

Traversing Your Stack

Dynatrace Training Module



In depth look at the Dynatrace model

Hosts

Process
Groups

Services

Traversing Your Stack

Hosts

What is a Process Group?

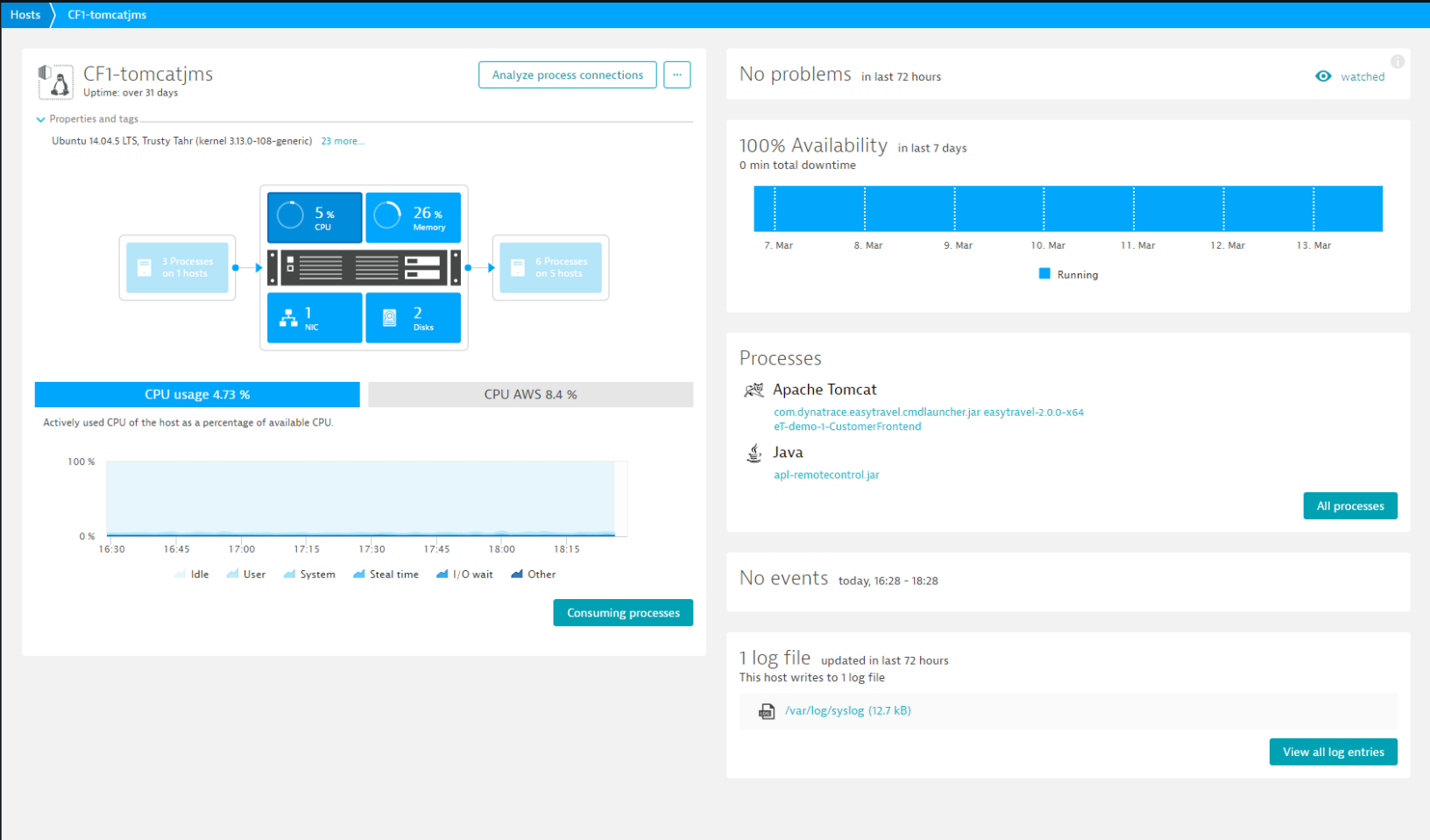
Analyzing Process Groups

What is a Service?

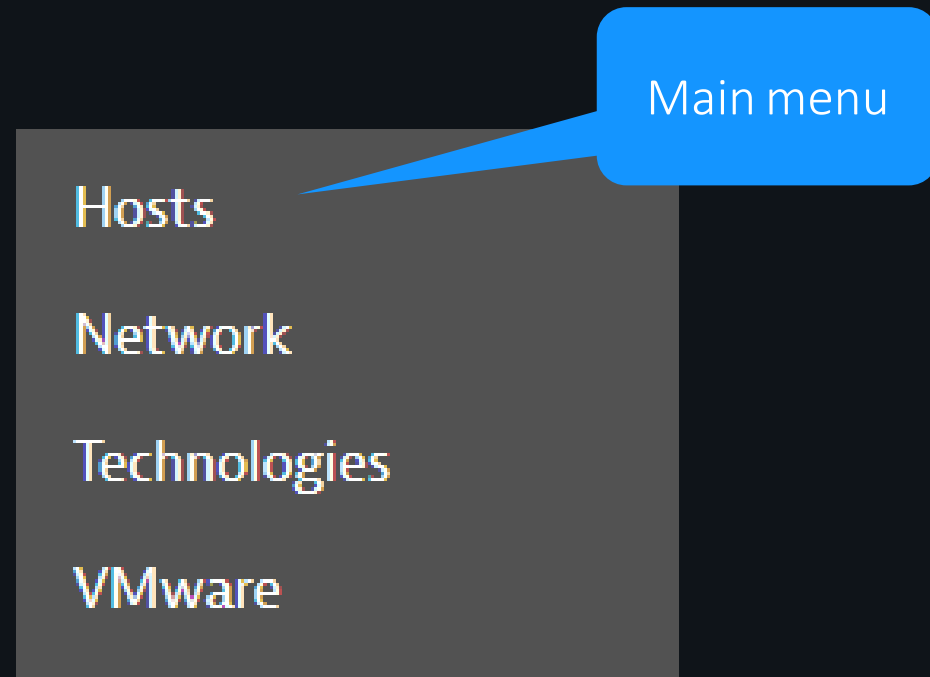
Analyzing Services

Viewing and Filtering

Hosts

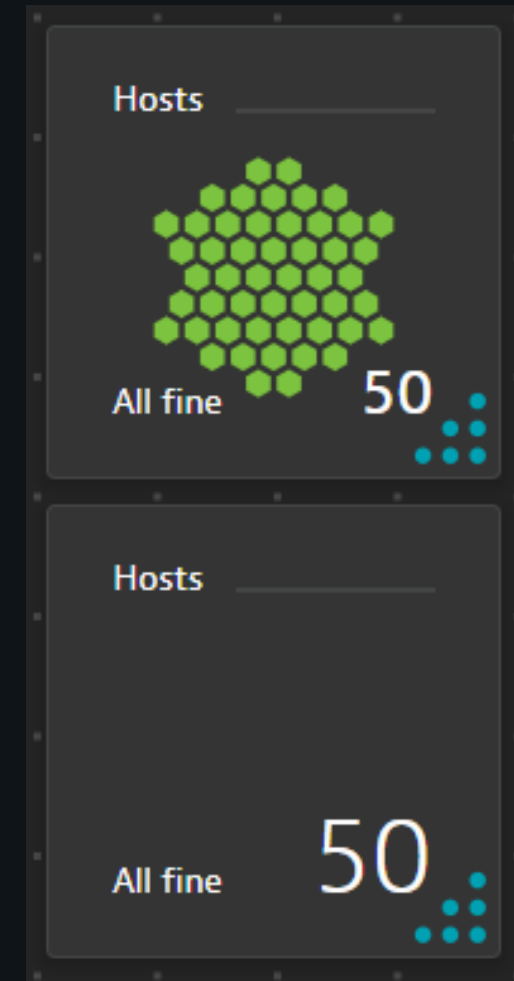


How do I analyze Hosts?



OR

As a dashboard tile



How do I analyze Hosts?

Search your environment...

Last 6 hours

Hosts

Search

All25

▼ State

Running25

▼ Type

VMware6

EC212

OpenStack4

XEN1

Azure Web Sites1

Physical host1

▼ Datacenters

East US 21

Travel-demo6

All hosts

25 hosts

Monitor another host

Name	State ▼	Operating system	Virtualization	CPU usage	Memory usage	Disk latency	Network traffic
WIN4-tomcat	running	Windows	Xen	0.31 %	24 % of 7.5 GB	11.9 ms	22.2 kbit/s
RD0003FF816864	running	Windows	HyperV	9.56 %	70 % of 1.75 GB	-	-
OpenStackControllerNode	running	Linux	-	9.54 %	63 % of 47.2 GB	14.2 ms	12.5 Mbit/s
OpenStack-WebLauncher	running	Linux	KVM	5.63 %	27 % of 3.86 GB	26.2 ms	24.9 Mbit/s
OpenStack-CustomerFrontend-ThirdPartyContent	running	Linux	KVM	61 %	61 % of 1.95 GB	22 ms	24.9 Mbit/s
OpenStack-BusinessBackend	running	Linux	KVM	54 %	45 % of 1.95 GB	50.1 ms	1.2 Mbit/s
OpenStack-Apache	running	Linux	KVM	7.76 %	16 % of 3.86 GB	17.9 ms	38.7 Mbit/s
LBVM-mongo-haproxy-docker	running	Linux	VMware	24 %	73 % of 1.95 GB	39.8 ms	42.7 Mbit/s
LB-apache-php-varnish	running	Linux	Xen	34 %	37 % of 7.28 GB	1.64 ms	27 Mbit/s
ETVM-tomcat	running	Windows	VMware	15 %	71 % of 4 GB	18.2 ms	40.1 Mbit/s
CPU4	running	Linux	VMware	0.38 %	23 % of 1.96 GB	32 ms	17.7 kbit/s



CF1-tomcatjms

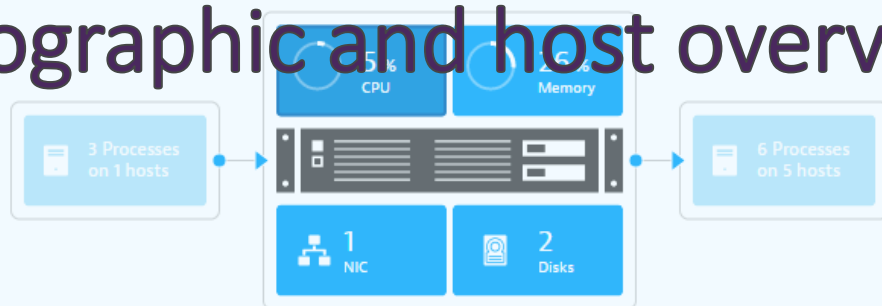
Uptime: over 31 days

[Analyze process connections](#)

▼ Properties and tags

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

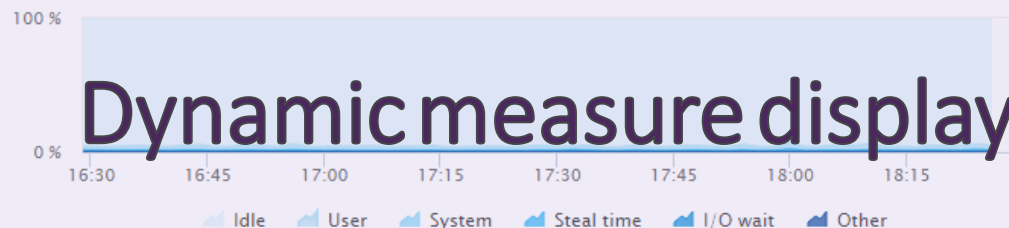
Infographic and host overview



CPU usage 4.73 %

CPU AWS 8.4 %

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



Dynamic measure display

[Consuming processes](#)

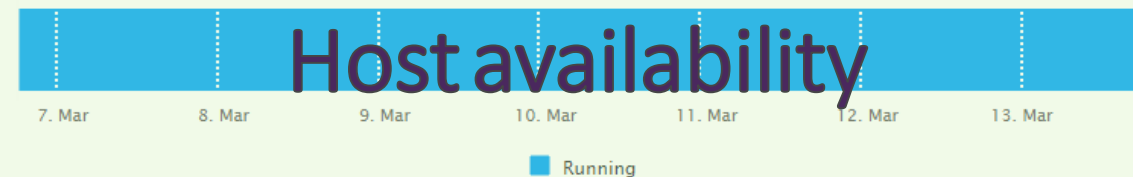
No problems in last 72 hours

Problem history

watched

100% Availability in last 7 days

0 min total downtime



Host availability

Processes



Apache Tomcat

[com.dynatrace.easytravel.cmdlauncher.jar](#) [easytravel-2.0.0-x64](#)
[eT-demo-1-CustomerFrontend](#)

Java

[apl-remotecontrol.jar](#)

Detected Processes

[All processes](#)

No events today, 16:28 - 18:28

Event logger

1 log file updated in last 72 hours

This host writes to 1 log file

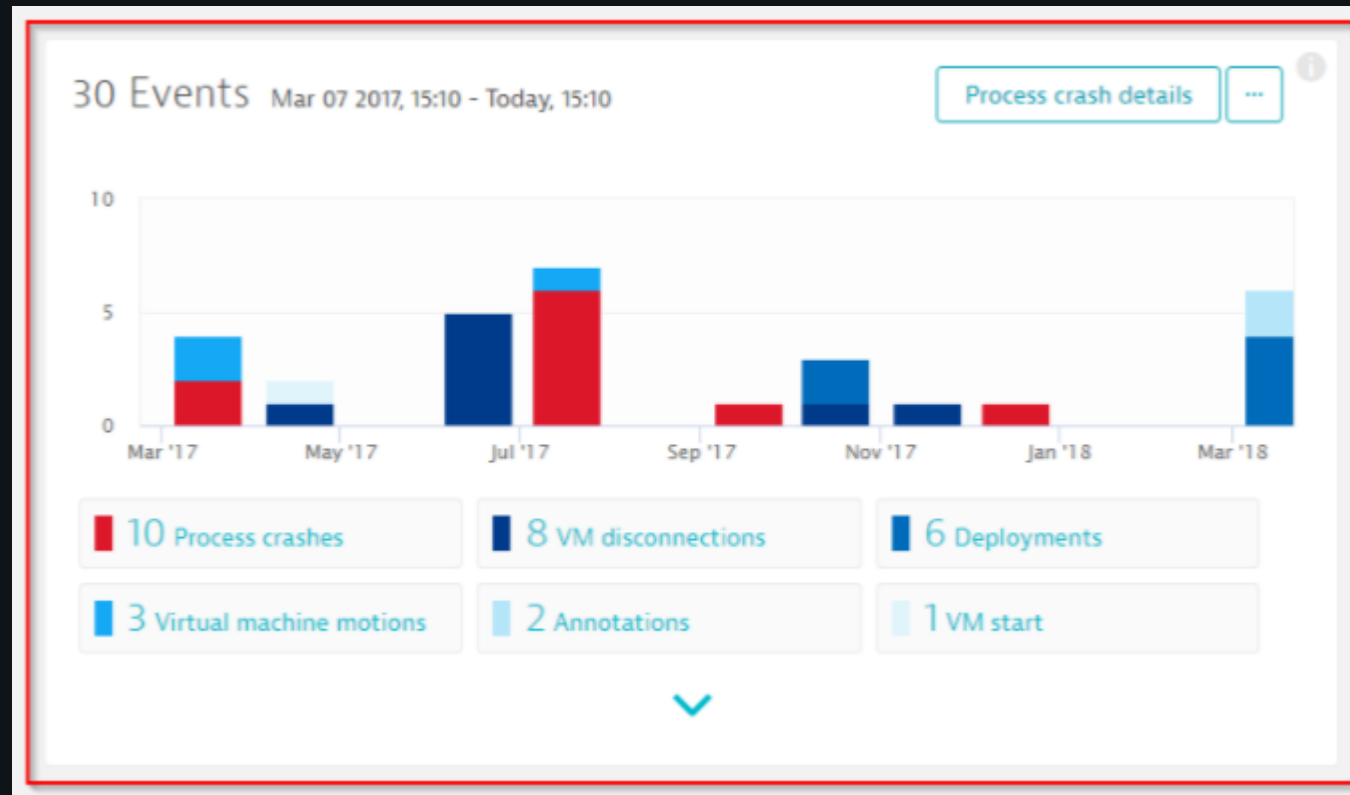
[/var/log/syslog \(12.7 kB\)](#)

Log files for host

[View all log entries](#)

Event Analytics

- Dynatrace detects more than 80 different built-in system event types
 - Including process crashes, deployment configuration changes, and VM motion events



Traversing Your Stack

Hosts

What is a Process Group? ◀

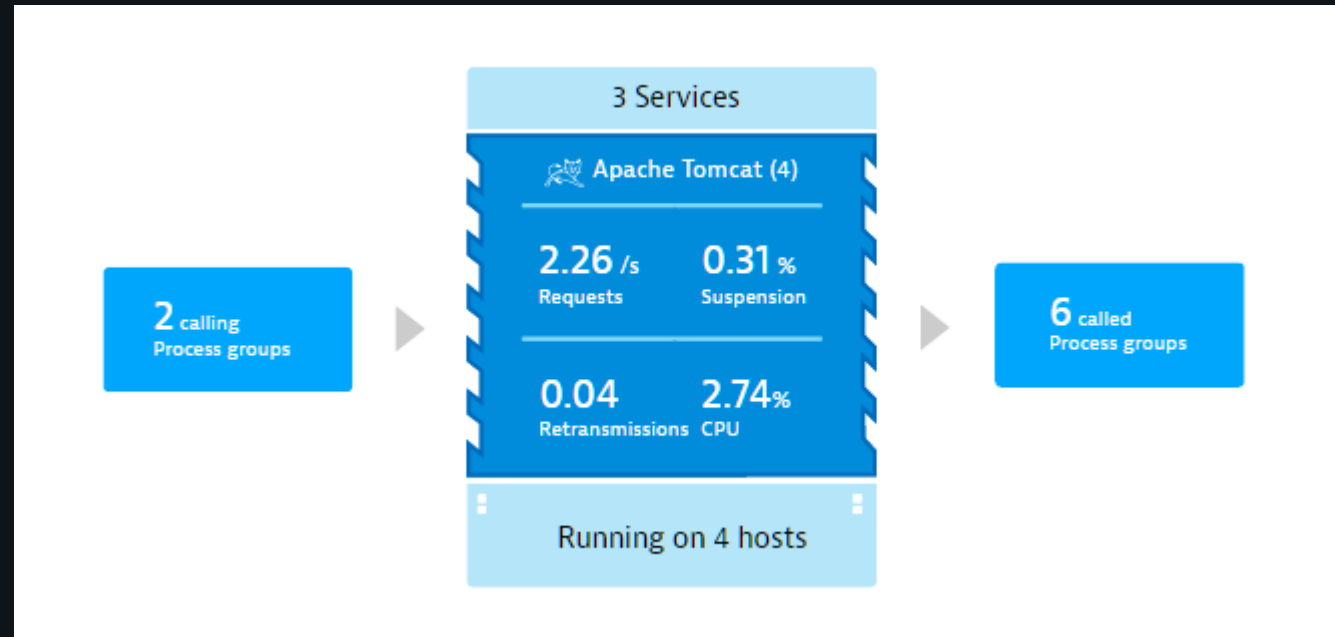
Analyzing Process Groups

What is a Service?

Analyzing Services

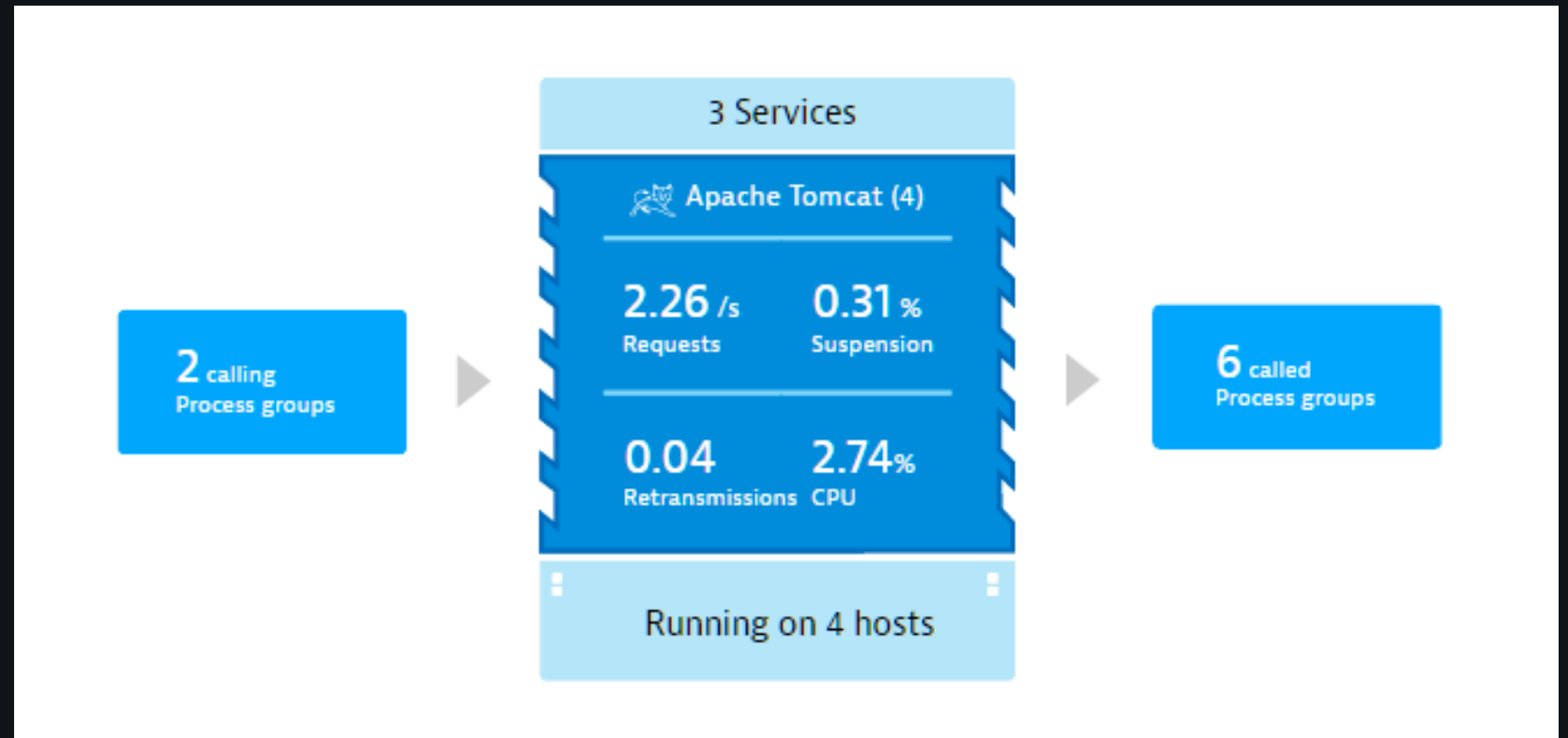
Viewing and Filtering

What is a Process Group?



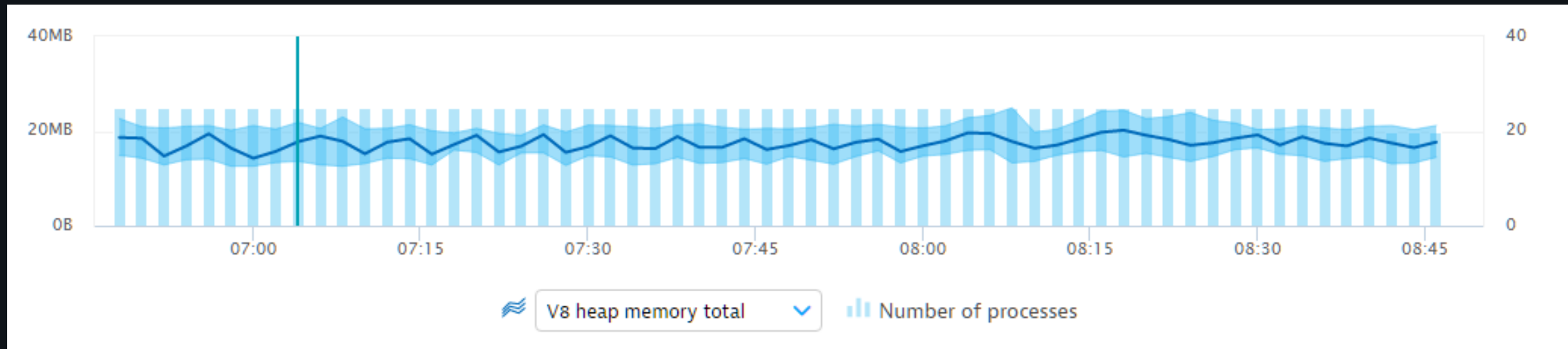
What are process groups

- Cluster of processes that belong together
 - Tomcat cluster, Jboss cluster, WebSphere cluster
- Run the same software
- Contain Services



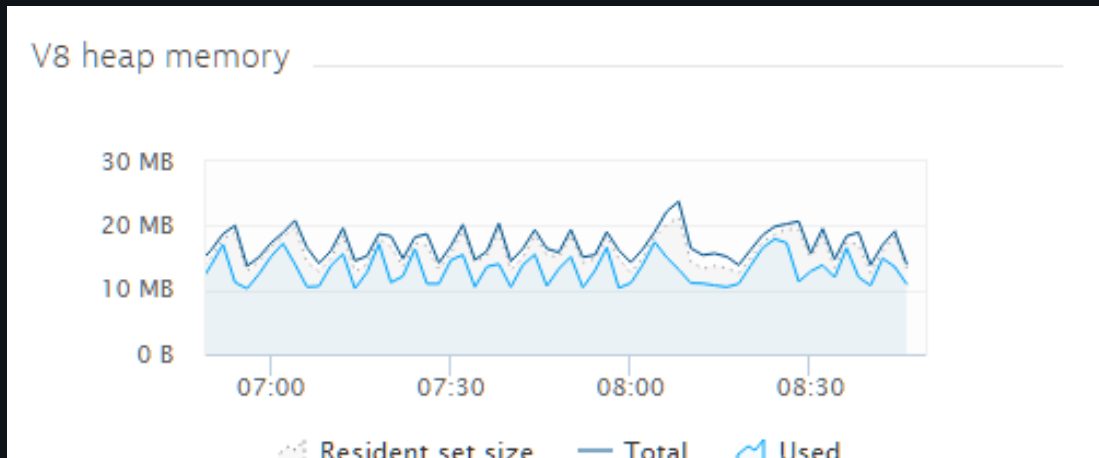
What are process groups

- Should be stable! (Deployment, version upgrade)
- Used as a point of configuration
- Process, Plugin metrics continuity
- Like agent groups, but automatic and for all types of processes



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)



Process	CPU ▾	Memory usage	Responsiveness	Worker processes
eT-demo-2-CustomerFrontend on CF4-tomcatjms	4.04 %	555 MB	0.27 ms	1
eT-demo-2-CustomerFrontend on CF2-tomcatjms	3.4 %	583 MB	0.26 ms	1
eT-demo-2-CustomerFrontend on CF1-tomcatjms	3.09 %	655 MB	0.26 ms	1
eT-demo-2-CustomerFrontend on CF3-tomcatjms	2.95 %	573 MB	0.26 ms	1

Processes vs Process group instance

- Technology specific – an instance can be a cluster node
- Multiple “nodes” per host possible



WebSphere AS Ir-aix-c01Node01Cell (Ir-aix-c01Node01 / server1)

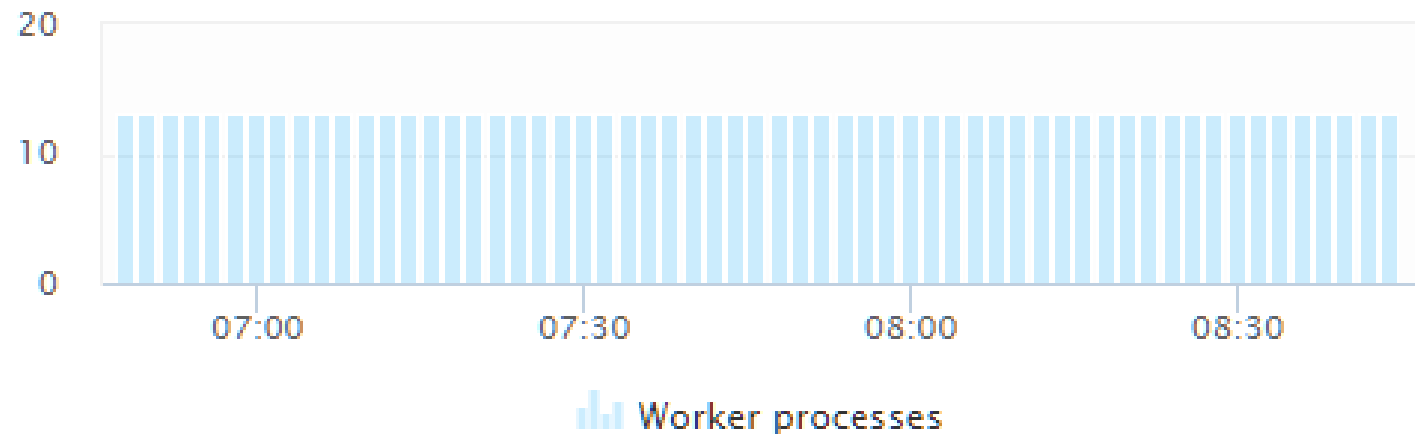
^ Properties

Type	IBM WebSphere (8.5.5.0)
Technologies	Java (IBM 1.6.0 SR5 FP1)
Process group	WebSphere AS Ir-aix-c01Node01Cell
Bitness	64-bit
Main class	com.ibm.wsspi.bootstrap.WSPreLauncher
Node name	Ir-aix-c01Node01 / server1
Websphere cell	Ir-aix-c01Node01Cell
Websphere node	Ir-aix-c01Node01
Websphere server	server1

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...


Worker processes



Process group detection

- Lots of built-in rules: Java, Jboss, WebSphere, Node.JS, Apache, Databases....
- Many special rules: Docker, Azure, CloudFoundry, OpenShift, ColdFusion....
- Intention: Stable, meaningful and meta data
- Examples:
 - Java Jar → use Jar-Name
 - Tomcat → use last part of CATALINA_HOME
 - WebLogic → use domain
 - CloudFoundry → apply "CF Application Name"

Process group detection (metadata)

 Apache Web Server apache2 running PHP

^ Properties

Type

Technologies

Process group

Ports

Bitness

Application version

Modules

Docker container

Docker image

Apache configuration path

Container id

IIS .NET

IIS app pool ruxitTaDotNet

View process group

...

^ Properties

Type

Technologies

Process group

Bitness

Modules

IIS App Pool

.NET (.NET Framework 4.6.1590.0), ASP.NET (v4.6.1590.0), CLR (FullCLR 4.6.1590.0), msmq (v4.6.1590.0), mssql compact client (v4.0.8876.1), and WCF (v4.6.1590.0)

spring-music-dev

View process group

...

^ Properties

Type

Technologies

Process group

Bitness

Main class

Catalina base

Catalina home

Cloud Foundry application

Cloud Foundry instance index

Cloud Foundry space id

Cloud Foundry space

Apache Tomcat (8.0.39.0)

Java (OpenJDK 1.8.0_111)

spring-music-dev

64-bit

org.apache.catalina.startup.Bootstrap

/home/vcap/app/java-buildpack/tomcat

/home/vcap/app/java-buildpack/tomcat

spring-music-dev

0

b4c8be3d-77fb-4acf-ab18-be148b66c33f

development

Azure web app host name

Azure web app site name

IIS app pool

Customize and why?

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

Traversing Your Stack

Hosts

What is a Process Group?

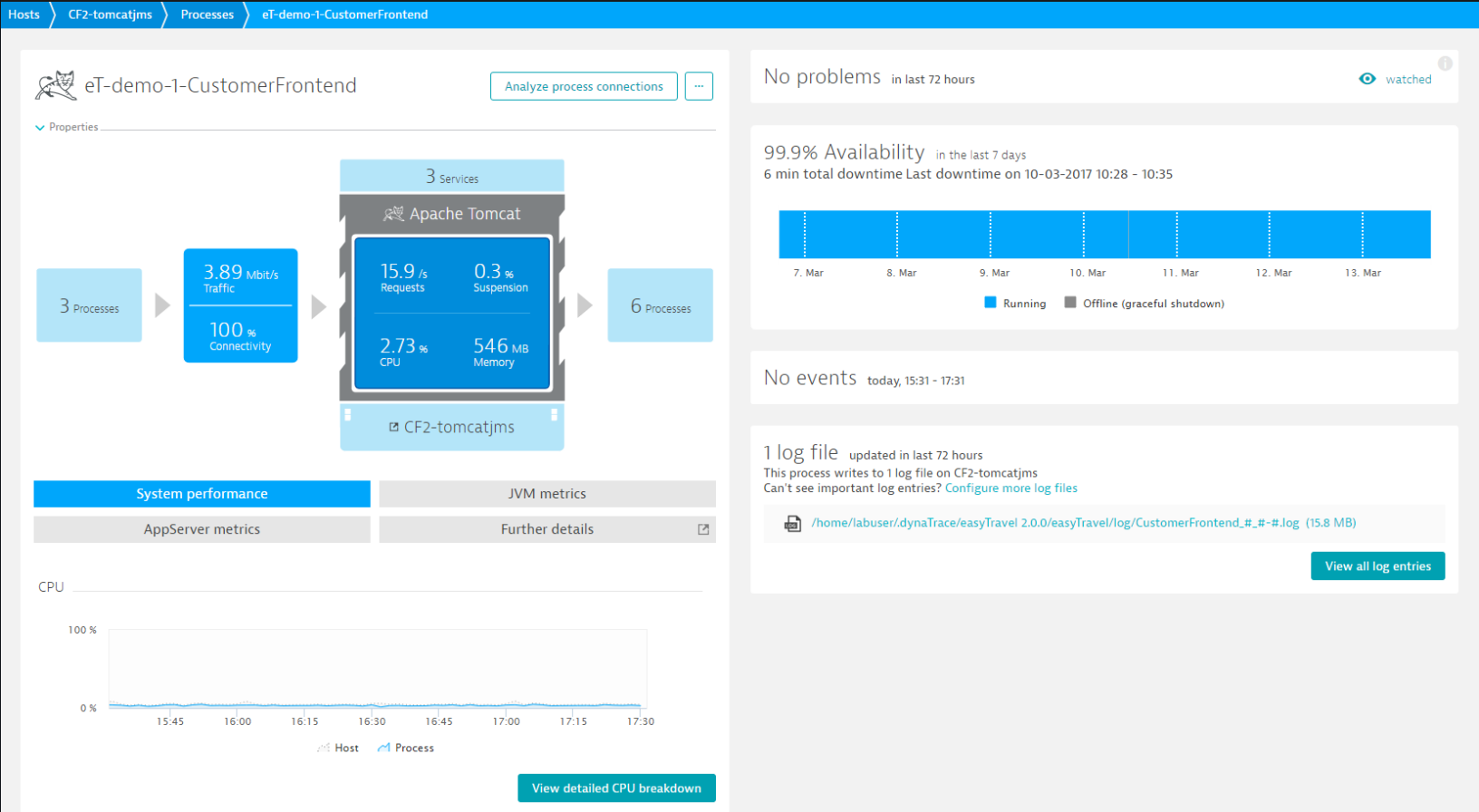
Analyzing Process Groups

What is a Service?

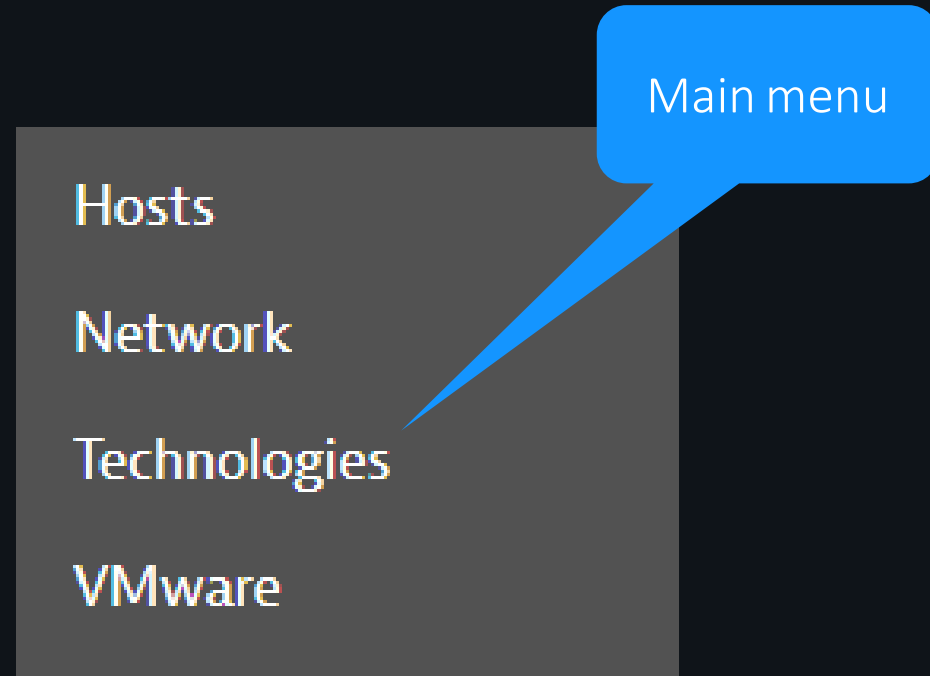
Analyzing Services

Viewing and Filtering

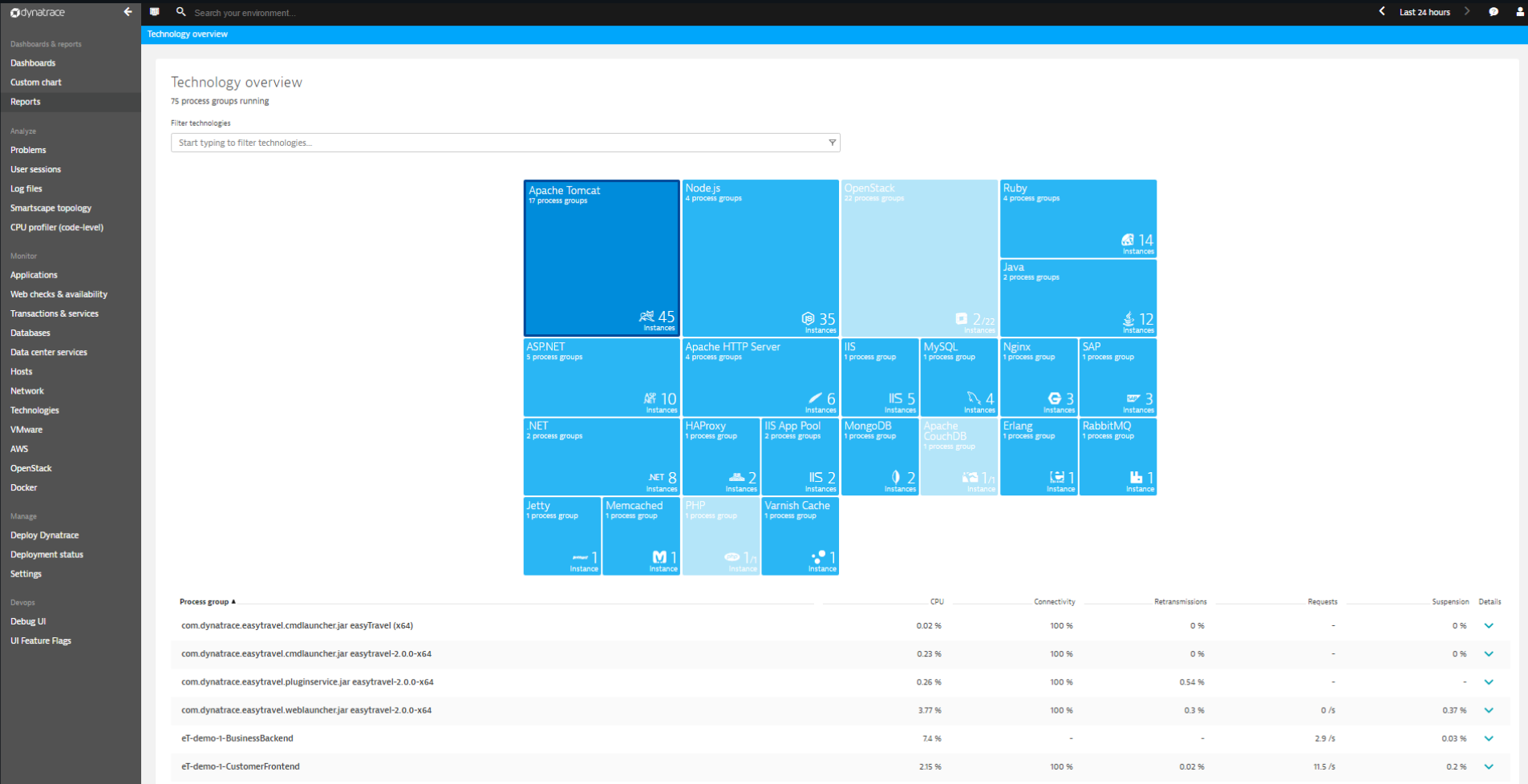
Analyzing Process Groups



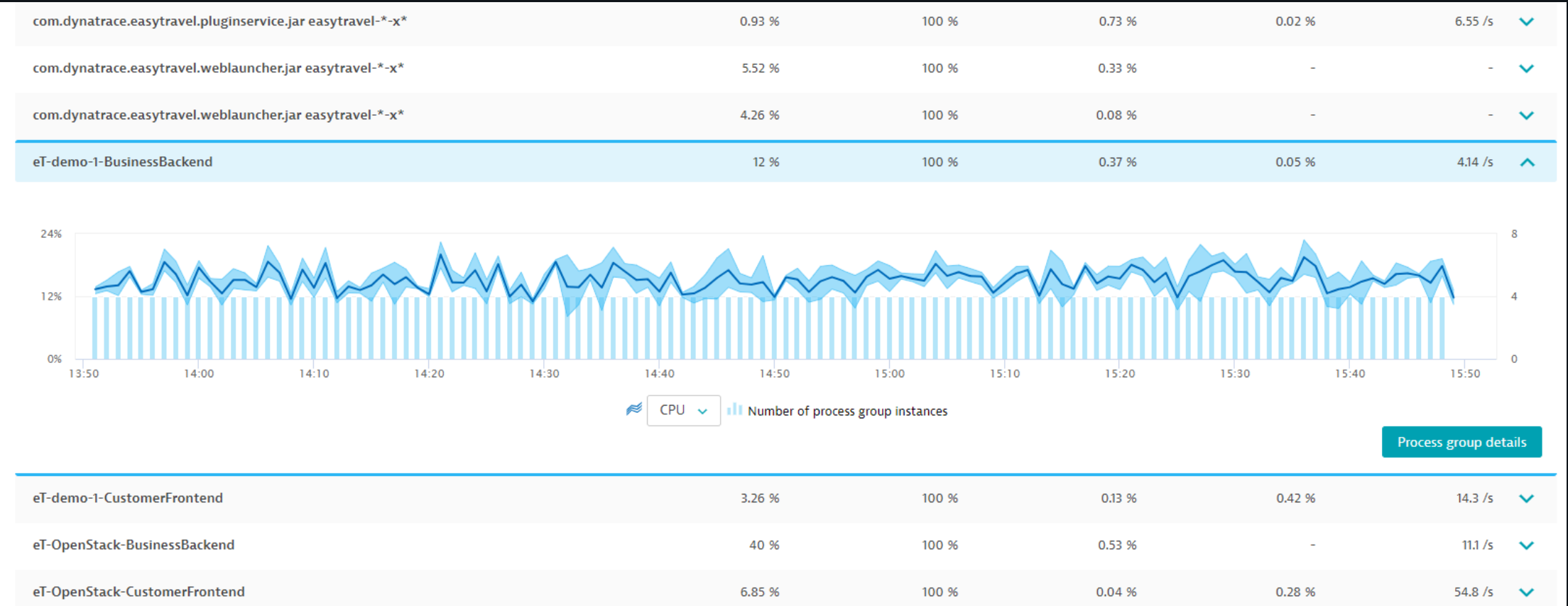
How do I analyze Process Groups?



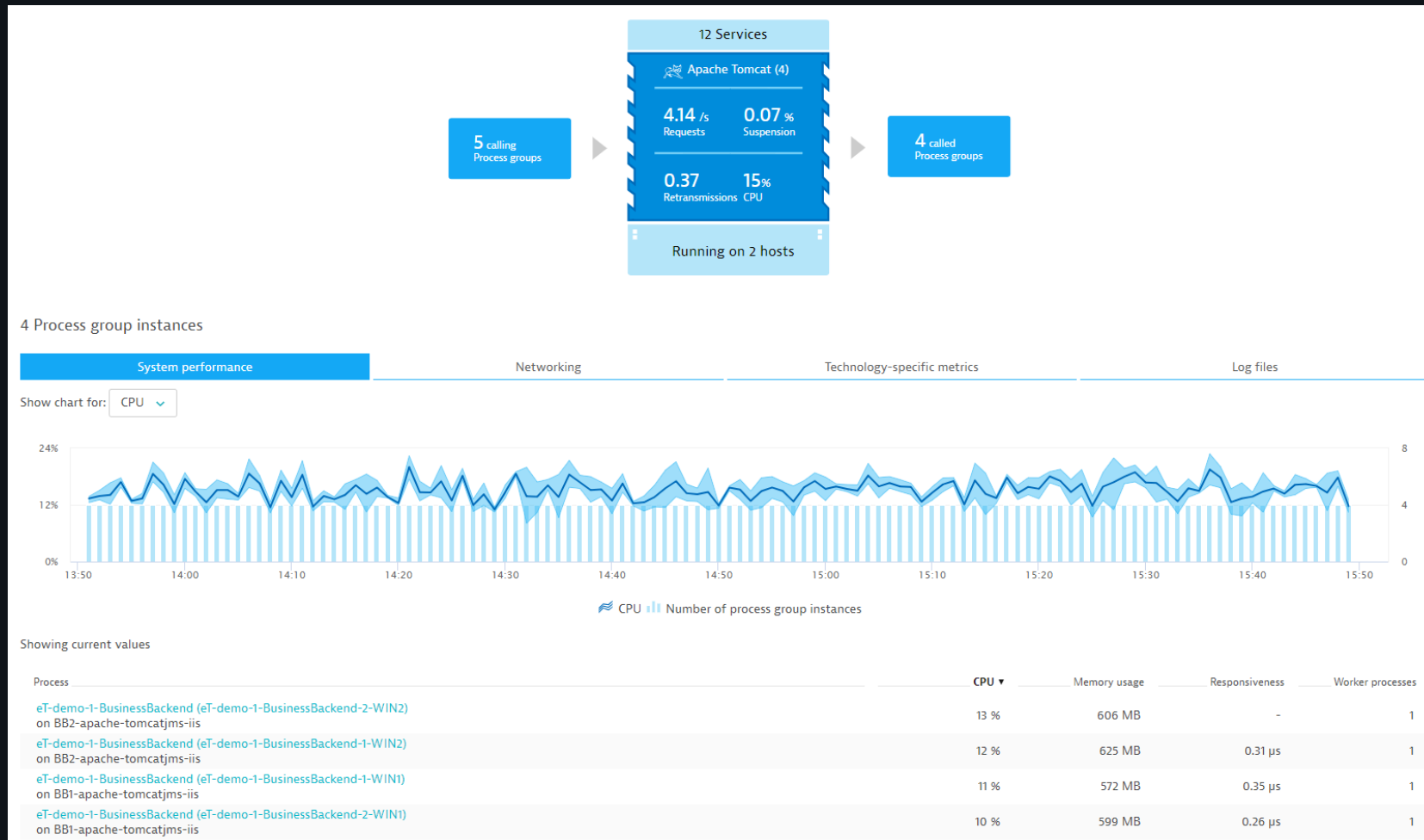
Technologies Monitoring



Process Group View



Process Group View



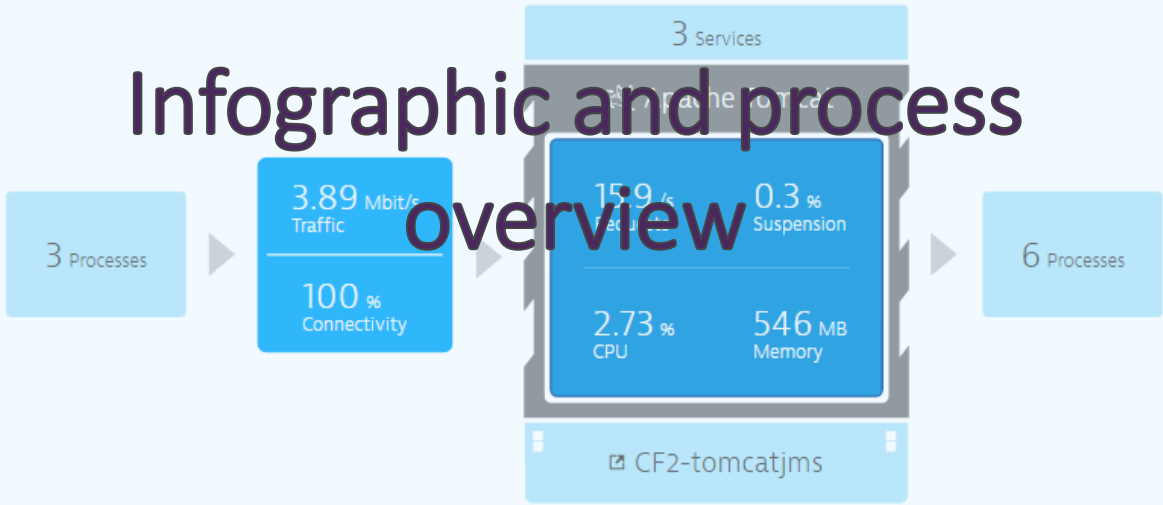


eT-demo-1-CustomerFrontend

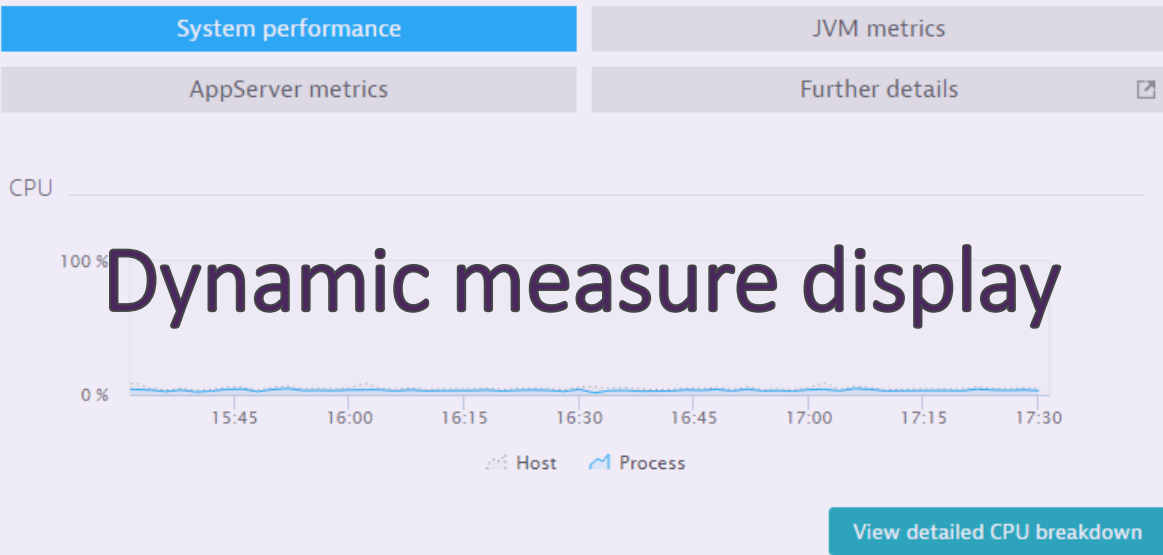
Analyze process connections



Properties



Infographic and process overview



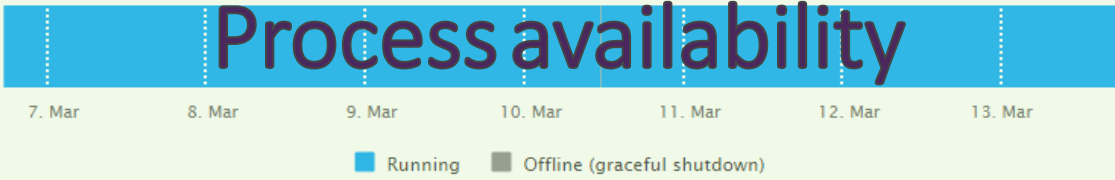
Dynamic measure display

No problems in last 72 hours

Problem history

watched

99.9% Availability in the last 7 days
6 min total downtime Last downtime on 10-03-2017 10:28 - 10:35



Process availability

No events today, 15:31 - 17:31

Event logger

1 log file updated in last 72 hours
This process writes to 1 log file on CF2-tomcatjms
Can't see important log entries? [Configure more log files](#)

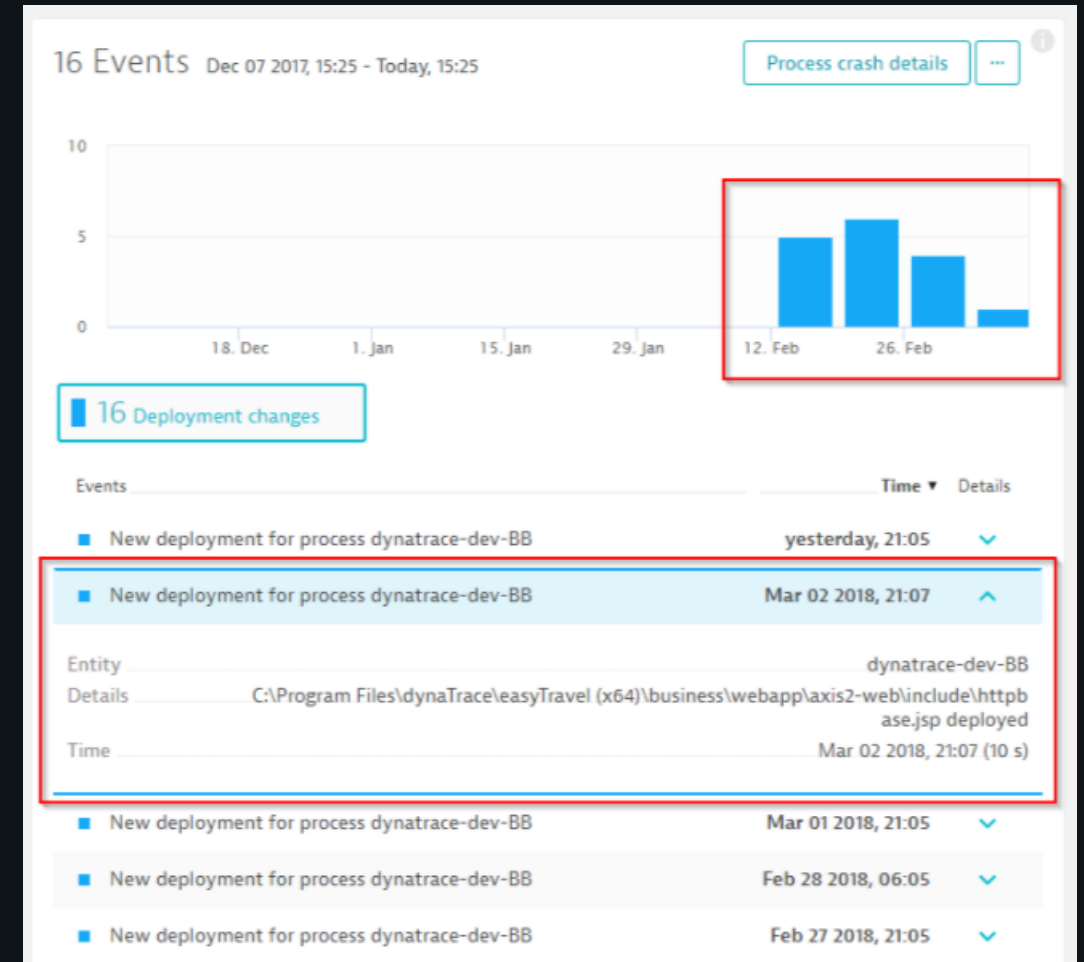
/home/labuser/dynatrace/trave-2019/easytrave/log/CustomerFrontend2427-logs (15.8 MB)

View all log entries

Log files for process

Event Analytics

- The Dynatrace events-analytics engine can process thousands of individual events, even over large analysis time frames
- Event analysis offers convenient drill-down and filtering options that make it easy to focus on specific points in time where high event activity occurred and then filter those events based on event type
- Most importantly, these events are those that Davis is monitoring for identifying problems and potential root causes



Traversing Your Stack

Hosts

What is a Process Group?

Analyzing Process Groups

What is a Service? ◀

Analyzing Services

Viewing and Filtering

What is a Service?

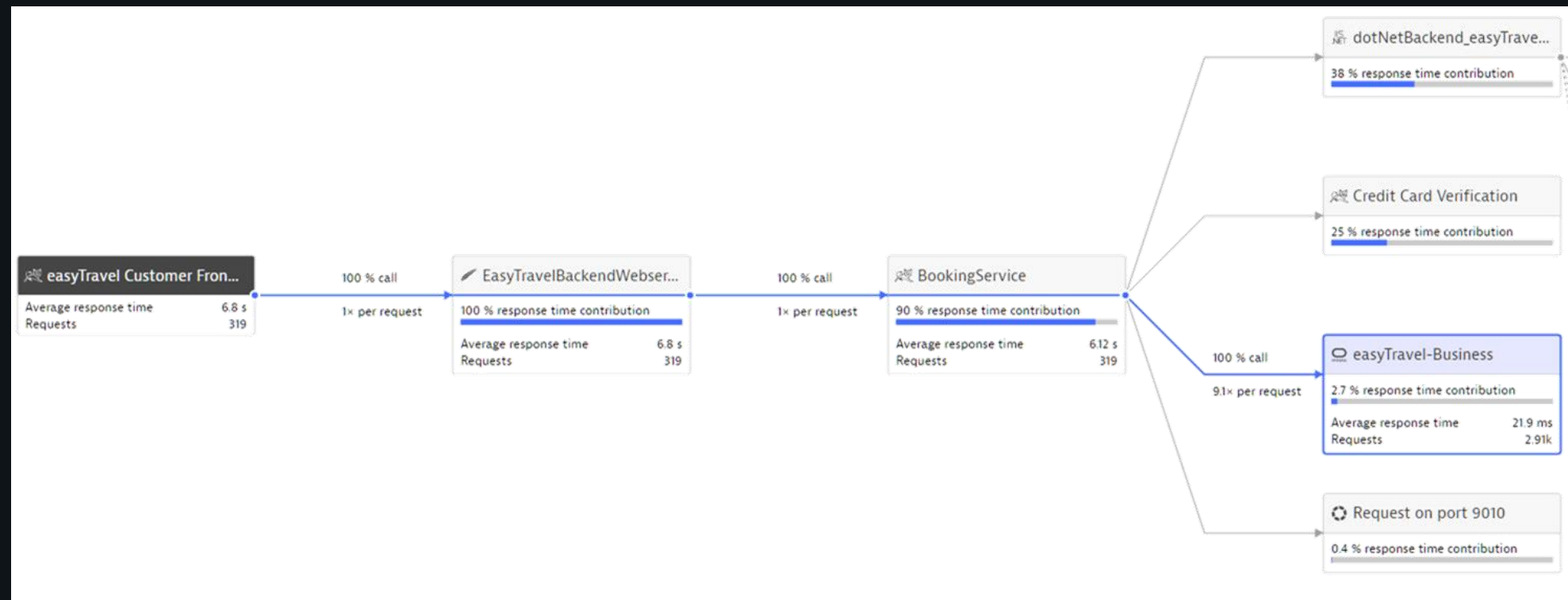


What are services

- Web applications consist of web pages that are served by web servers and web containers, for example Tomcat
- The web requests that are sent to a specific Tomcat server are an example of a server-side service
- Web and mobile applications are built upon services that process requests

What are services

- Such "server-side services" can take the form of web services, web containers, database requests, custom services, and more
- Services may in turn call other services



What are services

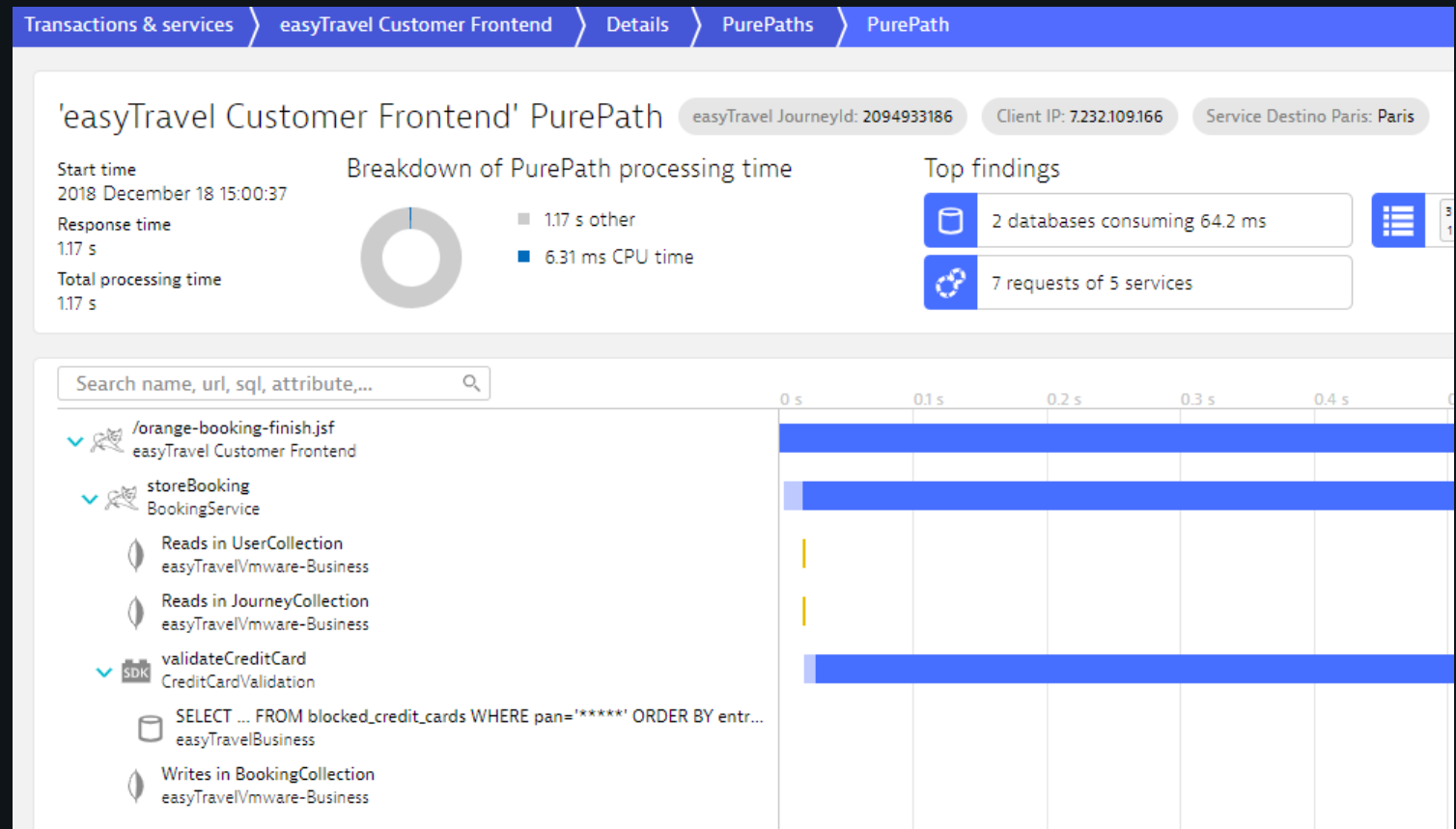
- Dynatrace monitoring of services extends all the way down to the monitoring of discrete methods
- By monitoring the performance of individual methods, we can offer you deeper insight into the performance of the services that drive your application
- Code level information is collected for each request
- Services are aggregated data of underlying PurePaths

Top web requests 08:54 - 09:09

Name	CPU vs total time consumption ▼	Median response time
/CalculateRecommendations		912 ms
/orange.jsf		180 ms
/orange-booking-finish.jsf		8.11 ms
/services/BookingService/storeBooking		1.18 s
/orange-booking-payment.jsf		9.02 ms
/services/JourneyService/findLocations		147 ms
Images		0.12 ms
/logout.jsf		160 ms
/contact-orange.jsf		8.01 ms
/orange-booking-review.jsf		25.4 ms

What is a PurePath?

- Timing and code level data context
- Recorded for every service
- More information later



Why Services?

- Dynatrace needs to monitor each tier in your application
- Needs to baseline at every tier!
- Baselining, and fault domain isolation work on Services and Service requests, “not” on PurePath

Top web requests 08:54 - 09:09		
Name	CPU vs total time consumption ▾	Median response time
/CalculateRecommendations	<div><div></div></div>	912 ms
/orange.jsf	<div><div></div></div>	180 ms
/orange-booking-finish.jsf	<div><div></div></div>	8.11 ms
/services/BookingService/storeBooking	<div><div></div></div>	1.18 s
/orange-booking-payment.jsf	<div><div></div></div>	9.02 ms
/services/JourneyService/findLocations	<div><div></div></div>	147 ms
Images	<div><div></div></div>	0.12 ms
/logout.jsf	<div><div></div></div>	160 ms
/contact-orange.jsf	<div><div></div></div>	8.01 ms
/orange-booking-review.jsf	<div><div></div></div>	25.4 ms

Database statements 08:55 - 09:10			Sort by response time
<pre>select location0_name as name1.2, from Location location0_ where (lower(location0_name) like '%[?] [%]') and (exists (select journey1_id from Journey journey1_ where journey1_destination_name=location0_name)) and (normalize location(? ?) is not null)</pre>	47.9 /min	22.6 ms	✓
<pre>{call verify_location(?)}</pre>	44.9 /min	20.5 ms	✓
<pre>select location2_name as col_0_0_, count(booking0_id) as col_1_0_, location2_name as name1.2, from Booking booking0_ inner join Journey journey1_ on booking0_journey_id=journey1_id and (journey1_tenant_name=?) inner join Location location2_ on journey1_start_name=location2_name group by location2_name order by</pre>	1.6 /min	16.9 ms	✓
<pre>select location2_name as col_0_0_, count(booking0_id) as col_1_0_, location2_name as name1.2, from Booking booking0_ inner join Journey journey1_ on booking0_journey_id=journey1_id and (journey1_tenant_name=?) inner join Location location2_ on journey1_destination_name=location2_name group by location2_name</pre>	1.6 /min	16.5 ms	✓
<pre>select booking0_id as id1_0_, booking0_bookingDate as bookingDate2_0_, booking0_journey_id as journey_id3_0_, booking0_user_name as user_name4_0_ from Booking booking0_ inner join Journey journey1_ on booking0_journey_id=journey1_id and (journey1_tenant_name=?) order by booking0_bookingDate desc fetch</pre>	4.8 /min	15.2 ms	✓
<pre>select journey0_id as id1_1_, journey0_amount as amount2_1_, journey0_description as description3_1_, journey0_destination_name as destination_name8_1_, journey0_fromDate as fromDate4_1_, journey0_name as name5_1_, journey0_content as content6_1_, journey0_start_name as start_name9_1_, journey0_tenant_name as</pre>	17.1 /min	14.1 ms	✓
<pre>select journey0_id as id1_1_, journey0_amount as amount2_1_, journey0_description as description3_1_, journey0_destination_name as destination_name8_1_, journey0_fromDate as fromDate4_1_, journey0_name as name5_1_, journey0_content as content6_1_, journey0_start_name as start_name9_1_, journey0_tenant_name as</pre>	5 /min	11.4 ms	✓
<pre>select count(booking0_id) as col_0_0_ from Booking booking0_ inner join Journey journey1_ on booking0_journey_id=journey1_id and (journey1_tenant_name=?)</pre>	6.2 /min	7.06 ms	✓
<pre>select sum(journey1_amount) as col_0_0_ from Booking booking0_ inner join Journey journey1_ on booking0_journey_id=journey1_id and (journey1_tenant_name=?)</pre>	1.67 /min	5.17 ms	✓
<pre>select journey0_id as id1_1_, journey0_amount as amount2_1_, journey0_description as description3_1_, journey0_destination_name as destination_name8_1_, journey0_fromDate as fromDate4_1_, journey0_name as name5_1_, journey0_content as content6_1_, journey0_start_name as start_name9_1_, journey0_tenant_name as</pre>	269 /min	4.26 ms	✓

Traversing Your Stack

Hosts

What is a Process Group?

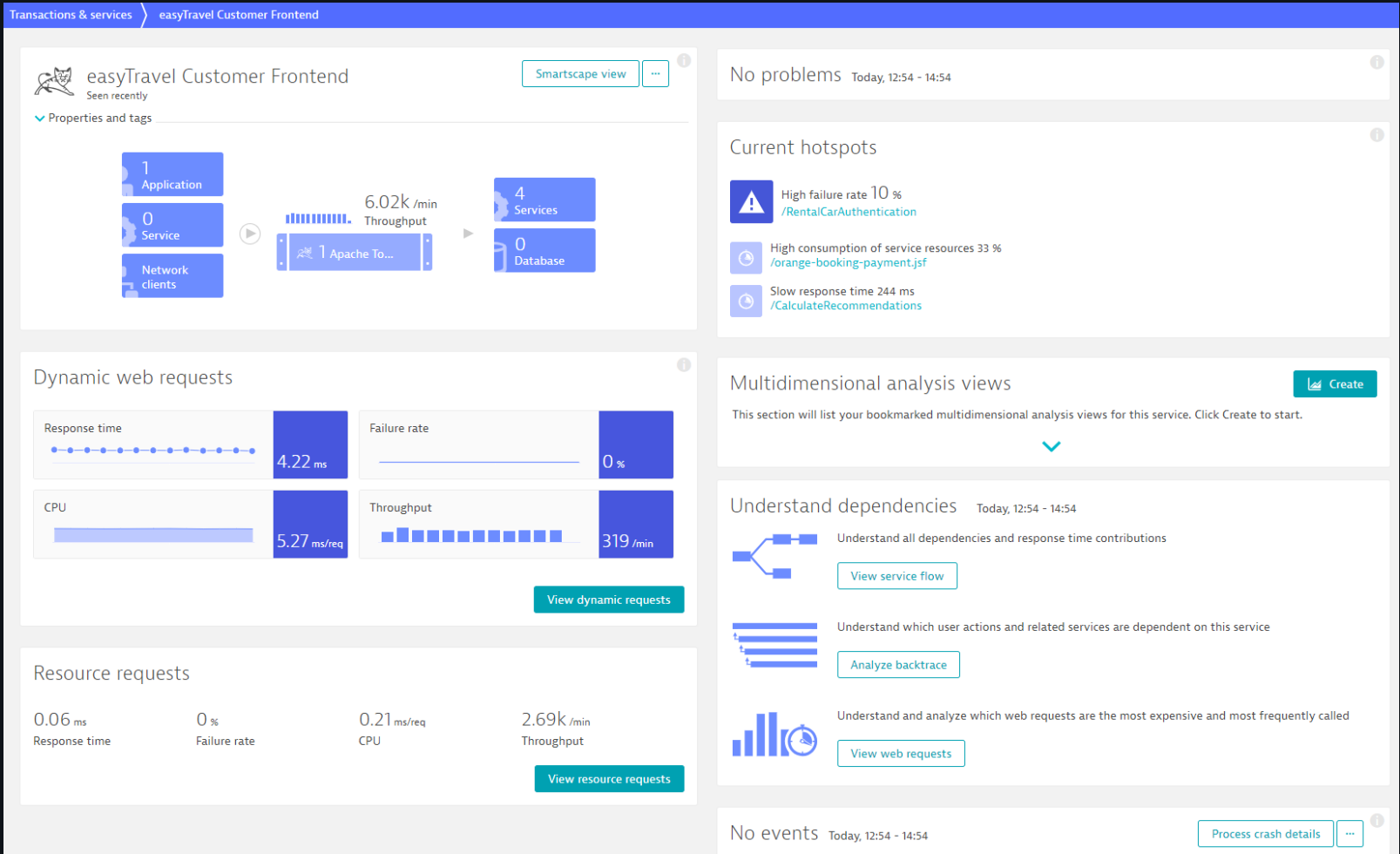
Analyzing Process Groups

What is a Service?

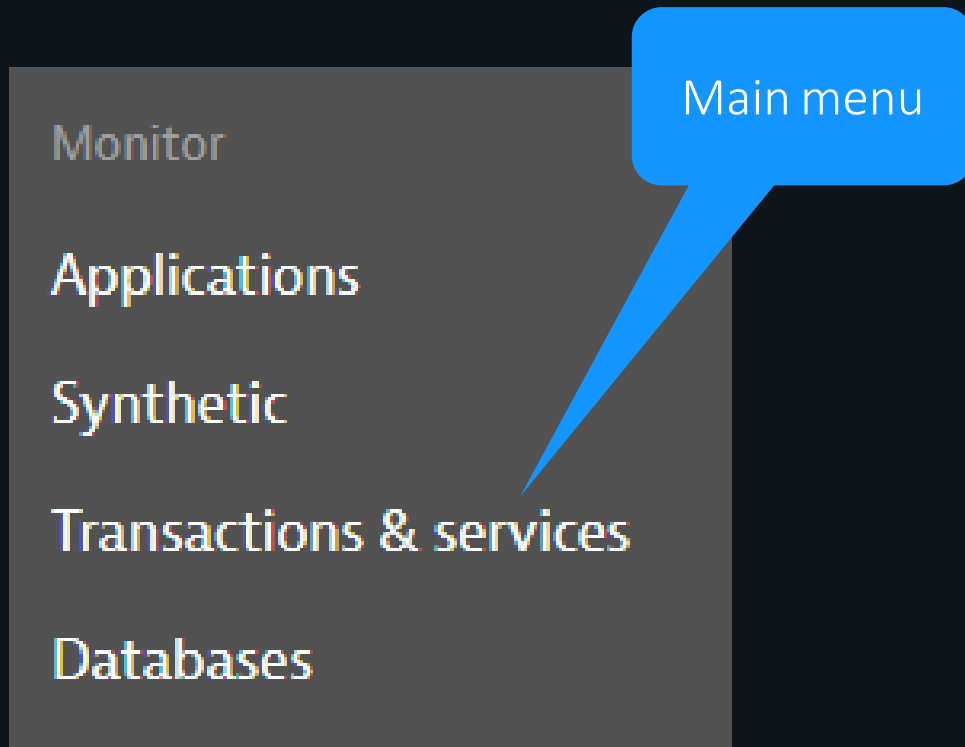
Analyzing Services ◀

Viewing and Filtering

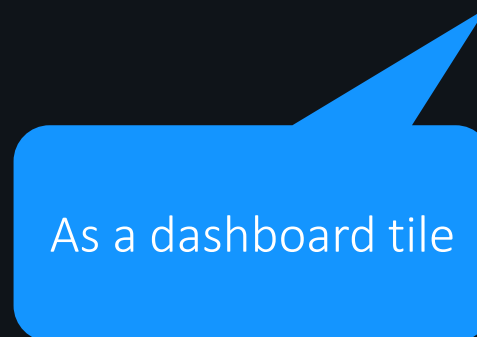
Analyzing Services



How do I analyze Services?



OR



Services Monitoring

←

🔍 Search Dynatrace demo1...

🔼

Dashboards & reports

Dashboards

Create custom chart

Reports

Analyze

Problems

User sessions

Log files

Smartscape topology

Diagnostic tools

Monitor

Applications

Synthetic

Transactions & services

Databases

Data center services

Hosts

Network

Technologies

VMware

AWS

Azure

Docker

Cloud Foundry

Kubernetes

Manage

Deploy Dynatrace

Deployment status

Settings

Transactions & services

⚙️ Services

Pin to dashboard

⋮

Filter by

Problem impact

any

edit

Service type

any

edit

Technology

any

edit

Tags

any

edit

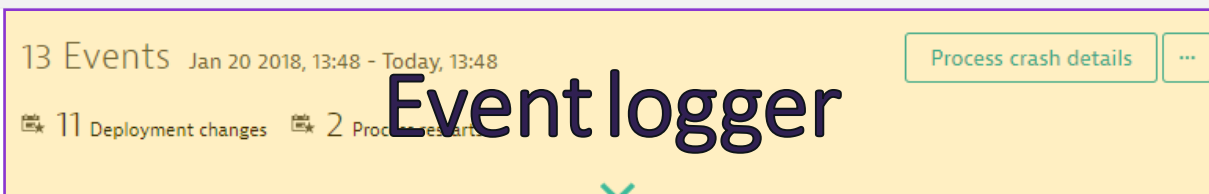
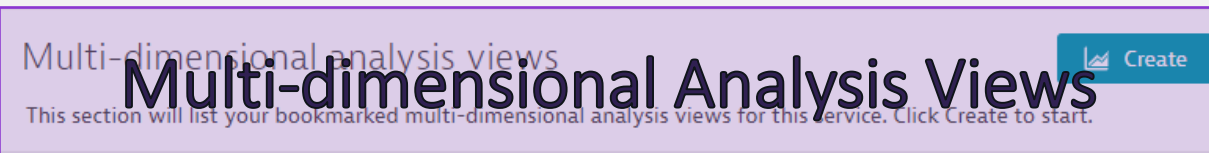
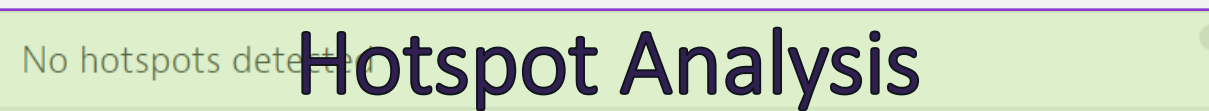
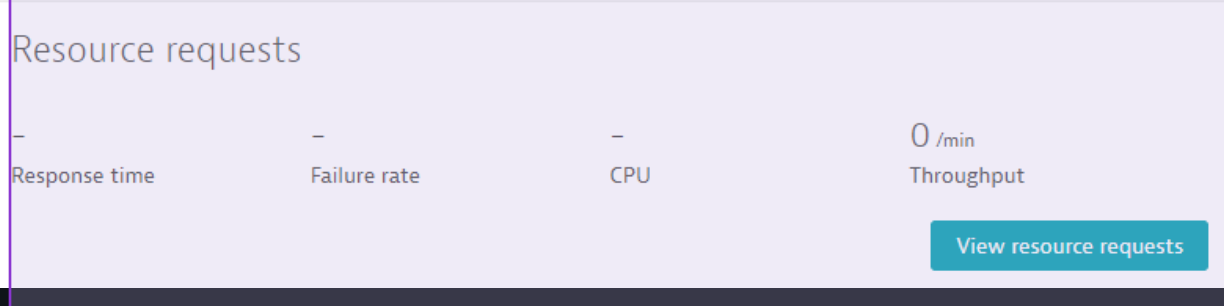
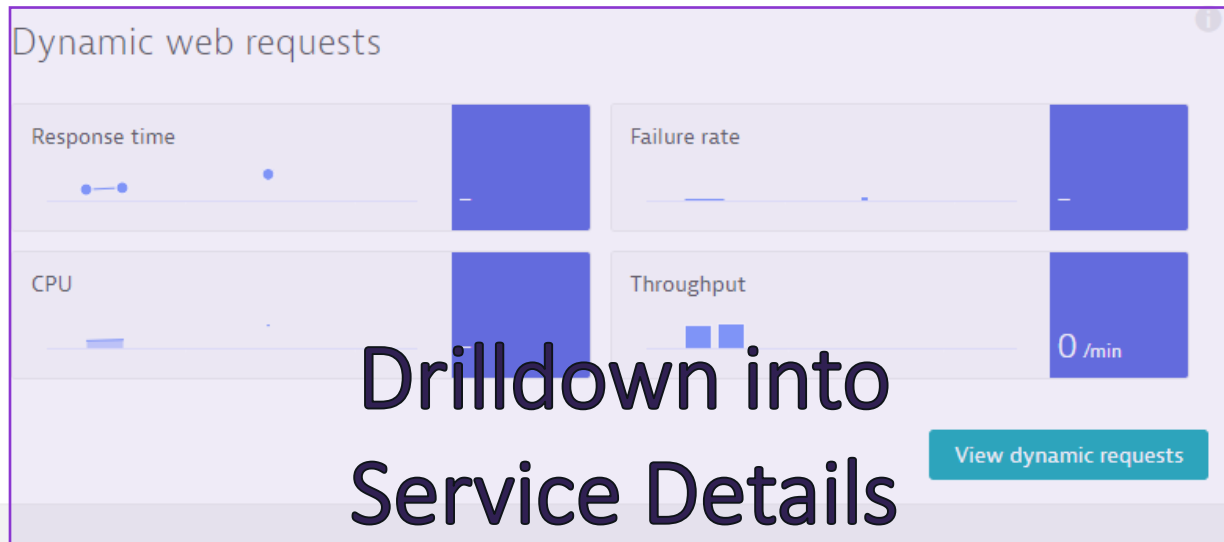
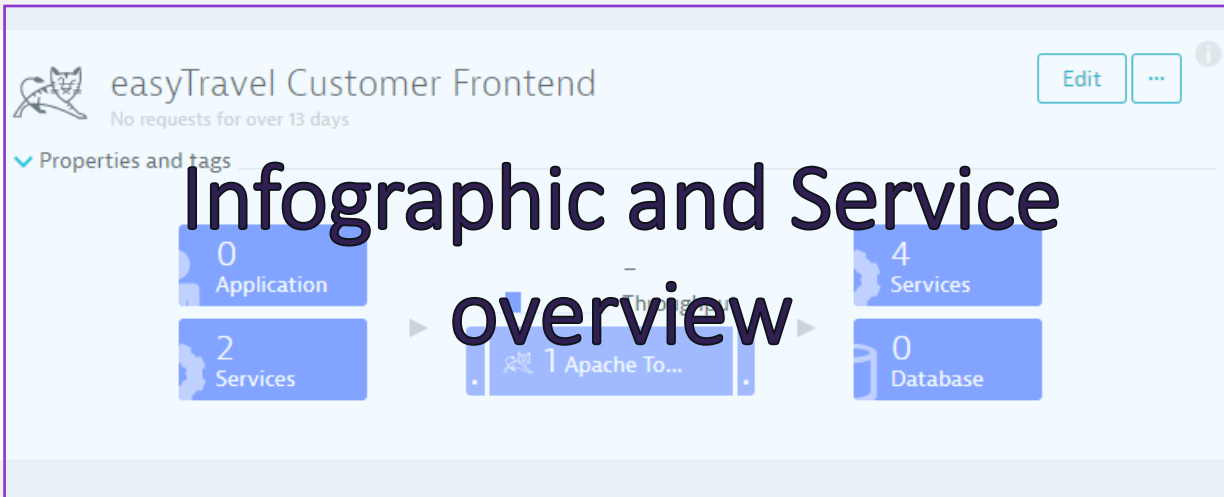
170 Services

showing

current values

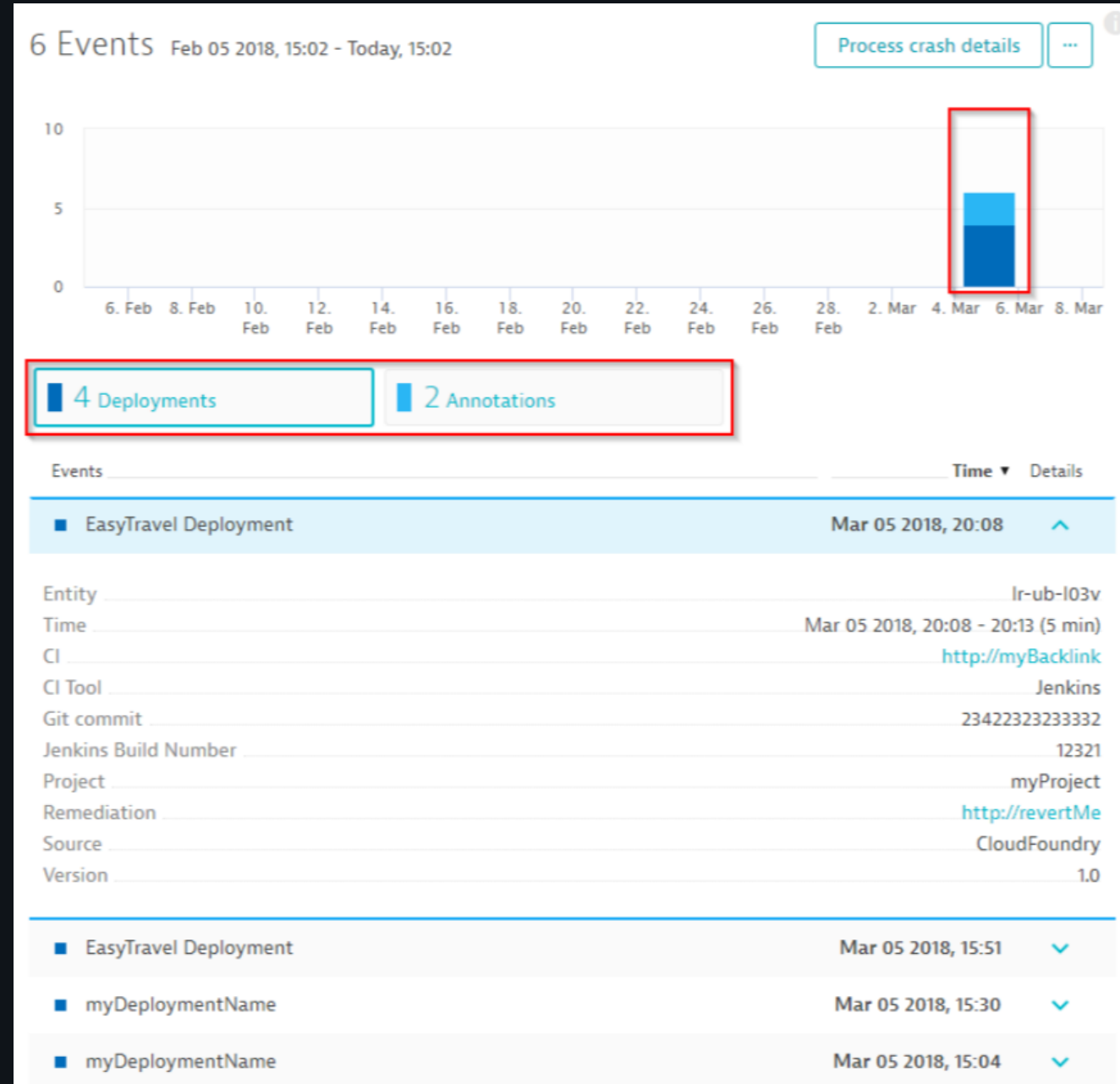
▼

Name	Response time median	Failure rate	Requests
fabric/SFDictionaryApplication/SFDictionaryService SFDictionaryService.exe	144 μs	0 %	36.5k /min
easyTravel Customer Frontend eT-vmware-demo-1-CustomerFrontend	237 μs	0 %	4.5k /min
easyTravel Customer Frontend eT-OpenStack-CustomerFrontend	143 μs	0 %	2.56k /min
EasyTravelWebserver:8079 eT-OpenStack-LoadBalancer	2.79 ms	0 %	2.56k /min
EasyTravelWebserver:8079 eT-demo-1-Frontend-LoadBalancer	2.03 ms	0.22 %	2.03k /min
easyTravel Customer Frontend eT-demo-1-CustomerFrontend	72.6 μs	0.23 %	1.92k /min
MF EasyTravelWebserver:8079 Apache Web Server easyTravel	1.11 ms	0 %	1.88k /min
MF easyTravel Customer Frontend com.dynatrace.easytravel.customer.frontend.jar easyTravel (**)	126 μs	0 %	1.88k /min
Requests executed in background threads of com.dynatrace.easytravel.launcher.CommandlineLauncher easytravel-*-* com.dynatrace.easytravel.launcher.CommandlineLauncher easytravel-*-*	1.49 ms	0.13 %	1.14k /min
easyTravel Customer Frontend com.dynatrace.easytravel.customer.frontend.jar easytravel-*-*	389 μs	0 %	968 /min
k8s easyTravel Customer Frontend k8s tomcat easytravel-frontend-*	87.7 μs	0 %	950 /min
CF demo-1 Requests executed in background threads of uemload.jar easytravel-uemload-demo* CF uemload.jar easytravel-uemload-demo*	2.02 ms	0.14 %	699 /min
EasyTravelWebserver:8999 eT-demo-1-Frontend-LoadBalancer	5.43 ms	0 %	695 /min



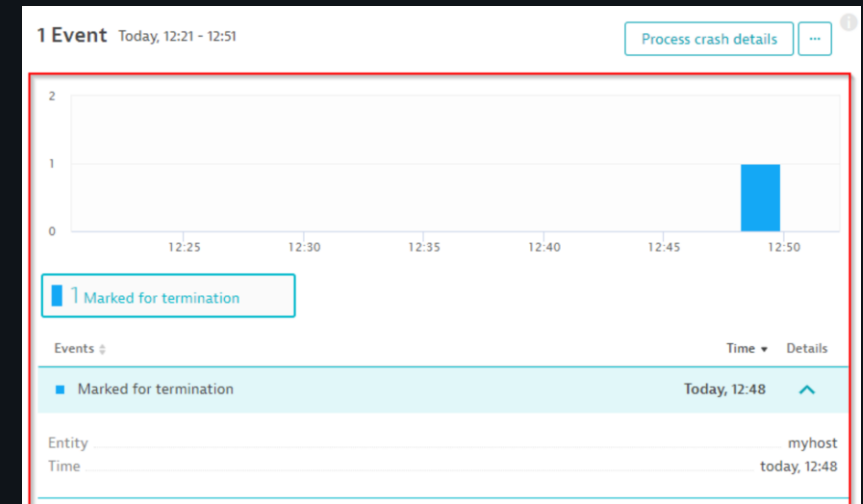
Event Analytics

- You can have custom events pushed into Dynatrace
- One popular use case is to have your continuous integration (CI) and build toolchain automatically report meta-information about software deployments
- Each custom event includes a set of custom key-value properties that your toolchain can use to report important context information



Custom Event – Marked for Termination

- Modern service infrastructure depends heavily on IT's ability to dynamically scale the number of hosts up or down, depending on the expected workload. Cloud providers, such as AWS, Azure, and GCP, help to automate the process of upscaling or downscaling compute power by providing autoscaling groups.
- Beyond cloud provider solutions, there are multiple additional frameworks and tools that help IT departments to dynamically adapt their compute power dynamically.
- To account for cases when Dynatrace cannot detect Auto-scaling events, you can inform Dynatrace Davis that a host is planned for shutdown or will be terminated within the next 60 minutes



Traversing Your Stack

Hosts

What is a Process Group?

Analyzing Process Groups

What is a Service?

Analyzing Services

Viewing and Filtering

Viewing and Filtering

←

Search Dynatrace demo1...

🔍

Dashboards & reports

Dashboards

Create custom chart

Reports

Analyze

Problems

User sessions

Log files

Smartscape topology

Diagnostic tools

Monitor

Applications

Synthetic

Transactions & services

Databases

Data center services

Hosts

Network

Technologies

VMware

AWS

Azure

Docker

Cloud Foundry

Kubernetes

Manage

Deploy Dynatrace

Deployment status

Settings

Transactions & services

⚙️

Services

📌 Pin to dashboard

⋮

▼ Filter by

Problem impact
any

edit

Service type
any

edit

Technology
any

edit

Tags
any

edit

170 Services

showing

current values

▼

Name	Response time median	Failure rate	Requests ▼
fabric/SFDictionaryApplication/SFDictionaryService SFDictionaryService.exe	144 μs	0 %	36.5k /min
easyTravel Customer Frontend eI-vmware-demo-1-CustomerFrontend	237 μs	0 %	4.5k /min
easyTravel Customer Frontend eI-OpenStack-CustomerFrontend	143 μs	0 %	2.56k /min
EasyTravelWebserver:8079 eI-OpenStack-LoadBalancer	2.79 ms	0 %	2.56k /min
EasyTravelWebserver:8079 eI-demo-1-Frontend-LoadBalancer	2.03 ms	0.22 %	2.03k /min
easyTravel Customer Frontend eI-demo-1-CustomerFrontend	72.6 μs	0.23 %	1.92k /min
MF EasyTravelWebserver:8079 Apache Web Server easyTravel	1.11 ms	0 %	1.88k /min
MF easyTravel Customer Frontend com.dynatrace.easytravel.customer.frontend.jar easyTravel (x*)	126 μs	0 %	1.88k /min
Requests executed in background threads of com.dynatrace.easytravel.launcher.CommandlineLauncher easytravel-*-* com.dynatrace.easytravel.launcher.CommandlineLauncher easytravel-*-*	1.49 ms	0.13 %	1.14k /min
easyTravel Customer Frontend com.dynatrace.easytravel.customer.frontend.jar easyTravel-*-*	389 μs	0 %	968 /min
k8s easyTravel Customer Frontend k8s tomcat easytravel-frontend-*	87.7 μs	0 %	950 /min
CF demo-1 Requests executed in background threads of uemload.jar easytravel-uemload-demo* CF uemload.jar easytravel-uemload-demo*	2.02 ms	0.14 %	699 /min
EasyTravelWebserver:8999 eI-demo-1-Frontend-LoadBalancer	5.43 ms	0 %	695 /min

Questions?



Simply smarter clouds