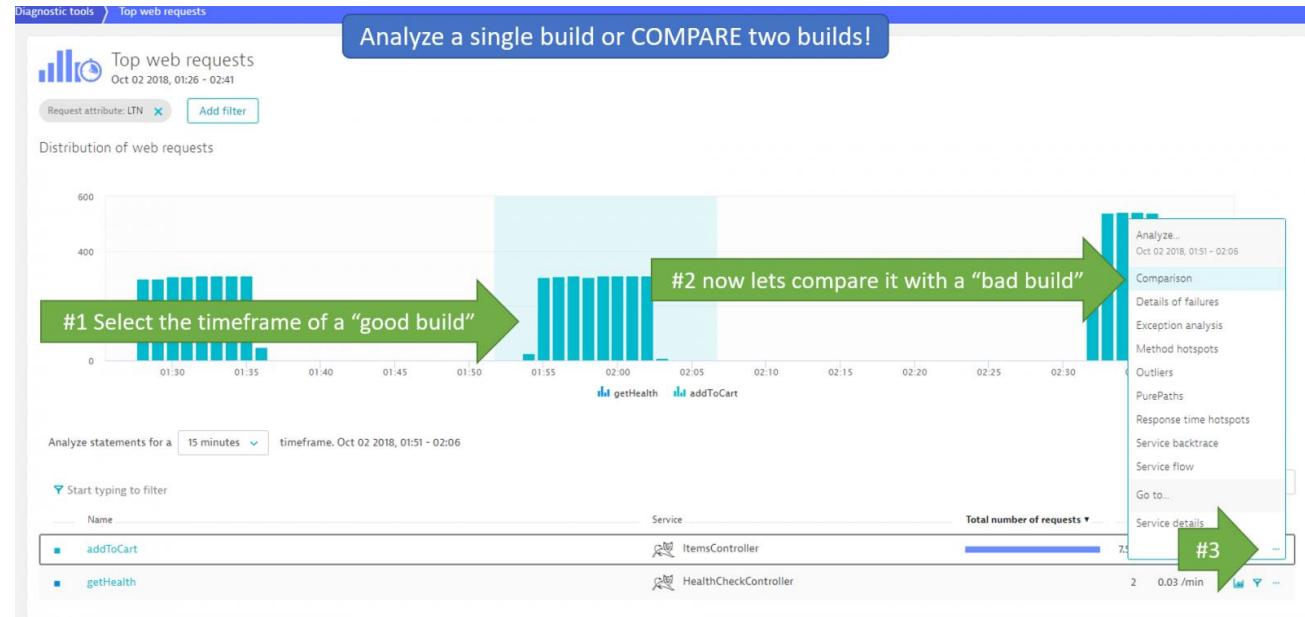
Score: **10**

Total Score: **10**

Dynatrace CI Integration

Analyse fehlgeschlagener Builds

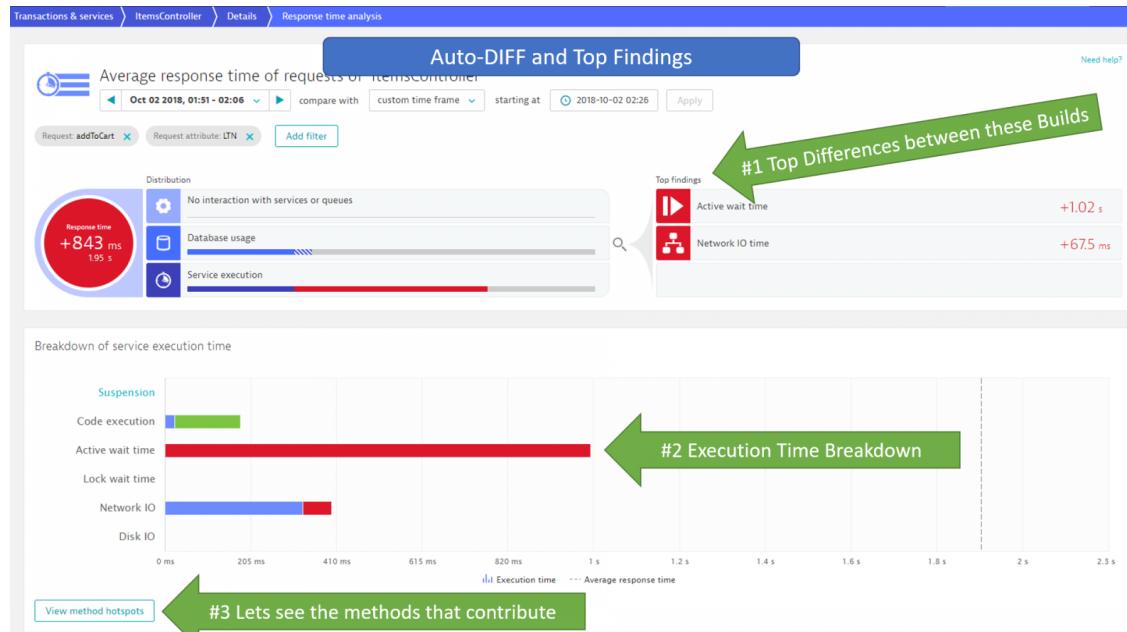
- Vergleich fehlgeschlagener und Erfolgreicher Builds:



Dynatrace CI Integration

Analyse fehlgeschlagener Builds

- Analyse der Unterschiede in Standard Dashboards, gefiltert auf Testdurchläufe



Useful links

- **Documentation**
 - help.dynatrace.com
- **Answers (forum)**
 - answers.dynatrace.com
 - RFE are posted/tracked here
- **Product News**
 - [amasol blog](#)
 - [Dynatrace blog](#)



Vielen Dank!

Christoph Bartsch | Patrick Hofmann